Comparative Genomics

Finding the most-frequently occurring motifs

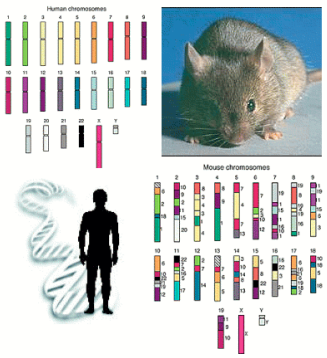
Write a Perl program to generate a report that serves as a preliminary study to compare and contrast certain features of DNA coding sequence between multiple organisms.

From: http://www.ornl.gov/sci/techresources/Human_Genome/faq/compgen.shtml

What is functional genomics?

Understanding the function of genes and other parts of the genome is known as functional genomics. The Human Genome Project (HGP) was just the first step in understanding humans at the molecular level. Though the project is complete, many questions still remain unanswered, including the function of most of the estimated 30,000 human genes. For just one example, researchers also don't know the role (if any) of single nucleotide polymorphisms (SNPs) --single DNA base changes within the genome-- or the role of non-coding regions and repeats in the genome.

What is comparative genomics? How does it relate to functional genomics?



Comparative genomics is the analysis and comparison of genomes from different species. The purpose is to gain a better understanding of how species have evolved and to determine the function of genes and non-coding regions of the genome. Researchers have learned a great deal about the function of human genes by examining their counterparts in simpler model organisms such as the mouse. Genome researchers look at many different features when comparing genomes: sequence similarity, gene location, the length and number of coding regions (called exons) within genes, the amount of non-coding DNA in each genome, and highly conserved regions maintained in organisms as simple as bacteria and as complex as humans.

Comparative genomics relies upon the use of computer programs that can line up multiple genomes and look for regions of similarity among them. One of the most widely used is BLAST, which is available from the National Center for Biotechnology Information. BLAST is a set of programs designed to perform similarity searches on all available sequence data.

Why is model organism research important? Why do we care what diseases mice get?

Functional genomics research is conducted using model organisms such as mice. Model organisms offer a cost-effective way to follow the inheritance of genes (that are very similar to human genes) through many generations in a relatively short time. Some model organisms studied in conjunction with the HGP were the bacterium *Escherichia coli*, yeast *Saccharomyces cerevisiae*, roundworm *Caenorhabditis elegans*, fruit fly *Drosophila melanogaster*, and laboratory mouse.

Starter Kit: You can download a starting Perl file to help you get started:
http://cs.wheatoncollege.edu/mleblanc/dna *(click on the **Ch. 9 Arrays and Hashes** link)*

INPUT: Using NCBI, collect at least five (5) FASTA format files of gene (coding) sequences. These files should be stored in a directory (folder) called **`inputGeneDirectory`**. This directory should be in the same directory (folder) as your Perl program. We are assuming the files are in FASTA format such that each file has an initial header line followed by lines of DNA sequence. Collect files from multiple organisms. ***Note**: We strongly recommend that you initially create some very, very small test files. You can use an editor (e.g., NotePad, WordPad, text-only files in Word) to generate test files by hand but be sure to save them as TEXT ONLY in the directory* `inputGeneDirectory`.

OUTPUT:

To the CONSOLE: Print the following (only) to the console window (not Excel file):

(1) A title of your report, including a "nickname" for your program
(2) A prompt for the user to enter the size of L-mer to use in the analysis between $MIN_SIZE(4) and $MAX_SIZE(8) bp. If the user enters an invalid number, you should print a warning message and ask for another input until you get an input in the acceptable range: 4 <= *input* <= 8.
(3) Print the motif size that was accepted.
(4) The filename of each file opened from your `inputGeneDirectory`
(5) The length of DNA sequence of each file in base pairs (bp)

In an EXCEL file (named: `output.xls`):

For each of the input files from `inputGeneDirectory`

(1) The filename
(2) The total number of motifs in the file (not the same as total bp; why?)
(3) TAB-delimited headings: Rank Motif Frequency Proportion
(4) TAB-delimited top-10 rankings of the motifs with the highest frequencies of occurrences appearing first

The output file (`output.xls`) will contain many tables, one for each input file. A sample table is shown below for *one* of the input files assuming L-mer = 4; this output is shown as viewed inside Excel. If you output TABs between your values, Excel will automatically make your output look nice.

FILE: humanXM_000000.fna			
Number of motifs: 930			
Rank	**Motif**	**Raw Count**	**Proportion**
1	CTCC	15	0.016
2	TCCT	15	0.016
3	CATG	14	0.015
4	TCCA	14	0.015
5	TTCC	13	0.014
6	CCAC	12	0.013
7	CCAT	12	0.013
8	CCTC	12	0.013
9	CTTC	12	0.013
10	ATCT	11	0.012

Requirements:

(1) Your program must contain proper documentation as discussed in class and requested on previous programs, including, your name, date last modified history, the status of the program when submitted, summary, input, and output.
(2) You must write a *good* comment for each variable that you use.
(3) Variables *must* have appropriate names that indicate to the reader how they are being used in the program, e.g., "`$DNAsequenceLength`", not "`$len`".
(4) You must include documentation (# comments) within your program.
(5) You must liberally use blank lines and dividing lines and good indentation
(6) You must use CONSTANTS for the minimum (4 bp) and maximum (8 bp) allowable motifs lengths and use the constant names in your program (e.g., 4 and 8 should not show up in your program after the initial declarations at the top of your program).
(7) You must PROMPT the user to enter the L-mer size of motif to use in your analysis, e.g.,

```
++++++++++++++++++++++++++++++++++++++++
          Comparative Genomics
            comGEN v1.0
++++++++++++++++++++++++++++++++++++++++

`-:-.   ,-;"`-:-.   ,-;"`-:-.   ,-;"`-:-.   ,-;"
   `=`,'=/     `=`,'=/     `=`,'=/     `=`,'=/
     y==/        y==/        y==/        y==/
   ,=,-<=`.    ,=,-<=`.    ,=,-<=`.    ,=,-<=`.
,-'-'   `-=_,-'-'   `-=_,-'-'   `-=_,-'-'   `-=_

What size motif do you want to use (enter an integer between 4 and 8): 3
WARNING:  3
 is an invalid length; try again.


What size motif do you want to use (enter an integer between 4 and 8): 6

Thanks.

Lmer = 6
```

(8) You must write a subroutine to print a title (to the console).
(9) You must finish the subroutines `readInDNA` and `breakIntoMotifs`
(10) You must use a CONSTANT for the number of words to include in the top-rankings. For example, if this constant is set to 10, your output will print the top-10 motifs.
(11) If a file contains fewer motifs than the top-N constant (see #10 above), your program should handle the situation and only print a shorter list of top ranked motifs.

(12) Your console output and Excel spreadsheet output ***must be*** professional and neat.
(13) You must submit a STAPLED hardcopy (printed in landscaped mode) of your source code.

(14) This program is due: **Mon, Nov. 12**.