GPU Multiple Sequence Aligner

Fourier-Space Cross-Correlation Alignment

April 29th 2013

Matthias A. Lee

Cross-Correlation Alignment

- Overview
  - Introduction
  - Cross-Correlation (Time-domain)
  - Cross-Correlation (Feq-domain)
  - DNA to Complex Time-Series
  - Alignment Results
  - Performance Results

Cross-Correlation Alignment

- Introduction
  - Common methods for alignment
    - Smith-Waterman
    - Needleman-Wunsch
    - Suffix trees
    - Many more..
  - Cross-Correlation
    - Well-studied Signal Processing method
    - Works well with Long sequences
    - Regular and Complementary matching in one

Signal Processing basics

- Cross-Correlation (time-domain)
  - Compute similarity between 2 time-series
    - Also yields time-delay/phase-shift
  - *O(n^2)* efficiency

Signal Processing basics

- Cross-Correlation (time-domain)
  - Compute similarity between 2 time-series
    - Also yields time-delay/phase-shift
  - *O(n^2)* efficiency

Signal Processing basics

- Cross-Correlation (time-domain)
  - Compute similarity between 2 time-series
    - Also yields time-delay/phase-shift
  - *O(n^2)* efficiency

Signal Processing basics

- Cross-Correlation (time-domain)
  - Compute similarity between 2 time-series
    - Also yields time-delay/phase-shift
  - *O(n^2)* efficiency

Signal Processing basics

- Cross-Correlation (time-domain)
  - Compute similarity between 2 time-series
    - Also yields time-delay/phase-shift
  - *O(n^2)* efficiency

Signal Processing basics

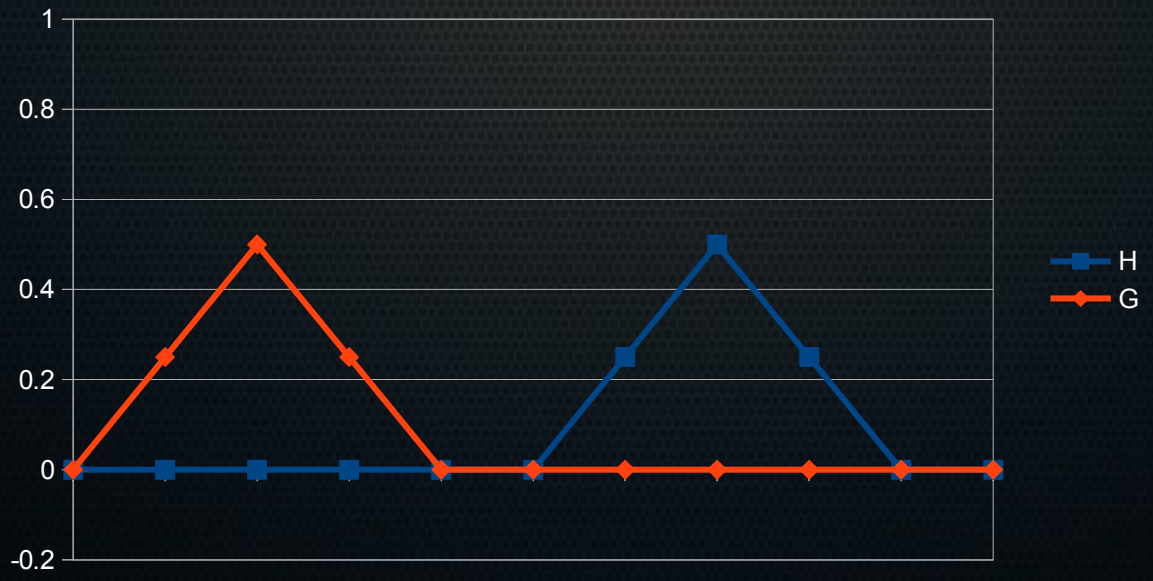
- Cross-Correlation (time-domain)
  - Compute similarity between 2 time-series
    - Also yields time-delay/phase-shift
  - *O(n^2)* efficiency

Signal Processing basics

- Cross-Correlation (time-domain)
  - Compute similarity between 2 time-series
    - Also yields time-delay/phase-shift
  - *O(n^2)* efficiency

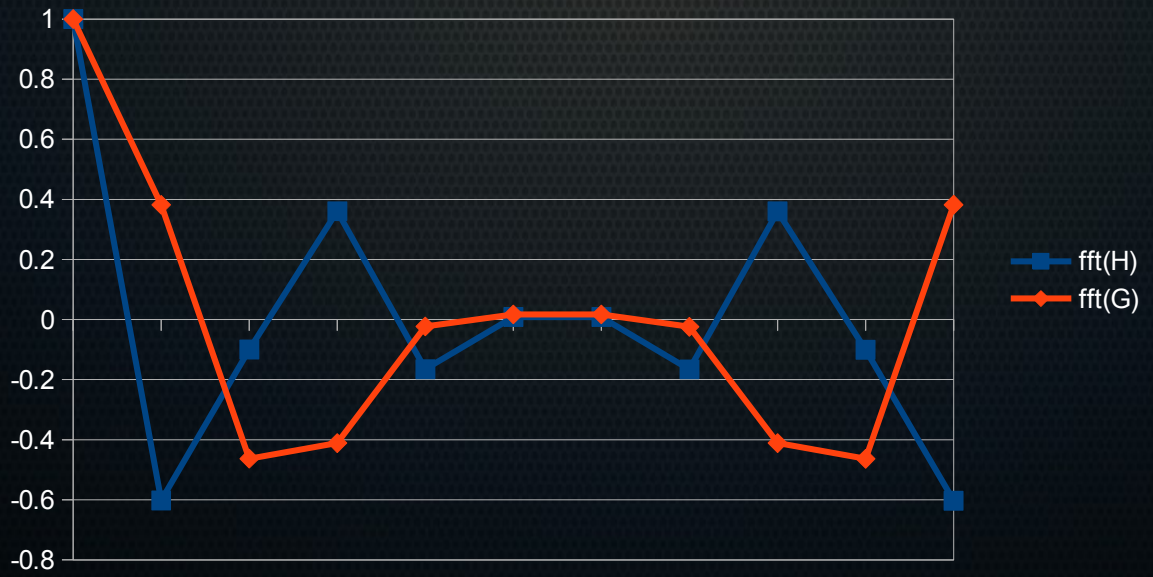
Signal Processing basics

- Cross-Correlation (Fourier-Space)
  - Compute similarity between 2 time-series
    - Also yields time-delay/phase-shift
  - *O(n log n)* efficiency when using FFT



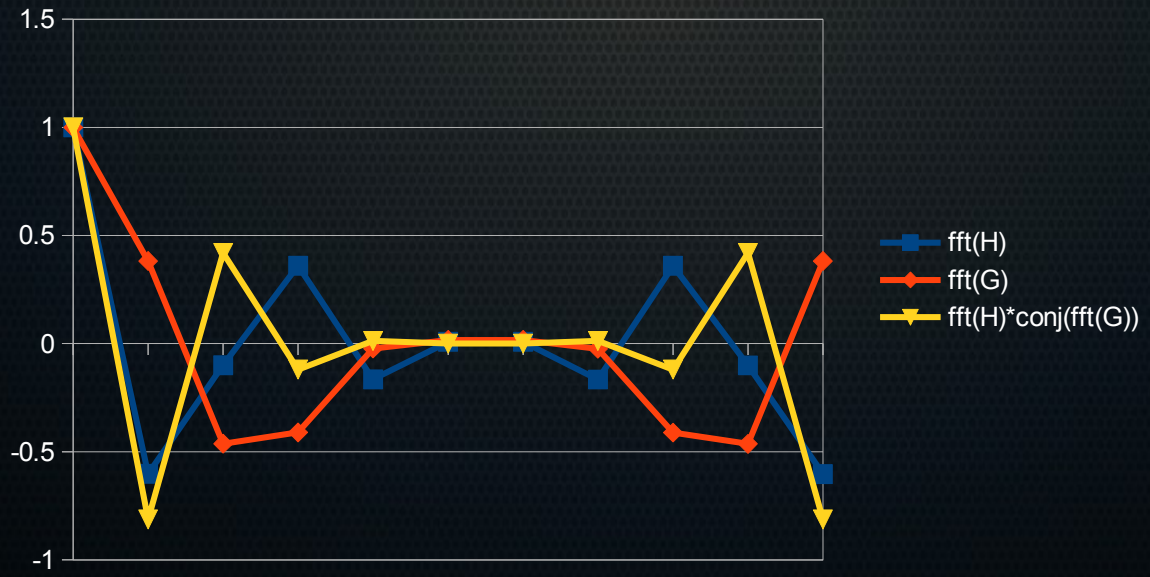
Signal Processing basics

- Correlation Theorem:
  - Corr(G,H) <==> FFT(G) • FFT*(H)
  - Both G & H must be of length n



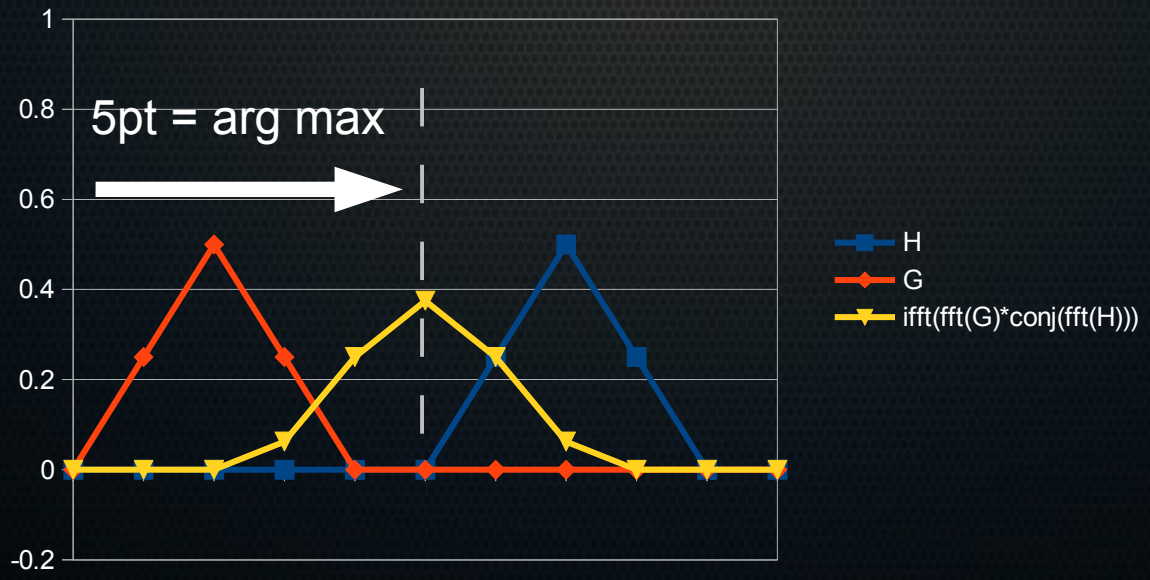
Signal Processing basics

- Correlation Theorem:
  - Sliding dot-product of FFT(G) and complex Conj(FFT(H))
  - F = FFT(G) • FFT*(H)



Signal Processing basics

- Correlation Theorem:
  - Correlation = max (FFT(G) • FFT*(H))
  - Shift = arg max (FFT(G) • FFT*(H))



Time-Series vs DNA/RNA

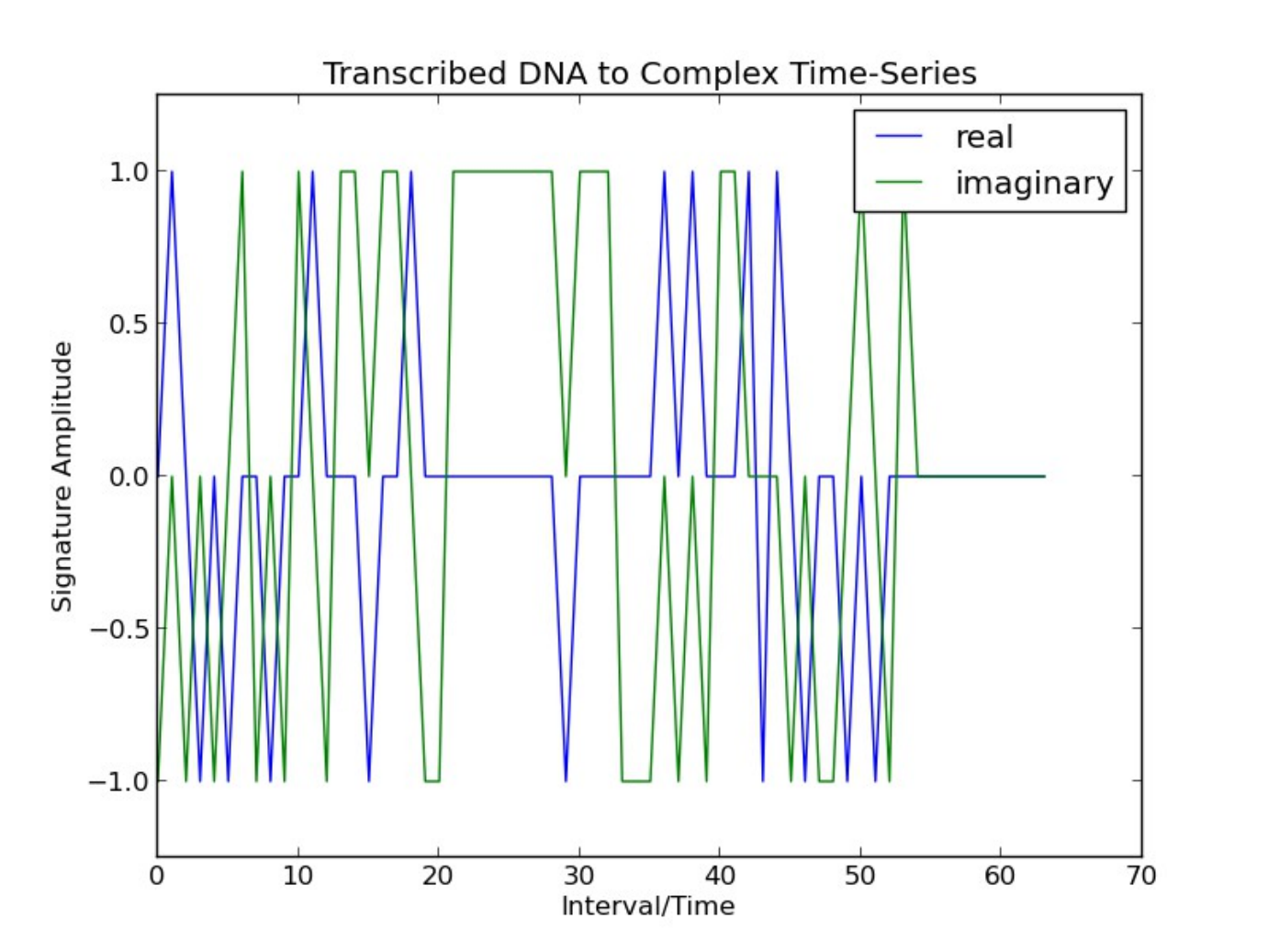
- Transcription to numeric time-series
  - How do we translate to numeric values?

Nucleotide	Transcription
A	1
U/T	-1
C	*i*
G	*-i*

  - Ex: AACGTGT => [1, 1, *-i*, *i*, -1, *i*, -1]

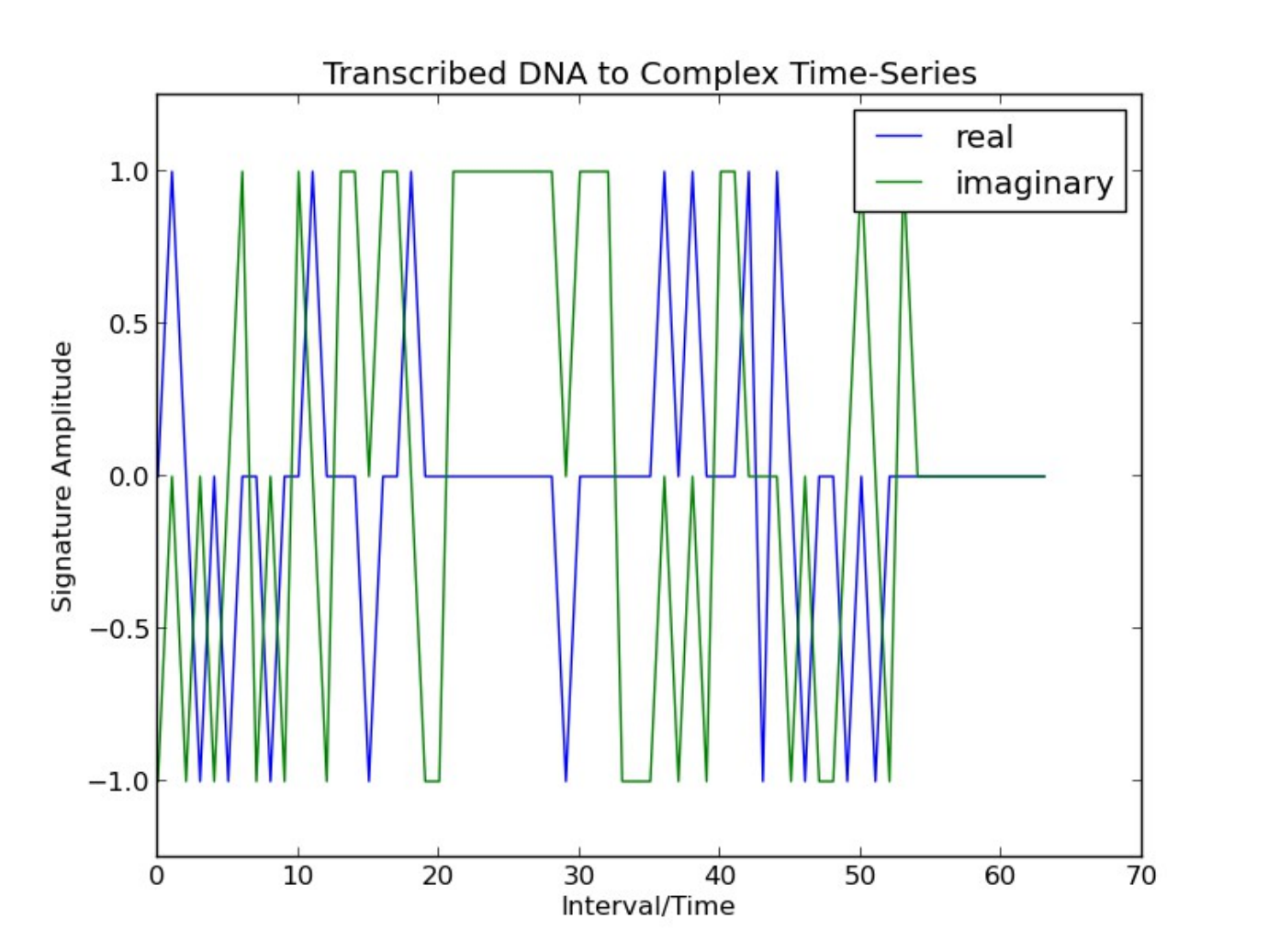
Time-Series Transcription

- gagugucgugcagccuccaggccccccccucccgggagagccauaguggucugc



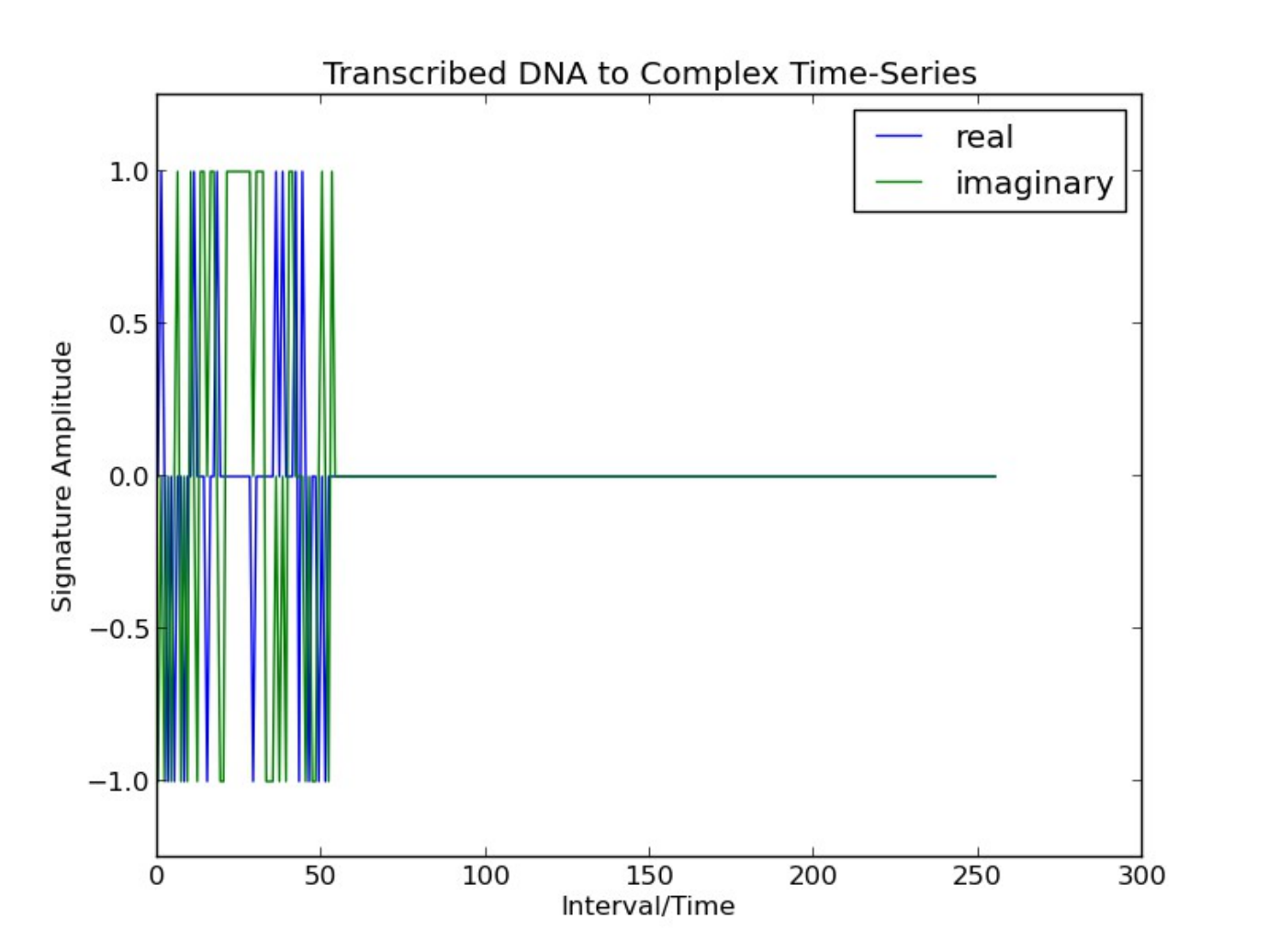
Time-Series Transcription

- len(G) = 54



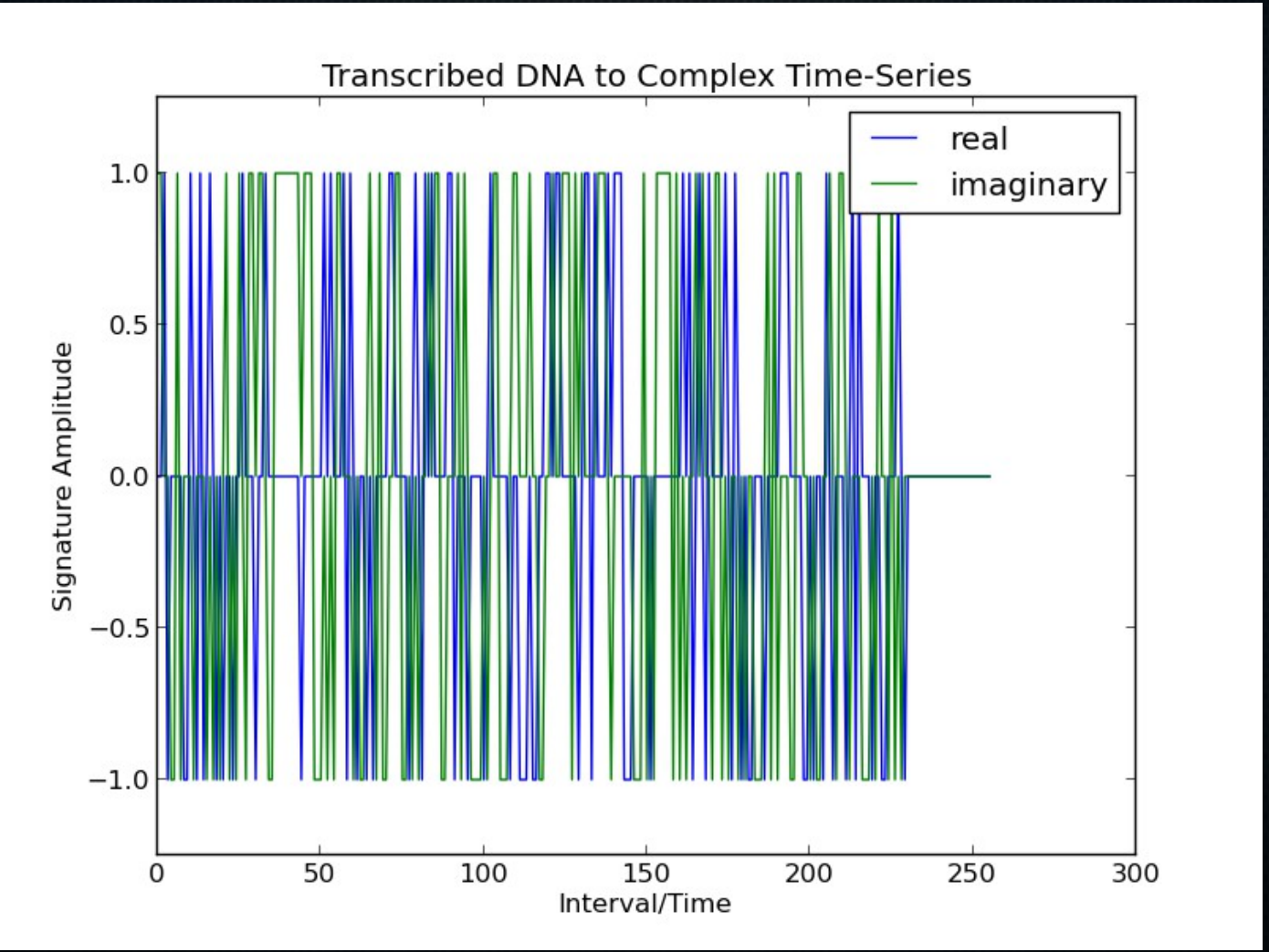
Time-Series Transcription

- len(G) = 54, must pad to size of H with zeros



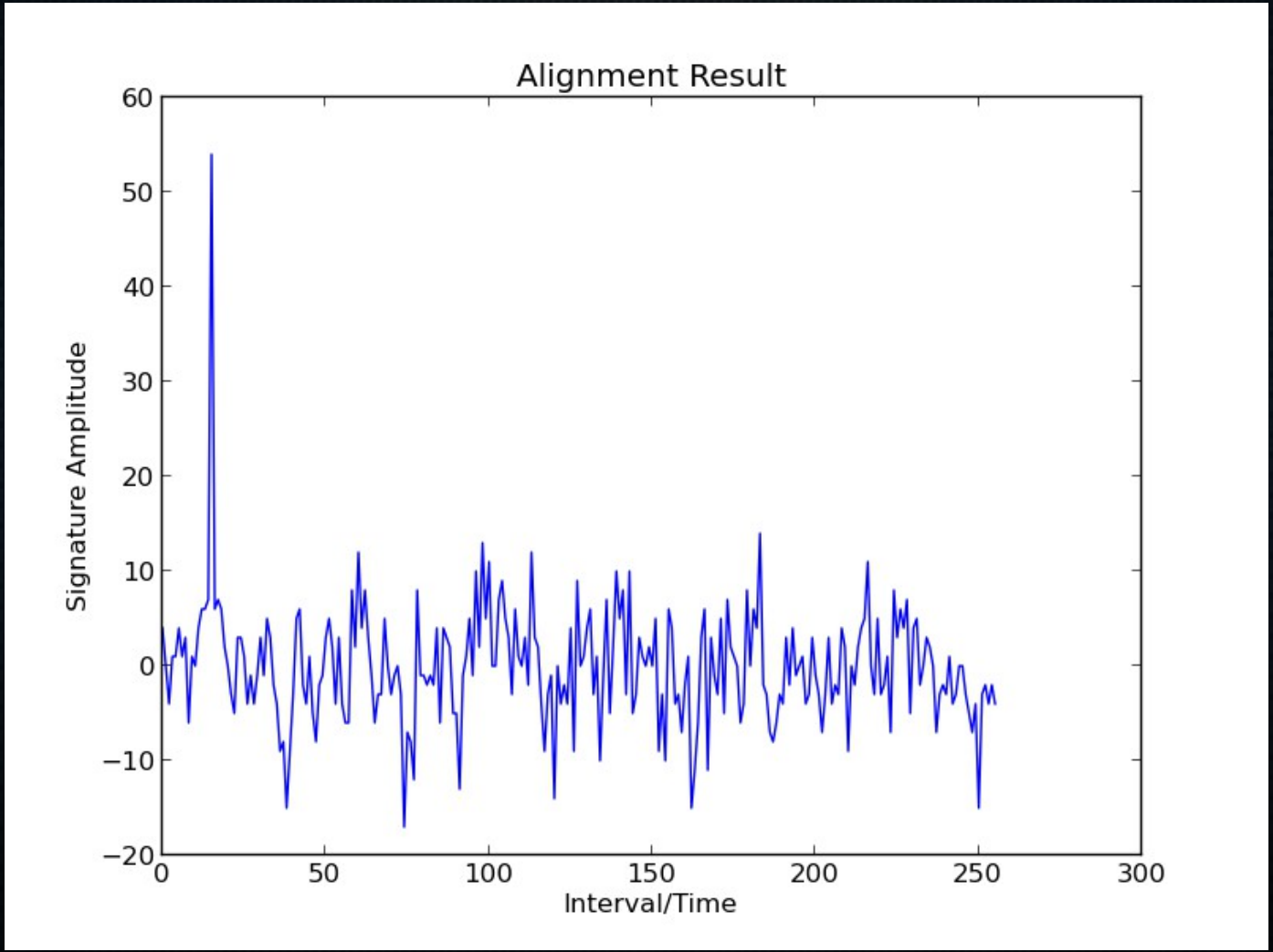
Time-Series Transcription

- len(H)



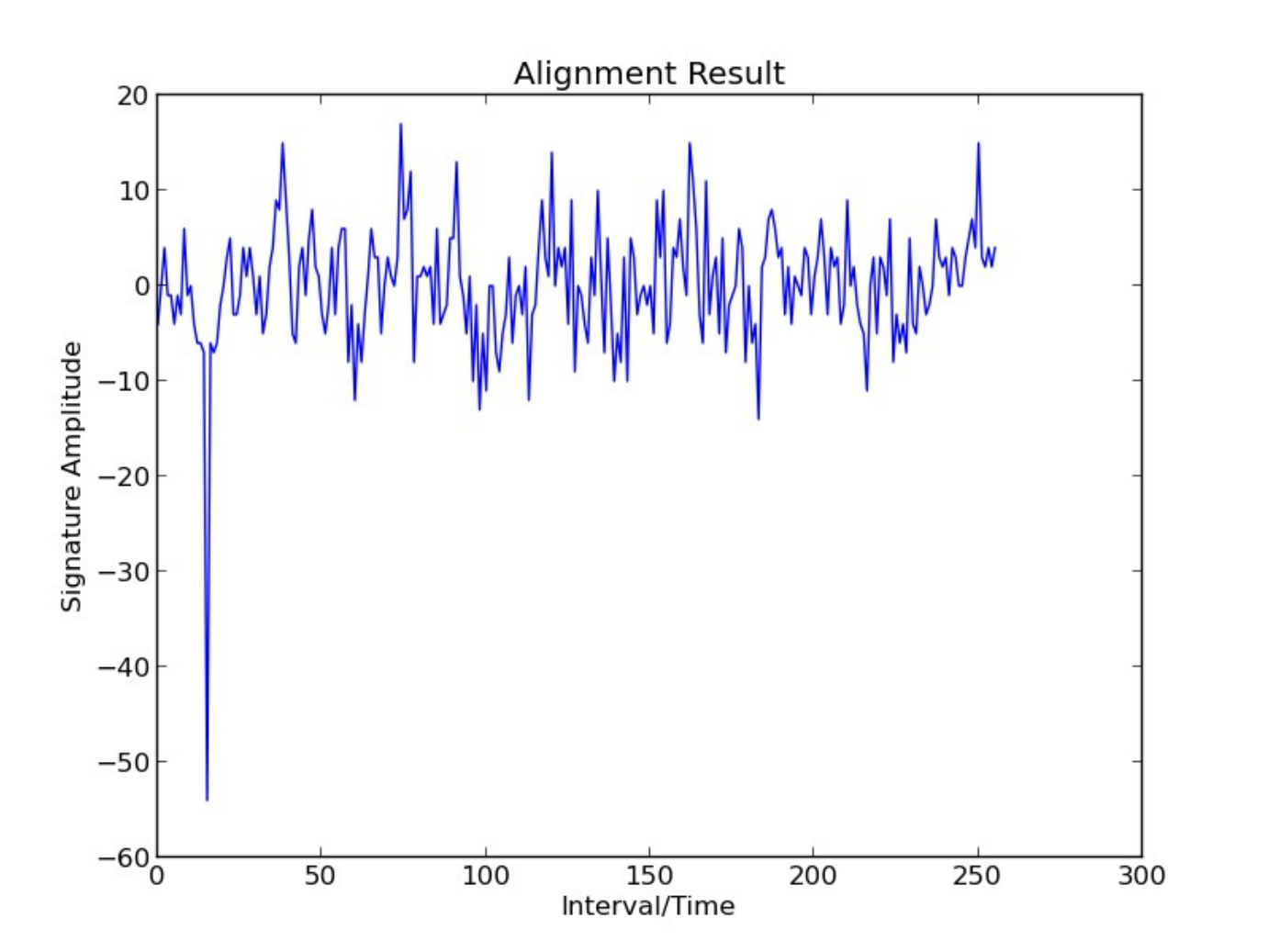
Alignment Calculation

- Alignment result, perfect match



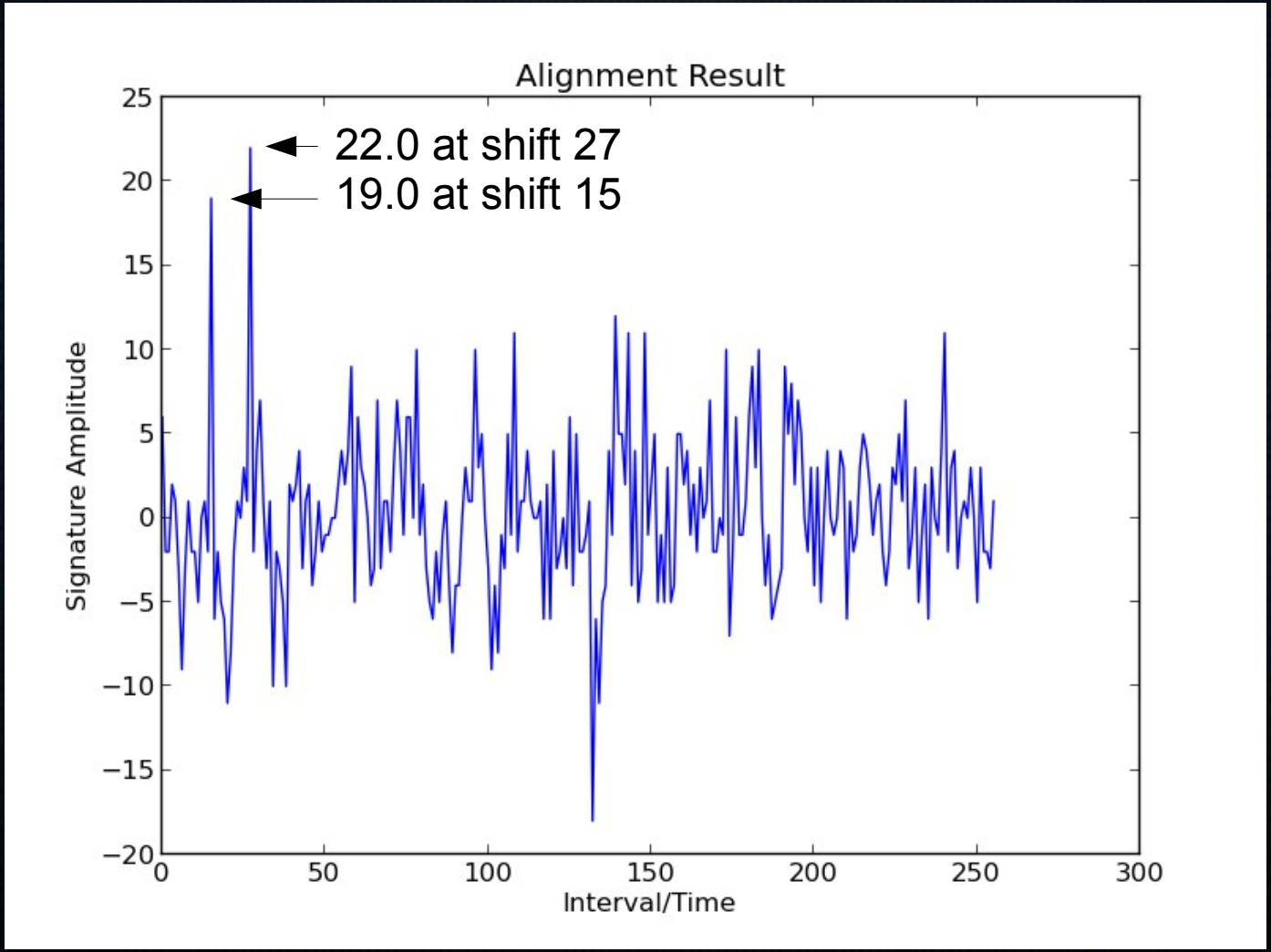
Alignment Calculation

- Alignment result, perfect “complementary” match



Alignment Calculation

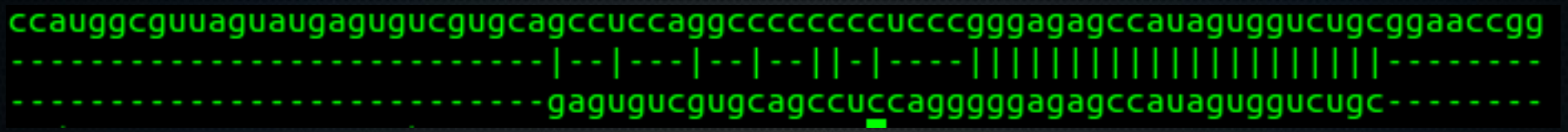
- Non-exact matches, “Split Peaks”



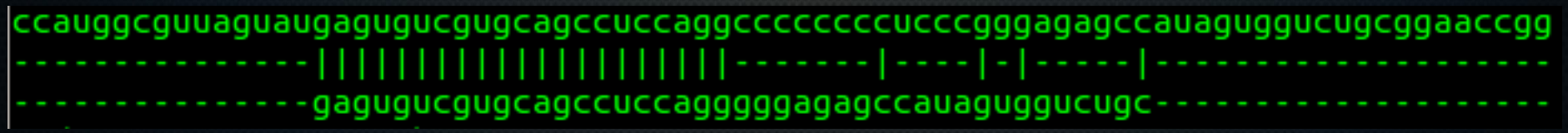
Alignment Calculation

- Non-exact matches, “Split Peaks”

22.0 at shift 27:



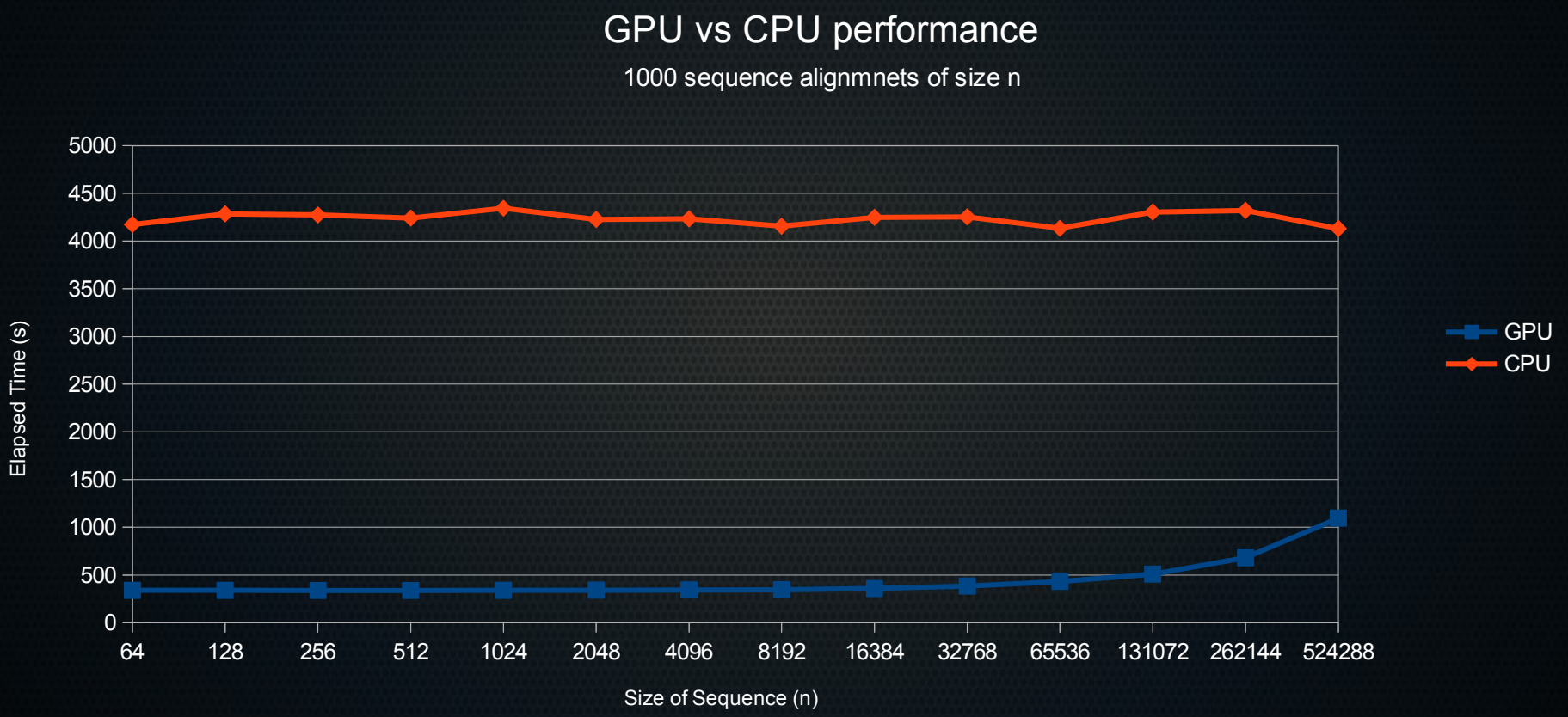
19.0 at shift 15



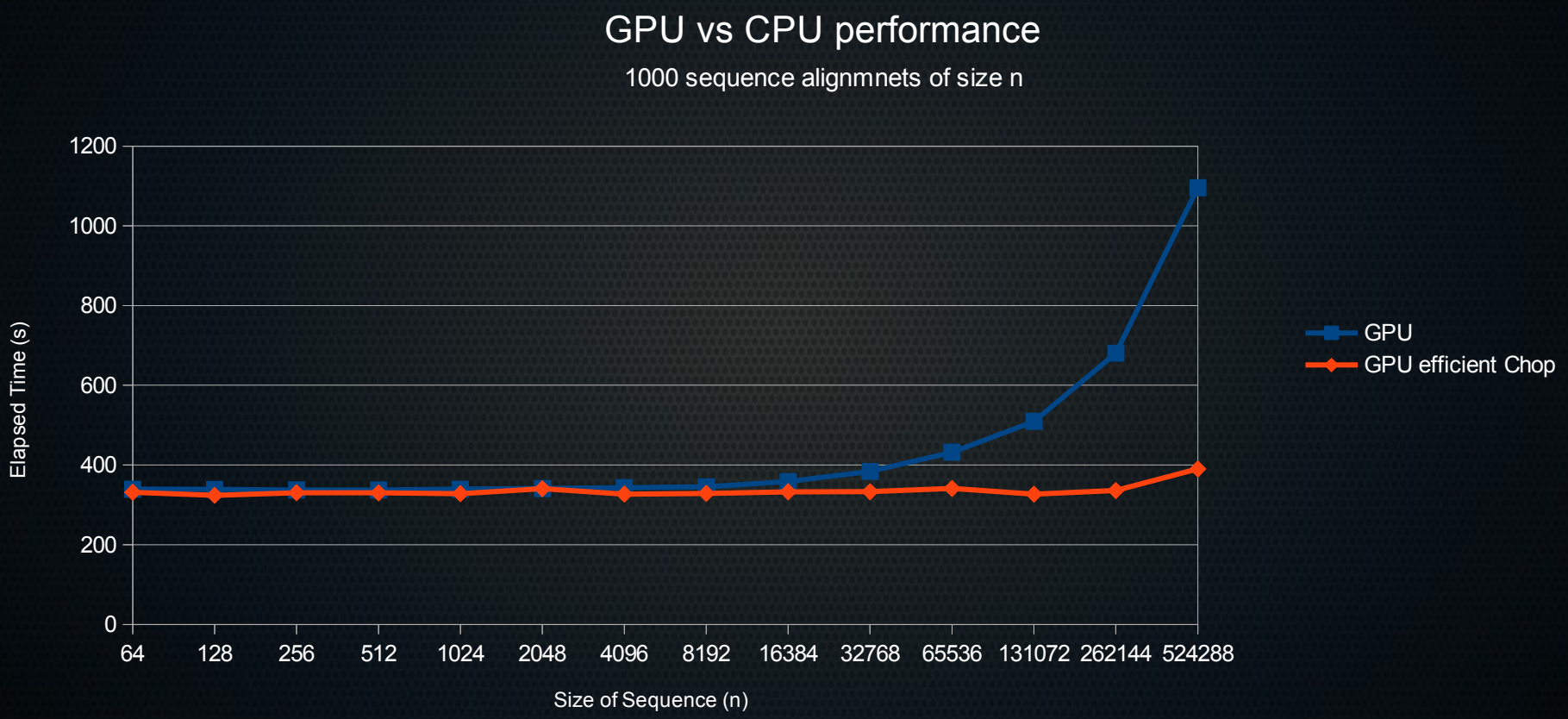
Why use GPUs?

- Parallelism. Parallelism. Parallelism.
- FFT can be parallelized well
- Cross-Correlations are large sets of fully independent calculations.
- When you have 2496 cores... many things look better parallel.
- Test system:
  - Dual Quad Xeon(2.3Ghz), 30GB DDR3 1333
  - Tesla K20c 5GB GDDR5, 2496 cores @ 706MHz

Alignment Performance



Alignment Performance



Possible improvements

- Use GPU “streams” to line up work
- Implement CPU code in C/C++
- Also return Complementary matches
- Implement more post processing
  - Combine partial matches

Thanks

- Questions?

- Github: https://github.com/madmaze/gpuFFTMSA