Algorithms and Applications of

Temporal Patterns: Motifs, Shapelets and Discords

Abdullah Mueen

Assistant Professor,
Department of Computer Science,
University of New Mexico, USA

Student Collaborators

Michael Yeh

Yan Zhu

Zach Zimmerman

Amanda Minnich

Hossein Hamooni

Nikan Chavoshi

Adnan I. Khair

What are Time Series?

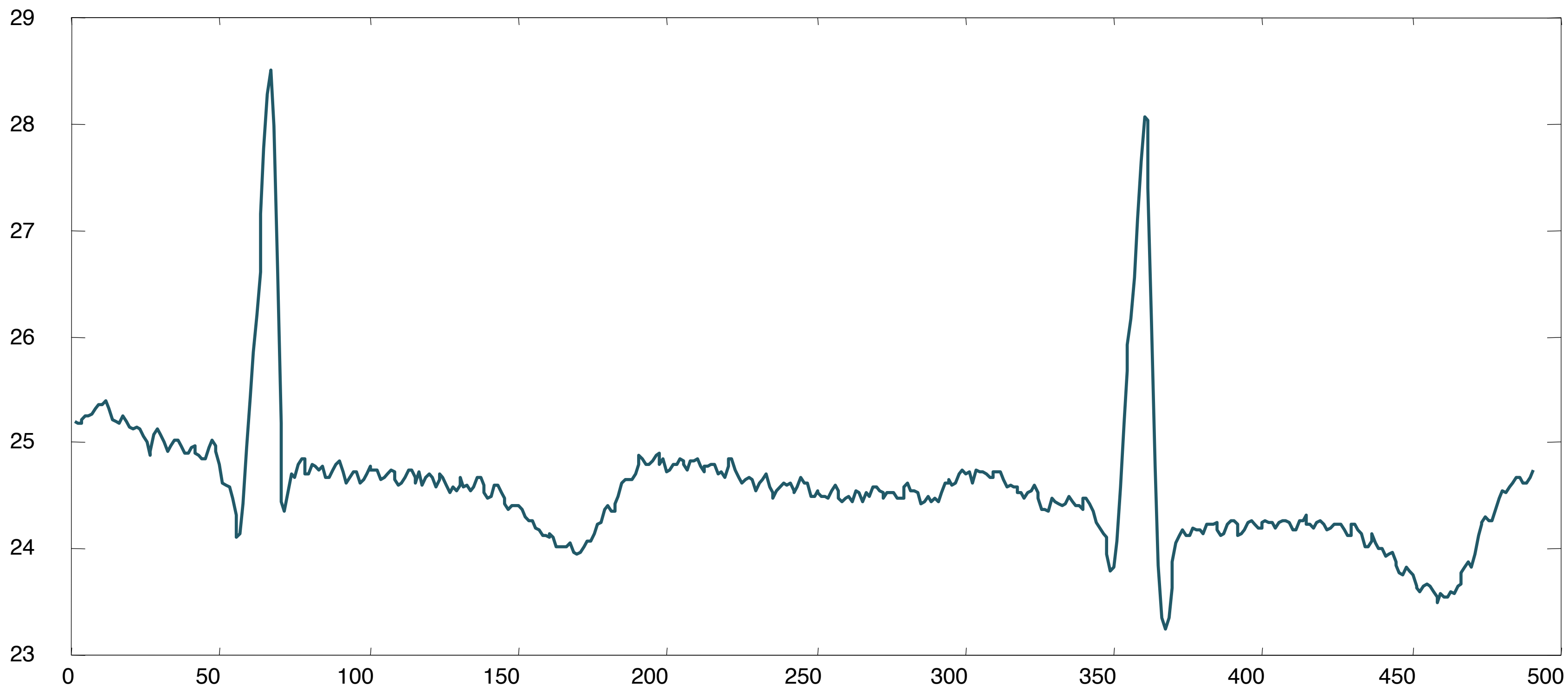
25.1750
25.2250
25.2500
25.2500
25.2750
25.3250
25.3500
25.3500
25.4000
25.4000
25.3250
25.2250
25.2000
25.1750

..

..

24.6250
24.6750
24.6750
24.6250
24.6250
24.6250
24.6750
24.7500

A time series is a collection of observations made sequentially in time.



Approaches

- In Computational Finance

  AR, MA, ARIMA, SARIMA, GARCH, …

  … fit models to the entire time series

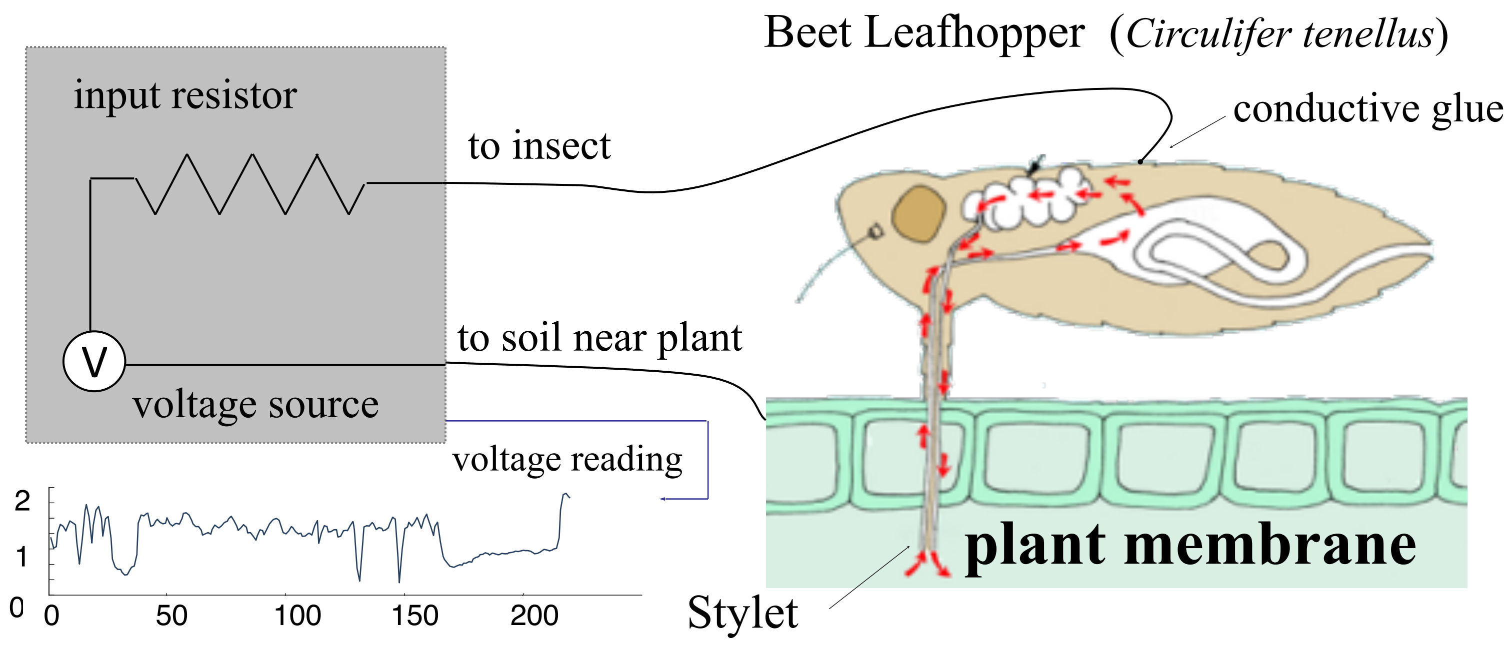
- In Statistical Machine Learning

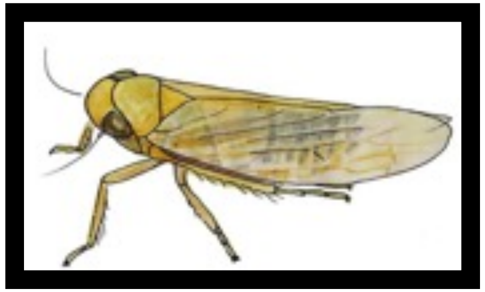
  HMM, GMM, Point Processes, …

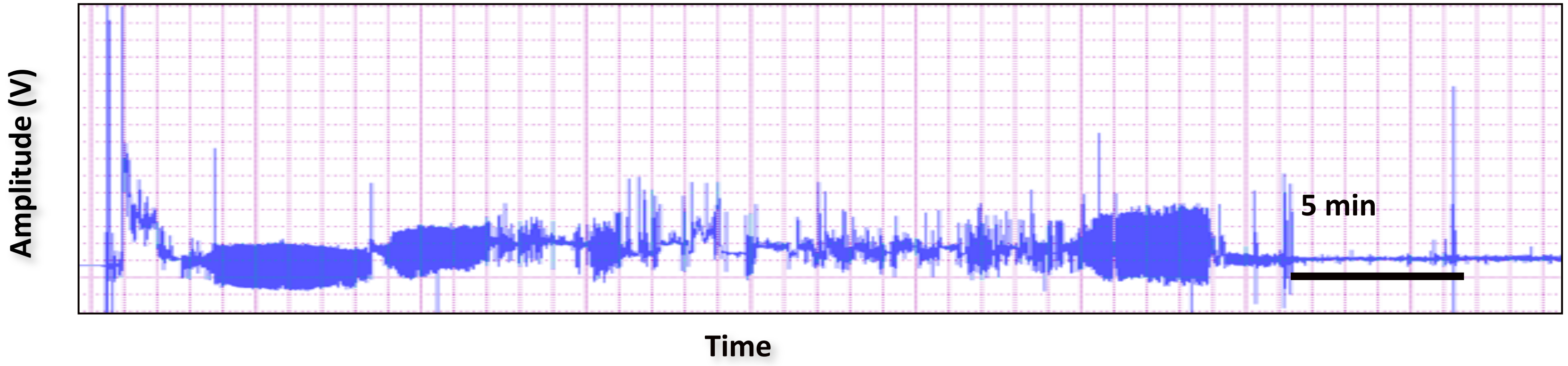
  … treat observations as samples of a random variable

– These models have strong global assumptions

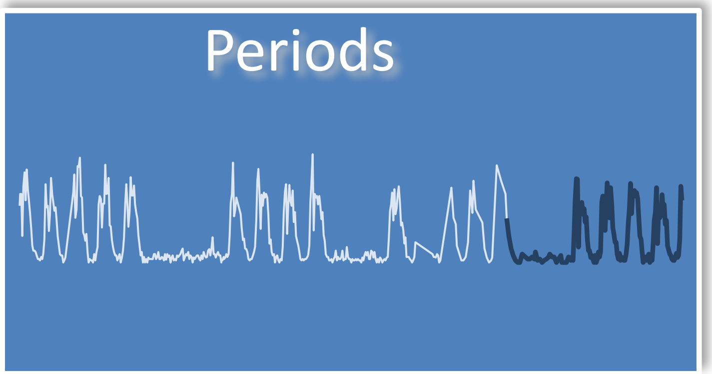
Understanding Time Series

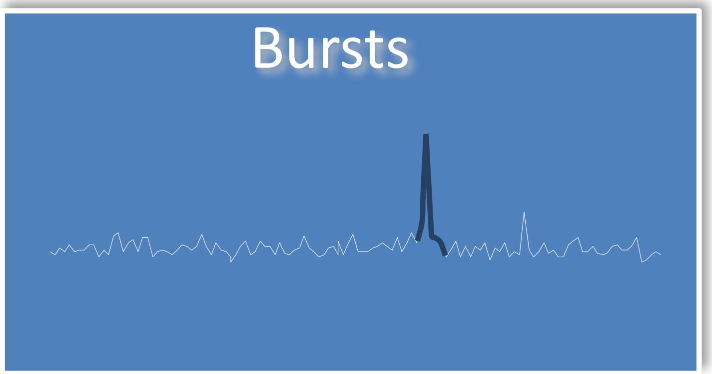


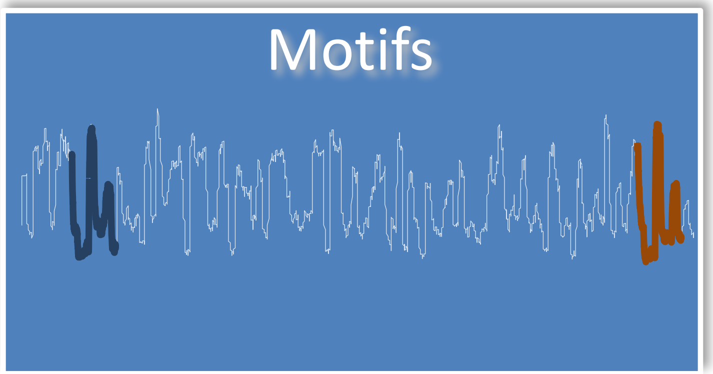


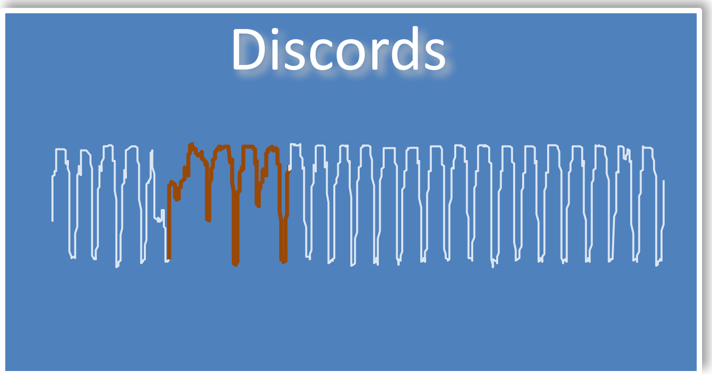


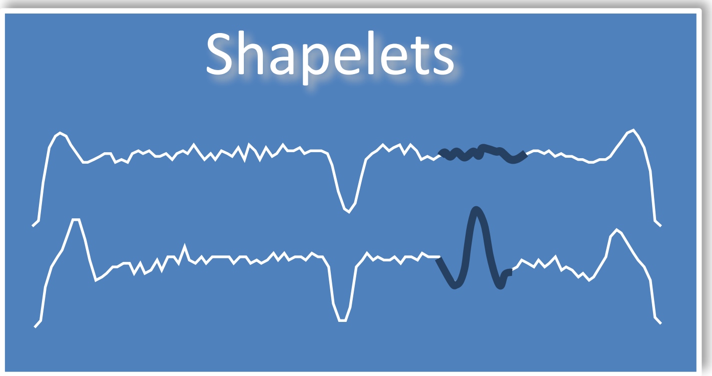
Time Series Patterns











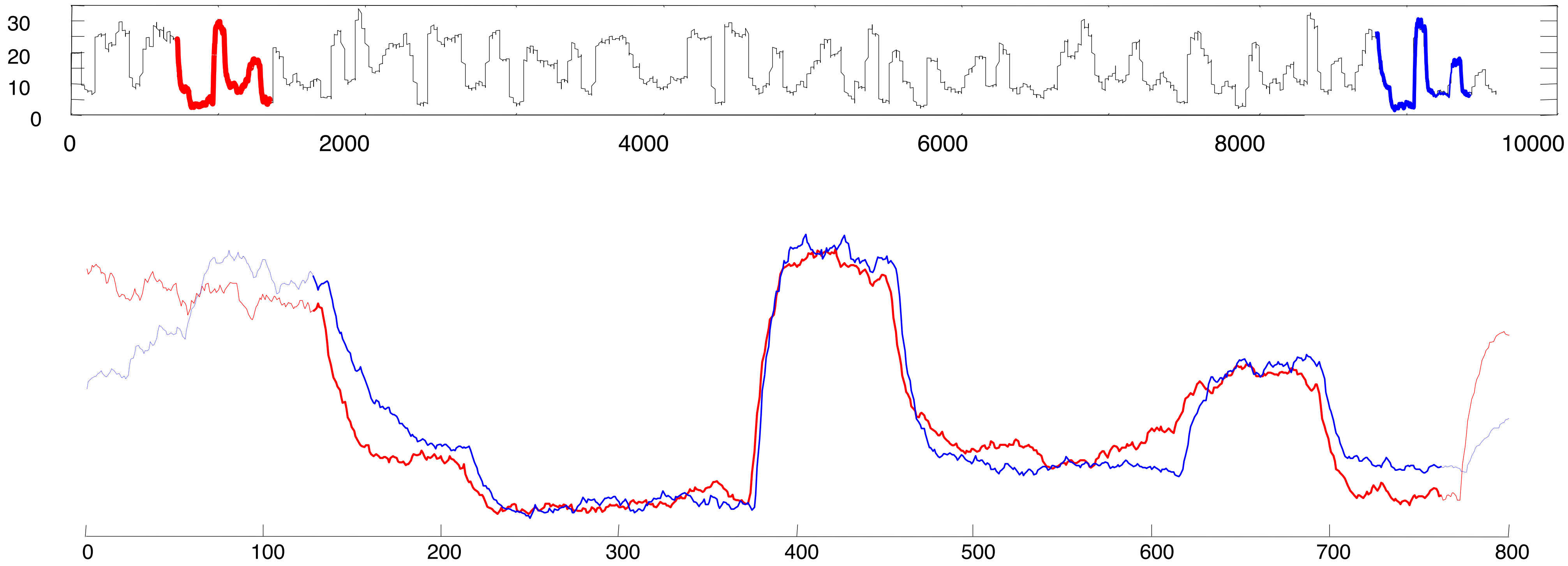
- Intuitive for human
- Good feature for high level learning
- Time series patterns exist in many domains.

Outline

- Motifs
  - Definition
  - Never Ending Learning
  - Bot Detection
- Shapelets
  - Definition
  - Surface Classification
  - Patient Classification
- Discords
  - Definition
  - Star Light Curves
  - Search Frequency

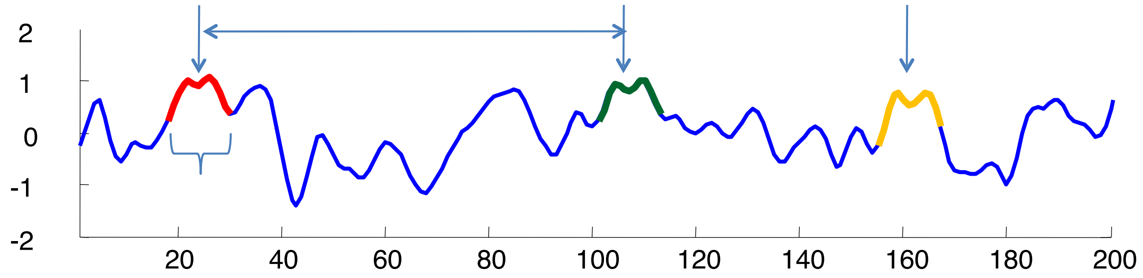
Repeated Pattern (Motifs)

Find the subsequences having very high similarity to each other.

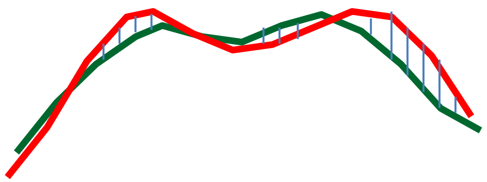


Finding Motifs in Time Series, Jessica Lin, Eamonn Keogh, Stefano Lonardi, Pranav Patel, KDD 2002

Definition of Time Series Motifs



1. Length of the motif
2. Support of the motif
3. Similarity of the Pattern
4. Relative Position of the Pattern

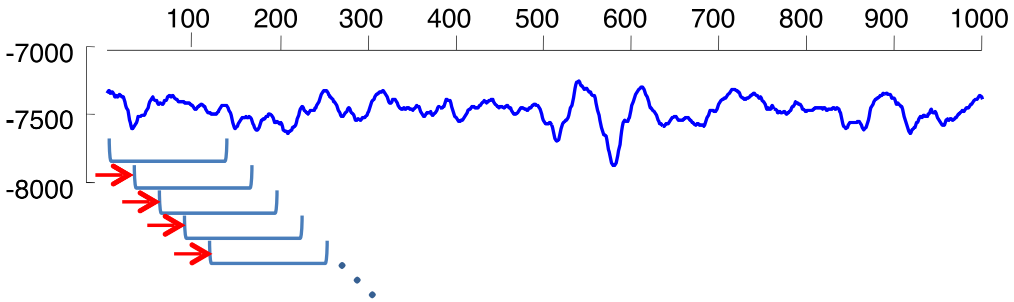


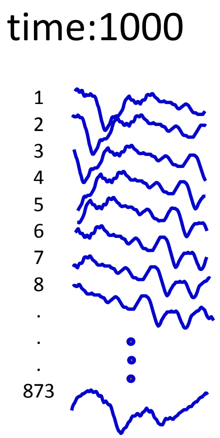
Given a **length,** the **most similar/least distant** **pair of** **non-overlapping** subsequences.

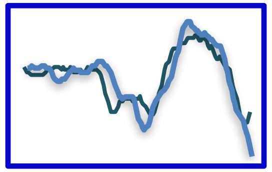
$$\hat{x}_i = \frac{x - \mu_x}{\sigma_x}, \hat{y}_i = \frac{y - \mu_y}{\sigma_y}$$

$$d(\hat{x}, \hat{y}) = \sqrt{\sum_i (\hat{x}_i - \hat{y}_i)^2}$$

Problem Formulation



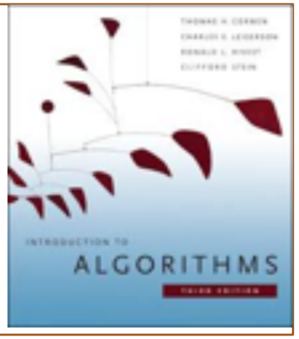




The most similar pair of non-overlapping subsequences

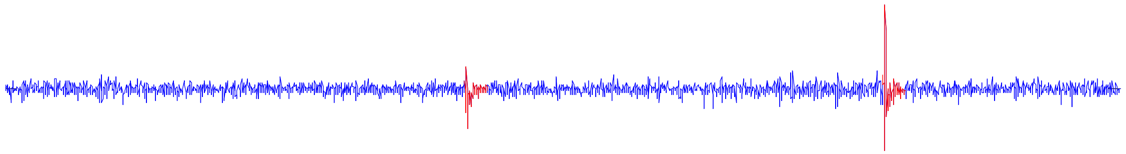
The closest pair of points in high dimensional space

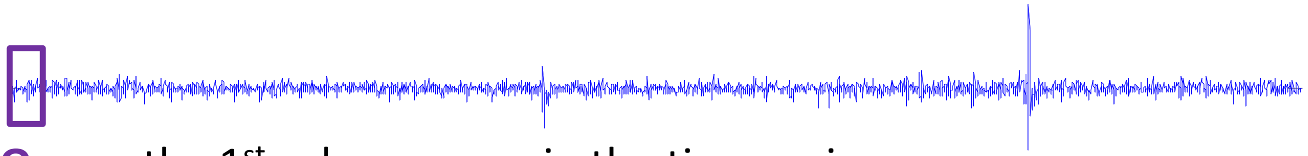
- Optimal algorithm in two dimension is *O(n log n)* (see textbook)
- For large dimensionality *d*, the optimal algorithm is effectively $O(n^2d)$
- STAMP: an $O(n^2 \log n)$ algorithm using FFT
- STOMP: an $O(n^2)$ algorithm by exploiting overlaps



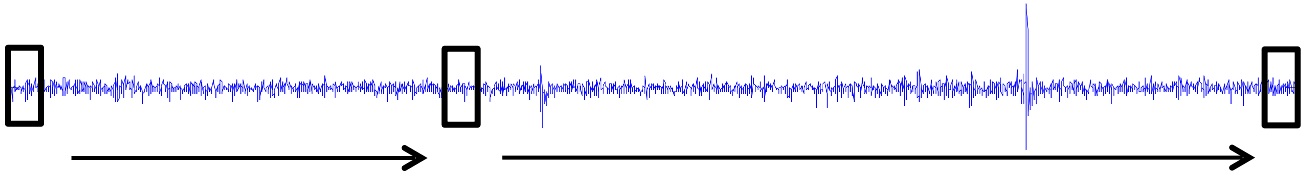
Definition Review: Distance Profile

A seismology time series, with two repeated earthquake patterns





Query, the 1st subsequence in the time series



Obtain the z-normalized Euclidean distance between **Query** and each window (subsequence) in the time series. We would obtain a vector like this:

$d_{1,1}$	$d_{2,1}$	…	$d_{n-m+1,1}$

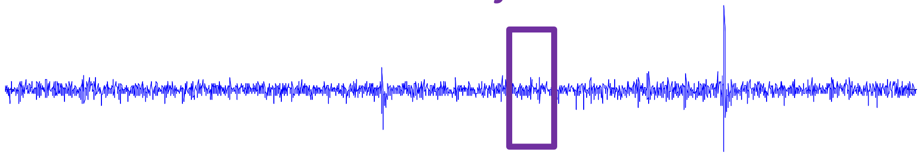
$\longrightarrow$ D_1

$d_{i,j}$ is the distance between the i^{th} subsequence and the j^{th} subsequence.

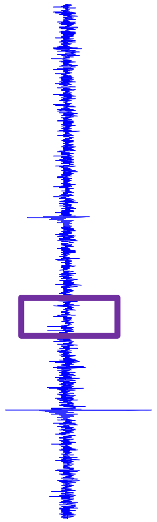
We can obtain D_2, D_3, … D_{n-m+1} similarly.

Definition Review: From Distance Profile to Matrix Profile

*j*th



*i*th



$d_{1,1}$	$d_{1,2}$	…	…	…	$d_{1,n-m+1}$
$d_{2,1}$	$d_{2,2}$	…	…	…	$d_{2,n-m+1}$
…	…	…	…	…	…
$d_{i,1}$	$d_{i,2}$	…	$\boldsymbol{d_{i,j}}$	…	$d_{i,n-m+1}$
…	…	…	…	…	…
$d_{n-m+1,1}$	$d_{n-m+1,2}$	…	…	…	$d_{n-m+1,n-m+1}$

$Min(D_1)$ $Min(D_2)$ $Min(D_i)$ $Min(D_{n-m+1})$

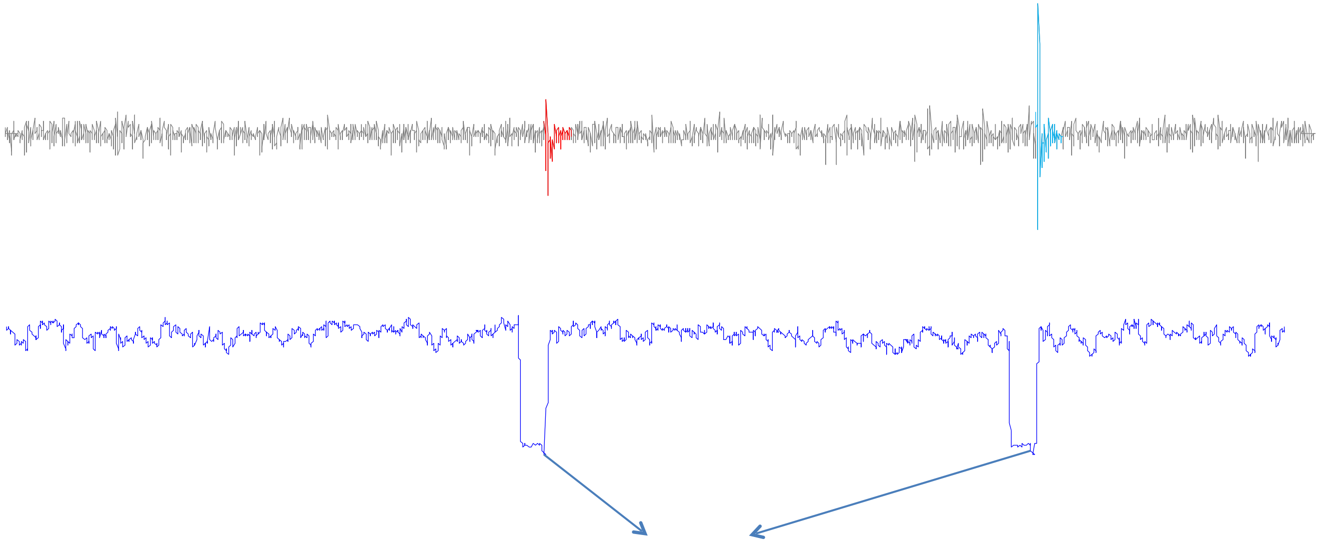
P_1	P_1	…	…	…	P_{n-m+1}

Note: this distance matrix is symmetric!

$d_{i,j}$ is the distance between the *i*th window and the *j*th window of the time series

Matrix Profile: a vector of distance between each subsequence and its nearest neighbor

From Matrix Profile to Motif

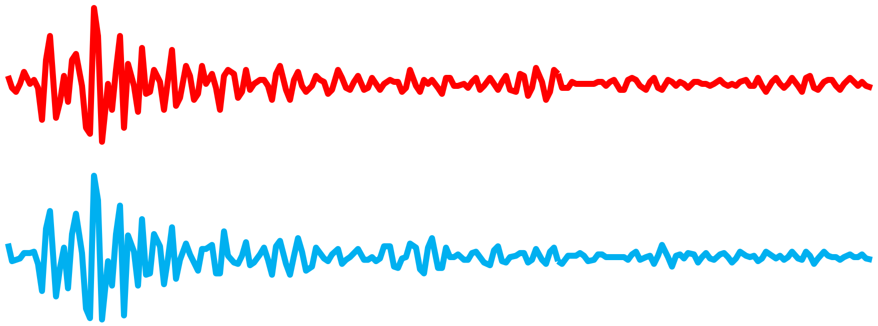


time series

matrix profile

A pair of minimum points

The Matrix Profile has two minimum points. This pair of minimum points correspond to the 1st motif in the time series.
(*the closest pair of subsequences in the time series*)



- ❖ It is sufficient to show that Matrix Profile Can be computed in
- ❖ *STAMP : $O(n^2 \log n)$*
- ❖ *STOMP : $O(n^2)$*

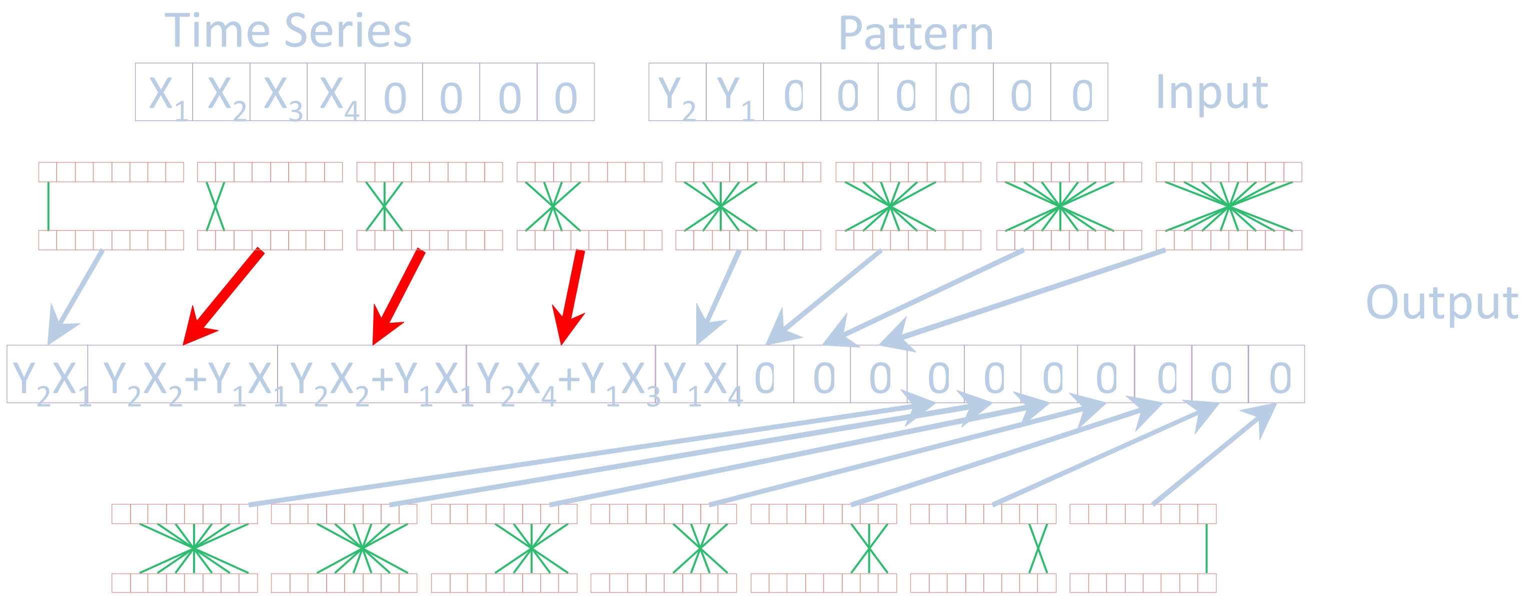
STAMP: Scalable Time series Anytime Matrix Profile

- STAMP uses MASS for each subsequence to calculate the Matrix Profile
- *Mueen's Algorithm for Similarity Search* (MASS) is an algorithm to find distance profiles in $O(n \log n)$
  – Convolution based method
    - $O(n \log n)$ cost to obtain all the sliding dot products
  – On-the-fly normalization
    - Use the dot products to calculate normalized distances in $O(n)$ cost

Chin-Chia Michael Yeh, Yan Zhu, Liudmila Ulanova, Nurjahan Begum, Yifei Ding, Hoang Anh Dau, Diego Furtado Silva, Abdullah Mueen, Eamonn Keogh (2016). **Matrix Profile I: All Pairs Similarity Joins for Time Series: A Unifying View that Includes Motifs, Discords and Shapelets.** IEEE ICDM 2016.

Mueen's Algorithm for Similarity Search (MASS) (1 of 3)

- Double the time series by appending zeros
- Reverse the normalized query and append zeros to match length
- Use FFT based convolution technique
  - $ifft(\,fft(x).fft(y)\,)$



Mueen's Algorithm for Similarity Search (MASS) (2 of 3)

- Since query is normalized, we reform the distance function by applying $\sum y = 0, \sum y^2 = m$
- $d^2(x, y) = \sqrt{2m\left(1 - \frac{\sum xy}{m\sigma_x}\right)}$
- The distance can be calculated by using the sliding dot products and sliding standard deviations
- Sliding means and standard deviations can be calculated in one linear scan

Mueen’s Algorithm for Similarity Search (MASS) (3 of 3)

- Produces a “distance profile” of the query to the subsequences of the time series. Every distance is reported, nothing is abandoned
- Can be used to answer K-NN queries, range queries, and density estimation all at the same time
- Data and query independent execution.
- Can be further optimized when multiple queries are issued

STOMP: Scalable Time series Optimum Matrix Profile

An ***O(n²)*** time, ***O(n)*** space algorithm called STOMP to compute Matrix Profile

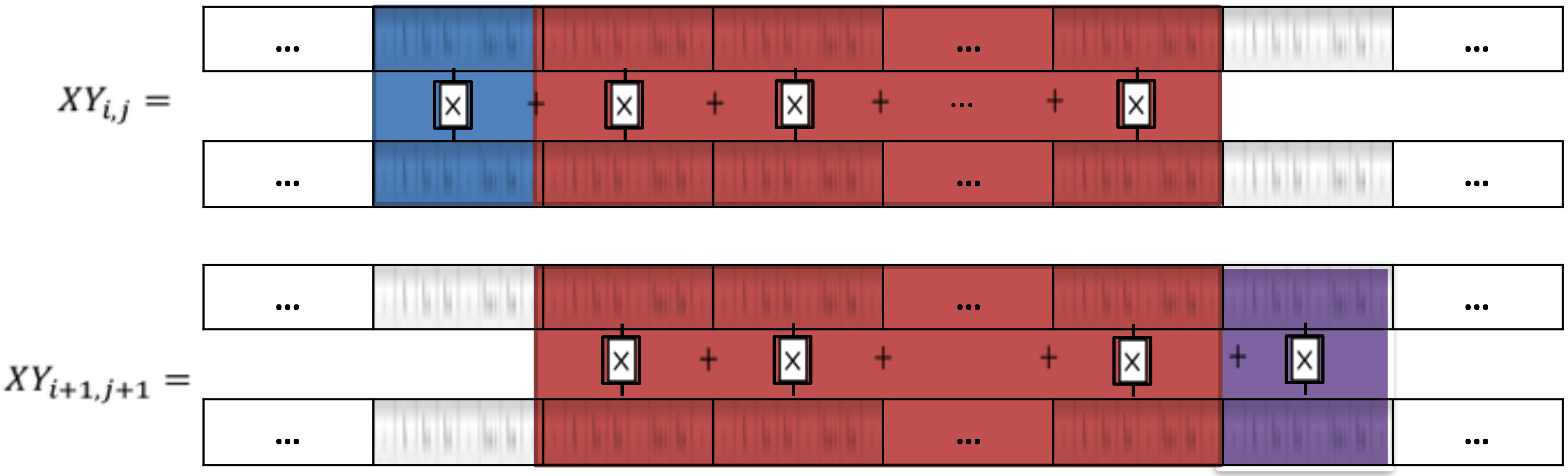
To see how it works, let us first introduce an important formula:

Dot product of the *i*th window and the *j*th window. Once we know $XY_{i,j}$, it takes *O(1)* time to compute $d_{i,j}$.

$$d_{i,j} = \sqrt{2m\left(1 - \frac{XY_{i,j} - m\mu_i\mu_j}{m\sigma_i\sigma_j}\right)}$$

We precompute and store the means and stdivs in *O(n)* space.

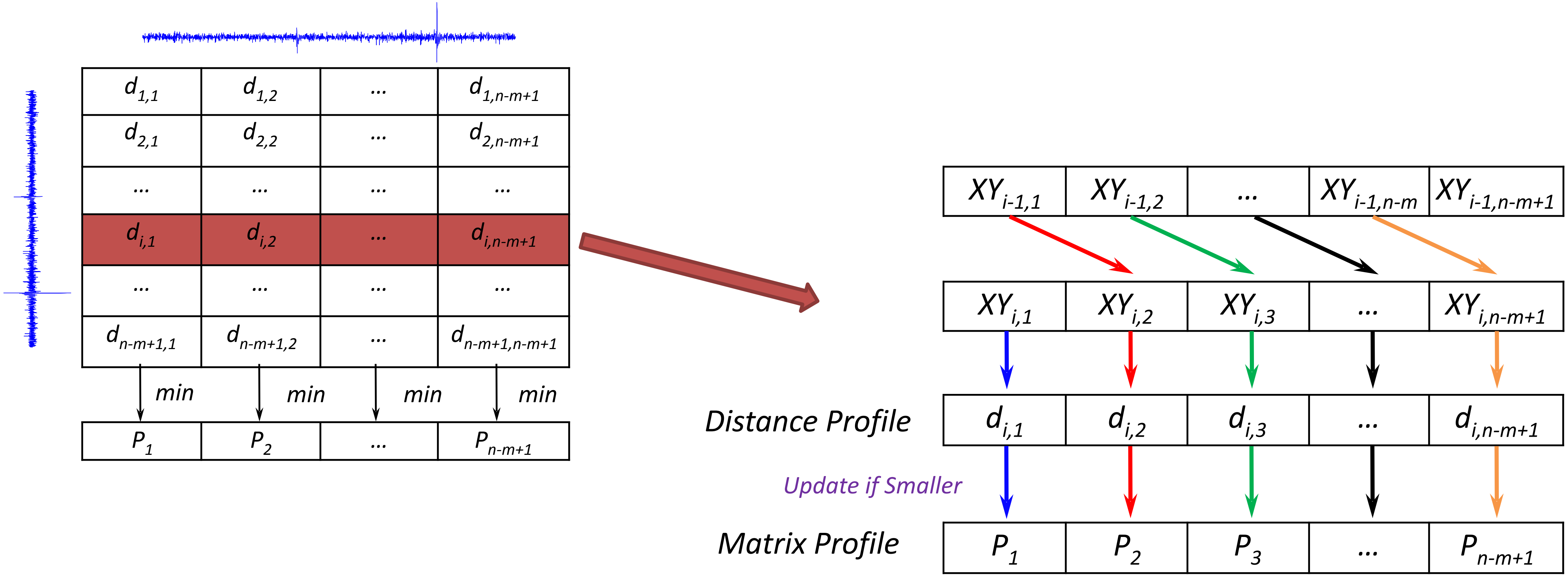
The relationship between $XY_{i,j}$ and $XY_{i+1,j+1}$



$$XY_{i+1,j+1} = XY_{i,j} - t_i t_j + t_{i+m} t_{j+m}$$

$O(1)$ time complexity!

STOMP Algorithm: Computing the i^{th} line



We pre-calculate $XY_{i,1}$ and $XY_{1,j}$ $(i\ ,\ j=1,2,3,\ldots,n\text{-}m+1)$.
Then iterate through $i=2,\ 2,\ 3,\ \ldots,\ n\text{-}m+1$.

Comparison of STAMP, STOMP and GPU-STOMP

For a fix subsequence length m=256: time

Algorithm *n*	2^{17}	2^{18}	2^{19}	2^{20}
STAMP	15.1 min	1.17 hours	5.4 hours	24.4 hours
STOMP	4.21 min	0.3 hours	1.26 hours	5.22 hours
GPU-STOMP	10 sec	18 sec	46 sec	2.5 min

For large data, and **for the very first time in the literature, 100,000,000**

Algorithm ***m* | *n***	**2000 | 17,279,800**	**400 | 100,000,000**
STAMP (*estimated*)	*36.5 weeks*	*25.5 years*
STOMP (*estimated*)	*8.4 weeks*	*5.4 years*
GPU-STOMP	9.27 hours	12.13 days

UNIVERSITY OF CALIFORNIA UC RIVERSIDE

Comparing the speed of STOMP with existing algorithms

For a time series of length 2^{18}: CPU time(memory usage)

Algorithm	512	1,024	2,048	4,096
STOMP	501s (14MB)	506s (14MB)	490s (14MB)	490s (14MB)
Quick-Motif	27s (65MB)	151s (90MB)	630s (295MB)	695s (101MB)
MK	2040s (1.1GB)	N/A (>2GB)	N/A (>2GB)	N/A (>2GB)

Note: the time and space cost of STOMP is independent of how the data looks.

UC RIVERSIDE UNIVERSITY OF CALIFORNIA

Outline

- Motifs
  - Definition
  - Never Ending Learning
  - Bot Detection
- Shapelets
  - Definition
  - Surface Classification
  - Patient Classification
- Discords
  - Definition
  - Star Light Curves
  - Search Frequency

Motifs allow us to learn, forever, without an explicit teacher…

If you have parallel texts, then over time you can learn a dictionary with high accuracy.

…And God said, "Let there be light"; and there was light. And God saw the light, that it was good. And God …

…Y dijo Dios: Sea la luz; y fue la luz. Y vio Dios que la luz *era* buena . Y llamó Dios a …

Motifs allow us to learn, forever, without an explicit teacher…

If you have parallel texts, then over time you can learn a dictionary with high accuracy.

…And God said, “Let there be light”; and there was light. And God saw the light, that it was good. And God …

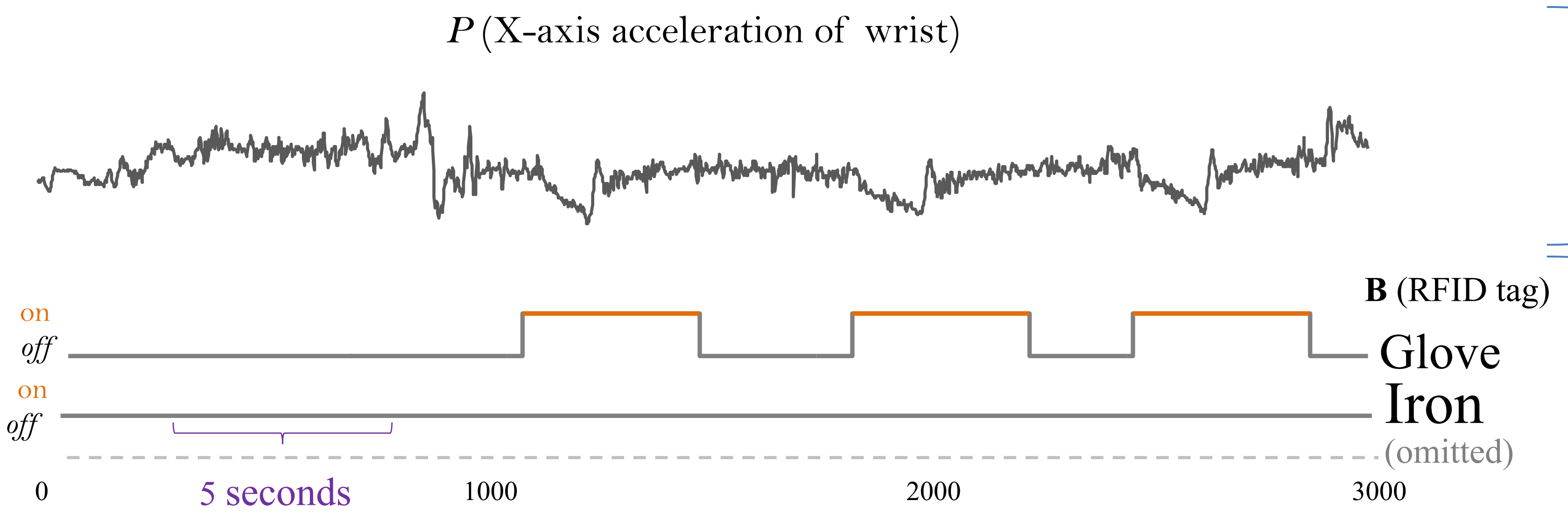
…Y dijo Dios: Sea la luz; y fue la luz. Y vio Dios que la luz *era* buena . Y llamó Dios a …

Note the mapping is non-linear, the learning algorithms in this domain are non-trivial.

Suppose however that the unknown “language” is not *discrete*, but *real-valued* time series? In this case, repeated pattern discovery can help*…

*Yuan Hao, Yanping Chen et al (2013). Towards Never-Ending Learning from Time Series Streams. SIGKDD 2013

Motifs allow us to learn, forever, without an explicit teacher…

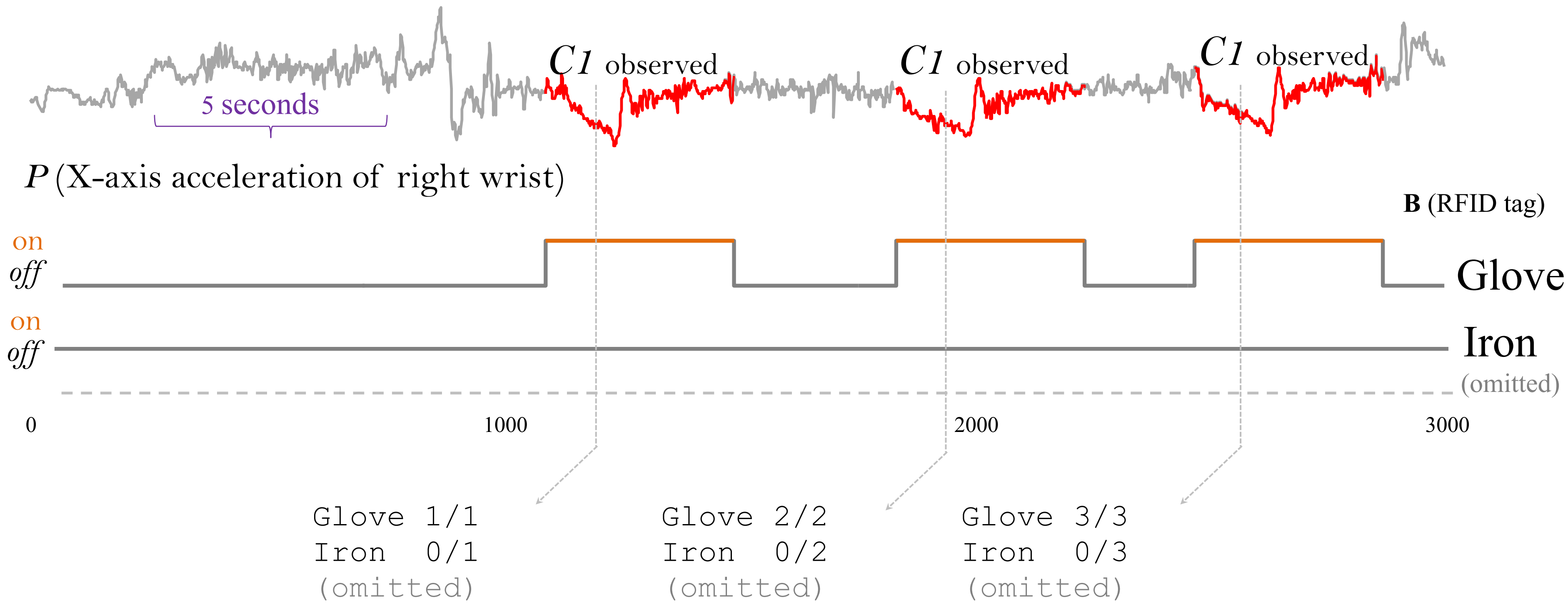


Real-valued “text”

Binary “text”

This dataset contains standard IADL housekeeping activities (vacuuming, ironing, dusting, brooming,, watering plants etc). We have a discrete (binary) “text” that notes if the hand is near a cleaning instrument, and a real-valued accelerometer “text”

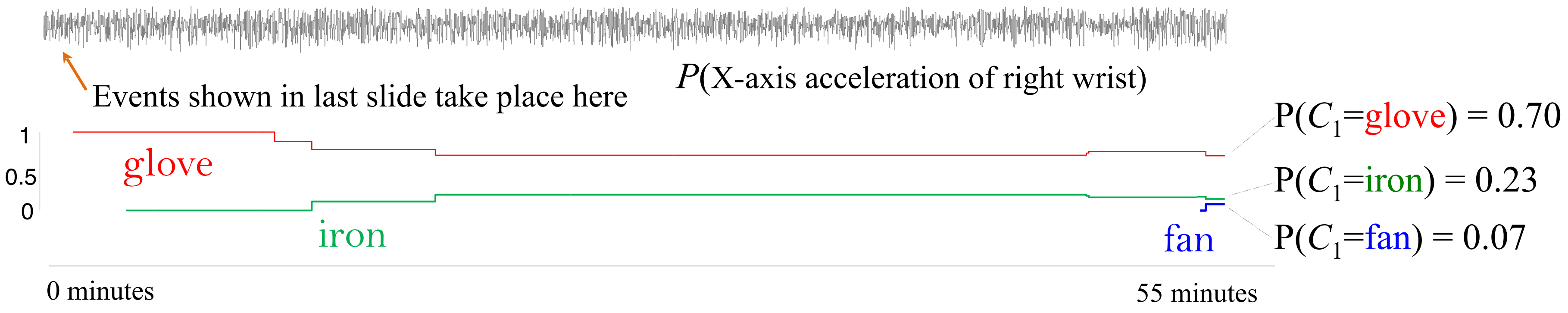
We can run motif discovery on the time series stream. If we find motifs, we can see if they correlate with the discrete streams…



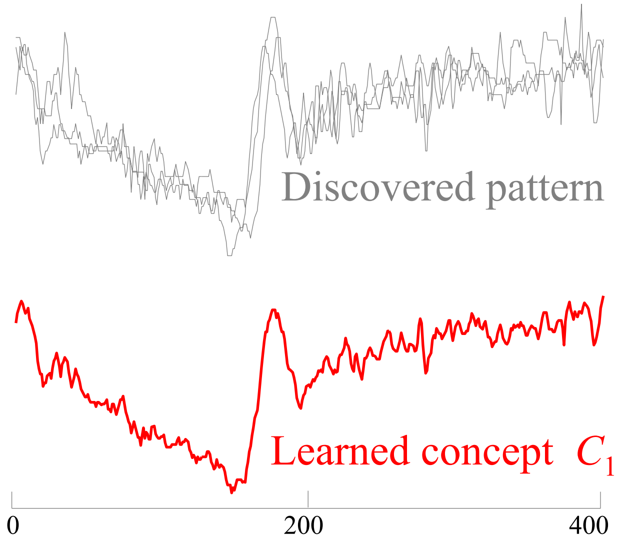
In this snippet, the motifs seem to correlate with the presence of a glove…

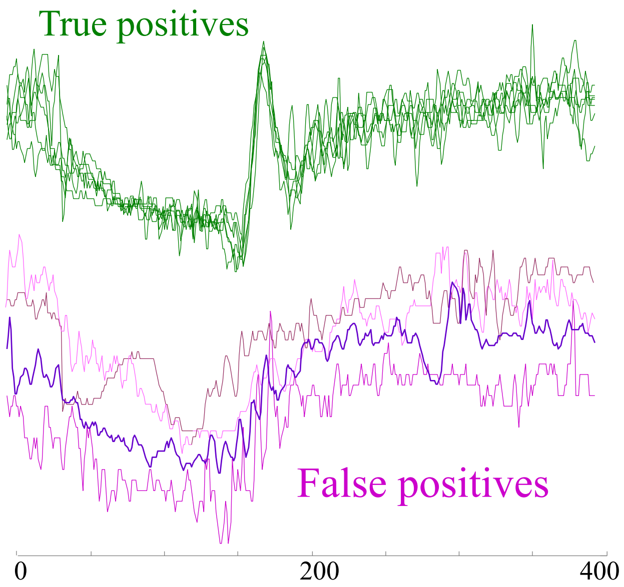
How well does this work?

Over an hour of activity, we learn to recognize a behavior in the time series that indicates the user is putting on a glove.



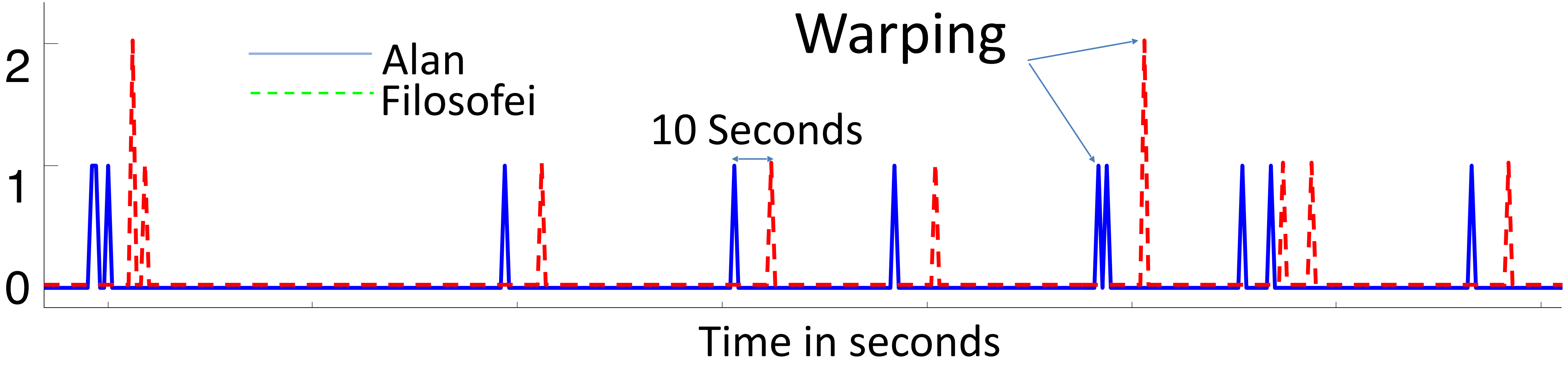
Note: There are false positives, but we considered only a single axis for simplicity.





Twitter Bots

Twitter Bots



- Correlation Coefficient is ~0
- Maximum Cross-correlation is 0.72
- Warping Invariant Correlation is 0.99
- 100 million users → over a million users over 10 tweets
- $\sim 10^{18}$ floating point operations

AWarp: Warping Distance for Asynchronous Time Series A Mueen, N Chavoshi, N Abu-El-Rub, H Hamooni, A Minnich, ICDM 2016.

DeBot Archive

590K Bots Since August 2015

http://cs.unm.edu/~chavoshi/debot/api.html

Python API Available at:

https://github.com/nchavoshi/debot_api

Outline

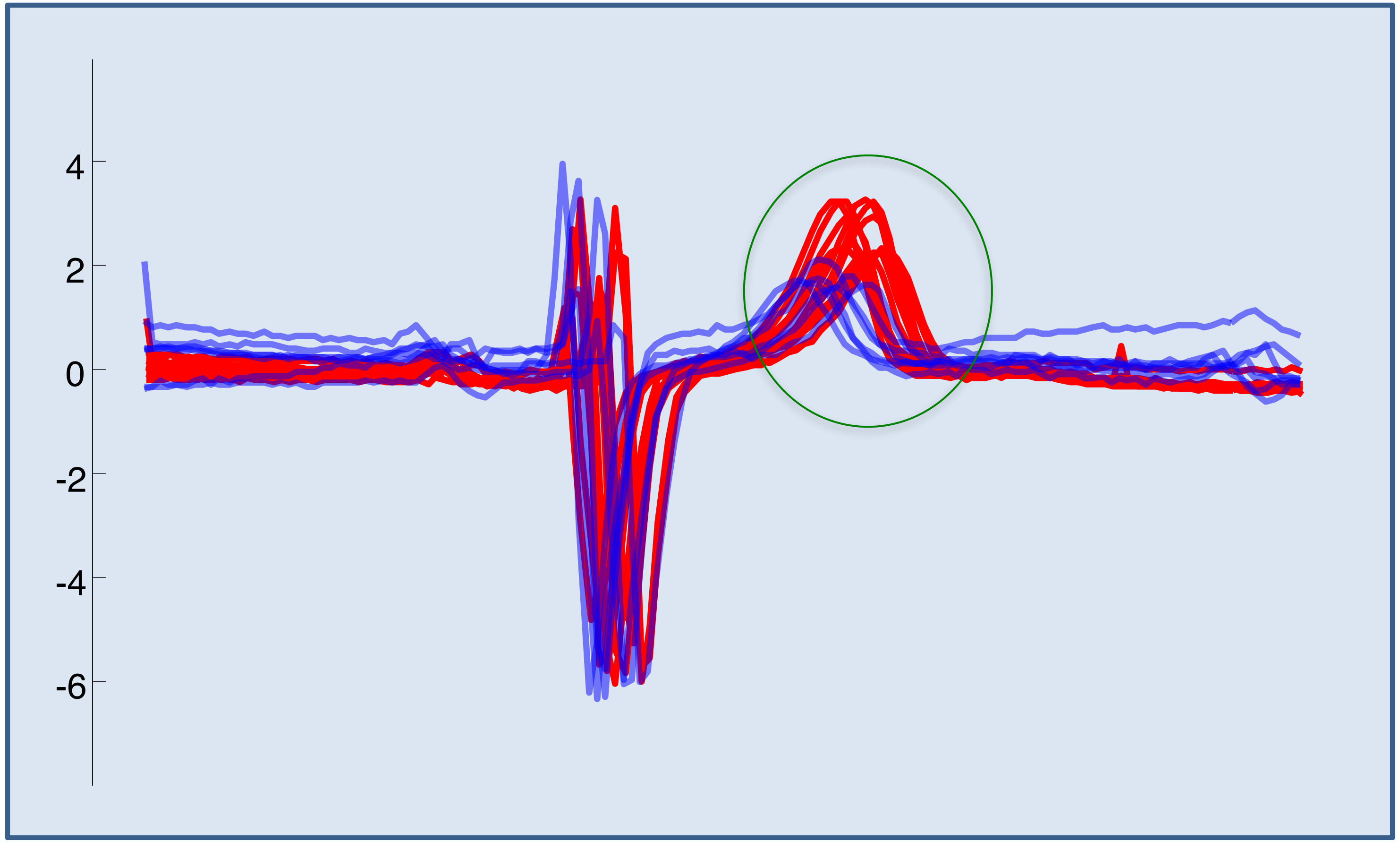
- Motifs
  - Definition
  - Never Ending Learning
  - Bot Detection
- Shapelets
  - Definition
  - Surface Classification
  - Patient Classification
- Discords
  - Definition
  - Star Light Curves
  - Search Frequency

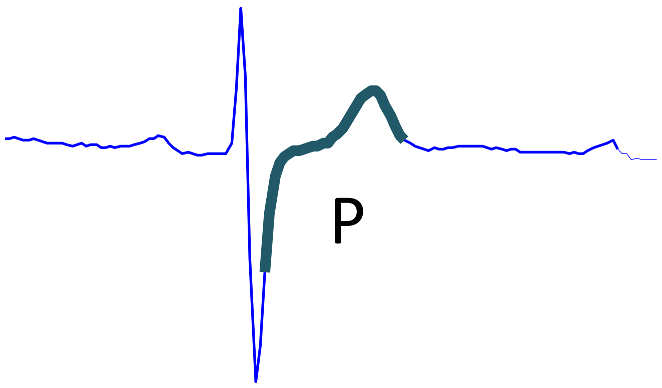
Discriminating Patterns (Shapelet)

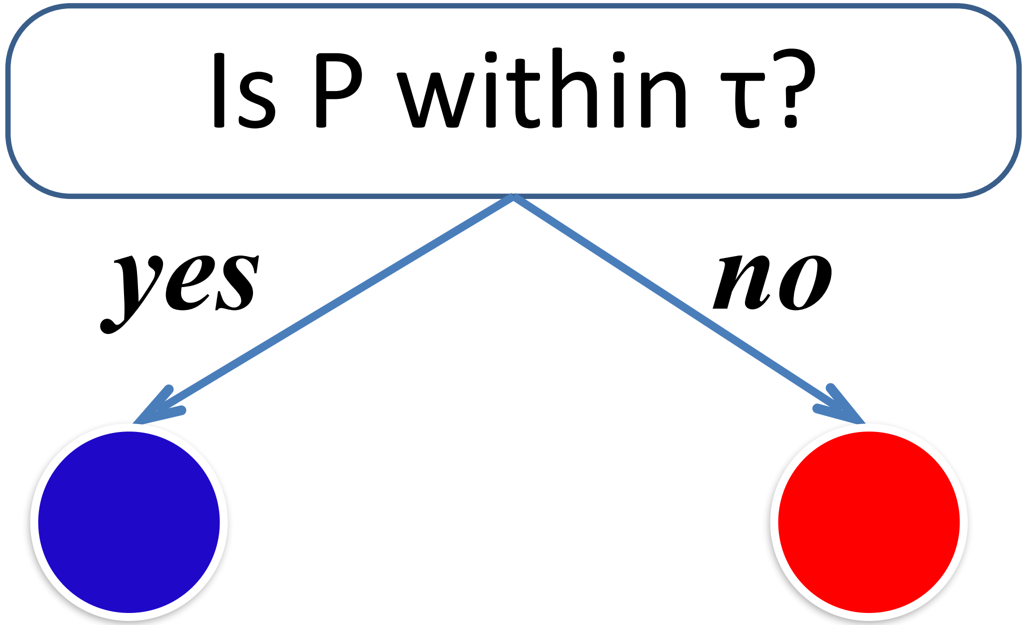
Data from a 67 year old male

ECG date: 11/12/1990

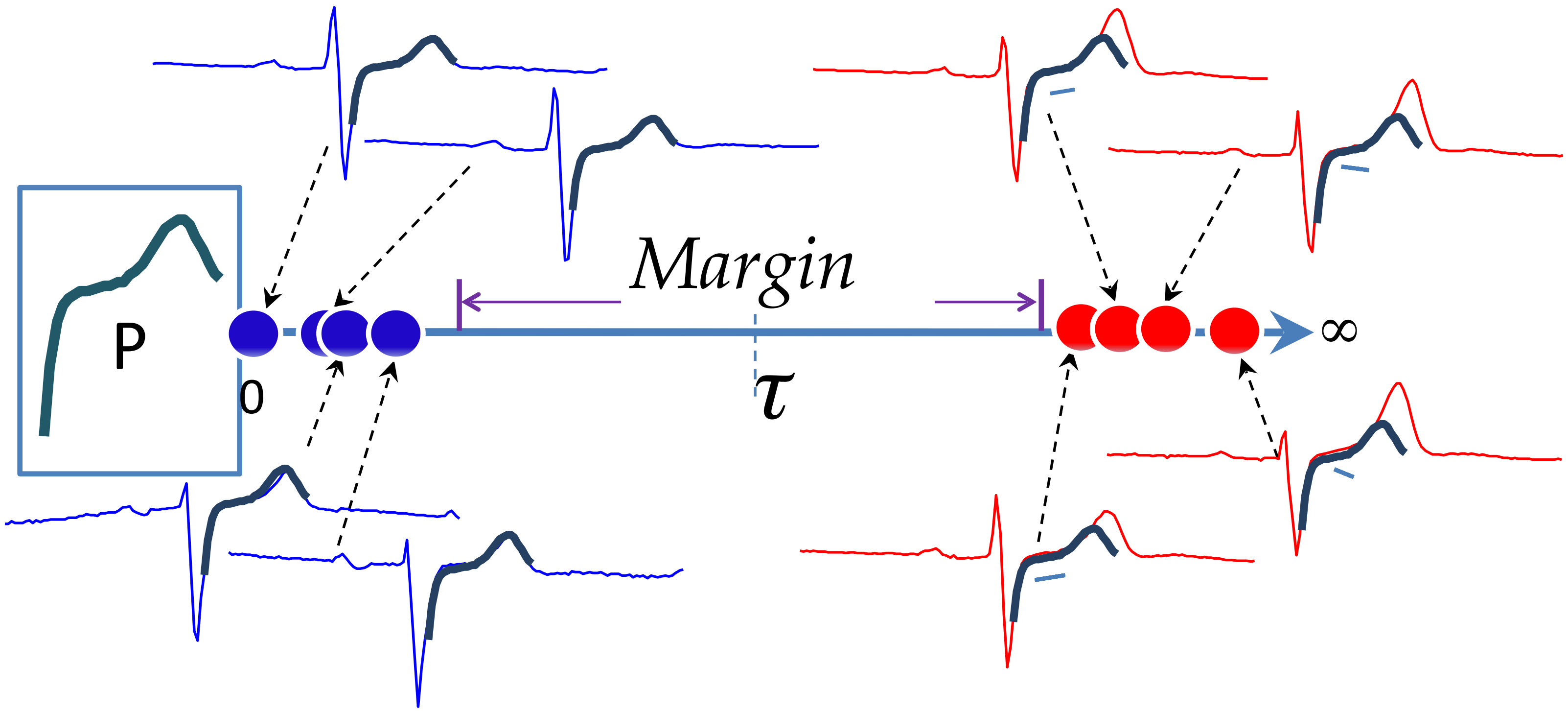
ECG date: 11/17/1990







Problem Definition



Shapelet is the pattern that splits the dataset for *maximum information gain*

Breaking ties by maximizing *separation margin*

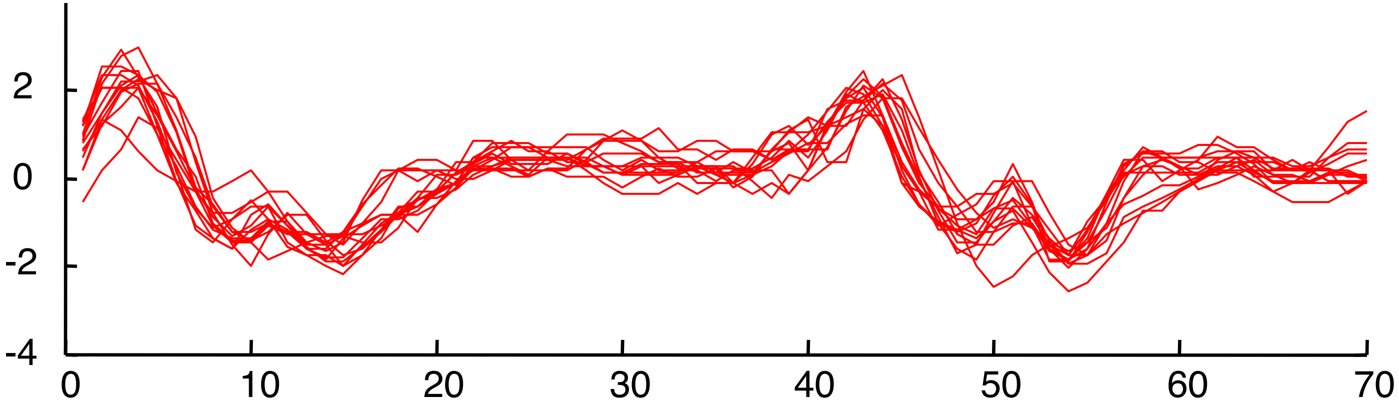
$$I(p,\tau) = E(D) - \frac{N_{left}}{N} E(D_{left}) - \frac{N_{right}}{N} E(D_{right})$$

Information Gain

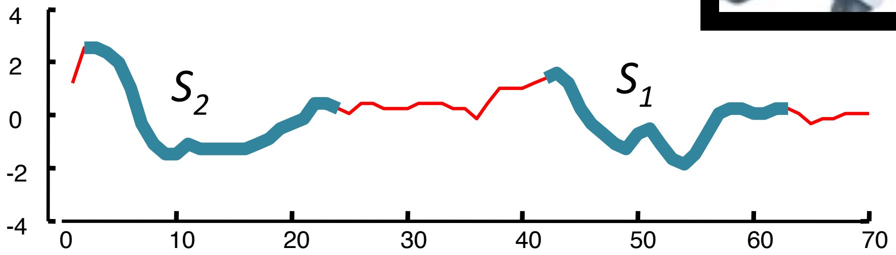
$$E(D) = \sum_{i \in C} p_i \log p_i = \sum_{i \in C} \frac{N_i}{N} \log \frac{N_i}{N}$$

Entropy

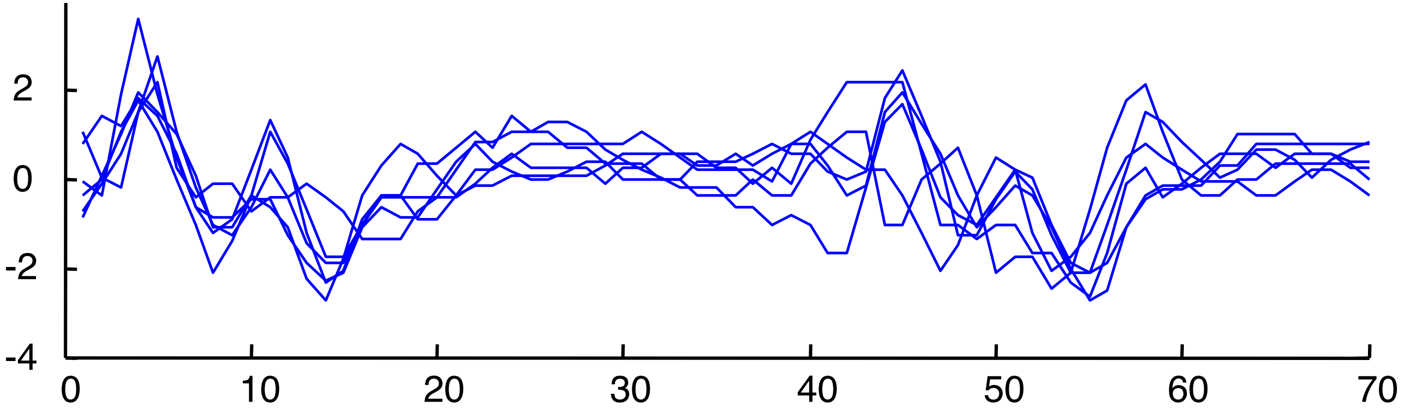
Shapelets: Surface Classification



Walking on Carpet (Soft)



Walking on Carpet



Walking on Cement (Hard)

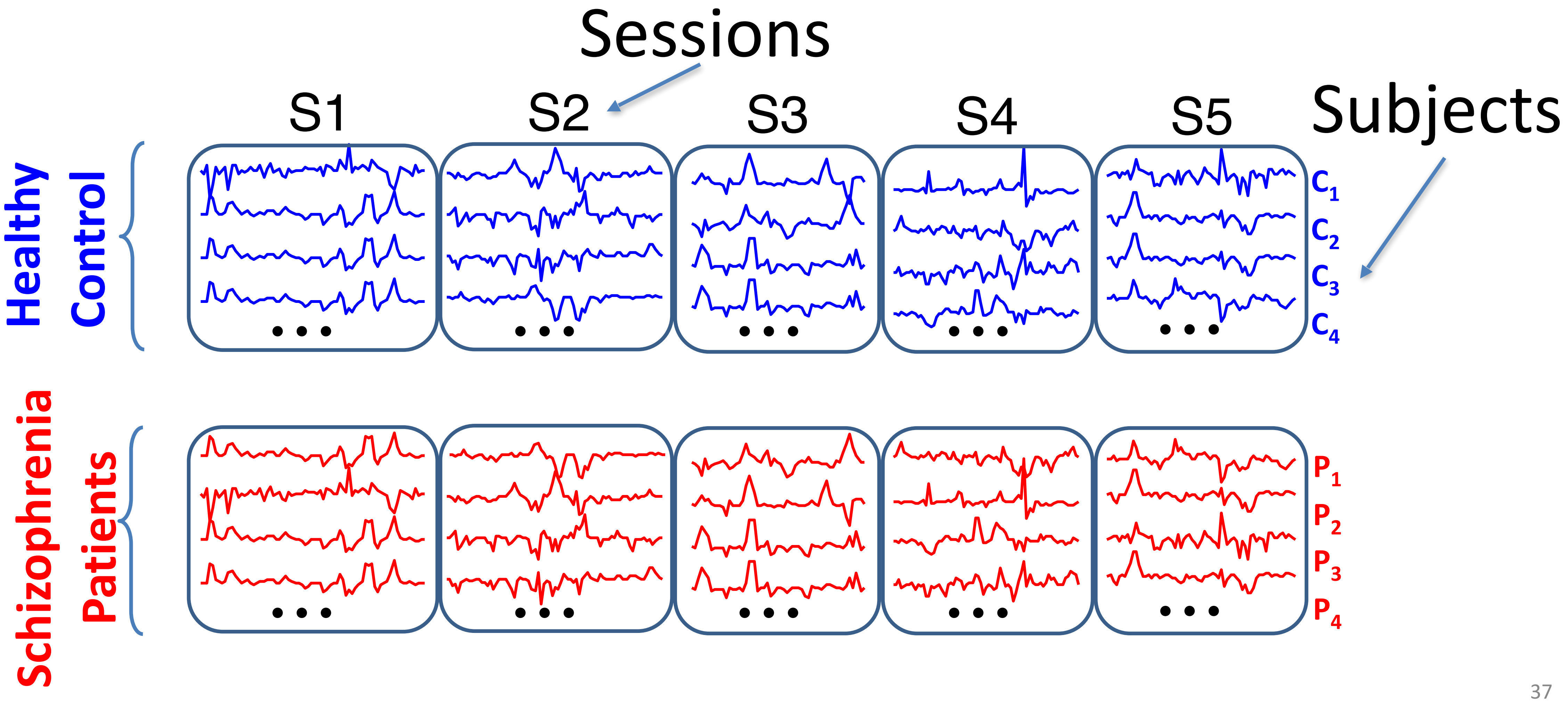
1-NN	69.55%
1-NN DTW	72.55%
1-NN DTW S-C	69.55%
Shapelet (S_1)	**93.34%**
Shapelets (S_1 and S_2)	**96.34%**

Electroencephalography

ICA

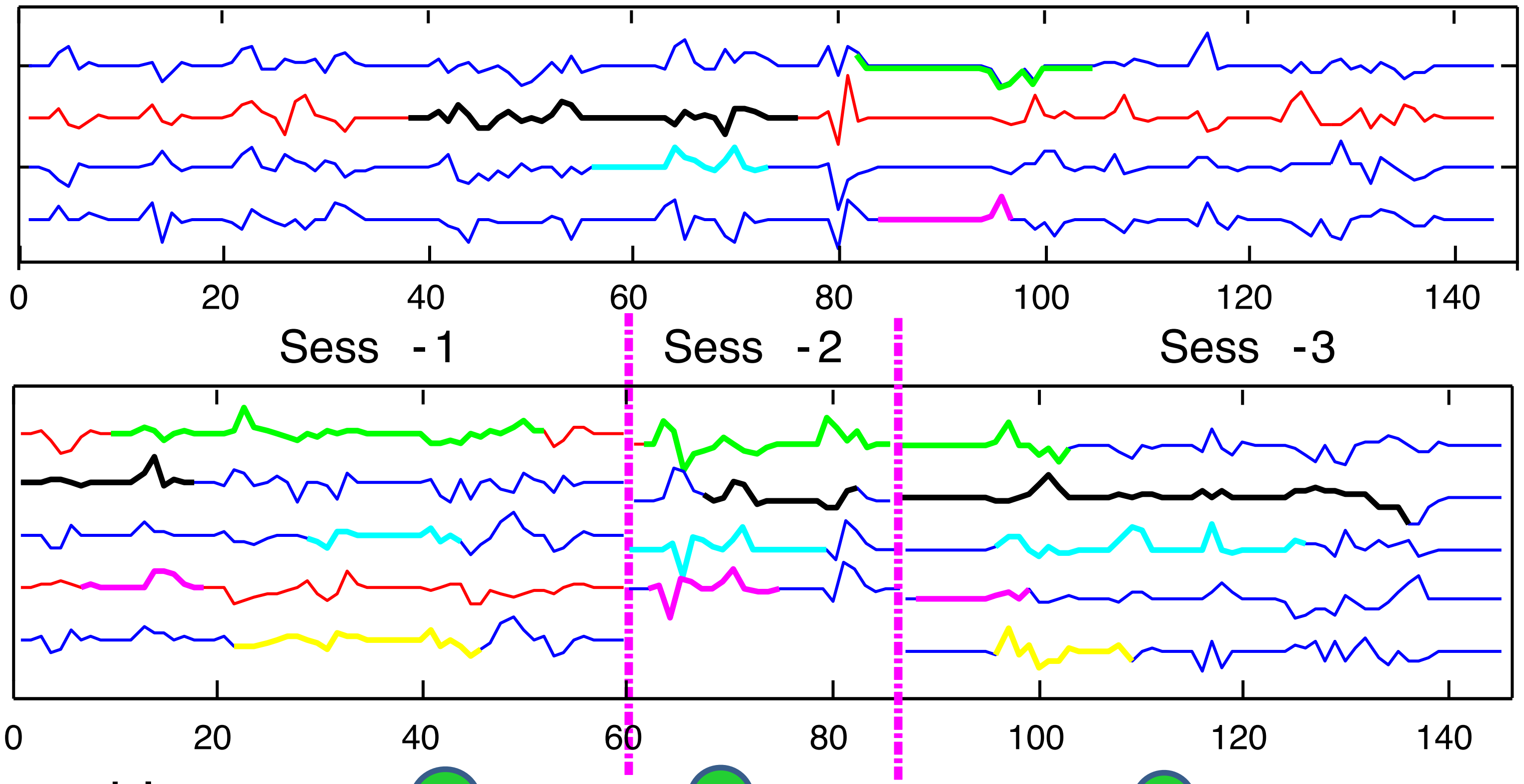
Functionally Independent Components

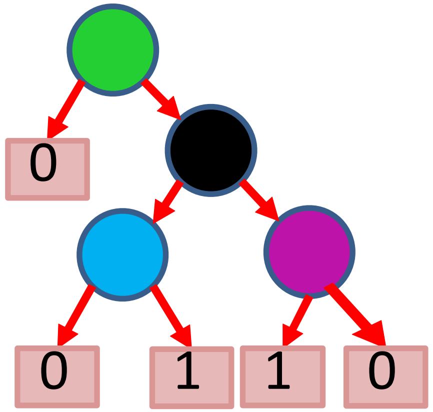
Design an objective biological marker-based diagnostic test for mental disorders



Sensory Gating task for fMRI

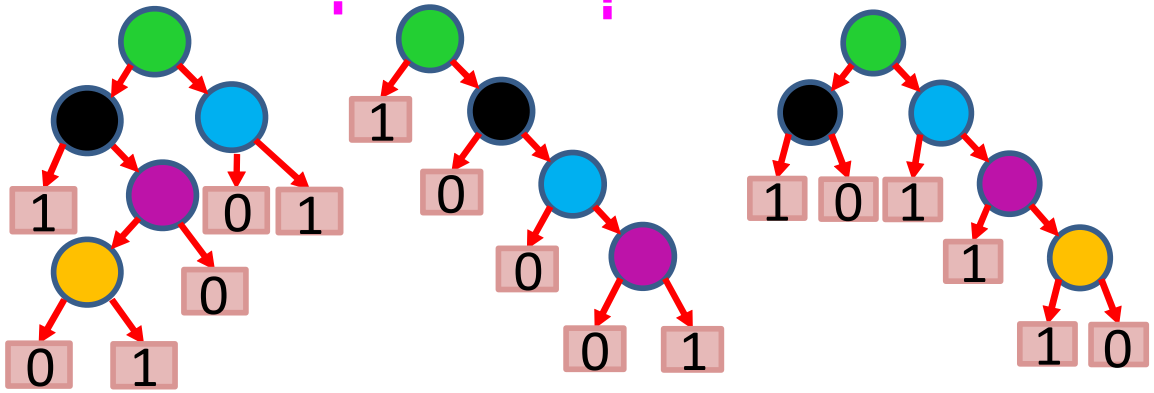
Concatenation





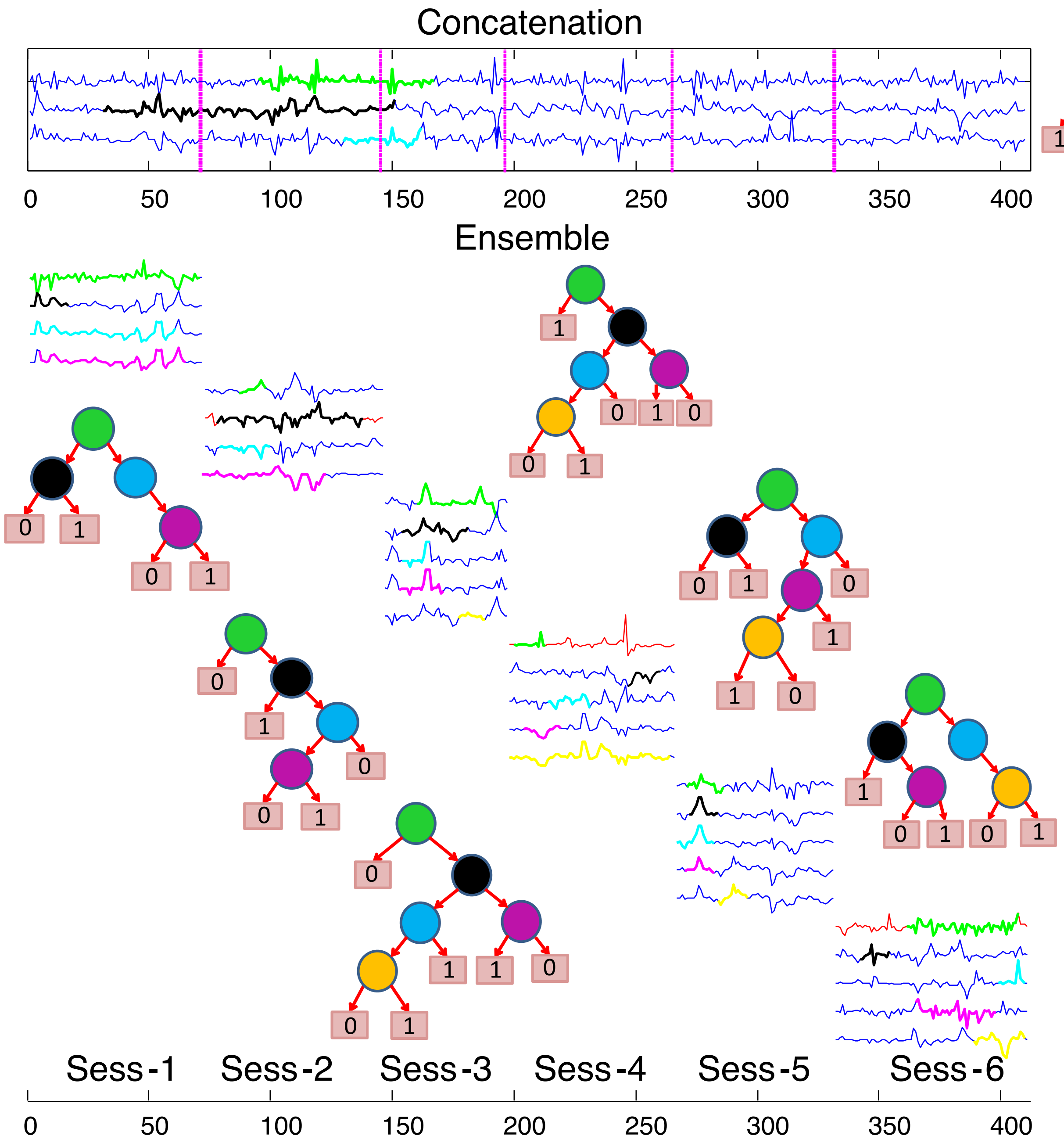
66%

Ensemble



76%

Multi-modal sensory integration task for fMRI



64%

72%

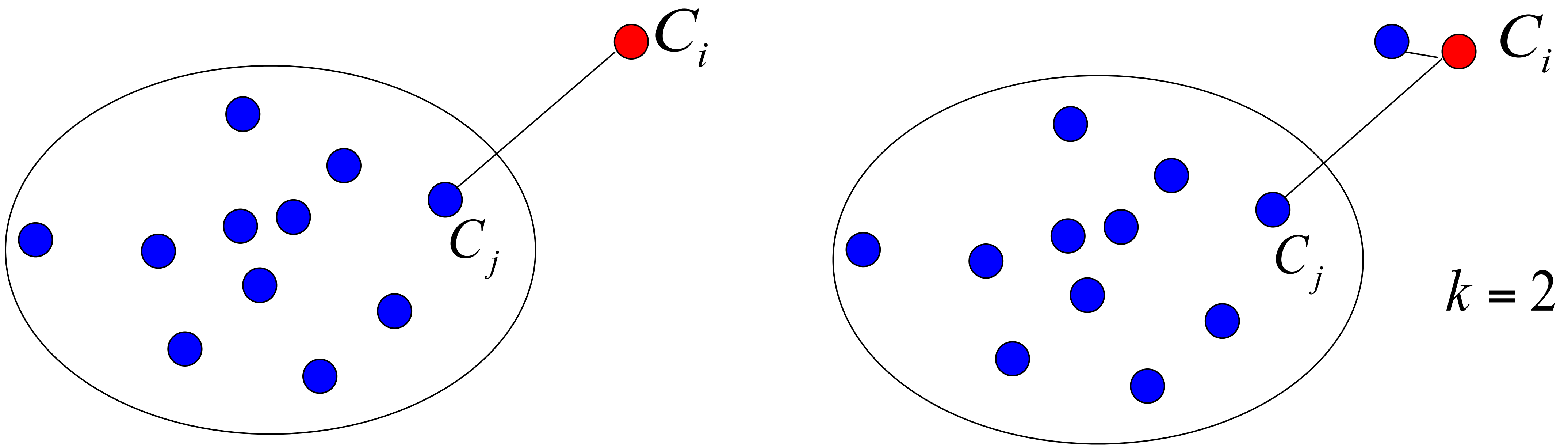
Outline

- Motifs
  - Definition
  - Never Ending Learning
  - Bot Detection
- Shapelets
  - Definition
  - Surface Classification
  - Patient Classification
- Discords
  - Definition
  - Star Light Curves
  - Search Frequency

Anomalous Pattern: Discords

The subsequence with maximal distance to its nearest neighbor

The subsequence with maximal distance to its *k-th* nearest neighbor

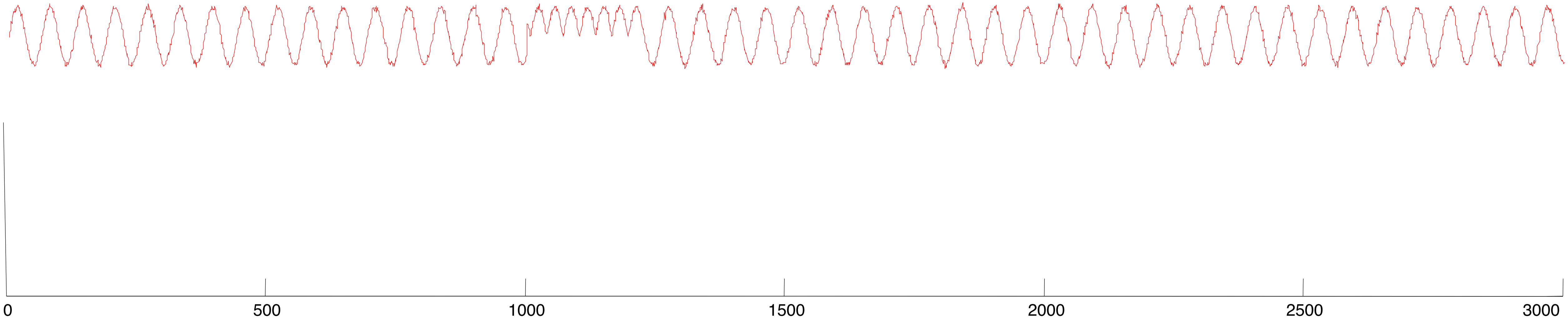


The matrix profile identifies discords

The highest values correspond to the **time series discords**

To see this, let us consider another dataset. Below is a slightly noisy sine wave. I have added an anomaly by taking the absolute value in the region between 1,000 and 1,200.

What would the matrix profile look like for this time series? (next slide).



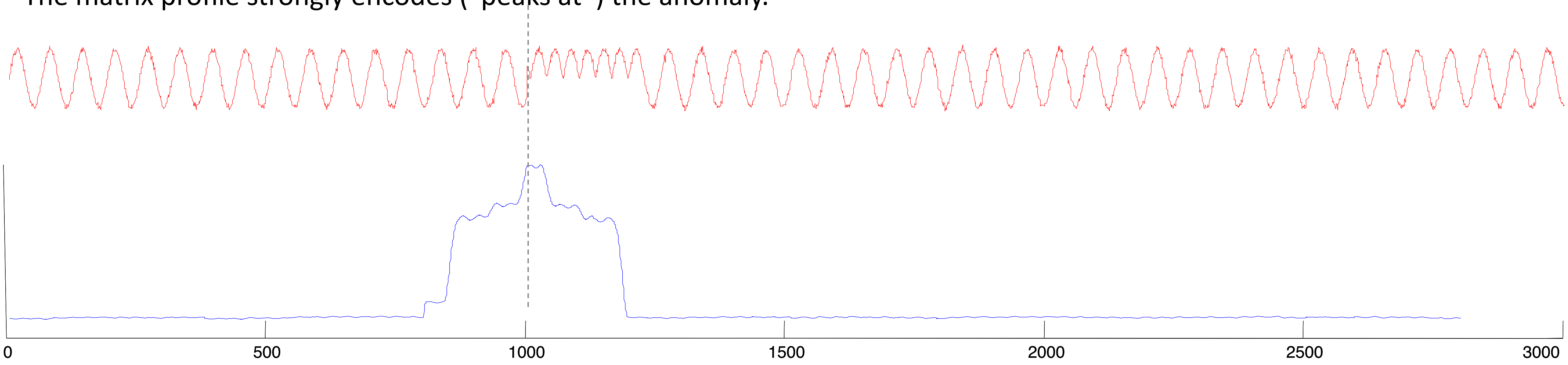
Vipin Kumar performed an extensive empirical evaluation and noted that "*..on 19 different publicly available data sets, comparing 9 different techniques (time series discords) is the best overall technique.*". V. Chandola, D. Cheboli, V. Kumar. Detecting Anomalies in a Time Series Database. UMN TR09-004

The matrix profile identifies discords

The highest values correspond to the **time series discords**

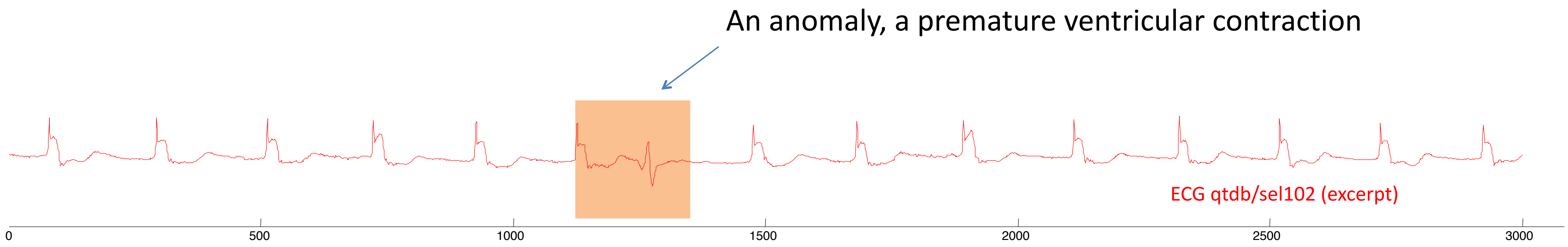
To see this, let us consider another dataset. Below is a slightly noisy sine wave. I have added an anomaly by taking the absolute value in the region between 1,000 and 1,200.

The matrix profile strongly encodes ("peaks at") the anomaly.



Vipin Kumar performed an extensive empirical evaluation and noted that "*..on 19 different publicly available data sets, comparing 9 different techniques (time series discords) is the best overall technique.*". V. Chandola, D. Cheboli, V. Kumar. Detecting Anomalies in a Time Series Database. UMN TR09-004

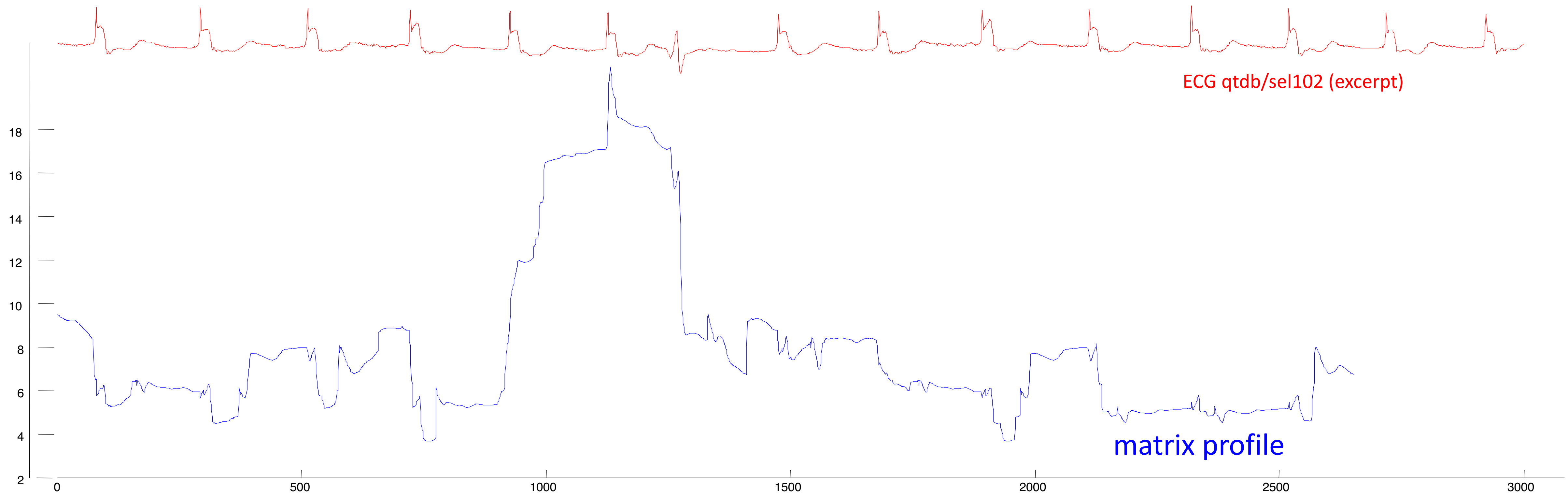
Matrix Profiles as Anomaly Detectors: 1 of 2



Let us use a matrix profile to see if we can spot this anomaly (next slide)

Matrix Profiles as Anomaly Detectors: 2 of 2

The alignment of the peak of the matrix profile and the ground truth is sharp and perfect!



Discords in Light-curve Data

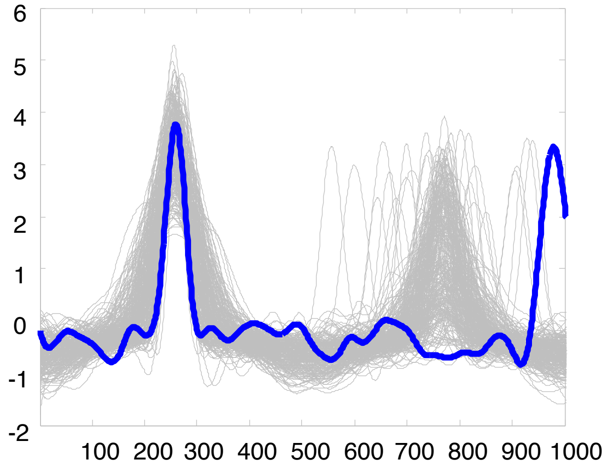
Star light-curve data from the Optical Gravitational Lensing Experiment (OGLE)

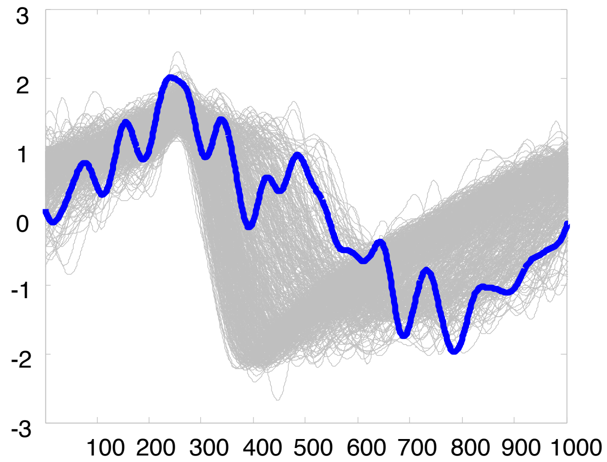
Three classes of light-curves

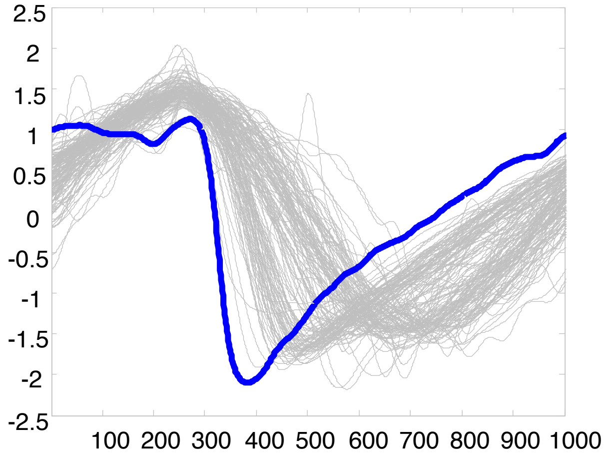
Eclipsed binaries

Cepheids

RR Lyrae variables

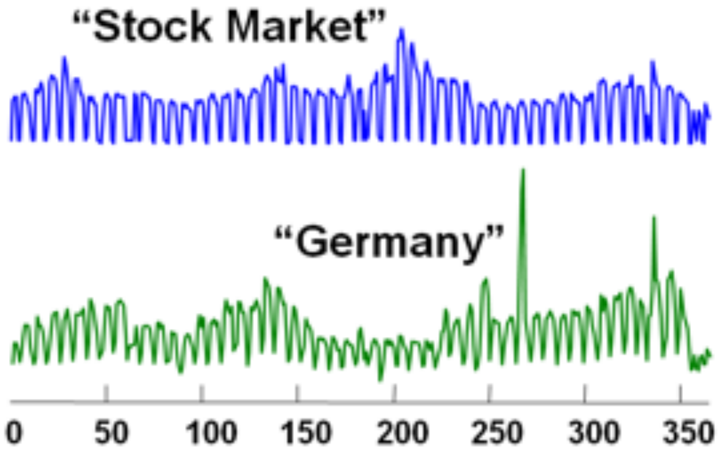




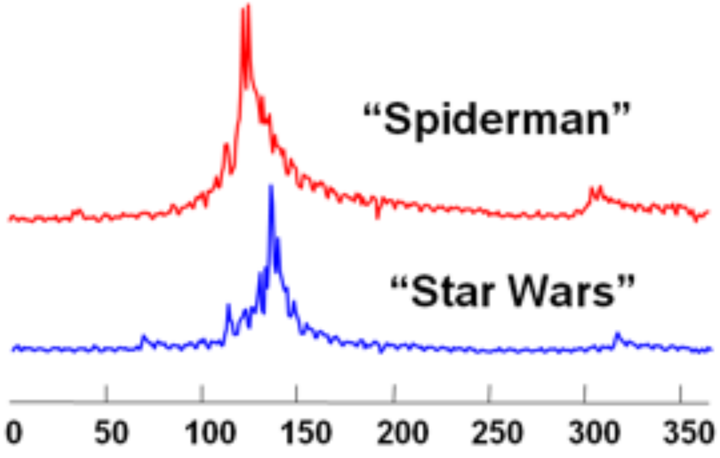


Discords in Search Data

MSN web queries made in 2002



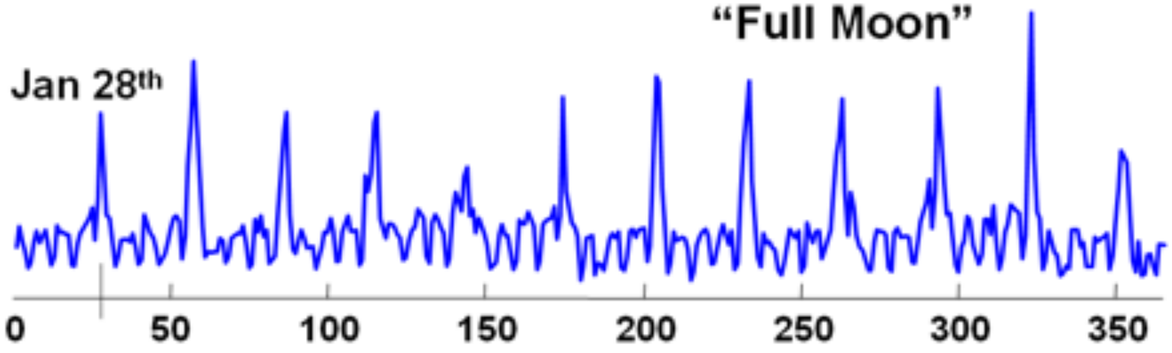
patterns dominated by a weekly cycle



anticipated bursts

The most significant discord using rotation invariant Euclidean distance

periodicity 29.5 days – the length of a synodic month



Conclusion

- Motifs: Repeated Patterns in Time Series
  - Case studies in dictionary building and bot detection
- Shapelets: Discriminating Patterns
  - Case studies in patient and surface classification
- Discords: Anomalous Patterns in Time Series
  - Case studies in Astronomical and Search Frequency Data

Questions and Comments?

THANK YOU