Matrix Profile XIV: Scaling Time Series Motif Discovery with GPUs to Break a Quintillion Pairwise Comparisons a Day and Beyond

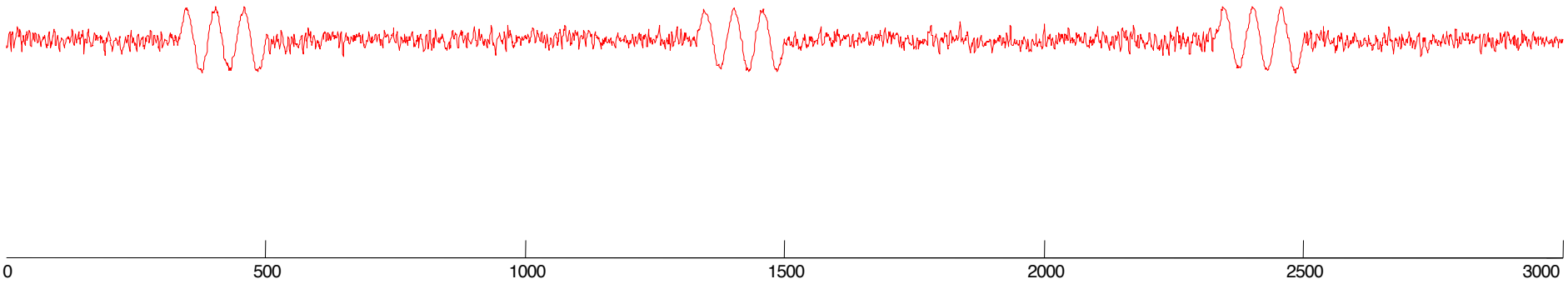
Zachary Zimmerman, Kaveh Kamgar, Nader Shakibay Senobari, Yan Zhu, Brian Crites, Gareth Funning, Philip Brisk, Eamonn Keogh

UC Riverside

Contents

What is the Matrix Profile?

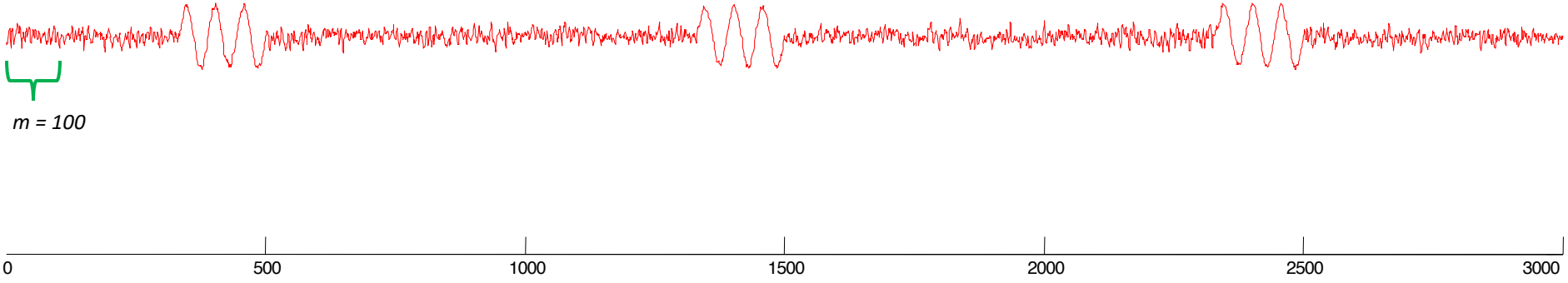
Assume we have a time series *T*, lets start with a synthetic one...



$|T| = n =$
$3{,}000$

Note that for many time series data mining tasks, we are not interested in any *global* properties of the time series, we are only interested in small *local* subsequences, of this length, *m*

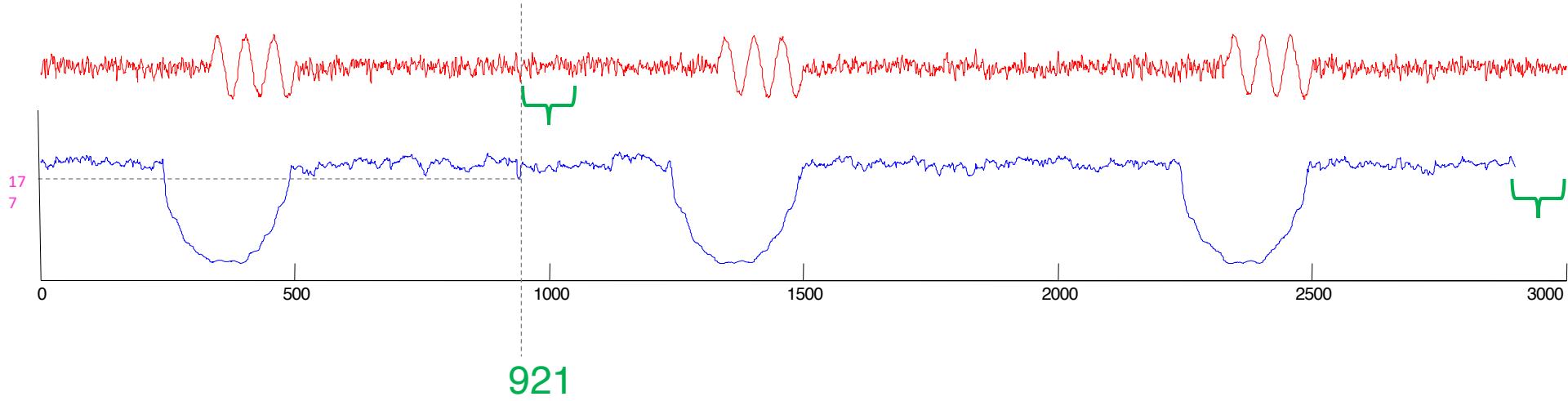
These subsequences might be about the length of individual heartbeats (for ECGs), individual days (for social media behavior), individual words (for speech analysis) etc



We can create a companion “time series”, called a Matrix Profile or MP.

The matrix profile at the i^{th} location records the distance of the subsequence in T, at the i^{th} location, to its nearest neighbor under z-normalized Euclidean Distance (or Pearson Correlation).

For example, in the below, the subsequence starting at 921 happens to have a distance of 177.0 to its nearest neighbor (wherever it is).

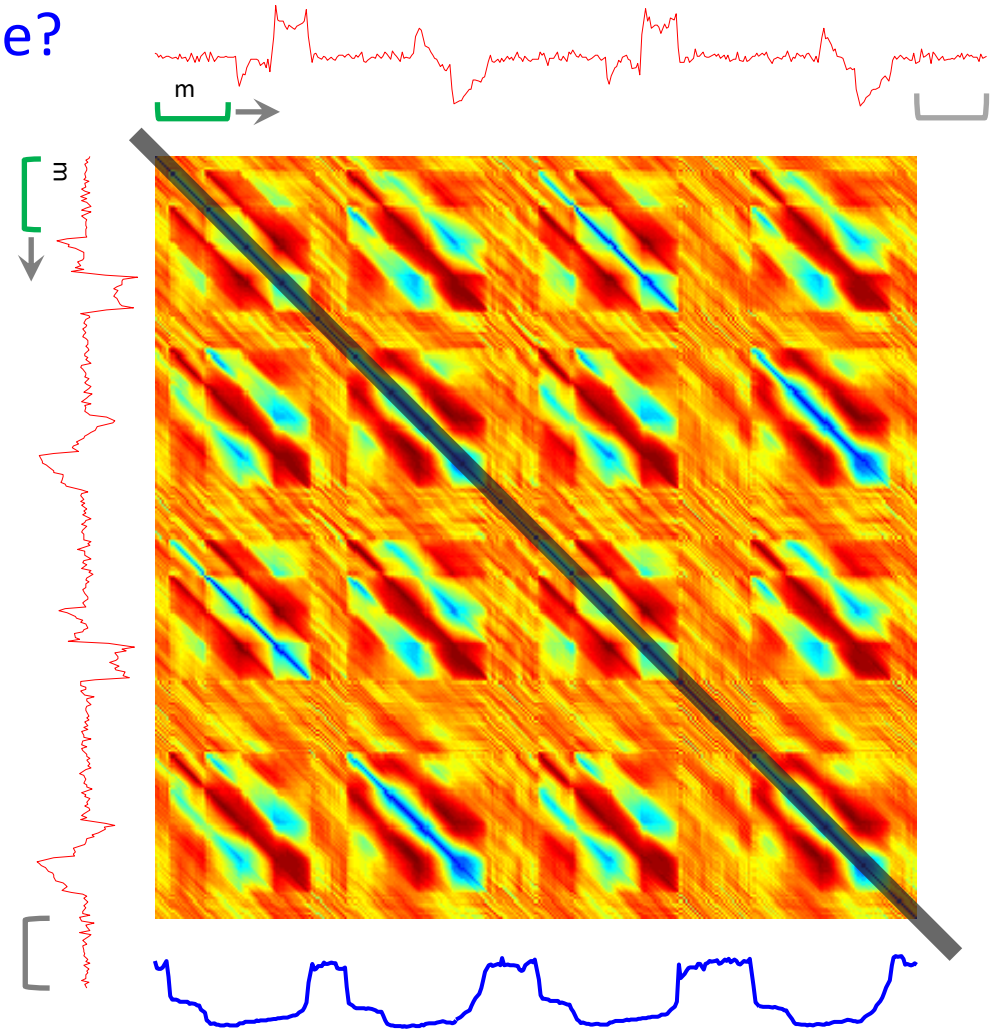


Why is it called the Matrix Profile?

One naïve way to compute it would be to construct a distance matrix of all pairs of subsequences of length *m*.

For each column, we could then "project" down the smallest (*non diagonal*) value to a vector, and that vector would be the Matrix Profile.

While in general we could never afford the memory to do this (4TB for just |T|= one million), for most applications the Matrix Profile is the *only* thing we need from the full matrix, and we can compute and store it very efficiently. (as we will see later)

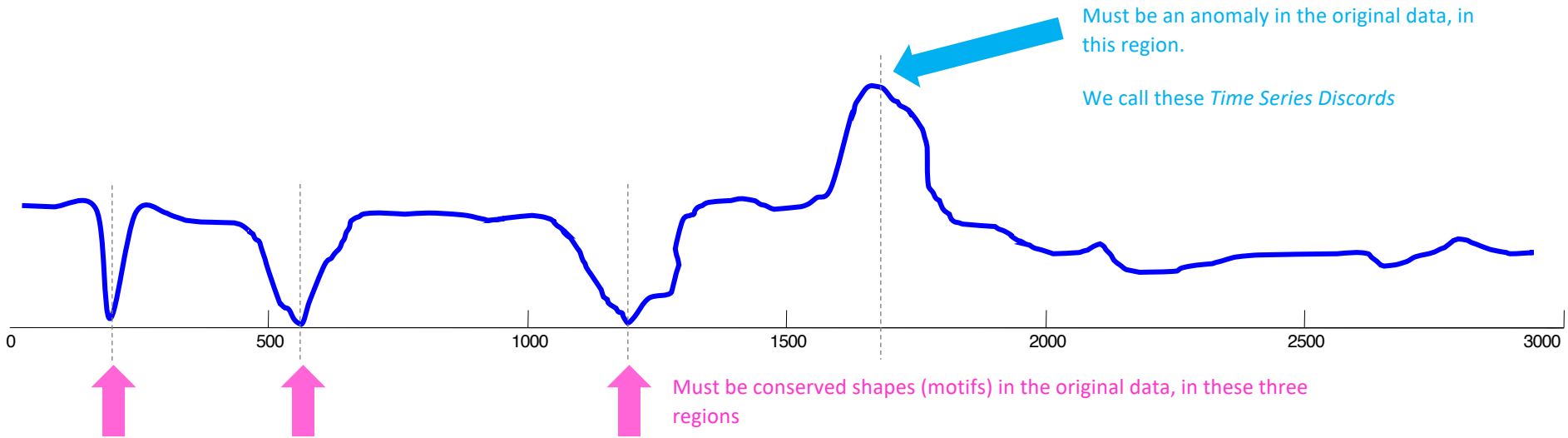


Key:
Small distances are blue
Large distances are red
Dark stripe is excluded

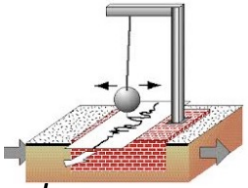
How to “read” a Matrix Profile

Where you see relatively low values, you know that the subsequence in the original time series must have (at least one) relatively similar subsequence elsewhere in the data (such regions are “motifs” or reoccurring patterns)

Where you see relatively high values, you know that the subsequence in the original time series must be unique in its shape (such areas are “discords” or anomalies).

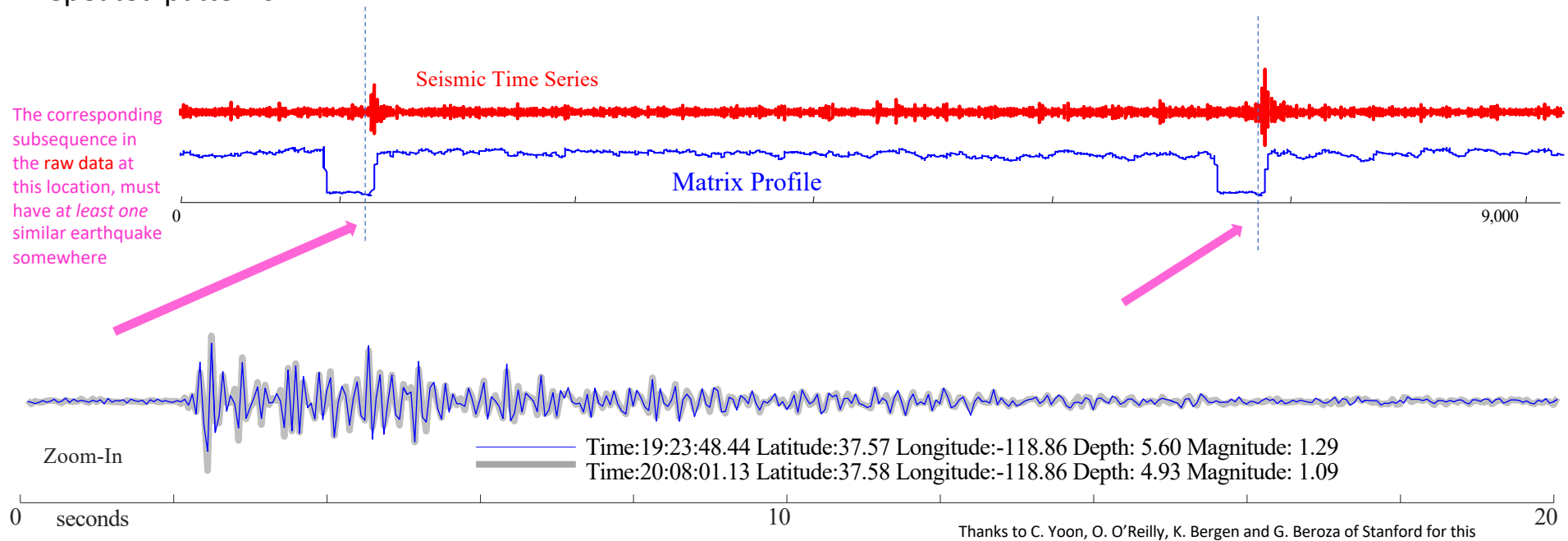


Seismology Example

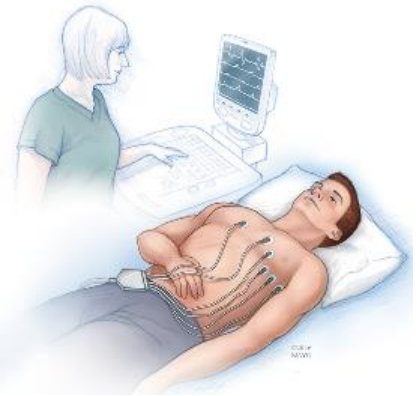


If we see low values in the MP of a seismograph, it means there must have been a *repeated earthquake*. Repeated earthquakes can happen *decades* apart.

Many fundamental problems seismology, including the discovery of foreshocks, aftershocks, triggered earthquakes, swarms, volcanic activity and induced seismicity, can be reduced to the discovery of these repeated patterns.



Thanks to C. Yoon, O. O'Reilly, K. Bergen and G. Beroza of Stanford for this

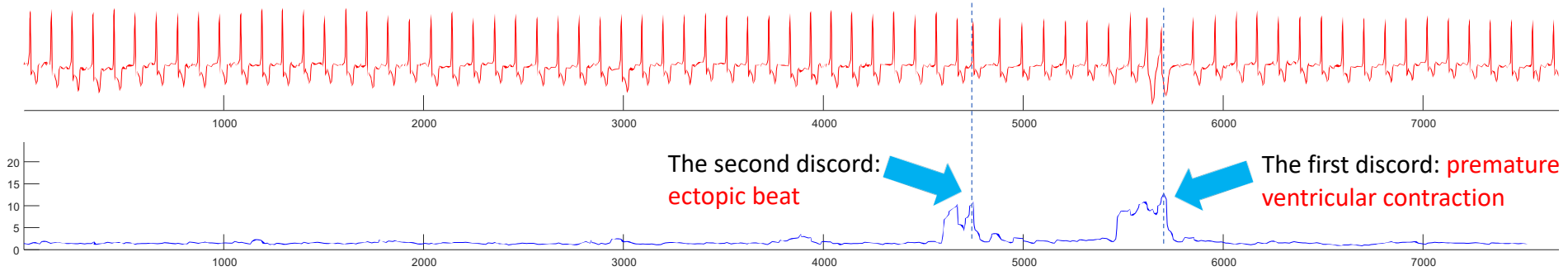


Electrocardiogram Example

(MIT-BIH Long-Term ECG Database)

In this case there are two anomalies annotated by MIT cardiologists. The Matrix Profile clearly indicates them.

Here the subsequence length was set to 150, but we still find these anomalies if we *half* or *triple* that length.



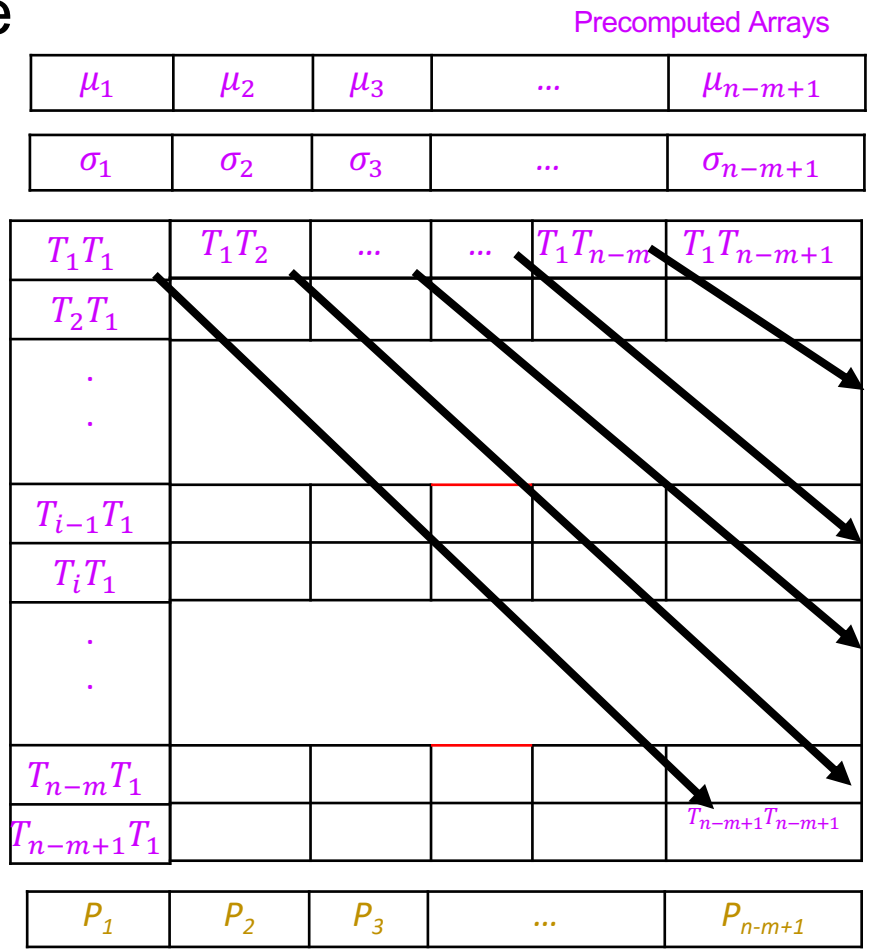
Scaling the Matrix Profile

SCAMP: Scalable Matrix Profile

In the interest of time, I will not get into the algebra and algorithmic details in this talk. In brief, we can exploit the fact that our only dependency is along the diagonal of the distance matrix to speed up the computations.

In the GPU we can assign each thread a set of diagonals and compute the distances along them.

We can use a similar strategy to improve performance on the CPU.



Scaling the Matrix Profile calculation

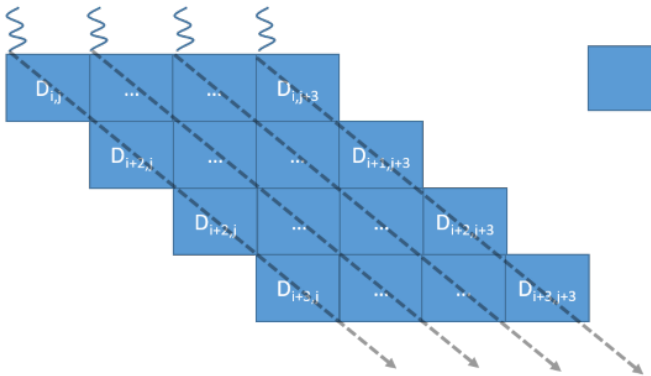
- Performance for Input time series of length 2 million:
  - Initial CPU Implementation: 1 CPU thread -> **4.2 days**
  - Initial GPU Implementation: K80 GPU -> 3.2 hours
  - Optimized CPU implementation: 4 CPU thread -> 6.5 minutes (900x)
  - Optimized GPU Implementation: V100 GPU -> **5 seconds (2300x)**

- Cloud implementation 40 GPU cluster allowed us to do 1 billion in < 10 hours
  - This is on the order of 10^18 (a quintillion) pairwise comparisons
  - COST ~ 500 USD (~ 0.80 USD per quadrillion comparisons)

Scaling the Matrix Profile Calculation

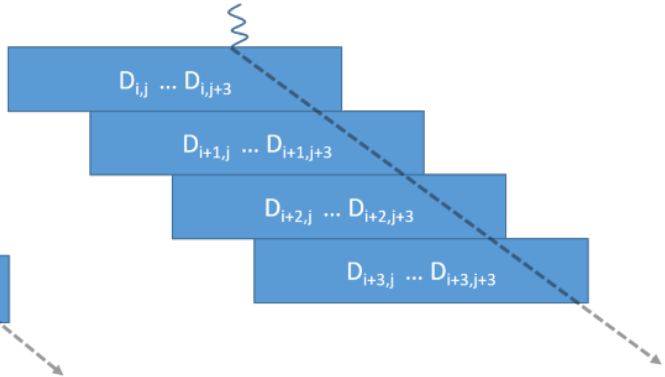
- These speedups came as the result of improvements in the following areas:
  - Better Algorithmic Complexity
  - Use of Modern Hardware
  - Use of Relevant Hardware Features
    - *Intelligent shared memory and register utilization, smart atomic ops…*
  - Architecture Aware Code
    - *Memory Access Patterns, ILP and latency hiding…*
  - Algebraic Improvements to Problem Formulation
    - *Fewer instructions*
    - *Lower Precision is an option*
    - *Cheaper GPUs can be used*

Scaling the Matrix Profile calculation: Architecture Awareness / Feature Utilization (GPU Example)



Old Method:

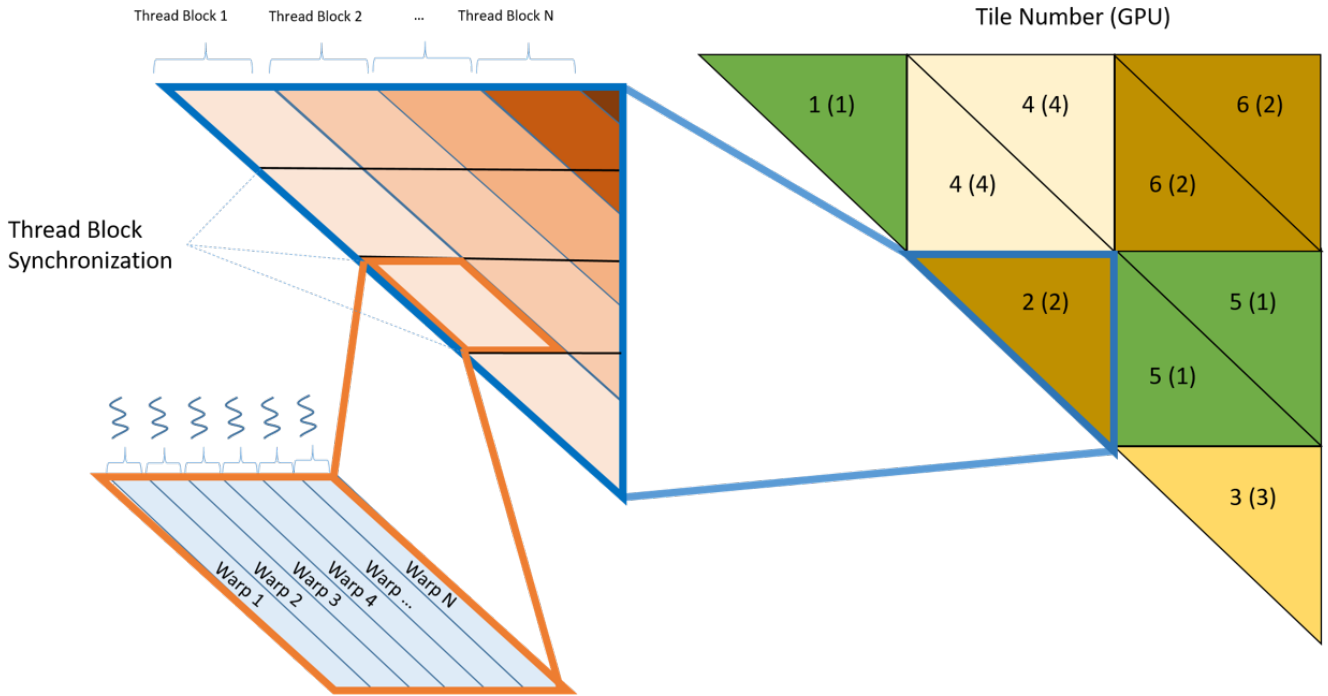
- One load per dependancy per cell
- 32 MP updates: 2 per cell
- Meta-diagonals are shorter



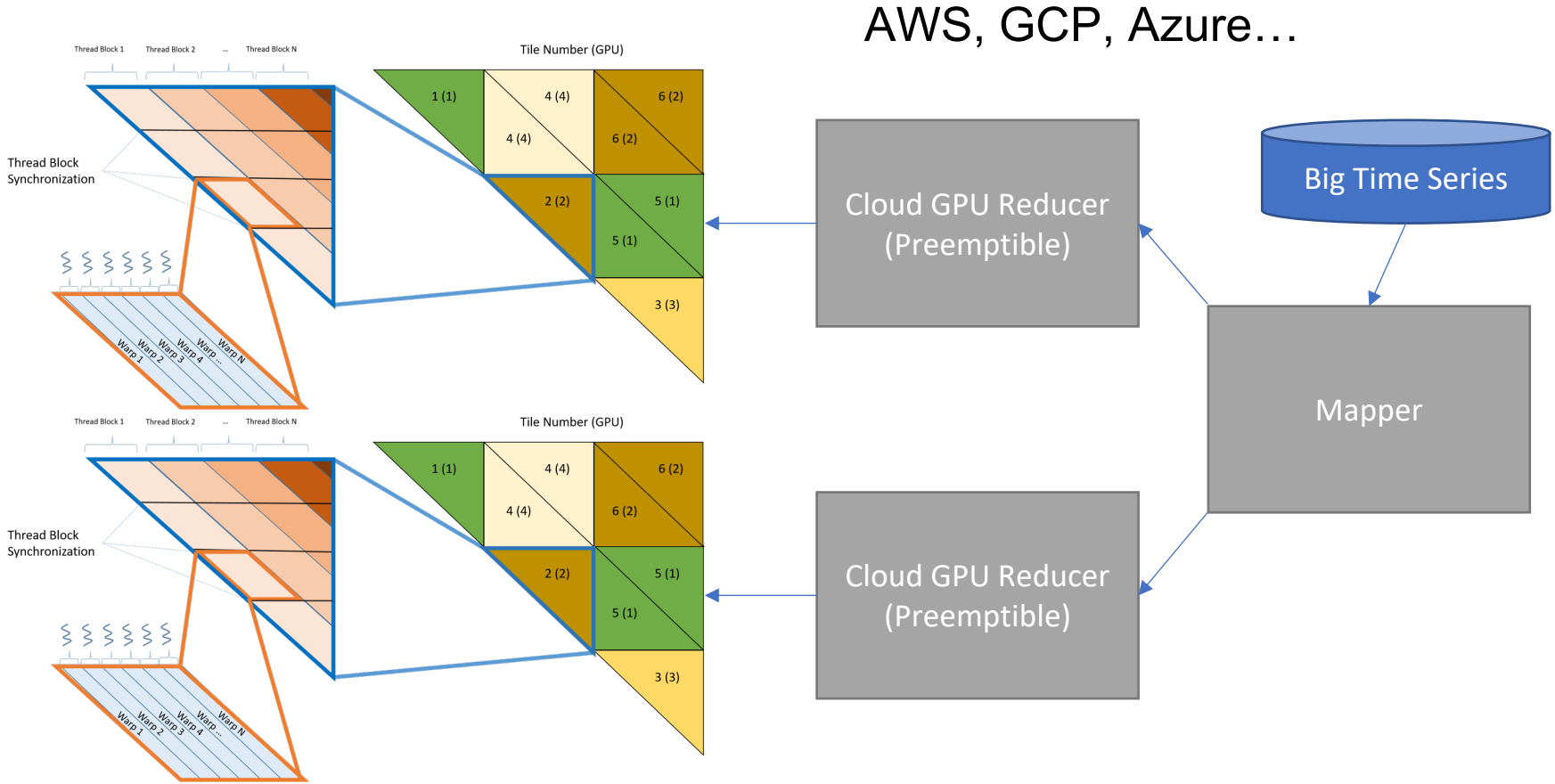
New Method:

- One vectorized load per dependancy per row
- 11 MP updates: 4 rows, 7 columns
- Meta-diagonals are 4x larger -> 4x data reuse

Scaling the Matrix Profile Calculation: Tiling



Scaling the Matrix Profile Calculation: Tiling and Distributed Computation

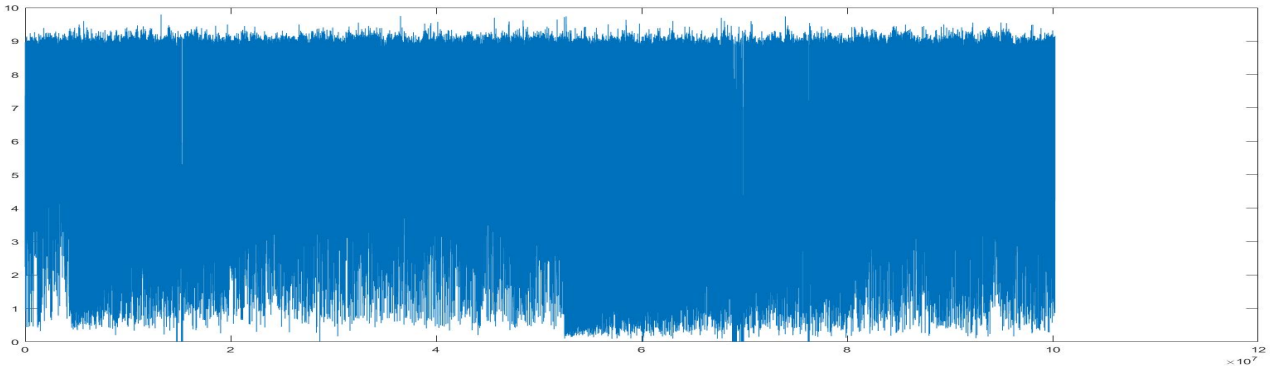


Results

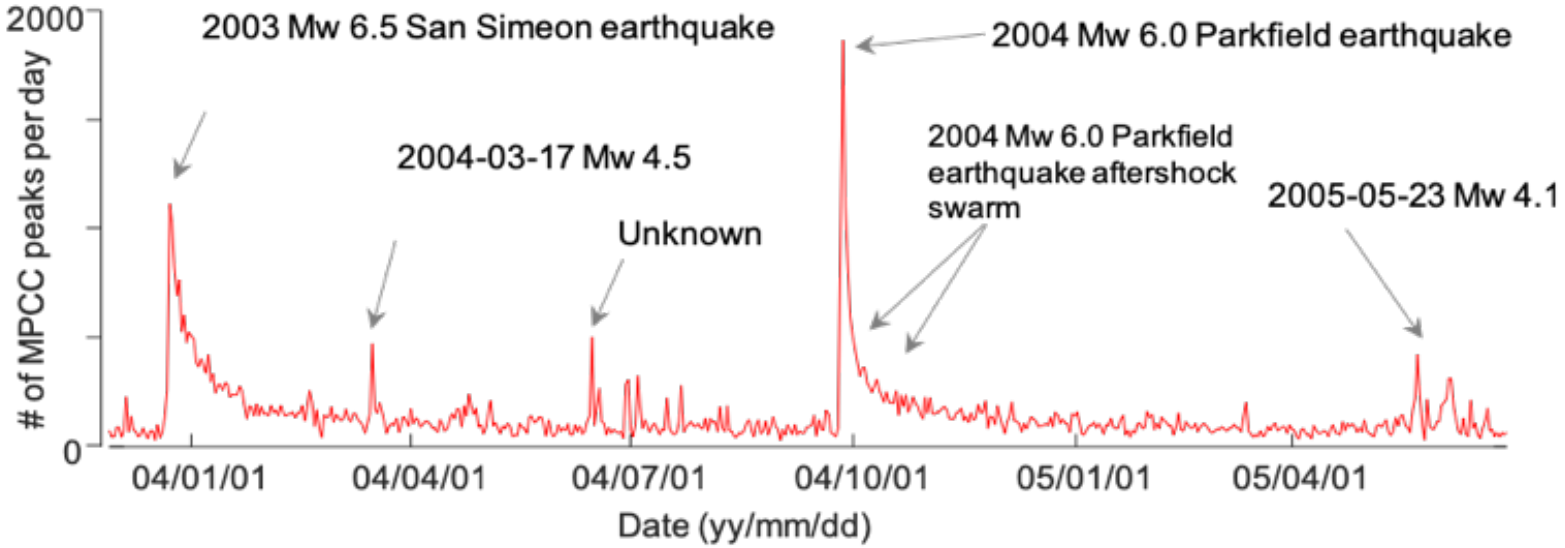
Scaling the Matrix Profile: Results

Dataset	Parkfield 1B	Cascadia Subduction Zone
Size	1 Billion	1 Billion
Total GPU time	375.2 hours	375.3 hours
Spot Job Time	2.5 days	10hours 3min
Approximate Spot Cost	480 USD	620 USD

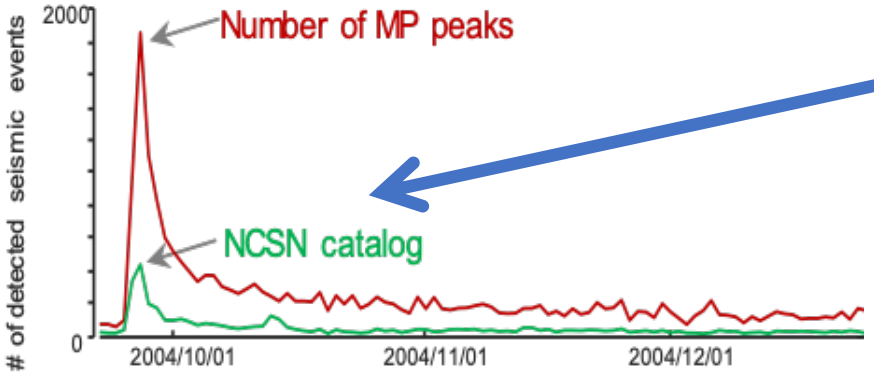
Parkfield 580 days @ 20Hz Matrix Profile



What does SCAMP find?

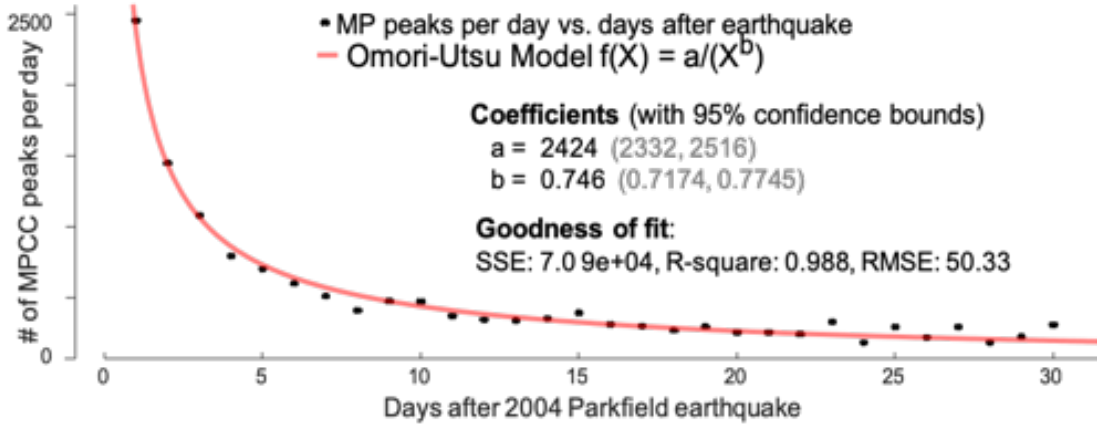


What does SCAMP find?



16x more events detected than are in the seismic catalog

Our findings fit the aftershock rate model for the Parkfield Earthquake



Conclusion

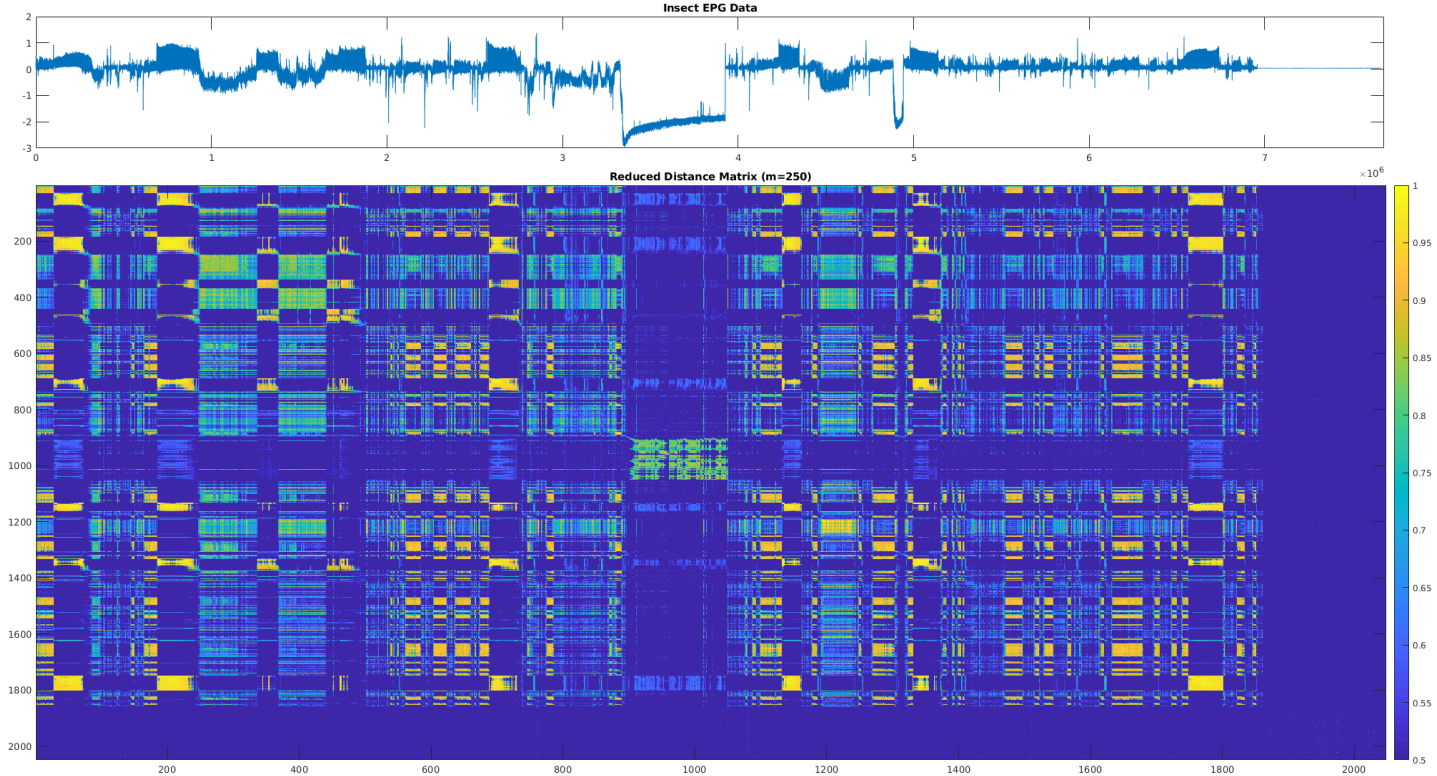
- Introduced the Matrix Profile data structure and gave a preview of its applications.

- Introduced an open-source, scalable framework for computing the Matrix Profile on both CPUs and GPUs, locally and in the cloud.

- Showed that by using the performance of SCAMP we can exactly search huge datasets and uncover new insights.

What’s Next?

- What else can we do with this computational pattern?
  - Frequency of matches?
  - Generate multiple matches?


Insect EPG Data
2
1
0
-1
-2
-3
0
1
2
3
4
5
6
7
Reduced Distance Matrix (m=250)
×10^6
200
400
600
800
1000
1200
1400
1600
1800
2000
1
0.95
0.9
0.85
0.8
0.75
0.7
0.65
0.6
0.55
0.5

Thanks for listening! Questions?

- Supporting Webpage (MP papers can be found here): https://www.cs.ucr.edu/~eamonn/MatrixProfile.html

- SCAMP source code: https://github.com/zpzim/SCAMP