Building an Automated Scientist within a 9th grade Biology class
Exploring Gene Combinations through Python Programming
Lesson plan and implementation process

Brenda McGarry-Rivera
El Paso Leadership Academy East HS
2023 CREEDS Summer Institute
The University of Texas of El Paso

Introduction:
The proposed lesson plan aims to bridge the gap between biology and computer science while encouraging students to explore various fields within STEM (Science, Technology, Engineering, and Mathematics). By integrating computer science into the biology curriculum, we seek to create an engaging learning environment that fosters curiosity and interest in both subjects.

Purpose:
The primary goal of this implementation is twofold: to expose students to diverse career paths within STEM and to cultivate their interest in computer science. By incorporating coding exercises and data analysis into biology lessons, we aim to demonstrate the practical applications of technology in scientific research. Additionally, we hope to instill in students the importance of adapting to technological advancements and equip them with the skills needed for future academic and professional pursuits.

Learning and Language Objectives:

1. Introduce students to the interdisciplinary connections between biology and computer science.
2. Foster curiosity and interest in computer science through hands-on coding activities.
3. Provide students with practical experience in utilizing technology for scientific analysis and problem-solving.
4. Encourage critical thinking and creativity in applying coding skills to biological concepts.
5. Prepare students for future academic endeavors by equipping them with relevant technological skills.

Implementation:

The lesson plan will incorporate interactive activities, coding exercises, and real-world examples to engage students in exploring the intersection of biology and computer science. By utilizing platforms such as Google Colab and Code for Life, students will have the opportunity to analyze genetic data, predict gene combinations, and gain insights into the practical applications of coding in biological research. Furthermore, collaborative projects and discussions will encourage students to think critically about the role of technology in shaping the future of scientific inquiry.

Conclusion:

Through this interdisciplinary STEM exploration, we aim to inspire students to pursue careers in fields such as bioinformatics, computational biology, and data science. By fostering a curiosity-driven learning environment and equipping students with valuable coding skills, we hope to empower the next generation of STEM innovators and problem-solvers.

Funding Allocation:

Part of the stipend previously received will be allocated towards purchasing licenses for Quizizz and Google Colab to enhance interactive learning and collaborative coding experiences. As well as, for color and poster size printing visual aids, diagrams, concept tables, and instructional materials related to gene combinations and Python programming.
Implementation date:

The lesson will be implemented between mid-May and the first week of June, coinciding with the completion of students' STAAR tests. Additionally, it will align with our campus's project-based learning week, the timing of which is pending approval from the administrative department. Furthermore, high-performing students will be divided into two separate groups to participate in this unit.

2 weeks lesson plan:

Week 1:

Day 1-2:

Will be utilized to review basic genetics concepts and vocabulary through interactive activities using Quizlet and Quizizz:

This vocabulary will include the following words:

1) Gene: A unit of heredity that is passed from parent to offspring and determines certain traits.
2) Allele: Alternative forms of a gene that can result in different variations of a trait.
3) Genotype: The genetic makeup of an organism, typically represented by letters (e.g., AA, Aa, or aa).
4) Phenotype: The observable physical characteristics or traits of an organism, influenced by its genotype and environmental factors.
5) Dominant: A trait that is expressed when present, even if only one allele is inherited.
6) Recessive: A trait that is expressed only when two copies of the recessive allele are present.
7) Homozygous: Having two identical alleles for a particular gene (e.g., AA or aa).
8) Heterozygous: Having two different alleles for a particular gene (e.g., Aa).
9) Punnett Square: A diagram used to predict the possible genetic outcomes of a cross between two individuals.
10) Genotype ratio: The ratio of different genotypes that result from a genetic cross.
11) Phenotype ratio: The ratio of different phenotypes that result from a genetic cross.
12) Monohybrid cross: A genetic cross that involves only one trait.
13) Dihybrid cross: A genetic cross that involves two traits.
14) Inheritance: The passing of genetic traits from parent to offspring.
15) Mutation: A change in the DNA sequence of a gene that can result in altered traits or characteristics.
16) Homologous chromosomes: Paired chromosomes with genes for the same traits arranged in the same order.
17) Chromosome: Thread-like structures in the nucleus of a cell that contain genetic information (DNA).
18) Genetic variation: Differences in genetic makeup between individuals of the same species.

To continue with an Introduction to the concept of gene combinations and Punnett squares. A discussion of the importance of connecting computer science and biology. And most importantly, administer a pre-assessment to gauge students' understanding of genetics concepts and basic Python programming.

Day 3-4:

- These days, an introduction to basic computer science concepts and Python programming language, once again utilizing quizizz to help students get familiar
with the following vocabulary:
1) Algorithm: A sequence of steps designed to solve a specific problem or perform a task.
2) Variable: A named storage location in computer memory used to store data that can change during program execution.
3) Data Type: Specifies the type of data that can be stored in a variable, such as integers, floats, strings, or booleans.
4) Conditional Statement: A statement that evaluates a condition and executes a block of code if the condition is true.
5) Loop: A control flow statement that repeatedly executes a block of code as long as a specified condition is true.
6) Function: A reusable block of code that performs a specific task. Functions can accept input parameters and return output.
7) List: A data structure that stores a collection of items in a specific order. Lists are mutable and can contain elements of different data types.
8) Dictionary: A data structure that stores key-value pairs, allowing efficient lookup and retrieval of values based on keys.
9) String: A sequence of characters enclosed in quotation marks. Strings are used to represent text data in Python.
10) Boolean: A data type that represents logical values of true or false.
11) Input/Output (I/O): The process of transferring data to and from a computer system, often involving user input and program output.
12) Comment: An annotation in the source code of a program that provides additional information or explanations for humans reading the code.

The use of visual aids such as diagrams and block-based coding apps/websites like Turtle will be essential to reinforce programming concepts.

The teacher will connect biology concepts to coding by exploring how Python can be used to simulate genetic crosses and predict outcomes.

**Week 2:**
Day 1-2:

- Review Google Colab and its features.
- Revisit previously used data from the Keystone species unit and Genetics: Gene combinations unit.
- Discuss the importance of data collection in science and other fields.

Day 3-4:

- Students will engage in a training classifier activity to understand the basics of classifier models.
- Pose a question for the activity: "Does the extinction of a species have a positive or negative consequence on the environment?"
- Practice basic coding skills using Turtle and Code for Life.

Day 5:

- Introduce a more complex exercise where students will use Python to predict gene combinations instead of using Punnett squares.
- Provide genetic data from Kaggle and guide students through coding exercises to predict possible outcomes of genetic crosses.
- Reflect on the connections between biology and computer science and the importance of interdisciplinary approaches in research.

Growth and understanding measurements:

- Pre-assessment: Assess students' understanding of genetics concepts and basic Python programming.
- Formative assessments: Monitor students' progress during coding exercises and activities.
- Summative assessment: Evaluate students' ability to predict gene combinations using Python coding in Google Colab.
• Reflection: Have students reflect on the connections between biology and computer science and the significance of interdisciplinary approaches.