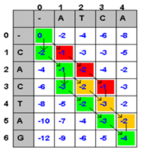
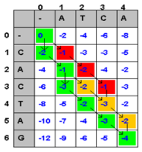
Bioinformatics

Three
Pairwise Sequence Alignment



		0	1	2	3	4
		-	A	T	C	A
0	-	0	-2	-4	-6	-8
1	C	-2	-1	-3	-3	-5
2	A	-4	-1	-2	-4	-2
3	C	-6	-3	-2	-1	-3
4	T	-8	-5	-2	-3	-2
5	A	-10	-7	-4	-3	-2
6	G	-12	-9	-6	-5	-4

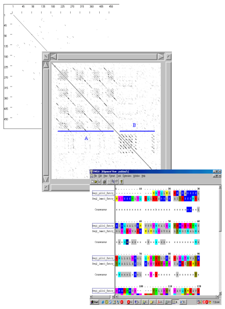
Sami Khuri
Department of Computer Science
San José State University
San José, California, USA
sami.khuri@sjsu.edu
www.cs.sjsu.edu/faculty/khuri



		0	1	2	3	4
		-	A	T	C	A
0	-	0	-2	-4	-6	-8
1	C	-2	-1	-3	-3	-5
2	A	-4	-1	-2	-4	-2
3	C	-6	-3	-2	-1	-3
4	T	-8	-5	-2	-3	-2
5	A	-10	-7	-4	-3	-2
6	G	-12	-9	-6	-5	-4

©2012 Sami Khuri

Pairwise Sequence Alignment



- **Homology**
- **Similarity**
- **Global string alignment**
- **Local string alignment**
- **Dot matrices**
- **Dynamic programming**
- **Scoring matrices**
- **BLAST**

©2012 Sami Khuri

Sequence Alignment

- **Sequence alignment** is the procedure of comparing sequences by searching for a series of individual characters or character patterns that are in the same order in the sequences.
  - Comparing two sequences gives us a **pairwise sequence alignment**.
  - Comparing more than two sequences gives us **multiple sequence alignment**.

©2012 Sami Khuri

Pairwise vs Multiple Alignment

- Pairwise Sequence Alignment
  - Infer biological relationships from the sequence similarity
- Multiple Sequence Alignment
  - Infer sequence similarity from biological relationships

    Starting point: sequences that are biologically related. Use the MSA to infer phylogenetic relationships. They can help elucidate biological facts about proteins since most conserved regions are biologically significant. MSA's can help formulate and test hypotheses about protein 3-D structure and function.

©2012 Sami Khuri

Why Do We Align Sequences?

- The basic idea of aligning sequences is that **similar DNA sequences** generally produce **similar proteins**.
- To be able to predict the characteristics of a protein using only its sequence data, the **structure** or **function** information of known proteins with similar sequences can be used.
- To be able to check and see whether two (or more) genes or proteins are evolutionarily related to each other.

©2012 Sami Khuri

Importance of Alignments

- Alignment methods are at the core of many of the software tools used to search the databases.
- Alignment is the task of locating equivalent regions of two or more sequences to maximize their similarity.
- In order to assess the similarity of two sequences it is necessary to have a quantitative measure of their alignment, which includes the degree of similarity of two aligned residues as well as accounting for insertions and deletions.

Understanding Bioinformatics by Zvelebil and Baum

©2012 Sami Khuri

Query Sequence

If a query sequence is found to be significantly similar to an already annotated sequence (DNA or protein), we can use the information from the annotated sequence to possibly infer **gene structure** or **function** of the query sequence.

Aligning Sequences

- There are many sequences, a handful of which have known structure and function.
- If two sequences align, they are similar, maybe because of a common ancestor.
- If they are similar, they might have the same structure or function.
- If one of them has known structure or function, then the alignment gives some insight about the structure or function of the other sequence.

Similarity and Difference

- The similarity of two DNA sequences taken from different organisms can be explained by the theory that all contemporary genetic material has one common ancestral DNA.
- Differences between families of contemporary species resulted from mutations during the course of evolution.
  - Most of these changes are due to local mutations between nucleotide sequences.

When To Do The Pairwise Comparison?

- You have a strong suspicion that two sequences are homologues.
  - Two sequences are homologues, when they share a common ancestor.

Homology and Similarity

Homology

- Evolutionary related sequence.
- A common ancestral molecular sequence.

Similarity

- Sequences that share certain sequence patterns.
- Directly observable from alignment.

Homology

- Common ancestry
- Sequence (and usually structure) conservation
- Homology is not a measurable quantity
- Homology can be inferred, under suitable conditions.

Similarity

- Most common method for comparing sequences.
- Similarity is a measurable quantity
- The degree of similarity depends on the parameters used (alphabet, scoring matrix, etc.).

Identity

- Most objective measuring entity for comparing sequences.
- Identity is well-defined.
- Identity can be quantified by several methods:
  - Percent of identical characters.
  - The number of identical matches divided by the length of the aligned region.

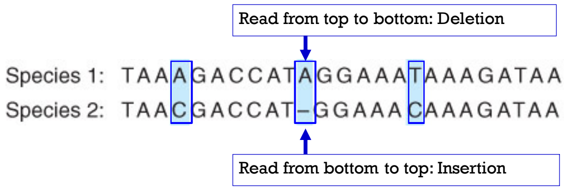
Similarity and Difference

- The similarity of two DNA sequences taken from different organisms can be explained by the theory that all contemporary genetic material has one common ancestral DNA.
- Differences between families of contemporary species resulted from mutations during the course of evolution.
  - Most of these changes are due to local mutations between nucleotide sequences.

Evolution and Alignments

- Alignments reflect the probable evolutionary history of two sequences.
- Residues that align and that are not identical represent substitutions.
- Sequences without correspondence in aligned sequences are interpreted as indels and in an alignment are gaps.

How do we Compare Sequences?



Read from top to bottom: Deletion

```
Species 1:  TAAAGACCATAGGAAATAAAGATAA
Species 2:  TAACGACCAT–GGAAACAAAGATAA
```

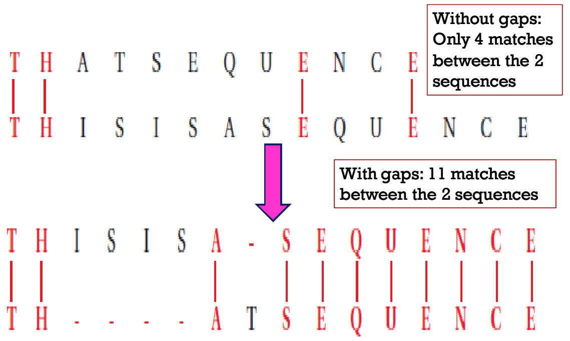
Read from bottom to top: Insertion

Determining the similarity of two genes by aligning their nucleotide sequences as well as possible; the differences due to mutation are shown in boxes.

Quantifying Alignments

- How should alignments be scored?
  - Do we use +1 for a match and -1 for a mismatch?
- Should we allow gaps to open the sequence so as to produce better matches elsewhere in the sequence?
  - If gaps are allowed, how should they be scored?

Need for Gaps: An Example



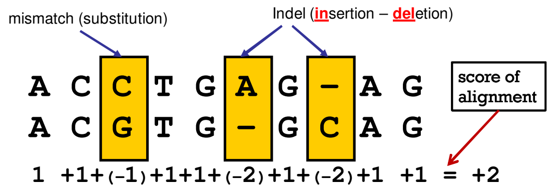
Without gaps: Only 4 matches between the 2 sequences

With gaps: 11 matches between the 2 sequences

Understanding Bioinformatics by Zvelebil and Baum

Scoring a Pairwise Alignment

- The two sequences are 70% identical



- Score of the alignment where:

  Match → +1   Mismatch → -1   Indel → -2

Problem Definition

Given:

- Two sequences.
- A scoring system for evaluating match or mismatch of two characters.
- A penalty function for gaps in sequences.

Find:

- An **optimal pairing** of sequences that retains the order of characters in each sequence, perhaps introducing gaps, such that the total score is optimal.

Local and Global Alignments

- Global alignment
  - find alignment in which the total score is highest, perhaps at the expense of areas of great local similarity.
- Local alignment
  - find alignment in which the highest scoring subsequences are identified, at the expense of the overall score.
  - Local alignment can be obtained by performing minor modifications to the global alignment algorithm.

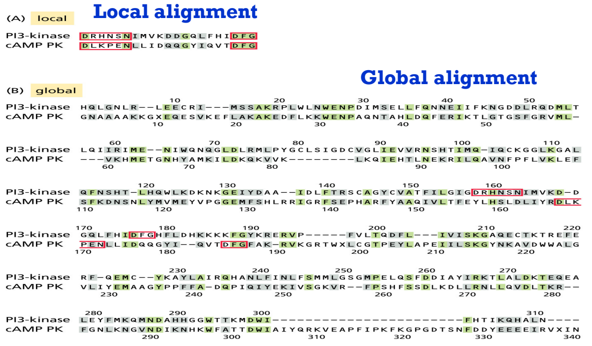
Shall we perform: Global or Local Alignment?

- **Global Alignment**:
  - Are these two sequences generally the same?
- **Local Alignment**:
  - Do these two sequences contain high scoring subsequences?
- Local similarities may occur in sequences with different structure or function that share common substructure or subfunction.

Local Sequence Alignment

- The **optimal local alignment** of two sequences is the one that finds the longest segment of high sequence similarity between the two sequences.

Example: Local and Global Alignments



Local alignment

```
(A) local

PI3-kinase  DRHNSNIMVKDDGQLFHIDFG
cAMP PK     DLKPENLLIDQQGYIQVTDFG
```

Global alignment

```
(B) global
                      10         20         30         40         50
PI3-kinase  HQLGNLR--LEECRI---MSSAKRPLWLNWENPDIMSELLFQNNEIIFKNGDDLRQDMLT
cAMP PK     GNAAAAKKGXEQESVKEFLAKAKEDFLKKWENPAQNTAHLDQFERIKTLGTGSFGRVML-
                    10         20         30         40         50

                    60         70         80         90        100        110
PI3-kinase  LQIIRIME--NIWQNQGLDLRMLPYGCLSIGDCVGLIEVVRNSHTIMQ-IQCKGGLKGAL
cAMP PK     ---VKHMETGNHYAMKILDKQKVVK---------LKQIEHTLNEKRILQAVNFPFLVKLEF
               60         70         80                 90        100

                       120        130          140        150        160
PI3-kinase  QFNSHT-LHQWLKDKNKGEIYDAA--IDLFTRSCAGYCVATFILGIGDRHNSNIMVKD-D
cAMP PK     SFKDNSNLYMVMEYVPGGEMFSHLRRIGRFSEPHARFYAAQIVLTFEYLHSLDLIYRDLK
            110        120        130        140        150        160

            170        180        190          200        210        220
PI3-kinase  GQLFHIDFGHFLDHKKKKFGYKRERVP-----FVLTQDFL---IVISKGAQECTKTREFE
cAMP PK     PENLLIDQQGYI--QVTDFGFAK-RVKGRTWXLCGTPEYLAPEIILSKGYNKAVDWWALG
            170        180          190        200        210        220

                   230        240        250        260        270
PI3-kinase  RF-QEMC--YKAYLAIRQHANLFINLFSMMLGSGMPELQSFDDIAYIRKTLALDKTEQEA
cAMP PK     VLIYEMAAGYPPFFA-DQPIQIYEKIVSGKVR--FPSHFSSDLKDLLRNLLQVDLTKR--
               230        240        250        260        270        280

              280        290        300                             310
PI3-kinase  LEYFMKQMNDAHHGGWTTKMDWI--------------------------FHTIKQHALN----
cAMP PK     FGNLKNGVNDIKNHKWFATTDWIAIYQRKVEAPFIPKFKGPGDTSNFDDYEEEEIRVXIN
                  290        300        310        320        330        340
```

Understanding Bioinformatics by Zvelebil and Baum

Methods of Alignment

A) By hand - slide sequences on two lines of a word processor

B) Dot plot (also known as dot matrix)

- with windows

C) Rigorous mathematical approach

- Dynamic programming (optimal but slow)

D) Heuristic methods (fast but approximate)

- BLAST and FASTA
  - Word matching and hash tables

A) Pairwise Sequence Alignment by Hand

- Write sequences across the page in two rows.
- Place identical or similar characters in the same column.
- Place non-identical characters either in the same column as a mismatch or opposite a gap in the other sequence.

B) Dot Matrix Method (I)

- Dot matrices are the simplest means of comparing two sequences.
- Dot matrices are designed to answer the following questions:
  - Where are all sites of similarity between my sequence and a second sequence?
  - Where are all sites of internal similarity in my sequence?
- Dot plots are not quantitative, they are qualitative.

The Dot Matrix Method (II)

- Dot plots place one sequence on the X axis, the other on the Y axis and compare the sequence on one axis with that on the other:
  - If the sequences match according to some criteria, a dot is placed at the XY intercept.
    - The dots populate a 2-dimentional space representing similarity between the sequences along the X and the Y axes.
- Dot plots present a visual representation of the similarity between two sequences, but do not give a numerical value to this similarity.

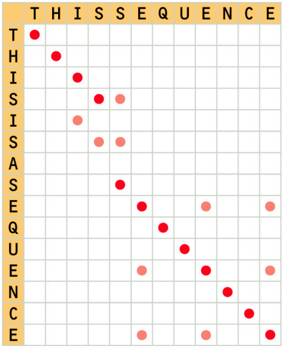
Dot Matrices

	A	T	G	C	C	T	A	G
A	*						*	
T		*				*		
G			*					*
C				*	*			
C				*	*			
T		*				*		
A	*						*	
G			*					*

Window Size = 1

The diagonal line always appears when a sequence is compared to itself.

Background Noise



Red dots represent identities that are meaningful – they are true matchings of identical residue-pairs.

Pink dots represent identities that are due to noise – they are matchings of random identical residues-pairs

Understanding Bioinformatics by Zvelebil and Baum

Improving Dot Matrices

- In a dot matrix, detection of matching regions may be improved by **filtering** out random matches.
- **Filtering** is achieved by using a sliding window to compare the two sequences.

Sliding Window

4

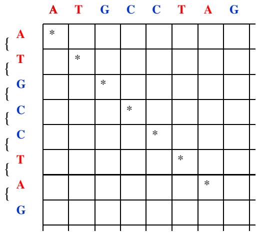
GAA CTCA TACGAATTCACATTAGAC

Window Size: Number of characters to compare

Stringency: Number of characters that have to match exactly

There are some defaults values for the window size and for the stringency, but one has to play around with the numbers to see what gives the best result.

Dot Matrices with Windows

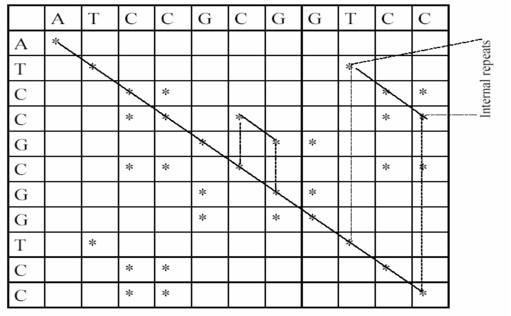


Window Size = 2

Compare two nucleotides at a time.

Windows **filter** out the noise.

Internal Repeats



Number of letters to compare (window)=1

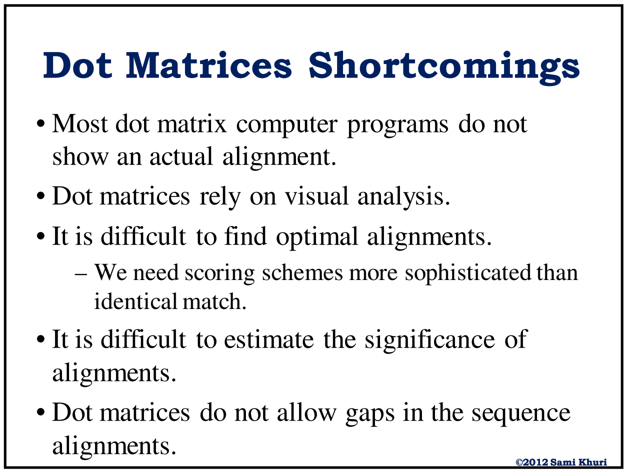
Determining Window Size

DNA Sequences

- A typical window size is15.
- A suitable match (stringency) requirement in this window is 10.

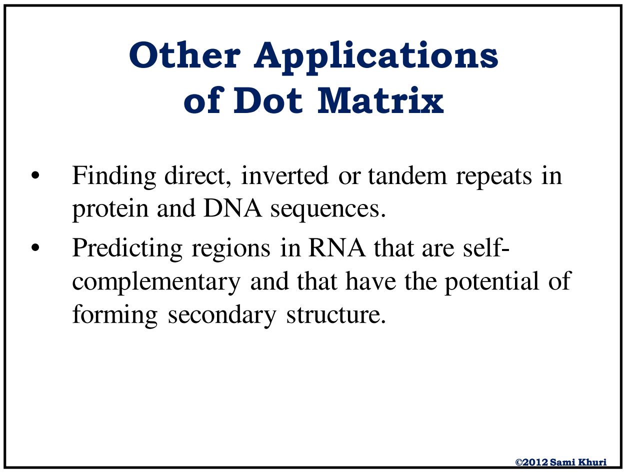
Protein Sequence

- Often the matrix is not filtered, but a window size requirement is 2 or 3.
- A match requirement of 2 will highlight matching regions.



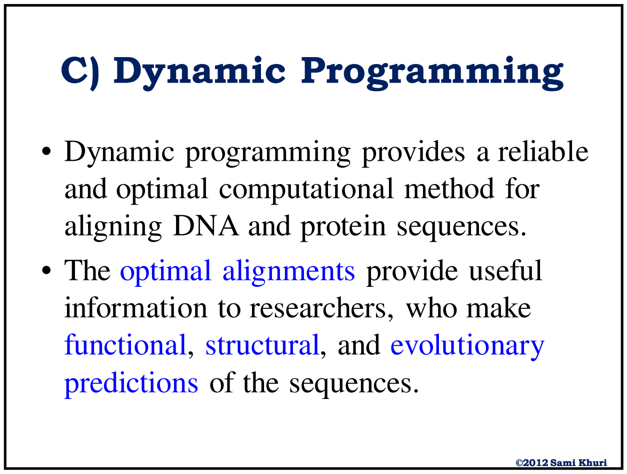
Dot Matrices Shortcomings

- Most dot matrix computer programs do not show an actual alignment.
- Dot matrices rely on visual analysis.
- It is difficult to find optimal alignments.
  - We need scoring schemes more sophisticated than identical match.
- It is difficult to estimate the significance of alignments.
- Dot matrices do not allow gaps in the sequence alignments.



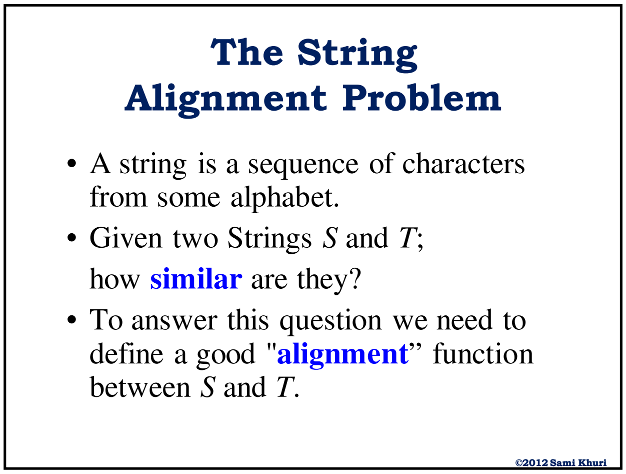
Other Applications of Dot Matrix

- Finding direct, inverted or tandem repeats in protein and DNA sequences.
- Predicting regions in RNA that are self-complementary and that have the potential of forming secondary structure.



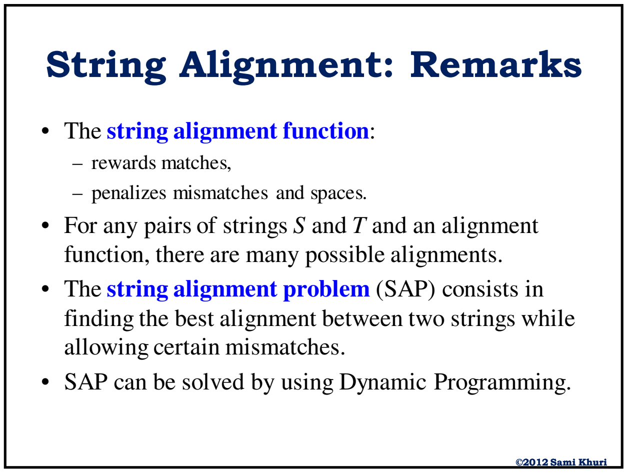
C) Dynamic Programming

- Dynamic programming provides a reliable and optimal computational method for aligning DNA and protein sequences.
- The optimal alignments provide useful information to researchers, who make functional, structural, and evolutionary predictions of the sequences.



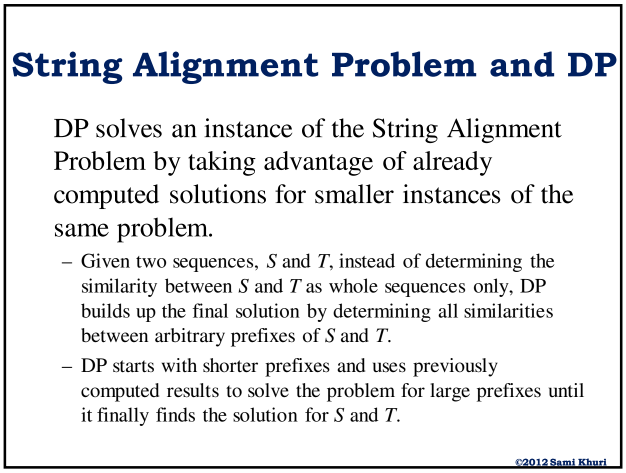
The String Alignment Problem

- A string is a sequence of characters from some alphabet.
- Given two Strings *S* and *T*;
  how **similar** are they?
- To answer this question we need to define a good "**alignment**” function between *S* and *T*.



String Alignment: Remarks

- The **string alignment function**:
  - rewards matches,
  - penalizes mismatches and spaces.
- For any pairs of strings *S* and *T* and an alignment function, there are many possible alignments.
- The **string alignment problem** (SAP) consists in finding the best alignment between two strings while allowing certain mismatches.
- SAP can be solved by using Dynamic Programming.



String Alignment Problem and DP

DP solves an instance of the String Alignment Problem by taking advantage of already computed solutions for smaller instances of the same problem.

- Given two sequences, *S* and *T*, instead of determining the similarity between *S* and *T* as whole sequences only, DP builds up the final solution by determining all similarities between arbitrary prefixes of *S* and *T*.
- DP starts with shorter prefixes and uses previously computed results to solve the problem for large prefixes until it finally finds the solution for *S* and *T*.

SAP: Optimal Alignments

- Use the alignment function previously seen.
- Given two strings *S* and *T* over some alphabet, with $|S| = n$ and $|T| = m$.

  Define $a(i, j)$ to be the value of an **optimal alignment** of strings:

  $S[1], S[2], \ldots, S[i]$ and

  $T[1], T[2], \ldots\ldots, T[j]$

  a(n, m) is the value of an **optimal alignment** of S and T.

SAP: Basis Relation

- The dynamic programming algorithm will compute each a(*i*, *j*), $0 \le i \le n$ and $0 \le j \le m$, only **once**, by considering the values already computed for smaller indexes *i* and *j*.
- Define

$$a(i,0) = \sum_{k=1}^{i} p(S[k], -)$$

and

$$a(0,j) = \sum_{k=1}^{j} p(-, T[k])$$

where *p* is the alignment function.
a(*i*,0) means that the first *i* characters of *S* are aligned with no characters of *T*. In other words, the *i* characters of *S* are matched with *i* spaces (i.e. "-"). Similarly for *a*(0, *j*).

SAP: Recurrence Relation

In general:

$$a(i,j) = \max \begin{cases} a(i-1, j-1) + p(S[i], T[j]) \\ a(i-1, j) + p(S[i], -) \\ a(i, j-1) + p(-, T[j]) \end{cases}$$

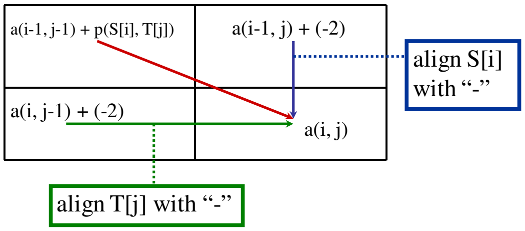
Recall: $p(S[i], T[j]) = \begin{cases} +1 \text{ if } S[i]=T[j] \\ -1 \text{ if } S[i] \neq T[j] \end{cases}$

and $p(S[i], -) = -2$
$p(-, T[j]) = -2.$

SAP: Computing a(n,m)

- DP uses a table of size (n+1) × (m+1).
- $a(i, j)$ corresponds to the optimal alignment of the i^{th} prefix of *S* with the j^{th} prefix of *T*.
- The dynamic programming algorithm fills in the entries of the table (matrix) by computing the values of $a(i, j)$ from top to bottom, left to right.
- The value of the optimal alignment is given by $a(n,m)$.

Filling Entry a(i,j) in the Table



DP: Bookkeeping and Retracing

- Draw lines crossing the entries in the matrix to show from which entry in the matrix we derived the maximum score for each entry a(i, j).
- To determine the solution of the optimal alignment, simply retrace the steps from entry a(*n*, *m*) to entry a(0, 0).

Global Alignment

Example

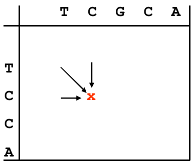
Align the following sequences:

S = TCCA
T = TCGCA

Solution

Use Needleman Wunsch Algorithm

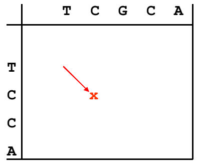
Three Possible Paths



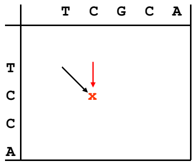
Any given point in the matrix can be reached from three possible positions.

=> Best scoring alignment ending in any given point in the matrix can be found by choosing the highest scoring of the three possibilities.

Computing the Score (I)

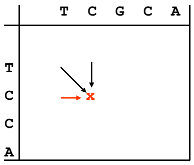


$$a(i,j) = \max \begin{cases} a(i\text{-}1,j\text{-}1) + p(i,j) \end{cases}$$



$$a(i,j) = \max \begin{cases} a(i\text{-}1,j\text{-}1) + p(i,j) \\ a(i\text{-}1,j) - (\text{gap\_penalty}) \end{cases}$$

Computing the Score (II)

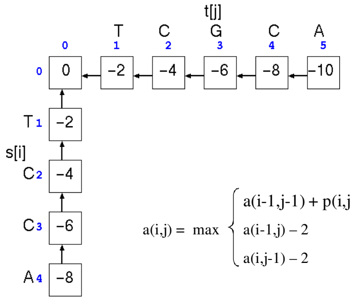


$$a(i,j) = \max \begin{cases} a(i\text{-}1,j\text{-}1) + p(i,j) \\ a(i\text{-}1,j) - (\text{gap\_penalty}) \\ a(i,j\text{-}1) - (\text{gap\_penalty}) \end{cases}$$

Each new score is found by choosing the maximum of three possibilities. For each square in the matrix: keep track of where the best score came from.

Fill in scores one row at a time, starting in upper left corner of matrix, ending in lower right corner.

Needleman Wunsch: Example (I)



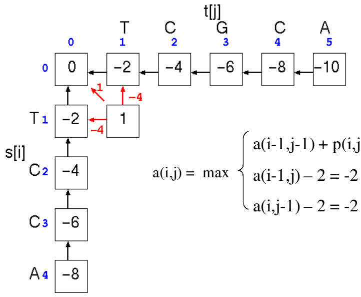
$$a(i,j) = \max \begin{cases} a(i\text{-}1,j\text{-}1) + p(i,j) \\ a(i\text{-}1,j) - 2 \\ a(i,j\text{-}1) - 2 \end{cases}$$

Penalty Function

	A	C	G	T
A	1	−1	−1	−1
C	−1	1	−1	−1
G	−1	−1	1	−1
T	−1	−1	−1	1

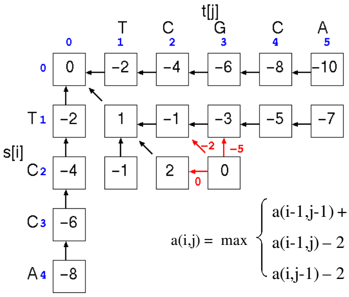
Gaps: −2

Needleman Wunsch: Example (II)



$$a(i,j) = \max \begin{cases} a(i\text{-}1,j\text{-}1) + p(i,j) = 0 + 1 \\ a(i\text{-}1,j) - 2 = -2\ -2 \\ a(i,j\text{-}1) - 2 = -2\ -2 \end{cases}$$

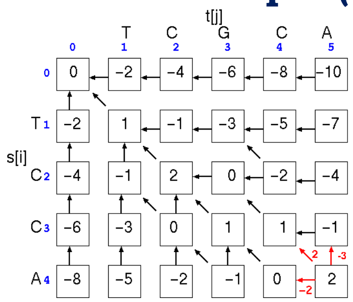
Needleman Wunsch: Example (III)



			T	C	G	C	A
		0	1	2	3	4	5
	0	0	−2	−4	−6	−8	−10
T	1	−2	1	−1	−3	−5	−7
C	2	−4	−1	2	0		
C	3	−6					
A	4	−8					

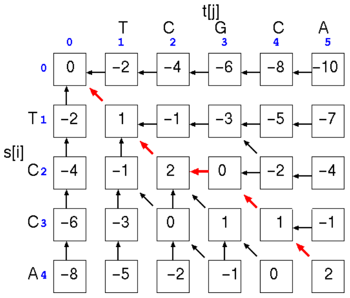
$$a(i,j) = \max \begin{cases} a(i-1,j-1) + p(i,j) = -1 + (-1) \\ a(i-1,j) - 2 = -3 + (-2) \\ a(i,j-1) - 2 = 2 + (-2) \end{cases}$$

Needleman Wunsch: Example (IV)



			T	C	G	C	A
		0	1	2	3	4	5
	0	0	−2	−4	−6	−8	−10
T	1	−2	1	−1	−3	−5	−7
C	2	−4	−1	2	0	−2	−4
C	3	−6	−3	0	1	1	−1
A	4	−8	−5	−2	−1	0	2

Needleman Wunsch: Example (V)



			T	C	G	C	A
		0	1	2	3	4	5
	0	0	−2	−4	−6	−8	−10
T	1	−2	1	−1	−3	−5	−7
C	2	−4	−1	2	0	−2	−4
C	3	−6	−3	0	1	1	−1
A	4	−8	−5	−2	−1	0	2

Solution:

```
T C – C A
: :   : :
T C G C A
1+1–2+1+1 = 2
```

Drawback of DP for SAP

- The major drawback of dynamic programming is the fact that the table of size (n+1)×(m+1) uses $O(nm)$ space.
- It is easy to compute a(n, m) in linear space since all we have to do at any given time during the computation is save two rows of the matrix, not more.
- The only values needed when computing $a(i, j)$ are found in rows i and i-1.
- But it is not easy to find the optimal alignment in linear space.

Sub-Optimal Alignment

- The best alignment from a biological point of view, may not be the best alignment from a computational point of view.
- The ultimate goal is to align **functional** regions.
- The software can only align regions of sequence **similarity**.
- Sub-optimal alignments may not have the best sequence alignment, but may have helical regions or active sites aligned better than the “optimal” alignment.

Global Alignment

- The **dynamic programming** method we studied so far was designed by Needleman and Wunsch (1970).
- Their dynamic algorithm gives a **global alignment** of sequences.
- We now turn our attention to **local alignments**.

Local Alignment (I)

- A modification of the dynamic programming algorithm for sequence alignment provides a **local sequence alignment** giving the highest-scoring local match between two sequences (Smith and Waterman 1981).
- Local alignments are usually more meaningful than global matches because they include patterns that are conserved in the sequences.

Local Alignment (II)

- The rules for calculating scoring values are slightly different with local alignment.
- The most important difference being:
  - Recall that the scoring system must include negative scores for mismatches
- With local alignment, when a dynamic programming scoring matrix value becomes negative, that value is set to zero, which has the effect of terminating any alignment up to that point.

Global and Local Alignments

- **Global Alignment**:
  - Are these two sequences generally the same?
- **Local Alignment**:
  - Do these two sequences contain high scoring subsequences?
- Local similarities may occur in sequences with different structure or function that share common substructure or subfunction.

Local Sequence Alignment

- Their dynamic algorithm gives a global alignment of sequences.
- A modification of the dynamic programming algorithm for sequence alignment provides a local sequence alignment giving the highest-scoring local match between two sequences (Smith and Waterman 1981).
- Local alignments are usually more meaningful than global matches because they include patterns that are conserved in the sequences.

Local Alignments

		G	A	A	C	G	T	A	G	G	C	G	T	A	T
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
A	0	0	1	1	0	0	0	1	0	0	0	0	0	1	0
T	0	0	0	0	0	0	1	0	0	0	0	0	1	0	2
A	0	0	1	1	0	0	0	2	0	0	0	0	0	2	0
C	0	0	0	0	2	0	0	0	1	0	1	0	0	0	1
T	0	0	0	0	0	1	1	0	0	0	0	0	1	0	1
A	0	0	1	1	0	0	0	2	0	0	0	0	0	2	0
C	0	0	0	0	2	0	0	0	0	0	1	0	0	0	0
G	0	1	0	0	0	3	1	0	1	1	0	2	0	0	0
G	0	1	0	0	0	1	2	0	0	2	0	1	1	0	0
A	0	0	2	1	0	0	0	3	1	0	0	0	0	2	0
G	0	1	0	0	0	1	0	0	4	2	0	1	0	0	0
G	0	1	0	0	0	1	0	0	1	5	3	1	0	0	0
G	0	1	0	0	0	1	0	0	1	2	4	4	2	0	0

Thus, the best local alignment achieved from the above Dynamic Programming is:

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

Scoring Systems

- Use of the **dynamic programming** method requires a scoring system for
  - the comparison of symbol pairs (**nucleotides** for DNA sequences & **amino acids** for protein sequences),
  - a scheme for insertion/deletion (gap) penalties.
- The most commonly used scoring systems for protein sequence alignments are the log odds form
  - of the **PAM250** matrix and
  - the **BLOSUM62** matrix.
- A number of other choices are available.

Successful Substitution Matrices

- The most successful **substitution matrices** are the ones that use actual evidence of what has happened during evolution, and are based on the analysis of alignments of numerous homologs of well-suited proteins from many different species.

Understanding Bioinformatics by Zvelebil and Baum

©2012 Sami Khuri

Scoring Matrices (I)

- Upon evaluating a sequence alignment, we are really interested in knowing whether the alignment is random or meaningful.
- A **scoring matrix** (table) or a **substitute matrix** (table) is a table of values that describe the probability of a residue (amino acid or base) pair occurring in an alignment.

©2012 Sami Khuri

Scoring Matrices (II)

- The alignment algorithm needs to know if it is more likely that a given amino acid pair has occurred **randomly** or that it has occurred as a result of an **evolutionary** event.
- Similar amino acids are defined by high-scoring matches between the amino acid pairs in the substitution matrix.

©2012 Sami Khuri

The Roles of the Scoring Matrices

The quality of the alignment between two sequences is calculated using a **scoring system** that favors the matching of related or identical amino acids and penalizes poorly matched amino acids and gaps.

©2012 Sami Khuri

Amino Acid Substitution Matrices (I)

- For proteins, an **amino acid substitution matrix**, such as the Dayhoff percent accepted mutation matrix 250 (**PAM250**) or BLOSUM substitution matrix 62 (**BLOSUM62**) is used to score matches and mismatches.
- Similar matrices are available for aligning DNA sequences.

©2012 Sami Khuri

Amino Acid Substitution Matrices (II)

- In the **amino acid substitution matrices**, amino acids are listed both across the top of a matrix and down the side, and each matrix position is filled with a score that reflects how often one amino acid would have been paired with the other in an alignment of related protein sequences.

©2012 Sami Khuri

PAM Matrices

Point Accepted Mutation

- An **accepted mutation** is any mutation that doesn't kill the protein or organism; that is, amino acid changes "accepted" by natural selection.

One PAM (**PAM1**) = 1% of the amino acids have been changed.

Dayhoff Amino Acid Substitution Matrices

- **PAM Matrices** are Dayhoff amino acid substitution or percent accepted mutation matrices.
- This family of matrices lists the likelihood of change from one amino acid to another in homologous protein sequences during evolution.
- These predicted changes are used to produce **optimal alignments** between two protein sequences and to score the alignment.

Extrapolating PAM1

The assumption in this evolutionary model is that the amino acid substitutions observed over short periods of evolutionary history can be extrapolated to longer distances.

Constructing More PAM Matrices

- The **PAM1** Matrix is best used for comparing sequences where 1% or less of the amino acids have changed.
- What do you do with sequences that are more divergent?
- You multiply the PAM1 matrix by itself N times to get a new matrix that works best with sequences that have PAM2, PAM20, PAM100, PAM200, etc.
- For example PAM20 = $(\text{PAM1})^{20}$

PAM Matrices for Low Level of Similarities

- As seen, **PAM1** matrix could be multiplied by itself N times, to give transition matrices for comparing sequences with lower and lower levels of similarity due to separation of longer periods of evolutionary history.
- The PAM120, PAM80, and PAM60 matrices should be used for aligning sequences that are 40%, 50%, and 60% similar, respectively.

PAM250 Matrix

- The PAM250 matrix provides a better-scoring alignment than lower-numbered PAM matrices for distantly related proteins of 14-27% similarity.
- Scoring matrices are also used in database searches for similar sequences.

How Good are PAM Matrices?

- The Dayhoff PAM matrices have been criticized because they are based on a small set of closely related proteins.
- Scoring matrices obtained more recently, such as the **BLOSUM** matrices, are based on a much larger number of protein families.

BLOSUM vs PAM

- The BLOSUM matrix was constructed from actual substitutions.
- The BLOSUM matrix was derived from much more recently than the Dayhoff matrices, in the early 1990's, using local multiple alignments rather than global alignments.

Understanding Bioinformatics by Zvelebil and Baum

BLOSUM Matrices

- The **BLOSUM** scoring matrices (especially BLOSUM62) appear to capture more of the distant types of variations found in protein families.
- Another criticism: PAM scoring matrices are not much more useful for sequence alignment than simpler matrices, such as the ones based on chemical grouping of amino acid side chains.

BLOSUM

- **Blo**cks **Sum**
  - created from BLOCKS database.
- Currently the most widely used comparison matrix.
- More sensitive than PAM or other matrices
- Finds more sequences that are related
- The BLOSUM matrices are based on an entirely different type of sequence analysis and a much larger data set than the Dayhoff PAM Matrices.

BLOSUM II

- The protein families were originally identified by Bairoch in the Prosite catalog.
- The catalog provides lists of proteins that are in the same family because they have a similar biochemical function.

Families of Related Proteins

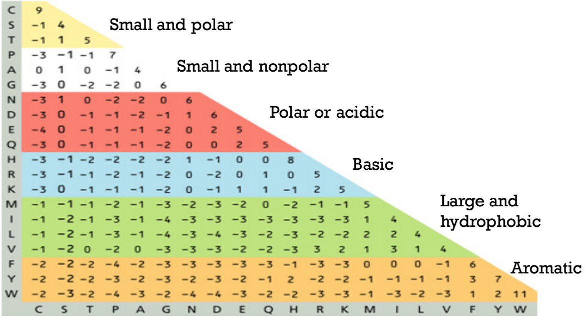
- The matrix values are based on the observed amino acid substitutions in around 2000 conserved amino acid patterns, called blocks.
- The blocks were found in a database of protein sequences (Prosite) representing more than 500 families of related proteins and act as signatures of these protein families.

BLOSUM62

- The blocks that characterized each family provided a type of multiple sequence alignment for that family.
- **BLOSUM62** represents a balance between information content and data size.

©2012 Sami Khuri

BLOSUM62 Substitution Matrix



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

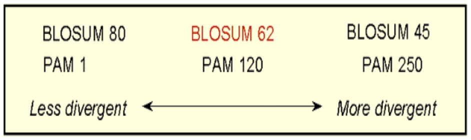
Small and polar
Small and nonpolar
Polar or acidic
Basic
Large and hydrophobic
Aromatic

Colors indicate different physicochemical properties of the residues.

©2012 Sami Khuri

Comparison: PAM and BLOSUM Matrices

The **PAM** model is designed to track the evolutionary origins of proteins, whereas the **BLOSUM** model is designed to find their conserved domains.



©2012 Sami Khuri

D) Approximate Methods BLAST

- **B**asic **L**ocal **A**lignment **S**earch **T**ool
  - –Altschul et al. 1990,1994,1997
- Heuristic method for local alignment
- Designed specifically for database searches
- Idea: Good alignments contain short lengths of exact matches.

©2012 Sami Khuri

The BLAST Family

- **blastp**: compares an amino acid query sequence against a protein sequence database.
- **blastn**: compares a nucleotide query sequence against a nucleotide sequence database.
- **blastx**: compares the six-frame conceptual translation products of a nucleotide query sequence (both strands) against a protein sequence database.

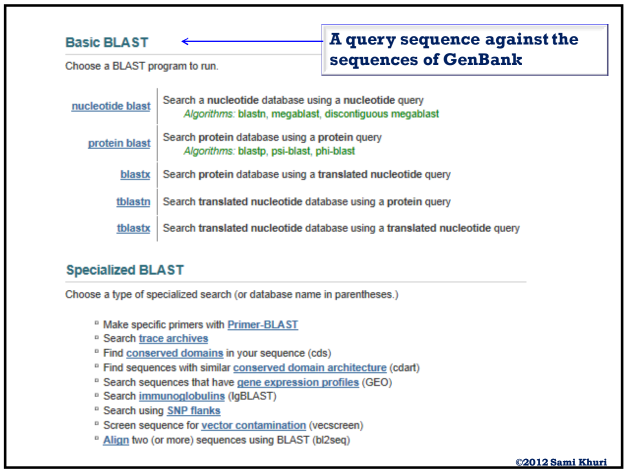
©2012 Sami Khuri

Chance or Homology?

- In all methods of sequence comparison, the fundamental question is whether the similarities perceived between two sequences are due to chance, and are thus of little biological significance, or whether they are due to the derivation of the sequences from a common ancestral sequence, and are thus homologous.

Understanding Bioinformatics by Zvelebil and Baum

©2012 Sami Khuri



A query sequence against the sequences of GenBank

Basic BLAST

Choose a BLAST program to run.

nucleotide blast	Search a **nucleotide** database using a **nucleotide** query *Algorithms:* blastn, megablast, discontiguous megablast
protein blast	Search **protein** database using a **protein** query *Algorithms:* blastp, psi-blast, phi-blast
blastx	Search **protein** database using a **translated nucleotide** query
tblastn	Search **translated nucleotide** database using a **protein** query
tblastx	Search **translated nucleotide** database using a **translated nucleotide** query

Specialized BLAST

Choose a type of specialized search (or database name in parentheses.)

- Make specific primers with Primer-BLAST
- Search trace archives
- Find conserved domains in your sequence (cds)
- Find sequences with similar conserved domain architecture (cdart)
- Search sequences that have gene expression profiles (GEO)
- Search immunoglobulins (IgBLAST)
- Search using SNP flanks
- Screen sequence for vector contamination (vecscreen)
- Align two (or more) sequences using BLAST (bl2seq)

The Expected Value

```
SW:P11A BOVIN P32871 PHOSPHATIDYLINOSITOL 3-KINAS  (1068) 2228 493 1.2e-138
SW:P11A HUMAN P42336 PHOSPHATIDYLINOSITOL 3-KINAS  (1068) 2216 490 7.4e-138
SW:P11A MOUSE P42337 PHOSPHATIDYLINOSITOL 3-KINAS  (1068) 2204 488 4.5e-137
SW:P11B HUMAN P42338 PHOSPHATIDYLINOSITOL 3-KINAS  (1070) 1126 254 1.1e-66
```

The **e-value** tells us how likely it is that the similarity between the query sequence and the database sequence is due to chance.

The lower the **e-value**, the more likely it is that the two sequences are truly similar and not just chance matches