XXIV International Conference
Data Analytics and Management in Data Intensive Domains (DAMDID/RCDL’2022),
5–7 October 2022, ITMO University, St. Petersburg, Russia

Time series analytics: acceleration with parallel algorithms

Mikhail Zymbler[1], Yana Kraeva[1], Andrey Goglachev[1],
Alexey Yurtin[1], Andrey Poluyanov[2]

[1]South Ural State University, Chelyabinsk, Russia

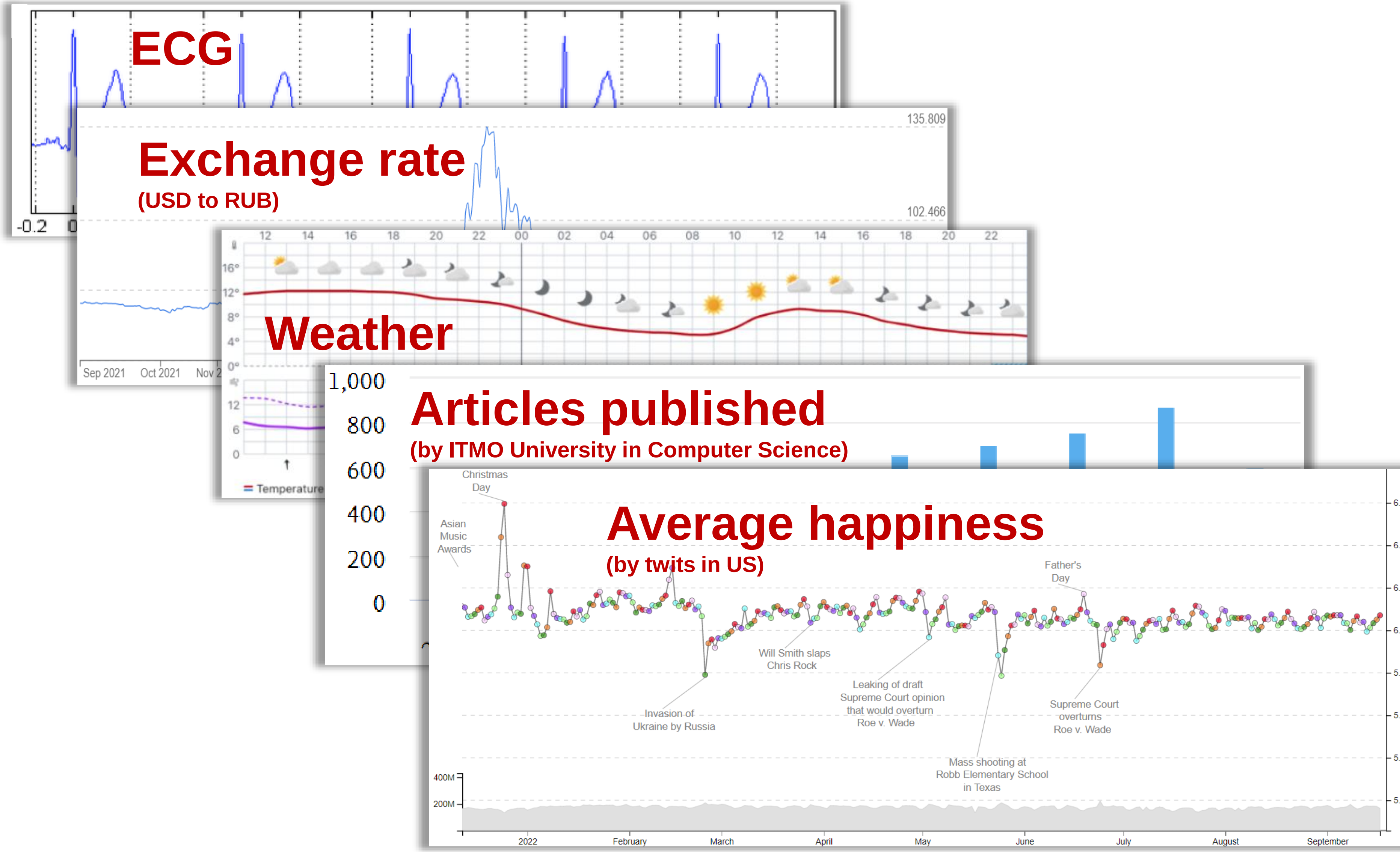
[2]Sobolev Institute of Mathematics SB RAS, Omsk, Russia

This work was financially supported by the Russian Foundation for Basic Research (grant No. 17-07-00463) and by the Ministry of Science and Higher Education of the Russian Federation (government orders FENU-2020-0022 and FWNF-2022-0016).

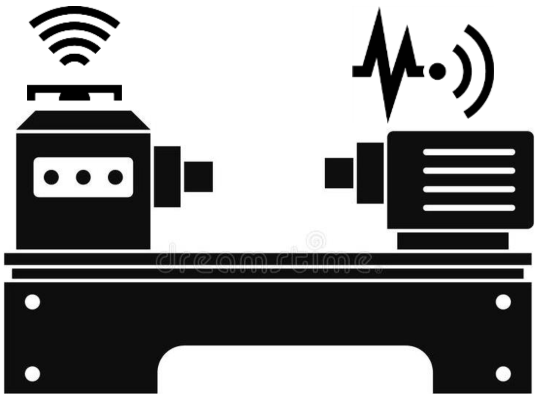
Outline

- **Introduction**
  - What are and what can we mine from time series?
- How can parallel algorithms accelerate time series analytics?
  - Pattern discovery
  - Anomaly detection
  - Imputation of missing values
- Further research: how can parallel algorithms provide online time series analytics?

People measure everything over time, ...

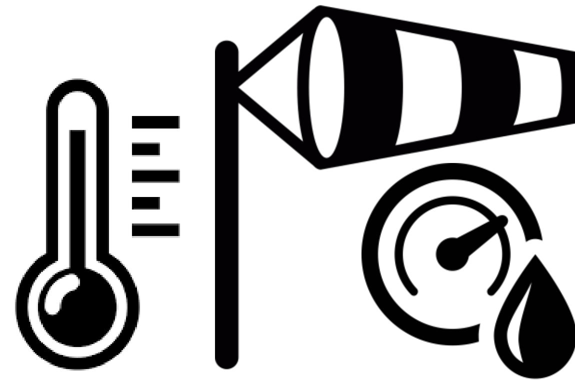


… so, time series are ubiquitous



Smart manufacturing, Predictive maintenance

Internet of Things



Weather forecasting, Climate modelling

Personal healthcare

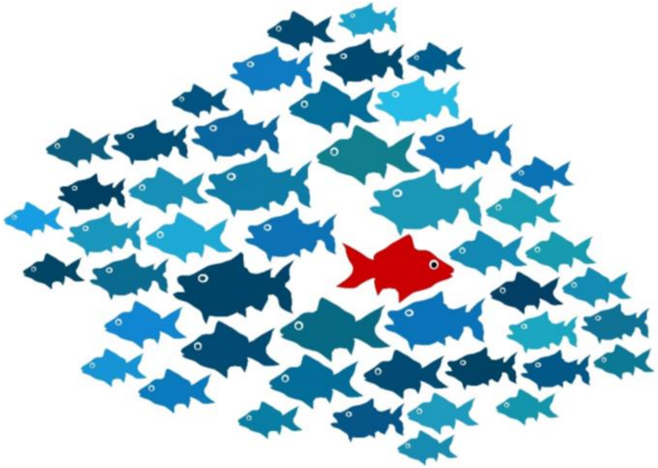
Bioinformatics, Cheminformatics

Business and economics

Learning management systems, Personal educational trajectories

What can we mine from time series?

Patterns

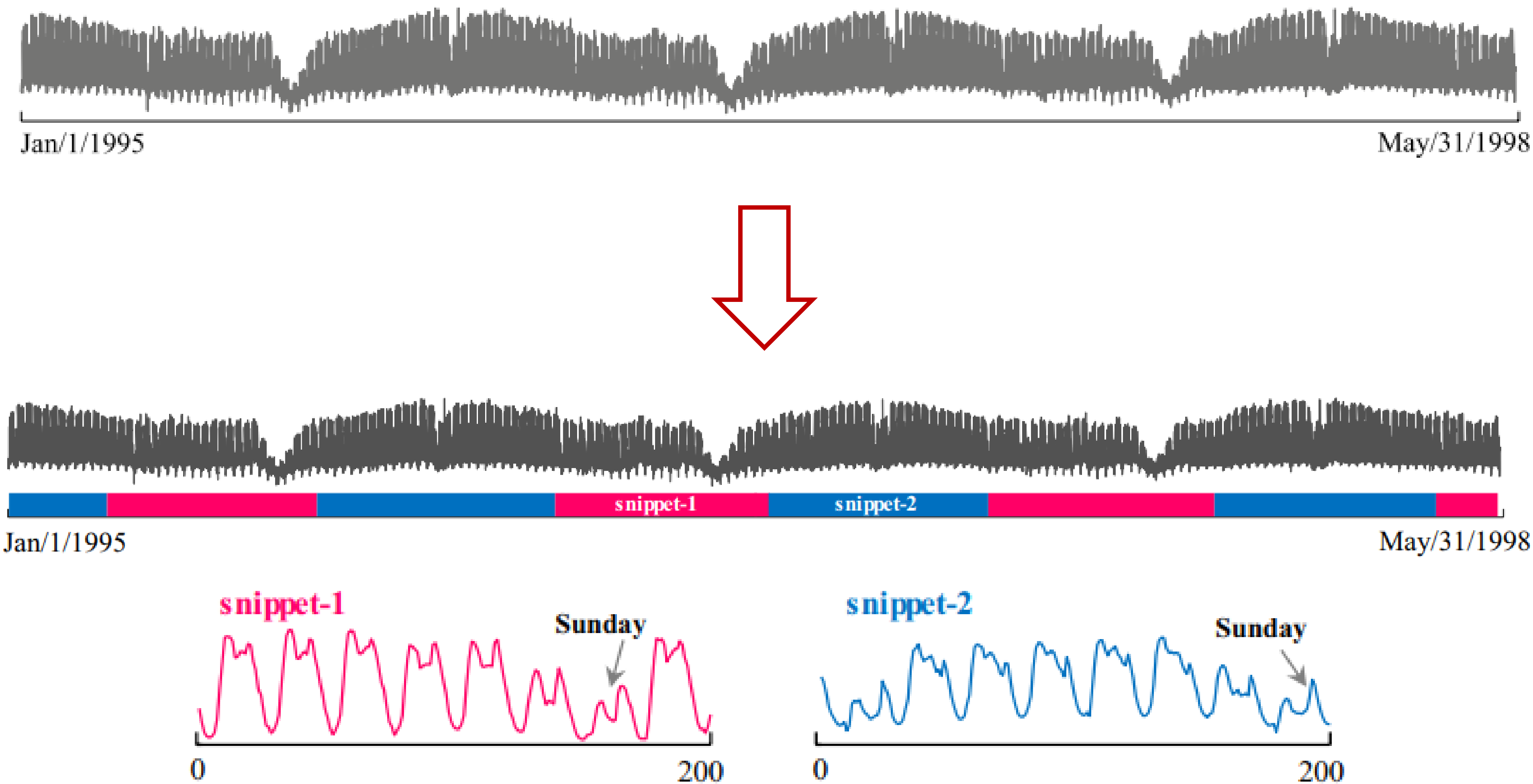


Anomalies

Imputation of missing values

Pattern discovery: power demand

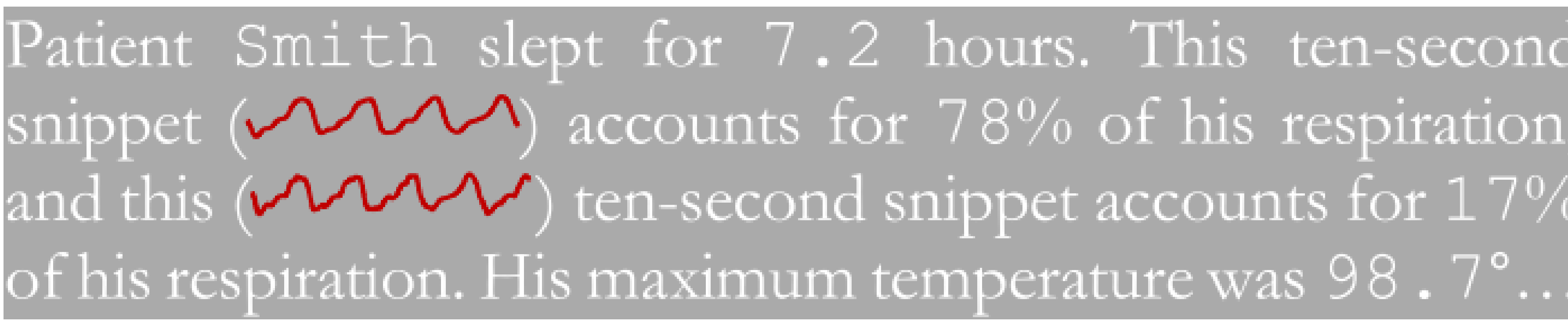
Energy consumption in an Italian city for 3 years



Snippets show typical weekly intervals in warm and cold seasons

Pattern discovery: medicine

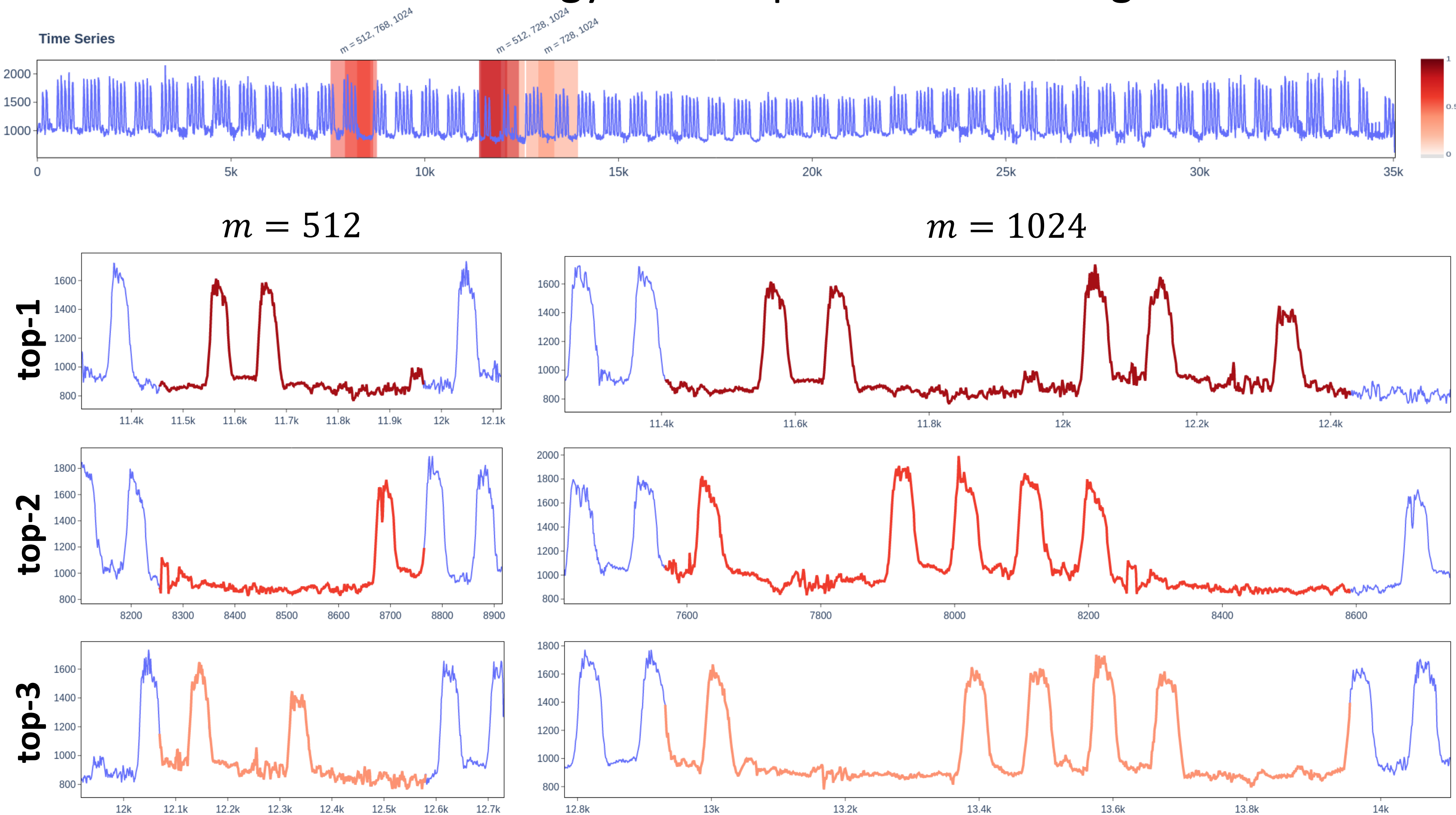
A patient's motor activity according to the chest accelerometer measurements



A patient's respiratory activity in studies of apnea syndrome

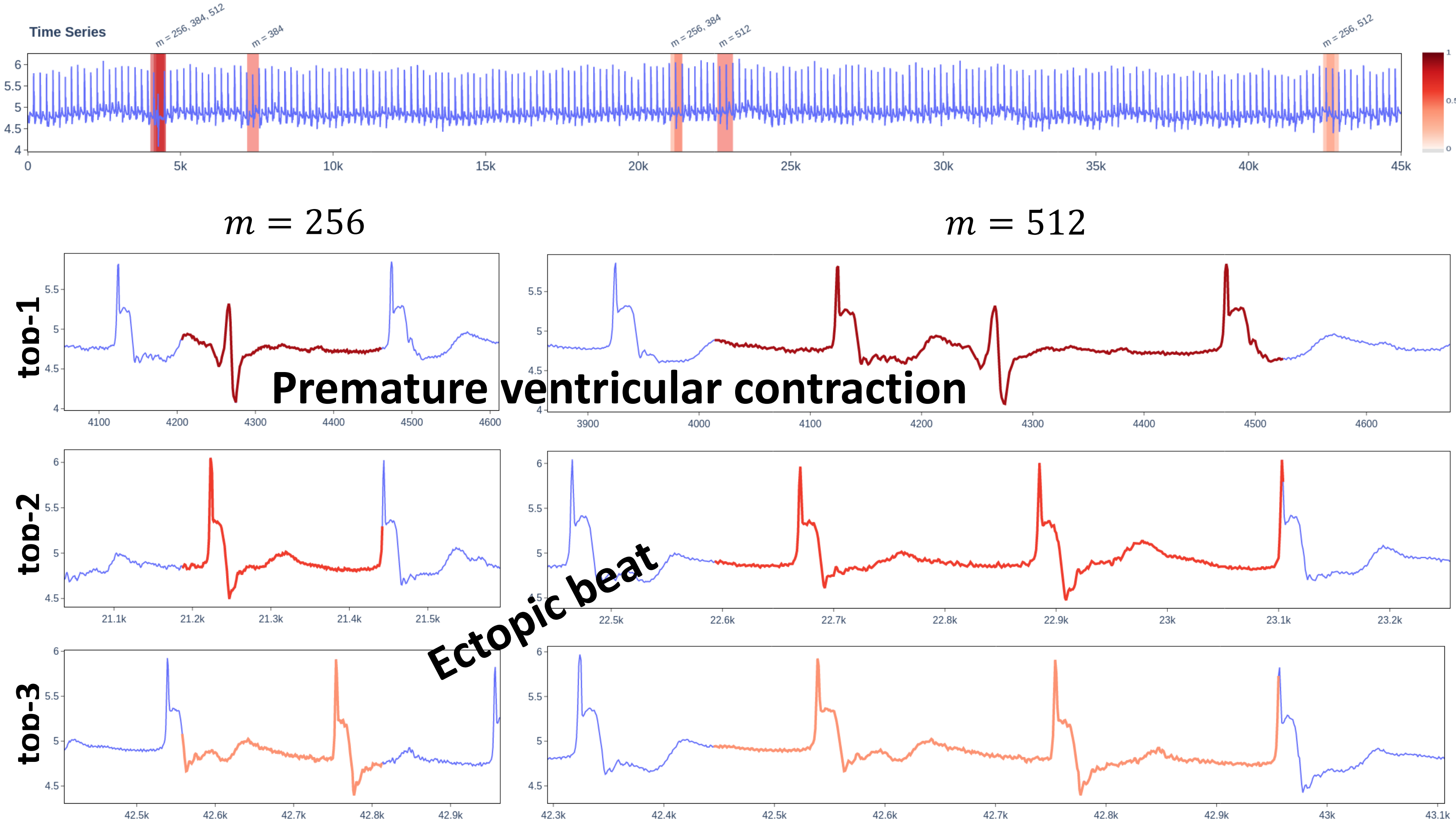
Anomaly detection: power demand

Annual energy consumption of a building



Anomaly detection: medicine

Electrocardiogram of an adult patient



Imputation of missing values

MAREL Carnot system autonomously measures +15 chemical and biological parameters each 20 min. in the English Channel

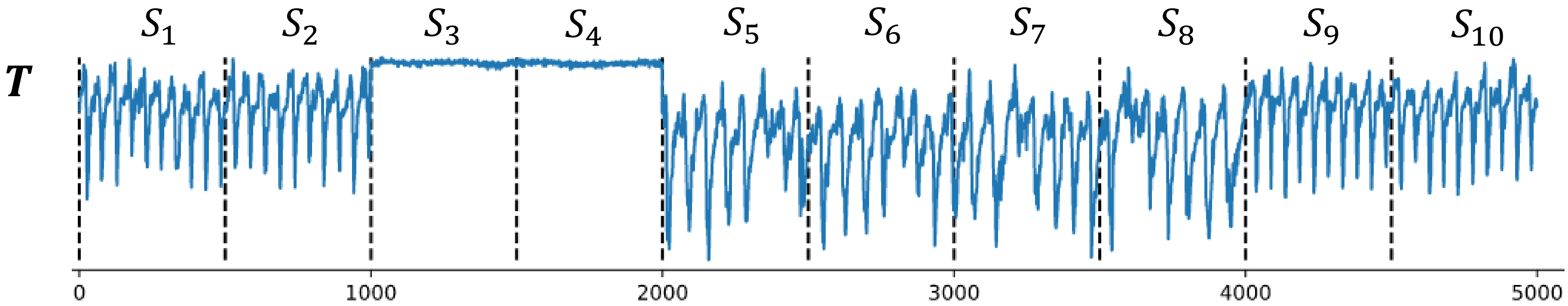
Madrid Road Traffic Management System provides each 15 min. the data from +3500 automatic traffic recorders deployed through the city road network

DEBS challenge: Real-time event-based sports analytics
Positioning sensors of tracking frequency 200 Hz (15K events per second) are located near to players' boots and goalkeeper hands

Outline

- Introduction
- **Parallel pattern discovery**
- Parallel anomaly detection
- Parallel imputation of missing values
- Online time series analytics with parallel algorithms

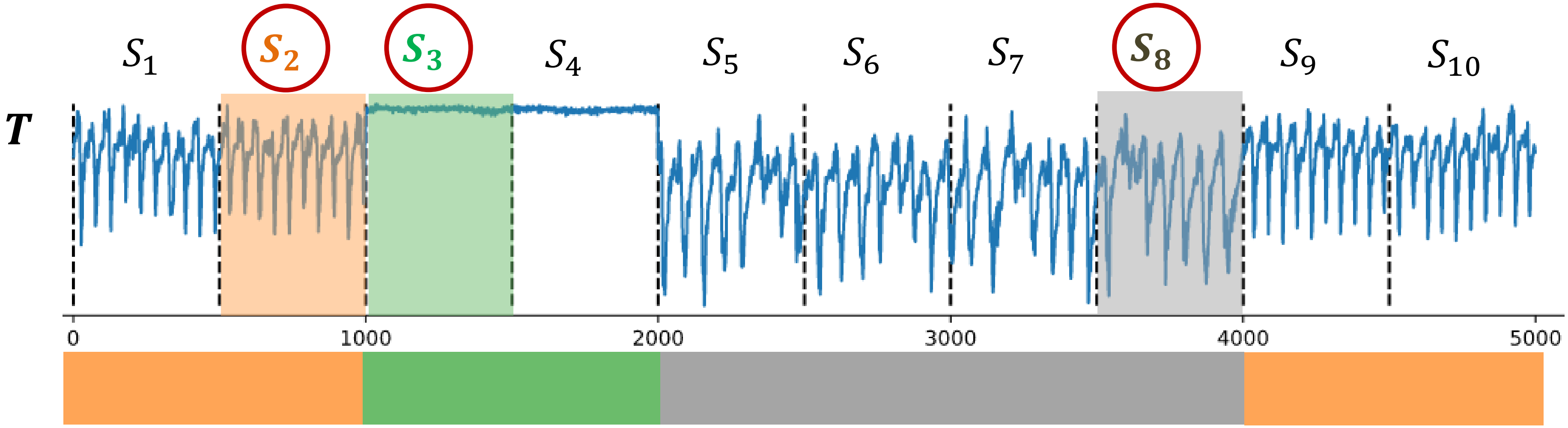
How to formalize patterns? Snippets[*]



1. A time series is represented as a set of non-overlapped segments

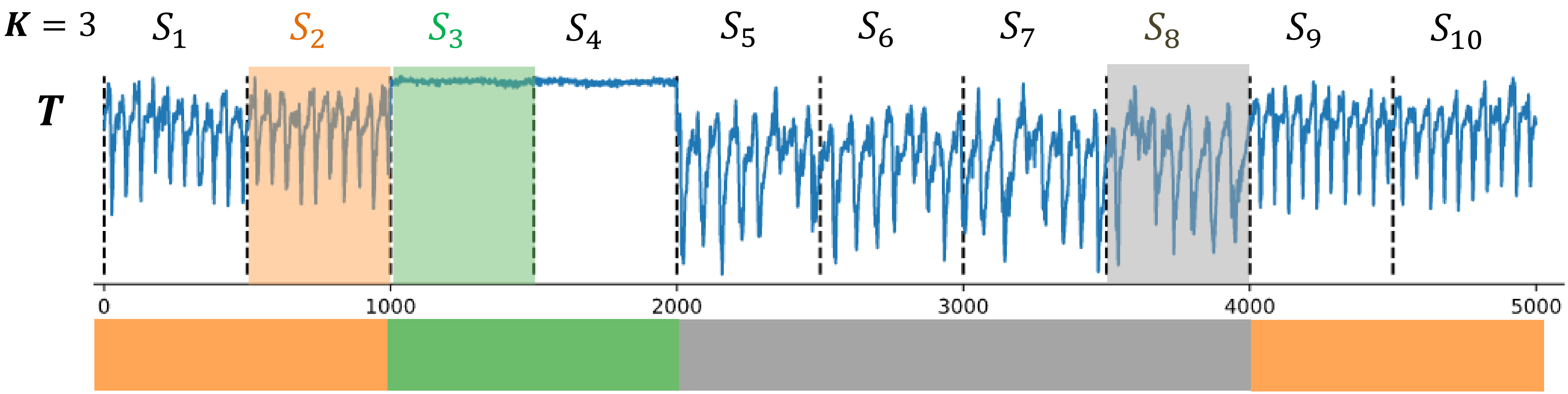
[*] Imani S. *et al.* Introducing time series snippets: a new primitive for summarizing long time series. Data Min. Knowl. Discov. 2020. Vol. 34, no. 6. P. 1713-1743. DOI: [10.1007/s10618-020-00702-y](https://doi.org/10.1007/s10618-020-00702-y)

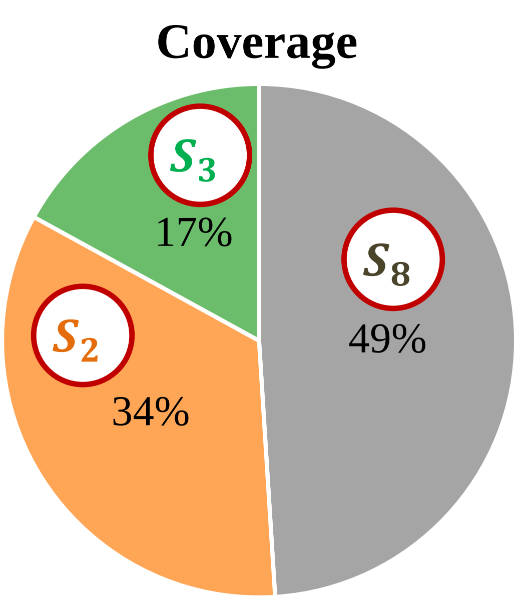
How to formalize patterns? Snippets



1. A time series is represented as a set of non-overlapped segments
2. For each segment, find its **nearest neighbors**

How to formalize patterns? Snippets



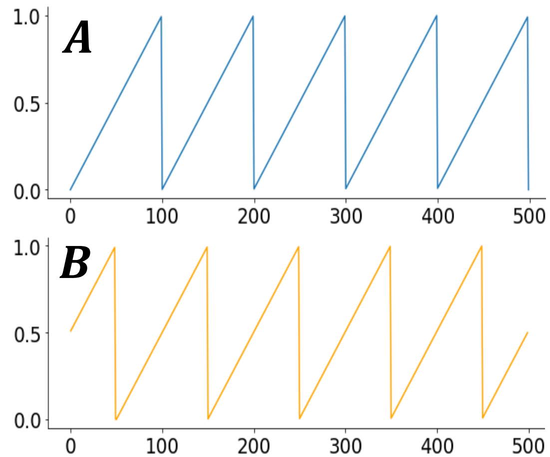


1. A time series is represented as a set of non-overlapped segments
2. For each segment, find its **nearest neighbors**
3. For each segment, compute its coverage and take top-K segments

How to measure similarity? MPdist*

- Two m**-length time series** are more similar w.r.t. **MPdist**, the more ℓ**-length** $(3 \leq \ell \leq m)$ **subsequences** close to each other w.r.t. **normalized Euclidean distance**, are in them
- MPdist is
  - a distance measure (not a metric), i.e., it holds the identity and symmetry axioms but not the triangle inequality
  - phase-invariant

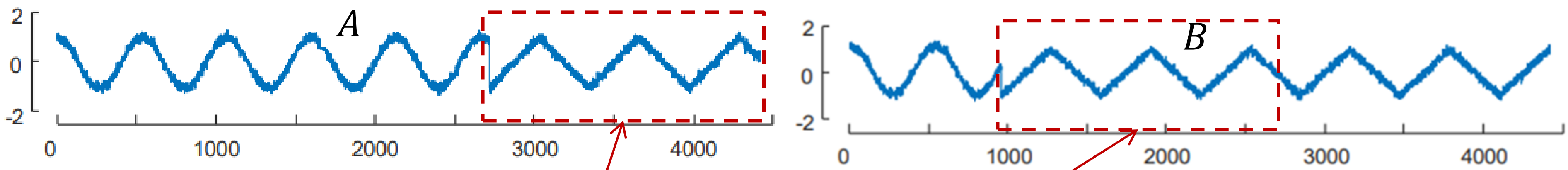
$\mathrm{ED}(A, B)$	11.2
MPdist(A, B)	**0**



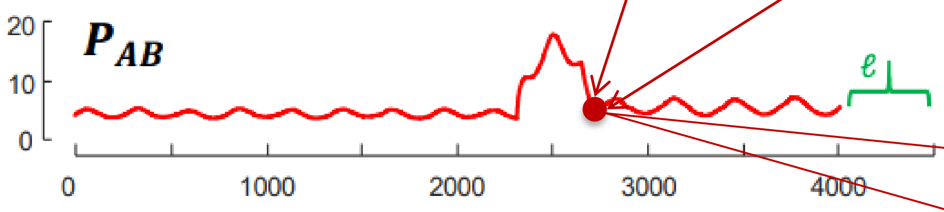
* Gharghabi S. *et al.* An ultra-fast time series distance measure to allow data mining in more complex real-world deployments. Data Min. Knowl. Discov. 2020. Vol. 34. P. 1104–1135. DOI: [10.1007/s10618-020-00695-8](https://doi.org/10.1007/s10618-020-00695-8)

MPdist: Matrix profile AB

$$|A| = |B| = m$$



Meaningful subsequence length: $3 \le \ell \le m$ (typically, $\lceil 0.3m \rceil < \ell \le \lceil 0.8m \rceil$)



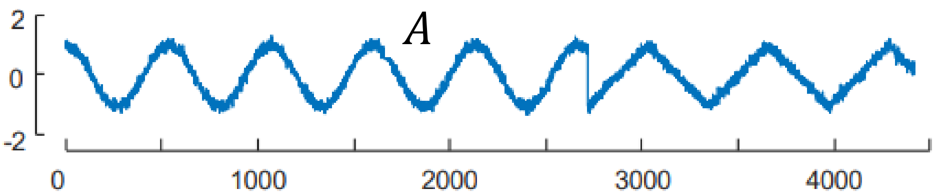
$$\{P_{AB}(i) = \mathrm{ED}_{\mathrm{norm}}(A_{i,\ell}, B_{j,\ell})\}_{i=1}^{m-\ell+1}$$

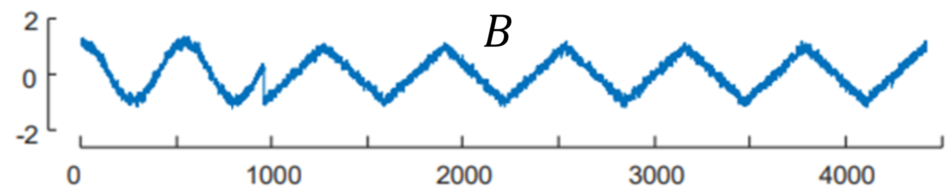
$$, \quad B_{j,\ell} = \arg \min_{1 \le q \le m-\ell+1} \mathrm{ED}_{\mathrm{norm}}(A_{i,\ell}, B_{q,\ell})$$

Normalized Euclidean distance between
i-th subsequence in A
and its nearest subsequence in B

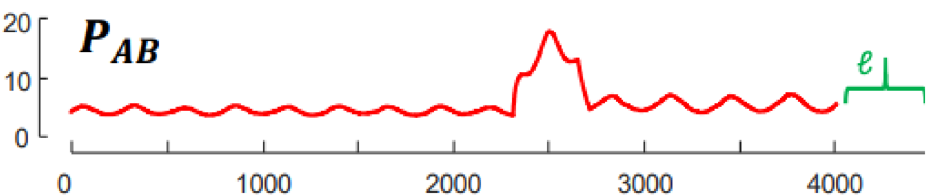
MPdist: Matrix profile BA

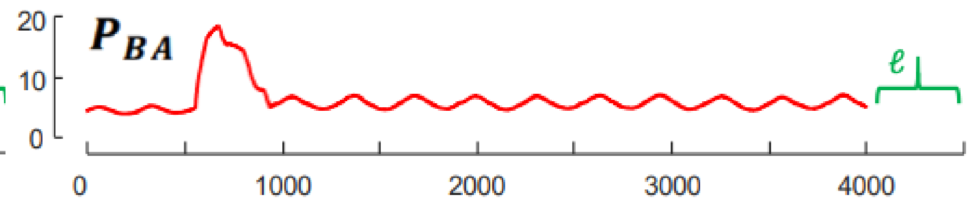
$$|A| = |B| = m$$





Meaningful subsequence length: $3 \le \ell \le m$ (typically, $\lceil 0.3m \rceil < \ell \le \lceil 0.8m \rceil$)





$$\left\{P_{AB}(i) = \mathrm{ED_{norm}}\left(A_{i,\ell}, B_{j,\ell}\right)\right\}_{i=1}^{m-\ell+1}$$

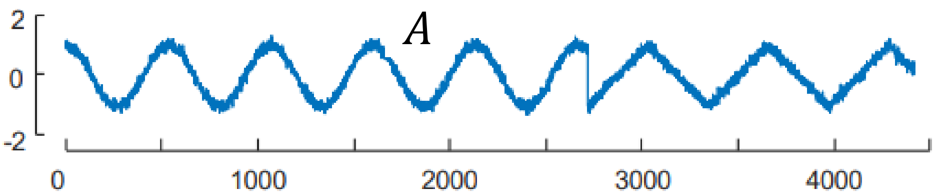
$$B_{j,\ell} = \arg \min_{1 \le q \le m-\ell+1} \mathrm{ED_{norm}}\left(A_{i,\ell}, B_{q,\ell}\right)$$

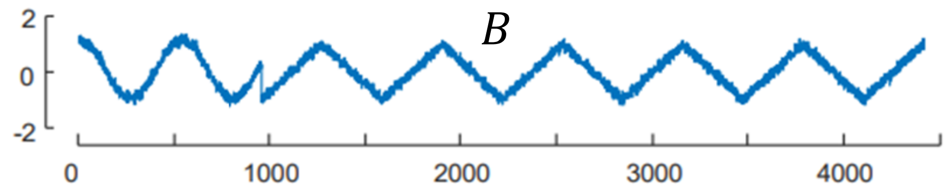
$$\left\{P_{BA}(i) = \mathrm{ED_{norm}}\left(B_{i,\ell}, A_{j,\ell}\right)\right\}_{i=1}^{m-\ell+1}$$

$$A_{j,\ell} = \arg \min_{1 \le q \le m-\ell+1} \mathrm{ED_{norm}}\left(B_{i,\ell}, A_{q,\ell}\right)$$

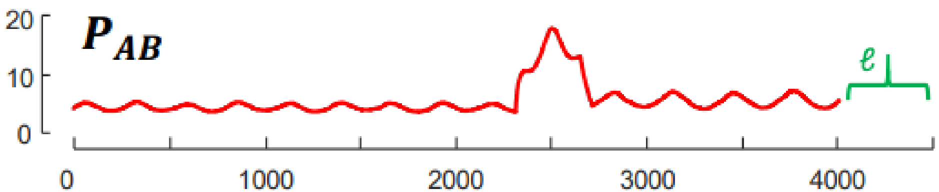
MPdist: Matrix profile ABBA

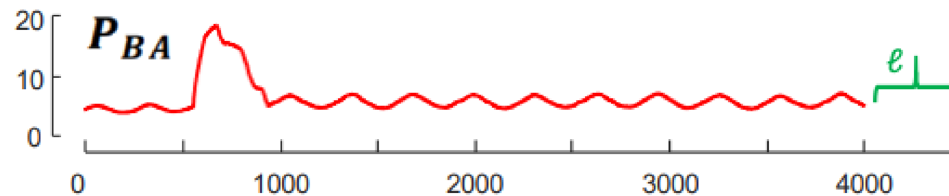
$$|A| = |B| = m$$



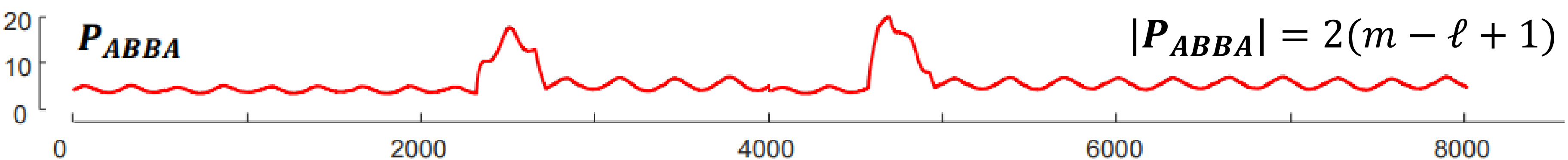


$$3 \leq \ell \leq m$$





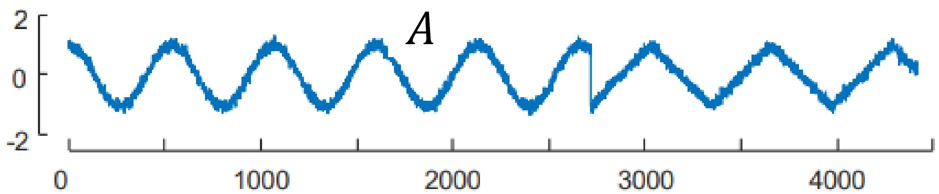
$$P_{ABBA} = P_{AB} \odot P_{BA}$$

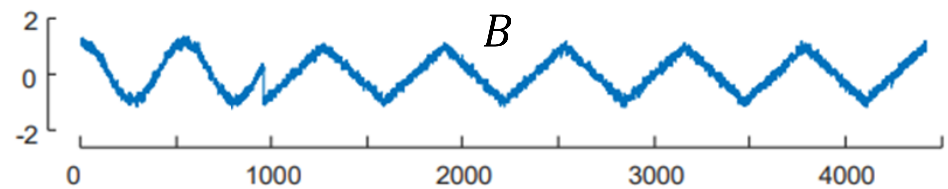


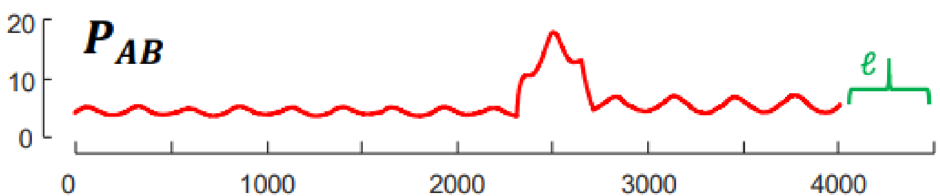
$$|P_{ABBA}| = 2(m - \ell + 1)$$

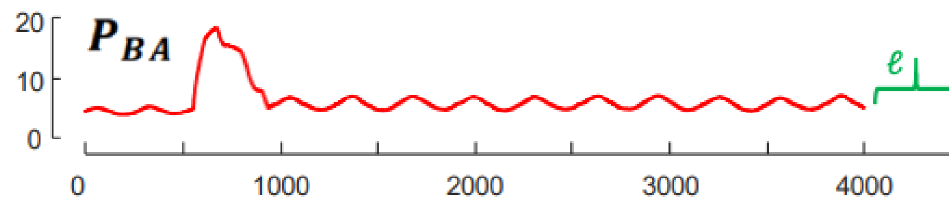
MPdist: Eventual calculation

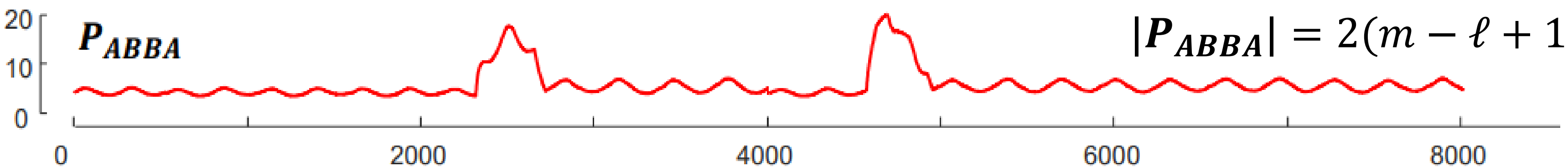
$|A| = |B| = m$ $\quad 3 < \ell \leq m$









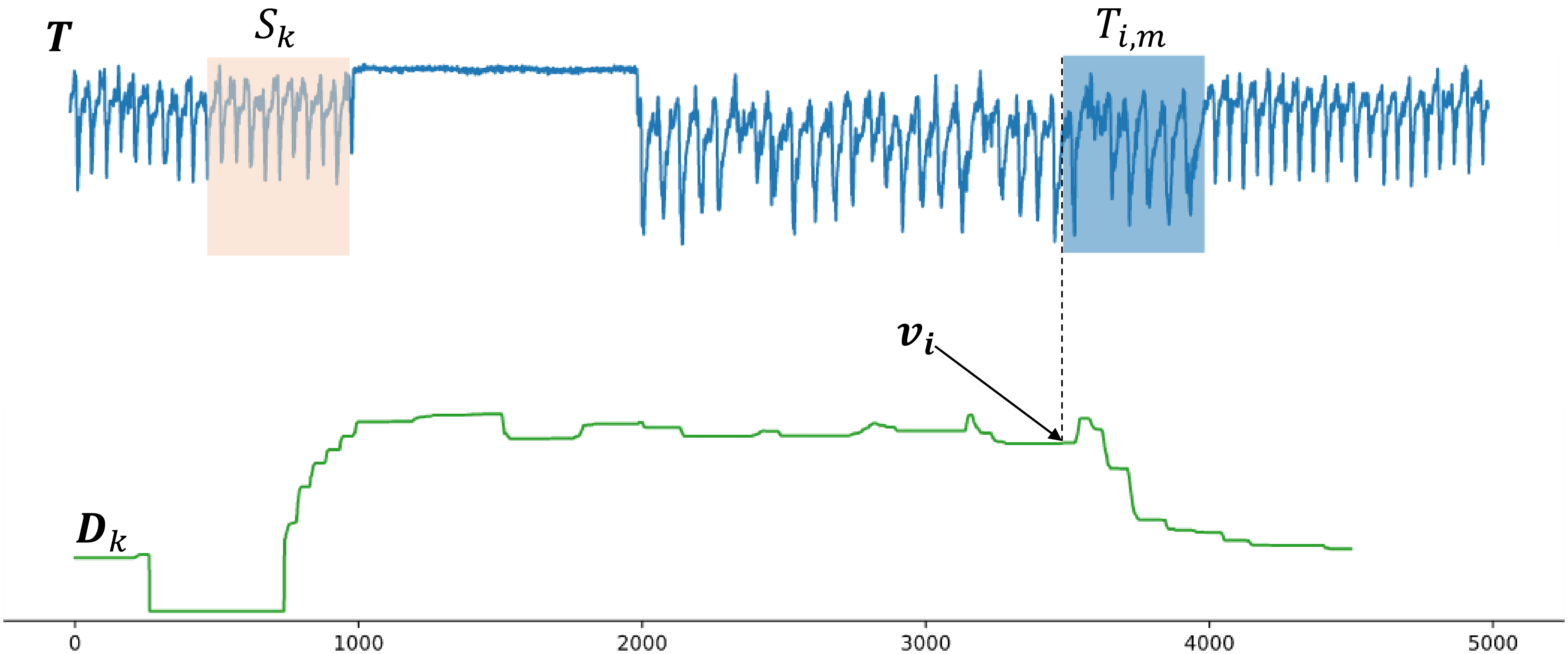


$|\boldsymbol{P}_{ABBA}| = 2(m - \ell + 1)$

$$\mathrm{MPdist}_{\ell}(A,B) = \begin{cases} SortedP_{ABBA}(k), & |P_{ABBA}| > k \\ SortedP_{ABBA}\big(2(m-\ell+1)\big), & |P_{ABBA}| \leq k' \end{cases}$$

where $k = \lceil 0.05 \cdot 2m \rceil = \lceil 0.1m \rceil$.

MPdist profile of a segment



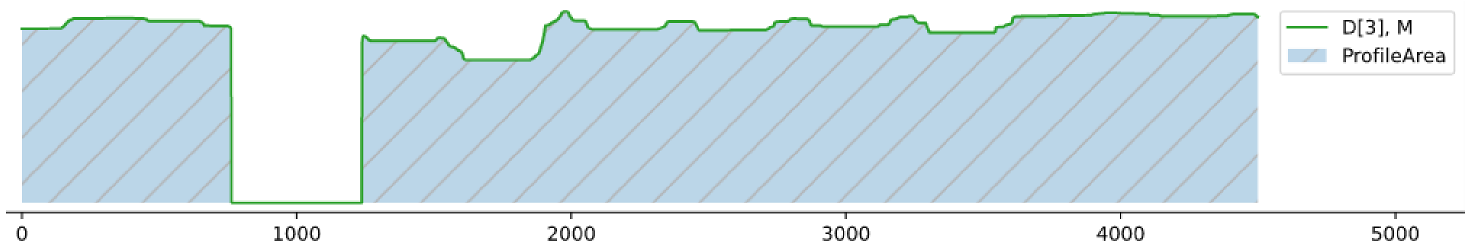
$$D_k(S_k, T) = [v_1, \dots, v_{n-m+1}],$$
$$v_i = \mathrm{MPdist}_\ell(S_k, T_{i,m})$$

Snippet discovery: top-1 snippet

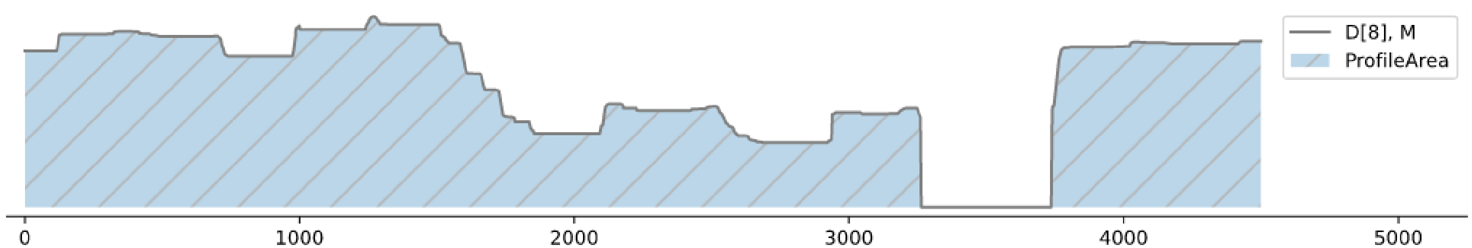
Discovery C_1

i	$ProfileArea$
1	60813
2	60371
3	74451
4	75141
5	56766
6	57729
7	58713
8	53769
9	62127
10	61286

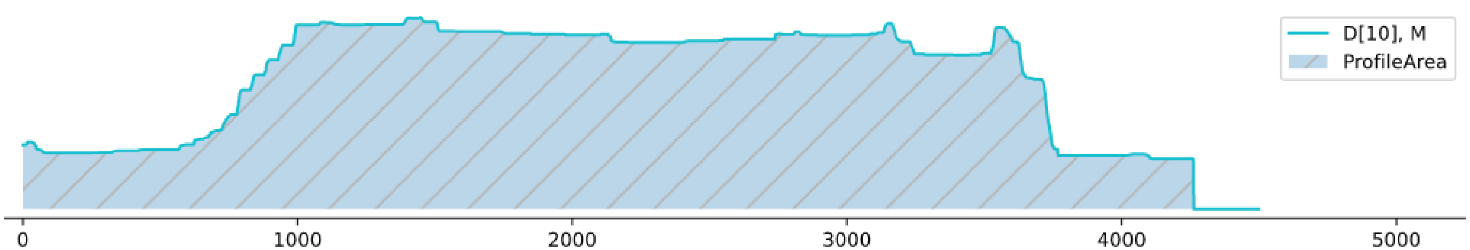
$C_1.index = 8$



$ProfileArea(\{D_3\}) = 74451$

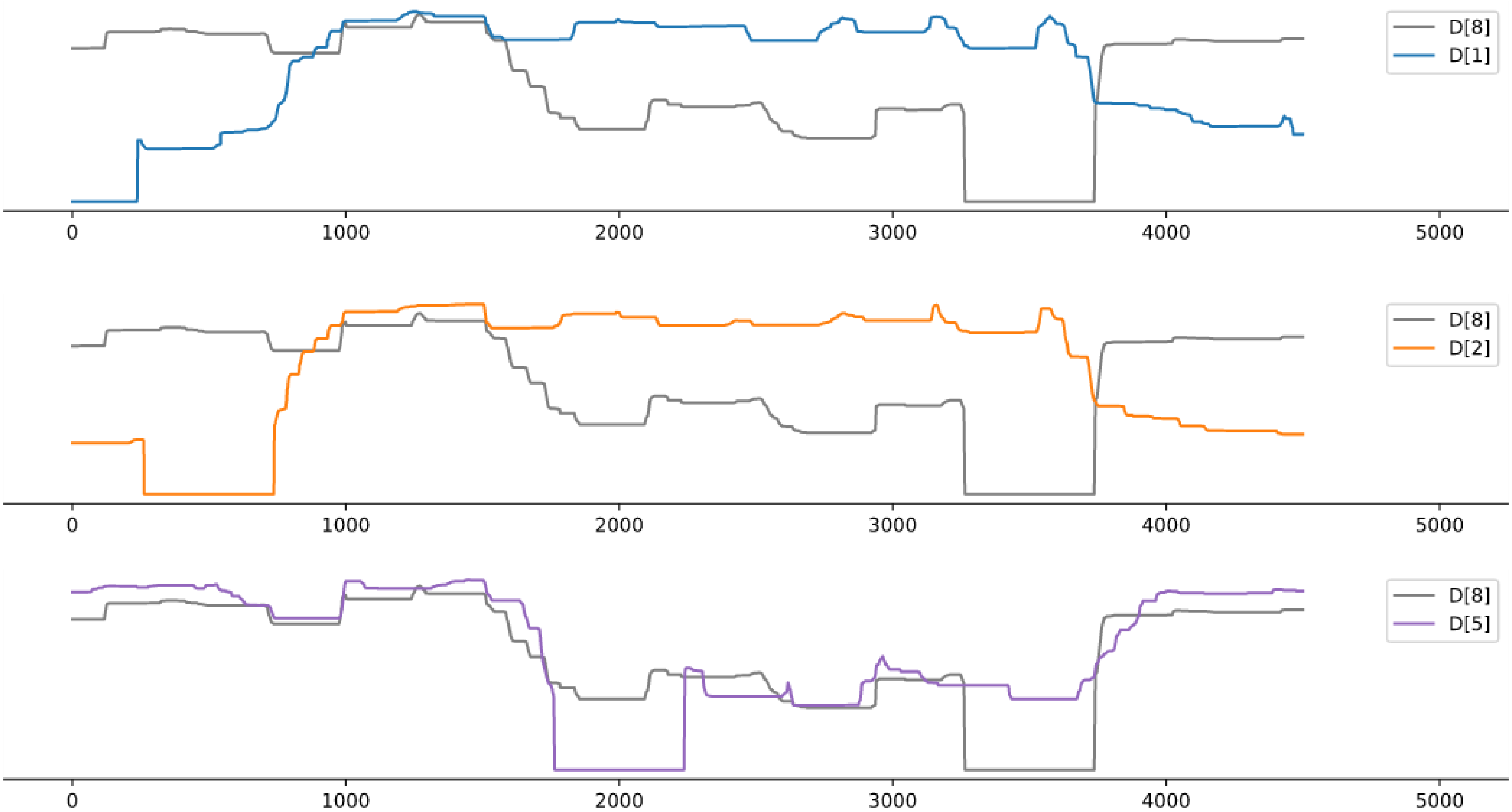


$ProfileArea(\{D_8\}) = 53769$



$ProfileArea(\{D_{10}\}) = 61286$

Snippet discovery: top-2 snippet

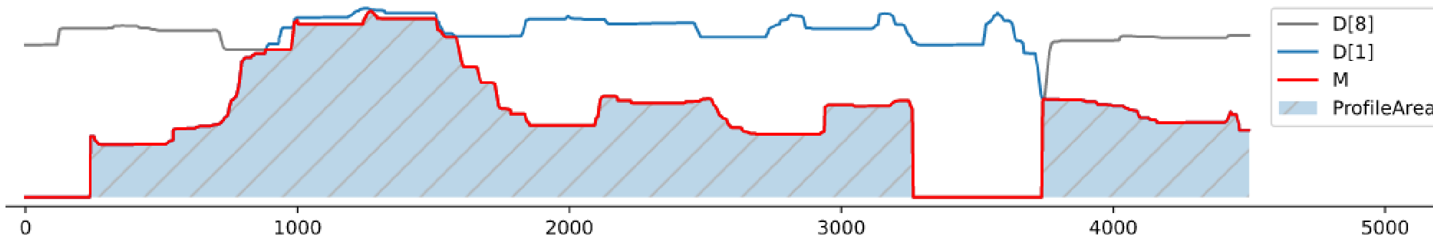


Snippet discovery: top-2 snippet

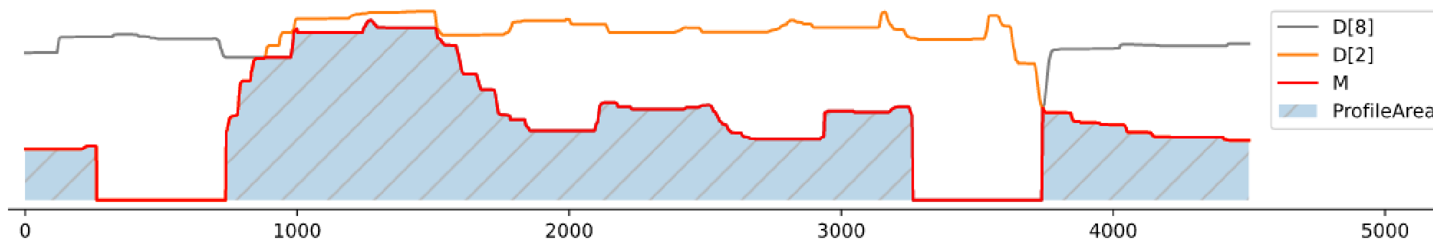
C_2

i	$ProfileArea$
1	38394
2	35769
3	45629
4	45908
5	48857
6	49264
7	48975
9	36684
10	36482

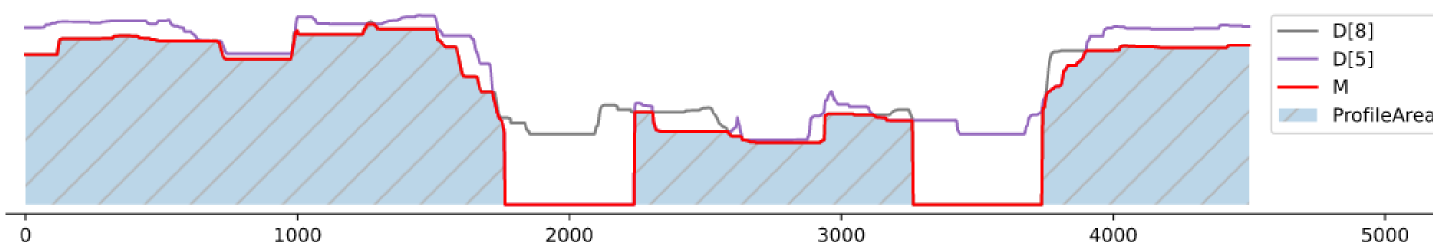
$C_2.index = 2$



$$ProfileArea(\{D_8, D_1\}) = 38394$$

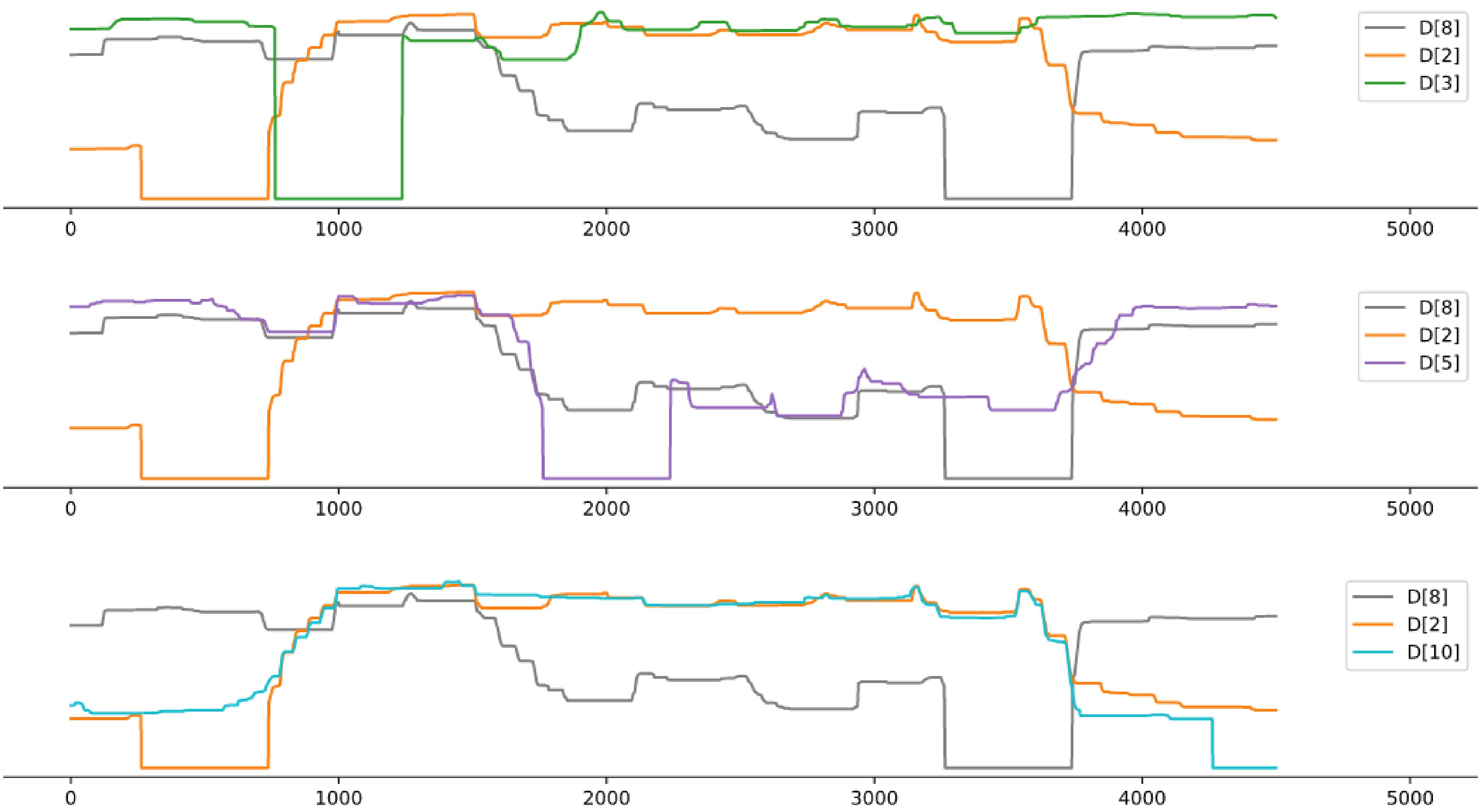


$$ProfileArea(\{D_8, D_2\}) = 35769$$



$$ProfileArea(\{D_8, D_5\}) = 48867$$

Snippet discovery: top-3 snippet

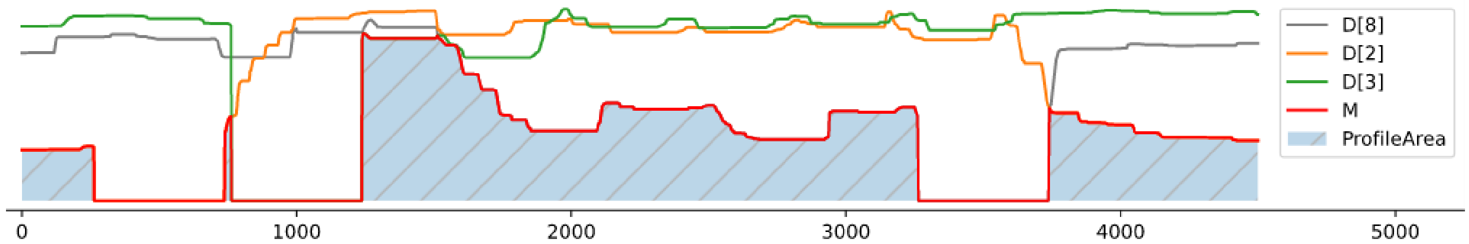


Snippet discovery: top-3 snippet

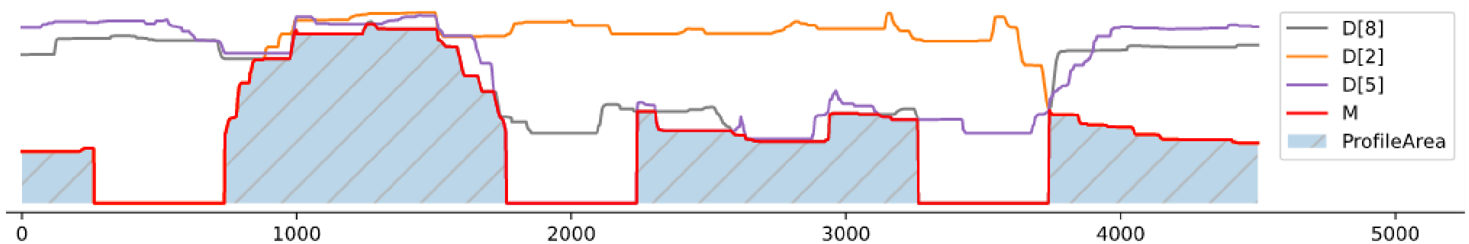
C_3

i	$ProfileArea$
1	34475
3	27899
4	27908
5	31168
6	31532
7	31672
9	31654
10	33044

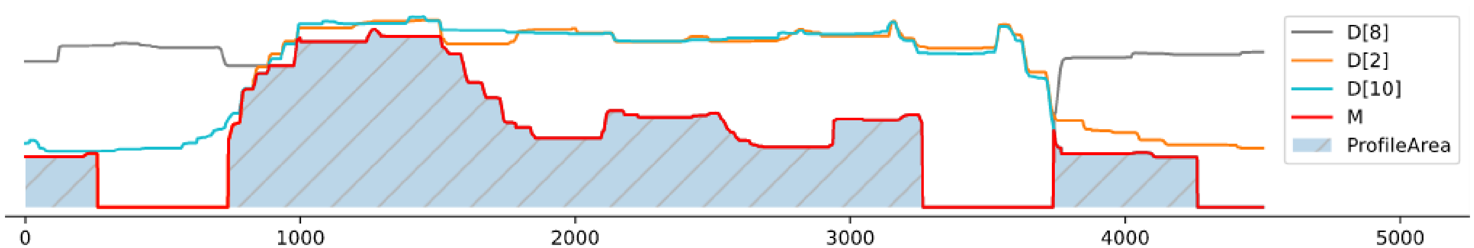
$C_3.index = 3$



$ProfileArea(\{D_8, D_2, D_3\}) = 27899$

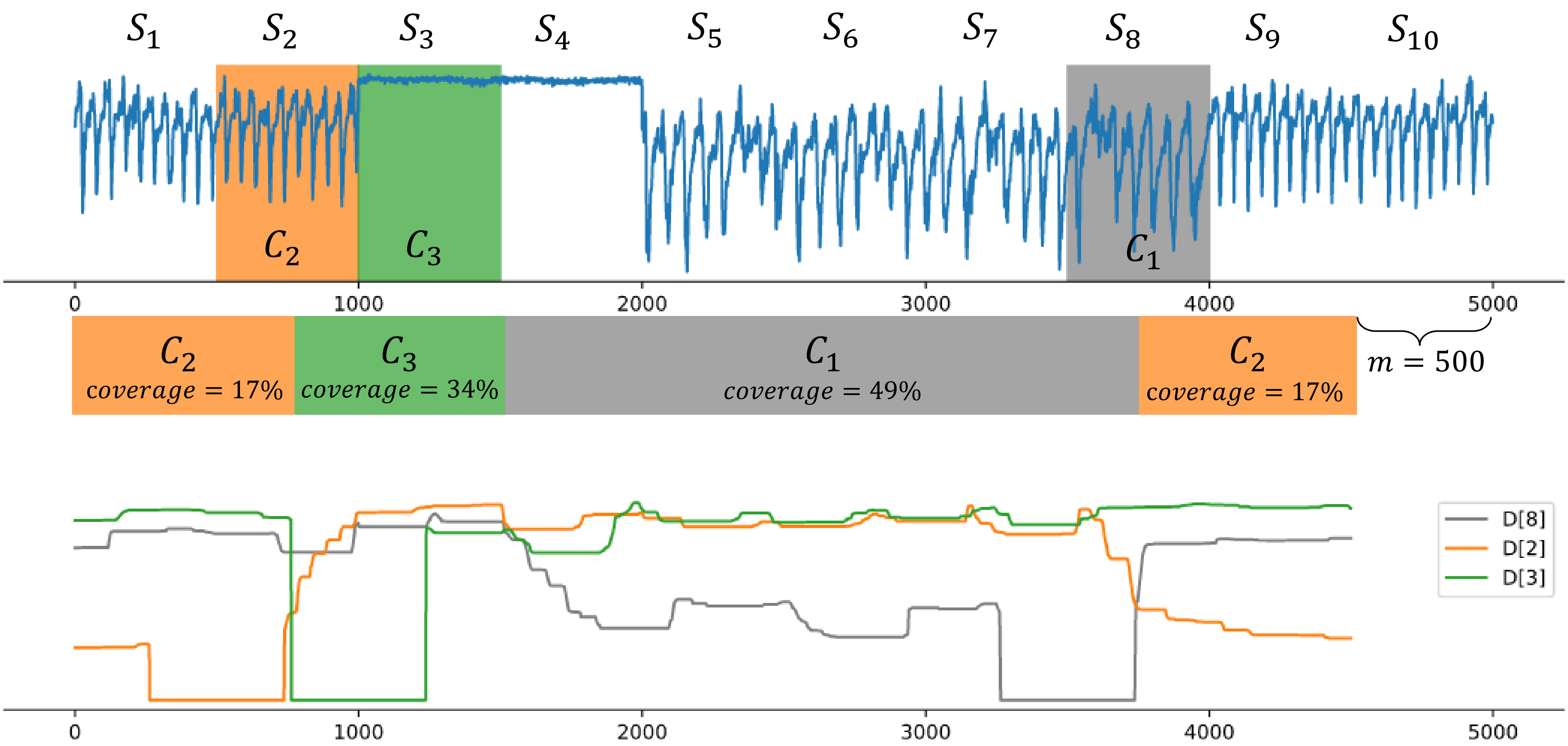


$ProfileArea(\{D_8, D_2, D_5\}) = 31168$



$ProfileArea(\{D_8, D_2, D_{10}\}) = 33044$

Snippet discovery: Resulting snippets

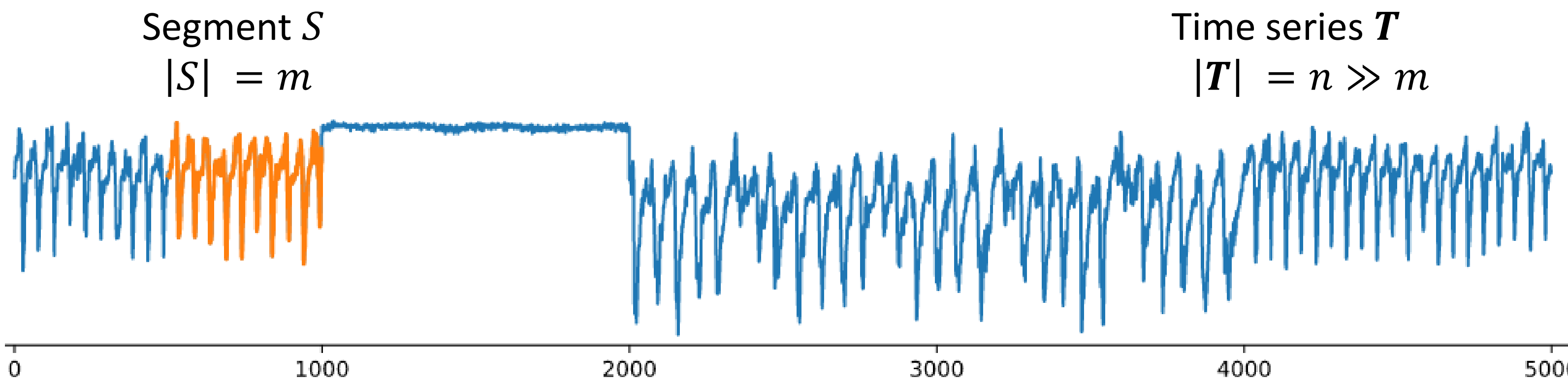


PSF: Parallel Snippet-Finder for GPU*

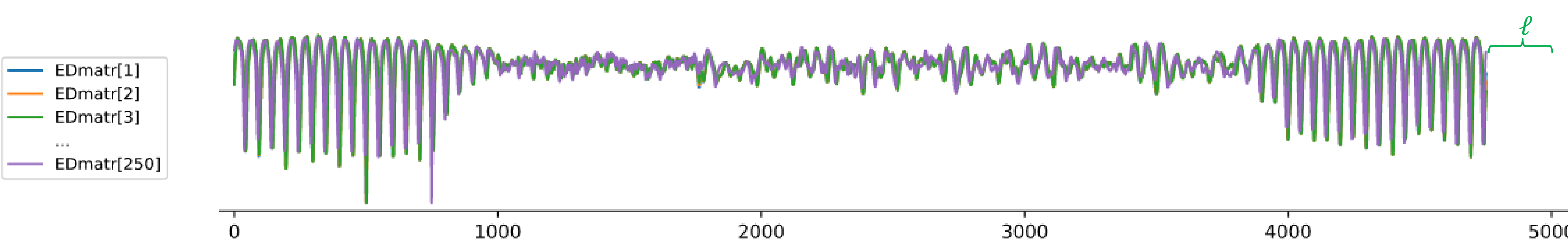
Step	**Snippet-Finder**, complexity $O(n^2 \cdot \frac{n-m}{m})$	**PSF (Parallel Snippet-Finder)**
1. Calculation of matrix profile P_{AB}	$\{P_{AB}(i) = \mathrm{ED_{norm}}(A_{i,\ell}, B_{j,\ell})\}_{i=1}^{m-\ell+1}$, $B_{j,\ell} = \arg\min_{1 \le q \le m-\ell+1} \mathrm{ED_{norm}}(A_{i,\ell}, B_{q,\ell})$	Calculate the ED_{matr} matrix of normalized Euclidean distances $allP_{AB}(i,j) = \min_{j \le c \le j+m-\ell+1} ED_{matr}(i,c)$
2. Calculation of matrix profile P_{BA}	$\{P_{BA}(i) = \mathrm{ED_{norm}}(B_{i,\ell}, A_{j,\ell})\}_{i=1}^{m-\ell+1}$, $A_{j,\ell} = \arg\min_{1 \le q \le m-\ell+1} \mathrm{ED_{norm}}(B_{i,\ell}, A_{q,\ell})$	$allP_{BA}(j) = \min_{1 \le i \le m-\ell+1} ED_{matr}(i,j)$
3. Calculation of matrix profile P_{ABBA}	$P_{ABBA} = P_{AB} \odot P_{BA}$	$P_{ABBA} = allP_{AB}(i, m-\ell) \odot allP_{BA}(i)$
4. Calculation of MPdist profile	$\mathrm{MPdist}_{\ell}(A,B) = \begin{cases} SortedP_{ABBA}(k), & \lvert P_{ABBA}\rvert > k \\ SortedP_{ABBA}(2(m-\ell+1)), & \lvert P_{ABBA}\rvert \le k \end{cases}$	

*Zymbler M., Goglachev A. Fast Summarization of Long Time Series with Graphics Processor. Mathematics. 2022. Vol. 10, No. 10. Article 1781. DOI: 10.3390/math10101781

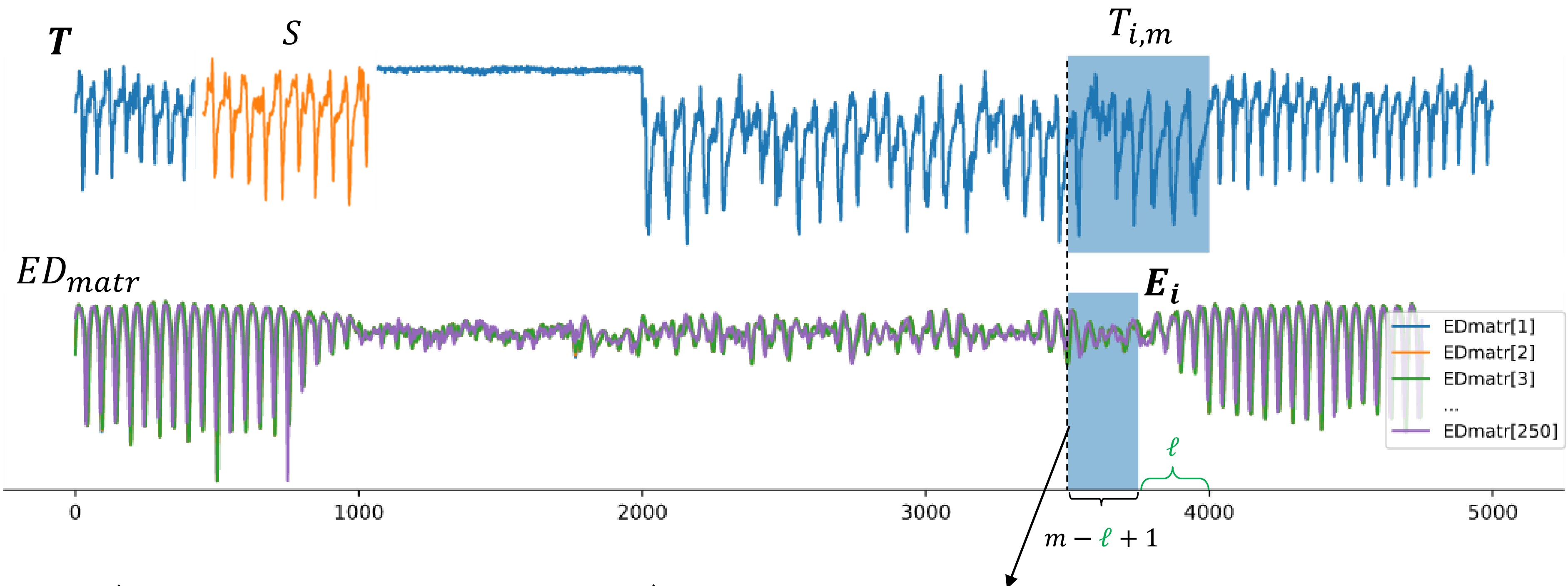
Snippet discovery: MPdist profile



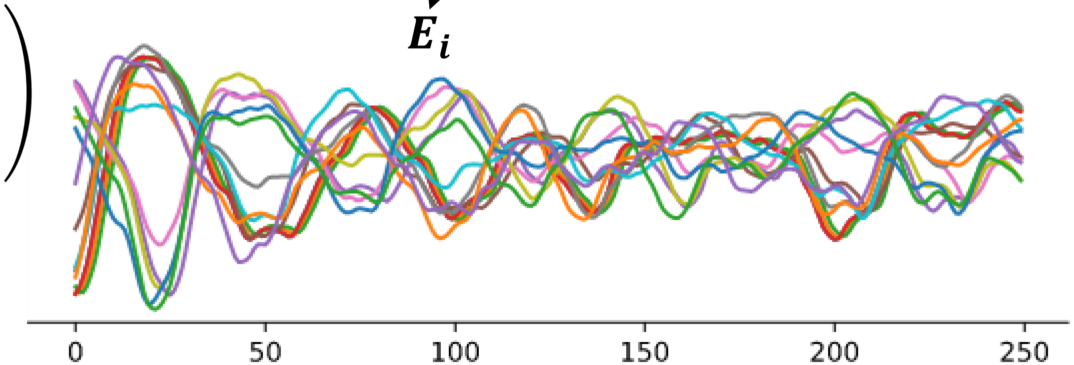
$$\boldsymbol{ED_{matr}}(S, \boldsymbol{T}, \ell) = \begin{pmatrix} d_{1,1} & \cdots & d_{1,n-\ell+1} \\ \cdots & \ddots & \cdots \\ d_{m-\ell+1,1} & \cdots & d_{m-\ell+1,n-\ell+1} \end{pmatrix}, \qquad d_{i,j} = \mathrm{ED_{norm}}(S_{i,\ell}, T_{j,\ell})$$



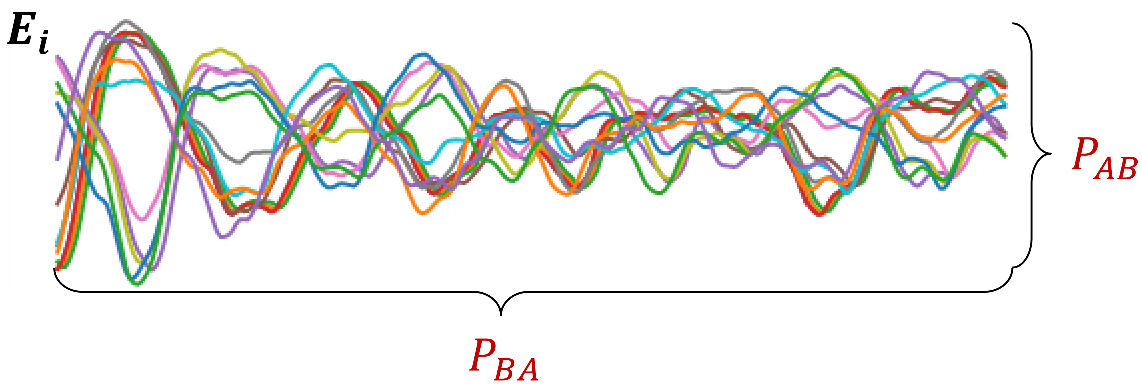
Snippet discovery: MPdist profile



$$E_i = \begin{pmatrix} d_{1,i} & \dots & d_{1,i+m-\ell+1} \\ \dots & \ddots & \dots \\ d_{m-\ell+1,i} & \dots & d_{m-\ell+1,i+m-\ell+1} \end{pmatrix}$$

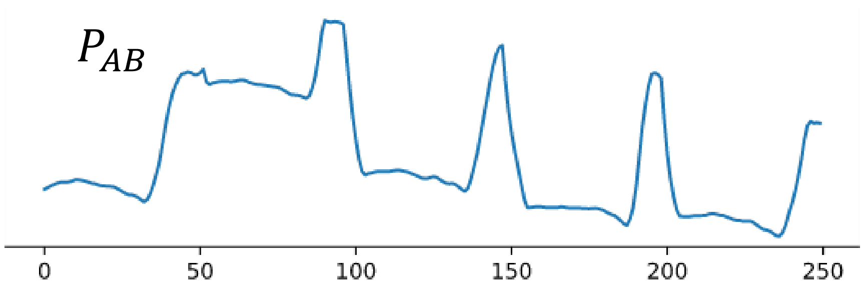


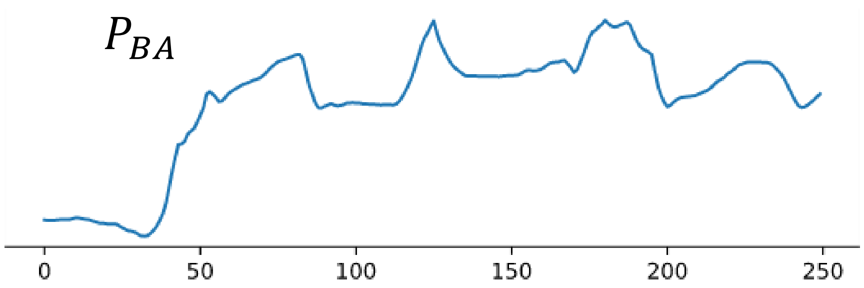
Snippet discovery: MPdist profile



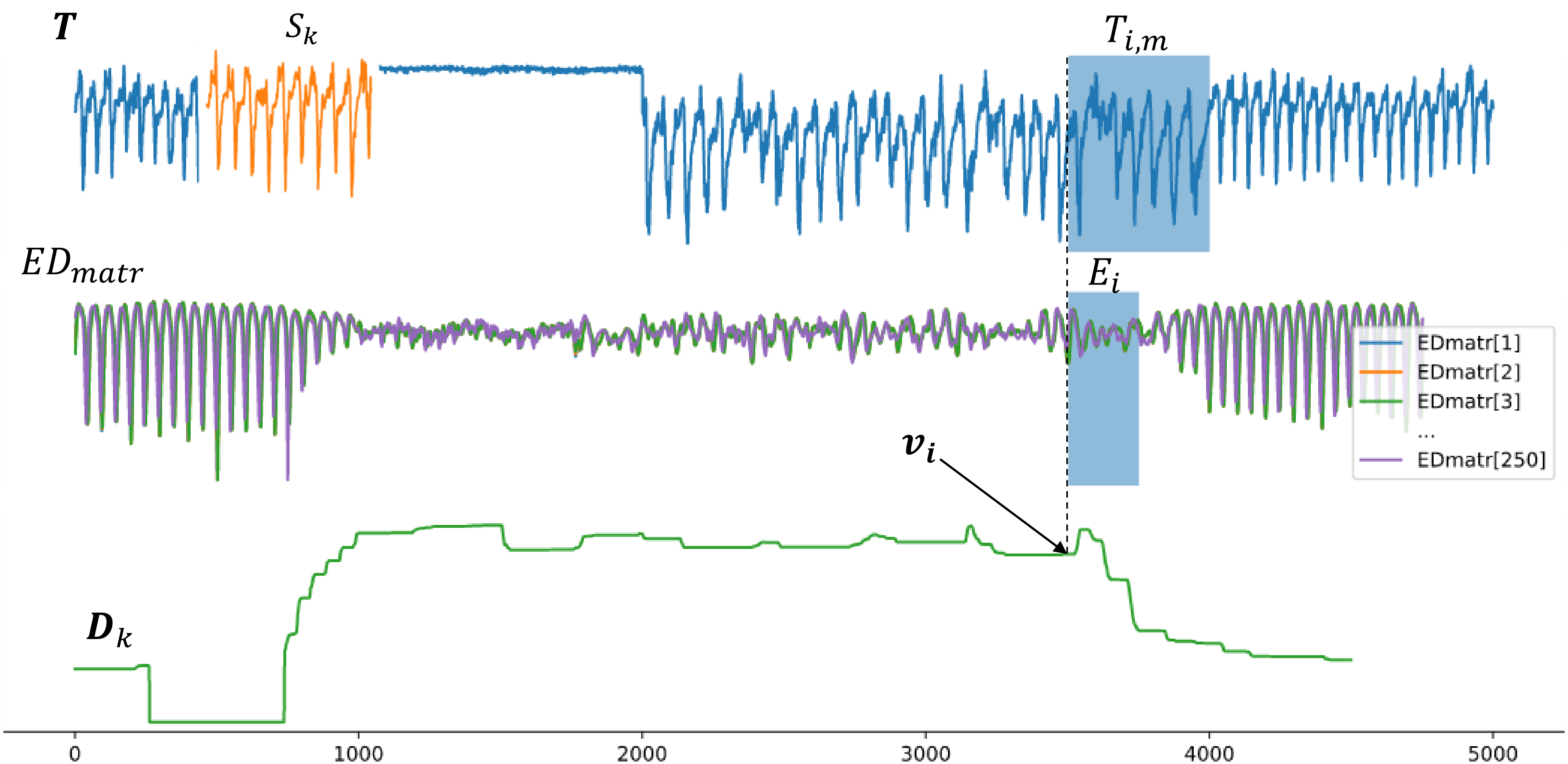
$$P_{AB}(i) = \min_{1 \le j \le m-\ell+1} E(i,j), \qquad 1 \le i \le m - \ell + 1$$

$$P_{BA}(j) = \min_{1 \le i \le m-\ell+1} E(i,j), \qquad 1 \le j \le m - \ell + 1$$



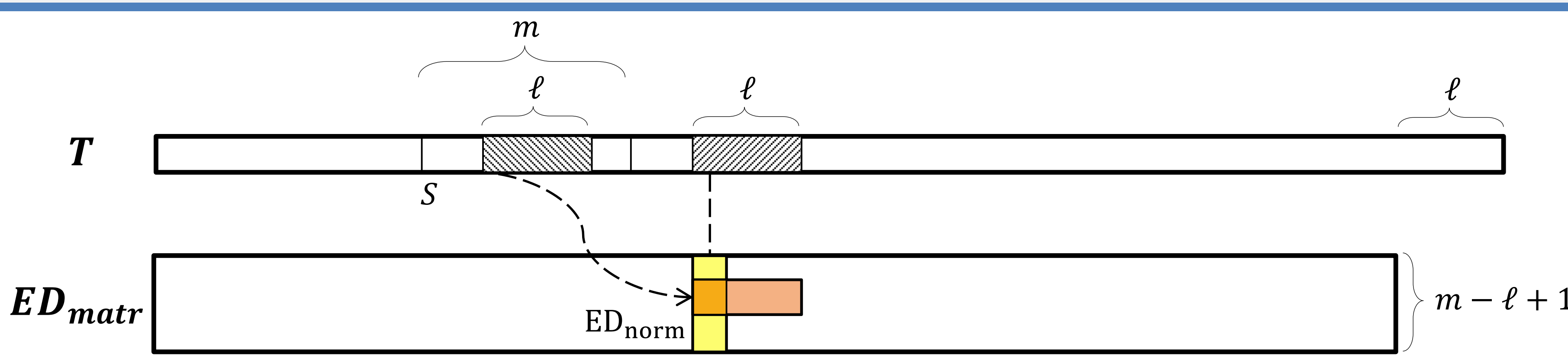


Snippet discovery: MPdist profile



$$D_k(S_k, T) = [v_1, v_2, \dots, v_{n-m+1}], \qquad v_i = \mathrm{MPdist}_\ell(S_k, T_{i,m})$$

Parallel snippet discovery: ED_{matr}



$$\text{ED}_{\text{norm}}(T_{i,m}, T_{j,m}) = \sqrt{2m(1 - P_{i,j})}$$

$$P_{i,j} = \overline{QT}_{i,j} \cdot \frac{1}{\|T_{i,m} - \mu_i\|} \cdot \frac{1}{\|T_{j,m} - \mu_j\|},$$

$$T_{i,m} - \mu_i = (t_i - \mu_i, \ldots, t_{i+m-1} - \mu_i),$$

$$\mu_i = \frac{1}{m} \sum_{j=i}^{i+m} t_j,$$

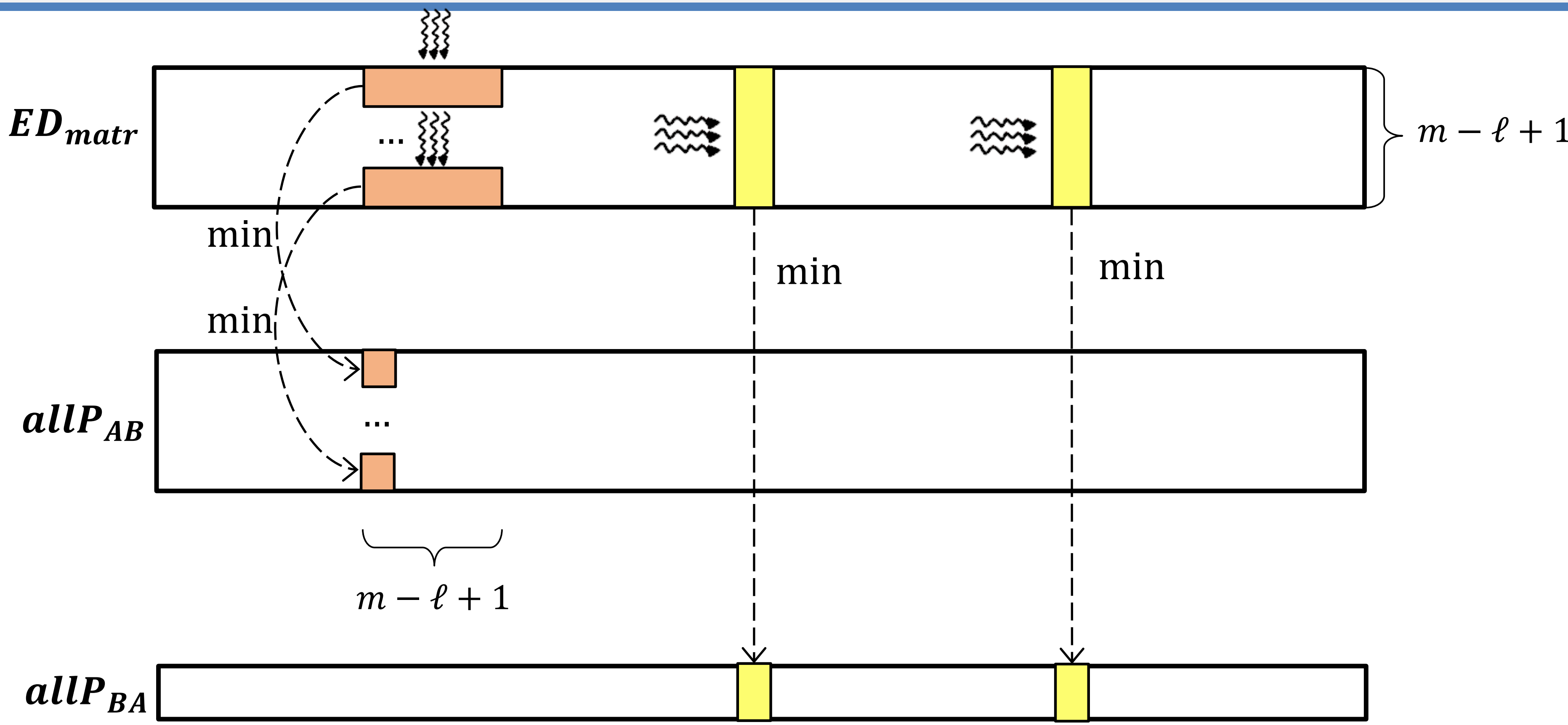
$$dg_0 = 0; dg_i = (t_{i+m-1} - \mu_i) + (t_{i-1} - \mu_{i-1}),$$

$$df_0 = 0; df_i = \frac{t_{i+m-1} - t_{i-1}}{2},$$

$$\overline{QT}_{i,j} = \overline{QT}_{i-1,j-1} + df_i \cdot dg_j + df_j \cdot dg_i, \quad *$$

* 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. P. 74–86. DOI: [10.1145/3357223.3362721](https://doi.org/10.1145/3357223.3362721)

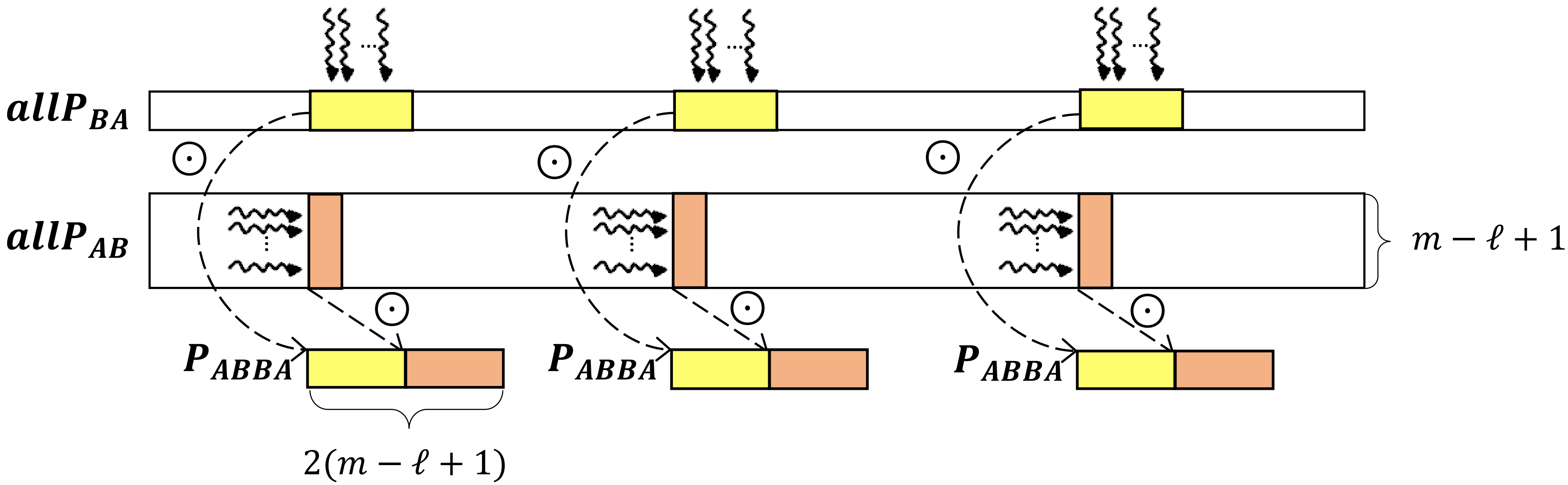
Parallel snippet discovery: $allP_{AB}$ and $allP_{BA}$



$$allP_{AB}(i,j) = \min_{j \le c \le j+m-\ell+1} ED_{matr}(i,c)$$

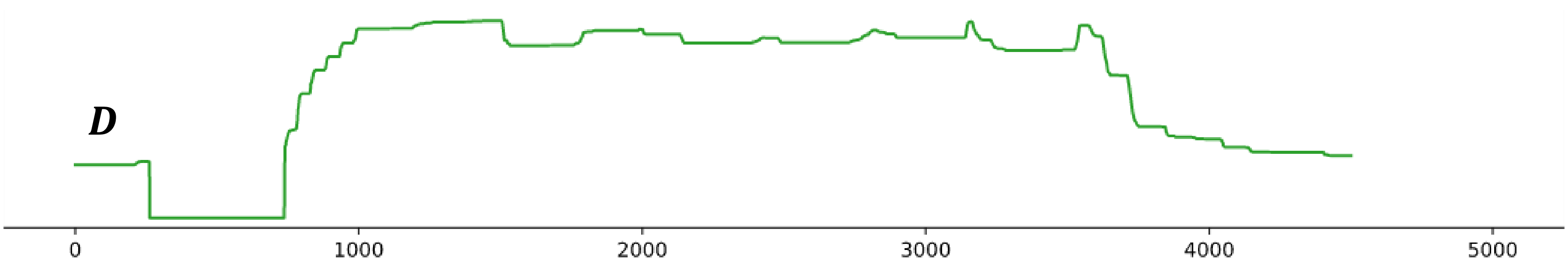
$$allP_{BA}(j) = \min_{1 \le i \le m-\ell+1} ED_{matr}(i,j)$$

Parallel snippet discovery: P_{ABBA}



$$\text{MPdist}_\ell(A, B) = \begin{cases} SortedP_{ABBA}(k), & |P_{ABBA}| > k \\ SortedP_{ABBA}(2(m - \ell + 1)), & |P_{ABBA}| \le k\text{'} \end{cases}$$

where $k = \lceil 0.05 \cdot 2m \rceil = \lceil 0.1m \rceil$.



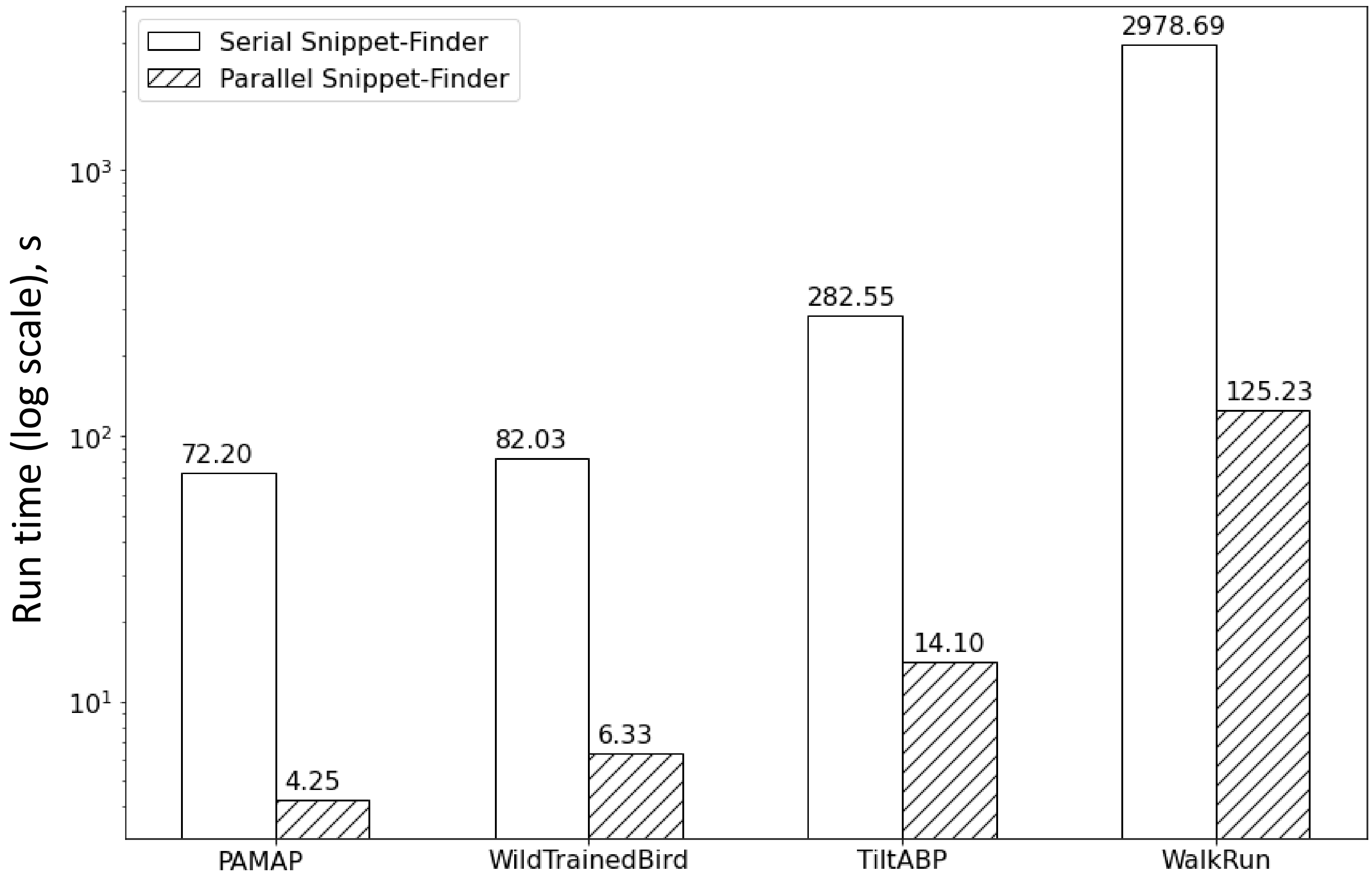
Parallel snippet discovery: Experiments

- **Hardware**
  - NVIDIA Tesla V100 SXM2 (5120 cores @1.3 GHz)
- **Data**

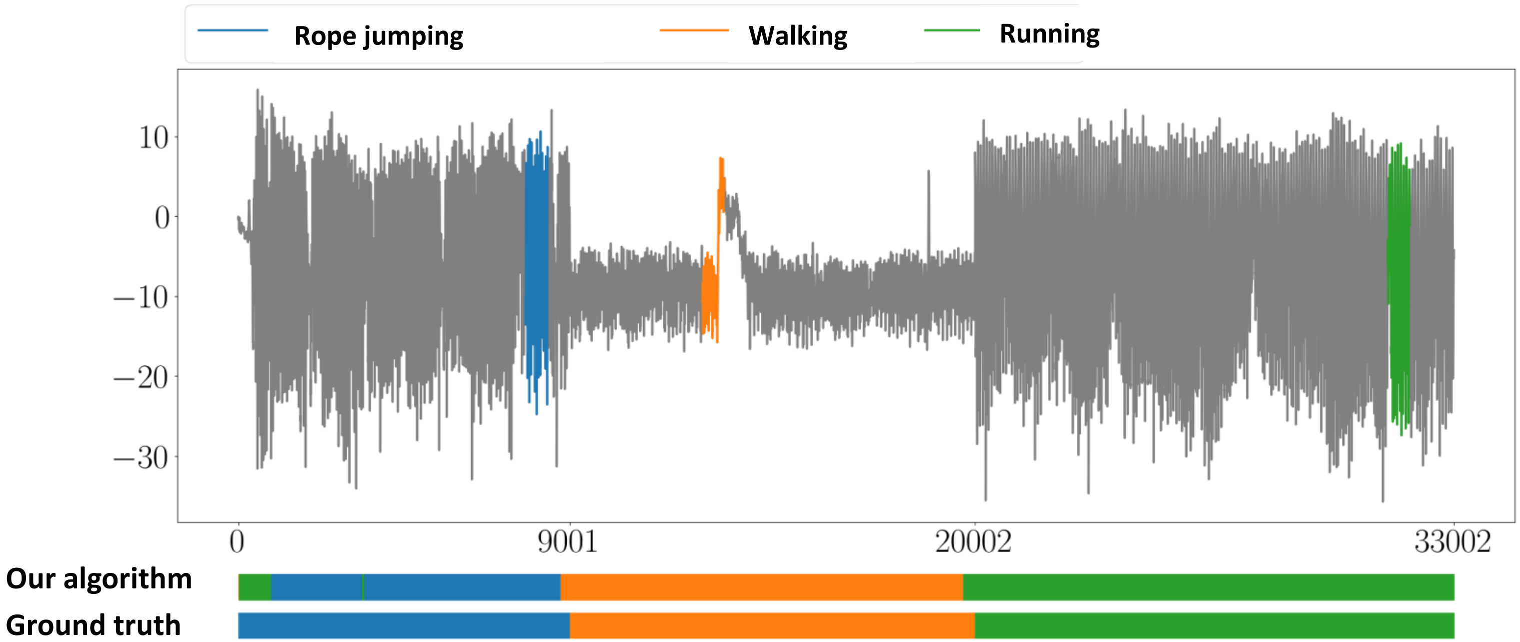
Time series	Length n	Snippet length m	Domain
WildVTrainedBird[1]	20 002	900	Physiological indicators of bird vital activity
PAMAP[2]	20 002	600	Wearable accelerometer readings during various types of human physical activity
WalkRun[2]	100 000	240	
TiltABP[1]	40 000	630	Human blood pressure during rapid tilts

[1] Imani S., et al. Introducing time series snippets: a new primitive for summarizing long time series. Data Min. Knowl. Discov. 2020. Vol. 34, no. 6. P. 1713-1743. DOI: 10.1007/s10618-020-00702-y
[2] Reiss A., Stricker D. Introducing a new benchmarked dataset for activity monitoring. ISWC 2012. P. 108–109. DOI: 10.1109/ISWC.2012.13

Parallel snippet discovery: Performance



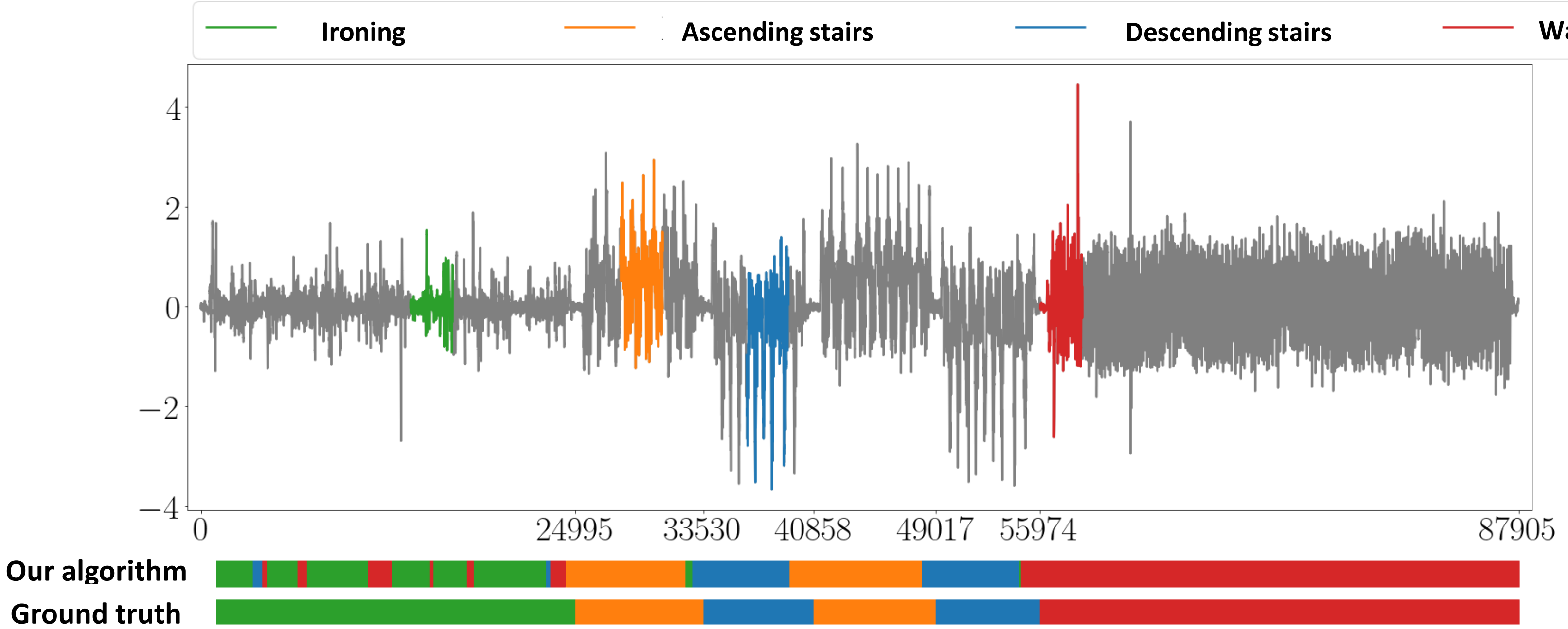
Parallel snippet discovery: Case* studies

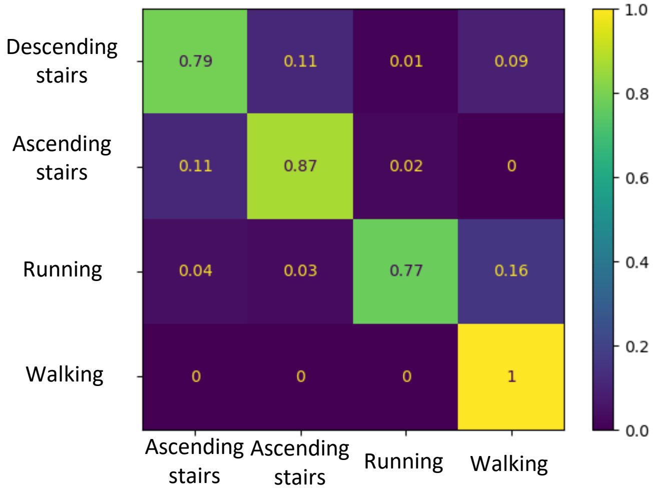


Activity	Precision	Recall	F1-score
Rope jumping	1	0.87	0.93
Walking	0.98	0.97	0.97
Running	0.77	1	0.87

* Reiss A., Stricker D. Introducing a new benchmarked dataset for activity monitoring. ISWC 2012. P. 108–109. DOI: 10.1109/ISWC.2012.13

Parallel snippet discovery: Case* studies





Activity	Precision	Recall	F1-score
Descending stairs	0.80	0.79	0.80
Ascending stairs	0.87	0.87	0.87
Ironing	0.97	0.77	0.86
Walking	0.86	1	0.92

* Reiss A., Stricker D. Introducing a new benchmarked dataset for activity monitoring. ISWC 2012. P. 108–109. DOI: 10.1109/ISWC.2012.13

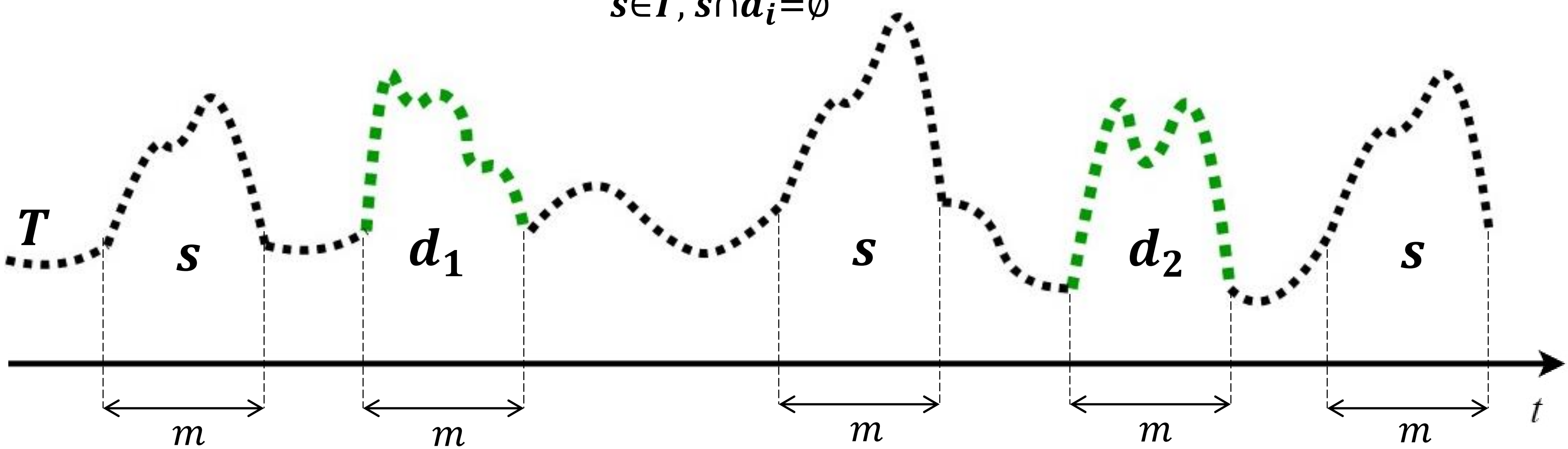
Outline

- Introduction
- Parallel pattern discovery
- **Parallel anomaly detection**
- Parallel imputation of missing values
- Online time series analytics with parallel algorithms

How to formalize anomalies? Discords*

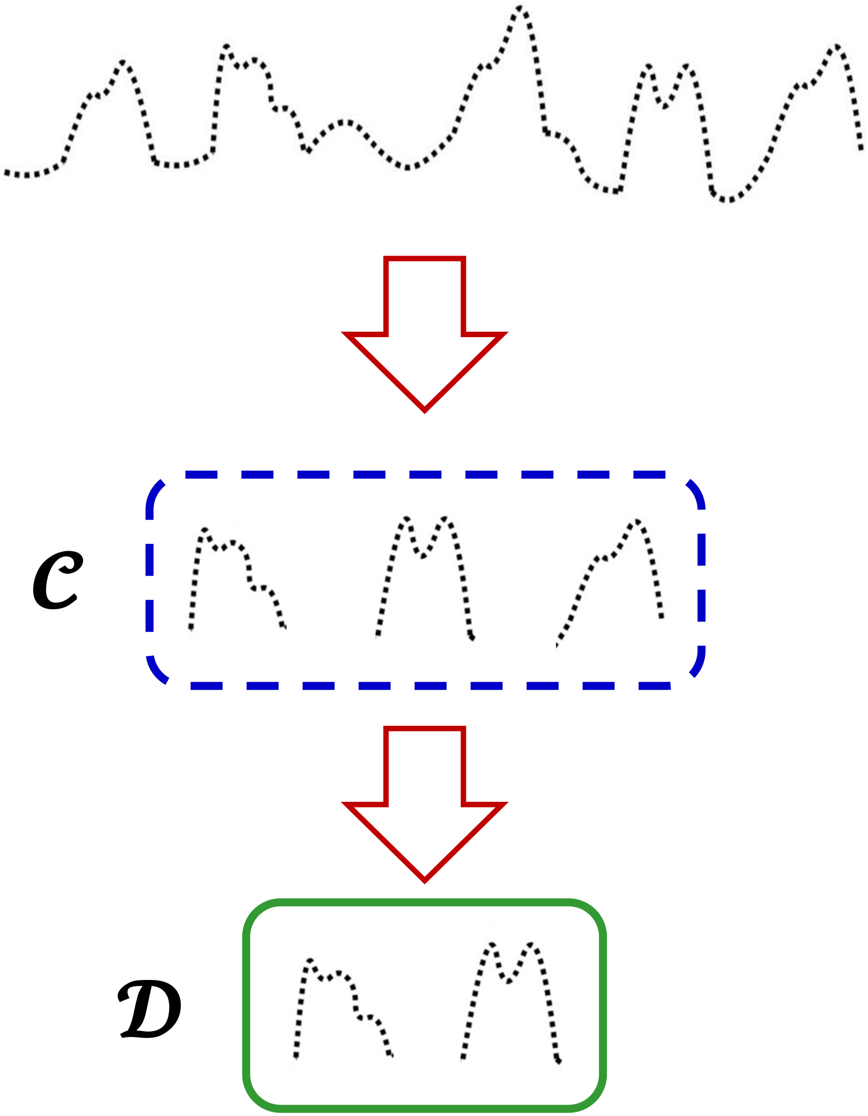
- Discord is a subsequence whose nearest neighbor is at least at a given threshold far away
- We are given: $\boldsymbol{T}$, discord length m, threshold r
- We are to find: $\boldsymbol{\mathcal{D}} = \{\boldsymbol{d}_1, \boldsymbol{d}_2, \ldots\}$

$$\boldsymbol{d}_i \in \boldsymbol{\mathcal{D}} \Leftrightarrow \min_{\boldsymbol{s} \in \boldsymbol{T},\, \boldsymbol{s} \cap \boldsymbol{d}_i = \emptyset} \mathrm{ED}(\boldsymbol{d}_i, \boldsymbol{s}) \geq r$$



* Yankov D., et al. Disk aware discord discovery: finding unusual time series in terabyte sized datasets. Knowl. Inf. Syst. 17(2): 241–262. 2008.

Discord discovery



1. Selection

Through one full scan of the time series, create a **set of candidates** to discords

2. Refinement

Through one full scan of the time series, **prune false positives** from the set above

Discord discovery: Selection

while not end of $\boldsymbol{T}$

    get current subsequence $\boldsymbol{s}$

    $isCandidate := \text{TRUE}$

    for each $\boldsymbol{c_i} \in \boldsymbol{\mathcal{C}}$ **and** $\boldsymbol{s} \cap \boldsymbol{c_i} = \emptyset$

        if $\text{ED}(\boldsymbol{s}, \boldsymbol{c_i}) < r$ **then**

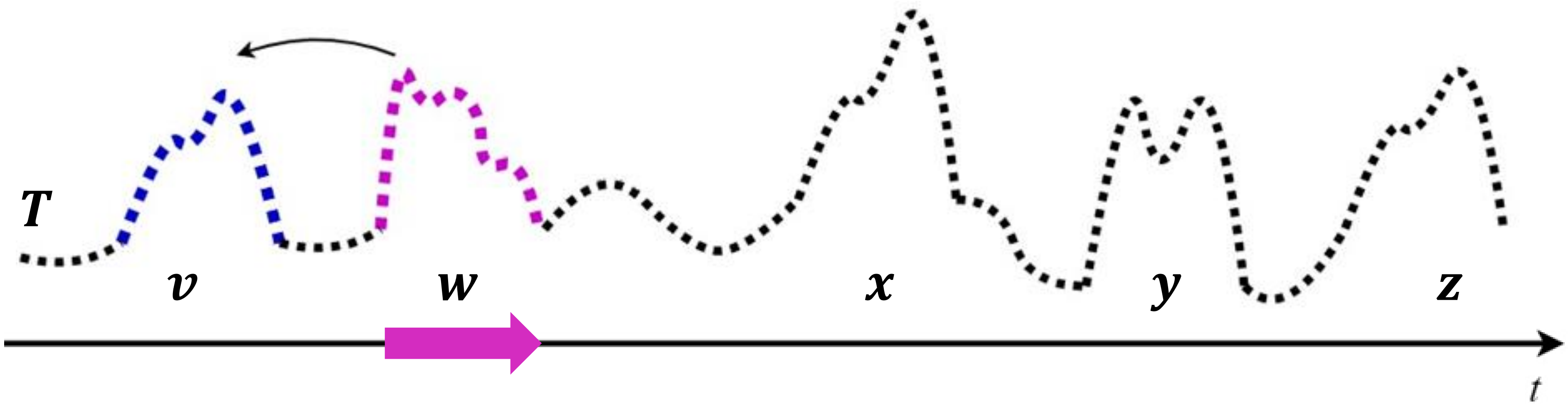
            $\boldsymbol{\mathcal{C}} := \boldsymbol{\mathcal{C}} \setminus \boldsymbol{c_i}$; $isCandidate := \text{FALSE}$

    if $isCandidate = \text{TRUE}$ **then** $\boldsymbol{\mathcal{C}} := \boldsymbol{\mathcal{C}} \cup \boldsymbol{s}$

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

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

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



Discord discovery: Selection

while not end of $\boldsymbol{T}$

  get current subsequence $\boldsymbol{s}$

  $isCandidate$:= TRUE

  for each $\boldsymbol{c_i} \in \boldsymbol{\mathcal{C}}$ **and** $\boldsymbol{s} \cap \boldsymbol{c_i} = \emptyset$

    if $\mathrm{ED}(\boldsymbol{s}, \boldsymbol{c_i}) < r$ **then**

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

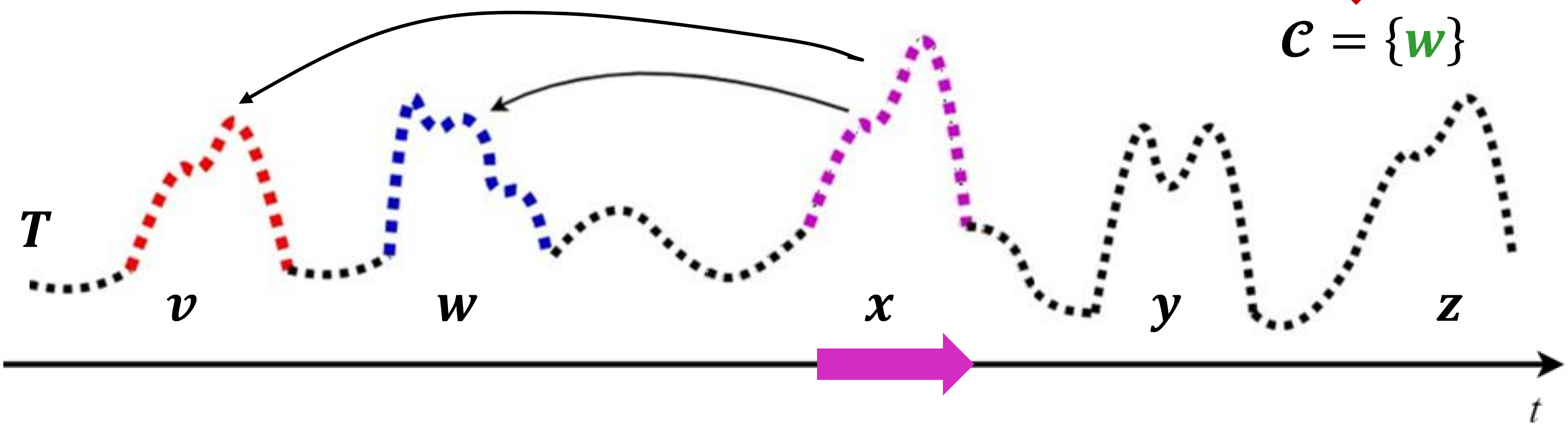
  if $isCandidate$ = TRUE **then** $\boldsymbol{\mathcal{C}} := \boldsymbol{\mathcal{C}} \cup \boldsymbol{s}$

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

$\mathrm{ED}(\boldsymbol{x}, \boldsymbol{v}) < r$

$\mathrm{ED}(\boldsymbol{x}, \boldsymbol{w}) \geq r$

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



Discord discovery: Selection

while not end of $\boldsymbol{T}$

  get current subsequence $\boldsymbol{s}$

  $isCandidate$:= TRUE

  for each $\boldsymbol{c_i} \in \boldsymbol{\mathcal{C}}$ **and** $\boldsymbol{s} \cap \boldsymbol{c_i} = \emptyset$

    if $\mathrm{ED}(\boldsymbol{s}, \boldsymbol{c_i}) < r$ **then**

      $\boldsymbol{\mathcal{C}} \coloneqq \boldsymbol{\mathcal{C}} \setminus \boldsymbol{c_i}$; $isCandidate \coloneqq$ FALSE

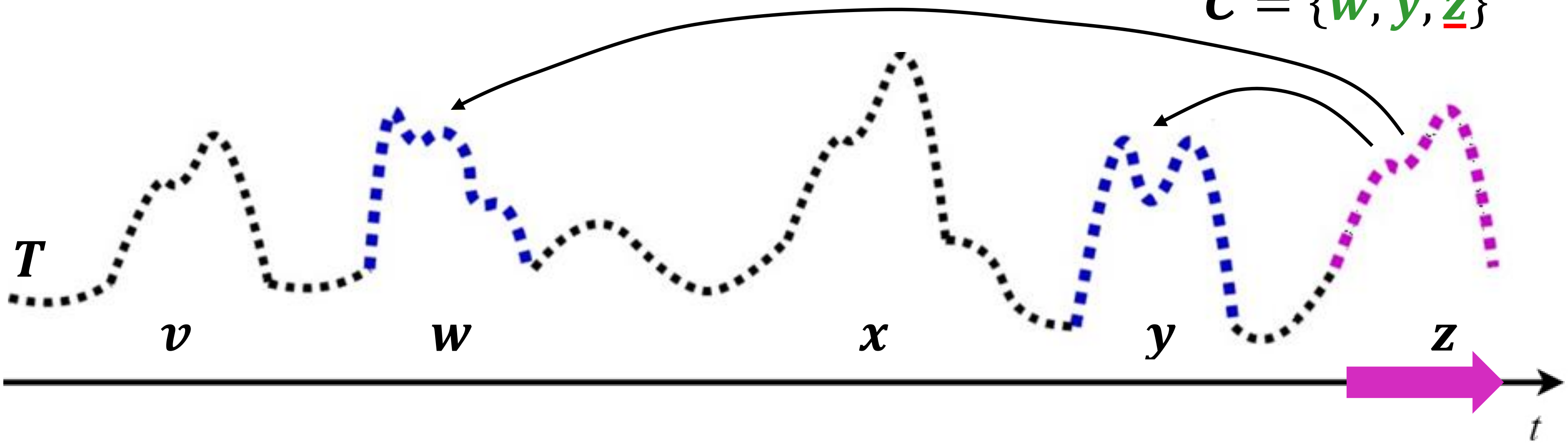
  if $isCandidate =$ TRUE **then** $\boldsymbol{\mathcal{C}} \coloneqq \boldsymbol{\mathcal{C}} \cup \boldsymbol{s}$

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

$\mathrm{ED}(\boldsymbol{z}, \boldsymbol{w}) \geq r$

$\mathrm{ED}(\boldsymbol{z}, \boldsymbol{y}) \geq r$

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



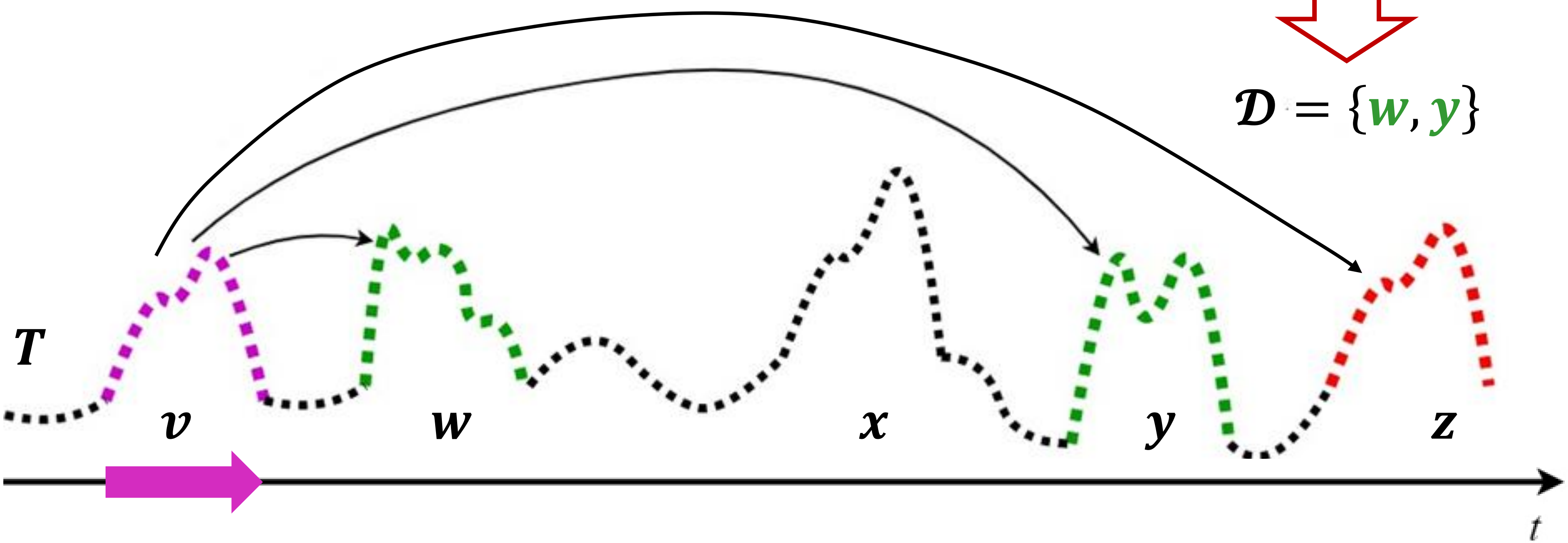
Discord discovery: Refinement

$\mathcal{D} := \mathcal{C}$
while not end of T
get current subsequence s
for each $d_i \in \mathcal{D}$ **and** $s \cap d_i = \emptyset$
if $\mathrm{ED}(s, d_i) < r$ **then**
$\mathcal{D} := \mathcal{D} \setminus d_i$

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

$\mathrm{ED}(v, w) \geq r$
$\mathrm{ED}(v, y) \geq r$
$\mathrm{ED}(v, z) < r$

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



Discord discovery: Refinement

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

while not end of $\boldsymbol{T}$

  get current subsequence $\boldsymbol{s}$

  for each $\boldsymbol{d_i} \in \mathcal{D}$ **and** $\boldsymbol{s} \cap \boldsymbol{d_i} = \emptyset$

    if $\mathrm{ED}(\boldsymbol{s}, \boldsymbol{d_i}) < r$ **then**

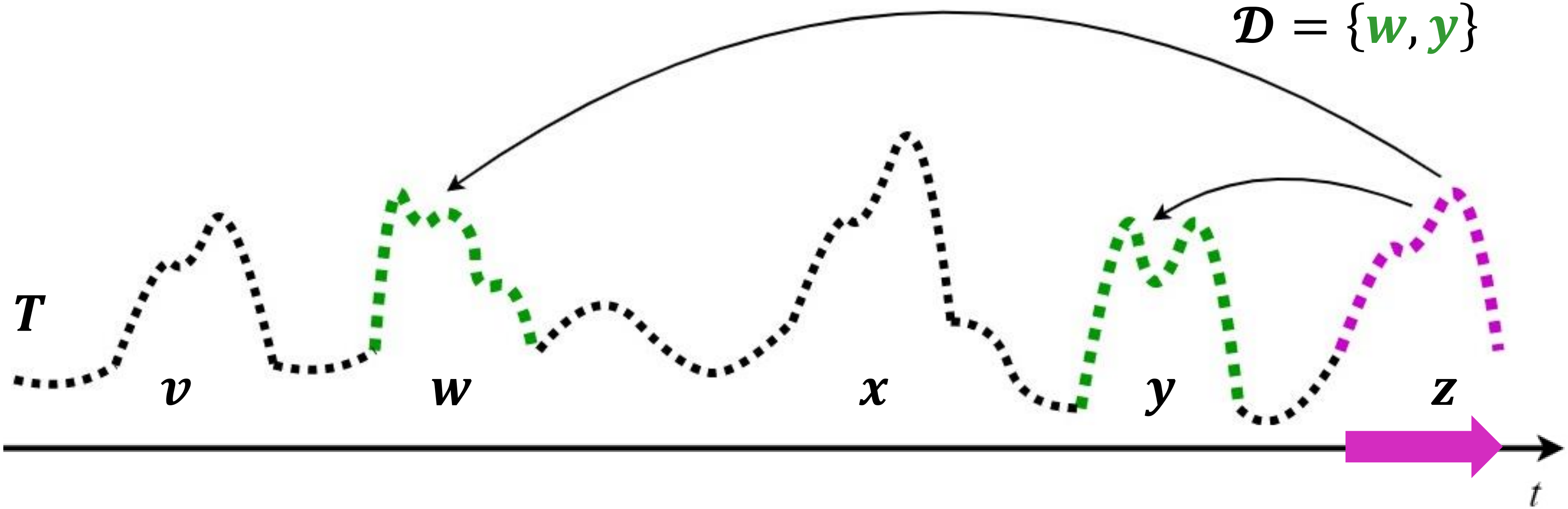
      $\mathcal{D} := \mathcal{D} \setminus \boldsymbol{d_i}$

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

$\mathrm{ED}(\boldsymbol{z}, \boldsymbol{w}) \geq r$

$\mathrm{ED}(\boldsymbol{z}, \boldsymbol{y}) \geq r$

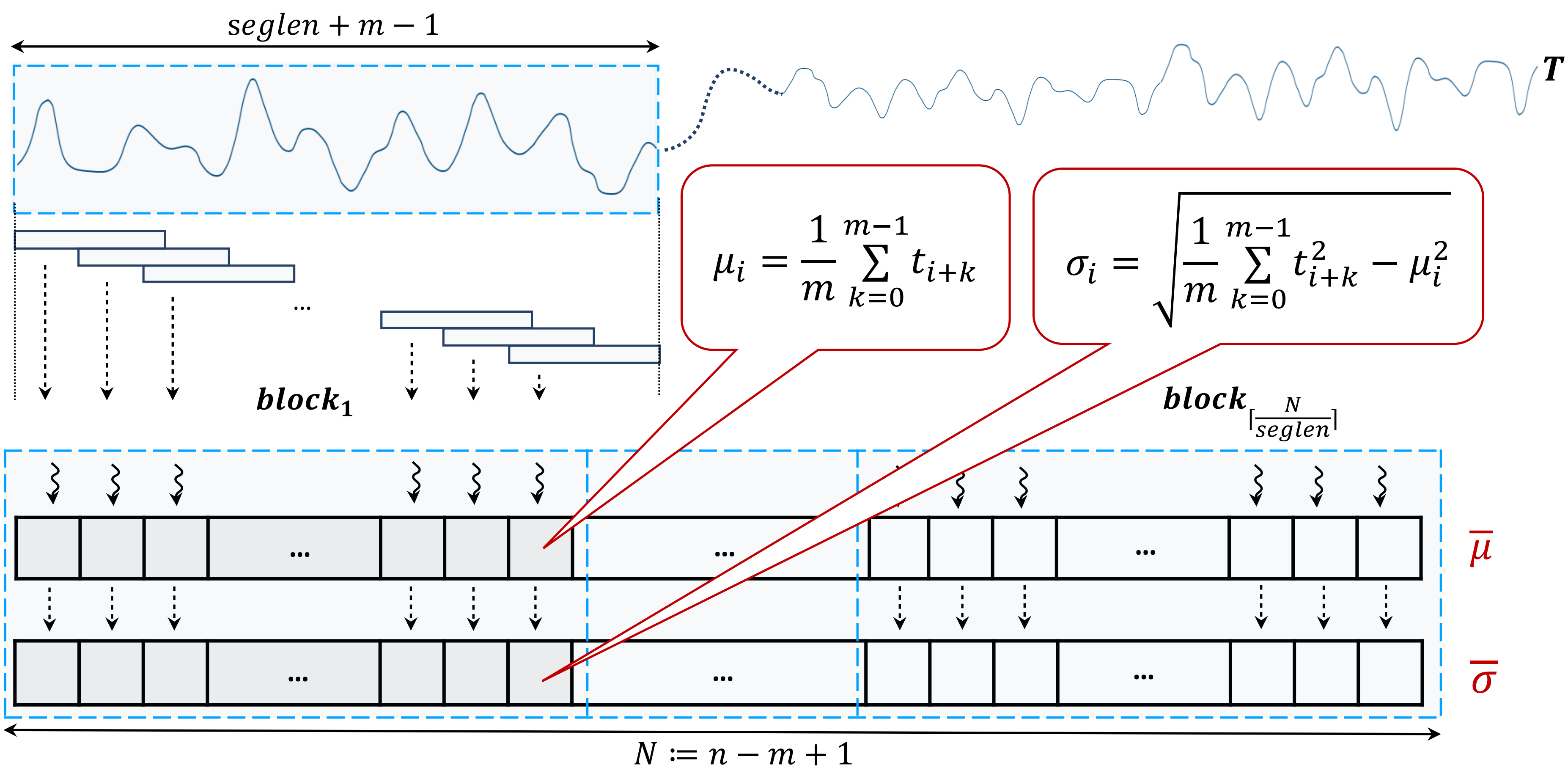
$\mathcal{D} = \{\boldsymbol{w}, \boldsymbol{y}\}$



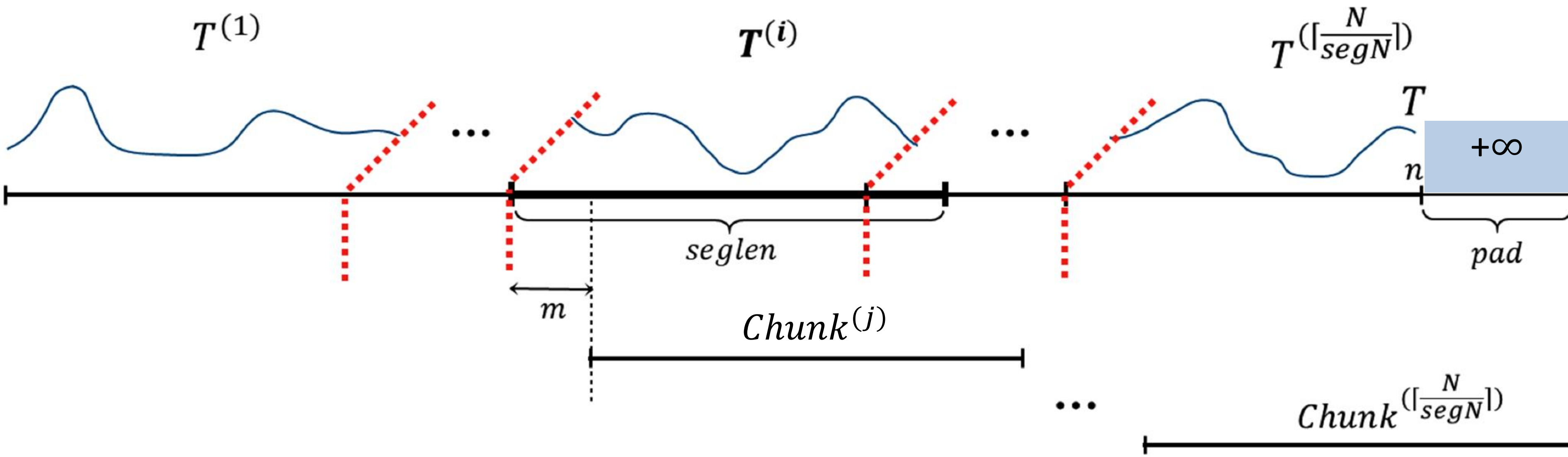
Parallel discord discovery: Preprocessing

For highest possible performance, let us use **quadratic ED**

$$\mathrm{ED}^2_{\mathrm{norm}}(T_{i,m}, T_{j,m}) = 2m\left(1 - \frac{<T_{i,m}, T_{j,m}> - m\mu_i\mu_j}{m\sigma_i\sigma_j}\right)$$

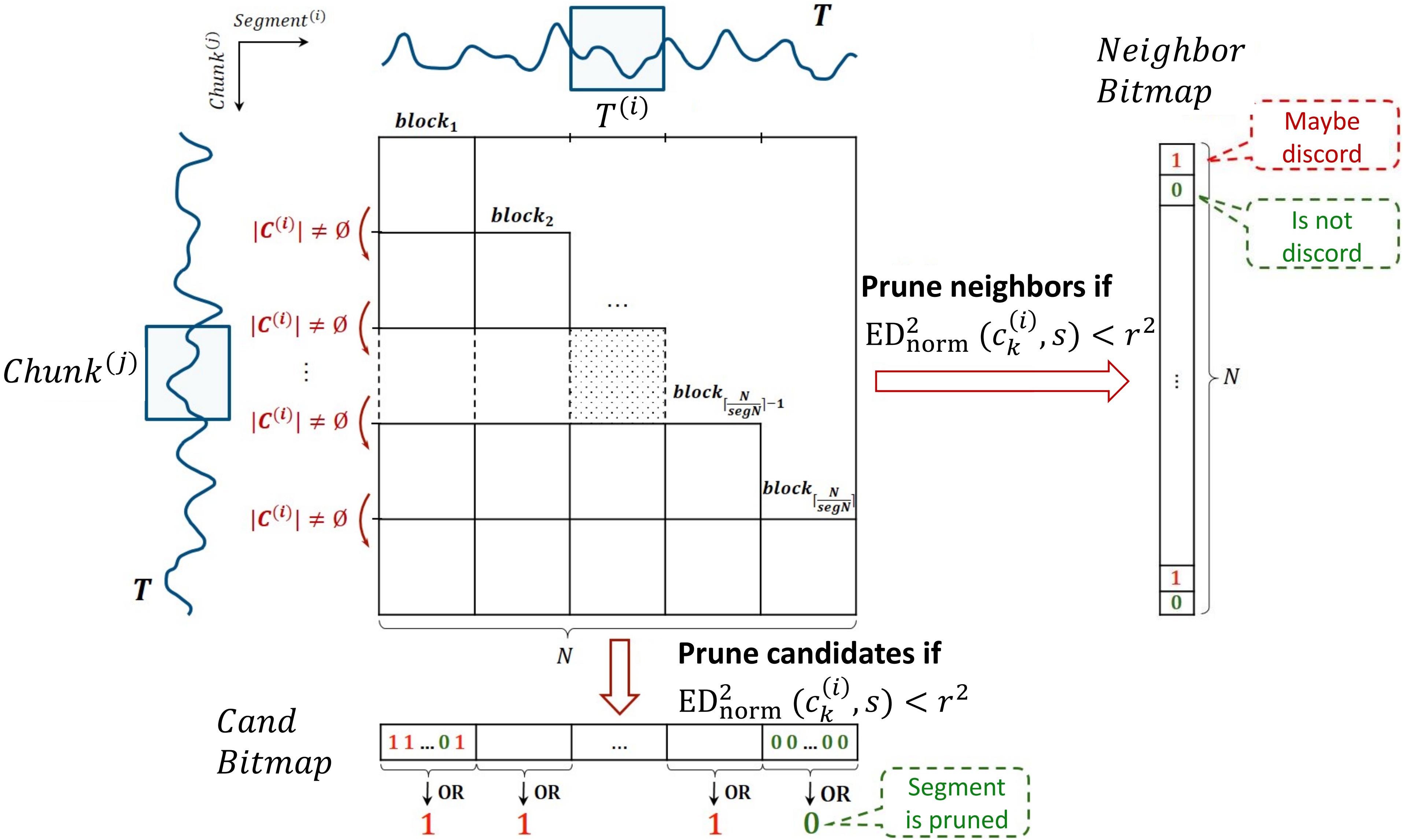


Parallel discord discovery: Segmentation

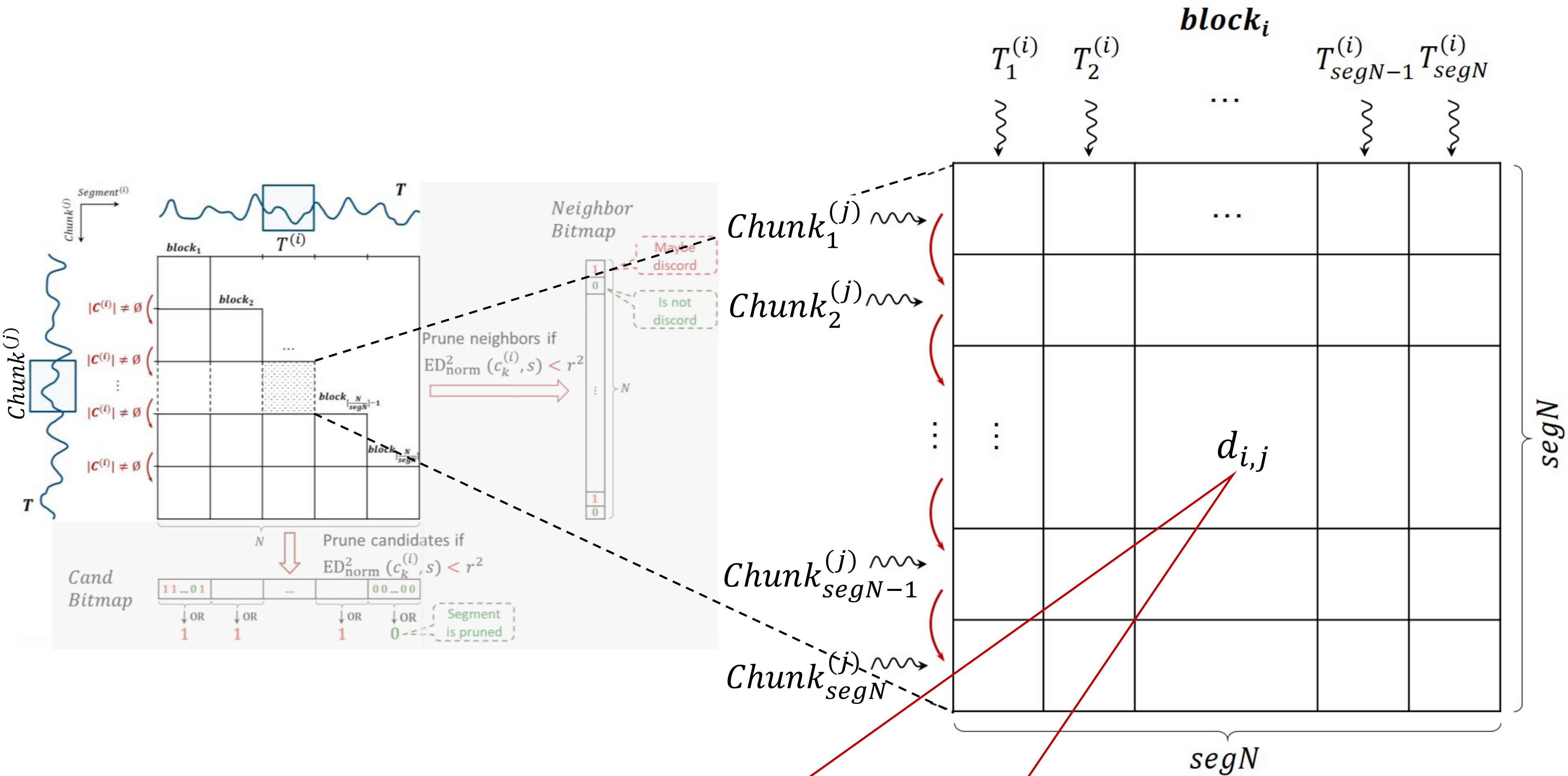


Parameter	Semantic
$T^{(i)}$	Segment to select (prune) candidates
$seglen = segN + m - 1$	Segment length
$segN = k \cdot warpsize$	# candidates in a segment
$warpsize = 32$	# threads in a group within a thread block
$Chunk^{(j)}$	Interval to test its subsequences against a seg. candidates
pad	# dummy elements

Parallel discord discovery: Selection (blocks)



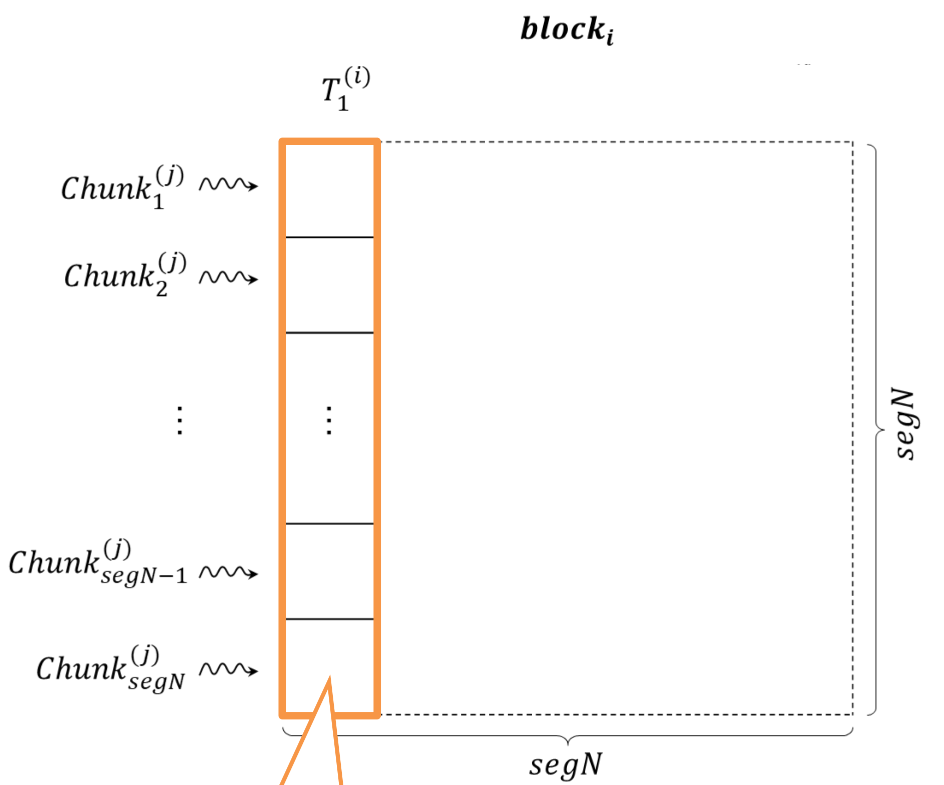
Parallel discord discovery: Selection (threads)



$$d_{i,j} = \mathrm{ED}^2_{\mathrm{norm}}(T_{i,m}, T_{j,m}) = 2m\left(1 - \frac{< T_{i,m}, T_{j,m} > - m\mu_i\mu_j}{m\sigma_i\sigma_j}\right)$$

Parallel discord discovery: Selection (threads)

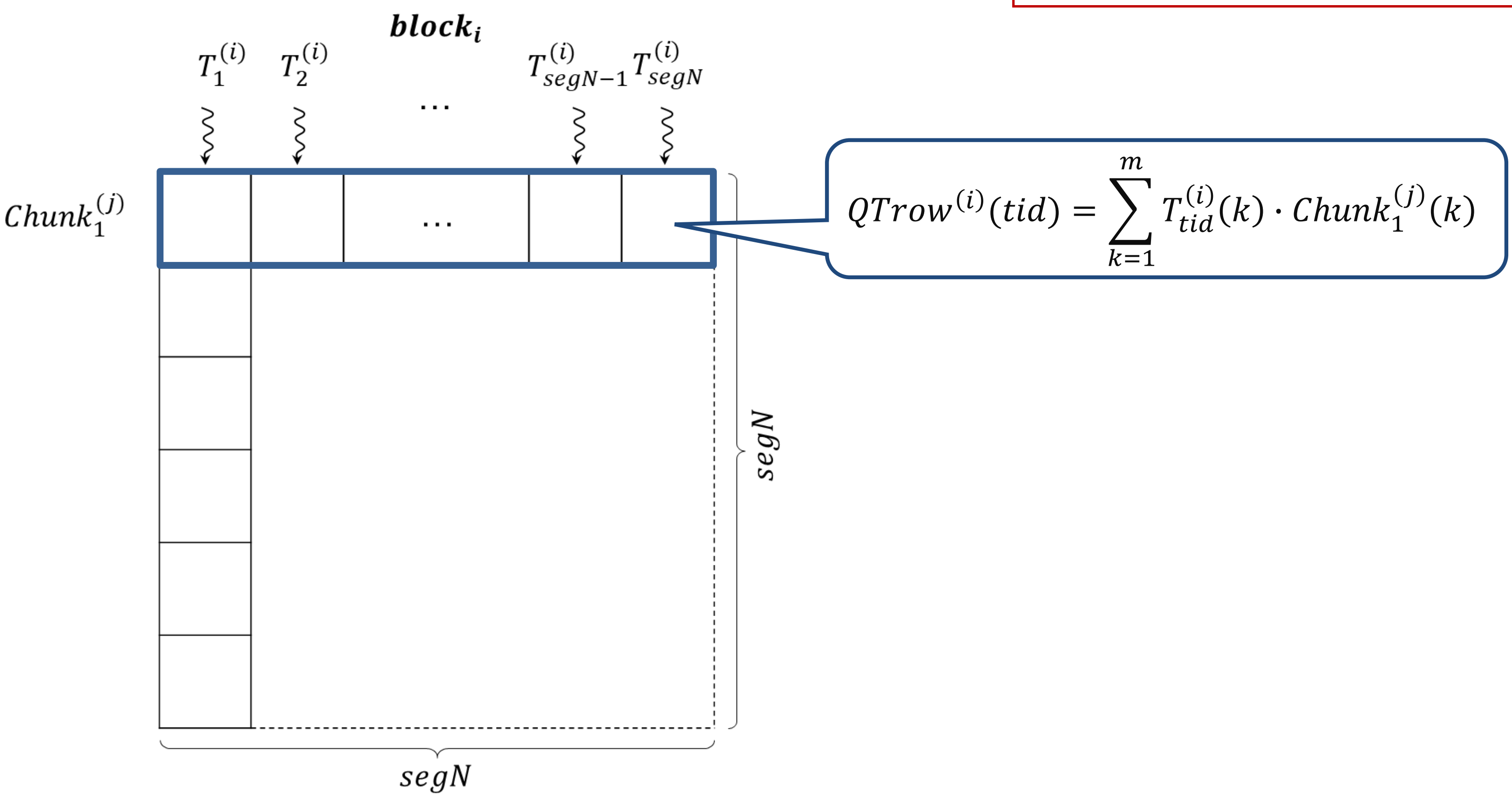
$$d_{i,j} = 2m\left(1 - \frac{< T_{i,m}, T_{j,m} > - m\mu_i\mu_j}{m\sigma_i\sigma_j}\right)$$



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

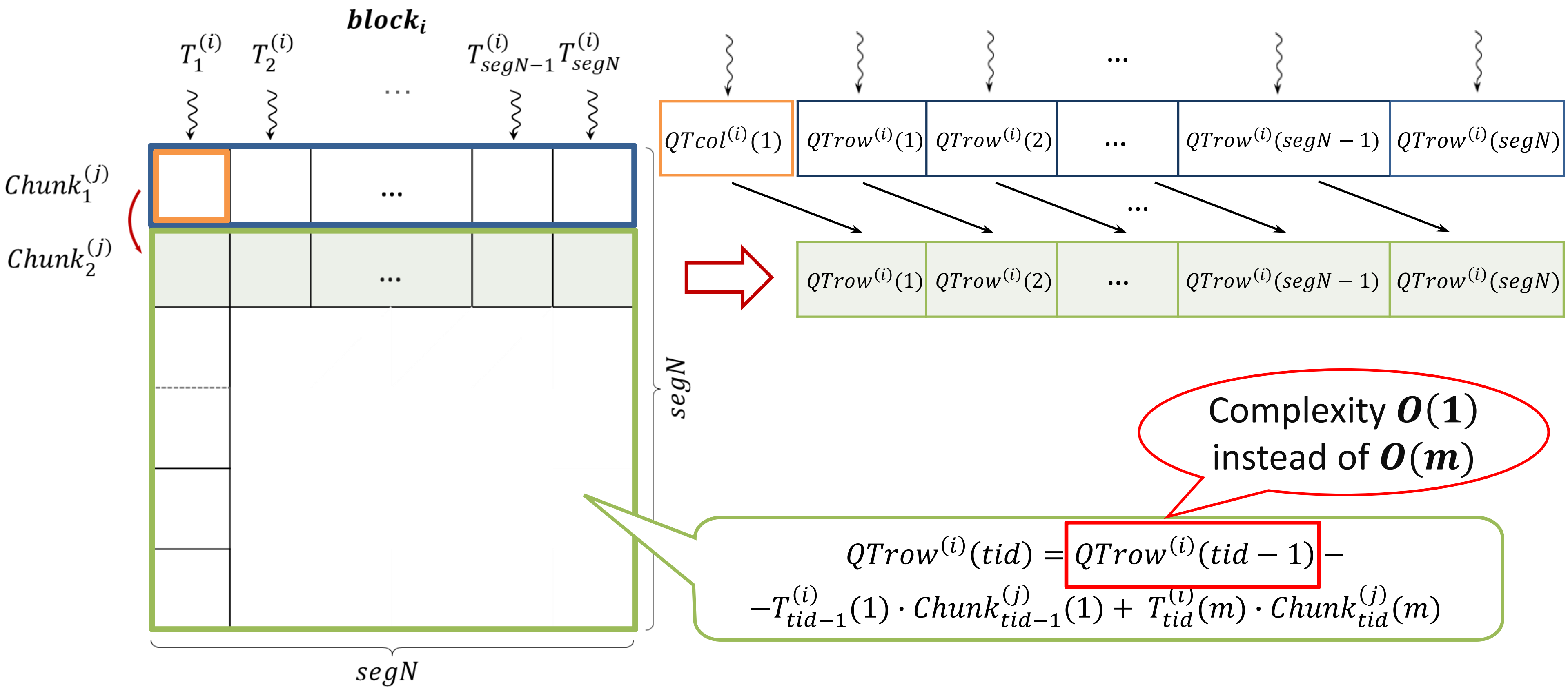
Parallel discord discovery: Selection (threads)

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



Parallel discord discovery: Selection (threads)

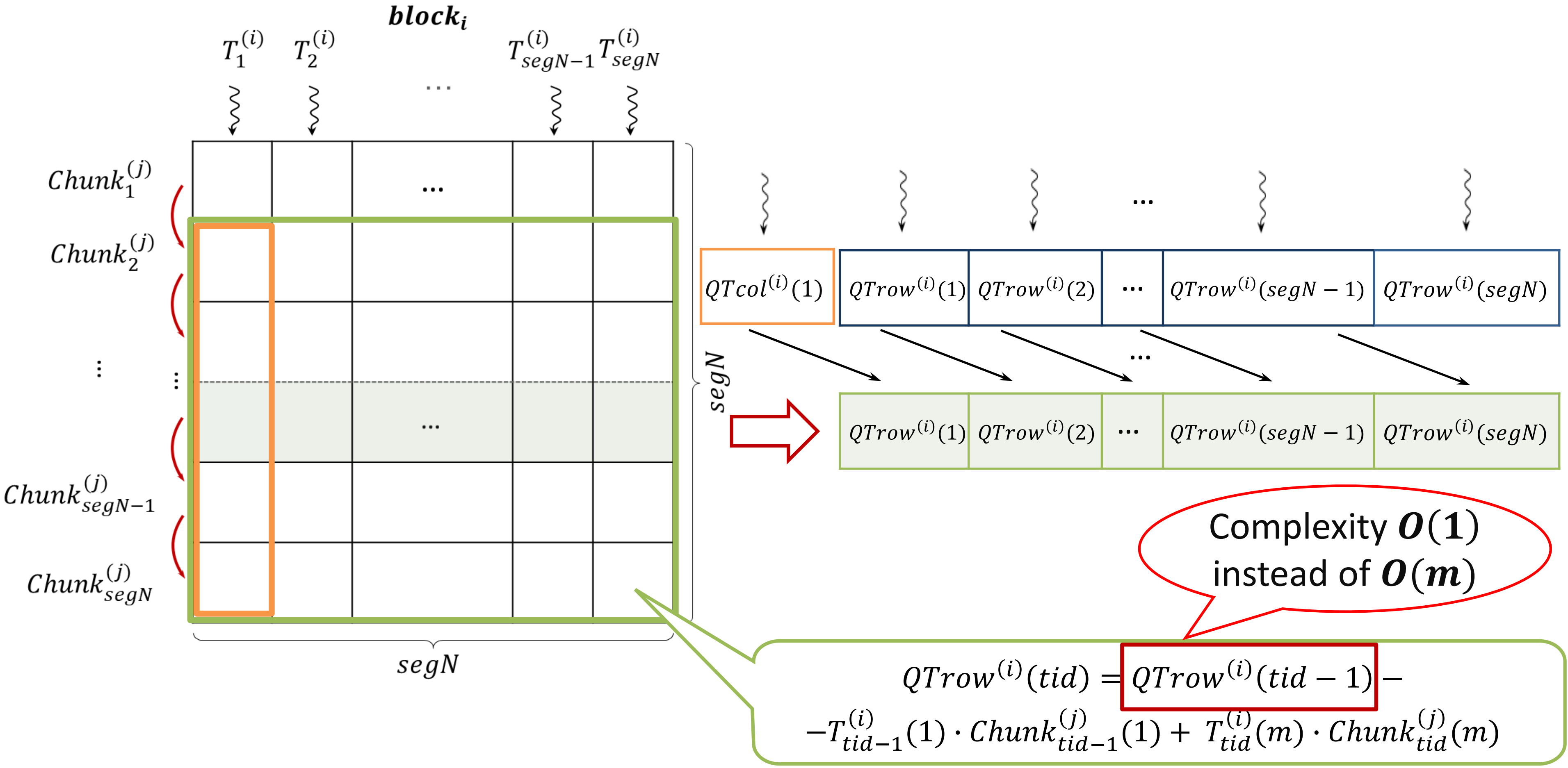
$$d_{i,j} = 2m\left(1 - \frac{< T_{i,m}, T_{j,m} > - m\mu_i\mu_j}{m\sigma_i\sigma_j}\right)$$



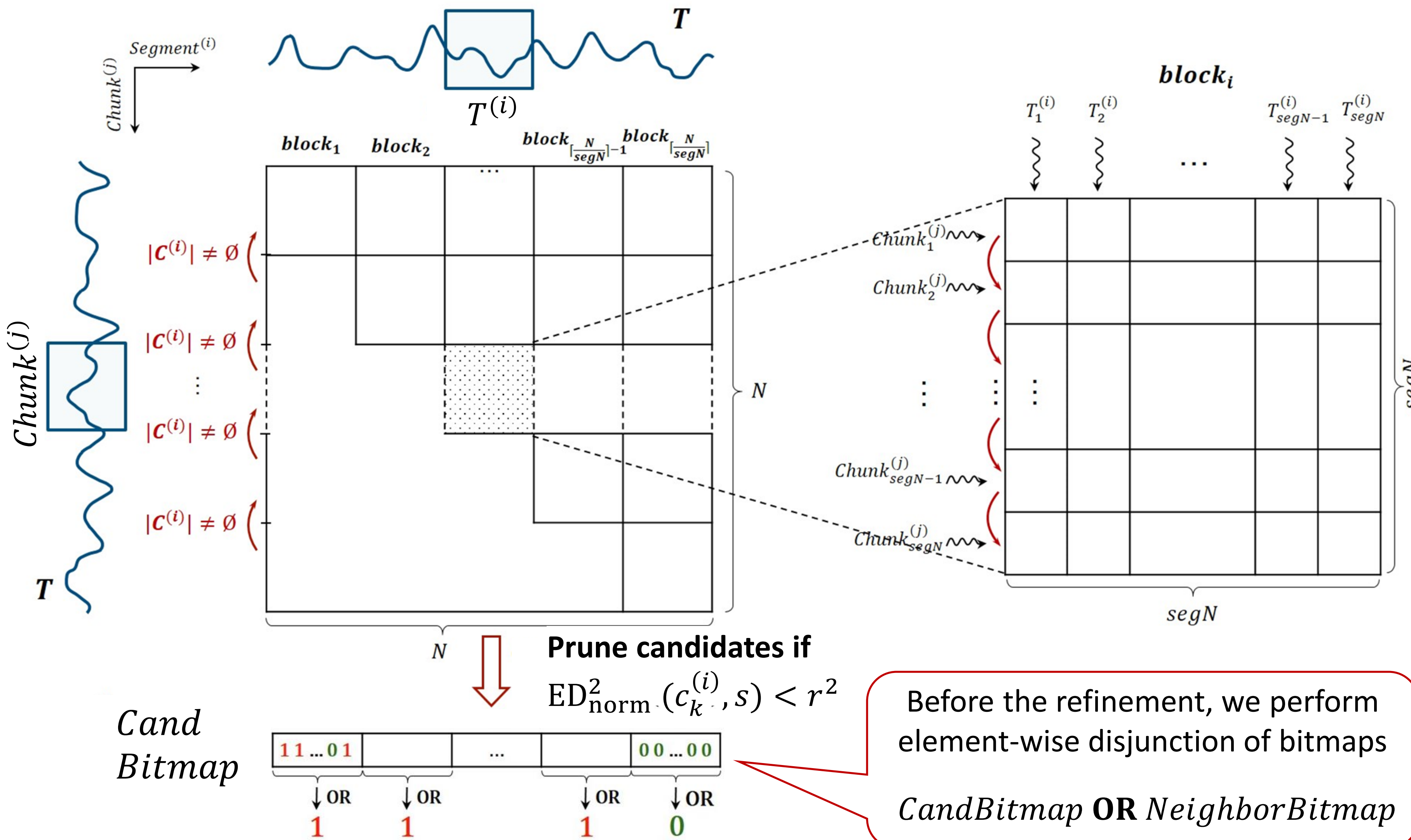
$$QTrow^{(i)}(tid) = QTrow^{(i)}(tid-1) - T^{(i)}_{tid-1}(1) \cdot Chunk^{(j)}_{tid-1}(1) + T^{(i)}_{tid}(m) \cdot Chunk^{(j)}_{tid}(m)$$

Parallel discord discovery: Selection (threads)

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



Parallel discord discovery: Refinement



Before the refinement, we perform element-wise disjunction of bitmaps

CandBitmap **OR** *NeighborBitmap*

Parallel discord discovery: Experiments

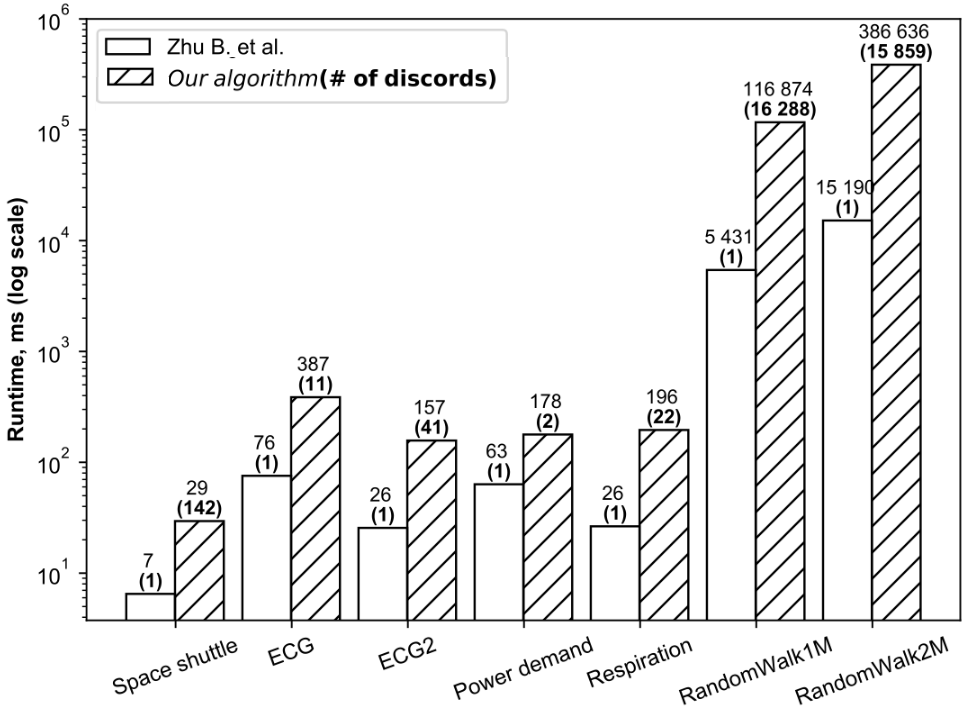
- **Hardware**
  - NVIDIA Tesla V100 SXM2 (5120 cores @1.3 GHz)
- **Data**

Time series	Length, n	Discord, m	Domain
Space shuttle	5 000	150	Measurements of a sensor on the NASA spacecraft[1]
ECG	45 000	200	ECG of an adult patient[2]
ECG2	21 600	400	
Power demand	33 220	750	Annual energy consumption of an office[3]
Respiration	24 125	250	Human breathing by chest expansion[4]
RandomWalk1M	10^7	512	Synthetic time series
RandomWalk2M	$2 \cdot 10^7$	512	

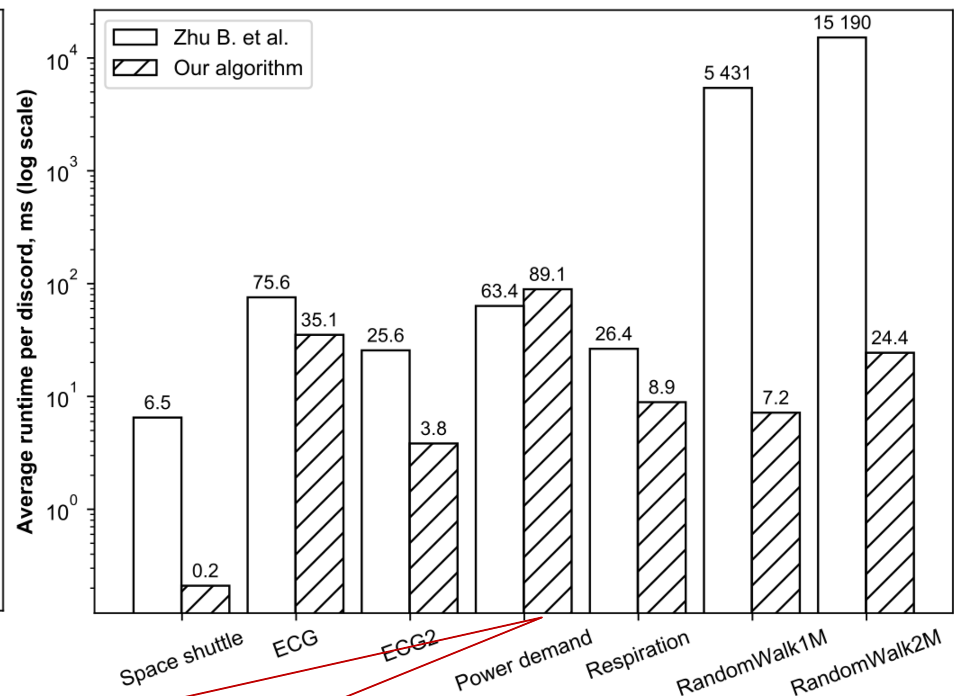
[1] Ferrell B., *et al.* NASA shuttle valve data 2005. URL: http://www.cs.fit.edu/~pkc/nasa/data/
[2] Goldberger A., *et al.* PhysioBank, PhysioToolkit, and PhysioNet: components of a new research resource for complex physiologic signals. Circulation 101(23): 215–220. DOI: 10.1161/01.CIR.101.23.e215
[3] van Wijk J.J., *et al.* Cluster and calendar based visualization of time series data. INFOVIS'99: 4–9. DOI: 10.1109/INFVIS.1999.801851
[4] Keogh E., *et al.* HOT SAX: Finding the most unusual time series subsequence: Algorithms and applications. ICDM 2004: 440–449. URL: http://www.cs.ucr.edu/~eamonn/discords/

Parallel discord discovery: Experiments

Running time
to discover **all** discords



Average running time
to discover **one** discord

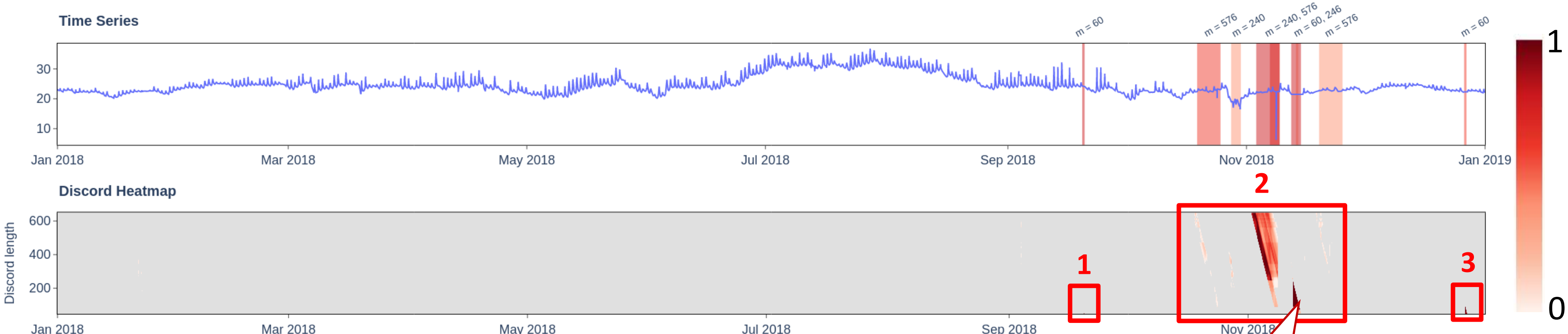


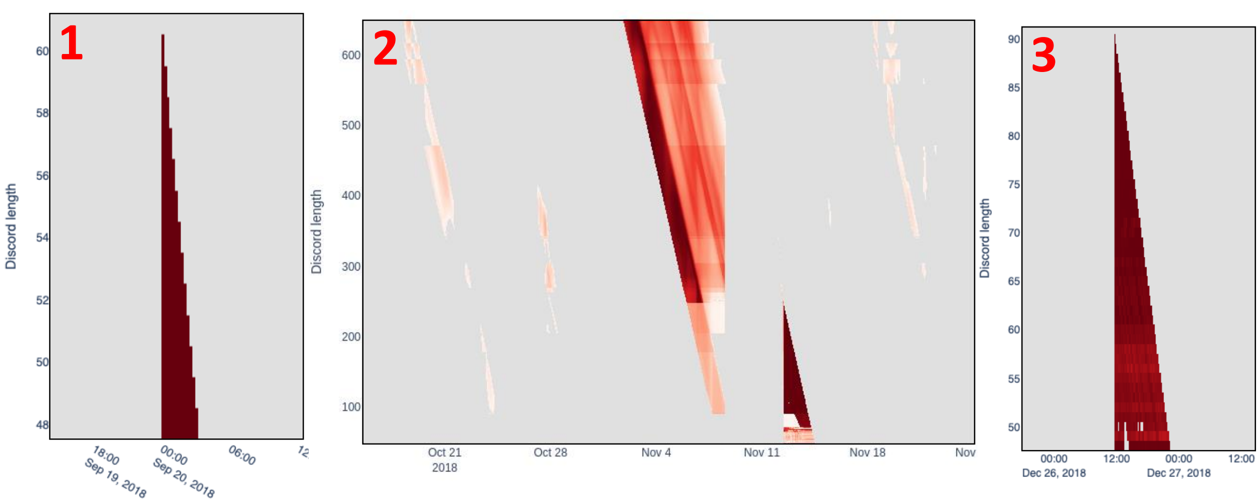
Parallel version is up to 30 times faster (for real data)

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

Parallel discord discovery: Case studies

Anomalies of a t° sensor (freq. is 4 times per hour) for 0.5-7 days in the SUSU Campus Smart heating system





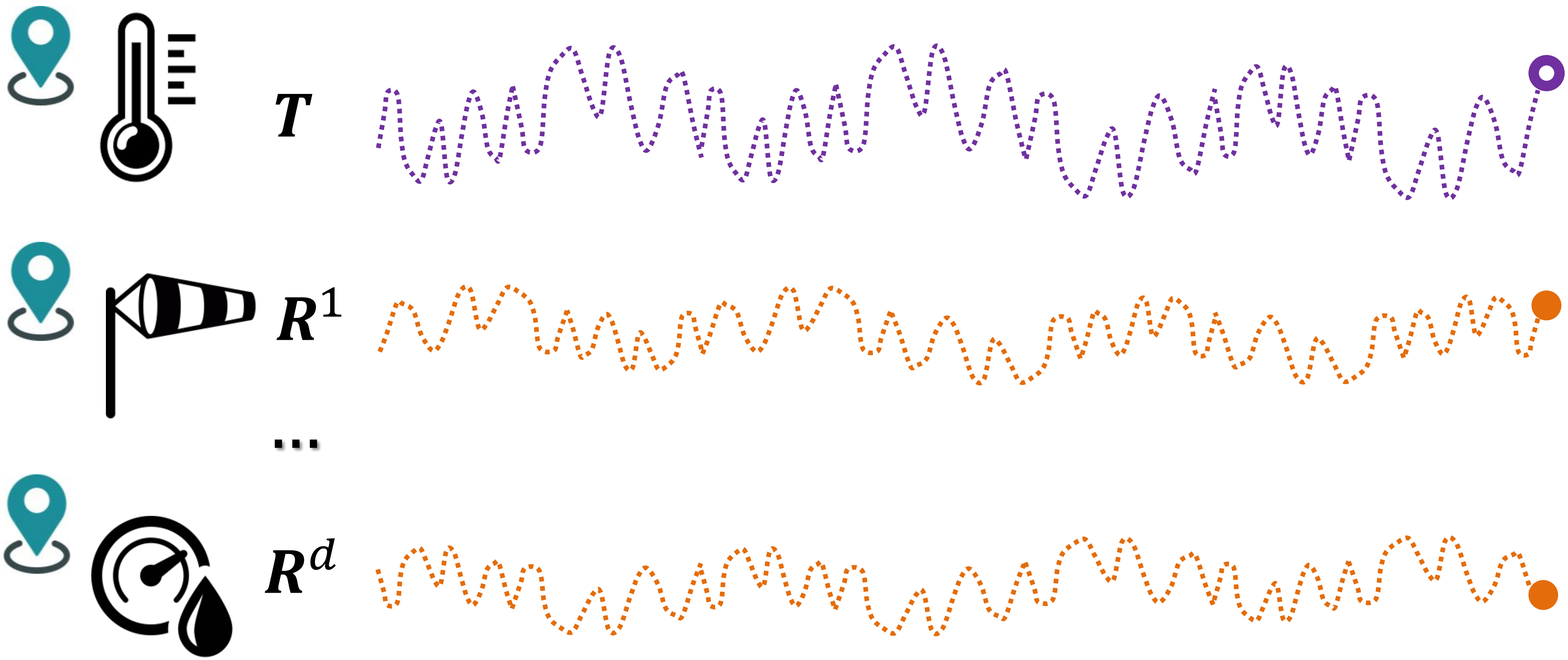
Minmax normalization

$$map_{i,m} = \frac{T_{i,m}.nndist - \min_{d \in D_i} d.nndist}{\max_{d \in D_i} d.nndist - \min_{d \in D_i} d.nndist}$$

Outline

- Introduction
- Parallel pattern discovery
- Parallel anomaly detection
- **Parallel imputation of missing values**
- Online time series analytics with parallel algorithms

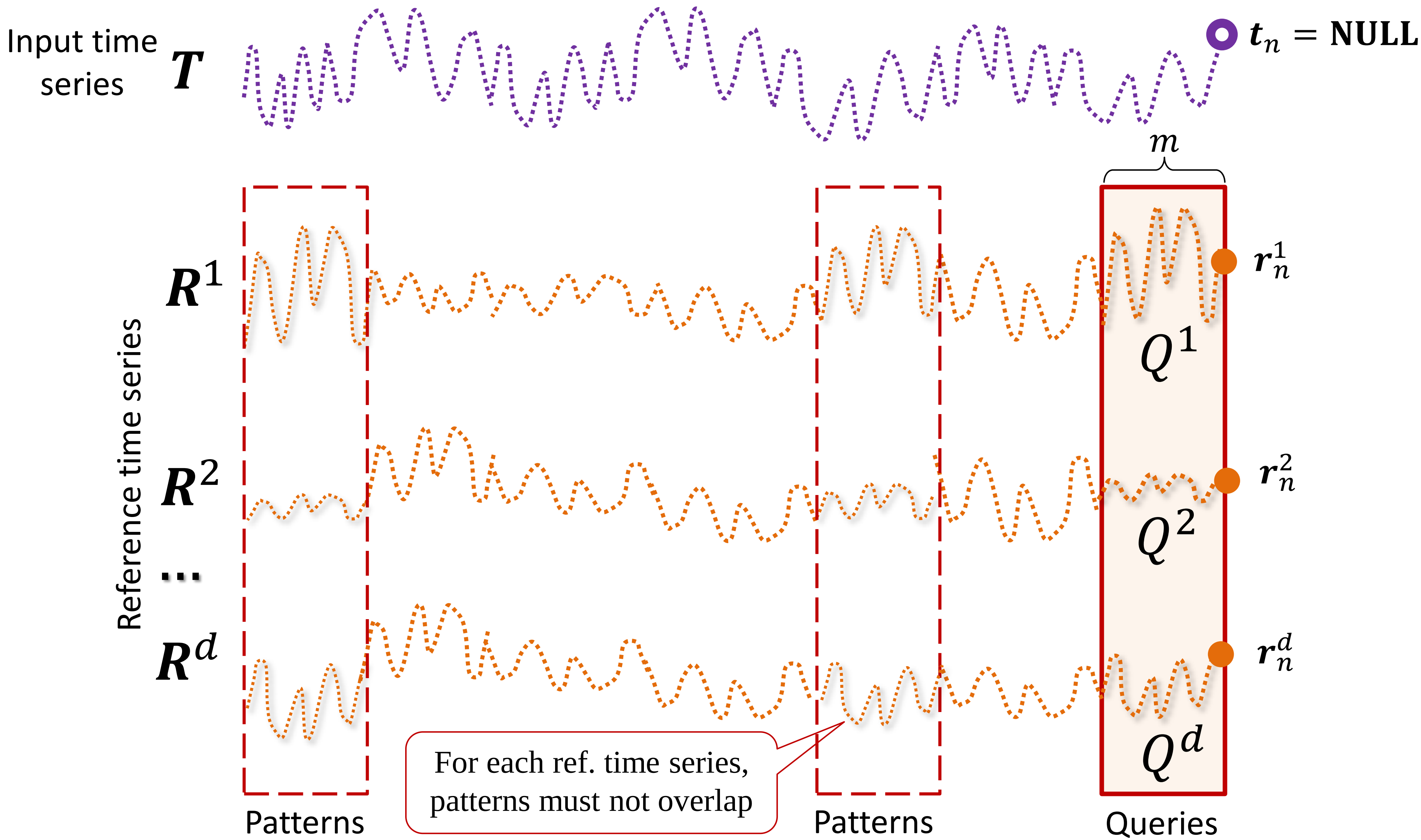
Imputation through reference time series



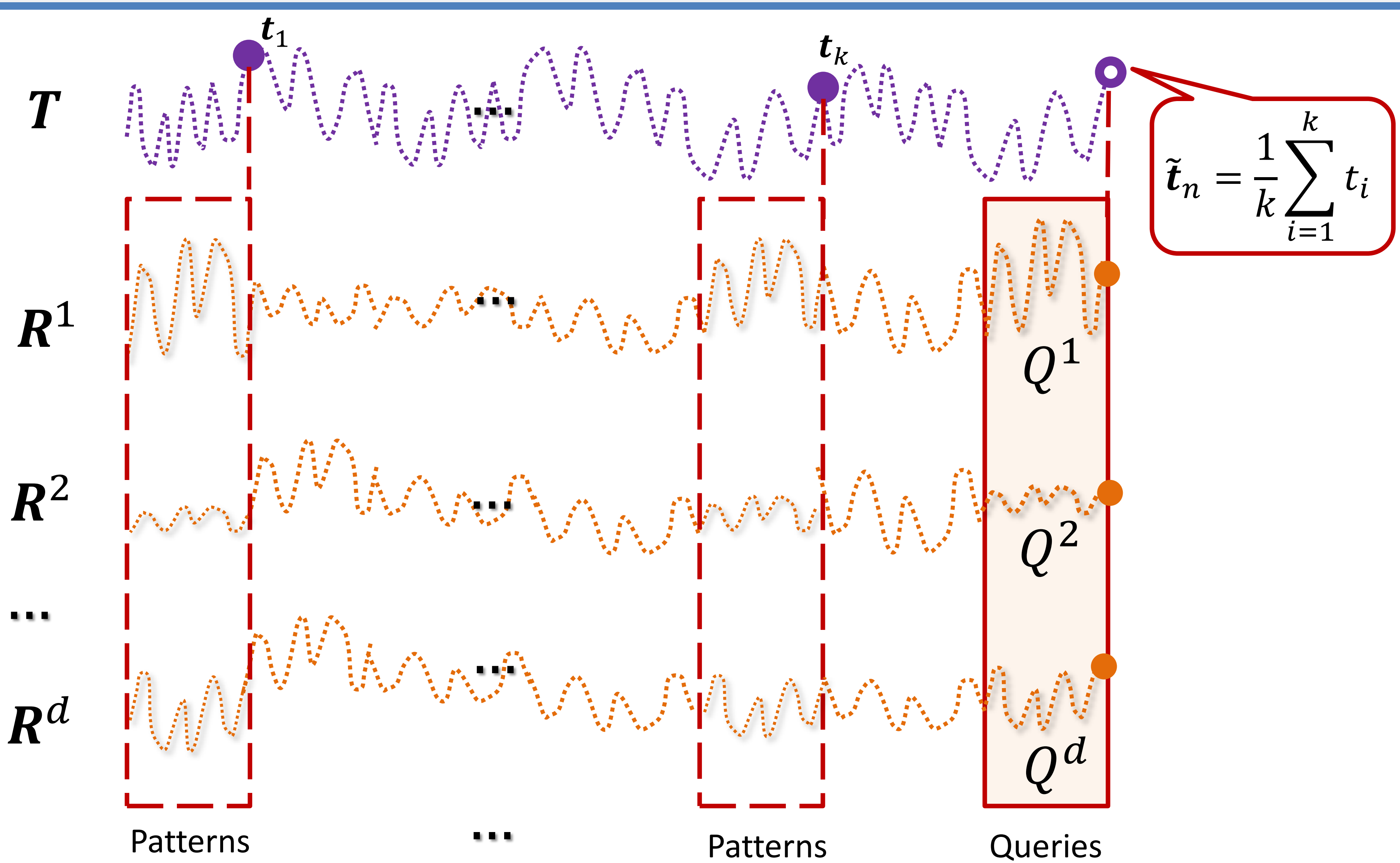
Heuristics

the time series undergo imputation and the reference time series behave similarly in the same subsequences

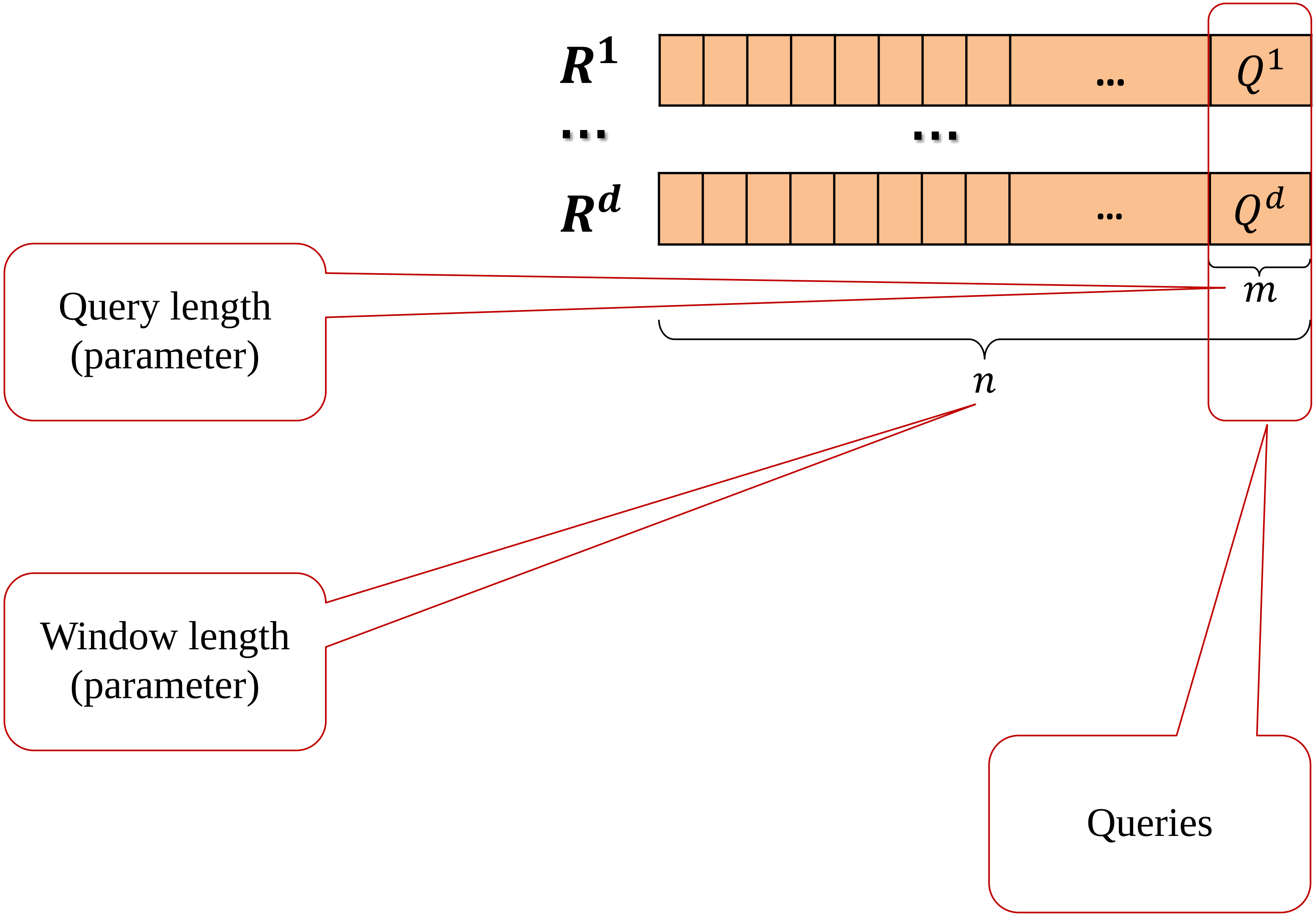
Imputation: Find k patterns



Imputation: Reconstruction of missing value



Parallel imputation: Reference time series



Parallel imputation: Pattern search

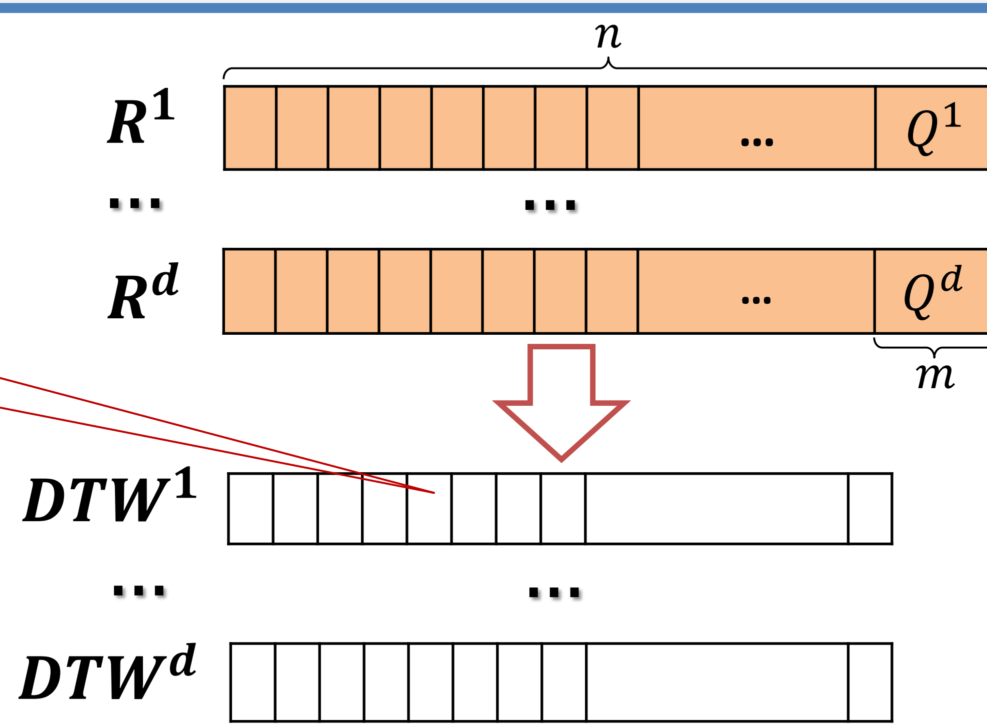
Distance between the interval $R^i[j:m]$ and the query Q^i w.r.t. the DTW distance measure

1. Pattern search

for $i := 1$ **to** d **do**

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

    $DTW^i(R^i[j:m], Q^i)$



Parallel imputation: Scoring

1. Pattern search

for $i := 1$ **to** d **do**

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

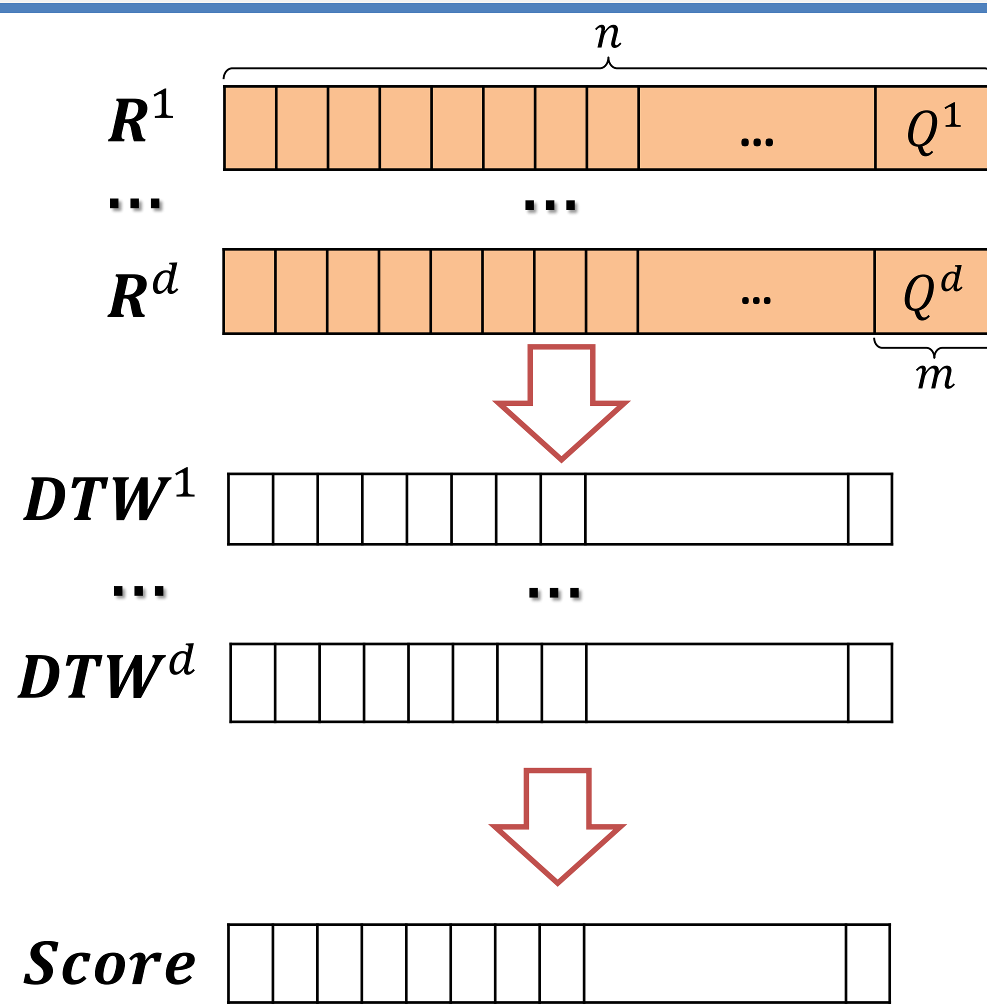
    $DTW^i(R^i[j:m], Q^i)$

2. Scoring

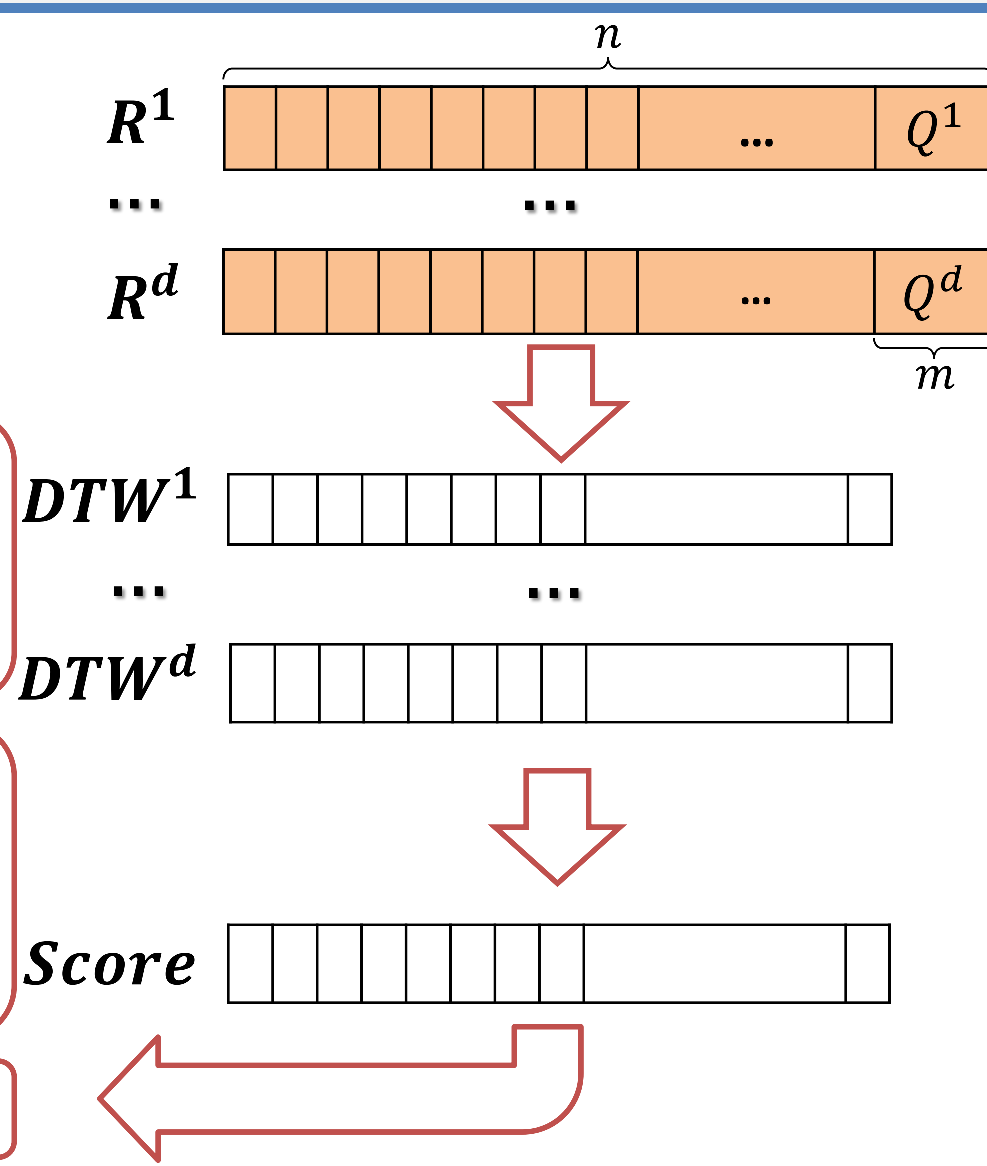
for $j := 1$ **to** $n - m + 1$ **do**

  for $i := 1$ **to** d **do**

    $Score(j) := Score(j) + \frac{weight(j,i)}{DTW^i(j)+\varepsilon}$



Parallel imputation: Reconstruction



1. Pattern search

for $i := 1$ **to** d **do**

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

  $DTW^i(R^i[j : m], Q^i)$

2. Scoring

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

  for $i := 1$ **to** d **do**

  $Score(j) := Score(j) + \frac{weight(j,i)}{DTW^i(j)+\varepsilon}$

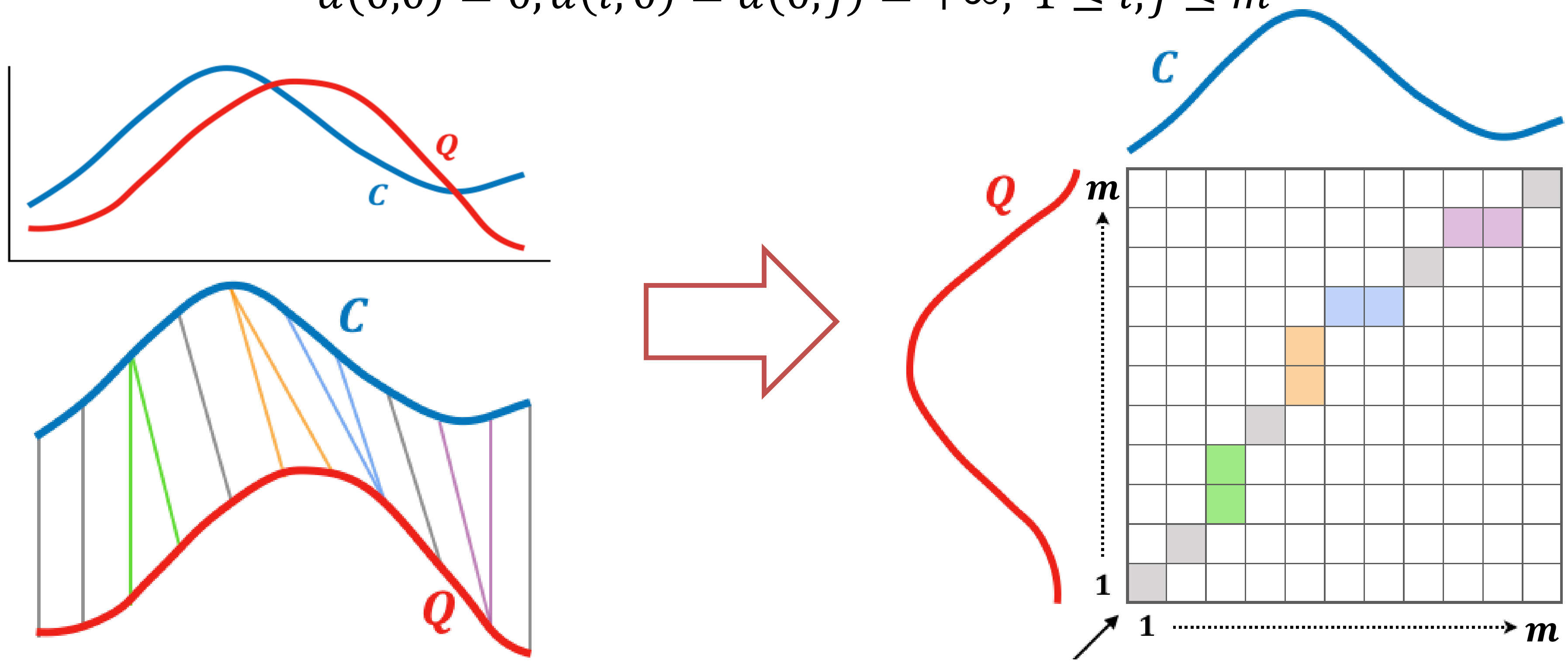
3. Selection of TOP-k intervals, reconstruction

DTW (Dynamic Time Warping) distance measure*

$$\mathbf{DTW}(Q, C) = d(m, m)$$

$$d(i,j) = (q_i - c_i)^2 + \min \begin{cases} d(i-1, j) \\ d(i, j-1) \\ d(i-1, j-1) \end{cases}$$

$$d(0,0) = 0, d(i,0) = d(0,j) = +\infty;\ 1 \le i,j \le m$$



* Berndt D.J., Clifford J. Using Dynamic Time Warping to Find Patterns in Time Series. KDD & AAAI Workshop 1994. TR-WS-94-03. P. 359-370.

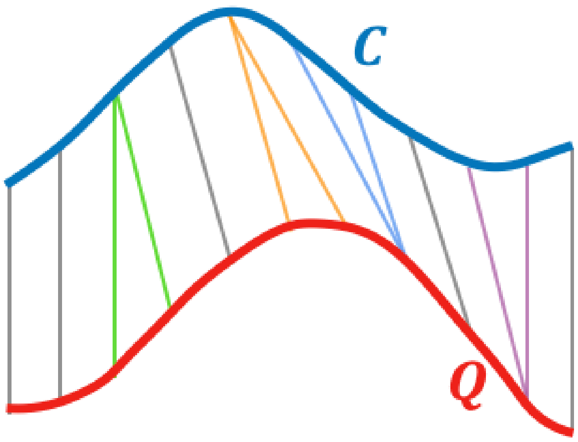
DTW: complexity *vs.* accuracy

Complexity
$\boldsymbol{O(m^2)}$

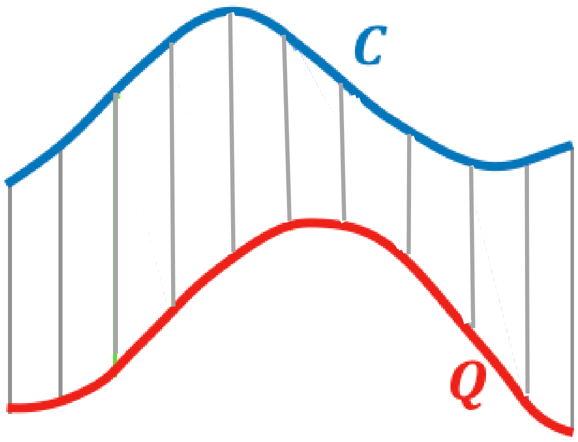
$$\mathbf{DTW}(Q, C) = d(m, m)$$

$$d(i,j) = (q_i - c_i)^2 + \min \begin{cases} d(i-1, j) \\ d(i, j-1) \\ d(i-1, j-1) \end{cases}$$

$$d(0,0) = 0, d(i,0) = d(0,j) = +\infty;\ 1 \le i,j \le m$$



DTW is better to measure the similarity by shape than the Euclidean distance



DTW acceleration: Sakoe–Chiba band*

Complexity $O(mr)$

$$\mathbf{DTW}(Q, C) = d(m, m)$$

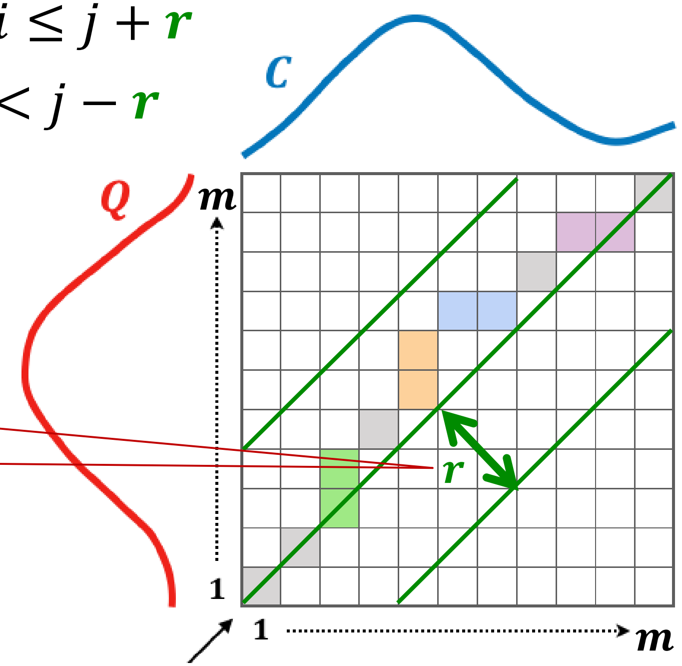
$$d(i,j) = (q_i - c_i)^2 + \min \begin{cases} d(i-1, j) \\ d(i, j-1) \\ d(i-1, j-1) \end{cases}$$

$$d(0,0) = 0, d(i,0) = d(0,j) = +\infty;\ 1 \le i, j \le m;$$

$$\mathbf{0} \le \boldsymbol{r} \le \boldsymbol{m} - \mathbf{1}, j - \boldsymbol{r} \le i \le j + \boldsymbol{r}$$

$$d(i,j) = +\infty, j + \boldsymbol{r} < i < j - \boldsymbol{r}$$

Complexity *vs.* accuracy trade-off



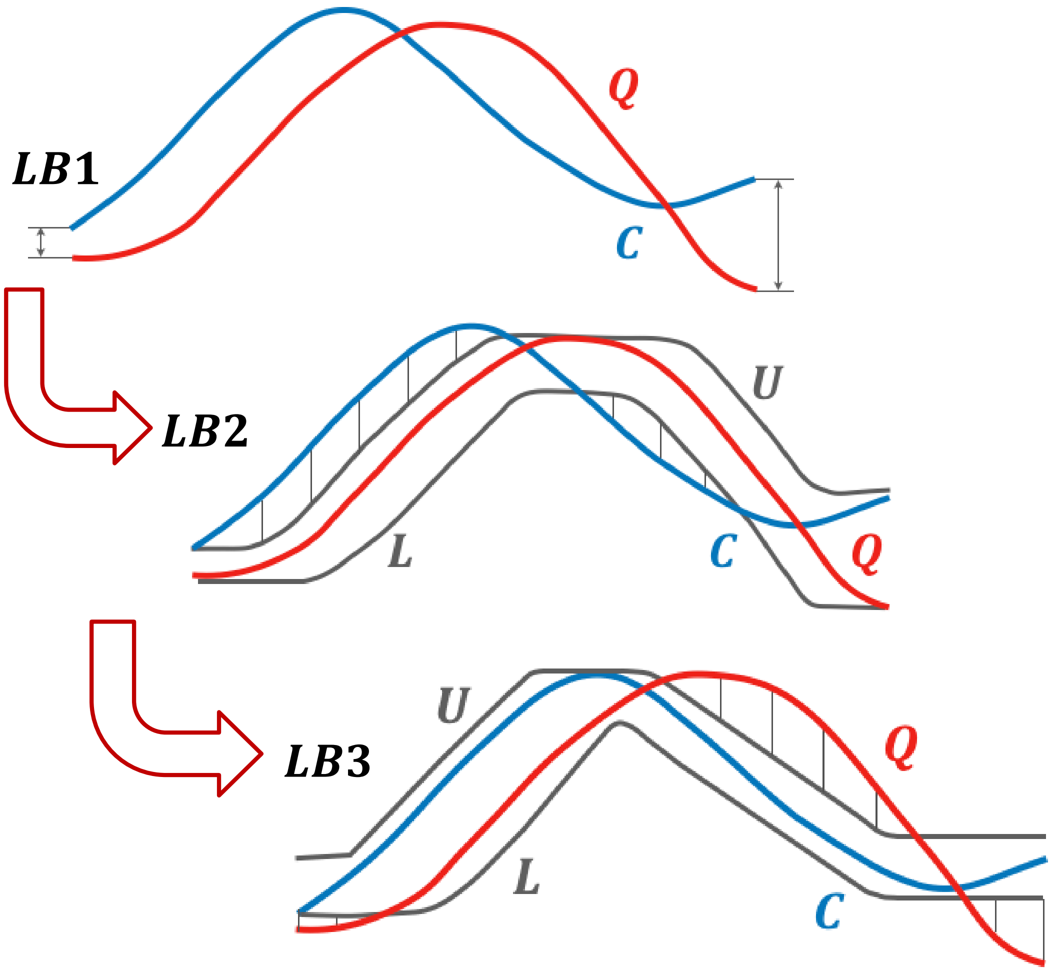
* Sakoe H., Chiba S. Dynamic Programming Algorithm Optimization for Spoken Word Recognition. IEEE Trans. on Acoustics, Speech, and Signal Processing. 1978. Vol. 26. P. 43-49.

DTW acceleration: Lower bounding*

- Lower bound function (LB)
  - LB: $\mathbb{R}^m \times \mathbb{R}^m \rightarrow \mathbb{R}_+$, complexity is lower than $O(m^2)$ $\forall\, R[i{:}m], Q{:}\ \mathrm{LB}(R[i{:}m], Q) \leq \mathrm{DTW}(R[i{:}m], Q)$
- Lower bounding pruning
  - Best-so-far minimum of DTW: bsf
  - if $\mathrm{LB}(R[i{:}m], Q) > bsf$, then $\mathrm{DTW}(R[i{:}m], Q) > bsf$, so $R[i{:}m]$ is clearly dissimilar to Q without DTW calculation
  - It works only if $R[i{:}m]$ and Q are z-normalized
  - LBs can be applied in a cascade

* Rakthanmanon T., *et al.* Addressing Big Data Time Series: Mining Trillions of Time Series Subsequences Under Dynamic Time Warping. ACM Trans. Knowl. Discov. Data. 2013. Vol. 7, no. 3. 10:1–10:31.

Lower bounding: Cascade of LBs

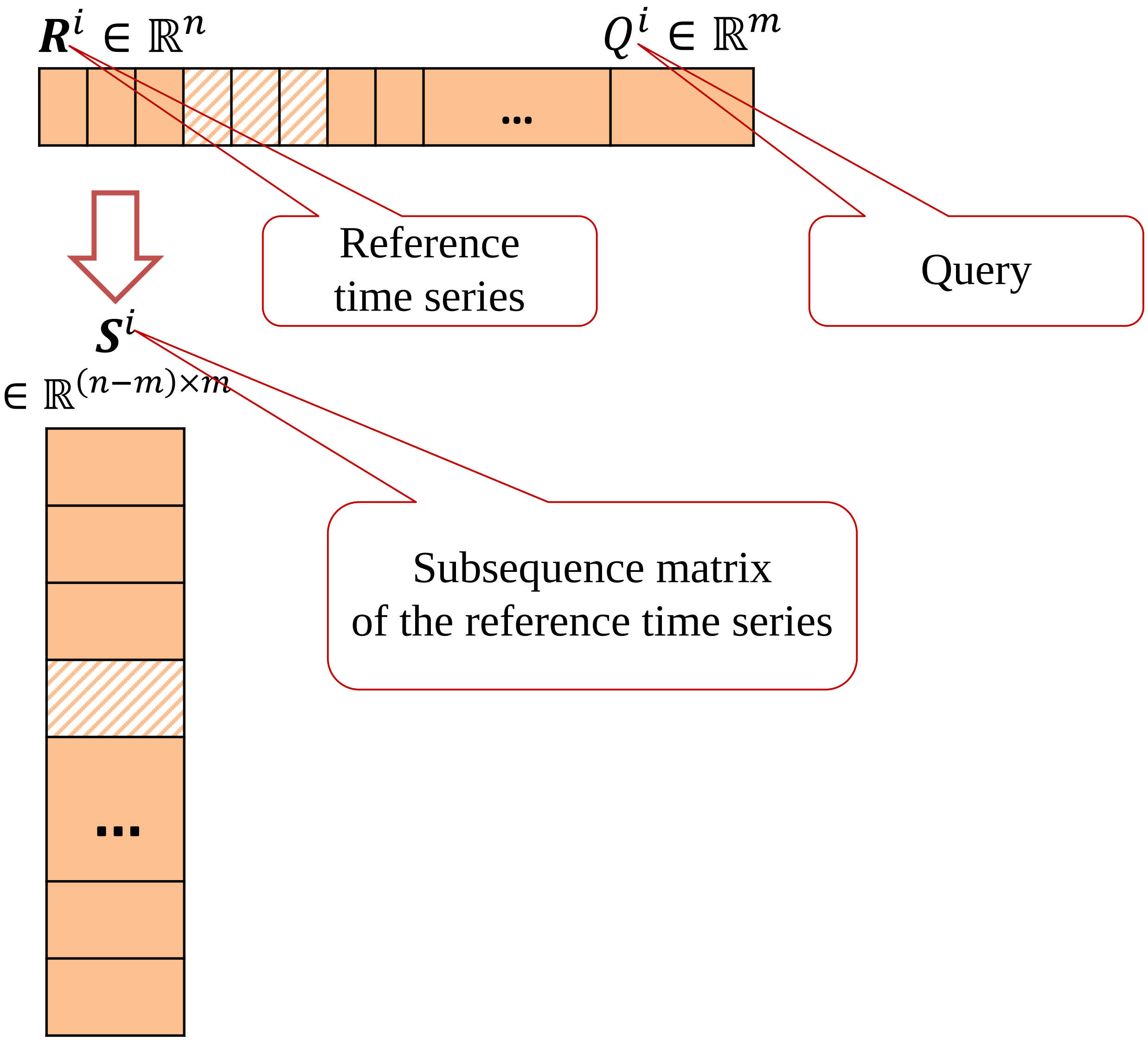


…

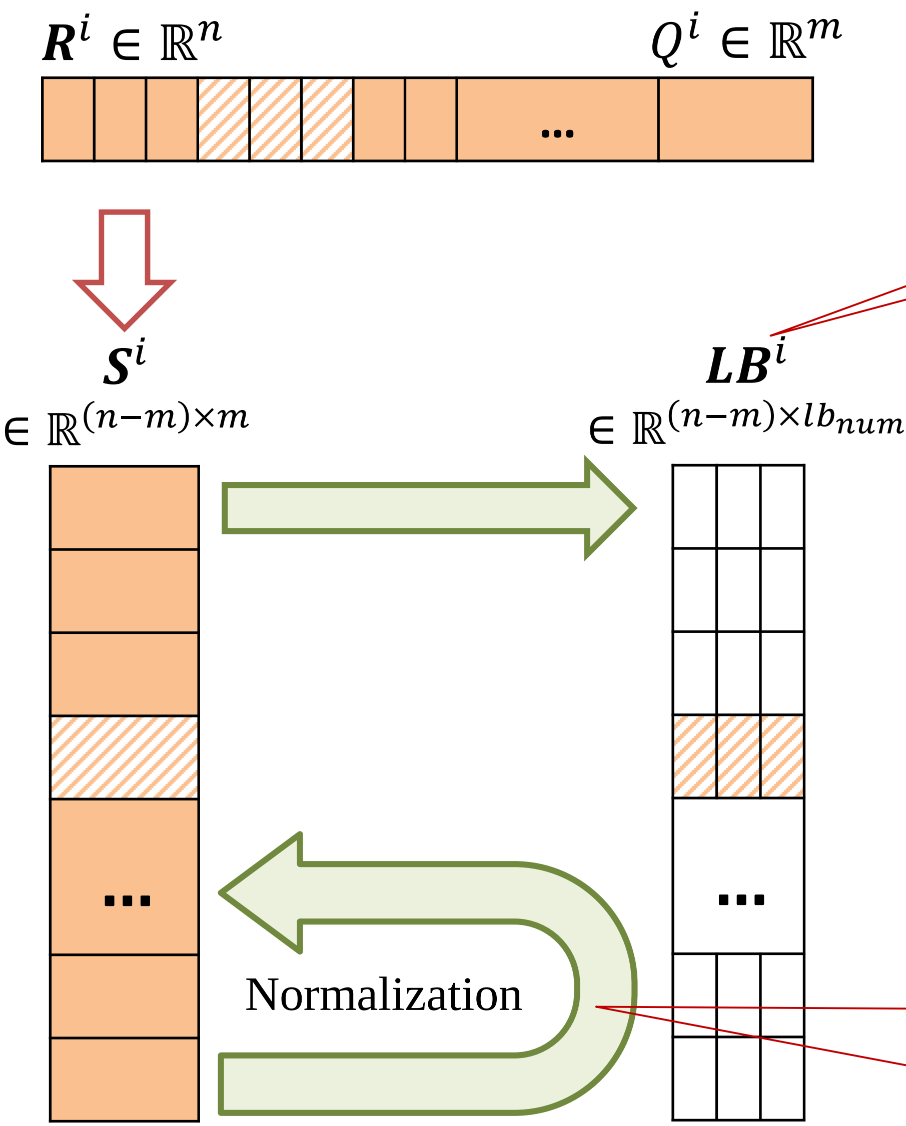
$$bsf := \min(\mathbf{DTW}(Q,C), bsf_{prev})$$

Lower bound	Complexity
LB_KimFL $\boldsymbol{LB1}(Q,C) = (q_1 - c_1)^2 + (q_m - c_m)^2$	$O(1)$
LB_KeoghEC $\boldsymbol{LB2}(Q,C) = \sum_{i=1}^{m} \begin{cases} (c_i - u_i)^2, & c_i > u_i \\ (c_i - \ell_i)^2, & c_i < \ell_i \\ 0, & otherwise \end{cases}$ $u_i = \max_{i-r \le k \le i+r} q_k$ $\ell_i = \min_{i-r \le k \le i+r} q_k$	$O(m)$
LB_KeoghEQ $\boldsymbol{LB3}(Q,C) = \boldsymbol{LB2}(C,Q)$	$O(m)$
LB_Yi, *LB_PAA*, *LB_FTW*, …	…
$\mathbf{DTW}(Q,C)$	$O(mr)$

Parallel imputation: Pattern search



Parallel imputation: Lower bounds



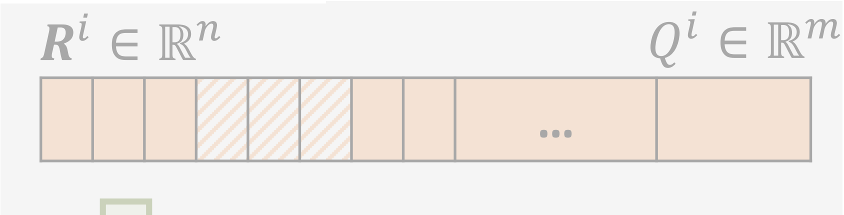
Matrix
of lower bounds

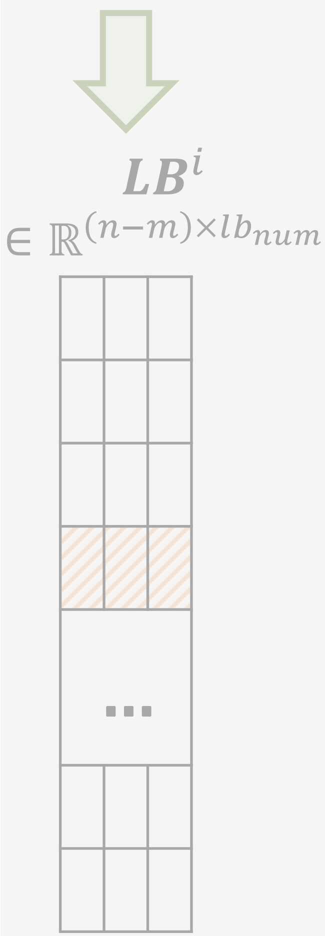
Ahead calculation of all LBs
for all subsequences
is significantly **redundant**
(cf. LB cascade in serial search)
but
it is **parallel**

$$R[1:m] \Rightarrow \hat{R}\,[1:m], \quad \hat{r}_i = \frac{r_i-\mu}{\sigma},$$

$$\mu = \frac{1}{m}\sum_{i=1}^{m} r_i\,, \quad \sigma^2 = \frac{1}{m}\sum_{i=1}^{m} r_i^2 - \mu^2$$

Parallel imputation: Initializing bsf

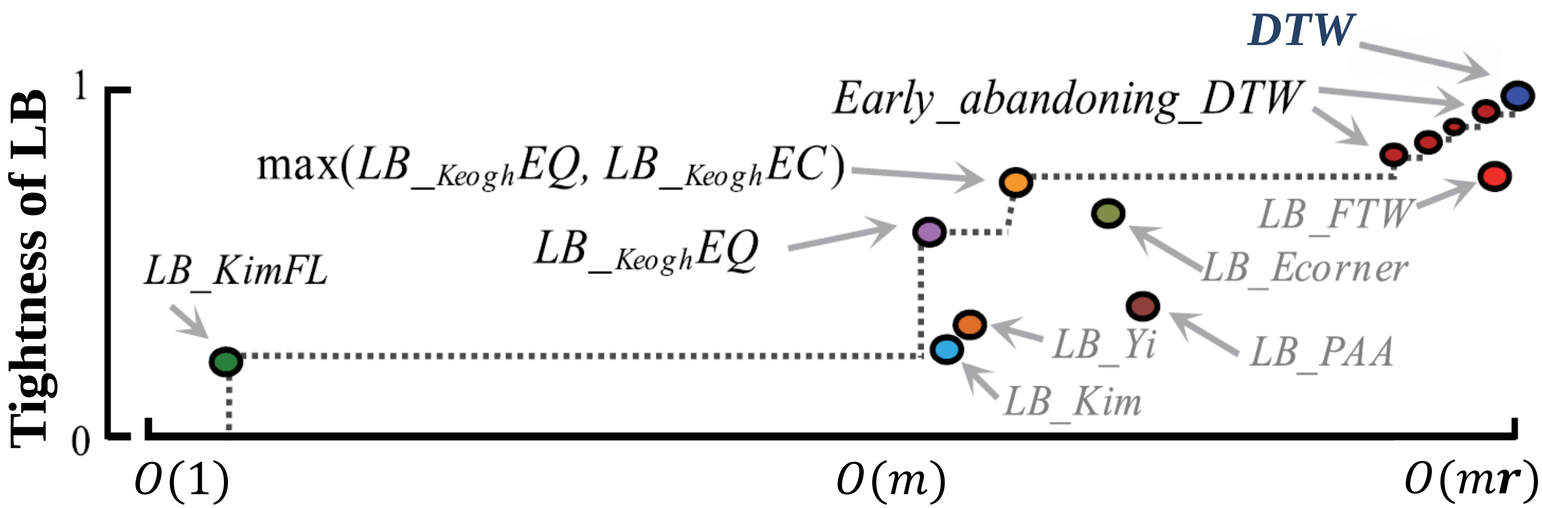




$$bsf_{init} := \mathrm{DTW}(Q^i, C)$$

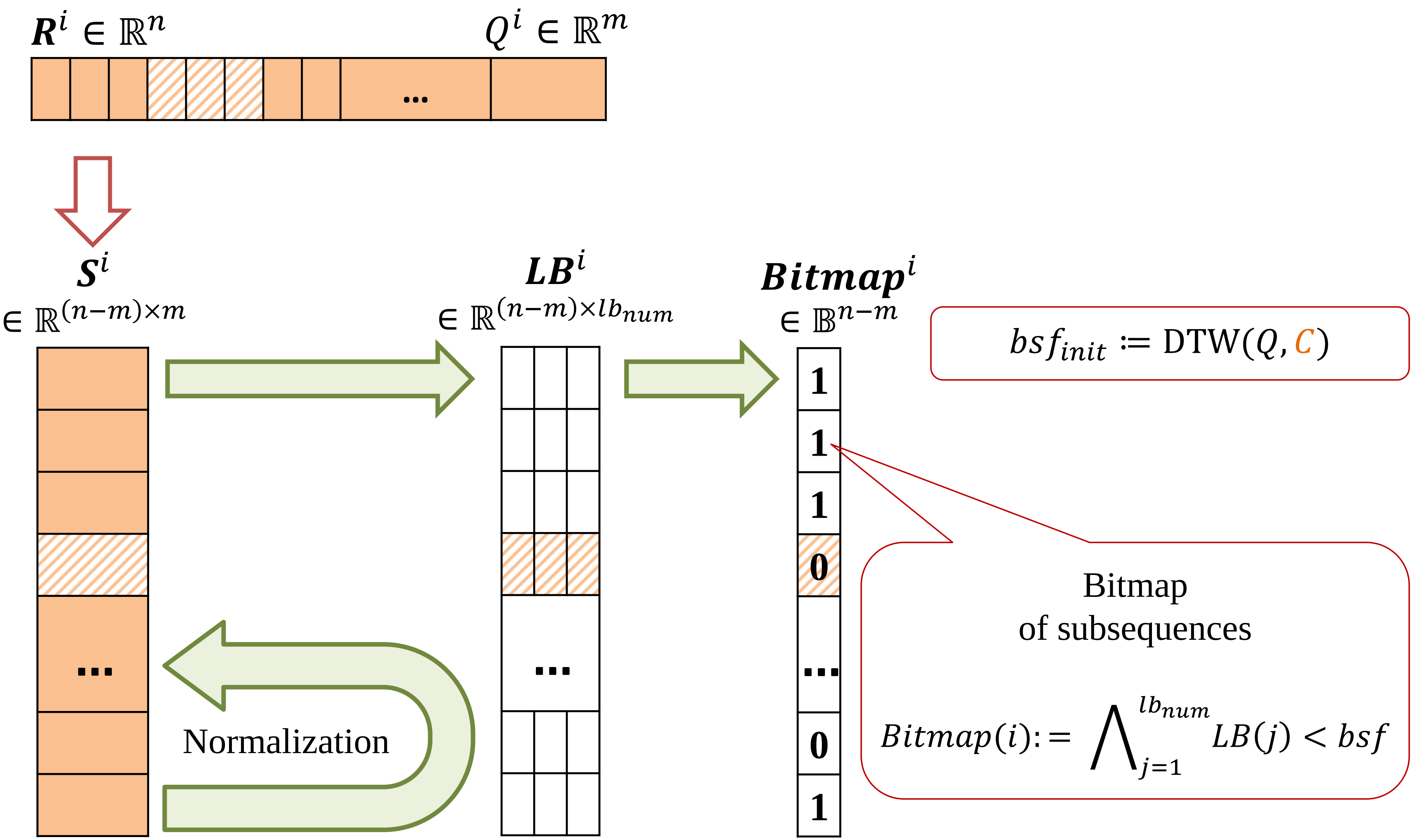
$$C = \arg \min_{1 \le p \le n-m} \max_{1 \le j \le lb_{num}} \boldsymbol{LB}_j^i(Q^i, \boldsymbol{R}[p:m])$$

$$Tightness^* = \frac{\boldsymbol{LB}(C,Q)}{\boldsymbol{DTW}(C,Q)}$$

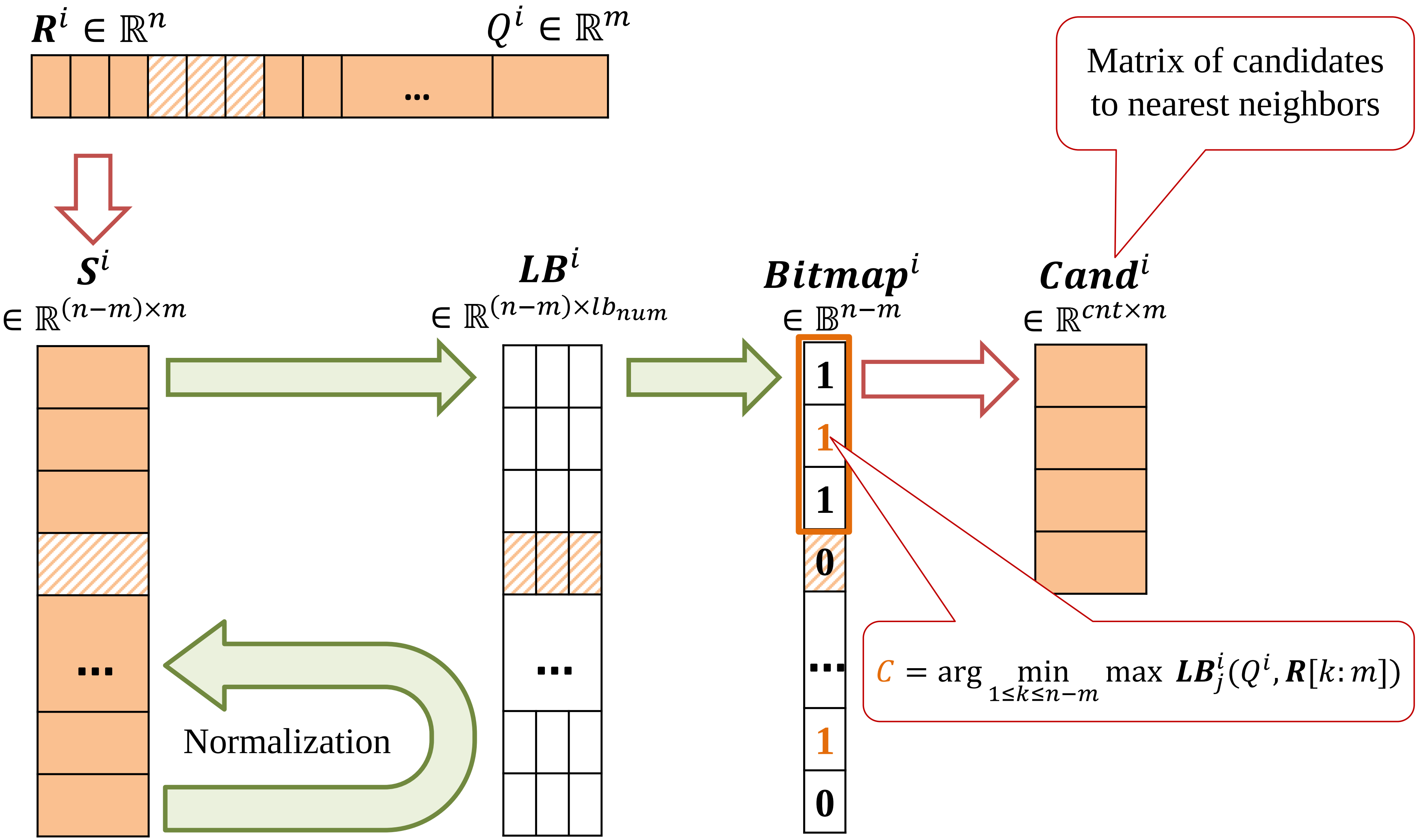


* Rakthanmanon T., *et al.* Addressing big data time series: Mining trillions of time series subsequences under Dynamic Time Warping. ACM Trans. Knowl. Discov. Data. 2013. Vol. 7, no. 3. 10:110:31. DOI: 10.1145/2500489.

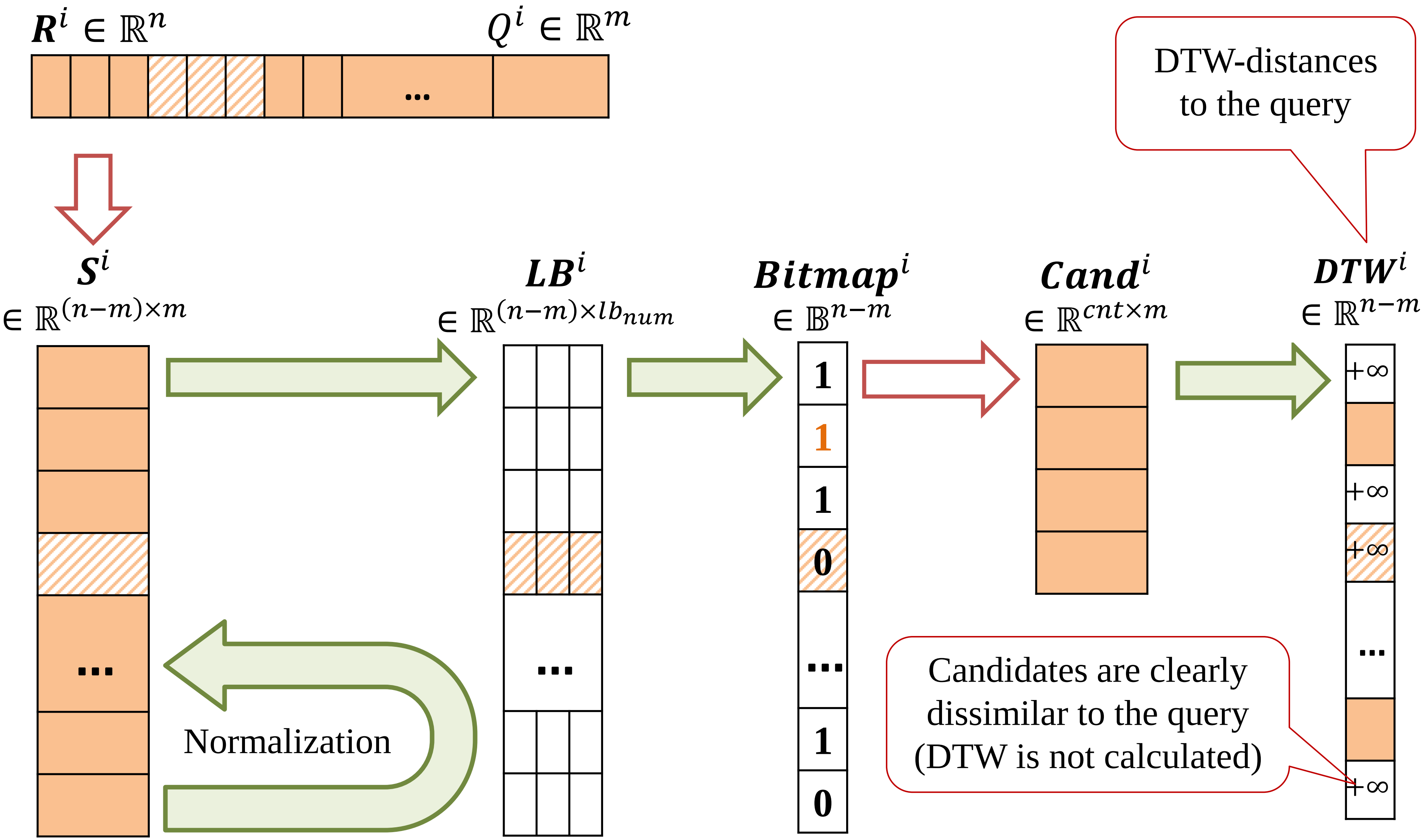
Parallel imputation: Lower bounding



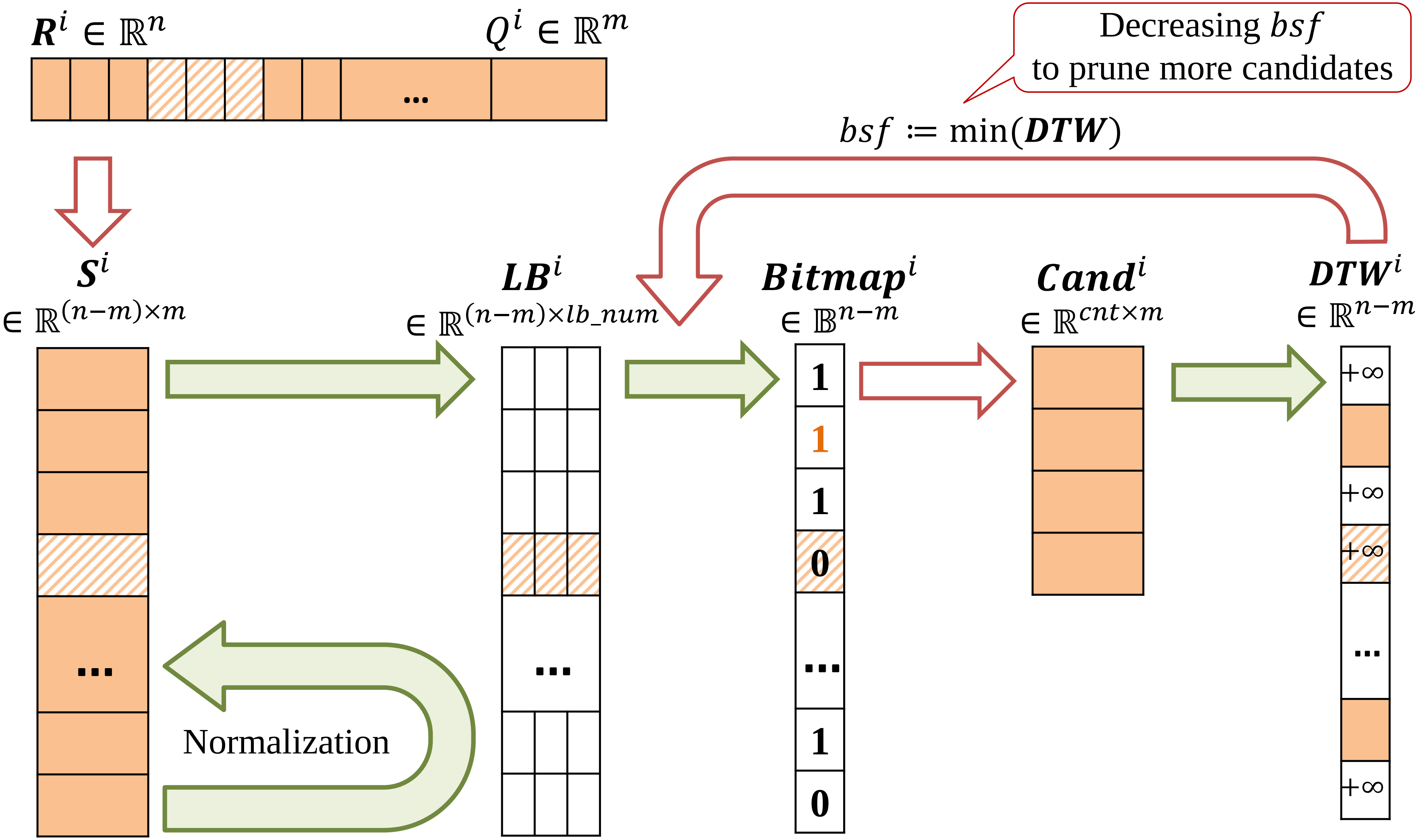
Parallel imputation: Candidate matrix



Parallel imputation: DTW calculation



Parallel imputation: Improving bsf

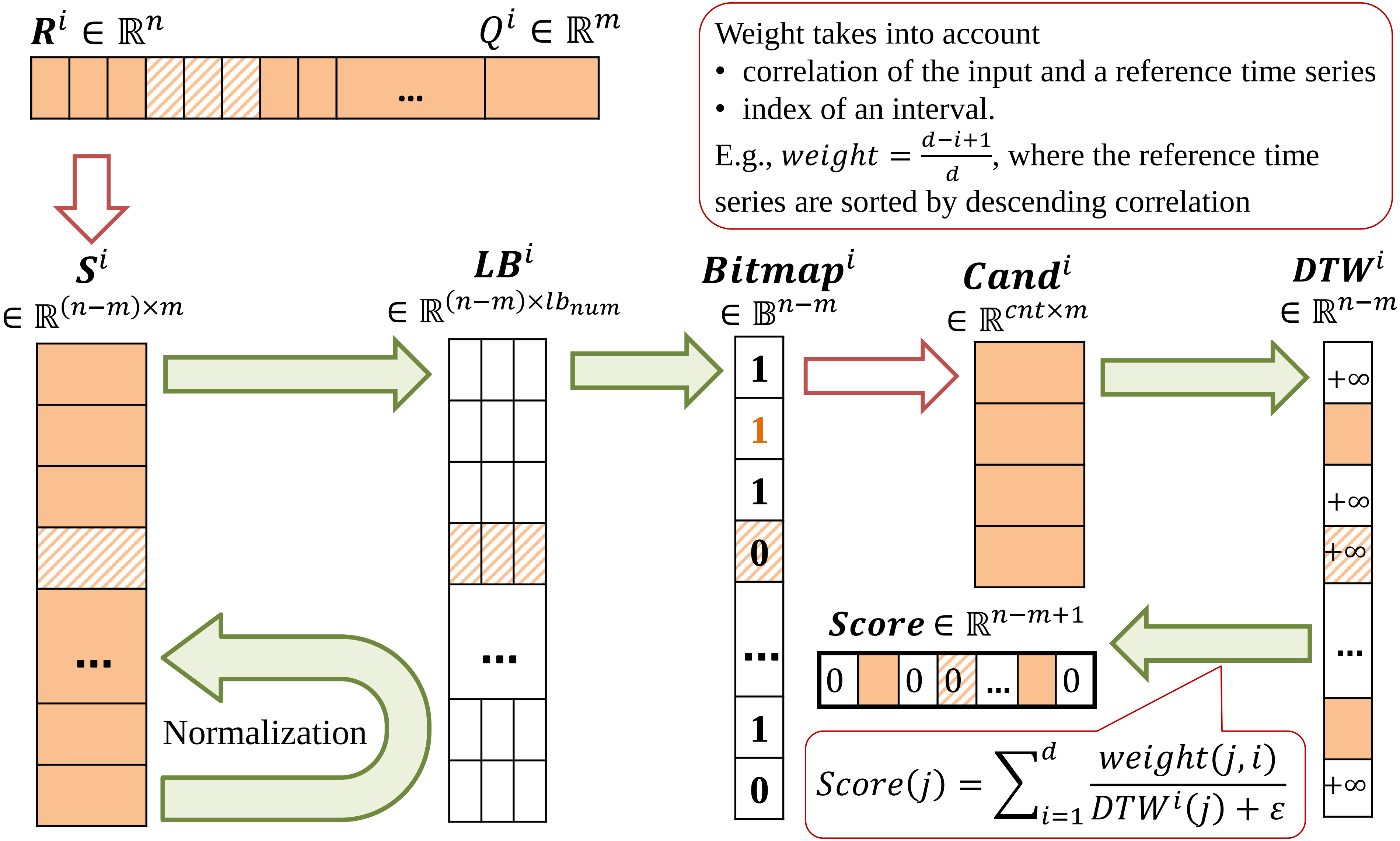


Parallel imputation: Scoring

Weight takes into account

- correlation of the input and a reference time series
- index of an interval.

E.g., $weight = \frac{d-i+1}{d}$, where the reference time series are sorted by descending correlation



Parallel imputation: Experiments

- **Hardware**: Intel Xeon E5-2687W v2 (8 cores @3.40 GHz)
- **Data**

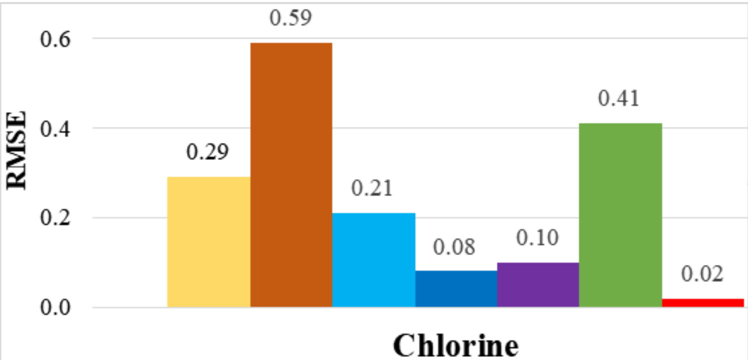
Dataset	# t.s., $d+1$	Length, $n \cdot 10^3$	Domain
BAFU	10	50	Water discharge in Swiss rivers
Chlorine	50	1	Simulation of the chlorine concentration in a drinking water system
Climate	10	5	Weather in locations of North America
MADRID	10	25	Road traffic (AVR statistics) in Madrid
MAREL	10	50	Characteristics of sea water in the English Channel

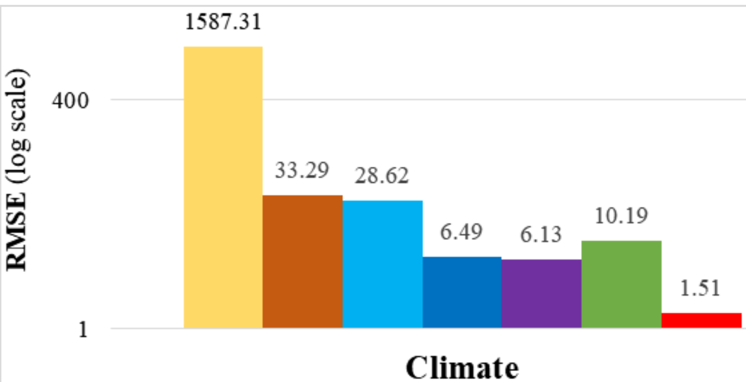
- **Rivals**: ORBITS[1], OGD-Impute[2], SPIRIT[3], SAGE[4], TKCM[5]
- **Setup** under ORBITS[1] framework:
  - Scenario: imputation of last 10% points
  - Rivals: best recommended parameters
  - ParaDI: $m = 50, k = 3, r = 0.25m$

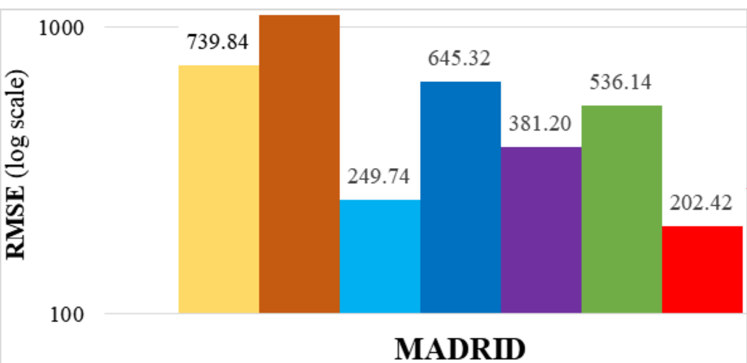
[1] Khayati M., *et al.* ORBITS: Online Recovery of Missing Values in Multiple Time Series Streams. Proc. VLDB Endow. 2020. Vol. 14, no. 3. P. 294-306.
[2] Anava O., *et al.* Online Time Series Prediction with Missing Data. Proc. ICML 2015. P. 2191-2199.
[3] Papadimitriou S., *et al.* Streaming Pattern Discovery in Multiple Time-Series. VLDB 2005. P. 697-708.
[4] Balzano L., *et al.* Streaming PCA and Subspace Tracking: The Missing Data Case. Proc. of IEEE. 2018. Vol. 106, no. 8. P. 1293-1310.
[5] Wellenzohn K., *et al.* Continuous Imputation of Missing Values in Streams of Pattern-Determining Time Series. EDBT 2017. P. 330-341.

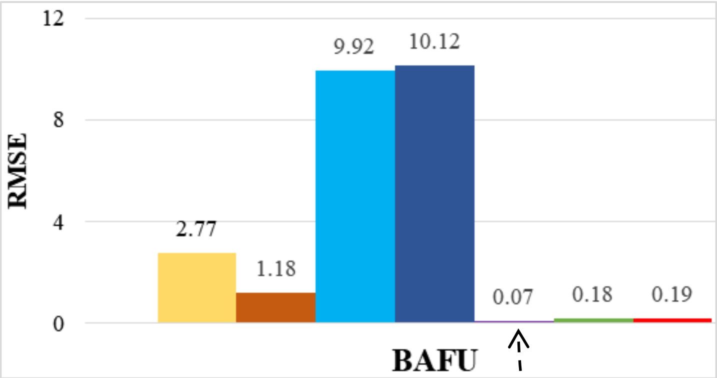
Experiments: accuracy

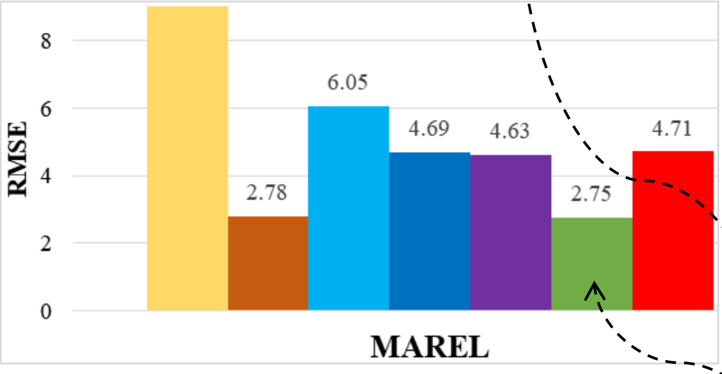
$$\text{RMSE} = \sqrt{\frac{1}{h}\sum_{i=1}^{h}(t_i - \tilde{t}_i)^2}$$











Parallel algorithm is **ahead** w.r.t. accuracy of many (but **not all**) rivals

SAGE OGD-Impute ORBITS (k=2) ORBITS (k=3)
SPIRIT TKCM **Our algorithm**

Outline

- Introduction
- Parallel pattern discovery
- Parallel anomaly detection
- Parallel imputation of missing values
- **Online time series analytics with parallel algorithms**
  - Online anomaly discovery
  - Online imputation of missing values

Mining time series: parallel algs *vs.* ANNs

Method	Labeling ⇨	Learning ⇨	Tuning ⇨	Mining
Parallel algorithms	 "Another brick in the wall" by Pink Floyd		 Turtle **15-35 km/h,** running/swimming	 Cheetah **100 km/h**

Method	Labeling ⇨	Learning ⇨	Tuning ⇨	Mining
Artificial Neural Networks	 Yusuke Suzuki (racewalker) **15.6 km/h**	 Turtle **15-35 km/h,** running/swimming	 Turtle **15-35 km/h,** running/swimming	 Spirostomum ambiguum **724 km/h**

Mining time series: parallel algs **plus** ANNs

Parallel algorithms

Labeling

Cheetah
100 km/h

Artificial Neural Networks

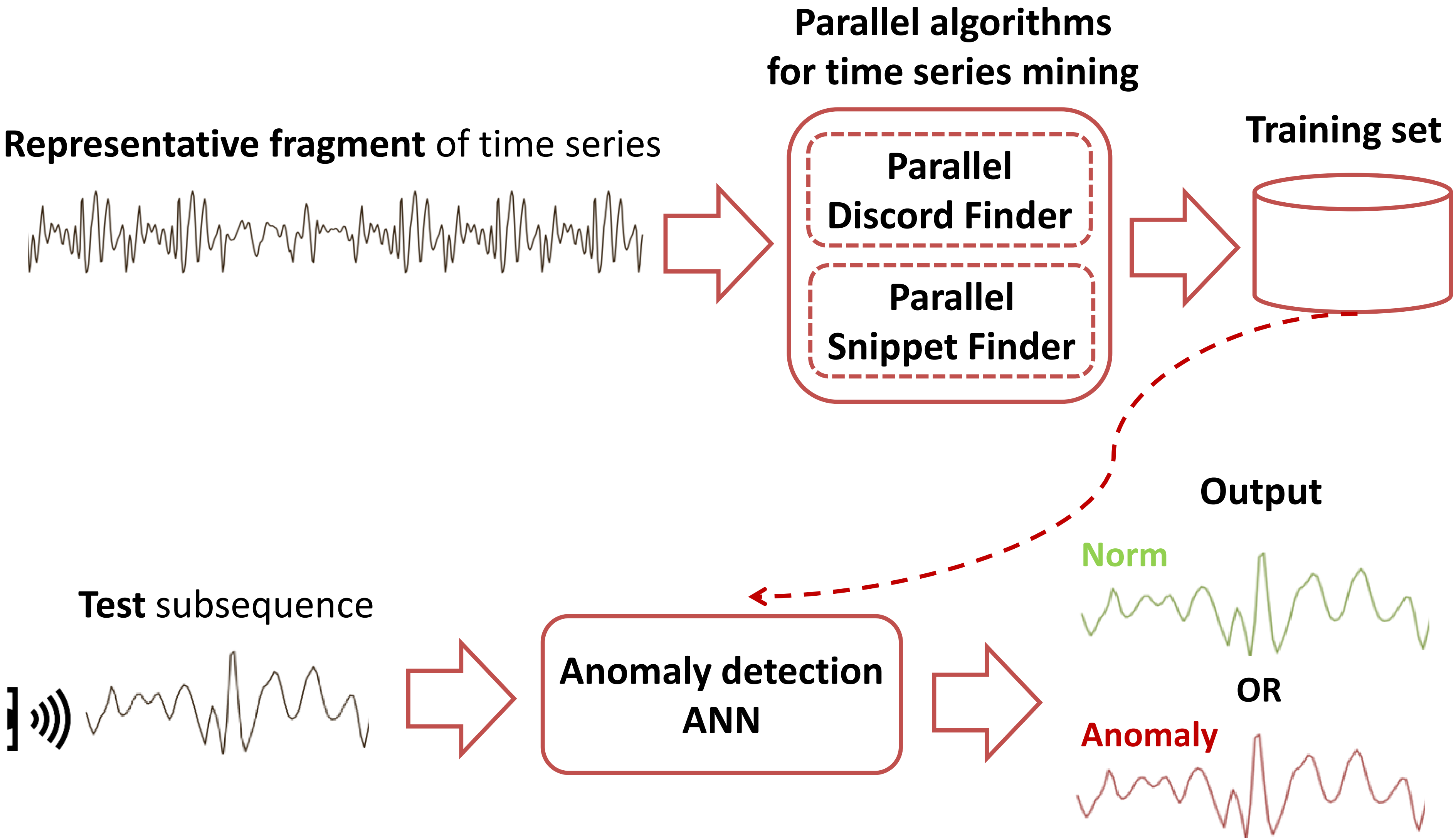
Learning ⇨ **Tuning** ⇨ **Mining**

Turtle
15-35 km/h,
running/swimming

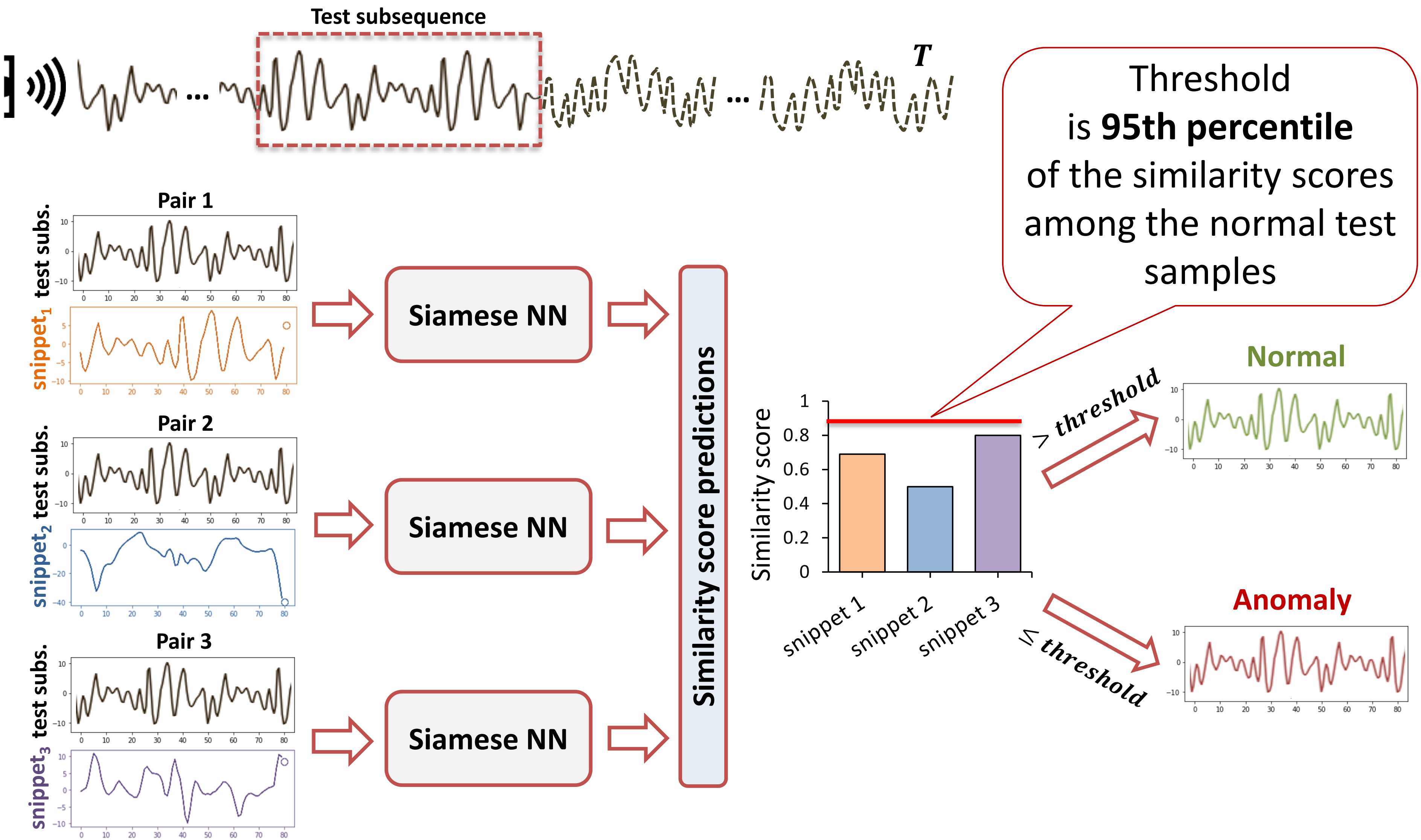
Turtle
15-35 km/h,
running/swimming

Spirostomum ambiguum
724 km/h

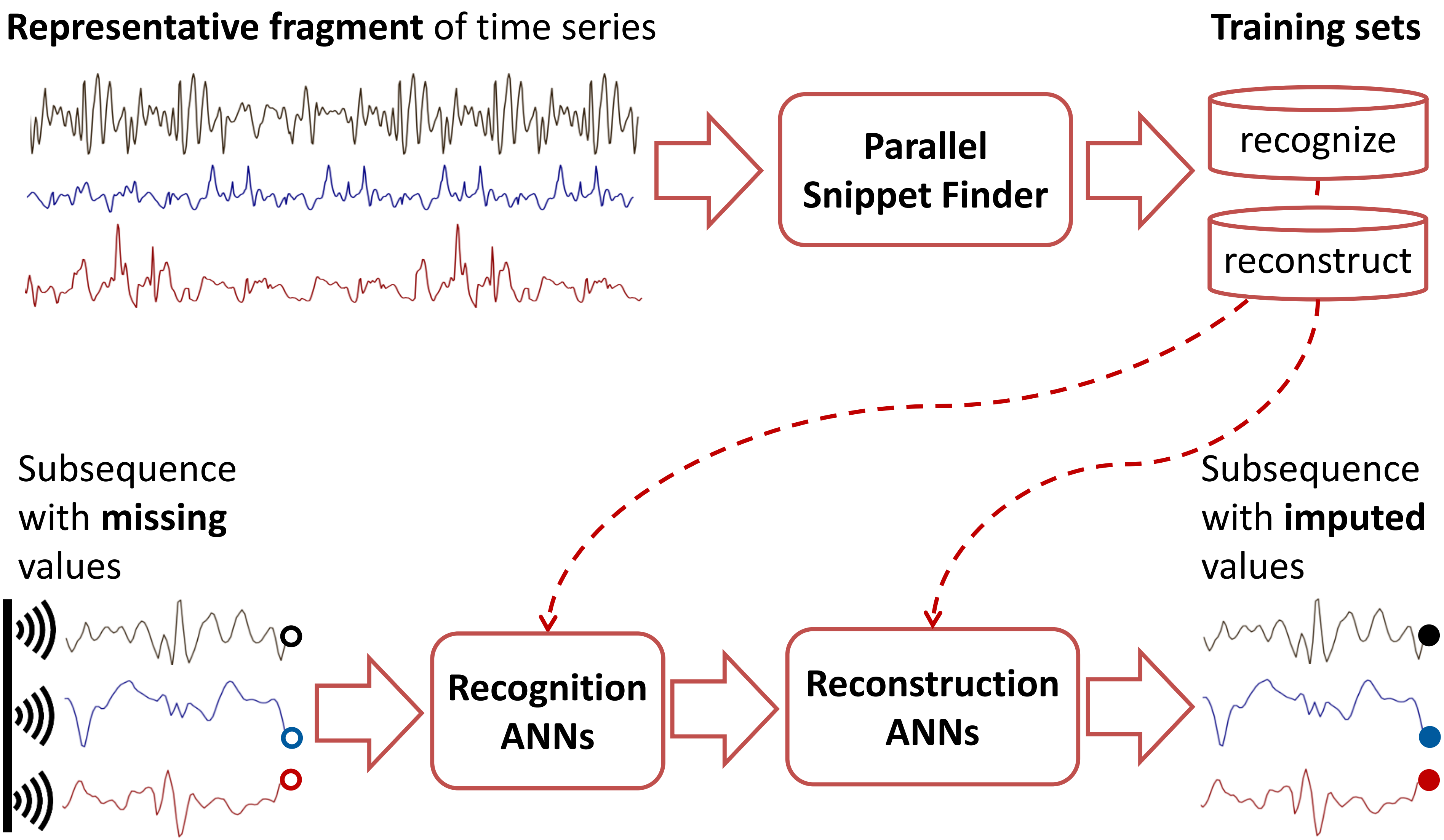
Online time series anomaly detection



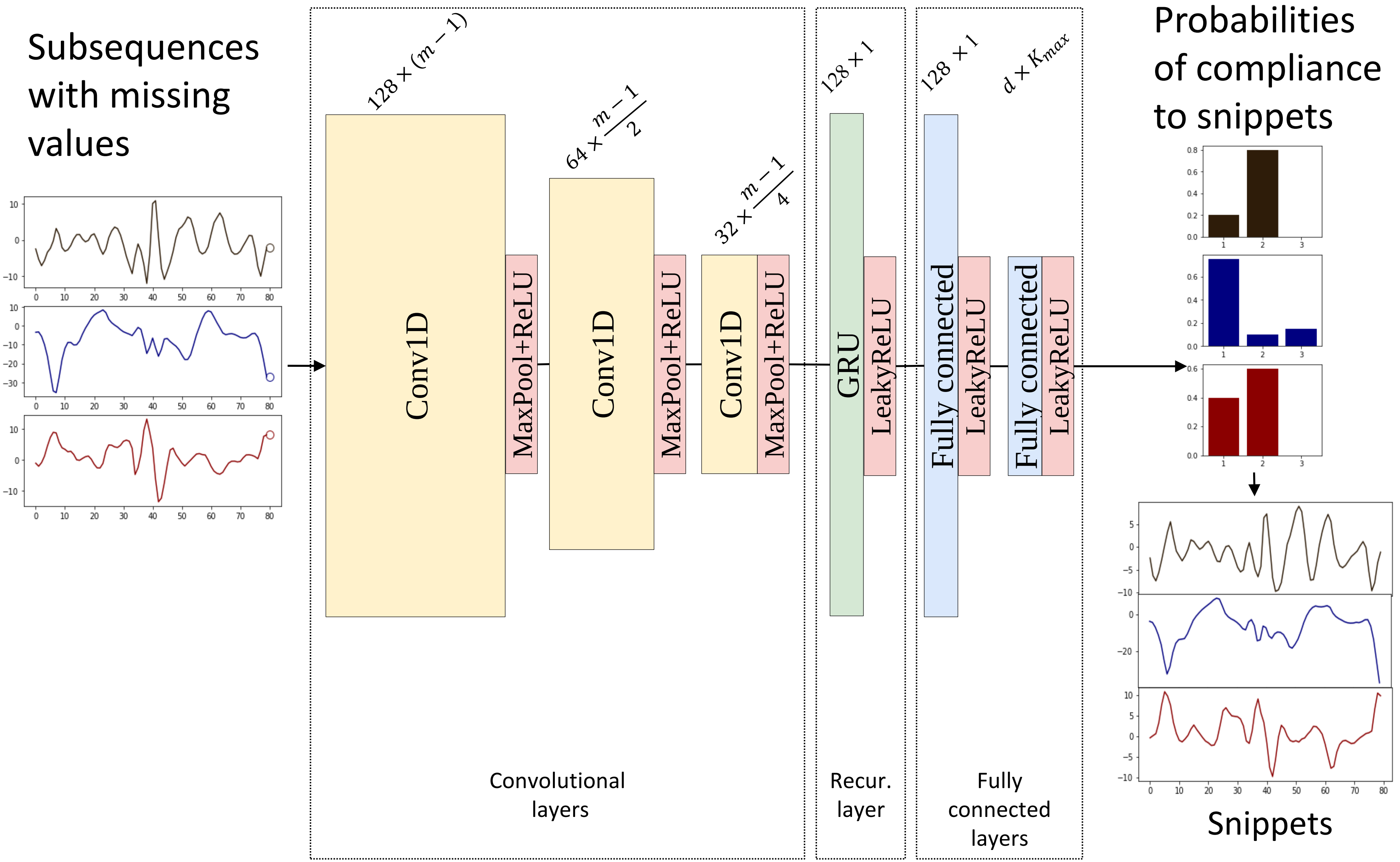
Online time series anomaly detection



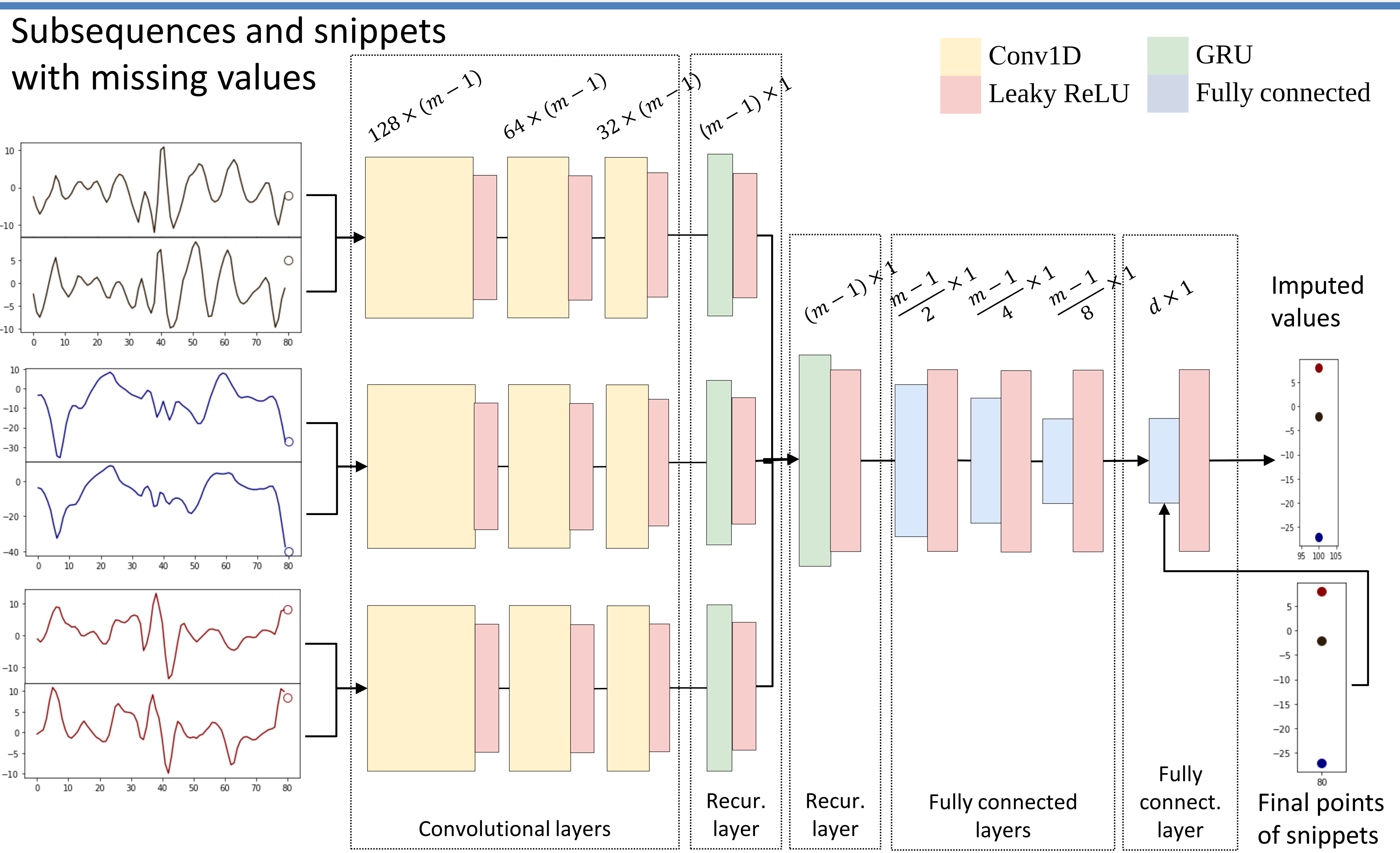
Online imputation of missing values



Online imputation: Recognizer



Online imputation: Reconstructor



Conclusions

- Parallel algorithms can significantly accelerate time series mining
- Parallel algorithms together with ANNs can provide online time series mining

Thank you for paying attention! Questions?

Mikhail Zymbler,
South Ural State University, Chelyabinsk, Russia
mzym@susu.ru
https://mzym.susu.ru