BMC Bioinformatics

Methodology article

Visualizing the genome: techniques for presenting human genome data and annotations

Ann E Loraine* and Gregg A Helt

Address: Bioinformatics Department, Affymetrix, Inc., 6550 Vallejo Street, Ste. 100, Emeryville, CA, 94608, U.S.A

E-mail: Ann E Loraine* - ann_loraine@affymetrix.com; Gregg A Helt - gregg_helt@affymetrix.com

*Corresponding author

Published: 30 July 2002

BMC Bioinformatics 2002, **3**:19

Received: 2 May 2002

Accepted: 30 July 2002

This article is available from: http://www.biomedcentral.com/1471-2105/3/19

Abstract

Background: In order to take full advantage of the newly available public human genome sequence data and associated annotations, biologists require visualization tools ("genome browsers") that can accommodate the high frequency of alternative splicing in human genes and other complexities.

Results: In this article, we describe visualization techniques for presenting human genomic sequence data and annotations in an interactive, graphical format. These techniques include: one-dimensional, semantic zooming to show sequence data alongside gene structures; color-coding exons to indicate frame of translation; adjustable, moveable tiers to permit easier inspection of a genomic scene; and display of protein annotations alongside gene structures to show how alternative splicing impacts protein structure and function. These techniques are illustrated using examples from two genome browser applications: the Neomorphic GeneViewer annotation tool and ProtAnnot, a prototype viewer which shows protein annotations in the context of genomic sequence.

Conclusion: By presenting techniques for visualizing genomic data, we hope to provide interested software developers with a guide to what features are most likely to meet the needs of biologists as they seek to make sense of the rapidly expanding body of public genomic data and annotations.

Background

The goal of the publicly funded human genome project is to provide all scientists with a reference genome sequence, and, in so doing, accelerate the pace of biomedical research. In addition to the sequence itself, the public data providers have also provided annotations on the sequence, notations describing the location and extent of genes and other biologically meaningful features. These genomic annotations typically include three basic types: single-base annotations such as such as the location of single-nucleotide polymorphisms (SNPs), single-span annotations such as the location and extent of individual transposable elements, and multi-span annotations such as the locations of a gene's complement of exons and introns inferred from cDNA-to-genomic sequence alignments or predicted by gene-finding programs [1,2].

These location-based feature annotations typically possess annotations of their own, such as scores describing their believability, information about the analysis programs used to generate them, what type of biological entity they represent, and other descriptive data. Although the basic

sequence data are valuable and useful, biologists are typically more interested in the higher-level, location-based annotations on the sequence, since these annotations relate the sequence data to biological pathways and systems. All three levels of genomic data can be adequately described using text, but biologists can make sense of the information more effectively when it is presented in an interactive, graphical format.

In recognition of the value of graphical representation, genomic data providers have developed graphical tools which generally follow a Web-based, client-server model in which server-side programs create image-mapped genomic "scenes" that are then displayed in the user's browser. Two prominent examples of this Web-based approach include the LocusLink (N.C.B.I.) evidence viewer [3] and the U.C.S.C. genome gateway's genome browser [4]. These and similar Web sites provide valuable services, but are limited by the inability of a client-server model to provide a truly interactive user experience, since each interaction with the genome scene on display requires a round-trip from the user's desktop to the server and back again. Furthermore, the image-map format is currently limited in its ability to support gestural interactions, such as dragging scrollbars to pan the display or change its magnification.

When exploring a genomic region, biologists typically need to interact with the scene in a much richer fashion than is currently possible using simple, hyperlinked images. As biologists begin to explore a genomic scene, they formulate new questions about what they see. In order to answer these questions, they often need to modify what they see, such as by adding or deleting data, panning to the left or right, or changing the scale of the view. The client-server model is limited because it constrains navigation, often requiring multiple clicks to make the simplest adjustments. Hybrid approaches such as interactive, Java applets downloaded from a server and which run in a Java virtual machine on the user's desktop, have also been attempted [5], but Java applets are problematic in that security concerns restrict their ability to load data files located on the user's personal computer. Thus to gain the full benefit of genome project data, users require desktop software that can present the data in a fully interactive environment conducive to exploration and which also allows users to view their own custom data. In this article, we describe techniques for presenting human genomic sequence data and annotations in an interactive, graphical format, using examples from a prototype protein domain viewer (ProtAnnot) and from the Neomorphic GeneViewer, part of a genome browser and annotation tool written initially for The Institute for Genomic Research (TIGR) to support annotation of the *Arabidopsis* genome. Our aim in this paper is not to showcase these two applications, but rather to provide interested software developers with a guide to what features are most likely to meet the needs of biologists.

Results

Representing gene structures

Representing gene structures is probably the most crucial aspect of any genome browser application since genes and the proteins they encode are relevant to all aspects of biomedical research. In light of this, we focus here on techniques for representing the key elements of human gene structures and their encoded proteins. In this scheme, a gene structure is defined as the relative placement of feature elements that make up a gene onto a single linear axis defined by the DNA sequence. Feature elements that make up a gene include: exons, 5' and 3' untranslated regions, coding regions, start and stop codons, introns, 5' transcriptional control elements, 3' polyadenylation signals, splice site boundaries, and protein-based annotations of the coding regions. Each of these feature types require specialized methods of presentation that depend upon the type of data being shown.

Visualizing sequence using semantic zooming

Although gene structures and their annotations are mainly what interest biologists, the ability to view the sequence data in the context of a gene structure is a critical feature of any genome browser application [5]. In order to interpret and assess a proposed or known gene structure, biologists need to be able to inspect individual bases that influence the gene structure's believability. For example, to assess whether the reported 3' end of a gene is correct, a biologist may wish to search the sequence near the end of the final exon for putative polyadenylation sites. Similarly, a biologist may wish to examine the dinucleotide sequence at the 5' and 3' boundaries of putative introns, since these bases are highly conserved in human genes [6].

Figure 1 demonstrates how semantic zooming [7], a visualization technique in which objects change their representation according to their level of magnification (zoom level), can be used to convey sequence information alongside a representation of a gene structure. Figure 1a shows a gene structure at low magnification that was inferred from the alignment between a cDNA sequence and its putative genomic region of origin. The pattern of aligned spans, shown as tall rectangles, defines a gene structure containing six putative exons and five putative introns, including one unusually small intron interrupting the 3' UTR. Human introns are typically much larger than this (Ann Loraine, unpublished results), and so a biologist attempting to evaluate this gene might doubt whether this intron is correct.

Figure 1b shows a close-up view of the questionable intron together with the genomic sequence. The zoomed-in

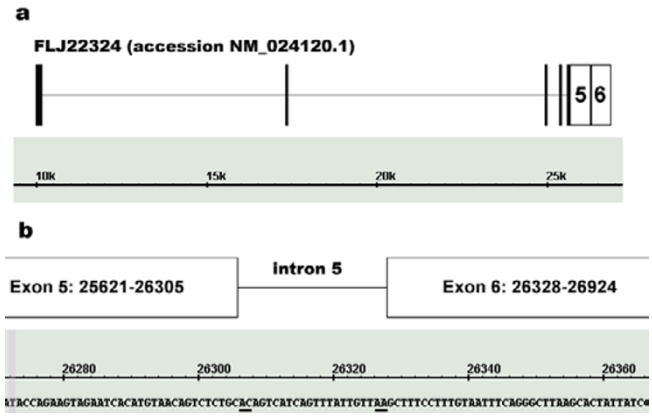

Figure 1
Annotated GeneViewer screen capture showing FLJ22324, a six-exon gene inferred from a cDNA-to-genomic sequence alignment.(a) The high-level structure at low zoom. **(b)** A close-up view of a questionable small intron separating exons 5 and 6. Dinucleotide bases at this intron's 5' and 3' boundaries are underlined.

view in Figure 1b reveals that the bases flanking the intron deviate from the expected "GT--intron--AG" consensus sequence for intron boundaries, thus lending credence to the idea that this insertion in the genomic sequence relative to the aligned cDNA may represent an alignment error, polymorphism, or experimental artifact rather than a true intron. In this example, changing the scale of the display (zooming "in") allows the user to inspect the genomic sequence while at the same time maintaining a sense of context. In other words, when looking at individual bases at high magnification, the user is able to maintain a sense of place because the sequence appears alongside the same landmarks that were visible at low zoom.

Another use of semantic zooming involves labeling or otherwise annotating display elements with text. For example, at low zoom, an object may be too small to fit a label, such as with the 5' exons in Figure 1a. At high zoom, objects take over more screen real estate and therefore become large enough to show more detailed labels. Thus semantic zooming allows the primary sequence data, location-based features on the sequence, and text-based annotations on these features to be presented together in a single highly adjustable scene. Ideally, the user should be able to select items in the display and retrieve further information about the gene being shown. One natural solution would be to implement hyperlinks within the display that would allow the user to open a browser window showing information retrieved from on-line databases, such as GenBank and PubMed.

One- versus two-dimensional zooming

Zooming as a visualization technique for display and navigation of complex data sets has typically been implemented in two dimensions. For example, applications built using the Java-based Jazz toolkit for graphical user interfaces [8] provide point-based or "camera" zooming, in which the operation of zooming is best understood as a change in height of a virtual camera poised above a single point in the display. That is, in camera-based zooming, the entire scene appears to expand or contract in every direction around a central focal point as the user zooms in or out.

Genome browser applications represent a one-dimensional world in that they display location-based features across a single axis defined by the genomic sequence data itself. In genome browser applications, the axis perpendicular to the sequence axis has no meaning and therefore could be used to sort information based on feature or analysis type, as will be discussed later. Thus, for display of genome data, camera-based, point-centered zooming as provided in Jazz is not appropriate, since DNA sequence and its annotations are one-dimensional. As with Jazz-based applications, the focus for zooming should still be the point where the user last clicked, but the result of zooming should be a stretching of the sequence axis rather than a change in the user's relative height above the genomic scene. Although Jazz has been designed to support exploration of two-dimensional data spaces, it may be possible to restrict its zooming behavior to single dimensions. Since the source code is freely available, we urge readers interested in building genomic viewers to investigate the Jazz toolkit [9].

Single and dual-sequence annotations

Gene structures are typically deduced using one of two methods. The simplest type of gene structure prediction is based on output from gene-finding programs that analyze the primary genomic sequence without reference to any other sequence. For example, gene prediction programs produce simple gene structures based solely on analysis of primary genomic sequence data. These simple annotations may easily be shown as linked, multi-span annotations representing the complement of exons that make up these hypothetical gene structures.

Although sequence analysis programs are useful tools, their ability to accurately describe human gene structures is severely hampered by the complex nature of human genes [6]. Approximately a third to over half of all human genes produce multiple transcript variants [6,10], and few gene prediction programs are able to identify more than one variant per gene. Despite many years of development, gene-finding programs are still limited in their ability to accurately describe human gene structures, and for this

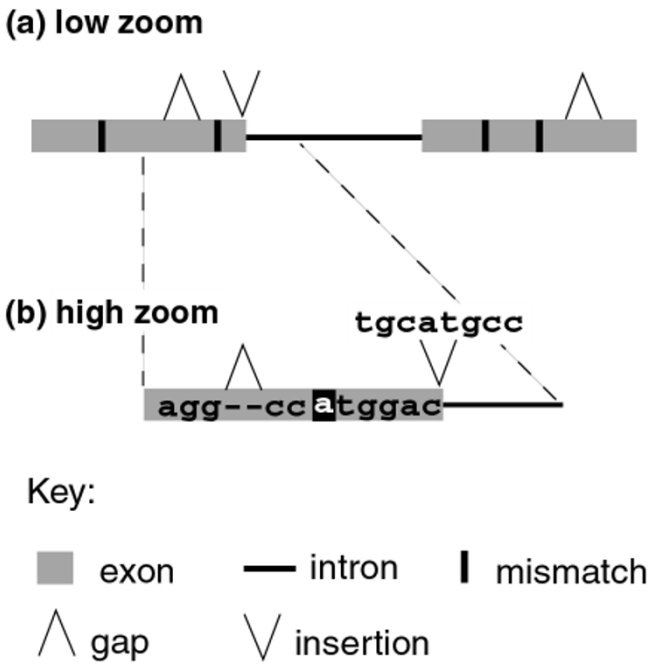

Figure 2
Hypothetical example showing how semantic zooming could be used to represent gene structure annotations based on cDNA-to-genomic sequence alignments.(a) Low zoom. **(b)** High zoom.

reason, applications that display them should make their hypothetical nature clear to the user.

In contrast, a dual-sequence or pairwise annotations describe the relationship between the genomic sequence and the independently produced sequence of some other biological molecule. For example, programs such as sim4 [11] and blat [12] are designed to align cDNA and genomic sequence and are often used to infer gene structures in genomic DNA. Regions in the genomic sequence that match regions in the cDNA sequence typically are used to delimit exons, while gaps in the cDNA partner of the alignment that exceed some pre-defined length threshold are typically used to delimit introns.

Annotations like these are often more valuable than simple gene predictions because they incorporate more information, such as experimental evidence for expression provided by cDNA sequence. And because they incorporate more information, their representation in a genome browser requires a more sophisticated approach that shows the exon-intron organization of the inferred gene structure as well as the alignment that was used to produce it.

Figure 2 presents a hypothetical example of how semantic zooming could be implemented in order to provide users with detailed information about how an alignment between a cDNA and genomic sequence was used to generate a gene structure. At low zoom (2a), insertions in the cDNA partner of the alignment are shown as "V" characters. These insertions represent bases that were present in the cDNA sequence but which were absent in the alignment between the cDNA and the genomic sequence. A biologist might interpret these insertions to mean that the genomic sequence in this region is incorrect, contains a run of "N" ambiguity characters representing a gap in the assembly, or that the region itself is polymorphic.

Similarly, exons inferred from aligned spans may contain short gaps in the cDNA portion of the alignment which were too short to be interpreted as introns. The low zoom image in Figure 3a represents these gaps as inverted "V" shapes. Single-base disagreements between the genomic sequence and the aligned cDNA, such as would be caused by single-nucleotide polymorphisms, are indicated by black rectangles superimposed on the inferred exons.

The main difference between the low zoom (3a) and the high zoom (3b) views is that at high zoom, the sequence of the cDNA partner in the alignment is shown. At this higher level of magnification, the sequence of the cDNA is shown superimposed on the exons. Gaps within exons are represented in the usual fashion, while the sequence associated with each insertion is displayed above the "V" character.

Dealing with complexity

The number and type of annotations can vary enormously from region to region depending both on the sequence itself as well as the number and types of analyses that have been done. Thus, it is difficult to design a program that can arrange sequence-based annotations in a fashion that is not overly confusing or complex. However, since genomic sequence has only one important axis, which can be shown either in vertical or horizontal orientation, the application developer can use the other axis to organize information in ways that expose biologically meaningful patterns in the data, such as regions that are densely annotated with ESTs and likely to be highly expressed.

One commonly-used technique for dealing with complexity of genomic annotations is to sort items into horizontal tiers based on some common attribute, such as the kind of analysis that was done to produce them. This approach is being used by the U.C.S.C. genome browser, which provides multiple rows or "tracks" featuring analyses contributed by many different groups [4].

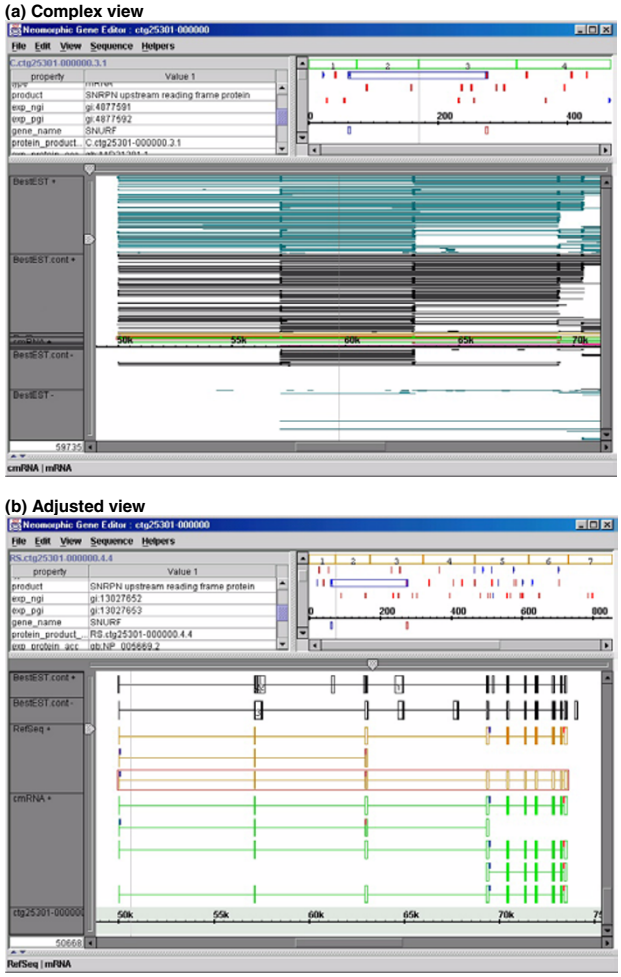

Figure 3
SNURF locus.(a) The full scene is shown with multiple annotation types sorted into labeled tiers. **(b)** A simplified scene is shown in which several tiers shown in (a) have been hidden, collapsed, or moved to new positions. The horizontal slider has been used to expand the display in the vertical direction.

Even so, as the available analyses accumulate, the potential for creating very complex scenes increases greatly. Although it is good to have more access to more data, for the purposes of understanding a gene it is important to give the user the power to simplify or re-organize the scene as needed. Sorting items into distinct tiers that can be moved, hidden, collapsed, or stretched is one way that application developers can give users greater freedom to modify a display when the amount of data being shown exceeds the user's ability to comprehend the full scene.

An example of a complex scene organized into adjustable tiers is shown in Figure 3, which shows two screen captures from the GeneViewer display tool. Both screen captures present a view of the human SNURF gene, which gives rise to an unusual bicistronic transcript encoding two different proteins [13]. Figure 3a presents a complex view of the locus, shown here as annotated with hundreds of features derived from EST-to-genomic and cDNA-to-genomic sequence alignments.

The screen capture in Figure 3b shows a simpler version in which the display has been simplified by collapsing, moving, and hiding several tiers. The tiers containing features based on EST-to-genomic sequence alignments have been collapsed, thus forming primitive clusters summarizing all the expressed regions associated with this locus. The plus- and minus-strand EST tiers have also been moved to the position immediately above the tiers showing cDNA-to-genome alignments (labeled cmRNA+ and RefSeq+), thus making it easier to compare the boundaries of items shown in each tier. This feature is especially important to users interested in alternative splicing, since the presence of ESTs that align to regions not covered by cDNA-to-genomic alignments can indicate the existence of novel variants.

Another way to help users visualize a genomic scene using tiers is to allow the user to stretch tiers in the direction perpendicular to the sequence axis. Allowing zooming in the vertical direction, as is shown in Figure 3b, accommodates situations where the entire scene cannot fit into the viewable area. In addition, this kind of vertical stretching allows users with visual disabilities to make selected regions of the scene larger and easier to see.

Protein in the context of genomic sequence

Although the intron-exon organization of genes is interesting, biologists typically are more interested in the proteins that genes encode. Therefore, a genomic viewer should incorporate information describing how the genomic sequence is translated into protein.

One common convention for representing coding regions is to show translated and untranslated regions as shaded and unshaded boxes, respectively. This convention has been used for many years to distinguish coding regions in the print medium and therefore has the advantage of leveraging users' expectations of how gene structures should look when presented in software.

Merely indicating the translated regions is not sufficient, however, because many genes give rise to multiple transcript forms due to alternative splicing, alternative promoter choice and other mechanisms that generate transcript and protein diversity. In many cases, alternative transcript structure causes shifts in frame. To accommodate this, we suggest that genome browsers use shading,

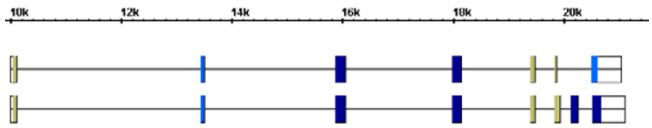

Figure 4
Using color to represent frame of translation at the ARG1 locus. Coding regions in each exon are colored according to which frame of the genomic sequence is translated. A different color for overlapping exons from different transcripts indicates these exons are translated in different frames.

color or some other method to indicate the frame of translation for each coding exon. Figure 4, a screen capture from the ProtAnnot prototype viewer, provides an example of how color can be used to indicate frame. In this case, alternative splicing at the ARG1 (arginase 1) locus [14] produces two distinct variants which differ at their 3' ends. The result is that the final exon in both variants is translated in two different frames, as can easily be seen by comparing their color. Similar examples are common among human genes (Ann Loraine, unpublished observations), and therefore understanding alternative transcript structure requires a depiction of translation frame in addition to the relative placement of introns and exons along the sequence axis.

In addition to providing a visual representation of translation frame, a genome browser should also provide a visual representation of motifs that are embedded in the protein sequence. In recent years, numerous protein sequence analysis methods have been developed that allow researchers to identify conserved functional domains within protein sequence. Since alternative transcripts arising from the same gene can often produce protein isoforms that differ with respect to their domain composition, tools that display these protein domains in the context of the genomic structure of genes can help biologists better understand how alternative transcript structure affects protein function.

Figure 5 presents a visualization of the human PLAT locus, which encodes at least two distinct proteins. Both are forms of a tissue-type plasminogen activator, a secreted serine protease which activates another protease (plasmin) responsible for dissolving blood clots [15]. As shown in Figure 5, these forms differ with respect to their protein domain composition. Regions in each that encode high-scoring matches to entries in the Pfam database [16] are shown below each variant as linked green spans. Both forms contain Kringle, Serine protease, and EGF-like motifs, but one form (b) contains an additional Type I Fibronectin motif that is not present in the other, a result of alternative splicing of the third exon. The other form (a) lacks this motif and is expressed in melanotic melanoma; this suggests that form (a) may play a role in the pathology of cancer cells, and that its lack of the fibronectin motif, named for the extracellular matrix protein fibronectin, may be involved. By showing both forms in the same scene, together with protein annotations, ProtAnnot provides a summary view of how alternative splicing affects biologically meaningful features in the protein sequence. Since a high proportion of human genes produce multiple forms, browsers that project protein annotations onto genomic sequence could help biologists investigate the functional significance of alternative splicing and other mechanisms that generate transcript diversity.

Discussion

Several efforts are now underway to build interactive, desktop genomic data visualization software. For example, the open source Apollo project, a collaboration between the Sanger Center and the Berkeley Drosophila Genome Project, is now building a desktop Java-based application that supports viewing and editing of genome annotations. FlyBase is using Apollo to re-annotate the *Drosophila* genome, and source code and compiled versions are freely available for download [17]. Other publicly available visualization software packages now being developed include OmniGene [18] and the GUI packages associated with BioJava [19] and BioPerl [20].

Although most such browsers provide some mechanism to support one-dimensional zooming, to our knowledge, none provides interactive, dynamic zooming in which the scene changes its appearance in response to gestural interactions. Instead, these applications have typically implemented zooming interactions using graphical

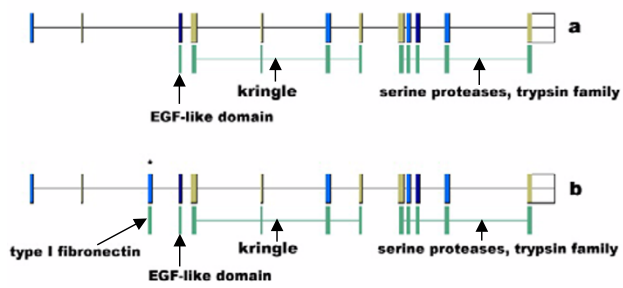

Figure 5
Protein motifs detected by Pfam are displayed beneath alternative transcript structures (a,b) at the PLAT locus. Alignments between genomic sequence and cDNA sequences **(a)** BC002795.1 and **(b)** NM_0009301 are shown. Regions encoding matches to Pfam motifs PF00008 (EGF-like domain), PF00051 (Kringle), PF00089 (Serine proteases, trypsin family), and PF00039 (Type I fibronectin) are shown as linked green rectangles below each alignment.

components such as buttons to allow users to change the scale of the view in a stepwise fashion. One problem with this solution is that these stepwise interactions often require multiple point-and-click operations to achieve the desired zoom level. Using sliders, scrollbars, or timed key presses to signal a request to change the zoom level could alleviate this problem by allowing the user to adjust the scene in a single motion.

Similarly, few genome browsers implement visual encoding of frame to indicate how alternative splicing impacts the encoded proteins. Since probably half of all human, multi-exon genes produce multiple variants, a genome browser written to display human genomic data should somehow incorporate a visual representation of frame. Although we have used color-coded exons to indicate this, other techniques, including shading or some form of labeling, could also be used. For example, WormBase, a Web-based interface to the *C. elegans* genome, recently began showing coding regions separately from genes structures as a series of color-coded rectangles sorted vertically according to frame of translation [21]. We would urge other developers to adopt similar conventions for representing coding frame, since gene structures are most meaningful when they include information about how each exon contributes to the sequence of the encoded proteins. And since biologists are typically most interested in the proteins that genes encode, genome browser developers could best serve their users by incorporating protein sequence annotations, such as matches to protein sequence profiles, into genomic scenes. We have described one way to do this using the prototype ProtAnnot application, but other techniques besides those we have presented here might be used.

Conclusions

In support of the public, open source visualization projects, we have presented techniques for displaying human genomic sequence data, including semantic zooming for display of sequence and alignment information; use of color to indicate the frame of translated exons; adjustable, moveable tiers to permit easier inspection of a genomic scene; and display of protein domains in the context of genomic sequence to facilitate functional interpretation of transcript variants.

Biology, more so perhaps than other scientific disciplines such as mathematics and physics, relies heavily on visual representations to communicate results, including images such as micrographs that show primary data, as well as images that convey interpretations of the primary data, such as diagrams of gene structures. Genome sequence data and annotations are more complex than traditional data sources, since their interpretation requires both a representation of the results as well as a representation of how the results were produced. Furthermore, in order to believe the results, biologists need a way to explore the data easily. The best way to provide this is to give them interactive, dynamic software that presents genomic information within an easy-to-explore environment that capitalizes on users' innate ability to recognize potentially meaningful patterns embedded in the data.

Methods

Visualization applications

The visualization applications described here were developed in the Java programming language using the Java-based Neomorphic Genome Software Development Kit (NGSDK), a framework for developing genomic browser applications.

cDNA-to-genomic alignments

Alignments between cDNA and genomic sequences were produced using pslayout, a fast cDNA-to-genomic sequence alignment tool written by Jim Kent [22].

Authors' contributions

Ann Loraine contributed code and design work to the GeneViewer genome browser. Ann Loraine designed the ProtAnnot viewer, and Gregg Helt implemented it. Both authors read and approved the final manuscript.

Note

An earlier, shorter version of this article appears in the Proceedings of the 2002 IEEE Bioinformatics Conference.

Acknowledgements

Melissa Cline and Michael Siani-Rose provided valuable comments on the manuscript. Joseph Morris, Eric Blossom, Ed Erwin, Shaw Sun, Steve Chervitz, Adam Tracy, Cyrus Harmon, and Hari Tammana worked on the GeneViewer software and/or the NGSDK. TIGR, Neomorphic, and Affymetrix supported and guided development of the GeneViewer software.

References

1. Burge C, Karlin S: **Prediction of complete gene structures in human genomic DNA.** *J Mol Biol* 1997, **268:**78-94
2. Kulp D, Haussler D, Reese MG, Eeckman FH: **A generalized hidden Markov model for the recognition of human genes in DNA.** *Proc Int Conf Intell Syst Mol Biol* 1996, **4:**134-42
3. [http://www.ncbi.nlm.nih.gov]
4. Kent WJ, Sugnet CW, Furey TS, Roskin KM, Pringle TH, Zahler AM, Haussler D: **The human genome browser at UCSC.** *Genome Res* 2002, **12:**996-1006
5. Helt GA, Lewis S, Loraine AE, Rubin GM: **BioViews: Java-based tools for genomic data visualization.** *Genome Res* 1998, **8:**291-305
6. Lander ES, Linton LM, Birren B, Nusbaum C, Zody MC, Baldwin J, Devon K, Dewar K, Doyle M, FitzHugh W, *et al*: **Initial sequencing and analysis of the human genome.** *Nature* 2001, **409:**860-921
7. Bederson B, Hollan JD, Perlin K, Meyer J, Bacon D, Furnas GW: **Pad++: A zoomable graphical sketchpad for exploring alternate interface physics.** *J. of Visual Languages and Computing* 1996, **7:**3-31
8. Bederson B, Myer J, Good L: **Jazz: an extensible zoomable user interface graphics toolkit in Java.** *In: ACM UIST; San Diego, CA.* 2000, 171-181
9. [http://www.cs.umd.edu/hcil/jazz]

10. Mironov AA, Fickett JW, Gelfand MS: **Frequent alternative splicing of human genes.** *Genome Res* 1999, **9:**1288-93
11. Florea L, Hartzell G, Zhang Z, Rubin GM, Miller W: **A computer program for aligning a cDNA sequence with a genomic DNA sequence.** *Genome Res* 1998, **8:**967-74
12. Kent WJ: **BLAT-the BLAST-like alignment tool.** *Genome Res* 2002, **12:**656-64
13. Gray TA, Saitoh S, Nicholls RD: **An imprinted, mammalian bicistronic transcript encodes two independent proteins.** *Proc Natl Acad Sci U S A* 1999, **96:**5616-21
14. Iyer R, Jenkinson CP, Vockley JG, Kern RM, Grody WW, Cederbaum S: **The human arginases and arginase deficiency.** *J Inherit Metab Dis* 1998, **21:**86-100
15. [http://www.ncbi.nlm.nih.gov/LocusLink/LocRpt.cgi?l=5327]
16. Bateman A, Birney E, Durbin R, Eddy SR, Howe KL, Sonnhammer EL: **The Pfam protein families database.** *Nucleic Acids Res* 2000, **28:**263-6
17. [http://www.ensembl.org/apollo]
18. [http://omnigene.sourceforge.net]
19. [http://www.biojava.org]
20. [http://www.bioperl.org]
21. [http://www.wormbase.org]
22. Kent WJ, Haussler D: **Assembly of the working draft of the human genome with gigassembler.** *Genome Res* 2001, **11:**1541-8