Signal Processing on Databases

Jeremy Kepner

Lecture 6: Bio Sequence Cross Correlation

This work is sponsored by the Department of the Air Force under Air Force Contract #FA8721-05-C-0002. Opinions, interpretations, recommendations and conclusions are those of the authors and are not necessarily endorsed by the United States Government.
Outline

- Introduction
- Algorithm
- Implementation
- Results
- Summary
Relative Cost per DNA Sequence

- Big Data
- Energy Efficient
- Portable Sequencer
- High Volume Sequencer

© Thermo Fisher Scientific Inc. All rights reserved. This content is excluded from our Creative Commons license. For more information, see http://ocw.mit.edu/help/faq-fair-use/.

© Oxford Nanopore Technologies. All rights reserved. This content is excluded from our Creative Commons license. For more information, see http://ocw.mit.edu/help/faq-fair-use/.

Wetterstrand KA. DNA Sequencing Costs: Data from the NHGRI Large-Scale Genome Sequencing Program Available at: www.genome.gov/sequencingcosts. Accessed 03/08/2012
Example Disease Outbreak
May-July 2011 - Virulent *E. Coli* Outbreak Germany

Conclusions: Identification of *E. Coli* source too late to have substantial impact on illnesses
Publishing sequence data allowed for broad community to fully characterize pathogen

Sequencing and crowd source analysis showed promising potential -> Still too slow
Example Processing Timeline

<table>
<thead>
<tr>
<th>Complex Background</th>
<th>Collection &amp; Shipment of Sample</th>
<th>Sample Preparation &amp; Sequencing</th>
<th>Analysis of Samples</th>
<th>Time to Actionable Data</th>
</tr>
</thead>
<tbody>
<tr>
<td>Current</td>
<td>2-30 days</td>
<td>1-3 days</td>
<td>7-14 days</td>
<td>10-45 days</td>
</tr>
<tr>
<td>Goal</td>
<td>on site</td>
<td>3 hours</td>
<td>&lt;12 hours</td>
<td>&lt;1 day</td>
</tr>
</tbody>
</table>

- Processing plays a key part in accelerating the overall time to solution

© image sources unknown. All rights reserved. This content is excluded from our Creative Commons license. For more information, see [http://ocw.mit.edu/help/faq-fair-use/](http://ocw.mit.edu/help/faq-fair-use/).

Slide inspired by Nicole Rosenzweig, ECBC
DNA Sequence Matching

Goal

• Quickly compare two sets of DNA

Applications

• Identification
• Mixture Analysis
• Kinship Analysis
• Ancestry Analysis

Uses: disease outbreaks, criminal investigations, personal medicine, …

• Challenge: sequencing matching takes a long time, can we make it faster?
Outline

- Introduction
- Algorithm
- Implementation
- Results
- Summary
### RNA Reference Set

<table>
<thead>
<tr>
<th>SeqID</th>
<th>sequence</th>
</tr>
</thead>
<tbody>
<tr>
<td>AB000106.1_1-1343</td>
<td>ggaatctgccctgtggctgg</td>
</tr>
<tr>
<td>AB000278.1_1-1410</td>
<td>caggcctaacacatgcaagt</td>
</tr>
<tr>
<td>AB000389.1_1-1508</td>
<td>ttgatctgtgctctagattgaa</td>
</tr>
<tr>
<td>AB000390.2_1-1428</td>
<td>catgcaagtcgagcggagaag</td>
</tr>
</tbody>
</table>

### Collected Sample

<table>
<thead>
<tr>
<th>SeqID</th>
<th>sequence</th>
</tr>
</thead>
<tbody>
<tr>
<td>G6J0L4R01AUUY3</td>
<td>TAGATACTGCTGCTCCCC</td>
</tr>
<tr>
<td>G6J0L4R01DKLJM</td>
<td>TTTTTTCGTCCTGCTGCT</td>
</tr>
<tr>
<td>G6J0L4R01DOSEN</td>
<td>TTATCGGCTGCTGCTCCCC</td>
</tr>
<tr>
<td>G6J0L4R01EOS3L</td>
<td>AGGTGTCTGCTGCTCTCA</td>
</tr>
</tbody>
</table>

---

**Sequence Matching \( \equiv \)**

**Sparse Matrix Multiply in D4M**

- Associative arrays provide a natural framework for sequence matching
Using 10mer distribution can quickly select reference 10mers that maximally differentiate sample sequences and eliminate most 10mers.
0.5% Selection Results

- **Sample (20MB):**
  - NGS from Roche 454

- **Reference (500MB):**
  - Virus DNA from GenBank

- All strong matches detected using 0.5% of data
Cumulative Probability of Detection

- 100% detection of all true matches > 100
• Measured matches > 10 are always matches
Finding Matches

A = A1 * A2'
Ak = CatKeyMul(A1, A2')

• Find sequences with >6 word matches

Am = Ak(Row(A > 6), Col(A > 6))

(AB001520.1_1-1428, G6J0L4R01B4UPM)
aaatctttaa;aatctttaa;ctttaaataa;ggggaccagc;taaatctttta;ttttaaataaa;ttttaaataaa;

(AB002634.1_1-1419, G6J0L4R01EDJVA)
aatatgcgttt;atatgcgtttt;atgtcgttttc;gtcgtttccct;tgctttccct;tgctttccct;

• Associative array cat multiply preserves pedigree of matches
Sequence Alignment

• Show relative alignments of sequences

A1(Row(Am), Val(Am)) + A2(Row(Am), Val(Am))

reference   sample

AB001520.1_1-1428  G6J0L4R01B4UPM
aatctttaa  564  155
aatcttttaa  1227  156
ctttaataa  1376  159
gggaccagc  877  58
taaaccttaa  563  154
ttaataaaaa  1378  161
ttaataaaaa  1377  160

AB002634.1_1-1419  G6J0L4R01EDJVA
aatgtgcgttt  933  300
aatgtgcgttt  934  301
atgtgcgtttc  935  302
atgtgcgtttc  937  304
gtctcagttc  1211  37
tgtcagttc  938  305
tgtcagttc  936  303

• Sequence alignment found by indexing into associative array
Outline

- Introduction
- Algorithm
- Implementation
- Results
- Summary
A D4M query returns a sparse matrix or graph from a triple store…

Triple store are high performance distributed databases for heterogeneous data

…for statistical signal processing or graph analysis in Matlab

D4M Stores Giant Sparse Matrices in Accumululo Triple Store Database
C-Parser

- Generate n-mer words from Fasta DNA data in D4M format
- Preprocess & filter the sequences
  - Ignore bad, common sequences
  - Break output files into manageable chunks, say 5MB
  - Generate reverse sequences
  - Break up big sequences into subsequences to preserve locality
Database Table Splits

- Initial inserts bottleneck on one tablet server until it fills up and splits
- Performance booster: pre-split table among several tablet servers for instant parallel insertion
  - Use advanced knowledge of row data patterns to choose splits
- Created functions to set and query table splits

**No Splitting**

**Table Splits**
Accumulator Columns

- Accumulator columns allow counting to be done on insert
  - Example: sequence counting
    - Row ID = 10 mer
    - Column = Count
    - Value = Count
    - Insert (aatcttttaa, Count, 2) → DB has (Doc1, ‘bird’, 2)
    - Insert (aatcttttaa, Count, 3) → DB has (aatcttttaa, Count, 5)

- Works with any commutative operation
  - Addition, maximum, minimum, etc.

Database Accumulator

Courtesy of Jan Van der Spiegel. Used with permission.
Outline

- Introduction
- Algorithm
- Implementation
- Results
- Summary
Table Split Performance

Split vs. No-Split Performance

- Pre-Splitting tables appropriately can double ingest rates at higher Np in multinode database environments
Human DNA DB Ingest Performance

Accumulo Instance8 Tablet Server (2 cores each)

Extrapolated Run Times

- 4.5 GB human Fasta file
- C Parser took 25 minutes
- 101 GB of row, col files
- Database ingest time ~10 hours
Leveraging “Big Data” Technologies for High Speed Sequence Matching

- High performance triple store database trades computations for lookups
- Used Apache Accumulo database to accelerate comparison by 100x
- Used Lincoln D4M software to reduce code size by 100x

Graph:
- x-axis: code volume (lines)
- y-axis: run time (seconds)
- D4M
- BLAST
- 100x faster
- 100x smaller

Legend:
- D4M
- D4M + Triple Store
- BLAST
• Big data is found across a wide range of areas
  – Document analysis
  – Computer network analysis
  – DNA Sequencing

• Currently there is a gap in big data analysis tools for algorithm developers

• D4M fills this gap by providing algorithm developers composable associative arrays that admit linear algebraic manipulation
Example Code & Assignment

- Example Code
  - d4m_api/examples/2Apps/4BioBlast

- Assignment
  - None