Array programming for Biology

Authors

- **Knut Dagestad Rand** ✉
  Biomedical Informatics research group, Department of Informatics, University of Oslo, Oslo, Norway; Centre for Bioinformatics, University of Oslo, Oslo, Norway

- **Ivar Grytten**
  Biomedical Informatics research group, Department of Informatics, University of Oslo, Oslo, Norway

- **Milena Pavlović**
  Biomedical Informatics research group, Department of Informatics, University of Oslo, Oslo, Norway; UiORealArt Convergence Environment, University of Oslo, Oslo, Norway

- **Chakravarthi Kanduri**
  Biomedical Informatics research group, Department of Informatics, University of Oslo, Oslo, Norway; UiORealArt Convergence Environment, University of Oslo, Oslo, Norway

- **Geir Kjetil Sandve**
  Biomedical Informatics research group, Department of Informatics, University of Oslo, Oslo, Norway; Centre for Bioinformatics, University of Oslo, Oslo, Norway; UiORealArt Convergence Environment, University of Oslo, Oslo, Norway

✉ — Correspondence possible via GitHub Issues or email to Knut Dagestad Rand <knutdr@math.uio.no>.

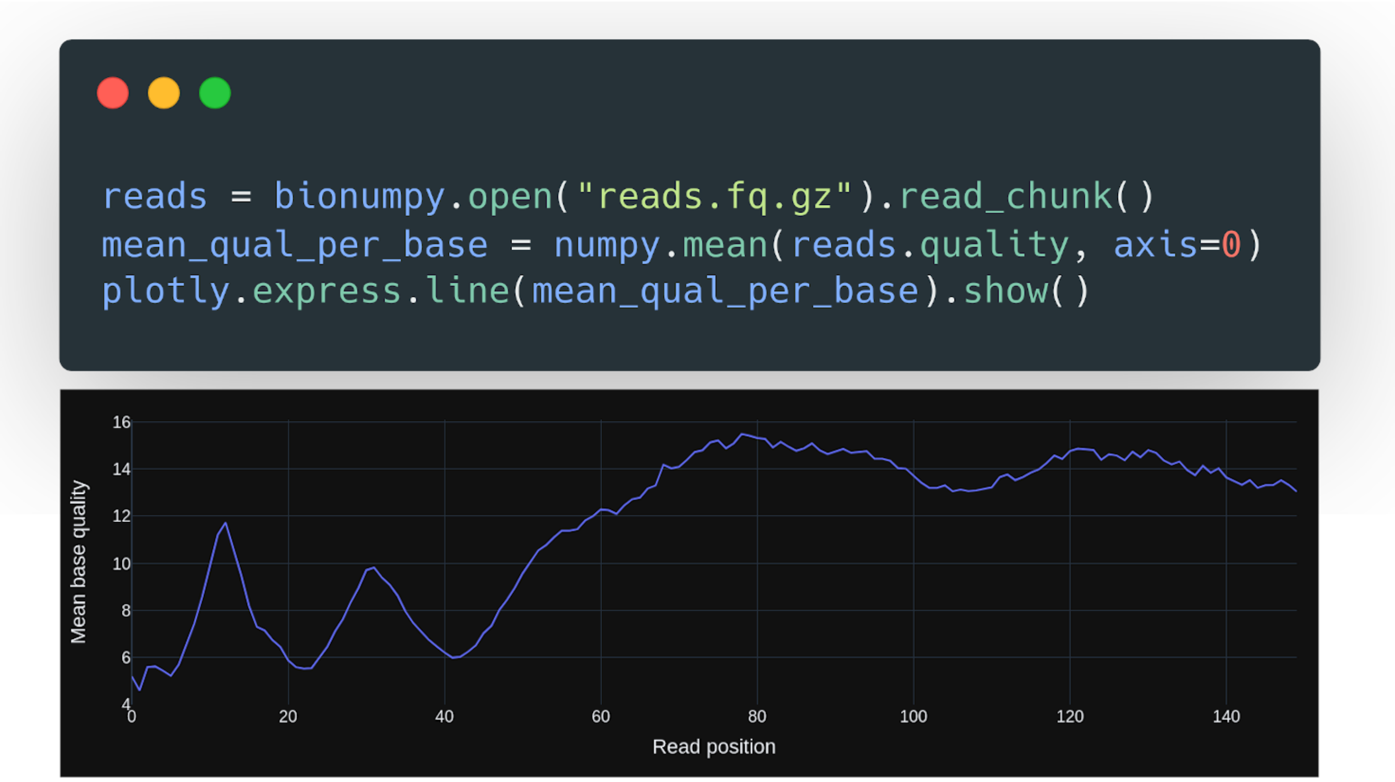
Array Programming for Biology

Python is a widely used programming language for scientific computing, in large part due to the powerful *array programming* library NumPy [1], which makes it easy to write clean, vectorized and computationally efficient code for handling large datasets. A challenge with using array programming in biology is that the data is often non-numeric and variable-length (e.g. DNA sequences), inhibiting out-of-the-box use of standard array programming techniques. This may push bioinformaticians to instead rely on complex, custom pipelines of UNIX commands that are non-transparent and error-prone. Furthermore, the impracticality of developing efficient code directly in high-level languages like Python has led to tool developers almost exclusively relying on low-level languages like C and C++ (or hybrid implementations using e.g. Cython [2] or Numba [3]), making it more difficult for computational biologists to understand and contribute to core methods in the field.

We present the BioNumPy package, which enables efficient and intuitive array programming on biological data in Python. Internally, this is handled by a ragged data structure (similar to [4]) that numerically encodes variable-length sequence data in continuous memory blocks, along with arrays describing the sequence lengths and encoding (see Supplementary Material). BioNumPy supports a broad range of bioinformatics analyses, with the main philosophy being that data structures should behave as closely as possible to standard numeric NumPy arrays. This means that BioNumPy is easy to learn for users familiar with NumPy or with array programming languages like R and Matlab. BioNumPy is open-source and freely available at https://github.com/bionumpy/bionumpy/, and can be installed through the Python package manager pip. BioNumPy comes with extensive documentation and a user guide that makes it easy to use for a wide range of molecular biology datasets and problems.

BioNumPy is able to read and write biological datasets (e.g FASTQ, FASTA, BED, GTF, BAM, or VCF-files) directly to/from NumPy-like data structures, providing efficient access to the data through an intuitive and easy-to-use API. The data can then be processed and analysed efficiently using a NumPy-like interface or be combined with other biological data by using for instance BioPython [5] which has a richer ecosystem.

In Figure 1 we showcase BioNumPy by reading sequenced reads from a FASTQ file and plotting the average base quality per read position. Both the sequences and base qualities are represented in NumPy-compatible arrays, so that NumPy-functionality like e.g. *np.mean* can be used.



```
reads = bionumpy.open("reads.fq.gz").read_chunk()
mean_qual_per_base = numpy.mean(reads.quality, axis=0)
plotly.express.line(mean_qual_per_base).show()
```

Figure 1: Example of BioNumPy usage. We read a chunk from a FASTQ file, use NumPy to get the average base quality per read position and use Plotly to plot the results. Axis labels are not included in the code and have been added for clarity.

We show through a range of experiments that BioNumPy is considerably faster than existing Python packages for common bioinformatics tasks and, in many cases, as fast as tools written in C/C++ (see Supplementary Material). We also showcase how BioNumPy enables seamless machine learning on biological sequence data by reproducing parts of a recent machine learning benchmark study [6] using very few lines of code (Supplementary Material).

In conclusion, we believe that BioNumPy bridges a long-lasting gap by making array programming practical for the field of biology.

References

1. **Array programming with NumPy**
Charles R Harris, KJarrod Millman, Stéfan J van der Walt, Ralf Gommers, Pauli Virtanen, David Cournapeau, Eric Wieser, Julian Taylor, Sebastian Berg, Nathaniel J Smith, … Travis E Oliphant
Nature (2020-09-16) https://doi.org/ghbzf2
DOI: 10.1038/s41586-020-2649-2 · PMID: 32939066 · PMCID: PMC7759461

2. **Cython: The Best of Both Worlds**
Stefan Behnel, Robert Bradshaw, Craig Citro, Lisandro Dalcin, Dag Sverre Seljebotn, Kurt Smith
Computing in Science &amp; Engineering (2011-03) https://doi.org/fcvqn4
DOI: 10.1109/mcse.2010.118

3. **Numba**
Siu Kwan Lam, Antoine Pitrou, Stanley Seibert
Proceedings of the Second Workshop on the LLVM Compiler Infrastructure in HPC (2015-11-15) https://doi.org/gf3nks
DOI: 10.1145/2833157.2833162

4. **Awkward Array**
Jim Pivarski, Ianna Osborne, Ioana Ifrim, Henry Schreiner, Angus Hollands, Anish Biswas, Pratyush Das, Santam Roy Choudhury, Nicholas Smith, Manasvi Goyal
Zenodo (2024-01-12) https://doi.org/gtw4wd
DOI: 10.5281/zenodo.4341376

5. **Biopython: freely available Python tools for computational molecular biology and bioinformatics**
Peter JA Cock, Tiago Antao, Jeffrey T Chang, Brad A Chapman, Cymon J Cox, Andrew Dalke, Iddo Friedberg, Thomas Hamelryck, Frank Kauff, Bartek Wilczynski, Michiel JL de Hoon
Bioinformatics (2009-03-20) https://doi.org/d7zwd2
DOI: 10.1093/bioinformatics/btp163 · PMID: 19304878 · PMCID: PMC2682512

6. **Benchmarking of deep neural networks for predicting personal gene expression from DNA sequence highlights shortcomings**
Alexander Sasse, Bernard Ng, Anna E Spiro, Shinya Tasaki, David A Bennett, Christopher Gaiteri, Philip L De Jager, Maria Chikina, Sara Mostafavi
Nature Genetics (2023-11-30) https://doi.org/gs7569
DOI: 10.1038/s41588-023-01524-6 · PMID: 38036778