Chapter 7
Introducing and Assessing Computational Thinking in the Secondary Science Classroom

Hillary Swanson, Gabriella Anton, Connor Bain, Michael Horn and Uri Wilensky

Abstract The importance of computational thinking (CT) as a goal of science education is increasingly acknowledged. The representational affordances of computational tools are changing the way knowledge can be constructed, expressed, and understood across disciplines. Our group has worked to explicitly characterize CT practices used by computational STEM researchers (CT-STEM practices) and to develop computational science curricula that teach both CT-STEM practices and science content. We have previously characterized four strands of CT-STEM practices: *data practices*, *modeling and simulation practices, computational problem-solving practices*, and *systems thinking practices*. In this chapter, we show that a group of 9th grade students developed competencies for *modeling and simulation practices* as a result of their engagement in our computational biology curriculum. As evidence, we present findings from a quantitative analysis of students' written responses to assessments given before and after their participation in three computational biology units. Results suggest that the computational biology curriculum helped students develop a number of important competencies for the strand on *modeling and simulation practices*. Our work contributes to the field's understanding of how science curricula can be designed to foster students' development of CT-STEM practices and how this development can be assessed.

H. Swanson (✉) · G. Anton · C. Bain · M. Horn · U. Wilensky
School of Education and Social Policy, Northwestern University, Evanston, USA
e-mail: hillary.swanson@northwestern.edu

G. Anton
e-mail: gabriellaanton3.2020@u.northwestern.edu

C. Bain
e-mail: connorbain@u.northwestern.edu

M. Horn
e-mail: michael-horn@northwestern.edu

U. Wilensky
e-mail: uri@northwestern.edu

S.-C. Kong and H. Abelson (eds.), *Computational Thinking Education*,
https://doi.org/10.1007/978-981-13-6528-7_7

7.1 Introduction

The importance of computational thinking (CT) as a goal of science education is increasingly acknowledged (Quinn, Schweingruber, & Keller, 2012; Wilensky, Brady, & Horn, 2014). Teaching CT in the context of science not only presents students with a more authentic image of science as it is practiced today, it also increases access to powerful modes of thinking and marketable skills for many careers (Levy & Murnane, 2004). It is estimated that by 2020, one out of every two STEM jobs will be in computing (Kaczmarczyk & Dopplick, 2014). However, students from groups that have been historically underrepresented in STEM fields (such as women and racial minorities) are less likely to enroll in computer science (CS) classes (Margolis, 2008; Margolis & Fisher, 2003) and thus are not engaging in CT practices through traditional channels. Our goal is to improve access for all students, especially those underrepresented in CS, by embedding CT practices in subjects such as biology, chemistry, and physics, which all secondary students are expected to take.

We believe that developing CT practices in the context of science subjects is a productive endeavor. Our group has worked to explicitly characterize key activities relevant to particular CT-STEM practices as specific learning objectives and used these to guide our development of science curricula and assessments. In this paper, we show that a group of 9th grade students (ages 14–15 years) developed competencies for *modeling and simulation practices* as a result of their engagement in our computational biology curriculum.

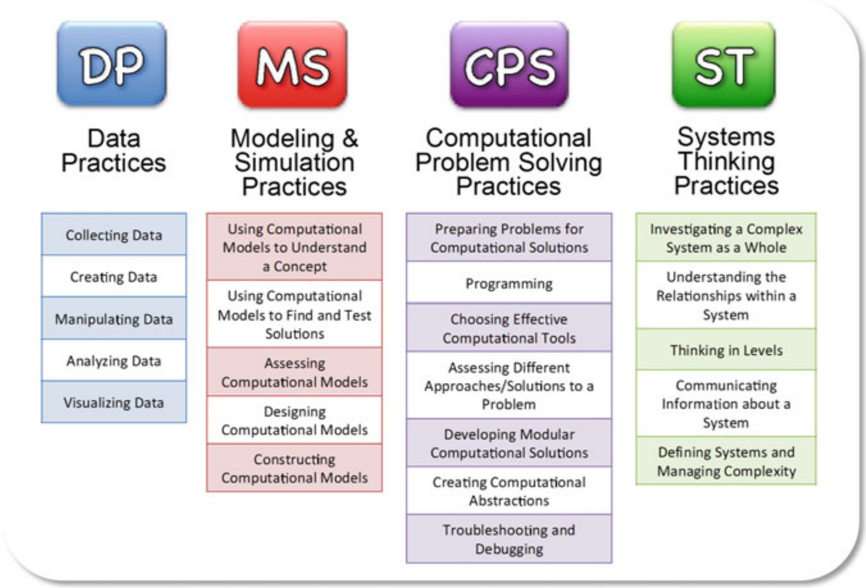
7.2 Theoretical Orientation

Our perspective on computational thinking is motivated by Wilensky and Papert's (2010) restructuration theory, which demonstrates that the representational form in which knowledge is encoded significantly influences how it may be understood and learned. Restructuration theory builds on a history of psychological and historical research that has argued that representational forms shape human knowledge and understanding, both at the individual and societal level (e.g., Goody, 1977; Papert, 1980; Olson, 1994; diSessa, 2001). In light of this theory, it is clear that the representational affordances of computational tools are changing the way knowledge can be constructed, expressed, and understood across disciplines.

Building on this perspective, our group has worked to characterize the nature of computational thinking practices in the STEM disciplines. On the basis of interviews with computational STEM researchers, we developed an operational definition of CT-STEM as a set of practices and organized these as a taxonomy (Weintrop et al., 2016). The taxonomy categorizes CT-STEM practices in terms of four major strands: *data practices*, *modeling and simulation practices*, *computational problem-solving* *practices*, and *systems thinking practices*. Figure 7.1 depicts the practices within each of these four strands.

Though they are not unique to STEM, these CT practices are common to the STEM disciplines. In this way, they differ from the domain-general CT practices characterized by Wing (2006) (e.g., *using computer science concepts to solve problems and design systems*), the National Research Council (2010) (e.g., *heuristic reasoning, search strategies, and problem abstraction* and *decomposition*), and Brennan and Resnick (2012) (e.g., *being incremental and iterative*, *testing and debugging*, *reusing and remixing*, and *abstracting and modularizing*). We identified key activities relevant to each of the CT-STEM practices in our taxonomy and proposed those as learning objectives. We have used these learning objectives to guide our development of curricula and assessments that foster and evaluate students' development of computational thinking practices in STEM subjects at the secondary level.

In the study described herein, we analyze student gains in the *modeling and simulation practices* strand of the taxonomy. We build on work we have done using agent-based modeling in science classrooms (Blikstein & Wilensky, 2009; Sengupta & Wilensky, 2009; Horn & Wilensky, 2012; Horn, Brady, Hjorth, Wagh, & Wilensky, 2014; Levy & Wilensky, 2009; Wilensky, 2003; Wilensky & Reisman, 2006). In future work, we plan to analyze each of the four strands and gains in summative assessments of CT-STEM practices.



DP

Data Practices

- Collecting Data
- Creating Data
- Manipulating Data
- Analyzing Data
- Visualizing Data

MS

Modeling & Simulation Practices

- Using Computational Models to Understand a Concept
- Using Computational Models to Find and Test Solutions
- Assessing Computational Models
- Designing Computational Models
- Constructing Computational Models

CPS

Computational Problem Solving Practices

- Preparing Problems for Computational Solutions
- Programming
- Choosing Effective Computational Tools
- Assessing Different Approaches/Solutions to a Problem
- Developing Modular Computational Solutions
- Creating Computational Abstractions
- Troubleshooting and Debugging

ST

Systems Thinking Practices

- Investigating a Complex System as a Whole
- Understanding the Relationships within a System
- Thinking in Levels
- Communicating Information about a System
- Defining Systems and Managing Complexity

Fig. 7.1 Computational thinking in STEM taxonomy

7.3 Method

We show that a group of 9th grade students developed competencies for *modeling and simulation practices* as a result of their engagement in our computational biology curriculum. As evidence, we present findings from a quantitative analysis of 133 9th grade students' written responses to assessments given before and after their participation in three computational biology units.

7.3.1 Study Design

The data in this study come from the fourth iteration of a design-based research cycle (Collins, Joseph, & Bielaczyc, 2004). The implementation spanned the 2015–2016 school year and was tested in three 9th grade biology classrooms at a partner secondary school in a Midwestern city in the United States. Students were given a CT-STEM practices pre-test (Weintrop et al., 2014) at the beginning of the school year and a CT-STEM practices post-test at the end of the school year. Over the year they participated in three CT-STEM biology units; each unit approximately four days long. We investigated the role of the CT-STEM science units in students' development of competencies for *modeling and simulation practices* by looking for statistically significant gains in student scores for particular items from pre- to post-test.

7.3.2 Participants

We partnered with a public secondary school (serving grades 7–12) in an economically depressed neighborhood in a large city in the Midwestern region of the United States. The school was selected for its diversity and for the willingness of its teachers to participate in our study. The size of the school was typical for an urban public secondary school, with approximately twelve hundred students enrolled (71.1% Black/African American, 24.5% Hispanic/Latino, 1.6% Asian, 0.3% American Indian, 0.2% Pacific Islander, 0.9% Bi-Racial, 1.4% White), with 62% from low income households. The school is characterized as selective-enrollment, meaning that the student population is academically advanced and highly motivated. We addressed our research questions by analyzing a selection of the pre- and post-test responses given by participating 9th grade biology students. A total of 133 of these students, distributed across three biology teachers, took both tests. Due to time constraints, a number of these students did not complete the entire assessment. Ten students did not complete the assessment item measuring our first learning objective and 24 did not complete the assessment item measuring our second learning objective; these students' responses were therefore removed from the analyzed datasets.

7.3.3 CT-STEM Units

The students participated in three computationally-enriched biology units over the course of the school year. Each unit took approximately four class periods and emphasized the exploration and manipulation of computational models of scientific phenomena or concepts. The first unit was on predator-prey dynamics and ecosystem stability. For this unit, students explored population dynamics in a simulation of an ecosystem consisting of three organisms (grass, sheep, and wolves) (Wilensky, 1997b). Students investigated the population-level effects of parameters for individual organisms (e.g., reproduction rate) by running the simulation with different values for each organism. Through their exploration, the students learned about the complex population dynamics that emerge from the interactions between individual organisms. The second unit was on HIV. For this unit, students explored a model that simulated the diffusion of the infectious disease through a population (Wilensky, 1997c). Students investigated the effects of parameters for individual interactions (such as the probability of individuals to form a couple and the probability of the disease to transfer between partners) on the rate of the spread of the disease. The third unit was on genetics. For this unit students explored a model that allowed them to change mating rules in a population of fish. Students investigated how changing parameters such as life span and mating choice could bring about changes in the overall allele frequencies in the population (Novak & Wilensky, 2011).

All units were meant to help students develop expertise regarding learning objectives related to *modeling and simulation practices* by engaging with science content through the exploration of NetLogo (Wilensky, 1999) simulations. NetLogo simulations were chosen because the agent-based modeling environments make complex systems phenomena (such as those featured in the biology units) more intuitively accessible (Wilensky, 2001). Additionally, the NetLogo user interface makes transparent the relationship between a model's code and the phenomenon it simulates. This makes NetLogo a powerful tool for scaffolding students' transition from consumers to designers and builders of computational models.

7.3.4 Data Collection

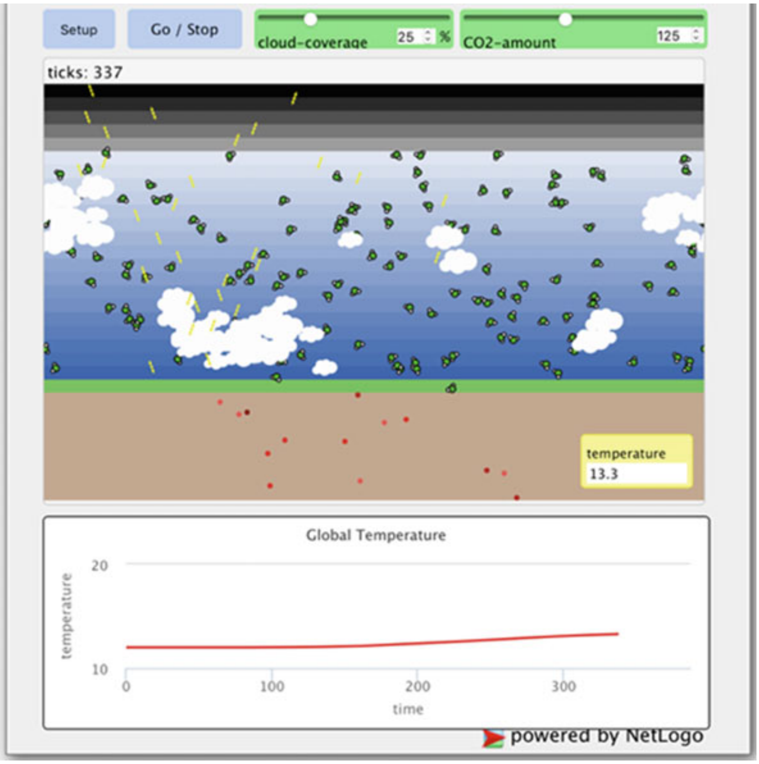
The pre- and post-test were each given during one 50-min class period at the beginning and end of the school year. Students took the tests individually on school laptops in their biology classrooms. The pre- and post-tests were not designed to evaluate students' science content knowledge. Rather, they were meant to evaluate their development of competencies relevant to CT-STEM practices. In this chapter, we present results concerned with two particular learning objectives within our *modeling and simulation practices* strand.

The first learning objective focuses on an activity relevant to the CT-STEM practice *using computational models to understand a concept* and states that a student should be able to "explore a model by changing parameters in the interface or code."

This is a very basic activity but it plays an important role in students' (and scientists') abilities to learn about the relationship between particular parameters and system behavior at the macro-level.

The second learning objective focuses on an activity relevant to the CT-STEM practice *assessing computational models* and states that a student should be able to "identify the simplifications made by a model." This activity is important to students' epistemological development, as it relates to their understanding of a computational model as a tool that is both powerful and limited with regards to the construction of new knowledge.

Both pre- and post-tests required students to interact with computational simulations which they were given basic instructions on how to operate. For the pre-test, students interacted with a simulation (shown in Fig. 7.2) that modeled climate change and showed the relationship between temperature and amount of CO_2 in the atmosphere (Tinker & Wilensky, 2007). For the post-test, students explored a simulation (shown in Fig. 7.3) that modeled the relationship between the pressure of a gas, its



cloud-coverage 25 %
CO2-amount 125
ticks: 337
temperature 13.3

Fig. 7.2 Screenshot of pre-test simulation that models the relationship between temperature and atmospheric CO_2 levels

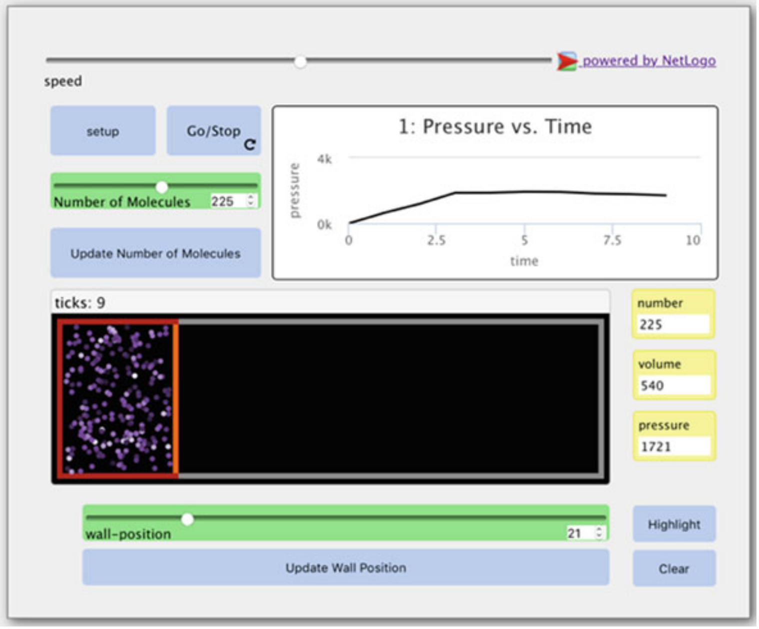


Fig. 7.3 Screenshot of post-test simulation that models the relationship between the pressure of a gas, its volume, and the number of particles

volume, and the number of particles in a sealed environment (Wilensky, 1997a, 2003; Wilensky, Novak, & Levy, 2005).

To assess students' abilities to *explore a model by changing parameters in the interface or code*, we analyzed their responses to test items (quoted below) that asked them to attend to the relationships between adjustable parameters and system-level characteristics. To assess students' abilities to *identify simplifications made by a model*, we analyzed their responses to test items (quoted below) that asked them for the ways in which the simulations differed from the real-world. These assessment items were selected to investigate students' development with respect to the same learning objectives across two very different computationally modeled phenomena.

7.3.5 Analytic Approach

We used a combined top-down (learning objective driven) and bottom-up (data driven) approach to create rubrics for characterizing students' competencies with respect to each learning objective and evaluating their responses to pre- and post-test questions. Two researchers then analyzed students' responses to the two assessment items for both pre-and post-tests. They coded responses (identifying the competencies outlined in the rubrics) and then scored them. The researchers' inter-rater reliability for the pre-test was 97% for the item measuring the first learning objective and 90% for the item measuring the second learning objective. Inter-rater reliabilities for the post-test items were 95% and 80% respectively.

7.3.5.1 Learning Objective 1: Explore a Model by Changing Parameters

For the pre-test, students were asked to *explore a model* by changing its parameters in the context of the greenhouse gas simulation. In particular, they responded to the prompt: "Set cloud coverage to 0%. Take some time to experiment with different settings for the 'CO_2-amount' slider. What happens to the temperature if you increase the amount of the CO_2 in the model?" For the post-test, students were asked to *explore the model* in the context of the gas-law simulation. In particular, they responded to the question: "What values for container size and number of particles will result in the lowest pressure in the container? What steps did you take to come up with these values?" It is important to note that while both items are concerned with students' abilities to learn about a parameter's influence on a system's behavior, they are inversely structured. While the pre-test item instructs students to change a parameter and report its effect on the system, the post-test item instructs students to change parameters until they achieve a specified system behavior. We argue that while they are different, both items are concerned with the causal relationship between parameter values and system-level behavior and are therefore comparable assessments of students' abilities to *explore a model by changing parameters in the interface or code*.

We examined students' pre- and post-test responses, sorting responses into categories based on similarities that were relevant to the learning objective. Three categories emerged that were representative of response types across both pre- and post-test. These are *comparing across trials, attending to explanatory factors*, and *attending to parameter-system relationships*. We identified these as three competencies relevant *to exploring a model by changing parameters in the interface or code*. These competencies are outlined, described, and illustrated with examples from the data in Table 7.1.

We scored students' responses by awarding one point for each competence demonstrated in their response and taking the sum of these points. This resulted in scores ranging from 0 to 3. We characterize the distribution of competencies (demonstrated in both pre- and post-test) in our findings section.

7.3.5.2 Learning Objective 2: Identify Simplifications Made by a Model

As part of the pre-test, students were asked to identify the simplifications made by the greenhouse simulation. As part of the post-test, they were asked to identify the simplifications made by the gas-law simulation. For both tests, they responded to the

Table 7.1 Pre- and post-test rubric for analyzing students' responses and characterizing the competencies they drew upon when *exploring a model by changing parameters in the interface or code*

<table>
<tr><td colspan="2"><i>Comparing across trials</i>
Response compares data across multiple simulation trials. When exploring a model to learn more about the dynamics of, or test a hypothesis regarding, a complex system, it is important to observe more than one simulation run. This is because complex systems are inherently random and the results of changing a parameter vary over different simulation trials. A pattern of cause-effect relationships will hover around an average tendency, but this average tendency may not be exactly embodied in one (or several) simulation trials. So, if a student only runs one trial, they may have a misguided impression of a pattern in system behavior. It is also a good idea to run multiple trials in order to systematically compare the effects of different parameter values on system behavior.</td></tr>
<tr><td><i>Pre-test</i></td><td>"When I increase the amount of CO_2 the earth heats up much faster than it would if the setting was lower."</td></tr>
<tr><td><i>Post-test</i></td><td>"To come up with these values I first tried putting the number of particles and the container size at its max. After that, I tried the number of particles at its minimum and the container size at its maximum."</td></tr>
<tr><td colspan="2"><i>Attending to explanatory factors</i>
Response provides some explanation for the relationship between system parameters and macro-level patterns. Explanations such as this convey the students' reasoning and suggest that they are not only attending to cause and effect, but that they are going one step further and trying to make sense of the relationship between cause and effect—a fundamental activity of science.</td></tr>
<tr><td><i>Pre-test</i></td><td>"The carbon dioxide blocks the IR from reaching the sky but doesn't stop the sunlight from reaching the ground the higher you increase the Carbon Dioxide."</td></tr>
<tr><td><i>Post-test</i></td><td>"A bigger area and less particles shouldn't produce a large amount of pressure since it's a lot of space for the particles."</td></tr>
<tr><td colspan="2"><i>Attending to parameter-system relationships</i>
Response describes relationship between system parameters and macro-level patterns. It is important to attend to outcomes of the simulation when tinkering with or testing parameters, in order to notice relationships between cause and effect. Simple qualitative characterizations of the relationships within a system are a foundation for constructing more detailed or mathematical relationships. A simple qualitative understanding of a cause-effect relationship can be a powerful tool for reasoning about system dynamics and for conveying the big ideas about the relationships within a system to others. In the scientific world, these "others" might be collaborators or members of the scientific community at-large.</td></tr>
<tr><td><i>Pre-test</i></td><td>"The temperature increases."</td></tr>
<tr><td><i>Post-test</i></td><td>"I slid the wall-position to its maximum and the number of particles to its minimum."</td></tr>
</table>

question: "All computational simulations are only approximations of reality. What are some of the simplifications of this simulation that make it different from the real world?"

We examined students' pre- and post-test responses, sorting responses into categories based on similarities that were relevant to the learning objective we were analyzing. Five categories emerged that were representative of response types across both pre- and post-test. These are *attending to general issues, attending to rep-*

resentational issues, attending to issues of controllability, attending to issues of completeness, and attending to procedural limitations. We identified these as five competencies relevant to *identifying simplifications made by a model*. These competencies are arranged in order of increasing sophistication, described and illustrated with examples from the data in Table 7.2.

General comments about accuracy and representational limitations seemed to be the easiest to make with attention to mere surface-features. Responses that identified these simplifications were therefore awarded the lowest score (one point). The completeness of the model and control given to its various parameters seemed to require more careful consideration of the interface and comparison with the real-world. Responses were therefore awarded a slightly higher score (two points) for identi-

Table 7.2 Pre- and post-test rubric for analyzing students' responses and characterizing the competencies they drew upon when *identifying simplifications made by a model*

Attending to general issues—score: 1 Response refers to general, rather than specific, inaccuracies or missing factors. This suggests that students understand that the model is not an accurate depiction of reality, however they have not done the cognitive work of identifying a particular limitation.	
Pre-test	"In reality, other factors could come into play rather than just CO_2 and clouds."
Post-test	"Inaccuracy in particles and wall position can make it different from the real world."
Attending to representational issues—score: 1 Response refers to representational limitations of the model. This suggests that students understand that the model is not an accurate depiction of reality. This is not a "meaningful" limitation compared to other limitations that students mentioned, as the simplification does not influence the interactions between the elements of the model and therefore does not influence the outcome of any given simulation trial.	
Pre-test	"Obviously, sunlight is not a bunch of little sticks raining down."
Post-test	"It's not actually life size."
Attending to issues of controllability—score: 2 Response refers to the existence of control over factors in the model that one does not have control over in real life. This suggests that students understand the model is different from reality because it allows certain conditions to be tested by being varied, which is impossible to do in reality.	
Pre-test	"Because you can control how much CO_2 and cloud coverage there is."
Post-test	"In real life, you cannot add or subtract molecules nor can you adjust the wall positioning."
Attending to issues of completeness—score: 2 Response refers to specific elements or factors that are missing from, or extraneous to, the model. These students recognize that a model is an approximation of reality. They have compared it with the real world and identified factors that are found in the real world but missing from the model. It is probable they believe these factors are somehow important to the model and would change the outcome of a simulation trial. Limitations such as these are important for scientists to identify, because they help them interpret their results and recognize their limitations.	

(continued)

Table 7.2 (continued)

Pre-test	"There are humans on earth and humans also can add to the amount of heat."
Post-test	"The real world, does not have this many boundaries and an infinite number of particles."
Attending to procedural limitations—score: 3 Response refers to interactions, behaviors, or relationships within the model that differ from real life. Limitations such as this are extremely important for scientists to recognize, as they are related to how successful the model is at approximating reality. Procedural limitations of the model influence the outcome of a simulation run in an important way: if the simulation does not reproduce patterns found in real-world data, something about the encoded theoretical model is *wrong* and needs to be revised.	
Pre-test	"CO_2 might not speed up that much when it absorbs IR light."
Post-test	"Particles don't travel in and out of the room in this simulation, when in real life they do."

fying these simplifications. Finally, comments about the procedural correctness of behavior and interactions within the model required students to run the model and track cause and effect relationships between elements at the micro-level and compare this with scientific laws or theories. Responses were therefore awarded the highest score (three points) for these simplifications. Responses that were not coded for any of the three competencies were given a score of zero. For our statistical analysis, we counted the point-value of the highest competence demonstrated in a student's response. Scores ranged from 0 to 3. We characterize the distribution of competencies (demonstrated in both pre- and post-test) in our findings section.

7.4 Findings

To test whether the computational biology units played a role in developing competencies for *modeling and simulation practices*, pre- and post-test scores for the two items were compared using a Wilcoxon signed-rank test and competence frequencies were compared using McNemar's tests. We report the results of our analysis below.

7.4.1 Learning Objective 1: Explore a Model by Changing Parameters

The class average for the pre-test item measuring students' ability to *explore a model by changing parameters in the interface or code* was a score of 1.24. The average for the post-test item was a score of 1.46. The *p*-value obtained using a paired Wilcoxon signed-rank test (with continuity correction) was 0.01389 ($V = 1175.5$). The difference in student scores is therefore statistically significant at the 5% level,

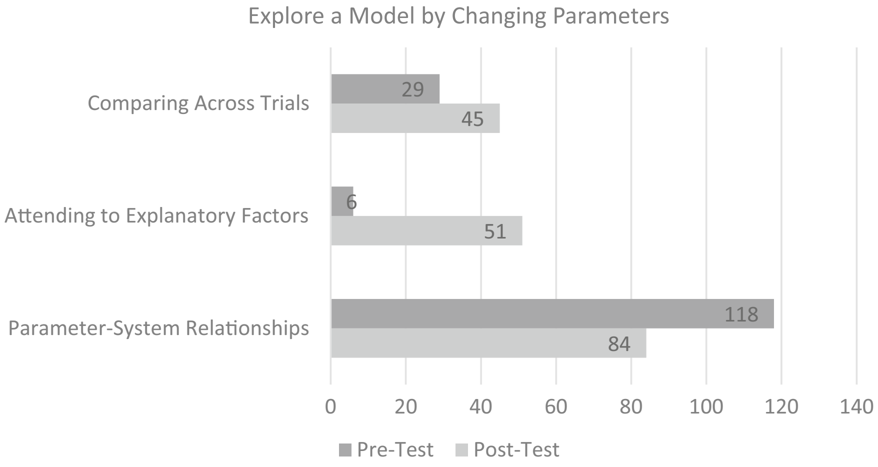


Fig. 7.4 Frequencies of competencies demonstrated in students' responses to the pre- and post-test items assessing their mastery of learning objective 1

which supports the claim that engagement in our curriculum helped students improve with regards to this learning objective. To gain a more nuanced understanding of how students developed their abilities to *explore a model*, we compared the frequencies of competencies they demonstrated in pre- and post-test responses. The bar chart (Fig. 7.4) illustrates the number of students *comparing across trials*, *attending to explanatory factors*, and *attending to parameter-system relationships*, on both the pre- and post-test.

Notably, the frequencies increased from pre- to post-test for *comparing across trials* and *attending to explanatory factors*. Frequencies decreased for *attending to parameter-system relationships*. Below, we present results of statistical analyses that show whether these changes in frequency may have been the result of students' participation in our computational biology units.

7.4.1.1 Comparing Across Trials

An increase in *comparing simulation results across multiple trials* suggests students have adopted a more systematic approach or learned the value of tinkering in exploration. An example of a student response that suggests a student is comparing simulation results across multiple trials is "When I increase the amount of CO_2 the earth heats up much faster than it would if the setting was lower." A McNemar's test on our counts (without continuity correction) results in a test statistic (χ^2) of 4.2667 and a *p*-value of 0.03887, which is a significant effect. This is evidence that engagement in our computational biology curriculum improved students' abilities to *explore a model* by encouraging more students to compare results across multiple trials.

7.4.1.2 Attending to Explanatory Factors

An increase in *attending to explanatory factors* suggests more students are drawing inferences from the model visualization to understand the mechanisms that produce system behavior. An example of a student response that suggests a student is attending to explanatory factors is “The carbon dioxide blocks the IR from reaching the sky but doesn’t stop the sunlight from reaching the ground the higher you increase the Carbon Dioxide.” A McNemar’s test (with continuity correction) results in a test statistic (χ^2) of 37.961 and a p-value of less than 0.001, which is a significant effect. This is evidence that engagement in our computational biology curriculum improved students’ abilities to *explore a model* by encouraging them to attend to explanatory factors.

7.4.1.3 Attending to Parameter-System Relationships

A decrease in *attending to parameter-system relationships* suggests fewer students are articulating inferences from the model visualization about the macro-level relationships between parameter settings and system behavior. An example of a student response that suggests a student is attending to parameter-system relationships is “The temperature increases,” when the student is asked to describe what happens to the environment as atmospheric CO_2 increases. A McNemar’s test (without continuity correction) results in a test statistic (χ^2) of 27.524 and a p-value of less than 0.001, which is a significant effect. This decrease may be the result of a difference in the form of pre- and post-test items. While the pre-test item asked students *only* what would happen to a macroscopic characteristic of the system as a parameter was varied, the post-test item asked students for *both* the parameter value that resulted in a particular system characteristic and the steps they took to find that value. It is possible that the additional question caused students to shift their attention away from answering the question focused on parameter-system relationships.

Our analysis of the pre/post changes in competence frequencies suggests that the students are improving with regard to purposeful exploration of the model by comparing more than one simulation trial and attending to explanatory factors. This suggests they began to look more closely at the model and to understand the interactions at the micro-level that explained the macro-level phenomenon.

7.4.2 Learning Objective 2: Identify Simplifications Made by a Model

The class average for the pre-test item measuring students’ ability to *identify simplifications made by a model* was a score of 1.39. Their average post-test score was 1.63. The p-value obtained using the Wilcoxon signed-rank test was 0.02 (V = 647.5). The

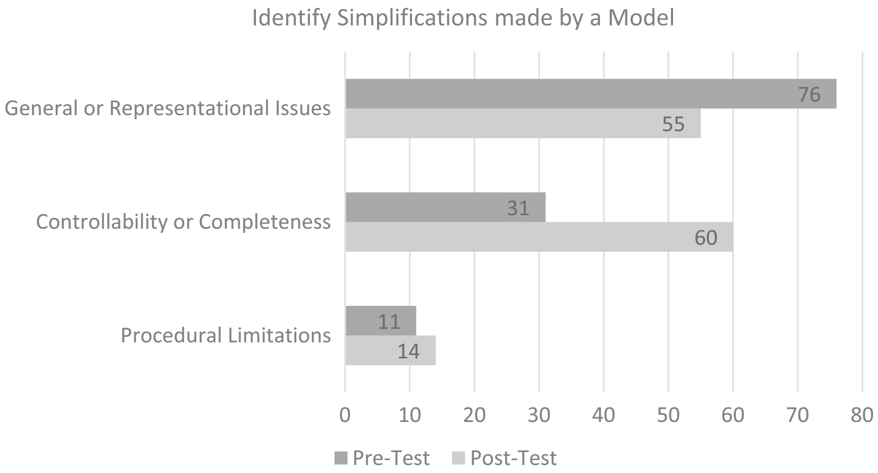


Fig. 7.5 Frequencies of competencies demonstrated in students' responses to the pre- and post-test items assessing their mastery of learning objective 2

difference in student scores is therefore statistically significant (at the 5% significance level) and this supports our claim that engagement in our curriculum helped students improve with regards to this learning objective. To gain a more nuanced understanding of how students developed their abilities to *identify the simplifications made by a model*, we compared the frequencies of competencies they demonstrated in pre- and post-test responses. For ease of coding, we combined competencies of the same score. This is reflected in the bar chart (Fig. 7.5), which illustrates the number of students noting *general or representational limitations*, *controllability or completeness limitations*, and *procedural limitations*, on both pre and post-test.

The frequency decreased from pre- to post-test for *attending to general or representational limitations* and increased from pre- to post-test for both *attending to limitations of controllability or completeness* and *attending to procedural limitations*. Below, we present results of statistical analyses that show whether these changes in frequency may have been the result of students' participation in our computational biology units.

7.4.2.1 Attending to General or Representational Issues

A decrease in *attending to general or representational limitations* of the model suggests fewer students are distracted by surface-level limitations that are inconsequential to the model's ability to simulate reality. An example of a student response that attends to general or representational limitations is "Obviously, sunlight is not a bunch of little sticks raining down." A McNemar's test (without continuity correction) results in a test statistic (χ^2) of 9 and a p-value of 0.0027, which is a significant effect. This is evidence that engagement in our computational biology curriculum improved students' abilities to *assess a model* by encouraging them to pay less attention to limitations with the model's surface features.

7.4.2.2 Attending to Limitations of Controllability or Completeness

An increase in *attending to limitations of controllability or completeness* suggests students have begun to pay attention to limitations with elements of the model's deeper structure. An example of a student response that attends to limitations of controllability or completeness in a model is "In real life, you cannot add or subtract molecules nor can you adjust the wall positioning." A McNemar's test on our counts (without continuity correction) results in a test statistic (χ^2) of 15.868 and a p-value of less than 0.001, which is a significant effect. This is evidence that engagement in our computational biology curriculum improved students' abilities to *assess a model* by encouraging them to attend to deeper model limitations, such as controllability and completeness limitations.

7.4.2.3 Attending to Procedural Limitations

An increase in *attending to procedural limitations* of the model suggests more students have begun to pay attention to elements of the model's deeper structure. An example of a student response that attends to procedural limitations is "CO_2 might not speed up that much when it absorbs IR light." A McNemar's test (without continuity correction) results in a test statistic (χ^2) of 0.42857 and a p-value of 0.5127, which is not a significant effect. Though we do see an increase in the number of students enacting this practice, there is not enough evidence to say that engagement in our computational biology curriculum improved students' abilities to assess the procedural limitations of a computational model.

Our analysis of the pre/post changes in competence frequencies suggests that students developed expertise in *assessing a model* by shifting their attention from limitations with the surface features of the model to limitations with elements of its deeper structure. More specifically, students shifted from identifying general limitations or limitations with the visual representation to limitations with a model's controllability and completeness.

7.5 Discussion

We have presented findings from a quantitative analysis of 133 9th grade students' written responses to assessments given before and after their participation in three computational biology units. Our results suggest that our curriculum helped students develop a number of important competencies for *exploring a model by changing* *parameters in the interface or code*, such as *comparing simulation results across multiple trials* and moving beyond merely describing relationships between a parameter and system behavior, to *attending to explanatory factors* in the model. Our results also suggest that students developed important competencies for *identifying simplifications made by a model*, such as shifting attention from *general and representational limitations* with the model to deeper limitations such as *model completeness and controllability*. While our results are encouraging, we can't rule out the possibility that limitations of our experimental design (such as asymmetries between pre- and post-test items discussed earlier) may have influenced our findings.

Our work is concerned with characterizing students' engagement in computational thinking practices in their secondary science classrooms. It is therefore in conversation with scholarship on the nature of computational thinking and the nature of computational thinking in STEM. Previously, we created a taxonomy of computational thinking practices used by experts in computational STEM disciplines. The findings presented here provide insight into how students can develop expertise with respect to *modeling and simulation practices* by characterizing, at a fine grain-size, the competencies students draw upon when *exploring a model by changing its parameters in the interface or code* and *identifying simplifications made by a model*. Our research program continues to uncover the space of competencies relevant to CT-STEM practices representing all strands of our taxonomy and investigate how these competencies can be developed through engagement with our computationally-enriched science curriculum. In future work, we aim to connect our quantitative treatment with qualitative analysis of student utterances, NetLogo log files, and work.

While the units investigated by this study featured NetLogo, other CT-STEM units (which have been created as part of a larger curricular design effort) feature modeling environments such as Molecular Workbench (Concord Consortium, 2010) and PhET (Perkins et al., 2006). Other units introduce students to computational tools for data analysis and problem solving, such as CoDAP (Finzer, 2016). Exposing students to a diverse range of computational tools is meant to help them develop a flexible set of CT-STEM practices.

In addition to understanding how our curriculum can support students' development of CT-STEM practices, our research aims to understand how engagement in these practices can support students' science content learning. Research already points to the productivity of computational tools for science learning (Guzdial, 1994; National Research Council, 2011; Redish & Wilson, 1993; Repenning, Webb, & Ioannidou, 2010; Sengupta, Kinnebrew, Basu, Biswas, & Clark, 2013; Sherin, 2001; Taub, Armoni, Bagno, & Ben-Ari, 2015; Wilensky & Reisman, 2006). As described by restructuration theory, the representational form of knowledge influences how it can be understood. The advance of computational tools has afforded representations that have had profound influence on the way scientists understand phenomena. We argue that these same tools can also be employed in science learning to make complex content more accessible to students, while at the same time broadening engagement with computational thinking.

Acknowledgements We acknowledge, with thanks, the permission of the International Conference on Computational Thinking Education 2017 (CTE 2017) to allow us to extend our previous work, "Swanson, H., Anton, G., Bain, C., Horn, M., & Wilensky, U. (2017). Computational Thinking in the Science Classroom. In S. C. Kong, J. Sheldon, & K. Y. Li (Eds.), *Conference Proceedings of International Conference on Computational Thinking Education 2017*. Hong Kong: The Education University of Hong Kong", in preparing this book chapter. This work was supported by the Spencer Foundation and the National Science Foundation (CNS-1138461, CNS 1441041, DRL-1640201).

References

Blikstein, P., & Wilensky, U. (2009). An atom is known by the company it keeps: A constructionist learning environment for materials science using agent-based modeling. *International Journal of Computers for Mathematical Learning, 14*(2), 81–119.

Brennan, K., & Resnick, M. (2012, April). New frameworks for studying and assessing the development of computational thinking. In *Proceedings of the 2012 Annual Meeting of the American Educational Research Association, Vancouver, Canada* (pp. 1–25).

Collins, A., Joseph, D., & Bielaczyc, K. (2004). Design research: Theoretical and methodological issues. *Journal of the learning sciences, 13*(1), 15–42.

Concord Consortium. (2010). Molecular workbench. *Java simulations and modeling tools (2004–2013).*

diSessa, A. A. (2001). *Changing minds: Computers, learning, and literacy*. MIT Press.

Finzer, W. (2016). Common online data analysis platform (CODAP). Emeryville, CA: The Concord Consortium. [Online: concord. org/codap].

Goody, J. (1977). *The domestication of the savage mind*. New York: Cambridge University Press.

Guzdial, M. (1994). Software-realized scaffolding to facilitate programming for science learning. *Interactive Learning Environments, 4*(1), 001–044.

Horn, M. S., & Wilensky, U. (2012). NetTango: A mash-up of NetLogo and Tern. In *AERA 2012*.

Horn, M. S., Brady, C., Hjorth, A., Wagh, A., & Wilensky, U. (2014, June). Frog pond: A code-first learning environment on evolution and natural selection. In *Proceedings of the 2014 Conference on Interaction Design and Children* (pp. 357–360). ACM.

Kaczmarczyk, L., & Dopplick, R. (2014). *Rebooting the pathway to success: Preparing students for computing workforce needs in the United States*. Education Policy Committee, Association for Computing Machinery.

Levy, F., & Murnane, R. (2004). *The new division of labor: How computers are creating the new job market*. Princeton, NJ: Princeton University Press.

Levy, S. T., & Wilensky, U. (2009). Students' learning with the connected chemistry (CC1) curriculum: Navigating the complexities of the particulate world. *Journal of Science Education and Technology, 18*(3), 243–254.

Margolis, J. (2008). *Stuck in the shallow end: Education, race, and computing*. Cambridge: The MIT Press.

Margolis, J., & Fisher, A. (2003). *Unlocking the clubhouse: Women in computing*. Cambridge: The MIT Press.

National Research Council. (2010). *Report of a workshop on the scope and nature of computational thinking*. Washington, DC: The National Academies Press.

National Research Council. (2011). *Learning science through computer games and simulations*. Washington, DC: The National Academies Press.

Novak, M., & Wilensky, U. (2011). *NetLogo fish tank genetic drift model*. Northwestern University, Evanston, IL: Center for Connected Learning and Computer-Based Modeling.

Olson, D. R. (1994). *The world on paper*. New York: Cambridge University Press.

Papert, S. (1980). *Mindstorms: Children, computers, and powerful ideas*. New York, NY: Basic Books Inc.

Perkins, K., Adams, W., Dubson, M., Finkelstein, N., Reid, S., Wieman, C., et al. (2006). PhET: Interactive simulations for teaching and learning physics. *The Physics Teacher, 44*(1), 18–23.

Quinn, H., Schweingruber, H., & Keller, T. (Eds.). (2012). *A framework for K-12 science education: Practices, crosscutting concepts, and core ideas*. National Academies Press.

Redish, E. F., & Wilson, J. M. (1993). Student programming in the introductory physics course: MUPPET. *American Journal of Physics, 61,* 222–232.

Repenning, A., Webb, D., & Ioannidou, A. (2010). Scalable game design and the development of a checklist for getting computational thinking into public schools. In *Proceedings of the 41st ACM Technical Symposium on Computer Science Education* (pp. 265–269).

Sengupta, P., & Wilensky, U. (2009). Learning electricity with NIELS: Thinking with electrons and thinking in levels. *International Journal of Computers for Mathematical Learning, 14*(1), 21–50.

Sengupta, P., Kinnebrew, J. S., Basu, S., Biswas, G., & Clark, D. (2013). Integrating computational thinking with K-12 science education using agent-based computation: A theoretical framework. *Education and Information Technologies, 18*(2), 351–380.

Sherin, B. L. (2001). A comparison of programming languages and algebraic notation as expressive languages for physics. *International Journal of Computers for Mathematical Learning, 6*(1), 1–61.

Taub, R., Armoni, M., Bagno, E., & Ben-Ari, M. (2015). The effect of computer science on physics learning in a computational science environment. *Computing Education, 87,* 10–23.

Tinker, R., & Wilensky, U. (2007). *NetLogo Climate Change model*. Northwestern University, Evanston, IL: Center for Connected Learning and Computer-Based Modeling.

Weintrop, D., Beheshti, E., Horn, M., Orton, K., Jona, K., Trouille, L., et al. (2016). Defining computational thinking for mathematics and science classrooms. *Journal of Science Education and Technology, 25*(1), 127–147.

Weintrop, D., Beheshti, E., Horn, M. S., Orton, K., Trouille, L., Jona, K., & Wilensky, U. (2014). Interactive assessment tools for computational thinking in high school STEM classrooms. In D. Reidsma, I. Choi, & R. Bargar (Eds.), *Proceedings of Intelligent Technologies for Interactive Entertainment: 6th International Conference, INTETAIN 2014, Chicago, IL, USA* (pp. 22–25). Springer International Publishing.

Wilensky, U. (1997a). *NetLogo GasLab gas in a box model*. Northwestern University, Evanston, IL: Center for Connected Learning and Computer-Based Modeling. http://ccl.northwestern.edu/netlogo/models/GasLabGasinaBox.

Wilensky, U. (1997b). *NetLogo wolf sheep predation model*. Northwestern University, Evanston, IL: Center for Connected Learning and Computer-Based Modeling. http://ccl.northwestern.edu/netlogo/models/WolfSheepPredation.

Wilensky, U. (1997c). *NetLogo AIDS model*. Northwestern University, Evanston, IL: Center for Connected Learning and Computer-Based Modeling. http://ccl.northwestern.edu/netlogo/models/AIDS.

Wilensky, U. (1999). *NetLogo*. Northwestern University, Evanston, IL: Center for Connected Learning and Computer-Based Modeling. http://ccl.northwestern.edu/netlogo/.

Wilensky, U. (2001). Modeling nature's emergent patterns with multi-agent languages. In *Proceedings of EuroLogo* (pp. 1–6).

Wilensky, U. (2003). Statistical mechanics for secondary school: The GasLab multi-agent modeling toolkit. *International Journal of Computers for Mathematical Learning, 8*(1), 1–41.

Wilensky, U., Brady, C. E., & Horn, M. S. (2014). Fostering computational literacy in science classrooms. *Communications of the ACM, 57*(8), 24–28.

Wilensky, U., Novak, M., & Levy S. T. (2005). *NetLogo connected chemistry 6 volume and pressure model*. Northwestern University, Evanston, IL: Center for Connected Learning and Computer-Based Modeling.

Wilensky, U., & Papert, S. (2010). Restructurations: Reformulations of knowledge disciplines through new representational forms. In J. Clayson & I. Kalas (Eds.), *Proceedings of the Constructionism 2010 Conference*. Paris, France, 10–14 Aug 2010 (p. 97).

Wilensky, U., & Reisman, K. (2006). Thinking like a wolf, a sheep, or a firefly: Learning biology through constructing and testing computational theories—An embodied modeling approach. *Cognition and instruction, 24*(2), 171–209.

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