Presentation for Assistant Professor position

Tanmoy Mondal

March 22, 2024

Curriculum Vitae

Education

Nov. 2012 - Dec. 2015
Ph.D in Computer Science
LIFAT, Poly-Tech Tours
Tours, France

2007 - 2009
Master of Tech. in Mecatronics & Robotics
Indian Institute of Engineering Science and Technology (IIEST)
Shibpur, West Bengal, India

2003 - 2007
Bachelor of Tech. in Information Tech.
West Bengal University of Technology
West Bengal, India

Professional Experiences

Before Ph.D | 4 Years 5 months

INDUSTRIES

- April 2012 – Oct. 2012 (8 months) — **Technical Lead**, iGATE (Capgemini Entreprises), Pune, India
- June 2011 – March 2012 (10 months) — **Imaging Scientist**, Oxyent-Medical Pvt. Ltd. Mumbai, India

LABS

- June 2010 – June 2011 (1 Year) — **Project Engineer**, Center for Development of Advanced Computing (CDAC), Pune, India
- Aug. 2008 – June 2010 (1 year 11 months) — **Researcher**, Central Scientific Instruments Org. (CSIR), Chandigarh, India

After Ph.D | 7 Years

INDUSTRY

- Jan 2023 – Present (1 Year 2 months) — **Research Scientist**, VOGO, Grenoble, France

LABORATORY

- June 2022 – Dec 2022 (6 months) — **Senior Post-Doc**, BII Unit, A*STAR, Singapore
- Oct. 2020 – Avril, 2022 (1 Year 6 months) — **Researcher**, IMT Atlantique, Brest, France
- Feb. 2019 – Mai 2020 (1 Year 4 months) — **Senior Research Engineer**, ZENITH Team, INRIA Montpellier, France
- May 2017 – Jan. 2019 (1 Year 9 months) — **Post-Doctorate**, L3i Team, University of La Rochelle, France
- July 2016 – Jan. 2017 (7 months) — **Research Engineer**, LIRIS Team, INSA Lyon, France

INDUSTRY

- Jan. 2016 – May 2016 (5 months) — **Startup Founder**, ArrayTree, France

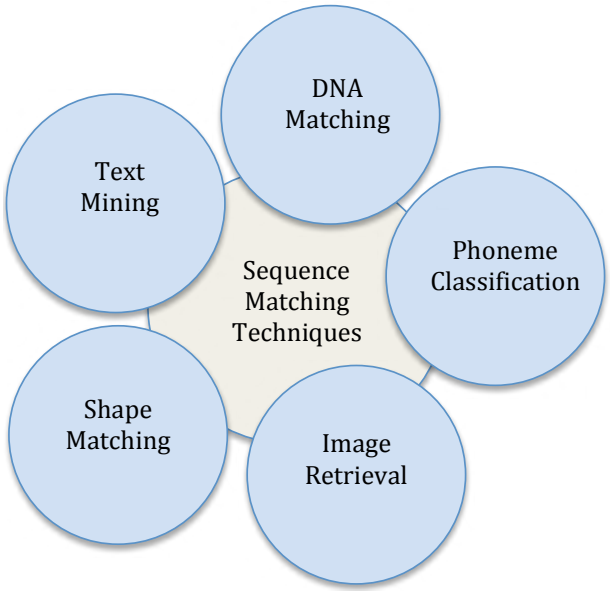
Summary of Presentation

- Ph.D Work : Comparative Study of *Dynamic Time Warping (DTW)*
- Ph.D Work : *Flexible Sequence Matching*
- Ph.D Work : *Exemplary Sequence Cardinality*
- What is Matrix Profile
- Post-Doc Work : AAMP algorithm
- Post-Doc Work : kNN Matrix Profile

Principal related work carried out during Ph.D

Time Series Matching to "Word Spotting"

- Modern digital devices accumulated huge time series data
- Important need to : querying and indexing
- Prevalent ares are:



- Many techniques extracts significant "features"
- One is image column based

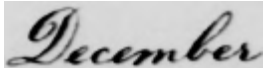


Fig: Query word

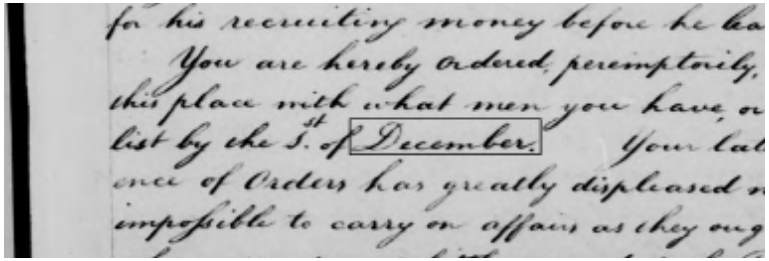


Fig: Sample scanned image

Fig: Sample key-word *Alexandria*

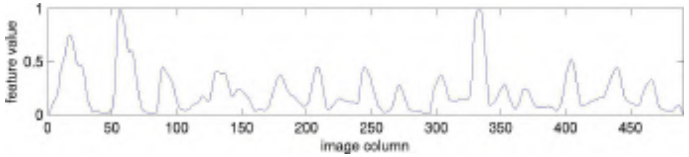


Fig: Normalized projection profile

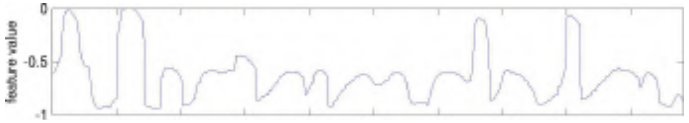


Fig: Normalized upper word profile [1]

[1] T.M.Rath and R.Manmatha,"Word spotting for historical documents", IJDAR,vol.9,no.2-4,pp.139-152,Aug.2006.

Architecture de «Word Spotting»

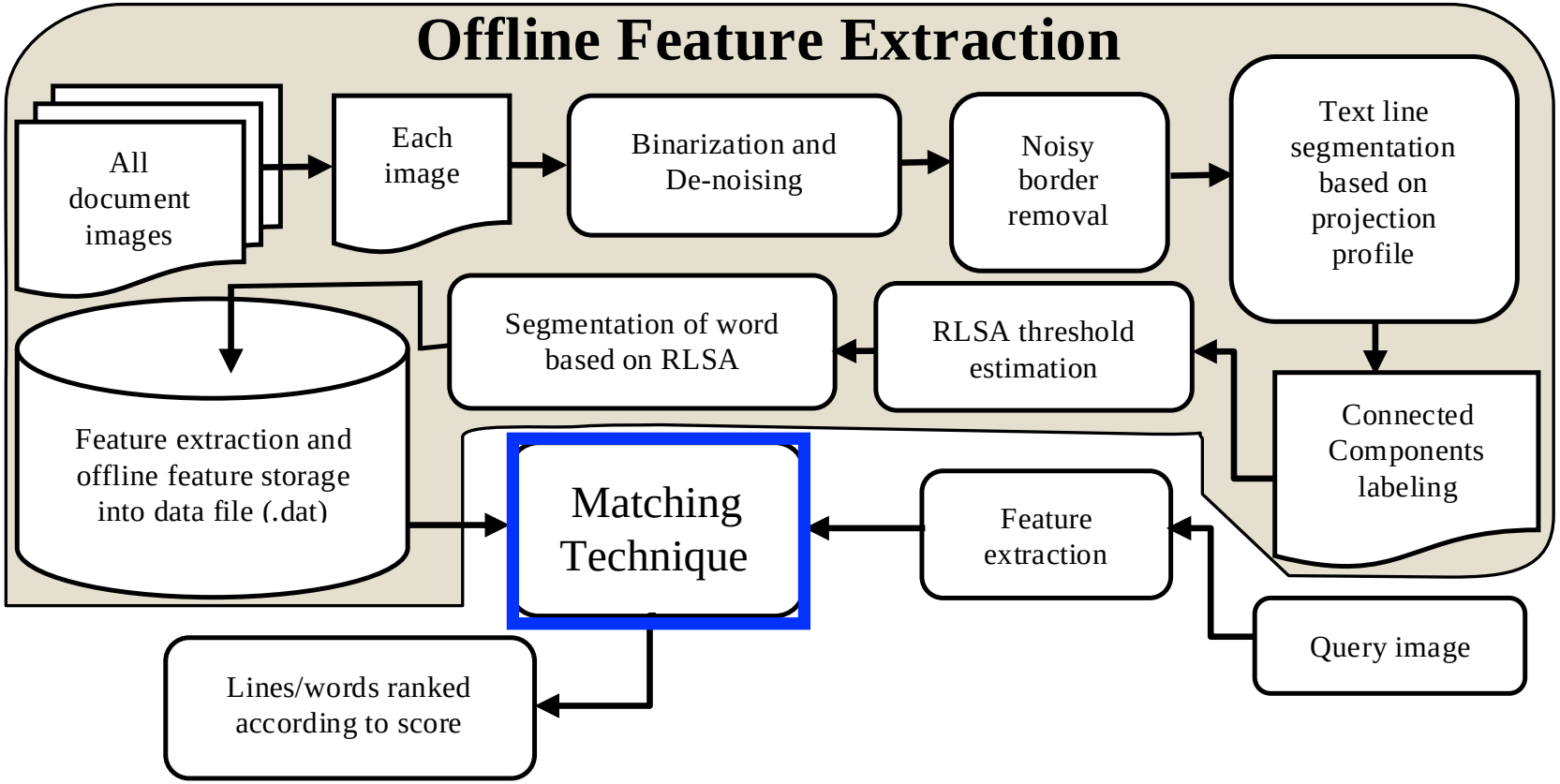


Fig: The block diagram of word spotting system

Feature Extraction

Column Based Feature Extraction:

N°	Feature set Description
F1.	Projection Profile
F2.	Background-to-ink transition
F3.	Upper Profile
F4.	Lower Profile
F5.	Distance between upper and lower Profile
F6.	Number of foreground pixels
F7.	Center of gravity (C.G.) of foreground pixels
F8.	Transition at C.G.

Features Based on «Slit Style HOG»

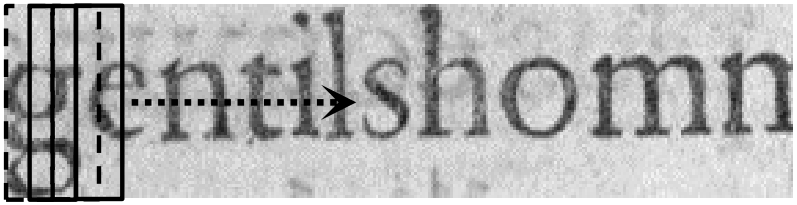


Fig : Feature extraction using slit style sliding window

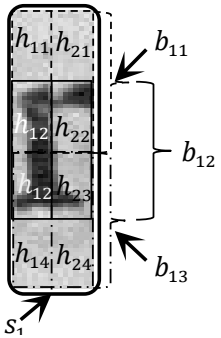


Fig : Block normalization technique

Datasets

Dataset Name	No of Query Images	No of Target Words	No of Pages	Properties
GW-15-Col	15	2340	10	Hand Written manuscript of George Washington(GW) (1755)
GW-90-Col	90	4860	20	
GW-15-HOG	15	675 (lines)	20	

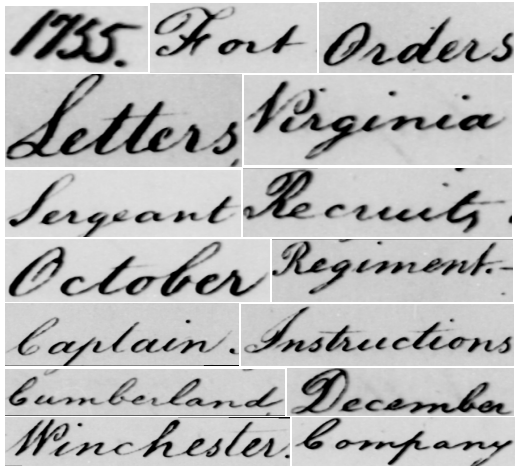


Fig: Some query images used for GW dataset

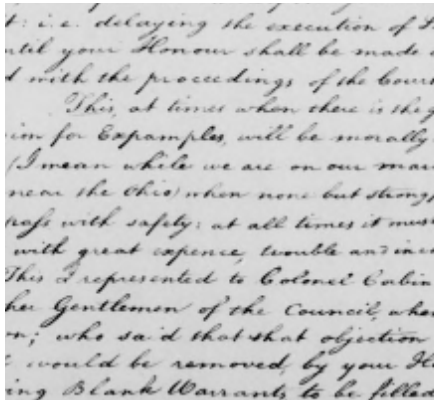


Fig: The sample page from GW dataset

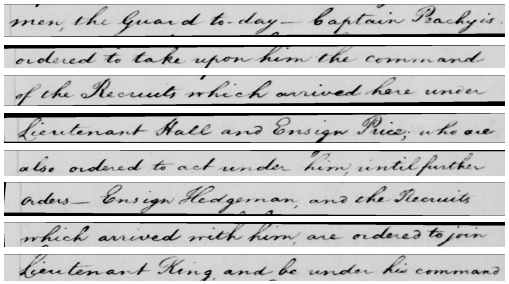


Fig: Some segmented text lines from GW dataset

Datasets

Dataset Name	No. of Query Images	No. Target Words	No of Pages	Properties
Japan-4-HOG	4	1575	92	Japanese Script

Fig: Some query images used for Japanese dataset

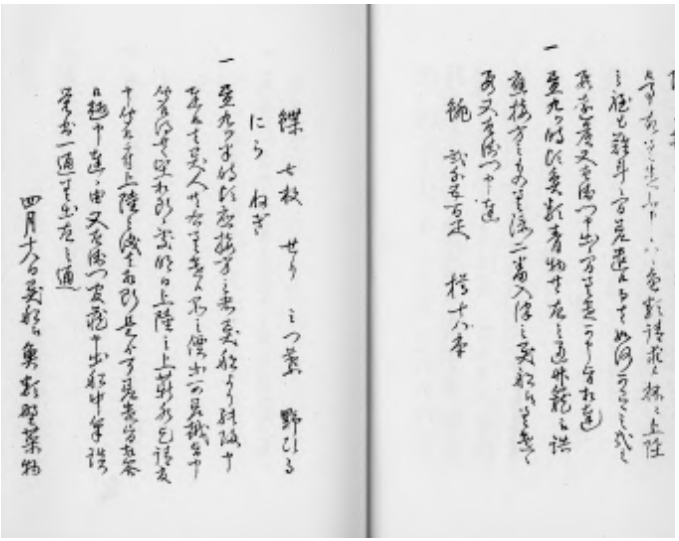


Fig: Sample page from the Japanese dataset

Comparative Study of Sequence Matching Techniques for "Word Spotting"

"Dynamic Time Warping"

- DTW measures similarity between two time series $X = x_1, x_2, x_3, \ldots, x_p$ and $Y = y_1, y_2, y_3, \ldots, y_q$
- We construct a $p \times q$ matrix where the (i^{th}, j^{th}) element of the matrix contains the distance: $\mathfrak{D}(x_i, y_j) = (x_i - y_j)^2)$
- The path cost matrix $\mathfrak{P}$ is calculated by using $\mathfrak{D}$
- The best warping path (W) is contiguous set of matrix elements, which defines an optimal mapping between X and Y.

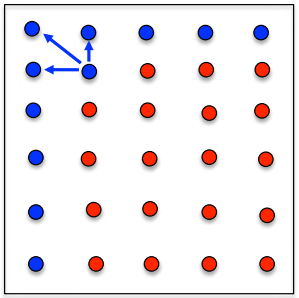


Fig: The distance matrix

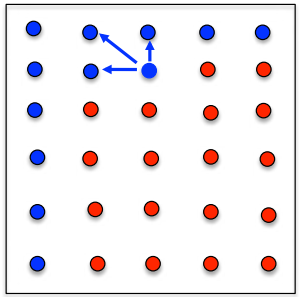


Fig: The "path cost" matrix

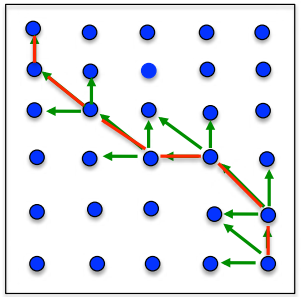


Fig: The "warping path" algorithm

$$\mathfrak{P}_{(1,1)} = \mathfrak{D}_{(1,1)}$$

$$\mathfrak{P}_{(i,0)} = \mathfrak{P}_{(i-1,0)} + \mathfrak{D}_{(i,0)} \quad 1<i\leq p$$

$$\mathfrak{P}_{(0,j)} = \mathfrak{P}_{(0,j-1)} + \mathfrak{D}_{(0,j)} \quad 1<j\leq q$$

$$\mathfrak{P}_{(i,j)} = \mathfrak{D}_{(i,j)} + min \begin{cases} \mathfrak{P}_{(i,j-1)} \\ \mathfrak{P}_{(i-1,j-1)} \\ \mathfrak{P}_{(i-1,j)} \end{cases}$$

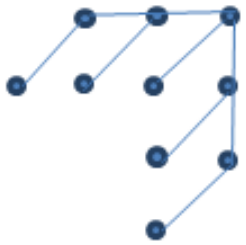
DTW with varying "step size" conditions

- In the case of classical DTW, the warping path can get stuck at some position
- To avoid such situation, other step size conditions of DP-path have been proposed in the literature.

$$\min\begin{pmatrix} \mathfrak{P}(t,\tau-1)+\mathfrak{D}(t,\tau) \\ \mathfrak{P}(t-1,\tau-1)+2\times\mathfrak{D}(t,\tau) \\ \mathfrak{P}(t-1,\tau)+\mathfrak{D}(t,\tau) \end{pmatrix}$$

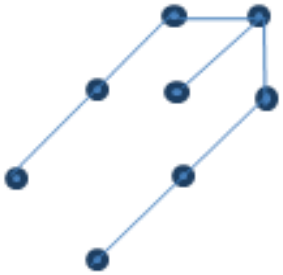
0

1/2



1

2



3

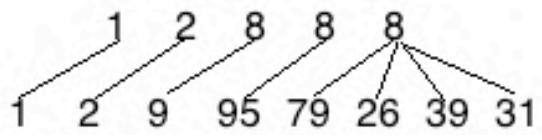
Comparative study on 6 databases

No.	Technique	Dataset-GW-15	Dataset-CESR-10	Dataset-GW-HOG	Dataset-Japanese-HOG	Dataset-GW-90	Dataset-Bentham
1.	0-Sym_1	0.3668	0.6449	×	×	0.1323	0.3286
2.	0-Sym_2 (classical DTW)	0.4576	0.8503	×	×	0.1573	0.4525
3.	1-Sym	0.3074	0.6922	×	×	0.1180	0.3533
4.	1-Asym	0.2188	0.220	×	×	0.0166	0.0458
5.	2-Sym	0.2038	0.5512	×	×	0.0513	0.1788
6.	2-Asym	0.1936	0.2029	×	×	0.0152	0.0417
7.	3-Sym	0.4324	0.8515	×	×	0.1702	0.4389
8.	0.5-Sym	0.2776	0.7168	×	×	0.1156	0.3901
9.	0.5-Asym	0.4642	0.8402	×	×	0.1663	0.4224
10.	SC-Band	0.5876	0.7961	×	×	×	×
11.	Itakura parallelogram	0.6017	0.8013	×	×	0.2226	0.5265
12.	DDTW	0.2180	0.5645	×	×	×	0.065
13.	Value Derivative DTW	0.2265	0.6510	×	×	×	×
14.	Weighted Hybrid DTW	0.3554	0.8621	×	×	0.146	0.358
15.	PDTW	0.3466	0.8929	×	×	0.157	0.432
16.	PDDTW	0.2563	0.9564	×	×	×	0.178
17.	WDTW	0.3309	0.5854	×	×	×	×
18.	WDDTW	0.2431	0.2319	×	×	×	×
19.	LDTW	0.5374	0.7577	×	×	0.229	0.499
20.	Sin transform	0.4213	0.6790	×	×	0.102	0.335
21.	Cos transform	0.2426	0.5653	×	×	×	×
22.	Hilbert transform	0.5380	0.8907	×	×	0.245	0.566
23.	SSDTW	0.3349	0.8409	0.645	0.7295	0.1181	0.3276
24.	DTW-CW	0.3603	0.8756	0.643	0.734	0.1358	0.3537
25.	MJ-DTW	0.3413	0.7779	×	×	0.1835	0.4370
26.	LCSS	0.0354	0.1496	×	×	×	×
27.	1DLCSS	0.0489	0.2006	×	×	×	×
28.	DDLCSS	0.0489	0.2024	×	×	×	
29.	OSB	0.2785	0.7982	0.3914	0.6089	×	×
30.	MVM	0.2026	0.5168	0.3495	0.3970	×	×
31.	CDP	0.3619	0.9176	0.7194	0.7473	0.1713	0.4354
32.	Fast-DTW	0.4348	0.7806	×	×	×	×

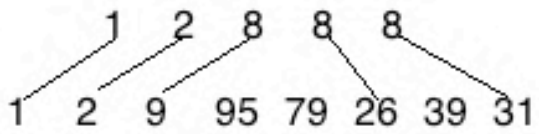
- **Tanmoy Mondal**, Nicolas Ragot, Jean-Yves Ramel, Umapada Pal, "Comparative Study of Conventional Time Series Matching Techniques for Word Spotting ", Pattern Recognition (PR) 73 :, 47-64, 2018.
- **Tanmoy Mondal**, Nicolas Ragot, Jean-Yves Ramel, Umapada Pal, Performance evaluation of DTW and its variants for word spotting in degraded documents, ICDAR 2015 : 1141-1145

Flexible Sequence Matching

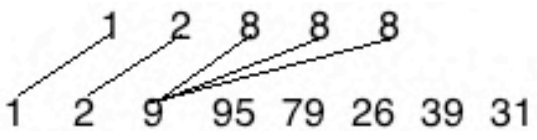
Flexible Sequence Matching



Partial Matching Capability of DTW

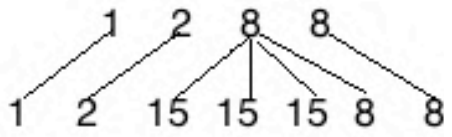


Partial Matching Capability of MVM

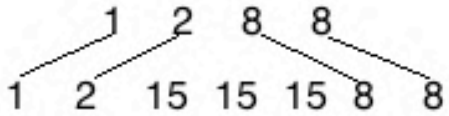


Partial Matching Capability of FSM

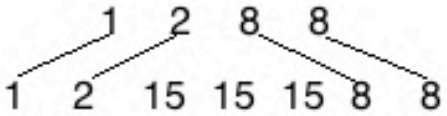
- DTW cannot ignore noisy elements
- MVM cannot have many-to-one and one-to-many matching. It is forced to find the same number of correspondences
- The FSM algorithm has overcome these two problems



One-to-Many matching by DTW



One-to-Many matching by MVM



One-to-Many matching by FSM

- DTW cannot give correct match
- MVM and FSM are able to give the right matches by skipping noisy elements

Formation de l'algorithme FSM

- FSM creates a relation $\mathbb{R}$ between x (query) and y (target), of lengths p and q: $x = (x_1, x_2, \ldots.., x_p)$ and $y = (y_1, y_2, \ldots.., y_q); p \leq q$
- It finds $y^{'}$ $(y^{'} \subset y)$ such that x best matches with $y^{'}$.
- The difference matrix $\mathfrak{D}$: $\mathfrak{D}_{i,j} = \sqrt{(y_j - x_i)^2}$; $1 \leq i \leq p; 1 \leq j \leq q$

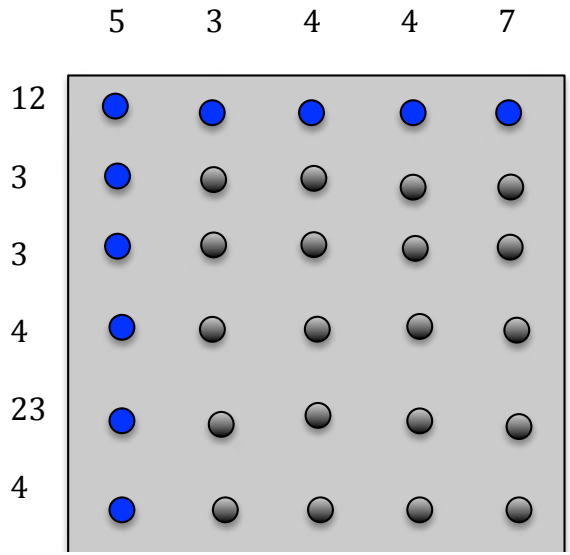


Fig: La matrice de distance

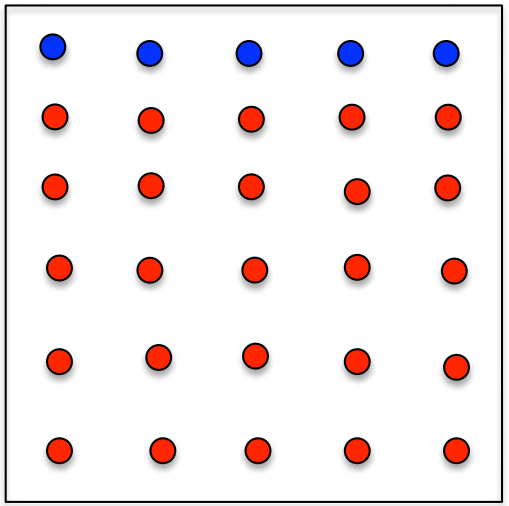


Fig: La DAG ($\mathbb{G}$) ou matrice «path cost» $\mathfrak{P}$

$$\mathfrak{P}(1, j) = \mathfrak{D}_{1,\, j} \text{ if } 1 \leq j \leq q$$

Formation de l'algorithme FSM ..

- Chaque cellule appartenant au i^{th} ligne est calculée en choisissant le nœuds parents à: – Ligne précédente $(i-1)$
  – Aux colonnes (k) à partir de $((i-1) - elasticity)$ Ã $((i-1) + elasticity)$

$$max[1, (i-1) - elasticity] \leq k \leq min[q, (i-1) + elasticity]$$

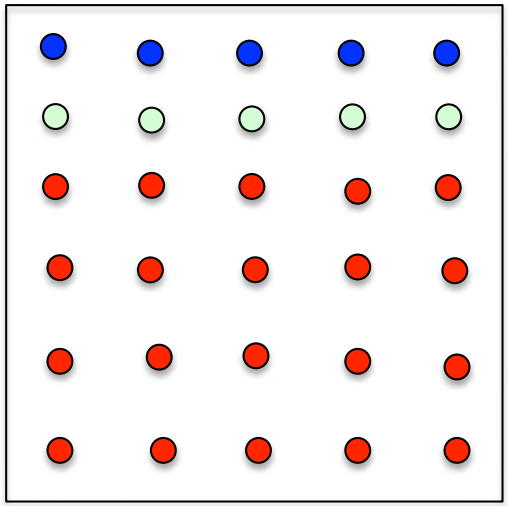


Fig: La matrice de «path cost»

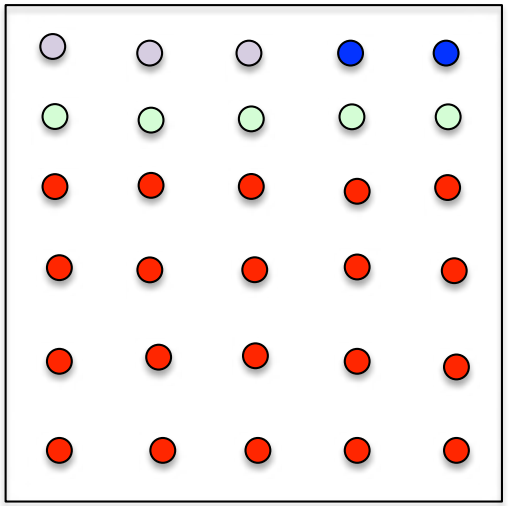


Fig: La matrice de «path cost»

Formation de l'algorithme FSM ..

- Par rapport à chacun de ces nœuds parents, les nœuds enfants ne peuvent appartenir qu' à :
  – Ligne Suivante
  – Colonne suivante $(k+1)$ jusqu'à $(k+1) + elasticity - |k-(i-1)|^{th}$

$2 \leq i \leq p$

$$k+1 \leq j \leq min(q, (k+1) + elasticity - max(0, \{k-(i-1)\}))$$
$$\{\mathfrak{P}(i-1,k) + \mathfrak{D}_{i,j} + (\mathfrak{S} \times (j-(k+1)))\}$$

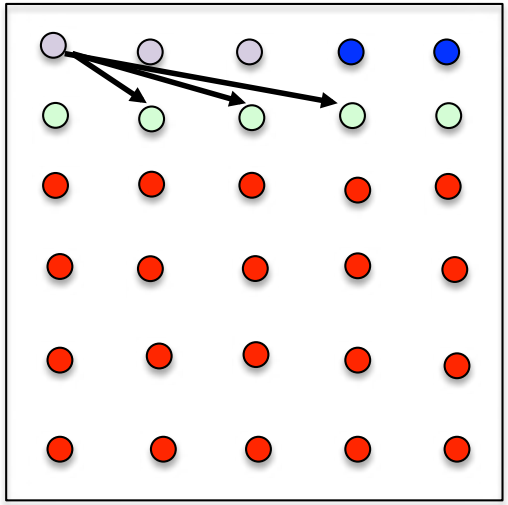


Fig: La matrice de «path cost»

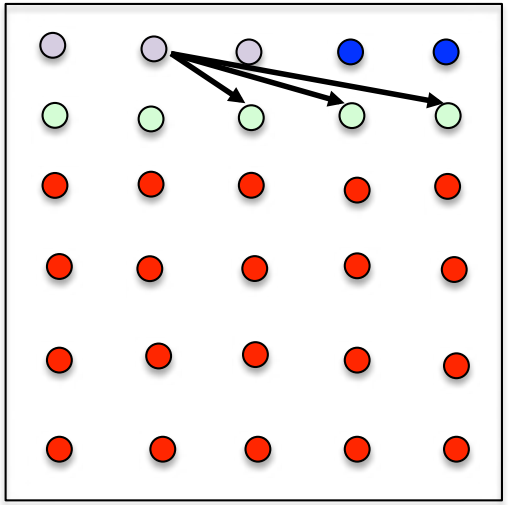


Fig: La matrice de «path cost»

Formation of the FSM algorithm ..

- Other possible links are:
  – «many-to-one» : link just below $(i-1,j)$ à (i,j) : $\{\mathfrak{P}(i-1,j)+\mathfrak{C}+\mathfrak{D}_{i,j}\}$
  – «one-to-many» : link just at left $(i,j-1)$ à (i,j) : $\{\mathfrak{P}(i,j-1)+\mathfrak{C}+\mathfrak{D}_{i,j}\}$
- A small penalty ($\mathfrak{C} = mean(\mathfrak{M}_b^{merged})$) is introduced to limit the numbers of many-to-one and one-to-many matching

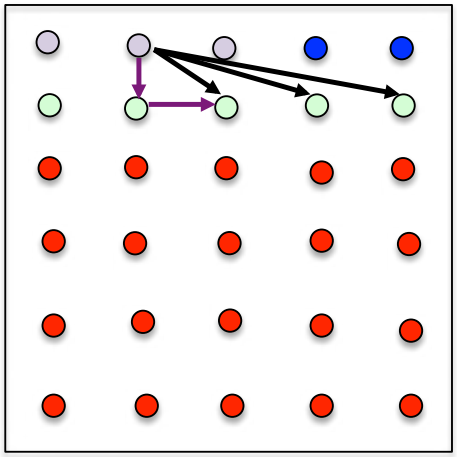


Fig: The path cost matrix

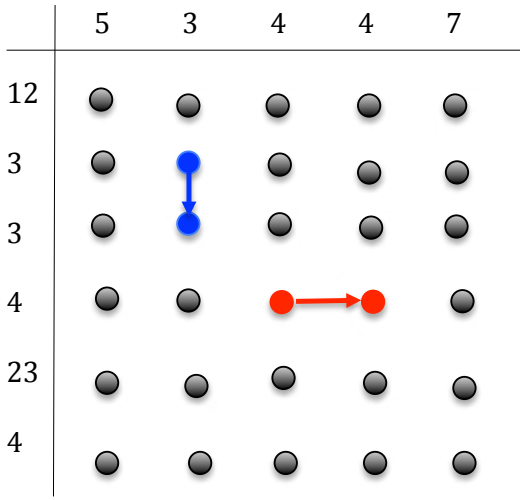


Fig: The illustration of one-to-many (blue) and many-to-one (red) matching on toy examples

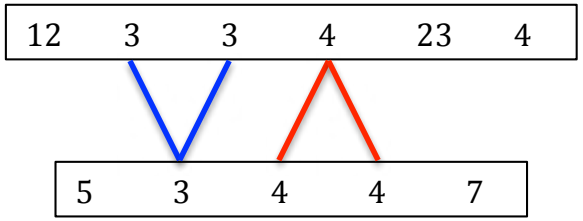


Fig: Corresponding matching elements of the query and target vectors

Some Visual Results

- FSM has outperformed other sequence matching techniques
- The noise or outliers (derivatives also) skipping capability helps FSM
- The special characteristics of FSM is helpful in other domains e.g. finance, video retrieval, shape matching etc.

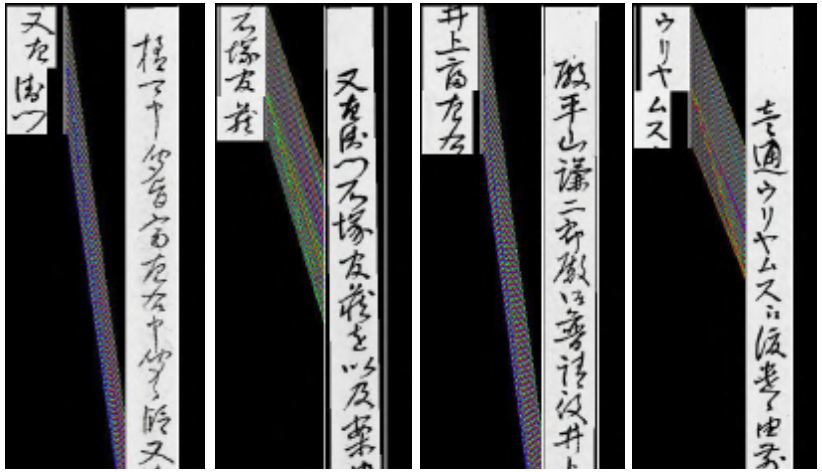


Fig: Four separate query words their matching

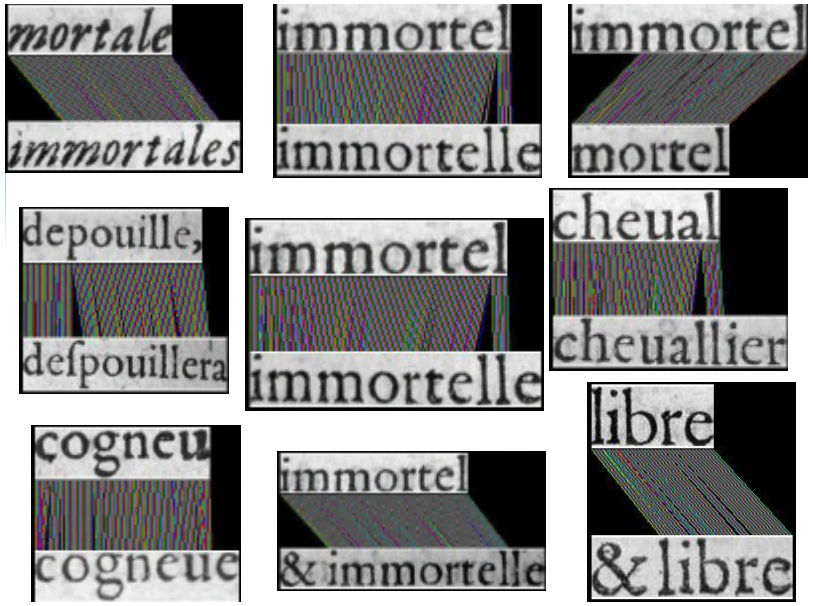


Fig: Example of matching by FSM

- **Tanmoy Mondal**, Nicolas Ragot, Jean-Yves Ramel, Umapada Pal, "Flexible Sequence Matching Technique: An Effective Learning-Free Approach for word-spotting", Pattern Recognition 60: 596-612 (2016)
- **Tanmoy Mondal**, Nicolas Ragot, Jean-Yves Ramel, Umapada Pal, Flexible Sequence Matching Technique : Application to Word Spotting in Degraded Documents, ICFHR 2014 : 210-215, 2013

Exemplary Sequence Cardinality

Introduction

- ESC has all the qualities as FSM
- ESC can skip noisy elements from query
- Choosing query become more easier
- The proposed system, is more robust to:
  - Degradation noise
  - Word derivatives
  - Improper segmentation issues
- For example, in French, the word ***cheval*** *(horse)* can have derivatives like *"chevallier"*,*"chevalerie"*, *"chevalier"*.

ESC : Mathematical Formulation

- To make the 1st element of the query skip-able, one null element (0) is is added at the beginning of both the query and target sequence (contrary to FSM)
- Modified dissimilarity matrix $(\mathfrak{Q})$ is created
- Another matrix $(\mathscr{M}_{i,j})$ is utilized to follow the indexes
- The immediate child of a skip-able node are connected with parents of their immediate parents (Grand parents).

$$\mathfrak{Q}_{i,j} = \begin{cases} \widehat{\mathfrak{D}}_{1,j} & 1 \leq j \leq q+1 \\ skipCost & \text{if } skipCost < \widehat{\mathfrak{D}}_{i,j} \\ \widehat{\mathfrak{D}}_{i,j} & \text{Sinon} \end{cases}$$

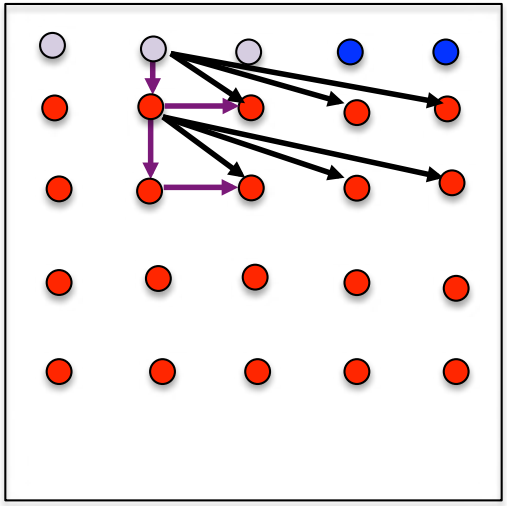


Fig: The "path cost" matrix

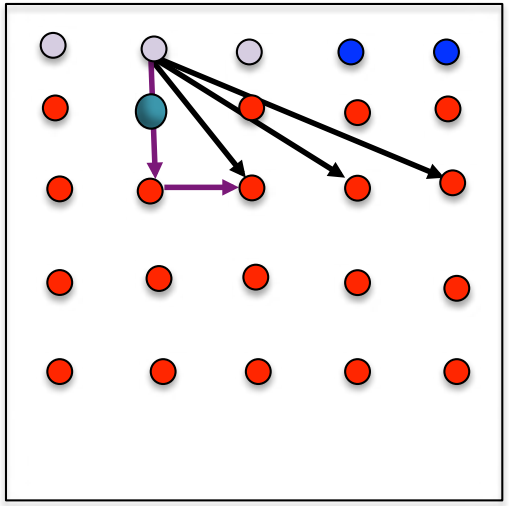


Fig: The "path cost" matrix

ESC: Formulation Mathématique

- Le chemin le plus court entre chaque paire de nœuds accessibles dans $\mathbb{G}$ peut être trouvé à partir de la matrice «path cost» $\mathscr{P}_{i,j}$.

$$\mathscr{P}_{1,j} = \mathfrak{Q}_{1,j} \quad \text{if } 1 \leq j \leq q+1$$

$$\mathscr{P}_{i,j} = \min \begin{pmatrix} \{\mathscr{P}_{i-1,k} + \mathfrak{Q}_{i,j} + \\ \quad (skipCost \times (j-(k+1)))\} \\ \{\mathscr{P}_{i,j-1} + \mathfrak{C} + \mathfrak{Q}_{i,j}\} \\ \{\mathscr{P}_{i-1,j} + \mathfrak{C} + \mathfrak{Q}_{i,j}\} \end{pmatrix} \text{ if } \mathfrak{L}$$

$$\mathfrak{L} : \begin{pmatrix} \{2 \leq i \leq q+1\} \\ \{(i-1) - elasticity \leq k \leq (i-1) + elasticity\} \\ \{k+1 \leq j \leq (k+1) + elasticity - |k-(i-1)|\} \end{pmatrix}$$

ESC: Mathematical Formulation

- The back tracking process would start from the cell at the last row and at the jth column of the path cost matrix $\mathscr{P}$, where $p + 1 \leq j \leq q + 1$
- ESC has the same complexity as FSM i.e. $\Theta((2.|q - p|^2).p)$

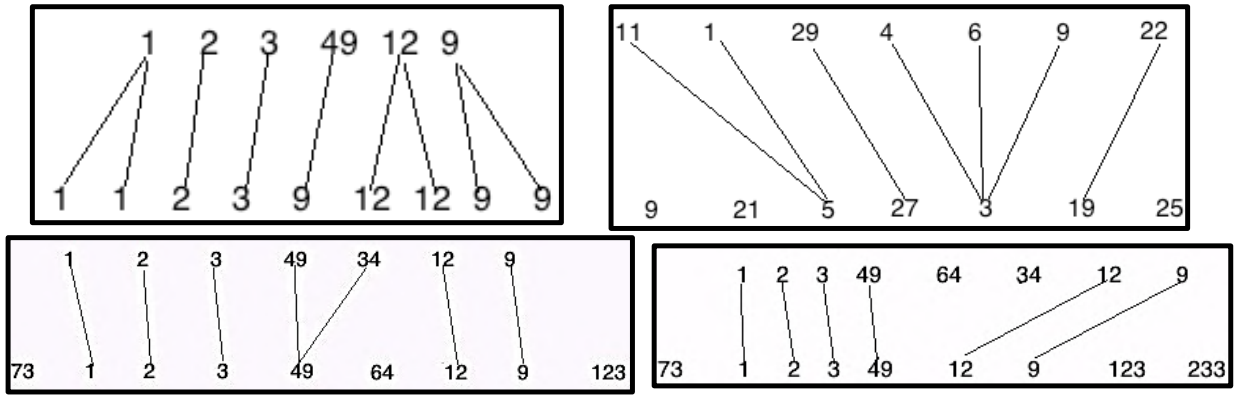


Fig: Matching ability of ESC on toy examples

Fig: Matching ability of ESC on some artificial images

Similarity Search in Time Series

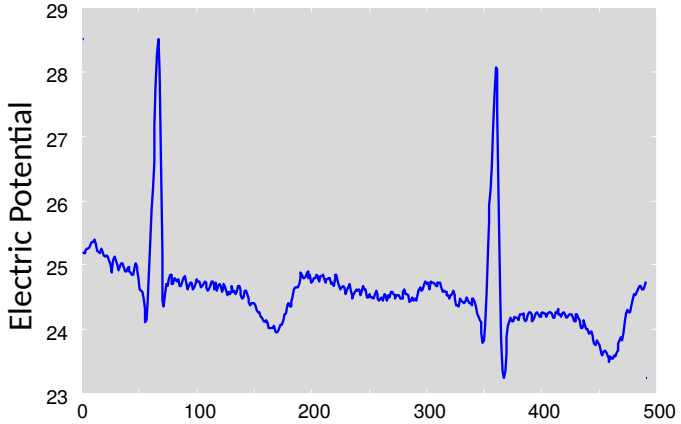
Similarity Search : From Distance Profile to Matrix Profile

What are Time Series? 1 of 2

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

More than most types of data, time series lend themselves to *visual* inspection and intuitions...

25.350
25.350
25.400
25.400
25.325
25.225
25.200
25.175
..
24.625
24.675
24.675
24.675



For example, looking at the numbers in this blue vector tells us nothing.

But after *plotting* the data, we can recognize a heartbeat, and possibly even diagnose this person's disease.

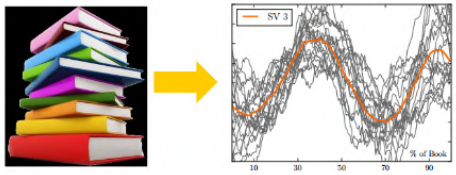
* Slide credit: Eamonn Keogh; `https://www.cs.ucr.edu/~eamonn/Matrix_Profile_Tutorial_Part2.pdf`

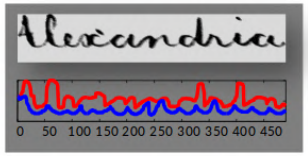
Similarity Search : From Distance Profile to Matrix Profile

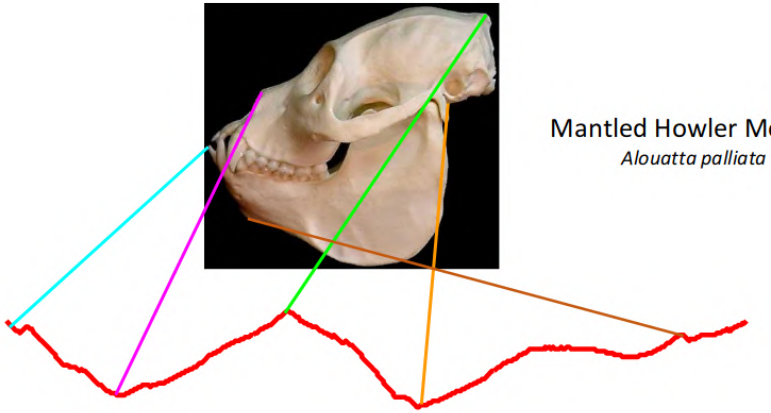
What are Time Series? 2 of 2

As an aside… (not the main point for today)

Many types of data that are not *true* time series can be fruitfully transformed into time series, including DNA, speech, textures, core samples, ASCII text, historical handwriting, novels and even *shapes*.







Mantled Howler Monkey
Alouatta palliata

* Slide credit: Eamonn Keogh; `https://www.cs.ucr.edu/~eamonn/Matrix_Profile_Tutorial_Part2.pdf`

Similarity Search : From Distance Profile to Matrix Profile

- A similarity measure compares two time series and produces a number representing their similarity
  - A distance measure is the opposite of similarity measure
- Lockstep Measures
  - Euclidean Distance
  - Correlation Coefficient
  - Cosine Similarity
- Elastic Measures
  - Dynamic Time Warping
  - Edit Distance
  - Longest Common Sub-sequence

* Slide credit: Eamonn Keogh; `https://www.cs.ucr.edu/~eamonn/Matrix_Profile_Tutorial_Part2.pdf`

Similarity Search : From Distance Profile to Matrix Profile

Euclidean Distance Metric

Given two time series

$\mathbf{x} = x_1 \ldots x_n$

and

$\mathbf{y} = y_1 \ldots y_n$

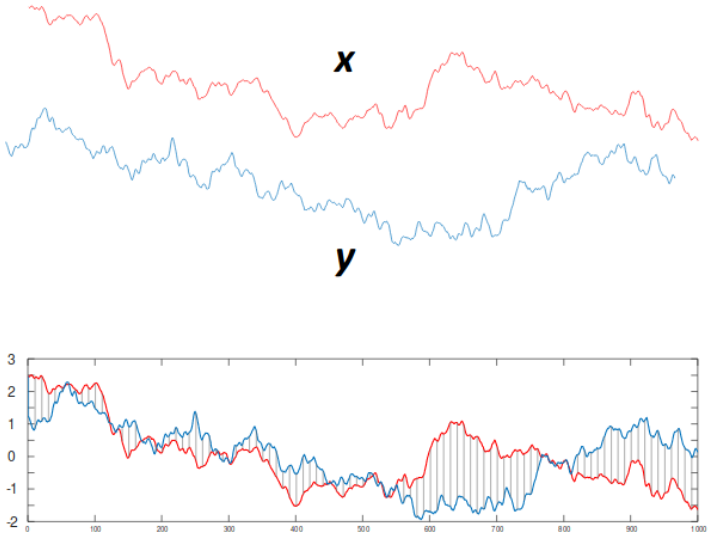
their z-Normalized Euclidean distance is defined as:

$$\hat{x}_i = \frac{x_i - \mu_x}{\sigma_x} \qquad \hat{y}_i = \frac{y_i - \mu_y}{\sigma_y}$$

```
function y = zNorm(x)
y = (x-mean(x))/std(x,1);
```

$$d(x,y) = \sqrt{\sum_{i=1}^{n} (\hat{x}_i - \hat{y}_i)^2}$$

```
function d = EuclideanDistance(x,y)
d = sqrt(sum((x-y).^2));
```



* Slide credit: Eamonn Keogh; https://www.cs.ucr.edu/~eamonn/Matrix_Profile_Tutorial_Part2.pdf

Similarity Search : From Distance Profile to Matrix Profile

Pearson's Correlation Coefficient

- Given two time series $\boldsymbol{x}$ and $\boldsymbol{y}$ of length m.
- Correlation Coefficient:

$$corr(\boldsymbol{x}, \boldsymbol{y}) = \frac{(E(x) - \mu_x)(E(y) - \mu_y)}{\sigma_x \sigma_y} = \frac{\sum_{i=1}^{m} x_i y_i - m\mu_x\mu_y}{m\sigma_x\sigma_y}$$

  - Where $\mu_x = \frac{\sum_{i=1}^{m} x_i}{m}$ and $\sigma_x^2 = \frac{\sum_{i=1}^{m} x_i^2}{m} - \mu_x^2$
- Sufficient Statistics:

$$\sum_{i=1}^{m} x_i y_i \quad \sum_{i=1}^{m} x_i \quad \sum_{i=1}^{m} y_i \quad \sum_{i=1}^{m} x_i^2 \quad \sum_{i=1}^{m} y_i^2$$

The sufficient statistics can be calculated in one linear scan. Given the sufficient statistics, correlation coefficient is a constant operation. Note the use of the dot product, which is the key component of many lockstep measures.

* Slide credit: Eamonn Keogh; `https://www.cs.ucr.edu/~eamonn/Matrix_Profile_Tutorial_Part2.pdf`

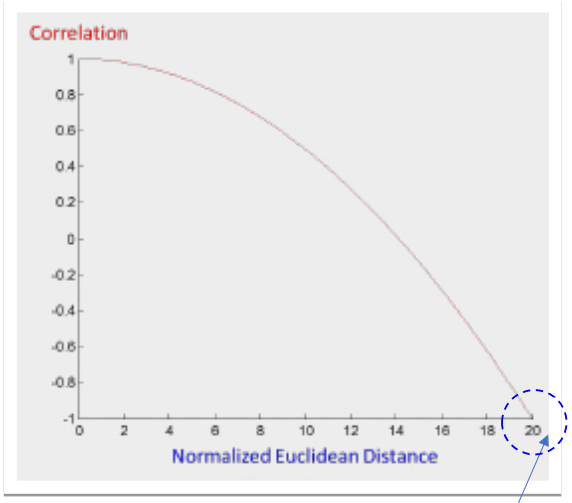
Similarity Search : From Distance Profile to Matrix Profile

Relationship with Euclidean Distance

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

- Maximizing correlation coefficient can be achieved by **minimizing normalized Euclidean distance and vice versa**
- Correlation coefficient is **bounded between -1 and 1**, while z-normalized Euclidean distance is **bounded between zero and a positive number** dependent on m

m = Length of Sub-Sequence



20 for m = 100

* Slide credit: Eamonn Keogh; `https://www.cs.ucr.edu/~eamonn/Matrix_Profile_Tutorial_Part2.pdf`

Similarity Search : From Distance Profile to Matrix Profile

Working Formula

$$d(\hat{\boldsymbol{x}}, \hat{\boldsymbol{y}}) = \sqrt{2m\left(1 - \frac{\sum_{i=1}^{m} x_i y_i - m\mu_x\mu_y}{m\sigma_x\sigma_y}\right)}$$

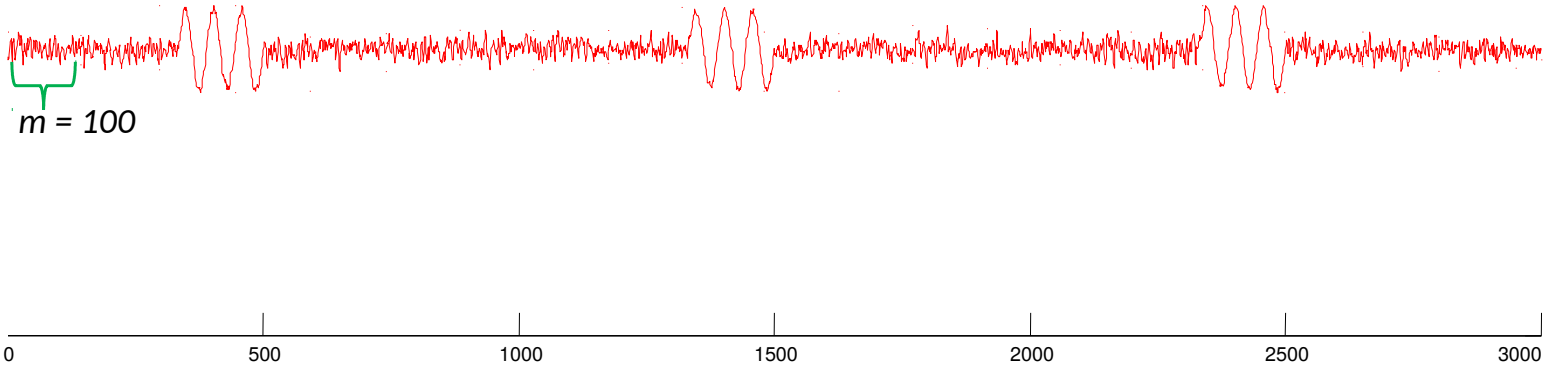
- We will use the above z-Normalized Euclidean distance as the similarity measure for the rest of the presentation

* Slide credit: Eamonn Keogh; `https://www.cs.ucr.edu/~eamonn/Matrix_Profile_Tutorial_Part2.pdf`

Similarity Search : From Distance Profile to Matrix Profile

What is Matrix Profile ?

– Intuition behind the Matrix Profile: Assume we have a time series T, lets start with a synthetic one...
– we are only interested in small *local* sub-sequences, of this length, m



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

* Slide credit: Eamonn Keogh; `https://www.cs.ucr.edu/~eamonn/Matrix_Profile_Tutorial_Part2.pdf`

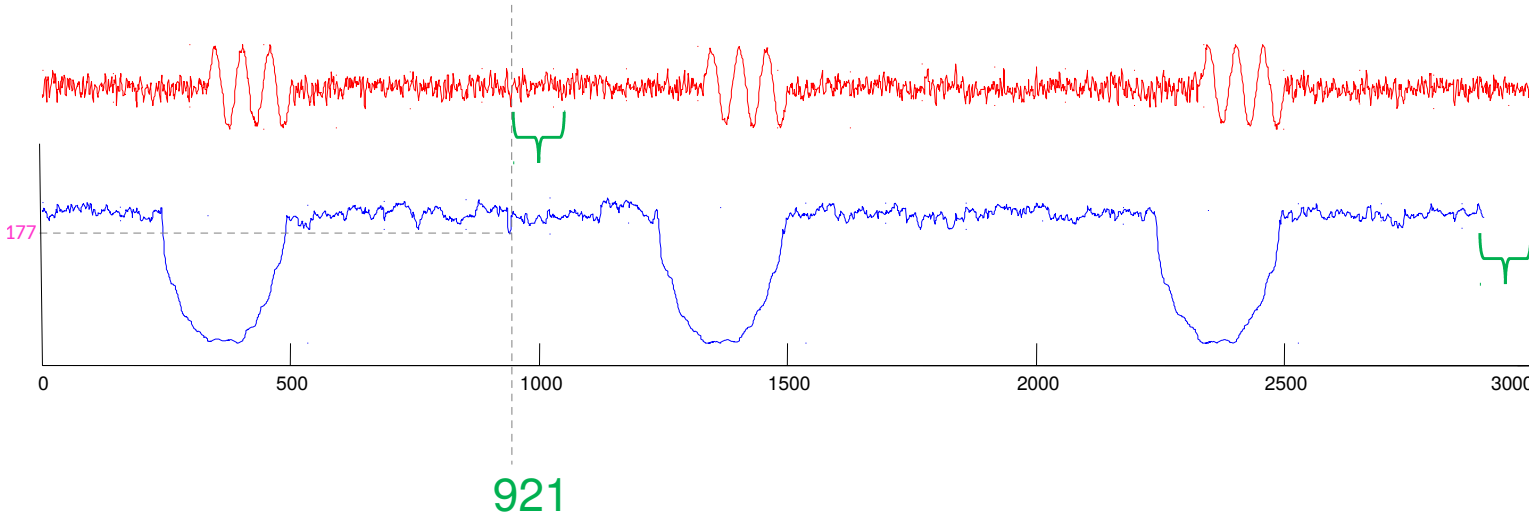
Similarity Search : From Distance Profile to Matrix Profile

What is Matrix Profile ?

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

The matrix profile of a time series T, records at the ith location the distance of its nearest neighbor sub-sequence under z-normalized Euclidean Distance

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

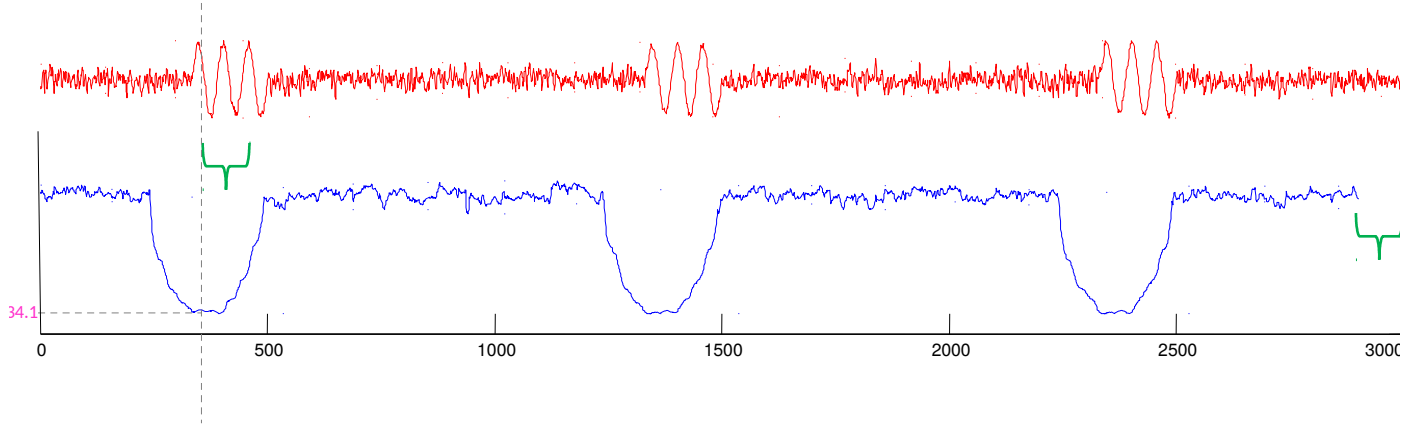


* Slide credit: Eamonn Keogh; `https://www.cs.ucr.edu/~eamonn/Matrix_Profile_Tutorial_Part2.pdf`

Similarity Search : From Distance Profile to Matrix Profile

What is Matrix Profile ?

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



The Matrix Profile is always shown in blue.
The real time series data, is generally shown in red.

* Slide credit: Eamonn Keogh; `https://www.cs.ucr.edu/~eamonn/Matrix_Profile_Tutorial_Part2.pdf`

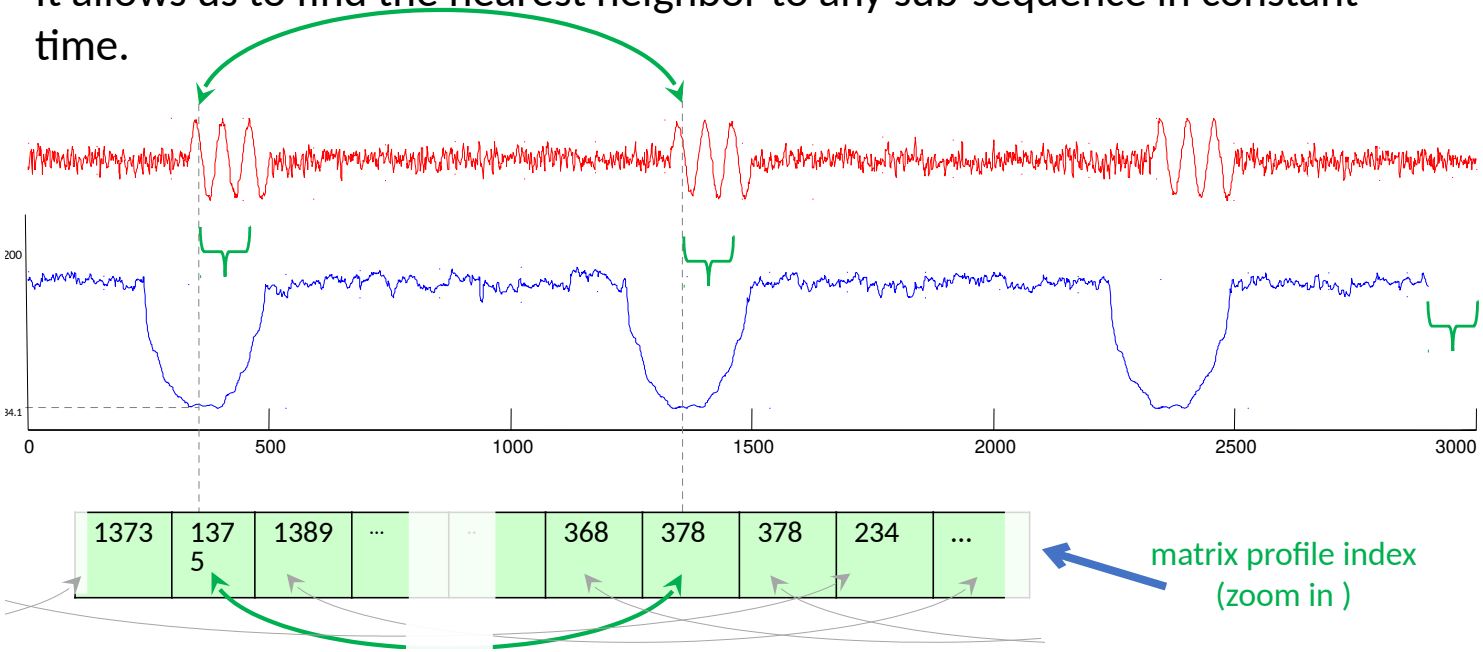
Similarity Search : From Distance Profile to Matrix Profile

What is Matrix Profile ?

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

The MPI contains integers that are used as pointers

It allows us to find the nearest neighbor to any sub-sequence in constant time.



* Slide credit: Eamonn Keogh; `https://www.cs.ucr.edu/~eamonn/Matrix_Profile_Tutorial_Part2.pdf`

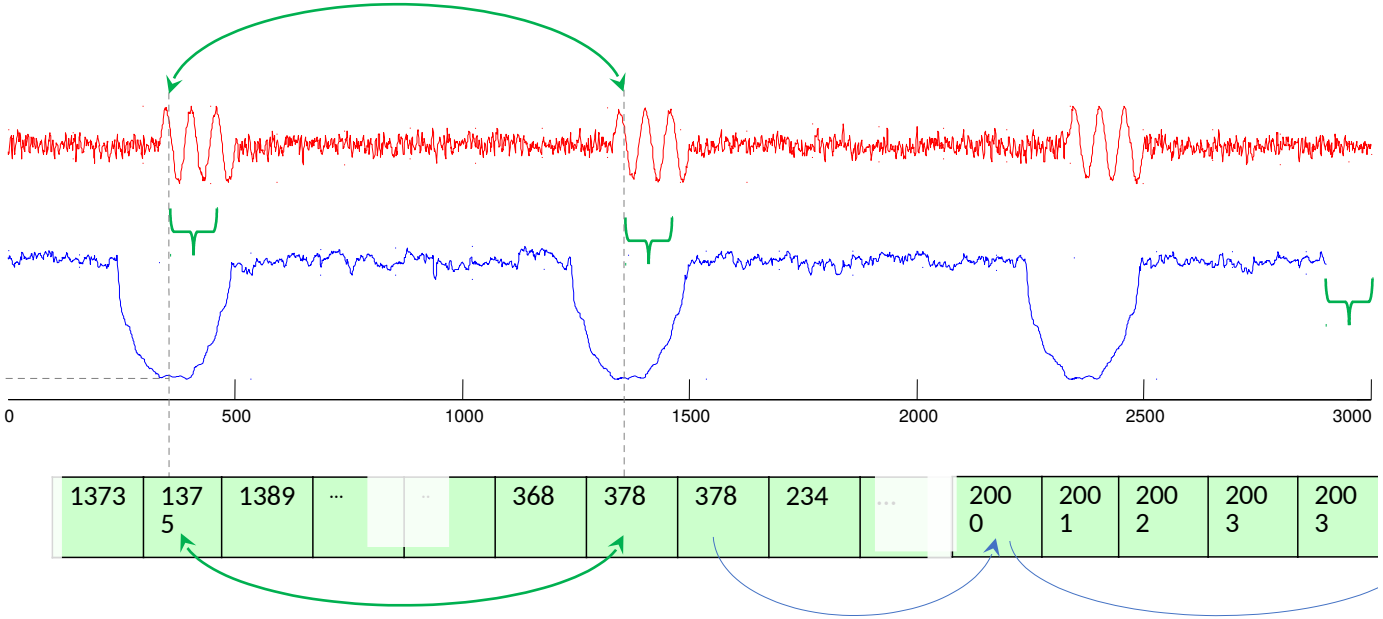
Similarity Search : From Distance Profile to Matrix Profile

What is Matrix Profile ?

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

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

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



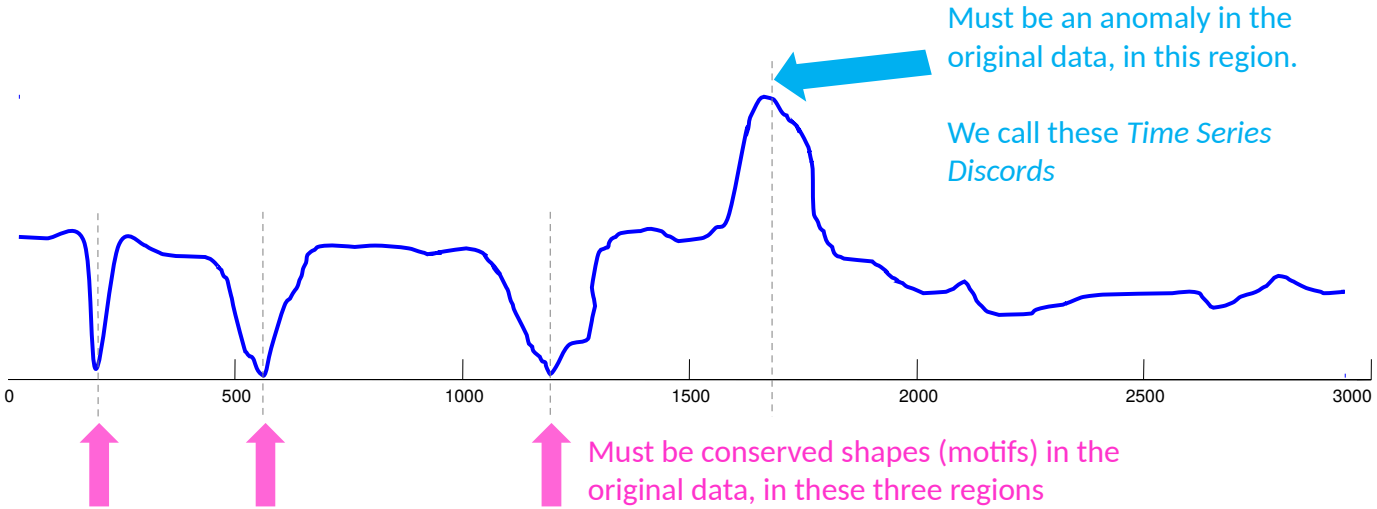
* Slide credit: Eamonn Keogh; `https://www.cs.ucr.edu/~eamonn/Matrix_Profile_Tutorial_Part2.pdf`

Similarity Search : From Distance Profile to Matrix Profile

How to "read" a Matrix Profile ?

Where you see relatively low values, you know that the sub-sequence in the original time series must have (at least one) relatively similar sub-sequence elsewhere in the data (**such regions are "motifs" or reoccurring patterns**)

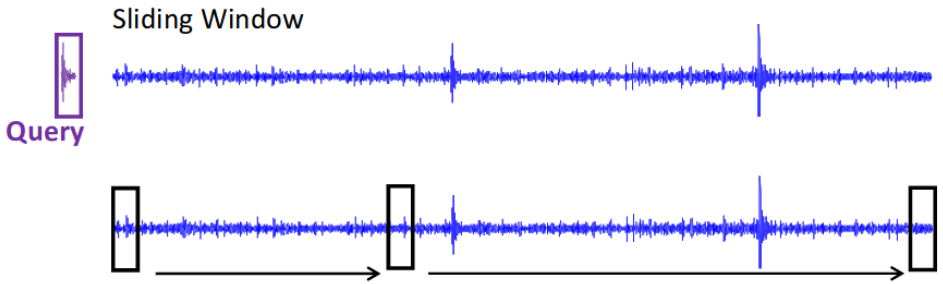
Where you see relatively high values, you know that the sub-sequence in the original time series must be unique in its shape (**such areas are "discords" or anomalies**).



* Slide credit: Eamonn Keogh; `https://www.cs.ucr.edu/~eamonn/Matrix_Profile_Tutorial_Part2.pdf`

Similarity Search : From Distance Profile to Matrix Profile

Distance Profile



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

d_1	d_2	...	d_{n-m+1}

$\longrightarrow$ D

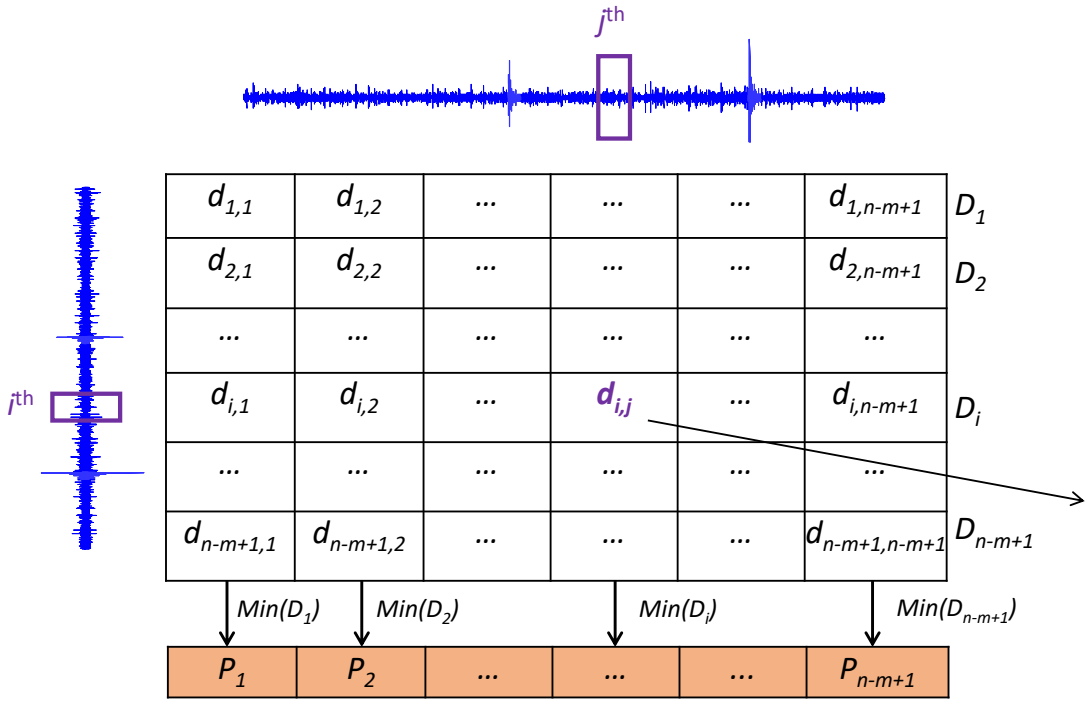
d_i is the distance between the ith subsequence and the query.

Recall, n is the length of the blue time series
and m is the length of the query

* Slide credit: Eamonn Keogh; `https://www.cs.ucr.edu/~eamonn/Matrix_Profile_Tutorial_Part2.pdf`

Similarity Search : From Distance Profile to Matrix Profile

Matrix Profile from Distance Profiles



The distance matrix is symmetric
The diagonal is zero
Cells close to the diagonal are very small

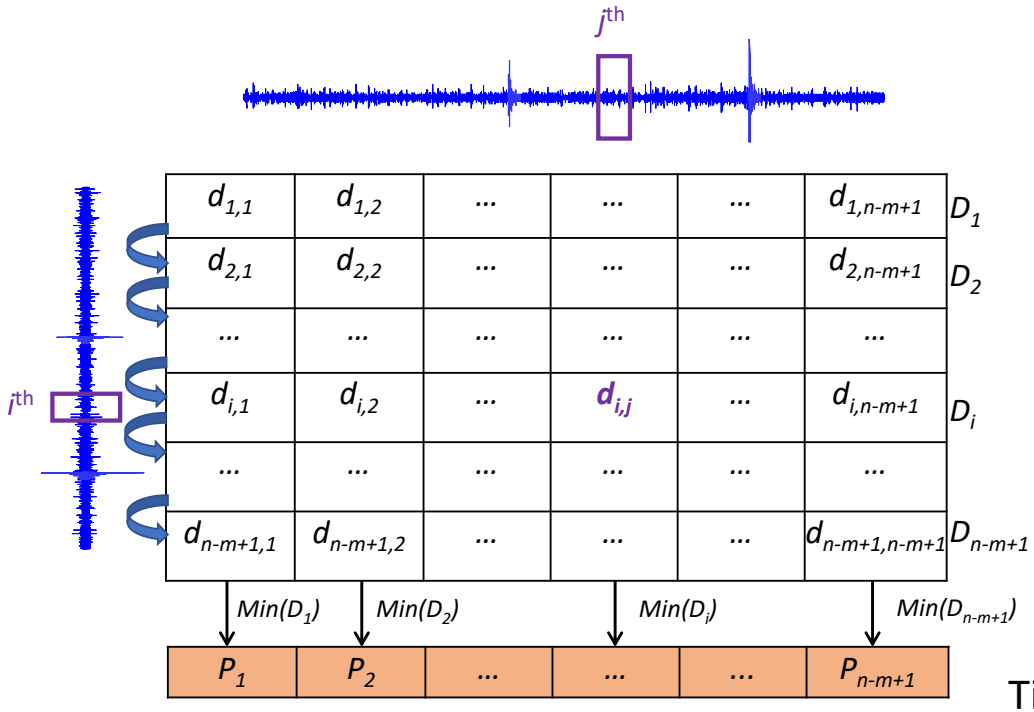
$d_{i,j}$ is the distance between the ith window and the jth window of the time series

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

* Slide credit: Eamonn Keogh; `https://www.cs.ucr.edu/~eamonn/Matrix_Profile_Tutorial_Part2.pdf`

Similarity Search : From Distance Profile to Matrix Profile

STMP: Scalable Time Series Matrix Profile Algorithm



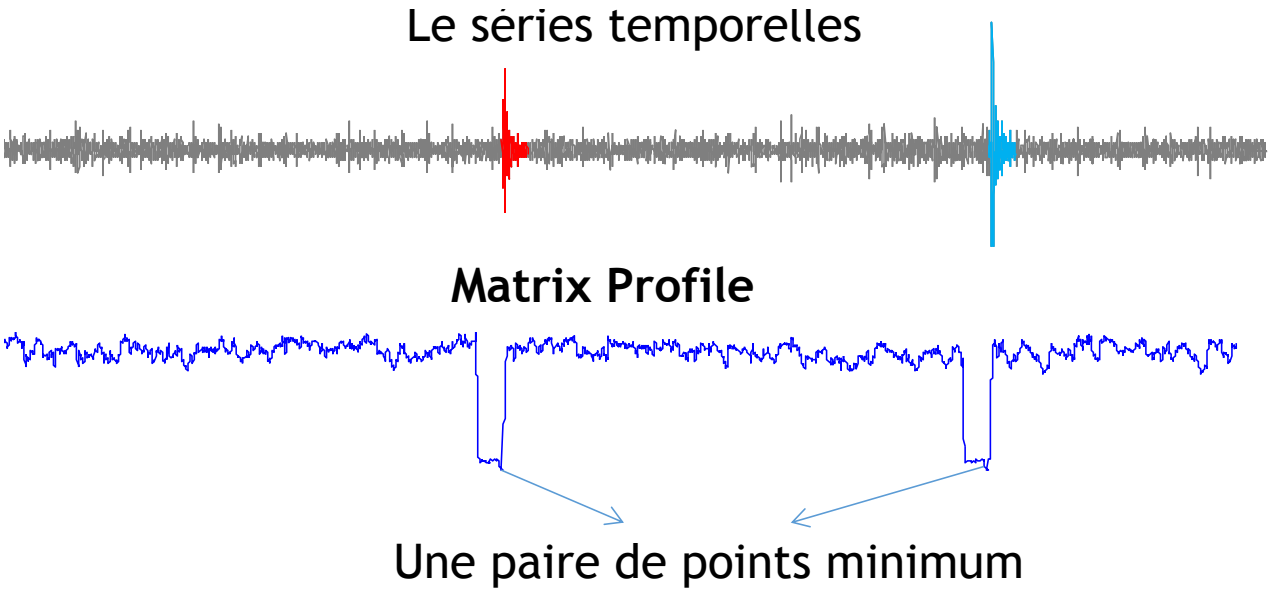
Matrix Profile: a vector of distance between each subsequence and its nearest neighbor

```
MP(1:n-m+1) = inf;
for i = 1:n-m+1
        d = MASS(T,T(i:i+m-1));
        MP = min([MP ; d]);
end
```

Time complexity of STMP is $O(n^2 \log n)$
Space complexity of STMP is $O(n)$

* Slide credit: Eamonn Keogh; `https://www.cs.ucr.edu/~eamonn/Matrix_Profile_Tutorial_Part2.pdf`

Similarity Search : From Distance Profile to Matrix Profile



Le **"Matrix Profil"** a deux points minimum. Cette paire de points minimum correspond au **1**er motif de la série temporelle.
(la paire de sous-séquences la plus proche de la série temporelles)

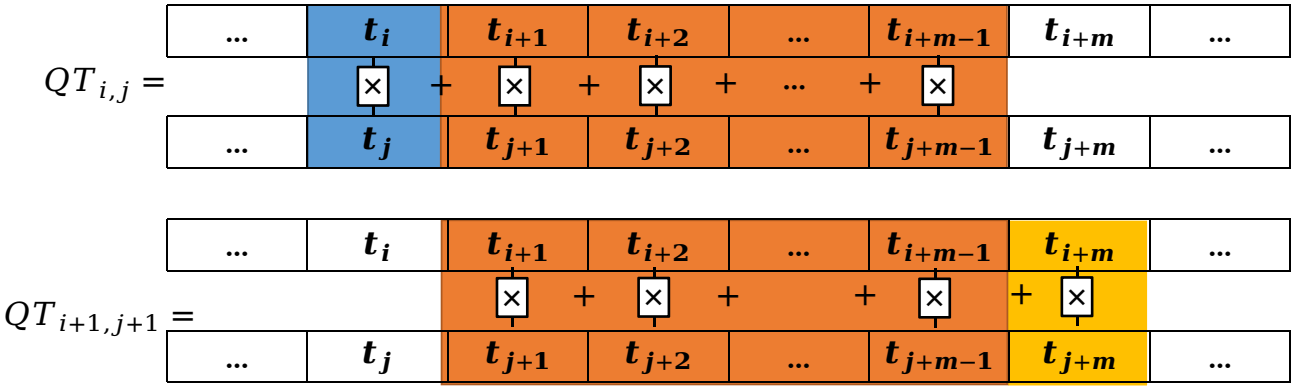
Similarity Search : From Distance Profile to Matrix Profile

Produit scalaire de la *ième* fenêtre et de la *jème* fenêtre. Une fois que nous connaissons $QT_{i,j}$, il faut du temps à *O(1)* pour calculer $d_{i,j}$

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

Nous avons précalculé et stocké les **Moyennes** et les **STDs** dans l'espace **O(n)**

La relation entre $QT_{i,j}$ et $QT_{i+1,j+1}$



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

$O(1)$ time complexity!

AAMP Algorithm: Using Classical Euclidean Distance

AAMP Algorithm: Using Classical Euclidean Distance

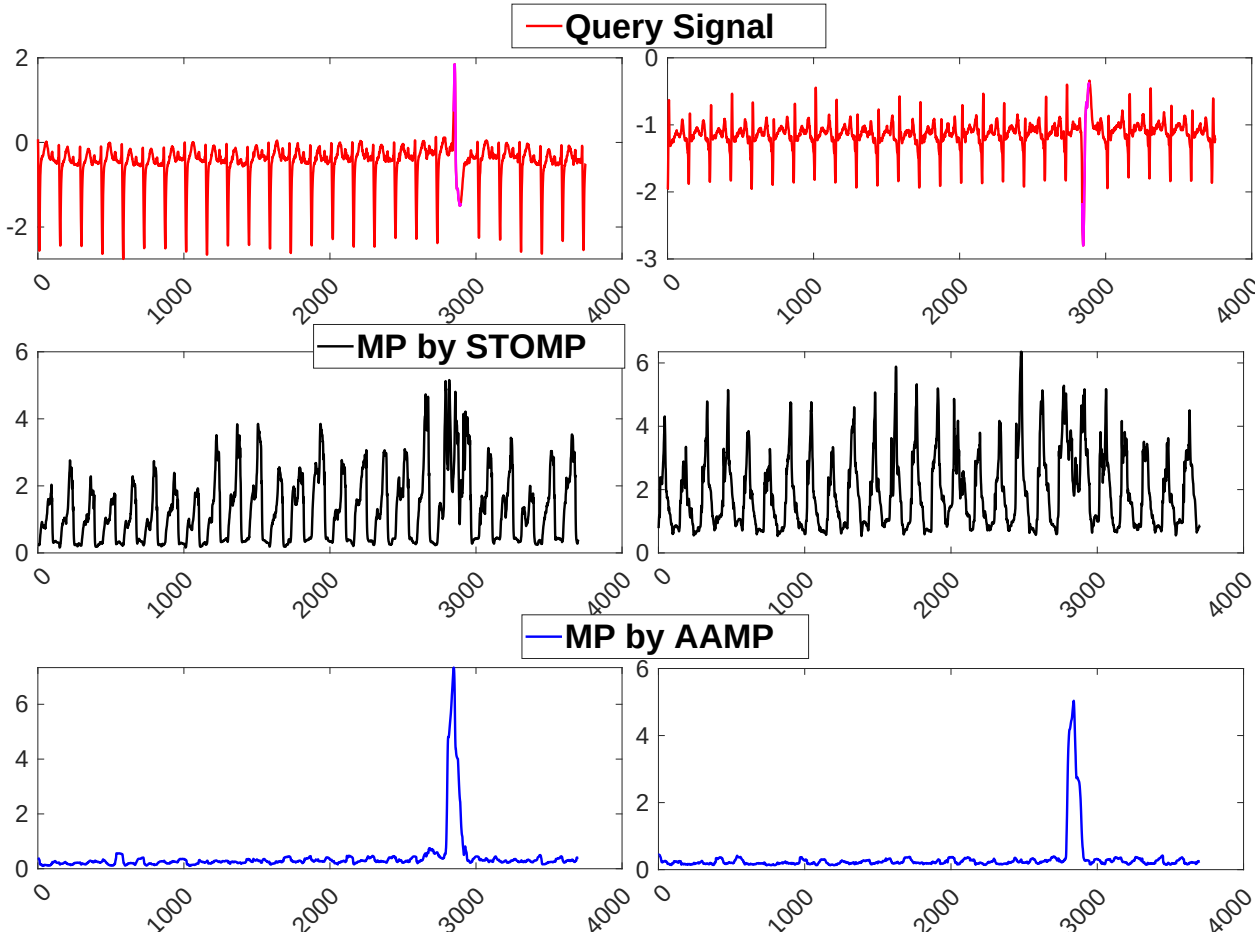


Fig : Top: different time series of the ECG dataset; Middle: matrix profile generated by normalized Euclidean z distance, using STOMP; Bottom: Matrix profile generated by non-normalized Euclidean distance, using AAMP

AAMP Algorithm: Using Classical Euclidean Distance

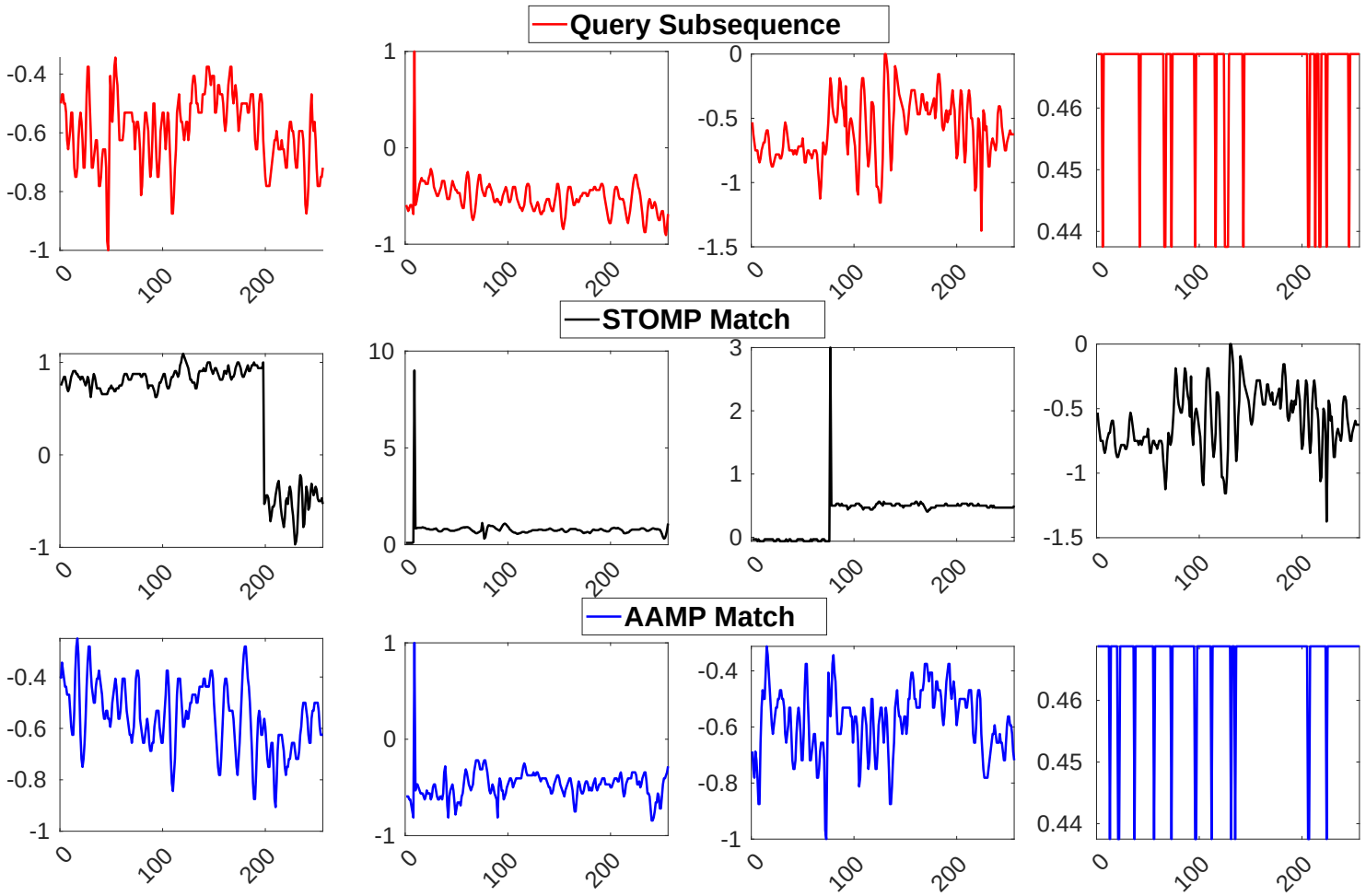


Fig : Top: Four subsequences of length 50 from the sheep database; Middle: nearest neighbors obtained by STOMP; Bottom: Nearest neighbors obtained by AAMP

AAMP Algorithm: Using Classical Euclidean Distance

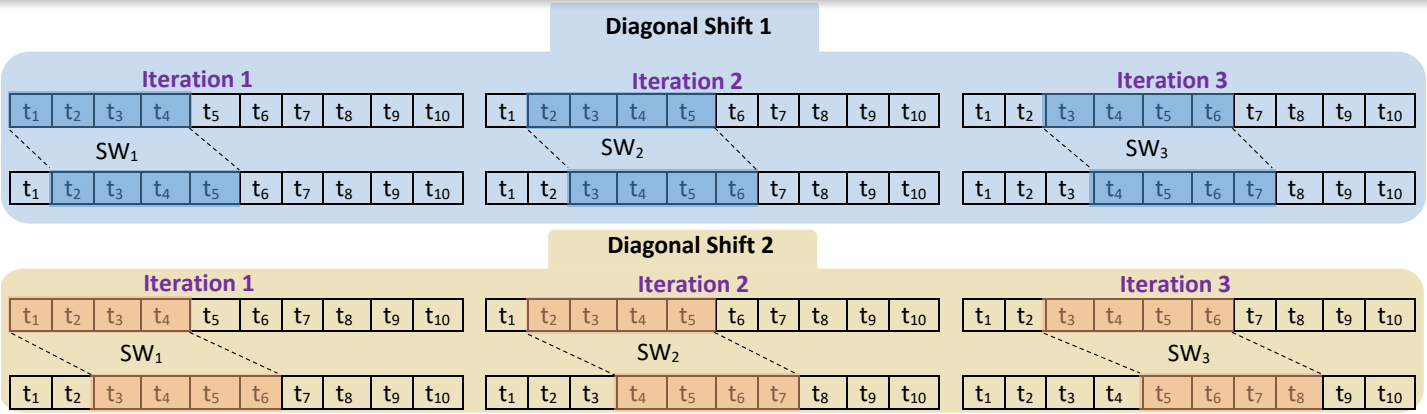


Fig : Exemple d'exécution AAMP sur une série temporelle de longueur $n = 10$, et avec longueur de sous-séquence $m = 4$

	SSq 1	SSq 2	SSq 3	SSq 4	SSq 5	SSq 6	SSq 7
SSq 1		1	2	3	4	5	6
SSq 2	1		1	2	3	4	5
SSq 3	2	1		1	2	3	4
SSq 4	3	2	1		1	2	3
SSq 5	4	3	2	1		1	2
SSq 6	5	4	3	2	1		1
SSq 7	6	5	4	3	2	1	

SSq = Sub-Sequence

Fig : Les sous-séquences sont organisées sous forme de matrice pour mieux comprendre l'algorithme AAMP

$D_{i,j}$: Euclidean distance between $T_{i,m}$ and $T_{j,m}$

$D_{i-1,j-1}$: Euclidean distance between $T_{i-1,m}$ and $T_{j-1,m}$

$$D_{i,j} = \sqrt{D^2_{i-1,j-1} - (t_{i-1} - t_{j-1})^2 + (t_{i+m-1} - t_{j+m-1})^2}$$

- Reza Akbarinia, Bertrand Cloez, **Tanmoy Mondal**, Florent Masseglia; Efficient Matrix Profile Algorithms for Normalized and Non-Normalized Distances; soumis en KDD; 2021;

AAMP Algorithm: Using Classical Euclidean Distance

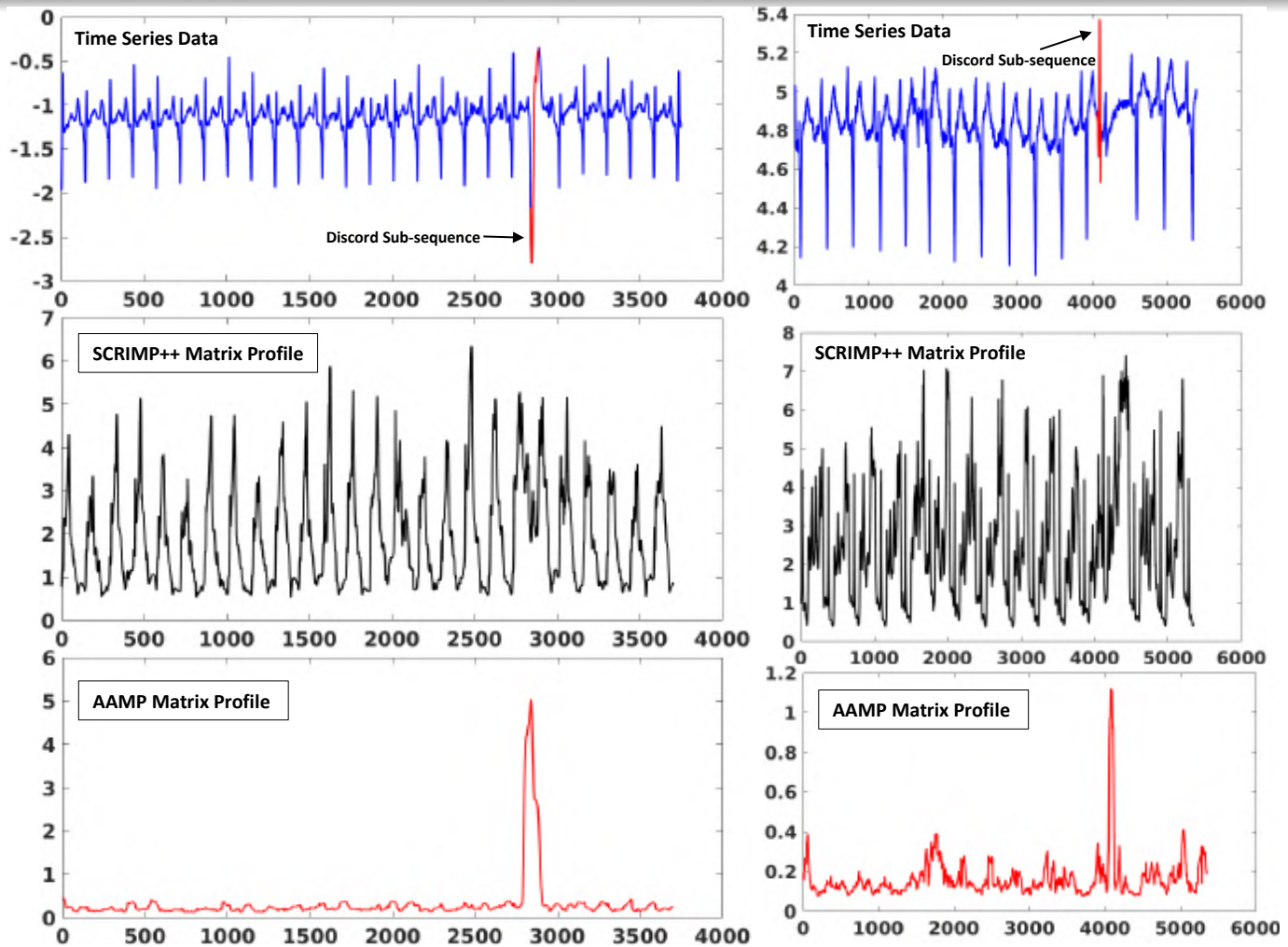


Fig : Top: two time series from a real ECG database. Discordances visible in these time series are marked by the color red. Middle: the matrix profile, obtained by the SCRIMP++ algorithm; Bottom: the profile of the matrix, obtained by the AAMP algorithm.

AAMP Algorithm: Using Classical Euclidean Distance

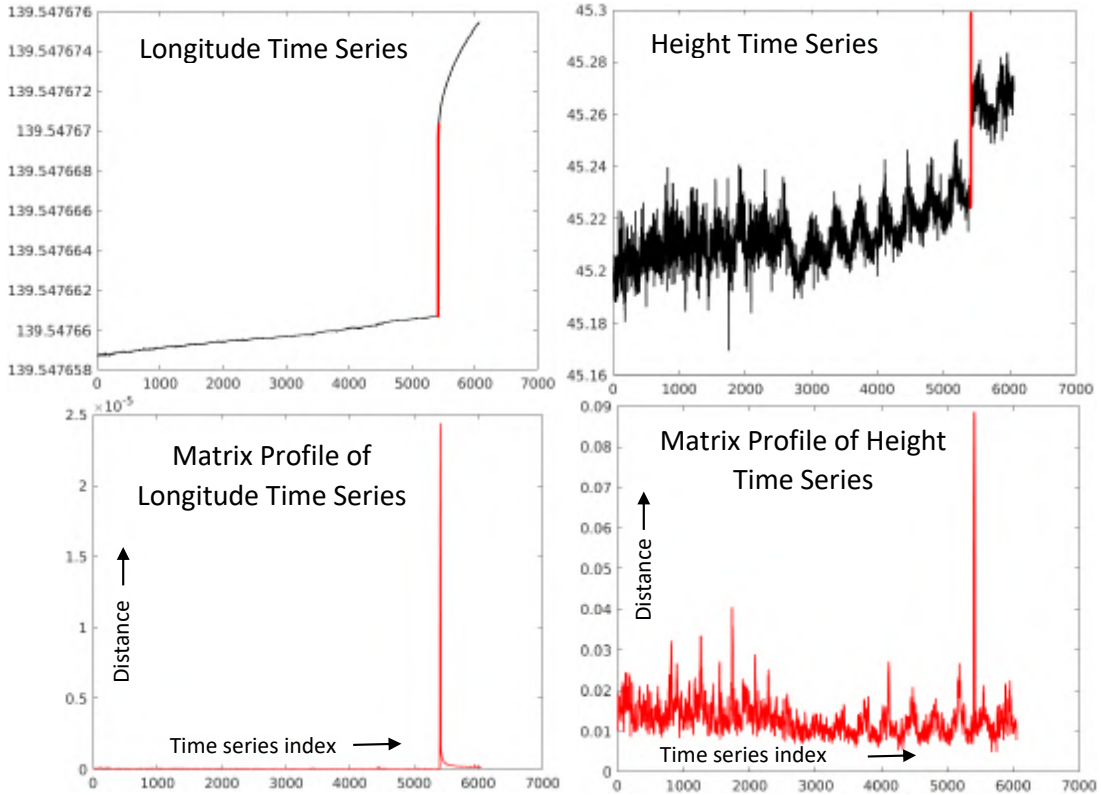


Fig : **Top:** the longitude and height time series of Seismic dataset (outliers are marked by red color); **Bottom:** the matrix profile obtained by AAMP algorithm.

AAMP Algorithm: Using Classical Euclidean Distance

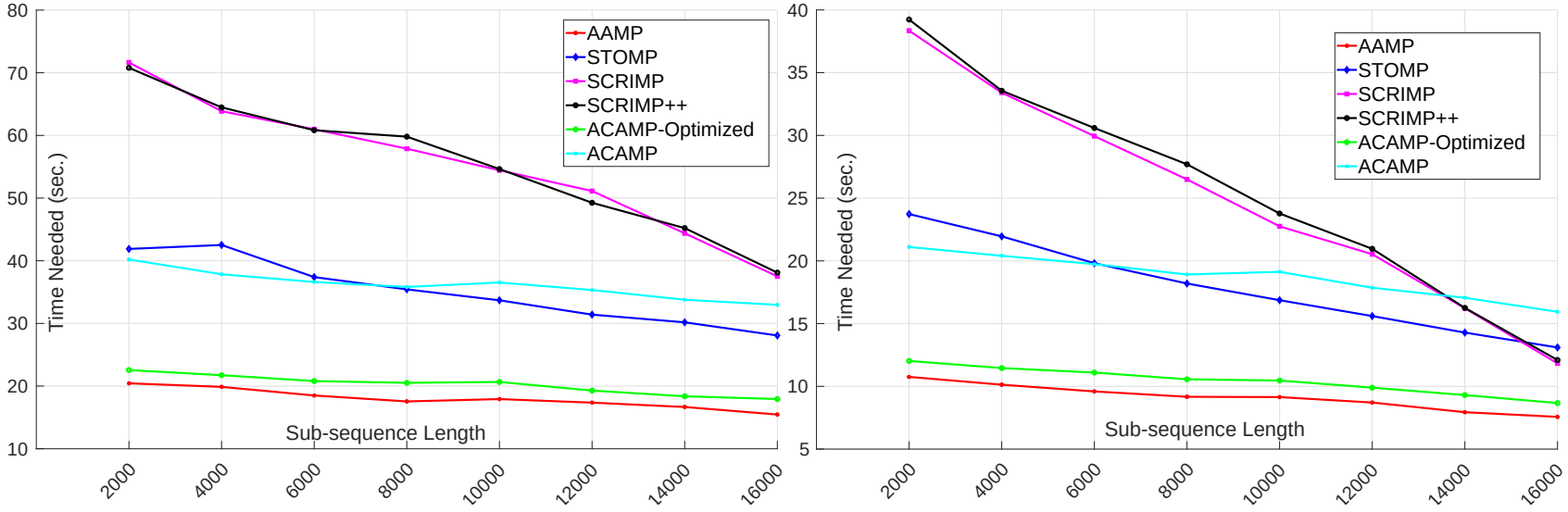


Fig : The execution times of six algorithms with increasing the subsequence length (m): a) Execution time of the six algorithms on a time series of length 68000 (protein dataset). b) Execution time of the six algorithms on a time series of length 50000 (sheep dataset).

AAMP Algorithm: Using Classical Euclidean Distance

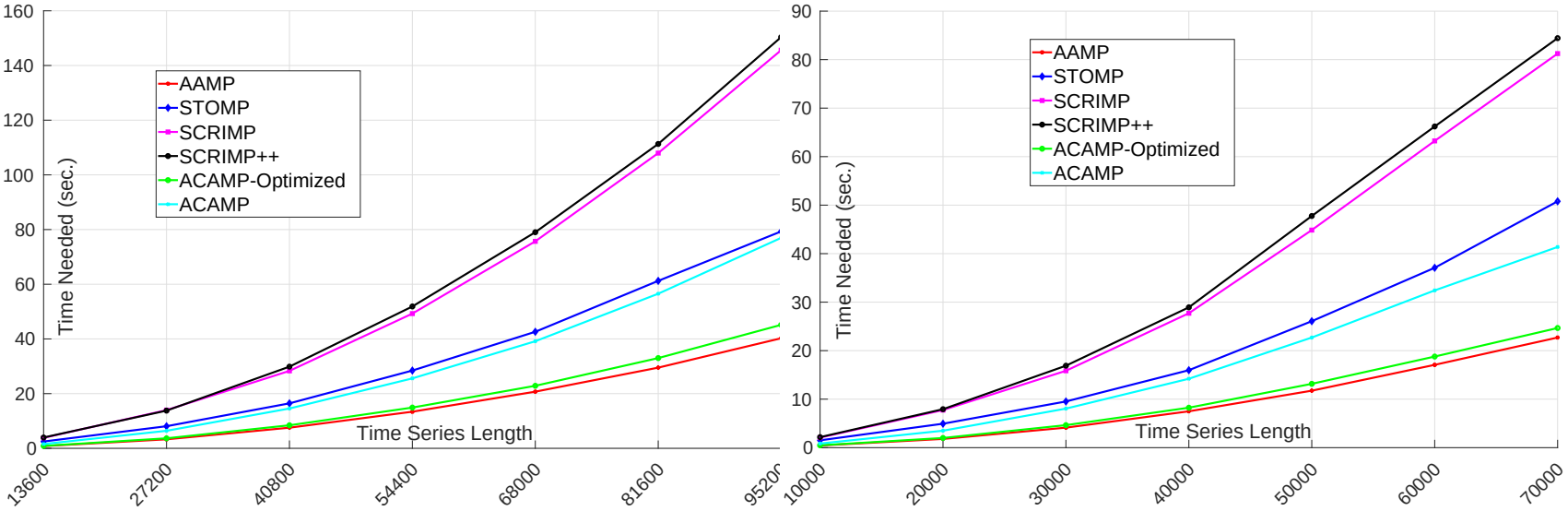
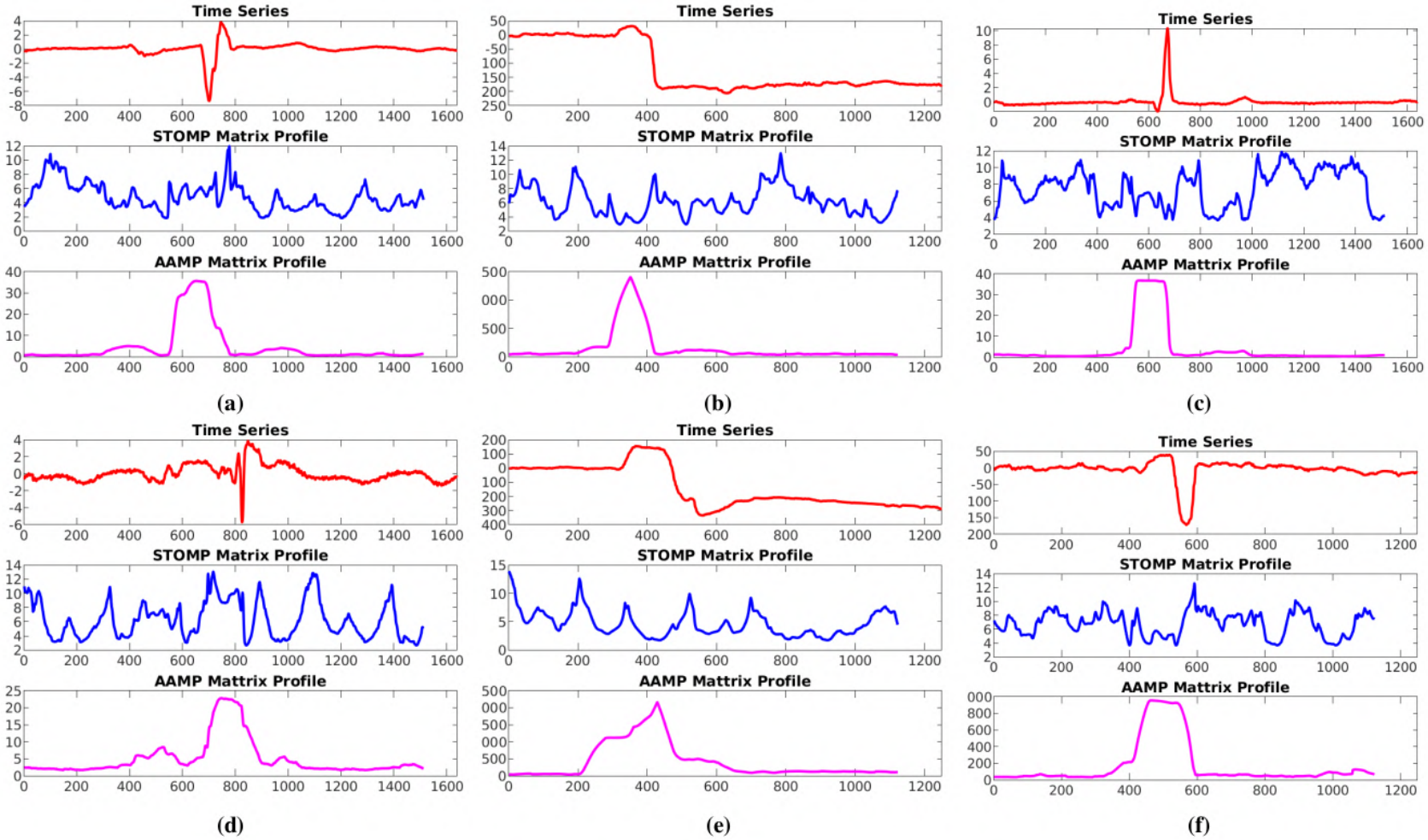


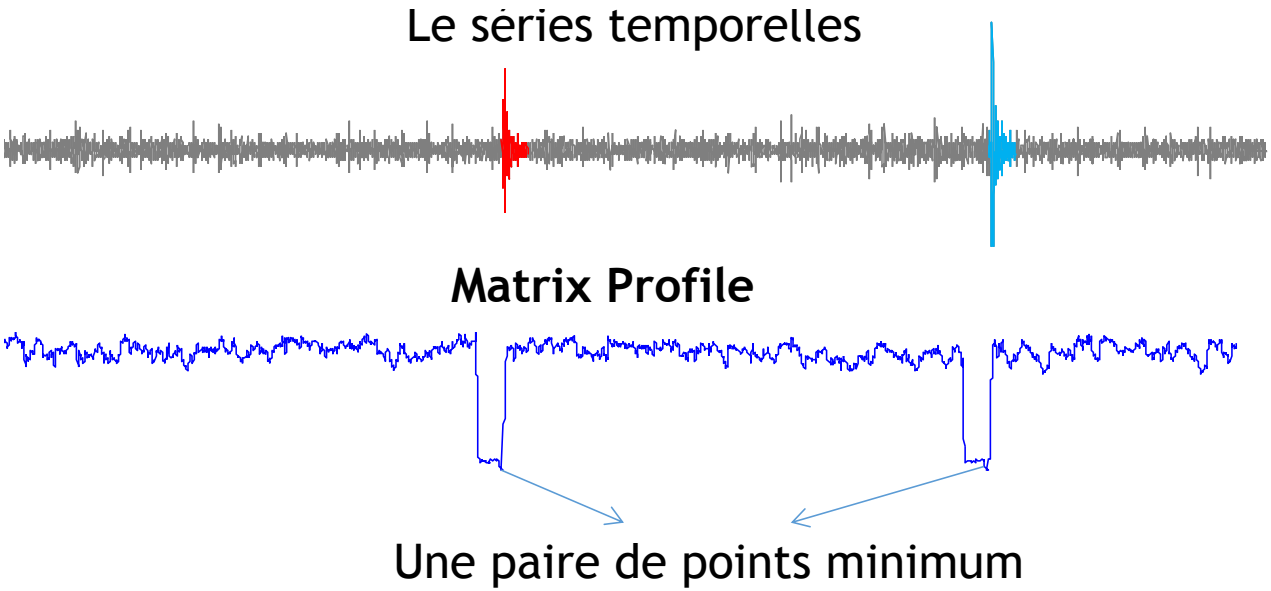
Fig : The execution time of six algorithms are plotted with the increase of time series length (n): c) Execution time of the six algorithms on variable time series length (protein dataset) with $m = 256$. d) execution time of the six algorithms on variable time series length (sheep dataset) with $m = 256$.

AAMP Algorithm: Using Classical Euclidean Distance



kNN Matrix Profile

Recap : What is 1NN Matrix Profile



Le **"Matrix Profil"** a deux points minimum. Cette paire de points minimum correspond au **1**er motif de la série temporelle. **(la paire de sous-séquences la plus proche de la série temporelles)**

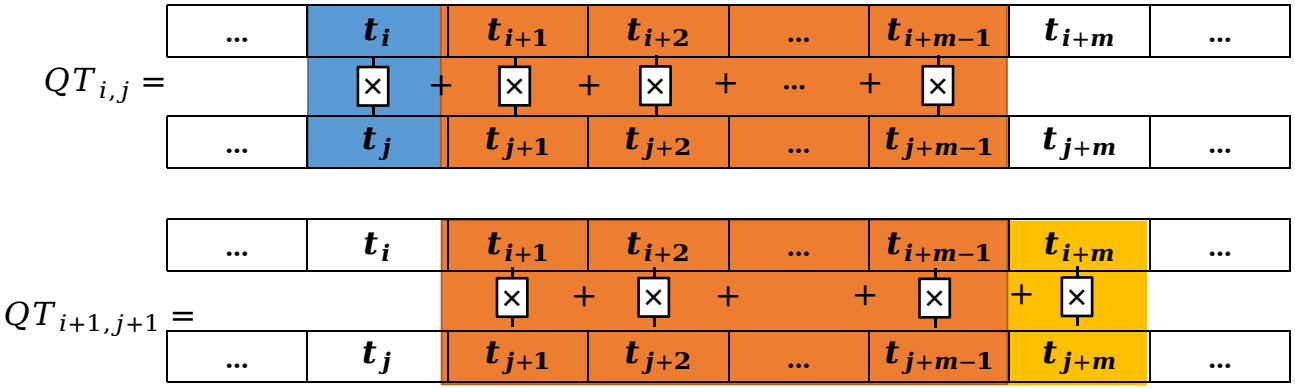
Recap : What is 1NN Matrix Profile

Produit scalaire de la *ième* fenêtre et de la *jème* fenêtre. Une fois que nous connaissons $QT_{i,j}$, il faut du temps à *O(1)* pour calculer $d_{i,j}$

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

Nous avons précalculé et stocké les **Moyennes** et les **STDs** dans l'espace **O(n)**

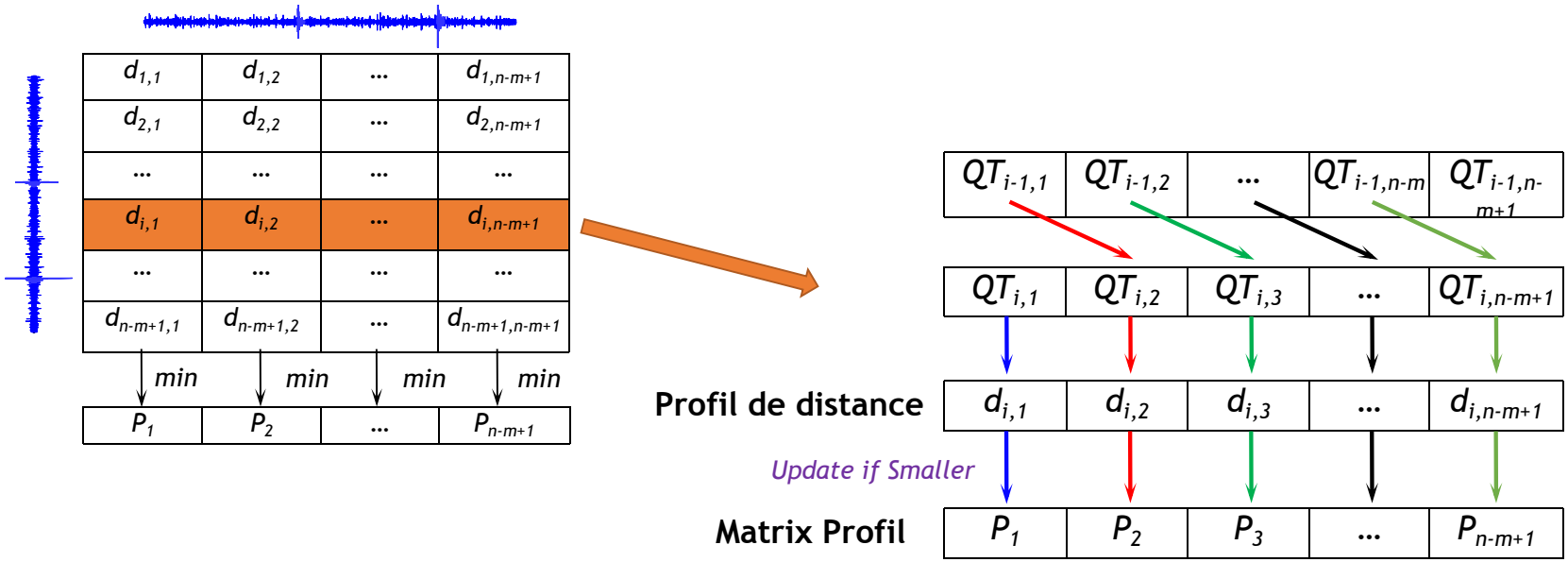
La relation entre $QT_{i,j}$ et $QT_{i+1,j+1}$



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

$O(1)$ time complexity!

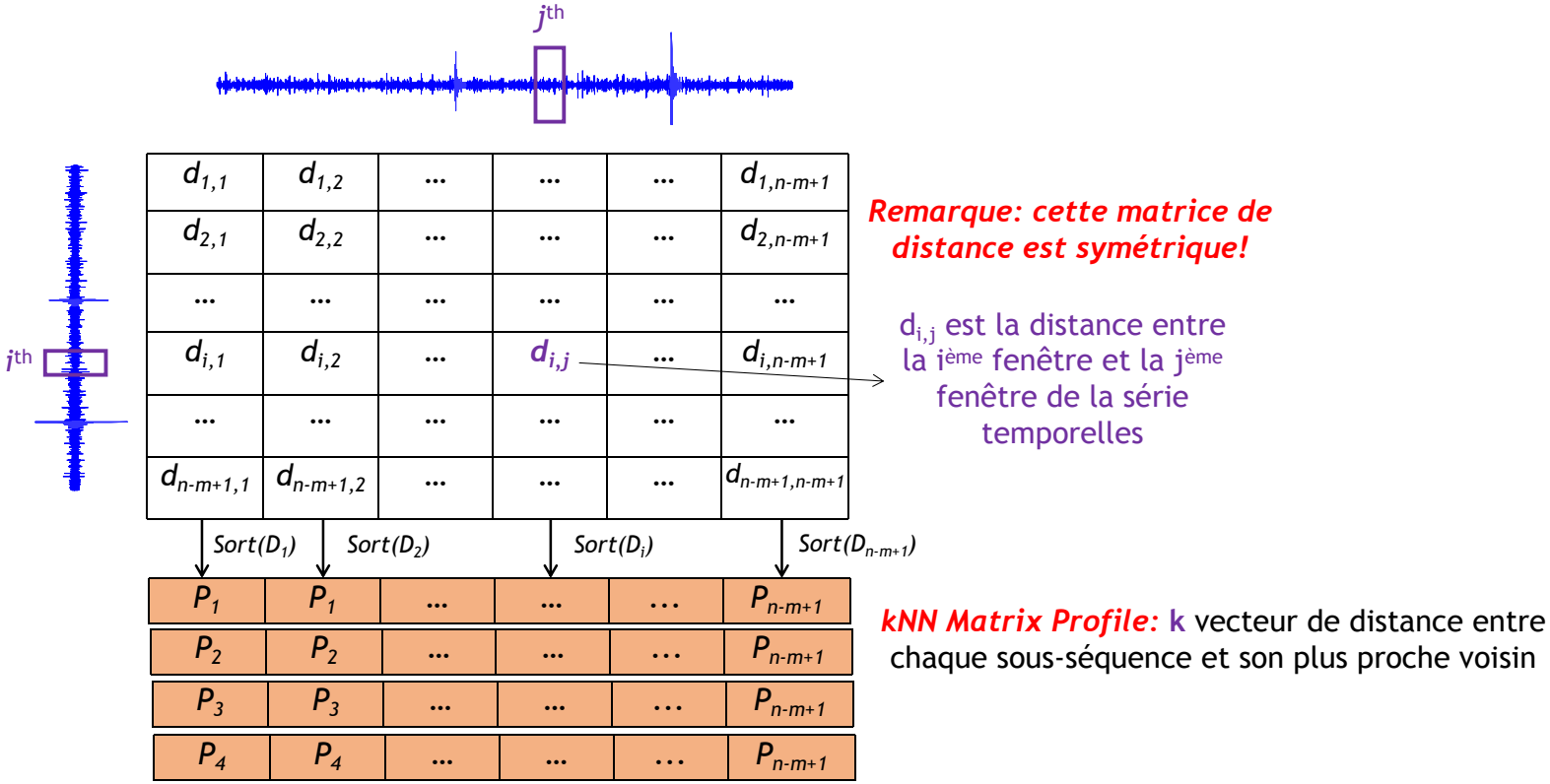
Recap : What is 1NN Matrix Profile



Nous pré-calculons $QT_{x,1}$ et $QT_{1,x}$ (x=1,2,3,...,n-m+1). Puis itérer à travers i=2, 2, 3, ..., n-m+1

kNN Matrix Profile

- Technique proposed for the formulation of kNN Matrix Profile
- Obtain distance with query sub-seqeunecs and the **target-1** then the **target-2** until the $\mathbf{K}^{th}$ target
- Continue to update the $(K+1)^{th}$ match; use sorting to find; the maximum value and discard it; or use the technique based on **"Priority Queue"**



Importance of kNN Matrix Profile

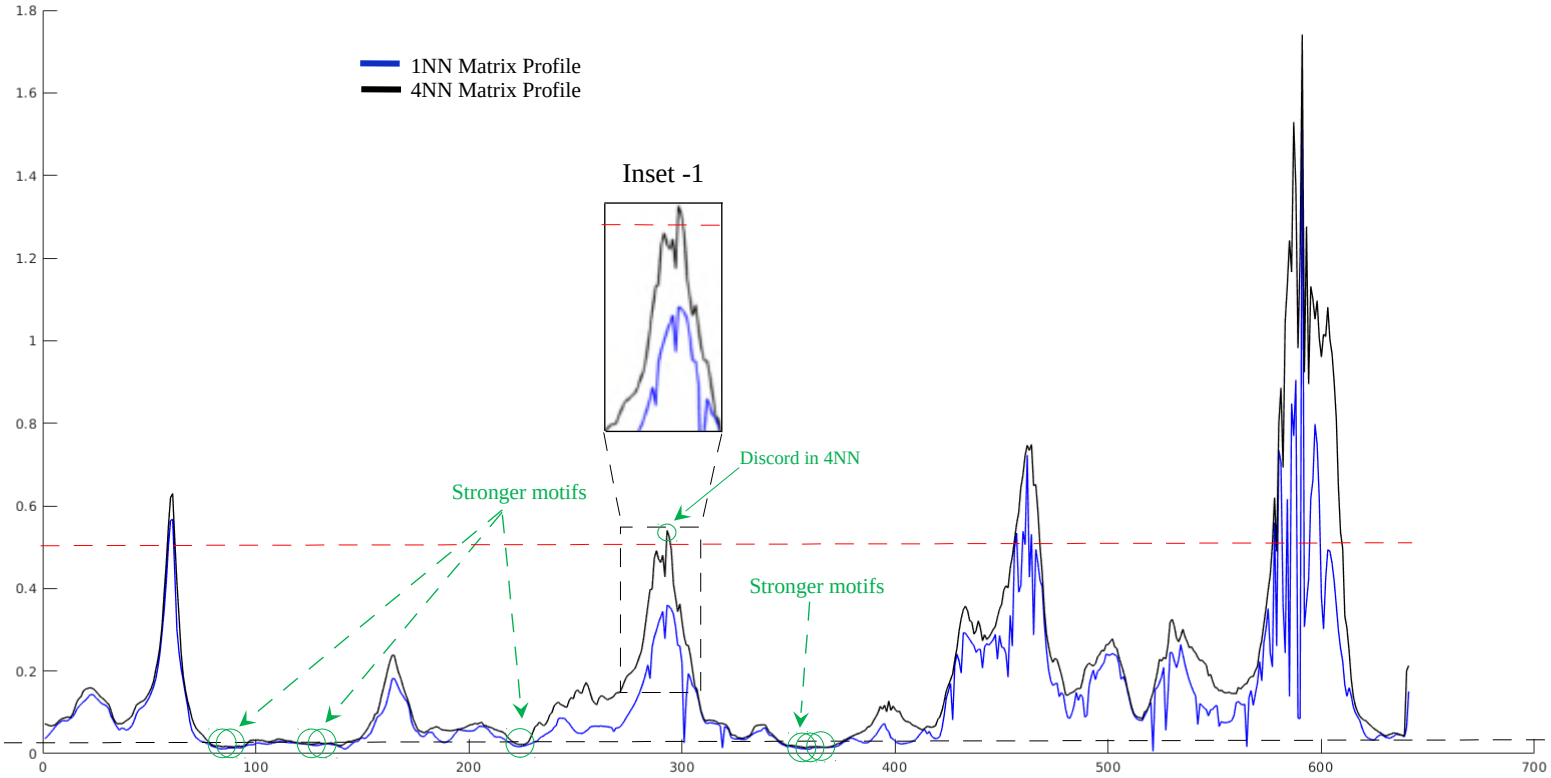


Fig. The 1NN and 4NN MP are plotted with different colors in which the motifs and discords are marked.

Importance of kNN Matrix Profile

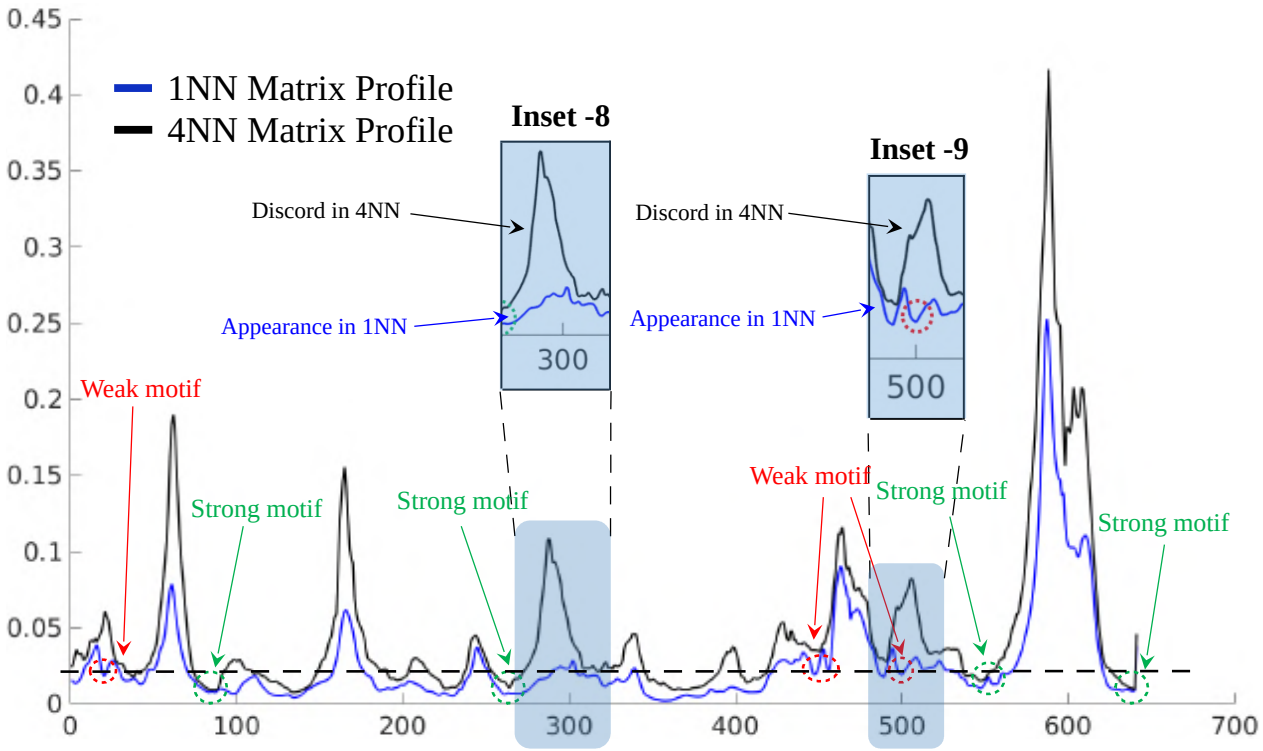


Fig : The 1NN and 4NN MP are plotted with different colors in which the motifs and discords are marked

- **Tanmoy Mondal**, Reza Akbarinia, Florent Masseglia; "Matrix Profile Based kNN Search over Large Time Series"; Data Mining and Knowledge Discovery (DMKD), 2023

Importance of kNN Matrix Profile

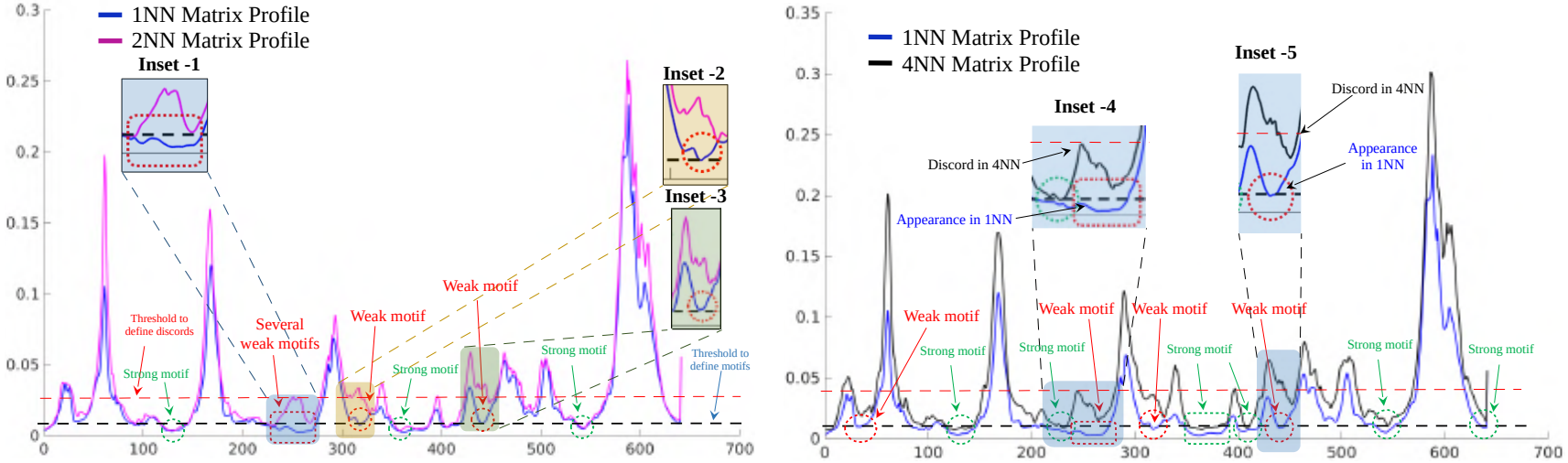


Fig. The extracted motifs and discords are illustrated in the plot : (a) The 1NN vs 2NN MP and (b) 1NN vs 4NN MP of a part of the time series is plotted

Importance of kNN Matrix Profile

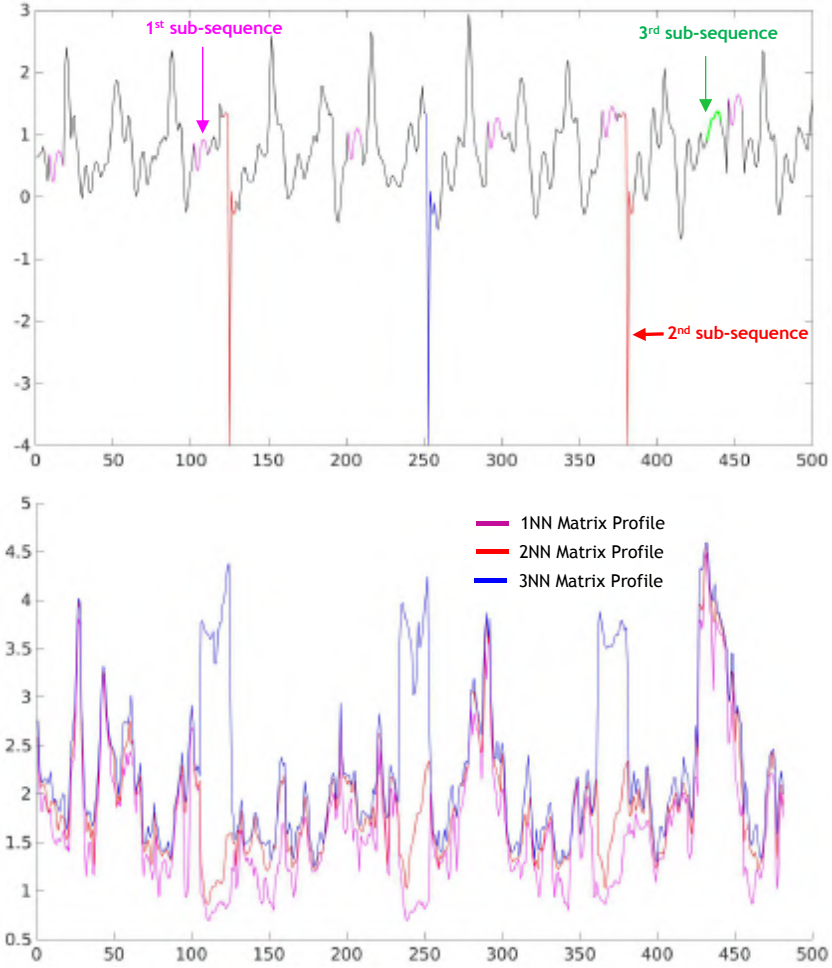


Fig. Special case of the outliers presence is depicted by considering a toy time series (top). The 1NN, 2NN and 3NN MPs are shown for the time series (bottom)

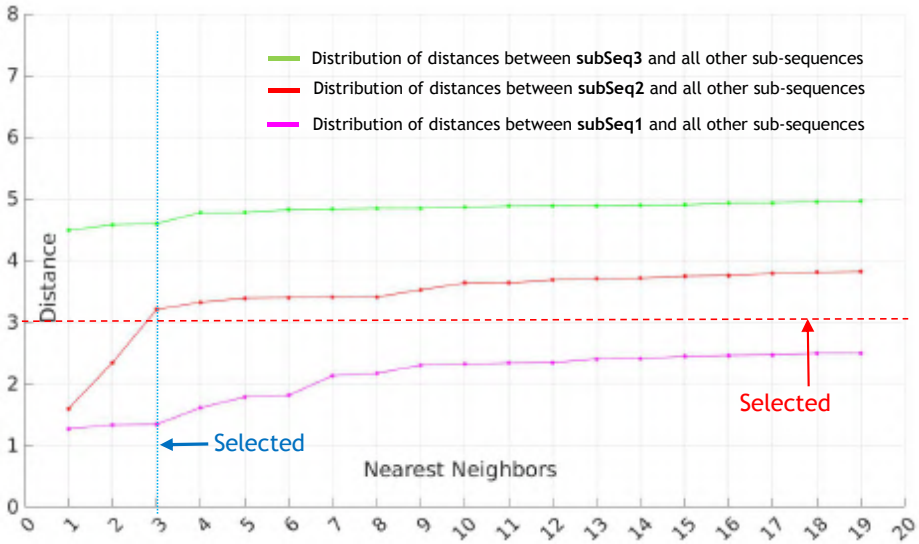


Fig. One special case of outliers detection

Importance of kNN Matrix Profile

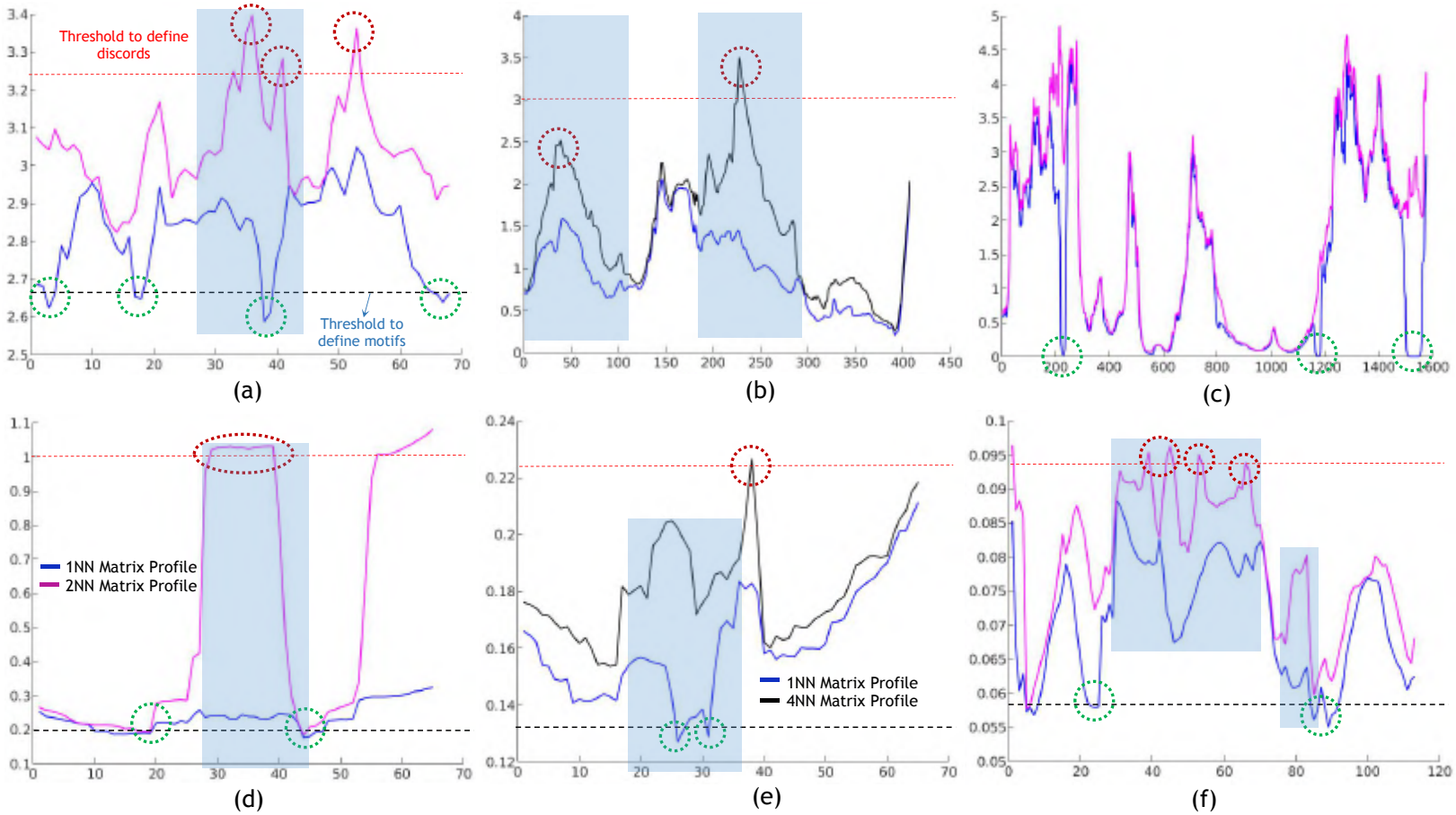


Fig. The results of UCR dataset

Importance of kNN Matrix Profile

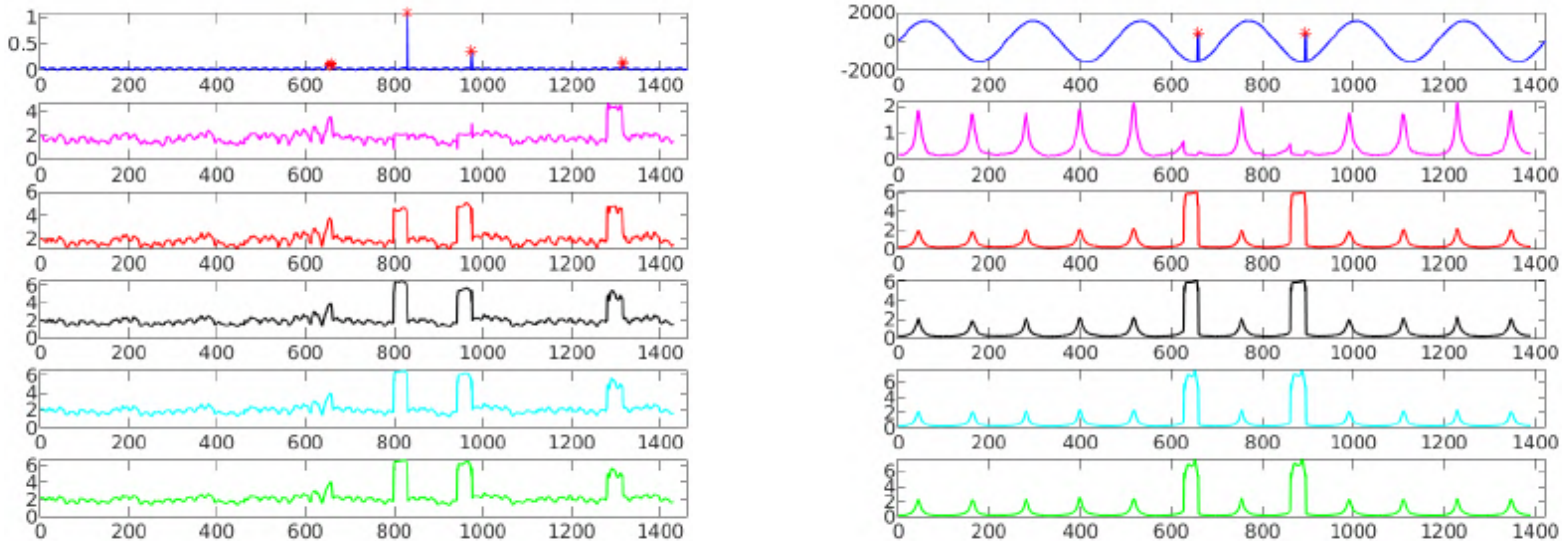


Fig. The usefulness of the kNN MP using Yahoo dataset. Time series is plotted at the top followed by 1N N , 2N N , 3N N , 4N N , and 5N N MPs.

Importance of kNN Matrix Profile

Table: The outlier detection accuracy of various kNN MP based on 3 high thresholds on the Yahoo dataset ("A1Benchmark-Real")

	Accuracies		
kNN MP	**Threshold (95%)**	**Threshold (90%)**	**Threshold (85%)**
1NN	0.317	0.413	0.469
2NN	0.349	0.485	0.556
3NN	0.386	0.509	0.584
4NN	0.439	0.522	0.630
5NN	0.458	0.553	0.653
6NN	0.490	0.566	0.673
7NN	0.500	0.610	0.686
8NN	0.509	0.622	0.698
9NN	0.522	0.629	0.704
10NN	0.542	0.643	0.720

Parallel Execution of kNN Matrix Profile

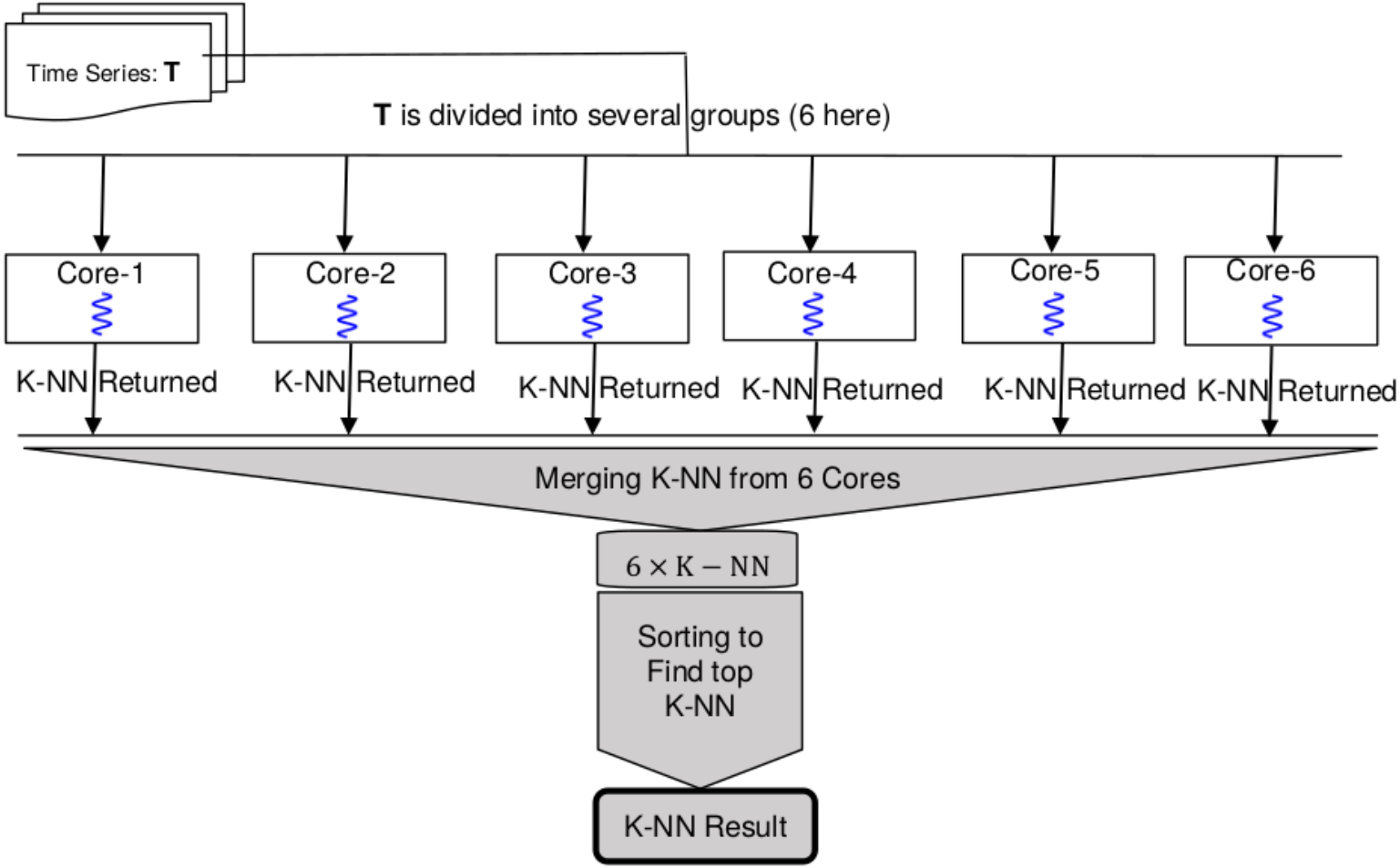


Fig : Architecture of parallel execution by using multiple cores

Performance : kNN Matrix Profile

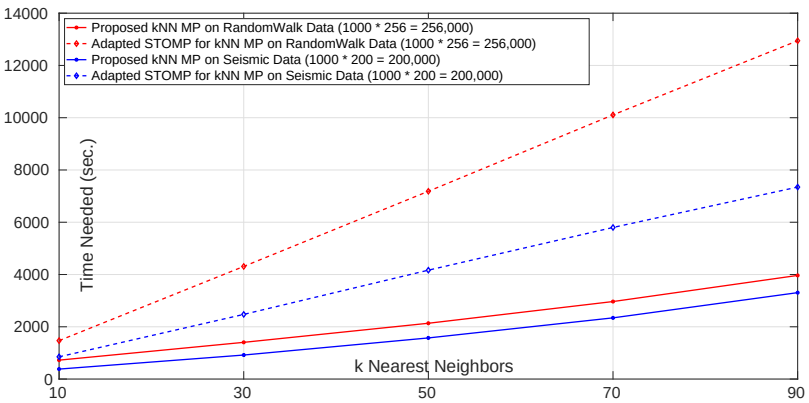


Fig. The variation of execution time with increasing k for generating *kNN* MP

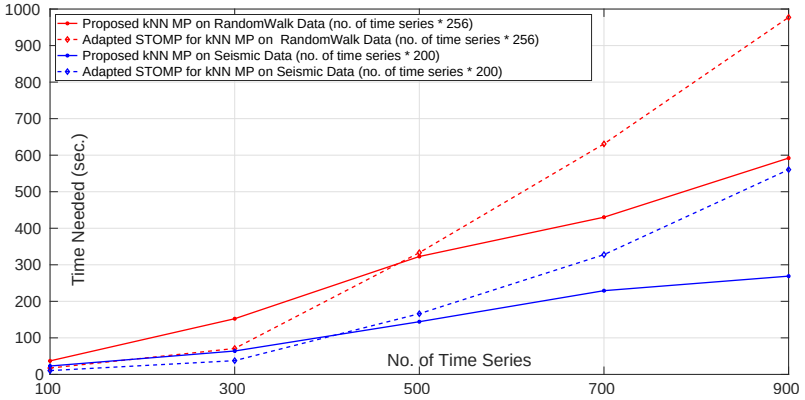


Fig. The variation of execution time with increasing the length of time series

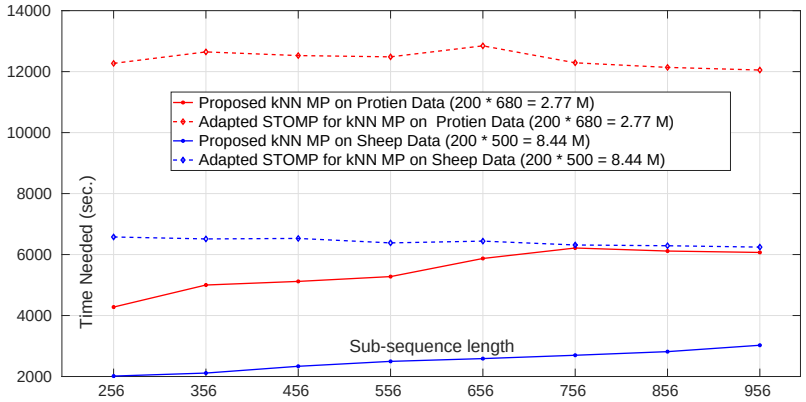


Fig. The variation of computational time with increasing the length of subsequence (m) for the protein and sheep datasets respectively

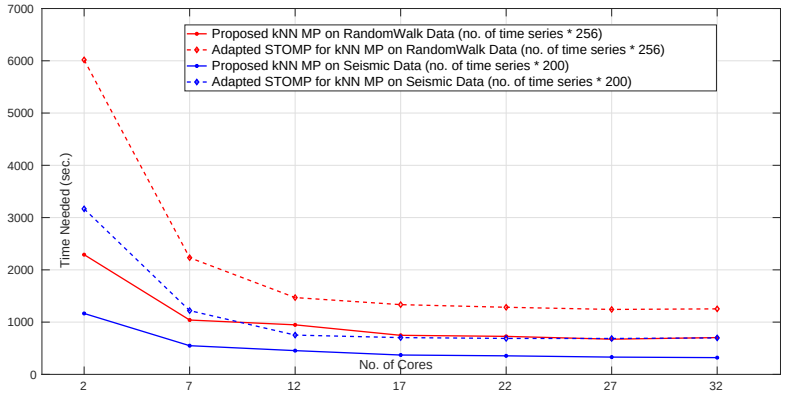


Fig. The computational time with increasing the number of cores for the random-walk and seismic datasets respectively

☺ Thank you for your attention.. ☺