FASTA - Pearson and Lipman (88)

- Earlier version by the same authors, FASTP, appeared in 85
- FAST-A(ll) is query-db similarity search tool
- Like BLAST, FASTA has various flavors
- By now FASTA3 is available
  - changes to FASTA2 and FASTA3 are not well documented
- FASTA looks for the highest scoring subalignments of the query and a few db sequences
  - one alignment per sequence
- The FASTA algorithm goes through 4 steps

Step 1 - find promising diagonals

- FASTA begins by searching for "initial regions": diagonals of high scoring conserved words of length $ktup$
  - $ktup$ defaults: 2 for AA, 6 for DNA
- A diagonal score is the sum of the scores of its conserved words minus the number of residues in between the $ktup$s
  - Conserved AA words are scored by BLOSUM50 (default)
  - DNA words by some constant ($ktup^2$?)

Step 1 - cont.

- Searching for the 10 best scoring diagonals is done similarly to BLAST
- Conserved pairs are identified using a table ($ktup^{|\Sigma|}$)
  - no automaton
- For each d the score and last position are kept
- If the score of the existing diagonal extended by the new word pair is positive, then rank the extended diagonal
  - Otherwise, a new diagonal is started and ranked

Step 2 - gapless alignments from diagonals

- Each of the 10 best diagonals is scored as a gapless alignment and an optimal subalignment is selected
  - no X-dropoff

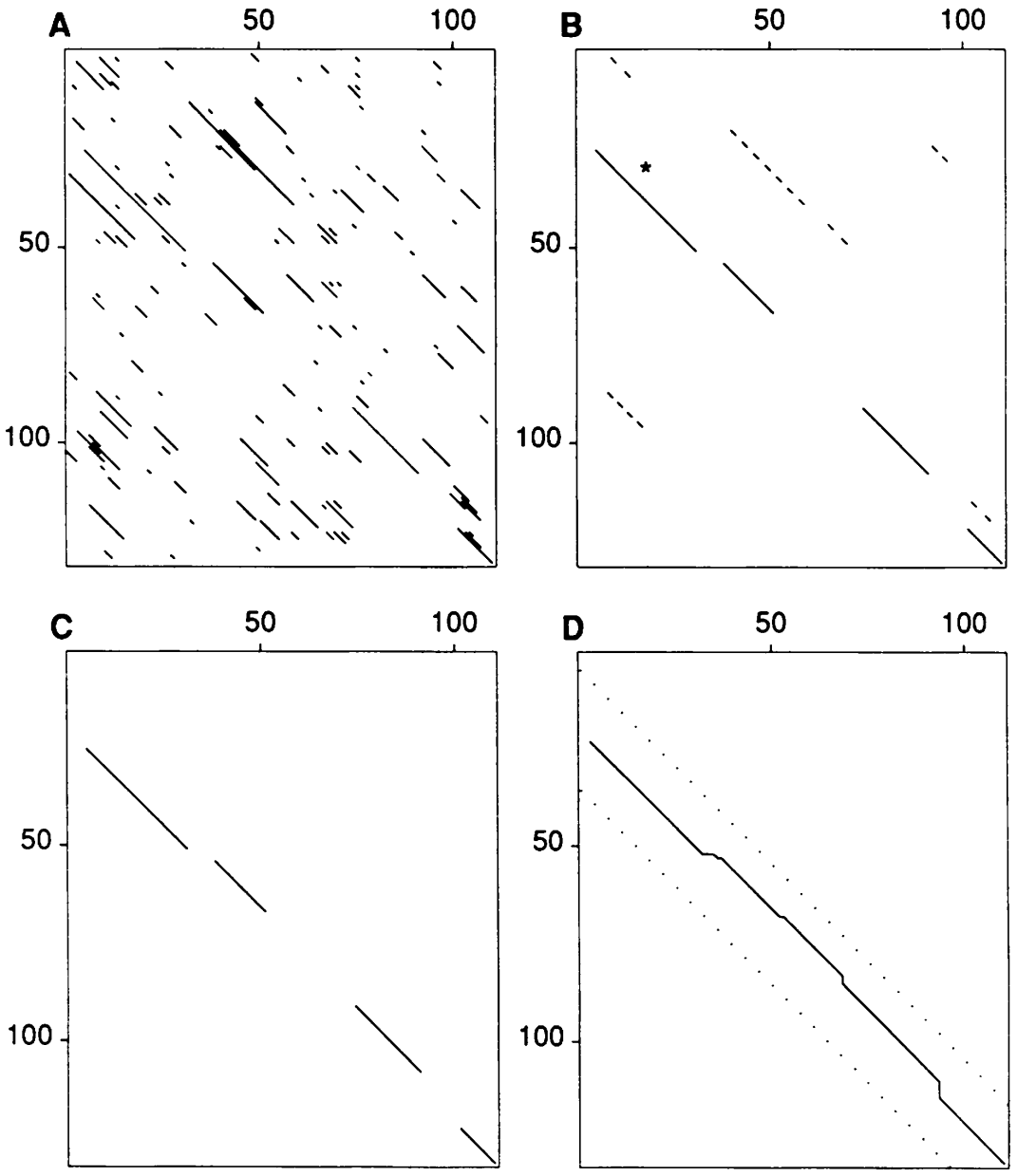
Step 3 - joining high-scoring diagonals

- Try to join consistent diagonals into a skeleton of a gapped alignment
  - consider only diagonals whose score $\geq$ cutoff value
- The score of the skeleton is the sum of the included diagonals minus a "joining penalty" for each gap (default 20)
- A simple DP on a graph will yield the optimal skeleton
- The score of the optimal skeleton is assigned to the corresponding db sequence

Step 4 - banded DP

- The highest scoring library sequences are selected for a banded (32) NW/SW
  - centered on the best initial region (diagonal) that was found in step 2
- The optimized score that FASTA reports is the resulting optimal SW score
- Starting with FASTA2
  - SW is no longer banded(?)
  - Scores are adjusted for db sequence length

FASTA in a picture



LFASTA

- FASTA tries to maximize the similarity score of an alignment based on joining non-overlapping initial regions
  - one alignment per sequence
- LFASTA looks for as many "disjoint" high scoring subalignments as there are
- The first two steps mirrors those of FASTA except that any initial region scoring above T is kept
- These diagonals are subjected first to a backward banded SW starting at its end
  - and continuing past its beginning till all scores are 0
- then to a forward banded SW starting where the maximal backward score was attained and extended till all scores are 0

LFASTA - cont.

- Check for merging of multiple initial regions
- How is T determined?

RDF2

- How to evaluate the statistical significance of FASTA's results?
  - use BLAST's method. . .
- RDF2 is designed to test whether an observed similarity score can be attributed to locally biased AA composition
- It takes the highest ranking optimized scores and shuffles the corresponding db sequences 100-200 times
  - invoking FASTA on each shuffle (db is one shuffled sequence)
  - collect scores of shuffled alignments
- Report the z-value of the observed score: $\frac{s-\mu}{\sigma}$
  - μ and σ are the observed moments of the shuffled scores
  - misleading: the distribution of best optimized score has a much heavier tail than the normal one

RDF2 - cont.

- Report in how many of the shuffles have we failed to reach the unshuffled score
- What about stretches of low complexity?
  - Shuffle within blocks
- What's wrong with this whole approach?
  - We were looking for sequences with a maximal unshuffled score
  - Given that, it is not true that the shuffled sequences follow a uniform distribution even under H_0