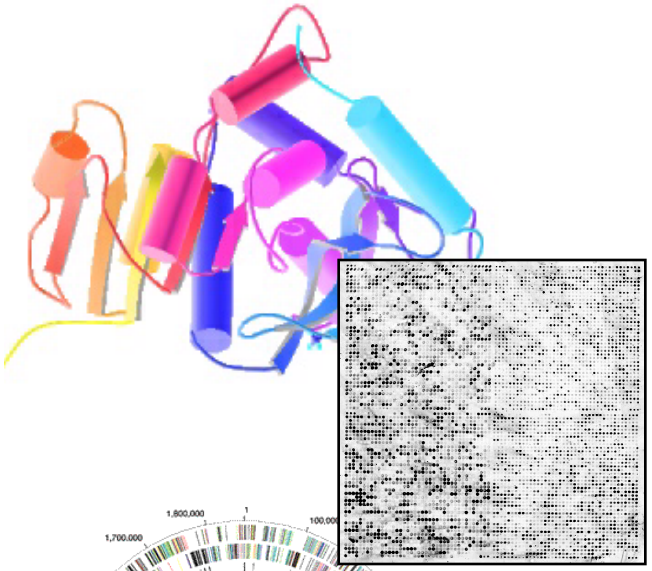
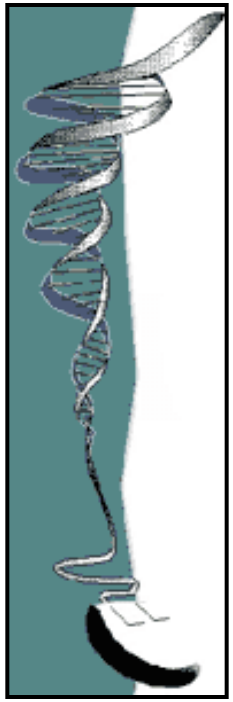
Biomed. Data Sci.
Fast Alignment





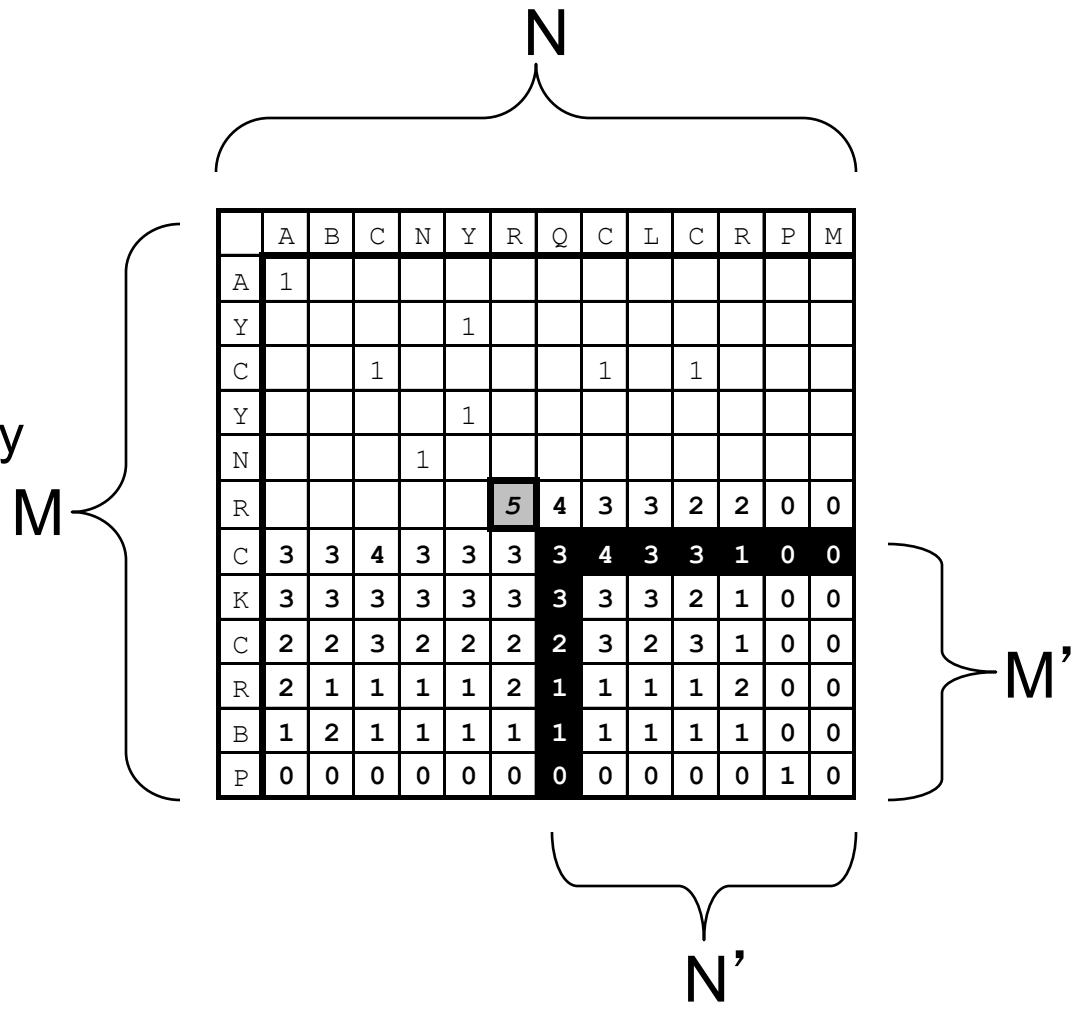
Mark Gerstein
Yale University
gersteinlab.org/courses/452
(last edit in spring ’20)

Computational Complexity

- The dynamic programming alignment algorithm is
  $O(n\ m) \sim O(n^2)$
  in speed and memory

$O(n^2)$ in speed and memory is not good enough for important applications

- database search
- short read alignment to reference genome



Fast sequence alignment

- Alignment via dynamic programming (NW/SW)
  - ◊ useful for aligning the small numbers of protein, DNA sequences available in the 1980s
- 1990s hundreds of thousands of protein sequences
- Today thousands of genome sequences

- => need for faster, more coarse-grained alignment methods
  - ◊ first application: find your favorite protein in a sequence database
  - ◊ next-gen seq application: align millions of short reads to a reference database

Computational Complexity

- Designing algorithms involves a trade-off between calculation time and memory usage & sensitivity

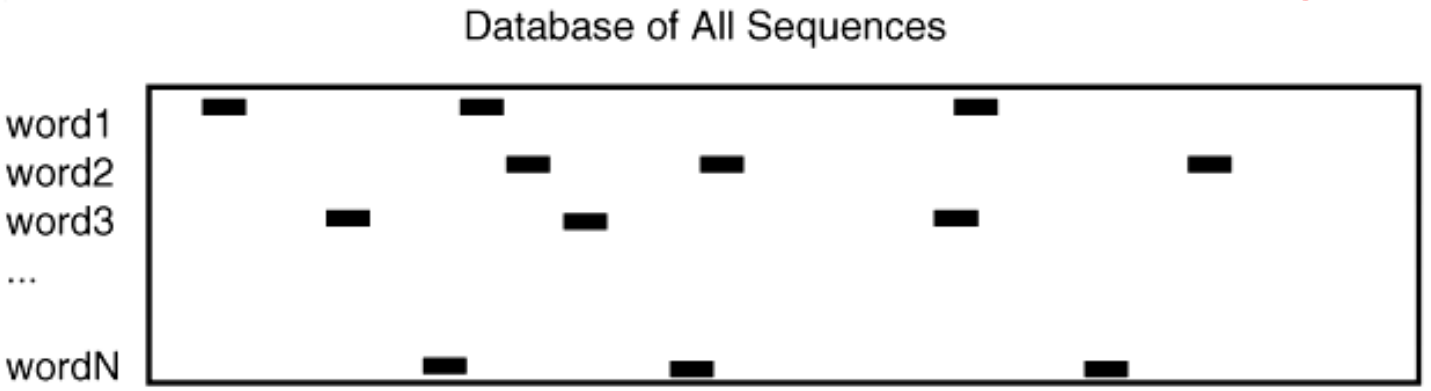
- Steps that can be pre-calculated and stored efficiently in memory speed up the algorithm

- FASTA (hashing the query)
- BLAST (more efficient query hashing)
- BLAT (hashing the DB)
- BWA / Bowtie (BW transform of the DB)

FASTA

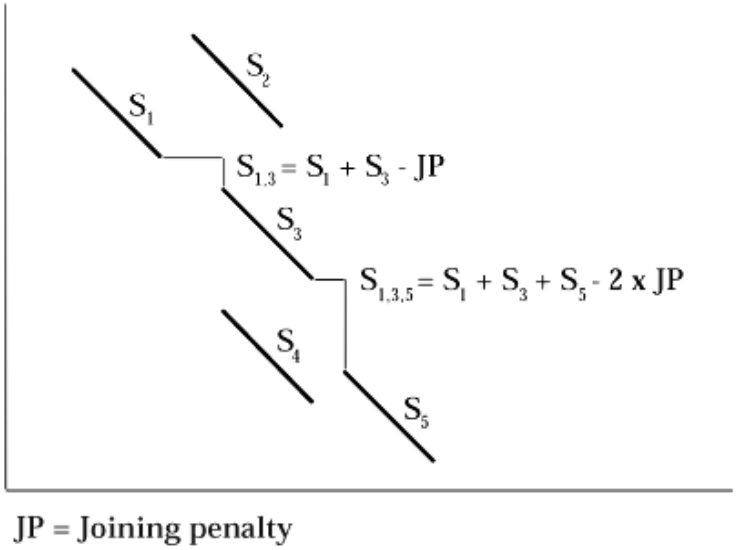
- Hash table of short words in the query sequence
- Go through DB and look for matches in the query hash (linear in size of DB)
- perl: $where{“ACT”} = 1,45,67,23....
- K-tuple determines word size (k-tup 1 is single aa)
- by Bill Pearson

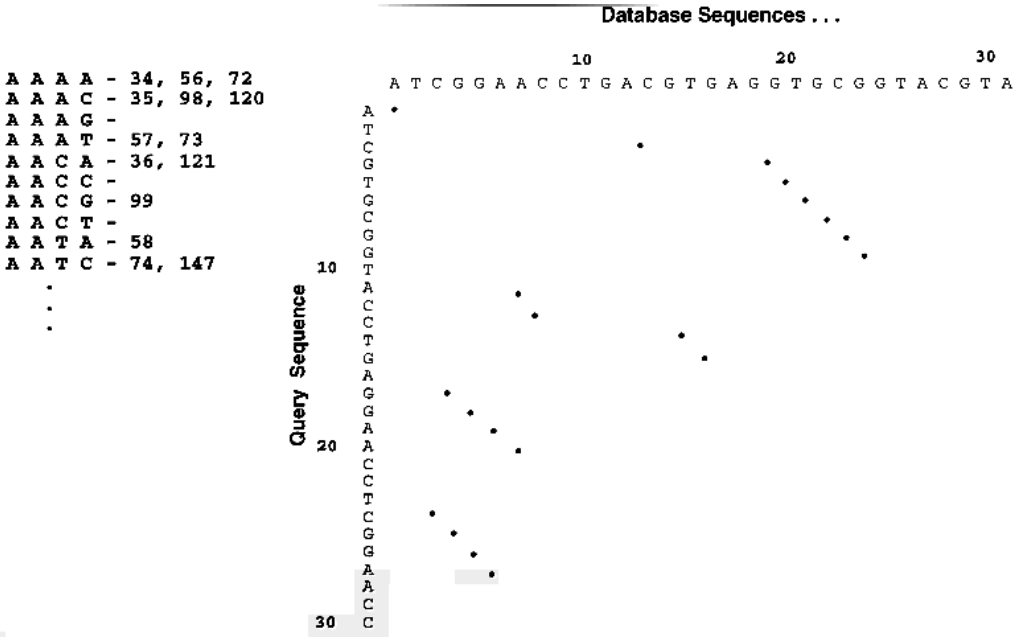
VLICT = 

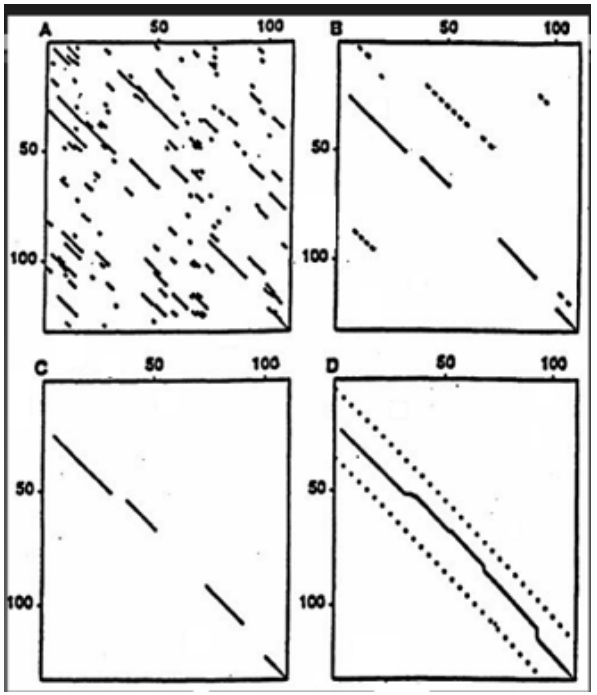


VLICTAVLM**VLICT**AAA**VLICT**MSDFFD

Join together query lookups into diagonals and then do a full alignment





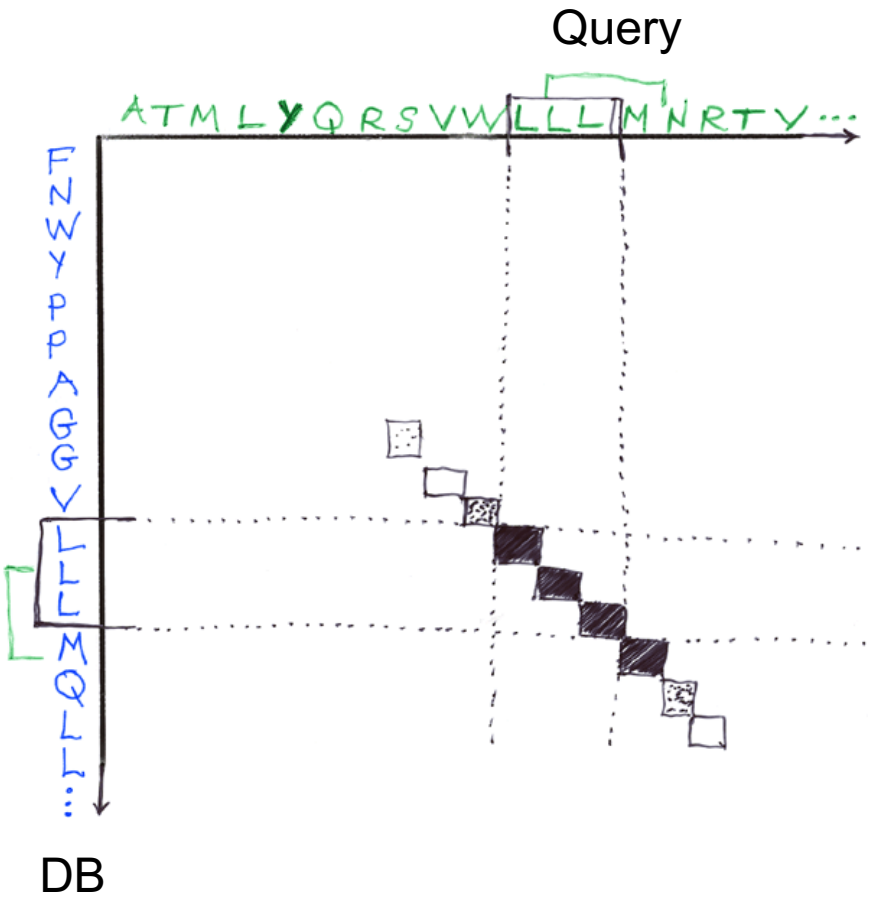


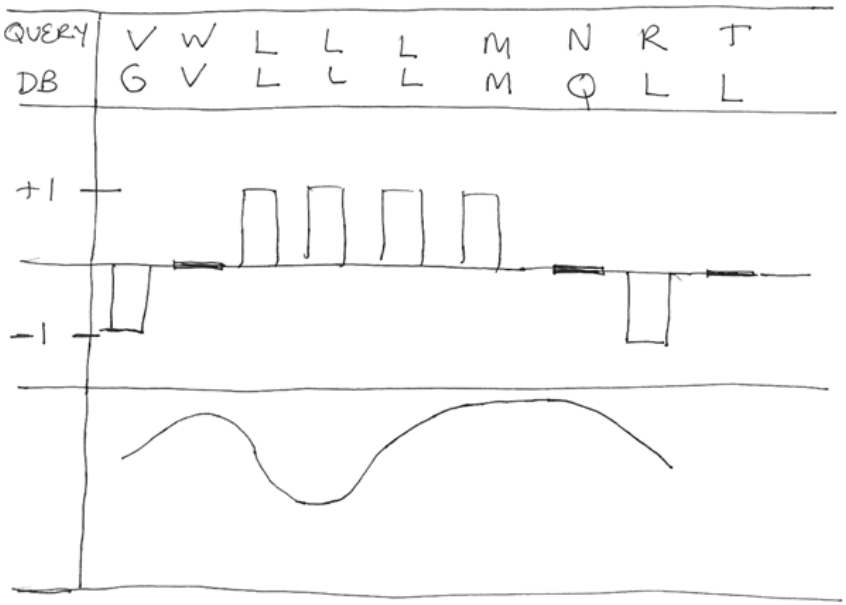
(Adapted from D Brutlag)

Basic Blast

- Altschul, S., Gish, W., Miller, W., Myers, E. W. & Lipman, D. J. (1990). Basic local alignment search tool. *J. Mol. Biol.* **215**, 403-410
- Indexes query
- Starts with all overlapping words from query
- Calculates “neighborhood” of each word using PAM matrix and probability threshold matrix and probability threshold
- Looks up all words and neighbors from query in database index
- Extends High Scoring Pairs (HSPs) left and right to maximal length
- Finds Maximal Segment Pairs (MSPs) between query and database
- Blast 1 does not permit gaps in alignments

BLAST: Basic Local Alignment Search Tool





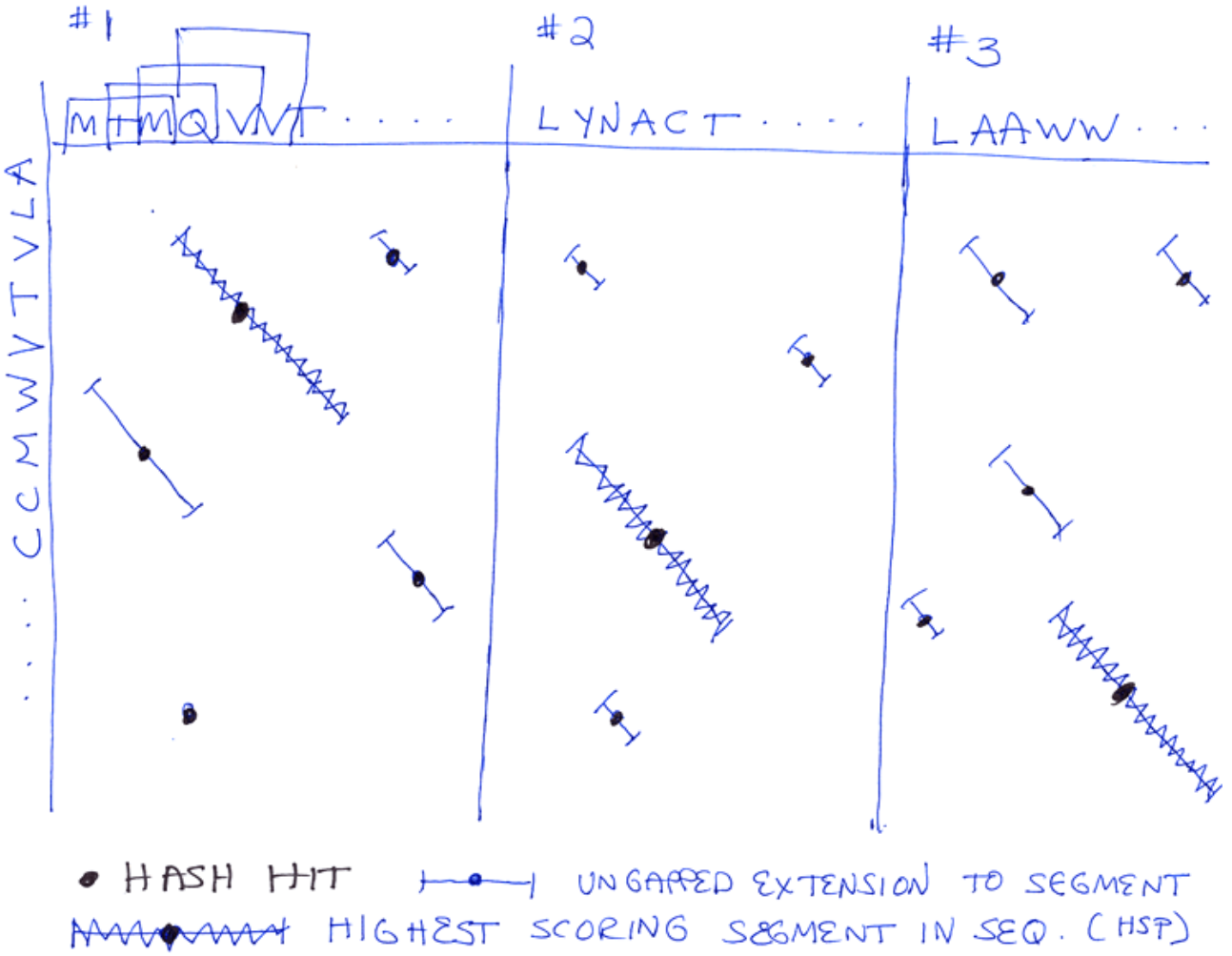
- Extend hash hits into High Scoring Segment Pairs (HSPs)
- Stop extension when total score doesn't increase
- Extension is O(N). This takes most of the time in BLAST

BLAST: Basic Local Alignment Search Tool

- In simple BLAST algorithm, find best scoring segment in each DB sequence
- Statistics of these scores determine significance

Number of hash hits
~ O(N*M*D)

where
N is query size
M is average size of seq in DB
D is DB size



Blast2: Gapped Blast

Nucleic Acids Research, 1997, Vol. 25, No. 17 3389–3402

Gapped BLAST and PSI-BLAST: a new generation of protein database search programs

Stephen F. Altschul*, Thomas L. Madden, Alejandro A. Schäffer[1], Jinghui Zhang, Zheng Zhang[2], Webb Miller[2] and David J. Lipman

National Cent
Bethesda, MD
Institute, Natio
Engineering, F

Received June 2

ABSTRACT

The BLAST
searching pr
similarities.
definitional,

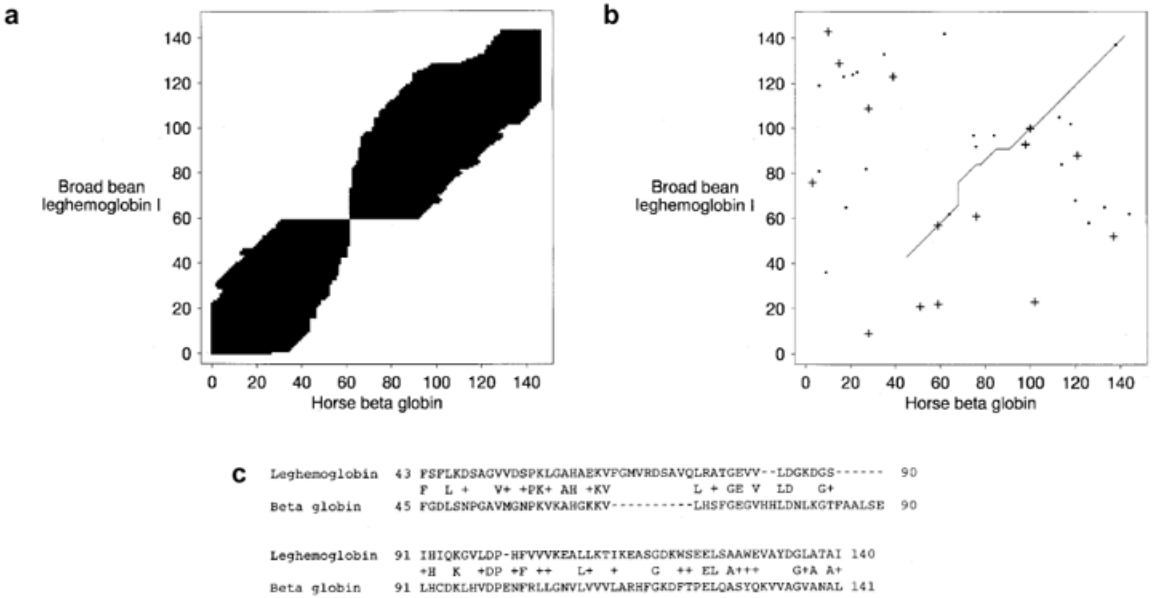
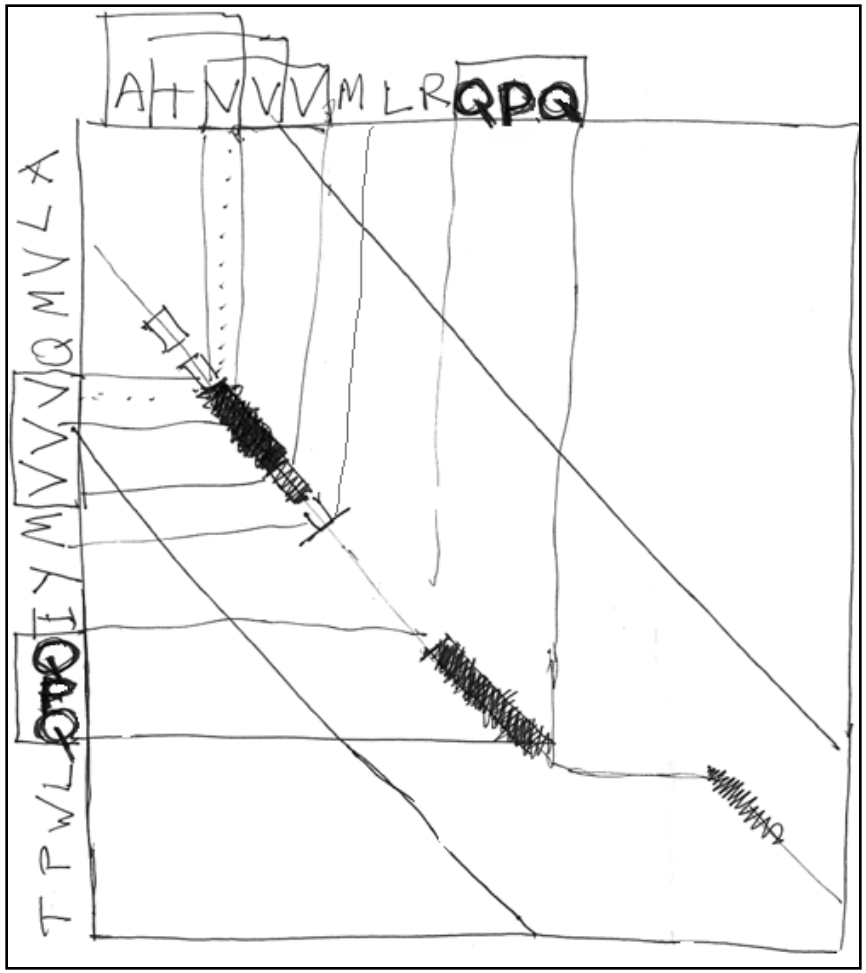


Figure 3. A gapped extension generated by BLAST for the comparison of broad bean leghemoglobin I (87) and horse β-globin (88). (**a**) The region of the path graph explored when seeded by the alignment of alanine residues at respective positions 60 and 62. This seed derives from the HSP generated by the leftward of the two ungapped extensions illustrated in Figure 2. The X_g dropoff parameter is the nominal score 40, used in conjunction with BLOSUM-62 substitution scores and a cost of $10 + k$ for gaps of length k. (**b**) The path corresponding to the optimal local alignment generated, superimposed on the hits described in Figure 2. The original BLAST program, using the one-hit heuristic with $T = 11$, is able to locate three of the five HSPs included in this alignment, but only the first and last achieve a score sufficient to be reported. (**c**) The optimal local alignment, with nominal score 75 and normalized score 32.4 bits. In the context of a search of SWISS-PROT (26), release 34 (21 219 450 residues), using the leghemoglobin sequence (143 residues) as query, the E-value is 0.54 if no edge-effect correction (22) is invoked. The original BLAST program locates the first and last ungapped segments of this alignment. Using sum-statistics with no edge-effect correction, this combined result has an E-value of 31 (21,22). On the central lines of the alignment, identities are echoed and substitutions to which the BLOSUM-62 matrix (18) gives a positive score are indicated by a '+' symbol.

Blast2: Gapped Blast

- Gapped Extension on Diagonals with two Hash Hits
- Statistics of Gapped Alignments follows Extreme Value Distribution (EVD) empirically



Short read alignment to a reference genome

- BLAT
- Burrows-Wheeler transform

BLAT

- "BLAST-like alignment tool"
- created by Jim Kent (UCSC) during assembly of the human genome

- Where BLAST builds an index of the query sequence, BLAT builds an index of the database.
  - ◊ Obviously, this will scan more quickly through the DB at the expense of building a huge hash table of the DB initially
  - ◊ DB index non-overlapping, potentially sacrificing some sensitivity for decreased memory usage

BLAT – Genome Res. (2002) 12: 656

Burrows Wheeler Transform

- What's next: more sophisticated ways of organizing the genome pre-search to speed things up beyond building the DB hash table as in BLAT

- High Level
  - ◊ Build a BWT of the genome (cyclically permuting, then sorting, then compressing)
  - ◊ Then build a prefix tree of this
  - ◊ Take each read and search along the prefix tree in linear time
  - ◊ Reverse the transform to find the location of the read in the genome from its position in the prefix tree.

Burrows and Wheeler, A Block-sorting Lossless Data Compression Algorithm, Digital Systems Research Center Report 124 (1994)

Burrows Wheeler Transform

- BWT is a reversible permutation of the characters in a string X

1. build matrix of cyclic rotations of X
2. sort matrix alphabetically

```
example:   X = acaacg

0 acaacg$          6 $acaacg
1 caacg$a          2 aacg$ac
2 aacg$ac          0 acaacg$
3 acg$aca   =>     3 acg$aca
4 cg$acaa          1 caacg$a
5 g$acaac          4 cg$acaa
6 $acaacg          5 g$acaac
```

Bowtie – Genome Biol. (2009) 10: R25

Burrows Wheeler Transform

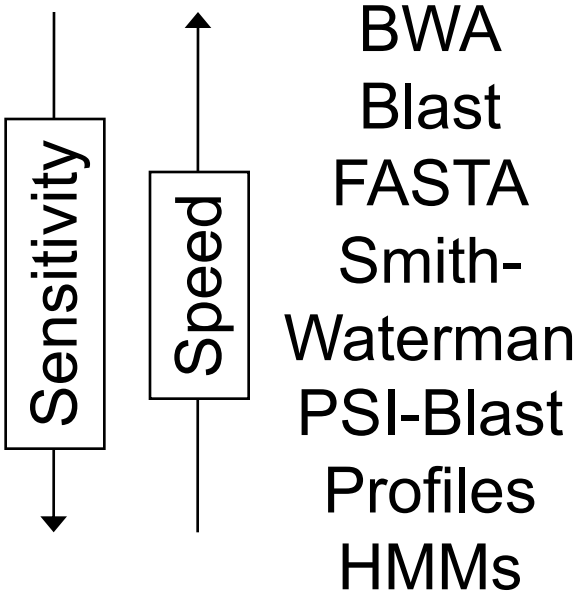
- BWT is a reversible permutation of the characters in a string X

1. build matrix of cyclic rotations of X
2. sort matrix alphabetically
3. Extract
   - last column L of BWT matrix
   - position I of original string X in matrix

Bowtie – Genome Biol. (2009) 10: R25

Speed v Sensitivity Tradeoff

PSI Blast as a form of Semi-supervised learning



What sequence alignment algorithms need to be designed next?

A couple of important problems:

- rapidly align a personal genome to a reference population of human genomes
  - ◊ with clinical turn-around time; with privacy => encryption?
- 3rd generation sequencers: long, error-prone reads
  - ◊ useful as scaffolds mixed with more accurate, cheaper short reads