ECS 124A Theory and Practice of Bioinformatics

Lab Assignment 3

Instructor: Ilias Tagkopoulos (iliast@ucdavis.edu)
TA: Jiying Li (jiyli@ucdavis.edu)
Due by: Thursday 11/10/2016 before class.

Scope:
The goal of this lab is for you to (a) learn the principles behind BLAST and its variations, (b) learn how to use BLAST, (c) learn more about programming techniques and implement them in PERL, (d) make your own, simplified version of BLAST, that builds on the same principles, (e) perform a sequence alignment for RNA-Seq samples by using various tools (BONUS EXERCISE!)

Deliverables:
Hand in answers to all the exercises/questions. For exercise 3 hand in your code for each step and printouts/files of the results wherever it is applicable.

Grading:
Exercise 1: 20
Exercise 2: 30
Exercise 3: 100
Exercise 4: 40 (BONUS!)
TOTAL : 150 pts (190 with bonus)

Note: Bonus points will be awarded towards lost points in ANY exercise and in any homework set. Exercise 4 is optional.

PART I: PERL and BLAST

Exercise 1. Regular expressions

Continue reading the second part of the Perl notes, which have been posted in Smartsite. Complete exercises 2.12, 2.13, 2.14 of those notes. In your lab report please reference then with the index one in front, i.e. exercise 2.12 will be 1.2.12, exercise 2.13 will be 1.2.13 and so forth.

Exercise 2. BLAST

In this part of the lab you will familiarize with BLAST. BLAST's main page is:

http://blast.ncbi.nlm.nih.gov/Blast.cgi
From there you can you to the BLAST handbook through help->NCBI Handbook:BLAST or just following this link:
http://www.ncbi.nlm.nih.gov/bookshelf/br.fcgi?book=handbook HYPERLINK "http://www.ncbi.nlm.nih.gov/bookshelf/br.fcgi?book=handbook&part=ch16"& HYPERLINK "http://www.ncbi.nlm.nih.gov/bookshelf/br.fcgi?book=handbook&part=ch16"part=ch16
Read some details about BLAST and answer (a paragraph for each) the following questions:

Question 2.1. What scores and statistics BLAST use? What does each one means and how is it calculated?

Question 2.2. What is the difference between BLAST, BLAST 2, PSI-BLAST, FASTA? When should each of these been used (trade-offs)?

Question 2.3. Read:
http://www.ncbi.nlm.nih.gov/BLAST/tutorial/Altschul-1.html
And explain what is the p-value for the top match of the following query (that comes from the movie Jurassic Park) :

```
>DinoDNA from JURASSIC PARK  p. 103 nt 1-1200
GCGTTGCTGGCGTTTTTCCATAGGCTCCGCCCCCCTGACGAGCATCACAA
AAATCGACGC
GGTGGCGAAACCCGACAGGACTATAAAGATACCAGGCGTTTCCCCCTGG
AAGCTCCCTCG
TGTTCCGACCCTGCCGCTTACCGGATACCTGTCCGCCTTTCTCCCTTCGGG
AAGCGTGGC
TGCTCACGCTGTACCTATCTCAGTTCGGTGTAGGTCGTTCGCTCCAAGCTG
GGCTGTGTG
CCGTTCAGCCCGACCGCTGCGCCTTATCCGGTAACTATCGTCTTGAGTCC
AACCCGGTAA
AGTAGGACAGGTGCCGGCAGCGCTCTGGGTCATTTTCGGCGAGGACCGC
TTTCGCTGGAG
ATCGGCCTGTCGCTTGCGGTATTCGGAATCTTGCACGCCCTCGCTCAAGC
CTTCGTCACT
CCAAACGTTTCGGCGAGAAGCAGGCCATTATCGCCGGCATGGCGGCCGA
CGCGCTGGGCT
GGCGTTCGCGACGCGAGGCTGGATGGCCTTCCCCATTATGATTCTTCTCGC
TTCCGGCGG
CCCGCGTTGCAGGCCATGCTGTCCAGGCAGGTAGATGACGACCATCAGG
GACAGCTTCAA
```

```
CGGCTCTTACCAGCCTAACTTCGATCACTGGACCGCTGATCGTCACGGCG
ATTTATGCCG
CACATGGACGCGTTGCTGGCGTTTTTCCATAGGCTCCGCCCCCCTGACGA
GCATCACAAA
CAAGTCAGAGGTGGCGAAACCCGACAGGACTATAAAGATACCAGGCGTT
TCCCCCTGGAA
GCGCTCTCCTGTTCCGACCCTGCCGCTTACCGGATACCTGTCCGCCTTTCT
CCCTTCGGG
CTTTCTCAATGCTCACGCTGTAGGTATCTCAGTTCGGTGTAGGTCGTTCGC
TCCAAGCTG
ACGAACCCCCCGTTCAGCCCGACCGCTGCGCCTTATCCGGTAACTATCGT
CTTGAGTCCA
ACACGACTTAACGGGTTGGCATGGATTGTAGGCGCCGCCCTATACCTTGT
CTGCCTCCCC
GCGGTGCATGGAGCCGGGCCACCTCGACCTGAATGGAAGCCGGCGGCAC
CTCGCTAACGG
CCAAGAATTGGAGCCAATCAATTCTTGCGGAGAACTGTGAATGCGCAAA
CCAACCCTTGG
CCATCGCGTCCGCCATCTCCAGCAGCCGCACGCGGCGCATCTCGGGCAG
CGTTGGGTCCT
```

Go through the calculations and see if the p-value matches the one reported by BLAST. If you have to make any assumptions please state them explicitly.

<u>Exercise 3. Create PERL-BLAST</u>

In this exercise we will create a toy blast program that we will call PERL-BLAST. It will use the same principles as BLAST does, for finding seed words first (k-meres) and then extending them to find potential alignments. First, please downloadthe four k-mere programs from Smartsite, under *Resources->Lab3, file kmer_v123_and_kmerfirst.txt*. Start from ***kmer1.pl*** and work your way to ***kmerfirst.pl***, executing each code and understanding how it works. The new elements of Perl that you will use include the **substr** function, the **length** function, data structures such as two-dimensional **hashes and lists**, the **defined** function. For better understanding, I encourage you to read any Perl book (like Johnson's) or online resource regarding these functions and data structures.

The program ***kmerfirst.pl*** finds the first position of each of the different kmers of length k. This program will be the starting point for your PERL-BLAST program. Your program should do the following things:

- Read in from a file a query string Q.
- For k = 4, use program ***kmerfirst.pl*** to find the first location of each different k-mer in Q.
- Successively read in one string at a time from a file called ***perlblastdata.txt*** that is located again under the *Resources>>Lab 3.*When a string S is read in, scan through its 4-mers, using the same hash as before. For this, extract and adapt what you need from ***kmerfirst.pl****.*
- Whenever a 4-mer in S is determined to be in Q, extract the location of the first occurrence of that 4-mer in Q. Then put the characters of Q and S in arrays (as we did in ***needleman.pl***) so that you can examine individual characters. Then scan left from the k-mer in Q and in S, as long as you find matching characters. Repeat to the right. Let L denote the length of the whole match obtained in this way. If L is greater than 10, then print a message that a good HSP has been found between Q and S, and print S.Notice that the same HSP gets reported multiple times. *Explain why that happens*.
- Now we will alter the code so that HSP are not reported multiple times. We can do it using a hash called **stringhash**: Whenever PERL-BLAST finds a reportable substring in a database, starting at position $i (e.g. in the database string), it searches whether $stringhash{$i} is defined. If it is, it does not report the string again. Otherwise it assigns the string to $stringhash{$i} and reports the string.
- We would like to process strings that are more than a single line long. So in the file each string will be held in consecutive lines, with strings separated by blank lines. That is analogous to each string being a paragraph instead of just a single line. To read in a paragraph, put the line *$/ = "";*somewhere in the program before the string is read. What does this line do ?
- Finally, we will make it so that if a k-merthat is present in the database string is also in the query string in multiple locations, then a search should be made from each occurrence of the k-mer in the query string, spanning outward left and right of each occurrence. To do that use **kmer4.pl**that is found under the *Resources>Lab 3* (file *kmer4.txt*) to build up a list of all occurrences of each distinct k-mer in the query string, and use it to implement this change.

Congratulations, you made your own version of BLAST! Regarding what to hand-in, for each step, provide the changes in the code that you implemented and a test case, where you show the desirable result. As a test case, use the file *perlblastdata.txt*. Your program should ask for a *threshold t*, and report each string in the database that has one or more substrings matching any substring in the querythat has a length at least *t*. Also report the actual length of the longest matching substring. Report the results for k=3 and k=4 when threshold t=6 and t=7 (4 combinations total).

Exercise 4. RNA-Seq mapping

For this exercise, you will use a software suite that is called GALAXY to analyze five RNA-Seq datasets. The datasets are paired-end 50bp reads from adrenal and brain tissues (500Kb region of chromosome 19, chr19:3000000:3500000). You can find the datasets here:

https://usegalaxy.org/u/jeremy/p/galaxy-rna-seq-analysis-exercise

The datasets contain the transcriptional profiles in those tissues and our task is to find genes that are differentially expressed (DEG) in the sample conditions versus the control sample. We have 3 programs to perform this task: edgeR, Cuffdiff and DESeq2. You are asked to do this analysis with each and finally report the results.

4.1. (10 points) Using Cuffdiff from Galaxy, complete the tutorial that is found in the previous link and report the top 100 DEG, their fold-change (or score) and their p-values.

4.2. (10 points) Perform the same analysis by using EdgeR, which can be found here:
http://www.bioconductor.org/packages/release/bioc/html/edgeR.html

Again report the top 100 DEG based on EdgeR, their fold-change (or score) and their p-values.

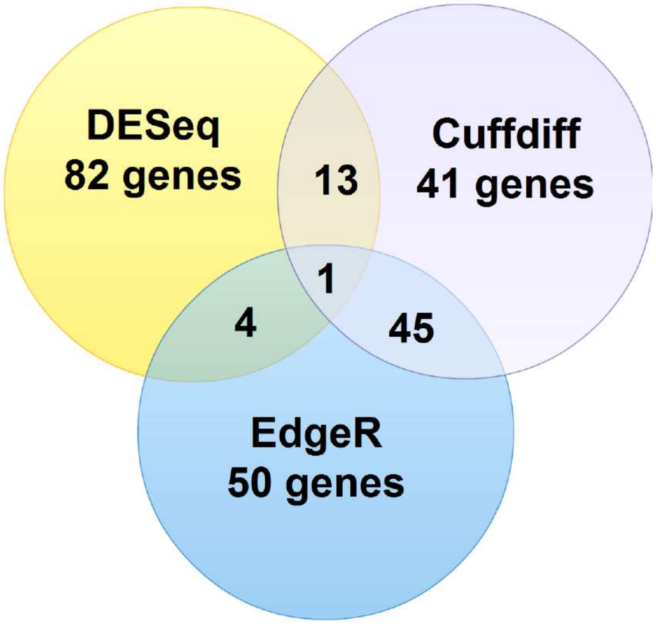
4.3.(10 points) Finally, perform the same analysis by running DESeq2:
http://bioconductor.org/packages/release/bioc/html/DESeq2.html

And (as you have guessed) report the top 100 DEG, their fold-change (or score) and their p-values. For EdgeR and DESeq, you can use HTSeq to process the alignment results (.bam files from TopHat on Galaxy):

http://www-huber.embl.de/users/anders/HTSeq/doc/overview.html

before you pass the results into EdgeR and DESeq.

4. 3. (10 points) Compare the results of the 100 DEG as generated by the three tools above. Create a single xls file that contains these genes (with their scores and p-values) and create a Venn Diagram with three sets (i.e. circles that overlap) to report the overlap of DEG in each case (as shown in the figure below).What do you observe? Why do they produce different results? What is the basis of each method and where can each be used?



NOTE: This exercise is not for the faint of heart! Remember this is a **bonus exercise** that you should only try after you complete the other 3 exercises and be ready to do a lot of trouble-shooting. Consult with the TA early on if you have any issues.

END OF LAB 3