Eamonn Keogh

UCR

Finding Approximately Repeated Patterns in Time Series

(The most Useful, and yet most Underutilized Primitive in Time Series Analytics)

data intelligence institute of Paris

UC RIVERSIDE

Slides are here: https://www.dropbox.com/s/i38eyidz1qo9pi3/Motifs.pptx?dl=0

Notes

- My internet reboots randomly once a week or so. If I disappear, take a three-minute break, I will be back ;-)
- I use essentially no math in the talk, I want to communicate *intuitions*.
- I occasionally show some code, this is just to demonstrate you can do amazing things with *just* 3 or 4 lines of code!
- I will show lots of case studies, to demonstrate the generality of these ideas.

Slides are here: https://www.dropbox.com/s/i38eyidz1qo9pi3/Motifs.pptx?dl=0

Overarching Philosophy

- There are many ways to analyze/mine time series data.
- Techniques include Fourier methods, wavelets, PCA, ARIMA models, Markov Models, feature extraction, deep learning etc.

- However, I claim that:

Most problems in time series analyses can be solved by simply "reasoning" about the shape similarity of local subsequences in the data.

Fundamental Assumption: *Conservation* is Key

If a pattern is *conserved*, there must be some mechanism that conserves it. This is true in linguistics, music, genetics, literature, religions….

For example, most words are *not* conserved in distance languages, vélo|bicycle|podílato but a handful are…

* Bengali: Bābā
* Mandarin : baba
* Polish : tata
* Swahili : baba
* Turkish : baba
* Xhosa: -tata
* Norwegian : papa
* Spanish : papá
* Swahili : baba
* English : papa
* Hindi : papa
* Indonesian : bapa

en.wikipedia.org/wiki/Mama_and_papa

Fundamental Assumption: *Conservation is Key*

If a pattern is *conserved*, there must be some mechanism that conserves it. This is true in linguistics, music, genetics, literature, religions….

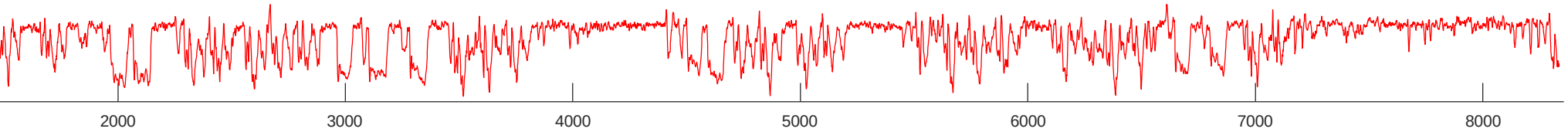
Conservation can yield surprises. There is more conserved DNA between a hippo and a dolphin, than a hippo and a pig..

* Bengali: Bābā
* Mandarin : baba
* Polish : tata
* Swahili : baba
* Turkish : baba
* Xhosa: -tata
* Norwegian : papa
* Spanish : papá
* Swahili : baba
* English : papa
* Hindi : papa
* Indonesian : bapa

en.wikipedia.org/wiki/Mama_and_papa

Dolphin …ACATTATCCGCATCTATA…
Hippo …ACATTATCGGCATCTATA…
Pig …TCTATCGCAATTTAATCA…

Fundamental Assumption: *Conservation is Key*



If a pattern is *conserved*, there must be some mechanism that conserves it. This is true in linguistics, music, genetics, literature, religions….

Much of my work asks *what* is conserved in time series, *when* is it conserved, and *why* was an expected conservation not observed…

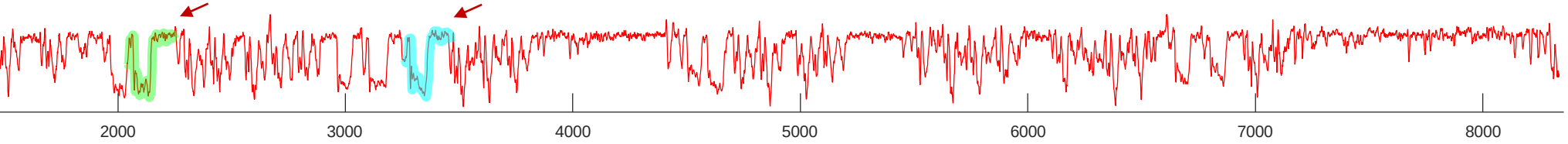
For discrete strings, *conserved* is easy to define, for example `papa =*a*a`. For *time series* it requires a distance function, here we will use Euclidean Distance.

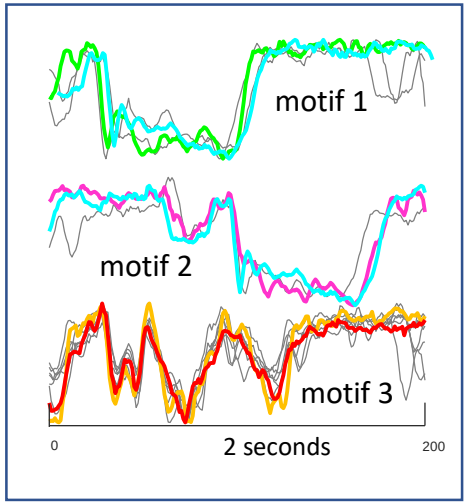
* Bengali: Bābā	* Norwegian : papa
* Mandarin : baba	* Spanish : papá
* Polish : tata	* Swahili : baba
* Swahili : baba	* English : papa
* Turkish : baba	* Hindi : papa
* Xhosa: -tata	* Indonesian : bapa

en.wikipedia.org/wiki/Mama_and_papa

```
Dolphin ...ACATTATCCGCATCTATA...
Hippo   ...ACATTATCGGCATCTATA...
Pig     ...TCTATCGCAATTTAATCA...
```

Fundamental Assumption: *Conservation is Key*





If a pattern is *conserved*, there must be some mechanism that conserves it. This is true in linguistics, music, genetics, literature, religions….

Much of our work asks *what* is conserved in time series, *when* is it conserved, and *why* was an expected conservation not observed…

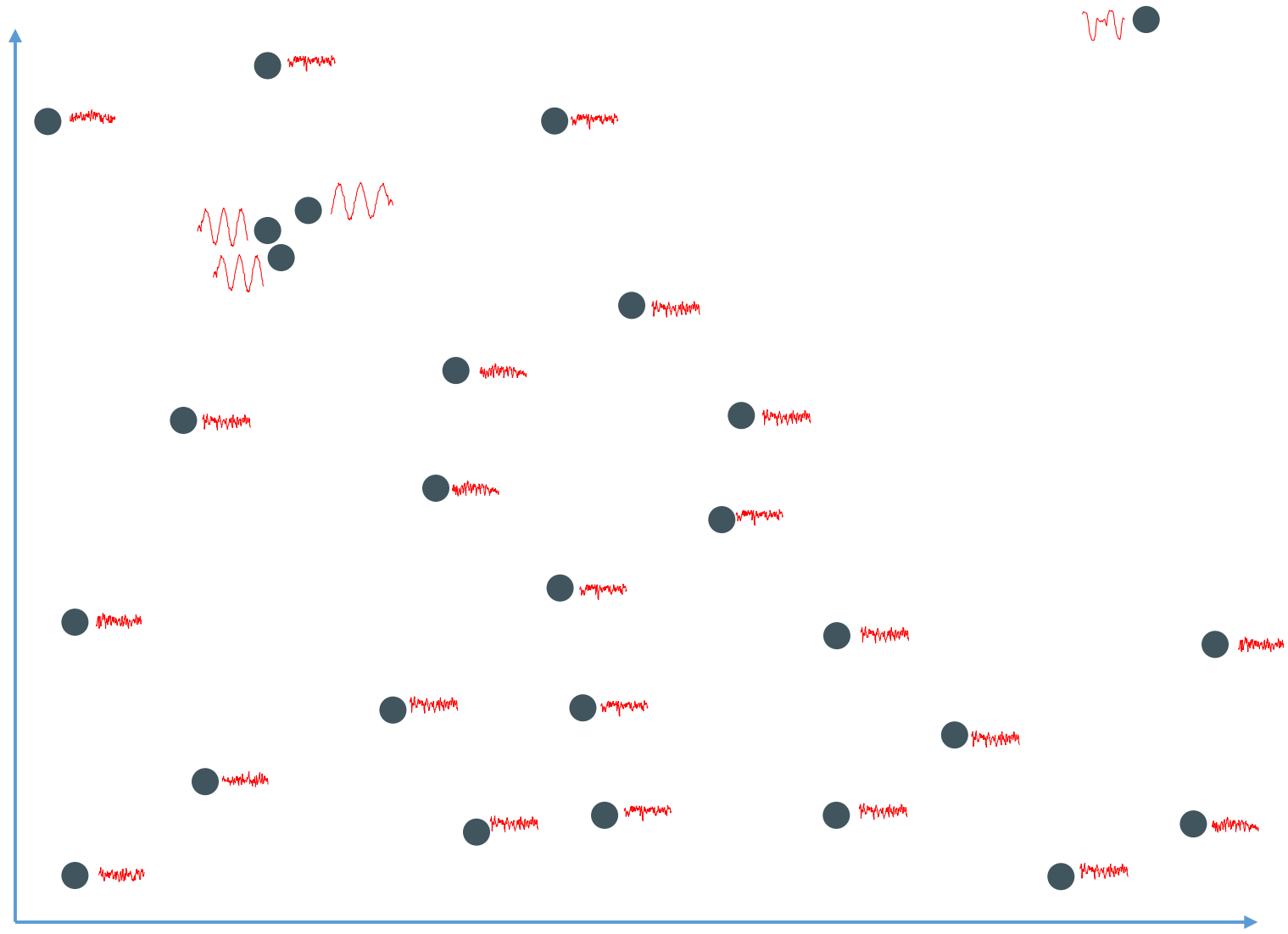
For discrete strings, *conserved* is easy to define, for example `papa =*a*a`. For *time series* it requires a distance function, here we will use Euclidean Distance.

* Bengali: Bābā	* Norwegian : papa
* Mandarin : baba	* Spanish : papá
* Polish : tata	* Swahili : baba
* Swahili : baba	* English : papa
* Turkish : baba	* Hindi : papa
* Xhosa: -tata	* Indonesian : bapa

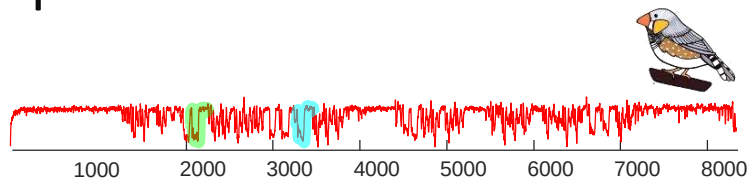
en.wikipedia.org/wiki/Mama_and_papa

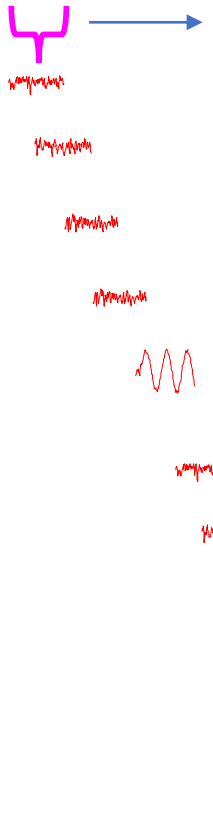
Dolphin …ACATTATCCGCATCTATA…
Hippo …ACATTATCGGCATCTATA…
Pig …TCTATCGCAATTTAATCA…

A Minor Visual Mapping Trick

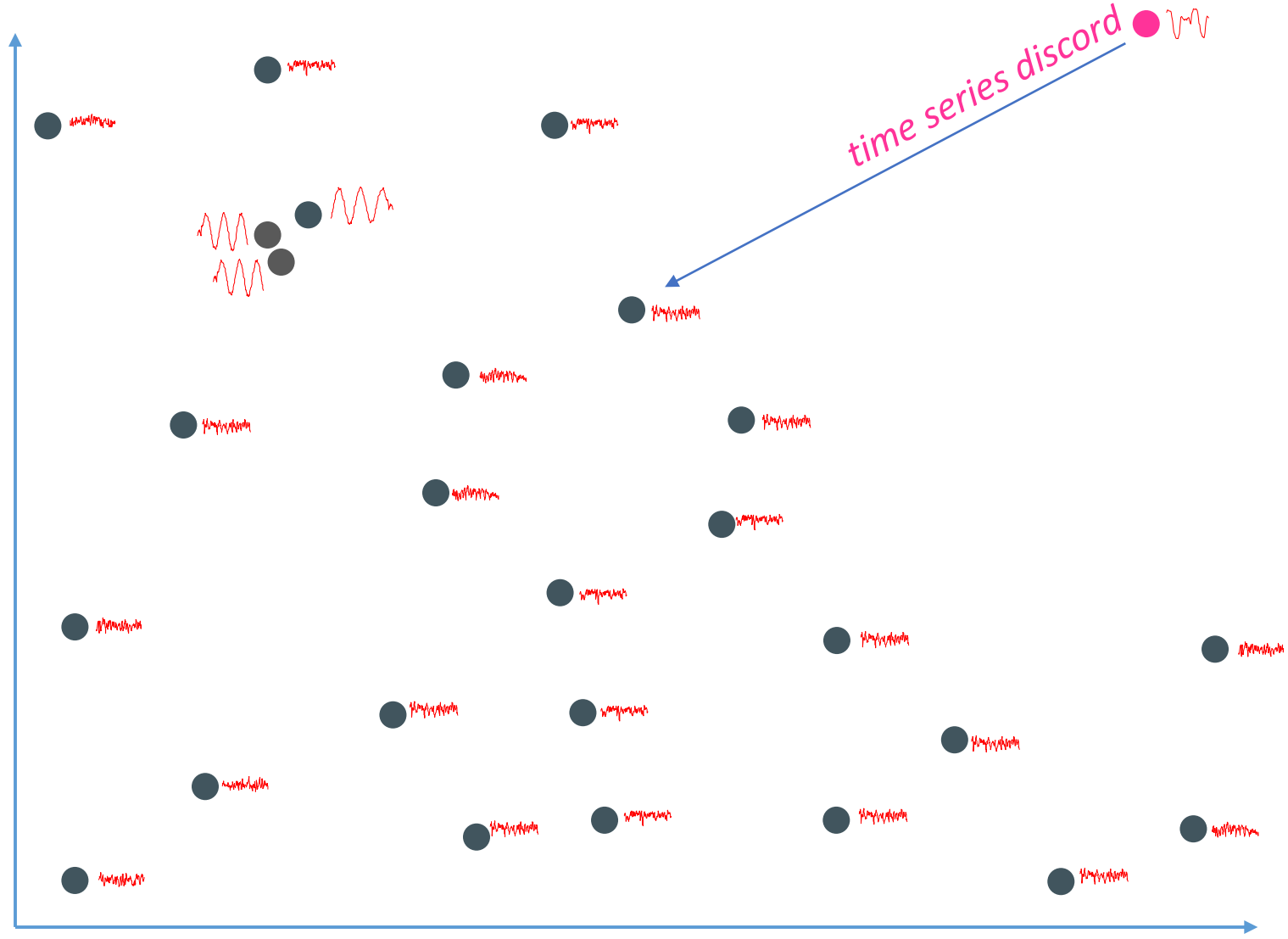


It is sometime useful to think of time series subsequences as points in *m*-dimensional space.





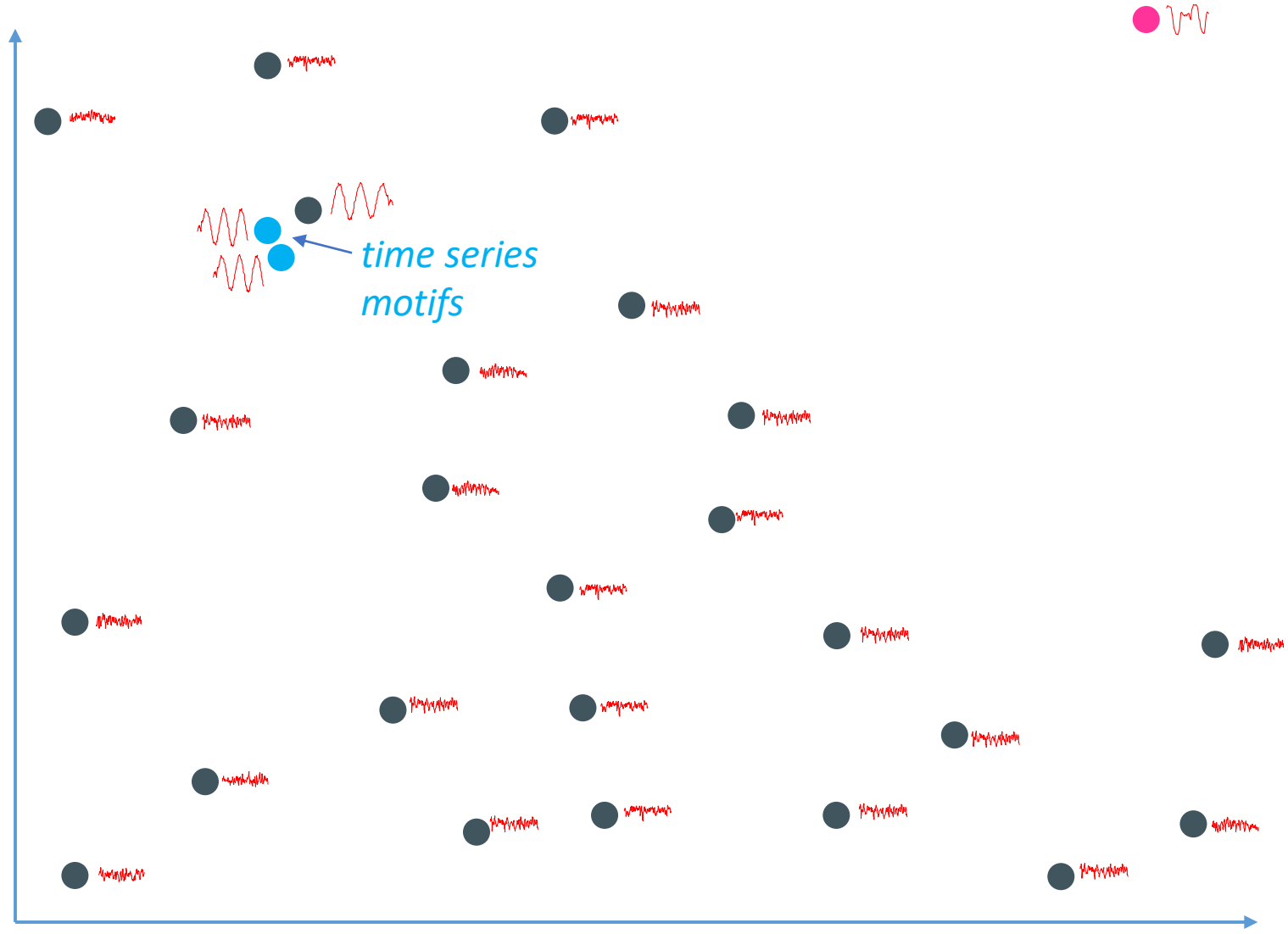
Recall this view of a time series…



There are several special elements here that are worth naming and discussing.

The point that is furthest from its nearest neighbor, is called a *time series discord*.

Recall this view of a time series…



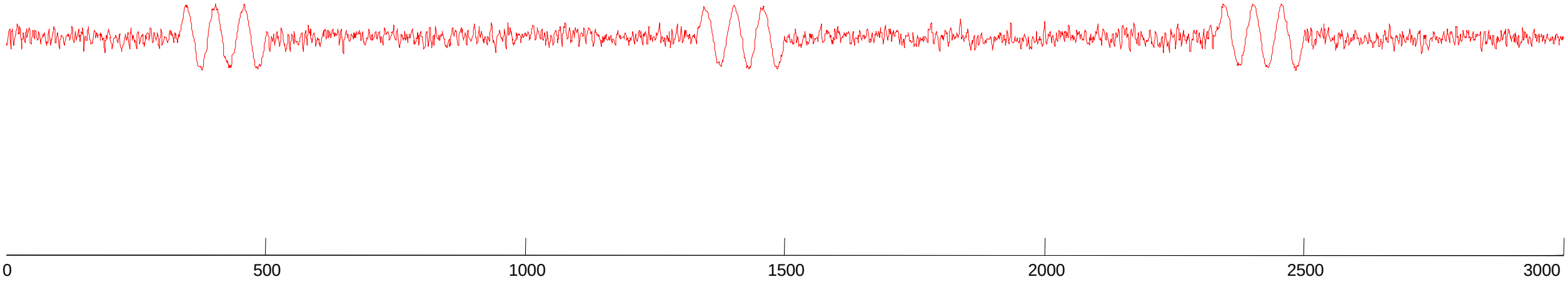
There are two special elements here that are worth naming, and discussing.

The point that is furthest from its nearest neighbor, is called a *time series discord*.

The pair of points that are closest together are call *time series motifs*.

We can use a data structure called the *Matrix Profile* to reason about *discords* and *motifs*

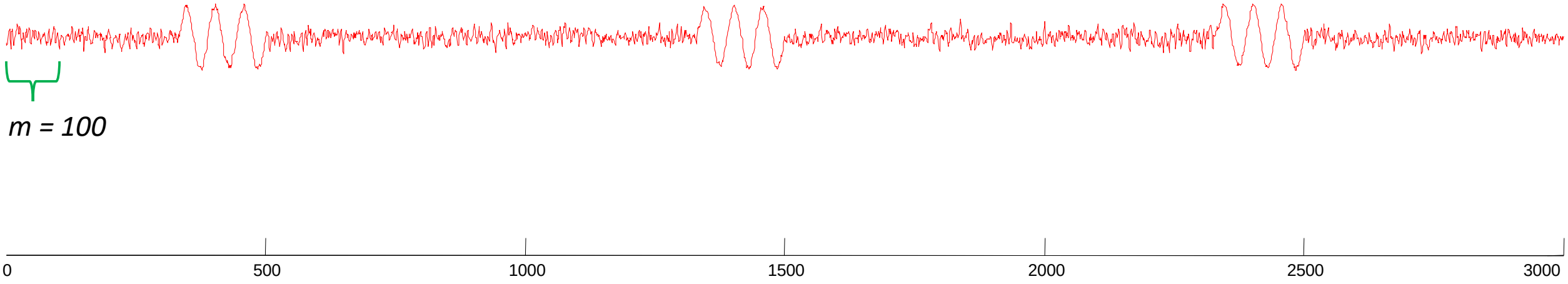
Intuition behind the Matrix Profile: Assume we have a time series *T*, lets start with a synthetic one...



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

Note that for most 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*

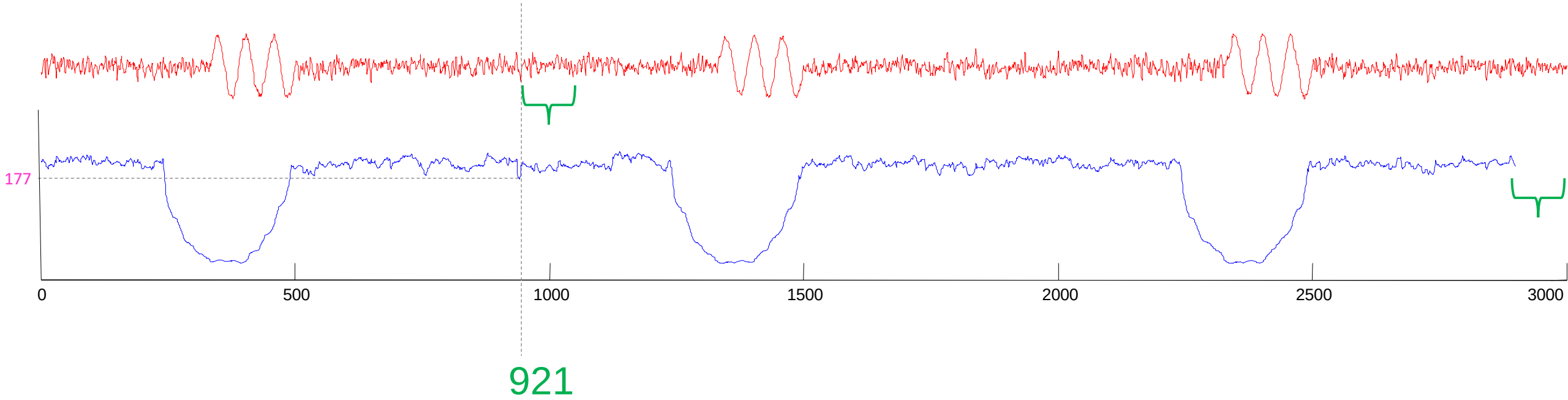
These subsequences might be about the length of individual heartbeats (for ECGs), individual days (for social media behavior), individual words (for speech analysis) etc



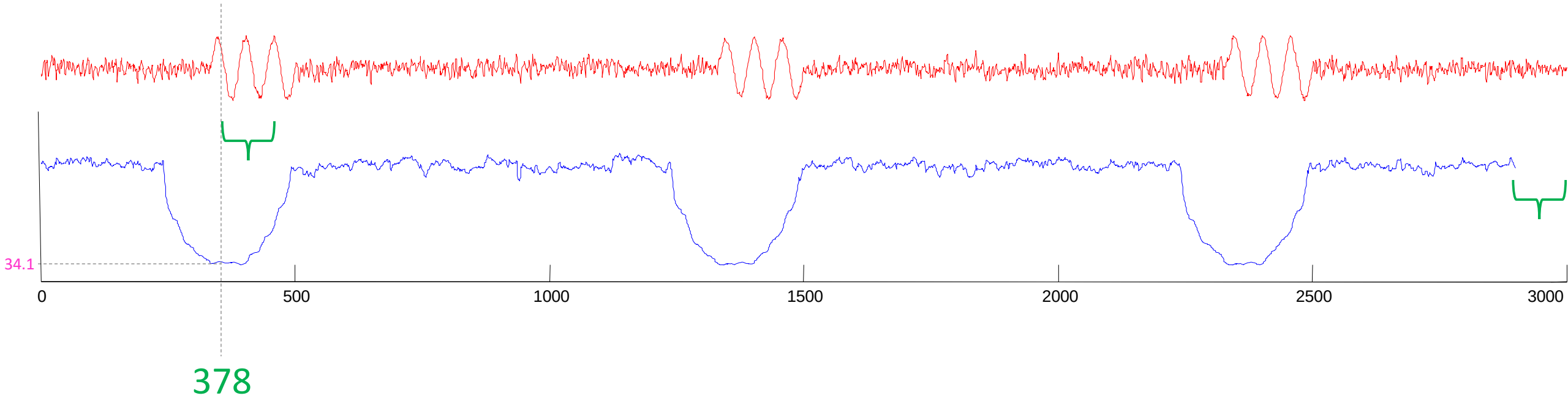
We can create a companion “time series”, called a Matrix Profile or MP.

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.

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).

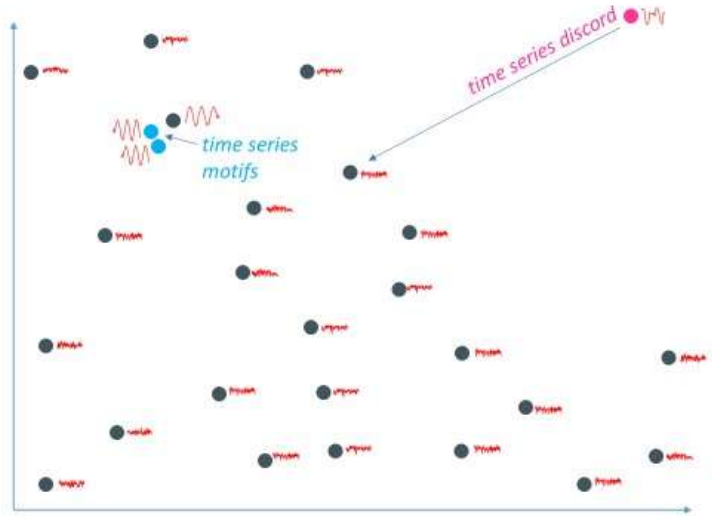


Another example. In the below, the subsequence starting at 378 happens to have a distance of 34.2 to its nearest neighbor (wherever it is).



These two views of the world are equivalent

Recall this view of a time series…



There are two special elements here that are worth naming, and discussing.

The point that is furthest from its nearest neighbor, is called a *time series discord*.

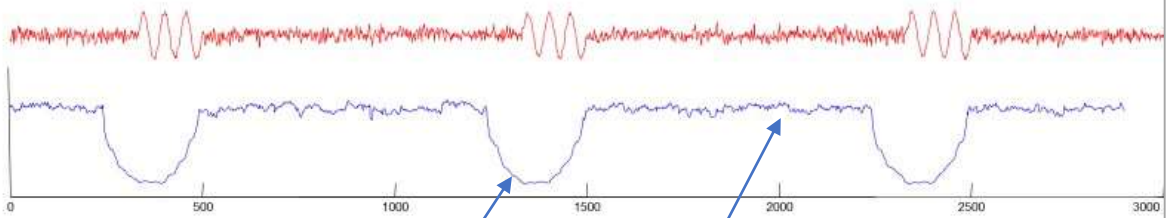
The pair of points that are closest together are call *time series motifs*.

We will use a data structure called the *Matrix Profile* to reason about discords and motifs

For the rest of this workshop….

The Matrix Profile is always shown in blue.

The real time series data, is generally shown in red.



time series motifs correspond to low values in the matrix profile

time series discords correspond to high values in the matrix profile

The Matrix Profile has exploded in popularity in the last 3 years

observations of the magnetosphere collected by the Cassini spacecraft in orbit around Saturn.. ..in this case, the best-performing method was the Matrix Profile.. Kiri L. Wagstaf et. al. NASA JPL.2020

(for an industrial IoT problem) *Matrix Profiles perform well with almost no parameterisation needed.* Anton et al ICDM 2018.

While there will never be a mathematical silver bullet, we have discovered that the Matrix Profile, a novel algorithm developed by the Keogh research group at UC-Riverside, is a powerful tool. Andrew Van Benschoten, lead engineer at Target.

If anybody has ever asked you to analyze time series data and to look for new insights then (the Matrix Profile) is definitely the open source tool that you'll want to add to your arsenal Sean Law, Ameritrade.

(for) intrusion detection in industrial network traffic, distances as calculated with Matrix Profiles rises significantly during the attacks. ..as a result, time series-based anomaly detection methods are capable of detecting deviations and anomalies. Schotten (2019).

The MatrixProfile technique is the state-of-the-art anomaly detection technique for continuous time series. Bart Goethals et. al. (ECML-PKDD 2019).

Based on the concept of Matrix Profile ..without relying on time series synchronization.. the Railway Technologies Laboratory of Virginia Tech has been developing an automated onboard data analysis for the maintenance track system Ahmadian et. al. JRC2019

Matrix Profile is the state-of-the-art similarity-based outlier detection method. Christian Jensen et. al. IJCAI-19

we use the exact method based on the Matrix Profile (to assess the effectiveness of therapy) Funkner et al Procedia 2019.

Recently, a research group from UCR have proposed a powerful tool - the Matrix Profile (MP) as a primitive...(we use it for) *fault detection* Jing Zhang et al. ICPHM 2019

Inspecting both graphs one can see that the matrix-profile algorithm was able to identify regions where there is a change on the power level over the observed band. F Lobao 2019.

RAMP builds upon an existing time series data analysis technique called Matrix Profile to detect anomalous distances...collected from scientific workflows in an online manner. Herath et. al. IEEE Big Data 2019

Based on obtained results for the considered data set, matrix profiles turned out to be most suitable for the task of anomaly detection Lohfink et al. VISSEC2019

The computation speed and exactness of the Matrix Profile make it a powerful tool and (our) results back this. Barry & Crane AICS 2019

(examining) *manufacturing batches considering raw amperage* (we found that the) *Matrix Profile highlights anomalies* Hillion & O'Connell of TIBCO Data Science. re:Invent 2019. [

we use the exact method based on the matrix profile to search for motifs can be used to monitor the patient's condition, to assess the effectiveness of therapy or to assess the physician's actions. Funknera et al.

(The Matrix Profile is a) *similarity join to measure the similarity between two given sequences. we opt for the median of the profile array as the representative distance* (3D Dancing Move Synthesis from Music)" Anh et al. IEEE Robotics and Automation Letters

We were amazed by the power of MP and seek to incorporate it into our framework Ye and Ageno.

..adopting the concept of (the) Matrix Profile, we conduct the first attempt to.. J. Zuo et. al. Big Data 20019

The accuracies obtained ...indicate that the Matrix Profile is useful for the task at hand instead of using the CNN features directly Dhruv Batheja

To speed up online bad PMU data detection a fast discovery strategy is introduced based on (the Matrix Profile) Zhu and Hill.

Specifically, ALDI uses the matrix profile method to quantify the similarities of daily subsequences in time series meter data, Zoltan Nagy, Energy & Buildings (2020)

Our two-fold approach first leverages the Matrix Profile technique for time series data mining.. Nichiforov 2020.

*the class of matrix profile algorithms.. ..is a promising approach, as it allows simplified post-processing and analysis steps by examining the resulting matrix profile structure*A. Raoofy et al.

We only require information about the time of several critical incidents to train our methods, as previously. To this end, we employ the Matrix Profle.. Bellas. et al.

a matrix-profile based algorithm applied across all trajectory data against a validation set revealed four significant motifs which we defined as motif A, B, C and D.. Fernandez Alvarez 2020.

The main building block of this (game analytics) algorithm is the matrix profile, Saadat and Sukthankar AAAI2020

We leverage the Matrix Profile (MP), ... to create a micro-service-based machinery monitoring solution Naskos et al 2021

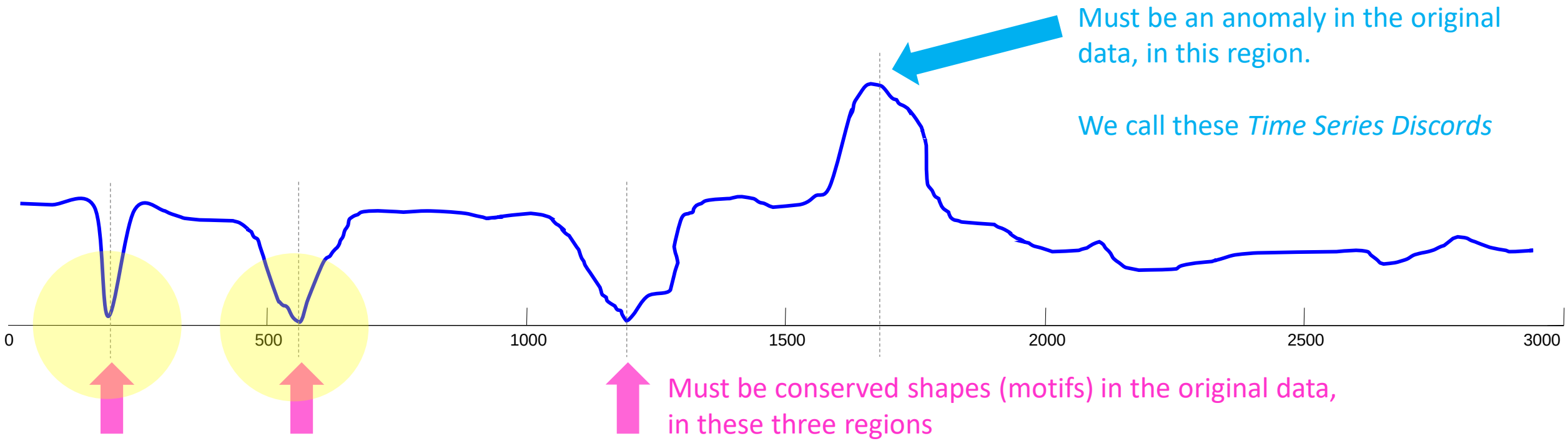
SLMAD uses statistical-learning and employs a robust box-plot algorithm and Matrix Profile (MP) to detect anomalies Team from Huawei/UCD.

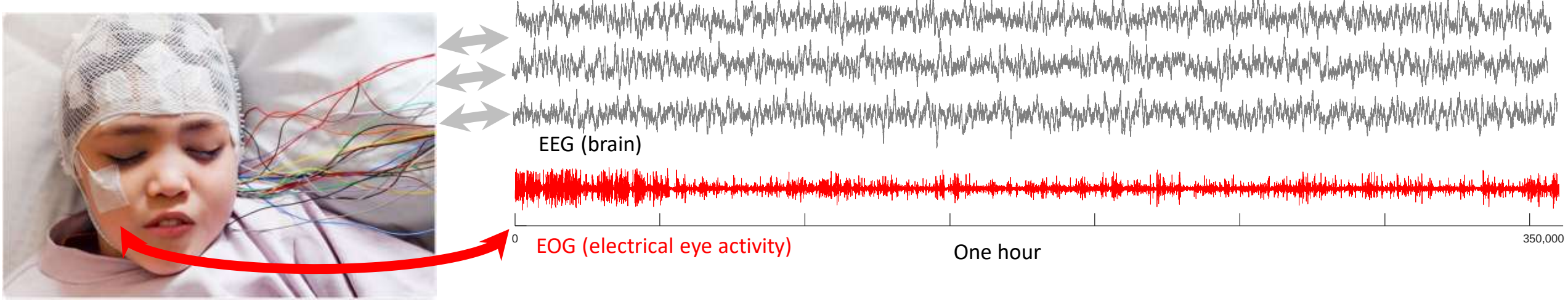
We found that all these similarity or randomness measures can be estimated with variants of the highly efficient Matrix Profile (MP) algorithm. `

Reading the 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).



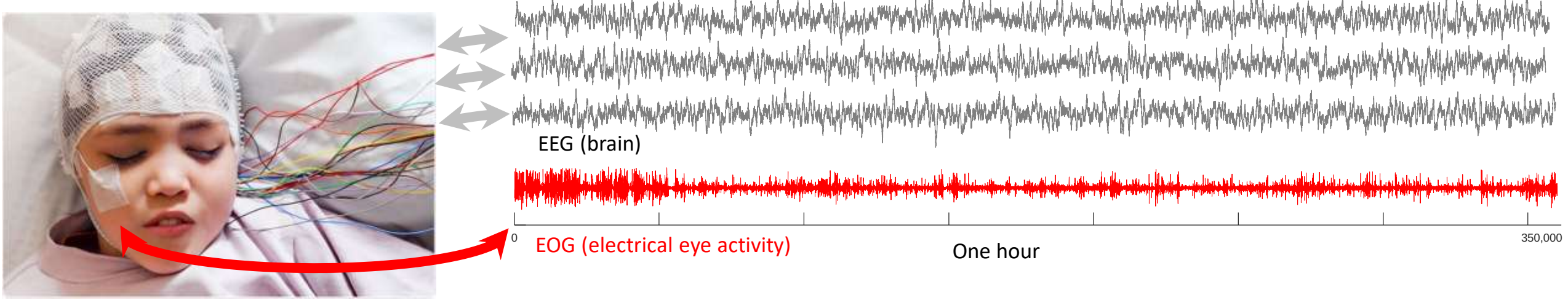


Let's jump into a case study

We have one hour of data, 64 EEG traces and one EOG trace.
The neuroscientists want to do some analysis on EEGs.
However, they don’t want to consider any time periods in which there are eye blinks.
So, we need to search the EOG time series for any eye-blinks, and record their locations.

Problem

- Eye-blinks can vary from person to person
- Even for a single person, the placement of the sensor will change what the blinks look like, so you can't directly compare shapes day to day
- For most people, the shape of the blink can be polymorphic



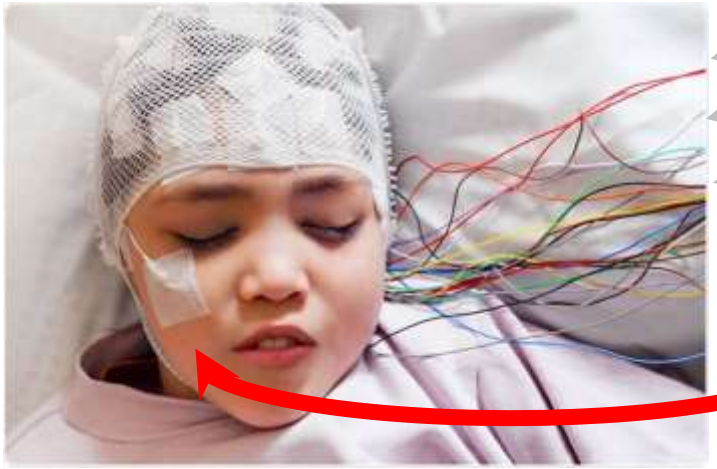
Insight

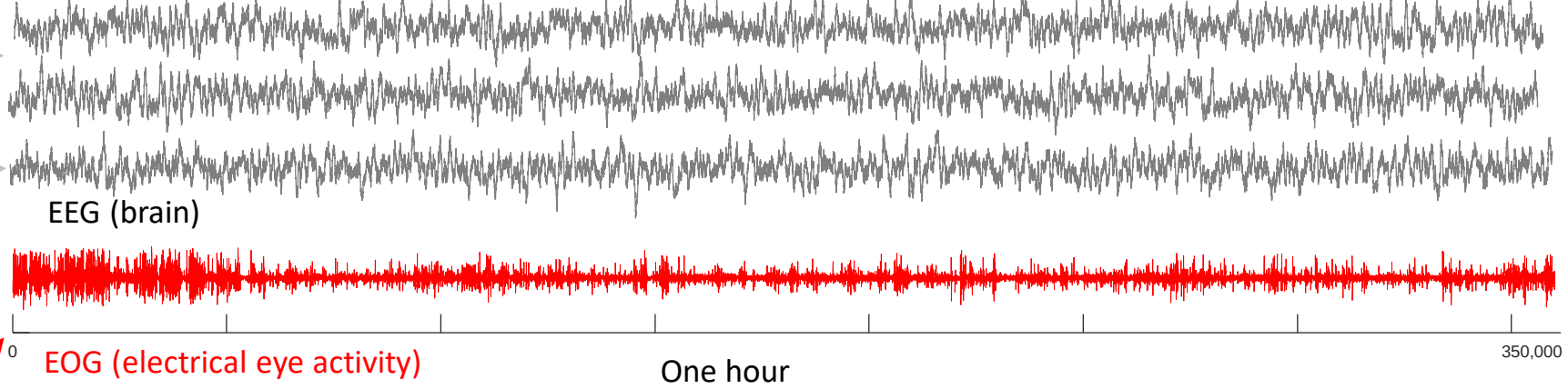
While we do not know the shape of an eyeblink, we do know that they tend to be **strongly conserved shapes**, in any given session.

So, we can run motif discovery, and the motifs are likely to be eye blinks, which we can then search for (using subsequence search) and delete the corresponding regions from the EEG. Lets run the code….

```
>> load eog_sample.mat
>> [matrixProfile profileIndex, motifIndex, discordIndex] = interactiveMatrixProfileVer3_website(eog_sample, 400);
```

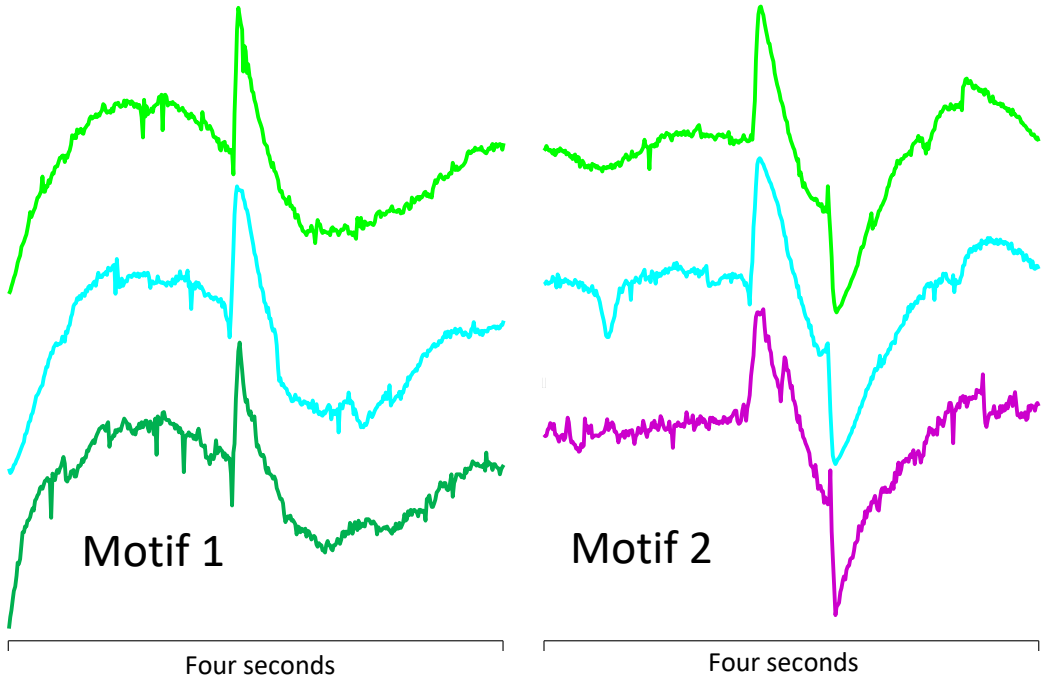
…and see if it worked (next slide)



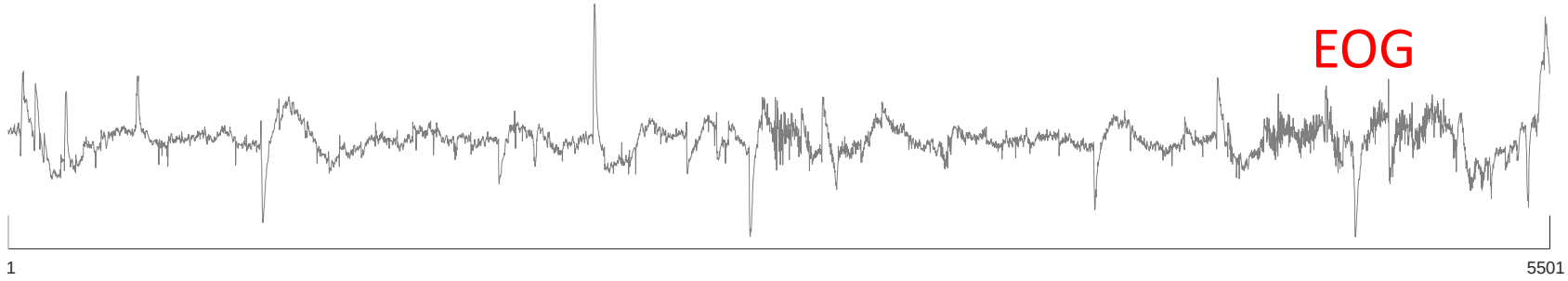


It worked perfectly!
The top-2 motifs are eye-blinks
The top-3 motif (not shown) is much less well conserved, and ignored.

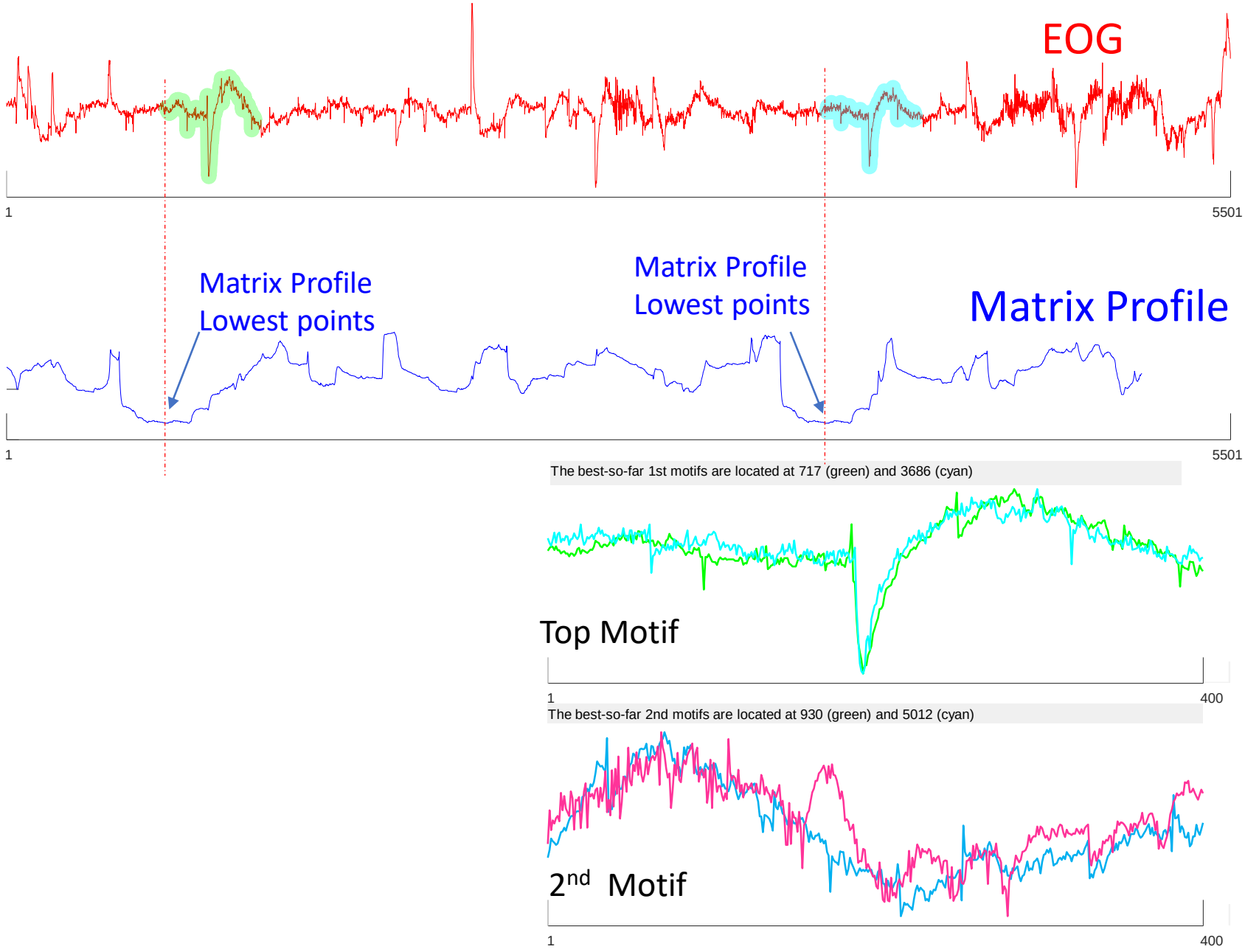
Exhaustive search for motifs in a dataset this size would take tens of minutes. (There are 6.125×10^{10} pairs that *could* be the best motif), but MP algorithms gives you these motifs in a second.



Let's show plots on a tiny subset of the data to get a better visual intuition.



Let's show plots on a tiny subset of the data to get a better visual intuition.



Let’s consider a new case study using motifs

Informal definition: Weakly labelled data

Weakly labelled data is data that is labeled, but imprecisely with regard to timing

For example

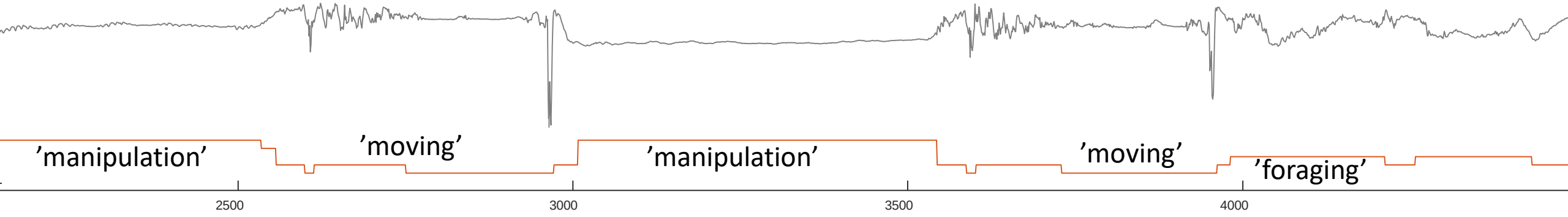
- *Sometime between April and June, the machine broke down.*
- *The machine always overheats within one or two hours of changing the infeed mix*.

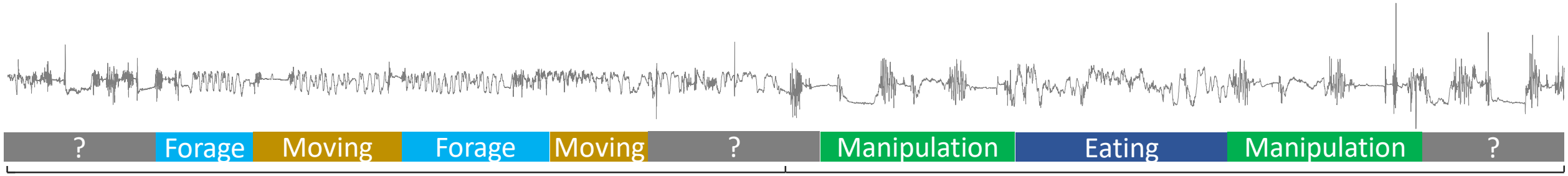
We want to go from these imprecise labels, to *exact* locations/shapes of behavior.

Intuition, find motifs that occur in the weakly labeled time regions, and then use similarity search to see if they are unique to those regions.

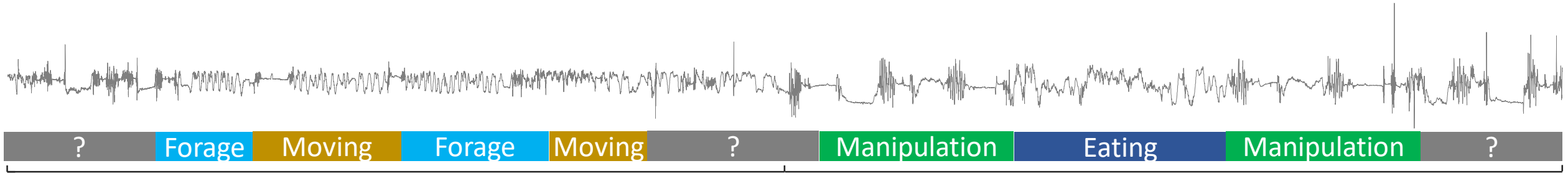
We have many hours of weakly labelled data of seal behavior.
It is weakly labelled because most of the labels are provided by human observers, paper-and-pencil with a wristwatch timings.
Semi-wild conditions mean the seals are not always fully observable.

Can we convert this to *exact* labels?





Here is a sample of data.

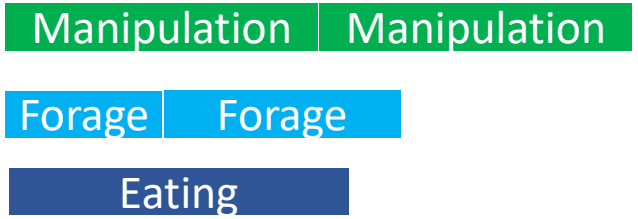


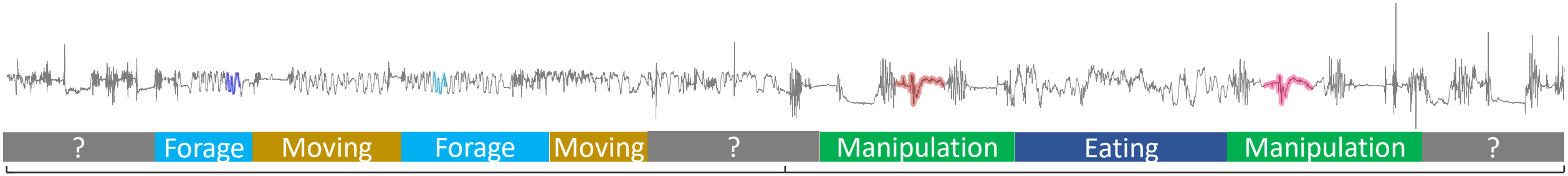
Here is a sample of data.

Let us try a simple idea.

Lets divide the data into the labeled sets, then look for motifs.

If we find nice motifs, we can then check to make sure that they are unique to the given label.



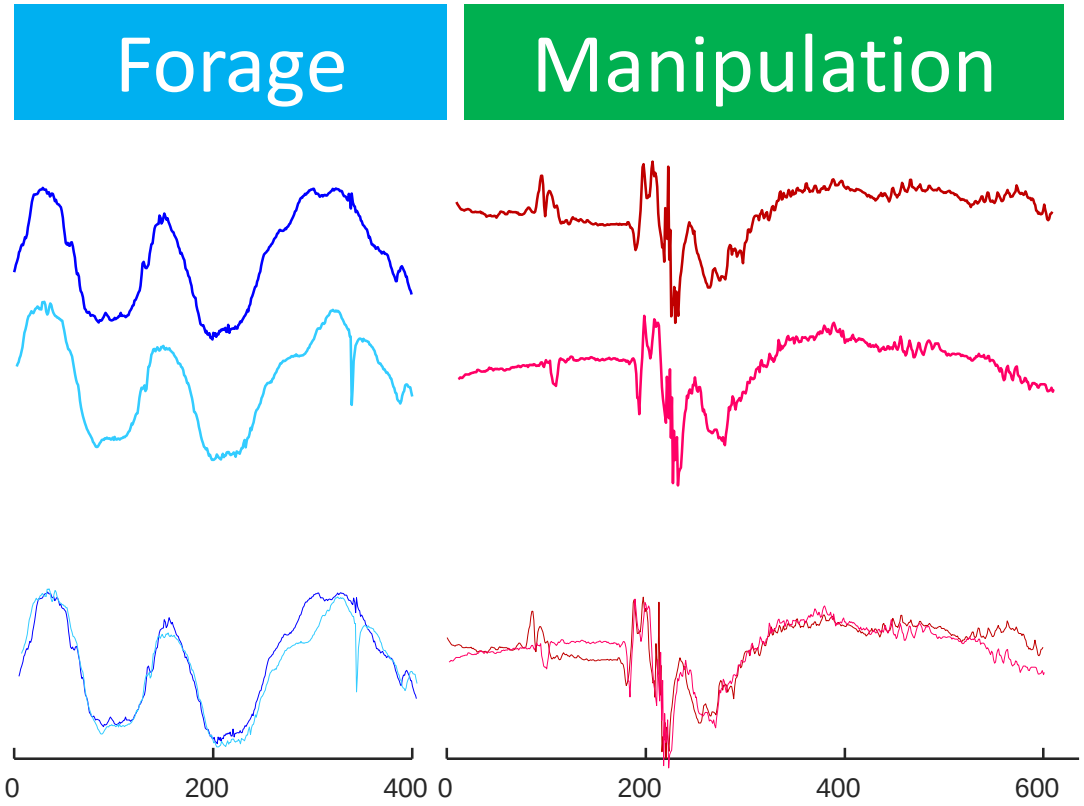


(we have hand waved over a few details)

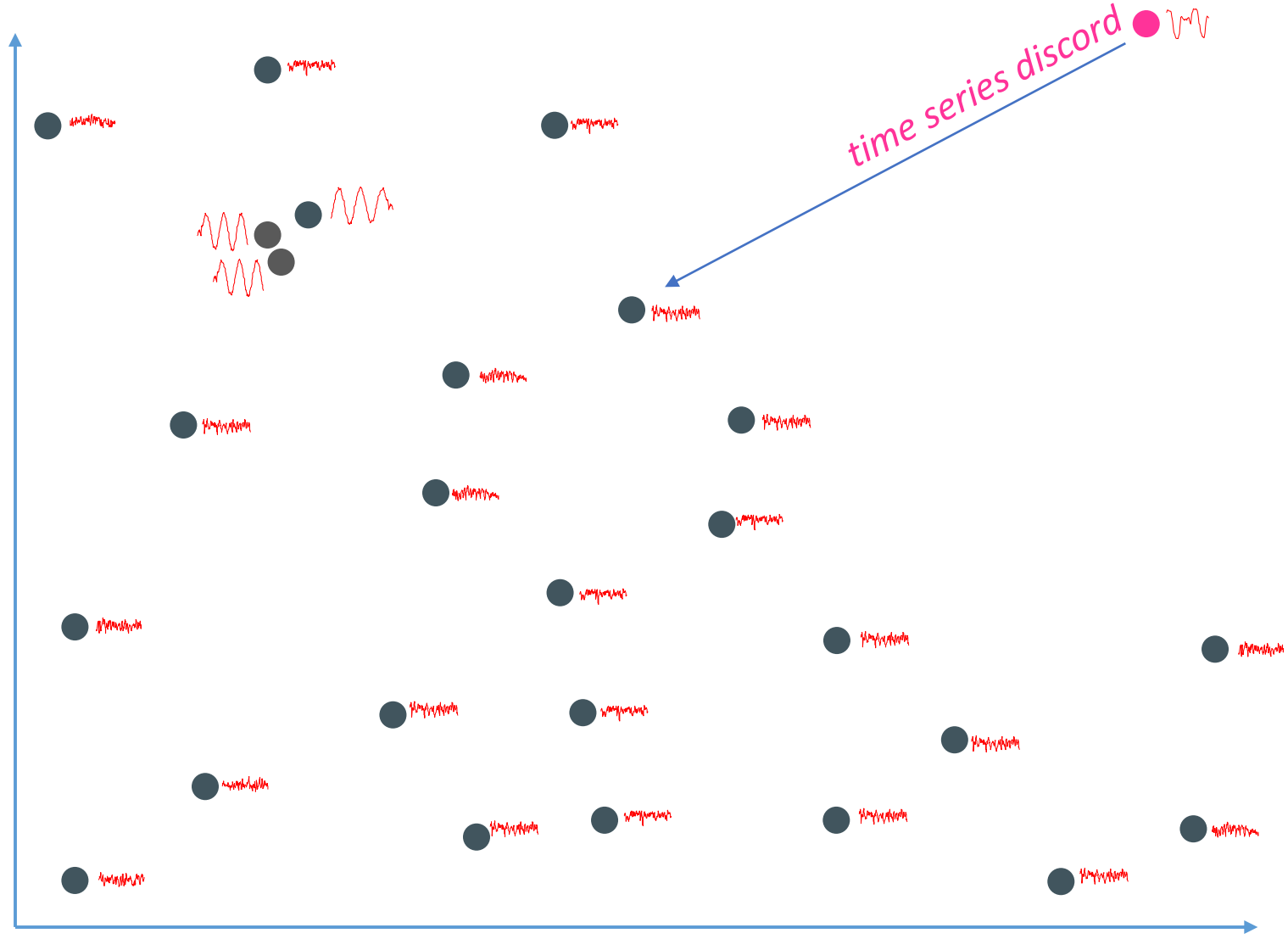
This works *very* well

We confirmed the “signatures” discovered with hold out data.

Video of Manipulation
https://vimeo.com/148414126



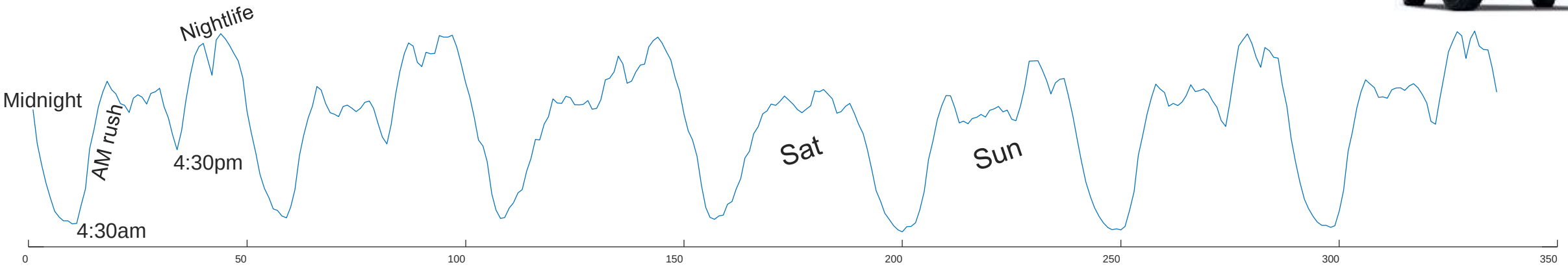
Recall this view of a time series…



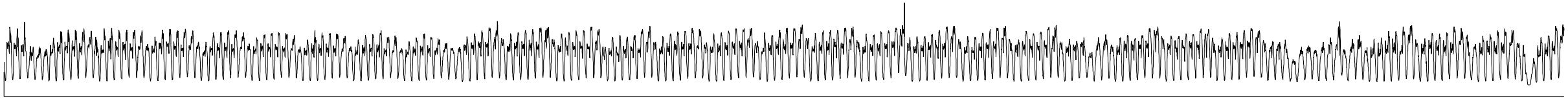
The point that is furthest from its nearest neighbor, is called a *time series discord*.

Time series discords are excellent anomaly/novelty detectors..

New York makes all its taxi information public.
Here is a random week



New York Taxi Demand



Given a large chunk of it, how can we make sense of it?

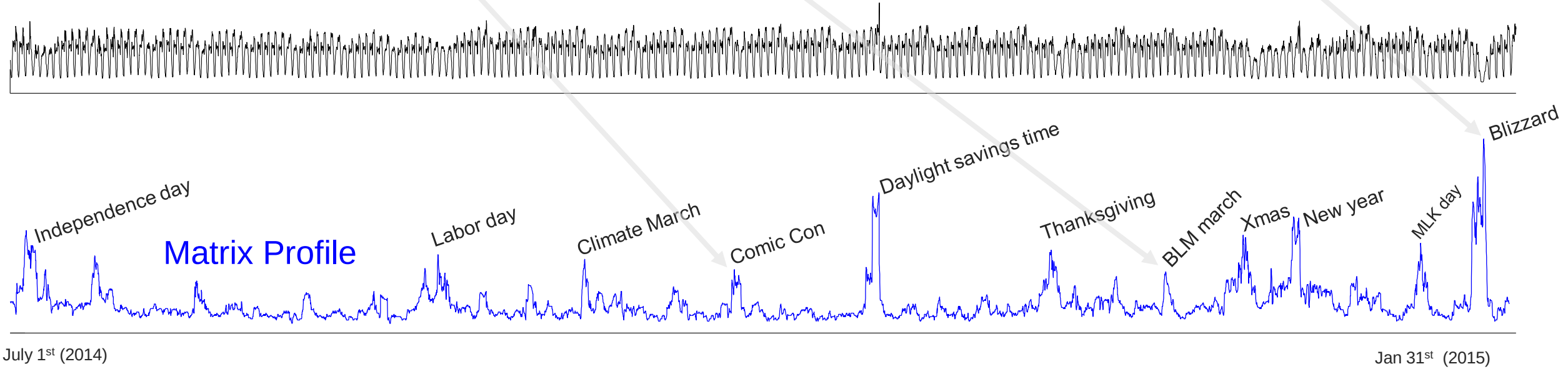
One idea is to compute the Matrix Profile, to find the most unusual patterns….

It is nice that the convention is growing to cover all sorts of fandoms inside this world. .... However, that would just be another day of traffic jams and over population in the city.

More than 25,000 people marched through Manhattan on Saturday, police officials said, in the largest protest in New York City since a grand jury declined this month to indict an officer in the death of an unarmed black man on Staten Island.

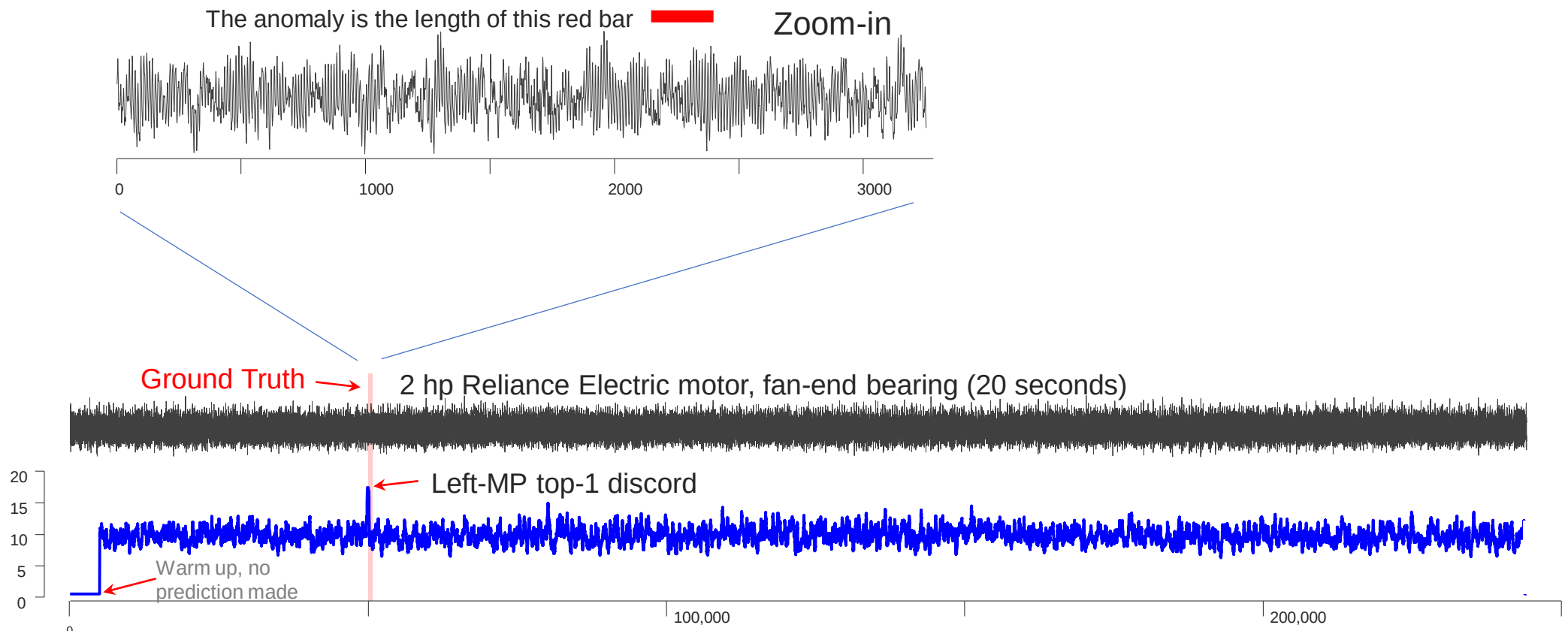
…Subway and bus service were suspended, and the Port Authority of New York and New Jersey closed Hudson River crossings. Thousands of flights were grounded, public transportation was suspended or curtailed, and travel bans were put in place in the half-dozen states in the path of the storm.

New York Taxi Demand

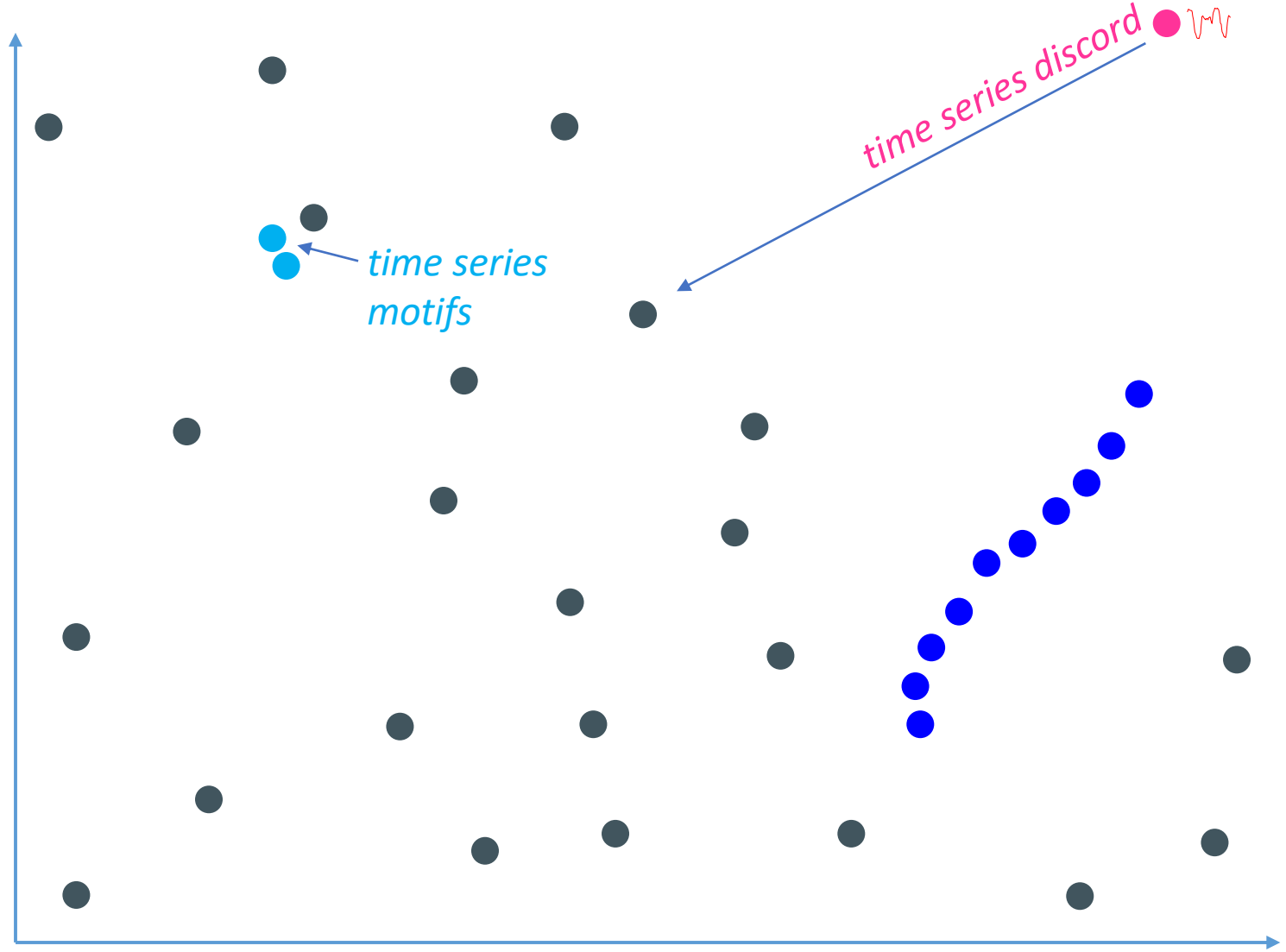


Time series discords have two astonishing and unique properties

1) They are superhuman (they can find anomalies that you cannot)
2) They are fast, 300,000 Hz on an old laptop (so we can handle the 12,000 Hz data below in real time)



Back to this view of time series once more



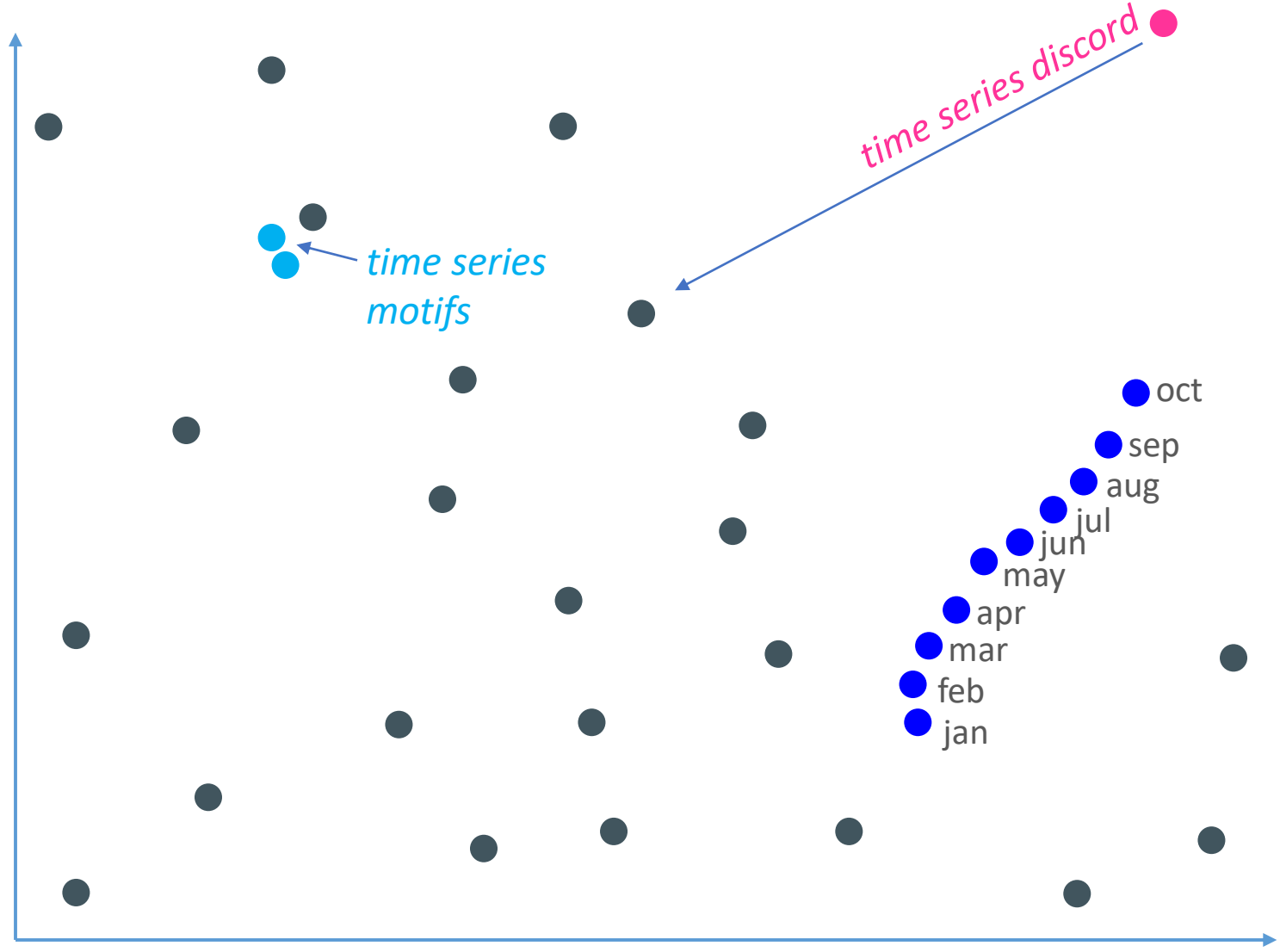
We have seen *time series discords* and *time series motifs*.

There is yet another type of pattern we can consider, time series chains.

We can also find these using a (slightly modified) *Matrix Profile.*

Suppose you saw these blue points…

Back to this view of a time series once more



We have seen *time series discords* and *time series motifs*.

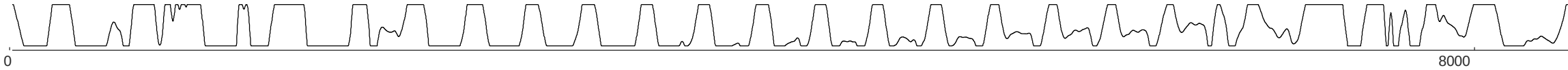
There is another type of pattern we can consider, time series chains.

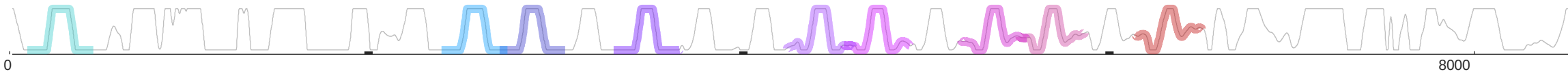
We can also find these using a (slightly modified) *Matrix Profile.*

Suppose you saw these blue points…
..lets annotate them by date of arrival
Where do you think November would be?

Case study

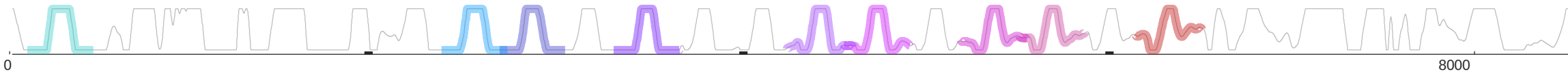
- Are there evolving patterns (chains) in this dataset of respiration?



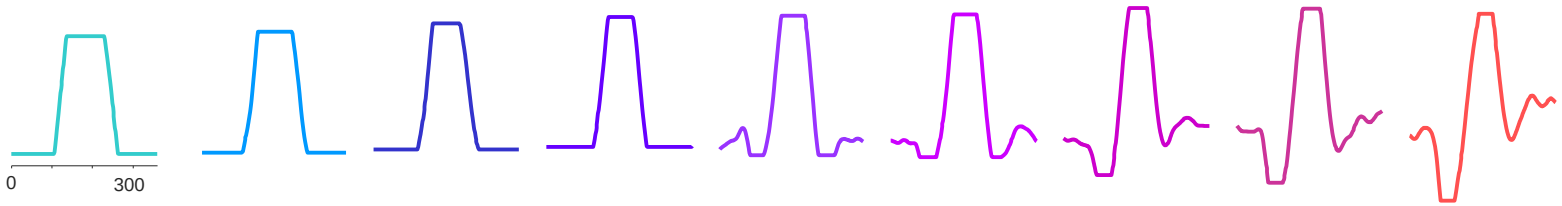
- 

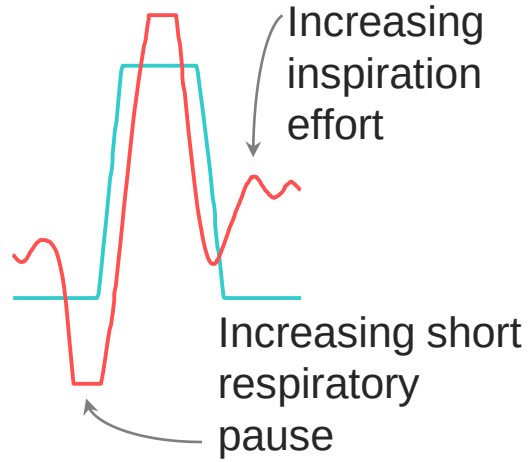
- Here is one, with nine links.

Dr. Greg Mason, an expert on cardiopulmonary interactions, says that the chain appears to be "*attempts to inspire against an obstruction coming the back of the tongue*".

- 

- Chains often have a *physical* interpretation, including this one.

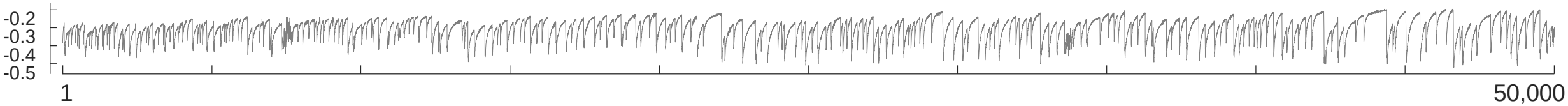




Entomology

This is a dataset reflecting the behavior of an insect, as it feeds on citrus.

Asian citrus psyllid

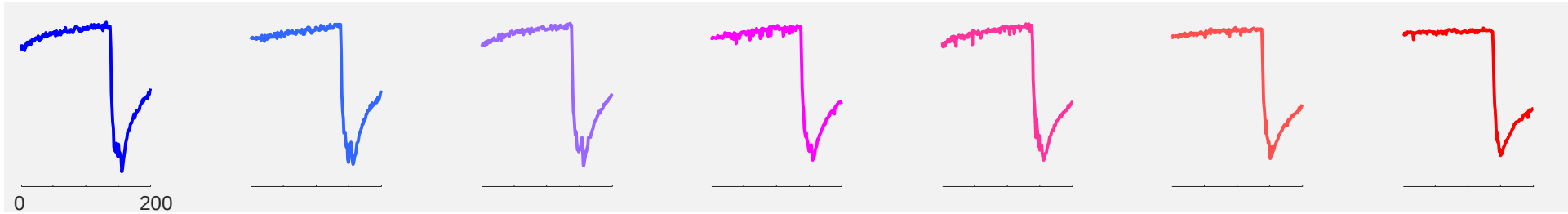


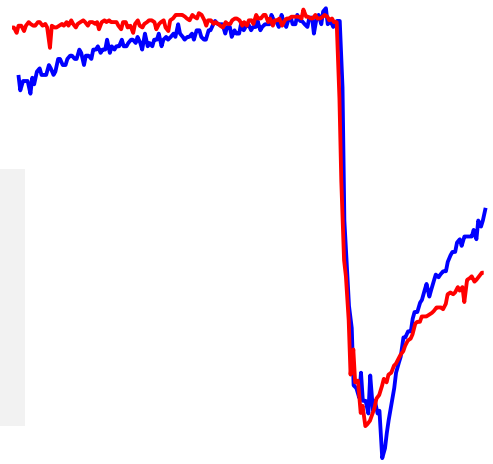
Entomology

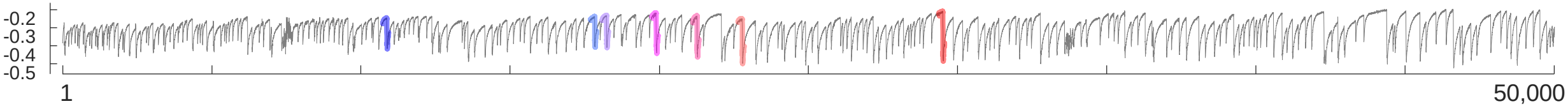
The chain discovered has a (tentative) physical meaning..

Asian citrus psyllid

The gradual plateauing of the peak corresponds to the slow exhausting the vein during a session of phloem sap ingestion

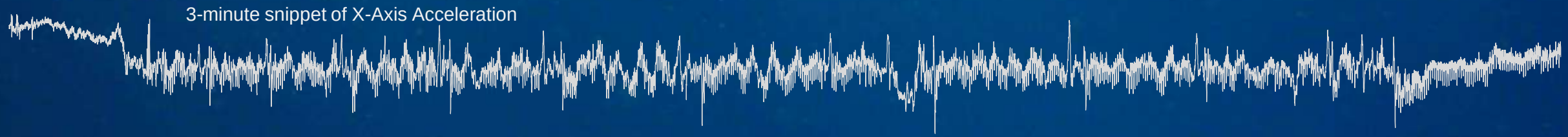






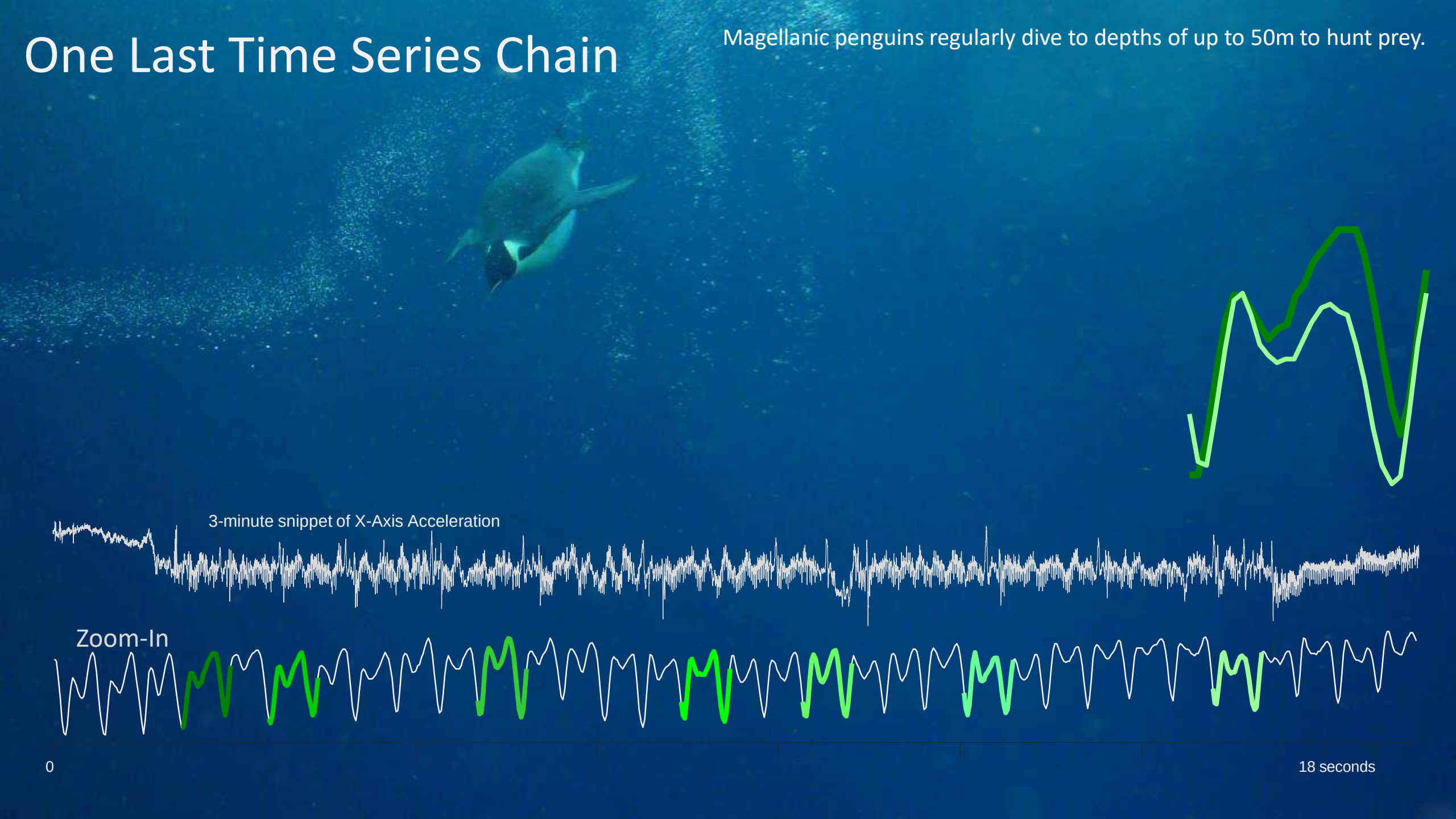
One Last Time Series Chain

Magellanic penguins regularly dive to depths of up to 50m to hunt prey.



One Last Time Series Chain

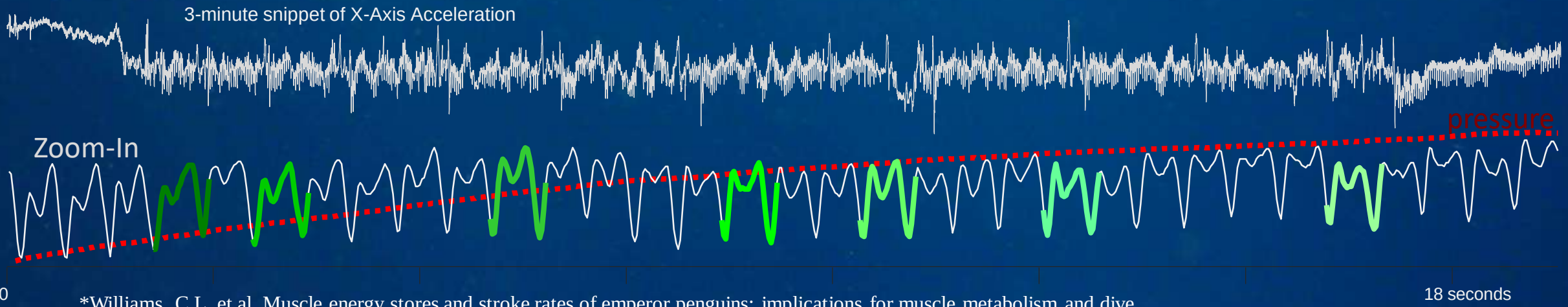
Magellanic penguins regularly dive to depths of up to 50m to hunt prey.



One Last Time Series Chain

Magellanic penguins regularly dive to depths of up to 50m to hunt prey. Penguins have typical body densities for a bird, but just before diving they take a very deep breath that makes them exceptionally buoyant. This positive buoyancy is difficult to overcome near the surface, but at depth, the compression of water pressure cancels it. In order to get to down to their hunting ground below sea level it is clear that "*locomotory muscle workload, varies significantly at the beginning of dives*"*.

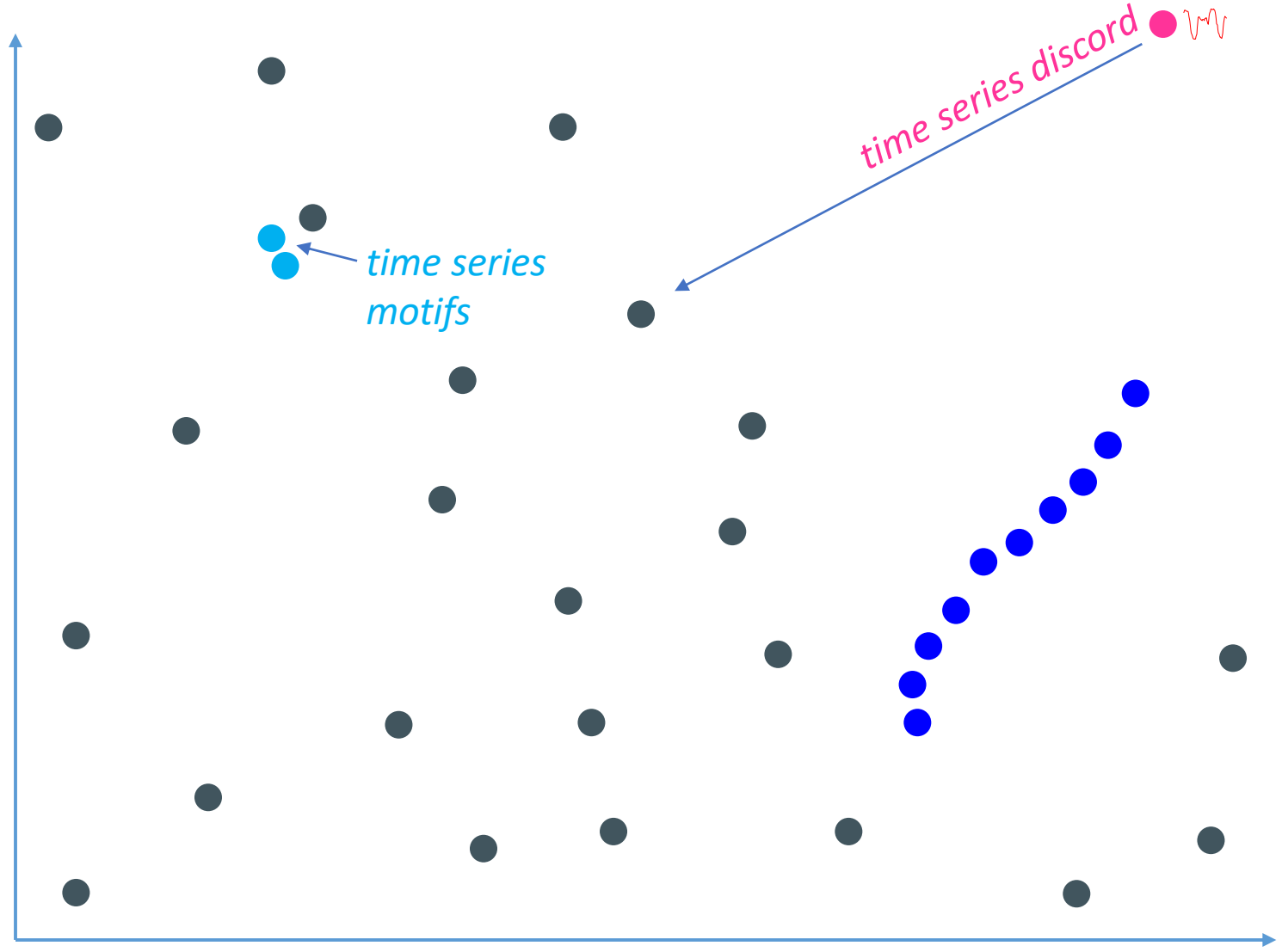
The snippet of time series shown in does not suggest much of a change in *stroke-rate*, however penguins are able vary the thrust of their flapping by twisting their wings. The chains we discovered shows this dramatic and evolving sprint downwards leveling off to a comfortable cruise.



*Williams, C.L. et al. Muscle energy stores and stroke rates of emperor penguins: implications for muscle metabolism and dive performance. *Physiological and Biochemical Zoology.85.2*(2011):120-133

Photo by Paul J. Ponganis

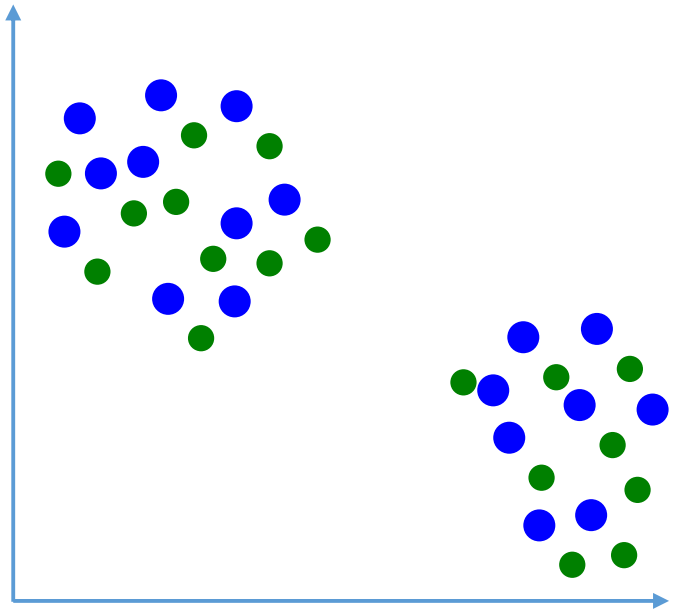
Once again, back to this view of time series

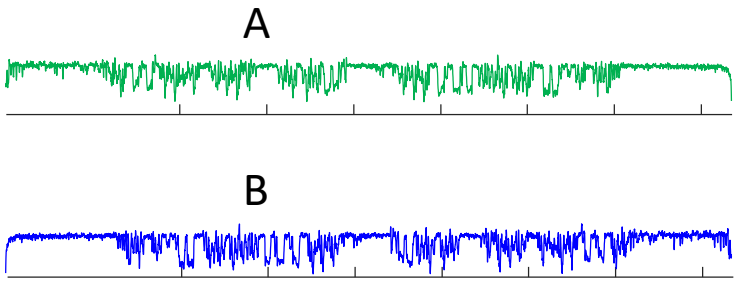


We have seen *time series discords*, *time series motifs* and time series chains.

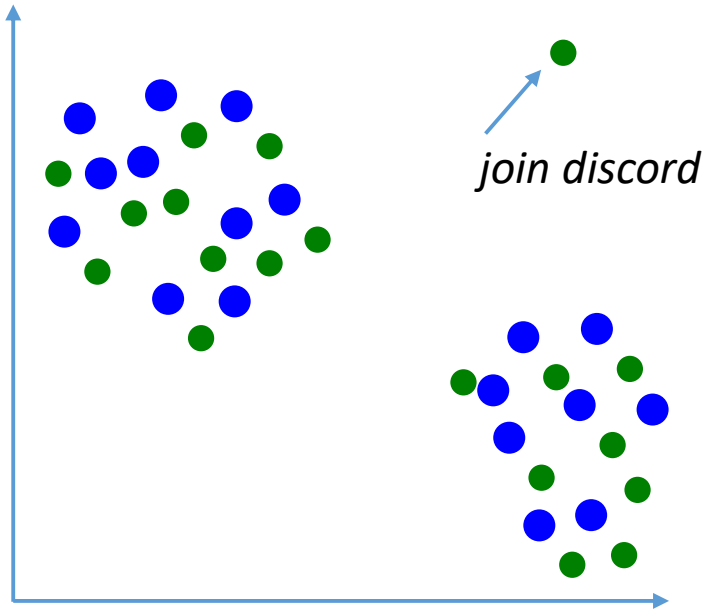
We can generalize even further…

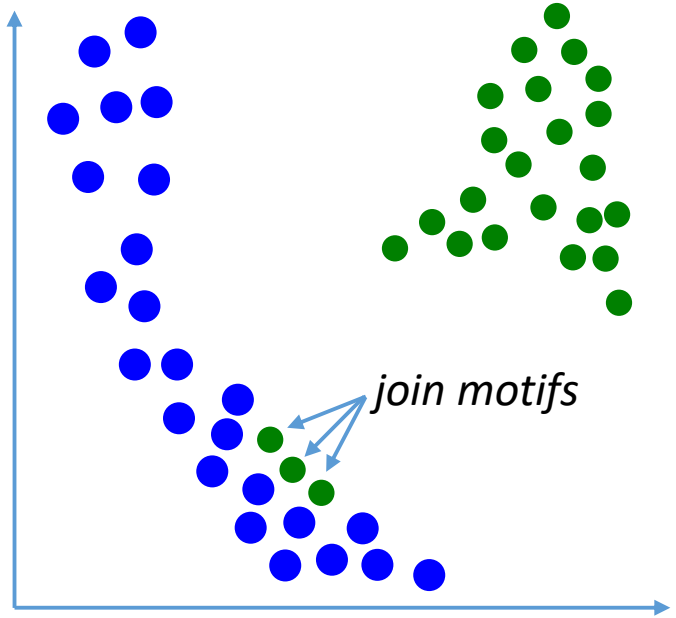
Generalizing to Joins





Generalizing to Joins





Two scenarios of interest: we do a $\mathbf{J}_{T_A T_B}$…

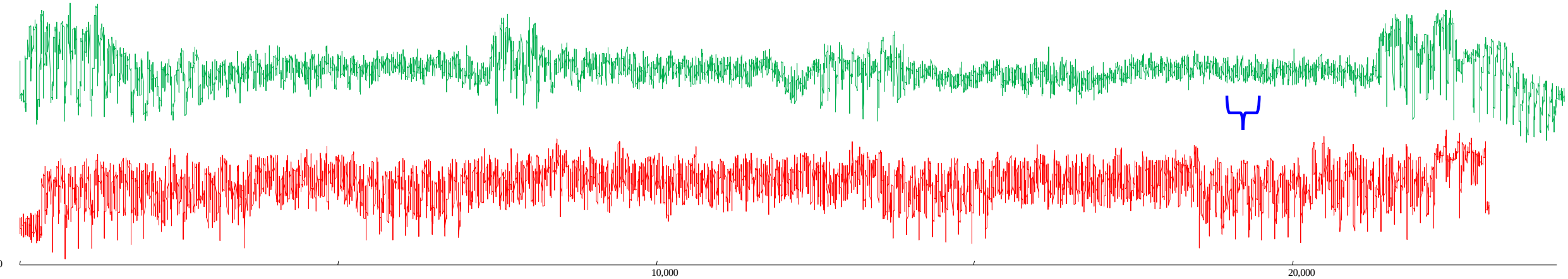
1) **The Golden Batch**: Here we have two time series that we think should be about the same. But when we join them, there is a *join discord*, a subsequence that appears only in only in **A**, but not in **B**, but why?

2) **The Suspicious Similarity:** Here we have two time series that we have *no* reason to think should be the same. But when we join them, there are *join motifs*, some subsequences from **A** appear in **B**, but why?

Join example

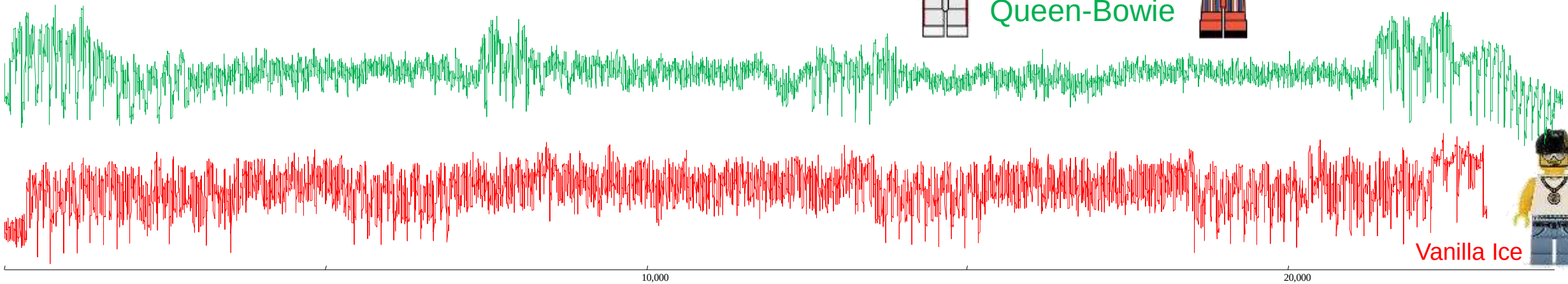
Can you see any common structure between the two time series below?

Hint, it is probably about this length 



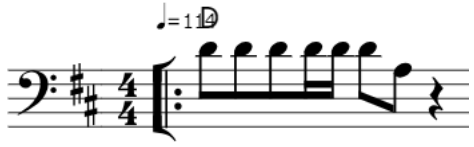
The data is the 2nd MFCC of two songs,
Under Pressure and *Ice Ice Baby*

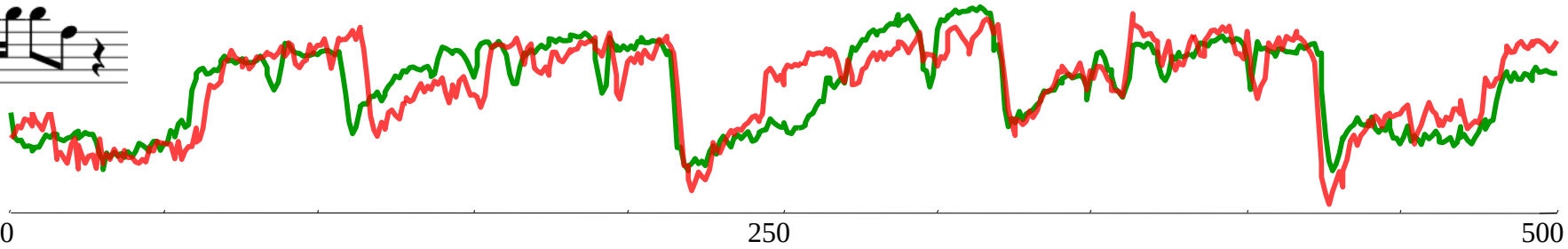
Queen-Bowie



Vanilla Ice

A zoom-in of the best conserved region between the two time series (the similarity join)

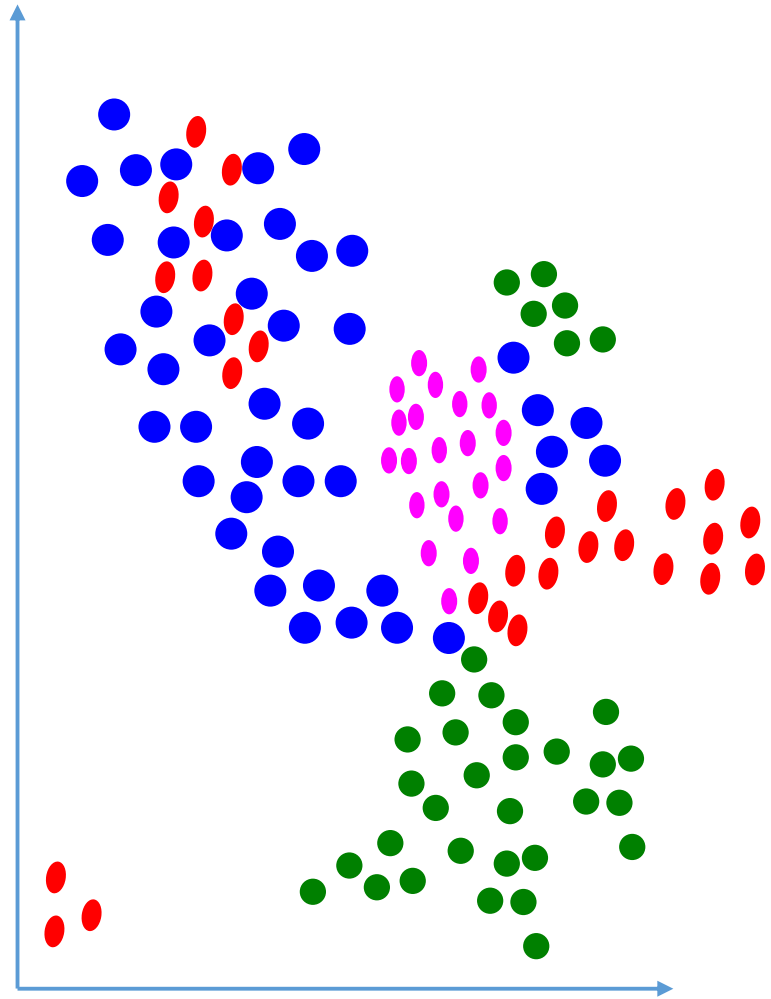




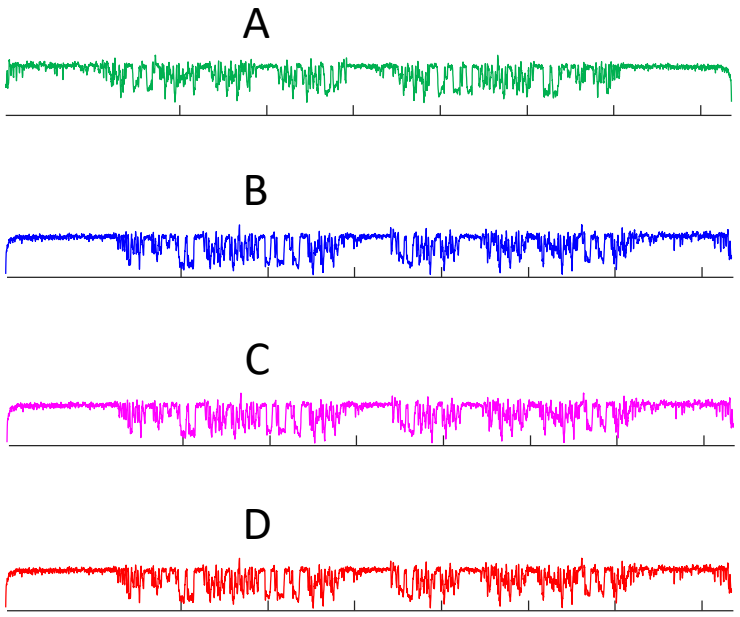
Queen-Bowie

Vanilla Ice

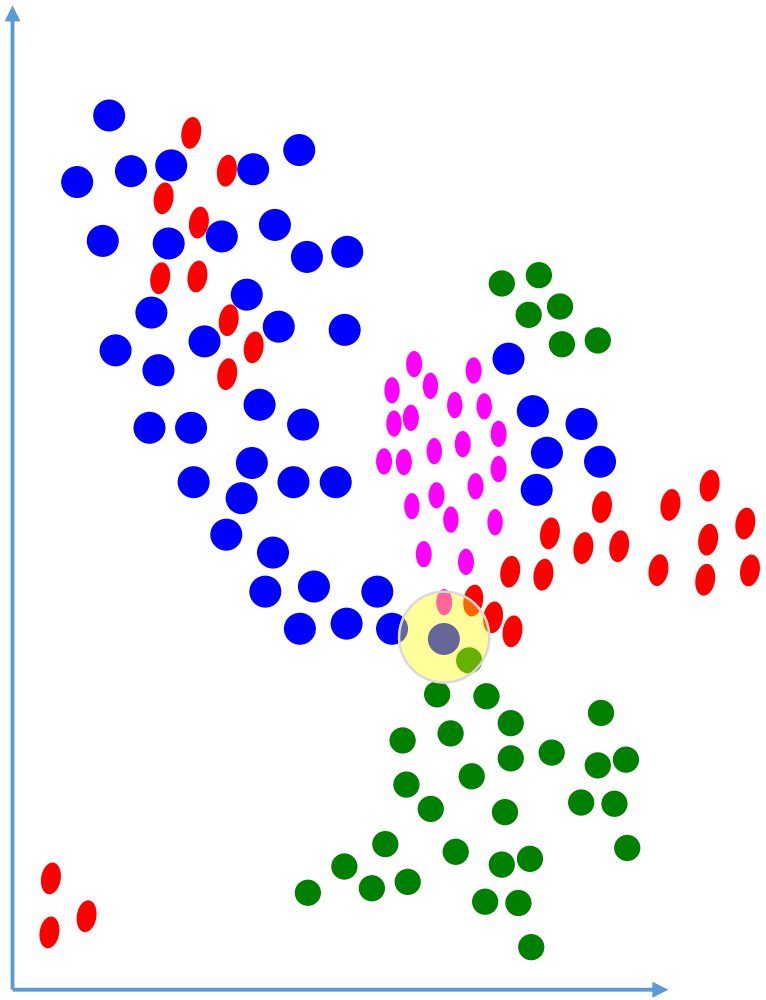
Generalizing to *Sets* of time series



Given a set of three or more time series. Is there a subsequence that (approximately) appears in *all* of them?



Generalizing to *Sets* of time series



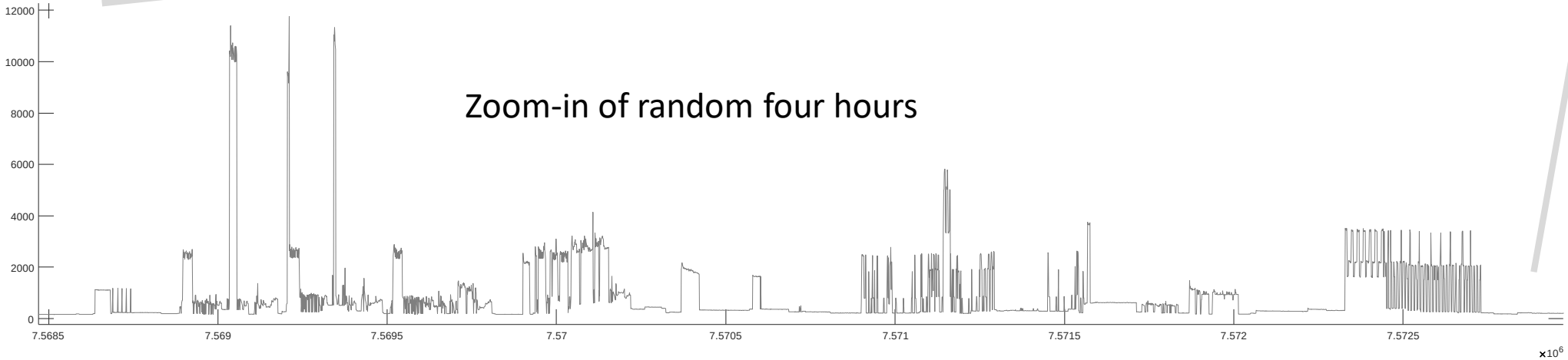
Given a set of three or more time series. Is there a subsequence that (approximately) appears in all of them?

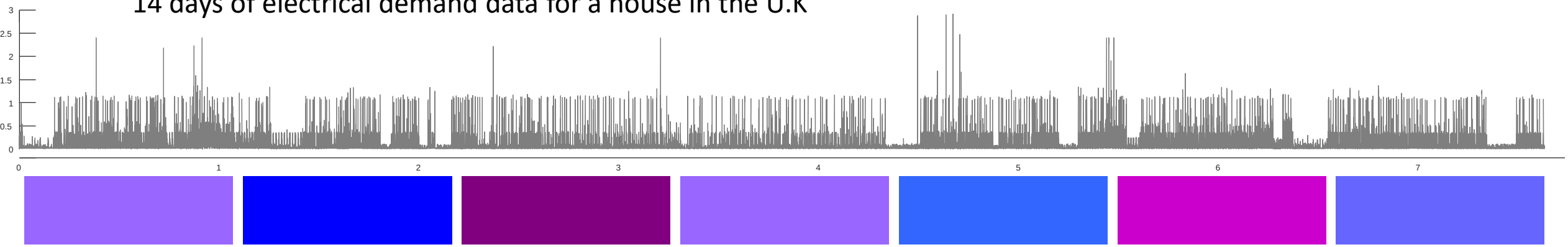
This is equivalent to asking, where can I place this yellow disk…

Making it as small as possible, and including at least one of each four colors.

14 days of electrical demand data for a house in the U.K

Zoom-in of random four hours



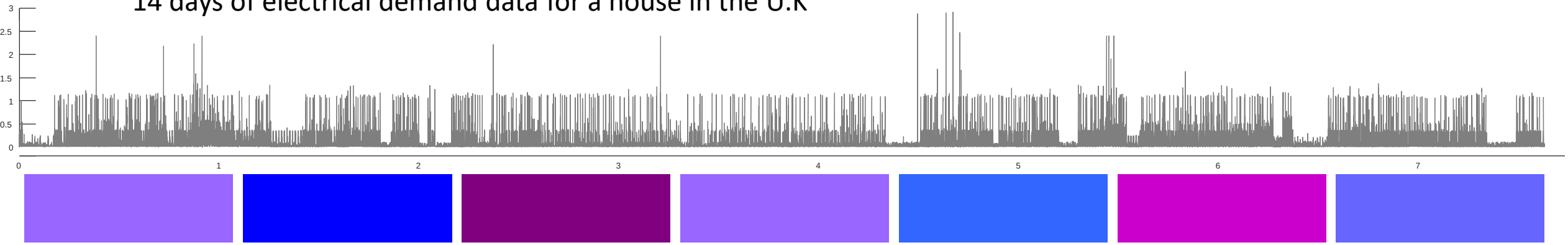


Lets divide our 14 days into seven 2-day chunks.

We can now ask the question, is there a two-minute pattern that happens, at least once, in each of the seven two-day chunks?

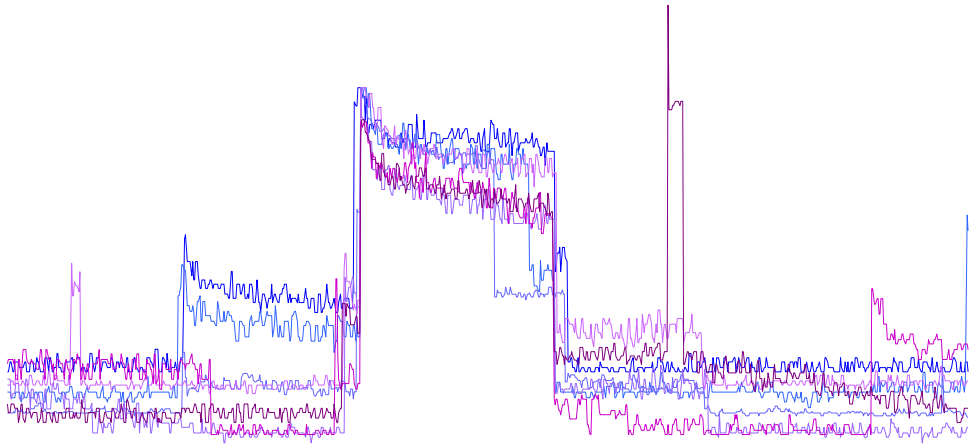
```
>> load TwoWeekElectrical
>> seven_two_day_chunks =  divide_data(T);
>> consensus_motifs = consensusMotifs(seven_two_day_chunks,800); % 800 is the length of subsequence
```

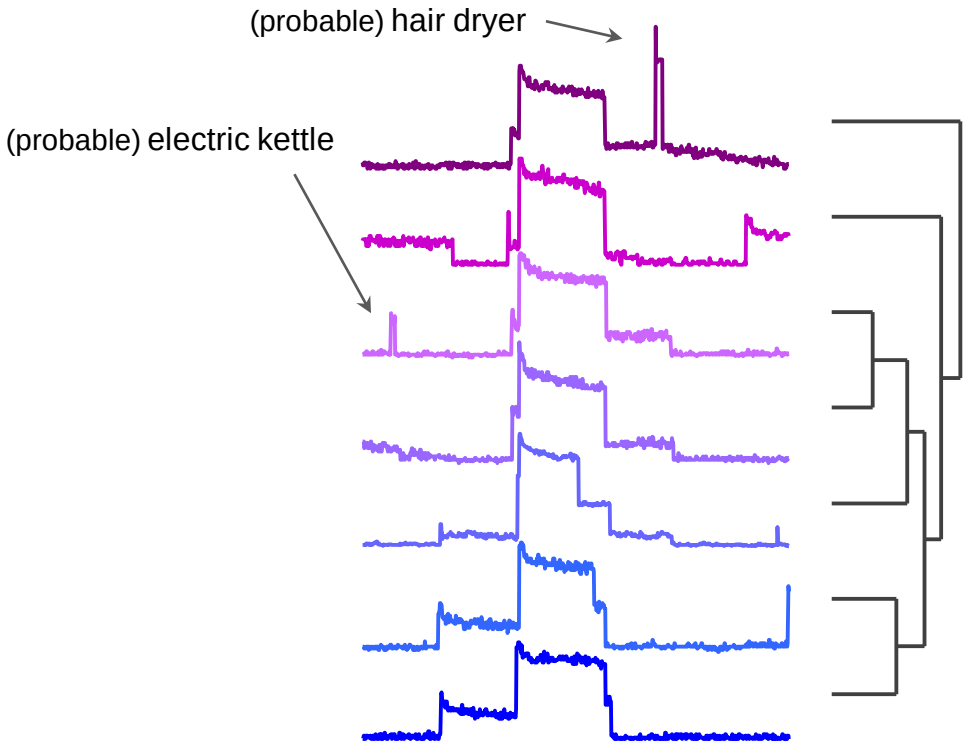
14 days of electrical demand data for a house in the U.K



Once we find the pattern, we can begin to ask new questions.

- What is this pattern?
- How do we explain the local differences?
- Are weekday patterns systematically different to weekend patterns?
- etc.



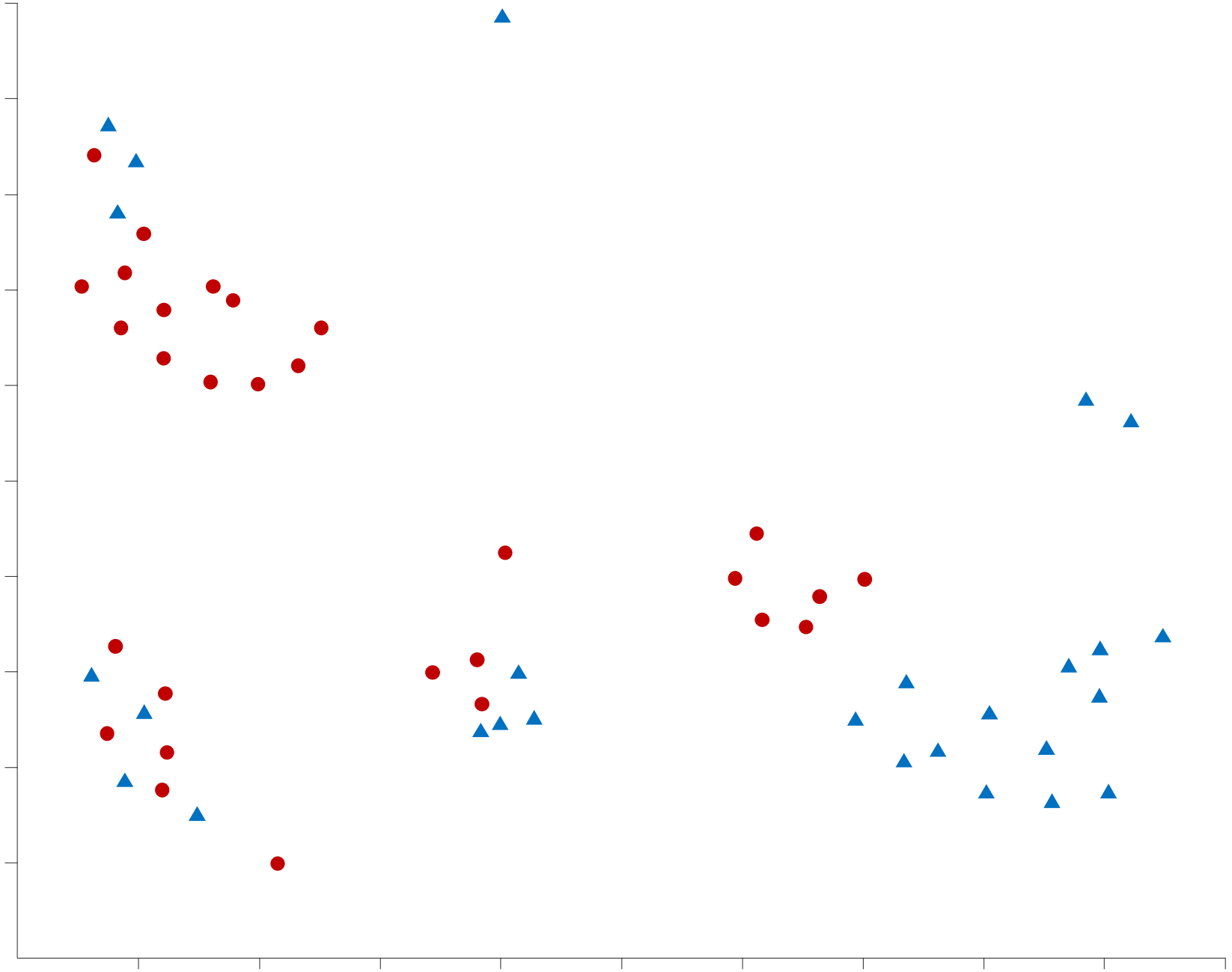


Mini Review

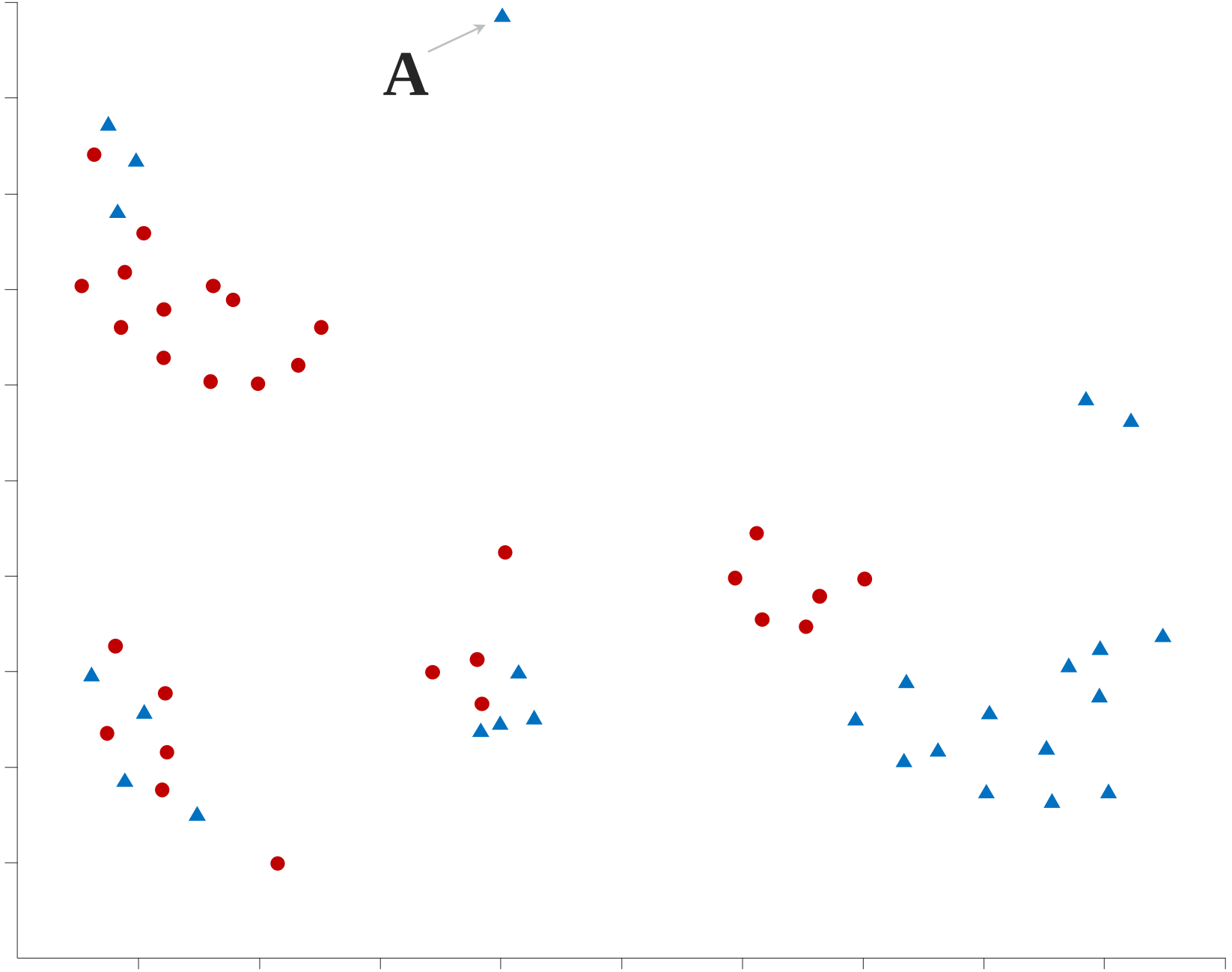
- Time series joins, and set-joins are fairly new primitives that have yet to be fully exploited.
- They let us reason about the presence and absence of previously unknow patterns
  - *What do we see in the winter, but not summer?*
  - *What do we see in cancer patients, but not in healthy patients?*
  - *What do we see in successful golf swings, but not in unsuccessful golf swings?*
  - *Etc.*

Lets see one last generalization

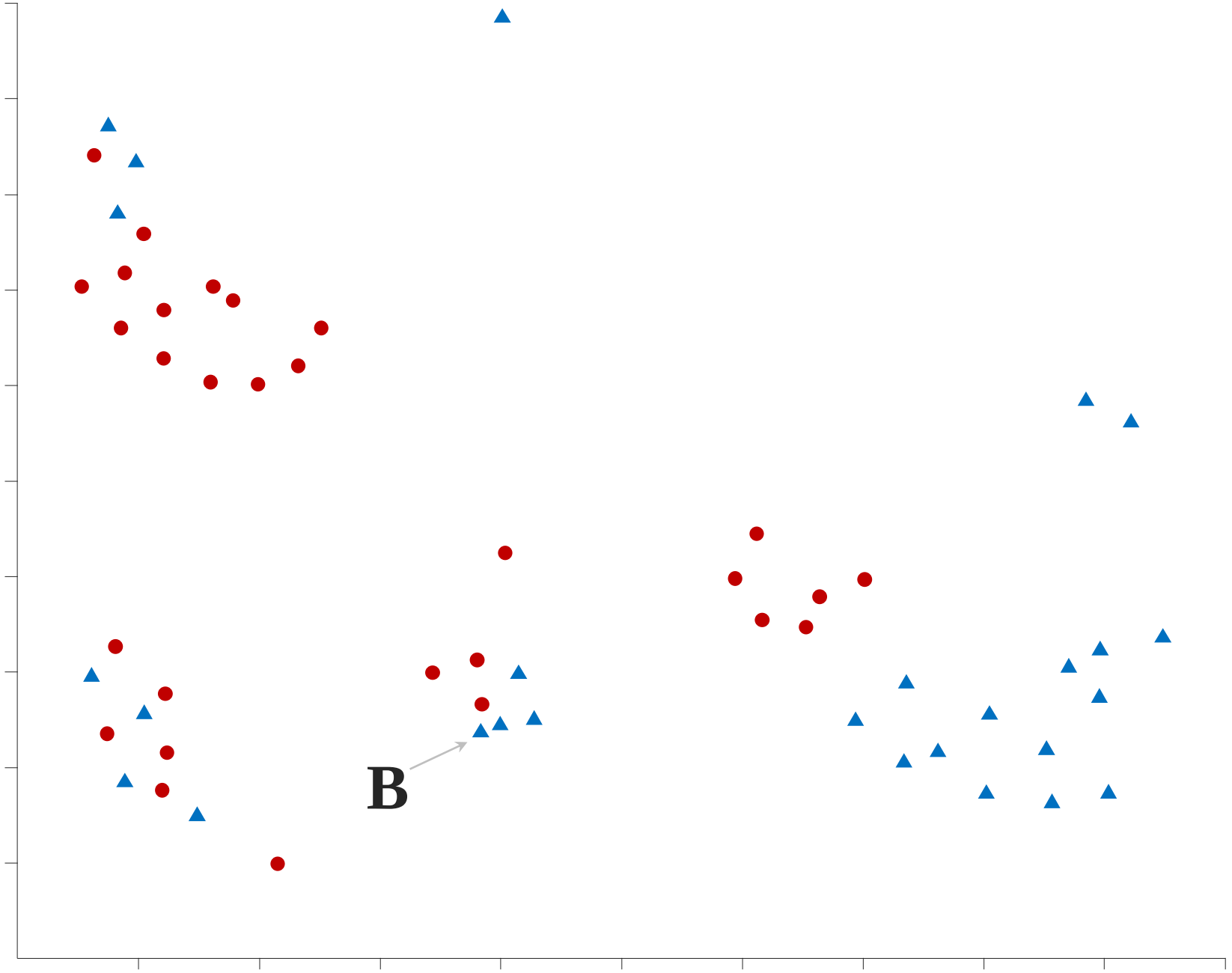
- This idea is called the *Contrast Profile*.
- It is the best idea in time series data mining published last year ;-)
- It only takes one line of code! (assuming the Matrix Profile is a primitive in your system)



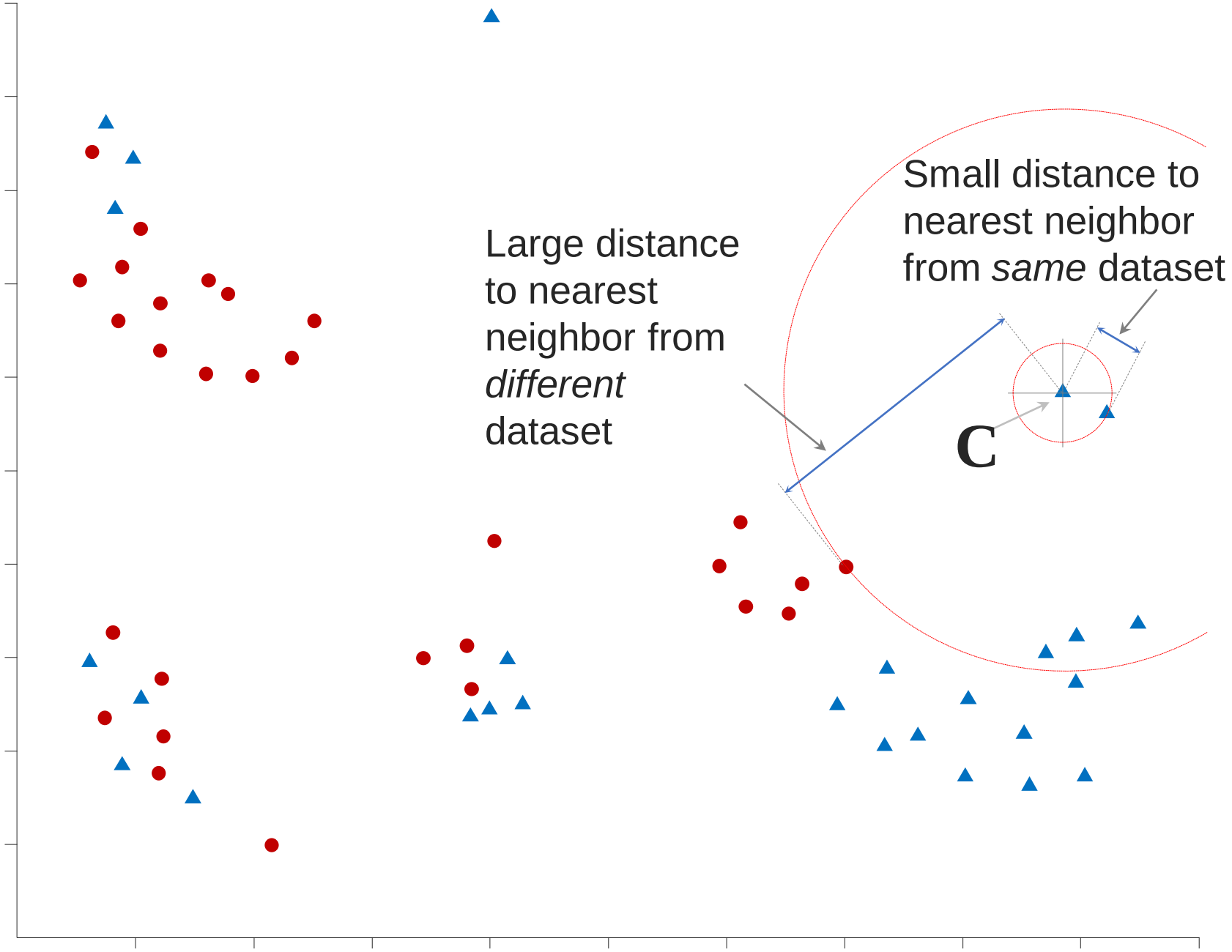
It may sometimes be useful to simultaneously reason about a subsequence's closeness to certain data and its distance to other data…



- Point **A** is far from its nearest neighbor in the non-target class, but it is also far from its nearest neighbor within its own target class.



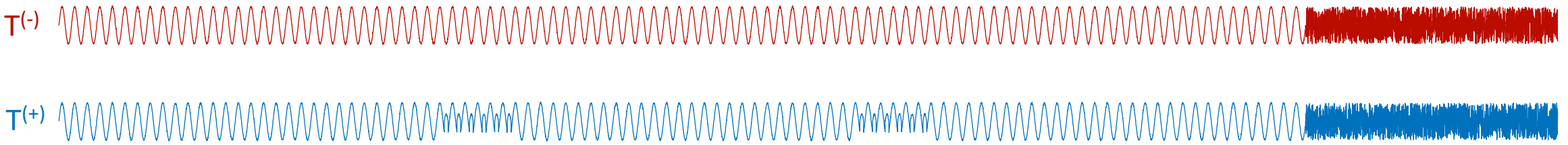
- Point **A** is far from its nearest neighbor in the non-target class, but it is also far from its nearest neighbor within its own target class.
- Point **B** in contrast is very close to its nearest neighbor in the target class, but it is also close to its nearest neighbors in non-target class.



- Point **A** is far from its nearest neighbor in the non-target class, but it is also far from its nearest neighbor within its own target class.
- Point **B** in contrast is very close to its nearest neighbor in the target class, but it is also close to its nearest neighbors in non-target class.
- Point **C** *is* both very far from its nearest neighbor in the non-target class *and* very close to its nearest neighbor in the target class

Contrast Profile: Toy Example (I)

- Given two toy time series $T^{(+)}$ and $T^{(-)}$, find behaviors in $T^{(+)}$ that are not in $T^{(-)}$



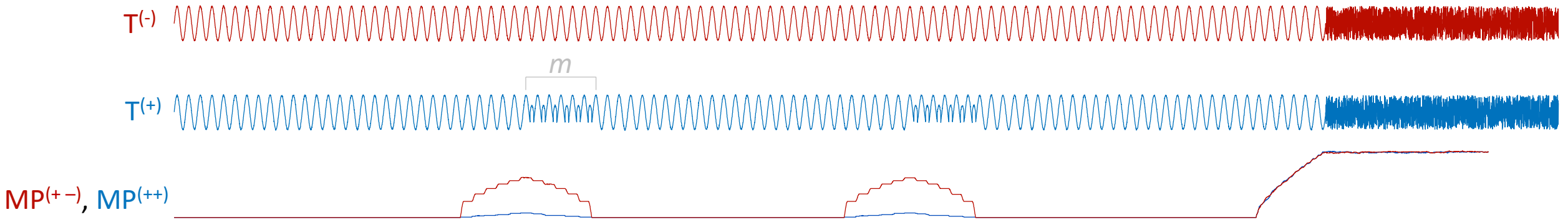
Key Insight: We cannot just ask what is in series $T^{(+)}$ but not in $T^{(-)}$, as any noise will always optimize that request.

We must ask what is *conserved* in $T^{(+)}$ and is not in $T^{(-)}$, but that is just

$$ContrastProfile = MP^{(+-)} - MP^{(++)}$$

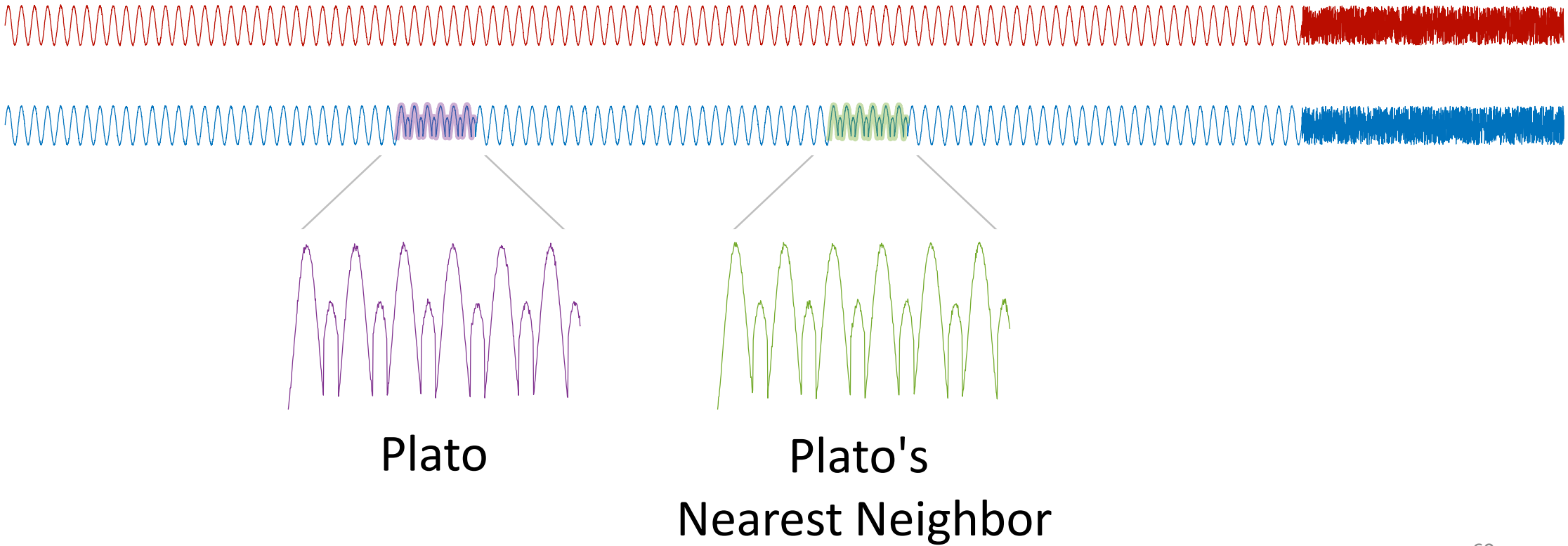
Contrast Profile: Toy Example (II)

- Focus on areas with the greatest difference between Matrix Profiles
  - The Matrix Profile represents the minimum nearest neighbor Euclidean Distance for each time index
  - Subsequence length *m*



Plato

- We call this subsequence a *Plato*
- Can be used for classification, because is *discriminative*

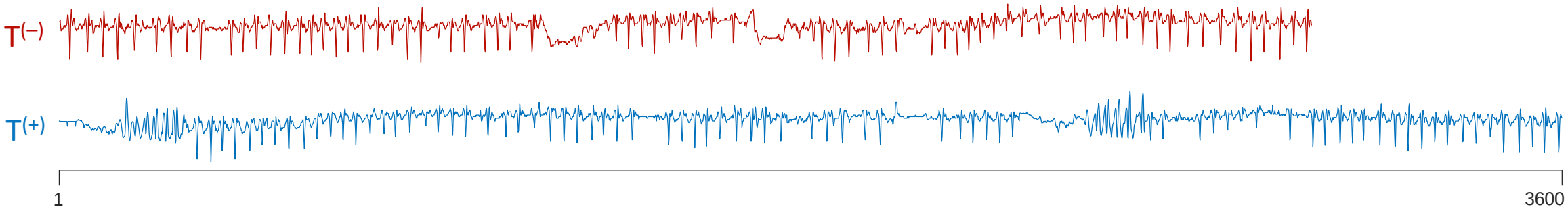


End-to-End Classification(I)

Farmers would like to know which chickens are dustbathing more than usual in order to treat them for mites.

Data: Accelerometer on chicken

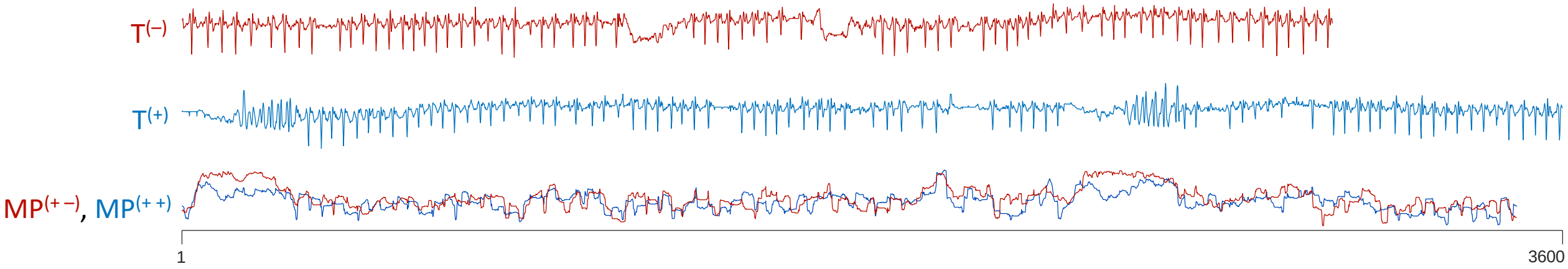
- $T^{(-)}$: Contains zero instances of dustbathing
- $T^{(+)}$: Contains at least two instances of dustbathing



* https://www.google.com/url?sa=i&url=https%3A%2F%2Fwww.istockphoto.com%2Fillustrations%2Fchicken&psig=AOvVaw0qNsf-mbJHBLcNjEpz4nP_&ust=1635285809227000&source=images&cd=vfe&ved=0CAsQjRxqFwoTCOjFqYbJ5vMCFQAAAAAdAAAAABAK

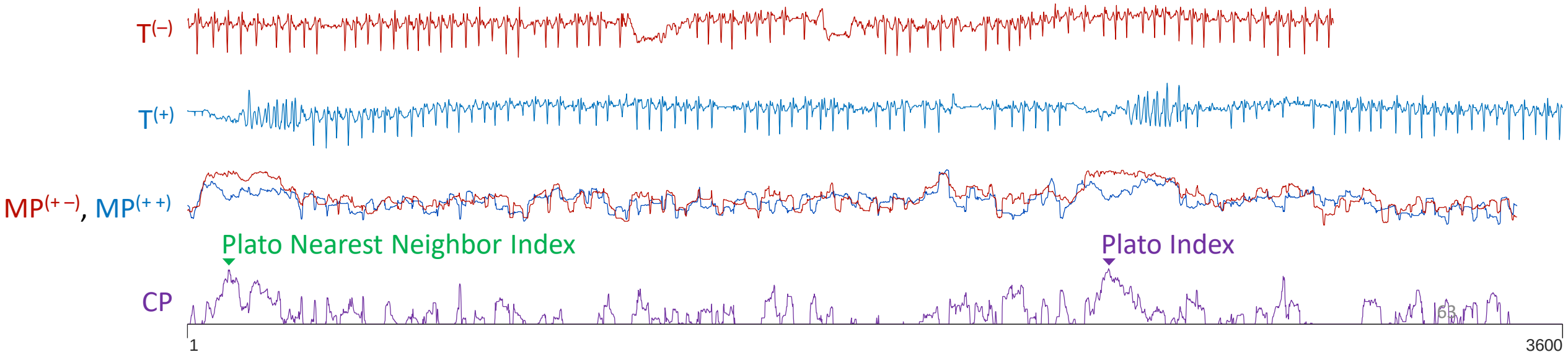
End-to-End Classification (II)

$MP^{(+-)}$ and $MP^{(++)}$ are similar, but differ in two key places.



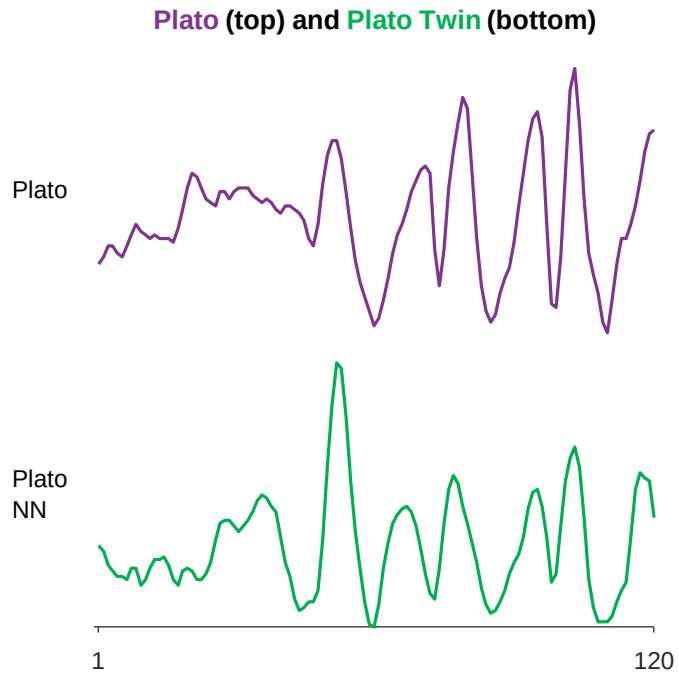
End-to-End Classification (III)

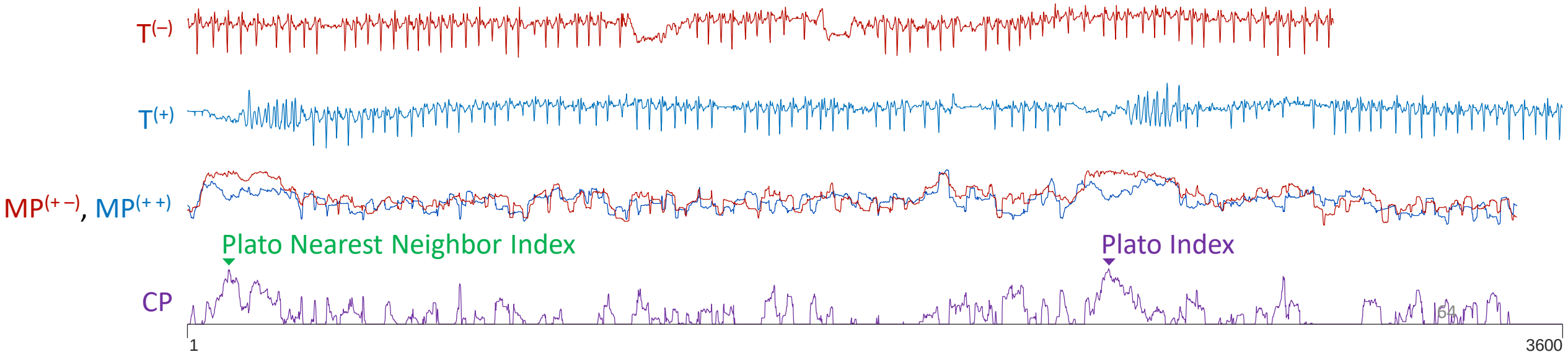
The Contrast Profile peaks where the behaviors between $T^{(+)}$ and $T^{(-)}$ are most contrasting.



End-to-End Classification (IV)

Notice how the Plato appears to have a somewhat flat prefix. This suggests that the start of the behavior is detected, which is helpful for identifying distinct instances.

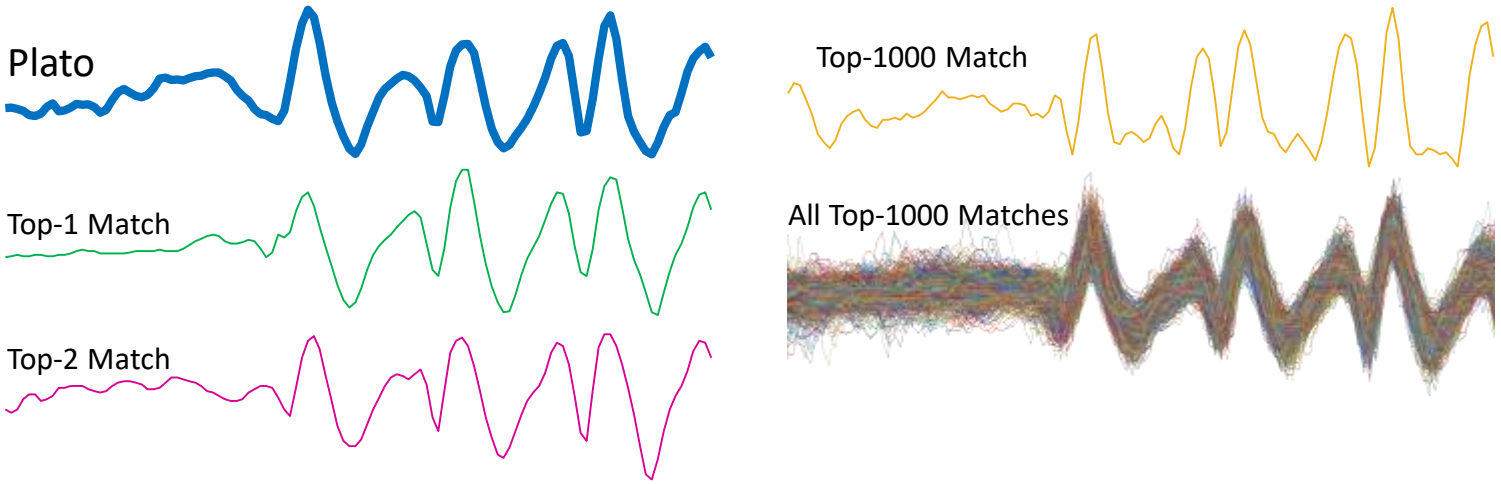




End-to-End Classification (V)

Probably the largest *real* dataset searched in a data mining paper, by two orders of magnitude

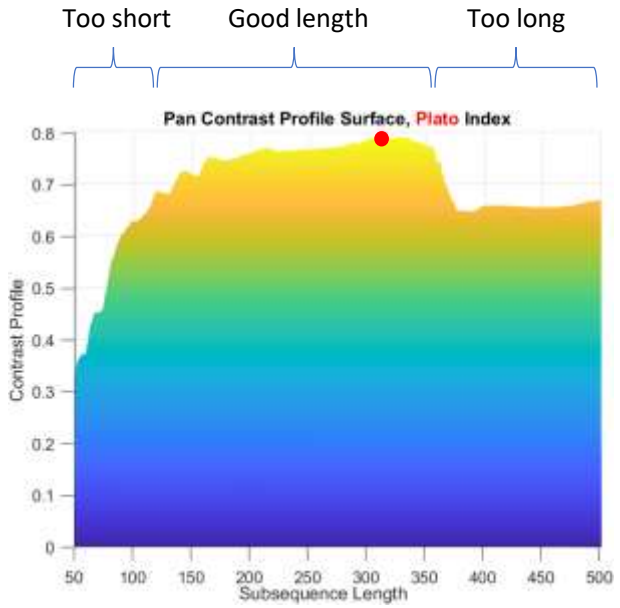
- We use the previously learned Plato to search a 12,679,054,727 datapoint (4 years) archive of chicken behavior for the one thousand best matches.
- Using MASS, this classification task is completed in 55 minutes.
- Given the limited training data, Matlab's off-the-shelf LSTM model [1] has an accuracy around the default rate **(<50%)** and would take nearly **1 year** to complete the classification.



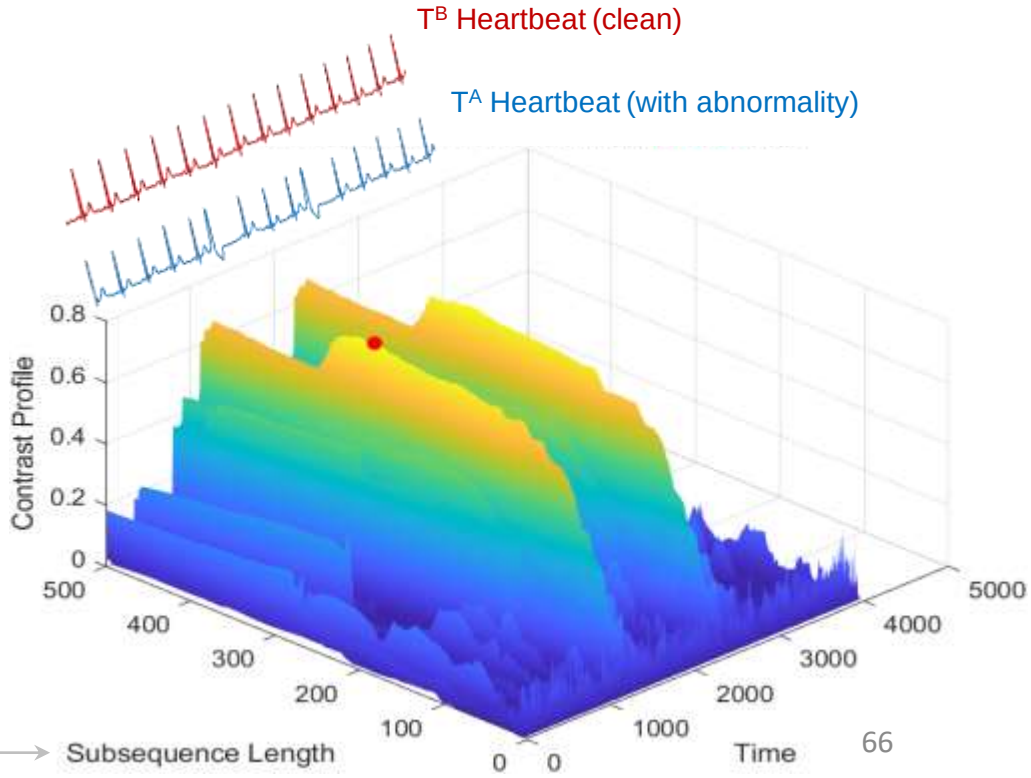
[1] "Sequence Classification Using Deep Learning." https://www.mathworks.com/help/deeplearning/ug/classify-sequence-data-using-lstm-networks.html (accessed Jan. 21, 2021).

Parameter Free Using Pan-Contrast Profile

- We can eliminate the only parameter!
- The Contrast Profile values range in [0,1], so different lengths can be compared.
- Thanks to existing optimization of MP, we can afford to compute all lengths!



- Red highlights the subsequence length that maximizes contrast.



The 1 parameter of Contrast Profile →

Data Exploration (I)

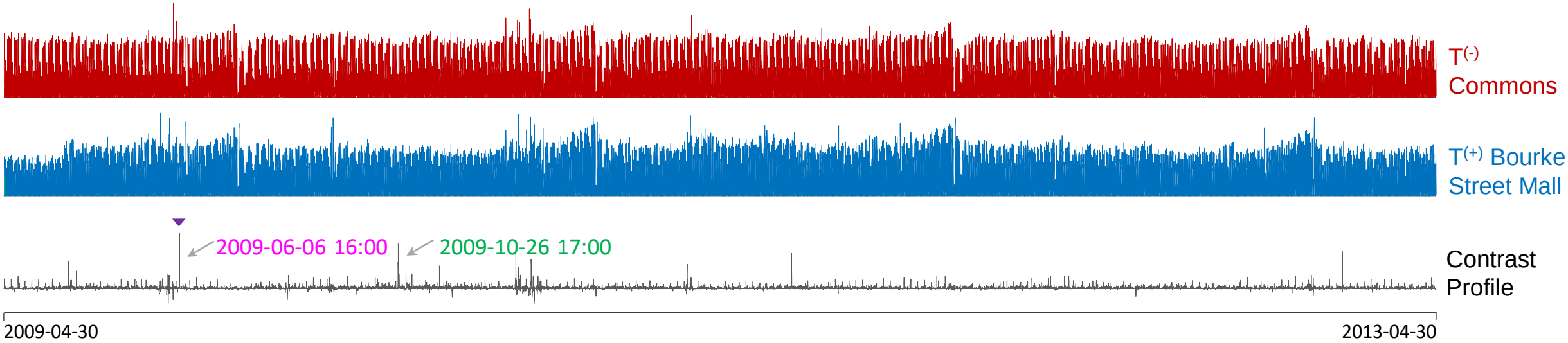
A city manager may wish to know:

"What happens only in Bourke Street but not elsewhere?"

Data: Pedestrian traffic in Melbourne, Australia [1]

$T^{(-)}$: Baseline location

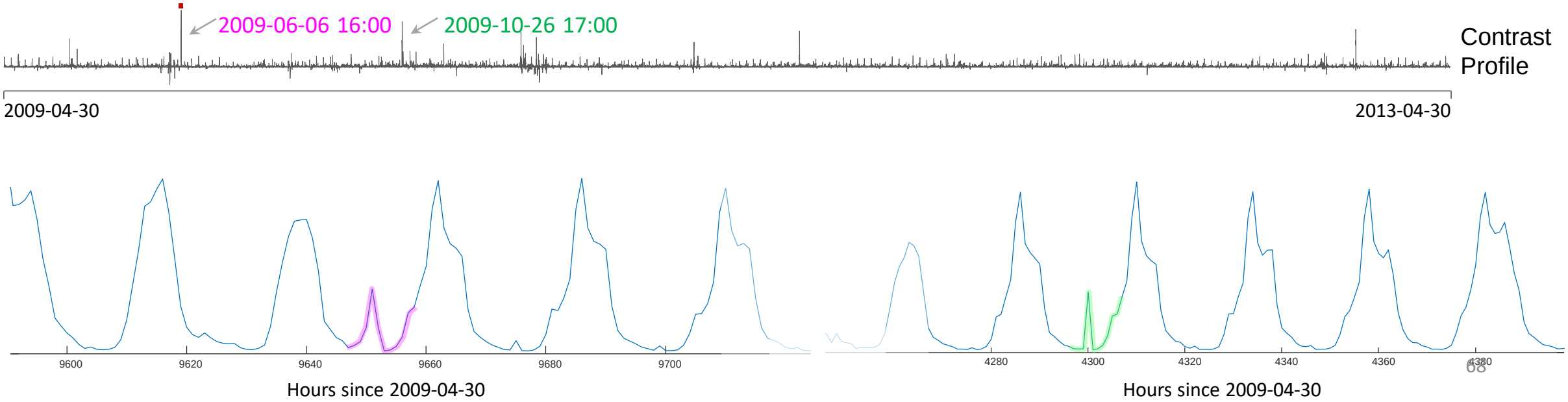
$T^{(+)}$: Bourke Street



[1] "City of Melbourne - Pedestrian Counting System." http://www.pedestrian.melbourne.vic.gov.au/#date=28-10-2021&time=8 (accessed Oct. 27, 2021).
[2] https://www.google.com/url?sa=i&url=https%3A%2F%2Fdepositphotos.com%2F241761348%2Fstock-illustration-silhouette-random-pedestrian-vector-white.html&psig=AOvVaw2W77d9LT67TFWX42SUWVPR&ust=1635459903248000&source=images&cd=vfe&ved=0CAsQjRxqFwoTCMDEq4zR6_MCFQAAAAAdAAAAABAT

Data Exploration (II)

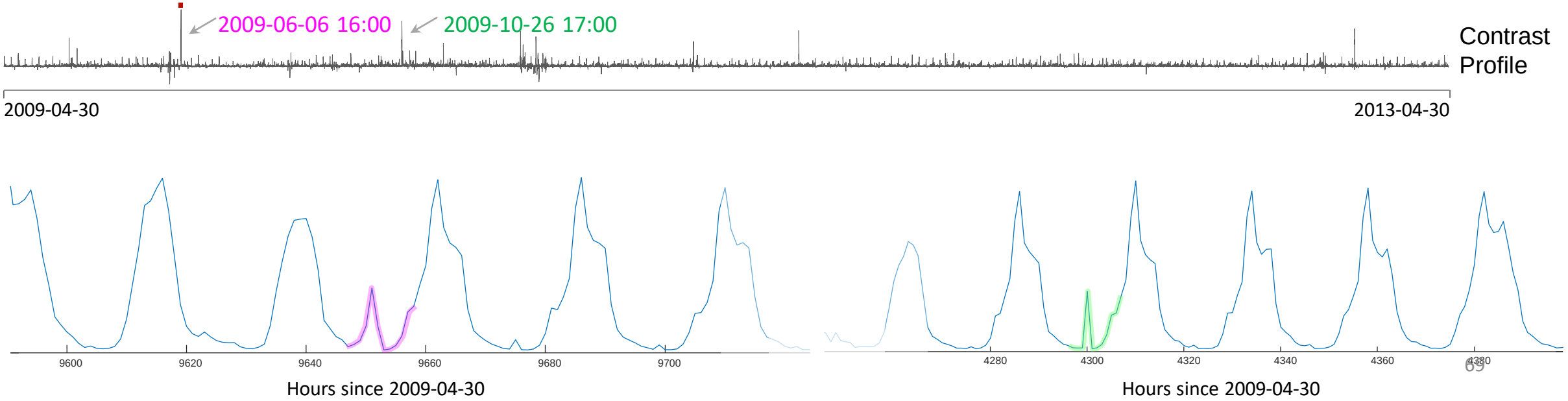
What caused this?



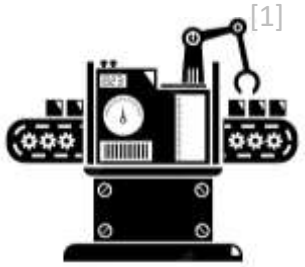
Data Exploration (II)

Crowdsourced explanation using Reddit:

"That was around the time of the new store opening up and I worked there at the time. We had .. (EOD fire drills) evacuations mid 2009-mid 2010. ..the Bourke doors being the only way to access Myer some days."



Interpretable Anomaly Discovery Using Industrial Data[2] (I)

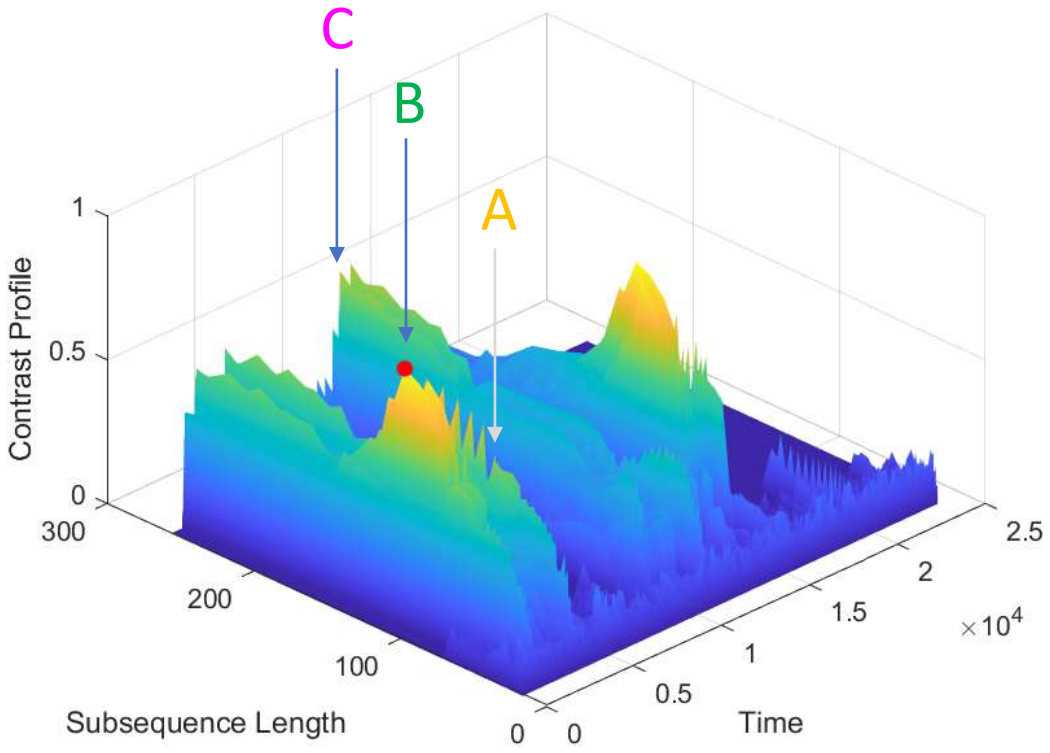

[1]

[1] https://www.google.com/url?sa=i&url=https%3A%2F%2Fwww.bigstockphoto.com%2Fimage-88671581%2Fstock-vector-industrial-machine&psig=AOvVaw3Mz509cGgF9B0GP06tzSlc&ust=1635288507473000&source=images&cd=vfe&ved=0CAsQjRxqFwoTCMDkjsrS5vMCFQAAAAAdAAAAABAJ

[2] https://www.kaggle.com/inIT-OWL/high-storage-system-data-for-energy-optimization

Interpretable Anomaly Discovery Using Industrial Data (II)

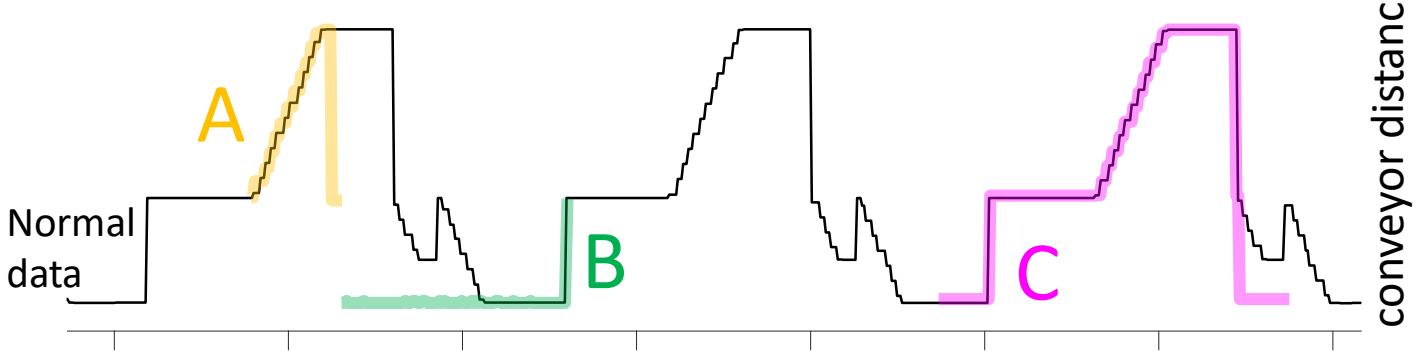
- $T^{(-)}$: "Normal" industrial process (real data!)
- $T^{(+)}$: "Has Abnormalities" in industrial process
- Identify anomalies so the abnormal industrial process can be fixed.
- Meaningful Platos are found at various subsequence lengths



C: Bottom Motion Aborted and Reset

B: Extended Reset After Complete Cycle

A: Abort On Top and Reset

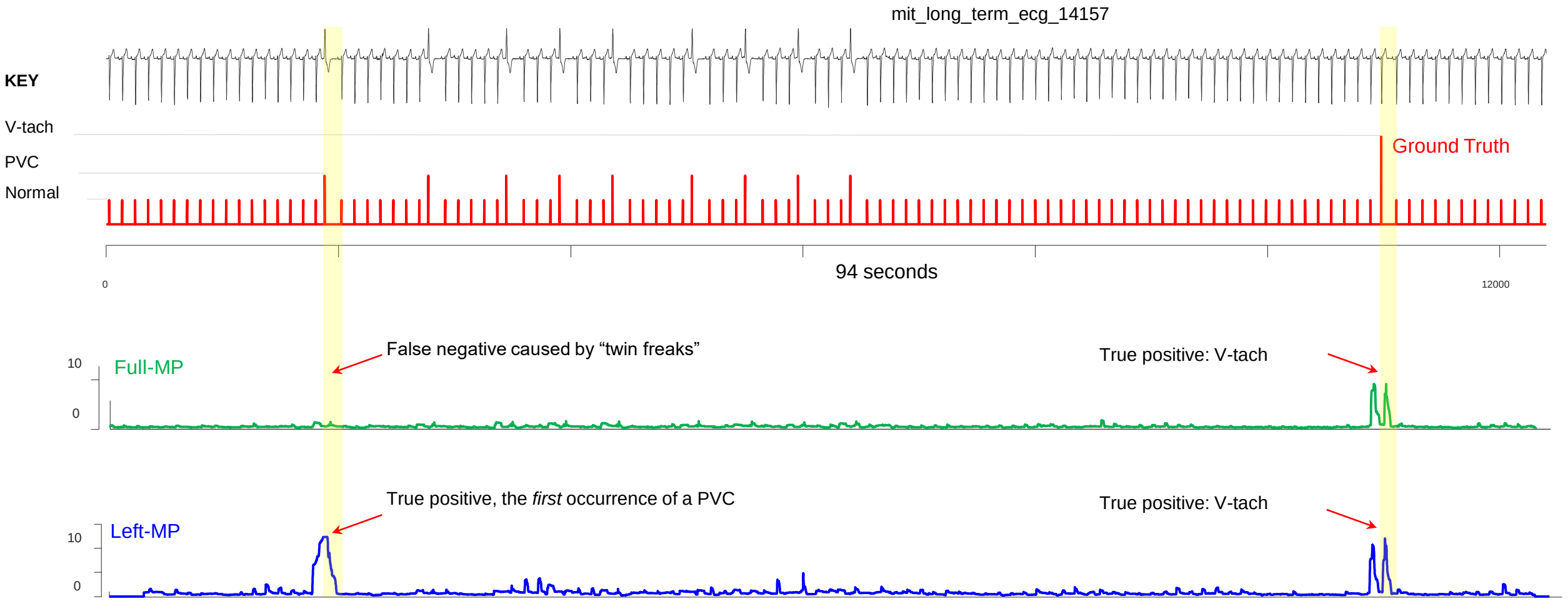


Conclusions

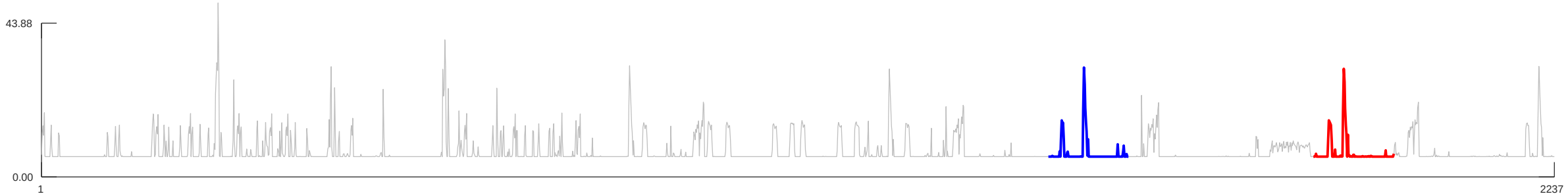
- We have seen a tour of shape-based analytics in time series data.
- All this ideas use conserved patterns (motifs) as their main primitive.
- All my code and datasets are publicly available
- I am always interested to hear of your problems/suggestions etc.

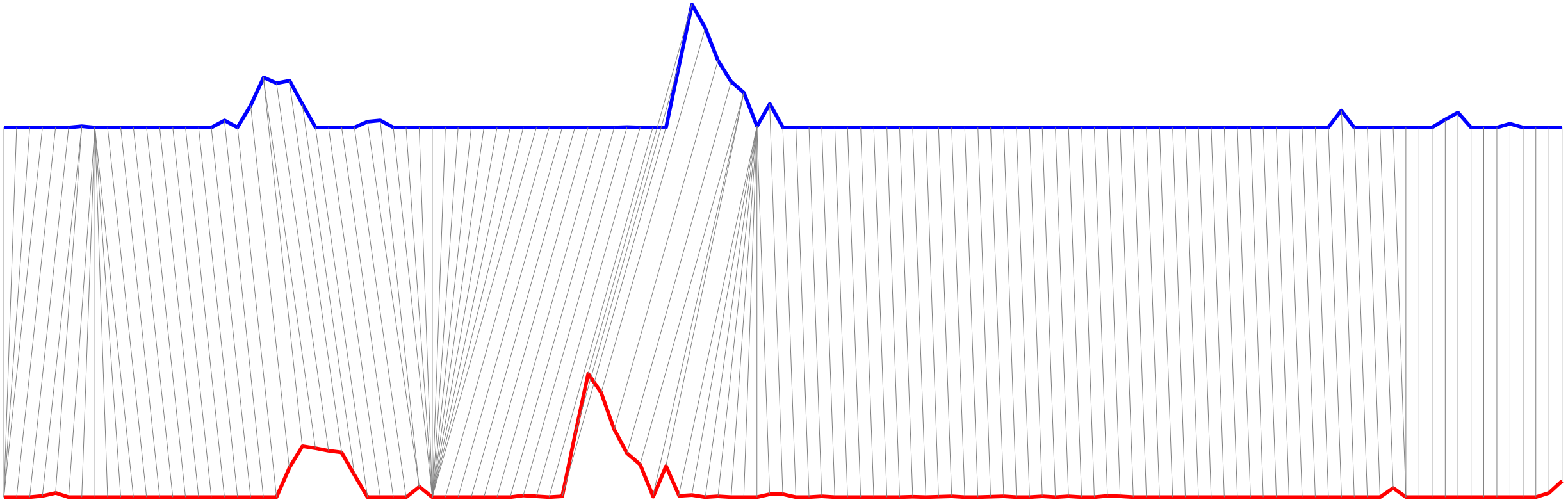
Questions!

Twin Freaks is a Non-Issue



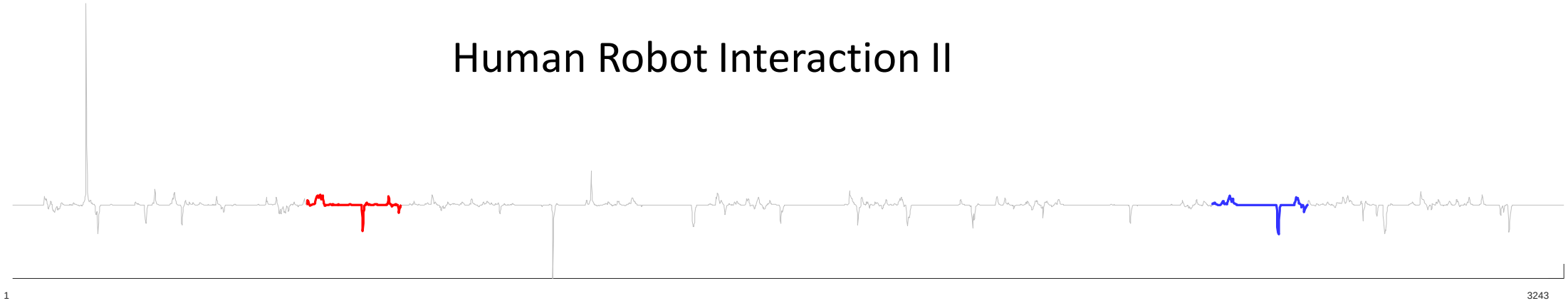
Water pressure in park in Portugal

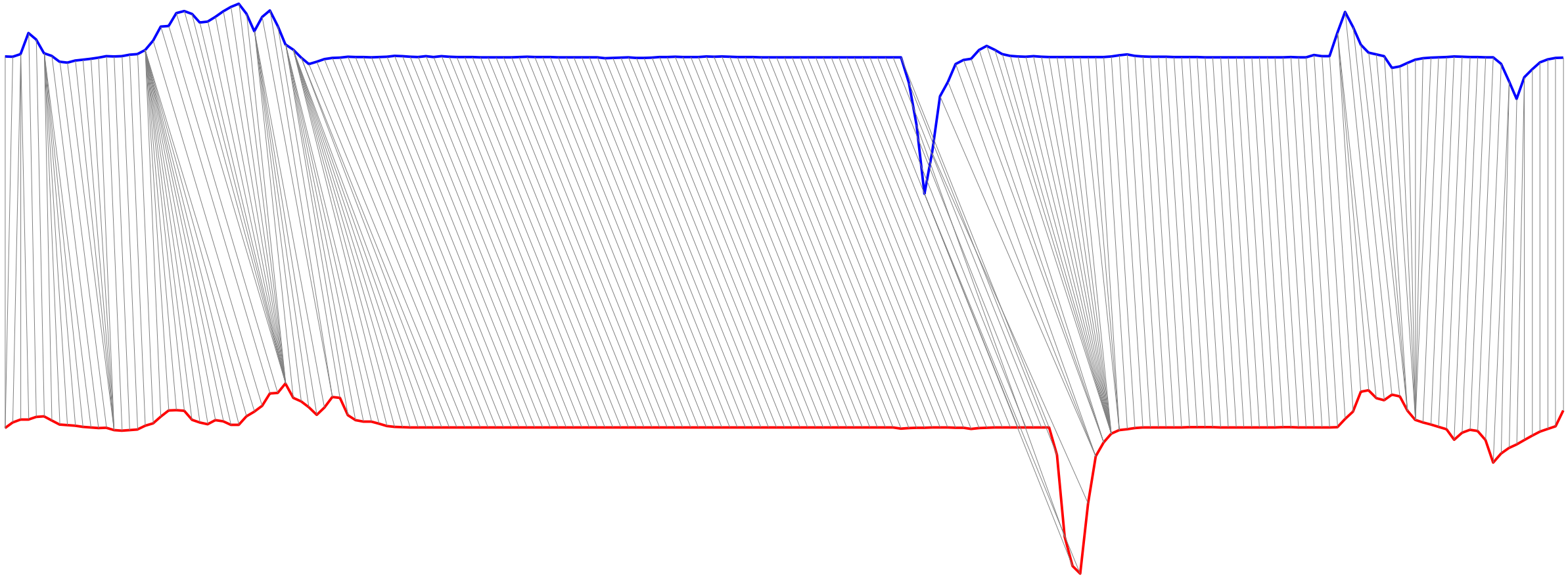




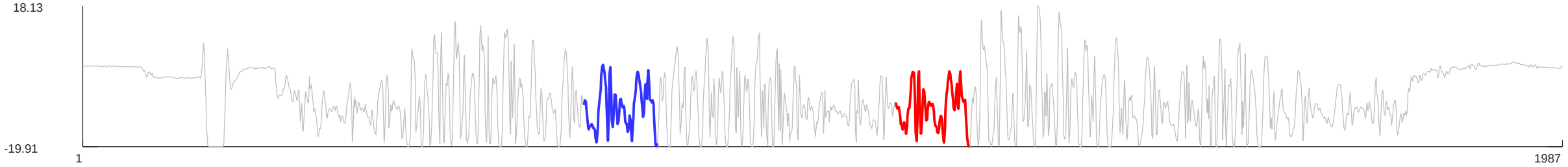
In this example, Phase I of SWAMP prunes 98.845% of the DTW calculations

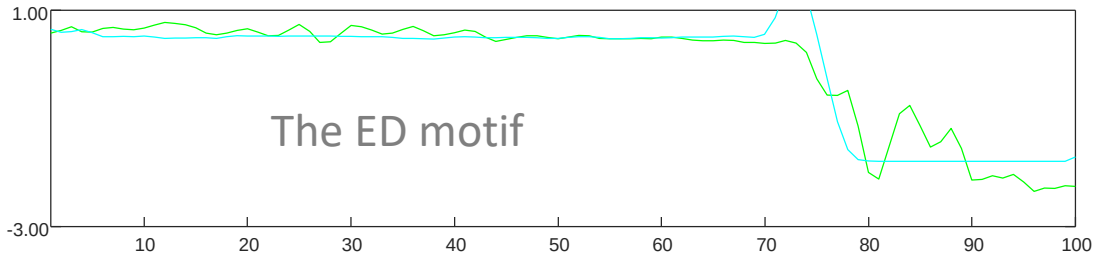
Human Robot Interaction II

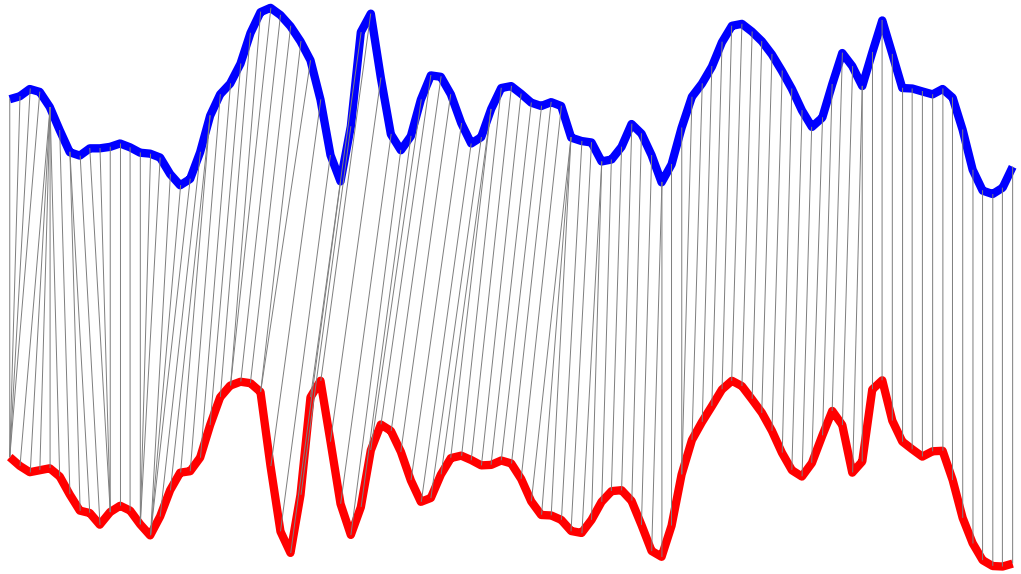




Almende Continuous Real-world Activities I

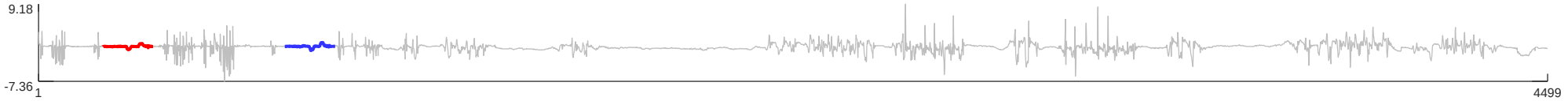




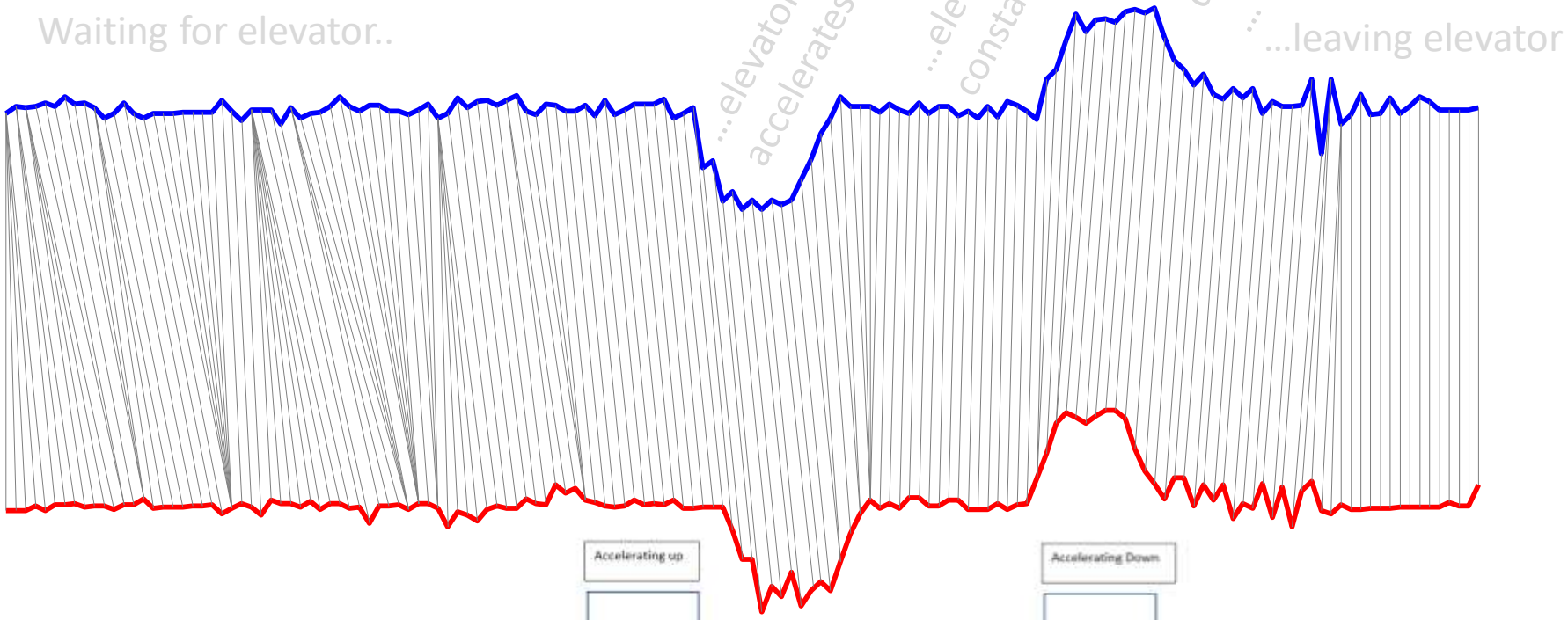


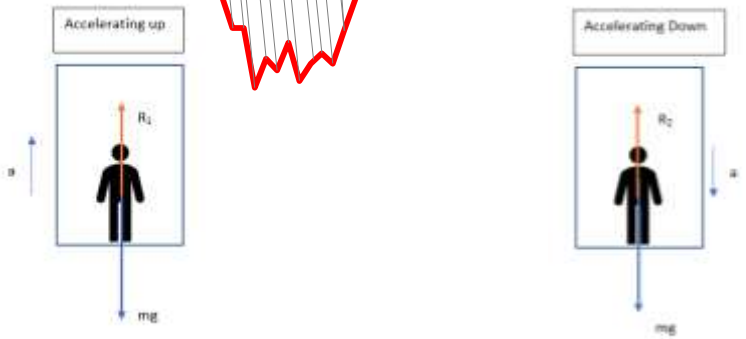
Based of an inspection of the video, the DTW motif corresponds to the transition from a run to a walk

Elevator



Why is the elevator slower in one of the runs?
We don't know, we speculate that there might have been more people in it.





Coarse In-building Localization with Smartphones *

Avinash Parnandi[1], Eric Le[2], Pradeep Vaghela[1], Aditya Kolli[1], Karthik Dantu[1], Sameera Poduri[1], Gaurav S. Sukhatme[1]

[1]Computer Science Department, [2]Ming Hsieh Department of Electrical Engineering
University of Southern California, Los Angeles, CA 90089, USA

Abstract. Geographic location of a person is important contextual information that can be used in a variety of scenarios like disaster relief, directional assistance, context-based advertisements, etc. GPS provides accurate localization outdoors but is not useful inside buildings. We propose an coarse indoor localization approach that exploits the ubiquity of smart phones with embedded sensors. GPS is used to find the building in which the user is present. The Accelerometers are used to recognize the user's dynamic activities (going up or down stairs or an elevator) to determine his/her location within the building. We demonstrate the ability to estimate the floor level of a user. We compare two techniques for activity classification, one is naive Bayes classifier and the other is based on dynamic time warping. The design and implementation of a localization application on the HTC G1 platform running Google Android is also presented.

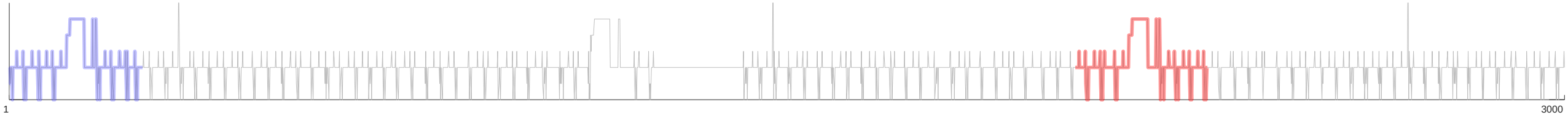
1 Introduction

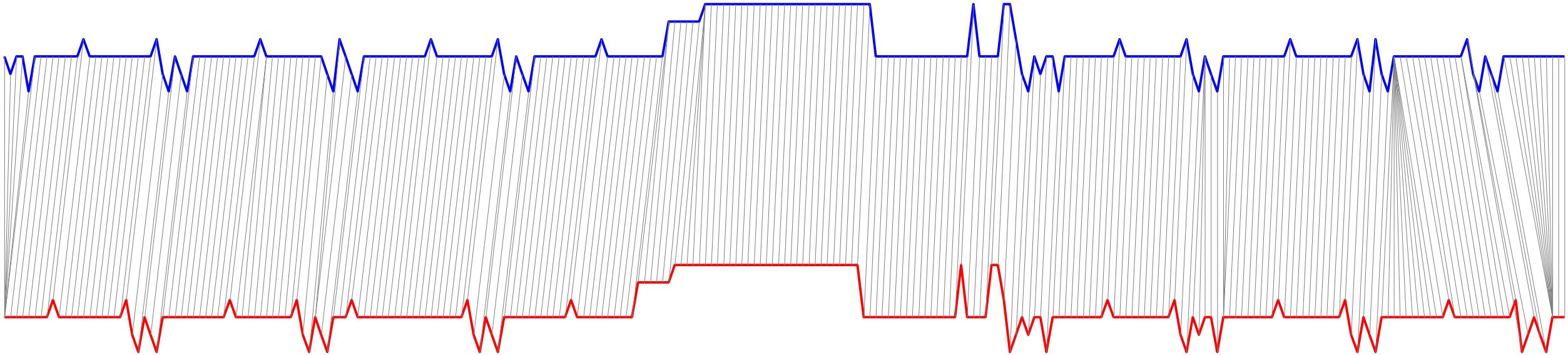
Indoor localization is a challenging problem facing the ubiquitous computing research community. Accurate location information is easily obtained outdoors using GPS. However, when we enter a building the reception of GPS signals become weak or is lost causing the inability to determine additional location information within the building. Existing solutions for indoor localization include techniques that use WiFi [4], RFID, Bluetooth [2], ultrasound [16], infrared [19] and GSM [17] [13] etc. Many of these solutions rely on external infrastructure or a network of nodes to perform localization. Note that improving the localization accuracy or increasing the availability of the services provided by these systems require the scaling of infrastructure which can be costly and a challenge on its own.

Our goal is to develop a localization system that is independent of external infrastructure. To this end, we built an indoor localization application that relies primarily on the sensors embedded in a smartphone to coarsely locate a person within a building. Our approach is based on user activity modeling. We use the GPS receiver to determine which building the user entered, and we use the accelerometer to determine what activities the user is performing within the building. We localize the user by coupling

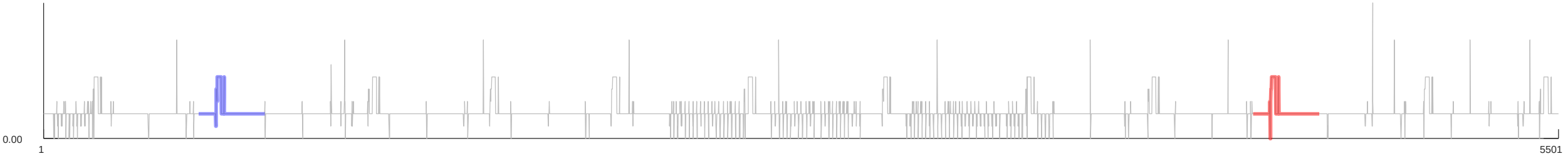
* This work was supported in part by NSF grants CCR-0120778 (CENS: Center for Embedded Networked Sensing), and by a gift from the Okawa Foundation. It was initiated as a project for the graduate course CS 546 Intelligent Embedded Systems taught at USC in Spring 2008.

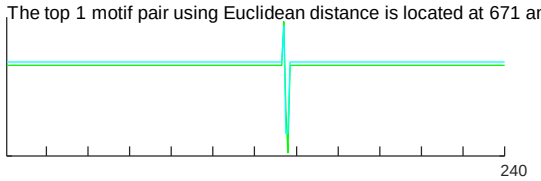
IoT devices network traces I

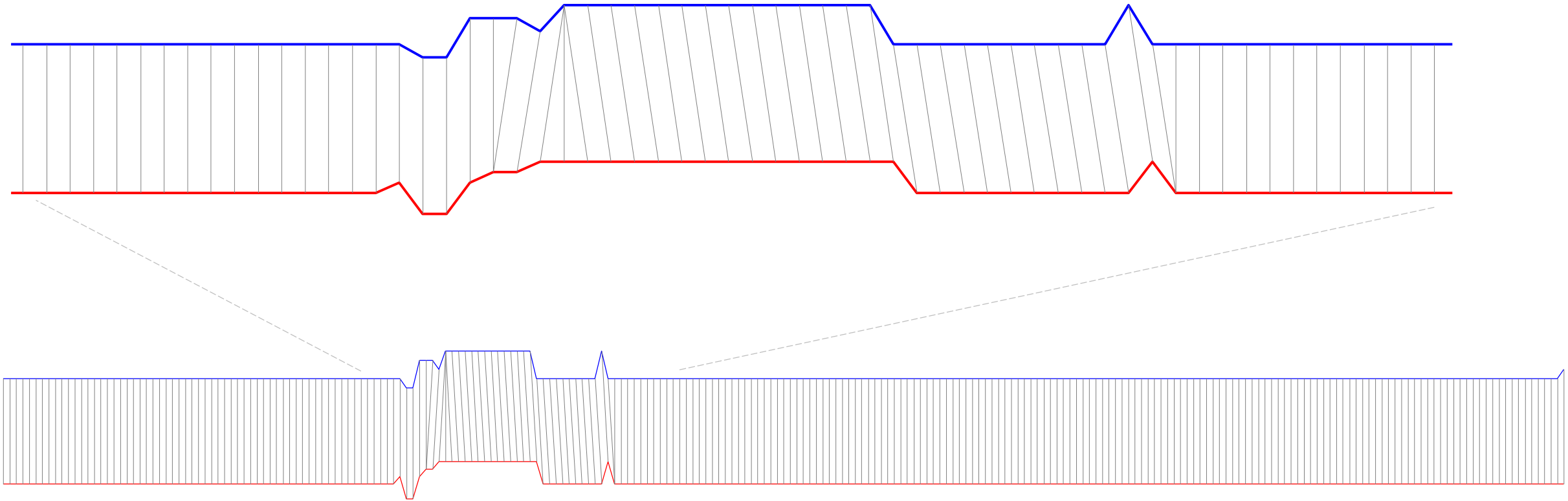


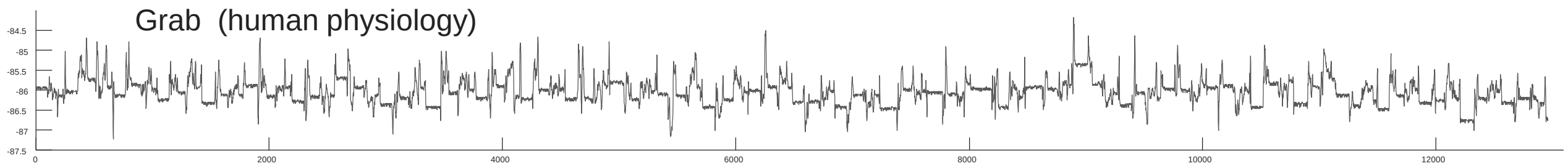


IoT devices network traces II

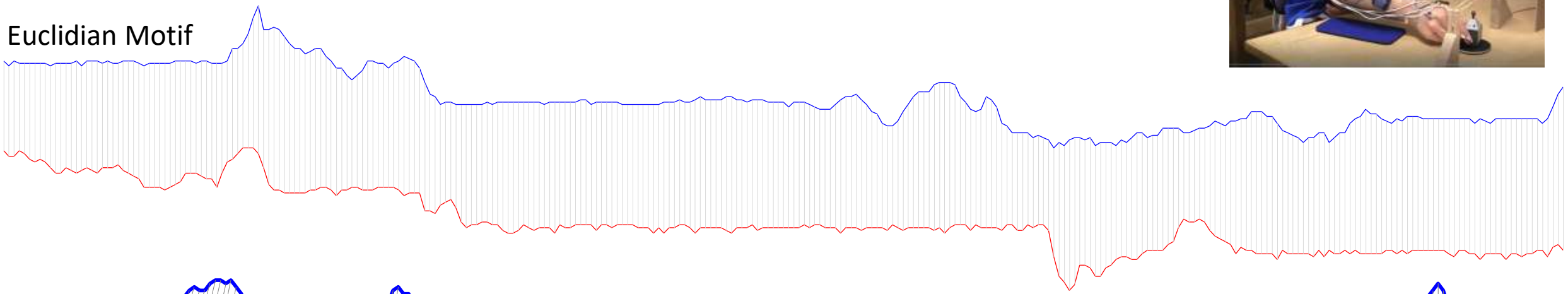


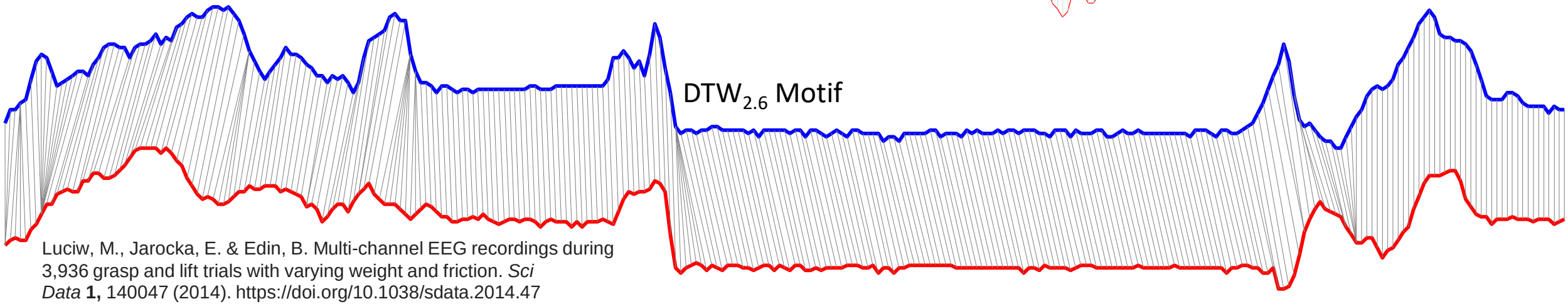






In this dataset, for subsequences of length 300, the Euclidian Motif is a pair that only vaguely resemble each other. By just allowing a tiny amount of warping (2.6%), the DTW shows a strongly conserved behavior.



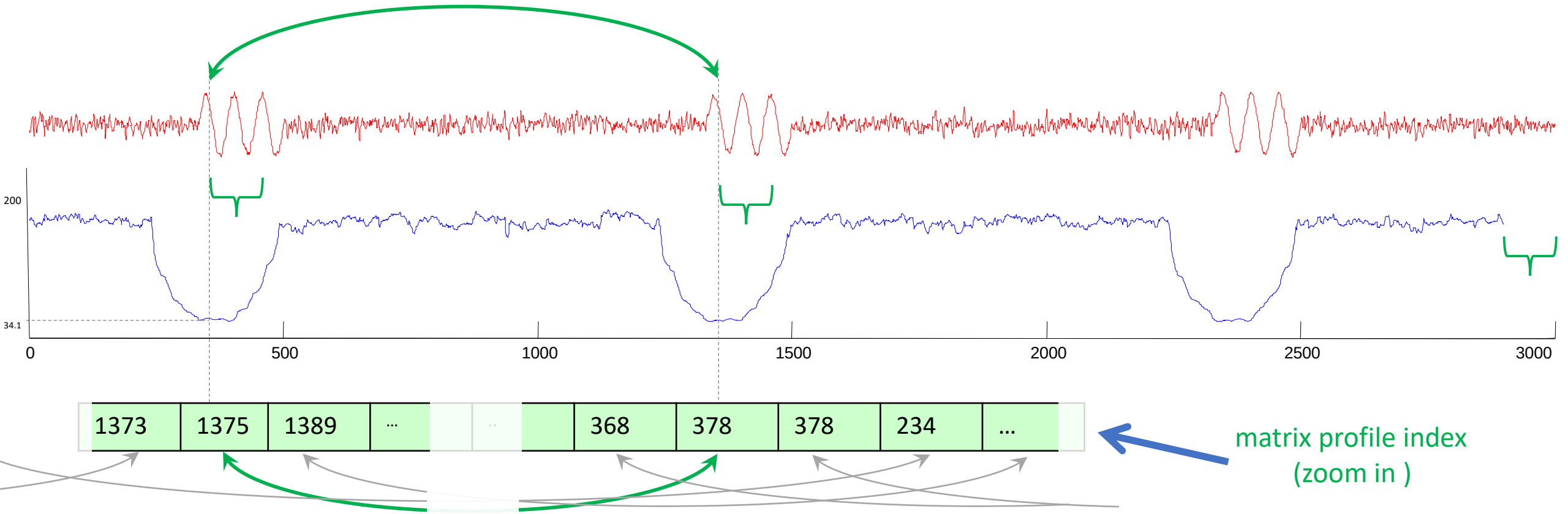


Luciw, M., Jarocka, E. & Edin, B. Multi-channel EEG recordings during 3,936 grasp and lift trials with varying weight and friction. *Sci Data* **1,** 140047 (2014). https://doi.org/10.1038/sdata.2014.47

We can create another companion sequence, called a matrix profile index.

The MPI contains integers that are used as pointers. As a practical matter, even 32-bits will let us have a MP of length 2,147,483,647, over two years of data at 60Hz. A 64-bit integer gives us ten billion years at 60Hz)

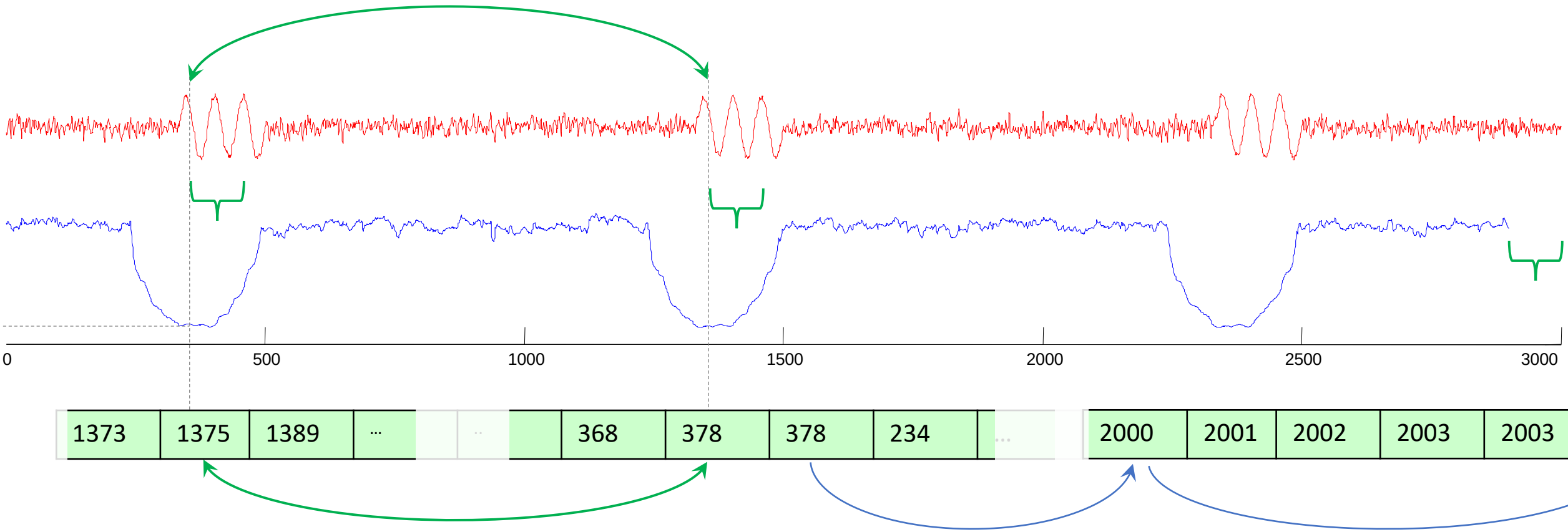
In the following slides we won't bother to show the matrix profile index, but be aware it exists, and it allows us to find the nearest neighbor to any subsequence in constant time.



Note that the pointers in the matrix profile index are not necessarily symmetric.

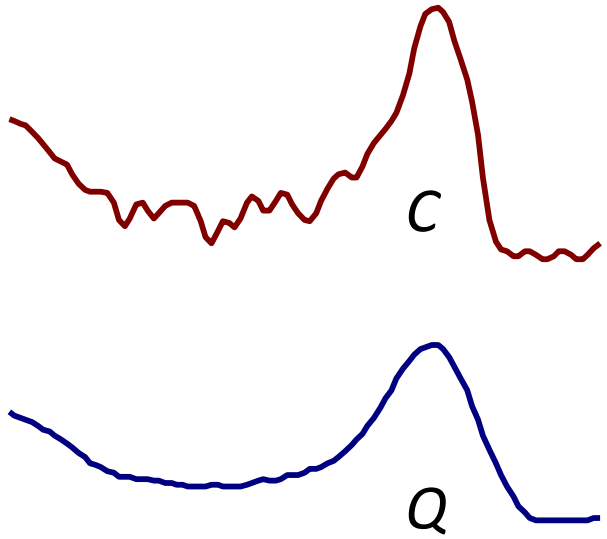
If **A** points to **B**, then **B** may or may not point to **A**

An interesting exception, the two smallest values in the MP must have the same value, and their pointers must be mutual. This is the classic *time series motif*.



Comparing two time series

How similar are these two-time series?



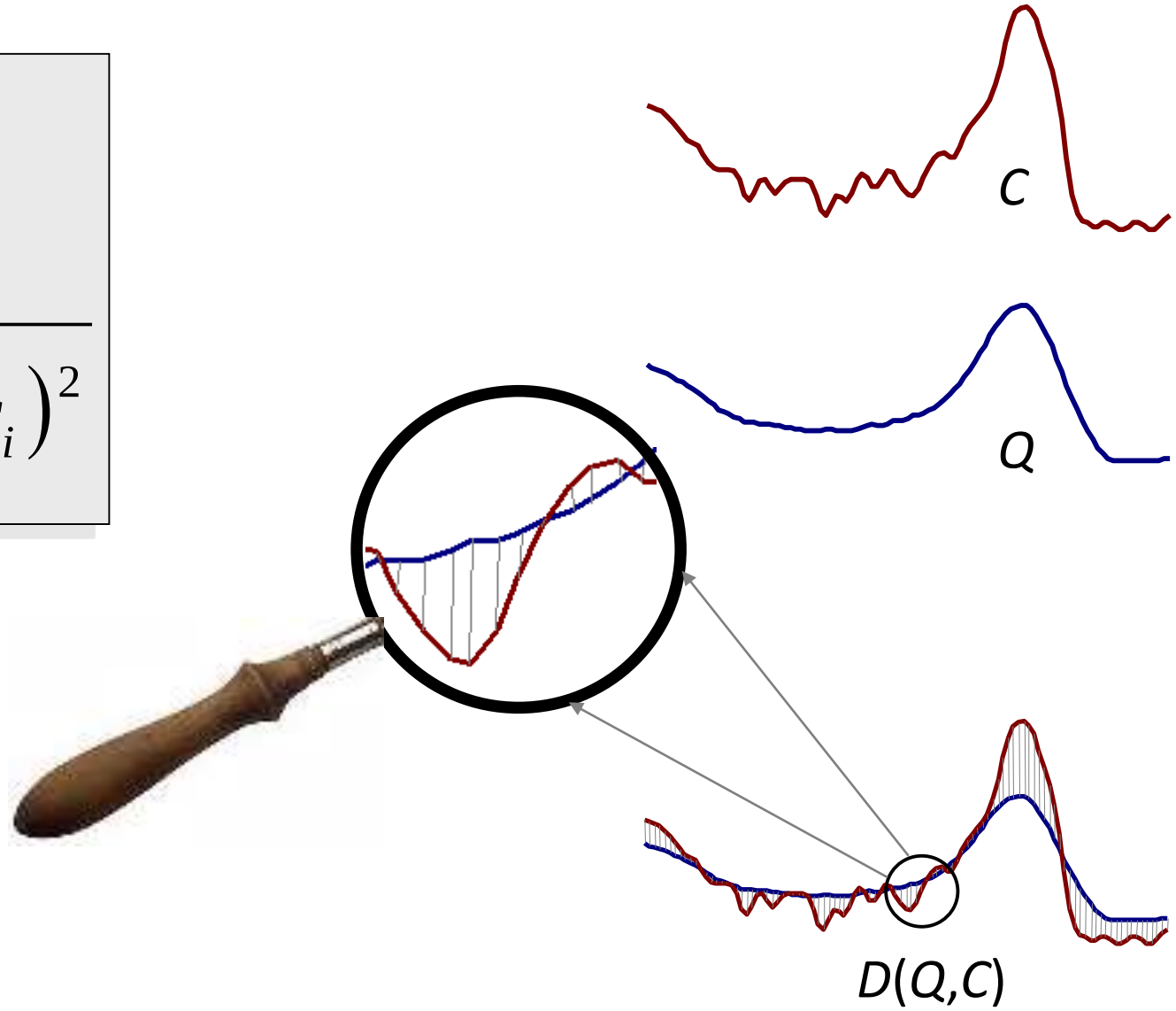
Euclidean Distance Metric

Given two time series:

$Q = q_1 \ldots q_n$

$C = c_1 \ldots c_n$

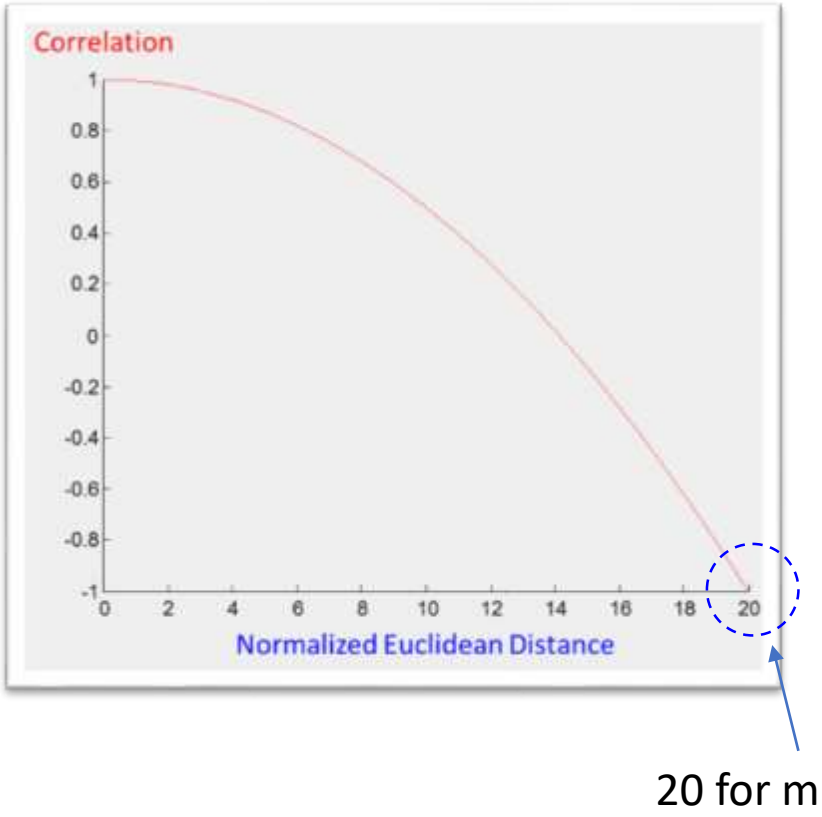
$$D(Q,C) \equiv \sqrt{\sum_{i=1}^{n}(q_i - c_i)^2}$$



Pearson's Correlation Vs Euclidean Distance

$$D(\hat{\boldsymbol{x}}, \hat{\boldsymbol{y}}) = \sqrt{2m(1 - corr(\boldsymbol{x}, \boldsymbol{y}))}$$

- Correlation coefficient does not obey triangular inequality, while Euclidean distance does
- Maximizing correlation coefficient can be achieved by minimizing z-normalized Euclidean distance and vice versa
- Correlation coefficient is bounded between -1 and 1, while z-normalized Euclidean distance is bounded between zero and a positive number dependent on m

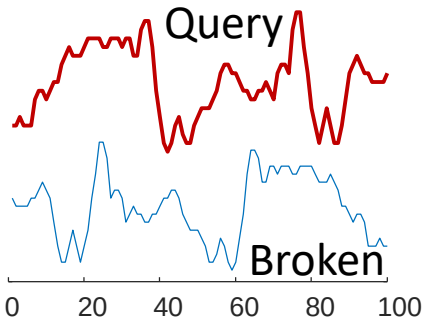


20 for m = 100

For any clustering, classificaon, motif discovery algorithm etc., both will give the same answer

Subsequence Search

While we often compare two time series of the same length, to get a single number, like this…



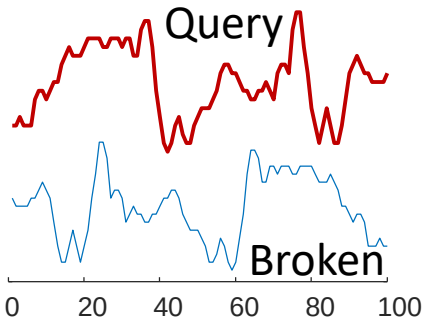
We can do this with MASS…

```
Distance  = MASS(Broken, Query);
```

MASS will return the Euclidean distance between the two z-normalized versions of the two equal size time series.

Subsequence Search

While we often compare two time series of the same length, to get a single number, like this…

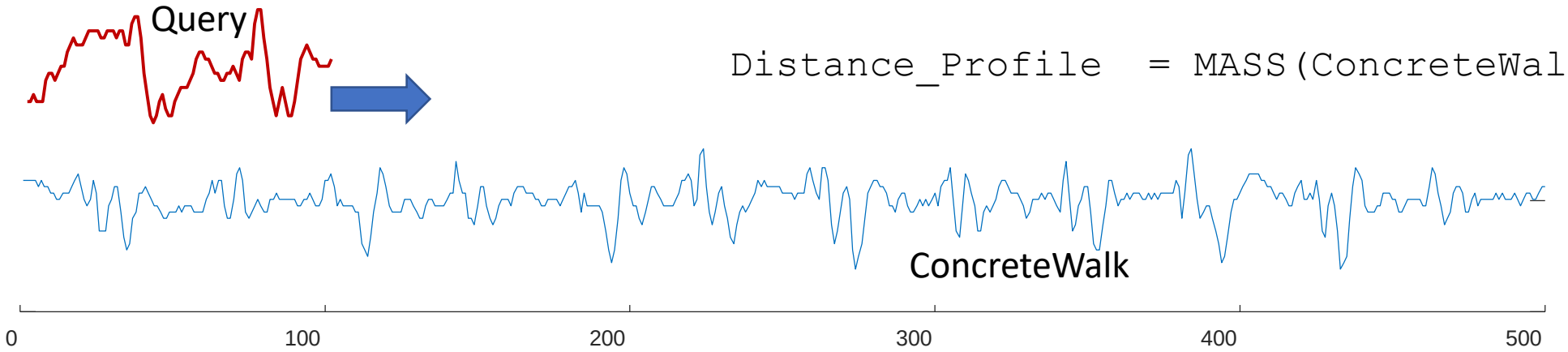


We can do this with MASS…

```
Distance  = MASS(Broken, Query);
```

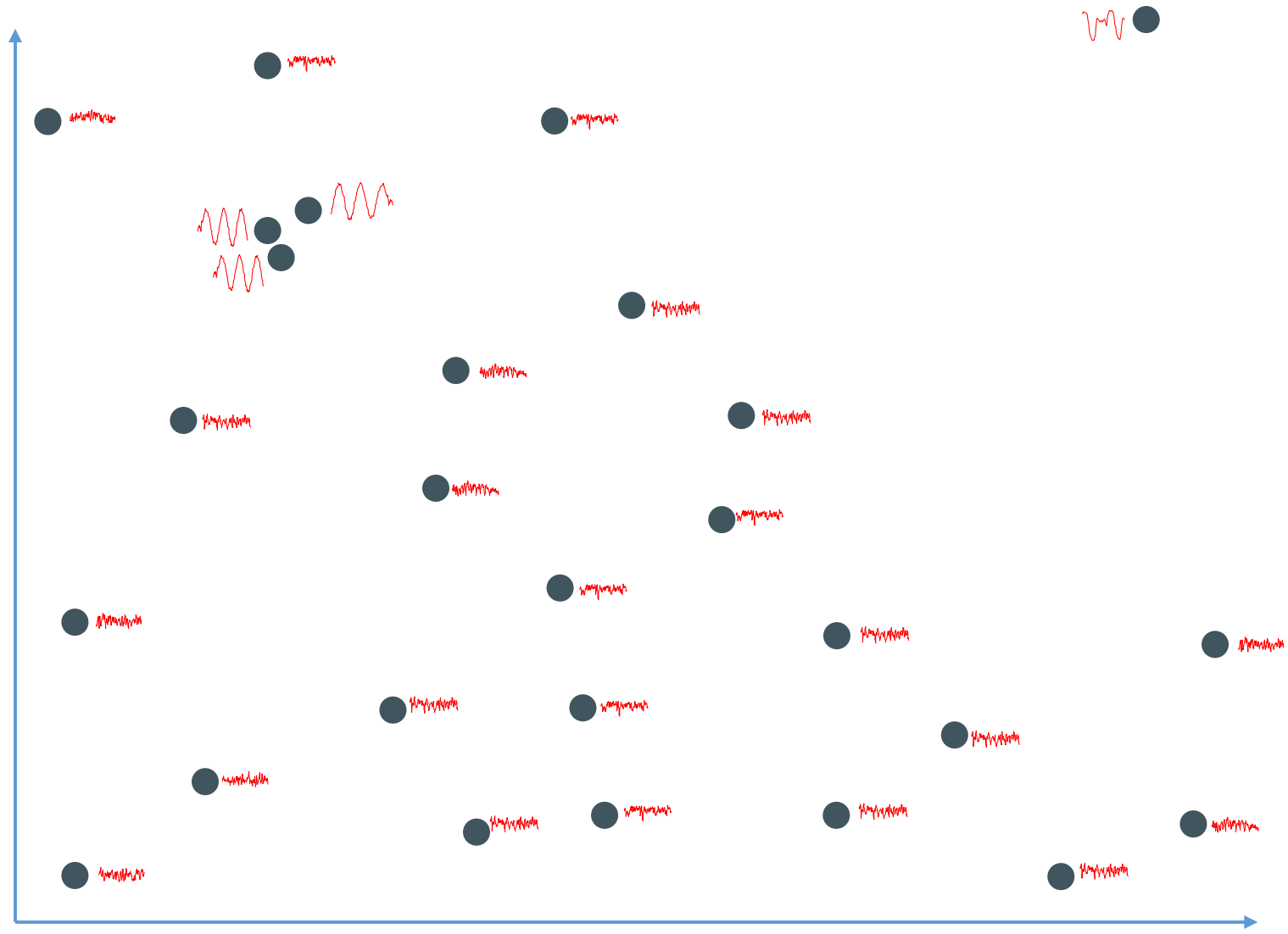
MASS will return the Euclidean distance between the two z-normalized versions of the two equal size time series.

However, a very common task is *subsequence search*. Comparing a short time series query against every possible offset in a much longer time series. MASS can also compute this:

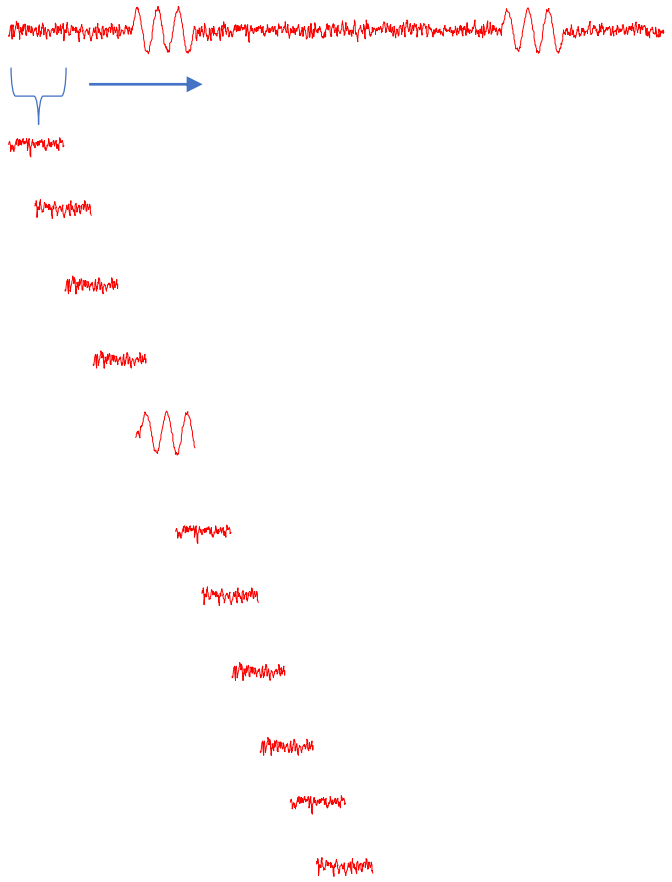



```
Distance_Profile  = MASS(ConcreteWalk, Query);
```

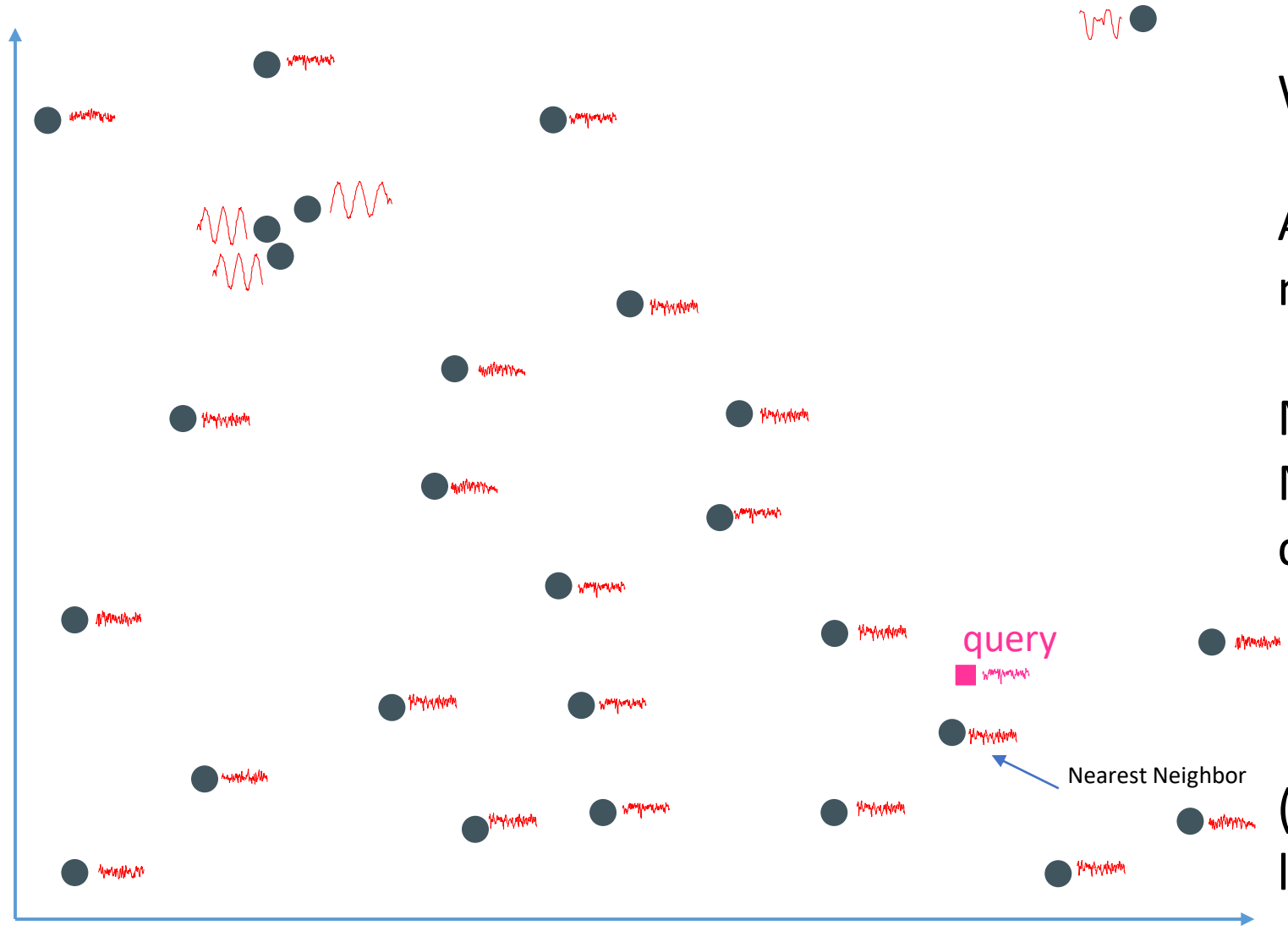
A Minor Visual Mapping Trick



It is sometime useful to think of time series subsequences as points in m-dimensional space.



Subsequence Search



In this world view, subsequence search is just this…

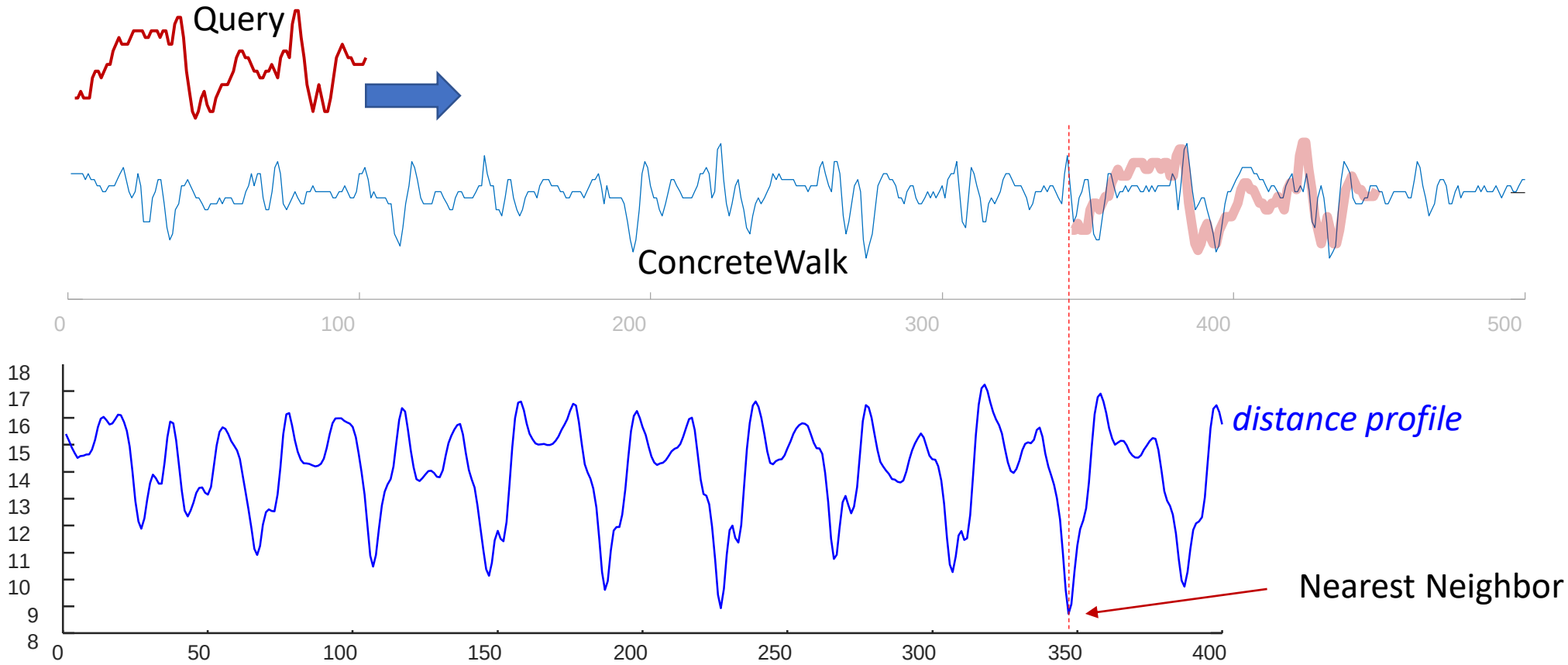
We are given a query, like this

And the task is to find the nearest neighbor…

MASS finds the nearest neighbor. More precisely, MASS finds the distance to all neighbors

(we will revisit this view of the data later)

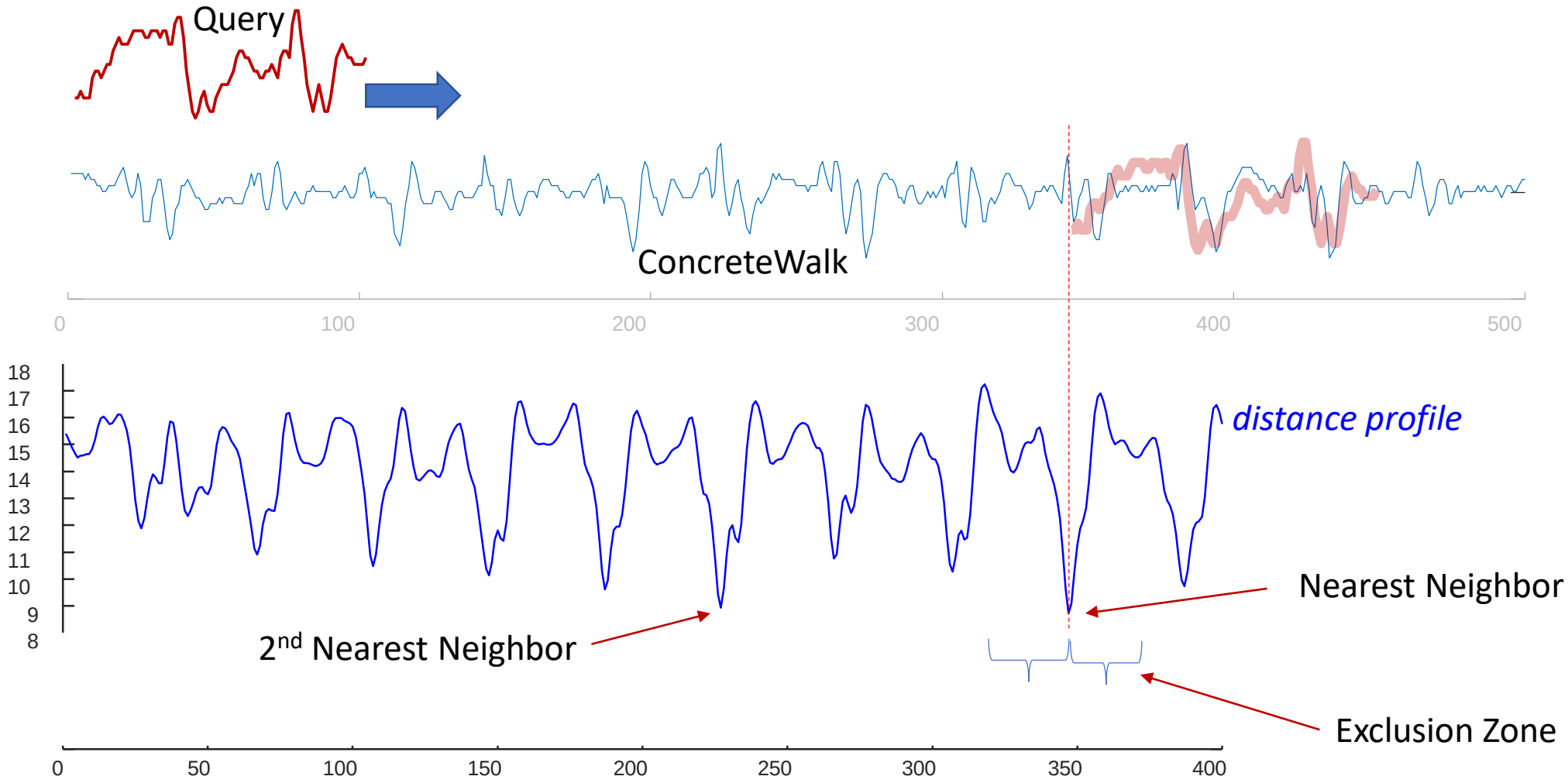
However, a very common task is *subsequence search*. Comparing a short time series query against every possible offset in a much longer time series. MASS can also compute this:




```
Distance_Profile  = MASS(robotdog, carpetquery)
```

If the short time series is of length *n*, and the long one is of length *m*, MASS will return a vector called the *distance profile*, of length m - n +1, which is the Euclidean distance of the query at all possible offsets.

How do we define the second nearest neighbor? If we just use the second lowest point in the distance profile, we will have a trivial match.

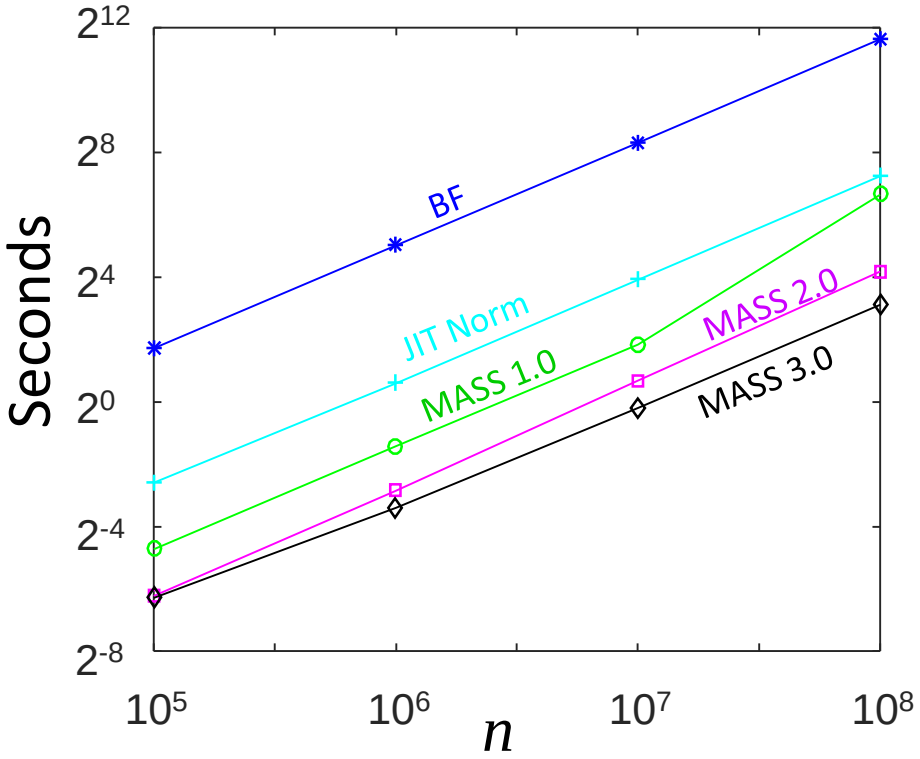


After we find the best match, we set up an exclusion zone to the left and right of it (typically ¼ the length of the query), an exclude it from consideration when looking for the second lowest point in the distance profile.

[a] Properties of MASS I

```
D = MASS(  LongerTimeSeries  ,  ShorterOrEqualLengthTimeSeries)
```

MASS is *very* fast. Suppose the `LongerTimeSeries` is of length 100,000,000, you can compute the distance profile in a few seconds on a commodity desktop, independent of the length of the shorter time series.



- Brute force Euclidean distance in a loop. Best you could do before 2012 1.1 hour
- Optimized with just in time normalization. Best you could do before 2015
- MASS 1.0 in 2015
- MASS 3.0 today 4 Seconds

It is hard to overstate how big of a deal this is. Some of the algorithms I use do a few billion distance measurements, they are trivial with MASS, but untenable without it.

[a] Abdullah Mueen, Yan Zhu, Michael Yeh, Kaveh Kamgar, Krishnamurthy Viswanathan, Chetan Kumar Gupta and Eamonn Keogh (2015), The Fastest Similarity Search Algorithm for Time Series Subsequences under Euclidean Distance, URL: http://www.cs.unm.edu/~mueen/FastestSimilaritySearch.html

[a] Properties of MASS II

```
D = MASS(  LongerTimeSeries  ,  ShorterOrEqualLengthTimeSeries)
```

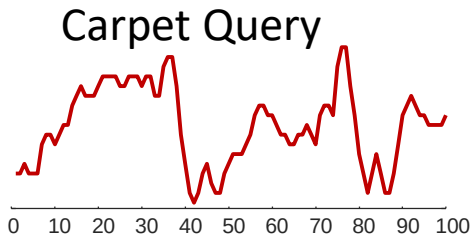
MASS can also support

- Weighted Euclidean distance (example below)
- Both z-normalized and non-normalized Euclidean distance
- Pearson correlation

MASS has been ported to GPUs, you can search a billion datapoints in a second.

Case Study *Have we ever seen a pattern that looks just like this?*

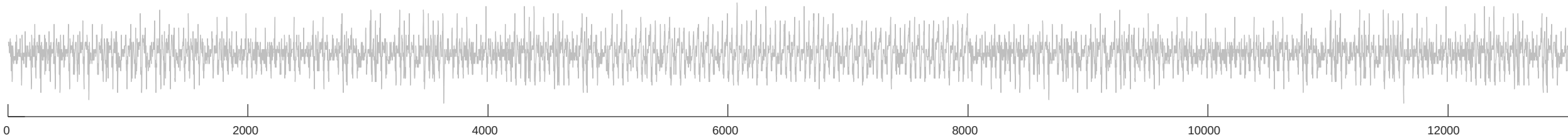
Lets do our first simple case study



- Suppose you know that this shape (*Carpet Query*) corresponds to a robot walking on carpet. The data is from the x-axis accelerometer mounted on the robot.

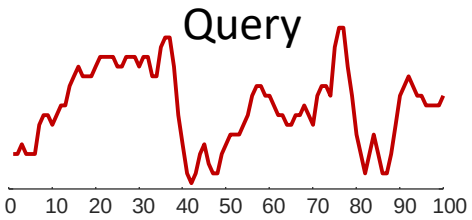
(*Where* you might get such a query shape, we will discuss later)

- You have a robot, that is only supposed to walk on concrete factory floor, not on carpet. Yesterday, it produced this time series…

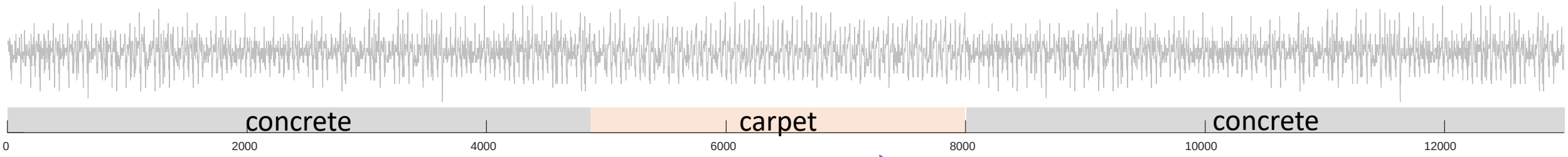


- Did it break the rule, and walk on the carpet yesterday?

Have we ever seen a pattern that looks just like this?



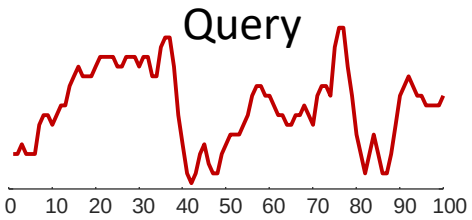
The dataset comes from an accelerometer inside a Sony AIBO robot dog. The query comes from a period when the dog was walking on carpet, the test data we will search comes from a time the robot walked on cement (for 5000 data points), then carpet (for 3000 data points), then back onto cement.



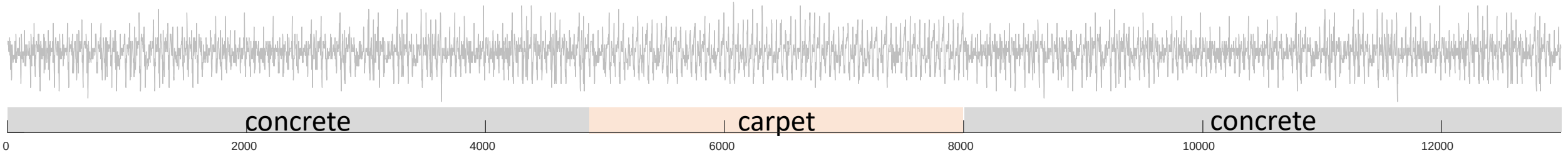
This is the ground truth, but pretend you do not know that…

Let's search for the top 16 matches…

Have we ever seen a pattern that looks just like this?

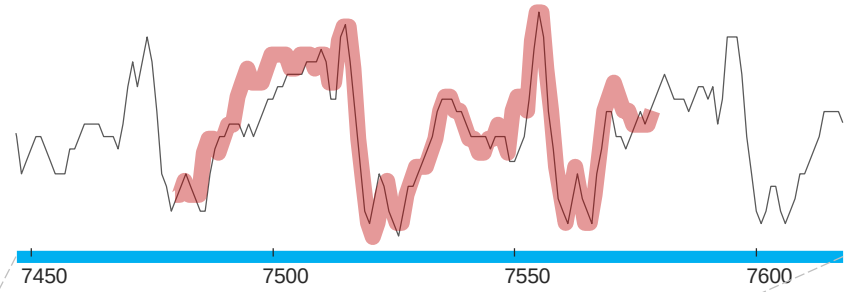


The dataset comes from an accelerometer inside a Sony AIBO robot dog. The query comes from a period when the dog was walking on carpet, the test data we will search comes from a time the robot walked on cement (for 5000 data points), then carpet (for 3000 data points), then back onto cement.



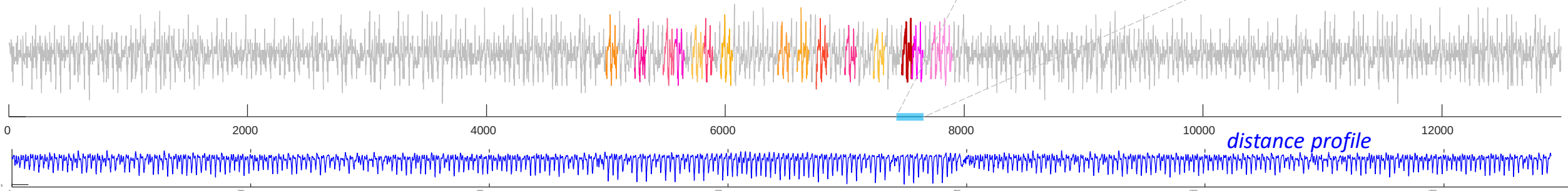
This task is trivial with MASS code…

```
>> load robot_dog.txt , load carpet_query.txt % load the data
>> dist = MASS(robot_dog ,carpet_query );     % compute a distance profile
>> [val loc] = min(dist);                     % find location of match
>> disp(['The best matching subsequence starts at ',num2str(loc)])
The best matching subsequence starts at 7479
```



The best match, shown in context

Below we plot the *16* best matches. Note that they all occur during the carpet walking period. This entire process takes about 1/10,000th of a second.



Can MASS do multi-dimensional queries?

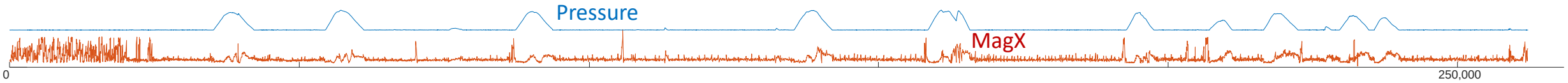
Yes! In fact, this is trivial!

Because we are working in the z-normalized space, we can just add distance profiles together, even if they are different units, for example *watts* and m^3/s

One caveat, while in principle you can do any number of dimensions, small amount of noise and differences add up exponentially fast.

So, in practice, I rarely see this as being useful for more than 2 to 4 dimensions.

Lets do another case study…

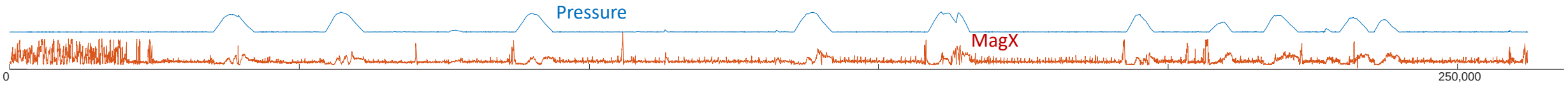


I have 262,144 data points that record a penguin's orientation (MagX) and the water Pressure as he hunts for fish.

Question: Does he ever change his bearing leftwards as he reaches the apex of his dive?

This time I don't have a user supplied template; I need to make the query from first principles

What does "change his bearing leftwards as he reaches the apex of his dive" look like?



I have 262,144 data points that record a penguin's orientation (MagX) and the pressure as he hunts for fish.

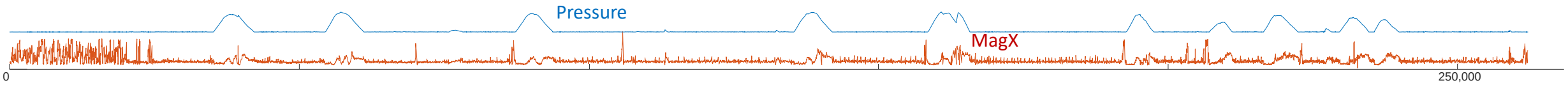
Question: Does he ever change his bearing leftwards as he reaches the apex of his dive.

This is easy to describe as a multidimensional search. The apex of a dive is just an approximately parabolic shape. I can create this with `query_pressure = zscore([[-500:500].^2]*-1)';` it looks like this 

I can create *bearing leftwards* with a straight rising line, like this `query_MagX = zscore([[-500:500]])';` It looks like this 

Done!

All we need now is to search for our query 



I have 262,144 data points that record a penguin's orientation (MagX) and the water/air pressure as he hunts for fish.

Question: Does he ever change his bearing leftwards as he reaches the apex of his dive.

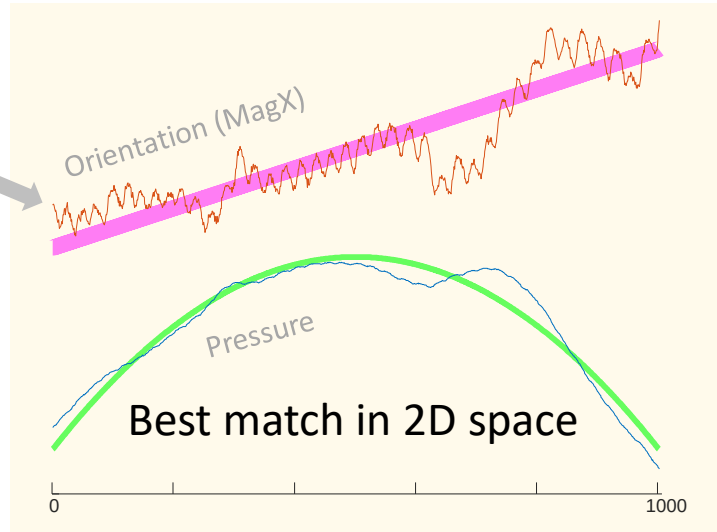
This is easy to describe as a multidimensional search. The apex of a dive is just an approximately parabolic shape. I can create this with `query_pressure = zscore([[-500:500].^2]*-1)';` it looks like this

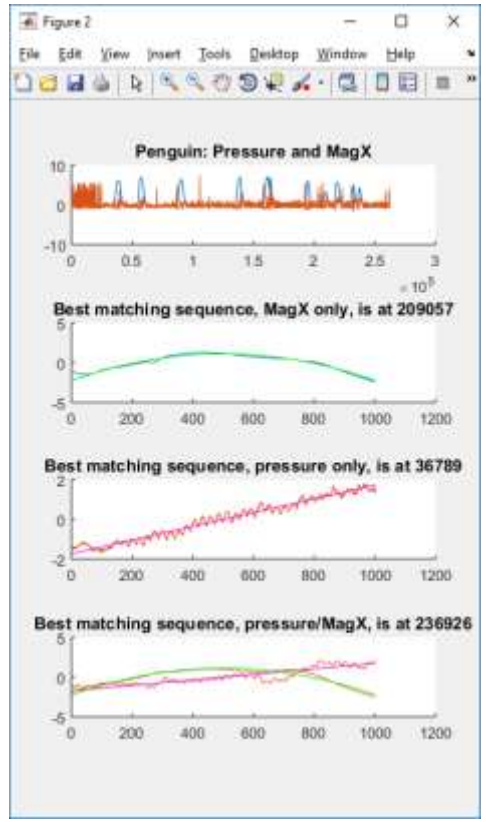
I can create *bearing leftwards* with a straight rising line, like this `query_MagX = zscore([[-500:500]])';` It looks like this

We have seen elsewhere in this document how to search for a 1D pattern. For this 2D case, all we have to do is add the two distance profiles together, before we find the minimum value.

Note that the best match location in 2D is different to either of the 1D queries.

What are the periodic bumps? They are wingstokes as the bird "flies" underwater






```
load penguintest.mat
figure;, hold on;

  query_pressure = zscore([[-500:500].^2]*-1)';
  dist_p = MASS_V2(penguintest(:,1),query_pressure);
  query_MagX = zscore([[-500:500]])';
  dist_m = MASS_V2(penguintest(:,2),query_MagX);
  [val,loc] = min([dist_m + dist_p]);          % find best match location in 2D
  plot(zscore(penguintest(loc:loc+length(query_MagX),2)),'color',[0.85 0.32 0.09])
  plot(zscore(query_MagX),'m')
  plot(zscore(penguintest(loc:loc+length(query_pressure),1)),'b')
  plot(zscore(query_pressure),'g')
  title(['Best matching sequence, pressure/MagX, is at ', num2str(loc)])
```

Mini Review

- Subsequence search is a *very* useful primitive.

- It lets you answer question like "*when have I seen them before*?"
- MASS lets you do it, effectively infinitely fast for most settings.
- The output of MASS, the *distance profiles*, can often be simply summed in various ways to create more exotic queries.
- Sometimes you might get the query template "by-eye" (*robot dog*), sometimes by first principles (*penguin*).
- Motif discovery (upcoming) can also provide query templates.

Mini Review

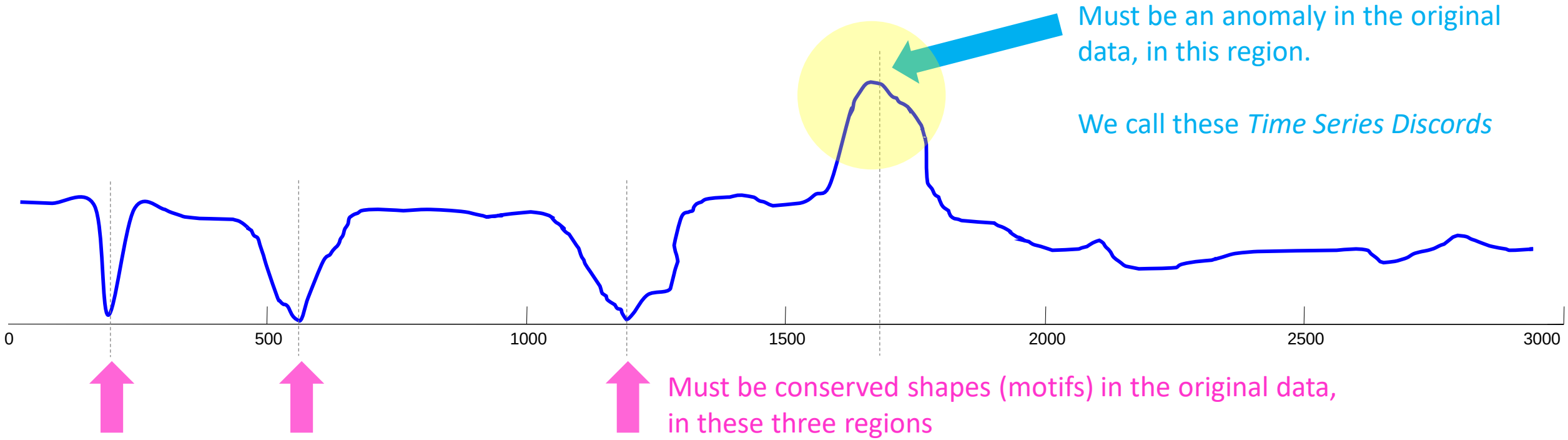
- Motif search is a *very* useful primitive.

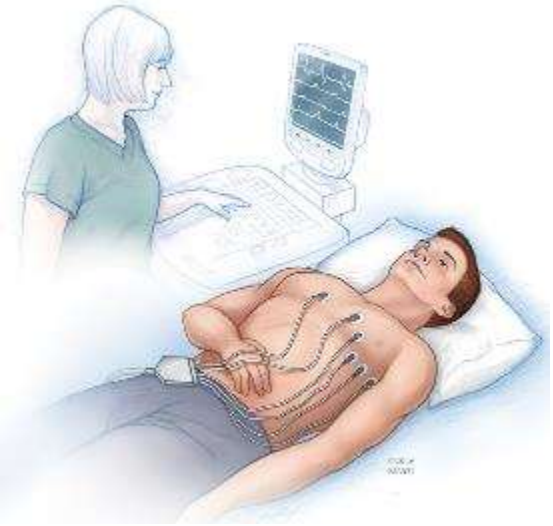
- Motifs can be desirable or undesirable behaviors.
- Motifs are conserved behavior. If something is conserved, there must be a mechanism that causes the conservation.
- Finding/understanding such mechanisms (which may be latent or cryptic) is often our goal

Reading the Matrix Profile

Next 15 minutes: 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)

Another 15 minutes: 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).



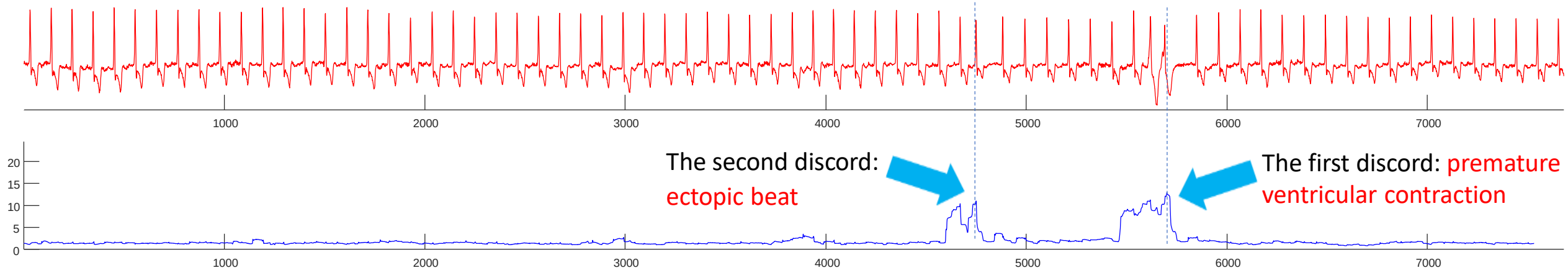


Electrocardiogram

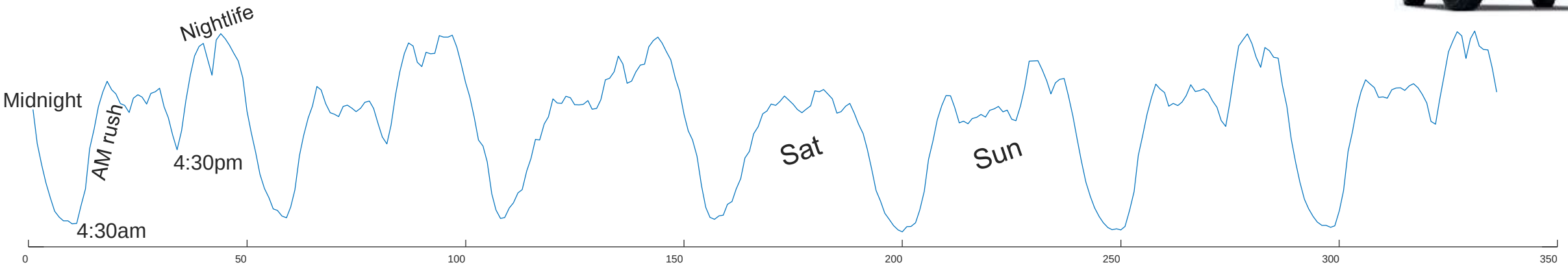
(MIT-BIH Long-Term ECG Database)

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

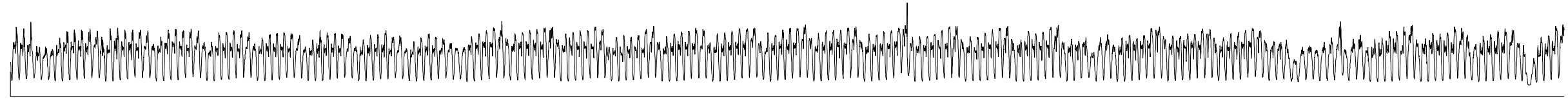
Here the subsequence length was set to 150, but we still find these anomalies if we *half* or *triple* that length.



New York makes all its taxi information public.
Here is a random week



New York Taxi Demand



July 1st (2014)

Jan 31st (2015)

Given a large chunk of it, how can we make sense of it?

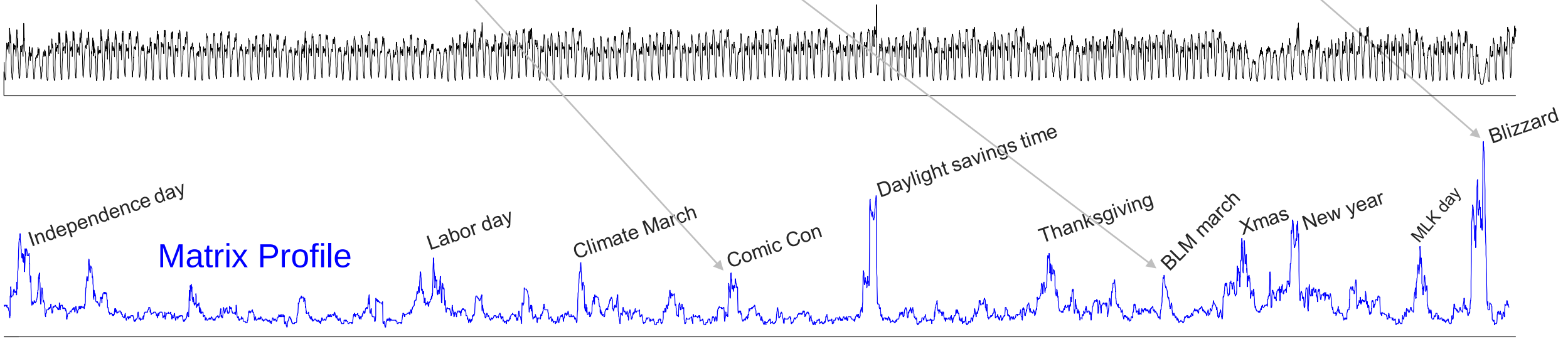
One idea is to compute the Matrix Profile, to find the most unusual patterns….

. It is nice that the convention is growing to cover all sorts of fandoms inside this world. .... However, that would just be another day of traffic jams and over population in the city.

More than 25,000 people marched through Manhattan on Saturday, police officials said, in the largest protest in New York City since a grand jury declined this month to indict an officer in the death of an unarmed black man on Staten Island.

Late Monday, driving bans had taken effect across New York City. Subway and bus service were suspended, and the Port Authority of New York and New Jersey closed Hudson River crossings. Thousands of flights were grounded, public transportation was suspended or curtailed, and travel bans were put in place in the half-dozen states in the path of the storm.

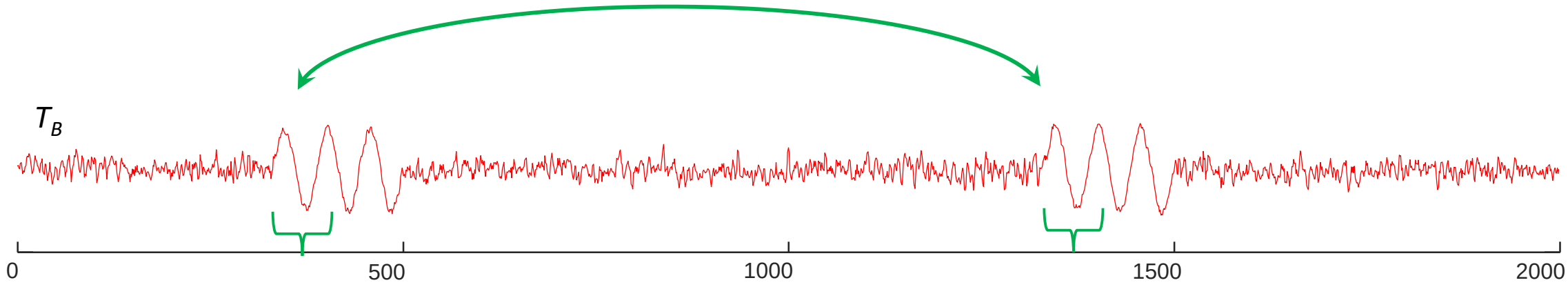
New York Taxi Demand



July 1st (2014)

Jan 31st (2015)

Up to this point, the Matrix Profile has been a *self-join* of a time series…..

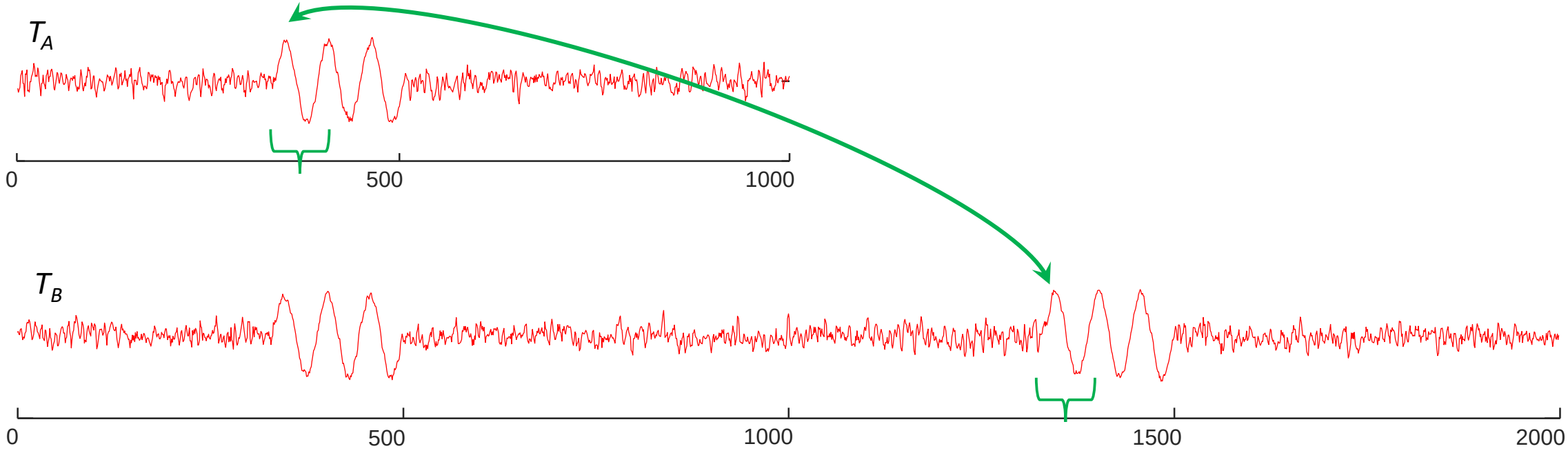


$| T_B | = 2{,}000$

However, we can also use the Matrix Profile to join two separate time series, an AB-join

Assume we have two time series T_A and T_B ...

Note that they can be of different lengths



$| T_A | = 1{,}000$ $| T_B | = 2{,}000$