L519: Bioinformatics: Theory & Application (3CR)

HW1 (Due: **Sep. 16 BEFORE** Lab session)

http://darwin.informatics.indiana.edu/col/courses/L519

INTRODUCTION:

There are two sessions to be completed. The session 1 is for Perl programming and the session 2 is for algorithm. In order to submit your completed homework (Session 1), please use drop box at the Oncourse. Though you may turn in handwritten session 2 at the lab class, using MS Word (doc), Acrobat (pdf), PostScript (ps) is strongly encouraged. These files can also be submitted through Oncourse.

QUESTION:

Don't hesitate to contact me (Haixu Tang : hating@indiana.edu) or AI (Junguk Hur : juhur@indiana.edu).

INSTRUCTION:

1. Please start to work the homework as soon as possible. For some of you without enough computational background may need much more time than others.
2. Include **README** file for each programming assignment. This is not supposed to be lengthy but should contain concrete and enough information;
   A. Function of the script
   B. Input / Output
   C. Sample usage
3. You should submit a single compressed file for the session 1. On the biokdd server, do as following.
   A. Go to your 'L519FALL2005' directory.
   B. >tar –zcvf YourNetworkID.tgz ./HW1 (Suppose HW1 is your subdirectory)
4. **Please ENJOY learning and practicing new things.**

WARNINGS: **YOU ARE SUPPOSED TO WORK IN GROUP FOR THE MINI CLASS PROJECT. HOWEVER, YOU MUST DO HOMEWORK SESSION 1 AND 2 ON YOUR OWN.**

---Section 1 ---

For section 1, you are required to write Perl scripts to do the following tasks.

- Note: Sequence file should be in **FASTA** format. Please refer to the following site for further information on FASTA format; (Reference 1, Reference 2)

1. **Write a Perl script which can find complementary sequence**.
   - A. Introduction to DNA double strands and complementary sequence
     - i. DNA has a double helix structure. Each base forms hydrogen bonds with one directly opposite it, forming base pairs (Watson-crick pair).
     - ii. 5’ AGCTAGCT 3’ – Watson strand
       3’ TCGATCGA 5’ – Crick strand
     - iii. The rules of base paring are
       1. **A** with **T**: the purine **adenine** (A) always pairs with the pyrimidine **thymine** (T)
       2. **C** with **G**: the pyrimidine **cytosine** (C) always pairs with the purine **guanine** (G)

   - B. Script input: A DNA sequence file in FASTA format.
   - C. Script output: **Two** files
     - i. A complementary sequence file in FASTA file.
     - ii. A file containing both original & complementary sequence. Align these two sequence properly..
   - D. Restrictions
     - i. Limit number of characters to 60 per sequence line
     - ii. Your Perl script should be able to accept **input filename** from command line either as option or argument
       1. Use Getopt::Long for option http://perldoc.perl.org/Getopt/Long.html
       2. Or use @ARGV array
     - iii. Output file names should include the input filename.
     - iv. Proper error message should be displayed.

E. **JUST FOR FUN**.

i. Pick up any human gene of your interest from NCBI GenBank.

ii. Prepare the following four sequences

1. Original (5’ -> 3’)	AAGGCCGGTTTT
2. Reverse (3’ -> 5’)	TTTTGGCCGGAA
3. Complementary (3’ -> 5’)	TTCCGGCCAAAA
4. Reverse complementary (5’ -> 3’)	AAAACCGGCCTT

iii. Go to NCBI BLAST web page and run BLAST (blastn) against ‘nr’ database.

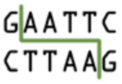
iv. Do you get the same or different matching results for all sequences?

2. Write a Perl script that can find occurrences of restriction enzyme recognition (cleavage) sites.

   A. Introduction to Restriction Enzyme

      i. A restriction enzyme (or restriction endonuclease) is an enzyme that cuts double-stranded DNA. The enzyme makes two incisions, one through each of the phosphate backbones of the double helix without damaging the bases.

      ii. For example, EcoRI restriction enzymes recognize the following sequence and cut them into two pieces.



      iii. DNA restriction enzymes recognize usually palindromic or partially palindromic sequences. There are also non-palindromic sequence recognizing RE such as Fok1 which recognize

```
GGATG
CCTAC
```

      iv. For more information, refer to http://en.wikipedia.org/wiki/Restriction_enzyme and http://www.promega.com/guides/re_guide/chapone/1_2.htm

   B. There is a database for restriction enzyme, which is called REBASE. Please go to the REBASE site and download the data file in any format you like.

      i. http://rebase.neb.com/rebase/rebase.html

   C. Write a Perl Script to find occurrence of a specific restriction enzyme site.

      i. Input

         1. REBASE raw data file.
         2. Genomic sequence file in FASTA same as in the script 1.
         3. Specific name(s) of Restriction enzyme(s)

      ii. Output

         1. A result file which shows the number of occurrences and positions of the given enzyme recognition site.

      iii. README file.

   D. Reference

      i. You may need to use regular expression. PERL Regular Expression

   E. Advice: Don’t forget that DNA is actually double stranded.

---------------------------------- Mini Group Project # 1 --------------------------------------

Mini group project #1 is sequential to the HW Section 1.

- GOAL
  - Create a web page in which users can search RE sites of their interests against user's own nucleotide sequence by using CGI.

- Users should be able to do the followings
  1. Nucleotide sequence
     A. Paste their nucleotide sequence into a text box
     B. Or select a file to upload
  2. Restriction enzyme
     A. Select a RE from a pull-down selection menu.
     B. Or directly input RE's name(s) or recognition sequence(s).
        i. You should provide instruction of what format users should use.

- Result page

1. Error message if any.
2. RE name, recognition sequence, number of occurrences, and positions

- References
  - L519 Lab 2 for simple CGI scripts
  - Any Perl CGI book
  - Google
- Restriction Summary Site

---Section 2 ---

For section 2, you are NOT required to write scripts. Simple pseudocode and its description would suffice.

1. Read the following introduction, and give an algorithm to solve the following '*Equivalent Words Problem*'

> In 1879, Lewis Carroll proposed the following puzzle to the readers of *Vanity Fair,* transform one English word into another by going through a series of intermediate English words, where each word in the sequence differs from the next by only one substitution. To transform *head* into *tail* one can use four intermediates:
>
> *head —> heal —> teal — tell —> tall —> tail.*
>
> We say that two words v and w are **equivalent** if v can be transformed into w by substituting individual letters in such a way that all intermediate words are English words present in an English dictionary.

Equivalent Words Problem:

Given two words and a dictionary, find out whether the words are equivalent.

Input: The dictionary, *V* (a set of words), and two words v and w from the dictionary.

Output: A transformation of **v** into w by substitutions such that all intermediate words belong to *V.* If no transformation is possible, output "v and w are not equivalent."

---

2. Give an algorithm which computes the optimal overlap alignment, and runs in time *O(nm)*.

Suppose that we have sequences v = $v_1 \ldots v_n$ and w = $w_1 \ldots w_m$, where v is longer than w. We wish to find a substring of v which best matches *all* of w. Global alignment won't work because it would try to align all of v. Local alignment won't work because it may not align all of w. Therefore this is a distinct problem which we call the *Fitting problem. Fitting* a sequence w into a sequence v is a problem of finding a substring v' of v that maximizes the score of alignments (v', w) among all substrings of v. For example, if v = GTAGGCTTAAGGTTA and w = TAGATA, the best alignments might be

	global	local	fitting
v w	`GTAGGCTTAAGGTTA` `-TAG----A---T-A`	`TAG` `TAG`	`TAGGCTTA` `TAGA--TA`
score	-3	3	2

The scores are computed as 1 for match, -1 for mismatch or indel (insertion/deletion or gap). Note that the optimal local alignment is not a valid fitting alignment. On the other hand, the optimal global alignment CONTAINS a valid fitting alignment, but it achieves a suboptimal score among all fitting alignments.