Lecture 9
Database Searching

Database Searching for Similar Sequences

- Database searching for similar sequences is ubiquitous in bioinformatics.
- Databases are large and getting larger
- Need fast methods

Types of Searches

- Sequence similarity search with query sequence
- Alignment search with profile (scoring matrix with gap penalties)
- Serch with position-specific scoring matrix representing ungapped sequence alignment
- Iterative alignment search for similar sequences that starts with a query sequence, builds a multiple alignmnet, and then uses the alignment to augment the search
- Search query sequence for patterns representative of protein families

From Bioinformatics by Mount

DNA vs Protein Searches

- DNA sequences consists of 4 characters (nucleotides)
- Protein sequences consist of 20 characters (amino acids)
- Hence, it is easier to detect patterns in protein sequences than DNA sequences
- Better to convert DNA sequences to protein sequences for searches.

Database Searching Efficacy

- To evaluate searching methods, selectivity and sensitivity need to be considered.
- Selectivity is the ability of the method not to find members known to be of another group (i.e. false positives).
- Sensitivity is the ability of the method to find members of the same protein family as the query sequence.

Protein Searches

- Easier to identify protein families by sequence similarity rather than structural similarity. (same structure does not mean same sequence)
- Use the appropriate gap penalty scorings
- Evaluate results for statistical significance.

History

- Historically dynamic programming was used for database sequence similarity searching.
- Computer memory, disk space, and CPU speed were limiting factors.
- Speed still a factor due to the larger databases and increase number of searches.
- FASTA and BLAST allow fast searching.

History

- The PAM250 matrix was used for a long time. It corresponds to a period of time where only 20% of the amino acids have remained unchanged.
- BLOSUM has replace PAM250 in most applications. BLAST use the BLOSUM62 matrix. FASTA uses the BLOSUM50 matrix.

Search Tools

- Similarity Search Tools
  - Smith-Waterman Searching
- Heuristic Search Tools
  - FASTA
  - BLAST

Dynamic Programming

- Use Smith-Waterman algorithm or an improvement thereof for local alignment.
- Compares individual characters in the full-length sequence
- Slower but more sensitive than FASTA or BLAST
- Finds optimal Alignment

FASTA

- Fast alignment of pairs of protein or DNA sequences
- Searches for matching sequence patterns or words called k-tuples corresponding to k consecutive matches in both sequences
- Local alignments are build based on these matches.
- Better for DNA searches than BLAST (k-tuple can be smaller than minimum of 7 for BLAST)
- No guarantee of finding exactly optimal alignment

FASTA Algorithm

- FASTA uses a search for regions of similarity by hashing
- In hashing, a lookup table showing the positions of each k-tuple is made for each sequence
- The relative positions of the k-tuple in each sequence are calculated by subtracting the postions of the first characters
- K-tuples having the same offset are considered to be aligned.
- Adjacent regions are joined if possible by inserting gaps.
- The highest scoring regions are aligned by dynamics programming

FASTA Algorithm

- The number of comparisons increases linearly with the average sequence length
- In dynamic programming and dot plots, the number of comparisons increases as the cube or square of the length, respectively.

Significance of FASTA Searches

- The average score is plotted against the log of the average sequence length in each length range.
- A line is fit with linear regression and the z-score is the number of standard deviations from the fitted line.
- A statistical distribution of alignment scores can be used to determine probabilities.

Versions of FASTA

- There are many versions of FASTA
- FASTA – compares like query sequence to library
- TFASTA – compares unlike query sequence to library
- FASTF/TFASTF – short fragments
- FASTX/FASTY – compares DNA in all forward reading frames

BLAST

- Basic local alignment search tool
- Faster than FASTA
- Searches for words of common length in both query sequence and library
- Confines search to the words that are the most significant (FASTA finds all words).
- Significance of word matches is calculated using BLOSUM62 and the log odds score

BLAST Algorithm

- The query sequence if filtered to remove regions of low complexity (not useful for meaningful sequence alignments)
- A list of 3 character words in the query sequence is made stepping forward on character at a time
- The query sequence words are evaluated for an exact match with a word in the database (log odds scores using BLOSUM62)

BLAST Algorithm

- The neighborhood word score threshhold (cutoff score) to retain only the 50 most significant ones
- An efficient search tree is made using the high scoring words
- The database is searched to find matches for the 50 significant words. These are used as seeds for ungapped alignment with the query sequence

BLAST Algorithm

- The alignments are extended as long as the similarity score increases and if overlap, they are combined.
- These high-scoring segment pairs are matched in the entire database and listed
- The statistical significance for these are calculated

Database Searching with a Scoring Matrix or Profile

- A combination of dynamic programming, genetic algorithms or hidden Markov models can be used to extract patterns from a multiple sequence alignment
- Pattern finding and statistical methods (expectation minimization and Gibbs samplng) can be used also
- Example: PROFILE HMM

Database Searching with a Position Specific Scoring Matrix

- The previous method can be used to make a position specific scoring matrix.
- The position specific scoring matrix is moved along the sequence to score every possible sequence position in the query sequence.
- The highest scoring positions are typically the best matches for the corresponding set of sequences in the database
- Examples: EMOTIF, MOTIF, PHI-BLAST, BLOCKS, Profilesearch