Algorithmic Bioinformatics (DD2450)

Assignment 1

1 Grades

There will be three batches of assignments, of which this is the first batch. Each batch will contain an implementation and a problem solving part. Each part of each batch will, basically, be scored as good, pass, or fail. You will need 5 good to get an A; 4 to get a B; 3 to get a C; 2 to get a D; and 1.5 to get an E. Several instances of pass will be converted to an instance of good in a rather ungenerous fashion.

2 Implementation

You have a part of mouse genome as the reference sequence and a number of short cDNA reads in a separate file. Both files are in *fasta format*. If you are not familiar with the fasta format, please consult

http://www.ncbi.nlm.nih.gov/blast/fasta.shtml

before you start processing the files.

Each student will receive his/her own dataset via email. Also you can download your dataset from the course homepage. Please click on the link with the first two characters in your email address.

2.1 Local alignment

Implement the DP local alignment algorithm from the lecuters. We define a hit as an incident of alignment of a read against the reference sequence when the number of mismatches is below a certain threshold. Please note that a single read can have various hits with the reference sequence and you need to count all of them in your report. In this assignment, use 5 as a threshold.

In both tasks 2.1 and 2.2 use the scores in Table 1 in your implementation.

2.2 Local-global alignment

Describe a local-global alignment algorithm with the following input and output:

Case	Score
match	+1
mismatch	-1
opening a gap	-0.5
extending a gap	-0.25

Table 1: Scores

Input: Two sequences X and Y.

Output: Optimal local-global alignment score of X and Y, i.e., the maximum of two $\gamma(X', Y)$ where $X' \subseteq X$.

Now implement an algorithm as effiecient as possible for this problem.

2.3 Implementation results

For both tasks 2.1 and 2.2 submit the following items separately:

Mismatch type	Statistics
A-G	
C-T	
G-A	
A-T	

Table 2: Sample result table

- A brief description of your methodology and what you did.
- Statistics of different mismatch types which your program detected. Please use Table 2 as a template. You need to submit your results for all 12 possible mismatch types.
- A comma separated text file which each row is (type of mismatch , position in the reference sequence). For example if you have a mismatch of type A-G in position 1245 in the reference sequence the corresponding entry in your file should look like:

  ```
  AG,1245
  ```

  The first position in the reference sequence is +1.
- Submit a reasonably commented version of your source code via email. You can use any programming language. Your programs must read the input files from the same directory as the source code. This makes testing of your codes much easier for us. Please notify us if you used a special version of a compiler/interpreter.

i	a_i	b_i
1		
2		
...		
...		
21		
22		

Table 3: Sample result table

After implementing both algorithms 2.1 and 2.2 answer the following:

- Statistics of mismatches in different positions in the reads: Consider a_i and b_i as the sum of the reads which have a mismatch at position i detected by local and local-global alignmet methods respectively . Assume the $5'$ side of each read as position $+1$. Use Table 3 as a template to submit your results.
- What is the main difference between a_i and b_i? Explain the reason for this difference.

3 Problems

3.1 Pairwise global aligment problems

For a sequence X, let $|X|$ be the length of X.

1. Describe an algorithm that, for two given string X and Y, in time $O(|X||Y|)$ computes the number of optimal global alignments of X and Y.
2. Describe an algorithm that, for two given string X and Y, in time $O(|X||Y|)$ picks an optimal global alignments of X and Y according to the uniform distribution over those.

3.2 Aligning against a trie

A trie is a data structure that can be used to store a number of sequences. The trie is actually a rooted tree where each edge is labeled with a symbol and each root to leaf path in the natural way describes a sequence (i.e., the sequence of symbols appearing on the edges of the path).
Describe an algorithm that given a DNA sequence X and a trie t, finds the sequence stored in t that is most similar to X. Most similar should be interpreted the same ways as it has been for global pairwise aligning during the lectures (you have a symbol similar matrix etcetera). The algorithm should run in time $O(|t||X|)$ where $|t|$ is the number of edges in the trie and $|X|$ is the length of X.