

Научный семинар

Лаборатории больших данных и машинного обучения ЮУрГУ

---

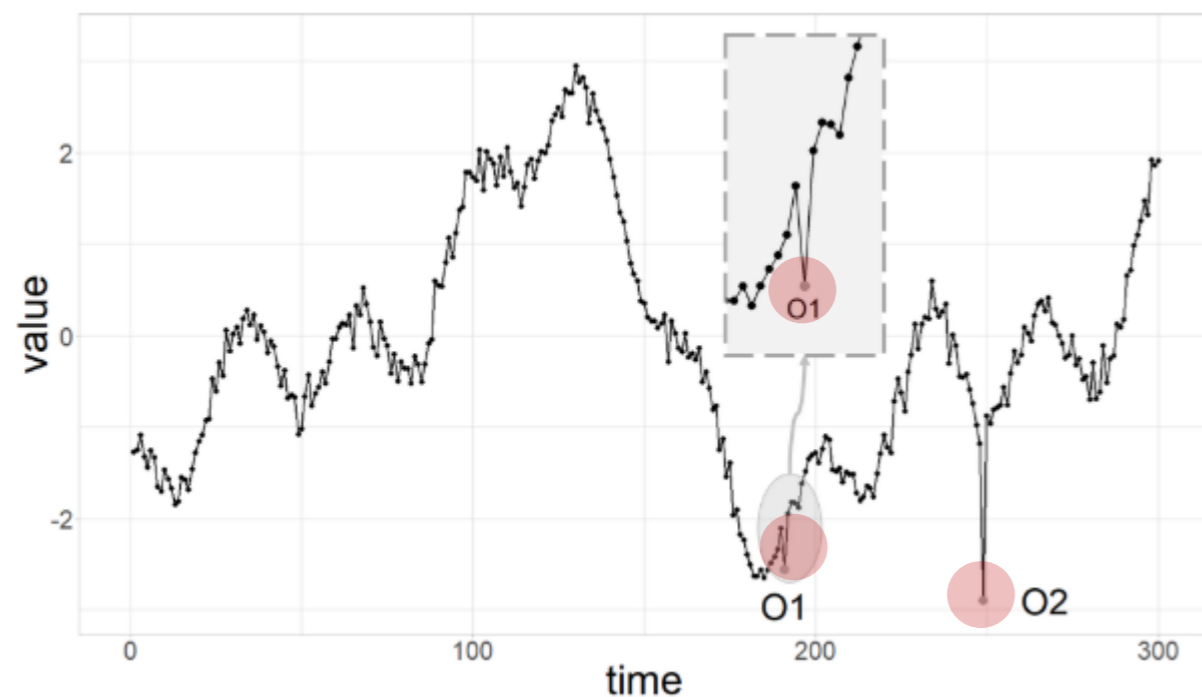
# Параллельный алгоритм поиска аномалий временного ряда для графического процессора

Я.А. Краева, М.Л. Цымблер

---

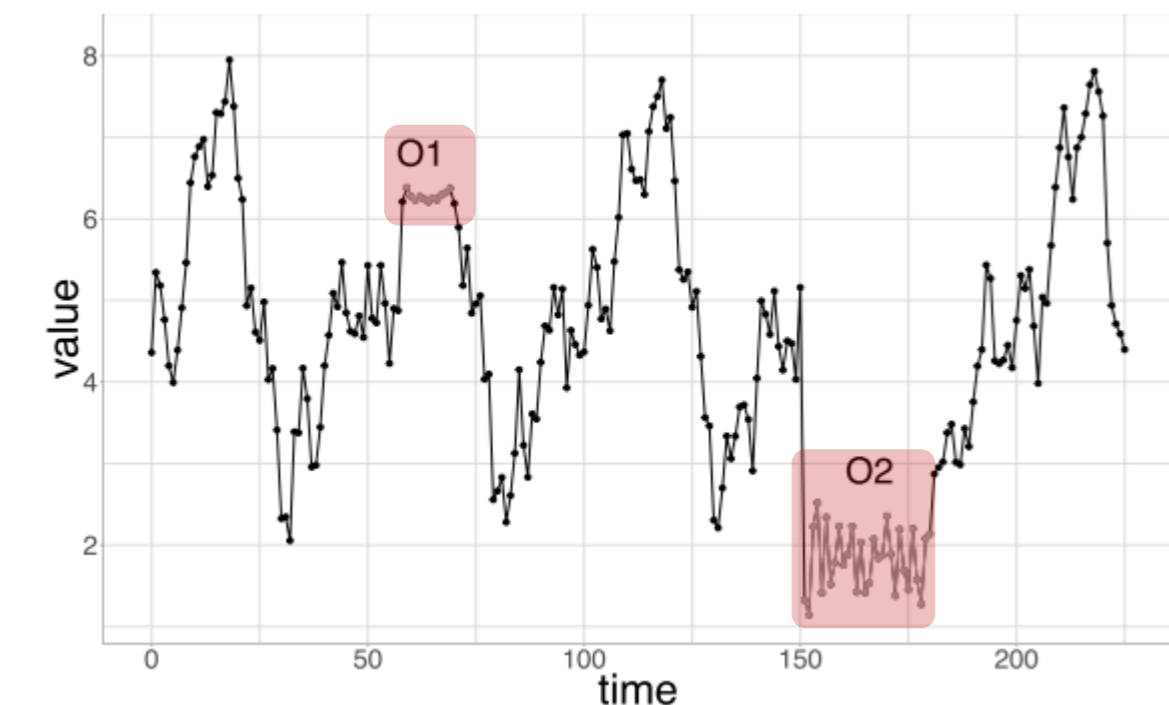
Челябинск, 2022

# Понятие аномалии в временного ряда<sup>1)</sup>



## Точечная аномалия

элемент ряда, который ведет себя необычно в определенный момент времени в сравнении либо с другими значениями (*глобальный выброс*), либо с соседними элементами (*локальный выброс*)



## Аномальная подпоследовательность

непрерывная подпоследовательность элементов ряда, совместное поведение которых необычно, хотя каждый элемент в отдельности не обязательно является точечной аномалией

<sup>1)</sup> Blazquez-Garca A., et al. A review on outlier/anomaly detection in time series data. ACM Comput. Surv. 54(3), 56:1-56:33 (2021). <https://doi.org/10.1145/3444690>

# Аномальная подпоследовательность: формализация

## Термин, источник

**Outliers** are the data points for which there are fewer than  $p$  other data points within distance  $d$ .

Knorr E., Ng N. Finding intensional knowledge of distance-based outliers. VLDB 1999. pp. 211-222.

**Outliers** are the top  $n$  data points whose distance to their  $k$ -th nearest neighbor is greatest.

Ramaswamy S. *et al.* Efficient algorithms for mining outliers from large dataset. SIGMOD 2000. pp. 427-438.

**Outliers** are the top  $n$  data points whose average distance to their  $k$  nearest neighbors is greatest.

Angiulli F., Pizzuti C. Fast outlier detection in high dimensional spaces. PKDD 2002. pp. 15-26.

**Discord in a time series** is a subsequence of length  $n$  whose distance to its nearest non-self match neighbor is greatest.

Keogh E. *et al.* HOT SAX: Efficiently finding the most unusual time series subsequence. ICDM 2005. pp. 226-233.

**Range discord in a time series** is a subsequence of length  $n$  whose distance to its nearest non-self match neighbor is at least  $r$ .

Yankov D. *et al.* Disk aware discord discovery: finding unusual time series in terabyte sized datasets. KAIS 2008. vol. 17, no. 2. pp. 241-262.

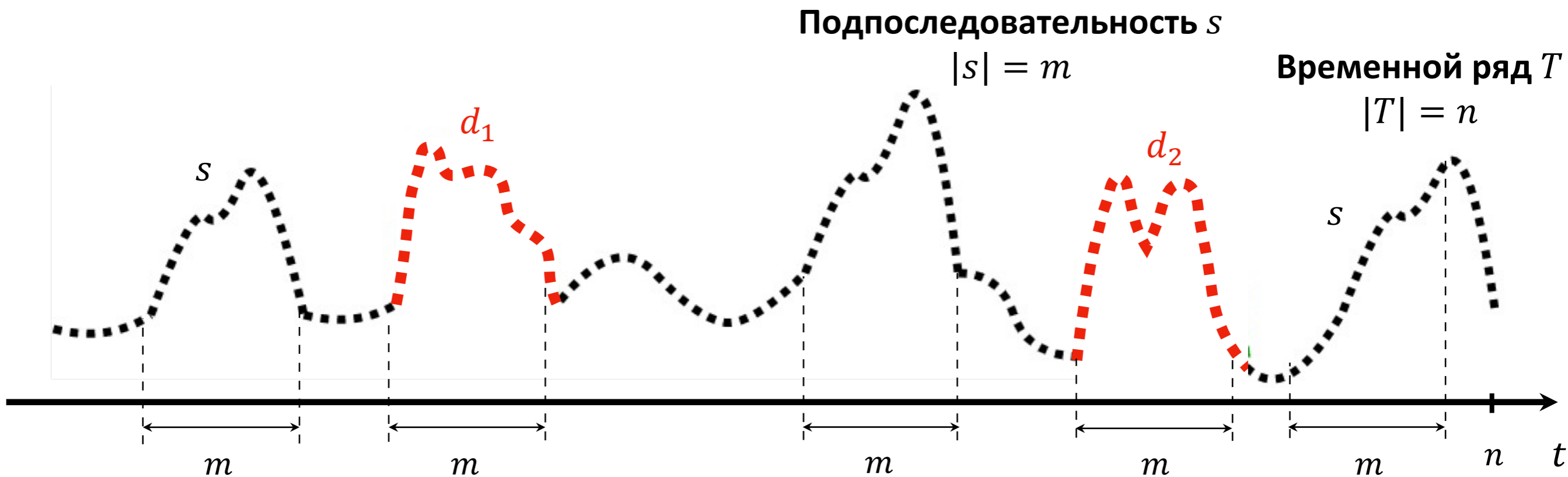
## Диссонансы – лучший подход для поиска аномалий временных рядов

Yeh C.-C.M. *et al.* Time series joins, motifs, discords and shapelets: a unifying view that exploits the matrix profile. **Data Min. Knowl. Discov. (Scopus TOP-1)** 2018. vol. 32, no. 1. pp. 83-123.

Chandola V. *et al.* Anomaly detection: A survey. ACM Comput. Surv. 2009. 41(3), 15:1-15:58.

# Постановка задачи

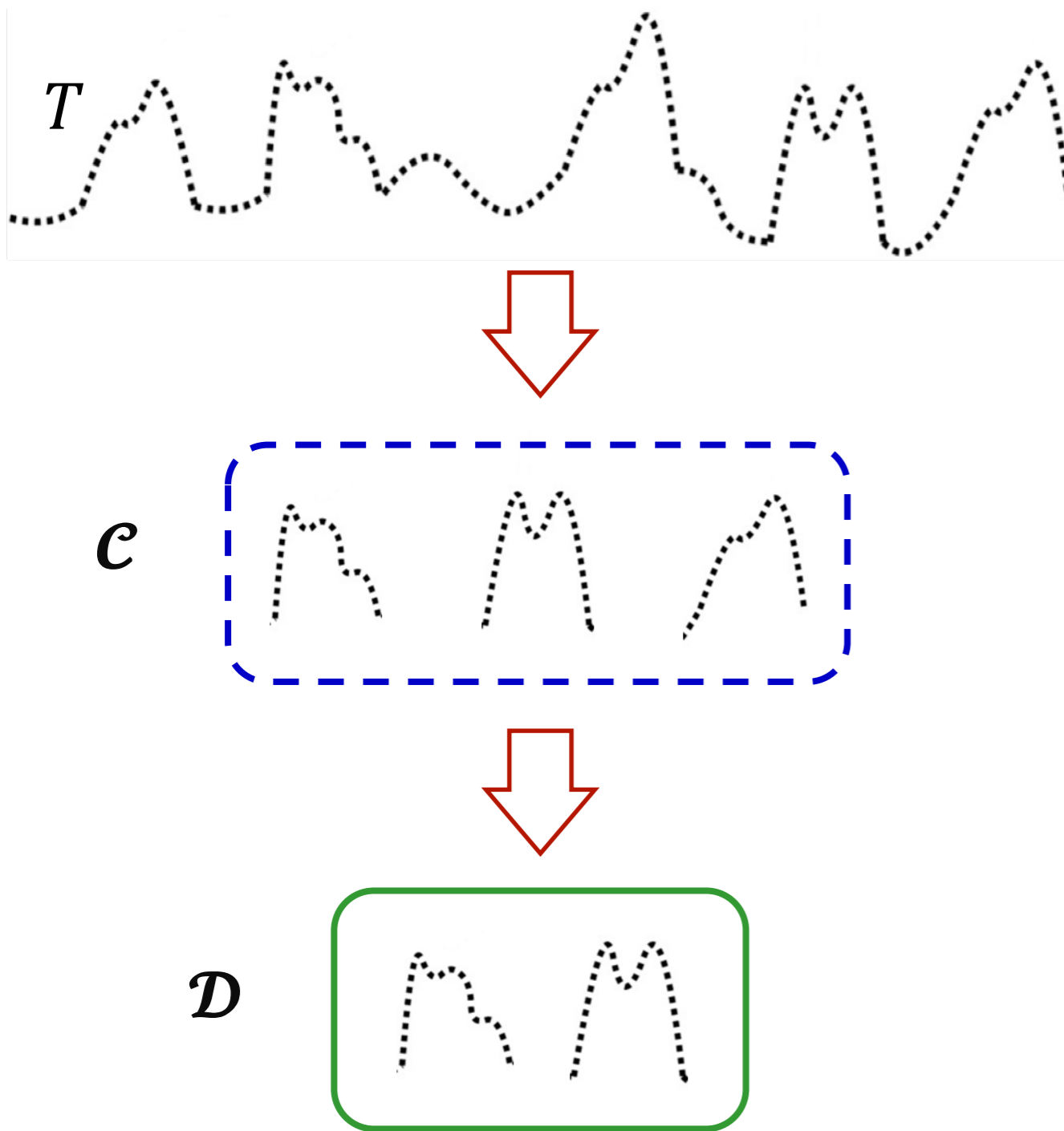
- **Диапазонный диссонанс**<sup>1)</sup> – подпоследовательность ряда заданной длины, расстояние от которой до ближайшего соседа не ниже заданного порога  $r$   
- Ближайший сосед – наиболее похожая непересекающаяся подпоследовательность
- **Дано:** временной ряд  $T$ , длина диссонанса  $m$ , порог  $r$
- **Найти:**  $D = \{d_1, d_2, \dots\}$ ,  $d_i \in D \Leftrightarrow \forall s \in T \min_{s \cap d_i = \emptyset} \text{dist}(d_i, s) \geq r$



Количество подпоследовательностей  $s$  длины  $m$  в ряде  $T$  длины  $n$ :  $N = n - m + 1$

<sup>1)</sup> Yankov D., Keogh E.J., Rebbapragada U. Disk aware discord discovery: finding unusual time series in terabyte sized datasets. Knowl. Inf. Syst. 17(2): 241-262. 2008.

# Последовательный алгоритм DRAG<sup>1)</sup>



## 1. Отбор

За одно сканирование ряда сформировать **множество кандидатов в диссонансы**

## 2. Очистка

За одно сканирование ряда **отбросить ложноположительных кандидатов**

<sup>1)</sup> Yankov D., Keogh E.J., Rebbapragada U. Disk aware discord discovery: finding unusual time series in terabyte sized datasets. Knowl. Inf. Syst.17(2): 241–262. 2008.

# Отбор кандидатов

пока не конец ряда  $T$ :

текущая подпоследовательность  $s$

Кандидат := TRUE

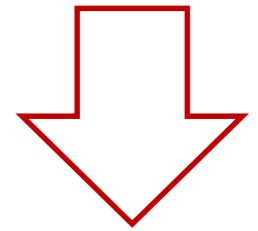
для всех  $c_i \in \mathcal{C}$  и  $s \cap c_i = \emptyset$

если  $\text{dist}(s, c_i) < r$  то

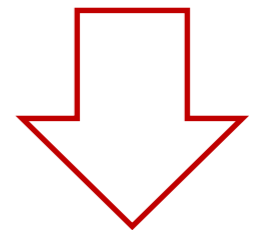
$\mathcal{C} := \mathcal{C} \setminus c_i$ ; Кандидат := FALSE

если Кандидат = TRUE то  $\mathcal{C} := \mathcal{C} \cup s$

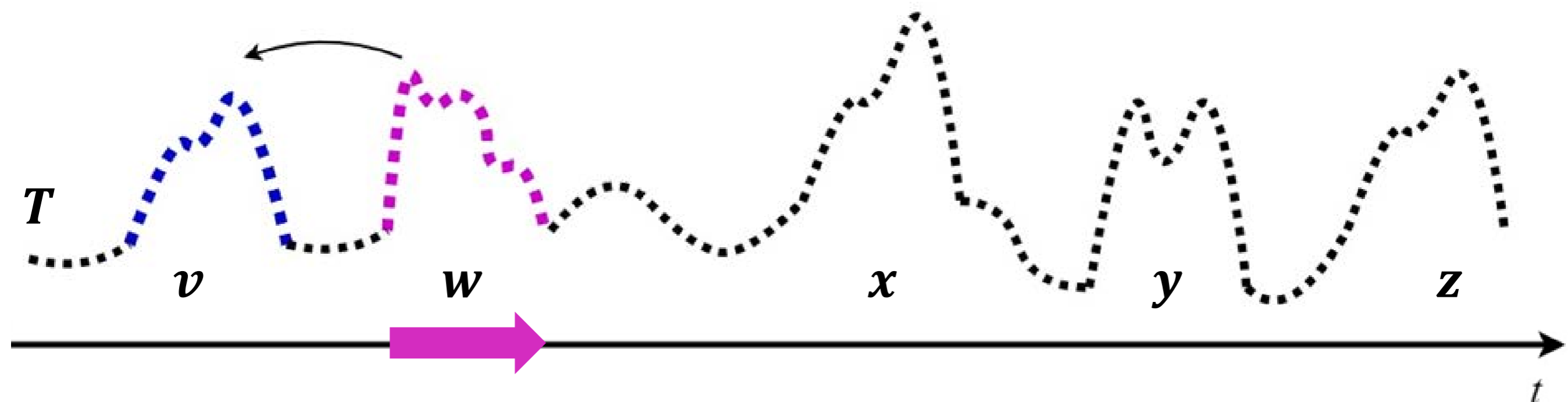
$$\mathcal{C} = \{v\}$$



$$\text{dist}(w, v) \geq r$$



$$\mathcal{C} = \{v, w\}$$



# Отбор кандидатов

пока не конец ряда  $T$ :

текущая подпоследовательность  $s$

Кандидат := TRUE

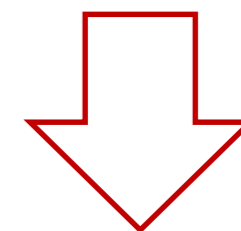
для всех  $c_i \in \mathcal{C}$  и  $s \cap c_i = \emptyset$

если  $\text{dist}(s, c_i) < r$  то

$\mathcal{C} := \mathcal{C} \setminus c_i$ ; Кандидат := FALSE

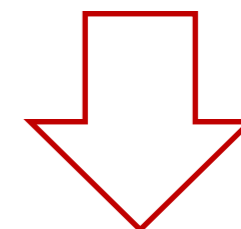
если Кандидат = TRUE то  $\mathcal{C} := \mathcal{C} \cup s$

$$\mathcal{C} = \{v, w\}$$

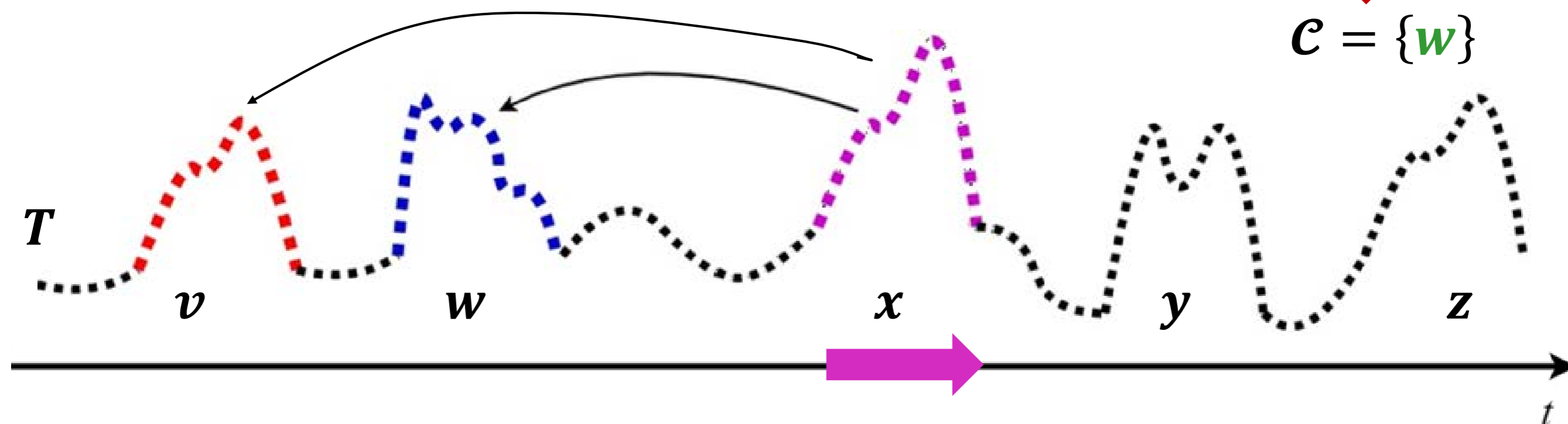


$$\text{dist}(x, v) < r$$

$$\text{dist}(x, w) \geq r$$



$$\mathcal{C} = \{w\}$$



# Отбор кандидатов

пока не конец ряда  $T$ :

текущая подпоследовательность  $s$

Кандидат := TRUE

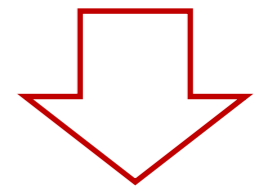
для всех  $c_i \in \mathcal{C}$  и  $s \cap c_i = \emptyset$

если  $\text{dist}(s, c_i) < r$  то

$\mathcal{C} := \mathcal{C} \setminus c_i$ ; Кандидат := FALSE

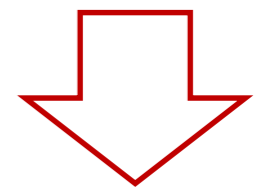
если Кандидат = TRUE то  $\mathcal{C} := \mathcal{C} \cup s$

$$\mathcal{C} = \{w, y\}$$

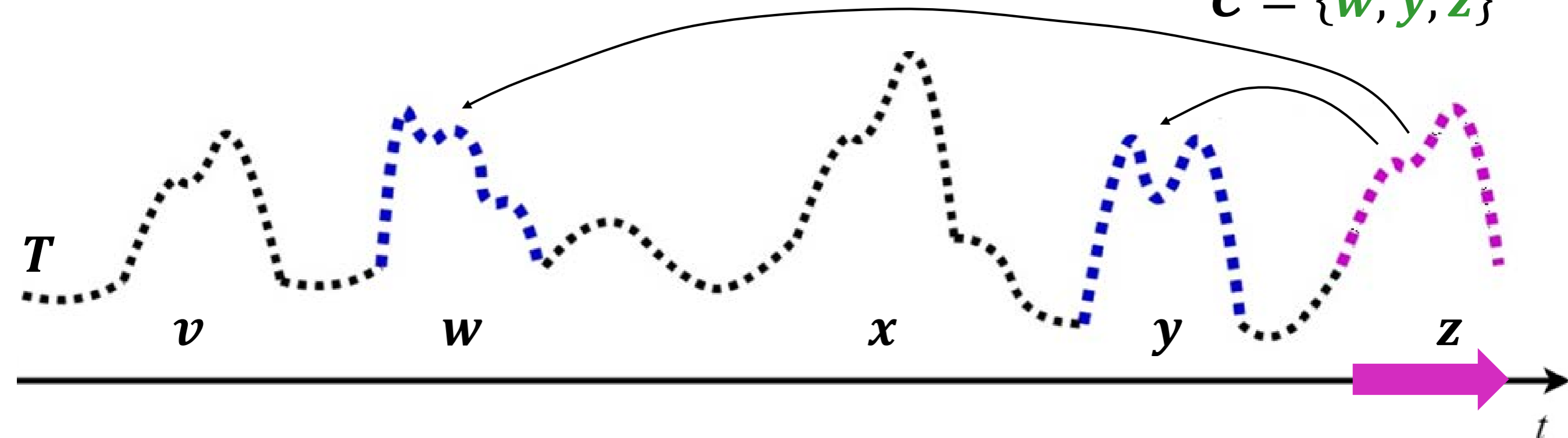


$$\text{dist}(z, w) \geq r$$

$$\text{dist}(z, y) \geq r$$



$$\mathcal{C} = \{w, y, z\}$$





# Очистка кандидатов

$$\mathcal{D} := \mathcal{C}$$

пока не конец ряда  $T$ :

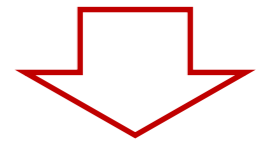
текущая подпоследовательность  $s$

для всех  $d_i \in \mathcal{D}$  и  $s \cap d_i = \emptyset$

если  $\text{dist}(s, d_i) < r$  то

$$\mathcal{D} := \mathcal{D} \setminus d_i$$

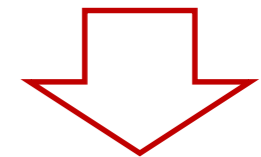
$$\mathcal{D} = \{w, y, z\}$$



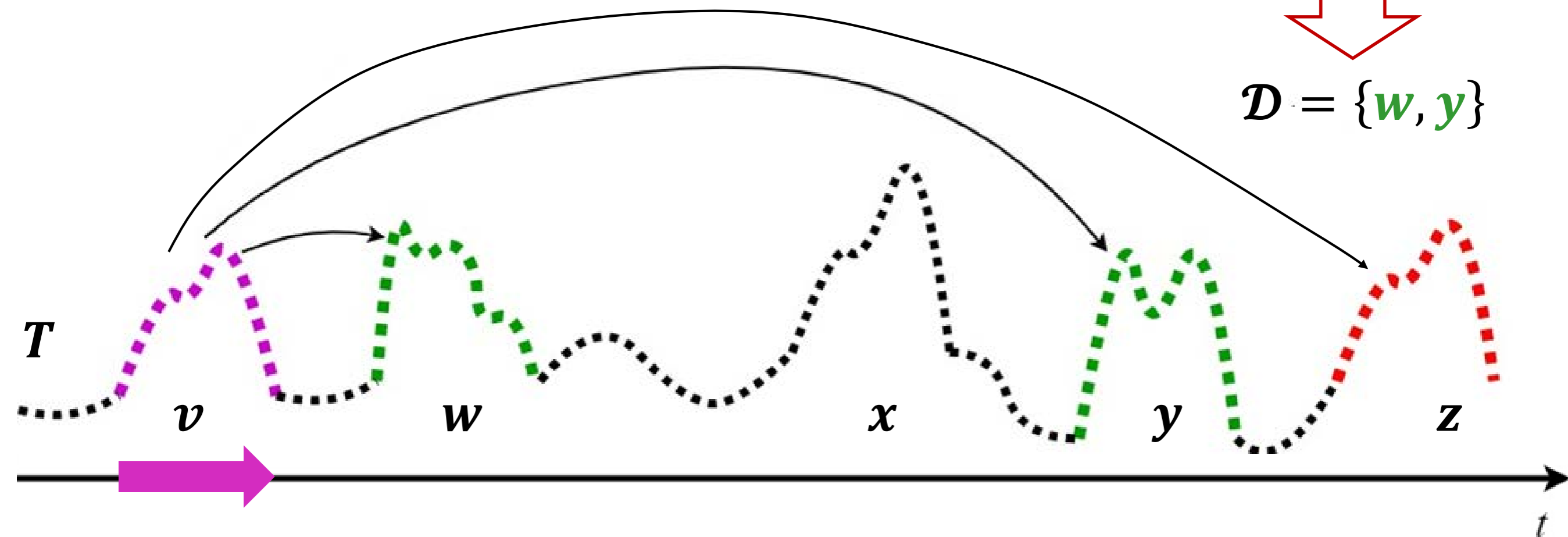
$$\text{dist}(v, w) \geq r$$

$$\text{dist}(v, y) \geq r$$

$$\text{dist}(v, z) < r$$



$$\mathcal{D} = \{w, y\}$$



# Очистка кандидатов

$\mathcal{D} := \mathcal{C}$

пока не конец ряда  $T$ :

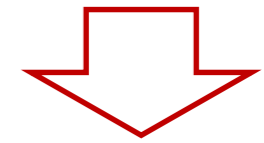
текущая подпоследовательность  $s$

для всех  $d_i \in \mathcal{D}$  и  $s \cap d_i = \emptyset$

если  $\text{dist}(s, d_i) < r$  то

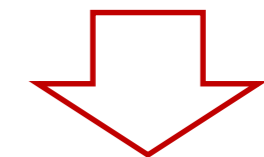
$\mathcal{D} := \mathcal{D} \setminus d_i$

$\mathcal{D} = \{w, y\}$

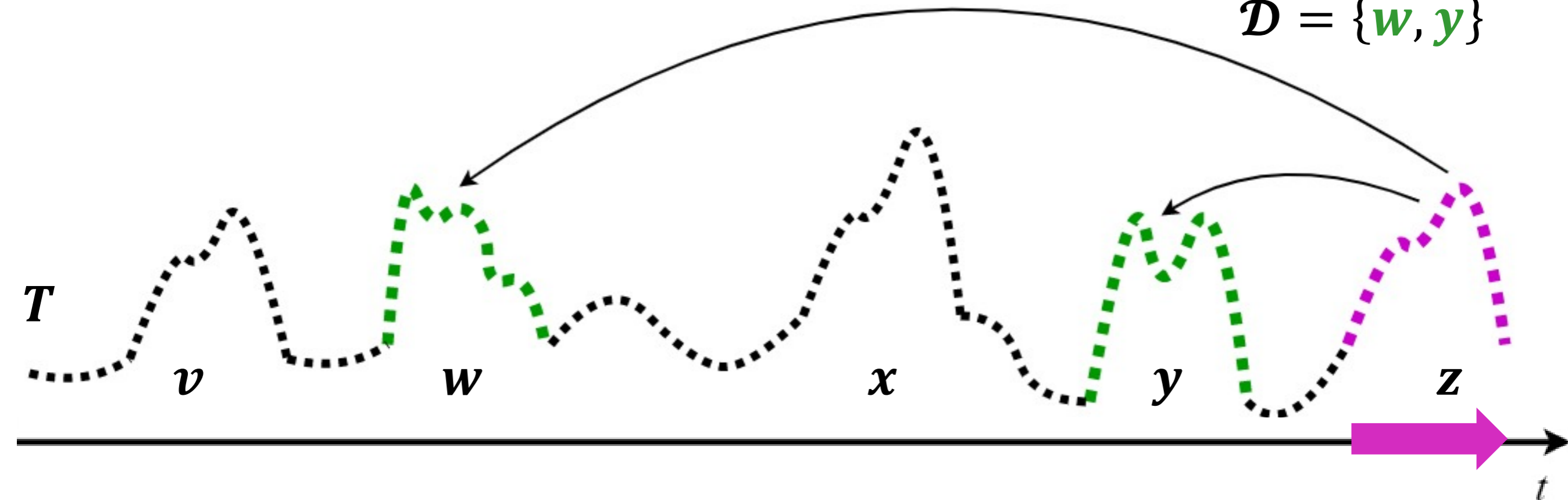


$\text{dist}(z, w) \geq r$

$\text{dist}(z, y) \geq r$



$\mathcal{D} = \{w, y\}$



# Проблемы алгоритма DRAG

- 1. Ручной подбор длины диссонанса  $m$** 
  - Не всегда заранее известна длина аномалии
- 2. Не работает для диапазона длин диссонанса**
  - Запуск DRAG для всех возможных длин вычислительно неосуществим
- 3. Ручной подбор порога  $r$** 
  - Слишком большой порог – нет диссонансов, слишком маленький порог – много ложноположительных аномалий
- 4. Необходимо распараллеливание**
  - Сложность в худшем случае  $O(n^2)$

# Последовательный алгоритм MERLIN<sup>1)</sup>

**Algorithm** MERLIN (IN  $T, minL, maxL, topK$ ; OUT  $\mathcal{D}$ )

```
1:  $\mathcal{D} \leftarrow \emptyset$ ;  $r \leftarrow 2\sqrt{minL}$ ;  $nnDist_{minL} \leftarrow -\infty$ 
2: while  $nnDist_{minL} < 0$  and  $|D_{minL}| < topK$  do
3:    $D_{minL} \leftarrow DRAG(T, minL, r)$ ;  $\mathcal{D} \leftarrow \mathcal{D} \cup D_{minL}$ ;  $nnDist_{minL} \leftarrow \min_{d \in D_{minL}} d.nnDist$ 
4:    $r \leftarrow 0.5 \cdot r$ 
-----
5: for  $i \leftarrow minL + 1$  to  $minL + 4$  do
6:    $nnDist_i \leftarrow -\infty$ 
7:   while  $nnDist_i < 0$  and  $|D_i| < topK$  do
8:      $r \leftarrow 0.99 \cdot nnDist_{i-1}$ 
9:      $D_i \leftarrow DRAG(T, i, r)$ ;  $\mathcal{D} \leftarrow \mathcal{D} \cup D_i$ ;  $nnDist_i \leftarrow \min_{d \in D_i} d.nnDist$ 
10:     $r \leftarrow 0.99 \cdot r$ 
-----
11: for  $i \leftarrow minL + 5$  to  $maxL$  do
12:    $\mu \leftarrow \text{Mean}(\{nnDist_k\}_{k=i-1}^{i-5})$ ;  $\sigma \leftarrow \text{Std}(\{nnDist_k\}_{k=i-1}^{i-5})$ ;  $r \leftarrow \mu - 2\sigma$ 
13:    $D_i \leftarrow DRAG(T, i, r)$ ;  $\mathcal{D} \leftarrow \mathcal{D} \cup D_i$ ;  $nnDist_i \leftarrow \min_{d \in D_i} d.nnDist$ 
14:   while  $nnDist_i < 0$  and  $|D_i| < topK$  do
15:      $D_i \leftarrow DRAG(T, i, r)$ ;  $\mathcal{D} \leftarrow \mathcal{D} \cup D_i$ ;  $nnDist_i \leftarrow \min_{d \in D_i} d.nnDist$ 
16:      $r \leftarrow r - \sigma$ 
17: return  $\mathcal{D}$ 
```

**Шаг 1.** Поиск диссонансов минимальной длины  $minL$   
 $r = 2\sqrt{minL}$

**Шаг 2.** Поиск диссонансов следующих четырех длин  
 $r = 0.99 \cdot nnDist_{m-1}$

**Шаг 3.** Поиск диссонансов всех оставшихся длин  
 $r = \mu - 2\sigma$

<sup>1)</sup> Nakamura T., et al. MERLIN: parameter-free discovery of arbitrary length anomalies in massive time series archives. ICDM 2020. pp. 1190-1195.  
<https://doi.org/10.1109/ICDM50108.2020.00147>.

- **Применение  $ED_{\text{norm}}^2$  в качестве функции расстояния**

- $ED_{\text{norm}}(T_{i,m}, T_{j,m}) = \sqrt{2m(1 - \frac{\langle T_{i,m}, T_{j,m} \rangle - m\mu_i\mu_j}{m\sigma_i\sigma_j})}$  <sup>3)</sup>

- Легко распараллеливается, отсутствие квадратного корня ускоряет вычисления

- **Сокращение избыточных вычислений в MERLIN**

- Вычисление  $\mu(T_{i,m})$  и  $\sigma(T_{i,m})$  включает в себя вычисление  $\mu(T_{i,m-1})$  и  $\sigma(T_{i,m-1})$ , на этом можно сэкономить

- **Распараллеливание DRAG**

- Отдельное распараллеливание фаз отбора и очистки

<sup>1)</sup> Zymbler M.L., Kraeva Ya.A. High-performance Time Series Anomaly Discovery on Graphics Processor (готовится к публикации).

<sup>2)</sup> Краева Я.А., Цымблер М.Л. Свидетельство о регистрации программы для ЭВМ «PALMAD: детектор аномалий различной длины во временном ряде на графическом процессоре» № 2022667716 от 23.09.2022.

<sup>3)</sup> Mueen A. et al. Fast approximate correlation for massive time-series data. SIGMOD 2010. pp. 171-182. ACM (2010).

<https://doi.org/10.1145/1807167.1807188>

# PALMAD: Parallel Arbitrary Length MERLIN-based Anomaly Discovery

**Algorithm** PALMAD (IN  $T$ ,  $minL$ ,  $maxL$ ,  $topK$ ; OUT  $\mathcal{D}$ )

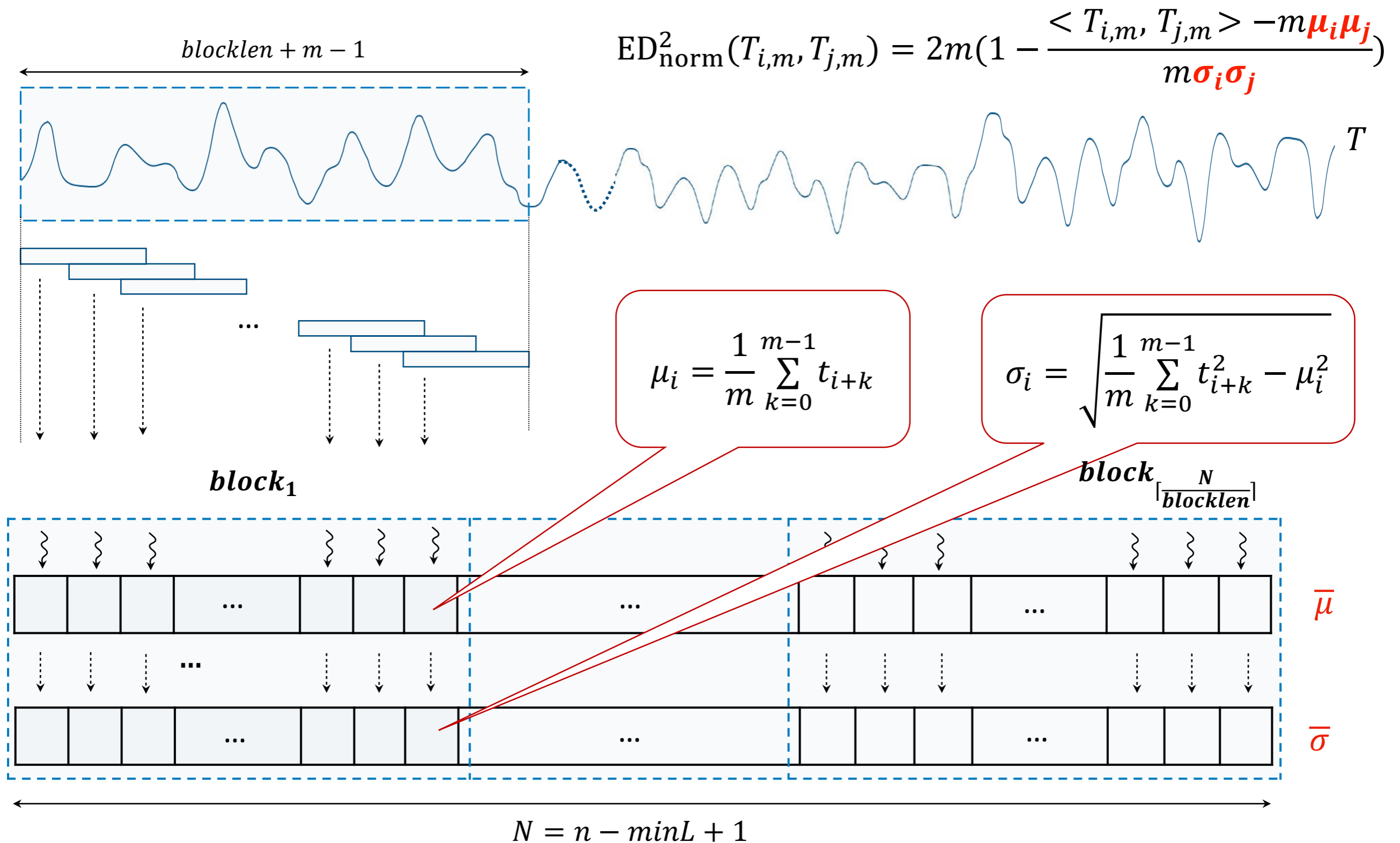
```
1:  $\mathcal{D} \leftarrow \emptyset$ ;  $r \leftarrow 2\sqrt{minL}$ ;  $nnDist_{minL} \leftarrow -\infty$ 
2:  $\{\bar{\mu}, \bar{\sigma}\} \leftarrow \text{CALCMEANSTD}(T, minL)$ 
3: while  $nnDist_{minL} < 0$  and  $|D_{minL}| < topK$  do
4:    $D_{minL} \leftarrow \text{PD3}(T, \bar{\mu}, \bar{\sigma}, minL, r^2)$ ;  $\mathcal{D} \leftarrow \mathcal{D} \cup D_{minL}$ ;  $nnDist_{minL} \leftarrow \min_{d \in D_{minL}} d.nnDist$ 
5:    $r \leftarrow 0.5 \cdot r$ 
6: for  $i \leftarrow minL + 1$  to  $minL + 4$  do
7:    $nnDist_i \leftarrow -\infty$ 
8:    $\{\bar{\mu}, \bar{\sigma}\} \leftarrow \text{UPDATEMEANSTD}(T, \bar{\mu}, \bar{\sigma}, i)$ 
9:   while  $nnDist_i < 0$  and  $|D_i| < topK$  do
10:     $r \leftarrow 0.99 \cdot nnDist_{i-1}$ 
11:     $D_i \leftarrow \text{PD3}(T, \bar{\mu}, \bar{\sigma}, i, r^2)$ ;  $\mathcal{D} \leftarrow \mathcal{D} \cup D_i$ ;  $nnDist_i \leftarrow \min_{d \in D_i} d.nnDist$ 
12:     $r \leftarrow 0.99 \cdot r$ 
13: for  $i \leftarrow minL + 5$  to  $maxL$  do
14:    $\mu \leftarrow \text{Mean}(\{nnDist_k\}_{k=i-1}^{i-5})$ ;  $\sigma \leftarrow \text{Std}(\{nnDist_k\}_{k=i-1}^{i-5})$ ;  $r \leftarrow \mu - 2\sigma$ 
15:    $\{\bar{\mu}, \bar{\sigma}\} \leftarrow \text{UPDATEMEANSTD}(T, \bar{\mu}, \bar{\sigma}, i)$ 
16:    $D_i \leftarrow \text{PD3}(T, \bar{\mu}, \bar{\sigma}, i, r^2)$ ;  $\mathcal{D} \leftarrow \mathcal{D} \cup D_i$ ;  $nnDist_i \leftarrow \min_{d \in D_i} d.nnDist$ 
17:   while  $nnDist_i < 0$  and  $|D_i| < topK$  do
18:     $D_i \leftarrow \text{PD3}(T, \bar{\mu}, \bar{\sigma}, i, r^2)$ ;  $\mathcal{D} \leftarrow \mathcal{D} \cup D_i$ ;  $nnDist_i \leftarrow \min_{d \in D_i} d.nnDist$ 
19:     $r \leftarrow r - \sigma$ 
20: return  $\mathcal{D}$ 
```

– сокращение  
избыточных  
вычислений

– распараллеливание  
DRAG (PD3, Parallel  
DRAG-based Discord  
Discovery)

– применение  $ED_{\text{norm}}^2$

# Предобработка: вычисление $\bar{\mu}$ и $\bar{\sigma}$



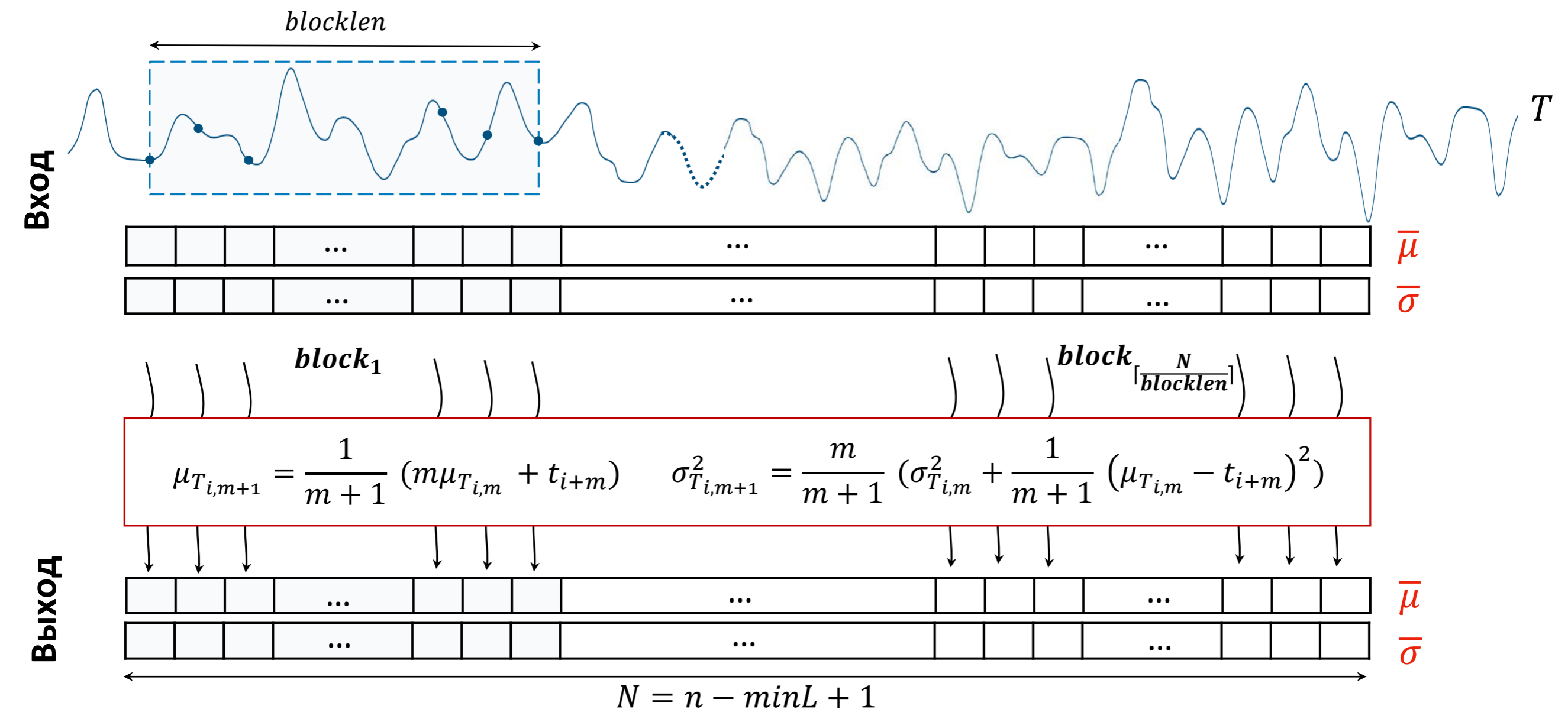
# Предобработка: обновление $\bar{\mu}$ и $\bar{\sigma}$

**Лемма.** Пусть дан ряд  $T$ ,  $|T| = n$  и подпоследовательности  $T_{i,m}$  и  $T_{i,m+1}$ .

Тогда

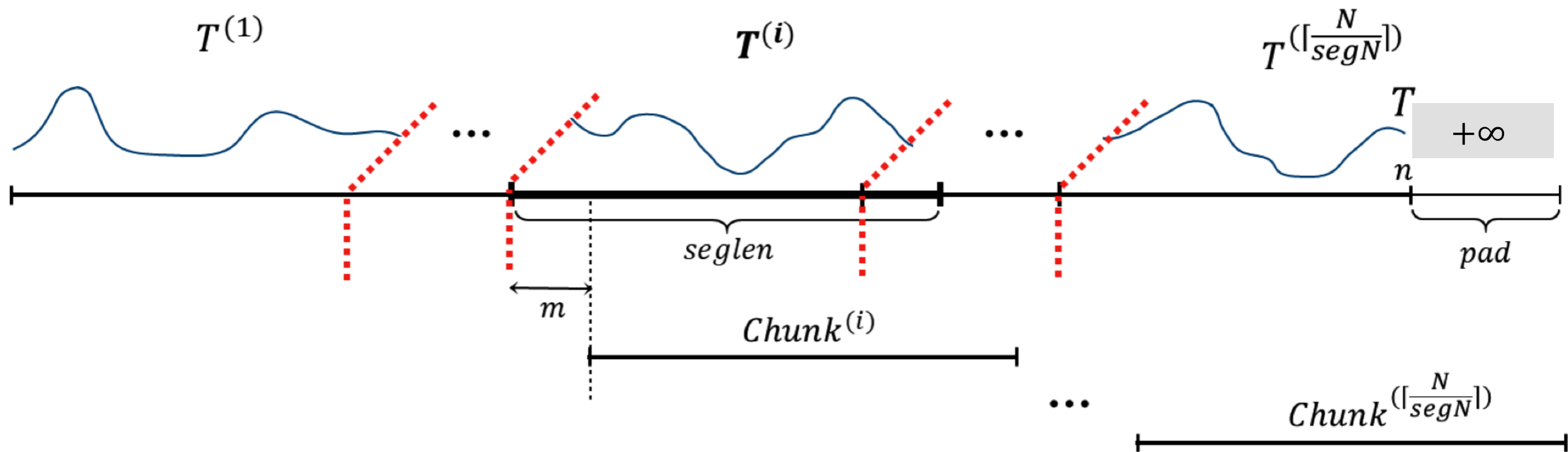
$$\mu_{T_{i,m+1}} = \frac{1}{m+1} (m\mu_{T_{i,m}} + t_{i+m}),$$

$$\sigma_{T_{i,m+1}}^2 = \frac{m}{m+1} \left( \sigma_{T_{i,m}}^2 + \frac{1}{m+1} (\mu_{T_{i,m}} - t_{i+m})^2 \right).$$



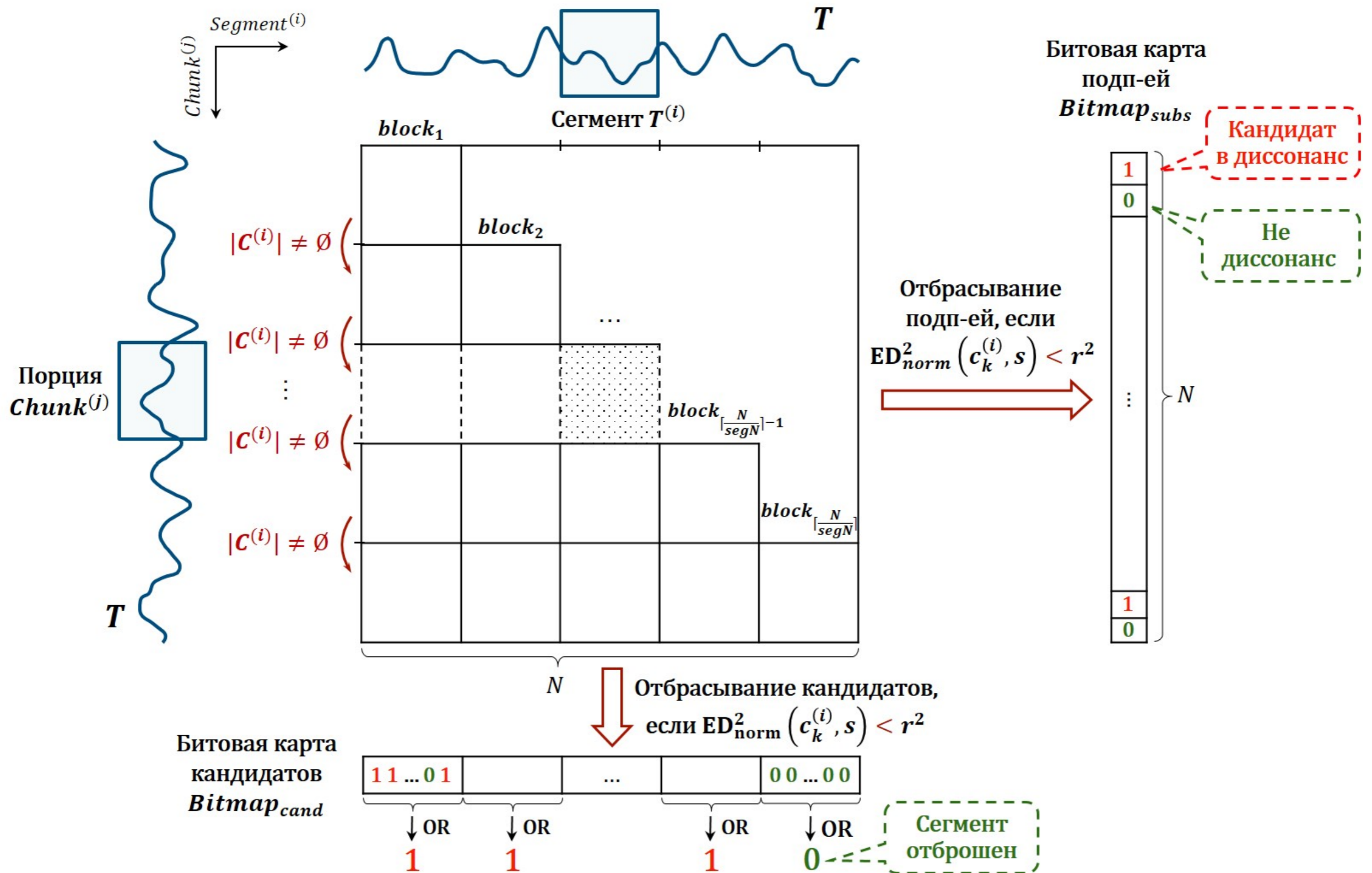


# PD3: сегментирование ряда

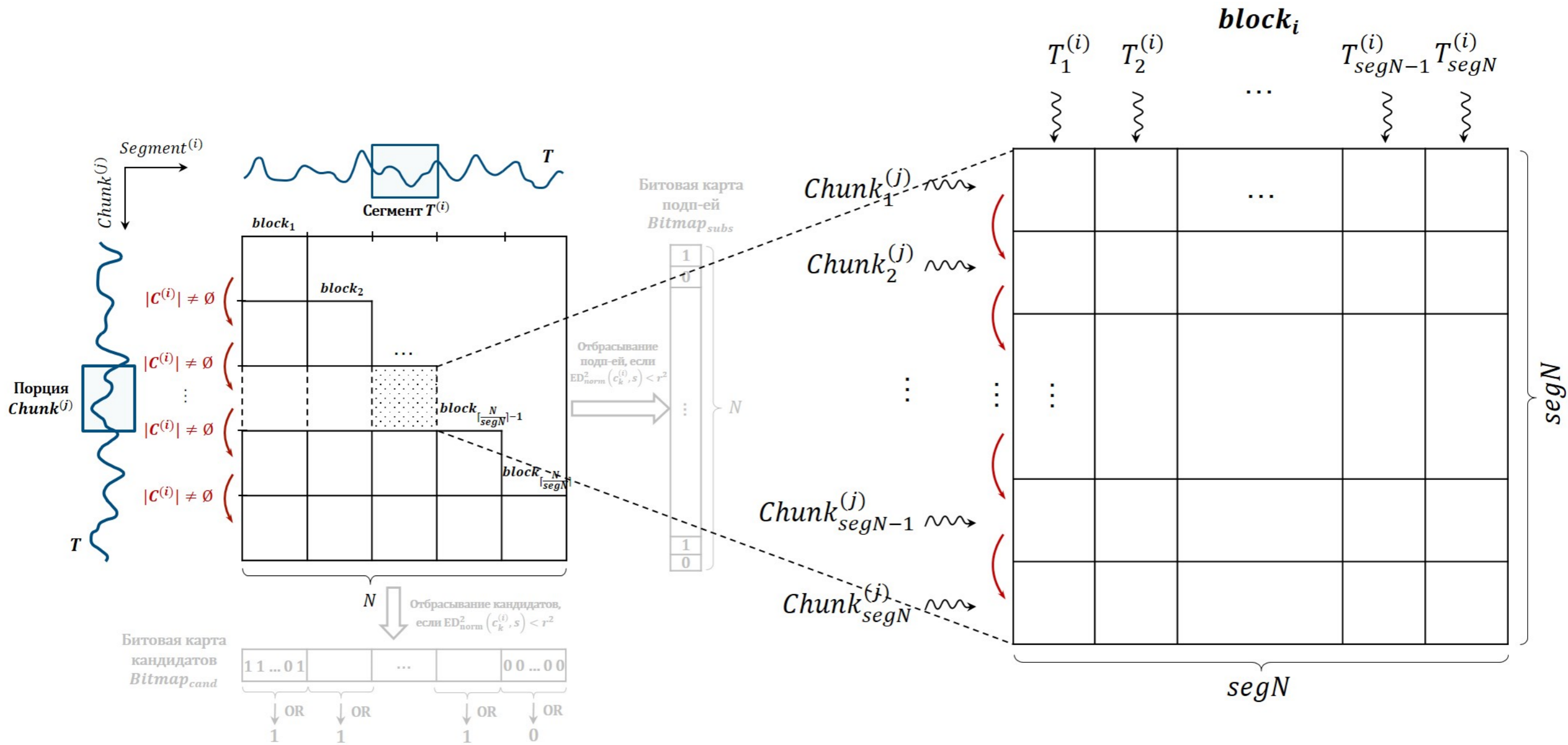


Переменная	Семантика
$seglen$	Длина сегмента, $seglen = segN + m - 1$
$segN$	Количество кандидатов в сегменте (параметр) кратно размеру варпа ( $warpSize = 32$ )
$T^{(i)}$	Сегмент кандидатов
$Chunk^{(j)}$	Элементы ряда для отбрасывания кандидатов на очередной итерации
$pad$	Количество фиктивных элементов ряда

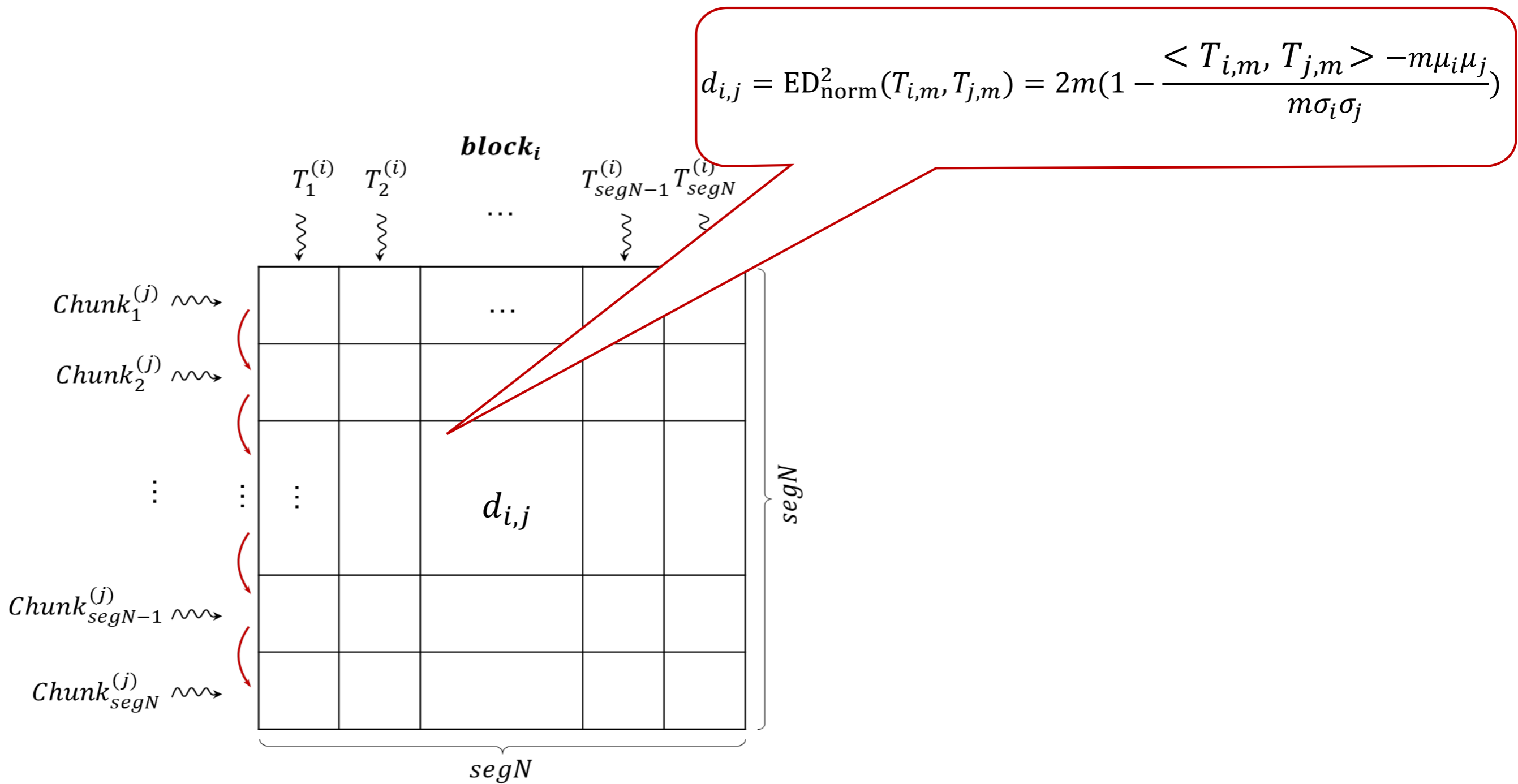
# PD3: Отбор кандидатов, блочное распараллеливание



# PD3: Отбор кандидатов, распараллеливание по нитям

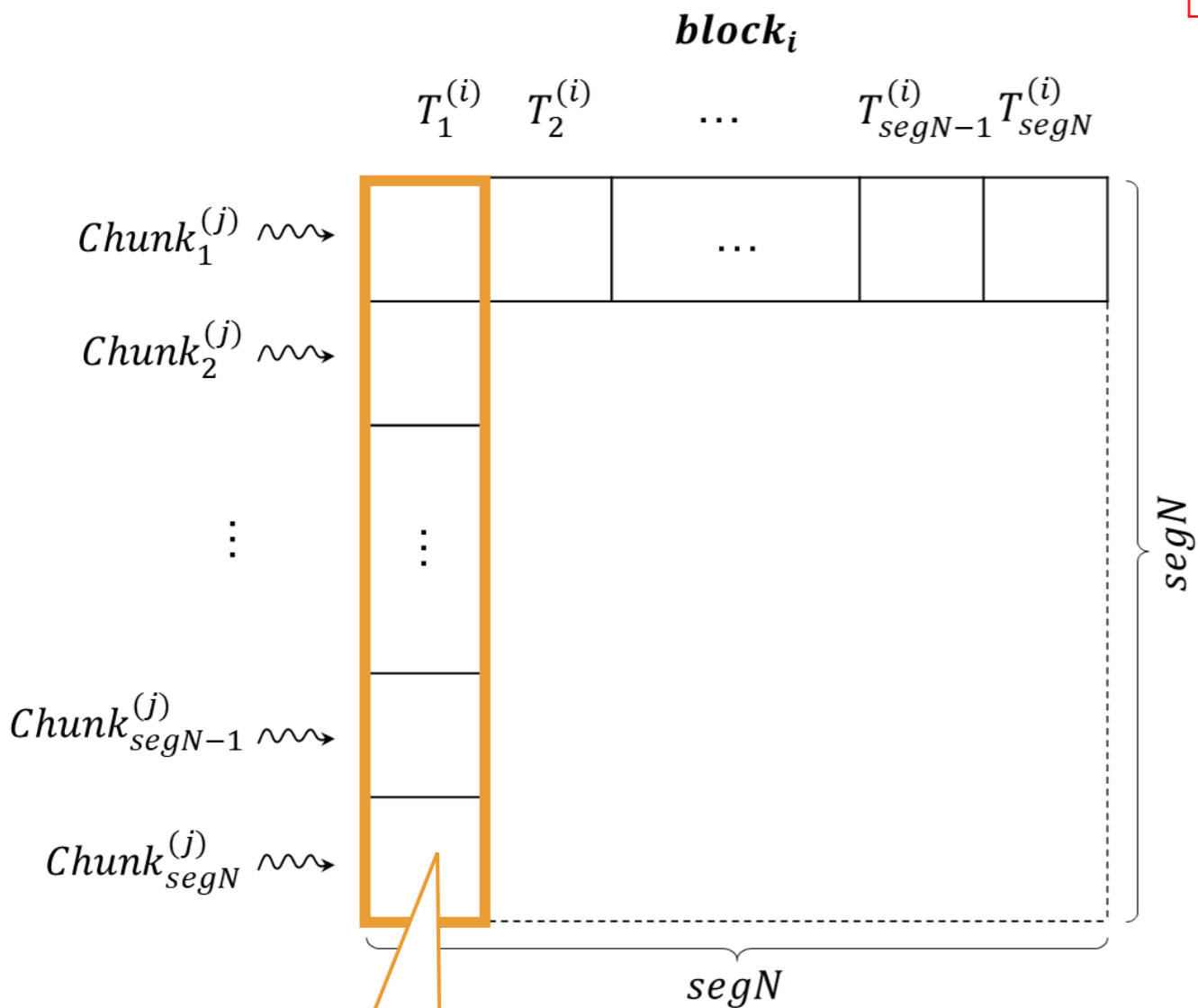


# PD3: Отбор кандидатов, распараллеливание по нитям



# PD3: Отбор кандидатов, вычисление первого столбца

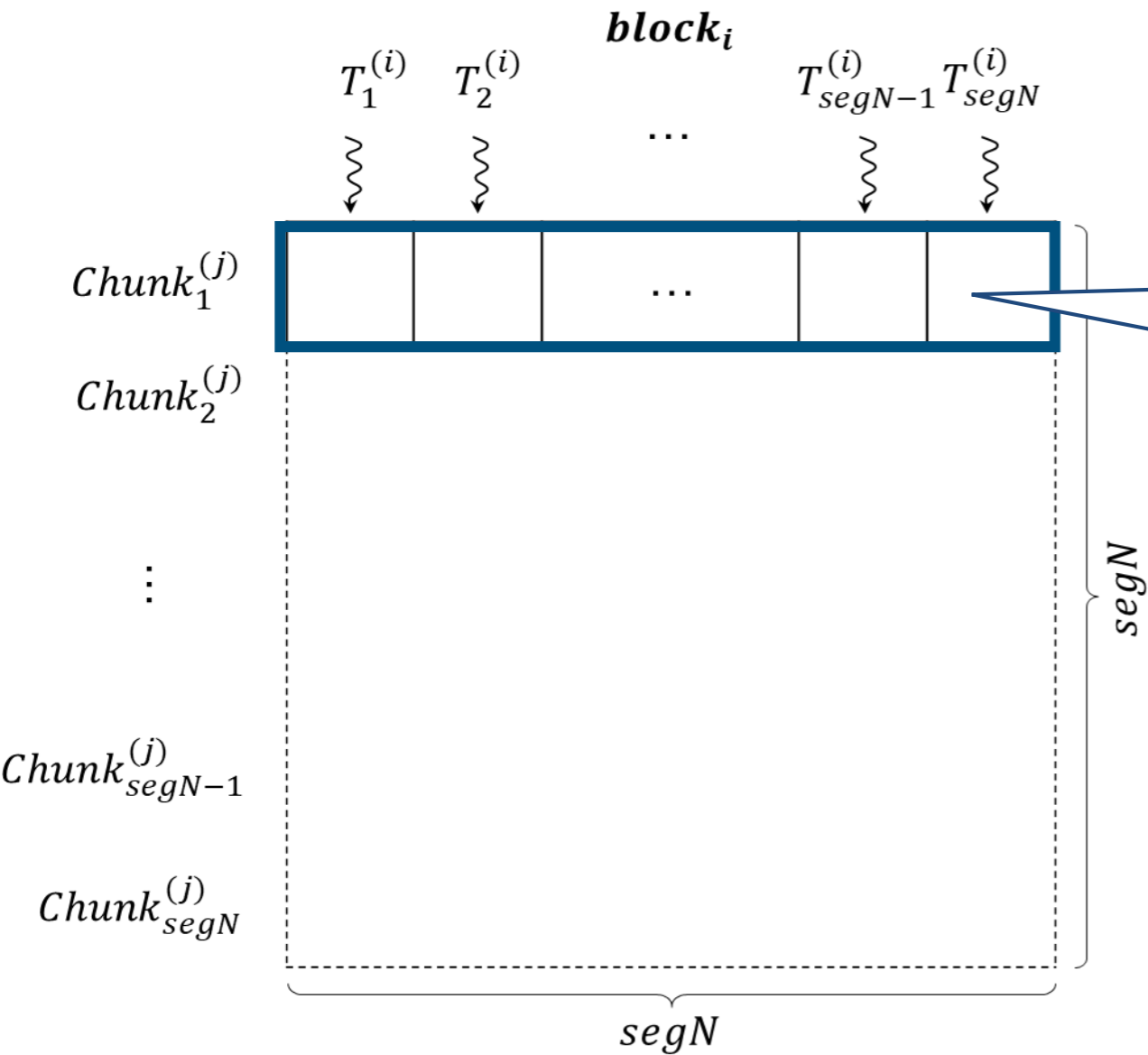
$$d_{i,j} = ED_{\text{norm}}^2(T_{i,m}, T_{j,m}) = 2m \left( 1 - \frac{\langle T_{i,m}, T_{j,m} \rangle - m\mu_i\mu_j}{m\sigma_i\sigma_j} \right)$$



$$QTcol^{(i)}(tid) = \sum_{k=1}^m T_1^{(i)}(k) \cdot Chunk_{tid}^{(j)}(k)$$

# PD3: Отбор кандидатов, вычисление первой строки

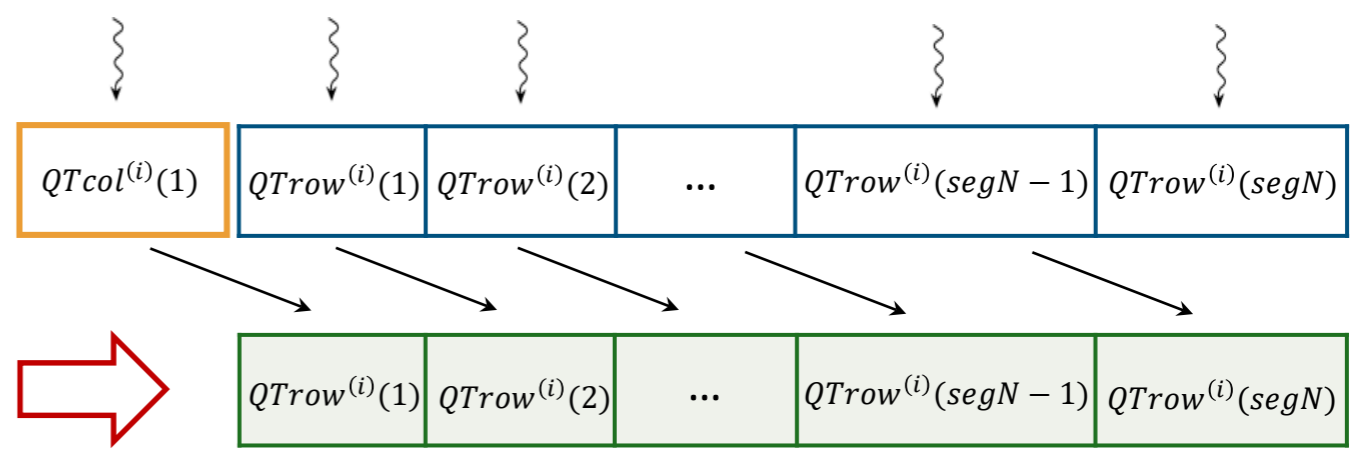
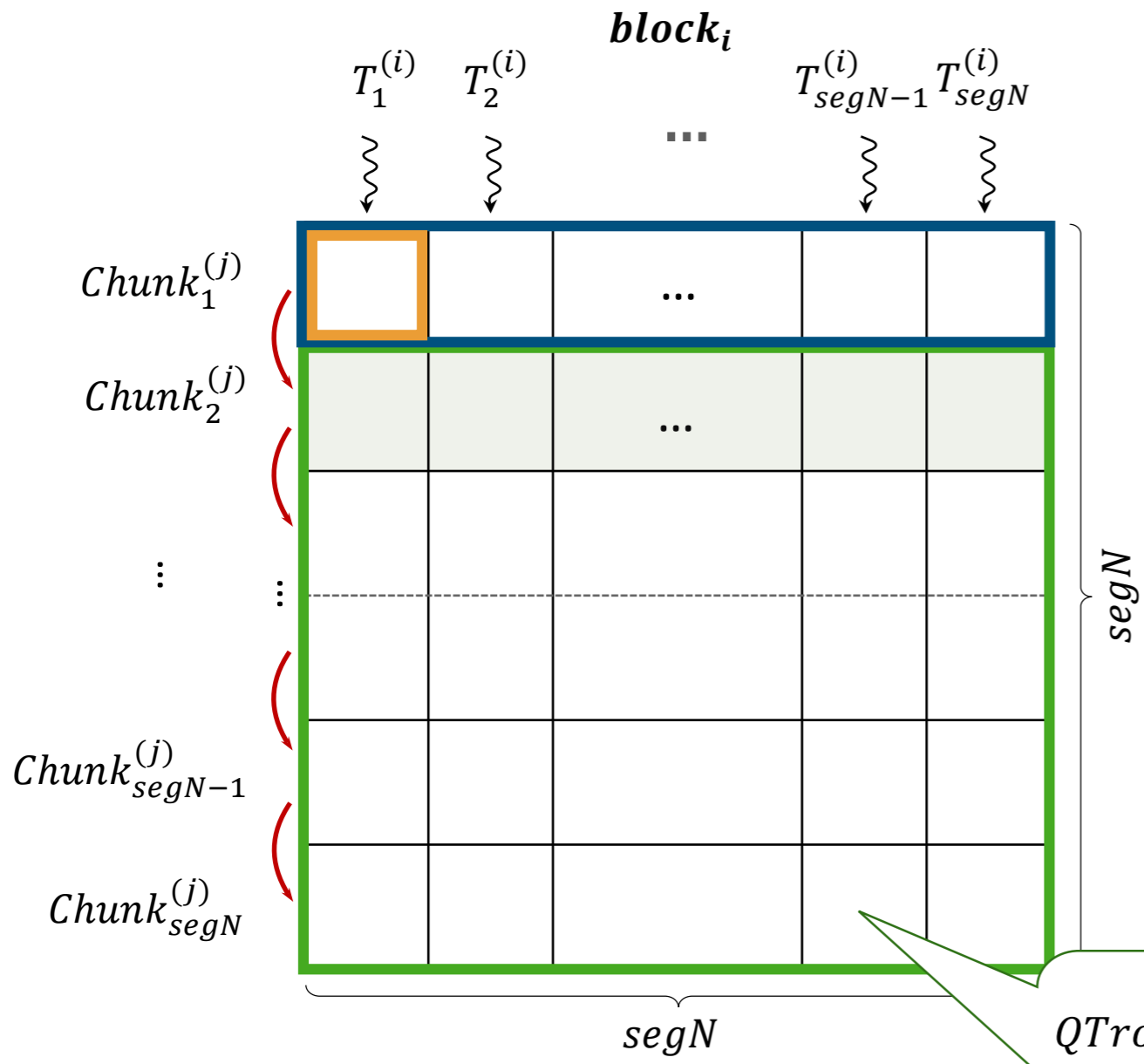
$$d_{i,j} = ED_{\text{norm}}^2(T_{i,m}, T_{j,m}) = 2m \left( 1 - \frac{\langle T_{i,m}, T_{j,m} \rangle - m\mu_i\mu_j}{m\sigma_i\sigma_j} \right)$$



$$QTrow^{(i)}(tid) = \sum_{k=1}^m T_{tid}^{(i)}(k) \cdot Chunk_1^{(j)}(k)$$

# PD3: Отбор кандидатов, вычисление второй строки

$$d_{i,j} = ED_{\text{norm}}^2(T_{i,m}, T_{j,m}) = 2m \left( 1 - \frac{\langle T_{i,m}, T_{j,m} \rangle - m\mu_i\mu_j}{m\sigma_i\sigma_j} \right)$$

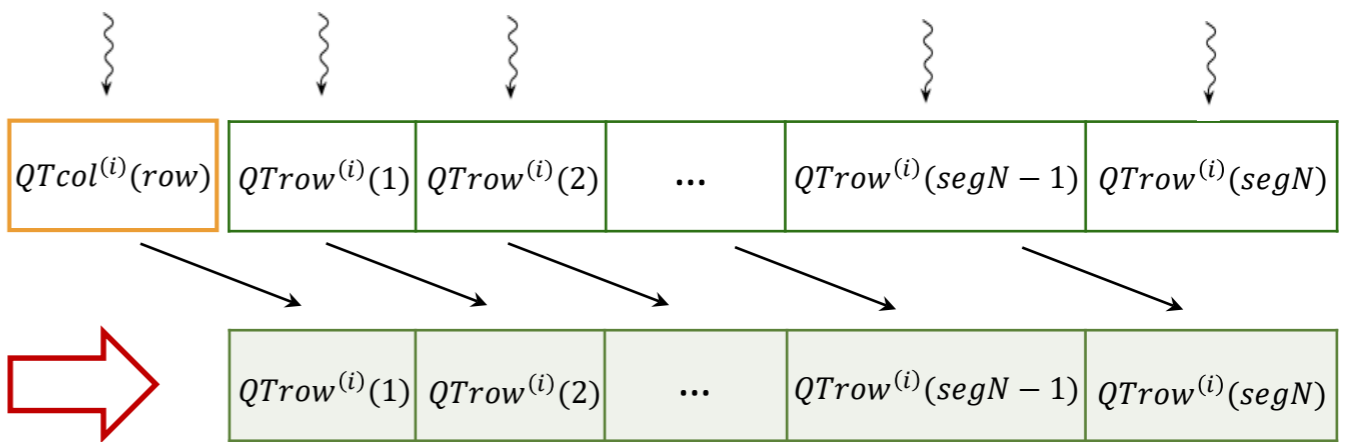
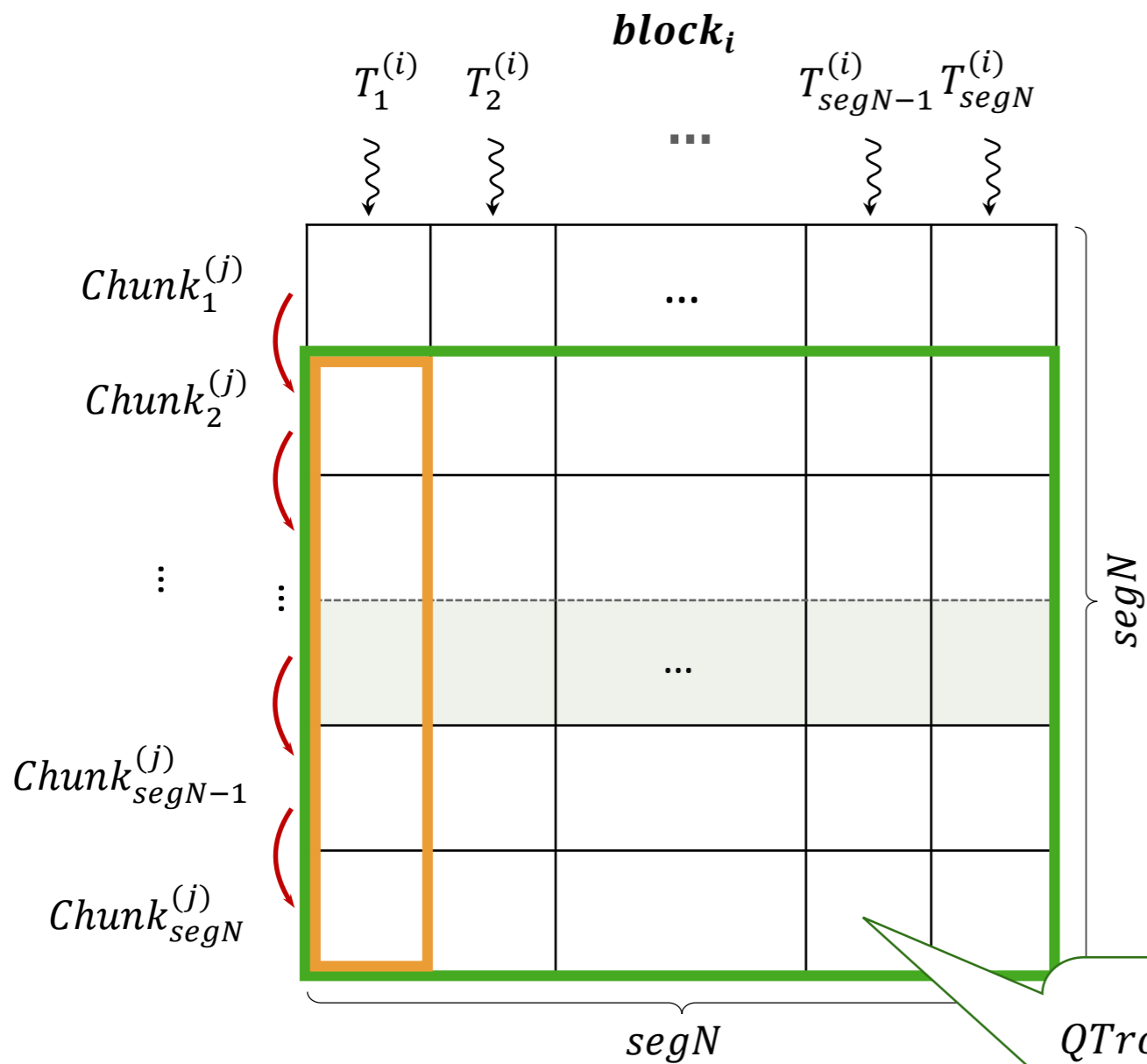


**Выч. сложность O(1) вместо O(m)!**

$$QTrow^{(i)}(tid) = QTrow^{(i)}(tid - 1) - T_{tid-1}^{(i)} \cdot Chunk_{tid-1}^{(j)}(1) + T_{tid}^{(i)} \cdot Chunk_{tid}^{(j)}(m)$$

# PD3: Отбор кандидатов, вычисление $i$ -й строки

$$d_{i,j} = ED_{\text{norm}}^2(T_{i,m}, T_{j,m}) = 2m \left( 1 - \frac{\langle T_{i,m}, T_{j,m} \rangle - m\mu_i\mu_j}{m\sigma_i\sigma_j} \right)$$



**Выч. сложность  $O(1)$  вместо  $O(m)$ !**

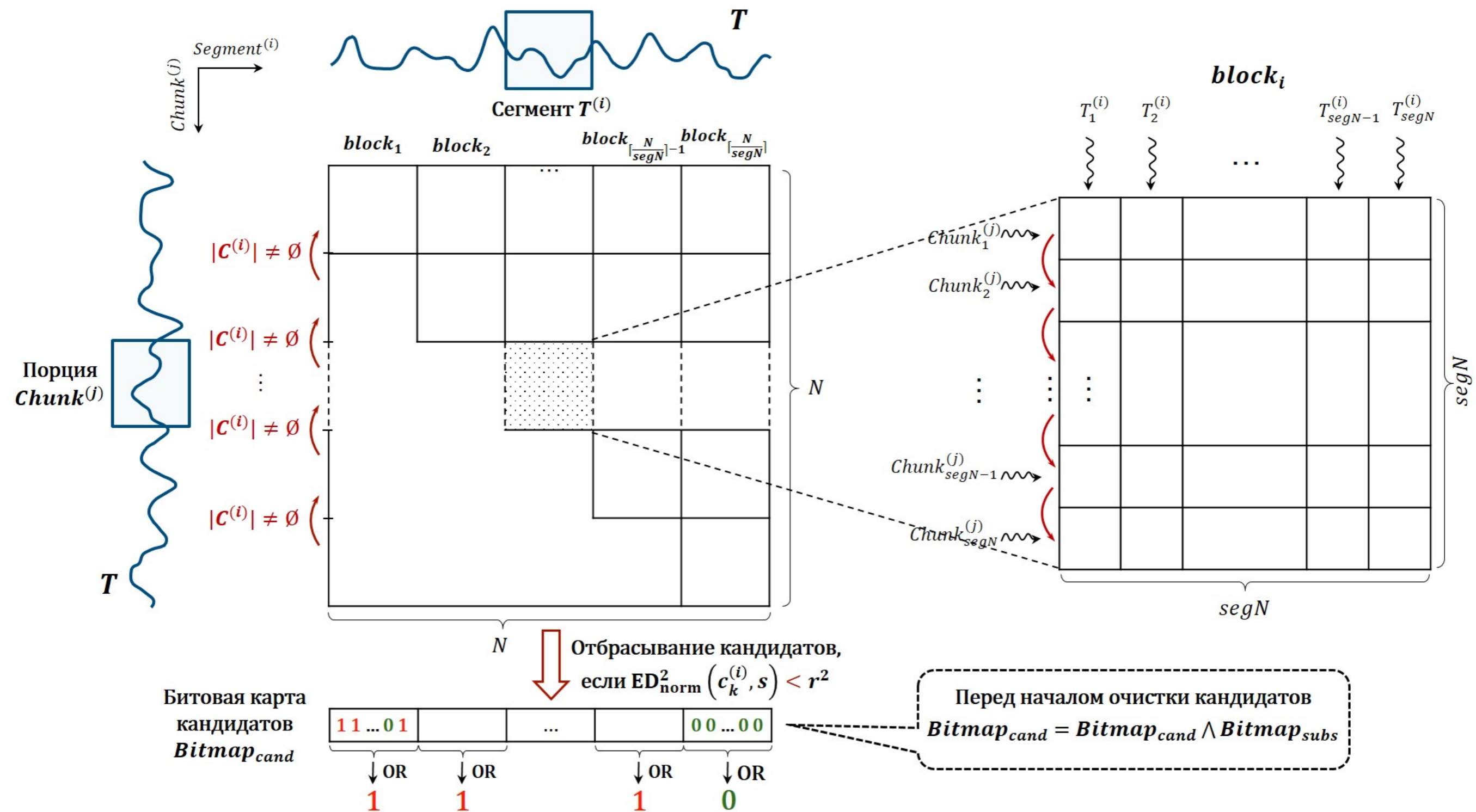
$$QTrow^{(i)}(tid) = QTrow^{(i)}(tid-1) - T_{tid-1}^{(i)} \cdot Chunk_{tid-1}^{(j)}(1) + T_{tid}^{(i)} \cdot Chunk_{tid}^{(j)}(m)$$



# РДЗ: Очистка кандидатов

Блочное распараллеливание

Распараллеливание по нитям



# Эксперименты

- **Аналоги** (поиск top-1 диссонанса на GPU)

- KBF\_GPU

Thuy T.T.H. et al. A new discord definition and an efficient time series discord detection method using GPUs. ICSED 2021. pp. 63–70. <https://doi.org/10.1145/3507473.3507483>.

- Zhu et al.

Zhu B. et al. A GPU Acceleration framework for motif and discord based pattern mining. IEEE Transactions on Parallel and Distributed Systems 32(8): 1987–2004. 2021. <https://doi.org/10.1109/TPDS.2021.3055765>.

- **Данные**<sup>1,2)</sup>

Временной ряд	Длина ряда, $n$	Длина диссонанса, $\min L = \max L$
Space shuttle	5 000	150
ECG	45 000	200
ECG2	21 600	400
Koski-ECG	100 000	458
Power demand	33 220	750
Respiration	24 125	250
RandomWalk1M	$1 \cdot 10^7$	512
RandomWalk2M	$2 \cdot 10^7$	512

- **Аппаратные платформы**

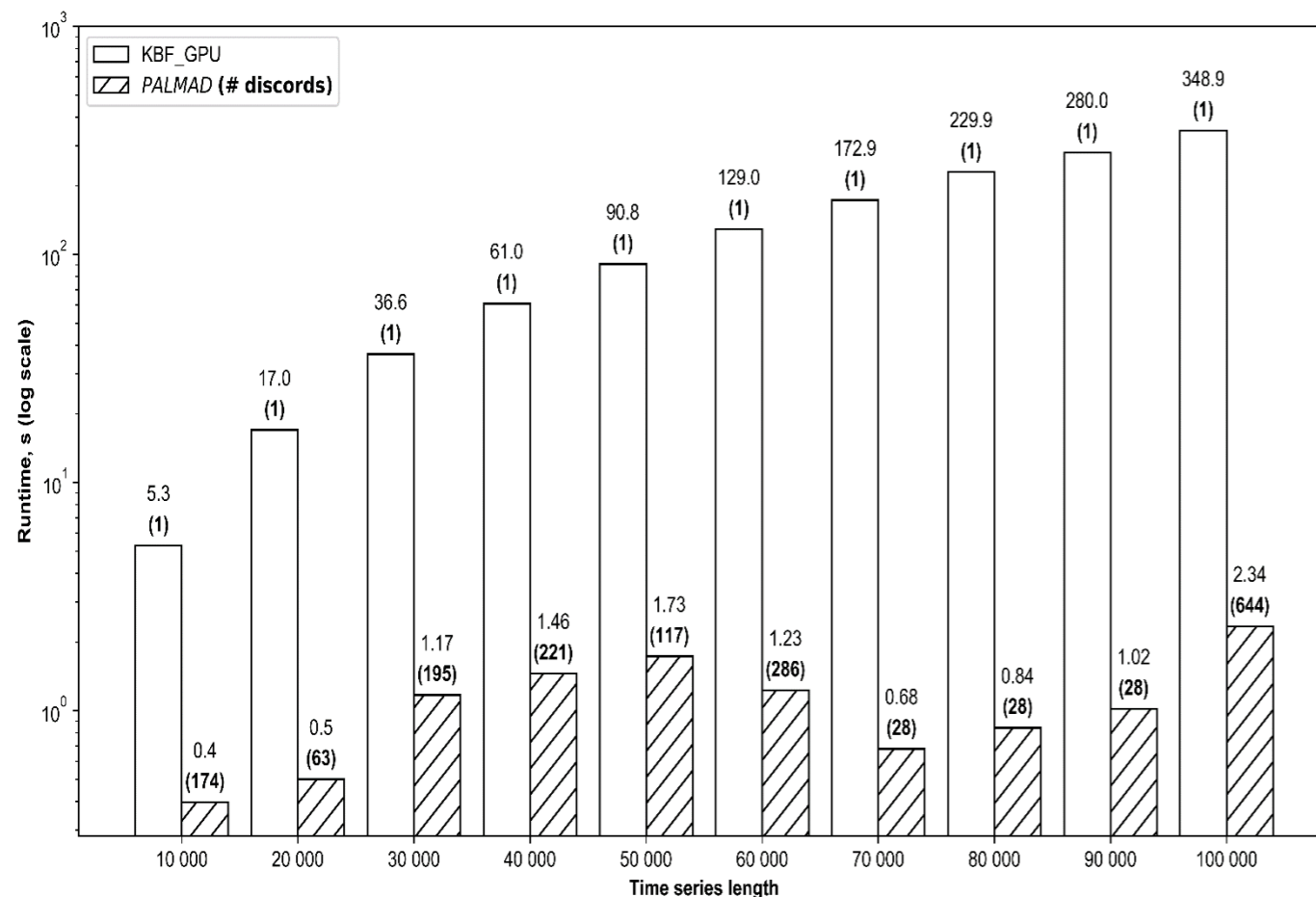
Характеристика	GPU-МГУ	GPU-ЮУрГУ
Производитель, семейство	NVIDIA Tesla	
Модель	P100	V100
# CUDA-ядер	3 584	5 120
Тактовая частота, GHz	1.19	1.3
Оперативная память, Gb	16	32
Пик. пр-ть (double), TFLOPS	4	7

<sup>1)</sup> Keogh E., Lin J., Fu A. HOT SAX: Finding the most unusual time series subsequence: Algorithms and applications. Proc. 5th IEEE Int. Conf. Data Mining 2004: 440–449. URL: <http://www.cs.ucr.edu/~eamonn/discords/>.

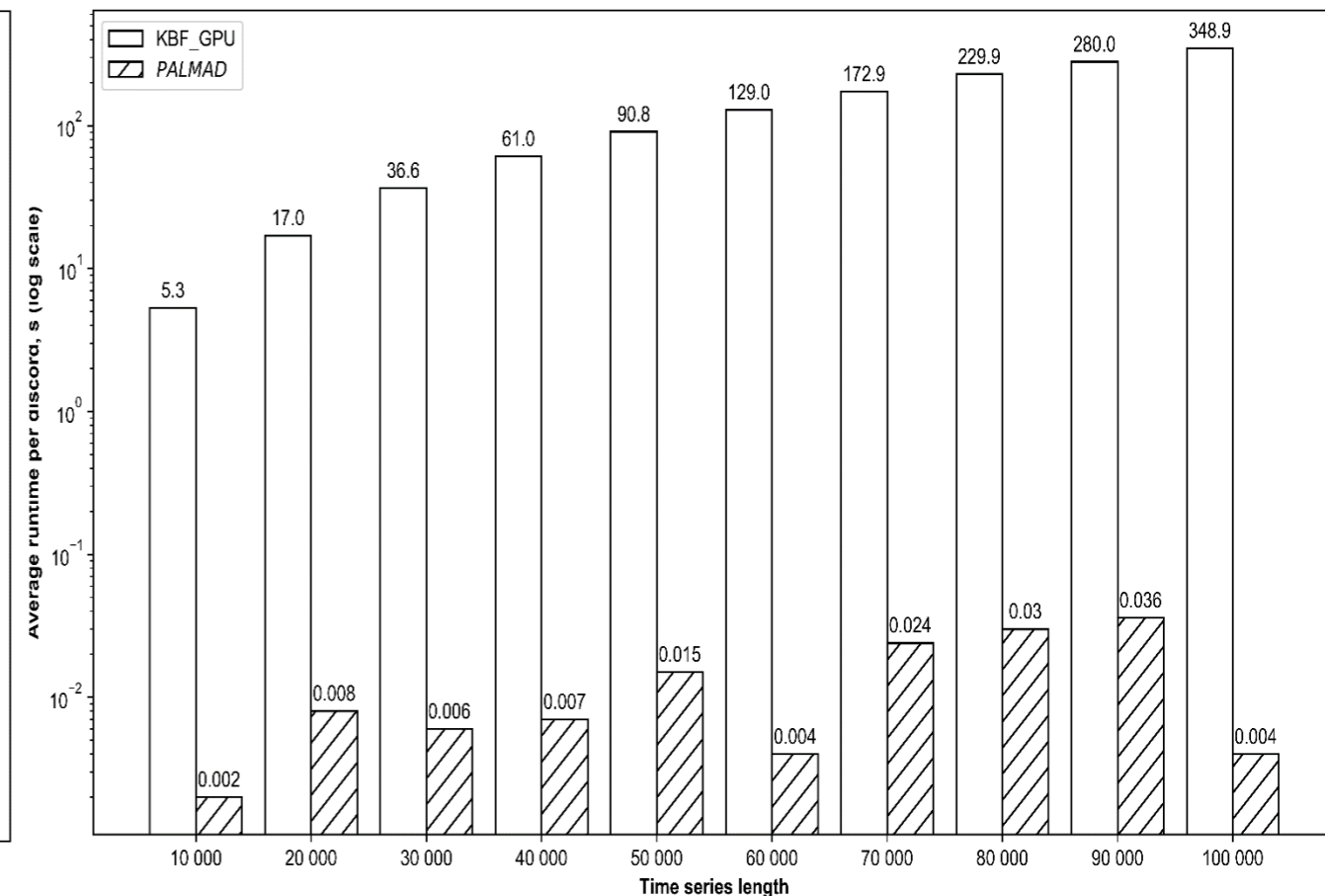
<sup>2)</sup> Pearson K. The problem of the random walk. Nature 72(394). <https://doi.org/10.1038/072342a0>.

# Производительность: сравнение с KBF\_GPU<sup>1)</sup>

## Время на поиск всех диссонансов



## Среднее время на поиск одного диссонанса

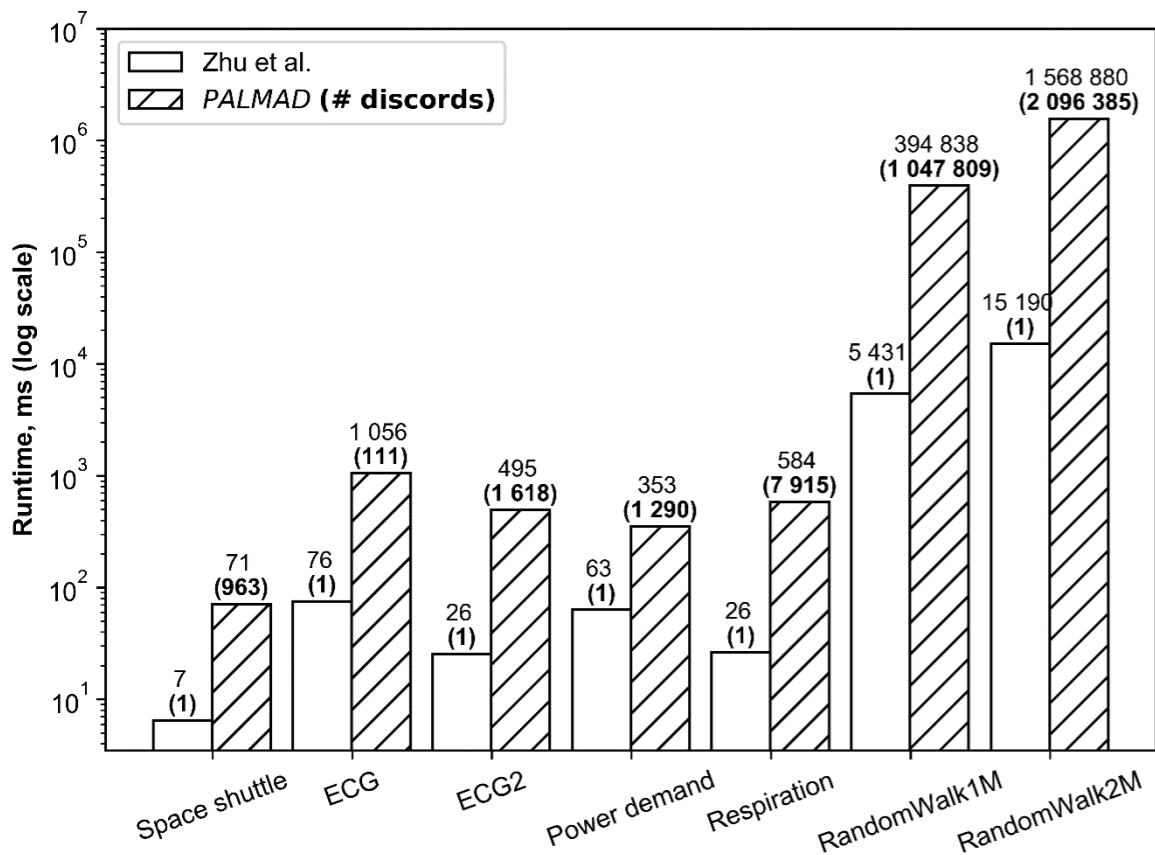


PALMAD опережает KBF\_GPU как по общему времени работы, так и по среднему времени на поиск одного диссонанса

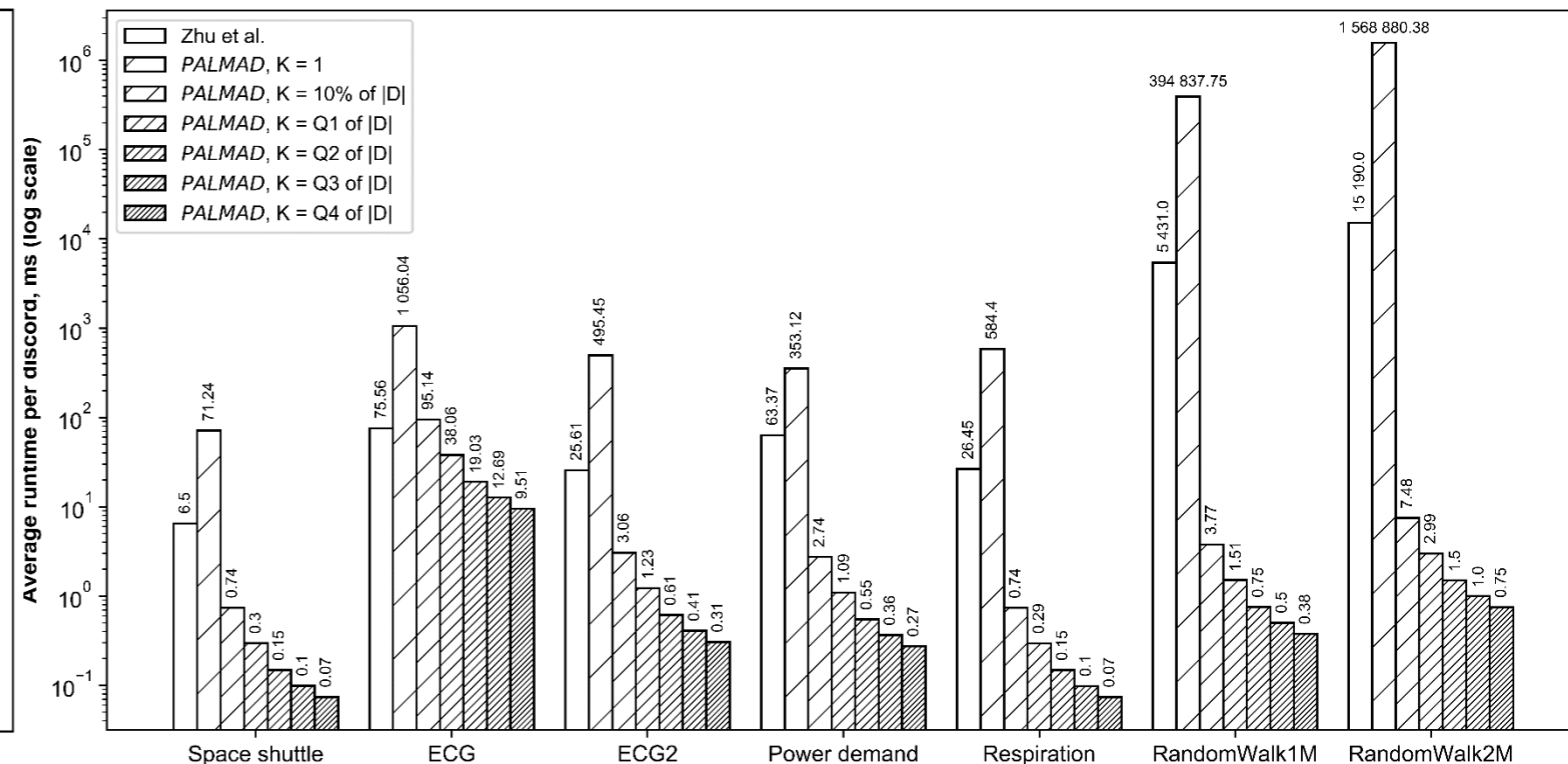
<sup>1)</sup> Thuy T.T.H. et al. A new discord definition and an efficient time series discord detection method using GPUs. ICSED 2021. pp. 63–70. <https://doi.org/10.1145/3507473.3507483>.

# Производительность: сравнение с Zhu et al.<sup>1)</sup>

## Время на поиск **всех** диссонансов



## Среднее время на поиск **одного** диссонанса

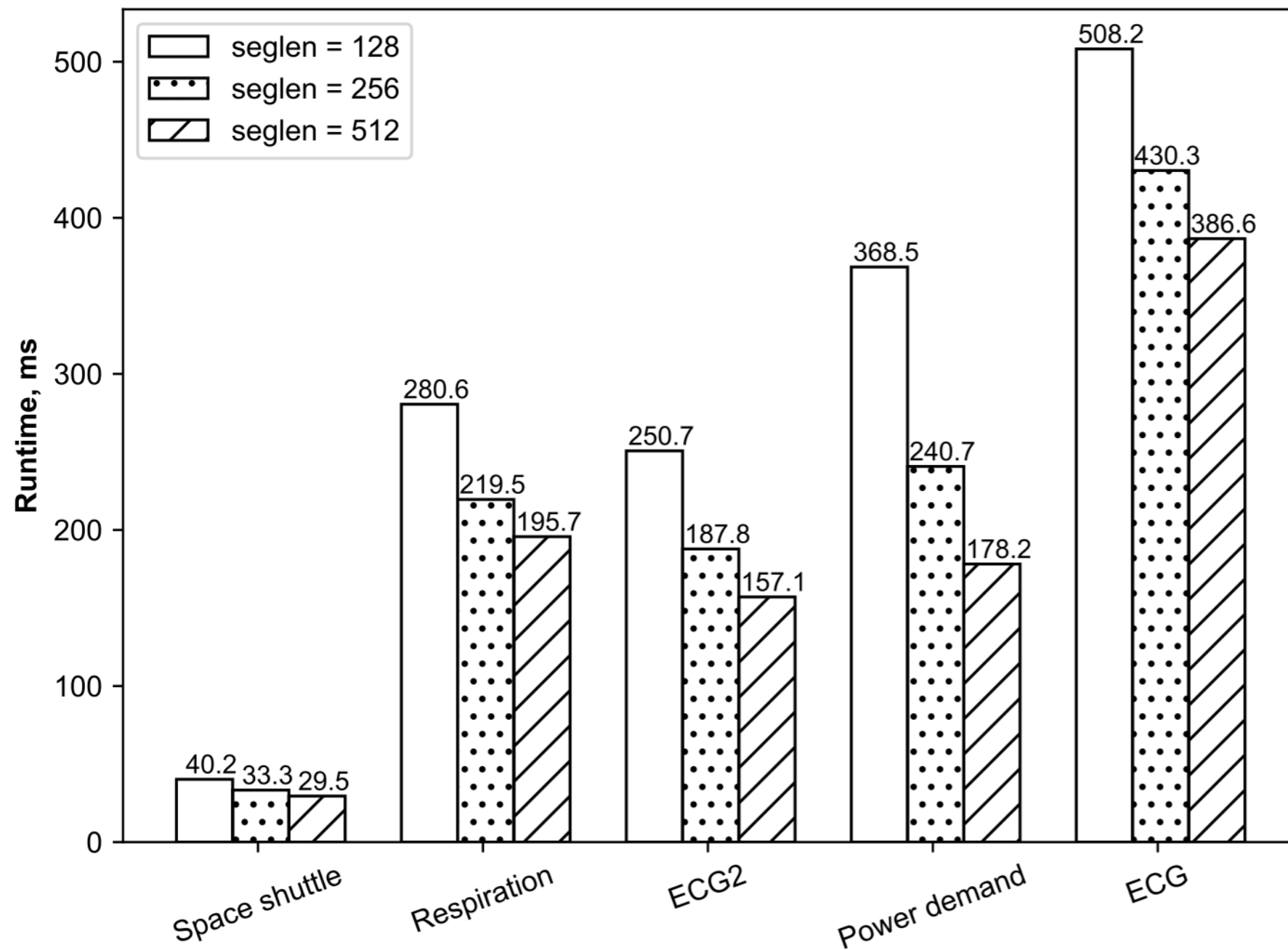


PALMAD отстает от алгоритма Zhu et al. по общему времени работы, но значительно опережает его по среднему времени на поиск одного диссонанса, начиная с поиска top- $k$  диссонансов, где  $k=10\%$  от фактического числа диссонансов

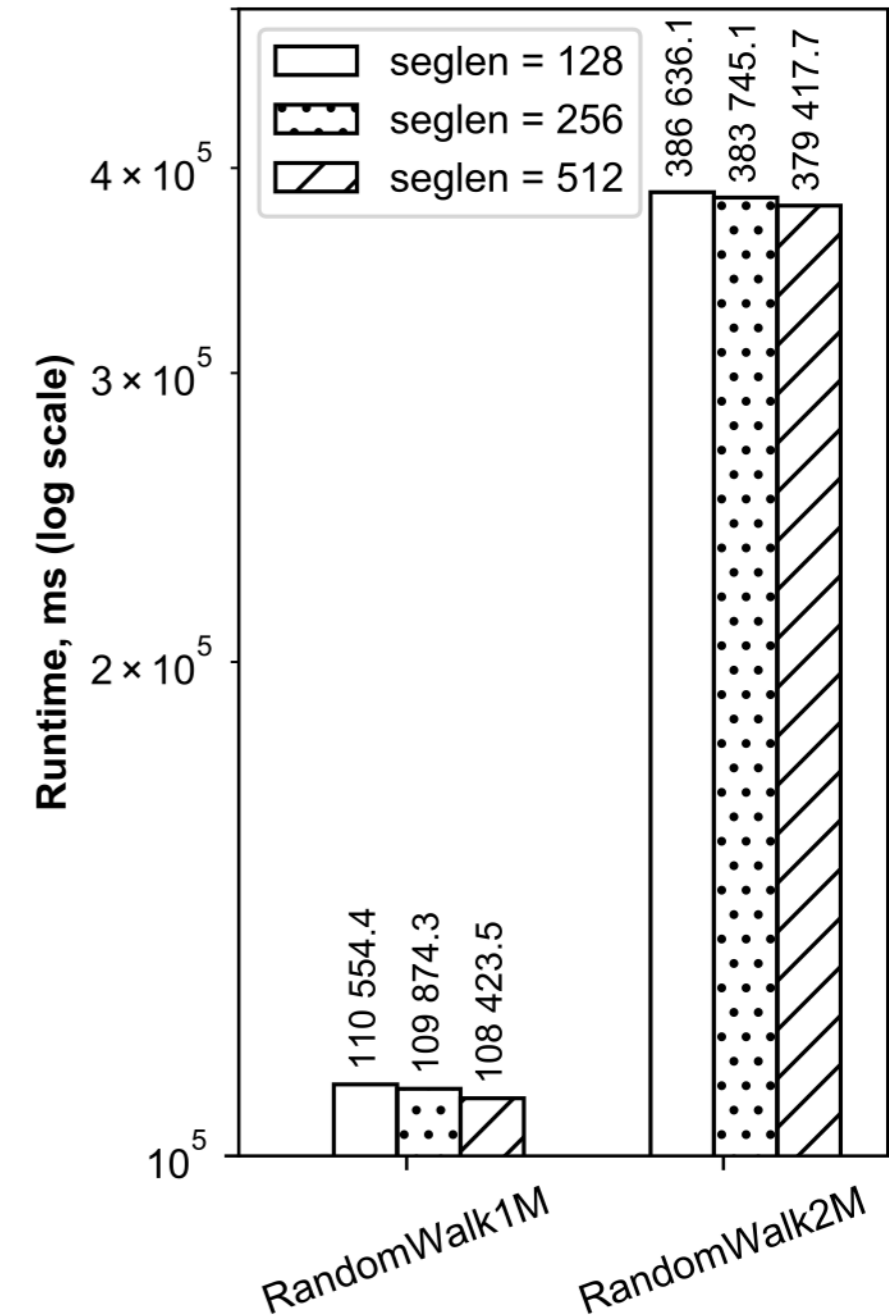
<sup>1)</sup> Zhu B. et al. A GPU Acceleration framework for motif and discord based pattern mining. IEEE Transactions on Parallel and Distributed Systems 32(8): 1987-2004. 2021. <https://doi.org/10.1109/TPDS.2021.3055765>.

# Производительность: влияние длины сегмента

## Реальные временные ряды



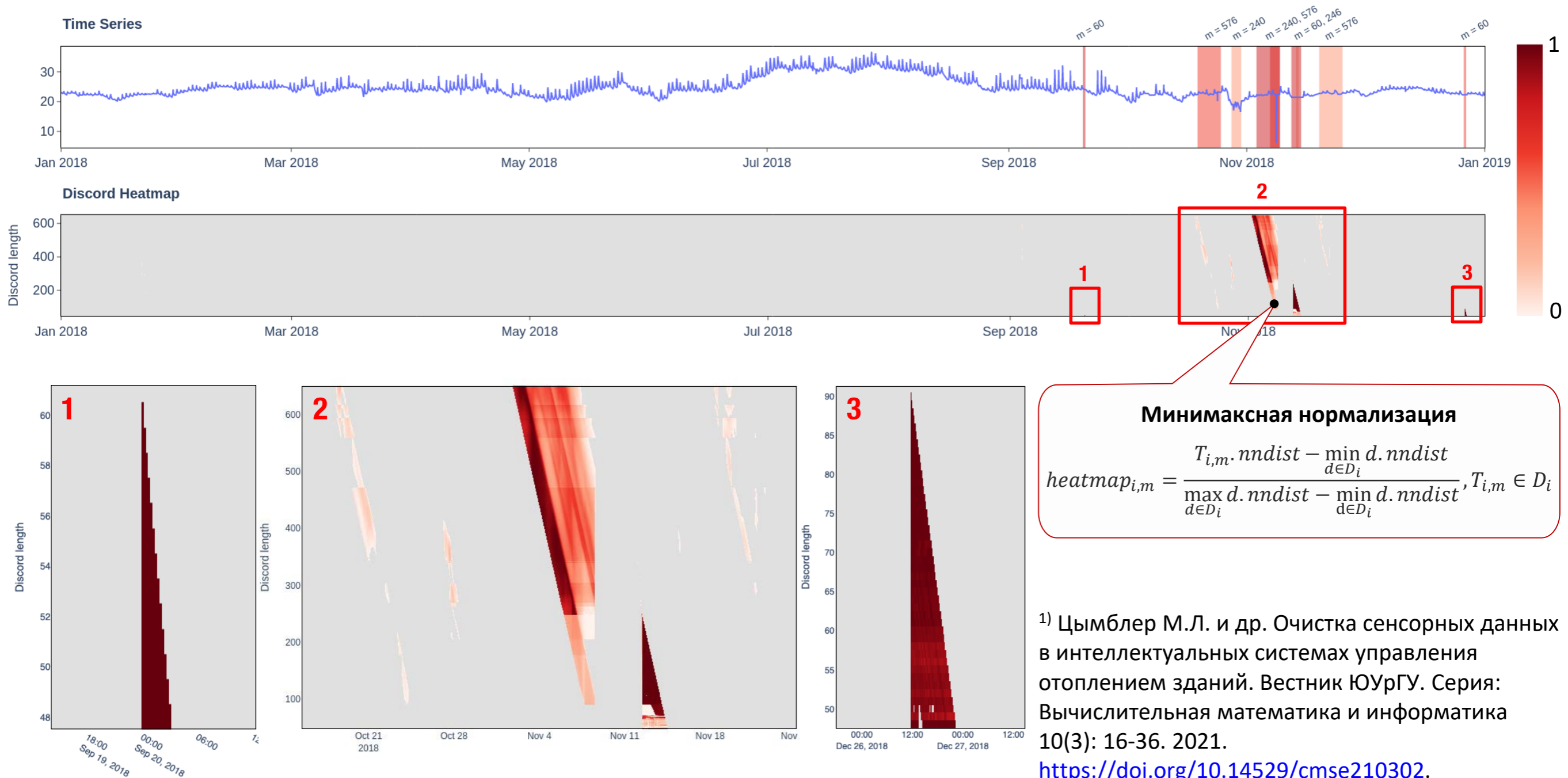
## Синтетические временные ряды



Производительность алгоритма пропорциональна длине сегмента

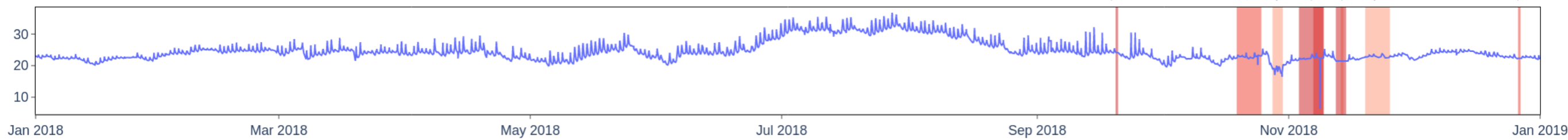
# Тематическое исследование: PolyTER<sup>1)</sup>

- **Данные:** показания температурного датчика (частота 4 раза в час) в ауд. 808/36, за январь-декабрь 2018 г.
- **Диапазон длин искомым диссонансов:** 0.5–7 суток



# Тематическое исследование: PolyTER

Time Series

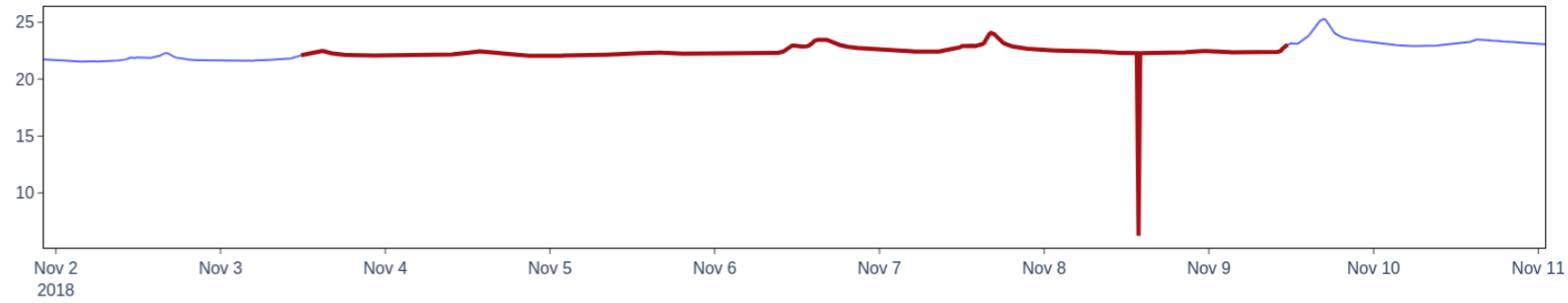
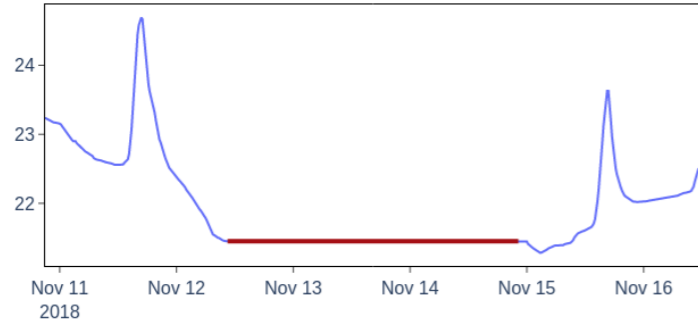
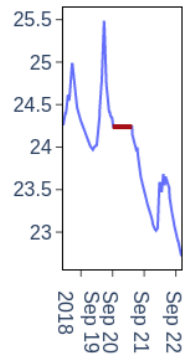


$m = 60$   
(0.5 суток)

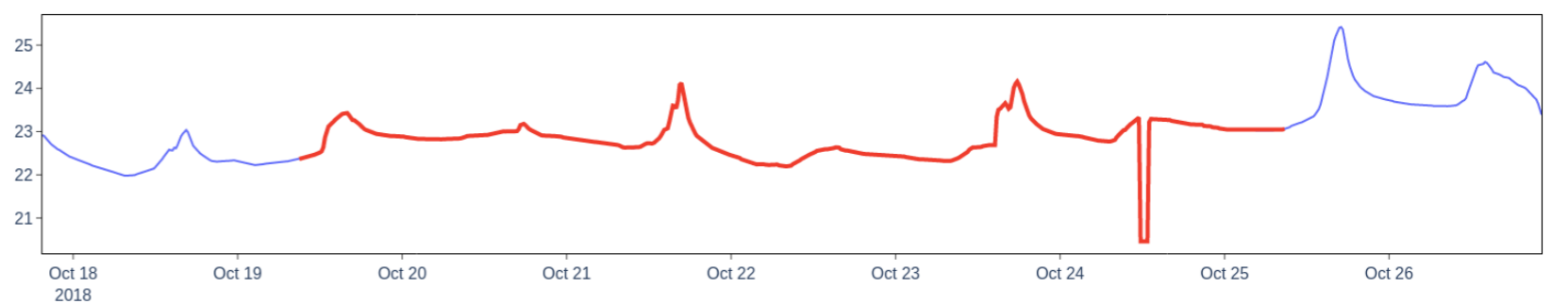
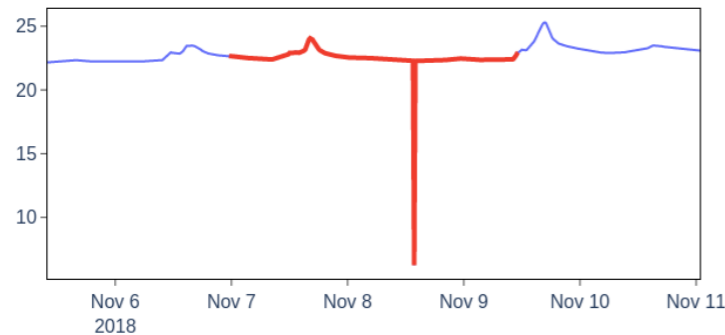
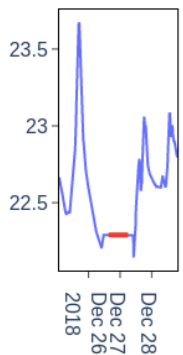
$m = 246$   
(2.5 суток)

$m = 576$   
(6 суток)

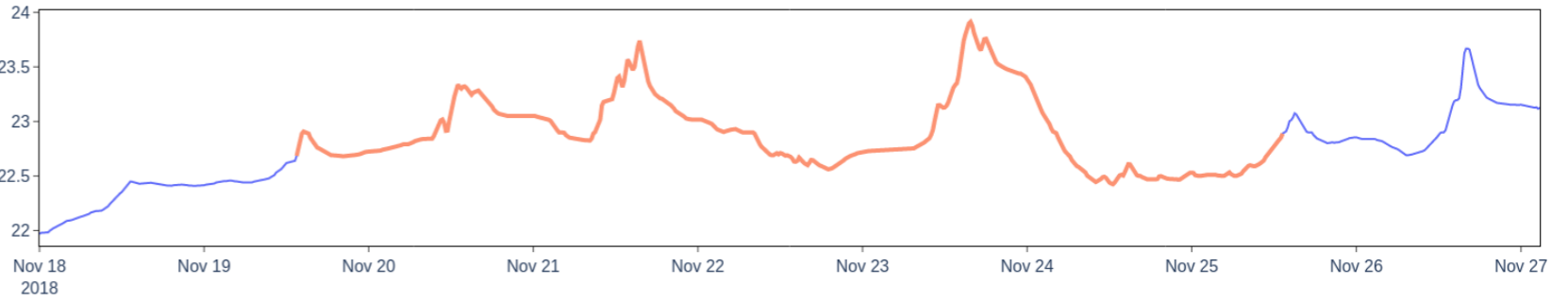
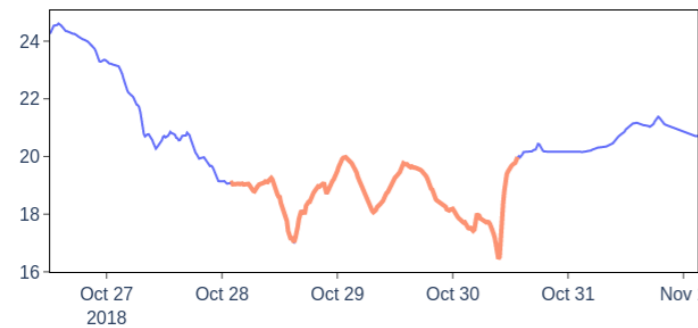
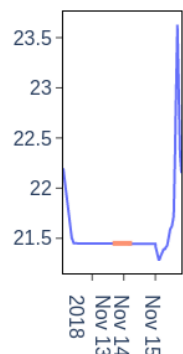
Тор-1 диссонанс



Тор-2 диссонанс



Тор-3 диссонанс



# Продолжение исследований

- Мера полезности диссонанса

$$Interest(T_{i,m} \in \mathcal{D}) = \begin{cases} \frac{heatmap_{i,m}}{norm(m)}, & m = maxL \\ \frac{heatmap_{i,m}}{norm(m)} \cdot \frac{1}{\sum_{m < len < maxL} Interest(T_{i,len} \in \mathcal{D})}, & minL \leq m < maxL \end{cases}$$

- Среди диссонансов с различными длинами считаем более ценными те (с учетом степени их аномальности), которые имеют меньшую длину и меньшее количество тривиальных совпадений
- Разработка версии PALMAD для высокопроизводительного кластера с узлами на базе GPU
- Применение PALMAD в нейросетевой модели для поиска аномалий временного ряда в режиме реального времени

**Спасибо за внимание! Вопросы?**

Яна Александровна Краева

kraevaya@susu.ru