After this lecture, you can...

... discuss speed-sensitivity trade-off in terms of RAM-CPU
... explain the full BLAST algorithm
... list the factors including heuristics that make BLAST fast
... interpret BLAST output/results
... decide which BLAST flavor to use for your similarity search
... explain PSI-BLAST profile searches and their sensitivity
... work with P-values obtained using permutation statistics
... list the factors influencing the E-value (# random hits)
... convert between P-value (probability) and E-value (expected frequency)
Searching a database

- We want to find a query sequence in the database: TGCTGCAGGACAACAGTT
- Could we make sequence alignments between the query and every database sequence?

– Making alignment matrices with all sequences in the database would require a lot of computer operations (CPU time)

An old slide

Random-access memory (RAM)

- A computer program can quickly find elements in the RAM
  - Accessing the contents of a variable
  - Finding a specific sub-sequence (k-mer) in a database, for example all GGAC instances in the sequence below
  - For longer k, you need more RAM to store all possible k-mers
### $k$-mer searches

- Sequences can be divided into shorter subsequences or $k$-mers
  - $k$-mers consist of $k$ nucleotides or amino acids

- We can store an index of all $k$-mers that occur in the database sequences in the RAM
  - All $k$-mers can be stored up to $k \approx 20$ for nucleotides and $k \approx 9$ for amino acids (but probably more since it is unlikely that all $k$-mers exist in the database sequences)

- If we split a query sequence into $k$-mers of the same length, we can rapidly identify all the database sequences containing them

---

### Searching a database

- We want to find a query sequence in the database: $\textit{TGCTGCAGGACAACAGTT}$
- Could we make sequence alignments between the query and every database sequence?
  - Making alignment matrices with all sequences in the database would require a lot of computer operations (CPU time)

- Could we store all possible $k$-mers in the database sequences in the RAM and find the query by using indexing?
  - Depending on the length of $k$ this would require a lot of RAM
  - This would limit the search to exact matches
Transcriptomics

- We want to discover what genes are expressed in a human tumor
- The human genome (3 Gb) can easily be stored in RAM
- Would this allow us to identify all expressed genes in the transcriptome?

Metagenomics

- We want to discover what microbes are living in the environment
- We might just be able to store all known microbes (>75 Gb) in the RAM
- Would this allow us to identify all microbes in the metagenome?
Index search is limited to exact matches

- Exact matches are fast, but may limit your results
- Human transcriptome:
  ...how about genetic variation between individuals?
  ...how about discovering novel mutations that may have caused the cancer?
  ...and sequencing errors?
- Metagenome:
  ...how diverse are microbes that live in nature?
  ...how about viruses, fungi, protists, etc.?
  ...how about discovering completely new species that have never been seen before?

Natural sequence divergence

- If we align metagenomic sequencing reads to a reference genome, we can distinguish multiple distinct SAR86 strains
  
  ![Sequence divergence graph]

  - The sequences at the top (~97% identity) belong to a strain that is closely related to the reference genome
  - The sequences below (~60-80% identity) are more distantly related strains
Best of both worlds

• If you allow more differences in your hit, ...
  ...your search will be more sensitive (can detect distant homology)
  ...but you can never store all possible sequences (=too much RAM)
  ...so it is necessary to use sequence alignment (=CPU operations)

• The solution is to combine the best of both worlds:
  – Quickly find potential hits using exact $k$-mers stored in an index (high RAM)
  – Extend only potential hits using sequence alignment (high CPU)

---

Basic Local Alignment Search Tool (BLAST)

• BLAST finds good potential homologs at reasonable speed
  – 10-50x faster than Smith-Waterman

• Terminology:
  – Query: sequence we search the database with
  – Hit or Subject: similar sequence found in the database

• BLAST is the most used bioinformatics program
  – More than 100,000 queries per day on the NCBI BLAST server
  – The BLAST article has been cited >58,000 times
The BLAST search algorithm

1. Identifies all words (length $W$) in the query
   - Default lengths: $W = 3$ for protein, $W = 11$ for DNA
   - Based on substitution scores

2. Quickly finds similar words in the database
   - “Similar” words are defined by using the substitution matrix (e.g. BLOSUM62)
   - All words in database sequences are stored in RAM
   - The index quickly locates all potential hit seqs

3. Extends seeds in both directions to find HSPs between query and hit
   - HSP: region that can be aligned with a score above a certain threshold

Example

Neighborhood words stored in the RAM allow potential hits to be rapidly retrieved

```
SLAALLFACKTPQGQRQLVNQWIKQPLMDKNRIERLNLVEA
FA +TP G R++ +W+
LGFSKYFATRTPGSRMLKRWLHDFSQSWCCAEFHHKWCVI
```

High-scoring Segment Pair (HSP)
BLAST input and output

BLAST input (query sequences)

MTQSSHAVAA FDLGAALRQE GLTETDYSEI QRDPNRAELG TFGV

protein_sequence_A

MLTETDYSEI QRRLGRDPNR AELGMFGVMN RAELGMFGY

protein_sequence_B

MHAVAAFQG LQSKQLTE TDYEQRL GRMFVMSH ECQYNRDA

protein_sequence_C

RPLLRPKPE FGAVVIV

BLAST output (hits)
What does a BLAST hit look like?

<table>
<thead>
<tr>
<th>Query and subject (hit) identifiers</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Query</strong>: P91002</td>
</tr>
</tbody>
</table>

**Output**:
- **Identifiers**: 8
  - Query and subject identifiers
- **Database**: RefSeq
- **Query length**: 163
- **Subject length**: 228

**Alignment**:
- **Score**: 339 (0.1%)
- **Expect**: 0.000001
- **Identity**: 45%
- **Positives**: 37 (9)
- **Negatives**: 131
- **Gaps**: 2 (0.1%)
Percent identities and positives

Note: positives are not reported for a DNA search

Alignment length, mismatches, gaps
Location of the HSP on the query and hit

E-value and bitscore

...more about E-values later
### BLAST flavors: direct searches

- **Nucleotide-nucleotide searches**
  - Nucleotide database & nucleotide query
  - blastn (default: W = 11 nucleotides)
    - Find homologous genes in different species
  - Megablast (default: W = 28 nucleotides)
    - Designed to efficiently find longer alignments between very similar nucleotide sequences
    - Best tool to find highly identical hits for a query sequence
    - For example: find sequences from the same species
  - Discontiguous Megablast
    - Uses discontiguous words (e.g. W = 11 nucleotides: AT - GT - AC - CG - CG - T)
    - For example, this can focus the search on codons (the third nucleotide of codons is less conserved due to the degeneracy of the genetic code → next slide)
    - Best tool to find nucleotide-nucleotide hits at larger evolutionary distances for protein-coding query sequences

- **Protein-protein searches**
  - Protein database & protein query sequences
  - blastp (default: W = 3 amino acids)
    - Find homologous proteins in different species

### BLAST flavors: translated searches

- We can exploit the conservation of protein sequences when aligning DNA sequences, by using translated searches

- This allows for more sensitive searches that detect homology at greater evolutionary distances
  - For example: homologous genes in distantly related species

- blastx and tblastx first translate the query from nucleotide into protein before identifying high-scoring words

- tblastn and tblastx use a translated database of nucleotide sequences stored as proteins
Profile alignment

- Alignments that use information about the sequence conservation into account are called profile alignments.
- In profile alignments, conserved residues have a bigger impact on the alignment score.
  - More conserved residues are weighed higher in the similarity score.
  - Less conserved residues are weighed lower in the similarity score.

→ Profile alignments are more sensitive in detecting homology than sequence alignments because all known variations are taken into account.
→ Profiles can be used to detect more distant homologs.

BLAST flavors: profile searches

- Position-Specific Iterated BLAST (PSI-BLAST) algorithm
  1. Perform initial blastp search with a query protein.
  2. Use the good database hits to construct a sequence profile.
  3. Search the database again, but with this sequence profile instead of with the original query sequence.
  4. Iterate steps 2-3.
Low-complexity regions

- Low-complexity regions can lead to high sequence similarity between unrelated sequences
- Solution: before performing a similarity search, low-complexity regions are often masked in the query so that they cannot contribute to the alignment score
  - This prevents identifying spurious hits

Low-complexity regions in BLAST output

- Lower case letters in the output show low-complexity sequence fragments that were ignored in the search
Faster tools are continually being developed.

The fastest algorithms generally use heuristics for speed. A practical method that is not guaranteed to be optimal, but sufficient for the present goals.

What heuristics are used by BLAST?

Bas E. Dutilh
Systems Biology: Bioinformatic Data Analysis
Utrecht University, February 27th 2017
Are these “real” hits or “spurious” hits?
What is our expectation?

• “Real” or “spurious” hit? This is a question of statistics!
• Statistics are **not well defined** for many bioinformatic analyses
  - For example, we need to consider the length of the sequence, the size of the database, the nucleotide composition of the sequences, and possible many other variables
• A simple solution is **data permutation**:
  - Permute (shuffle) the sequences 1000* times
  - Make 1000* new alignment matrices
  - Register if the alignment score of the permuted sequences is equal or higher than **Your Score**
  - This suggests that **Your Score** is not better than random

* or another high number
Relative frequency

The P-value is defined as the probability of observing a hit as good as, or better than \textbf{Your Score} by chance.

In permutation statistics, this corresponds to the fraction of times that the permuted score is equal or higher than \textbf{Your Score}.

Permutation statistics (shuffling)

- If the randomly permuted data rarely has a higher score than \textbf{Your score}, then the P-value is low, and your observation is meaningful.
- If the randomly permuted data often has a higher score than \textbf{Your score}, then the P-value is high, and your observation is not meaningful.
Permutation statistics

- In permutation statistics, the minimum P-value depends on the number of random permutations
  - For 100 permutations, the best P-value is <0.01
  - For 1000 permutations, the best P-value is <0.001

- Data permutation can randomize the signal, while preserving important characteristics of the data, such as:
  - Sequence length
  - Database size
  - Nucleotide composition
  - etc... (depending on your analysis)

- Permutation is a form of non-parametric statistics because it makes no assumptions about how the data is distributed

- Permutation statistics are often applied in bioinformatics
  - The computer can easily shuffle a dataset many times
  - Sometimes it can be challenging to figure out which aspects of the data to randomize!

Finding meaningful hits in a database

- How do we assess if a hit in the database is a real homolog?
How good is a hit?

- For a given query sequence, we want to find good hits
  - Highly similar sequence $\rightarrow$ likely to be true homologs

- We know how to quantify sequence similarity:
  - Alignment score
  - Percent of identical aligned residues
  - Percentage of positive-scoring aligned residues

- ...but the chance of finding a similar sequence (high score) by chance is larger in a larger database

E-value and bitscore

...more about E-values now
### Expect value (E-value)

- Number of hits you expect to find with a score, as good as, or better than [Your Score] (≥S) if the database was random.
- The E-value depends on [Your Score] and on:
  - The size of the database
  - The length of the query sequence
- The E-value is a parametric statistic because it assumes that the scores follow a Poisson distribution.

\[ E = Kmn e^{-\lambda S} \]

**Distribution of scores**
- **in a small database**
- **in a large database**

---

### Expect value (E-value)

- **E-value**: expected number of false positives at score S
  - “How often would you expect a hit at least this good (score ≥S) if the query and database were randomized?”
- Depends on:
  - Alignment score (S)
    - Higher alignment score: lower E-value ↓
  - Length of the query \( (m) \)
    - Longer query sequence: higher E-value ↑
  - Size of the database \( (n) \)
    - Larger database: higher E-value ↑
  - \( K \): constant for search space scaling
  - \( \lambda \): constant for substitution matrix correction
What is a good E-value?

• This is very difficult to say!
• Lower E-values are always better
  – Fewer false positives expected by chance in a database of this size
• As a rule of thumb:
  – E-values <10⁻⁶ for nucleotide blast (blastn, megablast) are good
  – E-values <10⁻³ for protein blast (blastp, blastx) are good

• Very low E-values are often given as exponents
• If you want to be very sure that your query and hit sequences are homologs, you should only trust extremely low E-values
• If you really have no other information about a protein, sometimes you might want to look at hits with high E-values
  – Can you trust a “best BLAST hit”? Cutoffs are important!

A low E-value: few false positives expected

• In the search below, we expect 10⁻¹⁴⁹ hits with a score of ≥436 bits by random chance
  – Given the database size and query sequence length, we expect
  – This is not much, so this is a good hit
A high E-value: a high risk of false positives

- In the search below, we expect 9.3 hits with a score of ≥38.9 bits by random chance
  - This is a lot, so this is a bad hit

E-value differs with different databases

- The same two sequences queryX (length m) and hitY will always give the same alignment and the same alignment score
- But if the database in which hitY was found is larger, the E-value becomes proportionally higher:
  \[ E = Kmne^{-\lambda S} \]
The E-value and the P-value

• E-value: expected number of hits with score ≥S by chance
  – The E-value is an expected number of false positives
  – The E-value ranges from 0 to the size of the database (n)

• P-value: probability of observing at least one hit with score ≥S by chance
  – The P-value is a probability of finding at least one false positive
  – The P-value ranges from 0 to 1

\[ P_S = 1 - e^{-E_S} \]

Your Score

The E-value and the P-value

• E-values and P-values are associated with a given score S
  – Example: if one expects to find 3 hits with score ≥S by chance (E = 3), the probability of finding at least one hit with score ≥S is 95% (P = 0.95)

• BLAST reports E-values rather than P-values because they are easier to interpret
  – Example: E-values of 5 and 10 expected false positives correspond to P-values of 0.993 and 0.99995, respectively

• For E-values < 0.01, P-values and E-values are nearly identical

The E-value and the P-value

- E-value: expected number of hits with score $\geq S$ by chance
- P-value: probability of observing at least one hit with score $\geq S$ by chance

$$P_S = 1 - e^{-E_S}$$