TIME SERIES DATA MINING

Ing. Petr Novák
novak@infoway.cz

Outline of talk

- Time series
- Similarity measures
- Time series representation
- Motif discovery
- TSMiner

Time series

What is time series…

- Popular statistical definition

  „The time series is a sequence of values of a statistical character (indicators) arranged in terms of time away from the past to the present. The indicator changes over time, the overall development can be divided into three components - trend, periodic and random.“

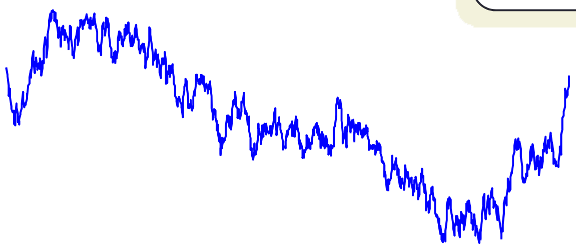
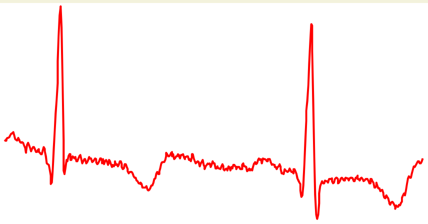
- **Is it the only option?**

Time series

People measure things…

- *Blood pressure*
- *Popularity of politicians*
- *Annual rainfall*
- *Stock value*

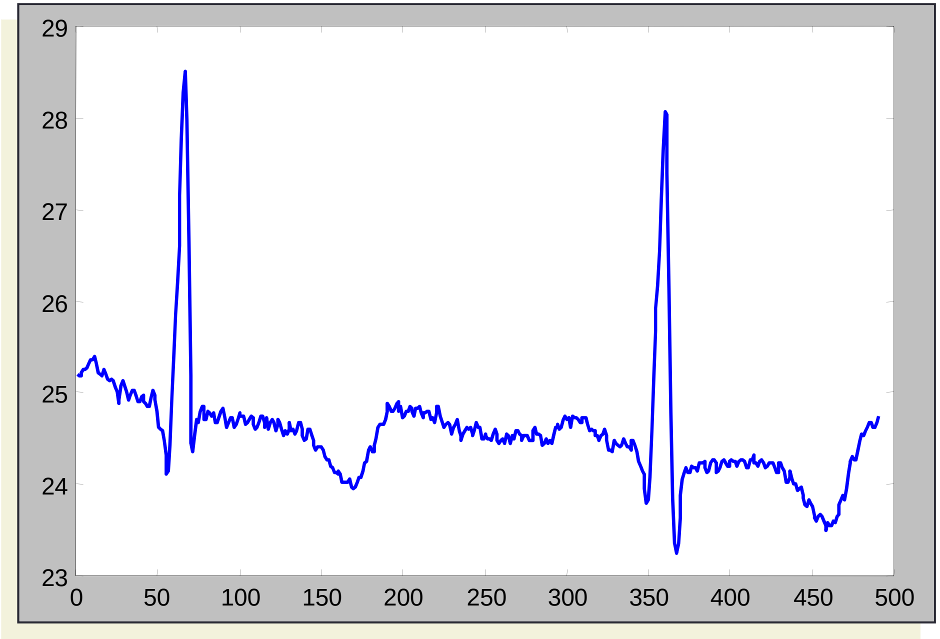
…and things change over time.


Time series

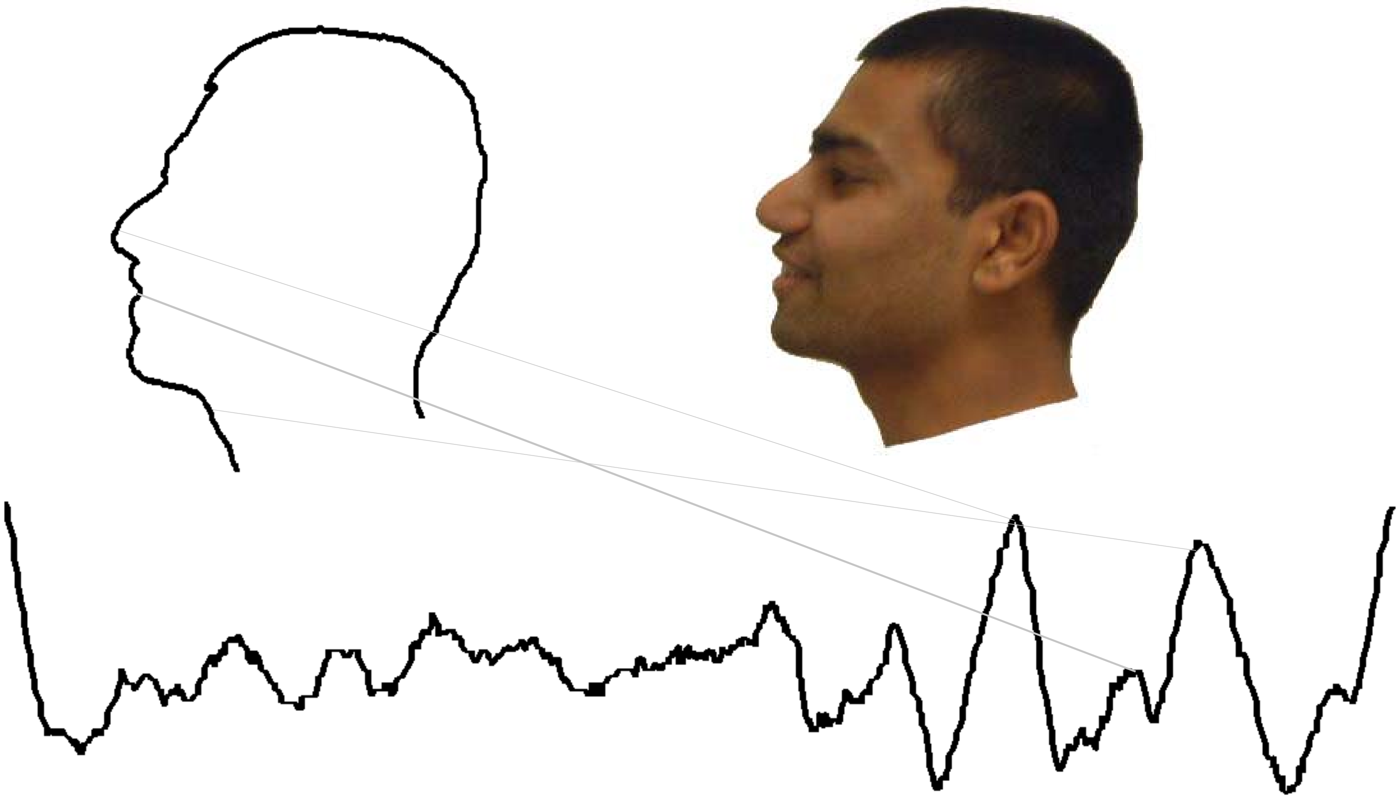
What is time series…

- „Time series is a collection of observations made sequentially in time.“



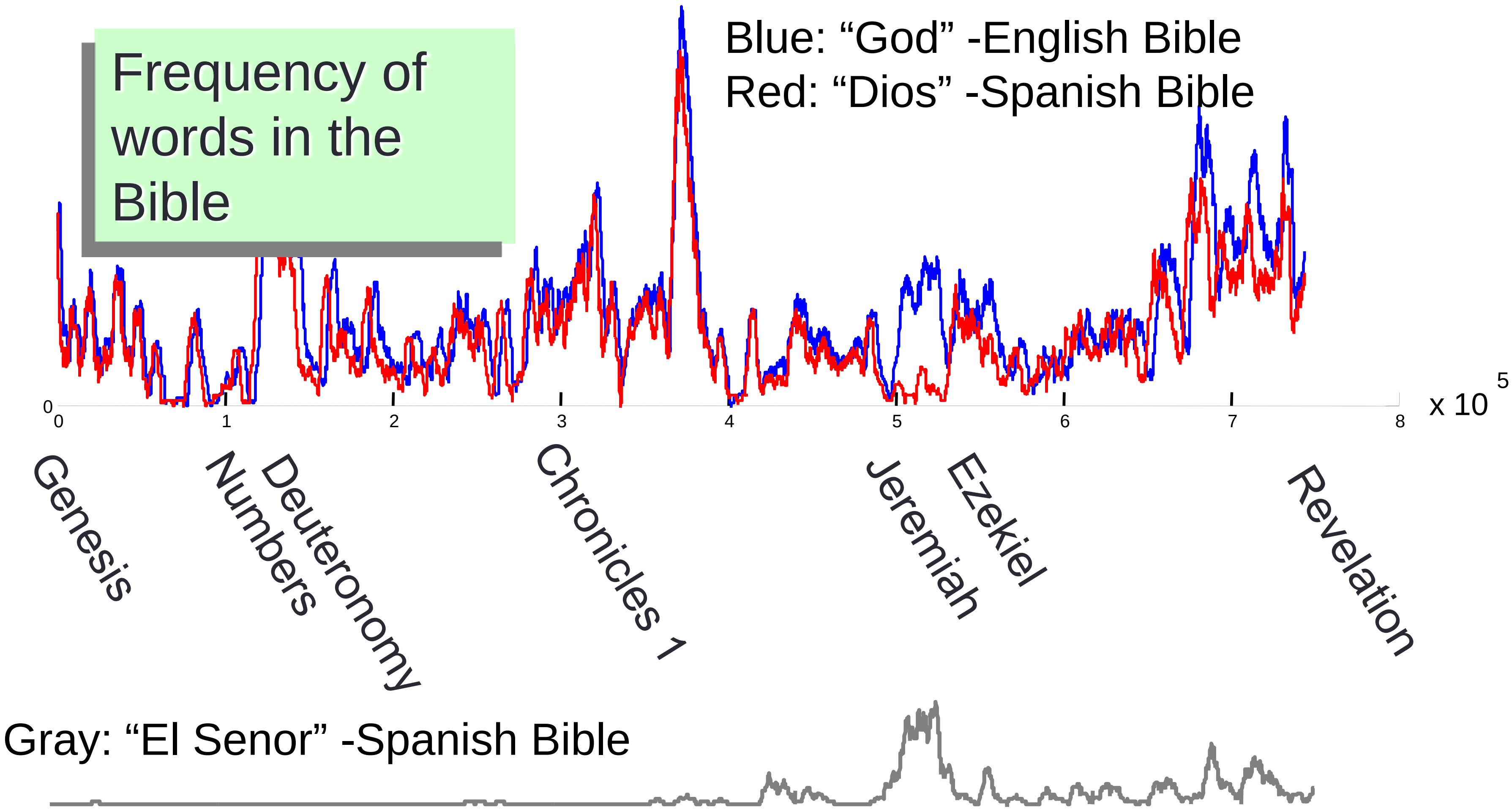
Time series

- Image data, may be thought of as time series



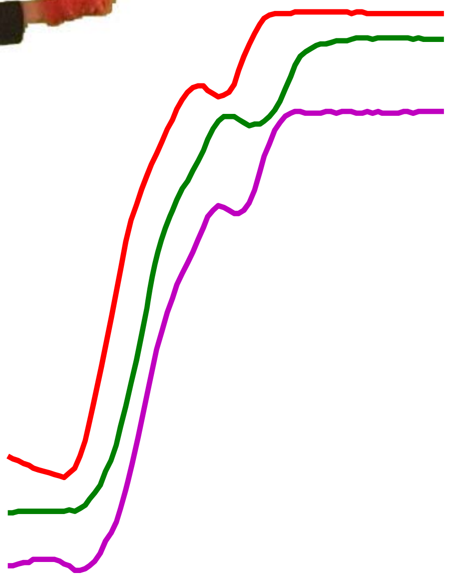
Time series

- Text data, may best be thought of as time series…



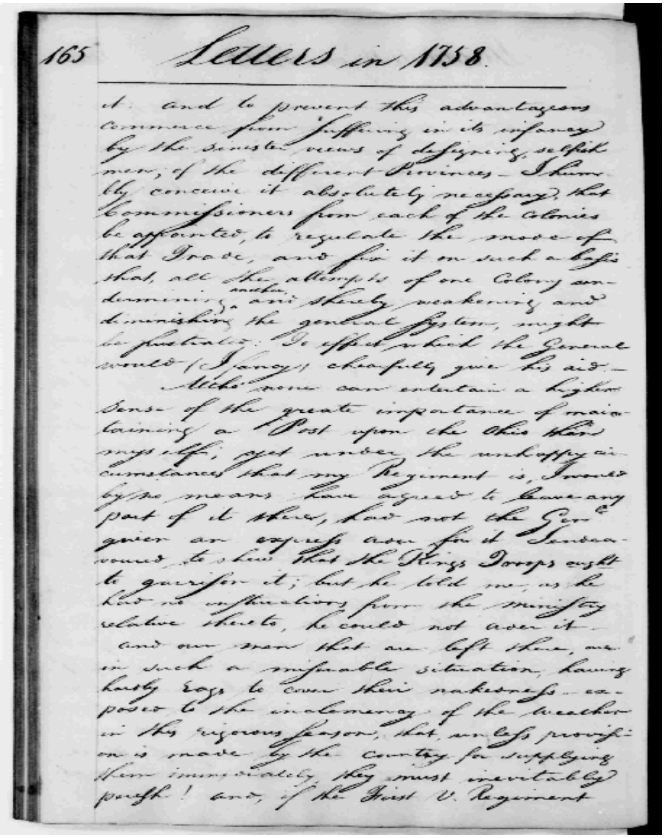
Time series

- Video data, may be thought of as time series



Time series

- Handwriting data, may be thought of as time series



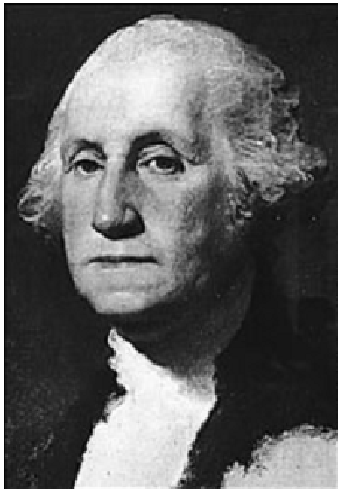
165 Letters in 1758.

it. and to prevent this advantageous commerce from suffering in its infancy by the sinister views of designing, selfish men, of the different Provinces — I humbly conceive it absolutely necessary, that Commissioners from each of the Colonies be appointed, to regulate the mode of that Trade, and fix it on such a basis that, all the attempts of one Colony undermining another and thereby weakening and diminishing the general System, might be frustrated. To effect which the General would (I fancy) chearfully give his aid. —

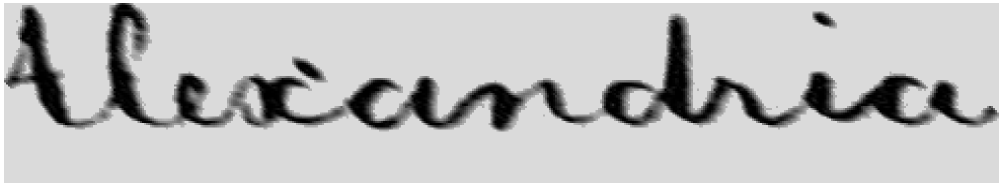
Altho' none can entertain a higher Sense of the greate importance of maintaining a Post upon the Ohio than myself; yet under the unhappy circumstances that my Regiment is, I would by no means have agreed to leave any part of it there, had not the Gen^l given an express order for it. I endeavoured to shew that the Kings Troops ought to garrison it; but he told me, as he had no instructions from the ministry relative thereto, he could not order it.

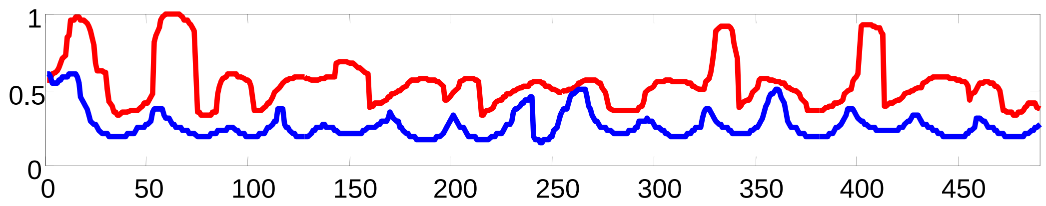
And our men that are left there, are in such a miserable situation, having hardly rags to cover their nakedness — exposed to the inclemency of the weather in this rigorous season, that, unless provision is made by the country for supplying them immediately they must inevitably perish! and, if the First V. Regiment

G. Washington manuscript



George Washington
1732-1799

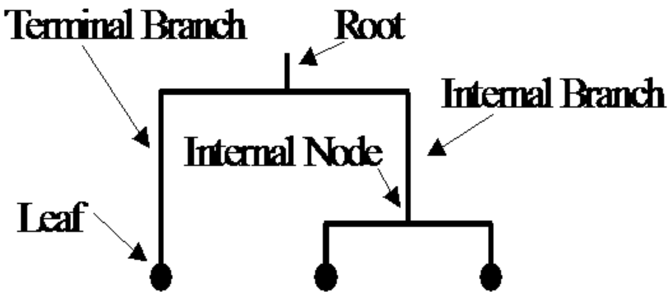


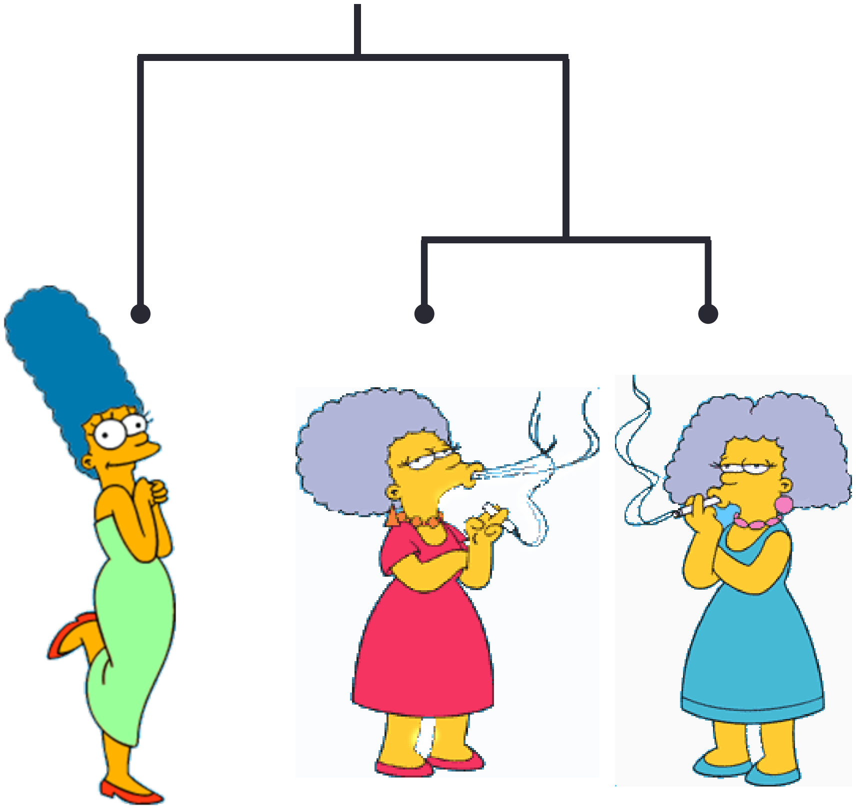


Small digression - dendrogram

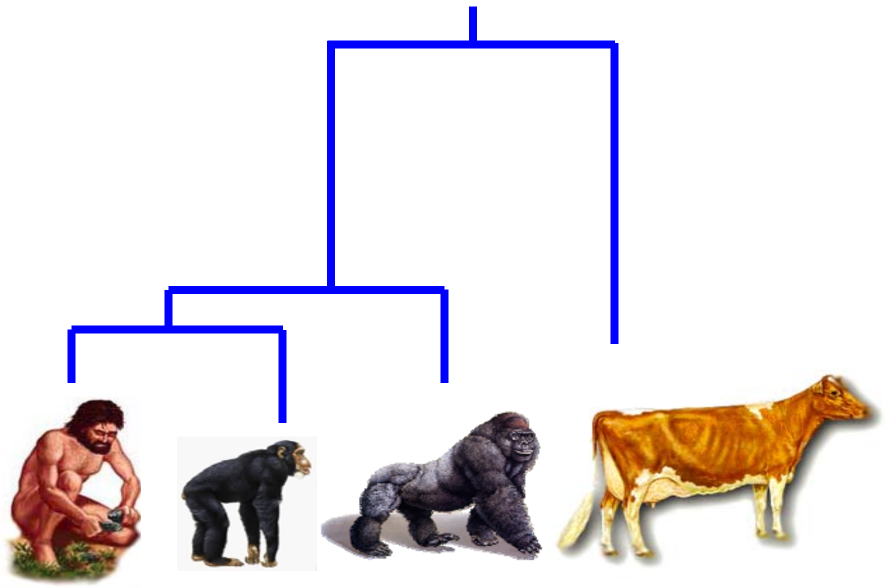
- A Useful Tool for Summarizing <u>Similarity Measurements</u>

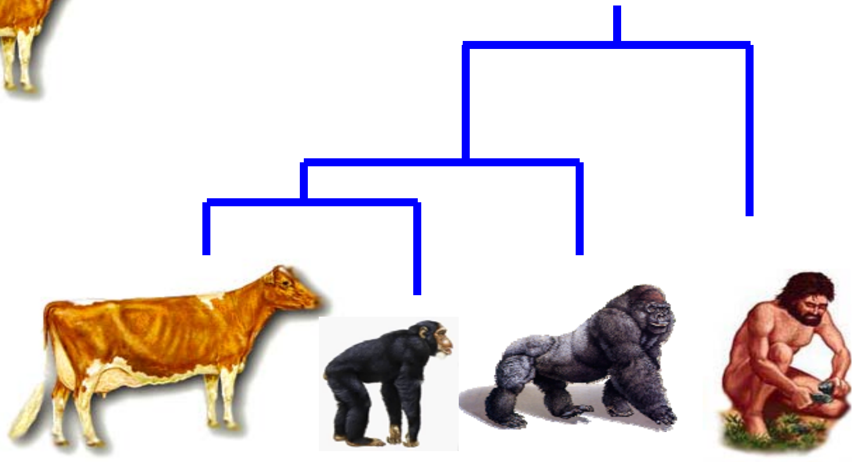
The similarity between two objects in a dendrogram is represented as the height of the lowest internal node they share.





Small digression - dendrogram



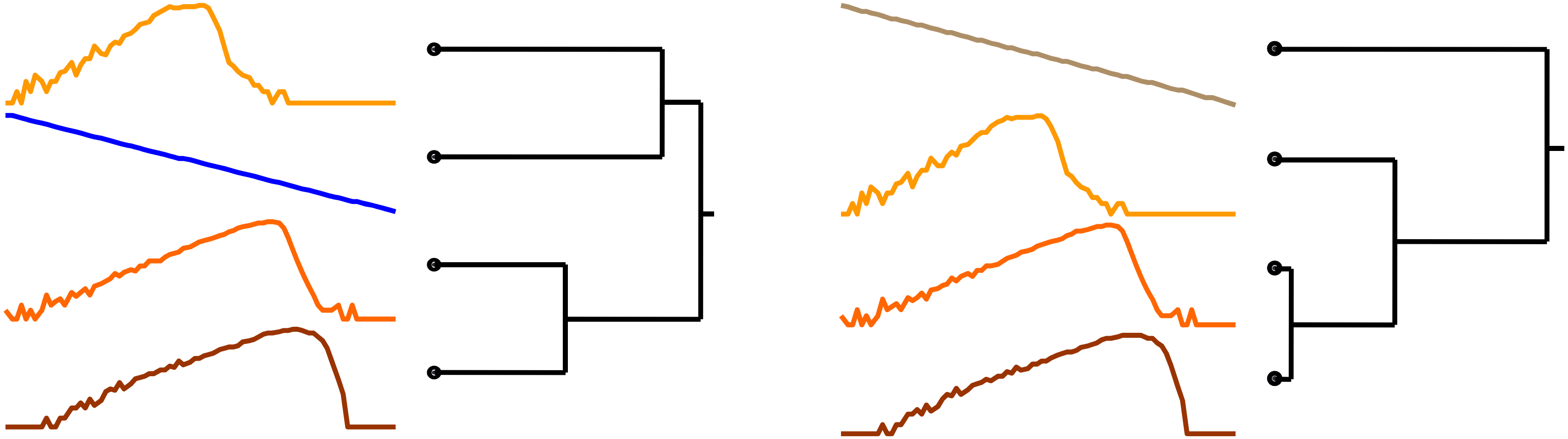


Time series

- Why is working with time series so difficult?
  - 1 hour of EKG – 1 gigabyte.
  - Space shuttle telemetry – 20 000 sensors send data every second, hundreds of GB per regular mission.
  - Database Macho – astronomical observations 20 millions stars, 3 terabytes per day.
  - 300 millions phone calls in AT&T network every day between 100 millions of customers.
  - 50 000 stock titles in USA, 100 000 trades per second.
- **<u>We need a data representation for efficient processing</u>**

Time series

- Why is working with time series so difficult?
  - We are dealing with subjectivity



- **<u>We need to measure the similarity of time series regardless of the subjective feeling</u>**

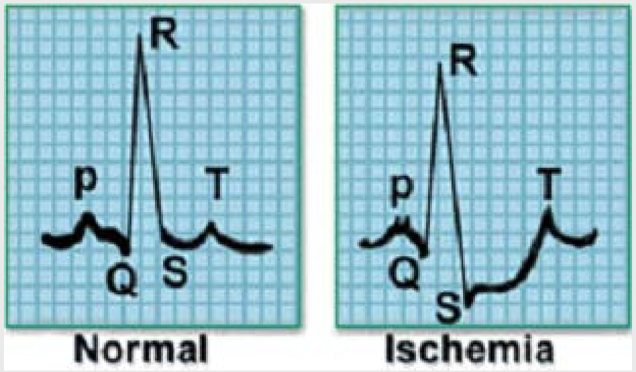
Time series

- Why is working with time series so difficult?
  - Different scales
  - Differing sampling rates
  - Noise, missing values

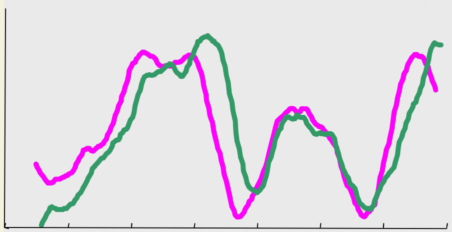
Time series

Clustering

Classifications

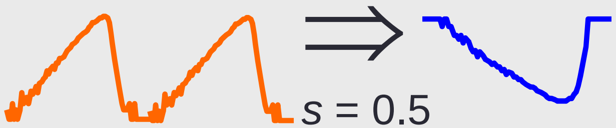


Motif discovery



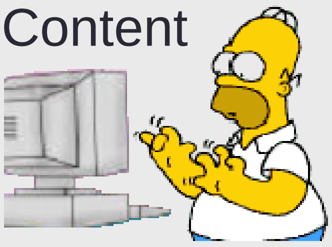
Rule discovery

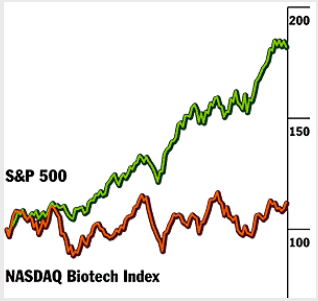
10



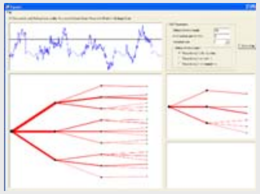
s = 0.5
c = 0.3

Query by Content





Visualization



Novelty detection

Time series

- For all these activities we need to determine the **similarity** between time series.

Time series similarity

- Similarity is hard to define, but we know it when we see it

Time series similarity

- We have to define **distance measure**

- **Definition: Let O1 and O2 are two objects from the universe of possible objects. The distance (dissimilarity) is denoted by D(O1, O2)**

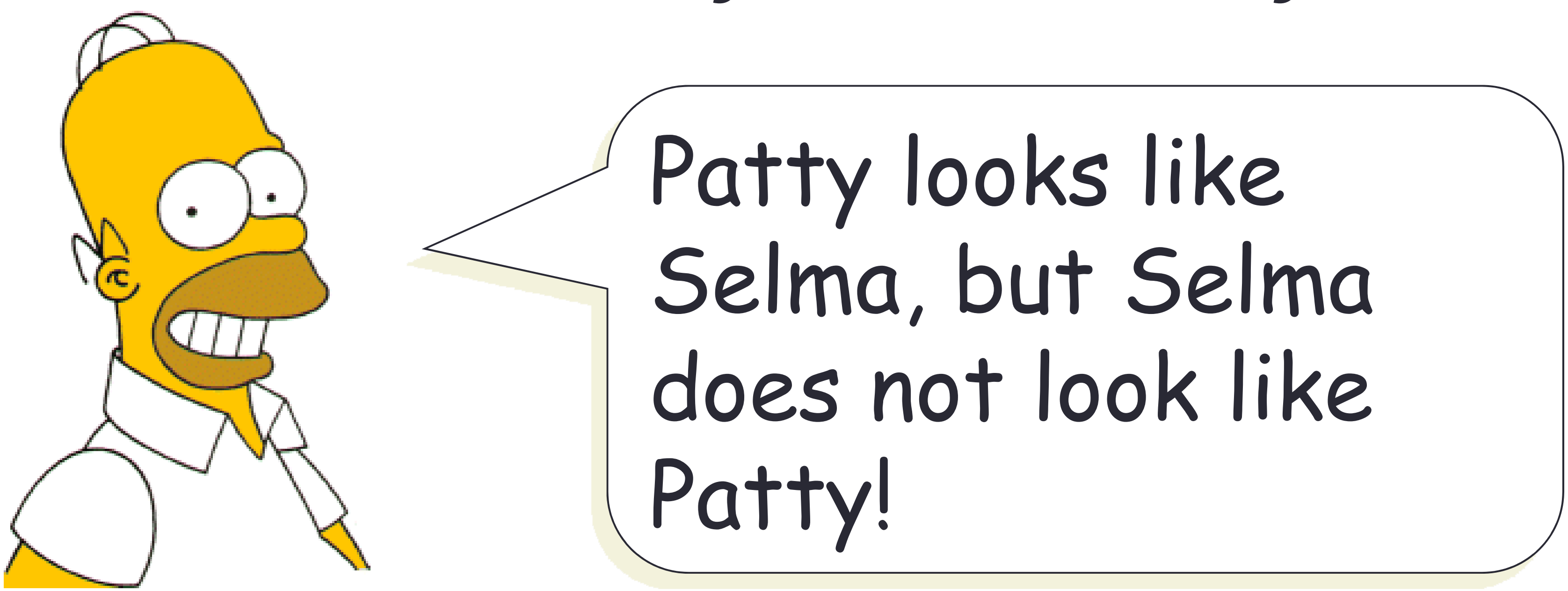
- Properties of distance measure
  - D(A, B) = D(B, A), symmetry
  - D(A, A) = 0, identity
  - D(A, B) = 0 iff A = B positivity
  - D(A, B) <= D(A, C) + D(B, C) triangular inequality

Time series similarity

D(A,B) = D(B,A) *Symmetry*

D(,) = D(,)

Otherwise you could say:



Time series similarity

D(A,A) = *0* *Identity*

D( , ) = 0

Otherwise you could say:

Marge looks more like Patty than Patty does!

Time series similarity

D(A,B) = *0 IIF A = B*   *Positivity*

D(,) = 0, IIF  = 

Otherwise you could say:

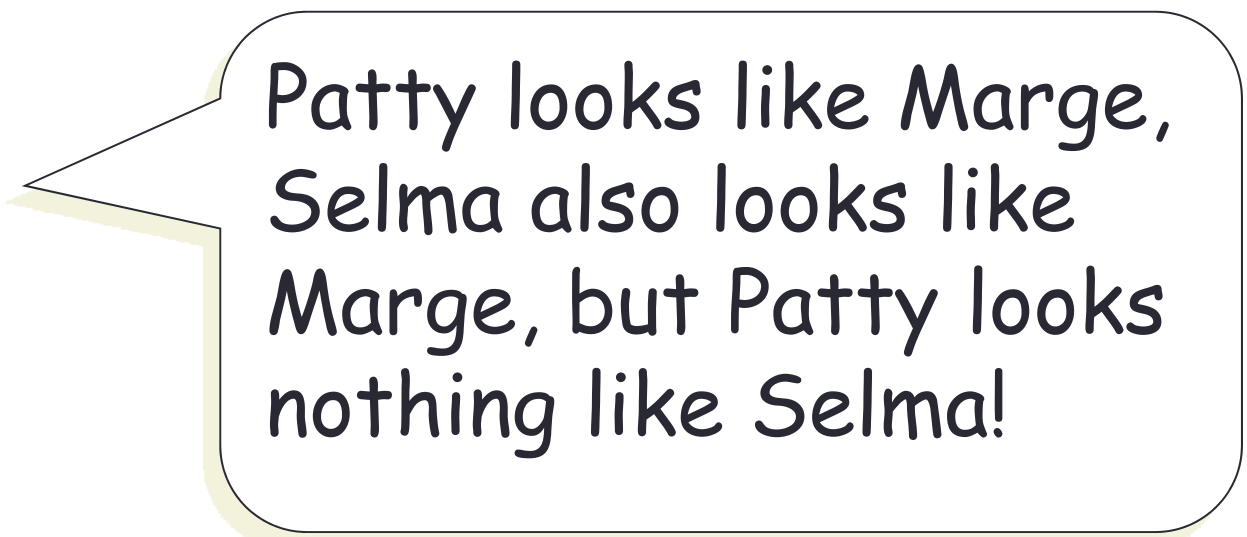
I know Patty and Marge are somehow different, but I can't tell them apart!

Time series similarity

D(A,B) ≤ D(A,C) + D(B,C) *Triangular inequality*

D( , ) ≤ D( , ) + D( , )

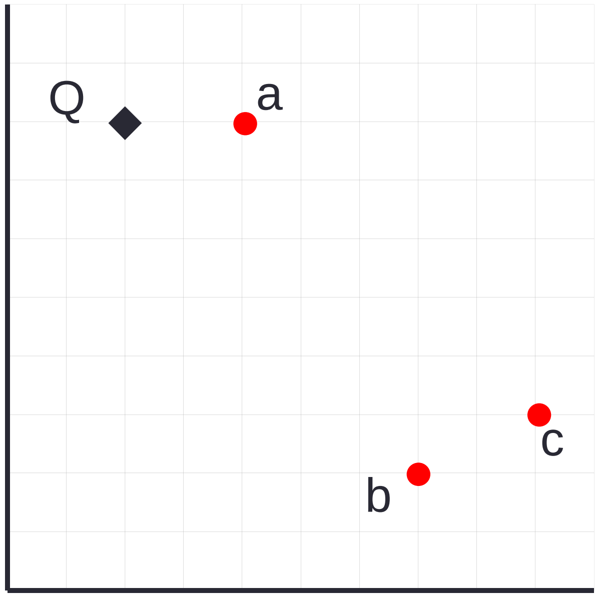
Otherwise you could say:



Time series similarity

- Importance of triangular inequality:
- We aer looking for the closest point to Q, in a database of 3 objects. Suppose that the triangular inequality holds, and that we have precomplied a table of distance between all the items in the database.

	a	b	c
a		6.70	7.07
b			2.30
c			



Time series similarity

- Importance of triangular inequality:
- We calculate that **a** is 2 units from **Q**, **b** is 7,81 from **Q**. We don't have to calculate distance from **c** to **Q**!

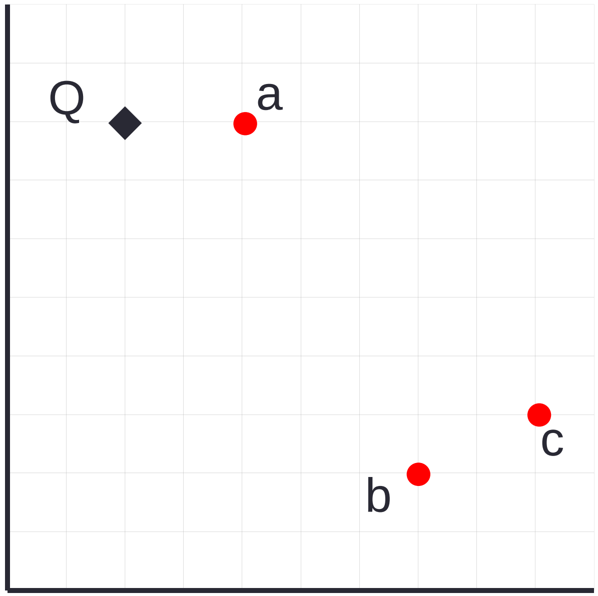
$$D(Q,\mathbf{b}) \leq D(Q,\mathbf{c}) + D(\mathbf{b},\mathbf{c})$$

$$D(Q,\mathbf{b}) - D(b,\mathbf{c}) \leq D(Q,\mathbf{c})$$

$$\mathbf{7.81} - \mathbf{2.30} \leq D(Q,\mathbf{c})$$

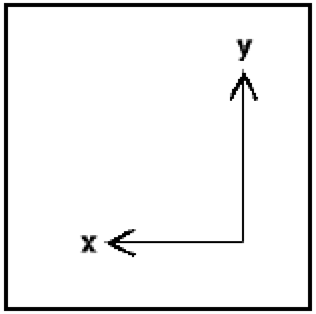
$$5.51 \leq D(Q,\mathbf{c})$$

	a	b	c
a		6.70	7.07
b			2.30
c			

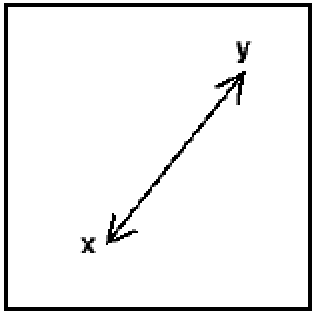


Time series similarity

- So we are looking for a suitable distance measure between two series
- Frequently used measures of similarity are based on a comparison of the overall shape of time-series
- **Minkowski metrics**



Manhattan



Euclidean

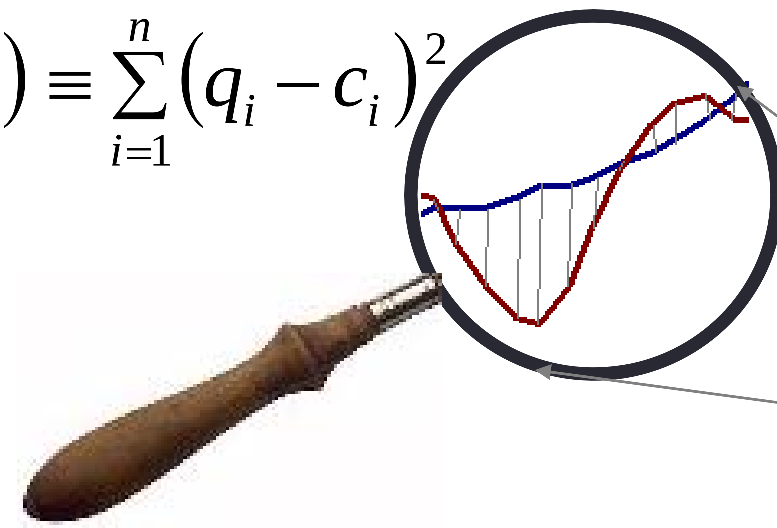
- **<u>Frequently used is not always the best</u>**

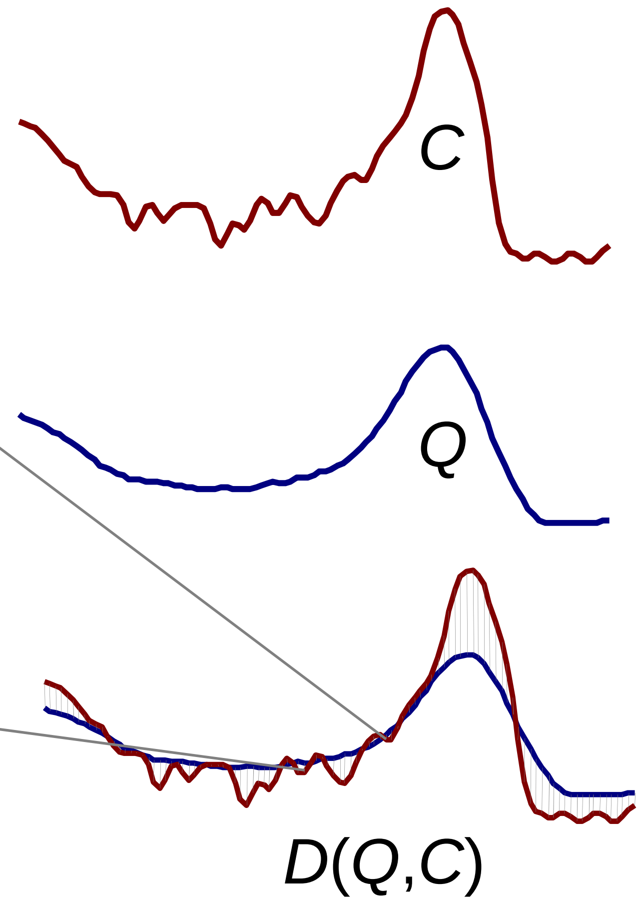
Time series similarity

- **Euclidean Distance Metric**
- Let Q and C are time series

$$D(Q,C) \equiv \sqrt{\sum_{i=1}^{n} (q_i - c_i)^2}$$

$$D_{squared}(Q,C) \equiv \sum_{i=1}^{n} (q_i - c_i)^2$$

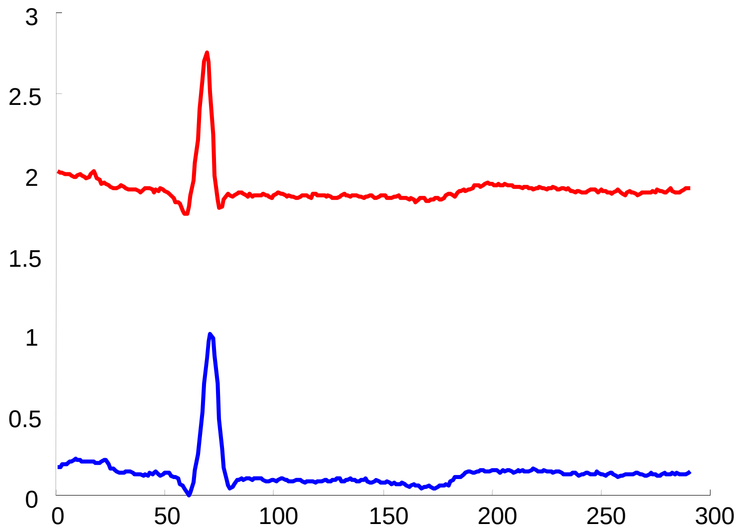


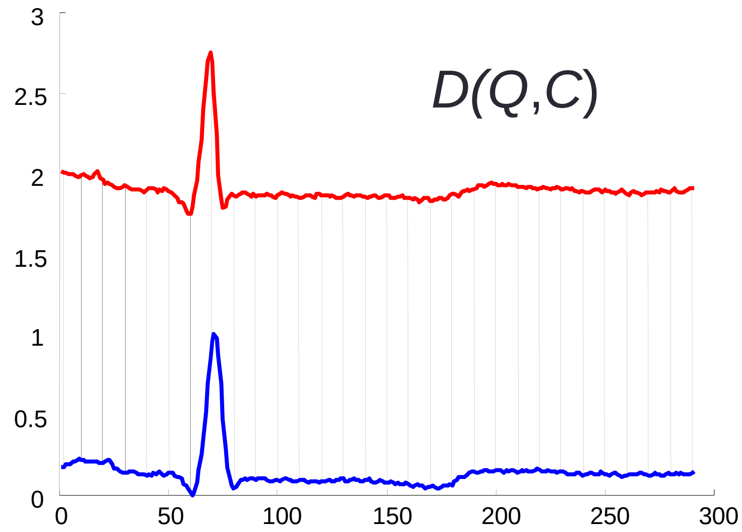


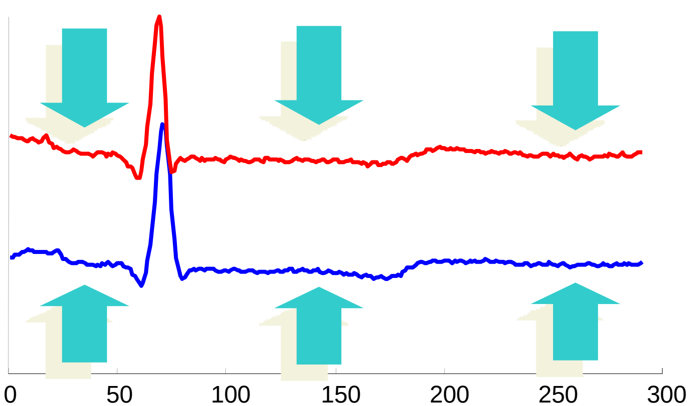
Time series similarity

- In most cases, the Euclidean metric does not give suitable results, due to "misrepresentation" in the source data.
  - Offset translation
  - Amplitude scaling
  - Linear trend
  - Noise

Time series similarity



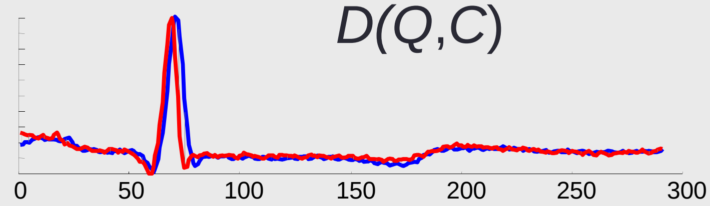




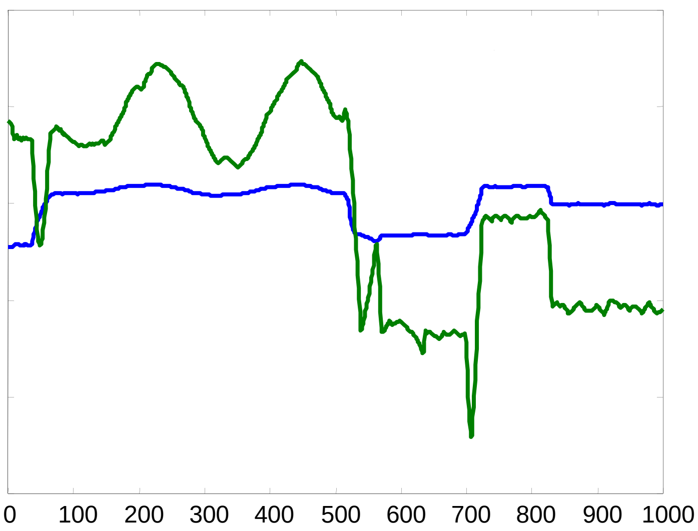
$Q = Q - \text{mean}(Q)$

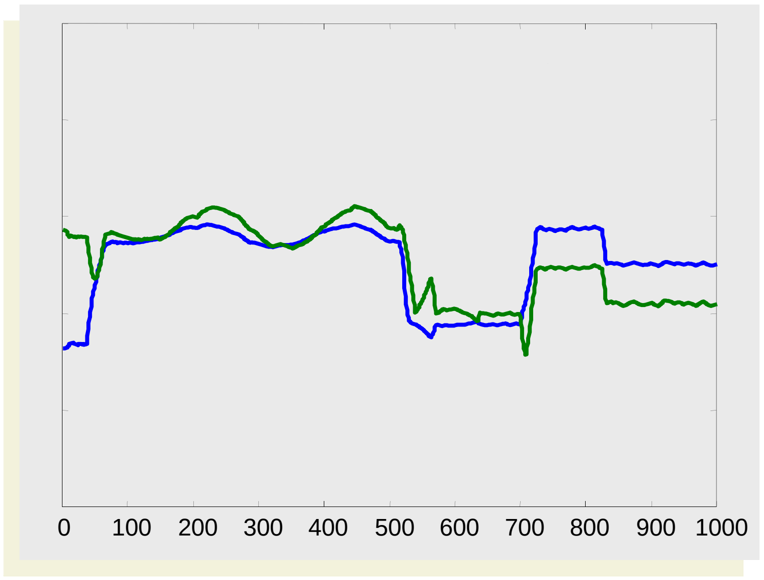
$C = C - \text{mean}(C)$

$D(Q,C)$



Time series similarity



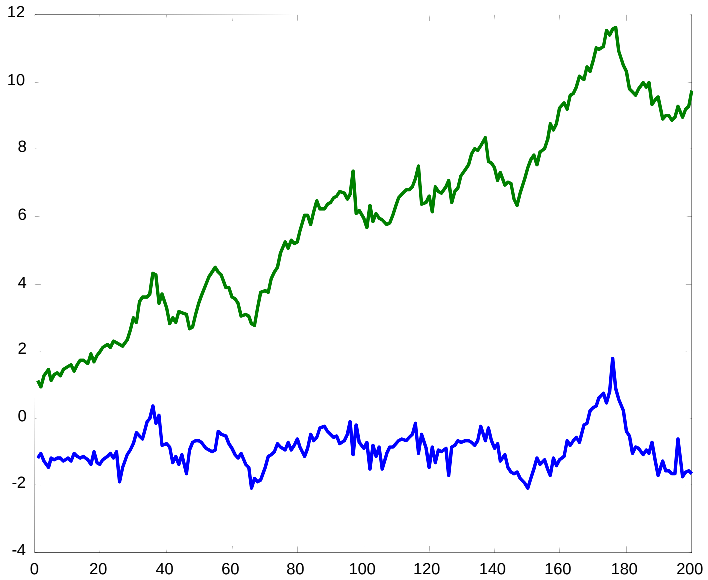


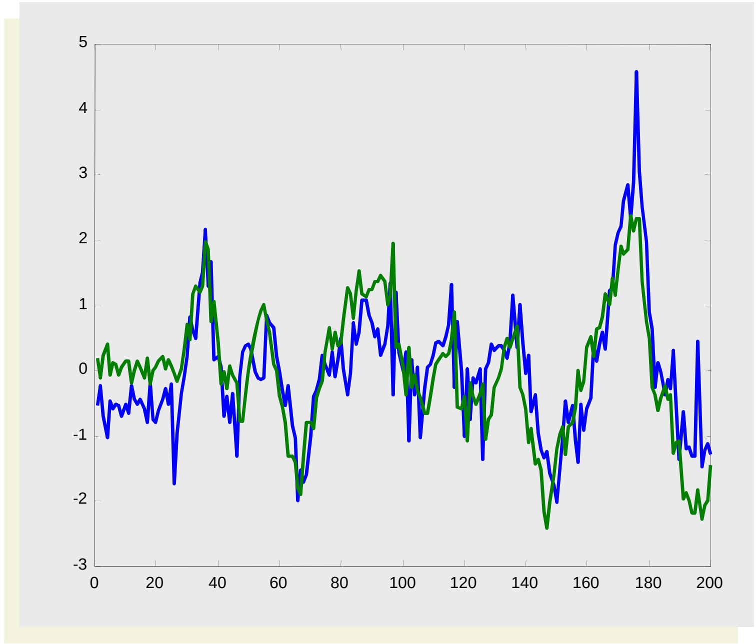
$Q = (Q - \text{mean}(Q)) / \text{std}(Q)$

$C = (C - \text{mean}(C)) / \text{std}(C)$

$D(Q,C)$

Time series similarity





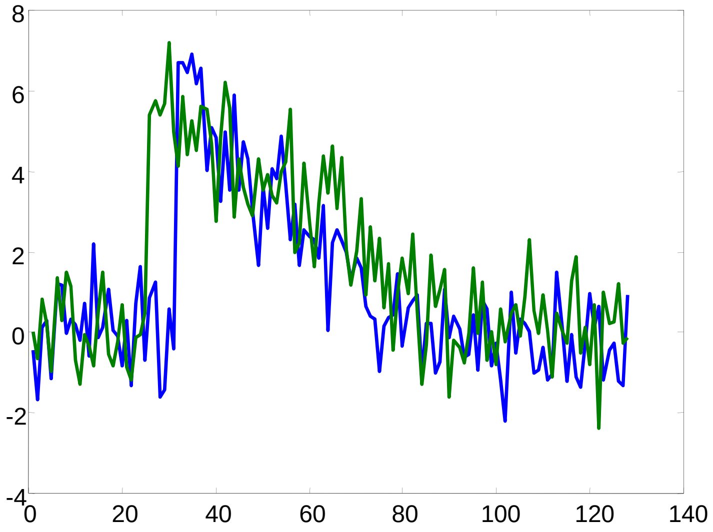
Find best linear approximation, than subtract that line from time series.

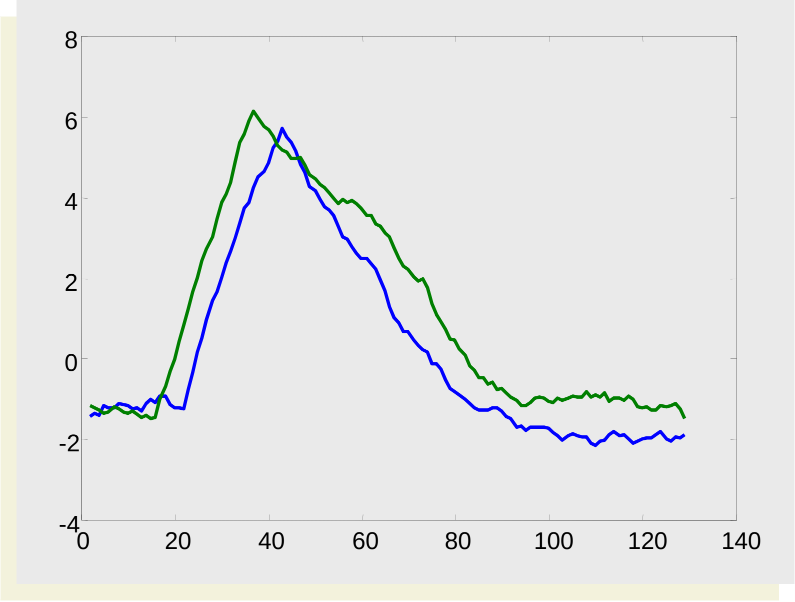
Removed linear trend

Removed offset translation

Removed amplitude scaling

Time series similarity





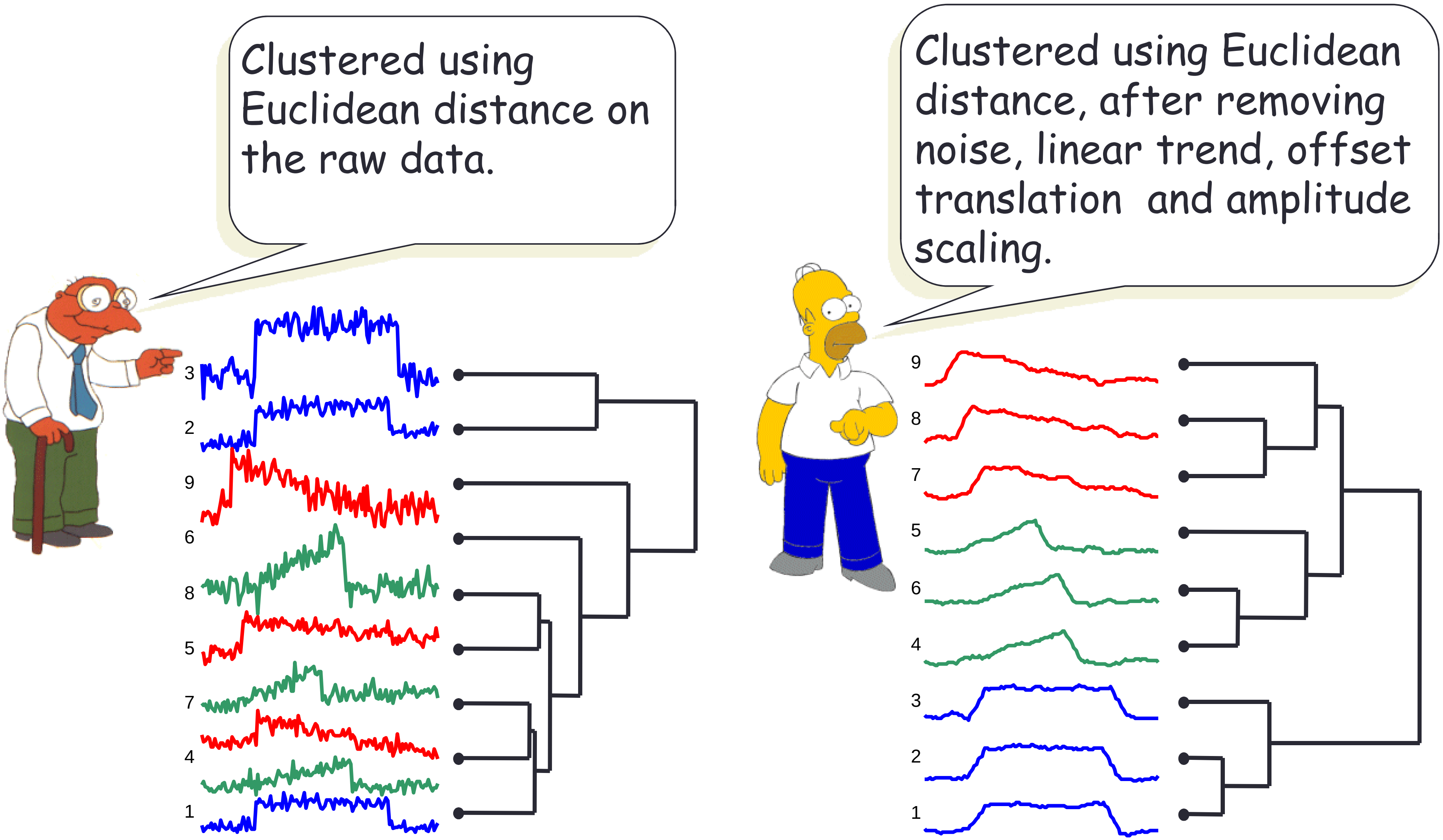
Replacing points with average of their neighbors

Q = smooth(Q)

C = smooth(C)

$D(Q,C)$

Time series similarity



Time series similarity

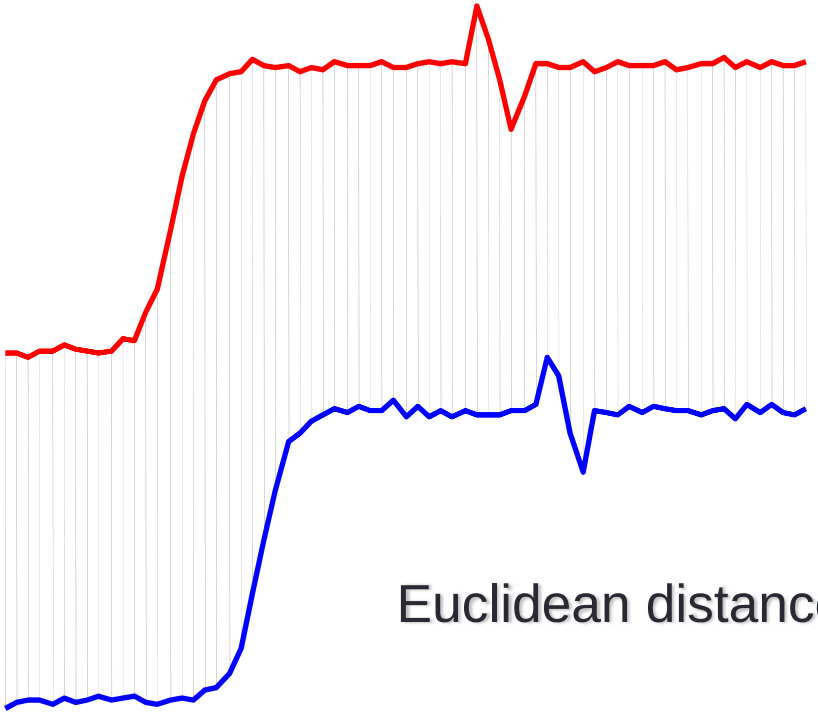
- To remeber – the “raw” time series may have distortions which we should remove before clustering, classification etc.

BUT

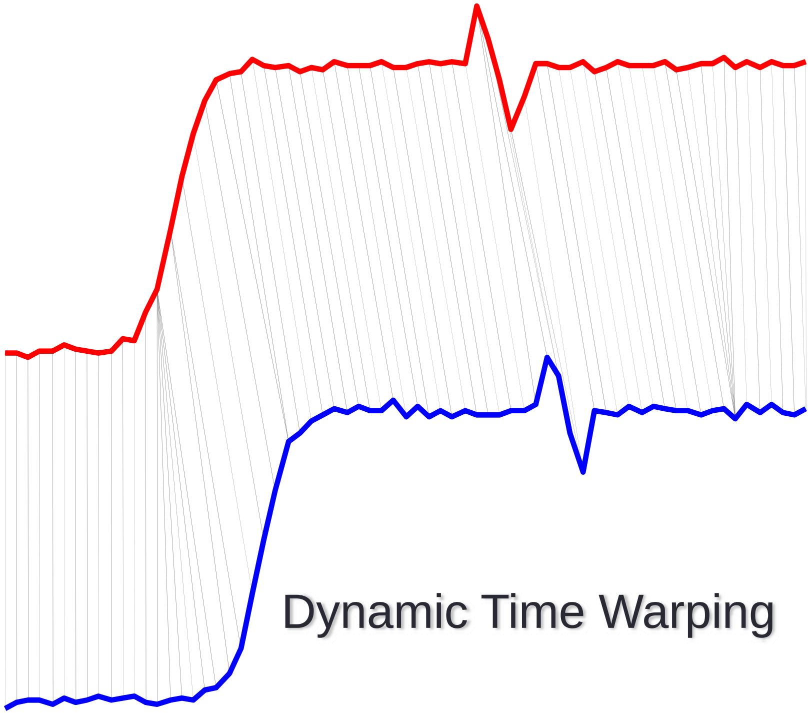
- Sometimes the distortions are the most interesting thing about the data!

Time series similarity

- Dynamic Time Warping (DTW)
- One method to deal with a phase shift between time series

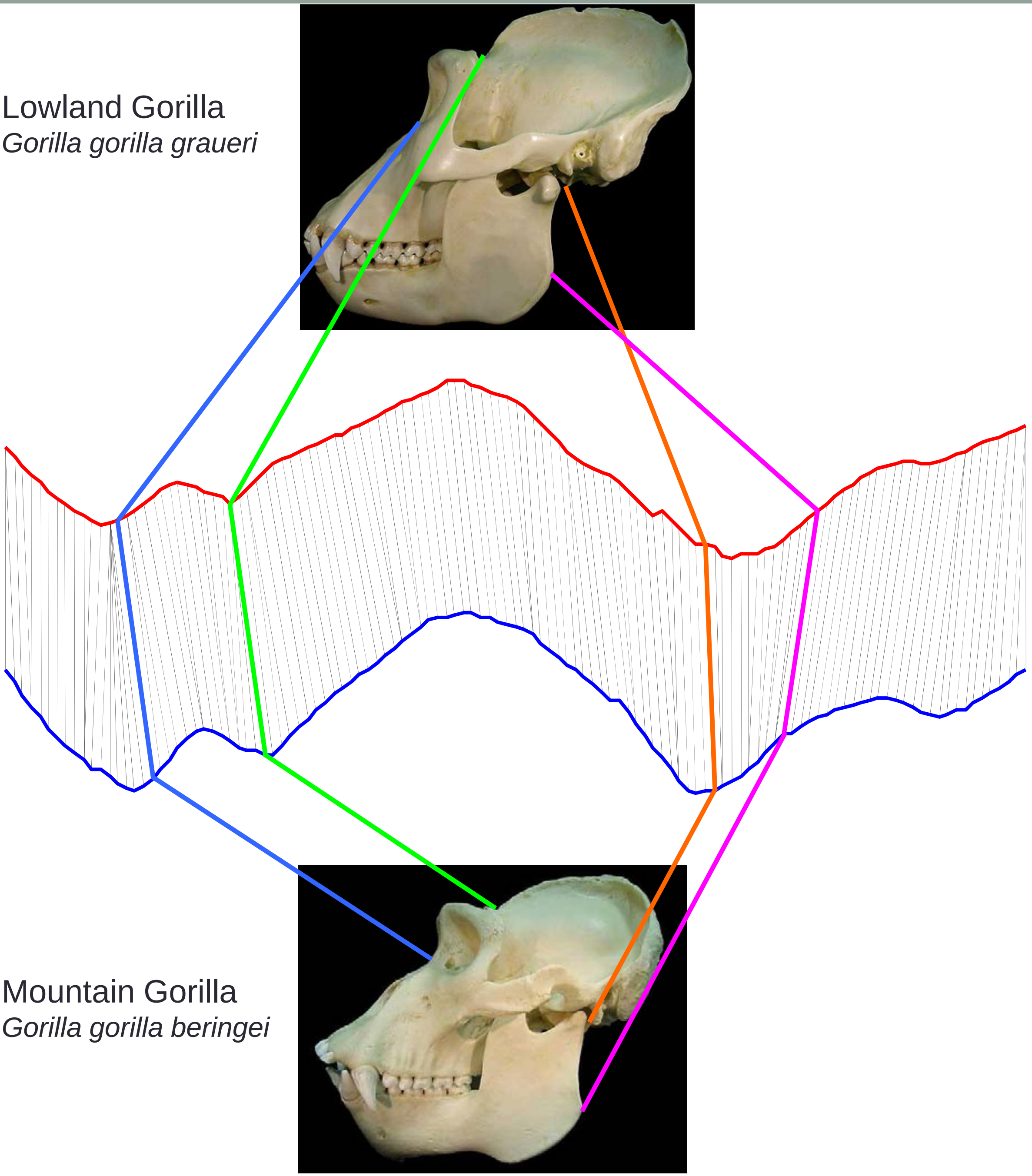


Euclidean distance

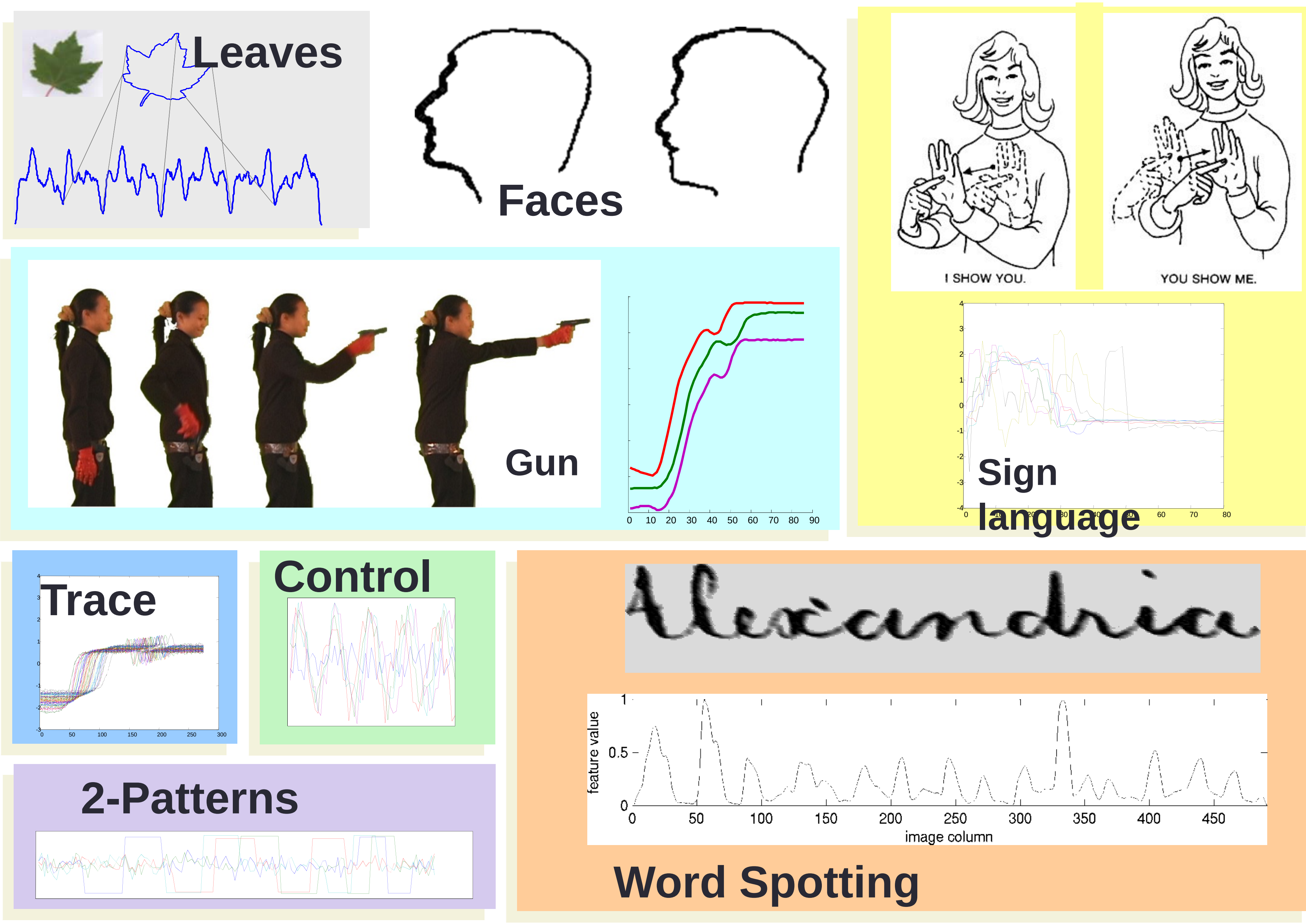


Dynamic Time Warping

Lowland Gorilla
Gorilla gorilla graueri



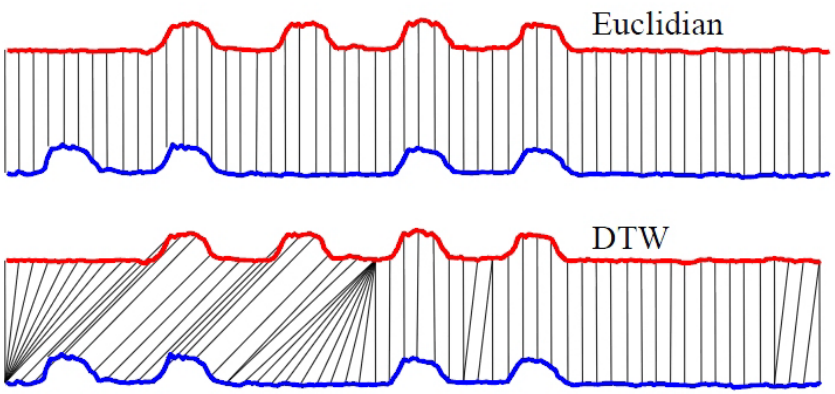
Mountain Gorilla
Gorilla gorilla beringei



Leaves
Faces
I SHOW YOU.
YOU SHOW ME.
Gun
Sign language
Trace
Control
2-Patterns
feature value
image column
Word Spotting

Time series similarity

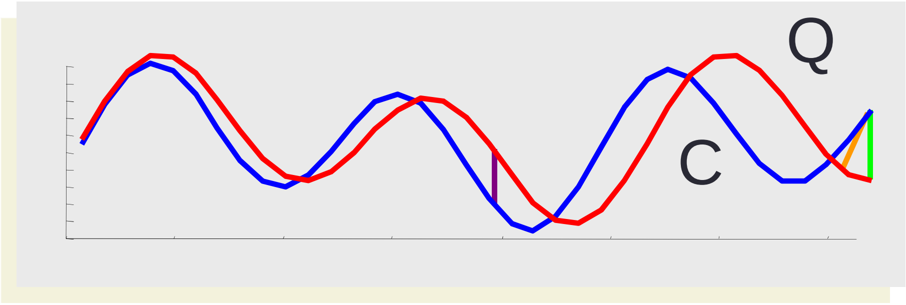
DTW is two to three orders of magnitude slower than Euclidean Distance (time in msec)

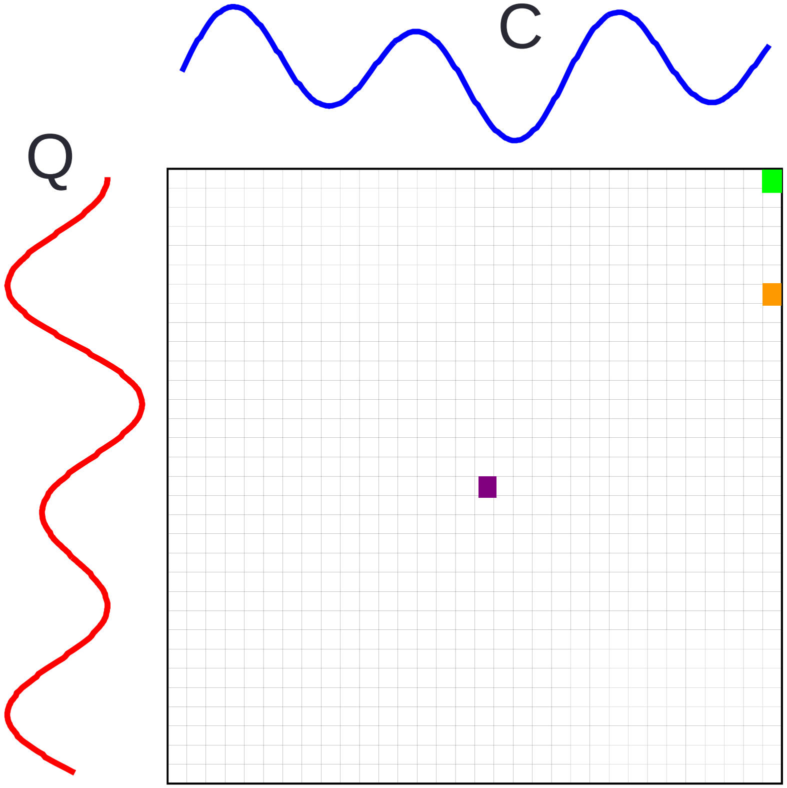


Dataset	Euclidean	DTW
Word Spotting	40	8,600
Sign language	10	1,110
GUN	60	11,820
Nuclear Trace	210	144,470
Leaves	150	51,830
(4) Faces	50	45,080
Control Chart	110	21,900
2-Patterns	16,890	545,123

Time series similarity

- We create a matrix the size of |Q| by |C|, then fill it in with the distance between every pair of point in our two time series.

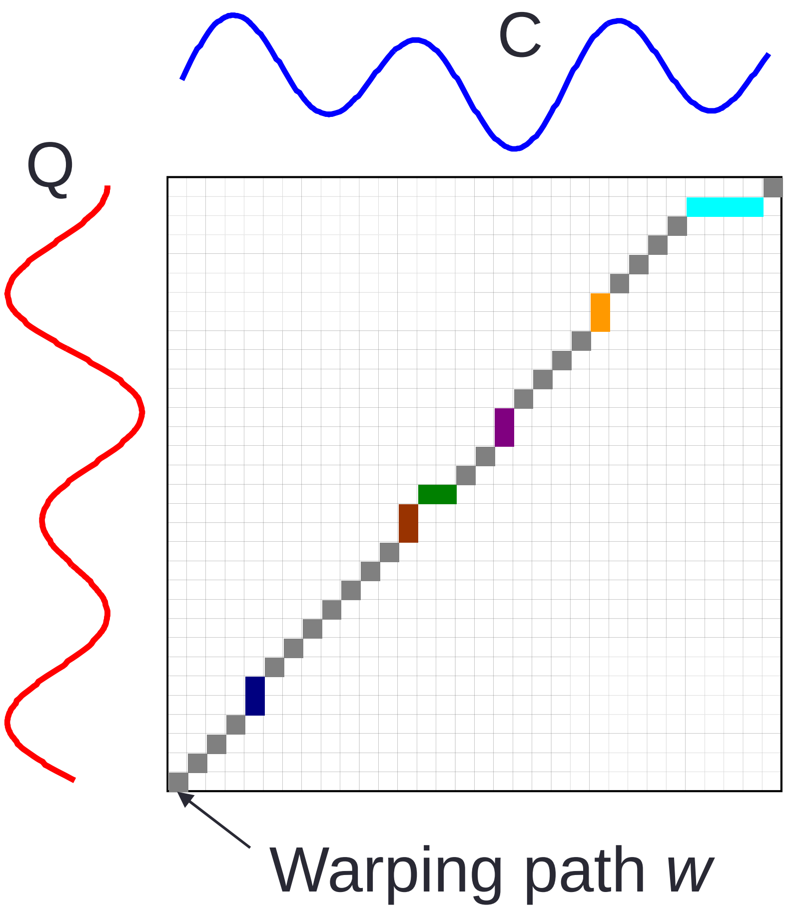




Time series similarity

- Every possible warping between two time series, is a path though the matrix. We want the best one.

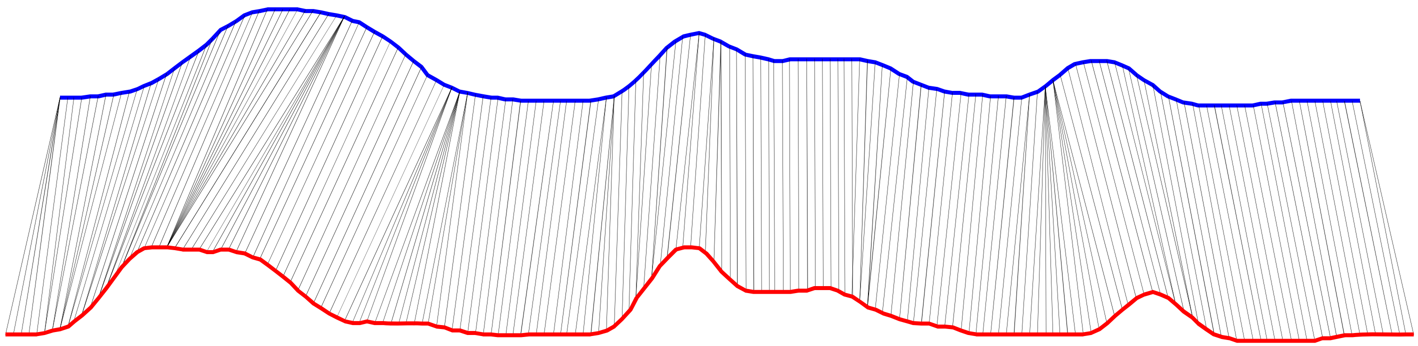
$$DTW(Q,C) = \min\left\{ \sqrt{\sum_{k=1}^{K} w_k} \Big/ K \right.$$



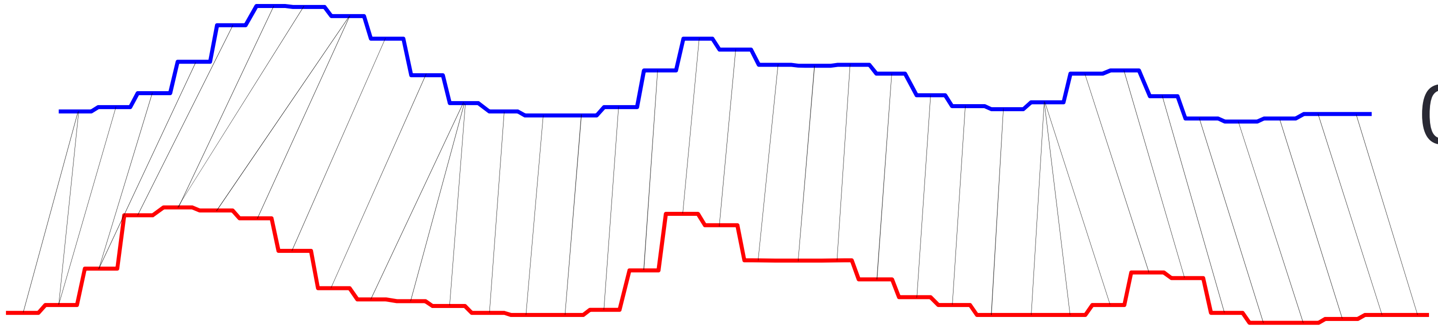
- Dynamic Time Warping gives much better results than Euclidean distance, but DTW is very slow to calculate!

Time series similarity

- Simple idea - approximate the time series with some compressed or downsampled representation, and do DTW on the new representation.



1.03 sec



0.07 sec

Time series similarity

- In general, it’s hard to speed up a single DTW calculation
- However, if we have to make many DTW calculations (which is almost always the case), we can potentiality speed up the whole process by **lowerbounding**.

- DTW(A,B)
- lower_bound_distance(A,B)

The true DTW function is very slow…

The lower bound function is very fast…

lower_bound_distance(A,B) ≤ DTW(A,B)

Time series similarity

- Lowerbounding – distance of points in space is less than or equal to the actual distance

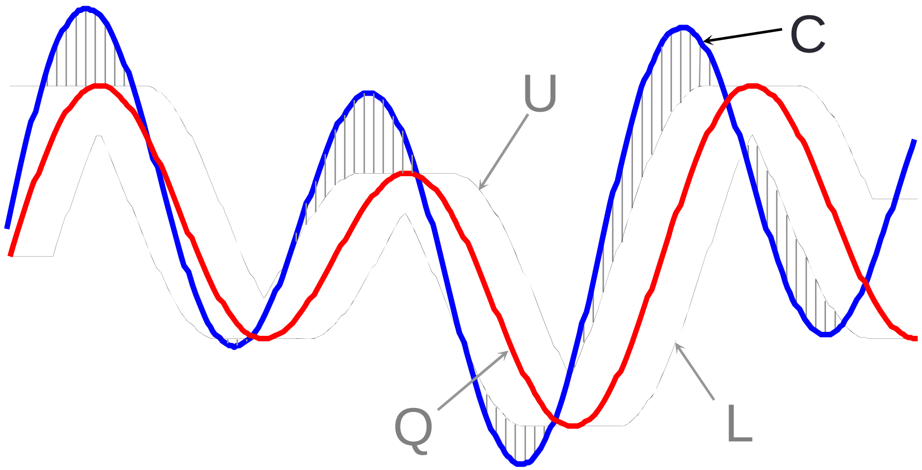
```
Algorithm Lower_Bounding_Sequential_Scan(Q)

best_so_far = infinity;
for all sequences in database
    LB_dist = lower_bound_distance(C_i, Q);
    if LB_dist < best_so_far
        true_dist = DTW(C_i, Q);
        if true_dist < best_so_far
            best_so_far = true_dist;
            index_of_best_match = i;
        endif
    endif
endfor
```

Time series similarity

Envelope - Based Lower Bound

LB_Keogh



$$LB\_Keogh(Q,C)=\sum_{i=1}^{n}\begin{cases}(q_i-U_i)^2 & if\ q_i>U_i\\(q_i-L_i)^2 & if\ q_i<L_i\\0 & otherwise\end{cases}$$

Time series – data preparation

Data cleansing

- Missing values
  - Skip values
  - Value estimation
    - Linear interpolation

- Noise reduction
  - Binning
  - Moving average

Data normalization

- Transforming data into the same range
  - Min-max

$$x_{norm} = \frac{x - x_{min}}{x_{max} - x_{min}}$$

  - Z-score

$$x_{norm} = \frac{x - \mu}{\sigma}$$

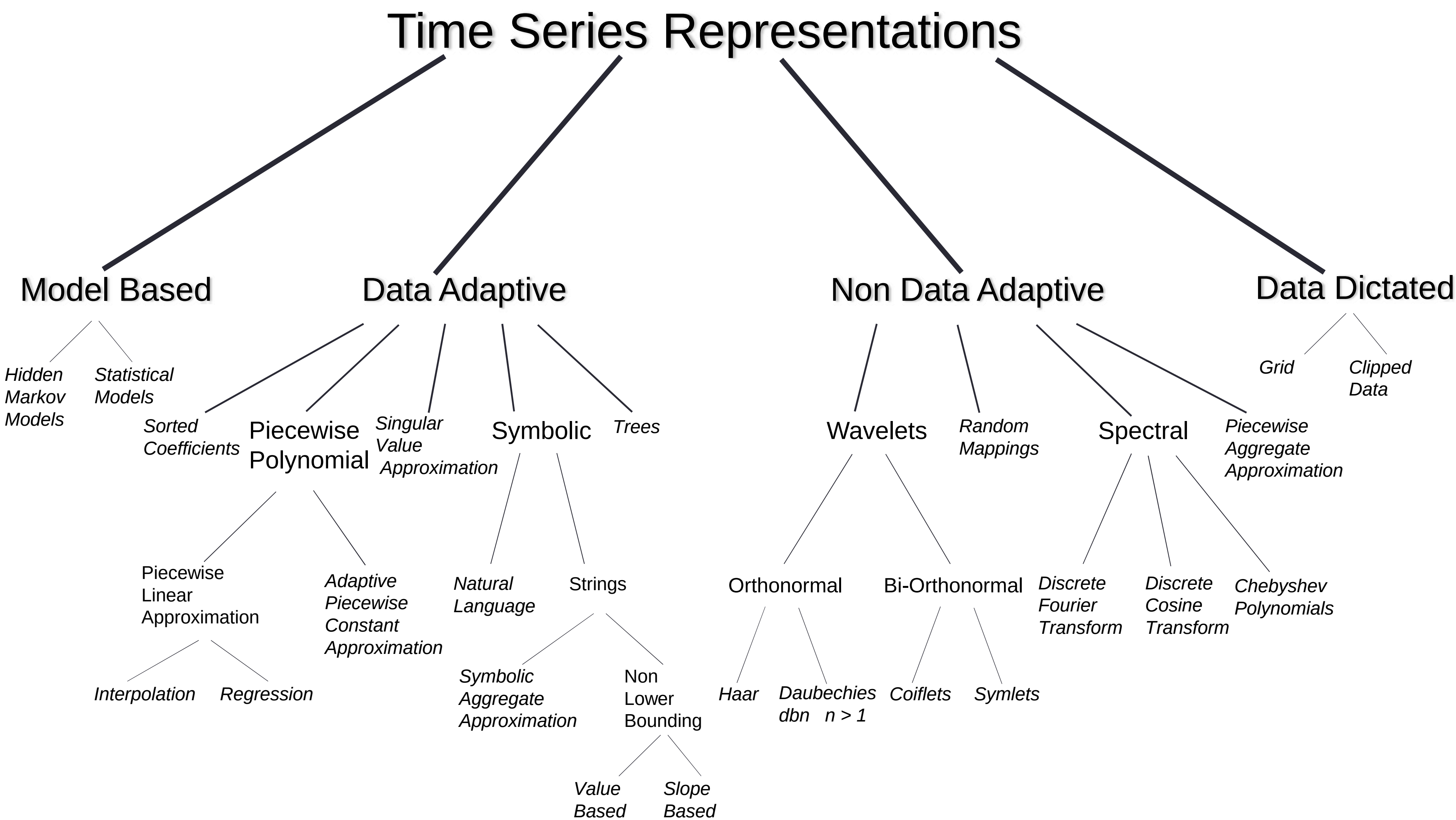
Time Series Representations

- We have already told how to define similarity, but how find it quickly?

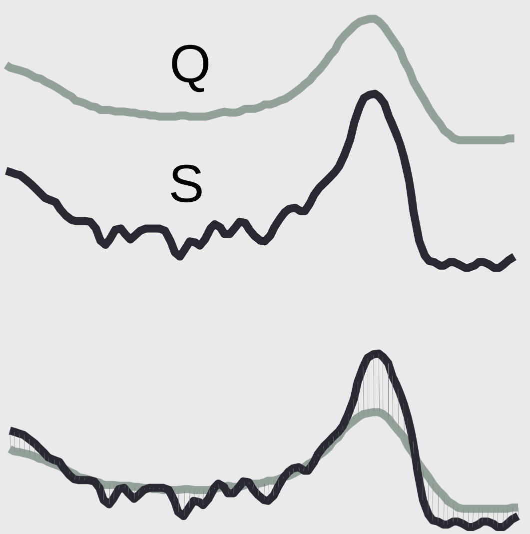
- Generic Data Mining Algorithm:
  - Create an approximation of the data, which will fit in main memory, yet retains the essential features of interest.
  - Approximately solve the problem at hand in main memory
  - Make few accesses to the original data on disk to confirm the solution obtained in step 2

- **But which approximation should we use?**

Time Series Representations



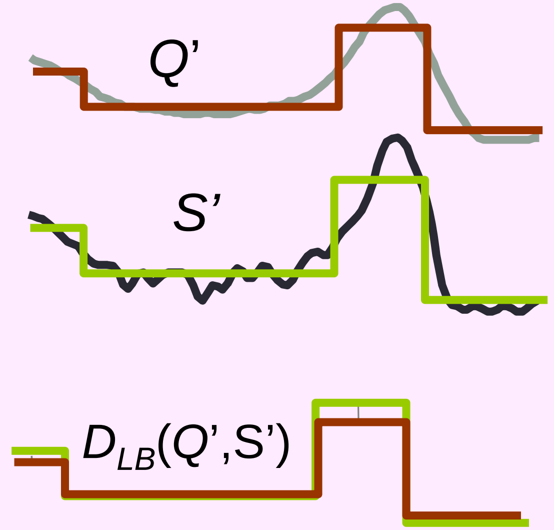
Time Series Representations



$$D(Q,S) \equiv \sqrt{\sum_{i=1}^{n}(q_i - s_i)^2}$$

Raw data

Approximation

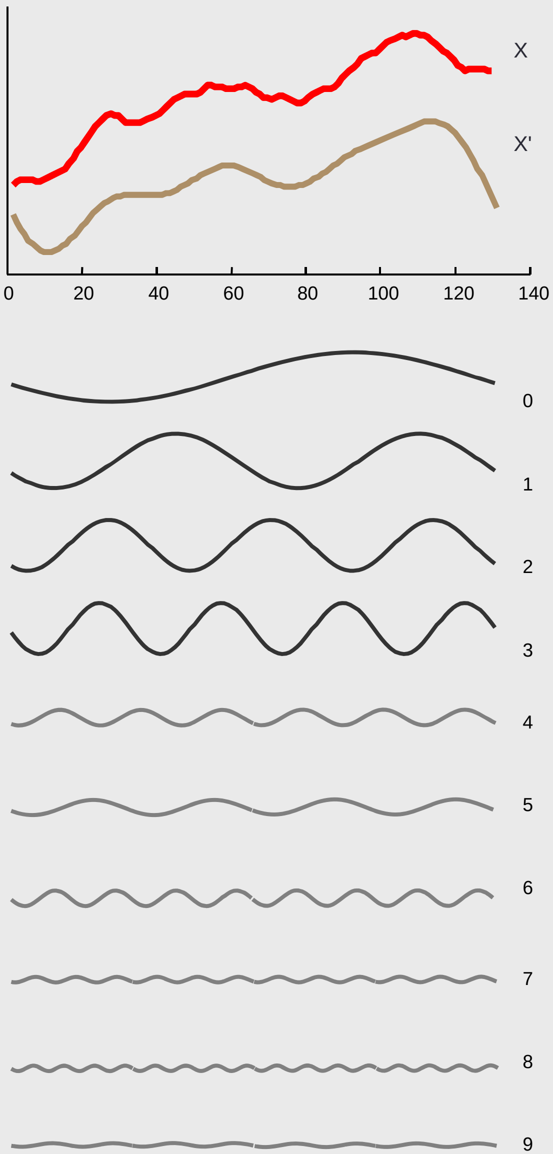


$$D_{LB}(Q',S') \equiv \sqrt{\sum_{i=1}^{M}(sr_i - sr_{i-1})(qv_i - sv_i)^2}$$

For every two time series Q and S approximation has to allow **lower bounding**

$$D_{LB}(Q',S') \leq D(Q,S)$$

Discrete Fourier Transform



Represent the time series as a linear combination of sines and cosines.

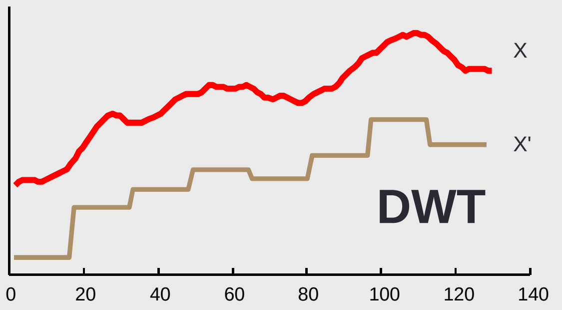
Jean Fourier

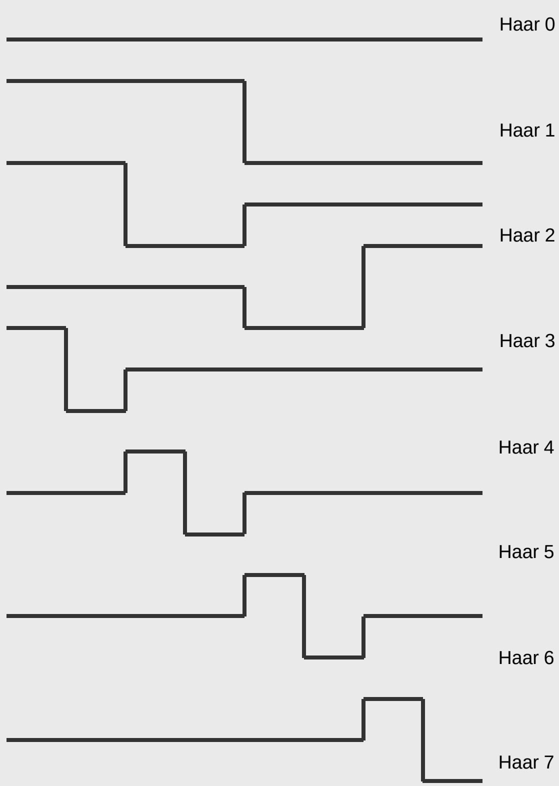
1768-1830

$$C(t) = \sum_{k=1}^{n} \left(A_k \cos(2\pi w_k t) + B_k \sin(2\pi w_k t) \right)$$

- Good ability to compress most natural signals, O(n*log(n)).
- Difficult to deal with sequences of different lengths.

Discrete Wavelet Transform





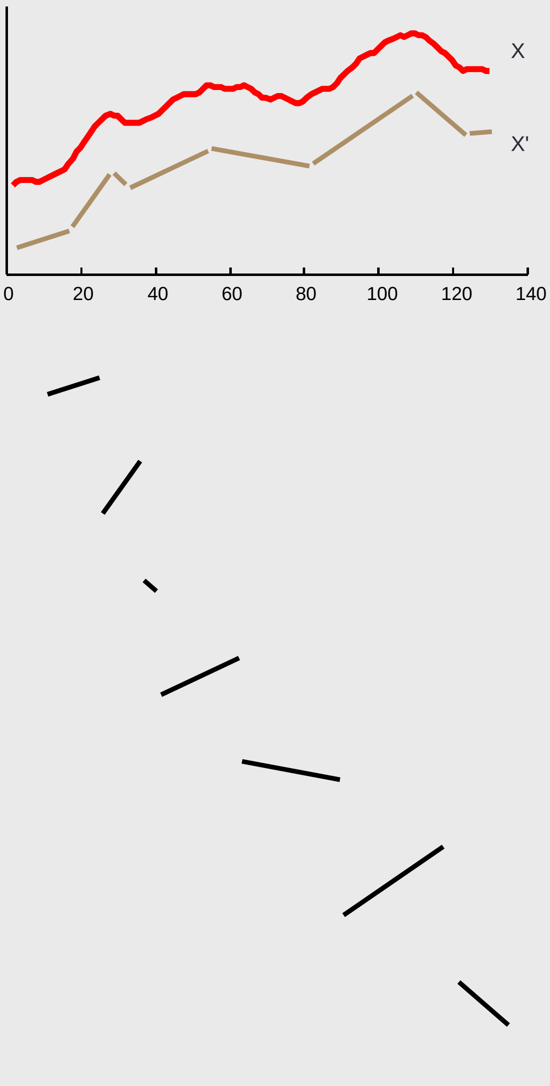
Represent the time series as a linear combination of Wavelet basis functions.

Alfred Haar

1885-1933

- Good ability to compress
- Fast linear time algorithms for DWT
- Able to support some interesting non-Euclidean similarity measures
- Signal must have a length n = 2^int

Piecewise Linear Approximation



Represent the time series as a sequence of straight lines. Lines could be **connected** (*N*/2) or **disconnected** (N/3). Series is replaced by segments, their number is much smaller than no of points in the original series.

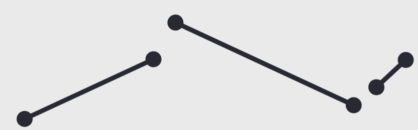
- O(n2N), which is too slow for data mining
- Faster heuristic solutions
  - Top-Down
  - Bottom-Up
  - Sliding Window
- Not suitable for indexing

Karl Friedrich Gauss

1777 - 1855

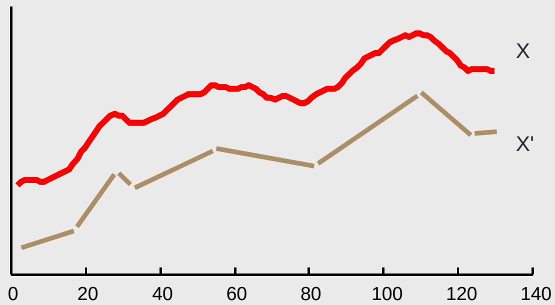
Each line segment has
- `length`
- `left_height`

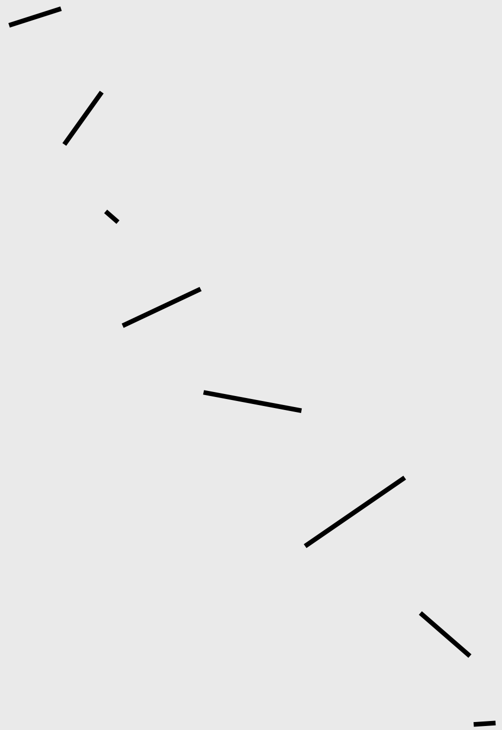
(`right_height` can be inferred by looking at the next segment)



Each line segment has
- `length`
- `left_height`
- `right_height`

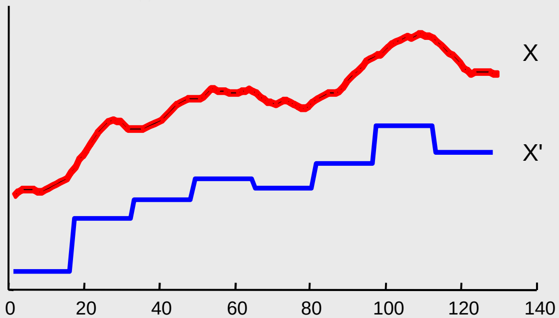
Piecewise Linear Approximation

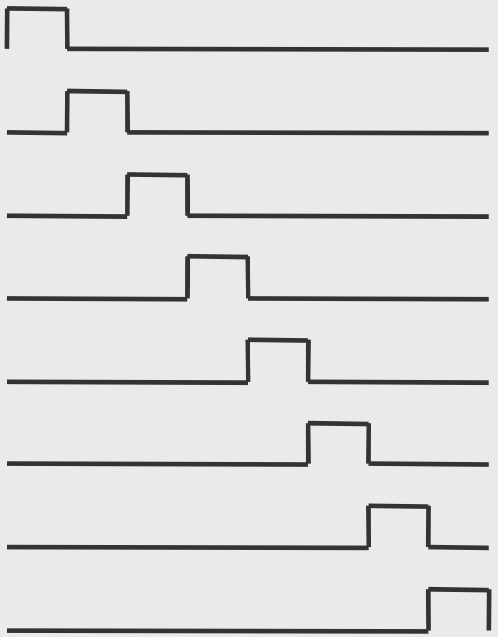




- **Sliding Windows**
  - Gradually from left increase the potential segment until the deviation from the original series does not exceed the limit set by the user
  - Relatively good example for stock data
  - Beware of extreme values
  - Variant with k-value adding is faster
- **Top-down**
  - Divide series at a suitable location (eg, minimum or maximum values) into two segments. Then further divide into smaller segments until the stopping criterion.
- **Bottom-up**
  - Complement previous - series is divided into n / 2 segments, which gradually combine.
- **SWAB** – sliding window and bottom-up
  - Buffer segments of length **w** is filled with sliding window, then bottom-up.

Piecewise Aggregate Approximation





Split series into **n** segments, calculate average for each segment and this value will replace all points in the segment. All segments have the same length.

$$\bar{x}_i = \frac{N}{n} \sum_{j=\frac{n}{N}(i-1)+1}^{\frac{n}{N}i} x_j$$

Given the reduced dimensionality representation we can calculate the approximate Euclidean distance

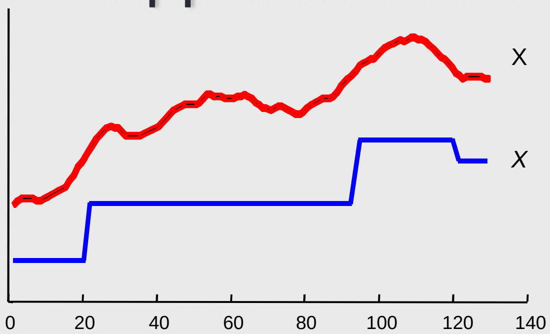
$$DR(\bar{X},\bar{Y}) \equiv \sqrt{\frac{n}{N}} \sqrt{\sum_{i=1}^{N} \left(\bar{x}_i - \bar{y}_i\right)^2}$$

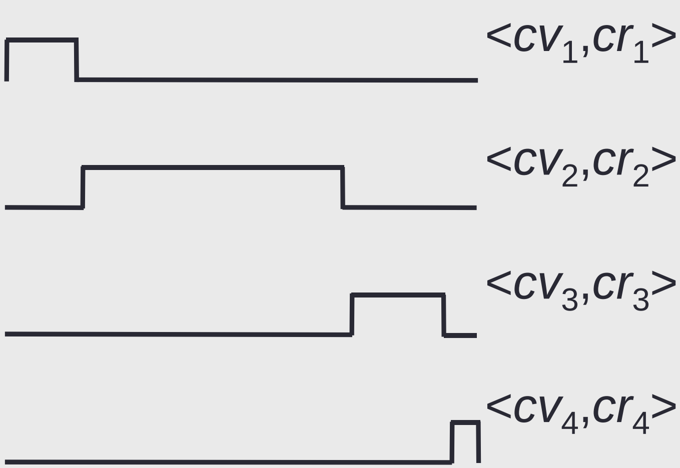
- This measure is provably lower bounding.
- Extremely fast to calculate
- Support series of arbitrary lengths
- Can support any Minkowski metric
- Supports non Euclidean measures
- Simple, intuitive

Time Series Representations

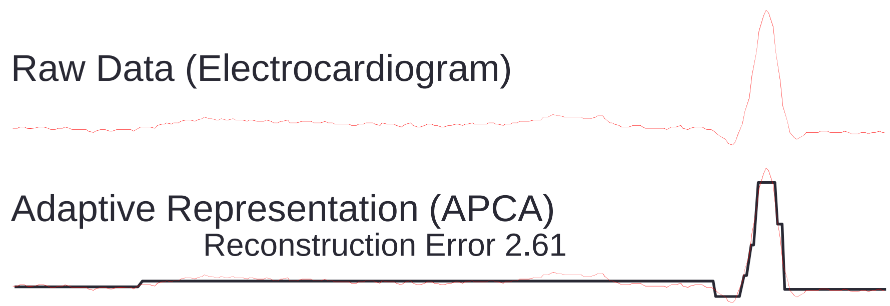
Piecewise
Aggregate
Approximation

Adaptive Piecewise Constant Approximation



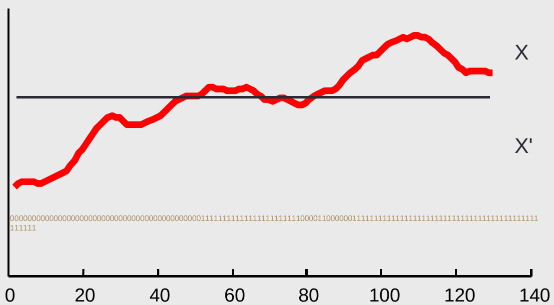


Generalize PAA to allow the piecewise constant segments to have arbitrary lengths. Note that we now need 2 coefficients to represent each segment, its value and its length.



- Many advantages, but challenging to index - implementation exists, but is very challenging.
- Very fast O(n)
- More efficient as other approaches
- Support series of arbitrary lengths.
- Supports non Euclidean measures.

Clipped data



…11000011000000111…

.

44 Zeros

23 Ones

4 Zeros

2 Ones

6 Zeros

49

Ones

44

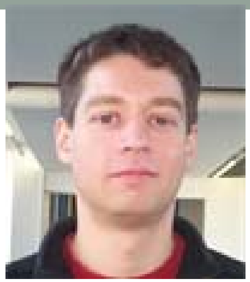
Zeros|23|4|2|6|49

Find the mean of a time series, convert values above the line to "1' and values below the line to "0".

Runs of "1"s and "0"s can be further compressed with run length encoding if desired.

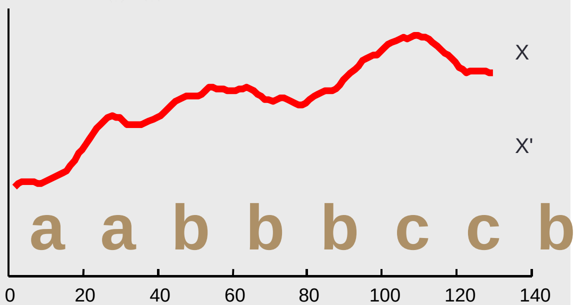
This representation does allow lower bounding.

Ultra compact representation which may be particularly useful for specialized hardware.



Tony Bagnall

Symbolic Approximation



a 0

a 1

b 2

b 3

b 4

c 5

c 6

7

b

Convert the time series into an alphabet of discrete symbols. Use string indexing techniques to manage the data.

- We could take advantage of a wealth of techniques from the very mature field of string processing and bioinformatics.
- How we should discretize the times series (discretize the values, the slope, shapes)?

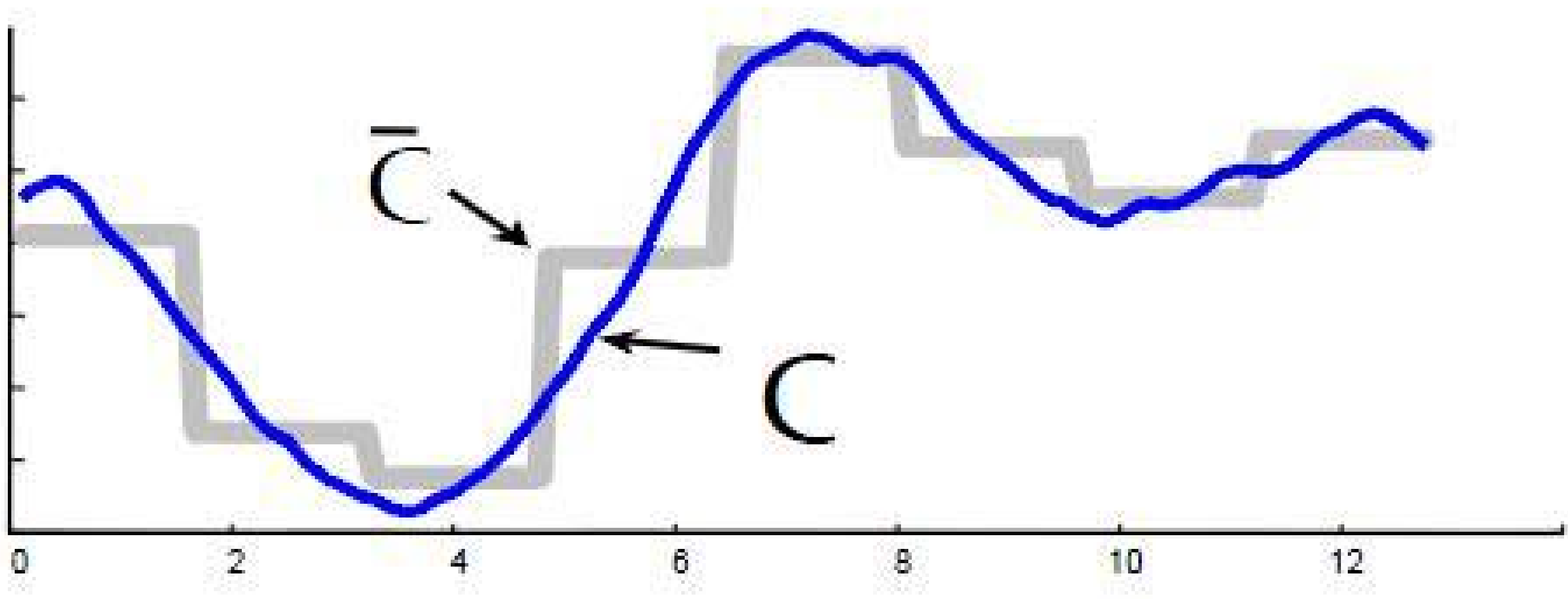
Symbolic Aggregate ApproXimation

- Algorithm for symbolic approximation
- 2002, Eamonn Keogh, University of California
- Significantly reduces the number of dimensions of the original time series
- Lower bounding of Euclidean distance

Symbolic Aggregate ApproXimation

- It consists of three steps
  - Normalisation of time series
    - The mean value is 0
  - PAA transformation
    - Divide time series into multiple segments, calculate the average for each segment, this value will replace all points in the segment
  - Discrete symbolic representation
    - Conversion time series to the alphabet symbols

Symbolic Aggregate ApproXimation



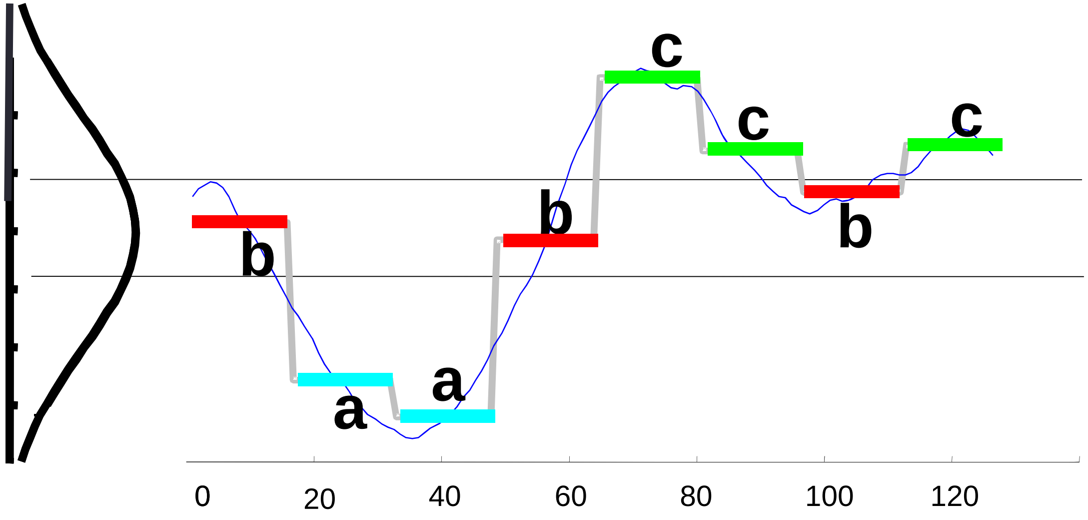
Symbolic Aggregate ApproXimation

- Normalized series has a normal distribution and shape of the Gaussian curve, we define the so-called breaking points, which divide the Gaussian curve into equal parts, discretization generates symbols with equal probability.

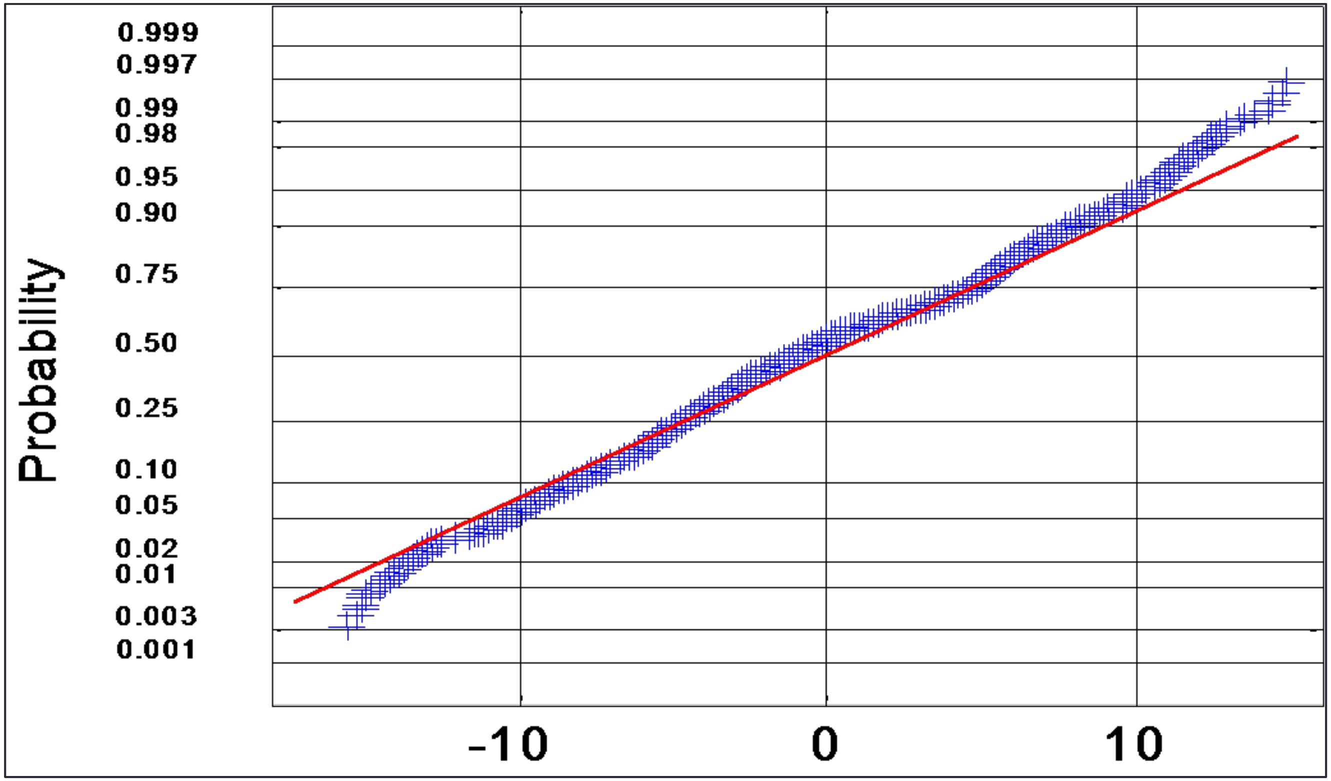
Symbolic Aggregate ApproXimation

β_i \ a	3	4	5	6	7	8	9	10
β_1	-0.43	-0.67	-0.84	-0.97	-1.07	-1.15	-1.22	-1.28
β_2	0.43	0	-0.25	-0.43	-0.57	-0.67	-0.76	-0.84
β_3		0.67	0.25	0	-0.18	-0.32	-0.43	-0.52
β_4			0.84	0.43	0.18	0	-0.14	-0.25
β_5				0.97	0.57	0.32	0.14	0
β_6					1.07	0.67	0.43	0.25
β_7						1.15	0.76	0.52
β_8							1.22	0.84
β_9								1.28

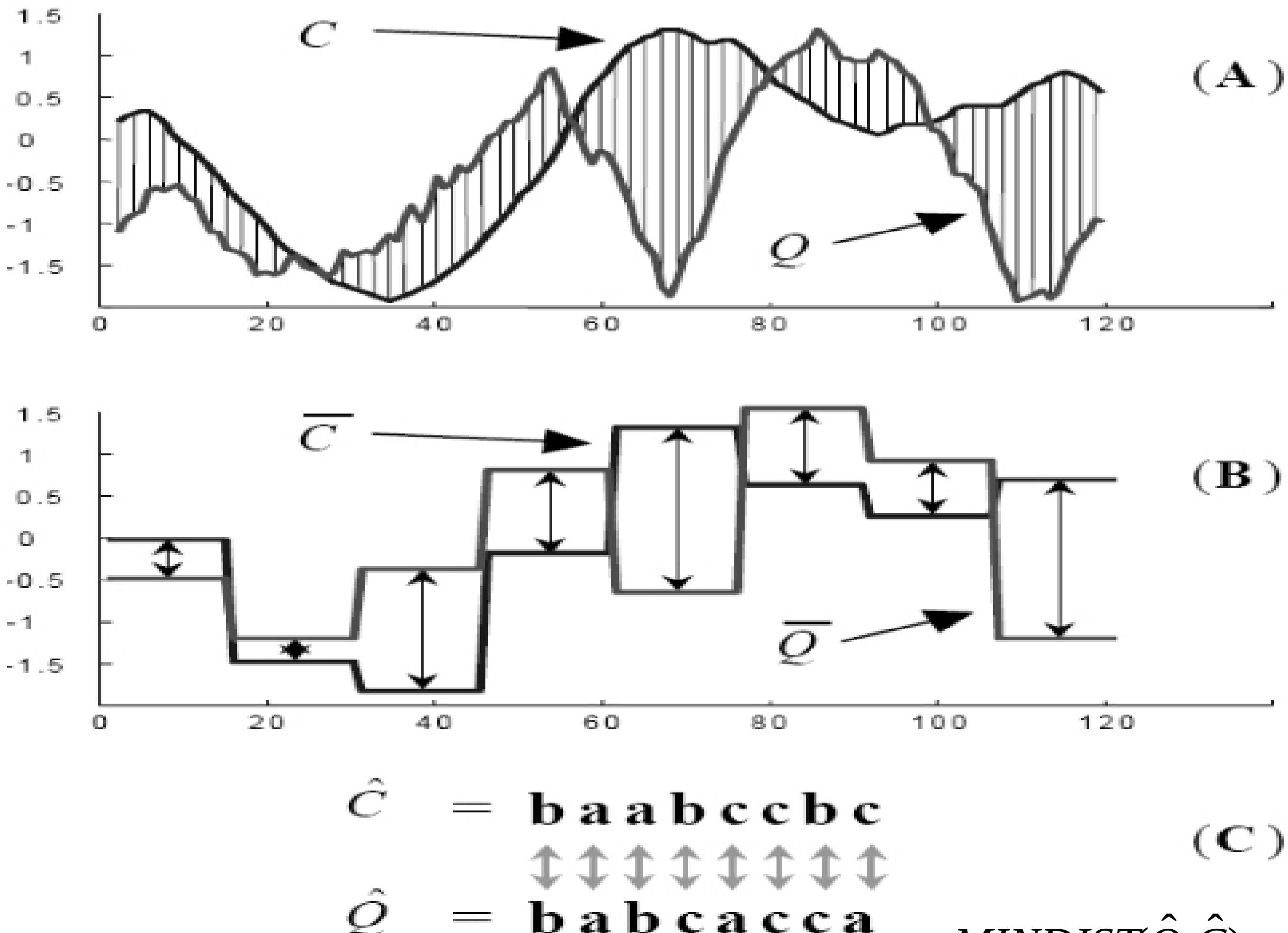
Symbolic Aggregate ApproXimation



Symbolic Aggregate ApproXimation

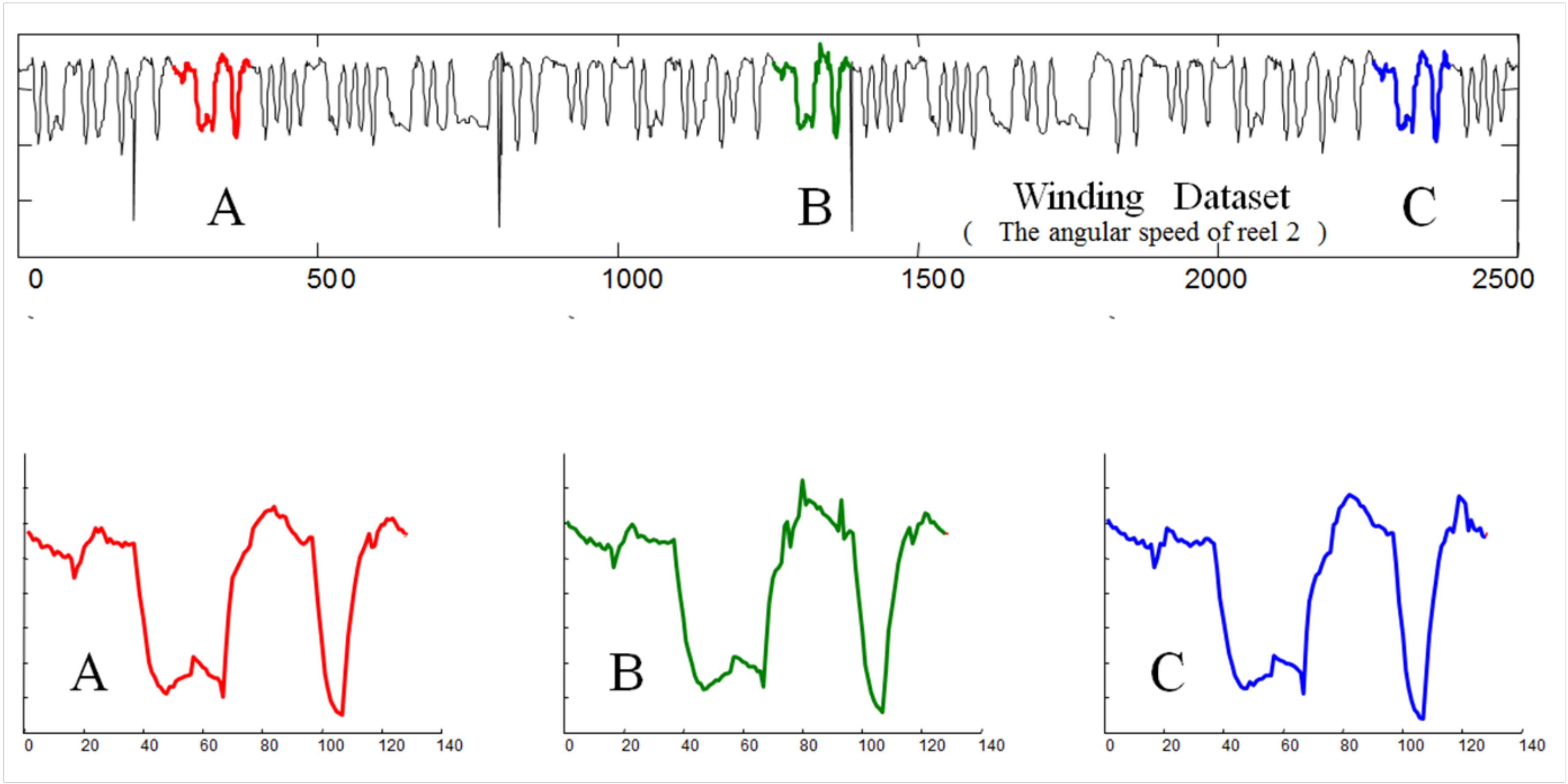


Symbolic Aggregate ApproXimation



$$MINDIST(\hat{Q},\hat{C}) \equiv \sqrt{\tfrac{n}{w}}\sqrt{\sum_{i=1}^{w}\left(dist(\hat{q}_i,\hat{c}_i)\right)^2}$$

Motif Discovery



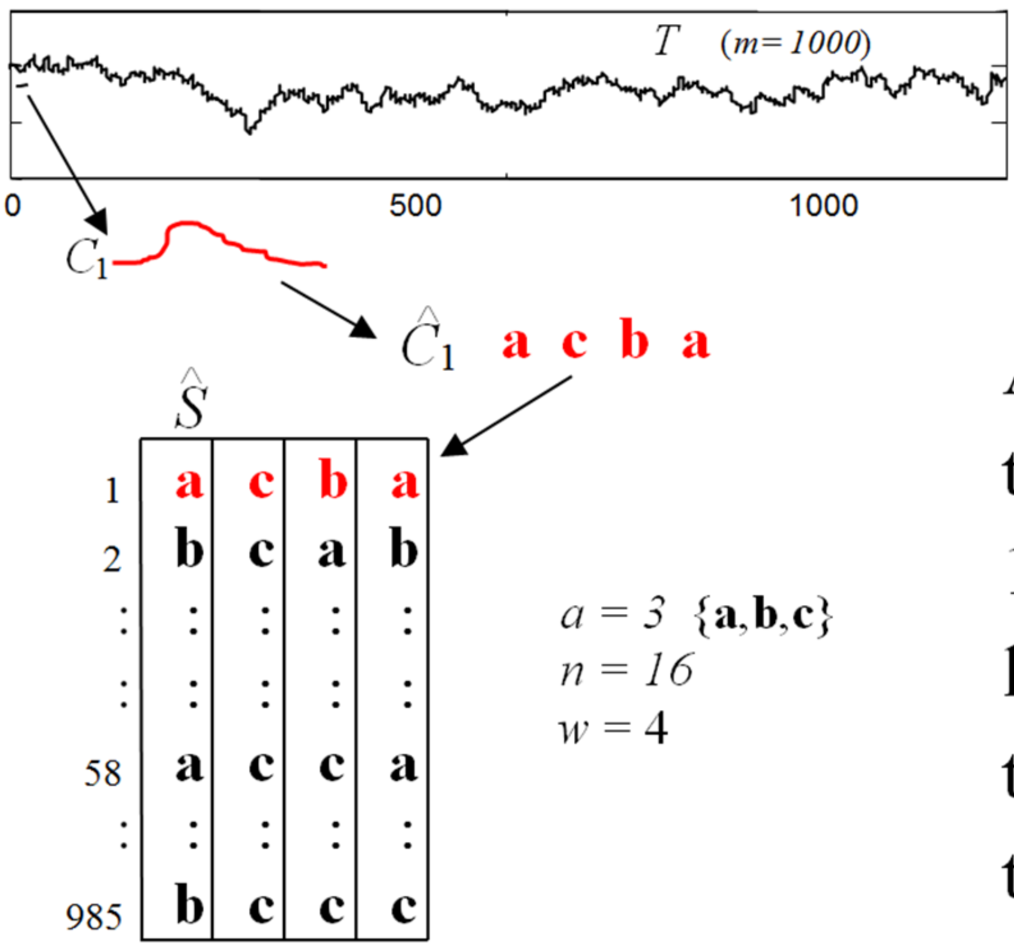
Motif Discovery

- We define motifs, but how do we find them?

- Brute force search algorithm is just too slow.

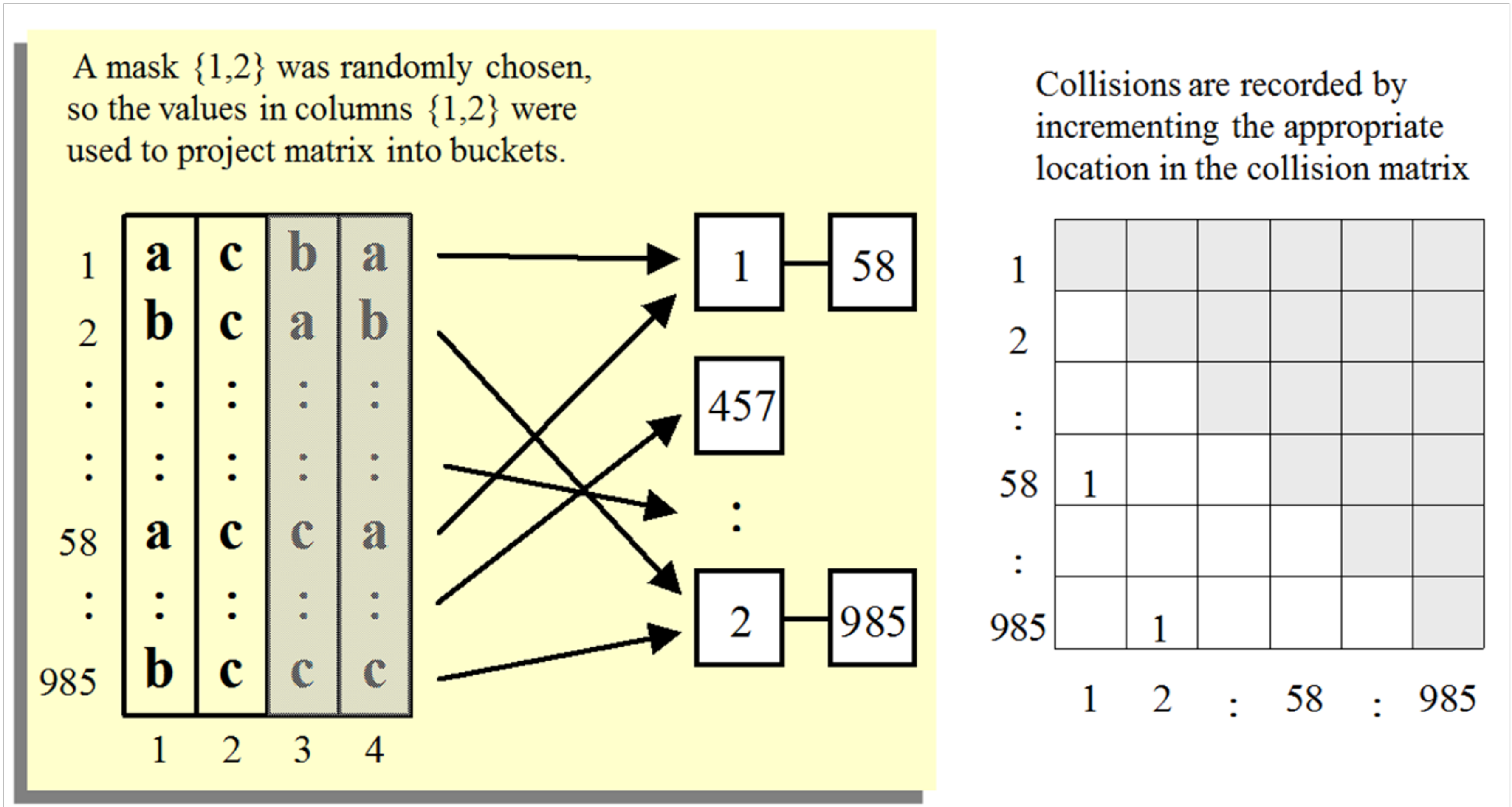
- The most reference algorithm is based on a idea from bioinformatics, **random projection*** and the fact that SAX allows use to lower bound discrete representations of time series.

Motif Discovery

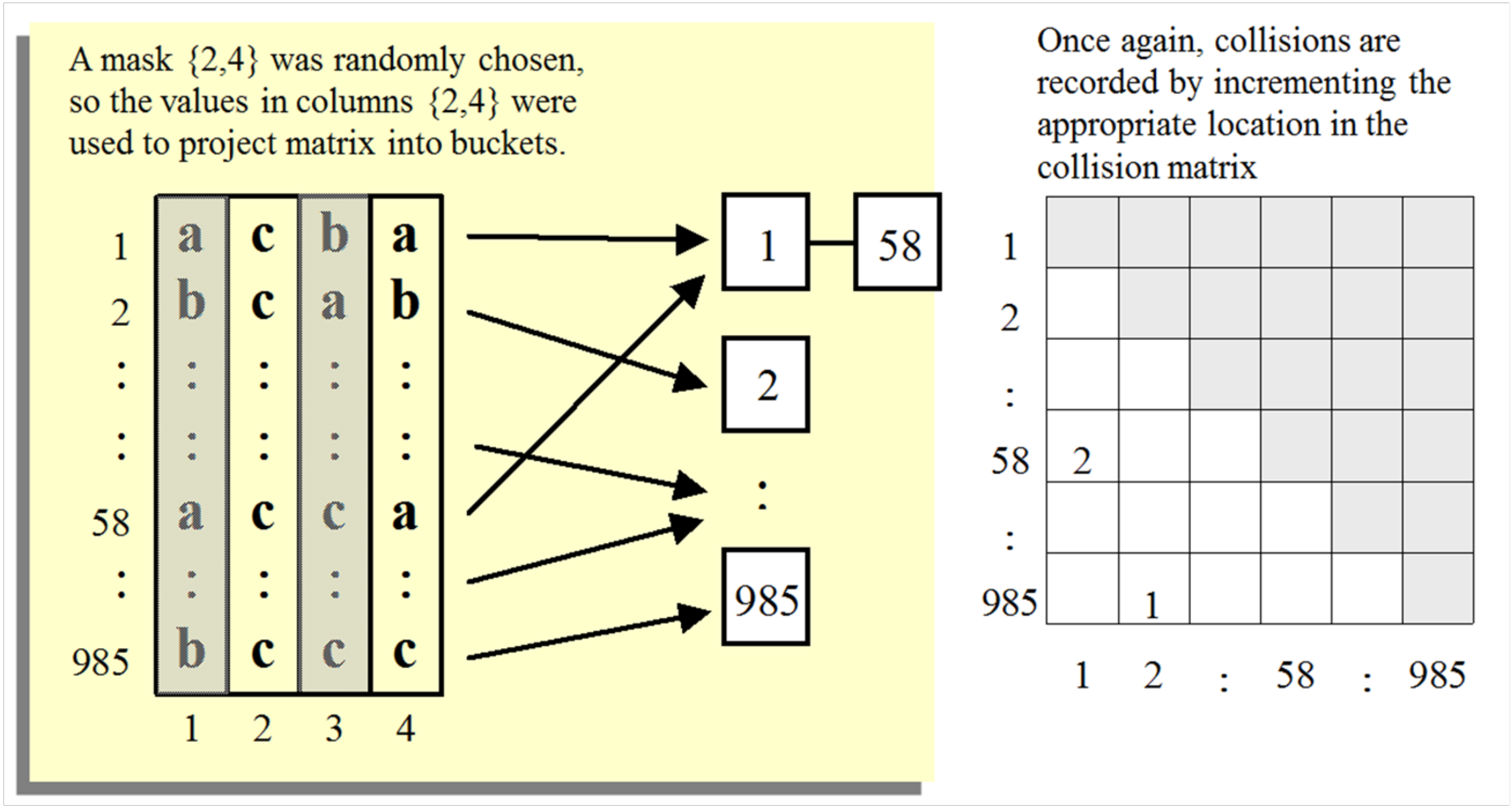


Assume that we have a time series T of length 1,000, and a motif of length 16, which occurs twice, at time T_1 and time T_{58}.

Motif Discovery



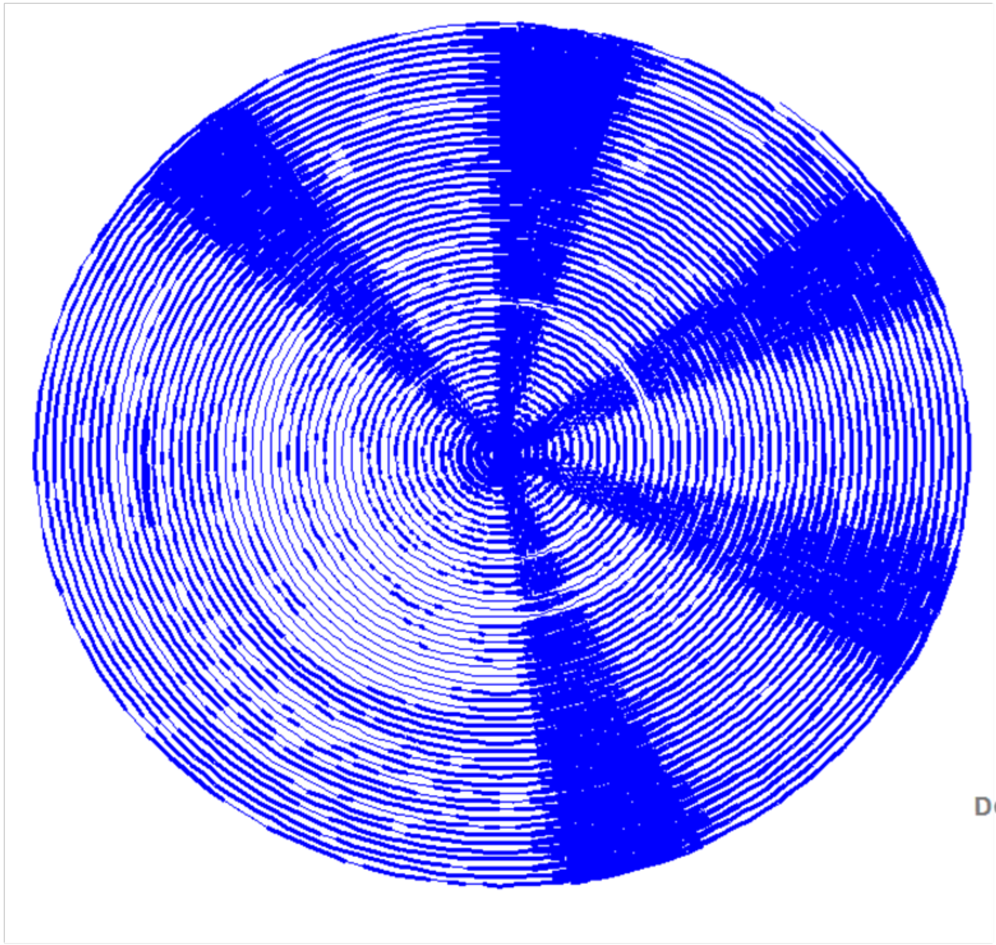
Motif Discovery

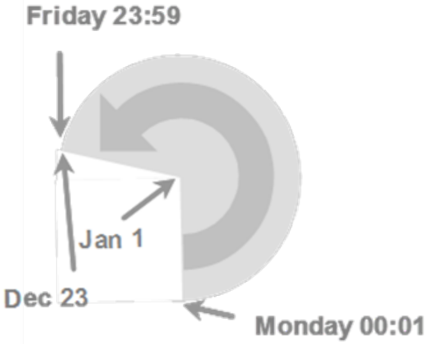


Motif Discovery

1						
2	2					
:	1	3				
58	**27**	2	1			
:	3	2	2	1		
985	0	1	2	1	3	

Visualization



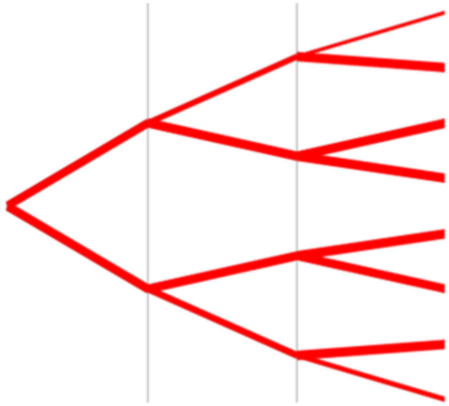


Visualization

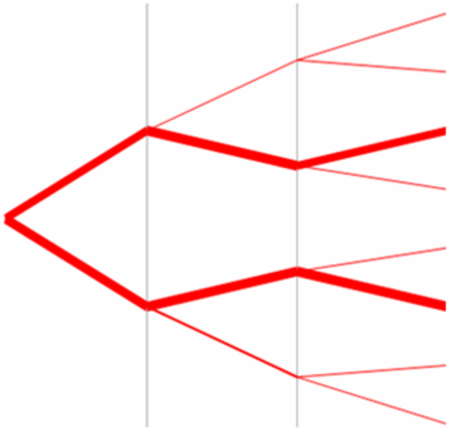
- Time Series Spiral
  - Simple and intuitive
  - Many extensions possible
  - Only useful on periodic data, and only then if you know the period

Visualization

010110010111100110100100001000101
001101101011100001010101110111110
001101101101111110100110010010001
101000111100110110100010111100010
110100110110011010000001001100010
011100000111010011001011000010100
10

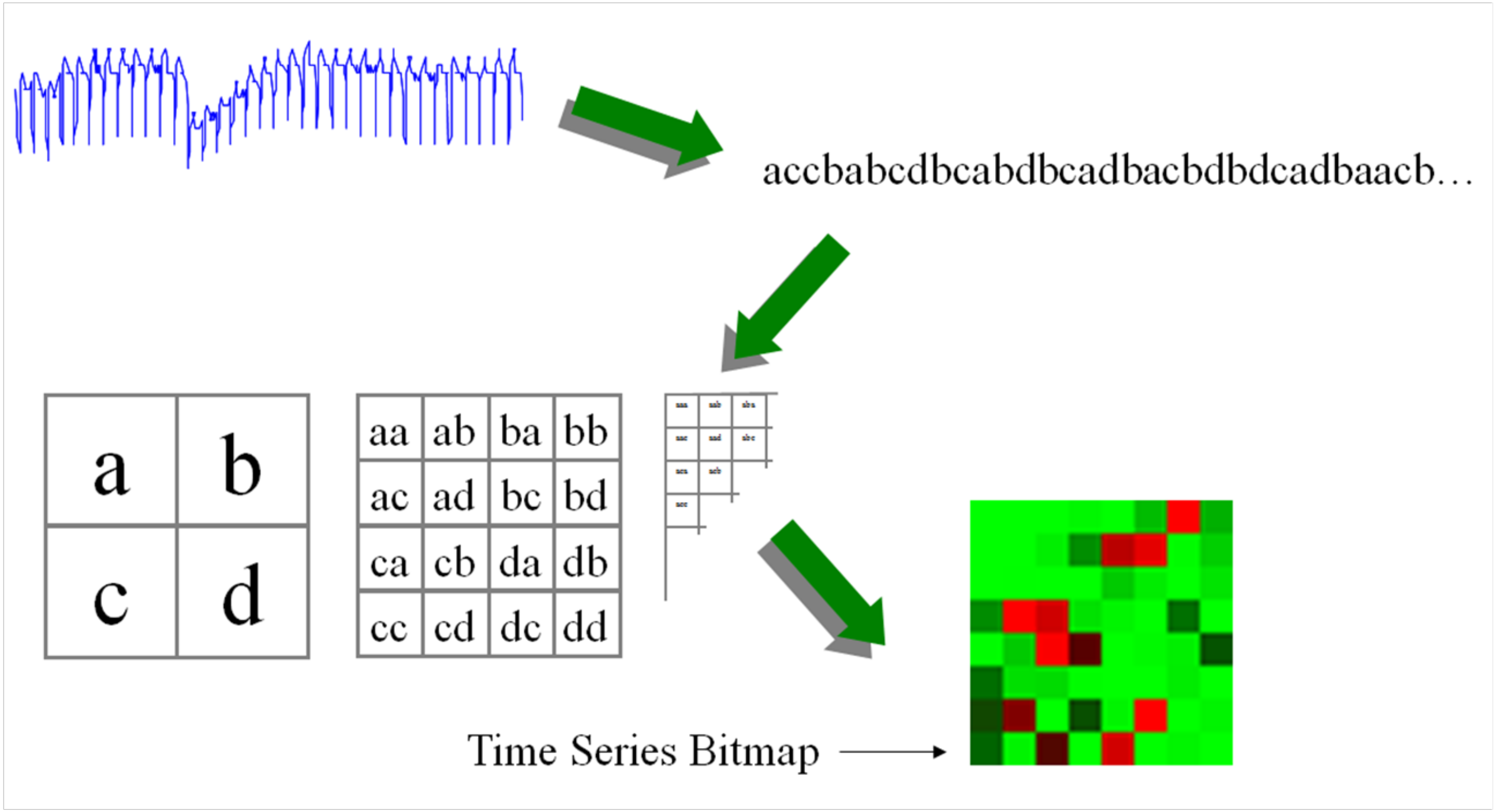


100010001010010001010101000001010
100010101110111101011010010111010
010101001110101010100101001010101
110101010010101010110101010010110
010111011110100011100001010000100
111010100011100001010101100101110
101



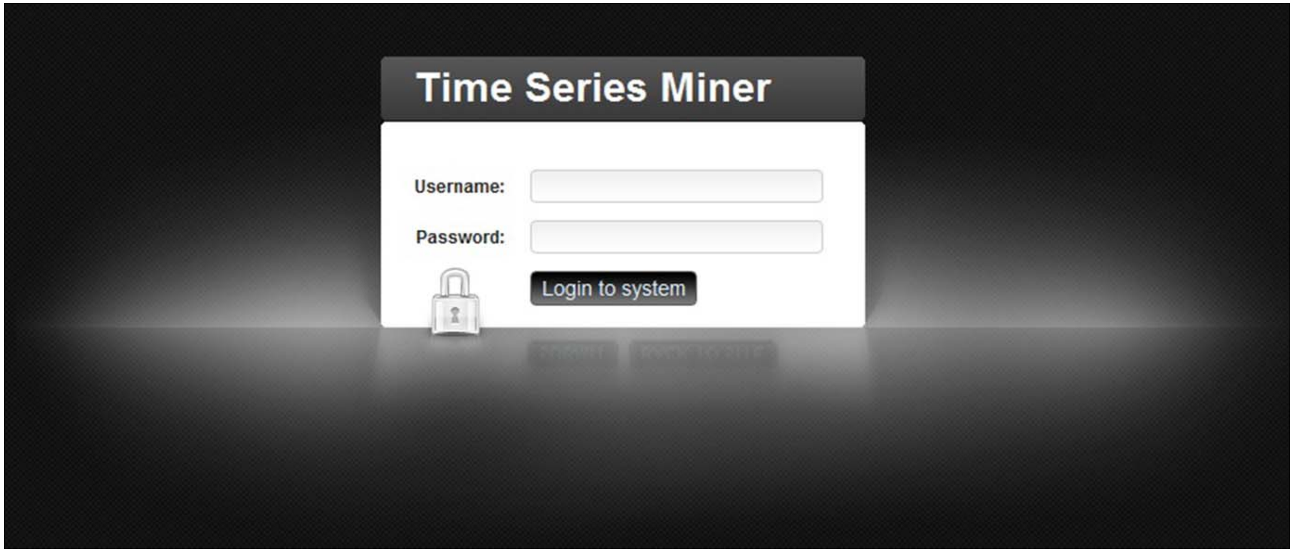
Lets put the sequences into a depth limited suffix tree, such that the frequencies of all triplets are encoded in the thickness of branches…

Visualization

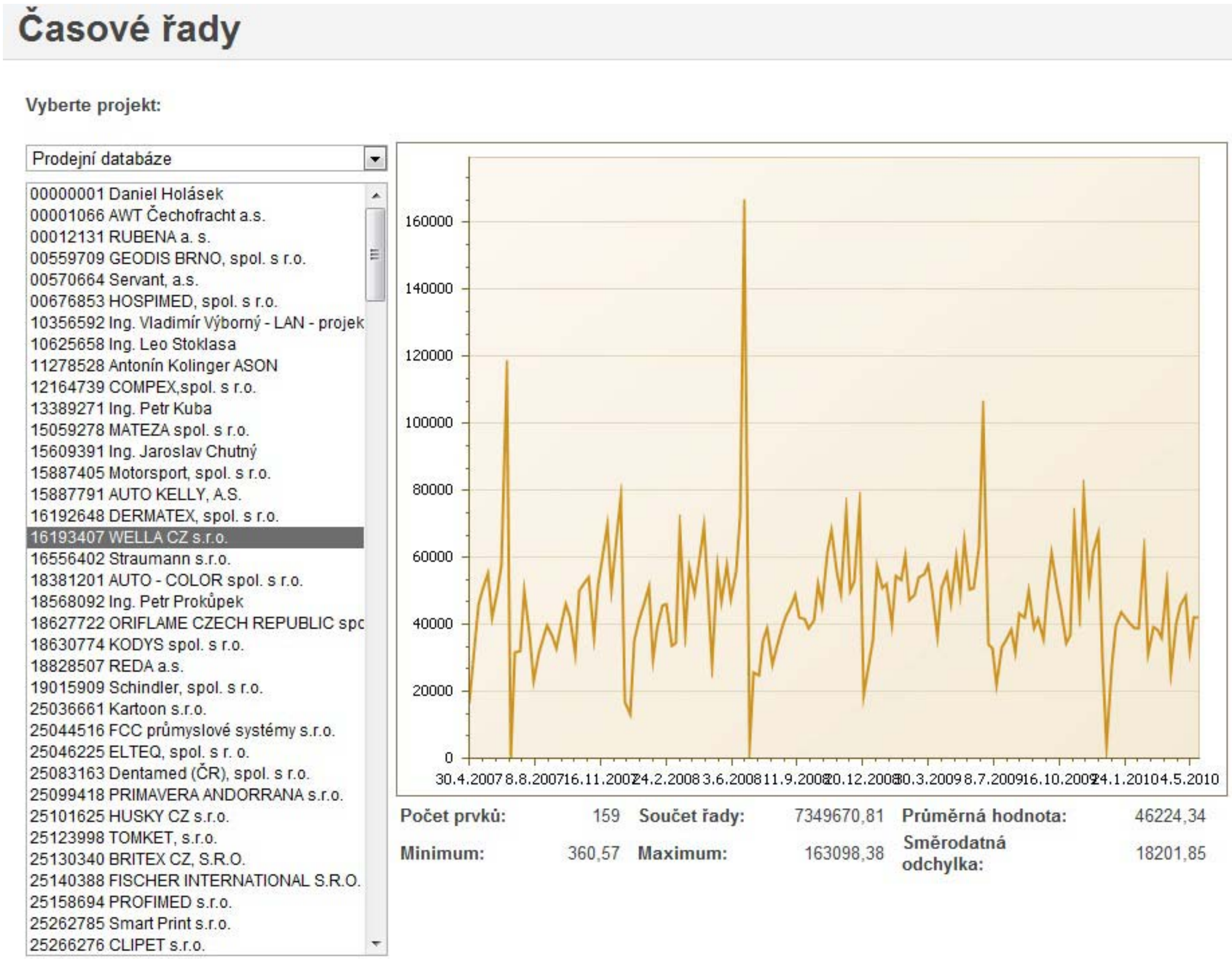


TSMiner

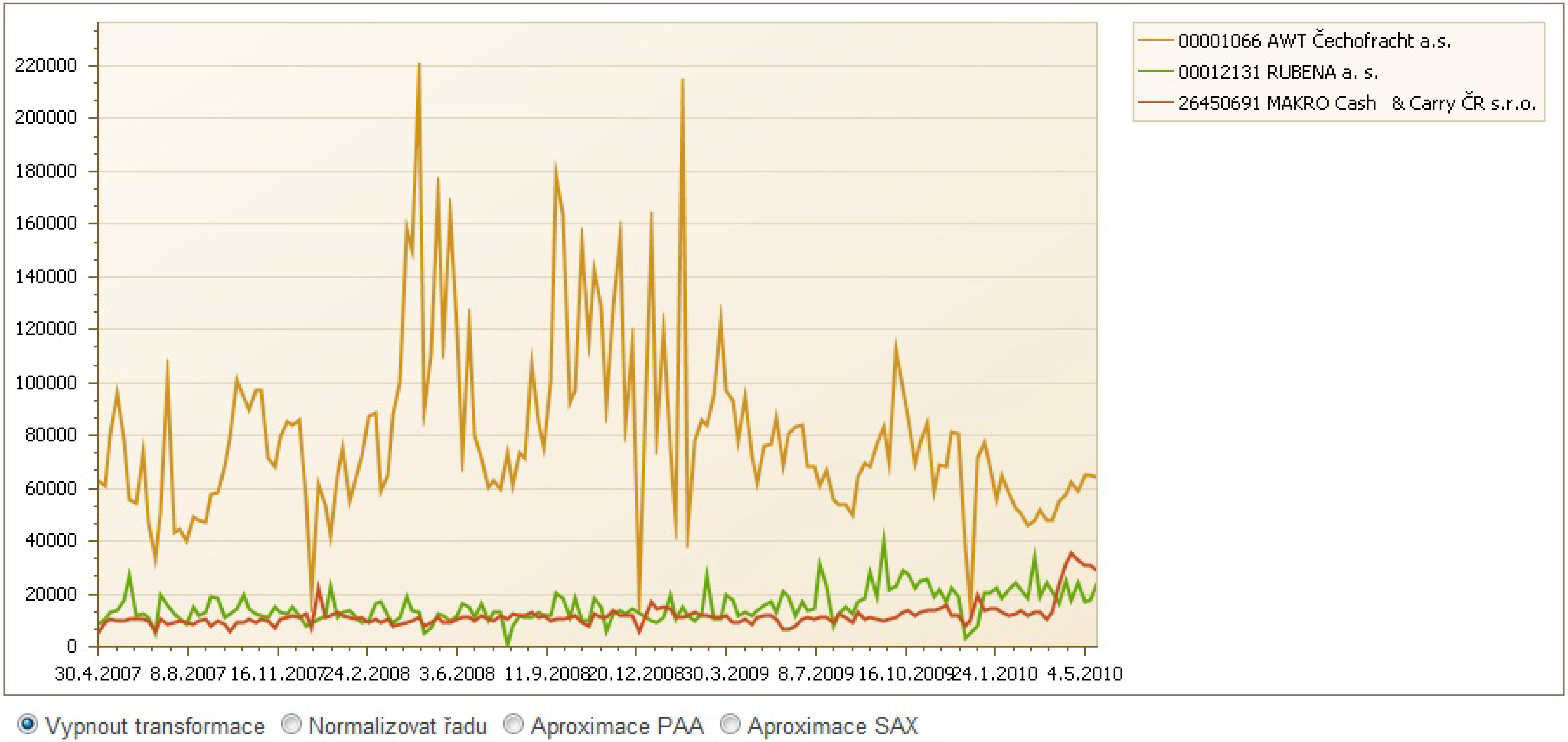
- www.tsminer.cz, www.tsminer.com, www.tsminer.eu
- Multi-tier application for time series datamining
  - MS SQL Server 2008 R2
  - ASP.NET, .NET Framework 4.0



TSMiner

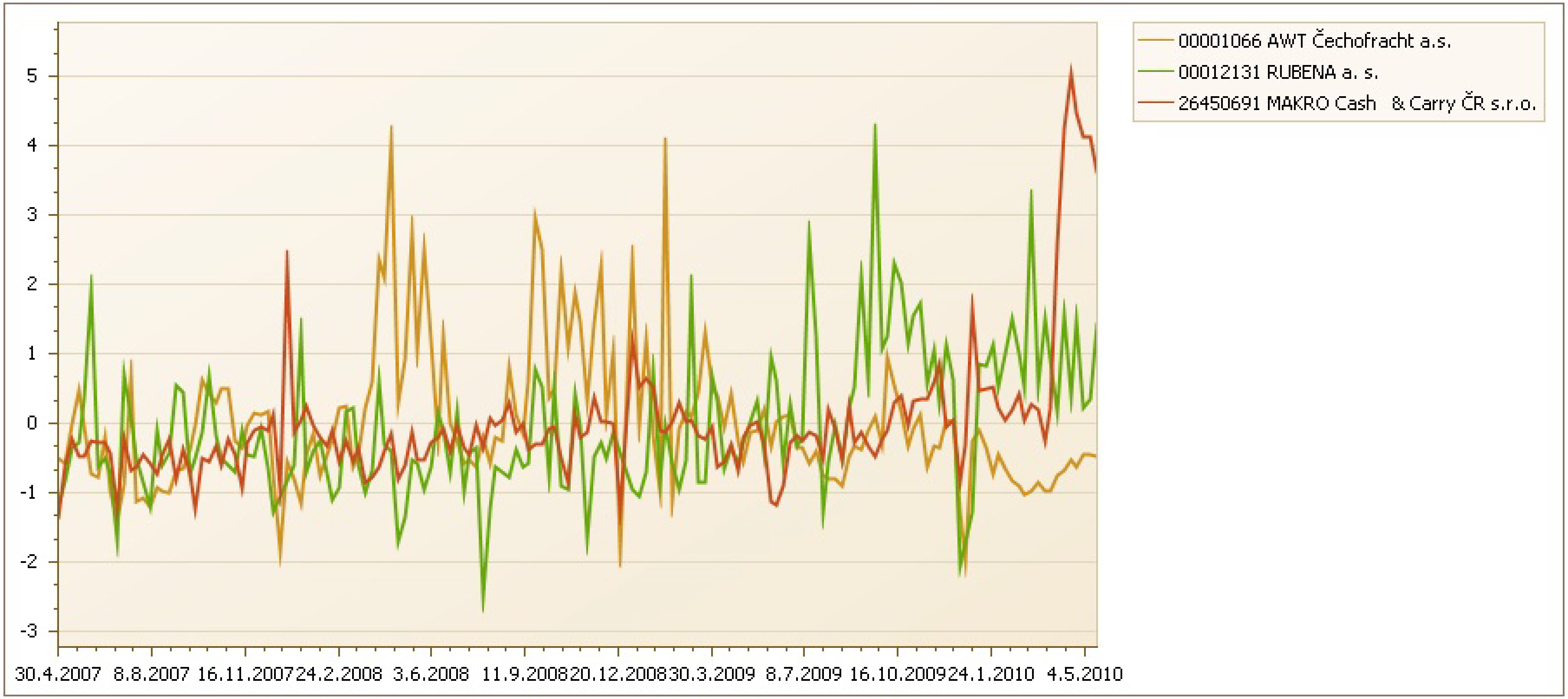

Časové řady
Vyberte projekt:
Prodejní databáze
00000001 Daniel Holásek
00001066 AWT Čechofracht a.s.
00012131 RUBENA a. s.
00559709 GEODIS BRNO, spol. s r.o.
00570664 Servant, a.s.
00676853 HOSPIMED, spol. s r.o.
10356592 Ing. Vladimír Výborný - LAN - projek
10625658 Ing. Leo Stoklasa
11278528 Antonín Kolinger ASON
12164739 COMPEX,spol. s r.o.
13389271 Ing. Petr Kuba
15059278 MATEZA spol. s r.o.
15609391 Ing. Jaroslav Chutný
15887405 Motorsport, spol. s r.o.
15887791 AUTO KELLY, A.S.
16192648 DERMATEX, spol. s r.o.
16193407 WELLA CZ s.r.o.
16556402 Straumann s.r.o.
18381201 AUTO - COLOR spol. s r.o.
18568092 Ing. Petr Prokůpek
18627722 ORIFLAME CZECH REPUBLIC spc
18630774 KODYS spol. s r.o.
18828507 REDA a.s.
19015909 Schindler, spol. s r.o.
25036661 Kartoon s.r.o.
25044516 FCC průmyslové systémy s.r.o.
25046225 ELTEQ, spol. s r. o.
25083163 Dentamed (ČR), spol. s r.o.
25099418 PRIMAVERA ANDORRANA s.r.o.
25101625 HUSKY CZ s.r.o.
25123998 TOMKET, s.r.o.
25130340 BRITEX CZ, S.R.O.
25140388 FISCHER INTERNATIONAL S.R.O.
25158694 PROFIMED s.r.o.
25262785 Smart Print s.r.o.
25266276 CLIPET s.r.o.
160000
140000
120000
100000
80000
60000
40000
20000
0
30.4.2007
8.8.2007
16.11.2007
24.2.2008
3.6.2008
11.9.2008
20.12.2008
30.3.2009
8.7.2009
16.10.2009
24.1.2010
4.5.2010
Počet prvků: 159
Součet řady: 7349670,81
Průměrná hodnota: 46224,34
Minimum: 360,57
Maximum: 163098,38
Směrodatná odchylka: 18201,85

TSMiner



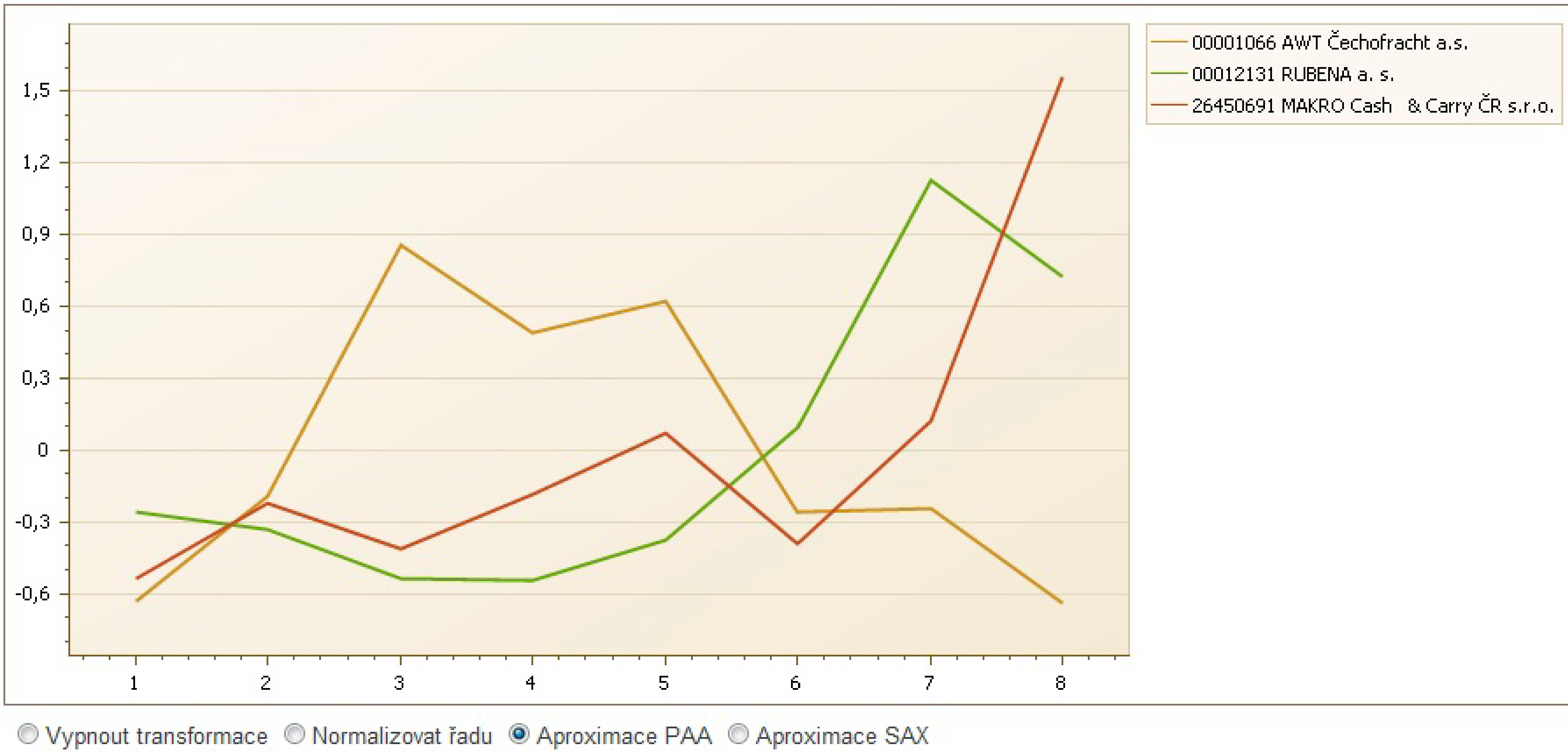
220000
200000
180000
160000
140000
120000
100000
80000
60000
40000
20000
0
30.4.2007
8.8.2007
16.11.2007
24.2.2008
3.6.2008
11.9.2008
20.12.2008
30.3.2009
8.7.2009
16.10.2009
24.1.2010
4.5.2010
00001066 AWT Čechofracht a.s.
00012131 RUBENA a. s.
26450691 MAKRO Cash & Carry ČR s.r.o.
Vypnout transformace
Normalizovat řadu
Aproximace PAA
Aproximace SAX

TSMiner



Vypnout transformace
Normalizovat řadu
Aproximace PAA
Aproximace SAX

TSMiner


00001066 AWT Čechofracht a.s.
00012131 RUBENA a. s.
26450691 MAKRO Cash & Carry ČR s.r.o.
1,5
1,2
0,9
0,6
0,3
0
-0,3
-0,6
1
2
3
4
5
6
7
8
Vypnout transformace
Normalizovat řadu
Aproximace PAA
Aproximace SAX

TSMiner

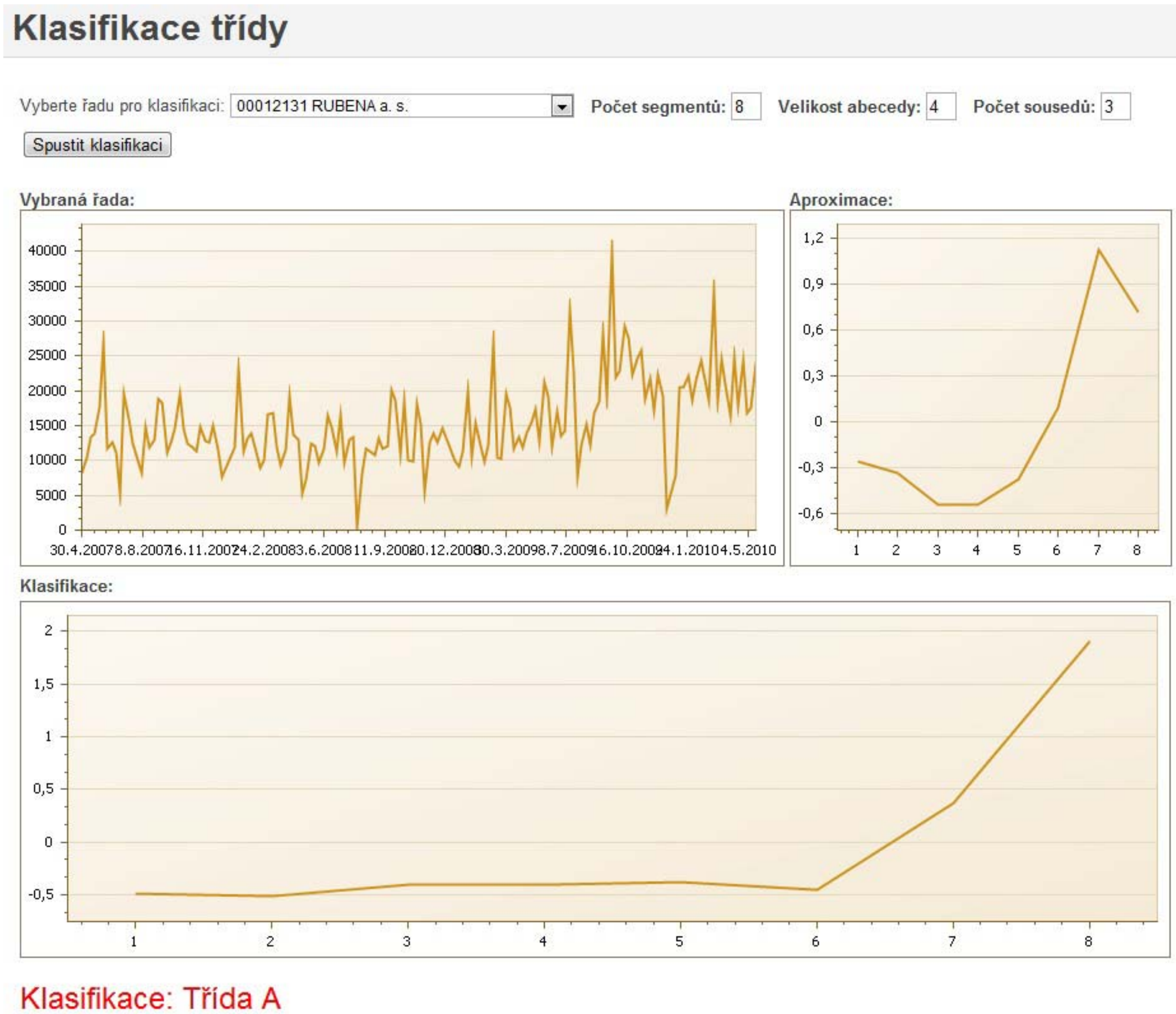
Vypnout transformace | Normalizovat řadu | Aproximace PAA | Aproximace SAX

Počet segmentů: 8

Velikost abecedy: 4

00001066 AWT Čechofracht a.s.	BBDCCBBB
00012131 RUBENA a. s.	BBBBBCDD
26450691 MAKRO Cash & Carry ČR s.r.o.	BBBBCBCD

TSMiner



Klasifikace třídy
Vyberte řadu pro klasifikaci:
00012131 RUBENA a. s.
Počet segmentů: 8
Velikost abecedy: 4
Počet sousedů: 3
Spustit klasifikaci
Vybraná řada:
Aproximace:
Klasifikace:
Klasifikace: Třída A

Main problems of time series analysis

- Pattern search w/o prior parameter setting
- Clustering of streamed data
- Time series merging – finding all shared subsequences
- „Why“ analysis in classification and clustering, automatic generation of explanation
- Weighed representation of time series
- Visualization of large time series

(This slide was translated to English by V. Svátek)

Thanks for your attention