Bioinformatics, 35(8), 2019, 1425–1426
doi: 10.1093/bioinformatics/bty807
Advance Access Publication Date: 21 September 2018
Applications Note

Sequence analysis

Squiggle: a user-friendly two-dimensional DNA sequence visualization tool

Benjamin D. Lee [1,2]

[1]Department of Computer Science, School of Engineering and Applied Sciences, Harvard University, Cambridge, MA 02138, USA and [2]Lab41, In-Q-Tel, Menlo Park, CA 94025, USA

Associate Editor: John Hancock

Received on July 25, 2018; revised on September 11, 2018; editorial decision on September 12, 2018; accepted on September 19, 2018

Abstract

Summary: Squiggle is a software tool that automatically generates interactive web-based two-dimensional graphical representations of raw DNA sequences. Built with ease of use in mind, Squiggle implements several prior sequence visualization algorithms and introduces novel visualization methods designed to maximize human usability.

Availability and implementation: Squiggle is written in Python 3 and freely available under the permissive MIT license. The open-source code can be downloaded from github.com/Lab41/squiggle as well as the Python Package Index. The installation instructions and user guide are available at squiggle.readthedocs.org.

Contact: benjamindlee@me.com

1 Introduction

With the rise of next-generation sequencing technologies, DNA sequence analysis has become an increasingly common tool both in bioinformatics and biology at large. For this reason, the ability to quickly inspect unannotated DNA sequences is crucial. However, raw sequence data, such as are contained within FASTA files, are not well suited to human exploration because they are often long sequences of visually similar characters. For example, when viewing multiple highly conserved coding sequences, there may be relatively few differing characters, which could be difficult to spot with the naked eye. To better display sequence data, there are a multitude of visualization algorithms that can turn the long sequences of As, Ts, Gs and Cs which comprise DNA into two-dimensional graphical representations (Gates, 1986; Qi and Qi, 2007; Randić *et al.*, 2003; Yau *et al.*, 2003).

Despite the numerous raw DNA sequence visualization methods described in the literature, a review of the literature did not identify any open-source software implementations of these methods. Therefore, I propose Squiggle, an open-source command line tool and Python package that turns nucleotide sequences into interactive browser-based two-dimensional visualizations.

2 Implementation

Squiggle is implemented in Python 3 and supports all versions of Python above 3.4, as well as PyPy. It is built around a single function, `transform()`, which takes DNA sequences and converts them into arrays of x and y coordinates using various visualization methods. At a high level, the command line tool parses sequences from provided FASTA files containing DNA sequences, calls the `transform()` function on each sequence, and then plots the transformed sequences. These visualizations may then be opened and explored using any modern web browser with or without an internet connection.

Squiggle automatically creates publication-ready visualizations which can be exported as images with labeled axes and legends, as well as configurable color palettes, dimensions, and layouts. The command line interface was designed with ease-of-use in mind. As such, the `squiggle` command requires only a single argument: a FASTA file containing a DNA sequence. Figure 1 shows an example Squiggle command and output. Multiple DNA sequences may be provided in the same FASTA file or as subsequent arguments.

2.1 Visualization methods

Squiggle implements several prior visualization methods (Gates, 1986; Qi and Qi, 2007; Randić *et al.*, 2003; Yau *et al.*, 2003) as well as introduces two new methods. The first, based on the method proposed by Yau et al. (2003), represents DNA sequences as a series of vectors in either the upward or downward direction (representing pyrimidine and purine bases, respectively) connected tip-to-tail. This method differs from the one introduced by Yau *et al.* in that the

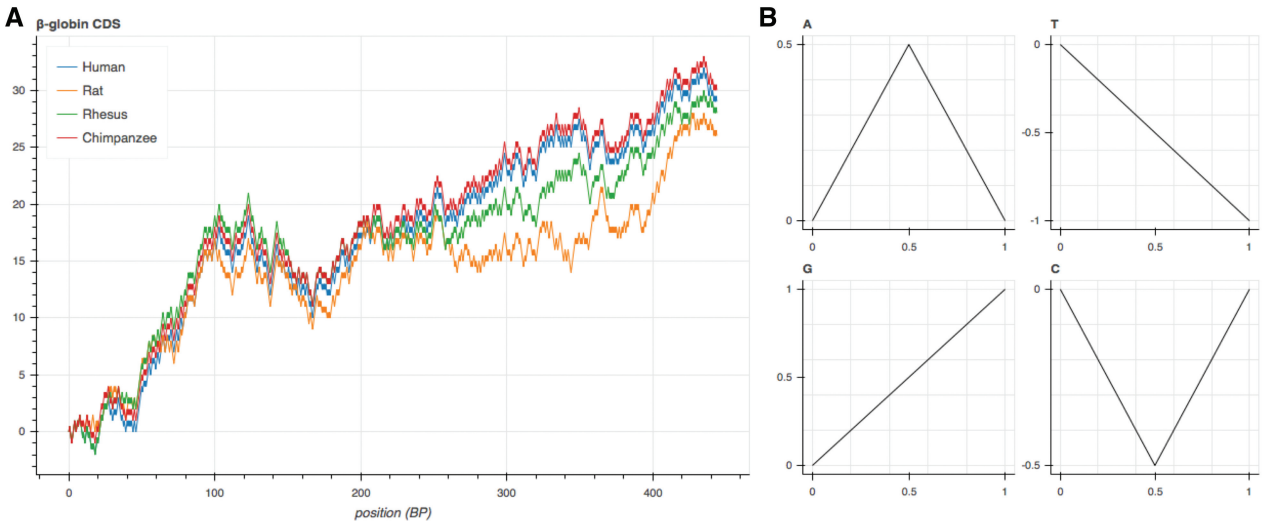

Fig. 1. (**A**) A Squiggle visualization of the β-globin CDSs for various species. This visualization was generated with the command `squiggle hbb.fasta -title"β-globin CDS"`. (**B**) The unique shape for each base using the Squiggle visualization method

vectors are not unit vectors, but rather are such that their projection onto the x axis has a length of 1. This modification has the significant advantage of allowing the x coordinate of a point in the visualization to correspond directly to its index in the sequence, thereby dramatically simplifying analysis.

The second novel method is based on the UCSC `.2bit` and the Qi *et al.* (2011) Huffman coding method. In essence, a DNA sequence is first converted into binary using the `.2bit` encoding scheme that maps $\texttt{T} \rightarrow \texttt{00}, \texttt{C} \rightarrow \texttt{01}, \texttt{A} \rightarrow \texttt{10}, \texttt{G} \rightarrow \texttt{11}$. For example, the DNA sequence `ATGC` becomes `10001101`. Then, starting at the origin, for each `1` and `0`, the vectors $\left(\frac{1}{2}, \frac{1}{2}\right)$ and $\left(\frac{1}{2}, -\frac{1}{2}\right)$, respectively, are connected tip-to-tail. This encoding results in each nucleotide having a distinctive shape, shown in Figure 1.

This visualization method has several desirable properties, some of which may not immediately be apparent. First, it is based on an open, common bioinformatics format. Second, there is no degeneracy in the encoding. Third, whether the overall GC-content is above or below 50% may be inferred from a glance based on whether the endpoint of the graph is above or below $y = 0$, respectively. Fourth, regions inside the gene with varying GC-content can be seen as peaks and valleys. Fifth, the visualization is limited to quadrants I and IV and is a function, unlike in Gates (1986). Sixth, as in the modified method described above, the x-axis corresponds directly with nucleotide position. Finally, this encoding method supports ambiguous nucleotides, which are defined as horizontal vectors of length 1.

3 Results

As a demonstration of the ease of use of Squiggle, the coding data sequences (CDSs) of the β-globin genes of various species were plotted using the novel Squiggle visualization method. Figure 1 shows the results of this visualization. Several features of the sequences can immediately be inferred from Figure 1. The relative GC-content of each CDS can be seen, with the chimpanzee CDS exhibiting greater GC-content than the human CDS (as evidenced by its greater final y coordinate), which in turn appears to exhibit greater GC-content than the rhesus macaque's and Norway rat's CDSs. Further, a phylogenetic relationship can be seen based on the overall sequence homology. Finally, the location within the sequence where the divergence is greatest is immediately apparent.

4 Conclusion

Squiggle is a useful command line tool for quickly and intuitively surveying DNA sequences as interactive two-dimensional graphs with minimal effort. Built using modern web technologies, Squiggle provides numerous visualization methods to capture various aspects of raw DNA sequences contained within FASTA files. This software is freely available on GitHub and the Python Package Index and is archived at Zenodo.

Acknowledgements

The author would like to thank Paul Gamble for his feedback on the usability of this software as well as the initial manuscript.

Conflict of Interest: none declared.

References

Gates,M.A. (1986) A simple way to look at DNA. *J. Theor. Biol.*, **119**, 319–328.

Qi,Z., and Qi,X. (2007) Novel 2D graphical representation of DNA sequence based on dual nucleotides. *Chem. Phys. Lett.*, **440**, 139–144.

Qi,Z.-H. *et al.* (2011) Using huffman coding method to visualize and analyze DNA sequences. *J. Computat. Chem.*, **32**, 3233–3240.

Randić,M. *et al.* (2003) Novel 2-D graphical representation of DNA sequences and their numerical characterization. *Chem. Phys. Lett.*, **368**, 1–6.

Yau,S. *et al.* (2003) DNA sequence representation without degeneracy. *Nucleic Acids Res.*, **31**, 3078–3080.