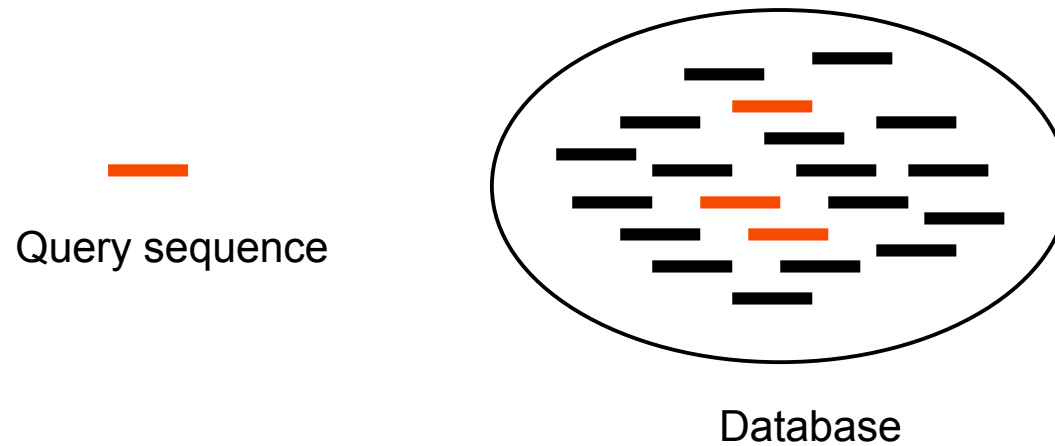

BLAST

Anders Gorm Pedersen
&
Rasmus Wernersson

Database searching

Using pairwise alignments to search
databases for similar sequences



Database searching

Most common use of pairwise sequence alignments is to search databases for related sequences. For instance: find probable function of newly isolated protein by identifying similar proteins with known function.

Most often, ***local*** alignment (“Smith-Waterman”) is used for database searching: you are interested in finding out if ANY domain in your protein looks like something that is known.

Often, full Smith-Waterman is too time-consuming for searching large databases, so heuristic methods are used (fasta, BLAST).

Database searching: heuristic search algorithms

FASTA (Pearson 1995)

Uses heuristics to avoid calculating the full dynamic programming matrix

Speed up searches by an order of magnitude compared to full Smith-Waterman

The statistical side of FASTA is still stronger than BLAST

BLAST (Altschul 1990, 1997)

Uses rapid word lookup methods to completely skip most of the database entries

Extremely fast

One order of magnitude faster than FASTA

Two orders of magnitude faster than Smith-Waterman

Almost as sensitive as FASTA

BLAST flavors

BLASTN

Nucleotide query sequence
Nucleotide database

BLASTP

Protein query sequence
Protein database

BLASTX

Nucleotide query sequence
Protein database
Compares all six reading frames
with the database

TBLASTN

Protein query sequence
Nucleotide database
"On the fly" six frame translation of
database

TBLASTX

Nucleotide query sequence
Nucleotide database
Compares all reading frames of
query with all reading frames of
the database

Searching on the web: BLAST at NCBI

Very fast computer dedicated to running BLAST searches

Many databases that are always up to date (e.g. NR and Human Genome)

Nice simple web interface

But you still need knowledge about BLAST to use it properly

The screenshot displays the NCBI BLAST web interface. The browser address bar shows the URL: <http://www.ncbi.nlm.nih.gov/blast/Blast.cgi?PAGE=Proteins&PROGRAM=blastp&>. The page title is "Protein BLAST: search protein databases using a protein query". The interface includes a navigation menu with "Home", "Recent Results", "Saved Strategies", and "Help". A "My NCBI" link is also present. The main content area is titled "Enter Query Sequence" and contains a large text input field for the query, a "Clear" button, and a "Query subrange" section with "From" and "To" input fields. Below this is an "Or, upload file" section with a "Choose File" button and a "Job Title" input field. The "Choose Search Set" section includes a "Database" dropdown menu set to "Non-redundant protein sequences (nr)", an "Organism" input field with a "Optional" label, and an "Entrez Query" input field with a "Optional" label. The "Program Selection" section features radio buttons for "blastp (protein-protein BLAST)", "PSI-BLAST (Position-Specific Iterated BLAST)", and "PHI-BLAST (Pattern Hit Initiated BLAST)", with a "Choose a BLAST algorithm" dropdown. At the bottom, there is a "BLAST" button, a "Search database nr using Blastp (protein-protein BLAST)" button, and a "Show results in a new window" checkbox. A footer contains copyright information and links for "Disclaimer", "Privacy", "Accessibility", "Contact", and "Send feedback on new interface".

When is a database hit significant?

- **Problem:**

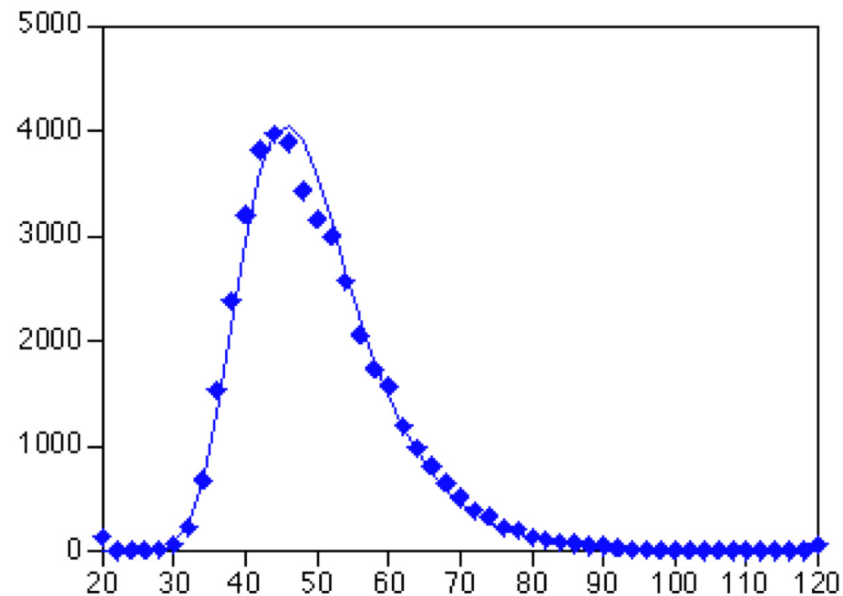
- Even unrelated sequences can be aligned (yielding a low score)
- How do we know if a database hit is meaningful?
- When is an alignment score sufficiently high?

- **Solution:**

- Determine the range of alignment scores you would expect to get for random reasons (i.e., when aligning unrelated sequences).
- Compare actual scores to the distribution of random scores.
- Is the real score much higher than you'd expect by chance?

Random alignment scores follow extreme value distributions

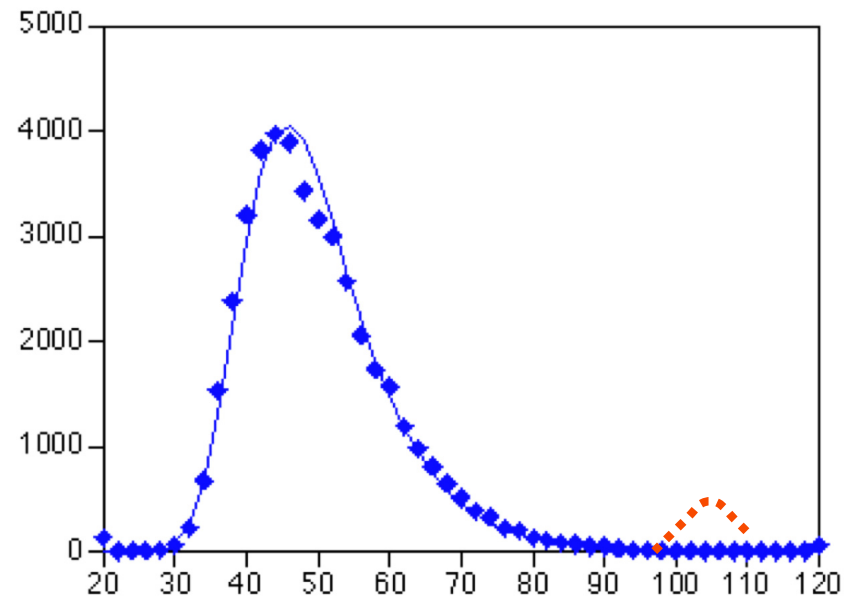
Searching a database of **unrelated** sequences result in scores following an extreme value distribution



The exact shape and location of the distribution depends on the exact nature of the database and the query sequence

Significance of a hit: one possible solution

- (1) Align query sequence to all sequences in database, note scores
- (2) Fit actual scores to a mixture of two sub-distributions: (a) an extreme value distribution and (b) a normal distribution
- (3) Use fitted extreme-value distribution to predict how many random hits to expect for any given score (the “**E-value**”)



Significance of a hit: example

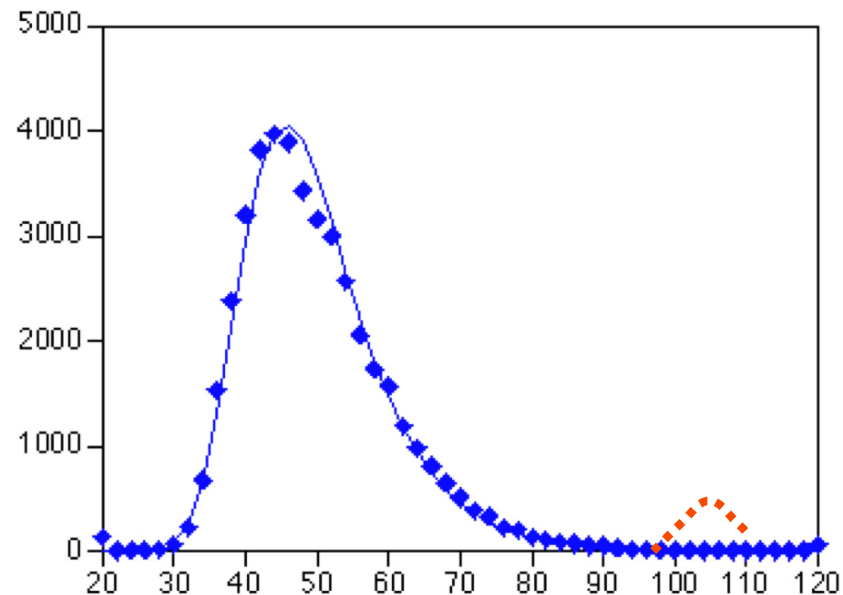
Search against a database of 10,000 sequences.

An extreme-value distribution (blue) is fitted to the distribution of all scores.

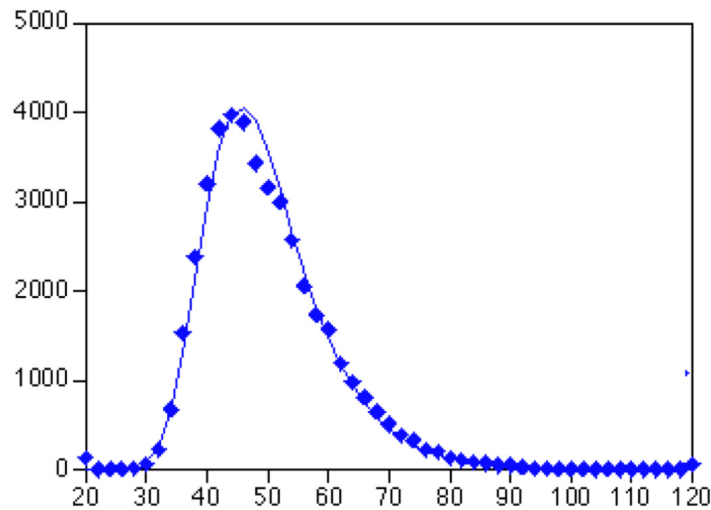
It is found that 99.9% of the blue distribution has a score below 112.

This means that when searching a database of 10,000 sequences you'd expect to get $0.1\% * 10,000 = 10$ hits with a score of 112 or better for random reasons

10 is the E-value of a hit with score 112. **You want E-values well below 1!**



Database searching: E-values in BLAST



BLAST uses precomputed extreme value distributions to calculate E-values from alignment scores

For this reason BLAST only allows certain combinations of substitution matrices and gap penalties

This also means that the fit is based on a different data set than the one you are working on

A word of caution: BLAST tends to overestimate the significance of its matches

E-values from BLAST are fine for identifying sure hits

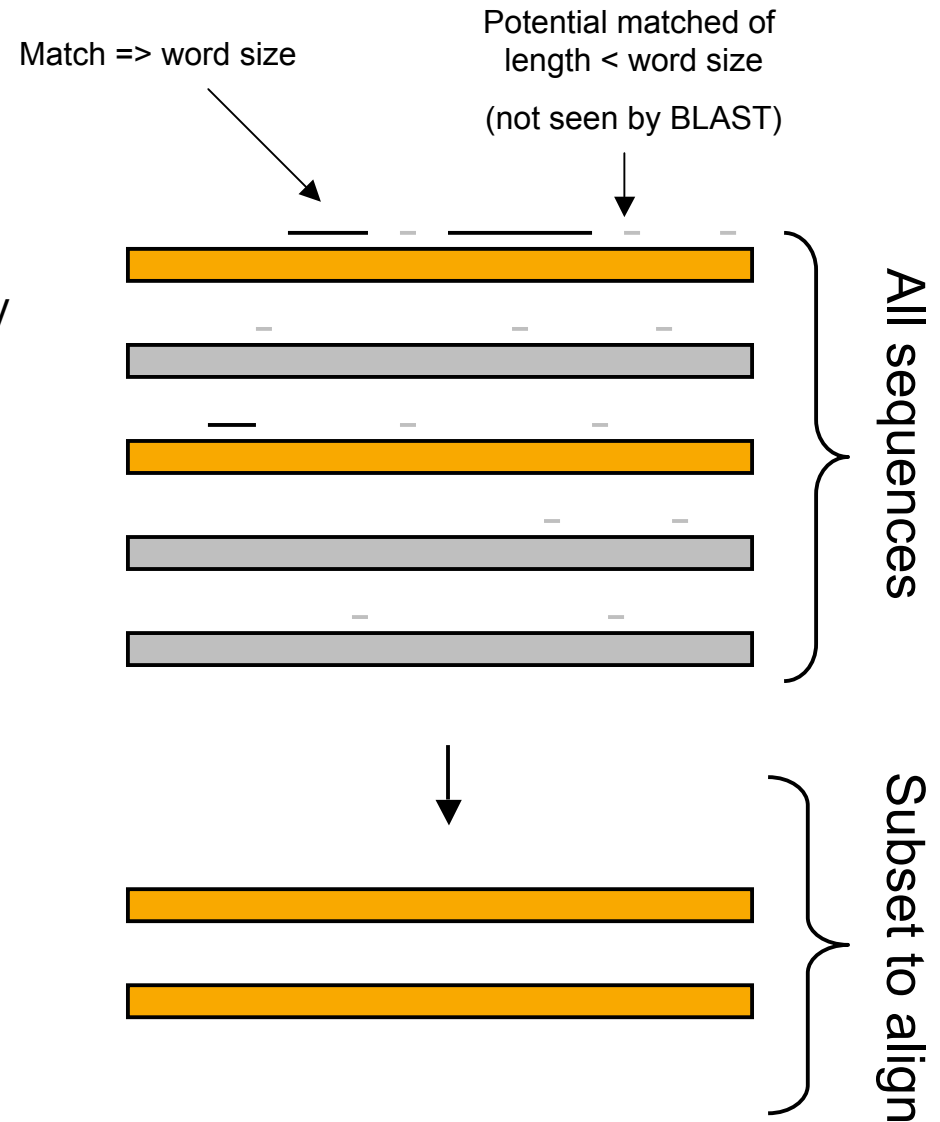
One should be careful using BLAST's E-values to judge if a marginal hit can be trusted (e.g., **you may want to use E-values of 10^{-4} to 10^{-5}**).

BLAST heuristics

- Best possible search:
 - Do full pairwise alignment (Smith-Watermann) between the query sequence and **all** sequences in the database.
 - (“ssearch” does this).
- BLAST speeds up the search by at least two orders of magnitude, by pre-screening the database sequences and only performing the full Dynamic Programming on “*promising*” sequences.
- This is done by indexing all databases sequences in a so-called **suffix-tree** which makes it very fast to search for perfect matching sub-strings.
 - A suffix tree is the quickest possible way (so far) to search for the *longest matching sub-string* between two strings.
- When a BLAST search is run, candidate sequences from the database is picked based on perfect matches to small sub-sequences in the query sequence. (**BLASTN** and **BLASTP** does this differently - more about this in a moment).
 - Full Smith-Waterman is then performed on these sequences.

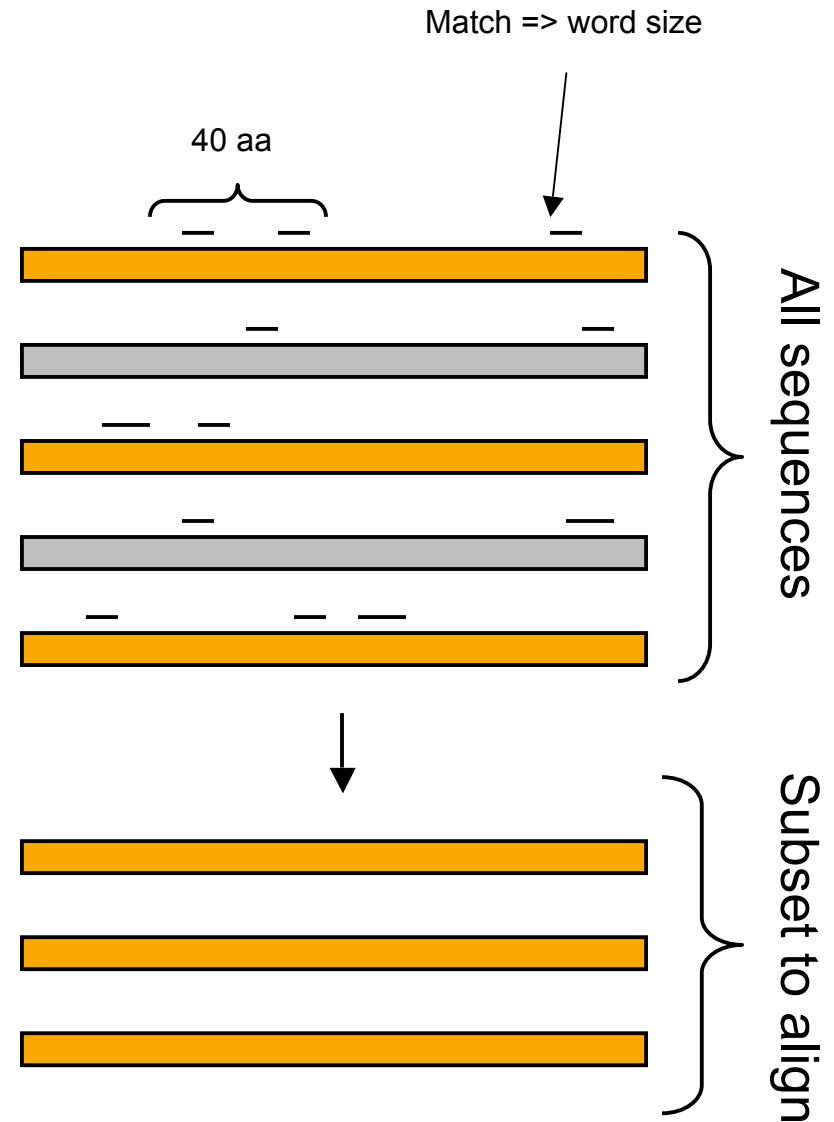
BLASTN

- Alignment matrix:
 - Perfect match: **1**
 - Mismatch: **-3**
- Notice: All mismatches are equally penalized:
 - E.g. A:G == A:C == A:A
 - More advanced models for DNA evolution do exist.
- Heuristics:
 - Perfect match “word” of the size: 7, 11 (default) or 15.



BLASTP

- Alignment matrix:
 - PAM and BLOSUM-series (default: BLOSUM 62)
- Notice: These alignment matrices incorporates knowledge about protein evolution.
- Heuristics:
 - 2 x “Near match” within a windows.
 - Default word length: 3 aa
 - Default window length: 40 aa



BLAST Exercise

