Pairwise Sequence Alignment

2020/10/21

1

Reminder:

There are four different nucleotides/bases that compose DNA and each has a one-letter symbol:

A (adenine)	G (guanine)	C (cytosine)	T (thymine)

There are 20 different amino acids that compose proteins and each has a one-letter symbol:

A (alanine)	C (cysteine)	D (aspartic acid)
E (glutamic acid)	F (phenylalanine)	G (glycine)
H (histidine)	I (isoleucine)	K (lysine)
L (leucine)	M (methionine)	N (asparagine)
P (proline)	Q (glutamine)	R (arginine)
S (serine)	T (threonine)	V (valine)
W (tryptophan)	Y (tyrosine)	

2020/10/21

2

WHAT IS SEQUENCE ALIGNMENT?

Sequence alignment– when sequences (protein or DNA) are compared, position by position, to establish the best correspondence between them

2020/10/21

Pairwise sequence alignment– the process of aligning two sequences

Example: Here two DNA sequences are aligned well:

```
AGTGCTT
||  ||||
AGAGCTT
```

Here they are not:

```
AGTGCTT--
     |
--AGAGCTT
```

3

WHY IS SEQUENCE ALIGNMENT/COMPARISON USEFUL?

2020/10/21

Allows one to determine if two sequence are actually related to each other and draw inferences regarding:

- evolutionary relationship
- similarity of function
- similarity of structure

Degree of similarity– reveals the evolutionary relatedness of the sequences

Differences– reflect changes that have occurred during evolution

Similarity between a new sequence and others that are well characterized– may indicate similar structure and function

4

HOW IS THE RELATEDNESS OF TWO SEQUENCES EXPRESSED?

1. identity
2. similarity
3. homology

2020/10/21

5

Identity– percentage of identical matches between two aligned sequences.

Example:

```
AVKLFV
|  ||
AARLF-
```

2020/10/21

Two methods to calculate percent identity (**I**):

$I = [(L_i \times 2)/(L_a + L_b)] \times 100\%$

L_i = # of aligned identical residues
L_a = total length of one sequence
L_b = total length of other sequence

$I = L_i/L_a \times 100\%$

L_a = total length of shorter sequence

6

Similarity– percentage of aligned residues that have similar physicochemical properties (and can more readily be substituted for each other).

Example:

```
AVKLFV
|..||
AARLF-
```

V and A have similar physicochemical properties; so do K and R.

2020/10/21

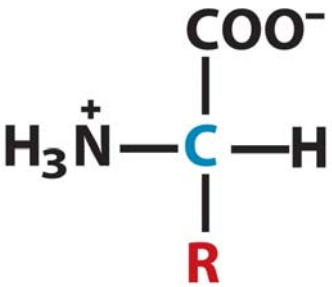
WHICH AMINO ACIDS HAVE SIMILAR PHYSICOCHEMICAL PROPERTIES?

The following is a review of the amino acids and some basic biochemistry…

7

Structure of an Amino Acid:

1. α carbon
2. amino group bonded to α carbon
3. carboxyl group bonded to α carbon
4. hydrogen bonded to α carbon
5. "R group"/"side chain" bonded to α carbon; 20 different common R groups

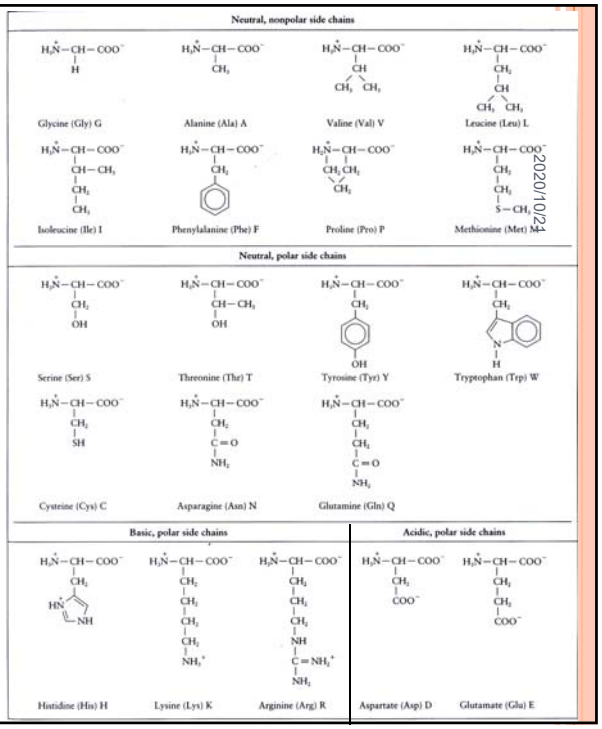


2020/10/21

8

Structures of the 20 amino acids, grouped by physicochemical properties.

One-letter and three-letter codes for each amino acid are shown.



2020/10/21

More information about the side chain properties of the 20 amino acids.

(From *Essential Bioinformatics*, by J. Xiong, p. 174)

TABLE 12.1. Twenty Standard Amino Acids Grouped by Their Common Side-Chain Features

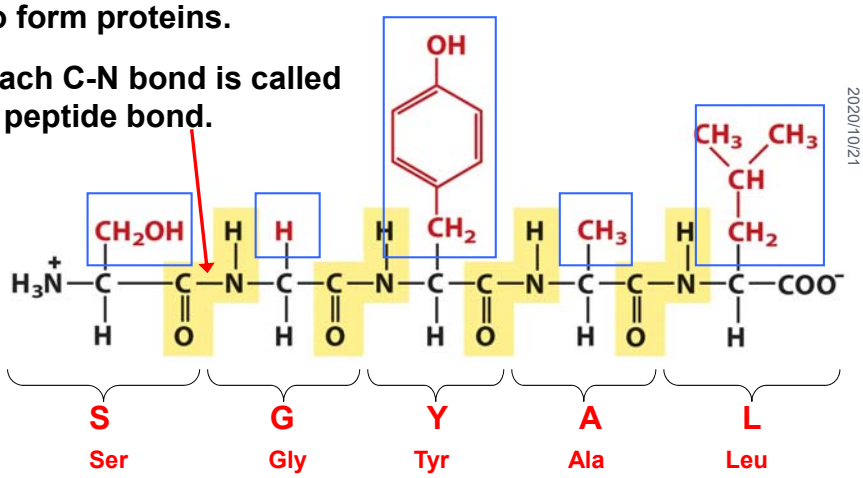
Amino Acid Group	Amino Acid Name	Three- and One-Letter Code	Main Functional Features
Small and nonpolar	Glycine Alanine Proline	Gly, G Ala, A Pro, P	Nonreactive in chemical reactions; Pro and Gly disrupt regular secondary structures
Small and polar	Cysteine Serine Threonine	Cys, C Ser, S Thr, T	Serving as posttranslational modification sites and participating in active sites of enzymes or binding metal
Large and polar	Glutamine Asparagine	Gln, Q Asn, N	Participating in hydrogen bonding or in enzyme active sites
Large and polar (basic)	Arginine Lysine Histidine	Arg, R Lys, K His, H	Found in the surface of globular proteins providing salt bridges; His participates in enzyme catalysis or metal binding
Large and polar (acidic)	Glutamate Aspartate	Glu, E Asp, D	Found in the surface of globular proteins providing salt bridges
Large and nonpolar (aliphatic)	Isoleucine Leucine Methionine Valine	Ile, I Leu, L Met, M Val, V	Nonreactive in chemical reactions; participating in hydrophobic interactions
Large and nonpolar (aromatic)	Phenylalanine Tyrosine Tryptophan	Phe, F Tyr, Y Trp, W	Providing sites for aromatic packing interactions; Tyr and Trp are weakly polar and can serve as sites for phosphorylation and hydrogen bonding

Note: Each amino acid is listed with its full name, three- and one-letter abbreviations, and main functional roles when serving as amino acid residues in a protein. Properties of some amino acid groups overlap.

2020/10/21

Amino acids are joined by amide linkages (shaded yellow) to form proteins.

Each C-N bond is called a peptide bond.



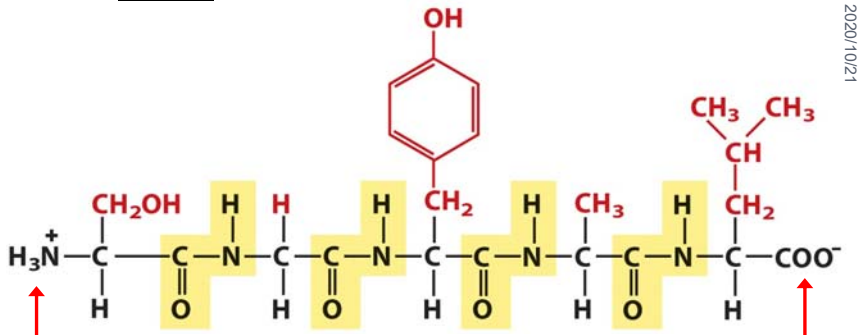
2020/10/21

The amide linkages are shaded in yellow and the side chains (R groups) of the amino acids are boxed in blue.

11

Fig 3-14, *Lehninger Principles of Biochemistry*, 4th ed.

A short chain of amino acids is called a **peptide** (usually less than ~100).
A longer chain of amino acids is called a **protein**.
When an amino acid is part of a peptide or protein, it is called an amino acid **residue**.



2020/10/21

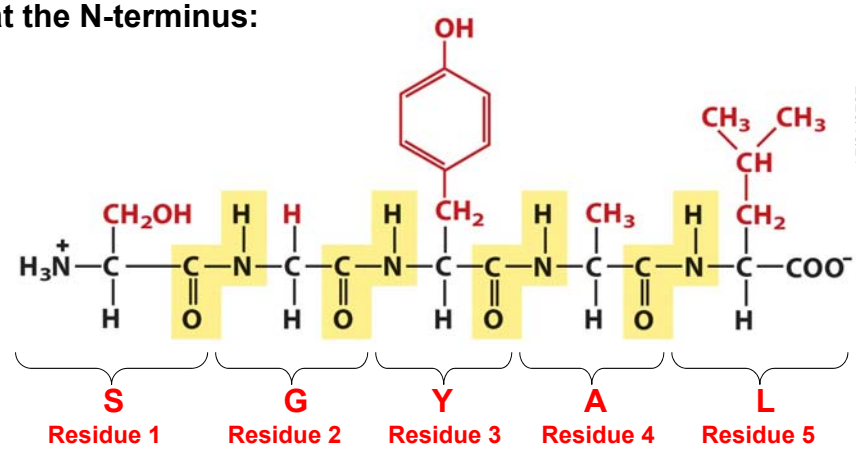
This end of the chain is called the **amino terminus**, or **N-terminus**, because there is a free amino group.

This end of the chain is called the **carboxyl terminus**, or **C-terminus**, because there is a free carboxyl group.

12

Fig 3-14, *Lehninger Principles of Biochemistry*, 4th ed.

Amino acid residues in a protein are numbered beginning at the N-terminus:



2020/10/21

The amino acid sequence of a protein is always written starting from the N-terminus. Therefore, the sequence of this peptide would be **SGYAL** (not LAYGS).

13

Fig 3-14, *Lehninger Principles of Biochemistry*, 4th ed.

Similarity– percentage of aligned residues that have similar physicochemical properties (and can more readily be substituted for each other).

Example:

```
AVKLFV
|..||
AARLF-
```

V and A are similar; K and R are similar.

Two methods to calculate percent similarity (**S**):

$S = [(L_s \times 2)/(L_a + L_b)] \times 100\%$

L_s = # of aligned identical or similar residues
L_a = total length of one sequence
L_b = total length of other sequence

$S = L_s/L_a \times 100\%$

L_a = total length of shorter sequence

2020/10/21

14

Homology– an inference that there is an evolutionary relationship between the sequences (that they are descended from a common ancestor).

VERY IMPORTANT– either they ARE homologous (are related), or they AREN'T homologous (aren't related). (It's like being pregnant– either you ARE or you AREN'T! It's incorrect to say two sequences are "somewhat" homologous.)

2020/10/21

HOW DOES ONE INFER HOMOLOGY OF SEQUENCES?

Based on percentage of identity/similarity.

BUT… two sequences could be identical just by chance.

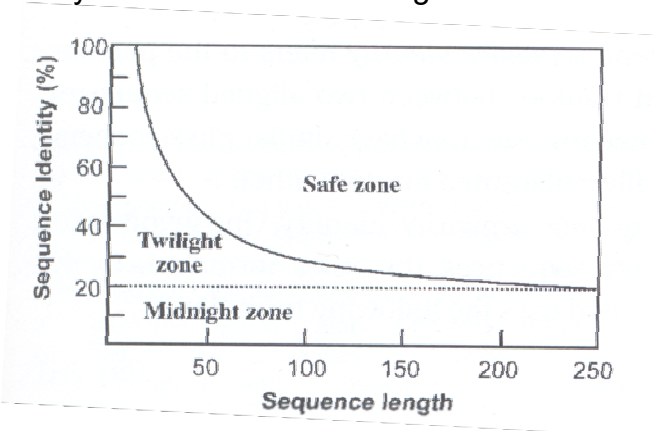
15

Two unrelated/nonhomologous sequences can be identical just by CHANCE.

SO… at what percentage identity is it safe to infer homology?

Any two sequences that fall within the "safe zone" can safely be considered homologous.

2020/10/21



16

Fig. 3.1, J. Xiong, *Essential Bioinformatics*

GOAL of pairwise alignment – find best pairing of two sequences so there is maximum correspondence between residues.

2020/10/21

Example: `A: CATTCAC;  B: CTCGCAGC`

```
A: C A T - T C A - C
   |   |     | |   |
B: C - T C G C A G C
```

Two METHODS for doing this:

- Global Alignment
- Local Alignment

17

Global Alignment– finds best alignment of two sequences over their entire lengths

Example:

```
CTGTCG-CTGCACG
-TGC-CG-TG----
```

2020/10/21

Most appropriate for two highly similar sequences of about the same length.

Local Alignment– finds regions with the highest similarity between two sequences and aligns these regions without regard to the rest of the sequences

Example:

```
CTGTCGCTGCACG--
-------TGC-CGTG
```

Most appropriate for two sequences containing only sections (“domains”/ “motifs”) of similarity; the two sequences can be of very different lengths.

18

Algorithms for generating global and local alignments may use one of three methods:

2020/10/21

1. Dot matrix (dot plot) method
2. Dynamic programming method
3. Word method

19

2020/10/21

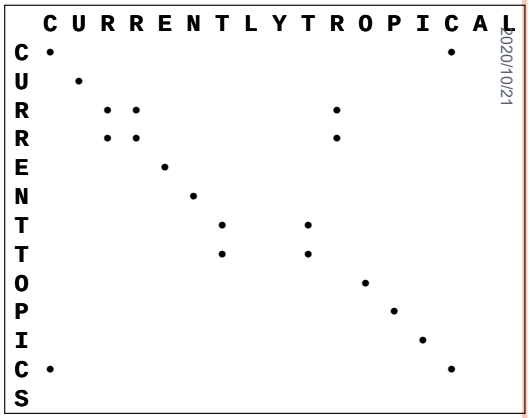
Dot matrix (dot plot) method

20

Dot matrix (dot plot) method– a graphical method using a two-dimensional matrix.

Example:

```
seq1   CURRENTLYTROPICAL
seq2   CURRENTTOPICS
```



Seq1 is placed on one axis of the matrix, and seq2 is placed on the other axis. A dot is placed at each position in the matrix where two residues are identical. (Sometimes dots are placed in positions where residues are similar too.)

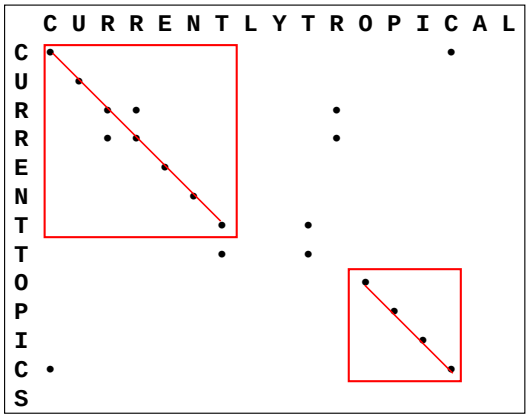
2020/10/21

21

When two sequences have regions of similarity, many dots line up forming diagonal lines.

The diagonals indicate that there are two "motifs" that are similar between these sequences: **CURRENT** & **OPIC**

2020/10/21



To generate the best alignment, link the diagonals by adding "gaps" as shown here:

```
CURRENTLYTROPICAL
|||||||  | ||||
CURRENT--T-OPICS-
```

22

NOISY PROBLEM

For long sequences, there may be too many dots ("**NOISE**"), obscuring the true alignment/diagonals.

2020/10/21

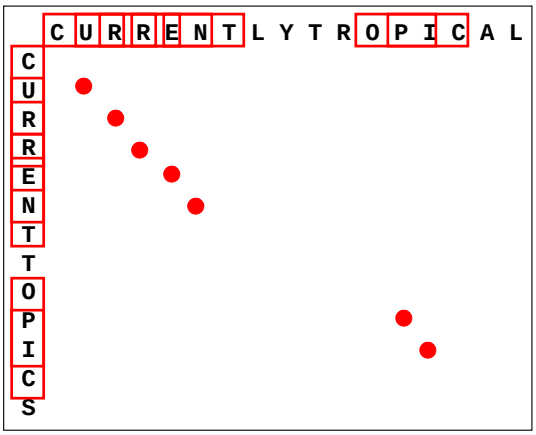
Use a filtering technique to solve this problem:
sliding **WINDOW**, or **TUPLE**.

- Choose window size
- Window slides across both sequences
- Dot is placed in matrix only when a certain number of residues within the window are identical or similar (a "threshold" is reached)
- Increase or decrease window size and threshold as needed to reduce noise

23

Using a window for filtering:
window = 3 residues;
threshold → all 3 residues within the window must be identical
The "noise dots" seen on the previous slide are now gone– only the important dots in regions of real similarity appear.

(By clicking the down arrow repeatedly in slide show mode you should be able to see the sliding window move across the sequences and the dots appear.)




```
CURRENTLYTROPICAL
|||||||  | ||||
CURRENT--T-OPICS-
```

2020/10/21

24

PROS of the dot matrix method:

- gives a direct visual statement of the relationship between two sequences
- displays all possible sequence matches

2020/10/21

CONS of the dot matrix method:

- software that generates the matrix may not construct a sequence alignment (it is up to the user to link the diagonals by adding gaps)
- lacks statistical rigor in assessing the quality of the alignment
- restricted to comparing only two sequences

In class you will use the dot matrix method to compare two protein sequences.

25

2020/10/21

Dynamic programming method

26

Dynamic programming method – generates a 2D alignment matrix containing scores for all possible alignments in order to quantitatively find the best alignment

Example:

```
seq1 ATTGC
seq2 AGGC
```

2020/10/21

Need a **SCORING SYTEM** to quantitatively determine the best alignment: $score(A) = \alpha w - \beta x - \gamma y$

w = #matches x = #mismatches y = #gaps

Reward for matches:	α	(positive score)
Mismatch penalty:	$-\beta$	(negative score)
Gap penalty:	$-\gamma$	(negative score)

What is a GAP?

Suppose the two sequences were aligned like this:
Inserting a gap allows the G's and C's to align.

```
seq1 ATTGC
     |  ||
seq2 A-GGC
```

Why a gap PENALTY?

A gap in a sequence alignment implies that an insertion or deletion event is presumed to have occurred during evolution. Insertions and deletions are rare compared to substitutions. Gaps are usually penalized to reflect this.

27

GLOBAL ALIGNMENT: SCORING

2020/10/21

Reward for matches:	10
Mismatch penalty:	-2
gat penalty:	-5

C	T	G	T	C	G	–	C	T	G	C
-	T	G	C	–	C	G	–	T	G	-
-5	10	10	-2	-5	-2	-5	-5	10	10	-5

Total score = 4x10-2x2-5x5 = 11

28

ALIGNMENT ALGORITHMS

2020/10/21

- Global alignment: Needleman-Wunsch algorithm
- Local alignment: Smith-Waterman algorithm
- NW and SW use ***dynamic programming***
- Variations:
  - Gap penalty functions
  - Scoring matrices

29

GLOBAL ALIGNMENT: ALGORITHM

2020/10/21

$$S_{1..i} = \text{Prefix of length i of S}$$
$$T_{1..j} = \text{Prefix of length j of T}$$

$$C(i,j) = \text{Cost of optimum alignment of } S_{1..i} \text{ and } T_{1..j}$$

$$w(a,b) = \begin{cases} +\alpha & \text{if } a = b \\ -\beta & \text{if } a \neq b \end{cases}$$

30

Theorem. *C(i,j) satisfies the following relationships:*

2020/10/21

Initial conditions:

$$C(i,0) = -i \cdot \gamma \qquad C(0,j) = -j \cdot \gamma$$

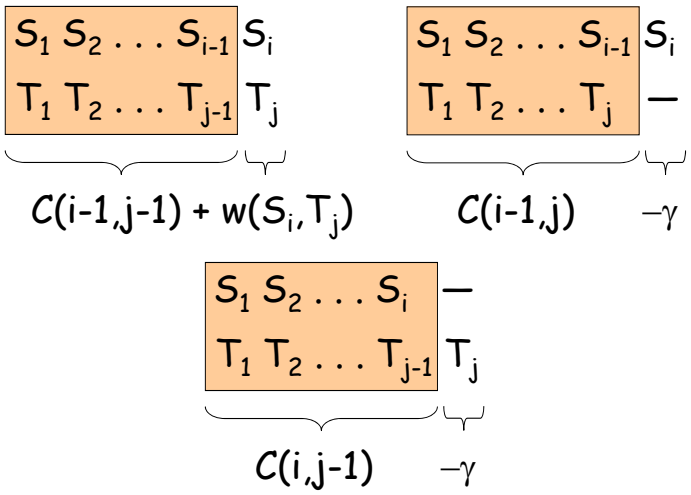
Recurrence relation: For $1 \le i \le n$, $1 \le j \le m$:

$$C(i,j) = \max \begin{cases} C(i-1,j-1) + w(S_i, T_j) \\ C(i-1,j) - \gamma \\ C(i,j-1) - \gamma \end{cases}$$

31

JUSTIFICATION

2020/10/21



32

EXAMPLE

Case 1: Line up S_i with T_j

```
                  i - 1   i
S: C  A  T  T  C  A  |C
T: C  -  T  T  C  A  |G
                  j -1   j
```

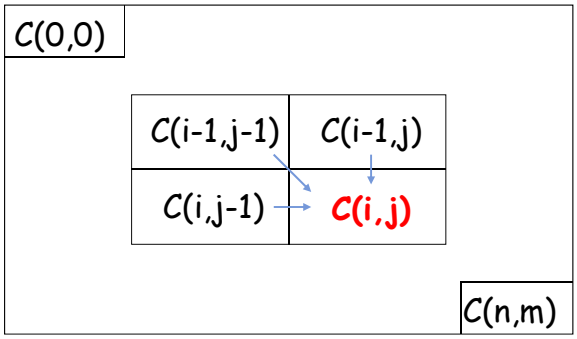
Case 2: Line up S_i with space

```
                  i - 1   i
S: C  A  T  T  C  A  -  |C
T: C  -  T  T  C  A  G  |-
                        j
```

Case 3: Line up T_j with space

```
                     i
S: C  A  T  T  C  A  C  |-
T: C  -  T  T  C  A  -  |G
                  j -1   j
```


COMPUTATION PROCEDURE



$$C(i,j) = \max\{C(i-1,j-1) + w(S_i, T_j),\ C(i-1,j) - \gamma,\ C(i,j-1) - \gamma\}$$

	λ	C	T	C	G	C	A	G	C
λ	0	-5	-10	-15	-20	-25	-30	-35	-40
C	-5	10	5						
A	-10								
T	-15								
T	-20								
C	-25								
A	-30								
C	-35								

2020/10/21

+10 for match, -2 for mismatch, -5 for space

35

	λ	C	T	C	G	C	A	G	C
λ	0	-5	-10	-15	-20	-25	-30	-35	-40
C	-5	10	5	0	-5	-10	-15	-20	-25
A	-10	5	8	3	-2	-7	0	-5	-10
T	-15	0	15	10	5	0	-5	-2	-7
T	-20	-5	10	13	8	3	-2	-7	-4
C	-25	-10	5	20	15	18	13	8	3
A	-30	-15	0	15	18	13	28	23	18
C	-35	-20	-5	10	13	28	23	26	33

2020/10/21

Traceback can yield both optimum alignments

```
S: C - T C G C A G C
T: C A T T - C A - C
```

```
S: C - T C G C A G C
T: C A T - T C A - C
```

Local Alignment: Motivation

2020/10/21

- **Ignoring stretches of non-coding DNA:**
  - Non-coding regions are more likely to be subjected to mutations than coding regions.
  - Local alignment between two sequences is likely to be between two exons.
- **Locating protein domains:**
  - Proteins of different kind and of different species often exhibit local similarities
  - Local similarities may indicate ”functional subunits”.

37

Local alignment: Example

2020/10/21

S = **G G T C T G A G**
T = **A A A C G A**

Match: +2 Mismatch and space: -1

Best local alignment:

G G T **C T G A** G
a a a **C – G A** -

Score = 5

38

Local Alignment: Algorithm

2020/10/21

$C[i, j]$ = Score of optimally aligning a suffix of s with a suffix of t.

$$C[i, j] = \max \begin{cases} C[i-1, j-1] + score(s[i], t[j]) \\ C[i-1, j] - \gamma \\ C[i, j-1] - \gamma \\ 0 \end{cases}$$

Initialize top row and leftmost column to zero.

39

2020/10/21

	λ	C	T	C	G	C	A	G	C
λ	0	0	0	0	0	0	0	0	0
C	0	1	0	1	0	1	0	0	1
A	0	0	0	0	0	0	2	0	0
T	0	0	1	0	0	0	0	1	0
T	0	0	1	0	0	0	0	0	0
C	0	1	0	2	0	1	0	0	1
A	0	0	0	0	1	0	2	0	0
C	0	1	0	1	0	2	0	1	1

+1 for a match, -1 for a mismatch, -5 for a space

40

Some Results

2020/10/21

- Most pairwise sequence alignment problems can be solved in $O(mn)$ time.
- Space requirement can be reduced to $O(m+n)$, while keeping run-time fixed [Myers88].
- Highly similar sequences can be aligned in $O(dn)$ time, where d measures the distance between the sequences [Landau86].

41

Reducing space requirements

2020/10/21

- O(mn) tables are often the limiting factor in computing large alignments
- There is a linear space technique that only doubles the time required [Hirschberg77]

42

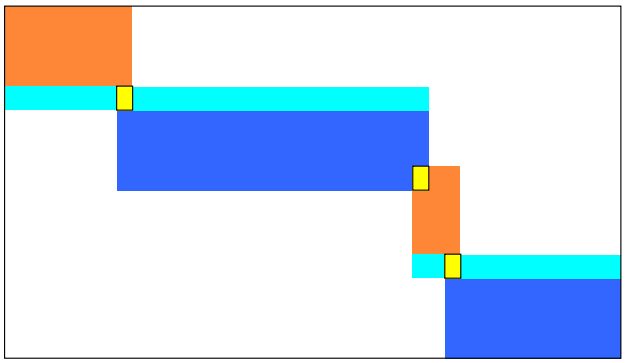
	λ	C	T	C	G	C	A	G	C
λ	0	0	0	0	0	0	0	0	0
C	0	10	5	10	5	10	5	0	10
A	0	5	8	5	8	5	20	15	10
T									
T									
C									
A									
G									

2020/10/21

IDEA: We only need the *previous* row to calculate the *next*

16

LINEAR-SPACE ALIGNMENTS



2020/10/21

mn + ½ mn + ¼ mn + 1/8 mn + 1/16 mn + … = 2 mn

44

Affine Gap Penalty Functions

2020/10/21

Gap penalty = $h + gk$

where

k = length of gap
h = gap opening penalty
g = gap continuation penalty

45

GAP PENALTIES should be…

- not TOO HIGH: otherwise gaps will never be inserted; reasonable alignment can't be produced;
- not TOO LOW: otherwise too many gaps will be inserted and unrelated sequences will be aligned with high scores.

2020/10/21

OPENING a gap versus EXTENDING a gap

- Higher penalty for opening than for extending (e.g., -12 to open a gap and -1 to extend the gap); called "affine" gap penalty when different penalties are used for opening and extending.
- Reasoning: if an insertion or deletion occurs, several adjacent residues are likely to have been inserted or deleted together, so the additional residues are not penalized as much as the initial opening of the gap.
- Example:

```
GATTTGCAC
||   ||||
GA---GCAC
```

total gap penalty = -12 + -1 + -1 = -14

TERMINAL / END gaps

- These are often not penalized.
- Example at right:

```
GAGCACTTT
||||||
GAGCAC---
```

46

2020/10/21

Substitution Matrix or Scoring Matrix

47

MATCH and MISMATCH SCORES

A table of match and mismatch scores called a **SUBSTITUTION MATRIX, or SCORING MATRIX,** is used to assign match and mismatch scores during dynamic programming.

2020/10/21

Examples:

A nucleotide scoring matrix:

	A	G	T	C
A	2			
G	-6	2		
T	-6	-6	2	
C	-6	-6	-6	2

This substitution matrix assumes that the frequencies of mutation are equal for all bases, since all mismatches receive a penalty of -6. However, this is not exactly true (not realistic).

Another nucleotide scoring matrix:

	A	G	T	C
A	2			
G	-5	2		
T	-7	-7	2	
C	-7	-7	-5	2

This substitution matrix assumes that transitions (G←→A or C←→T mutations) occur more frequently than transversions. Transitions are "easier," so the mismatch penalty is less severe (-5, rather than -7).

48

AMINO ACID SCORING MATRICES

Amino acid side chains have different physicochemical properties.

2020/10/21

Match and mismatch scores reflect the fact that residues with similar properties can be substituted more easily than those with different properties.

Example:

```
AVKLFV
|| |||
AVRLFV
```

K and R are both positively charged. Substitution may not affect structure/function of protein; minimal penalty for K/R substitution.

```
AVKLFV
|| |||
AVDLFV
```

D is negatively charged. Therefore, K/D substitution is less likely; high penalty for K/D substitution.

49

There are two common sets of amino acid scoring matrices:
PAM and BLOSUM matrices (which will be described next).

Both sets of matrices derive match and mismatch scores by analyzing the probabilities of amino acid substitutions in actual alignments of real proteins that are highly similar.

2020/10/21

Interpretation of the values in any amino acid scoring matrix (see the partial matrix shown below):

Positive scores– the frequency of amino acid substitutions found in a set of homologous sequences is greater than would have occurred by chance. (Example: Y replaces W in homologous proteins more often than expected by chance)

Zero scores– the frequency of amino acid substitutions found in a set of homologous sequences is equal to that expected by chance.

Negative scores – the frequency of amino acid substitutions found in a set of homologous sequences is less than that expected by chance. (Example: V replaces W less frequently in homologous proteins than would be expected by chance; this means V may simply replace W by chance)

The magnitude of a positive score reflects how often a particular amino acid occurs in nature (W is rare→11; V is not rare→4)

Part of an amino acid scoring matrix:

```
  W 11
  Y  2  7
  V -3 -1  4
     W  Y  V
```

50

PAM Matrices

Developed by Dayhoff et al. in 1978. They examined amino acid substitutions in a group of closely related proteins (shared ~85% sequence similarity).

2020/10/21

The substitutions they observed represent amino acid changes that do not change the function of the protein (since the proteins were so closely related), so they are called "accepted mutations" (amino acid changes "accepted" by natural selection).

PAM– stands for **P**oint **A**ccepted **M**utation
(Accepted Point Mutation would have made more sense!)

51

How were the values in the PAM matrices calculated?

The number of changes of each amino acid into every other amino acid were counted, and that number divided by a factor to normalize the value for variations in amino acid composition of the sequences and mutation rate of each amino acid. Finally, the log of this value was taken.

2020/10/21

The resulting values reflect the likelihood of amino acid substitutions. They were placed in a 20 x 20 matrix which shows the likelihood of all possible amino acid changes; this matrix was called the **PAM1 matrix**.

The values in the PAM1 matrix represent the probabilities that a given amino acid will mutate to another specific amino acid, within a sequence in which 1% of the residues have undergone mutation.

Therefore, the PAM1 matrix represents a substitution level of 1% of the changes expected to occur in 2,500 million years.

One PAM unit– one amino acid change per 100 residues

52

PAM1 matrix was derived from closely related sequences that are assumed to be separated by short periods of evolutionary time (only contain 1% of the changes expected to occur in 2,500 million years). **Therefore, the PAM1 matrix is appropriate only for aligning very similar sequences.**

2020/10/21

Other PAM Matrices for Sequences Separated by Longer Evolutionary Distances:

The PAM1 matrix can be multiplied by itself 'N' times to give matrices for comparing proteins that have undergone 'N' mutations.

Important assumption: a mutation at a given site is independent of previous mutations at that site.

Example: $(PAM1)^{250}$ = PAM250
PAM250 represents a substitution level of 250% of the changes expected to occur in 2,500 million years (250 mutations per 100 residues).

Each PAM matrix is designed to score alignments between sequences that have diverged by a particular degree of evolutionary distance.

PAM Matrix	**To be used for:**
PAM60	aligning sequences that have ~60% similarity
PAM80	aligning sequences that have ~50% similarity
PAM120	aligning sequences that have ~40% similarity
PAM250	aligning sequences that have ~20% similarity

53

The PAM250 Matrix

(residues grouped by physicochemical characteristics)

2020/10/21

	C	S	T	P	A	G	N	D	E	Q	H	R	K	M	I	L	V	F	Y	W
C Cys	12																			
S Ser	0	2																		
T Thr	-2	1	3																	
P Pro	-3	1	0	6																
A Ala	-2	1	1	1	2															
G Gly	-3	1	0	-1	1	5														
N Asn	-4	1	0	-1	0	0	2													
D Asp	-5	0	0	-1	0	1	2	4												
E Glu	-5	0	0	-1	0	0	1	3	4											
Q Gln	-5	-1	-1	0	0	-1	1	2	2	4										
H His	-3	-1	-1	0	-1	-2	2	1	1	3	6									
R Arg	-4	0	-1	0	-2	-3	0	-1	-1	1	2	6								
K Lys	-5	0	0	-1	-1	-2	1	0	0	1	0	3	5							
M Met	-5	-2	-1	-2	-1	-3	-2	-3	-2	-1	-2	0	0	6						
I Ile	-2	-1	0	-2	-1	-3	-2	-2	-2	-2	-2	-2	-2	2	5					
L Leu	-6	-3	-2	-3	-2	-4	-3	-4	-3	-2	-2	-3	-3	4	2	6				
V Val	-2	-1	0	-1	0	-1	-2	-2	-2	-2	-2	-2	-2	2	4	2	4			
F Phe	-4	-3	-3	-5	-5	-5	-4	-6	-5	-5	-2	-4	-5	0	1	2	-1	9		
Y Tyr	0	-3	-3	-5	-3	-5	-2	-4	-4	-4	0	-4	-4	-2	-1	-1	-2	7	10	
W Trp	-8	-2	-5	-6	-6	-7	-4	-7	-7	-5	-3	2	-3	-4	-5	-2	-6	0	0	17

Some criticisms of PAM matrices:

• The original PAM matrices were derived from only a small set of closely related proteins. (Given the information available in 1978, this was an important advance.) The Dayhoff data set has been augmented to include the 1991 protein database.

2020/10/21

• They are based on the assumption that forces responsible for sequence evolution over short evolutionary time spans are the same as those over longer time spans.

• They are based on the assumption that each amino acid position is equally mutable (but mutation hotspots are well known).

(For a nice discussion of PAM matrices, see *Fundamental Concepts of Bioinformatics*, by D.E. Krane & M.L. Raymer, 2003)

55

BLOSUM Matrices

Developed by Henikoff & Henikoff in 1992. They examined motifs within 500 protein families and aligned the motifs. When motifs can be aligned without introducing a gap, the resulting sequence alignment is called a "block."

2020/10/2

motif– conserved sequence of amino acids that confers a specific structure or function
block– an ungapped alignment of ~3 to 60 residues

They determined the frequency of amino acid substitutions in blocks and divided by the frequency of substitution expected by chance. The resulting values reflect the likelihood of amino acid substitutions. The values were placed in a 20 x 20 matrix which shows all possible amino acid changes– called a **BLOSUM matrix**.

BLOSUM– stands for **BLO**cks **SU**bstitution **M**atrix

56

BLOSUM Matrices (cont'd)

To avoid overrepresentation of amino acid substitutions that occur in the most closely related members of a family of proteins, sequences that were very similar were grouped together (clustered) to obtain a single "average" sequence before calculating the frequency of amino acid substitutions in the blocks. This produced a **SERIES of BLOSUM matrices:**

BLOSUM45– sequences that were 45% or more identical were clustered
(This matrix is weighted more heavily for distantly related sequences– those that have less than 45% identity)

BLOSUM62– sequences that were 62% or more identical were clustered
(This matrix is weighted more heavily for sequences that have less than 62% identity– it is an "intermediate" matrix)

BLOSUM90– sequences that were 90% or more identical were clustered
(This matrix is weighted less heavily for distantly related sequences, since only very closely related sequences are clustered)

Etc.

2020/10/21

57

The BLOSUM62 Matrix

(residues grouped by physicochemical characteristics)

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

2020/10/21

58

Comparison of PAM and BLOSUM Matrices

PAM Matrices

- Based on a model of evolution that assumes changes at one site are independent of previous changes at that site
- Prepared from sequences that are very similar (85% similarity)
- All amino acid positions are scored (global alignments)
- Matrices to compare more distantly related proteins are derived by extrapolation
- Designed to track evolutionary origins of proteins

BLOSUM Matrices

- Derived from amino acid changes observed in aligned regions of families of related proteins, regardless of degree of similarity (evolutionary distance) between family members
- Only amino acids in conserved blocks are scored (local alignments)
- Designed to find conserved domains in proteins

2020/10/21

59

Which matrix should be used when?

Matrix	Best Use	% Similarity the Matrix is Best Able to Detect
PAM40	short alignments that are highly similar	70-90
PAM160	detecting members of a protein family	50-60
PAM250	longer alignments of more divergent sequences	~30
BLOSUM90	short alignments that are highly similar	70-90
BLOSUM80	detecting members of a protein family	50-60
BLOSUM62	most effective in finding all potential similarities	30-40
BLOSUM30	longer alignments of more divergent sequences	<30

2020/10/21

PAM-BLOSUM Equivalencies:

PAM250 ≈ BLOSUM45
PAM160 ≈ BLOSUM62
PAM120 ≈ BLOSUM80
PAM30 ≈ BLOSUM90

"No single matrix is the complete answer for all sequence comparisons."

60

(Quote and table from A.D. Baxevanis & B.F.F. Ouellette, *Bioinformatics,* 3rd ed., John Wiley & Sons, Inc., 2005.)

Statistical Significance of Sequence Alignments

A sequence alignment shows that two sequences have a certain degree of similarity. **BUT**…could the observed alignment occur by random chance (occur between two unrelated sequences), or is the alignment statistically significant?

Example:
```
AVKLFV
|..||.
AARLFI    score = S
```

2020/10/21

Calculate a score (S) for this alignment. Then calculate a score (X) for the first sequence aligned with a **random, unrelated** sequence (the second sequence is shuffled randomly to simulate an unrelated sequence):

second sequence has been shuffled →
```
AVKLFV
 . ...
RAFILA   score = X
```

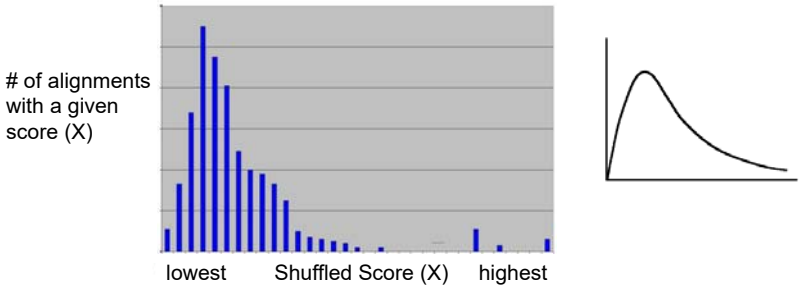
Shuffle the second sequence many times and calculate an alignment score each time:

```
AVKLFV              AVKLFV              AVKLFV
.  ...               . ...              .. ..
FRAAIL  score = X   RAIALF  score = X   IAFALR  score = X
```

61

Plot these scores (X's) to see the distribution of values. Carry out a statistical test to compare the original score (S) to all these scores.

Suppose you had 1000 scores from alignments with 1000 shuffled sequences. Make a plot of those scores as shown below. The distribution of scores will have a particular pattern, called the **Extreme Value Distribution.**



2020/10/21

Now compare your original score (S) to this distribution. If S falls in the extreme right margin, the alignment is probably not due to chance, and it is safe to infer that the sequences are homologous.

A **P-value** can be calculated to indicate the probability that the alignment is due to chance (the smaller the P-value, the less likely the alignment is due to chance).

$$P(X \geq S) = 1 - \exp[-Kmne^{-\lambda S}]$$

(*m* and *n* are sequence lengths; *K* and λ are constants that depend on the scoring matrix used)

62

This slide will be explained in class.

Interpreting P-values in terms of sequence homology

P-values	Interpretation
P-value < 10^{-50}	nearly identical match ("safe" to infer homology)
P-value 10^{-5} to 10^{-50}	"safe" to infer homology
P-value 10^{-1} to 10^{-5}	possible distant homologs
P-value > 10^{-1}	sequences may be randomly related

(the smaller the P-value, the less likely the alignment is due to chance)

2020/10/21

63