Similarity search in bioinformatics
Sequence alignment
BLAST: Basic Local Alignment Search Tool
Projects

Worms
- A project in my other class this term (CIS 410/510 Bioinformatics) provides an example of how biologists use similarity searches
- the goal of the project is to learn about the evolution of a particular type of protein called a transcription factor (TF)
- focus on six TF genes, look at their structure in three related species of worm:
  - Caenorhabditis elegans
  - Caenorhabditis briggsae
  - Caenorhabditis remanei
- C. elegans has been extensively studied
  - a very important “model organism” used in a variety of research projects
  - only 1mm long
  - has exactly 959 cells

Worms (cont’d)
- The genome of C. elegans was one of the first to be fully sequenced
  - tons of information available at WormBase (http://wormbase.org)
  - all of the genes have been identified and “mapped”
- The genomes of the other two species have not yet been “annotated”
  - C. briggsae:
    - DNA has been recorded and assembled into chromosomes
    - not all genes have been identified yet
  - C. remanei:
    - DNA fragments have been sequenced
    - only small “contigs” are available -- no chromosomes yet
    - genes not identified
Finding Genes

- One of the first steps in this project: find the genes in *briggsae* and *remanei*
  - *C. elegans*: retrieve gene sequences by name from WormBase
  - *C. briggsae* and *C. remanei*: need to scan the DNA to find the genes
- One method: use BLAST
  - pattern: gene from *C. elegans*
  - text: chromosome or contig from *C. briggsae* or *C. remanei*
  - results: list of similar sequences

Alignment

- When biologists want to compare two sequences they use a technique known as *alignment*
- To align two sequences:
  - write them on separate lines
  - line up letters in common
  - insert “gaps” as necessary to take advantage of common letters
- Here is an alignment of the two DNA strings used as an example in the previous lecture:
  - AATTACTA
  - * *** * *
  - AATTCTA
- A gap is shown as a string of one or more dashes (– characters)
Alignment Algorithm

- The algorithm shown last time for computing the edit distance between two strings can also be used to produce an alignment.
  - for sequences of length \(n\) and \(m\) make an \((n+1) \times (m+1)\) matrix
  - label the rows and columns with letters from the sequences
  - fill in the cells with the costs of sequences of edit operations
  - the best path from upper left to lower right defines the lowest cost alignment

Alignment Algorithm (cont’d)

- The edit operations for alignment are slightly different.
  - For string comparison they were copy, change, insert, and delete.
    - the goal was to transform \(S\) (the string labeling rows) into \(T\) (labels on columns)
  - For alignment the operations are just copy and insert.
    - insert is used to place a gap character in either string

Alignment Algorithm (cont’d)

- To understand the operations, think of an automaton with the two strings \(S\) and \(T\) on its input tapes.
  - The output is a set of columns of the alignment.
  - The edit operations associated with each move are:
    - diagonal: make a column from \(S_i\) and \(T_j\)
    - right: make a column with \(T_j\) and –
    - down: make a column with – and \(S_i\).

Alignment Example

- For the example shown in this matrix the alignment is the one we saw before:
  
  –ATTAC\(\)TA
  AA\(\)TT\(\)C\(\)TA

  - diagonal: make a column from \(S_i\) and \(T_j\)
  - right: make a column with \(T_j\) and –
  - down: make a column with – and \(S_i\).
Alignment Costs

- The cost function used to make a matrix for protein or DNA uses estimates of evolutionary parameters
  - frequency of mutations from one amino acid to another
  - probability of inserting or deleting a large part of a gene
- The matrix at right shows where two hemoglobin proteins (mouse and human) match exactly and where they have similar amino acids

DNA Example Again

- Here is the matrix for the DNA example again
- Costs have been added to each cell by the dynamic programming algorithm
  - match = 0
  - change = 1
  - gap = 2
- It turns out the lowest cost alignment has fewer matches

Types of Alignments

- The two main types of alignments are global and local
  - In a **global alignment** the algorithm aligns all of the characters in each input sequence
    - example: aligning proteins
  - In a **local alignment** the algorithm just looks at short pieces of each string
    - example: a local alignment of two complete chromosomes should identify similar genes

BLAST

- **BLAST = Basic Local Alignment Search Tool**
  - probably the most widely used of all bioinformatics applications
  - so common the name is now a verb: “we blasted our sequence against....”
- A BLAST database is a collection of sequences
- Users provide an input string (a query)
- BLAST will search the database by doing local alignments of the query with strings in the database
- Output is a set of alignments, sorted by match quality
Implementation

- BLAST is very efficient
  - the BLAST database is basically a big hash table
  - keys are small substrings
  - for DNA database keys are usually 11 or 12 letters long
  - for protein databases keys might be 3 or 4 letters
- Extract keys from query, look them up in the hash table, find where they occur in the database
- Try to extend these initial “hits” with more letters from the query

Web BLAST

- NCBI and many other organizations have BLAST servers
  - cut and paste a query sequence
  - submit query, wait for results

Web BLAST (cont’d)

- The result page has a summary of the high quality hits
- Click on a link to learn about the matching sequence
- In this example the query sequence was hemoglobin from a fish (tetraodon)
- Matches were found for zebrafish, chicken, human, mouse, frog, ...

BLAST On Your Machine

- There are many versions of BLAST that you can download and install on your own machine
  - NCBI BLAST and WU-BLAST (Washington Univ St Louis) are the most common
  - The software package includes an application that will build a BLAST database from your own sequences
- Run the program as you would any other command line application:
  \% blastall -p blastp -i seq1.fa -d Yersinia_pestis
  
- The output is a set of local alignments of the query with different database sequences (next slide)
BLAST Output

The output from “standalone BLAST” is printed in the terminal window:

```
>45443158 NC_005810
Length = 496
Score = 30.4 bits (67), Expect = 0.15
Identities = 32/149 (21%), Positives = 64/149 (42%), Gaps = 21/149 (14%)
Query: 77  VNSINDLINIIIPHFDKYPLKYGKYTSYLIFKNIIEKMKNKQHLTQKGLIDIIN------ 130
  +N+  +L+ ++  +     L   K T Y +F  +  + KKN  +  K ++ +I+
Sbjct: 313 INAFGELLQLVPENKRSELLVNNKKTKYTLFDIVCSREKNKTTMAFKNIVSLISKNEYK 372
Query: 131 --------LTYIMNPLGKRKINK--KELFEFLKI--KDFSISDENNPYTDFSLSNKFLYQ 178
  +L +N +K +    E+FE + +  ++  I+  +N     SL     +
Sbjct: 373 LIESKVNILRTVINLSLKTDDSLNLKEFSGFSLAEPHVINKLHSIDIVSLESKLYH 432
```

ID of matching sequence  Hit quality  Local alignment of query vs. matching sequence

This is the same information returned by web BLAST -- the web server just reformats it so it’s easier to read and use (e.g. links to hits)

Review

- Biologists use sequence alignment to describe the similarity between DNA or protein sequences
- To align two sequences write one over the other, and insert gap characters in a way that maximizes the number of matching characters
  ```
  - ATTACTA
  AATT--CTA
  ```
- Use a cost function that allows the alignment algorithm to pair up similar letters
- The cost function also needs a “gap penalty” to represent the cost of - characters
- The $O(n^2)$ dynamic programming algorithm for edit distance can be adapted for sequence alignment
- BLAST (the local alignment search tool) uses these ideas to search a database for sequences similar to a query sequence

Projects

- Activities for the problem set and lab might include:
  - Make a dot plot for DNA sequences
  - Do a hand alignment from a dot plot
  - Given a matrix with filled in costs and “breadcrumbs” (backward pointing arrows) find the optimal path
  - Given a path in a matrix write down the alignment defined by the path
  - Connect to a BLAST server, run some BLASTs, explain the results