Lecture outline

- FASTA Algorithm
- Statistical Significance of Sequence Comparison Results
  - Probability of matching runs
  - Karin-Altschul statistics
  - Extreme value distribution

FASTA

- Derived from logic of the dot plot
  - compute best diagonals from all frames of alignment
- Word method looks for exact matches between words in query and test sequence
  - construct word position tables
  - DNA words are usually 6 bases
  - protein words are 1 or 2 amino acids
  - only searches for diagonals in region of word matches = faster searching

Steps of FASTA

1. Find k-tups in the two sequences (k=1-2 for proteins, 4-6 for DNA sequences)
2. Create a table of positions for those k-tups

The offset table

```
position 1 2 3 4 5 6 7 8 9 10 11
proteinA n c s p t a . . . . .
proteinB . . . . . a c s p r k
```

	position in		offset
amino acid	protein A	protein B	pos A - posB
a	6	6	0
c	2	7	-5
k	-	11	
n	1	-	
p	4	9	-5
r	-	10	
s	3	8	-5
t	5	-	

Note the common offset for the 3 amino acids c,s and p

A possible alignment is thus quickly found -

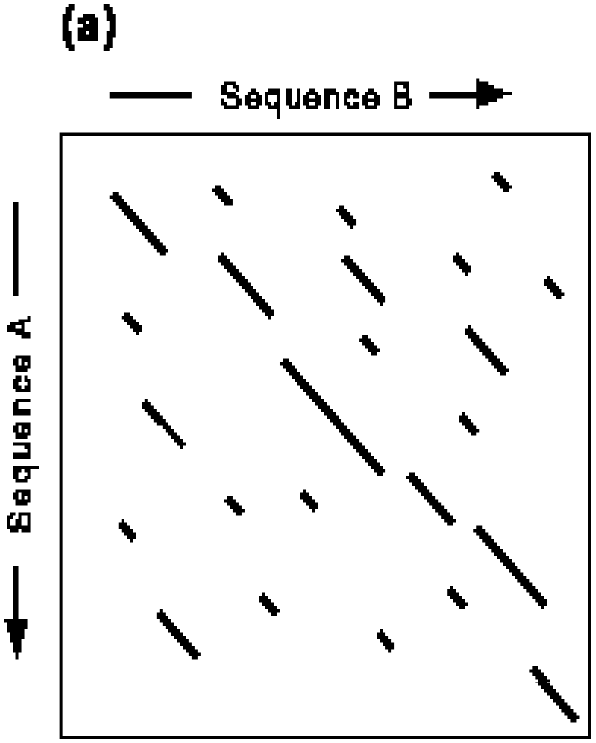
```
protein 1 n c s p t a
            | | |
protein 2 a c s p r k
```

FASTA

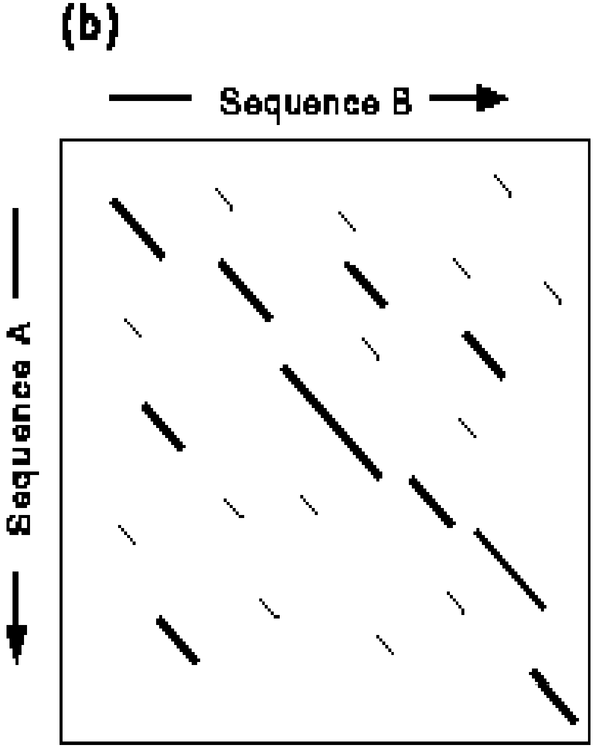
3. Select top 10 scoring “local diagonals” with matches and mismatches but no gaps.

4. Rescan top 10 diagonals (representing alignments), score with PAM250 (proteins) or DNA scoring matrix. Trim off the ends of the regions to achieve highest scores.

FASTA Algorithm



Find runs of identitical words



Re-score using PAM matrix

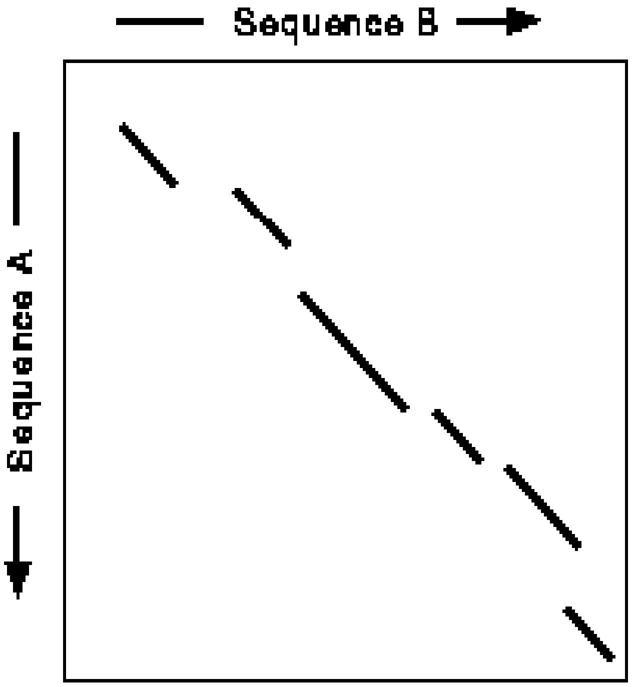
Keep top scoring segments

FASTA

5. After finding the best initial region, FASTA performs a DP global alignment centered on the best initial region.

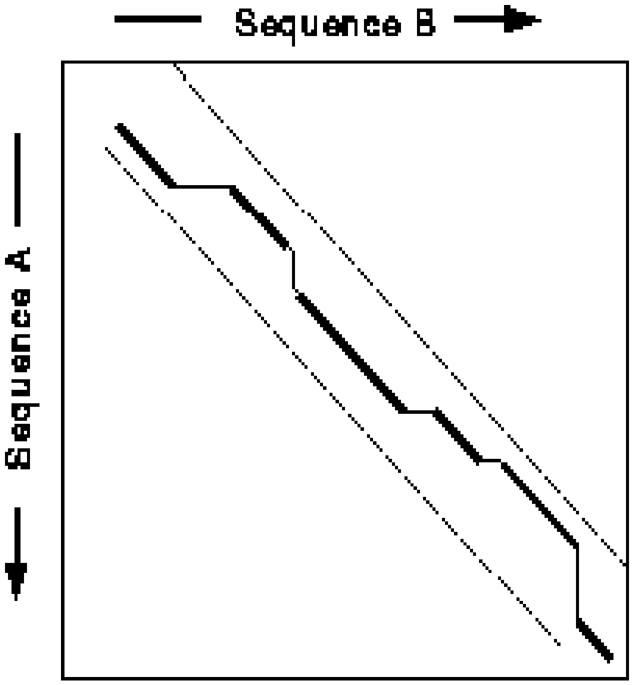
FASTA Alignments

(c)



Join segments using gaps, eliminate other segments

(d)



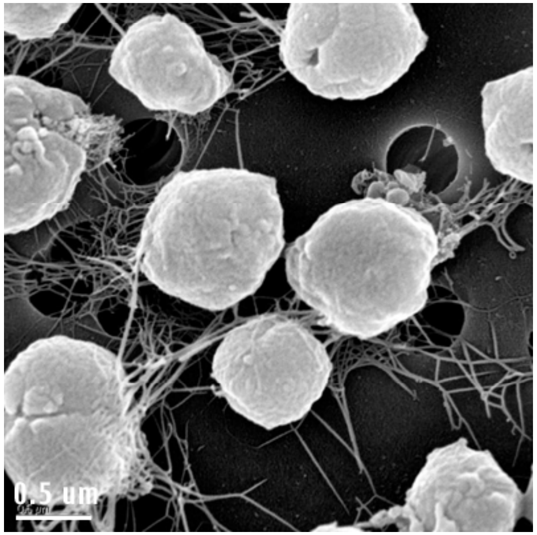
Use dynamic programming to create an optimal alignment

History of sequence searching

- 1970: NW
- 1981: SW
- 1985: FASTA
- 1990: BLAST
- 1997: BLAST2

The purpose of sequence alignment

- Homology
- Function identification
  - about 70% of the genes of *M. jannaschii* were assigned a function using sequence similarity (1997)



Similarity

- How much similar do the sequences have to be to infer homology?

- Two possibilities when similarity is detected:
  - The similarity is by chance
  - They evolved from a common ancestor – hence, have similar functions

Measures of similarity

- Percent identity:
  - 40% similar, 70% similar
  - problems with percent identity?
- Scoring matrices
  - matching of some amino acids may be more significant than matching of other amino acids
  - PAM matrix in 1970, BLOSUM in 1992
  - problems?

Statistical Significance

- Goal: to provide a universal measure for inferring homology
  - How different is the result from a random match, or a match between unrelated requences?
  - Given a set of sequences *not related* to the query (or a set of random sequences), what is the probability of finding a match with the same alignment score by chance?
- Different statistical measures
  - p-value
  - E-value
  - z-score

Statistical significance measures

- *p-value*: the probability that at least one sequence will produce the same score by chance

- *E-value*: expected number of sequences that will produce same or better score by chance

- *z-score*: measures how much standard deviations above the mean of the score distribution

How to compute statistical significance?

- Significance of a match-run
  - Erdös-Renyí
- Significance of local alignments without gaps
  - Karlin-Altschul statistics
  - Scoring matrices revisited
- Significance of local alignments with gaps
- Significance of global alignments

Analysis of coin tosses

●		●	●			●	●	●	●	●			

- Let black circles indicate heads
- Let p be the probability of a “head”
  - For a “fair” coin, p = 0.5
- Probability of 5 heads in a row is (1/2)^5=0.031
- The expected number of times that 5H occurs in above 14 coin tosses is 10*0.031 = 0.31

Analysis of coin tosses

- The expected number of a length *l* run of heads in *n* tosses.

$$E(l) \cong np^{l}$$

- What is the expected length *R* of the longest match in *n* tosses?

$$1 = np^{R} \longrightarrow R = \log_{1/p}(n)$$

Analysis of coin tosses

- (Erdös-Rényi) If there are n throws, then the expected length R of the longest run of heads is

$$R = \log_{1/\mathrm{p}}(n)$$

Example

- Example: Suppose n = 20 for a “fair” coin

$$R=\log_2(20)=4.32$$

  - In other words: in 20 coin tosses we expect a run of heads of length 4.32, once.

- Trick is how to model DNA (or amino acid) sequence alignments as coin tosses.

Analysis of an alignment

	H	I	K	T	Q	S	N	A	I	L
H	●									
E										
S						●				
R										
A								●		
I		●							●	
Q					●					
V										

- Probability of an individual match $p = 0.05$
- Expected number of matches: 10x8x0.05 = 4
- Expected number of two successive matches

  $\cong$10x8x0.05x0.05 = 0.2

Matching runs in sequence alignments

- Consider two sequences $a_{1..m}$ and $b_{1..n}$
- If the probability of occurrence for every symbol is p, then a match of a residue a_i with b_j is p, and a match of length l from a_i,b_j to a$_{i+l\text{-}1}$,$b_{j+l\text{-}1}$ is p^l.
- The head-run problem of coin tosses corresponds to the longest run of matches along the diagonals

Matching runs in sequence alignments

- There are m-l+1 x n-l+1 places where the match could start

$$E(l) \cong mnp^l$$

- The expected length of the longest match can be approximated as

$$R=\log_{1/p}(mn)$$

where m and n are the lengths of the two sequences.

Matching runs in sequence alignments

- So suppose $m = n = 10$ and we're looking at DNA sequences

$$R = \log_4(100) = 3.32$$

- This analysis makes assumptions about the base composition (uniform) and no gaps, but it's a good estimate.

Statistics for matching runs

- Statistics of matching runs:

$$E(l) \cong mnp^{l}$$

- Length versus score?
  - Consider all mismatches receive a negative score of -∞ and $a_i b_j$ match receives a positive score of $s_{i,j}$.
- What is the expected number of matching runs with a score x or higher?

$$E(S >= x) \propto mnp^{x}$$

  - Using this theory of matching runs, Karlin and Altschul developed a theory for statistics of local alignments without gaps (extended this theory to allow for mismatches).

Statistics of local alignments without gaps

- A scoring matrix which satisfy the following constraint:
  - The expected score of a single match obtained by a scoring matrix should be negative.

$$E(s_{i,j}) = \sum_{i,j} p_i p_j s_{i,j} < 0$$

  - Otherwise?
    - Arbitrarily long random sequences will get higher scores just because they are long, not because there's a significant match.
- If this requirement is met then the expected number of alignments with score *x* or higher is given by:

$$E(S \geq x) = Kmne^{-\lambda x}$$

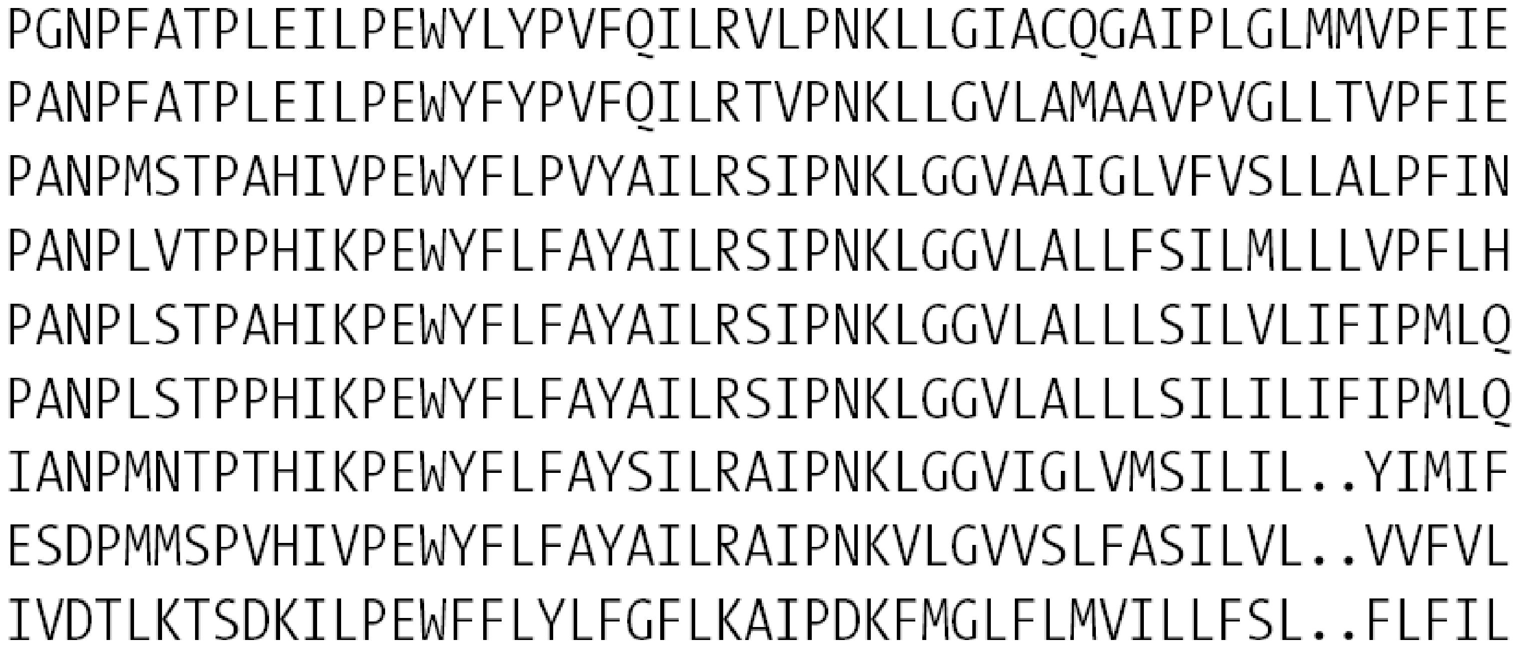
Statistics of local alignments without gaps

$$E(S \geq x) = Kmne^{-\lambda x}$$

- $K < 1$ is a proportionality constant that corrects the *mn* “space factor” for the fact that there are not really *mn* independent places that could have produced score $S \geq x$.
- K has little effect on the statistical significance of a similarity score
- λ is closely related to the scoring matrix used and it takes into account that the scoring matrices do not contain actual probabilities of co-occurence, but instead a scaled version of those values. To understand how λ is computed, we have to look at the construction of scoring matrices.

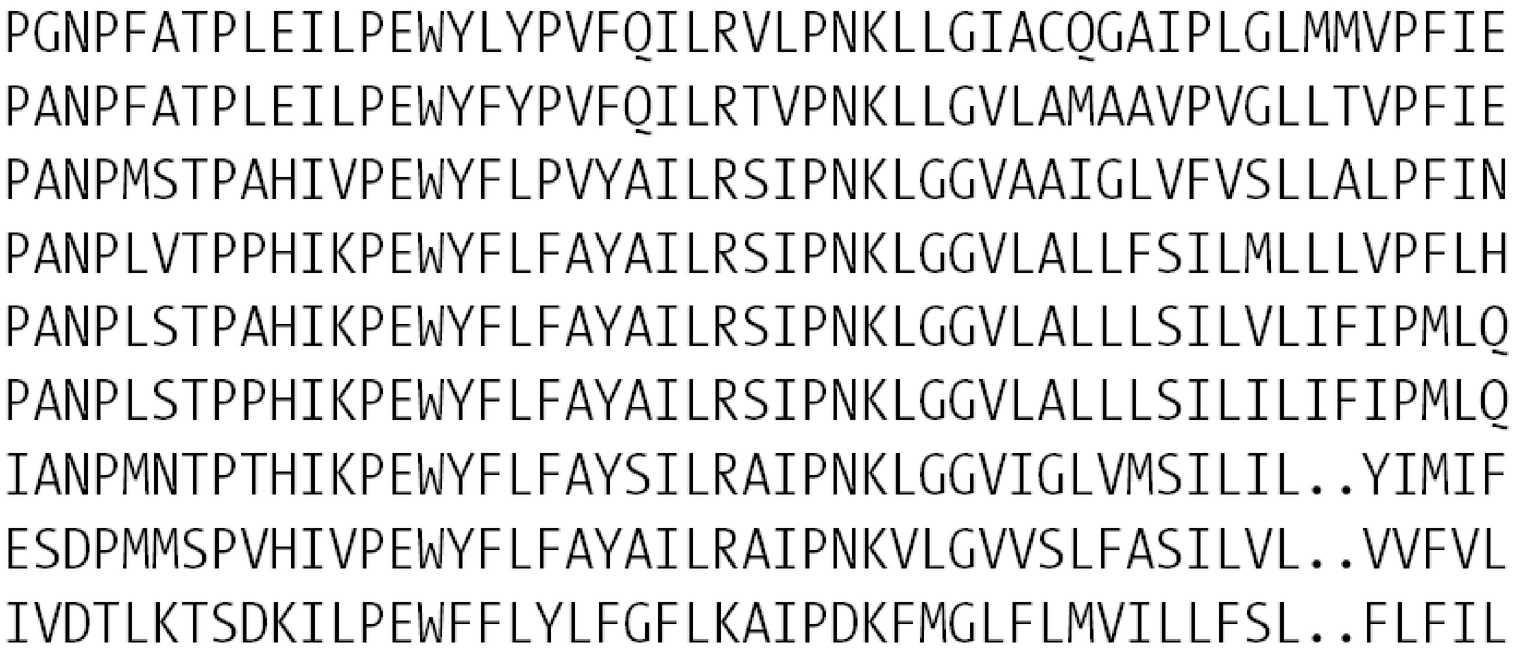
Scoring Matrices

- In 1970s there were few protein sequences available. Dayhoff used a limited set of families of protein sequences multiply aligned to infer mutation likelihoods.



```
PGNPFATPLEILPEWYLYPVFQILRVLPNKLLGIACQGAIPLGLMMVPFIE
PANPFATPLEILPEWYFYPVFQILRTVPNKLLGVLAMAAVPVGLLTVPFIE
PANPMSTPAHIVPEWYFLPVYAILRSIPNKLGGVAAIGLVFVSLLALPFIN
PANPLVTPPHIKPEWYFLFAYAILRSIPNKLGGVLALLFSILMLLLVPFLH
PANPLSTPAHIKPEWYFLFAYAILRSIPNKLGGVLALLLSILVLIFIPMLQ
PANPLSTPPHIKPEWYFLFAYAILRSIPNKLGGVLALLLSILILIFIPMLQ
IANPMNTPTHIKPEWYFLFAYSILRAIPNKLGGVIGLVMSILIL..YIMIF
ESDPMMSPVHIVPEWYFLFAYAILRAIPNKVLGVVSLFASILVL..VVFVL
IVDTLKTSDKILPEWFFLYLFGFLKAIPDKFMGLFLMVILLFSL..FLFIL
```

Scoring Matrices



```
PGNPFATPLEILPEWYLYPVFQILRVLPNKLLGIACQGAIPLGLMMVPFIE
PANPFATPLEILPEWYFYPVFQILRTVPNKLLGVLAMAAVPVGLLTVPFIE
PANPMSTPAHIVPEWYFLPVYAILRSIPNKLGGVAAIGLVFVSLLALPFIN
PANPLVTPPHIKPEWYFLFAYAILRSIPNKLGGVLALLFSILMLLLVPFLH
PANPLSTPAHIKPEWYFLFAYAILRSIPNKLGGVLALLLSILVLIFIPMLQ
PANPLSTPPHIKPEWYFLFAYAILRSIPNKLGGVLALLLSILILIFIPMLQ
IANPMNTPTHIKPEWYFLFAYSILRAIPNKLGGVIGLVMSILIL..YIMIF
ESDPMMSPVHIVPEWYFLFAYAILRAIPNKVLGVVSLFASILVL..VVFVL
IVDTLKTSDKILPEWFFLYLFGFLKAIPDKFMGLFLMVILLFSL..FLFIL
```

- Dayhoff represented the similarity of amino acids as a log odds ratio:

$$s_{ij} = \log(q_{ij} / p_i p_j)$$

where q_{ij} is the observed frequency of co-occurrence, and p_i, p_j are the individual frequencies.

Example

- If M occurs in the sequences with 0.01 frequency and L occurs with 0.1 frequency. By random pairing, you expect 0.001 amino acid pairs to be M-L. If the observed frequency of M-L is actually 0.003, score of matching M-L will be
  - $\log_2(3)=1.585$ bits or $\log_e(3) = \ln(3) = 1.1$ nats
- Since, scoring matrices are usually provided as integer matrices, these values are scaled by a constant factor. λ is approximately the inverse of the original scaling factor.

How to compute λ

- Recall that:

$$\lambda s_{ij} = \log(q_{ij} / p_i p_j)$$

$$\Rightarrow q_{ij} = p_i p_j e^{\lambda s_{ij}}$$

and: $\sum_{i=1}^{n} \sum_{j=1}^{i} q_{ij} = 1$ Sum of observed frequencies is 1.

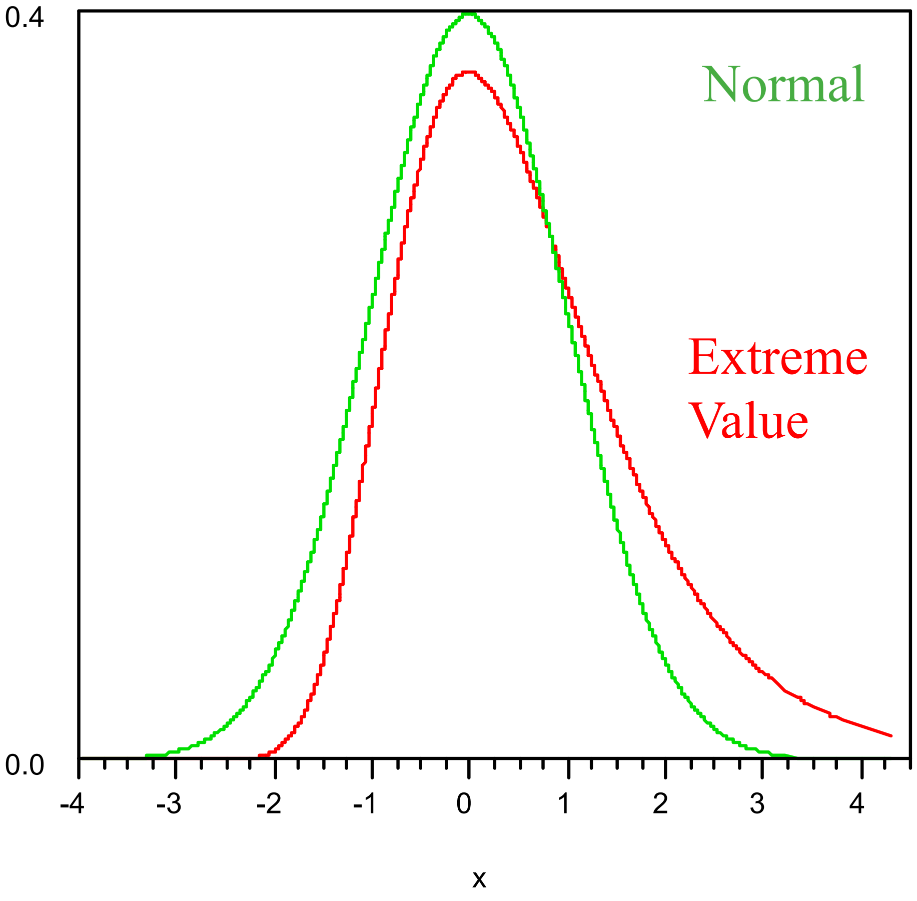
$$\Rightarrow \sum_{i=1}^{n} \sum_{j=1}^{i} p_i p_j e^{\lambda s_{ij}} = 1$$

Given the frequencies of individual amino acids and the scores in the matrix, λ can be estimated.

Extreme value distribution

- Consider an experiment that obtains the maximum value of locally aligning a random string with query string (without gaps). Repeat with another random string and so on. Plot the distribution of these maximum values.
- The resulting distribution is an extreme value distribution, called a *Gumbel distribution*.

Normal vs. Extreme Value Distribution



Normal distribution:

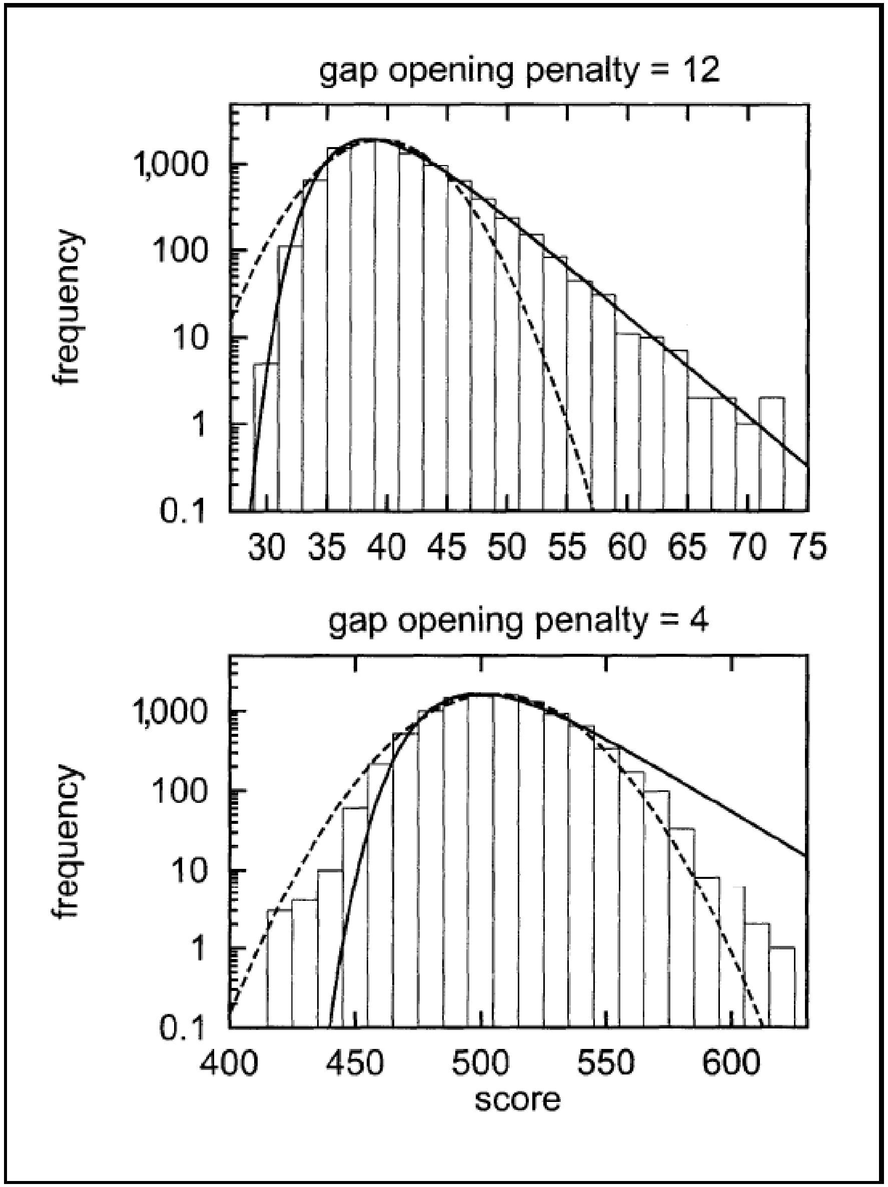
$y = (1/\sqrt{2\pi})e^{-x^2/2}$

Extreme value distribution:

$y = e^{-x - e^{-x}}$

Local alignments with gaps

- The EVD distribution is not always observed. Theory of local alignments with gaps is not well studied as in without gaps. Mostly empirical results. For example, BLAST allows only a certain range of gap penalties.



BLAST statistics

- Pre-computed λ and K values for different scoring matrices and gap penalties are used for faster computation.
- Raw score is converted to bit score:

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

- E-value is computed using

$$E = sss \cdot 2^{-S_{bit}}$$

$$sss = (m - L)(n - N \cdot L)$$

- m is query size, n is database size and L is the typical length of maximal scoring alignment.

FASTA Statistics

- FASTA tries to estimate the probability distribution of alignments for every query.
- For any query sequence, a large collection of scores is gathered during the search of the database.
- They estimate the parameters of the EVD distribution based on the histogram of scores.
- Advantages:
  - reliable statistics for different parameters
    - different databases, different gap penalties, different scoring matrices, queries with different amino acid compositions.

Statistical significance another example

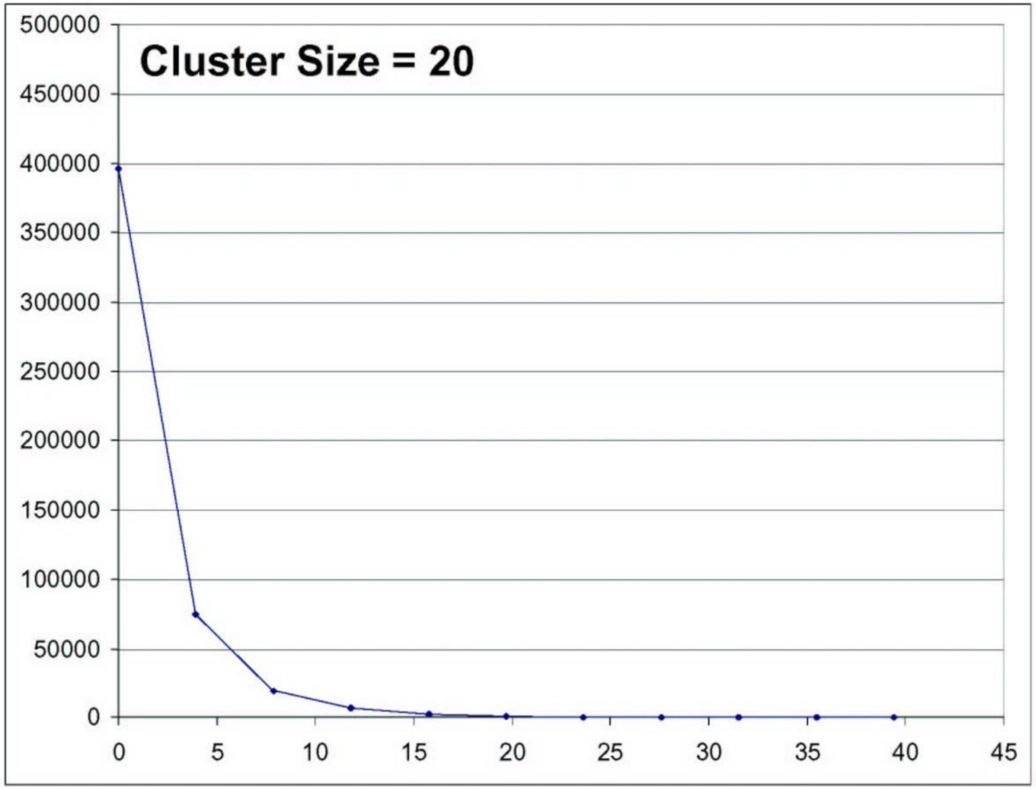
- Suppose, we have a huge graph with weighted edges and we want to find strongly connected clusters of nodes.
- Suppose, an algorithm for this task is given.
- The algorithms gives you the best hundred clusters in this graph.
- How do you define best?
- Cluster size?
- Total weight of edges?

Statistical significance

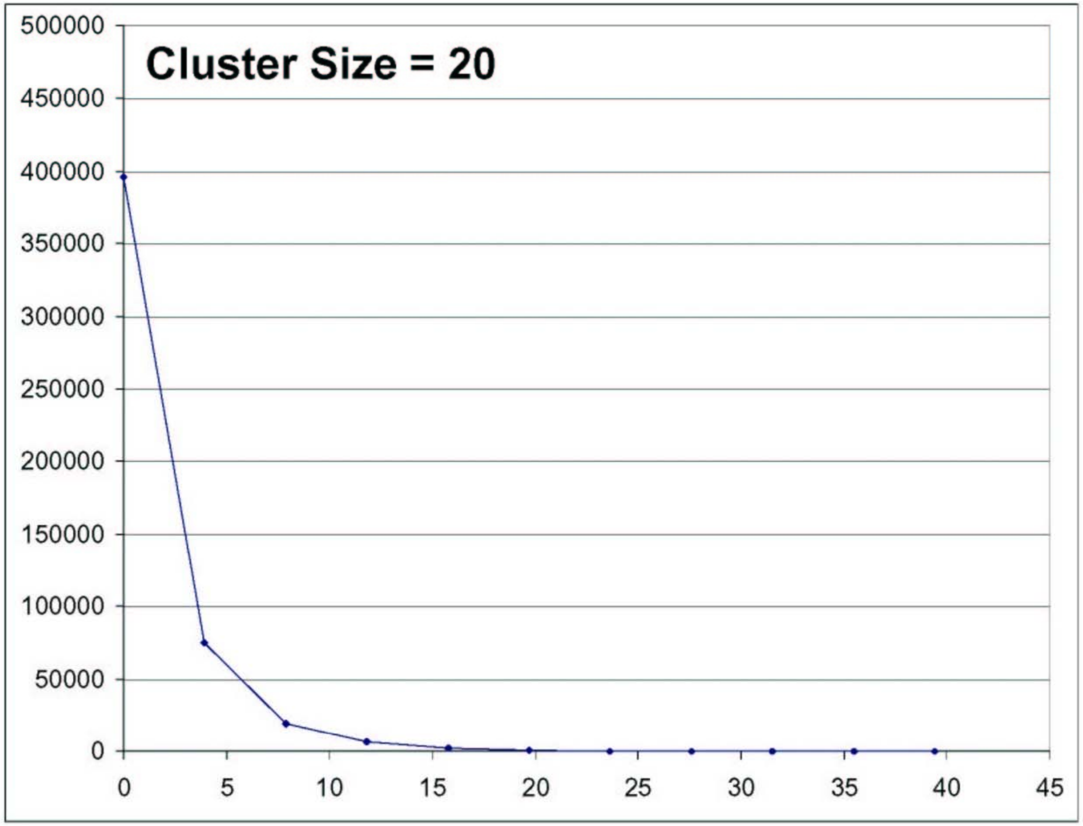
- How different is a found cluster of size N from a random cluster of the same size?
- This measure will enable comparison of clusters of different sizes.

Statistical significance of a cluster

- Use maximum spanning tree weight of a cluster as a quantitative representation of that cluster.
- And see what values random clusters get. (sample many random clusters)



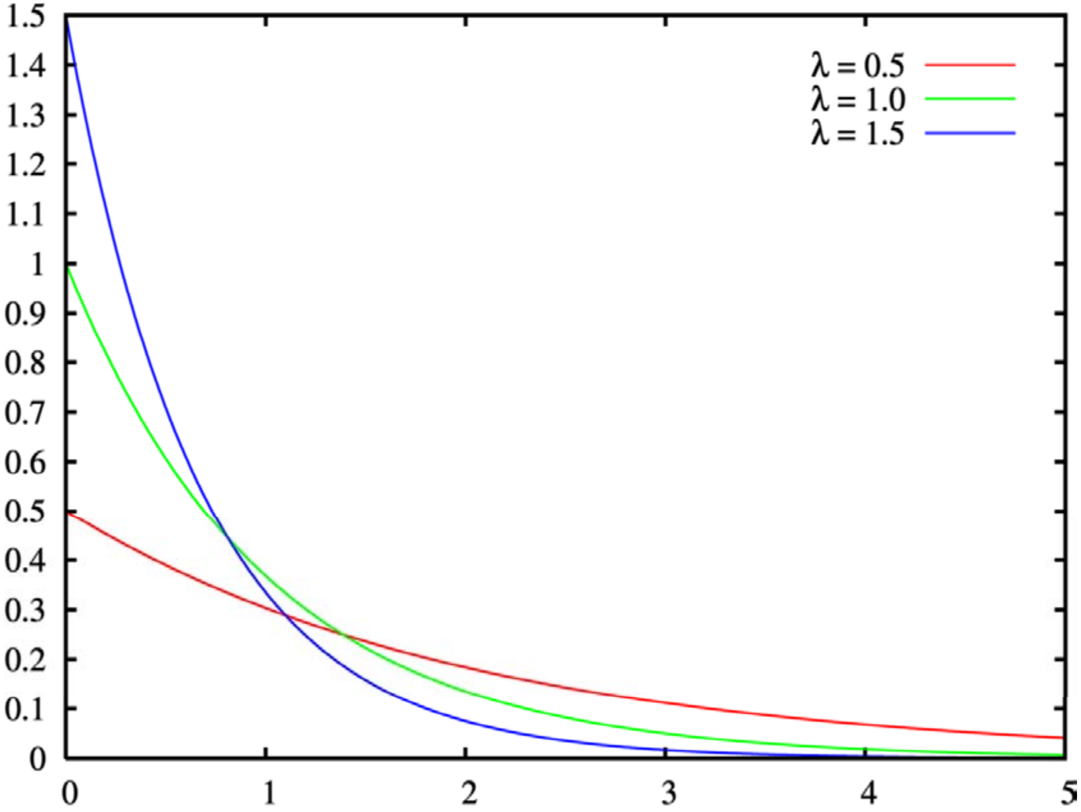
Statistical significance of a cluster



Looks like an exponential decay. We may fit an exponential distribution on this histogram.

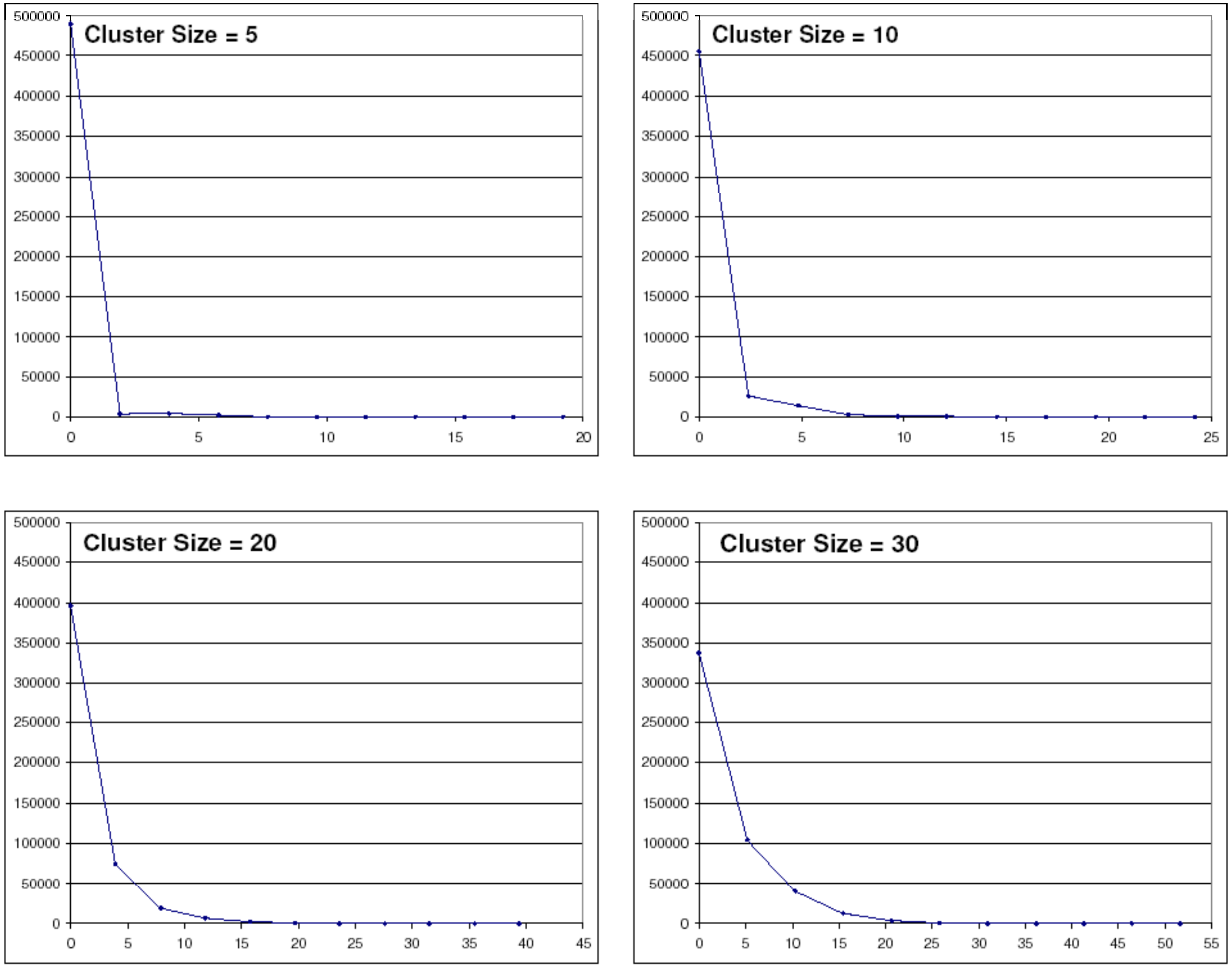
$$y = \lambda e^{-\lambda x}$$

Fitting an exponential



$$y = \lambda e^{-\lambda x}$$

Statistical significance of a cluster



After we fit an exponential distribution, we compute the probability that another random cluster gets a higher score than the score of found cluster.

$$P(x \geq w) = e^{-\lambda_k w}$$

Examples

- $\lambda_5 = 1.7$ for clusters of size 5 and $\lambda_{20} = 0.36$ for clusters of size 20.
- Suppose you have found a cluster of size 5 with weights of its edges sum up to 15 and you have found a cluster of size 20 with weight 45 which one would you prefer?

$$P(x \geq 15) = e^{-\lambda_5 15} = 8.42 \times 10^{-12}$$

$$P(x \geq 45) = e^{-\lambda_{20} 45} = 9.21 \times 10^{-8}$$