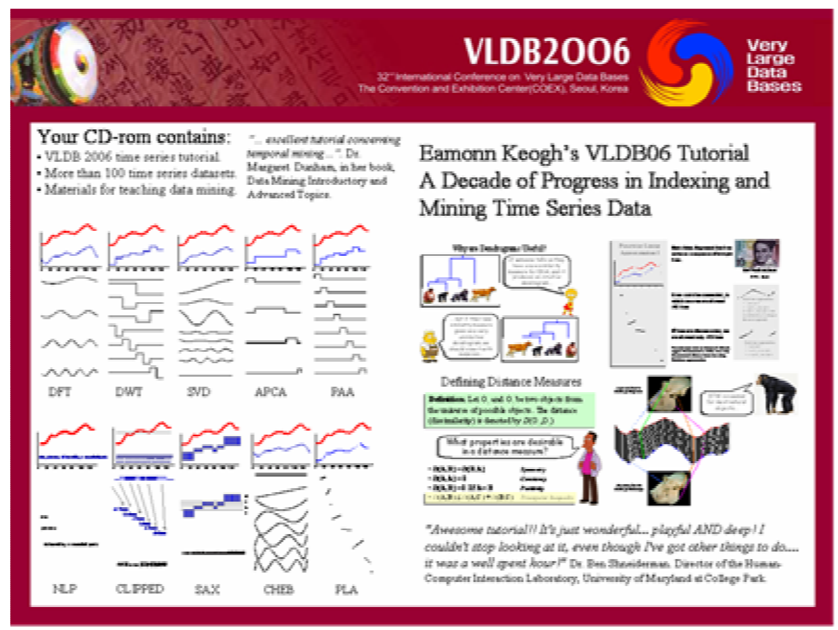
Introduction to Time Series Mining

Slides from Keogh Eamonn's tutorial:



Outline of Tutorial

- Introduction, Motivation
- The Utility of Similarity Measurements
  - Properties of distance measures
  - The Euclidean distance
  - Preprocessing the data
  - Dynamic Time Warping
  - Uniform Scaling

- Data Mining
  - Anomaly/Interestingness detection
  - Motif (repeated pattern) discovery

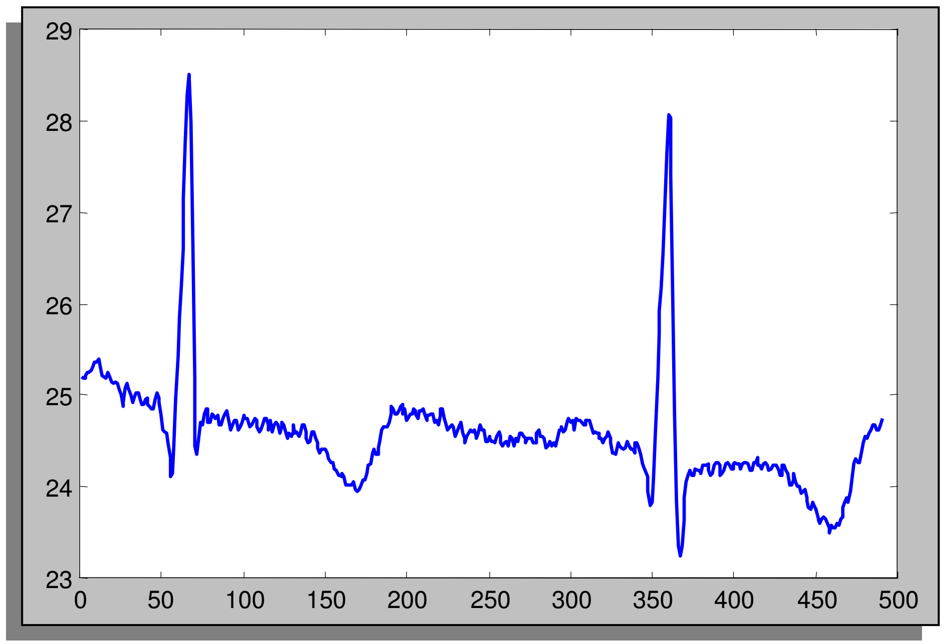
25.1750
25.2250
25.2500
25.2500
25.2750
25.3250
25.3500
25.3500
25.4000
25.4000
25.3250
25.2250
25.2000
25.1750

..

..
24.6250
24.6750
24.6750
24.6250
24.6250
24.6250
24.6750
24.7500

What are Time Series?

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



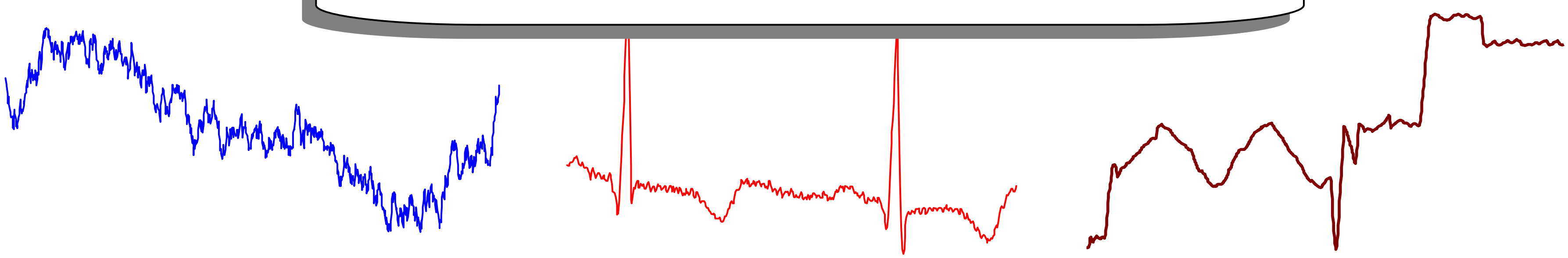
Virtually all similarity measurements, indexing and dimensionality reduction techniques discussed in this tutorial can be used with other data types

Time Series are Ubiquitous! I

People measure things...

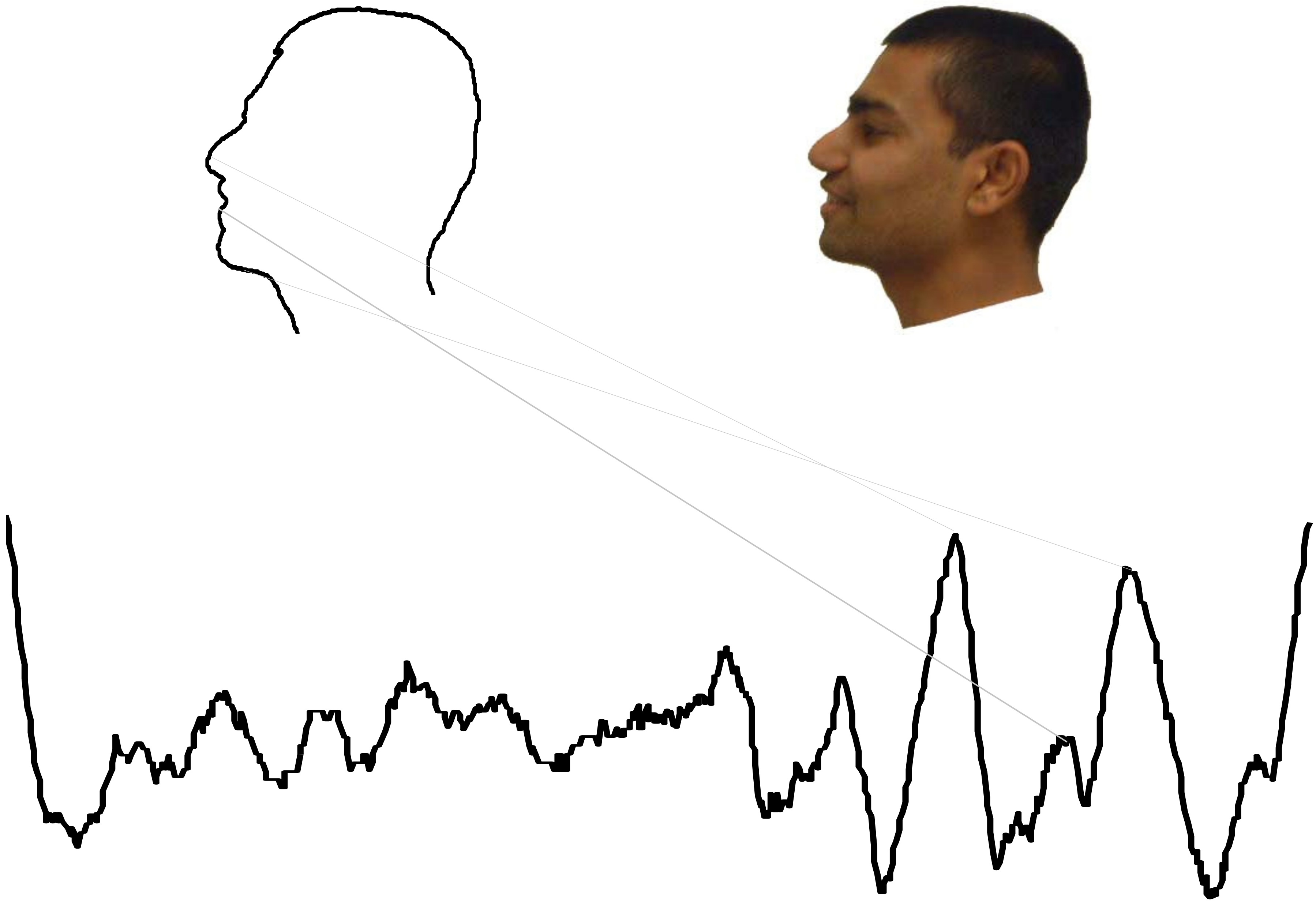
- *Their blood pressure*
- *George Bush's popularity rating*
- *The annual rainfall in Seattle*
- *The value of their Google stock*

...and things change over time...



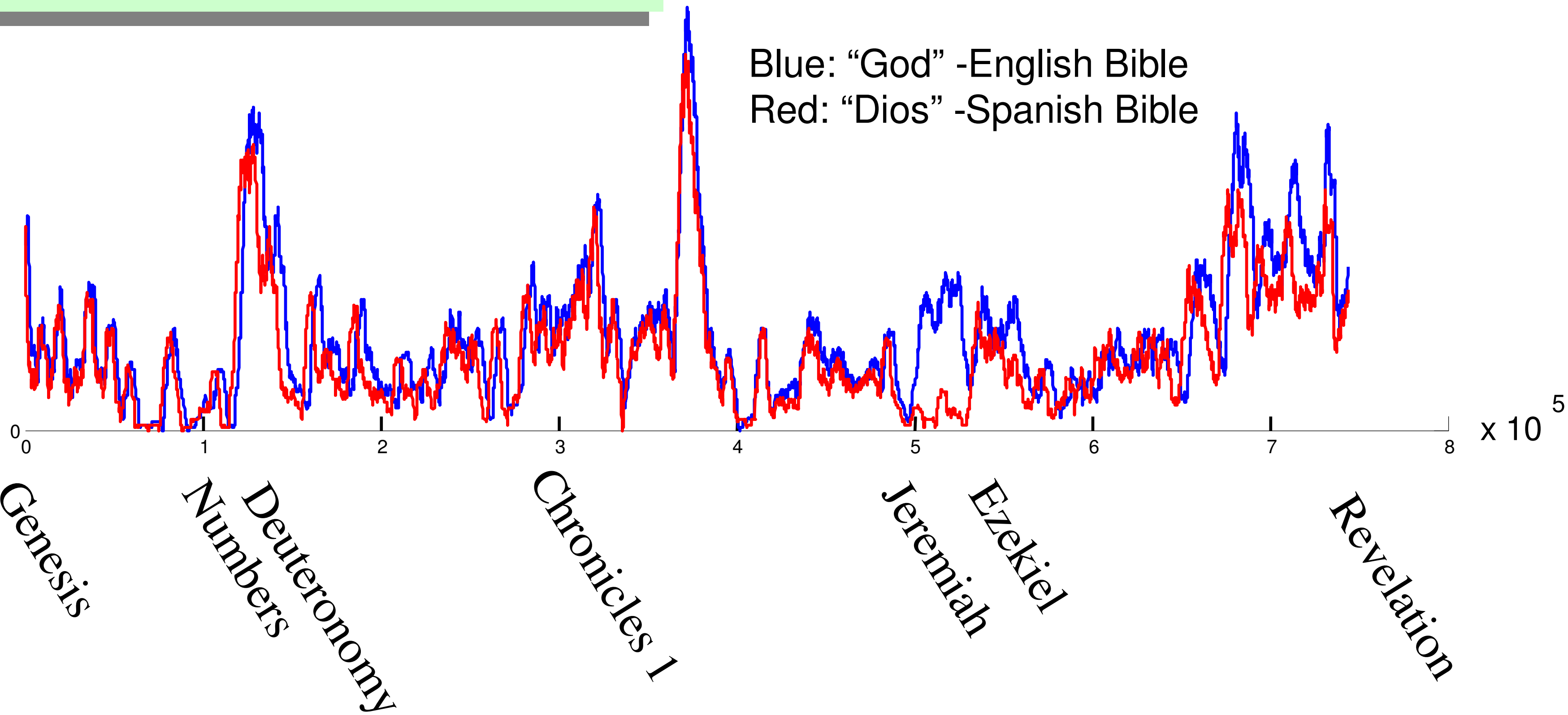
Thus time series occur in virtually every medical, scientific and businesses domain

Image data, may best be thought of as time series…

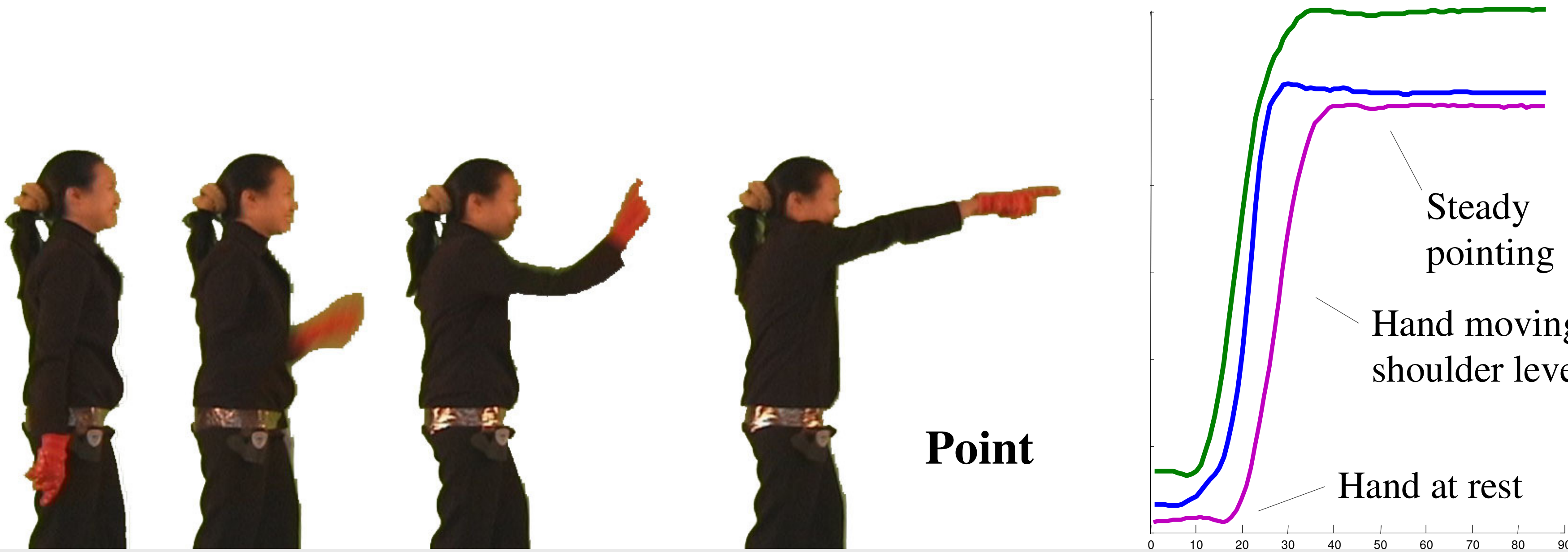


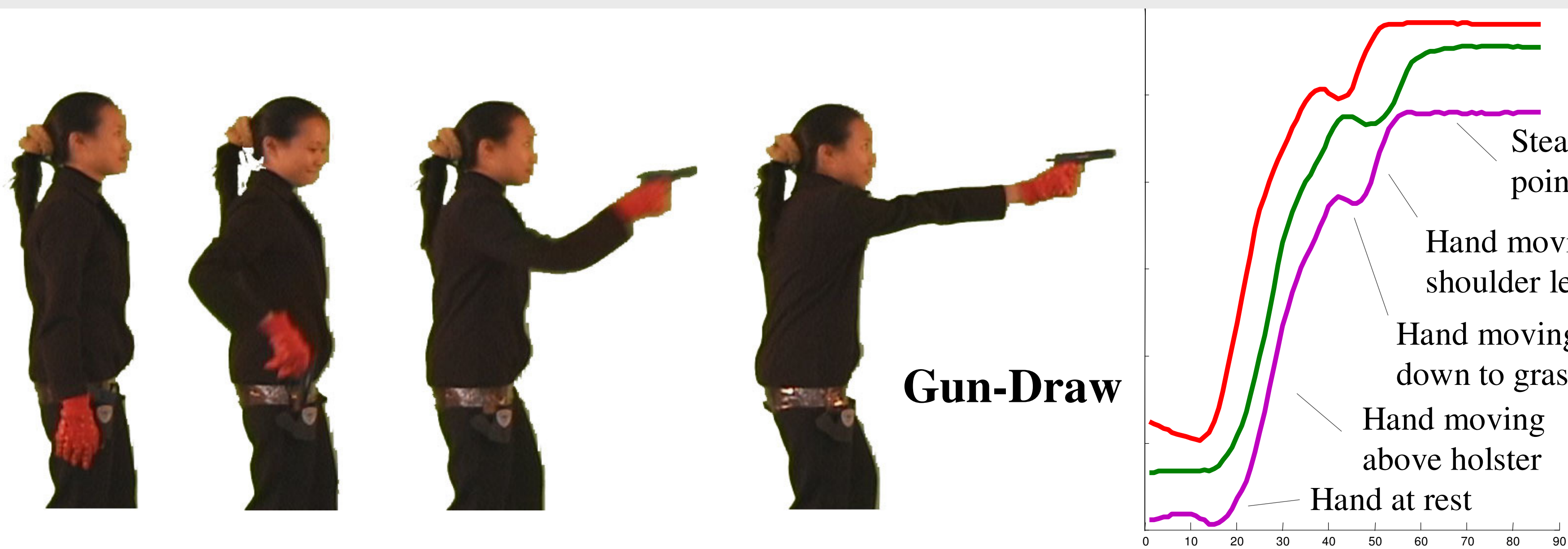
Text data, may best be thought of as time series…

The local frequency of words in the Bible



Video data, may best be thought of as time series…





Why is Working With Time Series so Difficult? Part I

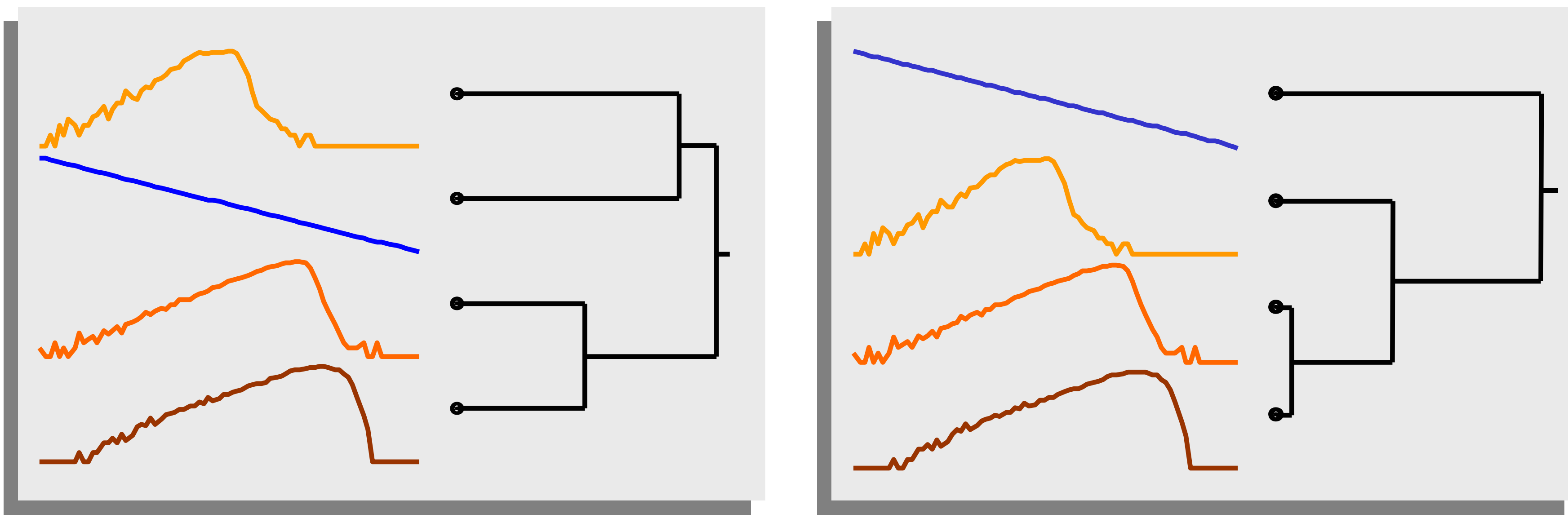
Answer: How do we work with very large databases?

- 1 Hour of EKG data: 1 Gigabyte.
- Typical Weblog: 5 Gigabytes per week.
- Space Shuttle Database: 200 Gigabytes and growing.
- Macho Database: 3 Terabytes, updated with 3 gigabytes a day.

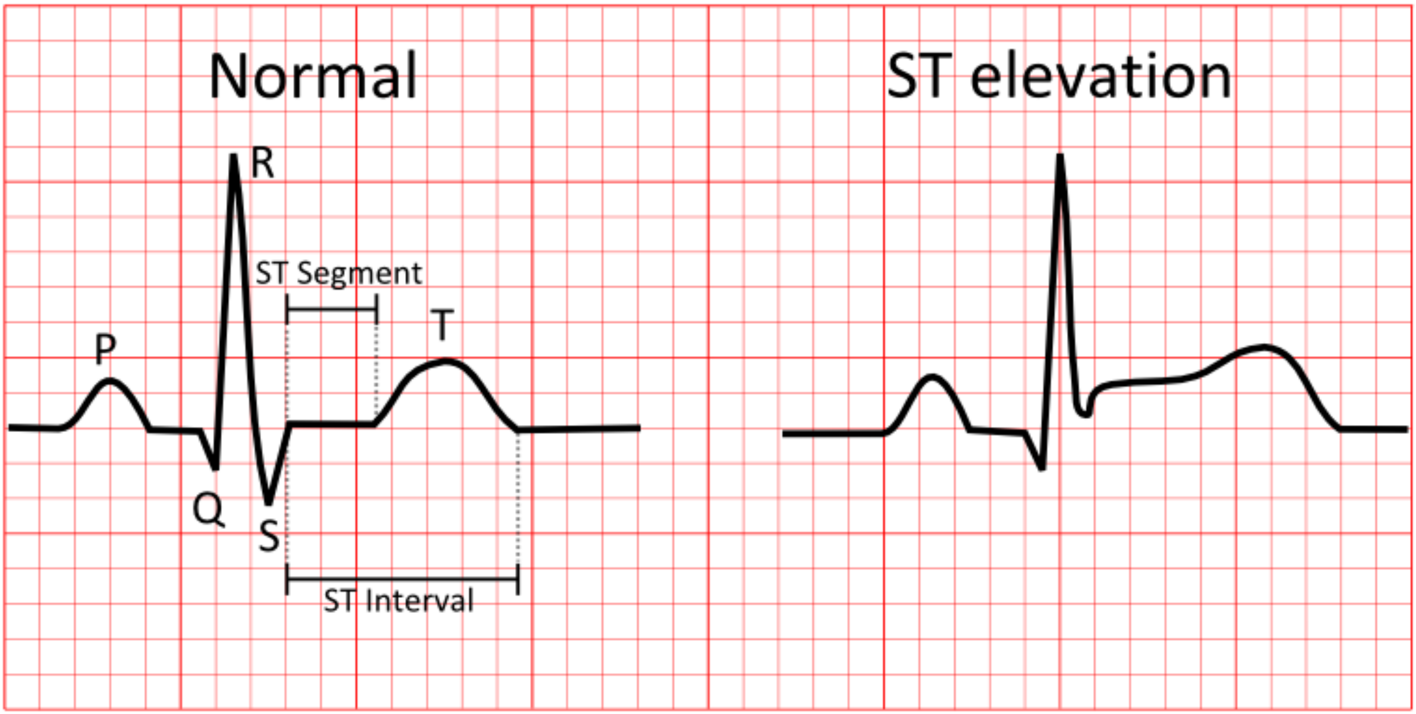
Since most of the data lives on disk (or tape), we need a representation of the data we can efficiently manipulate.

Why is Working With Time Series so Difficult? Part II

Answer: We are dealing with subjectivity



The definition of similarity depends on the user, the domain and the task at hand. We need to be able to handle this subjectivity.


Normal
R
ST Segment
T
P
Q
S
ST Interval
ST elevation

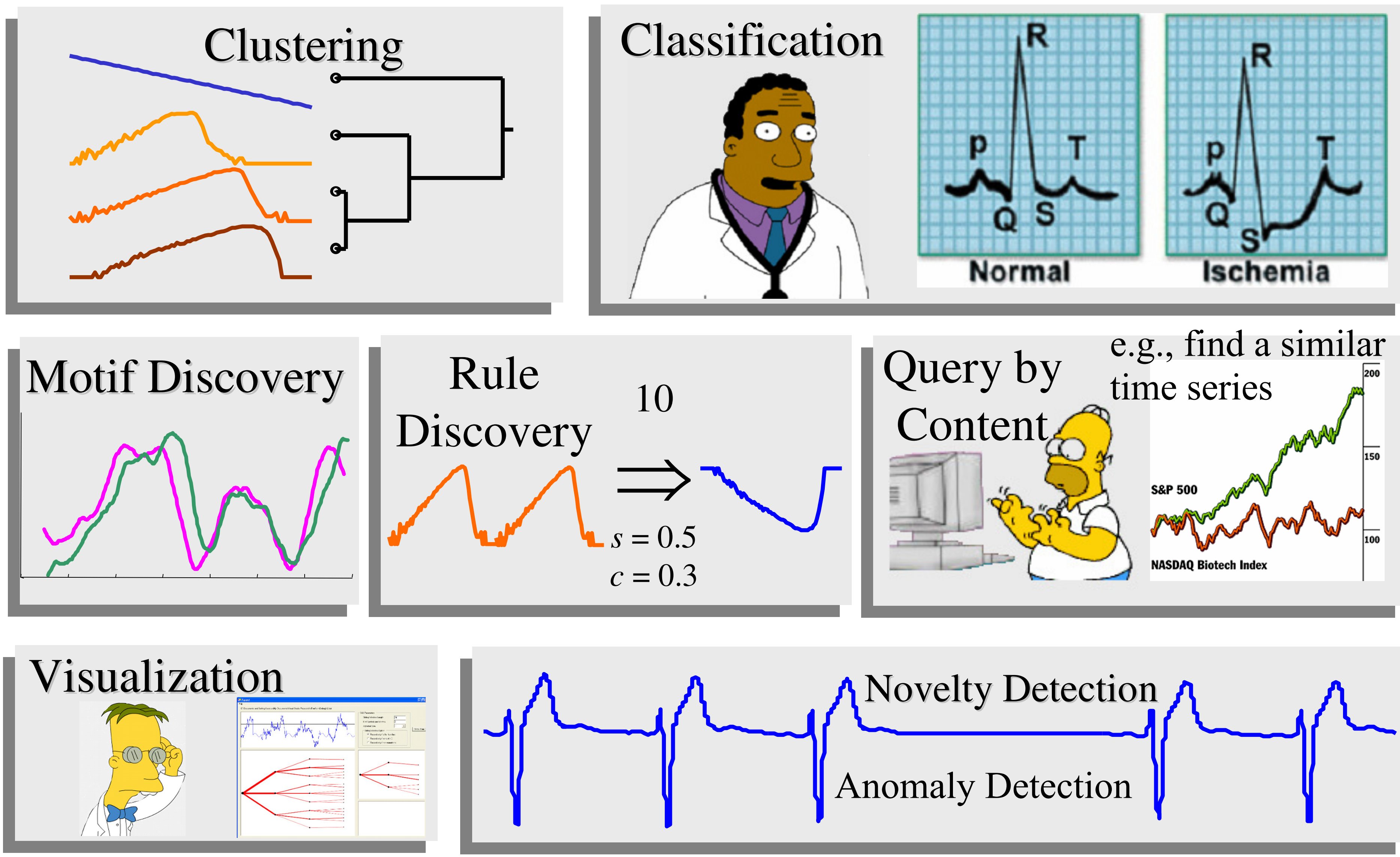
Why is working with time series so difficult? Part III

Answer: Miscellaneous data handling problems.

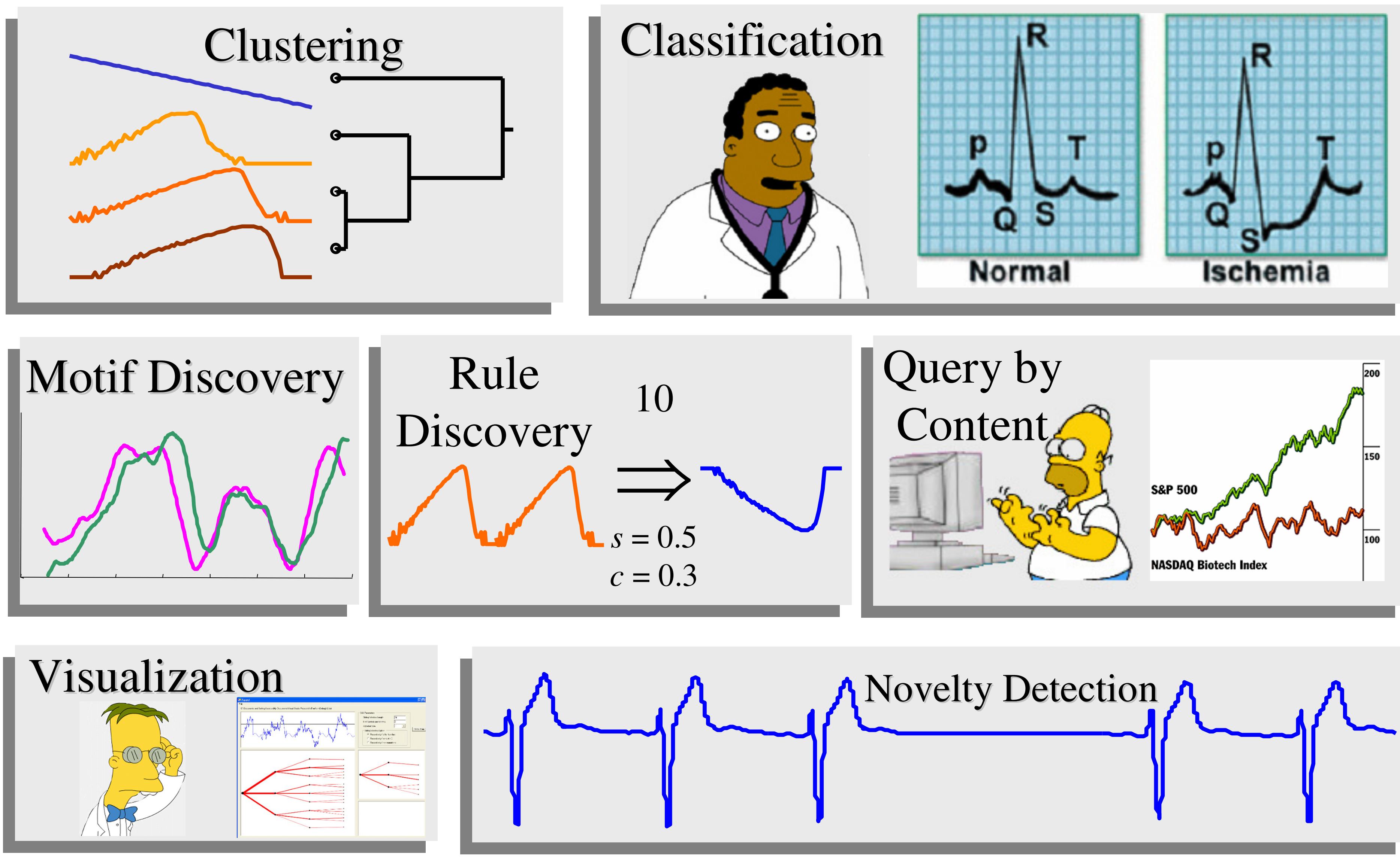
- Differing data formats.
- Differing sampling rates.
- Noise, missing values, etc.

We will not focus on these issues in this tutorial.

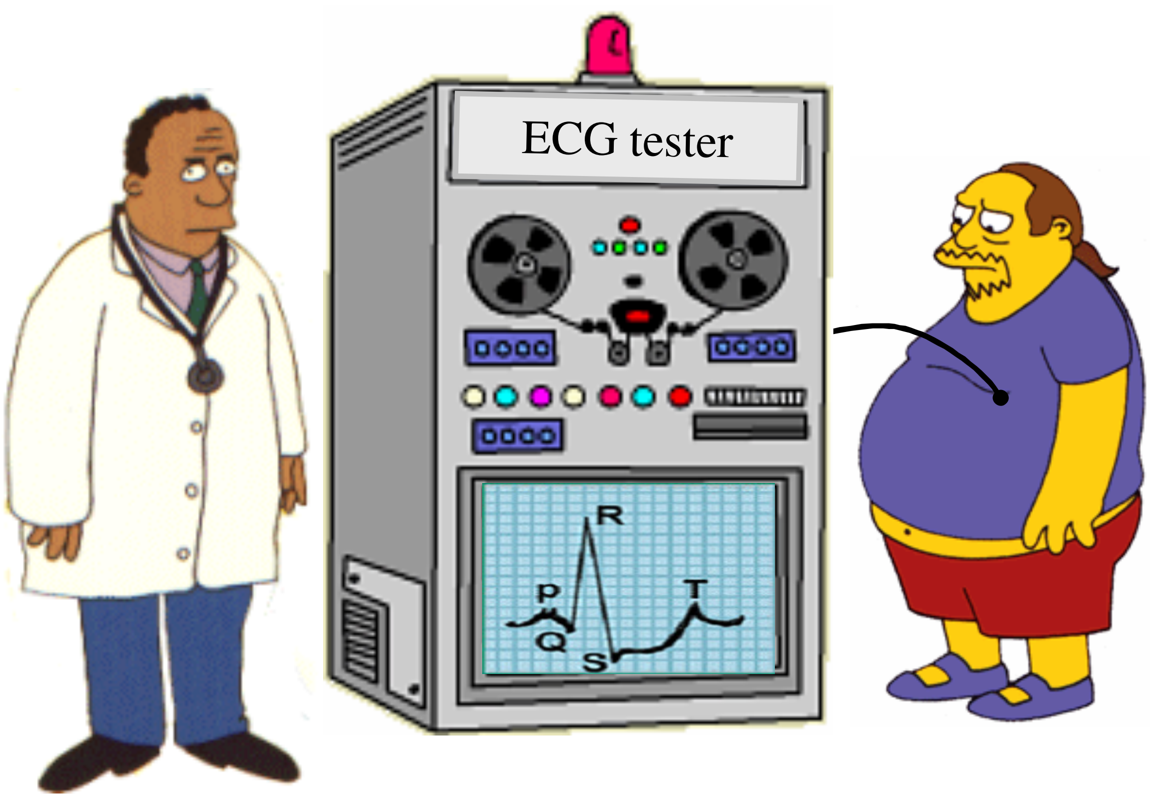
What do we want to do with the time series data?



All these problems require <u>similarity</u> matching



Here is a simple motivation for the first part of the tutorial



You go to the doctor because of chest pains. Your ECG looks strange…

You doctor wants to search a database to find **similar** ECGs, in the hope that they will offer clues about your condition...

- **How do we define similar?**

What is Similarity?

The quality or state of being similar; likeness; resemblance; as, a similarity of features. **Webster's Dictionary**

Similarity is hard to define, but…
"*We know it when we see it*"

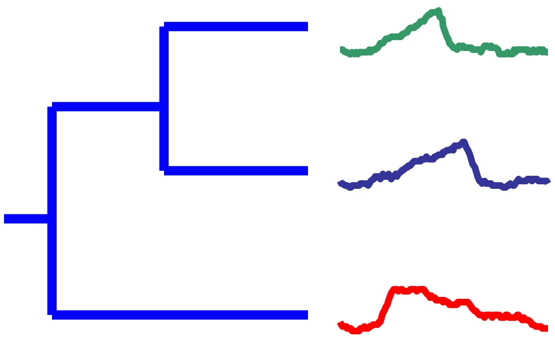
The real meaning of similarity is a philosophical question.

We will take a more pragmatic approach.

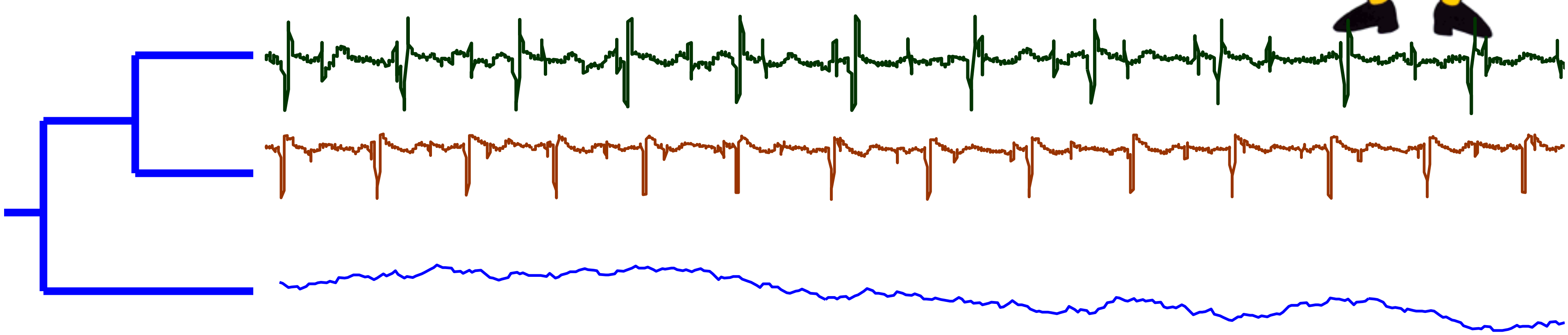
Two Kinds of Similarity

time series

Similarity at the level of *shape*
Next 40 minutes



Similarity at the *structural* level
Another 10 minutes



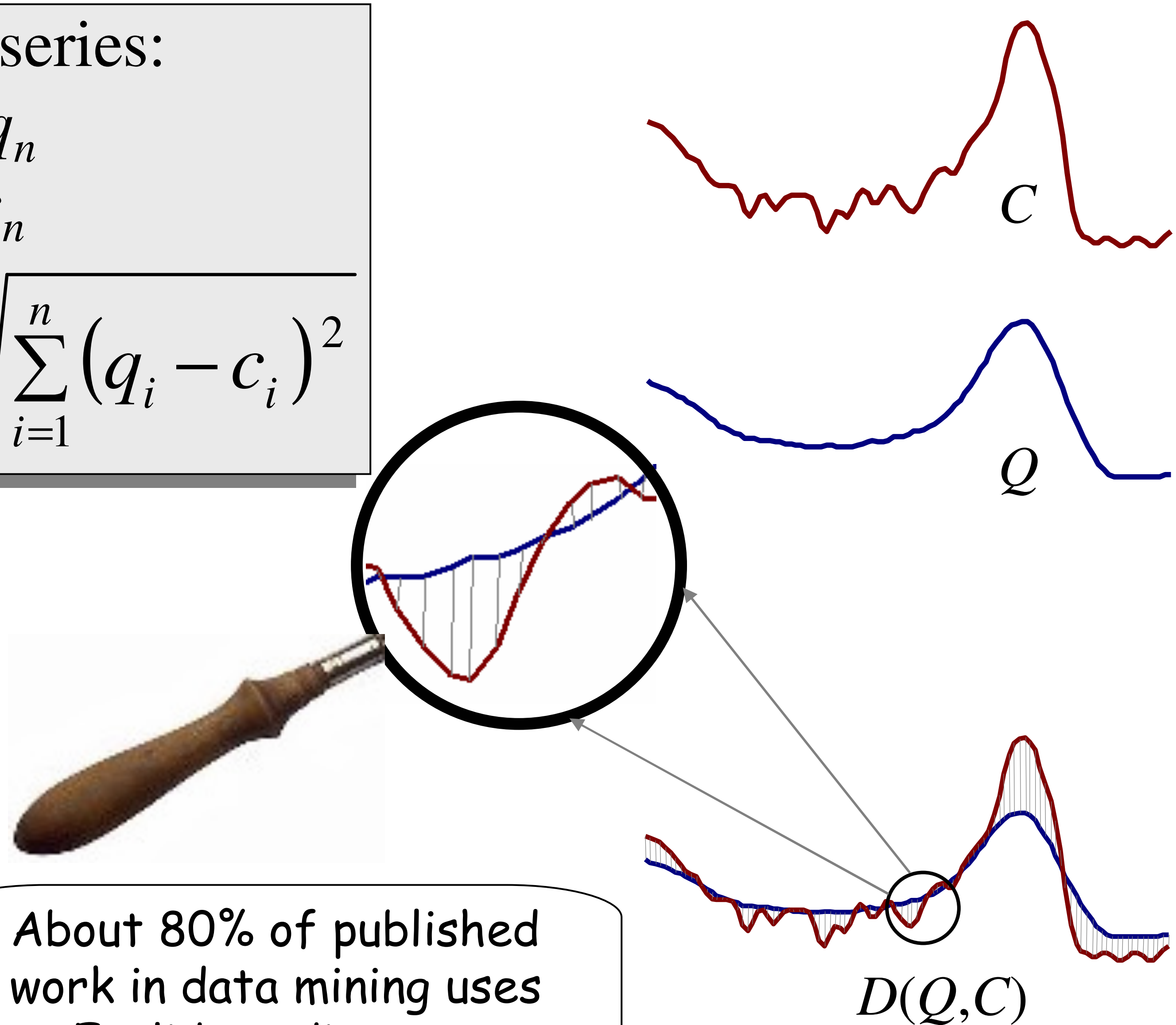
Euclidean Distance Metric

Given two time series:

$$Q = q_1 \ldots q_n$$

$$C = c_1 \ldots c_n$$

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



About 80% of published work in data mining uses Euclidean distance

Preprocessing the data before distance calculations

If we naively try to measure the distance between two "raw" time series, we may get very unintuitive results

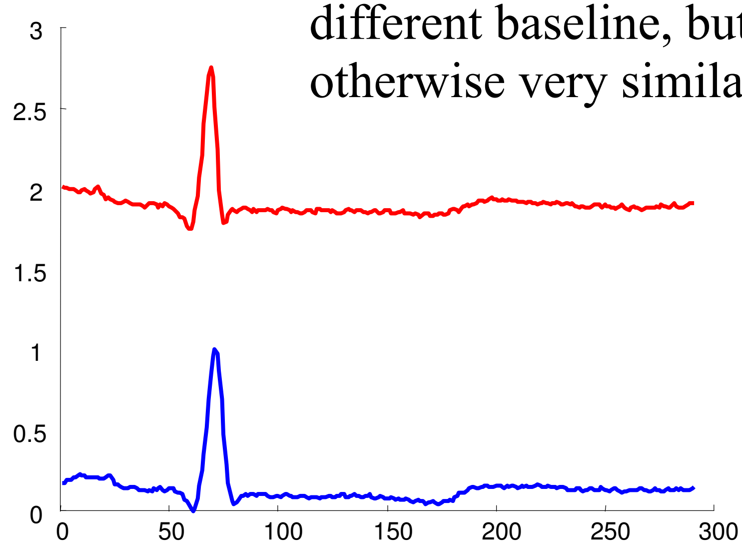
This is because Euclidean distance is very sensitive to some "distortions" in the data. For most problems these distortions are not meaningful, and thus we can and should remove them

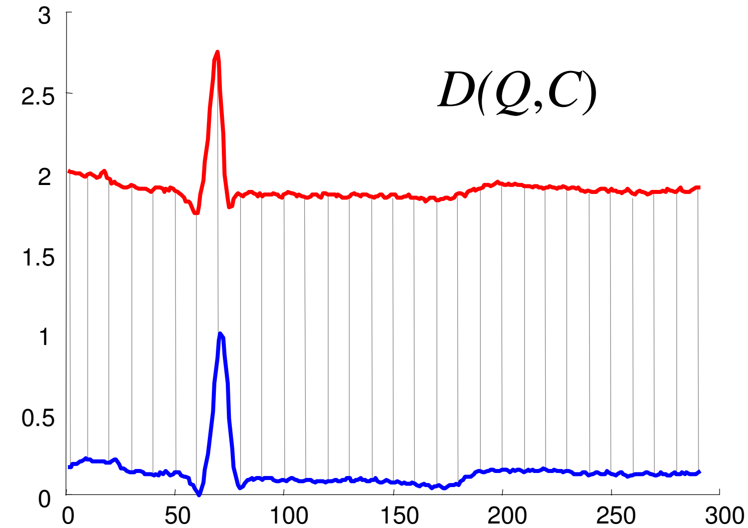
In the next few slides we will discuss the 4 most common distortions, and how to remove them

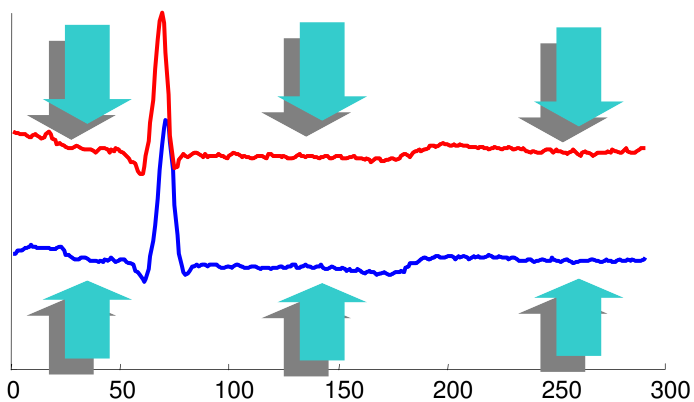
- Offset Translation
- Amplitude Scaling
- Linear Trend
- Noise

Transformation I: Offset Translation

The two time series have a different baseline, but are otherwise very similar



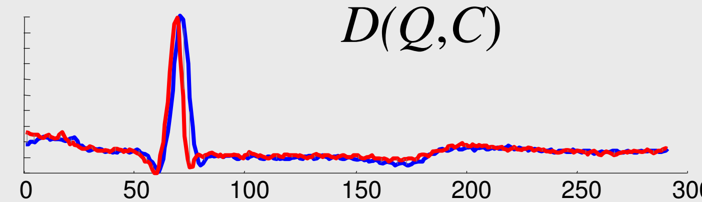




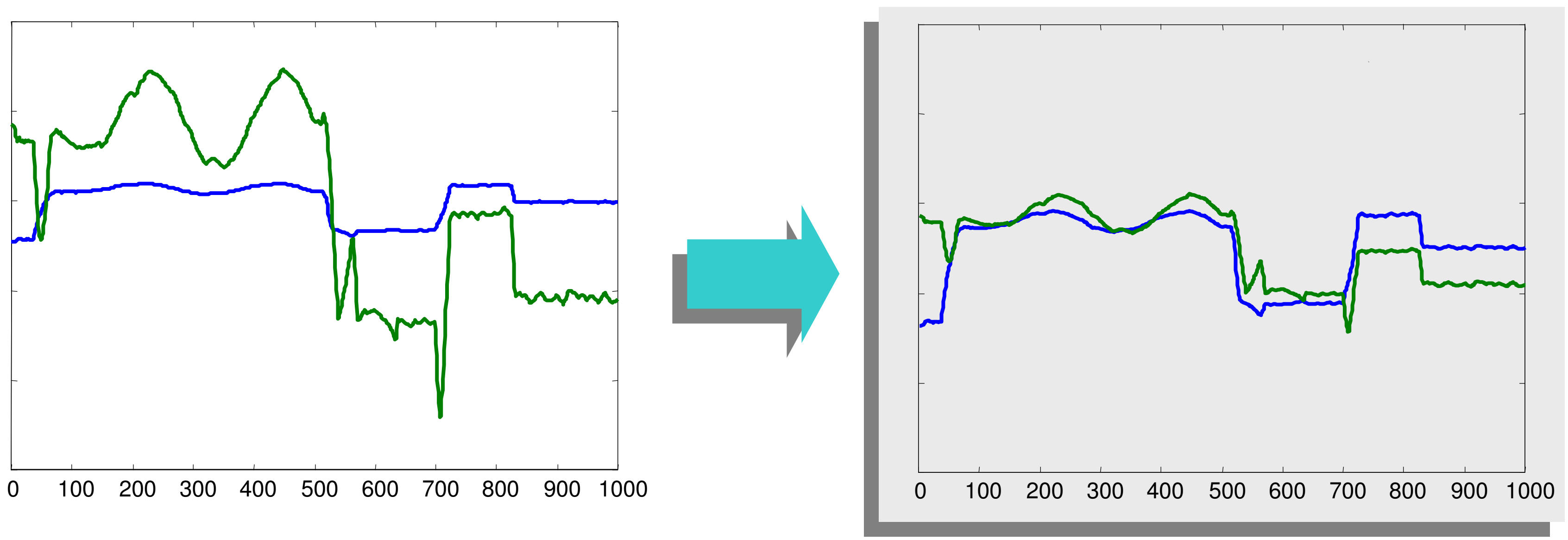
$$Q = Q - \text{mean}(Q)$$

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

$$D(Q,C)$$



Transformation II: Amplitude Scaling

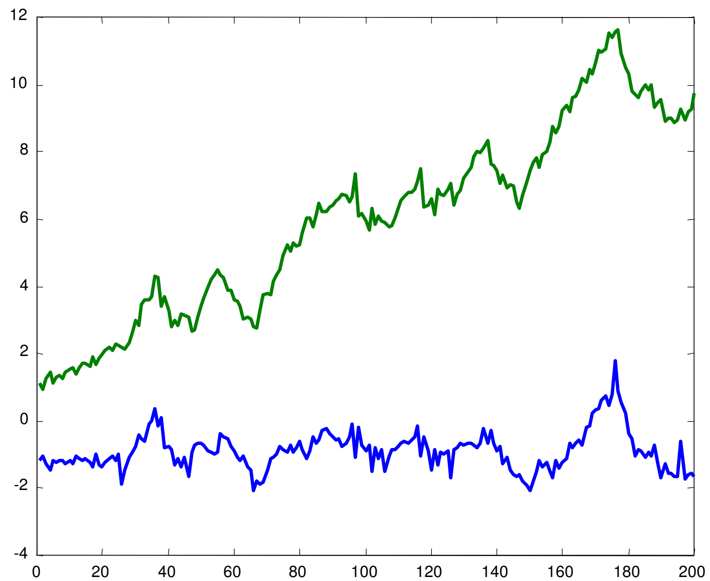


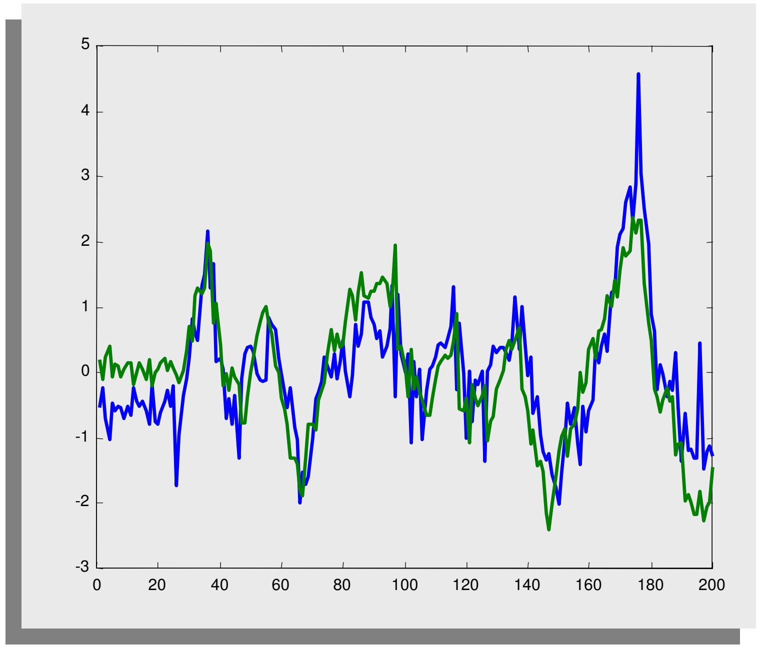
Q = (Q - mean(Q)) / std(Q)

C = (C - mean(C)) / std(C)

$D(Q,C)$

Transformation III: Linear Trend





The intuition behind removing linear trend is…

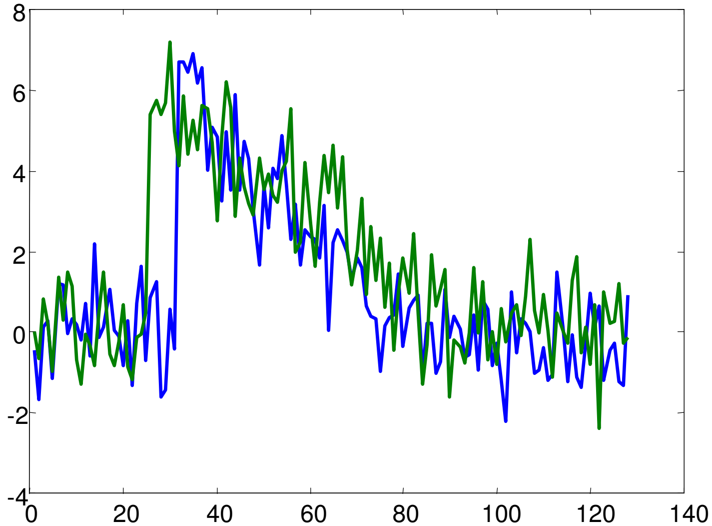
Fit the best fitting straight line to the time series, then subtract that line from the time series.

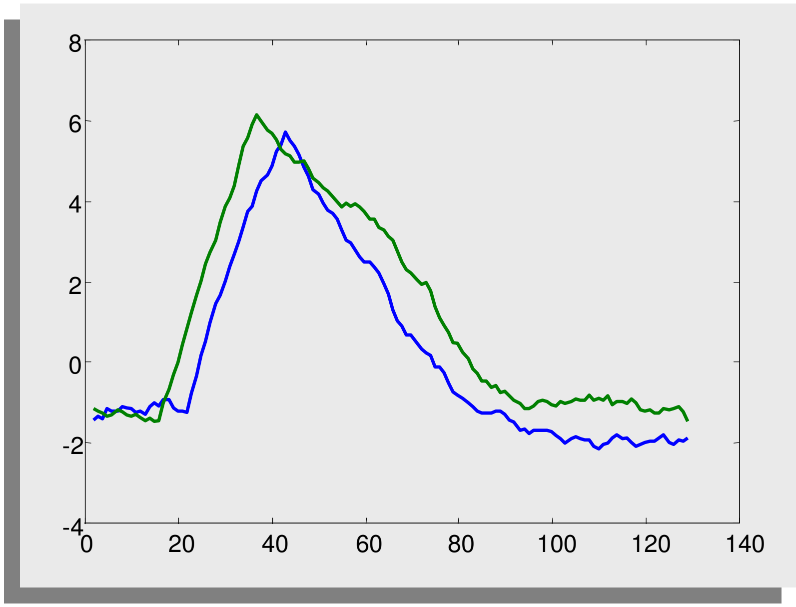
Removed **linear trend**

Removed offset translation

Removed amplitude scaling

Transformation IIII: Noise





The intuition behind removing noise is...

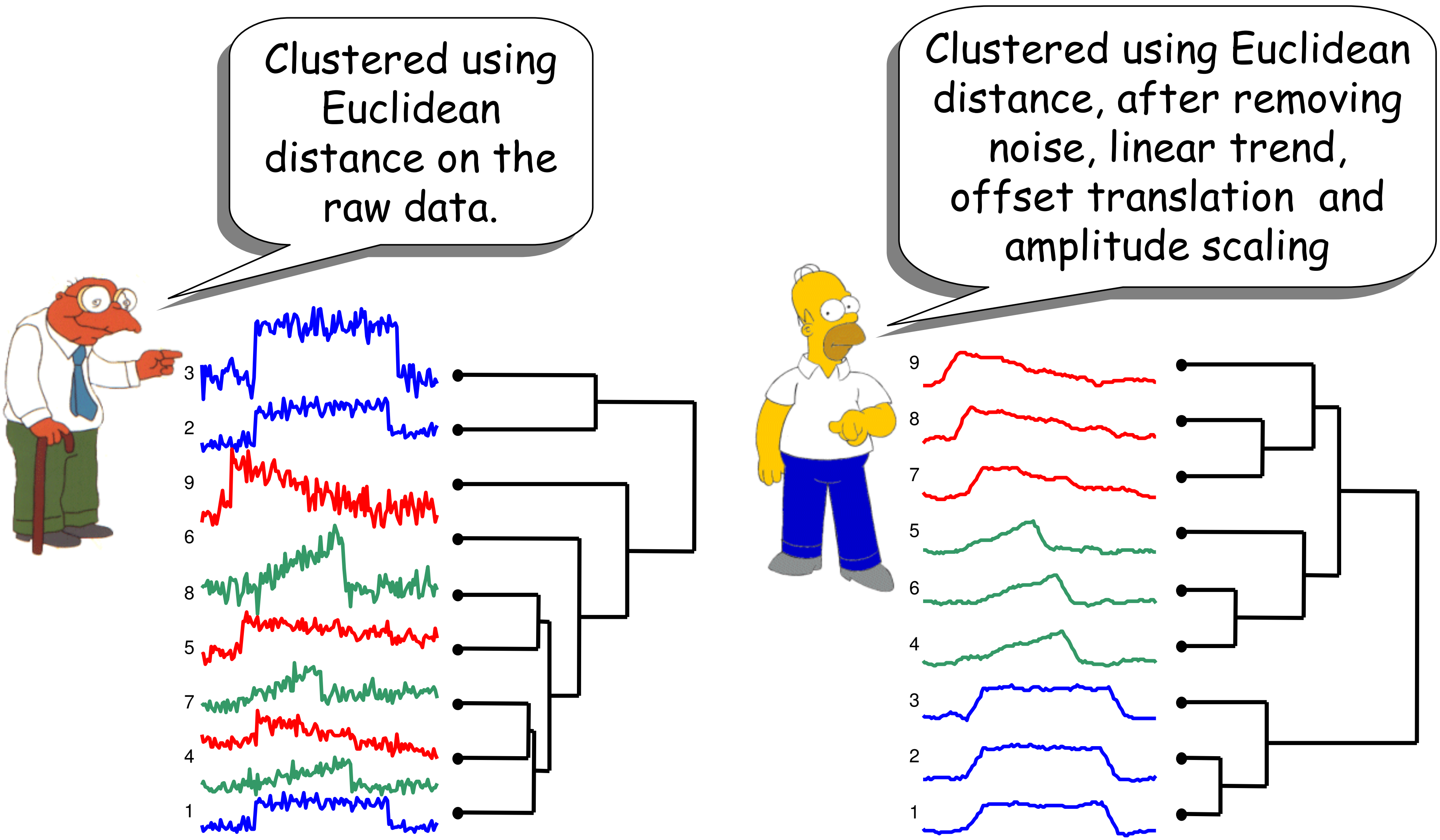
Average each datapoints value with its neighbors.

Q = smooth(Q)

C = smooth(C)

$D(Q,C)$

A Quick Experiment to Demonstrate the Utility of Preprocessing the Data



Summary of Preprocessing

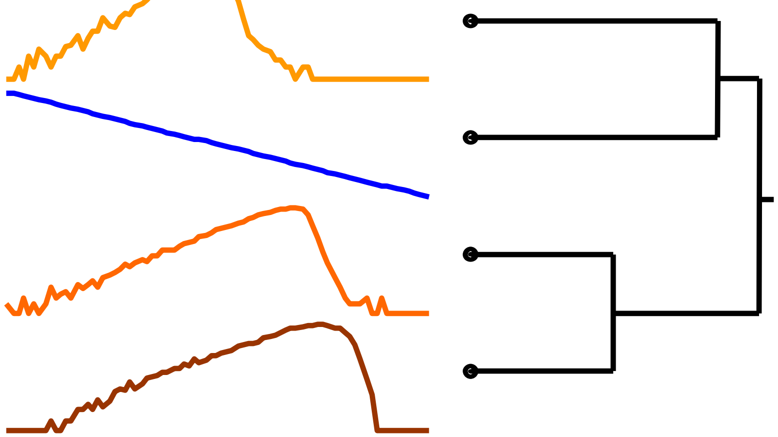
The "raw" time series may have distortions which we should remove before clustering, classification etc

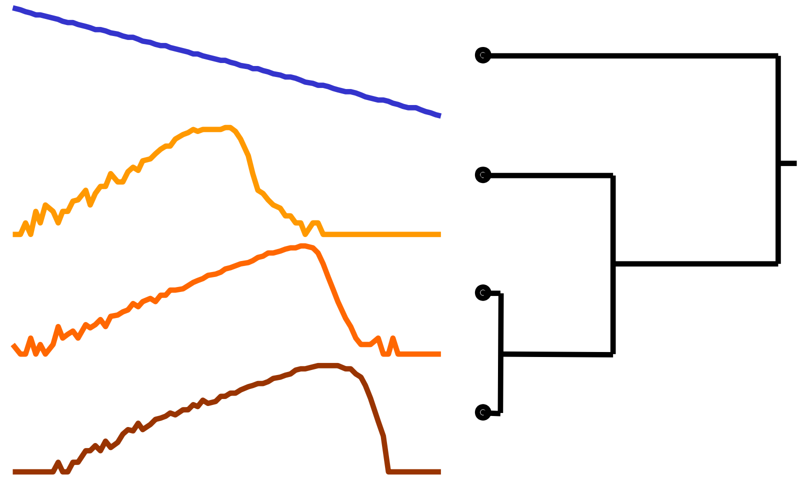
Of course, sometimes the distortions are the most interesting thing about the data, the above is only a general rule

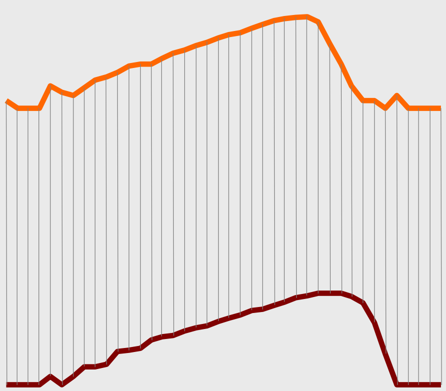
We should keep in mind these problems as we consider the high level representations of time series which we will encounter later (DFT, Wavelets etc). Since these representations often allow us to handle distortions in elegant ways

Sometimes, two time series can show a very similar behaviour, except for a slight temporal offset, or some temporal misalignments.

Dynamic Time Warping

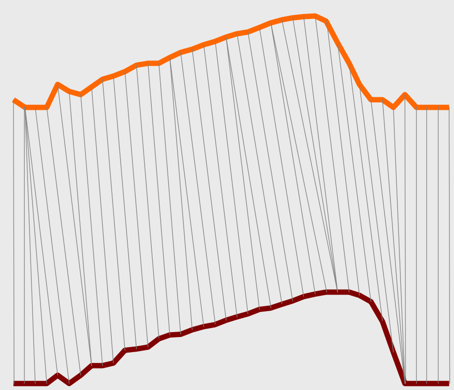






Fixed Time Axis

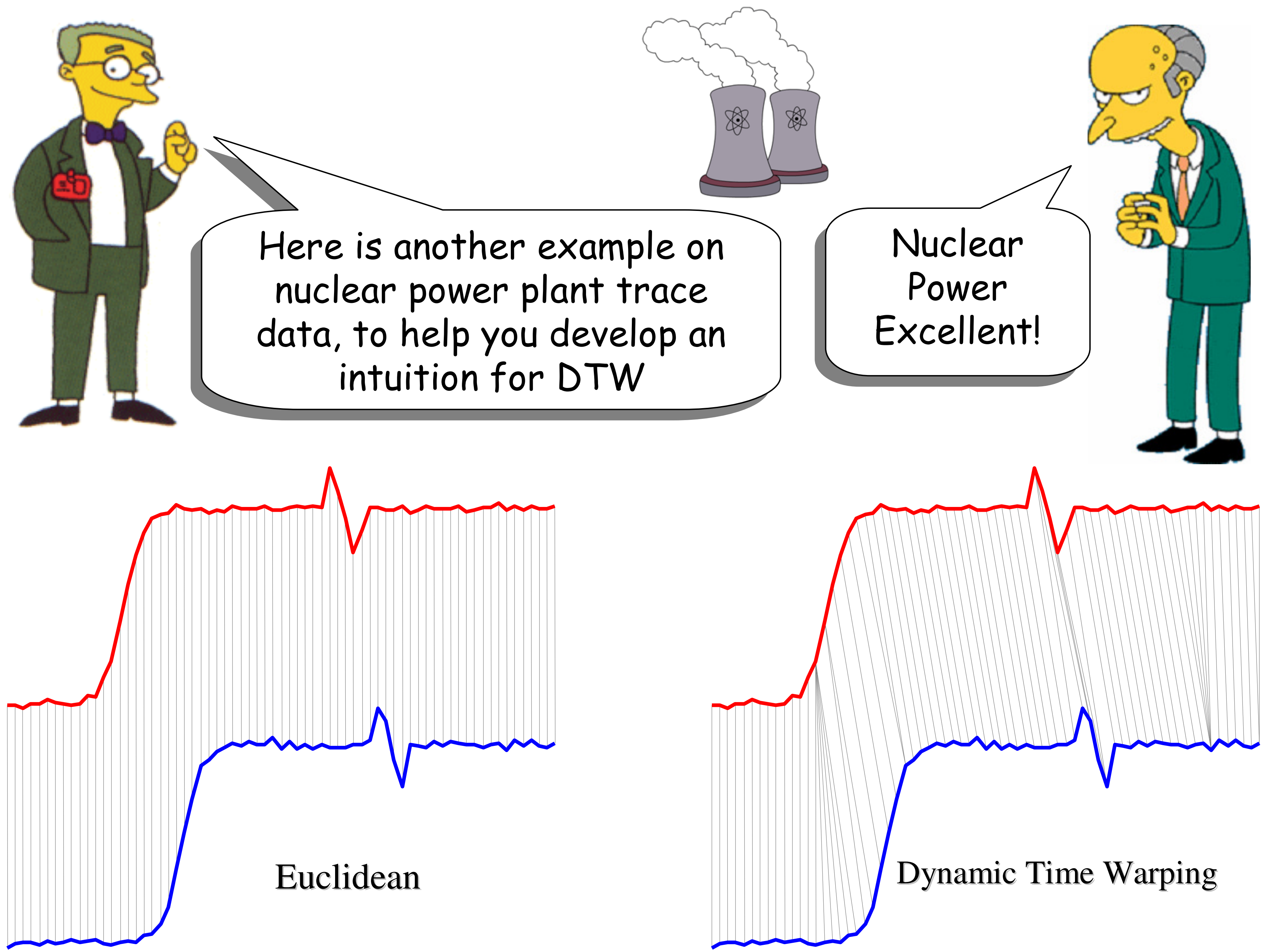
Sequences are aligned "one to one".

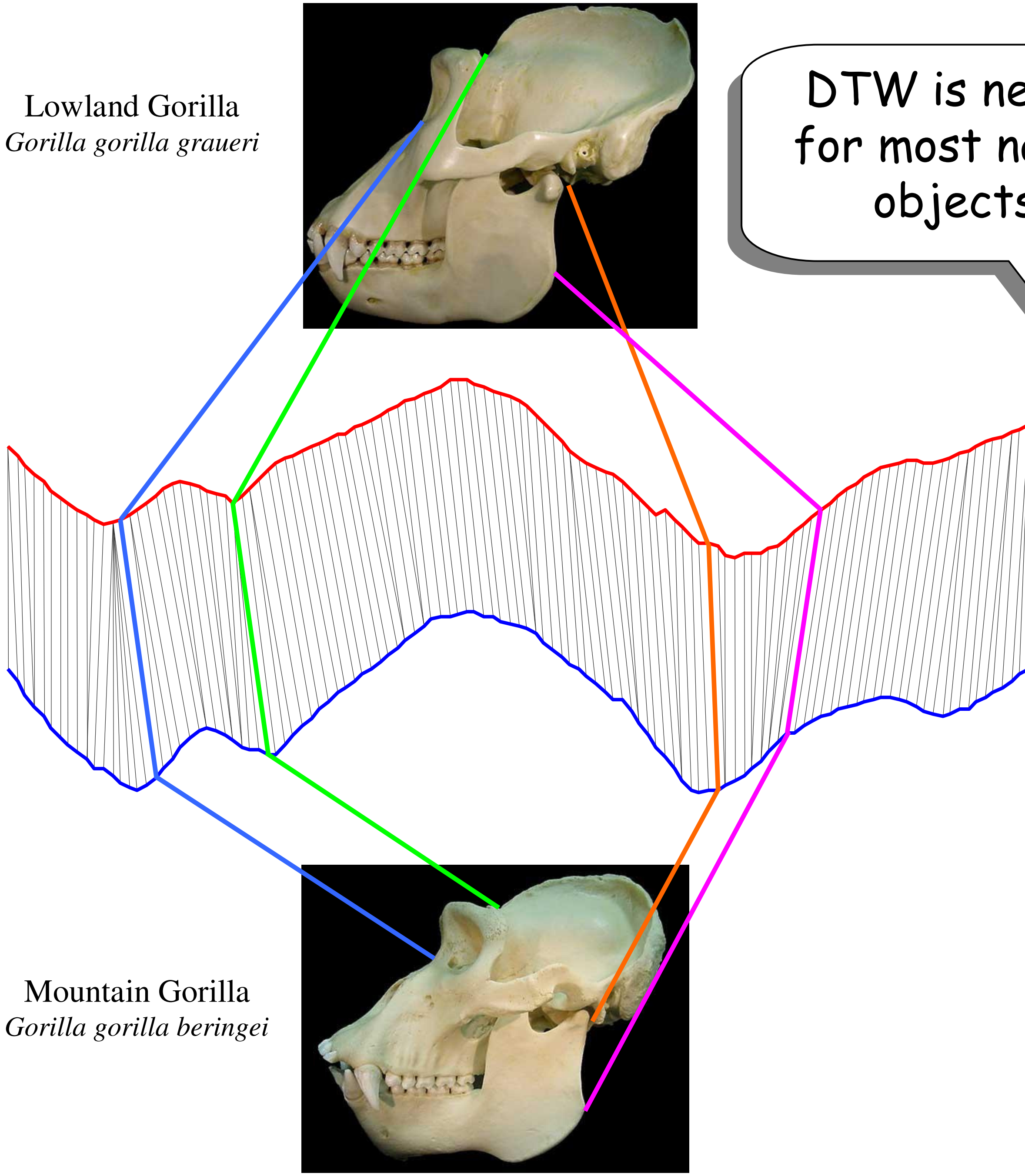


"Warped" Time Axis

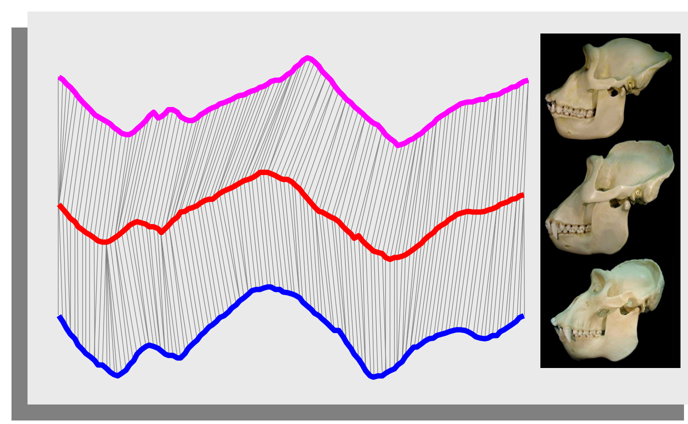
Nonlinear alignments are possible.

Note: We will first see the utility of DTW, then see how it is calculated.


Here is another example on nuclear power plant trace data, to help you develop an intuition for DTW
Nuclear Power Excellent!
Euclidean
Dynamic Time Warping

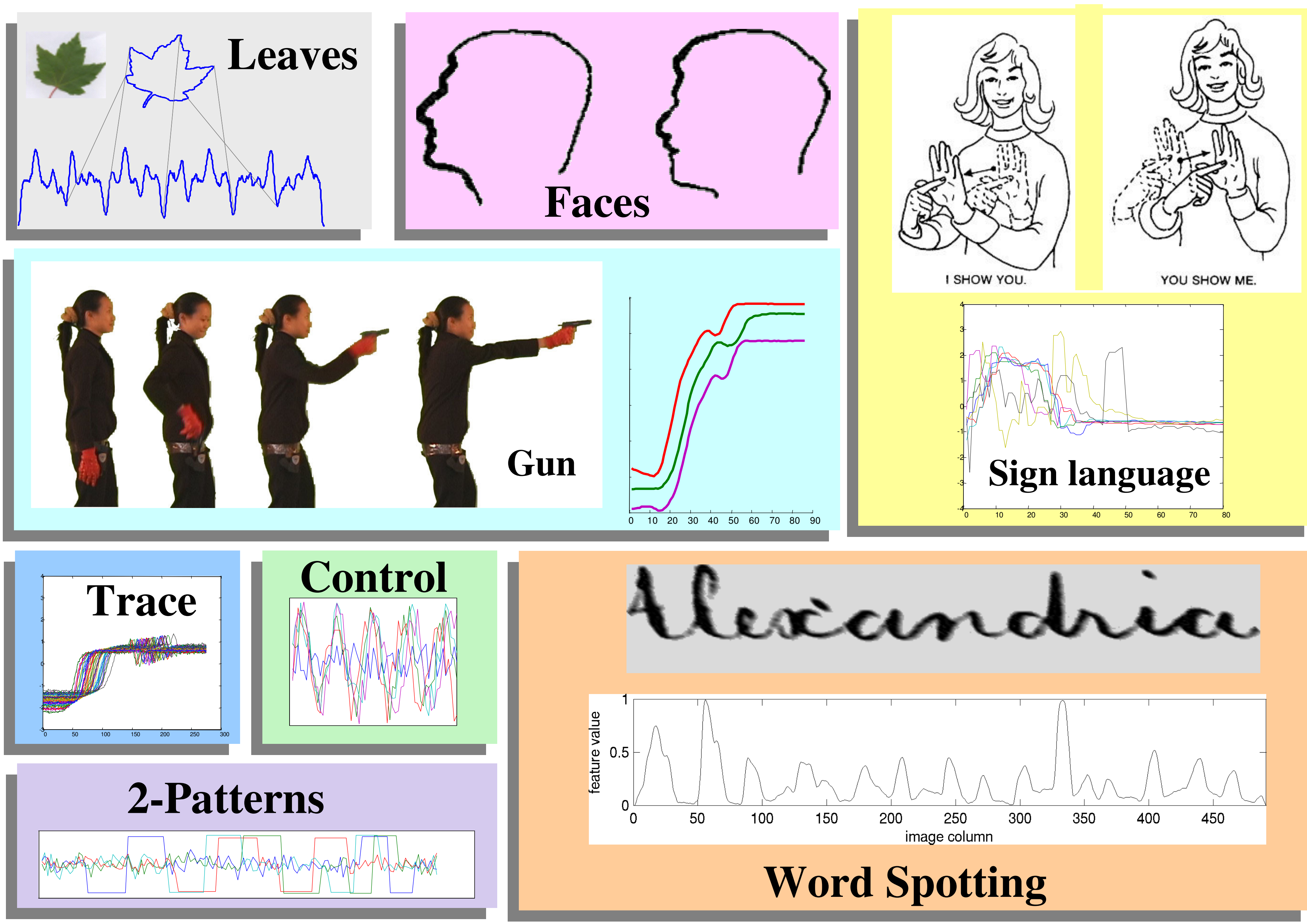


DTW is needed for most natural objects…



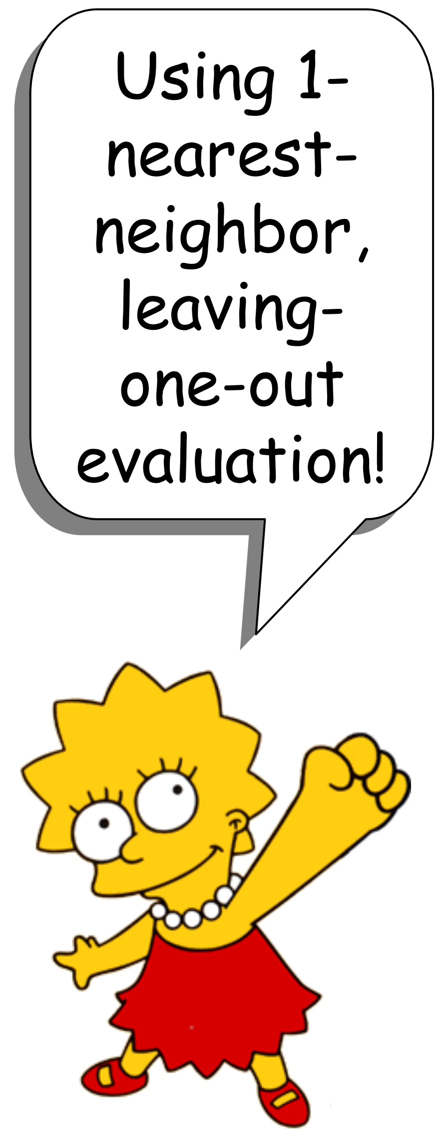
You can find all these datasets online on the UCR Time Series Classification Repository

Let us compare Euclidean Distance and DTW on some problems



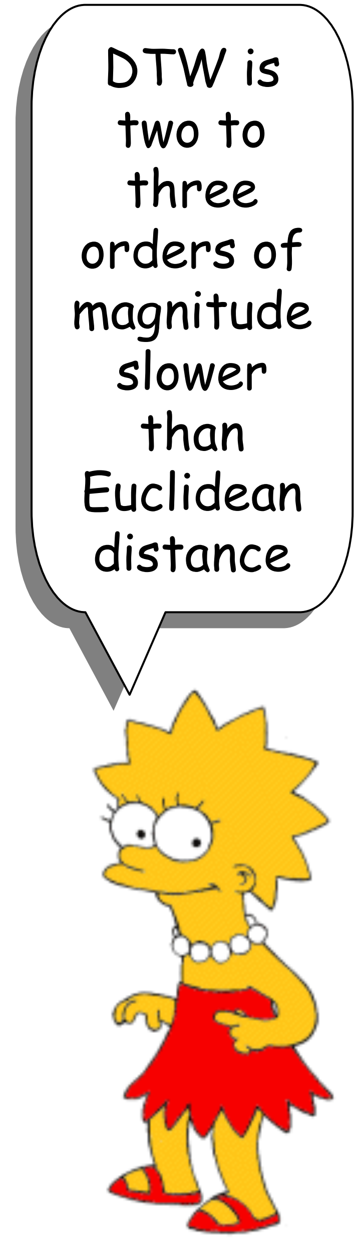
Results: Error Rate

Dataset	Euclidean	DTW
Word Spotting	4.78	1.10
Sign language	28.70	25.93
GUN	5.50	1.00
Nuclear Trace	11.00	0.00
Leaves#	33.26	4.07
(4) Faces	6.25	2.68
Control Chart*	7.5	0.33
2-Patterns	1.04	0.00



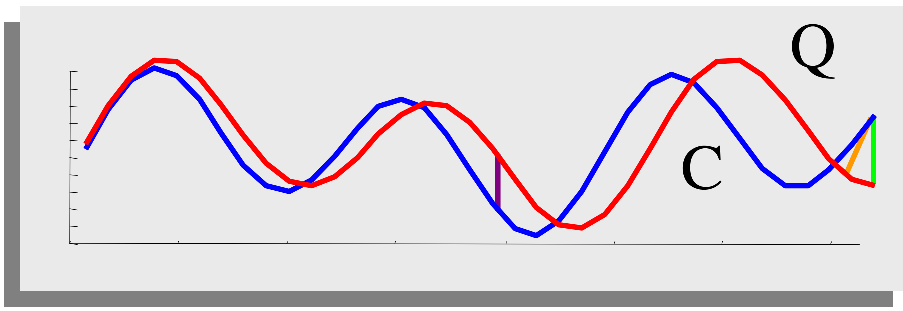
Results: Time (**msec**)

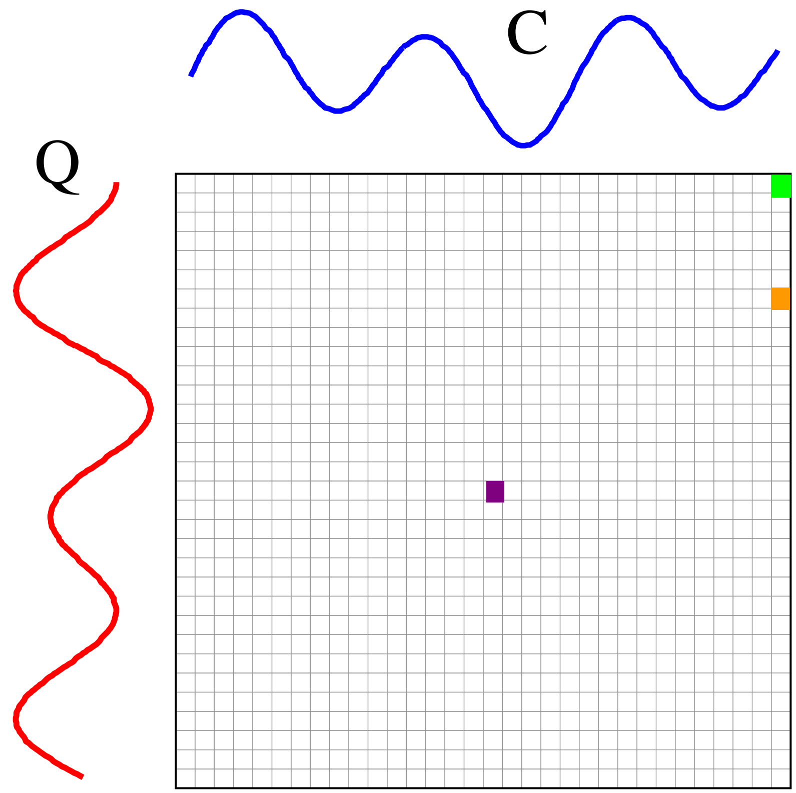
Dataset	Euclidean	DTW	
Word Spotting	40	8,600	215
Sign language	10	1,110	110
GUN	60	11,820	197
Nuclear Trace	210	144,470	687
Leaves	150	51,830	345
(4) Faces	50	45,080	901
Control Chart	110	21,900	199
2-Patterns	16,890	545,123	32



How is DTW Calculated? I

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.





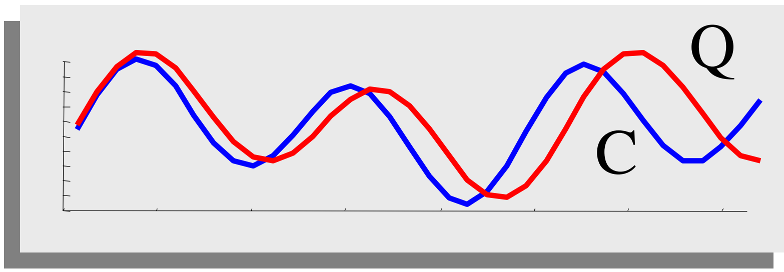
Thus, on the diagonal, you have the results of the classical Euclidean Distance metric applied on the two time series

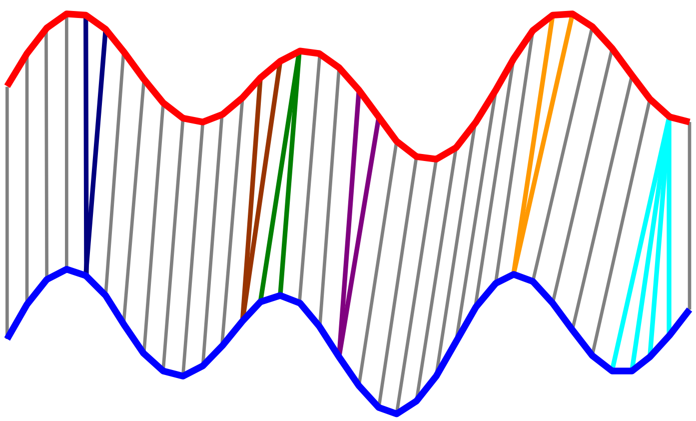
How is DTW Calculated? II

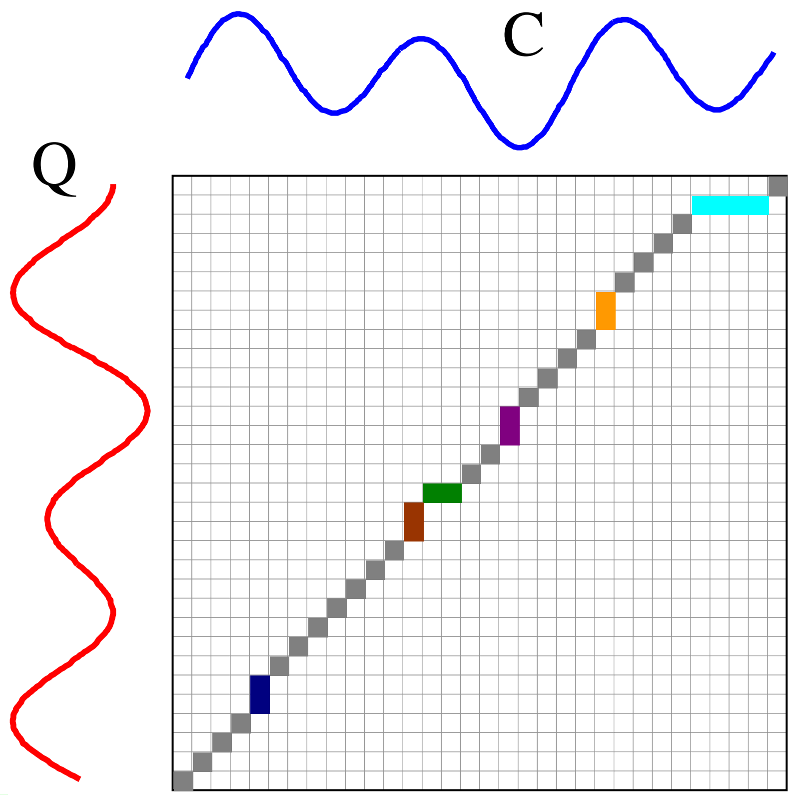
Every possible warping between two time series, is a path though the matrix. We want the best one…

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

where K is the warping path length





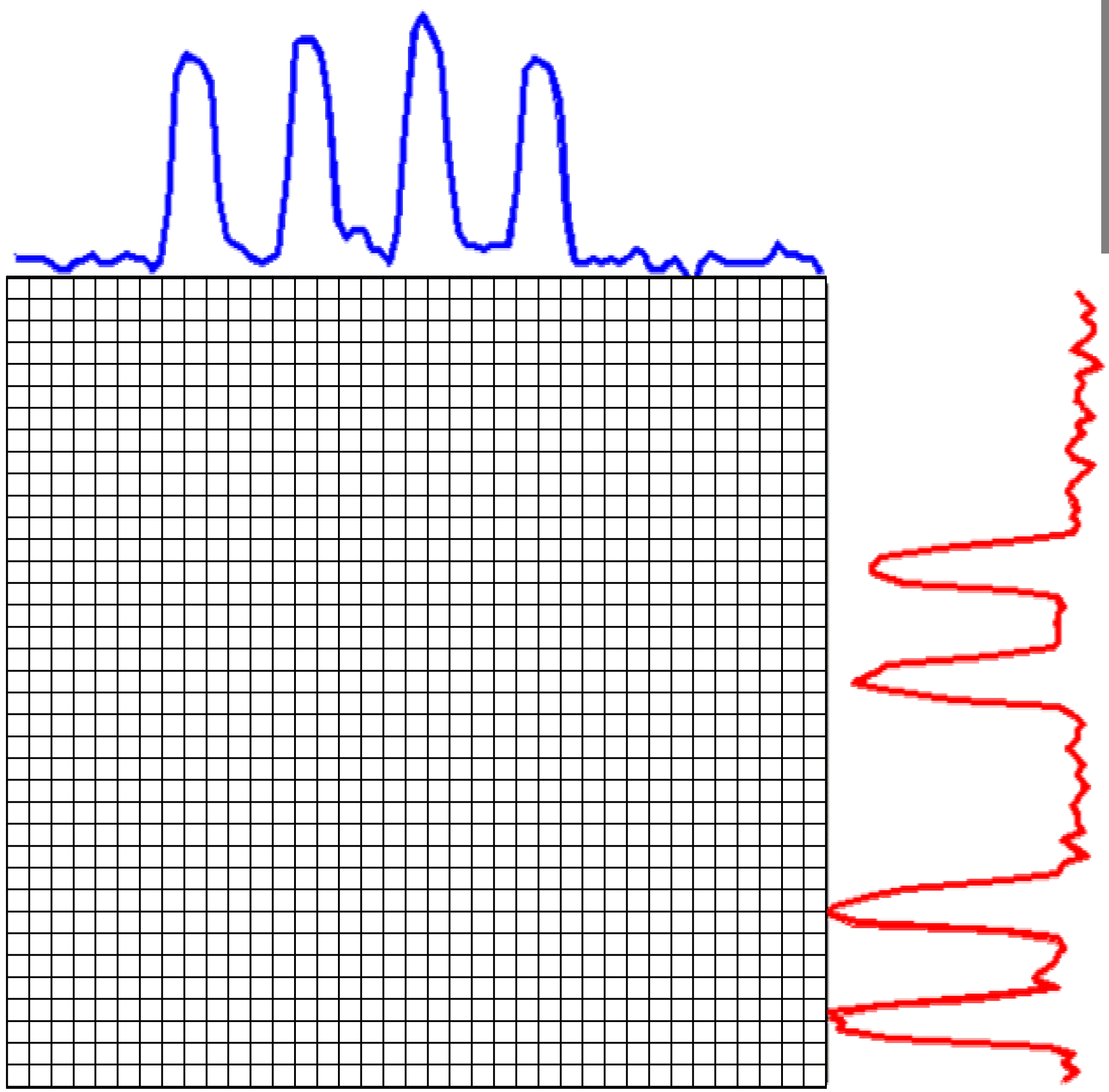


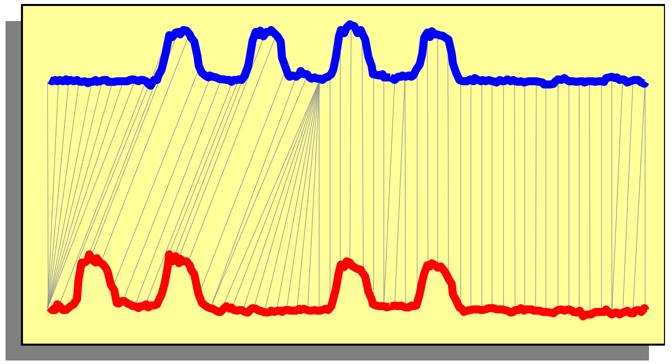
Warping path w

This recursive function gives us the minimum cost path

$\gamma(i,j) = d(q_i,c_j) + \min\{ \gamma(i\text{-}1,j\text{-}1), \gamma(i\text{-}1,j), \gamma(i,j\text{-}1) \}$

Let us visualize the cumulative matrix on a real world problem I

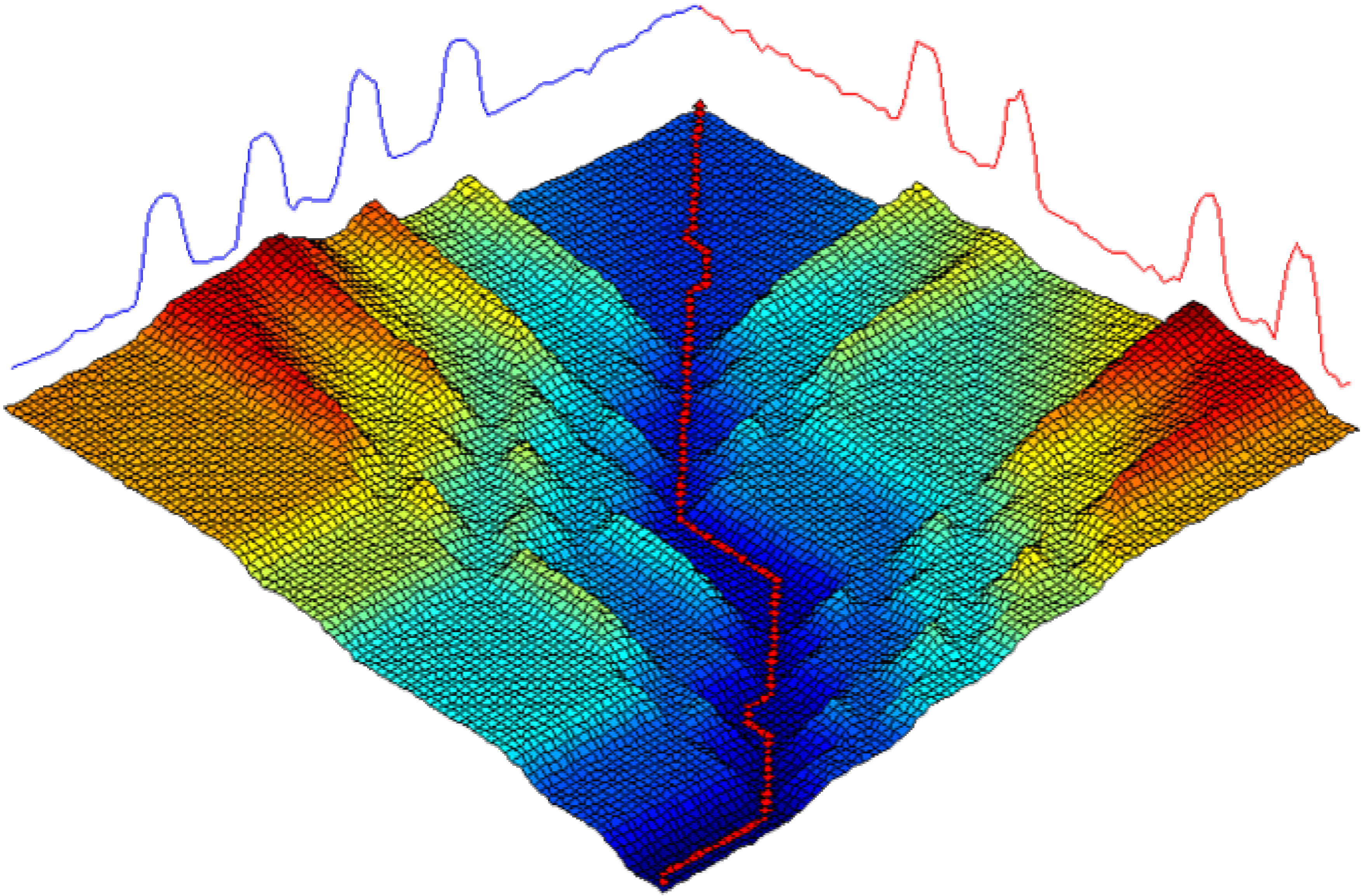




This example shows 2 one-week periods from the power demand time series.

Note that although they both describe 4-day work weeks, the blue sequence had Monday as a holiday, and the red sequence had Wednesday as a holiday.

Let us visualize the cumulative matrix on a real world problem II



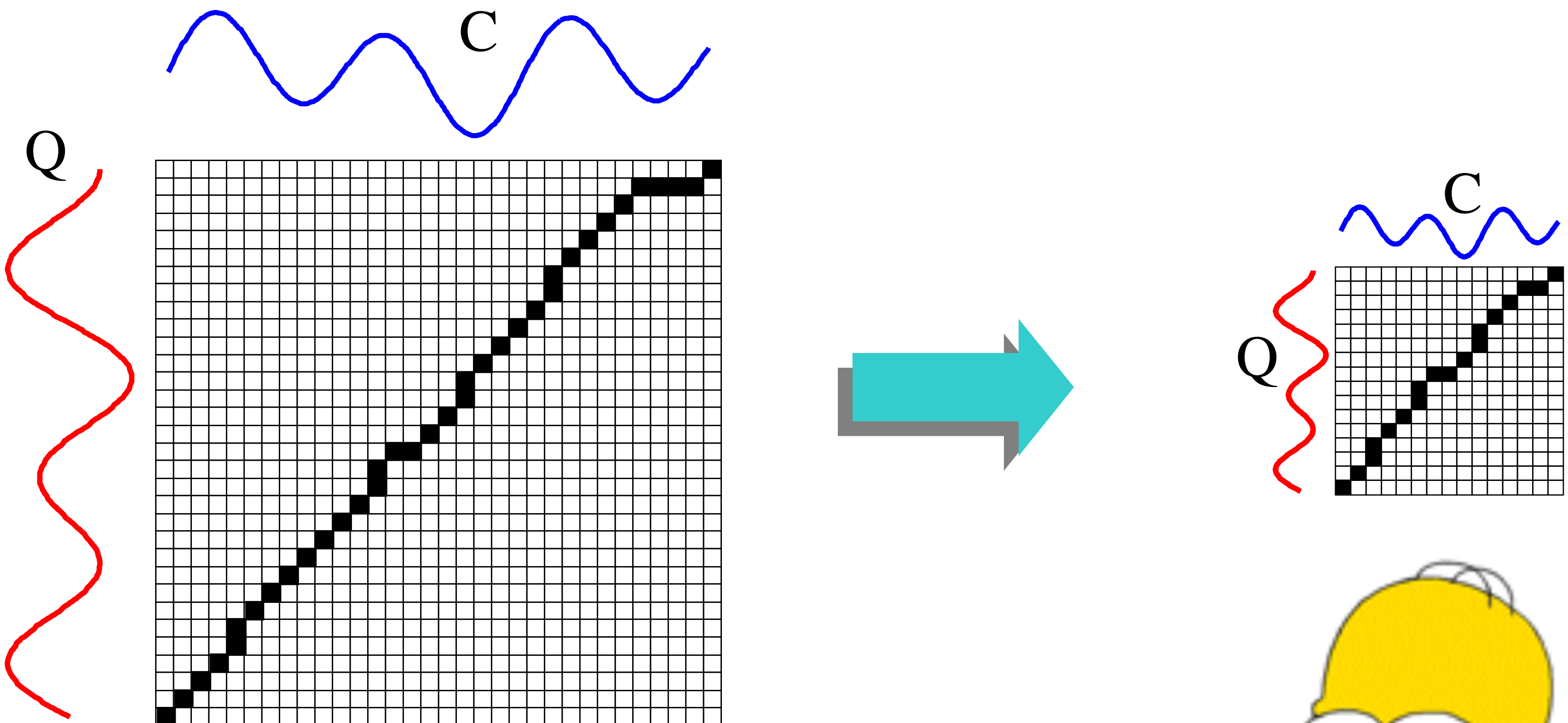
What we have seen so far…

• Dynamic Time Warping gives **much better** results than Euclidean distance on virtually all problems.

• Dynamic Time Warping is very very slow to calculate!

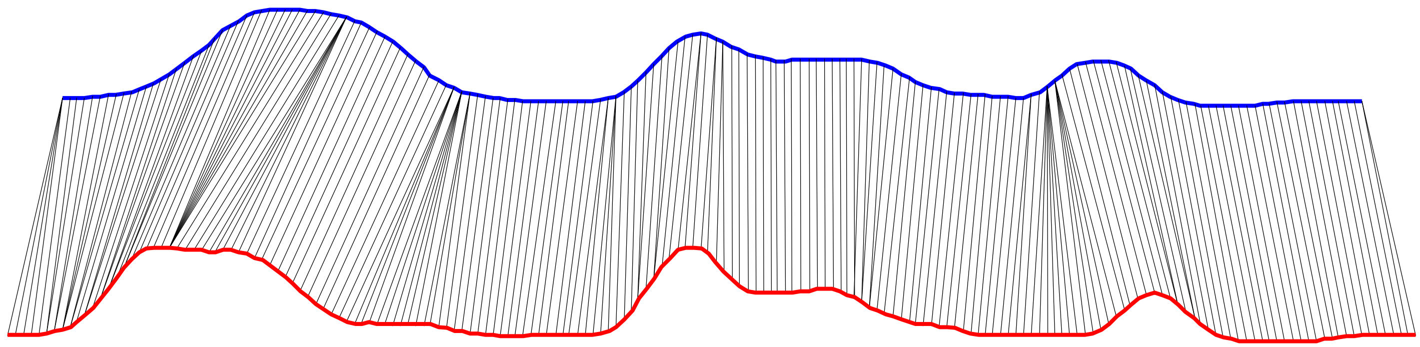
Is there anything we can do to speed up similarity search under DTW?

Fast Approximations to Dynamic Time Warp Distance I

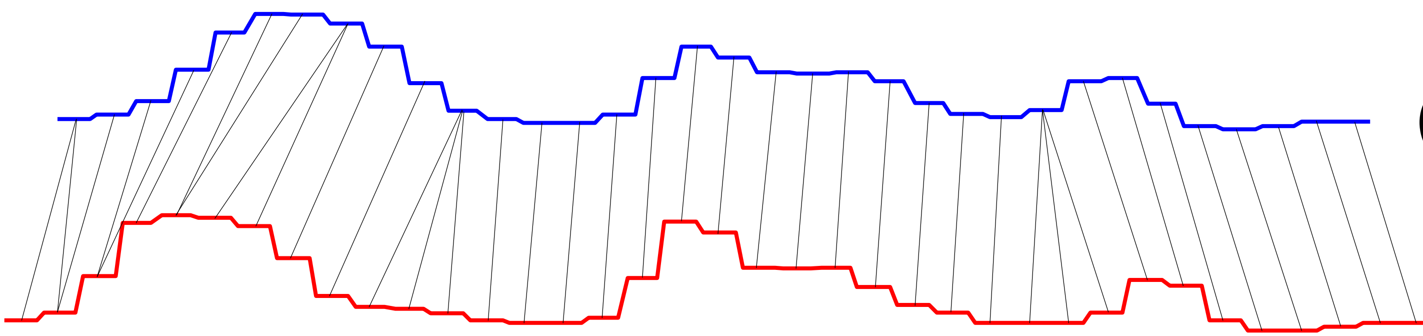


Simple Idea: Approximate the time series with some compressed or downsampled representation, and do DTW on the new representation. How well does this work...

Fast Approximations to Dynamic Time Warp Distance II



1.03 sec



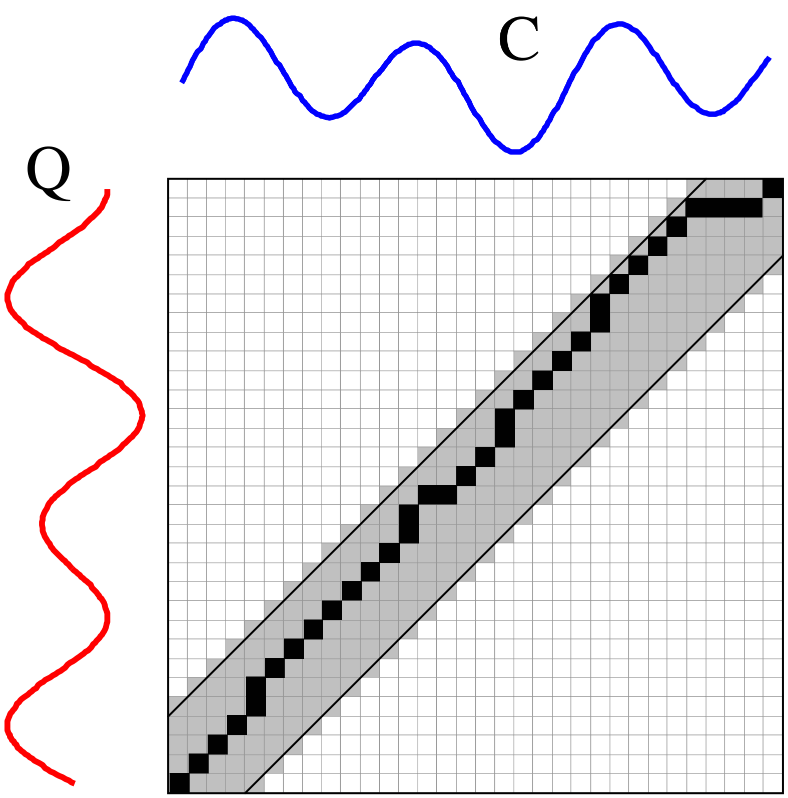
0.07 sec

… there is strong visual evidence to suggests it works well

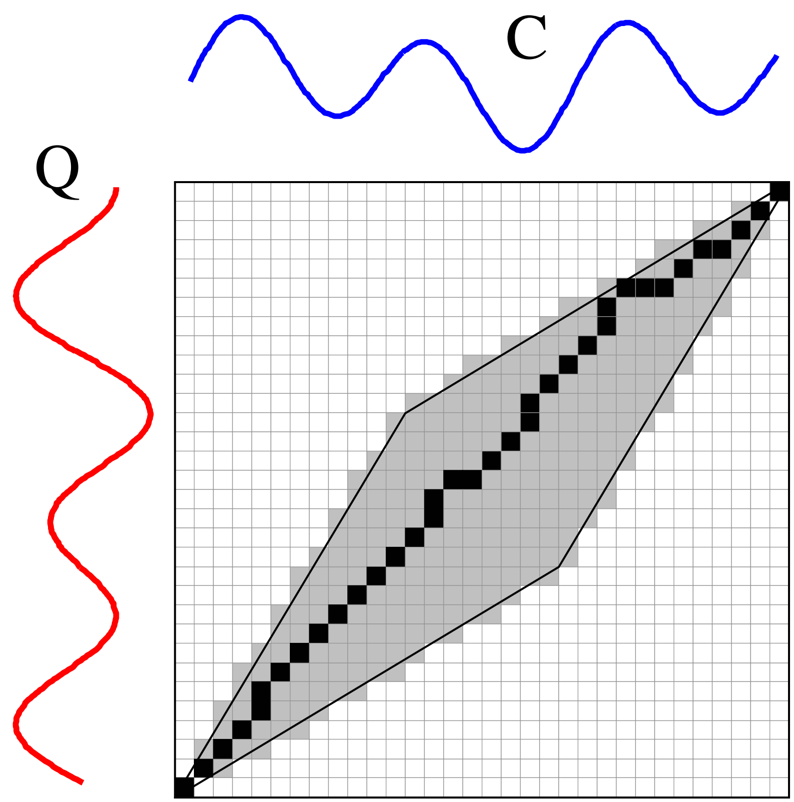
There is good experimental evidence for the utility of the approach on clustering, classification, etc

Global Constraints

- Slightly speed up the calculations
- Prevent pathological warpings

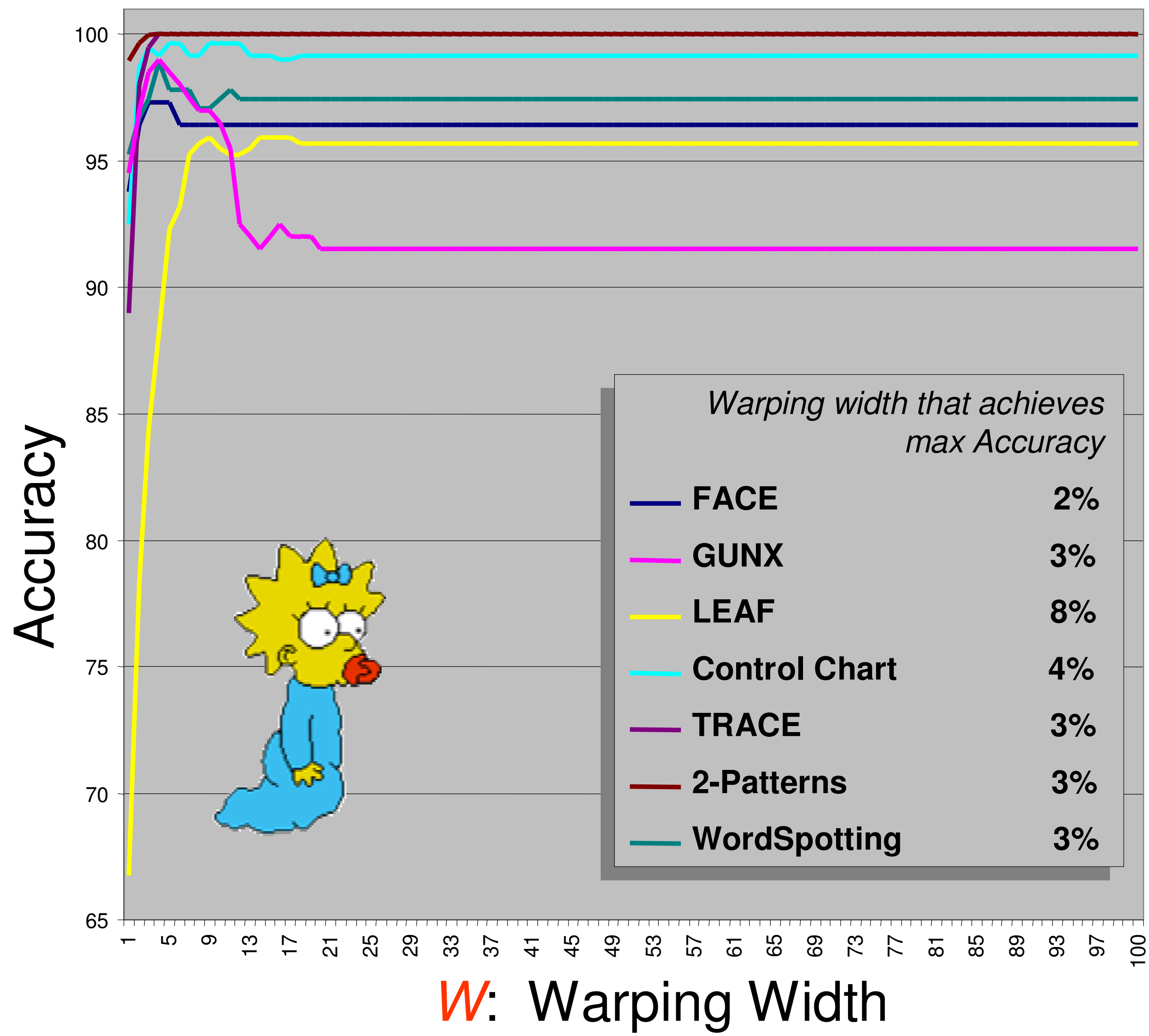


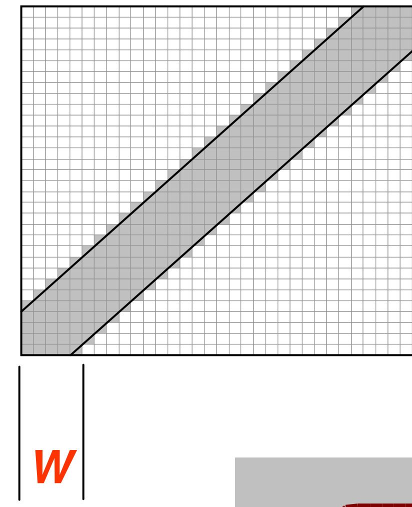
Sakoe-Chiba Band



Itakura Parallelogram

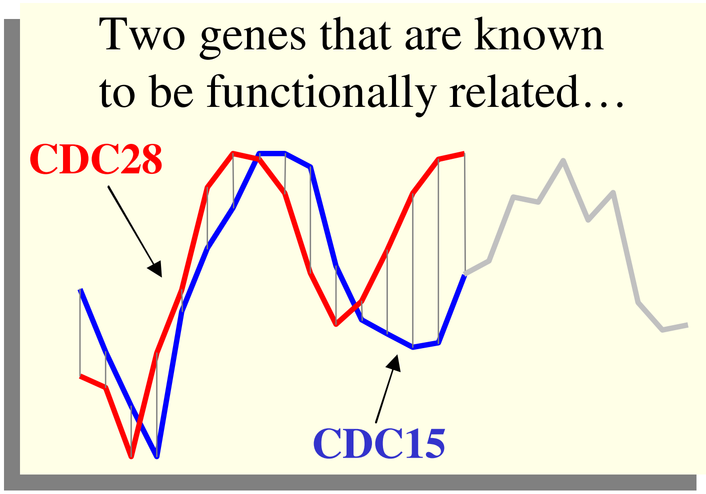
Accuracy vs. Width of Warping Window



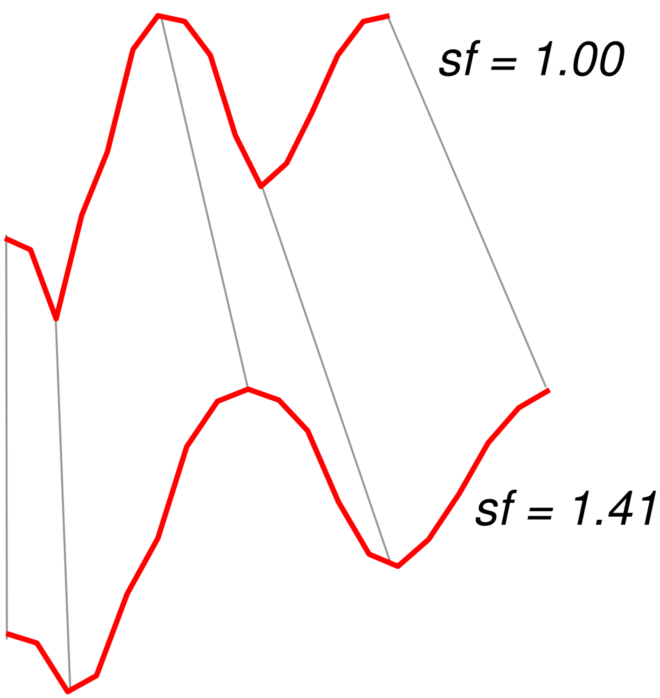


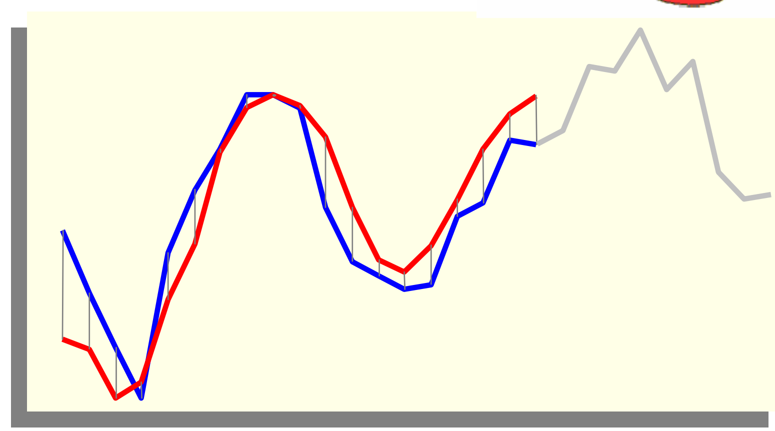
A global time scaling on a time series uniformly squeezes or stretches the time series on the time axis.

Uniform Scaling I



Sometimes global or *uniform scaling* is as important as DTW





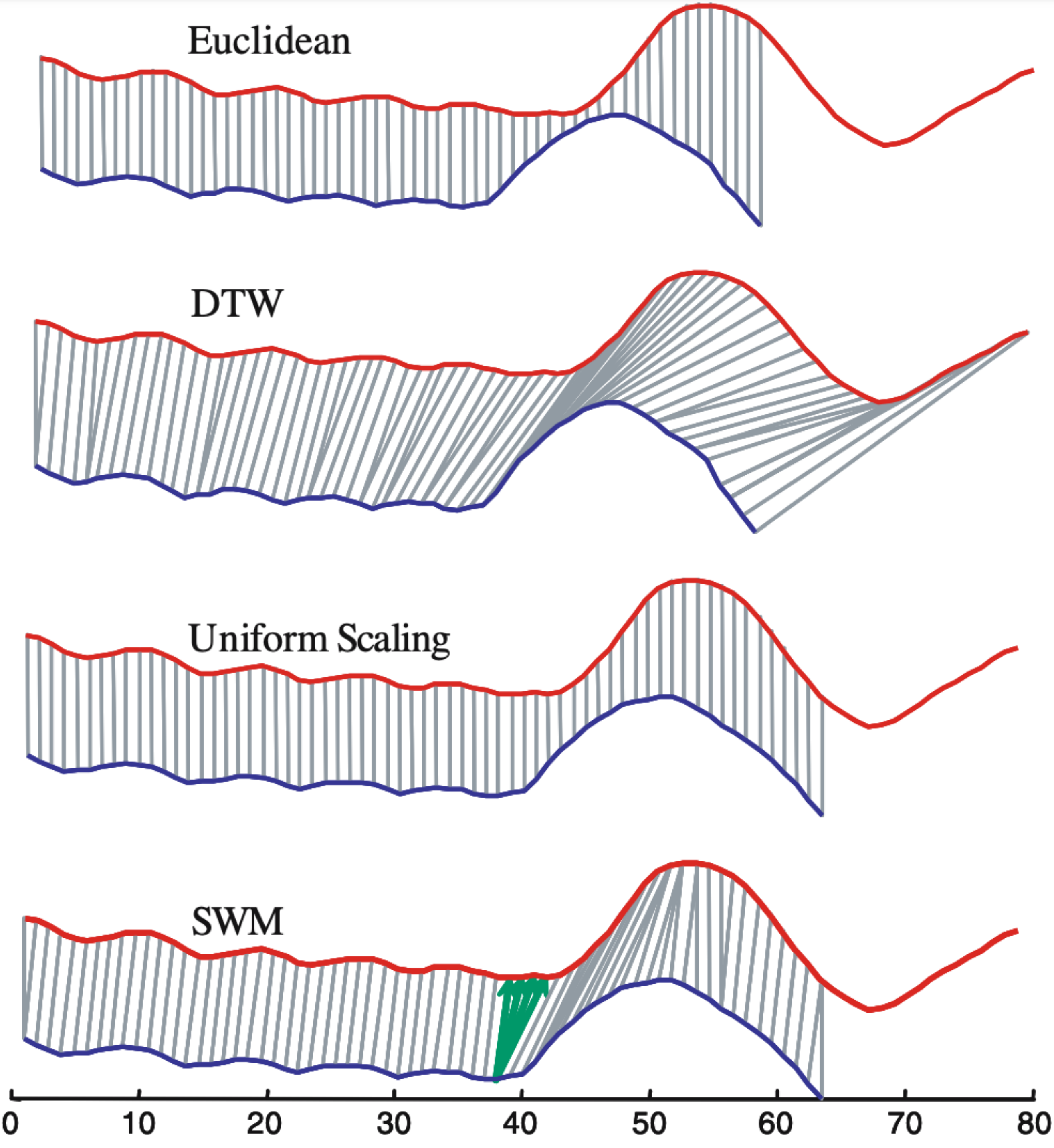
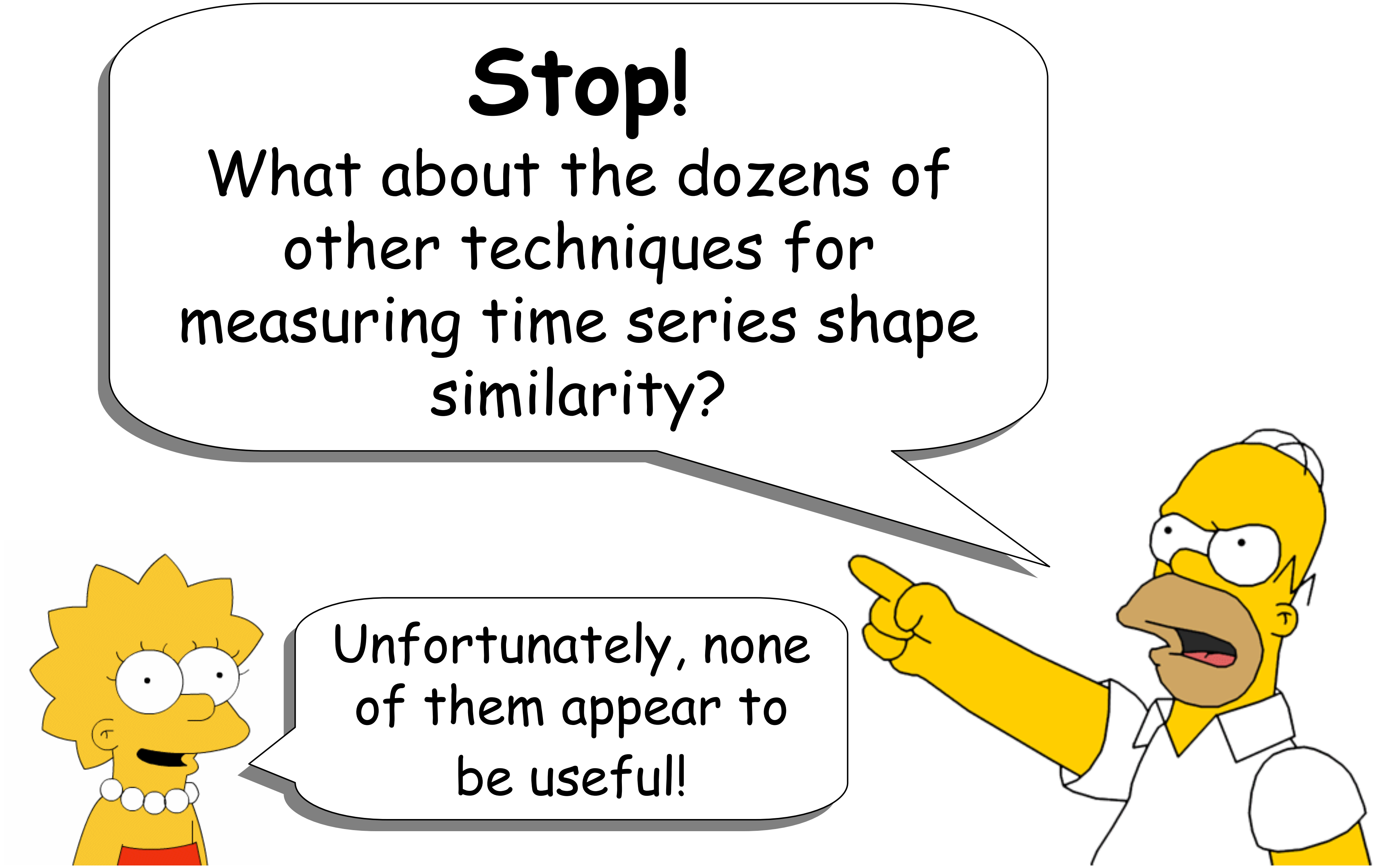


Fig. 1 Two examples of an athlete's trajectories aligned with various measures

Only Euclidean and DTW Distance are Useful

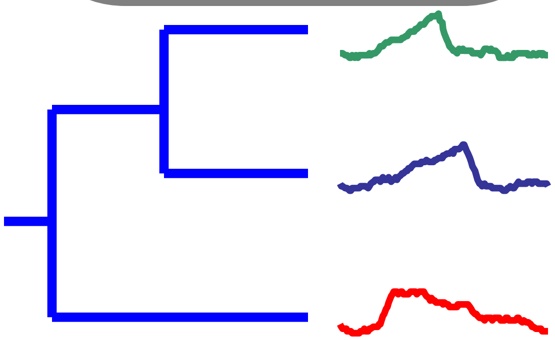


Classification Error Rates on
two publicly available datasets

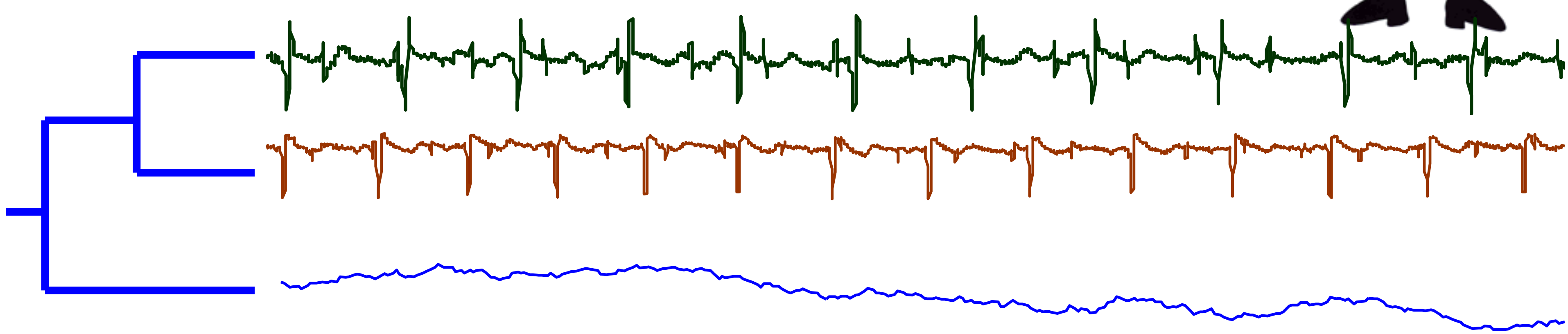
Approach	Cylinder-Bell-F'	Control-Chart
Euclidean Distance	***0.003***	***0.013***
Aligned Subsequence	**0.451**	**0.623**
Piecewise Normalization	**0.130**	**0.321**
Autocorrelation Functions	**0.380**	**0.116**
Cepstrum	**0.570**	**0.458**
String (Suffix Tree)	**0.206**	**0.578**
Important Points	**0.387**	**0.478**
Edit Distance	**0.603**	**0.622**
String Signature	**0.444**	**0.695**
Cosine Wavelets	**0.130**	**0.371**
Hölder	**0.331**	**0.593**
Piecewise Probabilistic	**0.202**	**0.321**

Two Kinds of Similarity

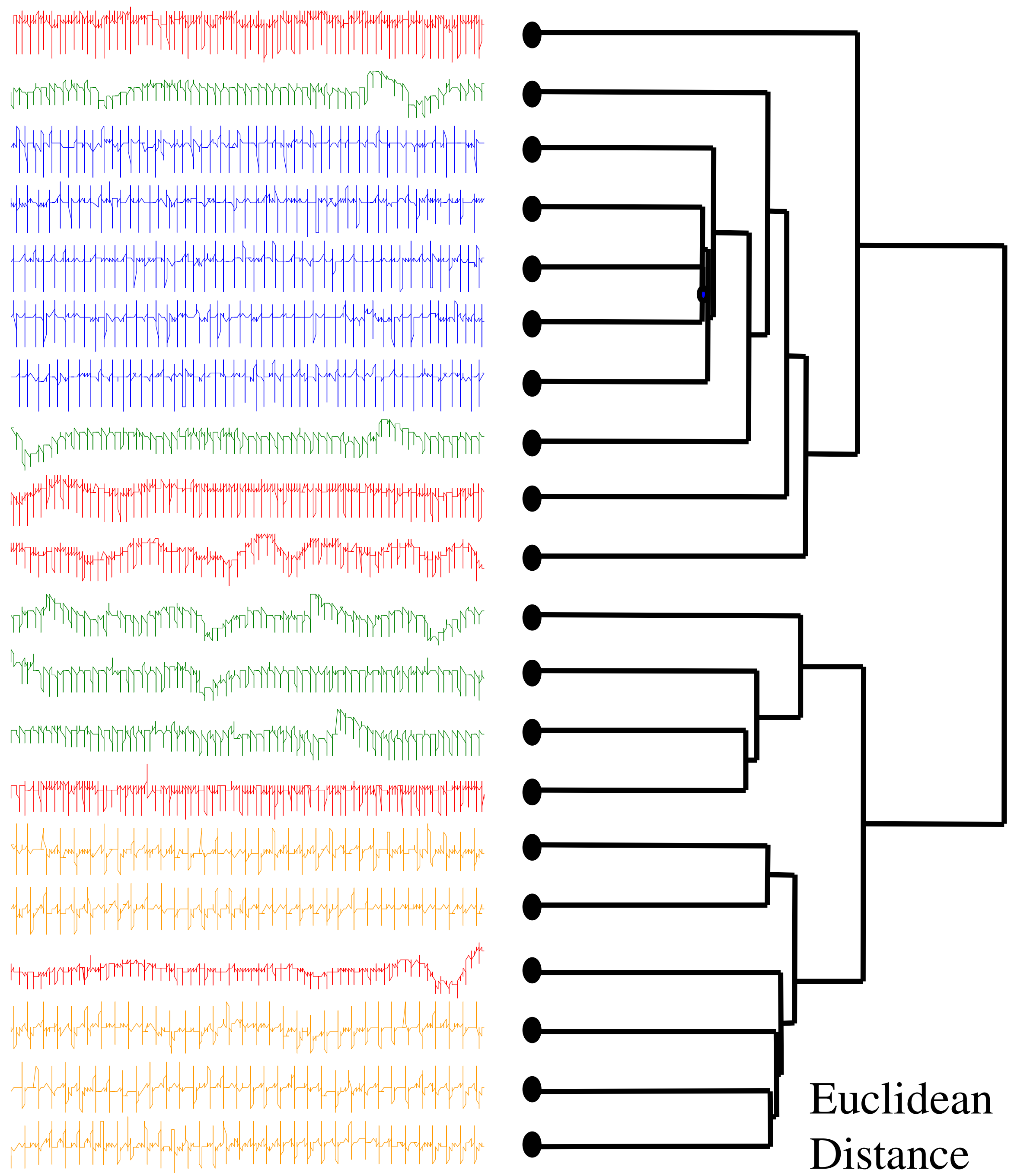
We are done with *shape* similarity



Let us consider similarity at the *structural* level for the next 10 minutes

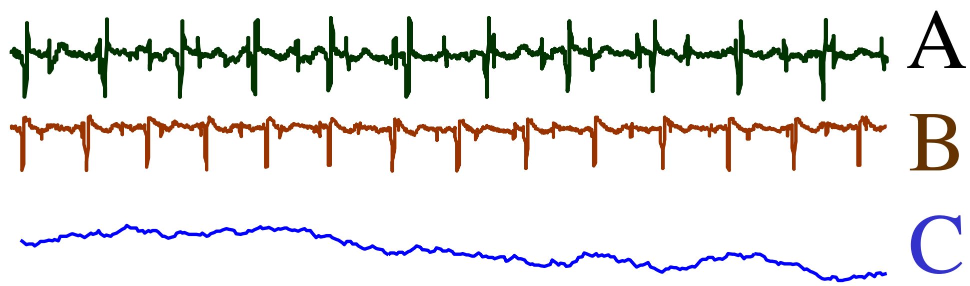



For long time series, *shape* based similarity will give very poor results. We need to measure similarly based on high level *structure*


Euclidean Distance

Structure or Model Based Similarity

The basic idea is to extract *global* features from the time series, create a feature vector, and use these feature vectors to measure similarity and/or classify



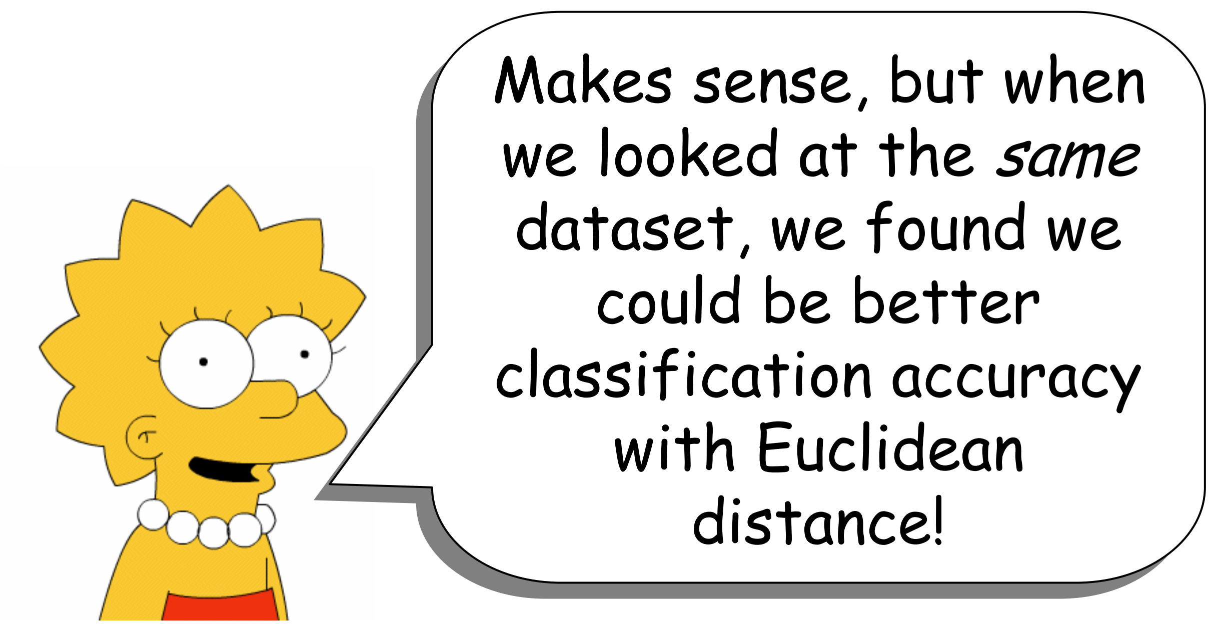
But which
- **features**?
- **distance measure/ learning algorithm**?

Feature \ Time Series	A	B	C
Max Value	11	12	19
Autocorrelation	0.2	0.3	0.5
Zero Crossings	98	82	13
Average	0.3	0.4	0.1
…	…	…	…

Feature-based Classification of Time-series Data

Nanopoulos, Alcock, and Manolopoulos

Features
mean
variance
skewness
kurtosis
mean (1st derivative)
variance (1st derivative)
skewness (1st derivative)
kurtosis (1st derivative)



Compression Based Dissimilarity

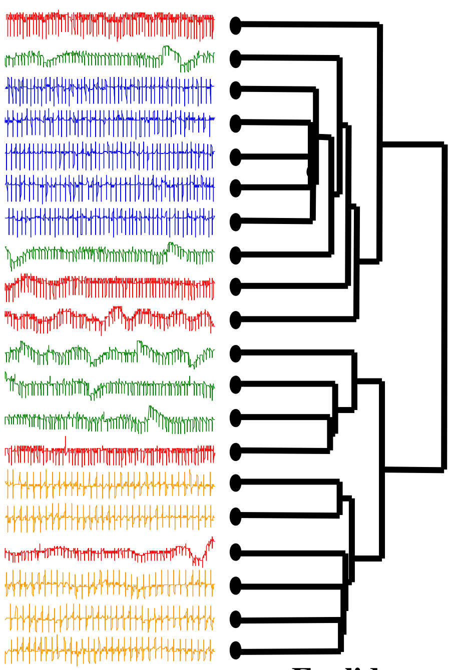
(**In general**) Li, Chen, Li, Ma, and Vitányi: (**For time series**) Keogh, Lonardi and Ratanamahatana

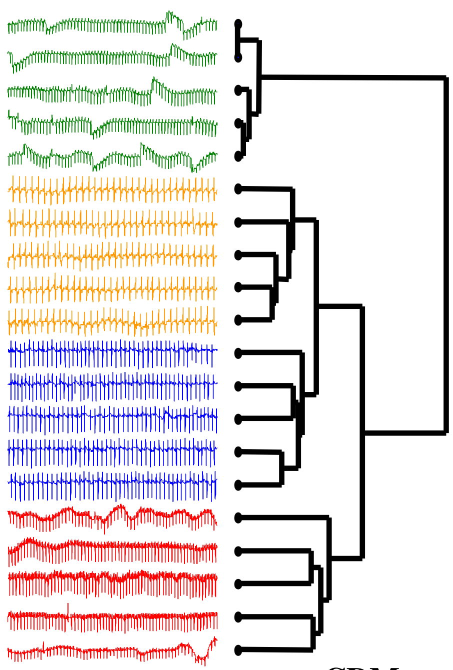
- features?
- distance measure/ learning algorithm?

Distance Measure

Co-Compressibility

$$CDM(x, y) = \frac{C(xy)}{C(x) + C(y)}$$


Euclidean


CDM

Features

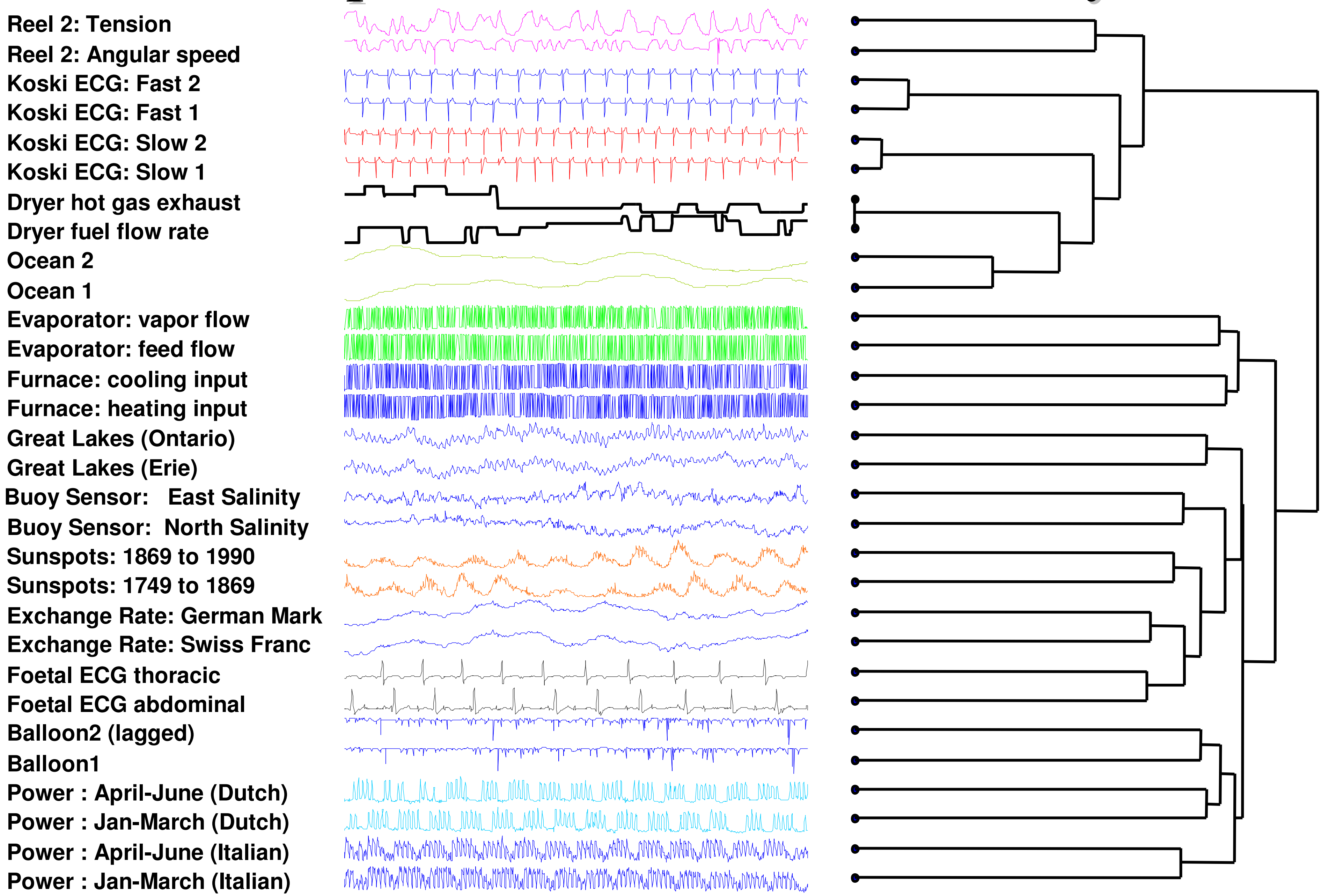
Whatever structure the compression algorithm finds...

The time series is first converted to the SAX symbolic representation*

SAX transforms a time-series X of length n into a string of arbitrary length ω, where ω«n typically, using an alphabet A of size a > 2.

Then, the principle of CBD is: the more patterns two strings share, the smaller is the compressed file size of their concatenated string.

Compression Based Dissimilarity



Summary of Time Series Similarity

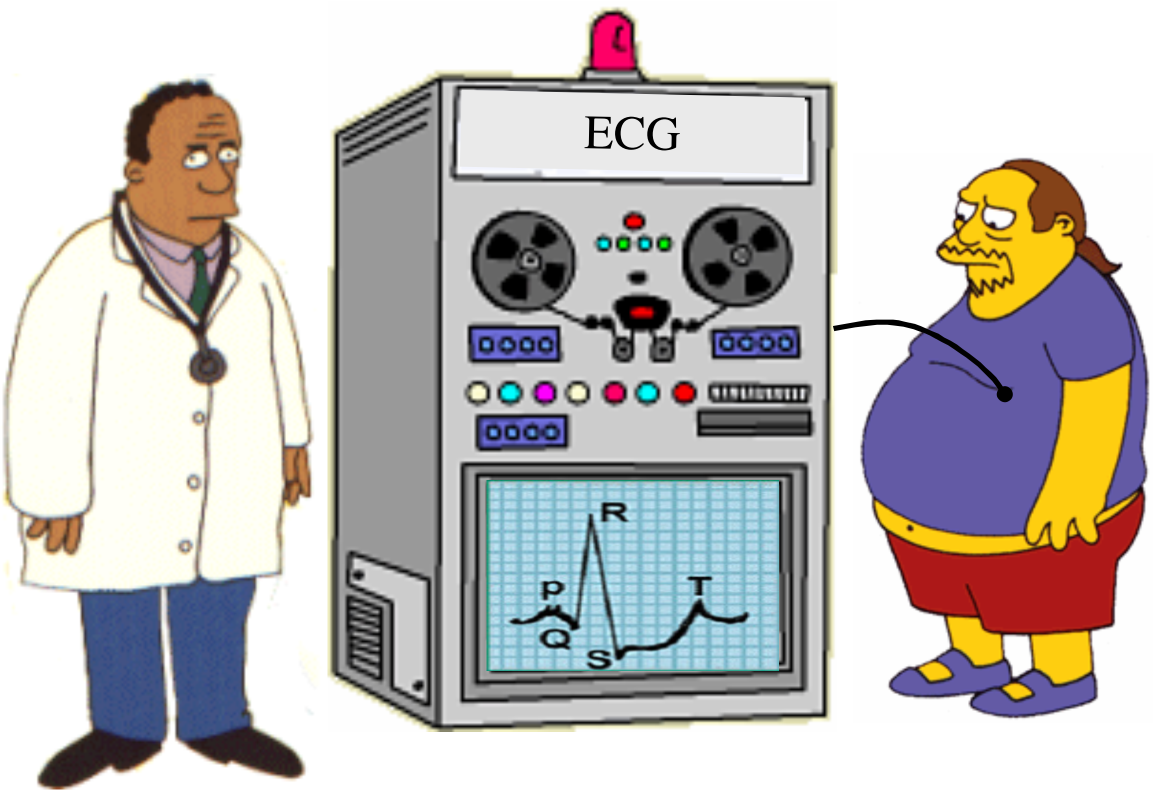
• If you have *short* time series, use DTW after searching over the warping window size

- Also, consider Uniform Scaling, and preprocessing

• If you have *long* time series, and you know nothing about your data, try compression based dissimilarity. (after converting it with SAX)

• If you do know something about your data, try to leverage of this knowledge to extract features.

Motivating example revisited…



You go to the doctor because of chest pains. Your ECG looks strange…

Your doctor wants to search a database to find **similar** ECGs, in the hope that they will offer clues about your condition...

Two questions:

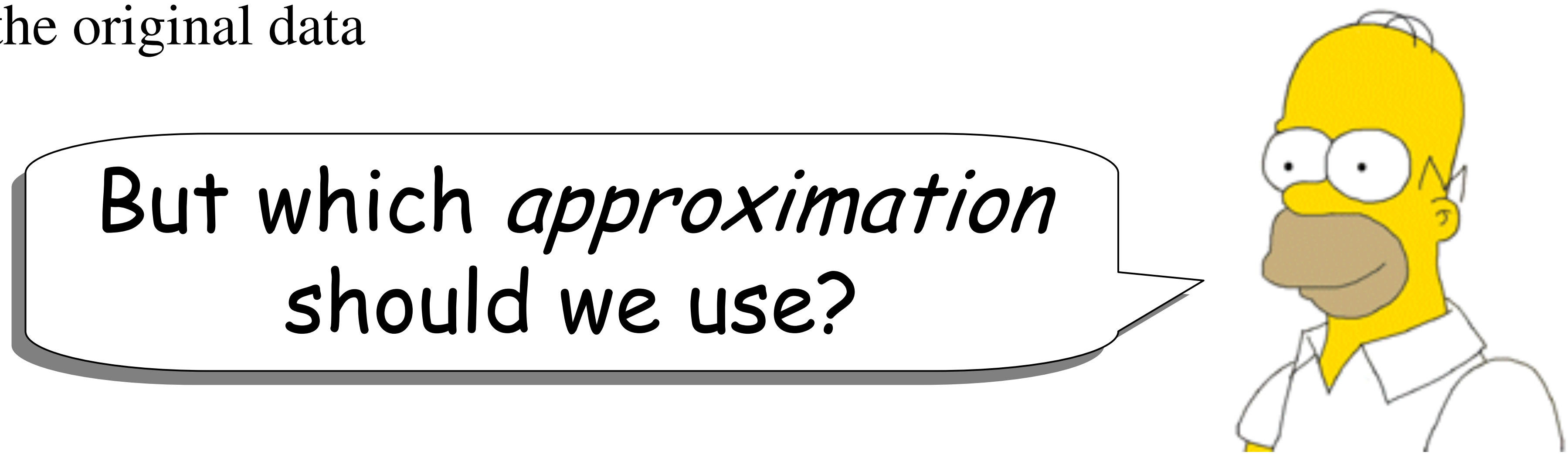
- How do we define similar?
- **How do we search quickly?**

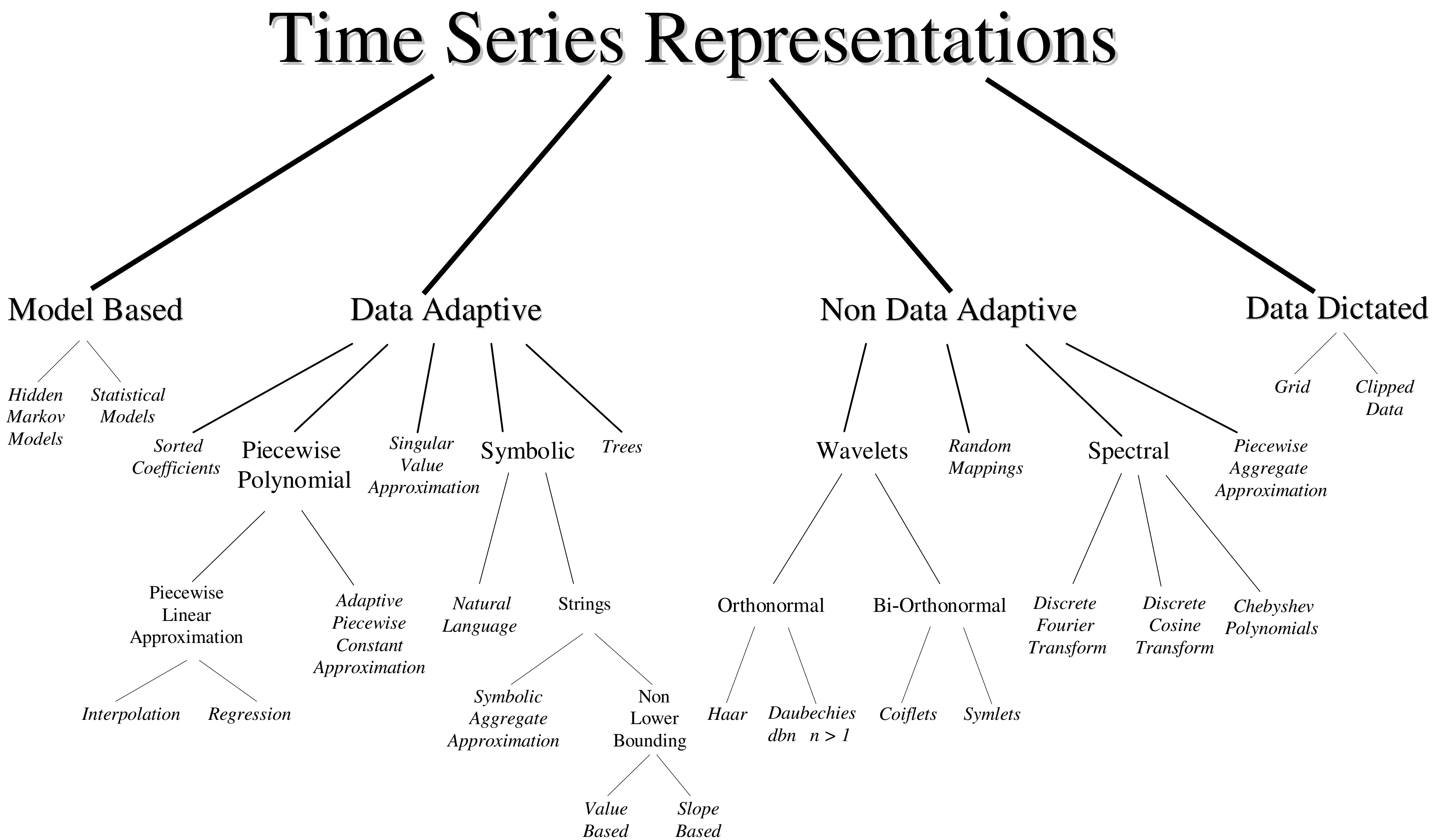
The Generic Data Mining Algorithm

- 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 (hopefully very few) accesses to the original data on disk to confirm the solution obtained in Step 2, or to modify the solution so it agrees with the solution we would have obtained on the original data




Time Series Representations
Model Based
Hidden Markov Models
Statistical Models
Data Adaptive
Sorted Coefficients
Piecewise Polynomial
Piecewise Linear Approximation
Interpolation
Regression
Adaptive Piecewise Constant Approximation
Singular Value Approximation
Symbolic
Natural Language
Strings
Symbolic Aggregate Approximation
Non Lower Bounding
Value Based
Slope Based
Trees
Non Data Adaptive
Wavelets
Orthonormal
Haar
Daubechies dbn n > 1
Bi-Orthonormal
Coiflets
Symlets
Random Mappings
Spectral
Discrete Fourier Transform
Discrete Cosine Transform
Chebyshev Polynomials
Piecewise Aggregate Approximation
Data Dictated
Grid
Clipped Data

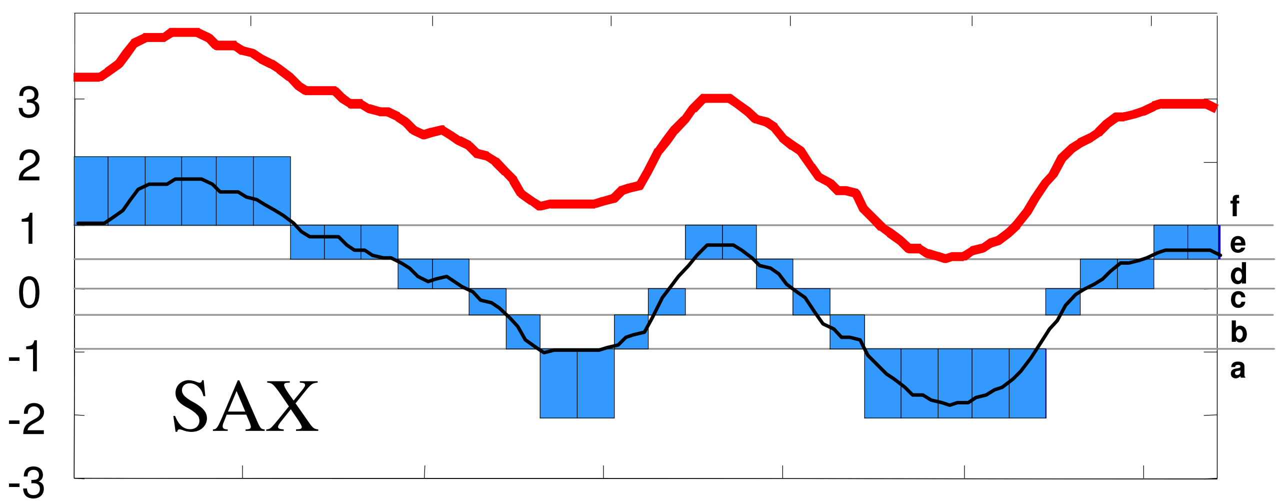
Lets take a tour of other time series problems

- Before we do, let us briefly revisit SAX, since it has some implications for the other problems…

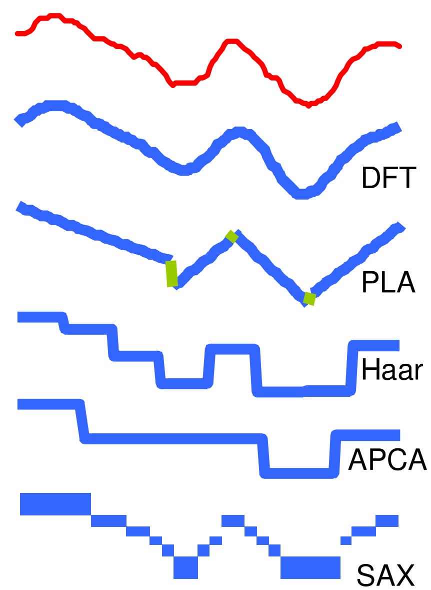
Exploiting Symbolic Representations of Time Series

There is now a lower bounding dimensionality reducing time series representation! It is called SAX (**S**ymbolic **A**ggregate Appro**X**imation)

I expect SAX to have a major impact on time series data mining in the coming years…



ffffffeeeddcbaabceedcbaaaaacddee



> Then you can rely on algorithms developed for text mining and bioinformatics!

Anomaly (interestingness) detection

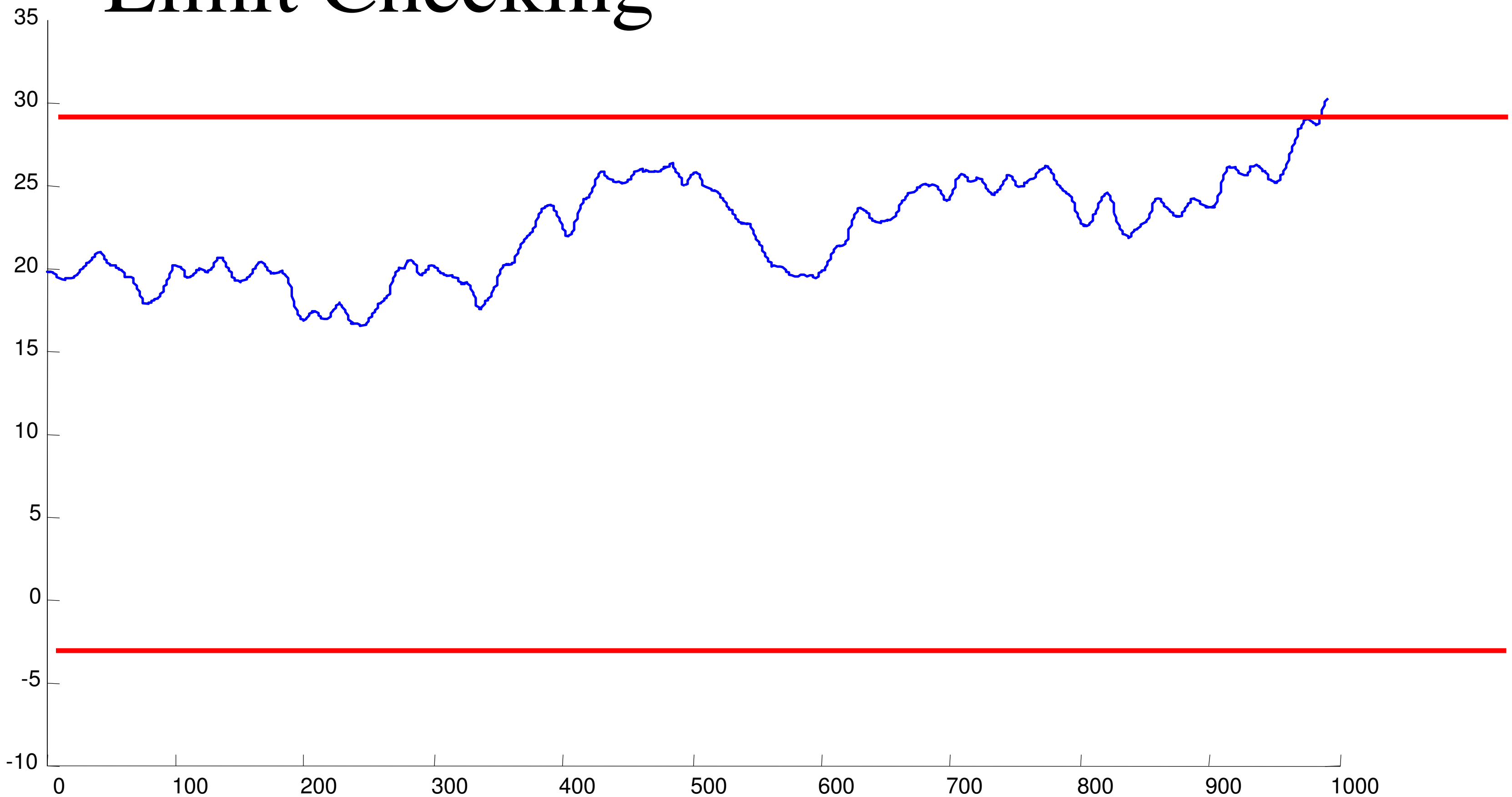
We would like to be able to discover surprising (unusual, interesting, anomalous) patterns in time series.

Note that we don't know in advance in what way the time series might be surprising

Also note that "surprising" is very context dependent, application dependent, subjective etc.

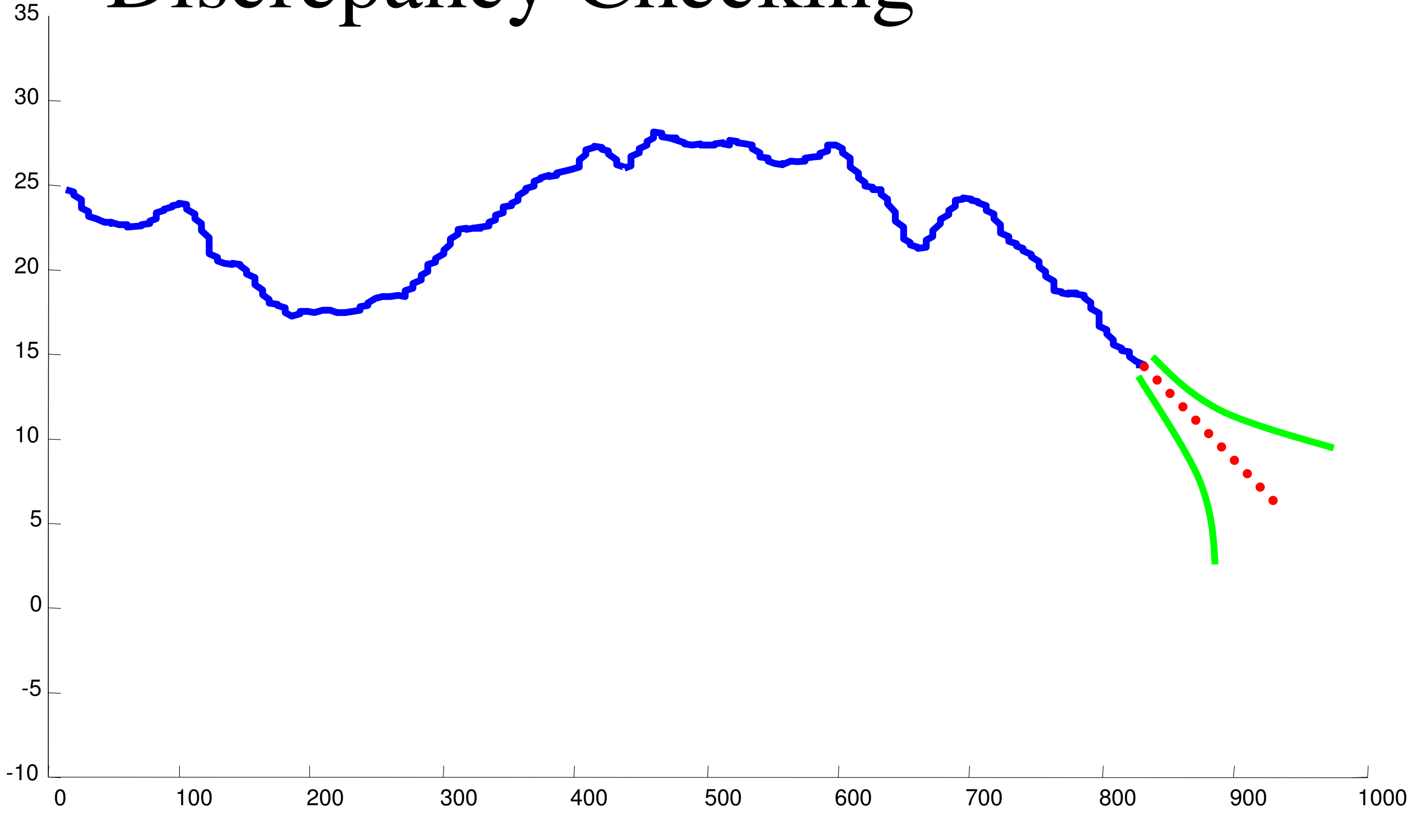
Simple Approaches I

Limit Checking



Simple Approaches II

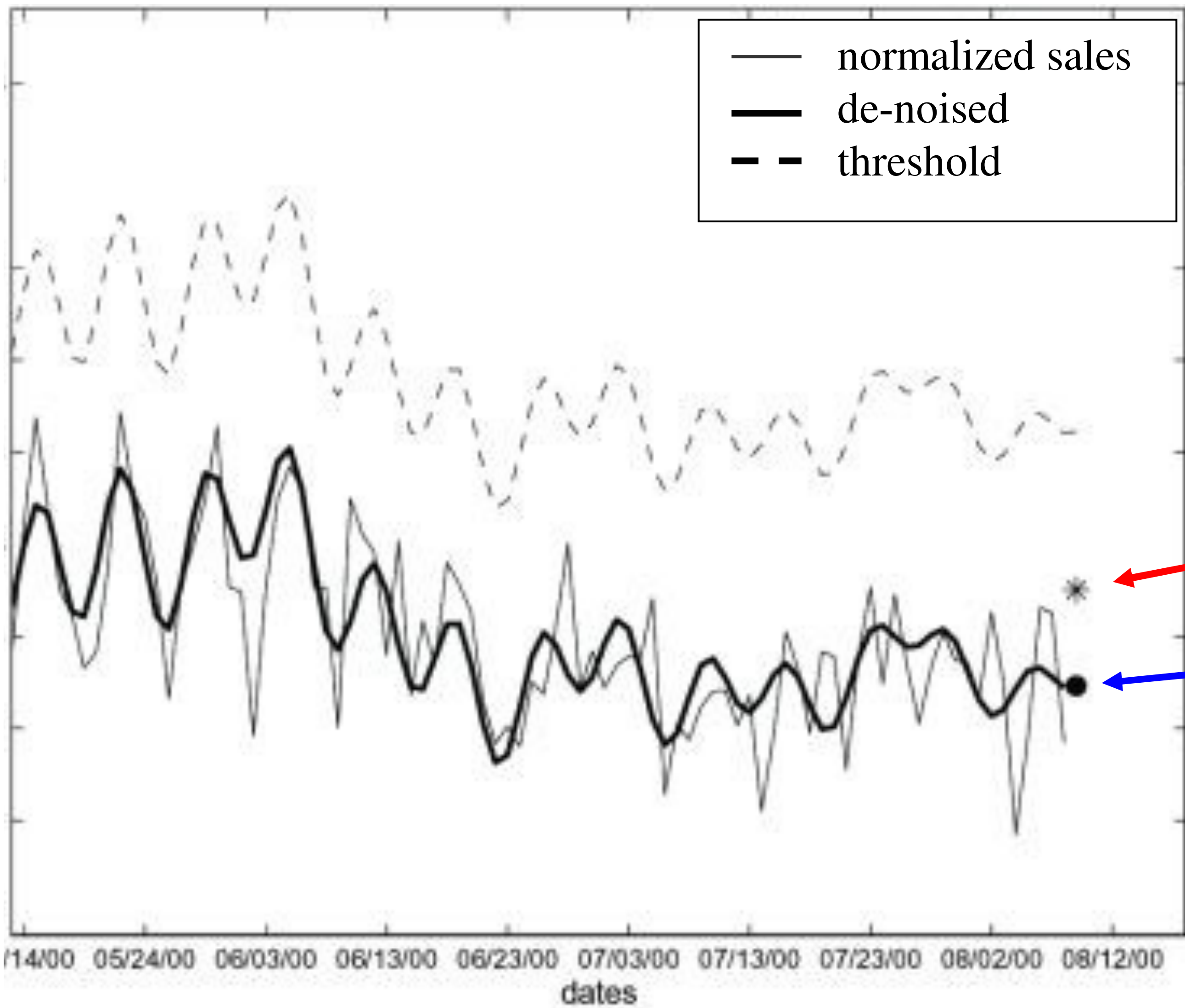
Discrepancy Checking



Discrepancy Checking: Example

Early statistical detection of anthrax outbreaks by tracking over-the-counter medication sales

Goldenberg, Shmueli, Caruana, and Fienberg



Actual value

Predicted value

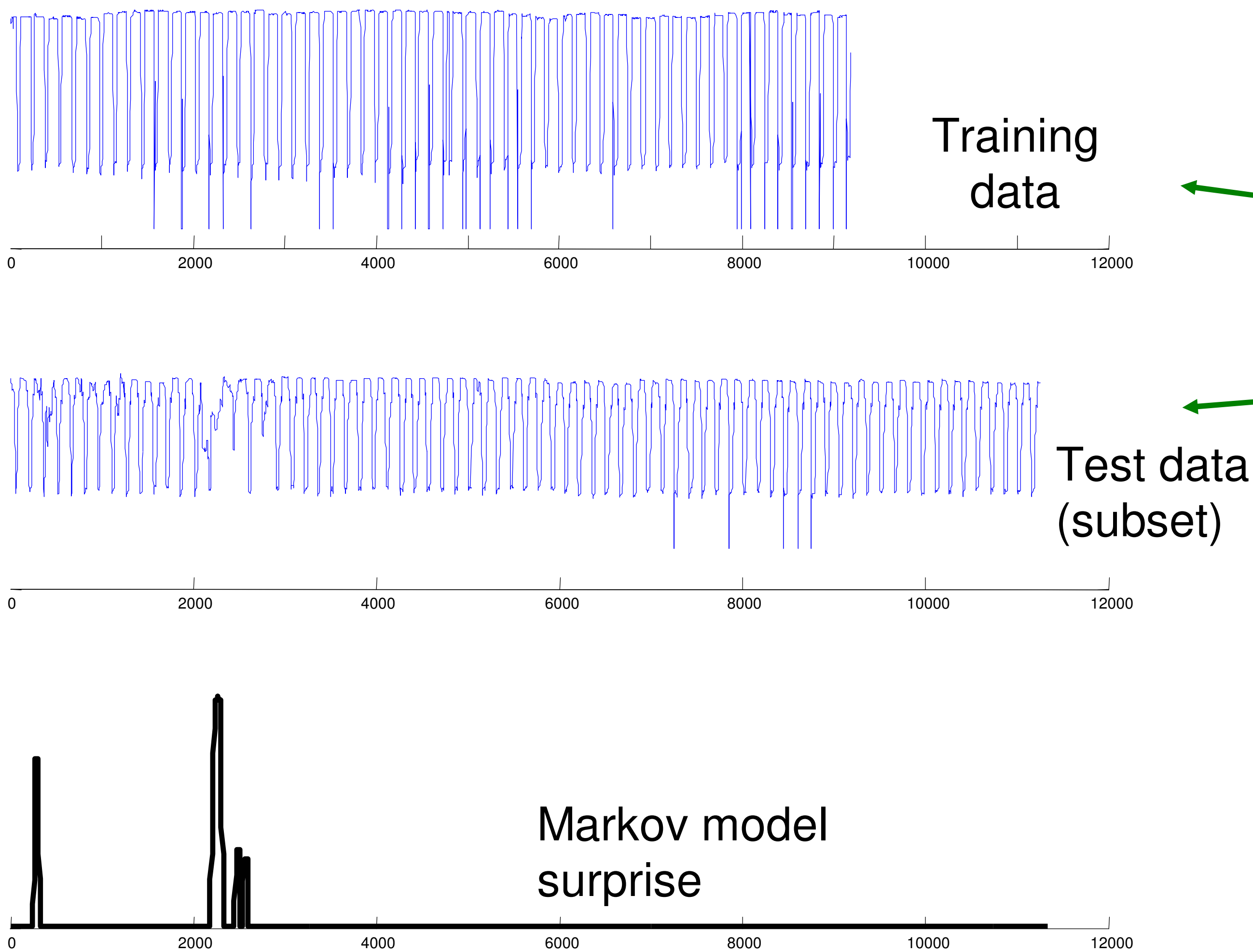
The actual value is greater than the predicted value, but still less than the threshold, so no alarm is sounded.

- Note that this problem has been solved for text strings

- You take a set of text which has been labeled "normal", you learn a Markov model for it.

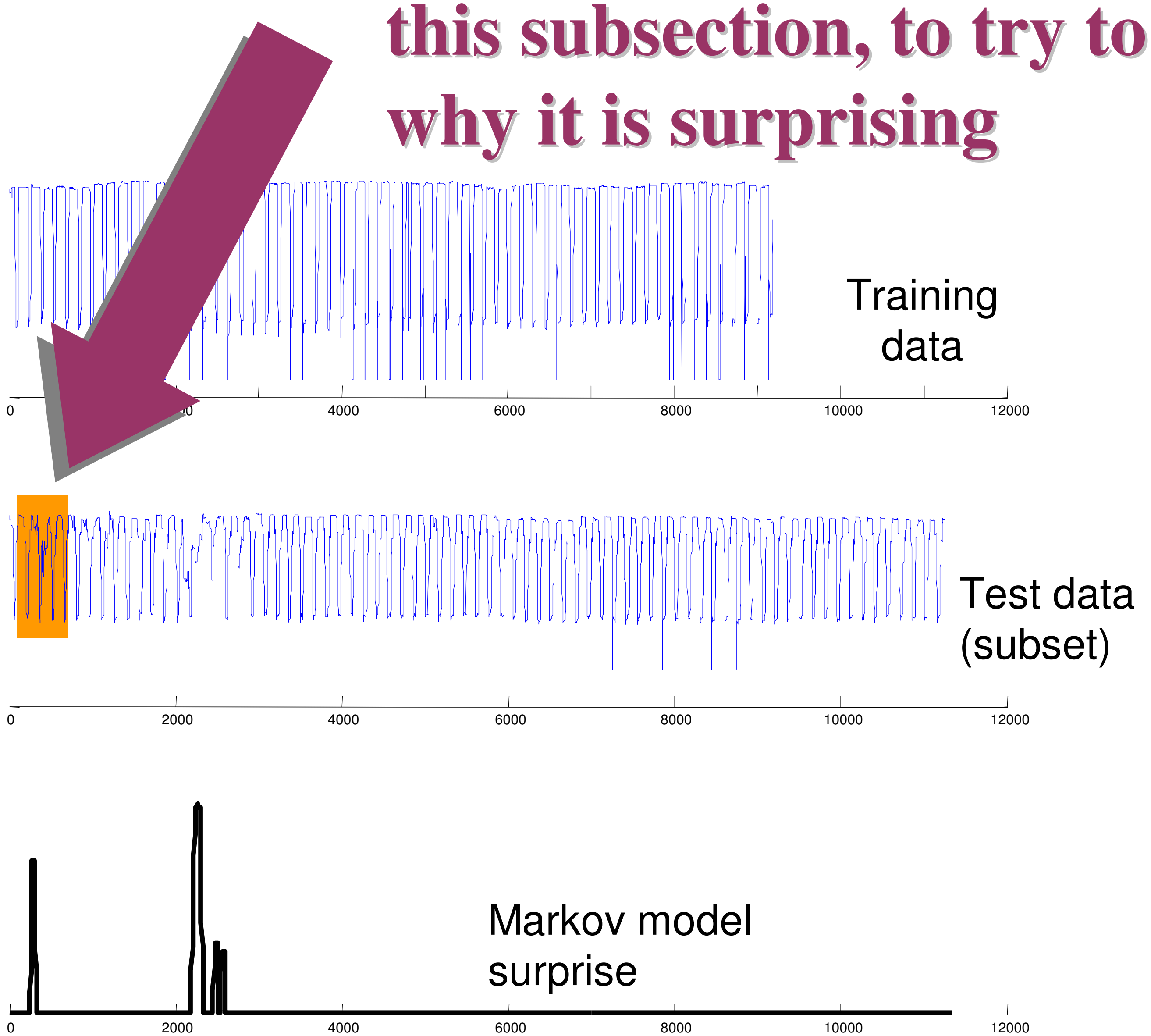
- Then, any future data that is not modeled well by the Markov model you annotate as *surprising*.

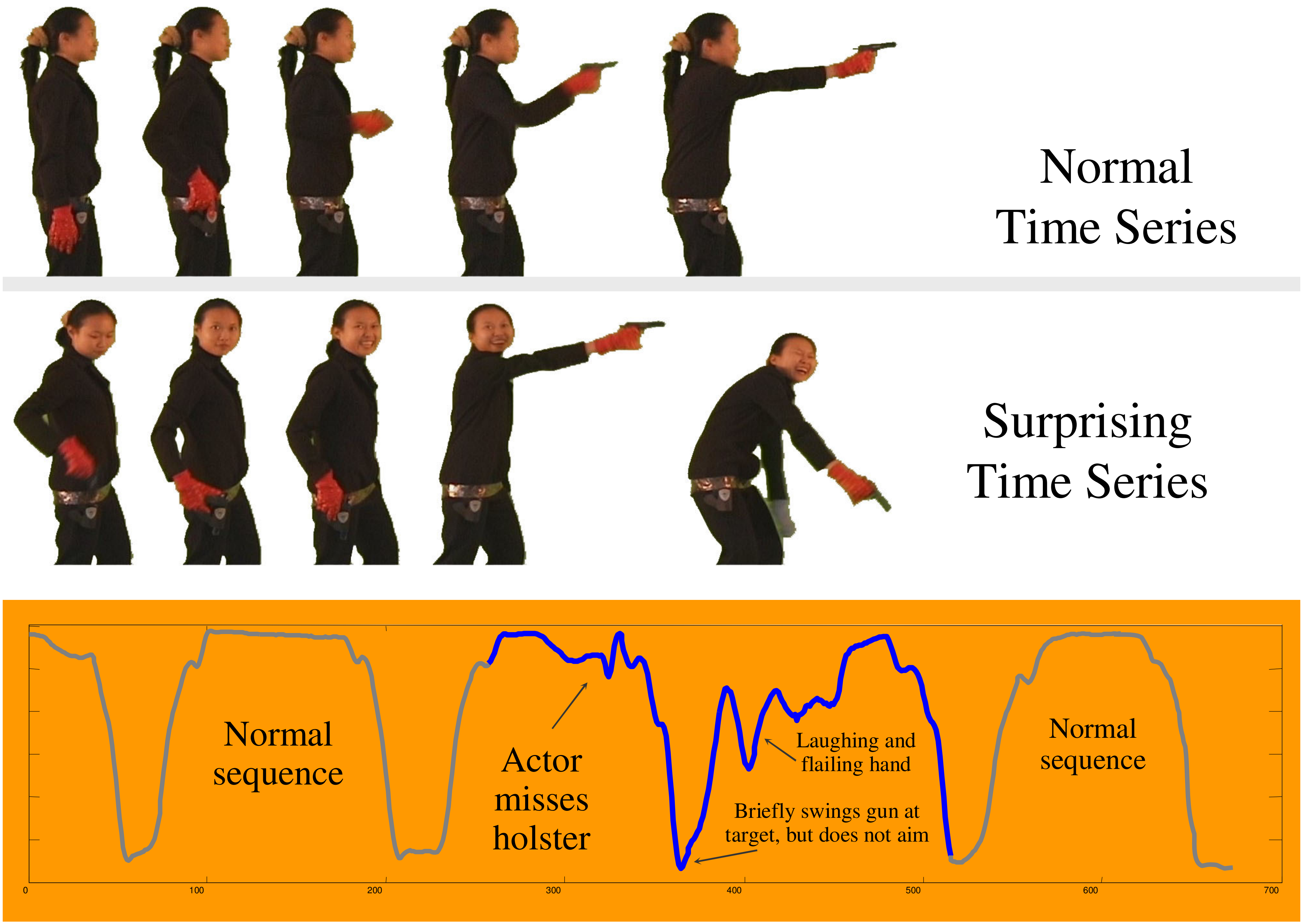
- Since we have just seen that we can convert time series to text (i.e SAX). Lets us quickly see if we can use Markov models to find surprises in time series…



These were converted to the symbolic representation.
I am showing the original data for simplicity

In the next slide we will zoom in on this subsection, to try to understand why it is surprising





Normal
Time Series
Surprising
Time Series
Normal
sequence
Actor
misses
holster
Laughing and
flailing hand
Briefly swings gun at
target, but does not aim
Normal
sequence
0
100
200
300
400
500
600
700

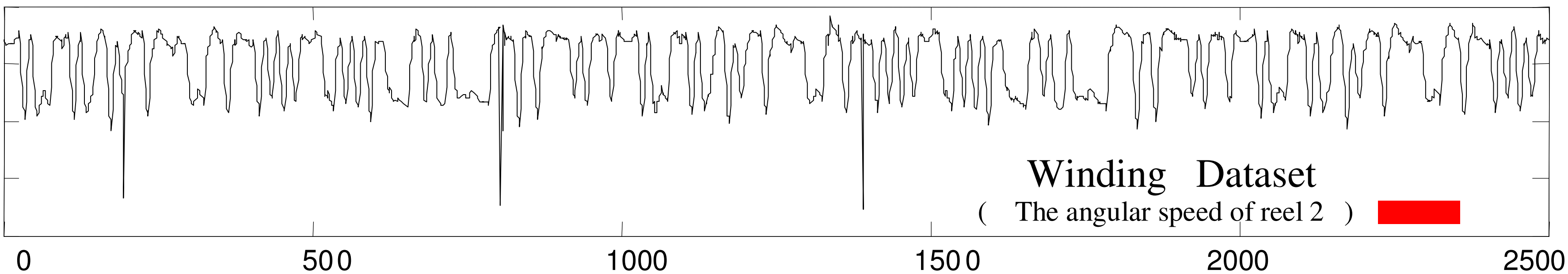
Anomaly (interestingness) detection

In spite of the nice example in the previous slide, the anomaly detection problem is wide open.

How can we find interesting patterns…

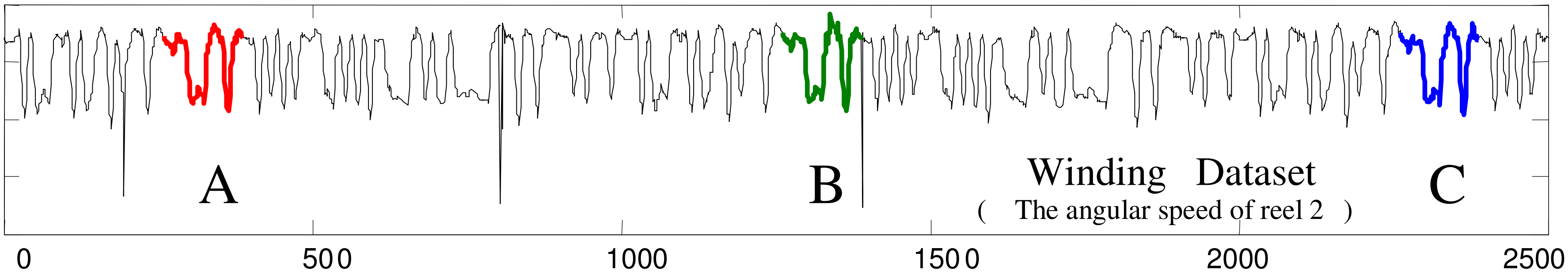
- Without (or with very few) false positives…
- In truly massive datasets...
- In the face of concept drift…
- With human input/feedback…
- With annotated data…

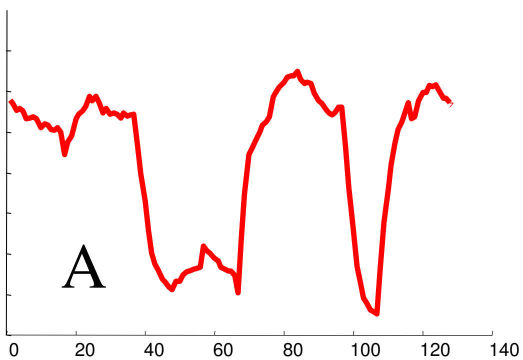
Time Series Motif Discovery
(finding repeated patterns)

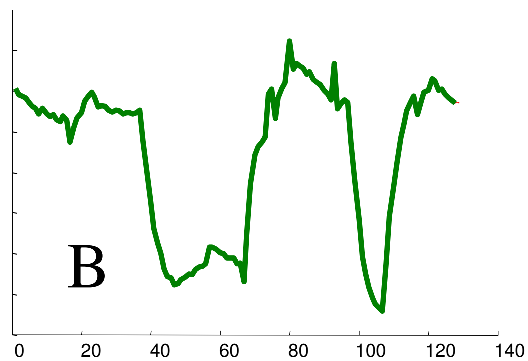


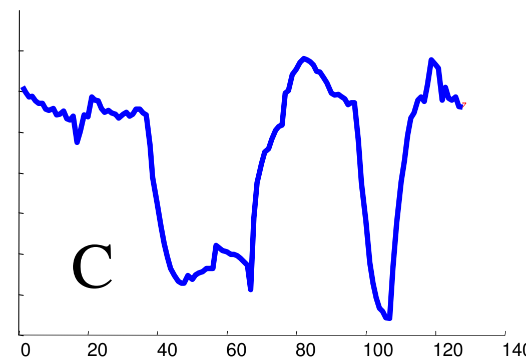
Time Series Motif Discovery

(finding repeated patterns)



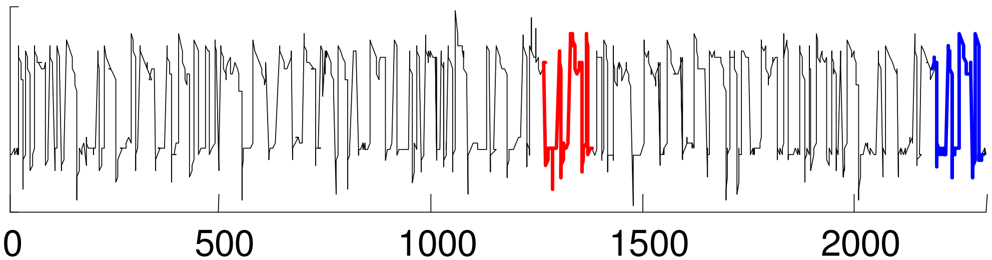


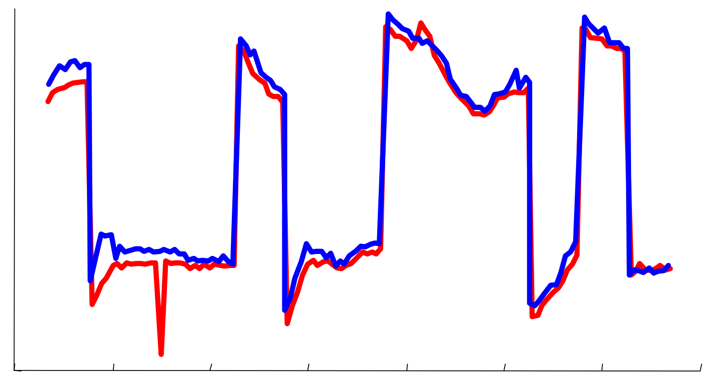




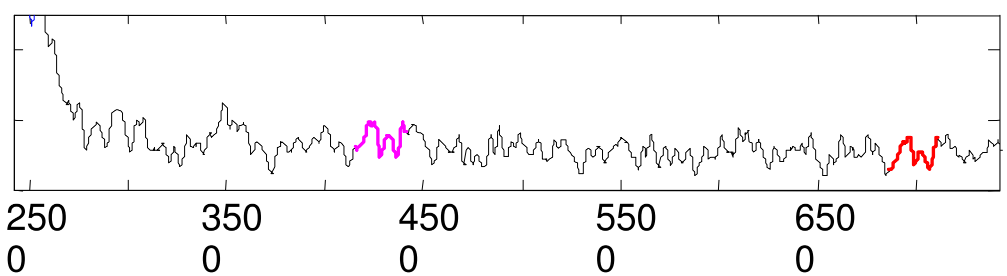
Some Examples of Real Motifs

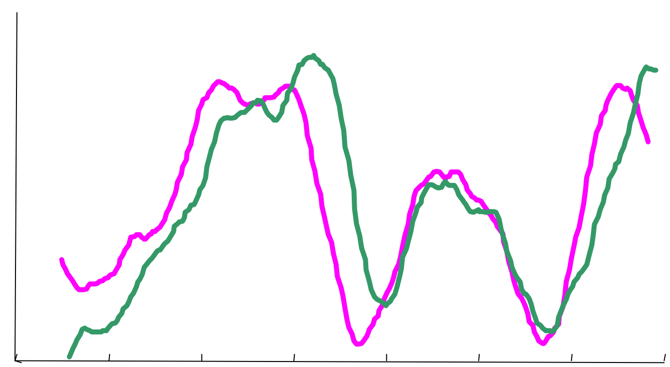
Motor 1 (DC Current)





Astrophysics (Photon Count)





Why Find Motifs?

· Mining **association rules** in time series requires the discovery of motifs. These are referred to as *primitive shapes* and *frequent patterns*.

· Several time series **classification algorithms** work by constructing typical prototypes of each class. These prototypes may be considered motifs.

· Many time series **anomaly/interestingness detection** algorithms essentially consist of modeling normal behavior with a set of typical shapes (which we see as motifs), and detecting future patterns that are dissimilar to all typical shapes.

· In **robotics**, Oates et al., have introduced a method to allow an autonomous agent to generalize from a set of qualitatively different *experiences* gleaned from sensors. We see these "*experiences*" as motifs.

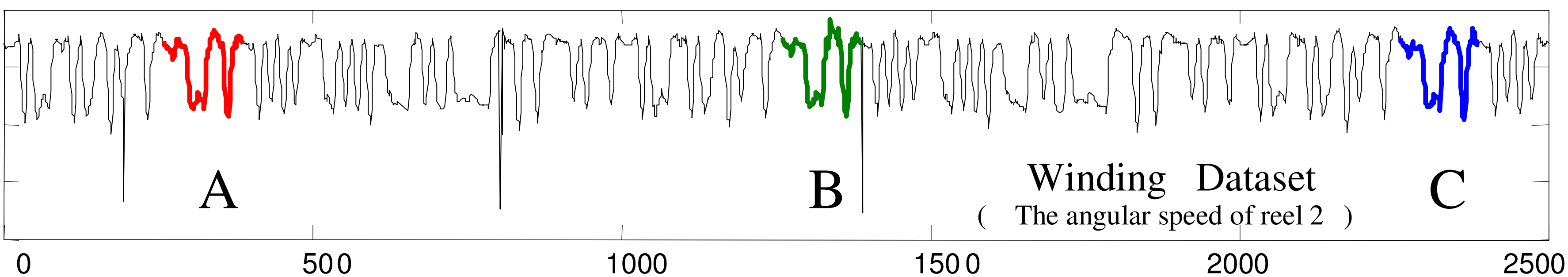
· In **medical data mining**, Caraca-Valente and Lopez-Chavarrias have introduced a method for characterizing a physiotherapy patient's recovery based of the discovery of *similar patterns*. Once again, we see these "*similar patterns*" as motifs.

• **Animation and video capture…** (Tanaka and Uehara, Zordan and Celly)

Motifs Discovery Challenges

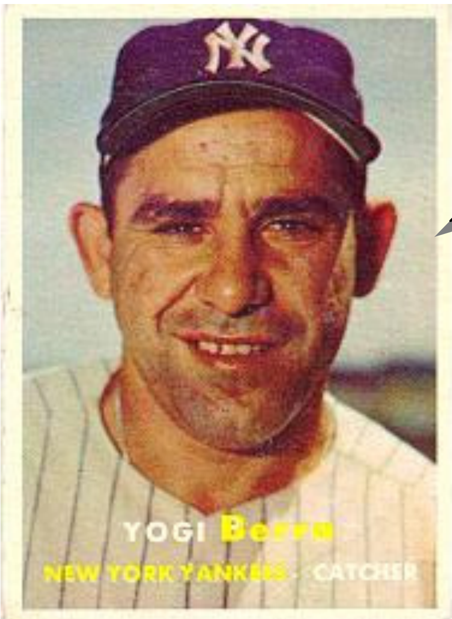
How can we find motifs…

- Without having to specify the length/other parameters
- In massive datasets
- While ignoring "background" motifs (ECG example)
- Under time warping, or uniform scaling
- While assessing their significance



Finding these 3 motifs requires about 6,250,000 calls to the Euclidean distance function

Time Series Prediction



Yogi Berra
1925 -

Prediction is hard, especially about the future

There are two kinds of time series prediction

- **Black Box**: Predict tomorrows electricity demand, given *only* the last ten years electricity demand.
- **White Box (side information)**: Predict tomorrows electricity demand, given the last ten years electricity demand *and* the weather report, *and* the fact that fact that the world cup final is on and…