#### Pairwise Sequence Alignment

BMI/CS 576
www.biostat.wisc.edu/bmi576/
Mark Craven
craven@biostat.wisc.edu
Fall 2011

### Pairwise alignment: task definition

#### Given

- a pair of sequences (DNA or protein)
- a method for scoring a candidate alignment

#### Do

 determine the correspondences between substrings in the sequences such that the similarity score is maximized

000000001 >>>>>> 000340701	CAGACTGCAAGATTCCCTGCATTACGGGGAGCCCTAAGCAGGCACCTTTAACCTTGAGCC	000000060 <<<<<< 000340642	
000000061 >>>>>> 000340641	TGCTCGCTAGCGACCTCCAGAGGGCCAGGATGGTTTGGGACCGGTGTGATAGAGCACTTT	000000120 <<<<<< 000340582	
000000121 >>>>>> 000340581	TCTAGTTTTAACGACTCTATTTGCATAACGGAGACCAAGCGTCCTCCCGCCGGTCTGCCG	000000180 <<<<<< 000340522	
000000181 >>>>>> 000340521	TGCTTTCTTCCCTCCTCCCGGTCCTCTGTACCCTCatctg-tgctccctctctgcttc	000000239 <<<<<< 000340462	
000000240 >>>>>> 000340461	ccagaagatccgttctcaaggggactcactcggagcttgagtcccgggctcttctccctc	000000299 <<<<<< 000340402	
000000300 >>>>>> 000340401	ggcccggcaagatccagaggcacctgggggagtggagttgagccagcc	000000359 <<<<<< 000340342	
000000360 >>>>>> 000340341	agccgcctggcactgcccqaccctaacaaagggaatcggactgcgccgtgcgagcgc	000000419 <<<<<< 000340282	
000000420 >>>>>> 000340281	tcgqtgggctcagcgcagccgccttcccgattgagagagctgcaagtgaGTAGGGG	000000479 <<<<<< 000340222	
000000480 >>>>>> 000340221	CGAGGCGGCCACGGTCTGAAGAAGGAGTCCAGGGCTGCGCTGAGAGCCACCCTAG	000000539 <<<<<< 000340164	
000000540 >>>>>> 000340163	CTGCGGGAGTAGGAGCCGCCGCCCGGGCTGGGGATGTCTGTC	000000599 <<<<<< 000340104	
000000600 >>>>>> 000340103	ACCGCCTTGGCCTCCCTCATCCCCACCCCACCCCCACCCCCGGCAGGCA	000000659 <<<<<< 000340075	
000000660 >>>>>> 000340074	CTGCGCACCCATAACAAACCCACGTGTGCGGGGACGCAGGAACCTACACTGCAGCGTCTG	000000719 <<<<<< 000340015	
000000720 >>>>>> 000340014	TATTGTGTAAGGCGTTGACTCGCGGAGTGAGGGTCTCTTTAGATCTTGCCCACTTTGGTA	000000779 <<<<<< 000339955	
000000780 >>>>>> 000339954	ACAGAGCAGGAAAACAAGCCCGGTGGAAGGAGAGGGGGGGG	000000839 <<<<<< 000339897	
000000840 >>>>>> 000339896	CACGTGTGCCCCGGCTGAACCTCTTTTGGTCAGG	000000873 <<<<<< 000339837	
000000874 >>>>>> 000339836	TTATTGTTCCAGCCAAACCAACGGTGCGCGCGTTTGGAGCCACCTACTACTGCTGGAGA         TTATTGTTCCAACGGGTGCGCGCTTGGAGGCCACCTACTGCTGGGTG	000000933 <<<<<< 000339789	

# DNA alignment example

genomic sequence for part of the 6T6Gal gene in mouse and rat

Maksimovic et al., Glycobiology 21:467-48, 2011

### Protein alignment example

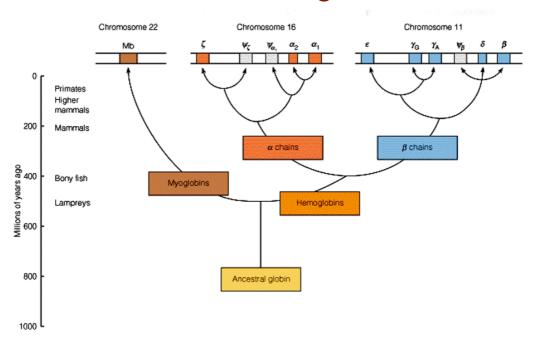
OprD	MKVMKWSAIALAVSAGSTQFAVADAFVSDQAEAKGFIEDSSLDLLLR	47
PhaK	MSGKTTTMNRTHFMSAACLATLALPVPAMADFIGDSHARLELR	43
	**.* ***.** * **	
OprD	NYYFNRDGKSGSGDRVDWTQGFLTTYESGFTQGTVGFGVDAFGYLGL	94
PhaK	NHYINRDFRQSNAPQAKAEEWGQGFTAKLESGFTEGPVGFGVDAMGQLGI	93
	*.*.***	
OprD	KLDGTSDKTGTGNLPVMNDGK-PRDDYSRAGGAVKVRISKTMLKWGEMQP	143
PhaK	KLDSSRDRRNTGLLPFGPNSHEPVDDYSELGLTGKIRVSKSTLRLGTLQP	143
	****** ** * **** * . *.**. *.	
OprD	TAPVFAAGGSRLFPQTATGFQLQSSEFEGLDLEAGHFTEGKEPTTVKSRG	193
PhaK	ILPVVVYNDTRLLASTFQGGLLTSQDVDGLTFNAGRLTKANLRDS-SGRD	192
	. **** * * * *******.	
OprD	ELYATYAGETAKSADFIGGRYAITDNLSASLYGAELEDIYRQYYLNSNYT	243
PhaK	DIGYGAASSDHLDFGGGSYAITPQTSVSYYYAKLEDIYRQQFVGLIDT	240
	* ** **.**** . *.* * *,****** *	
OprD	IPLASDQSLGFDFNIYRTNDEGKAKAGDISNTTWSLAAAYTLDAHTFT	291
PhaK	RPLSEGVSLRSDLRYFDSRNDGAERAGNIDNRNFNAMFTLGVRAHKFT	288
	** ** ****.*.** .**. **.*	
OprD	LAYQKVHGDQPFDYIGFGRNGSGAGGDSIFLANSVQYSDFNGPGEKSWQA	341
PhaK	ATWQQMSGDSAFPFVNGGDP-FTVNLVTYNTFTRAGLDSWQV	329
	* ** .* ***. * * * *** .*	
OprD	RYDLNLASYGVPGLTFMVRYINGKDIDGTKMSDNNVGYKNYGYGEDGKHH	391
PhaK	RYDYDFVAMGIPGLSFMTRYTDGRHAETATVSNGRER	366
	*** *.***.*** *	
OprD	ETNLEAKYVVQSGPAKDLSFRIRQAWHRANADQGEGDQNEFRLIVDYPLS	441
PhaK	ERDTDITYVIQSGPFKDVSLRWRNVTFRSGNGLTNAVDEN-RLIIGYTLA	415
	***.*** **.*.* * *	
OprD	IL 443	
PhaK	LW 417	

Alignment of the PhaK protein from Pseudomonas putida and OprD protein from Pseudomonas aeruginos

#### The role of homology in alignment

- homology: similarity due to descent from a common ancestor
- often we can infer homology from similarity
- thus we can sometimes infer structure/function from sequence similarity

## Homology example: evolution of the globins



#### Homology

- homologous sequences can be divided into two groups
  - *orthologous sequences*: sequences that differ because they are found in different species (e.g. human  $\alpha$ -globin and mouse  $\alpha$ -globin)
  - paralogous sequences: sequences that differ because of a gene duplication event (e.g. human  $\alpha$ -globin and human  $\beta$ -globin, various versions of both )

#### **DNA** sequence edits

substitutions: ACGA → AGGA

insertions: ACGA → ACCGGAGA

deletions: ACGGAGA → AGA

transpositions: ACGGAGA → AAGCGGA

inversions: ACGGAGA → ACTCCGA

#### Mismatches and gaps

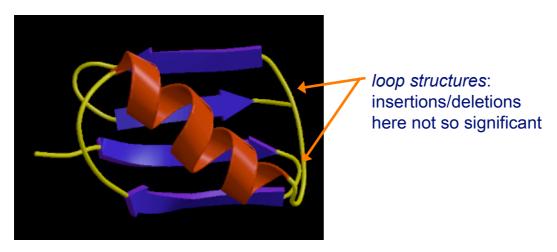
- substitutions in *homologous* sequences result in mismatches in an alignment
- insertions/deletions in *homologous* sequences result in mismatches in an alignment

#### Alignment scales

- for short DNA sequences (gene scale) we will generally only consider
  - substitutions
  - insertions/deletions
- for longer DNA sequences (genome scale) we will consider additional events
  - transpositions
  - inversions
- in this course we will focus on the case of short sequences

## Insertions/deletions and protein structure

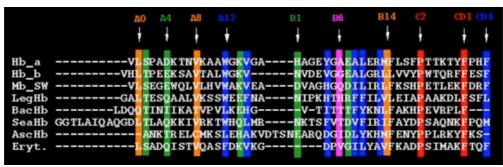
- Why is it that two "similar" sequences may have large insertions/deletions?
  - some insertions and deletions may not significantly affect the structure of a protein



#### Example alignment: globins

- figure at right shows prototypical structure of globins
- figure below shows part of alignment for 8 globins





#### Issues in sequence alignment

- the sequences we're comparing typically differ in length
- there may be only a relatively small region in the sequences that matches
- we want to allow partial matches (i.e. some amino acid pairs are more substitutable than others)
- variable length regions may have been inserted/ deleted from the common ancestral sequence

### Types of alignment

- global: find best match of both sequences in their entirety
- local: find best subsequence match
- semi-global: find best match without penalizing gaps on the ends of the alignment

### Scoring an alignment: what is needed?

- · substitution matrix
  - s(a,b) indicates score of aligning character a with character b
- · gap penalty function
  - -w(g) indicates cost of a gap of length g

#### Blosum 62 substitution matrix

```
BLOSUM62

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

#### Linear gap penalty function

- different gap penalty functions require somewhat different dynamic programming algorithms
- the simplest case is when a linear gap function is used

$$w(g) = -g \times d$$

where d is a constant

· we'll start by considering this case

#### Scoring an alignment

- the score of an alignment is the sum of the scores for pairs of aligned characters plus the scores for gaps
- · example: given the following alignment

• we would score it by  $s(\mathbf{V}, \mathbf{A}) + s(\mathbf{A}, \mathbf{I}) + s(\mathbf{H}, \mathbf{Q}) + s(\mathbf{V}, \mathbf{L}) - 3d + s(\mathbf{D}, \mathbf{G}) - 2d$ 

### The space of global alignments

some possible global alignments for ELV and VIS

 Can we find the highest scoring alignment by enumerating all possible alignments and picking the best?

### Number of possible alignments

- given sequences of length m and n
- assume we don't count as distinct  $^{C-}_{-G}$  and  $^{-C}_{G-}$
- we can have as few as 0 and as many as min{m, n} aligned pairs
- therefore the number of possible alignments is given by

$$\sum_{k=0}^{\min\{m,n\}} \binom{n}{k} \binom{m}{k} = \binom{n+m}{n}$$

#### Number of possible alignments

there are

$$\binom{2n}{n} = \frac{(2n)!}{(n!)^2} \approx \frac{2^{2n}}{\sqrt{\pi n}}$$

possible global alignments for 2 sequences of length *n* 

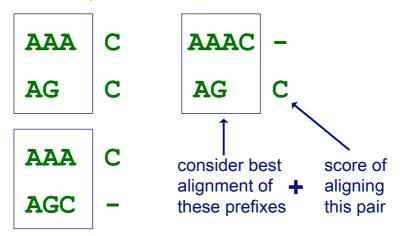
- e.g. two sequences of length 100 have  $\approx 10^{77}$  possible alignments
- but we can use dynamic programming to find an optimal alignment efficiently

# Pairwise alignment via dynamic programming

- first algorithm by Needleman & Wunsch, *Journal of Molecular Biology*, 1970
- dynamic programming: solve an instance of a problem by taking advantage of computed solutions for smaller subparts of the problem
- determine best alignment of two sequences by determining best alignment of all prefixes of the sequences

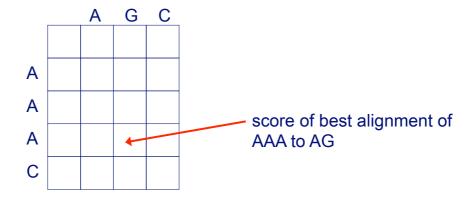
#### Dynamic programming idea

- consider last step in computing alignment of AAAC with AGC
- three possible options; in each we'll choose a different pairing for end of alignment, and add this to best alignment of previous characters



#### Dynamic programming idea

- given an *n*-character sequence *x*, and an *m*-character sequence *y*
- construct an (n+1) × (m+1) matrix F
- F(i, j) = score of the best alignment of x[1...i] with y[1...j]

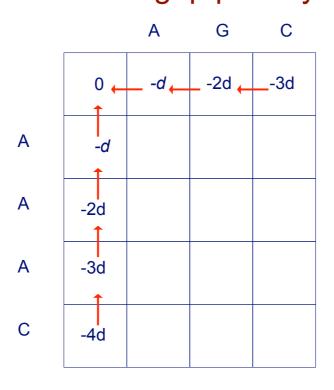


# DP algorithm for global alignment with linear gap penalty

 one way to specify the DP is in terms of its recurrence relation:

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

## Initializing matrix: global alignment with linear gap penalty



# DP algorithm sketch: global alignment

- initialize first row and column of matrix
- fill in rest of matrix from top to bottom, left to right
- for each F ( i, j ), save pointer(s) to cell(s) that resulted in best score
- F (m, n) holds the optimal alignment score; trace pointers back from F (m, n) to F (0, 0) to recover alignment

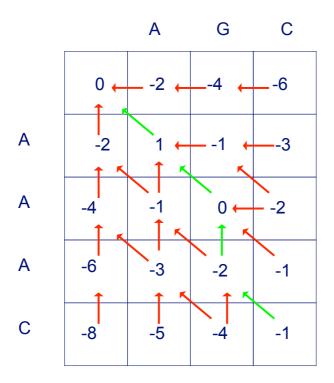
#### Global alignment example

suppose we choose the following scoring scheme:

$$s(x_i, y_i) =$$
+1 when  $x_i = y_i$ 
-1 when  $x_i \neq y_i$ 

d (penalty for aligning with a gap) = 2

### Global alignment example



#### one optimal alignment

x: A A A C y: A G - C

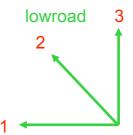
#### **DP** comments

- works for either DNA or protein sequences, although the substitution matrices used differ
- finds an optimal alignment
- the exact algorithm (and computational complexity) depends on gap penalty function (we'll come back to this issue)

### Equally optimal alignments

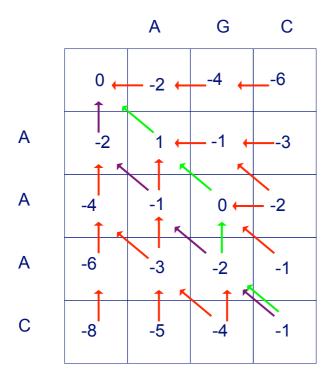
- many optimal alignments may exist for a given pair of sequences
- can use preference ordering over paths when doing traceback





 highroad and lowroad alignments show the two most different optimal alignments

#### Highroad & lowroad alignments



#### highroad alignment

x: A A A C y: A G - C

#### lowroad alignment

x: A A A C y: - A G C

#### Dynamic programming analysis

· recall, there are

$$\binom{2n}{n} = \frac{(2n)!}{(n!)^2} \approx \frac{2^{2n}}{\sqrt{\pi n}}$$

possible global alignments for 2 sequences of length *n* 

but the DP approach finds an optimal alignment efficiently

#### Computational complexity

- initialization: O(m), O(n) where sequence lengths are m, n
- filling in rest of matrix: O(mn)
- traceback: O(m + n)
- hence, if sequences have nearly same length, the computational complexity is

$$O(n^2)$$