The Journal of Open Source Software

Sugar: A Python framework for bioinformatics

Tom Eulenfeld [1]

1 Bioinformatics/High-Throughput Analysis, Friedrich Schiller University Jena, Germany

DOI: 10.21105/joss.08122

Software

- Review
- Repository
- Archive

Editor: Olexandr Konovalov

Reviewers:

- @tushardave26
- @widdowquinn

Submitted: 02 February 2025
Published: 24 July 2025

Summary

Modern bioinformatics requires the use of a variety of tools (e.g. BLAST for homology detection, Altschul et al., 1990) and databases (e.g. GenBank, Benson et al., 2012). Custom scripts often need to use the results of these tools, which are available in different formats.

Sugar is a Python framework for bioinformatics which aims to facilitate rapid application development. The package allows to read and write various sequence and annotation formats, i.e. FASTA, GenBank, Stockholm, GFF, GTF, BLAST and others. Since sugar uses a plugin system for reading and writing, new file formats can be added not only within the sugar package, but also within other packages, allowing for low barrier inclusion of new formats. Sugar includes classes for representing DNA/RNA sequences and annotations. The main functionality is exposed through methods of these classes and is therefore readily available.

Sugar can be used by researchers and students of bioinformatics alike. The package has already been used as a library in the AnchoRNA package (Eulenfeld et al., 2025) and in several scripts that form the basis of the Ritsch et al. (2024) publication. Sugar can be installed from PyPI. Online documentation and tutorials are available on the GitHub project site.

Statement of need

Other well-known frameworks for bioinformatics are Biopython (Cock et al., 2009) and Biotite (Kunzmann & Hamacher, 2018). Launched in 2000, Biopython contains a large collection of freely available tools. In contrast, sugar tries to focus on the basics of IO as well as sequence and annotation (resp. feature) manipulation. In Table 1, we compare the code for reading a FASTA sequence file using sugar and Biopython, respectively. In addition, the sugar code example demonstrates a typical bioinformatics file manipulation task, which is not easily possible with Biopython alone due to its lack of a GFF writer. The excellent Biotite package also handles DNA/RNA sequences and has an additional focus on protein structures. Sugar should not be seen as an alternative to these other packages, but rather as a complement. Therefore, adapters are provided, which can easily convert sugar objects into the corresponding objects of the Biopython and Biotite packages, and vice versa. In addition, sugar annotation objects can be plotted employing the DNA features viewer package (Zulkower & Rosser, 2020).

Table 1: Comparison of code to read a FASTA file using sugar (left) and Biopython (right). The code example using sugar also demonstrates reading of a BLAST result file, attaching the hit features to the sequences while discarding features belonging to different sequences, and writing the sequences with the corresponding features to a GFF file including a FASTA directive.

```
from sugar import read, read_fts
seqs = read('seqs.fasta')
fts = read_fts('hits.blastn')
seqs.fts = fts
seqs.write('seqs_annotated.gff')
```

```
from Bio import SeqIO
seqs = list(SeqIO.parse('seqs.fasta', 'fasta'))
```

Acknowledgements

The use of a plugin system for reading and writing was inspired by ObsPy (Krischer et al., 2015), a Python framework for seismology. I would like to thank the reviewers, Tushar Dave and Leighton Pritchard. This project was funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Germany's Excellence Strategy – EXC 2051 – Project ID 390713860.

References

Altschul, S. F., Gish, W., Miller, W., Myers, E. W., & Lipman, D. J. (1990). Basic local alignment search tool. *Journal of Molecular Biology*, *215*(3), 403–410. https://doi.org/10.1016/s0022-2836(05)80360-2

Benson, D. A., Cavanaugh, M., Clark, K., Karsch-Mizrachi, I., Lipman, D. J., Ostell, J., & Sayers, E. W. (2012). GenBank. *Nucleic Acids Research*, *41*(D1), D36–D42. https://doi.org/10.1093/nar/gks1195

Cock, P. J. A., Antao, T., Chang, J. T., Chapman, B. A., Cox, C. J., Dalke, A., Friedberg, I., Hamelryck, T., Kauff, F., Wilczynski, B., & Hoon, M. J. L. de. (2009). Biopython: Freely available Python tools for computational molecular biology and bioinformatics. *Bioinformatics*, *25*(11), 1422–1423. https://doi.org/10.1093/bioinformatics/btp163

Eulenfeld, T., Triebel, S., & Marz, M. (2025). AnchoRNA: Full virus genome alignments through conserved anchor regions. In *bioRxiv*. https://doi.org/10.1101/2025.01.30.635689

Krischer, L., Megies, T., Barsch, R., Beyreuther, M., Lecocq, T., Caudron, C., & Wassermann, J. (2015). ObsPy: A bridge for seismology into the scientific Python ecosystem. *Computational Science & Discovery*, *8*(1), 014003. https://doi.org/10.1088/1749-4699/8/1/014003

Kunzmann, P., & Hamacher, K. (2018). Biotite: A unifying open source computational biology framework in Python. *BMC Bioinformatics*, *19*(1). https://doi.org/10.1186/s12859-018-2367-z

Ritsch, M., Eulenfeld, T., Lamkiewicz, K., Schoen, A., Weber, F., Hölzer, M., & Marz, M. (2024). Endogenous Bornavirus-like elements in bats: Evolutionary insights from the conserved riboviral L-gene in microbats and its antisense transcription in Myotis daubentonii. In *Viruses* (No. 8; Vol. 16, p. 1210). https://doi.org/10.3390/v16081210

Zulkower, V., & Rosser, S. (2020). DNA Features Viewer: A sequence annotation formatting and plotting library for Python. *Bioinformatics*, *36*(15), 4350–4352. https://doi.org/10.1093/bioinformatics/btaa213