Introduction to Bioinformatics

Dan Lopresti
Professor and Chair
Computer Science and Engineering
Office PL 350
dal9@lehigh.edu
Motivation

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

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

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

Why is it interesting?
- Important problems.
- Massive quantities of data.
- Great 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\}.

Genomes

Set of chromosomes that determines an organism is known as its genome.

<table>
<thead>
<tr>
<th>GenBank Release 121.0 — December 15, 2000</th>
</tr>
</thead>
<tbody>
<tr>
<td>Species</td>
</tr>
<tr>
<td>--------------------------</td>
</tr>
<tr>
<td>Homo sapiens</td>
</tr>
<tr>
<td>Mus musculus</td>
</tr>
<tr>
<td>Drosophila melanogaster</td>
</tr>
<tr>
<td>Arabidopsis thaliana</td>
</tr>
<tr>
<td>Caenorhabditis elegans</td>
</tr>
<tr>
<td>Tetraodon nigroviridis</td>
</tr>
<tr>
<td>Oryza sativa</td>
</tr>
<tr>
<td>Rattus norvegicus</td>
</tr>
<tr>
<td>Bos taurus</td>
</tr>
<tr>
<td>Glycine max</td>
</tr>
<tr>
<td>Medicago truncatula</td>
</tr>
<tr>
<td>Trypanosoma brucei</td>
</tr>
<tr>
<td>Lycopersicon esculentum</td>
</tr>
<tr>
<td>Giardia intestinalis</td>
</tr>
<tr>
<td>Strongylocentrotus purpuratus</td>
</tr>
<tr>
<td>Entamoeba histolytica</td>
</tr>
<tr>
<td>Hordeum vulgare</td>
</tr>
<tr>
<td>Danio rerio</td>
</tr>
<tr>
<td>Zea mays</td>
</tr>
<tr>
<td>Saccharomyces cerevisiae</td>
</tr>
</tbody>
</table>

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

http://www.cbs.dtu.dk/databases/DOGS/
http://www.nsrl.ttu.edu/tmot1/mus_musc.htm
http://www.oardc.ohio-state.edu/seedid/single.asp?strID=324
Comparative Genomics

Mouse and Human Genetic Similarities

How did we decipher these relationships?


Courtesy Lisa Stubbs
Oak Ridge National Laboratory
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?
  - Can it be solved efficiently? Speed, memory, etc.
  - If we can't find good algorithm, can we prove task hard?
  - If known to be hard, is there approximation algorithm (works some of the time or comes close to optimal)?
- Conduct experimental evaluations (iterate above steps).
Macromolecules are chains of simpler molecules.

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

In DNA and RNA, they are nucleotides.
National Center for Biotechnology Information (NCBI), a branch of National Institutes of Health (NIH), maintains GenBank, a worldwide repository of genetic sequence data (all publicly available DNA sequences).

Massive quantities of sequence data ⇒ need for good computational techniques.

Reading DNA

Gel electrophoresis separates mixture of molecules in a gel media by application of an electric field.

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

Make DNA fragments that end at each base. Then run gel and read off sequence: ATCGTG

This is known as sequencing.

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

Original sequence: ATCGTGT CGA TAG CGC T

- G
  - ATCG
  - ATCGTG
  - ATCGTGTCG
  - ATCGTGTCGATAG
  - ATCGTGTCGATAGCGATAGCG

- A
  - ATCGTG
  - ATCGTGTCG
  - ATCGTGTCGATAG
  - ATCGTGTCGATAGCG

- T
  - ATCGTG
  - ATCGTGTCG
  - ATCGTGTCGAT
  - ATCGTGTCGATAGCGCT

- C
  - ATCGTG
  - ATCGTGTCG
  - ATCGTGTCGATAG
  - ATCGTGTCGATAGCGCT
Sequencing a Genome

Most genomes are enormous (e.g., $10^{10}$ base pairs for human). But current sequencing technology only allows biologists to determine $\sim 10^3$ base pairs at a time.

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)
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 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
Sequence Assembly

Simple model of DNA assembly is *Shortest Supersequence Problem*: given set of sequences, find shortest sequence $S$ such that each of original sequences is a subsequence of $S$.

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

```
ACCGT
CGTGC
TTAC
```

```
TTACCGTGC
----ACCGT----
---CGTGC----
```

```
TTAC
```

```
TTACCGTGC
```
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 final assembly: find an *optimal spanning forest* in overlap graph.

W = AGTATTGGCAATC
Z = AATCGATG
U = ATGCAAACCT
X = CCTTTTGGG
Y = TTGGCAATCA
S = AATCAGG
Sequence Assembly

- Look for paths of maximum weight: use greedy algorithm to select edge with highest weight at each step.
- Edge must connect nodes with in- and out-degrees <= 1.
- May end up with set of paths: each yields a contig.

**Diagram:**

```
  s ---- 5 ---- y ---- 9 ---- x ---- 3 ---- u ---- 3 ---- z

AGTATTGGCAATC
   ^         ^
   T       TTGGCAATCA
  W->Y->S

AGTATTGGCAATCAGG
AATCAGG

AGTATTGGCAATC
   ^         ^
   T       ATGCAAACCT
  Z->U->X

AATCGATG
   ^         ^
   ATGCAAAACCT

CCTTTTGG

AATCGATGCAAAACCTTTTGG
```

W→Y→S

AATCAGG

AATCGATG

CCTTTTGG
Sequence Comparison

What’s the problem? Kind of like google for biologists ...

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

Can’t expect exact matches (i.e., 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 replication:

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

In addition, errors can arise during 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?

<table>
<thead>
<tr>
<th>A</th>
<th>G</th>
<th>T</th>
<th>C</th>
<th>T</th>
<th>A</th>
<th>T</th>
<th>A</th>
</tr>
</thead>
</table>

\[\text{Difference} = 2\]

<table>
<thead>
<tr>
<th>A</th>
<th>T</th>
<th>T</th>
<th>C</th>
<th>T</th>
<th>G</th>
<th>T</th>
<th>A</th>
</tr>
</thead>
</table>

\[\text{Difference} = 8\]

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

<table>
<thead>
<tr>
<th>A</th>
<th>G</th>
<th>T</th>
<th>C</th>
<th>T</th>
<th>A</th>
<th>T</th>
<th>A</th>
</tr>
</thead>
</table>

| G | T | C | T | A | T | A | A |

\[\text{Difference} = 8\]

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

Instead, we’ll use technique known as dynamic programming.

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

Instead, we’ll use technique known as dynamic programming.

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

\[
\begin{align*}
\text{Sequence 1}_i & \quad \text{vs.} \quad \text{Sequence 2}_j \\
\text{Sequence 1}_i & \quad \text{vs.} \quad \text{Sequence 2}_j \\
\text{Sequence 1} & \quad \text{vs.} \quad \text{Sequence 2}_j \\
\text{Sequence 1}_i & \quad \text{vs.} \quad \text{Sequence 2}_j
\end{align*}
\]

Assume that we already know the best way to compare:
Sequence Comparison

So, best way to do this comparison:

Sequence 1 \(i\) vs. Sequence 2 \(j\)

Is best choice from following three cases:

1. Sequence 1 \(i\) vs. Sequence 2 + inserting \(j\)
2. Sequence 1 vs. Sequence 2 \(j\) + deleting \(i\)
3. Sequence 1 vs. Sequence 2 + substituting for \(i\) to \(j\)
Sequence Comparison

Normally, this computation builds table of distance values:

\[
d[i, j] = \min \begin{cases} 
  d[i-1, j] + 1 & \text{if } s[i] = t[j] \\
  d[i, j-1] + 1 & \text{if } s[i] \neq t[j] \\
  0 & \text{if } s[i] = t[j] \\
  1 & \text{if } s[i] \neq t[j]
\end{cases}
\]

<table>
<thead>
<tr>
<th></th>
<th>ε</th>
<th>sequence t</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>cost of inserting t</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Cost of inserting \( t \)
Sequence Comparison

By maintaining record of optimal decisions, we can recover delete / insert / substitute 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.
- 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

Human Chromosome X

Mouse Chromosome X

Reversal distance is minimum number of steps needed.
Interesting Sidenote

Early work on related problem, sorting by prefix reversals, was done in 1970’s by Christos Papadimitriou, a professor now at UC Berkeley, and one “William H. Gates” ...

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

Hypothesized reversals

Rat Consortium, Nature, 2004
Waardenburg’s Syndrome

Mouse provides insight into human genetic disorder:

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

“Splotch” mice:

- A breed of mice (with splotch gene) had similar symptoms.
- Scientists identified location of gene in mice.
- Gave clues as to where same gene is located in humans.
Building the “Tree of Life”

Scientists build phylogenetic trees to help understand evolutionary relationships. Reversal distance often used.

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

- Allows simultaneous measurement of transcription level for every gene in a genome (gene expression).
- Differential expression, want to find genes that behave similarly over time.
- One microarray can test ~10k genes.
- Data obtained much faster than we can process it!
- Must find ways to uncover patterns.

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

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

Each cell represents one gene’s expression over time
**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 mean-squared distance from each data point to nearest center.

**Sketch of algorithm:**

- Choose \( k \) initial center points randomly and cluster data.
- Calculate new centers for clusters using points in cluster.
- Re-cluster all data using new center points.
- Repeat second two steps until no data points change clusters, 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 data points change clusters.

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

Different genes that express similarly

Why Study Bioinformatics?

- Many unanswered questions ⇒ opportunities to make fundamental contributions (+ become rich and famous).
- Stretch your creativity and problem-solving skills.
- Cross-disciplinary teams: work with interesting people.
- Participate in unlocking the mysteries of life itself.
- Make the world a better place.
In CSE 308 / BioE 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.

**CSE 308 is not a programming course! It’s for BioS, BioE, CSE, and Math students.**

Questions: lopresti@cse.lehigh.edu or chen@cse.lehigh.edu
In CSE 397, 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

Thank you!