

UC Riverside

UC Riverside Electronic Theses and Dissertations

Title

Two Novel Primitives Availing Contrast for Time Series Data

Permalink

<https://escholarship.org/uc/item/3tr7m9mw>

Author

Mercer, Ryan

Publication Date

2023

Copyright Information

This work is made available under the terms of a Creative Commons Attribution License, available at <https://creativecommons.org/licenses/by/4.0/>

Peer reviewed|Thesis/dissertation

UNIVERSITY OF CALIFORNIA
RIVERSIDE

Two Novel Primitives Availing Contrast for Time Series Data

A Dissertation submitted in partial satisfaction
of the requirements for the degree of

Doctor of Philosophy

in

Computer Science

by

Ryan Matthew Mercer

December 2023

Dissertation Committee:

Dr. Eamonn Keogh, Chairperson

Dr. Stefano Lonardi

Dr. Vagelis Papalexakis

Dr. Christian Shelton

Copyright by
Ryan Matthew Mercer
2023

The Dissertation of Ryan Matthew Mercer is approved:

Committee Chairperson

University of California, Riverside

Acknowledgement

I cannot imagine a better advisor for myself than Dr. Eamonn Keogh. This may have been the most satisfying experience of my life.

I graciously thank Dr. Stefano Lonardi, Dr. Vagelis Papalexakis, and Dr. Christian Shelton for their guidance during my PhD degree and Dr. Simeon Bird for being a committee member in my oral qualifying exam.

I acknowledge funding from NSF award IIS 2103976.

I am thankful for my wonderful colleagues and coauthors at Dr. Keogh's Data Mining Lab for their support and friendship: Kaveh Kamgar, Audrey Der, Renjie Wu, Dr.

Amirali Darvishzadeh, Dr. Sadaf Tafazoli, Dr. Frank Madrid, Dr. Shima Imani, Dr. Zach Zimmerman, Dr. Sara Alaei, Dr. Shaghayegh Gharghabi, Dr. Anh Dau, and Dr. Michael Yeh.

There have been a number of friends who have tackled their PhD before me who have given me inspiration and guidance: Dr. Joseph Hughes, Dr. Cameron Yamanishi, Dr. Kunal Shah, Dr. Michael Rafuse, and Dr. Perla Ontiveros- Ángel. Without your inspiration I may have settled for less.

Mr. Stone, peanut gallery facilitator, you are likely the only one to read this.

My best friends: Jordi and Perla, you have been there in hard times.

Finally, I thank my family. My wife, my jaan, your blub sustains me. My dad cultivated my interests without pushing his own. Mom, you have always supported my education.

In the name of he who guideth me unto prosperity, Eamonn.

ABSTRACT OF THE DISSERTATION

Two Novel Primitives Availing Contrast for Time Series Data

by

Ryan Matthew Mercer

Doctor of Philosophy, Graduate Program in Computer Science

University of California, Riverside, December 2023

Dr. Eamonn Keogh, Chairperson

Time series data is a fundamental datatype in data mining, with its understanding deepened over the past decade through analyzing subsequences based on their similarity or dissimilarity to other data. Two known concepts in this realm are motifs, which are closely resembling subsequences, and discords, which are subsequences distant from their nearest neighbors. The unique characteristic of time series data is its amenability to real-time interventions besides offline analysis. For instance, monitoring an industrial process in real-time can lead to immediate actions to prevent imminent failures, based on predictive algorithms. Various tools are available to monitor and analyze time series data for known patterns, evolving behaviors, or unexpected anomalies which are crucial for pattern matching and anomaly detection.

In this dissertation, a novel analytical tool called the Contrast Profile is introduced. This tool facilitates a balanced examination of a subsequence's similarity and dissimilarity to

certain data, and is demonstrated to have numerous applications including anomaly detection, data exploration, and preprocessing unstructured data for classification. The Contrast Profile enables end-to-end classification in large datasets and helps in unveiling subtle patterns that might otherwise go unnoticed. The application of the Contrast Profile is showcased across a variety of domains including seismology, animal behavior, and cardiology, demonstrating its versatility and utility in different fields.

Moreover, a new primitive termed Novelets is introduced to monitor emerging behaviors in time series data. Novelets, unlike anomalies, chains, or motifs, initially appear as anomalies but are later identified as motifs. They have a broad relevance across multiple fields like science, medicine, and industry, aiding in prognostics and abnormal behavior detection. The concept of Novelets is demonstrated through various challenging domains, underlining its potential in enhancing the understanding and monitoring of emerging behaviors in time series data. Through the Contrast Profile and Novelets, this dissertation aims to significantly contribute to the field of time series data analysis, paving the way for more insightful data exploration and real-time interventions across a myriad of disciplines.

Table of Contents

1. Introduction	1
1.1 A time series primitive highlighting contrasting behaviors	2
1.2 A time series primitive highlighting emerging behaviors	4
2. The Contrast Profile: A Novel Time Series Primitive that Allows Real World Classification	8
2.1 Definitions And Notation	8
2.1.1 The Contrast Profile’s Robustness	15
2.1.2 The Contrast Profile Assumptions	17
2.1.3 Further Relaxations of the Contrast Profile Assumptions	19
2.1.4 A Case Study of the Relative Frequency Contrast Profile	23
2.1.5 General Contrast Profile Observations	26
2.1.6 Online Contrast Profile	30
2.1.7 Anytime Contrast Profile	33
2.2 Related Work	34
2.3 Algorithms That Exploit The Contrast Profile	37
2.3.1 End-to-End Time Series Classification	37
2.3.2 Exploratory Data Mining	38
2.3.3 Anomaly Detection: Solving the Twin Freak Problem	39
2.4 Experimental Evaluation	42
2.4.1 Insect Behavior Classification	42
2.4.2 Chicken Behavior Classification	44
2.4.3 Electrocardiogram Classification	46
2.4.4 Electrocardiogram Classification with Relative Frequency Contrast Profile	48
2.4.5 Electrocardiogram Classification with Top-K Platos	49
2.4.6 Detection of Novel Earthquake Events	51
2.5 Model Comparison	55
2.5.1 Comparison to Shapelets	56
2.5.2 Towards Robust Classification: A More Forceful Comparison to Shapelets	57
2.5.3 Comparison to LSTM	60

2.6 Conclusions	61
3. Novelets: A New Primitive that Allows Online Detection of Emerging Behaviors in Time Series	62
3.1 Definitions and Notation	63
3.1.1 Differentiating Novelets from Related Concepts.....	70
3.1.2 Related Work	73
3.1.3 The Novelet Assumptions	74
3.1.4 General Novelet Observations	75
3.1.5 Online Novelets	78
3.1.6 Additional Implementation Details.....	83
3.2 Exploratory Data Mining	90
3.2.1 Electrooculogram.....	90
3.2.2 Pedestrian Traffic.....	91
3.3 Experimental Evaluation.....	93
3.3.1 Bird Song Detection I.....	94
3.3.2 Bird Song Detection II	96
3.3.3 Industrial Process Monitoring: Bearings.....	99
3.3.4 Industrial Process Monitoring: Buggy Machine	104
3.4 Model Comparison	106
3.4.1 Comparison to Segmentation.....	106
3.5 Conclusions	107
4. Conclusions	109
Bibliography.....	111

List of Figures

Fig 1 Two short snippets of behavior from a chicken wearing a backpack accelerometer. The bottom time series is known to contain at least two examples of dustbathing behavior, whereas the top time series is known to be free of this behavior.....	3
Fig 2 A visual intuition of the “contrast” property. Of the three annotated points from the target class, only C is close to a member of its own class, while also being far from its nearest neighbor in the non-target class.....	4
Fig 3 A short respiration data snippet featuring normal, noisy, and emergent behaviors. Neither motifs nor discords capture either of the two instances of emergent behavior (highlighted in red). The Emergence Profile peaks at the <i>second</i> instance of the emergent behavior, as expected. The explanation of the cause and medical implications of this Novelet are available at [9].	5
Fig 4 <i>top</i>) A 27-second snippet of an ECG time series. <i>bottom</i>) A single heartbeat from earlier in the same dataset was used as a query to produce a distance profile, which has low values when the “sliding” query is similar to a subsequence and is minimized at the best match about five seconds in.	9
Fig 5 <i>top</i>) The ECG shown in Fig 4 with its MP128 (<i>bottom</i>). The lowest values of MP128 are the Top-1 motif pair, here two normal beats. Also, two PVCs shown highlighted with red bars for future reference.	10
Fig 6 <i>top</i>) Time series TB is a normal ECG time series from the same patient. <i>center</i>) Time series TA, which contains the behavior of interest, is the original ECG introduced in Fig 4. <i>bottom</i>) The top motif pair, where motifA is the unrequited nearest neighbor of motifB. The red bars foreshadow discovery of two PVCs.	11
Fig 7 <i>top-to-bottom</i>) Query time series T – contains normal heartbeats. Time series T + contains at least two instances of a behavior of interest. The top discord of the AB-join Matrix Profile (the highest peak), results from a noisy region in T +, far from the ground truth labeled with red bars. The top two candidates peak within the ground truth.	13
Fig 8 <i>top</i>) Two ECGs and their Contrast Profile. <i>bottom</i>) After adding and concatenating a severe magnitude of random noise to the two time series, the peak locations between the original and new CP are virtually identical, and the concatenated section has low values, meaning it will not cause spurious matches.....	16
Fig 9 A visual intuition of the “relative frequency contrast” property. Of the three annotated points from the target class, C has the largest contrast between the K^{th} neighbor in the target class and the K^{th} neighbor in the non-target class, which happens to be $K=3$	20
Fig 10 Time series consisting exclusively of a single sex were constructed by concatenating each sample’s heave axis, separated by NaN.....	23
Fig 11 Female 1-NN Female Plato and Male 1-NN Plato.....	24

Fig 12 Concatenated Female behaviors shown alongside heatmap of RFCP.....25

Fig 13 Zoomed in region of concatenated Female behaviors shown alongside heatmap of RFCP. The heatmap highlights a behavior occurring at least ten times, exclusively in the time series collected from females.25

Fig 14 *left*) The Pan-Contrast Profile for the example shown in Fig 8. A red dot indicates the largest value. *right*) A side view shows that the Contrast Profile is very robust to its only input parameter. Any subsequence length from 131 to 424 would have produced a score of at least 0.7.28

Fig 15 Anytime convergence plot for the ECG data shown in Fig 22.34

Fig 16 *top*) Two four-year-long time series and their CP12. *bottom*) The two contrasting subsequences shown in context.38

Fig 17 An example of CPAD. *top*) A time series that represents normal behavior of a fridge. *center*) A snippet from the same refrigerator, with the final moments swapped out for a different fridge from the same house. *bottom*) The Contrast Profile strongly spikes to indicate conserved novel behavior.41

Fig 18 *top*) 21 hours of Asian citrus psyllid (ACP) feeding behavior on citrus. *bottom*) A zoom-in of a small fraction of the data.43

Fig 19 *top-to-bottom*) The weakly-labeled instances shown in Fig 18 have their MP40 + –and MP40 + +computed to produce the CP40, which strongly peaks to indicate the location of the Plato.44

Fig 20 *top-to-bottom*) The weakly-labeled instances shown in Fig 19 have their MP120 + –and MP120 + +computed to produce the CP120, which strongly peaks to indicate the location of the Plato.45

Fig 21 The Plato used for dustbathing classification (*top.left*). Selected matches returned by a nearest neighbor search using the Plato discovered in Fig 20. The Top-1000 matches (*bottom.right*).45

Fig 22 *top-to-bottom*) Two weakly-labeled snippets from the first five minutes used to compute their MP91 + –and MP91 + +, which then produce the CP91, with high amplitude peak to indicate the location of the Plato.46

Fig 23 *top-to-bottom*) Two weakly-labeled snippets from first 5 minutes used to compute their RFCP91. The binary heatmap shows 8 floating columns below the PVC instances in T2 +. The low contrast regions below the columns indicates the common number of occurrences of PVC between T2 + and T2 –. The 7-NN is shown alongside the Plato to illustrate the high similarity.48

Fig 24 Investigation of discrepancies in Fig 23 have led to my identifying Plato-2 as a less frequent PVC morphology by using the Top-K Platos functionality of the Contrast Profile.50

Fig 25 In scenarios A and B, the CP100 appropriately ignores previously cataloged activity. In scenarios C and D, CP100 appropriately highlights uncataloged activity.54

Fig 26 Samples of each of the four classes in the Trace dataset overlaid with Plato.56

Fig 27 Accuracy comparison between classifying with RFCP Platos versus fast shapelets.58

Fig 28 Classes composed of a subset of features are difficult to classify using a single Plato, as seen in the MALLAT dataset. *left*) The unobscured first sample of each class. *right*) Classes with common features revealed together.....59

Fig 29 *top*) A short synthetic time series containing three signal types: a sine wave, a modified sine wave, and noise. *bottom*) A segment closely matching the modified sine wave is used as a query to generate distance profile DPAB,200. Lower values indicate subsequence locations which are more similar to the query64

Fig 30 *top*) Time series TB is synthetically created to contain equal lengths of sine wave and noise. *center*) Time series TA, first introduced in Fig 29, contains three signal types. *bottom*) MPAB,200 reveals behaviors common to both TA and TB as low values and behaviors unique to TA as higher values.65

Fig 31 *top*) The synthetic time series TA and its corresponding MPAA,200 (*bottom*). The top motif pair are outlined. The three instances of the altered sine wave, which are highlighted in red, each have a corresponding mid-valued bump in MPAA,200, which excludes them from consideration as either motifs or discords.65

Fig 32 *top*) My running example time series. *bottom*) The left Matrix Profile LMPAA,*m* will highlight the first instance of a repeated behavior as an anomaly, but the first instance of the modified sine is still a moderate distance value, which excludes it from classification as a top discord compared to noise.....66

Fig 33 *top*) My running example time series. *middle*) By subtracting LMP + +,200 from MP + -,200, the second instance of the emerging modified sine wave is clearly highlighted within EP200(*bottom*). Once a new behavior is learned, future instances are automatically suppressed by adding the first emergence to T -68

Fig 34 *top*) The first and second instances of the emerging modified sine wave are outlined in T +. *bottom*) The third instance of the emerging behavior has been suppressed in EP*m* by logically appending the first instance to T - (not shown).69

Fig 35 *left*) A surface plot of the Emergence Profile which contains two high-valued ridges corresponding to the second and third PVC instances. *right*) All but the smallest subsequence lengths will result in detection of the PVC beats, suggesting that Novelets are highly robust to at least one of the two user-supplied parameters.77

Fig 36 *left*) For various sampling rates used in this dissertation, I show the theoretical MTH as the point where the sampling rate curve descends below a real time gain of 1.0 (gray, dashed). *right*) I also explore the effect of buffer length on run-time gain.82

Fig 37 A demo Emergence Profile EP undergoing peak detection. The top peak P1 is first identified, then an exclusion zone of length m is placed on either side of the index. The next top peak outside of the exclusion zone P2 is identified. The leftmost peak is selected as the Novelet if it exceeds the novelty threshold, though it does not here.....87

Fig 38 When considering the synthetic time series introduced in Fig 29, the Emergence Profile which stores Novelets without context is unable to sufficiently suppress the third instance of the Novelet. Increasing the context to between $m/2$ to m can suppress rediscovery of future instances.....88

Fig 39 *top*) One of the top ten EOG Novelets corresponds to an unexpectedly long sequence of eye-blinks. Context in gray. *bottom*) A three-minute snippet from a twenty-two-hour EOG recording. The highlighted region of interest is a highly periodic pattern in an otherwise visually noisy time series.91

Fig 40 *top*) The Novelet and its nearest neighbor. *bottom*) Distance profile between the three-year time series and the Novelet. The distance profile reveals one repetition of the Novelet.....92

Fig 41 *Top*) The first half of the audio is set to T – so that future narration in T + is suppressed. *Bottom*) The Emergence profile peaks in four locations above 0.15.95

Fig 42 Histogram of non-overlapping peaks. A novelty threshold value of 0.15 gives a wide margin between the cluster of low-novelty peaks and the peaks with higher novelty.....95

Fig 43 The emergence of the two song compositions of the white-throated sparrow is detected using Novelets. Novelets and their second instances are shown in color with context shown in gray.....96

Fig 44 *top*) Synthesized time series T + with four embedded bird songs with locations indicated by ground truth annotations (red). *center*) Self-join Matrix Profile MP + +,141 with the nearest neighbor distance of the first instance indicated by a dashed line (gray). *bottom*) Emergence Profile EP141 with Novelet indicators at the first (cyan) and third instances (magenta) of the bird song.98

Fig 45 The first and third instances of the embedded bird song, along with context (gray).98

Fig 46 *top*) True positive, false negative, and false positive rates shown for 100 iterations of identifying bird songs in the background within the first two, three, and four song instances. *bottom*) The test repeated for bird songs in the foreground.....99

Fig 47 A histogram of Novelet counts for each test configuration. A majority of configurations have five or more detected Novelets.101

Fig 48 Two discovered Novelets shown with context (gray) and overlaid on their second instance.102

Fig 49 Visualizing the Matrix Profile nearest neighbor offsets reaffirms the observation that Novelets discovered a constant offset to their second occurrence.103

Fig 50 *left*) The first Novelet discovered, shown with context in gray. *right*) Using the Novelet as a query on the accelerometer data produces a distance profile which detects all instances of the six medium and six large ball drops, confirming that the Novelet detected was indeed caused by a collision..... 105

Fig 51 *top*) A time series of Pulsus Paradoxus with a visually non-trivial change point located at index 10,000. *bottom*) A threshold parameter sweep of Novelet indices. There is a stable Novelet index within two subsequence lengths of the ground truth..... 107

List of Tables

Table 1 The RelativeFrequencyContrastProfile Algorithm.....	20
Table 2 The RelativeFrequencyMatrixProfile Algorithm.	21
Table 3 The NearestNeighborSelection Algorithm.....	23
Table 4 The ContrastProfileI Algorithm.....	31
Table 5 MIT-14046 Classification Precision Summary.	51
Table 6 The OnlineNovelets Algorithm	79
Table 7 The BatchMPUpdate Algorithm.....	84

1. Introduction

In order to perform various data mining tasks on time series, it can be fruitful to annotate each subsequence of the time series with metadata indicating various properties. One such feature is a subsequence’s distance to its nearest neighbor within the same dataset. That information can be represented by the Matrix Profile [1]. Small values in the Matrix Profile are called *motifs*, and large values are called *discords*. Both motifs and discords have each been used in hundreds of research efforts. However, I argue that it may be useful to score subsequences with a new piece of meta-data that reflects the property that a subsequence is simultaneously close to its nearest neighbor in certain data but far from its nearest neighbor in other “black-listed” data. I call this property *Contrast*, and the vector that represents it the *Contrast Profile*. While the proposed representation has many uses, for clarity, I will introduce it in the context of subsequence extraction to allow *classification*.

In the last decade, the data mining community has generalized many of its previous *batch-only* algorithms to the more actionable *online* setting. In particular, there now exist mature tools to monitor time series for *known* behaviors (template matching) [2], previously unknown *repeated* behaviors (motifs) [1], evolving behaviors (chains) [3], and *unexpected* behaviors (anomalies or discords) [4]. However, there is almost no work to address the idea of *emerging* behaviors. Perhaps the reason this primitive has escaped the community’s attention is that this idea may *seem* to be encapsulated in other primitives, perhaps change detection, anomaly detection, or motif discovery. However, as I shall explicitly argue in Section 3.1.1, none of these notions captures the general sense of emerging behaviors. An emerging behavior can perhaps be thought of as an anomaly/discord that later transitions

to a motif. This view of emerging behaviors is more than just a mnemonic device, it offers a direction to create an emerging behavior discovery algorithm. This is because there is a widely used and mature framework to monitor for both motifs and discords, the Matrix Profile [1].

In this dissertation, I propose a novel primitive that identifies contrasting behaviors between two times series in Chapter 2. I then propose another novel primitive which highlights emerging behaviors in Chapter 3. Finally, I offer conclusions and directions for future work in Chapter 4. In the following text, I will show the detail of the motivations of each project.

1.1 A time series primitive highlighting contrasting behaviors

While the time series classification community is very active, most research efforts confine their work to datasets from the UCR archive or similar benchmark datasets [5]. However, for most of these archive datasets, the work of extracting the exemplars from a longer time series has already been done. Here, I argue that extracting the exemplars is actually the most difficult and critical task (this argument is more forcefully made in [6]). In a handful of cases, it may be obvious where the beginning and the end of an exemplar is within a longer time series. But, in many cases, these demarcations may not be clear. Consider Fig 1.*bottom*, which shows a time series known to have several examples of chicken dustbathing behavior [7]. Even to experts in avian biomechanics, it is not obvious where the dustbathing behavior is. Moreover, even if I knew the answer to that question from an ethological perspective, there is no guarantee that those subsequences would optimize classification accuracy.

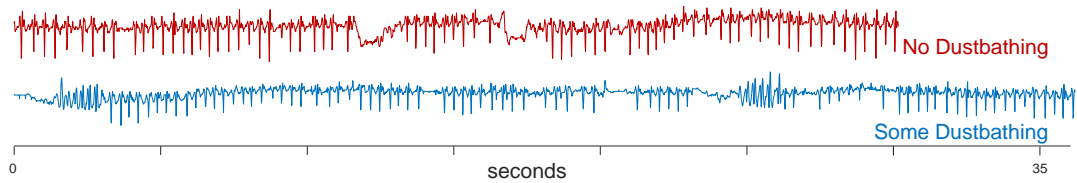


Fig 1 Two short snippets of behavior from a chicken wearing a backpack accelerometer. The bottom time series is known to contain at least two examples of dustbathing behavior, whereas the top time series is known to be free of this behavior.

This suggests that a technique is needed to annotate each subsequence of the time series with a value that simultaneously represents how close that subsequence is to its nearest neighbor *within* the same time series and how far it is from its nearest neighbor in the time series known to be free of the target behavior. This score would reveal the location of the uniquely conserved behavior, in this case, *dustbathing*.

In Fig 2, I give a visual intuition of the property of interest: abstracting time series subsequences to points in a high dimensional space. I explicitly consider three data points.

- 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. It is an anomaly that would score highly on the definition of time series *discord* [8].
- 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. This point would score highly on the definition of time series *motif* [1].
- 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. This is exactly the property I desire. I would like it to score highly on a proposed definition of *contrast*.

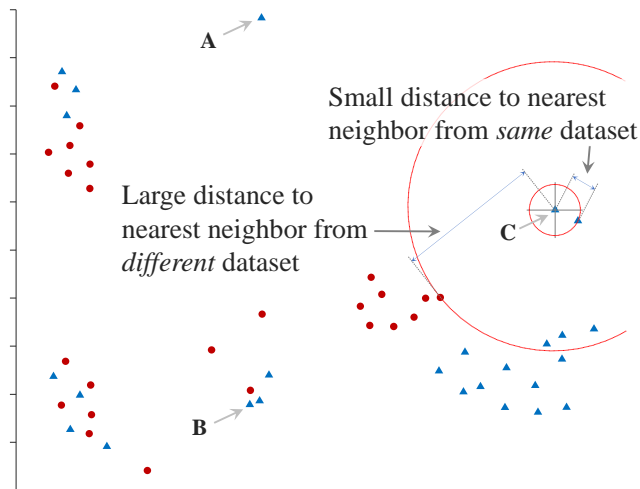




Fig 2 A visual intuition of the “contrast” property. Of the three annotated points from the target class, only C is close to a member of its own class, while also being far from its nearest neighbor in the non-target class.

1.2 A time series primitive highlighting emerging behaviors

In the case of anomaly detection, most techniques focus on identifying behaviors which do not correspond to previously observed behaviors. This can be useful in many cases, but in cases such as patient monitoring where there is often recording artifacts which do not reflect the state of the patient, the noise may act as a red herring which costs too much human attention. In cases this this, there may be a generally accepted healthy behaviors, but new repeated behaviors are of interest. Fig 3 shows an example of an emerging behavior I discovered in a respiration dataset. Here most cycles look like this , a healthy breath cycle. In addition, there are some noisy cycles. However, there is also this unusual shape , that emerges after forty seconds.

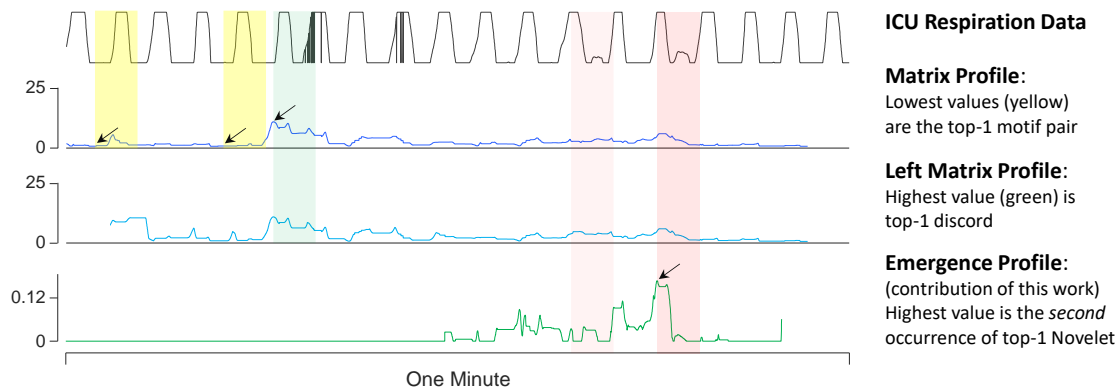


Fig 3 A short respiration data snippet featuring normal, noisy, and emergent behaviors. Neither motifs nor discords capture either of the two instances of emergent behavior (highlighted in red). The Emergence Profile peaks at the second instance of the emergent behavior, as expected. The explanation of the cause and medical implications of this Novelet are available at [9].

Note that while this shape is neither the minimum nor maximum of a Matrix Profile (or left Matrix Profile), I can expand and adapt the Matrix Profile framework to produce an *Emergence Profile*, allowing an emerging behavior discovery algorithm. I call my approach *Novelets*, Newly Observed Variation while Excluding Learned and Established Time Series.

To help make the actionability of Novelets clear, consider the following domains:

- **Medicine:** Much of medical telemetry is replete with motifs (i.e., repeated heartbeats or respiration cycles), and with discords (i.e., sensors artifacts, movement artifacts). In the presence of such data, new patterns can emerge, either naturally or as the result of an intervention. For example, in an ICU a doctor may give her patient Digoxin. This drug can change the shape of the heartbeat, generally for the better, but sometimes in negative ways. Moreover, it can take 0.5 to 6 hours to produce an effect [10]. The doctor may wish to receive an alert the moment a new heart rhythm emerges.
- **Science:** The white-throated sparrow (*Zonotrichia albicollis*) is a bird that is found mostly in Canada, and pleasingly, its distinctive song rhythm is often transcribed as “O-

oh sweet Canada, Canada, Canada". However, beginning in the early 2000s a new unexpected song variant emerged, roughly transcribed as "*O-oh sweet Cana, Cana, Cana*" [11]. Moreover, this newly emerged variant began to spread across the country, creating a flurry of academic interest [12]. In this case, the discovery of the emerging behavior was accidental [11], but one could imagine an algorithmic effort to discover novelty in bird songs. In a similar vein, there are literally thousands of research papers on the emergence of Zebra finch (*Taeniopygia guttata*) vocalizations [13][14]. Because birdsong development is a highly accessible example of gene-culture interaction it is useful for teasing out elements of the nature vs. nurture debate. However, surprisingly much of the work in emerging vocalizations still relies on tedious manual transcription [15].

- **Industry:** Industrial assets are often heavily monitored. For example, a single gas distillation column may have over one thousand sensors. For batch processes only a handful of patterns are normally observed, corresponding to different recipes being observed. Anomalous data is often associated with a spoiled batch. However, an emerging behavior is suggestive of an undocumented and unexpected change. For example, an operator may be unconsciously compensating for degrading infeed product by allowing a pasteurizing regime to run longer than normal [16]. The plant manager would like to be made aware of such emerging patterns.

In [17] I offered a brief introduction to some of the ideas I develop in this dissertation. In particular, in Section 2.3 I consider several real-world examples that closely model these

motivating examples. Note that unlike *anomalies* which are normally considered undesirable, Novelets may be beneficial, neutral, or negative, depending on the context.

2. The Contrast Profile: A Novel Time Series Primitive that Allows Real World

Classification

In this chapter, I introduce a definition of *contrast* that reflects this intuition for time series subsequences. As I will show, my proposed *Contrast Profile* is trivial to compute given just Matrix Profiles [1]. This is fortunate property as the *Contrast Profile* therefore inherits many of the desirable properties of the Matrix Profile, including its ability to be computed by many computational paradigms, such as anytime algorithms, incremental algorithms, and FPGA and GPU-accelerated computation [18]. Moreover, the Matrix Profile research community is diverse and active, and the *Contrast Profile* will be trivially able to take advantage of future developments.

The rest of this paper is organized as follows. In Section 2.1, we present the necessary definitions and notations. Section 2.2 sees a discussion of related work. In Section 2.3, we present several examples of data mining tasks that can exploit the Contrast Profile before experimentally demonstrating them in Section 2.4. Section 2.5 provides classification model comparisons and Section 2.6 offers conclusions.

2.1 Definitions And Notation

Our data type of interest is *time series*.

Definition 1: A *time series* $\mathbf{T} = t_1, t_2, \dots, t_n$ is a sequence of real-valued numbers.

Typically, one is not interested in global properties of a time series but rather shapes of small regions called *subsequences*.

Definition 2: A *subsequence* $\mathbf{T}_{i,m}$ is a contiguous subset of values from \mathbf{T} starting at index i with length m .

I can measure the distance between any two time series of equal length using a distance measure. In this dissertation, I use the ubiquitous z-normalized Euclidean distance [1]. One minor modification to the Euclidean distance is that I clip it at $\sqrt{(2m)}$ because values above this are anti-correlated in the Pearson Correlation space. This is done in order to make the greatest use of the normalized range when working with the Contrast Profile. If I need to measure the distance between a short time series and every subsequence from a long time series, I can produce a *distance profile*.

Definition 3: A *distance profile* $\mathbf{DP}_{i,m}^{(AB)}$ is the vector of distances between each subsequence in a query subsequence $\mathbf{T}_{i,m}^{(A)}$ and a reference time series $\mathbf{T}^{(B)}$.

The distance can be computed very efficiently using the MASS algorithm [2]. Fig 4 illustrates these definitions on a running example of a noisy electrocardiogram (ECG).

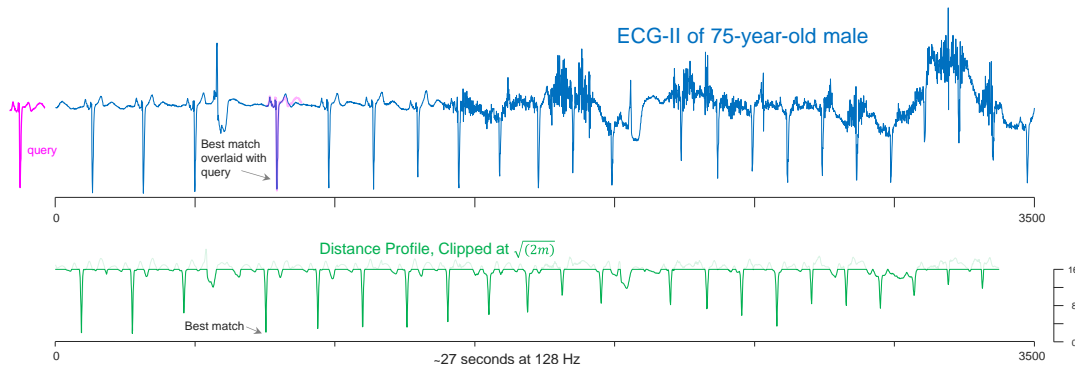


Fig 4 *top*) A 27-second snippet of an ECG time series. *bottom*) A single heartbeat from earlier in the same dataset was used as a query to produce a distance profile, which has low values when the “sliding” query is similar to a subsequence and is minimized at the best match about five seconds in.

Our proposed ideas leverage the *self-join Matrix Profile* [1].

Definition 4: A *self-join Matrix Profile* $\mathbf{MP}_m^{(AA)}$ of a time series $\mathbf{T}^{(A)}$ is a vector of Euclidean distances between every subsequence $\mathbf{T}_{i,m}^{(A)}$ and its nearest neighbor $\mathbf{T}_{j,m}^{(A)}$.

$$\text{Formally, } \mathbf{MP}_m^{(AA)} = \left[\min(\mathbf{DP}_{1,m}^{(AA)}), \min(\mathbf{DP}_{2,m}^{(AA)}), \dots, \min(\mathbf{DP}_{n-m+1,m}^{(AA)}) \right]$$

Fig 5 shows $\mathbf{MP}_{128}^{(AA)}$ for my running example. I can see that the top motifs are a pair of normal heartbeats. Using some out-of-band data (including advice of cardiologist Dr. Greg Mason), I annotated the location of two premature ventricular contractions. While these two beats are similar, they are not as well conserved as normal beats.

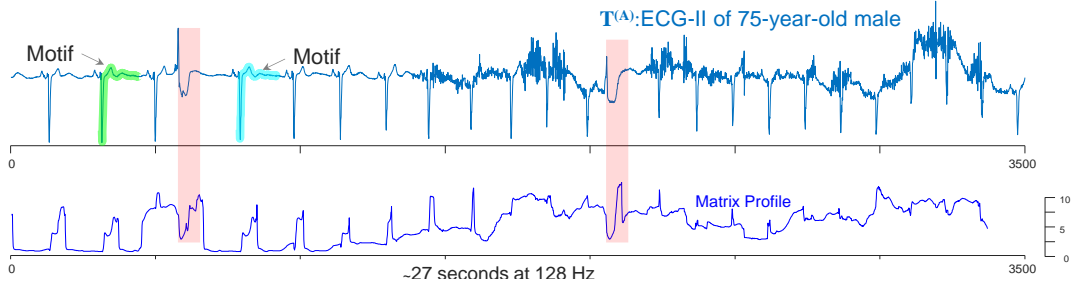


Fig 5 *top*) The ECG shown in Fig 4 with its \mathbf{MP}_{128} (*bottom*). The lowest values of \mathbf{MP}_{128} are the Top-1 motif pair, here two normal beats. Also, two PVCs shown highlighted with red bars for future reference.

In addition to subsequence comparisons *within* a time series, it can also be fruitful to make comparisons *between* two time series using the *AB-join Matrix Profile*.

Definition 5: An *AB-join Matrix Profile* $\mathbf{MP}_m^{(AB)}$ between reference time series $\mathbf{T}^{(A)}$ and a query time series $\mathbf{T}^{(B)}$ is a vector of Euclidean distances between each subsequence

$\mathbf{T}_{i,m}^{(A)}$ and its nearest neighbor $\mathbf{T}_{j,m}^{(B)}$. Formally,

$$\mathbf{MP}_{128}^{(AB)} = \left[\min(\mathbf{DP}_{1,m}^{(AB)}), \min(\mathbf{DP}_{2,m}^{(AB)}), \dots, \min(\mathbf{DP}_{n-m+1,m}^{(AB)}) \right]$$

Note that in general, $\mathbf{MP}_m^{(AB)} \neq \mathbf{MP}_m^{(BA)}$: even with equal lengths, they correspond to different reference time series.

Fig 6 shows $\mathbf{MP}_{128}^{(AB)}$ for my running example with a region of normal ECG from the same patient.

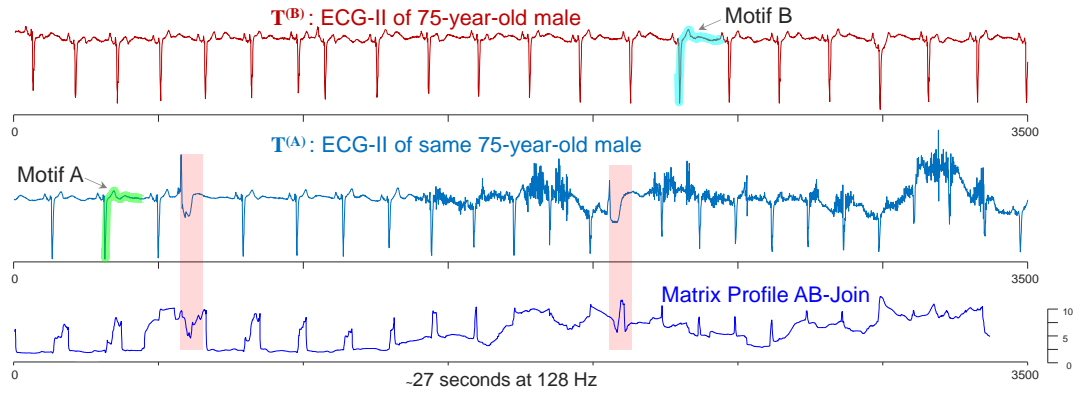


Fig 6 *top*) Time series $\mathbf{T}^{(B)}$ is a normal ECG time series from the same patient. *center*) Time series $\mathbf{T}^{(A)}$, which contains the behavior of interest, is the original ECG introduced in Fig 4. *bottom*) The top motif pair, where motif^(A) is the unrequited nearest neighbor of motif^(B). The red bars foreshadow discovery of two PVCs.

I now exploit an important observation. Note that $\mathbf{MP}_{128}^{(AA)}$ and $\mathbf{MP}_{128}^{(AB)}$ from the last two figures are very similar in most regions. This makes sense. A noisy $\mathbf{T}_{i,m}^{(A)}$ will tend to be just as far from any other $\mathbf{T}_{j,m}^{(A)}$ as it is from any $\mathbf{T}_{k,m}^{(B)}$ (An implication of theorem 1 of [19]). Moreover, a normal heartbeat in $\mathbf{T}^{(A)}$ will tend to have approximately the same low distance to another normal heartbeat, whether that beat happens to come from $\mathbf{T}^{(A)}$ or $\mathbf{T}^{(B)}$. The *only* places showing a significant difference are the locations corresponding to behaviors that are *unique* to $\mathbf{T}^{(A)}$: in this case, the two PVC beats.

I formalize these observations with my proposed representation, the *Contrast Profile*, specializing from the generic $\mathbf{T}^{(A)}$ and $\mathbf{T}^{(B)}$, to consider two time series $\mathbf{T}^{(+)}$ and $\mathbf{T}^{(-)}$ which have a mild assumption about their contents.

Definition 6: A *Contrast Profile* \mathbf{CP}_m is the difference between Matrix Profiles $\mathbf{MP}_m^{(+)}$ and $\mathbf{MP}_m^{(-)}$, where $\mathbf{MP}_m^{(+)}$ joins $\mathbf{T}^{(+)}$ with $\mathbf{T}^{(-)}$, and $\mathbf{MP}_m^{(-)}$ is the self-join of $\mathbf{T}^{(-)}$.

$$\mathbf{CP}_m = \mathbf{MP}_m^{(+)} - \mathbf{MP}_m^{(-)}$$

The Contrast Profile is defined for any two time series so long as m is shorter than the time series' lengths. However, I proposed to compute the Contrast Profile only when I believe that the two following assumptions are likely to be true:

- $\mathbf{T}^{(+)}$ contains at least two behaviors that are unique to the phenomena of interest.
- $\mathbf{T}^{(-)}$ contains zero behaviors of interest.

Under these assumptions, large values of \mathbf{CP}_m indicate behaviors that appear two or more times in $\mathbf{T}^{(+)}$ while absent from $\mathbf{T}^{(-)}$. Fig 7 gives a visual intuition of these definitions. Note that \mathbf{CP}_{128} peaks at the locations of the shape that is unique to $\mathbf{T}^{(+)}$ (i.e., the two PVC heartbeats).

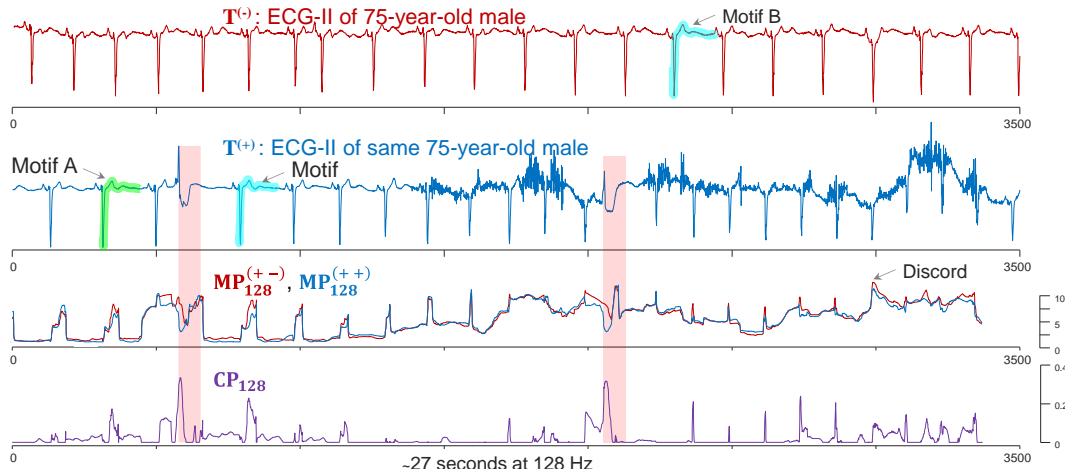


Fig 7 *top-to-bottom*) Query time series $\mathbf{T}^{(-)}$ contains normal heartbeats. Time series $\mathbf{T}^{(+)}$ contains at least two instances of a behavior of interest. The top discord of the AB-join Matrix Profile (the highest peak), results from a noisy region in $\mathbf{T}^{(+)}$, far from the ground truth labeled with red bars. The top two candidates peak within the ground truth.

The subsequence in $\mathbf{T}^{(+)}$ corresponding to the highest point in the Contrast Profile is called the *Plato*, a backronym of *Pattern likely able to organize*, which is suggestive of a *platonic* ideal for some behavior of interest.

While I use the Matrix Profile as the core function to compute the Contrast Profile, the value optimized is rather simple. The Plato is the subsequence in $\mathbf{T}^{(+)}$ with maximum difference between its nearest neighbor distance in $\mathbf{T}^{(-)}$ and nearest neighbor distance in $\mathbf{T}^{(+)}$. This could be discovered by a classic nested-loop, brute-force algorithm, requiring $O(|\mathbf{T}^{(+)}| (|\mathbf{T}^{(-)}| + |\mathbf{T}^{(+)}|) m)$. As m could be in the thousands, this is clearly intractable. As I will later show, by exploiting the Matrix Profile, I can completely remove the dependence on m to produce a highly scalable algorithm.

In order to allow for comparing Contrast Profiles of different subsequence lengths, I perform a normalization step. Z-normalized Euclidean distances have an upper bound of $2\sqrt{m}$, so it seems straightforward to divide all z-normalized Euclidean distances by this

upper bound. Though, I run into a subtle problem when computing the Contrast Profile's subtraction step. The goal of the Contrast Profile is to highlight subsequences which are better described by $\mathbf{MP}_m^{(++)}$ than $\mathbf{MP}_m^{(+-)}$, or where $\mathbf{MP}_m^{(+-)}$ distances are greater than $\mathbf{MP}_m^{(++)}$. Recall the following mapping from z-normalized Euclidean distance to Pearson correlation: 0 is correlated, $\sqrt{2m}$ is uncorrelated, and $2\sqrt{m}$ is anti-correlated. Now consider the following case for a possible subsequence $\mathbf{T}_{i,50}^{(+)}$ with subsequence length 50 (0:correlated, 10:correlated, 14.1:anti-correlated). $\mathbf{MP}_{i,50}^{(+-)}$ with distance 11 and $\mathbf{MP}_{i,50}^{(++)}$ with distance 10 results in $\mathbf{CP}_{i,50} = 11 - 10 = 1$. A positive non-zero contrast value suggests that the nearest neighbor subsequence of $\mathbf{T}_{i,m}^{(+)}$ within $\mathbf{MP}_{50}^{(++)}$ contains greater information relative to $\mathbf{T}_{i,m}^{(+)}$ than the nearest neighbor subsequence within $\mathbf{MP}_{50}^{(+-)}$. Though, a pair of anti-correlated subsequences contain greater information relative to each other than a pair of uncorrelated subsequences, which means $\mathbf{T}_{i,m}^{(+)}$ is better described by its nearest neighbor in $\mathbf{MP}_{i,50}^{(+-)}$ than in $\mathbf{MP}_{i,50}^{(++)}$. This is a contradiction. Rather than reason about how to compare correlated and anti-correlated subsequence pairs, I propose to ignore the anti-correlated case by clipping either the distance profile or Matrix Profile distances above $\sqrt{2m}$, as suggested in Definition 2.

Now that I have settled on an upper bound, I divide by $\sqrt{2m}$.

$$\widetilde{\mathbf{CP}}_m = (\mathbf{MP}_m^{(+-)} - \mathbf{MP}_m^{(++)}) / \sqrt{2m}$$

This caps the maximum value at 1, but it is still possible to encounter values below 0. Negative values occur when $\mathbf{MP}_m^{(++)}$ is less descriptive than $\mathbf{MP}_m^{(+-)}$, which I am uninterested in, so I can simply clip negative values to 0:

$$\mathbf{CP}_m = \max(\mathbf{0}, \widetilde{\mathbf{CP}}_m)$$

The Contrast Profile is now conveniently length normalized and bounded by [0,1].

To summarize, I have shown that at least for my running example, the Contrast Profile can be used to extract discriminating subsequences. This clearly has implications for several downstream algorithms, including classification and novelty/anomaly detection. However, before discussing these, in the next two sections I will consider the Contrast Profile's robustness to noise and the plausibility of the assumptions that warrant its use.

2.1.1 The Contrast Profile's Robustness

The robustness of the Contrast Profile definition is hinted at in Fig 8, as it correctly recovers the PVC patterns in spite of sporadic noise caused by motion artifacts. Moreover, I empirically test this robustness in my experimental section. However, it is worth explaining *why* it is so robust. Consider Fig 8.*top* which shows \mathbf{CP}_{128} computed from two relatively clean ECGs, to hint at the presence of novel patterns (PVCs) in $\mathbf{T}^{(+)}$. Moreover, as I show in [20] (for brevity), while this particular example shows $m = 128$, the PVCs are discovered with any setting in the range $m = [50:500]$.

Suppose I add some random noise to the time series and follow up by concatenating some random data to the end; how would it affect the Contrast Profile? As Fig 8.*bottom* shows, the position of the peaks in the relevant sections of \mathbf{CP}_{128} is essentially unchanged, even when the noise is severe.

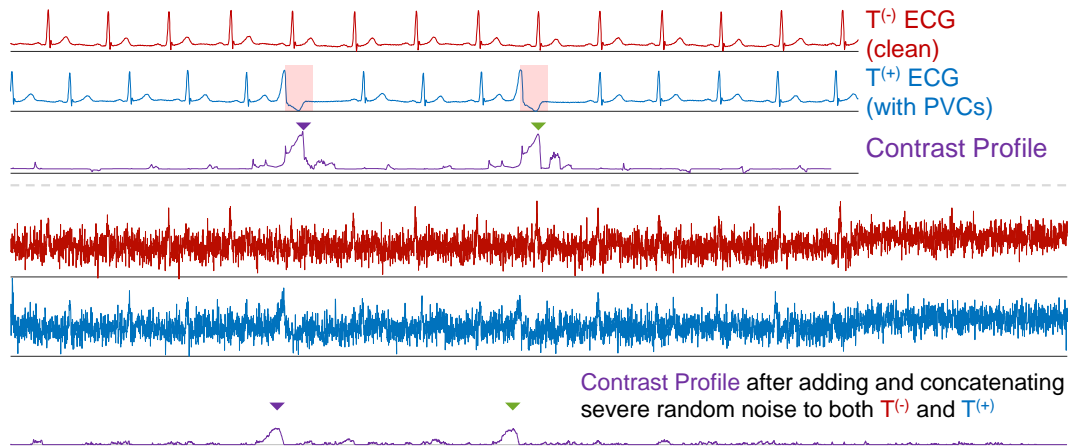


Fig 8 *top*) Two ECGs and their Contrast Profile. *bottom*) After adding and concatenating a severe magnitude of random noise to the two time series, the peak locations between the original and new **CP** are virtually identical, and the concatenated section has low values, meaning it will not cause spurious matches.

Let me examine this finding for each time series.

- If $\mathbf{T}^{(-)}$ has sections of noise or any irrelevant data, it makes no difference to the **CP**, because none of the irrelevant data will act as a nearest neighbor to any subsequence from $\mathbf{T}^{(+)}$.
- If $\mathbf{T}^{(+)}$ has sections of noise, it makes no difference to the **CP** because the noisy data will be approximately as far from its nearest neighbor in $\mathbf{T}^{(-)}$ ¹ as it is far from its nearest neighbor in $\mathbf{T}^{(+)}$. Thus, the relevant indices of the $\mathbf{MP}_m^{(+)} - \mathbf{MP}_m^{(-)}$ calculation will subtract two nearly equal numbers, resulting in a score less than or equal to zero.

Another consideration in robustness is the quality of the data. It is common for data to contain missing values. The Contrast Profile largely side steps responsibility for this issue by relying on existing implementations of the Matrix Profile, though NaN values are tolerated by the batch implementation of the Matrix Profile used in this dissertation. Another

¹Beyer et al. 1999) As an implication of theorem 1 of [19]) a noisy subsequence is an approximately equal and *large*, distance to all other subsequences.

considering is sampling rate, which I assume is uniform and unsegmented. Though, it is possible to support segmented captures by concatenating capture sessions with a NaN spacer if the base Matrix Profile implementation supports NaN. The Contrast Profile is not crippled by outliers in the form of extreme high or low values. They are treated as any other repeating or non-repeating shape. These considerations lead to the conclusion that the Contrast Profile can generally operate on raw data. In this dissertation, only the chicken dataset in Section 2.4.2 was preprocessed, but was done so to obfuscate biological significance. Though in general, additional domain knowledge is helpful and can be added through preprocessing. A rule of thumb is if meaningful motifs can be found with the Matrix Profile, then the Contrast Profile can identify those that are contrastive. While the Contrast Profile is remarkably robust to the quality of data, it is unlikely to be effective on time series domains for which feature based methods are more effective than shape, such as EEG signals.

Thus, so long as my mild assumptions are true, the Contrast Profile is likely to discover the discriminating behavior.

2.1.2 The Contrast Profile Assumptions

Recall that my assumptions for the Contrast Profile are that $\mathbf{T}^{(+)}$ contains at least two examples of the desired behavior, but that $\mathbf{T}^{(-)}$ contains zero examples. These are very mild assumptions, nevertheless it is worth considering when such assumptions are warranted.

Our chicken example, shown in Fig 1 and empirically revisited in Section 2.4.2, is one such documented example where this assumption is justified [7]. I argue that such examples are very common in many domains. For example, in batch processing with *delayed coking*

[16], it is possible to produce a bad batch, but not be sure exactly *when* the process began to fail. Thus, it is common to hear a petrochemical engineer report something like “*Everything was running perfectly on Monday, but then I had a couple of bad batches on Tuesday.*” Here I can compute the Contrast Profile with $\mathbf{T}^{(-)} \leftarrow \text{telemetry}(\text{Mon})$ and $\mathbf{T}^{(+)} \leftarrow \text{telemetry}(\text{Tues})$.

Moreover, in some cases I can actually *intervene* in a domain to ensure data that conforms to my assumptions. For example:

- While it was not necessary for my chicken work, I *could* have created a cage that had all the usual food and water supplies, but had a solid floor to prevent the bird from dustbathing. This would have made the $\mathbf{T}^{(-)}$ trivially true.
- For my insect example (cf. Section 2.4.1), suppose I hope to find the signature of “*cell rupture*” feeding as opposed to the more common “*salivary sheath*” feeding [21]. Since it is known what parts of a plant allow such feeding strategies, it would be easy for an entomologist to add/remove the appropriate plant parts into the insectary to build $\mathbf{T}^{(+)}$ and $\mathbf{T}^{(-)}$ with the Contrast Profile assumptions satisfied.

Finally, my assumptions are satisfied trivially in most examples relating to *human* behavior. For example, if I want to understand what effect (if any) a stretching warm-up routine has on a hurdler’s jumps, I can simply ask her to record her workouts with and without the warm-up routine.

2.1.3 Further Relaxations of the Contrast Profile Assumptions

As I noted in the previous section, the Contrast Profile assumptions are mild, and are trivial to satisfy in the cases where I can intervene before the data is collected. Nevertheless, even these weak assumptions may be difficult to satisfy in some cases. If the assumptions are not satisfied, and there is even a single example of the desired behavior in $\mathbf{T}^{(-)}$, then the Contrast Profile will fail to find a meaningful Plato.

Recall that I envisioned a hypothetical petrochemical engineer report something like “*Everything was running perfectly on Monday, but then we had a couple of bad batches on Tuesday.*” Here I wish to consider the case where their claim is even more relaxed and vague, something like “*We had fewer bad batches on Monday than on Tuesday.*”

To allow the application of the Contrast Profile to such situations, I must generalize it from the assumption of “*zero occurrences vs. at least two*” to “*more frequent in $\mathbf{T}^{(+)}$ than $\mathbf{T}^{(-)}$ ”.*

In Fig 9, I update the visualization of points in a high dimensional space originally shown in Fig 2. I again explicitly consider three data points.

- Point **A** is far from its 1st nearest neighbor in the non-target class *and* close to its 1st nearest neighbor in the target class. This is the originally defined 1-NN Contrast Profile.
- Point **B** in contrast is very close to its Kth nearest neighbor in the target class, but it is also close to its Kth nearest neighbors in non-target class.
- Point **C** is very far from its Kth nearest neighbor in the non-target class *and* very close to its Kth nearest neighbor in the target class, where K happens to be 3.

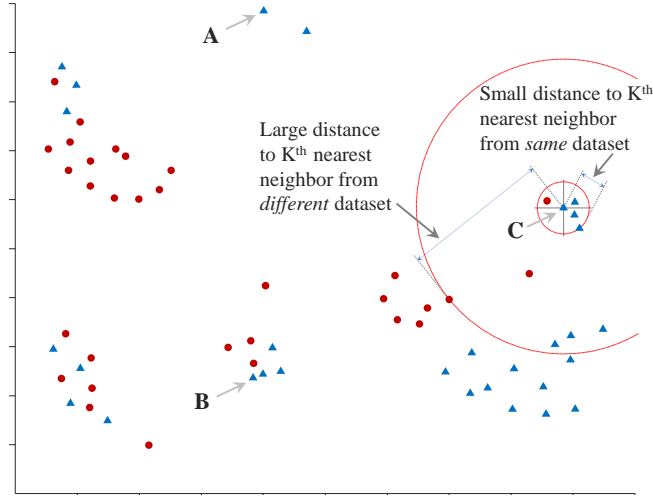


Fig 9 A visual intuition of the “relative frequency contrast” property. Of the three annotated points from the target class, **C** has the largest contrast between the K^{th} neighbor in the target class and the K^{th} neighbor in the non-target class, which happens to be $K=3$.

For clarity I call this variant of the Contrast Profile, the *Relative Frequency Contrast Profile* (RFCP).

Calculating the RFCP is trivial once the Relative Frequency Matrix Profile (RFMP) is defined for $\mathbf{RFMP}_m^{(+-)}$ and $\mathbf{RFMP}_m^{(++)}$, as shown in Table 1. The time complexity of \mathbf{RFCP}_m is $O(|\mathbf{T}^{(+)}| |\mathbf{T}| \log |\mathbf{T}|)$, which is slower by a factor of $\log |\mathbf{T}|$, where \mathbf{T} is the longer of $\mathbf{T}^{(+)}$ and $\mathbf{T}^{(-)}$. This indicates that the time complexity is independent of the number of nearest neighbors returned.

Table 1 The RelativeFrequencyContrastProfile Algorithm.

Algorithm: RelativeFrequencyContrastProfile ($\mathbf{T}^{(-)}, \mathbf{T}^{(+)}, m, \text{maxFreq}$)	
Input: negative time series $\mathbf{T}^{(-)}$, positive time series $\mathbf{T}^{(+)}$, subsequence length m , and maximum frequency of behaviors.	
Output: The Relative Frequency Contrast Profile \mathbf{RFCP}_m .	
1	$\mathbf{RFMP}_m^{(+-)} \leftarrow \text{RelativeFrequencyMatrixProfile}(\mathbf{T}^{(-)}, \mathbf{T}^{(+)}, m, \text{maxFreq}) // O(\mathbf{T}^{(+)} \mathbf{T}^{(-)} \log \mathbf{T}^{(-)})$
2	$\mathbf{RFMP}_m^{(++)} \leftarrow \text{RelativeFrequencyMatrixProfile}(\mathbf{T}^{(+)}, \mathbf{T}^{(+)}, m, \text{maxFreq}) // O(\mathbf{T}^{(+)} \mathbf{T}^{(+)} \log \mathbf{T}^{(+)})$
3	$\mathbf{RFCP}_m \leftarrow (\mathbf{RFMP}_m^{(+-)} - \mathbf{RFMP}_m^{(++)}) / \text{sqrt}(2 * m) // \text{with normalization}$
4	return \mathbf{RFCP}_m

Due to some subtleties which will be addressed shortly, I based the implementation of the RFMP on MASS in Table 2 [2]. On line 1 I set the *selfOffset* flag, which helps to generalize for both self-join and AB-join variants. Following this in line 2, $\mathbf{RFMP}_m^{(AB)}$ is initialized with zeros for each relevant index of $\mathbf{T}^{(B)}$ and *maxFreq* nearest neighbors. A loop over every subsequence $\mathbf{T}_{i,m}^{(A)}$ is started in line 3 which facilitates the KNN search. After computing the distance profile in line 4, nearest neighbors are selected in line 5 through a process described in Table 3. The *selfOffset* flag is added to *maxFreq* so that the first nearest neighbor (1-NN) can be ignored in line 6 if performing a self-join. The *maxFreq* nearest neighbors are then stored for the corresponding subsequence index.

Table 2 The RelativeFrequencyMatrixProfile Algorithm.

Algorithm: RelativeFrequencyMatrixProfile ($\mathbf{T}^{(B)}$, $\mathbf{T}^{(A)}$, m , <i>maxFreq</i>)	
Input: query time series $\mathbf{T}^{(B)}$, reference time series $\mathbf{T}^{(A)}$, subsequence length m , and maximum frequency of behaviors <i>maxFreq</i> .	
Output: The Relative Frequency Matrix Profile $\mathbf{RFMP}_{i,m}^{(AB)}$.	
1	$selfOffset \leftarrow 1$ if equals($\mathbf{T}^{(B)}$, $\mathbf{T}^{(A)}$), else 0 //flag indicating self-join Matrix Profile
2	$\mathbf{RFMP}_m^{(AB)} \leftarrow \text{zeros}(\mathbf{T}^{(A)} - m + 1, maxFreq)$
3	For $\mathbf{T}_{i,m}^{(A)}$ in $\mathbf{T}^{(A)}$:
4	$\mathbf{DP}_{i,m}^{(AB)} \leftarrow \text{MASS}(\mathbf{T}^{(B)}, \mathbf{T}_{i,m}^{(A)})$ // $O(\mathbf{T}^{(B)} \log \mathbf{T}^{(B)})$
5	$\mathbf{KNN}^{(AB)} \leftarrow \text{NearestNeighborSelection}(\mathbf{DP}_{i,m}^{(AB)}, m, maxFreq + selfOffset)$
6	$\mathbf{RFMP}_{i,m}^{(AB)} \leftarrow \mathbf{KNN}_{1+selfOffset,maxFreq}^{(AB)}$ //ignore first NN if performing self-join
7	return $\mathbf{RFMP}_{i,m}^{(AB)}$

One subtlety when implementing the RFMP is the careful setting of the exclusion zone required around nearest neighbors. For ease of explanation, imagine the entire distance matrix is stored when computing each of the matrix profiles where columns represent

distance profile $\mathbf{DP}_{i,m}^{(AB)}$ from the reference time series $\mathbf{T}^{(A)}$ to the query time series $\mathbf{T}^{(B)}$. If each column is sorted independently, the first row is the standard Matrix Profile $\mathbf{MP}_m^{(AB)}$, which is a vector of the distances from $\mathbf{T}^{(A)}$ to the 1-NN in $\mathbf{T}^{(B)}$. Since these are the shortest distances in each column due to sorting, it may seem straightforward to call the problem solved by simply selecting the next row for the 2-NN Matrix Profile and so on until the *maxFreq* Matrix Profile. The issue with this is *trivial matches* [1]. The next shortest distance in each column most likely corresponds to an overlapping subsequence to the 1-NN, which may be redundant with the original subsequence. In order to address this, an exclusion zone is set around each chosen nearest neighbor index until *maxFreq* subsequence indices are selected.

The NearestNeighborSelection algorithm is detailed in Table 3. The output vector for *maxFreq*-NN $\mathbf{KNN}^{(AB)}$ is initialized empty in line 1 along with line 2 clearing the vector **EXCLUDED_INDICES**, which annotates $\mathbf{DP}_{i,m}^{(AB)}$. The sorted index positions of $\mathbf{DP}_{i,m}^{(AB)}$ are stored in **SORTED_INDICES** $_{\mathbf{DP}_{i,m}^{(AB)}}$ which are then iterated over starting in line 4. If the annotation vector **EXCLUDED_INDICES** for sorted index j is clear in line 5, then the subsequence is chosen as a nearest neighbor. The nearest neighbor distance $\mathbf{DP}_{(i,m),j}^{(AB)}$ is appended to $\mathbf{KNN}^{(AB)}$ in line 6. In line 7, an exclusion zone of length m is centered around j . The maximum number of nearest neighbors returned through this method is $\lfloor |\mathbf{T}^{(B)}|/m \rfloor$, though the number of relevant results is dependent on the domain and specific time series.

The time complexity of the NearestNeighborSelection Algorithm is $O(|\mathbf{T}^{(B)}| \log |\mathbf{T}^{(B)}|)$, and the space complexity is $O(|\mathbf{T}^{(B)}|)$.

Table 3 The NearestNeighborSelection Algorithm.

Algorithm: NearestNeighborSelection ($\mathbf{DP}_{i,m}^{(AB)}$, m , $maxFreq$)	
Input: Distance profile $\mathbf{DP}_{i,m}^{(AB)}$, subsequence length m , and maximum frequency of behaviors $maxFreq$.	
Output: The top $maxFreq$ nearest neighbors $\mathbf{KNN}^{(AB)}$	
1	$\mathbf{KNN}^{(AB)} \leftarrow []$
2	$\mathbf{EXCLUDED_INDICES} \leftarrow \text{zeros}(\mathbf{DP}_{i,m}^{(AB)} , 1)$ //annotation vector
3	$\mathbf{SORTED_INDICES}_{\mathbf{DP}_{i,m}^{(AB)}} \leftarrow \text{argsort}(\mathbf{DP}_{i,m}^{(AB)})$
4	For j in $\mathbf{SORTED_INDICES}_{\mathbf{DP}_{i,m}^{(AB)}} : //j$ is an index in $\mathbf{T}^{(B)}$
5	If $\mathbf{EXCLUDED_INDICES}_j = 0$:
6	$\mathbf{KNN}^{(AB)} \leftarrow [\mathbf{KNN}^{(AB)}, \mathbf{DP}_{(i,m),j}^{(AB)}]$ // j is an index in $\mathbf{T}^{(B)}$, $\mathbf{DP}_{(i,m),j}^{(AB)}$ is a distance
7	$\mathbf{EXCLUDED_INDICES}_{j-m/2,m} \leftarrow \text{ones}(m, 1)$ //exclusion length m centered on j
8	return $\mathbf{KNN}^{(AB)}$

2.1.4 A Case Study of the Relative Frequency Contrast Profile

Here I investigate a seal behavior dataset [22], shown in Fig 10. The question I am interested in asking is “Are there behaviors that are more frequent in females than males (or vice versa)?”

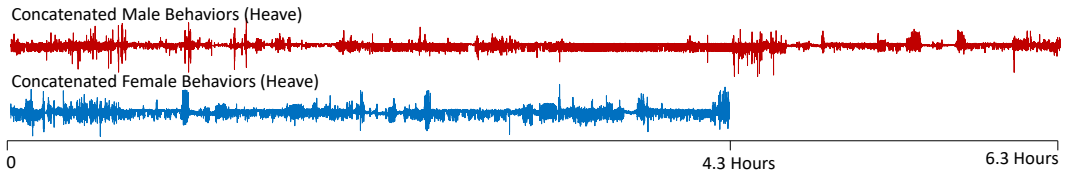


Fig 10 Time series consisting exclusively of a single sex were constructed by concatenating each sample’s heave axis, separated by NaN.

Our intuition is that animals have evolved some highly conserved behaviors that have become hard-wired in their DNA. In particular, I have a suspicion that the generally heavier males may perform some behaviors differently to the lighter females.

The time series for each sex was constructed by concatenating all time series in the Z-axis (Heave) with a single NaN between samples in order to exclude artifacts due to joining. Heave is vertical translational motion. For example, in humans, nodding the head in agreement produces acceleration in the heave axis. The female time series is of length 385,396 and the male time series is of length 562,585. To search for female specific behaviors, I set $\mathbf{T}^{(+)}$ to the female time series and $\mathbf{T}^{(-)}$ to the male time series. I swap the ordering when searching for male specific behaviors. I chose a subsequence length of 200 (4 sec.) based on a brief visual inspection of the data.

The 1-NN Contrast Profile is enough to reveal the highest contrast subsequences for both female and male seals, as shown in Fig 11. However, the Plato and the Plato's nearest neighbor come from a single individual seal. Thus, they are more likely to be the idiosyncrasies of an individual than a general sex-specific behavior.

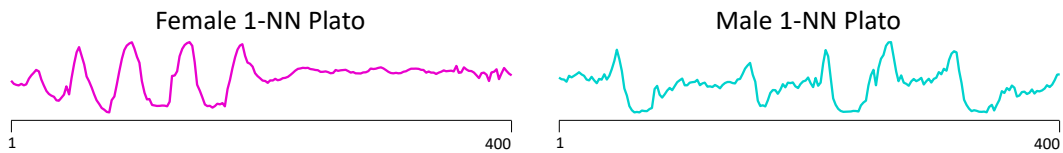


Fig 11 Female 1-NN Female Plato and Male 1-NN Plato.

By exploring the RFCP visually in heatmap form in Fig 12, I can investigate whether there are such highly repeated behaviors across many members of a sex.

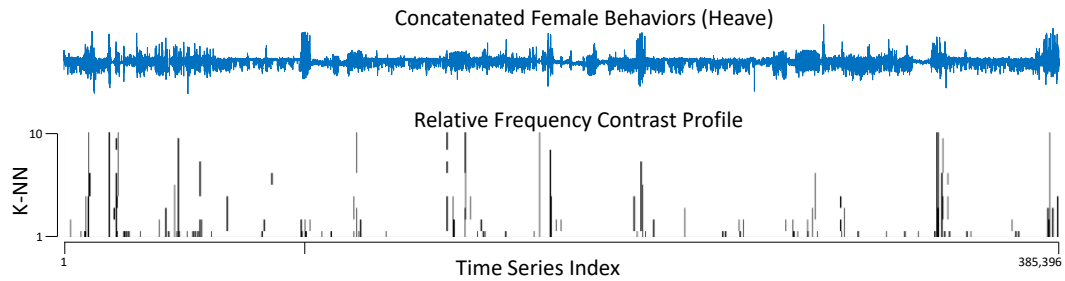


Fig 12 Concatenated Female behaviors shown alongside heatmap of RFCP.

I can zoom into the heatmap in Fig 13 to look at regions marked by black pillars extending from $K = 1$. Around index 99,121 there is a series of these pillars. I pick the central pillar and check whether it is a behavior common to many females.

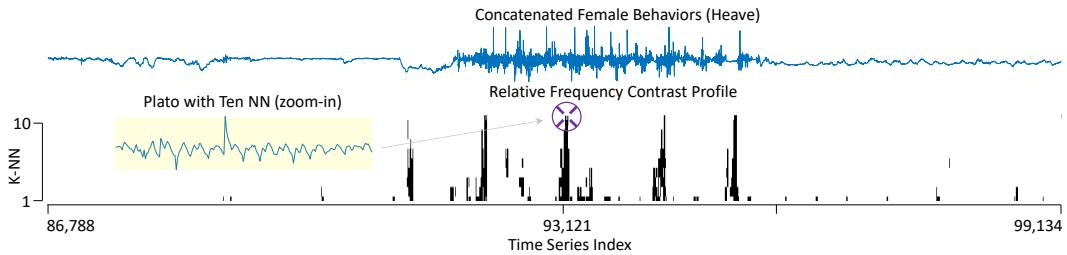


Fig 13 Zoomed in region of concatenated Female behaviors shown alongside heatmap of RFCP. The heatmap highlights a behavior occurring at least ten times, exclusively in the time series collected from females.

The ten nearest neighbors of the selected behavior are present in three of four female seals: Maxine, Miri, and Abbey. While I do not have video files to consult, I can check the behavior labels which the original data archivists manually annotated. All of the ten-NN are captured while the seals were on land and nine of ten correspond to “Travelling”, the last being “Other”. This fact, and the periodicity observed in the Plato suggests that behavior is *galumphing* [23], the fast “gait” of seals on land. The reader may wonder why this behavior stands out as female specific since galumphing is observed in both sexes. Seals are among the most sexually dimorphic of mammals. In this study the average mass of the females was 57 kg, whereas the males had an average mass of 115 kg. I speculate

that this large difference in mass affects the gait, significantly increasing the period for the male gait [24]. So while galumphing is observed in both sexes, there may be a sex specific period. For behaviors in the water, the neutral buoyancy of the seals largely cancels out the effects of mass on behaviors.

Finally, the reader may wonder if the classic Contrast Profile would not have worked here. The original seal behavior dataset [22] actually contains a mixture of species, including five Antipodean fur seals (*Arctocephalus forsteri*), and six Australian sea lions (*Neophoca cinerea*)²; these are very similar animals. However, there is a single young male subantarctic fur seal (*Arctocephalus tropicalis*), which at 30 kg is as small as the smaller females of the other species. Here my relaxed assumptions come in to play, as the single subantarctic fur seal can produce subsequences that violate the classic Contrast Profile Assumptions.

2.1.5 General Contrast Profile Observations

Note that while the two time series that are input into the Contrast Profile are denoted $\mathbf{T}^{(+)}$ and $\mathbf{T}^{(-)}$, there is nothing pejorative about the “negative” time series. It is simply a snippet of data which I know does not have some behavior. That behavior *could* be undesirable, say a seizure, or it could be desirable, say a critical depressurization phase in an industrial process.

² The original paper lists “two Australian fur seals (*Arctocephalus pusillus doriferus*) and three New Zealand fur seals (*Arctocephalus forsteri*)”, however it is now commonly accepted that these are the same species and are united under the more neutral name of Antipodean fur seals.

The Contrast Profile is bound between zero and one. A value of one corresponding to $\mathbf{T}_{i,m}^{(+)}$ means that $\mathbf{T}_{i,m}^{(+)}$ is a perfect motif in $\mathbf{MP}_m^{(++)}$ while also a maximum discord in $\mathbf{MP}_m^{(+-)}$ [1].

A value of zero means that $\mathbf{T}_{i,m}^{(+)}$ is conserved at least as much in $\mathbf{MP}_m^{(+-)}$ as $\mathbf{MP}_m^{(++)}$. This property is critically different from that of TS-Diff [25], which is optimized solely by maximizing $\mathbf{MP}_m^{(+-)}$, a definition that simply tends to point to the noisiest subsequence.

A useful property of the Contrast Profile is that it is length invariant and sampling-rate invariant. For example, I can meaningfully compare scores for length 50 and for length 60, and state which subsequence is better conserved. This provides me with the opportunity to remove the Contrast Profile's only parameter, the subsequence length. I propose the Pan-Contrast Profile (in the spirit of [26]). I can simply compute all Contrast Profiles in some range, and choose the Plato from the one that produces the highest value. To see why I can expect this to work, consider the two extreme cases.

- If m is too small, then I am only comparing tiny fragments of the time series. These are very unlikely to be discriminating.
- If m is too large, then I am comparing the most discriminating subsequence along with extra non-discriminating shapes padded to its prefix or suffix. These non-discriminating sections can only dull the contrast property.

In Fig 14, I show the Pan-Contrast Profile for the ECG shown in Fig 8; the example bears out my intuition above. The optimal Plato has a length of 313, which is about the length of the PVC, excluding the QRS peak, which it shares with healthy heartbeats.

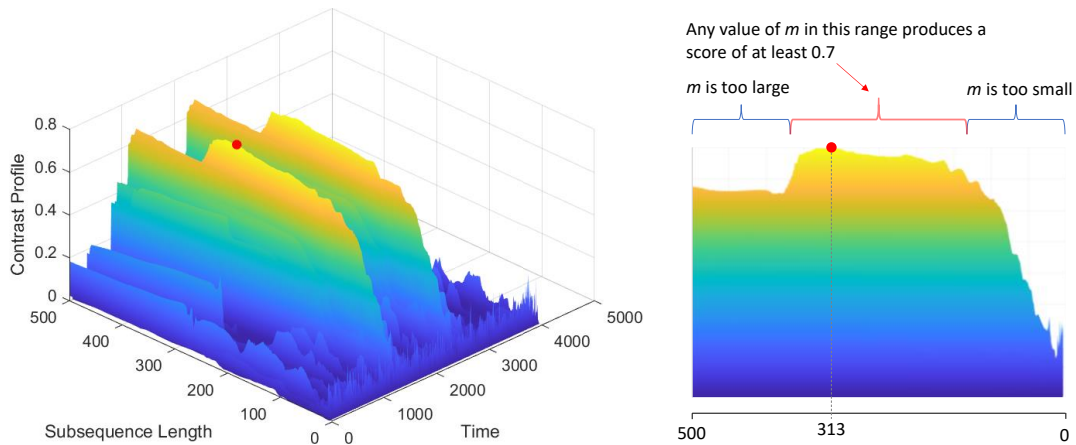


Fig 14 *left*) The Pan-Contrast Profile for the example shown in Fig 8. A red dot indicates the largest value. *right*) A side view shows that the Contrast Profile is very robust to its only input parameter. Any subsequence length from 131 to 424 would have produced a score of at least 0.7.

One additional takeaway from this experiment is the relative insensitivity of the Contrast Profile definition to its only parameter. Over a huge range of values (131 to 424) it produces nearly identical values in nearly identical locations.

A computation of a single Contrast Profile requires $O(|\mathbf{T}^{(+)}|^2 + |\mathbf{T}^{(+)}||\mathbf{T}^{(-)}|)$ time. To concretely ground this, the example shown in Fig 8 takes 0.182 seconds, and the full Pan-Contrast Profile shown in Fig 14 takes 82 seconds. Note that because the Contrast Profile is based on the Matrix Profile, it inherits many of the Matrix Profile’s desirable properties such as time complexity that is completely independent of the subsequence’s dimensionality, and the possibility of anytime, online, and GPU-accelerated computation [18].

Thus far, I have only defined the Top-1 Plato. However, it is possible that I may be interested in the Top-K Platos, as I may suspect that the behavior of interest is polymorphic. For example, unlike the simple PVC arrhythmia shown in Fig 22.*inset*, some arrhythmias such as bidirectional ventricular tachycardia can present themselves with a handful of

different shapes even from a single individual. If I am given $\mathbf{T}^{(+)}$ that has at least two examples of each manifestation, I would like to extract them all.

Recall that for time series discords, the Top-K discords correspond to the Top-K peaks in the Matrix Profile. However, that is not the case for the Contrast Profile. To discover the K^{th} Plato I must ensure that the influence of the $K^{\text{th}}-1$ Plato is first removed from the Contrast Profile. That is trivial to achieve, I simply concatenate the $K^{\text{th}}-1$ Plato to $\mathbf{T}^{(-)}$ and then recompute the Contrast Profile from scratch³. All subsequences in $\mathbf{T}^{(+)}$ that were similar to the K^{th} Plato will then be close to a subsequence in $\mathbf{T}^{(-)}$, and thus their original peaks will vanish.

A caveat to be aware of when evaluating the Contrast Profile is that since a profile is always computed, there will always be a maximum value. If $\mathbf{T}^{(+)}$ and $\mathbf{T}^{(-)}$ are both noise or are identical, a Plato will still be suggested. This problem is shared with the Matrix Profile. If ground truth labels are available, a straightforward validation step is to setup a classification test like those in Section 2.4. If ground truth is not available for a classification test, I recommend that for a given Plato, overlay it with its nearest neighbor within $\mathbf{T}^{(+)}$ and $\mathbf{T}^{(-)}$. If there is greater similarity within $\mathbf{T}^{(+)}$ than $\mathbf{T}^{(-)}$, then the Plato may be meaningful. Of course, defining the Plato on a single nearest neighbor may be fragile, so examining two or more nearest neighbors using the Relative Frequency Contrast Profile (Section 2.1.3) will deliver more robust Platons.

³ This is what *logically* must be done, however by caching distance calculations and only recomputing values that could have changed, the time and space overhead for the $K^{\text{th}}-1$ Plato is inconsequential.

2.1.6 Online Contrast Profile

The reader will appreciate that it may be useful to compute the Contrast Profile in an online fashion. While “online” could have several interpretations, I believe the most useful variant will be a fixed $\mathbf{T}^{(-)}$ with an incrementally updated $\mathbf{T}^{(+)}$ in the face of real-time data arrival. As I will show in Section 2.3.3, this interpretation maps onto a type of anomaly detection.

Assume that I start with a computed \mathbf{CP}_m of length n for $\mathbf{T}^{(+)}$, and some length for $\mathbf{T}^{(-)}$, and I wish to ingest an additional datapoint, the $n + 1$ datapoint. This will result in the creation of a new subsequence, NEW , which ends with the $n + 1$ datapoint.

What effect will subsequence NEW have on the current \mathbf{CP}_m , beyond lengthening it by one?

- If NEW is sufficiently dissimilar to any other subsequence in $\mathbf{T}^{(+)}$, then the previous n values of \mathbf{CP}_m will be unchanged regardless of NEW 's distance to its nearest neighbor in $\mathbf{T}^{(-)}$.
- If NEW is similar to one or more subsequences in $\mathbf{T}^{(+)}$, but also sufficiently close to its nearest neighbor in $\mathbf{T}^{(-)}$, then the previous n values of \mathbf{CP}_m will again be unchanged.
- If NEW is similar to one or more subsequences in $\mathbf{T}^{(+)}$, and it is far from any subsequence in $\mathbf{T}^{(-)}$, then I will have to update \mathbf{CP}_m corresponding to those subsequences.

From this, I can see that the previously computed \mathbf{CP}_m values can only increase or stay the same. They can never decrease. Then, adding the $n + 1$ value to \mathbf{CP}_m requires computing every index in $\mathbf{DP}_{NEW,m}^{(+-)}$ and $\mathbf{DP}_{NEW,m}^{(++)}$. After outlining the algorithm that maintains the

Contrast Profile *Incremental* (ContrastProfileI) in Table 4, I will explain how this process can be accomplished surprisingly efficiently by exploiting the MASS algorithm [2].

I denote the updated variables with ^{NEW} in the superscript. In line 1, each newly arriving time point $t^{(+)}$ is appended to the expanding time series $\mathbf{T}^{(+)}$. This completes the next subsequence *NEW* in $\mathbf{T}_m^{(+),NEW}$ in lines 2 and 3. Lines 4 and 5 correspond to updating the contrasting Matrix Profile by first calculating the distance profile $\mathbf{DP}_{last,m}^{(+)}$ between $\mathbf{T}^{(-)}$ and *NEW*, then appending the minimum of $\mathbf{DP}_{last,m}^{(+)}$ to $\mathbf{MP}_m^{(+)}$ and storing in $\mathbf{MP}_m^{(+),NEW}$.

Table 4 The ContrastProfileI Algorithm.

Algorithm: ContrastProfileI ($\mathbf{T}^{(-)}$, $\mathbf{T}^{(+)}$, $t^{(+)}$, $\mathbf{MP}_m^{(+)}$, $\mathbf{MP}_m^{(++)}$, m)	
Input: negative time series $\mathbf{T}^{(-)}$, positive time series $\mathbf{T}^{(+)}$, a new positive time point $t^{(+)}$ following $\mathbf{T}^{(+)}$, Matrix Profile $\mathbf{MP}_m^{(+)}$, Matrix Profile $\mathbf{MP}_m^{(++)}$, and subsequence length m .	
Output: The Contrast Profile \mathbf{CP}_m , the incrementally updated Matrix Profiles $\mathbf{MP}_m^{(+),NEW}$ and $\mathbf{MP}_m^{(++)},NEW$, and the current time series $\mathbf{T}^{(+),NEW}$.	
1	$\mathbf{T}^{(+),NEW} = [\mathbf{T}^{(+)}, t^{(+)}$]
2	$last \leftarrow n - m + 1$ // index of last subsequence in $\mathbf{T}^{(+),NEW}$
3	$NEW \leftarrow \mathbf{T}_{last,m}^{(+),NEW}$ // last subsequence in $\mathbf{T}^{(+),NEW}$ of length m
4	$\mathbf{DP}_{last,m}^{(+)} \leftarrow \text{MASS}(\mathbf{T}^{(-)}, NEW)$ // Begin AB-join update
5	$\mathbf{MP}_m^{(+),NEW} \leftarrow [\mathbf{MP}_m^{(+)}, \text{Min}(\mathbf{DP}_{last,m}^{(+)})]$
6	$\mathbf{DP}_{last,m}^{(++)} \leftarrow \text{MASS}(\mathbf{T}^{(+)}, NEW)$ // Begin self-join update
7	$\mathbf{MP}_m^{(++)'} \leftarrow \text{ElemWiseMin}(\mathbf{MP}_m^{(++)}, \mathbf{DP}_{last,m}^{(++)})$ // Update prev vals
8	$\mathbf{MP}_m^{(++)},NEW \leftarrow [\mathbf{MP}_m^{(++)'}, \text{Min}(\mathbf{DP}_{last,m}^{(++)})]$
9	$\mathbf{CP}_m \leftarrow (\mathbf{MP}_m^{(+),NEW} - \mathbf{MP}_m^{(++)},NEW) / \text{sqrt}(2m)$
10	return \mathbf{CP}_m

Because this is a Matrix Profile where the query time series is unchanging, the previously computed values are also unchanging. An extra line of work is done in lines 6 – 8 to update the self-join Matrix Profile because the query time series $\mathbf{T}^{(+)}$ has expanded. The self-join

distance profile between $\mathbf{T}^{(+)}$ and NEW is stored in $\mathbf{DP}_{last,m}^{(++)}$. The element-wise minimum between $\mathbf{MP}_m^{(++)}$ and $\mathbf{DP}_{last,m}^{(++)}$ is stored in $\mathbf{MP}_m^{(++)'}$, which is then updated to $\mathbf{MP}_m^{(++)NEW}$ after concatenating the minimum value of $\mathbf{DP}_{last,m}^{(++)}$. Finally, in lines 9 and 10, \mathbf{CP}_m is recomputed from the expanded $\mathbf{MP}_m^{(+),NEW}$ and updated and expanded $\mathbf{MP}_m^{(++)NEW}$.

The time complexity of *ContrastProfileI* is dominated by the MASS function, which performs an $O(n \log n)$ FFT operation. The time complexity begins as $O(|\mathbf{T}^{(+)}| \log |\mathbf{T}^{(+)}| + |\mathbf{T}^{(-)}| \log |\mathbf{T}^{(-)}|)$, but as the size of $\mathbf{T}^{(+)}$ dominates, the effective time complexity is $O(|\mathbf{T}^{(+)}| \log |\mathbf{T}^{(+)}|)$. Each time the function is called, MASS searches a slightly longer time series with n becoming $n + 1$. There are no conditional control statements, making the runtime value-invariant to the incoming data.

This time complexity discussion is a little indirect. A more intuitive way to measure the time requirements is by using the Maximum Time Horizon, which answers the question, “*How long can the Contrast Profile be maintained before the maintenance computation is slower than the sampling rate?*”

For example, consider the following two scenarios which refer to an Intel® Core i7-9700 CPU at 3.00GHz with 32 GB of memory (full worked details at [20]).

- If I have a Contrast Profile created with $\mathbf{T}^{(+)}$ and $\mathbf{T}^{(-)}$ both of length 10,000, and the data is arriving at 10Hz, then I can update the Contrast Profile for about 51 hours before the arrival rate is faster than my update time.

- Most automotive GPS loggers update at 1hz. If I have a Contrast Profile created with $\mathbf{T}^{(+)}$ and $\mathbf{T}^{(-)}$ both of length 10,000, with data arriving at 1Hz, then I can update the Contrast Profile for about 9.5 months before the arrival rate is faster than my update time.

Note that I do not specify the value of m in the above, as the update times are effectively invariant to the subsequence length due to the use of the MASS algorithm.

2.1.7 Anytime Contrast Profile

The computation time for the Contrast Profile is not particularly onerous relative to the tasks it can be used to solve. Nevertheless, it is natural to ask where the current limits of computation are in terms of the size of the datasets I can consider. Here I can take advantage of the fact that the Contrast Profile is largely comprised of calls to the *Matrix Profile*. In recent years Matrix Profile has become a widely studied algorithm, with multiple high-performance computational paradigms now available, including distributed computation and GPU implementations. Here I will show that I can also exploit the *anytime* computation property from the Matrix Profile [27].

Clearly it would be pointless to first compute $\mathbf{MP}_m^{(+-)}$, and only *then* compute $\mathbf{MP}_m^{(++)}$ in anytime fashion, as I would have to wait for about half the overall time to get the first approximation. Fortunately, the SCRIMP++ algorithm for the Matrix Profile has perfect *interruptibility* and *preemptability* [27]. This means I can spend $p\%$ of my time computing $\mathbf{MP}_m^{(+-)}$, then suspend it and then switch to spending $p\%$ of my time computing $\mathbf{MP}_m^{(++)}$, then toggle back to $\mathbf{MP}_m^{(+-)}$ etc.

The only question is what value ‘ p ’ should have. If there is a large cost in suspending and then resuming an algorithm, this can be a difficult question to answer. But for the SCRIMP++ algorithm, this cost is inconsequential (less than 0.0001% of the overall cost). Because of this, I can make p arbitrarily small. For simplicity I chose $p = 0.5\%$. I can measure the utility of the anytime property in two ways, the rate at which the best-so-far Contrast Profile converges on the final Contrast Profile (as measured by the RMSE), and by asking at what point the best-so-far Plato is as good as the final Plato. Here I define “as good as” to mean able to obtain within 1% of the latter’s classification accuracy on holdout data. In Fig 15 I answer both questions for the ECG example considered in Section 2.4.3.

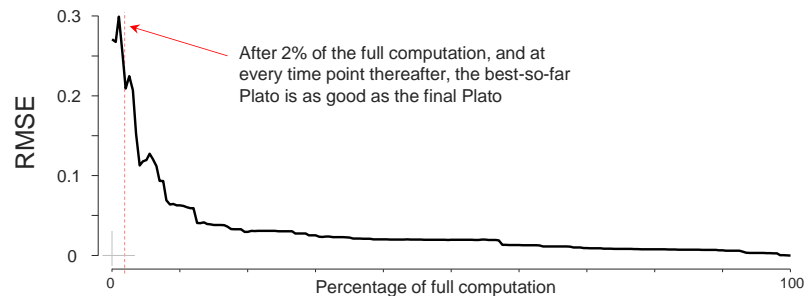


Fig 15 Anytime convergence plot for the ECG data shown in Fig 22.

Note that while the convergence plots for a Matrix Profile are strictly monotonic, this is not necessarily the case for the Contrast Profile. Nevertheless, as this figure shows, both numerically and semantically, the Contrast Profile converges very quickly.

2.2 Related Work

The work closest in spirit to the Contrast Profile is time series *contrast-set mining* by Lin and Keogh [25]. Their work proposes to solve the same problem of identifying subsequences that maximize the differences between two time series. This work predates the Matrix Profile but in retrospect can be seen as returning the subsequence pointed to by

the location that maximizes $\mathbf{MP}_m^{(+)}$. The problem with this definition is that it tends to simply return the most complex and/or noisiest subsequence. For example, for the data shown in Fig 8.*bottom*, it will return a subsequence of the pure noise at the right end of the $\mathbf{T}^{(+)}$ signal (see [5]).

If I generalize beyond time series, multiple instance learning, pioneered by Dietrich et al. [28], is another closely related concept. In order to identify a unique feature of a desired class, two labeled “bags” are provided: The class-positive bag must contain at least one sample with the desired property, while the class-negative bag must contain zero samples with the desired property. Multiple bags with these properties are required in order to identify the true instances within each positive bag. This is a much stronger assumption than requiring two true instances in a single positive “bag”, which is essentially the definition I propose. In addition, the requirement for zero instances of the target behavior in the negative bags is more restrictive than the relaxed assumption of fewer instances, proposed in Section 2.1.3. There have been some attempts to generalize this framework to time series by researchers noting that you cannot (meaningfully) convert the time series problem to a classic multiple instance framework by use of “sliding windows”. However, the proposed methods require significant feature engineering and a change of representation. In comparison, the Contrast Profile works directly on the *raw* data.

Shapelets [29] are defined on pre-segmented time series samples, but even if I suppose that they could be generalized to function with continuous time series, they will still have difficulty identifying point **C** in Fig 2. In the brute force algorithm, all possible subsequences within a range of subsequence lengths are initially candidates, which is a

desirable property that is shared by the Pan-Contrast Profile. Shapelets also make use of a type of contrastive function, *information gain* [30]. The main limitation for identifying point **C** is the assumption that the training classes are relatively free of labeling noise. Suppose one of the target classes contains only two true samples along with a dozen or more mislabeled samples, as seen in the introductory example of chicken telemetry in Fig 1. During the shapelet candidate elimination process, the first instance of dustbathing will be selected and the information gain will be computed for a set of candidates within $\mathbf{T}^{(+)}$ and $\mathbf{T}^{(-)}$. The second instance of dustbathing would be correctly classified and improve the information gain; however, it is likely that another non-dustbathing candidate would contain some subtle feature that would swing two or more mislabeled samples in $\mathbf{T}^{(+)}$ to the positive class, resulting in a greater information gain. This occurs because the information gain method was chosen under the assumption that the class would contain true labels. Thus, Shapelets work well at summarizing cleanly labeled and partitioned data, but they have difficulty under more relaxed assumptions.

Moreover, the Contrast Profile is tasked with finding discriminating exemplars, but is completely agnostic as to what classification algorithm will then be used (if *any*, as I will show, there are many uses of the Contrast Profile beyond classification). In contrast, [31] is tied to a particular Auto-Regressive Hidden Markov Model classification paradigm. Finally, the HMM requires the learning of six explicit parameters, whereas the Contrast Profile has one parameter, or zero parameters for the Pan-Contrast Profile.

The general literature of time series classification is vast; however, it is mostly orthogonal to this dissertation. As I noted above, virtually all research efforts only consider time series

objects after they have been extracted from a longer time series. However, they are silent as to *how* they can be extracted. In most cases, the community has bypassed this issue by only evaluating on the UCR archive, where many of the datasets were processed with human annotations and access to out-of-band information and domain knowledge.

2.3 Algorithms That Exploit The Contrast Profile

I believe that the Contrast Profile may be a useful primitive within dozens of higher-level algorithms. In this section I give some concrete examples.

2.3.1 End-to-End Time Series Classification

As I noted in the introduction, *given* discriminative subsequences (i.e., in the UCR format [5]) that characterize a behavior, time series classification is generally a simple task. I argue that *finding* such discriminative subsequences can be extremely difficult. Clearly the Contrast Profile has the potential to mitigate this difficulty. For concreteness, I outline a basic approach:

- Identify two snippets of time series that conform to the Contrast Profile assumptions (See Section 2.1.2).
- Run the Pan-Contrast Profile to discover the Plato.
- Use this Plato with a threshold t to discover similar instances, label them as the class that $\mathbf{T}^{(+)}$ represents.

Note that while the Euclidean distance is the natural distance measure to use, other measures such as DTW are possible [16]. I need to set a threshold; here I must resort to heuristics. For example I can use $3\times$ the distance for the Plato to its nearest neighbor (recall that I am assuming that the Plato's nearest neighbor is *also* an example of the desired

behavior). Finally, the above assumes that there is a single template for the desired behavior. If I think it may be polymorphic, I can use the technique discussed in Section 2.1.5 to find the Top-K Platos instead. This is a very simple technique for end-to-end classification, but as I will show on diverse real-word problems, *extremely* effective.

2.3.2 Exploratory Data Mining

I believe that the most frequent use of the Contrast Profile may be in exploratory data mining. That is, searching datasets with the goal of finding interesting hypotheses to further evaluate. As this is a difficult task to evaluate objectively, I will confine ourselves to a single example here. In Fig 16.*top* I show a pair of four-year long time series that record pedestrian traffic in two locations in Melbourne, Australia [32]. A city manager may wish to know, “*What happens only in Bourke Street but not elsewhere?*”

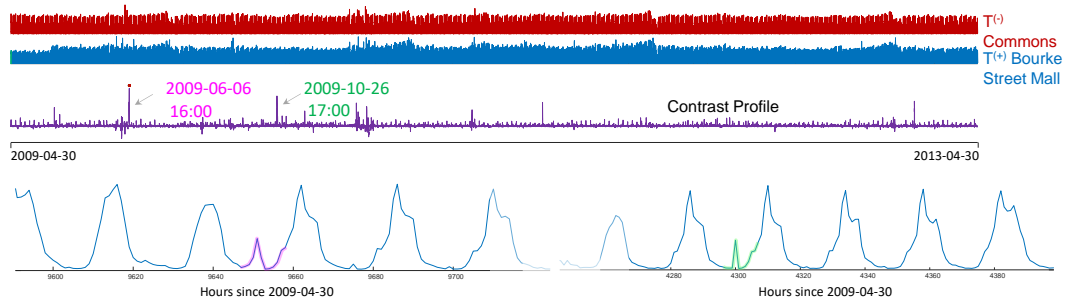


Fig 16 *top*) Two four-year-long time series and their CP_{12} . *bottom*) The two contrasting subsequences shown in context.

This is a difficult question to answer; the data have daily, weekly, and seasonal variation in addition to spikes and dips caused by both cultural events (protests, flash mobs, strikes) and severe weather events. As shown in Fig 16.*bottom* there is a subtle distinction between the two time series. Only the Bourke Street data has small spikes at what is otherwise the quietest part of the day. This might be a coincidence, but I have an additional 4.75 years of

data, and I find that this distinction holds up with a handful of similar events over the longer time period. What is the cause of this pattern? Using Reddit I crowdsourced that question, and one volunteer noted that the area is dominated by a branch of Myers, Australia’s largest department store chain. She went on to note *“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”*. In brief, the volunteer believes this burst is the evacuation of Myers during mandatory fire drills, with staff pouring out into the street. Whatever the reason, this example shows the potential utility of the Contrast Profile to find regularities that would otherwise be difficult to notice.

2.3.3 Anomaly Detection: Solving the Twin Freak Problem

While there are dozens of approaches for the task of anomaly detection in time series, it has long been known that shape-based methods (i.e. *discords* and their variants) can be competitive [8]. Here I propose a novel interpretation of anomaly detection, the Contrast Profile Anomaly Detector (CPAD). I propose creating a $\mathbf{T}^{(-)}$, which contains all the “allowable” behavior of a system, and then expanding $\mathbf{T}^{(+)}$ online as new data arrives (see Table 4). The $\mathbf{T}^{(-)}$ may be a natural time series or created from the concatenation of several time series. For example, if I wanted to build an anomaly detector to monitor pedestrian traffic at a train station, I could create a concatenation of one weekday, one weekend day, one bank holiday, one rainy day etc.

This idea is somewhat similar to “golden batch” monitoring, where engineers find or create an ideal representation of a manufacturing process, and continuously compare an ongoing process to it [16]. However, my formulation generalizes in several ways. I do not need to

obtain or understand the process in detail, I simply need to *obtain* data that I think covers the space of allowable behavior.

The CPAD has a very unusual property that makes it something of a hybrid between an anomaly detector and a regime-change detector. In particular, the first occurrence of a new pattern will not register a high score. It is only when the *second* occurrence is seen that the Contrast Profile value spikes to signal an anomaly.

I believe that there are problems for which this is the ideal behavior. Recall the $\mathbf{T}^{(+)}$ shown in Fig 7. It shows that ECGs are often contaminated with noise, which are typical sensor artifacts. I typically do not want an anomaly detector to alert for these (in any case, they are trivial to monitor with simpler methods). However, if I am monitoring a patient recovering in an ICU and I see novel beats, they may have sudden onset paroxysmal atrial fibrillation.

As with the previous task, I will confine my consideration to anecdotal examples, saving space for more forceful and quantitative evaluation of classification in the next section.

I consider data from a house that has individual appliances metered. This particular house happens to have two refrigerators. Refrigerator power demand has a very approximately square wave appearance, as the cycle between *on/off* compressor cycles. However, the timing of these cycles drift as the room temperature drifts and in response to someone opening the refrigerator door. In addition, at the very beginning of the *on* cycle, there is an increased demand as the compressor struggles to build rotational inertia. The latter variability is somewhat unique to each device depending on the rotational mass. This allows me to perform a natural experiment. I use a diverse selection of snippets from the

refrigerator time series as $\mathbf{T}^{(-)}$. I then use the online Contrast Profile algorithm outlined in Table 1 to monitor newly arriving data from the same stream. After 4.5 hours, I switched the leads to monitor the *other* refrigerator in the same house. This “refrigerator swap” models the situation where a system unexpectedly changes to allow instances from a new and unanticipated class. This is exactly what happens for certain cardiological diseases, or in batch processing if one component (say a valve or pump) begins to fail and then produces a different batch profile.

As Fig 17 shows, the difference caused by the refrigerator switch is too subtle to be seen, at least at this scale. Nevertheless, the CPAD algorithm clearly captures this event.

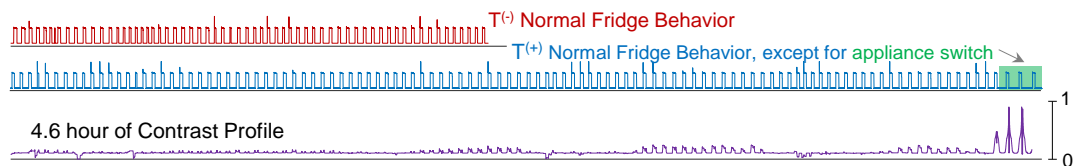


Fig 17 An example of CPAD. *top*) A time series that represents normal behavior of a fridge. *center*) A snippet from the same refrigerator, with the final moments swapped out for a different fridge from the same house. *bottom*) The Contrast Profile strongly spikes to indicate conserved novel behavior.

For this dataset I could update the online Contrast Profile 3,660 times faster than real time.

It is interesting to note that the classic discord definition has had one criticism levied at it for over a decade, in dozens of papers. In [33] the authors noted “*discords miss similar anomalies*” , likewise [34] notes the discords fail “*because my dataset includes several anomalies that are similar to one another*”. In other words, if there are two or more occurrences of undesirable behaviors, and they happen to have the same shape in each occurrence, then by definition it is a motif rather than a discord. Note that my framing of anomaly detection completely solves that problem, because as shown in this example, it

can *only* find anomalies that occur at least twice. Thus, the union of the discords discovered using the Matrix Profile [1] and the anomalies discovered with the proposed Contrast Profile can be used to create a hybrid-definition that includes all anomalies, independent of how often they occur.

2.4 Experimental Evaluation

To ensure that my experiments are reproducible, I have built a website [20] which contains all data/code for the results, in addition to many experiments that are omitted here for brevity. I have created a detailed document that with concrete details of my experiments, that I believe will allow anyone to reproduce all my experiments with less than one hour of effort. All experiments were conducted on an Intel® Core i7-9700 CPU at 3.00GHz with 32 GB of main memory.

While I hinted at several downstream algorithms that can exploit the Contrast Profile, I will mostly confine my attention to end-to-end classification. I remind the reader that the Contrast Profile is a data mining tool whose output can be used as the reference exemplar(s) for a nearest neighbor classifier, which allows for objective measures of utility.

As noted in the related work section, to the best of my knowledge, there is no other algorithm that performs prototype extraction under my assumptions. Where appropriate I consider the baseline of the default rate representing a *random* selection of K candidate behaviors.

2.4.1 Insect Behavior Classification

Sapsucking insects (insects in the orders Hemiptera and Homoptera) are insects that feed by sucking nutrients from plants. This behavior is typically not destructive by itself but can

spread diseases from plant to plant. Worldwide, across all crops/insects, this results in billions of dollars in crop losses each year. The primary tool used to study these insects is the electrical penetration graph (EPG), which as shown in Fig 18, produces a complex and noisy time series that reflects the insect's behavior [21].

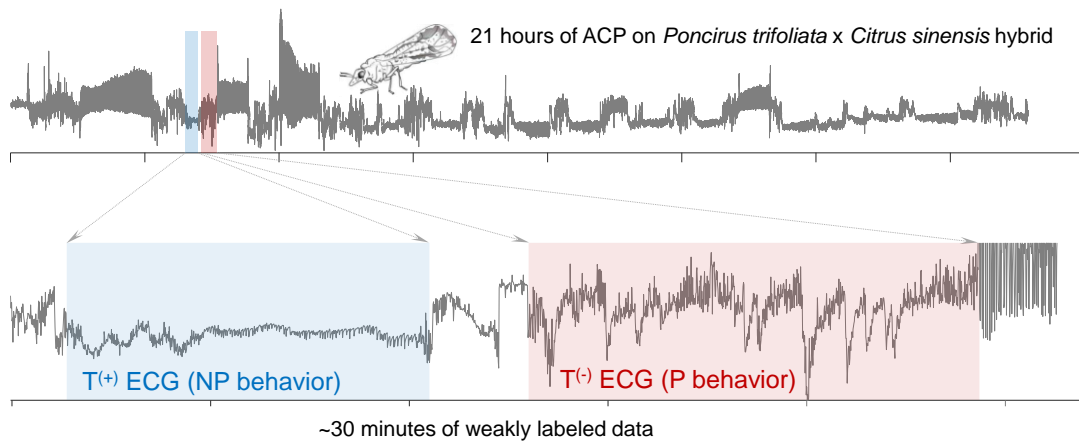


Fig 18 *top*) 21 hours of Asian citrus psyllid (ACP) feeding behavior on citrus. *bottom*) A zoom-in of a small fraction of the data.

I managed to obtain 21 hours of such data that was annotated by a combination of algorithms and humans (exploiting out-of-band information). Using the two regions shown in Fig 18.*bottom*, that conform to my algorithm's mild assumptions, I ran the Contrast Profile to produce the Plato shown in Fig 19.

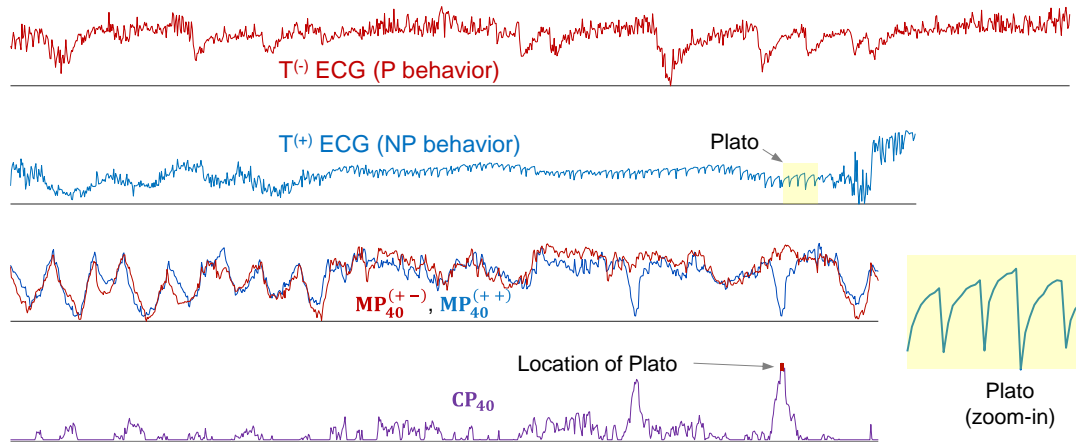


Fig 19 *top-to-bottom*) The weakly-labeled instances shown in Fig 18 have their $\mathbf{MP}_{40}^{(+ -)}$ and $\mathbf{MP}_{40}^{(+ +)}$ computed to produce the \mathbf{CP}_{40} , which strongly peaks to indicate the location of the Plato.

Using this template to find the Top-100 instances in the full dataset (excluding training data), the Plato had an error-rate of 7%, whereas the Top-1 motif in $\mathbf{T}^{(+)}$ had an error-rate of 32%, not much better than the default error rate of 36.9%

2.4.2 Chicken Behavior Classification

Here I revisit the chicken behavior example considered in Fig 1. First, I should explain why the data is weakly-labeled. The accelerometer worn by the bird *was* approximately synchronized with a video camera trained on the coop [35]. However, technical limitations meant that the synchronization had an error of up to ± 3 seconds [36]. By comparison, the dustbathing behavior I were tasked with quantifying is known to last about 0.5 to 3 seconds. Thus, a domain expert was able to locate thirty-second regions *with* and *without* the behavior, but not provide annotations at a *finer* temporal resolution. In Fig 20 I use the two time series shown in Fig 1 to compute \mathbf{CP}_{120} in an attempt to find a Plato that can act as a “signature” for dustbathing.

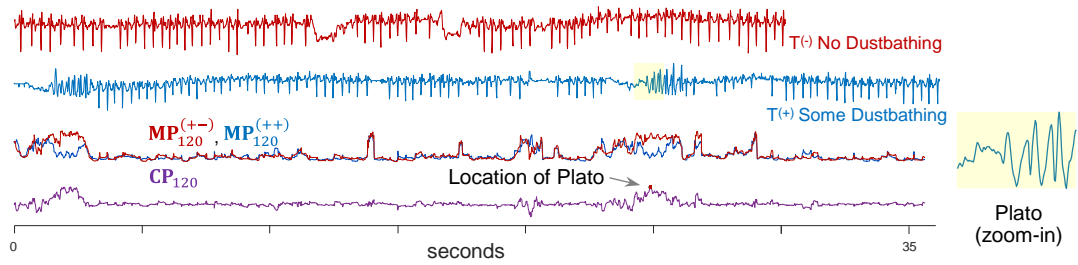


Fig 20 *top-to-bottom*) The weakly-labeled instances shown in Fig 19 have their $MP_{120}^{(+-)}$ and $MP_{120}^{(++)}$ computed to produce the CP_{120} , which strongly peaks to indicate the location of the Plato.

I used this Plato to search a 12,679,054,727 datapoint archive of chicken behavior for the one thousand best matches. The returned matches are shown in Fig 21.

Domain experts examined the results and confirmed that all the returned subsequences are true positives. Previously, dustbathing behavior was thought to be difficult to characterize using shape [37]. This is further evidence in the power of shape-based classification when the shape is selected well.

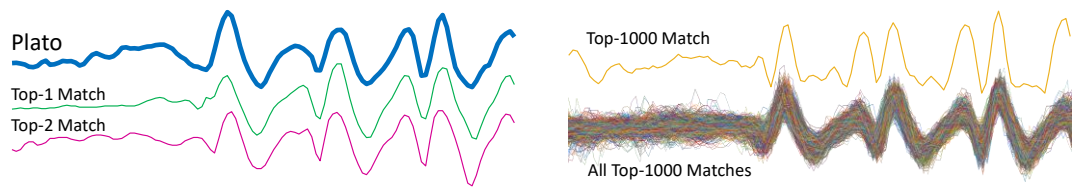


Fig 21 The Plato used for dustbathing classification (*top.left*). Selected matches returned by a nearest neighbor search using the Plato discovered in Fig 20. The Top-1000 matches (*bottom.right*).

The discovery of the Plato took 0.3 seconds. Surprisingly, the exact Top-1000 search in the 12.7 billion datapoints of disk-resident data (corresponding to four years of behavior) took only 55 minutes using the MASS algorithm.

2.4.3 Electrocardiogram Classification

I consider a 23.5 hour (10,828,800 datapoints at 128Hz), ECG dataset from a 46-year-old male (MIT 14046) [38]. The dataset has a beat-by-beat annotation created by a combination of specialized algorithms and human expert inspection.

I examine the first five minutes of the data to find a 30-second region free of anomalies $\mathbf{T}^{(-)}$ and a 30-second region with *at least* two (actually, *six*) anomalies $\mathbf{T}^{(+)}$, which I later learned are PVCs. Running the Pan-Contrast Profile for lengths 32 to 128 (a $\frac{1}{4}$ second to a full second) produced the 91-datapoint Plato shown in Fig 22.

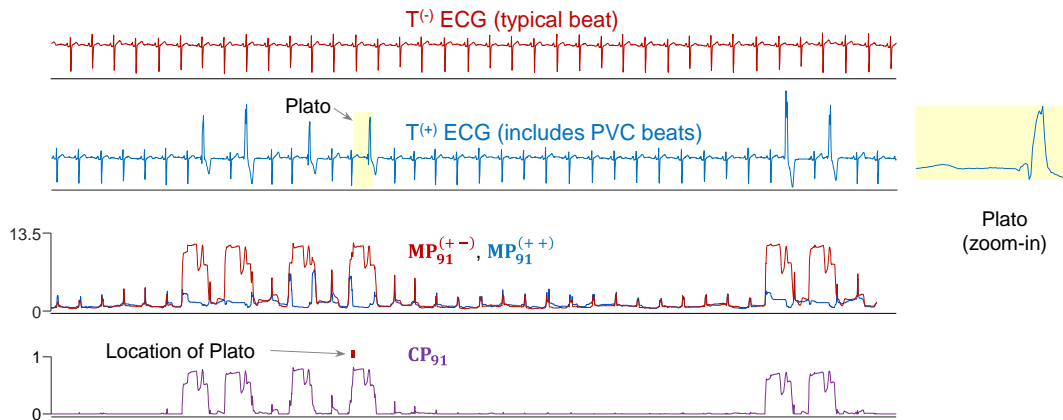


Fig 22 top-to-bottom) Two weakly-labeled snippets from the first five minutes used to compute their $\mathbf{MP}_{91}^{(+-)}$ and $\mathbf{MP}_{91}^{(++)}$, which then produce the \mathbf{CP}_{91} , with high amplitude peak to indicate the location of the Plato.

By (naively) extrapolating the density of PVCs I see in my tiny training snippets, I expect to see 3,934 additional PVCs in the remaining 23.42 hours of data. By retrieving the Top-3,934 nearest neighbors, I achieve 0.9992 precision. In contrast, the default rate is only 0.1645.

Here my recall was only 0.4031, because I underestimated the number of PVCs. If I “cheat” and retrieve the ground truth number of PVCs (there are 9,753), I obtain 0.9047 precision

and 0.9047 recall. As impressive as this is, these results appear to be pessimistic. An audit of my “false negatives” by cardiologist Greg Mason suggests that they are mostly mislabeled in the original data (discussion moved to [20]).

A recent paper surveyed thirteen approaches on datasets, including the one above (see Table 9 of [26]). While these works are not all directly comparable with each other or this dissertation, the accuracy I report would place me high in this list. More important however is the speed and simplicity of my method. Consider:

- The median number of parameters that need to be tuned by the thirteen methods in [26] is seven (most are based on CNNs or LSTMs); in contrast, I have no parameters to tune.
- For a much smaller dataset, [26] notes that it took 4.5 hours to train the model and then took 0.05 seconds to classify each beat. In contrast, I needed 16.3 seconds to learn my Plato model and 0.00000294 seconds to classify each beat.
- The other approaches worked with *extracted* beats and, thus, required domain dependent code to first do the extraction. In contrast, I worked with the *raw* data without explicitly extracting beats or using any cardiological knowledge.

Our ability to classify each heartbeat in just 0.00000294 seconds may strike the reader as being implausibly fast, but it is possible using Mueen’s MASS algorithm [2]. MASS takes just 0.3390 seconds to process a time series of length 10,828,800 containing 115,278 beats. This is a quarter of million times faster than real time. Of course, being so much faster than real time is of limited utility for monitoring an individual patient but is very useful for mining large data archives.

Finally, lest the reader think I chose an easy ECG, in [20] I repeated this experiment with other ECGs with similar results.

2.4.4 Electrocardiogram Classification with Relative Frequency Contrast Profile

I can revisit the experiment in the previous section to test the Relative Frequency Contrast Profile. As shown in Fig 23, I can replace the 30-second $\mathbf{T}^{(+)}$ and $\mathbf{T}^{(-)}$ regions used in Fig 22.*top* with new relaxed versions $\mathbf{T2}^{(+)}$ and $\mathbf{T2}^{(-)}$, which correspond to equally split snippets of the first five minutes of the MIT 14046. However, $\mathbf{T2}^{(-)}$ now has roughly half as many PVC beats as $\mathbf{T2}^{(+)}$, which violates the classic Contrast Profile assumption. If I compute the \mathbf{RFCP}_{91} with $maxFreq$ greater than the expected number of PVC beats, then I can search for types of PVC beats that are more frequent in $\mathbf{T2}^{(+)}$. The \mathbf{RFCP}_{91} heatmap in Fig 23.*bottom* reveals eight floating columns below the PVC beats in $\mathbf{T2}^{(+)}$. The low contrast regions below the columns indicate the number of beats in common.

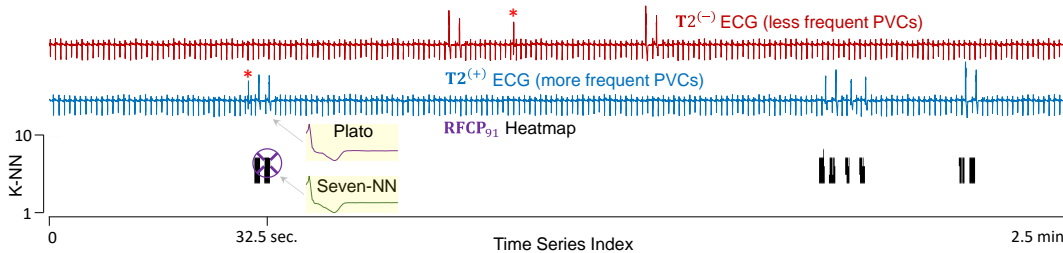


Fig 23 *top-to-bottom*) Two weakly-labeled snippets from first 5 minutes used to compute their \mathbf{RFCP}_{91} . The binary heatmap shows 8 floating columns below the PVC instances in $\mathbf{T2}^{(+)}$. The low contrast regions below the columns indicates the common number of occurrences of PVC between $\mathbf{T2}^{(+)}$ and $\mathbf{T2}^{(-)}$. The 7-NN is shown alongside the Plato to illustrate the high similarity.

Using the Plato discovered this way, I can search for the Top-3,934 nearest neighbors in the full 23.5 hour (10,828,800) MIT-14046 dataset (recall that 3,934 is a simple estimate of the number of PVCs in the full dataset based on extrapolation from the five minutes of

training data. Doing this I achieve a precision of 1.000, up from 0.9992. This accounts for an additional three correctly classified beats. By allowing ourselves to retrieve the ground truth number of PVC beats of 9,753, the precision increases from 0.9047 to 0.9199 when compared to the classic Contrast Profile. This represents an additional 148 correctly classified beats.

Before concluding this subsection, I turn my attention back to Fig 23 where there are two apparent discrepancies. The first spike initially appears to be a PVC spike like the rest, but it lacks a column in the \mathbf{RFCP}_{91} heatmap. On closer examination, its shape is fairly different. It does not appear in the heatmap because it does not have a nearest neighbor closer in $\mathbf{T2}^{(+)}$ than in $\mathbf{T2}^{(-)}$. In fact, its nearest neighbor distance is smallest to the third spike in $\mathbf{T2}^{(-)}$. This leads to the second discrepancy which is the heatmap columns should start at $K=6$ since there appear to be five PVC beats in $\mathbf{T2}^{(-)}$. The eight columns start at $K=5$ because the third spike in $\mathbf{T2}^{(-)}$ is dissimilar to the most frequent type of PVC beat in $\mathbf{T2}^{(+)}$.

2.4.5 Electrocardiogram Classification with Top-K Platos

I can confirm the presence of the less frequent PVC morphology found in Section 2.4.4 through use of the Top-K Platos functionality of the Contrast Profile, described in Section 2.1.5. This functionality is based on the classic Contrast Profile assumptions, which requires zero instances of the target behaviors in the negative class. With this in mind, I repartition the five-minute training time series so that $\mathbf{T3}^{(+)}$ corresponds to the first 250 seconds and $\mathbf{T3}^{(-)}$ is assigned the remaining 50 seconds, which lacks any apparent instances of PVC. As a reminder of the Top-K Platos algorithm, in each iteration the

previously learned Plato is concatenated to $\mathbf{T3}^{(-)}$, which effectively ignores all other instances of that shape when considering the next Plato to learn.

Viewing the distance profiles between the training time series and each of the Top-K Platos can help to clarify whether the learned Platos effectively represent behaviors within $\mathbf{T3}^{(+)}$. For clarity, I show only the Top-2 Platos in Fig 24 because the distance profiles of the Top-5 Platos revealed that the remaining three Platos were redundant. As I proposed in Section 2.4.4, the second most distinct shape in the time series, Plato-2, is a less frequent PVC morphology that appears only twice in the five-minute training time series. This is indicated by two locations with low distance in Fig 24.*bottom*. I can confirm the existence of these two morphologies by referencing Fig. 1.7 in [39].

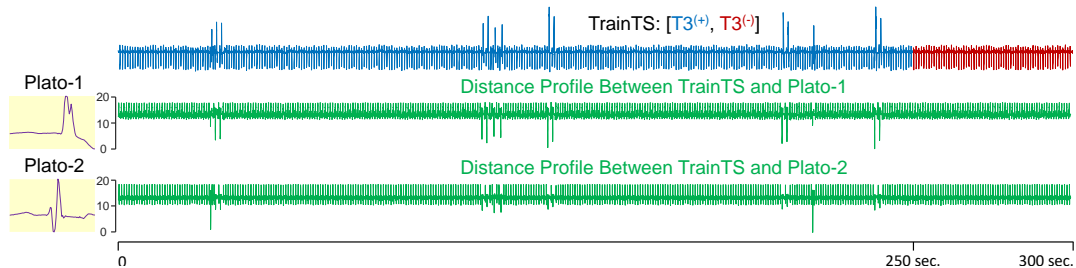


Fig 24 Investigation of discrepancies in Fig 23 have led to my identifying Plato-2 as a less frequent PVC morphology by using the Top-K Platos functionality of the Contrast Profile.

Now using the Top-2 Platos, I once again can search for the Top-3,934 nearest neighbors in the full 23.5 hour MIT-14046 dataset. When using the simple estimate of 3,934 PVCs in the full dataset, the testing precision is 0.9997, up from 0.9992 when using the classic Contrast Profile. The more significant improvement comes when searching for the ground truth number of 9,753 PVC beats. The precision is now 0.9480, up from 0.9047, which is an additional 422 correctly classified beats when compared to the classic Contrast Profile. This suggests that there may be additional PVC morphologies which were absent from the

training data. A summary of the precision results for the MIT-14046 dataset is shown in Table 5.

Table 5 MIT-14046 Classification Precision Summary.

Model	Top 3,934 Beats	Top 9,753 Beats	Default Rate
Classic CP	0.9992	0.9047	0.1645
RFCP	1.0000	0.9199	
Top-K Platos	0.9997	0.9480	

2.4.6 Detection of Novel Earthquake Events

I believe that seismology can benefit greatly from analysis incorporating the Contrast Profile. Data from seismic waves provides valuable information about the Earth's internal processes and structures. Exploring seismic data can be used to study earthquake origin, earthquake physics, fault behavior, and fault interaction, to determine Earth's velocity structure, and generally to constrain seismic hazards [40].

It is a simple process to detect earthquakes in case of large events and if there was good seismic sensor coverage since the earthquake signal would be above the background noise. In contrast, the detection of smaller earthquake signals is challenging, however, when the network of sensors is sparse or there is a high level of background noise (e.g., stations near the ocean can pick up waves crashing, and inland stations can pick up wind gusts). *Magnitude of completeness* is a term that seismologists use to describe the smallest earthquake magnitude that can be reliably detected [41]. The degree of completeness plays a major role in seismological hazard assessment and depends not just on the quality of the sensor network, but also on how sensitive the detection method is [42]. Traditional event detection methods in seismology use the ratio short-term average/long-term average (STA/LTA), which compute the average seismogram amplitude over a short time window

(i.e. several seconds) and over an extended time window (i.e. several minutes) [43]. The STA/LTA mostly functions based on the fact that a sudden increase in earthquake signal can cause a spike in the ratio [44]. This method is best suited for large events, but it is not sensitive to smaller earthquakes with weaker signals [45]. Therefore, many advanced methods have been developed in recent years, including query search (also known as template matching in seismology and signal processing), array processing and beam forming, statistical feature-based detections, local sensitive hashing using fingerprints, and machine learning methods [46][47][48][49], [50][51]. While these methods present advanced capabilities, there are many tradeoffs between flexibility, sensitivity, reliability and efficiency [49], thus most seismological observatories continue to employ both STA/LTA and human effort to detect and label earthquake signals [51][52][53].

The classic Matrix Profile has been recently proposed as a method to detect regular earthquakes as well as weak and novel seismic events like Low Frequency Earthquakes (LFEs) [18][54]. Seismologists are interested in LFEs since they can “potentially contribute to seismic hazard forecasting by providing a new means to monitor slow slip at depth” [55]. Despite its versatility and sensitivity for detecting any low signal-to-noise ratio events, applying Matrix Profile for detecting rare and novel events can be a challenging task. In most cases, the Matrix Profile may pick up events that were already in the catalog, requiring effort to eliminate these redundant discovered motifs. Here, I demonstrate that the Contrast Profile, which natively ignores known signals, produces reliable results for discovering rare, novel, and subtle seismic motifs.

To make my ideas concrete, here I demonstrate the value of the Contrast Profile for a single seismic station from the Northern California Earthquake Data Center (NCEDC) [56]. Digital seismic data has been recorded and stored at the NCSCN since 1984. The data is usually stored as snippets of seismic signals triggered by the detection of an event using traditional automated methods. In general, this is because the storage capacity at the time was not sufficient to store continuous seismic data records, which mostly contain background noise. Though recently, continuous data from some stations has been stored for the purpose of mining for novel events, such as LFE earthquakes.

In seismic stations, similarity in earthquake signal is determined principally by the source characteristics of the earthquakes and mainly by their point of origin. The closer two earthquakes are located, the more similar the waveform in a station far away will be [57]. There are normally many earthquakes that occur in close proximity to one another, which I refer to as *clusters* of nearby earthquakes. Often, one or more events in a cluster of earthquakes are large enough to be detected by conventional methods. As a result, the query search is very successful in detecting smaller events near cataloged events [47]. The main challenge of this approach is that if *all* members of a cluster of earthquakes are misidentified by conventional methods in data centers, then query search cannot be used to identify the events in a cluster. The Matrix Profile can be used for this case as a general form of similarity search. The problem is, these methods produce vast quantities of candidates, many of which have already been discovered.

Continuous earthquake data is huge and primarily contains background noise. For simplicity of presentation and to facilitate reproducibility, for the following experiment I

concatenated the snippets of the earthquake signals into two single traces of $\mathbf{T}^{(-)}$ and $\mathbf{T}^{(+)}$ as shown in Fig 25.

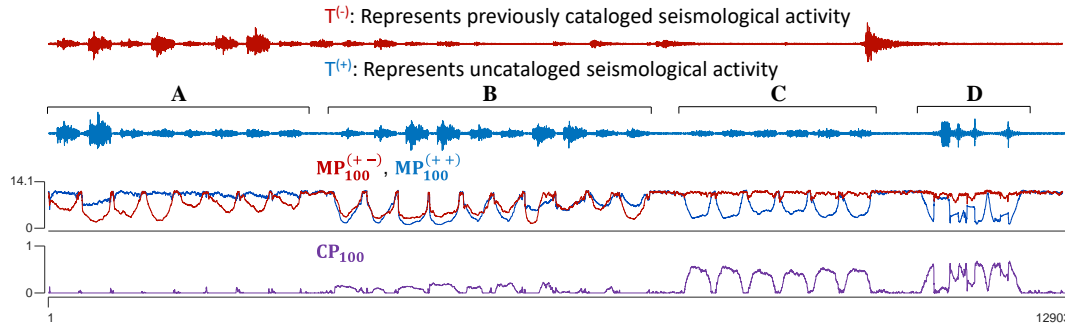


Fig 25 In scenarios A and B, the \mathbf{CP}_{100} appropriately ignores previously cataloged activity. In scenarios C and D, \mathbf{CP}_{100} appropriately highlights uncatalogued activity.

I included four different scenarios of earthquake signals in this data set in order to test whether the Contrast Profile can detect novel events in this set. Both $\mathbf{T}^{(-)}$ and $\mathbf{T}^{(+)}$ contain an earthquake snippet from eight clusters in scenario A. For example, I include a single earthquake in the $\mathbf{T}^{(-)}$ and a second earthquake from the same cluster in the $\mathbf{T}^{(+)}$. In scenario B, I included two events of a cluster in $\mathbf{T}^{(+)}$ and one event in $\mathbf{T}^{(-)}$. Scenario C includes two of the three clusters in $\mathbf{T}^{(+)}$, but none in $\mathbf{T}^{(-)}$. I have also included a section of the continuous record of the data that contains station artifacts (in informal seismologic terms, “glitches”); [58]) in D scenario. The gaps between sections A, B, C, and D are filled by background noise recorded from the same recording station. Other earthquake snippets and background noise are padded on the end of $\mathbf{T}^{(+)}$ to match the length of $\mathbf{T}^{(-)}$, this is a visual convenience, not something the Contrast Profile requires.

In scenarios A and B where a member of a cluster of earthquakes occurs in the $\mathbf{T}^{(-)}$, the Contrast Profile does not rise, as shown in Fig 25. When a cluster of events does not have any similar events in $\mathbf{T}^{(-)}$, such as scenario C, I observe a clear peak in the Contrast Profile.

It is possible to combine whole earthquake signals from the catalogs in order to form $\mathbf{T}^{(-)}$, and then search for novel events on continuous data by examining Contrast Profiles. A couple of examples of novel seismic events are earthquakes with unusual source characteristic [59] and LFEs, which usually require sophisticated methods and great effort to detect [55].

With the Contrast Profile, there is also very good detection of glitch signals in scenario **D**. Although station glitches tend to follow similar patterns most of the time, a universal deterministic way of detecting them has not been established in the seismology community, and several methods have been proposed to detect and remove these signals [NSPP]. The Contrast Profile clearly displays the glitches when the $\mathbf{T}^{(-)}$ dataset consists only of healthy seismic data (Fig 25). This provides an alternative way of detecting station glitches that can be used by seismologists.

In summary of this seismological case study, all events in scenarios **A** and **B** were appropriately ignored and all events in scenarios **C** and **D** were appropriately highlighted. The lack of sensitive parameters to set and clear and direct output representation of the Contrast Profile makes it very promising as a seismological tool.

2.5 Model Comparison

I will now compare the Contrast Profile as a classifier to two other leading time series models: shapelets and LSTM. I acknowledge that there is not a 1:1 correspondence in purpose of the models and I will explain the possible discrepancies.

2.5.1 Comparison to Shapelets

Time series shapelets have very different assumptions to the proposed Contrast Profile. However, the Contrast Profile can duplicate at least *some* elements of time series shapelets (the opposite is not true). To see this, consider the following:

I can produce a positive time series $\mathbf{T}^{(+)}$ by concatenating all exemplars of one class, and $\mathbf{T}^{(-)}$ by concatenating all exemplars of the other class(es). Given two such assembled time series, I can simply hand them over to the Contrast Profile. The **Trace** dataset from the UCR Archive [5], which does not have polymorphic classes, illustrates my purpose of extracting a single Plato. Using uniform m of $\left(\frac{1}{8} * sampleLength\right)$ for all four classes results in an error rate of 0.17, but the misclassifications come solely from the fourth class, whose main feature is a rising edge, which also appears in other classes. The error can be brought to 0.02 by training on multiple lengths and considering multiple nearest neighbors when learning the Plato, as will be discussed in more detail in Section 2.1.3. The foreshadowed Platons are shown in Fig 26. For comparison, the dataset default rate is 0.71, and the error rates for logical shapelets [60] and fast shapelets [61] are 0.01 and 0.09 respectively (results found at [62]).

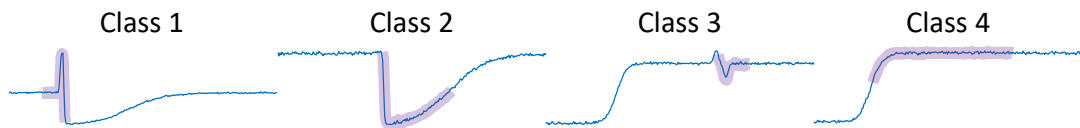


Fig 26 Samples of each of the four classes in the **Trace** dataset overlaid with Plato.

The Plato's error rate is competitive, given that shapelets have access to many strongly *labeled* extracted samples.

2.5.2 Towards Robust Classification: A More Forceful Comparison to Shapelets

In order to make a more forceful comparison to shapelets using the UCR Archive, I make use of the RFCP algorithm and set $maxFreq$ to the known number of training samples. Rather than demonstrating superiority, the purpose of this comparison is just to show that the class description power of Platos are commensurate with shapelets and I demonstrate this in a classification task. Due to this relaxed ambition, I am not concerned with highly optimizing the single Contrast Profile parameter.

I allow ourselves a choice of four subsequence length coefficients from the following set: $[\frac{1}{8}, \frac{1}{4}, \frac{1}{2}, \frac{3}{4}]$. Shapelets consider candidates of every length within a range, so they can be expected to perform at least slightly better. I do not anticipate this reduced length search being a limiting factor since my previous Pan-Contrast Profile experience indicates that contrast values are locally stable around a particular subsequence length. For each class within a dataset, the subsequence length resulting in a Plato with the greatest training contrast is chosen.

I learn one Plato for each class within a dataset in order to match the classic shapelets classifier, though it is possible to incorporate the Top-K Platos functionality into RFCP. For each trained class, the RFCP returns a data structure composed of $maxFreq$ rows and $|\mathbf{T}^{(+)}|$ columns. The rows of RFCP are flattened by calculating the root mean squared contrast (RMSC) of each column. The Plato corresponds to the index with the greatest RMSC. A comparison of the results are shown in Fig 27. The x-axis represents RFCP Plato accuracy, and the y-axis represents fast shapelet accuracy. To ensure the best possible results of my rivals, I used their results at [62]. The results I compare against contain only

a subset of the UCR Archive due to my rival's time constraints: Adiac, Beef, CBF, ChlorineConcentration, CinC_ECG_torso, Coffee, DiatomSizeReduction, ECG200, ECGFiveDays, FaceFour, FacesUCR, FISH, Gun_Point, ItalyPowerDemand, MALLAT, MedicalImages, MoteStrain, OliveOil, OSULeaf, SwedishLeaf, Symbols, Trace, TwoLeadECG, Yoga. I switch to presenting in terms of accuracy for visual intuition. The plotted points are weighted towards fast shapelets, but will soon investigate possible improvements for the RFCP Plato.

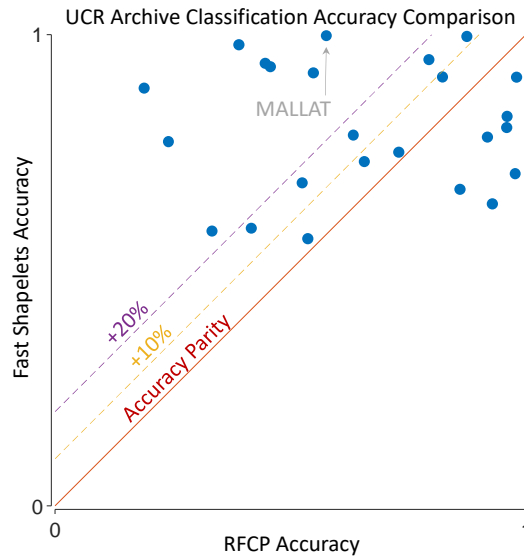


Fig 27 Accuracy comparison between classifying with RFCP Platos versus fast shapelets.

While the majority of plotted points favor fast shapelets, I will propose how they may be brought closer to parity. Incorporating the Top-K Platos functionality would help in datasets with polymorphic classes, though this would similarly also help fast shapelets. Another observation is that points further from the accuracy parity line are favored by the closer axis. These might indicate qualities within the datasets that are better captured by one of the classifiers.

MALLAT is one such dataset where classifying with logical shapelets excels despite a feature quality. RFCP Platos, which classify samples according to their most similar Plato, struggle when a subset of features are shared across classes. This concept is illustrated in Fig 28. Shapelets can handle this because using information gain to separate classes naturally lends itself to a decision tree which is not confused by this quality.

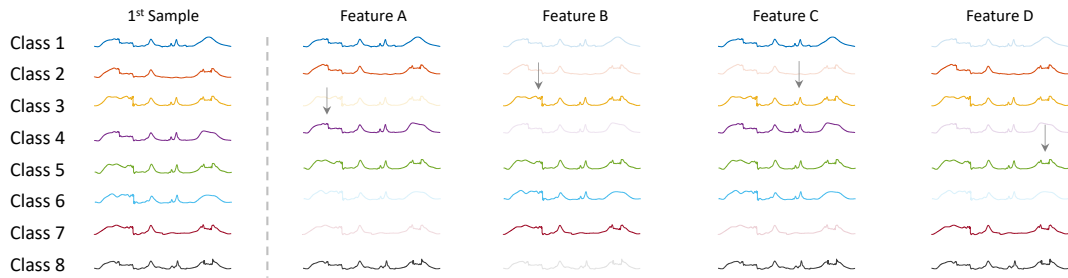


Fig 28 Classes composed of a subset of features are difficult to classify using a single Plato, as seen in the **MALLAT** dataset. *left*) The unobscured first sample of each class. *right*) Classes with common features revealed together.

I can test this theory by performing a lesion study that removes the observed feature quality.

By choosing to train and test on only classes 1 and 2, I remove the confusion caused by a subset of features appearing in multiple classes. My test results in an error of 0.0017, which strongly supports my theory. While the Plato's contrastive function is not as tightly integrated with traditional decision trees, it is conceivable that a tree architecture can be applied to Platos as well, which will greatly improve accuracy in datasets like **MALLAT**.

When considering time complexity, fast shapelets appears to have an advantage with $O(nm^2)$, with n training samples and sample length m . The terms n and m are grouped in the Contrast Profile since the algorithm runs on concatenated samples as a continuous time series, which results in the classic Contrast Profile running in $O((nm)^2)$ and the RFCP running in $O((nm)^2 \log(nm))$. Though, in practice, machine-specific optimizations play a

role as well. To give a concrete example from the UCR Archive’s **TRACE** dataset where $n=100$ and $m=275$, fast shapelets takes 158 seconds [62], the classic Contrast Profile takes 17 seconds, and the RFCP takes 673 seconds. The RFCP was relatively slow, but this is when running to completion on a single core. When incorporating GPUs and distributed systems, or exploiting the anytime property, the Contrast Profile becomes quite competitive since the contrast value of each time index can be computed independently, theoretically dividing out a factor of nm from the time complexity. The time complexity would then tend towards $O(nm)$ for the classic Contrast Profile and $O(nm \log(nm))$ for the RFCP, both lower than fast shapelets’ $O(nm^2)$ time complexity for datasets within the UCR Archive. Fast shapelets cannot benefit from such parallelization due to reliance on the SAX representation.

As an aside, I remind the reader that the UCR Archive was constructed with low labeling noise, if any. Thus, the prime advantage of the Contrast Profile was underutilized in this classification comparison. I expect that had datasets contained samples representing background noise and even samples from other classes, use of the RFCP would be even more competitive in comparison to shapelets as well as many other classifiers.

2.5.3 Comparison to LSTM

For the sake of completeness, I also include a comparison to an off-the-shelf LSTM approach [63]. This experiment comes with a disclaimer. The Contrast Profile is *not* a classification algorithm. It simply extracts patterns that can be used for nearest neighbor classification, one of the simplest classification algorithms.

I test on the chicken behavior classification from Section 2.4.2. As a reminder, the $\mathbf{T}^{(-)}$ (Class-0) training data contains behaviors to be ignored while the $\mathbf{T}^{(+)}$ (Class-1) training data contains at least two instances of the desired behavior, as well as other behaviors likely exhibited in $\mathbf{T}^{(-)}$. The LSTM's XTrain variable is constructed by extracting each subsequence from $\mathbf{T}^{(-)}$ and $\mathbf{T}^{(+)}$, then assigning a 0 or 1 accordingly in YTrain. Two parameter changes were made to the original model: [inputSize = 120, numClasses = 2]. The 51-minute training completed with 100% training accuracy. The test accuracy on a set of 2,000 samples balanced between dustbathing and non-dustbathing was 60.20% compared to 98.55% for the Plato. I have no doubt that the results for the LSTM could be improved by some preprocessing. However, my point here is simply that in this real-world challenging problem, I can quickly achieve results that are better than the community's current "go-to" solution, and I can do this with significantly less human effort and parameter tuning.

2.6 Conclusions

I have introduced the Contrast Profile, a novel data structure that allows a user or algorithm to reason about the differences between two time series. I reiterate that the Contrast Profile is *not* a classification algorithm, but it can help any downstream time series classification algorithms by finding discriminative prototypes. Beyond allowing end-to-end time series classification with only the weakest possible assumptions/annotations of the data, I have shown that the Contrast Profile has several other uses in data mining, including anomaly detection and data exploration.

There are several directions in which this dissertation could be extended. For example, in this dissertation I only consider the Euclidean Distance, however in some domains it is known that the Dynamic Time Warping (DTW) distance can be superior. Here, the fact that I directly exploit the Matrix Profile is an advantage, as in principle I should be able to simply “plug-in” a DTW-Matrix Profile. While DTW can be challenging to compute, recently SWAMP, a scalable DTW-Matrix Profile algorithm has been introduced [64]. I leave such considerations for future work.

Finally, I have shared all code and data with the community [20], to allow it to confirm and exploit my findings.

3. Novelets: A New Primitive that Allows Online Detection of Emerging Behaviors in Time Series

In this chapter, I introduce and discuss Novelets, a new Matrix Profile-based primitive that enables the online discovery of emerging behaviors. I demonstrate the practical applicability of Novelets in real-time monitoring of medical telemetry, wildlife, and industrial processes. Furthermore, I show that my algorithm has low sensitivity to two user-defined parameters, namely subsequence length and novelty threshold, providing a low learning curve for operators and high stability of results. By explicitly defining my ideas in terms of the Matrix Profile, I can take advantage of future developments in the MP community.

This chapter is organized as follows. In Section 3.1 I explain the essential definitions, notation, and related work. With these developed intuitions, I explain exploratory data

mining tasks in Section 3.2. I evaluate the utility of Novelet discovery in Section 3.3, followed by model comparisons in Section 3.4. Section 3.5 delivers conclusions.

3.1 Definitions and Notation

Our data type of interest is *time series*.

Definition 7: A *time series* $\mathbf{T} = t_1, t_2, \dots, t_n$ is a sequence of real-valued numbers.

For the task-at-hand I am not interested in any global properties of a time series but rather I am concerned with the shapes of small regions called *subsequences*.

Definition 8: A *subsequence* $\mathbf{T}_{i,m}$ is a contiguous subset of values from \mathbf{T} starting at index i with length m .

I can measure the distance between any pair of time series of equal length using a distance measure. I will use the ubiquitous z-normalized Euclidean distance [1]. One minor modification to the Euclidean distance is that I clip it at $\sqrt{2m}$ because values above this are anti-correlated in the Pearson Correlation space [65]. This is done in order to make the greatest use of the normalized range. If I wish to measure the distance between a short time series and every subsequence from a long time series, I can produce a *distance profile*.

Definition 9: A *distance profile* $\mathbf{DP}_i^{(AB),m}$ is the vector of distances between a query subsequence $\mathbf{T}_{i,m}^{(A)}$ and a reference time series $\mathbf{T}^{(B)}$.

The distance profile can be computed very efficiently using the MASS algorithm [2]. Fig 29 illustrates these definitions on a running example of a synthetic time series containing three types of signals. This example is comprised mostly of a (slightly noisy) sine wave which acts as the background pattern. There is also a region of pure noise, and three

examples of a modified sine wave that can be seen to “emerge” from the background pattern over time.

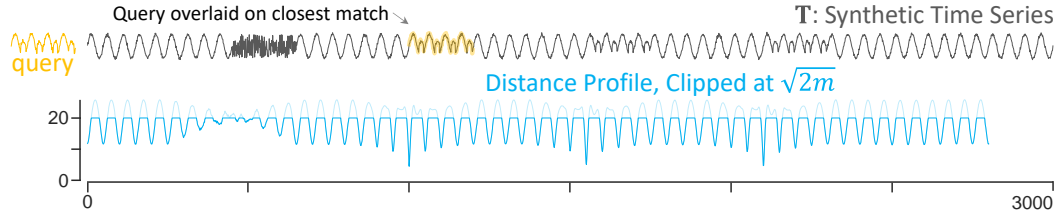


Fig 29 *top*) A short synthetic time series containing three signal types: a sine wave, a modified sine wave, and noise. *bottom*) A segment closely matching the modified sine wave is used as a query to generate distance profile $\mathbf{DP}^{(AB),200}$. Lower values indicate subsequence locations which are more similar to the query.

This conceptual element is helpful when constructing the *AB-join Matrix Profile* [1].

Definition 10: An *AB-join Matrix Profile* $\mathbf{MP}^{(AB),m}$ between reference time series $\mathbf{T}^{(A)}$ and a query time series $\mathbf{T}^{(B)}$ is a vector of Euclidean distances between each subsequence $\mathbf{T}_{i,m}^{(A)}$ and its nearest neighbor $\mathbf{T}_{j,m}^{(B)}$. Formally,

$$\mathbf{MP}^{(AB),m} = [\min(\mathbf{DP}_1^{(AB),m}), \min(\mathbf{DP}_2^{(AB),m}), \dots, \min(\mathbf{DP}_{n-m+1}^{(AB),m})]$$

Note that in general, $\mathbf{MP}^{(AB),m} \neq \mathbf{MP}^{(BA),m}$. Even if they do happen to have equal lengths, they correspond to different reference time series.

I incorporate $\mathbf{MP}^{(AB),m}$ into my running example in Fig 30. The top motifs are a pair of sine waves which are outlined in $\mathbf{T}^{(A)}$ and $\mathbf{T}^{(B)}$.

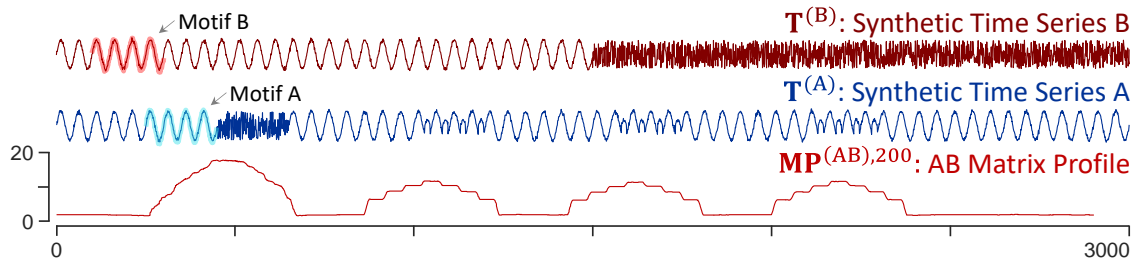


Fig 30 *top*) Time series $\mathbf{T}^{(B)}$ is synthetically created to contain equal lengths of sine wave and noise. *center*) Time series $\mathbf{T}^{(A)}$, first introduced in Fig 29, contains three signal types. *bottom*) $\mathbf{MP}^{(AB),200}$ reveals behaviors common to both $\mathbf{T}^{(A)}$ and $\mathbf{T}^{(B)}$ as low values and behaviors unique to $\mathbf{T}^{(A)}$ as higher values.

Subsequence comparisons can also be adapted to subsequences *within* a single time series using the *self-join Matrix Profile*.

Definition 11: A *self-join Matrix Profile* $\mathbf{MP}^{(AA),m}$ of a time series $\mathbf{T}^{(A)}$ is a specialization of the AB-join Matrix Profile with identical query and reference time series. An exclusion zone of length m is centered around each query index i for query subsequence $\mathbf{T}_{i,m}^{(A)}$ to suppress trivial nearest neighbors.

Fig 31 shows $\mathbf{MP}^{(AA),200}$ for my running example. Note that the embedded modified sine signal is neither a motif nor a discord.

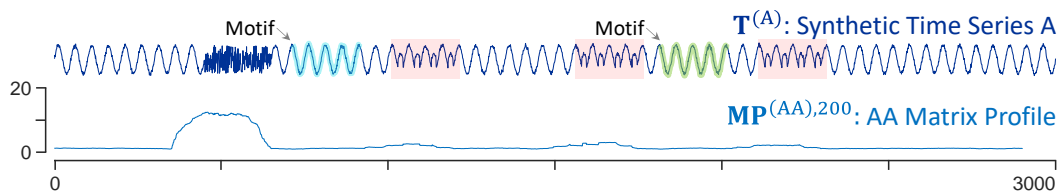


Fig 31 *top*) The synthetic time series $\mathbf{T}^{(A)}$ and its corresponding $\mathbf{MP}^{(AA),200}$ (*bottom*). The top motif pair are outlined. The three instances of the altered sine wave, which are highlighted in red, each have a corresponding mid-valued bump in $\mathbf{MP}^{(AA),200}$, which excludes them from consideration as either motifs or discords.

In online applications where future subsequences have not yet arrived, it can be useful to reason about the similarity between newly arriving subsequences and all the subsequences observed up to that point by utilizing the *left self-join Matrix Profile*.

Definition 12: A *left self-join Matrix Profile* $\mathbf{LMP}^{(AA),m}$ of a time series $\mathbf{T}^{(A)}$ is a specialization of the self-join Matrix Profile where each subsequence $\mathbf{T}_{i,m}^{(A)}$ is only compared to subsequences $\mathbf{T}_{j,m}^{(A)}$ where $j < i$.

The $\mathbf{LMP}^{(AA),200}$ of the running example is shown in Fig 32. The largest noticeable difference can be seen around the first instance of the modified sine. The difference in distances between the first and second instances offers a clue to discovering this elusive emerging behavior.

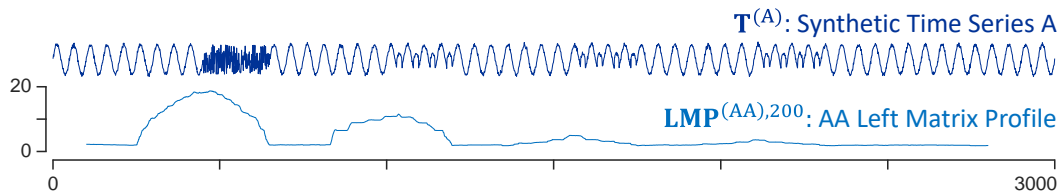


Fig 32 *top*) My running example time series. *bottom*) The left Matrix Profile $\mathbf{LMP}^{(AA),m}$ will highlight the first instance of a repeated behavior as an anomaly, but the first instance of the modified sine is still a moderate distance value, which excludes it from classification as a top discord compared to noise.

Note the similarity between $\mathbf{MP}^{(AB),200}$ from Fig 30 and $\mathbf{LMP}^{(AA),200}$ from Fig 32. They are very similar in most regions. This makes sense when the behaviors present in $\mathbf{T}^{(A)}$ and $\mathbf{T}^{(B)}$ are similar. Even noisy patches similarly have high distance within the two Matrix Profiles. Here is the key observation that informs my proposed algorithm:

Key Observation: $\mathbf{MP}^{(AB),m}$ and $\mathbf{LMP}^{(AA),m}$ only diverge in regions where there are emerging repeated behaviors in $\mathbf{T}^{(A)}$.

I am now in a position to introduce two new definitions. To do so, I will substitute the generic notations $\mathbf{T}^{(A)}$ and $\mathbf{T}^{(B)}$ for the specialized $\mathbf{T}^{(+)}$ and $\mathbf{T}^{(-)}$, which carry mild assumptions about their contents. In essence, I would like to identify emerging behaviors within $\mathbf{T}^{(+)}$ while suppressing those already known within $\mathbf{T}^{(-)}$.

Definition 13: An *Emergence Profile* \mathbf{EP}^m is the element-wise difference between Matrix Profiles $\mathbf{MP}^{(+ -),m}$ and $\mathbf{LMP}^{(++)},m$, where $\mathbf{MP}^{(+ -),m}$ joins $\mathbf{T}^{(+)}$ with $\mathbf{T}^{(-)}$, and $\mathbf{LMP}^{(++)},m$ is the left self-join of $\mathbf{T}^{(+)}$.

$$\mathbf{EP}^m = \mathbf{MP}^{(+ -),m} - \mathbf{LMP}^{(++)},m$$

The Emergence Profile is defined for any pair of time series which are longer than the subsequence length m . Moreover, it is still defined when the initial $\mathbf{T}^{(-)}$ is empty. In this case, $\mathbf{MP}^{(+ -),m}$ is defined such that it represents maximal dissimilarity between subsequences in $\mathbf{T}^{(+)}$ to $\mathbf{T}^{(-)}$. Concretely, it is a constant vector of length $|\mathbf{T}^{(+)}| - m + 1$, set to the maximum distance value. The first subsequence in $\mathbf{T}^{(+)}$ will emerge as the first Novelet and then it will be appended to the empty $\mathbf{T}^{(-)}$.

Fig 33 shows a discovered emerging behavior, identified as a peak in \mathbf{EP}^{200} . The discovery confirms by visual inspection that the corresponding location within $\mathbf{T}^{(+)}$ is the *second* instance of the behavior of interest (highlighted in red).

This definition resembles that of the recently introduced Contrast Profile [65], so I follow its normalization step which maximizes the effective normalization range by dividing \mathbf{EP}^m

by $\sqrt{2m}$, then clipping negative values to zero. This $\sqrt{2m}$ factor makes the results unitless, and independent of both the sampling rate and subsequence length.

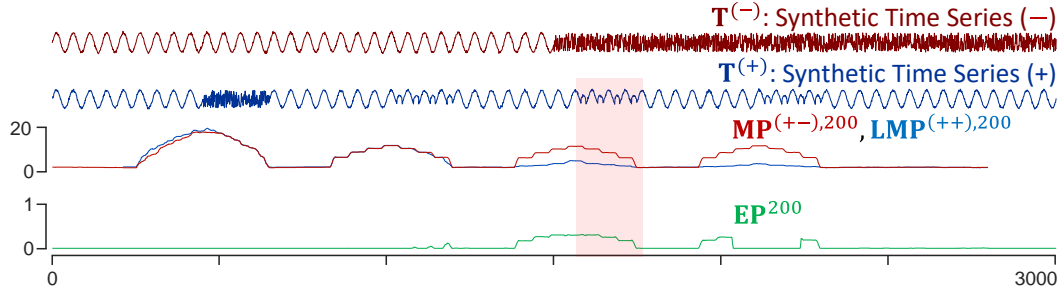


Fig 33 *top*) My running example time series. *middle*) By subtracting $\mathbf{LMP}^{(++) ,200}$ from $\mathbf{MP}^{(+),200}$, the second instance of the emerging modified sine wave is clearly highlighted within \mathbf{EP}^{200} (*bottom*). Once a new behavior is learned, future instances are automatically suppressed by adding the first emergence to $\mathbf{T}^{(-)}$.

The slight difference in definition leads to a desirable property. While the Contrast Profile must be completely updated as its time series grow, using the left self-join Matrix Profile means that the Emergence Profile is immutable as either $\mathbf{T}^{(+)}$ or $\mathbf{T}^{(-)}$ grows, a desirable property for online discovery of *Novelets*.

Definition 14: A *Novelet* $\mathbf{T}_{i,m}^{(+)}$ is the first instance of an emerging behavior. The second instance of an emerging behavior $\mathbf{T}_{j,m}^{(+)}$ produces a local maximum \mathbf{EP}_j^m above some novelty threshold d , and its nearest neighbor in $\mathbf{LMP}^{(++)}$ is the Novelet $\mathbf{T}_{i,m}^{(+)}$.

It is important to note the following. While a Novelet is the *first* instance an emerging behavior, it is never possible to recognize it as such at the instant it was encountered. It is only possible to *retroactively* recognize it as a Novelet, when I see a later occurrence. Before that time, I cannot discount the possibility that it was a literally unique event.

The identified first and second instances of the modified sine from the running example are outlined in Fig 34. The first two instances are enough to trigger a learning event, which results in the third instance being suppressed.

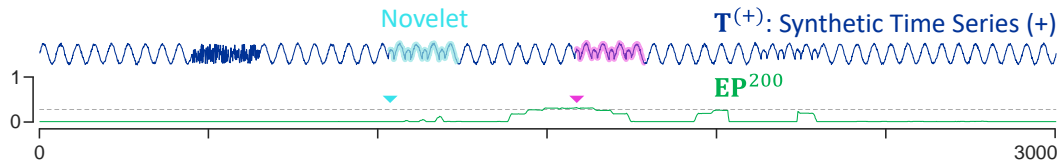


Fig 34 *top*) The first and second instances of the emerging modified sine wave are outlined in $\mathbf{T}^{(+)}$. *bottom*) The third instance of the emerging behavior has been suppressed in \mathbf{EP}^m by logically appending the first instance to $\mathbf{T}^{(-)}$ (not shown).

Whether discovered in batch or online mode, Novelets are discovered chronologically. Once discovered, a Novelet is logically appended to $\mathbf{T}^{(-)}$ so that future occurrences of the discovered behavior *are* suppressed in \mathbf{EP}^m .

Batch mode can generate \mathbf{EP}^m with a brute force nested for loop algorithm requiring $O(m|\mathbf{T}^{(+)}|(|\mathbf{T}^{(-)}| + |\mathbf{T}^{(+)}|))$. In real world applications, the cumulative length of all learned Novelets is much shorter than the length of $\mathbf{T}^{(-)}$, so this contribution can be ignored. If I consider the subsequence length m , which may be in the thousands, the naïve time complexity is clearly untenable. By building my definitions on top of the Matrix Profile, I can leverage a large and diverse community of researchers (academic and industrial) that continue to accelerate the Matrix Profile [1] [4][66], and I can apply Novelets in real time to diverse online applications.

The assumptions I make of the two time series inputs are quite mild. First, it is important to remember that behaviors occurring in $\mathbf{T}^{(-)}$ will be suppressed. If there are no pairs of sufficiently novel subsequences within $\mathbf{T}^{(+)}$, then no Novelets will be reported.

In addition, we are making common assumptions that the time series are regularly sampled and have no missing data. If these assumptions are unwarranted, it may be necessary to do data repair [67].

3.1.1 Differentiating Novelets from Related Concepts

While Novelets have analogues in text and other domains, I believe that they have not been considered by the time series data mining community before. Here I take the time to explicitly state the difference between Novelets and other time series primitives/definitions. Where appropriate, I will use the familiar trick of using text strings as a proxy for time series, with Hamming distance replacing Euclidean distance.

- **Time series motifs** [1]: While Novelets may be time series motifs, most motifs are not Novelets. For example, consider `..BeatzBeatyBeatxBeat..`. Here “Beat” is a strong motif that represents a known healthy heartbeat. If I exclude “Beat” when considering the incoming stream `..BeatwBeativBeatsPadakBeatqPatajBeatnBeat..`, I notice the less well conserved pattern “Pada/Pata” that emerges late in the string, which may indicate an emerging arrhythmia pattern. This analogy explains why motif discovery algorithms may overlook the emergence of less conserved motifs which may eventually become Novelets.
- **Time series discords/anomalies** [1][68]: An anomaly is typically a shape that occurs once, whereas a Novelet is the first instance of a repeated shape. For example, when de Vlamingh became the first European to see a black swan in 1697, it was a stunning anomaly to him. Since at least Roman times, the phrase “black swan” was used to mean

something that could not exist⁴. However, within days of the first sighting, black swans ceased to be anomalies, but simply became (temporarily) novel birds. As Fig 3 shows, anomalies are generally unique, and not newly emerging behaviors. It is only after I spot a *second* occurrence of an “anomaly” that I can reconsider it as an emerging pattern. While there are hundreds of anomaly detection algorithms [69][70] in the literature, none has this ability to retrospectively recategorize an “anomaly” as a newly emerged behavior.

- **Time series chains** [3]: Time series chains track gradually changing behaviors, while Novelets track sufficiently new behaviors. A chain must be of at least length three, whereas a Novelet may be detected with just two occurrences. When examining the word ladder “BaysBadsBadeBideTideTime”, I see a six-word chain of four letter words. Novelets will identify sufficiently new words along the chain, but will pass over words with high number of letters in common with a previously learned word. For instance, with a novelty threshold of three letters, the first novelet is “Bays” because it has three letters in common with “Bads”. Now every other word starting with B cannot be a novelet because they have at least one letter in common with “Bays”, and by definition, are one letter different from their neighbors, which leaves only two possible letters of novelty. “Tide” becomes the final novelet in this string because it has three letters in

⁴ The Roman satirist Juvenal wrote in AD 82 of *rara avis in terris nigroque simillima cygno* (“a rare bird in the lands, and very like a black swan”), meaning that since a black swan did not exist, the proposed “rare bird” did not exist. Here “rare bird” was not literally a bird, it is just something that did not exist, like an honest politician.

common with “Time” while having zero letters in common with “Bays”. When tracking tide levels over many years, time series chains will return the entire time series as a chain, while Novelets may return a dozen sufficiently distinct subsequences. I can best think of chains as a slowly changing behavior, say a slowly degrading batch cyclor or a slowly improving golf swing. In contrast, a Novelet is a completely novel behavior.

- **Time series segmentation/change detection** [71]: In a trivial case where there are two regimes, the output of segmentations and Novelets may both point to the start of each regime. For example, when monitoring an infant’s coaxed speech development, Novelets should identify the first instances of “MAMA” and “PAPA” in “MAMAzMAMAyMAMAxPAPAvPAPAtPAPA”. Segmentation will similarly identify the first “PAPA”, and the implied “MAMA” from the first regime. In a larger time series, segmentation will continue to identify changes in regimes, while previously discovered Novelets can be identified through a similarity search. Segmentation is most useful when there are relatively long regions representing a type of behavior. However, when monitoring a hatchling whose vocalizations may be less directed, such as in the string “CHIRPzTRILLyCHIRPvTRILLsCHEEP”, the utility of segmentation is dubious while Novelets would be able to identify the first instances of “CHIRP” and “TRILL”. As a practical matter, most time series segmentation algorithms only work well if the data is highly periodic, and they can see many periods per regime, say at least a few dozen walk cycles, followed by a few dozen run cycles. In contrast, Novelets only needs to see two examples of a new behavior.

- **Time series clustering** [72]: Time series subsequence clustering typically groups many (but critically, not *all* [73]) subsequences per cluster in an attempt to simplify class representation. While Novelets identify subsequences which are sufficiently different, there is no intent in the core functionality to assign group IDs to any subsequences similar to the Novelets. The string “DogHogBinGinFrogTin” contains two clusters of animals ending with “og” and objects ending with “in”. The Novelets with a new character threshold of two are “Dog” and “Bin”. When applied to song lyrics, Novelets would identify when new rhyme schemes *began*, whereas clusters would identify all words belonging to a scheme.

To summarize, I believe that Novelets represent a truly new primitive that is not captured by any existing definition.

3.1.2 Related Work

The previous section implicitly covered much of the related work. In addition, I should also consider rare time series motif discovery [74] which attempts to identify infrequent repeating patterns across large spans of time. In the effort outlined in [74], exactness guarantees are abandoned due to computational limitations. Instead, repeated patterns are identified with a probability proportional to their frequency. While the ambition is admirable, the SAX approximations at the heart of this approach require careful parameter tuning and allow false dismissals. Additionally, there is no mechanism to ignore well understood (background) patterns in order to focus on emerging patterns.

3.1.3 The Novelet Assumptions

Recall the mild assumptions of Novelets: $\mathbf{T}^{(-)}$ contains previously observed patterns which will be suppressed and only emerging patterns exceeding the novelty threshold will be identified within $\mathbf{T}^{(+)}$. In practice, Novelets can still be run with a completely cold start if there exist no established behaviors in $\mathbf{T}^{(-)}$, although there will be an early brief surge of emerging behaviors that will be reported, as *anything* will be considered novel in comparison to *nothing*.

One subtlety arises when processing time series in batches. Suppose Novelets are discovered on Monday, then Monday's batch is used as $\mathbf{T}^{(-)}$ in Tuesday's batch for Novelet discovery. There may be a single occurrence of a pattern within Monday's batch that was not reported as a Novelet. If a second instance occurs in Tuesday's batch, it will be suppressed. This can be solved by using an amnesic sliding window, as described in Section 3.1.5.

A pattern pair will only trigger Novelet discovery if they are sufficiently novel, but fortunately, setting the novelty threshold d is straightforward through a few heuristics. In practice, Novelet discovery has low sensitivity to the choice of d . A first sanity check is to run a self-join Matrix Profile on a representative training time series in order to determine if any motifs exist. The contribution of Novelets is to show where the first instance of an emerging motifs occurs, so if there are no motifs, then nothing will emerge. The next step is to identify whether there are any contrasting behaviors by running the Contrast Profile. With similar reasoning to the Matrix Profile, if there are no contrasting motifs, then a first contrasting motif cannot be identified.

Assuming that contrasting motifs exist, it is reasonable to set d to 80% of the contrast value of a compelling contrasting motif. While this process may require some trial and error, it is possible to quickly rediscover Novelets without recomputing the base $\mathbf{MP}^{(+ -),m}$ and $\mathbf{LMP}^{(++),m}$, which account for the bulk of the time expense. Note that the order in which Novelets are discovered can influence which future shapes are judged as Novelets. Setting d too low may result in shapes which appear out of phase or unintuitively novel. An alternative technique that eliminates this parameter is discussed in Section 3.4.1.

As is true with other habituating processes, an emerging pattern may pass under the novelty threshold due to past learned behaviors. If a sufficiently long time series has been monitored, the frequency of emerging behaviors slows to a trickle. Novelets may be useful as a canary-in-the-coal-mine for local change detection, so it may be beneficial to incorporate pruning functionality which forgets Novelets which have not been observed recently.

Finally, it is clear that Novelets only make sense in time series that have at least the potential to have repeated patterns conserved in *shape*. Some datasets may only have repeated patterns conserved in features or have shapes that are not well conserved under Euclidean Distance.

3.1.4 General Novelet Observations

The Emergence Profile is subsequence length normalized so that values are bound between zero and one. A value of one for subsequence $\mathbf{T}_{i,m}^{(+)}$ means that there exists a maximally conserved left nearest neighbor $\mathbf{T}_{j,m}^{(+)}$ while also a maximally dissimilar nearest neighbor

$\mathbf{T}_{k,m}^{(-)}$. Here, maximally dissimilar corresponds to uncorrelated subsequences, such as a pair of random noise subsequences, rather than anti-correlated subsequences.

The intentional consequence of subsequence length normalization is that Emergence Profiles may be compared across subsequence lengths and sampling rates. As there may exist emerging patterns at multiple scales, I can compute the Emergence Profile for all subsequence lengths within a range in order to determine appropriate subsequence lengths to focus on. If subsequence length m is too small, then I may report many sequential Novelets, though in reality, they represent pieces of a larger repeated pattern. If m is too large, then I am considering additional context of an emerging pattern, which may reduce its novelty score and prevent triggering Novelet discovery.

To demonstrate this, consider a snippet from an ECG containing some Premature Ventricular Contractions (PVC). Even healthy heartbeats can be challenging to model due to the variability in their beat length, but as Fig 35 shows, there is high stability in the Emergence Profile across subsequence lengths. By choosing a sufficiently high novelty threshold, I can inspect an Emergence Profile without the effects of past learned patterns. In Fig 35.*left* I show the “Pan-Emergence Profile” for an ECG from the Long-Term ECG Database with three instances of PVC [38]. The Pan-Emergence Profile is simply the Emergence Profile computed for every subsequence length in a given range, then plotted as a surface, which is first proposed in the Pan Matrix Profile [75].

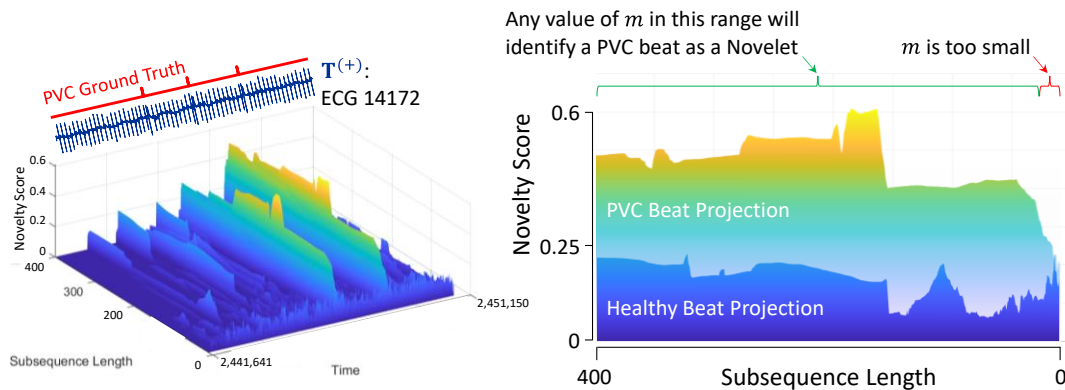


Fig 35 *left*) A surface plot of the Emergence Profile which contains two high-valued ridges corresponding to the second and third PVC instances. *right*) All but the smallest subsequence lengths will result in detection of the PVC beats, suggesting that Novelets are highly robust to at least one of the two user-supplied parameters.

There are two regions with high Emergence Profile values which align with the second and third PVC instances. This is emphasized in Fig 35.*right* where I see a projection of the healthy beats overlaid on a projection of the two PVC regions. This shows that all subsequence lengths in a large range have sufficiently high novelty threshold value. With a novelty threshold set to 0.25, only the PVC beats emerge as Novelets.

There are potential subtle pitfalls when using Novelets, but I suggest some workarounds. A Novelet may indicate a new behavior, but it may also reveal a twin freak anomaly. Just as it is impossible to know whether a behavior will be repeated having only seen the first instance, I cannot predict whether there will be more than two instances. One solution to this is to start a subprocess which pattern matches with Novelets and counts the number of following occurrences. Novelets may then be reasoned about as repeated patterns if they exceed n occurrences.

3.1.5 Online Novelets

The most compelling use case for Novelets is when used in an online fashion, allowing for the possibility of intervention to exploit or control the new behavior. There are a variety of definitions of “online”, but I use the useful variant that incrementally updates $\mathbf{T}^{(-)}$ and $\mathbf{T}^{(+)}$ at real-world sampling rates such that computed values of \mathbf{EP}^m are immutable.

The proposed immutability of \mathbf{EP}^m arises from the usage of its subcomponents $\mathbf{LMP}^{(++)},m$ and $\mathbf{MP}^{(+-)},m$. First, a left Matrix Profile is naturally immutable as newly arriving subsequences are only compared with past subsequences. The AB-join Matrix Profile $\mathbf{MP}^{(+-)},m$ may appear to require updating the entire profile if $\mathbf{T}^{(-)}$ grows, but my insight is that $\mathbf{T}^{(-)}$ grows only as Novelets are discovered. Any behaviors appended to $\mathbf{T}^{(-)}$ results in suppression of that behavior. A discovered Novelet which influences past $\mathbf{MP}^{(+-)},m$ values would invalidate its own discovery. Therefore, it is admissible to omit updating past values of $\mathbf{MP}^{(+-)},m$.

Now that intuitions have been developed for how variables need to be updated, I can discuss the naïve algorithm that updates the OnlineNovelets in Table 6 Following this, I show how the naïve update algorithm can be improved to support real-time discovery of Novelets.

TABLE 6 THE ONLINE NOVELETS ALGORITHM

Algorithm: OnlineNovelets ($\mathbf{T}^{(-),old}$, $\mathbf{T}^{(+),old}$, $t^{(+)}$, $\mathbf{MP}^{(+),m,old}$, $\mathbf{LMP}^{(++)},m,old$, $\mathbf{EP}^{m,old}$, m, d)	
Input: negative time series $\mathbf{T}^{(-),old}$, positive time series $\mathbf{T}^{(+),old}$, a new positive time point $t^{(+)}$ following $\mathbf{T}^{(+),old}$, Matrix Profile $\mathbf{MP}^{(+),m,old}$, Left Matrix Profile $\mathbf{LMP}^{(++)},m,old$, Emergence Profile $\mathbf{EP}^{m,old}$, subsequence length m , and novelty threshold d .	
Output: The Emergence Profile \mathbf{EP}^m , the incrementally updated Matrix Profiles $\mathbf{MP}^{(+),m}$ and $\mathbf{LMP}^{(++)},m$, the current positive time series $\mathbf{T}^{(+)}$, the current negative time series $\mathbf{T}^{(-)}$, and a <i>novelet</i> if discovered.	
1	$novelet \leftarrow []$
2	$\mathbf{T}^{(+)} = [\mathbf{T}^{(+),old}, t^{(+)}$]
3	$last \leftarrow \mathbf{T}^{(+)} - m + 1$ // index of newest subsequence of $\mathbf{T}^{(+)}$
4	$NEW \leftarrow \mathbf{T}_{last,m}^{(+)}$ // newest subsequence in $\mathbf{T}^{(+)}$ of length m
5	$\mathbf{DP}^{(+),m} \leftarrow \text{MASS}(NEW, \mathbf{T}^{(-),old})$ // Begin AB-join update
6	$\mathbf{MP}_{last}^{(+),m} \leftarrow \text{Min}(\mathbf{DP}^{(+),m})$
7	$\mathbf{DP}^{(++)},m \leftarrow \text{MASS}(NEW, \mathbf{T}^{(+),old})$ // Begin left self-join update
8	$\mathbf{LMP}_{last}^{(++)},m, j \leftarrow \text{Min}(\mathbf{DP}^{(++)},m)$ // Get min, argmin
9	$\mathbf{EP}_{last}^m \leftarrow (\mathbf{MP}_{last}^{(+),m} - \mathbf{LMP}_{last}^{(++)},m) / \sqrt{2m}$ // Length norm to $[0,1]$
10	If $\mathbf{EP}_{last}^m \geq d$
11	$novelet \leftarrow \mathbf{T}_{j,m}^{(+)}$ // The first instance of NEW behavior
12	$\mathbf{T}^{(-)} \leftarrow [\mathbf{T}^{(-),old}, novelet]$ // Suppress future occurrences

The algorithm begins in lines 1 and 2 by initializing an empty Novelet and appending the most recently arriving time point $t^{(+)}$ to $\mathbf{T}^{(+)}$. This newly arriving time point completes the newest valid subsequence NEW , located at index $last$ in lines 3 and 4. The query subsequence NEW and reference time series $\mathbf{T}^{(-),old}$ are input into MASS to generate a distance profile $\mathbf{DP}_{last}^{(+),m}$ in line 5. For this and the following index assignments, it is assumed that indexing past the existing vector length results in a vector expansion. Here

$\mathbf{DP}_{last}^{(+),m}$ represents the distances between the latest subsequence from $\mathbf{T}^{(+)}$ and all known subsequences which should be suppressed from $\mathbf{T}^{(-)}$. In line 6 the distance to the closest matching subsequence is then appended to $\mathbf{MP}_{last}^{(+),m}$. A similar process is completed for updating the Left Matrix Profile $\mathbf{LMP}_{last}^{(++)},m$ using the minimum distance of $\mathbf{DP}_{last}^{(++)},m$ in lines 7 and 8.

Now that the foundational Matrix Profile work is complete, line 9 subtracts the last distance value of $\mathbf{LMP}_{last}^{(++)},m$ from $\mathbf{MP}_{last}^{(+),m}$ to produce the newest length-normalized value of the Emergence Profile \mathbf{EP}_{last}^m . This difference value is the novelty score. The algorithm returns the updated vectors here if the novelty threshold has not been reached. Though if NEW exceeds a novelty score of d , then in lines 10 to 12, NEW 's nearest neighbor in $\mathbf{T}^{(+)}$ is returned as a Novelet and appended to $\mathbf{T}^{(-)}$ in order to be suppressed as a Novelet in future iterations.

In the simplified (for clarity of presentation) pseudo-code of Table 6, the time complexity of the OnlineNovelets algorithm is dominated by the $O(n \log n)$ MASS function, where n represents the larger of $|\mathbf{T}^{(+)}|$ and $|\mathbf{T}^{(-)}|$. After the arrival of b time points, the time complexity is $O(bn \log n)$. Eventually, when referencing some intermediate state, b is of equal magnitude to n , resulting in a time complexity of $O(n^2 \log n)$. By availing ourselves of the STOMP implementation of Matrix Profile [66], whose complexity is only $O(n^2)$, I can approach a time complexity of $O(bn)$ through processing newly arriving time points in sufficiently large batches of length b .

In practice, it can be difficult to gain intuition about an algorithm's computational demands without a concrete empirical grounding. To offer such a grounding I propose to consider the *maximum time horizon* (MTH): the answer to the question, “*How far back can the Emergence Profile refer before batch updates are slower than the sampling rate?*”

I consider the maximum time horizon of the two common sampling rates, run on an Intel® Core i7-9700 CPU at 3.00GHz with 32 GB of memory (full details at [9]).

- An Emergence profile initialized with ten periods in $\mathbf{T}^{(-)}$, with samples arriving at 100 Hz (a typical medical or human activity sampling rate), can reference back 4.4 hours when processing buffered input every minute.
- An Emergence profile initialized with ten periods in $\mathbf{T}^{(-)}$, with samples arriving at 1 Hz (a common industrial monitoring sampling rate), can reference back 3.2 years when processing buffered input every ten minutes.

The latter case is effectively *infinity*. Three years from now when I approach the MTH, Moore's law will allow me to reboot with a decade-long MTH. In the former case, I am close to being able to monitor a patient for an eight-hour operation recovery sleep. Moreover, I have a technique to gracefully degrade performance. As the dominating operations are the all-to-all pairwise comparisons, *time* will run out far before main *memory*. Since memory is of no consequence, it is possible to allow the OnlineNovelets algorithm to run until the MTH is reached, then transition to an *amnesic* mode where past time points are left out of future comparisons as new time points arrive. This slightly changes the meaning of Novelets to a behavior which emerges within a bounded time

window. A behavior repeated with an offset greater than the maximum time horizon will then be considered an anomaly, though any previously discovered Novelets will continue to be suppressed. When past samples are no longer referenced, the past results may be written to disk in a parallel sub-process, allowing the OnlineNovelets algorithm to effectively run indefinitely.

A further investigation of theoretical MTH for various sampling rates is shown in Fig 36.*left*. Real-time actionability is possible where the curves have a run-time gain (buffer-time divided by run-time) which is greater than or equal to 1.0. For example, the MTH for 1 kHz sampling is roughly 1.6 minutes ($|\mathbf{T}^{(+)}| = 10^5$). The subsequence length m and batch length for these curves were set to 100 and 1,000, respectively. In Fig 36.*right*, the choice of batch length is tested when sampling at 100 Hz with $|\mathbf{T}^{(+)}| = 10^6$. I notice that the choice of 1,000 ($10m$) as batch length for Fig 36.*left* was near the length of diminishing returns.

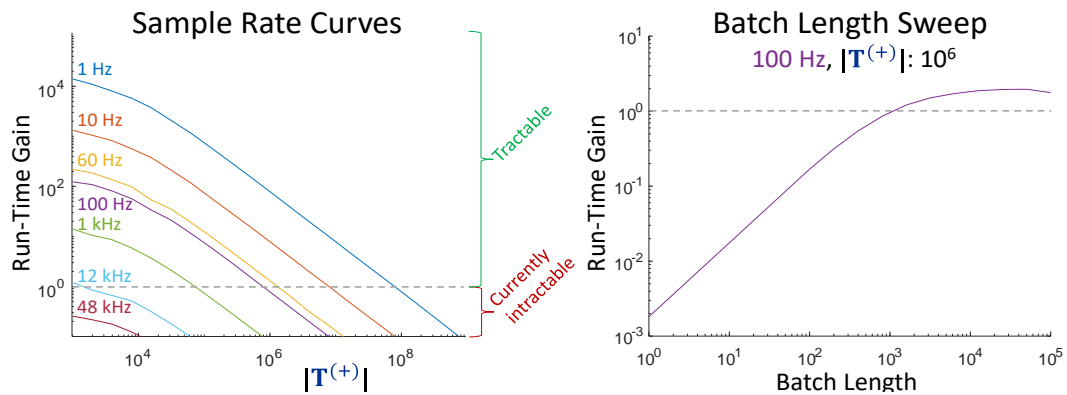


Fig 36 (*left*) For various sampling rates used in this dissertation, I show the theoretical MTH as the point where the sampling rate curve descends below a real time gain of 1.0 (gray, dashed). (*right*) I also explore the effect of buffer length on run-time gain.

3.1.6 Additional Implementation Details

The following Novelets implementation details have been useful for the investigation within this dissertation, though they are not essential to the Novelets definition. The reader may find optimizations which may better suit their use case.

3.1.6.1 Buffered Novelets

As noted in the previous section, processing incoming data when using online Novelets can improve the time complexity through use of the Matrix Profile. Rather than use MASS to compute a distance profile for each incoming time point, which is equivalent to the $O(n^2 \log n)$ STAMP algorithm [1], I update the **LMP⁽⁺⁺⁾** and **MP⁽⁺⁻⁾** Matrix Profiles through three calls to Matrix Profile routines, which makes use of the $O(n^2)$ STOMP algorithm. This technique reduces the time complexity from $O(bn \log n)$ to $O(bn)$. The BatchMPUpdate algorithm is described in Table 7.

TABLE 7 THE BATCHMPUPDATE ALGORITHM

Algorithm: $\text{BatchMPUpdate}(\mathbf{T}^{(-),old}, \mathbf{T}^{(+),old}, \mathbf{T}^{(+),new}, \mathbf{MP}^{(+),m,old}, \mathbf{LMP}^{(++)},m,old, \mathbf{LMPI}^{(++)},m,old, m)$	
Input: negative time series $\mathbf{T}^{(-),old}$, positive time series $\mathbf{T}^{(+),old}$, a new positive batch $\mathbf{T}^{(+),new}$ following $\mathbf{T}^{(+),old}$, Matrix Profile $\mathbf{MP}^{(+),m,old}$, Left Matrix Profile $\mathbf{LMP}^{(++)},m,old$ and its associated nearest neighbor indices $\mathbf{LMPI}^{(++)},m,old$, and subsequence length m .	
Output: The batch updated Matrix Profiles $\mathbf{MP}^{(+),m}$, $\mathbf{LMP}^{(++)},m$, and $\mathbf{LMPI}^{(++)},m$ and the current positive time series $\mathbf{T}^{(+)}$.	
1	$\mathbf{MP}^{(+),m,new} = \text{ABJoinMP}(\mathbf{T}^{(+),new}, \mathbf{T}^{(-),old}, m)$
2	$\mathbf{LMP}^{(new,old),m}, \mathbf{LMPI}^{(new,old),m} = \text{ABJoinMP}(\mathbf{T}^{(+),new}, \mathbf{T}^{(+),old}, m)$
3	$\mathbf{LMP}^{(new,new),m}, \mathbf{LMPI}^{(new,new),m} = \text{LeftSelfJoinMP}(\mathbf{T}^{(+),new}, m)$
4	$\mathbf{LMPI}^{(new,new),m} = \mathbf{LMPI}^{(new,new),m} + \mathbf{LMPI}^{(new,old),m} $
5	$\mathbf{LMP}^{(++)},m,new = \mathbf{LMP}^{(new,old),m}$
6	$\mathbf{LMPI}^{(++)},m,new = \mathbf{LMPI}^{(new,old),m}$
7	$\text{MIN_INDICIES} = \mathbf{LMP}^{(new,new),m} < \mathbf{LMP}^{(new,old),m}$
8	$\mathbf{LMP}_{\text{MIN_INDICIES}}^{(++)},m,new = \mathbf{LMP}_{\text{MIN_INDICIES}}^{(new,new),m}$
9	$\mathbf{LMPI}_{\text{MIN_INDICIES}}^{(++)},m,new = \mathbf{LMPI}_{\text{MIN_INDICIES}}^{(new,new),m}$
10	$\mathbf{MP}^{(+),m} = \text{Concatenate}(\mathbf{MP}^{(+),m,old}, \mathbf{MP}^{(+),m,new})$
11	$\mathbf{LMP}^{(++)},m = \text{Concatenate}(\mathbf{LMP}^{(++)},m,old, \mathbf{LMP}^{(++)},m,new)$
12	$\mathbf{LMPI}^{(++)},m = \text{Concatenate}(\mathbf{LMPI}^{(++)},m,old, \mathbf{LMPI}^{(++)},m,new)$
13	$\mathbf{T}^{(+)} = \text{Concatenate}(\mathbf{T}^{(+),old}, \mathbf{T}^{(+),new})$

The algorithm begins by addressing the behavior suppression step, which computes the distance between the new batch $\mathbf{T}^{(+),new}$ and the suppression time series $\mathbf{T}^{(-),old}$. The next eight lines approach updating the motif discovery step. On line 2, an AB-join Matrix Profile and nearest neighbor indices are computed between the new batch $\mathbf{T}^{(+),new}$ and the previously searched $\mathbf{T}^{(+),old}$. Since this is a batch operation where a pair of Novelets may exist within the batch, line 3 searches for motifs using a left self-join Matrix Profile within

$\mathbf{T}^{(+),new}$. Because the nearest neighbor indices within the new batch references the start of the batch, the frame of reference must be shifted to start at the beginning of the historic indices in $\mathbf{T}^{(+),old}$. The batch now has two separate distance profiles to the historic $\mathbf{T}^{(+),old}$ and new batch $\mathbf{T}^{(+),new}$, which must be consolidated. Lines 5 and 6 initialize the consolidated $\mathbf{LMP}^{(++)},m,new$ and $\mathbf{LMPI}^{(++)},m,new$ by setting them to the historic profile. If there are closer nearest neighbors within the new batch, they are identified on line 7 and set in lines 8 and 9. The algorithm finishes off in the last four lines by appending the new and old vectors.

The BatchMPUpdate algorithm can be run once for each newly processed batch, followed by Novelet discovery. Depending on the batch size, there may be multiple Novelets contained within a batch. This would require reupdating $\mathbf{MP}^{(+-),m}$ within a single batch update. It may be more convenient to keep discovered Novelets separate from $\mathbf{T}^{(-),old}$, and suppress previous Novelets in a separate subroutine while discovering new Novelets.

An additional consideration are the subsequences from the previous batch which either did not meet the minimum discovery length requirement (discussed in Section 3.1.6.2), or were simply incomplete subsequences. These unprocessed time points should be prepended to $\mathbf{T}^{(+),new}$.

3.1.6.2 Peak Selection

Fig 36 revealed how batch length affects execution time, though peak detection is another consideration when deciding on a minimum batch length.

When discovering Novelets online, selecting the first peak to exceed the Novelty threshold may not be ideal in most cases. It is often the case that the very next point has an even

higher novelty score. There are many online peak detection algorithms to choose from [76][77], though each is designed with specific properties in mind. My technique has some specific properties for the task of Novelet discovery.

The property I desire is that choosing a local peak will not affect a future greater peak within the exclusion length x . This corresponds to selecting the most novel behavior within a considered window. The exclusion length is the same as used in the Matrix Profile implementation, typically $m/2$, but for ease of explanation I have chosen to use m . Within a discovery window, I first identify the top peak. I apply the exclusion length x on either side of the top peak index. I then identify the next maximum outside the exclusion zone. The leftmost peak is chosen as the current candidate and compared against the novelty threshold. The active batch window is then shifted forward by x after the candidate peak, then the process is repeated. In the discovery window shown in Fig 37, this allows the maximum peak P1 to be selected on the next iteration. If the user feels that the misalignment for P2 is undesirable, then either the subsequence length or exclusion length should be shortened. The minimum discovery window length for this peak selection process is $2x$. When the first peak is directly in the middle of the discovery window, there must be at least one index outside the exclusion zone.

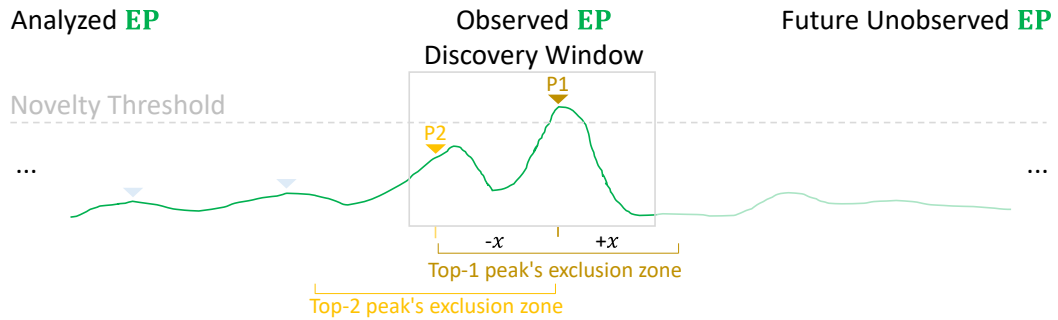


Fig 37 A demo Emergence Profile **EP** undergoing peak detection. The top peak P1 is first identified, then an exclusion zone of length m is placed on either side of the index. The next top peak outside of the exclusion zone P2 is identified. The leftmost peak is selected as the Novelet if it exceeds the novelty threshold, though it does not here.

3.1.6.3 Behavior Context

Whether Novelets are appended to $\mathbf{T}^{(-)}$ or to a separate list, the amount of context included will affect the utility of the learned Novelet. This choice is a “goldilocks decision”, where too little context will lack suppression power, while too much context will potentially result in suppression of immediately following Novelets. Context length will not affect the immediate Novelet, but it will affect the suppression of future instances of a Novelet and discovery of future Novelets.

When the context is too short, future instances of a Novelet may not be suppressed. This is illustrated in Fig 38 with zero context. While the subsequence containing the full Novelet is suppressed, subsequences containing a portion of the Novelet and other neighboring behaviors may be considered sufficiently novel to trigger a learned Novelet. When the third instance is encountered in Fig 38, **EP** dips sharply, but nearby indices remain unsuppressed. If the context is too long for Novelet behavior B1, then the context may contain enough of the first instance of another behavior B2. After B1 is appended to $\mathbf{T}^{(-)}$, it will suppress future instances of the behavior for consideration as a Novelet. If B2 is included within the

context of B1, then it may be sufficiently suppressed and miss discovery as a Novelet. While the **EP** looks very clean in Fig 38 when the context length is m , then there are two whole subsequences on either side of the Novelet which will be suppressed, despite having no influence on the triggering of the Novelet.

When the context length is roughly $m/2$, subsequences containing half of a previously learned Novelet will receive some degree of suppression. There is not an one-size fits all choice for context length, but $m/2$ is an intuitive balance. The trade off is false positive for short context lengths and false negatives for long context lengths.

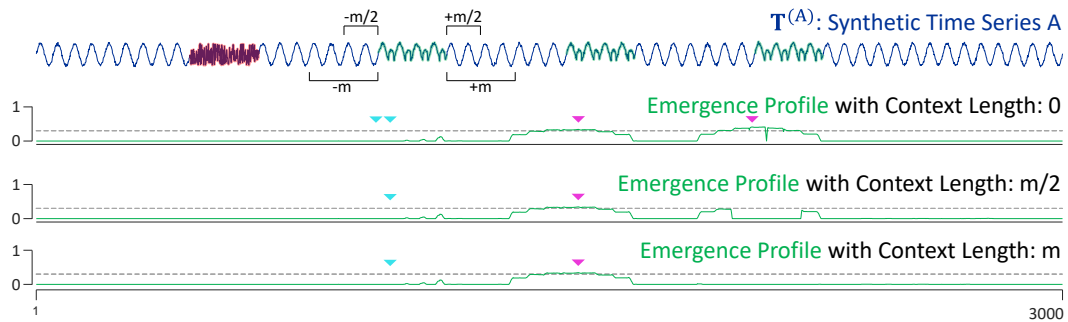


Fig 38 When considering the synthetic time series introduced in Fig 29, the Emergence Profile which stores Novelets without context is unable to sufficiently suppress the third instance of the Novelet. Increasing the context to between $m/2$ to m can suppress rediscovery of future instances.

3.1.6.4 Appending Novelets

Once a Novelet is learned, it is logically appended to $\mathbf{T}^{(-)}$, though in practice they are stored in a separate list of Novelets. Each processed buffer will need to suppress any behaviors within the of Novelets list. Had I stored Novelet without context, it would be appropriate to compute a distance profile using MASS, though since there is additional context included with the learned Novelet, an AB-join Matrix Profile is needed for each Novelet with context. If the context is sufficiently short, it may be less overhead to compute

many distance profiles, though I argue that this consideration should be handled within the Matrix Profile implementation, and is thus out of the scope of this dissertation.

3.1.6.5 Retroactive Identification Considerations

I claim to identify the first instance of an emerging behavior, though require at least two instances in order to do so. I cannot identify the first instance of a novel behavior without the second instance because I cannot distinguish it from noise or an anomaly.

Using my definition of Novelets, I can identify the second instance using the Emergence Profile definition presented in Section 3.1. Though, since I am computing left self-join Matrix Profiles, every new point will have a left nearest neighbor. It seems plausible that I can define an alternative Emergence Profile which will display peaks of first instances rather than second instances. Though when considering the details, problems come up.

First, let me assume that it is appropriate to update profile values in the following manner. For every new time point, compute its novelty score according to the OnlineNovelets algorithm in Table 6. Assign this novelty score to the left nearest neighbor. With this new profile, in order to identify whether a Novelet has emerged, the entire history of the Emergence Profile would need to be reexamined, which is not conducive to online discovery. Even with this concern aside, there is still a consistency issue. If the Emergence Profile is expanded with a new batch, previously discovered Novelets may become invalidated because peaks are mutable under this technique. So while I report the first instance of an emerging behavior retroactively, it is more practical that the Emergence Profile reflect discovery of the second instance.

3.2 Exploratory Data Mining

The discovery of emerging behaviors may be the end goal. Discovered Novelets can be reported to an end-user, and that may be the end of the analytical task. However, Novelets can potentially find patterns which have thus far gone unnoticed, and these patterns could be fed into any of dozens of downstream algorithms. Below I show some examples.

3.2.1 Electrooculogram

Time series telemetry from medical monitoring is often complex, leading to mostly feature-based or critical-limit monitoring. One such time series is an electrooculogram (EOG), which measures eye-related muscle movement. This is of significant interest, as eye movement can be a proxy for sleep-state and neurological disorders. I investigate the first record in the Sleep-EDF Database [38][78], which is roughly twenty-two hours long. When examining the top scoring Novelets with a three-second subsequence length, one Novelet was particularly interesting, as shown in Fig 39.*top*.

The center of the Novelet subsequence is a familiar shape corresponding to an eye-blink. What is interesting about it is that the leading and following sections are *also* part of blinks; by my now familiar analogy, in**kblink**lin. This is somewhat surprising, as blinks are normally thought to be random, one would not expect to see three in a row [79].

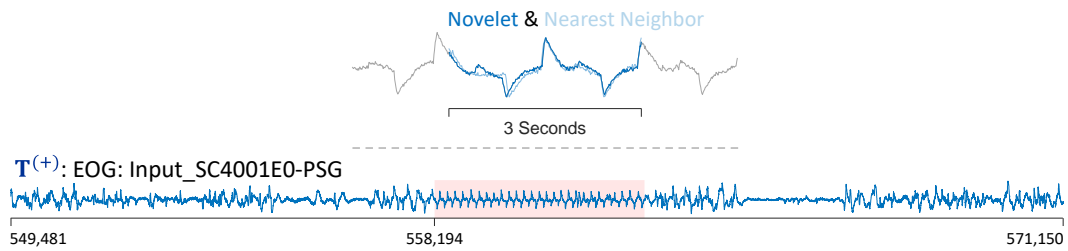


Fig 39 *top*) One of the top ten EOG Novelets corresponds to an unexpectedly long sequence of eye-blinks. Context in gray. *bottom*) A three-minute snippet from a twenty-two-hour EOG recording. The highlighted region of interest is a highly periodic pattern in an otherwise visually noisy time series.

Intrigued by this result, I searched the rest of the data using this Novelet as the template (using MASS, Definition 3) and discovered the extraordinary region shown in Fig 39.*bottom*. This region contains a series of twenty-six consecutive eye-blinks. I showed this data to Dr. Greg Mason who has more than forty years' experience in examining such data, and he found it to be astonishing. He had two theories that might explain this finding. The first is quotidian, perhaps a technician asked the patient to blink continuously to calibrate or test the recording apparatus. The second theory is that the patient has a rare neurological condition that produces a *fasciculation*. In either case, this demonstrates the utility of Novelets in the discovery of interesting and unexpected behaviors.

3.2.2 Pedestrian Traffic

Pedestrian traffic is a highly time-aligned behavior which often contains multi-scale periodicity. Typically such sensors measure the number of people passing through a control point in a given period of time. If a mechanical device, i.e., a *turnstile* is used, the counts can be essentially perfect. If a proximity sensor is used, as in the example shown in Fig 40.*top* there can be overcounts (a person with a large dog counted as *two*) or overcounts (a

child riding on her fathers shoulders), but in general they do accurately reflect the pedestrian ebb and flow throughout the day.

Motifs and discords are useful tools for uncovering the most conserved and most anomalous events within human behavior-based time series such as these multi-scale datasets [1].

Using Novelets, I have discovered that there are infrequent behaviors which are neither motifs nor discords. One such behavior, shown in Fig 40.*top* occurs at Tin Alley-Swanston St (West) at the University of Melbourne [32], only on two Friday evenings at 4pm, specifically February 2, 2017 and April 21, 2017. Other than this day of week and time similarity, the Novelet and its nearest neighbor do not appear to share a standard weekly, semesterly, or yearly periodicity. The peculiarity of these two events is reinforced when examining the distance profile in Fig 40.*bottom* between the three-year time series and the Novelet. There are clearly only two occurrences of this behavior.

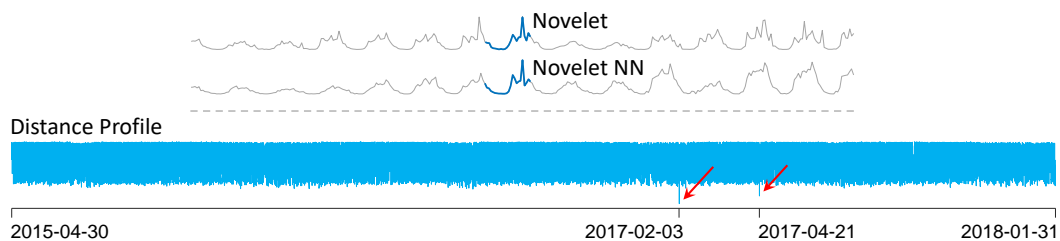


Fig 40 *top*) The Novelet and its nearest neighbor. *bottom*) Distance profile between the three-year time series and the Novelet. The distance profile reveals one repetition of the Novelet.

Given the physical location of the sensor, and how the behaviors are non-periodic, yet both occur on Fridays at 4pm, they may be campus recruitment events or other pop-up events. While it is difficult to know for certain with only historic information, when running

Novelets in real-time, it will be possible to flag such events as they occur in order to catalogue them for future reference.

In domains with a similar hourly sampling rate where behaviors are tracked at the 24-hour time scale, the maximum time horizon is essentially unlimited, where a thirty day buffer referencing the past thousand years takes less than a minute.

3.3 Experimental Evaluation

In order to ensure that all my experiments and figures are easily reproducible, I have built a website [9] that contains all the data/code used in this dissertation. All experiments were conducted on an Intel® Core i7-9700 CPU at 3.00GHz with 32 GB of main memory, unless otherwise stated.

I conducted experiments that test the following claimed properties of Novelets:

- With high probability, Novelets can discover emerging behaviors, even in noisily “imperfect” data.
- As a corollary to the above, *zero* novelets will be returned if there is no true emerging behavior. Some algorithms always return *some* structure. For example, the Matrix Profile always returns the best motif, even in random data. I do not want practitioners wasting their time considering spurious emerging patterns.
- My algorithm allows superhuman pattern discovery. Even if my algorithm could only discover patterns that are visual to the human eye, it would still have value, as human attention is a valuable resource. However, I will show that patterns in a time series which may be too subtle to be visually identified can still potentially be identified with Novelets.

- Novelets are robust to past false negatives such that if the first two instances of a behavior were not identified due to noise, either one is still a candidate in matching the k^{th} instance. This is an important property. The first occurrences of a new emerging pattern may be too tentative or noisy to be discovered, but I do not want processing them to spoil the chance of later discovering a slightly clearer version.

Given the above, my experiments are a mixture of somewhat anecdotal examples that show the diversity of problems that Novelets can be applied to, and more rigorous experiments designed to stress test my ideas. I begin with an example of the former.

3.3.1 Bird Song Detection I

Recall the motivating application in the introduction describing an observed change in the song of a white-throated sparrow. The song's rhythm changed from “*Canada*” to “*Cana*”. I believe that Novelets in conjunction with existing wildlife monitoring efforts could detect instances of such a change in behavior.

To avoid constructing and potentially contriving a dataset to test this idea, I will simply use the audio of a ten-minute long video describing the phenomenon [80]. The video in question includes narration, other ambient sounds, and music, however I do not edit it in any way. I use a single 62.5 Hz Mel-Frequency Cepstral Coefficient (MFCC) [81] as $\mathbf{T}^{(+)}$ and habituate to the narration in the first half of the video by partitioning it as $\mathbf{T}^{(-)}$ as shown in Fig 41. Because appropriate values of the novelty threshold are domain dependent, we first generate the Emergence Profile with the threshold set to the upper limit of one to reveal all peaks without suppression.

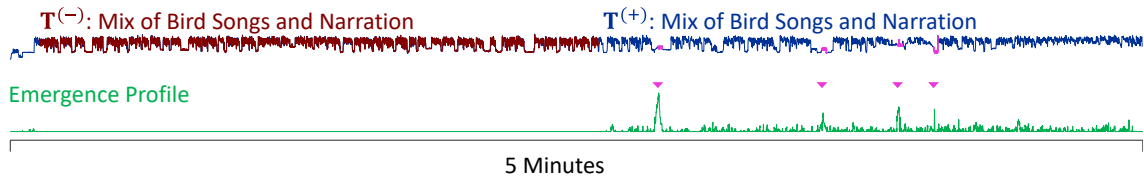


Fig 41 Top) The first half of the audio is set to $T^{(-)}$ so that future narration in $T^{(+)}$ is suppressed. Bottom) The Emergence profile peaks in four locations above 0.15.

When searching among two-second-long subsequences, four peaks stand out. To further investigate, we look at a histogram of peaks of non-overlapping subsequences, excluding zero as a peak. In Fig 42 we see four obvious peaks which are offset from a cluster of peaks. A threshold value of 0.15 provides a wide margin between the four peaks and the cluster of lower scored peaks.

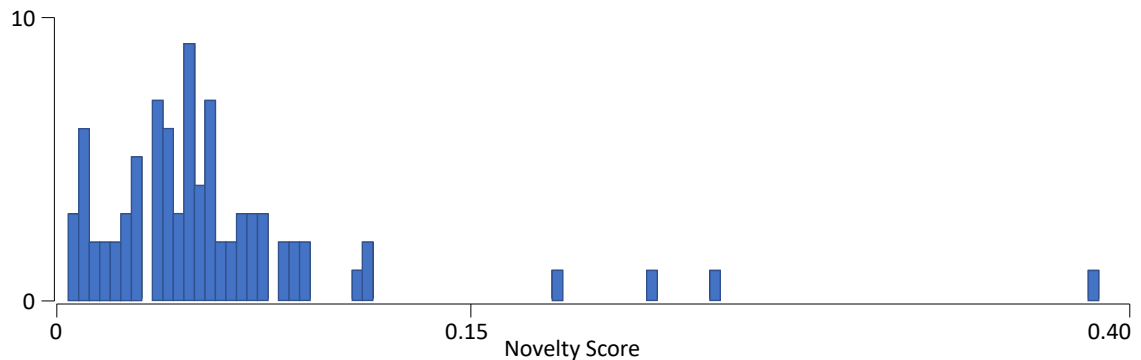


Fig 42 Histogram of non-overlapping peaks. A novelty threshold value of 0.15 gives a wide margin between the cluster of low-novelty peaks and the peaks with higher novelty.

The four Novelets identified are the song transcribed as “*Canada*”, the song transcribed as “*Cana*”, a solitary chirp, and silence preceding a higher amplitude sound. Fig 43 shows the two Novelets corresponding to the two song compositions.

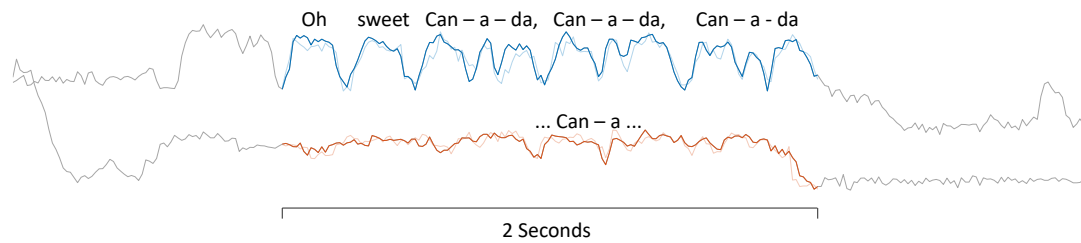


Fig 43 The emergence of the two song compositions of the white-throated sparrow is detected using Novelets. Novelets and their second instances are shown in color with context shown in gray.

There is a quality difference between the two Novelets shown in Fig 43. This is likely a result of two distinct birds singing at different pitches, which can appear quite different when decomposed as MFCC components. Nevertheless, this result is promising, given how little effort I needed to create this result, and the fact that I used no domain knowledge.

3.3.2 Bird Song Detection II

The previous section described an anecdotal example of the utility of my proposed idea. I will now more forcefully demonstrate the ability to identify a newly emerging bird song among a variety of other bird songs and forest sounds.

To demonstrate these properties in a principled manner, I have constructed two natural test sets with unambiguous ground truth, while also preserving real-world noise levels. Each test contains one hundred subtest iterations. Each subtest is composed of time series pairs, where $\mathbf{T}^{(-)}$ represents previously recorded forest sounds [82] known to lack the bird song of interest and $\mathbf{T}^{(+)}$ represents a similar distribution of forest sounds along with random insertions of four different instances of the bird song of interest. The bird songs of interest come from a white-crowned sparrow (*Zonotrichia leucophrys*) [83]. The bird songs WAV values are added to the forest audio WAV values with amplitudes scalers that emulate a bird in the foreground in one test and in the background for the second test. The time series

pairs are then decomposed into MFCC components, using default settings. In principle, MFCC is optimized for human speech, not bird songs. However, in practice it works well [84]. Because the domain and decomposition technique is essentially identical to the experiment in the previous section, we use the same novelty threshold of 0.15.

The test sets were constructed so that the prior probability of correctly selecting any combination of two of the four songs of interest within a single subtest (i.e., the default rate) is less than 1%.

Identifying the background bird songs is non-trivial. The synthesized time series with four instances of background bird songs in Fig 44.*top* shows the difficulty in identifying the target behaviors using visual inspection of amplitudes and shapes. The self-join Matrix Profile [1] is a standard tool for identifying motifs and discords within time series. The nearest neighbor distance of the identified Novelet is represented as a dashed gray line in Fig 44.*center*. The nearest neighbor distance is a moderate MP value, indicating that it is neither a motif nor discord. The Emergence Profile successfully identifies the first and third instances of the target behavior, as confirmed by the alignments of ground truth and Novelet indicators in Fig 44.*bottom*. The Emergence Profile of the test iteration shown in Fig 44.*bottom* illustrates Novelets' robustness to false negatives. While Novelets fail to identify the target behavior within the first two instances, the third instance matched with the first instance.

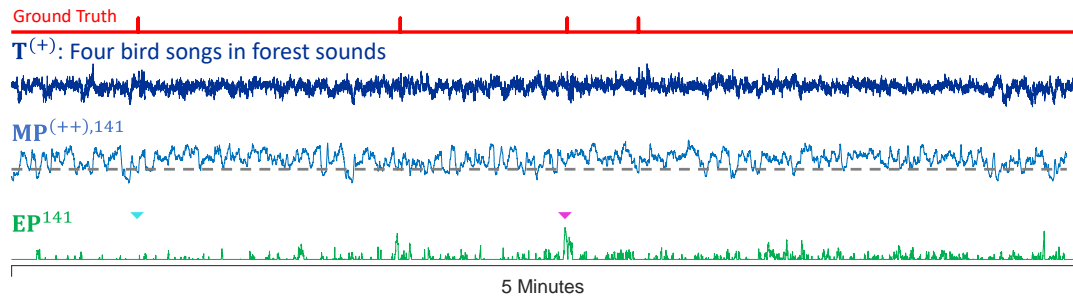


Fig 44 *top*) Synthesized time series $\mathbf{T}^{(+)}$ with four embedded bird songs with locations indicated by ground truth annotations (red). *center*) Self-join Matrix Profile $\mathbf{MP}^{(++)},141$ with the nearest neighbor distance of the first instance indicated by a dashed line (gray). *bottom*) Emergence Profile \mathbf{EP}^{141} with Novelet indicators at the first (cyan) and third instances (magenta) of the bird song.

The corresponding Novelet shown in Fig 45 demonstrates that a Novelet may still be discovered with non-trivial noise levels between two behavior instances.

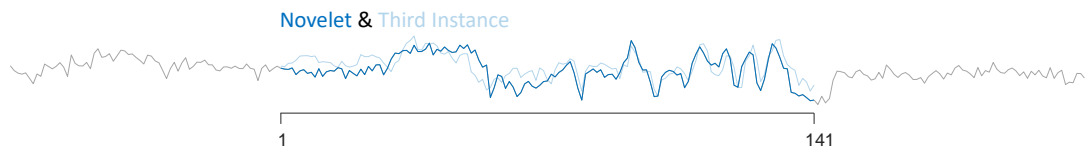


Fig 45 The first and third instances of the embedded bird song, along with context (gray).

The test results demonstrate high utility of Novelets in the area of wildlife monitoring. Of the 100 background song subtests, the true positive rates for discovering the Novelet within the first two, three, and four instances are 13%, 33%, and 44% respectively. These are remarkably high considering the default rate of a single subtest is just 0.63%. These true positive rates drastically increase when simulating foreground bird calls. True positive rates with discovery within the first two, three, and four instances are 79%, 89%, and 89% respectively. While there may be multiple Novelets discovered within a subtest, for ease of analysis, only the first Novelet of each subtest was evaluated for accuracy.

Further analysis of the results demonstrates a valuable property of Novelets. Because Novelet discovery is triggered by a novelty threshold, it is possible that no Novelets will

be returned. This means that a low true positive rate does not necessarily imply a high false positive rate. Fig 46 demonstrates this with a large false negative margin for both background and foreground bird song tests.

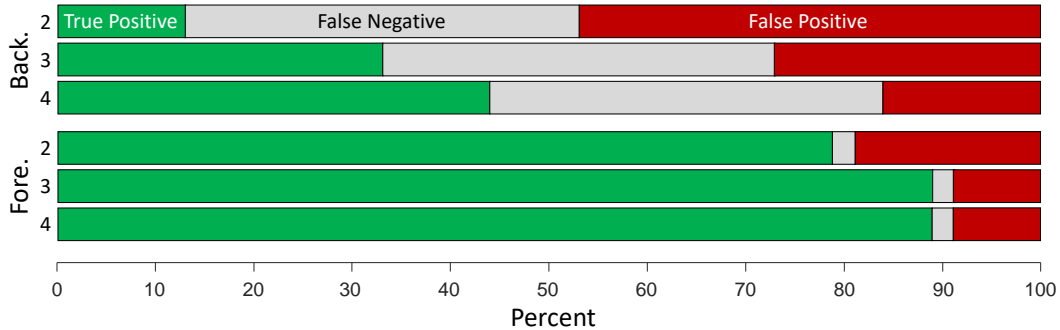


Fig 46 *top*) True positive, false negative, and false positive rates shown for 100 iterations of identifying bird songs in the background within the first two, three, and four song instances. *bottom*) The test repeated for bird songs in the foreground.

Even at the lowest true positive rate of 13% when discovering a background bird song within the first two instances, Novelets demonstrates high repeatability considering the difficulty of the task. The prior probability of positively identifying the first two instances in at least 13 of 100 iterations is 1.2×10^{-15} .

Finally, I consider scalability. Given an audio recording represented as MFCC at 60 Hz, a two second subsequence length, and twenty seconds of initialization, my algorithm will have an MTH of about twelve hours.

3.3.3 Industrial Process Monitoring: Bearings

Degradation of industrial components is often a gradual process where early prognosis can save large sums of money by reducing unplanned downtime. Typically, the sources of degradation such as wear, cracks, and corrosion are carefully modeled and tracked over a component's lifetime [85]. This requires detailed domain knowledge as the model

parameters are unique to the application and setting. My hypothesis is that *some* prognostic indications of wear may reveal themselves as newly emerging patterns. These novel patterns will not generally rise to the priority level of anomalies (which must be acted upon immediately) but can be reported to an engineer in say a daily or weekly report.

Case Western Reserve University's Bearing Data Center [86] provides datasets which capture component acceleration as bearings fail from seeded faults. The bearings are physically damaged in a variety of locations as a proxy for a range of failure sources. The dataset is split into four subsets: Normal and three types of failure configurations (labeled 48K, 12K, and FAN). Each configuration within a subset contains time series from multiple apparatus locations. In this comparison, only the drive-end (DE) time series is considered. The dataset contains times series captured at two different rates, 12 KHz and 48 KHz. Time series captured at 48 KHz are downsampled through mean pooling, but otherwise operated on identically.

In this domain, *classification* accuracy is typically used to demonstrate the utility of an algorithm. Clearly Novelets are not a classifier. By reframing the problem to the detection of new behaviors, then Novelets may reveal when operating conditions change. A member of the industrial plant health community, Chris Johnson, notes that when monitoring bearing health, "*Any changes in these signals that aren't caused by changes to operational conditions can indicate an issue with machine health* [87]." Considering this, we propose that an industrial process may be thought of as an evolving process, where small changes in the system result in new detectable time series behaviors.

In order to test this, I partition the first 5,000 samples of each time series as $\mathbf{T}^{(-)}$ in order to habituate to initial behaviors. With RPMs ranging between 1,720 to 1,797, the time series is analyzed with subsequences approximately the length of one rotation, which is about 400. As a manual training step, we set the novelty threshold to 0.25 so that zero Novelets are discovered within the Normal subset.

A failure configuration is marked as a positive detection when at least one Novelet is discovered. Within the fifty-two failure configurations of 48K class, forty-five contained at least one Novelet, which is an error rate of 13.5%. Similarly, 53 of 60 failure configurations of 12K class contained at least one Novelet, which is an error rate of 11.7%. Finally, in the FAN class, 40 of 45 configurations contained at least one Novelet, which is an error rate of 11.1%.

Marking a test configuration as a positive detection with only a single instance may be unfair, so in Fig 47 we study a histogram of Novelet counts for each configuration of the 12K class. When discounting the six configurations with zero detected Novelets, only six additional configurations have a count less than five.

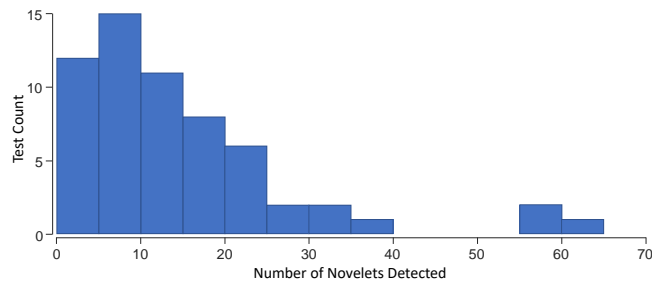


Fig 47 A histogram of Novelet counts for each test configuration. A majority of configurations have five or more detected Novelets.

The first two of ten discovered Novelets in a subtest of 48K are shown in Fig 48. Each Novelet is overlaid on its second instance. Recall, the subsequence length corresponds to about one rotation of the bearing. Note how surprisingly well conserved each Novelet is. This is an interesting test because the question of whether a failing process is conserved for a long period, or quickly decays into chaos is not immediately obvious. It is likely that this is a domain dependent question, but in the case of bearing failure, my results strongly support the hypothesis that new behaviors emerge as failure approaches.

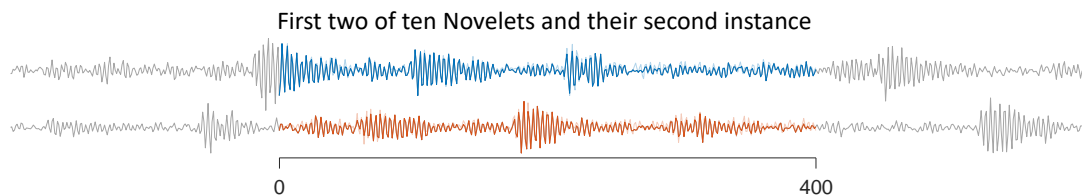


Fig 48 Two discovered Novelets shown with context (gray) and overlaid on their second instance. Most of the related work on this dataset (and it is *vast*, refer to [86] and references therein) typically attempts to explicitly classify between normal and a set of failure modes. While this is a worthy endeavor, it requires that there is robust labeled training data. This assumption greatly limits the utility of such techniques and makes it difficult to transfer the model from one motor to another. In contrast, the Novelet technique merely assumes that the user can point to a snippet of data thought to be healthy.

High speed industrial time series such as this tests the boundary of real-time actionability. The 12 KHz sampling rate here results in a computation time exceeding the arrival rate, though not by much. A 0.33 second buffer requires 1.26 seconds to process. As my current Emergence Profile is based on a vanilla Matrix Profile implementation, both future

software optimizations, and current GPU-based Matrix Profile algorithms such as SCAMP [4] offer a path to achieve the needed 3x speed increase in the very near future.

While I have framed the initial problem as a prognostic classification task, Novelets can also provide further insights from a data mining perspective. In the test explained above, all time series of NORMAL returned zero Novelets for a novelty threshold of 0.25. The threshold was manually chosen as a decision boundary to separate NORMAL from the other failing subsets. Though, by reducing the threshold, I should expect that for some value less than 0.25, but above 0.0, Novelets will emerge.

When examining the properties of Novelets discovered in the 3 horsepower NORMAL time series with a novelty threshold reduced to 0.2, I found a recurring structure between Novelets and their second occurrence. Upon a closer look, I found that for 30 of 31 Novelets discovered, the number of samples between the Novelet and its second occurrence was exactly 23,799.

This strong structure can be confirmed through Matrix Profile technique. One of the products of a Matrix Profile is the nearest neighbor index. I can create a new primitive by first subtracting an index i from its nearest neighbor index j . I then bin individual nearest neighbor offsets for all such index pairs, as shown in Fig 49.

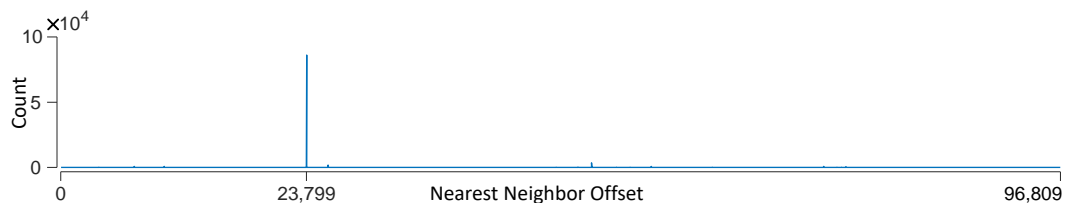


Fig 49 Visualizing the Matrix Profile nearest neighbor offsets reaffirms the observation that Novelets discovered a constant offset to their second occurrence.

While the time series was taken from the NORMAL subset, the constant offset may indicate a failing state within another part of the testing apparatus. An offset of 23,799 samples corresponds to 56.99 rotations at the reported 1,725 RPM. It occurred to me that motors often contain internal step-down gearing. It's conceivable that a damaged tooth has produced a signature every 57 rotations in a 3:19 gear ratio.

3.3.4 Industrial Process Monitoring: Buggy Machine

The experiments in the previous section are satisfying in that they show promising results in a well-studied dataset. However, I do not have the ability to intervene in this system to test the limits of my ideas. To allow for more “hands-on” experiments, I constructed a simple apparatus for measuring the vibration of a motor.

Here I wanted to model a problem that I heard about anecdotally. Industrial motors that power fans are typically housed in enclosures to prevent the ingress of birds, bats and insects. Surprisingly, it is the latter that is most likely to cause damage. While insects are small, they are much more likely to be able to circumvent a screen barrier, especially if it has a small puncture⁵. Moreover, while the mass of an individual insect is small, the swarming behavior of many insects means that a fan with a damaged screen might ingest hundreds of insects in a single night. If enough insects are ingested into a fan, they can cause damage to the system in multiple ways.

⁵ This story is reminiscent of, but is distinct from, the famous story of the discovery of the first computer “bug” (a moth) by Dr. Grace Hopper in 1945.

I hypothesized that insects hitting a computer fan may produce Novelets when measured with accelerometers. As a proxy for real insects, I have chosen to use three weight classes of cotton balls. The small, medium, and large balls weigh about 50 mg, 240 mg, and 660 mg, respectively.

Fig 50 summarizes my test, where I habituate on one minute of undisturbed fan motion, then drop each weight class six times from roughly three centimeters above the fan at twenty second intervals.

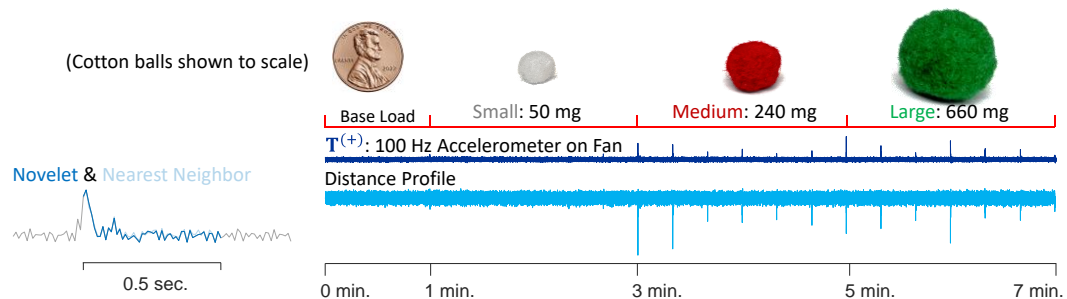


Fig 50 *left*) The first Novelet discovered, shown with context in gray. *right*) Using the Novelet as a query on the accelerometer data produces a distance profile which detects all instances of the six medium and six large ball drops, confirming that the Novelet detected was indeed caused by a collision.

A half-second Novelet discovers all the medium and large instances, but the small weight class is too light to register as a repeated acceleration shape. It is unlikely that a housefly (*Musca domestica*, 12 mg [88]) collision would register as a Novelet. However, these results suggest a larger insect collision *such as* a honeybee (*Bombus californicus*, 120 mg [89]) *would* likely register as a Novelet.

3.4 Model Comparison

3.4.1 Comparison to Segmentation

In some cases, Novelets may perform a function similar to semantic segmentation [90]. Though, as I have emphasized before, the essence of Novelets is different from segmentation and change detection. When a time series experiences a regime change, a Novelet may be discovered in the first few instances of a new behavior. Such a regime change may occur in patients experiencing onset of *Pulsus Paradoxus* where there is a physiological change in heart function. An existing example of this is found in [90], which I explore in Fig 51.

Thus far, I have downplayed the importance of the novelty threshold due to claimed stability over a range of threshold values. Here, I reinforce this with a visually intuitive technique for identifying segmentation points without having an intuition for choice of threshold value. By sweeping over threshold values and plotting locations of Novelet indices, I see regions with high Novelet stability.

The change point occurs at index 10,000 and I habituate on the first 1,000 time points. Each row in Fig 51.*bottom* represents Novelet indices for a given threshold value. It may be unintuitive to see Novelet index markers for higher thresholds which lack lower markers. Recall that Novelets are discovered in an online fashion based on local optima of the Emergence Profile, so the discovery of an early Novelet may affect the discovery of later Novelets.

One strategy for reasoning about Fig 51.*bottom* is to choose the index region with greatest frequency of discovered Novelets, which is 10,043. This is within just two heartbeats of the ground truth change point.

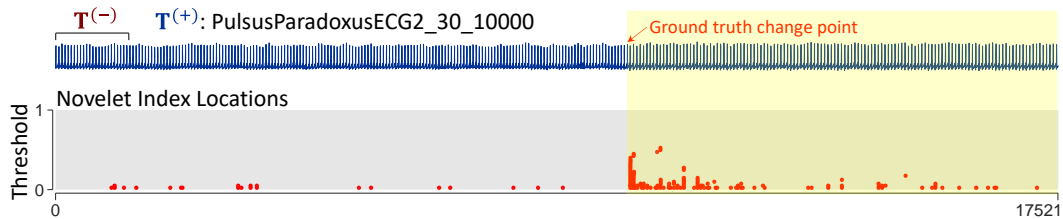


Fig 51 *top*) A time series of Pulsus Paradoxus with a visually non-trivial change point located at index 10,000. *bottom*) A threshold parameter sweep of Novelet indices. There is a stable Novelet index within two subsequence lengths of the ground truth.

This example is impressive in that the semantic change is impossible to see with the naked eye (this is true even if the data is seen on a larger plot, see [9]), and is only known due to out-of-band data available to the attending physician.

While Novelets merely duplicate the utility of SOTA segmentation algorithms in this example, one might imagine onset of a non-continuous and sporadic physiological behavior which might escape detection by segmentation, such as the PVC case described in Section 3.1.4.

3.5 Conclusions

I have introduced Novelets, a Matrix Profile based primitive that allows online discovery of emerging behaviors. I have shown the actionability of Novelets in real-time monitoring of medical telemetry, wildlife, and industrial processes. I have also demonstrated my algorithm's low sensitivity to the two user-set parameters: subsequence length and novelty threshold, providing a low learning curve for the operator and high stability of results. By defining my ideas explicitly in terms of the Matrix Profile, I can leverage future work by

the MP community. I have made all data and code available [9]. In future work I plan to investigate multidimensional Novelets in order to increase their actionability.

4. Conclusions

In this dissertation, I have introduced the Contrast Profile, a novel primitive that identifies contrasting behaviors given only weakly labeled time series. It has many downstream uses including exploratory data mining, classification, and anomaly detection. In addition, I have introduced another new primitive, Novelets, which identifies emerging repeated behaviors which has applications in wildlife monitoring, industrial process monitoring, and biosignal monitoring.

In Chapter 2 we discussed the Contrast Profile and evaluated its results in classification applications. When shape-based techniques are appropriate, it has the potential to decrease classification times by orders of magnitude compared to deep learning.

In addition, the first application of the Contrast Profile appeared in an automotive setting [91]. They used the Contrast Profile to identify distracted driving Platos with a downstream application in real-time behavior classification of nearby vehicles.

I discussed the limitations of the Contrast Profile. It operates on the assumption that there exist zero instances of the target behavior type in $\mathbf{T}^{(-)}$ time series which suppresses behaviors. If there is a single instance of the target behavior in $\mathbf{T}^{(-)}$, then it will not be discovered. Another limitation is that the Contrast Profile will always return a top peak, so it is important to either incorporate domain knowledge or visually validate the results.

In Chapter 3 I introduced Novelets, a primitive which allows for the online discovery of emerging behaviors in real-time. I have demonstrated its utility on a variety of real-world datasets that I have made publicly available.

I discussed the limitations of Novelets. Similar to the Contrast Profile, it operates under the assumption that zero instances of future emerging behaviors occur in the exclusion time series $\mathbf{T}^{(-)}$. Though if this is a concern, cold starting the algorithm with an empty $\mathbf{T}^{(-)}$ is possible and will only result in an initial burst of reported behaviors.

Both works are ripe with future work directions. Both works are defined with a single line definition based on Euclidean distance Matrix Profiles, though the definition is sufficiently general and would be compatible with a Matrix Profile generated using alternate distance measures. Shape-based techniques using Euclidean distance are brittle when the target behaviors contain a lot of warping, though feature-based techniques may be more robust to such distortions. To encourage the community to adopt and extend my ideas, I have made all code and supplemental material available [20][9].

Bibliography

- [1] C. M. Yeh *et al.*, “Matrix Profile I: All pairs similarity joins for time series: A unifying view that includes motifs, discords and shapelets,” in *2016 IEEE 16th International Conference on Data Mining (ICDM)*, Dec. 2016, pp. 1317–1322. doi: 10.1109/ICDM.2016.0179.
- [2] A. Mueen *et al.*, “The fastest similarity search algorithm for time series subsequences under Euclidean distance.” Accessed: Jan. 18, 2021. [Online]. Available: <https://www.cs.unm.edu/~mueen/FastestSimilaritySearch.html>
- [3] Y. Zhu, M. Imamura, D. Nikovski, and E. Keogh, “Introducing time series chains: a new primitive for time series data mining,” *Knowl Inf Syst*, vol. 60, no. 2, pp. 1135–1161, Aug. 2019, doi: 10.1007/s10115-018-1224-8.
- [4] Z. Zimmerman *et al.*, “Scaling Time Series Motif Discovery with GPUs: Breaking the Quintillion Pairwise Comparisons a Day Barrier,” p. 14.
- [5] H. A. Dau *et al.*, “Welcome to the UCR time series classification/clustering page.” Accessed: Jan. 17, 2021. [Online]. Available: https://www.cs.ucr.edu/~eamonn/time_series_data_2018/
- [6] B. Hu, Y. Chen, and E. Keogh, “Classification of streaming time series under more realistic assumptions,” *Data Min Knowl Disc*, vol. 30, no. 2, pp. 403–437, Mar. 2016, doi: 10.1007/s10618-015-0415-0.
- [7] A. C. Murillo, A. Abdoli, R. A. Blatchford, E. J. Keogh, and A. C. Gerry, “Parasitic mites alter chicken behaviour and negatively impact animal welfare,” *Scientific Reports*, vol. 10, no. 1, Art. no. 1, May 2020, doi: 10.1038/s41598-020-65021-0.
- [8] T. Nakamura, M. Imamura, R. Mercer, and E. Keogh, “MERLIN: Parameter-free discovery of arbitrary length anomalies in massive time series archives,” in *2020 IEEE International Conference on Data Mining (ICDM)*, Nov. 2020, pp. 1190–1195. doi: 10.1109/ICDM50108.2020.00147.
- [9] R. Mercer, “Novelets.” Accessed: Feb. 02, 2022. [Online]. Available: <https://sites.google.com/view/novelets>
- [10] P. Thornton, “Digoxin Uses, Dosage & Side Effects,” Drugs.com. Accessed: Mar. 08, 2022. [Online]. Available: <https://www.drugs.com/digoxin.html>
- [11] C. Wetzel, “Sparrows are singing a new song, in a rapid, unprecedented shift,” Animals. Accessed: Mar. 08, 2022. [Online]. Available: <https://www.nationalgeographic.com/animals/article/new-sparrow-birdsong-replaces-old-tune>
- [12] K. A. Otter, A. Mckenna, S. E. LaZerte, and S. M. Ramsay, “Continent-wide Shifts in Song Dialects of White-Throated Sparrows,” *Current Biology*, vol. 30, no. 16, pp. 3231–3235.e3, Aug. 2020, doi: 10.1016/j.cub.2020.05.084.
- [13] M. D. Beecher and S. E. Campbell, “The role of unshared songs in singing interactions between neighbouring song sparrows,” *Animal Behaviour*, vol. 70, no. 6, pp. 1297–1304, Dec. 2005, doi: 10.1016/j.anbehav.2005.03.008.
- [14] R. C. Berwick, K. Okanoya, G. J. L. Beckers, and J. J. Bolhuis, “Songs to syntax: the linguistics of birdsong,” *Trends in Cognitive Sciences*, vol. 15, no. 3, pp. 113–121, Mar. 2011, doi: 10.1016/j.tics.2011.01.002.
- [15] J. I. Benichov, S. E. Benezra, D. Vallentin, E. Globerson, M. A. Long, and O. Tchernichovski, “The Forebrain Song System Mediates Predictive Call Timing in Female and Male Zebra Finches,” *Current Biology*, vol. 26, no. 3, pp. 309–318, Feb. 2016, doi: 10.1016/j.cub.2015.12.037.
- [16] C. M. Yeh, Y. Zhu, H. A. Dau, A. Darvishzadeh, M. Noskov, and E. Keogh, “Online amnestic DTW to allow real-time golden batch monitoring,” in *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, Anchorage AK USA: ACM, Jul. 2019, pp. 2604–2612. doi: 10.1145/3292500.3330650.
- [17] R. Mercer and E. Keogh, “Matrix Profile XXV: Introducing Novelets: A Primitive that Allows Online Detection of Emerging Behaviors in Time Series,” presented at the 2022 IEEE International Conference on Data Mining (ICDM), IEEE Computer Society, Nov. 2022, pp. 338–347. doi: 10.1109/ICDM54844.2022.00044.
- [18] Y. Zhu *et al.*, “Exploiting a novel algorithm and GPUs to break the ten quadrillion pairwise comparisons barrier for time series motifs and joins,” *Knowl Inf Syst*, vol. 54, no. 1, pp. 203–236, Jan. 2018, doi: 10.1007/s10115-017-1138-x.

- [19] K. Beyer, J. Goldstein, R. Ramakrishnan, and U. Shaft, “When is ‘nearest neighbor’ meaningful?,” in *Database Theory — ICDT’99*, C. Beeri and P. Buneman, Eds., in Lecture Notes in Computer Science. Berlin, Heidelberg: Springer, 1999, pp. 217–235. doi: 10.1007/3-540-49257-7_15.
- [20] R. Mercer, “Contrast Profile.” Accessed: Jan. 05, 2021. [Online]. Available: <https://sites.google.com/view/contrastprofile>
- [21] D. S. Willett, J. George, N. S. Willett, L. L. Stelinski, and S. L. Lapointe, “Machine learning for characterization of insect vector feeding,” *PLoS Computational Biology*, vol. 12, no. 11, p. e1005158, Nov. 2016, doi: 10.1371/journal.pcbi.1005158.
- [22] M. A. Ladds, A. P. Thompson, D. J. Slip, D. P. Hocking, and R. G. Harcourt, “Seeing it all: Evaluating supervised machine learning methods for the classification of diverse otariid behaviours,” *PloS one*, vol. 11, no. 12, p. e0166898, Dec. 2016, doi: 10.1371/journal.pone.0166898.
- [23] *Galumphing: How seals move on land*, (Jun. 14, 2017). Accessed: Sep. 21, 2021. [Online Video]. Available: <https://www.youtube.com/watch?v=w36iwxDtTGs>
- [24] M. P. Beentjes, “Comparative terrestrial locomotion of the Hooker’s sea lion (*Phocarctos hookeri*) and the New Zealand fur seal (*Arctocephalus forsteri*): evolutionary and ecological implications,” *Zoological Journal of the Linnean Society*, vol. 98, no. 4, pp. 307–325, Apr. 1990, doi: 10.1111/j.1096-3642.1990.tb01204.x.
- [25] J. Lin and E. Keogh, “Group SAX: Extending the notion of contrast sets to time series and multimedia data,” in *Knowledge Discovery in Databases: PKDD 2006*, in Lecture Notes in Computer Science. Berlin, Heidelberg: Springer, 2006, pp. 284–296. doi: 10.1007/11871637_29.
- [26] O. Yildirim, U. B. Baloglu, R.-S. Tan, E. J. Ciaccio, and U. R. Acharya, “A new approach for arrhythmia classification using deep coded features and LSTM networks,” *Computer Methods and Programs in Biomedicine*, vol. 176, pp. 121–133, Jul. 2019, doi: 10.1016/j.cmpb.2019.05.004.
- [27] S. Zilberstein and S. Russell, “Approximate reasoning using anytime algorithms,” in *Imprecise and Approximate Computation*, S. Natarajan, Ed., in The Springer International Series in Engineering and Computer Science. , Boston, MA: Springer US, 1995, pp. 43–62. doi: 10.1007/978-0-585-26870-5_4.
- [28] T. G. Dietterich, R. H. Lathrop, and T. Lozano-Pérez, “Solving the multiple instance problem with axis-parallel rectangles,” *Artificial Intelligence*, vol. 89, no. 1, pp. 31–71, Jan. 1997, doi: 10.1016/S0004-3702(96)00034-3.
- [29] L. Ye and E. Keogh, “Time series shapelets: a new primitive for data mining,” in *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, in KDD ’09. New York, NY, USA: Association for Computing Machinery, Jun. 2009, pp. 947–956. doi: 10.1145/1557019.1557122.
- [30] L. Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone, *Classification and regression trees*. Boca Raton: Routledge, 2017. doi: 10.1201/9781315139470.
- [31] X. Guan, R. Raich, and W.-K. Wong, “Efficient multi-instance learning for activity recognition from time series data using an auto-regressive hidden Markov model,” in *Proceedings of The 33rd International Conference on Machine Learning*, PMLR, Jun. 2016, pp. 2330–2339. Accessed: Jan. 05, 2022. [Online]. Available: <https://proceedings.mlr.press/v48/guan16.html>
- [32] Pedestrian Counting System, “City of Melbourne - Pedestrian counting system,” Pedestrian Counting System. Accessed: Oct. 27, 2021. [Online]. Available: <http://www.pedestrian.melbourne.vic.gov.au/#date=28-10-2021&time=8>
- [33] Y. Bu, L. Chen, A. W.-C. Fu, and D. Liu, “Efficient anomaly monitoring over moving object trajectory streams,” in *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2009, pp. 159–168.
- [34] W. M. Kouadri, M. Ouziri, S. Benbernou, K. Echihabi, T. Palpanas, and I. B. Amor, “Quality of sentiment analysis tools: the reasons of inconsistency,” *Proceedings of the VLDB Endowment*, vol. 14, no. 4, pp. 668–681, 2020.
- [35] A. Abdoli, A. C. Murillo, A. C. Gerry, and E. J. Keogh, “Time Series classification: lessons learned in the (literal) field while studying chicken behavior,” in *2019 IEEE International Conference on Big Data (Big Data)*, Dec. 2019, pp. 5962–5964. doi: 10.1109/BigData47090.2019.9005596.

- [36] A. Abdoli, A. C. Murillo, C.-C. M. Yeh, A. C. Gerry, and E. J. Keogh, "Time series classification to improve poultry welfare," in *2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA)*, Dec. 2018, pp. 635–642. doi: 10.1109/ICMLA.2018.00102.
- [37] A. Abdoli *et al.*, "Fitbit for chickens? Time series data mining can increase the productivity of poultry farms," in *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, in KDD '20. New York, NY, USA: Association for Computing Machinery, Aug. 2020, pp. 3328–3336. doi: 10.1145/3394486.3403385.
- [38] A. L. Goldberger *et al.*, "PhysioBank, PhysioToolkit, and PhysioNet," *Circulation*, vol. 101, no. 23, pp. e215–e220, Jun. 2000, doi: 10.1161/01.CIR.101.23.e215.
- [39] N. Raghu and K. N. Manjunatha, *Arrhythmia detection using machine learning techniques*. 2019.
- [40] E. H. Field *et al.*, "Uniform California earthquake rupture forecast, version 3 (UCERF3)—The time-independent model," *Bulletin of the Seismological Society of America*, vol. 104, no. 3, pp. 1122–1180, Jun. 2014, doi: 10.1785/0120130164.
- [41] S. Wiemer and M. Wyss, "Minimum magnitude of completeness in earthquake catalogs: Examples from Alaska, the Western United States, and Japan," *Bulletin of the Seismological Society of America*, vol. 90, no. 4, pp. 859–869, Aug. 2000, doi: 10.1785/0119990114.
- [42] M. D. Petersen *et al.*, "2014 update of the United States national seismic hazard maps," p. 8, 2014.
- [43] A. Trnkoczy, "Understanding and parameter setting of STA/LTA trigger algorithm," p. 20, Sep. 1999.
- [44] R. Allen, "Automatic phase pickers: Their present use and future prospects," *Bulletin of the Seismological Society of America*, vol. 72, no. 6B, pp. S225–S242, Dec. 1982, doi: 10.1785/BSSA07206B0225.
- [45] B. K. Sharma, A. Kumar, and V. M. Murthy, "Evaluation of seismic events detection algorithms," *J Geol Soc India*, vol. 75, no. 3, pp. 533–538, Mar. 2010, doi: 10.1007/s12594-010-0042-8.
- [46] N. S. Senobari *et al.*, "Super-efficient cross-correlation (SEC-C): A fast matched filtering code suitable for desktop computers," *Seismological Research Letters*, vol. 90, no. 1, pp. 322–334, Nov. 2018, doi: 10.1785/0220180122.
- [47] Z. E. Ross, D. T. Trugman, E. Hauksson, and P. M. Shearer, "Searching for hidden earthquakes in Southern California," *Science*, May 2019, doi: 10.1126/science.aaw6888.
- [48] S. Rost and C. Thomas, "Array seismology: Methods and applications," *Reviews of Geophysics*, vol. 40, no. 3, pp. 2-1-2–27, 2002, doi: 10.1029/2000RG000100.
- [49] C. E. Yoon, O. O'Reilly, K. J. Bergen, and G. C. Beroza, "Earthquake detection through computationally efficient similarity search," *Science Advances*, Dec. 2015, doi: 10.1126/sciadv.1501057.
- [50] "Earthquake detection through computationally efficient similarity search." Accessed: Oct. 05, 2021. [Online]. Available: <https://www.science.org/doi/10.1126/sciadv.1501057>
- [51] K. J. Bergen, P. A. Johnson, M. V. de Hoop, and G. C. Beroza, "Machine learning for data-driven discovery in solid Earth geoscience," *Science*, vol. 363, no. 6433, p. eaau0323, Mar. 2019, doi: 10.1126/science.aau0323.
- [52] SCEDC, "Southern California Earthquake Data Center at Caltech." Accessed: Oct. 05, 2021. [Online]. Available: <https://scedc.caltech.edu/faq/index.html#reviewed>
- [53] K. Hutton, J. Woessner, and E. Hauksson, "Earthquake monitoring in southern California for seventy-seven years (1932–2008)," *Bulletin of the Seismological Society of America*, vol. 100, no. 2, pp. 423–446, Apr. 2010, doi: 10.1785/0120090130.
- [54] Y. Zhu *et al.*, "The Swiss army knife of time series data mining: ten useful things you can do with the matrix profile and ten lines of code," *Data Min Knowl Disc*, vol. 34, no. 4, pp. 949–979, Jul. 2020, doi: 10.1007/s10618-019-00668-6.
- [55] D. R. Shelly, G. C. Beroza, S. Ide, and S. Nakamura, "Low-frequency earthquakes in Shikoku, Japan, and their relationship to episodic tremor and slip," *Nature*, vol. 442, no. 7099, pp. 188–191, Jul. 2006, doi: 10.1038/nature04931.
- [56] NCEDC, "Northern California earthquake data center." UC Berkeley Seismological Laboratory, 2014. doi: 10.7932/NCEDC.

- [57] D. P. Schaff and F. Waldhauser, “Waveform cross-correlation-based differential travel-time measurements at the Northern California Seismic Network,” *Bulletin of the Seismological Society of America*, vol. 95, no. 6, pp. 2446–2461, Dec. 2005, doi: 10.1785/0120040221.
- [58] J.-R. Scholz *et al.*, “Detection, analysis, and removal of glitches from InSight’s seismic data From Mars,” *Earth and Space Science*, vol. 7, no. 11, p. e2020EA001317, 2020, doi: 10.1029/2020EA001317.
- [59] Z. Duputel, V. C. Tsai, L. Rivera, and H. Kanamori, “Using centroid time-delays to characterize source durations and identify earthquakes with unique characteristics,” *Earth and Planetary Science Letters*, vol. 374, pp. 92–100, Jul. 2013, doi: 10.1016/j.epsl.2013.05.024.
- [60] A. Mueen, E. Keogh, and N. Young, “Logical-shapelets: an expressive primitive for time series classification,” in *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, in KDD ’11. New York, NY, USA: Association for Computing Machinery, Aug. 2011, pp. 1154–1162. doi: 10.1145/2020408.2020587.
- [61] T. Rakthanmanon and E. Keogh, “Fast shapelets: A scalable algorithm for discovering time series shapelets,” in *Proceedings of the 2013 SIAM International Conference on Data Mining*, Society for Industrial and Applied Mathematics, May 2013, pp. 668–676. doi: 10.1137/1.9781611972832.74.
- [62] T. Rakthanmanon, “Fast shapelets - Supporting website.” Accessed: Sep. 28, 2021. [Online]. Available: <http://alumni.cs.ucr.edu/~rakhant/FastShapelet/>
- [63] MATLAB, “Sequence classification using deep learning.” Accessed: Jan. 21, 2021. [Online]. Available: <https://www.mathworks.com/help/deeplearning/ug/classify-sequence-data-using-lstm-networks.html>
- [64] S. Alaei, R. Mercer, K. Kamgar, and E. Keogh, “Time series motifs discovery under DTW allows more robust discovery of conserved structure,” *Data Min Knowl Disc*, vol. 35, no. 3, pp. 863–910, May 2021, doi: 10.1007/s10618-021-00740-0.
- [65] R. Mercer, S. Alaei, A. Abdoli, S. Singh, A. Murillo, and E. Keogh, “Matrix Profile XXIII: Contrast Profile: A Novel Time Series Primitive that Allows Real World Classification,” in *2021 IEEE International Conference on Data Mining (ICDM)*, Dec. 2021, pp. 1240–1245. doi: 10.1109/ICDM51629.2021.00151.
- [66] Y. Zhu *et al.*, “Matrix Profile II: Exploiting a Novel Algorithm and GPUs to Break the One Hundred Million Barrier for Time Series Motifs and Joins,” in *2016 IEEE 16th International Conference on Data Mining (ICDM)*, Dec. 2016, pp. 739–748. doi: 10.1109/ICDM.2016.0085.
- [67] A. Zhang, S. Song, J. Wang, and P. S. Yu, “Time series data cleaning: from anomaly detection to anomaly repairing,” *Proc. VLDB Endow.*, vol. 10, no. 10, pp. 1046–1057, Jun. 2017, doi: 10.14778/3115404.3115410.
- [68] Y. Lu, R. Wu, A. Mueen, M. A. Zuluaga, and E. Keogh, “Matrix Profile XXIV: Scaling Time Series Anomaly Detection to Trillions of Datapoints and Ultra-fast Arriving Data Streams,” in *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, in KDD ’22. New York, NY, USA: Association for Computing Machinery, Aug. 2022, pp. 1173–1182. doi: 10.1145/3534678.3539271.
- [69] A. Blázquez-García, A. Conde, U. Mori, and J. A. Lozano, “A Review on Outlier/Anomaly Detection in Time Series Data,” *ACM Comput. Surv.*, vol. 54, no. 3, p. 56:1-56:33, Apr. 2021, doi: 10.1145/3444690.
- [70] V. Chandola, A. Banerjee, and V. Kumar, “Anomaly detection: A survey,” *ACM Comput. Surv.*, vol. 41, no. 3, p. 15:1-15:58, Jul. 2009, doi: 10.1145/1541880.1541882.
- [71] T. Fu, “A review on time series data mining,” *Engineering Applications of Artificial Intelligence*, vol. 24, no. 1, pp. 164–181, Feb. 2011, doi: 10.1016/j.engappai.2010.09.007.
- [72] S. Aghabozorgi, A. Seyed Shirkhorshidi, and T. Ying Wah, “Time-series clustering – A decade review,” *Information Systems*, vol. 53, pp. 16–38, Oct. 2015, doi: 10.1016/j.is.2015.04.007.
- [73] E. Keogh and J. Lin, “Clustering of time-series subsequences is meaningless: implications for previous and future research,” *Knowl Inf Syst*, vol. 8, no. 2, pp. 154–177, Aug. 2005, doi: 10.1007/s10115-004-0172-7.
- [74] N. Begum and E. Keogh, “Rare time series motif discovery from unbounded streams,” *Proc. VLDB Endow.*, vol. 8, no. 2, pp. 149–160, Oct. 2014, doi: 10.14778/2735471.2735476.

- [75] F. Madrid, S. Imani, R. Mercer, Z. Zimmerman, N. Shakibay, and E. Keogh, "Matrix Profile XX: Finding and Visualizing Time Series Motifs of All Lengths using the Matrix Profile," in *2019 IEEE International Conference on Big Knowledge (ICBK)*, Nov. 2019, pp. 175–182. doi: 10.1109/ICBK.2019.00031.
- [76] G. K. Palshikar, "Simple Algorithms for Peak Detection in Time-Series," in *1st International Conference on Advanced Data Analysis, Business Analytics and Intelligence*, 2009.
- [77] B. N. Sumukha, R. C. Kumar, S. S. Bharadwaj, and K. George, "Online peak detection in photoplethysmogram signals using sequential learning algorithm," in *2017 International Joint Conference on Neural Networks (IJCNN)*, May 2017, pp. 1313–1320. doi: 10.1109/IJCNN.2017.7966004.
- [78] B. Kemp, A. H. Zwinderman, B. Tuk, H. A. C. Kamphuisen, and J. J. L. Obery, "Analysis of a sleep-dependent neuronal feedback loop: the slow-wave microcontinuity of the EEG," *IEEE Transactions on Biomedical Engineering*, vol. 47, no. 9, pp. 1185–1194, Sep. 2000, doi: 10.1109/10.867928.
- [79] R. W. Lawson, "Blinking and Sleep," *Nature*, vol. 165, no. 4185, Art. no. 4185, Jan. 1950, doi: 10.1038/165081b0.
- [80] *The White-throated Sparrow | Adorable Songster of the North*, (Jun. 04, 2021). Accessed: May 02, 2022. [Online Video]. Available: <https://www.youtube.com/watch?v=KsBj5nL0yUs>
- [81] S. Davis and P. Mermelstein, "Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 28, no. 4, pp. 357–366, Aug. 1980, doi: 10.1109/TASSP.1980.1163420.
- [82] *4K Forest Birdsong 2 - Birds Sing in the Woods - No Loop Realtime Birdsong - Relaxing Nature Video*, (Sep. 25, 2017). Accessed: May 02, 2022. [Online Video]. Available: <https://www.youtube.com/watch?v=XxP8kxUn5bc>
- [83] I. Cruickshank, *White-crowned Sparrow*. Accessed: May 05, 2022. [Online]. Available: <https://xeno-canto.org/251101>
- [84] D. Chakraborty, P. Mukker, P. Rajan, and A. D. Dileep, "Bird Call Identification Using Dynamic Kernel Based Support Vector Machines and Deep Neural Networks," in *2016 15th IEEE International Conference on Machine Learning and Applications (ICMLA)*, Dec. 2016, pp. 280–285. doi: 10.1109/ICMLA.2016.0053.
- [85] A. Muller, M.-C. Suhner, and B. Iung, "Formalisation of a new prognosis model for supporting proactive maintenance implementation on industrial system," *Reliability Engineering & System Safety*, vol. 93, no. 2, pp. 234–253, Feb. 2008, doi: 10.1016/j.res.2006.12.004.
- [86] D. Neupane and J. Seok, "Bearing Fault Detection and Diagnosis Using Case Western Reserve University Dataset With Deep Learning Approaches: A Review," *IEEE Access*, vol. 8, pp. 93155–93178, 2020, doi: 10.1109/ACCESS.2020.2990528.
- [87] E. E. Staff, "These Techniques Find Bearing Faults," *Efficient Plant*. Accessed: May 31, 2023. [Online]. Available: <https://www.efficientplantmag.com/2023/04/these-techniques-find-bearing-faults/>
- [88] "weight of musca domestica - Wolfram|Alpha." Accessed: May 10, 2022. [Online]. Available: <https://www.wolframalpha.com>
- [89] "weight of Bombus californicus - Wolfram|Alpha." Accessed: May 10, 2022. [Online]. Available: <https://www.wolframalpha.com>
- [90] S. Gharghabi, Y. Ding, C.-C. M. Yeh, K. Kamgar, L. Ulanova, and E. Keogh, "Matrix Profile VIII: Domain Agnostic Online Semantic Segmentation at Superhuman Performance Levels," in *2017 IEEE International Conference on Data Mining (ICDM)*, Nov. 2017, pp. 117–126. doi: 10.1109/ICDM.2017.21.
- [91] R. Mercer, S. Ucar, and E. Keogh, "Shape-Based Telemetry Approach for Distracted Driving Behavior Detection," in *2021 IEEE Conference on Standards for Communications and Networking (CSCN)*, Dec. 2021, pp. 118–123. doi: 10.1109/CSCN53733.2021.9686146.