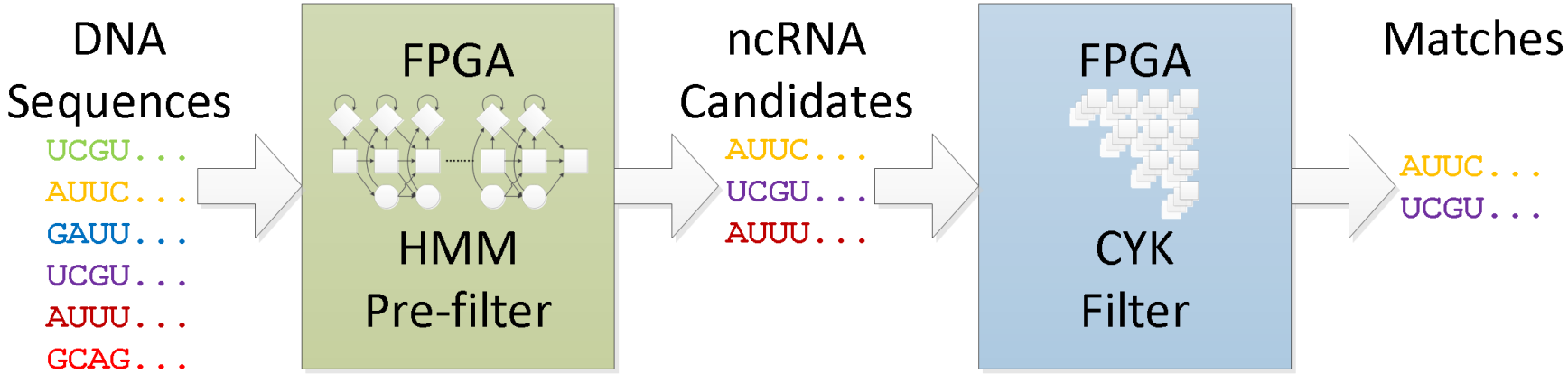
UNIVERSITY OF WASHINGTON
ELECTRICAL ENGINEERING

Accelerating ncRNA Homology Search with FPGAs

Nathaniel McVicar, Scott Hauck - ACME Lab
Walter L. Ruzzo - Computational & Synthetic Biology Group
University of Washington

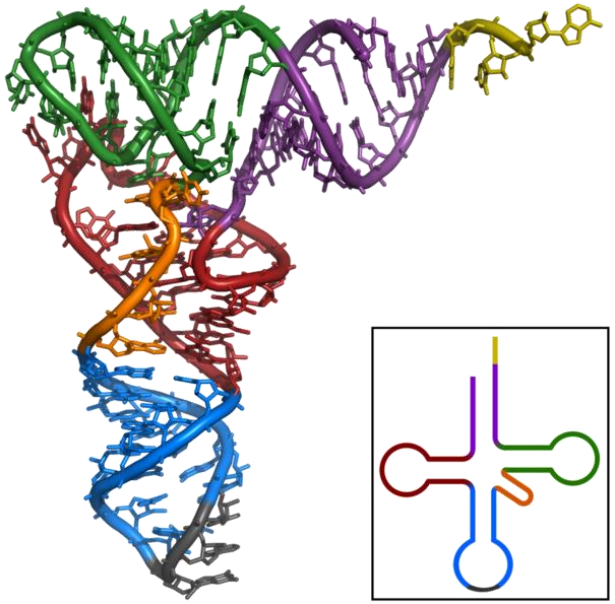
2-12-13

Overview


DNA
Sequences
UCGU...
AUUC...
GAUU...
UCGU...
AUUU...
GCAG...
FPGA
HMM
Pre-filter
ncRNA
Candidates
AUUC...
UCGU...
AUUU...
FPGA
CYK
Filter
Matches
AUUC...
UCGU...

What is ncRNA?

- First known RNA, mRNA translated to protein
- non coding RNAs do not code for protein
- ncRNA linked to disease in humans

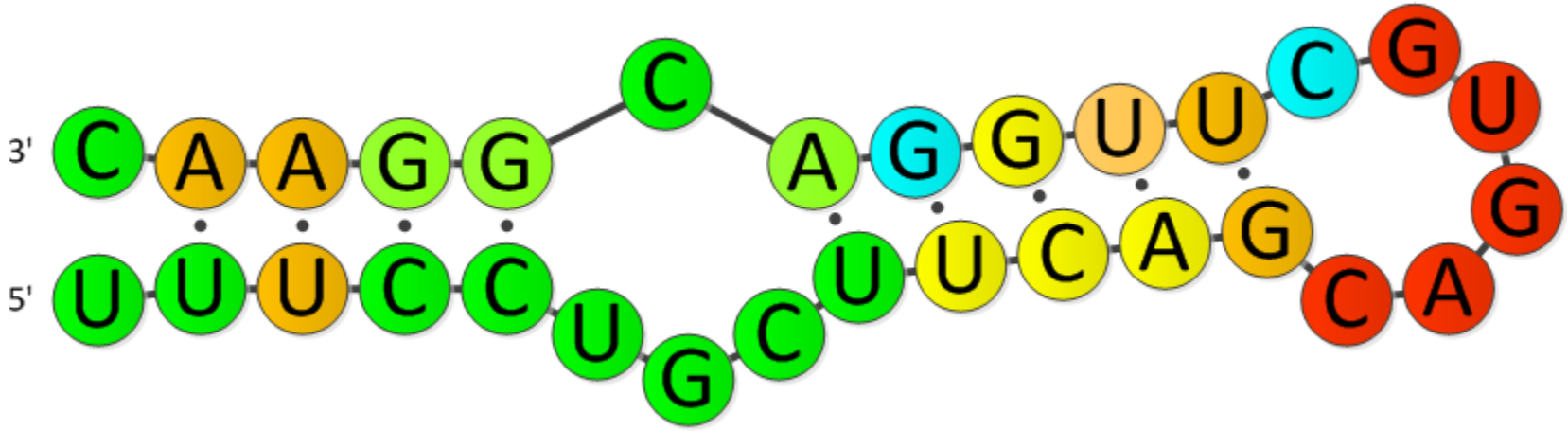


Yeast tRNA

Wikipedia

ncRNA Homology Search

- ncRNAs with the same function exist in many species
- Secondary structure critical to RNA's function
- IRE involved in regulation of iron metabolism genes

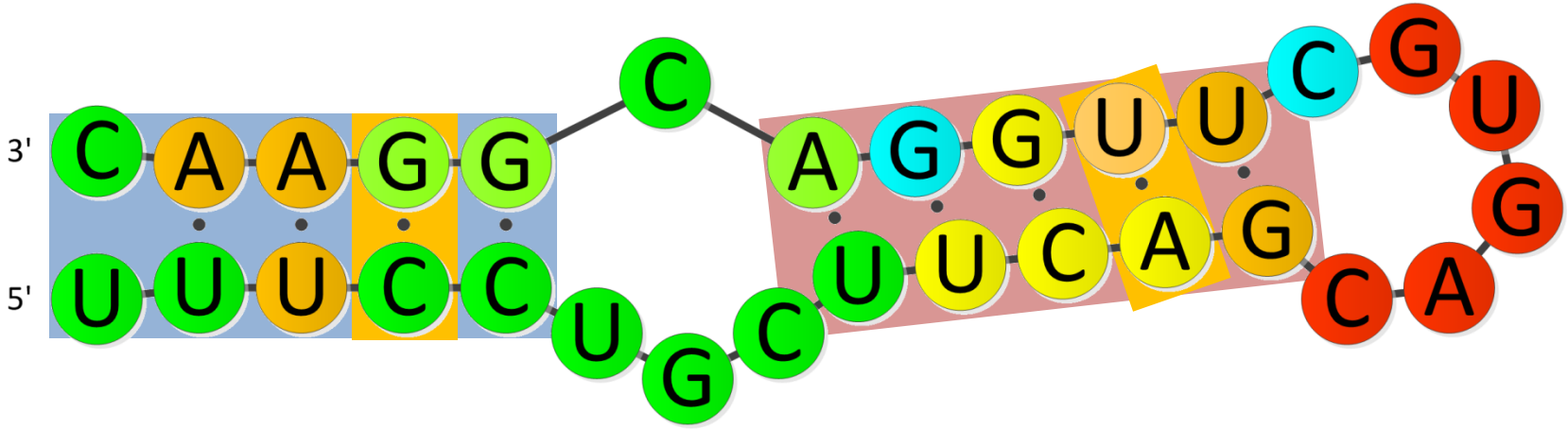


Iron response element (IRE)

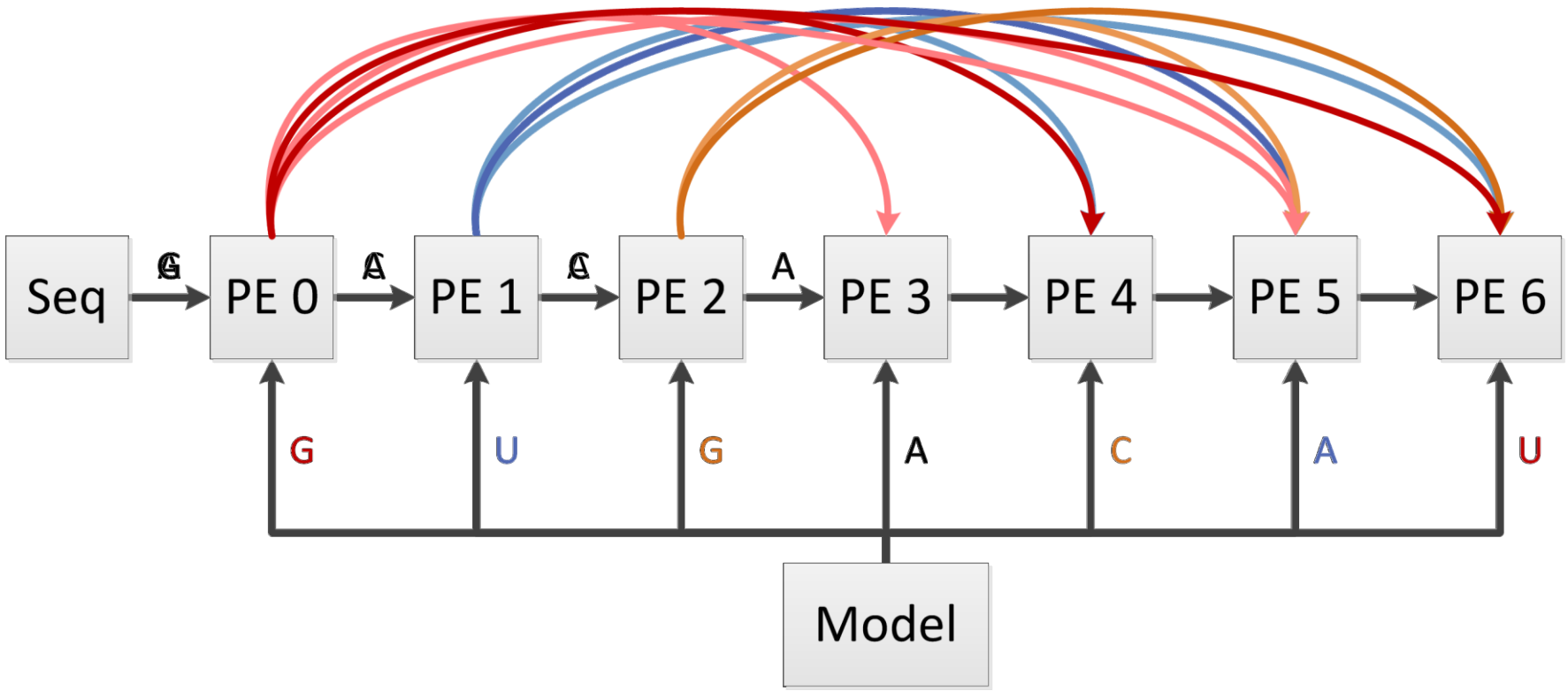
What is Covariance?

- Bases vary together so pairings preserved
  - Watson-Crick: A-U and C-G, wobble pair: U-G

```
rainbow smelt    AUUCUUGCCUCAACAGUGAUUGAACGGAAC
red junglefowl   AUUAUC..GGGGACAGUGUUUCCC.AUAAU
human            UUUCCUGCUUCAGCAGUGCUUGGACGGAAC
gray wolf        UCGUUC..GUCCUCAGUGCAGGGC.AACAG
```



Covariance Complicates Search



Infernal 1.0

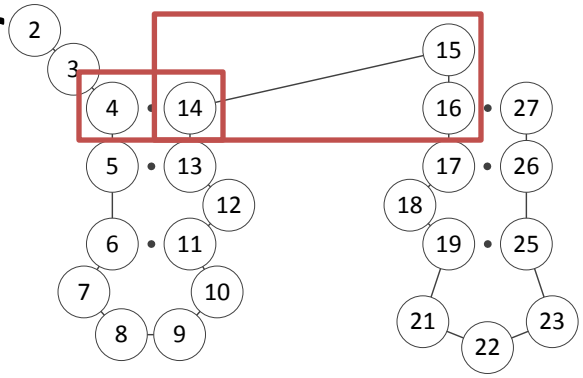
- Well known algorithms search for sequence
- Examples: BLAST, Smith-Waterman, Profile Hidden Markov Models, Burrows-Wheeler
- Infernal uses more costly Covariance Models

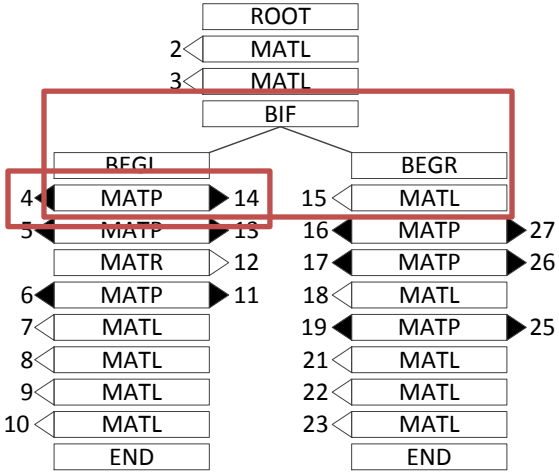
Nawrocki, Kolbe, Eddy, *Bioinformatics* (2009)

http://infernal.janelia.org/

Covariance Model

- Stochastic context-free grammar
- Can find probability of ncRNA sequence being a parse of tree
- Nodes match zero to two bases
- Differences from HMM type
  - MATP: matches pair of bases
  - BIF: bifurcation, allows multiple helices

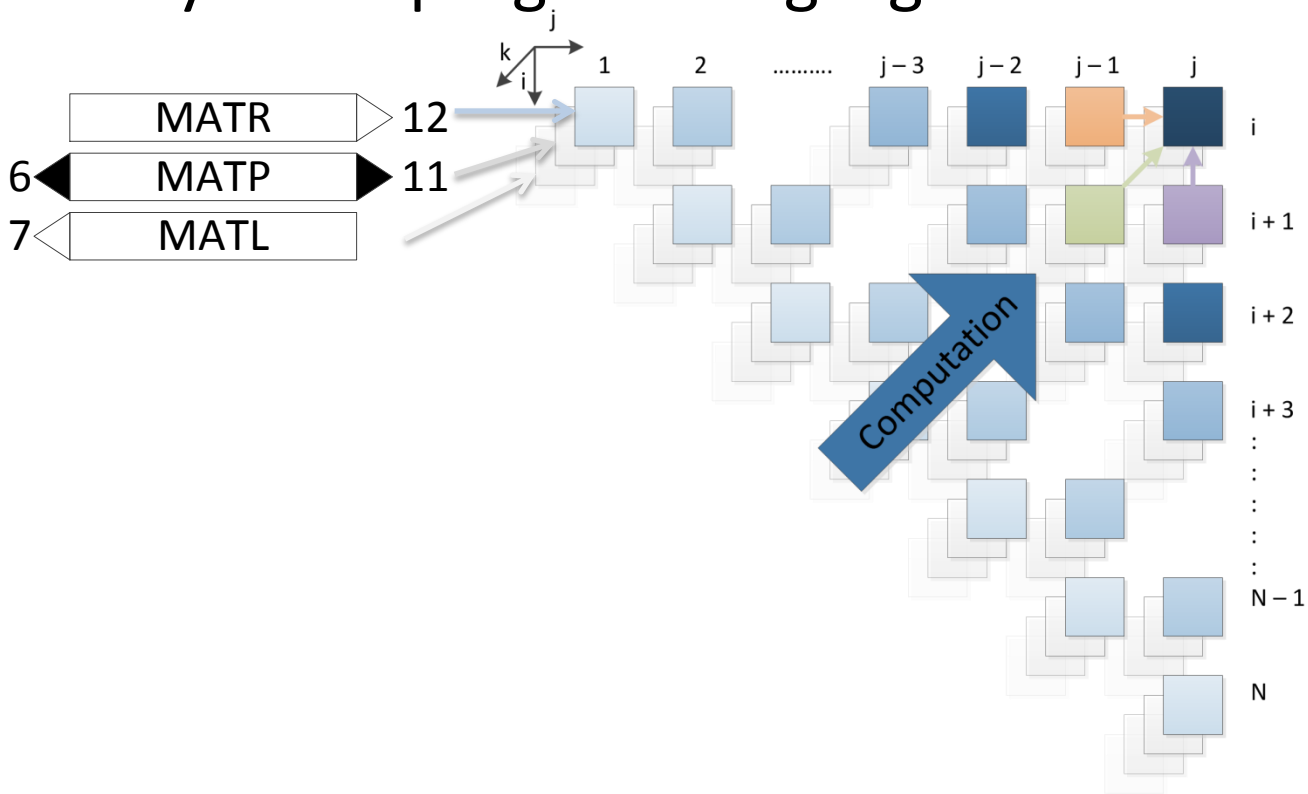




Eddy 2002

CYK Algorithm

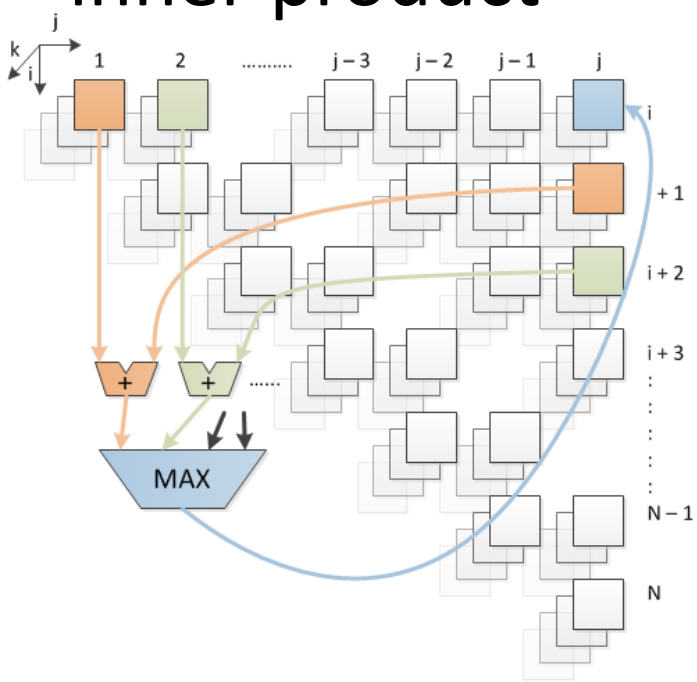
- Dynamic programming algorithm with 3D table

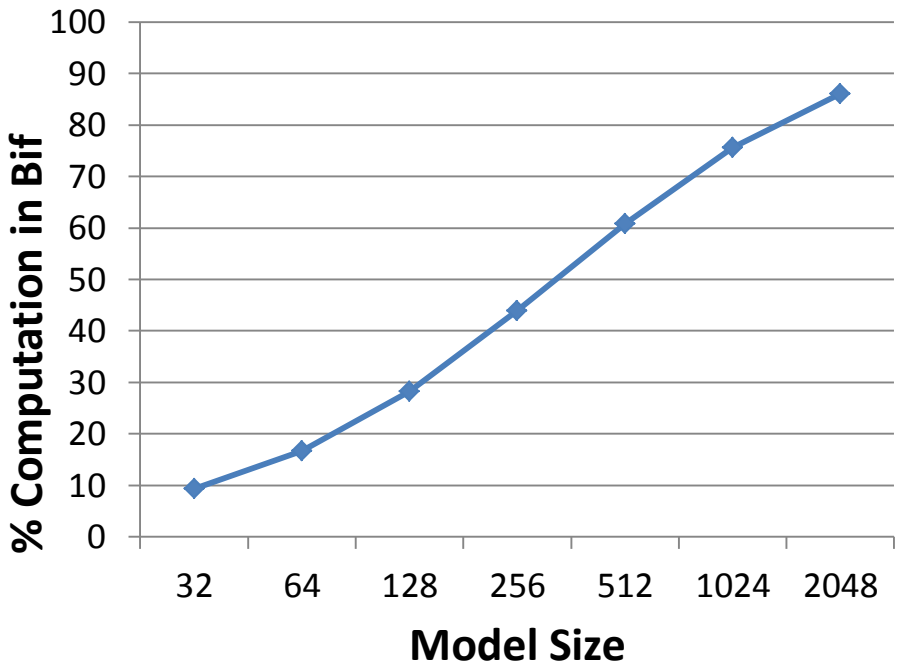


- Runtime $O(n^3+bif \cdot n^3)$ due to bifurcation

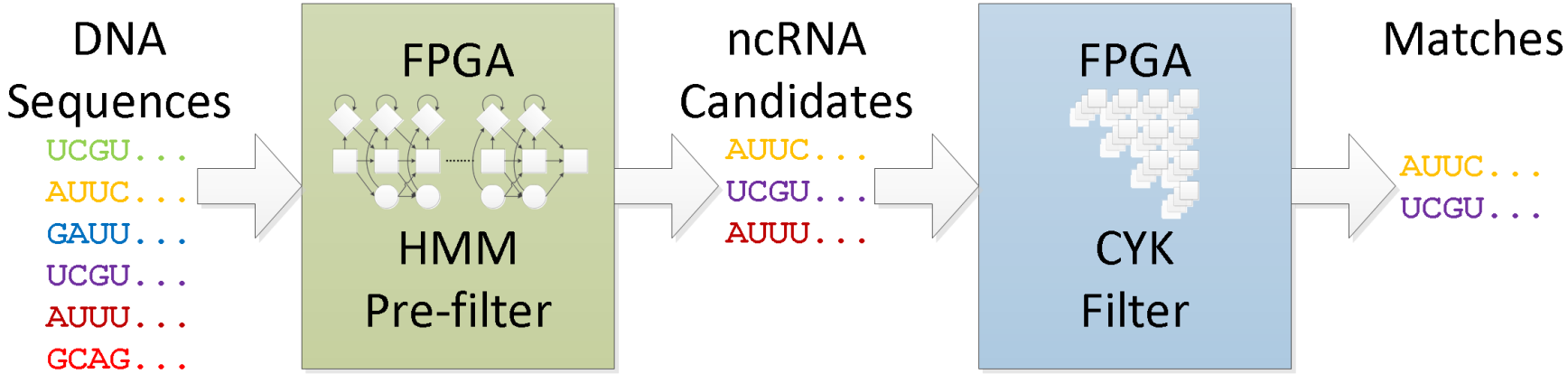
Bifurcation

- Most costly state
- Uses row column inner product



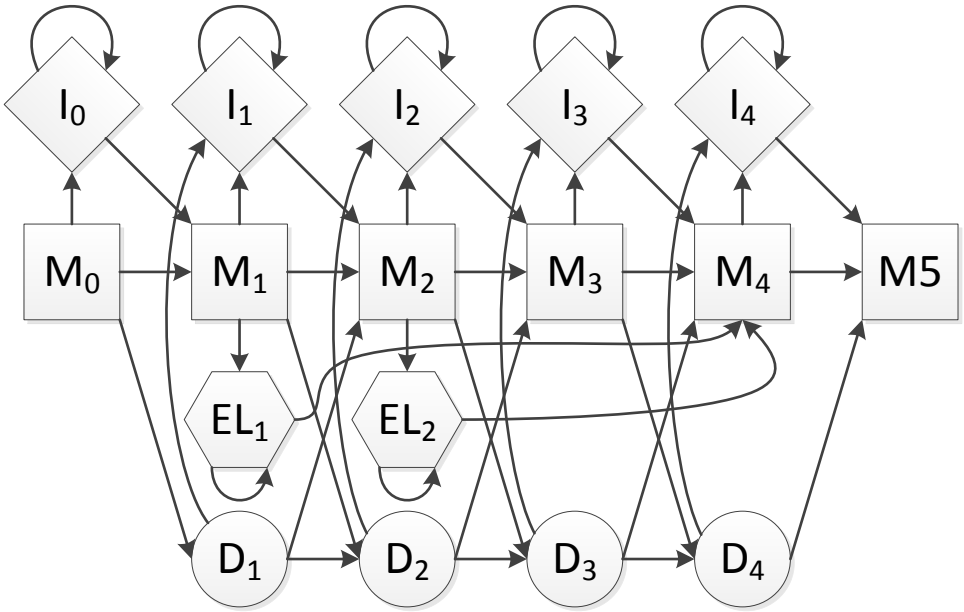


Overview Revisited



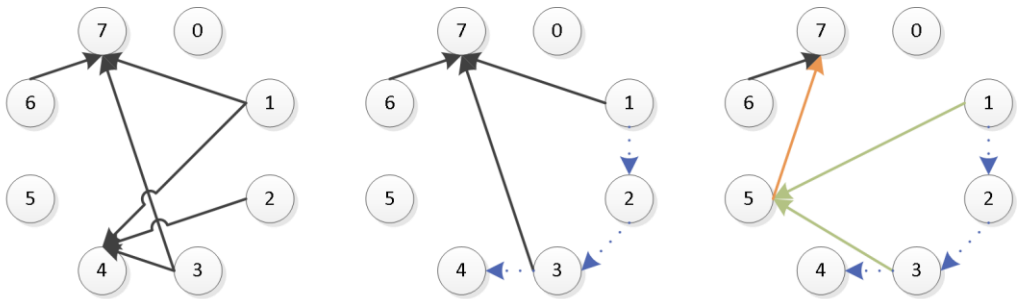
Hidden Markov Model Pre-filtering

- Runtime of CYK can be very long (weeks)
- HMMs can run much faster, $O(nl^2)$
  - Using Viterbi algorithm, similar to CYK for HMM



Combine EL States

- EL states introduce long data dependencies
  - Simple combination algorithm removes these



A. Initial Model

State	Cmd
1	W0
2	W1
3	W2
4	R 0, 1, 2
5	
6	W3
7	R 0, 2, 3

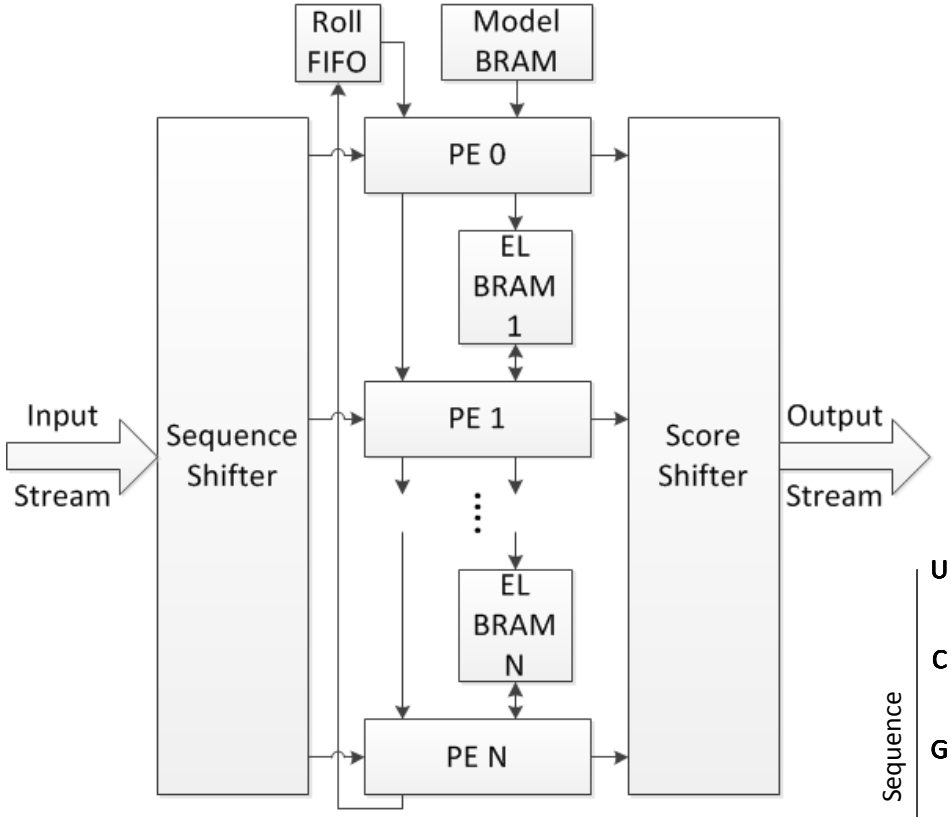
B. After Register

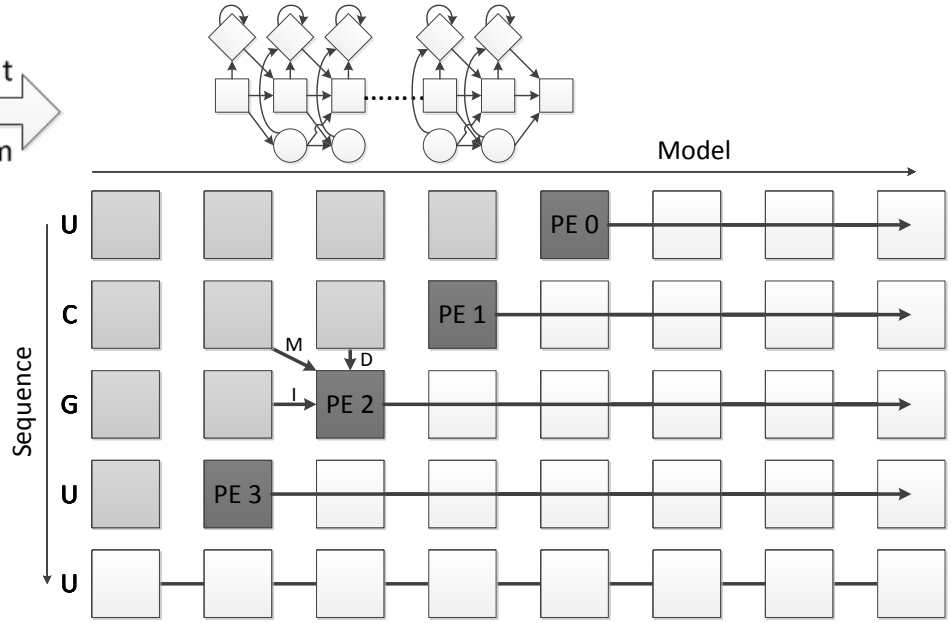
State	Cmd
1	W0, Reg
2	Reg
3	W1, Reg
4	Read Reg
5	
6	W2
7	R 0, 1, 2

C. After Combining

State	Cmd
1	W0, Reg
2	Reg
3	W1, Reg
4	Read Reg
5	R 0,1 W0
6	W1
7	R 0, 1

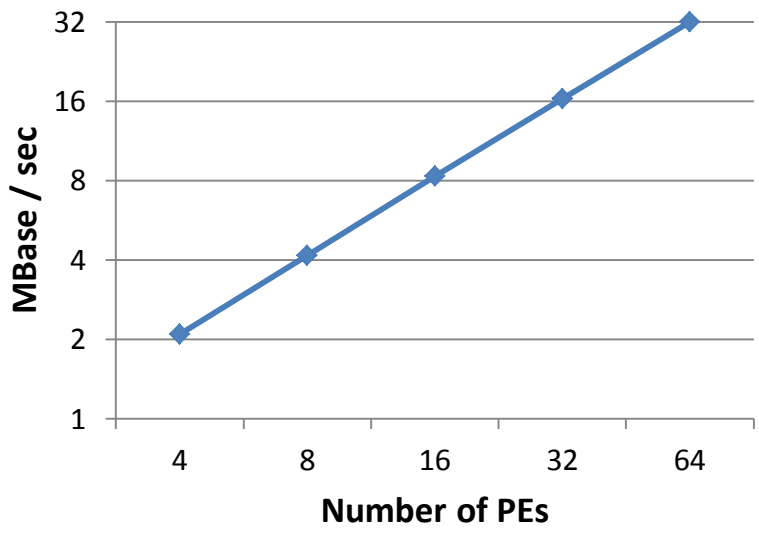
Viterbi on FPGA

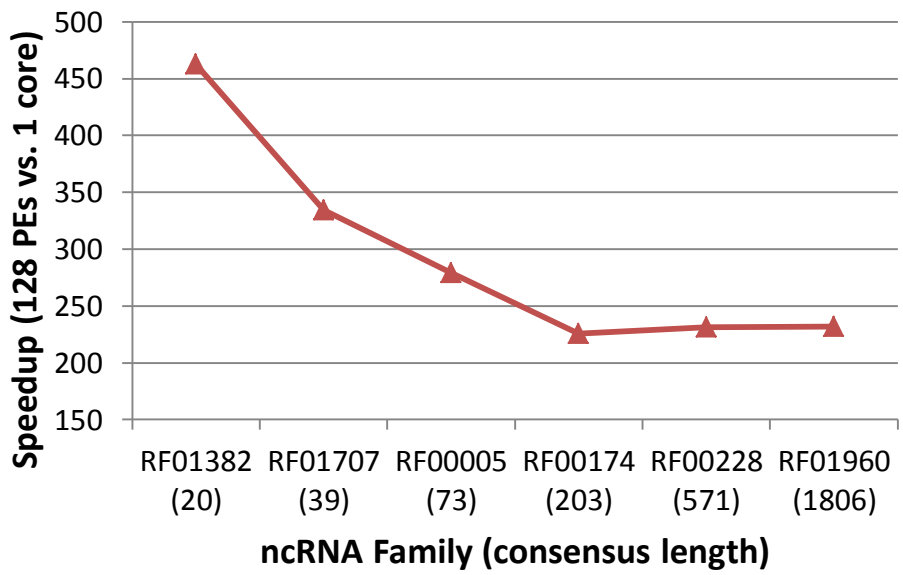




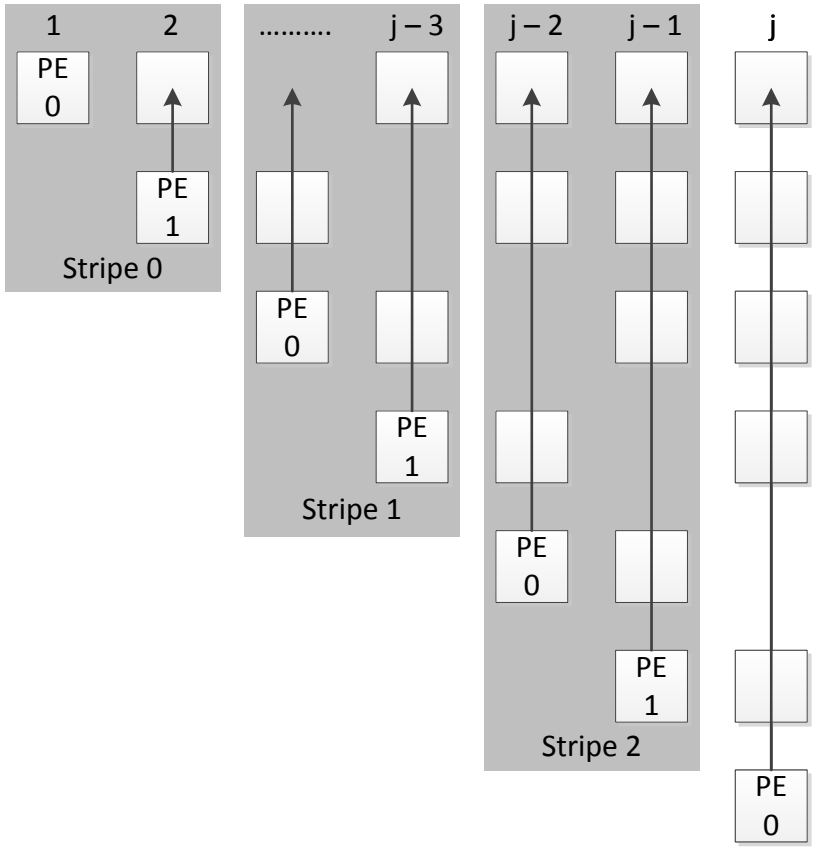
Viterbi Performance

- Scales linearly when adding PEs
- Speedup good across model lengths

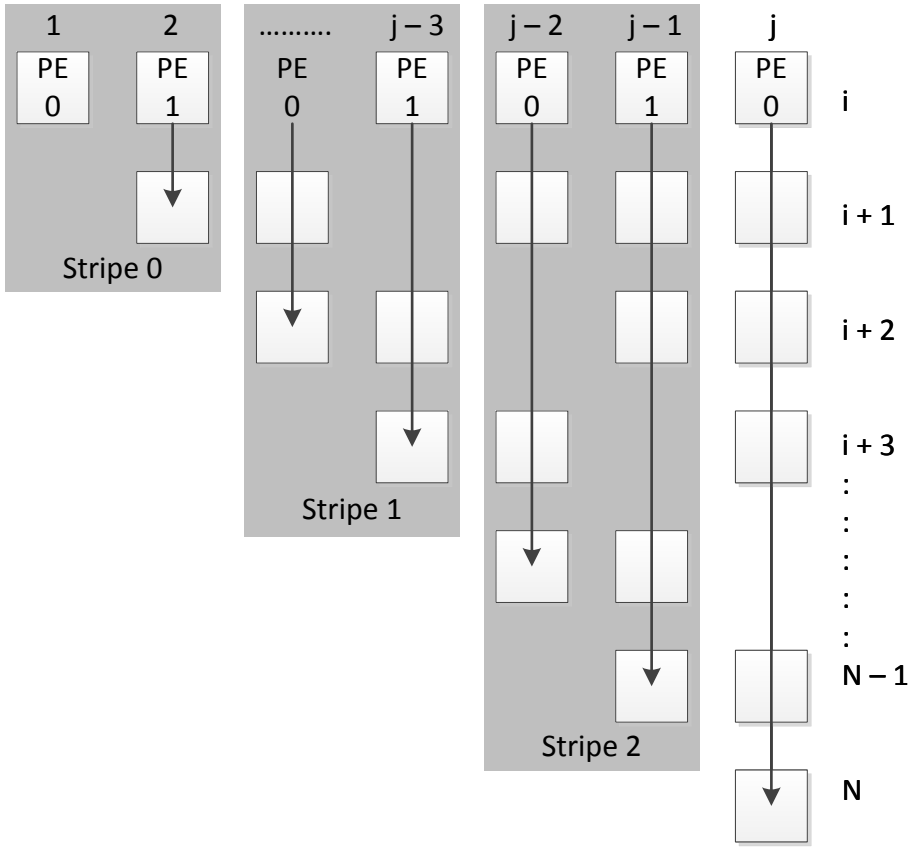




CYK PE Usage

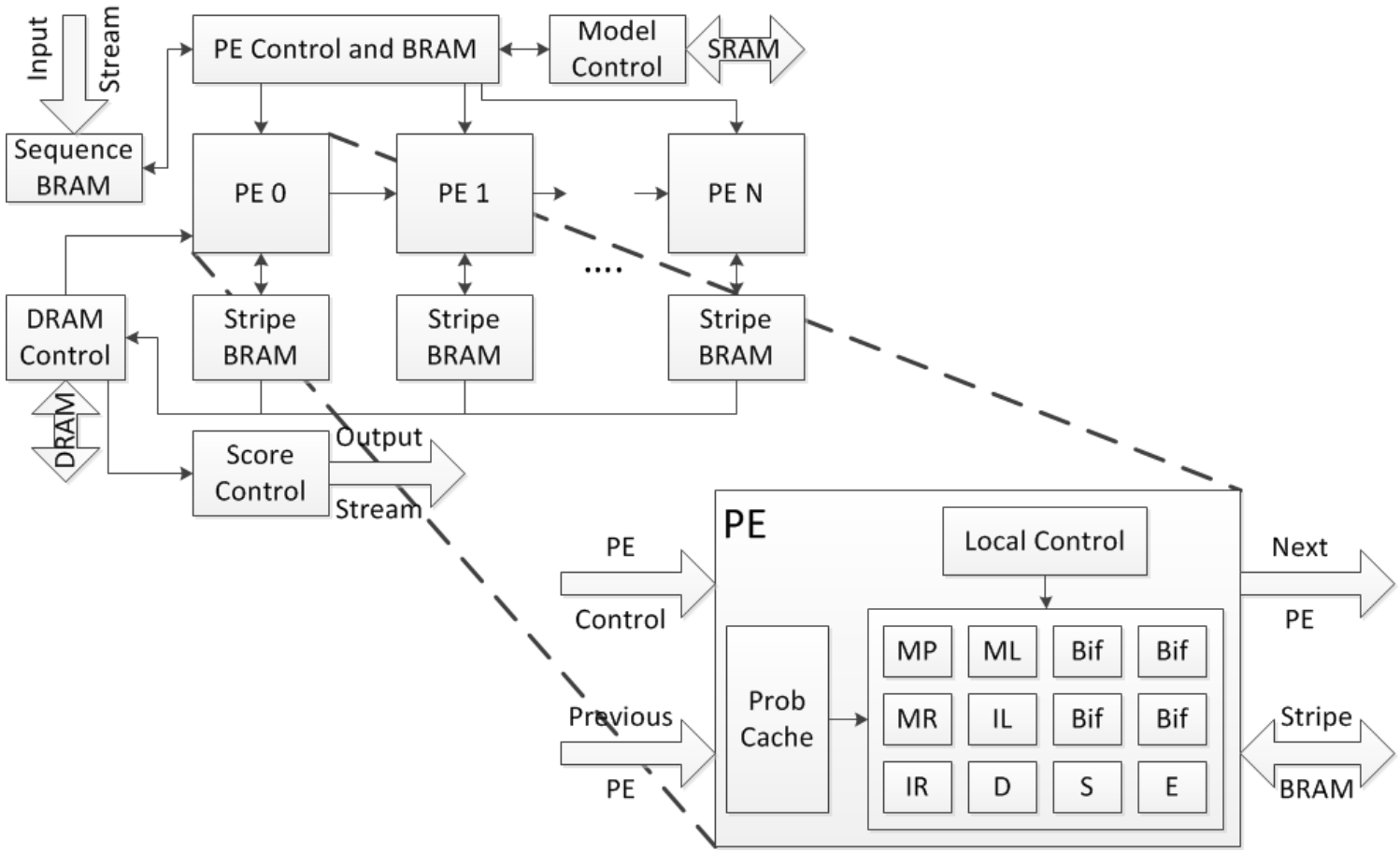


Regular Operation



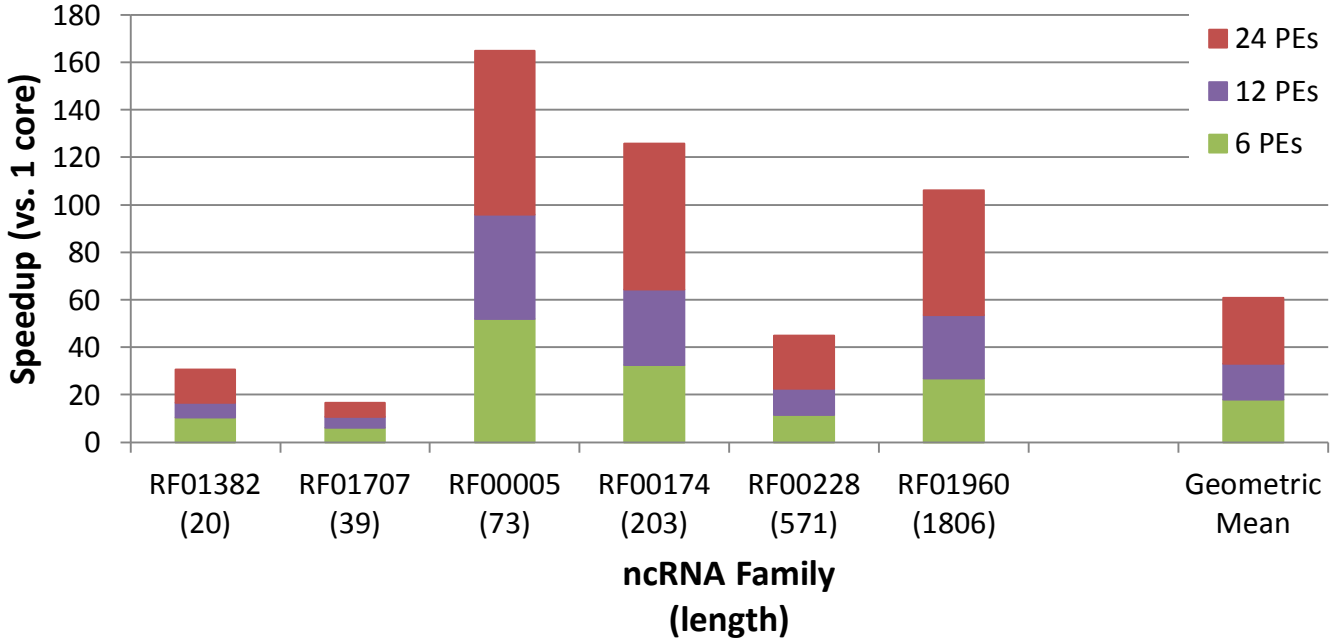
Bifurcation

CYK on FPGA


Input
Stream
PE Control and BRAM
Model Control
SRAM
Sequence BRAM
PE 0
PE 1
PE N
....
DRAM Control
Stripe BRAM
Stripe BRAM
Stripe BRAM
DRAM
Score Control
Output
Stream
PE
Local Control
PE
Control
Next
PE
Prob Cache
MP
ML
Bif
Bif
MR
IL
Bif
Bif
IR
D
S
E
Previous
PE
Stripe
BRAM

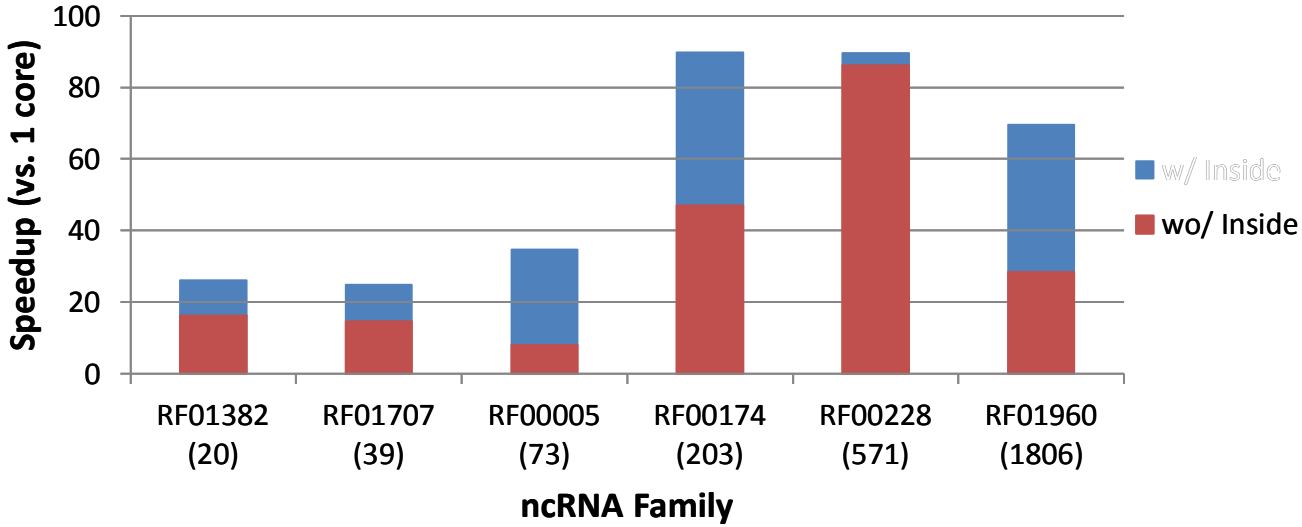
CYK Performance

- CYK Speedup much more variable than Viterbi
- Scales well with additional PEs



Estimated System Performance

Infernal software pipeline



Summary

- ncRNA is hugely important to life and secondary structure determines ncRNA function
- FPGAs provide massive parallelism for streaming application
- Applications with relatively complex control can be adapted to the streaming model
- For ncRNA homology search, runtimes can be reduced from weeks to hours
- One 4U server with 48 FPGAs could outperform 2000 full CPU cores

Thanks and Acknowledgments

- This material is based upon work supported by the National Science Foundation Graduate Research Fellowship under Grant No. DGE-0718124
- Pico Computing
- You, for listening

Any Questions?

- ncRNA is hugely important to life and secondary structure determines ncRNA function
- FPGAs provide massive parallelism for streaming application
- Applications with relatively complex control can be adapted to the streaming model
- For ncRNA homology search, runtimes can be reduced from weeks to hours
- One 4U server with 48 FPGAs could outperform 2000 full CPU cores

CYK Equations

$$S_{ij}^{y} = \begin{cases} \max_{z} [S_{i+1,j-1}^{z} + \log T_{yz} + \log E_{x_i,x_j}^{y}] & \text{match pair} \\ \max_{z} [S_{i+1,j}^{z} + \log T_{yz} + \log E_{x_i}^{y}] & \text{match/insert left} \\ \max_{z} [S_{i,j-1}^{z} + \log T_{yz} + \log E_{x_j}^{y}] & \text{match/insert right} \\ \max_{z} [S_{i,j}^{z} + \log T_{yz}] & \text{delete} \\ \max_{i<k\le j} [S_{i,k}^{y_{left}} + S_{k+1,j}^{y_{right}}] & \text{bifurcation} \end{cases}$$

Viterbi Equations

$$N(i) = N(i-1) + tr(N, N)$$

$$B(i) = \max \left\{ \begin{array}{l} N(i) + tr(N, B) \\ J(i-1) + tr(J, B) \end{array} \right\}$$

$$M(i,j) = e(M_j, \mathcal{S}[i]) + \max \left\{ \begin{array}{l} M(i-1, j-1) + tr(M_{j-1}, M_j) \\ I(i-1, j-1) + tr(I_{j-1}, M_j) \\ D(i-1, j-1) + tr(D_{j-1}, M_j) \\ B(i) + tr(B, M_j) \end{array} \right\}$$

$$I(i,j) = e(I_j, \mathcal{S}[i]) + \max \left\{ \begin{array}{l} M(i-1, j) + tr(M_j, I_j) \\ I(i-1, j) + tr(I_j, I_j) \end{array} \right\}$$

$$D(i,j) = \max \left\{ \begin{array}{l} M(i, j-1) + tr(M_{j-1}, D_j) \\ D(i, j-1) + tr(D_{j-1}, D_j) \end{array} \right\}$$

$$E(i) = \max\{M(i,j) + tr(M_j, E)\} \ (j = 0, \ldots, L_m - 1)$$

$$J(i) = \max \left\{ \begin{array}{l} J(i-1) + tr(J, J) \\ E(i) + tr(E, J) \end{array} \right\}$$

$$C(i) = \max \left\{ \begin{array}{l} C(i-1) + tr(C, C) \\ E(i) \end{array} \right\}$$

$$T(\mathcal{S}, \mathcal{M}) = C(N) + tr(C, T)$$

Fig. 2. The Viterbi algorithm

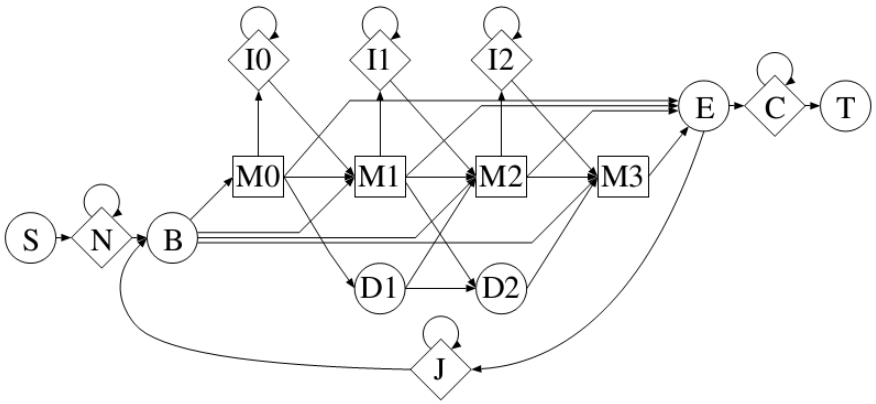


Fig. 1. Plan7