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

Maria Clara Isabel Sia*
Julieta Nabos
Proceso Fernandez

November 16, 2015

Context of the problem

(l, d) planted motif problem

- DNA motif finding: known as a difficult (NP-complete) problem in computational biology and computer science
- motifs: important sequences that occur repeatedly (but not cleanly, due to mutation) in DNA
- (l, d) planted motif problem: search for a common motif of length l, allowing for up to d mismatches due to mutation.

Context of the problem

(l, d) planted motif problem

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

S_1	`atcactcgttctcctctaatgtgtaaagacgtactaccgacctta`
S_2	`acgccgaccggtcccatccttgtatagctcctaacgggcatcagc`
S_3	`tcctgactgcatcgcgatctcggtagtttcctgttcatcattttt`
S_4	`ggccctcagcatcgtgcgtcctgctaacacattcccatgcagctt`
S_5	`tgaaaagaatttacggtaaaggatccacatccaatcatttgaaag`

Planted motif: ? ?

Context of the problem

(l, d) planted motif problem

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

S_1 `atcactcgttctcctctaatgtgtaaagacgtactaccgacctta`

S_2 `acgccgaccggtcccatccttgtatagctcctaacgggcatcagc`

S_3 `tcctgactgcatcgcgatctcggtagtttcctgttcatcattttt`

S_4 `ggccctcagcatcgtgcgtcctgctaacacattcccatgcagctt`

S_5 `tgaaaagaatttacggtaaaggatccacatccaatcatttgaaag`

Planted motif: `ccatcgtt`

Key concepts

(l, d) planted motif problem

- an *l-mer* is a sequence of length l
- two l-mers are *d-neighbors* if they have at most d mismatches

l=8, d=3,

$$x_1 = \texttt{cgatcctt} \qquad d_H(x_1, x_2) = 2$$
$$x_2 = \texttt{ccatcgtt}$$

EMS-GT

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

- generate - narrow down the search to a set of candidate motifs
- test - check each candidate to determine if it is a motif

Demonstration of algorithm

EMS-GT

S_1 atcactcgttctcctctaatgtgtaaagacgtactaccgacctta

S_2 acgccgaccggtccgatccttgtatagctcctaacgggcatcagc

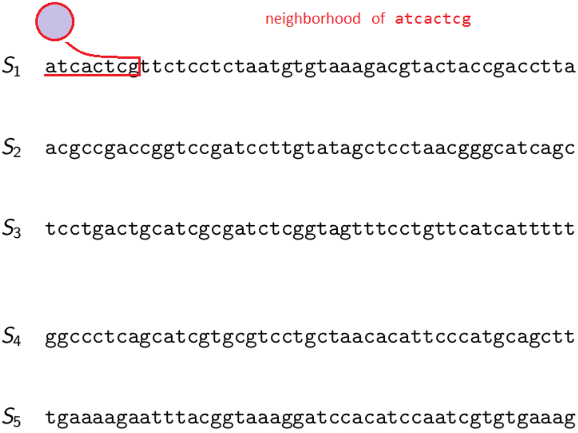
S_3 tcctgactgcatcgcgatctcggtagtttcctgttcatcatttt

S_4 ggccctcagcatcgtgcgtcctgctaacacattcccatgcagctt

S_5 tgaaaagaatttacggtaaaggatccacatccaatcgtgtgaaag

Demonstration of algorithm

EMS-GT



neighborhood of **`atcactcg`**

S_1 `atcactcgttctcctctaatgtgtaaagacgtactaccgacctta`

S_2 `acgccgaccggtccgatccttgtatagctcctaacgggcatcagc`

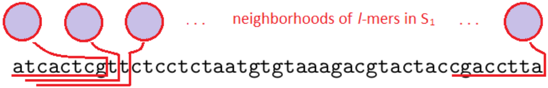
S_3 `tcctgactgcatcgcgatctcggtagtttcctgttcatcattttt`

S_4 `ggccctcagcatcgtgcgtcctgctaacacattcccatgcagctt`

S_5 `tgaaaagaatttacggtaaaggatccacatccaatcgtgtgaaag`

Demonstration of algorithm

EMS-GT



. . . neighborhoods of l-mers in S_1 . . .

S_1 atcactcgttctcctctaatgtgtaaagacgtactaccgacctta

S_2 acgccgaccggtccgatccttgtatagctcctaacgggcatcagc

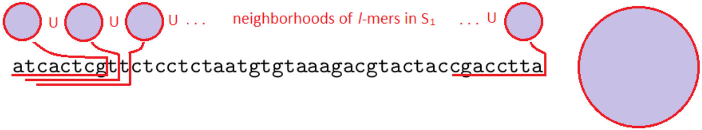
S_3 tcctgactgcatcgcgatctcggtagtttcctgttcatcatttt

S_4 ggccctcagcatcgtgcgtcctgctaacacattcccatgcagctt

S_5 tgaaaagaatttacggtaaaggatccacatccaatcgtgtgaaag

Demonstration of algorithm

EMS-GT



S_1 atcactcgttctcctctaatgtgtaaagacgtactaccgacctta

S_2 acgccgaccggtccgatccttgtatagctcctaacgggcatcagc

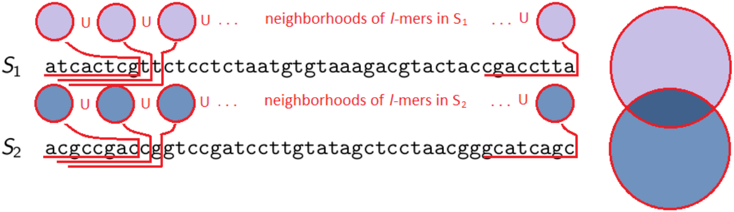
S_3 tcctgactgcatcgcgatctcggtagtttcctgttcatcattttt

S_4 ggccctcagcatcgtgcgtcctgctaacacattcccatgcagctt

S_5 tgaaaagaatttacggtaaaggatccacatccaatcgtgtgaaag

Demonstration of algorithm

EMS-GT



$\cup$ $\cup$ $\cup$ . . . neighborhoods of l-mers in S_1 . . . $\cup$

S_1 atcactcgttctcctctaatgtgtaaagacgtactaccgacctta

$\cup$ $\cup$ $\cup$ . . . neighborhoods of l-mers in S_2 . . . $\cup$

S_2 acgccgacggtccgatccttgtatagctcctaacgggcatcagc

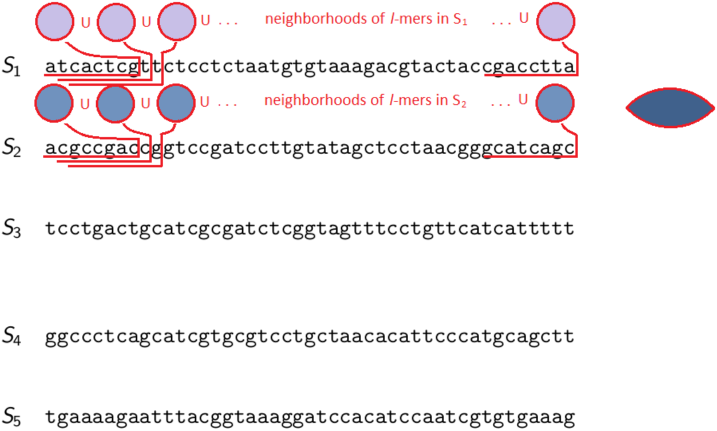
S_3 tcctgactgcatcgcgatctcggtagtttcctgttcatcatttt

S_4 ggccctcagcatcgtgcgtcctgctaacacattcccatgcagctt

S_5 tgaaaagaatttacggtaaaggatccacatccaatcgtgtgaaag

Demonstration of algorithm

EMS-GT


∪ ∪ ∪ . . . neighborhoods of *l*-mers in S_1 . . . ∪
∪ ∪ ∪ . . . neighborhoods of *l*-mers in S_2 . . . ∪

S_1 atcactcgttctcctctaatgtgtaaagacgtactaccgacctta

S_2 acgccgaccggtccgatccttgtatagctcctaacgggcatcagc

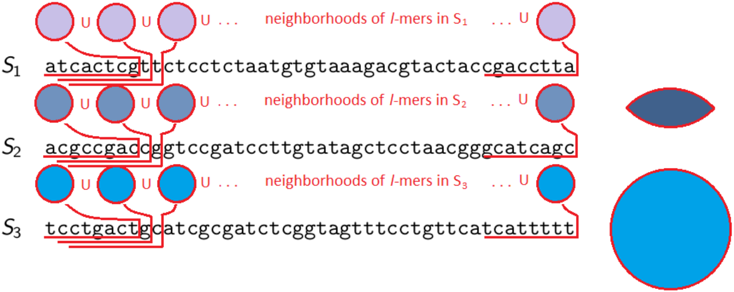
S_3 tcctgactgcatcgcgatctcggtagtttcctgttcatcatttt

S_4 ggccctcagcatcgtgcgtcctgctaacacattcccatgcagctt

S_5 tgaaaagaatttacggtaaaggatccacatccaatcgtgtgaaag

Demonstration of algorithm

EMS-GT



∪ ∪ ∪ . . . neighborhoods of l-mers in S_1 . . . ∪
∪ ∪ ∪ . . . neighborhoods of l-mers in S_2 . . . ∪
∪ ∪ ∪ . . . neighborhoods of l-mers in S_3 . . . ∪

S_1 atcactcgttctcctctaatgtgtaaagacgtactaccgacctta

S_2 acgccgaccggtccgatccttgtatagctcctaacgggcatcagc

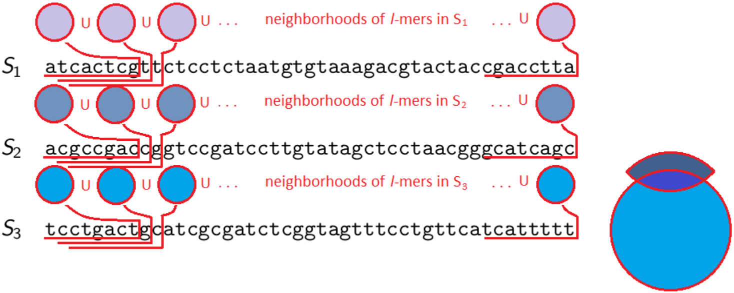
S_3 tcctgactgcatcgcgatctcggtagtttcctgttcatcattttt

S_4 ggccctcagcatcgtgcgtcctgctaacacattcccatgcagctt

S_5 tgaaaagaatttacggtaaaggatccacatccaatcgtgtgaaag

Demonstration of algorithm

EMS-GT



$\cup$ $\cup$ $\cup$ . . . neighborhoods of l-mers in S_1 . . . $\cup$

S_1 atcactcgttctcctctaatgtgtaaagacgtactaccgacctta

$\cup$ $\cup$ $\cup$ . . . neighborhoods of l-mers in S_2 . . . $\cup$

S_2 acgccgacggtccgatccttgtatagctcctaacgggcatcagc

$\cup$ $\cup$ $\cup$ . . . neighborhoods of l-mers in S_3 . . . $\cup$

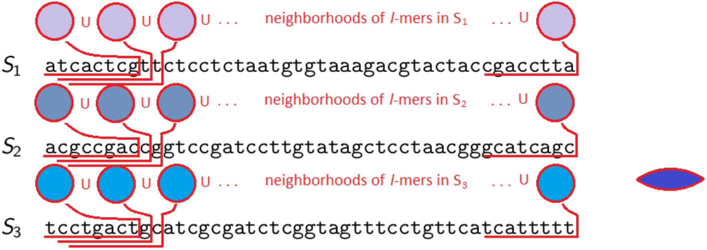
S_3 tcctgactgcatcgcgatctcggtagtttcctgttcatcatttt

S_4 ggccctcagcatcgtgcgtcctgctaacacattcccatgcagctt

S_5 tgaaaagaatttacggtaaaggatccacatccaatcgtgtgaaag

Demonstration of algorithm

EMS-GT



U U U . . . neighborhoods of l-mers in S_1 . . . U

S_1 atcactcgttctcctctaatgtgtaaagacgtactaccgacctta

U U U . . . neighborhoods of l-mers in S_2 . . . U

S_2 acgccgacggtccgatccttgtatagctcctaacgggcatcagc

U U U . . . neighborhoods of l-mers in S_3 . . . U

S_3 tcctgactgcatcgcgatctcggtagtttcctgttcatcatttt

S_4 ggccctcagcatcgtgcgtcctgctaacacattcccatgcagctt

S_5 tgaaaagaatttacggtaaaggatccacatccaatcgtgtgaaag

Representing sets with bits

EMS-GT

- ex. the *d*-neighborhood of acgt, for $d{=}2$ (67 neighbors)

acgt,

ccgt, gcgt, tcgt, aagt, aggt, atgt,

acat, acgt, actt, acga, acgc, acgg,

cagt, cggt, ctgt, ccat, ccct, cctt, ccga, ccgc, ccgg,

gagt, gggt, gtgt, gcat, gcct, gctt, gcga, gcgc, gcgg,

tagt, tggt, ttgt, tcat, tcct, tctt, tcga, tcgc, tcgg,

aaat, aact, aatt, aaga, aagc, aagg, agat, agct, agtt,

agga, aggc, aggg, atat, atct, attt, atga, atgc, atgg,

acaa, acac, acag, acca, accc, accg, acta, actc, actg.

Representing sets with bits

EMS-GT

- We know there are only 4^l possible l-mers that can be formed with the DNA bases {`a`, `c`, `g`, `t`};

Representing sets with bits

EMS-GT

- We know there are only 4^l possible l-mers that can be formed with the DNA bases {a, c, g, t};
- thus, EMS-GT can represent any set of l-mers with 4^l bits:
  - set to 1 if the corresponding l-mer is a member of the set,
  - set to 0 otherwise.

Representing sets with bits

EMS-GT

- We know there are only 4^l possible l-mers that can be formed with the DNA bases {a, c, g, t};
- thus, EMS-GT can represent any set of l-mers with 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.

Representing sets with bits

EMS-GT

- ex. the *d*-neighborhood of acgt, for $d = 2$ (67 neighbors)

acgt,

ccgt, gcgt, tcgt, aagt, aggt, atgt,

acat, acgt, actt, acga, acgc, acgg,

cagt, cggt, ctgt, ccat, ccct, cctt, ccga, ccgc, ccgg,

gagt, gggt, gtgt, gcat, gcct, gctt, gcga, gcgc, gcgg,

tagt, tggt, ttgt, tcat, tcct, tctt, tcga, tcgc, tcgg,

aaat, aact, aatt, aaga, aagc, aagg, agat, agct, agtt,

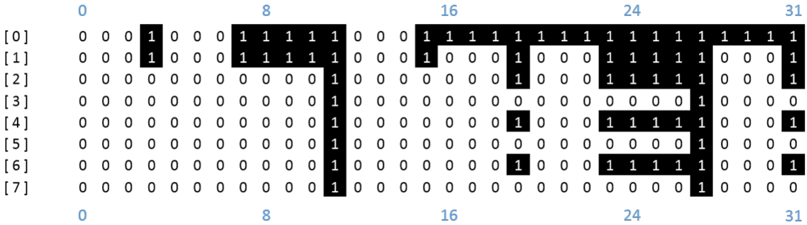
agga, aggc, aggg, atat, atct, attt, atga, atgc, atgg,

acaa, acac, acag, acca, accc, accg, acta, actc, actg.

Representing sets with bits

EMS-GT

- ex. the *d*-neighborhood of `acgt`, for $d{=}2$ (67 neighbors)
- $l{=}4$

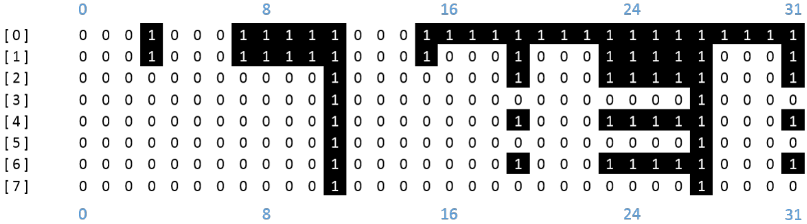


	0								8								16								24							31
[0]	0	0	0	1	0	0	0	1	1	1	1	1	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
[1]	0	0	0	1	0	0	0	1	1	1	1	1	0	0	0	1	0	0	0	1	0	0	0	1	1	1	1	1	0	0	0	1
[2]	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	1	1	1	1	1	0	0	0	1
[3]	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	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

Representing sets with bits

EMS-GT

- ex. the d-neighborhood of acgt, for $d{=}2$ (67 neighbors)
- $l{=}4 \;\rightarrow\; 4^l = 256$ possible l-mers

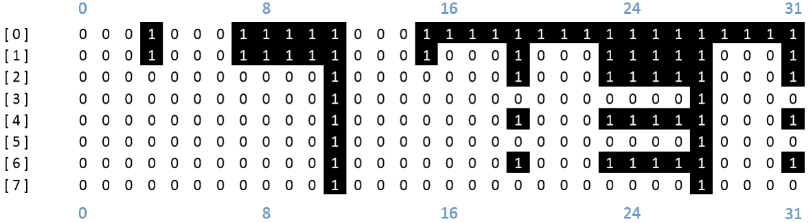


	0								8								16								24							31
[0]	0	0	0	1	0	0	0	1	1	1	1	1	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
[1]	0	0	0	1	0	0	0	1	1	1	1	1	0	0	0	1	0	0	0	1	0	0	0	1	1	1	1	1	0	0	0	1
[2]	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	1	1	1	1	1	0	0	0	1
[3]	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	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

Representing sets with bits

EMS-GT

- ex. the *d*-neighborhood of acgt, for $d=2$ (67 neighbors)
- $l=4 \;\rightarrow\; 4^l = 256$ possible *l*-mers $\;\rightarrow\; \frac{4^l}{32} = 8$ 32-bit integers

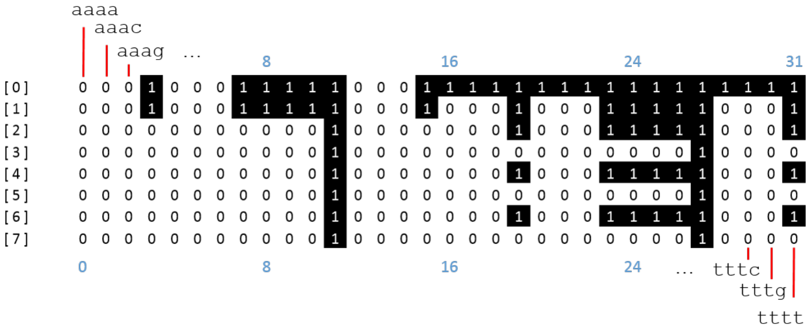


	0								8								16								24							31
[0]	0	0	0	1	0	0	0	1	1	1	1	1	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
[1]	0	0	0	1	0	0	0	1	1	1	1	1	0	0	0	1	0	0	0	1	0	0	0	1	1	1	1	1	0	0	0	1
[2]	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	1	1	1	1	1	0	0	0	1
[3]	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	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

Representing sets with bits

EMS-GT

- ex. the d-neighborhood of acgt, for $d{=}2$ (67 neighbors)
- $l{=}4 \;\rightarrow\; 4^l = 256$ possible l-mers $\;\rightarrow\; \frac{4^l}{32} = 8$ 32-bit integers

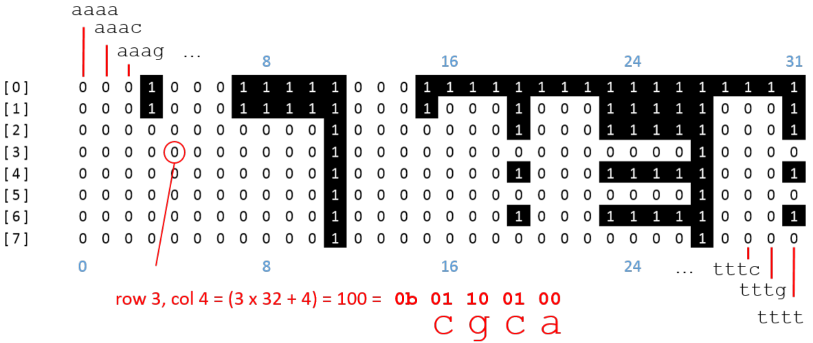


	0								8								16								24							31
[0]	0	0	0	1	0	0	0	1	1	1	1	1	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
[1]	0	0	0	1	0	0	0	1	1	1	1	1	0	0	0	1	0	0	0	1	0	0	0	1	1	1	1	1	0	0	0	1
[2]	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	1	1	1	1	1	0	0	0	1
[3]	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	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							

Representing sets with bits

EMS-GT

- ex. the d-neighborhood of acgt, for d=2 (67 neighbors)
- l=4 $\rightarrow$ 4^l = 256 possible l-mers $\rightarrow$ $\frac{4^l}{32}$ = 8 32-bit integers



									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

Building neighborhoods in blocks

EMS-GT

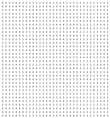
Given the neighborhood bit-array N_x for l-mer x,
if we partition N_x into blocks of 4^k bits each, $k < l$,
each block conforms to one of $(k + 2)$ patterns.

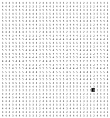
Building neighborhoods in blocks

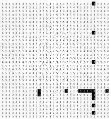
EMS-GT

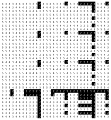
Given the neighborhood bit-array N_x for l-mer x,
if we partition N_x into blocks of 4^k bits each, $k < l$,
each block conforms to one of $(k+2)$ patterns.

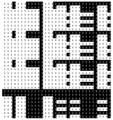
ex. patterns in $N_{\texttt{acgtacgtacgt}}$ for $k{=}5$ → $4^5{=}32{\times}32$ bits per block

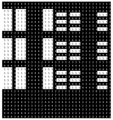

Pattern -1

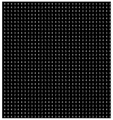

Pattern 0


Pattern 1


Pattern 2


Pattern 3


Pattern 4

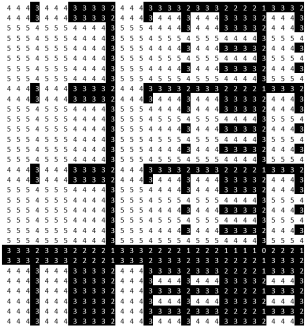

Pattern 5

Building neighborhoods in blocks

EMS-GT

- **prefix** = first (l-k) characters, k-suffix = last k characters

acgtaaaaaaaa →



4	4	4	3	4	4	4	3	3	3	3	2	4	4	4	3	3	3	3	2	3	3	3	2	2	2	2	1	3	3	3	2
4	4	4	3	4	4	4	3	3	3	3	2	4	4	4	3	4	4	4	3	4	4	4	3	3	3	3	2	4	4	4	3
5	5	5	4	5	5	5	4	4	4	4	3	5	5	5	4	4	4	4	3	4	4	4	3	3	3	3	2	4	4	4	3
5	5	5	4	5	5	5	4	4	4	4	3	5	5	5	4	5	5	5	4	5	5	5	4	4	4	4	3	5	5	5	4
5	5	5	4	5	5	5	4	4	4	4	3	5	5	5	4	4	4	4	3	4	4	4	3	3	3	3	2	4	4	4	3
5	5	5	4	5	5	5	4	4	4	4	3	5	5	5	4	5	5	5	4	5	5	5	4	4	4	4	3	5	5	5	4
5	5	5	4	5	5	5	4	4	4	4	3	5	5	5	4	4	4	4	3	4	4	4	3	3	3	3	2	4	4	4	3
5	5	5	4	5	5	5	4	4	4	4	3	5	5	5	4	5	5	5	4	5	5	5	4	4	4	4	3	5	5	5	4
4	4	4	3	4	4	4	3	3	3	3	2	4	4	4	3	3	3	3	2	3	3	3	2	2	2	2	1	3	3	3	2
4	4	4	3	4	4	4	3	3	3	3	2	4	4	4	3	4	4	4	3	4	4	4	3	3	3	3	2	4	4	4	3
5	5	5	4	5	5	5	4	4	4	4	3	5	5	5	4	4	4	4	3	4	4	4	3	3	3	3	2	4	4	4	3
5	5	5	4	5	5	5	4	4	4	4	3	5	5	5	4	5	5	5	4	5	5	5	4	4	4	4	3	5	5	5	4
5	5	5	4	5	5	5	4	4	4	4	3	5	5	5	4	4	4	4	3	4	4	4	3	3	3	3	2	4	4	4	3
5	5	5	4	5	5	5	4	4	4	4	3	5	5	5	4	5	5	5	4	5	5	5	4	4	4	4	3	5	5	5	4
5	5	5	4	5	5	5	4	4	4	4	3	5	5	5	4	4	4	4	3	4	4	4	3	3	3	3	2	4	4	4	3
5	5	5	4	5	5	5	4	4	4	4	3	5	5	5	4	5	5	5	4	5	5	5	4	4	4	4	3	5	5	5	4
4	4	4	3	4	4	4	3	3	3	3	2	4	4	4	3	3	3	3	2	3	3	3	2	2	2	2	1	3	3	3	2
4	4	4	3	4	4	4	3	3	3	3	2	4	4	4	3	4	4	4	3	4	4	4	3	3	3	3	2	4	4	4	3
5	5	5	4	5	5	5	4	4	4	4	3	5	5	5	4	4	4	4	3	4	4	4	3	3	3	3	2	4	4	4	3
5	5	5	4	5	5	5	4	4	4	4	3	5	5	5	4	5	5	5	4	5	5	5	4	4	4	4	3	5	5	5	4
5	5	5	4	5	5	5	4	4	4	4	3	5	5	5	4	4	4	4	3	4	4	4	3	3	3	3	2	4	4	4	3
5	5	5	4	5	5	5	4	4	4	4	3	5	5	5	4	5	5	5	4	5	5	5	4	4	4	4	3	5	5	5	4
5	5	5	4	5	5	5	4	4	4	4	3	5	5	5	4	4	4	4	3	4	4	4	3	3	3	3	2	4	4	4	3
5	5	5	4	5	5	5	4	4	4	4	3	5	5	5	4	5	5	5	4	5	5	5	4	4	4	4	3	5	5	5	4
3	3	3	2	3	3	3	2	2	2	2	1	3	3	3	2	2	2	2	1	2	2	2	1	1	1	1	0	2	2	2	1
3	3	3	2	3	3	3	2	2	2	2	1	3	3	3	2	3	3	3	2	3	3	3	2	2	2	2	1	3	3	3	2
4	4	4	3	4	4	4	3	3	3	3	2	4	4	4	3	3	3	3	2	3	3	3	2	2	2	2	1	3	3	3	2
4	4	4	3	4	4	4	3	3	3	3	2	4	4	4	3	4	4	4	3	4	4	4	3	3	3	3	2	4	4	4	3
4	4	4	3	4	4	4	3	3	3	3	2	4	4	4	3	3	3	3	2	3	3	3	2	2	2	2	1	3	3	3	2
4	4	4	3	4	4	4	3	3	3	3	2	4	4	4	3	4	4	4	3	4	4	4	3	3	3	3	2	4	4	4	3
4	4	4	3	4	4	4	3	3	3	3	2	4	4	4	3	3	3	3	2	3	3	3	2	2	2	2	1	3	3	3	2
4	4	4	3	4	4	4	3	3	3	3	2	4	4	4	3	4	4	4	3	4	4	4	3	3	3	3	2	4	4	4	3

← **acgtaaa**ttttt

- due to the alphabetical ordering, l-mers in the same block all have the same prefix, and differ only in their k-suffixes

Building neighborhoods in blocks

EMS-GT

For the neighborhood N_x of l-mer x,

- x's **prefix** determines which patterns apply to which blocks;
- x's k-suffix determines the structure of the patterns

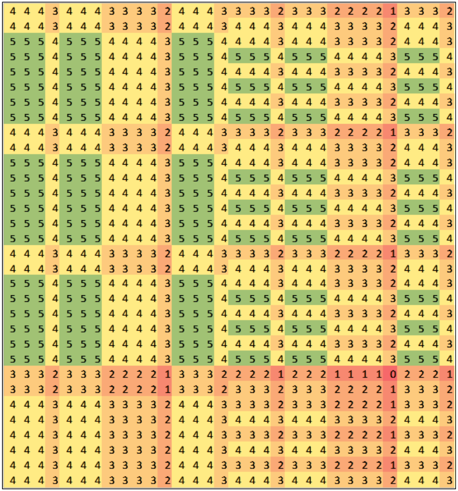
Building neighborhoods in blocks

EMS-GT

- ex. d=5, k=5

  $x = \texttt{acgtacgtacgt}$

- color map: number of mismatches from x's suffix `tacgt` of all possible k-suffixes



4	4	4	3	4	4	4	3	3	3	3	2	4	4	4	3	3	3	3	2	3	3	3	2	2	2	2	1	3	3	3	2
4	4	4	3	4	4	4	3	3	3	3	2	4	4	4	3	4	4	4	3	4	4	4	3	3	3	3	2	4	4	4	3
5	5	5	4	5	5	5	4	4	4	4	3	5	5	5	4	4	4	4	3	4	4	4	3	3	3	3	2	4	4	4	3
5	5	5	4	5	5	5	4	4	4	4	3	5	5	5	4	5	5	5	4	5	5	5	4	4	4	4	3	5	5	5	4
5	5	5	4	5	5	5	4	4	4	4	3	5	5	5	4	4	4	4	3	4	4	4	3	3	3	3	2	4	4	4	3
5	5	5	4	5	5	5	4	4	4	4	3	5	5	5	4	5	5	5	4	5	5	5	4	4	4	4	3	5	5	5	4
5	5	5	4	5	5	5	4	4	4	4	3	5	5	5	4	4	4	4	3	4	4	4	3	3	3	3	2	4	4	4	3
5	5	5	4	5	5	5	4	4	4	4	3	5	5	5	4	5	5	5	4	5	5	5	4	4	4	4	3	5	5	5	4
4	4	4	3	4	4	4	3	3	3	3	2	4	4	4	3	3	3	3	2	3	3	3	2	2	2	2	1	3	3	3	2
4	4	4	3	4	4	4	3	3	3	3	2	4	4	4	3	4	4	4	3	4	4	4	3	3	3	3	2	4	4	4	3
5	5	5	4	5	5	5	4	4	4	4	3	5	5	5	4	4	4	4	3	4	4	4	3	3	3	3	2	4	4	4	3
5	5	5	4	5	5	5	4	4	4	4	3	5	5	5	4	5	5	5	4	5	5	5	4	4	4	4	3	5	5	5	4
5	5	5	4	5	5	5	4	4	4	4	3	5	5	5	4	4	4	4	3	4	4	4	3	3	3	3	2	4	4	4	3
5	5	5	4	5	5	5	4	4	4	4	3	5	5	5	4	5	5	5	4	5	5	5	4	4	4	4	3	5	5	5	4
5	5	5	4	5	5	5	4	4	4	4	3	5	5	5	4	4	4	4	3	4	4	4	3	3	3	3	2	4	4	4	3
5	5	5	4	5	5	5	4	4	4	4	3	5	5	5	4	5	5	5	4	5	5	5	4	4	4	4	3	5	5	5	4
4	4	4	3	4	4	4	3	3	3	3	2	4	4	4	3	3	3	3	2	3	3	3	2	2	2	2	1	3	3	3	2
4	4	4	3	4	4	4	3	3	3	3	2	4	4	4	3	4	4	4	3	4	4	4	3	3	3	3	2	4	4	4	3
5	5	5	4	5	5	5	4	4	4	4	3	5	5	5	4	4	4	4	3	4	4	4	3	3	3	3	2	4	4	4	3
5	5	5	4	5	5	5	4	4	4	4	3	5	5	5	4	5	5	5	4	5	5	5	4	4	4	4	3	5	5	5	4
5	5	5	4	5	5	5	4	4	4	4	3	5	5	5	4	4	4	4	3	4	4	4	3	3	3	3	2	4	4	4	3
5	5	5	4	5	5	5	4	4	4	4	3	5	5	5	4	5	5	5	4	5	5	5	4	4	4	4	3	5	5	5	4
5	5	5	4	5	5	5	4	4	4	4	3	5	5	5	4	4	4	4	3	4	4	4	3	3	3	3	2	4	4	4	3
5	5	5	4	5	5	5	4	4	4	4	3	5	5	5	4	5	5	5	4	5	5	5	4	4	4	4	3	5	5	5	4
3	3	3	2	3	3	3	2	2	2	2	1	3	3	3	2	2	2	2	1	2	2	2	1	1	1	1	0	2	2	2	1
3	3	3	2	3	3	3	2	2	2	2	1	3	3	3	2	3	3	3	2	3	3	3	2	2	2	2	1	3	3	3	2
4	4	4	3	4	4	4	3	3	3	3	2	4	4	4	3	3	3	3	2	3	3	3	2	2	2	2	1	3	3	3	2
4	4	4	3	4	4	4	3	3	3	3	2	4	4	4	3	4	4	4	3	4	4	4	3	3	3	3	2	4	4	4	3
4	4	4	3	4	4	4	3	3	3	3	2	4	4	4	3	3	3	3	2	3	3	3	2	2	2	2	1	3	3	3	2
4	4	4	3	4	4	4	3	3	3	3	2	4	4	4	3	4	4	4	3	4	4	4	3	3	3	3	2	4	4	4	3
4	4	4	3	4	4	4	3	3	3	3	2	4	4	4	3	3	3	3	2	3	3	3	2	2	2	2	1	3	3	3	2
4	4	4	3	4	4	4	3	3	3	3	2	4	4	4	3	4	4	4	3	4	4	4	3	3	3	3	2	4	4	4	3

Building neighborhoods in blocks

EMS-GT

- ex. d=5, k=5

  x = acgtacgtacgt

- if prefix = cgacatc

  (6 mismatches from acgtacg)

- we cannot use any k-suffix to form a neighbor of x

```
4 4 4 3 4 4 4 3 3 3 3 2 4 4 4 3 3 3 3 2 3 3 3 2 2 2 2 1 3 3 3 2
4 4 4 3 4 4 4 3 3 3 3 2 4 4 4 3 4 4 4 3 4 4 4 3 3 3 3 2 4 4 4 3
5 5 5 4 5 5 5 4 4 4 4 3 5 5 5 4 4 4 4 3 4 4 4 3 3 3 3 2 4 4 4 3
5 5 5 4 5 5 5 4 4 4 4 3 5 5 5 4 5 5 5 4 5 5 5 4 4 4 4 3 5 5 5 4
5 5 5 4 5 5 5 4 4 4 4 3 5 5 5 4 4 4 4 3 4 4 4 3 3 3 3 2 4 4 4 3
5 5 5 4 5 5 5 4 4 4 4 3 5 5 5 4 5 5 5 4 5 5 5 4 4 4 4 3 5 5 5 4
5 5 5 4 5 5 5 4 4 4 4 3 5 5 5 4 4 4 4 3 4 4 4 3 3 3 3 2 4 4 4 3
5 5 5 4 5 5 5 4 4 4 4 3 5 5 5 4 5 5 5 4 5 5 5 4 4 4 4 3 5 5 5 4
4 4 4 3 4 4 4 3 3 3 3 2 4 4 4 3 3 3 3 2 3 3 3 2 2 2 2 1 3 3 3 2
4 4 4 3 4 4 4 3 3 3 3 2 4 4 4 3 4 4 4 3 4 4 4 3 3 3 3 2 4 4 4 3
5 5 5 4 5 5 5 4 4 4 4 3 5 5 5 4 4 4 4 3 4 4 4 3 3 3 3 2 4 4 4 3
5 5 5 4 5 5 5 4 4 4 4 3 5 5 5 4 5 5 5 4 5 5 5 4 4 4 4 3 5 5 5 4
5 5 5 4 5 5 5 4 4 4 4 3 5 5 5 4 4 4 4 3 4 4 4 3 3 3 3 2 4 4 4 3
5 5 5 4 5 5 5 4 4 4 4 3 5 5 5 4 5 5 5 4 5 5 5 4 4 4 4 3 5 5 5 4
5 5 5 4 5 5 5 4 4 4 4 3 5 5 5 4 4 4 4 3 4 4 4 3 3 3 3 2 4 4 4 3
5 5 5 4 5 5 5 4 4 4 4 3 5 5 5 4 5 5 5 4 5 5 5 4 4 4 4 3 5 5 5 4
4 4 4 3 4 4 4 3 3 3 3 2 4 4 4 3 3 3 3 2 3 3 3 2 2 2 2 1 3 3 3 2
4 4 4 3 4 4 4 3 3 3 3 2 4 4 4 3 4 4 4 3 4 4 4 3 3 3 3 2 4 4 4 3
5 5 5 4 5 5 5 4 4 4 4 3 5 5 5 4 4 4 4 3 4 4 4 3 3 3 3 2 4 4 4 3
5 5 5 4 5 5 5 4 4 4 4 3 5 5 5 4 5 5 5 4 5 5 5 4 4 4 4 3 5 5 5 4
5 5 5 4 5 5 5 4 4 4 4 3 5 5 5 4 4 4 4 3 4 4 4 3 3 3 3 2 4 4 4 3
5 5 5 4 5 5 5 4 4 4 4 3 5 5 5 4 5 5 5 4 5 5 5 4 4 4 4 3 5 5 5 4
5 5 5 4 5 5 5 4 4 4 4 3 5 5 5 4 4 4 4 3 4 4 4 3 3 3 3 2 4 4 4 3
5 5 5 4 5 5 5 4 4 4 4 3 5 5 5 4 5 5 5 4 5 5 5 4 4 4 4 3 5 5 5 4
3 3 3 2 3 3 3 2 2 2 2 1 3 3 3 2 2 2 2 1 2 2 2 1 1 1 1 0 2 2 2 1
3 3 3 2 3 3 3 2 2 2 2 1 3 3 3 2 3 3 3 2 3 3 3 2 2 2 2 1 3 3 3 2
4 4 4 3 4 4 4 3 3 3 3 2 4 4 4 3 3 3 3 2 3 3 3 2 2 2 2 1 3 3 3 2
4 4 4 3 4 4 4 3 3 3 3 2 4 4 4 3 4 4 4 3 4 4 4 3 3 3 3 2 4 4 4 3
4 4 4 3 4 4 4 3 3 3 3 2 4 4 4 3 3 3 3 2 3 3 3 2 2 2 2 1 3 3 3 2
4 4 4 3 4 4 4 3 3 3 3 2 4 4 4 3 4 4 4 3 4 4 4 3 3 3 3 2 4 4 4 3
4 4 4 3 4 4 4 3 3 3 3 2 4 4 4 3 3 3 3 2 3 3 3 2 2 2 2 1 3 3 3 2
4 4 4 3 4 4 4 3 3 3 3 2 4 4 4 3 4 4 4 3 4 4 4 3 3 3 3 2 4 4 4 3
```

Building neighborhoods in blocks

EMS-GT

- ex. d=5, k=5

  x = acgtacgtacgt

- if prefix = agacatc

  (5 mismatches from acgtacg)

- adding the k-suffix tacgt

  forms a neighbor of x

```
4 4 4 3 4 4 4 3 3 3 3 2 4 4 4 3 3 3 3 2 3 3 3 2 2 2 2 1 3 3 3 2
4 4 4 3 4 4 4 3 3 3 3 2 4 4 4 3 4 4 4 3 4 4 4 3 3 3 3 2 4 4 4 3
5 5 5 4 5 5 5 4 4 4 4 3 5 5 5 4 4 4 4 3 4 4 4 3 3 3 3 2 4 4 4 3
5 5 5 4 5 5 5 4 4 4 4 3 5 5 5 4 5 5 5 4 5 5 5 4 4 4 4 3 5 5 5 4
5 5 5 4 5 5 5 4 4 4 4 3 5 5 5 4 4 4 4 3 4 4 4 3 3 3 3 2 4 4 4 3
5 5 5 4 5 5 5 4 4 4 4 3 5 5 5 4 5 5 5 4 5 5 5 4 4 4 4 3 5 5 5 4
5 5 5 4 5 5 5 4 4 4 4 3 5 5 5 4 4 4 4 3 4 4 4 3 3 3 3 2 4 4 4 3
5 5 5 4 5 5 5 4 4 4 4 3 5 5 5 4 5 5 5 4 5 5 5 4 4 4 4 3 5 5 5 4
4 4 4 3 4 4 4 3 3 3 3 2 4 4 4 3 3 3 3 2 3 3 3 2 2 2 2 1 3 3 3 2
4 4 4 3 4 4 4 3 3 3 3 2 4 4 4 3 4 4 4 3 4 4 4 3 3 3 3 2 4 4 4 3
5 5 5 4 5 5 5 4 4 4 4 3 5 5 5 4 4 4 4 3 4 4 4 3 3 3 3 2 4 4 4 3
5 5 5 4 5 5 5 4 4 4 4 3 5 5 5 4 5 5 5 4 5 5 5 4 4 4 4 3 5 5 5 4
5 5 5 4 5 5 5 4 4 4 4 3 5 5 5 4 4 4 4 3 4 4 4 3 3 3 3 2 4 4 4 3
5 5 5 4 5 5 5 4 4 4 4 3 5 5 5 4 5 5 5 4 5 5 5 4 4 4 4 3 5 5 5 4
5 5 5 4 5 5 5 4 4 4 4 3 5 5 5 4 4 4 4 3 4 4 4 3 3 3 3 2 4 4 4 3
5 5 5 4 5 5 5 4 4 4 4 3 5 5 5 4 5 5 5 4 5 5 5 4 4 4 4 3 5 5 5 4
4 4 4 3 4 4 4 3 3 3 3 2 4 4 4 3 3 3 3 2 3 3 3 2 2 2 2 1 3 3 3 2
4 4 4 3 4 4 4 3 3 3 3 2 4 4 4 3 4 4 4 3 4 4 4 3 3 3 3 2 4 4 4 3
5 5 5 4 5 5 5 4 4 4 4 3 5 5 5 4 4 4 4 3 4 4 4 3 3 3 3 2 4 4 4 3
5 5 5 4 5 5 5 4 4 4 4 3 5 5 5 4 5 5 5 4 5 5 5 4 4 4 4 3 5 5 5 4
5 5 5 4 5 5 5 4 4 4 4 3 5 5 5 4 4 4 4 3 4 4 4 3 3 3 3 2 4 4 4 3
5 5 5 4 5 5 5 4 4 4 4 3 5 5 5 4 5 5 5 4 5 5 5 4 4 4 4 3 5 5 5 4
5 5 5 4 5 5 5 4 4 4 4 3 5 5 5 4 4 4 4 3 4 4 4 3 3 3 3 2 4 4 4 3
5 5 5 4 5 5 5 4 4 4 4 3 5 5 5 4 5 5 5 4 5 5 5 4 4 4 4 3 5 5 5 4
3 3 3 2 3 3 3 2 2 2 2 1 3 3 3 2 2 2 2 1 2 2 2 1 1 1 1 0 2 2 2 1
3 3 3 2 3 3 3 2 2 2 2 1 3 3 3 2 3 3 3 2 3 3 3 2 2 2 2 1 3 3 3 2
4 4 4 3 4 4 4 3 3 3 3 2 4 4 4 3 3 3 3 2 3 3 3 2 2 2 2 1 3 3 3 2
4 4 4 3 4 4 4 3 3 3 3 2 4 4 4 3 4 4 4 3 4 4 4 3 3 3 3 2 4 4 4 3
4 4 4 3 4 4 4 3 3 3 3 2 4 4 4 3 3 3 3 2 3 3 3 2 2 2 2 1 3 3 3 2
4 4 4 3 4 4 4 3 3 3 3 2 4 4 4 3 4 4 4 3 4 4 4 3 3 3 3 2 4 4 4 3
4 4 4 3 4 4 4 3 3 3 3 2 4 4 4 3 3 3 3 2 3 3 3 2 2 2 2 1 3 3 3 2
4 4 4 3 4 4 4 3 3 3 3 2 4 4 4 3 4 4 4 3 4 4 4 3 3 3 3 2 4 4 4 3
```

Building neighborhoods in blocks

EMS-GT

- ex. d=5, k=5

  x = acgtacgtacgt

- if prefix = agacatg

  (4 mismatches from acgtacg)

- any k-suffix with up to 1 mismatch from tacgt forms a neighbor of x

4	4	4	3	4	4	4	3	3	3	3	2	4	4	4	3	3	3	3	2	3	3	3	2	2	2	2	1	3	3	3	2
4	4	4	3	4	4	4	3	3	3	3	2	4	4	4	3	4	4	4	3	4	4	4	3	3	3	3	2	4	4	4	3
5	5	5	4	5	5	5	4	4	4	4	3	5	5	5	4	4	4	4	3	4	4	4	3	3	3	3	2	4	4	4	3
5	5	5	4	5	5	5	4	4	4	4	3	5	5	5	4	5	5	5	4	5	5	5	4	4	4	4	3	5	5	5	4
5	5	5	4	5	5	5	4	4	4	4	3	5	5	5	4	4	4	4	3	4	4	4	3	3	3	3	2	4	4	4	3
5	5	5	4	5	5	5	4	4	4	4	3	5	5	5	4	5	5	5	4	5	5	5	4	4	4	4	3	5	5	5	4
5	5	5	4	5	5	5	4	4	4	4	3	5	5	5	4	4	4	4	3	4	4	4	3	3	3	3	2	4	4	4	3
5	5	5	4	5	5	5	4	4	4	4	3	5	5	5	4	5	5	5	4	5	5	5	4	4	4	4	3	5	5	5	4
4	4	4	3	4	4	4	3	3	3	3	2	4	4	4	3	3	3	3	2	3	3	3	2	2	2	2	1	3	3	3	2
4	4	4	3	4	4	4	3	3	3	3	2	4	4	4	3	4	4	4	3	4	4	4	3	3	3	3	2	4	4	4	3
5	5	5	4	5	5	5	4	4	4	4	3	5	5	5	4	4	4	4	3	4	4	4	3	3	3	3	2	4	4	4	3
5	5	5	4	5	5	5	4	4	4	4	3	5	5	5	4	5	5	5	4	5	5	5	4	4	4	4	3	5	5	5	4
5	5	5	4	5	5	5	4	4	4	4	3	5	5	5	4	4	4	4	3	4	4	4	3	3	3	3	2	4	4	4	3
5	5	5	4	5	5	5	4	4	4	4	3	5	5	5	4	5	5	5	4	5	5	5	4	4	4	4	3	5	5	5	4
5	5	5	4	5	5	5	4	4	4	4	3	5	5	5	4	4	4	4	3	4	4	4	3	3	3	3	2	4	4	4	3
5	5	5	4	5	5	5	4	4	4	4	3	5	5	5	4	5	5	5	4	5	5	5	4	4	4	4	3	5	5	5	4
4	4	4	3	4	4	4	3	3	3	3	2	4	4	4	3	3	3	3	2	3	3	3	2	2	2	2	1	3	3	3	2
4	4	4	3	4	4	4	3	3	3	3	2	4	4	4	3	4	4	4	3	4	4	4	3	3	3	3	2	4	4	4	3
5	5	5	4	5	5	5	4	4	4	4	3	5	5	5	4	4	4	4	3	4	4	4	3	3	3	3	2	4	4	4	3
5	5	5	4	5	5	5	4	4	4	4	3	5	5	5	4	5	5	5	4	5	5	5	4	4	4	4	3	5	5	5	4
5	5	5	4	5	5	5	4	4	4	4	3	5	5	5	4	4	4	4	3	4	4	4	3	3	3	3	2	4	4	4	3
5	5	5	4	5	5	5	4	4	4	4	3	5	5	5	4	5	5	5	4	5	5	5	4	4	4	4	3	5	5	5	4
5	5	5	4	5	5	5	4	4	4	4	3	5	5	5	4	4	4	4	3	4	4	4	3	3	3	3	2	4	4	4	3
5	5	5	4	5	5	5	4	4	4	4	3	5	5	5	4	5	5	5	4	5	5	5	4	4	4	4	3	5	5	5	4
3	3	3	2	3	3	3	2	2	2	2	1	3	3	3	2	2	2	2	1	2	2	2	1	1	1	1	0	2	2	2	1
3	3	3	2	3	3	3	2	2	2	2	1	3	3	3	2	3	3	3	2	3	3	3	2	2	2	2	1	3	3	3	2
4	4	4	3	4	4	4	3	3	3	3	2	4	4	4	3	3	3	3	2	3	3	3	2	2	2	2	1	3	3	3	2
4	4	4	3	4	4	4	3	3	3	3	2	4	4	4	3	4	4	4	3	4	4	4	3	3	3	3	2	4	4	4	3
4	4	4	3	4	4	4	3	3	3	3	2	4	4	4	3	3	3	3	2	3	3	3	2	2	2	2	1	3	3	3	2
4	4	4	3	4	4	4	3	3	3	3	2	4	4	4	3	4	4	4	3	4	4	4	3	3	3	3	2	4	4	4	3
4	4	4	3	4	4	4	3	3	3	3	2	4	4	4	3	3	3	3	2	3	3	3	2	2	2	2	1	3	3	3	2
4	4	4	3	4	4	4	3	3	3	3	2	4	4	4	3	4	4	4	3	4	4	4	3	3	3	3	2	4	4	4	3

Building neighborhoods in blocks

EMS-GT

- ex. d=5, k=5

  x = acgtacgtacgt

- if prefix = agatatg

  (3 mismatches from acgtacg)

- any k-suffix with up to

  2 mismatches from tacgt

  forms a neighbor of x

4	4	4	3	4	4	4	3	3	3	3	2	4	4	4	3	3	3	3	2	3	3	3	2	2	2	2	1	3	3	3	2
4	4	4	3	4	4	4	3	3	3	3	2	4	4	4	3	4	4	4	3	4	4	4	3	3	3	3	2	4	4	4	3
5	5	5	4	5	5	5	4	4	4	4	3	5	5	5	4	4	4	4	3	4	4	4	3	3	3	3	2	4	4	4	3
5	5	5	4	5	5	5	4	4	4	4	3	5	5	5	4	5	5	5	4	5	5	5	4	4	4	4	3	5	5	5	4
5	5	5	4	5	5	5	4	4	4	4	3	5	5	5	4	4	4	4	3	4	4	4	3	3	3	3	2	4	4	4	3
5	5	5	4	5	5	5	4	4	4	4	3	5	5	5	4	5	5	5	4	5	5	5	4	4	4	4	3	5	5	5	4
5	5	5	4	5	5	5	4	4	4	4	3	5	5	5	4	4	4	4	3	4	4	4	3	3	3	3	2	4	4	4	3
5	5	5	4	5	5	5	4	4	4	4	3	5	5	5	4	5	5	5	4	5	5	5	4	4	4	4	3	5	5	5	4
4	4	4	3	4	4	4	3	3	3	3	2	4	4	4	3	3	3	3	2	3	3	3	2	2	2	2	1	3	3	3	2
4	4	4	3	4	4	4	3	3	3	3	2	4	4	4	3	4	4	4	3	4	4	4	3	3	3	3	2	4	4	4	3
5	5	5	4	5	5	5	4	4	4	4	3	5	5	5	4	4	4	4	3	4	4	4	3	3	3	3	2	4	4	4	3
5	5	5	4	5	5	5	4	4	4	4	3	5	5	5	4	5	5	5	4	5	5	5	4	4	4	4	3	5	5	5	4
5	5	5	4	5	5	5	4	4	4	4	3	5	5	5	4	4	4	4	3	4	4	4	3	3	3	3	2	4	4	4	3
5	5	5	4	5	5	5	4	4	4	4	3	5	5	5	4	5	5	5	4	5	5	5	4	4	4	4	3	5	5	5	4
5	5	5	4	5	5	5	4	4	4	4	3	5	5	5	4	4	4	4	3	4	4	4	3	3	3	3	2	4	4	4	3
5	5	5	4	5	5	5	4	4	4	4	3	5	5	5	4	5	5	5	4	5	5	5	4	4	4	4	3	5	5	5	4
4	4	4	3	4	4	4	3	3	3	3	2	4	4	4	3	3	3	3	2	3	3	3	2	2	2	2	1	3	3	3	2
4	4	4	3	4	4	4	3	3	3	3	2	4	4	4	3	4	4	4	3	4	4	4	3	3	3	3	2	4	4	4	3
5	5	5	4	5	5	5	4	4	4	4	3	5	5	5	4	4	4	4	3	4	4	4	3	3	3	3	2	4	4	4	3
5	5	5	4	5	5	5	4	4	4	4	3	5	5	5	4	5	5	5	4	5	5	5	4	4	4	4	3	5	5	5	4
5	5	5	4	5	5	5	4	4	4	4	3	5	5	5	4	4	4	4	3	4	4	4	3	3	3	3	2	4	4	4	3
5	5	5	4	5	5	5	4	4	4	4	3	5	5	5	4	5	5	5	4	5	5	5	4	4	4	4	3	5	5	5	4
5	5	5	4	5	5	5	4	4	4	4	3	5	5	5	4	4	4	4	3	4	4	4	3	3	3	3	2	4	4	4	3
5	5	5	4	5	5	5	4	4	4	4	3	5	5	5	4	5	5	5	4	5	5	5	4	4	4	4	3	5	5	5	4
3	3	3	2	3	3	3	2	2	2	2	1	3	3	3	2	2	2	2	1	2	2	2	1	1	1	1	0	2	2	2	1
3	3	3	2	3	3	3	2	2	2	2	1	3	3	3	2	3	3	3	2	3	3	3	2	2	2	2	1	3	3	3	2
4	4	4	3	4	4	4	3	3	3	3	2	4	4	4	3	3	3	3	2	3	3	3	2	2	2	2	1	3	3	3	2
4	4	4	3	4	4	4	3	3	3	3	2	4	4	4	3	4	4	4	3	4	4	4	3	3	3	3	2	4	4	4	3
4	4	4	3	4	4	4	3	3	3	3	2	4	4	4	3	3	3	3	2	3	3	3	2	2	2	2	1	3	3	3	2
4	4	4	3	4	4	4	3	3	3	3	2	4	4	4	3	4	4	4	3	4	4	4	3	3	3	3	2	4	4	4	3
4	4	4	3	4	4	4	3	3	3	3	2	4	4	4	3	3	3	3	2	3	3	3	2	2	2	2	1	3	3	3	2
4	4	4	3	4	4	4	3	3	3	3	2	4	4	4	3	4	4	4	3	4	4	4	3	3	3	3	2	4	4	4	3

Building neighborhoods in blocks

EMS-GT

- ex. d=5, k=5

  x = acgtacgtacgt

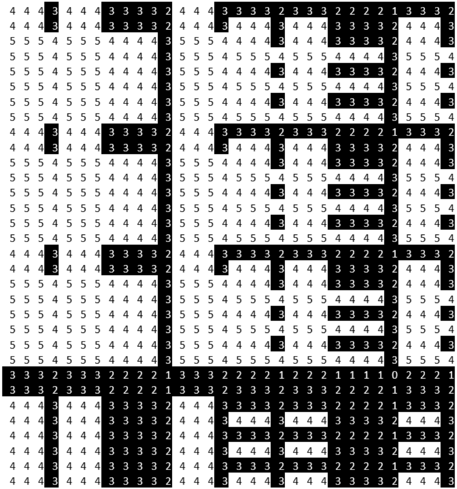
- if prefix = acatatg

  (2 mismatches from acgtacg)

- any k-suffix with up to

  3 mismatches from tacgt

  forms a neighbor of x



4	4	4	3	4	4	4	3	3	3	3	2	4	4	4	3	3	3	3	2	3	3	3	2	2	2	2	1	3	3	3	2
4	4	4	3	4	4	4	3	3	3	3	2	4	4	4	3	4	4	4	3	4	4	4	3	3	3	3	2	4	4	4	3
5	5	5	4	5	5	5	4	4	4	4	3	5	5	5	4	4	4	4	3	4	4	4	3	3	3	3	2	4	4	4	3
5	5	5	4	5	5	5	4	4	4	4	3	5	5	5	4	5	5	5	4	5	5	5	4	4	4	4	3	5	5	5	4
5	5	5	4	5	5	5	4	4	4	4	3	5	5	5	4	4	4	4	3	4	4	4	3	3	3	3	2	4	4	4	3
5	5	5	4	5	5	5	4	4	4	4	3	5	5	5	4	5	5	5	4	5	5	5	4	4	4	4	3	5	5	5	4
5	5	5	4	5	5	5	4	4	4	4	3	5	5	5	4	4	4	4	3	4	4	4	3	3	3	3	2	4	4	4	3
5	5	5	4	5	5	5	4	4	4	4	3	5	5	5	4	5	5	5	4	5	5	5	4	4	4	4	3	5	5	5	4
4	4	4	3	4	4	4	3	3	3	3	2	4	4	4	3	3	3	3	2	3	3	3	2	2	2	2	1	3	3	3	2
4	4	4	3	4	4	4	3	3	3	3	2	4	4	4	3	4	4	4	3	4	4	4	3	3	3	3	2	4	4	4	3
5	5	5	4	5	5	5	4	4	4	4	3	5	5	5	4	4	4	4	3	4	4	4	3	3	3	3	2	4	4	4	3
5	5	5	4	5	5	5	4	4	4	4	3	5	5	5	4	5	5	5	4	5	5	5	4	4	4	4	3	5	5	5	4
5	5	5	4	5	5	5	4	4	4	4	3	5	5	5	4	4	4	4	3	4	4	4	3	3	3	3	2	4	4	4	3
5	5	5	4	5	5	5	4	4	4	4	3	5	5	5	4	5	5	5	4	5	5	5	4	4	4	4	3	5	5	5	4
5	5	5	4	5	5	5	4	4	4	4	3	5	5	5	4	4	4	4	3	4	4	4	3	3	3	3	2	4	4	4	3
5	5	5	4	5	5	5	4	4	4	4	3	5	5	5	4	5	5	5	4	5	5	5	4	4	4	4	3	5	5	5	4
4	4	4	3	4	4	4	3	3	3	3	2	4	4	4	3	3	3	3	2	3	3	3	2	2	2	2	1	3	3	3	2
4	4	4	3	4	4	4	3	3	3	3	2	4	4	4	3	4	4	4	3	4	4	4	3	3	3	3	2	4	4	4	3
5	5	5	4	5	5	5	4	4	4	4	3	5	5	5	4	4	4	4	3	4	4	4	3	3	3	3	2	4	4	4	3
5	5	5	4	5	5	5	4	4	4	4	3	5	5	5	4	5	5	5	4	5	5	5	4	4	4	4	3	5	5	5	4
5	5	5	4	5	5	5	4	4	4	4	3	5	5	5	4	4	4	4	3	4	4	4	3	3	3	3	2	4	4	4	3
5	5	5	4	5	5	5	4	4	4	4	3	5	5	5	4	5	5	5	4	5	5	5	4	4	4	4	3	5	5	5	4
5	5	5	4	5	5	5	4	4	4	4	3	5	5	5	4	4	4	4	3	4	4	4	3	3	3	3	2	4	4	4	3
5	5	5	4	5	5	5	4	4	4	4	3	5	5	5	4	5	5	5	4	5	5	5	4	4	4	4	3	5	5	5	4
3	3	3	2	3	3	3	2	2	2	2	1	3	3	3	2	2	2	2	1	2	2	2	1	1	1	1	0	2	2	2	1
3	3	3	2	3	3	3	2	2	2	2	1	3	3	3	2	3	3	3	2	3	3	3	2	2	2	2	1	3	3	3	2
4	4	4	3	4	4	4	3	3	3	3	2	4	4	4	3	3	3	3	2	3	3	3	2	2	2	2	1	3	3	3	2
4	4	4	3	4	4	4	3	3	3	3	2	4	4	4	3	4	4	4	3	4	4	4	3	3	3	3	2	4	4	4	3
4	4	4	3	4	4	4	3	3	3	3	2	4	4	4	3	3	3	3	2	3	3	3	2	2	2	2	1	3	3	3	2
4	4	4	3	4	4	4	3	3	3	3	2	4	4	4	3	4	4	4	3	4	4	4	3	3	3	3	2	4	4	4	3
4	4	4	3	4	4	4	3	3	3	3	2	4	4	4	3	3	3	3	2	3	3	3	2	2	2	2	1	3	3	3	2
4	4	4	3	4	4	4	3	3	3	3	2	4	4	4	3	4	4	4	3	4	4	4	3	3	3	3	2	4	4	4	3

Building neighborhoods in blocks

EMS-GT

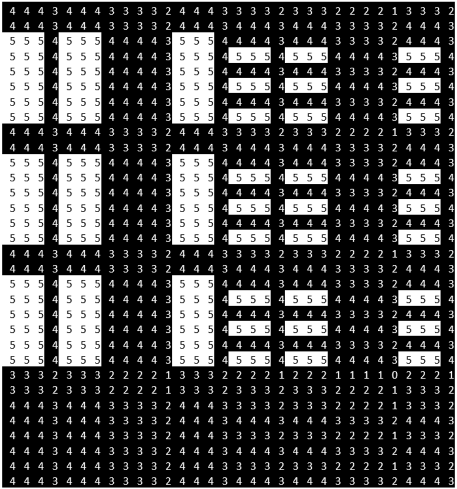
- ex. d=5, k=5

  x = acgtacgtacgt

- if prefix = acatacg

  (1 mismatch from acgtacg)

- any k-suffix with up to 4 mismatches from tacgt forms a neighbor of x



4	4	4	3	4	4	4	3	3	3	3	2	4	4	4	3	3	3	3	2	3	3	3	2	2	2	2	1	3	3	3	2
4	4	4	3	4	4	4	3	3	3	3	2	4	4	4	3	4	4	4	3	4	4	4	3	3	3	3	2	4	4	4	3
5	5	5	4	5	5	5	4	4	4	4	3	5	5	5	4	4	4	4	3	4	4	4	3	3	3	3	2	4	4	4	3
5	5	5	4	5	5	5	4	4	4	4	3	5	5	5	4	5	5	5	4	5	5	5	4	4	4	4	3	5	5	5	4
5	5	5	4	5	5	5	4	4	4	4	3	5	5	5	4	4	4	4	3	4	4	4	3	3	3	3	2	4	4	4	3
5	5	5	4	5	5	5	4	4	4	4	3	5	5	5	4	5	5	5	4	5	5	5	4	4	4	4	3	5	5	5	4
5	5	5	4	5	5	5	4	4	4	4	3	5	5	5	4	4	4	4	3	4	4	4	3	3	3	3	2	4	4	4	3
5	5	5	4	5	5	5	4	4	4	4	3	5	5	5	4	5	5	5	4	5	5	5	4	4	4	4	3	5	5	5	4
4	4	4	3	4	4	4	3	3	3	3	2	4	4	4	3	3	3	3	2	3	3	3	2	2	2	2	1	3	3	3	2
4	4	4	3	4	4	4	3	3	3	3	2	4	4	4	3	4	4	4	3	4	4	4	3	3	3	3	2	4	4	4	3
5	5	5	4	5	5	5	4	4	4	4	3	5	5	5	4	4	4	4	3	4	4	4	3	3	3	3	2	4	4	4	3
5	5	5	4	5	5	5	4	4	4	4	3	5	5	5	4	5	5	5	4	5	5	5	4	4	4	4	3	5	5	5	4
5	5	5	4	5	5	5	4	4	4	4	3	5	5	5	4	4	4	4	3	4	4	4	3	3	3	3	2	4	4	4	3
5	5	5	4	5	5	5	4	4	4	4	3	5	5	5	4	5	5	5	4	5	5	5	4	4	4	4	3	5	5	5	4
5	5	5	4	5	5	5	4	4	4	4	3	5	5	5	4	4	4	4	3	4	4	4	3	3	3	3	2	4	4	4	3
5	5	5	4	5	5	5	4	4	4	4	3	5	5	5	4	5	5	5	4	5	5	5	4	4	4	4	3	5	5	5	4
4	4	4	3	4	4	4	3	3	3	3	2	4	4	4	3	3	3	3	2	3	3	3	2	2	2	2	1	3	3	3	2
4	4	4	3	4	4	4	3	3	3	3	2	4	4	4	3	4	4	4	3	4	4	4	3	3	3	3	2	4	4	4	3
5	5	5	4	5	5	5	4	4	4	4	3	5	5	5	4	4	4	4	3	4	4	4	3	3	3	3	2	4	4	4	3
5	5	5	4	5	5	5	4	4	4	4	3	5	5	5	4	5	5	5	4	5	5	5	4	4	4	4	3	5	5	5	4
5	5	5	4	5	5	5	4	4	4	4	3	5	5	5	4	4	4	4	3	4	4	4	3	3	3	3	2	4	4	4	3
5	5	5	4	5	5	5	4	4	4	4	3	5	5	5	4	5	5	5	4	5	5	5	4	4	4	4	3	5	5	5	4
5	5	5	4	5	5	5	4	4	4	4	3	5	5	5	4	4	4	4	3	4	4	4	3	3	3	3	2	4	4	4	3
5	5	5	4	5	5	5	4	4	4	4	3	5	5	5	4	5	5	5	4	5	5	5	4	4	4	4	3	5	5	5	4
3	3	3	2	3	3	3	2	2	2	2	1	3	3	3	2	2	2	2	1	2	2	2	1	1	1	1	0	2	2	2	1
3	3	3	2	3	3	3	2	2	2	2	1	3	3	3	2	3	3	3	2	3	3	3	2	2	2	2	1	3	3	3	2
4	4	4	3	4	4	4	3	3	3	3	2	4	4	4	3	3	3	3	2	3	3	3	2	2	2	2	1	3	3	3	2
4	4	4	3	4	4	4	3	3	3	3	2	4	4	4	3	4	4	4	3	4	4	4	3	3	3	3	2	4	4	4	3
4	4	4	3	4	4	4	3	3	3	3	2	4	4	4	3	3	3	3	2	3	3	3	2	2	2	2	1	3	3	3	2
4	4	4	3	4	4	4	3	3	3	3	2	4	4	4	3	4	4	4	3	4	4	4	3	3	3	3	2	4	4	4	3
4	4	4	3	4	4	4	3	3	3	3	2	4	4	4	3	3	3	3	2	3	3	3	2	2	2	2	1	3	3	3	2
4	4	4	3	4	4	4	3	3	3	3	2	4	4	4	3	4	4	4	3	4	4	4	3	3	3	3	2	4	4	4	3

Building neighborhoods in blocks

EMS-GT

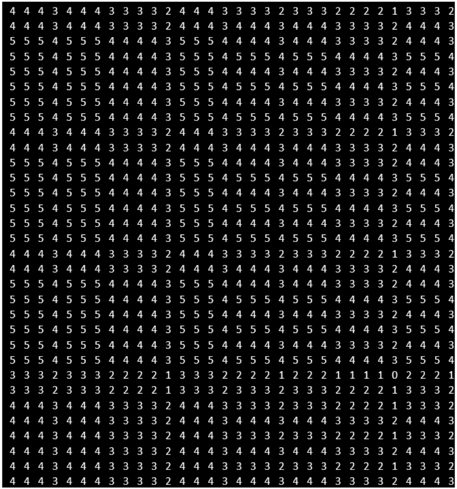
- ex. d=5, k=5

  x = `acgtacgtacgt`

- if prefix = `acgtacg`

  (0 mismatches, x's actual prefix)

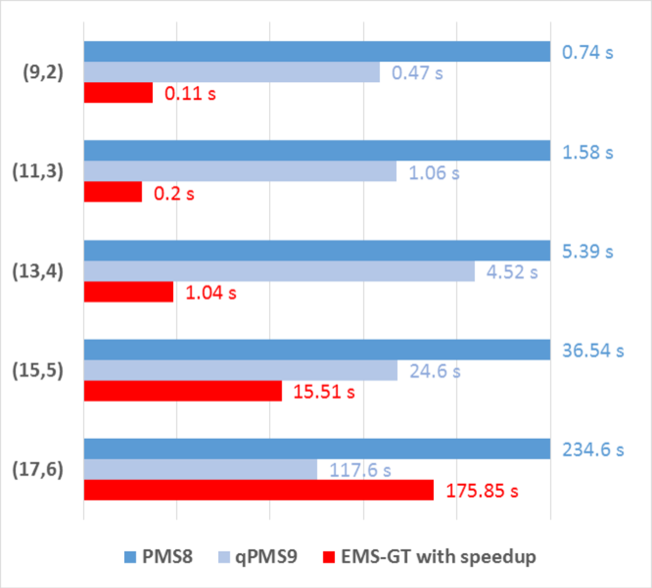
- any k-suffix forms a neighbor of x, since all k-suffixes have at most 5 mismatches from `tacgt`



```
4 4 4 3 4 4 4 3 3 3 3 2 4 4 4 3 3 3 3 2 3 3 3 2 2 2 2 1 3 3 3 2
4 4 4 3 4 4 4 3 3 3 3 2 4 4 4 3 4 4 4 3 4 4 4 3 3 3 3 2 4 4 4 3
5 5 5 4 5 5 5 4 4 4 4 3 5 5 5 4 4 4 4 3 4 4 4 3 3 3 3 2 4 4 4 3
5 5 5 4 5 5 5 4 4 4 4 3 5 5 5 4 5 5 5 4 5 5 5 4 4 4 4 3 5 5 5 4
5 5 5 4 5 5 5 4 4 4 4 3 5 5 5 4 4 4 4 3 4 4 4 3 3 3 3 2 4 4 4 3
5 5 5 4 5 5 5 4 4 4 4 3 5 5 5 4 5 5 5 4 5 5 5 4 4 4 4 3 5 5 5 4
5 5 5 4 5 5 5 4 4 4 4 3 5 5 5 4 4 4 4 3 4 4 4 3 3 3 3 2 4 4 4 3
5 5 5 4 5 5 5 4 4 4 4 3 5 5 5 4 5 5 5 4 5 5 5 4 4 4 4 3 5 5 5 4
4 4 4 3 4 4 4 3 3 3 3 2 4 4 4 3 3 3 3 2 3 3 3 2 2 2 2 1 3 3 3 2
4 4 4 3 4 4 4 3 3 3 3 2 4 4 4 3 4 4 4 3 4 4 4 3 3 3 3 2 4 4 4 3
5 5 5 4 5 5 5 4 4 4 4 3 5 5 5 4 4 4 4 3 4 4 4 3 3 3 3 2 4 4 4 3
5 5 5 4 5 5 5 4 4 4 4 3 5 5 5 4 5 5 5 4 5 5 5 4 4 4 4 3 5 5 5 4
5 5 5 4 5 5 5 4 4 4 4 3 5 5 5 4 4 4 4 3 4 4 4 3 3 3 3 2 4 4 4 3
5 5 5 4 5 5 5 4 4 4 4 3 5 5 5 4 5 5 5 4 5 5 5 4 4 4 4 3 5 5 5 4
5 5 5 4 5 5 5 4 4 4 4 3 5 5 5 4 4 4 4 3 4 4 4 3 3 3 3 2 4 4 4 3
5 5 5 4 5 5 5 4 4 4 4 3 5 5 5 4 5 5 5 4 5 5 5 4 4 4 4 3 5 5 5 4
4 4 4 3 4 4 4 3 3 3 3 2 4 4 4 3 3 3 3 2 3 3 3 2 2 2 2 1 3 3 3 2
4 4 4 3 4 4 4 3 3 3 3 2 4 4 4 3 4 4 4 3 4 4 4 3 3 3 3 2 4 4 4 3
5 5 5 4 5 5 5 4 4 4 4 3 5 5 5 4 4 4 4 3 4 4 4 3 3 3 3 2 4 4 4 3
5 5 5 4 5 5 5 4 4 4 4 3 5 5 5 4 5 5 5 4 5 5 5 4 4 4 4 3 5 5 5 4
5 5 5 4 5 5 5 4 4 4 4 3 5 5 5 4 4 4 4 3 4 4 4 3 3 3 3 2 4 4 4 3
5 5 5 4 5 5 5 4 4 4 4 3 5 5 5 4 5 5 5 4 5 5 5 4 4 4 4 3 5 5 5 4
5 5 5 4 5 5 5 4 4 4 4 3 5 5 5 4 4 4 4 3 4 4 4 3 3 3 3 2 4 4 4 3
5 5 5 4 5 5 5 4 4 4 4 3 5 5 5 4 5 5 5 4 5 5 5 4 4 4 4 3 5 5 5 4
3 3 3 2 3 3 3 2 2 2 2 1 3 3 3 2 2 2 2 1 2 2 2 1 1 1 1 0 2 2 2 1
3 3 3 2 3 3 3 2 2 2 2 1 3 3 3 2 3 3 3 2 3 3 3 2 2 2 2 1 3 3 3 2
4 4 4 3 4 4 4 3 3 3 3 2 4 4 4 3 3 3 3 2 3 3 3 2 2 2 2 1 3 3 3 2
4 4 4 3 4 4 4 3 3 3 3 2 4 4 4 3 4 4 4 3 4 4 4 3 3 3 3 2 4 4 4 3
4 4 4 3 4 4 4 3 3 3 3 2 4 4 4 3 3 3 3 2 3 3 3 2 2 2 2 1 3 3 3 2
4 4 4 3 4 4 4 3 3 3 3 2 4 4 4 3 4 4 4 3 4 4 4 3 3 3 3 2 4 4 4 3
4 4 4 3 4 4 4 3 3 3 3 2 4 4 4 3 3 3 3 2 3 3 3 2 2 2 2 1 3 3 3 2
4 4 4 3 4 4 4 3 3 3 3 2 4 4 4 3 4 4 4 3 4 4 4 3 3 3 3 2 4 4 4 3
```

Performance

EMS-GT


(9,2)
0.74 s
0.47 s
0.11 s
(11,3)
1.58 s
1.06 s
0.2 s
(13,4)
5.39 s
4.52 s
1.04 s
(15,5)
36.54 s
24.6 s
15.51 s
(17,6)
234.6 s
117.6 s
175.85 s
PMS8
qPMS9
EMS-GT with speedup