

# Матричный профиль временного ряда

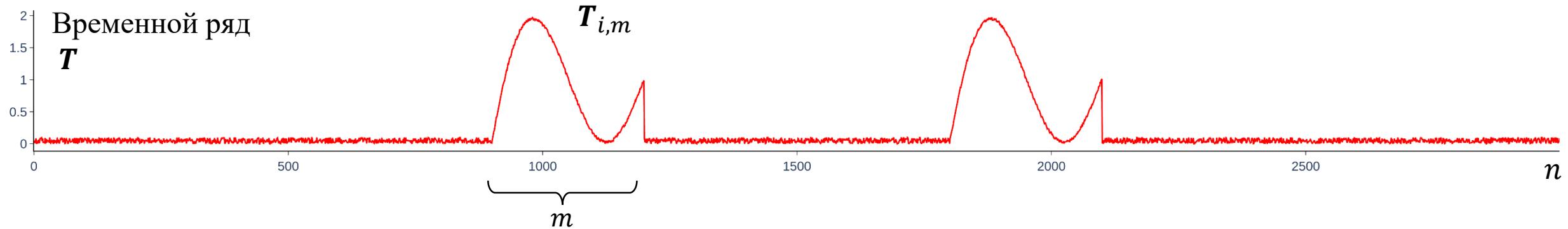


*What is the Matrix? Control.  
Morpheus*

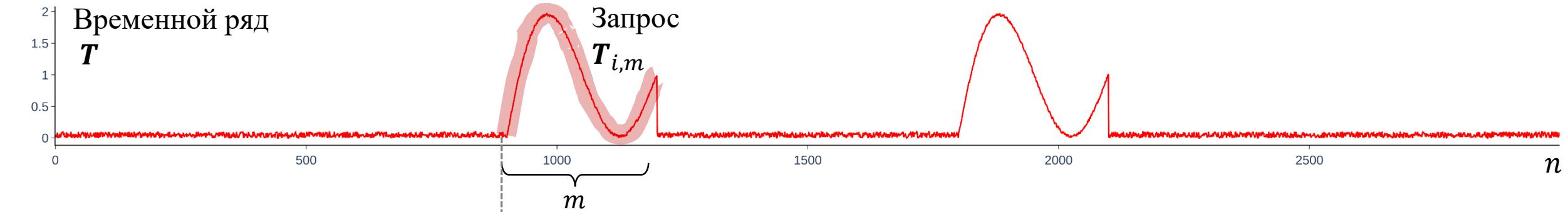
# Содержание

- Понятие матричного профиля
- Примеры задач, решаемых на основе матричного профиля
- Алгоритмы вычисления матричного профиля

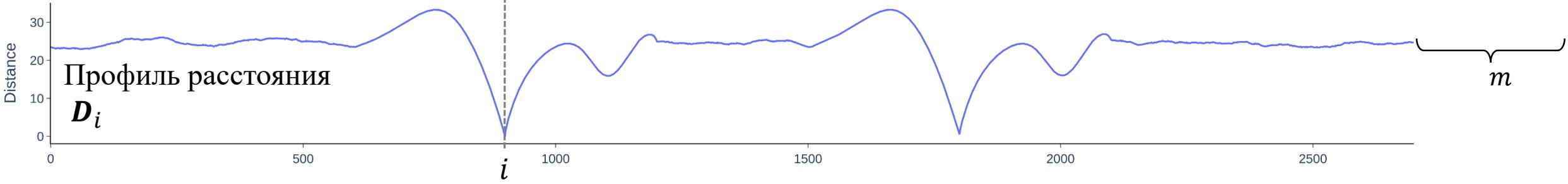
# Матричный профиль: Временной ряд



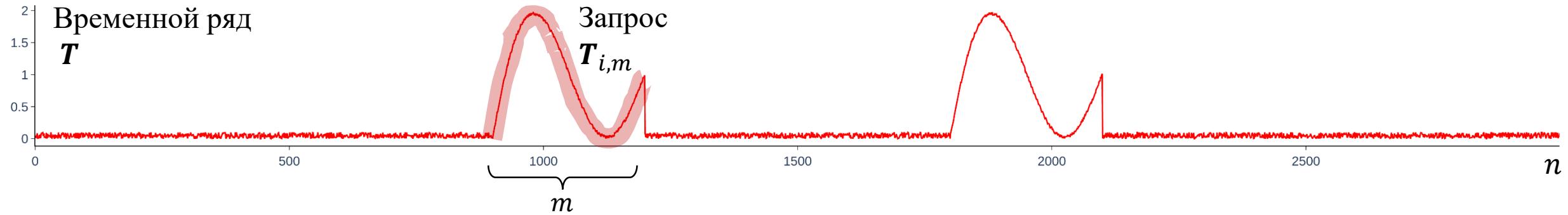
# Матричный профиль: Профиль расстояния



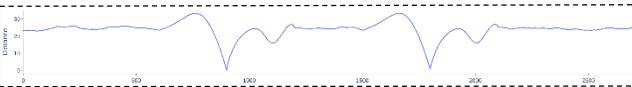
$$D_i = \{\text{Dist}(Q, T_{j,m})\}_{j=1}^{n-m+1}, \quad Q = T_{i,m}, \quad 1 \leq i \leq n - m + 1$$



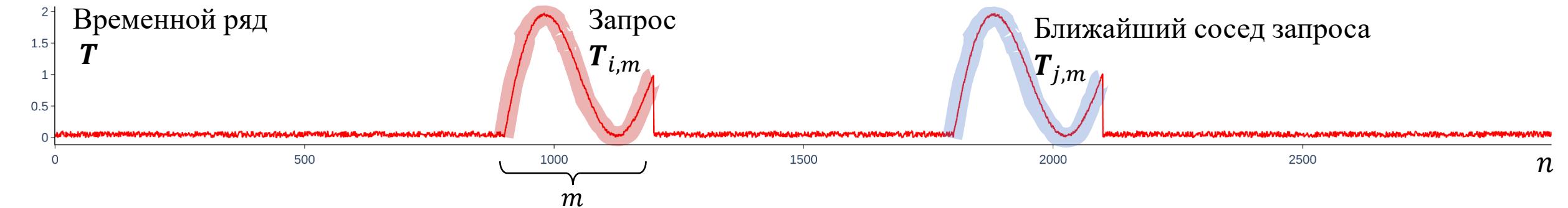
# Матричный профиль: Матрица профилей расстояния



	$T_{1,m}$	$T_{2,m}$	...	$T_{i,m}$	...	$T_{n-m+1,m}$
$D_1$	0	Dist( $\textcolor{red}{T}_{1,m}, T_{2,m}$ )				
$D_2$	Dist( $\textcolor{red}{T}_{2,m}, T_{1,m}$ )	0				
...	...		0		...	
$D_i$	Dist( $\textcolor{red}{T}_{i,m}, T_{1,m}$ )	Dist( $\textcolor{red}{T}_{i,m}, T_{2,m}$ )	...	0		
...		...			0	...
$D_{n-m+1}$	Dist( $\textcolor{red}{T}_{n-m+1,m}, T_{1,m}$ )	Dist( $\textcolor{red}{T}_{n-m+1,m}, T_{2,m}$ )	...	Dist( $\textcolor{red}{T}_{n-m+1,m}, T_{i,m}$ )	...	0

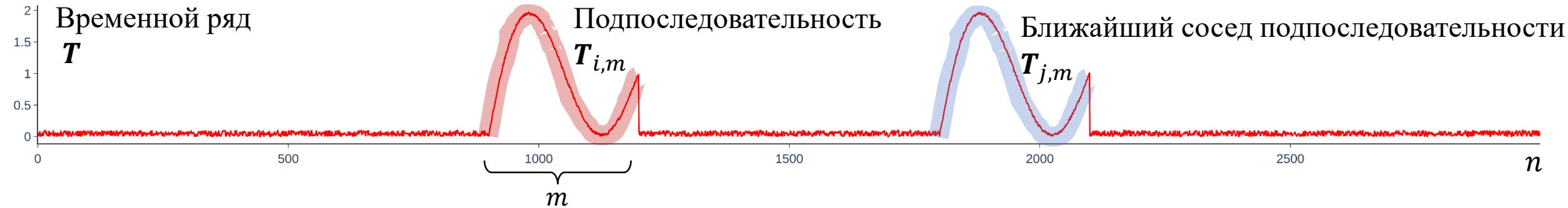


# Матричный профиль: Поиск ближайших соседей



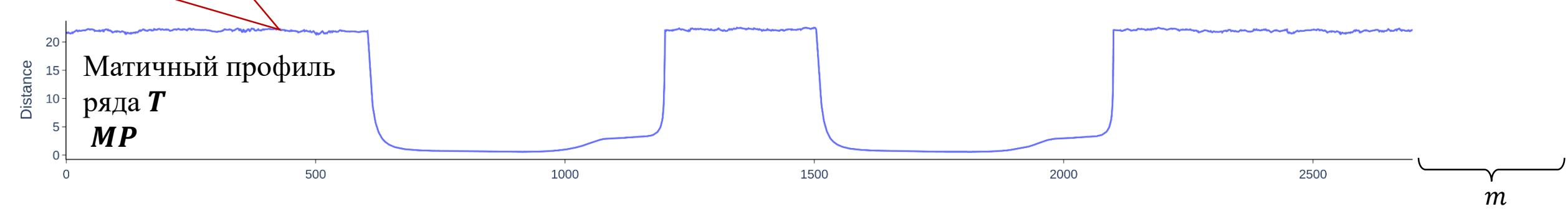
	$T_{1,m}$	$T_{2,m}$	...	$T_{i,m}$	...	$T_{n-m+1,m}$
$\min_{\substack{1 \leq i < n-m+1 \\  i-1 >m}} \text{Dist}(T_{i,m}, T_{1,m})$	$D_1$	0	$\text{Dist}(T_{1,m}, T_{2,m})$			
	$D_2$	$\text{Dist}(T_{2,m}, T_{1,m})$	0			
	...	...	0	...		
$\min_{\substack{1 \leq i < n-m+1 \\  i-j >m}} \text{Dist}(T_{i,m}, T_{j,m})$	$D_i$	$\text{Dist}(T_{i,m}, T_{1,m})$	$\text{Dist}(T_{i,m}, T_{2,m})$	...	0	
	...	...	...	...	0	
Соседи не пересекаются: $ i - j  > m$	$D_{n-m+1}$	$\text{Dist}(T_{n-m+1,m}, T_{1,m})$	$\text{Dist}(T_{n-m+1,m}, T_{2,m})$	...	$\text{Dist}(T_{n-m+1,m}, T_{i,m})$	...

# Матричный профиль

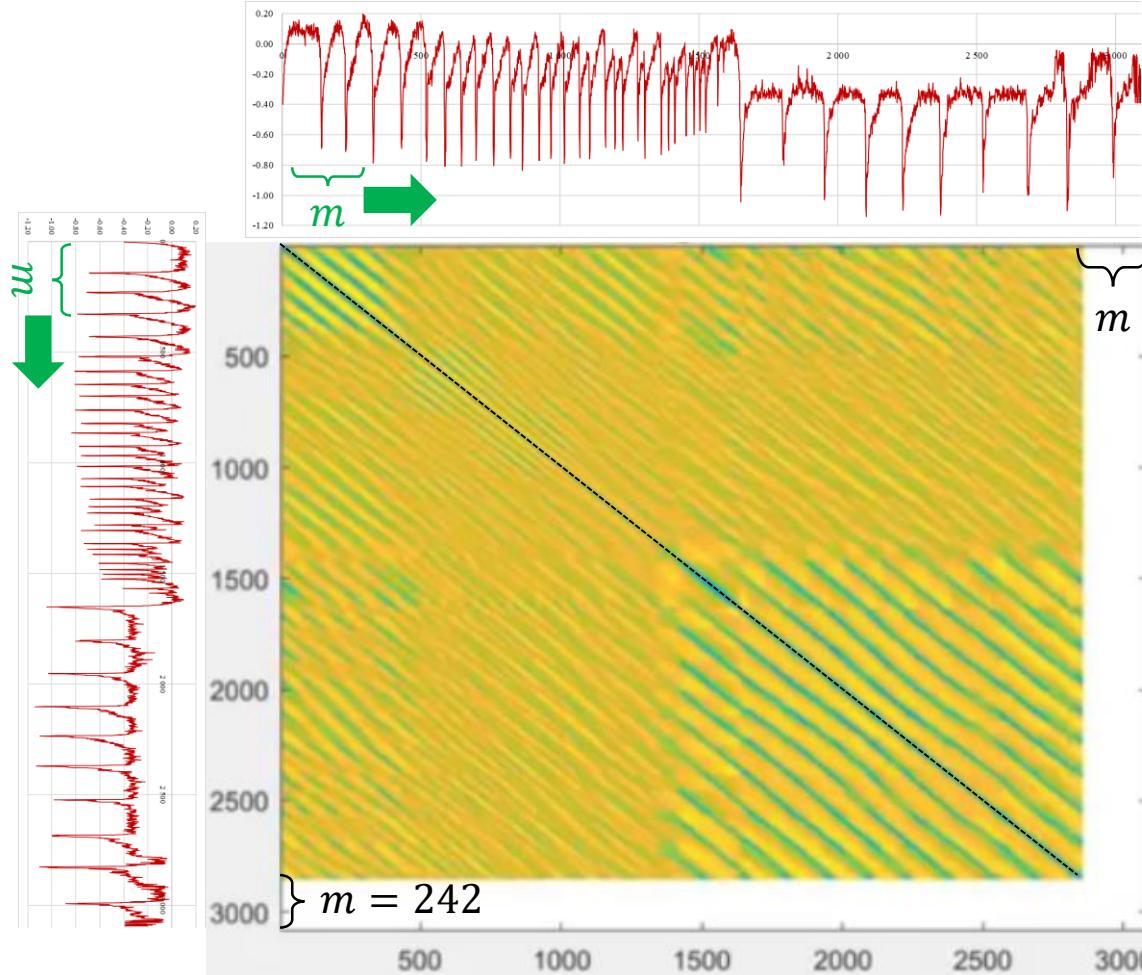


**Расстояния  
до ближайшего соседа  
подпоследовательностей ряда**

$$MP_T^m(i) = \min_{\substack{T_{j,m} \in S_T^m \\ |i-j|>m}} D_i$$



# Почему профиль матричный?

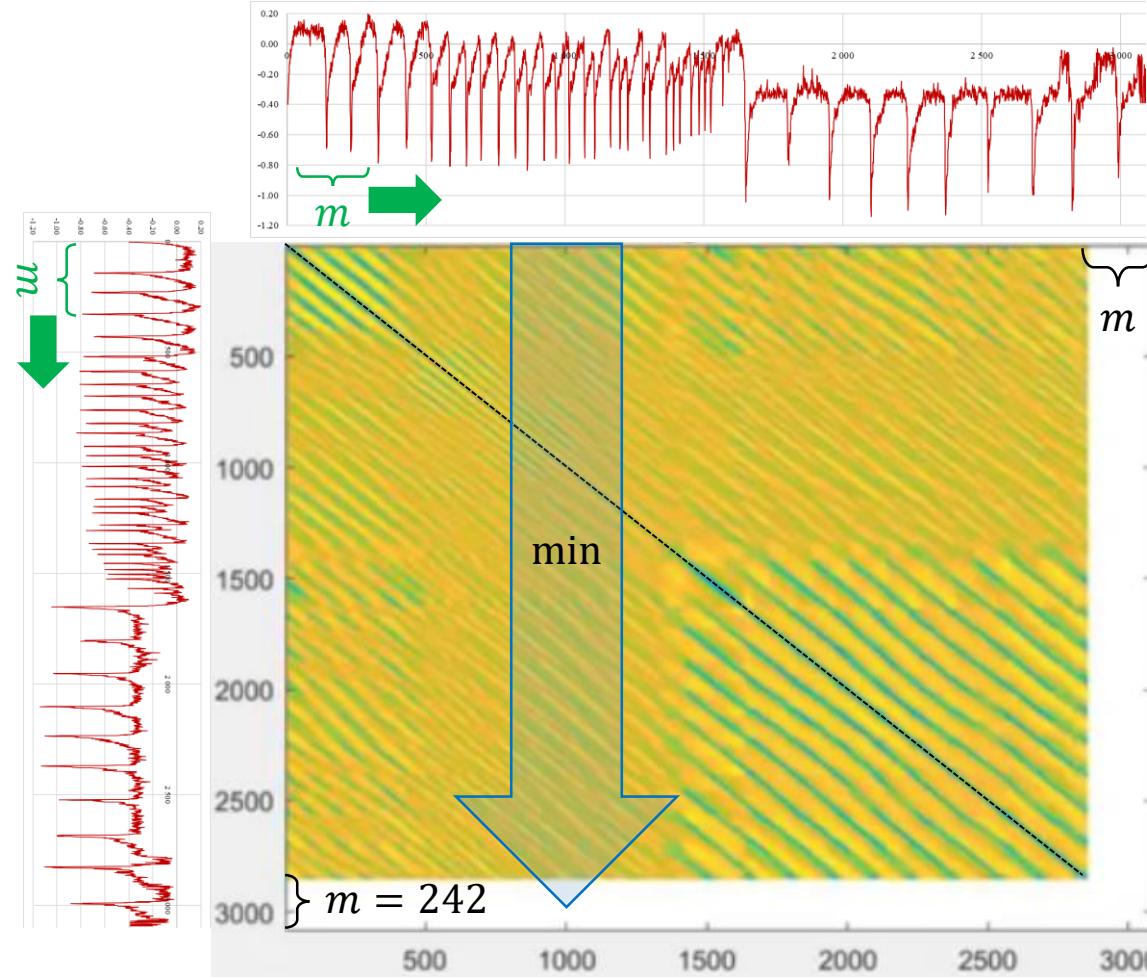


**Матрица расстояний**

$$DistMatr_T^m \in \mathbb{R}^{(n-m+1) \times (n-m+1)}$$

$$DistMatr_T^m(i, j) = \text{Dist}(T_{i,m}, T_{j,m})$$

# Почему профиль матричный?



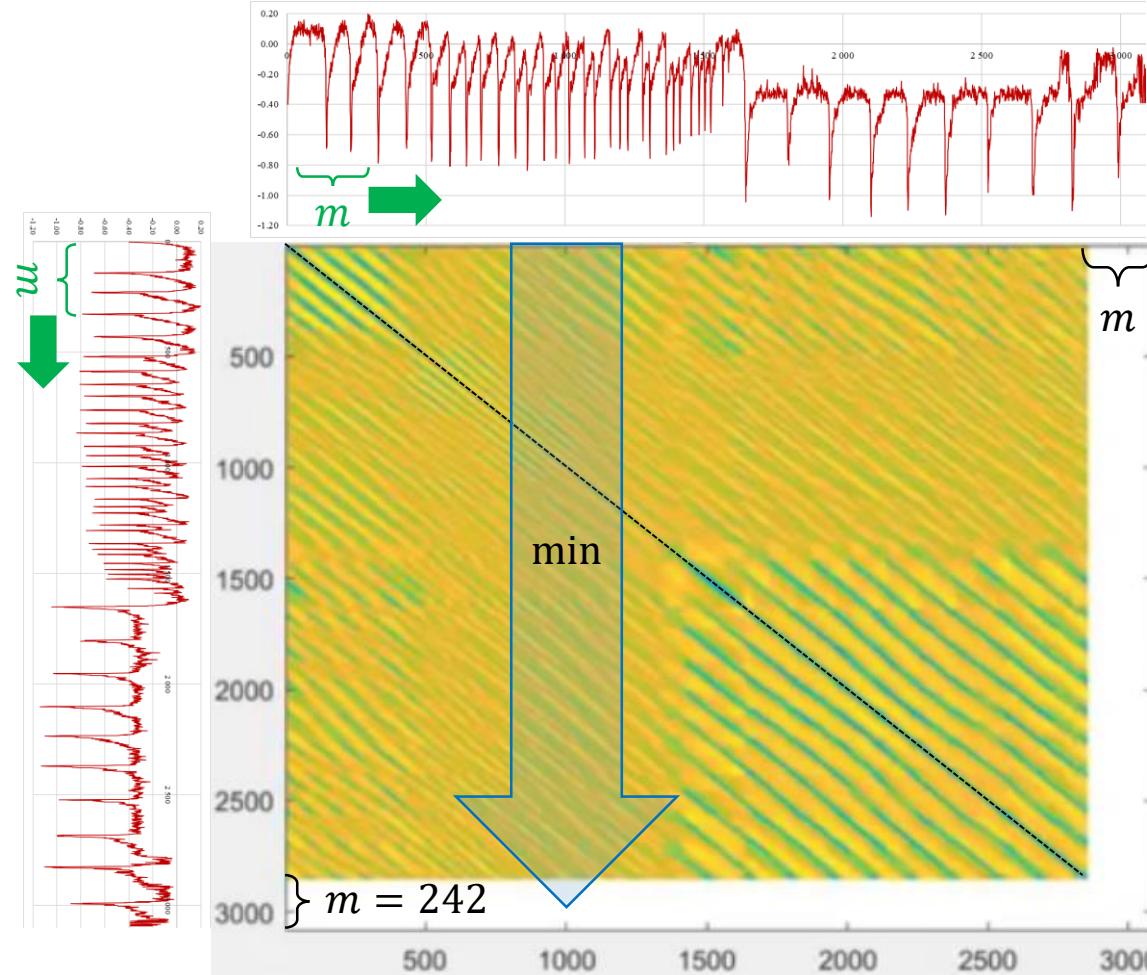
## Матрица расстояний

$$DistMatr_T^m \in \mathbb{R}^{(n-m+1) \times (n-m+1)}$$

$$DistMatr_T^m(i, j) = \text{Dist}(T_{i,m}, T_{j,m})$$

$$MP_T^m(i) = \min_{\substack{1 \leq j \leq n-m+1 \\ |i-j|>m}} DistMatr(j, i)$$

# Почему профиль матричный?



## Матрица расстояний

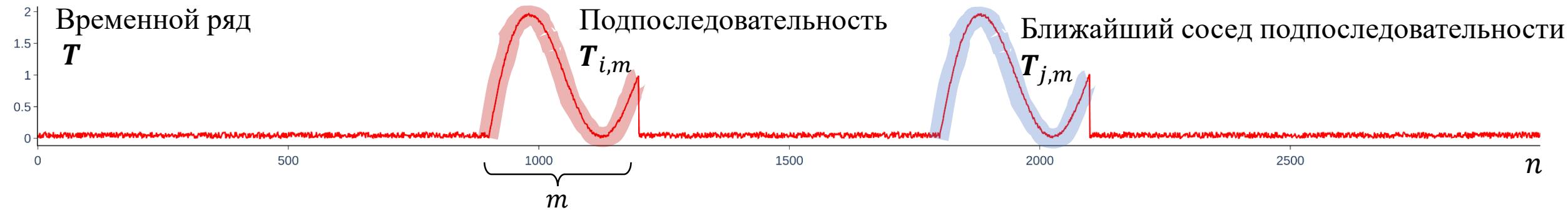
$$DistMatr_T^m \in \mathbb{R}^{(n-m+1) \times (n-m+1)}$$

$$DistMatr_T^m(i, j) = \text{Dist}(T_{i,m}, T_{j,m})$$

$$MP_T^m(i) = \min_{\substack{1 \leq j \leq n-m+1 \\ |i-j|>m}} DistMatr(j, i)$$

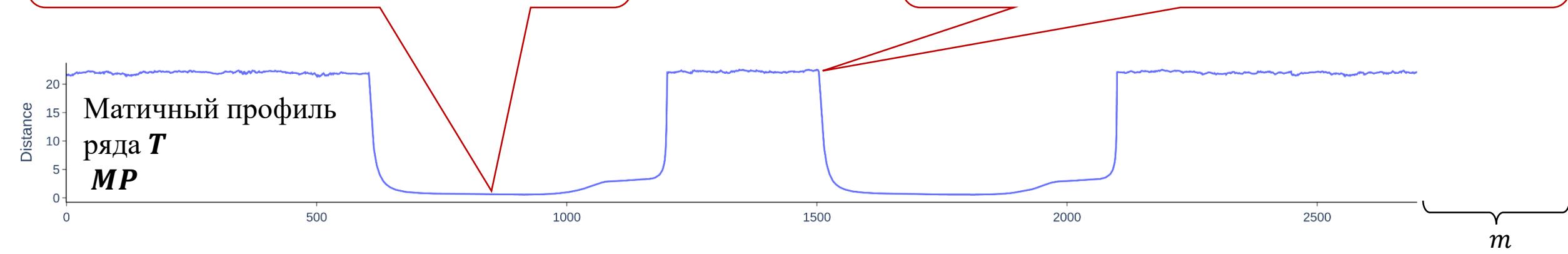
**Получение МП через матрицу расстояний  
ВЫЧИСЛИТЕЛЬНО НЕ ЭФФЕКТИВНО  
и ЗАТРАТНО ПО ПАМЯТИ (4 Тб для  $n = 10^6$ )**

# Простое понимание матричного профиля

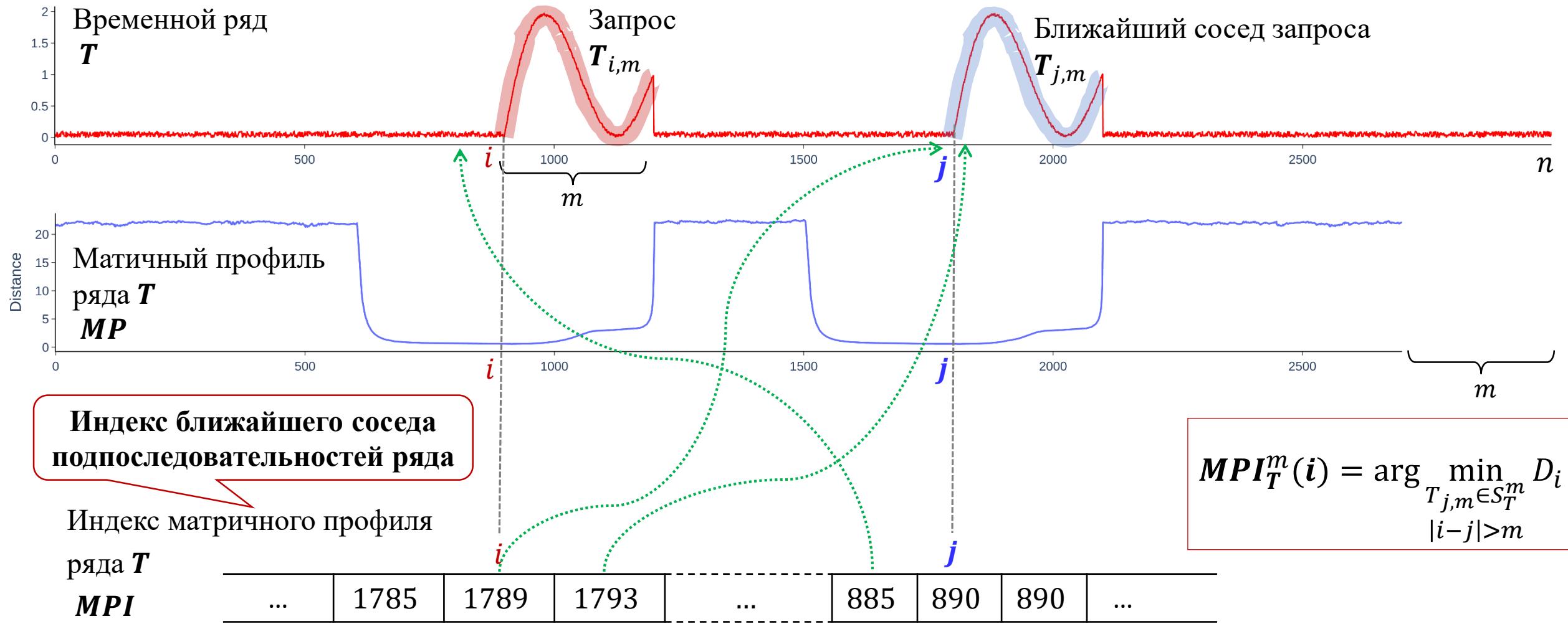


Локальные минимумы МП  
соответствуют мотивам (шаблонам) ряда

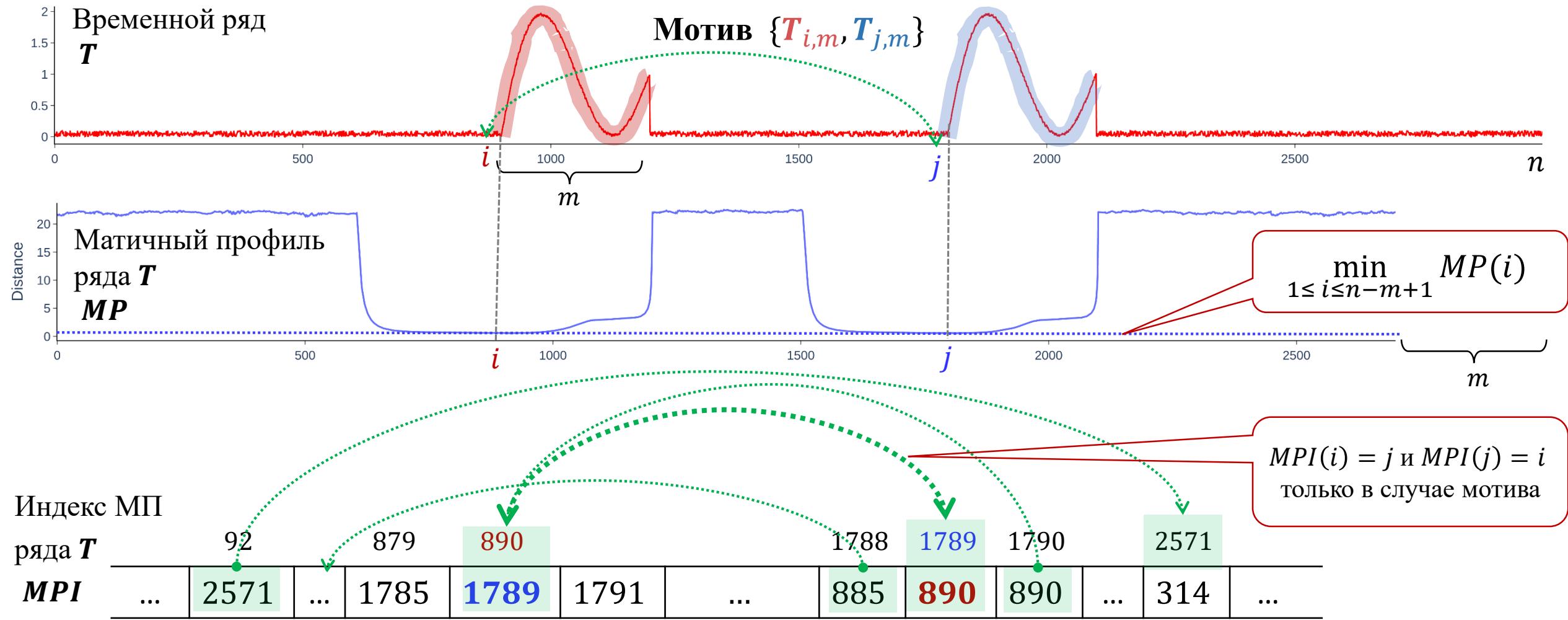
Локальные максимумы МП  
соответствуют диссонансам (аномалиям) ряда



# Индекс матричного профиля



# Индекс МП не симметричен в общем случае



# Функция $\text{Dist}(\cdot, \cdot)$ для матричного профиля

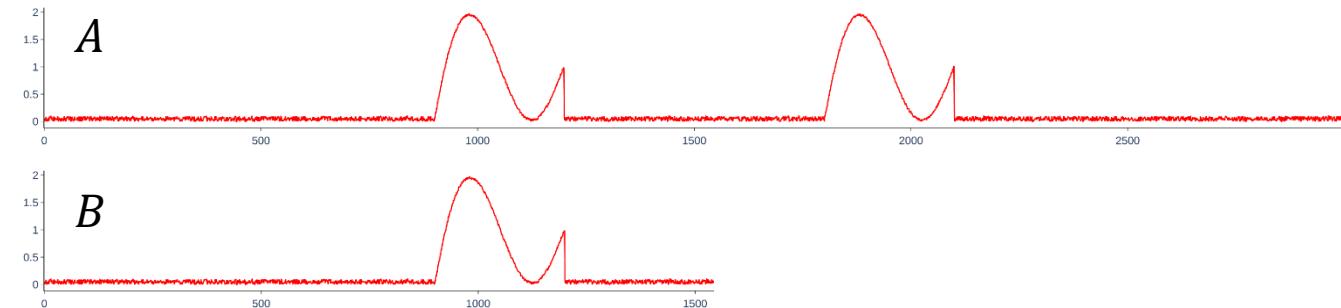
- ED
- $\text{ED}^2$
- $\text{ED}_{\text{norm}}$
- $\text{ED}_{\text{norm}}^2$
- DTW
- Hamming
- ...

Можно использовать любую функцию расстояния (метрику или не-метрику), это вопрос двух факторов:

- релевантность функции предметной области
- сложность (быстрота) вычисления МП

# Обобщение: Матричный профиль соединения (Join MP)

- Ряды  
 $A, |A| = n_A, B, |B| = n_B$



- Профиль расстояния

$$D_i = \{\text{Dist}(A_{i,m}, B_{j,m})\}_{j=1}^{n_B-m+1}, \quad 1 \leq i \leq n_A - m + 1$$

- МП соединения

$$MPjoin_{AB}^m(i) = \min_{1 \leq j \leq n_B - m + 1} D_i(j)$$

- Индекс МП соединения

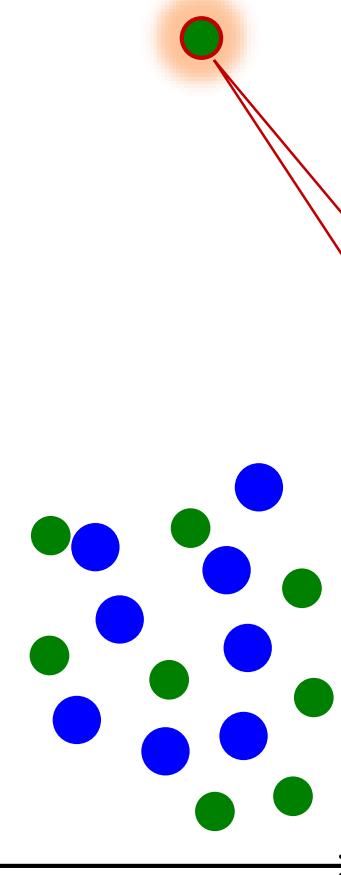
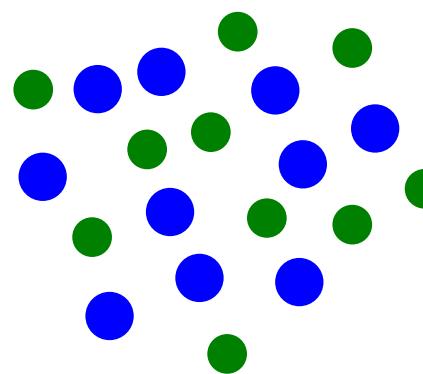
$$MPIjoin_{AB}^m(i) = \arg \min_{1 \leq j \leq n_B - m + 1} D_i(j)$$

Нет условия недопустимости  
тривиального совпадения

В общем случае  
построение МП соединения –  
не коммутативная операция:

$$MPjoin_{AB}^m \neq MPjoin_{BA}^m$$

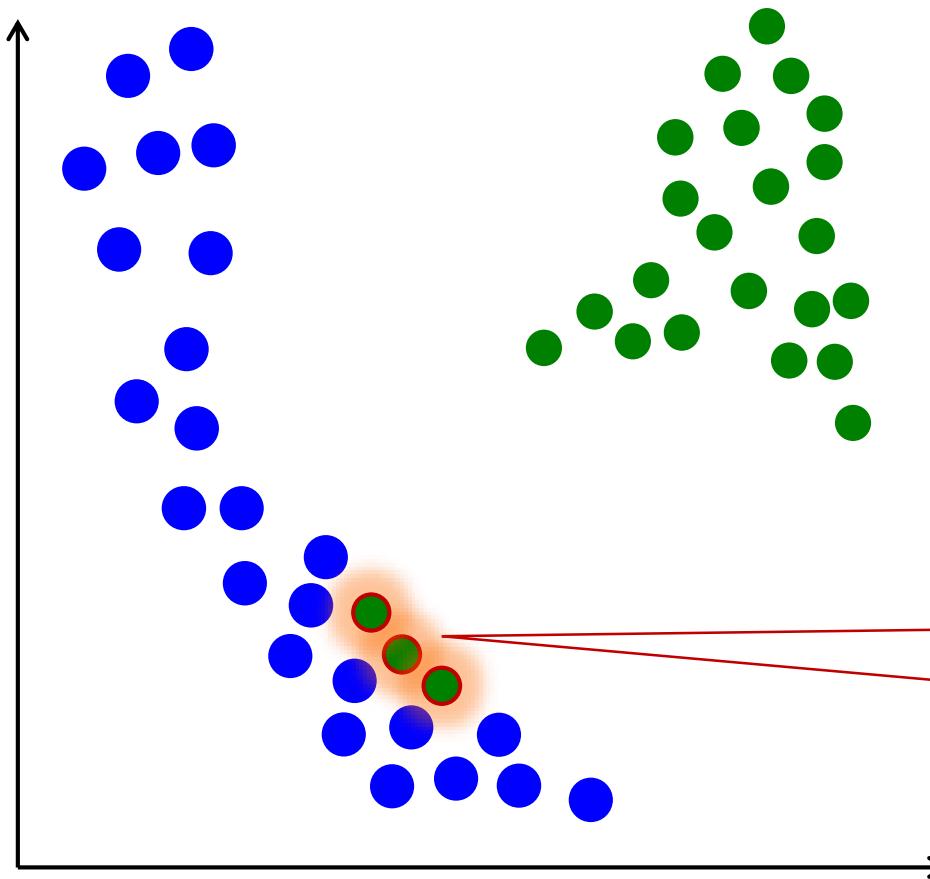
# Простое понимание МП соединения: диссонансы



- Мы предполагаем, что временные ряды  $A$  и  $B$  существенно похожи
- При соединении  $A$  и  $B$  мы находим подпоследовательность, которая встречается только в  $A$ , но отсутствует в  $B$

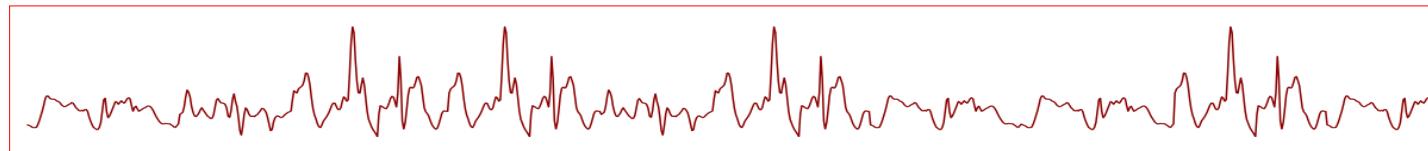
Диссонанс  
соединения

# Простое понимание МП соединения: мотивы



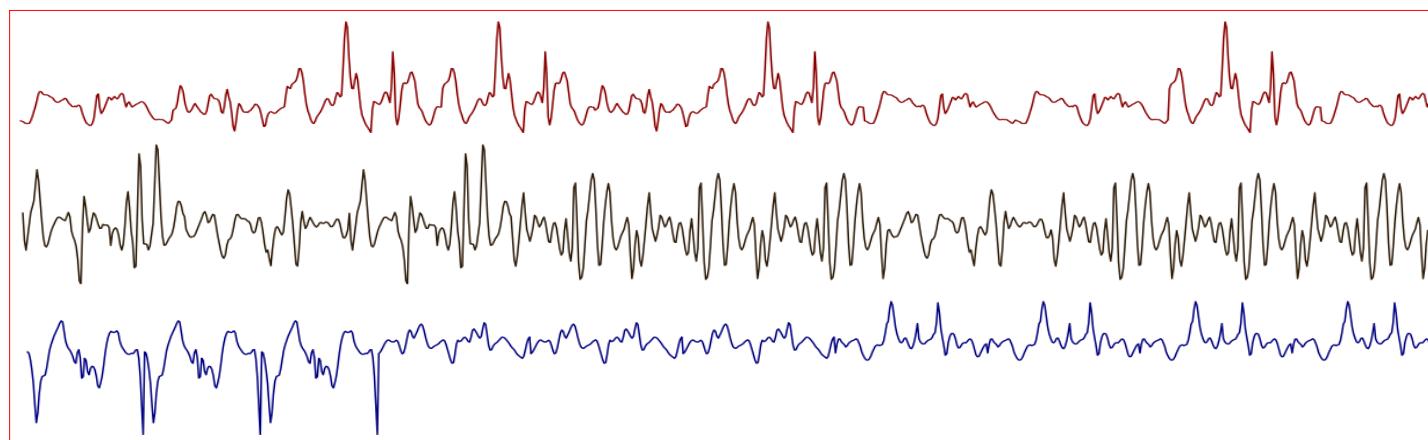
- Мы предполагаем, что временные ряды  $A$  и  $B$  существенно отличаются
- При соединении  $A$  и  $B$  мы находим подпоследовательности из  $A$ , которые встречаются в  $B$

# Матричный профиль многомерного ряда

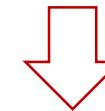
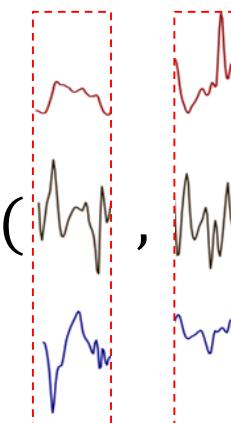


$\text{Dist}(\textcolor{red}{\cdot}, \textcolor{red}{\cdot})$

$\text{ED, ED}^2,$   
 $\text{ED}_{\text{norm}}, \text{ED}_{\text{norm}}^2,$   
 $\text{DTW, ...}$



$\text{Dist}(\textcolor{brown}{\cdot}, \textcolor{brown}{\cdot})$



**Агрегация**  
 $\text{Dist}(\textcolor{brown}{\cdot}, \textcolor{brown}{\cdot})$   
 $\text{Dist}(\textcolor{brown}{\cdot}, \textcolor{brown}{\cdot})$   
 $\text{Dist}(\textcolor{blue}{\cdot}, \textcolor{blue}{\cdot})$

$\text{median}(\{\text{ED}_{\text{norm}}^2(\cdot, \cdot)\}_{i=1}^d)$

# Содержание

- Понятие матричного профиля
- **Примеры задач, решаемых на основе матричного профиля**
- Алгоритмы вычисления матричного профиля

# Матричный профиль – базис для решения большинства задач интеллектуального анализа временных рядов\*



- Диссонансы
- Мотивы
- Шейплеты
- Сниппеты
- Цепочки
- Сравнение рядов
- ...



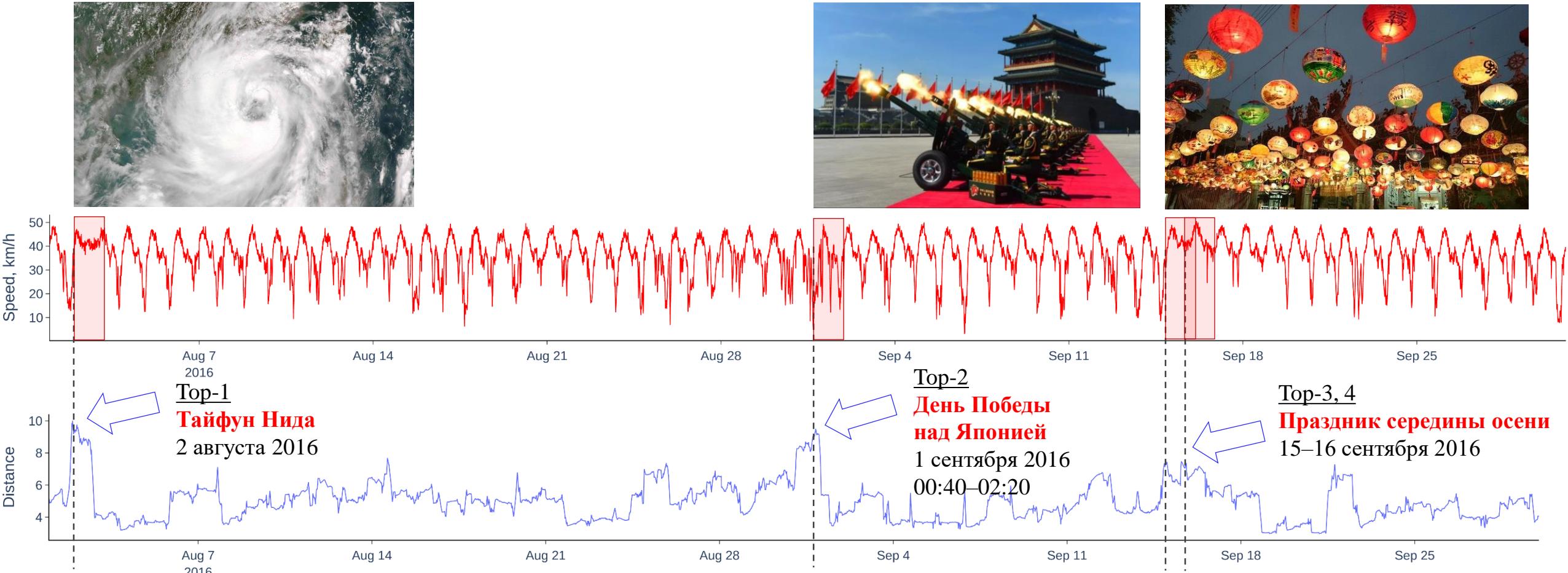
Имонн Кеог  
(Калифорнийский  
университет  
в Риверсайде, США)

[Eamonn Keogh](#)  
(University  
of California, Riverside,  
USA)

\* Zhu Y. 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. Discov. 34(4): 949-979 (2020). DOI: [10.1007/s10618-019-00668-6](https://doi.org/10.1007/s10618-019-00668-6).

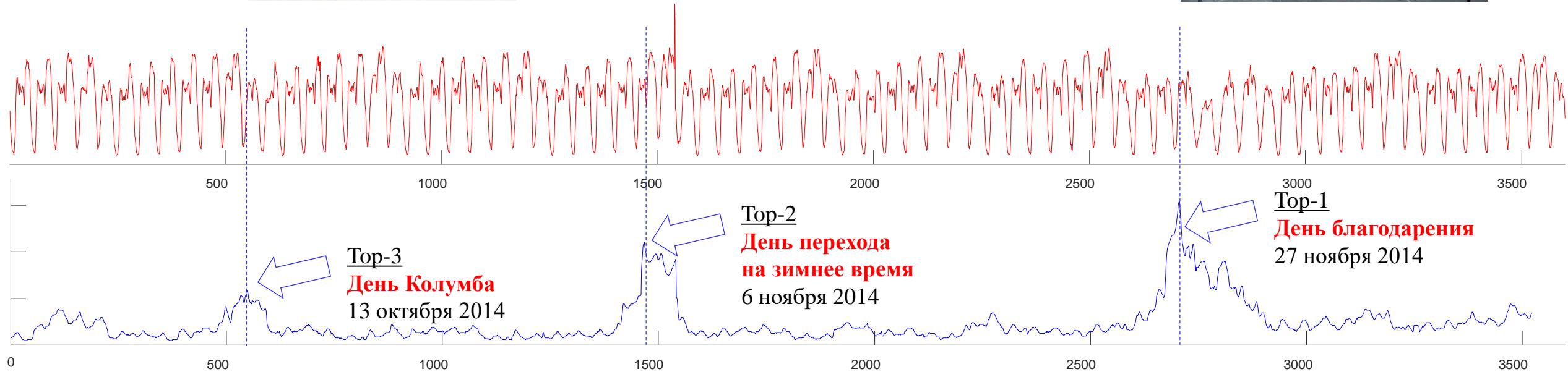
# Поиск аномалий

## Скорость городского трафика Гуанчжоу\*



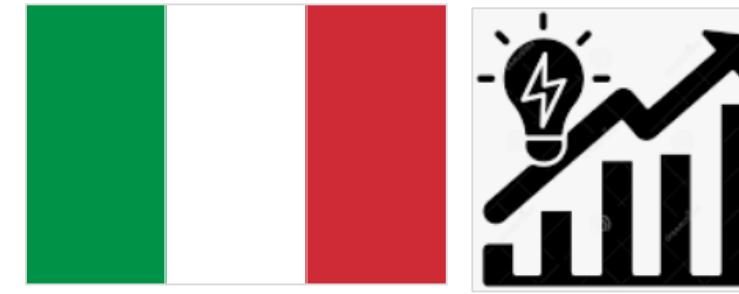
\* Chen X., Chen Y., He Z. Urban traffic speed dataset of Guangzhou, China. 2018. DOI: [10.5281/zenodo.1205229](https://doi.org/10.5281/zenodo.1205229).

# Поиск аномалий

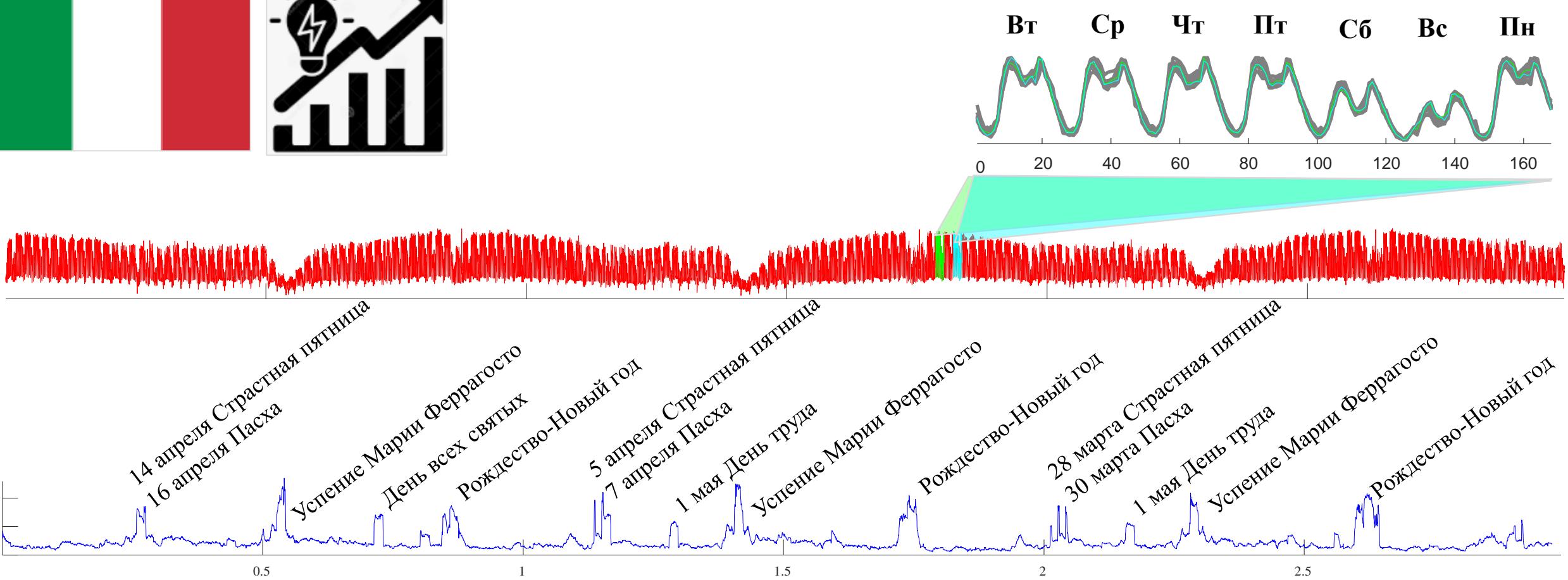


\* 2014 New York City Taxi Trips. URL: <https://www.kaggle.com/datasets/kentonlp/2014-new-york-city-taxi-trips>.

# Поиск мотивов

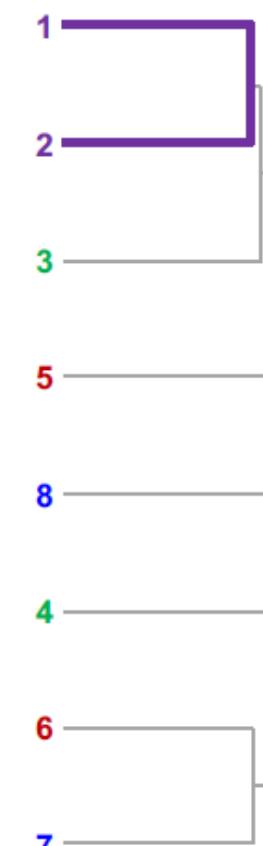


Энергопотребление в Италии 1995-1998 гг.



# Сравнение временных рядов

Евклидово  
расстояние

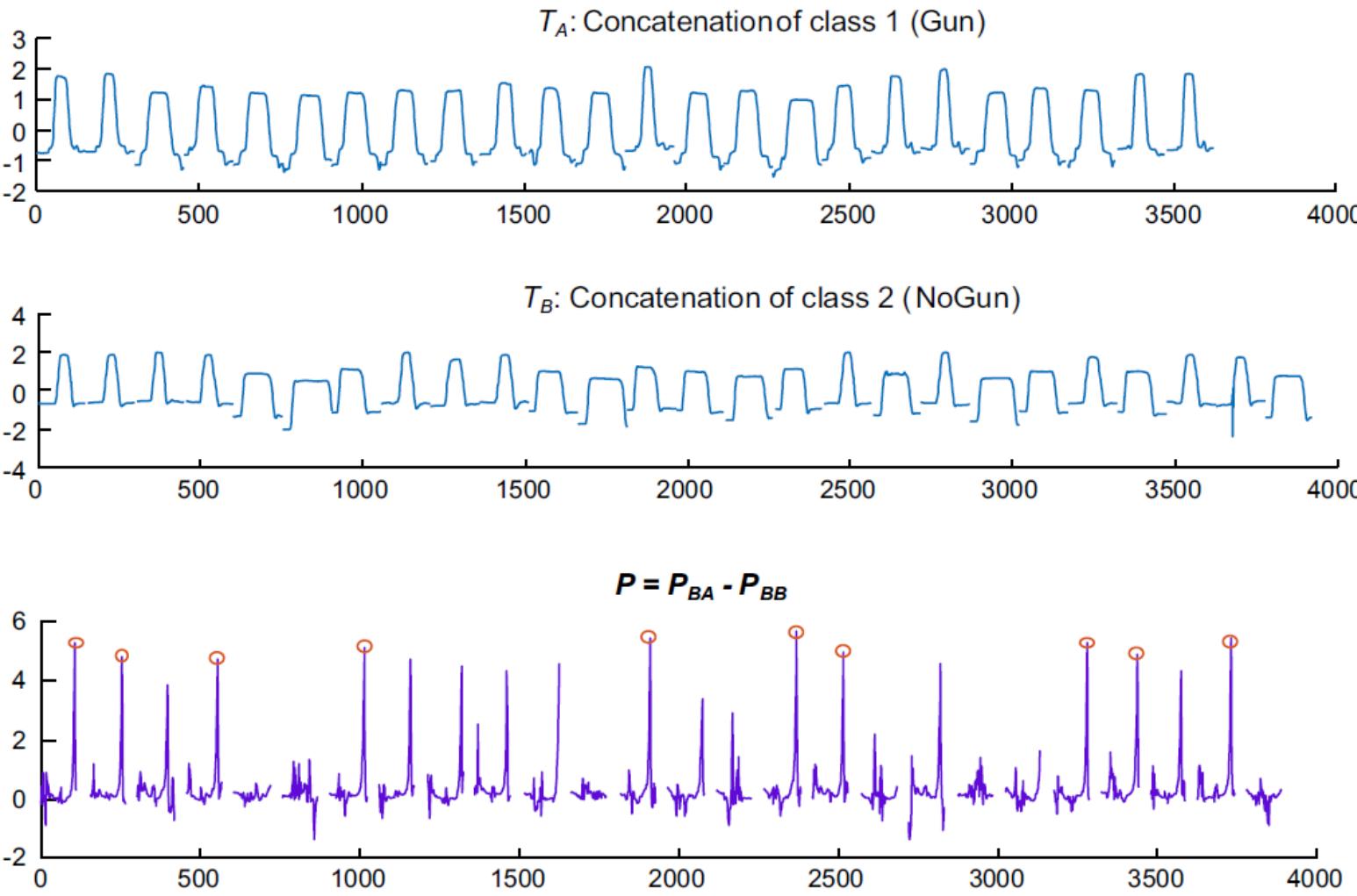


**MSMD**  
(Min Shared  
Motif Distance)

- Сравнение 10 с фрагментов ЭКГ четырех пациентов
- Кластеризация на основе евклидова расстояния дает не адекватные итоги, в отличие от применения матричного профиля

$$\{P, I\} := MPjoin(A, B, m)$$
$$MSMD := \min(P)$$

# Поиск шейплетов (shapelet)



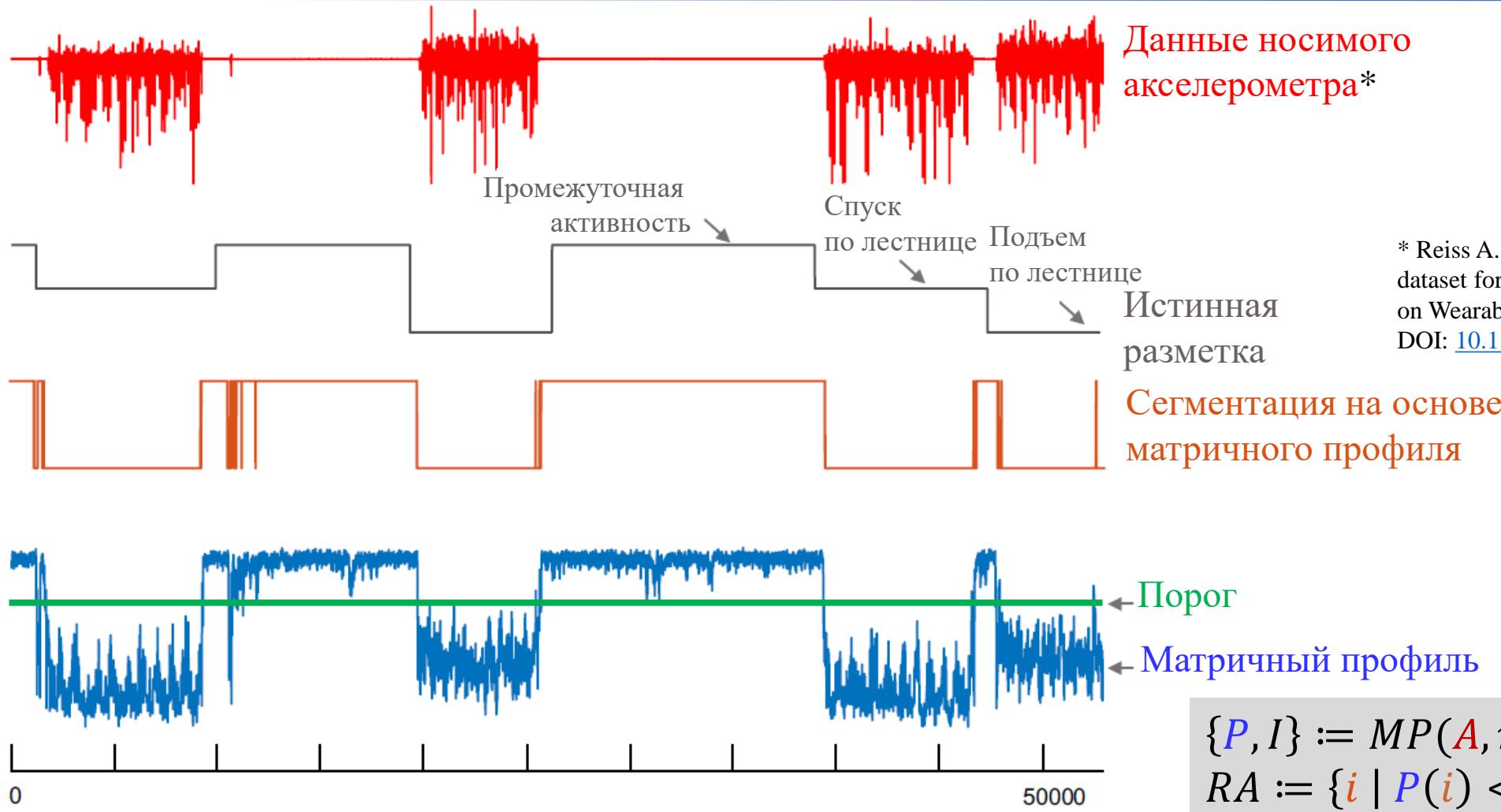
- Шейплет – лучший (наиболее репрезентативный) представитель класса подпоследовательностей (рядов)
- $A$  и  $B$  – классы подпоследовательностей,  $T_A$  и  $T_B$  – ряды, полученные склейкой подпоследовательностей из своих классов (NaN разделяет каждую пару)
- Шейплеты – подпоследовательности, дающие топ- $k$  максимумы в матричном профиле разницы  $P = P_{BA} - P_{BB}$

```

 $\{P_{BB}, Ibb\} := MPjoin(B, B, m)$ 
 $\{P_{BA}, Iba\} := MPjoin(B, A, m)$ 
 $P := P_{BA} - P_{BB}$ 
 $TopKshapelets := TopMax(P, k)$ 

```

# Сегментация повторяющихся активностей



\* Reiss A., Stricker D. Introducing a new benchmarked dataset for activity monitoring. Proc. of the 16th Int. Symp. on Wearable Computers (ISWC). 2012.  
DOI: [10.1109/ISWC.2012.13](https://doi.org/10.1109/ISWC.2012.13).

$$\begin{aligned} \{P, I\} &:= MP(A, m) \\ RA &:= \{i \mid P(i) < \alpha \cdot (\min(P) + \max(P))\} \end{aligned}$$

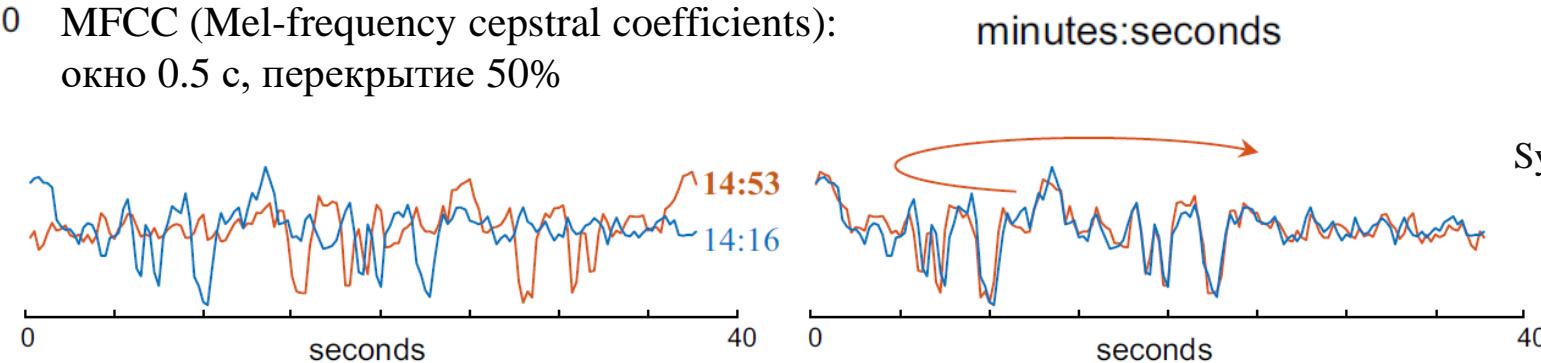
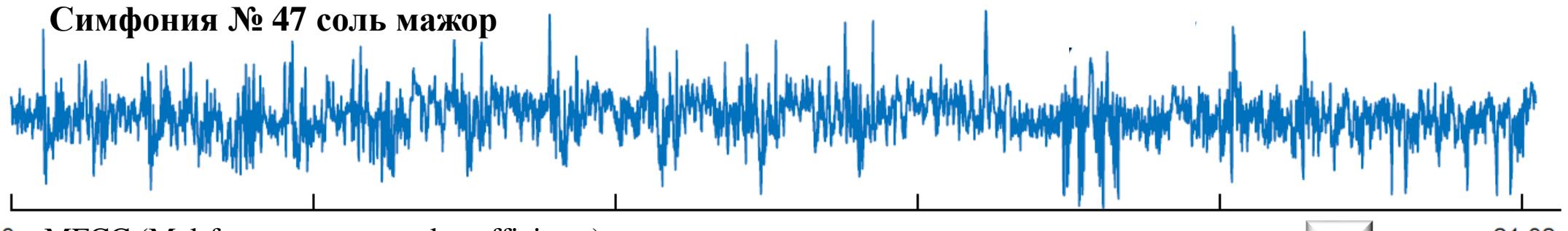
# Поиск перевертышей (semordnilap: god↔dog, lived↔devil, ...)



**Йозеф Гайдн**  
(Joseph Haydn)  
1732-1809

**Найденный  
мотив**

**Ноты  
мотива**

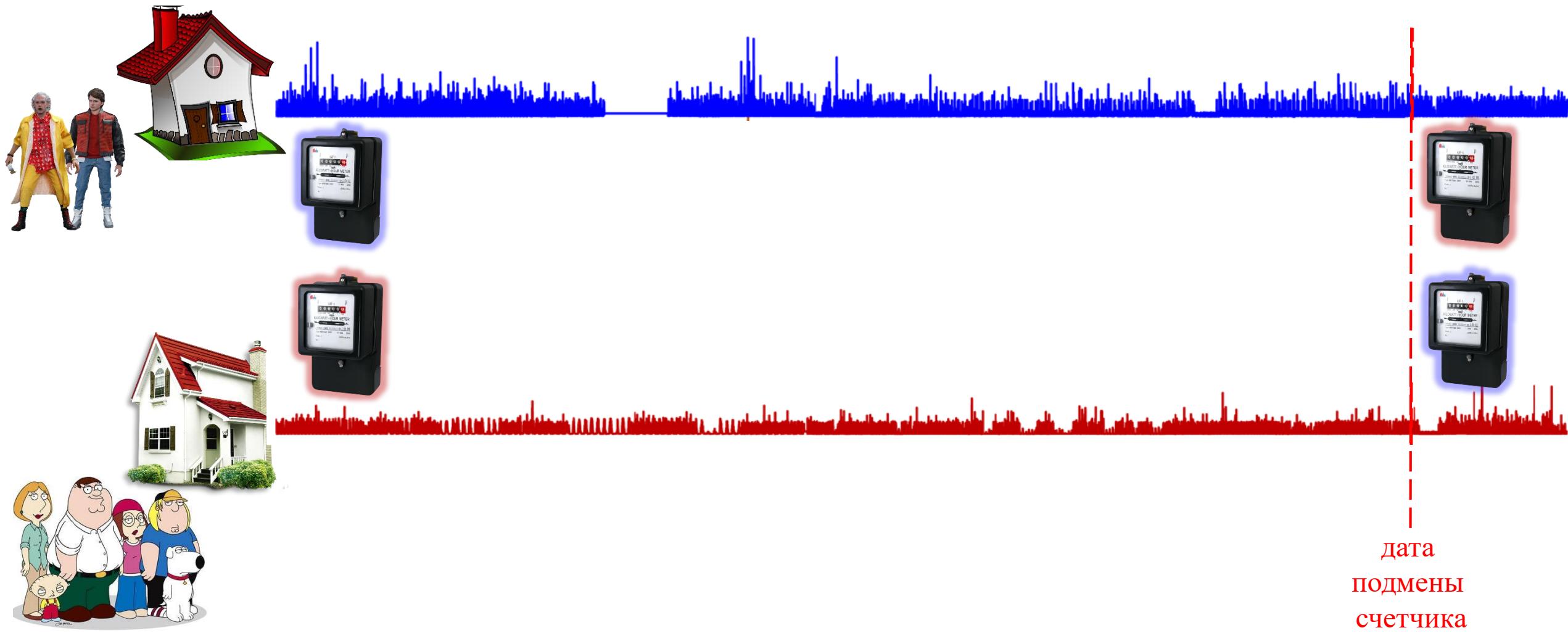


Joseph Haydn,  
Symphony No. 47 in G major “Palindrome”,  
directed by Bruno Weil

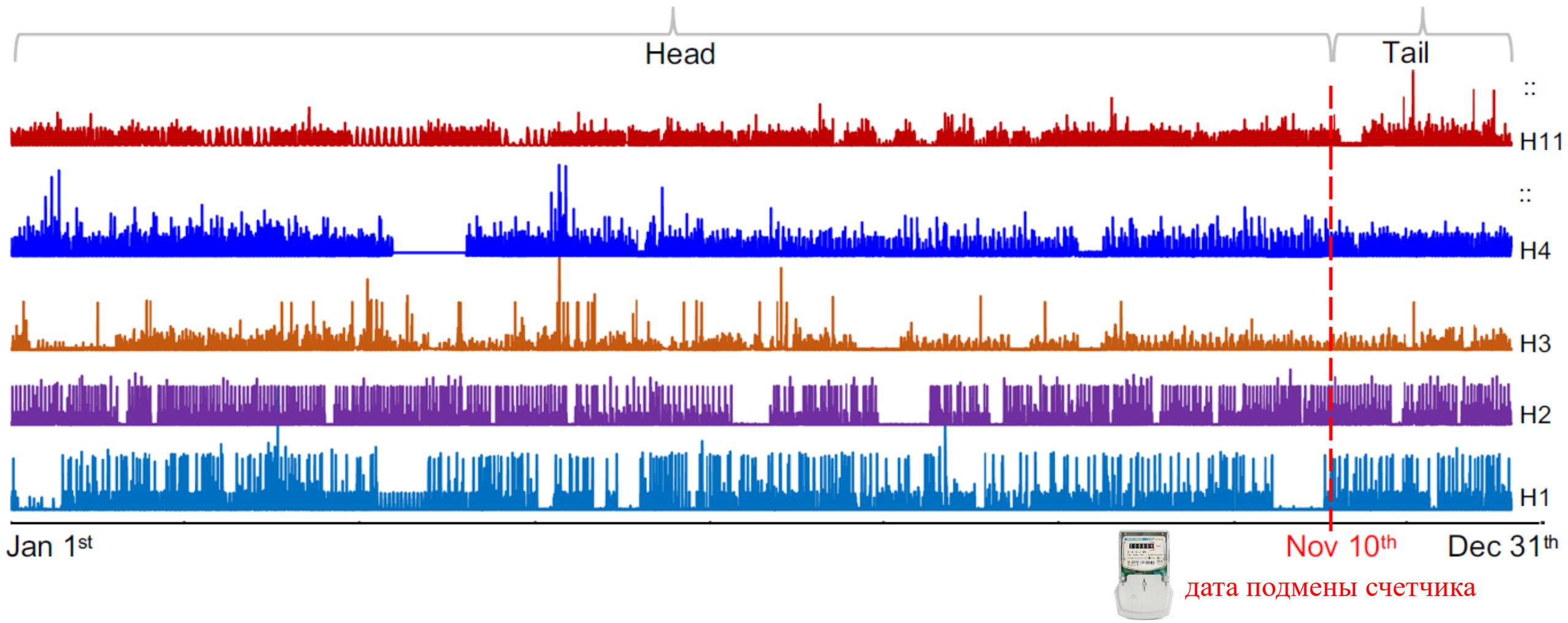


$cisum := Reverse(music)$   
 $\{P, I\} := MPjoin(music, cisum, 150) // 37.5 \text{ с}$

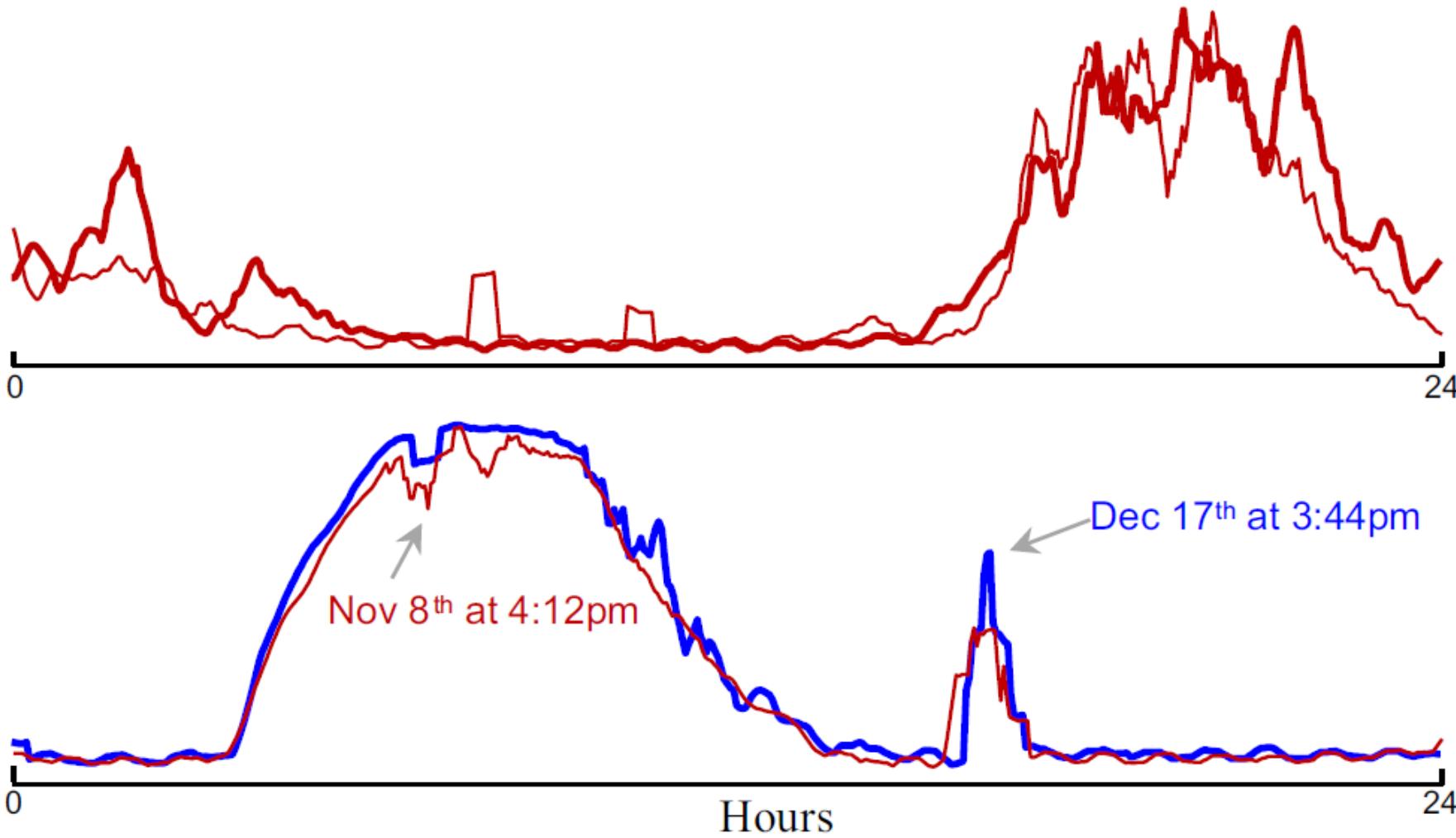
# Раскрытие кражи электричества подменой счетчиков (meter-swapping)



# Раскрытие кражи электричества подменой счетчиков (meter-swapping)



# Раскрытие кражи электричества подменой счетчиков (meter-swapping)



Top-1 мотив  
 $MPjoin(Head(H_{11}), Tail(H_{11}))$

$ED(left, right) = 9.56$   
(менее среднего расстояния  
по всем домам)

Top-1 мотив  
 $MPjoin(Head(H_{11}), Tail(H_4))$

$ED(left, right) = 2.85$

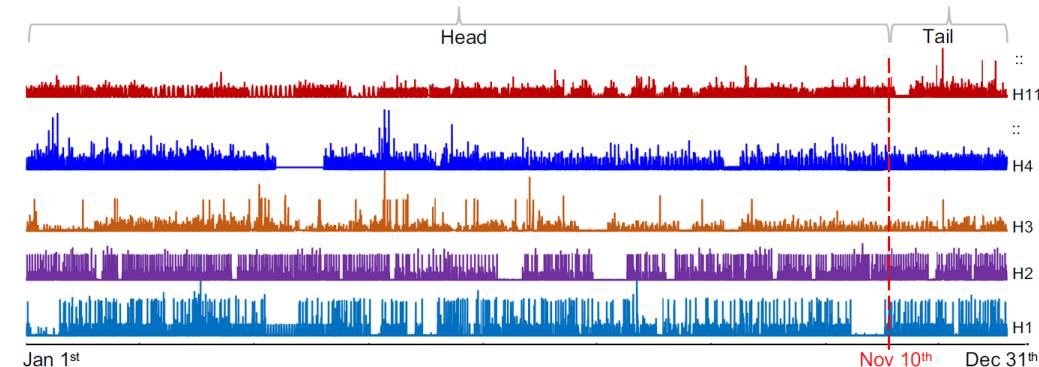
# Раскрытие кражи электричества подменой счетчиков (meter-swapping)

$$SwapScore(H_i, H_j) = \frac{\min MPjoin(Head(H_i), Tail(H_j))}{\min MPjoin(Head(H_i), Tail(H_i)) + \varepsilon}$$

```

minScore := +∞
for i := 1 to NumHouse do
  {P, I} := MPjoin(Head(Hi), Tail(Hi), m)
  minP := min(P)
  for j := i + 1 to NumHouse do
    {J, JI} := MPjoin(Head(Hi), Tail(Hj), m)
    SwapScore := min(J)/(minP + ε)
    if SwapScore < minScore then
      minScore := SwapScore
      suspect := {Hi, Hj}

```

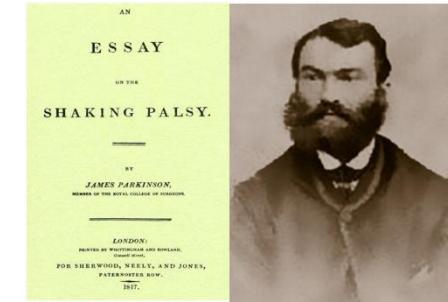


*suspect = {H<sub>11</sub>, H<sub>4</sub>}*

top-1 мотив  
*Head(H<sub>11</sub>), Tail(H<sub>4</sub>)*

Кражи со стороны H4

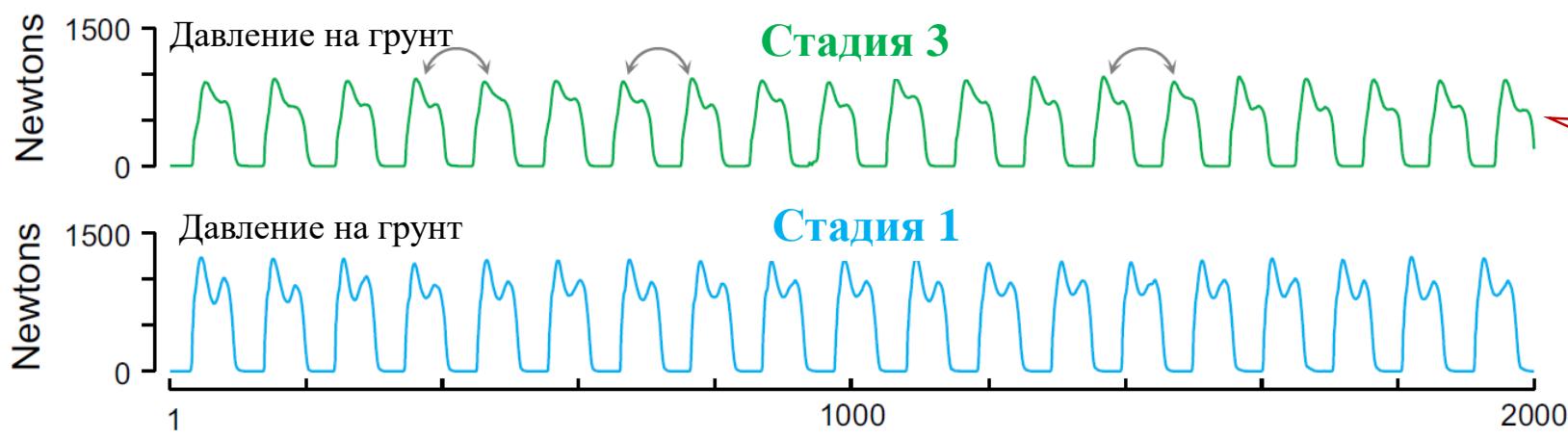
# Оценка тяжести болезни Паркинсона



**Джеймс Паркинсон**  
(James Parkinson)  
1755-1824

Шкала Хён—Яра  
(Hoehn M.M., Yahr M.D.)

Стадия	Симптоматика
0	Нет признаков заболевания
1	Проявления на одной из конечностей
1.5	Проявления на одной из конечностей и туловище
2	Двусторонние проявления без постуральной неустойчивости
2.5	Двусторонние проявления с постуральной неустойчивостью
3	Двусторонние проявления. Постуральная неустойчивость. Способность к самообслуживанию
4	Обездвиженность, потребность в посторонней помощи. Способность ходить и/или стоять без поддержки
5	Обездвиженность, инвалидизация



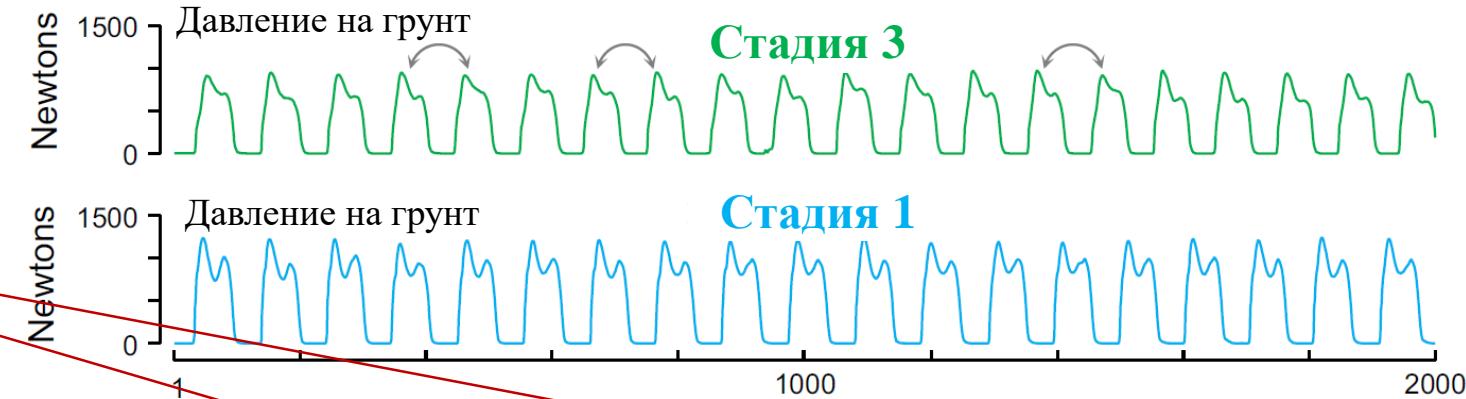
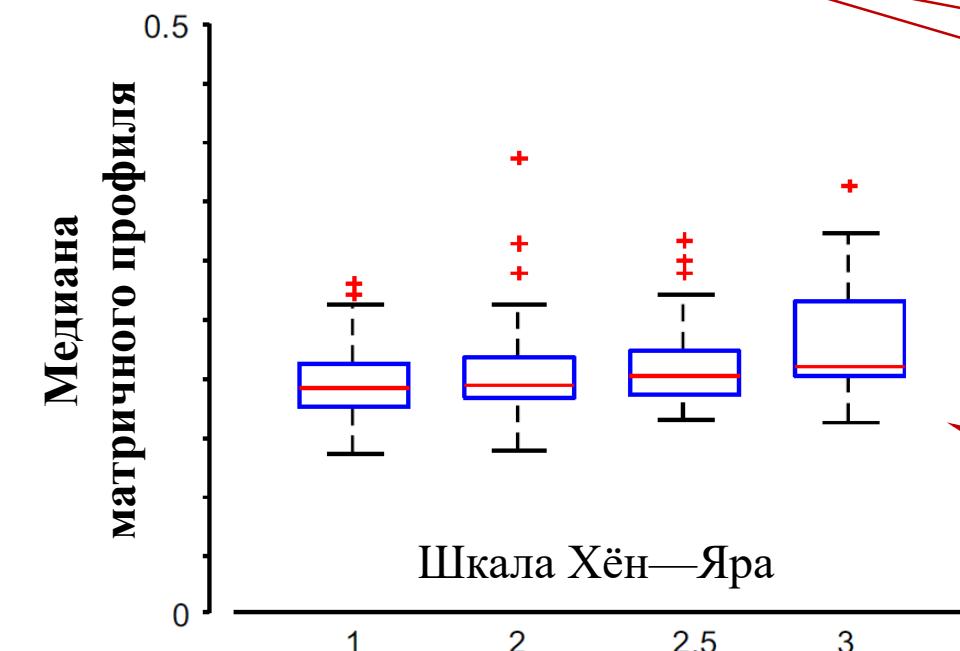
При нарушениях  
двигательной активности  
циклы походки  
повторяются не идеально

# Оценка тяжести болезни Паркинсона

```

for  $i := 1$  to  $NumPatients$  do
     $\{P, I\} := MP(PatientGait(i), m)$ 
     $HoehnYahr(i) := \text{median}(P)$ 

```



Нет отличий между соседями на ранних стадиях болезни,  
сильные отличия на поздних стадиях.  
Можно взять медиану матричного профиля в качестве индикатора

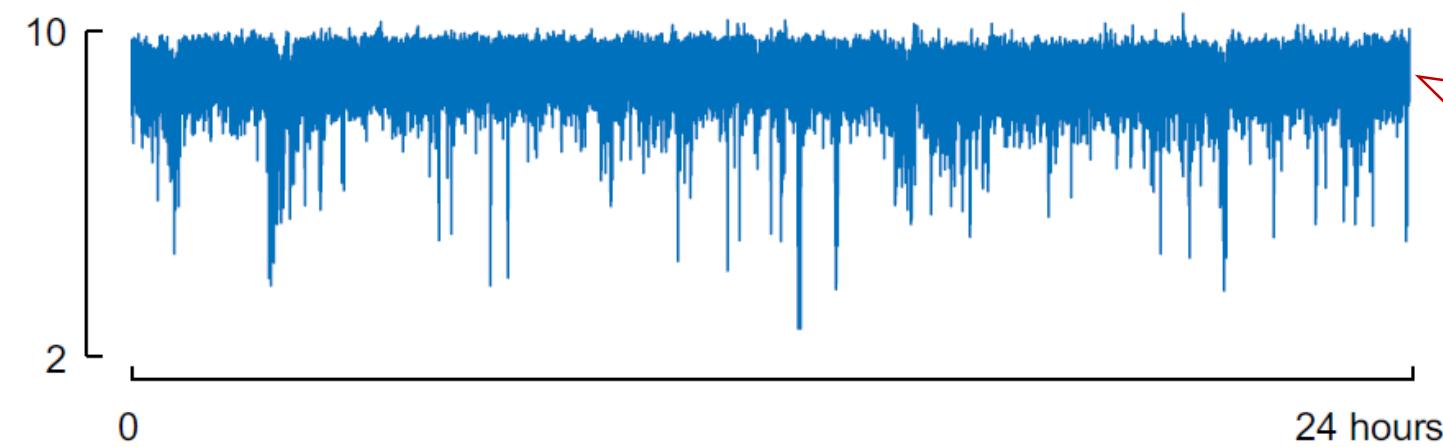
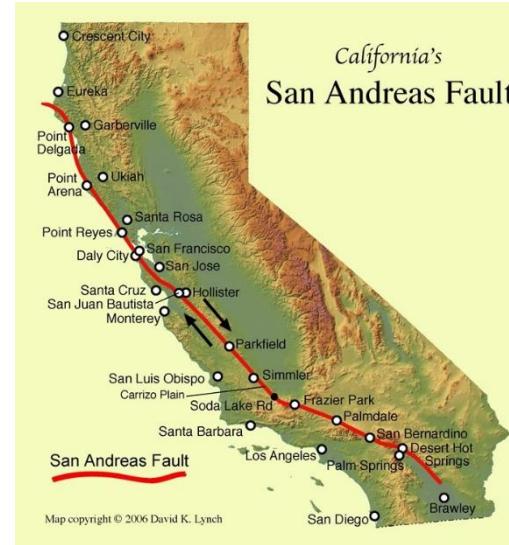
Эксперимент на наборе PhysioBank\*  
(93 ряда: 73 – стадия 1, 20 – стадии 2, 2.5, 3)  
Медиана матричного профиля увеличивается на поздних стадиях

\* Goldberger A.L., et al. PhysioBank, PhysioToolkit, and PhysioNet: Components of a new research resource for complex physiologic signals. Circulation. 2000. 101(23), e215–e220. DOI: [10.1161/01.cir.101.23.e215](https://doi.org/10.1161/01.cir.101.23.e215)

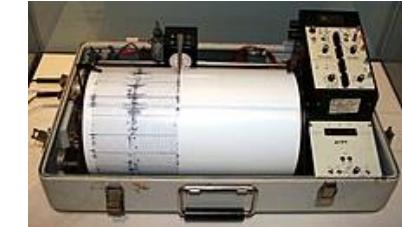
# Обнаружение низкочастотных землетрясений (LFE, low-frequency earthquake)



Разлом Сан-Андреас,  
Калифорния (США)



- Станции **FROB** и **JCNB** в 10 км друг от друга снимают показания сейсмографа возле разлома Сан-Андреас (частота 20 Гц, 1.728 млн. точек за сутки)



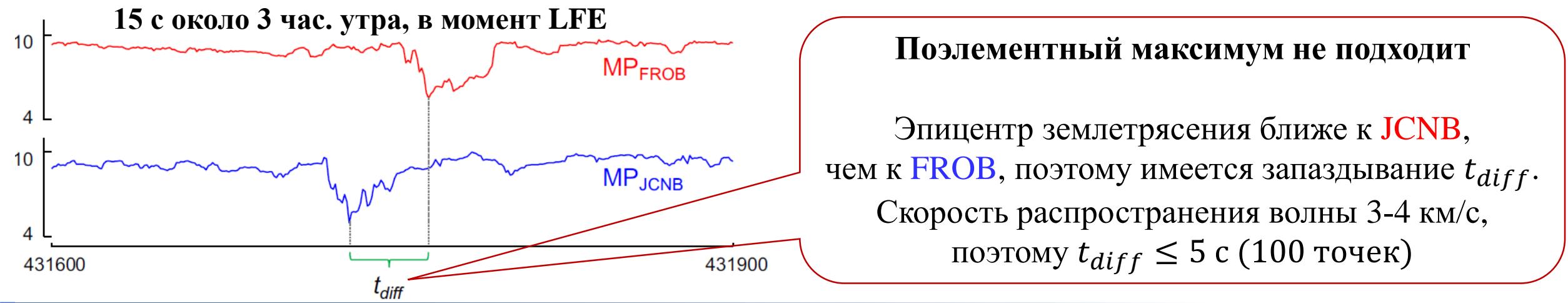
- Как автоматически фильтровать ложные LFE?**

Матричный профиль суточной записи  
сейсмографа 9 октября 2007  
на станции **FROB**

Лишь 10% «впадин» –  
истинные землетрясения

# Обнаружение низкочастотных землетрясений (LFE, low-frequency earthquake)

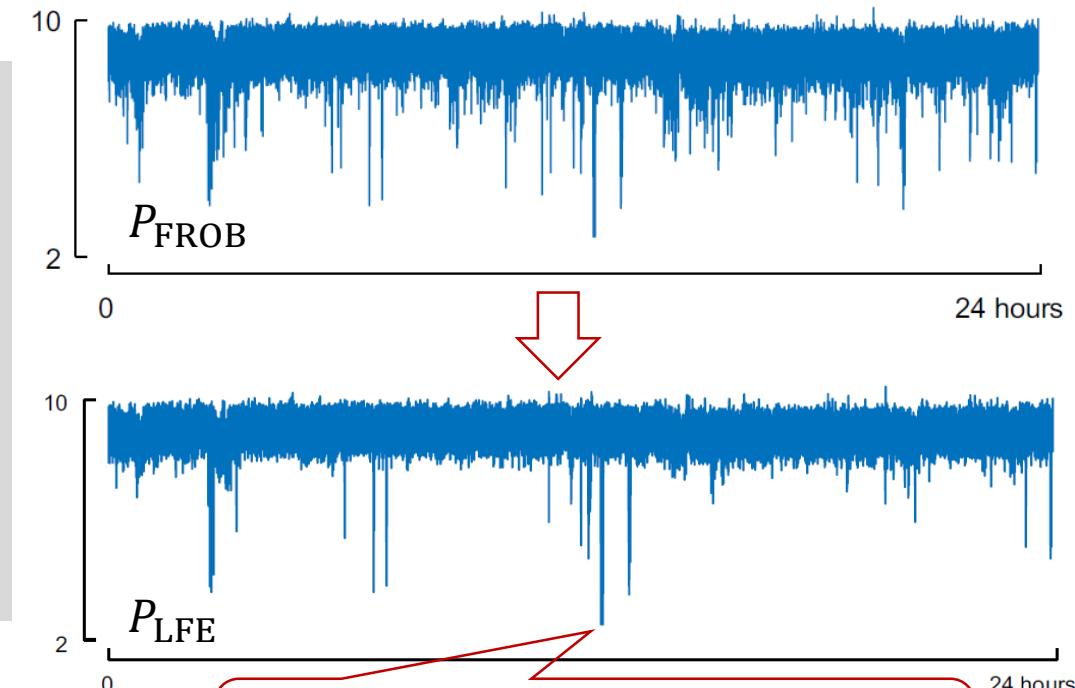
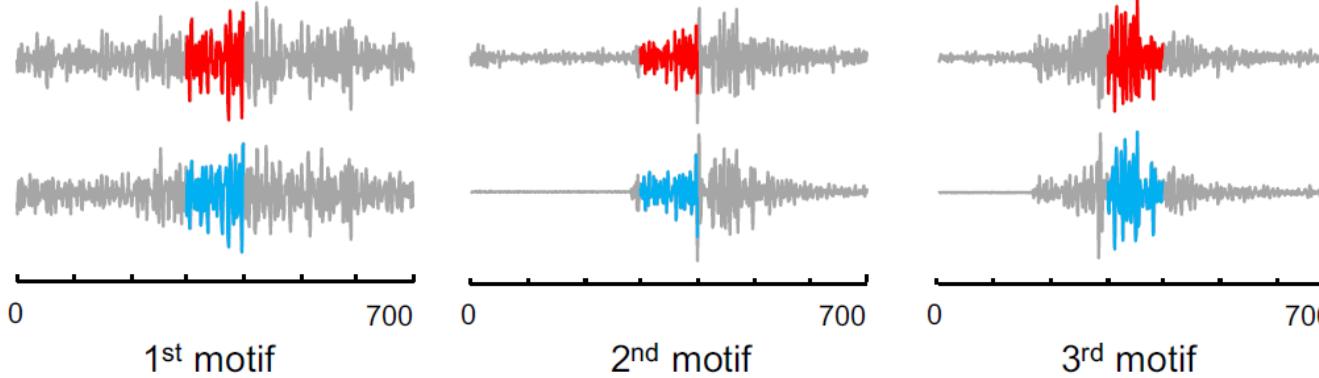
- Шумы в сейсмографе локальны, но LFE обнаруживается им в близкие (но не идентичные) моменты времени
- В момент истинного LFE матричные профили разных станций показывают низкие значения. Наоборот, при ложном LFE *один* профиль покажет низкие значения, остальные – высокие
- Для фильтрации возьмем поэлементный максимум матричного профиля (?)



# Обнаружение низкочастотных землетрясений (LFE, low-frequency earthquake)

```

 $\{P_{FROB}, I_{FROB}\} := MP(FROB, m)$ 
 $\{P_{JCNB}, I_{JCNB}\} := MP(JCNB, m)$ 
for  $i := 1$  to  $|FROB|$  do
     $minVal := \min_{\max(i-100, 1) \leq k \leq \min(i+100, |JCNB|)} P_{JCNB}(k)$ 
     $minIdx := \arg \min_{\max(i-100, 1) \leq k \leq \min(i+100, |JCNB|)} P_{JCNB}(k)$ 
     $P_{LFE}(i) := \max(P_{FROB}(i), minVal)$ 
     $I_{LFE}(i) := minIdx$ 
  
```



Матричный профиль  
с истинными землетрясениями

Топ-3 мотива, найденных по очищенному  
матричному профилю (истинные LFE)

# Содержание

- Понятие матричного профиля
- Примеры задач, решаемых на основе матричного профиля
- **Алгоритмы вычисления матричного профиля**
  - STAMP
  - STOMPR
  - SCRIMP, PreSCRIMP, SCRIMP++

# Алгоритм STAMP (Scalable Time series Anytime Matrix Profile<sup>\*</sup>)

## Algorithm STAMP

**Input:** ряды  $A, B$ , длина подп-ти  $m$

**Output:** МП  $P_{AB}$ , индекс МП  $I_{AB}$

**Variables:**  $D \in \mathbb{R}^{n_A-m+1}$  профиль расст-я

$P_{AB} := \overline{+\infty}; I_{AB} := \bar{0}$

**for**  $j := 1$  **to**  $n_B - m + 1$  **do**

$D := MASS(B_{j,m}, A)$

**for**  $i := 1$  **to**  $n_A - m + 1$  **do**

**if**  $D(i) \leq P_{AB}(i)$  **then**

$P_{AB}(i) := D(i)$

$I_{AB}(i) := j$

**return**  $P_{AB}, I_{AB}$

\* Yeh C.M., Zhu Y., Ulanova L., Begum N., Ding Y., Dau H.A., Silva D.F., Mueen A., Keogh E.J. Matrix Profile I: All pairs similarity joins for time series: A unifying view that includes motifs, discords and shapelets. Proc. of the IEEE 16th Int. Conf. on Data Mining, ICDM 2016, Barcelona, Spain, 12–15 December 2016. pp. 1317–1322. <https://doi.org/10.1109/ICDM.2016.0179>

# Алгоритм STAMP (Scalable Time series Anytime Matrix Profile)

## Algorithm STAMP

**Input:** ряды  $A, B$ , длина подп-ти  $m$

**Output:** МП  $P_{AB}$ , индекс МП  $I_{AB}$

**Variables:**  $D \in \mathbb{R}^{n_A - m + 1}$  профиль расст-я

$P_{AB} := \overline{+\infty}; I_{AB} := \bar{0}$

**for**  $j := 1$  **to**  $n_B - m + 1$  **do**

$D := MASS(B_{j,m}, A)$

**for**  $i := 1$  **to**  $n_A - m + 1$  **do**

**if**  $D(i) \leq P_{AB}(i)$  **then**

$P_{AB}(i) := D(i)$

$I_{AB}(i) := j$

**return**  $P_{AB}, I_{AB}$

Алгоритм MASS  
(Mueen's Algorithm for Similarity Search)  
вычисляет профиль  
z-нормализованного евклидова расстояния,  
имеет сложность  $O(n \log_2 n)$   
(см. лекцию «Поиск по образцу»)

Сложность  
STAMP

$O(n^2 \log_2 n)$

# Алгоритм STAMP (Scalable Time series Anytime Matrix Profile)

## Algorithm STAMP

**Input:** ряды  $A, B$ , длина подп-ти  $m$

**Output:** МП  $P_{AB}$ , индекс МП  $I_{AB}$

**Variables:**  $D \in \mathbb{R}^{n_A-m+1}$  профиль расст-я

$P_{AB} := \overline{+\infty}; I_{AB} := \bar{0}$

**for**  $j := 1$  **to**  $n_B - m + 1$  **do**

$D := MASS(B_{j,m}, A)$

**for**  $i := 1$  **to**  $n_A - m + 1$  **do**

**if**  $D(i) \leq P_{AB}(i)$  **then**

$P_{AB}(i) := D(i)$

$I_{AB}(i) := j$

**return**  $P_{AB}, I_{AB}$

Может быть реализовано  
как поэлементная (векторная) операция

*ElementWiseMin( $P_{AB}, I_{AB}, D, i$ )*

# Алгоритм STAMP (Scalable Time series **Anytime** Matrix Profile)

## Algorithm STAMP

**Input:** ряды  $A, B$ , длина подп-ти  $m$ ,  
флаг *CheckForUserInterrupt*  
**Output:** МП  $P_{AB}$ , индекс МП  $I_{AB}$   
**Variables:**  $D \in \mathbb{R}^{n_A-m+1}$  профиль расст-я

```
 $P_{AB} := +\infty; I_{AB} := \bar{0}$ 
for each  $j \in [1, n_B - m + 1]$  in random order do
     $D := MASS(B_{j,m}, A)$ 
    ElementWiseMin( $P_{AB}, I_{AB}, D, i$ )
    if CheckForUserInterrupt = TRUE then
        print “Приближенное решение”,  $P_{AB}, I_{AB}$ 
    if GetUserChoice = “Улучшить решение” then
        continue
    else
        break
```

# Алгоритм STAMP (Scalable Time series **Anytime** Matrix Profile)

## Algorithm STAMP

**Input:** ряды  $A, B$ , длина подп-ти  $m$ ,  
флаг *CheckForUserInterrupt*

**Output:** МП  $P_{AB}$ , индекс МП  $I_{AB}$

**Variables:**  $D \in \mathbb{R}^{n_A-m+1}$  профиль расст-я

$P_{AB} := +\infty; I_{AB} := \bar{0}$

**for each**  $j \in [1, n_B - m + 1]$  **in random order do**

$D := MASS(B_{j,m}, A)$

$ElementWiseMin(P_{AB}, I_{AB}, D, i)$

**if** *CheckForUserInterrupt* = TRUE **then**

print “Приближенное решение”,  $P_{AB}, I_{AB}$

**if** *GetUserChoice* = “Улучшить решение” **then**

**continue**

**else**

**break**

**Алгоритм с отсечением по времени\***

выдает допустимое (приближенное) решение,  
даже если он прерван до своего завершения

\* Zilberstein S., Russell S. Approximate reasoning using anytime algorithms. Natarajan S. (eds) Imprecise and Approximate Computation. The Springer Int'l. Series in Eng. and Comp. Sci. Vol 318. 1995. [https://doi.org/10.1007/978-0-585-26870-5\\_4](https://doi.org/10.1007/978-0-585-26870-5_4)

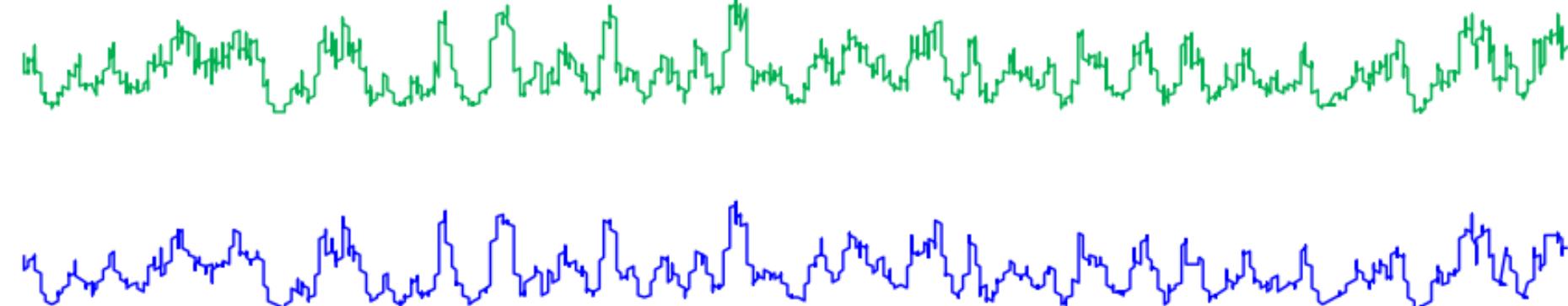
# Алгоритм STAMP (Scalable Time series Anytime Matrix Profile)

$$\text{Root Mean Square Error} = \sqrt{\frac{1}{n} \sum_{i=1}^n (\text{MP}_i - \widetilde{\text{MP}}_i)^2}$$



Приближенный  
матричный профиль  
(ряд Random Walk)  
после 1000 итераций

Точный  
матричный профиль  
(ряд Random Walk)



# Алгоритм STOMP (Scalable Time series Ordered-search Matrix Profile)\*

- Сравнение с STAMP
  - также алгоритм anytime (с отсечением по времени)
  - вычисление профилей расстояния выполняются в естественном (а не в случайном) порядке подпоследовательностей ряда  $B$  и оптимизируются
- Применяет формулу  $D_{T_{j,m}}(i) = \sqrt{2m\left(1 - \frac{\langle T_{j,m}, T_{i,m} \rangle - m\mu_{T_{j,m}}\mu_{T_{i,m}}}{m\sigma_{T_{j,m}}\sigma_{T_{i,m}}}\right)}$   
(см. лекцию «Поиск по образцу»)
  - $\mu$  и  $\sigma$  вычисляются за  $O(1)$  с помощью массивов кумулятивных сумм (функция *movstd*)
  - сложность вычисления  $\langle T_{j,m}, T_{i,m} \rangle$  также приводится к  $O(1)$

\* Yeh C.M., Zhu Y., Ulanova L., Begum N., Ding Y., Dau H.A., Silva D.F., Mueen A., Keogh E.J. Matrix Profile I: All pairs similarity joins for time series: A unifying view that includes motifs, discords and shapelets. Proc. of the IEEE 16th Int. Conf. on Data Mining, ICDM 2016, Barcelona, Spain, 12–15 December 2016. pp. 1317–1322. <https://doi.org/10.1109/ICDM.2016.0179>

# Алгоритм STOMP: Вычисление скалярных произведений за $O(1)$

- Введем обозначение:  $QT_{j,i} = \langle T_{j,m}, T_{i,m} \rangle$
- Запишем  $QT_{j-1,i-1}$  как сумму произведений:  
(1)  $QT_{j-1,i-1} = \sum_{k=0}^{m-1} t_{j-1+k} \cdot t_{i-1+k}$
- Запишем  $QT_{j,i}$  как сумму произведений:  
(2)  $QT_{j,i} = \sum_{k=0}^{m-1} t_{j+k} \cdot t_{i+k}$
- Найдем разность (2)–(1) и получим:  
$$QT_{j,i} = QT_{j-1,i-1} - t_{j-1} \cdot t_{i-1} + t_{j+m-1} \cdot t_{i+m-1} \blacksquare$$
- Для вычисления  $QT_{1,1}$  используем  $MASS(T_{1,m}, T)$

# Алгоритм STOMP (Scalable Time series Ordered-search Matrix Profile)

## Algorithm STOMP

**Input:** ряды  $A, B$ , длина подп-ти  $m$   
**Output:** МП  $P_{AB}$ , индекс МП  $I_{AB}$

Сложность STOMP  
 $O(n^2)$

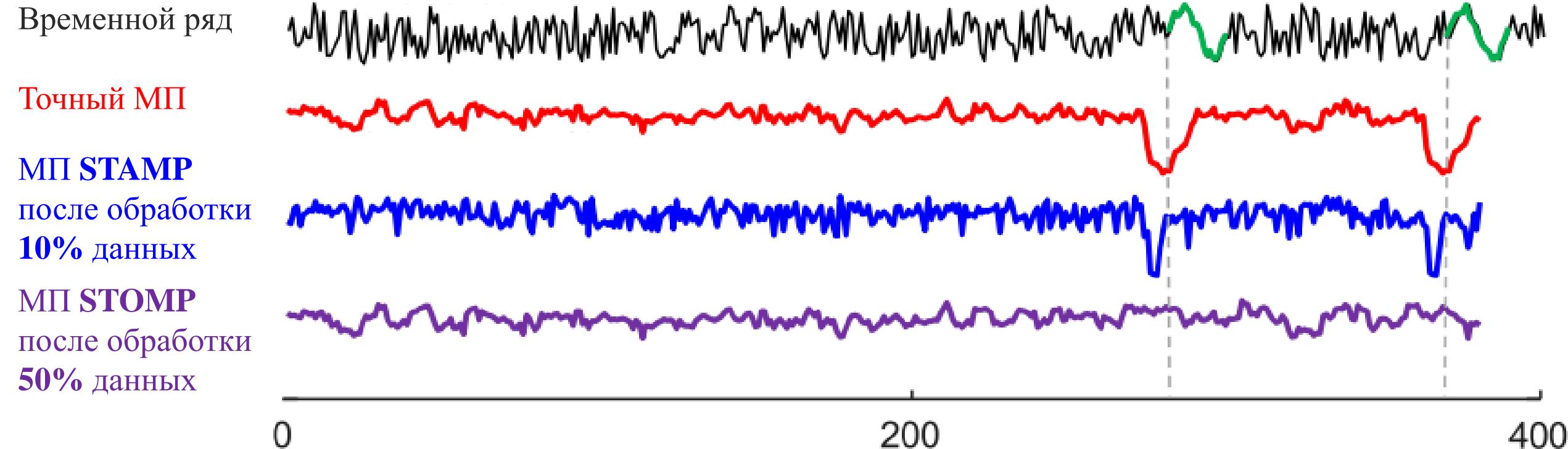
```

 $\{D, QT\} := MASS(B_{1,m}, A); \{DB, QTB\} := MASS(A_{1,m}, B)$ 
 $P_{AB} := D; I_{AB} := 1$ 
for  $i := 2$  to  $n_B - m + 1$  do
    for  $j := n_B - m + 1$  downto 2 do
         $QT(j) := QT(j - 1) - a_{j-1} \cdot b_{i-1} + a_{j+m-1} \cdot b_{i+m-1}$ 
     $QT(1) := QTB(i)$ 
     $\{\mu_B, \sigma_B, \mu_A, \sigma_A\} := CalcMeanStd(B_{i,m}, A)$ 
     $D := CalcDistProfile(QT, \mu_B, \sigma_B, \mu_A, \sigma_A)$ 
     $\{P_{AB}, I_{AB}\} := ElementWiseMin(P_{AB}, I_{AB}, D, i)$ 
return  $\{P_{AB}, I_{AB}\}$ 

```

$$D_{T_{j,m}}(i) = \sqrt{2m(1 - \frac{\langle T_{j,m}, T_{i,m} \rangle - m\mu_{T_{j,m}}\mu_{T_{i,m}}}{m\sigma_{T_{j,m}}\sigma_{T_{i,m}}})}$$

# Когда STOMP хуже STAMP



Мотивы расположены только справа, а случайные данные – слева.

Сходимость STOMP будет быстрой, как в STAMP,  
но лучшие мотивы не будут обнаружены до финальных итераций алгоритма

# Алгоритм SCRIMP («скупой»)\*

**Algorithm SCRIMP**

**Input:** ряд  $T$ , длина подп-ти  $m$

**Output:** МП  $P_T$ , индекс МП  $I_T$

$\{\mu_T, \sigma_T\} := \text{CalcMeanStd}(T, m)$

$P_T := +\infty; I_T := 1$

$Order := \text{RandPermutation}(m/4 + 1..n - m + 1)$

**for**  $k \in Order$  **do**

**for**  $i := 1$  **to**  $n - m + 2 - k$  **do**

**if**  $i = 1$  **then**

$QT := \text{DotProduct}(T_{1,m}, T_{k,m})$

**else**

$QT := QT - t_{i-1} \cdot t_{i+k-2} + t_{i+m-1} \cdot t_{i+m+k-2}$

$d := \text{CalcDistance}(QT, \mu_i, \sigma_i, \mu_{i+k-1}, \sigma_{i+k-1})$

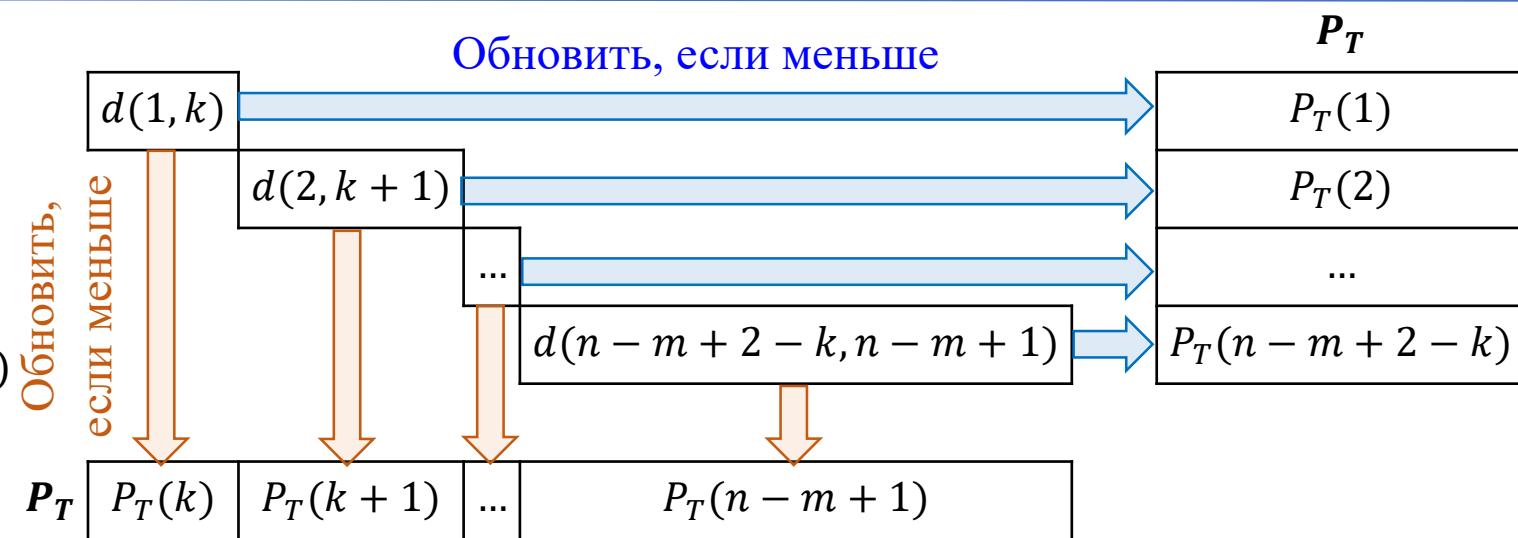
**if**  $d < P_T(i)$  **then**

$P_T(i) := d; I_T(i) := i + k - 1$

**if**  $d < P_T(i + k - 1)$  **then**

$P_T(i + k - 1) := d; I_T(i) := i + k - 1$

**return**  $\{P_T, I_T\}$



На каждом шаге обрабатывается случайная диагональ матрицы расстояний между подпоследовательностями  $S_T^m$  (т.е. SCRIMP – алгоритм вида anytime)

\* Zhu Y., Yeh C.M., Zimmerman Z., Kamgar K., Keogh E. Matrix Profile XI: SCRIMP++: Time Series Motif Discovery at Interactive Speeds. Proc. of the IEEE 18th Int. Conf. on Data Mining, ICDM 2018, Singapore, November 17-20, 2018. pp. 837-846. <https://doi.org/10.1109/ICDM.2018.00099>.

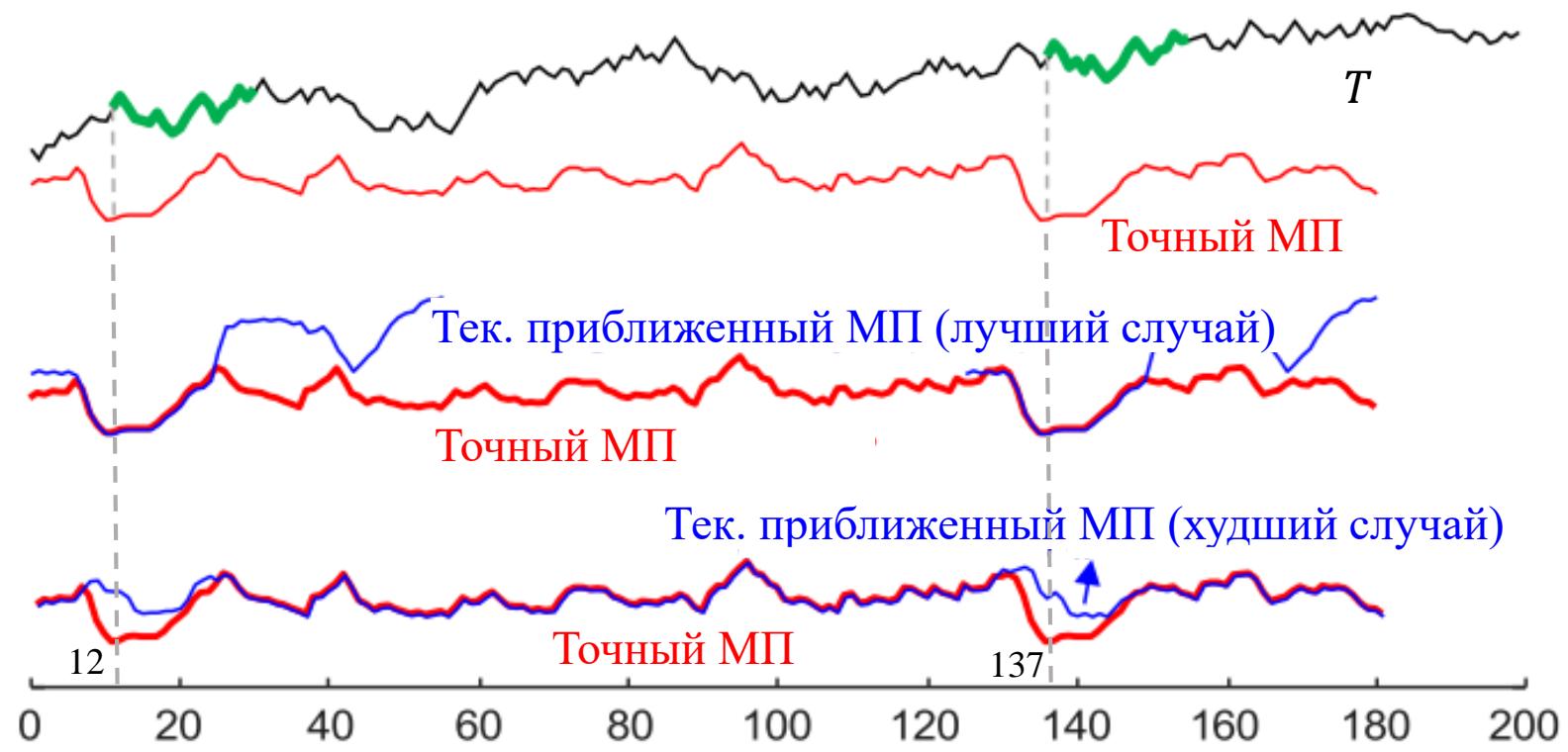
# В чем проблема SCRIMP



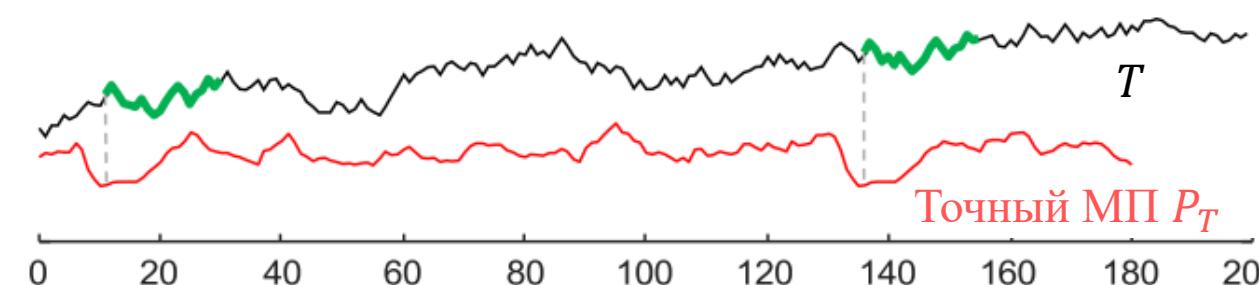
Для поиска  
локаций мотивов  
мы используем  
SCRIMP

Мы хотим  
от SCRIMP  
нахождения  
локаций мотивов

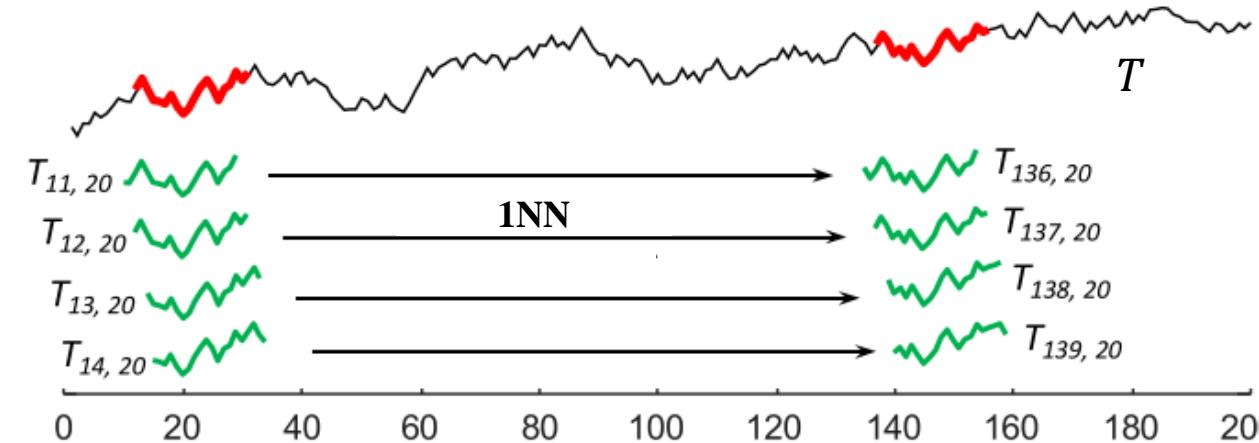
Чтобы найти мотивы, нужно вычислить диагональ матрицы расстояний, которая начинается с  $d(1,126)$  на самой ранней стадии. Иначе надо ждать 100% итераций. В отличие от STOMP, SCRIMP не зависит от локации мотива, но вероятность обработки до  $k$ -й итерации невысока:  $k/n-m+1$



# Свойство последовательного сохранения соседства (Consecutive Neighborhood Preserving Property)



$I_T$	1	2	3	4	...	7	8	9	...	24	25	...
	56	57	112	113	...	116	133	134	...	149	150	...



Если  $T_{i,m}$  очень похожа на  $T_{j,m}$ ,  
то с высокой вероятностью  
 $T_{i+1,m}$  очень похожа на  $T_{j+1,m}$

# Алгоритм PreSCRIMP (SCRIMP с предобработкой данных)

## Algorithm *PreSCRIMP*

**Input:** ряд  $T$ , длина подп-ти  $m$ , интервал  $s$

**Output:** МП  $P_T$ , индекс МП  $I_T$

$\{\mu_T, \sigma_T\} := \text{CalcMeanStd}(T, m)$

$P_T := +\infty; I_T := 1$

$Order := \text{RandPermutation}(1..n - m + 1 \text{ step } s)$

**for**  $i \in Order$  **do**

$D := \text{MASS}(T_{i,m}, T)$

$\{P_T, I_T\} := \text{ElementWiseMin}(P_T, I_T, D, i)$

$\{P_T(i), I_T(i)\} := \min D$

$j := I_T(i)$

$QT := \text{CalcDotProduct}(P_T(i), \mu_{T_{i,m}}, \sigma_{T_{i,m}}, \mu_{T_{j,m}}, \sigma_{T_{j,m}})$

$QT2 := QT$

**for**  $k := 1$  **to**  $\min(s - 1, n - m + 1 - \max(i, j))$  **do**

$QT := QT - t_{i+k-1} \cdot t_{j+k-1} + t_{i+k+m-1} \cdot t_{j+k+m-1}$

$d := \text{CalcDistance}(QT, \mu_{T_{i+k,m}}, \sigma_{T_{i+k,m}}, \mu_{T_{j+k,m}}, \sigma_{T_{j+k,m}})$

**if**  $d < P_T(i + k)$  **then**

$P_T(i + k) := d; I_T(i + k) := j + k$

**if**  $d < P_T(j + k)$  **then**

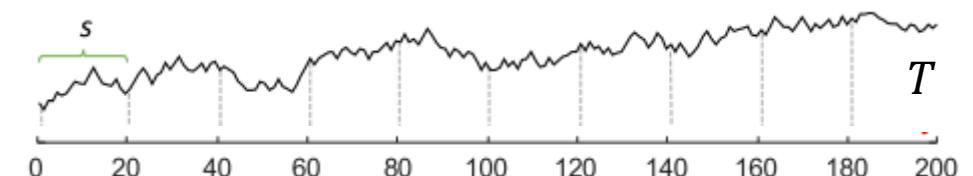
$P_T(j + k) := d; I_T(j + k) := i + k$

Выполним сэмплинг подпоследов-тей с **интервалом  $s$** .

Пусть выбрана  $T_{i,m}$  и  $\theta_{1\text{NN}}(T_{i,m}, T_{j,m}) = \text{TRUE}$ .

Тогда с высокой вероятностью  $\theta_{1\text{NN}}(T_{i+k,m}, T_{j+k,m}) = \text{TRUE}$

где  $k \in (1 - s, 2 - s, \dots, -2, -1, 1, 2, \dots, s - 2, s - 1)$



$QT := QT2$

**for**  $k := 1$  **to**  $\min(s - 1, i - 1, j - 1)$  **do**

$QT := QT - t_{i-k+m} \cdot t_{j-k+m} + t_{i-k} \cdot t_{j-k}$

$d := \text{CalcDistance}(QT, \mu_{T_{i-k,m}}, \sigma_{T_{i-k,m}}, \mu_{T_{j-k,m}}, \sigma_{T_{j-k,m}})$

**if**  $d < P_T(i - k)$  **then**

$P_T(i - k) := d; I_T(i - k) := j - k$

**if**  $d < P_T(j - k)$  **then**

$P_T(j - k) := d; I_T(j - k) := i - k$

**return**  $\{P_T, I_T\}$

# Алгоритм PreSCRIMP (SCRIMP с предобработкой данных)

## Algorithm *PreSCRIMP*

**Input:** ряд  $T$ , длина подп-ти  $m$ , интервал  $s$

**Output:** МП  $P_T$ , индекс МП  $I_T$

$\{\mu_T, \sigma_T\} := \text{CalcMeanStd}(T, m)$

$P_T := +\infty; I_T := 1$

$Order := \text{RandPermutation}(1..n - m + 1 \text{ step } s)$

**for**  $i \in Order$  **do**

$D := MASS(T_{i,m}, T)$

$\{P_T, I_T\} := \text{ElementWiseMin}(P_T, I_T, D, i)$

$\{P_T(i), I_T(i)\} := \min D$

$j := I_T(i)$

$QT := \text{CalcDotProduct}(P_T(i), \mu_{T_{i,m}}, \sigma_{T_{i,m}}, \mu_{T_{j,m}}, \sigma_{T_{j,m}})$

$QT2 := QT$

**for**  $k := 1$  **to**  $\min(s - 1, n - m + 1 - \max(i, j))$  **do**

$QT := QT - t_{i+k-1} \cdot t_{j+k-1} + t_{i+k+m-1} \cdot t_{j+k+m-1}$

$d := \text{CalcDistance}(QT, \mu_{T_{i+k,m}}, \sigma_{T_{i+k,m}}, \mu_{T_{j+k,m}}, \sigma_{T_{j+k,m}})$

**if**  $d < P_T(i + k)$  **then**

$P_T(i + k) := d; I_T(i + k) := j + k$

**if**  $d < P_T(j + k)$  **then**

$P_T(j + k) := d; I_T(j + k) := i + k$

Вычисление МП для выбранной подпоследовательности  $T_{i,m}$ ,  
где  $\theta_{1NN}(T_{i,m}, T_{j,m}) = \text{TRUE}$

$QT := QT2$   
**for**  $k := 1$  **to**  $\min(s - 1, i - 1, j - 1)$  **do**  
         $QT := QT - t_{i-k+m} \cdot t_{j-k+m} + t_{i-k} \cdot t_{j-k}$   
         $d := \text{CalcDistance}(QT, \mu_{T_{i-k,m}}, \sigma_{T_{i-k,m}}, \mu_{T_{j-k,m}}, \sigma_{T_{j-k,m}})$   
        **if**  $d < P_T(i - k)$  **then**  
             $P_T(i - k) := d; I_T(i - k) := j - k$   
        **if**  $d < P_T(j - k)$  **then**  
             $P_T(j - k) := d; I_T(j - k) := i - k$   
**return**  $\{P_T, I_T\}$

# Алгоритм PreSCRIMP (SCRIMP с предобработкой данных)

## Algorithm PreSCRIMP

**Input:** ряд  $T$ , длина подп-ти  $m$ , интервал  $s$   
**Output:** МП  $P_T$ , индекс МП  $I_T$   
 $\{\mu_T, \sigma_T\} := \text{CalcMeanStd}(T, m)$   
 $P_T := +\infty; I_T := 1$   
 $Order := \text{RandPermutation}(1..n - m + 1 \text{ step } s)$   
**for**  $i \in Order$  **do**  
     $D := \text{MASS}(T_{i,m}, T)$   
     $\{P_T, I_T\} := \text{ElementWiseMin}(P_T, I_T, D, i)$   
     $\{P_T(i), I_T(i)\} := \min D$   
     $j := I_T(i)$   
     $QT := \text{CalcDotProduct}(P_T(i), \mu_{T_{i,m}}, \sigma_{T_{i,m}}, \mu_{T_{j,m}}, \sigma_{T_{j,m}})$   
     $QT2 := QT$   
    **for**  $k := 1$  **to**  $\min(s - 1, n - m + 1 - \max(i, j))$  **do**  
         $QT := QT - t_{i+k-1} \cdot t_{j+k-1} + t_{i+k+m-1} \cdot t_{j+k+m-1}$   
         $d := \text{CalcDistance}(QT, \mu_{T_{i+k,m}}, \sigma_{T_{i+k,m}}, \mu_{T_{j+k,m}}, \sigma_{T_{j+k,m}})$   
        **if**  $d < P_T(i + k)$  **then**  
             $P_T(i + k) := d; I_T(i + k) := j + k$   
        **if**  $d < P_T(j + k)$  **then**  
             $P_T(j + k) := d; I_T(j + k) := i + k$

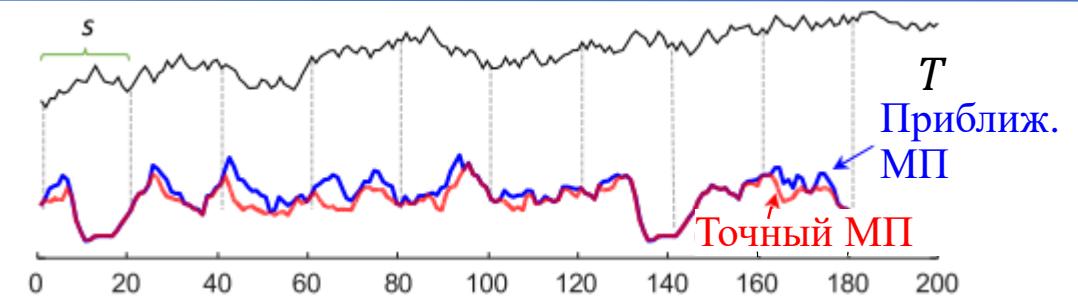
- Уточнение МП для окрестности подпоследовательности  $T_{i,m}$
1. Вычисляем попарные расстояния  $(T_{i+1,m}, T_{j+1,m}), \dots$  до следующей отобранной подпоследовательности или конца ряда
  2. Вычисляем попарные расстояния  $(T_{i-1,m}, T_{j-1,m}), \dots$  до предыдущей отобранной подпоследовательности или начала ряда
  3. Уточняем МП и индекс МП, если нашли меньшее расстояние

$QT := QT2$   
**for**  $k := 1$  **to**  $\min(s - 1, i - 1, j - 1)$  **do**  
     $QT := QT - t_{i-k+m} \cdot t_{j-k+m} + t_{i-k} \cdot t_{j-k}$   
     $d := \text{CalcDistance}(QT, \mu_{T_{i-k,m}}, \sigma_{T_{i-k,m}}, \mu_{T_{j-k,m}}, \sigma_{T_{j-k,m}})$   
    **if**  $d < P_T(i - k)$  **then**  
         $P_T(i - k) := d; I_T(i - k) := j - k$   
    **if**  $d < P_T(j - k)$  **then**  
         $P_T(j - k) := d; I_T(j - k) := i - k$   
**return**  $\{P_T, I_T\}$

# Алгоритм PreSCRIMP (SCRIMP с предобработкой данных)

**Algorithm** *PreSCRIMP*

**Input:** ряд  $T$ , длина подп-ти  $m$ , интервал  $s$   
**Output:** МП  $P_T$ , индекс МП  $I_T$   
 $\{\mu_T, \sigma_T\} := \text{CalcMeanStd}(T, m)$   
 $P_T := +\infty; I_T := 1$   
 $Order := \text{RandPermutation}(1..n - m + 1 \text{ step } s)$   
**for**  $i \in Order$  **do**  
     $D := MASS(T_{i,m}, T)$   
     $\{P_T, I_T\} := \text{ElementWiseMin}(P_T, I_T, D, i)$   
     $\{P_T(i), I_T(i)\} := \min D$   
     $j := I_T(i)$   
     $QT := \text{CalcDotProduct}(P_T(i), \mu_{T_{i,m}}, \sigma_{T_{i,m}}, \mu_{T_{j,m}}, \sigma_{T_{j,m}})$   
     $QT2 := QT$   
    **for**  $k := 1$  **to**  $\min(s - 1, n - m + 1 - \max(i, j))$  **do**  
         $QT := QT - t_{i+k-1} \cdot t_{j+k-1} + t_{i+k+m-1} \cdot t_{j+k+m-1}$   
         $d := \text{CalcDistance}(QT, \mu_{T_{i+k,m}}, \sigma_{T_{i+k,m}}, \mu_{T_{j+k,m}}, \sigma_{T_{j+k,m}})$   
        **if**  $d < P_T(i + k)$  **then**  
             $P_T(i + k) := d; I_T(i + k) := j + k$   
        **if**  $d < P_T(j + k)$  **then**  
             $P_T(j + k) := d; I_T(j + k) := i + k$



PreSCRIMP хорошо приближает МП, особенно для минимумов МП (мотивы)

$QT := QT2$   
**for**  $k := 1$  **to**  $\min(s - 1, i - 1, j - 1)$  **do**  
     $QT := QT - t_{i-k+m} \cdot t_{j-k+m} + t_{i-k} \cdot t_{j-k}$   
     $d := \text{CalcDistance}(QT, \mu_{T_{i-k,m}}, \sigma_{T_{i-k,m}}, \mu_{T_{j-k,m}}, \sigma_{T_{j-k,m}})$   
    **if**  $d < P_T(i - k)$  **then**  
         $P_T(i - k) := d; I_T(i - k) := j - k$   
    **if**  $d < P_T(j - k)$  **then**  
         $P_T(j - k) := d; I_T(j - k) := i - k$   
**return**  $\{P_T, I_T\}$

# Алгоритм PreSCRIMP (SCRIMP с предобработкой данных)

## Algorithm *PreSCRIMP*

**Input:** ряд  $T$ , длина подп-ти  $m$ , интервал  $s$

**Output:** МП  $P_T$ , индекс МП  $I_T$

$\{\mu_T, \sigma_T\} := \text{CalcMeanStd}(T, m)$

$P_T := +\infty; I_T := 1$

$Order := \text{RandPermutation}(1..n - m + 1 \text{ step } s)$

**for**  $i \in Order$  **do**

$D := MASS(T_{i,m}, T)$

$\{P_T, I_T\} := \text{ElementWiseMin}(P_T, I_T, D, i)$

$\{P_T(i), I_T(i)\} := \min D$

$j := I_T(i)$

$QT := \text{CalcDotProduct}(P_T(i), \mu_{T_{i,m}}, \sigma_{T_{i,m}}, \mu_{T_{j,m}}, \sigma_{T_{j,m}})$

$QT2 := QT$

**for**  $k := 1$  **to**  $\min(s - 1, n - m + 1 - \max(i, j))$  **do**

$QT := QT - t_{i+k-1} \cdot t_{j+k-1} + t_{i+k+m-1} \cdot t_{j+k+m-1}$

$d := \text{CalcDistance}(QT, \mu_{T_{i+k,m}}, \sigma_{T_{i+k,m}}, \mu_{T_{j+k,m}}, \sigma_{T_{j+k,m}})$

**if**  $d < P_T(i + k)$  **then**

$P_T(i + k) := d; I_T(i + k) := j + k$

**if**  $d < P_T(j + k)$  **then**

$P_T(j + k) := d; I_T(j + k) := i + k$

**Сложность:** PreSCRIMP  $O(\frac{1}{s} n^2 \log_2 n)$ , STOMP/STAMP:  $O(n^2 \log_2 n)$

## Как выбирать $s$ ?

Невыбранная подпоследовательность должна пересекаться с одной из выбранных минимум в  $1 - \frac{s}{2m}$  точках. Типичное значение:  $s = \frac{m}{4}$ .

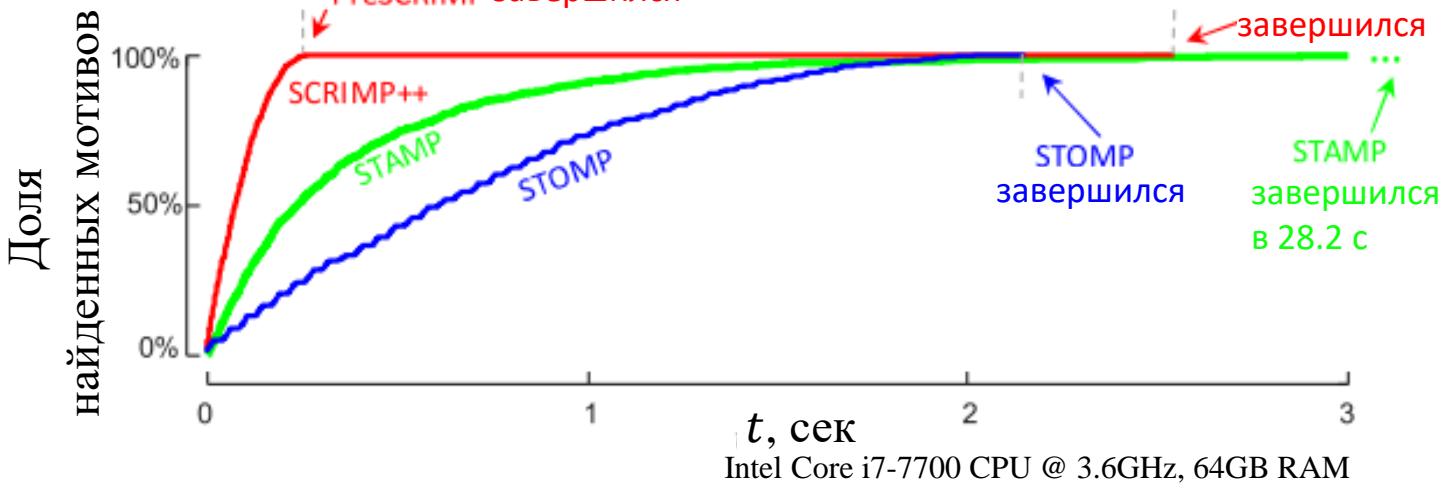
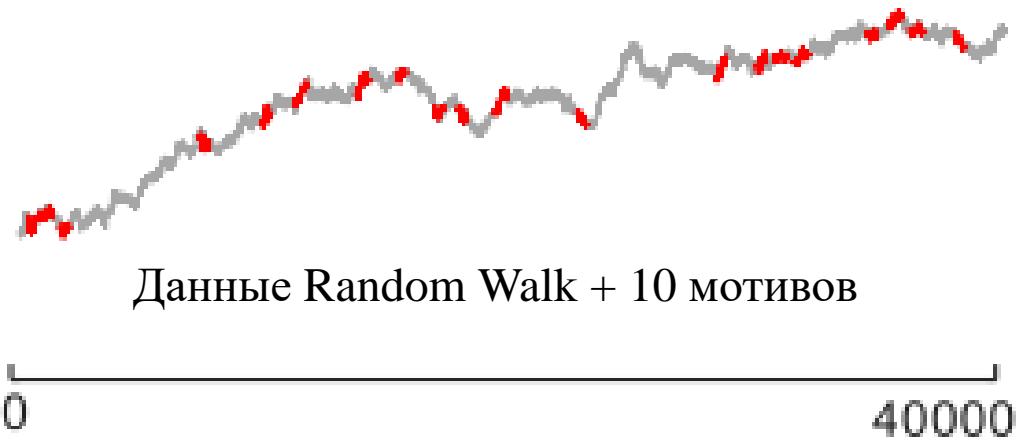
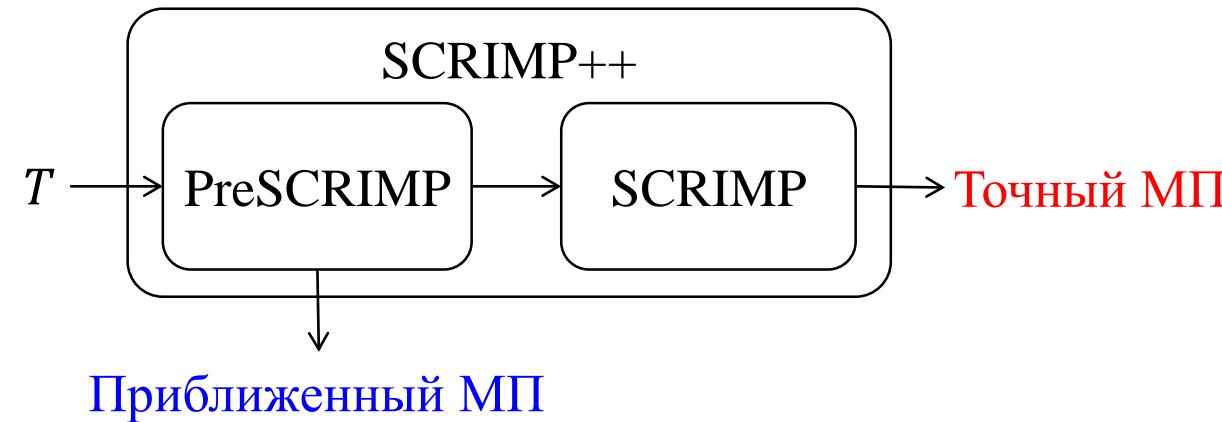
Тогда сложность PreSCRIMP  $O(\frac{1}{m} n^2 \log_2 n)$ ,  
т.е. небольшая часть от сложности STOMP/STAMP.

```

    QT := QT2
    for  $k := 1$  to  $\min(s - 1, i - 1, j - 1)$  do
        QT := QT -  $t_{i-k+m} \cdot t_{j-k+m} + t_{i-k} \cdot t_{j-k}$ 
        d :=  $\text{CalcDistance}(QT, \mu_{T_{i-k,m}}, \sigma_{T_{i-k,m}}, \mu_{T_{j-k,m}}, \sigma_{T_{j-k,m}})$ 
        if  $d < P_T(i - k)$  then
             $P_T(i - k) := d; I_T(i - k) := j - k$ 
        if  $d < P_T(j - k)$  then
             $P_T(j - k) := d; I_T(j - k) := i - k$ 
    return  $\{P_T, I_T\}$ 

```

# Алгоритм SCRIMP++ = PreSCRIMP+SCRIMP



# Алгоритмы вычисления матричного профиля

Наиболее быстрый алгоритм для случая одномерного ряда

Алгоритм	Anytime	Инкрементный	Параллельный	Платформа	Многомерный
STAMP <sup>1</sup>	✓			CPU	
STAMPI <sup>1</sup>		✓		CPU	
STOMP <sup>1</sup>	✓			CPU	
GPU-STOMP <sup>2</sup>	✓		✓	GPU	
SCRIMP++ <sup>3</sup>				CPU	
SCRIMP++ <sup>4</sup>	✓		✓	cluster	
SCAMP <sup>5</sup>			✓	CPU, GPU, cluster	
TraTSA <sup>6</sup>			✓	FPGA	
(MP) <sup>N</sup>			✓	cluster	✓

- Yeh C.-C.M. et al. Matrix Profile I: All pairs similarity joins for time series: A unifying view that includes motifs, discords and shapelets. ICDM 2016. pp. 1317-1322. <https://doi.org/10.1109/ICDM.2016.0179>.
- Zhu Y. et al. Matrix profile II: Exploiting a novel algorithm and GPUs to break the one hundred million barrier for time series motifs and joins. ICDM 2016. pp. 739-748. <https://doi.org/10.1109/ICDM.2016.0085>
- Zhu Y. et al. Matrix profile XI: SCRIMP++: time series motif discovery at interactive speeds. ICDM 2018. pp. 837-846. <https://doi.org/10.1109/ICDM.2018.00099>
- Pfeilschifter G. Time series analysis with matrix profile on HPC systems // Master thesis, Department of Informatics, Technical University of Munich, Germany. 2019. URL: <http://mediatum.ub.tum.de/doc/1471292/1471292.pdf>.
- Zimmerman Z. et al. Matrix Profile XIV: Scaling time series motif discovery with GPUs to break a quintillion pairwise comparisons a day and beyond. SoCC 2019. pp. 74-86. <https://doi.org/10.1145/3357223.3362721>
- Fernandez R.Q. et al. TraTSA: A Transprecision framework for efficient time series analysis. Journal of Computational Science. Vol.63, 2022. Art. 101784. <https://doi.org/10.1016/j.jocs.2022.101784>.
- Raoofy A. et al. Time series mining at petascale performance. LNCS. 2020. Vol. 12151. pp. 104-123. [https://doi.org/10.1007/978-3-030-50743-5\\_6](https://doi.org/10.1007/978-3-030-50743-5_6).

# Литература

1. Yeh C.M., Zhu Y., Ulanova L., Begum N., Ding Y., Dau H.A., Silva D.F., Mueen A., Keogh E.J. Matrix Profile I: All pairs similarity joins for time series: A unifying view that includes motifs, discords and shapelets. Proc. of the IEEE 16th Int. Conf. on Data Mining, ICDM 2016, Barcelona, Spain, 12-15 December 2016. pp. 1317-1322.  
<https://doi.org/10.1109/ICDM.2016.0179>.
2. Zhu Y., Yeh C.M., Zimmerman Z., Kamgar K., Keogh E. Matrix Profile XI: SCRIMP++: Time Series Motif Discovery at Interactive Speeds. Proc. of the IEEE 18th Int. Conf. on Data Mining, ICDM 2018, Singapore, November 17-20, 2018. pp. 837-846.  
<https://doi.org/10.1109/ICDM.2018.00099>.
3. Zhu Y., Gharghabi S., Silva D.F., Dau H.A., Yeh C.-C.M., Senobari N.S., Almaslukh A., Kamgar K., Zimmerman Z., Funning G., Mueen A., Keogh E. 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. Discov. 34(4): 949-979 (2020).  
<https://doi.org/10.1007/s10618-019-00668-6>.
4. Keogh E.J. The UCR Matrix Profile Page. URL:  
<https://www.cs.ucr.edu/~eamonn/MatrixProfile.html>.