Pairwise Sequence Alignment

Stuart M. Brown
NYU School of Medicine

w/ slides byFourie Joubert

Protein Evolution

"For many protein sequences, evolutionary history can be traced back 1-2 billion years"

-William Pearson

- When we align sequences, we assume that they share a common ancestor
  - They are then homologous
- Protein fold is much more conserved than protein sequence
- DNA sequences tend to be less informative than protein sequences

Definition

- Homology: related by descent
- Homologous sequence positions

Orthologous and paralogous

- Orthologous sequences differ because they are found in different species (a speciation event)
- Paralogous sequences differ due to a gene duplication event
- Sequences may be both orthologous and paralogous

Pairwise Alignment

- The alignment of two sequences (DNA or protein) is a relatively straightforward computational problem.
  - There are lots of possible alignments.
- Two sequences can **always** be aligned.
- Sequence alignments have to be **scored**.
- Often there is **more than one** solution with the same score.

Methods of Alignment

- By hand - slide sequences on two lines of a word processor
- Dot plot
  - with windows
- Rigorous mathematical approach
  - Dynamic programming (slow, optimal)
- Heuristic methods (fast, approximate)
  - BLAST and FASTA
    - Word matching and hash tables0

Align by Hand

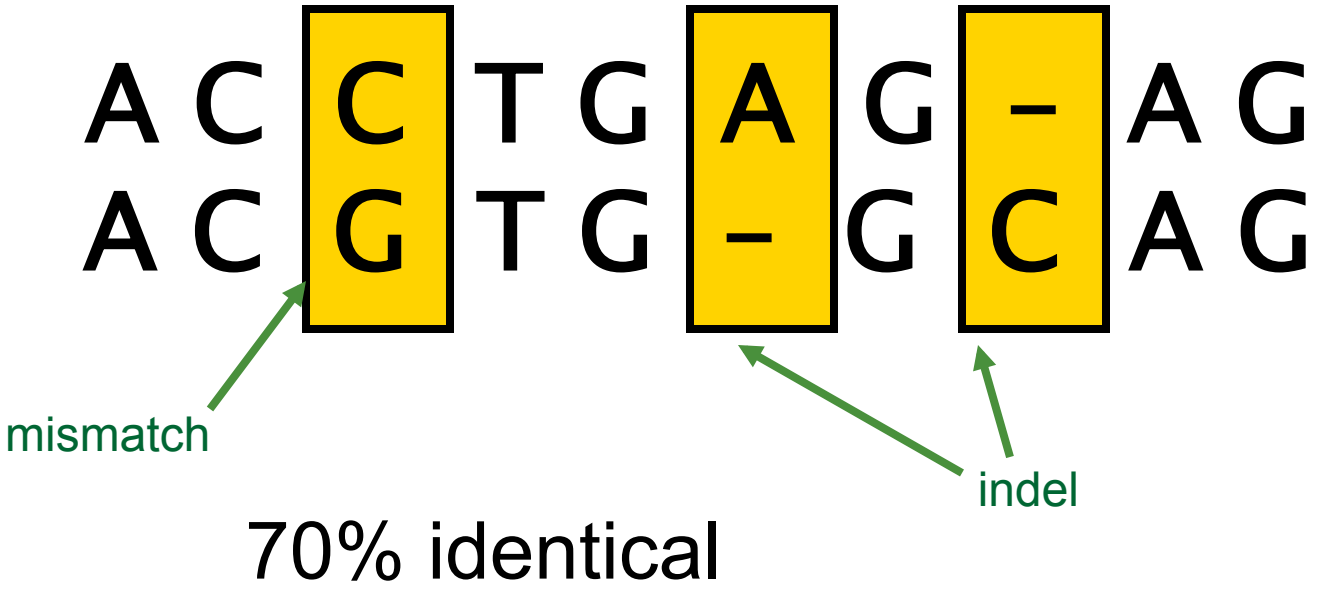
GATCGCCTA_TTACGTCCTGGAC <--

--> AGGCATACGTA_GCCCTTTCGC

You still need some kind of scoring system to find the best alignment

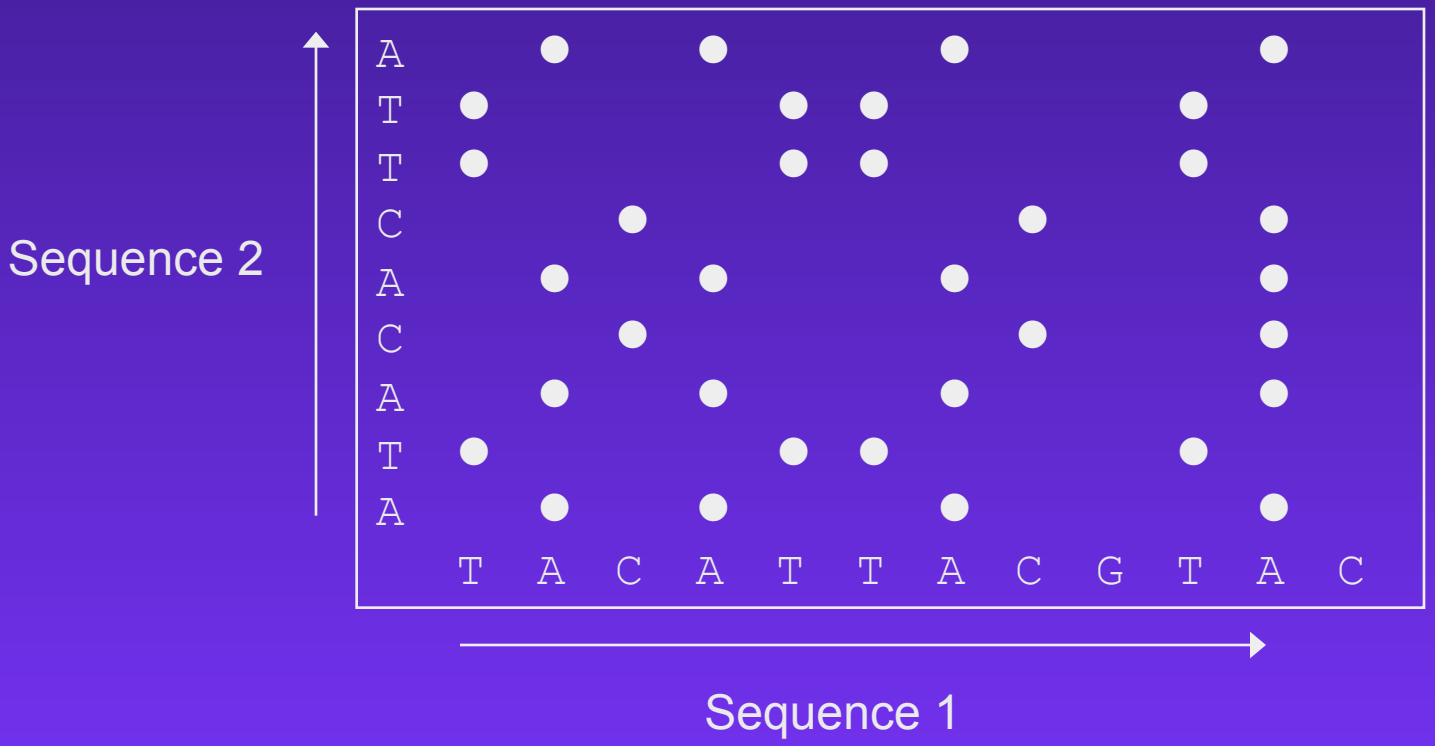
Percent Sequence Identity

- The extent to which two nucleotide or amino acid sequences are invariant



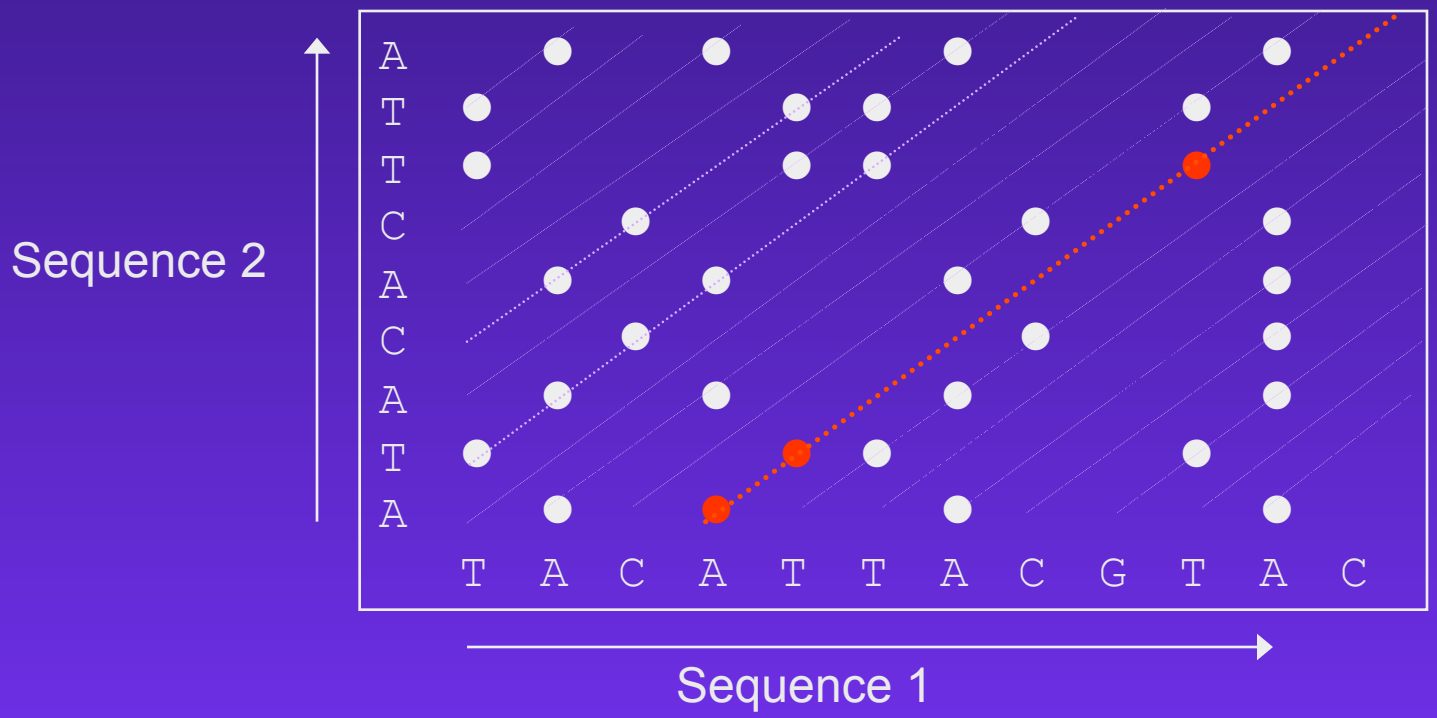
Dotplot:

A dotplot gives an overview of all possible alignments



Dotplot:

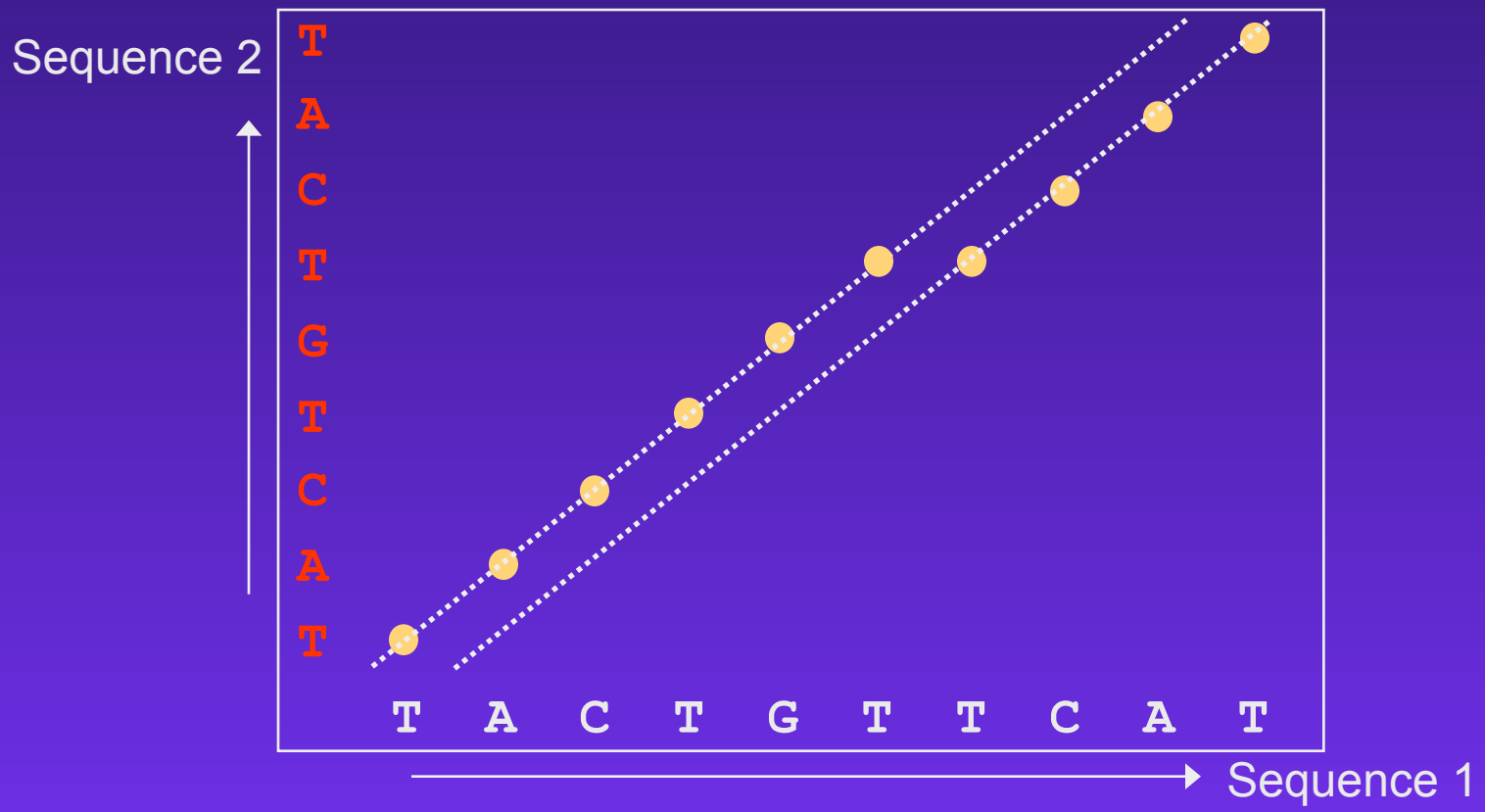
In a dotplot each diagonal corresponds to a possible (ungapped) alignment



One possible alignment:

```
T A C A T T A C G T A C
  | |       |
  A T A C A C T T A
```

Insertions / Deletions in a Dotplot



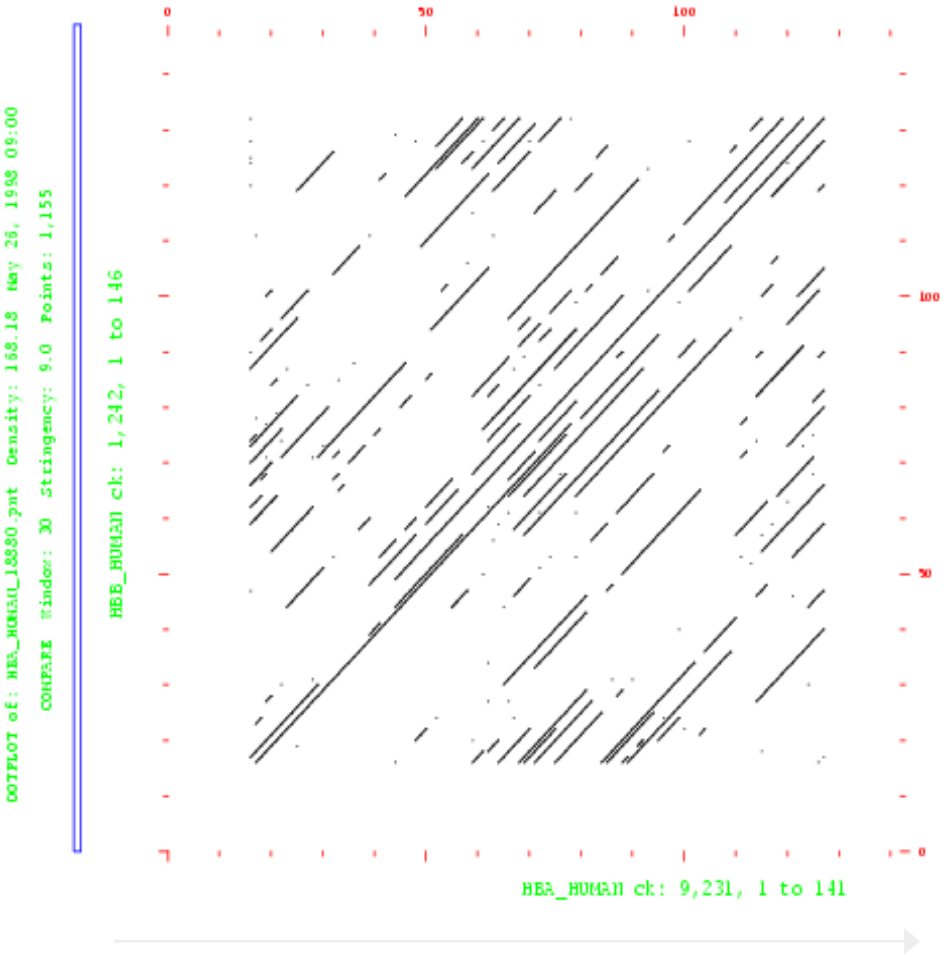

```
T A C T G - T C A T
| | | | |   | | | |
T A C T G T T C A T
```

HUSAR

Dotplot

(Window = 130 / Stringency = 9)

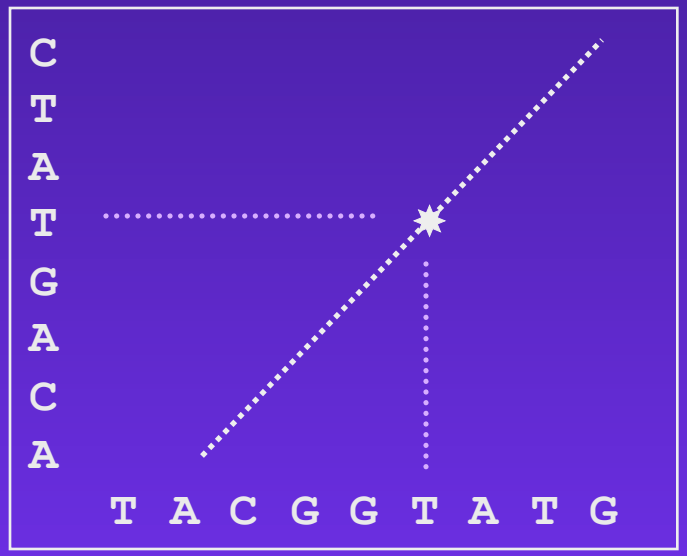
Hemoglobin β-chain



Hemoglobin α-chain

Word Size Algorithm

Word Size = 3



Window / Stringency

Scoring Matrix Filtering

Matrix: PAM250

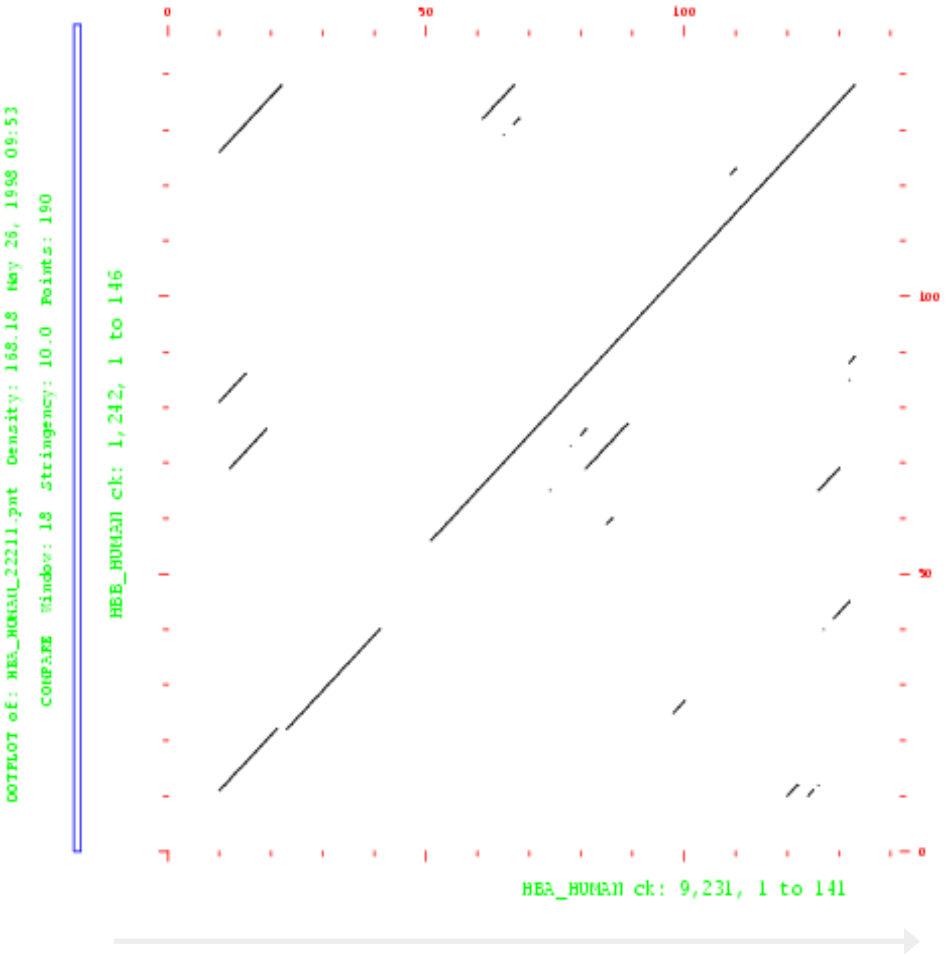
Window = 12

Stringency = 9

Dotplot

(Window = 18 / Stringency = 10)

Hemoglobin β-chain



Hemoglobin α-chain

Considerations

- The window/stringency method is more sensitive than the wordsize method (ambiguities are permitted).
- The smaller the window, the larger the weight of statistical (unspecific) matches.
- With large windows the sensitivity for short sequences is reduced.
- Insertions/deletions are not treated explicitly.

Alignment methods

- Rigorous algorithms = Dynamic Programming
  - Needleman-Wunsch (global)
  - Smith-Waterman (local)
- Heuristic algorithms
  (faster but approximate)
    - BLAST
    - FASTA

Basic principles of dynamic programming

- Creation of an **alignment path matrix**

- **Stepwise** calculation of score values

- **Backtracking** (evaluation of the optimal path)

Dynamic Programming

- Dynamic Programming is a very general programming technique.
- It is applicable when a large search space can be structured into a succession of stages, such that:
  - the initial stage contains trivial solutions to sub-problems
  - each partial solution in a later stage can be calculated by recurring a fixed number of partial solutions in an earlier stage
  - the final stage contains the overall solution

Creation of an alignment path matrix

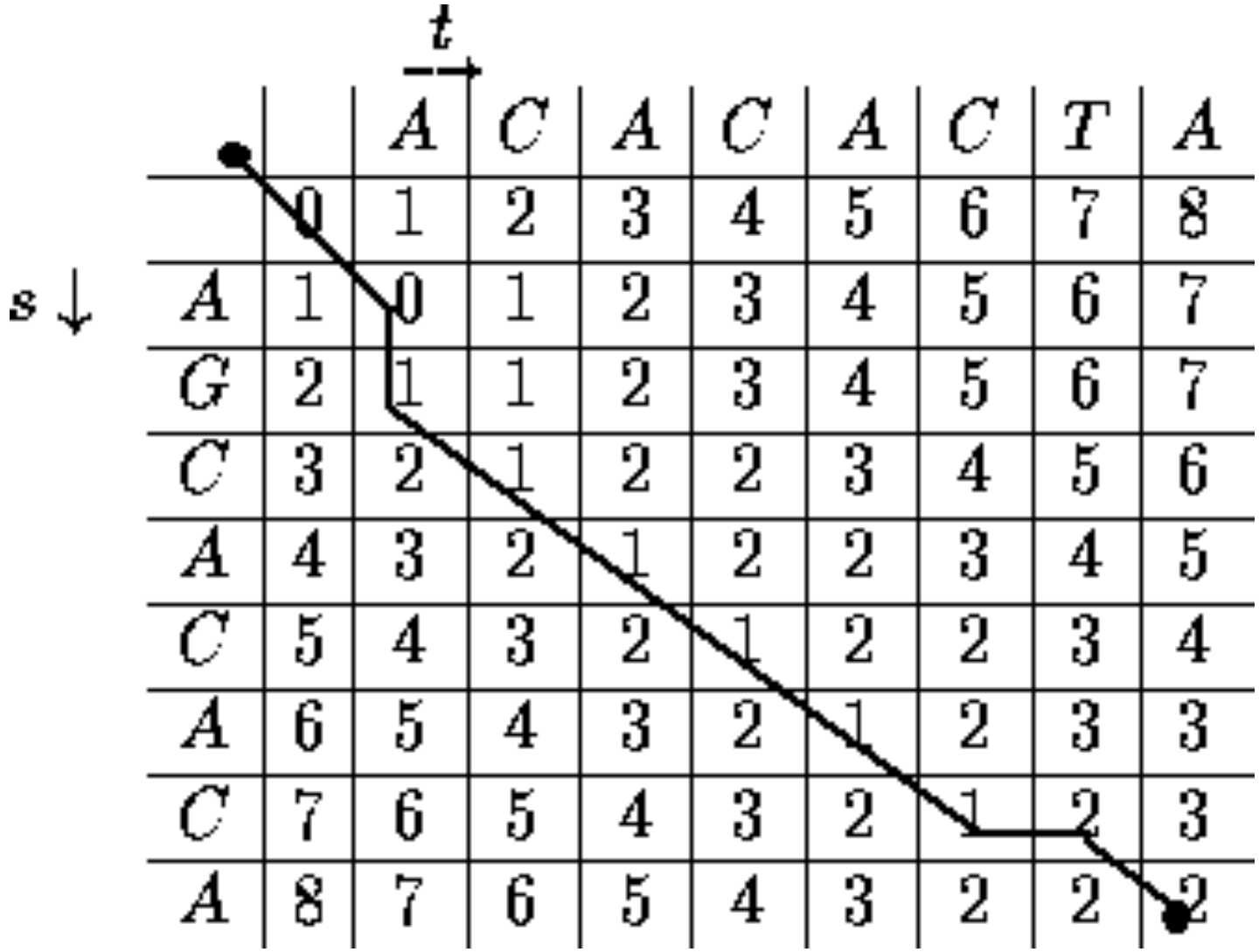
Idea:
Build up an optimal alignment using previous solutions for optimal alignments of smaller subsequences

- Construct matrix F indexed by i and j (one index for each sequence)
- $F(i,j)$ is the score of the best alignment between the initial segment $x_{1\ldots i}$ of x up to x_i and the initial segment $y_{1\ldots j}$ of y up to y_j
- Build $F(i,j)$ recursively beginning with $F(0,0) = 0$

$s:\ A\,G\,C\,A\,C\,A\,C\,-\,A$
$t:\ A\,-\,C\,A\,C\,A\,C\,T\,A$

or

$A\,G\,-\,C\,A\,C\,A\,C\,A$
$A\,C\,A\,C\,A\,C\,T\,-\,A$

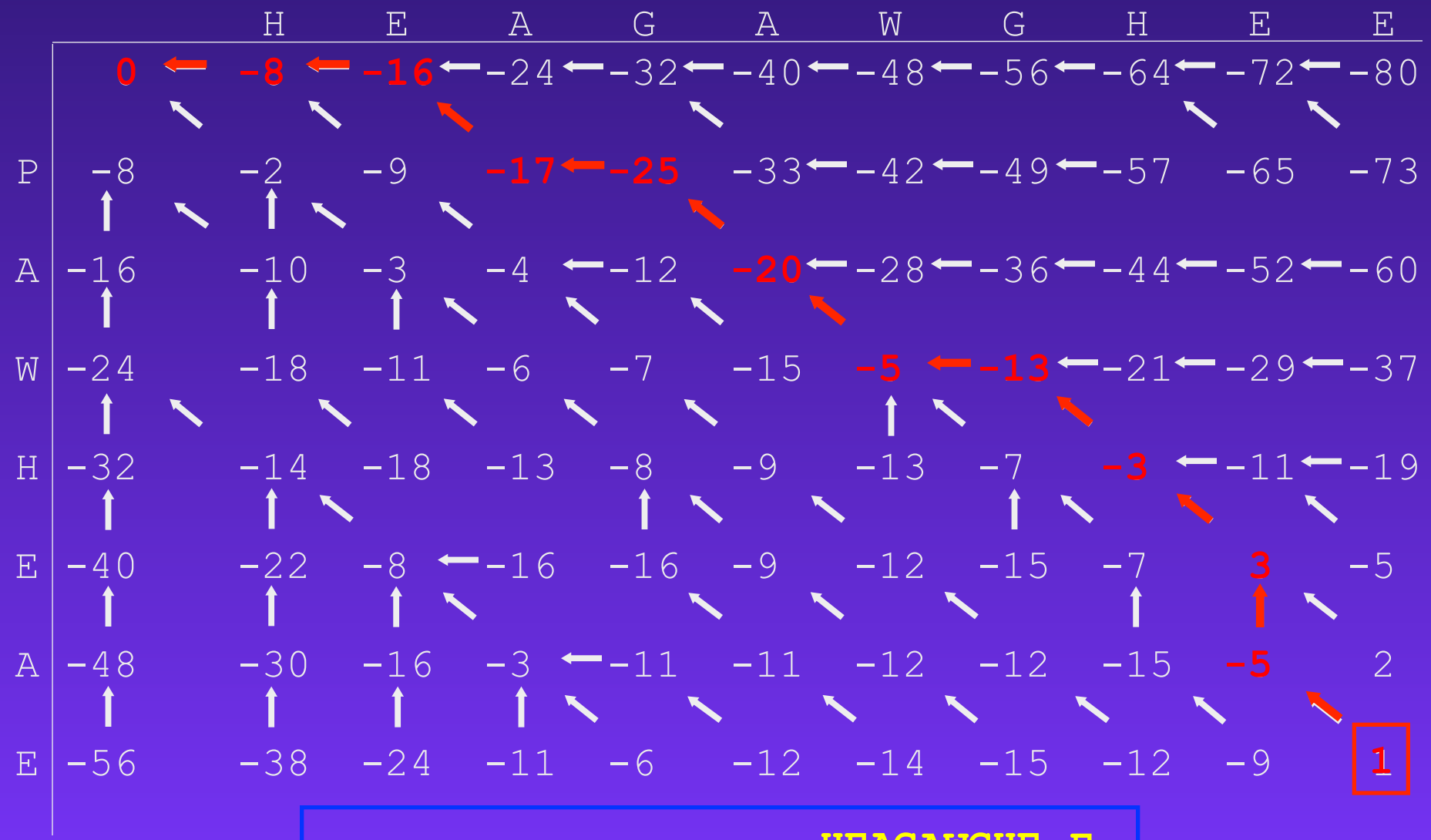


		A	C	A	C	A	C	T	A
	0	1	2	3	4	5	6	7	8
A	1	0	1	2	3	4	5	6	7
G	2	1	1	2	3	4	5	6	7
C	3	2	1	2	2	3	4	5	6
A	4	3	2	1	2	2	3	4	5
C	5	4	3	2	1	2	2	3	4
A	6	5	4	3	2	1	2	3	3
C	7	6	5	4	3	2	1	2	3
A	8	7	6	5	4	3	2	2	2

Creation of an alignment path matrix

- If $F(i\text{-}1,j\text{-}1)$, $F(i\text{-}1,j)$ and $F(i,j\text{-}1)$ are known we can calculate $F(i,j)$
- Three possibilities:
  - x_i and y_j are aligned, $F(i,j) = F(i\text{-}1,j\text{-}1) + s(x_i, y_j)$
  - x_i is aligned to a gap, $F(i,j) = F(i\text{-}1,j) - d$
  - y_j is aligned to a gap, $F(i,j) = F(i,j\text{-}1) - d$
- The best score up to (i,j) will be the **largest** of the three options

Backtracking

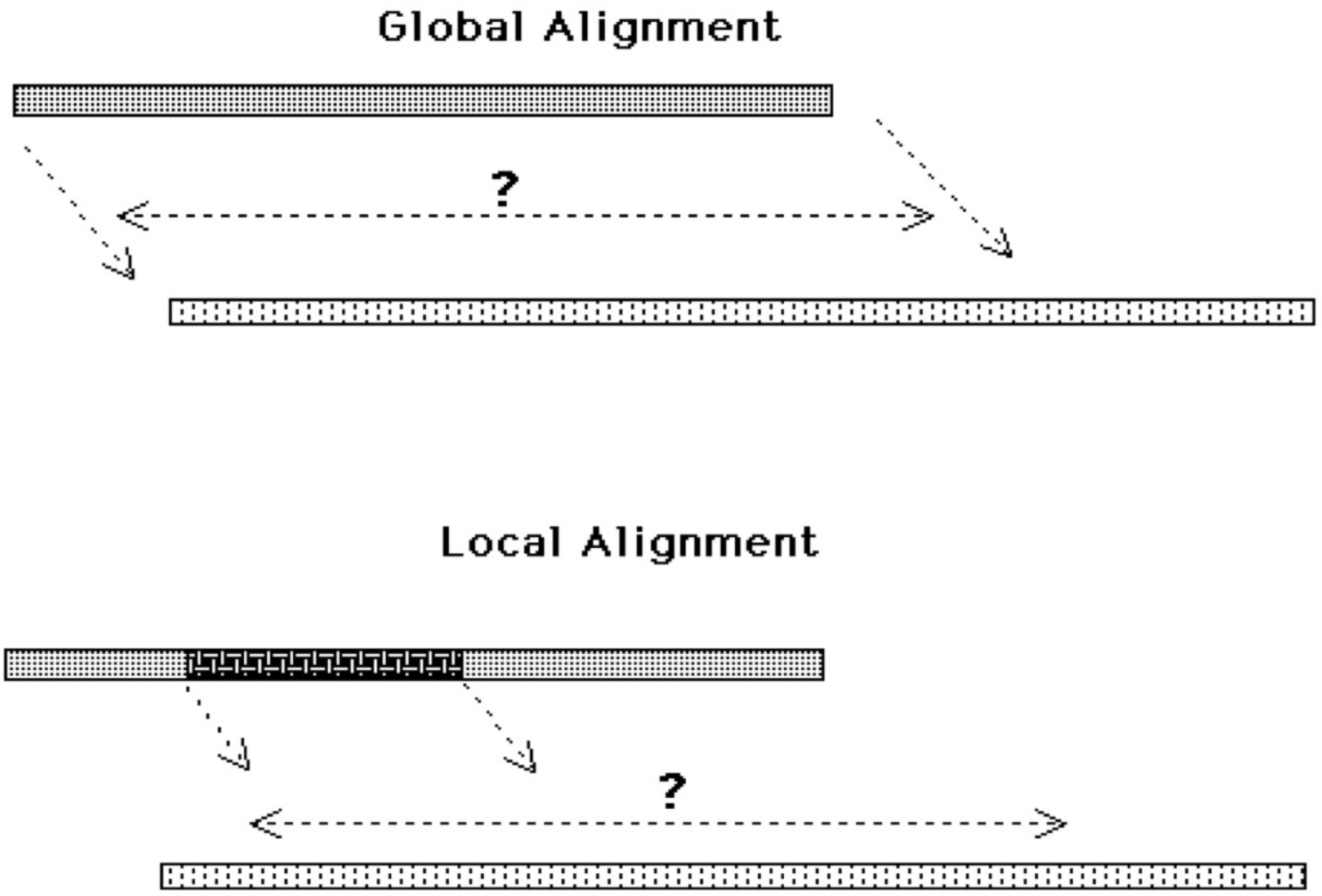


Optimal global alignment:

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

Global vs. Local Alignments

- Global alignment algorithms start at the beginning of two sequences and add gaps to each until the end of one is reached.
- Local alignment algorithms finds the region (or regions) of highest similarity between two sequences and build the alignment outward from there.


Global Alignment
?
Local Alignment
?

Global Alignment

Two closely related sequences:

```
1 TGTCGATTAAGCGGTCGTAGCTGACCTGAGATTGCCCGATGGCGTAGTAGCTGACC 56
  |||||||||  ||||||||||   |||||||| ||  ||||||||||||||||  |||||
1 TGTCGATTATGCGGTCGTAG..GACCTGAGTTTCCCCGATGGCGTAGTAGGTGACC 54
```

needle *(Needleman & Wunsch)* creates an **end-to-end** alignment.

Global Alignment

Two sequences sharing several regions of local similarity:

Global Alignment (Needleman -Wunsch)

- The the Needleman-Wunsch algorithm creates a global alignment over the length of both sequences (needle)
- Global algorithms are often not effective for highly diverged sequences - do not reflect the biological reality that two sequences may only share limited regions of conserved sequence.
  - Sometimes two sequences may be derived from ancient recombination events where only a single functional domain is shared.
- Global methods are useful when you want to force two sequences to align over their entire length

Local Alignment (Smith-Waterman)

- Local alignment
  - Identify the most similar sub-region shared between two sequences
  - Smith-Waterman
  - EMBOSS: water

Parameters of Sequence Alignment

Scoring Systems:

- Each symbol pairing is assigned a numerical value, based on a symbol comparison table.

Gap Penalties:

- Opening: The cost to introduce a gap
- Extension: The cost to elongate a gap

DNA Scoring Systems
-very simple

Sequence 1

Sequence 2

```
actaccagttcatttgatacttctcaaa
     | | |     |       ||
     taccattaccgtgttaactgaaaggacttaaagact
```

	A	G	C	T
A	**1**	0	0	0
G	0	**1**	0	0
C	0	0	**1**	0
T	0	0	0	**1**

Match: 1
Mismatch: 0
Score = 5

Protein Scoring Systems

Sequence 1 PTHPLASKTQILPEDLASEDLTI

Sequence 2 PTHPLAGERAIGLARLAEEDFGM

Scoring matrix

	C	S	T	P	A	G	N	D	.	.
C	9									
S	-1	4								
T	-1	1	5							
P	-3	-1	-1	7						
A	0	1	0	-1	4					
G	-3	0	-2	-2	0	6				
N	-3	1	0	-2	-2	0	5			
D	-3	0	-1	-1	-2	-1	1	6		
.										
.										

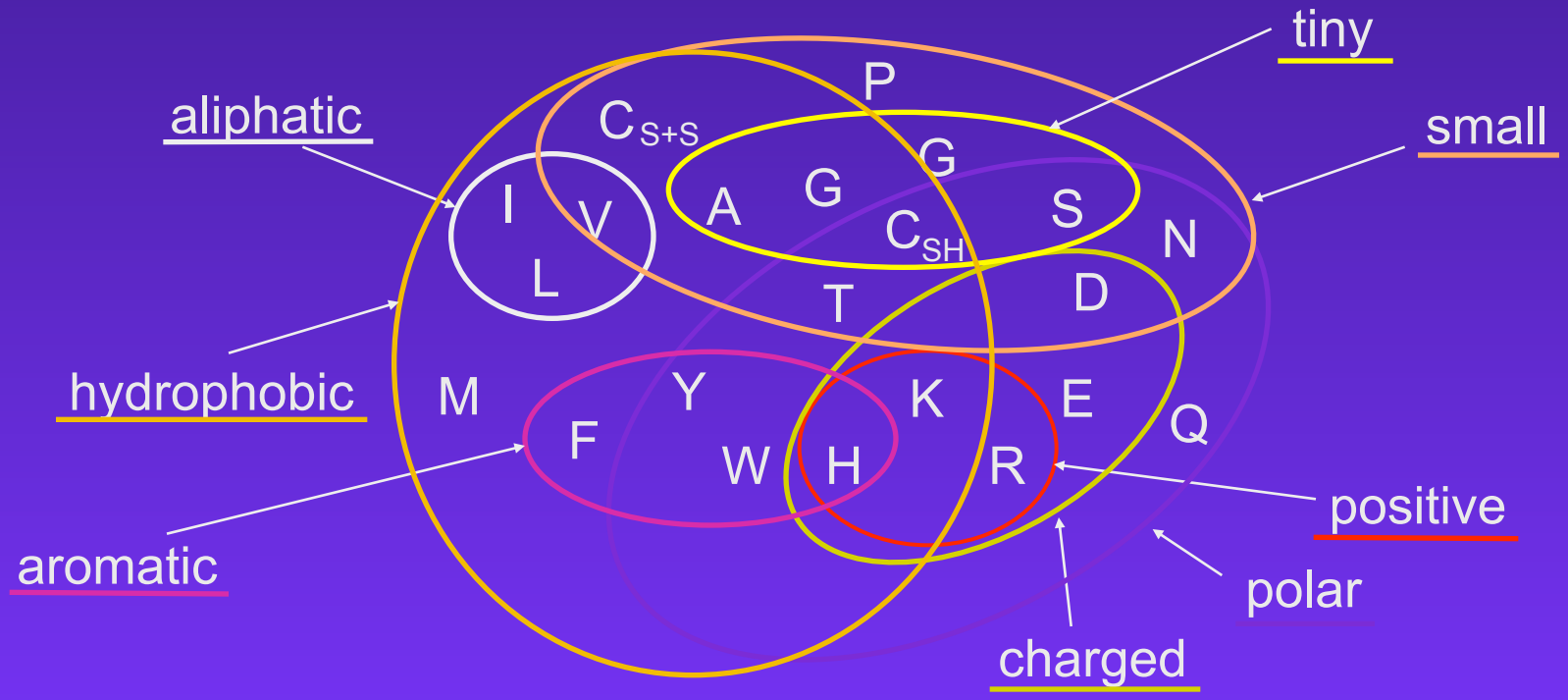
T:G = -2

T:T = 5

Score = 48

Protein Scoring Systems

- Amino acids have different biochemical and physical properties that influence their relative replaceability in evolution.



Protein Scoring Systems

- Scoring matrices reflect:
  - # of mutations to convert one to another
  - chemical similarity
  - observed mutation frequencies
  - the probability of occurrence of each amino acid

- Widely used scoring matrices:
  - **PAM**
  - **BLOSUM**

PAM matrices

- Family of matrices PAM 80, PAM 120, PAM 250
- The number with a PAM matrix represents the evolutionary distance between the sequences on which the matrix is based
- Greater numbers denote greater distances

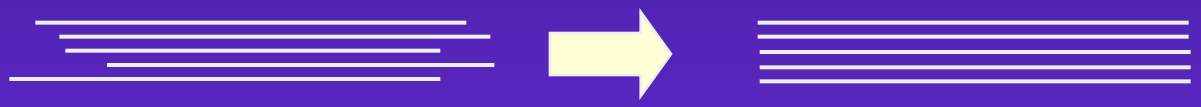
PAM (Percent Accepted Mutations) matrices

- The numbers of replacements were used to compute a so-called PAM-1 matrix.
- The PAM-1 matrix reflects an average change of 1% of all amino acid positions. PAM matrices for larger evolutionary distances can be extrapolated from the PAM-1 matrix.
- PAM250 = 250 mutations per 100 residues.
- Greater numbers mean bigger evolutionary distance

.

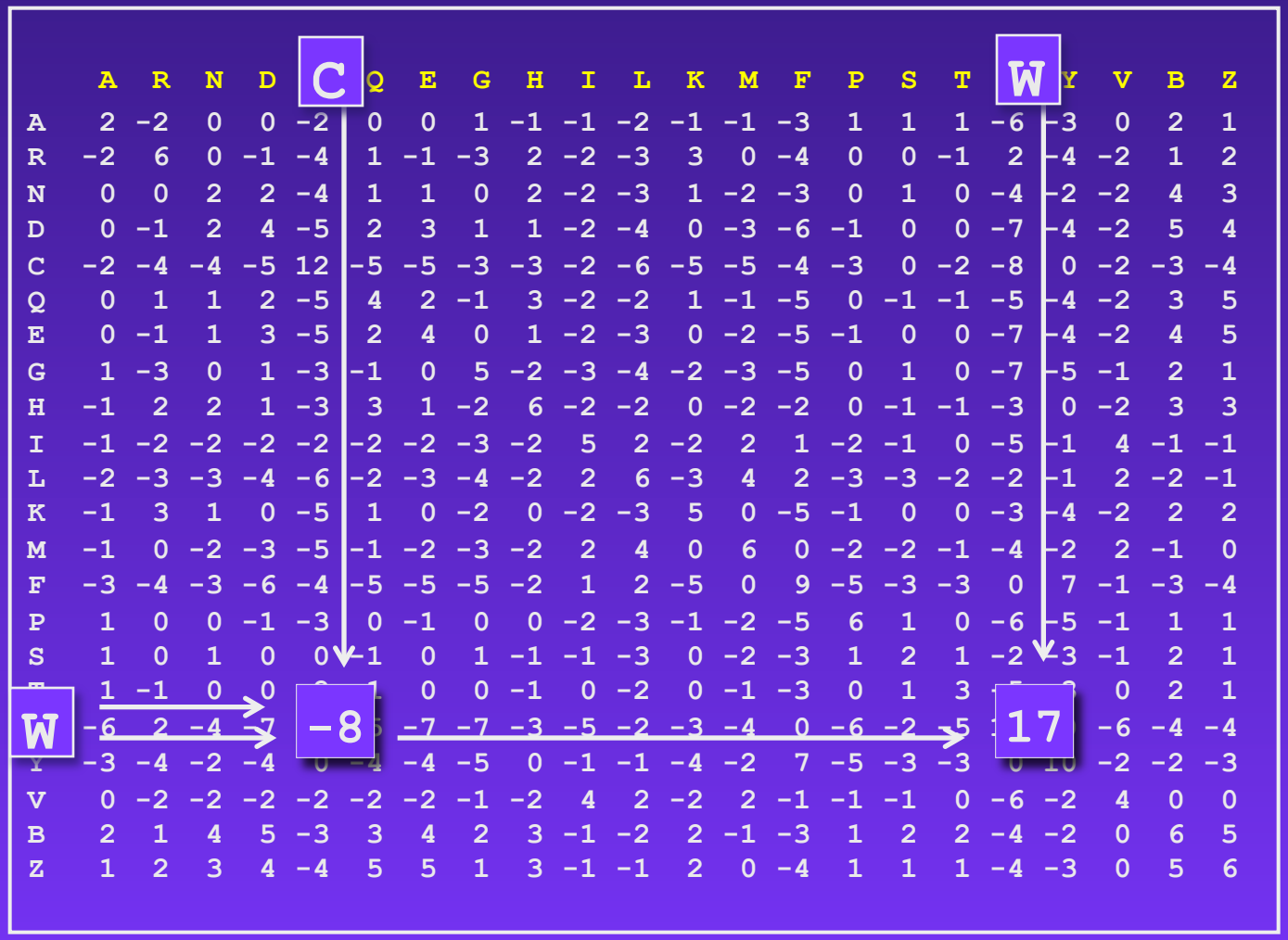
PAM (**P**ercent **A**ccepted **M**utations) **matrices**

- Derived from global alignments of **protein families** . Family members share at least 85% identity (Dayhoff *et al*., 1978).



- Construction of phylogenetic tree and ancestral sequences of each protein family
- Computation of number of replacements for each pair of amino acids

PAM 250



	A	R	N	D	C	Q	E	G	H	I	L	K	M	F	P	S	T	W	Y	V	B	Z
A	2	-2	0	0	-2	0	0	1	-1	-1	-2	-1	-1	-3	1	1	1	-6	-3	0	2	1
R	-2	6	0	-1	-4	1	-1	-3	2	-2	-3	3	0	-4	0	0	-1	2	-4	-2	1	2
N	0	0	2	2	-4	1	1	0	2	-2	-3	1	-2	-3	0	1	0	-4	-2	-2	4	3
D	0	-1	2	4	-5	2	3	1	1	-2	-4	0	-3	-6	-1	0	0	-7	-4	-2	5	4
C	-2	-4	-4	-5	12	-5	-5	-3	-3	-2	-6	-5	-5	-4	-3	0	-2	-8	0	-2	-3	-4
Q	0	1	1	2	-5	4	2	-1	3	-2	-2	1	-1	-5	0	-1	-1	-5	-4	-2	3	5
E	0	-1	1	3	-5	2	4	0	1	-2	-3	0	-2	-5	-1	0	0	-7	-4	-2	4	5
G	1	-3	0	1	-3	-1	0	5	-2	-3	-4	-2	-3	-5	0	1	0	-7	-5	-1	2	1
H	-1	2	2	1	-3	3	1	-2	6	-2	-2	0	-2	-2	0	-1	-1	-3	0	-2	3	3
I	-1	-2	-2	-2	-2	-2	-2	-3	-2	5	2	-2	2	1	-2	-1	0	-5	-1	4	-1	-1
L	-2	-3	-3	-4	-6	-2	-3	-4	-2	2	6	-3	4	2	-3	-3	-2	-2	-1	2	-2	-1
K	-1	3	1	0	-5	1	0	-2	0	-2	-3	5	0	-5	-1	0	0	-3	-4	-2	2	2
M	-1	0	-2	-3	-5	-1	-2	-3	-2	2	4	0	6	0	-2	-2	-1	-4	-2	2	-1	0
F	-3	-4	-3	-6	-4	-5	-5	-5	-2	1	2	-5	0	9	-5	-3	-3	0	7	-1	-3	-4
P	1	0	0	-1	-3	0	-1	0	0	-2	-3	-1	-2	-5	6	1	0	-6	-5	-1	1	1
S	1	0	1	0	0	-1	0	1	-1	-1	-3	0	-2	-3	1	2	1	-2	-3	-1	2	1
T	1	-1	0	0	-2	-1	0	0	-1	0	-2	0	-1	-3	0	1	3	-5	-3	0	2	1
W	-6	2	-4	-7	-8	-5	-7	-7	-3	-5	-2	-3	-4	0	-6	-2	-5	17	0	-6	-4	-4
Y	-3	-4	-2	-4	0	-4	-4	-5	0	-1	-1	-4	-2	7	-5	-3	-3	0	10	-2	-2	-3
V	0	-2	-2	-2	-2	-2	-2	-1	-2	4	2	-2	2	-1	-1	-1	0	-6	-2	4	0	0
B	2	1	4	5	-3	3	4	2	3	-1	-2	2	-1	-3	1	2	2	-4	-2	0	6	5
Z	1	2	3	4	-4	5	5	1	3	-1	-1	2	0	-4	1	1	1	-4	-3	0	5	6

PAM - limitations

- Based on only one original dataset
- Examines proteins with few differences (85% identity)
- Based mainly on small globular proteins so the matrix is biased

BLOSUM matrices

- Different BLOSUM*n* matrices are calculated independently from BLOCKS (ungapped local alignments)
- BLOSUM*n* is based on a cluster of BLOCKS of sequences that share at least *n* percent identity
- BLOSUM62 represents closer sequences than BLOSUM45

BLOSUM (**Blo**cks **Su**bstitution **M**atrix)

- Derived from alignments of domains of **distantly** related proteins (Henikoff & Henikoff,1992).

- Occurrences of each amino acid pair in each column of each block alignment is counted.

- The numbers derived from all blocks were used to compute the BLOSUM matrices.

A - C = 4
A - E = 2
C - E = 2
A - A = 1
C - C = 1

The Blosum50 Scoring Matrix

	A	R	N	D	C	Q	E	G	H	I	L	K	M	F	P	S	T	W	Y	V	B	Z	X	*
A	5	-2	-1	-2	-1	-1	-1	0	-2	-1	-2	-1	-1	-3	-1	1	0	-3	-2	0	-2	-1	-1	-5
R	-2	7	-1	-2	-4	1	0	-3	0	-4	-3	3	-2	-3	-3	-1	-1	-3	-1	-3	-1	0	-1	-5
N	-1	-1	7	2	-2	0	0	0	1	-3	-4	0	-2	-4	-2	1	0	-4	-2	-3	4	0	-1	-5
D	-2	-2	2	8	-4	0	2	-1	-1	-4	-4	-1	-4	-5	-1	0	-1	-5	-3	-4	5	1	-1	-5
C	-1	-4	-2	-4	13	-3	-3	-3	-3	-2	-2	-3	-2	-2	-4	-1	-1	-5	-3	-1	-3	-3	-2	-5
Q	-1	1	0	0	-3	7	2	-2	1	-3	-2	2	0	-4	-1	0	-1	-1	-1	-3	0	4	-1	-5
E	-1	0	0	2	-3	2	6	-3	0	-4	-3	1	-2	-3	-1	-1	-1	-3	-2	-3	1	5	-1	-5
G	0	-3	0	-1	-3	-2	-3	8	-2	-4	-4	-2	-3	-4	-2	0	-2	-3	-3	-4	-1	-2	-2	-5
H	-2	0	1	-1	-3	1	0	-2	10	-4	-3	0	-1	-1	-2	-1	-2	-3	2	-4	0	0	-1	-5
I	-1	-4	-3	-4	-2	-3	-4	-4	-4	5	2	-3	2	0	-3	-3	-1	-3	-1	4	-4	-3	-1	-5
L	-2	-3	-4	-4	-2	-2	-3	-4	-3	2	5	-3	3	1	-4	-3	-1	-2	-1	1	-4	-3	-1	-5
K	-1	3	0	-1	-3	2	1	-2	0	-3	-3	6	-2	-4	-1	0	-1	-3	-2	-3	0	1	-1	-5
M	-1	-2	-2	-4	-2	0	-2	-3	-1	2	3	-2	7	0	-3	-2	-1	-1	0	1	-3	-1	-1	-5
F	-3	-3	-4	-5	-2	-4	-3	-4	-1	0	1	-4	0	8	-4	-3	-2	1	4	-1	-4	-4	-2	-5
P	-1	-3	-2	-1	-4	-1	-1	-2	-2	-3	-4	-1	-3	-4	10	-1	-1	-4	-3	-3	-2	-1	-2	-5
S	1	-1	1	0	-1	0	-1	0	-1	-3	-3	0	-2	-3	-1	5	2	-4	-2	-2	0	0	-1	-5
T	0	-1	0	-1	-1	-1	-1	-2	-2	-1	-1	-1	-1	-2	-1	2	5	-3	-2	0	0	-1	0	-5
W	-3	-3	-4	-5	-5	-1	-3	-3	-3	-3	-2	-3	-1	1	-4	-4	-3	15	2	-3	-5	-2	-3	-5
Y	-2	-1	-2	-3	-3	-1	-2	-3	2	-1	-1	-2	0	4	-3	-2	-2	2	8	-1	-3	-2	-1	-5
V	0	-3	-3	-4	-1	-3	-3	-4	-4	4	1	-3	1	-1	-3	-2	0	-3	-1	5	-4	-3	-1	-5
B	-2	-1	4	5	-3	0	1	-1	0	-4	-4	0	-3	-4	-2	0	0	-5	-3	-4	5	2	-1	-5
Z	-1	0	0	1	-3	4	5	-2	0	-3	-3	1	-1	-4	-1	0	-1	-2	-2	-3	2	5	-1	-5
X	-1	-1	-1	-1	-2	-1	-1	-2	-1	-1	-1	-1	-1	-2	-2	-1	0	-3	-1	-1	-1	-1	-1	-5
*****	-5	-5	-5	-5	-5	-5	-5	-5	-5	-5	-5	-5	-5	-5	-5	-5	-5	-5	-5	-5	-5	-5	-5	1

BLOSUM (**Blo**cks **Su**bstitution **M**atrix)

- Sequences within blocks are clustered according to their level of identity.
- Clusters are counted as a single sequence.
- Different BLOSUM matrices differ in the percentage of sequence identity used in clustering.
- The number in the matrix name (e.g. 62 in BLOSUM62) refers to the percentage of sequence identity used to build the matrix.
- Greater numbers mean smaller evolutionary distance.

PAM Vs. BLOSUM

PAM100	=	BLOSUM90
PAM120	=	BLOSUM80
PAM160	=	BLOSUM60
PAM200	=	BLOSUM52
PAM250	=	BLOSUM45

More distant sequences

- PAM120 for general use
- PAM60 for close relations
- PAM250 for distant relations

- BLOSUM62 for general use
- BLOSUM80 for close relations
- BLOSUM45 for distant relations

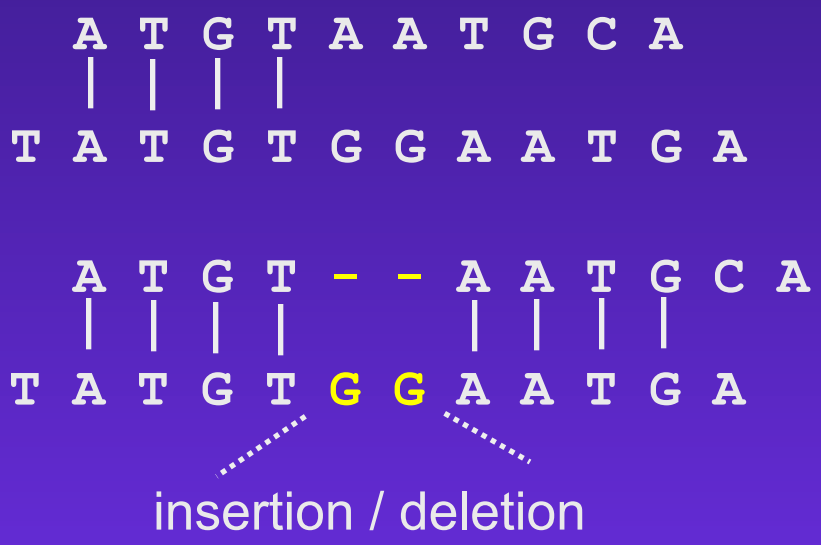
TIPS on choosing a scoring matrix

- Generally, BLOSUM matrices perform better than PAM matrices for local similarity searches (Henikoff & Henikoff, 1993).

- When comparing **closely related** proteins one should use **lower PAM or higher BLOSUM** matrices, for **distantly related** proteins **higher PAM or lower BLOSUM** matrices.

- For database searching the commonly used matrix is BLOSUM62.

Scoring Insertions and Deletions



The creation of a gap is **penalized** with a negative score value.

Why Gap Penalties?

Gaps not permitted Score: 0

```
1 GTGATAGACACAGACCGGTGGCATTGTGG 29
  |||    |   | |||     |    || || |
1 GTGTCGGGAAGAGATAACTCCGATGGTTG 29
```

Match = 5
Mismatch = -4

Gaps allowed but not penalized Score: 88

```
1 GTG.ATAG.ACACAGA..CCGGT..GGCATTGTGG 29
  |||  ||  |  |  |  |||   ||   |   |   ||  ||  |
1 GTGTAT.GGA.AGAGATACC..TCCG..ATGGTTG 29
```

Why Gap Penalties?

- The optimal alignment of two similar sequences is usually that which
  - **maximizes** the number of matches and
  - **minimizes** the number of gaps.
  - There is a tradeoff between these two
    - adding gaps reduces mismatches
- Permitting the insertion of arbitrarily many gaps can lead to high scoring alignments of **non-homologous** sequences.
- Penalizing gaps forces alignments to have relatively few gaps.

Gap Penalties

- How to balance gaps with mismatches?

- Gaps must get a steep penalty, or else you'll end up with nonsense alignments.

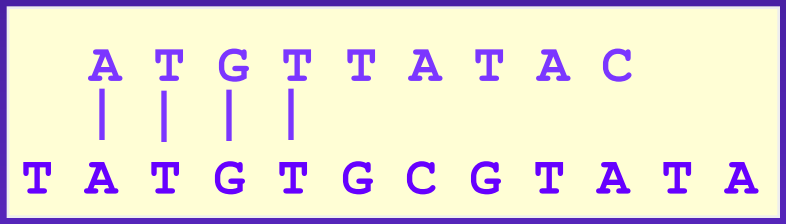
- In real sequences, muti-base (or amino acid) gaps are quit common
  - genetic insertion/deletion events

- "Affine" gap penalties give a big penalty for each new gap, but a much smaller "gap extension" penalty.

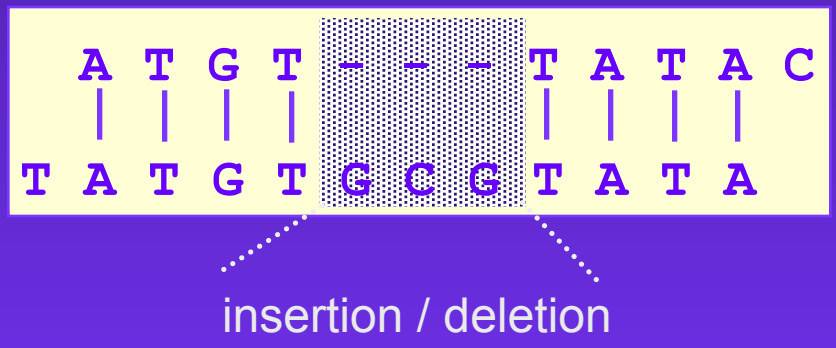
Scoring Insertions and Deletions

match = 1
mismatch = 0

Total Score: 4



Total Score: 8 - 3.2 = 4.8



Gap parameters:

d = 3 (gap opening)

e = 0.1 (gap extension)

g = 3 (gap lenght)

$\gamma(g)$ = -3 - (3 -1) 0.1 = -3.2

Modification of Gap Penalties

Score Matrix: BLOSUM62

gap opening penalty = 3
gap extension penalty = 0.1
score = **6.3**

```
1  ...VLSPADKFLTNV 12
       ||||
1  VFTELSPAKTV.... 11
```

gap opening penalty = 0
gap extension penalty = 0.1
score = **11.3**

```
1  V...LSPADKFLTNV 12
   |    ||||  |   | |
1  VFTELSPA.K..T.V 11
```