Teaching bioinformatics programming in high school: a case report

Helena Lott Costa[1], **Giovana C. F. Maia** [1], **Lucas G. S. Chaves** [1], **Diego Mariano** [1], **Raquel C. de Melo-Minardi** [1]

[1]Laboratório de Bioinformática e Sistemas (LBS)
Universidade Federal de Minas Gerais (UFMG), Belo Horizonte, Brazil

raquelcm@dcc.ufmg.br

Abstract. *Bioinformatics requires professionals with knowledge in computing and biological sciences, but teaching it to young people remains a challenge. This article reports on a programming course focused on bioinformatics for high school students. The pilot project, launched in 2024 in Belo Horizonte, Brazil, aimed to integrate programming and Molecular Biology. Using Inquiry-Based Learning (IBL) and gamification, the course engaged students effectively. Activities were divided into quarterly stages, teaching programming through Scratch and projects involving Molecular Biology, such as DNA transcription. The initiative successfully motivated and engaged students in learning Molecular Biology. We hope that the strategies presented here can be adopted by teachers and help inspire a new generation of bioinformaticians.*

1. Introduction

Bioinformatics is a recent research area that encompasses the use of computer science techniques for analyzing biological data [dos Santos et al. 2023]. As it is a new research area, it lacks professionals capable of dealing with both biological and computational problems. Thus, training qualified professionals is a challenge in education. For example, teaching programming to undergraduate students in biological sciences has been adopted as a strategy for training future bioinformaticians [Mariano et al. 2019]. In addition, the interest in this area among children and adolescents has also been discussed previously [Mariano et al. 2022]. In fact, teaching programming to young people can be useful in fostering critical thinking. However, bringing programming to primary and secondary education is still an open challenge since it requires infrastructure, technologies, and qualified professionals to provide such teaching.

The recent technological advancement fosters significant sociocultural changes that impact the educational sphere through connectivity and the availability of technological devices, establishing a new paradigm in teaching and learning processes [Coll 2013, Nicolete et al. 2021]. This reality is reflected in Brazil with the publication of Law No. 13.415/2017, which amended the National Education Guidelines and Framework Law (LDB), increasing the total course load and reorganizing the curriculum of the new high school, "*Base Nacional Comum Curricular*" (BNCC). The BNCC provides guidelines that guide the development of curricula for public and private schools in Brazil, in addition to formative itineraries [Brasil 2017].

Formative itineraries are created by the educational network itself in the form of disciplines, projects, and workshops that can focus on deepening one of the knowledge areas outlined in the BNCC and technical and professional training (FTP). The main objective of the new high school is to bring educational institutions closer to the realities of students, considering technological advancements and the needs of the current labor market [Brasil 2017]. To assist in creating new itineraries, the government provided a supplement to the BNCC that guides the inclusion of computing in the basic education cycle, determining general and specific competencies and skills to be developed in high school [Brasil 2022].

Thus, the goals set by the Brazilian government for the personalization of education and the integration of students into the labor market can be achieved by teaching programming languages in schools, which aids in teaching Science and Mathematics [Grandell et al. 2006, Bodin 2020]. Despite the notable benefits, choosing a programming language is complex, and it is essential to emphasize the need for a language that is easy to handle, with more intuitive syntax, immediate feedback, and free access, such as the Scratch language [Grandell et al. 2006, Bodin 2020](Grandell, 2006; Bodin, 2020). Moreover, the interaction between programming and Biology, as outlined by the BNCC, can bring students closer to the labor market, given the growing demand for Bioinformatics, motivating students to engage in studies by creating a tangible life goal [Lewitter and Bourne 2011, Marques et al. 2014]. Additionally, including this area in schools enhances the understanding of abstract subjects like genetics and promotes an interdisciplinary approach involving Computer Science, Biology, and Mathematics [Marques et al. 2014, Vasconcelos et al. 2022].

Given Brazil's current educational landscape, implementing the new high school requires differentiated strategies. Teachers must be brought closer to students' realities and interests. Furthermore, the new documents for incorporating computing into basic education must also be considered. In this context, Bioinformatics emerges as an essential area for creating a new formative itinerary. In this work, we present a report on the application of programming teaching for bioinformatics into basic education. Our goal is for this proposal to serve as a tool for deepening the teaching of Molecular Biology through a pilot project in a school by creating a Formative Itinerary that includes programming teaching with projects that stimulate computational thinking and programming skills combined with Molecular Biology content. Moreover, in the offered discipline, teaching strategies will be used to promote student autonomy by introducing Information and Communication Technologies (ICTs) as a study object in Basic Education, achieving the competencies and skills proposed by the BNCC.

1.1. Related works

Although there are few studies on the integration of computing into high school, all emphasize its importance in promoting autonomy, creativity, and logical-mathematical reasoning, as well as interdisciplinary skills [Nascimento 2009, Nicolete et al. 2021, Silva 2017]. The articles highlight the need to create detailed lesson plans, focusing primarily on the Scratch language, to promote student protagonism in an innovative learning environment [Nascimento 2009, Scaico et al. 2013, Silva 2017].

Additionally, [Martins et al. 2017] identifies the Inquiry-Based Learning (IBL) methodology, in which students understand the topics covered through questions they raise based on challenges developed by the teacher. However, this methodology does not address some limitations in teaching programming, such as student engagement. Thus, [França and Reategui 2013] emphasizes that the use of gamification, a methodology that employs game elements to engage students, is necessary in a combined manner to fill the mentioned gap, as this strategy represents a form of motivation for teaching and learning [Silva et al. 2019].

For example, Silva and collaborators proposed a game called proteinGO to teach structural bioinformatics [Silva et al. 2019]. In the game, students were asked to interactively identify electrostatic interactions performed between pairs of amino acids in proteins. The game allowed both students to become interested in structural bioinformatics and researchers to build a manually curated database of intermolecular contacts. Contact detection is a task that can be solved by several computational strategies, such as those based on cutoff distance and those based on the Delaunay triangulation. However, these strategies can introduce false positives or false negatives. Manual curation is a simple but costly task.

Finally, to deepen the understanding of programming logic through the Scratch language, using a combination of IBL and gamification methodologies, the application of concepts in Bioinformatics will be fundamental. Using this area at the end of the course will highlight Molecular Biology and Genetics concepts, as learning these required abstractions [Castro et al. 2020, Martins et al. 2020]. For this, creating practical lesson plans and games that investigate databases, enabling students to construct their own learning, is considered effective [Castro et al. 2020, Martins et al. 2018].

2. Methodology and Results

We began the Formative Itinerary by conducting two classes to introduce computing and programming concepts. Concepts of hardware, software, operating systems, and algorithms were presented. Then, we explained how computers convert human language into binary language. These steps were essential in creating the students' first contact with the content, as most of the basic cycle students have no prior knowledge of programming. Next, the course addressed the concepts of the Scratch programming language. To consolidate what was learned in the theoretical classes, we challenged the students to develop a computer game that was widely known to them. Molecular Biology reviews were conducted to ensure the students' understanding of the concepts. Afterward, the students applied the learned concepts by integrating them with programming in Scratch, creating games and quizzes. Figure 1 presents an overview of the methodology adopted in the course.

2.1. Participants and location of activities

The project began in February 2024, coinciding with the start of the school year at *Colégio Nossa Senhora do Monte Calvário*, a private school located at *Av. do Contorno, 9384 - Barro Preto, Belo Horizonte* (Brazil). A course titled "Program's Coding - The Code of Life" was created in collaboration with the school as part of the formative pathway in the area of Natural Sciences and their technologies, as outlined by the National Common Core Curriculum and in compliance with Law No. 13,415/2017. The class was taught once a week over two terms, lasting a total of 8 months.

In total, 15 students, aged between 15 and 18, enrolled in the course. The classes were held in the school's computer lab, where computers were available for each student

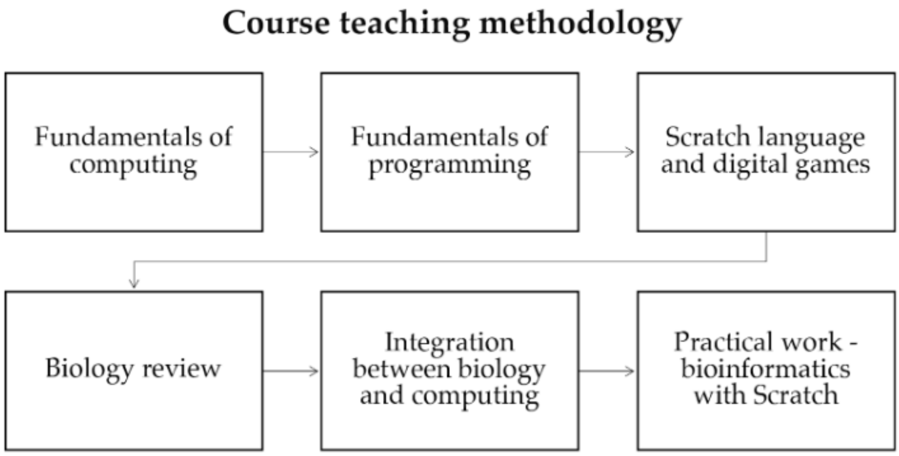


Figure 1. Methodology adopted in the course.

and the teacher, with the latter connected to an overhead projector used for demonstrating activities to the students.

2.2. Quartely schedule development

The classes were divided into two stages over the year, each lasting four months, according to the school calendar. Initially, a questionnaire was provided to assess the students' understanding of programming. Based on the results, the students were then taught basic programming concepts through both practical and theoretical lessons. These concepts included what hardware and software are, operating systems, programming languages, algorithms, and programming logic. It was found that none of the students had prior knowledge of the subject.

To conclude the first stage, and after studying the theoretical part, the students were introduced to the Scratch platform `https://scratch.mit.edu/`, where they learned how to navigate the platform and the programming language by being taught the use of each available block, such as "Events" and "Motion." In total, eight types of block categories were explored in eight 50-minute lessons. The categories studied were motion, looks, sound, events, control, sensing, operators, and variables. Initially, a theoretical explanation of these blocks was given. Then, challenges containing the studied blocks were used for the students to create scripts that command the indicated characters (sprites).

At the end of this stage, the students consolidated their learning by combining all the blocks to create two games. To test and reinforce the knowledge they had acquired, they were required to program a game common to all, where the goal was to create a maze for a mouse to navigate, aiming to collect cheese, called "Cheese Chase." The other game involved the students' creativity in creating a program in Scratch that addressed Molecular Biology. All the projects were submitted to the teacher and the school. The projects were graded in the course according to Table 1.

2.3. Developed games "Cheese Chase"

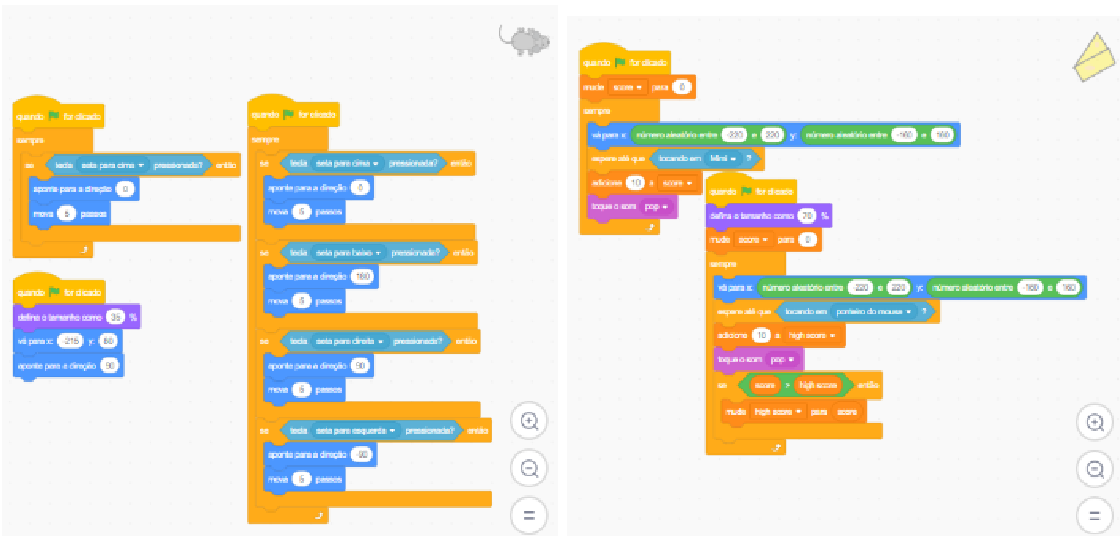
In this section, we describe an example activity that used gamification strategies to teach students the fundamentals of Scratch language. Working in groups, the students devel-

Table 1. Scoring and activities completed by the students, totaling 60 points.

Grade	Activity
18	Conclusion of the challenges related to the blocks studied in each category provided by Scratch.
18	Creation of the ”Cheese Chase” game.
18	Development of a program that addresses topics in Molecular Biology.

oped the ”Cheese Chase” program, a game in which the user controls a mouse to collect cheese. Each cheese collected increases the player’s score. However, according to the rules, the mouse cannot touch the maze, the beetles, or the ghost; if it does, the game ends as a penalty.

As an illustration of the developed project, the work of one group was selected and made available at the link: `https://scratch.mit.edu/projects/998053869`. Figure 2 contains some of the codes developed by the students. The groups did not face difficulties constructing the algorithm for each sprite; however, they encountered significant challenges translating them into Scratch language. As a result, 8 lessons were allocated for creating and submitting the completed game.



(a) (b)

Figure 2. (A) Code developed for controlling the ”Mouse” sprite. (B) Code used for the random appearance of cheese in the maze, along with creating a variable that associates the mouse touching the cheese with an increase in the user’s score.

The biggest challenge encountered by the student groups in coding the mouse was the correct use of the ”if” conditional statement. Regarding the cheese code, the students struggled with using variables to change the game score when the mouse touched the cheese. Three lessons were spent in this stage, including one dedicated to reviewing the theoretical content of variable commands.

Scripts were also developed containing code for the appearance and disappearance of the ghost, which follows the mouse throughout the game. Another script controls the beetles, which move randomly around the space. When touching the mouse, both sprites trigger the appearance of the game-over message. It is important to note that the groups also created a sprite for the maze to facilitate interaction between the mouse and the beetles in the labyrinth.

The students did not struggle coding the beetle and the ”Game Over” sprite. However, they found it challenging to make the ghost sprite move randomly around the space, appearing and disappearing. Despite the challenges, the students were able to complete the project independently.

2.4. Interactions involving Molecular Biology

Finally, students were encouraged to combine the knowledge acquired in game development with expertise in molecular biology and elaborate on a practical project. Since there were different high school grades in the classroom, conducting a review lesson on Molecular Biology was necessary. Immediately afterward, the students were organized into groups to develop a program that addressed the concepts learned in class. They created two question-and-answer games about the DNA transcription process, an interaction that demonstrates to the user what happens at the cellular level when someone experiences tissue damage, relating it to the cellular regeneration process, a memory game that links codons with the corresponding amino acids, and an interaction that illustrates phenotype and genotype.

To illustrate the work developed, two groups were selected, and their projects were made available at the links `https://scratch.mit.edu/projects/1043530530` and `https://scratch.mit.edu/projects/1018839642`. All the projects were conceived and created by the students. They encountered fewer difficulties constructing the script than the ”Cheese Chase” project, requiring four 50-minute lessons to complete the entire process. This indicates that the strategy of practicing algorithm development and script creation beforehand in a well-known game was essential to the success of this particular project.

After completing the activity, the students created a poster to showcase the project developed at *Colégio Nossa Senhora do Monte Calvário*. This action will allow for scientific dissemination among students of all grades at the school, as well as parents and families.

2.5. A brief summary: what to do and what not to do

To facilitate the reproduction of the curriculum in other schools, we elaborate five ”DOs” and ”DON‘Ts” recommendations throughout the teaching process based on our perspective.

2.5.1. Do

1. **Choose the right methodology**: The chosen methodology is key to the success of the curriculum. Strategically taught content promotes a more dynamic learning experience. The use of IBL and gamification methods was fundamental, as IBL sparked students’ curiosity, and gamification reinforced content through competition.

2. **Use a game previously known to the students**: The selection of the game ”Cheese Chase,” which was widely known, was crucial for student engagement, as they felt challenged to develop a program they had played before.
3. **Dedicate a lesson to developing each sprite's script**: For each sprite used in the ”Cheese Chase” game, a slide was created with the rules and guidelines for each character. Each lesson focused on coding one sprite, allowing all groups to work simultaneously providing greater support.
4. **Review Molecular Biology content**: The new teaching format allows students to choose their formative path subjects, resulting in students from different grades attending the same class. This review phase was important for leveling the content across groups.
5. **Divide them into groups**: Allowing students to form groups of up to three was a good strategy, as they could share knowledge with each other, facilitating joint learning.

2.5.2. Don't

1. **Do not let students select their own group members**: While students formed their own groups and completed the scripts satisfactorily, it's important for the teacher to identify those who excel in programming logic and Molecular Biology during the introductory classes. This allows for strategically forming groups that pair students with different skill levels for better results.
2. **Do not wait for students to ask questions**: Teaching programming in basic education is challenging, so the teacher must constantly check on students' activities. While the chosen methodology promotes autonomy, more in-depth guidance from the teacher is needed.
3. **Do not ignore student needs**: Given the novelty of teaching programming combined with Molecular Biology, there will be challenges in the learning process. Listening to students' difficulties and revisiting content is crucial to keeping them motivated.
4. **Do not skip steps**: The development of the curriculum requires care and patience. After reviewing the Molecular Biology content, two introductory lessons should be scheduled: one for the group to create and submit their idea to the teacher and another for developing the algorithm, followed by the final scripting phase.
5. **Do not keep student projects unpublished**: Students feel valued and motivated when their projects are shared. Moreover, showcasing their work can inspire other students to engage with the content and take the course.

3. Conclusion

In this article, we presented a case report on the adoption of programming for bioinformatics in high school. The formative pathway motivated the students to create projects involving Molecular Biology using programming. Developing these projects in Scratch was challenging but satisfying. As the project is still ongoing, we will conduct questionnaires to assess the level of learning in the Scratch programming language. Additionally, we will evaluate whether the integration of programming teaching with Molecular Biology was sufficient to enhance the understanding of biological concepts. Nevertheless, it is necessary to implement the pathway in other schools to assess learning on a larger scale. We believe that the partial results are good indicators that the proposal presented here has great potential in student training and can be adopted by other teachers who seek to carry out different forms of teaching to achieve better autonomous learning for students.

4. Acknowledgements

The authors would like to thank the research funding agencies CAPES, FAPEMIG, and CNPq. Also, to the management and coordination team of *Colégio Nossa Senhora do Monte Calvário*, for providing materials, support, and enabling the execution of the project. To the students for their dedication to the activities carried out. This study was financed in part by the *Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES)* - Finance Code 001.

References

[Bodin 2020] Bodin, A. (2020). *Python in High School: Algorithms and Mathematics*. Exo7.

[Brasil 2017] Brasil (2017). LEI Nº 13.415, DE 16 DE FEVEREIRO DE 2017. `https://www.planalto.gov.br/ccivil_03/_ato2015-2018/2017/lei/l13415.htm`.

[Brasil 2022] Brasil (2022). Base Nacional Comum Curricular. Computação: complemento à BNCC. `http://portal.mec.gov.br/docman/fevereiro-2022-pdf/236791-anexo-ao-parecer-cneceb-n-2-2022-bncc-computacao/file`.

[Castro et al. 2020] Castro, J. A., Santos, M. C., Ramalho, S. A., et al. (2020). Bioinformática como objeto de aprendizagem digital (oad) para o ensino da biologia molecular. *Instituto Federal de Educação, Ciência e Tecnologia Baiano - Campus Governador Mangabeira*.

[Coll 2013] Coll, C. (2013). El currículo escolar en el marco de la nueva ecología del aprendizaje. *Aula de Innovación Educativa*, (219):31–36.

[dos Santos et al. 2023] dos Santos, L. M., Mariano, D., and de Melo-Minardi, R. C. (2023). The impact of artificial intelligence in life sciences through bioinformatics. *Rev. UFMG*, 30:32–59.

[França and Reategui 2013] França, R. M. and Reategui, E. B. (2013). SMILE-BR: aplicação de conceitos de gamificação em um ambiente de aprendizagem baseado em questionamento. *II Congresso Brasileiro de Informática na Educação*.

[Grandell et al. 2006] Grandell, L., PeltomÄki, M., Back, R. J., et al. (2006). Why complicate things? introducing programming in high school using python. *Conferences in Research and Practice in Information Technology*, 52.

[Lewitter and Bourne 2011] Lewitter, F. and Bourne, P. (2011). Teaching bioinformatics at the secondary school level. *Plos Computational Biology*, 7.

[Mariano et al. 2019] Mariano, D. et al. (2019). Introducing programming skills for life science students. *Biochemistry and Molecular Biology Education*, 47(3):288–295.

[Mariano et al. 2022] Mariano, D., Santos, L. H., Meleiro, L. P., de Lima, L. H. F., Martins, L. F., and de Melo-Minardi, R. C. (2022). Using computers to improve biofuel production. *Front. Young Minds*, 10:751195.

[Marques et al. 2014] Marques, I., Almeida, P., Alves, R., et al. (2014). Bioinformatics projects supporting life-sciences learning in high schools. *Plos Computational Biology*, 10.

[Martins et al. 2018] Martins, A. S., Tavares, F., and Fonseca, M. J. (2018). Mining the genome: Using bioinformatics tools in the classroom to support student discovery of genes. *The American Biology Teacher*, 80(8):619–624.

[Martins et al. 2020] Martins, A. S., Tavares, F., Fonseca, M. J., et al. (2020). Bioinformatics-based activities in high school: Fostering students' literacy, interest, and attitudes on gene regulation, genomics and evolution. *Front. Microbiology*, 11.

[Martins et al. 2017] Martins, A. S., Tavares, F., and Lencastre, L. (2017). Integração da bioinformática nos currículos do ensino básico e secundário. *Atlas de Conferência Nacional*.

[Nascimento 2009] Nascimento, J. K. F. (2009). Informática aplicada à educação. *Universidade de Brasília*, page 84.

[Nicolete et al. 2021] Nicolete, P. C., Cristiano, M. A. S., Santos, A. C., et al. (2021). Informática na educação básica pública brasileira: análise sobre sua importância, tendências e desafios. *Educação Temática Digital*, 23(3):794–815.

[Scaico et al. 2013] Scaico, P. D., Azevedo, S., Alencar, Y., et al. (2013). Ensino de programação no ensino médio: Uma abordagem orientada ao design com a linguagem scratch. *Revista Brasileira de Informática na Educação*, 21(2).

[Silva 2017] Silva, J. C. (2017). Ensino de programação para alunos do ensino básico: Um levantamento das pesquisas realizadas no brasil. *TCC (Ciência da Computação) - Universidade Federal da Paraíba (UFPB)*.

[Silva et al. 2019] Silva, M. F. M., Martins, P. M., Mariano, D. C. B., et al. (2019). Proteingo: Motivation, user experience, and learning of molecular interactions in biological complexes. *Entertainment Computing*, 29:31–42.

[Vasconcelos et al. 2022] Vasconcelos, R. A., Sousa, F. B., and Coutinho, T. J. D. (2022). Bioinformática para a educação básica: capacitando docentes para o uso de ferramentas computacionais em sala de aula. *REnCiMa*, 13(4):1–16.