6.096 Algorithms for Computational Biology

Professor: Manolis Kellis *Problem Set 1*

Instructions

This problem set is due in lecture on Friday March 4th, 2005. For problems with a programming component, the algorithms should be implemented using python. If you're not familiar with python, there is a good tutorial at python.org. When handing in your solutions, you should turn in your python code. Collaboration is permitted, but the write-up and the code should be your own. Please acknowledge your collaborators.

Suggested readings

- CLRS appendix C.5 The tails of the binomial distribution (the probability of obtaining at least k successes out of n trials.
- Jones and Pevzner's book has three chapters on Motif discovery: 4.6, 5.5, 12.2 and 12.3. The book is on reserve at the library.

Motif discovery

You will discover over-represented short strings of length k (motifs) in intergenic sequences from Saccharomyces Cerevicie (baker's yeast). You will explore different methods presented in class. On the course website, you will find two data files. The first contains the S.c. intergenic sequence separated by #. The second file contains information about the conservation of the sequence across several yeast species. A * means conserved, and a blank, not conserved. Note that the sequence information for the other species is not necessary in our search provided. You will discover motifs using the following methods.

1 Exhaustive search

Create a table listing all the strings of length 6 from the alphabet $DNA = A, C, G, T$. Scan the intergenic sequences to obtain a count for each the motifs in the table. List the top 100 motifs in order of decreasing appearance.

1.1 Analysis

Let the genome length be denoted by m and the length of the motif be k, what is the running time of the exhaustive search?

1.2 Conservation

Do additional data sources help in motif discovery? You have a file containing information about the conservation of bases across different yeast species. There is a $*$ if a base is conserved and a blank if it is not conserved. The base positions exactly agree with the bases file that you worked with in the Exaustive implementation. Perform an exhaustive search of all motifs of length 6, increasing the count only if all the bases in the motifs are also conserved,i.e. if there are 6 consecutive stars. How do your results compare to the previous results? Use this to derive a conservation rate of the motifs i.e. the frequency of conservation.

2 Content-based addressing

In the exhaustive search problem, the genome had to be scanned 4^k times in order to produce a count of the different motifs. Devise an algorithm in which you get an exact count of each motif by scanning the genome only once (despite the size of the motif). What data structure did you use? Provide an implementation of this algorithm.

3 Basic Sampling

In order to avoid scanning the whole genome, we can estimate the most common motifs by sampling. Random sampling works as follows: Randomly pick a genome position and record the six-mer that appears starting at the random position.

3.1 Sampling probabilities

Let α denote the fraction of occurence of a motif in the intergenic regions. If you see m samples, then what is the expected number of occurences of the motif in the sample? Also given α, what is the probability of not observing this motif in the m samples? Note that each sample taken can be seen as an independent Bernoulli trial, with probability α of success. (HINT: this should be a straight forward calculation)

3.2 Chernoff Bounds

Let $\hat{\alpha}$ be the empirical probability of obtaining a given motif, that is, the number of times that the motif appears in the sample divided by the number of samples. If you take m independent samples, then the following Chernoff Bounds hold:

$$\begin{aligned} Pr[\hat{\alpha} \leq \alpha(1-\epsilon)] &\leq e^{-m\alpha\epsilon^2/2} \\ Pr[\hat{\alpha} \geq \alpha(1+\epsilon)] &\leq e^{-m\alpha\epsilon^2/3} \end{aligned}$$

(If you're interested in these general bounds, a good advanced book is Alon and Spencer's *The Probabilistic Method*).

3.3 α

From your experiments in the exhaustive-search problem, what is the fraction of occurences of the most frequent motif, call itα?

3.4 Sample probability upper bound

If you sample the intergenic sequence randomly at 1000 positions, what is the probability of obtaining $\hat{\alpha} \leq .9\alpha$? what about 10000 times?

3.5 Sample probability lower bound

If you sample the intergenic genomic sequence 1000 times, what is the probability of obtaining $\hat{\alpha} \geq 1.2\alpha$? what about 10000 times?

4 Random Projections

In the second lecture we introduced the idea of random projections. This is another probabilistic tool useful in motif finding and in many other contexts. If the query sequence is of length d, then we use k random positions out of the d possible positions. Therefore the search space is reduced and this technique reduces the dimensionality of the problem. Random projections allows for two sequences to match if they are similar, but not necessarily equal. Therefore, motifs discovery is possible, even with the presence of mutations. Here we explore the probabilistic properties of a simple version of random projections.

4.1 Probability of a perfect match

Given two sequences v and w of the same length, let the *Hamming distance* between them, $d_H(v, w)$ be the number of positions in which they differ. This is a very useful notion of distance for sequences. Let the Hamming distance between two strings of length d is zero, then if we take k random positions (with replacement), what is the probability that the two strings agree at these position? What does this imply about the sensitivity of the approach? What is the probability that the two strings agree at these position if the Hamming distance is one?

4.2 Probability of a match given the Hamming distance

In general, if the Hamming distance between two sequences v, w is $d_H(v, w)$, what is the probability that k random positions agree?

4.3 Collision plots

A collision is an event in which two sequences have the same random projection. Plot the probability of collision of two sequences against their Hamming distance for $d = 10$ and $k = 3$. How does it change as k increases i.e what if $k = 1$ What if $k = d$?

4.4 Repeated projections

You can extend the previous ideas to several random projections. Let l be the number of random projections. Calculate the probability of having at least one equal projection. Plot this probability against the hamming distance between two sequences. Try $d = 10$, $k = 3$, and $l = 5$.