Basic **L**ocal **A**lignment **S**earch **T**ool

Stephen Altschul, Warren Gish, Webb Miller, Eugene Myers, David Lipman

Presented By: Hakim Mohd Azhan and Zoe Hansen

Introduction

Discover sequence homology to a known protein

Measure of similarity between sequences to **distinguish significant relationship**

Variations of dynamic programming algorithm

- **Needleman & Wunsch**
- Assign scores to insertions, deletions and replacement
- Compute to find the least with mutations
- Minimizing the evolutionary distance / maximizing the similarities

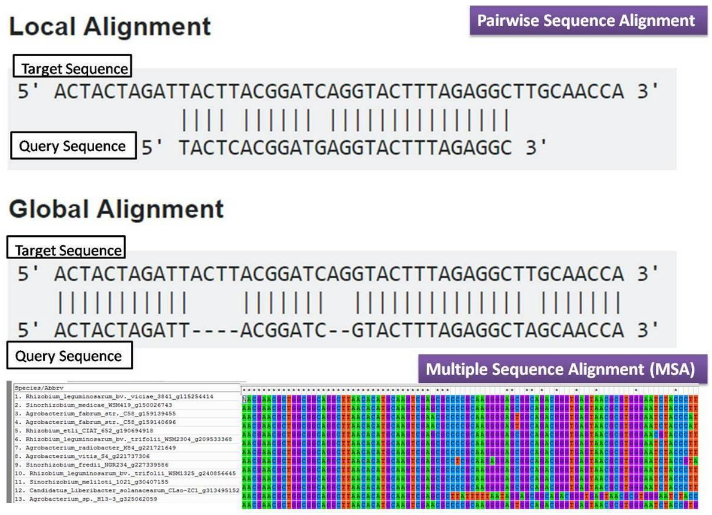
Impractical for **large databases**

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

Sensor Selection based on Minimum Redundancy Maximum Relevance for Activity Recognition in Smart Homes: 5th ICCST 2018, Kota Kinabalu, Malaysia, 29-30 August 2018

Introduction

- Rapid heuristic algorithm has been developed
  - **Allow large databases** to be searched
  - Implicit in the **algorithm itself**
- FASTP program
  - Find **similarities** based on **identities**
  - Using the **PAM matrix**
  - Allow **conservative replacements** and **identities** to increment the similarity score
  - **Indirect approximation** of minimal evolution measures
- BLAST
  - Basic Local Alignment Search Tool
  - Employs measure based on **well-defined mutation scores**
  - **Direct approximation** of results
  - **Faster** than existing heuristic algorithms
  - Detect weak but biologically **significant sequence similarities**


Local Alignment
Pairwise Sequence Alignment
Target Sequence
5' ACTACTAGATTACTTACGGATCAGGTACTTTAGAGGCTTGCAACCA 3'
Query Sequence
5' TACTCACGGATGAGGTACTTTAGAGGC 3'
Global Alignment
Target Sequence
5' ACTACTAGATTACTTACGGATCAGGTACTTTAGAGGCTTGCAACCA 3'
5' ACTACTAGATT----ACGGATC--GTACTTTAGAGGCTAGCAACCA 3'
Query Sequence
Multiple Sequence Alignment (MSA)
Species/Abbrv
1. Rhizobium_leguminosarum_bv._viciae_3841_g115254414
2. Sinorhizobium_medicae_WSM419_g150026743
3. Agrobacterium_fabrum_str._C58_g159139455
4. Agrobacterium_fabrum_str._C58_g159140696
5. Rhizobium_etli_CIAT_652_g190694918
6. Rhizobium_leguminosarum_bv._trifolii_WSM2304_g209533368
7. Agrobacterium_radiobacter_K84_g221721649
8. Agrobacterium_vitis_S4_g221737306
9. Sinorhizobium_fredii_NGR234_g227339586
10. Rhizobium_leguminosarum_bv._trifolii_WSM1325_g240856645
11. Sinorhizobium_meliloti_1021_g30407155
12. Candidatus_Liberibacter_solanacearum_CLso-ZC1_g313495152
13. Agrobacterium_sp._H13-3_g325062059

https://www.majordifferences.com/2016/05/difference-between-pairwise-and-multiple-sequence-alignment.html#.XEpuZ1xKjb0

Glossary

Substitution matrices : collection of scores for aligning nucleotides or amino acid with one another (PAM 120, BLOSUM 62)

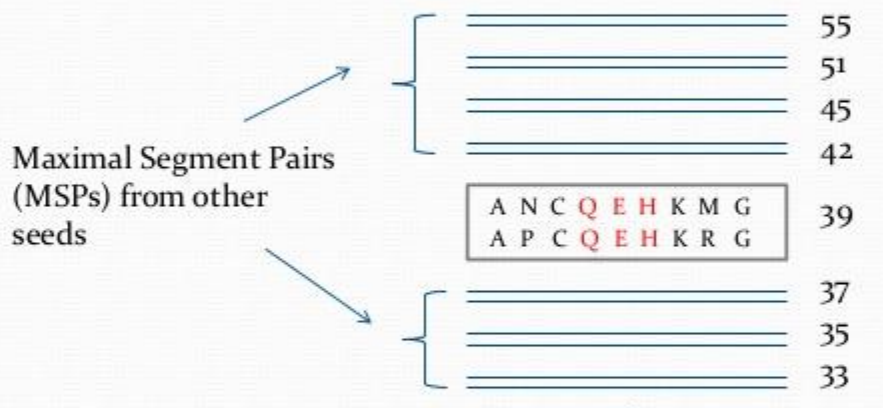
Maximal Segment Pair (MSP) : Highest scoring pair of identical length segments chosen from 2 sequences

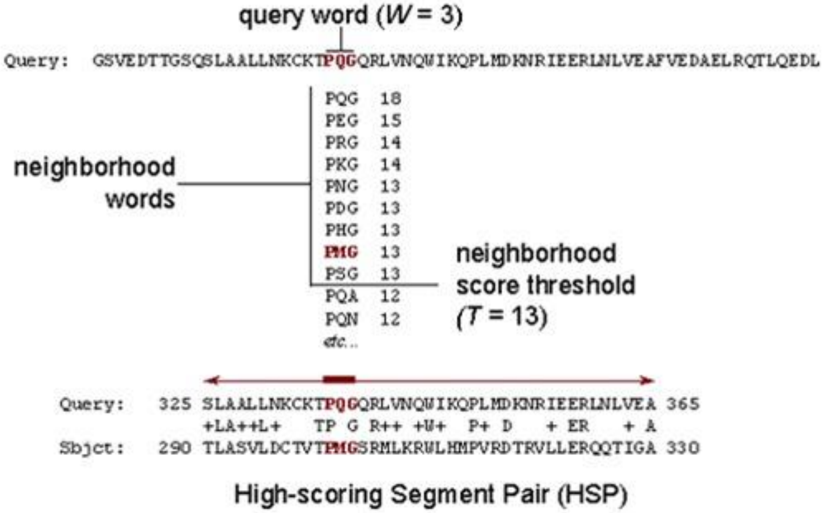
High-Scoring Pair (HSP) : A local alignment with no gaps that achieves one of the highest alignment score in a given search

Global Alignment : Optimize the overall alignment of two sequences which may includes large stretches of low similarity

Local Alignment : Relatively conserved subsequences and a single comparison

Altschul, et al.*(*1990*)* **215**:403-10.
https://www.biostars.org/p/210490/

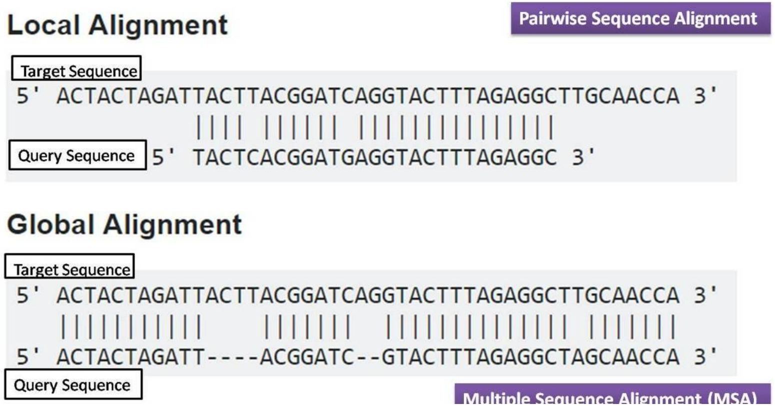



query word (*W* = 3)
neighborhood words
neighborhood score threshold (*T* = 13)
High-scoring Segment Pair (HSP)

```
Query:  GSVEDTTGSQSLAALLNKCKTPQGQRLVNQWIKQPLMDKNRIEERLNLVEAFVEDAELRQTLQEDL

PQG  18
PEG  15
PRG  14
PKG  14
PNG  13
PDG  13
PHG  13
PMG  13
PSG  13
PQA  12
PQN  12
etc...

Query:  325 SLAALLNKCKTPQGQRLVNQWIKQPLMDKNRIEERLNLVEA 365
            +LA++L+   TP G R++ +W+  P+ D   + ER   + A
Sbjct:  290 TLASVLDCTVTPMGSRMLKRWLHMPVRDTRVLLERQQTIGA 330
```


Pairwise Sequence Alignment
Multiple Sequence Alignment (MSA)

Local Alignment

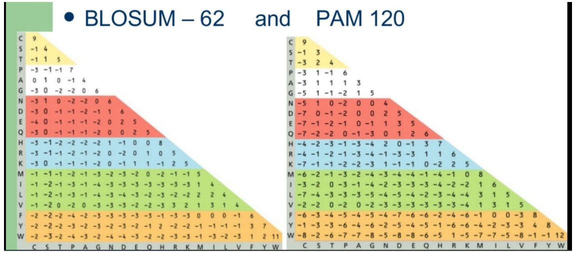
```
Target Sequence
5' ACTACTAGATTACTTACGGATCAGGTACTTTAGAGGCTTGCAACCA 3'
             |||| |||||| |||||||||||||||
Query Sequence
          5' TACTCACGGATGAGGTACTTTAGAGGC 3'
```

Global Alignment

```
Target Sequence
5' ACTACTAGATTACTTACGGATCAGGTACTTTAGAGGCTTGCAACCA 3'
   |||||||||||    |||||||  |||||||||||||| |||||||
5' ACTACTAGATT----ACGGATC--GTACTTTAGAGGCTAGCAACCA 3'
Query Sequence
```

https://www.ncbi.nlm.nih.gov/books/NBK62051/
https://www.slideshare.net/jumbooctobre/blast-2013-1, slide 17

Methods - Maximal Segment Pair Measure



- BLOSUM – 62 and PAM 120

	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

	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

First, begin with a ***matrix of similarity scores*** for all possible pairs of residues

- Identities/conservative replacements = **+**
- Unlikely replacements = **-**

For amino acid comparisons, use a **PAM120 matrix** or a **BLOSUM matrix**, depending on your intended comparisons

For DNA comparisons, assign a **+5** value to identities, and a **-4** value to mismatches (can use other point values)

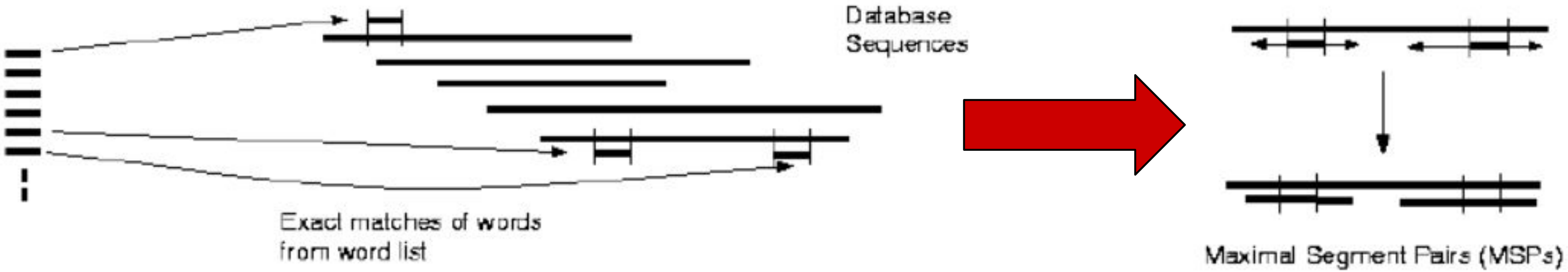
Sequence Segment = “contiguous stretch of residues of any length”

Similarity Score = “the sum of the similarity values for each pair of aligned residues”

Altschul et al. (1990), pg. 404

Methods - Maximal Segment Pair Measure

Maximal Segment Pair (MSP) = "highest scoring pair of identical length segments chosen from two sequences"


Database Sequences
Exact matches of words from word list
Maximal Segment Pairs (MSPs)

A segment pair is **locally maximal** if its score is not improved by extension or shortening

Importantly, BLAST has **mathematical tractability**, and we can estimate frequencies of paired residues in our maximal segments

Altschul et al. (1990), pg. 404
http://bioinfo3d.cs.tau.ac.il/Education/CS99b/class_notes/class3.html

Methods - Approximation of MSP Scores

Only a portion of sequences will be homologous to the query sequence we use.

S = a cutoff score established for examining MSPs

Sequences that meet our score, ***S***, can:

- Share significant similarity with our query sequence
- Be a set of high-scoring random sequences
- Be distantly related to our query sequence

BLAST focuses on sequences whose similarity with the query sequence is more likely to exceed our specified score, ***S***, than those whose scores will not.

- Seek out only segment pairs that contain a word pair with a score of at least ***T***
- ***The lower our threshold, T, the more likely we are to find a segment pair with a score of at least S that contains a word pair of at least T.***

Altschul et al. (1990), pg. 404

Methods - Implementation

Three algorithmic steps:

1) Compiling a list of high-scoring words
2) Scanning a database for hits
3) Extending those hits

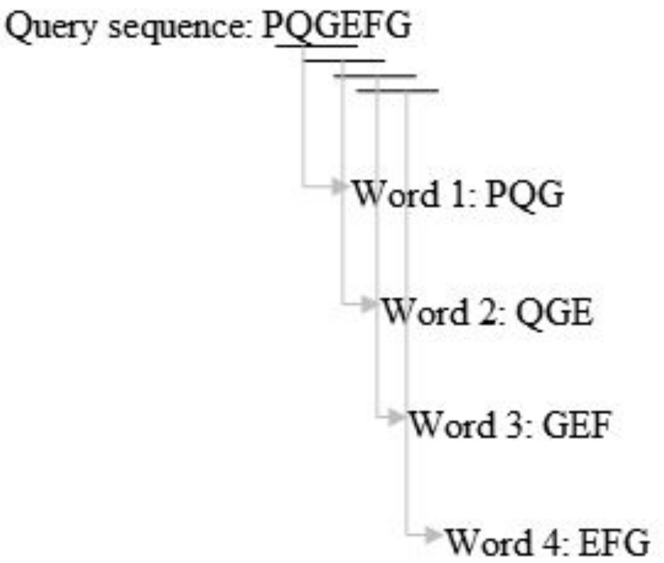
Implementation of these steps depends on whether we are working with **DNA** or **protein** sequences

Methods - Implementation

Three algorithmic steps:

1) **Compiling a list of high-scoring words**
2) Scanning a database for hits
3) Extending those hits

For proteins, the list consists of all words (w-mers) that score least *T* when compared to a word in the query sequence.



Adapted from Biological Sequence Analysis I, Current Topics in Genome Analysis, Wikipedia

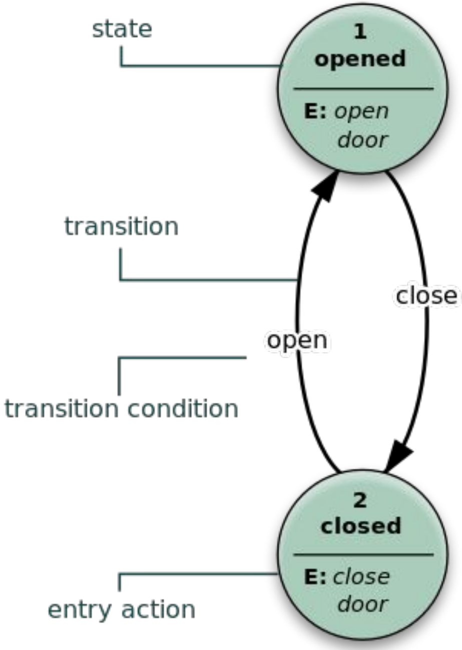
Methods - Implementation

Three algorithmic steps:

1) Compiling a list of high-scoring words
2) **Scanning a database for hits**
3) Extending those hits

Two separate approaches were investigated:

- Map each word to an integer between 1 and 20^w, and create an array
  - The *i*th entry of such an array points to a list of all occurrences of the *i*th word
- Use a **"deterministic finite automaton"**
  - *Mealy paradigm*, rather than a *Moore paradigm*
  - Output considers both the current state and the current inputs, rather than solely the current state



Mealy, George H. (1955). *A Method for Synthesizing Sequential Circuits*. Bell System Technical Journal. pp. 1045–1079. via Wikipedia
https://en.wikipedia.org/wiki/State_diagram

Methods - Implementation

Three algorithmic steps:

1) Compiling a list of high-scoring words
2) Scanning a database for hits
3) **Extending those hits**

Extend a current hit to find a local MSP

- Stop extending in one direction when the score of a segment pair falls below the best score found for a shorter extension

For DNA, the database is compressed to increase efficiency of the scanning and extension steps

Methods have been developed to deal with the non-randomness of DNA sequences

- AT-rich regions
- Highly repetitive elements

Results - Performance of BLAST with Random Sequences

We can evaluate the statistical significance of MSP scores using two parameters:

- ***K*** : set of probabilities of the occurrence of individual residues
- ***λ*** : set of scores for aligning pairs of residues

For two random sequences of lengths *m* and *n*, the probability of finding a segment pair with a score greater than or equal to *S* is:

$$1 - e^{-y} \quad \text{where} \quad y = Kmn\, e^{-\lambda S}$$

We can use this formula to approximate the score that an MSP must have to be distinguishable from chance similarities within the database

Altschul et al. (1990), pg. 405-406

Results - Performance of BLAST with Random Sequences

Central idea of BLAST:

“confine attention to segment pairs that contain a word pair of length *w* with a score of at least *T*”

Therefore, we are interested in the proportion of segment pairs which contain a word pair and are of a given score.

q = the probability that a segment pair will fail to contain a word pair with a score of at least ***T***

“The longer an MSP, the more independent chances it effectively has for containing a word with a score of at least **T**”

- **q** should decrease exponentially with increasing MSP score, ***S***

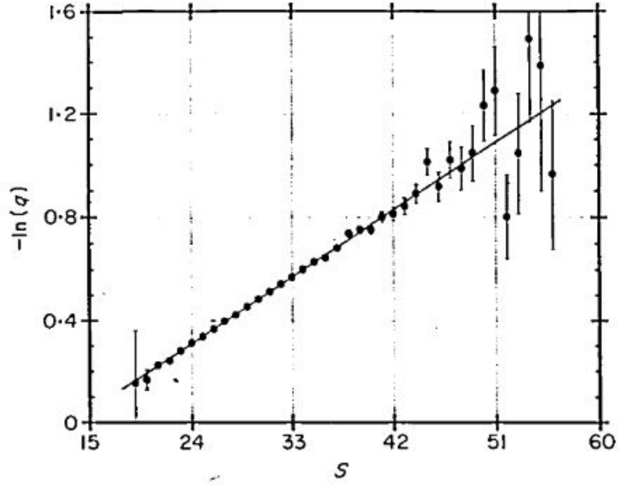


Figure 1. The probability q of BLAST missing a random maximal segment pair as a function of its score S.

Altschul et al. (1990), pg. 406

Results - Word Length and Threshold Parameters

The two parameters that we must set prior to executing BLAST are ***w*** and ***T***

- How do we choose these values?
- We must consider the **time** requirement associated with the three steps:
  1) Compiling the word list
  2) Scanning the database for hits
  3) Extending our hits to look for scores that exceed our cutoff

The time required for Step (3) is proportional to the number of hits we obtain from our search, which is directly dependent on our ***w*** and ***T*** settings.

Results - Word Length and Threshold Parameters

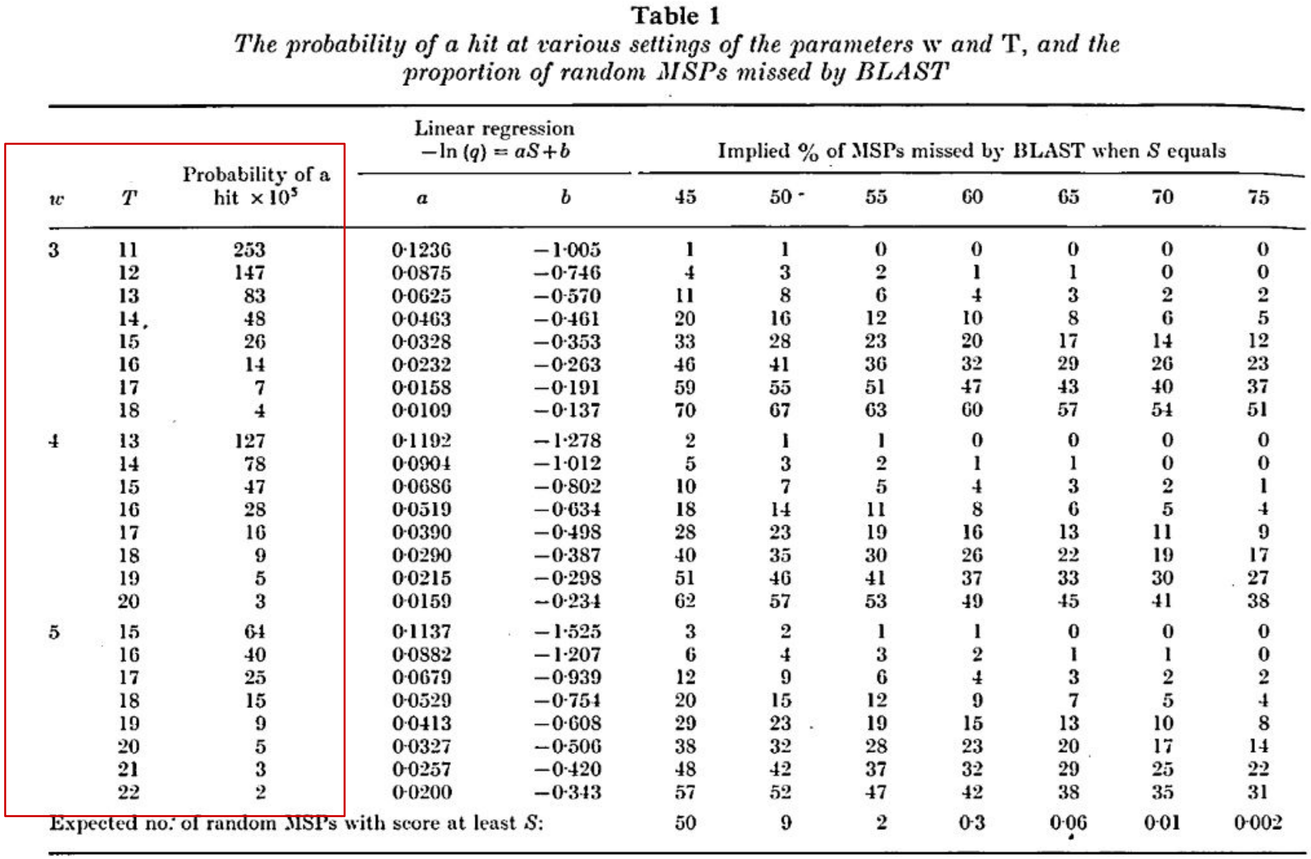


Table 1

The probability of a hit at various settings of the parameters w and T, and the proportion of random MSPs missed by BLAST

			Linear regression $-\ln(q) = aS+b$		Implied % of MSPs missed by BLAST when S equals						
w	T	Probability of a hit $\times 10^5$	a	b	45	50	55	60	65	70	75
3	11	253	0·1236	−1·005	1	1	0	0	0	0	0
	12	147	0·0875	−0·746	4	3	2	1	1	0	0
	13	83	0·0625	−0·570	11	8	6	4	3	2	2
	14	48	0·0463	−0·461	20	16	12	10	8	6	5
	15	26	0·0328	−0·353	33	28	23	20	17	14	12
	16	14	0·0232	−0·263	46	41	36	32	29	26	23
	17	7	0·0158	−0·191	59	55	51	47	43	40	37
	18	4	0·0109	−0·137	70	67	63	60	57	54	51
4	13	127	0·1192	−1·278	2	1	1	0	0	0	0
	14	78	0·0904	−1·012	5	3	2	1	1	0	0
	15	47	0·0686	−0·802	10	7	5	4	3	2	1
	16	28	0·0519	−0·634	18	14	11	8	6	5	4
	17	16	0·0390	−0·498	28	23	19	16	13	11	9
	18	9	0·0290	−0·387	40	35	30	26	22	19	17
	19	5	0·0215	−0·298	51	46	41	37	33	30	27
	20	3	0·0159	−0·234	62	57	53	49	45	41	38
5	15	64	0·1137	−1·525	3	2	1	1	0	0	0
	16	40	0·0882	−1·207	6	4	3	2	1	1	0
	17	25	0·0679	−0·939	12	9	6	4	3	2	2
	18	15	0·0529	−0·754	20	15	12	9	7	5	4
	19	9	0·0413	−0·608	29	23	19	15	13	10	8
	20	5	0·0327	−0·506	38	32	28	23	20	17	14
	21	3	0·0257	−0·420	48	42	37	32	29	25	22
	22	2	0·0200	−0·343	57	52	47	42	38	35	31
Expected no. of random MSPs with score at least S:					50	9	2	0·3	0·06	0·01	0·002

Results - Word Length and Threshold Parameters

To identify the optimal ***T*** setting, the authors investigated the execution time vs. the number of words generated for each value of ***T***.

- The number of words generated increases exponentially with decreasing ***T***

$$aW + bN + cNW/20^w$$

- *W* = number of words generated
- *N* = number of residues in the database
- *a, b, c* = constants

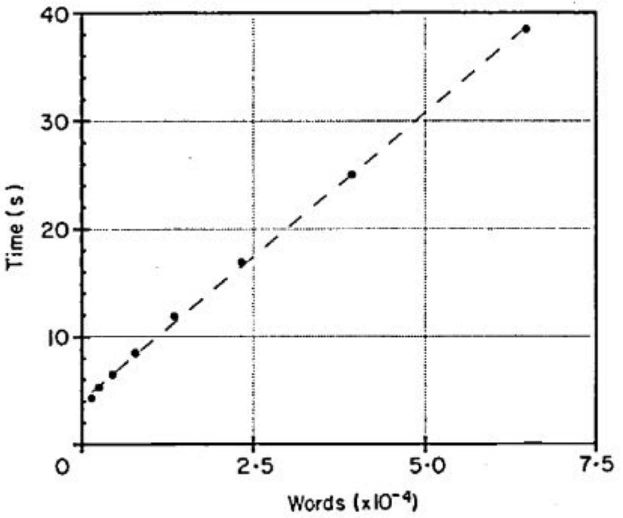


Figure 2. The central processing unit time required to execute BLAST on the PIR protein database (Release 23·0) as a function of the size of the word list generated. Points correspond to values of the threshold parameter *T* ranging from 13 to 20. Greater values of *T* imply fewer words in the list.

Results - Word Length and Threshold Parameters

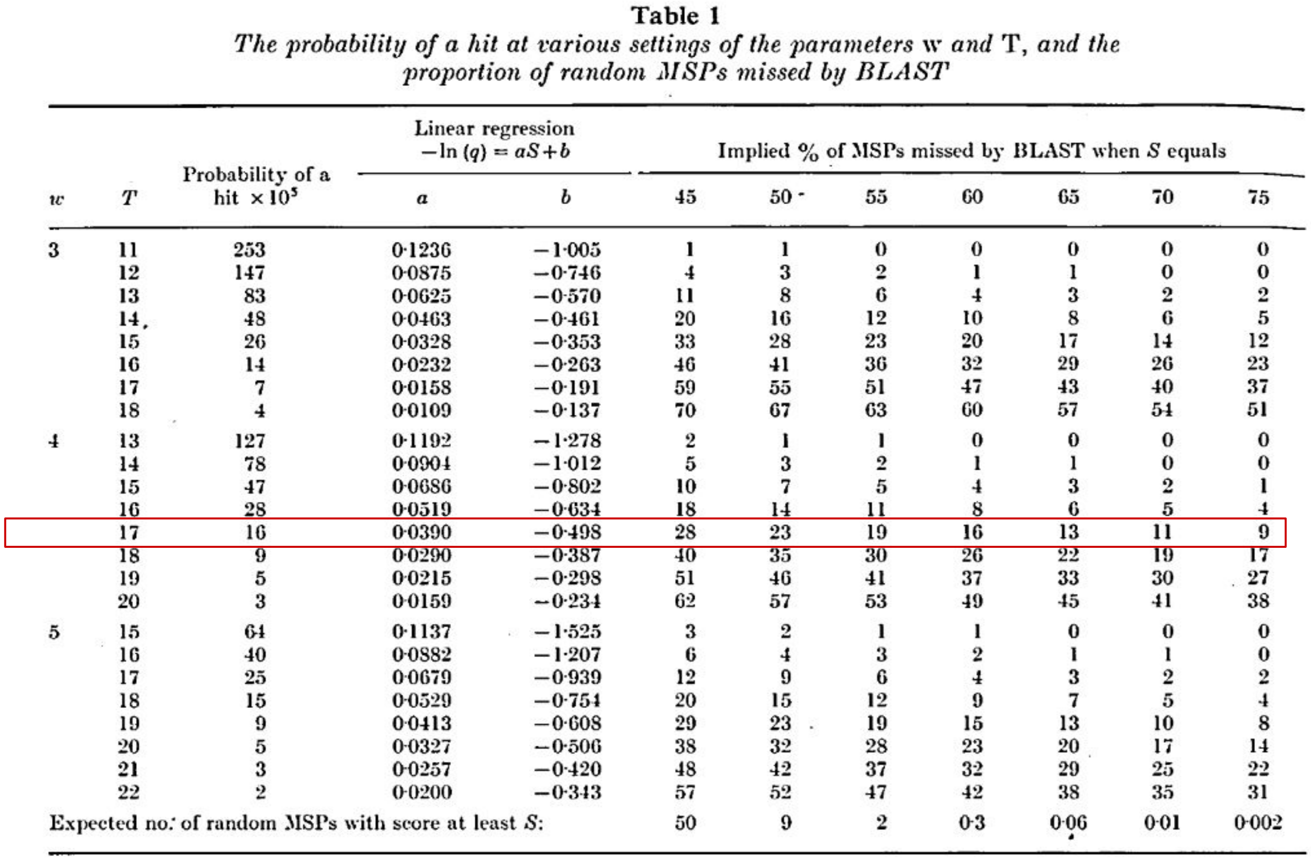


Table 1

The probability of a hit at various settings of the parameters w and T, and the proportion of random MSPs missed by BLAST

			Linear regression $-\ln(q) = aS+b$		Implied % of MSPs missed by BLAST when S equals						
w	T	Probability of a hit $\times 10^5$	a	b	45	50	55	60	65	70	75
3	11	253	0·1236	−1·005	1	1	0	0	0	0	0
	12	147	0·0875	−0·746	4	3	2	1	1	0	0
	13	83	0·0625	−0·570	11	8	6	4	3	2	2
	14	48	0·0463	−0·461	20	16	12	10	8	6	5
	15	26	0·0328	−0·353	33	28	23	20	17	14	12
	16	14	0·0232	−0·263	46	41	36	32	29	26	23
	17	7	0·0158	−0·191	59	55	51	47	43	40	37
	18	4	0·0109	−0·137	70	67	63	60	57	54	51
4	13	127	0·1192	−1·278	2	1	1	0	0	0	0
	14	78	0·0904	−1·012	5	3	2	1	1	0	0
	15	47	0·0686	−0·802	10	7	5	4	3	2	1
	16	28	0·0519	−0·634	18	14	11	8	6	5	4
	17	16	0·0390	−0·498	28	23	19	16	13	11	9
	18	9	0·0290	−0·387	40	35	30	26	22	19	17
	19	5	0·0215	−0·298	51	46	41	37	33	30	27
	20	3	0·0159	−0·234	62	57	53	49	45	41	38
5	15	64	0·1137	−1·525	3	2	1	1	0	0	0
	16	40	0·0882	−1·207	6	4	3	2	1	1	0
	17	25	0·0679	−0·939	12	9	6	4	3	2	2
	18	15	0·0529	−0·754	20	15	12	9	7	5	4
	19	9	0·0413	−0·608	29	23	19	15	13	10	8
	20	5	0·0327	−0·506	38	32	28	23	20	17	14
	21	3	0·0257	−0·420	48	42	37	32	29	25	22
	22	2	0·0200	−0·343	57	52	47	42	38	35	31
Expected no. of random MSPs with score at least S:					50	9	2	0·3	0·06	0·01	0·002

Results - Word Length and Threshold Parameters

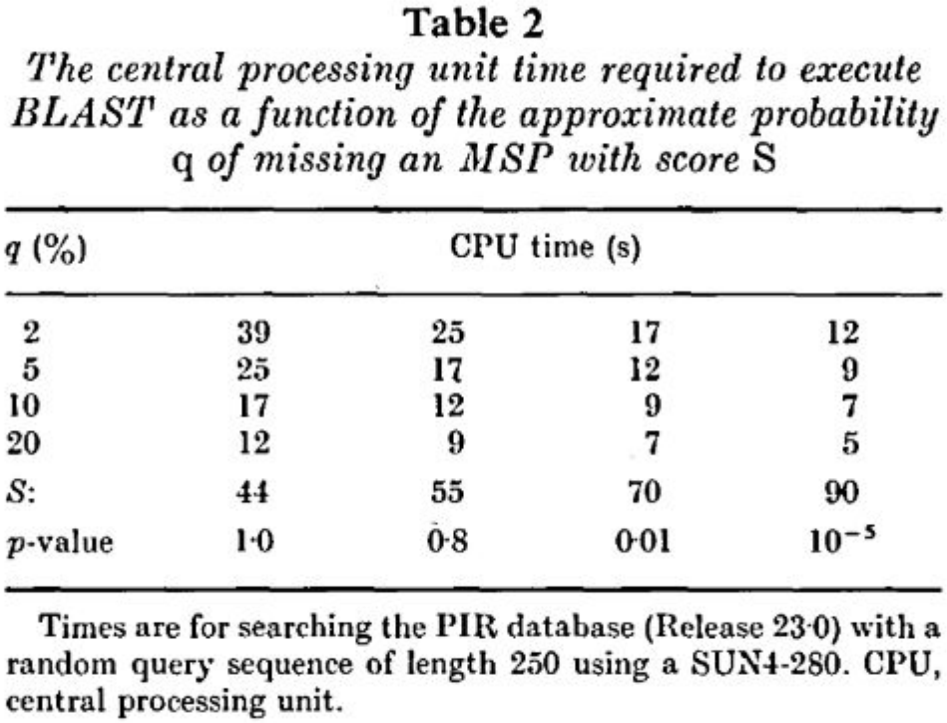


Table 2

The central processing unit time required to execute BLAST as a function of the approximate probability q *of missing an MSP with score* S

q (%)	CPU time (s)			
2	39	25	17	12
5	25	17	12	9
10	17	12	9	7
20	12	9	7	5
S:	44	55	70	90
p-value	1·0	0·8	0·01	10^{-5}

Times are for searching the PIR database (Release 23·0) with a random query sequence of length 250 using a SUN4-280. CPU, central processing unit.

Results - Performance of BLAST with Homologous Sequences

REAL DATA : Proteins compared to superfamilies

Computing the true MSP score with the BLAST approximation (constant w=4)

- **43 misses**, not 24 misses (T=17)
  - Uniform pattern of conservation
- **2 misses**, not 8 misses (T=17)

*PIR - Protein Information Resource

True MSPs

BLAST Approximation

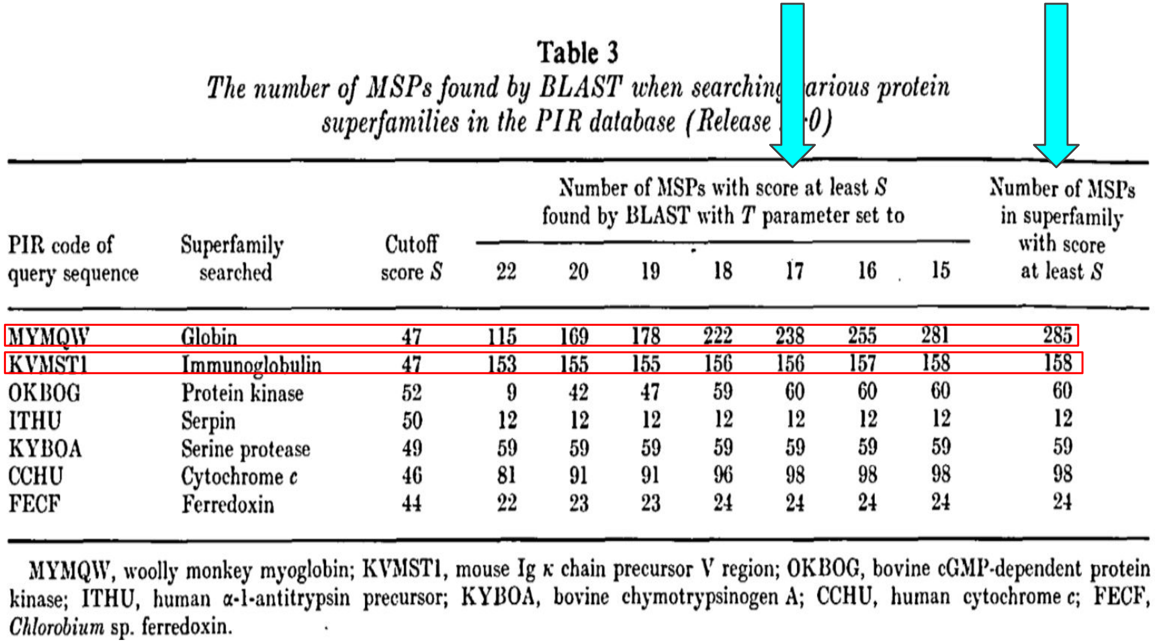


Table 3

The number of MSPs found by BLAST when searching various protein superfamilies in the PIR database (Release [illegible]0)

PIR code of query sequence	Superfamily searched	Cutoff score *S*	Number of MSPs with score at least *S* found by BLAST with *T* parameter set to							Number of MSPs in superfamily with score at least *S*
			22	20	19	18	17	16	15	
MYMQW	Globin	47	115	169	178	222	238	255	281	285
KVMST1	Immunoglobulin	47	153	155	155	156	156	157	158	158
OKBOG	Protein kinase	52	9	42	47	59	60	60	60	60
ITHU	Serpin	50	12	12	12	12	12	12	12	12
KYBOA	Serine protease	49	59	59	59	59	59	59	59	59
CCHU	Cytochrome *c*	46	81	91	91	96	98	98	98	98
FECF	Ferredoxin	44	22	23	23	24	24	24	24	24

MYMQW, woolly monkey myoglobin; KVMST1, mouse Ig κ chain precursor V region; OKBOG, bovine cGMP-dependent protein kinase; ITHU, human α-1-antitrypsin precursor; KYBOA, bovine chymotrypsinogen A; CCHU, human cytochrome *c*; FECF, *Chlorobium* sp. ferredoxin.

Results - Performance of BLAST with Homologous Sequences

Distribution of mutations **more clustered** than predicted by Poisson process

BLAST approximation perform **better on real sequences** than the random model

Finding high-scoring MSPs **quickly**

Comparable sensitivity and yields **fewer false positives**

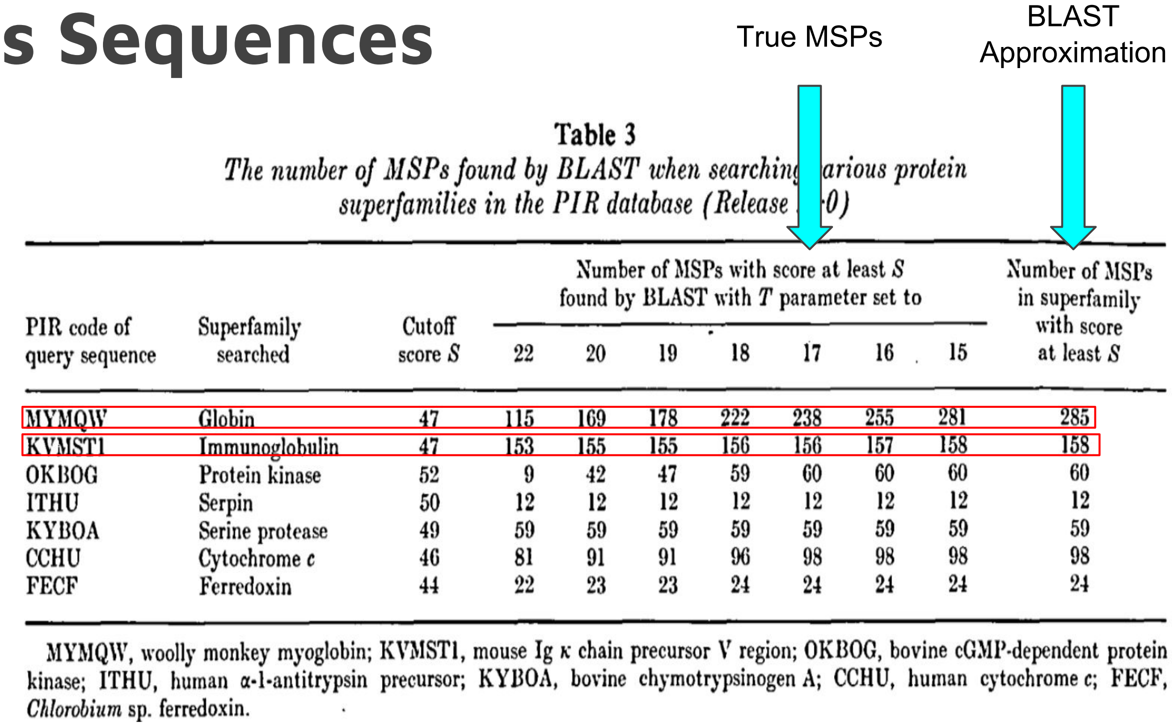

True MSPs
BLAST Approximation

Table 3

The number of MSPs found by BLAST when searching various protein superfamilies in the PIR database (Release [illegible]·0)

PIR code of query sequence	Superfamily searched	Cutoff score S	Number of MSPs with score at least S found by BLAST with T parameter set to							Number of MSPs in superfamily with score at least S
			22	20	19	18	17	16	15	
MYMQW	Globin	47	115	169	178	222	238	255	281	285
KVMST1	Immunoglobulin	47	153	155	155	156	156	157	158	158
OKBOG	Protein kinase	52	9	42	47	59	60	60	60	60
ITHU	Serpin	50	12	12	12	12	12	12	12	12
KYBOA	Serine protease	49	59	59	59	59	59	59	59	59
CCHU	Cytochrome c	46	81	91	91	96	98	98	98	98
FECF	Ferredoxin	44	22	23	23	24	24	24	24	24

MYMQW, woolly monkey myoglobin; KVMST1, mouse Ig κ chain precursor V region; OKBOG, bovine cGMP-dependent protein kinase; ITHU, human α-1-antitrypsin precursor; KYBOA, bovine chymotrypsinogen A; CCHU, human cytochrome c; FECF, *Chlorobium* sp. ferredoxin.

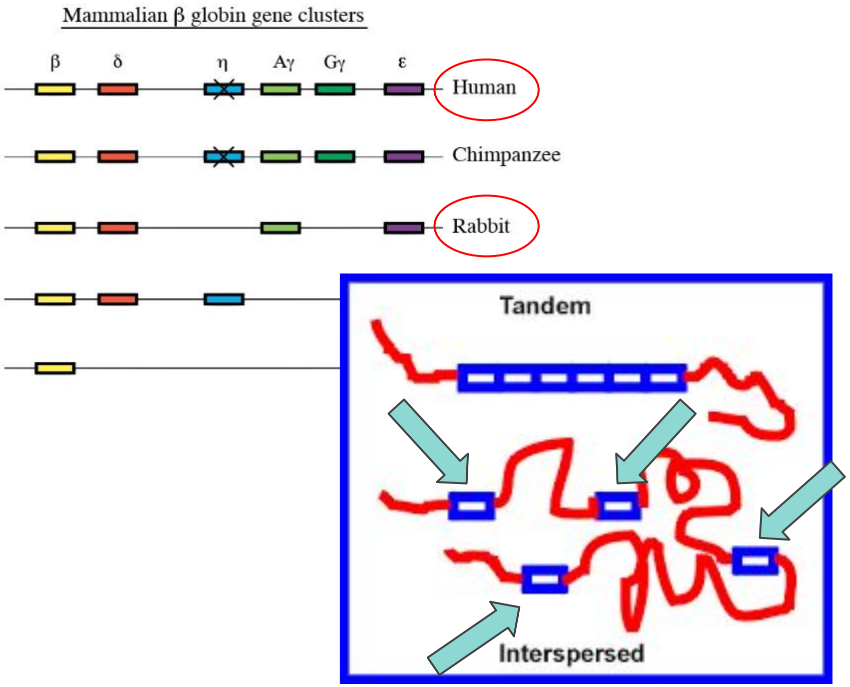
Results - Performance of BLAST with Long Sequences

Locate locally similar regions that can be aligned **without gaps**

Three main classes of similar regions are exhibited: **genes, long interspersed repeats, certain anticipated weaker similarities**

LINE (long interspersed repeat sequences)

Intergene similarities within b-clusters gene



https://biologos.org/blogs/dennis-venema-letters-to-the-duchess/pseudogenes-intelligent-design-and-kitzmiller--part-2

Results - Performance of BLAST with Long Sequences

Applied variant (match score: 5, mismatch score = -4)

Smaller w give **more alignments**

Provides **no essential new information**

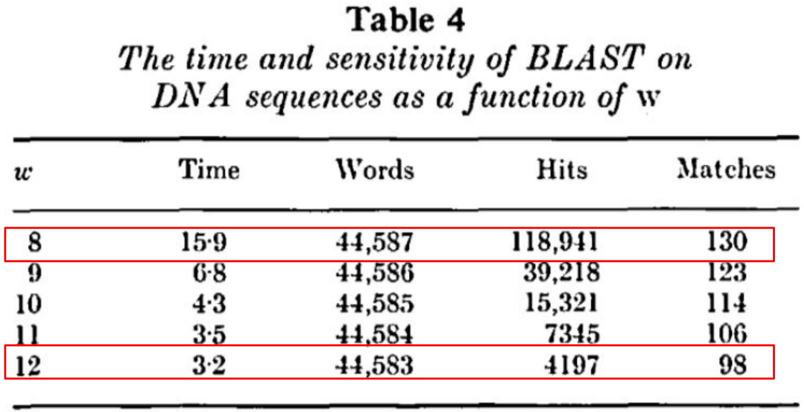


Table 4

The time and sensitivity of BLAST on DNA sequences as a function of w

w	Time	Words	Hits	Matches
8	15·9	44,587	118,941	130
9	6·8	44,586	39,218	123
10	4·3	44,585	15,321	114
11	3·5	44,584	7345	106
12	3·2	44,583	4197	98

Conclusions

BLAST can be implemented in a number of ways and utilized in a variety of contexts.

Variation:

- **Allow gaps** in the extension step
- **Shared memory** version that loads compressed DNA file into memory **once** to allow subsequent searches to **skip this step**
- Compare **DNA sequence to protein database** to allow six reading frames
- **Detect distant protein** homologies
- Permits a **fast programs** for database searching