

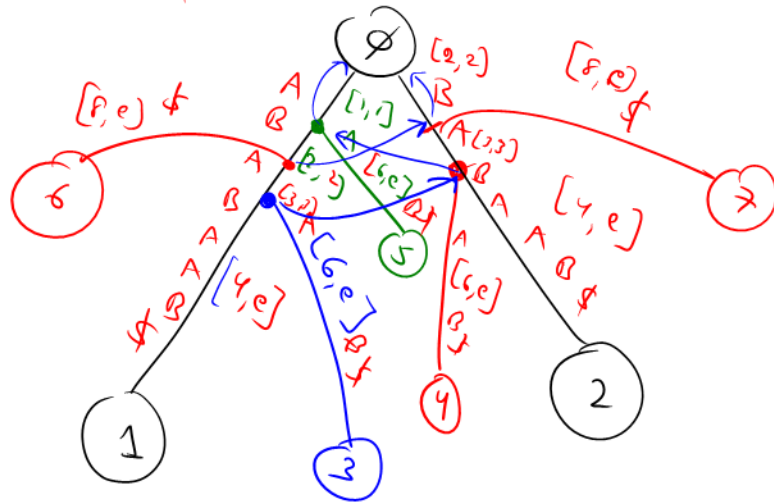
## Suffix Tree

$O(n^2)$  time & space  $\leftarrow$  naive

$O(n)$  space & time  $\leftarrow$  Ukkonen's Algo

$$\Sigma = \{A, B\}$$
$$[s, e] \leftarrow [\text{start}, \text{end}]$$
$$2^{\text{nd}} \text{ supply: } [2, e]$$

<u>e</u>	<u>explicit last suffix</u>
1	1
2	2
3	3
4	4
5	5
6	
7	
8	



suffix links

$$s(ABA) = BA$$

$$S(AB) = n$$

$$S(A) = \emptyset$$

$$S(BA) = A$$

$$\gamma(\beta) = \emptyset$$

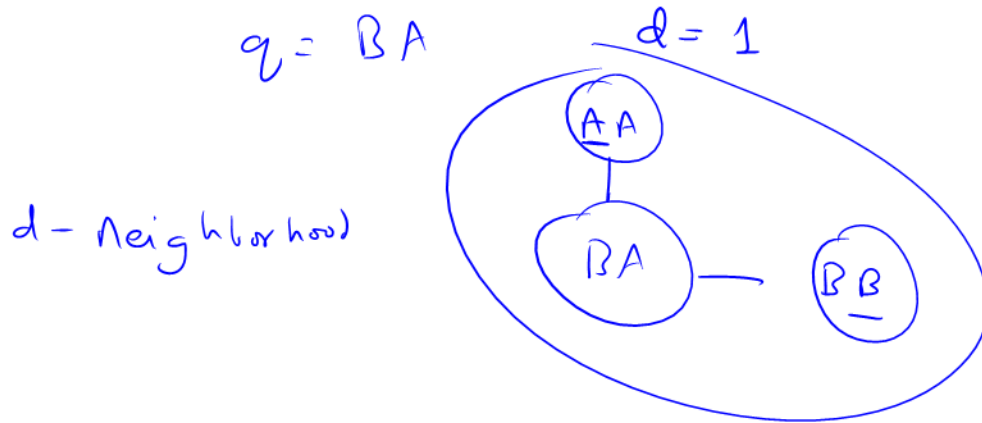
Mississippi t ← try it!

1) querying  $q = BA$  : answer :  $(2, 4)$

$O(|g|)$  time + # of matches

2) Inexact matches :  $|q| = l \leftarrow$  length of the query  
 $d \leftarrow \#$  of mismatches  
 $d < l$

Naive approach:



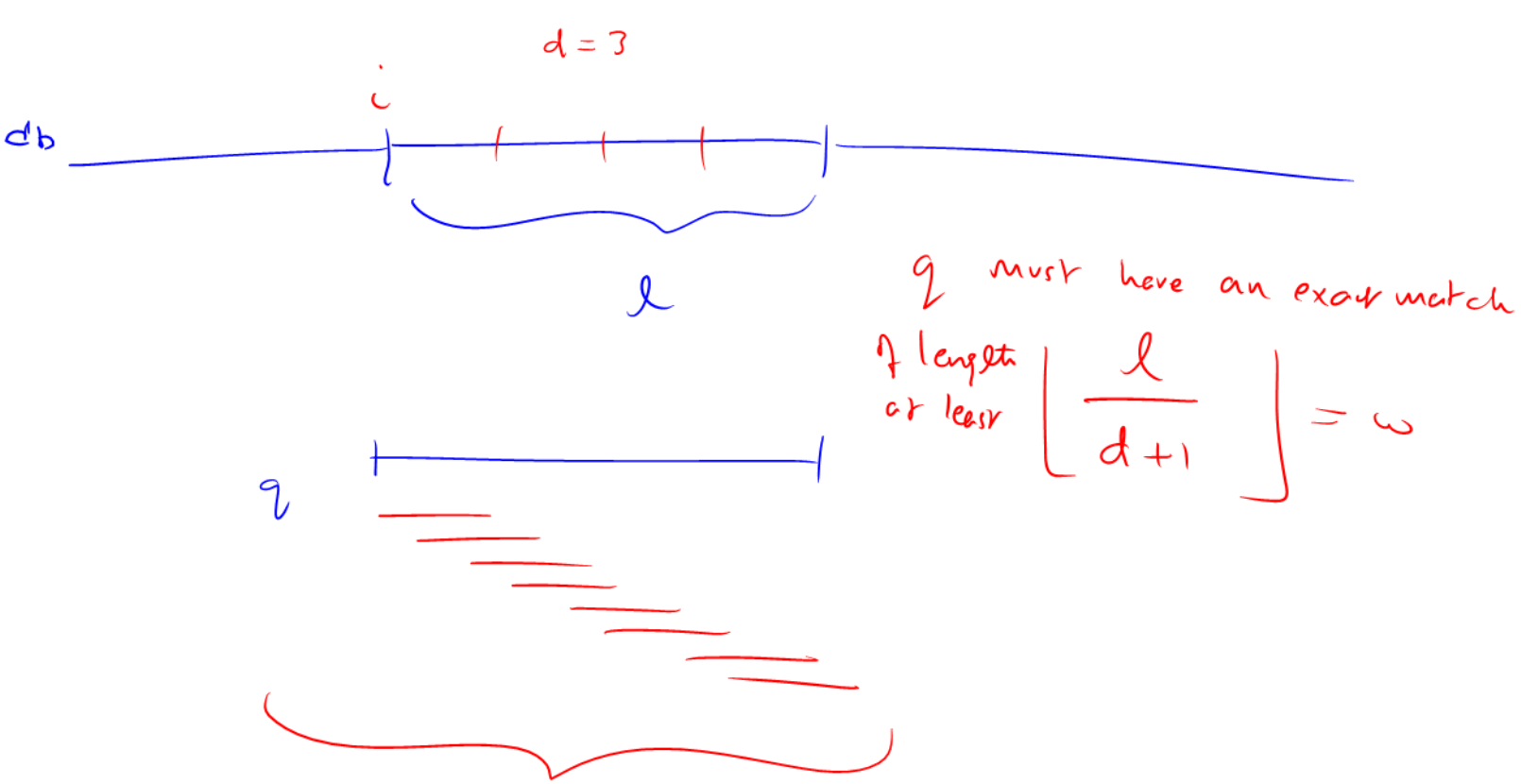
Use the  
 suffix tree to  
 spell out the  
 neighbors "with  
 reverse counts"

Suffix Tree : Human genome

$$O(|\Sigma|^d) \leftarrow \text{size of the neighborhood}$$

$d$  has to be small

---

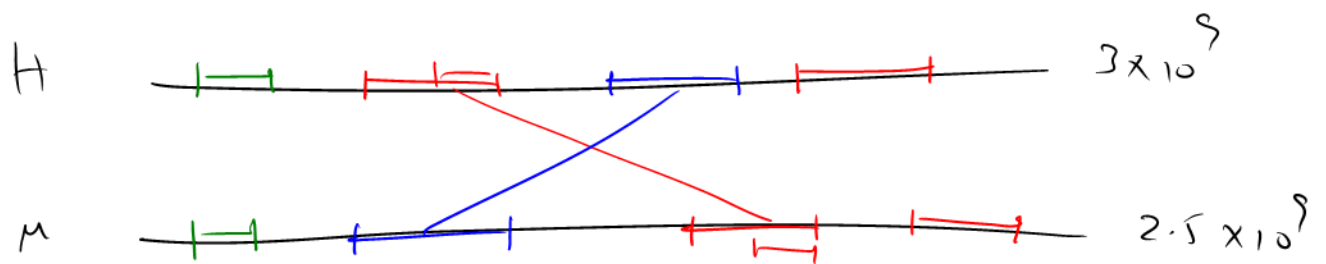


generate subquery of len  $w$

→ look for an exact match

→ extend to longer matches

## 2) Genome Alignment



MUMer

→ Maximal Unique Matches

$O(n)$  time!

maximal  $\Rightarrow$  it cannot be extended to left or right in either string.

Unique  $\Rightarrow$  One occurrence in each string

Handwritten notes illustrating the construction of a DFA for the regular expression  $(AB)^*$ .

The DFA is defined by the following components:

- States:**  $q_0, q_1, q_2$
- Start State:**  $q_0$  (indicated by an arrow pointing to it)
- Final State:**  $q_2$  (indicated by a double circle around it)
- Transitions:**
  - $q_0 \xrightarrow{A} q_1$
  - $q_1 \xrightarrow{B} q_2$
  - $q_2 \xrightarrow{A} q_1$
  - $q_2 \xrightarrow{B} q_2$

The DFA is represented as a directed graph:

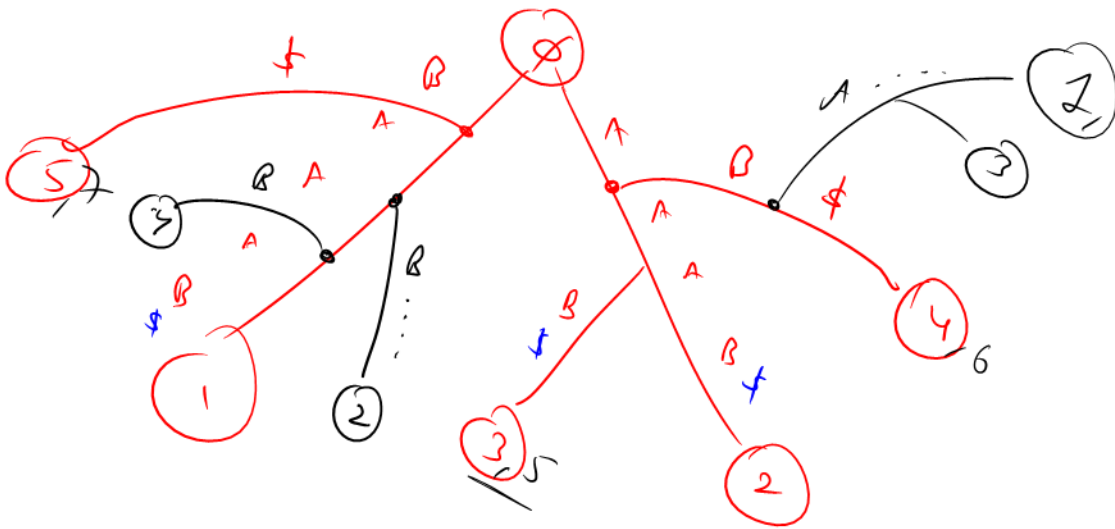
```

graph LR
    start(( )) --> q0((q0))
    q0 -- A --> q1((q1))
    q1 -- B --> q2(((q2)))
    q2 -- A --> q1
    q2 -- B --> q2
  
```

Examples of strings accepted by the DFA:

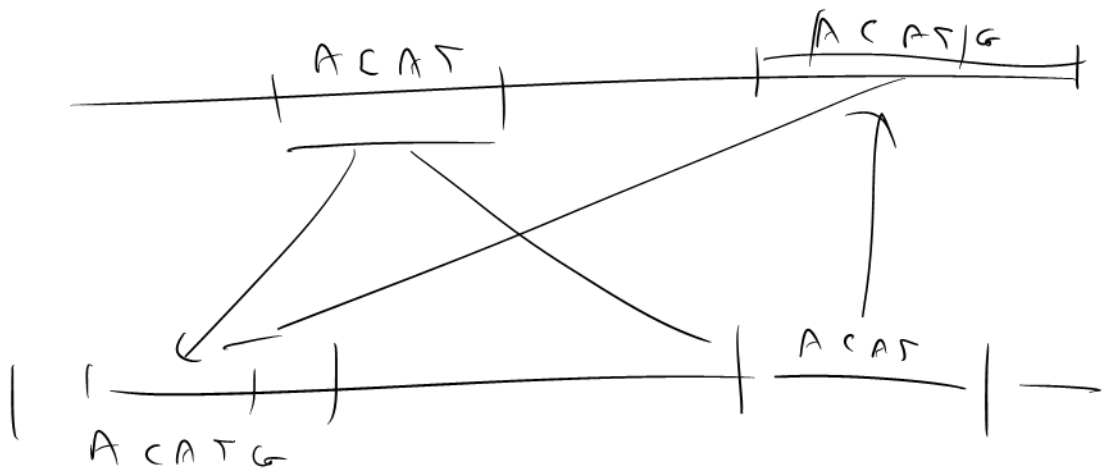
- $I_1: ABABAB$  (The string is shown with a blue underline under the entire string and a red circle around the final state  $q_2$  in the transition diagram.)
- $M: BAAB\$$  (The string is shown with a red circle around the final state  $q_2$  in the transition diagram, and a blue underline under the final state  $q_2$  in the transition diagram.)

- 1) build ST for one of the strings
  - 2) add  $2^n$  string to the suffix tree
- generalized  
suffix tree

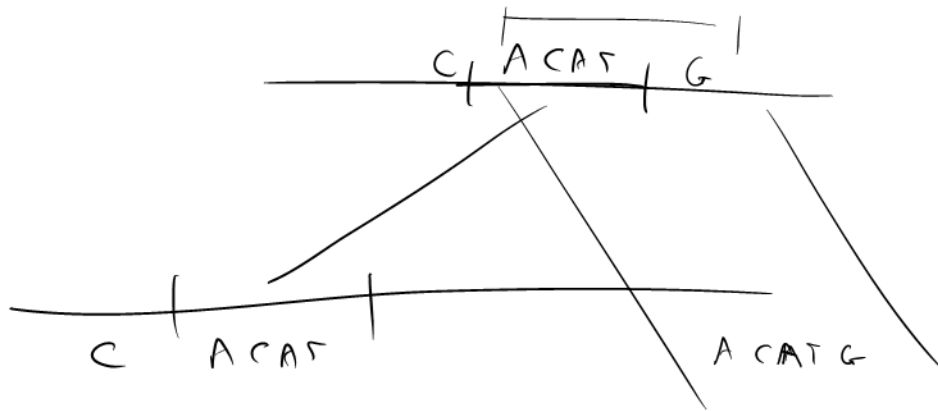


Unique matches: A A B  $\begin{pmatrix} M & H \\ 3 & 5 \end{pmatrix}$   $\xrightarrow{\text{Maximal}}$  ✓

RAA  $(\begin{smallmatrix} M & H \\ 1 & 4 \end{smallmatrix})$  ✓

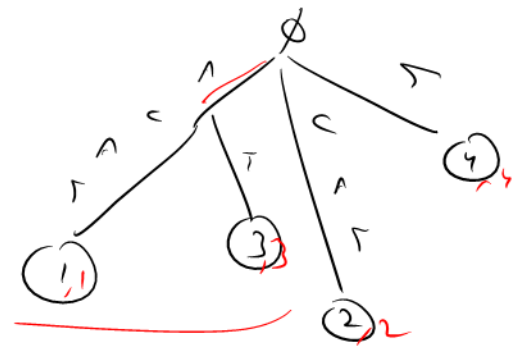


ACAT ← not unique



H: (ACAT)

M: (ACAT)



ACAT  
CAT  
AT  
T

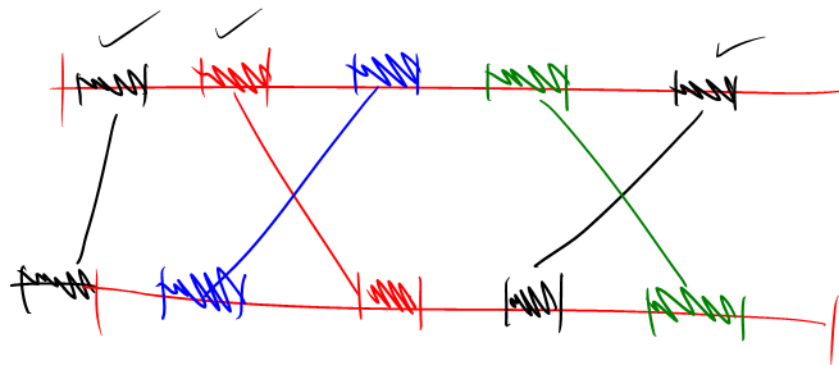
1) find the num:

$$\underbrace{O(n)}_{\text{suffix tree}} + \underbrace{O(m)}_{\text{maximal cliques over the \# of unique matches (m)}}$$

$$O(n |\Sigma|) \leftarrow \text{real work}$$

maximal cliques  
over the # of  
unique matches (m)

2)



find the maximum weighted subsequence of segments  
between the 2 genomes : DP

suffix arrays

sorted list of suffixes of a DB string

1 2 3 4 5 6 7 8  
A B A B A A B \$

suffix

array  $L_1 \rightarrow$

lex  
order

$L_2 \rightarrow$

$L_3 \rightarrow$

$\rightarrow$

$R \rightarrow$

8	:	\$
5	:	A A B \$
6	:	A B \$
3	:	A B A A B \$
1	:	A B A B A B \$
7	:	B \$
4	:	B A A B \$
2	:	B A B A A B \$

Conceptual

$$\$ < A < B$$

total order on  $\Sigma \cup \$$

q: BA ?

$$O(|q| \log \frac{n}{|q|})$$



DB size

+

$O(\# \text{ matches})$

$$\underline{\underline{O(\log n) + O(|q|)}}$$

how to do this?

Answer: 2, 4