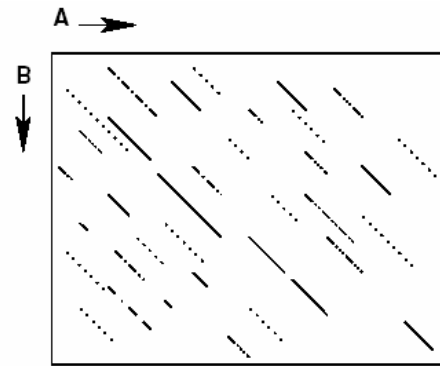


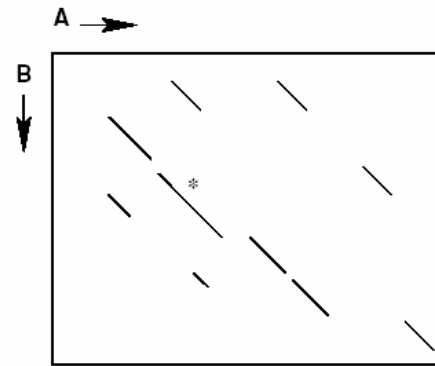
FASTA

FASTA: algorithm (4 steps)

Localize the 10 best regions of similarity between the two seq. Each identity between two "word" is represented by a dot



Identify all k-tuple matches



score the 10 best scoring regions using a scoring matrix

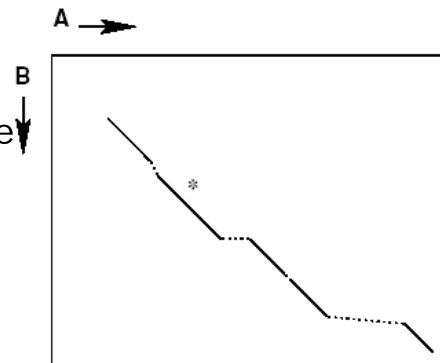
→ Init1 score

Each diagonal: ungapped alignment

The smaller the k,
The sensitive the method but slower

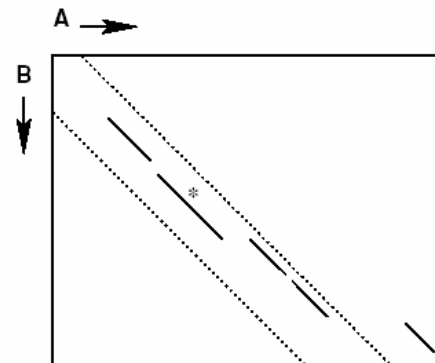
Find the best combination of the diagonals -> compute a score.

Only those sequences with a score higher than a threshold will go to the fourth step



Apply joining procedure

→ Initn score



Apply limited DP

→ Opt score

DP applied around
The best scoring diagonal.

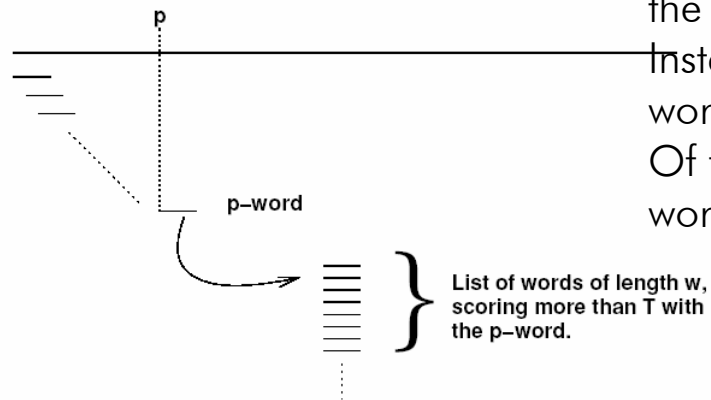
BLAST

BLAST1: Algorithm

With $w=2$:
($20 \times 20 = 400$
Possible words,
 $w=3$, 8000
Possible words,...)

First step:

For each position p of the query, find the list or words of length w scoring more than T when paired with the word starting at p :

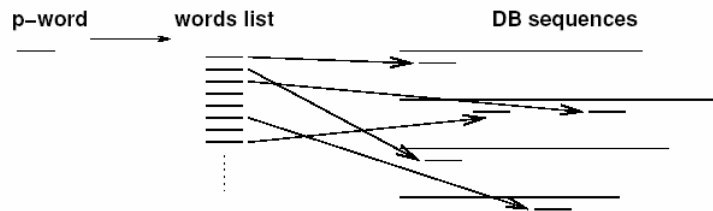


Quickly locate ungapped similarity regions between the sequences.

Instead of comparing each word of the query with each word of the DB: create a list of "similar" words.

Second step:

For each words list, identify all exact matches with DB sequences:

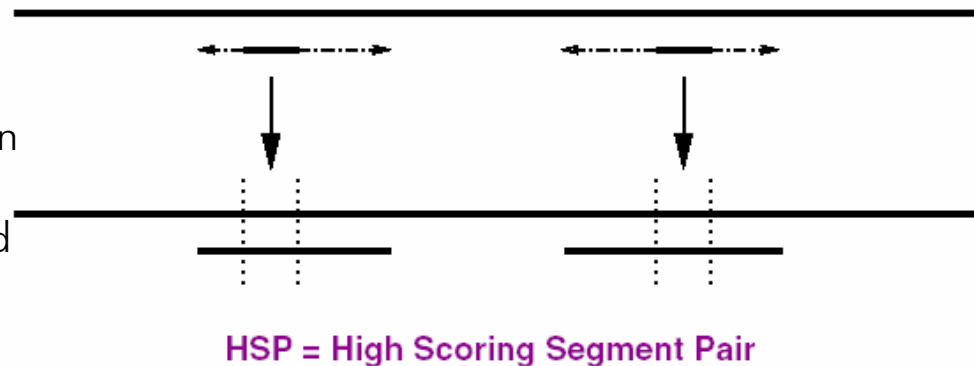


BLAST1: Algorithm

Third step:

For each word match («hit»), extend ungapped alignment in both directions. Stop when S decreases by more than X from the highest value reached by S .

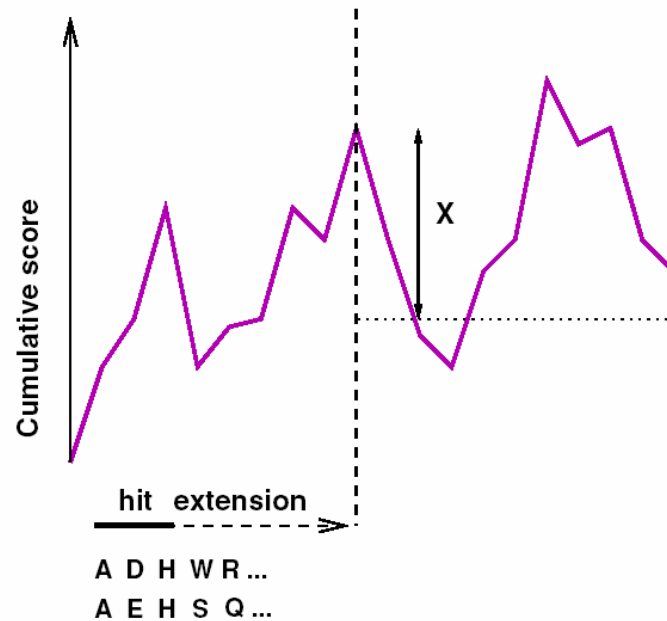
Each match is then extended. The extension is stopped as soon as the score decreases more than X when compared with the highest value obtained. During the extension process



Reports all HSPs having score S above a threshold, or equivalently, having E -value below a threshold.

BLAST1: Algorithm

Ungapped extension of hits



Each match is then extended. The extension is stopped as soon as the score decreases more than X when compared with the highest value obtained. During the extension process

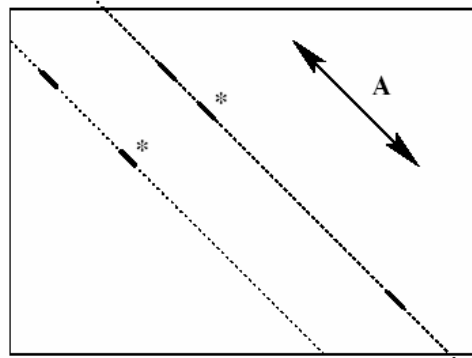
BLAST2: (NCBI)

The «two-hits» requirement

First step: as with BLAST1, generate lists of words scoring more than T with words of the query.

Second step: generation of hits: identify all word matches in DB sequences

Third step: extension of hits: requires a second hit on the same diagonal at a distance of less than A .



Additional step:
Gapped extension of the hits
slower-> therefore: requirement
of a second hits on the diagonal.
(hits not joined by ungapped
extensions could be part of the
same gapped alignment)

This step generates ungapped HSPs

Fourth step: gapped extension of HSPs having score above a threshold S_g