Sequence alignment

杨建益

Email: yangjy@nankai.edu.cn

Webpage: http://yanglab.nankai.edu.cn/

Course: http://yanglab.nankai.edu.cn/teaching/bioinformatics/

Office: 数学科学学院，419室

Content

1. Why to make sequence alignment?
2. What is a sequence alignment?
3. How to derive a mutation matrix-PAM
4. How to derive a mutation matrix-BLOSUM
5. Gap penalty
6. Dynamic programming
   a. Global alignment: Needleman-Wunsch
   b. Local alignment: Smith-Waterman
7. Heuristic algorithms

Heuristic algorithms

- One of the major task of database mining is to search for homology of a query sequence against a large sequence database such as UniProt which involves millions of sequences. Although dynamic programming can provide accurate solution of alignment, it is too slow for large scale database searching.

- Some heuristic algorithms, FASTA and BLAST, are designed to provide approximate alignment but with significantly increased speed (~50 times faster).

FASTA

1. FASTP
Lipman & Pearson, Science (1985) 227, 1435

2. FASTA
Pearson & Lipman, PNAS (1988) 85, 2444.

3. Lookup table
Dumas & Ninio, NAR (1982) 197.

https://www.ebi.ac.uk/Tools/sss/fasta/

FASTA

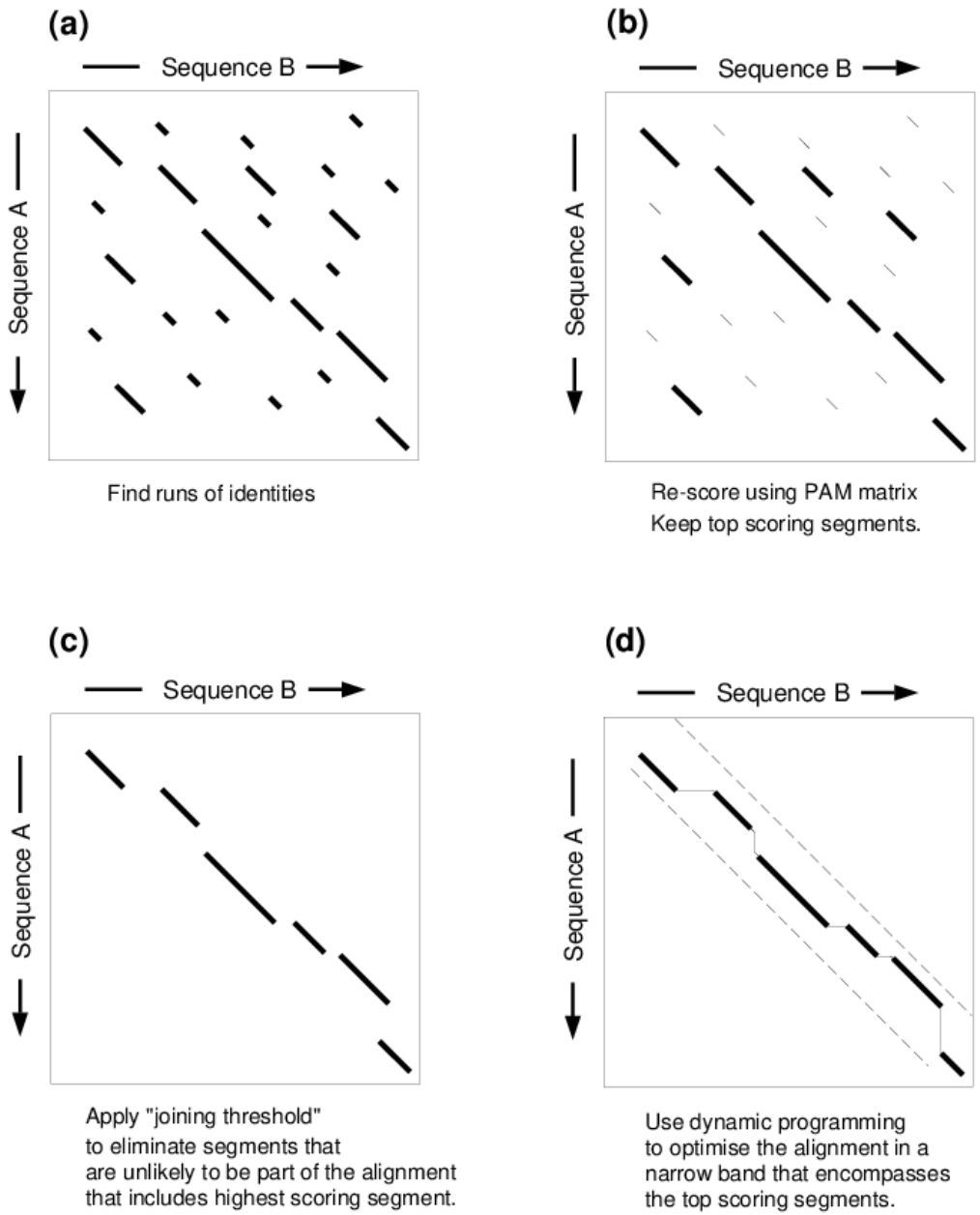
Four steps:

1. Identify common k-word (look-up table)

2. Score diagonals (PAM) to find 10 best diagonals

3. Join high scoring diagonals

4. Optimize alignment by DP

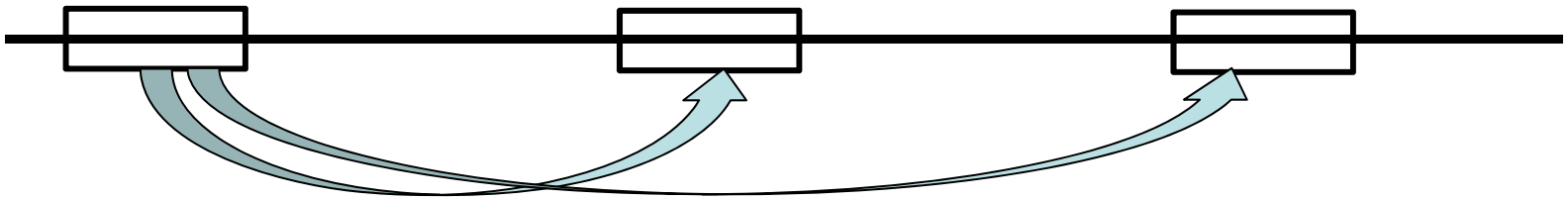


Dumas-Ninio look-up table

Original question:

For a sequence of length N, how to quickly find whether or not it contains repeated subsequences (length =k)?

Naïve methods: comparing every word with every other word of the sequence.



The time cost will increase with $O(N^2/2)$.

J P Dumas and J Ninio. Efficient algorithms for folding and comparing nucleic acid sequences. Nucleic Acids Res (1982) 10: 197-206.

Dumas-Ninio look-up table (example of k=2)

Question:

Given a sequence "TCGGATTCGTACGGTACGGATC", how to quickly find the locations of all the most frequently appeared words (length=2)?

1, Label the sequences by numbers

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22
T	C	G	G	A	T	T	C	G	T	A	C	G	G	T	A	C	G	G	A	T	C
TC	CG	GG	GA	AT	TT	TC	CG	GT	TA	AC	CG	GG	GT	TA	AC	CG	GG	GA	AT	TC	

2, Map the word sequence to numerical sequence

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
GG	GC	AG	CG	GT	TT	AT	AA	GA	TA	TG	CA	CT	TC	AC	CC

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	pos.
T	C	G	G	A	T	T	C	G	T	A	C	G	G	T	A	C	G	G	A	T	C	
TC	CG	GG	GA	AT	TT	TC	CG	GT	TA	AC	CG	GG	GT	TA	AC	CG	GG	GA	AT	TC		
14	4	1	9	7	6	14	4	5	10	15	4	1	5	10	15	4	1	9	7	14		code

Dumas-Ninio look-up table (example of k=2)

3, Construct T/M-matrix to record the locations of all words in one scan

pos.	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22
	T	C	G	G	A	T	T	C	G	T	A	C	G	G	T	A	C	G	G	A	T	C
	TC	CG	GG	GA	AT	TT	TC	CG	GT	TA	AC	CG	GG	GT	TA	AC	CG	GG	GA	AT	TC	
code	14	4	1	9	7	6	14	4	5	10	15	4	1	5	10	15	4	1	9	7	14	

T-matrix

code	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16						
pos.	3			2		6	5		4					1								
														7								

M-matrix

pos.	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	
pos.							1															

Dumas-Ninio look-up table (example of k=2)

3, Construct T/M-matrix to record the location of all words in one scan

pos.	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22
	T	C	G	G	A	T	T	C	G	T	A	C	G	G	T	A	C	G	G	A	T	C
	TC	CG	GG	GA	AT	TT	TC	CG	GT	TA	AC	CG	GG	GT	TA	AC	CG	GG	GA	AT	TC	
code	14	4	1	9	7	6	14	4	5	10	15	4	1	5	10	15	4	1	9	7	14	

T-matrix

code	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16						
pos.	3			2		6	5		4					1								
				8										7								

M-matrix

pos.	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	
pos.							1	2														

Dumas-Ninio look-up table (example of k=2)

3, Construct T/M-matrix to record the location of all words in one scan

pos.	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22
	T	C	G	G	A	T	T	C	G	T	A	C	G	G	T	A	C	G	G	A	T	C
	TC	CG	GG	GA	AT	TT	TC	CG	GT	TA	AC	CG	GG	GT	TA	AC	CG	GG	GA	AT	TC	
code	14	4	1	9	7	6	14	4	5	10	15	4	1	5	10	15	4	1	9	7	14	

T-matrix

code	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16						
pos.	3			2		6	5		4					1								
				8	9					10				7	11							
				12																		

M-matrix

pos.	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	
pos.							1	2				8										

Dumas-Ninio look-up table (example of k=2)

3, Construct T/M-matrix to record the location of all words in one scan

pos.	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22
	T	C	G	G	A	T	T	C	G	T	A	C	G	G	T	A	C	G	G	A	T	C
	TC	CG	GG	GA	AT	TT	TC	CG	GT	TA	AC	CG	GG	GT	TA	AC	CG	GG	GA	AT	TC	
code	14	4	1	9	7	6	14	4	5	10	15	4	1	5	10	15	4	1	9	7	14	

T-matrix

code	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16						
pos.	3			2		6	5		4					1								
				8	9					10				7	11							
	13			12																		

M-matrix

pos.	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	
pos.							1	2				8	3									

Dumas-Ninio look-up table (example of k=2)

3, Construct T/M-matrix to record the location of all words in one scan

pos.	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22
	T	C	G	G	A	T	T	C	G	T	A	C	G	G	T	A	C	G	G	A	T	C
	TC	CG	GG	GA	AT	TT	TC	CG	GT	TA	AC	CG	GG	GT	TA	AC	CG	GG	GA	AT	TC	
code	14	4	1	9	7	6	14	4	5	10	15	4	1	5	10	15	4	1	9	7	14	

T-matrix

code	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16						
pos.	3			2		6	5		4					1								
				8	9					10				7	11							
	13			12	14					15					16							
	18			17			20		19					21								

M-matrix

pos.	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	
pos.							1	2				8	3	9	10	11	12	13	4	5	7	

Dumas-Ninio look-up table (example of k=2)

pos.	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22
	T	C	G	G	A	T	T	C	G	T	A	C	G	G	T	A	C	G	G	A	T	C
	TC	CG	GG	GA	AT	TT	TC	CG	GT	TA	AC	CG	GG	GT	TA	AC	CG	GG	GA	AT	TC	
code	14	4	1	9	7	6	14	4	5	10	15	4	1	5	10	15	4	1	9	7	14	

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
GG	GC	AG	CG	GT	TT	AT	AA	GA	TA	TG	CA	CT	TC	AC	CC

T-matrix

code	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16						
pos.	3			2		6	5		4					1								
				8	9					10				7	11							
	13			12	14					15					16							
	18			17			20		19					21								
pos.	18	0	0	17	14	6	20	0	19	15	0	0	0	21	16	0						

M-matrix

pos.	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
pos.							1	2				8	3	9	10	11	12	13	4	5	7

Using lookup table, we can quickly trace back identity and location of words.

'GG' appears at positions: 18, 13,3

'TC' appears at positions: 21, 7,1

etc

Dumas-Ninio look-up table

When we make an alignment, we only need to trace a limited number paths to find the matched words, i.e. from T to M

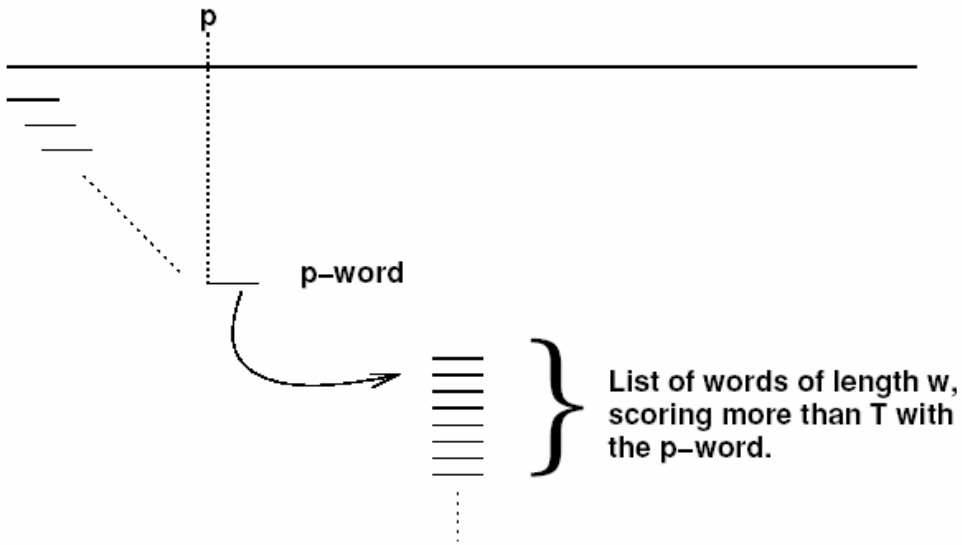
Advantage: fast $O(2N)$ vs. $O(N^2)$

Defect: only identical residue pairs can be aligned

BLAST

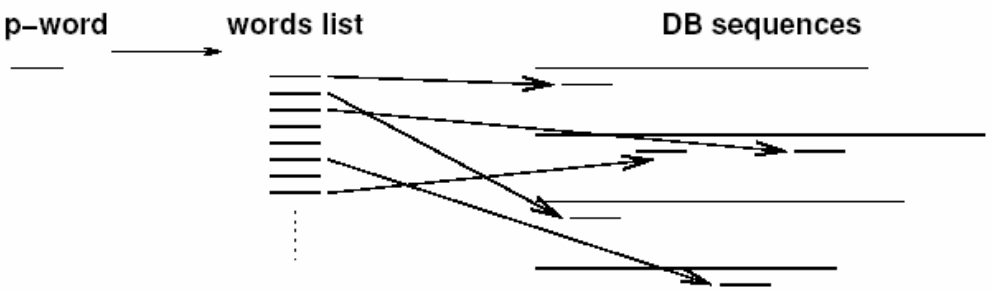
First step:

For each position p of the query, find the list or words of length w scoring more than T when paired with the word starting at p:



Second step:

For each words list, identify all exact matches with DB sequences:



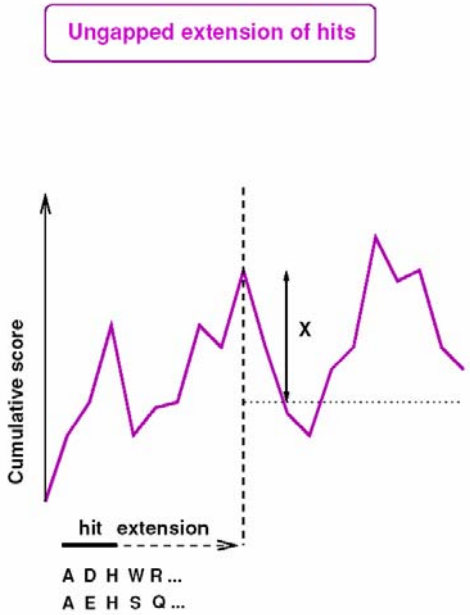
BLAST

Third step:

ungapped extension

For each word match («hit»), extend ungapped alignment in both directions. Stop when S decreases by more than X from the highest value reached by S.

HSP = High Scoring Segment Pair

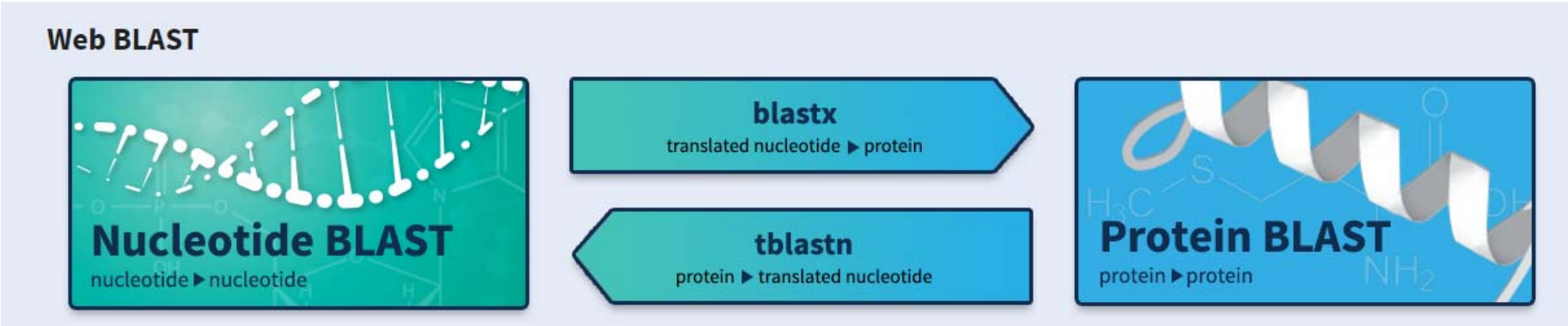


BLAST

https://blast.ncbi.nlm.nih.gov/Blast.cgi

Different types of BLAST programs:

- Nucleotide BLAST (blastn)
- Protein BLAST (blastp)
- Position-Specific Iterative BLAST (PSI-BLAST)
- …



https://ftp.ncbi.nlm.nih.gov/blast/executables/LATEST/

Significance of alignment in BLAST: E-value

For any alignment, we can have an alignment score (S). The score itself does not tell how significant it is.

Definition: The ***E*-value** of an alignment with score S is the ***expected number*** of alignments to be found with score ≥S in two random sequences (of same lengths and letter compositions).

$$\textbf{E-value} = \boldsymbol{Kmne^{-\lambda S}}$$

This eq is an approx. Exact solution is an open quiz

K and *λ* represent natural scales for the search space and the scoring system respectively. *m* and *n* are the sizes of the query and template sequences.

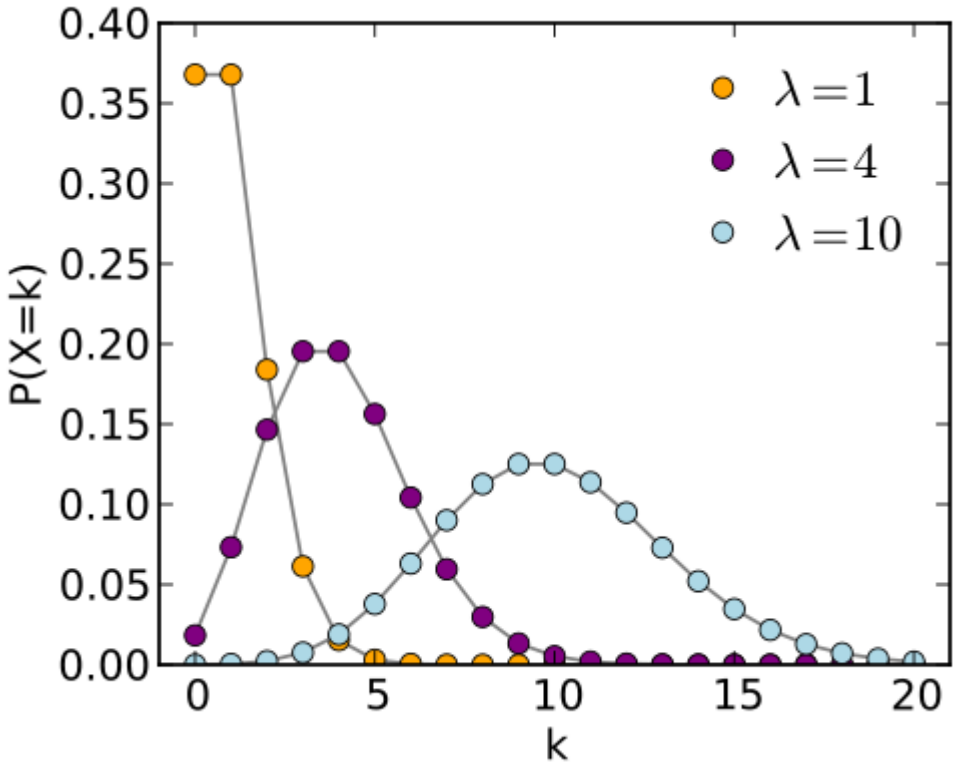
In general, the typical threshold for a good E-value from a BLAST search is 0.001 or lower. An alignment of low E-value means that that alignment is highly unique, and not due to error.

For a proof of the equation, see Karlin & Altschul, PNAS (1990), 87, 2264; PNAS (1993), 90, 5873.

Poisson distribution

If the ***expected number*** of an event to occur is λ, the probability that there are exactly *k* occurrences (*k* = 0, 1, 2, ...) is equal to

$$p(X = k) = \frac{\lambda^k e^{-\lambda}}{k!}$$



Significance of alignment in BLAST: P-value

Definition: The P-value of an alignment with score S is the *likelihood* that two random sequences will have (at least one) alignments with score ≥ S.

Relation between P-value and E-value

If E(S) is the expected number of alignment with score ≥S, the likelihood of getting exactly *k* such (independent) alignments is

$$\textbf{P-value} = \frac{E(S)^k e^{-E(S)}}{k!}$$

$$p(X = k) = \frac{\lambda^k e^{-\lambda}}{k!}$$

- Likelihood of getting 0 such alignment: $e^{-E(S)}$

- Likelihood of getting at least one such alignment: $1 - e^{-E(S)}$

Content

1. Bioinformatics databases
2. Sequence alignment and database searching
3. Phylogenic tree and multiple sequence alignment
4. Protein structure alignment
5. Protein secondary structure prediction
6. Protein tertiary structure prediction
7. Protein function prediction

Next class

- **Neighbor-joining method (for constructing phylogenetic tree)** N. Saitou and M. Nei. The neighbor-joining Method: A new method for reconstructing hylogenetic tree. Mol Biol Evol. (1987) 4: 406-425.
- **UPGMA**: https://en.wikipedia.org/wiki/UPGMA
- **PSI-BLAST (The most often-used algorithm for sequence-profile alignment)** S. F. Altschul et al. Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. (1997) 25, 3389-3402
- **Hidden Markov Model (for multiple sequence alignment)** Haussler, D., Krogh, A., Mian, I. S., & Sjölander, K. (1993). Protein modeling using hidden Markov models: Analysis of globins. In: Proceedings of the Hawaii International Conference on System Sciences volume 1 pp. 792-802.
- **Sequence profile** Gribskov, Mclanchlan, Eisenberg. Profile analysis: Detection of distantly related proteins. PNAS (1987) 84, 4355-58
- **Henikoff weight** Steven Henikoff and Jorja G. Henikoff, Position-based sequence weights, Journal of Molecular Biology. Volume 243, Issue 4, 4 November 1994, Pages 574-578

Next class

- **Profile-profile alignments:**

Anna R. Panchenko. Finding weak similarities between proteins by sequence profile comparison. Nucleic Acids Research, 2003, Vol. 31, No. 2 683.

Edgar & Sjolander, A comparison of scoring functions for protein sequence profile alignment. Bioinformatics (2004) 20, 1301-8

G Wang, R. Dunbrack JR. Scoring profile-to-profile sequence alignments. Protein Sci. 13:1612-1626, 2004