
LOGOMAKER: BEAUTIFUL SEQUENCE LOGOS IN PYTHON

A bioRxiv preprint

Ammar Tareen

Simons Center for Quantitative Biology
Cold Spring Harbor Laboratory
tareen@cshl.edu

Justin B. Kinney

Simons Center for Quantitative Biology
Cold Spring Harbor Laboratory
jkinney@cshl.edu

May 10, 2019

ABSTRACT

Sequence logos are visually compelling ways of illustrating the biological properties of DNA, RNA, and protein sequences, yet it is currently difficult to generate such logos within the Python programming environment. Here we introduce Logomaker, a Python API for creating publication-quality sequence logos. Logomaker can produce both standard and highly customized logos from any matrix-like array of numbers. Logos are rendered as vector graphics that are easy to stylize using standard matplotlib functions. Methods for creating logos from multiple-sequence alignments are also included. **Availability and Implementation:** Logomaker can be installed using the pip package manager and is compatible with both Python 2.7 and Python 3.6. Source code is available at <http://github.com/jbkinney/logomaker>. **Supplemental Information:** Documentation is provided at <http://logomaker.readthedocs.io>. **Contact:** jkinney@cshl.edu.

Introduction

Sequence logos provide evocative graphical representations of the functional properties of DNA, RNA, and protein sequences. Logos consist of characters stacked upon one another at a series of integer-valued positions, with the height of each character conveying some type of information about its biological importance. This graphical representation was introduced by Schneider and Stephens (1990) for illustrating statistical properties of multiple-sequence alignments. Although the specific representation they advocated is still widely used, sequence logos have since evolved into a general data visualization strategy that can be used to illustrate many different kinds of biological information (Kinney and McCandlish, 2019). For example, logos can be used to illustrate base-pair-specific contributions to protein-DNA binding energy (Foat *et al.*, 2006), the effects of mutations in massively parallel selection experiments (Liachko *et al.*, 2013), and the saliency footprints of deep neural networks (Shrikumar *et al.*, 2017; Jaganathan *et al.*, 2019).

Making customized logos, however, remains surprisingly difficult. A substantial number of software tools for generating sequence logos have been described, but almost all of these tools limit the kinds of logos one can make and the ways in which those logos can be styled. An important exception is ggseqlogo (Wagih, 2017), which enables the creation of sequence logos within the R programming environment from arbitrary user-provided data. Importantly, ggseqlogo renders logos using native vector graphics, which facilitates post-hoc styling and the incorporation of logos into publication-ready figures. However, similar software is not yet available for Python. Because many biological data analysis pipelines are written in Python, there is a clear need for such logo-generating capabilities.

Implementation

Logomaker is a flexible Python API for creating publication-quality sequence logos. Logomaker takes as input a matrix (specifically, a pandas data frame) in which columns represent characters, rows represent positions, and values represent character heights (Figure 1A). This enables the creation of logos representing any type of data. The resulting logo is drawn using vector graphics embedded within a standard matplotlib Axes object, thus facilitating a high level of customization as well as incorporation into complex figures that contain other graphical elements. Logomaker provides a variety of options for styling the characters within a logo, including the choice of font, color scheme, vertical and horizontal padding, etc.. Logomaker also

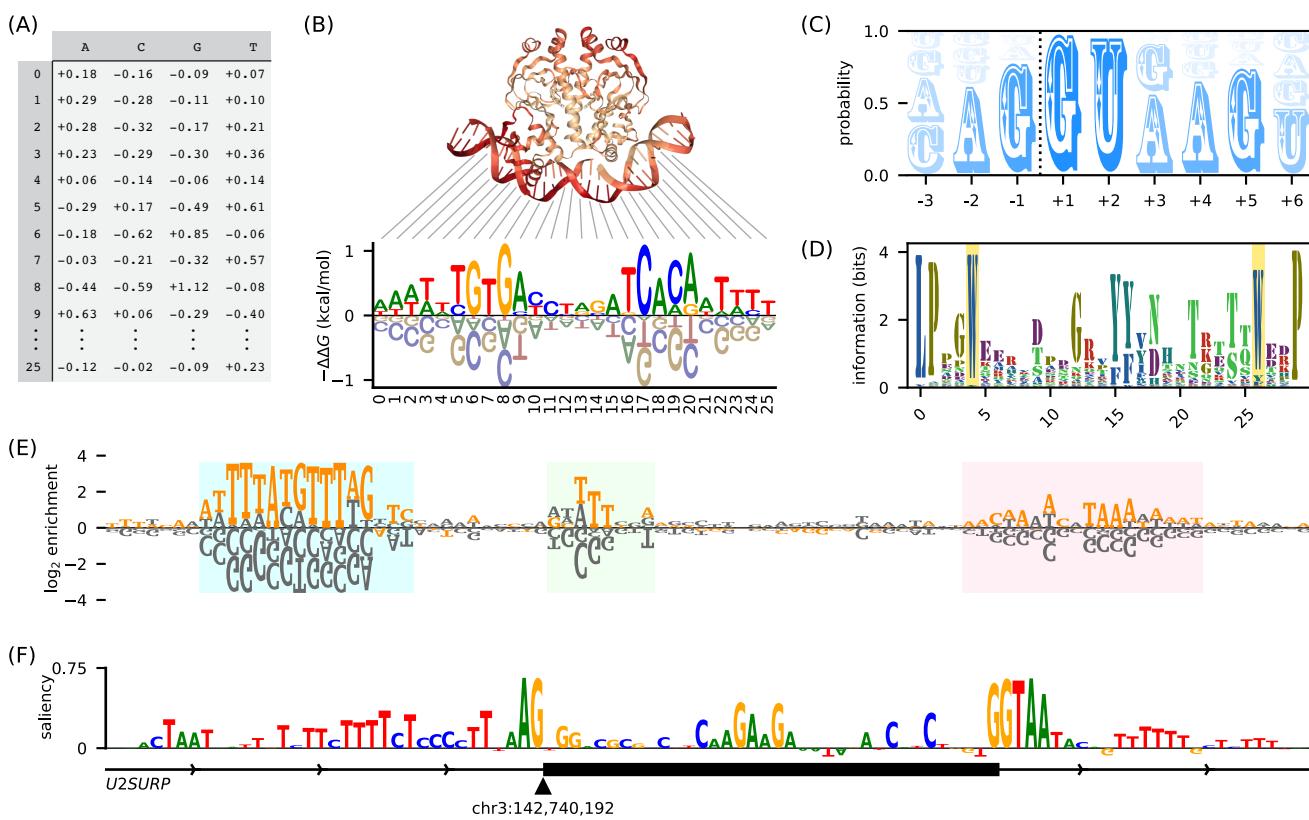


Figure 1: Logomaker logos can represent diverse types of data. (A) Example input to Logomaker. Shown is an energy matrix for the transcription factor CRP; elements of this data frame represent $-\Delta\Delta G$ values contributed by each possible base (columns) at each nucleotide position (rows). Data are from Kinney *et al.* (2010). (B) An energy logo for CRP created by passing the data frame in panel A to Logomaker. The structural context of each nucleotide position is indicated [PDB 1CGP Parkinson *et al.* (1996)]. (C) A probability logo computed from all annotated 5' splices sites in the human genome (Frankish *et al.*, 2019). The dashed line indicates the intron/exon boundary. (D) An information logo computed from a multiple alignment of WW domain sequences [PFAM RP15 (Finn *et al.*, 2014)], with the eponymous positions of this domain highlighted. (E) An enrichment logo representing the effects of mutations within the ARS1 replication origin of *S. cerevisiae*. Orange characters indicate the ARS1 wild-type sequence; highlighted regions correspond (from left to right) to the A, B1, and B2 elements of this sequence (Rao and Stillman, 1995). Data (unpublished; collected by JBK) are from a mutARS-seq experiment analogous to the one reported by Liachko *et al.* (2013). (F) A saliency logo (Shrikumar *et al.*, 2017) representing the importance of nucleotides in the vicinity of U2SURP exon 9, as predicted by a deep neural network model of splice site selection. Logo adapted (with permission) from Figure 1D of Jaganathan *et al.* (2019).

enables the highlighting of specific sequences within a logo (Figure 1E), as well as the use of value-specific transparency in logos that illustrate probabilities (Figure 1C). If desired, users can further customize individual characters within any rendered logo.

Because sequence logos are still commonly used to represent the statistics of multiple sequence alignments, Logomaker provides methods for processing such alignments into matrices that can then be used to generate logos. Multiple types of matrices can be generated in this way, including matrices that represent probabilities (Figure 1C), log odds ratios (Figure 1E), or the information values described by Schneider and Stephens (1990) (Figure 1D). Methods for transforming between these types of matrices are also provided. Finally, Logomaker also supports the creation of matrices (and thus logos) that represent neural network saliency maps (Shrikumar *et al.*, 2017), as in Figure 1F.

Conclusion

Logomaker thus fills a major need in the Python community for flexible logo-generating software. Indeed, preliminary versions of Logomaker have already been used to generate logos for multiple publications (Belliveau *et al.*, 2018; Wong *et al.*, 2018; Forcier

et al., 2018; Nguyen *et al.*, 2018; Barnes *et al.*, 2019; Kinney and McCandlish, 2019). Logomaker is thoroughly tested, has minimal dependencies, and can be installed from PyPI by executing `pip install logomaker` at the command line. A step-by-step tutorial on how to use Logomaker, as well as comprehensive documentation, are available at <http://logomaker.readthedocs.io>.

Funding

This work was supported by a CSHL/Northwell Health Alliance grant to JBK and by a National Institutes of Health Cancer Center Support Grant [5P30CA045508].

Acknowledgements

We thank Polly Fordyce, William Ireland, David McCandlish, and Bruce Stillman for helpful discussions. We also thank Kyle Farh and Kishore Jaganathan for providing data for the saliency logo in Figure 1F.

References

Barnes, S. L. *et al.* (2019). Mapping DNA sequence to transcription factor binding energy in vivo. *PLoS Comput Biol*, **15**(2), e1006226.

Belliveau, N. M. *et al.* (2018). Systematic approach for dissecting the molecular mechanisms of transcriptional regulation in bacteria. *Proc Natl Acad Sci USA*, **115**(21), E4796–E4805.

Finn, R. D. *et al.* (2014). Pfam: the protein families database. *Nucl Acids Res*, **42**(Database issue), D222–30.

Foat, B. *et al.* (2006). Statistical mechanical modeling of genome-wide transcription factor occupancy data by MatrixREDUCE. *Bioinformatics*, **22**(14), e141–9.

Forcier, T. L. *et al.* (2018). Measuring cis-regulatory energetics in living cells using allelic manifolds. *eLife*, **7**, 1129.

Frankish, A. *et al.* (2019). GENCODE reference annotation for the human and mouse genomes. *Nucl Acids Res*, **47**(D1), D766–D773.

Jaganathan, K. *et al.* (2019). Predicting Splicing from Primary Sequence with Deep Learning. *Cell*, **176**(3), 535–548.e24.

Kinney, J. B. and McCandlish, D. M. (2019). Massively parallel assays and quantitative sequence-function relationships. *Annu. Rev. Genom. Hum. G.*, **In Press**.

Kinney, J. B. *et al.* (2010). Using deep sequencing to characterize the biophysical mechanism of a transcriptional regulatory sequence. *Proc Natl Acad Sci USA*, **107**(20), 9158–9163.

Liachko, I. *et al.* (2013). High-resolution mapping, characterization, and optimization of autonomously replicating sequences in yeast. *Genome Res*, **23**(4), 698–704.

Nguyen, H. Q. *et al.* (2018). Quantitative mapping of protein-peptide affinity landscapes using spectrally encoded beads. *bioRxiv*, doi: [10.1101/306779](https://doi.org/10.1101/306779).

Parkinson, G. *et al.* (1996). Structure of the CAP-DNA complex at 2.5 angstroms resolution: a complete picture of the protein-DNA interface. *J Mol Biol*, **260**(3), 395–408.

Rao, H. and Stillman, B. (1995). The origin recognition complex interacts with a bipartite DNA binding site within yeast replicators. *Proc Natl Acad Sci USA*, **92**(6), 2224–2228.

Schneider, T. D. and Stephens, R. M. (1990). Sequence logos: a new way to display consensus sequences. *Nucl Acids Res*, **18**(20), 6097–6100.

Shrikumar, A. *et al.* (2017). Learning Important Features Through Propagating Activation Differences. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 3145–3153.

Wagih, O. (2017). ggseqlogo: a versatile R package for drawing sequence logos. *Bioinformatics*, **33**(22), 3645–3647.

Wong, M. S. *et al.* (2018). Quantitative Activity Profile and Context Dependence of All Human 5' Splice Sites. *Mol Cell*, **71**(6), 1012–1026.e3.