Lab 2-2: DNA Analysis: DNA Manipulation, Codon Usage Bias and Gene Density

Due date: Thursday, April 12, end of the lab period..

Please note, **Lab 3** instructions will be given to you on *April 12*. The initial stage of **Lab 3** will involve CSC teams only, your **CHEM 441** colleagues won't work with you on **Lab 3** until *April 17*.

About the Lab

This is a joint lab. Each **CSC 448** team will continue working with their **CHEM 441** teammates from **Lab 2-1.** This lab continues the path of DNA analysis. This is also your first "long" (i.e., multi lab period) lab.

The tentative plan for this lab is a s follows.

Time	CSC 448	CHEM 441
April 5 lab	Studying assignment	Studying assignment
April 5 lab	Discuss assignment/map out solution	
April 5 – 10	Software development	Data collection
April 10 lab	Assembly, testing, modification, and use of software	
April 10 – 11	Preparing deliverables	
April 12	Submission of deliverables	Submission of deliverables

Lab Assignment

In **Lab 2-1** you have collaborated on building a program that computes the GC-content of a given DNA sequence. GC-content is just one of the measures that biologists use in analyzing individual DNA fragments and comparing multiple fragments to each other. Other such measures (codon usage bias, gene density, enthropy, etc.) have been discussed in class.

In this lab, you will do the following:

1. Create a library of DNA string manipulation and analysis functions[1] that will be used to complete the other parts of this lab as well as in the labs to come.
2. Working with your **CHEM 441** teammates, improve, refactor[2] and enhance the GC-content computation program from **Lab 2-1** to produce the GC-content computation for sliding frames in a given DNA sequence.
3. Working with your **CHEM 441** teammates, design and implement a collection of software (that uses the functionality of your DNA analysis library) that performs the specific analyses requested by the **CHEM 441** students.

DNA Manipulation and Analysis Library

The DNA manipulation and analysis library shall contain the functionality we have discussed in lecture. In particular, the following shall be contained in it:

1. Complementing nucleotides.
2. Constructing a reverse complement of a DNA sequence.
3. Computing GC-content of a DNA sequence.
4. Computing various measures of codon usage bias: frequency of optimal codons, **RCSU** of an amino acid, Codon Adaptation Index (CAI), effective number of codons, scaled χ^2.
5. Computing Shannon's information enthropy of a DNA sequence.
6. Computing various versions of gene density estimators.

[1] I will use the term functions when referring to a single unit of functionality that needs to be implemented as a separate unit of code. In different programming languages such units of code are referred to as functions, procedures, classes and methods, routines and so on. We will use the word function to represent any of these generically.

[2] To make the core functionality a part of your library.

Your library should also contain some *reusable* functionality for parsing DNA sequences out of FASTA files.

Please note, that some of the computations will require input information beyond the input DNA sequence. Your **CHEM 441** teammates will discuss with you the structure of the other data files that will be used throughout this lab (and some future labs).

General Notes on Working on Software Requirements

These comments are based on the observations of your interactions with **CHEM 441** students during **Lab 2-1** and on the results of these interactions.

Lab 2-2 assumes that a lot of information about the software to be built will be provided for you by your **CHEM 441** teammates. In fact, the final sets of programs each team will produce may be somewhat different due to the decisions **CHEM 441** members of the team make about the way in which they intend to analyze their data.

In your discussions, please pay specific attention to the following aspects of the program design (this is **general advice, applicable to all programs for this and future labs**):

- **Input specifications.** Make your your **CHEM 441** teammates provide exact descriptions of the input data your program(s) will take.
- **Customization.** As we all know, the same program, with the help of a few settings and/or parameters can produce a number of different things (e.g., the **Lab 2-1** GC-content program became much more functional once the start and end position paremeters were introduced and the program became able to compute GC-content of any substring in the input DNA sequence).

  Please, **discuss** with your **CHEM 441** teammates all customization requirements and specifications. They might not understand that certain things are possible, so, if you believe you can simplify their work by adding some customization parameters[3], please be proactive about it. Remember that one of the goals of each lab is for you to deliver software that **CHEM 441** students will use to actually perform research tasks.
- **Output specifications.** Please be aware that the data your program outputs may need to be used in some other software. E.g., the output of the new **GC-content** program may need to be rendered as a graph using, for example, MS Excel. Discuss the format of the output — **especially for programs that are supposed to produce a lot of output** — with your **CHEM 441** teammates. If necessary implement

[3] Most of the parameters are supposed to be discussed in **CHEM 441** lecture, but some **CHEM 441** teams may miss them, or not realize how to pass this information properly onto you.

dual output functionality: dumping output in human readable form on the screen (or into a GUI), **and** dumping it into a file that can be later used with third-party software.

GC-content computation

In lecture we discussed the issue of tracking the change in GC-content of a DNA sequence over the length of the sequence. The ability to spot fragments of a large DNA sequence that have unusually high or unusually low GC-content is an important aid to the analytical tasks **CHEM 441** students must perform for their **Lab 2-2** deliverables.

Discuss with your **CHEM 441** teammates the design of the new program for computing GC-content over a sliding window. The **CHEM 441** students should have the requirements specifications for this version of the program ready for you.

Refactor and improve your **Lab 2-1** GC-content computation program to address the new set of specifications and deliver it to your **CHEM 441** teammates.

DNA Sequence Analysis

Your **CHEM 441** teammates will need to compute codon usage bias and gene density in order to successfully complete their analytical tasks for the lab. Discuss with them which specific measures they want to use, as well as how they want to use the software the produces this information for them.

Using the library of the DNA manipulation and analysis tools that you will implement for this lab, implement the programs that perform the tasks **CHEM 441** students ask you for. Make certain you negotiate the appropriate input and output formats prior to building the software.

Note: some of the functionality to develop for the DNA manipulation and analysis library may be left unused in this lab. However, some unused functionality will become used at a later stage (e.g., *enthropy* will be featured in **Lab 3** if it is not needed for this lab).

GUI Use

There is no outright requirement that your programs use GUI. However, for some of them, a graphical user interface can significantly improve the experiences of **CHEM 441** students with the software. If you are using Java, you can use the GUI code provided to you for **Lab 2-1** as a template, and fill it as necessary. If you are using other programming languages, you can simply extend the GUI of your final **Lab 2-1** submission.

Discuss the use of GUI with your **CHEM 441** partners. Explain the trade-offs associated with building GUI-enabled software to them (the main trade-off being - longer development time). If GUI is desired, determine if a command-line prototype with full/partial functionality needs to be delivered before full GUI-enabled program is completed.

Submission Instructions

These instructions are for your graded deliverables for **CSC 448**. **CHEM 441** students have their own set of deliverables: they rely on being able to run your software to produce them.

Use handin to submit all your files. For your submission, prepare a `README` file, which you will submit separately from the rest of the files (the rest of the files will be submitted as a single archive). The `README` file shall contain the following information:

1. **Team name.** (Yes! time to pick a team name is NOW!)
2. List of team members.
3. Description of submitted files. List the filenames, and explain what these files are. If your submission has a lot of file - no need to explain every one of them, **but** you must provide a list of compilable/runnable files and explain what they do.
4. Instructions on comipling (if needed) and running all programs that you submit. This should include both the compilation instructions and the instructions (including and command-line parameters) for running the program.
5. A brief description of the DNA manipulation and analysis library that you created. Include the list of classes (if applicable) and functions/methods implemented, and brief descriptions of each. If using a language capable of generating automatic documentation at compilation time, you can submit the automatically generated documentation (e.g., Javadocs) and simply reference the file name/names in the `README` file.
6. A brief description of the work breakdown on the CS side of the team. Specify, which parts of the software were developed jointly, and which parts were developed separately. (This is for my information only, this information will not be used for grading purposes).

In addition to the `README` file, submit the following:

1. Requirements specifications that you received from your **CHEM 441** teammates. There may be multiple requirements specs for different tasks within the lab: submit all of them.
2. All code your developed for this lab.
3. Any additional files (documentation, data, unit tests, etc...) you deem necessary. If included, specify what these files are in the `README` file.

Put all your files except for the `README` file into a single `.zip` or `.tar.gz` archive, called `lab02-2.zip` or `lab02-2.tar.gz`. Submit the archive and the `README` file using the following `handin` command. Only one submission per team, please!

Use the following command to submit:

```
$handin dekhtyar 448-lab2-2 <files>
```

PS. You **may** be asked to also submit, either in hardcopy, or in softcopy, your team's evaluation of requirements specifications received from your **CHEM 441** teammates. One of the learning objectives of **CHEM 441** is to teach students to write good requirements for bioinformatics software. This requires helpful feedback. If we do decide to go with requirements evaluation on this lab, the evaluation form(s) and instructions on how to fill them out will be given to you separately.