The Journal of Open Source Software

ExonViz: Transcript visualization made easy

Redmar van den Berg [1,2,3], **Marlen C. Lauffer** [1,2], **and Jeroen F. J. Laros** [1,4]

1 Department of Human Genetics, Leiden University Medical Center, The Netherlands **2** Dutch Center for RNA Therapeutics, Department of Human Genetics, Leiden University Medical Center, The Netherlands **3** Department of Hematology, Leiden University Medical Center, The Netherlands **4** Department of Bioinformatics and Computational Services, National Institute of Public Health and the Environment, The Netherlands

DOI: 10.xxxxxx/draft

Software

- Review
- Repository
- Archive

Editor:

Submitted: 17 October 2025
Published: unpublished

Summary

Transcripts of a gene contain one or more **exons**, which encode the functional parts of the transcript, and **introns**, which are removed in a process called **splicing**. A single gene typically encodes multiple **transcripts** by including different exons. Protein coding genes include one or more coding exons, which encode a protein using three-letter sequences called **codons**. These codons do not necessarily coincide with exon boundaries, because a single codon can span two exons. If the codon boundaries of adjacent exons are not aligned, an often detrimental **frame shift** is introduced. Taking exon boundary reading frames, which we will call **exon boundary frames**, into account is crucial when considering the effect of mutations and when designing genetic therapies such as exon skipping.

ExonViz is a Python package and web application that creates biologically accurate RNA transcript figures, including features such as coding regions, genetic variants and exon boundary frames.

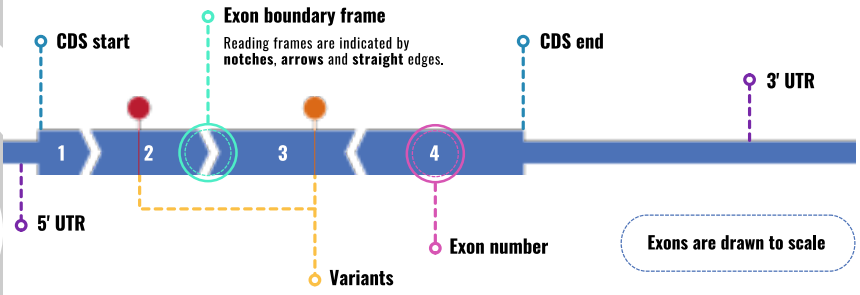

Figure 1: Example transcript highlighting ExonViz features. **5' UTR**: Non coding region at the start of the transcript. **CDS start**: Start of the coding region. **CDS end**: End of the coding region. **3' UTR**: Non coding region at the end of the transcript.

Statement of need

Visualization of transcripts, including features like exon boundary frames, coding and non coding regions is important within the field of clinical and human genetics (Walker et al., 2023). Illustrating the exon structure and the location of variants is common practice, especially when new genes, variants or transcripts have been discovered. These illustrations are also used to assess potential genetic treatment options (e.g., canonical exon skipping), in teaching settings, in diagnostics, to identify mutational hotspots and for genetic counseling. In particular

knowledge about the exon boundary frames aids in the assessment of the pathogenicity of genetic variants using the ACMG-AMP guidelines (Richards et al., 2015), when evaluating exon spanning deletions (Cheerie et al., 2025) and when interpreting the effects of splice altering variants (Walker et al., 2023).

To date, most people have to resort to manually drawing transcripts with tools like Illustrator, Photoshop or BioRender, or forgo illustrations altogether. Creating transcript visualizations should be quick and easy to be utilized in clinical and day to day settings, rather than to create a bespoke figure for a manuscript or presentation.

State of the field

Several tools have been made available to visualize various aspects of genes and transcripts. ggtranscript (Gustavsson et al., 2022) and wiggleplotr (Alasoo, 2017)) can visualize transcripts and exons, while tools like genepainter (Mühlhausen et al., 2015) or Swan (Reese & Mortazavi, 2021) can be used to visualize different transcript isoforms. Variants can be shown on the transcript with Variant View (Ferstay et al., 2013). However, all of these tools require substantial expertise to setup and retrieve the required transcript models, which make them hard to use for users with minimal technical expertise. Furthermore, none of them have the option to indicate exon boundary frames.

To our knowledge there are currently no easily usable tools available which allow a non-technical user to quickly draw all features required for a comprehensive overview of a transcript's structure and the location of variants of interest.

Software design

ExonViz is written in Python 3, its web interface is build using Flask. To avoid the complexities of retrieving and processing various different transcript definitions, ExonViz uses the public Mutalyzer API (Lefter et al., 2021) to fetch transcript annotations. This gives ExonViz access to all transcripts defined in the RefSeq (O'Leary et al., 2016) and Ensembl (Harrison et al., 2023) databases across many species, ranging from human and mouse to fruit fly and coelacanth.

Reverse strand transcript are inverted so that ExonViz always visualizes transcripts in their forward orientation. This avoids the complications that come with the inverted annotations for transcripts which are annotated on the reverse strand of the chromosome. Variants are assigned to their corresponding exon, which also contains the size, coding region and other features which are required to draw an exon. Exons can be split to ensure they do not exceed the specified page width, analogous to how long words can be split over multiple lines.

ExonViz can also read and write the normalized exon and variant models, allowing the user to specify custom transcripts and exons in a simple TSV format.

Method

ExonViz visualizes the exon boundary frames by using different shapes for the boundary of exons. Figure 2 shows all possible combinations of exon and codon boundaries, and the corresponding exon boundary shapes. When the exon and codon boundaries coincide (frame 0) the exons are drawn with a straight edge, as is the case of exon 1 and 2. Exon 2 ends one base into the codon (in frame 1), which is drawn using an arrow on the end of the exon. Exon 3 starts in frame 1, and is drawn with a notch at the start of the exon. This is reversed for the boundary between exons 3 and 4, which is in frame 2. Since the exons of a transcript should fit together, exons in conflicting frames (*e.g.* because of a frame shift inducing variant) are easily spotted due to the fact that the exon boundaries do not fit together.

The Journal of Open Source Software

Figure 2: Visualization of the relation between codons and exon boundary frames. The shapes of the exons illustrate the relation between the exon boundaries and the codon boundaries.

The output of ExonViz is an SVG figure generated using the svg-py library, which can be used directly or modified using modern graphical editing programs. It is also possible to output the transcript and variants in TSV format, edit the transcript using any text editor or spreadsheet program, and draw the modified transcript using ExonViz. The online documentation has a number of examples of custom transcripts that can be visualized this way.

Research impact statement

ExonViz has proven to be a useful resource to quickly visualize exon reading frames and check the location of variants in a transcript. ExonViz is actively being used in the field of personalized medicine and is one of the recommended resources in the latest consensus guidelines for assessing pathogenic variants for RNA therapies (Cheerie et al., 2025). In addition, the ExonViz website has been used to generate over 8000 transcript figures between September 2023 and September 2025.

Conclusion

To our knowledge, ExonViz is the first publicly available application that allows for automatic visualization of transcripts with additional features such as exon boundary frames and variants along the transcript. ExonViz can be used for illustrations within publications, assessment of treatment options, teaching purposes and genetic counseling. Figures generated by ExonViz are free to use under the Creative Commons BY license. Furthermore, we allow the user to construct their own transcripts, for example to visualize non-standard exons or alternative isoforms. ExonViz can be accessed as a web application via exonviz.rnatherapy.nl or installed via PyPI. The source code is available on Github.

AI usage disclosure

No generative AI tools have been used for ExonViz, the documentation or this manuscript.

Acknowledgments

We would like to thank the members of the Dutch Center for RNA Therapeutics for their ideas, suggestions and their feedback on earlier versions of ExonViz. We also thank Maximilian Haeussler and his colleagues at the UCSC for their efforts implementing exon boundary frame information into the UCSC Genome Browser. We also thank Nanieke van den Berg for creating Figure 1.

References

Alasoo, K. (2017). Wiggleplotr: Make read coverage plots from BigWig files. *Bioconductor. Doi*, *10*, B9.

Cheerie, D., Meserve, M. M., Beijer, D., Kaiwar, C., Newton, L., Tavares, A. L. T., Verran, A. S., Sherrill, E., Leonard, S., Sanders, S. J., & others. (2025). Consensus guidelines for

assessing eligibility of pathogenic DNA variants for antisense oligonucleotide treatments. *The American Journal of Human Genetics*. https://doi.org/10.1016/j.ajhg.2025.02.017

Ferstay, J. A., Nielsen, C. B., & Munzner, T. (2013). Variant view: Visualizing sequence variants in their gene context. *IEEE Transactions on Visualization and Computer Graphics*, *19*(12), 2546–2555. https://doi.org/10.1109/TVCG.2013.214

Gustavsson, E. K., Zhang, D., Reynolds, R. H., Garcia-Ruiz, S., & Ryten, M. (2022). Ggtranscript: An r package for the visualization and interpretation of transcript isoforms using ggplot2. *Bioinformatics*, *38*(15), 3844–3846. https://doi.org/10.1093/bioinformatics/btac409

Harrison, P. W., Amode, M. R., Austine-Orimoloye, O., Azov, A. G., Barba, M., Barnes, I., Becker, A., Bennett, R., Berry, A., Bhai, J., Bhurji, S. K., Boddu, S., BrancoÂ Lins, P. R., Brooks, L., Ramaraju, S. B., Campbell, L. I., Martinez, M. C., Charkhchi, M., Chougule, K., … Yates, A. D. (2023). Ensembl 2024. *Nucleic Acids Research*, *52*(D1), D891–D899. https://doi.org/10.1093/nar/gkad1049

Lefter, M., Vis, J. K., Vermaat, M., Dunnen, J. T. den, Taschner, P. E. M., & Laros, J. F. J. (2021). Mutalyzer 2: next generation HGVS nomenclature checker. *Bioinformatics*, *37*(18), 2811–2817. https://doi.org/10.1093/bioinformatics/btab051

Mühlhausen, S., Hellkamp, M., & Kollmar, M. (2015). GenePainter v. 2.0 resolves the taxonomic distribution of intron positions. *Bioinformatics*, *31*(8), 1302–1304. https://doi.org/10.1093/bioinformatics/btu798

O'Leary, N. A., Wright, M. W., Brister, J. R., Ciufo, S., Haddad, D., McVeigh, R., Rajput, B., Robbertse, B., Smith-White, B., Ako-Adjei, D., & others. (2016). Reference sequence (RefSeq) database at NCBI: Current status, taxonomic expansion, and functional annotation. *Nucleic Acids Research*, *44*(D1), D733–D745. https://doi.org/10.1093/nar/gkv1189

Reese, F., & Mortazavi, A. (2021). Swan: A library for the analysis and visualization of long-read transcriptomes. *Bioinformatics*, *37*(9), 1322–1323.

Richards, S., Aziz, N., Bale, S., Bick, D., Das, S., Gastier-Foster, J., Grody, W. W., Hegde, M., Lyon, E., Spector, E., & others. (2015). Standards and guidelines for the interpretation of sequence variants: A joint consensus recommendation of the american college of medical genetics and genomics and the association for molecular pathology. *Genetics in Medicine*, *17*(5), 405–423. https://doi.org/10.1038/s41525-022-00329-6

Walker, L. C., Hoya, M. de la, Wiggins, G. A., Lindy, A., Vincent, L. M., Parsons, M. T., Canson, D. M., Bis-Brewer, D., Cass, A., Tchourbanov, A., & others. (2023). Using the ACMG/AMP framework to capture evidence related to predicted and observed impact on splicing: Recommendations from the ClinGen SVI splicing subgroup. *The American Journal of Human Genetics*, *110*(7), 1046–1067. https://doi.org/10.1016/j.ajhg.2023.06.002