

EST_GENOME: a program to align spliced DNA sequences to unspliced genomic DNA

Richard Mott

This note describes the program EST_GENOME for aligning spliced DNA to unspliced genomic DNA. It is written in ANSI C and has been tested under Digital OSF3.2. The source code and documentation are available from ftp://www.sanger.ac.uk/ftp/pub/badger/est_genome.2.tar.Z.

The prediction of genes in uncharacterized genomic DNA sequence is currently one of the main problems facing sequence annotators. Methods based on *de novo* prediction, e.g. searching for motifs like the splice-site consensus, or on statistical properties such as biased codon usage, etc. (Solovyev *et al.*, 1994; Hebsgaard *et al.*, 1996) have been only partially successful, and investigators have often found that the surest way of predicting a gene is by alignment with a homologous protein sequence (Birney *et al.*, 1996; Gelfand *et al.*, 1996; Huang and Zhang, 1996), or a spliced gene product [an expressed sequence tag (EST), mRNA or cDNA], particularly now that a large number of ESTs are available (Hillier *et al.*, 1996).

Standard alignment tools are not ideal for finding the correct alignment of a spliced product to genomic DNA, because of the large introns which can occur in the genomic sequence and because the programs ignore the conserved sequences found at donor/acceptor splice sites (intron/exon boundaries). In addition, very large genomic DNA sequences can be hard to align using quadratic-space dynamic programming because they require too much memory.

The program EST_GENOME addresses this problem. It allows large introns, can recognize splice sites and uses limited memory. This combination of features makes a powerful and useful tool. EST_GENOME is used routinely at the Sanger Centre to help annotate human genomic sequence. As it is slow compared with search methods like BLAST (Altschul *et al.*, 1990), we first screen genomic DNA against dbEST using BLASTN. Any matching ESTs are realigned using EST_GENOME.

The algorithm uses a modification of Smith and Waterman (1981). The penalty structure used to score an alignment is as follows (defaults are in parentheses). Aligned bases score +*match* (1) or cost –*mismatch* (1) as appropriate. An indel in

either sequence outside of an intron costs –*gap* (2) (there is no gap initiation cost), and an intron (gap of arbitrary length in the genomic sequence only) costs –*intron* (40), unless it starts with GT and ends with AG (or CT and AC if the splicing direction is reversed) when it costs –*splice* (20). Thus, a gap of length L costs $L \cdot \text{gap}$ in the spliced sequence and either $\min\{L \cdot \text{gap}, \text{intron}\}$ or $\min\{L \cdot \text{gap}, \text{splice}\}$ in the genome.

The numerical difference between *intron* and *splice* allows some slack in marking intron end-points. Sometimes the choice of boundaries which minimize indel and mismatch costs does not coincide exactly with the splice consensus, but, provided *intron* – *splice* exceeds the extra mismatch/indel costs incurred, the alignment will respect the proper boundaries. If the alignment's introns still do not start/end with GT/AG (or CT/AC), then this may indicate errors in the sequences. The default parameters generally work well except that exons shorter than *splice* may be skipped. Intron penalties should always be greater than the longest expected random match (typically 10–15 bp) to avoid spurious matches.

The details of the algorithm are as follows. Let $X(i, j)$ be the score of the best local similarity ending at base i in the spliced sequence and j in the genomic sequence. Let $B(i)$ be the score of the best local alignment found so far that ends at i in the spliced sequence. Let $C(i)$ be the genome coordinate to which $B(i)$ refers. Let $S(i)$ and $G(j)$ be the nucleotides at positions i in the spliced and j in the genomic sequences, respectively. Then we have:

$$X(i, j) \leftarrow \max \begin{cases} X(i-1, j) & -\text{gap} \\ X(i-1, j-1) & +D \\ X(i, j-1) & -\text{gap} \\ B & \\ 0 & \end{cases}$$

$$D \leftarrow \begin{cases} \text{match} & \text{if } S(i) = G(j) \\ -\text{mismatch} & \text{otherwise} \end{cases}$$

$$B \leftarrow \begin{cases} B(i) - \text{splice} & \text{if } C(i), j \text{ are a donor-acceptor pair} \\ B(i) - \text{intron} & \text{otherwise} \end{cases}$$

$$(B(i), C(i)) \leftarrow \begin{cases} (X(i, j), j) & \text{if } X(i, j) > B(i) \\ (B(i), C(i)) & \text{otherwise} \end{cases}$$

Informatics Group, Sanger Centre, Wellcome Trust Genome Campus, Hinxton Hall, Cambridge CB10 1SA, UK

Present address: Smith Kline Beecham Research & Development, Bioinformatics, New Frontiers Science Park (North), Third Avenue, Harlow, Essex CM19 5AW, UK. E-mail: richard_mott-1@sbphrd.com

