Optimal Mismatch Algorithm
Maximal Shift Algorithm

CS 655 : Analyzing Sequences

Presented by :Archana

Optimal Mismatch Algorithm

- Sunday D.M. , 1990
- Idea : Scan pattern from least frequent to most frequent character

Optimal Mismatch Algorithm

- Sunday D.M. , 1990
- Idea : Scan pattern from least frequent to most frequent character
- Need to know frequency of each character in alphabet

Preprocessing

- Sort pattern based on frequency
- Bad-character shift
- Good-suffix shift

Sorting pattern

Text = gcatcgcagagagtatacagtacg

Pattern = gcagagag

Sorting pattern

Text = gcatcgcagagagtatacagtacg

Pattern = gcagagag

Character	g			
Frequency	7			

Sorting pattern

Text = gcatcgcagagagtatacagtacg

Pattern = gcagagag

Character	g	c	a	t
Frequency	7	5	8	4

Sorting pattern

Text = gcatcgcagagagtatacagtacg

Pattern = gcagagag

Character	g	c	a	t
Frequency	7	5	8	4

i	0	1	2	3	4	5	6	7
x[i]	g	c	a	g	a	g	a	g
pat[i].loc								
pat[i].c								

Sorting pattern

Text = gcatcgcagagagtatacagtacg

Pattern = gcagagag

Character	g	c	a	t
Frequency	7	5	8	4

i	0	1	2	3	4	5	6	7
x[i]	g	c	a	g	a	g	a	g
pat[i].loc								
pat[i].c								

Sorting pattern

Text = gcatcgcagagagtatacagtacg

Pattern = gcagagag

Character	g	c	a	t
Frequency	7	5	8	4

i	0	1	2	3	4	5	6	7
x[i]	g	c	a	g	a	g	a	g
pat[i].loc	1							
pat[i].c	c							

Sorting pattern

Text = gcatcgcagagagtatacagtacg

Pattern = gcagagag

Character	g	c	a	t
Frequency	7	5	8	4

i	0	1	2	3	4	5	6	7
x[i]	g	c	a	g	a	g	a	g
pat[i].loc	1	7	5	3	0			
pat[i].c	c	g	g	g	g			

Sorting pattern

Text = gcatcgcagagagtatacagtacg

Pattern = gcagagag

Character	g	c	a	t
Frequency	7	5	8	4

i	0	1	2	3	4	5	6	7
x[i]	g	c	a	g	a	g	a	g
pat[i].loc	1	7	5	3	0	6	4	2
pat[i].c	c	g	g	g	g	a	a	a

Preprocessing

- Sort pattern based on frequency
- Bad-character shift
- Good-suffix shift

Bad Character Shift

Bad Character Shift

- T[k+m] is always involved in the next match

Bad Character Shift

- T[k+m] is always involved in the next match

k	0	1	2	3	4	5	6	7	
T[k]	g	c	a	a	g	c	a	g	
x[i]	g	c	a	g	a	g	a	g	

Bad Character Shift

- T[k+m] is always involved in the next match

k	0	1	2	3	4	5	6	7	
T[k]	g	c	a	a	g	c	a	g	a
x[i]	g	c	a	g	a	g	a	g	

Character	a			
B-C shift				

Bad Character Shift

- T[k+m] is always involved in the next match

k	0	1	2	3	4	5	6	7	
T[k]	g	c	a	a	g	c	a	g	a
x[i]	g	c	a	g	a	g	a	g	

Character	a			
B-C shift	8-6			

Bad Character Shift

- T[k+m] is always involved in the next match

k	0	1	2	3	4	5	6	7	
T[k]	g	c	a	a	g	c	a	g	a
x[i]	g	c	a	g	a	g	a	g	

Character	a			
B-C shift	2			

Bad Character Shift

- T[k+m] is always involved in the next match

k	0	1	2	3	4	5	6	7	
T[k]	g	c	a	a	g	c	a	g	c
x[i]	g	c	a	g	a	g	a	g	

Character	a	c		
B-C shift	2	7		

Bad Character Shift

- T[k+m] is always involved in the next match

k	0	1	2	3	4	5	6	7	
T[k]	g	c	a	a	g	c	a	g	g
x[i]	g	c	a	g	a	g	a	g	

Character	a	c	g	
B-C shift	2	7	1	

Bad Character Shift

- T[k+m] is always involved in the next match

k	0	1	2	3	4	5	6	7	
T[k]	g	c	a	a	g	c	a	g	t
x[i]	g	c	a	g	a	g	a	g	

Character	a	c	g	t
B-C shift	2	7	1	9

Preprocessing

- Sort pattern based on frequency
- Bad-character shift
- Good-suffix shift

Good-Suffix Shift

- GS[j] is the minimum left shift so that pat[I[0]]....pat[I[j-1]] matched their aligned characters in pat, but pat[I[j]] does not

i	0	1	2	3	4	5	6	7	8
adaptegGs	1	3	4	2	7	7	7	7	7

Example

g c a t c g c a g a g a g t a t a c a g t a c g

g c a g a g a g

i	0	1	2	3	4	5	6	7
pat[i].loc	1	7	5	3	0	6	4	2
pat[i].c	c	g	g	g	g	a	a	a
adaptegGs	1	3	4	2	7	7	7	7

Character	a	c	g	t
B-C shift	2	7	1	9

Example

gcatcgcagagagtatacagtacg

gcagagag

i	0	1	2	3	4	5	6	7
pat[i].loc	1	7	5	3	0	6	4	2
pat[i].c	c	g	g	g	g	a	a	a
adaptegGs	1	3	4	2	7	7	7	7

Character	a	c	g	t
B-C shift	2	7	1	9

Example

```
g c a t c g c a g a g a g t a t a c a g t a c g
      g c a g a g a g
```

i	0	1	2	3	4	5	6	7
pat[i].loc	1	7	5	3	0	6	4	2
pat[i].c	c	g	g	g	g	a	a	a
adaptegGs	1	3	4	2	7	7	7	7

Character	a	c	g	t
B-C shift	2	7	1	9

Example

gcatcgcagagagtatacagtacg

gcagagag

i	0	1	2	3	4	5	6	7
pat[i].loc	1	7	5	3	0	6	4	2
pat[i].c	c	g	g	g	g	a	a	a
adaptegGs	1	3	4	2	7	7	7	7

Character	a	c	g	t
B-C shift	2	7	1	9

Example

gcatcgcagagagtatacagtacg

gcagagag

i	0	1	2	3	4	5	6	7
pat[i].loc	1	7	5	3	0	6	4	2
pat[i].c	c	g	g	g	g	a	a	a
adaptegGs	1	3	4	2	7	7	7	7

Character	a	c	g	t
B-C shift	2	7	1	9

Example

gcatcgcagagagtatacagtacg

gcagagag

i	0	1	2	3	4	5	6	7
pat[i].loc	1	7	5	3	0	6	4	2
pat[i].c	c	g	g	g	g	a	a	a
adaptegGs	1	3	4	2	7	7	7	7

Character	a	c	g	t
B-C shift	2	7	1	9

Example

gcatcgcagagagtatacagtacg

gcagagag

i	0	1	2	3	4	5	6	7
pat[i].loc	1	7	5	3	0	6	4	2
pat[i].c	c	g	g	g	g	a	a	a
adaptegGs	1	3	4	2	7	7	7	7

Character	a	c	g	t
B-C shift	2	7	1	9

Example

gcatcgcagagagtatacagtacg

gcagagag

i	0	1	2	3	4	5	6	7
pat[i].loc	1	7	5	3	0	6	4	2
pat[i].c	c	g	g	g	g	a	a	a
adaptegGs	1	3	4	2	7	7	7	7

Character	a	c	g	t
B-C shift	2	7	1	9

Maximal Shift Algorithm

- Pattern is scanned from the character that gives to the maximum shift to that which gives minimum shift.

Maximal Shift Algorithm

- Pattern is scanned from the character that gives to the maximum shift to that which gives minimum shift.

i	0	1	2	3	4	5	6	7
x[i]	g	c	a	g	a	g	a	g
minShift[i]	1	2	3	3	2	2	2	2
pat[i].loc								
pat[i].c								

Maximal Shift Algorithm

- Pattern is scanned from the character that gives to the maximum shift to that which gives minimum shift.

i	0	1	2	3	4	5	6	7
x[i]	g	c	a	g	a	g	a	g
minShift[i]	1	2	3	3	2	2	2	2
pat[i].loc	3							
pat[i].c	g							

Maximal Shift Algorithm

- Pattern is scanned from the character that gives to the maximum shift to that which gives minimum shift.

i	0	1	2	3	4	5	6	7
x[i]	g	c	a	g	a	g	a	g
minShift[i]	1	2	3	3	2	2	2	2
pat[i].loc	3	2						
pat[i].c	g	a						

Maximal Shift Algorithm

- Pattern is scanned from the character that gives to the maximum shift to that which gives minimum shift.

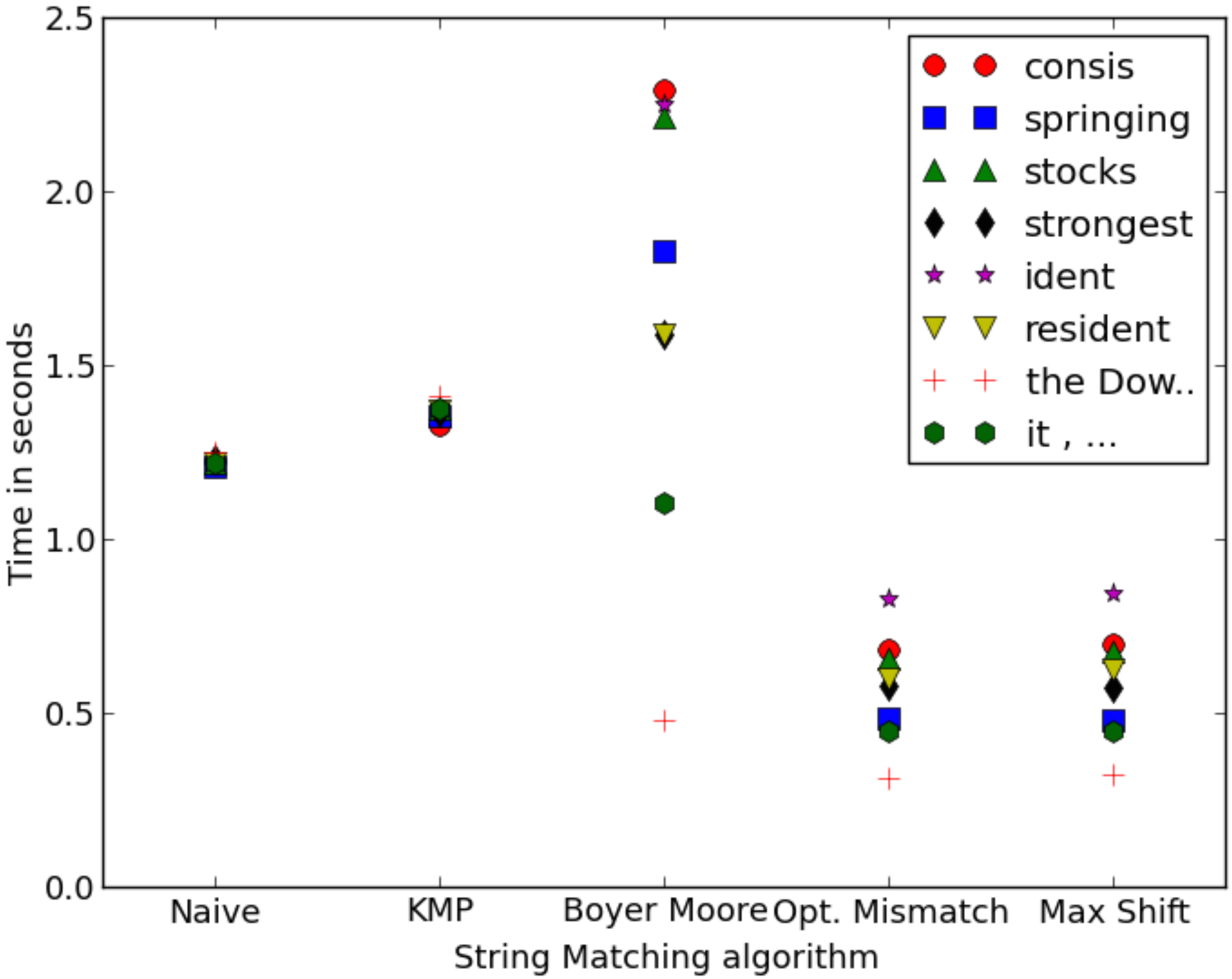
i	0	1	2	3	4	5	6	7
x[i]	g	c	a	g	a	g	a	g
minShift[i]	1	2	3	3	2	2	2	2
pat[i].loc	3	2	7	6	5	4	1	
pat[i].c	g	a	g	a	g	a	c	

Maximal Shift Algorithm

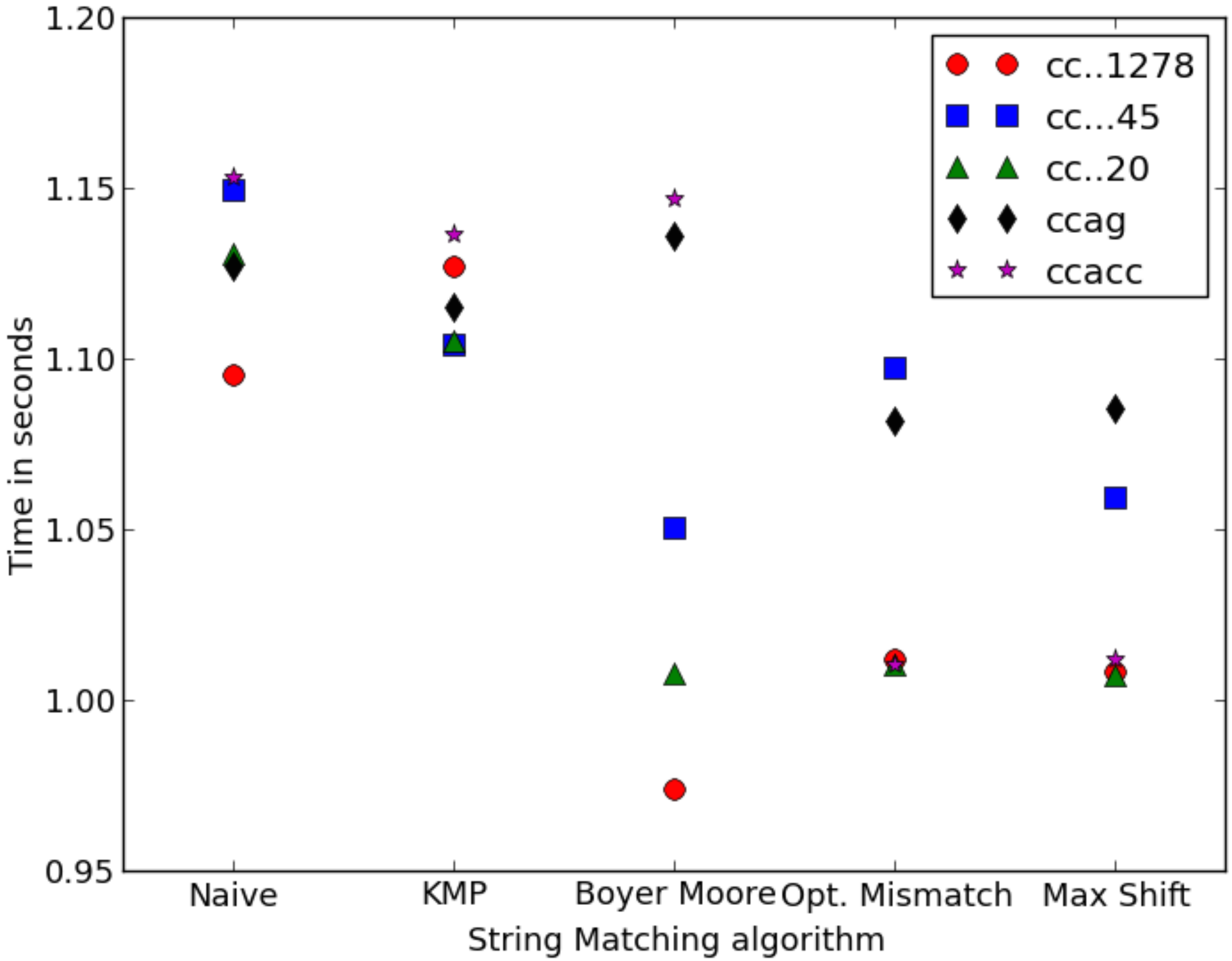
- Pattern is scanned from the character that gives to the maximum shift to that which gives minimum shift.

i	0	1	2	3	4	5	6	7
x[i]	g	c	a	g	a	g	a	g
minShift[i]	1	2	3	3	2	2	2	2
pat[i].loc	3	2	7	6	5	4	1	0
pat[i].c	g	a	g	a	g	a	c	9

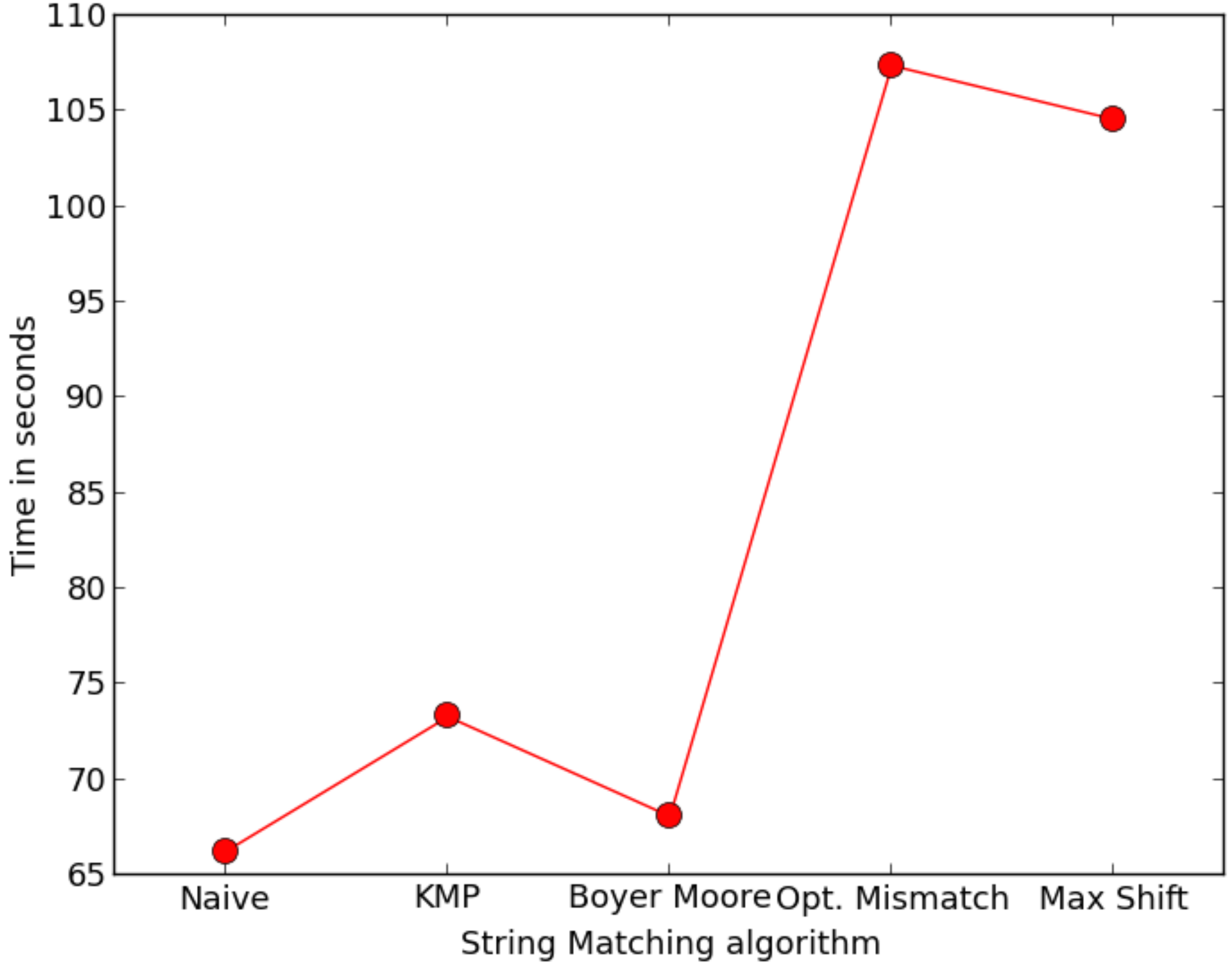
Newswire text



Nucleotide bases



Worst-case scenario


Time in seconds
Naive
KMP
Boyer Moore
Opt. Mismatch
Max Shift
String Matching algorithm