Database search

Gabriella Trucco

Email: gabriella.trucco@unimi.it

DB search

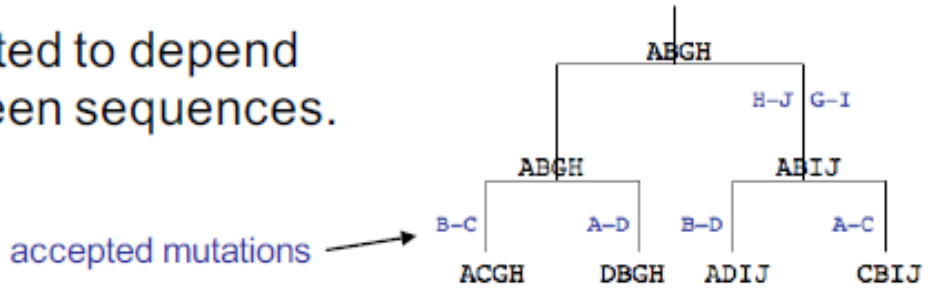
- Databases created to store the large quantity of sequence data
- Need for efficient programs to be used in queries of these databases
- *query* sequence that must be compared to all those already in the database, in search of local similarities
- The quadratic complexity: methods unsuitable for searching large databases
- To speed the search, novel and faster methods based on heuristics have been developed

PAM matrices

- When comparing protein sequences, simple scoring schemes are not enough.
  - Amino acids have biochemical properties that influence their relative replacebility in an evolutionary scenario.
- Use of scoring scheme that reflects these probabilities
- Direct observation of actual substitution rates
- PAM matrices: the acronym PAM stands for *Point Accepted Mutations,* or *Percent of Accepted Mutations*

PAM matrices

The substitution score is expected to depend on the rate of divergence between sequences.



The **PAM matrices** derived by Dayhoff (1978):

- are based on evolutionary distances.
- have been obtained from carefully aligned closely related protein sequences (71 gapless alignments of sequences having at least 85% similarity).

M. Dayhoff

Dayhoff *et al.* (1978). A model of evolutionary change in proteins. In *Atlas of Protein Sequence and Structure*, vol. 5, suppl. 3, 345–352. National Biomedical Research Foundation, Silver Spring, MD, 1978.

PAM matrices

PAM = Percent (or Point) Accepted Mutation

The PAM matrices are **series of scoring matrices**, each reflecting a certain level of divergence:

PAM = unit of evolution (1 PAM = 1 mutation/100 amino acid)

- PAM1 proteins with an evolutionary distance of 1% mutation/position
- PAM50 idem for 50% mutations/position
- PAM250 250% mutations/position (a position could mutate several times)

Dayhoff *et al.* (1978). A model of evolutionary change in proteins. In *Atlas of Protein Sequence and Structure*, vol. 5, suppl. 3, 345–352. National Biomedical Research Foundation, Silver Spring, MD, 1978.

PAM matrices

- Choose an evolutionary distance at which to compare the sequences.
- These matrices are functions of this distance.
  - 250-PAM matrix: comparing sequences that are 250 units of evolution apart.
- For each evolutionary distance we have
  - a *probability transition matrix M* and
  - a *scores matrix S.*
- The scores matrix is obtained from the probability matrix

PAM matrices

- Ingredients for building the 1-PAM matrix *M*:
  - A list of *accepted mutations*
  - The *probabilities of occurrence* p_a for each amino acid *a*
- A*ccepted mutation:* mutation that occurred and was positively selected by the environment; that is, it did not cause the demise of the particular organism where it occurred
- Align two homologous proteins from different species
- Each position where the sequences differ will give us an accepted mutation.
- We consider these accepted mutations as undirected events

PAM matrices

- 1-PAM matrix: consider immediate mutations, *a* -> *b,* not mediated ones like *a* —> c —> *b.*
- The probabilities of occurrence can be estimated by computing the relative frequency of occurrence of amino acids over a large protein sequence set.

$$\sum_a p_a = 1$$

Example

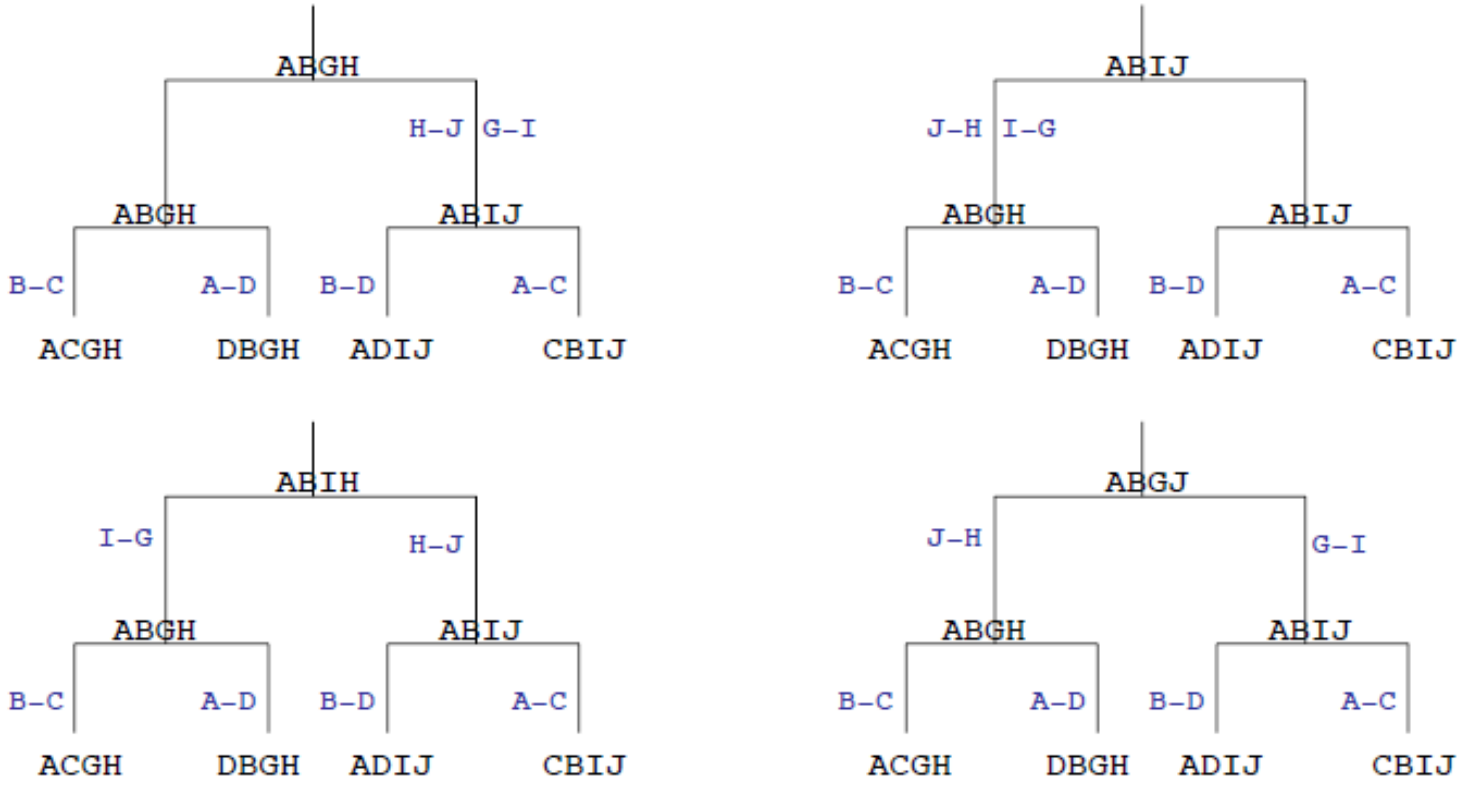
To illustrate how the PAM substitution matrices have been derived, we will consider the following artificial ungapped aligned sequences:

```
A C G H
D B G H
A D I J
C B I J
```

Example taken from Borodovsky & Ekisheva (2007) Problems and Solutions in Biological sequence analysis. *Cambridge Univ Press*.

Derivation of PAM matrices

Phylogenetic trees (maximum parsimony)



Here are represented the four more parsimonious (minimum of substitutions) phylogenetic trees for the alignment given above.

Derivation of PAM matrices

Matrix of accepted point mutation counts (A)

	A	B	C	D	G	H	I	J
A		0	4	4	0	0	0	0
B	0		4	4	0	0	0	0
C	4	4		0	0	0	0	0
D	4	4	0		0	0	0	0
G	0	0	0	0		0	4	0
H	0	0	0	0	0		0	4
I	0	0	0	0	4	0		0
J	0	0	0	0	0	4	0	

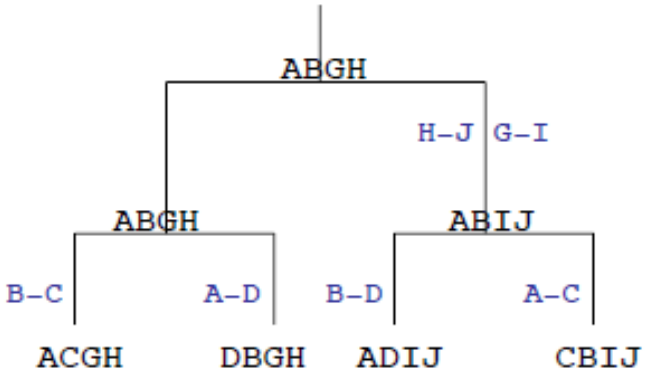
For each pair of different amino acids (i,j), the total number a_{ij} of substitutions from i to j along the edges of the phylogenetic tree is calculated.

(they are indicated in blue on the previous slide)

Derivation of PAM matrices

Each edge of a given tree is associated with the ungapped alignment of the two sequences connected by this edge.

Thus, any tree shown above generates 6 alignments. For example the first phylogenetic tree generates the following alignments:



A B G H	A B G H	A B G H
A B G H	A B I J	A C G H
A B G H	A B I J	A B I J
D B G H	A D I J	C B I J

Those alignments can be used to assess the "relative mutability" of each amino acid.

PAM matrices

- f_{ab}: number of times the mutation *a* <-> *b* occurs
- Undirected mutations → $f_{ab} = f_{ba}$
- Total number of mutations in which *a* was involved: $f_a = \sum_{b \neq a} f_{ab}$

- Total number of amino acid occurrences involved in mutations: $f = \sum_a f_a$
  - f is twice the total number of mutations

- 1-PAM transition probability matrix *M:* 20 x 20 matrix with M_{ab} being the probability of amino acid *a* changing into amino acid *b.*

PAM matrices

- *a* and *b* may be the same, in which case we have the probability of *a* remaining unchanged during this particular evolutionary interval.
- Computation of M_{aa} is done based on the ***relative mutability*** of amino acid *a,* defined as

$$m_a = \frac{f_a}{100 f p_a}$$

- Mutability of an amino acid: probability that the given amino acid will change in the evolutionary period of interest.
- The probability of *a* remaining unchanged is the complementary probability

$$M_{aa} = 1 - m_a.$$

Relative mutability

The relative mutability is defined by the ratio of the total number of times that amino acid *j* has changed in all the pair-wise alignments (in our case 6x4=24 alignments) to the number of times that *j* has occurred in these alignments, i.e.

$$m_j = \frac{number\ of\ changes\ of\ j}{number\ of\ occurrences\ of\ j}$$

Relative amino acid mutability values m_j *for our example*

Amino acid	A	B	I	H	G	J	C	D
Changes (substitutions)	8	8	4	4	4	4	8	8
Frequency of occurrence	40	40	24	24	24	24	8	8
Relative mutability m_j	0.2	0.2	0.167	0.167	0.167	0.167	1	1

The relative mutability accounts for the fact that the different amino acids have different mutation rates. This is thus the probability to mutate.

PAM matrices

- Probability of *a* changing into *b:* computed as the product of the conditional probability that *a* will change into *b,* given that *a* changed, times the probability of *a* changing.
- We estimate the conditional probability as the ratio between the *a* <-> *b* mutations and the total number of mutations involving *a.*

$$
\begin{aligned}
M_{ab} &= \Pr(a \rightarrow b) \\
&= \Pr(a \rightarrow b \mid a \text{ changed}) \Pr(a \text{ changed}) \\
&= \frac{f_{ab}}{f_a} m_a.
\end{aligned}
$$

- Use of a simplified model of protein evolution.

PAM matrices

Mutational probability matrix derived by Dayhoff for the 20 amino acids

	A	R	N	D	C	Q	E	G	H	I	L	K	M	F	P	S	T	W	Y	V
A	9867	2	9	10	3	8	17	21	2	6	4	2	6	2	22	35	32	0	2	18
R	1	9913	1	0	1	10	0	0	10	3	1	19	4	1	4	6	1	8	0	1
N	4	1	9822	36	0	4	6	6	21	3	1	13	0	1	2	20	9	1	4	1
D	6	0	42	9859	0	6	53	6	4	1	0	3	0	0	1	5	3	0	0	1
C	1	1	0	0	9973	0	0	0	1	1	0	0	0	0	1	5	1	0	3	2
Q	3	9	4	5	0	9876	27	1	23	1	3	6	4	0	6	2	2	0	0	1
E	10	0	7	56	0	35	9865	4	2	3	1	4	1	0	3	4	2	0	1	2
G	21	1	12	11	1	3	7	9935	1	0	1	2	1	1	3	21	3	0	0	5
H	1	8	18	3	1	20	1	0	9912	0	1	1	0	2	3	1	1	1	4	1
I	2	2	3	1	2	1	2	0	0	9872	9	2	12	7	0	1	7	0	1	33
L	3	1	3	0	0	6	1	1	4	22	9947	2	45	13	3	1	3	4	2	15
K	2	37	25	6	0	12	7	2	2	4	1	9926	20	0	3	8	11	0	1	1
M	1	1	0	0	0	2	0	0	0	5	8	4	9874	1	0	1	2	0	0	4
F	1	1	1	0	0	0	0	1	2	8	6	0	4	9946	0	2	1	3	28	0
P	13	5	2	1	1	8	3	2	5	1	2	2	1	1	9926	12	4	0	0	2
S	28	11	34	7	11	4	6	16	2	2	1	7	4	3	17	9840	38	5	2	2
T	22	2	13	4	1	3	2	2	1	11	2	8	6	1	5	32	9871	0	2	9
W	0	2	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	9976	1	0
Y	1	0	3	0	3	0	1	0	4	1	1	0	0	21	0	1	1	2	9945	1
V	13	2	1	1	3	2	2	3	3	57	11	1	17	1	3	2	10	0	2	9901

For clarity, the values have been multiplied by 10000

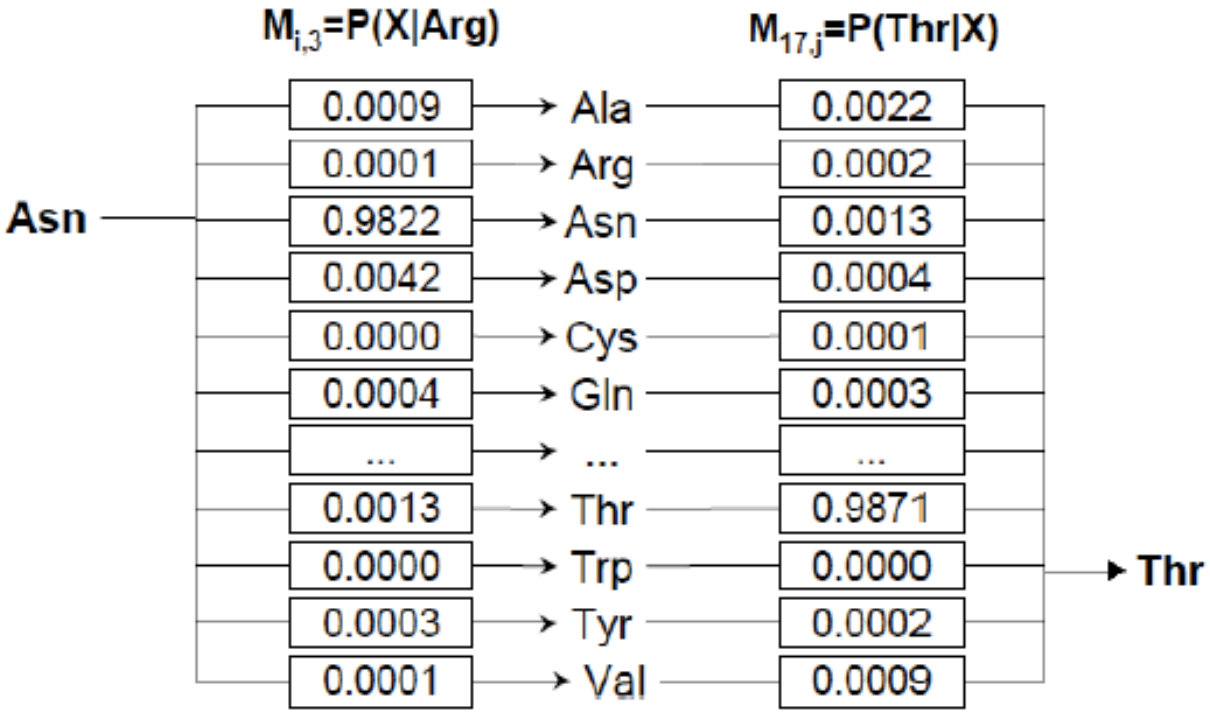
This matrix corresponds to an evolution time period giving 1 mutation/100 amino acids, and is refered to as the **PAM1 matrix.**

Source: Dayhoff, 1978

PAM matrices

- What is the probability that *a* will change into *b* in two PAM units of evolution?
  - In the first unit period *a* changes into any amino acid c with probability M_{ac}
  - Then *c* changes into *b* in the second period with probability M_{cb}
  - The final figure is nothing more than M^2_{ab} that is, an entry in the square of M.
- M^k*:* transition probability matrix for a period of *k* units of evolution.

From PAM1 to PAM2



P(Asn -> Thr)= P(Asn -> Ala -> Thr) + P(Asn -> Arg -> Thr) + ... + P(Asn -> Val -> Thr)
= (0.0009)(0.0001) + (0.0001)(0.0002) + ... + (0.0001)(0.009)

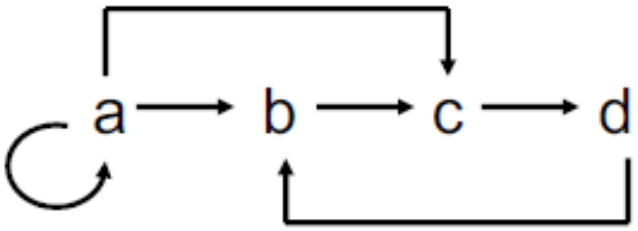
line 3 of PAM1 column 17 of PAM1

=> Matrix product: PAM2 = PAM1 x PAM1

Source: J. van Helden

From PAM1 to PAM2, PAM100, PAM250, …

Remark (from graph theory)



	a	b	c	d
a	1	1	1	0
b	0	0	1	0
c	0	0	0	1
d	0	1	0	0

Matrix **Q** indicates the number of paths going from one node to another in 1 step

	a	b	c	d
a	1	1	2	1
b	0	0	0	1
c	0	1	0	1
d	0	1	1	1

Matrix $\mathbf{Q}^2$ indicates the number of paths going from one node to another in 2 steps

	a	b	c	d
a	…	…	…	…
b	…	…	…	…
c	…	…	…	…
d	…	…	…	…

Matrix $\mathbf{Q}^n$ indicates the number of paths going from one node to another in *n* steps

Source: J. van Helden

PAM matrices

- Scoring matrices: the entries are related to the ratio between two probabilities, namely, the probability that a pair is a mutation as opposed to being a random occurrence. This is called a *likelihood* or *odds* ratio.
- Let us then compute this ratio for two amino acids *a* and *b.* Suppose that we paired *a* with *b* in a given alignment. Taking the point of view of *a,* the probability that *b* is there in the other sequence due to a mutation is M_{ab}.
- There is a chance of P_b for a random occurrence of *b.* The ratio is then

$$\frac{M_{ab}}{P_b}$$

PAM matrices

- Scoring matrix for *k* PAM distance:

$$score_k(a, b) = 10 \log_{10} \frac{M_{ab}^k}{p_b}$$

- Sometimes we have two sequences and no information on their evolutionary distance. Recommended approach: compare the sequences using two or three matrices that cover a wide range, for instance, 40 PAM, 120 PAM, and 250 PAM.
- In general, low PAM numbers are good for finding short, strong local similarities, while high PAM numbers detect long, weak ones.

PAM matrices

PAM250 derived by Dayhoff for the 20 amino acids

	A	R	N	D	C	Q	E	G	H	I	L	K	M	F	P	S	T	W	Y	V
A	13	6	9	9	5	8	9	12	6	8	6	7	7	4	11	11	11	2	4	9
R	3	17	4	3	2	5	3	2	6	3	2	9	4	1	4	4	3	7	2	2
N	4	4	6	7	2	5	6	4	6	3	2	5	3	2	4	5	4	2	3	3
D	5	4	8	11	1	7	10	5	6	3	2	5	3	1	4	5	5	1	2	3
C	2	1	1	1	52	1	1	2	2	2	1	1	1	1	2	3	2	1	4	2
Q	3	5	5	6	1	10	7	3	7	2	3	5	3	1	4	3	3	1	2	3
E	5	4	7	11	1	9	12	5	6	3	2	5	3	1	4	5	5	1	2	3
G	12	5	10	10	4	7	9	27	5	5	4	6	5	3	8	11	9	2	3	7
H	2	5	5	4	2	7	4	2	15	2	2	3	2	2	3	3	2	2	3	2
I	3	2	2	2	2	2	2	2	2	10	6	2	6	5	2	3	4	1	3	9
L	6	4	4	3	2	6	4	3	5	15	34	4	20	13	5	4	6	6	7	13
K	6	18	10	8	2	10	8	5	8	5	4	24	9	2	6	8	8	4	3	5
M	1	1	1	1	0	1	1	1	1	2	3	2	6	2	1	1	1	1	1	2
F	2	1	2	1	1	1	1	1	3	5	6	1	4	32	1	2	2	4	20	3
P	7	5	5	4	3	5	4	5	5	3	3	4	3	2	20	6	5	1	2	4
S	9	6	8	7	7	6	7	9	6	5	4	7	5	3	9	10	9	4	4	6
T	8	5	6	6	4	5	5	6	4	6	4	6	5	3	6	8	11	2	3	6
W	0	2	0	0	0	0	0	0	1	0	1	0	0	1	0	1	0	55	1	0
Y	1	1	2	1	3	1	1	1	3	2	2	1	2	15	1	2	2	3	31	2
V	7	4	4	4	4	4	4	4	5	4	15	10	4	10	5	5	5	72	4	17

For clarity, the values have been multiplied by 100

This matrix corresponds to an evolution time period giving 250 mutation/100 amino acids (i.e. an evolutionary distance of 250 PAM), and is refered to as the **PAM250 matrix.**

Source: Dayhoff, 1978

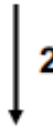
PAM matrices

Interpretation of the PAM250 matrix

	A	R	N	D	...
A	13	6	9	9	...
R	3	17	4	3	...
N	4	4	6	7	...
D	5	4	8	11	...
C	2	1	1	1	...
Q	3	5	5	6	...
E	5	4	7	11	...
G	12	5	10	10	...
H	2	5	5	4	...
I	3	2	2	2	...
L	6	4	4	3	...
K	6	18	10	8	...
M	1	1	1	1	...
F	2	1	2	1	...
P	7	5	5	4	...
S	9	6	8	7	...
T	8	5	6	6	...
W	0	2	0	0	...
Y	1	1	2	1	...
V	7	4	4	4	...

In comparing 2 sequences at this evolutionary distance (250 PAM), there is:

* * * * **A** * * * * *



* * * * **A** * * * * *	**probability of 13%**
* * * * **R** * * * * *	**probability of 3%**
* * * * **N** * * * * *	**probability of 4%**
* * * * **W** * * * * *	**probability of 0%**

...

Source: Dayhoff, 1978

Local alignment

- Local alignment seeks similar segments of unspecified length from the 2 sequences being compared.
- Rigorous method: local dynamic programming, time is proportional to the product of lengths of sequences it compares.
- BLAST: linear time heuristic algorithm.
- Basic Local Alignment Search Tool – a family of most popular sequence search program including: Basic BLAST, Gapped BLAST, Psi - BLAST
- Main idea (basic BLAST): Homologous sequences are likely to contain a short high scoring similarity region a hit
- Each hit gives a seed that BLAST tries to extend on both sides

BLAST

- BLAST programs among the most frequently used to search sequence databases worldwide.
- BLAST: Basic Local Alignment Search Tool.
- D*atabase:* collection of sequences
- BLAST returns a list of *high-scoring segment pairs* between the query sequence and sequences in the database.
- A *segment* is a substring of a sequence.
- Given two sequences, a *segment pair* between them is a pair of segments of the same length, one from each sequence

BLAST

- Because the substrings in a segment pair have the same length, we can form a gapless alignment with them.
- This alignment can be scored using a matrix of substitution scores.
- No gap-penalty functions are needed, as there are no gaps.
- The score thus obtained is by definition the score of the segment pair.

Example

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

K	A	L	M	R
V	A	K	N	S
−4	3	−4	−3	−1

→ Total: −9

BLAST

- Given a query sequence, BLAST returns all segment pairs between the query and a database sequence with scores above a certain threshold *S.*
- The parameter *S* can be set by the user, although a default value is provided in most servers that run the program.
- M*aximum segment pair* (MSP) between two sequences: segment pair of maximum score.
  - Measure of sequence similarity and can be computed precisely by dynamic programming.
  - BLAST estimates this number much faster than any dynamic programming method.
  - *locally* maximal segment pairs: those that cannot be improved further by extending or shortening them.

BLAST

- Finds certain "seeds," which are very short segment pairs between the query and a database sequence.
- Seeds extended in both directions, without including gaps, until the maximum possible score for extensions of this particular seed is reached.
- Not all extensions are looked at. The program has a criterion to stop extensions when the score falls below a carefully computed limit.

BLAST

We may think of BLAST as a three-step algorithmic procedure, undertaking the following tasks.

1. Compile list of high-scoring strings (or *words,* in BLAST jargon).
2. Search for hits — each hit gives a seed.
3. Extend seeds.

Step 1: find high scoring words

- For every word x of length w in Q make a list of words that when aligned to x score at least T.
- Example: Let x=AIV then score for AIA is 5+5+0 (dropped) and for AII 5+5+4 (taken)
- Number of words in the list depends on w and T, and is much less than 20^3 (typically about 50)

Step 1

MVRERKCILCHIVYGSKKEMDEHMRSMLHHRELENLKGRDIS

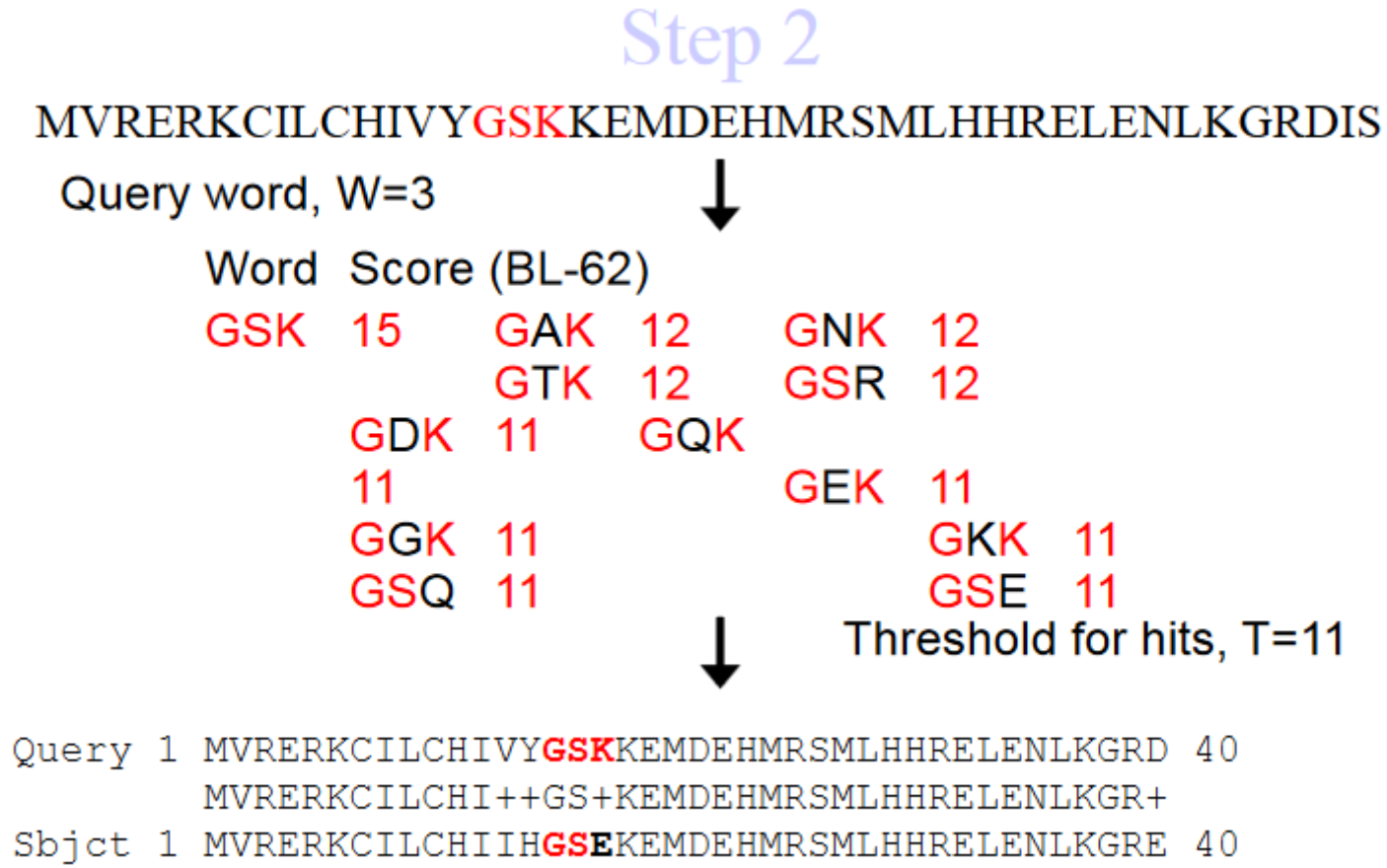
Query word, W=3 for proteins

(W=11 for nucleotides)

Word	Score (BL-62)
GSK	15
GAK	12
GNK	12
GTK	12
GSR	12
GDK	11
GQK	11
GEK	11
GGK	11
GKK	11
GSQ	11
GSE	11

Step 2 – Finding hits

Scan database for exact matching with the list of words compiled in step1 using techniques as hash table (requires preprocessing of a database)



Step 3: Extending hits

- Parameter: X (controlled by a user)
- Extend the hits in both ways along diagonal (ungapped alignment) until score drops more than X relative to the best score yet attained.
- Return the score highest scoring segment pair (HSP).