The Dynamic k-Mismatch Problem

Raphaël Clifford   Paweł Gawrychowski   **Tomasz Kociumaka**
Daniel P. Martin   Przemysław Uznański

University of BRISTOL
Uniwersytet Wrocławski
max planck institut informatik
The Alan Turing Institute
Uniwersytet Wrocławski

CPM 2022, Prague, June 28th, 2022

The k-mismatch problem

The k-mismatch problem

Given a pattern P of length m, a text T of length n, and a **threshold** k, for each position $r \leq n - m$, compute the **Hamming distance** $\mathrm{HD}(P, T[r \mathinner{.\,.} r + m))$ **if it does not exceed** k.

$\mathbf{k = 2}$

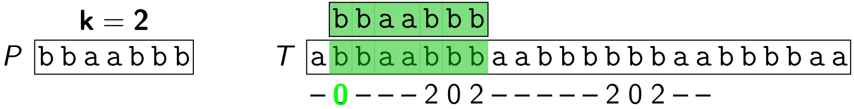
P `b b a a b b b` T `a b b a a b b b a a b b b b b b a a b b b b a a`

– 0 – – – 2 0 2 – – – – – 2 0 2 – –

The k-mismatch problem

The k-mismatch problem

Given a pattern P of length m, a text T of length n, and a **threshold** k, for each position $r \leq n - m$, compute the **Hamming distance** $\mathrm{HD}(P, T[r\,..\,r+m))$ **if it does not exceed** k.



$\mathbf{k = 2}$

b b a a b b b

P b b a a b b b

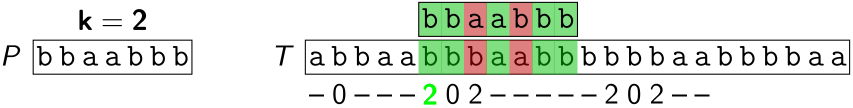
T a b b a a b b b a a b b b b b b a a b b b b a a

− **0** − − − 2 0 2 − − − − − 2 0 2 − −

The k-mismatch problem

The k-mismatch problem

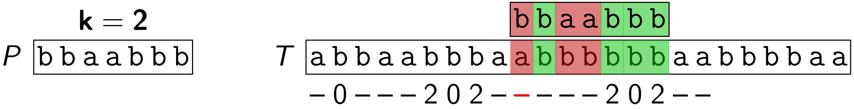
Given a pattern P of length m, a text T of length n, and a **threshold** k, for each position $r \leq n - m$, compute the **Hamming distance** $\mathrm{HD}(P, T[r \mathinner{.\,.} r+m))$ **if it does not exceed** k.



The k-mismatch problem

The k-mismatch problem

Given a pattern P of length m, a text T of length n, and a **threshold** k, for each position $r \leq n - m$, compute the **Hamming distance** $\mathrm{HD}(P, T[r \mathinner{.\,.} r+m))$ **if it does not exceed** k.



The k-mismatch problem

The k-mismatch problem

Given a pattern P of length m, a text T of length n, and a **threshold** k, for each position $r \leq n - m$, compute the **Hamming distance** $\mathrm{HD}(P, T[r\mathinner{.\,.} r+m))$ **if it does not exceed** k.

$\mathbf{k = 2}$

P `bbaabbb`

T `abbaabbbaabbbbbbaabbbbaa`

− 0 − − − 2 0 2 − − − − − − 2 0 2 − −

Algorithms:

	Fischer, Patterson Compl. Comput. 1974	$\widetilde{\mathcal{O}}(n\sigma)$ time
	⋮	⋮
	Gawrychowski, Uznański ICALP 2018	$\widetilde{\mathcal{O}}(n + nk/\sqrt{m})$ time

The k-mismatch problem

The k-mismatch problem

Given a pattern P of length m, a text T of length $n = \mathcal{O}(m)$, and a **threshold** k, for each $r \le n - m$, compute the **Hamming distance** $\mathrm{HD}(P, T[r \mathinner{.\,.} r+m))$ **if it does not exceed** k.

$\mathbf{k = 2}$

P `b b a a b b b`

T `a b b a a b b b a a b b b b b b a a b b b b a a`

− 0 − − − 2 0 2 − − − − − 2 0 2 − −

Algorithms:

Fischer, Patterson Compl. Comput. 1974	$\widetilde{\mathcal{O}}(n\sigma)$ time
⋮	⋮
Gawrychowski, Uznański ICALP 2018	$\widetilde{\mathcal{O}}(n + k\sqrt{n})$ time

The dynamic k-mismatch problem

The dynamic k-mismatch problem

Updates: Character substitutions in P and T.

Queries: Given $r \in [0 \mathinner{.\,.} n-m]$, compute $\mathrm{HD}(P, T[r \mathinner{.\,.} r+m))$ **if it does not exceed** k.

$\mathbf{k = 2}$

P `bbaabbb`

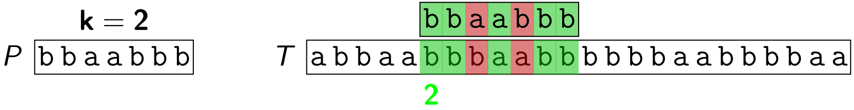
T `abbaabbbaabbbbbbaabbbbaa`

The dynamic k-mismatch problem

The dynamic k-mismatch problem

Updates: Character substitutions in P and T.

Queries: Given $r \in [0\,..\,n-m]$, compute $\mathrm{HD}(P, T[r\,..\,r+m))$ **if it does not exceed** k.



$\mathbf{k = 2}$

P b b a a b b b

b b a a b b b

T a b b a a b b b a a b b b b b b a a b b b b a a

2

The dynamic k-mismatch problem

The dynamic k-mismatch problem

Updates: Character substitutions in P and T.

Queries: Given $r \in [0\mathinner{.\,.} n-m]$, compute $\mathrm{HD}(P, T[r\mathinner{.\,.} r+m))$ **if it does not exceed** k.

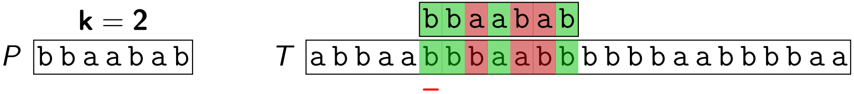
$\mathbf{k = 2}$

P `b b a a b a b` T `a b b a a b b b a a b b b b b b a a b b b b a a`

The dynamic k-mismatch problem

The dynamic k-mismatch problem

Updates: Character substitutions in P and T.
Queries: Given $r \in [0 \mathinner{.\,.} n-m]$, compute $\mathrm{HD}(P, T[r \mathinner{.\,.} r+m))$ **if it does not exceed** k.



$\mathbf{k = 2}$

P b b a a b a b

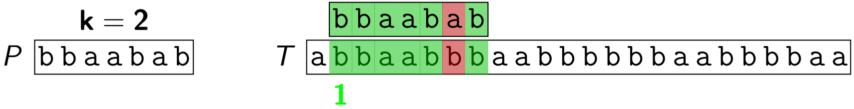
b b a a b a b

T a b b a a b b b a a b b b b b b a a b b b b a a

The dynamic k-mismatch problem

The dynamic k-mismatch problem

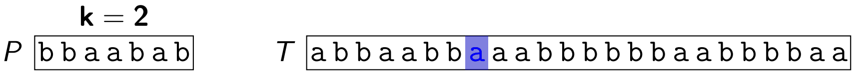
Updates: Character substitutions in P and T.
Queries: Given $r \in [0\mathinner{.\,.} n-m]$, compute $\mathrm{HD}(P, T[r\mathinner{.\,.} r+m))$ **if it does not exceed** k.



The dynamic *k*-mismatch problem

The dynamic *k*-mismatch problem

Updates: Character substitutions in P and T.
Queries: Given $r \in [0 \mathinner{.\,.} n-m]$, compute $\mathrm{HD}(P, T[r \mathinner{.\,.} r+m))$ **if it does not exceed** k.



$\mathbf{k = 2}$

P b b a a b a b

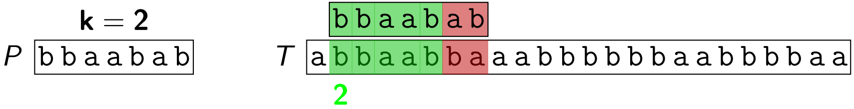
T a b b a a b b a a a b b b b b b a a b b b b a a

The dynamic k-mismatch problem

The dynamic k-mismatch problem

Updates: Character substitutions in P and T.

Queries: Given $r \in [0\,..\,n-m]$, compute $\mathrm{HD}(P, T[r\,..\,r+m))$ **if it does not exceed** k.



$\mathbf{k = 2}$

b b a a b a b

P b b a a b a b

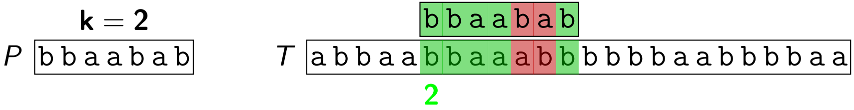
T a b b a a b b a a a b b b b b b a a b b b b a a

2

The dynamic k-mismatch problem

The dynamic k-mismatch problem

Updates: Character substitutions in P and T.
Queries: Given $r \in [0\,..\,n-m]$, compute $\mathrm{HD}(P, T[r\,..\,r+m))$ **if it does not exceed** k.



Results

Update	Query	Condition	Reference
$\widetilde{\mathcal{O}}(\sqrt{n}\cdot\sigma)$	$\widetilde{\mathcal{O}}(\sqrt{n}\cdot\sigma)$		Clifford, Grønlund, Larsen, Starikovskaya STACS 2018
$\widetilde{\mathcal{O}}(n^{3/4})$	$\widetilde{\mathcal{O}}(n^{3/4})$		
Lower bounds:			
$n^{1/2-\delta}$	$n^{1/2-\delta}$	$k = n$, OMv	Clifford, Grønlund, Larsen, Starikovskaya STACS 2018
$o(\frac{\log^{3/2} n}{\log^3 \log n})$	$o(\frac{\log^{3/2} n}{\log^3 \log n})$	$k = n$	

Results

Update	Query	Condition	Reference
$\widetilde{\mathcal{O}}(\sqrt{n} \cdot \sigma)$	$\widetilde{\mathcal{O}}(\sqrt{n} \cdot \sigma)$		Clifford, Grønlund, Larsen, Starikovskaya STACS 2018
$\widetilde{\mathcal{O}}(n^{3/4})$	$\widetilde{\mathcal{O}}(n^{3/4})$		
$\widetilde{\mathcal{O}}(1)$	$\widetilde{\mathcal{O}}(k)$		
$\widetilde{\mathcal{O}}(k)$	$\widetilde{\mathcal{O}}(1)$		This work
Lower bounds:			
$n^{1/2-\delta}$	$n^{1/2-\delta}$	$k = n$, OMv	Clifford, Grønlund, Larsen, Starikovskaya STACS 2018
$o(\frac{\log^{3/2} n}{\log^3 \log n})$	$o(\frac{\log^{3/2} n}{\log^3 \log n})$	$k = n$	

Results

Update	Query	Condition	Reference
$\widetilde{\mathcal{O}}(\sqrt{n} \cdot \sigma)$	$\widetilde{\mathcal{O}}(\sqrt{n} \cdot \sigma)$		Clifford, Grønlund, Larsen, Starikovskaya STACS 2018
$\widetilde{\mathcal{O}}(n^{3/4})$	$\widetilde{\mathcal{O}}(n^{3/4})$		
$\widetilde{\mathcal{O}}(1)$	$\widetilde{\mathcal{O}}(k)$		
$\widetilde{\mathcal{O}}(k)$	$\widetilde{\mathcal{O}}(1)$		This work
Lower bounds:			
$n^{1/2-\delta}$	$n^{1/2-\delta}$	$k = n$, OMv	Clifford, Grønlund, Larsen, Starikovskaya STACS 2018
$o(\frac{\log^{3/2} n}{\log^3 \log n})$	$o(\frac{\log^{3/2} n}{\log^3 \log n})$	$k = n$	
$\min(k, \sqrt{n})^{1-\delta}$	$\min(k, \sqrt{n})^{1-\delta}$	3SUM	This work

Results

Update	Query	Condition	Reference
$\widetilde{\mathcal{O}}(\sqrt{n} \cdot \sigma)$	$\widetilde{\mathcal{O}}(\sqrt{n} \cdot \sigma)$		Clifford, Grønlund, Larsen, Starikovskaya STACS 2018
$\widetilde{\mathcal{O}}(n^{3/4})$	$\widetilde{\mathcal{O}}(n^{3/4})$		
$\widetilde{\mathcal{O}}(1)$	$\widetilde{\mathcal{O}}(k)$		This work
$\widetilde{\mathcal{O}}(k)$	$\widetilde{\mathcal{O}}(1)$		
$\widetilde{\mathcal{O}}(x)$	$\widetilde{\mathcal{O}}(n/k + \sqrt{nk/x})$		
Lower bounds:			
$n^{1/2-\delta}$	$n^{1/2-\delta}$	$k = n$, OMv	Clifford, Grønlund, Larsen, Starikovskaya STACS 2018
$o(\frac{\log^{3/2} n}{\log^3 \log n})$	$o(\frac{\log^{3/2} n}{\log^3 \log n})$	$k = n$	
$\min(k, \sqrt{n})^{1-\delta}$	$\min(k, \sqrt{n})^{1-\delta}$	3SUM	This work

Results

Update	Query	Condition	Reference
$\widetilde{\mathcal{O}}(\sqrt{n} \cdot \sigma)$	$\widetilde{\mathcal{O}}(\sqrt{n} \cdot \sigma)$		Clifford, Grønlund, Larsen, Starikovskaya STACS 2018
$\widetilde{\mathcal{O}}(n^{3/4})$	$\widetilde{\mathcal{O}}(n^{3/4})$		
$\widetilde{\mathcal{O}}(1)$	$\widetilde{\mathcal{O}}(k)$		This work
$\widetilde{\mathcal{O}}(k)$	$\widetilde{\mathcal{O}}(1)$		
$\widetilde{\mathcal{O}}(\sqrt[3]{nk})$	$\widetilde{\mathcal{O}}(n/k + \sqrt[3]{nk})$		
Lower bounds:			
$n^{1/2-\delta}$	$n^{1/2-\delta}$	$k = n$, OMv	Clifford, Grønlund, Larsen, Starikovskaya STACS 2018
$o(\frac{\log^{3/2} n}{\log^3 \log n})$	$o(\frac{\log^{3/2} n}{\log^3 \log n})$	$k = n$	
$\min(k, \sqrt{n})^{1-\delta}$	$\min(k, \sqrt{n})^{1-\delta}$	3SUM	This work

Results

Update	Query	Condition	Reference
$\widetilde{\mathcal{O}}(\sqrt{n}\cdot\sigma)$	$\widetilde{\mathcal{O}}(\sqrt{n}\cdot\sigma)$		Clifford, Grønlund, Larsen, Starikovskaya STACS 2018
$\widetilde{\mathcal{O}}(n^{3/4})$	$\widetilde{\mathcal{O}}(n^{3/4})$		
$\widetilde{\mathcal{O}}(1)$	$\widetilde{\mathcal{O}}(k)$		
$\widetilde{\mathcal{O}}(k)$	$\widetilde{\mathcal{O}}(1)$		This work
$\widetilde{\mathcal{O}}(n^{2/3})$	$\widetilde{\mathcal{O}}(n^{2/3})$		
Lower bounds:			
$n^{1/2-\delta}$	$n^{1/2-\delta}$	$k = n$, OMv	Clifford, Grønlund, Larsen, Starikovskaya STACS 2018
$o(\frac{\log^{3/2} n}{\log^3 \log n})$	$o(\frac{\log^{3/2} n}{\log^3 \log n})$	$k = n$	
$\min(k,\sqrt{n})^{1-\delta}$	$\min(k,\sqrt{n})^{1-\delta}$	3SUM	This work

$\widetilde{\mathcal{O}}(1)$-time updates, $\widetilde{\mathcal{O}}(k)$-time queries

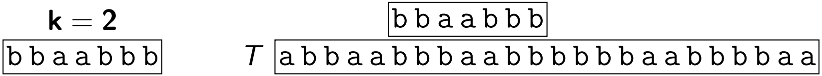
k = 2

`bbaabbb`

T `abbaabbbaabbbbbbaabbbbaa`

1. Maintain $P \cdot T$ in a dynamic data structure for **Longest Common Extension** queries.
   - $\mathcal{O}(\log n)$-time (w.h.p.) or $\mathcal{O}(\log^{1+o(1)} n)$-time (deterministic) updates and queries.

$\widetilde{\mathcal{O}}(1)$-time updates, $\widetilde{\mathcal{O}}(k)$-time queries



k = 2

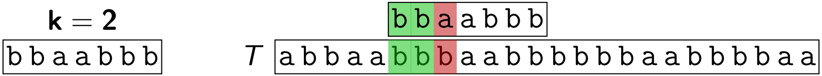
b b a a b b b

T a b b a a b b b a a b b b b b b a a b b b b a a

b b a a b b b

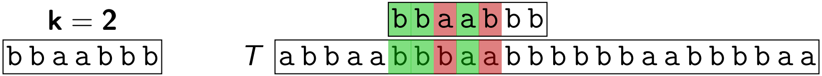
1. Maintain $P \cdot T$ in a dynamic data structure for **Longest Common Extension** queries.
    - $\mathcal{O}(\log n)$-time (w.h.p.) or $\mathcal{O}(\log^{1+o(1)} n)$-time (deterministic) updates and queries.
2. For each query, use LCE queries to locate the first $k + 1$ mismatches.

$\widetilde{\mathcal{O}}(1)$-time updates, $\widetilde{\mathcal{O}}(k)$-time queries



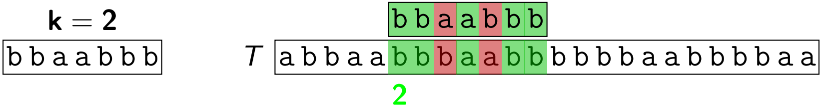
1. Maintain $P \cdot T$ in a dynamic data structure for **Longest Common Extension** queries.
   - $\mathcal{O}(\log n)$-time (w.h.p.) or $\mathcal{O}(\log^{1+o(1)} n)$-time (deterministic) updates and queries.
2. For each query, use LCE queries to locate the first $k+1$ mismatches.

$\widetilde{\mathcal{O}}(1)$-time updates, $\widetilde{\mathcal{O}}(k)$-time queries



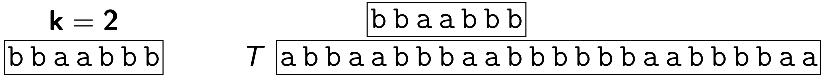
1. Maintain $P \cdot T$ in a dynamic data structure for **Longest Common Extension** queries.
    - $\mathcal{O}(\log n)$-time (w.h.p.) or $\mathcal{O}(\log^{1+o(1)} n)$-time (deterministic) updates and queries.
2. For each query, use LCE queries to locate the first $k+1$ mismatches.

$\widetilde{\mathcal{O}}(1)$-time updates, $\widetilde{\mathcal{O}}(k)$-time queries



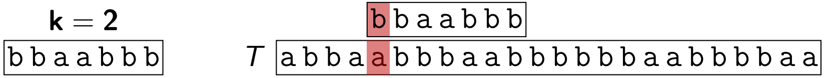
1. Maintain $P \cdot T$ in a dynamic data structure for **Longest Common Extension** queries.
   - $\mathcal{O}(\log n)$-time (w.h.p.) or $\mathcal{O}(\log^{1+o(1)} n)$-time (deterministic) updates and queries.
2. For each query, use LCE queries to locate the first $k+1$ mismatches.

$\widetilde{\mathcal{O}}(1)$-time updates, $\widetilde{\mathcal{O}}(k)$-time queries



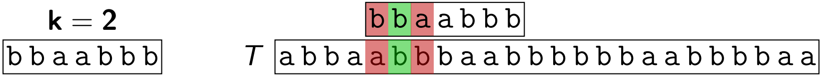
1. Maintain $P \cdot T$ in a dynamic data structure for **Longest Common Extension** queries.
    - $\mathcal{O}(\log n)$-time (w.h.p.) or $\mathcal{O}(\log^{1+o(1)} n)$-time (deterministic) updates and queries.
2. For each query, use LCE queries to locate the first $k+1$ mismatches.

$\widetilde{\mathcal{O}}(1)$-time updates, $\widetilde{\mathcal{O}}(k)$-time queries



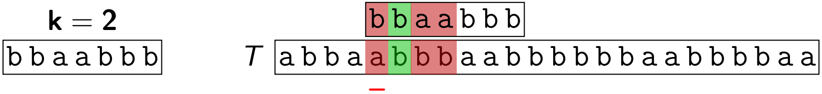
1. Maintain $P \cdot T$ in a dynamic data structure for **Longest Common Extension** queries.
    - $\mathcal{O}(\log n)$-time (w.h.p.) or $\mathcal{O}(\log^{1+o(1)} n)$-time (deterministic) updates and queries.
2. For each query, use LCE queries to locate the first $k+1$ mismatches.

$\widetilde{\mathcal{O}}(1)$-time updates, $\widetilde{\mathcal{O}}(k)$-time queries



1. Maintain $P \cdot T$ in a dynamic data structure for **Longest Common Extension** queries.
    - $\mathcal{O}(\log n)$-time (w.h.p.) or $\mathcal{O}(\log^{1+o(1)} n)$-time (deterministic) updates and queries.
2. For each query, use LCE queries to locate the first $k+1$ mismatches.

$\widetilde{\mathcal{O}}(1)$-time updates, $\widetilde{\mathcal{O}}(k)$-time queries



1. Maintain $P \cdot T$ in a dynamic data structure for **Longest Common Extension** queries.
   - $\mathcal{O}(\log n)$-time (w.h.p.) or $\mathcal{O}(\log^{1+o(1)} n)$-time (deterministic) updates and queries.
2. For each query, use LCE queries to locate the first $k+1$ mismatches.

The k-mismatch queries

Theorem (Charalampopoulos, K, Wellnitz (FOCS'20))

All k-mismatch occurrences of P in T, represented as $\mathcal{O}(k^2)$ arithmetic progressions of common difference (augmented with a Hamming distance), can be computed in $\widetilde{\mathcal{O}}(k^2)$ time.

$\mathbf{k = 2}$

P b b a a b b b

T a b b a a b b b a a b b b b b b a a b b b b a a

– 0 – – – 2 0 2 – – – – – 2 0 2 – –

The k-mismatch queries

Theorem (Charalampopoulos, K, Wellnitz (FOCS'20))

All k-mismatch occurrences of P in T, represented as $\mathcal{O}(k^2)$ arithmetic progressions of common difference (augmented with a Hamming distance), can be computed in $\widetilde{\mathcal{O}}(k^2)$ time.

$\mathbf{k = 2}$

P `bbaabbb`

T `abbaabbbaabbbbbbaabbbbaa`

– 0 – – – 2 0 2 – – – – – – 2 0 2 – –

Naive application:

1. At update, recompute all the k-mismatch occurrences: $\widetilde{\mathcal{O}}(k^2)$ time.
2. At query, find the arithmetic progression containing the given position (if any): $\widetilde{\mathcal{O}}(1)$ time.

Structural characterization of k-mismatch occurrences

Theorem (Charalampopoulos, K, Wellnitz (FOCS'20))

At least one of the following holds:

1. *The number of k-mismatch occurrences of P in T is $\mathcal{O}(k)$.*
2. *The pattern P is almost periodic:* $\mathrm{HD}(P, Q^{\infty}[0\mathinner{.\,.} m)) \leq 2k$ *for some Q of length $\mathcal{O}(\frac{m}{k})$.*

Structural characterization of k-mismatch occurrences

Theorem (Charalampopoulos, K, Wellnitz (FOCS'20))

At least one of the following holds:

1. *The number of k-mismatch occurrences of P in T is $\mathcal{O}(k)$.*
2. *The pattern P is almost periodic:* $\mathrm{HD}(P, Q^{\infty}[0\mathinner{.\,.} m)) \le 2k$ *for some Q of length $\mathcal{O}(\frac{m}{k})$.*

Let us ignore the second case. How to improve update time from $\widetilde{\mathcal{O}}(k^2)$ to $\widetilde{\mathcal{O}}(k)$?

Structural characterization of k-mismatch occurrences

Theorem (Charalampopoulos, K, Wellnitz (FOCS'20))

At least one of the following holds:

1. *The number of k-mismatch occurrences of P in T is $\mathcal{O}(k)$.*
2. *The pattern P is almost periodic:* $\mathrm{HD}(P, Q^{\infty}[0\mathinner{.\,.} m)) \le 2k$ *for some Q of length $\mathcal{O}(\frac{m}{k})$.*

Let us ignore the second case. How to improve update time from $\widetilde{\mathcal{O}}(k^2)$ to $\widetilde{\mathcal{O}}(k)$?

1. Every k updates, compute the set O of all $2k$-mismatch occurrences of P in T.

Structural characterization of k-mismatch occurrences

Theorem (Charalampopoulos, K, Wellnitz (FOCS'20))

At least one of the following holds:

1. *The number of k-mismatch occurrences of P in T is $\mathcal{O}(k)$.*
2. *The pattern P is almost periodic:* $\mathrm{HD}(P, Q^\infty[0\mathinner{.\,.} m)) \le 2k$ *for some Q of length $\mathcal{O}(\frac{m}{k})$.*

Let us ignore the second case. How to improve update time from $\widetilde{\mathcal{O}}(k^2)$ to $\widetilde{\mathcal{O}}(k)$?

1. Every k updates, compute the set O of all $2k$-mismatch occurrences of P in T.
2. At subsequent updates, maintain $\mathrm{HD}(P, T[i\mathinner{.\,.} i+m))$ for all $i \in O$.

Structural characterization of k-mismatch occurrences

Theorem (Charalampopoulos, K, Wellnitz (FOCS'20))

At least one of the following holds:

1. *The number of k-mismatch occurrences of P in T is $\mathcal{O}(k)$.*
2. *The pattern P is almost periodic:* $\mathrm{HD}(P, Q^{\infty}[0\mathinner{.\,.} m)) \leq 2k$ *for some Q of length $\mathcal{O}(\frac{m}{k})$.*

Let us ignore the second case. How to improve update time from $\widetilde{\mathcal{O}}(k^2)$ to $\widetilde{\mathcal{O}}(k)$?

1. Every k updates, compute the set O of all $2k$-mismatch occurrences of P in T.
2. At subsequent updates, maintain $\mathrm{HD}(P, T[i\mathinner{.\,.} i+m))$ for all $i \in O$.
3. At queries, return the stored value for $i \in O$ and $> k$ for $i \notin O$.

Structural characterization of k-mismatch occurrences

Theorem (Charalampopoulos, K, Wellnitz (FOCS'20))

At least one of the following holds:

1. *The number of k-mismatch occurrences of P in T is $\mathcal{O}(k)$.*
2. *The pattern P is almost periodic:* $\mathrm{HD}(P, Q^{\infty}[0\,..\,m)) \le 2k$ *for some Q of length $\mathcal{O}(\frac{m}{k})$.*

Let us ignore the second case. How to improve update time from $\widetilde{\mathcal{O}}(k^2)$ to $\widetilde{\mathcal{O}}(k)$?

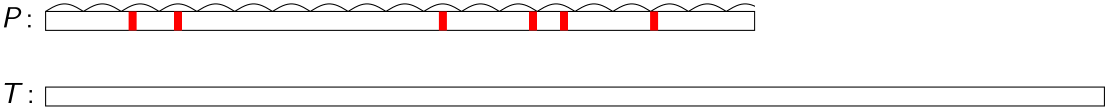
1. Every k updates, compute the set O of all $2k$-mismatch occurrences of P in T.
2. At subsequent updates, maintain $\mathrm{HD}(P, T[i\,..\,i+m))$ for all $i \in O$.
3. At queries, return the stored value for $i \in O$ and $> k$ for $i \notin O$.

Query time: $\mathcal{O}(1)$
Update time: $\widetilde{\mathcal{O}}(k^2/k + |O|) = \widetilde{\mathcal{O}}(k)$ amortized.

Structural characterization: Almost periodic pattern

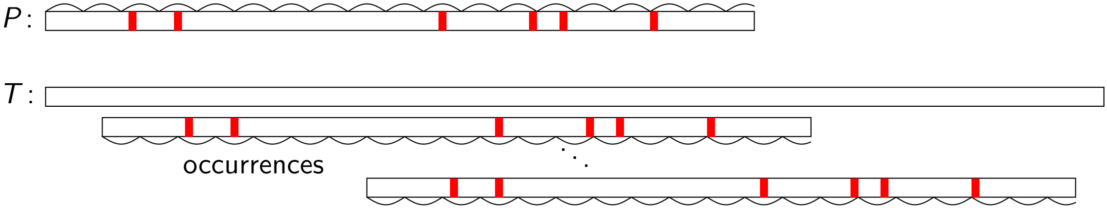
1. Recall: $\mathrm{HD}(P, Q^\infty[0\,.\,.\,m)) \le 2k$ for $|Q| = \mathcal{O}(m/k)$.

Structural characterization: Almost periodic pattern



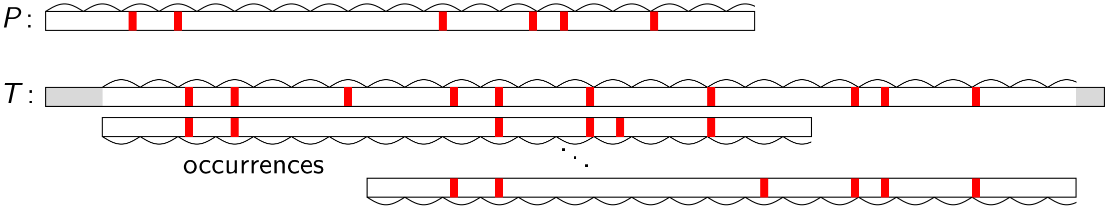
1. Recall: $\mathrm{HD}(P, Q^{\infty}[0\,..\,m)) \leq 2k$ for $|Q| = \mathcal{O}(m/k)$.
2. Suppose that $n \leq 1.5m$ (otherwise, partition T into $\mathcal{O}(1)$ pieces).

Structural characterization: Almost periodic pattern



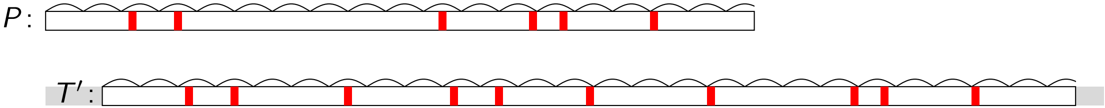
1. Recall: $\mathrm{HD}(P, Q^\infty[0\mathinner{.\,.} m)) \le 2k$ for $|Q| = \mathcal{O}(m/k)$.
2. Suppose that $n \le 1.5m$ (otherwise, partition T into $\mathcal{O}(1)$ pieces).
3. Consider the leftmost and rightmost k-mismatch occurrences of P in T.

Structural characterization: Almost periodic pattern



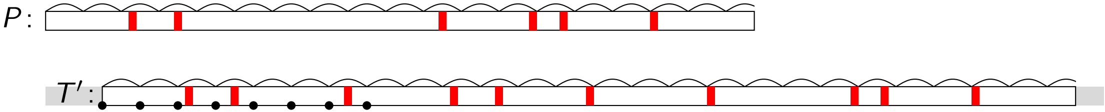
1. Recall: $\mathrm{HD}(P, Q^{\infty}[0\mathinner{.\,.} m)) \le 2k$ for $|Q| = \mathcal{O}(m/k)$.
2. Suppose that $n \le 1.5m$ (otherwise, partition T into $\mathcal{O}(1)$ pieces).
3. Consider the leftmost and rightmost k-mismatch occurrences of P in T.

Structural characterization: Almost periodic pattern



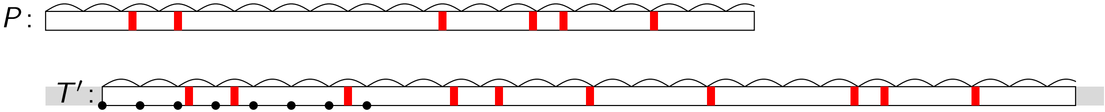
1. Recall: $\mathrm{HD}(P, Q^\infty[0\,.\,.\,m)) \le 2k$ for $|Q| = \mathcal{O}(m/k)$.
2. Suppose that $n \le 1.5m$ (otherwise, partition T into $\mathcal{O}(1)$ pieces).
3. Consider the leftmost and rightmost k-mismatch occurrences of P in T.
4. $\mathrm{HD}(T', Q^\infty[0\,.\,.\,|T'|)) \le 6k$.

Structural characterization: Almost periodic pattern



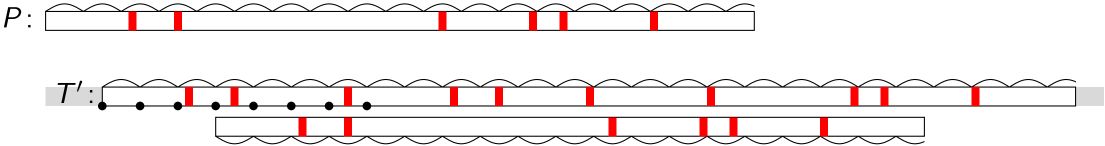
1. Recall: $\mathrm{HD}(P, Q^{\infty}[0\mathinner{.\,.} m)) \le 2k$ for $|Q| = \mathcal{O}(m/k)$.
2. Suppose that $n \le 1.5m$ (otherwise, partition T into $\mathcal{O}(1)$ pieces).
3. Consider the leftmost and rightmost k-mismatch occurrences of P in T.
4. $\mathrm{HD}(T', Q^{\infty}[0\mathinner{.\,.}|T'|)) \le 6k$.
5. $O \subseteq \{\min O, \min O + |Q|, \ldots, \max O\}$.

Structural characterization: Almost periodic pattern



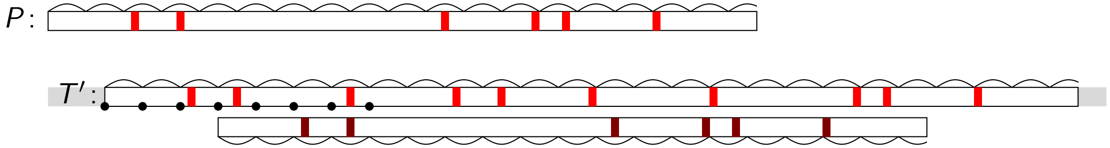
Computing $\mathrm{HD}(P, T[i\,..\,i+m))$ for $i \in O$:

Structural characterization: Almost periodic pattern



Computing $\mathrm{HD}(P, T[i\,..\,i+m))$ for $i \in O$:

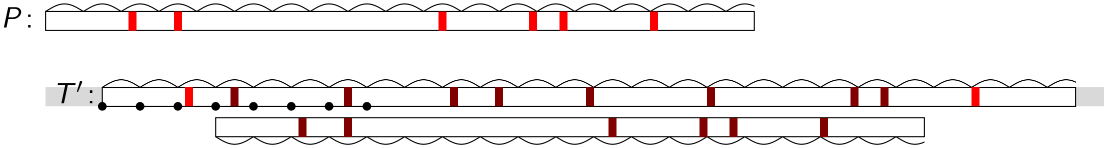
Structural characterization: Almost periodic pattern



Computing $\mathrm{HD}(P, T[i\,..\,i+m))$ for $i \in O$:

- $+$ $\mathrm{HD}(P, Q^{\infty}[0\,..\,m))$

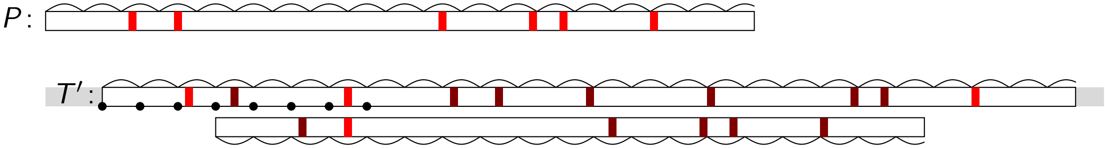
Structural characterization: Almost periodic pattern



Computing $\mathrm{HD}(P, T[i\mathinner{.\,.} i+m))$ for $i \in O$:

- $+$ $\mathrm{HD}(P, Q^{\infty}[0\mathinner{.\,.} m))$
- $+$ $\mathrm{HD}(T[i\mathinner{.\,.} i+m), Q^{\infty}[0\mathinner{.\,.} m))$

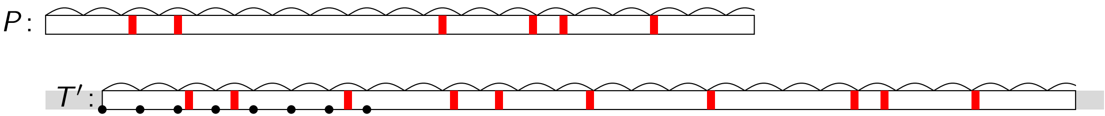
Structural characterization: Almost periodic pattern



Computing $\mathrm{HD}(P, T[i \mathinner{.\,.} i+m))$ for $i \in O$:

- $+$ $\mathrm{HD}(P, Q^\infty[0 \mathinner{.\,.} m))$
- $+$ $\mathrm{HD}(T[i \mathinner{.\,.} i+m), Q^\infty[0 \mathinner{.\,.} m))$
- $-$ Correcting term for aligned mismatches (-1 or -2 per aligned pair)

Data structure for the almost periodic case

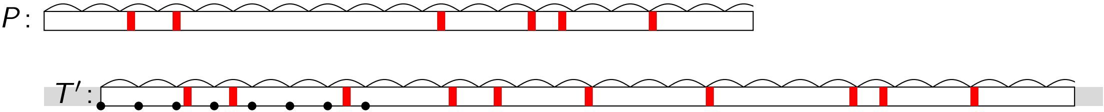


Recall that the structure is computed every k steps for threshold $2k$.

Components:

Needed at query time:

Data structure for the almost periodic case



Recall that the structure is computed every k steps for threshold $2k$.

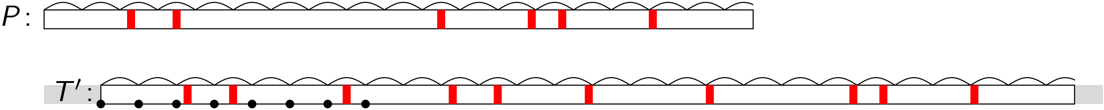
Components:

1. Mismatches between P and $Q^\infty[0\,.\,.\,m)$.
   - Size $\mathcal{O}(k)$; update time $\mathcal{O}(1)$.

Needed at query time:

$+$ $\mathrm{HD}(P, Q^\infty[0\,.\,.\,m))$;

Data structure for the almost periodic case



Recall that the structure is computed every k steps for threshold $2k$.

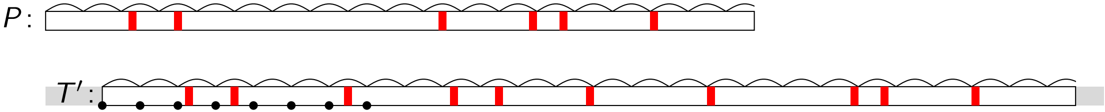
Components:

1. Mismatches between P and $Q^{\infty}[0\,..\,m)$.
    - Size $\mathcal{O}(k)$; update time $\mathcal{O}(1)$.
    - Total count needed.

Needed at query time:

$+$ $\mathrm{HD}(P, Q^{\infty}[0\,..\,m))$;

Data structure for the almost periodic case



Recall that the structure is computed every k steps for threshold $2k$.

Components:

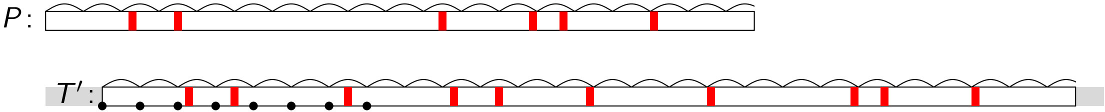
1. Mismatches between P and $Q^\infty[0\,..\,m)$.
   - Size $\mathcal{O}(k)$; update time $\mathcal{O}(1)$.
   - Total count needed.
2. Mismatches between T' and $Q^\infty[0\,..\,|T'|)$.
   - Size $\mathcal{O}(k)$; update time $\widetilde{\mathcal{O}}(1)$

Needed at query time:

$+$ $\mathrm{HD}(P, Q^\infty[0\,..\,m))$;

$+$ $\mathrm{HD}(T[i\,..\,i+m), Q^\infty[0\,..\,m))$;

Data structure for the almost periodic case



Recall that the structure is computed every k steps for threshold $2k$.

Components:

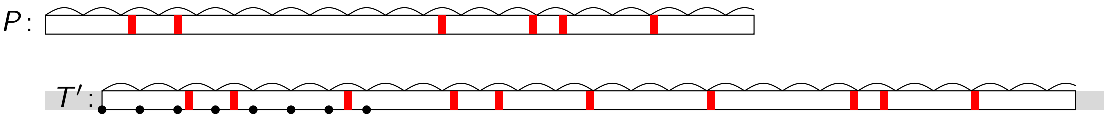
1. Mismatches between P and $Q^\infty[0\,..\,m)$.
   - Size $\mathcal{O}(k)$; update time $\mathcal{O}(1)$.
   - Total count needed.
2. Mismatches between T' and $Q^\infty[0\,..\,|T'|)$.
   - Size $\mathcal{O}(k)$; update time $\widetilde{\mathcal{O}}(1)$
   - Rank functionality needed.

Needed at query time:

$+$ $\mathrm{HD}(P, Q^\infty[0\,..\,m))$;

$+$ $\mathrm{HD}(T[i\,..\,i+m), Q^\infty[0\,..\,m))$;

Data structure for the almost periodic case



Recall that the structure is computed every k steps for threshold $2k$.

Components:

1. Mismatches between P and $Q^\infty[0\,..\,m)$.
   - Size $\mathcal{O}(k)$; update time $\mathcal{O}(1)$.
   - Total count needed.
2. Mismatches between T' and $Q^\infty[0\,..\,|T'|)$.
   - Size $\mathcal{O}(k)$; update time $\widetilde{\mathcal{O}}(1)$
   - Rank functionality needed.
3. All non-zero correcting terms.
   - Size $\mathcal{O}(k^2)$; update time $\widetilde{\mathcal{O}}(k)$.

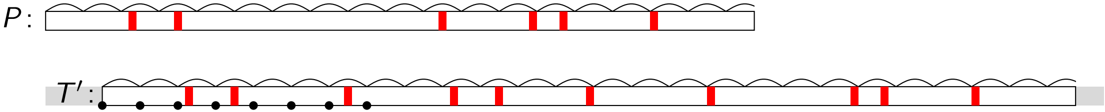
Needed at query time:

$+$ $\mathrm{HD}(P, Q^\infty[0\,..\,m))$;

$+$ $\mathrm{HD}(T[i\,..\,i+m), Q^\infty[0\,..\,m))$;

$-$ Correcting term for i.

Data structure for the almost periodic case



Recall that the structure is computed every k steps for threshold $2k$.

Components:

1. Mismatches between P and $Q^\infty[0\,..\,m)$.
   - Size $\mathcal{O}(k)$; update time $\mathcal{O}(1)$.
   - Total count needed.
2. Mismatches between T' and $Q^\infty[0\,..\,|T'|)$.
   - Size $\mathcal{O}(k)$; update time $\widetilde{\mathcal{O}}(1)$
   - Rank functionality needed.
3. All non-zero correcting terms.
   - Size $\mathcal{O}(k^2)$; update time $\widetilde{\mathcal{O}}(k)$.
   - Random access needed.

Needed at query time:

$+$ $\mathrm{HD}(P, Q^\infty[0\,..\,m))$;

$+$ $\mathrm{HD}(T[i\,..\,i+m), Q^\infty[0\,..\,m))$;

$-$ Correcting term for i.

$\widetilde{\mathcal{O}}(1)$-time updates, $\widetilde{\mathcal{O}}(k)$-time queries: Summary

Algorithm structure:

1. Every k steps **analyze** the pattern with respect to the threshold $2k$.

$\widetilde{\mathcal{O}}(1)$-time updates, $\widetilde{\mathcal{O}}(k)$-time queries: Summary

Algorithm structure:

1. Every k steps **analyze** the pattern with respect to the threshold $2k$.
2. If the pattern is far from periodic:
    - Identify the set O of $2k$-mismatch occurrences of P in T.
    - Throughout the next k steps, maintain $\mathrm{HD}(P, T[i\mathinner{.\,.} i+m))$ for $i \in O$.

$\widetilde{\mathcal{O}}(1)$-time updates, $\widetilde{\mathcal{O}}(k)$-time queries: Summary

Algorithm structure:

1. Every k steps **analyze** the pattern with respect to the threshold $2k$.
2. If the pattern is far from periodic:
   - Identify the set O of $2k$-mismatch occurrences of P in T.
   - Throughout the next k steps, maintain $\mathrm{HD}(P, T[i \mathinner{.\,.} i+m))$ for $i \in O$.
3. Otherwise: $\mathrm{HD}(P, Q^{\infty}[0 \mathinner{.\,.} m)) \le 4k$.
   - Identify a fragment T' with $\mathrm{HD}(T', Q^{\infty}[0 \mathinner{.\,.} |T'|)) \le 12k$ containing all $2k$-mismatch occs.
   - Throughout the next k steps, maintain the mismatches with Q^{∞} and the correcting terms.

Tight lower bound for $k \leq \sqrt{n}$

3SUM Problem

Given sets $A, B, C \subseteq [-n \mathinner{.\,.} n)$ of total size k, decide if there exist $a \in A$, $b \in B$, and $c \in C$ with $a + b + c = 0$.

Conjecture: 3SUM cannot be solved in $\mathcal{O}(k^{2-\delta})$ time (for sufficiently large n; $n \geq k^3$ suffices).

Tight lower bound for $k \leq \sqrt{n}$

3SUM$^+$ Problem

Given sets $A, B, C \subseteq [-n\,..\,n)$ of total size k, for each $c \in C$, decide if there exist $a \in A$ and $b \in B$ with $a + b + c = 0$.

Conjecture: 3SUM cannot be solved in $\mathcal{O}(k^{2-\delta})$ time (for sufficiently large n; $n \geq k^3$ suffices).
Conditioned on the 3SUM conjecture, 3SUM$^+$ cannot be solved in $\mathcal{O}(k^{2-\delta})$ time for $n \geq k^{2+\epsilon}$.

Tight lower bound for $k \leq \sqrt{n}$

3SUM$^+$ Problem

Given sets $A, B, C \subseteq [-n\,..\,n)$ of total size k, for each $c \in C$, decide if there exist $a \in A$ and $b \in B$ with $a + b + c = 0$.

Conjecture: 3SUM cannot be solved in $\mathcal{O}(k^{2-\delta})$ time (for sufficiently large n; $n \geq k^3$ suffices).
Conditioned on the 3SUM conjecture, 3SUM$^+$ cannot be solved in $\mathcal{O}(k^{2-\delta})$ time for $n \geq k^{2+\epsilon}$.

Reduction:

- Initialize $P = 0^{|P|}$, $T = 0^{|T|}$ for $|P| \geq 4n, |T| \geq 5n$.
- Set $P[a + 2n] := 1$ for all $a \in A$.
- Set $T[3n - b] := 1$ for all $b \in B$.
- Report all $c \in C$ with $\mathrm{HD}(P, T[c + n\,..\,c + n + |P|)) < |A| + |B|$.

Conclusions and open problems

Main result: Tight bounds for the dynamic k-mismatch problem for $k \leq \sqrt{n}$.

- $\widetilde{\mathcal{O}}(1)$-time updates, $\widetilde{\mathcal{O}}(k)$-time queries;
- $\widetilde{\mathcal{O}}(k)$-time updates, $\widetilde{\mathcal{O}}(1)$-time queries.

Conclusions and open problems

Main result: Tight bounds for the dynamic k-mismatch problem for $k \leq \sqrt{n}$.

- $\widetilde{\mathcal{O}}(1)$-time updates, $\widetilde{\mathcal{O}}(k)$-time queries;
- $\widetilde{\mathcal{O}}(k)$-time updates, $\widetilde{\mathcal{O}}(1)$-time queries.

Further result: $\widetilde{\mathcal{O}}(x)$-time queries and $\widetilde{\mathcal{O}}(\frac{n}{k} + \sqrt{\frac{nk}{x}})$-time updates for any $x \in [1\mathinner{.\,.} k]$.

Conclusions and open problems

Main result: Tight bounds for the dynamic k-mismatch problem for $k \le \sqrt{n}$.

- $\widetilde{\mathcal{O}}(1)$-time updates, $\widetilde{\mathcal{O}}(k)$-time queries;
- $\widetilde{\mathcal{O}}(k)$-time updates, $\widetilde{\mathcal{O}}(1)$-time queries.

Further result: $\widetilde{\mathcal{O}}(x)$-time queries and $\widetilde{\mathcal{O}}(\frac{n}{k} + \sqrt{\frac{nk}{x}})$-time updates for any $x \in [1\mathinner{.\,.} k]$.

Open problems:

1. Conditional lower bounds for $k \ge \sqrt{n}$.
2. Dynamic pattern matching with k edits.

Conclusions and open problems

Main result: Tight bounds for the dynamic k-mismatch problem for $k \leq \sqrt{n}$.

- $\widetilde{\mathcal{O}}(1)$-time updates, $\widetilde{\mathcal{O}}(k)$-time queries;
- $\widetilde{\mathcal{O}}(k)$-time updates, $\widetilde{\mathcal{O}}(1)$-time queries.

Further result: $\widetilde{\mathcal{O}}(x)$-time queries and $\widetilde{\mathcal{O}}(\frac{n}{k} + \sqrt{\frac{nk}{x}})$-time updates for any $x \in [1\,..\,k]$.

Open problems:

1. Conditional lower bounds for $k \geq \sqrt{n}$.
2. Dynamic pattern matching with k edits.

Thank you for your attention!