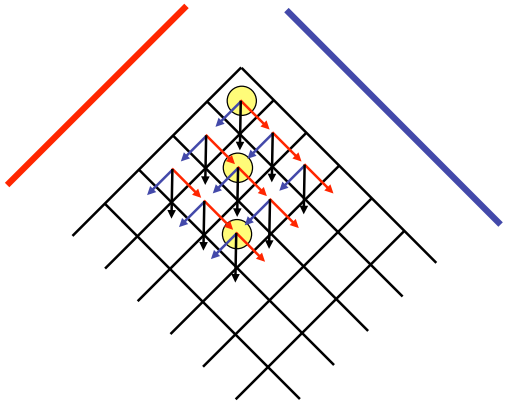
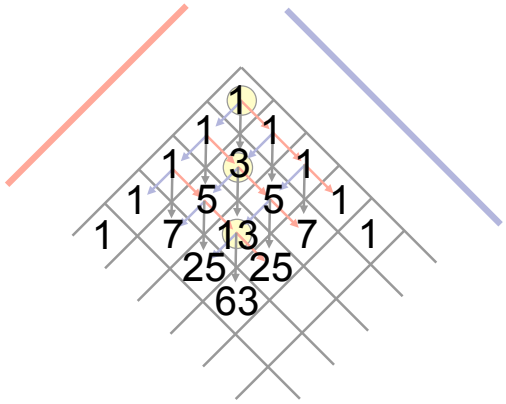
Sequence alignment - part 2

Other dynamic programming applications

Allowing gaps, how many possible alignments are there?



$>10^{74}$ alignments for 100 x 100 residues

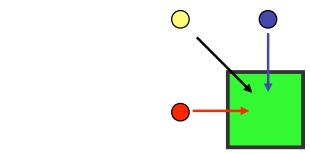


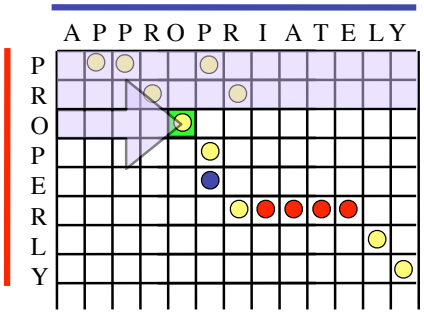
Examine how to produce the following alignment:

A P P R O P - R I A T E L Y

- - P R O P E R - - - - L Y

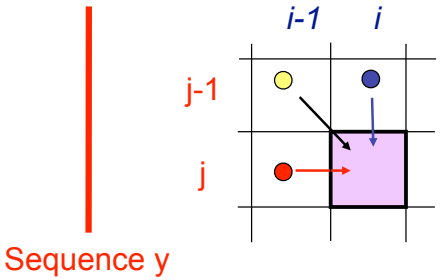
Global alignment algorithm step 1:
Fill out a matrix of scores: each cell maximises its score by examining its three already evaluated neighbours. It inherits one of their scores, and either pays a gap penalty or acquires score from the aligned residues.





Gap in blue sequence Gap in red sequence

Sequence x



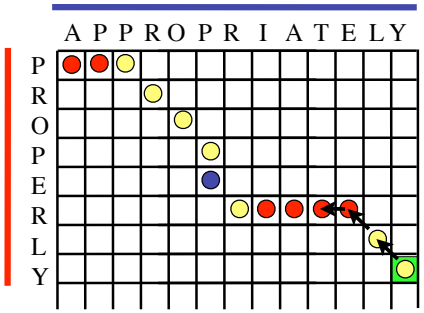
F(*i,j*) is the score of the best alignment of subsequence $X_{1...i}$ and subsequence $y_{1...j}$

Recursive: F(*i,j*) depends on previously evaluated elements of F

$$F(i,j) = \max \begin{cases} F(i-1,j-1) + score(x_i, y_j) \\ F(i, j-1) - d \\ F(i-1, j) - d \end{cases}$$

where d = gap penalty
score() = score from aligning two residues

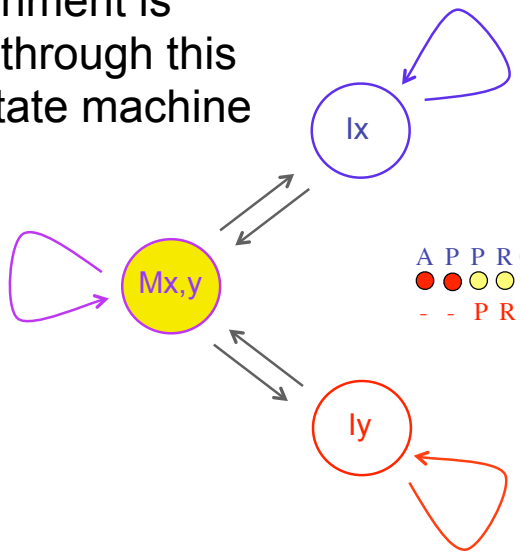
For each cell, a *traceback pointer* records from which parent the best score was inherited

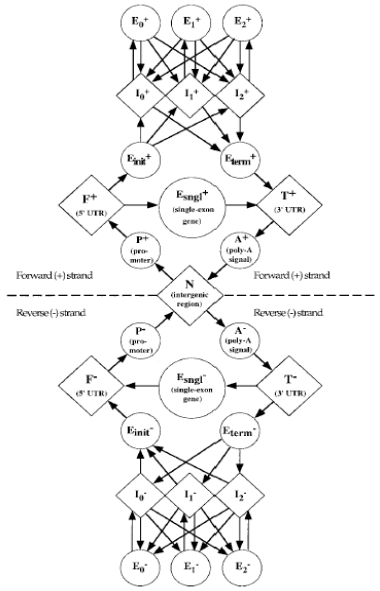


Algorithm step 2:
Traceback: to generate the alignment, iteratively follow the pointers back from the final cell evaluated

- O (M x N) time, memory
- Variants in recursion rules and boundary conditions (see practical):
  - best global alignment
  - best local alignment
  - best overlap between two sequences
  - best set of repeated matches
- Affine and linear gap penalties
- Linear memory variants of the above

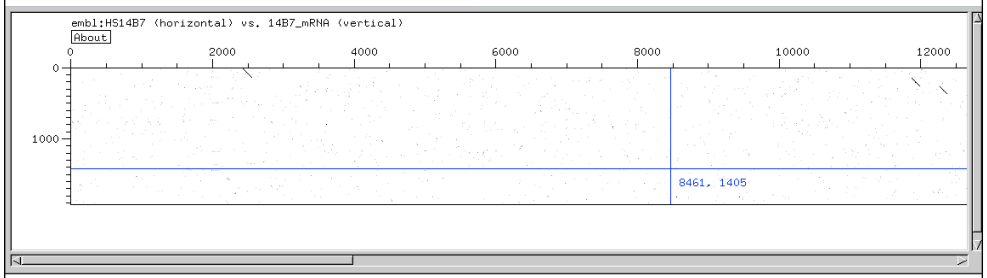
An alignment is a path through this finite state machine





Burge, C. and Karlin, S. (1997)
Prediction of complete gene structures in human genomic DNA. J. Mol. Biol. 268, 78-94

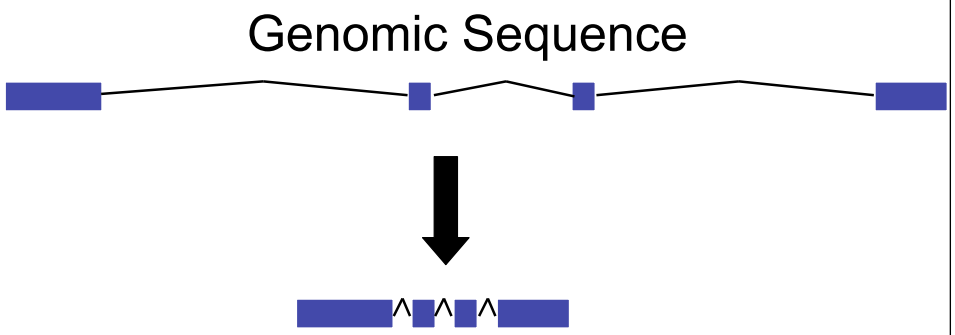
cDNA/ genomic sequence alignment



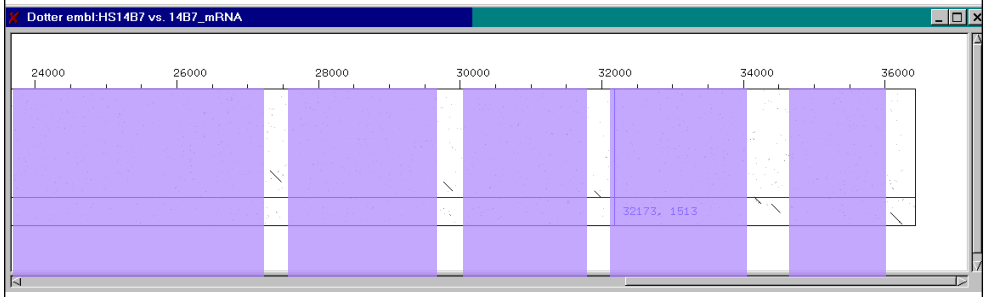
Alignment of a cDNA (vertical) to a genomic sequence (horizontal)

Alignment algorithm allows mismatches, insertions/ deletions, and understands introns.

Efficiency: if can roughly identify exons with low false negative rate, then no need to include most of intron sequence in DP calculation

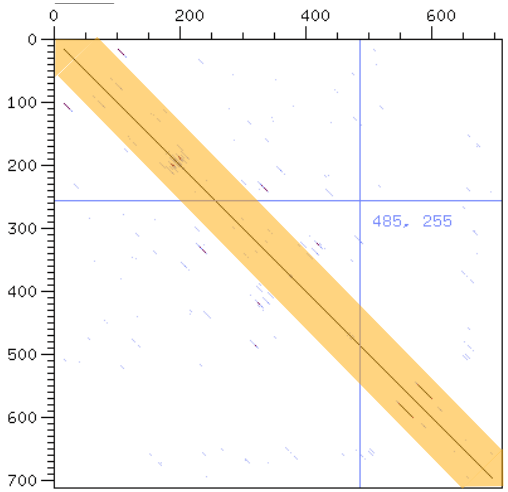


In practice, dynamic programming is used to piece together the exons once they have been roughly located

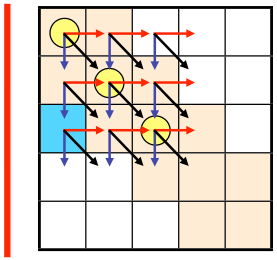


Same approach for *Protein* to genomic sequence alignment. Method must be tolerant of frameshifts, sequence errors, introns: e.g. GeneWise

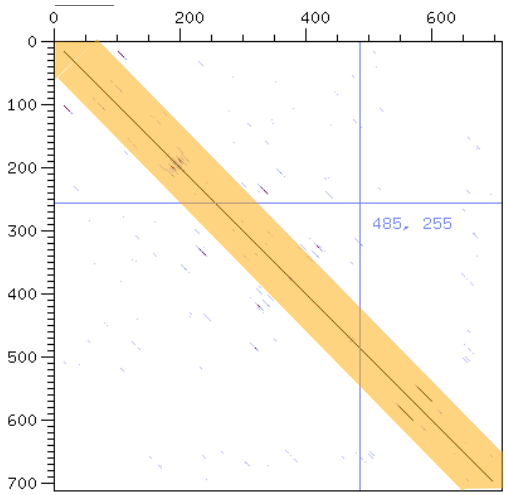
Efficiency: Banded Dynamic Programming avoids need to calculate whole matrix



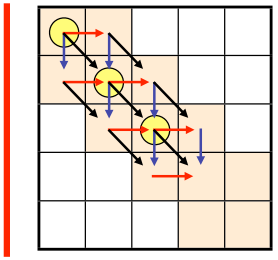
Use e.g. when can place bounds on likely number of gaps



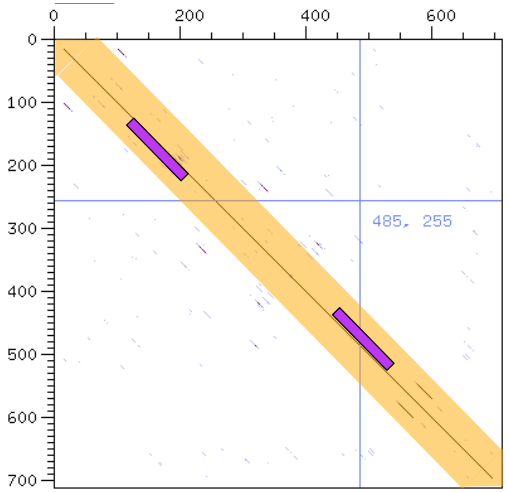
Efficiency: Banded Dynamic Programming avoids need to calculate whole matrix



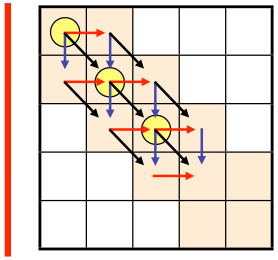
Completely ignore cells outside band



Efficiency: Banded Dynamic Programming avoids need to calculate whole matrix



Use to join/ extend heuristic matches



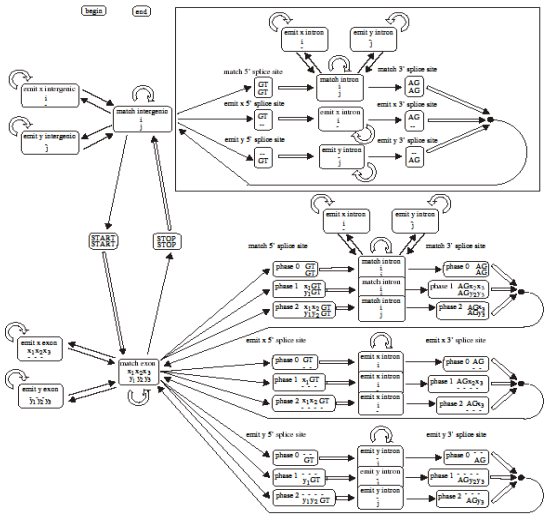
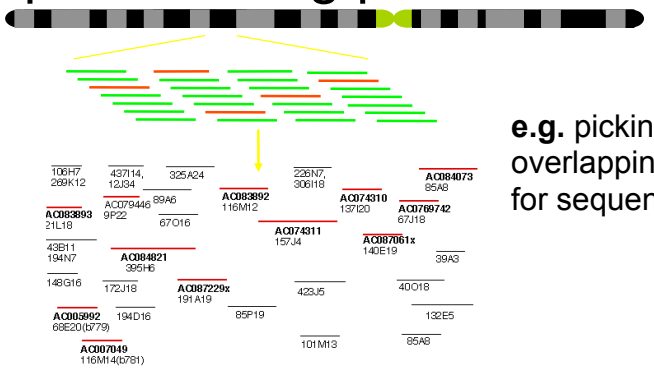


Fig. 2. States and transitions of the pair HMM of DOUBLESCAN. States are shown as boxes with rounded corners, transitions as arrows.

Meyer and Durbin 2002 Bioinformatics *18* 1309-18

Optimum tiling paths: 1



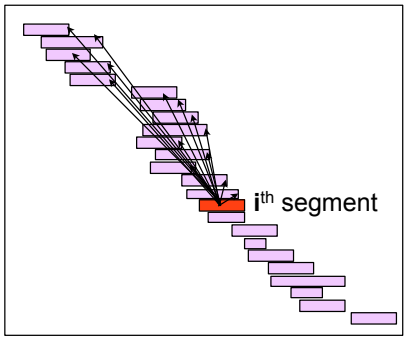
e.g. picking minimally overlapping sets of BACs for sequencing

Other examples of picking tiling paths from a large number of candidate segments:

- overlapping sets of PCR products for screening a region
- design of well-space oligonucleotide probes for microarrays

May want some overlap, minimal overlap, or a specified gap.

Optimum tiling paths: 2



Sweep through segments left to right evaluating:

$$s(i) = \max_{j=1}^{j<i} [s(j) + q(i) - g(i,j)]$$

(*ad hoc*) quality of segment ***i***

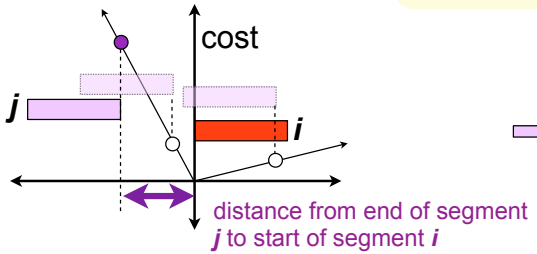
cost of *gap* or *overlap* between segments ***i*** and ***j***

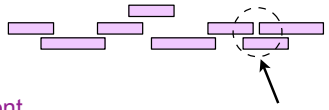
- Each segment ***i*** keeps a **pointer** to the best element ***j***
- **Traceback** from the segment with the highest score

Optimum tiling paths: 3

Influence of the gap/overlap cost function, g(*i,j*)

$$s(i) = \max_{j=1}^{j<i} [s(j) + q(i) - g(i,j)]$$





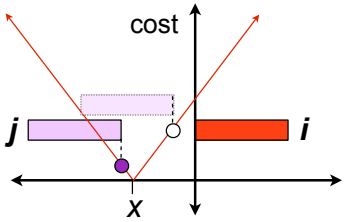
Favours using large overlap to cover small gap

- Segments that **abut** have no penalty
- The greater the **overlap** the greater the penalty
- The greater the **gap** the greater the penalty
- The rate of growth of the overlap cost is less than that of the gap cost so a tiling path with overlaps will be favoured over one with gaps

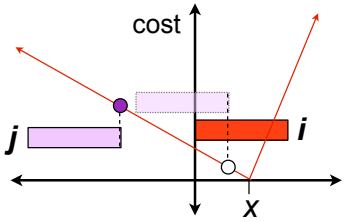
Optimum tiling paths: 4

Influence of g(*i,j*)

$$s(i) = \max_{j=1}^{j<i} [s(j) + q(i) - g(i,j)]$$

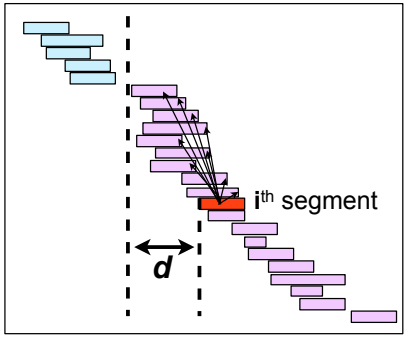


Prefer a gap of length *x*, and penalise shorter and longer gaps equally



Prefer an overlap of length *x*, and prefer less rather than more overlap

Optimum tiling paths: 5



'Banding': avoid considering segments that are greater than distance ***d*** from the start of the ***ith*** segment

References

- *Biological Sequence Analysis,* Durbin, Eddy, Krogh, Mitchison Cambridge University Press: Chapter 2: *pairwise sequence alignment*
- Henikoff and Henikoff - *Amino acid substitution matrices from protein blocks* – PNAS 1992
- Eddy SR - *Where did the BLOSUM62 alignment score matrix come from?* – Nature BioTech. 2004
- Eddy SR - *What is dynamic programming?* – Nature BioTech 2004

http://www.sanger.ac.uk/resources/software/seqtools/ (Dotter)

- Sonhammer EL *et al*. - *A dot-matrix program with dynamic threshold control suited for genomic DNA and protein sequence analysis* – Gene 1995

https://www.ebi.ac.uk/about/vertebrate-genomics/software/exonerate (Exonerate)

- Slater GS, Birney E. - *Automated generation of heuristics for biological sequence comparison* - BMC Bioinformatics (2005) 6:31

Genome Informatics Practical 1

1) Examine the spreadsheets in pairwise_durbin.xls
(Available in Practicals section of Genome Informatics moodle)
Variants on pairwise sequence alignment:
global, local, overlap, repeat.
Look at the recursion relations, and see how they are initialised
at the edges.
Fill out the matrices.
How does one find the correct location to start the traceback from?
Carry out the traceback manually: is the path unambiguous?
First four tabs use linear gap penalties. (If you are curious, the second four tabs illustrate affine penalties: d=12, e=2)

2) By hand, globally align "HEAT" and "HAT
Scoring: +1 match; -1 mismatch; -2 gap

For programmers:

3) In a language of your choice, write code that identifies overlaps between two input sequences.

HOMEWORK BEFORE NEXT PRACTICAL:
http://perldoc.perl.org/perlintro.html

Worksheet - Optimum Tiling Paths

Avaliable on wiki:

- Calculate optimal path given tiles and penalties
- Write program to calculate optimal path

- Answers + example code on wiki after next lecture