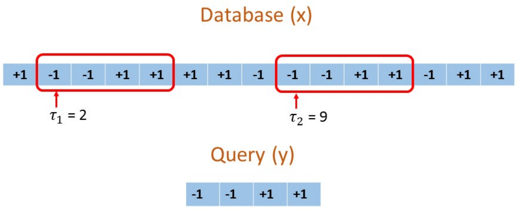
Sub-string/Pattern Matching in Sub-linear Time Using a Sparse Fourier Transform Approach

Krishna R. Narayanan
Joint work with Nagaraj T. Janakiraman, Avinash Vem & J.F. Chamberand

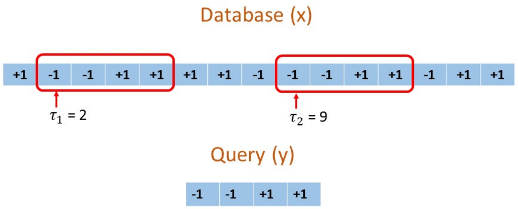
Department of Electrical and Computer Engineering
Texas A&M University

Problem Statement



- Database/String: $\underline{x} = [x[0], x[1], \cdots, x[N-1]]$ (length N)
- Query/Substring: $\underline{y} = [y[0], y[1], \cdots, y[M-1]]$ (length $M = N^{\mu}$)
- Signal Model: $x[i]$'s are i.i.d r.v. from $\mathcal{A} = \{+1, -1\}$ (extensions possible)

Problem Statement

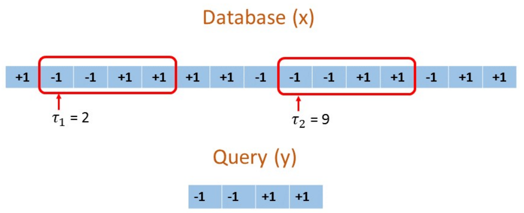


- Database/String: $\underline{x} = [x[0], x[1], \cdots, x[N-1]]$ (length N)
- Query/Substring: $\underline{y} = [y[0], y[1], \cdots, y[M-1]]$ (length $M = N^{\mu}$)
- Signal Model: $x[i]$'s are i.i.d r.v. from $\mathcal{A} = \{+1, -1\}$ (extensions possible)

Determine all the L locations $\underline{\tau} = [\tau_1, \tau_2, \cdots \tau_L]$ with high probability where

1. Exact Matching: $\underline{y}$ appears exactly in $\underline{x}$
   - $\underline{y} := \underline{x}[\tau : \tau + M - 1]$

Problem Statement

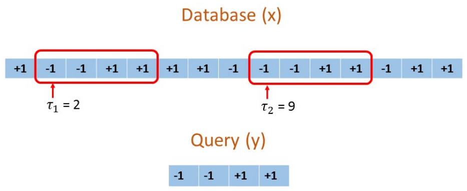


- Database/String: $\underline{x} = [x[0], x[1], \cdots, x[N-1]]$ (length N)
- Query/Substring: $\underline{y} = [y[0], y[1], \cdots, y[M-1]]$ (length $M = N^{\mu}$)
- Signal Model: $x[i]$'s are i.i.d r.v. from $\mathcal{A} = \{+1, -1\}$ (extensions possible)

Determine all the L locations $\underline{\tau} = [\tau_1, \tau_2, \cdots \tau_L]$ with high probability where

1. Exact Matching: $\underline{y}$ appears exactly in $\underline{x}$
   - $\underline{y} := \underline{x}[\tau : \tau + M - 1]$
2. Approximate Matching: $\underline{y}$ is a noisy substring of $\underline{x}$
   - $\underline{y} := \underline{x}[\tau : \tau + M - 1] \odot \underline{b}$
   - $\underline{b}$ is a noise sequence with $d_H(\underline{y}, \underline{x}[\tau : \tau + M - 1]) \leq K$

Notation



Symbol	*Meaning*
N	Size of the string or database in symbols
$M = N^{\mu}$	Length of the query in symbols
$L = N^{\lambda}$	Number of matches
K	$\max_{\tau} d_H(\underline{x}[\tau : \tau + M - 1], \underline{y})$
η	$\frac{K}{M}$

Probabilistic recovery

$\mathbb{P}(\hat{\underline{\tau}} \neq \underline{\tau}) \to 0$ as $N \to \infty$

Main Result

Symbol	*Meaning*
N	Size of the string or database in symbols
$M = N^{\mu}$	Length of the query in symbols
$L = N^{\lambda}$	Number of matches
K	$\max_{\tau} d_H(\underline{x}[\tau : \tau + M - 1], \underline{y})$
η	$\frac{K}{M}$

Theorem 1

Assume that a sketch of $\underline{x}$ can be precomputed and stored. Then for the exact pattern matching and approximate pattern matching (with $K = \eta M,\ 0 \leq \eta \leq 1/6$) problems, our algorithm has

- *Sketching complexity:* $O(\frac{N}{M} \log N) = O(N^{1-\mu} \log N)$ *samples*
- *Computational complexity:* $O(\max(N^{1-\mu} \log^2 N, N^{\mu+\lambda} \log N))$
- *a decoder for which* $\mathbb{P}(\hat{\mathcal{T}} \neq \mathcal{T}) \to 0$ *as* $N \to \infty$

Main Result

Symbol	*Meaning*
N	Size of the string or database in symbols
$M = N^{\mu}$	Length of the query in symbols
$L = N^{\lambda}$	Number of matches
K	$\max_{\tau} d_H(\underline{x}[\tau : \tau + M - 1], \underline{y})$
η	$\frac{K}{M}$

Theorem 1

Assume that a sketch of $\underline{x}$ can be precomputed and stored. Then for the exact pattern matching and approximate pattern matching (with $K = \eta M,\ 0 \leq \eta \leq 1/6$) problems, our algorithm has

- *Sketching complexity:* $O(\frac{N}{M} \log N) = O(N^{1-\mu} \log N)$ *samples*
- *Computational complexity:* $O(\max(N^{1-\mu} \log^2 N, N^{\mu+\lambda} \log N))$
- *a decoder for which* $\mathbb{P}(\hat{\mathcal{T}} \neq \mathcal{T}) \to 0$ *as* $N \to \infty$

Note

Particularly when $L = O(1)$ or $L < \frac{N}{M}$ (i.e. $\lambda < 1 - \mu$) our algorithm has a sub-linear time and space complexity.

Some Prior Work

Exact Matching

- **Boyer and Moore 1977**: First occurrence of the match (only τ_1)
  - Average complexity - $O(N^{1-\mu} \log N)$ (sublinear)
  - Worst case complexity - $O(N \log N)$
- **Goodrich, Atallah, and Tamassia 2005**: BWT, suffix-arrays based indexing
  - Time complexity - $O(M + L)$ (sublinear)
  - Storage Complexity - $O(N\ H_k(X) \log^\epsilon N) + o(N)$ bits (linear)
  - Read alignment in Bio-informatics community[**li2009fast**; **li2010fast**]

Approximate Matching

- **Chang and Marr 1994**: Generalization of Boyer and Moore 1977
  - Average time complexity - $O(NK/M \log N)$ (sub-linear only when $K \ll M$)
- **zhang2003approximate**: Approximate Matching using BWT
  - Worst case time complexity: $O(\min\{M(M-K)|\mathcal{A}|^k \log \frac{N}{|\mathcal{A}|}, NM \log \frac{N}{|\mathcal{A}|}\})$
  - Complexity grows with $|\mathcal{A}|$ and K
- **Andoni et al. 2013**: $O(N/M^{0.359})$ (sub-linear even when $K = O(M)$)
  - Combinatorial in nature

Some Prior Work

Sparse Fourier Transform Approach

- Pawar and Ramchandran 2014: Robust Sparse Fourier Transform
  - Sparse graph code approach
  - Computational complexity : $O(N \log N)$
- Hassanieh et al. 2012: Faster GPS receiver
  - Exploited sparsity in Correlation function R_{XY}

Motivation

- **Cross-correlation** ($\underline{r}$):

$$r[m] = (\underline{x} * \underline{y})[m] \triangleq \sum_{i=0}^{M-1} x[m+i]y[i], \quad 0 \leq m \leq N-1$$

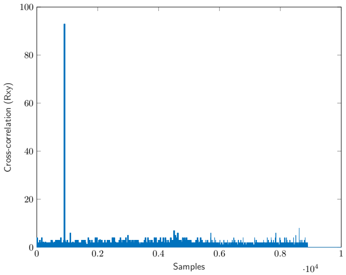
- **Naive implementation**: $O(MN) = O(N^{1+\mu})$ (super-linear complexity)
- **Fourier Transform Approach**: $O(N \log N)$ complexity

$$\underline{r} = \mathcal{F}_N^{-1}\{\ \mathcal{F}_N\{\underline{x}\}\ \odot\ \mathcal{F}_N\{\underline{y}'\}\ \},\quad \underline{y}' = \underline{y}^*[-n]$$

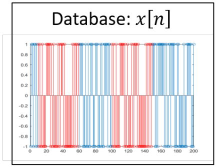
Key Observation

- $\underline{r}$ is Sparse with some noise.

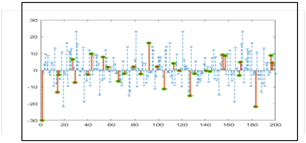
$$r[m] \ = \begin{cases} M, & \text{if } m \in \mathcal{T} \\ n_m, & m \in [N] - \mathcal{T} \end{cases}$$



Example

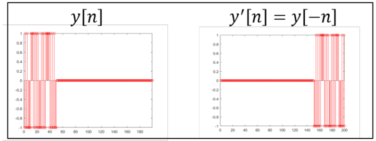


N-pt. FFT

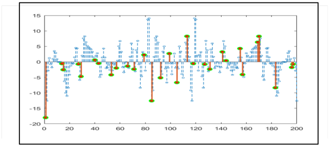


Sketch of x: $X[S]$

$\odot$

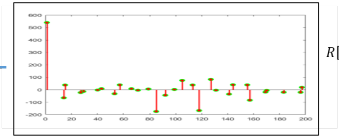


N-pt. FFT



Sketch of y': $Y'[S]$

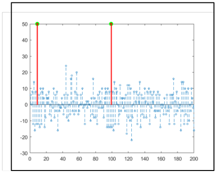
||



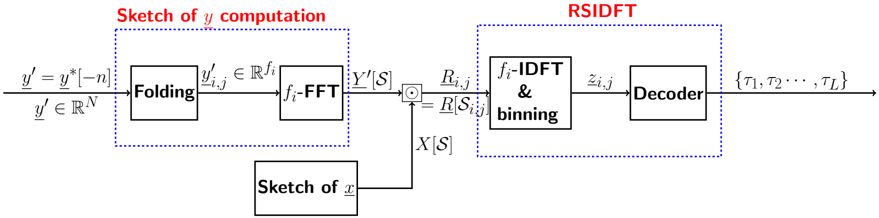
$R[S] = X[S] \odot Y'[S]$

RSIDFT

$\tau_1 = 10 \;\; \tau_2 = 100$



Sparse Fourier Transform Approach


Sketch of $\underline{y}$ computation
RSIDFT
$\underline{y}' = \underline{y}^*[-n]$
$\underline{y}' \in \mathbb{R}^N$
Folding
$\underline{y}'_{i,j} \in \mathbb{R}^{f_i}$
f_i-FFT
$\underline{Y}'[\mathcal{S}]$
$\odot$
$\underline{R}_{i,j}$
$= \underline{R}[\mathcal{S}_{i,j}]$
$X[\mathcal{S}]$
Sketch of $\underline{x}$
f_i-IDFT & binning
$\underline{z}_{i,j}$
Decoder
$\{\tau_1, \tau_2 \cdots, \tau_L\}$

$$\boxed{\underline{r} = \underset{3}{\mathcal{F}_N^{-1}} \ \{\underset{1}{\mathcal{F}_N\{\underline{x}\}} \odot \ \underset{2}{\mathcal{F}_N\{\underline{y}'\}}\}}$$

1. *Sketch of $\underline{x}$:* Assume $\underline{X}[l] = \mathcal{F}\{\underline{x}\}$ is precomputed at positions $l \in \mathcal{S}$.
2. *Sketch of $\underline{y}$:* Compute $\underline{Y}'[l] = \mathcal{F}\{\underline{y}'\}$ for $l \in \mathcal{S}$.
   - Only M non-zero values in $\underline{y}'$ - Efficient computation (folding and adding)
3. *Sparse $\mathcal{F}^{-1}$*:
   - Robust Sparse Inverse Fourier Transform (RSIDFT)
   - Efficient Implementation- sublinear time and sampling complexity

Robust Sparse Inverse Fourier Transform(RSIDFT)

Main Idea

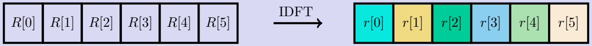
- Sub-sampling in frequency corresponds to aliasing in time
- Aliased coefficients $\Leftrightarrow$ parity check constraints of GLDPC codes
- CRT guided sub-sampling induces a code good for Peeling decoder
- R-FFAST- proposed by Pawar and Ramchandran 2014

Key modifications

- Optimized for the induced noise model
- Correlation peak is always positive
- Take advantage in decoding algorithm - sub-linear time complexity

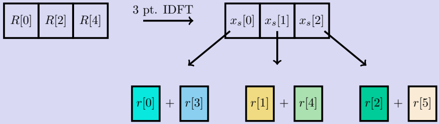
Aliasing and Sparse Graph Codes

IDFT Computation($N = 6$)

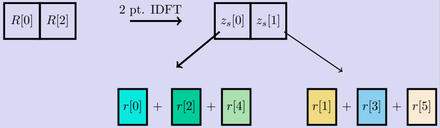


$R[0]$ $R[1]$ $R[2]$ $R[3]$ $R[4]$ $R[5]$
IDFT
$r[0]$ $r[1]$ $r[2]$ $r[3]$ $r[4]$ $r[5]$

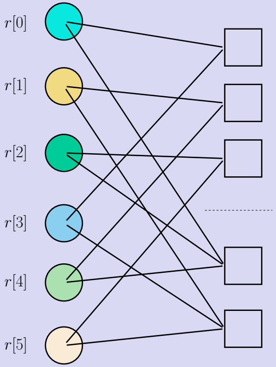
x_s: Sub-sampled by $P_1 = 2$



z_s: Sub-sampled by $P_2 = 3$



Factor graph



$x_s[0] = r[0] + r[3]$

$x_s[1] = r[1] + r[4]$

$x_s[2] = r[2] + r[5]$

$z_s[0] = r[0] + r[2] + r[4]$

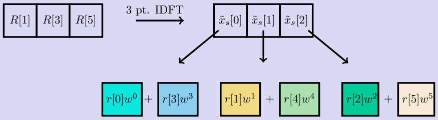
$z_s[1] = r[1] + r[3] + r[5]$

Aliasing and Sparse Graph Codes

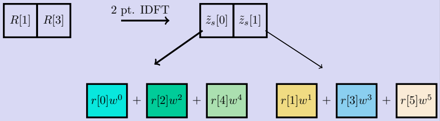
IDFT Computation($N=6$)

$$\boxed{R[0]}\boxed{R[1]}\boxed{R[2]}\boxed{R[3]}\boxed{R[4]}\boxed{R[5]} \xrightarrow{\text{IDFT}} \boxed{r[0]}\boxed{r[1]}\boxed{r[2]}\boxed{r[3]}\boxed{r[4]}\boxed{r[5]}$$

$\tilde{x}_s$: Sub-sampled by $P_1 = 2$ (shifted)

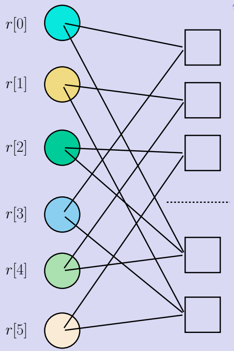


$\tilde{z}_s$: Sub-sampled by $P_2 = 3$ (shifted)



$R[1]$ $R[3]$
2 pt. IDFT
$\tilde{z}_s[0]$ $\tilde{z}_s[1]$
$r[0]w^0$ + $r[2]w^2$ + $r[4]w^4$
$r[1]w^1$ + $r[3]w^3$ + $r[5]w^5$

Factor graph



$w = e^{j\frac{2\pi}{6}}$

$x_s[0] = r[0] + r[3]$
$\tilde{x}_s[0] = r[0]w^0 + r[3]w^3$

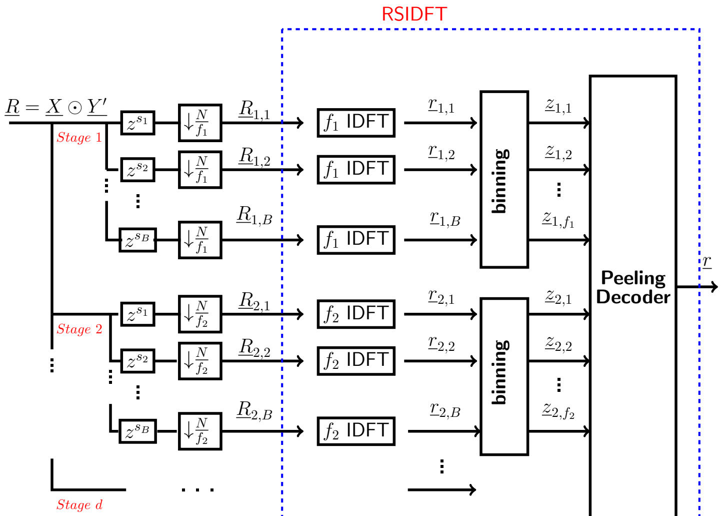
$x_s[1] = r[1] + r[4]$
$\tilde{x}_s[1] = r[1]w^1 + r[4]w^4$

$x_s[2] = r[2] + r[5]$
$\tilde{x}_s[2] = r[2]w^2 + r[5]w^5$

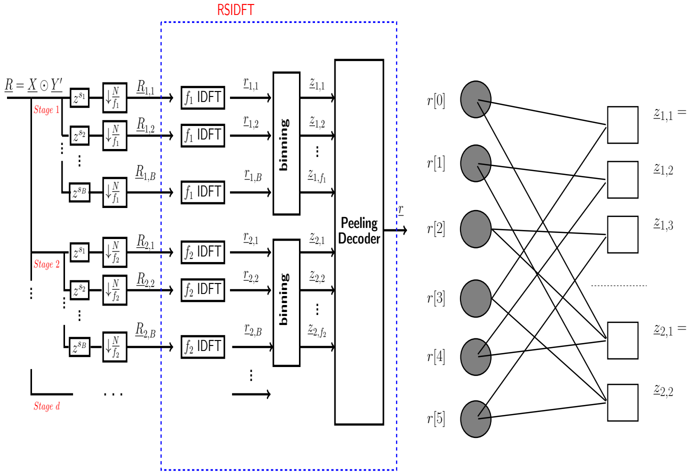
$z_s[0] = r[0] + r[2] + r[4]$
$\tilde{z}_s[0] = r[0]w^0 + r[2]w^2 + r[4]w^4$

$z_s[1] = r[1] + r[3] + r[5]$
$\tilde{z}_s[1] = r[1]w^1 + r[3]w^3 + r[5]w^5$

RSIDFT Framework



RSIDFT Framework



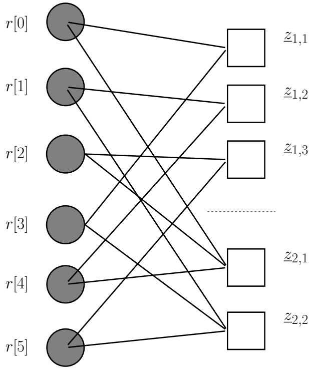
$$w_i^k := \left(e^{j\frac{2\pi s_i}{N}}\right)^k$$

$$\underline{z}_{1,1} = \begin{bmatrix} r_{1,1}[1] & = & r[0]w_1^0 + r[3]w_1^3 \\ r_{1,2}[1] & = & r[0]w_2^0 + r[3]w_2^3 \\ \vdots & & \\ r_{1,B}[1] & = & r[0]w_B^0 + r[3]w_B^3 \end{bmatrix}$$

$$\underline{z}_{2,1} = \begin{bmatrix} r_{2,1}[1] & = & r[0]w_1^0 + r[2]w_1^2 + r[4]w_1^4 \\ r_{2,2}[1] & = & r[0]w_2^0 + r[2]w_2^2 + r[4]w_2^4 \\ \vdots & & \\ r_{2,B}[1] & = & r[0]w_B^0 + r[2]w_B^2 + r[4]w_B^4 \end{bmatrix}$$

RSIDFT-Decoding (Peeling Decoder)

$$w_i^k := \left(e^{j\frac{2\pi s_i}{N}}\right)^k$$



$$\underline{z}_{1,1} = \begin{bmatrix} r_{1,1}[1] & = & r[0]w_1^0 + r[3]w_1^3 \\ r_{1,2}[1] & = & r[0]w_2^0 + r[3]w_2^3 \\ \vdots & & \\ r_{1,B}[1] & = & r[0]w_B^0 + r[3]w_B^3 \end{bmatrix}$$

$$\underline{z}_{2,1} = \begin{bmatrix} r_{2,1}[1] & = & r[0]w_1^0 + r[2]w_1^2 + r[4]w_1^4 \\ r_{2,2}[1] & = & r[0]w_2^0 + r[2]w_2^2 + r[4]w_2^4 \\ \vdots & & \\ r_{2,B}[1] & = & r[0]w_B^0 + r[2]w_B^2 + r[4]w_B^4 \end{bmatrix}$$

Observations:

$$\underline{z}_{i,k} = \begin{bmatrix} r_{i,1}[k] \\ r_{i,2}[k] \\ \vdots \\ r_{i,B}[k] \end{bmatrix}^T$$

Decoding- 3 steps

1. Bin Classification
2. Position Identification
3. Peeling Process

Decoder

Bin Classification

- Classify each check-node - Zero-ton / Single-ton / Multi-ton
- Threshold constraints on first observation $z_{i,k}[1] = z$
- Threshold varies with η
  - different for exact($\eta = 0$) and approximate matching

$$\widehat{\mathcal{H}}_{i,j} = \begin{cases} \mathcal{H}_z & z/M < \gamma_1 \\ \mathcal{H}_s & \gamma_1 < z/M < \gamma_2 \\ \mathcal{H}_d & \gamma_2 < z/M < \gamma_3 \\ \mathcal{H}_m & z/M > \gamma_3 \end{cases}$$

where $(\gamma_1, \gamma_2, \gamma_3) = (\frac{1-2\eta}{2}, \frac{3-4\eta}{2}, \frac{5-6\eta}{2})$

Decoder

Position Identification

- Observation:

$$\underline{z}_{i,k} = \begin{bmatrix} 1 & 1 & \dots & 1 \\ \omega^{ks_2} & \omega^{(k+f_i)s_2} & \dots & \omega^{(k+(g_i-1)f_i)s_2} \\ \vdots & \vdots & \ddots & \vdots \\ \omega^{ks_B} & \omega^{(k+f_i)s_B} & \dots & \omega^{(k+(g_i-1)f_i)s_B} \end{bmatrix} \times \begin{bmatrix} r[k+(0)f_i] \\ r[k+(1)f_i] \\ \vdots \\ r[k+(g_i-1)f_i] \end{bmatrix}$$

- Column that gives maximum correlation with the observation

$$\boxed{\hat{k} = \underset{k \in \{j+lf_i\}}{\arg\max} \ \underline{z}_{i,j}^{\dagger} \mathbf{W}[:, l]}$$

Decoder

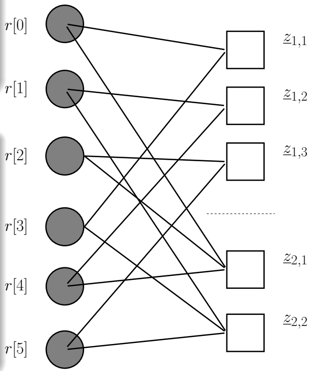
Peeling Process:

Exact Matching

- Remove a decoded variable node's contribution from all participating bin nodes

Approximate Matching

- Remove a decoded variable node's contribution only from neighboring single-tons and double-tons
- Avoid error propagation



$$w_i^k := \left(e^{j\frac{2\pi s_i}{N}}\right)^k$$

$$\underline{z}_{1,1} = \begin{bmatrix} r_{1,1}[1] & = & r[0]w_1^0 + r[3]w_1^3 \\ r_{1,2}[1] & = & r[0]w_2^0 + r[3]w_2^3 \\ \vdots & & \\ r_{1,B}[1] & = & r[0]w_B^0 + r[3]w_B^3 \end{bmatrix}$$

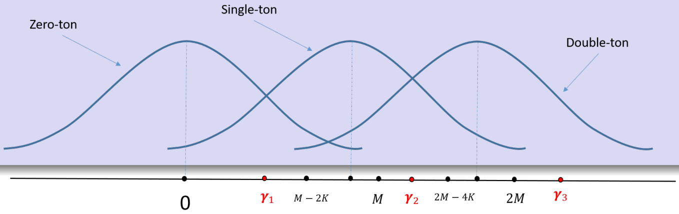
$$\underline{z}_{2,1} = \begin{bmatrix} r_{2,1}[1] & = & r[0]w_1^0 + r[2]w_1^2 + r[4]w_1^4 \\ r_{2,2}[1] & = & r[0]w_2^0 + r[2]w_2^2 + r[4]w_2^4 \\ \vdots & & \\ r_{2,B}[1] & = & r[0]w_B^0 + r[2]w_B^2 + r[4]w_B^4 \end{bmatrix}$$

Error Analysis

Error Events

- *$\mathcal{E}_1$-Bin Classification*: Bin is wrongly classified
- *$\mathcal{E}_2$-Pos. Identification*: Position of singleton is identified wrongly, given a singleton
- *$\mathcal{E}_3$-Peeling Process*: Peeling process fails to recover the L significant correlation coefficients, given $\mathbb{P}(\mathcal{E}_1) = \mathbb{P}(\mathcal{E}_2) = 0$

$\mathcal{E}_1$-Bin Classification



Error Analysis

$\mathcal{E}_2$-*Pos. Identification*

- $\underline{z}_{i,j} = r[j_p]\ \underline{w}_{j_p} + \sum_{k\neq p} n_k \underline{w}_{j_k}$
- Mutual Incoherence property to bound the cross-correlation(noise) term
  - $\log N$ measurements (shifts) suffices

$\mathcal{E}_3$-*Peeling Process*

- Tools from Coding theory to analyze Sparse Graph Codes
- Density Evolution to quantify Error Probability
- # of check-nodes is a function of sparsity (query length)

Error Analysis

Error Events

- *$\mathcal{E}_1$-Bin Classification*: Bin is wrongly classified
- *$\mathcal{E}_2$-Pos. Identification*: Position of singleton is identified wrongly, given a singleton
- *$\mathcal{E}_3$-Peeling Process*: Peeling process fails to recover the L significant correlation coefficients, given $\mathbb{P}(\mathcal{E}_1) = \mathbb{P}(\mathcal{E}_2) = 0$

Error Probability

$$\begin{aligned}\mathbb{P}(\mathcal{E}_{\text{total}}) &\leq \mathbb{P}(\mathcal{E}_1) + \mathbb{P}(\mathcal{E}_2) + \mathbb{P}(\mathcal{E}_3) \\ &\leq 6e^{-\frac{N^{\mu+\alpha-1}(1-6\eta)^2}{16}} + 2e^{-N^{\mu+\alpha-1}\ c_1(\eta)} + e^{-c_3 N^{c_4 \alpha}}\end{aligned}$$

$$\boxed{\mathbb{P}(\mathcal{E}_{\text{total}}) \to 0 \ \text{ if } \ \alpha > 1-\mu}$$

Complexity Analysis

Sample Complexity

Total # of samples required (S) $= O\left(dBN^{\alpha}\right) = O(N^{1-\mu}\log N)$

Computational Complexity

$$\underline{r} = \underset{II}{\mathcal{F}_N^{-1}}\left\{\mathcal{F}_N\{\underline{x}\} \odot \underset{I}{\mathcal{F}_N\{\underline{y}'\}}\right\}$$

- Sketch of Query:

$$C_I = dB\,(\underbrace{N^{\mu}}_{\text{Folding}} + \underbrace{N^{\alpha}\log N^{\alpha}}_{\text{Shorter FFTs}}) = O(\max(N^{1-\mu}\log^2 N, N^{\mu}\log N))$$

- RSIDFT:

$$C_{II} = dB\left(\underbrace{O(N^{\alpha}\log N^{\alpha})}_{\text{Shorter IFFTs /block/stage}} + \underbrace{L\ N^{1-\alpha}}_{\text{Correlations}}\right) = O(\max(N^{1-\mu}\log^2 N, N^{\mu+\lambda}\log N))$$

$$\boxed{C_{\text{total}} = \max(C_{\text{I}}, C_{\text{II}}) = O(\max(N^{1-\mu}\log^2 N, N^{\mu+\lambda}\log N))}$$

Simulation Results

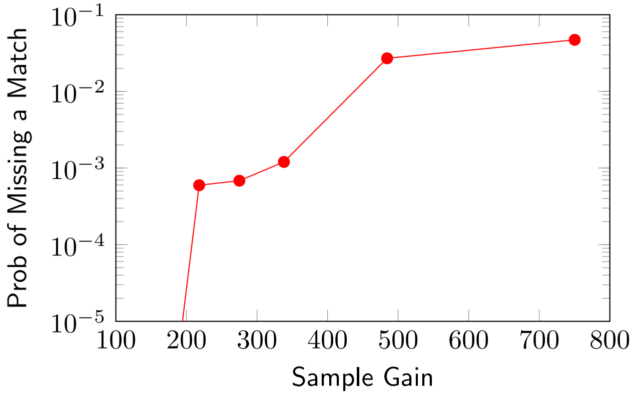


Figure: Plot of Probability of Missing a Match vs. Sample Gain for Exact Matching of a substring of length $M = 10^5$ from a equiprobable binary $\{+1,-1\}$ sequence of length $N = 10^{12}$, divided into $G = 10^5$ blocks each of length $\tilde{N} = 10^7$. The substring was simulated to repeat in $L = 10^6$ locations uniformly at random.

Simulation Results

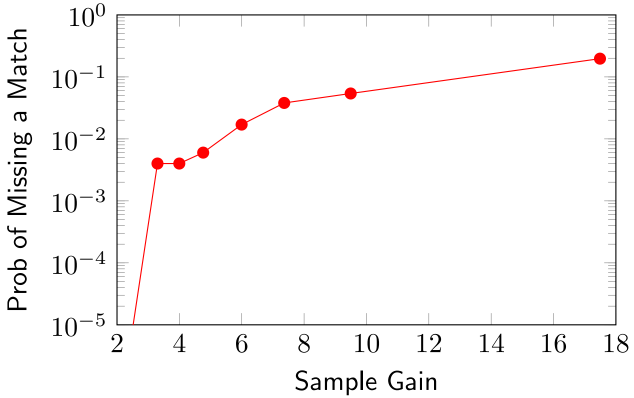


Figure: Plot of Probability of Missing a Match vs. Sample Gain for Exact Matching of a substring of length $M = 10^3$ from a equiprobable binary {+1,-1} sequence of length $N = 10^{12}$, divided into $G = 10^6$ blocks each of length $\tilde{N} = 10^6$. The substring was simulated to repeat in $L = 10^6$ locations uniformly at random.

Questions?

Thank you!