9 Operational Research

9.3 Protein Comparison in Bioinformatics (8 units)

This project is computationally intensive but mostly self-contained mathematically. Some understanding of random variables (covered in the Part IA Probability course) is required.

Introduction

Sequence comparison and alignment, combined with the systematic collection and search of databases containing biomolecular sequences, both DNA and protein, has become an essential part of modern molecular biology. Molecular sequence data is important because two proteins with similar sequences often have similar functions or structures. This means that we can learn about the function of a protein in humans by studying the functions of proteins with similar sequences in simpler organisms such as yeast, fruit flies, or frogs.

In this project we will examine methods for comparison of two sequences.

We will work with two strings S and T of lengths m and n respectively, composed of characters from some finite alphabet. Write S_i for the ith character of S, and $S[i,j]$ for the substring $S_i, \ldots, S_j$. If $i > j$ then $S[i,j]$ is the empty string. A *prefix* of S is a substring $S[1,k]$ for some $k \leqslant m$ (possibly the empty prefix). Similarly, a *suffix* of S is a substring $S[k,m]$ with $k \geqslant 1$.

1 The edit distance

This section follows work originally done by Needleman and Wunsch [4], although the notation we use is slightly different.

Suppose $S =$ `fruit` and $T =$ `berry`. We can transform S into T by

1. **R**eplacing f with b,
2. **I**nserting e,
3. **M**atching r with r,
4. **R**eplacing u with r,
5. **R**eplacing i with y,
6. **D**eleting t.

We call `RIMRRD` the *editing transcript*. The *alignment* of S and T is read vertically character by character and is given by:

```
RIMRRD
f ruit
berry
```

There are, of course, many possible ways to transform one string into another; but from an evolutionary point of view there must be a cost associated with any action other than matching. The *optimal* edit transcripts are those that involve the least number of edit operations (**R**eplace, **I**nsert, and **D**elete). We define the *edit distance* $d(S,T)$ to be the minimal number of edits between S and T. Let $D(i,j) = d(S[1,i], T[1,j])$. Observe that $d(S,T) = D(m,n)$.

Question 1 Prove that for all $i, j > 0$

$$D(i,j) = \min\{D(i-1,j)+1, D(i,j-1)+1, D(i-1,j-1)+s(S_i,T_j)\},$$

where $s(a,b)$ is some suitable function which you should determine. Explain your reasoning carefully. What boundary conditions $D(0,0)$, $D(i,0)$, and $D(0,j)$ did you use?

Question 2 Write a program to find the edit distance between two strings. Use it to find the edit distance between `shesells` and `seashells`. What is the complexity of your algorithm?

Usually, we are interested in finding an optimal editing transcript and alignment of S and T rather than just $d(S,T)$. This can be done by assigning a pointer when calculating $D(i,j)$, pointing to one of $D(i-1,j)$, $D(i,j-1)$, or $D(i-1,j-1)$.

Question 3 Modify your algorithm so as to produce one possible optimal alignment between two strings. Take proteins A and B from the file *proteins.txt* on the CATAM website. (Both proteins are myoglobin, protein A is for the duckbill platypus and protein B for yellowfin tuna.) Find the edit distance between them, and give the first 50 steps of an optimal alignment.

2 Scoring matrix

A protein is essentially a long sequence of amino acids. Approximately twenty types of amino acid (the exact number is species dependent) are involved in the construction of each protein. A gene is a sequence of DNA which can be translated into a sequence of amino acids, i.e., a protein. Mutations in DNA will lead to changes in the sequence of amino acids, and some mutations are more likely than others. In this section we adjust our scoring algorithm in order to capture some of these biological considerations.

The adjustment is achieved by replacing the scoring function $s(a,b)$ which you found in Section 1. There are various schemes for assessing the probability of a mutation from amino acid a to amino acid b; currently the two dominant schemes are the PAM matrices introduced by Dayhoff [2], and the BLOSUM matrices of Henikoff and Henikoff [3].

For historical reasons, we will talk about maximising a score rather than minimizing a distance. Let $v(S,T)$ be the maximum score of all edit transcripts from S to T.

Question 4 Using the BLOSUM matrix *blosum.txt* from the CATAM website for the scoring function s, and scoring -8 for each **I**nsert or **D**elete, find the score v between proteins A and B and give the first 50 steps of the optimal alignment.*

3 Scoring for gaps

Some mechanisms for DNA mutations involve the deletion or insertion of large chunks of DNA. Proteins are often composed of combinations of different domains from a relatively small repertoire; so two protein sequences might be relatively similar over several regions, but differ in other regions where one protein contains a certain domain but the other does not.

*If your version of MATLAB has the *Bioinformatics Toolbox* installed, the appropriate BLOSUM matrix can be generated using the command `blosum(62,'order','CSTPAGNDEQHRKMILVFYW')`.

At some computational cost, we can still align two protein strings taking gaps into account. Let $w(l) < 0$, $l \geqslant 1$, be the score of deleting (or inserting) a sequence of amino acids of length l from (or into) a protein. Let $v_{\text{gap}}(S, T)$ be the gap-weighted score between S and T, and write $V_{\text{gap}}(i, j)$ for $v_{\text{gap}}(S[1, i], T[1, j])$. Then

$$\begin{aligned} V_{\text{gap}}(i,j) &= \max\{E(i,j), F(i,j), G(i,j)\}, \\ E(i,j) &= \max_{0 \leqslant k \leqslant j-1} \{V_{\text{gap}}(i,k) + w(j-k)\}, \\ F(i,j) &= \max_{0 \leqslant k \leqslant i-1} \{V_{\text{gap}}(k,j) + w(i-k)\}, \\ G(i,j) &= V_{\text{gap}}(i-1, j-1) + s(S_i, T_j). \end{aligned}$$

Iterating the above equations on the n by m grid has complexity of $O(mn^2 + nm^2)$. Happily, if $w(l)$ takes some fixed value u for all $l \geqslant 1$, then there exists an algorithm for finding v_{gap} which has complexity $O(mn)$.

Question 5 Find and implement such an algorithm. Explain how your algorithm works, and why it has complexity $O(mn)$. What boundary conditions do you use?

Question 6 Take proteins C and D from the file *proteins.txt* on the CATAM website. (Both proteins are keratin structures in humans.) Using the BLOSUM matrix from Section 2 for the scoring function s, and $u = -12$ as the fixed score of insertion/deletion, find the gap-weighted score $v_{\text{gap}}(C, D)$ and give the first 50 steps of the optimal alignment.

4 Statistical significance

We may now ask at what threshold a score $v_{\text{gap}}(S, T)$ should be declared to have biological significance?

Let us simplify the problem slightly. Suppose there are only two letters in our alphabet, a and b, corresponding, say, to hydrophobic and hydrophilic amino acids. Let $s(a, a) = s(b, b) = 1$ and $s(a, b) = s(b, a) = -1$. Let U^n be a random protein of length n: all the amino acids $U_1^n, \ldots, U_n^n$ are independent and identically distributed, with $P(U_i^n = a) = p$ and $P(U_i^n = b) = 1 - p$.

Question 7 Consider two random proteins U^n and V^n, independent and identically distributed. Let the score of inserting/deleting a sequence of length l be fixed: $w(l) = u$ for all $l \geqslant 1$. Prove that for all $0 \leqslant p \leqslant 1$, and for all $u \leqslant 0$,

$$\liminf_{n \to \infty} \frac{\mathbb{E}(v_{\text{gap}}(U^n, V^n))}{n} > 0.$$

(Note: if x_n is a sequence of real numbers, then $\liminf_{n\to\infty} x_n$ is defined by

$$\liminf_{n \to \infty} x_n = \lim_{n \to \infty} \inf_{m > n} x_m.$$

Note that the limit is always guaranteed to exist, though it may be $+\infty$ or $-\infty$. This is discussed further in most textbooks on analysis.)

In fact, $\lim_{n\to\infty} n^{-1}\mathbb{E}(v_{\text{gap}}(U^n, V^n))$ exists and is strictly positive. One way to obtain an excellence mark in this project, though not the only way, is to show that this limit exists.

Question 8 Let $u = -3$, $p = \frac{1}{2}$. Write a program to estimate $n^{-1}\mathbb{E}(v_{\text{gap}}(U^n, V^n))$. Now vary n and estimate the limit of $n^{-1}\mathbb{E}(v_{\text{gap}}(U^n, V^n))$. Explain how you arrive at your estimate of the limit.

5 Local Alignment

Full alignment of proteins is meaningful when the two strings are members of the same family. For example, the full sequences of the oxygen-binding proteins myoglobin and haemoglobin are very similar. Often, though, only a small region of the protein is critical to its function and only this region will be conserved throughout the evolutionary process. When we identify two proteins which perform similar functions but look superficially different, it is useful to identify these highly conserved regions.

We aim to find a pair of substrings S' and T' of S and T with the highest alignment score, namely,

$$v_{\text{sub}}(S,T) = \max\{v(S',T') : S' \text{ a substring of } S,\ T' \text{ a substring of } T\}.$$

(For simplicity, we will use the same scoring as in Section 2. We will also write $s(-,a) = s(a,-) < 0$ for the score of an insertion or deletion.)

Finding $v_{\text{sub}}(S,T)$ seems to be of much higher complexity than solving the global alignment problem, as there are $\Theta(n^2m^2)$ combinations of substrings of S and T. Amazingly, we will solve it using an algorithm whose complexity is still only $O(mn)$.

We will first define a slightly easier problem. Suppose we restrict ourselves to suffixes of S and T:

$$v_{\text{sfx}}(S,T) = \max\{v(S',T') : S' \text{ a suffix of } S,\ T' \text{ a suffix of } T\}.$$

Question 9 Prove carefully that

$$v_{\text{sub}}(S,T) = \max\{v_{\text{sfx}}(S',T') : S' \text{ a prefix of } S,\ T' \text{ a prefix of } T\}.$$

Question 10 Write $V_{\text{sfx}}(i,j)$ for $v_{\text{sfx}}(S[1,i],T[1,j])$. Prove that

$$V_{\text{sfx}}(i,j) = \max\begin{cases} 0, \\ V_{\text{sfx}}(i-1,j-1) + s(S_i,T_j), \\ V_{\text{sfx}}(i-1,j) + s(S_i,-), \\ V_{\text{sfx}}(i,j-1) + s(-,T_j), \end{cases}$$

with boundary conditions $V_{\text{sfx}}(i,0) = V_{\text{sfx}}(0,j) = 0$.

Question 11 Find v_{sub} for proteins C and D, using the BLOSUM matrix from Section 2 and $s(a,-) = s(-,a) = -2$ for all amino acids a.

References

[1] Altschul, S. and Erickson, B.W. *Optimal sequence alignment using affine gap costs.* Bulletin of Mathematical Biology 48: 603-616, 1986

[2] Dayhoff, M.O., Schwartz, R.M. and Orcutt, B.C. *A model of evolutionary change in proteins.* Atlas of Protein Sequence and Structure 5: 345-352, 1978

[3] Henikoff, S. and Henikoff J.G. *Amino acid substitution matrics from protein blocks.* Proceeding of the National Academy of Science 89: 10915-10919, 1992

[4] Needleman, S.B. and Wunsch, C.D. *A general method applicable to the search for similarities in the amino acid sequence of two proteins.* Journal of Molecular Biology 48: 443-453, 1970

