EXERCISE 5

PROTEIN PREDICTION I FOR CS

BLAST & PSI-BLAST
BLAST: Basic Local Alignment Search Tool
Recap: Dynamic Programming

- DP produces optimal local alignments
- But: runtime of $O(m \cdot n)$
- Not feasible to compare a sequence with millions of other sequences
BLAST: Basic Local Alignment Search Tool

- Search database for similar sequences
- Uses heuristics
  - Filter target sequences using high-scoring words
  - Do not allow gaps in local alignments
- Faster runtime, but alignments not always optimal
Searching for *words*

- Generate all words of length $w$ that when aligned with the query have a score greater or equal to $T$
- Search database for target sequences containing at least one of those words
- Extend words to find local high-scoring segment pairs (HSP) with a score greater or equal to $S$
  - Extend (without gaps) in one direction first, then the other
  - Stop extending into a direction once score drops $X$ below highest score encountered
  - Keep shortest highest-scoring HSP (always include original word)
Example: \( w = 3, T = 13 \)

Query: MGPRARPAFLLLMLLLQTAVL

Words:
- MGP  (Score = 5 + 6 + 7 = 18)
- LGP  (Score = 2 + 6 + 7 = 15)
- IGP  (Score = 1 + 6 + 7 = 14)
- MSP  (Score = 5 + 0 + 7 = 12)
- ...  
- GPR  (Score = 6 + 7 + 5 = 18)
- GPK  (Score = 6 + 7 + 2 = 15)
- GPD  (Score = 6 + 7 - 2 = 11)
- ...  
- PRA  (Score = 7 + 5 + 4 = 16)
- PKA  (Score = 7 + 2 + 4 = 13)
- PSA  (Score = 7 - 1 + 4 = 10)
- ...
Example: \( \textit{words} = \{\text{MGP}, \ldots, \text{FLL}, \ldots\} \)

Word: MGP
Targets: MGPRARPALF LLRTVAA

Word: FLL
Targets: MGELM AFLLPLIIIVL MVKHS

...
**Example:** word = FLL, $S = 20$, $X = 5$

**Query:** MGPRARPAFLLLMLLQTAVL

<table>
<thead>
<tr>
<th>Segment</th>
<th>Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>FLL</td>
<td>14</td>
</tr>
<tr>
<td>FLLP</td>
<td>11</td>
</tr>
<tr>
<td>FLLPL</td>
<td>13</td>
</tr>
<tr>
<td>FLLPLI</td>
<td>15</td>
</tr>
<tr>
<td>FLLPLII</td>
<td>17</td>
</tr>
<tr>
<td>FLLPLIIIV</td>
<td>15</td>
</tr>
<tr>
<td>FLLPLIIIVL</td>
<td>14</td>
</tr>
<tr>
<td>FLLPLIIIVLM</td>
<td>13</td>
</tr>
<tr>
<td>FLLPLIIIVLMV</td>
<td>17</td>
</tr>
<tr>
<td>FLLPLIIIVLMVK</td>
<td>15</td>
</tr>
<tr>
<td>AFLLPLII</td>
<td>21</td>
</tr>
<tr>
<td>MAFLLPLII</td>
<td>19</td>
</tr>
<tr>
<td>LMAFLLPLII</td>
<td>17</td>
</tr>
<tr>
<td>ELMAFLLPLII</td>
<td>16</td>
</tr>
</tbody>
</table>

**Target:** MGELMAFLLPLIIIVLMVKHS
**Example**

Query: MGPRARP**AFLLLLMLL**QTAVL
Target: MGELM**AFLLPLII**VLMVKHS

**Local alignment / HSP**

Score: 21

Query: 8 AFLLLLMLL 15
     ||||  +++

Target: 6 AFLLPLII 13
Statistical significance of HSPs

- Is the local alignment / HSP just a random hit?
- Calculate E-Value
  - Expected number of HSPs with the same or higher score found when comparing two random sequences
- Take size of query sequence and database into account
Normalize score $S$

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

- $\lambda$ and $K$ are empirical parameters

Calculate E-Value

$E = \frac{m \cdot n}{2^{s'}}$

- $m$ and $n$ are the size (number of residues) of the query sequence and database
The different flavors of BLAST

- blastp: protein-protein BLAST
- blastn: nucleotide-nucleotide BLAST
- blastx: nucleotide-protein BLAST (translates query)
- tblastn: protein-nucleotide BLAST (translates database)
- tblastx: nucleotide-nucleotide BLAST (translates both)
Runtime analysis

- $w$ regulates number of words
  - Small $w$: too few words
  - Large $w$: too many words

- $T$ regulates number of hits
  - Small $T$: too many hits
  - Large $T$: too many missed targets

- More than 90% of runtime is spent on extending hits
- How can we reduce the number of hits without missing too many targets?
Two-Hit Method

- Idea: significant HSPs often contain more than one word
- Only consider targets containing at least two words
  - Can be the same word
  - Both words must not overlap
  - Both words must be same distance \( A \) apart within the query and target sequence
- Extend only if distance \( A \) is small enough
- \( T \) must be lowered to get same sensitivity
Two-Hit Method

Gapped BLAST

- Improve local alignments by allowing gaps
- High computational costs (dynamic programming)
  - Explore only a limited alignment-path
  - Allow gaps only for HSPs with score greater or equal to $S_g$
Heuristic

- $S_g$ chosen to trigger one gapped extension per 50 sequences
- Start with central residue pair of highest scoring length-11 fragment in HSP
- Extend in both directions (dynamic programming)
- Ignore cells with scores $X_g$ below highest score
Gapped BLAST with Two-Hit Method

- Significantly reduced time for gapless extensions
  - Old BLAST: more than 90% of runtime
- Slightly more overhead
  - Determine two-hit method targets
  - Calculate gapped extensions
- But: new BLAST about three times faster than old BLAST
PSI-BLAST: Position-Specific Iterated BLAST
PSI-BLAST: Position-Specific Iterated BLAST

- BLAST: uses only a specific substitution matrix
- PSI-BLAST: iteratively generates and refines a PSSM
  - First iteration with standard substitution matrix
  - Collect local alignments into MSA and generate PSSM
  - Use PSSM for subsequent iterations
  - Terminate after \( n \) iterations or when no new hits are found
- Higher sensitivity to distant homologs
- Only available for protein-protein search
PSI-BLAST

1. Search database for similar sequences to query sequence
2. Calculate local alignments / HSPs
3. Construct a multiple sequence alignment from local alignments
4. Generate a PSSM from the multiple sequence alignment
Exercise
**Homework tasks**
- Implement a protein database object
- Implement the basic BLAST algorithm
- Implement the two-hit method

**Performance guidelines**
- Optimize your code for the tasks
- Do not use multi-threading
- Do not push solutions that run longer than 60s and/or need more than 2GBs of RAM for the offline tests
  - Best case: you fail the tests
  - Worst case: you overload the build servers (creating problems for everyone else)
Two-Hit extension algorithm

- Two hits $H_L$ and $H_R$ are HSP-candidates if...
  - they are on the same diagonal
  - they do not overlap and $L < R$
  - none of them is already included in another HSP on the same diagonal
  - $\text{distance}(H_L, H_R) \leq A$, with the distance being calculated between the two words’ start positions

- For two HSP-candidates $H_L$ and $H_R$ try to extend $H_R$ to the left (towards $H_L$) first
  - If the extension reaches end of $H_L$ they form a HSP
  - If they form a HSP try to extend right, too

- Process HSP-candidates closer to the start of the query sequence first
  - Distance to start of query measured from $H_L$
Thank you!

QUESTIONS?