Science & Education (2020) 29:1035–1077
https://doi.org/10.1007/s11191-020-00141-7

SI: NATURE OF STEM

Understanding Biological Evolution Through Computational Thinking

a K-12 Learning Progression

Dana Christensen[1] · Doug Lombardi[2]

Published online: 22 July 2020

Abstract
Computational thinking is a contemporary science and engineering practice that has been introduced to the US science classrooms due to its emphasis in the *Next Generation Science Standards* (NGSS). However, including computational thinking into science instruction may be challenging. Therefore, for biological evolution (an essential theory within biology that spans across temporal and organizational scales), we recommend integrating computational thinking into evolution teaching to overcome misconceptions, reinforce the nature of science (NOS), and allow student embodiment (as students become emerged in their models, i.e., personification). We present a learning progression, which outlines biological evolution learning coupled with computational thinking. The defined components of computational thinking (input, integration, output, and feedback) are integrated with biology student roles. The complex nature of both teaching computational thinking and biological evolution lends toward a learning progression that identifies instructional context, computational product, and computational process and spans from simple to complex. Two major themes of biological evolution, unity and diversity have each been paired with both computational thinking and specific corresponding NGSS standards at levels of increasing complexity. There are virtually no previous studies which relate computation and evolution across scales, which paves the way for questions of importance, support, benefits, and overall student achievement in relation to the advancement of science in education.

1 Introduction

Biological science is a rich domain that may facilitate students' exploration of the natural living world. Currently, in the USA, many educators are developing, implementing, and

✉ Dana Christensen
tug25828@temple.edu

[1] Temple University, Philadelphia, PA, USA

[2] University of Maryland, College Park, MD, USA

revising biology courses (a standard life science course) in a manner that is heavily influenced by the *Next Generation Science Standards* (NGSS Lead States 2013). The three-dimensional framework underlying the NGSS suggests to optimize science learning via classroom instruction that integrates scientific practices, disciplinary core ideas, and crosscutting concepts (National Research Council 2012). Further, this three-dimensional framework seems to be impacting how biology is taught throughout the world (Lederman 2019). However, integrating even two of these dimensions, such as computational thinking (a scientific practice) and biological evolution: unity and diversity (a disciplinary core idea), is quite challenging, in part due to a lack of theoretical grounding (Weintrop et al. 2016). In this paper, we present a learning progression that integrates computational thinking and biological evolution to facilitate students' conceptual learning of fundamental scientific content and nature of science (NOS). Although we present this progression through the perspective of three-dimensional learning specified in the NGSS, we argue that the integration of computational thinking and biological evolution is practical and beneficial for science learning within a broader worldwide context.

The theory of evolution—which provides the scientific explanation for the mechanisms that drive living populations, and in turn, species of living organisms to change over time through the forces of natural selection—informs every aspect of modern biology (Campbell et al. 2000; McCain and Kampourakis 2018). Many famous biologists agree with Dobzhansky's (1973) claim that "Nothing in biology makes sense except in the light of evolution." As students master foundational evolutionary ideas (e.g., natural selection, the concept that inheritable traits that help an organism survive and reproduce within its environment become more common in a population over time), they come to a deeper understanding of "the how" and "the why" of the processes that drive life's changes over various time and organizational scales. As students (and scientists) develop richer knowledge of evolution, they can hypothesize about biological topics to a greater extent (e.g., how species are related, inherent human behaviors, various aspects of molecular genetics, and ecosystem dynamics; Griffith and Brem 2004).

Many educators, researchers, and institutions have developed a variety of learning tools, practices, and interventions to facilitate student learning of biological evolution. For example, the Concord Consortium (2018) lists eleven current and three former research projects associated with students' learning about evolution. In order to achieve scientific literacy (encouraging science learning as praxis by allowing students to identify solutions that transform practical relevant scientific problems; Levinson 2010; Aikenhead 2007) by graduation, the National Research Council (2012) recommends that computational thinking should be integrated into life science courses through the K-12 curriculum. In order to be effective, computation should be infused into curricular materials using the theory that blends computational, scientific, and educational perspectives (Sengupta et al. 2013). This is especially important because modern science, particularly biology, is becoming increasingly computational in nature (e.g., bioinformatics, data analytics, genomics, ecosystem modeling) and technological change (in general) is exponential. Because using computational thinking is inherently cross-curricular, it allows students to identify and solve technological problems which have and will continue to arise within democratic societies (Shen 1975).

Computational thinking is essential for biologists (and students) to better understand life processes and systems because evolution spans across several temporal and organizational scales (Guo et al. 2016). Conversely, some have speculated that deep understanding of computational thinking can occur through the perspective of evolution because both biological life and computer code emerge through and are guided by evolutionary progressions (Toffoli 2004). For example, students may need to explore the properties of individual agents such as cells that make up tissues and eventually organs and organ systems within organisms. As students assign properties to cells using computational tools (i.e., coding), the emergent properties of organisms become more evident to students (because different levels of the biological organization may display different ontological properties; Chi et al. 2012). This level of understanding may occur through computation as students envision themselves within models they develop using these thought processes (i.e., embodiment, which also aligns with the NOS). In this way, emergent properties of biological evolution parallel emergent computation (and the tools required in learning emergent computation resemble those used by scientists in the field).

Computational thinking in science sits at the intersection of (a) scientific disciplinary knowledge (e.g., biological knowledge), and (b) computation and (c) mathematics as constructs for problem-solving (Denning 2017). Computational thinking is the thought process involved in formulating problems (e.g., algorithms) within a particular domain (e.g., biology), such that solutions (e.g., representations) may be presented through the most effective steps (Aho 2012). However, this notion of computational thinking is sufficiently vague for application to instruction by typical educators. Further, computational thinking neither has a clear definition nor explicit methods for inclusion in science learning (Wing 2006), even though the NGSS framework encourages computational thinking and provides educators the flexibility to design engaging learning experiences for students (i.e., emphasizing the development of subject-specific skills alongside NOS practices as facilitated through computational thinking; National Research Council 2012). Therefore, this paper presents a learning progression integrating computational thinking (i.e., a common practice employed by biologists) alongside evolutionary concepts.

Our proposed learning progression represents three aspects of computation that should be present in classroom environments: (1) the computational context (as provided by instructor), (2) computational product (as produced by a student), and (3) the computational process (the actual act of student development of and with the computational components). The student's computational process as facilitated by the instructor includes students' reasoning about and implementation of four computational components: input, integration, output, and feedback (as modified from Weintrop et al. 2016). This process may allow students to predict phenomena at a variety of biological levels, resulting in a higher level of cognitive engagement, and thereby deepening students' understanding of evolution. In parallel, as scientists solve problems (in conjunction with our computational components), it requires thinking about their data (input), the relationship of the variables (integration), the results (output), and re-modeling their original assessment (feedback). Computational thinking in the way we have defined it (within the context of education) fully encompasses this scientific process through the use of the computational components (further aligning it with NOS practices).

We also considered educational theory in the development of our learning progression. As students work actively through using computational thinking in cooperation with their classmates, understanding can be brought from a social context to the individual (Vygotsky 1962). In this learning process, students transform and rationalize new information based on their prior knowledge and internalized mental schema (Piaget 1976). This combination of social and cognitive constructivist philosophies outlines an active process of shared and individual knowledge construction that may promote deep life science learning through computational thinking (Basawapatna et al. 2013; Fisher et al. 2016; Guo et al. 2016; Wing 2006).

We have aligned our learning progression with an eye toward international relevance. Although we developed this learning through the perspective of the NGSS, in part, we noted that many countries' science standards parallel the NGSS (Moore et al. 2015). For example, such standards promote approaches and methods that encourage students to ask questions, evaluate evidence, and justify the validity of their own ideas, as well as the ideas of others (National Research Council 2010). Studies indicate that incorporation of scientific practices (e.g., computational thinking) is common to science courses in many countries, with considerable attention also given to closely related NOS concepts (Research Council 2012). Further, the International Society for Technology Education claimed that global standards are quite similar to the NGSS in emphasizing computational thinking (and related concepts), as well as embedding computation across disciplines (Grover and Pea 2013). The NGSS also added research-derived benchmarks that align the standards with international tests (e.g., Canada, England, Finland, Hong Kong, Hungary, Ireland, Japan, Singapore South Korea, and Taiwan; Moore et al. 2015).

Thus far, we have introduced the alignment between evolution learning and computational thinking, and how such learning may be facilitated through a learning progression. We have identified the importance (and global relevance) of computational thinking within the NGSS as supported by educational theory. We now delve deeper into each of the aspects, and specifically, discuss our theoretical framework that integrates (a) relevant challenges of biological evolution education (conceptual change, misconceptions, social and emotional implications, and its relationship to the NOS and NGSS), with (b) distinct characteristics of computational thinking, and (c) illustrate their relationships to the NGSS and biology learning. We then present, in some detail, a proposed learning progression for biological evolution as supported by computational thinking, which emerged from our theoretical framework. The proposed learning progression describes a transition from simple to complex and includes instructional contexts, computational process, and computational products that would be displayed and practiced by students within classrooms. Two major themes of biological evolution have been identified as unity and diversity and have each been paired with computational thinking processes and specific NGSS standards. The final section concludes the paper by highlighting gaps in our current understanding of how learning about biological evolution may be facilitated by computational thinking. Addressing these research gaps has the potential to direct future researchers and educators toward fully fleshing out a more robust and effective learning progression through the development of specific student tasks, assignments, and assessment tools. Further, we suggest that practitioners use it to incorporate computation into lessons they already teach as a practical starting point.

2 Theoretical Framework

2.1 Learning about Biological Evolution

As individuals become scientifically literate, they tend to use their scientific knowledge to holistically shape their worldview (Stocum 2015). As this type of thinking becomes the norm, it may cause societal shifts in scientific thinking. For example, evolutionary biologists are frequently called upon to bridge gaps between various disciplines such as biology, medicine, and psychology because evolutionary concepts provide interdisciplinary explanations (Antolin et al. 2012). This is especially important at a time when students have access to a variety of competing information. Increased scientific literacy allows students to better navigate interdisciplinary scientific constructs of societal importance (including evolution and climate change; Sinatra and Lombardi 2020). Thus, more general reasoning and cognition required for scientific literacy may be learned through one topic (e.g., evolution) and transferred to another (e.g., climate change; Beggrow and Sbeglia 2019).

Having knowledge of biological evolution is required by citizens of a democracy to make informed long-term environmental decisions, to cultivate sustainable agriculture, to stay ahead of pathogenic diseases, to battle genetic conditions, and to make sense of human emotion among many other phenomena. Evolutionary principles lay the foundations for student understanding of modern medicine, sustainable conservation, and human psychology (Sinatra et al. 2003). Learning evolution urges students to think like scientists, recognize biological processes, and better grasp the dynamics of nature and the limits of science. Identifying how and why scientific inquiries and questions are important within the holistic domain of biology requires a genuine and authentic understanding of biological evolution (Alters and Nelson 2002).

Understanding biological evolution also allows students to better appreciate biology as a discipline as it can be observed at and between all levels of biological organization (Campbell et al. 2000). The two major themes of biological observations in a scientific study that have stimulated students and scientists at various levels of complexity are the exploration of the unity and diversity of life. In other words, how is there such a diversity of life on Earth and among all this diversity, how can the various similarities among organisms be explained? We know that organisms are related to each other but understanding exactly how and why may answer these two questions at various levels of biological organization and temporal scales (College Board 2009).

Teaching biology without ensuring proper student understanding of evolution may impede students' understanding of order and coherence that fosters a systematic understanding of life, as well as how these living systems interact with non-living systems that they encounter in other domains (e.g., energy systems encountered in physics and chemistry). Specifically, defining and understanding relations between variables within living and non-living systems is a common practice in biology learning. However, it is challenging for students to compare fundamental concepts in the physical sciences to biology, due in part, to the compounding interplay of variables that are involved in these systems (Guo et al. 2016). Computation may be one way to bridge the gap between biological evolution learning, variable relationships, and mathematics, similar to what also occurs in physical science classrooms (Gross 2004).

2.2 Misconceptions about Evolution and Conceptual Change

Conceptual change about biological evolution may be facilitated by computational thinking because of the greater potential for deep engagement (e.g., engagement in the content similar to practicing biologists; Dole and Sinatra 1998). This may be emphasized specifically by reasoning about individual biological elements at multiple levels of organization within biological systems (as prompted by computational thinking; Toffoli 2004). Biology students may have ideas about the nature of biological evolution, but these ideas may be unsophisticated and or incorrect. For example, Coley et al. (2017) reported that students' naive intuitions about biology persist from middle school to the university level, revealing little influence of high school biology on students' learning. Specifically, students persist in thinking about evolution in ways that are (a) teleological (i.e., causal reasoning in which a goal, purpose, function, or outcome of an event is taken as the cause of that event; Keil 2006), (b) essentialist (i.e., some unobservable essential property, such as an "underlying reality" or "true nature" conveys category identity and causes observable similarities among category members; Gelman 2003), and (c) anthropocentric (i.e., attribute human characteristics to non-human or inanimate objects; Gee 2013). As such, evolution is a difficult concept for students to learn and may require conceptual change (Sinatra et al. 2003).

Conceptual change has been of strong interest in science education for many decades and must occur in order for most students to learn and understand evolution due to common misconceptions, such as humans evolved from modern-day apes or the impossibility that complex life forms arose from very simple ones (Sinatra et al. 2008). A relatively early theoretical position on conceptual change was established by Posner et al. (1982), which later became known as the Conceptual Change Model (CCM; Pintrich et al. 1993). The CCM model incorporates Piaget's knowledge assimilation theory (1976), which states that when students encounter a new idea it must "fit into" what they already know. The CCM assumes that students have their own ideas on concepts (e.g., naive theories about biology and biological evolution) that are inconsistent with scientific understanding. Reconstruction of these naive theories may be facilitated through effective instruction (e.g., students' active reflection on their existing conceptions to help resolve the misconceptions they may have). The CCM posits that conceptual change occurs in four sequential steps: (a) students must be dissatisfied with their prior conception, (b) students must find the new concept intelligible and (c) plausible, and (d) students should see how the new concept would be beneficial (e.g., in opening new avenues of inquiry and expanding toward topics that hold student interest; Zirbel 2004). Some biology curricula, such as the modeling-based labs in advanced placement (AP) Biology (College Board 2012), use the CCM as a learning framework. AP courses are provided at the high school level and prepare students to take a subject-specific test in which they can sometimes earn college credit for the course which is dependent on performance.

Success in conceptual change requires appropriate teacher facilitation and sufficient cognitive activation experienced by students, both of which are heavily dependent on student's activities and interactions within a classroom (Duit and Tesch 2010). As students become involved with computation, they undergo unique learning experiences that allow for immersion within their computational models, this phenomenon is known as embodiment (which can facilitate transformative experiences; Jacobson and Wilensky 2006). In biology learning, embodiment occurs as students imagine and develop new ideas about how certain biological agents interact within their own constructed systems (Jacobson and Wilensky 2006). Students understand, imagine, and personify what it is like to be a biological entity (such as a cell) in order to properly develop the appropriate computational processes (e.g., setting up parameters for cells). As students become immersed within their computational processes (e.g., via modeling), the likelihood of conceptual change increases because embodiment makes evolution more plausible (believing it to be truthful), intelligible (knowing what it means), and fruitful (useful) (Vosniadou et al. 2008).

Students must grasp specific concepts at various biological levels in order to understand evolution. Frequently, these steps of understanding occur in a specific order, and these steps are naturally built into the NGSS. In order to understand natural selection (the driving force of evolution), students must also understand the concepts of variation and fitness of organisms, variance of individuals within a population, reproduction over time, and the concept of inheritance and heredity (Campbell et al. 2000). The typical steps in understanding natural selection are (1) spontaneous mutations variation, (2) a change in the environment, (3) individuals with suitable characteristics survive, (4) characteristics are inherited, and (5) frequency of these characteristics increases in the population over many generations (Brumby 1979).

Even many advanced biology students believe the common misconception that a change in the environment induces mutation which adapts individuals to changed conditions, and that these are the acquired characters are passed on (i.e., Lamarck's Theory of Inheritance of Acquired Characteristics, deemed incorrect through Darwin's research; Catley et al. 2005). In reality, mutation is spontaneous and most often random, and only sometimes results from explicit environmental change (e.g., radiation which is frequently detrimental to life). Additional common misconceptions include the distortion of the time scales of evolution or the belief that adaptation is a process that drives toward a positive end (Brumby 1979). Students may represent these concepts computationally in order to combat these misconceptions through proper development of their input, integration, and output, based on real biological facts and principles (e.g., representing randomness). This becomes especially helpful to students as they start to develop ways in which their model outputs can re-inform the next set of input based on phenomena that occur in nature (Chandrasekharan and Nersessian 2015).

Instructional practices associated with evolution should reinforce the ideas that many processes in the natural and physical world are open-ended and dynamic. Because evolution is an emergent process, it is not neatly bounded and ongoing without a clear start or end, ontological shifts in student thinking may be required for students to comprehensively understand the evolutionary theory (i.e., the misconception that evolution is a direct process as compared to the accepted perception that evolution is an emergent process; Chi et al. 2012). Interestingly, such an ontological shift toward evolution as an emergent process may be facilitated by computational thinking because (the student process of) computation itself is an emergent process (Berland and Wilensky 2015).

2.3 Social and Emotional Aspects of Evolution Learning

Unique ideas and thought processes that are just outside of students' comfort zones, while still maintaining a low level of frustration, are optimal for developing a manageable environment to confront and consider conceptions counter to naive theories (Vygotsky 1978). It is important to consider that learning about evolution affects students emotionally (Sinatra et al. 2003). Maintaining comfortable emotions and sustaining motivation is imperative in learning about evolution (Broughton et al. 2013). Using computational thinking in classrooms has the potential to promote positive experiences when learning about evolution (Ioannidou et al. 2011).

Cognitive dissonance may occur for students learning biological evolution if those close to them do not accept the notion or if evolution conflicts with religious beliefs. For example, students may hold a belief that as people accept the theory of evolution, they become increasingly racist and selfish, or that there is an inverse relationship between acceptance of evolution and personal spirituality (Griffith and Brem 2004). Some biology students may worry that the acceptance of evolution will diminish their sense of purpose and self-determination as they confront the idea that they may only be a mass of evolving neurons with no divine direction (Sinatra et al. 2003; Campbell et al. 2000). Not only do students have this worry, but it is also a major concern of biology teachers, which may lead to apprehension toward the subject matter (Griffith and Brem 2004).

There is a possibility that evolution learning fosters frustration because associated concepts are complex and/or abstract (Mead et al. 2018). Educators who have more knowledge of biology and scientific practices, and have also been exposed to scientific experiences (e.g., research projects) are less likely to have misconceptions and accept the theory (Nehm et al. 2009). Therefore, not only is it important for students and educators to maintain comfortable emotions while learning the subject matter, but also that they have the scientific aptitude (Mead et al. 2018).

In order for students to let go of old theories and accept new ones (e.g., about biological evolution), the new idea needs must be interconnected with other ideas they already have about the world (Piaget 1976). It is also beneficial for learning if their peers show an interest or belief in these new ideas (Dole and Sinatra 1998). Fostering a positive dialog about differences in cultural or religious beliefs coupled with emphasizing the nature of science may be one of the most effective instructional methods for teaching evolution, especially when it is controversial for students (Pobiner et al. 2018).

Computational thinking incorporates many of these aspects (positive emotions, motivation, personal relevance, and productive social context) and may provide a student learning experience that promotes a deep understanding of biological evolution through embodiment (Wilensky and Reisman 2006). Interestingly, standards of effective computational tools support positive learning experiences by encouraging (a) low threshold experiences, where students can quickly produce working code; (b) high ceiling experiences, where students can produce code that solves sophisticated problem solving; (c) proper scaffolding, where curriculum that sequentially builds students' skills and knowledge from low threshold to high ceiling code; (d) transferability, where students can apply their learned coding skills and knowledge outside the classroom context; (e) equity, where students use their coding skills and knowledge to promote greater accessibility; and (d) systemic and sustainable experiences, where students can apply their coding skills and knowledge into other academic domains (Ioannidou, Bennett, Repenning, Koh, & Basawapatna, 201 l).

2.4 Evolution and Teaching the Nature of Science (NOS)

A deeper understanding of the NOS as a systematic interplay between science, intellectual and cultural traditions, and contemporary issues may help students to better understand and accept the scientific validity of evolution (AAAS 1990; Nelson et al. 2019). Biology instruction also requires unique solutions and specific implementations tailored to address misconceptions while developing student understanding of the NOS. For example, students may need to engage in evaluating the connections between lines of evidence and alternative explanations that are both scientific (i.e., the theory of biological evolution) and non-scientific (e.g., intelligent design) as suggested by Heddy and Nadelson (2013). Such an evaluation may be facilitated through computational thinking (considering student use of and engagement with computational input, output, integration, and feedback) and could help to reconstruct common misconceptions about biological evolution, such as teleologicalism, essentialism, and anthropocentrism (Sinatra et al. 2008). It is reasonable to hypothesize that computation should strengthen student knowledge and NOS processes contributing to biology understanding, but it is unclear to what degree and in what ways (Gallagher et al. 2011).

Language of evolution is another challenge for students because some terms such as design, need, theory, and adaptation have everyday meanings, but are highly specific when learning about evolution. This language may contribute to misconceptions and are also linked to student understanding of the NOS (Sinatra et al. 2008). For example, the theory of evolution may be mistaken as an untested hunch due to everyday misuse of the word theory, rather than as an observable process that can be seen at all levels of biological organization from the micro- to the macro-biological levels. These concepts are not limited to biology learning, but more broadly associated with the NOS. Evolution is an example of a specific topic where confusion between everyday language and scientific language may contribute to students' misunderstanding. Therefore, computational thinking (e.g., using computation as a tool) may clear some of this confusion because students actively use NOS vocabulary as they construct knowledge.

NOS instruction also involves teaching about methodological principles of scientific knowledge, the nature of scientific understanding, and the limits of scientific knowledge. As students better understand scientific practices and modes of thinking, emotional demands may diminish for those who view evolution as contrary to their religious beliefs (Sinatra et al. 2008). Furthermore, engaging in scientific practices may foster open-mindedness and the idea that science involves the process of knowledge construction; science is neither an intrinsic source for answers nor categorically a contradiction to religious beliefs. Dole and Sinatra (1998) claimed that high levels of engagement required by complex activities such as inquiry, personal reflection, and justifying reasoning, encourages students to compare their beliefs to the content of evolution. This willingness to think deeply about complex problems may allow for questioning of personal beliefs (Sinatra et al. 2003). Thus, higher engagement with NOS and scientific inquiry via computational thinking may help students to practice methods facilitating their acceptance of the theory.

NGSS has chosen to embed the NOS within their three-dimensional framework of (a) scientific practices, (b) disciplinary core ideas, and (c) crosscutting concepts (Lederman and Lederman 2014). This framework posits that as students engage in scientific practices, they will develop a deep understanding of both scientific concepts and the NOS (NGSS Lead States 2013). The NGSS framework claimed that integrating scientific practices and content "will require substantial redesign of current and future curricula in order to provide increasingly sophisticated [science learning] experiences across grades" (p. 247). In the case of biological evolution, increasingly sophisticated learning experiences (such as computational thinking) could be used to reconstruct naive theories in a way that productively engages students. For example, because emergent properties associated with evolution provide difficulty for students (Chi et al. 2012), the learning tasks associated with biological evolution in classrooms should allow for exploration of these properties, while also exploring the relationship between evidence and explanations. Computational thinking is a novel way to engage students and would mimic how scientists gather and evaluate reliable evidence and construct valid explanations (Lederman and Lederman 2014).

2.5 NGSS: Biological Evolution as Unity and Diversity

In the USA, educational stakeholders formulated the *Next Generation Science Standards* (NGSS Lead States 2013) based on the science education framework released the year prior (National Research Council 2012). The NGSS integrates scientific practices, disciplinary core idea, and crosscutting concepts to form performance expectations (aka learning standards).

Many of these performance expectations grant flexibility for educators and include both computational thinking as a scientific practice and biological evolution as a disciplinary core idea. The NGSS threads in specific scientific practices in which evolutionary biologists engage, such as computational thinking, both of which can be used to explore life's unity and diversity.

There are a variety of NGSS performance expectations that may address by integrating biological unity and diversity with computational thinking (NGSS Lead States 2013). Educators already incorporate these standards into their classrooms using little (or very simple forms of) computation. Per the NGSS, elementary students are expected to develop appropriate NOS processes and simple biological principles that set them up for more advanced biological evolution principles. For example, by third-grade students should recognize life cycles and commonalities between life forms. In middle school, students are expected to analyze and interpret data for patterns in the fossil record that document existence, diversity, extinction, and the change of life forms through Earth's history (NGSS Performance Expectation [PE] MS-LS4-1). Students should explore this standard with the understanding that natural laws operate in the past the same way they do today (i.e., students are expected to apply these thought processes to historical evidence). Students should also be able to construct explanations based on evidence that describes the genetic variation of traits in a population. This includes those traits that increase an individual's probability of surviving and reproducing in a specific environment (NGSS PE MS-LS4-4). At the middle school level, students are also expected to gather and synthesize information about technologies that influence the way humans alter inheritance of desired traits in organisms (e.g., artificial selection or genetic engineering). Students should also be able to use mathematical representations to support explanations of how natural selection may lead to increases or decreases in specific traits over time (NGSS PE MS-LS4-6).

At the high school level, the NGSS (2013) prompts students to display and communicate appropriate information regarding common ancestry and that biological evolution is supported by many lines of evidence (NGSS PE HS-LS4-1). Students are expected to construct an explanation based on evidence that the process of evolution results from four factors: (a) the potential for a species to increase in number, (b) the heritable genetic variation of individuals in a species due to mutation and sexual reproduction, (c) the competition for limited resources, and (d) the proliferation of those organisms that are better able to survive and reproduce in their environment (NGSS PE HS-LS4-2). Statistics and probability should be used to support student explanations that organisms with advantageous heritable traits tend to increase in an environment as compared to organisms without the trait (NGSS PE HS-LS4-3). Students should be able to construct an explanation based on evidence for how natural selection leads to the adaptation of populations (NGSS PE HS-LS4-4). Students should also be able to evaluate evidence supporting the claims that changes in environmental conditions may result in (a) an increase in the number of individuals of some species, (b) the emergence of new species over time, and (c) the extinction of other species (NGSS PE HS-LS4-5). These biological evolution areas from the NGSS can be naturally explored using the NOS. The integration of computational thinking into these PEs may facilitate deeper engagement with more coherency about different grade levels.

3 Computational Thinking

Thoughts on computational thinking have progressed over time in terms of its definition, its necessity and role in education, and its relations to various scientific disciplines. Depending on the content area, specific reasoning topics or modes of thinking related to computational thinking have emerged. This is especially due to exponential growth in technology, which in turn, has contributed to theory (e.g., game theory and theoretical bioinformatics; Holyoak and Morrison 2012). Computational thinking for example has been associated with the rising field of computer science since the 1940s. Perlis, Simon, and Newell (1967) wondered if computational thinking would span across fields by the mid-1960s through integration within different domains; interestingly, this is one of the goals of the NGSS. For the purposes of our learning progression in terms of initial implication, it is suggested that educators consider current lessons they have and identify ways to incorporate computation. We suspect this assimilation process may become more complex over time as practitioners become more familiar with the constructs.

Certainly, this integration has not been fully realized and the question of how and why computational thinking spans across domains is still relevant. For example, should computation associated with the weather (e.g., atmospheric computation) be integrated within the context of meteorology or should computational thinking be taught as a domain general process? Some think of computational thinking solely within the specific context of designing models and developing the skills that are required for developing and designing software packages that are implemented by electronic machines (e.g., computers). Specifically, in the realm of education, it is thought that computational thinking will allow students to become better problem solvers in a digital world and across different disciplines (Denning 2017).

Newel et al. (1967) argued that computer science was a legitimate area of study because of the complex thought process required by humans (Denning and Freeman 2009). Thinking associated with computational analysis yields important empirical and theoretical results (Holyoak and Morrison 2012). Although there is an overlap between computer science and computational science, and there are distinct differences, computer science is specifically focused on engineering, theory, experimentation, design, and associated practices that are associated with computer technology. Although computational thinking may involve computers, the direct focus is using computation to explore scientific (and other) disciplinary problems (i.e., computational thinking may be viewed as a trifold intersection of a domain [science] knowledge, computational processes, and mathematical constructs; Aho 2012). Marr (1977) suggested that complex systems, such as computers and the human mind, have different levels of analysis: computational, algorithmic, and representational, with the computational level preceding the algorithmic level, which in turn precedes the representational level. Marr stressed the importance of the computational thinking as foundational. Aho (2012) similarly claimed that computational models are abstractions that are at the heart of computational thinking.

Today's teachers struggle with what computational thinking is, how it is assessed, and if it is appropriate for everyone to learn (Denning 2017). The relevant problems regarding computational thinking are (1) that there is no consistent definition for science educators (Selby and Woollard 2013), (2) that researchers and educators are unsure of most effective methods of implementation in science classrooms (Wilensky 2014), (3) and exactly if and how computational thinking is beneficial to science students (Speth et al. 2009).

3.1 Computational Thinking within the Learning and Teaching Process

Vygotsky (1978) claims that knowledge is constructed through the interactions with others and using tools (e.g., the multiplicity of technologies that surround us, including digital artifacts of the media world and specialized processes of the digital world; Gadanidis 2017). New technology is immersive in nature in that it supports, disrupts, and reorganizes human thinking and leaves humans to act and be acted upon based on actor-network theory (Latour 2005). Therefore, key elements of computational thinking can allow us to think-with and in the learning and the teaching processes (Gadanidis 2017).

With the exponential increase in technology, ethical concerns over fields such as bioinformatics, simulations (abstractions), and artificial intelligence (AI) are of interest to psychologists, mathematicians, computer scientists, biologists, cognitive scientists, and engineers. Moor's computer ethics and bioethics have not yet been considered together; however, the rise of bioinformatics alone warrants research in terms of the future of health care and other socioeconomic implications (Hongladarom 2006). Artificial intelligence already has a platform in formal education, as AI can act as a tutor, facilitate open online courses, and is responsible for many online searches. Here, we define AI as the study associated with developing computers and software capable of intelligent behavior. Agency and associated features of self-regulation and self-learning are key aspects of AI and synonymously, student agency is emphasized in computational thinking education environments. Today's students should explore the use, development, limits, future projections, and ethical concerns of AI.

Science education, artificial intelligence, and computational thinking each have three key elements which include agency, modeling, and abstraction (Gadanidis 2017) all of which are important contexts for classroom incorporation. Our trajectory of engagement with abstraction seems to intensify as a society, especially as digital code writes war machinery (e.g., weaponized drones), stock market transactions, robotics, and AI (Tamatea 2019). Abstraction may play a particularly important role in AI development, as the programmer "tells" the computer how to recognize certain features and set parameters based on certain elements and variables. Abstraction is also the key element of computational thinking because concepts are generally symbolic and represented (or manipulated) by code. Although Piaget claims that young children may not be capable of abstract thinking (Gadanidis 2017), abstraction may allow students to better manage complexity by reducing complex information and details. In other words, exposing students to (simple forms of) computational thinking at younger ages may allow for more effective learning later on (Gadanidis 2017).

Students must assimilate knowledge (i.e., domain specific content) into what they already know about computational processes per Piaget's knowledge assimilation theory (1976). Proper scaffolding of computational processes may facilitate a higher ceiling for students to abstract, automate, and dynamically model concepts, for both computational and content knowledge. For example, as students change computer code, they may simultaneously change parameters within their models. This allows students to manipulate and model information related to the content (e.g., students can "play with concepts" and bring them to life; Gadanidis 2017). In such a learning scenario, computational thinking includes (1) formulating problems for use with a computer to facilitate the solution; (2) logically organizing and analyzing data; (3) representing data through abstractions; (4) automating solutions through algorithmic processes; (5) identifying, analyzing, and implementing possible solutions as the most efficient and effective combination of steps and resources; and (6) generalizing and transferring this process to a variety of problem areas (Ioannidou, Bennett, Repenning, Koh, & Basawapatna, 2011). We have merged and emphasized these (computational) components in our learning progression.

Peel et al. (2019) conducted a study in which natural selection was learned through computational thinking using an "unplugged" design method, meaning students used hand-written "computational" explanations. Such thinking might be considered simple because computational tools were not considered. Students performed all components without technology. It was identified that abstractions associated with natural selection can be "extracted" from the computer, while still allowing students to emphasize the scientific process. This unplugged version was performed because students did not require access to a computer, nor did they need previous knowledge of coding. The results of the study indicated knowledge transfer of natural selection across contexts, and it reduced student misconception on the topic. It is reasonable to hypothesize that "re-plugging" and incorporating additional computational components in a similar study might result in greater knowledge gains.

3.2 Distinct Characteristics of Computational Thinking

Scientific thinking alone does not necessarily involve computational thinking. Computation is the descriptive word, computational tasks are the actual acts performed by students, and computational technologies are the (usually computer) programs that allow for students to perform computational thinking. Therefore, many aspects of scientific thought, such as creativity and critique, often fall outside of the idea of computational thinking during assessment (Grover and Pea 2013).

Computational thinking parallels computational science just as scientific thinking parallels science content. As shown in Table 1, computational thinking in the context of science learning is at the intersection of computer science, mathematics, and a specific science discipline. Computer science relates specifically to information sciences and associated technologies. Applied mathematics includes numerical models and statistics. Science disciplines involve knowledge and epistemic stances of a particular community of research and practice (e.g., biology, physics, or chemistry). When scientists use computational thinking, they apply computation, including information technologies, programed steps, and algorithms to conduct observations, collect data, generate lines of evidence, and construct valid explanations about phenomena.

Table 1 Computational science components and associated domains and student skills

	Information science	Mathematics	Scientific discipline
Fields and examples	Computer science, data structures, and information technology	Statistics, logarithms, graphs, and variables	Biology*, chemistry, physics, marine science, medicine, and physiology
Student learned skills	Data structures, refinement, appropriate programs to use, computer languages, computer theory, and development of data	Relationship of input and output, applied mathematics, appropriate representations, and modeling	Content from discipline, questions posed from discipline, applied theory from discipline, and appropriate selection of context from discipline

Note: Biology is our specific discipline of interest

In terms of a science learning progression, computational thinking would involve a students' understanding and application of input, integration, output, and feedback based on sets of questions and problems within specific scientific domains. Applied mathematical practices, such as algorithms and statistics, would also be crucial to developing and understanding concepts within domains at appropriate ability levels. Therefore, computational thinking would involve the knowledge and application of appropriate tools that merge the mathematical aspects and computational tools to ultimately display, revise, and become immersed within (scientific) content-specific models. Merging these educational aspects for the purpose of gaining knowledge is the type of higher level learning that is expected of professionals within the field of biology; we propose this is also beneficial for biology students. For example, under the context of input, scientists often think about how and where they might obtain their data (e.g., from a database, simulations, or the environment), and how they might format and store these data. Under the context of integration, scientists reason about variable relations and the appropriate tools for modeling these relations (e.g., interface-friendly tools or hard coding). Knowledge of and classroom participation in these types of NOS processes may potential facilitate students' biological understanding (NGSS Lead States 2013).

Student understanding of the most efficient practices and specific programs associated with their scientific practices is a key component in learning through computation. A student may model a scientific phenomenon using these aspects of computation; however, these aspects are distinct from modeling. Computation allows students' models to be dynamic, changing, and emergent. Computation may be in the form of a model, but not all computations are forms of models.

3.3 NGSS Computational Thinking: Practice 5

Because the NGSS has incorporated computational thinking into its core practices, Weintrop et al. (2016) provided four categories including data practices, modeling and simulation practices, computational problem solving practices, and system thinking practices as essential components of computational integration for science curriculums. This type of thinking should be woven throughout mathematics and science practices due to increased demand of technology education and the interdisciplinary benefits that may be associated with domain emphasis (Wing 2006). Thus far, there are no examples of disciplinary content paralleled with computational components as a progression with increasing complexity.

The NGSS states that computation is a fundamental tool for science, engineering, and mathematics learning in order to represent physical variables, recognize, express, and apply their relationships across levels. According to the reasoning laid out in the NGSS framework (National Research Council 2012), computational thinking in primary school may be rudimentary but would still build on students' prior experience and allow students to understand that mathematics may be used to describe the natural world. At this level, students are counting and using patterns, while also designing simple graphs and alternate solutions to a problem. At the elementary level, students are using computation and mathematics to analyze data and compare alternative design solutions. Students are also expected to organize data sets to understand patterns and relationships. Computation may be used to describe, measure, estimate, or graph scientific questions or problems while incorporating simple algorithms.

At the middle school level, students move to larger sets of data and use this data to support explanations and arguments. Students should be using digital tools to identify trends and support scientific conclusions. Students may be able to create algorithms to solve or design solutions. At the high school level, students should be revamping various types of functions and computational tools for statistical analysis to model data. Simple simulations are created and used based on mathematical models or assumptions. Students should be revising or designing models that represent scientific processes or systems. Students should be comparing their models to the real world and what is known about their phenomenon (NGSS Lead States 2013). The NGSS expects a gradual and achievable sequence of objectives (as listed above) through a series of events as students navigate through mathematical and computational thinking; however, the description of application as provided by the NGSS is vague.

Basic algorithms and equations are more often introduced in physics and chemistry courses when compared to biology (Gobert and Buckley 2000). Activities that occur in these physical science classrooms may present mathematical practices and concepts plainly to students, lending to more straight forward infusion of computational thinking in these areas. Foundational topics of biology are rarely presented to students within integrated mathematical contexts. Rather, biology is more often presented as a narrative via a series of worded phrases, rules, and/or conditions that students must memorize (Gross 2004). For example, in physics, teachers often use a mathematical equation to represent Newton's Second Law of Motion. Further, in many science classrooms, regardless of the domain, teachers may integrate computational thinking through the context of interface-friendly tasks and games. These friendly and gamified tasks remove the need for coding, which may limit the degree and depth of students' computational thinking (Grover 2011). Although there may advantages to these, such friendly interfaces and games are not widely used by biologists when conducting scientific investigations. Our proposed learning progression accommodates simpler interfaces, but also provides a richer context of and increasing complexity for computational thinking (Guo et al. 2016).

3.4 Computational Thinking to Support Biology Evolution Learning

There is an important relationship between scientific knowledge, the learning of science, and pedagogy (Driver et al. 1994). Adapting these NOS practices to classrooms provide benefits of authenticity for science learning, and this includes techniques, attitudes, and social interactions. Vygotsky's Social Constructivist theory (1962) states that knowledge is co-constructed and it is imperative that individuals learn from each other in a way that increased social engagement (Sinatra et al. 2015). Although technology is important for classrooms and integration has been difficult to incorporate in practice, it allows students to become active learners (while using skills that are interactive contemporary and relevant; Edelson 1998).

Some complex systems are integrated into the life science curriculum such as evolution, equilibrium, and homeostasis, but the overall theme that conjoins these ideas has not been developed. There are methods (e.g., agent-based modeling, information flows, system environment interaction, developmental trajectories, self-organization) which allow for both domain-based qualitative reasoning and quantitative modeling. The gaps in the curriculum—which may be due to inherent lack of computational implementation—do not allow students to cognitively bridge between separate curricular elements and prevents a conceptual framework of coherence. Complex system perspectives that may be explored through computational modeling and network analysis provide new methods and insights for learning science research. This research may potentially extend theory in the learning sciences through computational modeling of systems for learners and educators. Computation has the ability to enhance learning science research involving micro- and macro-levels of cognitive learning (Jacobson and Wilensky 2006).

The study of complex systems in association with computational technologies allow researchers to study aspects of the world in which structure and order may coexist at many different scales of time, space, and organization. These ideas are not limited to the sciences and are being integrated into other professions, such as engineering, medicine, finance, and law. Such widespread applicability further supports that integration of computation thinking into domain learning. For example, from the biological perspective, interdependence and coevolution with emergent patterns formed by self-organization are fundamental concepts to biology that are not frequently practiced by students (Jacobson and Wilensky 2006).

One of the best ways to represent the concepts, intricacies, and complexities of biological evolution is through computation due to its robustness and disguised simplicity (Toffoli 2004). In short, computational processes have the ability to stand unaffected although input variables may change. This implies that the computational output may be very different from the input although the code is relatively straight forward for users (which promotes various degrees of complexity for users [students]). For example, a simple population growth rate formula consisting of a few variables may be used to re-inform itself through proper implementation of coding and biological principles to develop a vast display simulating multiple generations of associated evolutionary changes (there are various avenues this can be implemented using computation). Another examples are the changes within representative strings of letters, including A's, T's, G's, and C's, corresponding to the nucleotide sequences (adenine, thymine, guanine, and cytosine) found within the genome of an organism (which may be generated by users [i.e., students] or found in a database).

These sequences (from the previous examples) may become computationally modified through a very simple "random" function via simulated generations to display a quantitative analysis of natural genetic mutations found in nature. The mathematics behind the random function is hidden, as it takes a value (x) as defined by the user (i.e., students) and multiplies this x value by "rand(x)" to provide a totally random output value that differs for each iteration. In turn, these sequences may further be translated into the respective proteins and/or altered and potentially displayed as defective proteins based on the random mutations that would occur in nature. Students could use simple compare functions to read through A, T, G, and C nucleotide sequences of various organisms to develop a broad, yet accurate evolutionary relationships in the form of phylogenetic trees. Educators could use scaffolding and appropriate differentiation with the assistance of our learning progression to determine appropriate classroom practices based on the biological examples above. For example, if students initially lack coding skills, they can perform the activity using excel or google sheets (simple) versus software such as the R stats package (a free coding program, which naturally lends toward more complexity).

Modeling is an important aspect of computation and integrating this practice has been very difficult in classrooms (Lehrer and Schauble 2006). Students can start practicing computational thinking in the years prior to kindergarten because computational modeling involves a complex form of epistemology that must be developed early on in student careers (in order for them to properly inquire about natural systems later on in their education). Models that work best for students are the type in which they can see the direct relationship between the natural world (e.g., as in modeling a compost pile), and students tend to find these types most relatable and easiest to understand (Krajcik 2012). Therefore, students can use computation to supplement these types of models. Professional scientists use models and computational thinking at all levels when solving problems, so when an idea translates to the classroom, it should result in a variety of representations and models that can be used to build layers of description, and potentially display different aspects of represented phenomena (Toffoli 2004). Computational representations at different levels should relate to each other in ways that students can understand so their ideas circulate and interlock, resulting in a system that enhances student understanding of the natural world (Lehrer and Schauble 2006).

Students at all levels have difficulty working at the intersection of mathematics, computer science, and a scientific discipline; these aspects are what prevent students from performing genuine research in the ways scientists do. When life science students conduct genuine research (usually through working on a portion of a mentor's larger research project as a graduate student), it is usually the first time they are faced with the task of developing their own questions while concurrently developing mechanisms using computation (e.g., coupling input with statistics; Ryder et al. 1999). However, inquiry-based research projects and other activities that develop the foundation for students to start developing these basic scientific skills can be executed at the middle and high school levels (NGSS Lead States 2013). Learning and doing cannot really be separated when talking about scientific practices. This implies that students should learn to do something, while using appropriate knowledge and tools to meet learning objectives (Bybee 2011).

In a study using NetLogo, Wilensky and Reisman (2006) found that students were able to make sense of problems and develop explanatory models on their own through computational processes. During programming, students tended to go back and forth between new hypotheses and researching existing solutions. For example, Talia (pseudonym)—a student who participated in the study—frequently used a computational model is to simulate predator and prey relatonships across generations. Talia developed the idea that many model components were random, which is very difficult for students to understand on their own without using a computational tool. As her models did not match what would occur in nature, she continued to change her parameters until the output matched what would make sense to her working knowledge. As she set out to correct her model, she underwent the process of debugging and sought out literature that was associated with her phenomenon.

Paul (pseudonym), another student in Wilensky and Reisman's (2006) NetLogo study, used the program to study the pattern of flashes displayed by a specific species of firefly. In nature, there are rules and patterns that alter the flashing patterns of these insects. Paul made initial assumptions, such as the fact that all fireflies follow the same rules or that the fireflies' mechanism becomes synchronized upon coming into contact with other fireflies. As his research continued, he developed a decision-making process and unique tools that merged identifying relevant content knowledge with computational thinking (via embodiment). Paul actually developed new knowledge from his model that contributed directly to the field, and this information would have been nearly impossible to discover through field work alone. In many cases (such as Talia or Paul's), students start to think differently about biological rules and phenomena when using computational thinking. Their interaction with the NOS through the use of this program supported biological evolution learning through computation.

3.5 The Relationship between Biological Evolution, Levels of Organization and Computation

As students work with computational programs, they may learn to set up problems and relate their working knowledge with tools that they have available. They may better lay out their problem at hand and intuitively select programs that may maximize input or have the right capacity for a working statistical analysis. Students may select which programs work well together to select the most appropriate display output for the intended audience (Chandrasekharan and Nersessian 2015). The only way students may understand the best practices is by working with these specific programs, and eventually learning these new tools becomes easier. Students may start to think about content knowledge and data as sharing a relationship that is more intimate and interrelated since output may be looped back and used as new input within their systems (Jacobson and Wilensky 2006). Students may develop examples at all biological levels from computational scratch which would also enforce or challenge their working content knowledge in biology. The various levels of biological organization, relevant examples, and computational aspects are displayed in Table 2.

Evolutionary thinking is required as early as in the middle school in order for students to properly model these biological systems later on in their educational careers. Lehrer and Schauble (2006) claim that building blocks of evolution understanding include variability, change, and ecosystems. Variability distinguishes between directed and random variation, and students must understand the relationship between the change in organisms, populations, and ecosystems and discover ways of describing interactions between organisms. Understanding involves coordinating change at all levels of biological organization and relating this to how it works in ecosystems. Lehrer and Schauble claim that modeling and thinking required for modeling allow students to reason more deeply about evolutionary questions. Ideally by the time students reach high school, they have worked through the process required for complex evolutionary thinking through the process of various activities (such as modeling) as displayed in our proposed learning progression.

Forrest claimed (1990) that emergent computation requires the explicit and the emergent levels of a system to be developed and displayed through computation and information that is not present at lower levels will exist at the level of collective activities. There must be a collection of agents and interactions between agents, and these agents must follow instructions which have directions at the macroscopic level, parallelling most biological systems. If the phenomenon being observed is also computation itself, there may be feedback between the levels. This idea is stressed in computational learning and as a principle in biology, especially in terms of evolution. Three themes of this phenomena are self organization, collective phenomena, and cooperative behavior (Forrest 1990). The sum of the parts within a system is greater than individual parts, a common theme in biology as well as computation.

Biological functions are a result of mechanisms that occur at various scales and biological levels of organization (Campbell et al. 2000) as seen in Table 2. Modeling and simulations are computational tools necessary for describing, predicting, and understanding these mechanisms in a quantitative and integrative way. Dada and Mendes claimed (2011) that understanding biological functions should be aided by computational thinking and encompass and span various spatial and temporal scales. Understanding systems in biology is not possible without using computational technology and looking at various levels of biology. These scales range from molecules, genes, and proteins through to cells, tissues, organs, organisms, and the interactions with other organisms and the environment (Guo et al. 2016). Time scales may range from microsecond to hundreds of thousands of years. In order to understand behavior of a system, it requires various interactions that occur on these diverse scales. Exploration of these items occur now at the educational and professional level, but the need is growing exponentially in the field of biology especially for research. Researchers may explore a top down or bottom up approach and approaches may be discrete or continuous. There are general

Table 2 Levels of biological organization and applicable computational representations

Level of biological organization	Broad examples	Applicable examples
Biosphere	Earth	-Ecosystem interactions -Ecosystem modeling -Global nutrient and energy flow
Ecosystem	Estuary, open ocean, deciduous forest, and tundra	-Environmental influence on biota at various levels -Ecosystem modeling (nutrients & energy)
Community	Grassland community, coral community, pond community, and gastrointestinal microbiota	-Community structure and changes -Interspecies interactions: i.e., competition -Communal influence on organisms
Population	Honeybee population, osprey population, and seagrass population	-Variation -Interspecies interactions: i.e., competition -Growth, decline, and carrying capacity -Genetic (allele) changes within populations
Organism	Pitch pine tree, horseshoe crab, blue jay, and diatom	-Competition -Reproduction -Energy conservation due to processes associated with homeostasis
Tissues and organs	Muscle tissue, heart, and osteons (bone tissue)	-Similarities and differences within and among organisms hold evolutionary explanations (vestigial structures, homologous structures) -Exploring explanations for variations within populations and between species based on environment
Organelles and cells	Nucleus, mitochondria, chloroplast, flagella, neurons, and osteocytes	-Cell interactions within and between organisms operating under evolutionary principles -Organelle structure operating under energy conservation principles -Cell Structure operates under energy conservation -Cellular evolutionary differences and similarities between organisms
Atoms, Molecules and macromolecules	Na+, Cl+, H_2O, CO_2, protein, lipids, DNA, RNA, and carbohydrates (sugars)	-Diffusion, transport, and molecule interaction working under energy conservation principles and availability within environment. -Similarities and differences between organisms -Production, inhibition, blockage, and development of systems using specific compounds based on environmental pressures -Variation at this level (gene or protein) -Mutations occur at this level

Note: Examples of levels of biological organization and evolutionary (including: unity and diversity) principles associated that can be explored computationally in learning progression

computer languages and platforms that researchers may use, or explicit programs tailored for specific content-based computational tasks. Regardless of the approach, in order to explore systems within biology, all levels may be explored and integrated through computation (Dada et al. 2011).

Of the few documented interventions that combine evolution learning with computational thinking, there are none that span between the organizational levels of biology. Guo et al. (2016) conducted a study involving high school students in which agent based software was used to simulate frog population changes over time. Although the students did make knowledge gains in evolution, additional instruction and computational tools that would allow students to make better connections between micro- and macro-levels of biology were required (Guo et al. 2016). Students have trouble understanding the connections between these levels; the process of evolution, which operates at and between these scales, presents a promising opportunity for student exploration of these level connections. At any given time, topics in biology may be taught at a single level (molecular, cellular, anatomic, organismic, or ecological level) and ideas often become isolated from one another.

Making various biological level connections and connections between the micro- and macro-levels contribute significantly to biological evolution learning. This may be because in biological systems, the explanations of mechanisms of phenomena (such as biological evolution) apparent at one scale often lie at a different scale (Parker et al. 2012). Students' difficulties in making micro- to macro-scale connections is sometimes referred to as "slippage between levels" (or disconnects between levels) and is also associated with fragmented and compartmentalized knowledge (Brown and Schwartz 2009). This problem has not received much attention in the literature on evolution education (Jördens et al. 2016). Fluidity between these levels allows students to reason across them and contribute to biological literacy (Brown and Schwartz 2009). There are five chief strategies that encourage thinking across levels in biology (Parker et al. 2012), including (a) distinguishing different levels of organization, (b) interrelating concepts at the same level of organization (horizontal coherence), (c) interrelating concepts at different levels of organization (vertical coherence), (d) thinking back and forth between levels (yo-yo thinking), and (e) meta-reflecting about the question of which levels have been transcended (Jördens et al. 2016). Connecting macro- and micro-level connections may be important; however, there is a gap in research explicitly identifying multiple levels within the micro- and macro-level ranges and considering biological connections in the ways we have defined them (Jördens et al. 2016). And, to our knowledge, none have used computational thinking to facilitate making these connections.

Cross-level understandings may be explored through embodied experiences while using computational thinking. For example, as students better understand behavior of molecules, it allows them to get a better grasp on cellular processes, or as students develop an understanding of agent/organism behaviors, it may allow them to better explain ecosystem processes. These types of relationships would not only emphasize each topic, but should allow students to better understand biology as a whole (Jacobson and Wilensky 2006). It is of benefit for all students to have the ability to think like scientists and use tools that promote scientific inquiry (Wilensky and Reisman 2006). Even more so, computational thinking is not limited to science; computational skills and specific software programs are increasingly used by professionals across disciplines and within associated careers.

3.6 Computational Thinking Becomes a Working Tool for Students

Computational thinking and learning is encouraged for student access in STEM careers because it becomes a working tool in problem solving that may also be applied in other domains. Computational experiences are becoming increasingly available to students due to reduced prices of technology and due to easier access to data and improved methods of data streaming during experiments (Jacobson and Wilensky 2006). Realistic ways of doing science in the field involve both direct observations and computational modeling practices. It is becoming increasingly difficult for students to distinguish models and simulations from observations. Simulation translates everything including algorithms into digital information and uses computation to construct any object, even if it is an abstraction (Lenhard 2010). This is due to the fact that many models have controlled sensors or external devices that stream and interpret live data. Practicing scientists have opportunities to use these technologies, and there is a need to understand how students and teachers may best learn to use them in practice.

These complex system informatics and representation tools allow student development in thinking about, interpreting, and representing data through relevant complex systems, concepts, and principles, such as those associated with evolution. Designing learning environments and selecting appropriate tools that make organizing conceptual framework explicit to students is also important for instructors (Jacobson and Wilensky 2006). In short, students should be able to distinguish between data, and merge ideas and tools associated with computation, to concurrently allow them to thrive within content areas (Kong and Abelson 2019).

This emphasis of student awareness with respect to computation and abstraction is a specific example and extension of Baudrillard's (1994) book that described simulacra and simulation, as well as the relations between reality, symbols, and society. This book was one of the inspirations for the famous science fiction movie associated with computer science and programming, *The Matrix* (Wachowski et al. 1999), where many of the characters unknowingly start their lives immersed in a computational abstraction of reality (Tamatea 2019). Baudrillard's (1994) concern was that individuals within a society would not be able to understand or know the difference between truth, reality, and simulation; in other words, the abstraction would be indistinguishable from reality. Copies of copies of copies of original information would be the entity that reaches individuals and that the hyper-real rather than the real would inform discussions. For example, an individual may have never been to France; however, they know it exists due to news stories, pictures, maps, and personal accounts. What they know of the country is not based on personal observations, but is informed solely by societal constructs and abstractions. This idea may be similar to the "realness" of a scientific concept, such as the idea of a cell and how students may or may not understand and explore it. One of the criticisms of *The Matrix* film is that the select few main characters know when they are in the simulated world and when they are in the real world. The characters in the film use programming as a tool to better navigate and understand the simulation, and this idea is synonymous with technology infiltrating the sciences and science learning. Baudrillard (1994) states that abstraction will eventually be the only means through which we access the real because we to prefer the map of France (which is an abstraction) over the territory (which is real; Tamatea 2019). Learning coding seems to be one effective way in which we can engage with abstractions.

Students should be able to make distinctions between what they experience and what is constructed in science and simulations; without building these tools (i.e., through computational thinking), students may have difficulty distinguishing what is scientifically valid. Computational thinking practices would allow students to get a better understanding of the development of scientific simulations and representations (even though they might not fully understand them). Pioneer science, hypothetical experiments, in situ experiments, and computational models should be distinguishable for students. In today's world, understanding information science and respective abstractions as it relates to science such as biology is extremely important because of the radical changes information science has already made (and will likely continue to make) in our society (Tamatea 2019).

Children are exposed to digital technologies at an early age, but children and adults alike often do not understand how these technologies work computationally. For example, most students know how to perform a Google search, but may not understand the computational mechanism of the search (Alexander 2018). These searches involve selection and ranking of both valid and invalid information found on the internet, and without some knowledge of these algorithmic processes, some students may not successfully distinguish between credible information sources and plausible explanations (Sinatra and Lombardi 2020). In classroom biology learning, integrating computational thinking, and evolution—via our proposed learning progressions—may address, in some small way, the potential threat of disinformation through increased scientific literacy.

4 A Learning Progression for Computational Thinking in Evolution

Learning progressions (LPs) vary across the educational literature; however, Berland and McNeill (2010) describes three explicit components of LPs. The initial component (1) is described as a developmental progression for how understanding develops, the second component (2) is described as a progression in increasing levels of complexity of the disciplinary knowledge and practices, and the third component (3) is described as using pathways to support student learning. Analysis of disciplinary knowledge is essential for identifying the big ideas in science such as biological evolution, which is the primary focus of learning progressions. Our specific LP, called learning biological evolution through computational thinking (LBECT), is modified from Berland and McNeill (2010) framework, which was grounded in both science studies of disciplinary practice and research on student learning. LBECT-LP focuses on merging dominant ideas in biological evolution and computational thinking practices. Our goal is to assist practitioners in increasing computational thinking in classrooms; therefore, the LBECT-LP is flexible. Educators must consider where they themselves, their students, and their districts are realistically situated in terms of practical application when considering how to incorporate the LBECT-LP.

5 Instructional Context, Computational Product, and Computational Process

As shown in Table 3, the LBECT-LP has three dimensions: (a) instructional context (i.e., educator role), (b) computational process (i.e., student activities), and (c) computational product (i.e., student artifacts). These three dimensions can be either simple, developing, or complex depending upon the level and progression of the student. Each of the three components of the learning progression may be evaluated per student as they show evidence from the simple, developing, or complex categories. The LBECT-LP includes various components, including student engagement by considering multiple student perspectives, constructive group interaction, and complex thought processes to integrate multiple pieces of information (i.e., computational process). These components continually align with learning biological evolution through computation across grade levels. Students have some idea of what biological evolution is; however, their elementary viewpoints or misconceptions may contribute to their multiple perspectives. Viewing evolution at various time scales and levels of organization provides various alternate perspectives. Computation activities may also be highly interactive and involve multiple smaller pieces of evidence to construct the larger idea if properly scaffolded (i.e., instructional context).

Table 3 specifically represents these components through students' classroom behaviors (i.e., computational process) that promote learning about biological evolution. For example, the data used in the computational process in the LBECT-LP may be provided by the teacher or developed by the student through the computational process, depending on the level of the student (i.e., instructional context). Students may produce various complex outputs, which they then will have to reasonably evaluate. Students will learn to make arguments that support their claims through the computational process. These claims will involve decision making regarding the data, computational reasoning, and the principles of evolution as their understanding improves within their domain area. Instructors can develop differentiated activities, rubrics, and assignments based on these three dimensions (i.e., instructional context). The NGSS standards can be appropriately assigned by the teacher depending on grade level using our learning progression.

As students progress from simple toward complex, questions initially provided through instruction will be closely defined with limited sets of answers and data sets will be provided (Table 3). Eventually, little scaffolding is required and students must determine which data are appropriate and how to select their data using computational reasoning. The simple computational product is not necessarily represented efficiently or appropriately, and students have difficulty describing their computational process. As student abilities increase in complexity, their representations take various forms and become appropriate for the design at hand. As students move toward a complex computational process, they are able to describe and justify input, manipulate proper modification, or synthesize their own data. Students eventually develop and justify their mathematical representations and integrations. At the highest levels of complexity students may develop new computational tools (i.e., computational products), representations, or challenge domain-specific content and ideas.

Input, integration, output, and feedback are integral components of computation as displayed in Table 4. As students develop their computational process and product through instruction (Table 3), they will need to think more specifically about their computation. As students integrate content knowledge into their computational processes and products, they will not only be developing their content knowledge and computational skills, but they will also be improving their communication skills (regarding computation).

Knowledge and communication skills will be developed through students experiencing this computational process and developing computational products. Specific computational component (input, integration, output, and feedback) examples are directly related to explicit scientific and computational knowledge and communication skills as displayed by specific student roles in Table 4. For example, as students decide what information is important and relevant to their scientific discipline to solve problems related to the phenomena at hand, they are contributing to their working knowledge associated with the scientific discipline. As students decide what equations to use and how they are related, they are further building upon their scientific knowledge. As they decide which information is important to display as output, they are evaluating which components of their scientific system is relevant. As students decide how to reintegrate information to inform their system, they are further engaging in making associations between working scientific knowledge within their content area.

As students are contributing to their scientific knowledge through these aspects of computation, they are also developing their computational knowledge as displayed in Table 4. As students decide which scientific input to use, they are better understanding how to define variables, a crucial component in computation. While students write their programs in computer code using relevant scientific information, they learn additional logical

Table 3 Learning biological evolution through computational thinking: learning progression

Dimension of LP	Simple	→	Complex
Instructional context	-Question closely defined with limited set of answers -Data set is small -Detailed scaffolds -Data set limited to appropriate data	-Data set is large -Question has multiple potential answers	-Moderate scaffolds -Students define or develop data set -Data set includes appropriate and inappropriate data -No scaffolds
Computational product	-Idea is computationally represented, may be inappropriate -Not most appropriate design or output representation -Student unable to produce written component describing computational process	Idea is computationally represented appropriately -More than one design was considered for output representation -Student able to produce some written component describing computational process	Idea is computationally represented in various forms -Most appropriate design or computational representation understood -Student able to produce various types of open-ended written component describing computational process -Student spontaneously developing computational ideas as related to content area
Computational process	-Weak development of input and little use of data modification and data synthesis -Weak development of mathematical/statistical component -Student unable to make connection between domain idea and computational aspect -Model unable to re-inform itself -Student unable to produce written component describing computational process	-Development of input and proper use of data modification and adequate data synthesis -Development of mathematical/statistical component -Student able to make connection between domain idea and computational aspect -Model able to reform itself -Student able to produce written component describing computational process	-Student able to describe input and proper use of data modification and adequate data synthesis -Student uses combinations of mathematical/statistical components -Student develops new connections between domain idea and computational aspect -Model able to reform itself and other models and various output methods -Student able to develop new computational processes and ideas associated with domain

Table 4 Components of computational thinking and student role

Computational thinking component	Input	Integration	Output	Feedback
Scientific knowledge	Student decides the variables to include based on scientific knowledge and practices.	Student decides which aspects of working science knowledge are important and which applied science concepts are important.	Students decide if and or how output applies to their problem or phenomena.	Does this knowledge support or contradict working science knowledge?
		Student identifies the strengths and weaknesses of each program based on their scientific problem.	Students compare output to working knowledge and body of science knowledge.	Student decides how to re assess personal working knowledge.
				What other types of questions might it solve?
				Did I discover anything new that may contribute to science?
Computational knowledge	Students decide the "type" of variables and how they are represented.	Students select how aspects are represented or modified in the program?	Students validate and justify their computation.	What other modifications or variables may make it more accurate?
		Students decide which formulas will be used.	Students ensure robustness.	What modifications may make it more robust?
		Students decide which variables will interact	Students compare types of output.	Is the computational process efficient?
		Students decide which program they will use and develop working knowledge as to why.		
Communication	Students decide complexity and type of interface.	Is any pseudocode required?	Students decide the most appropriate format for output (graph, visual, numbers, sound, website, package, file etc.) based on audience.	How can this tool be used and applied to other disciplines?
	Will it be for working problem solving only or communicated?	Who needs to read or use the program?		How may I modify to make it easier for others to use?
				How can I make this a standalone package so that others can use it to solve the same types of problems?

Note: Student role may be in the form of question, action, or artifact. Feedback is for students to decide on how to revamp model or how output might affect input computational skills such as loops and if-then statements. When students display their output, they are learning which tools are available to visualize various representations and decide which is most relevant, another essential component of computation and communication. As students re-inform their models, they are practicing revision, while ensuring their models are valid and robust, additional critical components of computation.

Various aspects of communication are a critical component of the learning process, but also a component of the computational and scientific learning aspect as displayed in Table 4. As students input their information and integrate their code, pseudocode may be used. Pseudocode is a text written into computer code that is not processed by the computer and only used as notes or a means to communicate with others reading their code. As students decide which information to display in their pseudocode, it dictates how many others may use their written script. When students decide how their code and packages run, it dictates how user friendly it is for others and essentially who can viably use their work.

As students progress from simple toward complex in the learning progression, they will develop advancing understanding of scientific knowledge and computational knowledge. Students will start with simple scientific information as well as simple types of computation. Instructors can scaffold this progression for students (instructional context). For example, they may provide an interface heavy program for students exhibiting simple computation. The most developed types of scientific understanding occur as students start to make connections within and between their scientific domains, as well as when students design and conduct their own research through asking their own questions in the context of gaps in scientific knowledge. The most advanced types of computational knowledge include efficient means of writing raw computer code and developing packages and software in various computer languages. Although these are the most complex aspects of the learning progression, simple computation may involve programs with user friendly interfaces and are required for initial incorporation of the LP (e.g., MS excel, scratch or app inventor).

6 Construct Maps: Biological Unity and Biological Diversity

To break down the LBECT-LP further, we developed four construct maps representing student learning of biological evolution through computational thinking. Shown in Figs. 1, 2, 3, and 4, these maps represent two fundamentally important ideas for learning about biological evolution: unity (Figs. 1 and 2) and diversity (Figs. 3 and 4). We developed these construct maps by adapting Plummer and Krajcik (2010) LP for lunar motion and spatial reasoning. These maps reveal a structured process toward a more complete student understanding of evolution. Based on Plummer (2012), the LP shown in each map is dependent upon the instructional design that supports student progress toward understanding. The LP presents cognitive challenges as presented by the lower levels of understanding evolution as students' progress to deeper levels of understanding (Plummer and Krajcik 2010). This idea of progression is accompanied and supported by computational thinking and integrates embodiment, which can further enhance learning.

Each construct map has the scientific explanation for separate biological phenomenon as the top anchor. These construct maps can be staked or aligned to create a full learning progression toward a core idea and biological evolution. For each of the two topics (unity and diversity), there are three integrated elements of the LP: the model of cognition, instructional design, and assessment (Plummer and Krajcik 2010). As students gain scientific

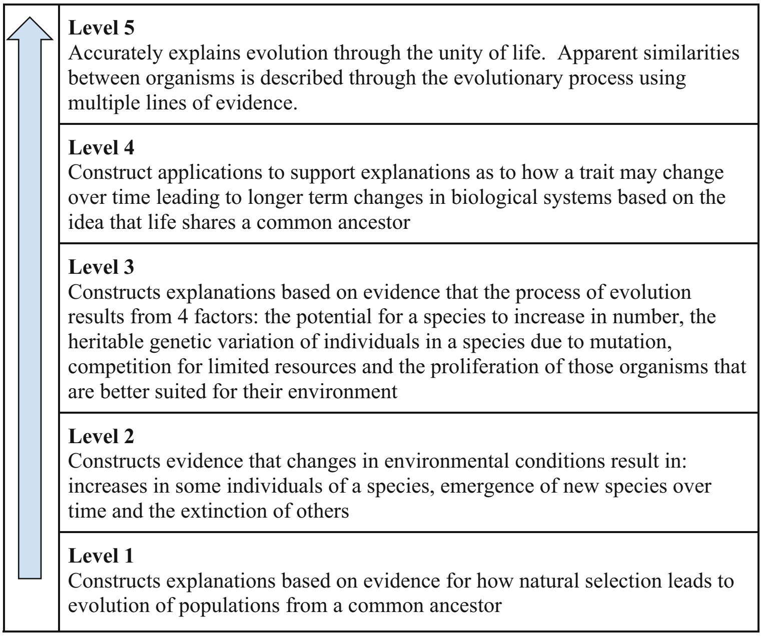

Fig. 1 Student understanding of evolution via unity at increasing complexity

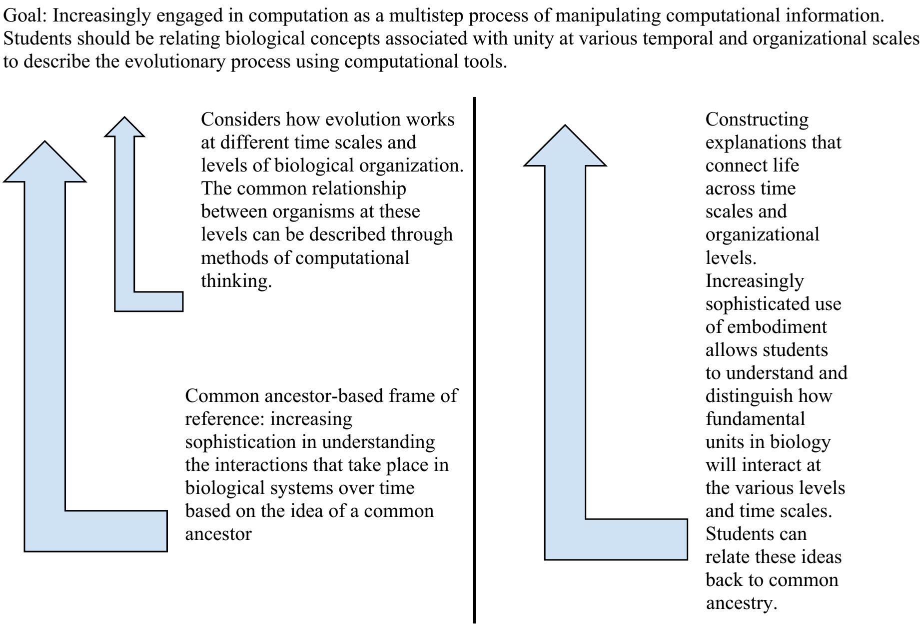

Fig. 2 Computational thinking associated with unity progress. Students should be increasingly engaged in computation as a multi-step process of manipulating evolutionary information and embodiment. Students will be exploring and producing evidence for the unity of life as explained by the evolutionary process through computation

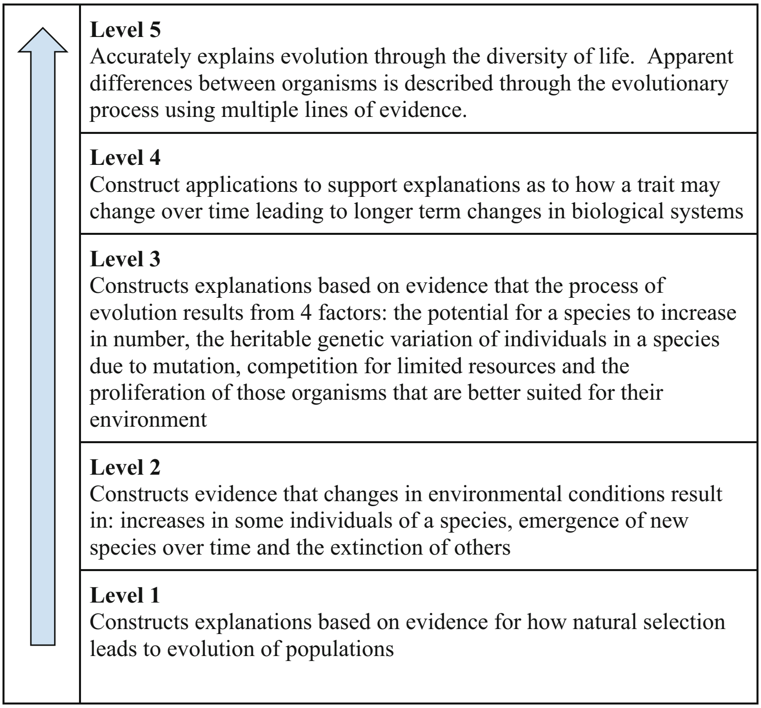

Fig. 3 Student understanding of evolution via diversity at increasing complexity

knowledge through learning the NOS and applying computational thinking, emerging ideas, capacities for computation, alternative representations, and causal reasoning. Students should be engaged in higher ways of knowing, learning, and thinking about ideas, evidence, and claims associated with evolution. As the LPs start earlier in a student's education, the communication, thought processes, and computational abilities will become more advanced in association with biology (Duschl et al. 2011).

Conceptual change about evolution would be facilitated by computational reasoning because computational thinking will increase students' cognitive engagement (Dole and Sinatra 1998). Of the various factors that allow conceptual change associated with evolution, embodiment, is key. Embodiment is achieved through the necessary components of computational thinking as students put themselves within computational models to think how the model agents and components of the model would interact (Wilensky and Reisman 2006). For example, in the Dole and Sinatra (1998) conceptual change framework, embodiment could promote a higher degree of cognitive engagement through greater elaborative connections and more metacognitive reflection. Computational thinking and its facilitated embodiment relates to the importance of how evolution spans across temporal and organizational levels of biology, while it also parallels emergent computation through learning.

Pathways to really understand evolution become evident through the learning progression that includes instructional context, computational product, and computational process, as paralleled by appropriate construct maps displaying the process of unity and diversity of life. Each construct has different component of evolution at top of anchor can be stacked or aligned to focus on that single idea (Wilson 2009). The bottom anchor is the evidence and learning progress of students.

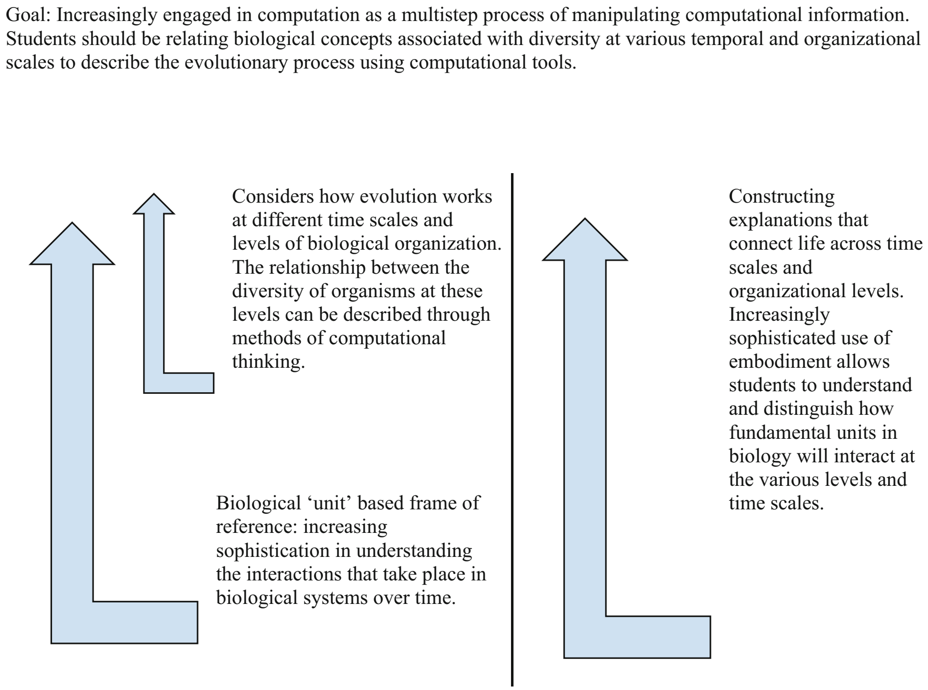

Fig. 4 Computational thinking associated with diversity progress. Students should be increasingly engaged in computation as a multi-step process of manipulating evolutionary information. Students will be exploring and producing evidence for the diversity of life as explained by the evolutionary process through computation

Two major ideas unity and diversity are displayed as construct maps in Figs. 1, 2, 3, and 4. The understanding of evolution through unity and diversity each represent a different map and each has levels 1–5 which correlate with NGSS standards. The corresponding table displays how computation will be used to support the learning progression of evolution. The major difference between unity and diversity learning would be that unity focuses on common ancestry and uses this as a perspective to explore evidence and generate computational processes; whereas, the diversity component focuses on the diversifying of organisms through more of a phylogenetic approach or looking at things from a larger environmental aspect. Aspects of both unity and diversity can view all levels of organization and a variety of time scales; however, unity would naturally align with smaller levels and diversity at larger levels. This can and should be done computationally for both unity and diversity, using computation as a bridge between physical and temporal scales.

6.1 Unity

As students understand evolution through unity, they initially construct explanations based on the evidence that there is a common ancestor and natural selection drives evolution. Eventually, students construct evidence that changes in environmental conditions allow for the increase in some individuals of a species while a decrease might occur in others. Students will then be able to construct explanations based on evidence that evolution results from four factors: the potential for species to increase in number, heritable genetic information is required, variation is random, competition is required, and those better suited to their environment have the increased ability to reproduce. Eventually, students may develop applications to support explanations as to how a trait may change over time leading to larger changes in biological systems, may describe the unity of life through similarities between organisms, and use multiple lines of evidence in their explanations.

As students initially do this, a common ancestor-based frame of reference may be. As increasing sophistication is achieved, students may understand the biological-level interactions that take place based on the idea of a common ancestor. For example, at a greater level of sophistication, students may be able to use tools such as the National Center for Biotechnology Information (https://www.ncbi.nlm.nih.gov/) and explore evolutionary relationships based on molecular evidence such as DNA, RNA, and protein sequencing. Students may construct explanations that connect life across time scales and organizational levels with an increasingly sophisticated use of embodiment that allow them to understand and distinguish how fundamental units in biology interact at various organizational levels and time scales. Eventually, students may consider how evolution works at these different scales, and how these levels may be described and explored through methods of computational thinking.

6.2 Diversity

As students learn evolution through diversity, they may construct explanations based on evidence for how natural selection leads to evolution of populations. They may construct evidence that changes in environmental conditions, resulting in the increase of some individuals, and evidence on how new species emerge or go extinct over time. For example, students may explore multiple generation population changes through modification of micro-scale entities (alleles) with the assistance of computation. They may also construct explanations based on evidence from the four factors (listed above), and construct explanations to support how a trait may change over time leading to long-term changes in biological systems and how apparent differences between organisms may be explained through multiple lines of evidence.

Students may use computational thinking to reason about a biological unit-based frame of reference. With increased understanding of the interactions between these units, students may also develop increased understanding of biological systems over time. Students may use embodiment to understand and distinguish how these fundamental units of biology will interact at various levels and time scales. These ideas will be achieved through methods of computational thinking.

7 Computational Lesson Examples

The Advanced Placement (AP) Biology Curriculum as developed by College Board stresses aspects of computation and modeling reflected by the NGSS. The College Board is the nonprofit organization that connects high school students toward college success and is responsible for the development of all AP courses, as well as other tasks such as administration of the SAT (College Board 2012). The AP biology curriculum also promotes the importance of understanding evolution, reflecting evolutionary evidence, and relevance within every section. AP biology courses are thought to be the most complex life science course offered at the high school level and should prepare students for the scientific thinking required of college students. Although the AP curriculum stresses that evolution is important in biology learning and emphasizes computation to do so, it does not identify how and to what degree the activities specifically contribute to learning of biological evolution and biology as a whole (Román-González et al. 2017).

As outlined by the College Board, there are two labs in the AP biology curriculum that require and emphasize computational thinking in order to display and work through evolutionary content (College Board 2012). When teaching these labs using computational thinking, instructors would need to structure the lessons with proper scaffolding, providing the appropriate amount of materials to facilitate the lesson as appropriate for students (simple, developing, or complex based on the LBECT-LP), and emphasize the input, integration, output, and feedback mechanisms for students coupled with appropriate computational activities. In one lab example, a phylogenetic tree activity is performed. NCBI, a free website (https://www.ncbi.nlm.nih.gov/), is used in this lab, and has nucleotide (DNA and RNA) and amino acid sequences for specific proteins and DNA regions from a variety of organisms as discovered by working scientists. Because DNA and RNA sequences encode for specific proteins, both the nucleotide and protein sequences may be used for inquiry. Students can search for protein or DNA sequences for specific known proteins from different organisms. The website has the ability to match similarities in the sequences and develops phylogenetic trees based on the similarities between the sequences. Students start to understand the relationship between the molecular scale sequences and how and why phylogenetic trees are developed based on evolutionary relationships. Students have the ability to sequence the data in a primitive way (interface friendly tasks) or using tools that are very robust (raw coding) depending on the level of student. Students work through exporting information; they frequently inquire about different levels of organization and the relationship to evolution while developing ways to navigate this information using computational tools. Students may develop the ability to see how and why this type of sequencing is most accurate and how it relates to anatomy of organisms and or other types evolutionary evidence such as embryology or the fossil record.

The Hardy-Weinberg (H-W) lab, another resource in the AP biology lab manual (College Board 2012), uses a simple equation to display changes in population over time through the display of allele frequencies in a spreadsheet. In studying this law of genetic equilibrium, students explore the relationship between evolution and the change in allele frequency by using a computational model to demonstrate what can happen over many generations. In this AP biology, lab students develop a specific trait that is represented by alleles: either homozygous dominant, heterozygous, or homozygous recessive. Each of the allele combinations or genes represents a different physical feature or phenotype that is hypothetical and described by students. This trait may be indicative of various hypothetical phenotypes such as fur color, ear shape, and cell structure. Students then develop a method of input, which must involve incorporation of the Hardy-Weinberg equation to ensure that their program feeds a proper output. Students color code their work so that they understand and properly display input, working code, and mathematical equations as well as an output. Their output varies based on randomness that is built into the model.

These numbers of alleles can be compared to populations in the nature, and it can be observed if species are undergoing evolution or not. In the case, species are undergoing evolution, and questions may be asked why (e.g., environmental changes, adaptations, and human impacts). Students work with the computer programs, such as common spreadsheet software, in order to develop a model which mimics two successive generations. Students could design their computational products so that there is a single input value that the rest of the model runs on. The integration would be the student's depiction of the Hardy-Weinberg equation, and the output would be the new allele frequencies and respective graphs for each generation. The generations output informs the next generation, and students should recognize this as the feedback.

Students identify how the Hardy-Weinberg equation relates to population changes. Students then are required to modify their population in that a certain “made up” or hypothetical phenotype (heterozygous for example) has an advantage or disadvantage. Students must then alter their equations so that a certain phenotype has a, for example, double advantage in survival or become less advantageous by 50%. Students must then describe how this affects their populations after a series of generations. Their output from one generation becomes the input for the next and so on, with graphical output displaying the allele frequency changes over time. For example, hypothetical homozygous recessive (rr) sea otter individuals have long hair, and because the hypothetical local environment experienced an average 5° temperature increase due to a new nuclear power plant, it might result in a 50% loss in the homozygous recessive otter individuals. Students would then be expected to model how that 50% loss affect future generations evolutionarily using the appropriate equation and modifications.

Students essentially develop methods to display a series of generations within a population. As students model their simulated populations based on code, modifications and proper display they eventually work to generate proper working explanations, which involve multiple levels of organization as well as a proper understanding of time to explain evolution. These explanations merge working biological principles with computational principles and the models may become more advanced as students develop their skills and content knowledge. Assessment would ideally include computational and biological components and display a level of simple, developing, and complex as defined by the instructor. Although these labs are already computational in nature, often instructors become frustrated as the instructions are vague, and resources are outdated. Issues in assessing and developing these types of activities are addressed in the next section.

8 Directions for Future Research

An overwhelming gap in research stretches across computational thinking, specifically for students enrolled in biology courses (Grover 2011). The student-guided construction of scientific models may lead to internal generation of knowledge and information that contradicts what students currently believe, a necessary component in conceptual change (Sinatra et al. 2008). Learning about specific controversial topics such as evolution may lead to conceptual change if common obstacles between conceptual change and evolution are overcome through computational thinking, contributing to student understanding of biology, and a more advanced understanding of the NOS. Further, facilitating computational learning probably requires appropriate scaffolding; however, the best practices for helping students to think computationally to foster conceptual change and deep learning are unknown.

Although there has been some model-based reasoning research conducted among science learners at higher educational levels, there is a gap in research about learning evolution through computation, with virtually no research at the K-12 levels (Jacobson and Wilensky 2006). There has been some research and investigation on student learning of evolution and biological systems through agent-based modeling; however, no studies have related computational thinking and reasoning to learning evolution across scales (Aho 2012), nor have they defined computational thinking in the ways that we have.

The proposed framework and learning progression allows for teachers to modify their current classroom activities in order to make their lessons and instructional units more computational. We provide this avenue by arranging and outlining ideas for instructional context (instruction), computational product (artifacts), and student roles (computational process) within the learning progression. We also provided two example lessons for a biology classroom that were modified to be more computational from the AP lab manual and included the respective corresponding aspects of the progression (e.g., simple vs. complex student roles). We recognize that this idea is novel; therefore, testing of the LBECT-LP (or at least aspects of it) by the research community is warranted. Such research would include tests of lessons, instructional units, and assessments based on the LBECT-LP to better understand impacts on classroom practice and effectiveness.

The learning outcomes, pedagogies, and tools for deeper conceptual understanding and knowledge transfer for topics associated with evolution are not well known, especially about how these topics relate to overall biology understanding (Jacobson and Wilensky 2006). There is a definite urgency to incorporate computational thinking practices into scientific curriculums according to the NGSS; however, the incorporation strategies are vague. Definitions for computational thinking as applied in classrooms are still unclear. The computational instructional practices, computational process, and computational products are not well defined for instructors. We have provided structure for educators through the LBECT-LP, while still maintaining flexibility so that teachers may modify their lessons as they see necessary.

It is unclear how to assess computation alone; therefore, assessment of other content knowledge such as biology while using computation as a learning tool may be better suited for students and teachers. The development of an assessment for computational skills alone or the assessment of computation alongside and integrated within content knowledge is necessary in order to monitor student progress and is essential for continued research. The type of information that this assessment would measure is still unknown (i.e., computational knowledge and or biological evolution knowledge). There are also no assessment tools that measure computational skills (in the way we have defined them) as applied across disciplines (Werner et al. 2012).

Content knowledge, such as biological evolution, may be assessed based on specific skills that are developed by computation. For example, computational thinking may aid in evolution learning by facilitating modeling biological process at and between levels of organization. There are biological evolution assessments that test knowledge at the macro-scale, the micro-scale, and specific concepts (such as natural selection or phylogenetic tree thinking), but there are no current assessments which hone in on skills associated with learning evolution between scales. Based on the LBECT-LP, computational knowledge may be assessed conceptually on the four computational components: input, integration, output, and feedback. Not only could students be tested on these four concepts, but can also be assessed using written components which integrate these four components with the content knowledge. For example, students may be asked: "In your population model depicting evolution, what is the input, output, integration, and feedback," these types of questions would directly assess the computational process as well as the computational product (from the LBECT-LP). Results from these assessments may be scaled using the simple, developing, and complex model from the LBECT-LP, and as defined by the instructor for the specific computational tasks.

Future research could involve developing specific interventions that use computational thinking to teach scientific content. Assessments required in future research may test the specific content knowledge (i.e., biological evolution) and computational thinking knowledge. The results from these interventions may be compared to groups of students who have not received the computational interventions to identify if and how computational thinking contributes to learning. Although such an experimental study has not been published at this time, the first author performed a pilot study to begin validating data collection instruments (measuring both knowledge of evolution and computation) and inform two computational interventions in the context of biological evolution. Each assessment was given to each student three times, one in the form of a pretest, one after the first computational intervention, and one after the second computational intervention. Results indicated there was a strong correlation between the pretest computational knowledge scores and the posttest 2 computational knowledge scores, posttest intervention 1 computational scores and the posttest intervention 2 computational scores, and posttest 1 and posttest 2 evolution scores. Follow-up univariate tests indicated that there was a significant increase in biology knowledge scores over time, with large effect size; but no significant difference in computational scores over time. Based on these results, the first author made adjustments to the interventions to more explicitly emphasize computational reasoning based on the LBECT-LP, and conducted a follow-up, quasi-experimental study. Initial results revealed that of the two computational interventions, one was successful in producing in biological knowledge gains over time, suggesting that the instructional context and computational process were influential on participant knowledge gains between the intervention groups. Analysis of participant artifacts revealed that biological levels identified and biological level connections made by participants also differed between the two computational intervention groups. Participants who had stronger gains in evolution knowledge made more connections across biological levels.

There are a variety of reasons why computational thinking is not supported in biology classrooms including teacher preparation and their comfort with the technology and concepts. Frequently, schools and districts lack sufficient funding for technology, hardware, software, and professional development required for teachers to become properly trained using these tools. Teachers and administrators are frequently unaware of the potential programs that are available, how the programs may be integrated into curriculums, and the variety of disciplines that may utilize the programs. There are some programs such as R or C++ that are free for users, but require a large amount of time in order to learn. With this robust source, teachers would need to become very knowledgeable in developing activities and properly scaffolded lessons that merge content with the interfaces. Many of these realizations arose from planning the quasi-experimental design studies with high school and college teachers and administrators.

If teachers are open to using computational thinking in their classroom, in many cases, they view it as a package (that is interface heavy) for one lesson and learn applications for specific concepts. However, they may not see computation as a robust tool that may be developed and used as a working biological instrument to be used across various applications. Most software that is user friendly and appealing to teachers is specific for performing one type of activity. Although the software itself may be advanced, it can be expensive and is counter intuitive to our proposed learning progression. For example, an electronic probe that measures and displays conductivity can only be used to measure and display conductivity in the way that it was written into the software. We consider these types of activities simple within the learning progression. As students use inexpensive or free programs such as R or C++, it would allow them to take the conductivity data (obtained from a probe for example) and imagine and write [code for] the ways it might be manipulated or displayed. These types of activities lend toward the developing (or complex) levels of the LP. It may be difficult for educators to identify appropriate software, and specifically recognize how they may use the software to scaffold lessons through the simple, developing, and complex components of the learning progression.

It takes a great deal of time for students, teachers, and professionals in the field to learn computer languages. The payoff for learning these languages is great because the tools are then available for future use. The component of language learning is displayed by the computational complexity aspect of the learning progression. Computational processes are difficult to assess, and teachers may have difficulty in developing rubrics; however, these rubrics may be made more concrete through the use of our LP. Interpretation of assignment requirements may also be difficult or frustrating for students.

In many cases, the first time students encounter programming, especially to find a means to an answer questions on their own, it is at the masters or independent research level of education. Independent thinking in which students develop tools or discover new knowledge should not be an experience reserved for graduate level students. Often, lack of computational thinking skills is the reason students have difficulty in independent scientific thinking. Specific implementation of computational thought processes in biology classrooms has not been explored quantitatively or qualitatively. It is unknown how computational processes will help students to better understand evolution and in turn how this might strengthen their knowledge within the domain of biology as a whole. Using computation should strengthen student knowledge and NOS processes, but it is unclear in what ways. Students may become more comfortable developing new computational tools or applying these skills to other disciplines. It is also unknown how computational processes explored through this type of learning progression may relate to overall student achievement or collaborative learning outcomes.

9 Conclusion

Along with teacher apprehension to teaching evolution, educators are also apprehensive to tools associated with computational thinking; this may be alleviated with the assistance of the LBECT-LP. For example, there is positive and strong association between teachers' science education and experiences as a scientist and likelihood and degree to which evolution is taught in their classrooms (Nehm et al. 2009). Not exposing our young science learners to computational thinking at an early age is a major disservice to them and may hinder the future of STEM fields. Computational methods incorporated into biology classes to learn biological evolution is controversial; however, it has the potential to alleviate misconceptions, reinforce the nature of science, and encourage student embodiment. The learning progression we provided may be used by a variety of educational, science, and computational professionals to consider the most appropriate application and integration (of the related concepts). The LBECT-LP would encourage students to practice thought processes that may encourage higher-order science understanding and promote curricular relationships across disciplinary domains. As teachers implement these strategies, it would provide them with a new robust resource and alternative ways to promote learning more similar to those in the science field.

Increased scientific literacy that promotes equitable problem-solving and action to address local, regional, and global challenges requires improved science learning (McDonald 2016). However, as the sciences have become more cross disciplinary, there have been little methods of implementation that are domain content heavy. The trends of coding and STEM in general are becoming more popular for students; however, it is important that the content is implemented in ways that would be most beneficial to all students, which include merging these ideas with their core content subjects. Although our argument to incorporate computational thinking is directly related to science courses, and specifically biology, the NGSS stresses the concept across various STEM fields. Not only should computational thinking be stressed in science courses but in all courses which emphasize STEM and the cross-cutting concepts. Students should envision computational thinking and its associated tools as processes they can use outside of computer science courses. As these technologies and associated processes gain popularity and become implemented, it is imperative that they are actually promoting student success within the STEM fields. Understanding the learning processes, hardships, and benefits of these processes is essential for continued growth within STEM education to promote success among all types of students.

Although the theoretical framework lends toward a full learning progression, very small components (e.g., simple computation from the LBECT-LP) can be initially used by districts to faciliate implementation in its entirety. It is difficult for researchers to provide particular examples of interventions for educators because teachers have different training and teaching styles, and many districts have limited resources (e.g., computers and software). Within the NGSS, the standards allow teachers freedom to develop classroom activities, just as this LBECT-LP would allow freedom for teachers to develop their own interventions and written assessment methods to integrate computation. Because biological evolution is specific, other [scientific] topics taught computationally may have their own set of NGSS standards as well as learning objectives that would need to be aligned with different LPs. This would take extensive knowledge of a computational expert coupled with scientists and educational professionals. In order to alleviate this, computation should be taught during college courses (for prospective teachers) through genuine research experiences in their science or pedagogical courses. However, computational integration is not frequent in these courses and ironically part of our argument to advocate for computation earlier in education. Lack of computational knowledge and resources limits teachers, prospective teachers, and paradoxically even those within the scientific community.

The LBECT-LP provides specific content (unity and diversity), state standards, and (computational) strategies for biology instructors or those in closely related fields. It promotes holistic type learning and structure for biology teachers to implement lessons which are new, cross-curricular, and engaging. Not only would it promote students to behave more like scientists, but it would also encourage this type of teaching from instructors. It may also inform teacher trainings, curriculum writers, science learning companies, technology companies, and computational biologists and or encourage their interaction. Because computational thinking is emphasized in the NGSS, it is applicable to other scientific fields. Computational thinking may also be used to reinforce NOS processes in other science disciplines.

Getting individuals to think like scientists early on in education is critical for democratic societies. This is especially true with the rise in AI, bioinformatics, and threat of abstractions becoming indistinguishable from reality. A scientifically minded type of population may better understand and critically evaluate itself and its role in the natural world. Allowing students to develop creative and critical viewpoints contributes to resisting anti scientific schools of thought (Longbottom and Butler 1999). Using logic, creating variables, and tools to simulate reality is inseparable from computational thinking; therefore, it is essential that we provide this opportunity for students in order provide them the best education possible within their biology courses.

Compliance with Ethical Standards

Conflict of Interest The authors declare that they have no conflict of interest.

References

Aho, A. V. (2012). Computation and computational thinking. *Computer Journal, 55*(7), 832–835. https://doi.org/10.1093/comjnl/bxs074.

Aikenhead, G. (2007). Expanding the research agenda for scientific literacy. Promoting scientific literacy. *Science Education Research in Transaction*, 64.

Alexander, P. A. (2018). Information management versus knowledge building: Implications for learning and assessment in higher education. *Methodology of Educational Measurement and Assessment*, 43–56. https://doi.org/10.1007/978-3-319-74338-7_3.

Alters, B. J., & Nelson, C. E. (2002). Perspective: Teaching evolution in higher education. *Evolution, 56*(10), 1891–1901. https://doi.org/10.1111/j.0014-3820.2002.tb00115.x.

American Association for the Advancement of Science. (1990). Project 2061 (AAAS). Science for All Americans. https://doi.org/10.1177/027046769001000206.

Antolin, M. F., Jenkins, K. P., Bergstrom, C. T., Crespi, B. T., De, S., et al. (2012). Evolution and medicine in undergraduate education: A prescription for all biology students. *Evolution, 66*(6), 1991–2006. https://doi.org/10.1111/j.1558-5646.2011.01552.x.

Ayala, F. J. (1977). "Nothing in biology makes sense except in the light of evolution" Theodosius Dobzhansky: 1900–1975. *Journal of Heredity, 68*(1), 3–10. https://doi.org/10.1093/oxfordjournals.jhered.a108767.

Barr, V., & Stephenson, C. (2011). Bringing computational thinking to K-12: What is involved and what is the role of the computer science education community? *ACM Inroads, 2*(1), 48–54. https://doi.org/10.1145/1929887.1929905.

Basawapatna, A. R., Repenning, A., Koh, K. H., & Nickerson, H. (2013). The zones of proximal flow: Guiding students through a space of computational thinking skills and challenges. In *Proceedings of the ninth annual international ACM conference on International computing education research* (pp. 67-74).

Baudrillard, J. (1994). *Simulacra and simulation*. University of Michigan press.

Beggrow, E. P., & Sbeglia, G. C. (2019). Do disciplinary contexts impact the learning of evolution? Assessing knowledge and misconceptions in anthropology and biology students. *Evolution: Education and Outreach, 12*(1), 1.

Berland, L. K., & McNeill, K. L. (2010). A learning progression for scientific argumentation: Understanding student work and designing supportive instructional contexts. *Science Education, 94*(5), 765–793. https://doi.org/10.1002/sce.20402.

Berland, M., & Wilensky, U. (2015). Comparing virtual and physical robotics environments for supporting complex systems and computational thinking. *Journal of Science Education and Technology, 24*(5), 628–647. https://doi.org/10.1007/s10956-015-9552-x.

Britner, S. L. (2008). Motivation in high school science students: A comparison of gender differences in life, physical, and earth science classes. *Journal of Research in Science Teaching, 45*, 955–970. https://doi.org/10.1002/tea.20249.

Britner, S. L., & Pajares, F. (2006). Sources of science self-efficacy beliefs of middle school students. *Journal of Research in Science Teaching, 43*(5), 485–499. https://doi.org/10.1002/tea.20131.

Broughton, S. H., Sinatra, G. M., & Nussbaum, E. M. (2013). "Pluto has been a planet my whole life!" Emotions, attitudes, and conceptual change in elementary students' learning about Pluto's reclassification. *Research in Science Education, 43*(2), 529–550. https://doi.org/10.1007/s11165-011-9274-x.

Brown, M. H., & Schwartz, R. S. (2009). Connecting photosynthesis and cellular respiration: Preservice teachers' conceptions. *Journal of Research in Science Teaching, 46*(7), 791–812.

Brumby, M. (1979). Problems in learning the concept of natural selection. *Journal of Biological Education, 13*(2), 119–122. https://doi.org/10.1080/00219266.1979.9654240.

Bybee, R. W. (2011, December). Scientific and engineering practices in K-12 classrooms: Understanding a framework for K-12 science education. *The Science Teacher, 78*(9), 34 Retrieved from http://link.galegroup.com.libproxy.temple.edu/apps/doc/A274521193/AONE?u=temple_main&sid=AONE&xid=8073c03a.

Campbell, N. A., Mitchell, L. G., Reece, J. B., & Taylor, M. R. (2000). *Biology: Concepts & connections*. Menlo Park: Benjamin Cummings.

Catley, K., Lehrer, R., & Reiser, B. (2005). Tracing a prospective learning progression for developing understanding of evolution. *Paper Commissioned by the National Academies Committee on test design for K-12 Science achievement,* 67. Retrieved from https://s3.amazonaws.com/academia.edu.documents/35284613/Paper_Commissioned_by_the_National_Academies_Committee_on_Test_Design_for_K-12_Science_

Achievement_2005_Catley.pdf?AWSAccessKeyId=AKIAIWOWYYGZ2Y53UL3A&Expires=1535338785&Signature=q0AjjacnesePFecYJlLbwMUUlw8%3D&response-content-disposition=inline%3B%20filename%3DTracing_a_prospective_learning_progressi.pdf.
Chandrasekharan, S., & Nersessian, N. J. (2015). Building cognition: The construction of computational representations for scientific discovery. *Cognitive Science, 39*(8), 1727–1763. https://doi.org/10.1111/cogs.12203.
Chi, M. T., Roscoe, R. D., Slotta, J. D., Roy, M., & Chase, C. C. (2012). Misconceived causal explanations for emergent processes. *Cognitive Science, 36*(1), 1–61. https://doi.org/10.1111/j.1551-6709.2011.01207.x.
Cohen, J. E. (2004). Mathematics is biology's next microscope, only better; biology is mathematics' next physics, only better. *PLoS Biology, 2*(12), 439. https://doi.org/10.1371/journal.pbio.0020439.
Coley, J. D., Arenson, M., Xu, Y., & Tanner, K. D. (2017). Intuitive biological thought: Developmental changes and effects of biology education in late adolescence. *Cognitive Psychology, 92*, 1–21. https://doi.org/10.1016/j.cogpsych.2016.11.001.
College Board. (2009). *AP® Biology Lab Manual for Students*. New York: The College Board.
College Board. (2012). *AP Biology: Course and exam description effective fall 2012*. New York: The College Board.
Concord Consortium. (2018). *Research projects: Biology*. Retrieved from https://concord.org/our-work/research-projects/ on April 22.
Cordova, J. R., Sinatra, G. M., Jones, S. H., Taasoobshirazi, G., & Lombardi, D. (2014). Confidence in prior knowledge, self-efficacy, interest and prior knowledge: Influences on conceptual change. *Contemporary Educational Psychology, 39*(2), 164–174. https://doi.org/10.1016/j.cedpsych.2014.03.006.
Dada, J. O., & Mendes, P. (2011). Multi-scale modelling and simulation in systems biology. *Integrative Biology, 3*(2), 86–96. https://doi.org/10.1039/c0ib00075b.
Denning, P. J. (2017). Remaining trouble spots with computational thinking. *Communications of the ACM, 60*(6), 33–39. https://doi.org/10.1145/2998438.
Denning, P. J., & Freeman, P. A. (2009). The profession of IT Computing's paradigm. *Communications of the ACM, 52*(12), 28–30. https://doi.org/10.1145/1610252.1610265.
Denning, P. J., Martell, C. H., & Cerf, V. (2015). *Great principles of computing*. MIT Press Retrieved from https://books.google.com/books?hl=en&lr=&id=AjFkBgAAQBAJ&oi=fnd&pg=PR7&dq=Denning,+P.+J.,+Martell,+C.+H.,+%26+Cerf,+V.+(2015).+Great+principles+of+computing.+MIT+Press.&ots=mDb0z-aapQ&sig=CoAHAqh01P3c8ul3YguCzhFTrWE#v=onepage&q&f=false.
Dobzhansky, T. (1973). Nothing in biology makes sense except in the light of evolution. *The American Biology Teacher, 35*(3), 125–129. https://doi.org/10.2307/4444260.
Dole, J. A., & Sinatra, G. M. (1998). Reconceptalizing change in the cognitive construction of knowledge. *Educational psychologist, 33*(2-3), 109-128. https://doi.org/10.1207/s15326985ep3302&3_5.
Dole, J., Sinatra, G. M., & Reynolds, R. (1991). The effects of strong beliefs on text processing: The case of evolution and creationism. Paper presented at the National Reading Conference Annual Meeting Palm Springs, CA.
Driver, R., Asoko, H., Leach, J., Scott, P., & Mortimer, E. (1994). Constructing scientific knowledge in the classroom. *Educational Researcher, 23*(7), 5–12. https://doi.org/10.3102/0013189X023007005.
Duit, R., & Treagust, D. F. (2003). Conceptual change: A powerful framework for improving science teaching and learning. *International Journal of Science Education, 25*(6), 671–688. https://doi.org/10.1080/09500690305016.
Duit, R., & Tesch, M. (2010). *On the role of the experiment in science teaching and learning–visions and the reality of instructional practice*. In M. Kalogiannakis, D. Stavrou, & P. G. Michaelides (Eds.).
Duit, R. H., & Treagust, D. F. (2012). Conceptual change: Still a powerful framework for improving the practice of science instruction. In *Issues and challenges in science education research* (pp. 43–54). Netherlands: Springer. https://doi.org/10.1007/978-94-007-3980-2_4.
Duit, R., Gropengießer, H., Kattmann, U., Komorek, M., & Parchmann, I. (2012). The model of educational reconstruction–a framework for improving teaching and learning science. In *Science education research and practice in Europe* (pp. 13–37). Rotterdam: Sense Publishers.
Dunbar, K. (2000). How scientists think in the real world: Implications for science education. *Journal of Applied Developmental Psychology, 21*(1), 49–58. https://doi.org/10.1016/S0193-3973(99)00050-7.
Dunbar, K. N., & Klahr, D. (2012). Scientific thinking and reasoning. In K. J. Holyoak & R. G. Morrison (Eds.), *The Oxford handbook of thinking and reasoning* (pp. 701–718). New York: Oxford University Press.
Duschl, R. A. (2012). The second dimension—Crosscutting concepts. *Science and Children, 49*(6), 10–14 Retrieved from http://link.galegroup.com.libproxy.temple.edu/apps/doc/A280005164/AONE?u=temple_main&sid=AONE&xid=9d53f1e4.
Duschl, R. A., & Grandy, R. (2013). Two views about explicitly teaching nature of science. *Science & Education, 22*(9), 2109–2139. https://doi.org/10.1007/s11191-012-9539-4.

Duschl, R., Maeng, S., & Sezen, A. (2011). Learning progressions and teaching sequences: A review and analysis. *Studies in Science Education, 47*(2), 123–182. https://doi.org/10.1080/03057267.2011.604476.

Edelson, D. C. (1998). Realising authentic science learning through the adaptation of scientific practice. International handbook of science education, 1, 317–331. Retrieved from http://www.growingtallpoppies.com/wp-content/uploads/2015/06/handbook.pdf.

Fisher, M. E., Giakoumis, S., & Dorner, M. A. (2016). The benefits of an engaged constructivist pedagogy in the sciences.

Forrest, S. (1990). Emergent computation: Self-organizing, collective, and cooperative phenomena in natural and artificial computing networks. *Physica D, Nonlinear Phenomena, 42*(1–3), 1–11. https://doi.org/10.1016/0167-2789(90)90063-U.

Gadanidis, G. (2017). Artificial intelligence, computational thinking, and mathematics education. *The International Journal of Information and Learning Technology, 34*(2), 133–139. https://doi.org/10.1108/IJILT-09-2016-0048.

Gallagher, S. R., Coon, W., Donley, K., Scott, A., & Goldberg, D. S. (2011). A first attempt to bring computational biology into advanced high school biology classrooms. *PLoS Computational Biology, 7*(10), e1002244. https://doi.org/10.1371/journal.pcbi.1002244.

Gee, H. (2013). *The accidental species: Misunderstandings of human evolution*. Chicago: University of Chicago Press.

Gelman, S. A. (2003). *The essential child: Origins of essentialism in everyday thought*. New York: Oxford University Press.

Gobert, J. D., & Buckley, B. C. (2000). Introduction to model-based teaching and learning in science education. *International Journal of Science Education, 22*(9), 891–894. https://doi.org/10.1080/095006900416839.

Griffith, J. A., & Brem, S. K. (2004). Teaching evolutionary biology: Pressures, stress, and coping. *Journal of Research in Science Teaching, 41*(8), 791–809. https://doi.org/10.1002/tea.20027.

Gross, L. J. (2004). Points of view: The interface of mathematics and biology: Interdisciplinarity and the undergraduate biology curriculum: Finding a balance. *Cell Biology Education, 3*(2), 85–87. https://doi.org/10.1187/cbe.04-03-0040.

Grover, S. (2011). *Robotics and engineering for middle and high school students to develop computational thinking*. Paper presented at the annual meeting of the American Educational Research Association, New Orleans, LA.

Grover, S., & Pea, R. (2013). Computational thinking in K–12: A review of the state of the field. *Educational Researcher, 42*(1), 38–43. https://doi.org/10.3102/0013189X12463051.

Guo, Y., Wagh, A., Brady, C., Levy, S. T., Horn, M. S., & Wilensky, U. (2016). Frogs to think with: Improving Students' computational thinking and understanding of evolution in a code-first learning environment. In *Proceedings of the 15th International Conference on Interaction Design and Children* (pp. 246-254). ACM. https://doi.org/10.1145/2930674.2930724.

Heddy, B. C., & Nadelson, L. S. (2013). The variables related to public acceptance of evolution in the United States. *Evolution: Education and Outreach, 6*(1), 1–14. https://doi.org/10.1186/1936-6434-6-3.

Heddy, B. C., & Sinatra, G. M. (2013). Transforming misconceptions: Using transformative experience to promote positive affect and conceptual change in students learning about biological evolution. *Science Education, 97*(5), 723–744. https://doi.org/10.1002/sce.21072.

Holyoak, K. J., & Morrison, R. G. (2012). *The Oxford handbook of thinking and reasoning*. New York: Oxford University Press.

Hongladarom, S. (2006). Ethics of bioinformatics: A convergence between bioethics and computer ethics. *Asian Biotechnology and Development Review, 9*(1), 37–44. https://doi.org/10.1007/978-3-642-23960-1_36.

Ioannidou, A., Bennett, V., Repenning, A., Koh, K. H., & Basawapatna, A. (2011). Computational. Thinking Patterns, In *Proceedings of the 2011 Annual Meeting of the American. Educational Research Association (AERA)*, New Orleans, Louisiana, USA, April 66–12,

Jacobson, M. J., & Wilensky, U. (2006). Complex systems in education: Scientific and educational importance and implications for the learning sciences. *The Journal of the Learning Sciences, 15*(1), 11–34. https://doi.org/10.1207/s15327809jls1501_4.

Jördens, J., Asshoff, R., Kullmann, H., & Hammann, M. (2016). Providing vertical coherence in explanations and promoting reasoning across levels of biological organization when teaching evolution. *International Journal of Science Education, 38*(6), 960–992.

Keil, F. C. (2006). Explanations and understanding. *Annual Review Psychology, 57*, 227–254. https://doi.org/10.1146/annurev.psych.57.102904.190100.

Kenyon, L., Schwarz, C., & Hug, B. (2008). The benefits of scientific modeling. *Science and Children, 46*(2), 40–44 Retrieved from http://web.a.ebscohost.com.libproxy.temple.edu/ehost/pdfviewer/pdfviewer?vid=1&sid=601d4d8c-5e34-4402-bce5-e7221b49dc63%40sessionmgr4009.

Ketelhut, D. J. (2007). The impact of student self-efficacy on scientific inquiry skills: An exploratory investigation in River City, a multi-user virtual environment. *Journal of Science Education and Technology, 16*(1), 99–111. https://doi.org/10.1007/s10956-006-9038-y.

King, R. D. (2011). Rise of the robo scientists. *Scientific American, 304*(1), 72–77. https://doi.org/10.1038/scientificamerican0111-72.

Kong, S. C., & Abelson, H. (Eds.). (2019). *Computational thinking education*. Springer.

Krajcik, J. (2012). Engaging students in scientific practices: What does constructing and revising models look like in the science classroom? *Science and Children, 49*(7), 10–13 Retrieved from http://web.a.ebscohost.com.libproxy.temple.edu/ehost/pdfviewer/pdfviewer?vid=1&sid=3e42c09a-bc72-4394-a2d9-d97e2bb16bda%40sessionmgr4010.

Latour, B. (2005). *Reassembling the social: An introduction to actor-network-theory*. Oxford: Oxford University Press.

Lawson, A. E., & Worsnop, W. A. (1992). Learning about evolution and rejecting a belief in special creation: Effects of reflective reasoning skill, prior knowledge, prior belief and religious commitment. *Journal of Research in Science Teaching, 29*(2), 143–166. https://doi.org/10.1002/tea.3660290205.

Lawson, A. E., Banks, D. L., & Logvin, M. (2007). Self-efficacy, reasoning ability, and achievement in college biology. *Journal of Research in Science Teaching, 44*(5), 706–724. https://doi.org/10.1002/tea.20172.

Lederman, N. G. (2019). Contextualizing the relationship between nature of scientific knowledge and scientific inquiry. *Science & Education, 28*, 249–267. https://doi.org/10.1007/s11191-019-00030-8.

Lederman, J. S., Lederman, N. G., Bartos, S. A., Bartels, S. L., Meyer, A. A., & Schwartz, R. S. (2014). Meaningful assessment of learners' understandings about scientific inquiry—The views about scientific inquiry (VASI) questionnaire. *Journal of Research in Science Teaching, 51*(1), 65–83.

Lehrer, R., & Schauble, L. (2006). *Cultivating model-based reasoning in science education*. New York: Cambridge University Press.

Lenhard, J. (2010). Computation and simulation. *The Oxford handbook on interdisciplinarity*, 246–258.

Leutenegger, S., & Edgington, J. (2007). A games first approach to teaching introductory programming. In *ACM SIGCSE Bulletin* (Vol. 39, no. 1, pp. 115-118). ACM. https://doi.org/10.1145/1227310.1227352.

Levinson, R. (2010). Science education and democratic participation: An uneasy congruence? *Studies in Science Education, 46*(1), 69–119.

Liao, Y. W., & She, H. C. (2009). Enhancing eight grade students' scientific conceptual change and scientific reasoning through a web-based learning program. *Educational Technology & Society, 12*(4), 228–240 Retrieved from http://web.b.ebscohost.com.libproxy.temple.edu/ehost/pdfviewer/pdfviewer?vid=2&sid=003d3cbb-123e-452c-806a-3a9cd761b647%40pdc-v-sessmgr04.

Llewellyn, D. (2012). *Teaching high school science through inquiry and argumentation*. New York: Corwin Press.

Longbottom, J. E., & Butler, P. H. (1999). Why teach science? Setting rational goals for science education. *Science Education, 83*(4), 473–492. https://doi.org/10.1002/(SICI)1098-237X(199907)83:4<473::AID-SCE5>3.0.CO;2-Z.

Lye, S. Y., & Koh, J. H. L. (2014). Review on teaching and learning of computational thinking through programming: What is next for K-12? *Computers in Human Behavior, 41*, 51–61. https://doi.org/10.1016/j.chb.2014.09.012.

Maltese, A. V., & Tai, R. H. (2011). Pipeline persistence: Examining the association of educational experiences with earned degrees in STEM among US students. *Science Education, 95*(5), 877–907. https://doi.org/10.1002/sce.20441.

Marr, D. (1977). Artificial intelligence—a personal view. *Artificial Intelligence, 9*(1), 37–48. https://doi.org/10.1016/0004-3702(77)90013-3.

Mau, W. C. (2003). Factors that influence persistence in science and engineering career aspirations. *The Career Development Quarterly, 51*(3), 234–243. https://doi.org/10.1002/j.2161-0045.2003.tb00604.x.

McCain, K., & Kampourakis, K. (2018). Which question do polls about evolution and belief really ask, and why does it matter? *Public Understanding of Science, 27*(1), 2–10. https://doi.org/10.1177/0963662516642726.

McComas, W. F. (1996). Ten myths of science: Reexamining what we think we know about the nature of science. *School Science and Mathematics, 96*(1), 10–16. https://doi.org/10.1111/j.1949-8594.1996.tb10205.x.

McDonald, C. V. (2016). STEM Education: A review of the contribution of the disciplines of science, technology, engineering and mathematics. *Science Education International, 27*(4), 530–569.

Mead, R., Hejmadi, M., & Hurst, L. D. (2018). Scientific aptitude better explains poor responses to teaching of evolution than psychological conflicts. *Nature Ecology & Evolution, 2*(2), 388. https://doi.org/10.1038/s41559-017-0442-x.

Moore, T. J., Tank, K. M., Glancy, A. W., & Kersten, J. A. (2015). NGSS and the landscape of engineering in K-12 state science standards. *Journal of Research in Science Teaching, 52*(3), 296–318. https://doi.org/10.1002/tea.21199.

National Research Council. (2010). *Exploring the intersection of science education and 21st century skills: A workshop summary*. National Academies Press.

National Research Council. (2012). *A framework for K-12 science education: Practices, crosscutting concepts, and core ideas*. National Academies Press.

Nehm, R. H., Kim, S. Y., & Sheppard, K. (2009). Academic preparation in biology and advocacy for teaching evolution: Biology versus non-biology teachers. *Science Education, 93*(6), 1122–1146. https://doi.org/10.1002/sce.20340.

Nelson, C. E., Scharmann, L. C., Beard, J., & Flammer, L. I. (2019). The nature of science as a foundation for fostering a better understanding of evolution. *Evolution: Education and Outreach, 12*(1), 6. https://doi.org/10.1186/s12052-019-0100-7.

Newell, A., Perlis, A. J., & Simon, H. A. (1967). Computer Science. *Science, 157*(3795), 1373–1374.

NGSS Lead States. (2013). *Next generation science standards: For states, by states*. National Academies Press. https://doi.org/10.17226/18290.

Parker, J. M., Anderson, C. W., Heidemann, M., Merrill, J., Merritt, B., Richmond, G., & Urban-Lurain, M. (2012). Exploring undergraduates' understanding of photosynthesis using diagnostic question clusters. *CBE—Life Sciences Education, 11*(1), 47–57.

Peel, A., Sadler, T. D., & Friedrichsen, P. (2019). Learning natural selection through computational thinking: Unplugged design of algorithmic explanations. *Journal of Research in Science Teaching, 56*(7), 983–1007. https://doi.org/10.1002/tea.21545.

Perlis, A. J. (1982). Special feature: Epigrams on programming. *ACM SIGPLAN Notices, 17*(9), 7–13. https://doi.org/10.1145/947955.1083808.

Pezzulo, G., Barsalou, L. W., Cangelosi, A., Fischer, M. H., McRae, K., & Spivey, M. J. (2013). *Computational grounded cognition: a new alliance between grounded cognition and computational modeling, Frontiers in Psychology*, 3. https://doi.org/10.3389/fpsyg.2012.00612.

Piaget, J. (1976). *Piaget's theory*, *In Piaget and his school (pp. 11–23)*. Berlin, Heidelberg: Springer.

Pintrich, P. R., Marx, R. W., & Boyle, R. B. (1993). Beyond cold conceptual change: The role of motivational beliefs and classroom contextual factors in the process of conceptual change. *Review of Educational Research, 63*, 167–199. https://doi.org/10.3102/00346543063002167.

Plummer, J. D. (2012). Challenges in defining and validating an astronomy learning progression. In *Learning progressions in science* (pp. 77–100). Brill Sense.

Plummer, J. D., & Krajcik, J. (2010). Building a learning progression for celestial motion: Elementary levels from an earth-based perspective. *Journal of Research in Science Teaching, 47*(7), 768–787. https://doi.org/10.1002/tea.20355.

Pluta, W. J., Chinn, C. A., & Duncan, R. G. (2011). Learners' epistemic criteria for good scientific models. *Journal of Research in Science Teaching, 48*(5), 486–511. https://doi.org/10.1002/tea.20415.

Pobiner, B., Beardsley, P. M., Bertka, C. M., & Watson, W. A. (2018). Using human case studies to teach evolution in high school AP biology classrooms. *Evolution: Education and Outreach, 11*(1), 3. https://doi.org/10.1186/s12052-018-0077-7.

Posner, G. J., Strike, K. A., Hewson, P. W., & Gertzog, W. A. (1982). Accommodation of a scientific conception: Toward a theory of conceptual change. *Science Education, 66*(2), 211–227. https://doi.org/10.1002/sce.3730660207.

Pratt, H., & Bybee, R. W. (2012). *The NSTA reader's guide to a framework for K-12 science education*. Arlington: NSTA Press.

Qin, H. (2009). Teaching computational thinking through bioinformatics to biology students. In *ACM SIGCSE Bulletin* (Vol. 41, no. 1, pp. 188-191). ACM.

Román-González, M., Moreno-León, J., & Robles, G. (2017). Complementary tools for computational thinking assessment. *In Proceedings of International Conference on Computational Thinking Education (CTE 2017), S. C Kong, J Sheldon, and K. Y Li (Eds.). The Education University of Hong Kong* (pp. 154-159)

Ryder, J., Leach, J., & Driver, R. (1999). Undergraduate science students' images of science. *Journal of Research in Science Teaching: The Official Journal of the National Association for Research in Science Teaching, 36*(2), 201–219. https://doi.org/10.1002/(SICI)1098-2736(199902)36:2<201::AID-TEA6>3.0.CO;2-H.

Schmaltz, R. M., Jansen, E., & Wenckowski, N. (2017). Redefining critical thinking: Teaching students to think like scientists. *Frontiers in Psychology, 8*, 459. https://doi.org/10.3389/fpsyg.2017.00459.

Schwarz, C. V., Reiser, B. J., Davis, E. A., Kenyon, L., Achér, A., Fortus, D., et al. (2009). Developing a learning progression for scientific modeling: Making scientific modeling accessible and meaningful for learners. *Journal of Research in Science Teaching, 46*(6), 632–654. https://doi.org/10.1002/tea.20311.

Schweingruber, H., Thomas, K., & Helen, Q. (2012). *A framework for K-12 science education: Practices, crosscutting concepts, and core ideas*. National Academies Press.
Selby, C., & Woollard, J. (2013). *Computational thinking: the developing definition* (6pp). University of Southampton (E-prints).
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. https://doi.org/10.1007/s10639-012-9240-x.
Shen, B. S. (1975). Science literacy and the public understanding of science. In *Communication of scientific information* (pp. 44-52). Karger Publishers.
Sherman, M., & Martin, F. (2015). The assessment of mobile computational thinking. *Journal of Computing Sciences in Colleges, 30*(6), 53–59 Retrieved from http://www.cs.uml.edu/~fredm/papers/p53-sherman.pdf.
Sinatra, G. M., Heddy, B. C., & Lombardi, D. (2015). The challenges of defining and measuring student engagement in science.
Sinatra, G. M., & Lombardi, D. (2020). Evaluating sources of scientific evidence and claims in the post-truth era may require reappraising plausibility judgments. *Educational Psychologist, 55*(3). https://doi.org/10.1080/00461520.2020.1730181.
Sinatra, G. M., Southerland, S. A., McConaughy, F., & Demastes, J. W. (2003). Intentions and beliefs in students' understanding and acceptance of biological evolution. *Journal of Research in Science Teaching, 40*(5), 510–528. https://doi.org/10.1002/tea.10087.
Sinatra, G. M., Brem, S. K., & Evans, E. M. (2008). Changing minds? Implications of conceptual change for teaching and learning about biological evolution. *Evolution: Education and Outreach, 1*(2), 189–195. https://doi.org/10.1007/s12052-008-0037-8.
Singh, K., Granville, M., & Dika, S. (2002). Mathematics and science achievement: Effects of motivation, interest, and academic engagement. *The Journal of Educational Research, 95*(6), 323–332. https://doi.org/10.1080/00220670209596607.
Speth, E. B., Long, T. M., Pennock, R. T., & Ebert-May, D. (2009). Using Avida-ED for teaching and learning about evolution in undergraduate introductory biology courses. *Evolution: Education and Outreach, 2*(3), 415–428.
Stocum, D. L. (2015). Reflections on scientific literacy, worldviews, and education. *Journal of Civic Literacy, 2*(1), 4–23.
Tamatea, L. (2019). Compulsory coding in education: Liberal-humanism, Baudrillard and the 'problem' of abstraction. *Research and Practice in Technology Enhanced Learning, 14*(1), 1–29. https://doi.org/10.1186/s41039-019-0106-3.
The AP Biology Development Committee. (2012). *AP biology investigative labs: An inquiry-based approach*. New York: The College Examination Board.
Toffoli, T. (2004). Nothing makes sense in computing except in the light of evolution. *IJUC, 1*(1), 3–29 Retrieved from http://research.cs.queensu.ca/home/akl/cisc879/papers/PAPERS_FROM_UNCONVENTIONAL_COMPUTING/VOLUME_1_Issue_1/TOFFOLI.pdf.
Vosniadou, S., Vamvakoussi, X., & Skopeliti, I. (2008). The framework theory approach to the problem of conceptual change. In S. Vosniadou (Ed.), *International handbook of research on conceptual change* (pp. 3–34). New York: Routledge, Taylor, & Francis Group.
Vygotsky, L. S. (1962). *Thought and language*. Cambridge: MIT Press.
Vygotsky, L. (1978). Interaction between learning and development. *Readings on the Development of Children, 23*(3), 34–41 Retrieved from http://www.psy.cmu.edu/~siegler/vygotsky78.pdf.
Wachowski, A., Wachowski, L., Reeves, K., Fishburne, L., Moss, C.-A., Warner Bros., Village Roadshow Pictures, Silver Pictures. (1999). *The matrix*. Burbank: Warner Home Video.
Weintrop, D., Beheshti, E., Horn, M., Orton, K., Jona, K., Trouille, L., & Wilensky, U. (2016). Defining computational thinking for mathematics and science classrooms. *Journal of Science Education and Technology, 25*(1), 127–147. https://doi.org/10.1007/s10956-015-9581-5.
Welcome to NCBI. (2019). Retrieved from: https://www.ncbi.nlm.nih.gov/.
Werner, L., Denner, J., Campe, S., & Kawamoto, D. C. (2012). The fairy performance assessment: Measuring computational thinking in middle school. In *Proceedings of the 43rd ACM technical symposium on Computer Science Education* (pp. 215-220). ACM https://doi.org/10.1145/2157136.2157200.
Wilensky, U. (2014). *Computational thinking through modeling and simulation*. Whitepaper presented at the summit on future directions in computer education. Orlando, FL.
Wilensky, U., & Papert, S. (2010). Restructurations: Reformulations of knowledge disciplines through new representational forms. Constructionism. Retrieved from https://pdfs.semanticscholar.org/a684/d3149266ce9e55dc366b8acbbc6e905f8c80.pdf.

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. https://doi.org/10.1207/s1532690xci2402_1.

Wing, J. M. (2006). Computational thinking. *Communications of the ACM, 49*(3), 33–35 Retrieved from https://www.microsoft.com/en-us/research/wp-content/uploads/2012/08/Jeannette_Wing.pdf.

Wood, W. B. (2009). Revising the AP biology curriculum. *Science, 325*(5948), 1627–1628. https://doi.org/10.1126/science.1180821.

Zirbel, E. L. (2004). Framework for conceptual change. *Astronomy Education Review, 3*(1), 62–76. https://doi.org/10.3847/AER2004007.