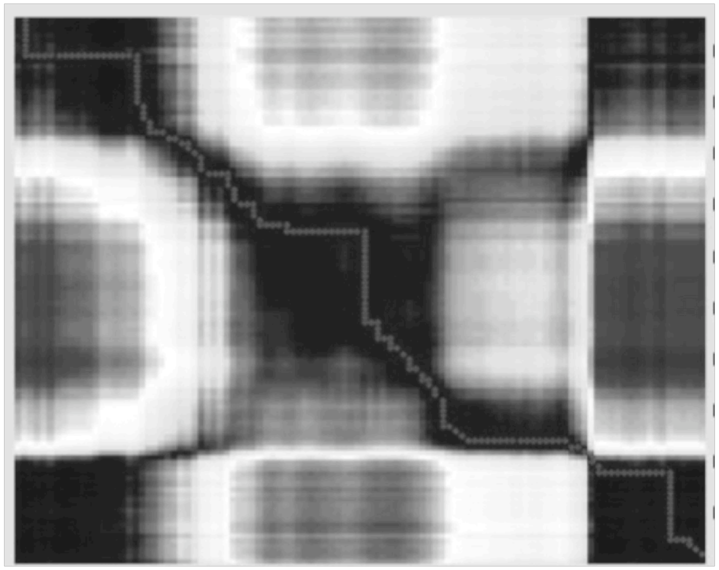
# Dynamic Programming for Pairwise Alignment

Professor Gail L. Rosen



```
GAATTCAG
| | || |
GGA-TC-G
```

# Dynamic Programming invented around WWII

1950: Allocate missiles to reflect maximum damage to targets – multistage solution needed

Solve:

$$E(D) = \sum_{i=1}^{N} p_i(S_i)V_i$$

Subject to:

$$\sum_{i=1}^{N} S_i = S$$

$$0 \leq S_i \leq S, \qquad i = 1, 2, \ldots, N$$

$V_i$: $i_{th}$ target

$S_i$: # of S missiles allocated to $i_{th}$ target

$p_i\,(S_i)$: Probability that $i_{th}$ target will be destroyed by $S_i$ missiles

# Dynamic Time Warping: Optimization for Aligning two time signals

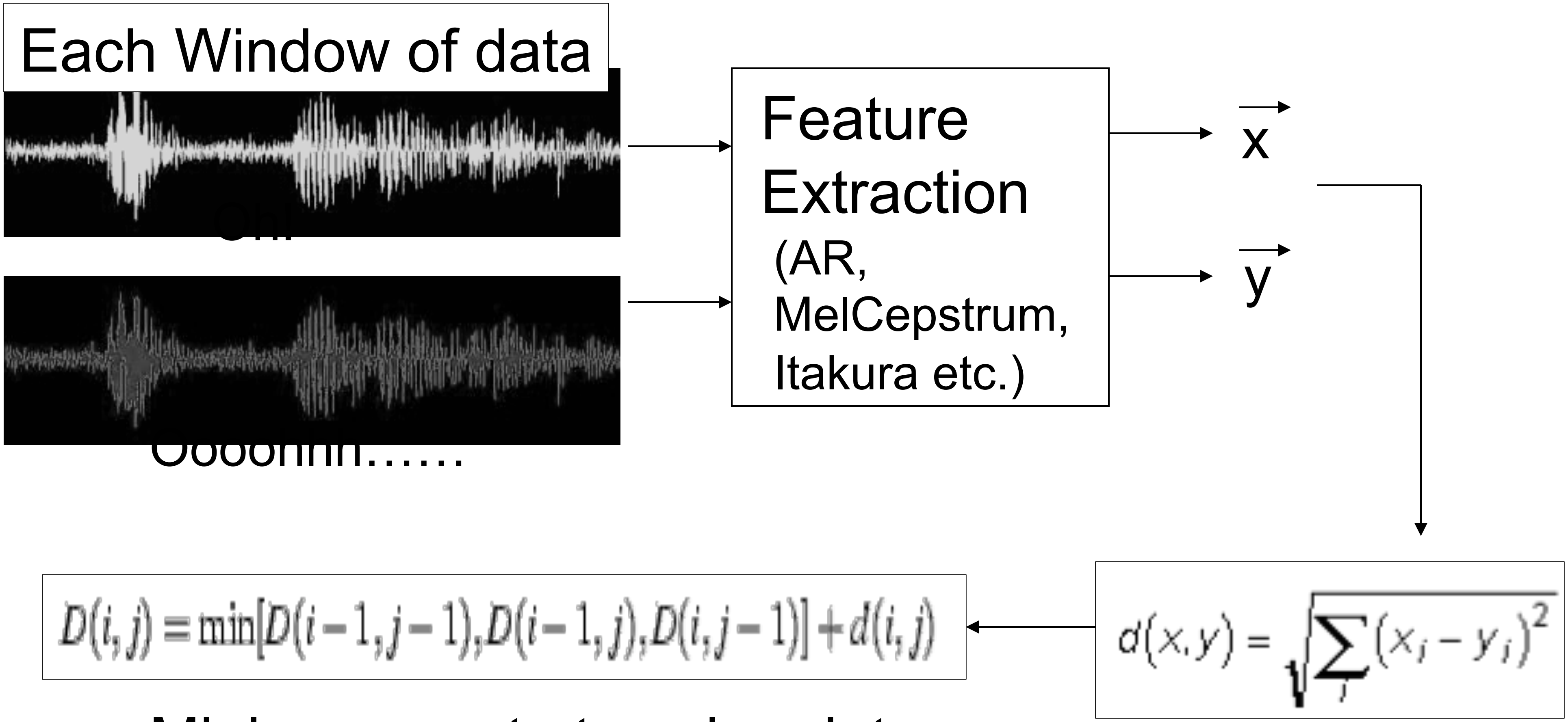
- ✓Published by Vintsyuk in 1968
- ✓Optimized for Speech in 1978

Nice Matlab example:
http://labrosa.ee.columbia.edu/matlab/dtw/

# Same algorithm – different optimization constraints

- ✓ DTW: minimize cost function (usually of distance measure between signals)
- ✓ DNA alignment: maximize similarity score between DNA sequences

# DTW for Speech



Each Window of data

Oh!

Ooooohhh……

Feature Extraction (AR, MelCepstrum, Itakura etc.)

$\vec{x}$

$\vec{y}$

$$d(x,y) = \sqrt{\sum_i (x_i - y_i)^2}$$

Distance Measure

$$D(i,j) = \min[D(i-1,j-1), D(i-1,j), D(i,j-1)] + d(i,j)$$

Minimum cost at each point

# Needleman-Wunsch Algorithm (DP for DNA)

✓1970: Local pairwise similarity

Each Window of data

GAATTCAG $\vec{x}$

GGATCG $\vec{y}$

Distance Measure:
Reward (Score) for match
Penalty for mismatch or indel

$$D(i,j) = \max\left[D(i-1,j-1), D(i-1,j), D(i,j-1)\right] + d(i,j)$$

Maximum score at each point

```
GAATTCAG
|  | || |
GGA-TC-G
```

# Three different point-mutations

✓Mismatch/Substitution

✓Insertion

✓Deletion

"Gap-insertion"
Or just plain "gap"

# Different scoring rules -> different alignments

Dynamic programming: global alignment

Match=5, mismatch = -4, gap  = -2

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

Score = 8*5 + 0 + 6*-2 = 28
```

# Scoring rules/matrices

- ✓ Why are they important?
  - ✓ Choice of scoring rule can dramatically influence the sequence alignments obtained and, therefore, the analysis being done
  - ✓ Different scoring matrices have been developed for different situations; using the wrong one can make a big difference.
- ✓ What do they mean?
  - ✓ Scoring matrices implicitly represent a particular theory of evolution
  - ✓ Elements of the matrices specify relationships between amino acid residues or nucleotides

# Substitution Matrice

Log-odds ratio -> log likelihood ratio that the pair (a,b) is related vs unrelated (depends on scoring matrices)

The alignment score is the log likelihood that the sequences have common ancestry

# Alignment Scores measure likelihood of common ancestor

$$\mathbf{a} = [a_1 a_2 .. a_N] \quad \mathbf{b} = [b_1 b_2 .. b_N]$$

Two DNA sequences

$$p_{\mathbf{a}} p_{\mathbf{b}} = \prod_i p_{a_i} \prod_i p_{b_i}$$

sequences are independent at each position i

$$q_{\mathbf{a},\mathbf{b}}$$

sequences have joint probability

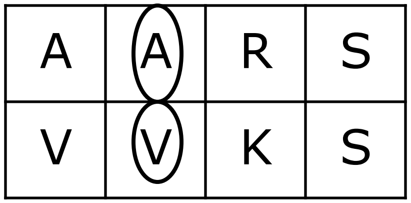
$$\prod_i \frac{q_{a,b_i}}{p_{a_i} p_{b_i}}$$

Odds-Ratio

# Substitution Matrix Calculations

✓ p's are background independent frequencies

Probability of *a* occurring in a position in one sequence

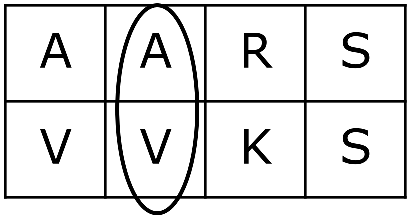


| | | | |
|---|---|---|---|
| A | A | R | S |
| V | V | K | S |

We need scoring terms for each aligned residue pair
Models: Random model (R): letter a occurs with frequency $p_a$

✓ q's are joint probabilities

Probability of A and V occurring in both sequences jointly



| | | | |
|---|---|---|---|
| A | A | R | S |
| V | V | K | S |

Models: Match model (M): aligned pairs of residues have joint probability $p_{ab}$
$p_{ab}$=probability that a and b came from common ancestor residue

# Log-Odds Ratio

$$\sum_i \underbrace{log\left(\frac{q_{a,b_i}}{p_{a_i}p_{b_i}}\right)}$$

Adding log-odds substitution scores gives the log-odds of the alignment

Substitution Matrix Score

* Positive if probability of alignment is greater than chance

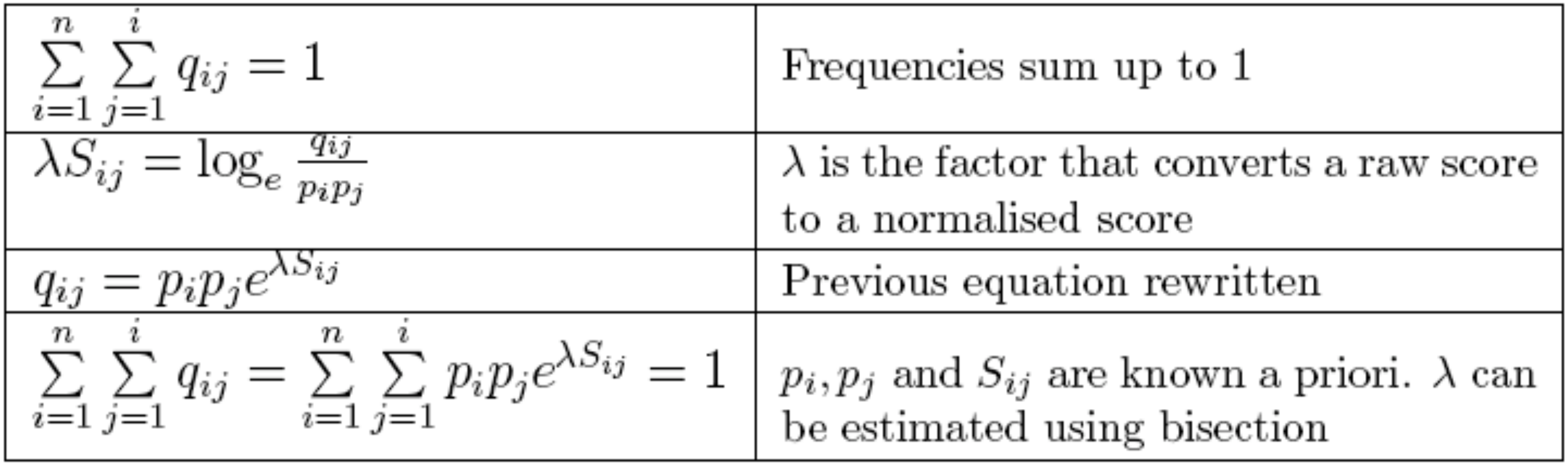
* Negative if probability of alignment is less than chance

# GAP Penalties – mostly heuristic

- ✓Substitutions can be derived
- ✓Gaps are usually heuristic and can vary
  - ✓ Linear
  - ✓ Extension

# Can get “normalized score” – estimate lambda

$$S_{ij} = \log \frac{q_{ij}}{p_i p_j}$$



| | |
|---|---|
| $\sum_{i=1}^{n} \sum_{j=1}^{i} q_{ij} = 1$ | Frequencies sum up to 1 |
| $\lambda S_{ij} = \log_e \frac{q_{ij}}{p_i p_j}$ | $\lambda$ is the factor that converts a raw score to a normalised score |
| $q_{ij} = p_i p_j e^{\lambda S_{ij}}$ | Previous equation rewritten |
| $\sum_{i=1}^{n} \sum_{j=1}^{i} q_{ij} = \sum_{i=1}^{n} \sum_{j=1}^{i} p_i p_j e^{\lambda S_{ij}} = 1$ | $p_i, p_j$ and $S_{ij}$ are known a priori. $\lambda$ can be estimated using bisection |

# Mathematical Formulation of E-value

$$\mathbf{E} = \mathbf{m} \cdot \mathbf{n} \cdot \mathbf{p}^{\mathrm{S}}$$

(Expected number of matches)

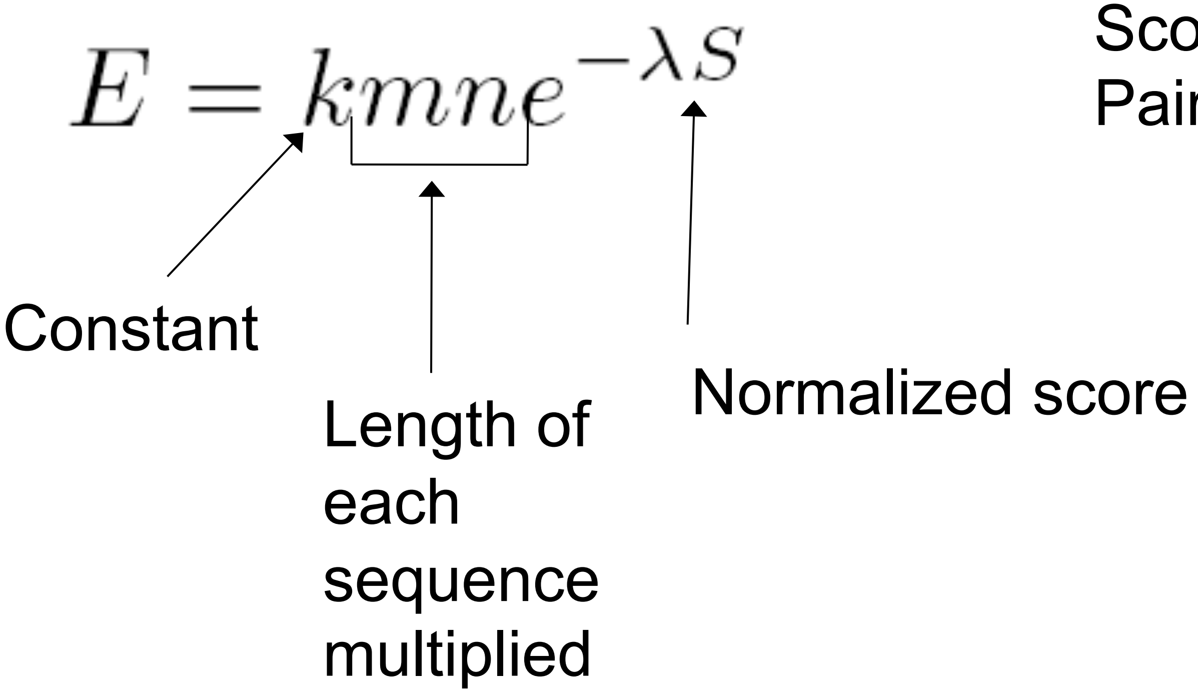
P: probability of match
S: number of matches

We are not doing a simple string match. Need more complicated

# Ultimate expression

expected number of HSPs with score at least *S is given by the formula*

HSP= Highest Scoring Pairs



$$E = k\underbrace{mn}e^{-\lambda S}$$

Constant

Length of each sequence multiplied

Normalized score

# Relation of E value to Number of matches occurring by chance

$$P(X = x) = e^{-E} \cdot \frac{E^x}{x!}$$

the chance of finding zero HSPs with score >=*S is* $e^{-E}$

P-VALUE: Prob of finding at least one HSP by chance:

$$P = 1 - e^{-E}$$

# BLAST

✓ Reports E-value rather than P-value

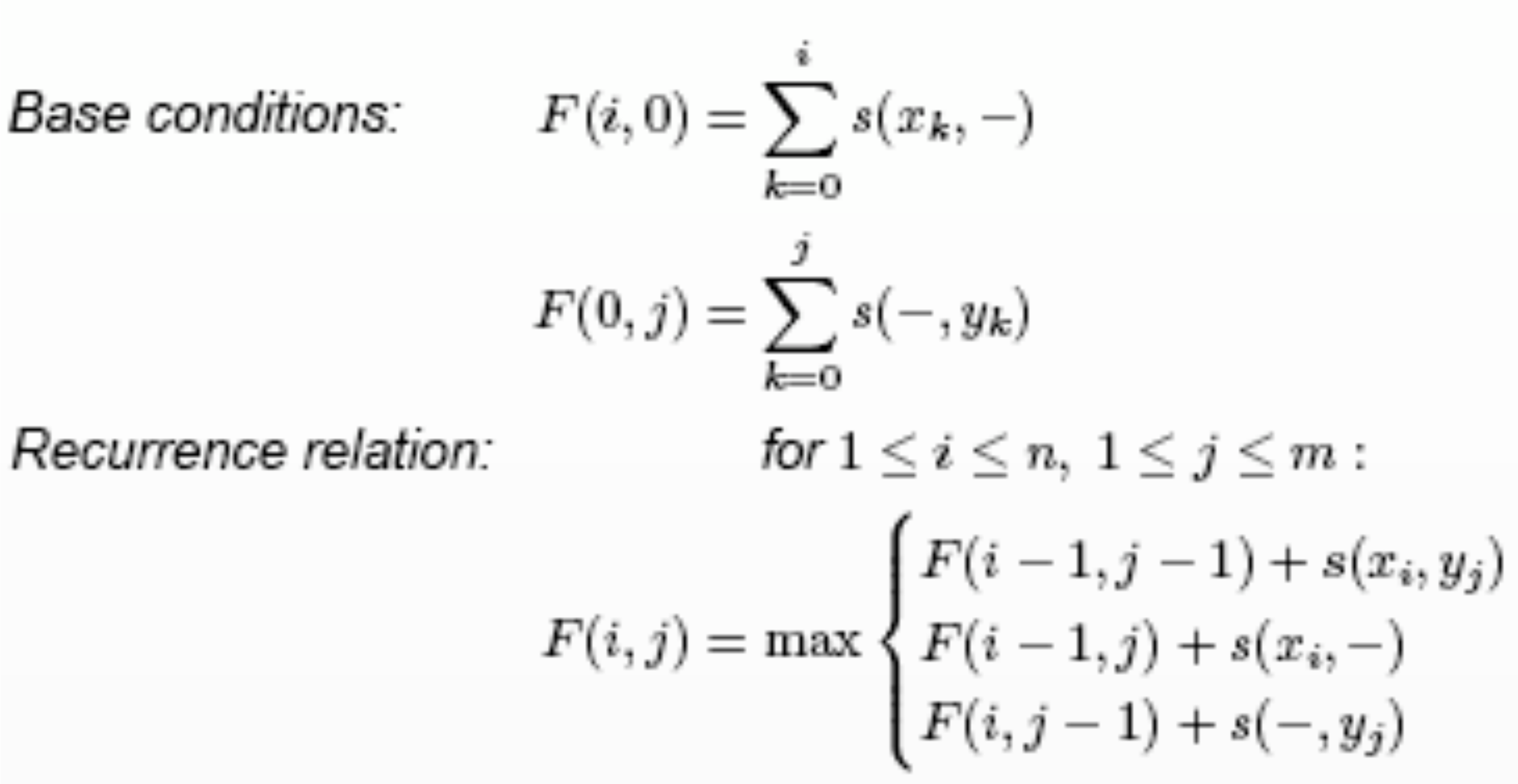
# Have Scoring and Gap-penalty values – Now algorithms to align

- ✓Global: Needleman-Wunsch algorithm
- ✓Local: Smith-Waterman algorithm
- ✓Multi: CLUSTAL
- ✓Fast search: BLAST

# Needleman-Wunsch Global Alignment

- ✓ Say we have a substitution/gap scoring scheme
- ✓ How do we do an alignment?

# Needleman-Wunsch Algorithm



*Base conditions:*

$$F(i,0) = \sum_{k=0}^{i} s(x_k, -)$$

$$F(0,j) = \sum_{k=0}^{j} s(-, y_k)$$

*Recurrence relation:* for $1 \leq i \leq n,\ 1 \leq j \leq m:$

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

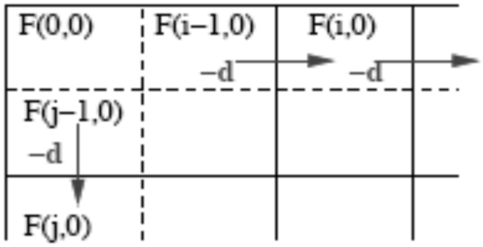
**Base condition:** Only way to align first $i$ elements of sequence $x$ with 0 elements of $y$: align each element with a space (gap) in $y \Rightarrow$ score for one element is $s(x_k, -) \Rightarrow$ total score is $F(i, 0) = \sum_{k=0}^{i} s(x_k, -)$.

**Recurrence:** three cases:

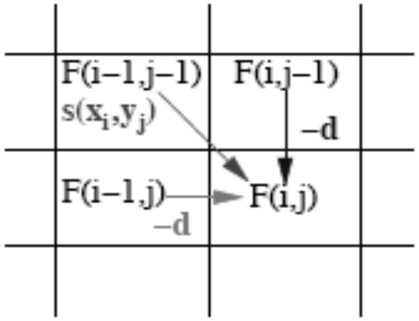
- **Aligning $x_i$ with $y_j$** $\Rightarrow$ total score is $s(x_i, y_j)$ plus score of aligning $i-1$ elements of $x$ with $j-1$ elements of $y$: $F(i-1, j-1) + s(x_i, y_j)$.
- **Aligning $x_i$ with space in $y$:** $s(x_i, -)$ plus score of aligning $i-1$ elements of $x$ with $j$ elements of $y$: $F(i-1, j) + s(x_i, -)$.
- **Aligning $y_j$ with space in $x$:** $F(i, j-1) + s(-, y_j)$ by analogy.

Starting from $F(0,0) = 0$, systematically fill whole matrix $(F)_{ij}$:

for $i = 0$ or $j = 0$, calculate new value from left-hand (upper) value.

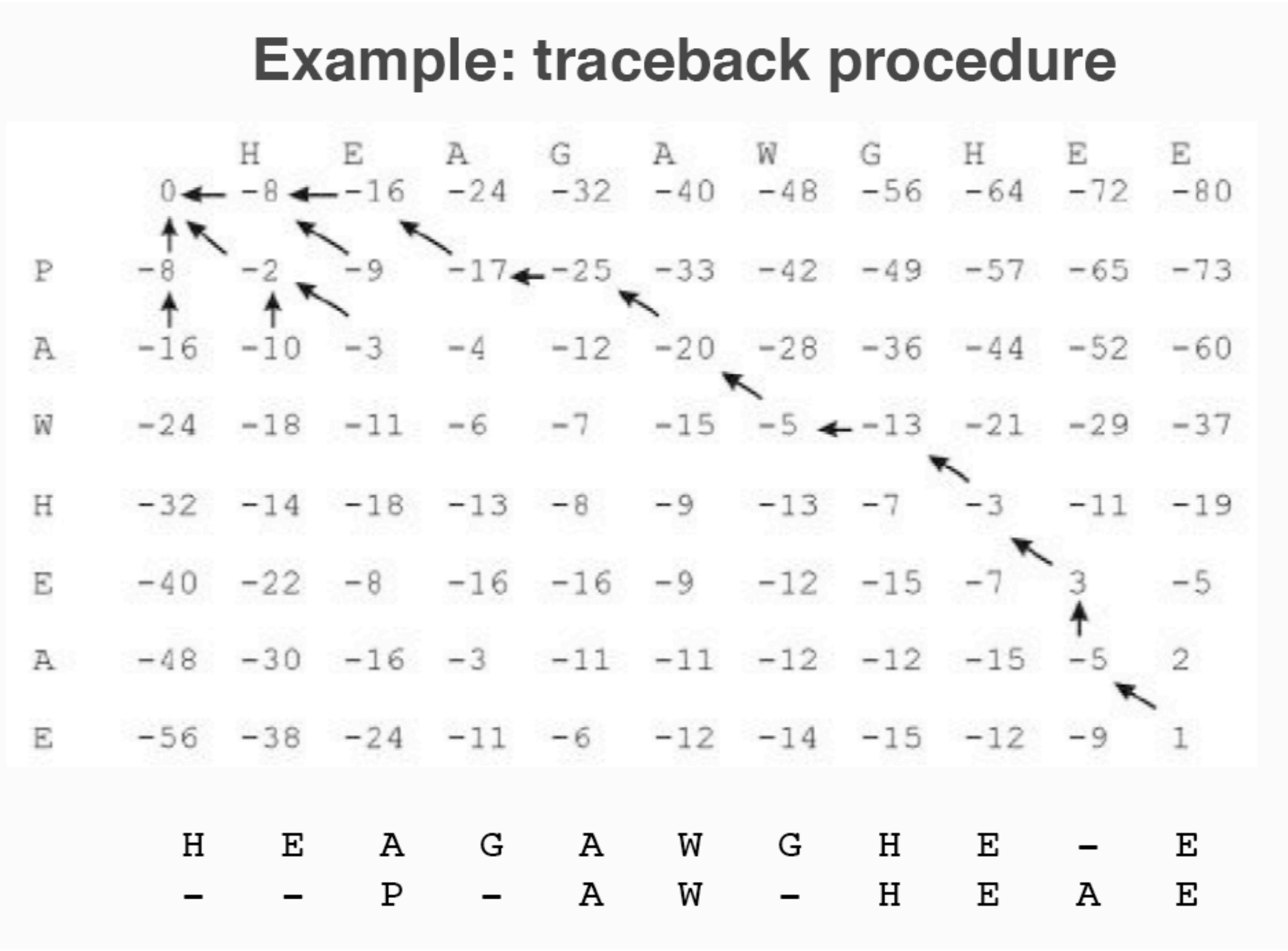


for $i, j \geq 1$, calculate the bottom right-hand corner of each square of 4 cells from one of the 3 other cells:



keep pointer back to the cell from which it was derived
$\Rightarrow$ **traceback pointer**.

# Example Traceback Path



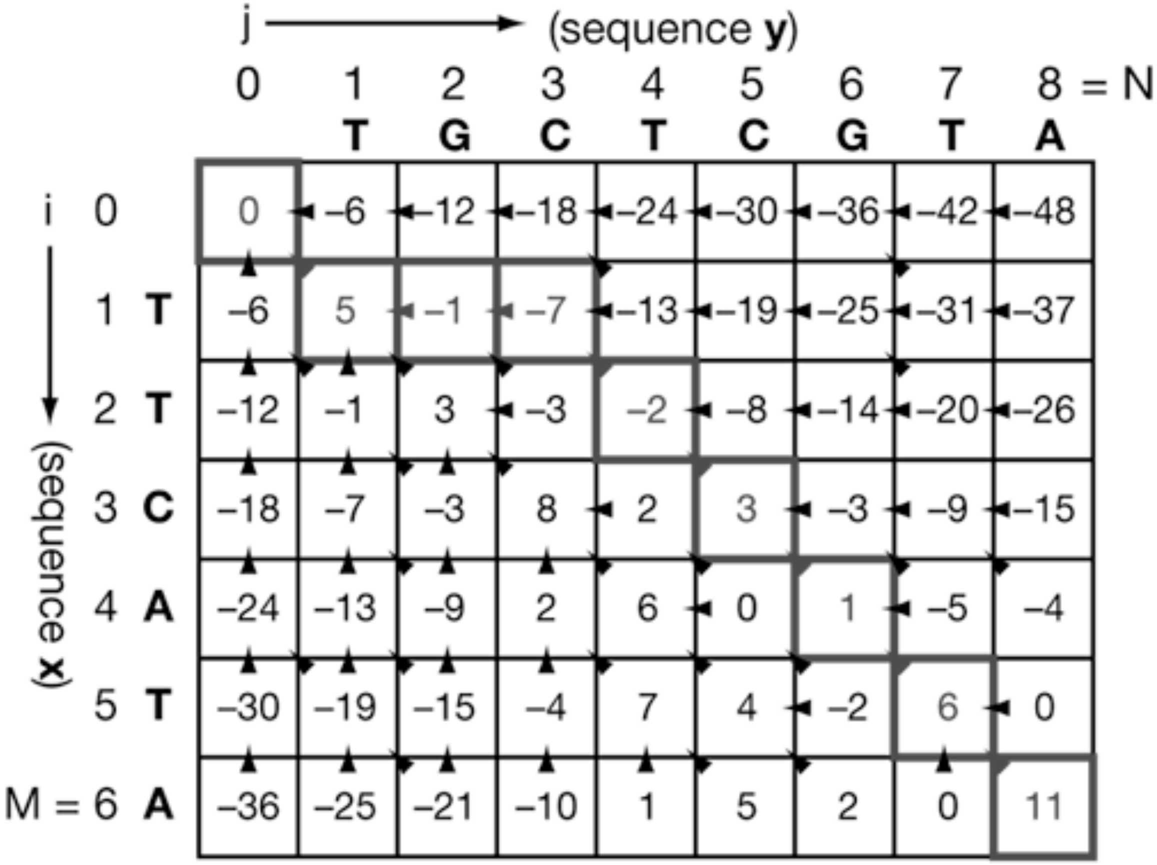
## Example: traceback procedure

|   |   | 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 |

|   |   |   |   |   |   |   |   |   |   |   |
|---|---|---|---|---|---|---|---|---|---|---|
| H | E | A | G | A | W | G | H | E | - | E |
| - | - | P | - | A | W | - | H | E | A | E |

# Closer Look at Traceback

Dynamic programming matrix:



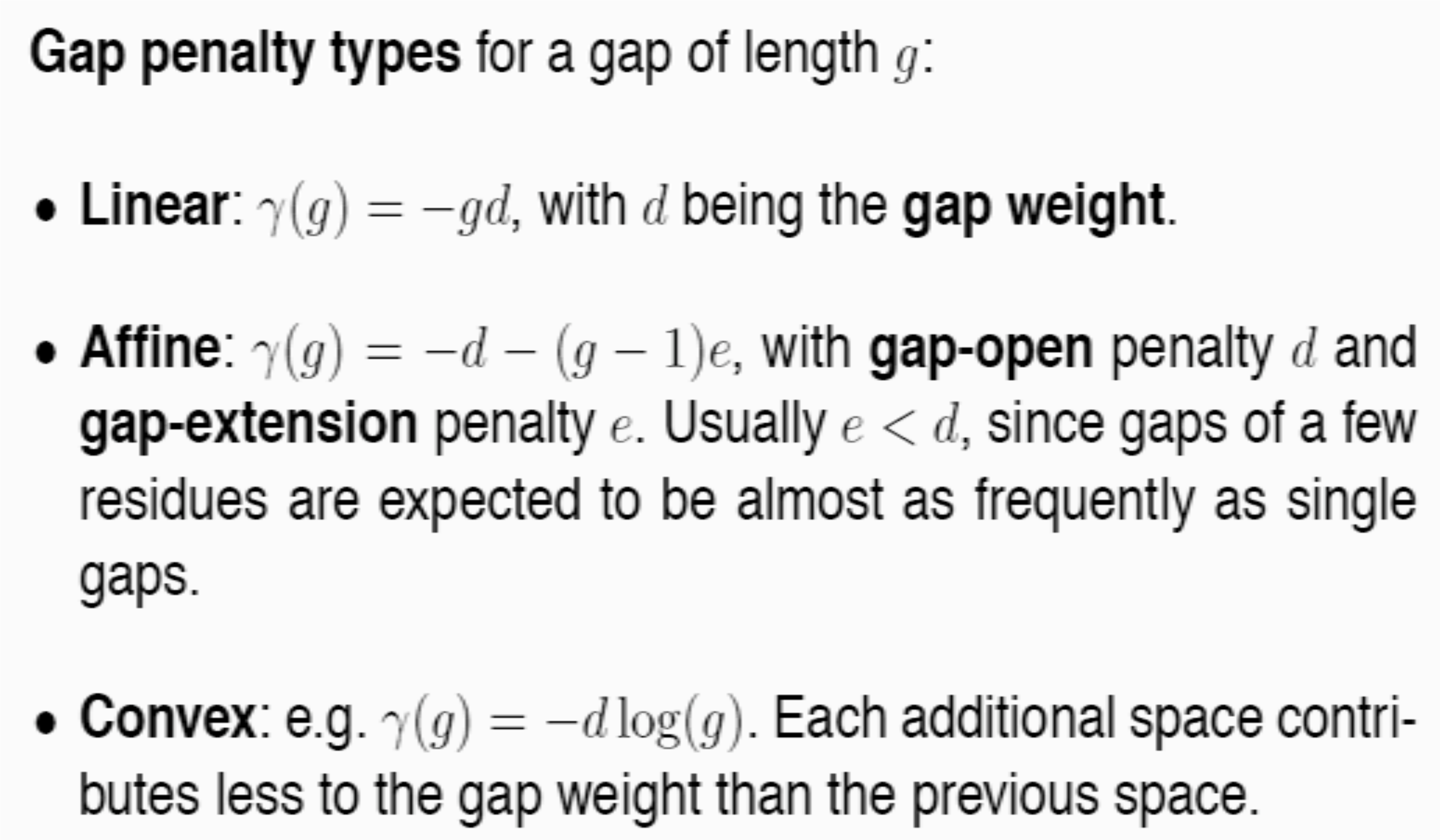
| i | | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 = N |
|---|---|---|---|---|---|---|---|---|---|---|
| | | | T | G | C | T | C | G | T | A |
| 0 | | 0 | –6 | –12 | –18 | –24 | –30 | –36 | –42 | –48 |
| 1 | T | –6 | 5 | –1 | –7 | –13 | –19 | –25 | –31 | –37 |
| 2 | T | –12 | –1 | 3 | –3 | –2 | –8 | –14 | –20 | –26 |
| 3 | C | –18 | –7 | –3 | 8 | 2 | 3 | –3 | –9 | –15 |
| 4 | A | –24 | –13 | –9 | 2 | 6 | 0 | 1 | –5 | –4 |
| 5 | T | –30 | –19 | –15 | –4 | 7 | 4 | –2 | 6 | 0 |
| M = 6 | A | –36 | –25 | –21 | –10 | 1 | 5 | 2 | 0 | 11 |

Optimum alignment scores 11:

```
 T   –   –   T   C   A   T   A
 T   G   C   T   C   G   T   A
+5  –6  –6  +5  +5  –2  +5  +5
```

scoring system of +5 for a match, -2 for a mismatch and -6 for each insertion or deletion.The cells in the optimum path are shown in red. Arrowheads are 'traceback pointers,' indicating which of the three cases were optimal for reaching each cell. (Some cells can be reached by two or three different optimal paths of equal score)

# Gap Penalties



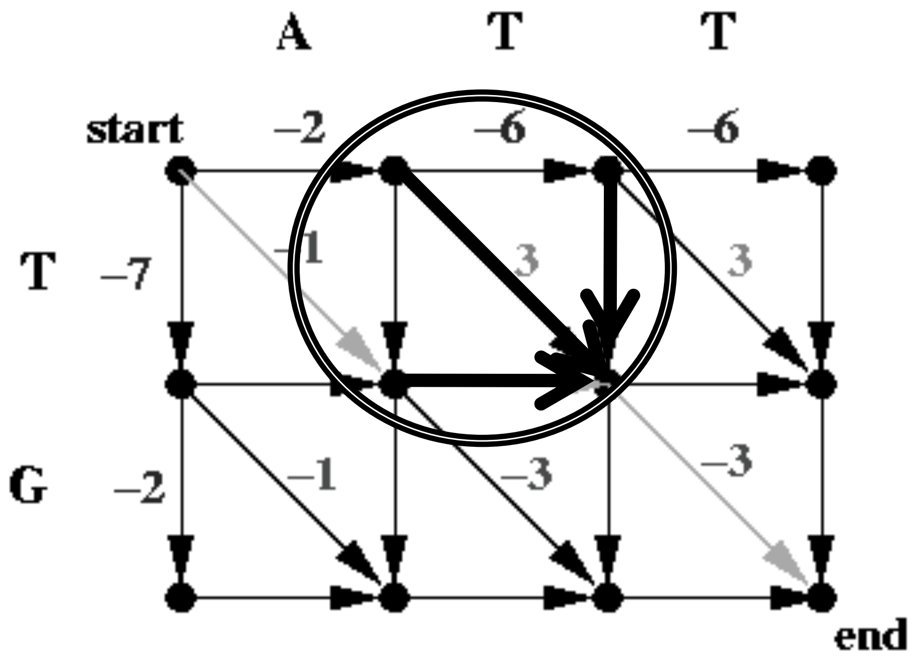
**Gap penalty types** for a gap of length $g$:

- **Linear**: $\gamma(g) = -gd$, with $d$ being the **gap weight**.

- **Affine**: $\gamma(g) = -d - (g-1)e$, with **gap-open** penalty $d$ and **gap-extension** penalty $e$. Usually $e < d$, since gaps of a few residues are expected to be almost as frequently as single gaps.

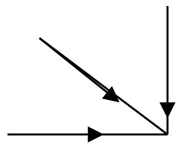
- **Convex**: e.g. $\gamma(g) = -d\log(g)$. Each additional space contributes less to the gap weight than the previous space.

# Illustration of the max operator



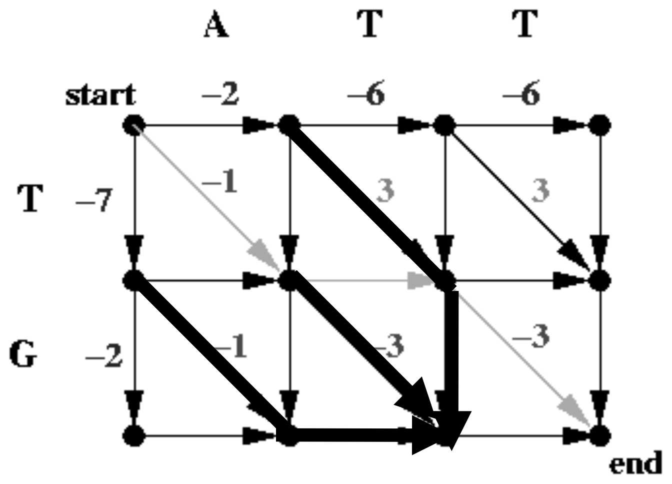
# Enhancements

Instead of



$$D(i,j) = \max \{ D(i-1,j), D(i-1,j-1), D(i,j-1) \}$$

✓ "Fancier" Traceback/Alignment function



Because a straight warp is desired, a path constraint was placed to disallow two non-diagonal paths to occur in sequence. The following was implemented in the final design:

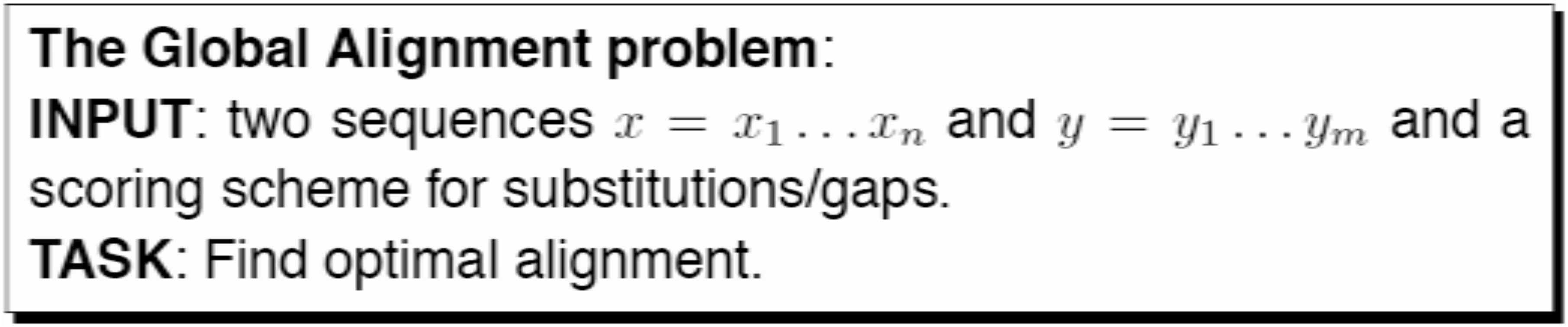
$$D(i,j) = \max \{ D(i-1,j-2), D(i,j-1), D(i-1,j-1), D(i-1,j), D(i-2,j-1), \}$$

# Global vs. Local Alignment

✓Global does alignment over all sequence

✓Local aligns smaller subsequences

# Global Alignment Problem

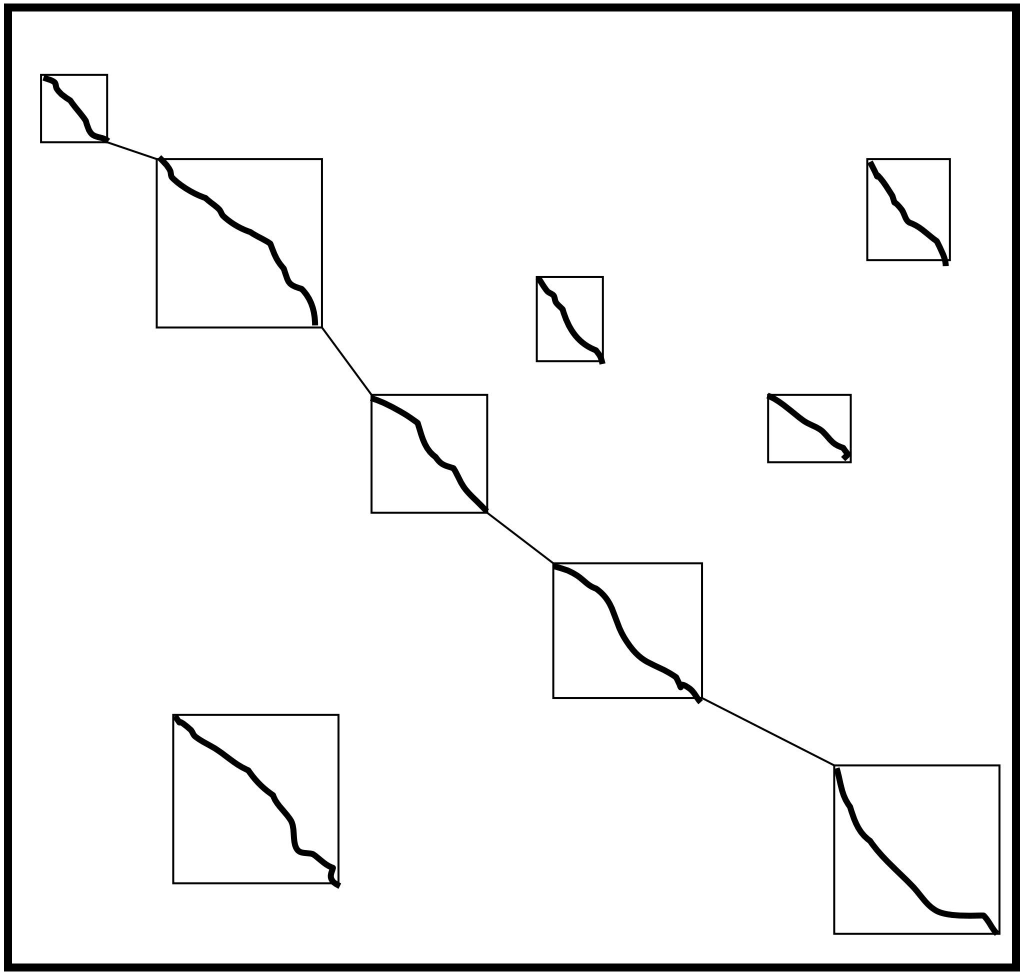


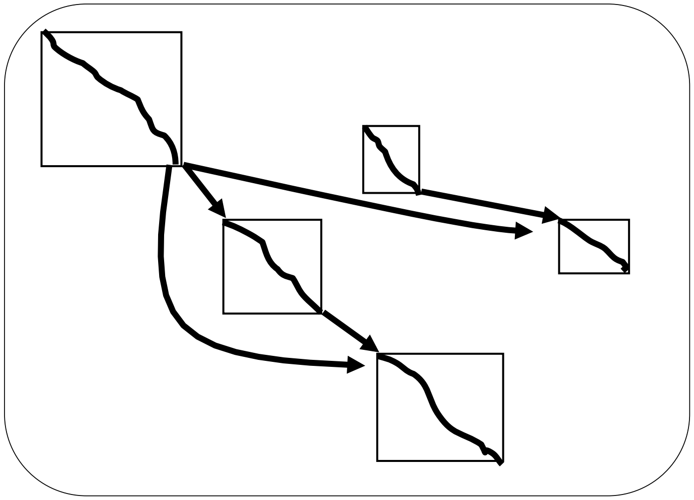
**The Global Alignment problem**:
**INPUT**: two sequences $x = x_1 \ldots x_n$ and $y = y_1 \ldots y_m$ and a scoring scheme for substitutions/gaps.
**TASK**: Find optimal alignment.

# N-W vs. S-W:  major differences

- ✓S-W limits minimum values at 0
- ✓N-W must start trace-back in lower-right hand corner (global)
- ✓S-W starts trace-back at highest number and goes to “diagonal 0”

# Local Alignments (can have connection to global)





Each local alignment has a weight

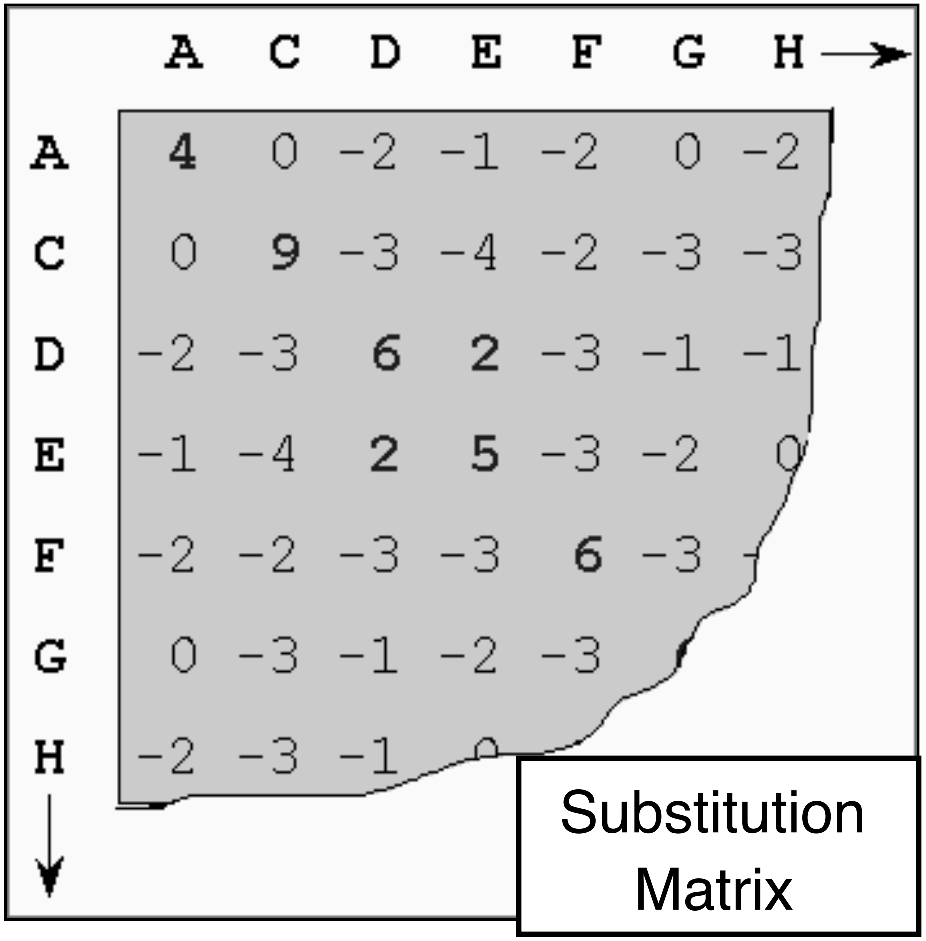
FIND the chain with highest total weight

# Smith-Waterman Algorithm for Local Alignment

- ✓ Smith-Waterman truncates all negative scores to 0, with the idea being that as the alignment score gets smaller, the local alignment has come to an end.

Two examples

# Log-odds into Matrix



|   | A | C | D | E | F | G | H |
|---|---|---|---|---|---|---|---|
| A | **4** | 0 | -2 | -1 | -2 | 0 | -2 |
| C | 0 | **9** | -3 | -4 | -2 | -3 | -3 |
| D | -2 | -3 | **6** | **2** | -3 | -1 | -1 |
| E | -1 | -4 | **2** | **5** | -3 | -2 | 0 |
| F | -2 | -2 | -3 | -3 | **6** | -3 |  |
| G | 0 | -3 | -1 | -2 | -3 |  |  |
| H | -2 | -3 | -1 | 0 |  |  |  |

Substitution Matrix

- **–Two most used matrices are:**
  - **–PAM (Percent Accepted Mutation)**
    - **–Based on explicit evolutionary model**
    - **–Represents a specific evolutionary distance**
  - **–BLOSUM (BLOck SUbstitution Matrices)**
    - **–Based on empirical frequencies**
    - **–Always a blend of distances as seen in protein databases**

# PAM (Percent Accepted Mutations) # - (# of mutations out of 100 AAs)

  - ✓ Based on explicit evolutionary model
  - ✓ PAM-1 is a scoring system for sequences in which 1% of the residues have undergone mutation: 1 pair in a 100 residue segment
  - ✓ PAM-250 represents 250% mutation, i.e., an average of 2.5 accepted mutation per residue (multiple mutations per pair) a very distant relationship

- ✓ Important to remember:
  - ✓ A value less than 0 or greater than 0 indicates that the frequency is less than or greater than that expected by chance, respectively
  - ✓ Smaller-numbered matrices correspond to closely related sequences
  - ✓ Larger-numbered matrices correspond to more distantly related sequences

# Problems with PAM

- ✓ **PAM** model assumes all residues are equally mutable (**mutation for A←→G,T←→C are more likely than A←→T, G←→C )**
- ✓ Model devised using the most mutable positions rather than the most conserved positions, i.e., those that reflect chemical and structural properties of importance
- ✓ Derived from a biased set of sequences: small globular proteins available in the database in 1978

# BLOSUM vs. PAM

✓Most popular BLOSUM 62

# BLOSUM (# -- use seqs less than #% identical)

- ✓ Important to remember:
  - ✓ matrices constructed using multiple alignments of evolutionarily divergent but highly conserved proteins
  - ✓ Every possible identity or substitution is assigned a score based on its observed frequences in the alignment of related proteins.
  - ✓ Pairs that are more likely than chance will have positive scores, and those less likely will have negative scores
  - ✓ Larger-numbered matrices correspond to more recent (less) divergence
  - ✓ Smaller-numbered matrices correspond to more distantly related sequences
  - ✓ By default,  BLOSUM62 is often used

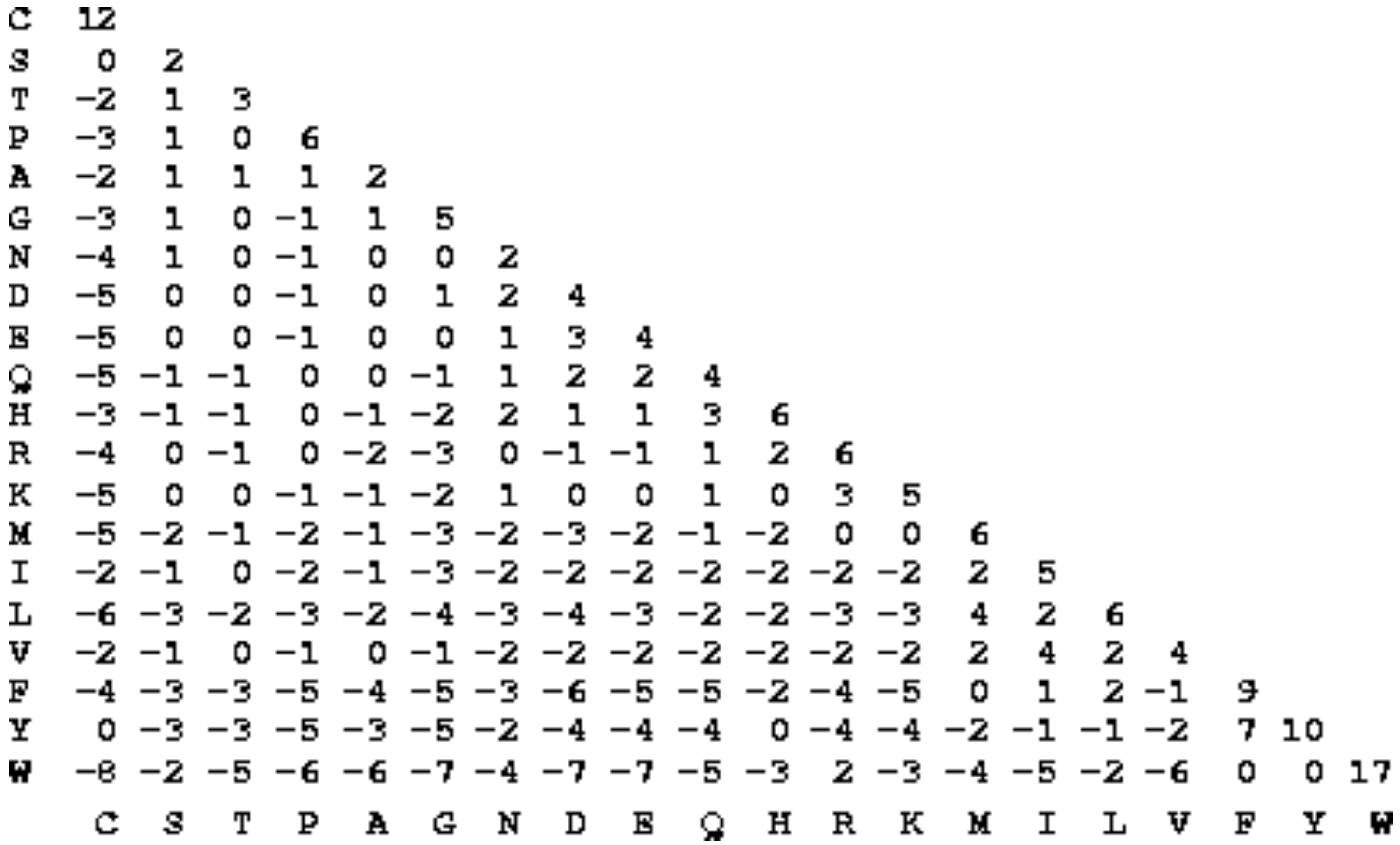
# Two major scoring matrices

- ✓PAM = accepted point mutation
  - ✓71 trees with 1572 accepted mutations, sequences with >85% identity
  - ✓PAM1 means average of 1% change over all amino acids
  - ✓1 PAM = 10my evolutionary distance
- ✓BLOSUM = Blocks substitution matrices
  - ✓Based on BLOCKS database (Henikoff & Henikoff, 1992) of over 2000 conserved amino acid patterns in over 500 proteins

# Example: PAM Matrices

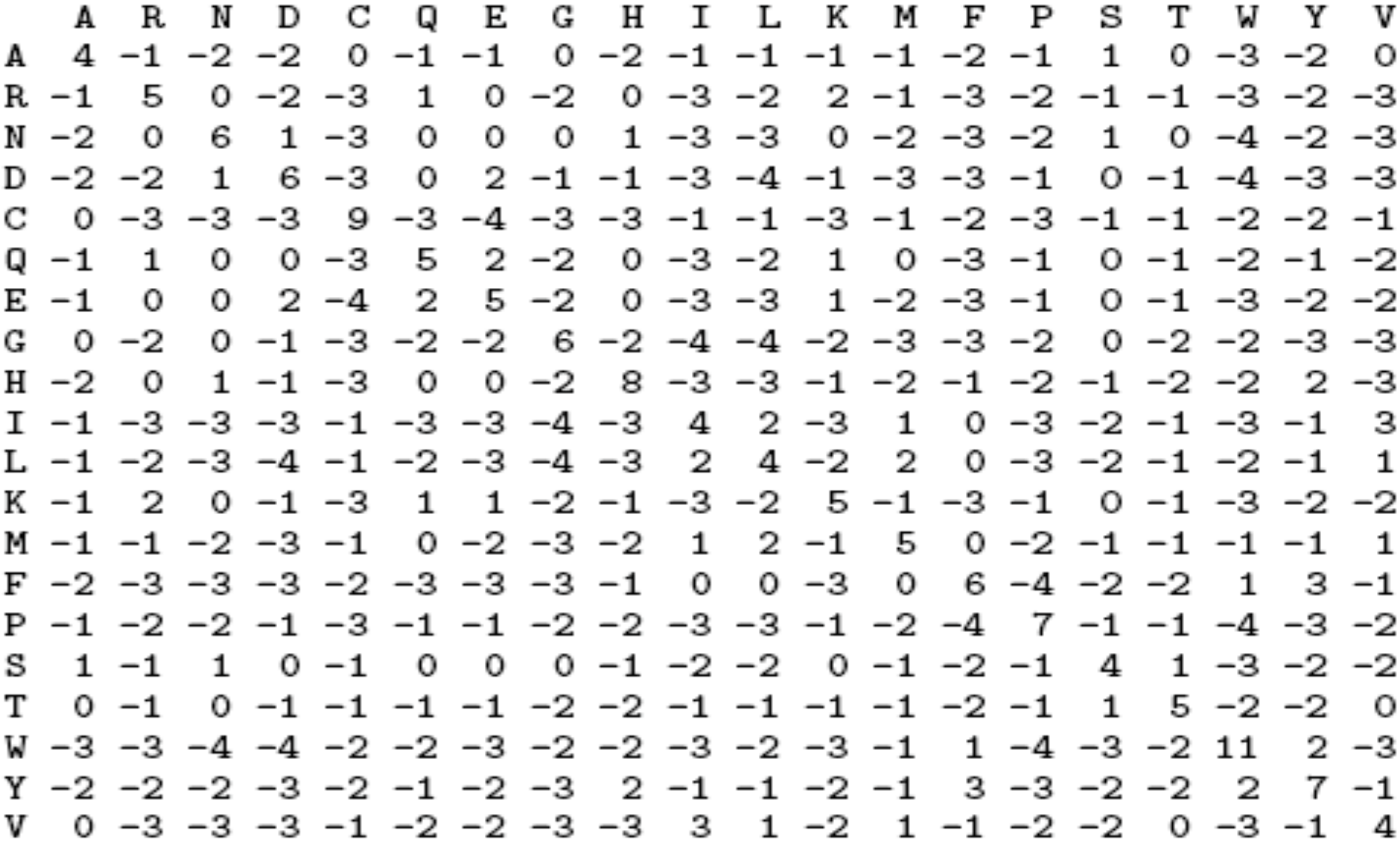
- ✓Point Accepted Mutations
  - ✓ For Amino Acid alignment
  - ✓ Biased by differing rates of mutation in different protein families
- ✓Train on evolutionary alignments
- ✓Develop Mutation Probability Matrix (MPM) that:
  - ✓ has likelihood of sequence b being replaced by a on off-diagonal
  - ✓ Has no residue change on diagonal

# Example PAM Substitution matrix



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

# Example Substitution Matrix for scoring: BLOSUM Matrix



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

Figure 9: BLOSUM62 substitution matrix

# Alignment Summary

- ✓Distance rewards matches and penalizes substitutions/gaps
- ✓Calculate all paths that create an alignment
- ✓Find the optimal path for the alignment
- ✓Applications:  Find similar sequences (whether different, shortened elongated)

  Phylogenetic Trees (evolution)

# Class Excercises

- ✓ Explore Alignment in Matlab
  - ✓ http://www.mathworks.com/access/helpdesk/help/toolbox/bioinfo/ug/fp35834dup12.html
  - ✓ http://www.mathworks.com/access/helpdesk/help/toolbox/bioinfo/index.html?/access/helpdesk/help/toolbox/bioinfo/ug/bqk11hk-1.html

# In-class Exercise

✓ Comparing Eyeless gene

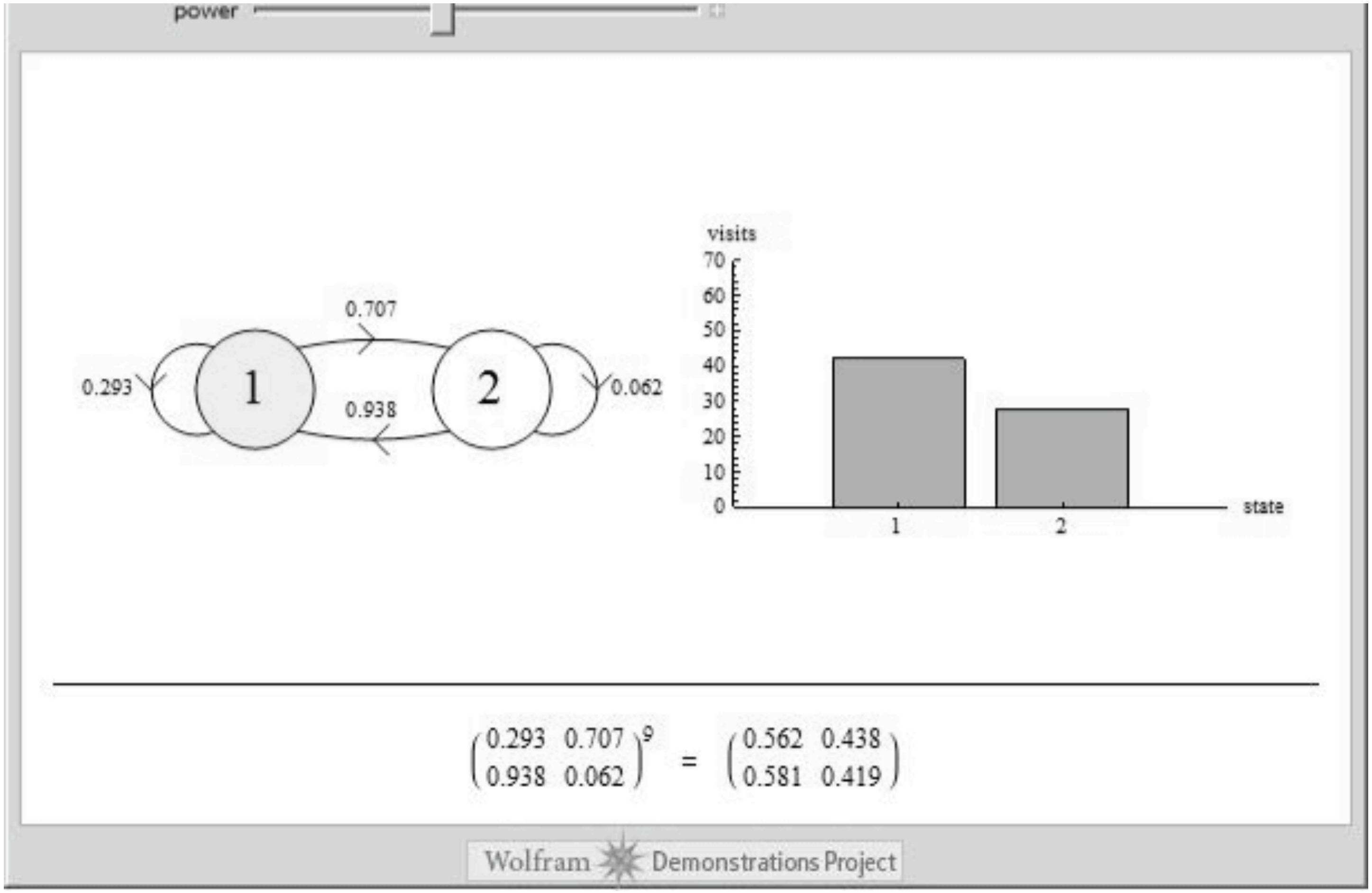
# Class Excercises

- ✓ #  Align bovine insulin precursor (P01317) and human insulin precursor (P01308) using global alignment and the default settings. Submit a printout of the resulting alignment. Give a short description (2-3 sentences) of the characteristics of the alignment produced.
- ✓ # Reduce the gap open penalty to 1.0 and repeat the alignment. Submit a printout of the resulting alignment. Do you see any differences? If so, describe them.

# Class Excercises

- ✓ # Repeat #1 using local alignment. Submit a printout of the resulting alignment. Do you see any differences with the global alignment? If so, describe them.
- ✓ # Perform a global alignment of the human hemoglobin beta chain (P02023) with the hemocyanin A chain from the American tarantula (P14750). Both of these molecules are globins. Submit a printout of the resulting alignment. Give a short description (2-3 sentences) of the characteristics of the alignment produced.

# End for today: Markov Chains next time



$$\begin{pmatrix} 0.293 & 0.707 \\ 0.938 & 0.062 \end{pmatrix}^9 = \begin{pmatrix} 0.562 & 0.438 \\ 0.581 & 0.419 \end{pmatrix}$$