Wei J.
Kyiv College at Qilu University of Technology
Hretskyi I. O.
Kyiv National University of Technologies and Design

BIOPYTHON: IMPORTANT APPLICATION TOOLS IN THE FIELD OF BIOINFORMATICS AND COMPUTATIONAL MOLECULAR BIOLOGY

Abstract. *Understanding the fundamentals of nucleic acid sequences that encode proteins is one thing; however, the advent of high-throughput next-generation sequencing has ushered in an era of vast biological data. To advance our scientific endeavors amidst this deluge of information, we must harness a suite of life science computing tools, with bioinformatics being a key example. Biopython stands out as a valuable resource in this context–a collection of free Python modules dedicated to computational molecular biology. It encompasses a rich array of modules, scripts, and web links that connect to a multitude of online resources. The scientific community has warmly embraced Biopython for its prowess as a parser for diverse file formats and its facilitation of access to online sequence analysis services. In this paper, we aim to elucidate the pivotal role Biopython plays in fostering the field of bioinformatics.*

Keywords: *Biopython, Bioinformatics, Computational Molecular Biology, High-throughput Sequencing, Next-generation Sequencing, Nucleic Acid Sequences.*

Вей Ц., бакалавр
Київський інститут Технологічного університету Цілу
Грецький І. О., доц.
Київський національний університет технологій та дизайну

BIOPYTHON: ВАЖЛИВІ ПРИКЛАДНІ ЗАСОБИ В ГАЛУЗІ БІОІНФОРМАТИКИ ТА ОБЧИСЛЮВАЛЬНОЇ МОЛЕКУЛЯРНОЇ БІОЛОГІЇ

Анотація. *Поява високопродуктивного секвенування наступного покоління відкрила еру величезних обсягів біологічних даних. При просуванні наукових досліджень серед цього потоку інформації виникає потреба використовувати різноманітні інструменти для обробки біологічної інформації, серед яких одне з ключових місць займає біоінформатика. Biopython виділяється як цінний ресурс у цьому контексті – це колекція безкоштовних модулів Python, орієнтованих на комп'ютерну молекулярну біологію. Biopython містить широкий набір модулів, скриптів і вебпосилань, що з'єднують користувачів з численними онлайн-ресурсами. Наукова спільнота високо оцінює Biopython за його можливості парсингу різних форматів файлів і доступу до онлайн-сервісів для аналізу послідовностей.*

Ключові слова: *Biopython, біоінформатика, обчислювальна молекулярна біологія, високопродуктивне секвенування, секвенування наступного покоління, послідовності нуклеїнових кислот.*

Introduction. The advent of high-throughput next-generation sequencing technology has sparked an exponential surge in biometric data across genomics, proteomics, and other life sciences. This deluge of data has not only quickened the pace of research but also significantly propelled scientific advancement through advanced biometric data analysis. In the pursuit of seamlessly integrating life science applications with theoretical frameworks, the field of bioinformatics has stepped in, bringing with it a cornucopia of tools, data, and online services [1]. Among these resources, Biopython stands out, with its adoption rate consistently on the rise.

As a bioinformatics processing service, Biopython is a treasure trove of online resources, further expanded through web links [2]. It boasts the ability to recognize and parse a diverse array of bioinformatics file formats, including BLAST, FASTA, and GenBank [3]. Moreover, Biopython serves as a gateway to NCBI, ExPASy online services, and an arsenal of powerful tools such as sequence alignment, motif analysis, clustering algorithms, structural biology modules, and phylogenetic analysis modules [4].

The significance of Biopython, however, transcends its technical prowess. It empowers bioinformatics researchers and molecular biologists with a robust, adaptable, and user-friendly programming toolkit. By presenting a standardized interface for engaging with biological data, Biopython enhances research efficiency and minimizes the time spent on data preprocessing, allowing researchers to devote more focus to data analysis and interpretation. Furthermore, Biopython's compatibility with multiple operating systems and file formats fosters cross-platform research, simplifying the sharing and analysis of data across various platforms. This has, in turn, ignited the development of bioinformatics by offering a suite of modern programming tools that nurture the growth of new bioinformatics methods and algorithms, thereby driving the field forward.

Bioinformatics tools and databases like Jalview and the SIB Swiss Institute of Bioinformatics are pivotal for multiple sequence comparison editing, visualization, and analysis. They offer researchers a wealth of resources in proteins and proteomes, glycomics, structural biology, genes and genomes, and evolutionary and systems biology. Equipped with built-in DNA, RNA, and protein sequence and structure visualization and analysis capabilities, as well as linked views of aligned DNA and protein products, these databases and software tools provide formidable support for processing and analyzing the ever-growing volume of biometric data. This facilitates research and development in the life sciences, enabling researchers to analyze data more efficiently, spur scientific discovery, and catalyze innovation in the realm of bioinformatics.

Biopython, revered as a pivotal application in the domains of bioinformatics and computational molecular biology, holds an esteemed position among scientific researchers, especially those in biostatistics. Its well-earned reputation is a result of its unique features and the dedicated efforts of its development team. Comprised of a committed group of volunteers, these individuals not only sustain the platform's daily functions but are also highly responsive to user feedback, ensuring continuous functional refinement and enhancement.

Biopython is designed to keep pace with the latest trends in life science research, excelling in the management and analysis of extensive datasets through advanced algorithmic approaches. This capability is essential to its cutting-edge and innovative character. The platform is tailored to meet the needs of a wide array of users by adopting a robust interdisciplinary strategy, thus enhancing its practicality across various scientific disciplines.

Through the ongoing integration of user suggestions and the fostering of collaborative efforts, Biopython maintains its state-of-the-art functionalities, mirroring the dynamic evolution of life sciences. Its dedication to interdisciplinarity extends its utility and establishes Biopython as a multifaceted solution adaptable to the unique requirements of different scientific domains. This flexibility is a hallmark of Biopython's adaptability and underscores its importance as an indispensable tool in contemporary biological research.

Basic features. Biopython has emerged as an invaluable programming tool in the realm of bioinformatics, boasting a suite of robust features and an intuitive design that has endeared it to developers worldwide. This powerful toolkit significantly lightens the load for developers tasked with parsing the intricacies of bioinformatics file formats by equipping them with a collection of meticulously crafted, reusable libraries [5]. These libraries not only streamline the coding process but also bolster the clarity and maintainability of the code, making it more accessible to both new and seasoned programmers.

When it comes to sequence processing and file parsing, Biopython transcends the limitations of conventional string manipulation by introducing feature-rich sequence objects [6]. These objects are endowed with an array of biologically relevant methods that go beyond the scope of standard operations. For example, users can effortlessly carry out sophisticated molecular biology procedures such as determining the complementary strands of a sequence, transcription, and translation–tasks that are pivotal to molecular biology research. Biopython simplifies these complex operations, making them not only feasible but also swift and efficient.

The SeqIO module in Biopython serves as a versatile data processing tool, akin to a Swiss Army knife, by providing a unified interface for reading and writing a variety of sequence file formats [7]. This capability empowers users to handle data from diverse databases and studies with a consistent approach, thereby significantly enhancing the adaptability and productivity of data processing tasks.

Biopython's prowess in multiple sequence alignment is equally noteworthy. It offers an array of alignment tools, encompassing both global and local alignment algorithms, along with an assortment of sequence analysis utilities. These tools are adept at assisting researchers in calculating sequence GC content, identifying open reading frames, and much more. Moreover, they are proficient in uncovering sequence homologies, a critical aspect for deciphering the roles of genes and proteins. The comprehensive nature of these functions not only facilitates the research process for scientists but also streamlines data organization, thereby markedly enhancing research productivity.

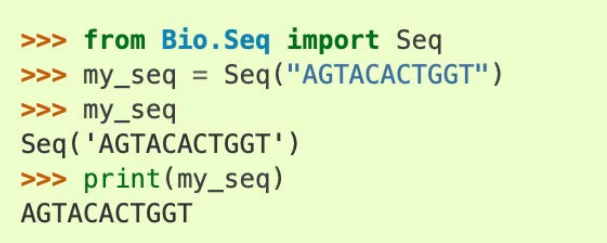
Biopython's integrated online search engine is a standout feature that has garnered widespread acclaim among researchers. It permits users to directly fetch data from renowned databases like NCBI and ExPASy without the need to exit their programming environment. This feature substantially boosts the efficiency of data retrieval and analysis, empowering researchers to swiftly extract pertinent information from extensive biological databases [8].

Moreover, Biopython's interoperability with other databases through BioSQL greatly simplifies the storage and management of sequence data. This feature minimizes the time spent on data compilation and fosters collaboration across different platforms, enabling seamless data sharing and analysis.

The ability of Biopython to perform quality control and filtering on sequencing data is indispensable for achieving reliable sequencing outcomes. It ensures the precision of data, thereby laying a solid foundation for subsequent analysis and research endeavors. In the age of prevalent high-throughput sequencing technologies, this feature is of paramount importance. Biopython has solidified its status as an essential tool for bioinformatics researchers, thanks to its robust features and user-friendly design. It not only enhances research efficiency but also fuels the advancement of scientific discoveries. With its flexibility and extensibility, Biopython is poised to evolve in tandem with the ever-growing bioinformatics field, presenting a world of boundless opportunities for future research endeavors.

Basic operating procedure. Biopython's foundational operations serve as the bedrock for all data analyses conducted within its platform, and mastering these processes is essential for advancing to more complex tasks such as multi-sequence comparisons and sequence feature analysis [9]. In other words, a thorough understanding of Biopython's core functionalities is a prerequisite for delving into the advanced aspects of sequence feature outputs and multi-sequence comparisons.

Processing Sequences. In the world of bioinformatics, sequences are rightfully central, and Biopython's mechanism for dealing with sequences relies heavily on the Seq object, which is pretty much the primary tool we use when working with biological data. Most of the time when we think of a sequence, we have a string of letters like ‘AGTACACTGGT’ in mind. You can create a Seq object by following these steps – “>>>” indicates that the Python prompt is immediately followed by what you want to type (fig. 1).



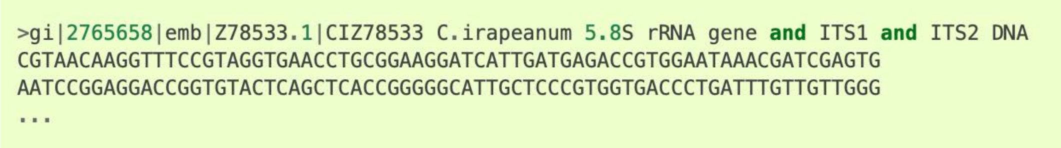
```
>>> from Bio.Seq import Seq
>>> my_seq = Seq("AGTACACTGGT")
>>> my_seq
Seq('AGTACACTGGT')
>>> print(my_seq)
AGTACACTGGT
```

Source: compiled by the author based on [9].

Fig. 1. **Code in BioPython to create a Seq object**

The other most important class is SeqRecord or Sequence Record, which retains additional annotation information about a sequence (as a Seq object), including ID, name, and description, and the Bio.SeqIO module, which reads and writes to the sequence file format, can work with the SeqRecord object.

Simple FASTA parsing example. If you open the FASTA file *ls_orchid.fasta* for lady slipper orchids in your preferred text editor, you'll see that the file begins like this (fig. 2).

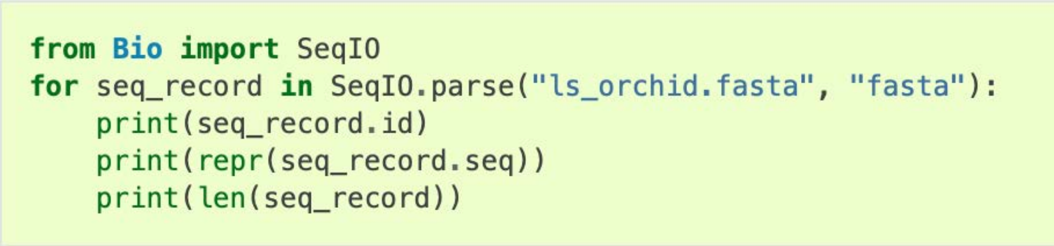


```
>gi|2765658|emb|Z78533.1|CIZ78533 C.irapeanum 5.8S rRNA gene and ITS1 and ITS2 DNA
CGTAACAAGGTTTCCGTAGGTGAACCTGCGGAAGGATCATTGATGAGACCGTGGAATAAACGATCGAGTG
AATCCGGAGGACCGGTGTACTCAGCTCACCGGGGGCATTGCTCCCGTGGTGACCCTGATTTGTTGTTGGG
...
```

Source: compiled by the author based on [9].

Fig. 2. **Example of FASTA file**

It contains 94 records, with each row starting with a ‘>’, (greater than sign) followed immediately by a sequence of one or more rows. Now try the following Python code this (fig. 3).

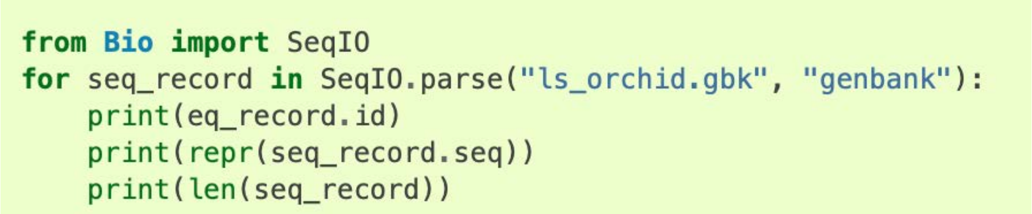


```
from Bio import SeqIO
for seq_record in SeqIO.parse("ls_orchid.fasta", "fasta"):
    print(seq_record.id)
    print(repr(seq_record.seq))
    print(len(seq_record))
```

Source: compiled by the author based on [9].

Fig. 3. **FASTA parsing example in Biopython**

Simple GenBank parsing example. Now let's load a GenBank file *ls_orchid.gbk* – note that the code here is almost identical to the code above that handles the FASTA file – the only difference is that we've changed the filename and the formatting string of the file (fig. 4).



```
from Bio import SeqIO
for seq_record in SeqIO.parse("ls_orchid.gbk", "genbank"):
    print(eq_record.id)
    print(repr(seq_record.seq))
    print(len(seq_record))
```

Source: compiled by the author based on [9].

Fig. 4. **GenBank parsing example in Biopython**

Future Development. The advent of next-generation sequencing technologies [7] has unleashed a torrent of biological data, presenting researchers with a formidable challenge. Professionals in this field are now required not only to possess a robust foundation in both biology and computer science but also to have a deep understanding of the essence of bioinformatics. It is not enough to simply manipulate ready-made online tools; more importantly, they must be able to discern the underlying nature of biological problems and develop and implement algorithms and scripts to tackle these challenges effectively.

For students in life sciences, acquiring programming skills tailored to bioinformatics is an essential, albeit daunting, task. Consequently, there is an urgent need for the support of robust online platform tools. If Biopython were to allocate more resources to creating educational materials and training courses, it could greatly assist both novice and seasoned researchers in mastering and utilizing this toolkit more efficiently [8]. Meanwhile, the active developer community behind Biopython will continue to propel it forward, introducing new features to address the evolving demands of scientific research.

In the midst of the rapid advancements in bioinformatics, Biopython is anticipated to incorporate more cutting-edge technologies. This includes the application of artificial intelligence and machine learning to sequence analysis, as well as leveraging cloud computing platforms for managing large-scale data analysis [9, 10]. Furthermore, as bioinformatics increasingly intersects with other disciplines, Biopython may expand its interdisciplinary capabilities to support research in precision medicine, agricultural genomics, environmental microbiology, and more. By doing so, it could play a pivotal role in several cutting-edge areas of life sciences [11].

Conclusion. Biopython has emerged as a crucial tool in the hands of researchers within the dynamic and revolutionary domain of bioinformatics, owing to its versatility and forward-thinking perspective [12]. It not only keeps pace with the latest developments in scientific research but also bolstered by an active developer community, Biopython continuously evolves and improves to address the ever-changing demands of the research community [13].

The sophistication of Biopython extends beyond its keen understanding of scientific research trends; it also demonstrates a commitment to being responsive to user needs. By integrating online resources and tools from various platforms, Biopython offers a robust and adaptable platform for life science researchers [14]. This platform enhances capabilities for interdisciplinary applications and aligns with society's aim of nurturing individuals who can navigate multiple fields of study.

The ongoing advancement of Biopython has significantly expedited the scientific research process and hastened the arrival of new discoveries [15]. Equipping researchers with efficient and user-friendly tools, Biopython facilitates the rapid and precise processing and analysis of vast biological data sets [16]. It has been instrumental in driving breakthroughs in critical areas such as genomics, proteomics, and systems biology.

Moreover, Biopython's interdisciplinary character promotes collaboration across different scientific disciplines, offering fresh viewpoints and solutions to intricate scientific challenges [17]. As the life sciences continue to progress and broaden, Biopython is poised to persist in its role as a catalyst for scientific discovery and technological innovation, providing enhanced support for future researchers and encouraging exploration in life sciences to delve deeper and reach further [18].

Biopython is more than just a tool; it is an ecosystem, a hub that brings together researchers, developers, and educators. By reducing the barriers to bioinformatics analysis, it sparks interest and engagement in life sciences among a wider audience, thereby infusing the field with renewed vigor. With the impending integration of artificial intelligence, machine learning, and other cutting-edge technologies with Biopython, the future of bioinformatics research is poised to become increasingly intelligent and personalized. Biopython will continue to play an essential role in this evolution, guiding researchers onto a path of discovery that is brighter and more enlightening.

References

1. Branco, Iu., Choupina, A. (2021). Bioinformatics: new tools and applications in life science and personalized medicine. *Applied Microbiology and Biotechnology*, 105: 937–951.
2. Shajii, A., Numanagić, I., Leighton, A. T. et al. (2021). A Python-based programming language for high-performance computational genomics. *Nat Biotechnol*, 39: 1062–1064.
3. Kunzmann, P., Müller, T. D., Greil, M. et al. (2023). Biotite: new tools for a versatile Python bioinformatics library. *BMC Bioinformatics*, 24(236). DOI: https://doi.org/10.1186/s12859-023-05345-6.
4. Cock, P., Antao, T., Chang, J. T. et al. (2009). Biopython: Freely available Python tools for computational molecular biology and bioinformatics. *Bioinformatics*, 25(11): 1422–1423. DOI: 10.1093/bioinformatics/btp163.
5. Chapman, B., Chang, J. (2000). Biopython: Python tools for computational biology. *ACM SIGBIO Newsletter*, 20(2): 15–19. DOI: https://doi.org/10.1145/360262.360268.
6. Chapman, B. A., Chang, J. T. (2011). Biopython: an enhanced suite of Python libraries for computational biology. *ACM SIGBIO Newsletter*, 51–52.
7. Druce, M., Cock, P. J. A. (2012). SeqIO: a Biopython module for handling sequence data. *Proceedings of the 12th Python in Science Conference*.
8. Prlić, A. et al. (2019). Biopython in 2019: new developments and an increased focus on collaboration. *BMC Genomics*, 20(3): 1–6.
9. Chang, J., Chapman, B., Friedberg, I., Hamelryck, T., de Hoon, M., Cock, P., ... Wilczynski, B. (2010). Biopython tutorial and cookbook. *Update*, 15–19.
10. Schatz, M. C. (2019). Next-generation sequencing technologies for genomic medicine. *Genome Medicine,* 11(1): 1–3.
11. Talevich, E., Invergo, B. M., Cock, P. J. et al. (2012). Bio.Phylo: A unified toolkit for processing, analyzing and visualizing phylogenetic trees in Biopython. *BMC Bioinformatics*, 13(209). DOI: https://doi.org/10.1186/1471-2105-13-209.
12. Le Cun, Y., Bengio, Y., Hinton, G. (2015). Deep learning. *Nature*, 521(7553): 436–444.
13. Angermueller, C., Pärnamaa, T., Parts, L., Stegle, O. (2016). Deep learning for computational biology. *Molecular Systems Biology*, 12(7), 862.
14. Teytelman, L., Thaney, K., Prlić, A. (2016). Bioinformatics for the masses. *Nature Methods*, 13(1), 29.
15. Chapman, B. A., Chang, J. T. (2011). Biopython: an enhancing tool for bioinformatics. *Methods in Molecular Biology*, 694: 301–320.
16. Diego, M., Martins, P., Santos, L. H., Cardoso de Melo-Minardi, R. (2019). Introducing Programming Skills for Life Science Students. *Biochem Mol Biol Educ*, 47(3): 280–295.
17. Grüning, B. et al. (2018). Biopython, a practical approach to biological computing for life scientists. *GigaScience*, 7(1): 1–8.
18. Cock, P. J. A., Grierson, D. S. (2013). Biopython as a tool for computational molecular biology and bioinformatics education. *Briefings in Bioinformatics*, 14(4): 453–457.