DNA Project 7

This will be our last procedural project… after that, we're all objects, all the time! The idea behind this assignment is to give you some more practice with arrays and to introduce you to the exciting field of bioinformatics. Save your project in a file called DNA.java. And as always, carefully follow our Project Correctness Checklist and Coding Conventions to receive full points. **Our Write to File page will help in getting your output to a file.**

Brief description of what you'll do

Your goal will be to write a program which reads in an input file containing named sequences of nucleotides and can produce information about each of them. Specifically, for each nucleotide sequence, you will:

- count the occurrences of each of the four nucleotides (A, C, G, T)
- calculate the mass percentage occupied by each nucleotide type, rounded to one digit past the decimal point
- report the codons (trios of nucleotides) present in each sequence
- predict whether or not the sequence encodes a protein using a series of heuristics

For our purposes, a sequence encodes a protein if it meets the following constraints:

- begins with a valid start codon (ATG)
- ends with a valid stop codon (TAA, TAG, or TGA)
- contains at least 4 codons, including its start and stop codons
- at least 30% of its mass is Cytosine and Guanine

To compute mass percentages, use the following as the mass of each nucleotide:

- Adenine (A): 135.128
- Cytosine (C): 111.103
- Guanine (G): 151.128
- Thymine (T): 125.107

Note that these are approximations designed for this assignment and not exact constraints used in computational biology!

Requirements

For this assignment you should have at least four class constants to make your program more readable and modifiable. **Please order your class constants as they are shown here to aid in auto grading;** note only the first two will be modified during grading:

- minimum number of codons a valid protein must have, as an integer (default of 4)
- percentage of mass from C and G in order for a protein to be valid, as an integer (default of 30)
- number of unique nucleotides (4, representing A, C, G, and T – this will NOT be changed for testing, but shows good design for scalability)
- number of nucleotides per codon (3 – also will not be changed, but should be a constant to design for scalability, thanks)

You must eliminate redundancy as much as possible throughout the program. Specifically, **you should use the following arrays**:

- nucleotide counts
- mass percentages
- codons

DNA Project 7

You should use methods in this program to provide structure and avoid redundancy. For full credit on this program, you must have at least four non-trivial methods other than main. You should decide for yourself exactly what methods to use, but for full credit, your methods should obey constraints such as (but not necessarily limited to) the following:

- No one method should be overly long.
- Each significant array in your program should be filled with data in its own individual method.
- Looking solely at main should present a clear idea of all of the major tasks your program is doing.
- Your methods should accept arrays as parameters or return arrays as their result values as appropriate.
- **ALL OUTPUT MUST BE CREATED IN A SINGLE OUTPUT METHOD THAT TAKES ALL THE COMPUTED DATA AS PARAMETERS!!** (note this is bold and in capitals, so it is very important)

You should follow past stylistic guidelines about indentation, whitespace, identifiers, and localizing variables. You should place comments at the beginning of your program, at the start of each method, and on complex sections of code.

A sample run

Assume you have a data file named dna.txt with the following content:

```
cure for cancer protein
ATGCCACTATGGTAG
captain picard hair growth protein
ATgCCAACATGgATGCCcGATAtGGATTgA
bogus protein
CCATtAATgATCaCAGTt
```

The user should be prompted in the following way for input and output files (user input underlined):
This program reports information about DNA nucleotide sequences that may encode proteins.

```
Input file name? dna.txt
Output file name? output.txt
You should then produce the following output in the file output.txt:
Name: cure for cancer protein
Nucleotides: ATGCCACTATGGTAG
Nucleotide counts: [4, 3, 4, 4]
Mass percentages: [27.3, 16.8, 30.6, 25.3]
Codons: [ATG, CCA, CTA, TGG, TAG]
Encodes a protein: yes

Name: captain picard hair growth protein
Nucleotides: ATGCCAACATGGATGCCCGATATGGATTGA
Nucleotide counts: [9, 6, 8, 7]
Mass percentages: [30.7, 16.8, 30.5, 22.1]
Codons: [ATG, CCA, ACA, TGG, ATG, CCC, GAT, ATG, GAT, TGA]
Encodes a protein: yes

Name: bogus protein
Nucleotides: CCATTAATGATCACAGTT
```

DNA Project 7

```
Nucleotide counts: [6, 4, 2, 6]
Mass percentages: [35.1, 19.3, 13.1, 32.5]
Codons: [CCA, TTA, ATG, ATC, ACA, GTT]
Encodes a protein: no
```

Download dna.txt with its corresponding output. Also try ecoli.txt with output here. Use quickdiff to compare your output.

Implementation hints

- Start by dealing with file input
- Read each protein's name and sequence of nucleotides and print them
- Count As, Cs, Gs and Ts and put counts in an array of size 4 (can use charAt)
- Convert counts into mass percentages
- Revisit String methods to break sequence into codons and examine codons (you can also find them on our Syntax sheet)
- Read up about how to Write to File on this linked page. Please use `PrintStream` to accomplish this

Some biology

(Optional details, the science behind this problem) Deoxyribonucleic acid (DNA) is a complex biochemical macromolecule that carries genetic information for cellular life forms and some viruses. DNA consists of long chains of chemical compounds called nucleotides. Four nucleotides are present in DNA: Adenine (A), Cytosine (C), Guanine (G), and Thymine (T). DNA has a double-helix structure containing complementary chains of these four nucleotides connected by hydrogen bonds.

Certain portions of the DNA contain nucleotide sequences called genes, most of which encode instructions for building proteins. These proteins are responsible for carrying out most of the life processes of the organism. DNA is also the mechanism through which genetic information from parents is passed on during reproduction.

The nucleotides in a gene are organized into groups of three, called codons. Codons are typically written as the first letters of their three nucleotides, such as TAC or GGA. Each codon uniquely encodes a single amino acid, which is a building block of proteins.

The process of building proteins from DNA has two major phases called transcription and translation, in which a gene is replicated into an intermediate form called mRNA, which is then processed by a structure called a ribosome to build the chain of amino acids encoded by the codons of the gene.

The ranges of DNA that encode proteins occur between a start codon (which we will assume to be ATG) and a stop codon (which is any of TAA, TAG, or TGA). Not all regions of DNA contain protein-encoding genes; large portions that do not lie between a valid start and stop codon are called intergenic DNA and have other (possibly unknown) function.

Computational biologists examine large DNA data files to find patterns and important information, such as which regions encode particular proteins. They are also sometimes interested in the percentages of mass accounted for by each of the four nucleotide types. Often a high percentage of Cytosine (C) and Guanine (G) are found in regions of the DNA that contain important genetic data.

Project graciously borrowed from UW CSE 142.