Sketch of the Blast Approach

When comparing two sequences, the various versions of the Blast program perform some combination of the following steps.

1. "Word hits" are identified. By *word* we mean a contiguous segment of one of the sequences that has a pre-determined length (such as 10 for DNA sequences or 3 for protein sequences). A *word hit* consists of a pair of very similar words from each of the sequences. For DNA sequences the words must be identical, whereas for protein sequences they must have a high alignment score. This process is extremely fast, taking time roughly proportional to the sum of the sequence lengths. Intuitively, we make a table of all words in the first sequence, then "look up" each word in the second sequence to see where (and if) it occurs in the first sequence.

2. Each word hit is extended in both directions without introducing gaps. In essence, the word hit is equivalent to a short diagonal path in the alignment graph, and we try to extend the path in both directions so as to increase its score. When we reach a region where extending the path seems only to lower its score, we abandon the search. This step is fast because it searches a region of the alignment graph only 1 node wide, and because we only consider diagonal edges. Each resulting gap-free alignment is called an *HSP*, for high-scoring segment pair, and HSPs whose score is not sufficiently large are discarded.

3. Starting at a central point of each remaining HSPs, we extend in both directions using dynamic programming. To avoid searching the entire alignment graph, we ignore points where the cumulative score falls more than X (a parameter) below the highest score previously seen (during the current extension step). Each of these extensions is quite expensive relative to the previous step, but it is done far less frequently.

4. Once the entire database has been searched, sequences that match the query sequence with comfortably high scores are compared to see which sequence entries are highly conserved, and a *position-specific scoring matrix* is created using this information. For instance, suppose that 10 database sequences strongly match the query, that the A in query sequence position 37 aligns with an A in 9 of the 10 database sequences, while the A at query position 119 is aligned with a hodge-podge of letters in the other sequences. With the position-specific scores, aligning the A at position 37 with an A would be awarded a very high score, whereas aligning the A at position 119 with an A wouldn't score much higher than aligning it with a different letter. Then the database is searched again, using this custom-built matrix for scoring alignments. Of course, the process can be repeated using matching sequences discovered by the second pass through the database.

The original Blast (Altschul *et al.* 1990) performed only steps 1 and 2. The current version of Blastp (for searching protein databases) uses the observation that instead of extending every word hit as in step 2, it works better to use shorter words and require that two hits lie close together on the same diagonal before computing an HSP.

Gapped Blast (made public in 1997) adds step 3. It turns out that to do as well as the older "ungapped" Blast, one can use a higher cutoff value for HSP scores. The reason is that if an interesting alignment contains, say, 3 critical HSPs, then ungapped Blast needs a threshold no larger than the minimum of their scores, but gapped Blast can use the maximum of their scores, since if it finds one of the HSPs then the gapped extension phase will find the others. Using these observations, gapped Blast runs about three times faster than does ungapped Blast, when adjusted to achieve the same sensitivity. Step 4 is applied by position-specific iterated (PSI) Blast.

References

Altschul, S. F., W. Gish, E. Myers, W. Miller and D. J. Lipman (1990) A basic local alignment search tool. *J. Mol. Biol.* **215**, 403–410.

Altschul, S. F., T. Madden, A. Schaffer, J. Zhang, Z. Zhang, W. Miller and D. J. Lipman (1997) Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. *Nucleic Acids Research* **23**, 3389–3402.