18.417 Introduction to Computational Molecular Biology

Lecture 6: September 28, 2004 — Scribe: Lele Yu
Lecturer: Ross Lippert — Editor: Sam Kaufman

Dynamic Programming: Gene Discovery

Introduction

This lecture discusses some techniques for gene discovery using dynamic programming algorithms. We begin with some necessary biological background.

6.1 The Central Dogma of molecular biology

The central dogma of molecular biology is that DNA is transcribed to RNA which is translated to protein. In translation, three RNA base pairs code for an amino acid.

To find coding regions, we consider the null hypothesis of modeling the non-coding part of the genome as a Bernoulli sequence. Under this model, there's a $\frac{1}{64}$ probability for each codon. Deviations from random indicate potential coding regions.

Codon usage. We can see the probability of a codon for each amino acid. There are repeats for each tRNA coding to guard against damage. For prokaryotes, codon bias is indicated by tRNA frequency.

6.2 ORFs Open Reading Frames

In a random sequence, 3/64 of the time, we will see a stop codon. So, theres an average period of 20 codons before a stop codon. These regions between stops are ORFs. ORFs and statistical usage of codon gives us reasonable coding regions. The average length of a protein is 300 amino acids, and random ORFs of this size are rare. This method is good for prokaryotes because theres no splicing.

If there are too many ORFs of length 40 aa, analysis using statistics of codons is no longer useful because its much easier to randomly get a region of 40aa than 300aa.

6.3 Intron/exon structure

In eukaryotes, splicing (the presence of introns, i.e. noncoding regions interspersed in exons) can foul this up. Exons are 2% of the gene. Exons average length is 130 bp $\approx$ 43 aa.

The basic intron exon/structure is as in the diagram:

Exon1 GT ———— AG Exon2 GT ————- AG Exon3

i.e. there are upstream and downstream regions ("donor" - GT and "acceptor" - AG sites), two bases of which are highly conserved (there is more information in these regions, but less clear). These tags as well as ORF length and codon usage rates can be used to find potential exons in eukaryotes. Using everything, you'll find 2/3 of existing exons.

6.4 Exon Chaining

Given a set of candidate exons, we'd like to make a guess about a possible gene. This amounts to choosing a set of non-overlapping exons in order. We represent each candidate exon as a triple (l_i, r_i, s_i), where l is the left end of the exon, r is the right end of the exon, and s is a score indicating intrinsic goodness, e.g. from stats. We'd like to output a sequence of such exons with $r_i < l_{i+1}$, so they don't overlap, which maximizes $\sum s_i$. This can be done in O(n) time using dynamic programming.

6.5 Similarity based gene searching/structuring

Another way to find genes in DNA is to match to their known products, mRNA and proteins.

6.5.1 Matching known (spliced) mRNA

Suppose we have known (spliced) mRNA, corresponding (roughly) to the transcribed exons. There are two schools of thought on how to align such an mRNA to the full DNA sequence containing both introns and exons. One is to do a gapped alignment, with a strong mismatch penalty and low gap penalty, with a high gap start cost to reduce the risk of getting random letters from introns. The second is to do local alignments and collect any high scoring local alignments.

6.5.2 Amino Acid to DNA alignment

Sometimes we might not be lucky enough to have known mRNA and will have only the amino acid sequence. This introduces ambiguity because of codon redundancies. Methods to do this vary according to their moral rectitude – i.e. how principled they are.

A totally unprincipled method is to do translation on the DNA in all three frames and then to try to align the amino acid sequences to each other. The problem is that any gap in a nucleotide sequence alignment creates huge protein changes. (One way around this is to force indels and mismatches to come in triples.) Generally, this method is too pessimistic.

A principled method would align a hypothetical intermediate DNA sequence ("iDNA") against the input DNA sequence, and then align the translation of the iDNA to the given amino acid sequence, and optimize the sum of the scores of the two alignments.

A semi-principled method limits the inserts in DNA relative to iDNA.

6.6 Spliced alignment problem

The spliced alignment problem shows up in a setting where regions of spliced cDNA have been aligned to a target genome in pieces, producing a series of high scoring local alignments. The input is a set of boxes with scores, represent as 5-tuples $(l_i, r_i, t_i, b_i, s_i)$ where the entries stand for left, right, top, bottom, and score. The output is a non-overlapping subset of the boxes, which is order consistent and maximizes the sum of the scores. Dynamic programs (real movers and shakers) exist which can solve this problem in $O(n^2)$ to $O(nlogn)$ time.