Bioinformatics and BLAST

Overview

- Recap of last time
- Similarity discussion
- Algorithms:
  - Needleman-Wunsch
  - Smith-Waterman
  - BLAST
- Implementation issues and current research

Recap from Last Time

- Genome consists of entire DNA sequence of a species
- DNA is the *blueprint*
- Contains individual *genes*
- Genes are composed of four nucleotides (A,C,G,T)
- RNA transcribes DNA into proteins
- Proteins consist of 20 amino acids
- Proteins perform the actual *functions*

Recap from Last Time

- Identify similarities among the same gene from pig, sheep, and rabbit
- Things that were hard:
  - Manually matching letters
  - Deciding what “similar” meant
- Things that would have made it harder:
  - What if you had the whole genome?
  - What if there were missing letters/gaps?

Why do similarity search?

- Similarity indicates conserved function
- Human and mouse genes are more than 80% similar at sequence level
- But these genes are small fraction of genome
- Most sequences in the genome are not recognizably similar
- Comparing sequences helps us understand function
  - Locate similar gene in another species to understand your new gene
  - Rosetta stone

Issues to consider

- Dealing with gaps
  - Do we want gaps in alignment?
  - What are disadvantages of
    - Many small gaps?
    - Some big gaps?

Warning: similarity not transitive!

- If 1 is “similar” to 2, and 3 is “similar” to 2, is 1 similar to 3?
- Not necessarily
  - AAAAAABBBBBB is similar to AAAAAA and BBBBBB
  - But AAAAAA is not similar to BBBBBB
- “not transitive unless alignments are overlapping”

Summary

- Why are biological sequences similar to one another?
  - Start out similar, follow different paths
- Knowledge of how and why sequences change over time can help you interpret similarities and differences between them

BLAST

- **B**asic **L**ocal **A**lignment **S**earch **T**ool
- Algorithm for comparing a given sequence against sequences in a database
- A match between two sequences is an *alignment*
- Many BLAST databases and web services available

Example BLAST questions

- Which bacterial species have a protein that is related in lineage to a protein whose amino-acid sequence I know?
- Where does the DNA I’ve sequenced come from?
- What other genes encode proteins that exhibit structures similar to the one I’ve just determined?

Background: Identifying Similarity

- Algorithms to match sequences:
  - Needleman-Wunsch
  - Smith Waterman
  - BLAST

Needleman-Wunsch

- Global alignment algorithm
- An example: align COELACANTH and PELICAN
- Scoring scheme: +1 if letters match, -1 for mismatches, -1 for gaps

```
COELACANTH   COELACANTH
P-ELICAN--   -PELICAN--
```

Needleman-Wunsch Details

- Two-dimensional matrix
- Diagonal when two letters align
- Horizontal when letters paired to gaps

	C	O	E	L	A	C	A	N	T	H
P	C P	O								
E			E E							
L				L L						
I					A I					
C						C C				
A							A A			
N								N N	T -	H -

Needleman-Wunsch

- In reality, each cell of matrix contains score and pointer
- Score is derived from scoring scheme (-1 or +1 in our example)
- Pointer is an arrow that points up, left, or diagonal
- After initializing matrix, compute the score and arrow for each cell

Algorithm

- For each cell, compute
  - Match score: sum of preceding diagonal cell and score of aligning the two letters (+1 if match, -1 if no match)
  - Horizontal gap score: sum of score to the left and gap score (-1)
  - Vertical gap score: sum of score above and gap score (-1)
- Choose highest score and point arrow towards maximum cell
- When you finish, trace arrows back from lower right to get alignment

Smith-Waterman

- Modification of Needleman-Wunsch
  - Edges of matrix initialized to 0
  - Maximum score never less than 0
  - No pointer unless score greater than 0
  - Trace-back starts at highest score (rather than lower right) and ends at 0
- How do these changes affect the algorithm?

Global vs. Local

- Global – both sequences aligned along entire lengths
- Local – best subsequence alignment found
- Global alignment of two genomic sequences may not align exons
- Local alignment would only pick out maximum scoring exon

Complexity

- O(mn) time and memory
- This is impractical for long sequences!
- Observation: during fill phase of the algorithm, we only use two rows at a time
- Instead of calculating whole matrix, calculate score of maximum scoring alignment, and restrict search along diagonal

Other Observations

- Most boxes have a score of 0 – wasted computation
- Idea: make alignments where positive scores most likely (approximation)
- BLAST

Caveats

- Alignments play by computational, not biological, rules
- Similarity metrics may not capture biology
- Approximation may be preferred to reduce computational costs
- Any two sequences can be aligned
  - challenge is finding the proper meaning

BLAST

- Set of programs that search sequence databases for statistically significant similarities
- Complex- requires multiple steps and many parameters
- Five traditional BLAST programs:
  - BLASTN – nucleotides
  - BLASTSP, BLASTX, TBLASTN, TBLASTX - proteins

BLAST Algorithm

- Consider a graph with one sequence along X axis and one along Y axis
- Each pair of letters has score
- Alignment is a sequence of paired letters (may contain gaps)

Observations

- Recall Smith-Waterman will find maximum scoring alignment between two sequences
- But in practice, may have several good alignments or none
- What we really want is all *statistically significant* alignments

Observations (continued)

- Searching entire search space is expensive!
- BLAST can explore smaller search space
- Tradeoff: faster searches but may miss some hits

BLAST Overview

- Three heuristic layers: seeding, extension, and evaluation
- Seeding – identify where to start alignment
- Extension – extending alignment from seeds
- Evaluation – Determine which alignments are statistically significant

Seeding

- Idea: significant alignments have *words* in common
- *Word* is a defined number of letters
- Example: MGQLV contains 3-letter words MGQ, GQL, QLV
- BLAST locates all common words in a pair of sequences, then uses them as seeds for the alignment
- Eliminates a lot of the search space

What is a word hit?

- Simple definition: two identical words
- In practice, some good alignments may not contain identical words
- *Neighborhood* – all words that have a high similarity score to the word – at least as big as a threshold *T*
- Higher values of T reduce number of hits
- Word size *W* also affects number of hits
- Adjusting T and W controls both speed and sensitivity of BLAST

Some notes on scoring

- Amino acid *scoring matrices* measure similarity
- Mutations likely to produce similar amino acids
- Basic idea: amino acids that are similar should have higher scores
- Phenylalanine (F) frequently pairs with other hydrophobic amino acids (Y,W,M,V,I,L)
- Less frequently with hydrophilic amino acids (R,K,D,E, etc.)

Scoring Matrices

- PAM (**P**ercent **A**ccepted **M**utation)
  - Theoretical approach
  - Based on assumptions of mutation probabilities
- BLOSUM (**BLO**cks **SU**bstitution **M**atrix)
  - Empirical
  - Constructed from multiply aligned protein families
  - Ungapped segments (blocks) clustered based on percent identity

Seeding Implementation Details

- BLASTN (nucleotides)– seeds always identical, T never used
- To speed up BLASTN, increase W
- BLASTP uses W size 2 or 3
- To speed up protein searches, set W=3 and T to a large value

Extension

- Once search space is seeded, alignments generated by extending in both directions from the seed
- Example:

```
The quick brown fox jumps over the lazy dog.
The quiet brown cat purrs when she sees him.
```

- Can align first six characters
- How far should we continue?

Extension (continued)

- *X* parameter – How much is score allowed to drop off after last maximum?
- Example (assume identical scores +1 and mismatch scores -1)

```
The quick brown fox jump
The quiet brown cat purr
123 45654 56789 876 5654 <- score
000 00012 10000 123 4345 <-drop off
                           score
```

What is a good value of X?

- Small X risks premature termination
- But little benefit to a very large X
- Generally better to use large value
- W and T better for controlling speed than X

Implementation Details

- Extension differs in BLASTN and BLASTP
- Nucleotide sequences can be stored in compressed state (2 bits per nucleotide)
- If sequence contains N (unknown), replace with random nucleotide
- Two-bit approximation may cause extension to terminate prematurely

Evaluation

- Determine which alignments are statistically significant
- Simplest: throw out alignments below a score threshold *S*
- In practice, determining a good threshold complicated by multiple high scoring pairs (HSPs)

Implementation Details

- Recall three phases: seeding, extension, evaluation
- In reality, two rounds of extension and evaluation, gapped and ungapped
- Gapped extension and evaluation only if ungapped alignments exceed thresholds

Storage/Implementation

- Most BLAST databases are either a collection of files and scripts or simple relational schema
- What are limitations of these approaches?

Limitations of BLAST

- Can only search for a single query (e.g. find all genes similar to TTGGACAGGATCGA)
- What about more complex queries?
- “Find all genes in the human genome that are expressed in the liver and have a TTGGACAGGATCGA (allowing 1 or 2 mismatches) followed by GCCGCG within 40 symbols in a 4000 symbol stretch upstream from the gene”

Evaluating complex queries

- Idea: write a script that make multiple calls to a BLAST database
- An example query plan:
  1. Search for all instances of the two patterns in the human genome
  2. Combine results to find all pairs within 40 symbols of each other
  3. Consult a gene database to see if pair is upstream of any known gene
  4. Consult database to check if gene expressed in liver