Development of Complex Curricula for Molecular Bionics and Infobionics Programs within a consortial* framework**

Consortium leader

PETER PAZMANY CATHOLIC UNIVERSITY

Consortium members

SEMMELWEIS UNIVERSITY, DIALOG CAMPUS PUBLISHER

The Project has been realised with the support of the European Union and has been co-financed by the European Social Fund ***

**Molekuláris bionika és Infobionika Szakok tananyagának komplex fejlesztése konzorciumi keretben

***A projekt az Európai Unió támogatásával, az Európai Szociális Alap társfinanszírozásával valósul meg.

INTRODUCTION TO BIOINFORMATICS

(BEVEZETÉS A BIOINFORMATIKÁBA)

CHAPTER 4

Similarity searching and the BLAST algorithm

(hasonlóságkeresés és a BLAST algoritmus)

Sándor Pongor

2011.10.09..
TÁMOP – 4.1.2-08/2/A/KMR-2009-0006
2

Lecture outline

Similarity searching principles and main steps,

Sequence similarity, PAM and BLOSUM matrices

Alignment types (local, global, exhaustive, heuristic)

FASTA (briefly)

BLAST (principle)

Significance calculation

BLAST refinements

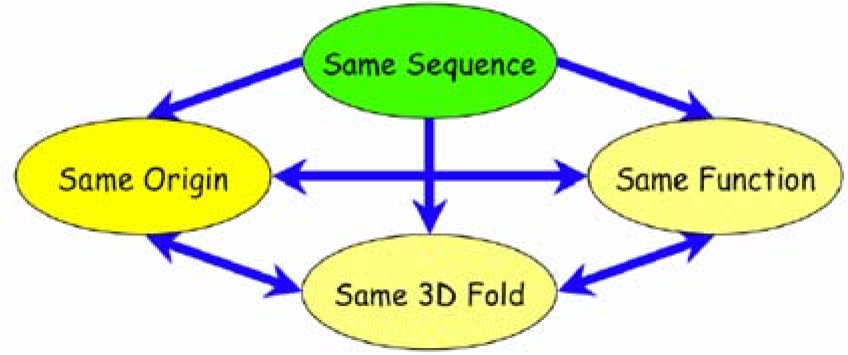
What similarity searching is..

Given a query and a database, find the entry in the database that is most similar to the query in terms of a numerical similarity measure (distance, similarity score, etc.)

In contrast: retrieval looks for an exact match to the query.

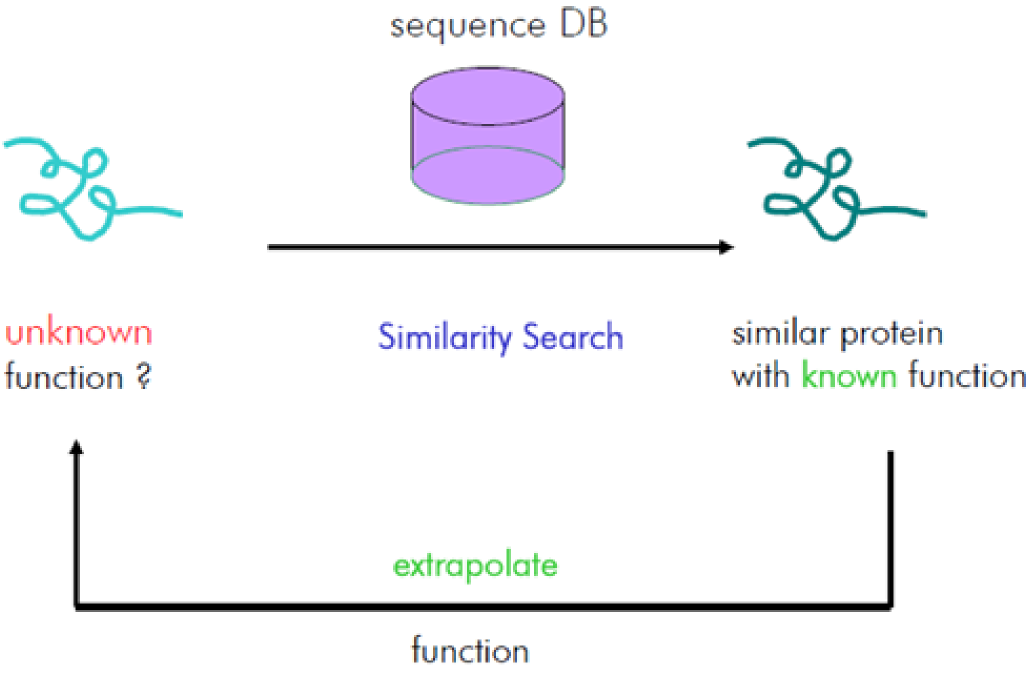
Is John in the list? Retrieval: Yes/No, based on *exact matching*. Similarity search: His brother Joe Brown is. So we can *classify* John into the Brown family, based on *approximate matching*.

The importance of similarity



similar sequences: probably have the same ancestor, share the same structure, and have a similar biological function

The use of similarity



Similar protein’s name (ID): Joe.
Class: Brown family

Starting stage: Query and DB are in the same format (the search format) and we have a similarity measure.

STEPS:

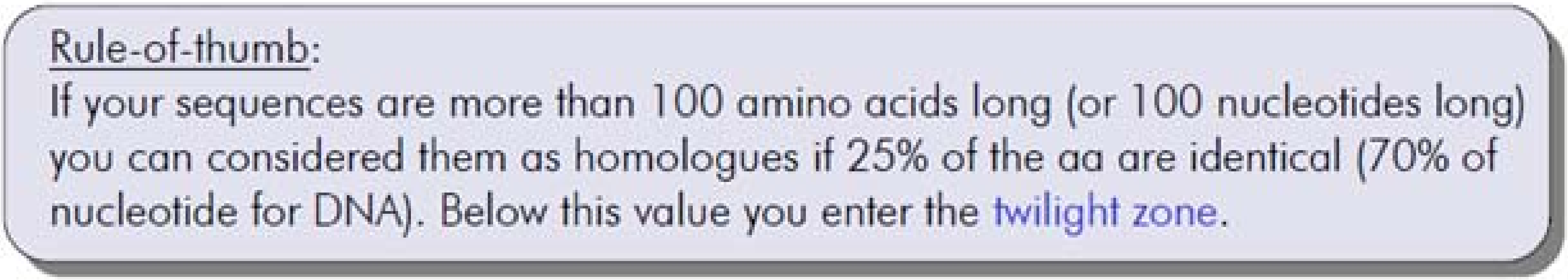
1. Compare query with all entries in the DB and register similarity score. Store results above some threshold (cutoff)
2. Calculate significance of the score
3. Rank entries according to similarity score or significance (top list)
4. Report the best hit (usually after some simple statistics, e.g. if it is higher than a threshold…)

We use a version of the edit distance and a specific substitution matrix (Dayhoff, BLOSUM, etc.)

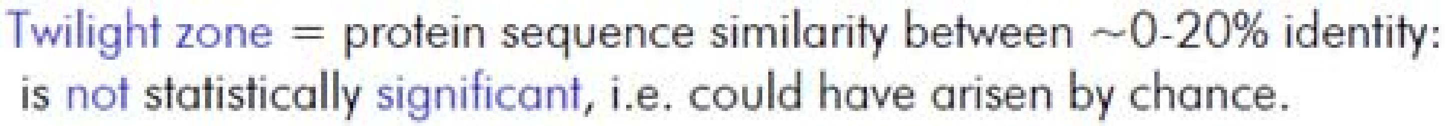
Exhaustive algorithms (Dynamic programming, Needleman Wunsch, Smith-Waterman) are expensive,

We use heuristics that make use of the properties of biological sequences (FASTA, BLAST)

Biological heuristics include a) local similarities are dense, b) similar regions are near each other, c) low complexity sequences excluded, etc.



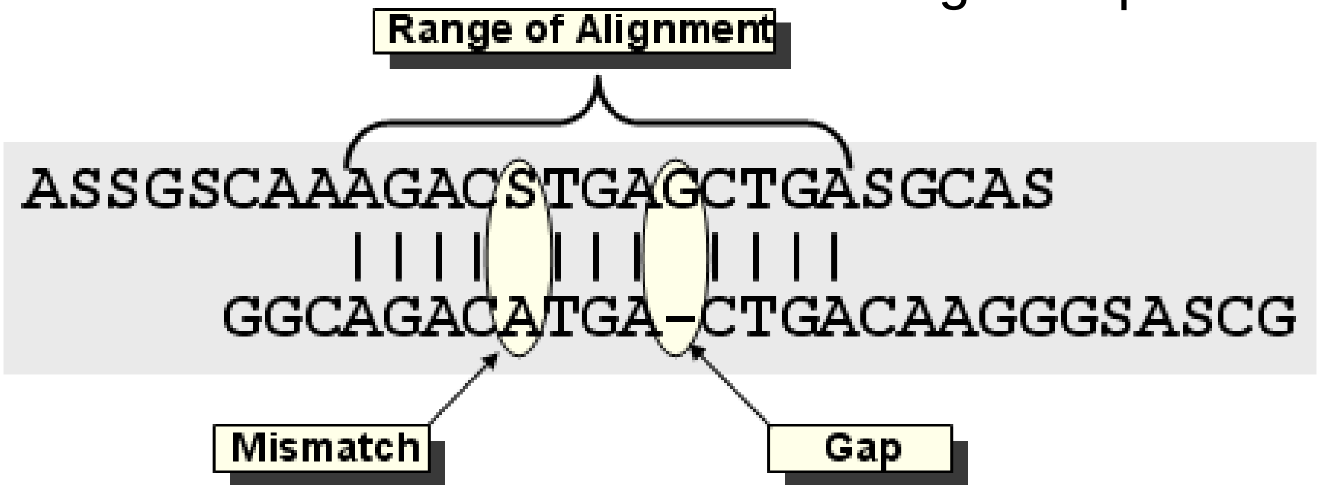
Rule-of-thumb:
If your sequences are more than 100 amino acids long (or 100 nucleotides long) you can considered them as homologues if 25% of the aa are identical (70% of nucleotide for DNA). Below this value you enter the twilight zone.



Twilight zone = protein sequence similarity between ~0-20% identity: is not statistically significant, i.e. could have arisen by chance.

Sequence similarity score

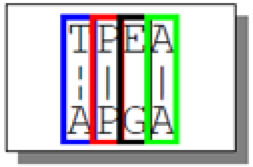
HSP, high scoring segment pair



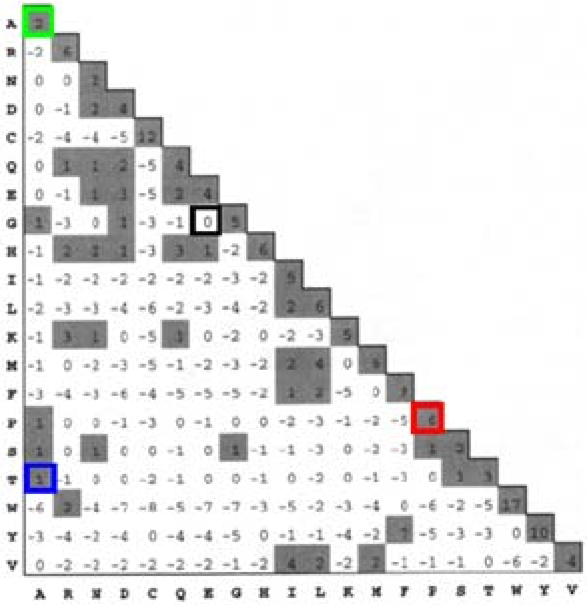
The score S is a sum of costs assigned to identities and mismatches, minus a penalty for gaps. Costs are stored in the substitution matrix

A simple example (without gaps):

Raw score of an alignment



Score = 1 + 6 + 0 + 2 = 9



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

For a match/mismatch we look up the value in the substitution matrix. The matrix is a lookup table…

Substitution matrices in details

The susbstitution matrix (also called scoring matrix) contains costs for amino acid identities and substitutions in an alignment.

It is a *20x20* symmetrical matrix that can be constructed from pairwise alignments of related sequences

“Related” means either

a) evolutionary relatedness described by an “approved” evolutionary tree (Dayhoff’s PAM matrices)

b) any sequence similarity as described in the PROSITE database (Hennikoffs BLOSUM matrices)

Groups of related sequences can be organized into a multiple alignment for calculation of the matrix elements.

Calculation of scoring matrices
from multiple alignments.

```
ASDEAKLVV
 |
ATDDAKLSI
 |     |
ASDEERITV
```

Matrix elements are calculated from the observed and expected frequencies ("log odds" principle). E.g. for S/T (indicated by red):

$$M(S/T) = \log\left(\frac{f(S/T)}{f(S)\times f(T)}\right)$$

The values are calculated from many (not just one) multiple alignments. The log odds values in the matrix are then normalized to a range (e.g. -5 to +15) depending on the application

PAM matrices

Percent **A**ccepted **M**utation: Unit of evolutionary change for protein sequences [Dayhoff78].

Calculated from related sequences organized into “accepted” evolutionary trees (71 trees, 1572 exchange [only])

20x20 matrix, columns add up to the no of cases observed.

	A	R	N	D	C
A	9867	2	9	10	3
R	1	9913	1	0	1
N	4	1	9822	36	0
D	6	0	42	9859	0
C	1	1	0	0	9973

All entries × 10^4

Converted into scoring matrix by log-odds and scaling

Pam_1 = 1% of amino acids mutate

Pam_30 = $(\text{Pam\_1})^{30}$ *(matrix multiplication)*

PAM 250

(the higher the numbers the higher the divergence)

Note: chemically similar amino acids are near each other …

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

2011.10.09

BLOSUM matrices

PAM uses evolutionarily related sequences, so they may not apply to divergent proteins

Henikoff constructed the BLOSUM (BLOck SUbstitution Matrix) series in the same way, but using short blocks of divergent sequences taken from the PROSITE database of multiple alignments. No "grand theory" involved…

In BLOSUM 62, the sequences are less than 62% identical. The higher the number the less divergent the proteins (in contrast to PAM).

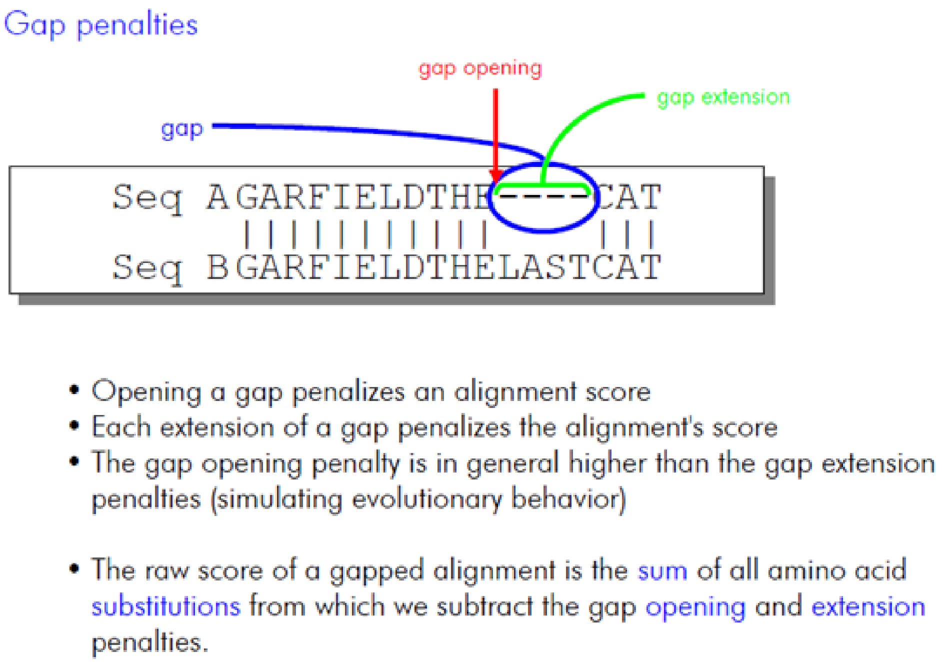
The most popular matrices today (they are deduced from much more alignments than PAM..)

Many other matrices possible

Unitary matrix: 1 if the characters are identical (diagonal elements), zero otherwise…

Such matrices are used for DNA...

Introduction to bioinformatics: Similarity Searching and BLAST



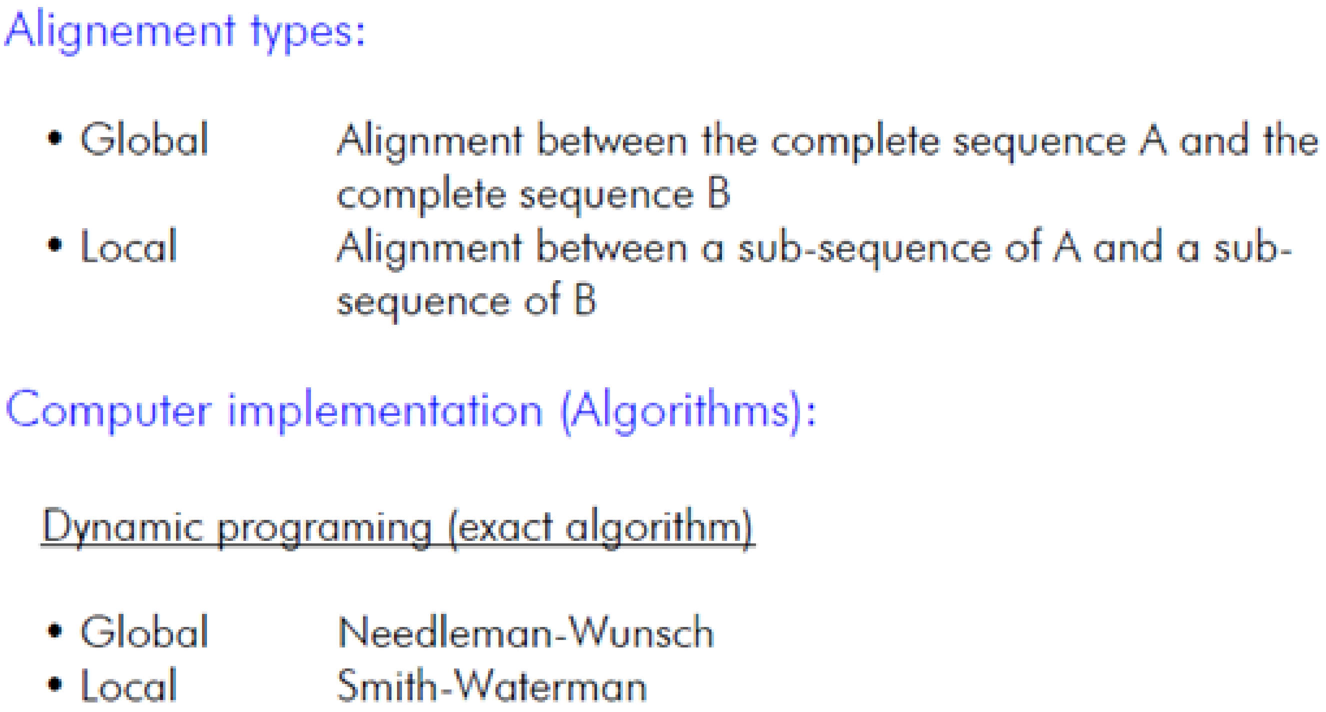
Gap penalties

```
Seq A GARFIELDTHE----CAT
      |||||||||||    |||
Seq B GARFIELDTHELASTCAT
```

- Opening a gap penalizes an alignment score
- Each extension of a gap penalizes the alignment's score
- The gap opening penalty is in general higher than the gap extension penalties (simulating evolutionary behavior)

- The raw score of a gapped alignment is the sum of all amino acid substitutions from which we subtract the gap opening and extension penalties.

Introduction to bioinformatics: Similarity Searching and BLAST



Alignement types:

- Global Alignment between the complete sequence A and the complete sequence B
- Local Alignment between a sub-sequence of A and a sub-sequence of B

Computer implementation (Algorithms):

Dynamic programing (exact algorithm)

• Global	Needleman-Wunsch
• Local	Smith-Waterman

Heuristic Sequence Alignment Why?

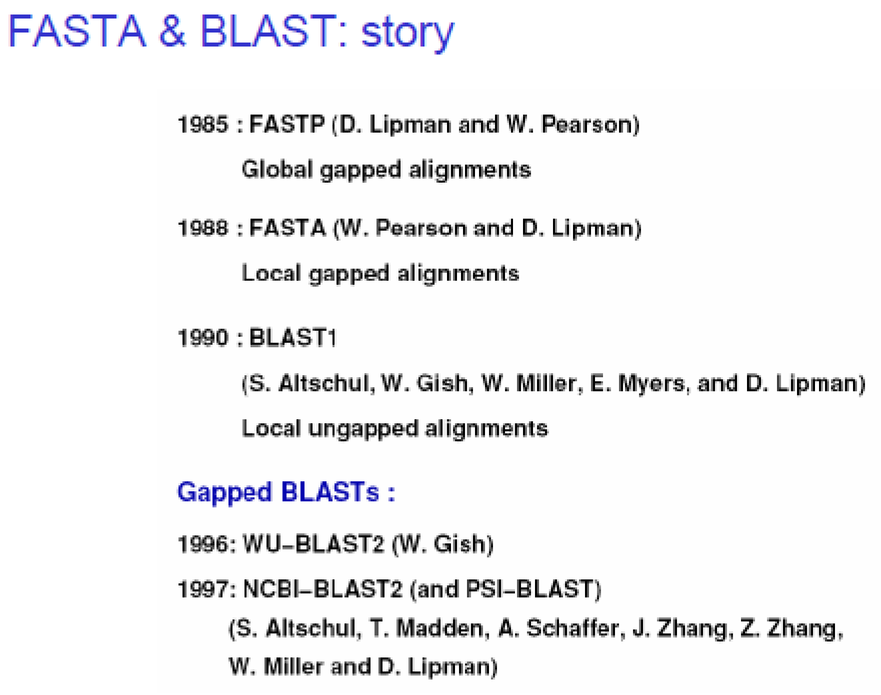
With the Dynamic Programming algorithm, time is proportional to the product of the lengths of the two sequences. This is too slow for genome analysis....

There are two methods that are at least 50-100 times faster than dynamic programming (FASTA and BLAST)

Dynamic Programming: computational method that provide the mathematically optimal alignment for two sequences, and a scoring system.

Heuristic Methods (e.g. BLAST, FASTA) prune the search space so they provide only aproximately best aligments. For related sequeces DP and heuristics give the same solution. For distantly related sequence the alignments differ...

Restricting the search space: a) Only search the selected sequences; b) Only scan some portions of the sequences (a part of the dynamic programming matrix)



FASTA & BLAST: story

1985 : FASTP (D. Lipman and W. Pearson)

Global gapped alignments

1988 : FASTA (W. Pearson and D. Lipman)

Local gapped alignments

1990 : BLAST1

(S. Altschul, W. Gish, W. Miller, E. Myers, and D. Lipman)

Local ungapped alignments

Gapped BLASTs :

1996: WU-BLAST2 (W. Gish)

1997: NCBI-BLAST2 (and PSI-BLAST)
(S. Altschul, T. Madden, A. Schaffer, J. Zhang, Z. Zhang,
W. Miller and D. Lipman)

The practical trick: Represent sequences as n-character words and positions. Transform the query or dbase into a hash table (list of words and positions). Makes things faaaast…

4 Steps of FASTA

1 Localize the 10 best regions of similarity between the two seq. Each identity between two "word" is represented by a dot

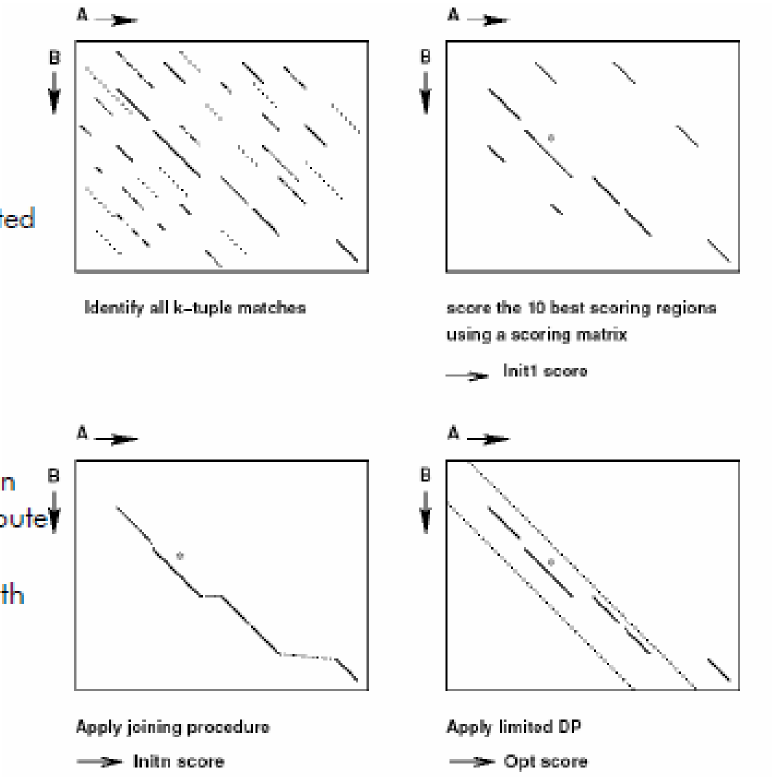
2 Each diagonal: ungapped alignment

The smaller the k, The sensitive the method but slower

(*k* is word size)

3 Find the best combination of the diagonals-> compute a score. Only those sequences with a score higher than a threshold will go to the fourth step

4 DP applied around The best scoring diagonal.



Identify all k–tuple matches

score the 10 best scoring regions using a scoring matrix

→ Init1 score

Apply joining procedure

→ Initn score

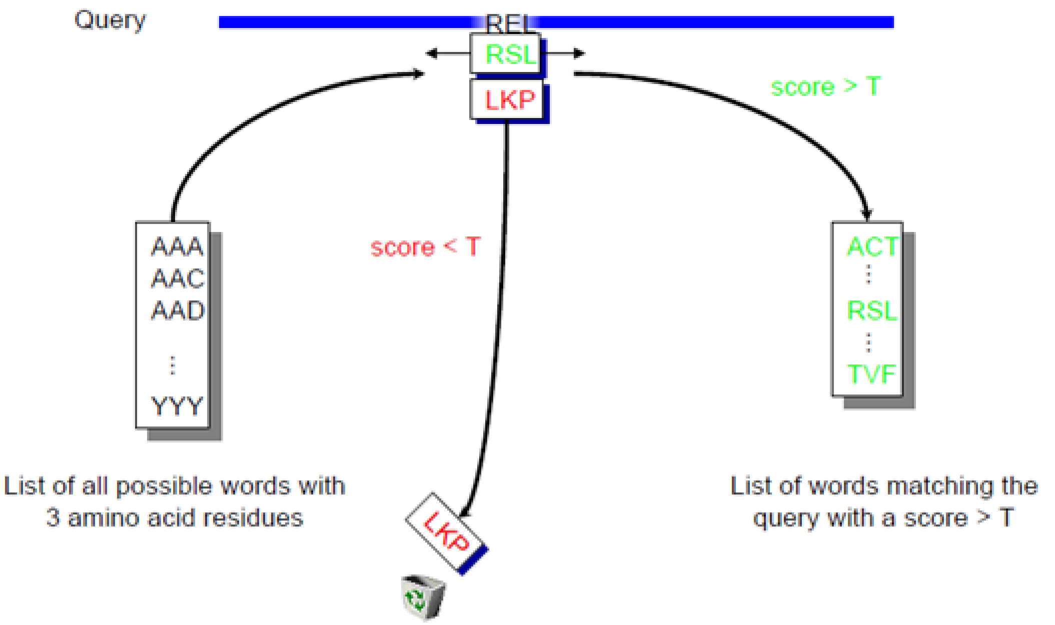
Apply limited DP

→ Opt score

BLAST algorithm in 4 steps

1. Blast algorithm: creating a list of similar words

⇨A substitution matrix is used to compute the word scores

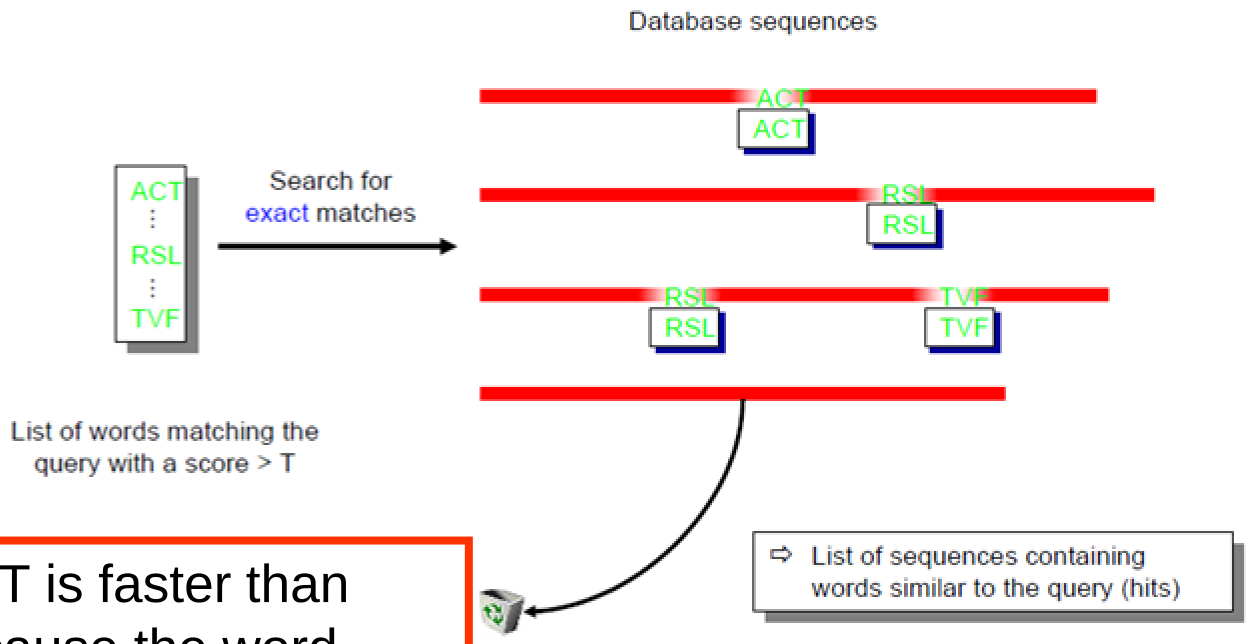


List of all possible words with 3 amino acid residues

List of words matching the query with a score > T

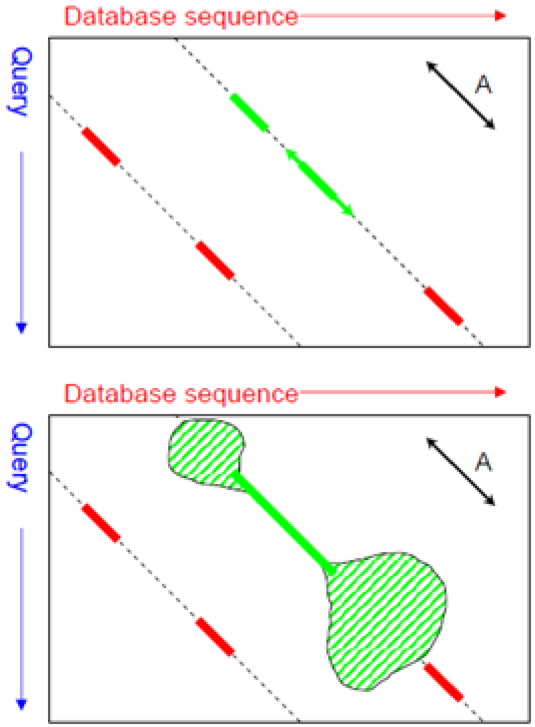
2011.10.09..

2. Blast algorithm: eliminating sequences without word hits



Note: BLAST is faster than FASTA) because the word occurrences in the dbase are pre-computed in a hash table

3. Blast algorithm: extension of hits



Ungapped extension if:
- 2 "Hits" are on the same diagonal but at a distance less than A

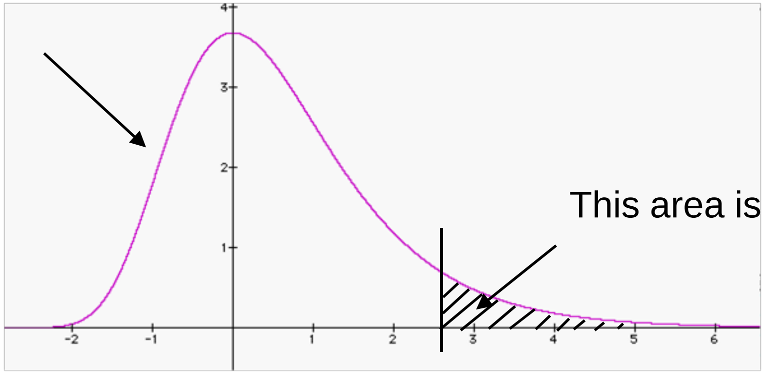
Extension using dynamic programming
- limited to a restricted region

Originally (BLAST1) the aligned regions (HSPs, high scoring pairs) were extended until the score went negative. The above, “2 hits” requirement exists since BLAST2, and increased accuracy dramatically.

2011.10.09..

Statistical significance

Density function (integral=1.0)



This area is the *p*-value

The *p*-value is the probability of observing data at least as extreme as that being observed.

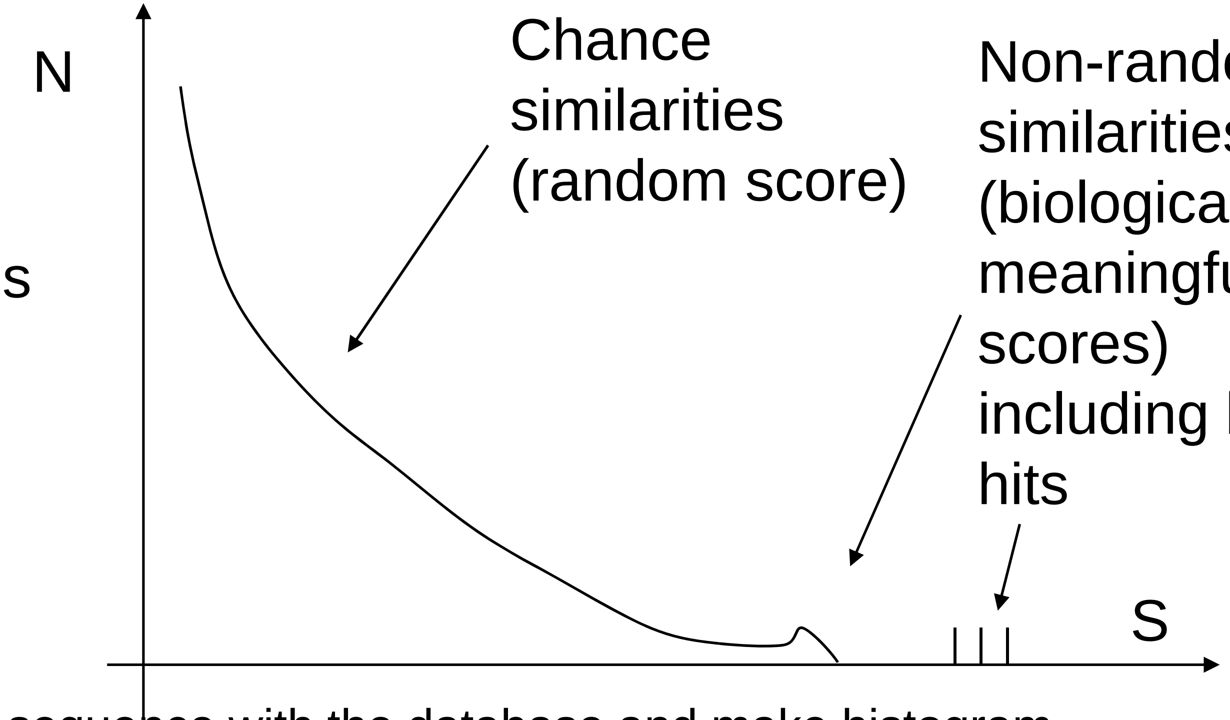
An engineer’s guide to significance

Significance: 1) The probability of finding a score by chance (p-value) ; 2) The number of times you expect to find a score >= a certain value by chance (E-value). (the smaller, the better)

You can estimate *p* by making a histogram of chance (random) scores, linearizing it and reading *p* from the linear curve.

An engineer’s guide to significance

A typical distribution of scores S



Frequency (No of times found in dbase)

Chance similarities (random score)

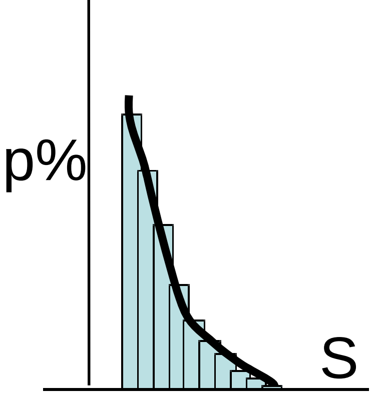
Non-random similarities (biologically meaningful scores) including best hits

1) Compare a sequence with the database and make histogram

2) Almost always: biologically meaningful scores are a negligible minority : ~the whole distribution is dominated by random scores

Estimation of significance from an unkown distribution

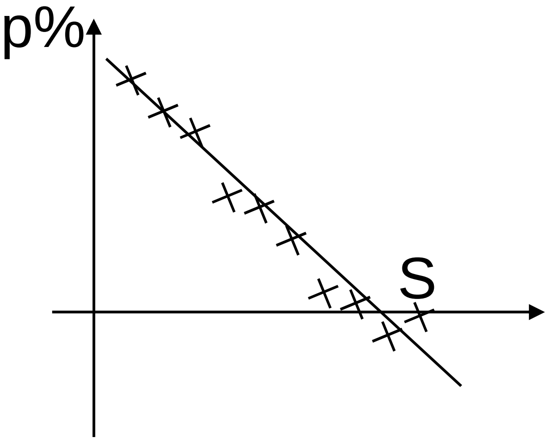
1) Draw % histogram of chance similarities



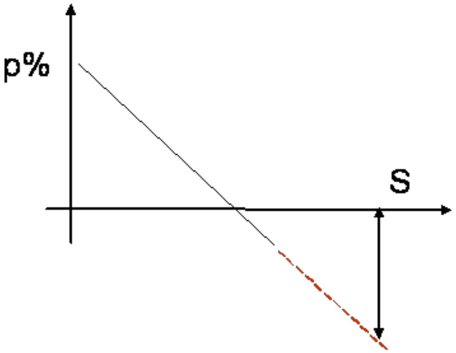
Linearize:

try log-lin, lin-log, log-log transformations

2) Fit straight line

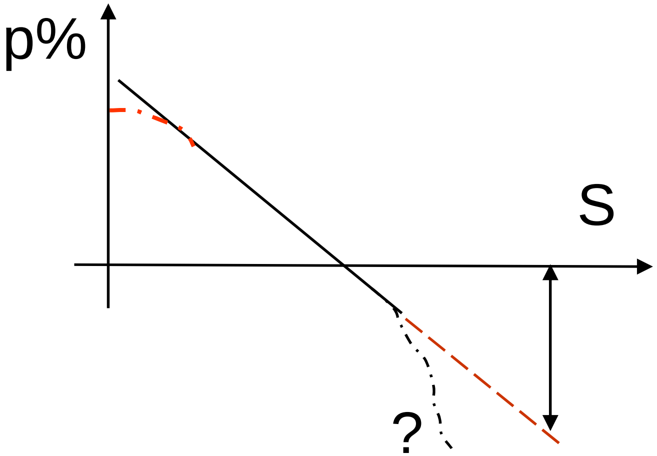


3) Read significance (y value) of a score S (x value) from curve.



An engineer's guide to significance

Estimating significance from an unknown distribution

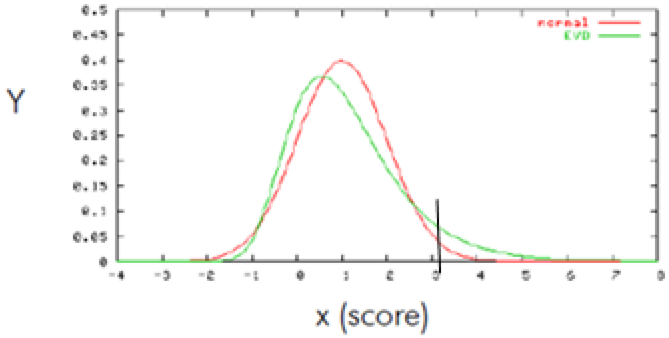


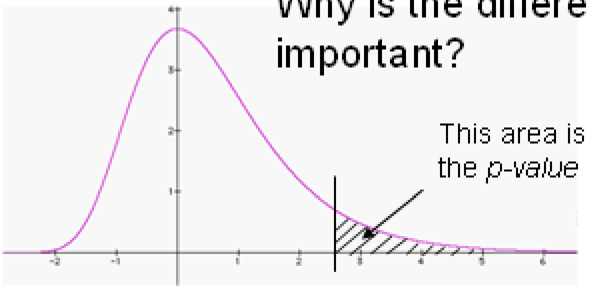
Where to get distribution data: 1) comparison with real sequences, omitting largest scores. 2) Using simulated, random-shuffled sequences. Neither is "correct" but both work quite well

Usually one has to extrapolate quite far since large S values are rare (red line)… True, but there is no other way.

Alignment Scores follow Extreme Value distribution

- Karlin and Altschul observed that in the framework of local alignments without gaps: the distribution of random sequence alignment scores follow an EVD.





Why is the difference important?

$$Y = \lambda \exp[-\lambda(x-\mu) - e^{-\lambda(x-\mu)}]$$

The p-value is the probability of observing data at least as extreme as that observed.

μ, λ : parameters depend on the length and composition of the sequences and on the scoring system

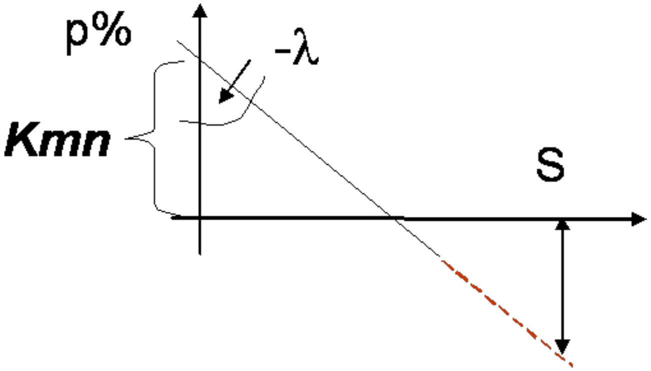
Just as the sum of many independent identically distributed (i.i.d) random variables tends to a normal distribution, the maximum of a large number of i.i.d. random variables tends to an extreme value distribution (EVD). Since use maximal alignments between query and db sequences, EVD is applicable!

Important formulas

The *Karlin-Altschul statistics* is based on Extreme Value distribution, the expected no of HSP-s, with score at least S is

$$E = Kmne^{-\lambda S}$$

where *m* is the query length, *n* is the database length, *K* and *λ* are constants. *m n* is called the search space.



Further important formulas

The raw score *S* depends on the scoring system (matrix), *K* and *λ*. The normalized bit-score *S'* is more "portable"

The probability *P* of finding at least one HSP with score >=S is

$$S' = \frac{\lambda S - \ln K}{\ln 2}$$

where E (right) is calculated by the Karlin-Altschul formula

$$P = 1 - e^{-E} = 1 - e^{-Kmne^{-\lambda S}}$$

3 facts to remember

1) For a database of N sequences, E= *p* x *N*

2) For real protein sequence similarities, *p* values are very very small. E-values are bigger, but for P<0.000001, P and E are practically identical…

3) Local alignment without gaps:
– Theoretical work: Karlin-Altschul statistics: → Extreme Value Distribution
– Local alignments with gaps:
– Empirical studies (shuffled sequences) : → Extreme Value Distribution.

2011.10.09..

How does BLAST calculate E-values?

- Every BLAST "hit" has a score, *x*, derived from the substitution matrix.
- Parameters for the EVD have been **previously calculated** and stored for *m* (the length of the database) and *n* (the length of the query).
- Now we can get P(S≥x), which is our "*p-value*"
- To get the expected number of times this score will occur over the whole database, we multiply by *m*. This is the “*e-value*” you see reported in BLAST.

Increasing BLAST specificity: removal of aspecific (biased composition) regions

SEG

Repetitive sequences will aspecifically match with many queries

```
CSGSCTECT   seq_1
CCCGCCGCC   seq_2
```

Sequence complexity is an empirical measure, proportional to the number of words (of arbitrary length) necessary to reproduce a sequence. Seq_2 is of *low complexity* because it can be rewritten using CC and CG only.

Low complexity regions have a biased composition, they are often very repetitive. SGSGSGS, GGGGG etc.

Low complexity regions can be removed replaced by XXX so that they will not take part in the alignment (SEG program of John Wootton). Has threshold parameters…

Problem: some interesting sequences ARE of low complexity

Increasing BLASTspecificity: iterative, position specific scoring 1

PSI-BLAST

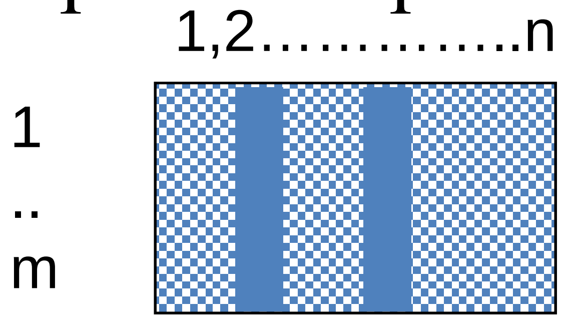
Multiple alignments are much more informative than simple alignments or similarity scores

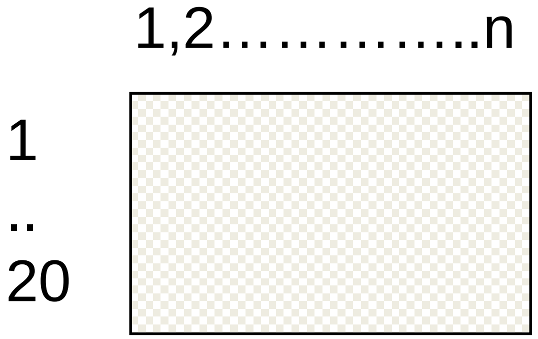
A multiple alignment of length *n* can be transformed into a frequency matrix of n x 20, which can be used as a query (in BLAST or in dynamic programming)

The PSI BLAST program can iteratively build such a matrix and use it in more and more specific searches.

Increasing BLASTspecificity: iterative, position specific scoring 2

PSI-BLAST





1) Multiple alignment of *n* positions (arbitrary no. *m* of sequences)

2) 20 x n position specific frequency matrix. Each cell is the % frequency of occurrence of an aa in that position.

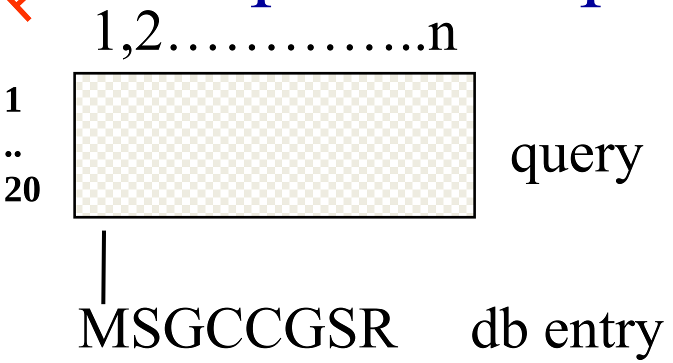
query

MSGCCGSR   db entry

3) Use the frequency matrix as a query

PSI-BLAST

Increasing BLASTspecificity: iterative, position specific scoring 3



PSI-BLAST iteratively includes new sequences into the multiple alignment

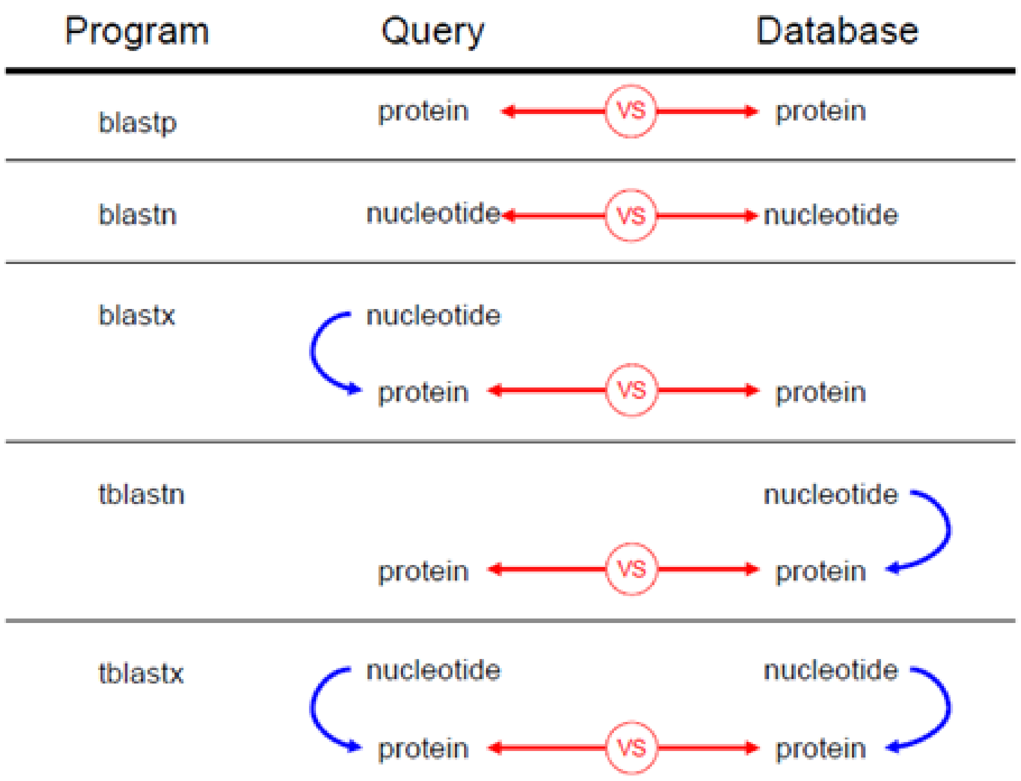
Comparing amin acid M of the entry with position 1 of the query yields a score S_1

$$S_1 = \sum_{i=1}^{20} f_i \times b_{M,i}$$

where the sum goes through the amino acids, f_i is the element of the frequency matrix and $b_{M,i}$ is the element of the BLOSUM matrix for M and amino acid *i* This is the ~same as using BLOSUM or PAM as a lookup table, so the alignment can be carried out by the same algorithm (BLAST, DP) !!! (f_i values are like weights)

BLAST: Basic Local Alignment Tool

Different programs are available according to the type of query



Program	Query		Database
blastp	protein	VS	protein
blastn	nucleotide	VS	nucleotide
blastx	nucleotide protein	VS	protein
tblastn	protein	VS	nucleotide protein
tblastx	nucleotide protein	VS	nucleotide protein

BLASTing with protein sequence queries:

blastp = Compares a protein sequence with a protein database. If you want to find something about the function of your protein, use blastp to compare your protein with other proteins contained in the databases

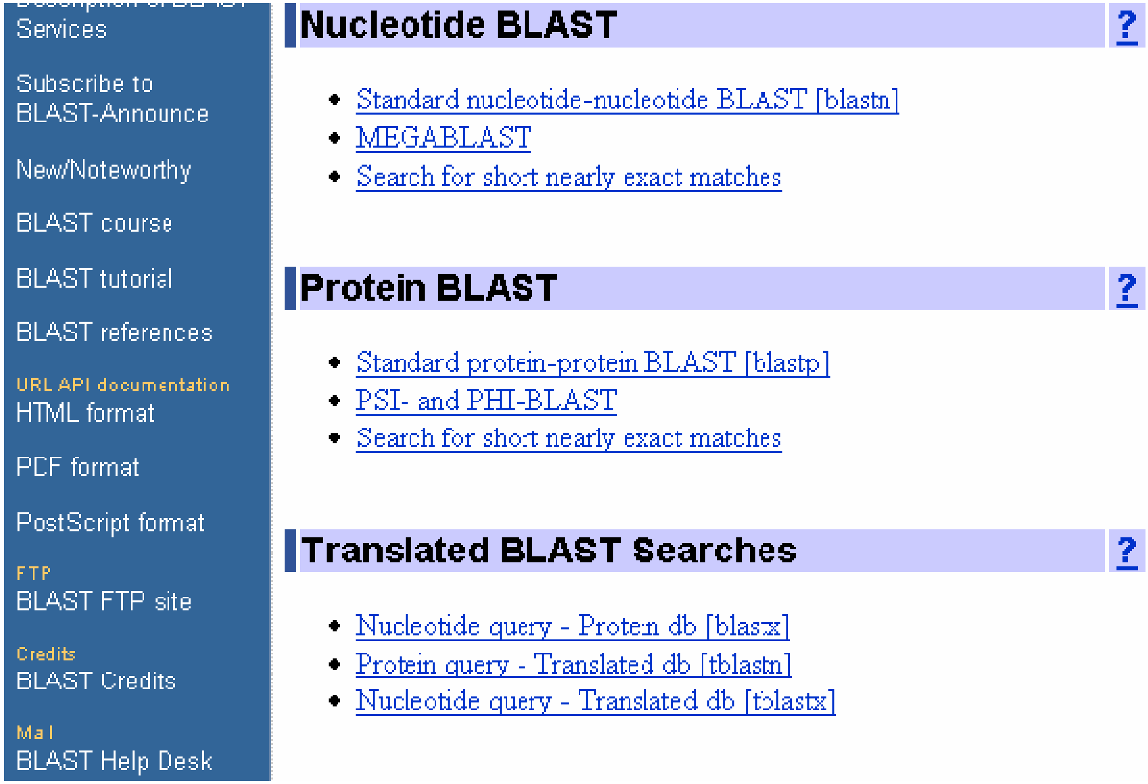
tblastn = Compares a protein sequence with a nucleotide database. If you want to discover new genes encoding proteins, use tblastn to compare. your protein with DNA sequences translated into their six possible reading frames

2011.10.09..

BLASTing with protein sequence queries:

At the NCBI BLAST server:

URL: http://www.ncbi.nlm.nih.gov/BLAST



Description of BLAST Services

Subscribe to BLAST-Announce

New/Noteworthy

BLAST course

BLAST tutorial

BLAST references

URL API documentation

HTML format

PCF format

PostScript format

FTP

BLAST FTP site

Credits

BLAST Credits

Mail

BLAST Help Desk

Nucleotide BLAST ?

- Standard nucleotide-nucleotide BLAST [blastn]
- MEGABLAST
- Search for sho:t nearly exact matches

Protein BLAST ?

- Standard protein-protein BLAST [blastp]
- PSI- and PHI-BLAST
- Search for sho:t nearly exact matches

Translated BLAST Searches ?

- Nucleotide query - Proten db [blasx]
- Protein query - Translated db [tblastn]
- Nucleotide query - Translated db [tolastx]

2011.10.09..

BLASTing with protein sequence queries:

European BLAST services:

EXPASY Switzerlannd
http://www.expasy.org/tools/blast/

EBI Hinxton UK

http://www.ebi.ac.uk/Tools/sss/

What you should know

Similarity searching, main steps

Sequence alignment (global, local, exhaustive, heuristic)

FASTA algorithm

BLAST algorithm

Reading significance from linearized histogram

BLAST statistics (E-value, p-value), how is BLAST calculating them…

Refinements (SEG, PSI-BLAST)

Different kinds of BLAST programs…

2011.10.09..