Introduction to Bioinformatics

Dan Lopresti
Professor and Chair
Computer Science & Engineering
Packard Lab 350
dal9@lehigh.edu
Motivation

“Biology easily has 500 years of exciting problems to work on.”
Donald Knuth (Stanford Professor & famous computer scientist)

By developing techniques for analyzing sequence data and related structures, we can attempt to understand molecular basis of life.

http://cmgm.stanford.edu/biochem218/
What is bioinformatics? Application of techniques from computer science to problems from biology.

Why is it interesting?

- Important problems.
- Massive quantities of data.
- Desperate need for efficient solutions.
- Success is rewarded.
Our genetic identity is encoded in long molecules made up of four basic units, the nucleic acids:

(1) Adenine,
(2) Cytosine,
(3) Guanine,
(4) Thymine.

To first approximation, DNA is a language over a four character alphabet, \( \{A, C, G, T\} \).
Complete set of chromosomes that determines an organism is known as its genome.

**GenBank Release 121.0 — December 15, 2000**

<table>
<thead>
<tr>
<th>Species</th>
<th>Haploid genome size</th>
<th>Bases</th>
<th>Entries</th>
</tr>
</thead>
<tbody>
<tr>
<td>Homo sapiens</td>
<td>3,400,000,000</td>
<td>6,702,881,570</td>
<td>3,918,724</td>
</tr>
<tr>
<td>Mus musculus</td>
<td>3,454,200,000</td>
<td>1,291,602,139</td>
<td>2,456,194</td>
</tr>
<tr>
<td>Drosophila melanogaster</td>
<td>180,000,000</td>
<td>487,561,384</td>
<td>166,554</td>
</tr>
<tr>
<td>Arabidopsis thaliana</td>
<td>100,000,000</td>
<td>242,674,129</td>
<td>181,388</td>
</tr>
<tr>
<td>Caenorhabditis elegans</td>
<td>100,000,000</td>
<td>203,544,197</td>
<td>114,553</td>
</tr>
<tr>
<td>Tetraodon nigroviridis</td>
<td>250,000,000</td>
<td>165,529,271</td>
<td>188,993</td>
</tr>
<tr>
<td>Oryza sativa</td>
<td></td>
<td></td>
<td>1,411</td>
</tr>
<tr>
<td>Rattus norvegicus</td>
<td></td>
<td></td>
<td>8,598</td>
</tr>
<tr>
<td>Bos taurus</td>
<td></td>
<td></td>
<td>9,473</td>
</tr>
<tr>
<td>Glycine max</td>
<td></td>
<td></td>
<td>14,1802</td>
</tr>
<tr>
<td>Medicago truncatula</td>
<td></td>
<td></td>
<td>44,360</td>
</tr>
<tr>
<td>Trypanosoma brucei</td>
<td></td>
<td></td>
<td>1,334</td>
</tr>
<tr>
<td>Lycopersicon esculentum</td>
<td></td>
<td></td>
<td>7,112</td>
</tr>
<tr>
<td>Giardia intestinalis</td>
<td></td>
<td></td>
<td>4,328</td>
</tr>
<tr>
<td>Strongylocentrotus purpurus</td>
<td></td>
<td></td>
<td>7,532</td>
</tr>
<tr>
<td>Entamoeba histolytica</td>
<td></td>
<td></td>
<td>9,938</td>
</tr>
<tr>
<td>Hordeum vulgare</td>
<td></td>
<td></td>
<td>—</td>
</tr>
<tr>
<td>Danio rerio</td>
<td>1,900,000,000</td>
<td>40,906,902</td>
<td>83,726</td>
</tr>
<tr>
<td>Zea mays</td>
<td>5,000,000,000</td>
<td>36,885,212</td>
<td>77,506</td>
</tr>
<tr>
<td>Saccharomyces cerevisiae</td>
<td>12,067,280</td>
<td>32,779,082</td>
<td>18,361</td>
</tr>
</tbody>
</table>

**Conclusion:** size does not matter! (But you already knew this. 😊)
Comparative Genomics

Here’s an amazing diagram:


How did we decipher these relationships?
Algorithms are Central

An algorithm is a precisely-specified series of steps to solve a particular problem of interest.

- Develop model(s) for task at hand.
- Study inherent computational complexity:
  - Can task be phrased as an optimization problem?
  - If so, can it be solved efficiently? Speed, memory, etc.
  - If we can't find a good algorithm, can we prove task is “hard”?
  - If known to be hard, is there approximation algorithm (one that works at least some of the time or comes close to optimal)?
- Conduct experimental evaluations (perhaps iterate above steps).
**Sequence Nature of Biology**

**Macromolecules** are chains of simpler molecules.

In the case of proteins, these basic building blocks are **amino acids**.

In DNA and RNA, they are **nucleotides**.

Macromolecules are chains of simpler molecules.

In the case of proteins, these basic building blocks are **amino acids**.

In DNA and RNA, they are **nucleotides**.
NCBI GenBank

National Center for Biotechnology Information (NCBI), which is branch of National Library of Medicine (NLM), which is branch of National Institutes of Health (NIH), maintains GenBank, a worldwide repository of genetic sequence data (all publicly available DNA sequences).

Reading DNA

Gel electrophoresis is the process of separating a mixture of molecules in a gel media by application of an electric field.

In general, DNA molecules with similar lengths will migrate the same distance.

Make DNA fragments that end at each base: A, C, G, T.

Then run gel and read off sequence: ATCGTG ...

http://www.apelex.fr/anglais/applications/sommaire2/sanger.htm
http://www.iupui.edu/~wellsctr/MMIA/htm/animations.htm
Reading DNA

Original sequence: \textit{ATCGTGTCGATAGCGCT}
Sequencing a Genome

Most genomes are enormous (e.g., $10^{10}$ base pairs in case of human). Current sequencing technology, on the other hand, only allows biologists to determine $\sim 10^3$ base pairs at a time.

This leads to some very interesting problems in bioinformatics ...

**Genetic linkage map**
(10$^7$ – 10$^8$ base pairs)

**Physical map**
(10$^5$ – 10$^6$ base pairs)

**Sequencing**
(10$^3$ – 10$^4$ base pairs)

ACTAGCCTGATCGATTTAGCAGCAG...
Sequencing a Genome

Genomes can also be determined using a technique known as *shotgun sequencing*.

Computer scientists have played an important role in developing algorithms for assembling such data.

It's kind of like putting together a jigsaw puzzle with millions of pieces (a lot of which are “blue sky”).

http://occawlonline.pearsoned.com/bookbind/pubbooks/bc_mcampbell_genomics_1/medialib/method/shotgun.html
Sequence Assembly

fragments

fragment assembly

target

original

contig

gap

Sequence Assembly

fragments

fragment assembly

target

original

contig

gap
A simple model of DNA assembly is the *Shortest Supersequence Problem*: given a set of sequences, find the shortest sequence $S$ such that each of original sequences appears as subsequence of $S$.

Look for overlap between *prefix* of one sequence and *suffix* of another:

$$\text{ACCGT}$$

$$\text{CGTGC}$$

$$\text{TTAC}$$

$$\text{TTACCGTGTC}$$
Sequence Assembly

Sketch of algorithm:

- Create an *overlap graph* in which every node represents a fragment and edges indicate overlap.
- Determine which overlaps will be used in the final assembly: find an *optimal spanning forest* in overlap graph.

W = AGTATTGGCAATC
Z = AATCGATG
U = ATGCAAAACCT
X = CCTTTTGG
Y = TTGGCAATCA
S = AATCAGG
Sequence Assembly

- Look for paths of maximum weight: use greedy algorithm to select edge with highest weight at every step.
- Selected edge must connect nodes with in- and out-degrees $\leq 1$.
- May end up with set of paths: each corresponds to a contig.

\[
\begin{align*}
AGTATTGGCAATC \\
TTGGCAATCA \\
AATCAGG \\
AATCGATG \\
ATGCAAACCT \\
CCTTTTGG \\
AGTATTGGCAATCAGG \\
AATCGATGCAAACCTTTTGG
\end{align*}
\]
Sequence Comparison

What's the problem? Google for biologists ...

- Given new DNA or protein sequence, biologist will want to search databases of known sequences to look for anything similar.
- Sequence similarity can provide clues about function and evolutionary relationships.
- Databases such as GenBank are far too large to search manually. To search them efficiently, we need an algorithm.

Shouldn't expect exact matches (so it's not really like google):

- Genomes aren't static: mutations, insertions, deletions.
- Human (and machine) error in reading sequencing gels.
Genomes Aren't Static

Sequence comparison must account for such effects.

http://www.accessexcellence.org/AB/GG/nhgri_PDFs/deletion.pdf  
http://www.accessexcellence.org/AB/GG/nhgri_PDFs/insertion.pdf
Genomes Aren't Static

Different kinds of mutations can arise during DNA replication:

http://www.accessexcellence.org/AB/GG/mutation.htm
The Human Factor

In addition, errors can arise during the sequencing process:

“...the error rate is generally less than 1% over the first 650 bases and then rises significantly over the remaining sequence.”

http://genome.med.harvard.edu/dnaseq.html

A hard-to-read gel (arrow marks location where bands of similar intensity appear in two different lanes):

Machines also make mistakes, of course!

Sequence Comparison

Why not just line up sequences and count matches?

\[
\begin{array}{ccccccc}
A & G & T & C & T & A & T \ A \\
\uparrow & \uparrow & \uparrow & \uparrow & \uparrow & \uparrow & \uparrow \\
A & T & T & C & T & G & T \ A \\
\end{array}
\]

\[\text{Difference} = 2\]

Doesn't work well in case of deletions or insertions:

\[
\begin{array}{ccccccc}
A & G & T & C & T & A & T \ A \\
\uparrow & \uparrow & \uparrow & \uparrow & \uparrow & \uparrow & \uparrow \\
G & T & C & T & A & T & A \\
\end{array}
\]

\[\text{Difference} = 8\]

One missing symbol at start of sequence leads to large difference!
Sequence Comparison

Instead, we'll use a technique known as *dynamic programming*.

- Model allows three basic operations: delete a single symbol, insert a single symbol, substitute one symbol for another.
- Goal: given two sequences, find the shortest series of operations needed to transform one into the other.

```
AGTCTATA
AGCTATA
AGCTGTA
```

*Delete T*

*Substitute G for A*
Sequence Comparison

How can we determine optimal series of operations?

- Approach is to build up longer solutions from previously computed shorter solutions.

- Say we want to compute solution at index $i$ in first sequence and index $j$ in second sequence:

Assume that we already know the best way to compare:

- $\text{Sequence 1}_i$ vs. $\text{Sequence 2}_j$
Sequence Comparison

So, best way to do this comparison:

\[ \text{Sequence 1} \rightarrow i \quad \text{vs.} \quad \text{Sequence 2} \rightarrow j \]

Is best choice from following three cases:

1. \[ \text{Sequence 1} \rightarrow i \quad \text{vs.} \quad \text{Sequence 2} \rightarrow j \quad + \text{inserting} \quad j \]
2. \[ \text{Sequence 1} \quad \text{vs.} \quad \text{Sequence 2} \rightarrow j \quad + \text{deleting} \quad i \]
3. \[ \text{Sequence 1} \quad \text{vs.} \quad \text{Sequence 2} \rightarrow j \quad + \text{substituting} \quad \text{for} \quad i \]
Sequence Comparison

Normally, this computation builds a table of distance values:

\[
d[i, j] = \min\left\{
\begin{array}{ll}
0 & \text{if } s[i] = t[j] \\
(i-1, j-1) + 1 & \text{if } s[i] \neq t[j] \\
\end{array}
\right.
\]

Normally, this computation builds a table of distance values:

<table>
<thead>
<tr>
<th></th>
<th>sequence s</th>
<th>sequence t</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>cost of inserting t</td>
<td>cost of inserting t</td>
</tr>
<tr>
<td>ε</td>
<td></td>
<td></td>
</tr>
<tr>
<td>s</td>
<td>sequence s</td>
<td>sequence t</td>
</tr>
<tr>
<td></td>
<td>cost of deleting s</td>
<td>cost of deleting s</td>
</tr>
</tbody>
</table>

\[
d[i, j] = \min\left\{
\begin{array}{ll}
0 & \text{if } s[i] = t[j] \\
(i-1, j-1) + 1 & \text{if } s[i] \neq t[j] \\
\end{array}
\right.
\]
Sequence Comparison

By keeping track of optimal decision, we can determine operations:
Genome Rearrangements

Recall what we saw earlier:

- 99% of mouse genes have homologues in human genome.
- 96% of mouse genes are in same relative location to one another.
- Mouse genome can be broken up into 300 synteny blocks which, when rearranged, yield human genome.
- Provides a way to think about evolutionary relationships.
Reversal Distance

Reversal distance is the minimum number of such steps needed.
Interesting Sidenote

Early work on a related problem, sorting by prefix reversals, was performed in 1970's by Christos Papadimitriou, a famous computer scientist now at UC Berkeley, and one “William H. Gates” ...

Yes, that Bill Gates ...
History of Chromosome X

Rat Consortium, Nature, 2004

Hypothesized reversals
Waardenburg’s Syndrome

Mouse provides insight into human genetic disorder:

- Waardenburg’s syndrome is characterized by pigmentary dysphasia.
- Disease gene linked to Chromosome 2, but not clear where it was located.

“Splotch” mice:

- A breed of mice (with splotch gene) had similar symptoms.
- Scientists succeeded in identifying location of gene in mice.
- This gave clues as to where same gene is located in humans.
Scientists build phylogenetic trees in an attempt to understand evolutionary relationships. Reversal distance is often used here.

Note: these trees are “best guesses” and certainly contain some errors!
DNA Microarrays

- Allows simultaneous measurement of the level of transcription for every gene in a genome (gene expression).
- Differential expression, changes over time.
- Single microarray can test ~10k genes.
- Data obtained faster than can be processed.
- Want to find genes that behave similarly.
- A pattern discovery problem.

\[ \text{green} = \text{repressed} \]
\[ \text{red} = \text{induced} \]
Using DNA Microarrays

- Track sample over a period of time to see gene expression over time.
- Track two different samples under same conditions to see difference in gene expressions.

Each box represents one gene’s expression over time

http://www.bioalgorithms.info/presentations/Ch10_Clustering.ppt
**DNA Microarrays**

*K-means clustering* is one way to organize this data:

- Given set of $n$ data points and an integer $k$.
- We want to find set of $k$ points that minimizes the mean-squared distance from each data point to its nearest cluster center.

Sketch of algorithm:

- Choose $k$ initial center points randomly and cluster data.
- Calculate new centers for each cluster using points in cluster.
- Re-cluster all data using new center points.
- Repeat second two steps until no data points are moved from one cluster to another or some other convergence criterion is met.
Clustering Microarray Data

- Pick $k = 2$ centers at random.
- Cluster data around these center points.
- Re-calculate centers based on current clusters.

From “Data Analysis Tools for DNA Microarrays” by Sorin Draghici.
Clustering Microarray Data

- Re-cluster data around new center points.

- Repeat last two steps until no more data points are moved into a different cluster.

From “Data Analysis Tools for DNA Microarrays” by Sorin Draghici.
Example of Hierarchical Clustering

Different genes that express similarly

Why Study Bioinformatics?

- Still many urgent open problems ⇒ lots of opportunities to make fundamental contributions (and become rich and famous).
- Stretch your creativity and problem-solving skills to the limit.
- Join a cross-disciplinary team – work with interesting people.
- Participate in unlocking the mysteries of life itself.
- Make the world a better place.
In CSE 308, we cover:

- Intro to molecular biology & algorithms,
- Genetic sequence comparison & alignment,
- Sequencing & assembly of DNA,
- DNA microarrays,
- Gene regulatory networks,
- Genome annotation,
- Transcription factor binding site prediction,
- Standard formats and sources for genomic data, etc.

Questions: lopresti@cse.lehigh.edu or chen@cse.lehigh.edu

CSE 308 is not a programming course! Good for BioS, BioE, CSE, and Math students
In CSE 350, we cover:

- Geometric modeling for proteins,
- Structure alignment & protein folding,
- Protein surfaces, cavities, electrostatics,
- Protein-protein and protein-DNA interfaces and interactions,
- Protein structure prediction, simulation, docking,
- Structural bioinformatics in pharmaceutical discovery,
- Function annotation, active site prediction, etc.

Questions: chen@cse.lehigh.edu

Recommended for seniors in BioS, BioE, CSE, and Math
Thank you!

GO LEHIGH!