Specific comments for this assignment:

While life sciences students have studied transcription and translation, perhaps multiple times, students from other disciplines (e.g., computer science) will find the terminology daunting (i.e., biologists are "encyclopedic"). That said, actualizing the series of steps in an algorithm is foreign to many students and this assignment provides an excellent example of the benefit of writing and rewriting an algorithm.

Have students work the steps of transcription and translation in pairs by hand, preferably on the board. Students enjoy making up new examples for others to try. If possible, working the problem by hand in interdisciplinary pairs shows the students by example why multiple perspectives are important when problem solving in the genomics space.

The provided starter kit contains a number of user-defined functions. Students must fully appreciate the mechanics of function calling and return at this point. How many and how much of each these functions that are fully or partially written beforehand by the instructor is an important pedagogical decision. This assignment is significantly more involved than the previous two and the starter kit as delivered is my current best estimate for level of difficulty for the undergraduates I teach.

User-defined functions are defined in a small module (BioDNA). I find students appreciate the example of working on their "own" module.

I use three-letter abbreviations for amino acids in this assignment. I have found the three-letter code helps students in their debugging since the lengths of the translated mRNA and resulting amino acid string are of equal lengths. Instructors will note inline documentation that allows an easy switch to the more common one-letter amino acid code.

The starter kit performs the transitions between reading frames in a sequential fashion. I have found that asking students at this point in the semester to implement this in a loop adds a level of difficulty that is best left for the next assignment. Other instructors may disagree and of course the code in the starter kit can be modified quite easily if desired.

The use of a dictionary data structure in this assignment is appropriate, but a reach for most novice programmers at this stage. That said, a "walkthrough" of the makeAminoAcidTable() and translate() functions offers an excellent teaching moment, especially for advanced students.

General comments for the entire set of DNA-focused programming assignments:

(0) Regularly allocate time in your class sessions to bring in colleagues, in particular, a biologist who can talk (briefly) of the beauty of DNA. In this case, the central dogma of biology is a central theme in biology lectures but this programming assignment helps actualize the processes. Having a guest lecturer provides excellent reinforcement for asking students to grapple with the fine details of each step.

(1) Give students a sense of the real demand for future scientists who can work in multidisciplinary groups on computationally-intensive problems. Don't be shy to share that the skills to be learned in this course can "get students a job" at the start of an exciting career. Many students in the life sciences have general plans to "be a doctor"; these assignments may be an undergraduate's first exposure to the excitement in and demand for computational scientists in regards to research and medicine.

(2) Use a "flipped classroom" where students watch lectures before/after class in order to maximize the amount of hands-on Python play. For example, (a) require students to watch lectures on biological topics (cf. Udacity's 'Tales of the Genome' MOOC) outside of class; and (b) leverage programming practice sites for student practice outside of class (cf. codeAcademy's Python course)