On the Communication Complexity of Approximate Pattern Matching

Jakob Nogler[2]

based on joint work with

Tomasz Kociumaka[1] **Philip Wellnitz**[3]

[1]Max Planck Institute for Informatics, SIC ($\rightarrow$ INSAIT)

[2]ETH Zurich

[3]National Institute of Informatics, SOKENDAI

Strings

- A *string* is a sequence of characters from an *alphabet*.

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
S	a	b	a	a	b	a	a	b	a	a	b	a	a	b	a	a	b

Strings

- A *string* is a sequence of characters from an *alphabet*.

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
S	a	b	a	a	b	a	a	b	a	a	b	a	a	b	a	a	b

$S[12]$

Strings

- A *string* is a sequence of characters from an *alphabet*.

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
S	a	b	a	a	b	a	a	b	a	a	b	a	a	b	a	a	b

$S[3..10)$ $S[12]$

Strings

- A *string* is a sequence of characters from an *alphabet*.

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
S	a	b	a	a	b	a	a	b	a	a	b	a	a	b	a	a	b

$S[3..10)$ $S[12]$

- An integer p is *a period* of a string S if $S[i] = S[i+p]$ for all $i \in \{0, \ldots, |S| - p - 1\}$.

Strings

- A *string* is a sequence of characters from an *alphabet*.

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
S	a	b	a	a	b	a	a	b	a	a	b	a	a	b	a	a	b

$S[3..10)$ $S[12]$

- An integer p is *a period* of a string S if $S[i] = S[i+p]$ for all $i \in \{0, \ldots, |S| - p - 1\}$.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
a	b	a	a	b	a	a	b	a	a	b	a	a	b	a	a	b

$p = 3$

Hamming and Edit Distance

- The *Hamming distance* $\mathrm{HD}(X, Y)$ measures the number of mismatching characters between strings X, Y.

Hamming and Edit Distance

- The *Hamming distance* $\mathrm{HD}(X, Y)$ measures the number of mismatching characters between strings X, Y.

X	b	b	a	a	a	b	b	b	a	b	b	a	a
Y	a	b	a	b	a	b	b	b	a	b	b	a	a

$$\mathrm{HD}(X, Y) = 2$$

Hamming and Edit Distance

- The *Hamming distance* $\mathrm{HD}(X, Y)$ measures the number of mismatching characters between strings X, Y.

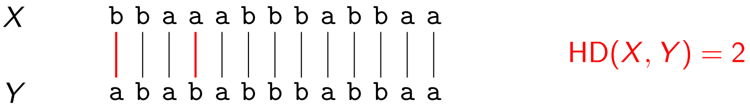
X	`b b a a a b b b a b b a a`
Y	`a b a b a b b b a b b a a`

$\mathrm{HD}(X, Y) = 2$

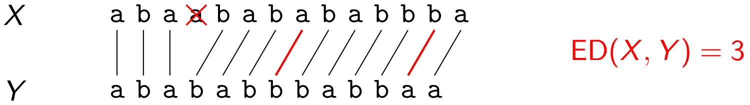
- The *edit distance* $\mathrm{ED}(X, Y)$ measures the minimum number of insertions, deletions, and substitutions of characters to transform X into Y.

Hamming and Edit Distance

- The *Hamming distance* $\mathrm{HD}(X, Y)$ measures the number of mismatching characters between strings X, Y.

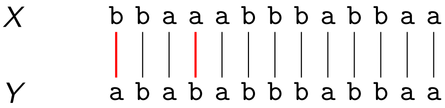


- The *edit distance* $\mathrm{ED}(X, Y)$ measures the minimum number of insertions, deletions, and substitutions of characters to transform X into Y.



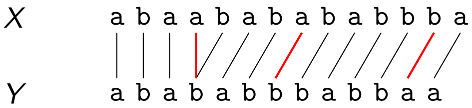
Hamming and Edit Distance

- The *Hamming distance* $\mathrm{HD}(X, Y)$ measures the number of mismatching characters between strings X, Y.



$\mathrm{HD}(X, Y) = 2$

- The *edit distance* $\mathrm{ED}(X, Y)$ measures the minimum number of insertions, deletions, and substitutions of characters to transform X into Y.



$\mathrm{ED}(X, Y) = 3$

Alignment

Pattern Matching (PM)

Text T, $\|T\| = n$	`a b a a b a b a b a b b b a b b a a a b b b a b b a a`
Pattern P, $\|P\| = m$	`a b a b a b b b a b b a a`

Pattern Matching (PM)

Text T, $|T| = n$ `a b a a b a b a b a b b b a b b a a a b b b a b b a a`

Pattern P, $|P| = m$ `a b a b a b b b a b b a a`

- **Exact PM:** Compute $\mathrm{Occ}(P, T) := \{x \mid T[x \mathinner{.\,.} x + m) = P\}$.

Pattern Matching (PM)

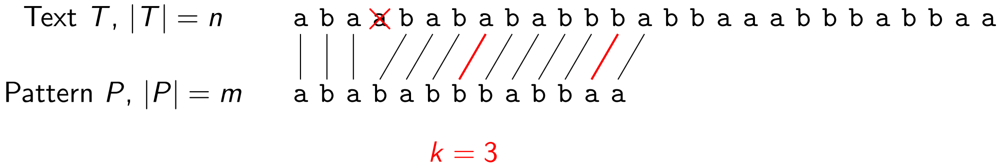
Text T, $|T| = n$ a b a a b a b a b a b b b a b b a a a b b b a b b a a

Pattern P, $|P| = m$ a b a b a b b b a b b a a

$k = 2$

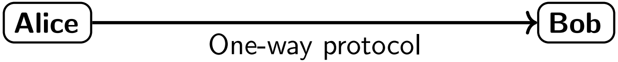
- **Exact PM:** Compute $\text{Occ}(P, T) := \{x \mid T[x \mathinner{.\,.} x+m) = P\}$.
- **PM with mismatches:** Compute $\text{Occ}_k^H(P, T) := \{x \mid \text{HD}(T[x \mathinner{.\,.} x+m), P) \le k\}$.

Pattern Matching (PM)

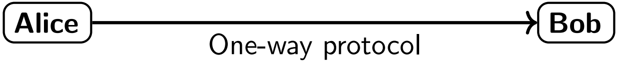


- **Exact PM:** Compute $\mathrm{Occ}(P,T) := \{x \mid T[x\,..\,x+m) = P\}$.
- **PM with mismatches:** Compute $\mathrm{Occ}^H_k(P,T) := \{x \mid \mathrm{HD}(T[x\,..\,x+m), P) \le k\}$.
- **PM with edits:** Compute $\mathrm{Occ}^E_k(P,T) := \{x \mid \exists y\ \mathrm{ED}(T[x\,..\,y), P) \le k\}$.

Communication Complexity



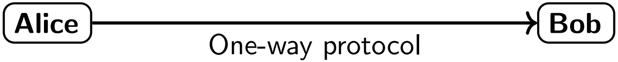
Communication Complexity



(1) Alice receives a PM instance.

Text T, Pattern P, Threshold k

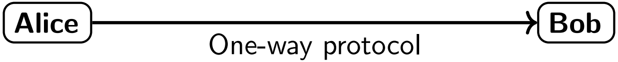
Communication Complexity



① Alice receives a PM instance.
Text T, Pattern P, Threshold k

② Alice compresses the input.

Communication Complexity

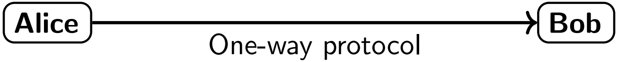


① Alice receives a PM instance.
Text T, Pattern P, Threshold k

② Alice compresses the input.

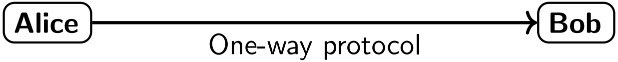
③ Alice sends compressed data to Bob.

Communication Complexity



1. Alice receives a PM instance. Text T, Pattern P, Threshold k
2. Alice compresses the input.
3. Alice sends compressed data to Bob.
4. Bob needs to reconstructs the output of the instance. $Set\ Occ_k^E(P, T)$

Communication Complexity



(1) Alice receives a PM instance.
Text T, Pattern P, Threshold k

(2) Alice compresses the input.

(3) Alice sends compressed data to Bob.

(4) Bob needs to reconstructs the output of the instance.
Set $Occ^E_k(P, T)$

Communication Complexity = “minimum # of machine words to send to Bob”

Example for Exact Pattern Matching

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29
Text T	a	b	b	a	b	a	a	b	a	a	b	a	a	b	a	a	b	a	a	b	a	a	b	a	a	b	a	a	b	b
Pattern P	a	b	a	a	b	a	a	b	a	a	b	a	a	b	a	a	b													

Alice needs to send to Bob the set $\mathrm{Occ}(P, T) = \{3, 6, 9, 12\}$

Example for Exact Pattern Matching

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29
Text T	a	b	b	a	b	a	a	b	a	a	b	a	a	b	a	a	b	a	a	b	a	a	b	a	a	b	a	a	b	b
Pattern P	a	b	a	a	b	a	a	b	a	a	b	a	a	b	a	a	b													

Alice needs to send to Bob the set $\text{Occ}(P, T) = \{3, 6, 9, 12\}$

She has more than one way how to do it:

Example for Exact Pattern Matching

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29
Text T	a	b	b	a	b	a	a	b	a	a	b	a	a	b	a	a	b	a	a	b	a	a	b	a	a	b	a	a	b	b
Pattern P	a	b	a	a	b	a	a	b	a	a	b	a	a	b	a	a	b													

Alice needs to send to Bob the set $\text{Occ}(P, T) = \{3, 6, 9, 12\}$

She has more than one way how to do it:

1. She can send $\text{Occ}(P, T)$ explicitly.

Example for Exact Pattern Matching

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29
Text T	a	b	b	a	b	a	a	b	a	a	b	a	a	b	a	a	b	a	a	b	a	a	b	a	a	b	a	a	b	b
Pattern P	a	b	a	a	b	a	a	b	a	a	b	a	a	b	a	a	b													

Alice needs to send to Bob the set $\mathrm{Occ}(P, T) = \{3, 6, 9, 12\}$

She has more than one way how to do it:

1. She can send $\mathrm{Occ}(P, T)$ explicitly.
2. She can send $\mathrm{Occ}(P, T)$ in a compressed form.

Example for Exact Pattern Matching

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29
Text T	a	b	b	a	b	a	a	b	a	a	b	a	a	b	a	a	b	a	a	b	a	a	b	a	a	b	a	a	b	b
Pattern P	a	b	a	a	b	a	a	b	a	a	b	a	a	b	a	a	b													

Alice needs to send to Bob the set $\mathrm{Occ}(P, T) = \{3, 6, 9, 12\}$

She has more than one way how to do it:

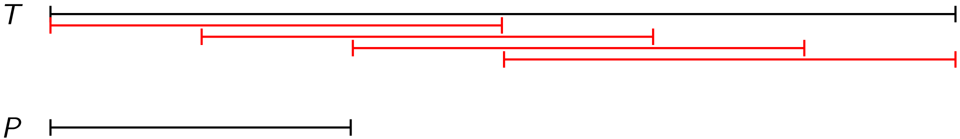
1. She can send $\mathrm{Occ}(P, T)$ explicitly.
2. She can send $\mathrm{Occ}(P, T)$ in a compressed form.
3. She can send P, T.

General Assumptions

1. $n \leq 3/2 \cdot m$

General Assumptions

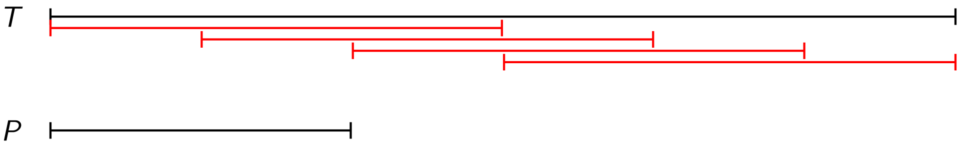
1. $\boldsymbol{n \leq 3/2 \cdot m}$



- Divide T into $\Theta(n/m)$ blocks of length $n \leq 3/2 \cdot m$, and apply protocol on each block.

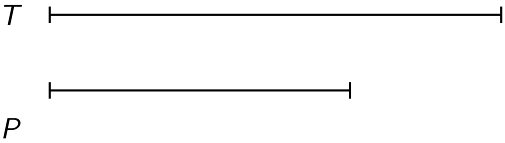
General Assumptions

1. $\boldsymbol{n \leq 3/2 \cdot m}$



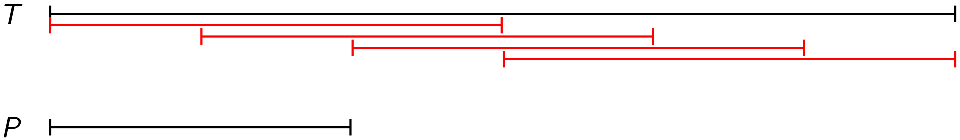
- Divide T into $\Theta(n/m)$ blocks of length $n \leq 3/2 \cdot m$, and apply protocol on each block.

2. **An exact/k-mismatch/k-edit occurrence of P aligns with prefix and suffix of T**



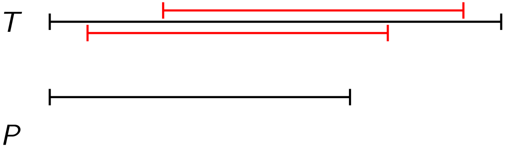
General Assumptions

1. $\boldsymbol{n \leq 3/2 \cdot m}$



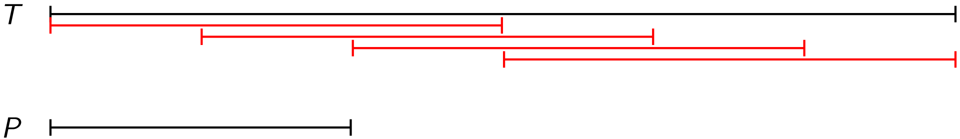
   - Divide T into $\Theta(n/m)$ blocks of length $n \leq 3/2 \cdot m$, and apply protocol on each block.

2. **An exact/k-mismatch/k-edit occurrence of P aligns with prefix and suffix of T**



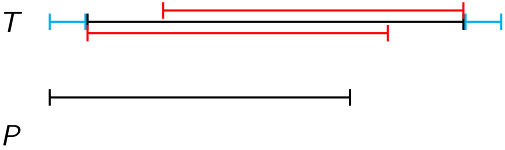
General Assumptions

1. $\boldsymbol{n \leq 3/2 \cdot m}$



- Divide T into $\Theta(n/m)$ blocks of length $n \leq 3/2 \cdot m$, and apply protocol on each block.

2. **An exact/k-mismatch/k-edit occurrence of P aligns with prefix and suffix of T**



Previous Results

	Upper Bound	**Lower Bound**	**Refererence**
Exact PM	$\mathcal{O}(1)$	$\Omega(1)$	Periodicity Lemma, **[FW65]**

Previous Results

	Upper Bound	Lower Bound	Refererence
Exact PM	$\mathcal{O}(1)$	$\Omega(1)$	Periodicity Lemma, **[FW65]**
PM with mismatches	$\mathcal{O}(k)$	$\Omega(k)$	**[CKP19]**

N.B. In **[CKP19]** Alice sends to Bob $\mathrm{Occ}_k^H(P, T)$ **plus** the *mismatch information* $\mathrm{MI}(x)$ for all $x \in \mathrm{Occ}_k^H(P, T)$, defined as

$$\mathrm{MI}(x) := \{(i, P[i], T[x+i]) \mid i \in [0..m) \text{ such that } P[i] \neq T[x+i]\}.$$

Previous Results

	Upper Bound	**Lower Bound**	**Refererence**
Exact PM	$\mathcal{O}(1)$	$\Omega(1)$	Periodicity Lemma, **[FW65]**
PM with mismatches	$\mathcal{O}(k)$	$\Omega(k)$	**[CKP19]**
PM with edits	$\mathcal{O}(k^3)$		**[CKW20]**

Previous Results

	Upper Bound	**Lower Bound**	**Refererence**
Exact PM	$\mathcal{O}(1)$	$\Omega(1)$	Periodicity Lemma, **[FW65]**
PM with mismatches	$\mathcal{O}(k)$	$\Omega(k)$	**[CKP19]**
PM with edits	$\mathcal{O}(k^3)$		**[CKW20]**
PM with edits	$\boldsymbol{\mathcal{O}(k \log n)}$	$\boldsymbol{\Omega(k)}$	**[KNW24]**

Exact Pattern Matching

Exact PM

Periodicity Lemma Readapted [FW65]

If $n \leq 3/2 \cdot m$ and $\{0, n-m\} \subseteq \text{Occ}(P, T)$, then $\gcd(\text{Occ}(P, T))$ is a period of T.

Text T a b a a b a a b a a b a a b a a b a a b a a b a a b

Pattern P a b a a b a a b a a b a a b a a b

Exact PM

Periodicity Lemma Readapted [FW65]

If $n \leq 3/2 \cdot m$ and $\{0, n-m\} \subseteq \mathrm{Occ}(P,T)$, then $\gcd(\mathrm{Occ}(P,T))$ is a period of T.

Text T	a b a a b a a b a a b a a b a a b a a b a a b a a b
Pattern P	a b a a b a a b a a b a a b a a b

Exact PM

Periodicity Lemma Readapted [FW65]

If $n \leq 3/2 \cdot m$ and $\{0, n-m\} \subseteq \mathrm{Occ}(P,T)$, then $\gcd(\mathrm{Occ}(P,T))$ is a period of T.

Text T a b a a b a a b a a b a a b a a b a a b a a b a a b

Pattern P a b a a b a a b a a b a a b a a b

Exact PM

Periodicity Lemma Readapted [FW65]

If $n \leq 3/2 \cdot m$ and $\{0, n-m\} \subseteq \mathrm{Occ}(P, T)$, then $\gcd(\mathrm{Occ}(P, T))$ is a period of T.

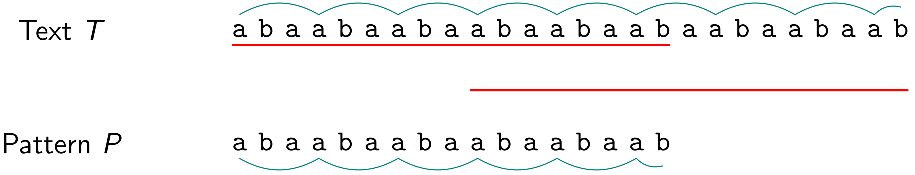
Text T a b a a b a a b a a b a a b a a b a a b a a b a a b

Pattern P a b a a b a a b a a b a a b a a b

Exact PM

Periodicity Lemma Readapted [FW65]

If $n \leq 3/2 \cdot m$ and $\{0, n-m\} \subseteq \mathrm{Occ}(P, T)$, then $\gcd(\mathrm{Occ}(P, T))$ is a period of T.

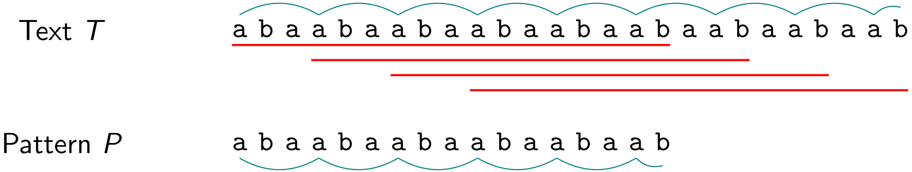


- g is a period of T and P, for $g := \gcd(\mathrm{Occ}(P, T))$.

Exact PM

Periodicity Lemma Readapted [FW65]

If $n \leq 3/2 \cdot m$ and $\{0, n-m\} \subseteq \mathrm{Occ}(P, T)$, then $\gcd(\mathrm{Occ}(P, T))$ is a period of T.

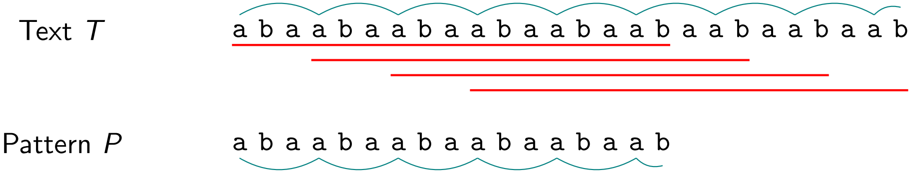


- g is a period of T and P, for $g := \gcd(\mathrm{Occ}(P, T))$.
- $A \subseteq \mathrm{Occ}(P, T)$ for $A := \{0, g, 2g, \ldots, n-m\}$.

Exact PM

Periodicity Lemma Readapted [FW65]

If $n \leq 3/2 \cdot m$ and $\{0, n-m\} \subseteq \mathrm{Occ}(P, T)$, then $\gcd(\mathrm{Occ}(P, T))$ is a period of T.



- g is a period of T and P, for $g := \gcd(\mathrm{Occ}(P, T))$.
- $A \subseteq \mathrm{Occ}(P, T)$ for $A := \{0, g, 2g, \ldots, n-m\}$.
- But for every $x \in \mathrm{Occ}(P, T)$, we have $x \mid g$. Thus, $x \in A$ and $A = \mathrm{Occ}(P, T)$.

Exact PM

Periodicity Lemma Readapted [FW65]

If $n \leq 3/2 \cdot m$ and $\{0, n-m\} \subseteq \mathrm{Occ}(P,T)$, then $\gcd(\mathrm{Occ}(P,T))$ is a period of T.

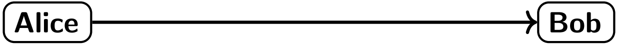
Text T a b a a b a a b a a b a a b a a b a a b a a b a a b

Pattern P a b a a b a a b a a b a a b a a b

- g is a period of T and P, for $g := \gcd(\mathrm{Occ}(P,T))$.
- $A \subseteq \mathrm{Occ}(P,T)$ for $A := \{0, g, 2g, \ldots, n-m\}$.
- But for every $x \in \mathrm{Occ}(P,T)$, we have $x \mid g$. Thus, $x \in A$ and $A = \mathrm{Occ}(P,T)$.
- In order to send A to Bob, it suffices that Alice sends two numbers: g and $|A|$.

Pattern Matching with Mismatches

What Alice Sends

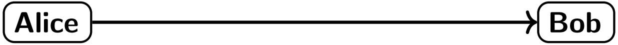


- Alice selects a subset

$$\{0, n-m\} \subseteq S \subseteq \mathrm{Occ}_k^H(P, T)$$

s.t. $\gcd(S) = \gcd(\mathrm{Occ}_k^H(P, T))$.

What Alice Sends



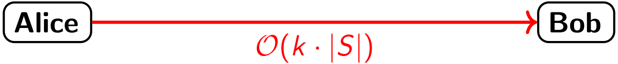
- Alice selects a subset

$$\{0, n-m\} \subseteq S \subseteq \mathrm{Occ}_k^H(P, T)$$

s.t. $\gcd(S) = \gcd(\mathrm{Occ}_k^H(P, T))$.

- Alice sends S and $\mathrm{MI}(x)$ for all $x \in S$.

What Alice Sends



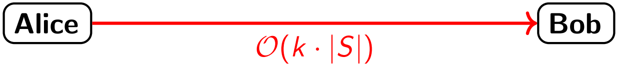
- Alice selects a subset

$$\{0, n-m\} \subseteq S \subseteq \mathrm{Occ}_k^H(P, T)$$

s.t. $\gcd(S) = \gcd(\mathrm{Occ}_k^H(P, T))$.

- Alice sends S and $\mathrm{MI}(x)$ for all $x \in S$.

What Alice Sends



- Alice selects a subset

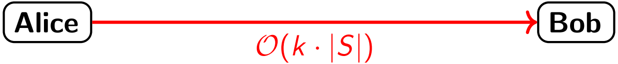
$$\{0, n-m\} \subseteq S \subseteq \mathrm{Occ}_k^H(P, T)$$

s.t. $\gcd(S) = \gcd(\mathrm{Occ}_k^H(P, T))$.

- Alice sends S and $\mathrm{MI}(x)$ for all $x \in S$.

We can choose S s.t. $|S| \leq \mathcal{O}(\log n)$.

What Alice Sends



- Alice selects a subset

$$\{0, n-m\} \subseteq S \subseteq \mathrm{Occ}_k^H(P, T)$$

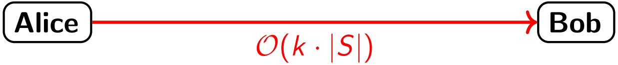
s.t. $\gcd(S) = \gcd(\mathrm{Occ}_k^H(P, T))$.

- Alice sends S and $\mathrm{MI}(x)$ for all $x \in S$.

We can choose S s.t. $|S| \leq \mathcal{O}(\log n)$.

- Construct S iteratively.

What Alice Sends



- Alice selects a subset

$$\{0, n-m\} \subseteq S \subseteq \mathrm{Occ}_k^H(P, T)$$

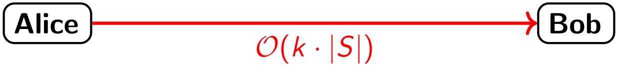
s.t. $\gcd(S) = \gcd(\mathrm{Occ}_k^H(P, T))$.

- Alice sends S and $\mathrm{MI}(x)$ for all $x \in S$.

We can choose S s.t. $|S| \leq \mathcal{O}(\log n)$.

- Construct S iteratively.
- Try to add to S elements from $\mathrm{Occ}_k^H(P, T)$ one by one.

What Alice Sends



- Alice selects a subset

$$\{0, n-m\} \subseteq S \subseteq \mathrm{Occ}_k^H(P, T)$$

s.t. $\gcd(S) = \gcd(\mathrm{Occ}_k^H(P, T))$.

- Alice sends S and $\mathrm{MI}(x)$ for all $x \in S$.

We can choose S s.t. $|S| \leq \mathcal{O}(\log n)$.

- Construct S iteratively.
- Try to add to S elements from $\mathrm{Occ}_k^H(P, T)$ one by one.
- For each element x either $\gcd(S \cup \{x\}) = \gcd(S)$ or $\gcd(S \cup \{x\}) \leq \gcd(S)/2$.

What Alice Sends

- Alice selects a subset

$$\{0, n - m\} \subseteq S \subseteq \mathrm{Occ}_k^H(P, T)$$

s.t. $\gcd(S) = \gcd(\mathrm{Occ}_k^H(P, T))$.

- Alice sends S and $\mathrm{MI}(x)$ for all $x \in S$.

We can choose S s.t. $|S| \leq \mathcal{O}(\log n)$.

- Construct S iteratively.
- Try to add to S elements from $\mathrm{Occ}_k^H(P, T)$ one by one.
- For each element x either $\gcd(S \cup \{x\}) = \gcd(S)$ or $\gcd(S \cup \{x\}) \leq \gcd(S)/2$.

What Alice Sends (Example)

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29
T	b	e	b	e	b	e	b	e	b	e	b	e	b	e	b	a	b	c	b	a	b	c	b	a	b	a	b	a	b	a
P	b	e	b	e	b	e	b	e	b	e	b	e	b	e	b	c	b	a	b	a	b	a	b	a						

$$k = 4$$

What Alice Sends (Example)

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29
T	b	e	b	e	b	e	b	e	b	e	b	e	b	e	b	a	b	c	b	a	b	c	b	a	b	a	b	a	b	a
P	b	e	b	e	b	e	b	e	b	e	b	e	b	e	b	c	b	a	b	a	b	a	b	a						

$$\boldsymbol{k = 4}$$

$$\mathrm{Occ}_k^H(P, T) = \{0\}$$

What Alice Sends (Example)

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29
T	b	e	b	e	b	e	b	e	b	e	b	e	b	e	b	a	b	c	b	a	b	c	b	a	b	a	b	a	b	a
P			b	e	b	e	b	e	b	e	b	e	b	e	b	e	b	c	b	a	b	a	b	a	b	a				

$$\boldsymbol{k = 4}$$

$$\mathrm{Occ}_k^H(P, T) = \{0, 2\}$$

What Alice Sends (Example)

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29
T	b	e	b	e	b	e	b	e	b	e	b	e	b	e	b	a	b	c	b	a	b	c	b	a	b	a	b	a	b	a
P					b	e	b	e	b	e	b	e	b	e	b	e	b	e	b	c	b	a	b	a	b	a	b	a		

$$\boldsymbol{k = 4}$$

$$\mathrm{Occ}_k^H(P, T) = \{0, 2, 4\}$$

What Alice Sends (Example)

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29
T	b	e	b	e	b	e	b	e	b	e	b	e	b	e	b	a	b	c	b	a	b	c	b	a	b	a	b	a	b	a
P							b	e	b	e	b	e	b	e	b	e	b	e	b	e	b	c	b	a	b	a	b	a	b	a

$$\boldsymbol{k = 4}$$

$$\mathrm{Occ}_k^H(P, T) = \{0, 2, 4, 6\}$$

What Alice Sends (Example)

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29
T	b	e	b	e	b	e	b	e	b	e	b	e	b	e	b	a	b	c	b	a	b	c	b	a	b	a	b	a	b	a

$$\boldsymbol{k = 4}$$

$$\mathrm{Occ}_k^H(P, T) = \{0, 2, 4, 6\}$$

Alice sends $S = \{0, 2, 6\}$

What Alice Sends (Example)

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29
T	b	e	b	e	b	e	b	e	b	e	b	e	b	e	b	a	b	c	b	a	b	c	b	a	b	a	b	a	b	a
P	b	e	b	e	b	e	b	e	b	e	b	e	b	e	b	c	b	a	b	a	b	a	b	a						
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23						

$$\boldsymbol{k = 4}$$

$$\mathrm{Occ}_k^H(P, T) = \{0, 2, 4, 6\}$$

Alice sends $S = \{\mathbf{0}, 2, 6\}$
and
$\{(15, \mathrm{c}, \mathrm{a}), (17, \mathrm{a}, \mathrm{c}), (21, \mathrm{a}, \mathrm{c})\}$

What Alice Sends (Example)

```
      0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29
T     b e b e b e b e b e b  e  b  e  b  a  b  c  b  a  b  c  b  a  b  a  b  a  b  a
          | | | | | | | | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
P         b e b e b e b e b  e  b  e  b  e  b  c  b  a  b  a  b  a  b  a
          0 1 2 3 4 5 6 7 8  9  10 11 12 13 14 15 16 17 18 19 20 21 22 23
```

$$\boldsymbol{k = 4}$$

$$\mathrm{Occ}_k^H(P, T) = \{0, 2, 4, 6\}$$

Alice sends $S = \{0, \mathbf{2}, 6\}$

and

$\{(15, \mathrm{c}, \mathrm{a}), (17, \mathrm{a}, \mathrm{c}), (21, \mathrm{a}, \mathrm{c})\}$

$\{(13, \mathrm{e}, \mathrm{a}), (19, \mathrm{a}, \mathrm{c})\}$

What Alice Sends (Example)

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29
T	b	e	b	e	b	e	b	e	b	e	b	e	b	e	b	a	b	c	b	a	b	c	b	a	b	a	b	a	b	a
P							b	e	b	e	b	e	b	e	b	e	b	e	b	e	b	c	b	a	b	a	b	a	b	a
							0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23

$$\boldsymbol{k = 4}$$

$$\mathrm{Occ}_k^H(P, T) = \{0, 2, 4, 6\}$$

Alice sends $S = \{0, 2, \mathbf{6}\}$

and

$\{(15, \mathrm{c}, \mathrm{a}), (17, \mathrm{a}, \mathrm{c}), (21, \mathrm{a}, \mathrm{c})\}$

$\{(13, \mathrm{e}, \mathrm{a}), (19, \mathrm{a}, \mathrm{c})\}$

$\{(9, \mathrm{e}, \mathrm{a}), (11, \mathrm{e}, \mathrm{c}), (13, \mathrm{e}, \mathrm{a})\}$

What Bob Receives (Example)

Bob receives $S = \{0, 2, 6\}$

and

$\{(15, \text{c}, \text{a}), (17, \text{a}, \text{c}), (21, \text{a}, \text{c})\}\}$

$\{(13, \text{e}, \text{a}), (19, \text{a}, \text{c})\}$

$\{(9, \text{e}, \text{a}), (11, \text{e}, \text{c}), (13, \text{e}, \text{a})\}$

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29
T	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
P	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?

What Bob Receives (Example)

Bob receives $S = \{0, 2, 6\}$

and

$\{(15, \text{c}, \text{a}), (17, \text{a}, \text{c}), (21, \text{a}, \text{c})\}$

$\{(13, \text{e}, \text{a}), (19, \text{a}, \text{c})\}$

$\{(9, \text{e}, \text{a}), (11, \text{e}, \text{c}), (13, \text{e}, \text{a})\}$

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29
T	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	a	?	c	?	a	?	c	?	?	?	?	?	?	?	?

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
P	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	c	?	a	?	?	?	a	?	?

What Bob Receives (Example)

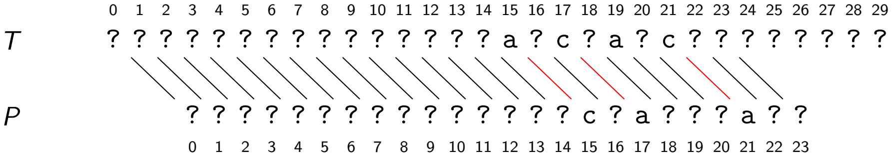
Bob receives $S = \{0, 2, 6\}$

and

$\{(15, \mathrm{c}, \mathrm{a}), (17, \mathrm{a}, \mathrm{c}), (21, \mathrm{a}, \mathrm{c})\}\}$

$\{(13, \mathrm{e}, \mathrm{a}), (19, \mathrm{a}, \mathrm{c})\}$

$\{(9, \mathrm{e}, \mathrm{a}), (11, \mathrm{e}, \mathrm{c}), (13, \mathrm{e}, \mathrm{a})\}$



What Bob Receives (Example)

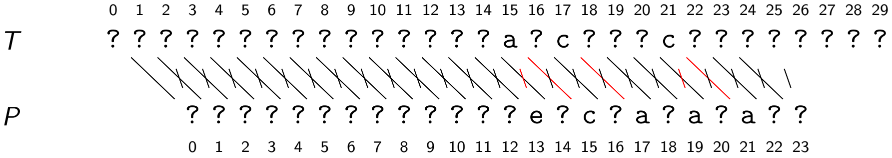
Bob receives $S = \{0, 2, 6\}$

and

$\{(15, \text{c}, \text{a}), (17, \text{a}, \text{c}), (21, \text{a}, \text{c})\}\}$

$\{(13, \text{e}, \text{a}), (19, \text{a}, \text{c})\}$

$\{(9, \text{e}, \text{a}), (11, \text{e}, \text{c}), (13, \text{e}, \text{a})\}$



What Bob Receives (Example)

Bob receives $S = \{0, 2, 6\}$

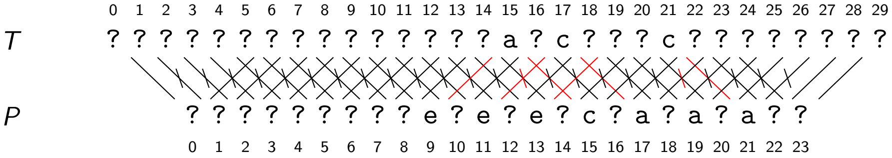
and

$\{(15, \mathrm{c}, \mathrm{a}), (17, \mathrm{a}, \mathrm{c}), (21, \mathrm{a}, \mathrm{c})\}\}$

$\{(13, \mathrm{e}, \mathrm{a}), (19, \mathrm{a}, \mathrm{c})\}$

$\{(9, \mathrm{e}, \mathrm{a}), (11, \mathrm{e}, \mathrm{c}), (13, \mathrm{e}, \mathrm{a})\}$

Inference graph $\mathbf{G}_S$



What Bob Receives (Example)

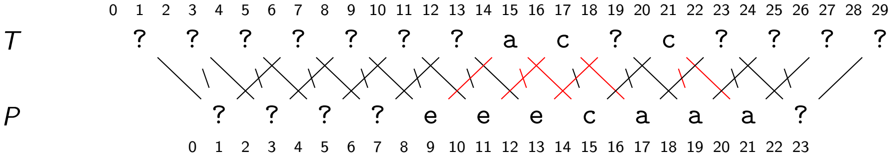
Bob receives $S = \{0, 2, 6\}$

and

$\{(15, \mathsf{c}, \mathsf{a}), (17, \mathsf{a}, \mathsf{c}), (21, \mathsf{a}, \mathsf{c})\}\}$

$\{(13, \mathsf{e}, \mathsf{a}), (19, \mathsf{a}, \mathsf{c})\}$

$\{(9, \mathsf{e}, \mathsf{a}), (11, \mathsf{e}, \mathsf{c}), (13, \mathsf{e}, \mathsf{a})\}$



Red connected component (at least one red edge)

What Bob Receives (Example)

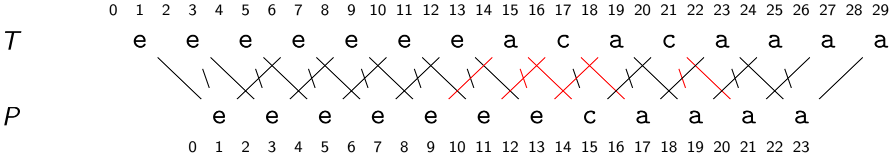
Bob receives $S = \{0, 2, 6\}$

and

$\{(15, \mathrm{c}, \mathrm{a}), (17, \mathrm{a}, \mathrm{c}), (21, \mathrm{a}, \mathrm{c})\}\}$

$\{(13, \mathrm{e}, \mathrm{a}), (19, \mathrm{a}, \mathrm{c})\}$

$\{(9, \mathrm{e}, \mathrm{a}), (11, \mathrm{e}, \mathrm{c}), (13, \mathrm{e}, \mathrm{a})\}$



Red connected component (at least one red edge)

What Bob Receives (Example)

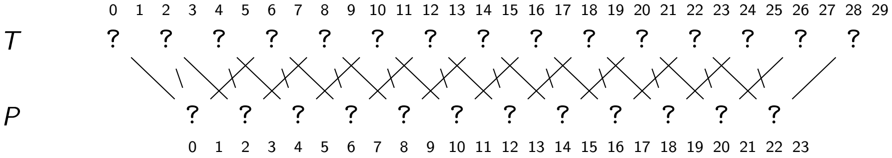
Bob receives $S = \{0, 2, 6\}$

and

$\{(15, \mathrm{c}, \mathrm{a}), (17, \mathrm{a}, \mathrm{c}), (21, \mathrm{a}, \mathrm{c})\}\}$

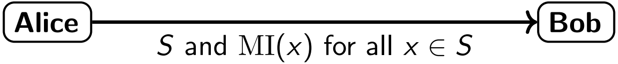
$\{(13, \mathrm{e}, \mathrm{a}), (19, \mathrm{a}, \mathrm{c})\}$

$\{(9, \mathrm{e}, \mathrm{a}), (11, \mathrm{e}, \mathrm{c}), (13, \mathrm{e}, \mathrm{a})\}$



Black connected component (no red edge)

What Bob Receives



- Alice selects a subset

  $$\{0, n-m\} \subseteq S \subseteq \mathrm{Occ}_k^H(P, T)$$

  s.t.
  $\gcd(S) = \gcd(\mathrm{Occ}_k^H(P, T))$.

- Alice sends S and $\mathrm{MI}(x)$ for all $x \in S$.

- Bob receives S and $\mathrm{MI}(x)$ for all $x \in S$.

- Bob constructs the graph $\mathbf{G}_S = (V, E)$:
  - $V = \{t_0, \ldots, t_{n-1}, p_0, \ldots, p_{m-1}\}$, and
  - $\{t_{i+j}, p_i\} \in E$ for all $i \in S, j \in [0..m)$, edge is red if is a mismatch, otw it is **black**.

The Structure of Connected Components in $\mathbf{G}_S$

Periodicity Lemma Readapted [FW65]

If $n \leq 3/2 \cdot m$ and $\{0, n - m\} \subseteq \mathrm{Occ}(P, T)$, then $\gcd(\mathrm{Occ}(P, T))$ is a period of T.

Lemma

Let $g := \gcd(S)$. Then, $\mathbf{G}_S$ has g connected components. Moreover, the i-th connected component for $i \in [0..g)$ contains

$$\{p_j \mid j \equiv_g i\} \cup \{t_j \mid j \equiv_g i\}.$$

The Structure of Connected Components in $\mathbf{G}_S$

Periodicity Lemma Readapted [FW65]

If $n \leq 3/2 \cdot m$ and $\{0, n-m\} \subseteq \mathrm{Occ}(P, T)$, then $\gcd(\mathrm{Occ}(P, T))$ is a period of T.

Lemma

Let $g := \gcd(S)$. Then, $\mathbf{G}_S$ has g connected components. Moreover, the i-th connected component for $i \in [0..g)$ contains

$$\{p_j \mid j \equiv_g i\} \cup \{t_j \mid j \equiv_g i\}.$$

- Construct strings $P^\$, T^\$$ from P, T by replacing each character with a sentinel character unique to the connected component in $\mathbf{G}_S$ the character is contained.

The Structure of Connected Components in $\mathbf{G}_S$

Periodicity Lemma Readapted [FW65]

If $n \leq 3/2 \cdot m$ and $\{0, n-m\} \subseteq \mathrm{Occ}(P, T)$, then $\gcd(\mathrm{Occ}(P, T))$ is a period of T.

Lemma

Let $g := \gcd(S)$. Then, $\mathbf{G}_S$ has g connected components. Moreover, the i-th connected component for $i \in [0..g)$ contains

$$\{p_j \mid j \equiv_g i\} \cup \{t_j \mid j \equiv_g i\}.$$

- Construct strings $P^\$, T^\$$ from P, T by replacing each character with a sentinel character unique to the connected component in $\mathbf{G}_S$ the character is contained.
- For each $x \in S$, we have $x \in \mathrm{Occ}(P^\$, T^\$)$. Thus, $S \subseteq \mathrm{Occ}(P^\$, T^\$)$.

The Structure of Connected Components in $\mathbf{G}_S$

Periodicity Lemma Readapted [FW65]

If $n \leq 3/2 \cdot m$ and $\{0, n-m\} \subseteq \mathrm{Occ}(P, T)$, then $\gcd(\mathrm{Occ}(P, T))$ is a period of T.

Lemma

Let $g := \gcd(S)$. Then, $\mathbf{G}_S$ has g connected components. Moreover, the i-th connected component for $i \in [0..g)$ contains

$$\{p_j \mid j \equiv_g i\} \cup \{t_j \mid j \equiv_g i\}.$$

- Construct strings $P^\$, T^\$$ from P, T by replacing each character with a sentinel character unique to the connected component in $\mathbf{G}_S$ the character is contained.
- For each $x \in S$, we have $x \in \mathrm{Occ}(P^\$, T^\$)$. Thus, $S \subseteq \mathrm{Occ}(P^\$, T^\$)$.
- As $\{0, n-m\} \subseteq \mathrm{Occ}(P^\$, T^\$)$, we can apply the Apply Periodicity Lemma.

The Structure of Connected Components in $\mathbf{G}_S$

Periodicity Lemma Readapted [FW65]

If $n \leq 3/2 \cdot m$ and $\{0, n-m\} \subseteq \mathrm{Occ}(P, T)$, then $\gcd(\mathrm{Occ}(P, T))$ is a period of T.

Lemma

Let $g := \gcd(S)$. Then, $\mathbf{G}_S$ has g connected components. Moreover, the i-th connected component for $i \in [0..g)$ contains

$$\{p_j \mid j \equiv_g i\} \cup \{t_j \mid j \equiv_g i\}.$$

- Construct strings $P^\$, T^\$$ from P, T by replacing each character with a sentinel character unique to the connected component in $\mathbf{G}_S$ the character is contained.
- For each $x \in S$, we have $x \in \mathrm{Occ}(P^\$, T^\$)$. Thus, $S \subseteq \mathrm{Occ}(P^\$, T^\$)$.
- As $\{0, n-m\} \subseteq \mathrm{Occ}(P^\$, T^\$)$, we can apply the Apply Periodicity Lemma.
- The period of $P^\$$ and $T^\$$ is at most $\gcd(\mathrm{Occ}(P^\$, T^\$)) \leq g$

The Structure of Connected Components in $\mathbf{G}_S$

Periodicity Lemma Readapted [FW65]

If $n \leq 3/2 \cdot m$ and $\{0, n-m\} \subseteq \mathrm{Occ}(P, T)$, then $\gcd(\mathrm{Occ}(P, T))$ is a period of T.

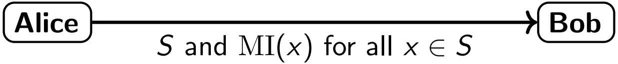
Lemma

Let $g := \gcd(S)$. Then, $\mathbf{G}_S$ has g connected components. Moreover, the i-th connected component for $i \in [0..g)$ contains

$$\{p_j \mid j \equiv_g i\} \cup \{t_j \mid j \equiv_g i\}.$$

- Construct strings $P^\$, T^\$$ from P, T by replacing each character with a sentinel character unique to the connected component in $\mathbf{G}_S$ the character is contained.
- For each $x \in S$, we have $x \in \mathrm{Occ}(P^\$, T^\$)$. Thus, $S \subseteq \mathrm{Occ}(P^\$, T^\$)$.
- As $\{0, n-m\} \subseteq \mathrm{Occ}(P^\$, T^\$)$, we can apply the Apply Periodicity Lemma.
- The period of $P^\$$ and $T^\$$ is at most $\gcd(\mathrm{Occ}(P^\$, T^\$)) \leq g$
- $\mathbf{G}_S$ has at most g connected components.

Bob constructs $P^{\#}$ and $T^{\#}$



- Alice selects a subset

  $$\{0, n-m\} \subseteq S \subseteq \mathrm{Occ}_k^H(P, T)$$

  s.t. $\gcd(S) = \gcd(\mathrm{Occ}_k^H(P, T))$.

- Alice sends S and $\mathrm{MI}(x)$ for all $x \in S$.

- Bob receives S and $\mathrm{MI}(x)$ for all $x \in S$.
- Bob constructs the graph $\mathbf{G}_S = (V, E)$:
  - $V = \{t_0, \ldots, t_{n-1}, p_0, \ldots, p_{m-1}\}$, and
  - $\{t_{i+j}, p_j\} \in E$ for all $i \in S, j \in [0..m)$, edge is red if is a mismatch, otw it is **black**.
- Bob construct strings $P^{\#}, T^{\#}$ from P, T by replacing each character contained in a black component in $\mathbf{G}_S$ with a sentinel character (unique to the component the character is contained).
- Bob computes $\mathrm{Occ}_k^H(P^{\#}, T^{\#})$.

Examples for $P^{\#}$ and $T^{\#}$

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29
$T^{\#}$	#	e	#	e	#	e	#	e	#	e	#	e	#	e	#	a	#	c	#	a	#	c	#	a	#	a	#	a	#	a
$P^{\#}$				#	e	#	e	#	e	#	e	#	e	#	e	#	e	#	c	#	a	#	a	#	a	#	a			
				0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23			

$P^{\#}$ and $T^{\#}$ preserve k-mismatch occurrences

Lemma

$\mathrm{Occ}_k^H(P^{\#}, T^{\#}) = \mathrm{Occ}_k^H(P, T)$.

$P^{\#}$ and $T^{\#}$ preserve k-mismatch occurrences

Lemma

$\mathrm{Occ}_k^H(P^{\#}, T^{\#}) = \mathrm{Occ}_k^H(P, T)$.

- $\mathrm{Occ}_k^H(P, T) \subseteq \mathrm{Occ}_k^H(P^{\#}, T^{\#})$:

$P^{\#}$ and $T^{\#}$ preserve k-mismatch occurrences

Lemma

$\mathrm{Occ}^H_k(P^{\#}, T^{\#}) = \mathrm{Occ}^H_k(P, T)$.

- $\mathrm{Occ}^H_k(P, T) \subseteq \mathrm{Occ}^H_k(P^{\#}, T^{\#})$:
  - $P^{\#}[i] = T^{\#}[j]$ implies $P[i] = T[j]$.

$P^{\#}$ and $T^{\#}$ preserve k-mismatch occurrences

Lemma

$\mathrm{Occ}_k^H(P^{\#}, T^{\#}) = \mathrm{Occ}_k^H(P, T)$.

- $\mathrm{Occ}_k^H(P, T) \subseteq \mathrm{Occ}_k^H(P^{\#}, T^{\#})$:
  - $P^{\#}[i] = T^{\#}[j]$ implies $P[i] = T[j]$.
  - Thus, $\mathrm{HD}(P^{\#}, T^{\#}[i..i+m)) \geq \mathrm{HD}(P, T[i..i+m))$ for all i.

$P^\#$ and $T^\#$ preserve k-mismatch occurrences

Lemma

$\text{Occ}^H_k(P^\#, T^\#) = \text{Occ}^H_k(P, T)$.

- $\text{Occ}^H_k(P, T) \subseteq \text{Occ}^H_k(P^\#, T^\#)$:
  - $P^\#[i] = T^\#[j]$ implies $P[i] = T[j]$.
  - Thus, $\text{HD}(P^\#, T^\#[i..i+m)) \geq \text{HD}(P, T[i..i+m))$ for all i.
- $\text{Occ}^H_k(P^\#, T^\#) \subseteq \text{Occ}^H_k(P, T)$:

$P^{\#}$ and $T^{\#}$ preserve k-mismatch occurrences

Lemma

$\mathrm{Occ}_k^H(P^{\#}, T^{\#}) = \mathrm{Occ}_k^H(P, T)$.

- $\mathrm{Occ}_k^H(P, T) \subseteq \mathrm{Occ}_k^H(P^{\#}, T^{\#})$:
  - $P^{\#}[i] = T^{\#}[j]$ implies $P[i] = T[j]$.
  - Thus, $\mathrm{HD}(P^{\#}, T^{\#}[i..i+m)) \geq \mathrm{HD}(P, T[i..i+m))$ for all i.
- $\mathrm{Occ}_k^H(P^{\#}, T^{\#}) \subseteq \mathrm{Occ}_k^H(P, T)$:
  - Fix $i \in \mathrm{Occ}_k^H(P, T)$ and $j \in [0..m)$

$P^{\#}$ and $T^{\#}$ preserve k-mismatch occurrences

Lemma

$\mathrm{Occ}_k^H(P^{\#}, T^{\#}) = \mathrm{Occ}_k^H(P, T)$.

- $\mathrm{Occ}_k^H(P, T) \subseteq \mathrm{Occ}_k^H(P^{\#}, T^{\#})$:
  - $P^{\#}[i] = T^{\#}[j]$ implies $P[i] = T[j]$.
  - Thus, $\mathrm{HD}(P^{\#}, T^{\#}[i..i+m)) \geq \mathrm{HD}(P, T[i..i+m))$ for all i.
- $\mathrm{Occ}_k^H(P^{\#}, T^{\#}) \subseteq \mathrm{Occ}_k^H(P, T)$:
  - Fix $i \in \mathrm{Occ}_k^H(P, T)$ and $j \in [0..m)$
  - As $i \mid g$ for $g = \gcd(S) = \gcd(\mathrm{Occ}_k^H(P, T))$, we have $j \equiv_g i + j$.

$P^{\#}$ and $T^{\#}$ preserve k-mismatch occurrences

Lemma

$\mathrm{Occ}_k^H(P^{\#}, T^{\#}) = \mathrm{Occ}_k^H(P, T)$.

- $\mathrm{Occ}_k^H(P, T) \subseteq \mathrm{Occ}_k^H(P^{\#}, T^{\#})$:
  - $P^{\#}[i] = T^{\#}[j]$ implies $P[i] = T[j]$.
  - Thus, $\mathrm{HD}(P^{\#}, T^{\#}[i..i+m)) \geq \mathrm{HD}(P, T[i..i+m))$ for all i.
- $\mathrm{Occ}_k^H(P^{\#}, T^{\#}) \subseteq \mathrm{Occ}_k^H(P, T)$:
  - Fix $i \in \mathrm{Occ}_k^H(P, T)$ and $j \in [0..m)$
  - As $i \mid g$ for $g = \gcd(S) = \gcd(\mathrm{Occ}_k^H(P, T))$, we have $j \equiv_g i + j$.
  - Thus, p_j and t_{j+i} are contained in the same connected component:
    - If the component is black, then $P^{\#}[j] = T^{\#}[j+i]$ and $P[j] = T[j+i]$.
    - If the component is red, then $P^{\#}[j] = P[j]$ and $P^{\#}[i+j] = P[i+j]$.

$P^{\#}$ and $T^{\#}$ preserve k-mismatch occurrences

Lemma

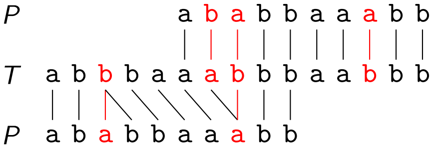
$\mathrm{Occ}_k^H(P^{\#}, T^{\#}) = \mathrm{Occ}_k^H(P, T)$.

- $\mathrm{Occ}_k^H(P, T) \subseteq \mathrm{Occ}_k^H(P^{\#}, T^{\#})$:
  - $P^{\#}[i] = T^{\#}[j]$ implies $P[i] = T[j]$.
  - Thus, $\mathrm{HD}(P^{\#}, T^{\#}[i..i+m)) \geq \mathrm{HD}(P, T[i..i+m))$ for all i.
- $\mathrm{Occ}_k^H(P^{\#}, T^{\#}) \subseteq \mathrm{Occ}_k^H(P, T)$:
  - Fix $i \in \mathrm{Occ}_k^H(P, T)$ and $j \in [0..m)$
  - As $i \mid g$ for $g = \gcd(S) = \gcd(\mathrm{Occ}_k^H(P, T))$, we have $j \equiv_g i + j$.
  - Thus, p_j and t_{j+i} are contained in the same connected component:
    - If the component is black, then $P^{\#}[j] = T^{\#}[j+i]$ and $P[j] = T[j+i]$.
    - If the component is red, then $P^{\#}[j] = P[j]$ and $P^{\#}[i+j] = P[i+j]$.
  - As j was arbitrary, $\mathrm{HD}(P^{\#}, T^{\#}[i..i+m)) = \mathrm{HD}(P, T[i..i+m))$ and $i \in \mathrm{Occ}_k^H(P^{\#}, T^{\#})$.

Pattern Matching with Edits

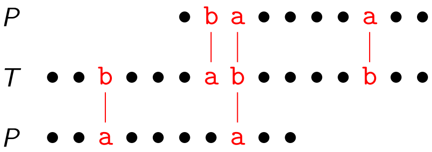
Finding a Period Structure

Suppose Alice takes a set S of alignments of cost at most k,



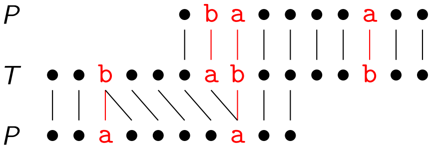
Finding a Period Structure

Suppose Alice takes a set S of alignments of cost at most k, and sends to Bob only the information about edits.



Finding a Period Structure

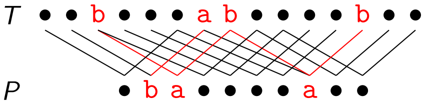
Suppose Alice takes a set S of alignments of cost at most k, and sends to Bob only the information about edits.



- Bob reconstructs the alignments in S.

Finding a Period Structure

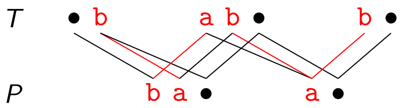
Suppose Alice takes a set S of alignments of cost at most k, and sends to Bob only the information about edits.



- Bob reconstructs the alignments in S.
- Bob makes a graph out of it.

Finding a Period Structure

Suppose Alice takes a set S of alignments of cost at most k, and sends to Bob only the information about edits.

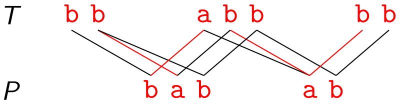


- Bob reconstructs the alignments in S.
- Bob makes a graph out of it.
- Bob selects red connected components,

.

Finding a Period Structure

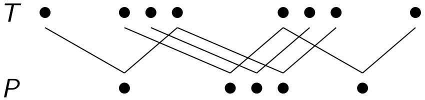
Suppose Alice takes a set S of alignments of cost at most k, and sends to Bob only the information about edits.



- Bob reconstructs the alignments in S.
- Bob makes a graph out of it.
- Bob selects red connected components, and propagates characters in them.

Finding a Period Structure

Suppose Alice takes a set S of alignments of cost at most k, and sends to Bob only the information about edits.

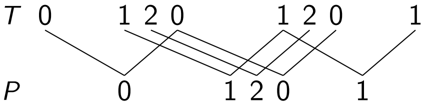


- Bob reconstructs the alignments in S.
- Bob makes a graph out of it.
- Bob selects red connected components, and propagates characters in them.
- Bob selects black connected components,

.

Finding a Period Structure

Suppose Alice takes a set S of alignments of cost at most k, and sends to Bob only the information about edits.



- Bob reconstructs the alignments in S.
- Bob makes a graph out of it.
- Bob selects red connected components, and propagates characters in them.
- Bob selects black connected components, and numbers them.

Finding a Period Structure

Suppose Alice takes a set S of alignments of cost at most k, and sends to Bob only the information about edits.

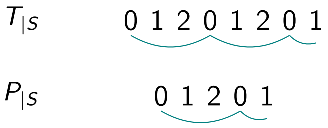
T 0 1 2 0 1 2 0 1

P 0 1 2 0 1

- Bob reconstructs the alignments in S.
- Bob makes a graph out of it.
- Bob selects red connected components, and propagates characters in them.
- Bob selects black connected components, and numbers them.

Finding a Period Structure

Suppose Alice takes a set S of alignments of cost at most k, and sends to Bob only the information about edits.



- Bob reconstructs the alignments in S.
- Bob makes a graph out of it.
- Bob selects red connected components, and propagates characters in them.
- Bob selects black connected components, and numbers them.

Mapping Back the Periodic Structure to the Original Strings

$T_{|S}$ **0 1 2 3 4 5 6 7 8 0 1 2 3 4 5 6 7 8 0 1 2 3 4 5 6 7 8 0 1 2 3 4 5 6 7 8 0 1 2 3 4 5**

$P_{|S}$ **0 1 2 3 4 5 6 7 8 0 1 2 3 4 5 6 7 8 0 1 2 3 4 5**

Mapping Back the Periodic Structure to the Original Strings

$T_{\|S}$	0 1 2 3 4 5 6 7 8 0 1 2 3 4 5 6 7 8 0 1 2 3 4 5 6 7 8 0 1 2 3 4 5 6 7 8 0 1 2 3 4 5
T	b 0 1 2 3 4 5 6 7 8 a 0 1 2 3 b 4 5 6 7 8 0 1 2 3 a a 4 5 6 7 8 a 0 1 2 3 4 5 6 7 8 0 1 2 a 3 4 5
$P_{\|S}$	0 1 2 3 4 5 6 7 8 0 1 2 3 4 5 6 7 8 0 1 2 3 4 5
P	a a 0 1 2 3 4 5 6 7 8 0 1 2 3 b b 4 5 6 7 8 0 1 2 3 a 4 5

Mapping Back the Periodic Structure to the Original Strings

$T_{|S}$ 0 1 2 3 4 5 6 7 8 0 1 2 3 4 5 6 7 8 0 1 2 3 4 5 6 7 8 0 1 2 3 4 5 6 7 8 0 1 2 3 4 5

T b 0 1 2 3 4 5 6 7 8 a 0 1 2 3 b 4 5 6 7 8 0 1 2 3 a a 4 5 6 7 8 a 0 1 2 3 4 5 6 7 8 0 1 2 a 3 4 5

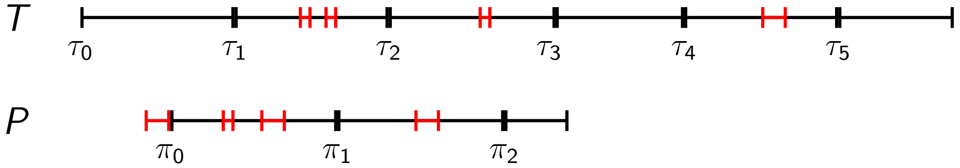
τ_0 τ_1 τ_2 τ_3 τ_4

$P_{|S}$ 0 1 2 3 4 5 6 7 8 0 1 2 3 4 5 6 7 8 0 1 2 3 4 5

P a a 0 1 2 3 4 5 6 7 8 0 1 2 3 b b 4 5 6 7 8 0 1 2 3 a 4 5

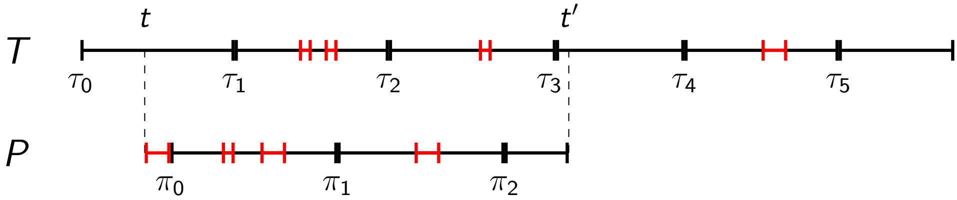
π_0 π_1 π_2

Alignments Covered by S



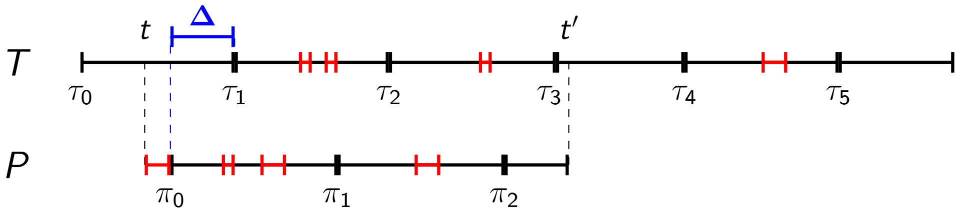
Alignments Covered by S

- Consider an arbitrary alignment $\mathcal{X} : P \rightsquigarrow T[\,t\,..\,t'\,)$ of cost at most k.



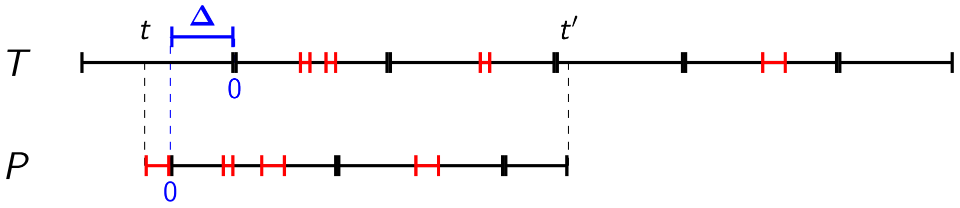
Alignments Covered by S

- Consider an arbitrary alignment $\mathcal{X} : P \rightsquigarrow T[\, t \mathinner{.\,.} t' \,)$ of cost at most k.
- Compute $\boldsymbol{\Delta = \min_i |\tau_i - t - \pi_0|}$.



Alignments Covered by S

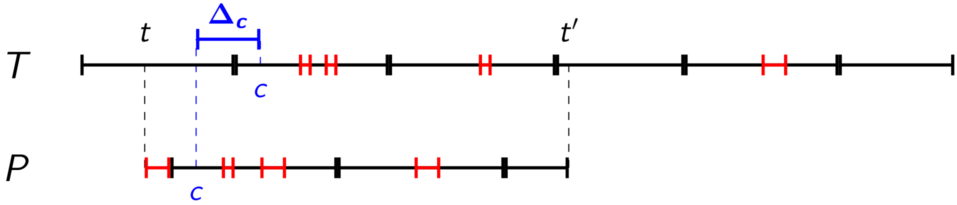
- Consider an arbitrary alignment $\mathcal{X} : P \rightsquigarrow T[\,t \,..\, t'\,)$ of cost at most k.
- Compute $\boldsymbol{\Delta = \min_i |\tau_i - t - \pi_0|}$.



Case $\Delta = \widetilde{\Omega}(k)$ (large)

Alignments Covered by S

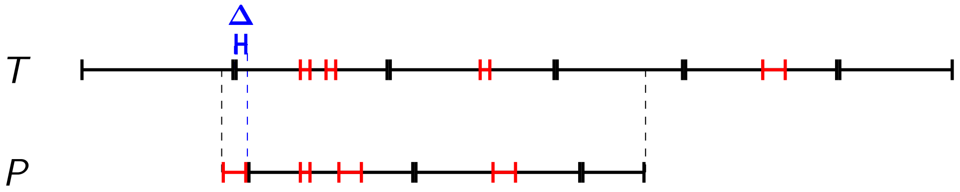
- Consider an arbitrary alignment $\mathcal{X} : P \rightsquigarrow T[\,t\,..\,t'\,)$ of cost at most k.
- Compute $\boldsymbol{\Delta = \min_i |\tau_i - t - \pi_0|}$.



Case $\Delta = \widetilde{\Omega}(k)$ (large) $\implies$ if $\mathcal{X}$ is added to S, the # of black components at least halves.

Alignments Covered by S

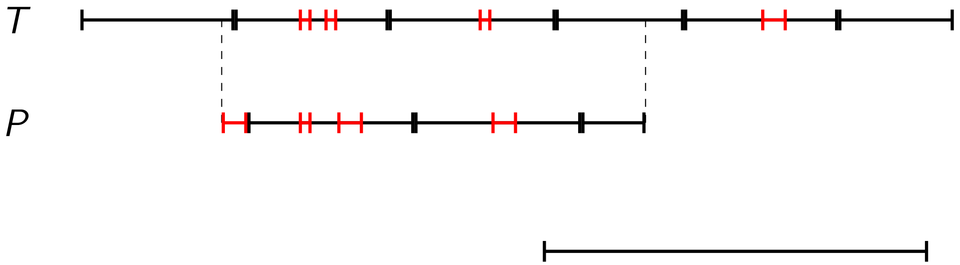
- Consider an arbitrary alignment $\mathcal{X} : P \rightsquigarrow T[\, t \,..\, t' \,)$ of cost at most k.
- Compute $\boldsymbol{\Delta = \min_i |\tau_i - t - \pi_0|}$.



Case $\Delta = \widetilde{\mathcal{O}}(k)$ (small)

Alignments Covered by S

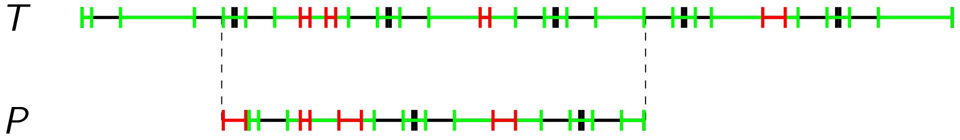
- Consider an arbitrary alignment $\mathcal{X} : P \rightsquigarrow T[\,t\,..\,t'\,)$ of cost at most k.
- Compute $\boldsymbol{\Delta = \min_i |\tau_i - t - \pi_0|}$.



Case $\boldsymbol{\Delta} = \widetilde{\mathcal{O}}(k)$ (small)

Alignments Covered by S

- Consider an arbitrary alignment $\mathcal{X} : P \rightsquigarrow T[\,t\,..\,t'\,)$ of cost at most k.
- Compute $\boldsymbol{\Delta = \min_i |\tau_i - t - \pi_0|}$.

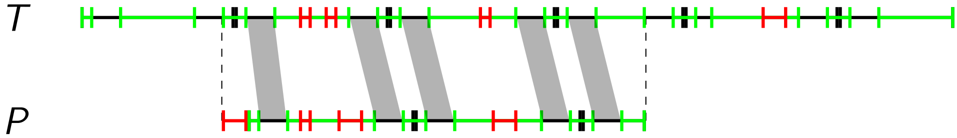


encoding cost $= \widetilde{\mathcal{O}}(k)$

Case $\Delta = \widetilde{\mathcal{O}}(k)$ (small)

Alignments Covered by S

- Consider an arbitrary alignment $\mathcal{X} : P \rightsquigarrow T[\, t \mathinner{.\,.} t' \,)$ of cost at most k.
- Compute $\boldsymbol{\Delta = \min_i |\tau_i - t - \pi_0|}$.



encoding cost $= \widetilde{\mathcal{O}}(k)$

Case $\boldsymbol{\Delta} = \widetilde{\mathcal{O}}(k)$ (small) $\Longrightarrow$ there exists alignment with the same cost of $\mathcal{X}$ that matches characters in the same uncovered black components.

Algorithmic Applications: Compressed Construction of $P^{\#}$ and $T^{\#}$

Compressibility of $P^{\#}$ and $T^{\#}$

- Let's take a step back.

Compressibility of $P^{\#}$ and $T^{\#}$

- Let's take a step back.
- What we saw for mismatches:
  1. Bob receives a set $S \subseteq \mathsf{Occ}_k^H(P, H)$ s.t. $|S| = \mathcal{O}(\log n)$ and $x \in \mathrm{MI}(x)$ for all $x \in S$.

Compressibility of $P^{\#}$ and $T^{\#}$

- Let's take a step back.
- What we saw for mismatches:
    1. Bob receives a set $S \subseteq \text{Occ}_k^H(P, H)$ s.t. $|S| = \mathcal{O}(\log n)$ and $x \in \mathrm{MI}(x)$ for all $x \in S$.
    2. Bob constructs graph $\mathbf{G}_S$, propagates characters through red components, and replaces sentinel characters in black component (unique to the connected component).

Compressibility of $P^{\#}$ and $T^{\#}$

- Let's take a step back.
- What we saw for mismatches:
  1. Bob receives a set $S \subseteq \mathsf{Occ}_k^H(P, H)$ s.t. $|S| = \mathcal{O}(\log n)$ and $x \in \mathrm{MI}(x)$ for all $x \in S$.
  2. Bob constructs graph $\mathbf{G}_S$, propagates characters through red components, and replaces sentinel characters in black component (unique to the connected component).
  3. Bob obtains string $P^{\#}$ and $T^{\#}$ equivalent to P and T w.r.t. PM with k mismatches.

Compressibility of $P^{\#}$ and $T^{\#}$

- Let's take a step back.
- What we saw for mismatches:
  1. Bob receives a set $S \subseteq \mathrm{Occ}_k^H(P, H)$ s.t. $|S| = \mathcal{O}(\log n)$ and $x \in \mathrm{MI}(x)$ for all $x \in S$.
  2. Bob constructs graph $\mathbf{G}_S$, propagates characters through red components, and replaces sentinel characters in black component (unique to the connected component).
  3. Bob obtains string $P^{\#}$ and $T^{\#}$ equivalent to P and T w.r.t. PM with k mismatches.
- $P^{\#}$ and $T^{\#}$ have low space representation of $\widetilde{\mathcal{O}}(k)$.

Compressibility of $P^{\#}$ and $T^{\#}$

- Let's take a step back.
- What we saw for mismatches:
  1. Bob receives a set $S \subseteq \mathrm{Occ}_k^H(P, H)$ s.t. $|S| = \mathcal{O}(\log n)$ and $x \in \mathrm{MI}(x)$ for all $x \in S$.
  2. Bob constructs graph $\mathbf{G}_S$, propagates characters through red components, and replaces sentinel characters in black component (unique to the connected component).
  3. Bob obtains string $P^{\#}$ and $T^{\#}$ equivalent to P and T w.r.t. PM with k mismatches.
- $P^{\#}$ and $T^{\#}$ have low space representation of $\widetilde{\mathcal{O}}(k)$.
- **But naive construction is linear in time! Can we do better?**

Compressibility of $P^{\#}$ and $T^{\#}$

- Let's take a step back.
- What we saw for mismatches:
  1. Bob receives a set $S \subseteq \text{Occ}_k^H(P, H)$ s.t. $|S| = \mathcal{O}(\log n)$ and $x \in \mathrm{MI}(x)$ for all $x \in S$.
  2. Bob constructs graph $\mathbf{G}_S$, propagates characters through red components, and replaces sentinel characters in black component (unique to the connected component).
  3. Bob obtains string $P^{\#}$ and $T^{\#}$ equivalent to P and T w.r.t. PM with k mismatches.
- $P^{\#}$ and $T^{\#}$ have low space representation of $\widetilde{\mathcal{O}}(k)$.
- **But naive construction is linear in time! Can we do better?**
- **Can we have fast construction of low-space representation of $P^{\#}, T^{\#}$, e.g. using grammars?**

Construction of $P^{\#}$ and $T^{\#}$

Theorem [KNW25]

Given S and $\mathrm{MI}(x)$ for all $x \in S$, we can construct a grammar-like representation $P^{\#}$ and $T^{\#}$ of size $\widetilde{\mathcal{O}}(k)$ in time $\widetilde{\mathcal{O}}(k^2)$. The grammar supports $\widetilde{\mathcal{O}}(1)$ time `PILLAR` operations.

Construction of $P^{\#}$ and $T^{\#}$

Theorem [KNW25]

Given S and $\mathrm{MI}(x)$ for all $x \in S$, we can construct a grammar-like representation $P^{\#}$ and $T^{\#}$ of size $\widetilde{\mathcal{O}}(k)$ in time $\widetilde{\mathcal{O}}(k^2)$. The grammar supports $\widetilde{\mathcal{O}}(1)$ time `PILLAR` operations.

- `PILLAR` operations: longest common prefix, internal pattern matching queries, etc...

Construction of $P^{\#}$ and $T^{\#}$

Theorem [KNW25]

Given S and $\mathrm{MI}(x)$ for all $x \in S$, we can construct a grammar-like representation $P^{\#}$ and $T^{\#}$ of size $\widetilde{\mathcal{O}}(k)$ in time $\widetilde{\mathcal{O}}(k^2)$. The grammar supports $\widetilde{\mathcal{O}}(1)$ time `PILLAR` operations.

- `PILLAR` operations: longest common prefix, internal pattern matching queries, etc...
- **[CKW20]**: output (representation of) $\mathrm{Occ}_k^H(P, T)$ using $\mathcal{O}(k^2)$ `PILLAR` operations.

Construction of $P^\#$ and $T^\#$

Theorem [KNW25]

Given S and $\mathrm{MI}(x)$ for all $x \in S$, we can construct a grammar-like representation $P^\#$ and $T^\#$ of size $\widetilde{\mathcal{O}}(k)$ in time $\widetilde{\mathcal{O}}(k^2)$. The grammar supports $\widetilde{\mathcal{O}}(1)$ time `PILLAR` operations.

- `PILLAR` operations: longest common prefix, internal pattern matching queries, etc...
- **[CKW20]**: output (representation of) $\mathrm{Occ}_k^H(P, T)$ using $\mathcal{O}(k^2)$ `PILLAR` operations.
- This means:
  1. Bob can construct a grammar for $P^\#$ and $T^\#$ in $\widetilde{\mathcal{O}}(k^2)$ time, and

Construction of $P^\#$ and $T^\#$

Theorem [KNW25]

Given S and $\mathrm{MI}(x)$ for all $x \in S$, we can construct a grammar-like representation $P^\#$ and $T^\#$ of size $\widetilde{\mathcal{O}}(k)$ in time $\widetilde{\mathcal{O}}(k^2)$. The grammar supports $\widetilde{\mathcal{O}}(1)$ time `PILLAR` operations.

- `PILLAR` operations: longest common prefix, internal pattern matching queries, etc...
- **[CKW20]**: output (representation of) $\mathrm{Occ}_k^H(P, T)$ using $\mathcal{O}(k^2)$ `PILLAR` operations.
- This means:
  1. Bob can construct a grammar for $P^\#$ and $T^\#$ in $\widetilde{\mathcal{O}}(k^2)$ time, and
  2. Bob can compute $\mathrm{Occ}_k^H(P^\#, T^\#) = \mathrm{Occ}_k^H(P, T)$ in $\widetilde{\mathcal{O}}(k^2)$ time.

Construction of $P^{\#}$ and $T^{\#}$

Theorem [KNW25]

Given S and $\mathrm{MI}(x)$ for all $x \in S$, we can construct a grammar-like representation $P^{\#}$ and $T^{\#}$ of size $\widetilde{\mathcal{O}}(k)$ in time $\widetilde{\mathcal{O}}(k^2)$. The grammar supports $\widetilde{\mathcal{O}}(1)$ time PILLAR operations.

- PILLAR operations: longest common prefix, internal pattern matching queries, etc...
- **[CKW20]**: output (representation of) $\mathrm{Occ}_k^H(P, T)$ using $\mathcal{O}(k^2)$ PILLAR operations.
- This means:
  1. Bob can construct a grammar for $P^{\#}$ and $T^{\#}$ in $\widetilde{\mathcal{O}}(k^2)$ time, and
  2. Bob can compute $\mathrm{Occ}_k^H(P^{\#}, T^{\#}) = \mathrm{Occ}_k^H(P, T)$ in $\widetilde{\mathcal{O}}(k^2)$ time.

Theorem [KNW25]

Given N equality equations of the form $X[i..j) = X[i'..j')$ on a length-n string X, we can construct in time $\widetilde{\mathcal{O}}(N^2)$ a grammar-like representation of size $\widetilde{\mathcal{O}}(N)$ of a strings Y which:

1. satisfies all N equations, and
2. $Y[i] = Y[j]$ only when dictated by the equations.

Algorithmic Applications: Quantum Algorithms

Turning the Previous Ideas into an Algorithm

- Instead of computing directly $\mathrm{Occ}_k^H(P, T)$ compute in through $\mathrm{Occ}_k^H(P^\#, T^\#)$.

Turning the Previous Ideas into an Algorithm

- Instead of computing directly $\mathrm{Occ}_k^H(P, T)$ compute in through $\mathrm{Occ}_k^H(P^{\#}, T^{\#})$.
- **Problem:** constructing $P^{\#}, T^{\#}$

  requires S s.t. $\{0, n-m\} \subseteq S \subseteq \mathrm{Occ}_k^H(P, T)$ and $\gcd(S) = \gcd(\mathrm{Occ}_k^H(P, T))$.

Turning the Previous Ideas into an Algorithm

- Instead of computing directly $\mathrm{Occ}_k^H(P, T)$ compute in through $\mathrm{Occ}_k^H(P^\#, T^\#)$.
- **Problem:** constructing $P^\#, T^\#$

  requires S s.t. $\{0, n-m\} \subseteq S \subseteq \mathrm{Occ}_k^H(P, T)$ and $\gcd(S) = \gcd(\mathrm{Occ}_k^H(P, T))$.

  But $\mathbf{Occ}_k^H(P, T)$ is exactly what we want to compute!

Turning the Previous Ideas into an Algorithm

- Instead of computing directly $\mathrm{Occ}_k^H(P, T)$ compute in through $\mathrm{Occ}_k^H(P^\#, T^\#)$.
- **Problem:** constructing $P^\#, T^\#$

  requires S s.t. $\{0, n-m\} \subseteq S \subseteq \mathrm{Occ}_k^H(P, T)$ and $\gcd(S) = \gcd(\mathrm{Occ}_k^H(P, T))$.

  But $\mathrm{Occ}_k^H(P, T)$ is exactly what we want to compute!
- **Workaround:**
  - Find $\mathrm{Occ}_k^H(P, T) \subseteq C \subseteq \mathrm{Occ}_{5k}^H(P, T)$
  - Compute S s.t. $\{0, n-m\} \subseteq S \subseteq C$ and $\gcd(S) = \gcd(C)$

Constructing S through a Candidate Set

[CKW20]: Find a candidate set $\mathrm{Occ}_k^H(P, T) \subseteq C$ in one of two forms.

- $|C| = \mathcal{O}(k)$
- C forms an arithmetic progression and $C \subseteq \mathrm{Occ}_{5k}^H(P, T)$

Constructing S through a Candidate Set

[CKW20]: Find a candidate set $\mathrm{Occ}_k^H(P,T) \subseteq C$ in one of two forms.

$|C| = \mathcal{O}(k)$

For each $x \in C$, distinguish between $x \in \mathrm{Occ}_k^H(P,T)$ and $x \notin \mathrm{Occ}_{5k}^H(P,T)$

C forms an arithmetic progression and $C \subseteq \mathrm{Occ}_{5k}^H(P,T)$

Constructing S through a Candidate Set

[CKW20]: Find a candidate set $\mathrm{Occ}_k^H(P,T) \subseteq C$ in one of two forms.

$|C| = \mathcal{O}(k)$

For each $x \in C$, distinguish between $x \in \mathrm{Occ}_k^H(P,T)$ and $x \notin \mathrm{Occ}_{5k}^H(P,T)$

C forms an arithmetic progression and $C \subseteq \mathrm{Occ}_{5k}^H(P,T)$

The set C satisfies $\mathrm{Occ}_k^H(P,T) \subseteq C \subseteq \mathrm{Occ}_{5k}^H(P,T)$.
Choose $\{0, n-m\} \subseteq S \subseteq C$ s.t. $\gcd(S) = \gcd(C)$ and $|S| = \mathcal{O}(\log n)$.

Construct compressed $P^{\#}$ and $T^{\#}$ and compute $\mathrm{Occ}_k^H(P^{\#}, T^{\#})$.

Our Results

We apply our communication complexity results to the quantum setting:

- Input string S given as oracle where queries can be made in **superposition**

Our Results

We apply our communication complexity results to the quantum setting:

- Input string S given as oracle where queries can be made in **superposition**
- **Query complexity** $Q(n)$: counts number of queries to oracle

Our Results

We apply our communication complexity results to the quantum setting:

- Input string S given as oracle where queries can be made in **superposition**
- **Query complexity** $Q(n)$: counts number of queries to oracle
- **Time complexity** $T(n)$: also counts the number of elementary gates

Our Results

We apply our communication complexity results to the quantum setting:

- Input string S given as oracle where queries can be made in **superposition**
- **Query complexity** $Q(n)$: counts number of queries to oracle
- **Time complexity** $T(n)$: also counts the number of elementary gates

	Query Complexity	**Time Complexity**	**Refererence**
PM with mismatches	$\widehat{\mathcal{O}}(k^{3/4}\sqrt{n})$	$\widehat{\mathcal{O}}(k\sqrt{n})$	**[JN23]**

Our Results

We apply our communication complexity results to the quantum setting:

- Input string S given as oracle where queries can be made in **superposition**
- **Query complexity** $Q(n)$: counts number of queries to oracle
- **Time complexity** $T(n)$: also counts the number of elementary gates

	Query Complexity	Time Complexity	Refererence
PM with mismatches	$\widehat{\mathcal{O}}(k^{3/4}\sqrt{n})$	$\widehat{\mathcal{O}}(k\sqrt{n})$	[JN23]
PM with edits	$\widehat{\mathcal{O}}(\sqrt{kn})$	$\widehat{\mathcal{O}}(\sqrt{k}n + k^{3.5})$	[KNW24]

Our Results

We apply our communication complexity results to the quantum setting:

- Input string S given as oracle where queries can be made in **superposition**
- **Query complexity** $Q(n)$: counts number of queries to oracle
- **Time complexity** $T(n)$: also counts the number of elementary gates

	Query Complexity	**Time Complexity**	**Refererence**
PM with mismatches	$\widehat{\mathcal{O}}(k^{3/4}\sqrt{n})$	$\widehat{\mathcal{O}}(k\sqrt{n})$	**[JN23]**
PM with edits	$\widehat{\mathcal{O}}(\sqrt{kn})$	$\widehat{\mathcal{O}}(\sqrt{kn} + k^{3.5})$	**[KNW24]**
PM with mismatches	$\widetilde{\mathcal{O}}(\sqrt{kn})$	$\widetilde{\mathcal{O}}(\sqrt{kn} + k^{2})$	**[KNW25]**

Our Results

We apply our communication complexity results to the quantum setting:

- Input string S given as oracle where queries can be made in **superposition**
- **Query complexity** $Q(n)$: counts number of queries to oracle
- **Time complexity** $T(n)$: also counts the number of elementary gates

	Query Complexity	**Time Complexity**	**Refererence**
PM with mismatches	$\widehat{\mathcal{O}}(k^{3/4}\sqrt{n})$	$\widehat{\mathcal{O}}(k\sqrt{n})$	**[JN23]**
PM with edits	$\widehat{\mathcal{O}}(\sqrt{kn})$	$\widehat{\mathcal{O}}(\sqrt{k}n + k^{3.5})$	**[KNW24]**
PM with mismatches	$\widetilde{\mathcal{O}}(\sqrt{kn})$	$\widetilde{\mathcal{O}}(\sqrt{kn} + k^2)$	**[KNW25]**
PM with edits	$\widehat{\mathcal{O}}(\sqrt{kn})$	$\widehat{\mathcal{O}}(\sqrt{kn} + k^{3.5})$	**[KNW25]**

Our Results

We apply our communication complexity results to the quantum setting:

- Input string S given as oracle where queries can be made in **superposition**
- **Query complexity** $Q(n)$: counts number of queries to oracle
- **Time complexity** $T(n)$: also counts the number of elementary gates

	Query Complexity	**Time Complexity**	**Refererence**
PM with mismatches	$\widehat{\mathcal{O}}(k^{3/4}\sqrt{n})$	$\widehat{\mathcal{O}}(k\sqrt{n})$	**[JN23]**
PM with edits	$\widehat{\mathcal{O}}(\sqrt{kn})$	$\widehat{\mathcal{O}}(\sqrt{kn} + k^{3.5})$	**[KNW24]**
PM with mismatches	$\widetilde{\mathcal{O}}(\sqrt{kn})$	$\widetilde{\mathcal{O}}(\sqrt{kn} + k^{2})$	**[KNW25]**
PM with edits	$\widehat{\mathcal{O}}(\sqrt{kn})$	$\widehat{\mathcal{O}}(\sqrt{kn} + k^{3.5})$	**[KNW25]**

➤ The number of queries are optimal for $k = o(n)$ (up to a logarithmic / subpolynomial factors).

Thanks!