BCB 444 Fall 2015 Project 6

Preparing raw DNA sequences for genome assembly

Due Oct. 21 at 1 p.m.

The jobs of this project are to analyze the DNA trace files produced by Sanger sequencing machines, determine the correct DNA bases of the DNA templates being sequenced, and to trim the raw DNA sequences according to their quality values and any matches to vector sequence fragments. This will *clean up* the data for the next project, in which you will attempt to assemble the sequences and (hopefully) obtain the original genomic sequences. Optionally, in this project you will also learn how to obtain, unpack, compile and install bioinformatic software. Different software tools have different installation difficulties. In this project, we choose one of the easiest software to install. Being able to download and install bioinformatic software is essential to any bioinformaticists, but it's an optional job in this project.

To summarize, Step I of the project asks you to visually inspect a few trace files and correct some unknown (N) bases manually. This will help you appreciate the difficulties in calling correct DNA bases and recognize that genomic data can always be uncertain. Step II of the project requires calling the bases of thousands of trace files automatically where manual corrections are no longer feasible, and extracts other useful information (e.g., the base-call quality values) from the trace files. The base-called sequences and quality data will be used in Step III to trim off low quality sequences in preparation for the genome assembly project next week. In Step IV, you will optionally learn to download and install the quality trimming software yourself instead of relying on the already installed version.

You will need to use your Class Account to carry out this project and to submit your results. Therefore, if you have not tried to login your accounts, you should do it to verify that you have no access problems. Reading of some bioinformatic software documents indicated in blue text below is needed for this and future projects, so you can learn on your own how to execute bioinformatic software correctly. Reading software documentation is considered part of the purposes of our project assignments, therefore you should not skip that effort and only copy the commands demonstrated in the class without knowing what they are doing. That being said, you may wish to follow the command line examples given in the lab tutorials if you are unsure about the correct usage of some software. Your knowledge about some most common bioinformatics software may be tested in the next midterm.

Step I Manual inspection of DNA trace files

Four DNA sequencing trace files are provided with this project description on Blackboard Learn. Download the files and utilize one of the two trace file viewers *4Peaks* or *CodonCode Aligner* already installed on the lab iMac computers to view these files. You may use other trace file viewers available to your computer platform to accomplish the same task if you like, but of course the TA won't be providing support for the installation and use of the other viewer software. Some free trace file viewers for different platforms are listed here:

http://seqcore.brcf.med.umich.edu/usechrom.html

The provided trace files were obtained from plasmids (small circular DNAs) with important clones inserted between base-pair positions 50-240 on the sequences. Unfortunately, these clones contained strong hairpin structures that can disturb the normal electrophoresis of labeled DNA molecules during sequencing runs, thus causing some unknown bases (N bases) to be called by the sequencing machine software. Your job is to inspect the four trace files and manually correct the N bases to one of the 4 known nucleotide bases A, C, G or T within the range of base locations 50-240 on each sequence. You may ignore all other N bases outside this range, but the whole sequence should be saved to the output (i.e., do NOT trim your output sequences).

Job 1: (4 pt)

Correct those N bases between 50-240 bp and save your output into a **single** multi-FASTA file for project submission. Name the FASTA file "`job1.seq`" and include the 4 corrected sequences named >seq1, >seq2, >seq3 and >seq4 in this file for submission.

Step II Automatic determination of DNA bases

Data for the second part of the project is stored in the `~cs444/pub/data/easy` and `~cs444/pub/data/moderate` directories that are accessible from your Class Account.

Warning! Do not copy the trace files to your home directories. This is especially important later when you handle next-generation sequence data; You may easily run out of your disk quota that way! Process the trace files in their existing directories within the class account, and save only the output to your own directories.

If you are unsure how to login to your Class Account, read the Blackboard Learn announcement or post your questions to its Project Discussion forum. Some jobs of this and future projects can be computationally intensive. In order to avoid disturbing other users of the shared Class Account server, and hopefully to finish your own jobs quicker, it is recommended that you perform the following check before you start a computational job:

Before you start your jobs, find out the CPU count by typing the "**top**" command, then press 1, and then note the number of CPU's listed. You may 'q'uit the "**top**" command and then type the "**w**" command to check the current job load. If the load average on the machine is more than the CPU count, **do not** run your jobs just yet. Wait a few minutes and check again. If the job load is heavier than the CPU can handle, **everyone's** jobs will have to wait. Waiting for a lighter job load period (usually early in weekend mornings) to run your jobs is a smart move, not to mention it's also a healthy life style. ☺

Job 2: (2 pt)

Obtain **phred** base-calls and quality values from the trace files stored in `~cs444/pub/data/easy`. Read the **phred** document in `~cs444/pub/doc` to determine the appropriate command line parameters to use. Store the sequence and quality value output into two multi-FASTA files in **your own directory**. We will need these files in later projects, so keep them after you finish this project.

Note: Do NOT run the **phred** command inside the Class Account public directories and attempt to generate output there; because you cannot write your output files there, the **phred** program will likely crash if you do this!

Repeat the above for more DNA trace files stored in `~cs444/pub/data/moderate`. Do not mix the data you obtained from the "moderate" date set with the data you obtained from the "easy" data set earlier; save them into different directories and/or files.

For both the "easy" and "moderate" data set base-calls, just submit the **sorted FASTA headers** of the DNA sequences from **phred**, like the following. **Do not** submit either the sequences themselves or their companion quality values! We do not need your DNA or quality data in this stage to grade your project submission. Your assembled genomic sequences can give us more information about the quality of your work later. Named your header files "`job2_easy_header.txt`" and "`job2_moderate_header.txt`" and include them in your project submission.

```
>ATNMA01TR 970 0 970 ABI
>ATNMA02TR 1002 0 1002 ABI
>ATNMA03TF 854 0 854 ABI
>ATNMA03TR 867 0 867 ABI
>ATNMA04TR 1045 0 1045 ABI
>ATNMA05TF 858 0 858 ABI
>ATNMA06TF 869 0 869 ABI
>ATNMA06TR 988 0 988 ABI
>ATNMA07TF 873 0 873 ABI
>ATNMA07TR 1046 0 1046 ABI
```

Step III   Trim raw DNA sequences for assembly

With **phred** base-calls and quality values obtained from the trace files in `~cs444/pub/data/easy` during the previous step, you may now use Lucy to select a useful data range of each sequence for genome assembly. Although you can use either the command-line or the GUI version of Lucy to perform this task, this project focuses on the command-line Lucy because that is the one that allows automation (i.e., allows Lucy to be part of a larger automatic genomic data processing pipeline). Read the document of Lucy to determine the appropriate command line parameters to use. The document was already downloaded with the command-line Lucy, or you can find it in the Class Account folder below:

```
less ~cs444/pub/doc/lucy.doc
```

In addition to trimming based on quality values, you also need to trim off vector fragments and remove vector contaminants from the data set using the PUC19 and PUC19splice files stored in the `~cs444/pub/data` directory. Again, read the Lucy document to determine how to use these two files to trim vector fragments. Note the term "trimming" here simply means to mark the beginning and ending of the good quality range along a sequence. We do not mean to physically remove the DNA bases outside of this good quality range. Each input sequence to the subsequent fragment assembly process (i.e, Project 7) should have the same length as before, except that they are now marked in their header line for their good quality region. Note also that the version of Lucy preinstalled in the class account is older and may not produce the same output from the command-line Lucy you are supposed to install in the following optional **Step IV** of the project, therefore do not use your installed Lucy for this Step or you may get different results.

Additional Notes: The version of **phred** we installed in the class account tends to call quality values lower than expected. To cope with this situation, you may wish to modify Lucy's quality cutoff criteria controlled by the `-error` option to increase the useful data range. However, blindly lowering the cutoff criteria will result in bad sequences getting into the genomic data assembly step, which can make your assembly effort in Project 7 more difficult. This option will become useful when you work on the next project, but for now, you don't need it. For project submission purpose, you should generate the debug information file when running Lucy. See below.

After you finish the quality trimming and vector removal of raw sequences obtained from the 'easy' data set, do the same job above for the 'moderate' data set obtained from trace files in `~cs444/pub/data/moderate`. Avoid mixing data from the easy data set with data from the moderate data set by saving them into different files and/or directories.

Job 3: (2 pt)

Keep the sequence data you obtained in this project for future use. For this Step, submit the debug information files `job3_easy.info` and `job3_moderate.info` obtained from each data set. Again, do *not* submit sequence or quality value files, but do save them for the next project.

(OPTIONAL STEP)

Step IV Download and install bioinformatic software

This project requires the use of the Lucy sequence quality trimming and vector removal software program. This program has two flavors, a command-line version that has been in use dating back to 1997, and a graphical user interface version that was recently refreshed. In the previous step you used the version we have already installed into the Class Account. However, if you become an independent bioinformaticist like you will have to do the installation yourself. As a practice for bioinformatics software installations, we will try to install both the latest command-line and GUI Lucy versions and compare their output to see if there is any difference.

To obtain the command-line Lucy, go to the following website, find the latest source code release and download it to your computer:

```
http://lucy.sourceforge.net/
```

Locate the download file `lucy1.20.tar.gz` on your computer. To compile the source code into a working program, you need a C compiler. On Linux, you should already have that. On Mac you may already have it, but if typing the '`gcc -v`' command results in '`command not found`' message, you need to install Apple's Xcode development software, and then `gcc` will become available. On Windows, your best choice will be the `gcc` compiler bundled in **cygwin** because even if you install the Microsoft Visual C++ compiler, it may not be compatible with the Unix oriented command-line Lucy program. It is beyond the scope of this project description to tell you exactly how you may obtain your C compiler for the various platforms you may have access to, but feel free to ask the TA or the instructor during the lab session if you need some helps on this.

Once you have located or installed a C compiler, type the following command (within a command terminal of course) to unpack the Lucy source code bundle:

```
tar zxvf lucy1.20.tar.gz
```

After the unpacking, you should find a new directory which includes, among many other files, the README.FIRST text file. Follow the **Installation** instructions in that file to compile Lucy. You may not be able to install the resulted executable program `lucy` to the suggested directory `/usr/local/bin`, but you can run the program where it is located without hindering the following tests. Finally, follow the **Testing** instructions also contained in that document file to test your newly compiled Lucy program. Save the produced `job4_lucy.info` file for project submission later.

In the next you should download and install the graphic user interface (GUI) version of Lucy2 as well. Go to the following website, sign up for a free account, then follow the instructions to download and install Lucy2 (the '2' stands for 'too' and it signals that it's the GUI version of Lucy):

http://www.complex.iastate.edu/download/Lucy2/index.html

There are three flavors of Lucy2 for the Linux, Mac and Windows platforms, and each may include 32- or 64-bit versions depending on the platform. Choose the right version(s) to download to your computer and then install it. You do not need to install all three flavors or download the source code release because compiling it from the source code will be much harder than the command-line version and is entirely beyond the scope of this project.

In Lucy2 distribution, there is also a README.txt text file. Locate that file and follow the **Basic operations** in it to test Lucy2 on the same provided set of test files. It is important that you do NOT trim the sequences (i.e., skip Step 3: Actual Trimming) in the operation instructions. Save your output and choose the '`lucy2`' filename. Lucy2 will then produce the output files `lucy2.seq`, `lucy2.qul` and `lucy2.info`. You may rename the last file to `job4_lucy2.info` for submission.

Job 4: (3 pt)

You need to submit **sorted versions** of the `job4_lucy.info` file and `job4_lucy2.info` file. The former is obtained from the command-line Lucy, and the later is obtained from the GUI Lucy2. Add a '`job4_compare.txt`' file in your submission, and in that file you should write either 'Lucy and Lucy2 produced the same trimming results' or 'Lucy and Lucy2 produced different trimming results'. You may use the '`sort`' command to sort the two files and the '`diff`' command to compare the two sorted files to determine what conclusion you should put into this `job4_compare.txt` file.

Files to submit for grading

Here we summarize files you should submit for grading in this project and how you should have obtained them. The total project score is 8 points, and you may optionally earn another 3 points if you complete Step IV and submit its output files correctly. The extra 3 points can be applied toward this or previous project deficiencies. However, the extra points are not applicable toward future projects. Each correctly produced output file is worth 1 point except `job1.seq`, which contains 4 sequences and is worth 4 points.

`job1.seq`	The manually inspected and corrected sequences >seq1, >seq2, >seq3 and >seq4
`iob2_easy_header.txt`	Header lines for phred base-called sequences from the easy data set
`job2_moderate_header.txt`	Header lines for phred base-called sequences from the moderate data set
`job3_easy.info`	Obtained in Step III of the project when running the class account provided Lucy program on the raw DNA sequences obtained from the **easy** data set
`job3_moderate.info`	Similar data obtained from the **moderate** data set in Step III
The following job files are optional	
`job4_lucy.info`	The debug information file obtained by following the command-line Lucy README.FIRST file instructions when operating on the provided '**atie**' set of files
`job4_lucy2.info`	The debug information file obtained by following the graphical user interface Lucy2 README.txt file instructions (but skip Step 3: Actual Trimming) when operating also on the provided '**atie**' set of files
`job4_compare.txt`	Your conclusion if Lucy and Lucy2 produced the same output or not; this is written by you based on the differences in output

Remember the two job 4 info files must be sorted before submission (consider using the Unix **sort** command). Copy the `job1.seq,job4_lucy2.info` and other files you generated on your local computers to the Class Account in order to submit them together with the other output files. Use the **submit** command to submit your project files. Read project submission instructions on Blackboard Learn to learn how to use the **submit** command to submit project files. You will still need to *submit* a simple notice about your project on Blackboard Learn saying you are done with the project and have submit it in the Class Account so the TA can get a grading form on Blackboard Learn to record your project score.

Any questions or comments about this project are welcome, but preferably they should be posted to the Project Discussion forum on the Blackboard Learn or asked during the lab session, so other students can share the answers.