



The Kalmanovitz Library and
The Center for Knowledge Management

BLAST: Design and Implementation

Gilberto da Gente
Bioinformatics Resource Specialist
Wednesday, July 9th, 2008
University of California, San Francisco



UCSF

Learning Objectives

- Local and Global alignment
- Dynamic programming
- BLAST Algorithm
- PHI-BLAST
- PSI-BLAST

Sequence Alignment

Basic Example

```
AAGCTGAATTCGAA  
AGGCTCATTCTGA
```

One possible alignment:

```
AAGCTGAATT-C-GAA  
AGGCT-CATTCTGA-
```

This alignment includes:

```
10 Matches  
2 Mismatches  
4 Indels
```

Other possible alignments:

```
A-AGCTGAATTC--GAA  
AG-GCTCA-TTTCTGA-
```

```
AAGCTGAATT-C-GAA  
AGGCT-CATTCTGA-
```

Which alignment is better?

Local vs. Global Alignment

- Global alignment – finds the best alignment across the whole two sequences.

Global alignment: forces alignment in regions which differ

```
ADLGAVFALCDRYFQ
|||||
ADLGRNQNCDRYYQ
```

- Local alignment – finds regions of similarity in small parts of the sequences.

```
ADLG          CDRYFQ
|||||        |||||
ADLG          CDRYYQ
```

Local alignment will return only regions of alignment

Dynamic Programming

and Optimal Alignment

- In mathematics and computer science, **dynamic programming** is a method of solving problems exhibiting the properties of **overlapping sub problems**
 - ✓ takes much less time than simple methods
- The Needleman–Wunsch algorithm is an example of dynamic programming
 - ✓ First application of dynamic programming to biological sequence comparison.

Needleman–Wunsch

- The Needleman–Wunsch algorithm performs a global alignment on two sequences
- The algorithm was proposed in 1970 by Saul Needleman and Christian Wunsch in their paper “A general method applicable to the search for similarities in the amino acid sequence of two proteins”, J Mol Biol. 48(3):443-53.
- Involves three steps
 - ✓ Matrix Initiation
 - ✓ Matrix fill
 - ✓ Matrix traceback

Matrix Initiation

- The first step in the global alignment dynamic programming approach is to **create a matrix** with $M + 1$ columns and $N + 1$ rows where M and N correspond to the size of the sequences to be aligned.
- Zero scores are placed in the first row and first column

	G	A	A	T	T	C	A	G	T	T	A
	0	0	0	0	0	0	0	0	0	0	0
G	0										
G	0										
A	0										
T	0										
C	0										
G	0										
A	0										

Matrix fill

For each position, $M_{i,j}$ is defined to be the maximum score at position i,j ;

$$M_{i,j} = \text{MAXIMUM of [}$$

- $M_{i-1,j-1} + S_{i,j}$ (match/mismatch in the diagonal)
- $M_{i,j-1} + w$ (gap in sequence #1)
- $M_{i-1,j} + w$ (gap in sequence #2)]

Two different ways to get the maximum score. In such a case, pointers are placed back to all of the cells that can produce the maximum score.

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

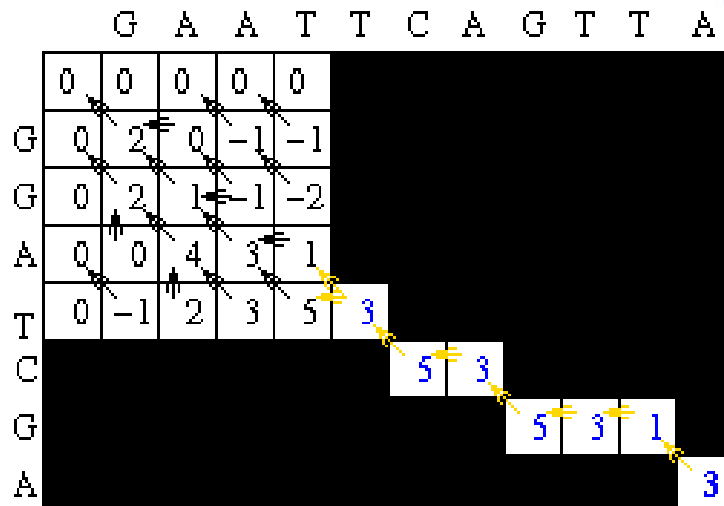
Matrix Traceback

- After the matrix fill step, the maximum global alignment score for the two sequences is 3.
- The traceback step will determine the actual alignment(s) that result in the maximum score.

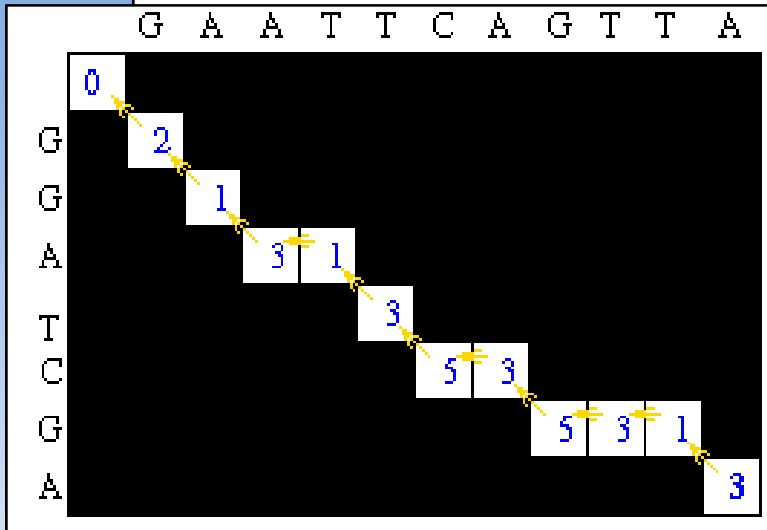
	G	A	A	T	T	C	A	G	T	T	A
	0	0	0	0	0	0	0	0	0	0	0
G	0	2	0	-1	-1	-1	-1	2	0	-1	
G	0	2	1	-1	-2	-2	-2	1	1	-1	
A	0	0	4	3	1	-1	-3	0	-1	0	0
T	0	-1	2	3	5	3	1	-1	-1	1	2
C	0	-1	0	1	3	4	5	3	1	-1	0
G	0	2	0	-1	1	2	3	4	5	3	1
A											3

Matrix Traceback

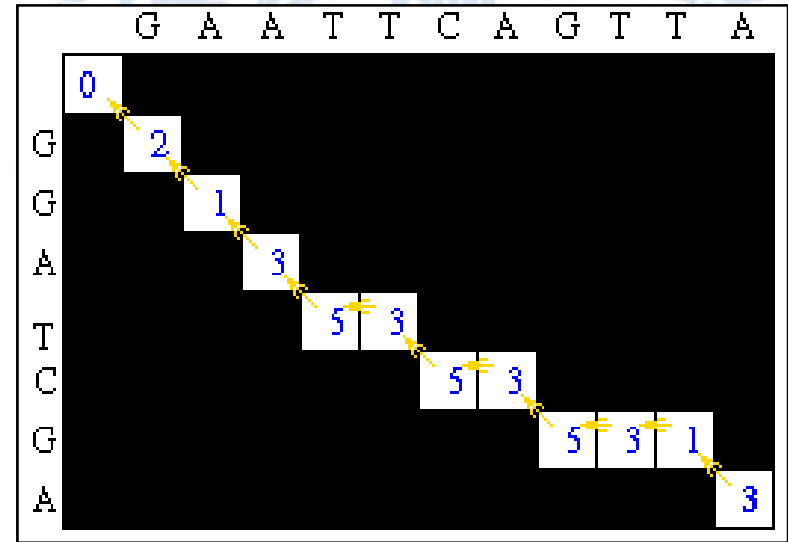
- The traceback step begins in the M,J position in the matrix, i.e. the position where both sequences are globally aligned.
- Since we have kept pointers back to all possible predecessors, the traceback step is simple. At each cell, we look to see where we move next according to the pointers



Optimal Global Alignments



G A A T T C A G T T A
 | | | | | | | |
 G G A _ T C _ G _ _ A



G A A T T C A G T T A
 | | | | | | | |
 G G A T _ C _ G _ _ A

Smith-Waterman

- The algorithm was first proposed by Temple Smith and Michael Waterman in 1981 in their paper "Identification of Common Molecular Subsequences". *Journal of Molecular Biology* 147: 195-197.
- Like the Needleman-Wunsch algorithm Smith-Waterman is a dynamic programming algorithm

Smith-Waterman

- The main difference to the Needleman-Wunsch algorithm is that **negative scoring matrix cells are set to zero**, which renders the local alignments visible
- **Backtracing starts at the highest scoring matrix cell and proceeds until a cell with score zero is encountered**, yielding the highest scoring local alignment.

Optimal Local Alignment

$$M_{i,j} = \text{MAXIMUM of}$$

$$M_{i-1,j-1} + S_{i,j} \quad \text{or}$$

$$M_{i,j-1} + w \quad \text{or}$$

$$M_{i-1,j} + w \quad \text{or}$$

$$0$$

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

GAATTCAGTTA

|||||||

GGATCGA

GAATTC_AGTTA

||| || |

GGA_TCGA

Perfect Match: +2

Mismatch: -1

Indel: -2

BLAST

- Basic Local Alignment and Search Tool
- Heuristic approach based on Smith Waterman dynamic programming algorithm
- Ubiquitous similarity search tool
- BLAST is a collection of programs
 - ✓ DNA vs DNA
 - ✓ DNA translation vs Protein
 - ✓ Protein vs Protein
 - ✓ Protein vs DNA translation
 - ✓ DNA translation vs DNA translation
- www, standalone, and network clients

BLAST Terminology

- **Segment**—a substring of a sequence
- **Segment pair of two sequences**—pair of segments of the same length (no gaps), one from each sequence
- **w-mer**—a substring (or word) of w characters
- **High scoring segment pairs** are called **HSPs**
- The highest scoring segment pair for the whole pairwise comparison is referred to as the **maximal-scoring segment pair (MSP)**

Steps in the Algorithm

1. Compile a list of high-scoring words in the query sequence
2. Find matches in the database for each high-scoring word
3. For each match in the database, extend the alignment in both directions

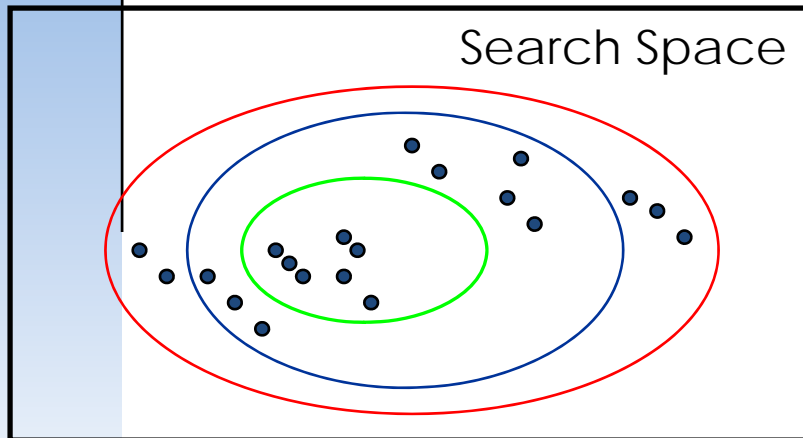
Step 1

- Compile a list of high-scoring words in the query sequence
- Defaults of $w=3$ for proteins, and $w=11$ for nucleic acid sequences
- The total number of words will be $n-w+1$
- Each word has a score t toward the query sequence computed using scoring matrix
- Threshold T : t -scores above T for any word pair indicates synonyms (**T is called the neighborhood word score threshold**)

Step 1: Visual

Query: IETVYAAAYLPKNTHPFL**YLS**LEISPQNVDVNVHPTKHEV**HFL**HEESILEV... w=3

Neighborhood Score Threshold	YLS	15	Neighborhood Words	HFL	18
	YLT	12		HFV	15
	<u>YVS</u>	<u>12</u>		HFS	14
	YIT	10		HWL	13
	etc ...			NFL	13
T = 11			<u>DFL</u>	<u>12</u>	
			HWV	10	
			etc ...		



● Word Hit

T = 13

T = 15

T = 19

Adjusting **T** can control the size of the neighborhood and the number of word hits in the search space.

Step 2

- For each word or synonym from the query, search for a hit in all database sequences
- Each hit is considered a seed alignment and is extended in both directions as long as the score of the alignment is increased
- If the score for the segment pair is higher than a threshold S , the score and the endpoints are stored.

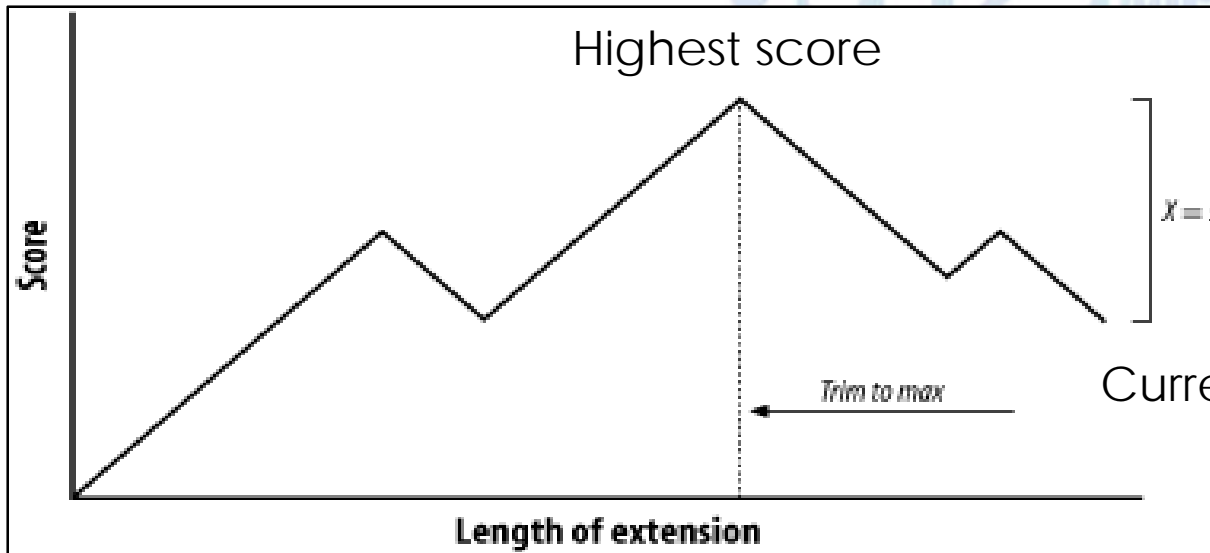
Step 2 : Visual

Seeds



Seed Extension

Query 1 IETVYAAYLPKNTHPFLYLSLEIS PQNV D VNVHPTKHEVHFLHEESI 47
 +E YA YL K F+YLSL +SP+ +D VNVHP+K VHFL+++ I
 Sbjct 287 LEETYAKYLHKGASYFVYLSLNMSPEQLD VNVHPSKRIVHFLYDQEI 333



X = Drop-off value for alignment (in bits) blastn 30, megablast 20, tblastx 0, all others 15

Step 3

- The HSP's of the entire database are compared to a cutoff score S , and those greater than S , are returned.
- Compute the statistical significance of each HSP score.
- If the calculated $E()$ for the database sequence meets the user given $E()$ for the program, this score is reported.

BLAST ver. 2

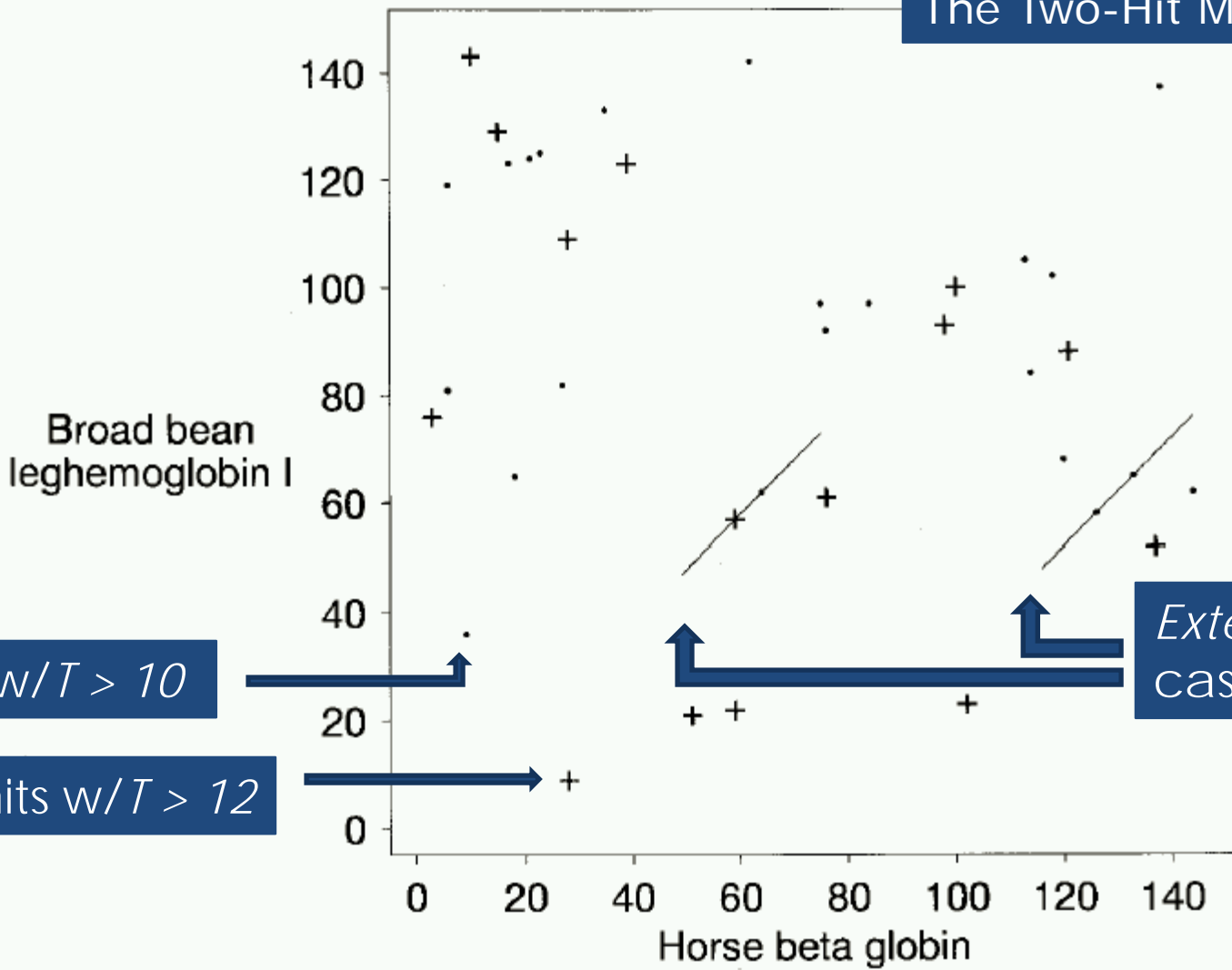
- Altshul et al. (1997)
- Improvements:
 - Two hit method for seeding => faster
 - Ability to generate gapped alignments => faster, better sensitivity
 - Iterated searching with position specific score matrix => better sensitivity

Two Hit Method

- Extension step typically accounts for 90% of BLAST's execution time
- Key idea: do extension only when there are two hits on the same diagonal within distance A of each other
- To maintain sensitivity, lower T parameter
 - ✓ more single hits found
 - ✓ but only small fraction have associated 2nd hit

Two Hit Method: Visual

The Two-Hit Method



hits $w/T > 10$

hits $w/T > 12$

Extend these cases

Gapped Alignments

- BLAST v1
 - Finds several alignments involving a single database sequence
 - When alignments are combined (no gaps), resulting alignment is statistically significant
 - When the alignments are not combined (gaps), individual alignments may not meet statistical threshold to be reported
- BLAST v2
 - Introduces an algorithm to generate gapped alignments overcoming issues with BLAST v1
 - Allowing gaps means that similar regions are not broken into several segments.
 - Gapped alignment algorithm uses DP to extend a central pair of aligned residues in both directions confined to a pre-defined strip of the DP path graph
 - Allows T to be raised increasing speed of initial database scan

PSI-BLAST & PHI-BLAST

Position-Specific Iterative & Pattern Hit Initiated

- Pattern or profile search methods are much more sensitive than pairwise comparison methods at detecting distant relationships
- Basic Idea:
 - BLAST searches may be iterated, with a position-specific score matrix generated from significant alignments found in round i used for round $i + 1$

Definition of Pattern or Motif

- An analysis representing the extent to which something exhibits various characteristics

Accepted PHI-BLAST Pattern Vocabulary	
ABCDEFGHIJKLMNPQRSTVWXYZU	Protein alphabet
ACGT	DNA alphabet
[]	means any one of the characters enclosed in the brackets e.g., [LFYT] means one occurrence of L or F or Y or T
-	nothing, used as a spacer to clearly separate each position
x	with nothing following means any residue
(n)	means the preceding residue is repeated n times
(m, n)	the preceding residue is repeated between m to n times (n > m)
>	only at the end of a pattern and means nothing it may occur before a period
.	may be used at the end, means nothing

<http://us.expasy.org/tools/scanprosite/scanprosite-doc.html>

Definition of Pattern or Motif

[LIVMF]-G-E-x-[GAS]-[LIVM]-x(5,11)-R-[STAQ]-A-x-[LIVMA]-x-[STACV]

Pattern Position	Pattern Syntax	Meaning
1	[LIVMF]	one of LIVMF
2	G	G
3	E	E
4	X	any one residue
5	[GAS]	one of GAS
6	[LIVM]	one of LIVM
7	X(5,11)	5 to 11 any residue
8	R	R
9	[STAQ]	one of STAQ
10	A	one A
11	X	any one residue
12	[LIVMA]	one of LIVMA
13	X	any one residue
14	[STACV]	any one of STACV

Note: total length of this motif/pattern is between 18 to 24 residues.

N-glycosylation site
N-{P}-[ST]-{P}

Glycosaminoglycan attachment site
S-G-x-G

cAMP- and cGMP-dependent protein kinase phosphorylation site
[RK](2)-x-[ST]

Position-Specific Scoring Matrix

- A PSSM is a motif descriptor
- The description includes a weight (score, probability, likelihood) for each symbol occurring at each position along the motif
- Examples of motifs:
 - Protein active sites, structural elements, zinc finger, intron/exon boundaries, transcription-factor binding sites, etc.

Position-Specific Scoring Matrix

Construction of PSSM is a multi-stage process:

1. Architecture of matrix
2. Create multiple alignment from which the matrix is derived
3. Calculate frequencies for each position
4. Applying BLAST to PSSM

Position-Specific Scoring Matrix

- 10 vertebrate donor site sequences aligned at exon/intron boundary

seq 1	GAGGTA AAC
seq 2	TCCGTA AGT
seq 3	CAGGTTGGA
seq 4	ACAGTCAGT
seq 5	TAGGTCATT
seq 6	TAGGTA CTG
seq 7	ATGGTA ACT
seq 8	CAGGTATAC
seq 9	TGTGTGAGT
seq 10	AAGGTAAGT

Position-Specific Scoring Matrix

- Calculate the absolute frequency of each nucleotide at each position

seq 1	GAGGTA AAC
seq 2	TCCGTA AGT
seq 3	CAGGTTG GA
seq 4	ACAGTCAGT
seq 5	TAGGTCATT
seq 6	TAGGTA CTG
seq 7	ATGGTAACT
seq 8	CAGGTATAC
seq 9	TGTGTGAGT
seq 10	AAGGTAAGT



	1	2	3	4	5	6	7	8	9
A									
C									
G									
T									

Position-Specific Scoring Matrix

- Calculate the absolute frequency of each nucleotide at each position

seq 1	GAGGTA AAC
seq 2	TCCGTA AGT
seq 3	CAGGTTG GA
seq 4	ACAGTCAGT
seq 5	TAGGTCATT
seq 6	TAGGTACTG
seq 7	ATGGTAACT
seq 8	CAGGTATAC
seq 9	TGTGTGAGT
seq 10	AAGGTAAGT



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

Position-Specific Scoring Matrix

- Calculate the relative frequency of each nucleotide at each position

seq 1	GAGGTA AAC
seq 2	TCCGTA AGT
seq 3	CAGGTTG GA
seq 4	ACAGTCAGT
seq 5	TAGGTCATT
seq 6	TAGGTA CTG
seq 7	ATGGTA ACT
seq 8	CAGGTAT AC
seq 9	TGTGTGAGT
seq 10	AAGGTAAGT



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



	1	2	3	4	5	6	7	8	9
A									
C									
G									
T									

Position-Specific Scoring Matrix

- Calculate the relative frequency of each nucleotide at each position

seq 1	GAGGTA AAC
seq 2	TCCGTA AGT
seq 3	CAGGTTG GA
seq 4	ACAGTCAGT
seq 5	TAGGTCATT
seq 6	TAGGTACTG
seq 7	ATGGTAACT
seq 8	CAGGTATAC
seq 9	TGTGTGAGT
seq 10	AAGGTAAGT



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



	1	2	3	4	5	6	7	8	9
A	0.3	0.6	0.1	0	0	0.6	0.7	0.2	0.1
C	0.2	0.2	0.1	0	0	0.2	0.1	0.1	0.2
G	0.1	0.1	0.7	1	0	0.1	0.1	0.5	0.1
T	0.4	0.1	0.1	0	1	0.1	0.1	0.2	0.6

Position-Specific Scoring Matrix

- What is the probability of finding CAGGTTGGA?
 - ✓ The product of the frequency of each nucleotide at each position:
 - ✓ C is 0.2 at position 1, A is 0.6 at position 2, etc -> $0.2 * 0.6 * 0.7 * 1 * 1 * 0.1 * 0.1 * 0.5 * 0.1$

	1	2	3	4	5	6	7	8	9
A	0.3	0.6	0.1	0	0	0.6	0.7	0.2	0.1
C	0.2	0.2	0.1	0	0	0.2	0.1	0.1	0.2
G	0.1	0.1	0.7	1	0	0.1	0.1	0.5	0.1
T	0.4	0.1	0.1	0	1	0.1	0.1	0.2	0.6