

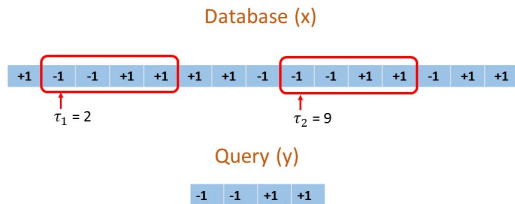
# Sub-string/Pattern Matching in Sub-linear Time Using a Sparse Fourier Transform Approach

Krishna R. Narayanan

Department of Electrical and Computer Engineering  
Texas A&M University

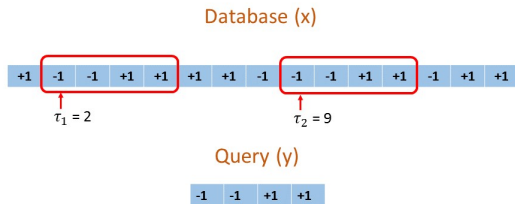


# Problem Statement



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

# Problem Statement

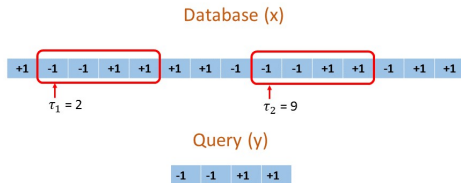


- **Database/String:**  $\underline{x} = [x[0], x[1], \dots, x[N-1]]$  (length  $N$ )
- **Query/Substring:**  $\underline{y} = [y[0], y[1], \dots, y[M-1]]$  (length  $M = N^\mu$ )
- **Signal Model:**  $x[i]$ 's are i.i.d. r.v. from  $\mathcal{A} = \{+1, -1\}$  (extensions possible)

Determine all the  $L$  locations  $\underline{\tau} = [\tau_1, \tau_2, \dots, \tau_L]$  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  $d_H(\underline{y}, \underline{x}[\tau : \tau + M - 1]) \leq K$

# Notations



<i>Symbol</i>	<i>Meaning</i>
$N$	Size of the string or database in symbols
$M = N^\mu$	Length of the query in symbols
$L = N^\lambda$	Number of matches
$K$	$\max_{\tau} d_H(\underline{x}[\tau : \tau + M - 1], \underline{y})$
$\eta$	$\frac{K}{M}$
$G = N^\gamma$	Number of blocks
$\tilde{N} = N^{1-\gamma}$	Length of one block
$f_i = N^\alpha$	Length of smaller point IDFT at each branch
$g_i = N/f_i$	Sub-sampling parameter
$B$	Number of shifts also referred to as branches
$d$	Number of stages in the FFAST algorithm

# Main Result

## Theorem 1

Assume that a sketch of  $\underline{x}$  of size  $O(\frac{N}{M} \log N)$  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, with the number of matches  $L$  scaling as  $O(N^\lambda)$ , our algorithm has

- a sketching function for  $\underline{y}$  that computes  $O(\frac{N}{M} \log N) = O(N^{1-\mu} \log N)$  *samples*
- a *computational complexity* of  $O(\max(N^{1-\mu} \log^2 N, N^{\mu+\lambda} \log N))$
- a decoder that recovers all the  $L$  matching positions with a *failure probability that approaches zero asymptotically*

# Main Result

## Theorem 1

Assume that a sketch of  $\underline{x}$  of size  $O(\frac{N}{M} \log N)$  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, with the number of matches  $L$  scaling as  $O(N^\lambda)$ , our algorithm has

- a sketching function for  $\underline{y}$  that computes  $O(\frac{N}{M} \log N) = O(N^{1-\mu} \log N)$  *samples*
- a *computational complexity* of  $O(\max(N^{1-\mu} \log^2 N, N^{\mu+\lambda} \log N))$
- a decoder that recovers all the  $L$  matching positions with a *failure probability that approaches zero asymptotically*

## Note

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

# Some Prior Work

## Exact Matching

- **Boyer and Moore 1977**: First occurrence of the match (only  $\tau_1$ )
  - Average complexity -  $O(N^{1-\mu} \log N)$  (sublinear)
  - Worst case complexity -  $O(N \log N)$

## Approximate Matching

- **Chang and Marr 1994**: Generalization of Boyer and Moore 1977
  - Average complexity -  $O(NK/M \log N)$  (sub-linear only when  $K \ll M$ )
- **Andoni et al. 2013**:  $O(N/M^{0.359})$  (sub-linear even when  $K = O(M)$ )
  - Combinatorial in nature

## Sparse Fourier Transform Approach

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

# Motivation

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

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

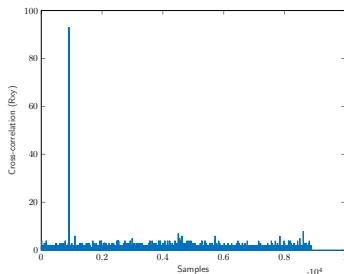
- **Naive implementation:**  $O(MN) = O(N^{1+\mu})$  (**super-linear** complexity)
- **Fourier Transform Approach:**  $O(N \log N)$  complexity

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

## Key Observation

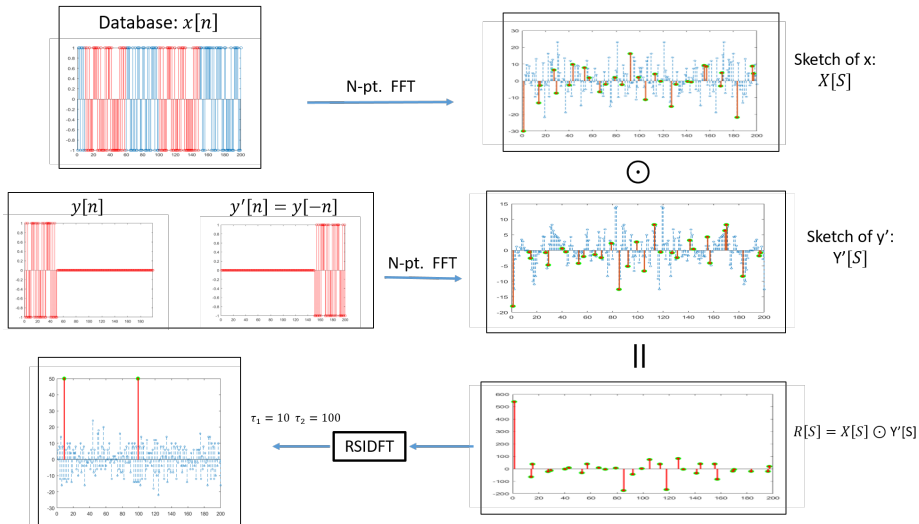
- $\underline{r}$  is **Sparse** with some noise.

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

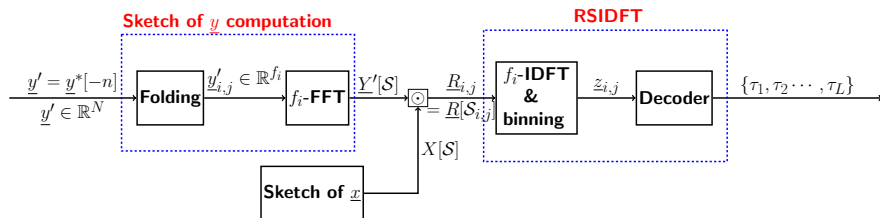




# Example



# Sparse Fourier Transform Approach



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

1. *Sketch of  $\underline{x}$*  : Assume  $\underline{X}[l] = \mathcal{F}\{\underline{x}\}$  is precomputed at positions  $l \in S$ .
2. *Sketch of  $\underline{y}$*  : Compute  $\underline{Y}'[l] = \mathcal{F}\{\underline{y}'\}$  for  $l \in S$ .
  - 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

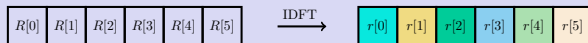
- **Sub-sampling** in frequency corresponds to **aliasing** in time
- Aliased coefficients  $\Leftrightarrow$  parity check constraints of **GLDPC codes**
- **CRT** guided sub-sampling induces a code good for **Peeling decoder**
- R-FFAST- proposed by Pawar and Ramchandran 2014

## Key modifications

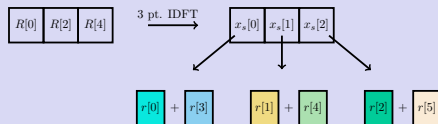
- Optimized for the induced noise model
- Correlation peak is always **positive**
- Take advantage in decoding algorithm - **sub-linear** time complexity

# Aliasing and Sparse Graph Codes

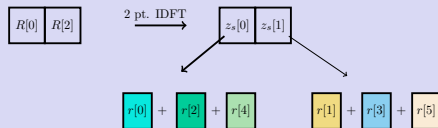
## IDFT Computation ( $N = 6$ )



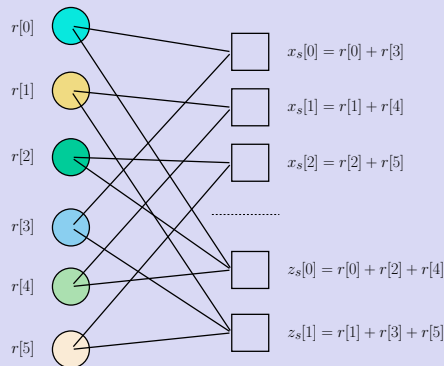
$x_s$ : Sub-sampled by  $P_1 = 2$



$z_s$ : Sub-sampled by  $P_2 = 3$

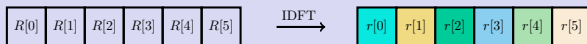


## Factor graph

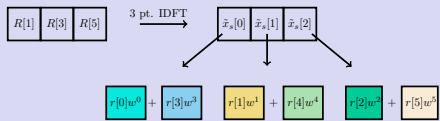


# Aliasing and Sparse Graph Codes

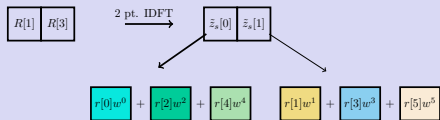
## IDFT Computation ( $N = 6$ )



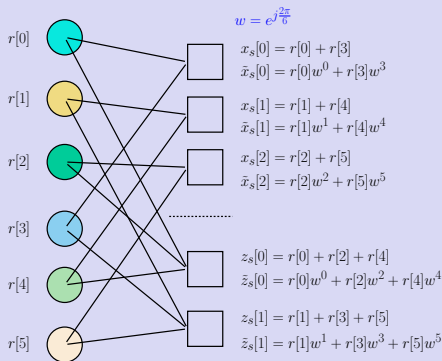
### $x_s$ : Sub-sampled by $P_1 = 2$ (shifted)



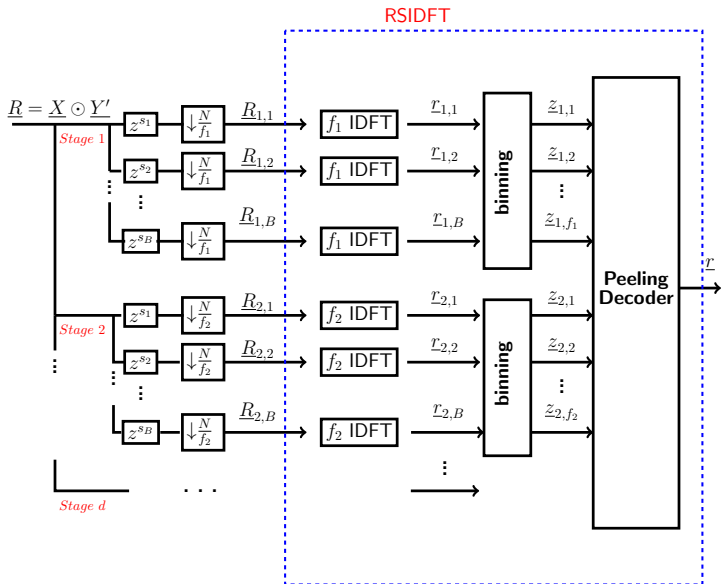
### $z_s$ : Sub-sampled by $P_2 = 3$ (shifted)



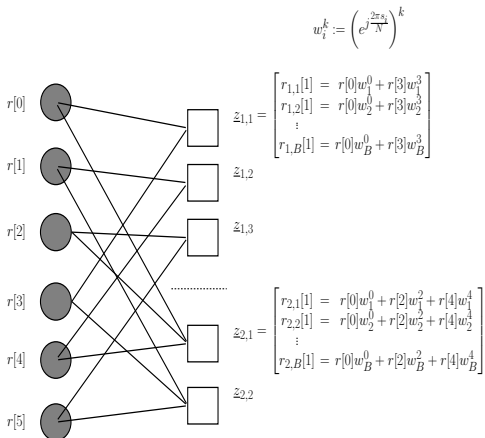
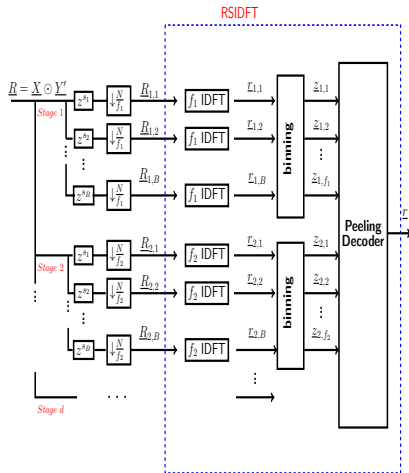
## Factor graph



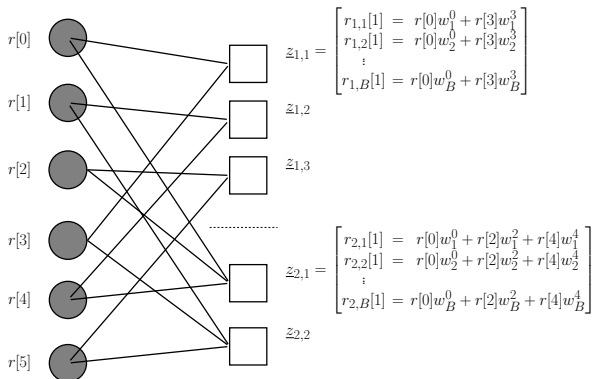
# RSIDFT Framework



# RSIDFT Framework



# RSIDFT-Decoding (Peeling Decoder)



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

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  $\eta$ 
  - different for exact( $\eta = 0$ ) and approximate matching

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

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:

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

- Column that gives **maximum correlation** with the observation

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

# Decoder

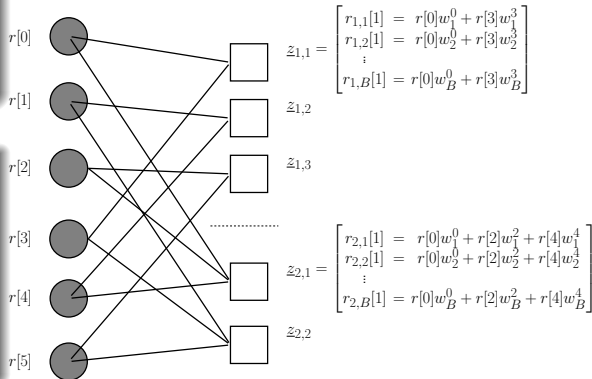
## Peeling Process:

### 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



# 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

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

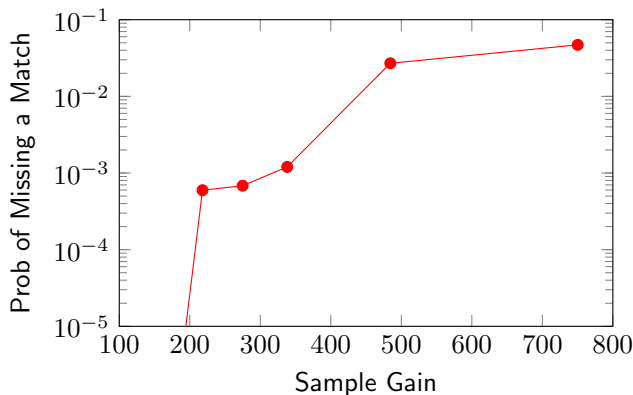
$$\boxed{\mathbb{P}(\mathcal{E}_{\text{total}}) \rightarrow 0 \text{ if } \alpha > 1 - \mu}$$

# Complexity Analysis

Sample Complexity

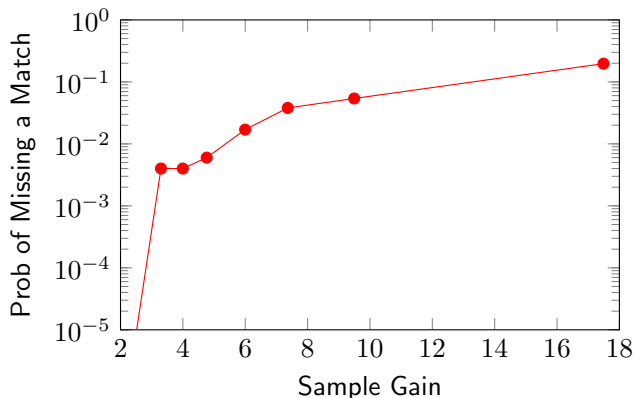
Computational Complexity

# 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!