DOI 10.18699/SBB-2022-672

Pygenomics: Python package for processing genomic intervals and bioinformatic data formats

Tamazian G.*, Cherkasov N., Kanapin A., Samsonova A.
Institute of Translational Biomedicine, Saint Petersburg State University, St. Petersburg, Russia
**g.tamazian@spbu.ru*

Key words: computational analysis, data formats, genomic intervals

Motivation and Aim: Computational analysis of genome sequencing data and its derivatives such as assembled genome sequences, annotated genes, repeats, and genomic variants plays an important role in modern bioinformatic studies. Such studies are usually implemented in the form of computational pipelines which combine invoking bioinformatic programs (e.g., genome assemblers or gene prediction programs) with extra routines that convert input or output files of the programs between various bioinformatic data formats and query the produced files. Many bioinformatic data formats are based on genomic intervals [1], and thus querying files in such formats requires operating with the intervals.
Methods and Algorithms: We present pygenomics – an open-source Python package that provides routines for reading and writing bioinformatic data in various formats and operating with genomic intervals. Pygenomics is implemented in pure Python and does not require any other libraries except for the Python standard library. The package is developed according to the functional programming paradigm that ensures immutability of the package entities, absence of side effects in the package functions except for ones related to input-output (I/O), and extendable stream-based I/O. The package source code is provided with type hints [2] and a property-based testing framework that is based on the Hypothesis package [3]. Interval operations in the package are implemented using implicit intervals trees [4].
Results: Pygenomics implements reading and writing from a number of bioinformatic data formats, including BED, GTF, GFF3, SAM, and VCF. The package provides the application programming interface (API) and the command-line interface (CLI) for calling its routines from a source code or as stand-alone programs respectively. Implementation of pygenomics in pure Python allows to seamlessly incorporate the package routines into Snakemake [5] pipelines and to run them using CPython and PyPy interpreters. Absence of external dependencies, implementation in pure Python, and the property-based testing framework facilitate deployment of pygenomics to various computational platforms. Examples of Python scripts that use the pygenomics package routines are available in the public GitLab repository: https://gitlab.com/gtamazian/pygenomics_examples.
Conclusion: Pygenomics provides routines that facilitate development of computational tools and pipelines for working with bioinformatic data. The routines are implemented in a transparent and robust way due to following the functional programming paradigm and utilizing features provided by the Python programming language and its standard library. The package with its source code is publicly available on GitLab: https://gitlab.com/gtamazian/pygenomics
Acknowledgements: This research was funded by RSF, grant 20-14-00072.

References

1. Lawrence M., Huber W., Pagès H., et al. Software for computing and annotating genomic ranges. *PLoS Comput Biol*. 2013;9(8):e1003118. doi: 10.1371/journal.pcbi.1003118.
2. Van Rossum G., Jukka L., Langa Ł. PEP 484 – Type hints. Python Enhancement Proposals. 2014.
3. MacIver D.R., Hatfield-Dodds Z. Hypothesis: A new approach to property-based testing. *J Open Source Softw.* 2019;4(43):1891. doi: 10.21105/joss.01891.
4. Li H., Rong J. Bedtk: finding interval overlap with implicit interval tree. *Bioinformatics.* 2021;37(9):1315-1316. doi: 10.1093/bioinformatics/btaa827.
5. Köster J., Rahmann S. Snakemake – a scalable bioinformatics workflow engine. *Bioinformatics*. 2012;28(19):2502-2502. doi: 10.1093/bioinformatics/bts480.