Optimization of Exact Algorithms for Planted (l,d)-Motif Search

Andrew Petersen - Loyola Marymount University
Edward Felekey - University of Connecticut
Samuel Van Pelt - Colorado Mesa University

Aug 1, 2013
Overview

- Background
- Planted (l,d)-Motif Search
- Implementations
- Results
Importance of Motifs

- Relevant to biology, the human body, disease
  - cellular regulatory networks
  - Gene expression
    - Transcription factor binding sites
  - DNA structure/organization within the nucleus
    - Learn about non-coding DNA
- Related to Protein structure, more complicated
- Large amount of biological data
  - Many genome projects
    - Human Genome Project, ENCODE
- Computational tools make this analysis possible
Describing Motifs

- **What is a Motif?**
  - Short pattern in large dataset
  - DNA or protein sequences

- **Consensus Sequences / Regular Expressions**

- **Sequence and Hamming Distance**
  - $d$, Hamming Distance, number of differing characters
  - ACGTAA, $d=2$
Algorithm Types

- Heuristic Algorithms
  - Fast, but miss some motifs
- Exact Algorithms
  - Finds every motif, but slower
Planted \((l,d)\)-Motif Search

**Problem Definition**

- Given \(t\) sequences of length \(n\) in alphabet \(\Sigma\), find every motif \(M\) of length \(l\), which is a Hamming distance of \(d\) away from at least one subsequence in every input sequence.

- \(t\) - Number of input sequences
- \(n\) - Length of each input sequence
- \(\Sigma\) - Alphabet of sequence
- \(M\) - Motif, the sequence to be found
- \(l\) - Length of motifs to return
- \(d\) - Hamming distance
  - Number of characters that can be changed in \(M\)
Example

- Input sequences: $n = 10; t = 2$
  - AAAA
  - GGGGG

- Motif requirements: $l = 4; d = 2$

- Motifs: AAGG, AGAG, AGGA, GAAG, GAGA, GGAA
Straightforward Approach

- Easy to check if a given sequence is a motif
- Hard to generate all the motifs
- Go through every l-mer in alphabet
PMS1 Algorithm

For each subsequence of length \( l \) in each input sequence, generate every \( l \)-mer that is a Hamming distance of \( d \) away. Any \( l \)-mer that is generated by every sequence is a motif.

- Given:
  - \( t \) input sequences of length \( n \)
  - Motif length \( l \)
  - Hamming distance \( d \)

- Generate candidate motifs from the input sequences
- If a potential motif is in every generated collection, it is a motif
Example

- Input sequence: n = 10;
  - AAAAAAAAAAAAA
- Motif requirements: l = 3; d = 1
- Generated from l-mer “AAA”:
  - AAG, AGA, GAA,
  - AAC, ACA, CAA,
  - AAT, ATA, TAA
Asymptotic Runtime

\[ O\left(t(n - l + 1)\binom{l}{d}(|\Sigma| - 1)^d \frac{l}{w}\right) \]

- **t** - Number of input sequences
- **n** - Length of each input sequence
- **\(\sum\)** - Alphabet of sequence
- **l** - Length of motifs to return
- **d** - Hamming distance
- **w** - Word length of computer
Overview of Implementations

- PMS1, with strings
- Encoding sequences in integers
- Intersect instead of Merge
- Custom Sort
- k - split
- 32/64-bit
Integer Encoding of l-mers

- Efficiently pack data
  - Don’t waste space
  - Normal character encoding - ASCII
- DNA alphabet has 4 characters (ACTG)
  - Only need 2 bits to store a character
  - 00 = G; 01 = C; 10 = T; 11 = A
- More easily use lower-level operations
Low-Level Operations

- Shift left/right
  - Ex: $\text{pow}(4,n)$ becomes $1<<(2*n)$
- AND, OR, XOR
  - Use XOR to set to zero
- Isolate characters
- Increment characters
- Convert from string to integers