การพัฒนาการคิดเชิงคำนวณของผู้เรียนชั้นมัธยมศึกษาปีที่ 4 ด้วยการปฏิบัติทางชีวสารสนเทศขั้นพื้นฐาน

ณัฐสิทธิ์ นรสิทธิ์[1] พงศ์ประพันธ์ พงษ์โสภณ[1*]
วรรณวิภา วงศ์แสงนาค[2,3] และสันติชัย อนุวรชัย[4]

[1]ภาควิชาการศึกษา คณะศึกษาศาสตร์, [2]ภาควิชาสัตววิทยา คณะวิทยาศาสตร์, [3]ศูนย์โอมิกส์ด้านเกษตร ทรัพยากร
ชีวภาพอาหารและสุขภาพ มหาวิทยาลัยเกษตรศาสตร์ กรุงเทพฯ 10900; [4]กลุ่มสาระการเรียนรู้วิทยาศาสตร์และ
เทคโนโลยี โรงเรียนสาธิตแห่งมหาวิทยาลัยเกษตรศาสตร์ ศูนย์วิจัยและพัฒนาการศึกษา กรุงเทพฯ 10900
[*]E-mail: pongprapan.p@ku.th

รับบทความ: 12 มีนาคม 2566 แก้ไขบทความ: 13 ตุลาคม 2566 ยอมรับตีพิมพ์: 5 ธันวาคม 2566

บทคัดย่อ

วิทยาการคำนวณได้เข้ามามีบทบาทในวงการวิทยาศาสตร์อย่างมาก โดยเฉพาะอย่างยิ่งในการจัดการข้อมูลทางวิทยาศาสตร์ขนาดใหญ่ มีความซับซ้อนสูง ผู้เรียนต้องเตรียมความพร้อมในการเป็นนักวิทยาศาสตร์ในยุคแห่งข้อมูลและเทคโนโลยีโดยการมีการคิดเชิงคำนวณ ทว่ายังไม่มีแนวทางปฏิบัติที่ชัดเจนในการจัดการเรียนรู้ที่ส่งเสริมการคิดเชิงคำนวณในชั้นเรียนวิทยาศาสตร์ ดังนั้นงานวิจัยมีเป้าหมายเพื่อ 1) วัดการคิดเชิงคำนวณของผู้เรียนก่อนและหลังเรียนด้วยการปฏิบัติทางชีวสารสนเทศขั้นพื้นฐาน และ 2) ศึกษาแนวปฏิบัติที่ดีในการใช้การปฏิบัติทางชีวสารสนเทศขั้นพื้นฐานเพื่อพัฒนาการคิดเชิงคำนวณ กลุ่มตัวอย่างคือนักเรียนชั้นมัธยมศึกษาปีที่ 4 โรงเรียนสาธิตแห่งหนึ่งในกรุงเทพฯ จำนวน 32 คน ผู้วิจัยออกแบบการจัดการเรียนรู้ แบ่งเป็น 2 ช่วง ได้แก่ การจัดการเรียนรู้โดยไม่ใช้คอมพิวเตอร์และใช้คอมพิวเตอร์ เก็บข้อมูลด้วยแบบวัดการคิดเชิงคำนวณ วิเคราะห์ข้อมูลด้วยสถิติเชิงพรรณนาและทดสอบความแตกต่างระหว่างค่าเฉลี่ยสองค่าที่ได้จากกลุ่มตัวอย่างสองกลุ่มที่ไม่เป็นอิสระต่อกัน (paired *t*–test) พบว่า ค่าเฉลี่ยคะแนนการคิดเชิงคำนวณก่อนและหลังเรียนเท่ากับ 17.78 (SD = 4.11) และ 21.65 (SD = 2.18) แตกต่างกัน ($t_{31, .05} = 7.08$, $p < .05$) รวมถึงค่าเฉลี่ยคะแนนทั้ง 4 องค์ประกอบเพิ่มขึ้นอย่างมีนัยสำคัญเช่นกัน และครูผู้สอนควรจัดการเรียนรู้โดยใช้การปฏิบัติทางชีวสารสนเทศที่ท้าทายและเชื่อมโยงกับชีวิตประจำวันต่อผู้เรียนอย่างชัดแจ้งและเนื้อหาสอดคล้องกับหลักสูตรวิทยาศาสตร์ของประเทศ เพื่อการใช้และพัฒนาการคิดเชิงคำนวณอย่างต่อเนื่อง

คำสำคัญ: ชีวสารสนเทศ การคิดเชิงคำนวณ วิทยาการคำนวณ

Development of Computational Thinking of Tenth–Grade Students Using Basic Bioinformatics Practices

Natthasit Norasit[1], Pongprapan Pongsophon[1*],
Wanwipa Vongsangnak[2,3] and Santichai Anuworrachai[4]

[1]Department of Education, Faculty of Education, [2]Department of Zoology, Faculty of Science, [3]Omics Center for Agriculture, Bioresources, Food, and Health, Kasetsart University (OmiKU), Bangkok, 10900, Thailand; [4]Department of Science and Technology, Kasetsart University Laboratory School, Center for Educational Research and Development, Bangkok, 10900, Thailand

[*]E-mail: pongprapan.p@ku.th

Received: 12 March 2023 Revised: 13 October 2023 Accepted: 5 December 2023

Abstract

One impact of computing in scientific fields and thinking processes lies in the processing of voluminous scientific data. Students therefore need to prepare themselves to confront the upcoming digital era and handle cutting–edge technology using computational thinking (CT); however, this is still absent from typical science classrooms. Hence, the purposes of this study were to 1) assess students' CT before and after learning basic bioinformatics practices and 2) study what are good practices to incorporate bioinformatics practices to enhance students' CT. Researchers designed four learning plans using inquiry–based learning and basic bioinformatics practices, having two parts: unplugged and plugged–in sessions. Data were collected using CT tests and analyzed using descriptive statistics and a paired *t*–test. The participants comprised 32 tenth–grade students in a science–technology emphasis program at a demonstration school in Bangkok, Thailand. The results showed CT pretest and posttest mean were significantly different by 17.78 (SD = 4.11) and 21.65 (SD = 2.18), respectively ($t_{31,\ .05} = 7.08$, $p < .05$). Additionally, the development of CT was evident in the improvement of all four CT components as well, and good practices to incorporate bioinformatics practices is to use real–life bioinformatics challenges explicitly and related to the standard science curriculum to maintain engagement in and persistence of CT usage.

Keywords: Bioinformatics, Computational thinking, Computing science

Introduction

Today, computer science permeates every aspect of human life, especially in scientific enterprises. Computer science concepts and their derivatives allow people to achieve things in diverse ways, far beyond what was possible in the past. It increases scientists' capabilities to conduct research or handle substantial work by using computational tools (Schatz, 2012). However, as learning facilitators, we see computing science as an opportunity for a great way to create learning experiences for students. We should help our students learn how to use computational tools intellectually, not treating them as a formidable obstacle, for learning, discovering new things, preparing to become citizens of the digital world, and making the world a better place (Dede, 2014). Therefore, we need students to learn the underlying principles of computer science, not just sophisticated programming languages, but general skills they should gain for their benefit, which we term computational thinking (CT). CT are mental processes for efficiently carrying out a complex task that involves computer science concepts. Through CT, humans may only involve their own thought processes or may incorporate electronic devices (Bocconi *et al*., 2016; Cuny *et al*., 2010; Lee *et al*., 2011; Wing, 2006; Yadav *et al*., 2017). CT is necessary for people at all levels and in all disciplines, including scientific fields (Grover and Pea, 2013). It enables us to deal with complex tasks or large unstructured datasets. CT is integrated and has value in science, but it has been underrepresented in the science classroom (Bocconi *et al*., 2016; Weintrop *et al*., 2016; Yadav *et al*., 2016).

Scientific knowledge and advancement are based on data; their value depends on how the data is used. Scientific data are often large and complex, termed "big data", and scientists make sense of phenomena using these data (Tan *et al*., 2009). Scientists engage in inquiry to discover the value and explanations behind natural phenomena and information. This process involves diligently searching for answers, and extracting insights from big data (IPST, 2018a). Similarly, in the field of biology, practitioners must rely on thinking processes to address scientific challenges and handle the abundance of biological data (National Research Council [NRC], 2005; Sul and Williams, 2011). Our students must also be prompted to thrive in the age of big data (NRC, 2009). The scientific field of handling biological data and applying computational techniques to understand those data is Bioinformatics (Luscombe *et al*., 2001). Bioinformatics is principally the study of biological data management, including data interpretation, storage, and retrieval, for scientists to study, do research which all require thinking computationally (Fillinger *et al*., 2019; NRC, 2005). Bioinformatics is also an intersection of pivotal modern biological sciences and computer science (Bayat, 2002; Fillinger *et al*., 2019; Khan *et al*., 2013). Hence, bioinformatics embedded with computational processes represents a potential route towards reinforcing students' CT. There is also a need to encourage students in the interdisciplinary field of bioinformatics amid a surge in demand (NRC, 2009). The authors do not want to force our students to learn bioinformatics over an extended time or in any special course, which might adversely affect their feelings about bioinformatics (Machluf and Yarden, 2013; Machluf *et al*., 2017). Therefore, in this present study, the authors designed learning units by integrating CT within normal science lessons and following national standard curricula. Thus, all students are able to have an effective learning process and gain CT without requiring additional lesson time (Form and Lewitter, 2011).

Thailand's national economic policies promote high–skilled human resources and target becoming a country strengthened by informatics (Division of Research and Educational Quality Assurance, 2017). However, the authors found no bioinformatics foundations and practices in existing high school science curricula, obvious learning outcomes, or supporting materials to bring bioinformatics to the classroom (IPST, 2017a). Science education must follow the same direction as policies, as bioinformatics is still underrepresented in a normal science classroom (Attwood *et al*., 2019; Bain *et al*., 2022). To achieve these changes, we must prepare our students' underlying CT processes that allow them to effectively perform tasks to give the best outcomes. One challenge is the lack of tangible teaching materials that support bioinformatics in high schools (IPST, 2017a), so the authors developed bioinformatics learning sessions related to Thailand's national content standards by adjusting bioinformatics into practical and more straightforward aspects suitable for a high school level to understanding of biological contents which already exists in school textbook. Both bioinformatics and CT are critical for living in the 21st Century (Attwood *et al*., 2019). Students engage in bioinformatics practices to gain in–depth understanding and broaden their modern scientific knowledge and develop CT ability. There is a connection between CT and bioinformatics, and bioinformatics practices harbor computational aspects and involve the use of CT processes; hence, in this research, the authors aim to assess students' CT development before and after learning with basic bioinformatics practices and to study good practices to incorporate bioinformatics practices to enhance students' CT development.

Research Objectives

1. To assess students' CT development before and after learning with basic bioinformatics practices.
2. To study what are good practices to incorporate bioinformatics practices to en-

hance students' CT development.

Research Methodology

The authors used purposive sampling (Creswell and Poth, 2018) by selecting one from five classes in a science–technology program from a university–facilitated school, Bangkok, Thailand. The participants consisted of 32 tenth–grade students (20 males and 12 females). Students varied in learning abilities based on their GPAX grades. We accommodated learning engagement by allowing them to work in small groups (3–4 persons) so that they could support each other and handle difficult bioinformatics tasks together (Qin, 2009). This study focused on 4 CT components and definitions as shown in Table 1 to align with computing curriculum (Bocconi *et al*., 2016; Dagiene and Sentance, 2016; Looi *et al*., 2018; IPST, 2017b). The authors designed four lesson plans (Table 2) that focused on four main components of CT by using inquiry–based learning and basic bioinformatics practices which aim to develop CT components.

Lesson plans using bioinformatics practices were divided into two sequential sessions: without using a computer (unplugged session) and using a computer (plugged–in session), which aim to promote CT usage and scaffold CT (Sun *et al*., 2021). For effective bioinformatics practices, we integrated basic bioinformatics practices into biology content (Table 2).

To assess students' development of CT in this study, we used a CT test adopted from the UK Bebras tasks, an international challenge on informatics and CT developed by experts in the fields of computer science and computing (University of Oxford, 2018). The authors selected UK Bebras tasks by considering age group, difficulty, and CT aspects. Then we got 8 tasks (Table 3) that students must solve within 30 minutes. For the record, the fewest number of tasks students can solve is 12 within 45 minutes (University of Oxford, 2018).

Table 1 Computational thinking components and definitions

Computational thinking components	Definition
1. Decomposition (DE)	To break a complex problem into small–scale subproblems or several components which can be handled easily or effectively
2. Pattern recognition (PR)/ Generalization (GE)	To identify patterns and find commonality in information which can be categorized and solved by applying similar solutions
3. Abstraction (AB)	To find essential details which must be used to solve problems while ignoring irrelevant redundant data
4. Algorithm/Algorithmic thinking (AL)	To order precise steps of actions to accomplish a task and achieve desired outcomes

Sources: Bocconi *et al*., 2016; Dagiene and Sentance, 2016; Looi *et al*., 2018

Table 2 Lesson plans using bioinformatics practices to promote students' computational thinking

Learning session 1: Unplugged bioinformatics practices (hands-on practices using given data)		
CT–embedded inquiry activities	**Learning objectives & Bioinformatics practices**	**Student's roles (CT Usage) & Teacher's roles**
1. Classification and diversity of ideal organisms, viruses, and protists (Time: 100 min)	**Learning objectives:** 1. Engaging in classification; categorizing and pattern matching of biological data 2. Becoming acquainted with bioinformatics processes **Bioinformatics practices:** – Data pattern recognition and data sorting processes	**Student's roles (CT Usage):** – Analyze the traits and intricate features of individual organisms (DE) – Identify commonalities and distinctions among specified organisms (PR) – Define criteria for grouping organisms systematically and creating a phylogenetic tree using selected criteria (AB) – Organize and executing plans for each step-in bioinformatics tasks (AL) **Teacher's Roles:** – Inspires students to participate actively in the bioinformatics process – Encourages inquiry by posing questions that spark curiosity and explore alternative solutions – Requires students to present and demonstrate their findings, stimulating using CT within each group – Provides feedback and offers alternative approaches to enhance learning
2. Bacterial classification using nucleotide sequences and tracking SARS–CoV–2 variants in Thailand using a sequence–based phylogenetic tree (Time: 100 min)	**Learning objectives:** 1. Engaging in organism classification using comparative nucleotide sequences obtained from shotgun sequencing algorithms 2. Understanding shotgun sequencing algorithm processes 3. Constructing a sequence-based phylogenetic tree **Bioinformatics practices:** – Shotgun sequencing algorithms – Sequence–based phylogenetic tree construction and analysis	**Student's roles (CT Usage):** – Analyze nucleotide contig sequences to identify patterns within given bacterial genomes (PR) – Eliminate redundant contigs (AB) – Examine component of provided data (DE) – Establish criteria and categorize virus into specific groups (PR) – Progress through steps to uncover a complete sequence in the task (AL) – Use viral structure and genomics data to create a relational diagram (AB) **Teacher's Roles:** – Provide guidance on how to compare nucleotide sequences for organism classification – Explain the intricacies of shotgun sequencing algorithms using visual aids, and interactive demonstrations, to ensure students comprehend the algorithmic processes. – Use real–life examples and case studies to illustrate the practical application. - Provide feedback on their phylogenetic tree construction

Table 2 (*continued*)

Learning session 2: Plugged-in bioinformatics practices		
CT–embedded inquiry activities	**Learning objectives & Bioinformatics practices**	**Student's roles (CT Usage) & Teacher's roles**
3. Searching through online biological databases and resources (Time: 100 min)	**Learning objectives:** 1. Retrieving data of organisms of interest from reliable online sources 2. Using bioinformatics interactive tools to construct a phylogenetic tree based on datasets from NCBI **Bioinformatics practices:** – Learning features of online biological databases (i.e., NCBI, Kew Gardens plants database) and data retrieval – Phylogenetic tree construction using phyloT interactive software	**Student's roles (CT Usage):** – Break down the procedure into distinct steps for extracting data (DE) – Delineate specific actions and performing iterative step–by–step procedures to obtain data (AL) – Navigate and retrieve authentic data; understanding technical terms, filters, and formulating precise search queries. (AB) – Compare different datasets of species to draw conclusion by hands-on or software (PR) **Teacher's Roles:** – Introduce scientific databases and their importance in study and research – Ensure all students have the devices and internet access. – Guide students how to conduct searches – Assist students during their database searches
4. Interpretation of data using online software and Python programming language (Time: 100 min)	**Learning objectives:** 1. Understanding and using sequence alignment using the online software "Basic Local Alignment Search Tool" (BLAST) and find relationship of unknown species sequences using BLAST 2. Using data from an online biological database and processes data using Python language **Bioinformatics practices:** – Pairwise sequence alignment using BLAST – Data retrieval and using basic Python programming for genomic data interpretation	**Student's roles (CT Usage):** – Obtain nucleotide sequence data related to the organism of interest (AB) – Use BLAST software for data comparison, leading to conclusions about organism relationships (PR) – Develop Python scripts to handle biological data and debugging involves finding and correcting errors to ensure the program runs as intended (AB & AL) – Create a pseudocode plan and then convert it into Python code. (DE & AL) **Teacher's Roles:** – Introduce importance of organism classification through pairwise sequence alignment and programming in Biology – Encourage peer–to–peer learning – Ensure all students have the devices and internet access. – Assist students when they encounter programming problems

Due to constraints in the allocated class time, we need to shorten the test duration to 30 minutes for 8 tasks. To confirm their reliability, three experts in the fields of science and computer science education evaluated the instructional materials and Bebras items. Following that, the authors conducted a trial with individuals who were not part of the study, in mixed–ability classes resembling the study group. This was done to ensure that the translation was accurate and to remove any tasks that proved to be either oversimplistic or excessively complex to solve. After the completion of the trial, the task difficulty levels ranged from moderate (challenging for students in grades 8 and 9) to advanced (challenging for students in grades 10 and 11). To assess students' CT levels, we adjusted the scoring criteria to a scale of +3 to 0 points, unlike the scoring used in the Bebras competition (+6 for correct, 0 for incorrect or unanswered) (University of Oxford, 2018) as this study aimed to measure CT development rather than rankings. Each task was evaluated with a maximum of 3 points, as indicated in Table 3, to reflect the CT growth of the students. While each task had the potential to evaluate multiple CT components as shown in table 3, each task specifically focused on assessing one of the four components: abstraction (AB), algorithmic thinking (AL), decomposition (DE), or pattern recognition (PR). This alignment was in accordance with both the core CT principles established by Dagiene and Sentance (2016) and Thailand's core standard computing curriculum (IPST, 2017b). To perform qualitative data analysis, the authors gathered evidence through the teacher's journal, in–class video recordings, and students' reflective journals. These sources were used to assess the bioinformatics concepts learned and observe the CT behaviours exhibited by students during bioinformatics tasks.

Table 3 List of selected computational test tasks

No.	Task names	Age groups and difficulty	Targeted CT components	Scoring criteria (Adjusted)
1	Arrow maze	Junior: B / Intermediate: A	Algorithm (AL)	**3 points** – the best answer or solution **2 points** – an alternative answer that was still correct but not the best **1 point** – an alternative answer that was still correct but not as good as 2 points **0 points** – incorrect answer or unanswered
2	Elevator	Intermediate: B / Senior: A		
3	Beaver land	Junior: B / Intermediate: A	Abstraction (AB)	
4	Balls	Intermediate: B / Senior: A		
5	Five sticks	Intermediate: A / Senior: A	Decomposition (DE)	
6	Sticks and shields	Intermediate: A / Senior: A		
7	Toll roads	Intermediate: B / Senior: B	Pattern recognition (PR)	
8	Candy maze	Intermediate: C / Senior: B		

Source: University of Oxford, 2018

Data analysis

Each CT component and overall CT mean scores were analysed using descriptive statistics and paired *t*–test. Qualitative data was analysed using the teacher's journal, video records, students' reflective journals by using an inductive approach (Creswell and Poth, 2018; Thomas, 2006) by analyzing students' reflective journals to assess the frequency of their usage of each CT component during class and observing their behaviour that reflex usage computational thought processes during the bioinformatics practices and to identify recurring effective practices for enhancing CT through bioinformatics.

Results

When we introduced bioinformatics, students appeared unfamiliar with it and barely realized its significance, and it seemed irrelevant to the scientific curriculum content that they prioritized. We encouraged them by merging practices with existing biology content knowledge and showed them how bioinformatics could enlighten them in that the knowledge learned actually came from bioinformatics methods; for example, bioinformatics processes to obtain nucleotide sequences of organisms using a shotgun sequencing algorithm and comparative genomics in the second CT–embedded bioinformatics practice shown in Figure 1. Each group must understand the importance of these practices and perform processes to acquire complete nucleotide sequences, using them to compare given nucleotide data to solve problems.

Figure 1 Student performed a tangible shotgun sequencing algorithm process to find complete nucleotide segments using printed paper in second inquiry activities – bacterial classification and sequencing

At the beginning of the bioinformatics learning module, we found that bioinformatics was apparently unfamiliar to the students, and they perceived no need to learn about it at this grade level. Therefore, we explicitly showed them how important and useful bioinformatics practices are in our lives, for instance, by tracking the spread of SARS–CoV–2 variants in Thailand in learning session 1, and demonstrated aspects related to their curriculum contents, for instance, why taxonomists classify organisms, or how biologists can categorize animals by comparing genomes. When they realized the value of bioinformatics, they seemed to pay more attention and put more effort into the learning process.

After the students became acquainted with bioinformatics and its practices using CT, we then proceeded to plugged-in sessions in which electronic devices (PC, smartphone, tablets) were used as powerful learning tools to give them access to authentic biological databases and encouraged them to use a computer programming language. Python was chosen because it has simple syntax, and students had learned Python in other school subjects, so a peer-assisted environment was created that helped plugged–in sessions progress smoothly. Challenges were presented in plugged-in sessions by engaging students to access reliable databases, retrieve SARS–CoV–2 genomic data from the National Center for Biotechnology Information (NCBI), and interpret them using Python code they had written themselves. The authors aimed for them to use only basic levels of Python code with which students at all cognitive levels could start writing their own code (not copying from other sources), such as using simple mathematical operations; conditionals and loops; and or, if, and else conditional statements.

A paired *t*–test was employed to assess the improvement in students' CT. The results indicated a statistically significant difference in students' CT scores before (M = 17.78, SD = 4.11) and after (M = 21.66, SD = 2.18) participating in inquiry-based learning with bioinformatics practices ($t_{31, .05} = 7.08$, $p < .05$), suggesting that students indeed experienced development in their CT abilities. This development was statistically evident across all four dimensions: algorithm ($t_{31, .05} = 4.45$, $p < .05$), abstraction ($t_{31, .05} = 3.90$, $p < .05$), decomposition ($t_{31, .05} = 5.21$, $p < .05$), and pattern recognition ($t_{31, .05} = 4.71$, $p < .05$), as illustrated in Table 4. On an individual basis, 30 out of 32 students exhibited higher CT scores, while only 2 students' CT scores remained unchanged.

Conclusion

The results of this study showed that 1) inquiry–based learning with basic bioinformatics practices embedded with computational processes and related to existing curriculum can promote the development of CT. Starting with unplugged sessions, students performed hands–on tangible bioinformatics to learn and understand the essential underlying processes. Subsequently, students used electronic devices as learning tools to access real scientific databases and could retrieve those data for interpretation using Python programming. These unplugged and plugged–in bioinformatics practices reinforced students' CT as reflected in CT test results. 2) good practices to incorporate bioinformatics practices is to use real–life bioinformatics challenges explicitly and in a way that is related to the standard science curriculum to maintain engagement in and persistence of CT usage. This study showed that our teaching approach will enable students to integrate biology and computing, fostering a multidisciplinary learning experience that helps them keep up with the evolving knowledge.

Table 4 Paired *t*–test results for computational thinking development

CT Components	Pretest		Posttest		*t*
	Mean	SD	Mean	SD	
1. Algorithm	4.78	1.36	5.75	0.67	4.45*
2. Abstraction	4.88	1.18	5.56	0.62	3.90*
3. Decomposition	3.69	1.80	5.00	1.37	5.21*
4. Pattern Recognition	4.44	1.34	5.34	0.94	4.71*
Overall CT Score	**17.78**	**4.11**	**21.66**	**2.18**	**7.08***

**p* < .05

Discussion

The authors designed learning bioinformatics modules which aim to identify effective practices for enhancing CT through bioinformatics. In the unplugged learning sessions, the authors found that the main characteristics of bioinformatics practices that students had to actively engage to develop CT were using real–world challenges or practical applications related to science curriculum contents, which helps students gain a deeper understanding of underlying processes. When we initially introduced the concept of bioinformatics to the students, it became apparent that they were largely unacquainted with it, and they had limited awareness of its significance. To elaborate on this, the first unplugged activity, classification of viruses, students learnt about viruses, which is corresponding to the SARS–CoV–2 pandemic outbreak situation and related to biodiversity content. Each group of students classified viruses using different criteria; genomic data, viral attributes, or hosts. This bioinformatics task challenged them to think about how scientists precisely group organisms that are all seemingly alike, so they had to choose the proper criteria. Therefore, during this bioinformatics activity, every step relied heavily on the CT processes as shown in Figure 2. Bioinformatics integrated with computational methods supports the cultivation of CT in students, consistent with the findings from Qin (2009) that life sciences students developed skills sets of CT after completing bioinformatics courses.

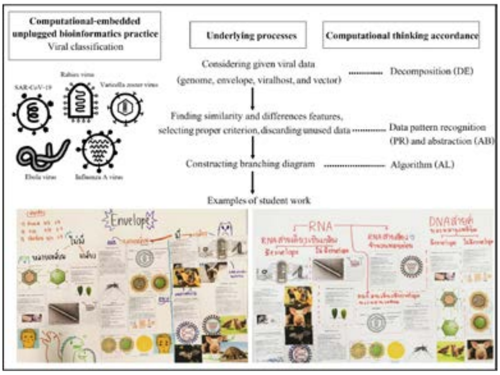


Figure 2 Schematic diagram showing the underlying bioinformatics process of real–world challenges related to science curriculum contents (Biodiversity and Systematics), which is in accordance with CT.

When students were primed with unplugged activities using hands–on tangible bioinformatics tasks to learn underlying scientific concepts and processes, such as organism classification, sequencing algorithms, or comparing genomes using printed papers. This allowed students to participate concretely through collaboratively engaging and practicing thought, which allowed them to use computational processes as suggested from the study of Form and Lewitter (2011) that teacher should create a basic representation of data analysis using traditional tools like pencils and paper before utilizing computer programs. The authors designed tasks to maintain students' attention and keep them engaged. One approach was setting a theme for each activity, i.e., systematically categorizing newly found species in a first inquiry activity or how to track the origins of SARS–CoV–2 using phylogeny (Martins *et al*., 2020; Putwattana, 2019; Qin, 2009; Udomsin and Porntrai, 2017). These hands–on tangible bioinformatics practices in unplugged sessions are critical for students to understand how computational algorithms are used for sequencing and alignment. Active involvement in computational methods influence the use and helping development of students' CT (Rubinstein and Chor, 2014; Stewart *et al*., 2021). The study by Kleinschmit *et al*. (2021) also revealed that without physical practices, students tend to overlook the computational algorithms and struggle to comprehend them.

During plugged–in sessions, online biological databases serve as valuable tools for students to explore bioinformatics while applying CT to navigate and understand their contents. In learning activities with electronic devices, students must think computationally. They need to understand the objectives of the learning tasks and possess effective search strategies for navigating intricate database interfaces to access scientific data. Then, students wrote Python scripts to interpret and obtain valuable information from raw data in databases, for example, the calculation of percentage GC content, which is one criterion for classifying organisms from an evolutionary perspective (Li and Du, 2014). Writing Python scripts to interpret bioinformatics data was quite hard for inexperienced students, so we applied strategies correlated to the use–modify–create framework from Lee *et al*. (2011), in which students were taught basic syntax and simple mathematical scripts adequate for them to adapt and use to interpret data. Investigating through programming activities involving bioinformatics, students had to write and design their own Python scripts to interpret genome data from databases. In the case of students who were unexperienced in Python, they adopted basic mathematical operations that were sufficient for our bioinformatics practices. Through this plugged–in session, while students wrote and modified Python code for use in another situation, they also had to debug their own scripts along the way. Those scaffolded programming bioinformatics task embedded with computational processes is a supportive strategy that can reinforce CT (Lee *et al*., 2011; Sentence and Csizmadia, 2017). Some students with programming skills can write complex commands using loops or conditional statements (Figure 3). Furthermore, students were assigned to work in small groups, which created a peer–assisted collaborative learning environment in which they helped each other to fix minor mistakes in syntax. This can drive an effective plugged–in session (Sondakh *et al*., 2020; Qin, 2009). Linking a designed task to curriculum content and using familiar real–life situations were important for keeping students engaged in bioinformatics activities (Form and Lewitter, 2011; Martins *et al*., 2020; Qin, 2009). For example, the authors used yeast *Saccharomyces* or bacteria *Lactobacillus* species, which are commonly used for fermented products or dairy products, as examples to draw students' attention, keep them actively engaged, and encourage them to explore those species through a complex interface of online databases. (Chanchaichaovivat *et al*., 2016)

```
with open("log.txt", "r") as f:
        g = f.read()
        g.replace("\n", "")
        f.close()

A = g.count("A")
T = g.count("T")
C = g.count("C")
G = g.count("G")
ma = len(g)

print(f"A: {A}, T: {T}, C: {C}, G: {G}, All: {ma}")
l = C + G

persen = l/ma *100

print(f"Percent: {persen}%")
```

```
genome = "AATCG"
no_basepairs = len(genome)
no_A = genome.count("A")
print("จีโนมของสิ่งมีชีวิตนี้ประกอบด้วย",no_basepairs,"คู่เบส")
print("จีโนมของสิ่งมีชีวิตนี้ประกอบด้วยเบส A จำนวน" , no_A, "ตัว")

corona =
```

Figure 3 Peer–assisted environment and electronic devices used as powerful learning tools to run BLAST (Basic Local Alignment Search Tool) software during plugged–in bioinformatics sessions (left). Examples of basic Python code written by students for SARS–CoV–2 genomic interpretation (right)

The significance of bioinformatics is in finding patterns in given information, such as nucleotide sequence overlapping and multiple sequence alignment for finding differences, as shown in Figure 4. When dealing with real scientific data, we should give students time to use their own strategies to manage and try out different ways of obtaining results (Kleinschmit *et al*., 2021). The use of bioinformatics which related to existing science curriculum was also suggested by the study of Machluf and Yarden (2013) that when teacher designs inquiry–based bioinformatics activities, teacher must link to preexisting curricula, and which are suitable for the high school student's cognitive level. The students gained their CT and knowledge of the scientific concepts through repeatedly working through the bioinformatics tasks by themselves. (Arık and Topçu, 2022; Chompunuch *et al*., 2022).

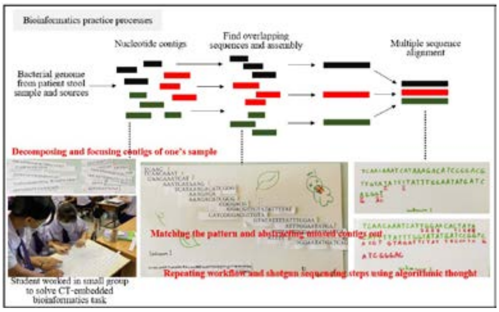


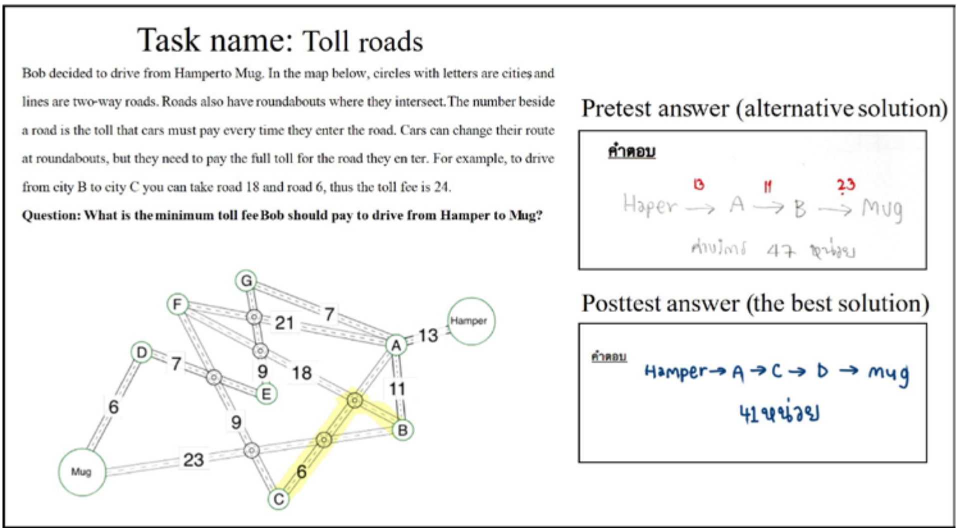
Figure 4 Schematic diagram showing the underlying bioinformatics process of real–world challenges related to science curriculum content (Genetics) linked to computational thinking

In the findings, it became evident that following the completion of the bioinformatics modules, 30 out of 32 students achieved improved scores, while only 2 students' scores remained unchanged. Personal performance in CT was influenced by many factors, such as past experience with computing and programming or any thinking process related to CT; analytical thinking, problem–solving skills, cognitive levels and mathematical achievements, which shaped students' ways of solving Bebras (Durak and Saritepeci, 2018). Moreover, because CT derives from a computing discipline, Hatlevik *et al*. (2018) found that experience in computer programming positively affects CT ability. And also using Bebras tasks as a CT assessment tool, each Bebras task has different computational components and difficulties, which may influence a student's CT test score (Izu *et al*., 2017). The authors found that performing bioinformatics practices required students to use computation–like thinking to handle complex processes by decomposing them into simple manageable steps, finding data patterns, abstracting essential details, abstaining from non-essential work, and choosing the correct way to accomplish a task. Students are confronted with the complex processes, ambiguity, and heterogeneity of biological big data, which pushes them to use CT (NRC, 2005). They must decompose the complex steps of a bioinformatics task into small accomplishable tasks, and then they must prioritize which should be done in the correct sequence. The results from each step are used for subsequent steps. To accomplish this, they must actively engage in the learning process and work effectively with their peers. The authors observed that students became aware of the use of CT embedded in activities through an analysis of students' reflective journals. Two examples from the students' journals are as follows:

> "*I practice finding similarity patterns in bacterial DNA sequence and then compare sequence between those bacteria*", said student No. 5

> "*I considered data on website and brought only necessary data to complete the assignment*", said student No.15

Hence, these bioinformatics practices foster the enhancement of students' CT. These systematic thought processes were incorporated into each bioinformatics task, leading to consistent and repeated application of CT over time. This development aligns with the findings of studies conducted by Israel–Fishelson and Hershkovitz (2019), Marques *et al*. (2014), and Rubenstein and Chor (2014). When we examined their CT posttest answer, we found that they thought more deliberately, chose better solutions, and abstained from minor mistakes than pretest answer (Figure 5).


Task name: Toll roads

Bob decided to drive from Hamperto Mug. In the map below, circles with letters are cities and lines are two-way roads. Roads also have roundabouts where they intersect. The number beside a road is the toll that cars must pay every time they enter the road. Cars can change their route at roundabouts, but they need to pay the full toll for the road they en ter. For example, to drive from city B to city C you can take road 18 and road 6, thus the toll fee is 24.

Question: What is the minimum toll fee Bob should pay to drive from Hamper to Mug?

Pretest answer (alternative solution)

คำตอบ

$\text{Haper} \xrightarrow{13} \text{A} \xrightarrow{11} \text{B} \xrightarrow{23} \text{Mug}$

ค่าโทรล์ 47 หน่วย

Posttest answer (the best solution)

คำตอบ

Hamper → A → C → D → mug

41หน่วย

Figure 5 Example of a student's pretest and posttest answers reflecting their computational thinking development

Implications

During plugged–in sessions, it's essential that all students have access to high-quality computer equipment, electronic devices, and reliable Wi–Fi for effective independent learning and seamless database utilization. Teachers should exercise caution when students use electronic devices, as they can potentially disrupt the learning process or be misused.

Human Subject Protection (HSP)

This project received ethical approval from Kasetsart University Research and Development Institute (KURDI) – study code KUREC–SS63/233 and was carried out in compliance with the international guidelines for human research standards including informed consent, personal data protection, respect for persons, and justice.

References

Arık, M., and Topçu, M. S. (2022). Computational thinking integration into science classrooms: Example of digestive system. **Journal of Science Education and Technology** 31: 99–115.

Attwood, T. K., Blackford, S., Brazas, M. D., Davies, A., and Schneider, M. V. (2019). A global perspective on evolving bioinformatics and data science training needs. **Briefings in bioinformatics** 20(2): 398–404.

Bain, S. A., Plaisier, H., Anderson, F., Cook, N., Crouch, K., Meagher, T. R., Ritchie, M. G., Wallace, E., and Barker, D. (2022). Bringing bioinformatics to schools with the 4273pi project. **PLoS Computational Biology** 18(1): e1009705.

Bayat, A. (2002). Science, Medicine, and the Future: Bioinformatics. **BMJ (Clinical Research Ed.)** 324(7344): 1018–1022.

Bocconi, S., Chioccariello, A., Dettori, G., Ferrari, A., and Engelhardt, K. (2016). Developing computational thinking in compulsory education – Implications for policy and practice. **EUR – Scientific and Technical Research Reports**. doi:10.2791/792158

Chanchaichaovivat, A., Kirdtabtim, S. and Phomphisutthimas, S. (2016). Application of Microorganisms in Sustainable Agriculture. **Journal of Research Unit on Science, Technology and Environment for Learning** 7(2): 398–413. (in Thai)

Chompunuch, P., Jantrasee R., and Saenna, P. (2022). Science academic hardiness and its relationship with science learning achievement of science–oriented program's students. **Journal of Research Unit on Science, Technology and Environment for Learning** 13(2): 250–270. (in Thai)

Creswell, J. W., and Poth, C. N. (2017). **Qualitative Inquiry and Research Design**. Los Angeles, USA: SAGE.

Dagiene, V., and Sentance, S. (2016). It's computational thinking! Bebras tasks in the curriculum. **Lecture Notes in Computer Science** 9973: 28–39.

Dede, C. (2014) **The Role of Digital Technologies in Deeper Learning**. Students at the Center: Deeper Learning Research Series. Retrieved from https://studentsatthecenterhub.org/resource/the-role-of-digital-technologies-in-deeper-learning, November 11, 2020.

Division of Research and Educational Quality Assurance. (2017). **Thailand 4.0 blueprint – Economic Model for Prosperity, Security, and Sustainability**. Retrieved from https://www.nstda.or.th/home/knowledge_post/blueprint-thailand-4, September 21, 2020.

Durak, H. Y., and Saritepeci, M. (2018). Analysis of the relation between computational thinking skills and various variables with the structural equation model. **Computers & Education** 116(1): 191–202.

Fillinger, S., de la Garza, L., Peltzer, A., Kohlbacher, O., and Nahnsen, S. (2019). Challenges of big data integration in the life sciences. **Analytical and Bioanalytical Chemistry** 411(26): 6791–6800.

Form, D., and Lewitter, F. (2011). Ten simple rules for teaching bioinformatics at the high school level. **PLoS Computational Biology** 7(10): e1002243.

Grover, S., and Pea, R. (2013). Computational thinking in K–12: A review of the state of the field. **Educational Researcher** 42(1): 38–43.

Hatlevik, O. E., Throndsen, I., Loi, M., and Gudmundsdottir, G. B. (2018). Students' ICT self–efficacy and computer and information literacy: Determinants and relationships. **Computers & Education** 118: 107–119.

Institute for the Promotion of Teaching Science and Technology [IPST], Thailand. (2017a). **Biology Curriculum Manual (Revised Version 2017 Based on the Basic Education Core Curriculum 2008)**. Retrieved from https://www.scimath.org/course-manual, October 5, 2020. (in Thai)

Institute for the Promotion of Teaching Science and Technology [IPST], Thailand. (2017b). **Standard Computing Science Curriculum Manual**. Retrieved from https://www.scimath.org/coursemanual, October 5, 2020. (in Thai)

Israel–Fishelson, R., and Hershkovitz, A. (2019). Persistence and achievement in acquiring computational thinking concepts: A large–scale log–based analysis. **Proceedings of E–Learn: World Conference on E–Learning in Corporate, Government, Healthcare, and Higher Education** (pp. 1002–1012). New Orleans, Louisiana, US: Association for the Advancement of Computing in Education (AACE). Retrieved from https://www.learntechlib.org/primary/p/211181, February 14, 2022.

Izu, C., Mirolo, C., Settle, A., Mannila, L., and Stupuriene, G. (2017). Exploring Bebras tasks content and performance: A multinational study. **Informatics in Education** 16(1): 39–59.

Khan, A. M., Tan, T. W., Schönbach, C., and Ranganathan, S. (2013). APBioNet – Transforming bioinformatics in the Asia–Pacific region. **PLoS Computational Biology** 9(10): e1003317.

Kleinschmit, A. J., Ryder, E. F., Kerby, J. L., Murdoch, B., Donovan, S., and Grandgenett, N. F. (2021). Community development, implementation, and assessment of a NIB LSE bioinformatics sequence similarity learning resource. **PLoS ONE** 16(9): e0257404.

Kleinschmit, A., Brink, B., Roof, S., Goller, C., and Robertson, S. D. (2019). Sequence similarity: An inquiry based and "under the hood" approach for incorporating molecular sequence alignment in introductory undergraduate biology courses. **CourseSource**. Retrieved from https://doi.org/10.24918/cs.2019.5, February 14, 2022.

Lee, I., Martin, F., Denner, J., Coulter, B., Allan, W., Erickson, J., Joyce, M. S., and Werner, L. (2011). Computational thinking for youth in practice. **ACM Inroads** 2(1): 32–37.

Li, X. Q., and Du, D. (2014). Variation, evolution, and correlation analysis of C+G content and genome or chromosome size in different kingdoms and phyla. **PLoS ONE** 9(2): e88339.

Looi, C. K., How, M. L., Longkai, W., Seow, P., and Liu, L. (2018). Analysis of linkages between an unplugged activity and the development of computational thinking. **Computer Science Education** 28(3): 255–279.

Luscombe, N. M., Greenbaum, D., and Gerstein, M. (2001). What is bioinformatics? A proposed definition and overview of the field. **Methods of Information in Medicine** 40(4): 346–358.

Machluf, Y., and Yarden, A. (2013). Integrating bioinformatics into senior high school: Design principles and implications. **Briefings in Bioinformatics** 14(5): 648–660.

Machluf, Y., Gelbart, H., Ben–Dor, S., and Yarden, A. (2017). Making authentic science accessible–the benefits and challenges of integrating bioinformatics into a high–school science curriculum. **Briefings in Bioinformatics** 18(1): 145–159.

Marques, I., Almeida, P., Alves, R., Dias, M. J., Godinho, A., and Pereira–Leal, J. B. (2014). Bioinformatics projects supporting life–sciences learning in high schools. **PLOS Computational Biology** 10(1): e1003404.

Martins, A., Fonseca, M. J., Lemos, M., Lencastre, L., and Tavares, F. (2020). Bioinformatics–based activities in high school: Fostering students' literacy, interest, and attitudes on gene regulation, genomics, and evolution. **Frontiers in Microbiology** 11(2410): 578099.

National Research Council [NRC] (US) Committee on a New Biology for the 21st Century: Ensuring the United States Leads the Coming Biology Revolution. (2019). **A New Biology for the 21st Century: Ensuring the United States Leads the Coming Biology Revolution**. US: National Academies Press.

National Research Council [NRC]. (2005). **Catalyzing Inquiry at the Interface of Computing and Biology**. Washington, DC: The National Academies Press.

Putwattana, N. (2019). Local wisdom and Science Education & Learning. **Journal of Research Unit on Science, Technology and Environment for Learning** 9(2): 352–369. (in Thai)

Qin, H. (2009). Teaching computational thinking through bioinformatics to biology students. **SIGCSE Bulletin** 41(1): 188–191.

Rubinstein, A., and Chor, B. (2014). Computational thinking in life science education. **PLOS Computational Biology** 10(11): e10 03897.

Schatz, M. C. (2012). Computational thinking in the era of big data biology. **Genome Biology** 13(11): 177.

Sentence, S., and Csizmadia, A. (2017). Computing in the curriculum: challenges and strategies from a teacher's perspective. **Education and Information Technologies** 22(2): 469–495.

Sırakaya, M., Alsancak Sırakaya, D., and Korkmaz, Ö. (2020). The impact of STEM attitude and thinking style on computational thinking determined via structural equation modeling. **Journal of Science Education and Technology** 29(4): 561–572.

Sondakh, D. E., Osman, K., and Zainudin, S. (2020). A proposal for holistic assessment of computational thinking for undergraduate: Content validity. **European Journal of Educational Research** 9(1): 33–50.

Stewart, W. H., Baek, Y., Kwid, G., and Taylor, K. (2021). Exploring factors that influence computational thinking skills in elementary students' collaborative robotics. **Journal of Educational Computing Research** 59(6): 1208–1239.

Sun, L., Hu, L., and Zhou, D. (2022). Single or combined? A study on programming to promote junior high school students' computational thinking skills. **Journal of Educational Computing Research** 60(2): 283–321.

Tan, T. W., Lim, S. J., Khan, A. M., and Ranganathan, S. (2009). A proposed minimum skill set for university graduates to meet the informatics needs and challenges of the "-omics" era. **BMC Genomics** 10(Suppl. 3): S36.

Thomas, D. R. (2006). A general inductive approach for analyzing qualitative evaluation data. **American Journal of Evaluation** 27(2): 237–246.

Tsai, C. Y. (2019). Improving students' understanding of basic programming concepts through visual programming language: The role of self–efficacy. **Computers in Human Behavior** 95: 224–232.

Udomsin, C., and Porntrai, S. (2017). Developing academic achievement in learning cell cycle and mitosis using a hands–on activity of science inquiry. **Journal of Research Unit on Science, Technology and Environment for Learning** 8(2): 326–340. (in Thai)

University of Oxford. (2018). **UK Bebras Computational Thinking Challenge**. Retrieved from http://www.bebras.uk, October 21, 2023.

Weintrop, D., Beheshti, E., Horn, M., Orton, K., Jona, K., Trouille, L., and Wilensky, U. (2015). Defining computational thinking for mathematics and science classrooms. **Journal of Science Education and Technology** 25(1): 127–147.

Wing, J. (2006). Computational thinking. **Communications of the ACM** 49(3): 33–35.

Yadav, A., Hong, H., and Stephenson, C. (2016). Computational thinking for all: Pedagogical approaches to embedding 21st century problem solving in K–12 classrooms. **Tech Trends** 60(6): 565–568.

Yadav, A., Stephenson, C., and Hong, H. (2017). Computational thinking for teacher education. **Communications of the ACM** 60(4): 55–62.