MAFFT

A Novel Method for Rapid Multiple Sequence Alignment Based on Fast Fourier Transform

Paper by Katoh, Misawa, Kuma, and Miyata, 2002.
Presented by Jun Inoue

Accuracy-Time Tradeoff in Multiple Alignment

- ClustalW, T-Coffee bad at handling large insertion/extension
  - Or just long sequences
- They're getting more accurate, but also slower
  - Mostly $O(N^2)$
  - Not much effort to improve speed
- As of 2002

Approach

- Address both speed and accuracy
- Rapid homology detection using FFT
- Simplified, faster scoring that works well with large ins/ext
- Trait-based structural alignment
- Iterative refinement (trade speed)

Trait-based Alignment

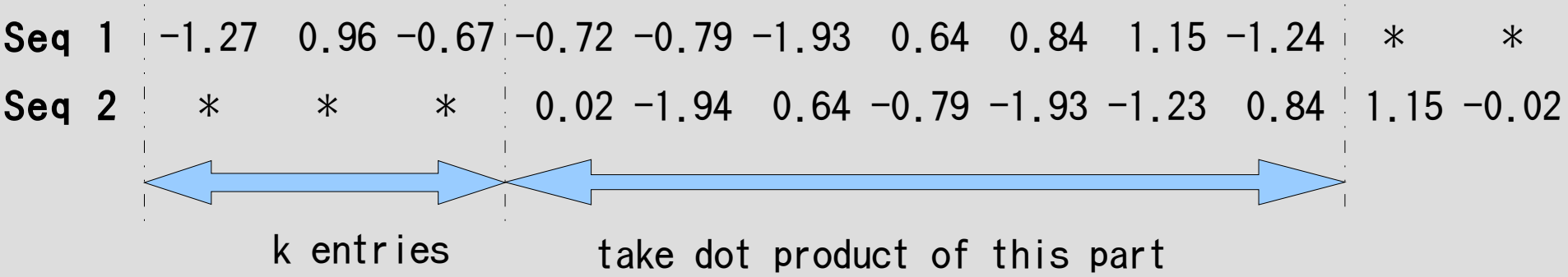
- Convert amino acid sequence to a pair (seq of volume, seq of polarity)

```
Given   : FSF
Produce: v=( 1.14, -1.24,  1.14) and
          p=(-1.19,  0.33, -1.19)
  where F = Phenylalanine, S = Serine
```

- Units are standard deviations from the mean among the 20 amino acids coded by codons
  - Makes data dimensionless

FFT for Homology Detection

- Correlation at positional lag k
  - c(k) := cv(k) + cp(k)
  - cv(k) := dot product of the overlapping regions of the sequences $v_1(n)$ and $v_2(n-k)$,
  - same for cp(k)
- High c(k) = homologous segments are aligned



FFT for Homology Detection

- Formally,

$$c_v(k) = v_1 * v_2 = \sum_n v_1(n)\, v_2(n-k)$$

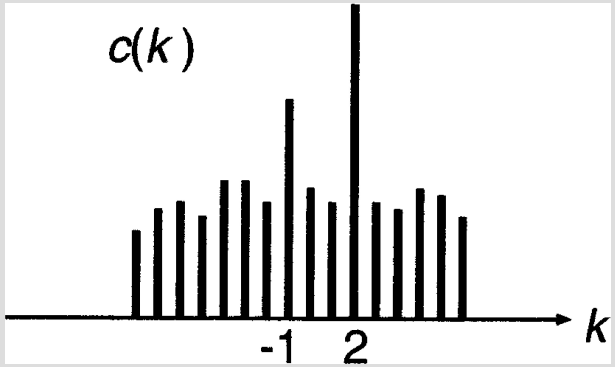
- Computing this for all k is $O(N^2)$
- Use FT identity

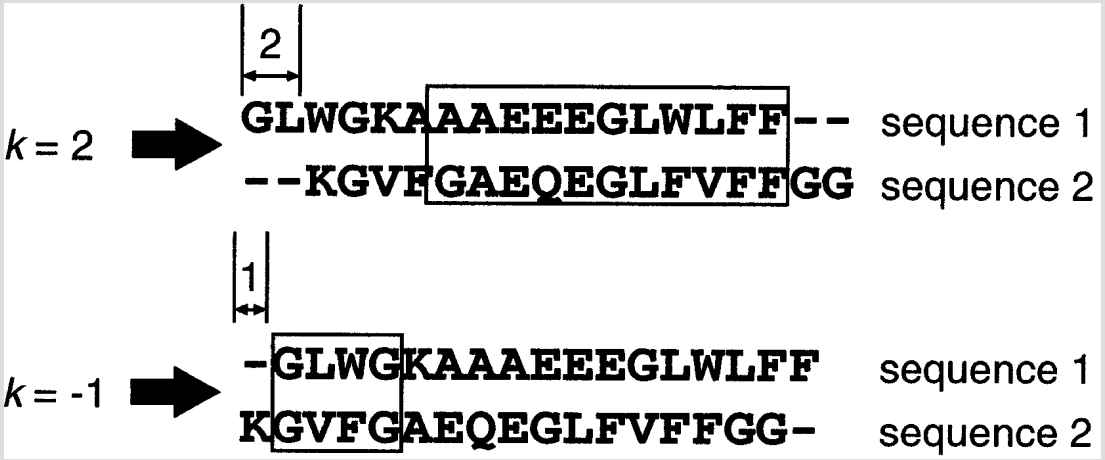
$$\mathcal{F}\, c_v = \mathcal{F}\,(v_1 * v_2) = mult(\overline{\mathcal{F}\, v_1}, \mathcal{F}\, v_2)$$

which gets us O(N) with O(N log N) setup.

FFT for Homology Detection

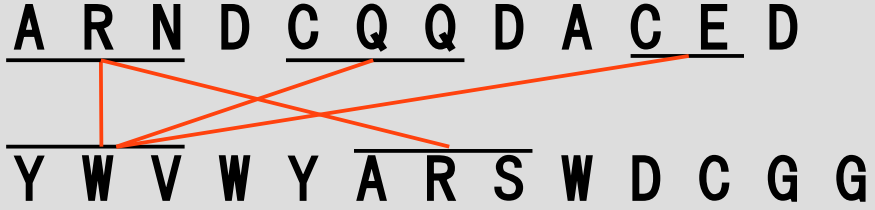
- High peaks indicate lags for which we (hopefully) have homologues aligned
- Sliding window analysis to find homologous segments





Completing the Alignment

- FFT homologue detection gives a many-to-many relation



- Align segments as atomic symbols
  - Effectively gives shorter sequence to align

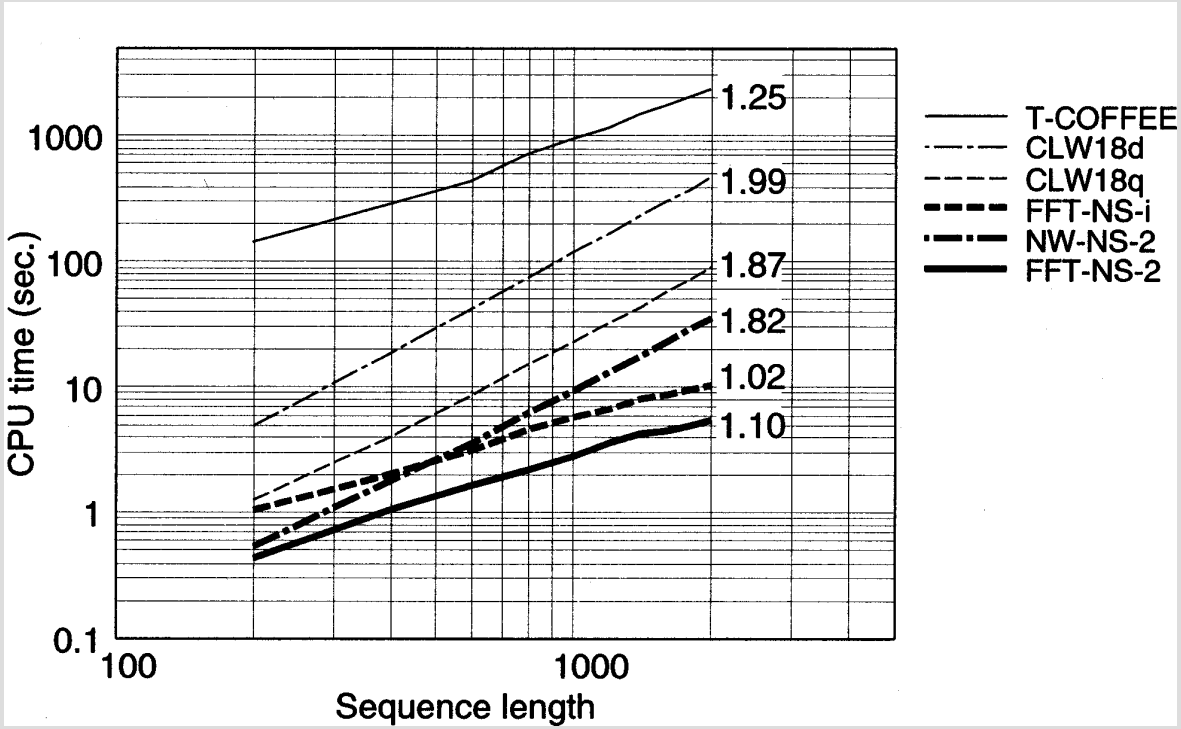
Multiple Alignment

- To align two groups G1, G2 of sequences:
  - Sum (with weight) the volume/polarity sequences of all sequences in G1, G2
- Find homologues as in two sequences

Scoring

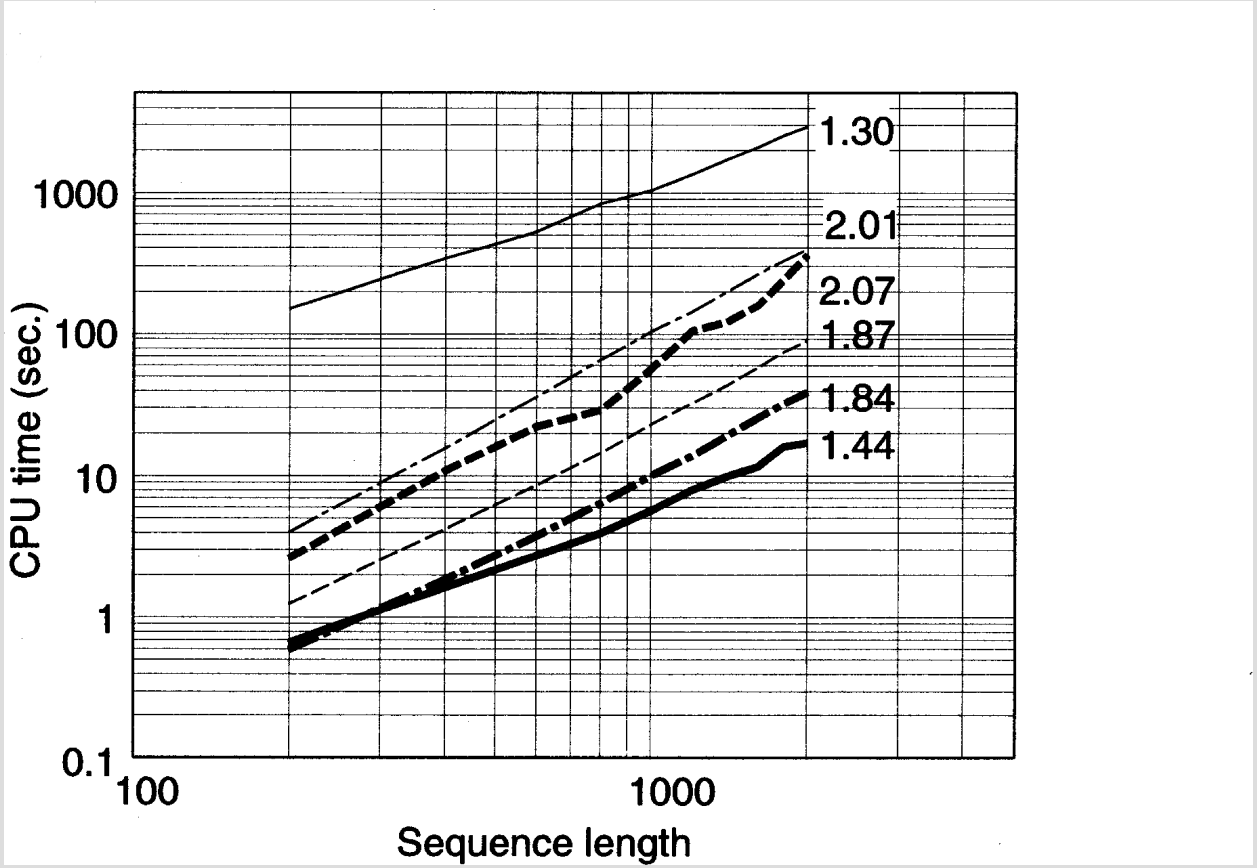
- A substitution matrix with some hairy normalization
  - helps exaggerate peaks in c(k)?
- Gap penalty based on where gap starts/ends
  - G(i, x) = Eopen (1 – (gs(x) + ge(i))/2)
  - gs(x) = # gaps starting at x
  - ge(i) = # gaps starting at i

Results



Performance vs length,
data has 35-85%
conservation rate

Results



Performance vs length,
data has 15-65%
conservation rate

Results

- Other impressive data points: Ribosomal Database Project (RDP-II)
  - T-COFFEE aborts with memory exhaustion on full test
  - T-COFFEE scores 0.806 SOP in stripped ver. Takes 35860 [sec] to complete
  - MAFFT score 0.816 SOP in stripped ver. Takes 181 [sec]

(Their) Conclusions

- MAFFT is fast, O(N)
- No need to have different algorithms for different sequences. Simple algorithms can match the accuracy of sophisticated algorithms like T-COFFEE

~~Nitpicking~~ Discussion

- They claim O(N) based on foregoing graphs
  - Sloppy! Where did the number of seqs go?
- Data set seems small
  - e.g. 59 seqs, 2810-5183 sites each
  - Was this the right ballpark back then? What about now?