**Enhancing Bioinformatics Education: Integrating Team-Based Learning and Cloud-Based Interactive Notebooks for Introducing Python Coding**

**Abstract**

This study introduces an innovative pedagogical approach in higher education, aimed at providing an engaging introduction to Python and data objects for students with no prior experience. The hypothesis posits that team-based learning (TBL) and cloud-based interactive notebooks can enhance students’ interest in Python and its applications in bioinformatics. TBL, a collaborative learning strategy, and interactive notebooks, a technological tool for executing code and visualizing data, were utilized in two courses attended by 28 postgraduate biomedical students. The study describes the methods and materials used, from course design and implementation to the application of TBL and interactive notebooks. The results, supported by quantitative measurement of student feedback, indicated a heightened interest in 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 beyond the scope of these courses, this acknowledgment reflects their engagement and commitment to continue learning Python. This study highlights the effectiveness of the combination of team-based learning (TBL) and cloud-based interactive notebooks in enhancing students' interest in Python within the context of bioinformatics education. Additionally, the approach has potential applicability for other related topics in the life sciences and data science education.

**Highlights**

* TBL and cloud-hosted programming environments are effective tools for bioinformatics education
* The use of TBL encourages activate participation of students
* Interactive notebooks spark interest in programming and are highly valued by students
* Cloud-hosted notebooks eliminate technical challenges often found in bioinformatics education

**Keywords**

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

**Introduction**

The integration of Python programming into biomedical education has become imperative due to the growing importance of computational approaches in life sciences1,2. High-throughput research technologies generate massive amounts of data, requiring specialized tools for analysis. Python, valued for its user-friendliness and extensive libraries, has become the go-to language in bioinformatics2.

Teaching Python to students with limited programming experience presents a challenge. Effective strategies involve introducing core concepts, connecting them to biological problems, and emphasizing practical applications2,3. Tools like Interactive Python notebooks, enabling real-time code execution and data visualization, and flipped classroom approaches have shown promise in enhancing the learning experience1,4. Recent adaptations for online formats, including those used during the COVID-19 pandemic, demonstrated positive outcomes by fostering peer learning and support4.

However, a key challenge in flipped classrooms and online education lies in student disengagement, often leading to high dropout rates5. Key factors contributing to this issue include students being easily distracted, lacking self-regulation skills, and feeling isolated6,7. The absence of face-to-face interaction and limited social presence exacerbate these problems8. Addressing these issues requires exploring alternative pedagogical approaches.

This paper investigates the novel integration of Team-Based Learning (TBL) with cloud-based interactive notebooks in Python instruction of biomedical students. TBL is a collaborative strategy promoting active learning and critical thinking9. Its structured framework promotes student accountability and engagement, consistently demonstrating positive outcomes in both online and in-class settings 10–12.

We hypothesize that the combined use of TBL and interactive notebooks can enhance student engagement and interest in Python and its bioinformatics applications. We posit that active learning strategies, coupled with hands-on tools like interactive notebooks, can make complex concepts more accessible and engaging.

This study explores an innovative approach to address the critical challenge of student disengagement in online and flipped classroom environments for Python programming in bioinformatics. By evaluating the effectiveness of TBL and interactive notebooks, we contribute valuable insights into improving student engagement in this vital field.

**Materials and Methods**

Course Participants

The first 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.

The second course was taught within the “Single-Cell Genomics For Beginners” workshop, offered to postgraduate students from all backgrounds interested in single-cell RNA sequencing technologies. 12 students (10 female, 2 male) participated in this course, hosted at the School of Medicine, University of Minho, Portugal.

The structure and contents of the teaching sessions were the same during both courses, with only minimal schedule changes due to practical reasons.

Pre-Class Materials

Students received slides and links introducing Python, its relevance for bioinformatics and an overview of Anaconda13, Jupyter notebooks14, and Google Colaboratory15. An interactive notebook, hosted on GitHub16 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. While our choice included GitHub and Google Colab, there are several similar alternatives for online repositories (GitLab17, Bitbucket18, SourceForge19, Launchpad20, Figshare21) and coding environments (Kaggle22, CoCalc23, Paperspace24).

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 minutes to solve the Individual Readiness Assurance Test (IRA), consisting of 22 multiple-choice questions on Python (Supplementary Data 1). 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. An additional notebook was made available as post-class material for self-study.

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

The interactive python notebooks (Python Introduction Notebook, Applications Notebooks 1 & 2, Additional Self-study Notebook) were made available through a Github repository (<https://github.com/Leo-GG/Innovative-Pedagogy-in-Bioinformatics>).

**Results**

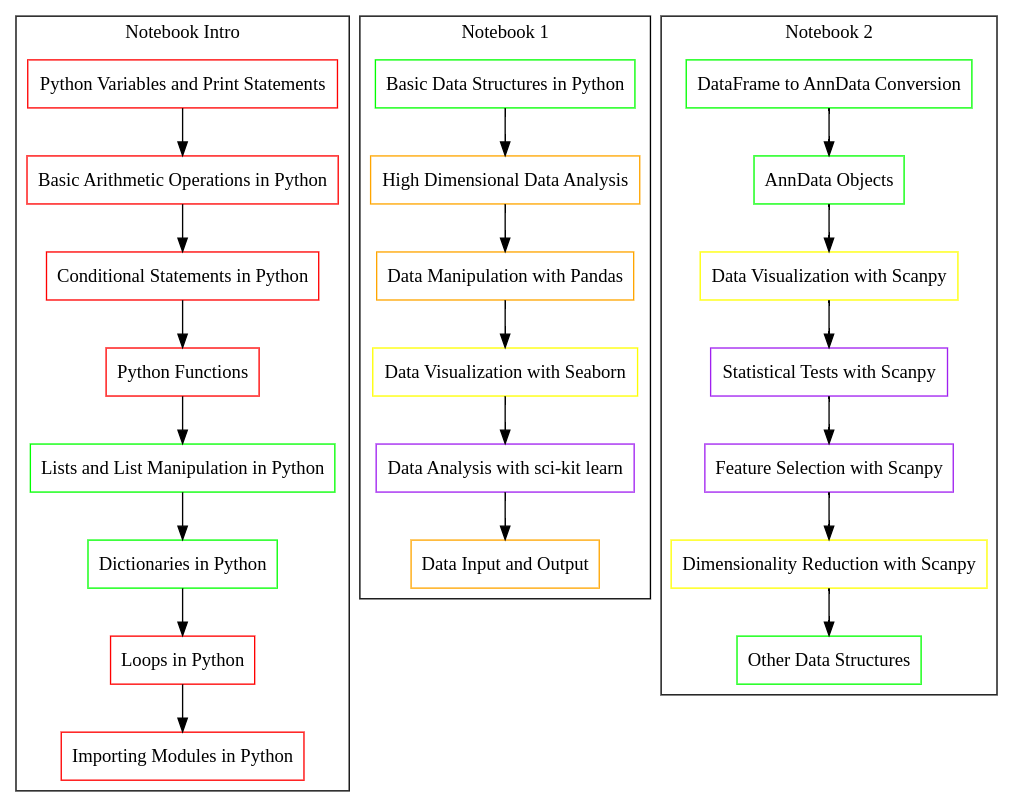
Interactive Notebooks Structure and Content

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.

Four notebooks were prepared: an introductory Python notebook for pre-class material and use during the TRA, two additional notebooks illustrating real-world applications of Python for bioinformatics analysis, provided during the final application case session, and a self-study notebook with exercises based on a practical example.

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 library25, 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 structure26 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 Scanpy27 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 library28. This library supports the analysis of multiomics data using the MuData structure28, acquainting students with more advanced bioinformatics analysis techniques (Figure 1).

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.** 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

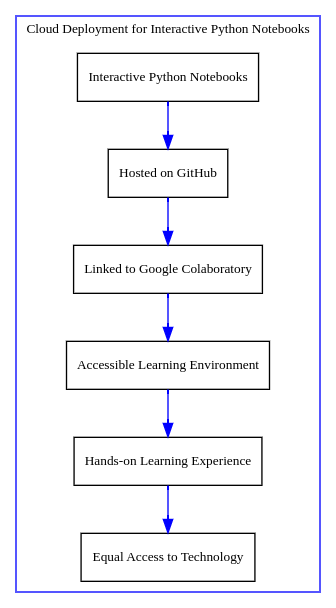
Interactive Python notebooks have emerged as a powerful tool in bioinformatics education, promoting hands-on learning and real-time engagement. However, their integration into the classroom often faces technical challenges, ranging from hardware limitations to software conflicts. These issues can create 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 (Figure 2).

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 hardware or operating system they had available, could access, read, and run the code without any difficulties.

This strategy not only eliminated potential technical obstacles but also promoted equal access to computational resources among students. It extended the learning environment beyond the physical classroom, enabling students to access the Jupyter 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.

We concluded that 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.

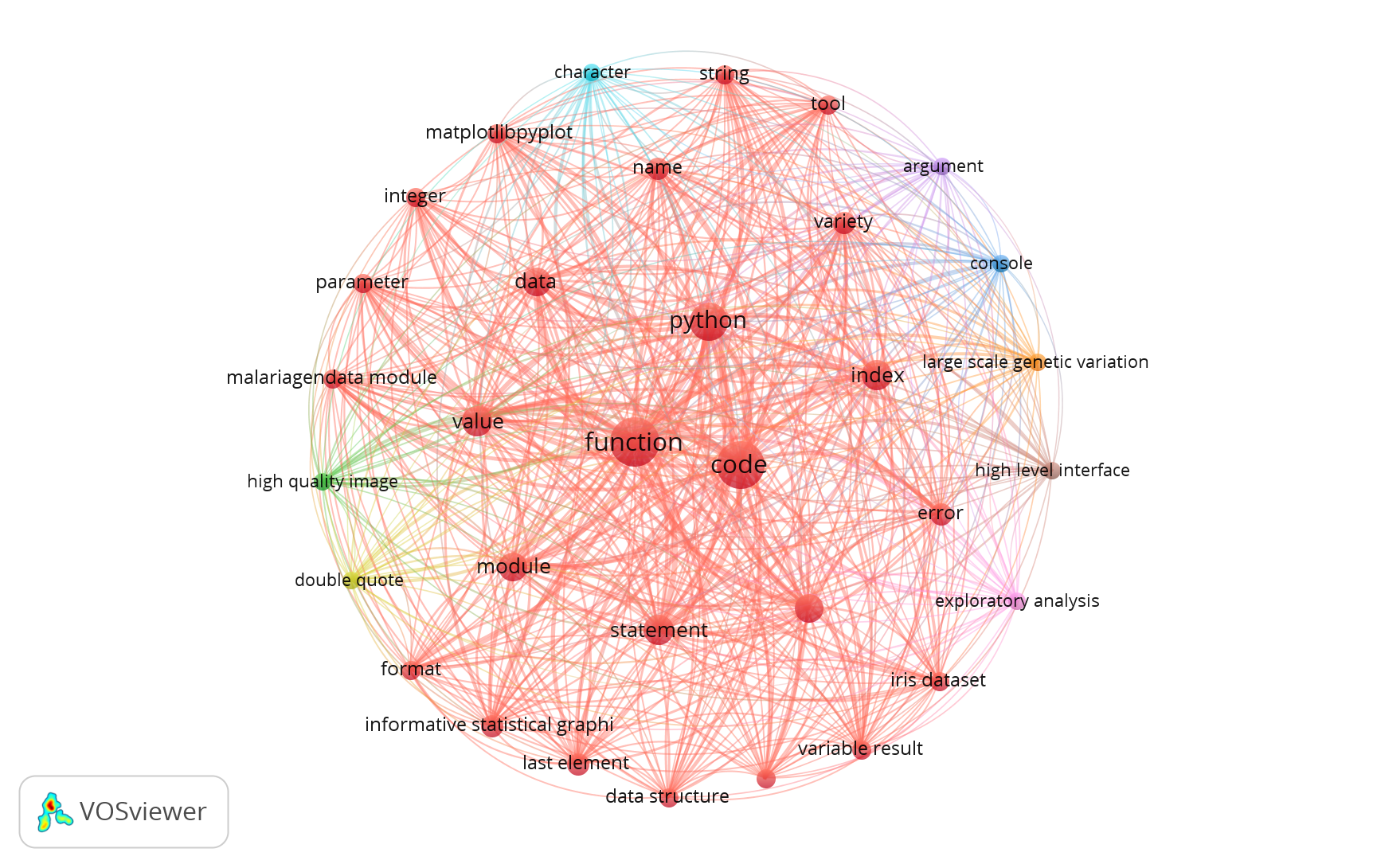


**Figure 2.** Cloud Deployment for Interactive Python Notebooks. This flowchart illustrates the process of deploying interactive Python notebooks for bioinformatics education in the cloud. The process begins with the creation of the notebooks, which are then hosted on GitHub (Node B). These notebooks 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

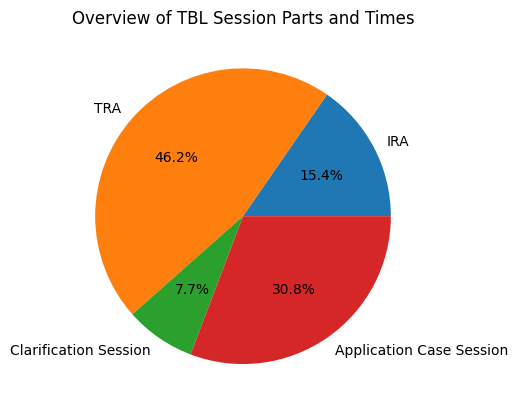
In preparation for the Team-Based Learning (TBL) session, students were given access to various resources 48 hours in advance. These included slides and links introducing Python, its relevance in bioinformatics and Google Colaboratory. The Python introductory notebook was made accessible, and students 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, Supplementary Data 1). 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 4).



**Figure 4.** 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 (Supplementary Data 2).

The average ratings across both courses (N=28 students) 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 4.07. The students reported gaining an understanding of major concepts related to Python, reflected by an average rating of 3.82. However, the development of skills necessary for researchers in this field received a moderate average rating of 3.39, 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 4.21. The use of Team-Based Learning (TBL) significantly enriched the students’ learning experience in this class, as evidenced by a high average rating of 4.36. The students also found that interactive notebooks were a valuable part of this course, as reflected by an average rating of 4.14 (Figure 5, Supplementary Data 2). We did not appreciate any significant differences between the answers on the two courses (p>0.05, two-sided Wilcoxon rank-sum test).

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.

A graph of a student

Description automatically generated with medium confidence

**Figure 5.** Student Feedback on Teaching Methods. This horizontal bar chart represents the average ratings of student feedback on various teaching methods on each of the two courses and on the whole data. The y-axis lists the seven statements that were rated, including ‘Increased Interest in Python’, ‘Increased Interest in Python in Bioinformatics’, ‘Understood Python concepts’, ‘Developed necessary skills’, ‘Use of computer enriched my learning’, ‘Use of TBL enriched my 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, and the error bars indicate the 95% confidence intervals. 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 fostered an equitable environment that promotes active learning and critical thinking.

The use of Team-Based Learning (TBL), a collaborative learning strategy with a well-defined structure, has been instrumental in encouraging active participation among students. The TBL sessions' structure, including Individual and Team Readiness Assurance Tests (IRATs and TRATs), facilitated engagement with the material on two levels: individually and collaboratively. This combined individual and team responsibility, coupled with face-to-face student interaction, likely fostered a deeper understanding of Python and its applications in bioinformatics, as previous research suggests (Parmelee, 2012, Shen et al., 2024; Silva et al., 2021; Mott & Peuker, 2015). The high ratings received for TBL in student feedback further underscore its positive impact on enriching the learning experience.

The implementation of cloud-hosted interactive Python notebooks has effectively addressed the technical challenges often associated with bioinformatics education when using live coding sessions. 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 summary, 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. This 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.

**Data and code availability**

The original data and code used in this study is publicly available at <https://github.com/Leo-GG/Innovative-Pedagogy-in-Bioinformatics>. This repository contains also the course materials that were used to implement the proposed teaching strategy.

**References**

1. Chapman, B. E. & Irwin, J. Python as a first programming language for biomedical scientists. in *Scipy* 12–17 (2015).

2. Mariano, D., Martins, P., Helene Santos, L. & de Melo-Minardi, R. C. Introducing programming skills for life science students. *Biochemistry and Molecular Biology Education* **47**, 288–295 (2019).

3. Ekmekci, B., McAnany, C. E. & Mura, C. An introduction to programming for bioscientists: a Python-based primer. *PLoS computational biology* **12**, e1004867 (2016).

4. Vladis, N. A. & Coleman, B. I. Moving a flipped class online to teach Python to biomedical Ph. D. students during COVID-19 and beyond. *Journal of Microbiology & Biology Education* **22**, 10–1128 (2021).

5. Purarjomandlangrudi, A., Chen, D. & Nguyen, A. Investigating the drivers of student interaction and engagement in online courses: A study of state-of-the-art. *Informatics in Education* **15**, 269–286 (2016).

6. Hew, K. F. & Huang, W. Promoting engagement in online learning beyond COVID-19: Possible strategies and directions for future research. *Future in Educational Research* **1**, 27–49 (2023).

7. Akar, S. G. M. Why do students disengage from online courses? *The Internet and Higher Education* **62**, 100948 (2024).

8. Khan, A., Egbue, O., Palkie, B. & Madden, J. Active learning: Engaging students to maximize learning in an online course. *Electronic Journal of e-learning* **15**, pp107-115 (2017).

9. Parmelee, D., Michaelsen, L. K., Cook, S. & Hudes, P. D. Team-based learning: a practical guide: AMEE guide no. 65. *Medical teacher* **34**, e275–e287 (2012).

10. Shen, J., Qi, H., Mei, R. & Sun, C. A comparative study on the effectiveness of online and in-class team-based learning on student performance and perceptions in virtual simulation experiments. *BMC Medical Education* **24**, 135 (2024).

11. Silva, E. C. e, Lino-Neto, T., Ribeiro, E., Rocha, M. & Costa, M. J. Going virtual and going wide: comparing Team-Based Learning in-class versus online and across disciplines. *Education and information technologies* **27**, 2311–2329 (2022).

12. Mott, J. & Peuker, S. Achieving high functioning teams using team based learning in flipped classrooms. in *2015 ASEE annual conference & exposition* 26–143 (2015).

13. Anaconda | The Operating System for AI. *Anaconda* https://www.anaconda.com/, Accessed 03/08/2024.

14. Project Jupyter. https://jupyter.org, Accessed 03/08/2024.

15. Bisong, E. & Bisong, E. Google colaboratory. *Building machine learning and deep learning models on google cloud platform: a comprehensive guide for beginners* 59–64 (2019).

16. GitHub: Let’s build from here · GitHub. https://github.com/, Accessed 03/08/2024.

17. The most-comprehensive AI-powered DevSecOps platform. https://about.gitlab.com/, Accessed 03/08/2024.

18. Atlassian. Bitbucket | Git solution for teams using Jira. *Bitbucket* https://bitbucket.org/product, Accessed 03/08/2024.

19. Compare, Download & Develop Open Source & Business Software - SourceForge. https://sourceforge.net/, Accessed 03/08/2024.

20. Launchpad. *Launchpad* https://launchpad.net/index.html, Accessed 03/08/2024.

21. figshare - credit for all your research. https://figshare.com/, Accessed 03/08/2024.

22. Kaggle: Your Machine Learning and Data Science Community. https://www.kaggle.com/, Accessed 03/08/2024.

23. Collaborative Calculation and Data Science. https://cocalc.com/, Accessed 03/08/2024.

24. NVIDIA H100 for AI & ML Workloads | Cloud GPU Platform | Paperspace. https://www.paperspace.com/, Accessed 03/08/2024.

25. McKinney, W. & Team, P. Pandas-Powerful python data analysis toolkit. *Pandas—Powerful Python Data Analysis Toolkit* **1625**, (2015).

26. Virshup, I., Rybakov, S., Theis, F. J., Angerer, P. & Wolf, F. A. anndata: Annotated data. *BioRxiv* 2021–12 (2021).

27. Wolf, F. A., Angerer, P. & Theis, F. J. SCANPY: large-scale single-cell gene expression data analysis. *Genome biology* **19**, 1–5 (2018).

28. Bredikhin, D., Kats, I. & Stegle, O. MUON: multimodal omics analysis framework. *Genome Biol* **23**, 42 (2022).