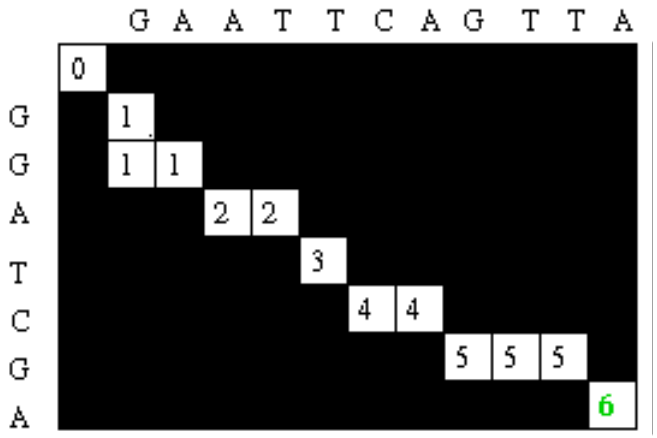
Global Sequence Alignment

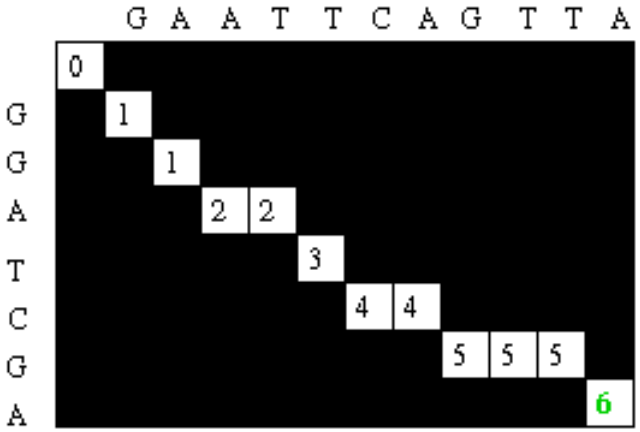
- Needleman-Wunsch-Sellers (**1970**) algorithm.
- Dynamic Programming (DP) based.
  - Overlapping Subproblems
  - Recurrence Relation
  - Table to store solutions to subproblems
  - Ordering of subproblems to fill table
  - Traceback to find solution

Global Alignment: An example

```
V: G A A T T C A G T T A
W: G G A T C G A
```




```
V: G - A A T T C A G T T A
   |   |   |     |       |
W: G G - A – T C – G - - A
```




```
V: G A A T T C A G T T A
   |   |   | |   |     |
W: G G A – T C – G - - A
```

Traceback

		G	A	A	T	T	C	A	G	T	T	A
	0	0	0	0	0	0	0	0	0	0	0	0
G	0	×1	←1	← 1	← 1	← 1	← 1	← 1	×1	← 1	← 1	← 1
G	0	×1	↑1	↑1	↑1	↑1	↑1	↑1	×2	← 2	← 2	← 2
A	0	↑1	↑1	×2	← 2	← 2	← 2	×2	↑2	↑2	↑2	×3
T	0	↑1	← 2	↑2	×3	×3	← 3	← 3	← 3	×3	×3	↑3
C	0	↑1	↑2	↑2	↑3	↑3	×4	← 4	← 4	← 4	← 4	← 4
G	0	↑1	↑2	↑2	↑3	↑3	↑4	↑4	×5	← 5	← 5	← 5
A	0	↑1	↑2	×3	↑3	↑3	↑4	×5	↑5	↑5	↑5	×6

```
V: G A - A T T C A G T T A
   |   |   |   |   |   |
W: G - G A – T C – G - - A
```

Recurrence Relation for Needleman-Wunsch-Sellers

- S[I, J] = MAXIMUM {

  S[I-1, J-1] + δ(V[I], W[J]),

  S[I-1, J] + δ(V[I], —),

  S[I, J-1] + δ(— , W[J])

  }

Generalizations of Similarity Function

- Mismatch Penalty = α
- Spaces (Insertions/Deletions, InDels) = β
- Affine Gap Penalties:

  (Gap open, Gap extension) = (γ,δ)
- Weighted Mismatch = $\Phi(a,b)$
- Weighted Matches = $\Omega(a)$

Types of Sequence Alignments

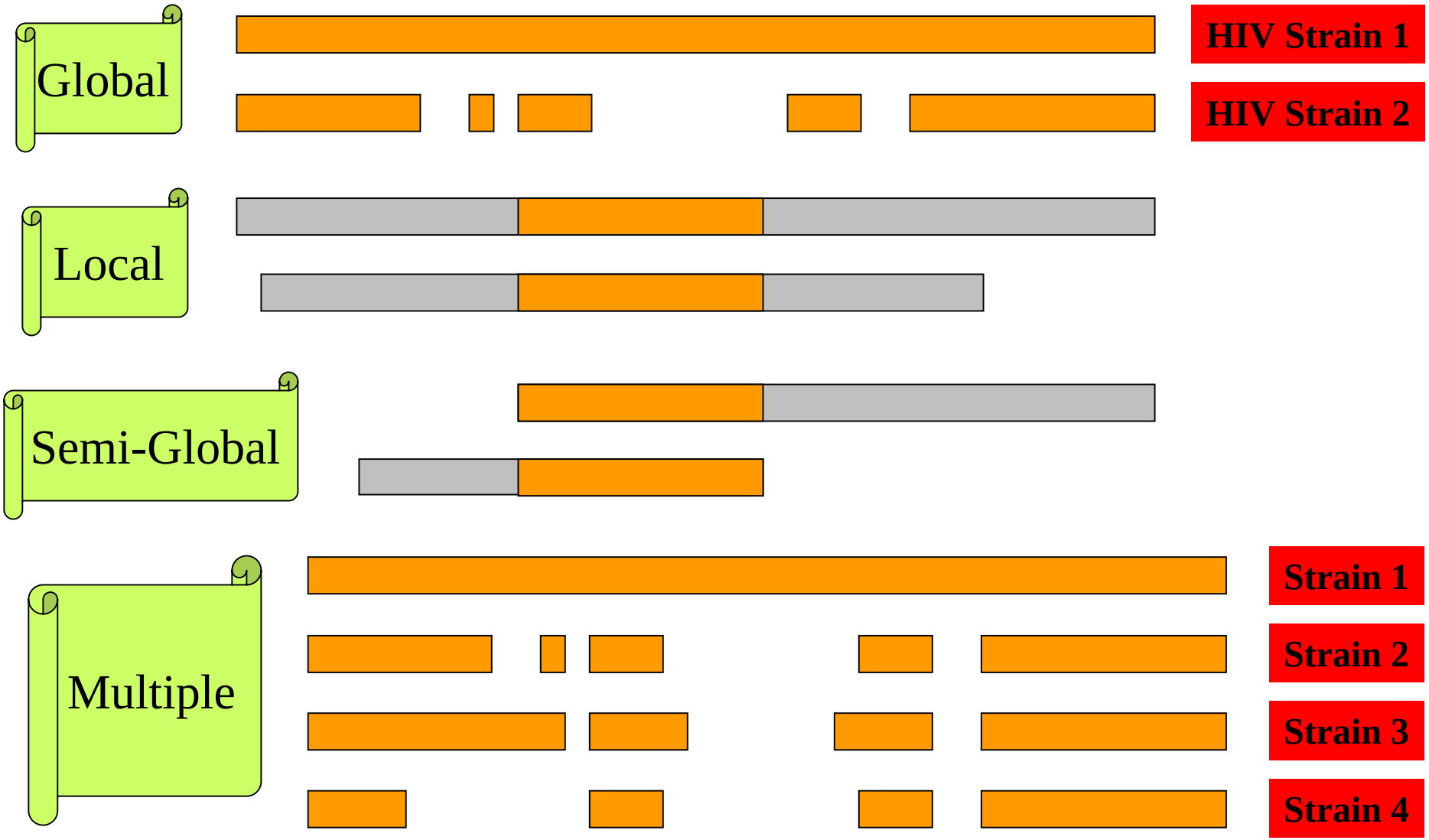
- Global Alignment: similarity over entire length
- Local Alignment: no overall similarity, but some segment(s) is/are similar
- Semi-global Alignment: end segments may not be similar
- Multiple Alignment: similarity between sets of sequences

Global vs Local Alignment

Global alignment

Local alignment

Types of Alignments



Local Sequence Alignment

- Useful when commonality is small and global alignment is meaningless. Often unaligned portions "mask" short stretches of aligned portions. Example: comparing long stretches of anonymous DNA; aligning proteins that share only some motifs or domains.
- Smith-Waterman Algorithm (**1981**)

Recurrence Relations (Global vs Local Alignments)

- S[I, J] = MAXIMUM {

  S[I-1, J-1] + δ(V[I], W[J]),

  S[I-1, J] + δ(V[I], —),

  S[I, J-1] + δ(— , W[J]) }

Global Alignment

- S[I, J] = MAXIMUM { 0,

  S[I-1, J-1] + δ(V[I], W[J]),

  S[I-1, J] + δ(V[I], —),

  S[I, J-1] + δ(— , W[J]) }

Local Alignment

Global Alignment: Example

		T	A	T	A	G	A	A	T	C	T	C
	0	-1	-2	-3	-4	-5	-6	-7	-8	-9	-10	-11
G	-1	-1	-2	-3	-4	-3	-4	-5	-6	-7	-8	-9
A	-2	-2	× 0	-1	-2	-3	-2	-3	-4	-5	-6	-7
T	-3	-1	-1	× 1	0	-1	-2	-3	-2	-3	-4	-5
C	-4	-2	-2	0	0	-1	-2	-3	-4	-1	-2	-3
T	-5	-3	-3	-1	-1	-1	-2	-3	-4	-2	-3	-3

Match +1
Mismatch –1
Gap (-1, -1)

```
V:   T A T A G A A T C T C
         | |   | |
W:   - - - - G A - T C - T
```

Local Alignment: Example

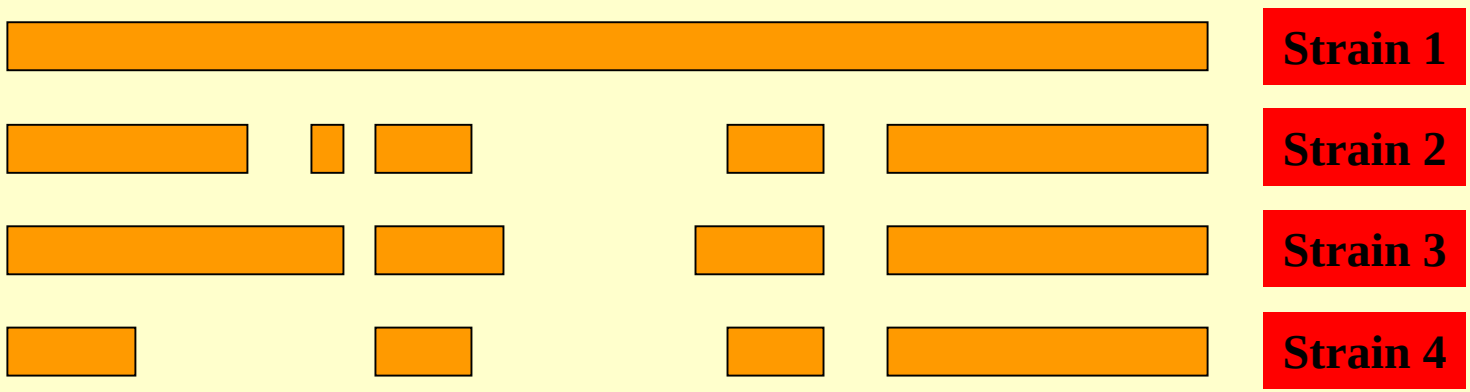
		T	A	T	A	G	A	A	T	C	T	C
	0	0	0	0	0	0	0	0	0	0	0	0
G	0	0	0	0	0	1	0	0	0	0	0	0
A	0	0	1	0	1	0	2	1	0	0	0	0
T	0	1	0	1	0	0	1	1	2	1	1	0
C	0	0	0	0	0	0	0	0	1	3	2	2
T	0	1	0	1	0	0	0	0	0	2	4	3

Match +1
Mismatch –1
Gap (-1, -1)

```
V:    T A T A G A A T C T C
          | |   | | |
W:    - - - - G A - T C T -
```

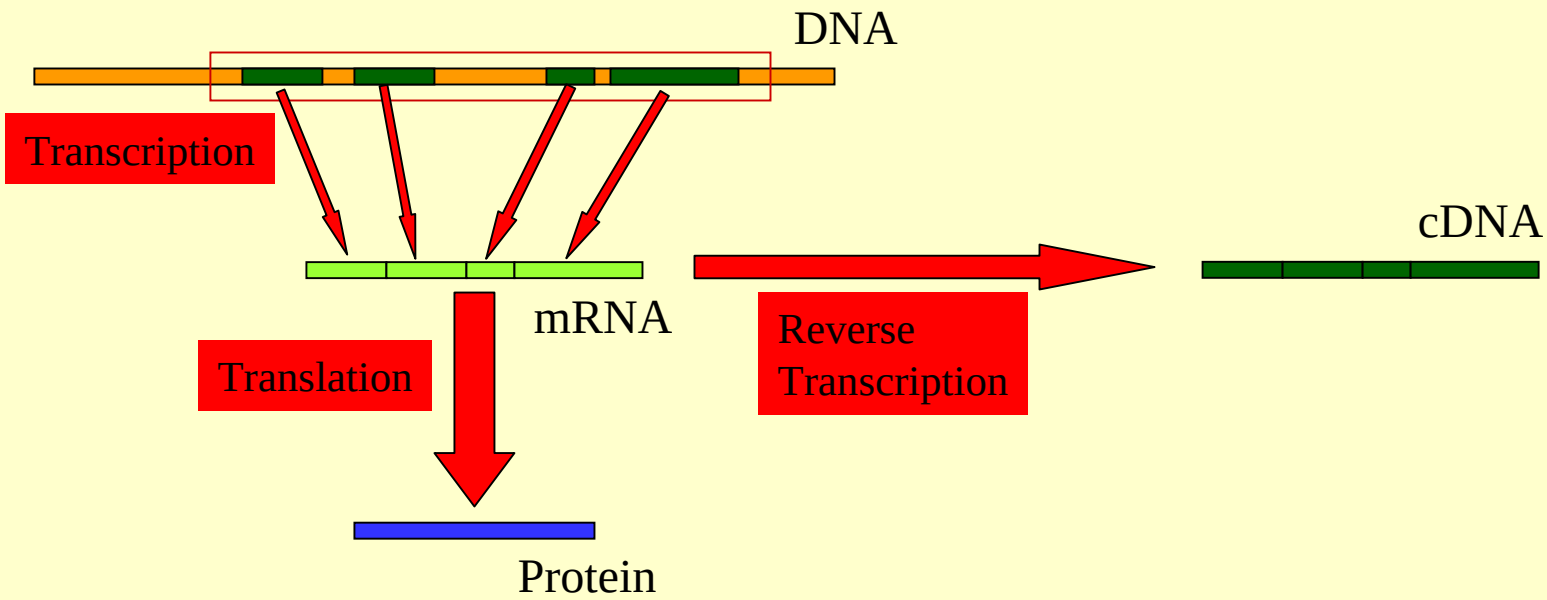
Why Gaps?

- Example: Finding the gene site for a given (eukaryotic) cDNA requires “gaps”.
- Example: HIV-virus strains



What is cDNA?

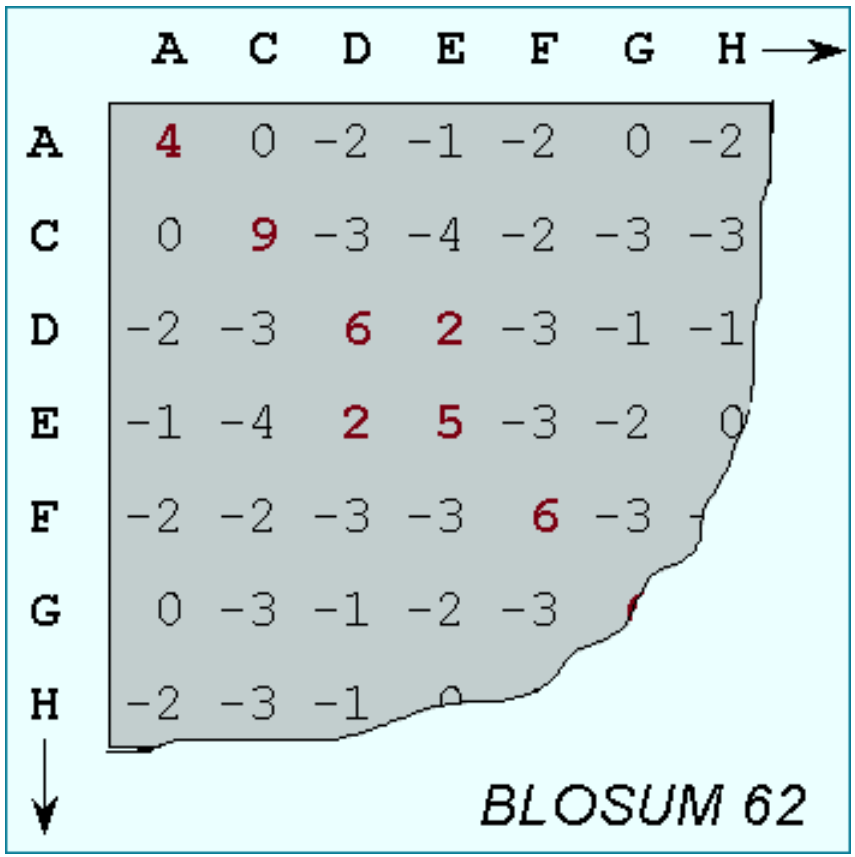
- cDNA = Copy DNA



Properties of Smith-Waterman Algorithm

- How to find all regions of "high similarity"?
  - Find all entries above a threshold score and traceback.
- What if: Matches = 1 & Mismatches/spaces = 0?
  - Longest Common Subsequence Problem
- What if: Matches = 1 & Mismatches/spaces = $-\infty$?
  - Longest Common Substring Problem
- What if the average entry is positive?
  - Global Alignment

How to score mismatches?



BLOSUM n Substitution Matrices

- For each amino acid pair a, b
  - For each BLOCK
    - Align all proteins in the BLOCK
    - Eliminate proteins that are more than **n**% identical
    - Count F(a), F(b), F(a,b)
    - Compute Log-odds Ratio

$$\log\left(\frac{F(a,b)}{F(a)F(b)}\right)$$

Henikoff & Henikoff, '92

Alternative Substitution Matrices

PAM250

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

Point Accepted Mutations (PAM)

- PAM is a unit of evolutionary distance.
- Protein sequences A and B are 1 PAM unit apart if one is converted to the other with an average of 1 accepted point mutation per 100 amino acids.
- Point Mutation ⇔ Substitutions (No InDels)
- Accepted ⇔ incorporated into protein and passed onto progeny

Dayhoff, 1978

True or False?

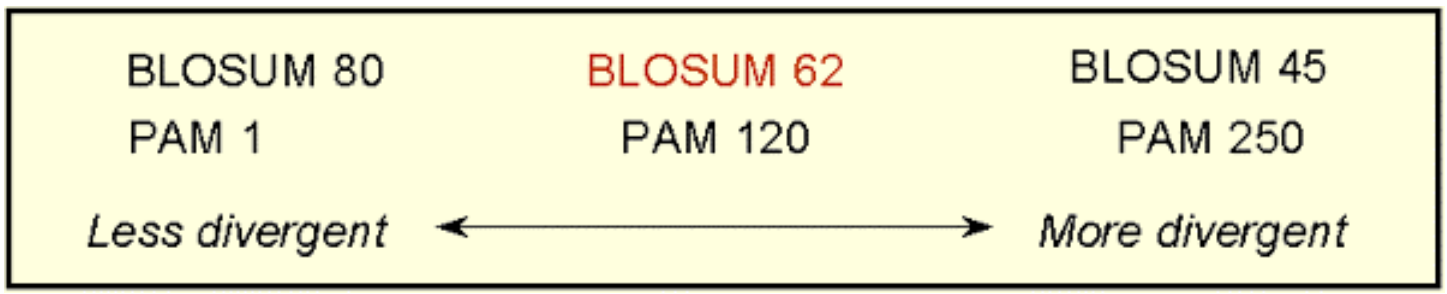
- If |A| = |B| = 400, and A and B are 1 PAM unit apart, then the expected number of differences between A and B is exactly 4.
- If |A| = |B|, and A and B are 100 PAM units apart, then they are expected to be different in every position.
- If A and B are 250 PAM units apart, then they are as distinct as a pair of random sequences. >15%

PAM Substitution Matrices

- Align very similar pairs of sequences (<15% difference).
- Identify and ignore InDels.
- For each amino acid pair (a,b) compute log-odds ratio:

$$\log\left(\frac{F(a,b)}{F(a)F(b)}\right)$$

PAM vs BLOSUM



Which Substitution Matrix?

- BLOSUM-62 matrix best for detecting most weak protein similarities.
- For particularly long and weak alignments, BLOSUM-45 matrix may be superior.
-

Query Length	Substitution Matrix	Gap Costs
<35	PAM 30	(9,1)
35-50	PAM 70	(10,1)
50-85	BLOSUM 80	(10,1)
>85	BLOSUM 62	(11,1)

General Bioinformatics Resources

- GenBank: (Portal) PubMed at NCBI, NIH
  - Try Lambda Cro (73101), Ecoli Sigma-70 (1SIG), Ecoli Sigma factor (1072030), Bacteriorhodopsin (14194473), 1baza vs. 1myka (P-22 Arc repressors)
- BLAST
- SwissPROT
- InterPro

BLAST & FASTA

- FASTA

[Lipman, Pearson ’85, ‘88]

- Basic Local Alignment Search Tool

[Altschul, Gish, Miller, Myers, Lipman ’90]

BLAST Overview

- Program(s) to search all sequence databases
- Tremendous Speed/Less Sensitive
- Statistical Significance reported
- WWWBLAST, QBLAST (send now, retrieve results later), Standalone BLAST, BLASTcl3 (Client version, TCP/IP connection to NCBI server), BLAST URLAPI (to access QBLAST, no local client)

Search Strategy

FASTA Strategy

- Find “hot spots” of length k (exact match) for each length k word in query.
- Locate “runs” of “hot spots”.
- Do detailed “Smith-Waterman” local alignment at these locations.

BLAST Strategy & Improvements

- Lipman et al.: speeded up finding “runs” of “hot spots”.
- Eugene Myers ’94: “Sublinear algorithm for approximate keyword matching”.
- Karlin, Altschul, Dembo ’90, ’91: “Statistical Significance of Matches”

BLAST Variants

- **Nucleotide BLAST**
  - Standard
  - MEGABLAST (Compare large sets, Near-exact searches)
  - Short Sequences (higher E-value threshold, smaller word size, no low-complexity filtering)
- **Protein BLAST**
  - Standard
  - PSI-BLAST (Position Specific Iterated BLAST)
  - PHI-BLAST (Pattern Hit Initiated BLAST; reg expr. Or Motif search)
  - Short Sequences (higher E-value threshold, smaller word size, no low-complexity filtering, PAM-30)
- **Translating BLAST**
  - Blastx: Search nucleotide sequence in protein database (6 reading frames)
  - Tblastn: Search protein sequence in nucleotide dB
  - Tblastx: Search nucleotide seq (6 frames) in nucleotide DB (6 frames)

BLAST Cont'd

- **RPS BLAST**
  - Compare protein sequence against Conserved Domain DB; Helps in predicting rough structure and function
- **Pairwise BLAST**
  - blastp (2 Proteins), blastn (2 nucleotides), tblastn (protein-nucleotide w/ 6 frames), blastx (nucleotide-protein), tblastx (nucleotide w/6 frames-nucleotide w/ 6 frames)
- **Specialized BLAST**
  - Human & Other finished/unfinished genomes
  - P. falciparum: Search ESTs, STSs, GSSs, HTGs
  - VecScreen: screen for contamination while sequencing
  - IgBLAST: Immunoglobin sequence database

BLAST Credits

- Stephen Altschul
- Jonathan Epstein
- David Lipman
- Tom Madden
- Scott McGinnis
- Jim Ostell
- Alex Schaffer
- Sergei Shavirin
- Heidi Sofia
- Jinghui Zhang

Useful Terms

- **E value:** Expectation value. expected # of alignments with scores equivalent to or better than S to occur by chance. The lower the E value, the more significant the score.
- **P value:** The probability of an alignment occurring with the given score, S, or better. Calculated by relating the observed score, S, to the expected distribution of HSP scores from comparisons of random sequences of the same length and composition as the query to the database. The most highly significant P values will be those close to 0.
- **HSP:** High-scoring segment pair. Local alignments with no gaps that achieve high alignment scores
- **Identity (Similarity):** The extent to which two (nucleotide or amino acid) sequences are invariant (similar).

Databases used by BLAST

- **Protein**
  - nr (everything), swissprot, pdb, alu, individual genomes
- **Nucleotide**
  - nr, dbest, dbsts, htgs (unfinished genomic sequences), gss, pdb, vector, mito, alu, epd
- **Misc**

Rules of Thumb

- Most sequences with significant similarity over their entire lengths are homologous.
- Matches that are > 50% identical in a 20-40 aa region occur frequently by chance.
- Distantly related homologs may lack significant similarity. Homologous sequences may have few absolutely conserved residues.
- A homologous to B & B to C $\Rightarrow$ A homologous to C.
- Low complexity regions, transmembrane regions and coiled-coil regions frequently display significant similarity without homology.
- Greater evolutionary distance implies that length of a local alignment required to achieve a statistically significant score also increases.

Rules of Thumb

- Results of searches using different scoring systems may be compared directly using normalized scores.
- If S is the (raw) score for a local alignment, the **normalized** score S' (in bits) is given by

$$S' = \frac{\lambda - \ln(\mathrm{K})}{\ln(2)}$$

  The parameters depend on the scoring system.
- **Statistically significant normalized score**,

$$S' > \log\left(\frac{N}{E}\right)$$

  where E-value = E, and N = size of search space.

Perl: Practical Extraction & Report Language

- Created by Larry Wall, early 90s
- Portable, “glue” language for interfacing C/Fortran code, WWW/CGI, graphics, numerical analysis and much more
- Easy to use and extensible
- OOP support, simple databases, simple data structures.
- From interpreted to compiled
- high-level features, and relieves you from manual memory management, segmentation faults, bus errors, most portability problems, etc, etc.
- Competitors: Python, Tcl, Java

Perl Features

- Perl – many features
  - Bit Operations, Pattern Matching, Subroutines, Packages & Modules, Objects, Interprocess Communication, Threads, Compiling, Process control
- Competitors to Perl: Python, Tcl, Java

BioPerl

- Routines for handling biosequence and alignment data.
- Why? Human Genome Project: Same project, same data. different data formats! Different input formats. Different output formats for comparable utility programs.
- BioPerl was useful to interchange data and meaningfully exchange results. “Perl Saved the Human Genome Project”
- Many routine tasks automated using BioPerl.
- String manipulations (string operations: substring, match, etc.; handling string data: names, annotations, comments, bibliographical references; regular expression operations)
- Modular: modules in any language

Sequencing Project

- a trace editor to analyze, and display the short DNA read chromatograms from DNA sequencing machines.
- a read assembler, to find overlaps between the reads and assemble them together into long contiguous sections.
- an assembly editor, to view the assemblies and make changes in places where the assembler went wrong.
- a database to keep track of it all.

Managing a Large Project

- Devise a common data exchange format.
- Use modules that have already been developed.
- Write Perl scripts to convert to and from common data exchange format.
- Write Perl scripts to “glue” it all together.

BioPerl Modules

- **Bio::PreSeq**, module for reading, accessing, manipulating, analyzing single sequences.
- **Bio::UnivAln**, module for reading, parsing, writing, slicing, and manipulating multiple biosequences (sequence multisets and alignments).
- **Bio::Struct**, module for reading, writing, accessing, manipulating, and analyzing 3D structures.
- Support for invoking **BLAST** and other programs.
- Listing: bioperl-1.0.2::Bio & here.
- BioPerl Tutorial

Miscellaneous

- pTk – to enable building Perl-driven GUIs for X-Window systems.
- BioJava
- BioPython
- The BioCORBA Project provides an object-oriented, language neutral, platform-independent method for describing and solving bioinformatics problems.

BioPerl: Structure

```
use Bio::Structure::IO;
$in = Bio::Structure::IO->new(-file => "inputfilename" , '-format' => 'pdb');
$out = Bio::Structure::IO->new(-file => ">outputfilename" , '-format' => 'pdb');
# note: we quote -format to keep older perl's from complaining.
while ( my $struc = $in->next_structure() ) {
    $out->write_structure($struc);
    print "Structure ",$struc->id," number of models: ",
            scalar $struc->model,"\n";
}
```

More Bioperl Modules

Bioperl-1.0.2::Bio::Structure::SecStr::DSSP
bioperl-1.0.2::Bio::Structure::SecStr::STRIDE
bioperl-1.0.2::Bio::Symbol
bioperl-1.0.2::Bio::Tools
bioperl-1.0.2::Bio::Tools::Alignment
bioperl-1.0.2::Bio::Tools::Bplite
bioperl-1.0.2::Bio::Tools::Blast
bioperl-1.0.2::Bio::Tools::HMMER
bioperl-1.0.2::Bio::Tools::Prediction
bioperl-1.0.2::Bio::Tools::Run::Alignment
bioperl-1.0.2::Bio::Tools::Sim4
bioperl-1.0.2::Bio::Tools::StateMachine
bioperl-1.0.2::Bio::Tree
bioperl-1.0.2::Bio::TreeIO