Biological Sequence Alignment

Wei Zhu

Rice Genome Annotation Workshop

May, 2007Intron Splicing (higher organisms)

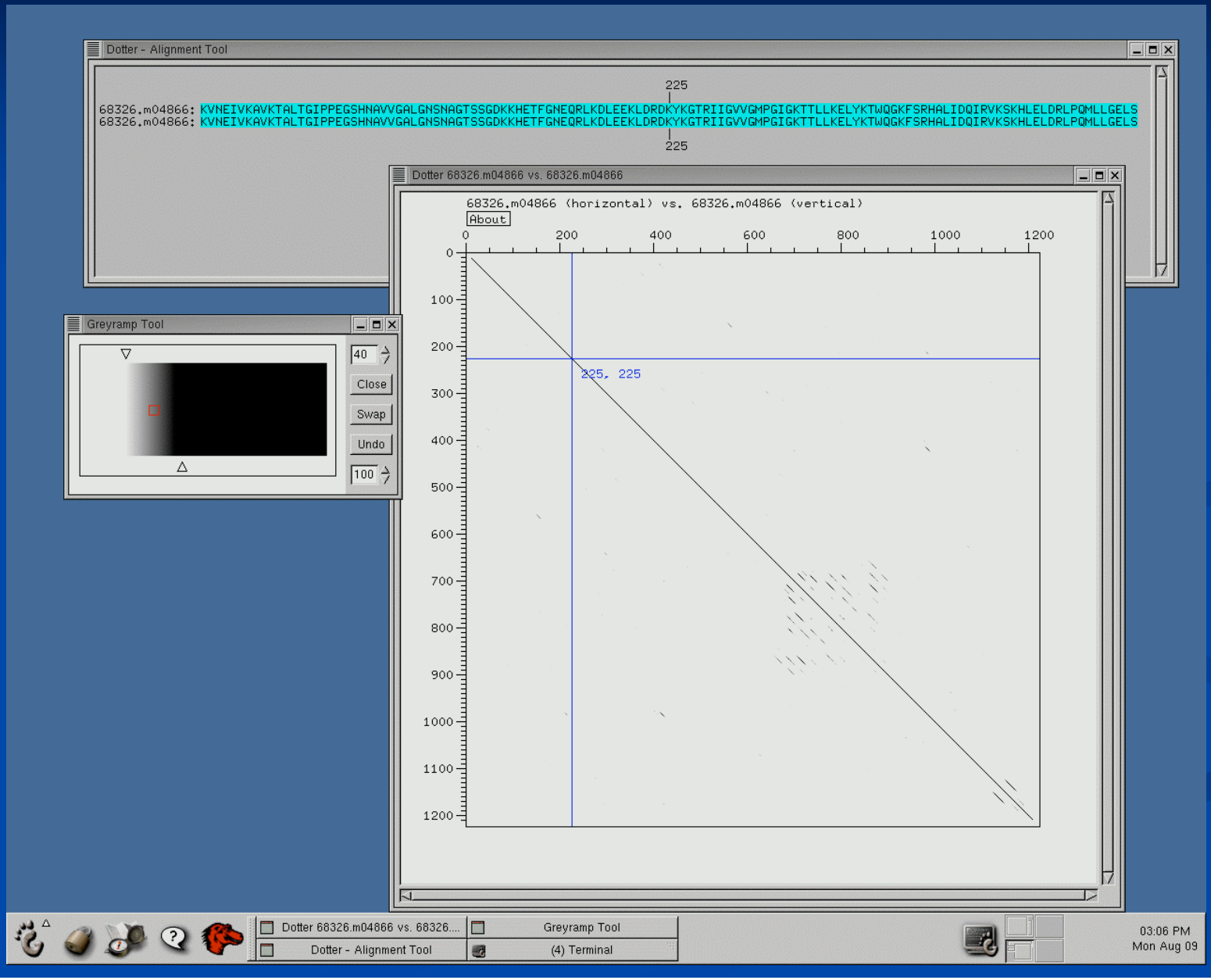
Reasons for Sequence Comparison

- To compare related sequences.
  - align two related protein sequences across their entire length, examine sequence differences.
- Identify local regions of sequence similarity
  - individual motifs or domains conserved.
- Search a sequence database for related sequences.
  - given a sequence, what is it? Find relatives for which some function may be known.
- Phylogenetic analysis (evolutionary trees)
  - sequence differences represent ticks in the molecular clock.
- Complete genome alignments
  - large scale duplications, inversions, translocations, including micro-differences such as single nucleotide polymorphisms (SNPs).

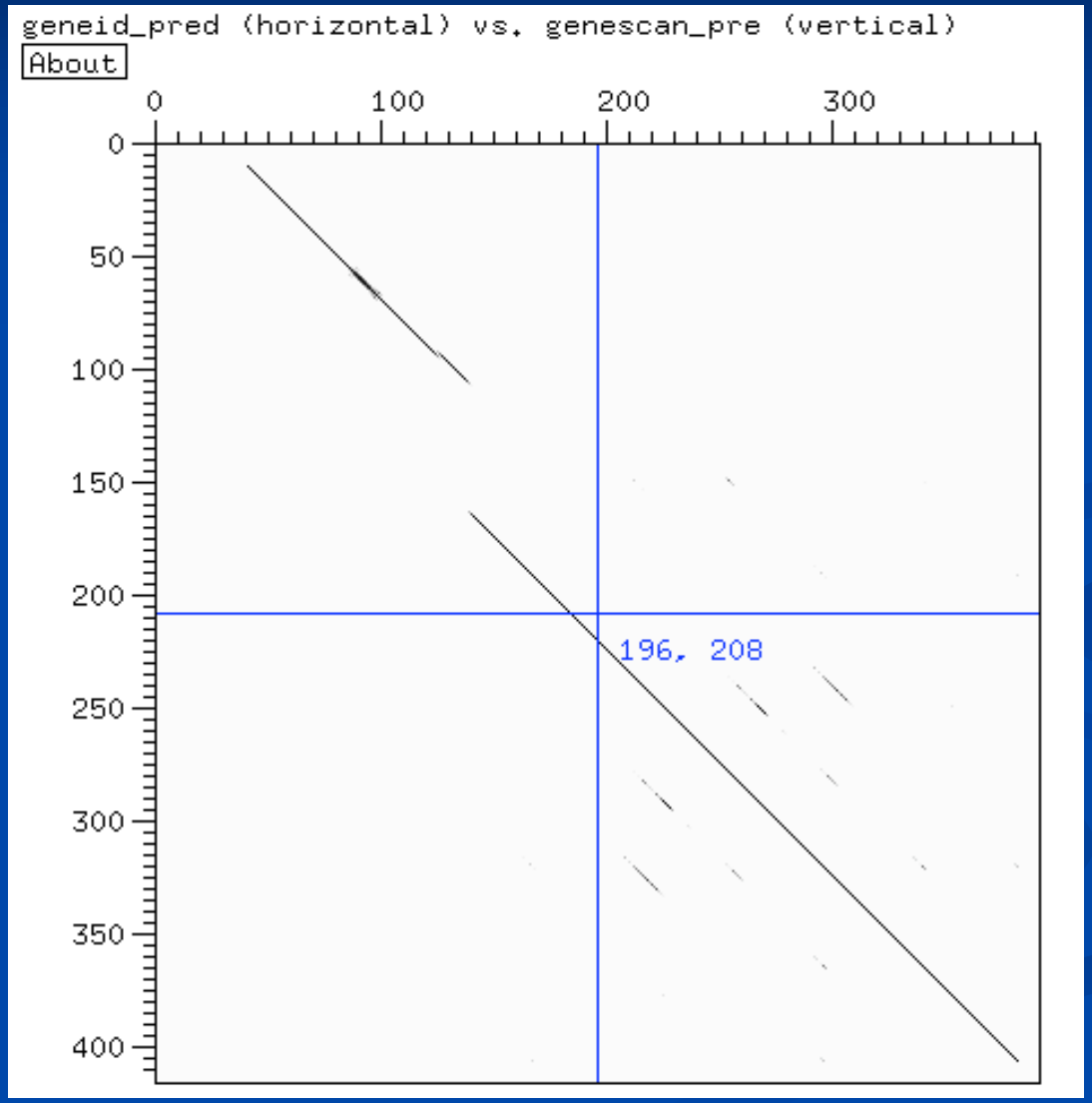
Simplest Method for Sequence Comparison *Dotter*

- Examine sequence similarities in the context of a dot plot.
- Regions of sequence similarity show up as diagonals in the plot.

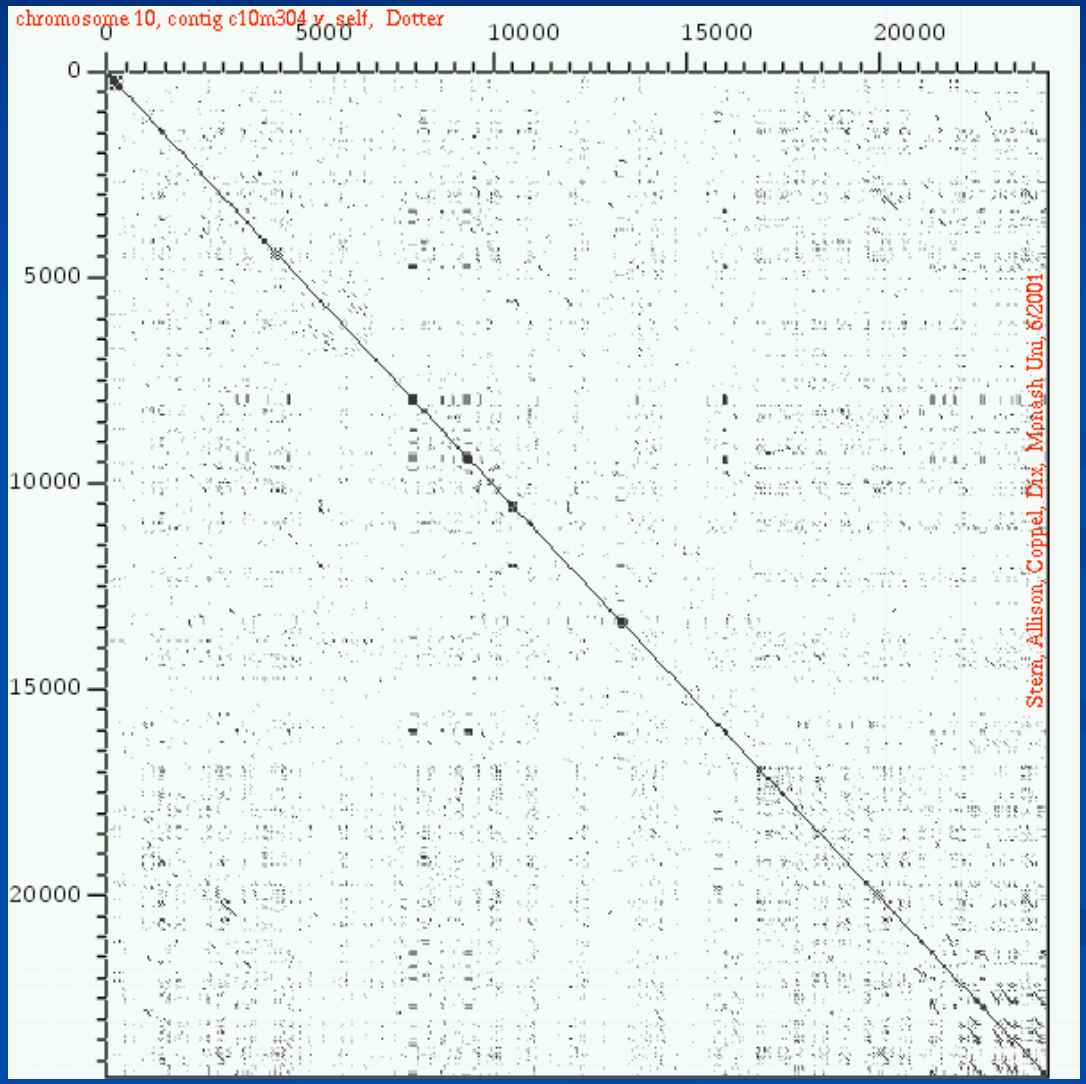
Dotter screenshot, rps4 protein self comparison



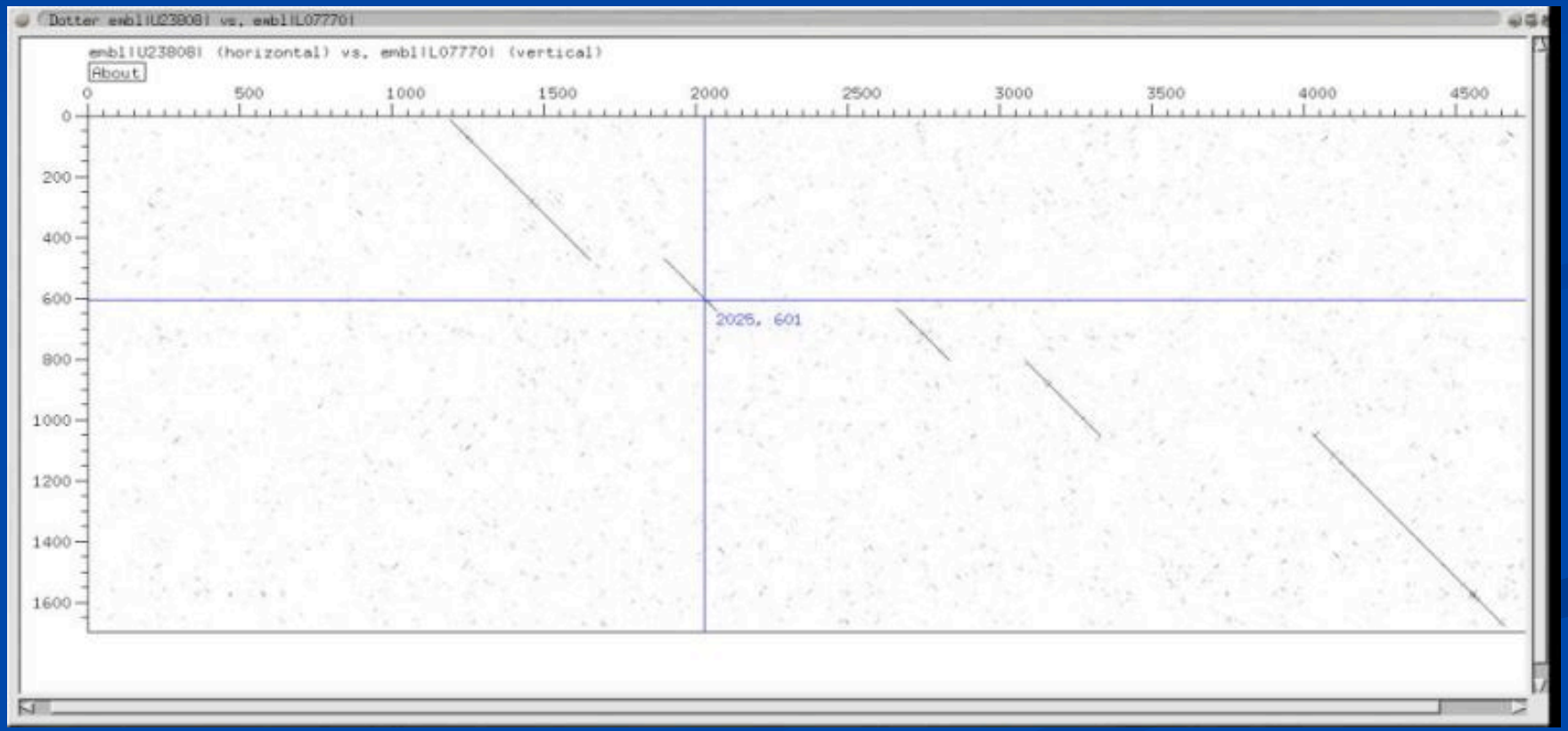
Insertion/Deletion

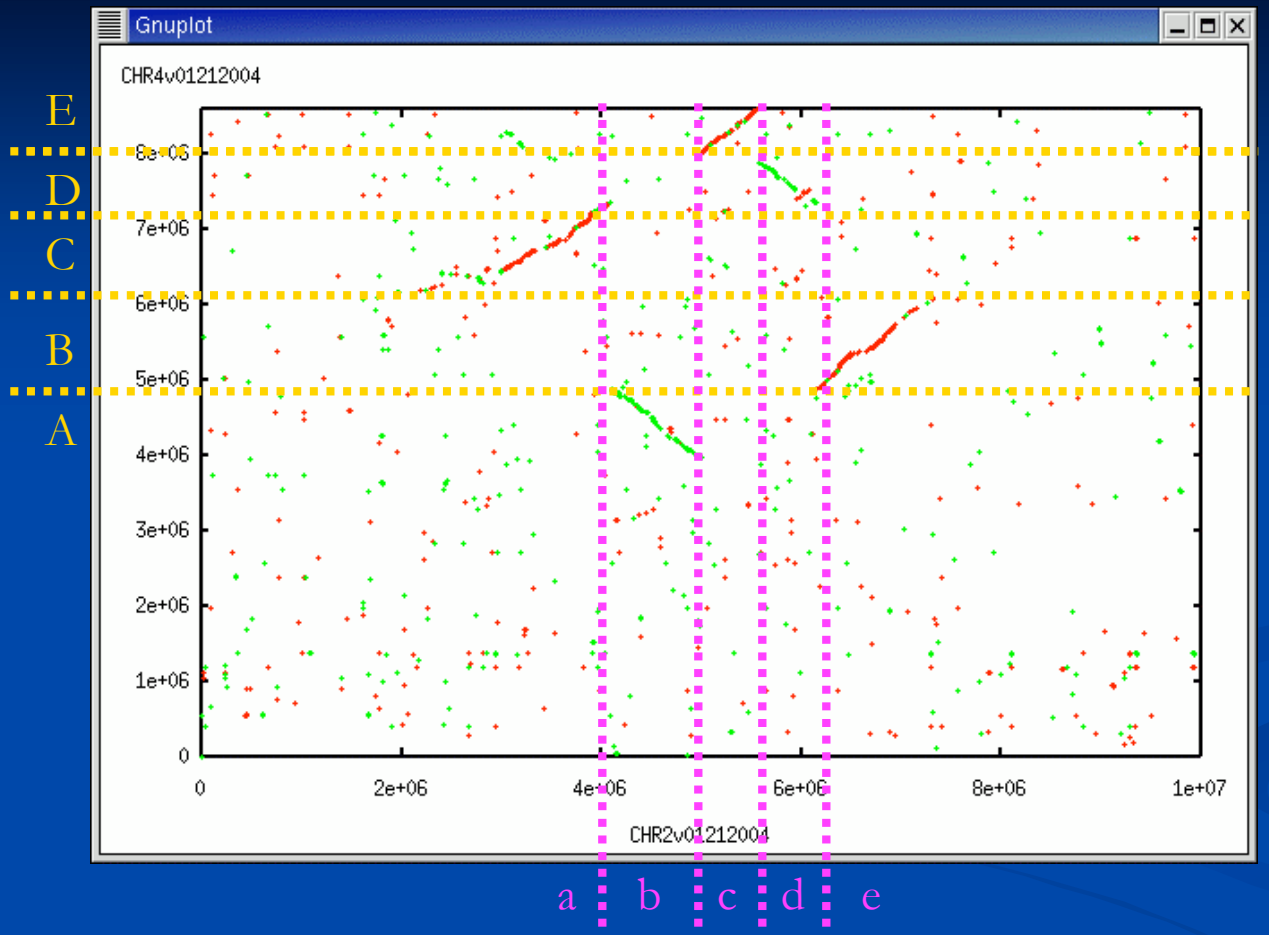


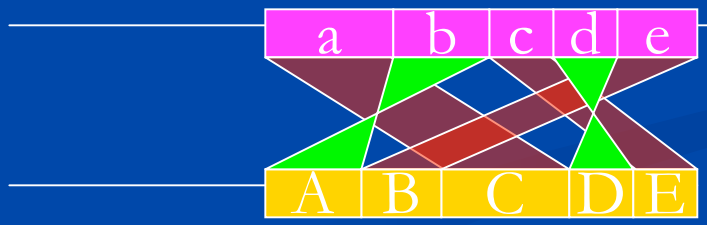
Repetitive Pattern



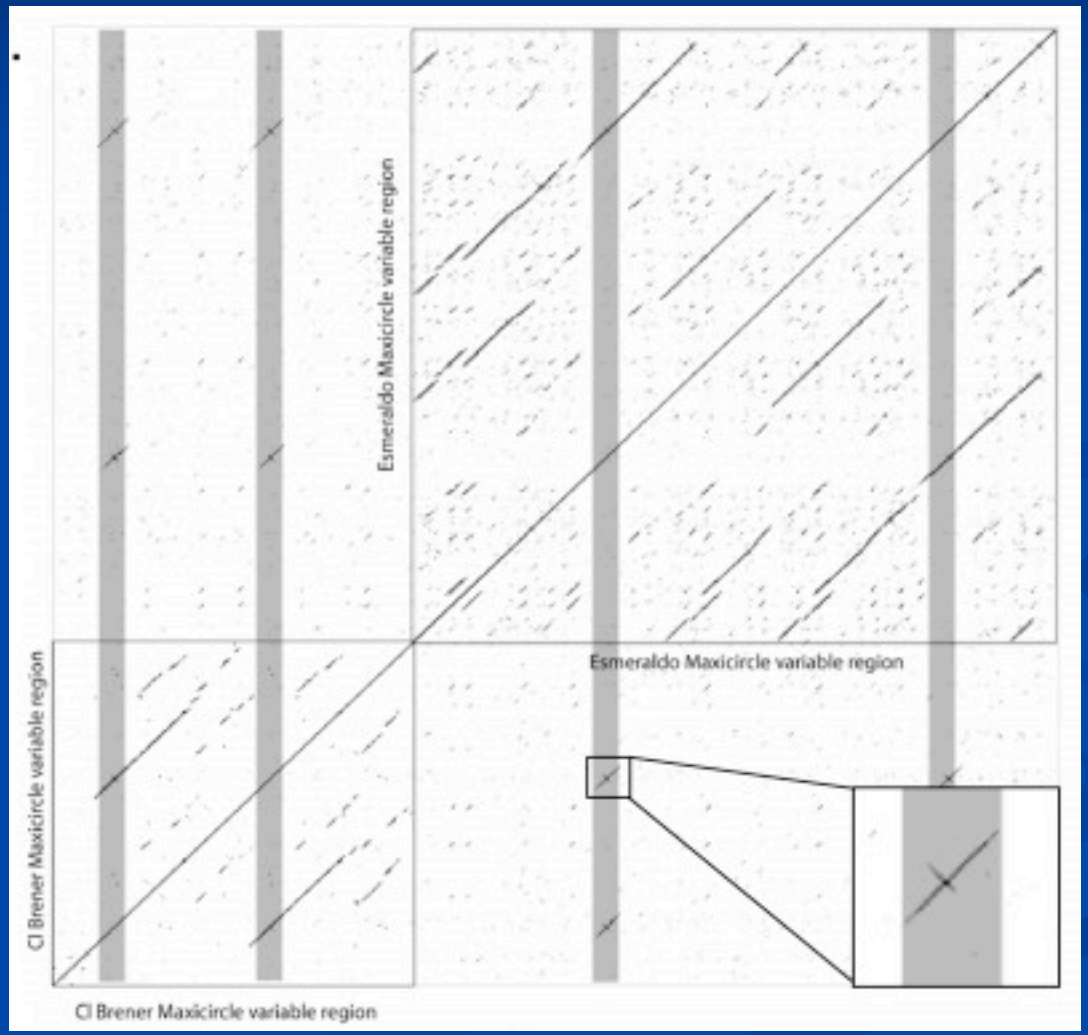
Spiced Alignment




Gnuplot
CHR4v01212004
E
D
C
B
A
8e+06
7e+06
6e+06
5e+06
4e+06
3e+06
2e+06
1e+06
0
0
2e+06
4e+06
6e+06
8e+06
1e+07
CHR2v01212004
a
b
c
d
e


a
b
c
d
e
A
B
C
D
E

Palindrome


Esmeraldo Maxicircle variable region
Esmeraldo Maxicircle variable region
Cl Brener Maxicircle variable region
Cl Brener Maxicircle variable region

Algorithmic Approaches to Sequence Alignment

- Dynamic Programming
- Heuristic search (seed and extension)
  - Fixed-length word matches

    indexing, hashing techniques
  - Suffix Trees

    exact word matching, arbitrary length

Forms of Sequence Alignments

Align 2 sequences:
•HEAGAWGHEE
•PAWHEAE

- Global alignment
  - every position in one sequence is aligned to a position in a second sequence or across a gap.

```
HEAGAWGHE-E
--P-AW-HEAE
```

- Local alignment
  - the alignment between two subsequences which has the highest sequence similarity.

```
AWGHE
AW-HE
```

I. Sequence Alignment Using Dynamic Programming

- Dynamic programming:
  - recurrence relation (divide and conquer)
  - tabular computation (matrix, cache subsolutions)
  - traceback (pointers to optimal sub-solutions)

- Global Alignment
  - Needleman-Wunsch (1970)

$$S(i,j) = \max \begin{pmatrix} S(i-1,j-1) + \text{match}(i,j), \\ S(i-1,j) + \text{gapPenalty}, \\ S(i,j-1) + \text{gapPenalty} \end{pmatrix}$$

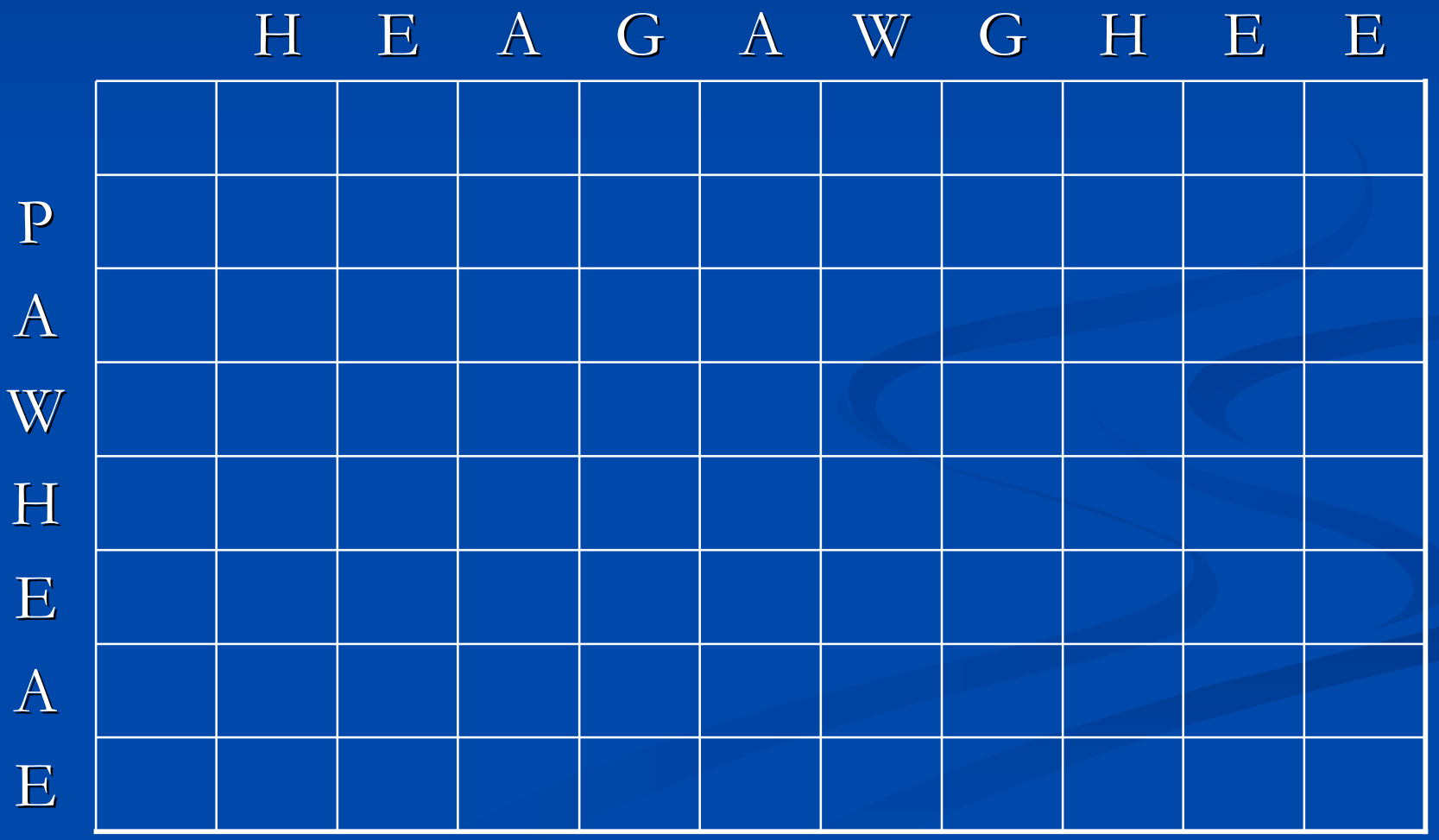
boundary conditions for end gaps:

$$S(i,0) = \text{gapPenalty} * i$$
$$S(0,j) = \text{gapPenalty} * j$$

Aligning two sequences: HEAGAWGHEE and PAWHEAE

Step I. Create a Matrix

example from "Biological Sequence Analysis", Durbin et al.



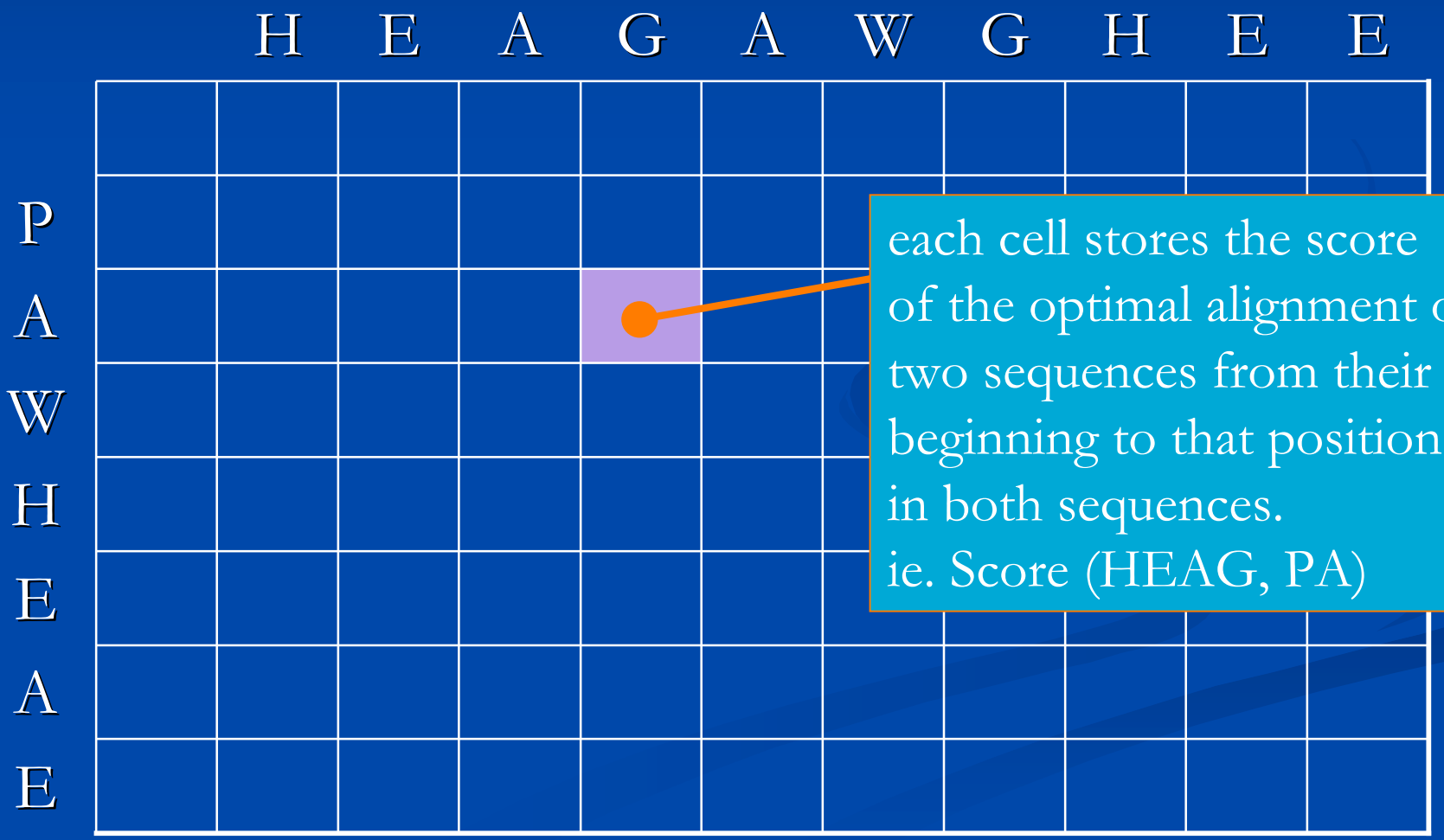
Affine Gap Penalties

affine_gap_penalty = gap_open+gap_extend*gap_length

Aligning two sequences: HEAGAWGHEE and PAWHEAE

Step I. Create a Matrix

example from "Biological Sequence Analysis", Durbin et al.



Step 2: Apply Boundary Conditions

example from Durbin et al.

*using gap = -8

boundary conditions for end gaps:

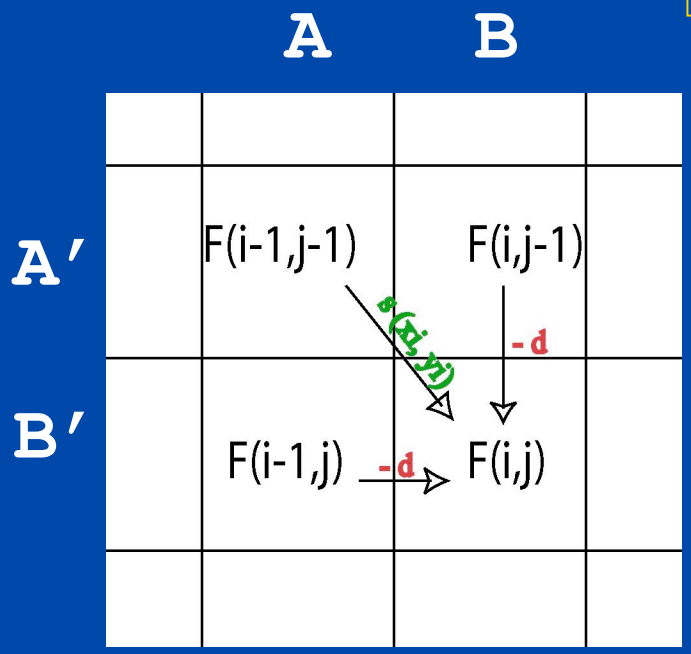
$S(i,0) = gapPenalty * i$

$S(0,j) = gapPenalty * j$

		H	E	A	G	A	W	G	H	E	E
	0	-8	-16	-24	-32	-40	-48	-56	-64	-72	-80
P	-8										
A	-16										
W	-24										
H	-32										
E	-40										
A	-48										
E	-56										

III: Score the Matrix

S(i,j) = max (
S(i-1,j-1) + match(i,j),
S(i-1,j) + gapPenalty,
S(i,j-1) + gapPenalty
)



Three choices for (optimal (...B,...B')):

- **(B,B') + optimal (...A,...A')**
- **(B,-) + optimal (...A,...B')**
- **(-,B') + optimal (...B,...A')**

*above, F(i,j) == S(i,j), I'm reusing an old figure.

Scoring Matrix: S(i, j)

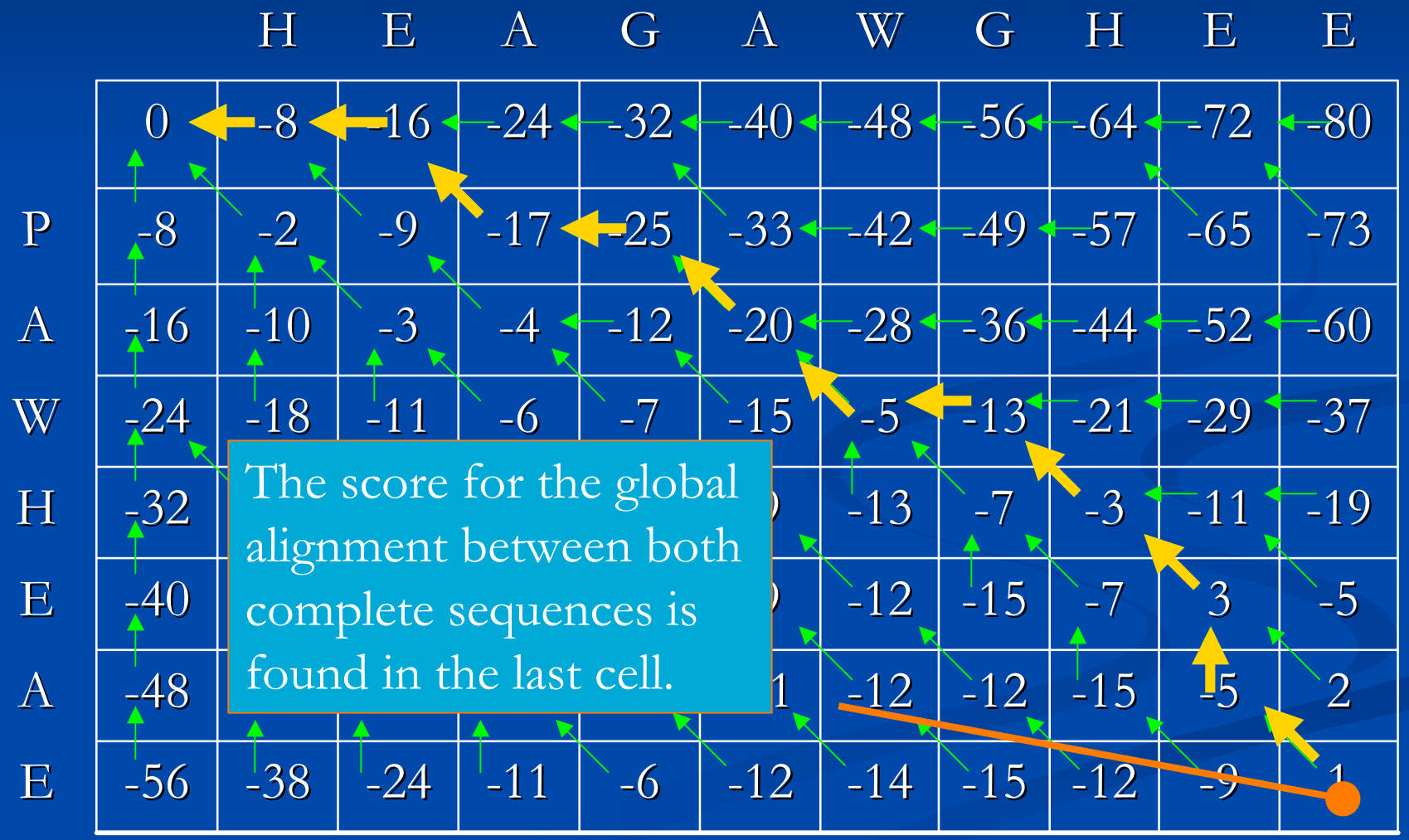
	C	S	T	P	A	G	N	D	E	Q	H	R	K	M	I	L	V	F	Y	W
C	**9**	-1	-1	-3	0	-3	-3	-3	-4	-3	-3	-3	-3	-1	-1	-1	-1	-2	-2	-2
S	-1	**4**	1	-1	1	0	1	0	0	0	-1	-1	0	-1	-2	-2	-2	-2	-2	-3
T	-1	1	**4**	1	-1	1	0	1	0	0	0	-1	0	-1	-2	-2	-2	-2	-2	-3
P	-3	-1	1	**7**	-1	-2	-1	-1	-1	-1	-2	-2	-1	-2	-3	-3	-2	-4	-3	-4
A	0	1	-1	-1	**4**	0	-1	-2	-1	-1	-2	-1	-1	-1	-1	-1	-2	-2	-2	-3
G	-3	0	1	-2	0	**6**	-2	-1	-2	-2	-2	-2	-2	-3	-4	-4	0	-3	-3	-2
N	-3	1	0	-2	-2	0	**6**	1	0	0	-1	0	0	-2	-3	-3	-3	-3	-2	-4
D	-3	0	1	-1	-2	-1	1	**6**	2	0	-1	-2	-1	-3	-3	-4	-3	-3	-3	-4
E	-4	0	0	-1	-1	-2	0	2	**5**	2	0	0	1	-2	-3	-3	-3	-3	-2	-3
Q	-3	0	0	-1	-1	-2	0	0	2	**5**	0	1	1	0	-3	-2	-2	-3	-1	-2
H	-3	-1	0	-2	-2	-2	1	1	0	0	**8**	0	-1	-2	-3	-3	-2	-1	2	-2
R	-3	-1	-1	-2	-1	-2	0	-2	0	1	0	**5**	2	-1	-3	-2	-3	-3	-2	-3
K	-3	0	0	-1	-1	-2	0	-1	1	1	-1	2	**5**	-1	-3	-2	-3	-3	-2	-3
M	-1	-1	-1	-2	-1	-3	-2	-3	-2	0	-2	-1	-1	**5**	1	2	-2	0	-1	-1
I	-1	-2	-2	-3	-1	-4	-3	-3	-3	-3	-3	-3	-3	1	**4**	2	1	0	-1	-3
L	-1	-2	-2	-3	-1	-4	-3	-4	-3	-2	-3	-2	-2	2	2	**4**	3	0	-1	-2
V	-1	-2	-2	-2	0	-3	-3	-3	-2	-2	-3	-3	-2	1	3	1	**4**	-1	-1	-3
F	-2	-2	-2	-4	-2	-3	-3	-3	-3	-3	-1	-3	-3	0	0	0	-1	**6**	3	1
Y	-2	-2	-2	-3	-2	-3	-2	-3	-2	-1	2	-2	-2	-1	-1	-1	-1	3	**7**	2
W	-2	-3	-3	-4	-3	-2	-4	-4	-3	-2	-2	-3	-3	-1	-3	-2	-3	1	2	**11**

Step III: Score Grid

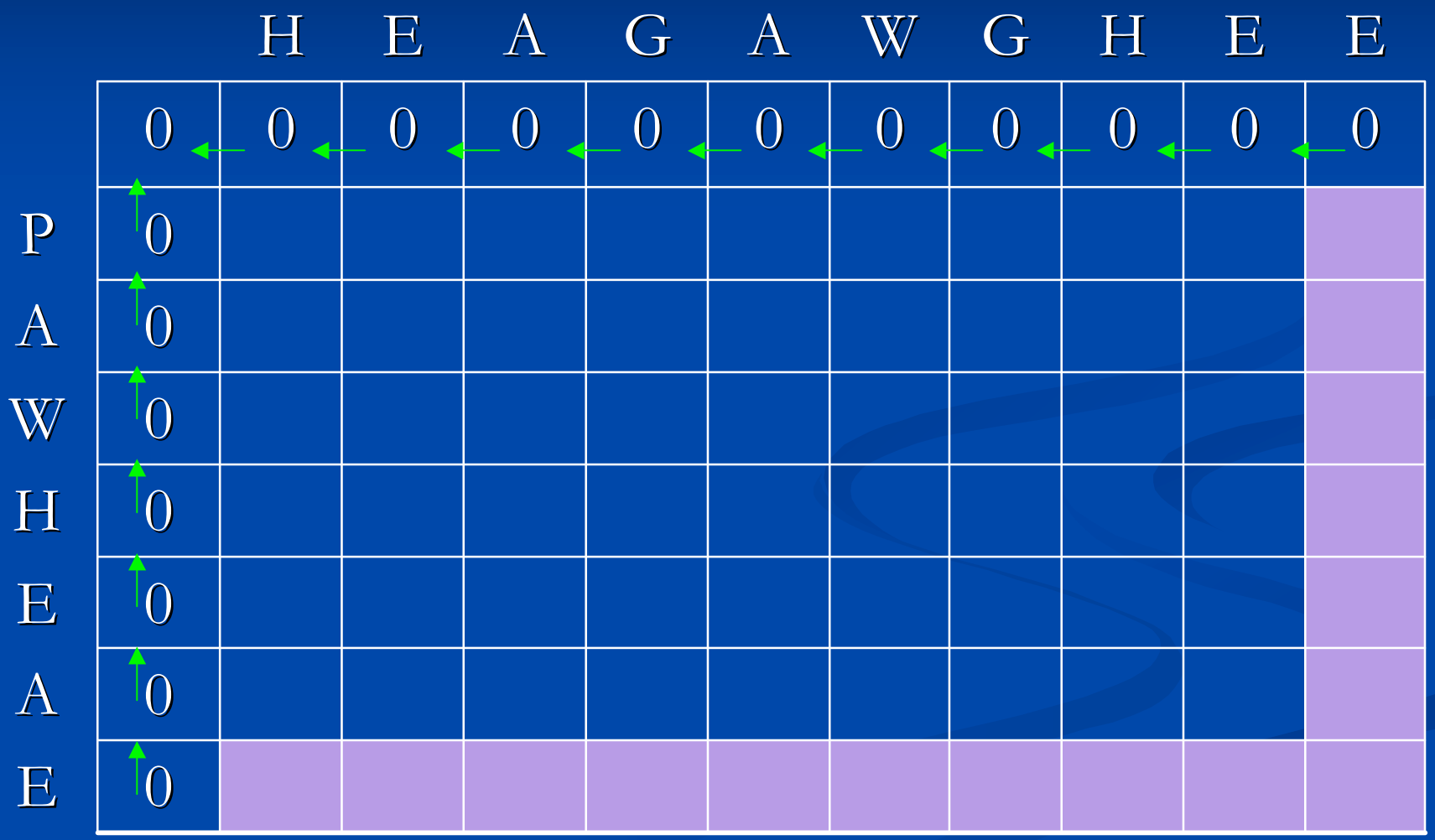
	H	E	A	G	A	W	G	H	E	E
	0	-8	-16	-24	-32	-40	-48	-56	-64	-72	-80
P	-8	-2	-9	-17	-25	-33	-42	-49	-57	-65	-73
A	-16	-10	-3	-4	-12	-20	-28	-36	-44	-52	-60
W	-24	-18	-11	-6	-7	-15	-5	-13	-21	-29	-37
H	-32	-14	-18	-13	-8	-9	-13	-7	-3	-11	-19
E	-40	-22	-8	-16	-16	-9	-12	-15	-7	3	-5
A	-48	-30	-16	-3	-11	-11	-12	-12	-15	-5	2
E	-56	-38	-24	-11	-6	-12	-14	-15	-12	-9	1

Final Step IV: Traceback

H E A G A W G H E – E
– – P – A W – H E A E



A Variation w/ Global Alignment: Remove End-gap penalties



* begin traceback from highest scoring cell in farthest row or column

Local Alignment

- Smith-Waterman (1981)

$$S(i,j) = \max\left(\begin{array}{l} S(i-1,j-1) + \text{match}(i,j), \\ S(i-1,j) + \text{gapPenalty}, \\ S(i,j-1) + \text{gapPenalty}, \\ 0 \end{array}\right)$$

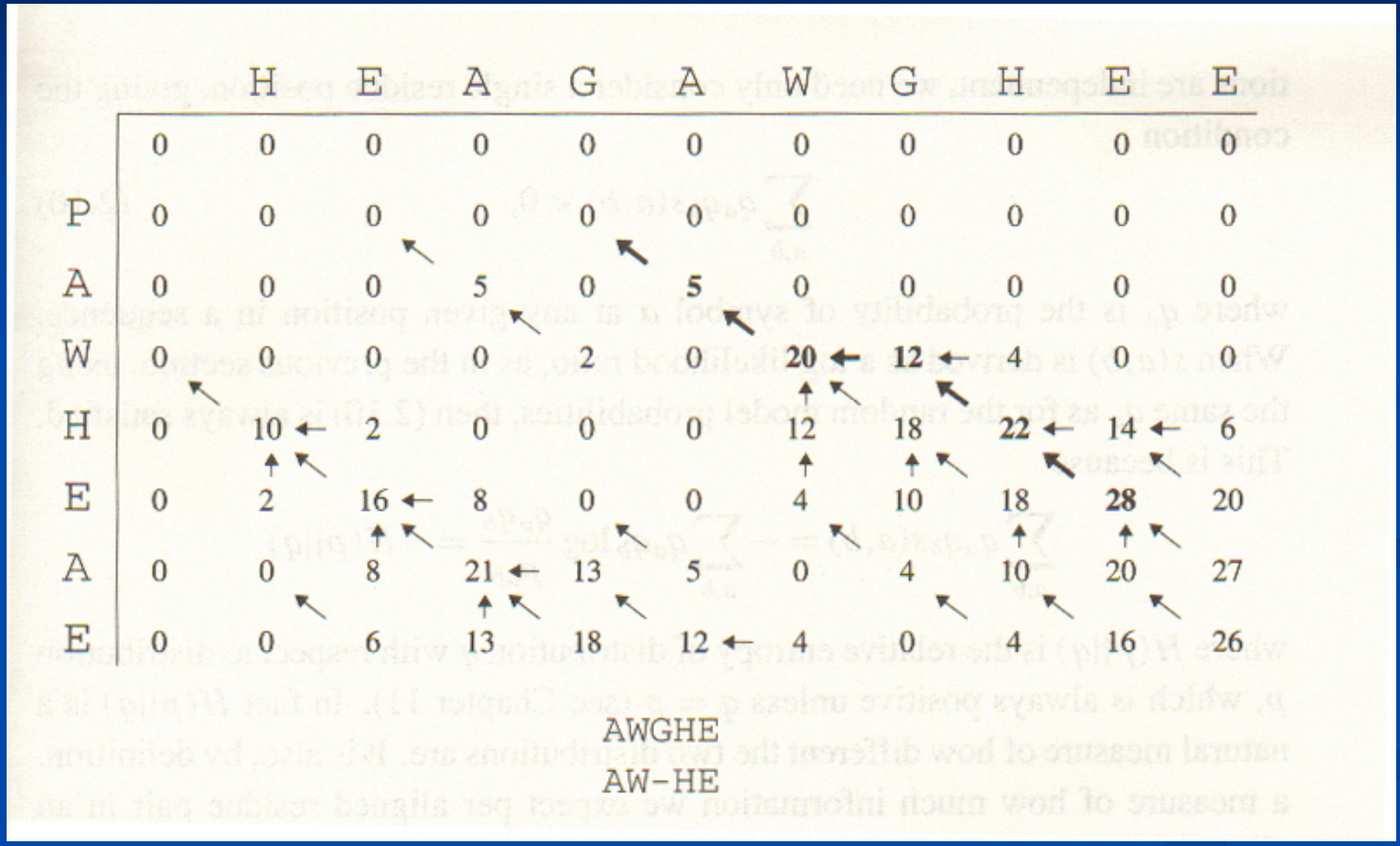
boundary conditions for end gaps:

$$S(i,0) = 0$$

$$S(0,j) = 0$$

* simpler version shown here, allowing for only single indel events at each position

Local Alignment



		H	E	A	G	A	W	G	H	E	E
	0	0	0	0	0	0	0	0	0	0	0
P	0	0	0	0	0	0	0	0	0	0	0
A	0	0	0	5	0	**5**	0	0	0	0	0
W	0	0	0	0	2	0	**20**	**12**	4	0	0
H	0	10	2	0	0	0	12	18	**22**	14	6
E	0	2	16	8	0	0	4	10	18	**28**	20
A	0	0	8	21	13	5	0	4	10	20	27
E	0	0	6	13	18	12	4	0	4	16	26

AWGHE
AW-HE

Begin traceback at the highest score, anywere in the matrix,
Stop when (0) encountered.

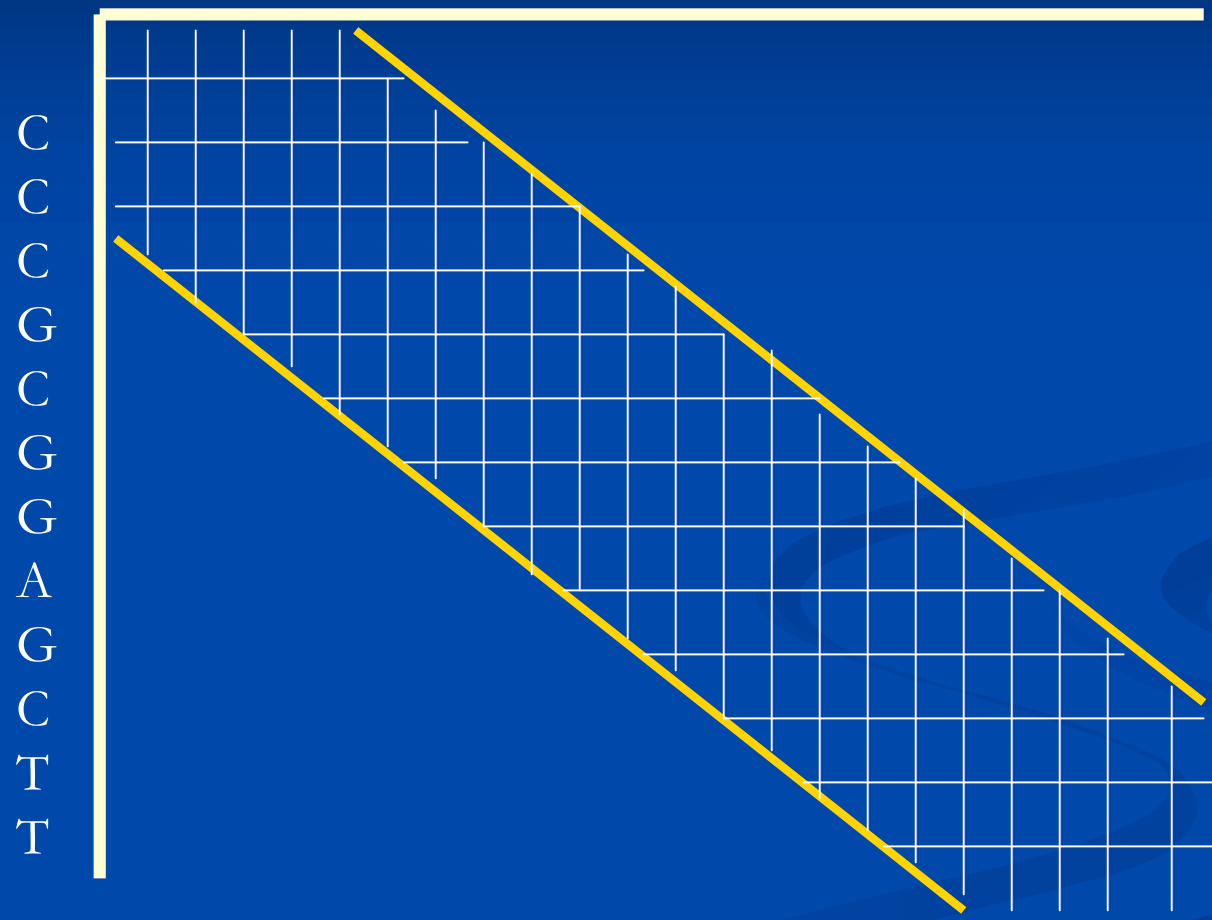
Biological Sequence Analysis, by R. Durbin, S. Eddy, A. Krogh, and G. Mitchison

Waterman-Eggert

- Extend Smith-Waterman local alignment to find all non-intersecting local alignments of sufficient similarity. (*Not Just the Single Best Alignment*)
- Very useful when alternative alignments exist between sequences, including repetitive domains.
- Algorithm:
  - Find best local alignment
  - Set S(i,j) = 0 for all matches in best alignment
  - Recalculate matrix
  - Repeat procedure until the best alignment score is below some minimum threshold.

Banded Alignments

GATCGGGCCCGCGGAGCTTTGCTGGGCT



*accelerate alignment speed by limiting computation to a diagonal band.
You must be confident that your best alignment is within this band, though.

Performance

- Typically:
  - memory: n*m
  - speed: n*m
- Can achieve linear memory usage:
  - can calculate a DP-matrix score given the current row and the previous row.
  - Therefore, can calculate the maximum alignment score using linear memory, keeping only two rows in memory at any time.
  - Problem is the traceback.
    - divide and conquer, ie. quicksort algorithm.

1988: memory: m, speed: n*m

(Myers & Miller, 1988, CABIOS)

Programs Implementing DP Alignment Algorithms
(Pearson's Fasta2 Tool Suite)

- Global Alignment
  - align (Needleman-Wunsch, but linear space)
  - align0 (variation without end-gap penalties)
- Local Alignment
  - ssearch (Smith Waterman)
  - lalign (Waterman-Eggert, but linear space [Huang & Miller,1991])

Visit: ftp://ftp.virginia.edu/pub/fasta

II. Seed and Extend

- Programs:
  - Pearson's FASTP (1985) and FASTA (1988)
  - BLAST (Altschul et al., 1990)
  - Gapped BLAST (Altschul et al., 1997)
  - BLAT (Kent, 2002)
  - many, many more. This heuristic is commonly employed.

Seed and Extend

- Given 2 sequences (X,Y):
  - index all words of fixed length in sequence X
  - examine all words in Y for occurrence in X
  - locally extend word matches to HSP (high scoring segment pair)
- After HSP identification:
  - cluster/group HSPs (order, orientation)
  - use DP to fill in gaps.

Word Comparisons Using Bytes

- Sequence compression, byte-comparisons:
  - A = 00
  - C = 01
  - G = 10
  - T = 11
- GACT = 10000111
- Fast byte comparisons to identify word matches

- Also, put entire database in memory
- ** Orders of magnitude faster than DP.

BLAST: Basic Local Alignment Search Tool

1990, Altschul et al.

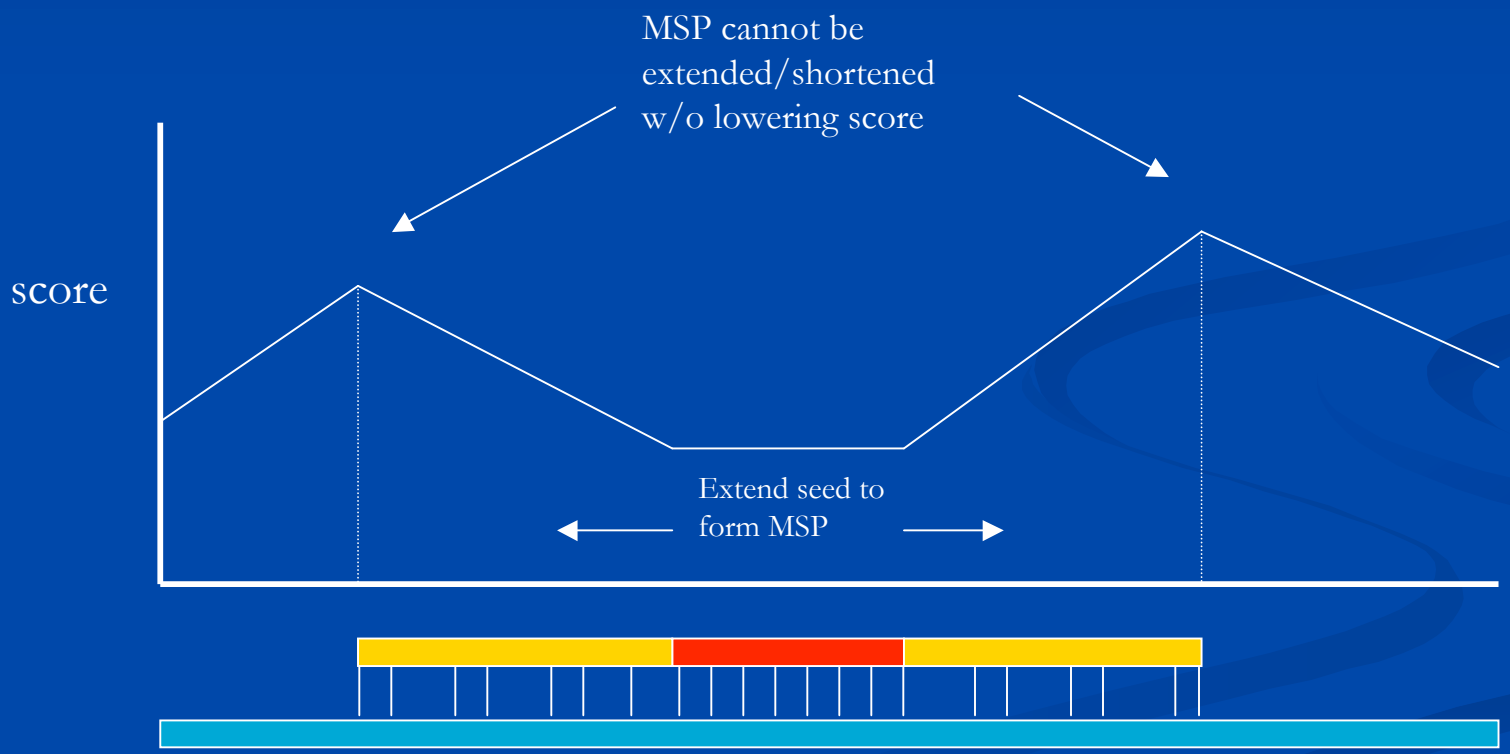
- MSP:
  - maximal segment pair from 2 sequences, identical length and highest score.
  - Locally maximal: score cannot be improved by lengthening or shortening the segment pair.

BLAST: Basic Local Alignment Search Tool

1990, Altschul et al.

- Find homologous sequence pairs
  - Defined as an MSP with a score of at least S which contains a word match with score at least T
- Algorithm:
  - Build word index for query
    query size n, word size w = (n-w+1) words
  - Scan database sequence, find high scoring word matches, extend word matches to find MSPs.

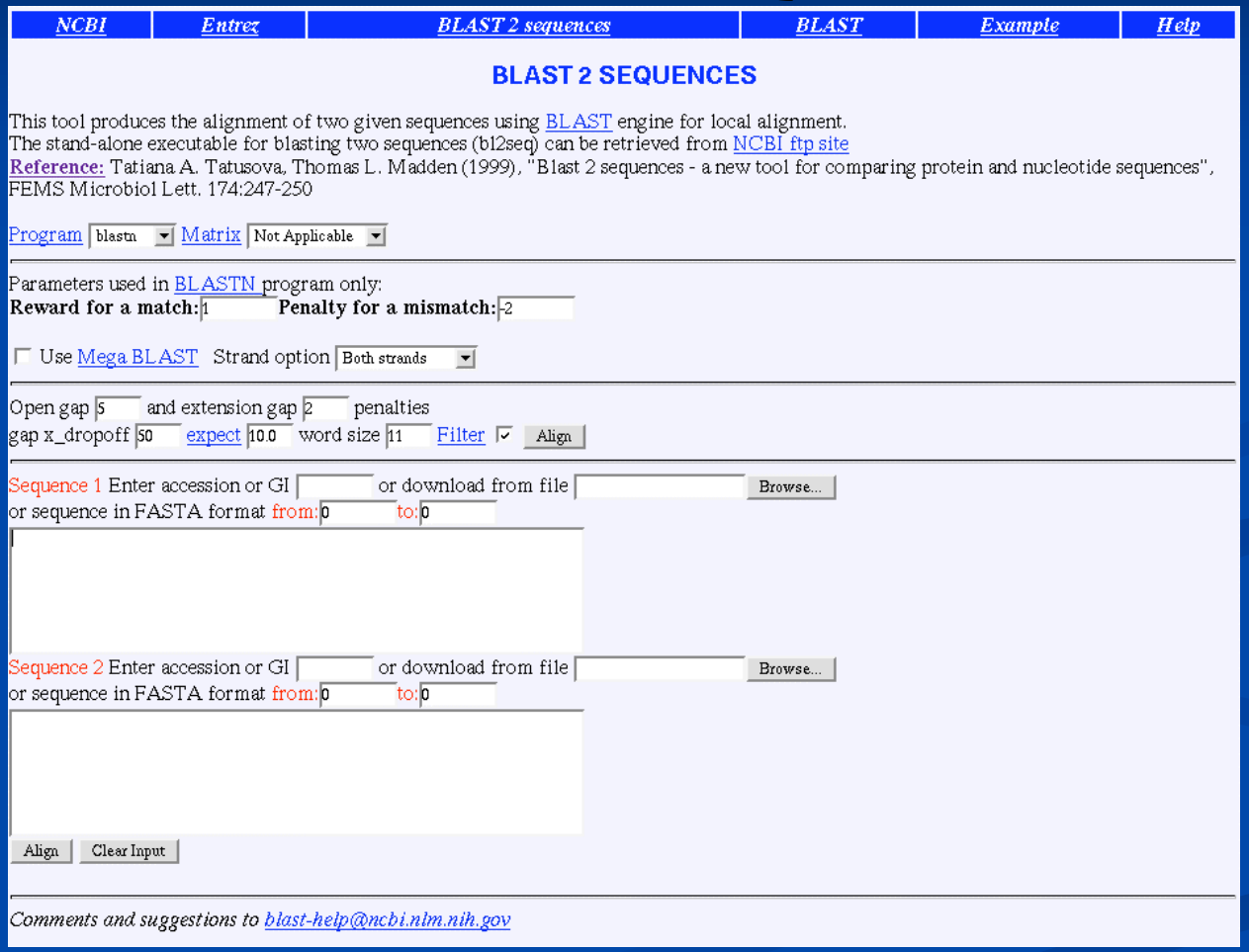
From Seed to MSP



Gapped Blast (1997, Altschul et al.)

- Major refinements:
  - Speed improvement: Require two non-overlapping word hits within distance *A* before extending to form an MSP.
  - Produces gapped alignments using DP (restricted to a band in the DP path graph).

BLAST for Pairwise Alignment
BL2SEQ



NCBI | Entrez | BLAST 2 sequences | BLAST | Example | Help

BLAST 2 SEQUENCES

This tool produces the alignment of two given sequences using BLAST engine for local alignment.
The stand-alone executable for blasting two sequences (bl2seq) can be retrieved from NCBI ftp site
Reference: Tatiana A. Tatusova, Thomas L. Madden (1999), "Blast 2 sequences - a new tool for comparing protein and nucleotide sequences", FEMS Microbiol Lett. 174:247-250

Program blastn Matrix Not Applicable

Parameters used in BLASTN program only:
Reward for a match: 1 **Penalty for a mismatch:** -2

Use Mega BLAST Strand option Both strands

Open gap 5 and extension gap 2 penalties
gap x_dropoff 50 expect 10.0 word size 11 Filter Align

Sequence 1 Enter accession or GI or download from file Browse...
or sequence in FASTA format from: 0 to: 0

Sequence 2 Enter accession or GI or download from file Browse...
or sequence in FASTA format from: 0 to: 0

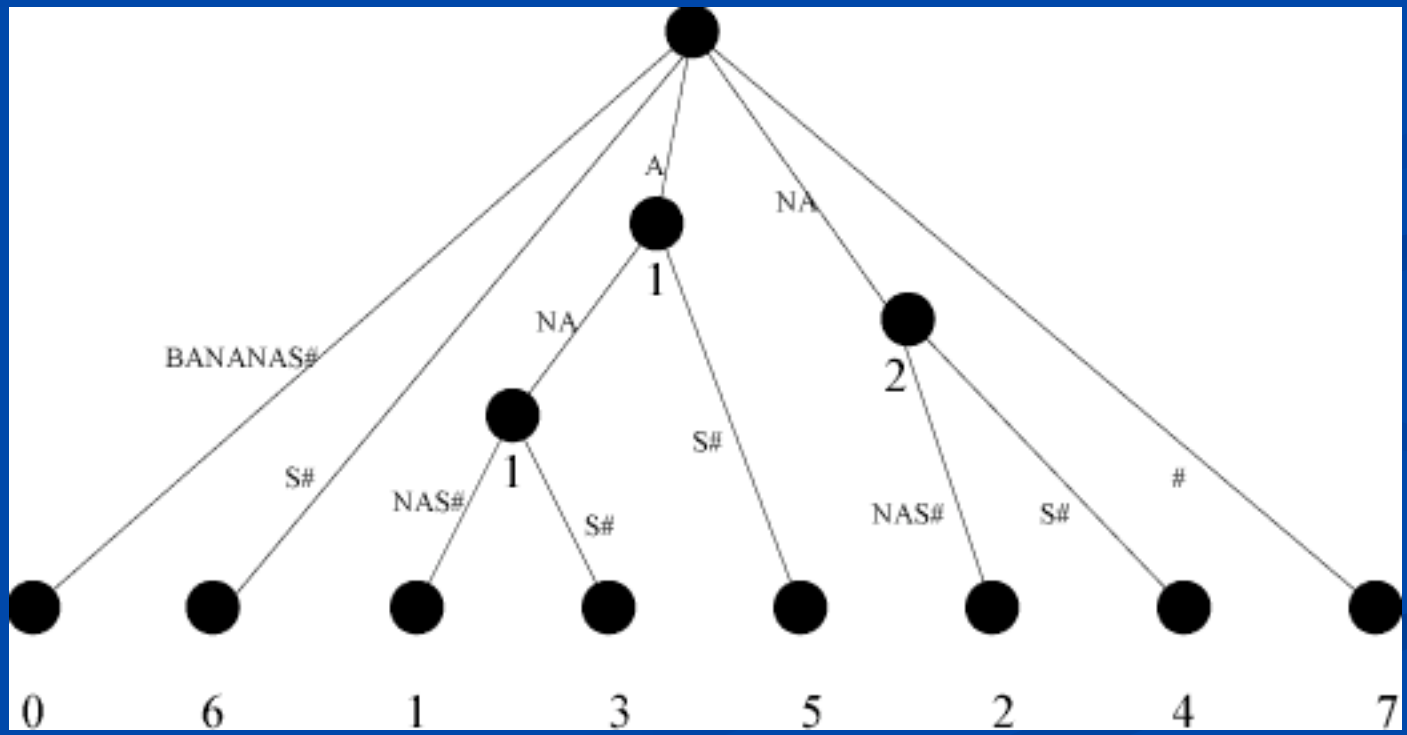
Align Clear Input

Comments and suggestions to blast-help@ncbi.nlm.nih.gov

http://www.ncbi.nlm.nih.gov/blast/bl2seq/bl2.html

III. Suffix Trees applied to Genome Alignments

- MUMmer, AVID
- linear time performance (linear construction, linear search)
- example for 'BANANAS'

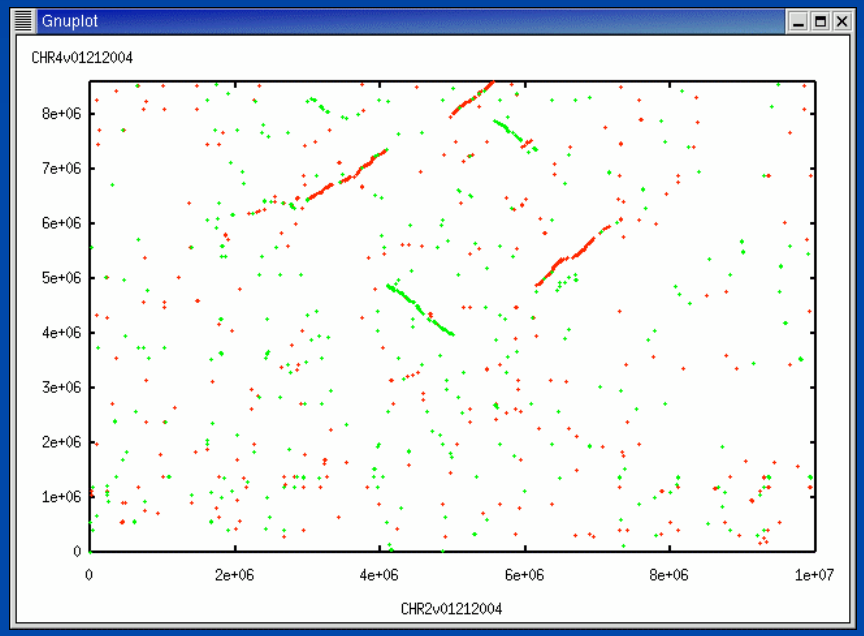


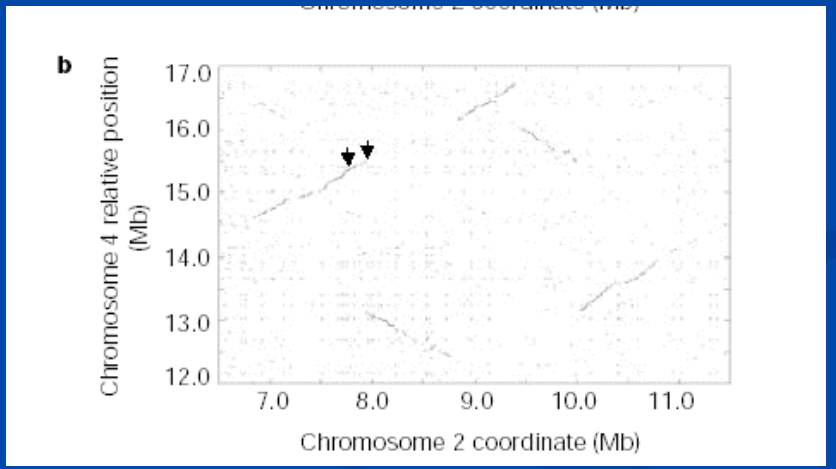
MUMmer

- MUM: Maximal Unique Match
  - Similar to MSP, but unique and identical in sequence
- Includes Nucmer (nucleotide mummer) and Promer (protein mummer) packages built on MUMmer.
- Algorithm:
  - Find all exact matches of at least length *L*
  - Cluster and join matches in close proximity
  - Maximum-length colinear chain of matches is extracted
  - DP to align gapped regions between matches and chain boundaries.

MUMmer

(Promer applied to genome duplications in Arabidopsis)

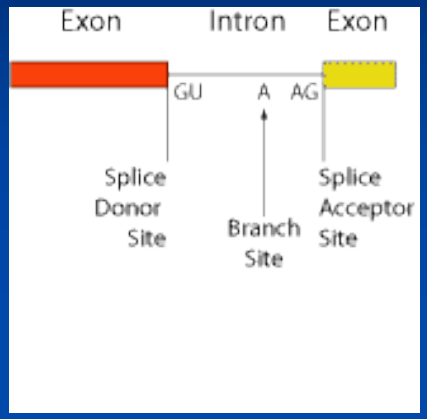


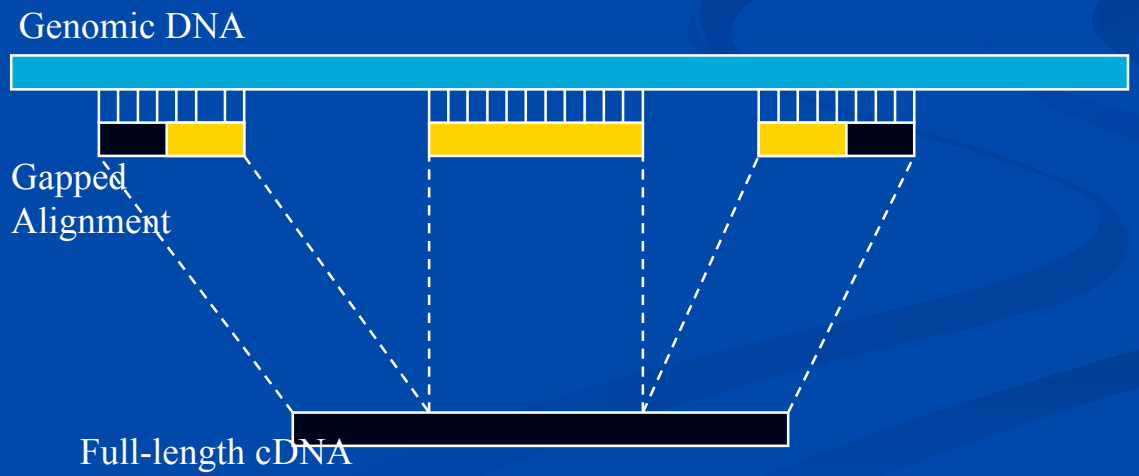


NATURE | VOL 402 | 16 DECEMBER 1999 | www.nature.com

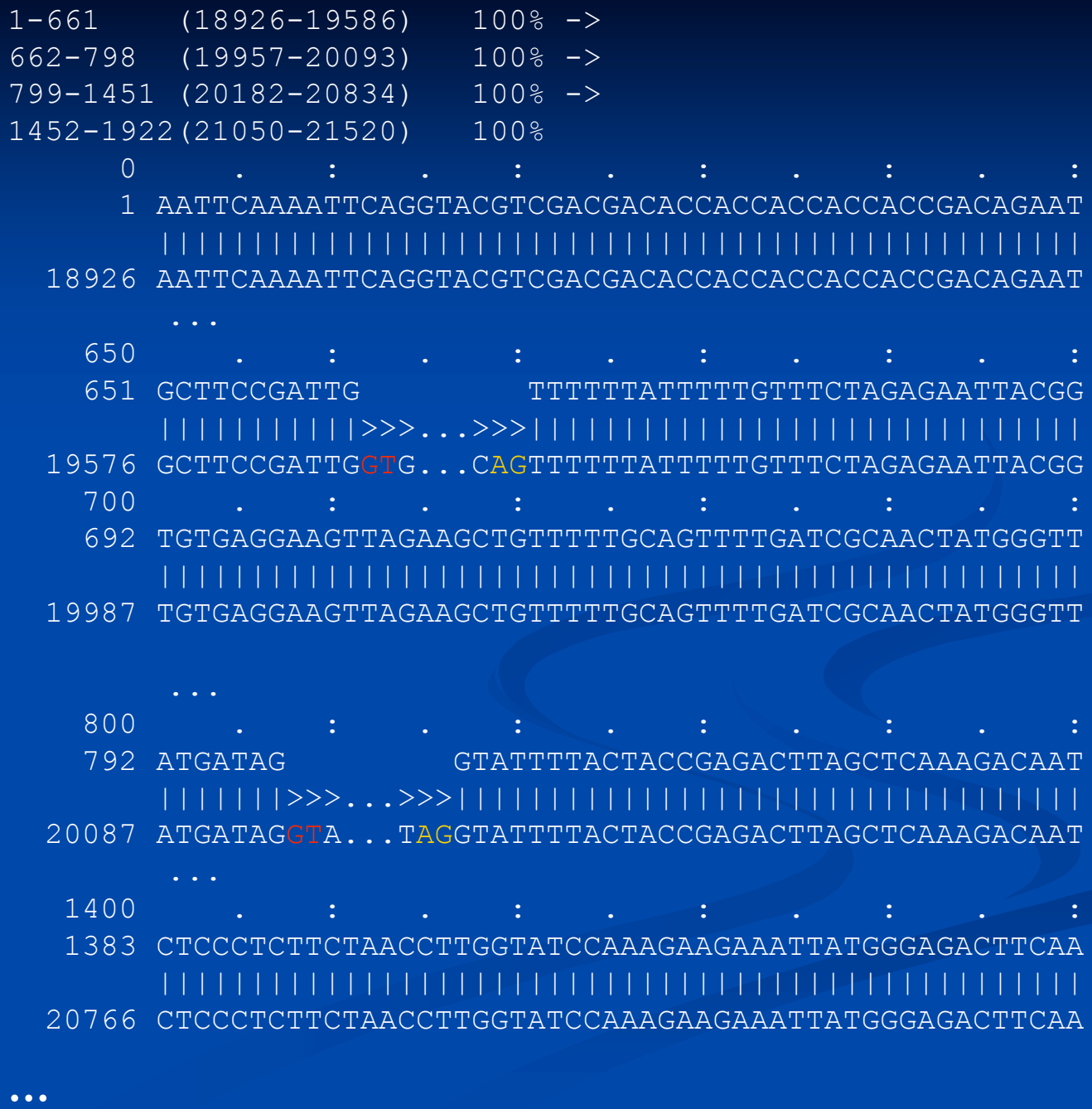
IV. Spliced Alignment

- AAT (1997)
- sim4 (1998)
- GeneSeqer (2000)
- BLAT (2002)





Sim4
Florea
(1998)



...

AAT (Huang, 1997)

- Protein and EST alignments, combine results

```
Query+                  58683 AATGCAGCAGGTAAATTTTTGTCATCTTTTTTCAAAACTCCTGCAACAATAACCATAAAC
                              :::::::::.----------------------------------------------------
SP|O04003|LG1_MAIZE       230   C  S  R
12071.+                   341 AATGCAGTAG
                              ||||||||||----------------------------------------------------
arab|TC185099 Squamo+     409 AATGCAGCAG
```

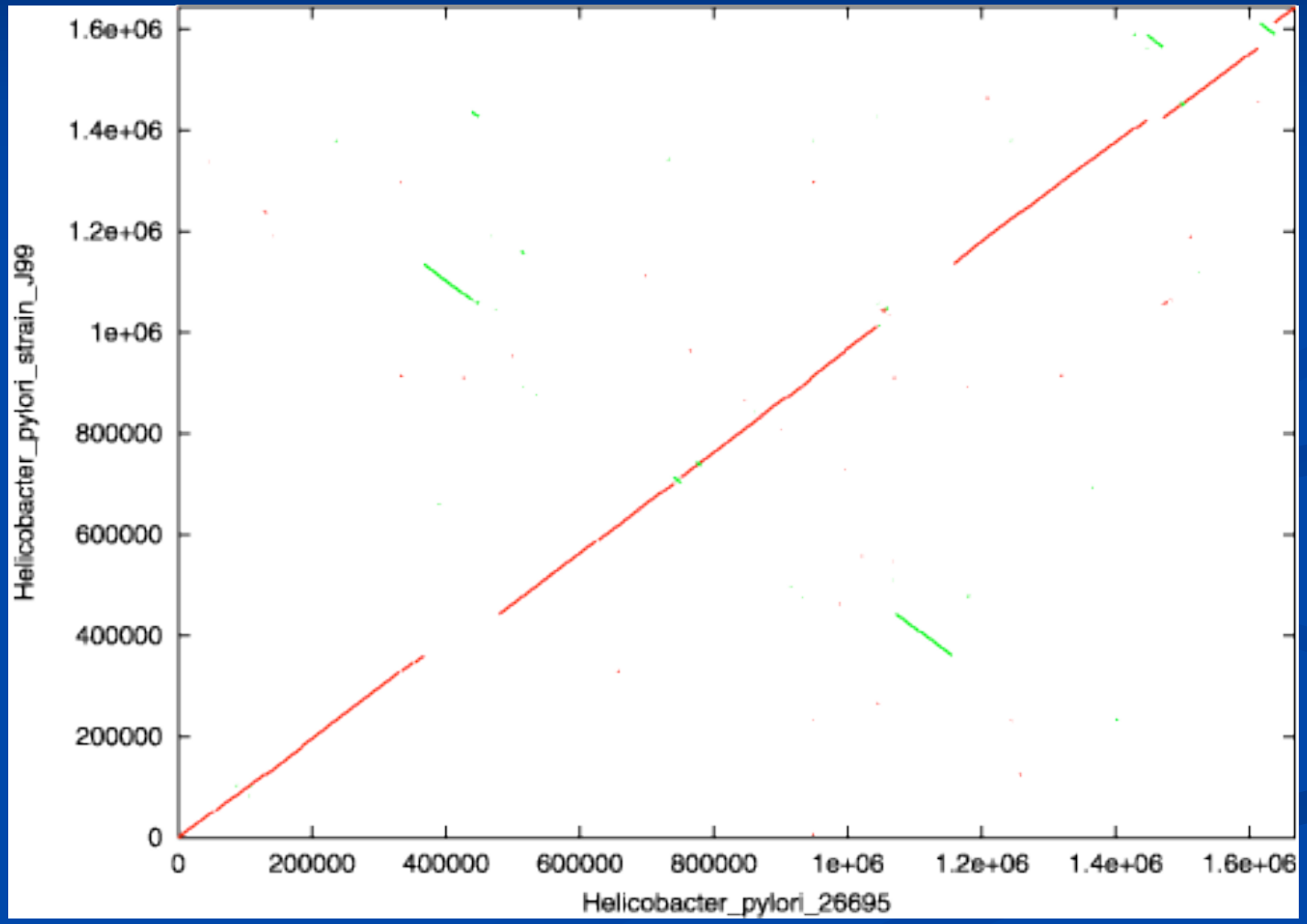
```
Query+                  58743 AAGAACAAGAAATATGAGACCTTTTACCTTTTGTTTTTAAGGTTTCACGCGCTCAGTGAG
                              ----------------------------------------.::::::  .:::  .:::
SP|O04003|LG1_MAIZE       233                                          F  H  L  L  D  E
                              ----------------------------------------||||||||||||||||||
arab|TC185099 Squamo+     419                                        GTTTCACGCGCTCAGTGAG
```

ftp://ftp.tigr.org/pub/software/AAT

IV. Alignment Visualization Tools

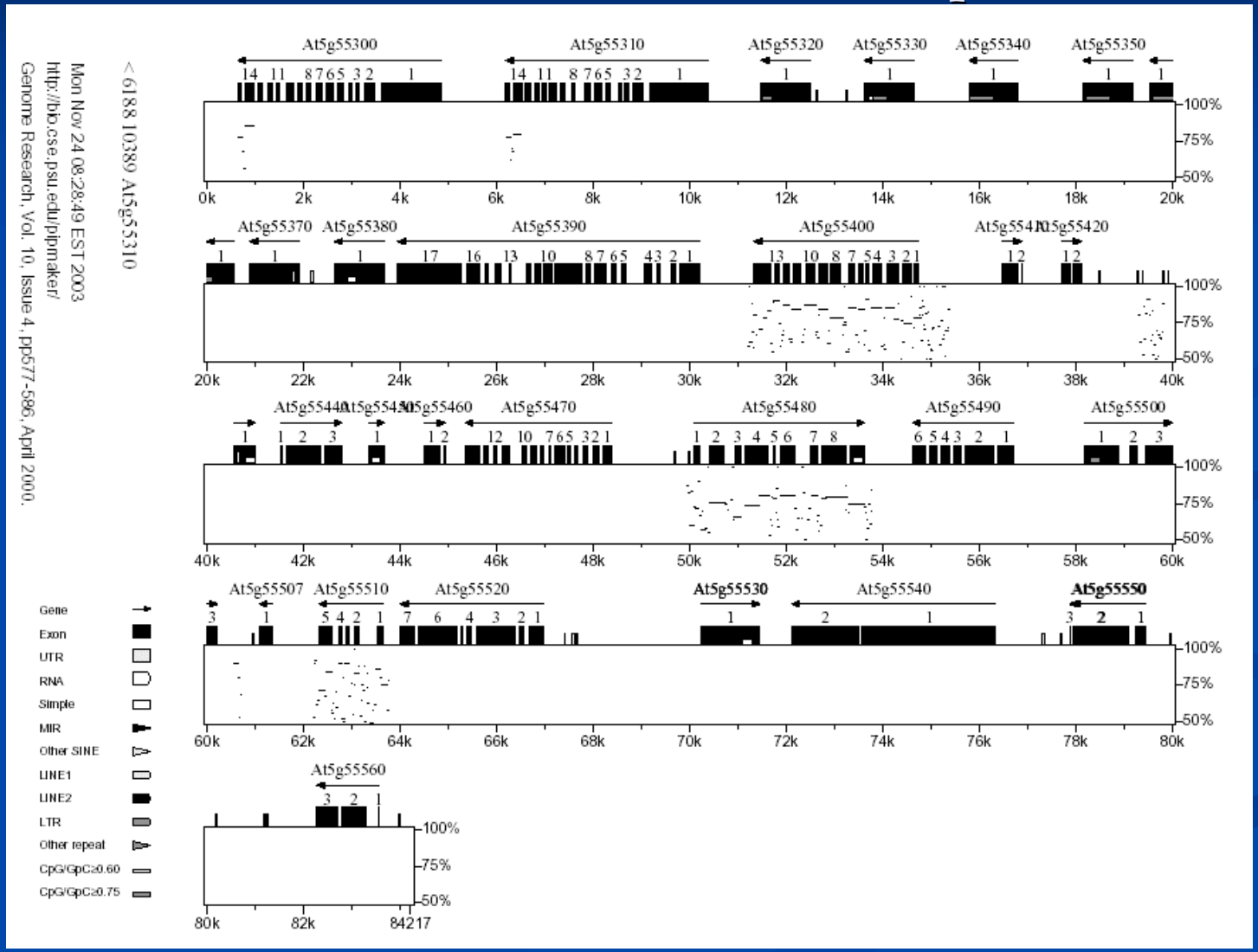
- XY-plot using Gnuplot
- PIPmaker
- VISTA

XY-plot of a MUMmer Alignment



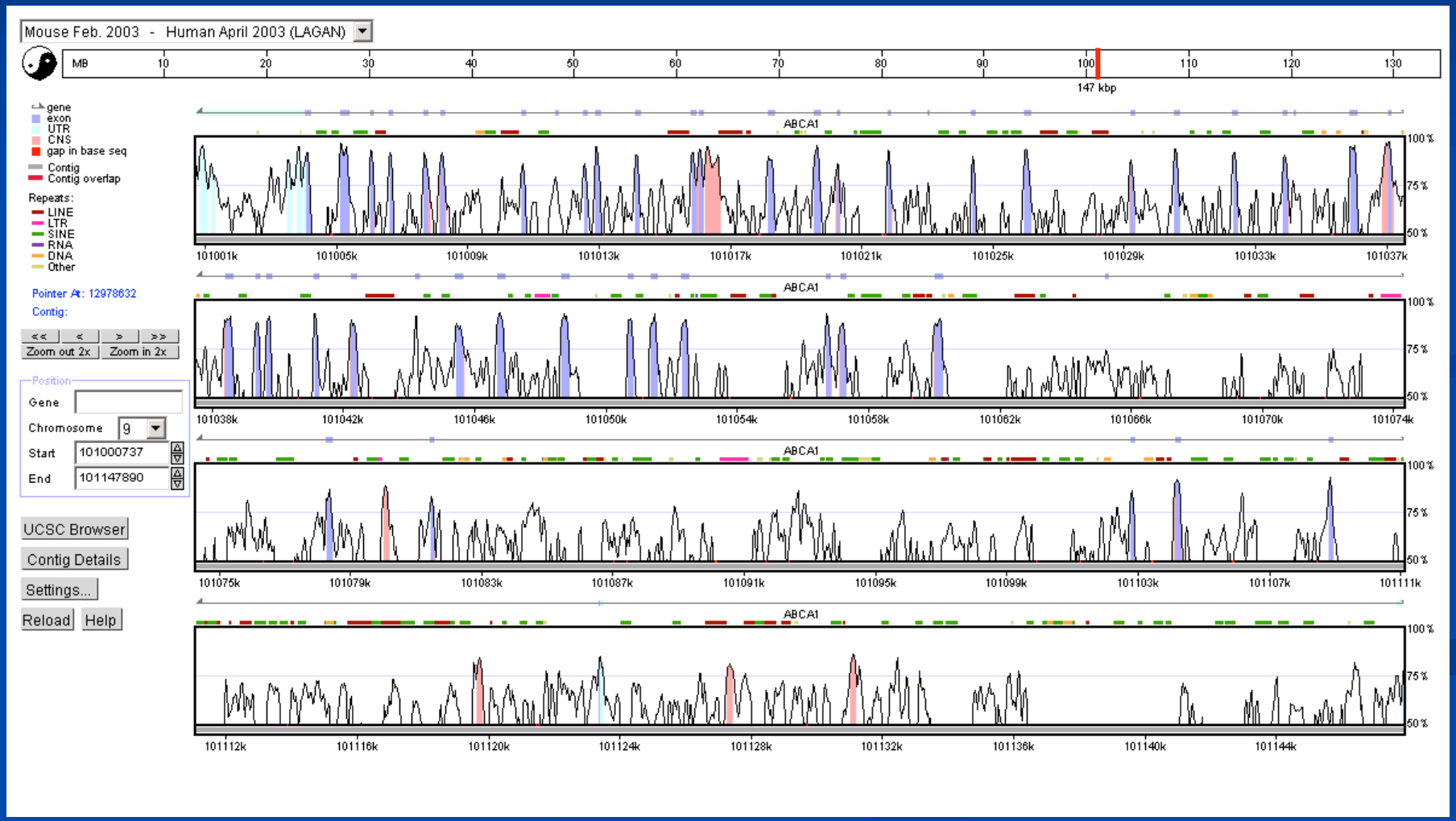
PIPmaker

Chromosome 4 vs. 5 of Arabidopsis



VISTA

Human vs. Mouse



Sequence Alignment Resources

- Books:
  - Biological sequence analysis, by Durbin,Eddy, Krogh, and Mitchison
  - Introduction to Computational Molecular Biology, by Setubal and Meidanis
  - An Introduction to Bioinformatics Algorithms, by Jones and Pevzner
  - Computational Molecular Biology, by Clote and Backofen
  - Bioinformatics: Sequence and Genome Analysis, by David Mount

Sequence Alignment Resources

- Software:
  - FASTA
    - http://fasta.bioch.virginia.edu/fasta/
  - BLAST
    - http://www.ncbi.nlm.nih.gov/BLAST/
    - http://blast.wustl.edu/blast/README.html
  - AAT (protein & transcript spliced genome alignments)
    - http://www.tigr.org/software/
  - MUMmer
    - http://mummer.sourceforge.net/
  - Numerous resources from Webb Miller's site, including PIPmaker, sim4, blastz, and others:
    - http://www.bx.psu.edu/miller_lab/