An Efficient Exact Solution to the (l,d) Planted Motif Problem

Maria Clara Isabel Sia*
Julieta Nabos
Proceso Fernandez

November 14, 2015

Introduction

The (l, d) planted motif problem

- motifs: repeated, biologically significant subsequences in DNA
- DNA motif finding must allow for mismatches due to mutation
- known as a difficult problem in computational biology and CS (NP-complete)

Introduction

The (l, d) planted motif problem

Find a motif of length $l{=}8$ across these 5 DNA sequences.
Each contains the motif with at most $d{=}2$ mismatches.

S_1	`atcactcgttctcctctaatgtgtaaagacgtactaccgacctta`
S_2	`acgccgaccggtccgatccttgtatagctcctaacgggcatcagc`
S_3	`tcctgactgcatcgcgatctcggtagtttcctgttcatcattttt`
S_4	`ggccctcagcatcgtgcgtcctgctaacacattcccatgcagctt`
S_5	`tgaaaagaatttacggtaaaggatccacatccaatcgtgtgaaag`

Planted motif: `ccatcgtt`

Introduction

Key concepts

- l-mer
- Hamming distance d_H
- d-neighbor

EMS-GT

Introduction

- an exact motif search (EMS) algorithm that uses the candidate generate-and-test (GT) approach

- exact algorithms search exhaustively to find all possible motifs,
  - as opposed to heuristic ones which sample/guess motifs

- *generate* - narrows the search to a set of candidate motifs
  test - checks each candidate to see if it is a motif

EMS-GT

Demonstration

S_1 `atcactcgttctcctctaatgtgtaaagacgtactaccgacctta`

S_2 `acgccgaccggtccgatccttgtatagctcctaacgggcatcagc`

S_3 `tcctgactgcatcgcgatctcggtagtttcctgttcatcattttt`

S_4 `ggccctcagcatcgtgcgtcctgctaacacattcccatgcagctt`

S_5 `tgaaaagaatttacggtaaaggatccacatccaatcgtgtgaaag`

EMS-GT

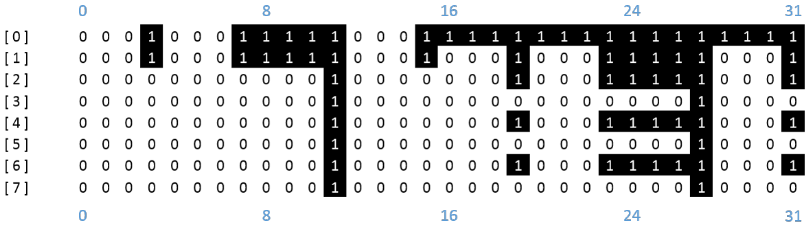
Representing sets

- EMS-GT must operate on sets of l-mers.
- There are 4^l possible l-mers that can be formed with $\{\mathtt{a,c,g,t}\}$
- Thus, to represent a set of l-mers, EMS-GT uses 4^l bits,
  - set to 1 if the corresponding l-mer is a member of the set,
  - set to 0 otherwise.
- For efficiency, EMS-GT stores the 4^l bits as $\frac{4^l}{32}$ 32-bit integers.

EMS-GT

Representing sets

- $N($ `acgt, 1` $)$ $\quad l{=}4;\ \ 4^l = 256,\ \frac{4^l}{32} = 8$

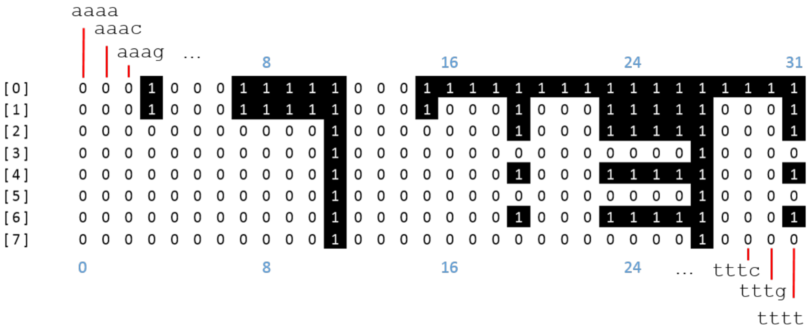


	0								8								16								24							31
[0]	0	0	0	1	0	0	0	1	1	1	1	1	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
[1]	0	0	0	1	0	0	0	1	1	1	1	1	0	0	0	1	0	0	0	1	0	0	0	1	1	1	1	1	0	0	0	1
[2]	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	1	1	1	1	1	0	0	0	1
[3]	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
[4]	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	1	1	1	1	1	0	0	0	1
[5]	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
[6]	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	1	1	1	1	1	0	0	0	1
[7]	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
	0								8								16								24							31

EMS-GT

Representing sets

- $N($ acgt, 1 $)$ $\quad l{=}4;\ \ 4^l = 256,\ \frac{4^l}{32} = 8$

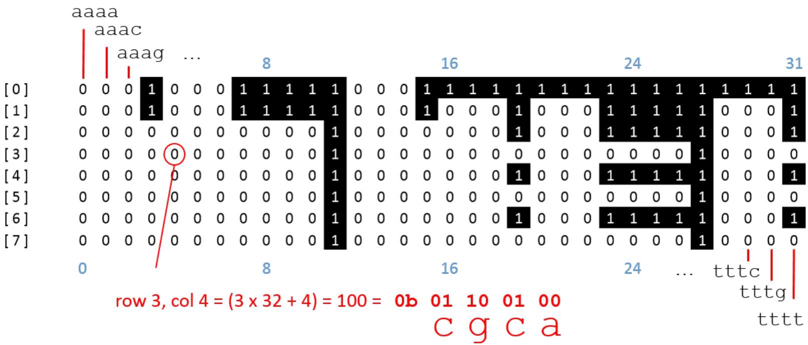


									8								16								24							31
[0]	0	0	0	1	0	0	0	1	1	1	1	1	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
[1]	0	0	0	1	0	0	0	1	1	1	1	1	0	0	0	1	0	0	0	1	0	0	0	1	1	1	1	1	0	0	0	1
[2]	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	1	1	1	1	1	0	0	0	1
[3]	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
[4]	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	1	1	1	1	1	0	0	0	1
[5]	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
[6]	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	1	1	1	1	1	0	0	0	1
[7]	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
	0								8								16								24							

EMS-GT

Representing sets

- $N($ acgt, 1 $)$ $\quad l{=}4;\ \ 4^l = 256,\ \frac{4^l}{32} = 8$



	0								8								16								24							31
[0]	0	0	0	1	0	0	0	1	1	1	1	1	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
[1]	0	0	0	1	0	0	0	1	1	1	1	1	0	0	0	1	0	0	0	1	0	0	0	1	1	1	1	1	0	0	0	1
[2]	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	1	1	1	1	1	0	0	0	1
[3]	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
[4]	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	1	1	1	1	1	0	0	0	1
[5]	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
[6]	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	1	1	1	1	1	0	0	0	1
[7]	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
	0								8								16								24							

row 3, col 4 = (3 x 32 + 4) = 100 = **0b 01 10 01 00**

c g c a

EMS-GT

Building sets

discuss recursive

EMS-GT

Building sets in blocks

EMS-GT

Results