**Title:** Innovative Pedagogy in Bioinformatics: Utilizing Team-Based Learning and cloud-based Interactive Notebooks for Python instruction

**Author names and affiliations:**

Abstract: This study introduces an innovative pedagogical approach in molecular and cellular life sciences education, providing an introduction to Python and complex data objects for students with no prior experience. The hypothesis posits that the use of team-based learning (TBL) and cloud-based interactive notebooks can enhance students’ interest in and understanding of Python and its applications in bioinformatics. TBL, a collaborative learning strategy, and interactive notebooks, a technological tool for executing Python code and visualizing data, were utilized in a course attended by 16 postgraduate biomedical students. The study provides a comprehensive description of the methods and materials used, ranging from course design and implementation to the application of TBL and interactive notebooks. The results, corroborated through quantitative evaluation and informal student feedback, indicated a heightened interest and understanding of Python. Both TBL and interactive notebooks received high ratings, suggesting their effectiveness and value in engaging students in Python instruction. While students recognized the necessity for continued practice and learning beyond the scope of this short course, this acknowledgment reflects their engagement and commitment to mastering Python. It underscores the importance of lifelong learning in the ever-evolving field of bioinformatics. The study emphasizes the potential transformative impact of TBL and cloud-based interactive notebooks in education in the field of bioinformatics.

**Keywords:**Bioinformatics Education, Python Instruction, Team-Based Learning (TBL), Interactive Notebooks, Biomedical Students, Complex data objects

**Introduction:**

The advent of high-throughput sequencing technologies has revolutionized the field of molecular and cellular life sciences, leading to an explosion of data. This surge in data has necessitated the use of specialized tools for its analysis. Python, a versatile programming language known for its simplicity and powerful libraries, has emerged as a widely used tool in bioinformatics. However, effectively teaching Python and its applications in bioinformatics to biomedical students, who often lack a background in computer science, poses a significant challenge.

In response to this challenge, educators have explored various pedagogical approaches. Among these, team-based learning (TBL), a collaborative learning strategy that encourages active learning and critical thinking, has shown promise. Studies have demonstrated that both online and in-class TBL can lead to comparable academic outcomes and even surpass traditional teaching methods in terms of student performance and perceptions.

Another innovative tool that has gained popularity in bioinformatics education is the use of interactive notebooks. These digital notebooks, which allow for the execution of Python code and visualization of data in real-time, have been successfully used in various settings, including Clinical Bioinformatics.

This paper presents an innovative pedagogical approach that integrates TBL and interactive notebooks into Python instruction for bioinformatics. To overcome the technical challenges often associated with interactive notebooks, we hosted them on a cloud-based platform, GitHub, and linked them to Google Colaboratory, an online coding environment based on Jupyter notebooks. This approach has not only eliminated potential technical obstacles but also promoted equal access to technology among students, extending the learning environment beyond the classroom. Our hypothesis is that the combined use of TBL and interactive notebooks can enhance students’ interest and understanding of Python and its bioinformatics applications. This hypothesis is based on the premise that active learning strategies like TBL and hands-on tools like interactive notebooks can make complex concepts more accessible and engaging.

In this study, we detail the implementation of this approach in a course delivered to 16 postgraduate biomedical students. We present the results of a quantitative evaluation and student feedback, which suggest that TBL and interactive notebooks are effective tools for teaching Python to biomedical students. This approach not only bridges the gap between biology and informatics but also fosters an environment that promotes active learning and critical thinking. The study further explores the potential of this pedagogical approach to transform bioinformatics education and pave the way for future research in this area.

**Materials and Methods:**

Course Design and Participants

The course, “Introduction to Python for Bioinformatics”, targeted postgraduate biomedical students with minimal programming experience. It was part of the Bioinformatics in Health Science course at the School of Medicine, University of Minho, Portugal. The session had 16 postgraduate students (12 female, 4 male) from biochemistry, biology, or biomedicine backgrounds. All were enrolled in master’s or doctoral programs.

Pre-Class Materials

Students received slides and links introducing Python, its relevance for bioinformatics and an overview of Anaconda, Jupyter, and Google Colaboratory. An interactive notebook, hosted on GitHub and linked to Google Colaboratory, provided a hands-on Python introduction. This setup allowed code interaction from any device without installations. Students also received a brief explanation of Team-Based Learning (TBL) and how to access pre-class materials.

Team-Based Learning (TBL)

TBL was the primary instructional strategy. Students were divided into teams of four. They prepared for class by exploring pre-class materials. They had 30 min to solve the Individual Readiness Assurance Test (IRA), consisting of 22 multiple-choice questions on Python. The same test was used as the Team Readiness Assurance Test (TRA), which took 1 hour and 30 minutes. Teams solved test problems using Python interactive notebooks with internet and consultation materials access. The instructor provided guidance and feedback. This was followed by a 15-minute clarification session and a 1-hour application case session. Interactive notebooks, hosted on GitHub and linked to Google Colaboratory, were used for Python code execution and data visualization. Each notebook contained text, images, Python code, and output. Students were encouraged to experiment with the code and observe real-time changes. A total of 3 notebooks were developed, including one introductory notebook for python language fundamentals and 2 additional interactive notebooks were developed for the cases of application of python to bioinformatics.

Student Feedback

After the session, an optional and anonymous survey was provided through Microsoft Forms to gather student ratings of teaching effectiveness. The form had 7 items, selected and slightly altered from the OIRA Item Bank, rated on a 5-point agreement scale. Statements included “My interest in Python has increased”, “The use of Team Based Learning (TBL) enriched my learning experience in this class”, and “Interactive notebooks were a valuable part of this course”. Survey data were analyzed using descriptive statistics. Student feedback was also collected informally throughout the course.

Access to materials and resources

All interactive python notebooks were made available, these include: Python Introduction Notebook: Link; Applications Notebook 1: Link; Applications Notebook 2: Link. For educators interested in adopting a similar approach, a guide on how to create and deploy interactive notebooks on the cloud is provided: Instructions for Teachers: Link.

**Results:**

Interactive Notebooks Structure and Content

The design of the interactive notebooks aimed to provide comprehensive yet accessible content, striking a balance between covering a broad range of topics and presenting information clearly.

Three notebooks were prepared: an introductory Python notebook for pre-class material and use during the TRA, and two additional notebooks illustrating real-world applications of Python for bioinformatics analysis, provided during the final application case session.

The introductory notebook covered various aspects of Python programming relevant to bioinformatics applications. It started with Python fundamentals, followed by sections on variables and print statements, basic arithmetic operations, conditional statements, function definitions and calls, list manipulation, dictionaries, loops, and module imports. The first application notebook also covered different aspects of Python programming for bioinformatics applications. It began with setting up the environment and importing basic libraries. It then delved into data structures for bioinformatics, basic data structures, the DataFrame data structure from the Pandas library, and how other libraries can be used with DataFrames. The notebook concluded with a section on reading and writing data in different formats using Pandas functions. The second application notebook introduced the Annotated Data (AnnData) data structure in a section titled “From DataFrame to AnnData”. This section guided students on how to convert a pandas DataFrame into an AnnData object, which facilitates efficient data storage, annotation, and analysis. Students were encouraged to appreciate the benefits of integrating multiple dimensions of information (numerical data of expression, laboratory or clinical metadata, gene information, and other structured or unstructured information) into a single complex data object that enables integrative visualization and statistical analysis. The subsequent section focused on data analysis, demonstrating the use of Scanpy for various tasks on the AnnData object, such as plotting, statistical tests, feature selection, and dimensionality reduction. The notebook concluded with a section titled “Into the multi-ome”, which introduced the Muon library. This library supports the analysis of multiomics data using the MuData structure, acquainting students with more advanced bioinformatics analysis techniques.

In summary, the notebooks were designed to provide a comprehensive introduction to Python for bioinformatics, from basic programming concepts to real-world applications, ensuring the content was both wide-ranging and understandable (Figure 1). The three notebooks served as pre-class material, in-class resources, and materials for the final application case session. They covered Python fundamentals, data structures, and libraries essential for bioinformatics, and demonstrated their application in real-world bioinformatics scenarios.

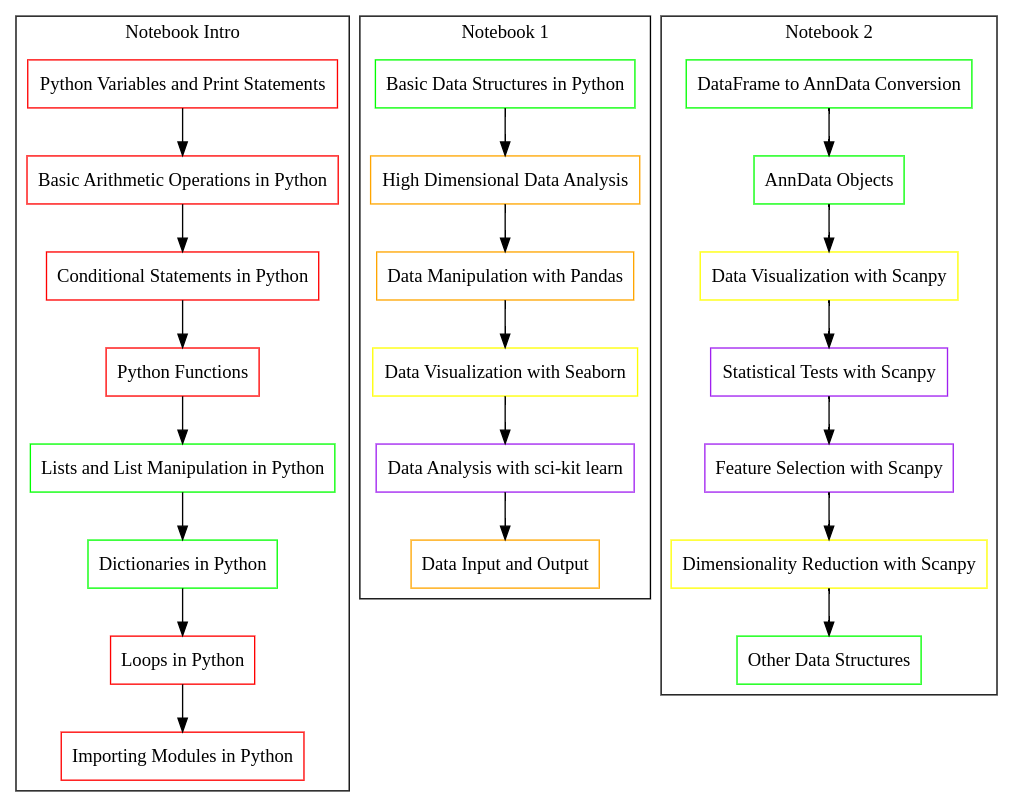


Figure 1: illustrates the content flow and interactions among three Python notebooks designed for the class. Each notebook focuses on different aspects of Python and data science, with shared colors indicating interconnected topics. The color legend is as follows: Red: Python Programming; Green: Data Structures; Orange: Data Manipulation; Yellow: Data Visualization; Purple: Data Analysis.

Implementation and Impact of Cloud-Hosted Interactive Python Notebooks

Interactive Python notebooks have emerged as a powerful tool in bioinformatics education, promoting hands-on learning and real-time engagement. However, their integration into the classroom often faces technical challenges, ranging from hardware limitations to software conflicts. These issues can create uncertainty and discourage the use of live coding in teaching sessions.

To address these challenges, we hosted the course’s interactive Python notebooks on GitHub and linked them directly to Google Colaboratory. This approach created a seamless and accessible learning environment, free from the technical difficulties associated with local computer environments. The cloud-based platforms, freely available for academic and research purposes, facilitated an interactive learning experience that deepened students’ understanding of Python and its applications in bioinformatics.

Students engaged successfully with the notebooks both at home and in the classroom, executing the provided code and experimenting with modifications. This hands-on approach reinforced the concepts presented in the notebooks. Furthermore, the cloud-based approach ensured that all students, regardless of their laptop type or operating system, could access, read, and run the code without any difficulties.

This strategy not only eliminated potential technical obstacles but also promoted equal access to the computational resources among students. It extended the learning environment beyond the physical classroom, enabling students to access the notebooks on any device. The implementation was highly successful, with no reported technical difficulties, indicating the robustness and reliability of this approach. The structure and content of the notebooks were well-received, demonstrating the effectiveness of this method in conveying complex bioinformatics concepts and techniques.

In conclusion, the use of cloud-hosted interactive Python notebooks for bioinformatics instruction was a successful strategy. It provided a comprehensive, hands-on, and accessible learning experience, enhancing students’ understanding of Python and its real-world applications in bioinformatics, and promoting equal access to technology.

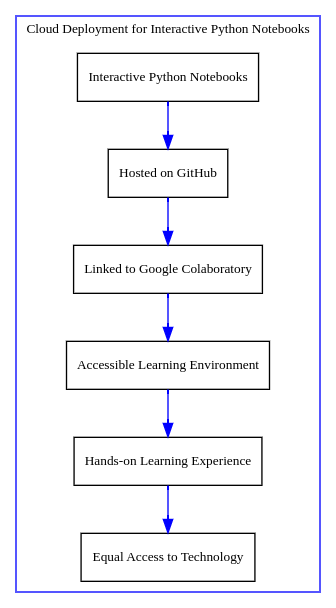


Figure 2. Cloud Deployment for Interactive Python Notebooks. This flowchart illustrates the process of deploying interactive Python notebooks for bioinformatics education in the cloud. The process begins with the creation of the notebooks, which are then hosted on GitHub (Node B). These notebooks are linked to Google Colaboratory (Node C), creating an accessible learning environment (Node D). This environment facilitates a hands-on learning experience (Node E), promoting equal access to technology among students (Node F). The arrows represent the flow from one step to the next, demonstrating the seamless integration of these steps in the deployment process. The use of cloud-hosted interactive Python notebooks was found to be a successful strategy in enhancing students’ understanding of Python and its applications in bioinformatics.

Implementation of Team-Based Learning

In preparation for the Team-Based Learning (TBL) session, students were given access to various resources 48 hours in advance. These included slides and links introducing Python, its relevance in bioinformatics and Google Colaboratory. The Python introductory notebook was made accessible, and students also received an explanation of TBL and guidance on accessing all pre-class materials.

On the day of the activity, TBL served as the primary instructional strategy. Students, divided into teams of four, had 30 minutes to solve the Individual Readiness Assurance Test (IRA), comprising 22 multiple-choice questions on Python (Figure 3). The same test was used as the Team Readiness Assurance Test (TRA), which took 1 hour and 30 minutes. Teams solved test problems using Python interactive notebooks, with access to the internet and consultation materials. The instructors provided guidance and feedback while encouraging that the teams were independent in solving the problems.

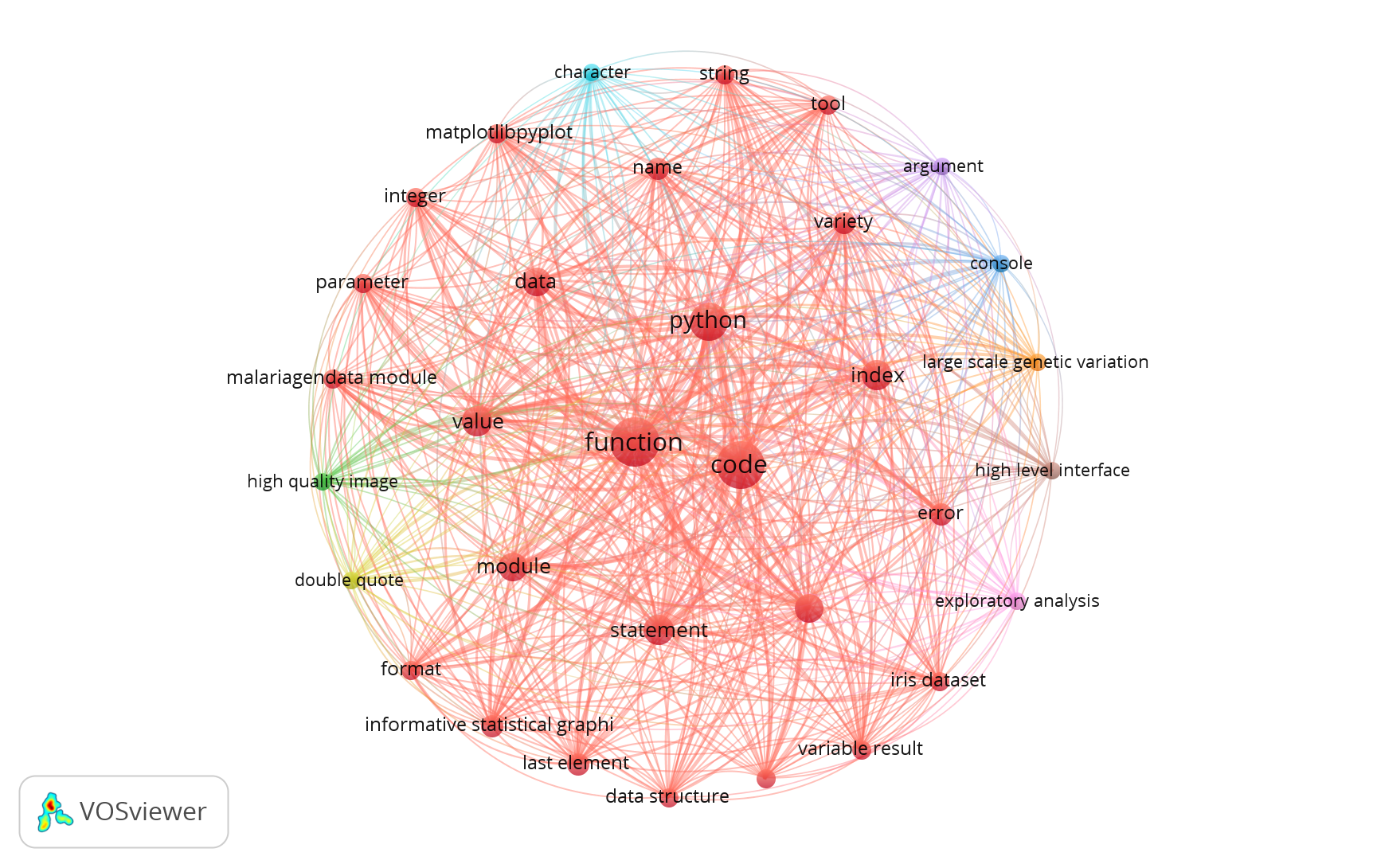


Figure 3: Co-occurrence Map of MCQ Questions used in TBL. This figure presents a co-occurrence map generated by the VOSviewer software, based on the text analysis of the 22 questions used in the Team-Based Learning (TBL) strategy. The map visualizes the relationships among key terms extracted from the questions. Nodes: Each node represents a distinct term identified in the text analysis. The size of a node corresponds to the term’s frequency in the analyzed text. Lines: The lines connecting the nodes represent the co-occurrence of terms within the text..

This co-occurrence map provides a visual representation of the interconnectedness of the key concepts within the TBL strategy questions, offering insights into the underlying themes and patterns.

The IRA was not evaluated to prevent disengagement, and students were aware that it was not being used for grading. The value of the IRA was explained to students as a tool for self-identifying individual limitations in the topic before moving to teamwork. Teamwork was dynamic, with students actively discussing explanations and rationales for specific answers, including consulting material and testing code in the interactive notebooks. Informal feedback collected by tutors during teamwork and results from the clarification session indicated that all teams were able to reach the most correct answers in all 22 questions and justify their choices using accurate knowledge and correct rationales. This was followed by a 1-hour application case session where the teams were guided to explore the two additional notebooks and explore the real-world applications.

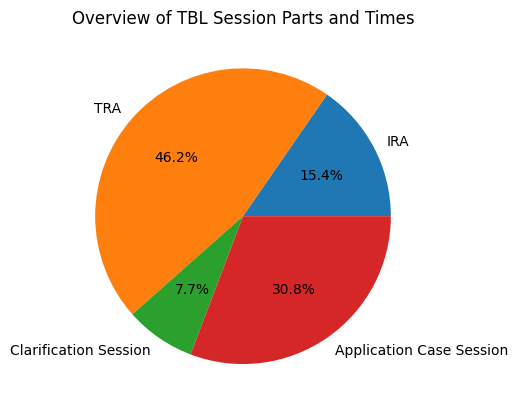


Figure 3.Overview of Team-Based Learning (TBL) Session Parts and Times. This pie chart provides a visual representation of the distribution of time spent on different activities during a TBL session. The activities include the Individual Readiness Assurance Test (IRA), the Team Readiness Assurance Test (TRA), a clarification session, and an application case session. The durations of these activities are represented as percentages of the total session time. The IRA, TRA, clarification session, and application case session account for 30 minutes (15.4%), 90 minutes (46.2%), 15 minutes (7.7%), and 60 minutes (30.8%) of the total session time, respectively. This figure illustrates the significant emphasis placed on the TRA in the TBL session, highlighting the importance of teamwork in this learning strategy.

Student Feedback

Following the session, an optional and anonymous survey was administered via Microsoft Forms to collect student ratings on teaching effectiveness. The survey consisted of seven items, which were chosen and slightly modified from the OIRA Item Bank, and were rated on a 5-point agreement scale.

The average ratings for the statements indicated a positive shift in the students’ interest in Python, which increased to an average rating of 4.00. There was also an increase in their interest in using Python in Bioinformatics analysis, as shown by an average rating of 4.25. The students reported gaining an understanding of major concepts related to Python, reflected by an average rating of 3.69. However, the development of skills necessary for researchers in this field received a moderate average rating of 3.13, suggesting an area for potential improvement. The use of the computer was found to enrich the students’ learning experience in this class, as indicated by an average rating of 4.19. The use of Team-Based Learning (TBL) significantly enriched the students’ learning experience in this class, as evidenced by a high average rating of 4.56. The students also found that interactive notebooks were a valuable part of this course, as reflected by an average rating of 4.06.

Overall, these ratings provide a quantitative measure of the students’ perceptions and experiences during the session. They indicate a positive shift in students’ interest in Python, a good understanding of Python’s major concepts, and a strong positive impact of TBL and the use of interactive notebooks on the learning experience. However, the rating for the development of necessary skills for researchers in the field was moderate, suggesting an area for potential improvement.

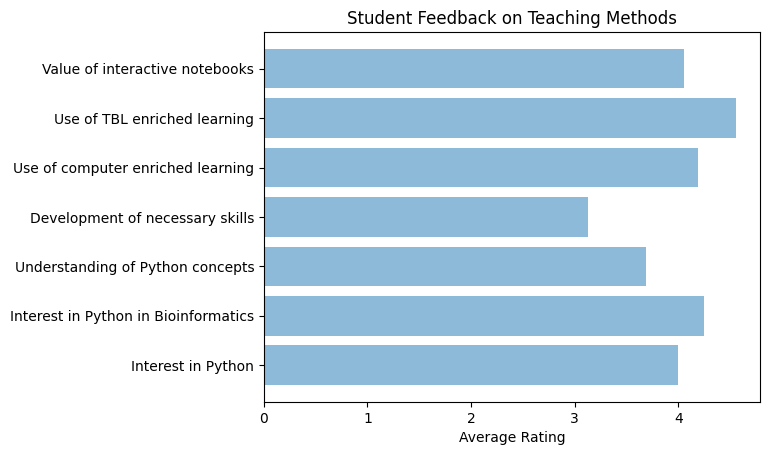


Figure 4. Student Feedback on Teaching Methods. This horizontal bar chart represents the average ratings of student feedback on various teaching methods. The y-axis lists the seven statements that were rated, including ‘Interest in Python’, ‘Interest in Python in Bioinformatics’, ‘Understanding of Python concepts’, ‘Development of necessary skills’, ‘Use of computer enriched learning’, ‘Use of TBL enriched learning’, and ‘Value of interactive notebooks’. The x-axis represents the average rating for each statement on a 5-point agreement scale. The length of each bar corresponds to the average rating received for each statement. The chart provides a visual representation of the students’ perceptions and experiences during the session.

**Discussion:**

The integration of Team-Based Learning (TBL) and cloud-hosted interactive Python notebooks in bioinformatics education, as explored in this study, has demonstrated substantial potential in enhancing the learning experience of postgraduate biomedical students. This innovative pedagogical approach has successfully bridged the gap between biology and informatics, fostering an equitable environment that promotes active learning and critical thinking.

The use of TBL, a collaborative learning strategy, has been pivotal in encouraging active participation among students. The structure of the TBL sessions, which included Individual and Team Readiness Assurance Tests, facilitated engagement with the material both individually and as part of a team. This dual engagement, coupled with face-to-face interaction among students, likely fostered a deeper understanding of Python and its applications in bioinformatics. The high ratings received for TBL in the student feedback underscore its effectiveness in enriching the learning experience.

The implementation of cloud-hosted interactive Python notebooks has effectively addressed the technical challenges often associated with bioinformatics education. By hosting the notebooks on GitHub and linking them to Google Colaboratory, we created an accessible and seamless learning environment. This approach not only eliminated potential technical obstacles but also promoted equal access to computational resources among students, extending the learning environment beyond the classroom. The positive shift in students’ interest in Python, as indicated by the student feedback, could attest to the success of this strategy.

The moderate rating received for the development of necessary skills for researchers in the field is not surprising. Given the short duration of the course and the fact that the students had no prior experience with Python, it is understandable that they may not yet feel fully confident in their newly acquired skills. Acquiring proficiency in Python, like any other language, requires time and practice. Therefore, this rating could be seen not as a shortcoming of the course, but rather as an indication of the students’ recognition of the need for further practice and learning beyond this short course. Future iterations of the course could consider incorporating more advanced topics or providing additional resources to further enhance skill development. Additionally, continued support and resources for self-study and practice could be provided to help students continue their learning journey and build confidence in their Python skills.

The students’ discussion and informal feedback provided valuable insights into their views on the capacity of AI to generate Python code and the role of AI as an assistant rather than a replacement for the programmer. It was highly consensual among the students that the bioinformatician should always understand the code used, as this is the only way to ensure accountability as scientists.

In conclusion, the combined use of TBL and interactive Python notebooks has proven to be a successful strategy for introducing biomedical postgraduate students to Python. It has enhanced students’ interest and understanding of Python and its applications in bioinformatics, thereby contributing to the advancement of bioinformatics education. This study serves as a valuable reference for educators seeking to adopt similar approaches in their courses and opens up new avenues for future research in this area. The potential of this pedagogical approach extends beyond bioinformatics and Python instruction, suggesting its applicability in other areas of computational biology and data science education. Further research is needed to explore these possibilities and to continuously improve upon these types of innovative teaching methodologies.

**Conclusions:**

This study has demonstrated the effectiveness of an innovative pedagogical approach that integrates Team-Based Learning (TBL) and cloud-hosted interactive Python notebooks in bioinformatics education. The approach has successfully enhanced the learning experience of postgraduate biomedical students, fostering an environment that promotes active learning and critical thinking.

The use of TBL has proven instrumental in encouraging active participation among students, facilitating a deeper understanding of Python and its applications in bioinformatics. The implementation of cloud-hosted interactive Python notebooks has effectively addressed the technical challenges often associated with bioinformatics education, promoting equal access to technology among students.

While students recognized the need for further practice and learning beyond this short course, the positive shift in their interest in Python and the high ratings received for TBL and the use of interactive notebooks underscore the success of this strategy.

The study concludes that the combined use of TBL and interactive Python notebooks is a successful strategy for introducing biomedical postgraduate students to Python. Further research is needed to continuously improve upon these types of innovative teaching methodologies. This study serves as a resource for educators seeking to adopt similar approaches in their courses.