SequenceJuxtaposer: Fluid Navigation For Large-Scale Sequence Comparison In Context

James Slack* Kristian Hildebrand*† Tamara Munzner* Katherine St. John‡

Abstract:

SequenceJuxtaposer is a sequence visualization tool for the exploration and comparison of biomolecular sequences. We use an information visualization technique called "accordion drawing" that guarantees three key properties: context, visibility, and frame rate. We provide context through the navigation metaphor of a rubber sheet that can be smoothly stretched to show more details in the areas of focus, while the surrounding regions of context are correspondingly shrunk. Landmarks, such as motifs or differences between aligned base pairs, are guaranteed to be visible even if located in the shrunken areas of context. Our graphics infrastructure for progressive rendering provides immediate responsiveness to user interaction by guaranteeing that we redraw the scene at a target frame rate. SequenceJuxtaposer supports interaction at 20 frames per second when browsing collections of several hundred sequences that comprise over 1.7 million total base pairs.

1 Introduction and Prior Work

Biomolecular sequence comparisons are essential in understanding underlying genomic patterns. Current sequence browsers [Hu02, Ke02, Wh] support examining the data in different views, where the interaction happens as discrete jumps. When only very small subsets of sequences are visible it is easy to lose track of the current location. The cognitive load of maintaining a mental model of navigation history is very high, and humans have a very limited capacity to do so [Zh91]. Exploration often entails backtracking to remember previous views. SequenceJuxtaposer allows users to interact with sequence data as if it is drawn on a rubber sheet with the borders tacked down [SSTR93]. Stretching certain areas causes the rest of the sheet to shrink accordingly, but landmarks remain visible in the periphery. We call this approach *accordion drawing* because the effect is similar to the stretching and shrinking of an accordion bellows. It is an example of a class of techniques known as "Focus+Context" in the information visualization literature [Fu86, LRP95], where overview and detail are drawn in a single view. We introduced accordion drawing with the TreeJuxtaposer system for visually comparing large phylogenetic trees [Mu03]. In this work, we present accordion drawing for biomolecular sequences, providing fluid

*Dept of Computer Science, U. of British Columbia, Vancouver, BC V6T 1Z4, Canada, {jslack,hilde,tmm}@cs.ubc.ca.

†Dept Media Systems, Bauhaus U., Weimar, Germany, kristian.hildebrand@medien.uni-weimar.de.

‡City University of New York, stjohn@lehman.cuny.edu.

exploration of large datasets with guarantees of three key properties: context, visibility and frame rate.

Comparing and analyzing sequences is a fundamental part of bioinformatics. Many text-based alignment tools were developed to address this problem, including [DGP96, MM92]. These tools only work well for aligning and analyzing a few small sequences. However, viewing the overall sequence structure becomes difficult when the length of a sequence exceeds the window size (often 80-100 nucleotides). Further advances in visualization of pairwise aligned sequences include [Ru00, Sc00, SBD03]. SequenceJuxtaposer complements recent web-based viewers that allow the search and display of genomic sequences integrated with annotation databases [Hu02, Ke02, Wh]. Other viewers, such as VISTA [Ma00] or phylo-VISTA [Sh03], are intended to run on a local client. Although these approaches have strengths, and many of them allow navigation of a great deal of information at different magnification levels, no approach allows users to compare multiple sequences with context preserved and fluid navigation between different magnification scales.

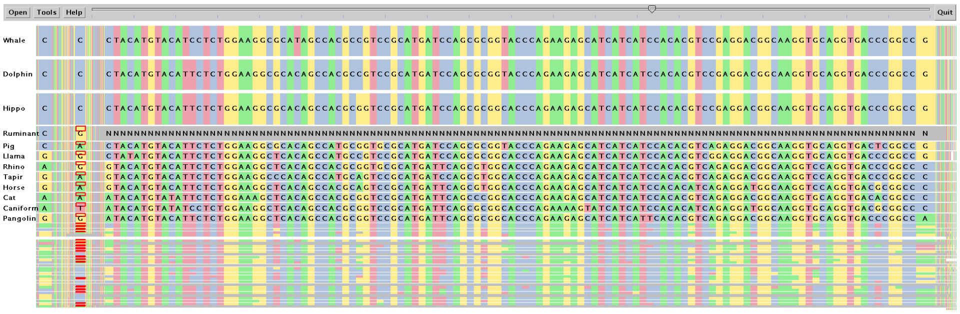
2 An Application

SequenceJuxtaposer allows fluid navigation of whole genomes, where the overall context of the genome is visible even when regions of interest are being examined in-depth. Our application currently handles DNA and RNA sequence data (or strings over the alphabet that include the nucleotide bases {A,C,G,T,U}, a symbol N for undetermined bases, and a symbol '-' for gaps).

The Murphy *et al.* [Mu01] dataset consists of molecular data for 22 genes and 44 mammals. The sequences are of length 16,397 bp and include 19 nuclear and 3 mitochondrial gene sequences for 42 placental and 2 marsupial outgroups. The placental mammals fall into four superordinal groups and the analysis of this dataset focused on resolving the interrelationships among these groups. Figure 1 shows all 44 sequences for the single gene CNR1, and when we find the appropriate difference threshold with the slider we can immediately see that whales, dolphins, and hippos are distinct from the other sequences. These three mammals form a clade. The ability to see the phylogenetic signal of clade membership by simply manipulating the difference slider shows SequenceJuxtaposer's power.

3 Algorithms

SequenceJuxtaposer uses advanced algorithms for drawing, searching and interaction. In the common case of redrawing frames our runtime algorithms are all sublinear. The user-initiated actions of changing difference thresholds or searching for motifs are linear. Our preprocessing algorithms are all subquadratic. Gaps that occur in all loaded sequences are automatically elided during preprocessing.



```
Whale     C  C  CTACATGTACATCCTCTGGAAGGCGCATAGCCACGCCGTCCGCATGATCCAGCGCGGTACCCAGAAGAGCATCATCATCCACACGTCCGAGGACGGCAAGGTGCAGGTGACCCGGCC G
Dolphin   C  C  CTACATGTACATTCTCTGGAAGGCGCACAGCCACGCCGTCCGCATGATCCAGCGCGGTACCCAGAAGAGCATCATCATCCACACGTCCGAGGACGGCAAGGTGCAGGTGACCCGGCC G
Hippo     C  C  CTACATGTACATTCTCTGGAAGGCGCACAGCCACGCGGTCCGCATGATCCAGCGCGGCACCCAGAAGAGCATCATCATCCACACGTCCGAGGACGGCAAGGTGCAGGTGACCCGGCC G
Ruminant  C  G  NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN N
Pig       C  A  CTACATGTACATTCTCTGGAAGGCGCACAGCCATGCGGTGCGCATGATCCAGCGCGGTACCCAGAAGAGCATCATCATCCACACGTCAGAGGACGGCAAGGTGCAGGTGACTCGGCC G
Llama     G  G  CTATATGTACATTCTCTGGAAGGCTCACAGCCATGCCGTCCGCATGATCCAGCGCGGCACCCAGAAGAGCATCATCATCCACACGTCGGAGGACGGCAAGGTGCAGGTGACCCGGCC G
Rhino     A  G  GTACATGTACATTCTCTGGAAGGCTCACAGCCACGCGGTGCGCATGATTCAGCGTGGCACCCAGAAGAGCATCATCATCCACACGTCAGAGGACGGCAAGGTCCAGGTGACCCGGCC C
Tapir     G  A  GTACATGTACATTCTCTGGAAGGCCCACAGCCATGCAGTCCGCATGATCCAGCGTGGCACCCAGAAGAGCATCATCATCCACACGTCAGAGGACGGCAAGGTCCAGGTGACCCGGCC C
Horse     G  A  GTACATGTACATTCTCTGGAAGGCTCACAGCCACGCAGTCCGCATGATTCAGCGTGGCACCCAGAAGAGCATCATCATCCACACATCAGAGGATGGCAAGGTCCAGGTGACGCGGCC C
Cat       A  A  ATACATGTATATTCTCTGGAAAGCTCACAGCCACGCGGTCCGCATGATTCAGCCCGGCACCCAGAAGAGCATCATCATCCACACGTCAGAGGACGGCAAGGTGCAGGTGACACGGCC C
Caniform  A  T  ATACATGTATATCCTCTGGAAGGCTCACAGCCACGCGGTCCGCATGATTCAGCGCGGCACCCAGAAAAGTATCATCATCCACACATCAGAGGATGGCAAGGTGCAGGTGACGCGGCC C
Pangolin  G  G  ATACATGTACATTCTCTGGAAGGCTCACAGCCACGCGGTCCGCATGATTCAGCGCGGCACCCAGAAGAGCATCATCATTCACACGTCAGAGGACGGCAAGGTGCAGGTGACCCGGCC A
```

Figure 1: We show the 1001 bp CNR1 gene, a subset of the Murphy *et al.* 44-sequence dataset [Mu01], after exploration revealed a meaningful threshold of differences in nucleotide site variability of 67%.

Differences: Computing the differences between aligned nucleotides in sequences has a processing cost of $O(nk)$, where n is the number of nucleotide positions and k is the number of sequences. We make a first pass through the sequences at a nucleotide position to find the "majority" nucleotide, that occurs most often. If the majority is less the threshold, which is interactively controllable by the user through a slider, then we make a second pass through that column to mark the nucleotides that do not match this majority. Gaps and undetermined base pairs are ignored.

Interaction: Rectangular areas can be expanded or contracted by selecting either an area of the window to freely resize with the mouse, or by picking a group of nucleotides to grow or shrink. Smooth transitions allow people to easily track visual landmarks, which is faster than reacquiring visual targets after an abrupt jump cut [RCM89]. Although the absolute position of an item changes when we resize regions, the relative ordering between items remains. With interactive resizing, the user drags the mouse in the window to define an active area and then resizes that area by moving the visible rubberband from one of its corners. Users can also trigger an animated transition to grow or shrink all nucleotides belonging to a particular group with a button press. The three main groups are the nucleotides marked as text search results, as alignment gaps, and as different according to the current threshold value. The look and feel of the fluid interaction is difficult to communicate through words and still images, so we also provide an accompanying video showing the system in action[*].

Drawing: Our drawing algorithms are designed for scalable performance even when the total number of nucleotides t in the dataset is much larger than the number of visible nucleotides v that can be seen in any single view. Using algorithms that are $O(t)$ would lead to unacceptably slow interactive performance, whereas $O(v)$ or $O(\log t)$ yields acceptable runtime results. The number of visible nucleotides depends on the number of pixels available for display. One way to ensure $O(v)$ rendering is to cull the geometric elements

[*] http://www.cs.ubc.ca/~tmm/papers/sj

in highly compressed areas when they are smaller than a single pixel. Our guarantee of landmark visibility requires checking for any nucleotides corresponding to a highly compressed onscreen area that should be marked. At preprocessing time we build a quadtree in $O(t \log t)$ time that maps the colored boxes, the geometric representation of a nucleotide, into a hierarchical decomposition of space. Before culling, we check quadtree cells against each of the two marked groups (differences and search results). The groups are kept as r contiguous ranges, where r can be at most $t/2$. We store the r ranges in a sorted tree for an $O(\log r)$ lookup, so the total cost of the guaranteed visibility check is $O(v \log r)$.
When resizing regions, we draw for a fixed amount of time and check for user interaction before continuing, which is called *progressive rendering* [BFGS86]. We use a priority queue to draw items in order of current onscreen size so the magnified areas are drawn before compressed ones. Our approach provides very high information density as well as extremely lightweight and fluid navigation.

Performance: Web-based sequence browsers impose minimal requirements on the local client machine; major computation is done with a large remote server farm [Hu02, Ke02, Wh]. SequenceJuxtaposer requires significant memory resources to load very large datasets to support data structures such as the quadtree-based accordion drawers.
Our prototype is written in Java using the GL4Java bindings to the OpenGL graphics library. The benchmarks were run on a 3.0GHz Pentium 4 with 2GB of main memory.
Our system's linear memory footprint allows it to be used for interactive exploration of both huge datasets (1.7 Mbp) on high-end desktop machines, and medium-sized datasets on low-end laptops. Preprocessing the entire Murphy *et al.* [Mu01] dataset of 721 Kbp takes 25 seconds of wall clock time. The time to draw the entire scene ranged from 5 seconds with no marked regions to 7 seconds for the worst case.

4 Conclusion

We plan to join TreeJuxtaposer with SequenceJuxtaposer for automatic linked navigation between phylogenetic trees and sequences. We propose using trees as a navigational tool, as a way to locate possible regulatory elements in aligned whole genomes and to classify higher level differences between very large datasets. Annotations will add a hierarchical series of landmarks that will provide layers of abstraction, allowing navigation at multiple levels of detail, from base pairs to entire genomes.
SequenceJuxtaposer incorporates the powerful information visualization technique of accordion drawing where details are always shown within a global context, landmarks are guaranteed to be visible, and the frame rate for redrawing the scene is guaranteed to provide realtime response. We can load hundreds of sequences with more than 1.7 Mbp with the ability to fluidly resize areas of interest. We allow exploration of differences between the sequences by interactively changing the threshold for marking differences at each nucleotide site. We also support searching for motifs with immediate visual feedback.
SequenceJuxtaposer is open source and available at `http://olduvai.sourceforge.net/sj`.

5 Acknowledgements

We thank Wayne Maddison for originally suggesting we attack this problem, David Hillis, David Haussler, Nina Amenta for helpful conversations, and Ciarán Llachlan Leavitt for comments on paper drafts. We also thank the Hillis lab at UT-Austin for several test datasets. Funding was provided by NSF/DEB-0121651/0121682, NSERC/RGPIN 262047-03, and Hildebrand was supported by the German Academic Exchange Service.

References

[BFGS86] Bergman, L., Fuchs, H., Grant, E., und Spach, S.: Image rendering by adaptive refinement. In: *SIGGRAPH*. S. 29–37. 1986.

[DGP96] Duret, L., Gasteiger, E., und Perrire, G.: LalnView: A graphical viewer for pairwise sequence alignments. *Comput. Applic. Biosci.* 12:507–51. 1996.

[Fu86] Furnas, G. W.: Generalized fisheye views. In: *Proc. SIGCHI*. S. 18–23. 1986.

[Hu02] Hubbard et al., T.: The Ensembl genome database project. *Nucleic Acids Research*. 30(1):38–41. 2002. `www.ensembl.org`.

[Ke02] Kent et al., W.: The human genome browser at UCSC. *Genome Res.* 12:996–1006. 2002. `genome.ucsc.edu`.

[LRP95] Lamping, J., Rao, R., und Pirolli, P.: A Focus+Content technique based on hyperbolic geometry for viewing large hierarchies. In: *Proc. SIGCHI*. S. 401–408. 1995.

[Ma00] Mayor et al., C.: VISTA: Visualizing global DNA sequence alignments of arbitrary length. *Bioinformatics*. 16:1046–1047. 2000. Application Note.

[MM92] Maddison, W. P. und Maddison, D. R.: *MacClade: Analysis of Phylogeny and Character Evolution. (User's manual)*. Sinauer Associates, Sunderland, MA. 1992.

[Mu01] Murphy et al., W. J.: Resolution of the early placental mammal radiation using Bayesian phylogenetics. *Science*. 294(5550):2348–51. 2001.

[Mu03] Munzner et al., T.: TreeJuxtaposer: Scalable tree comparison using Focus+Context with guaranteed visibility. *SIGGRAPH*. S. 453–462. 2003.

[RCM89] Robertson, G., Card, S., und Mackinlay, J.: The cognitive coprocessor architecture for interactive user interfaces. In: *Proc. UIST*. S. 10–18. 1989.

[Ru00] Rutherford et al., K.: Artemis: Sequence visualization and annotation. *Bioinformatics*. 16(10):944–5. 2000. Application Note.

[SBD03] Spell, R., Brady, R., und Dietrich, F.: BARD: A visualization tool for biological sequence analysis. In: *Proc. IEEE Symposium on Information Visualization*. S. 219–226. 2003.

[Sc00] Schwartz et al., S.: PipMaker: A web server for aligning two genomic DNA sequences. *Genome Research*. 10(4). 2000.

[Sh03] Shah et al., N.: Phylo-VISTA: an interactive visualization tool for multiple dna sequence alignments. *Bioinformatics*. 19. 2003. To appear, Application Note.

[SSTR93] Sarkar, M., Snibbe, S. S., Tversky, O. J., und Reiss, S. P.: Stretching the rubber sheet: A metaphor for viewing large layouts on small screens. In: *Proc. UIST*. S. 81–91. 1993.

[Wh] Wheeler et al., D. Database resources of the National Center for Biotechnology Information: 2002 update.

[Zh91] Zhang, J.: The interaction of internal and external representations in a problem solving task. In: *Proc. 13th Annual Conf. of Cog. Sci. Society*. Lawrence Erlbaum Assoc. 1991.