Bioinformatics, 33(22), 2017, 3645–3647
doi: 10.1093/bioinformatics/btx469
Advance Access Publication Date: 20 July 2017
Applications Note

OXFORD

Sequence analysis

ggseqlogo: a versatile R package for drawing sequence logos

Omar Wagih*

European Molecular Biology Laboratory, European Bioinformatics Institute, Wellcome Trust Genome Campus, Hinxton, Cambridge CB10 1SD, UK

*To whom correspondence should be addressed.

Associate Editor: John Hancock

Received on June 1, 2017; revised on June 30, 2017; editorial decision on July 15, 2017; accepted on July 18, 2017

Abstract

Summary: Sequence logos have become a crucial visualization method for studying underlying sequence patterns in the genome. Despite this, there remains a scarcity of software packages that provide the versatility often required for such visualizations. ggseqlogo is an R package built on the ggplot2 package that aims to address this issue. ggseqlogo offers native illustration of publication-ready DNA, RNA and protein sequence logos in a highly customizable fashion with features including multi-logo plots, qualitative and quantitative colour schemes, annotation of logos and integration with other plots. The package is intuitive to use and seamlessly integrates into R analysis pipelines.

Availability and implementation: ggseqlogo is released under the GNU licence and is freely available via CRAN-The Comprehensive R Archive Network https://cran.r-project.org/web/packages/ggseqlogo. A detailed tutorial can be found at https://omarwagih.github.io/ggseqlogo.

Contact: wagih@ebi.ac.uk

1 Introduction

Sequence logos are widely used to gain insight into underlying sequence patterns and are a crucial visualization in a range of bioinformatic analyses. For instance, they are used to highlight conserved positions in sequence alignments and study structural domain sequence similarity. Sequence logos also play a major role in visualizing DNA, RNA and protein binding sites, such as that of kinases, SH2/SH3 domains, transcription factors (TFs), RNA-binding proteins, nucleases, the nuclear ribonucleo proteins and countless more. Such binding sites are often clinically relevant and thus sequence logos further play a role in accentuating the impact of disease mutations. For example, they are widely used to explore disease variants that disrupt kinase-substrate binding sites and TF binding sites (Deplancke *et al.*, 2016; Wagih *et al.*, 2015). Given the invaluable nature of sequence logos, and the limited availability of logo-drawing libraries in R, it is important that appropriate visualization tools are made available.

An ideal sequence logo visualization package should be able to generate publication-ready logos of great quality, flexibly handle different input formats, be highly customizable (i.e. allow for quantitative and qualitative colour schemes, legends, different fonts and visual annotations), be integrable with other plots and pipelines, and allow for multiple logos to be visualized simultaneously. In R, there remains a lack of packages that exhaustively provide such functionality. For instance, `seqLogo` (Bembom, 2016) is only able to generate DNA-based logos and is limited in customizability and input formats. `motifStack` (Ou and Zhu, 2017) allows for drawing native DNA, RNA and protein sequence logos. However, it only allows matrices as input, visual attributes of the logo cannot be modified and logos cannot be combined with other plots. Finally, the CRAN package `RWebLogo` serves as a wrapper to the commonly used WebLogo Python library (Crooks *et al.*, 2004), which limits its customizability and does not allowing for sequence logos to be drawn natively in R, rather only saved to disk.

With these challenges in mind, I developed `ggseqlogo`, a package that extends `ggplot2` one of the most versatile and commonly used plotting packages in R (Wickham, 2009). This allows for almost any visual aspect of the logo to be modified, annotations to be overlaid, and seamless integration with other plots. The package additionally delivers numerous features including multi-logo

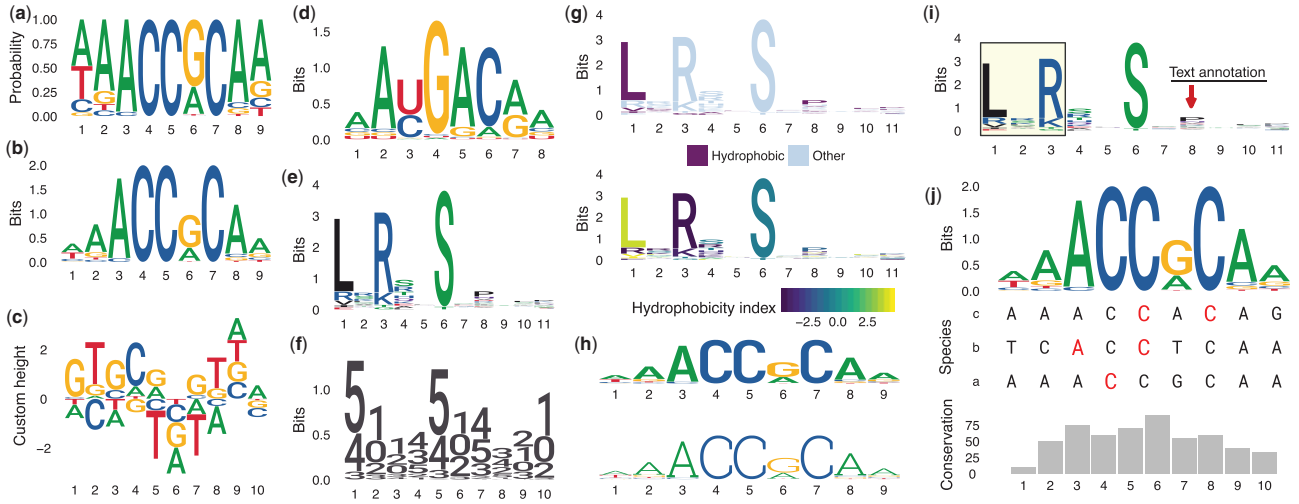

Fig. 1. Examples of sequence logos generated by `ggseqlogo`. These include binding sites of the transcription factor RUNX2 (DNA), RNA-binding protein FXR1 (RNA) and protein kinase PRKD1 (amino acids), as well as randomly generated sequences. (**a, b**) Examples of DNA sequence logos showing the two methods used to calculate letter height (a) relative frequency and (b) information content applied to DNA sequences. (**c**) An example of custom letter heights plotted using randomly generated data. (**d–f**) Examples of other supported sequence types including (**d**) RNA and (e) amino acids. (**f**) An example of a custom sequence types using numeric characters. (**g**) Examples of discrete (top) and continuous (bottom) colour schemes applied to highlight hydrophobic residues in an amino acid logo. (**h**) A small sample of the alternative fonts available in `ggseqlogo`. (**i**) A sample of annotations added to a sequence logo. (**j**) An example of a sequence logo combined with two other plots: a sequence alignment across different species and a bar plot showing positional-conservation of nucleotides

and batch plotting, qualitative and quantitative colour schemes, custom-height logos, custom-alphabet logos and different fonts. `ggseqlogo` is an easy-to-use package providing effortless generation of publication-quality sequence logos, allowing for integration into R analysis pipelines.

2 Description

The core visualization used in `ggseqlogo` relies on the use of polygons to draw elongated letters in `ggplot2`. Letters from a font are converted to high-resolution images and traced to provide coordinates of letters in 2 D space. To compute letter heights, `ggseqlogo` employs one of two commonly used methods. In a given position, the height of each letter is scaled proportionally to either (i) its relative frequency in the position or (ii) the amount of information contributed (measured in bits) using Shannon entropy (Schneider and Stephens, 1990) (Fig. 1a,b). In cases where neither height method is desired, `ggseqlogo` allows custom heights for each letter to be provided as a matrix for drawing of the logo (Fig. 1c). This is accommodating for pipelines that employ alternative statistical methods to compute letter heights (e.g. *z*-scores or log *P*-values), which then require visualization (Jessen *et al.*, 2013).

As input, `ggseqlogo` requires a sequence alignment as a character vector or position frequency matrix, depicting the count of each letter per position. The input data sequence type is flexible and can be one of DNA, RNA or protein (Fig. 1d–e), or a custom dictionary can optionally be defined to allow for non-standard logos. (Fig. 1f). Furthermore, providing a single matrix or alignment, `ggseqlogo` accepts a named list of sequence alignments or matrices, in which case logos are visualized in a grid using `ggplot2`'s faceting functions.

`ggseqlogo` provides numerous features that allow for tweaking of the sequence logo. For instance, users can select from a wide range of predefined discrete or continuous colour schemes. Continuous colour schemes allow for a gradient of colours (e.g. hydrophobicity index of an amino acid) (Fig. 1e). Custom colour schemes can also be defined (Fig. 1f). Other aspects such as the font used to draw the sequence logos are also easily interchangeable, with numerous serif and sans-serif fonts of varying weights included within the package.

In combination with `ggplot2`, annotation of sequence logos is straightforward, allowing elements such as rectangles, lines, arrows and text to be precisely overlaid onto the logo, programatically (Fig. 1i). Generated logos can also be combined with other plots, such as bar plots, generated in `ggplot2`. This is often useful for displaying position-specific attributes, such as sequences in an alignment (Fig. 1j). Such functionality further aids in honing the understanding provided by a figure by making the visualization appropriate in the context of the data.

3 Conclusion

`ggseqlogo` is a package for preparing publication-quality sequence logos in R, while providing high degree of flexibility and customizability through integration with `ggplot2`. This addresses many of the fallbacks of the current logo-drawing packages. Furthermore, `ggseqlogo` was developed to tightly integrate with `ggplot2` and be easily expandable to alternative height methods, making it a suitable visualization platform for implementing various height metrics in sequence logos.

Conflict of Interest: none declared.

References

Bembom,O. (2016) *seqLogo: Sequence logos for DNA sequence alignments.* R package version 1.40.0.

Crooks,G.E. *et al.* (2004) WebLogo: a sequence logo generator. *Genome Res.*, **14**, 1188–1190.

Deplancke,B. *et al.* (2016) The genetics of transcription factor DNA binding variation. *Cell*, **166**, 538–554.

Jessen,L.E. *et al*. (2013) SigniSite: identification of residue-level genotype-phenotype correlations in protein multiple sequence alignments. *Nucleic Acids Res*., **41**, W286–W291.

Ou,J. and Zhu,L.J. (2017) *motifStack: Plot stacked logos for single or multiple DNA, RNA and amino acid sequence*. R package version 1.20.0.

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

Wagih,O. *et al*. (2015) MIMP: predicting the impact of mutations on kinase-substrate phosphorylation. *Nat. Methods*, **12**, 531–533.

Wickham,H. (2009) *ggplot2: Elegant Graphics for Data Analysis*. Springer-Verlag, New York.