Computational Biology

Lecture 7

Saad Mneimneh

Database Search

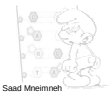
- Quadratic complexity not suitable for searching large databases
  - e.g. need to compare a query sequence to all sequences in a large database.
  - Alternative: Heuristics
    - BLAST
    - FAST
- Simple scoring scheme such as (+1, -1, -2) is not suitable for comparing protein sequences.
  - e.g. amino acids of similar size are more likely to get substituted for one another.
  - Alternative: Substitution matrix, $S(a,b)$ = score for aligning a with b
    - General approach for substitution matrices
    - PAM
    - BLOSUM

Saad Mneimneh

BLAST

(Basic Local Alignment Search Tool)

- BLAST returns a list of high scoring segment pairs between the query sequence and sequences in the database.
- A segment is a substring of a sequence.
- A segment pair is a pair of segments of the same length ➔ can from a gapless alignment.
- Basic BLAST is ungapped.
- Given a query sequence, BLAST returns all segment pairs between the query and a database sequence with score above a threshold S.
- S can be set by the user.



Saad Mneimneh

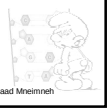
HOW does BLAST work?

- It finds certain “seeds” which are very short segment pairs between the query and the database sequence.
- These seeds are then extended in both directions without gaps, until the maximum possible score for extensions is reached.
- Time reduction: the extension stops when the score falls below a carefully computed limit *X*.

Saad Mneimneh

BLAST Algorithm

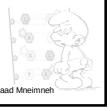
- For a given query sequence, compile a list of short high scoring strings (words in BLAST jargon)
- Search for hits – each hit gives a “seed”
- Extend “seeds”
- Return segments pairs with score > *S*.



Saad Mneimneh

k-mers

- How is the list of short high scoring strings obtained?
- k-mers: substrings of length *k*.
  - DNA sequence: all *k-mers*.
  - Protein sequence: all *k*-mers in addition to neighboring *k*-mers. A neighboring k-mer is a *k* length string that scores high with some *k*-mer of the sequence.
- Typical *k*: 3 or 4



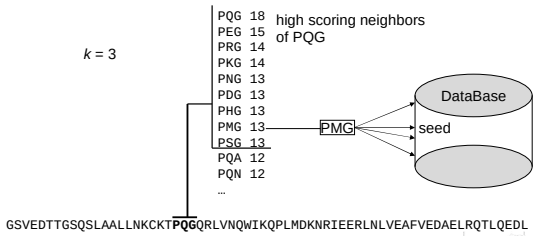
Saad Mneimneh

Database

- The database is hashed and indexed by all words of size *k*.
- Each word will point to the locations where it exists in the database.
- We have only 4^k words in case of DNA sequences and 20^k words in case of proteins.
- This is much less then the number of sequences stored in the database.

Saad Mneimneh

Overview



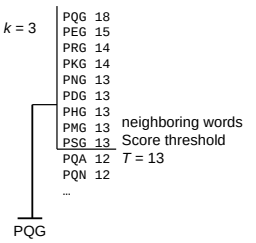
Saad Mneimneh

BLAST algorithm

- Split query into overlapping words of length *k* *(k*-mers*)*.
- For each word, find neighboring words that score at least *T*.
- Look into database where these words occur: seeds
- Extend each seed until score drops below *X*.
- If it scores > *S*, return segment pair.

Saad Mneimneh

Generating neighbors

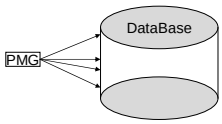


- For every amino acid in the word, try all possibilities
- Score the words
- Keep those with within threshold

Saad Mneimneh

Looking in database

- Each neighboring word gives a list of locations where it's found
- Follow pointers to obtain seeds



Saad Mneimneh

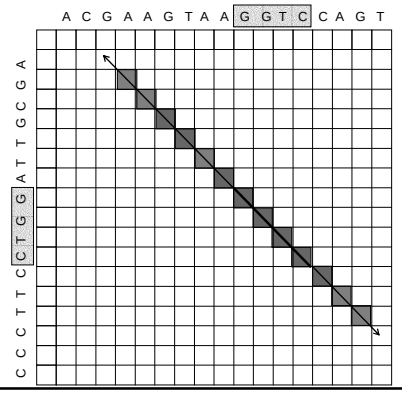
Extending seeds

```
GSVEDTTGSQSLAALLNKCKT PQGQRLVNQWIKQPLMDKNRIEERLNLVEAFVEDAELRQTLQEDL
          +LA++L+    TP G R++ +W+  P+ D     +ER     +A
          TLASVLDCTVT PMGSRMLKRWLHMPVRDTRVLLERQQTIGA
```

- Extend seed until score drops below *X*.
- Return highest scoring segment pair.

Saad Mneimneh

Example (thanks to serafim batzoglou)



seed: GGTC

$k = 4$

GTAAGGTCC
GTTAGGTCC

Saad Mneimneh

Why *k*-mers make sense?

- If two sequences have some level of similarity (say *L*%), they must contain a preserved *k*-mer for some *k*.
- Why?
- pigeonhole principle!

Saad Mneimneh

Example pigeonhole

- If we have 91 smurfs and 10 holes, there must be at least one hole with at least 10 smurfs.
- Proof: if non of the holes contain 10 smurfs, we have at most 9 x 10 = 90 smurfs!

Saad Mneimneh

Application to *k*-mers

- Two sequences of length 100 with > 90% similarity.
- There must be a preserved 10-mer.

Where will the 91st go?

Saad Mneimneh

Random model

- In the previous model, we cannot guarantee a *k*-mer for $k > 10$.
- What happens if we distribute the 91 similarities randomly?
- We get even better chance of having *k*-mers for other *k*s.

Saad Mneimneh

Running time

- n: length of query sequence
- s: number of seeds
- L: length of alignment
- Running time = $O(n + Ls)$
- For one sequence in the database,
  $s = O(n)$, $L = O(n)$ ➔ $O(n^2)$
  But in practice faster then Smith-Waterman.

Saad Mneimneh

Variations

- 2-hit BLAST
  - Require two seeds that are within 40 amino acids of each other to start considering a database sequence.
  - Reduce the space of potential hits, speeding up the algorithm.
- Gapped BLAST
  - BLAST with gaps, find a seed, then find more seeds and extend them, then join segments with gaps in a band around the main seed.

Saad Mneimneh

FAST

- Record all occurrences of windows of certain size k in the two sequences x and y (1-2 for DNA, 3-4 for proteins).
- If a window occurs at x_i and at y_j, we say it occurs at an offset $i - j$.
- Offset range is $1 - n$ to $m - 1$.

Saad Mneimneh

Example

- Window of size 2
- x = AGAGAG
- y = AAGAGAG
- The window AG occurs at x_1 and y_4, so it occurs at offset $1 - 4 = -3$. It also occurs at other offsets.
- What does it mean? Aligning x and y at offset -3 aligns the window AG.

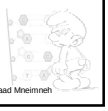
AGAGAG
AAG**AG**AG

- What is the offset that maximizes the number of aligned windows?

Saad Mneimneh

FAST algorithm

- Need
  - lookup table: contains all possible windows of size *k*, e.g. 4^k and their occurrence in *x* and *y*.
  - Offset vector: for each offsets, holds how many times that offset occurred.
- Fill the lookup table
- Compute the offset vector
- Choose the most frequent offset
- Align *x* and *y* at that offset



Saad Mneimneh

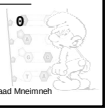
Example

x = AGAGAG
y = AAGAGAG

…	
AA	*y*: 1
AG	*x*: 1, 3, 5 *y*: 2, 4, 6
…	
GA	*x*: 2, 4 *y*: 3, 5
…	

-6	-5	-4	-3	-2	-1	0	1	2	3	4	5
0	1	0	3	0	5	0	3	0	1	0	0

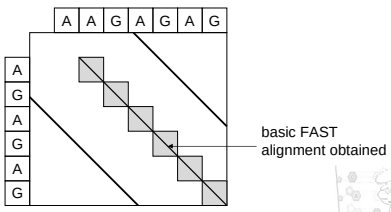
AGAGAG
AAGAGAG

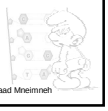


Saad Mneimneh

Variation

Run a bounded dynamic programming in a band centered at the offset diagonal.





Saad Mneimneh