Fair Use Agreement

This agreement covers the use of all slides on this CD-Rom, please read carefully.

- You may freely use these slides for teaching, if
  - You send me an email telling me the class number/ university in advance.
  - My name and email address appears on the first slide (if you are using all or most of the slides), or on each slide (if you are just taking a few slides).
- You may freely use these slides for a conference presentation, if
  - You send me an email telling me the conference name in advance.
  - My name appears on each slide you use.

- You may not use these slides for tutorials, or in a published work (tech report/ conference paper/ thesis/ journal etc). If you wish to do this, email me first, it is highly likely I will grant you permission.

(c) Eamonn Keogh, eamonn@cs.ucr.edu

Symbolic Representations of Time Series

Eamonn Keogh and Jessica Lin

Computer Science & Engineering Department
University of California - Riverside
Riverside,CA 92521
eamonn@cs.ucr.edu

Important! Read This!

These slides are from an early talk about SAX, some slides will make little sense out of context, but are provided here to give a quick intro to the utility of SAX. Read [1] for more details.

You may use these slides for any teaching purpose, so long as they are clearly identified as being created by Jessica Lin and Eamonn Keogh.

You may not use the text and images in a paper or tutorial without express prior permission from Dr. Keogh.

[1] Lin, J., Keogh, E., Lonardi, S. & Chiu, B. (2003). A Symbolic Representation of Time Series, with Implications for Streaming Algorithms. In *proceedings of the 8th ACM SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discovery.* San Diego, CA. June 13.

Outline of Talk

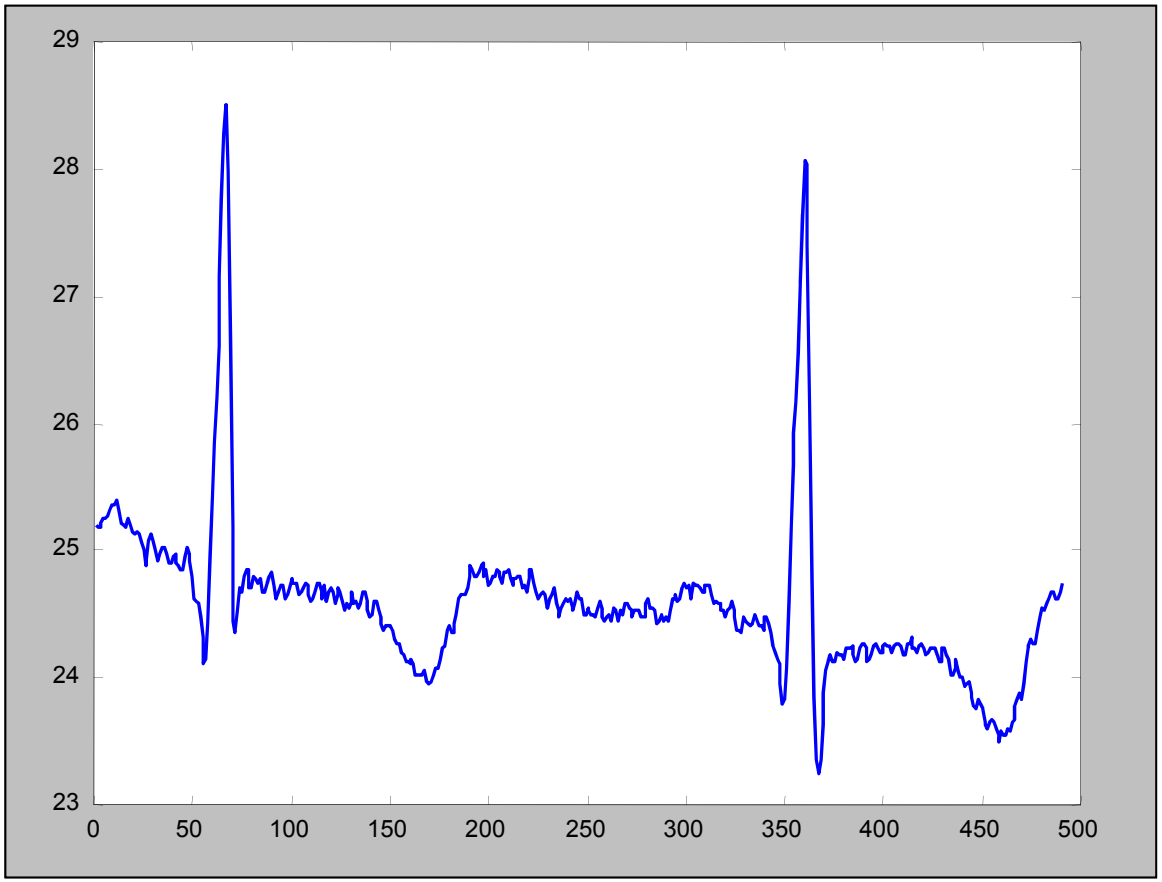
- **Prologue:** Background on Time Series Data Mining
- The importance of the **right** representation
- A new symbolic representation
    - motif discovery
    - anomaly detection
    - visualization

- Appendix: classification, clustering, indexing

What are Time Series?

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

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

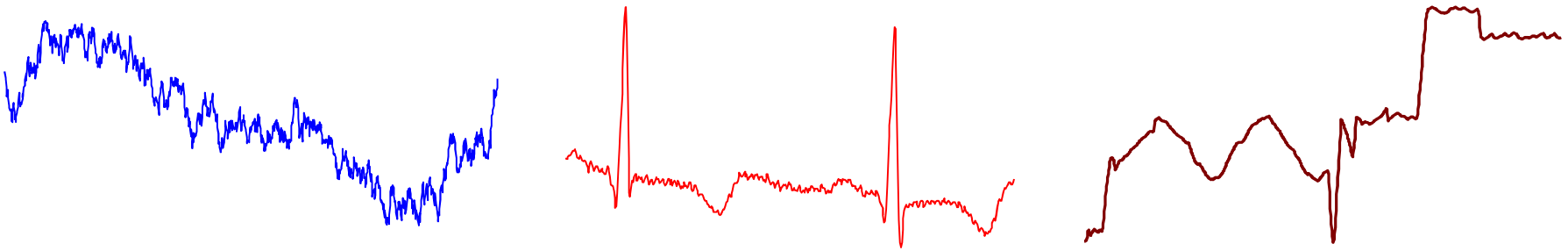


Time Series are Ubiquitous! I

People measure things...

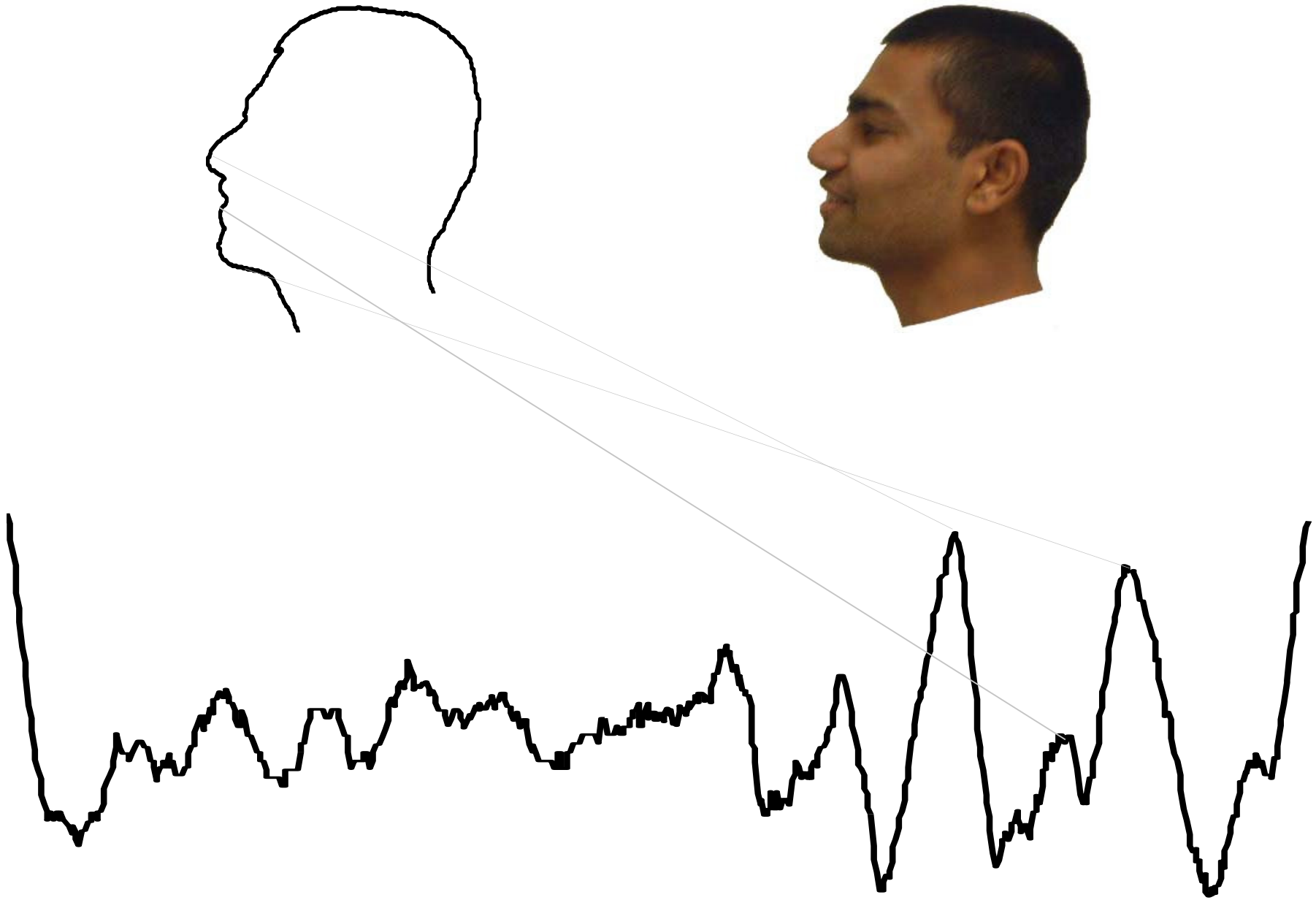
- *Schwarzeneggers popularity rating.*
- *Their blood pressure.*
- *The annual rainfall in New Zealand.*
- *The value of their Yahoo stock.*
- *The number of web hits per second.*

… and things change over time.



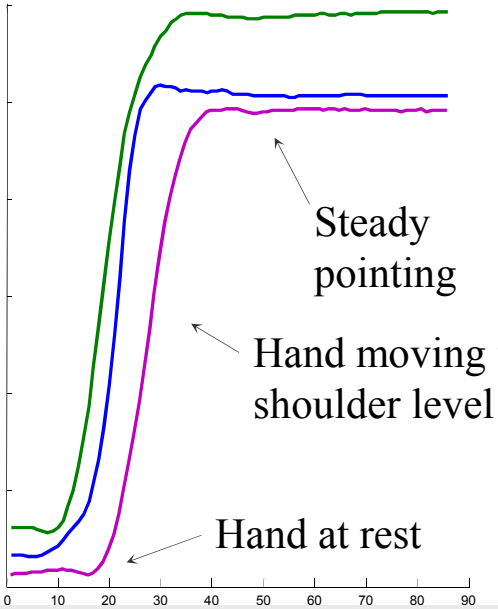
Thus time series occur in virtually every medical, scientific and businesses domain.

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

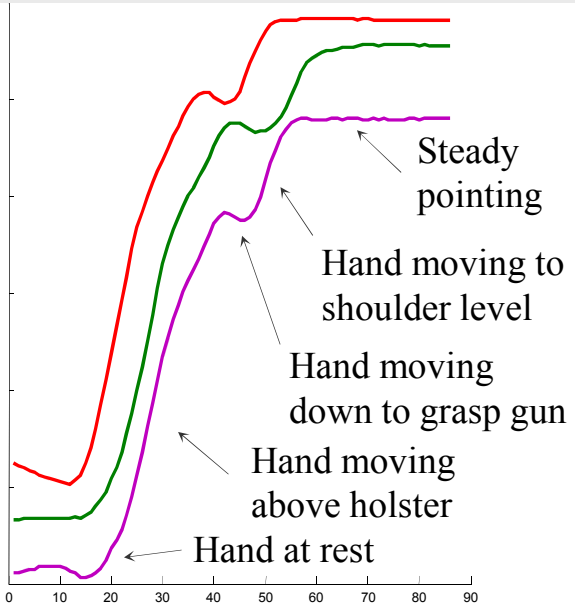


Video data, may best be thought of as time series…

Point



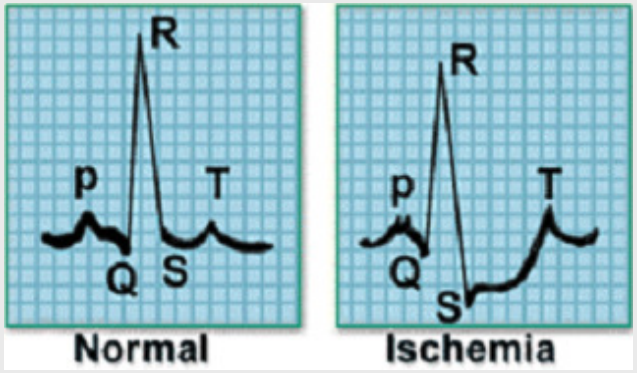
Gun-Draw



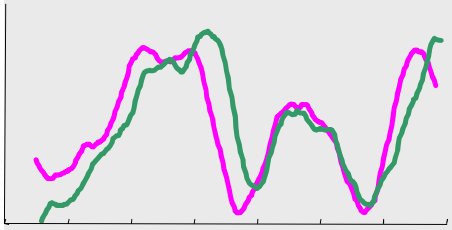
What do we want to do with the time series data?

Clustering

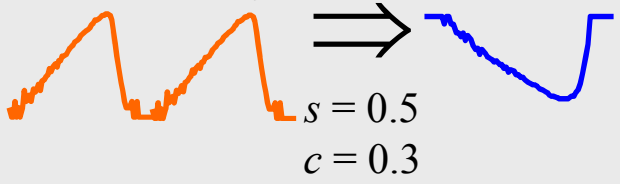
Classification



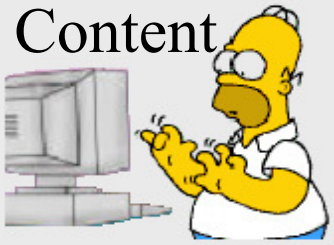
Motif Discovery

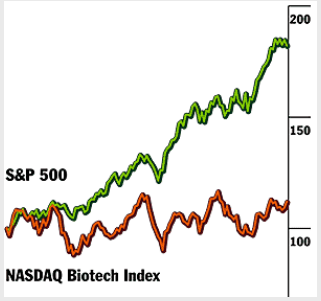


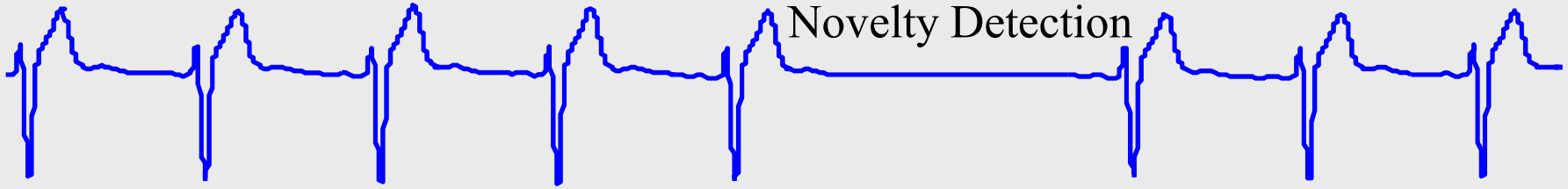
Rule Discovery



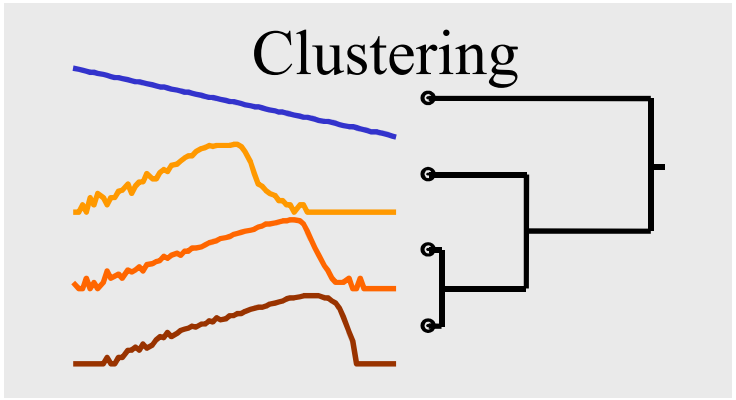
Query by Content

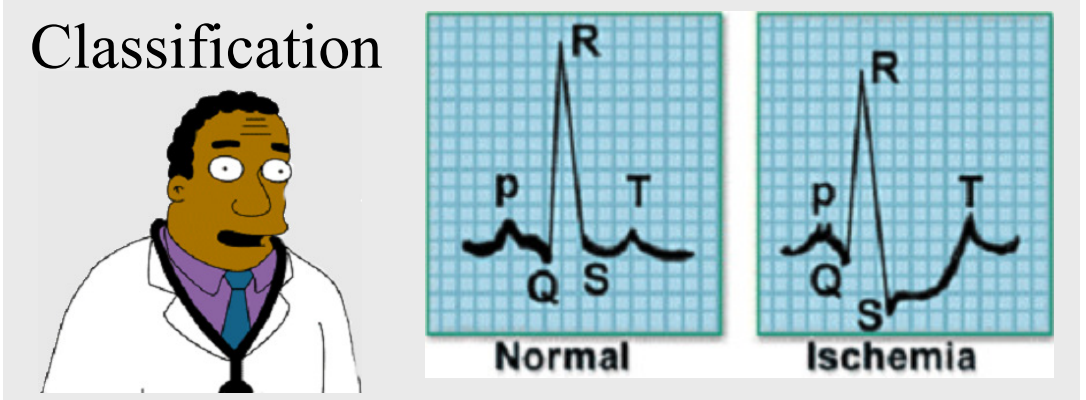


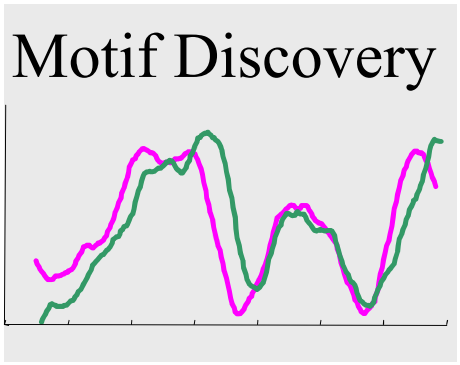


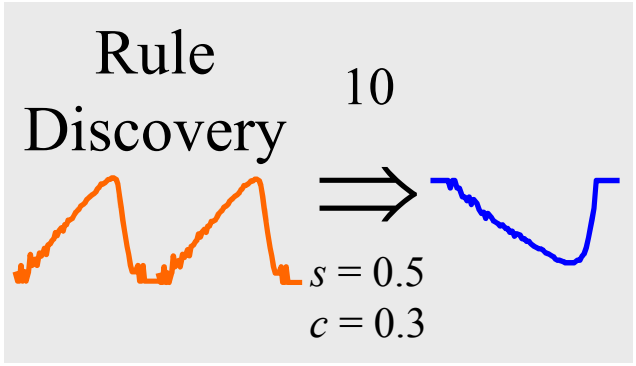


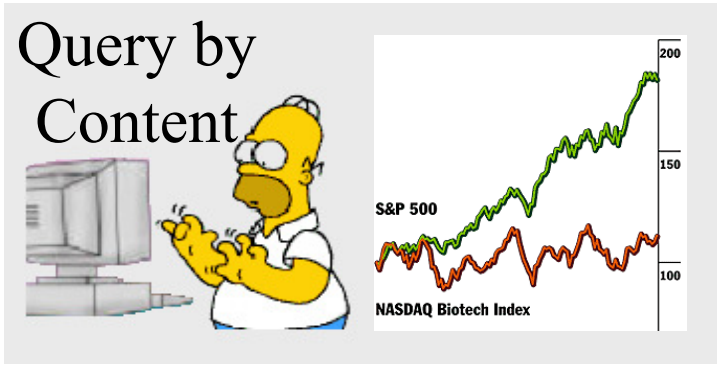
All these problems require **similarity** matching

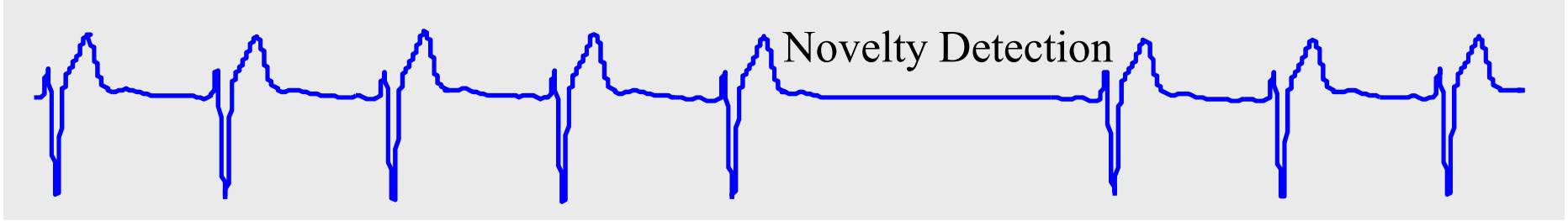












Euclidean Distance Metric

Given two time series

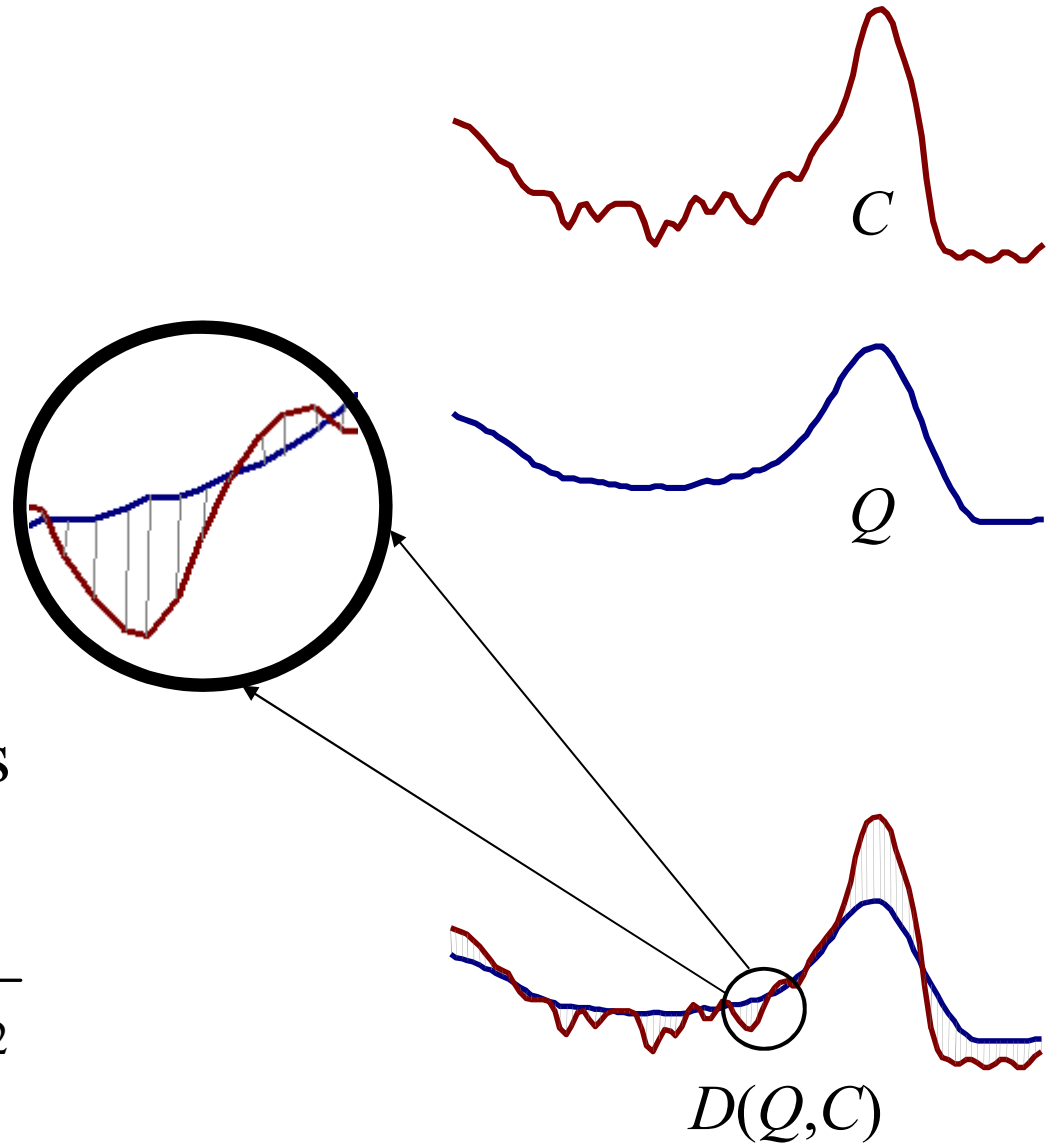
$$Q = q_1 \ldots q_n$$

and

$$C = c_1 \ldots c_n$$

their Euclidean distance is defined as:

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

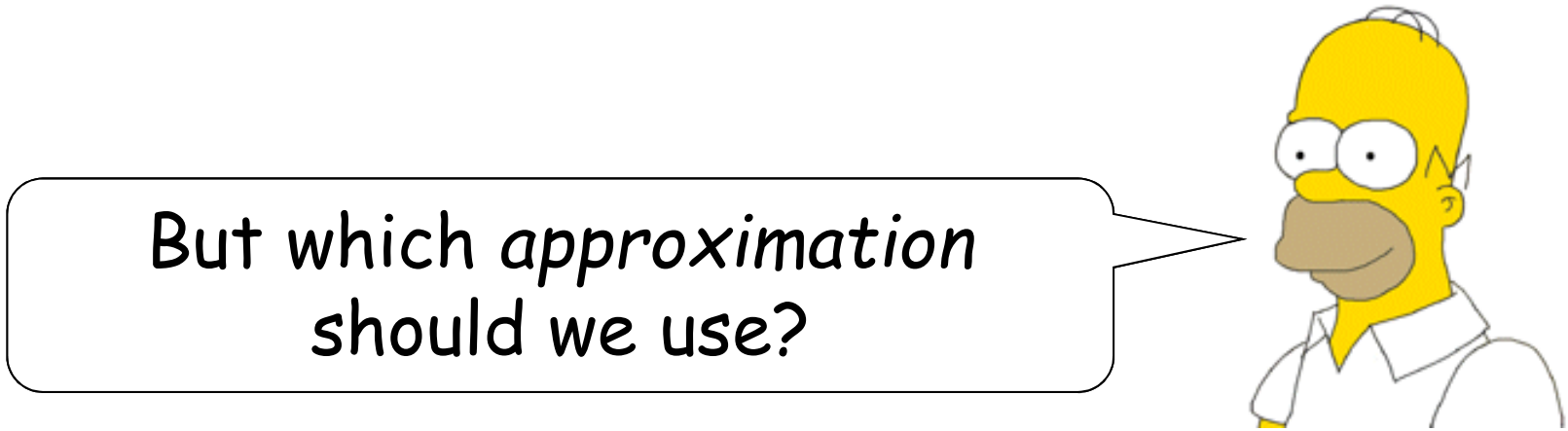


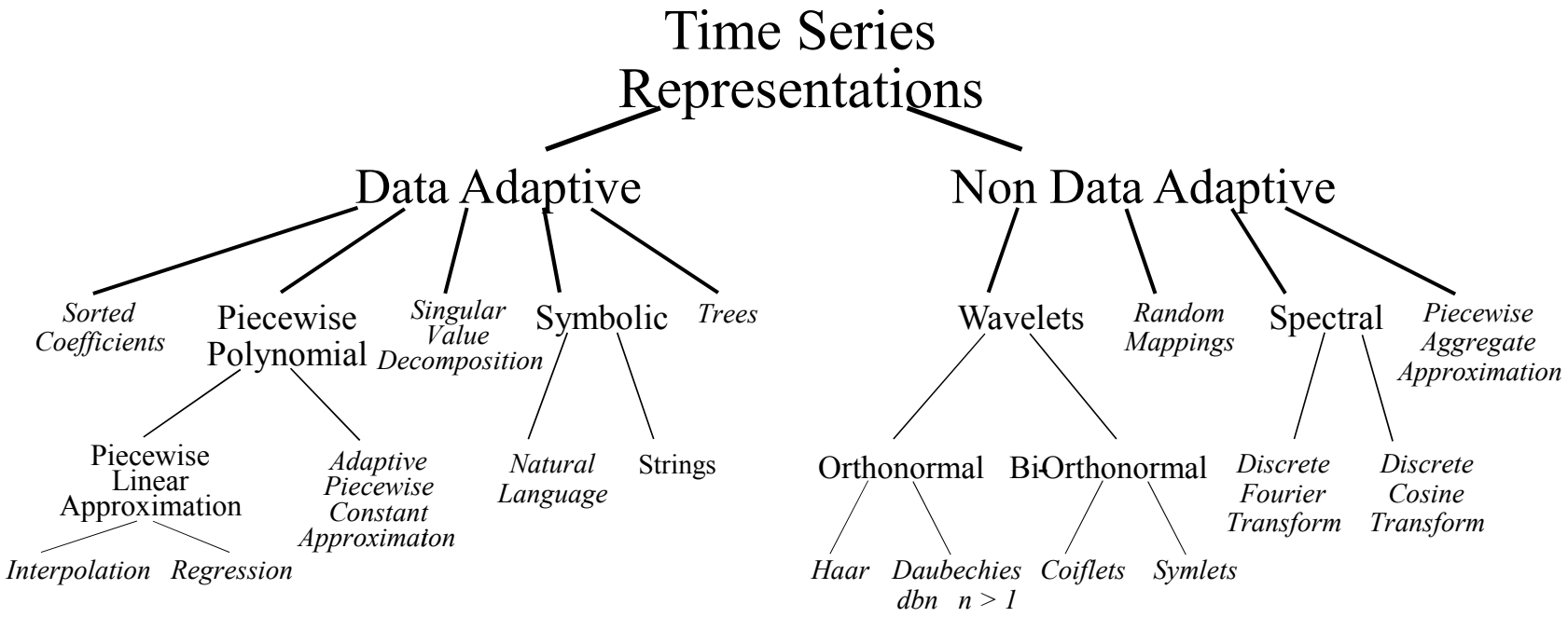
The Generic Data Mining Algorithm

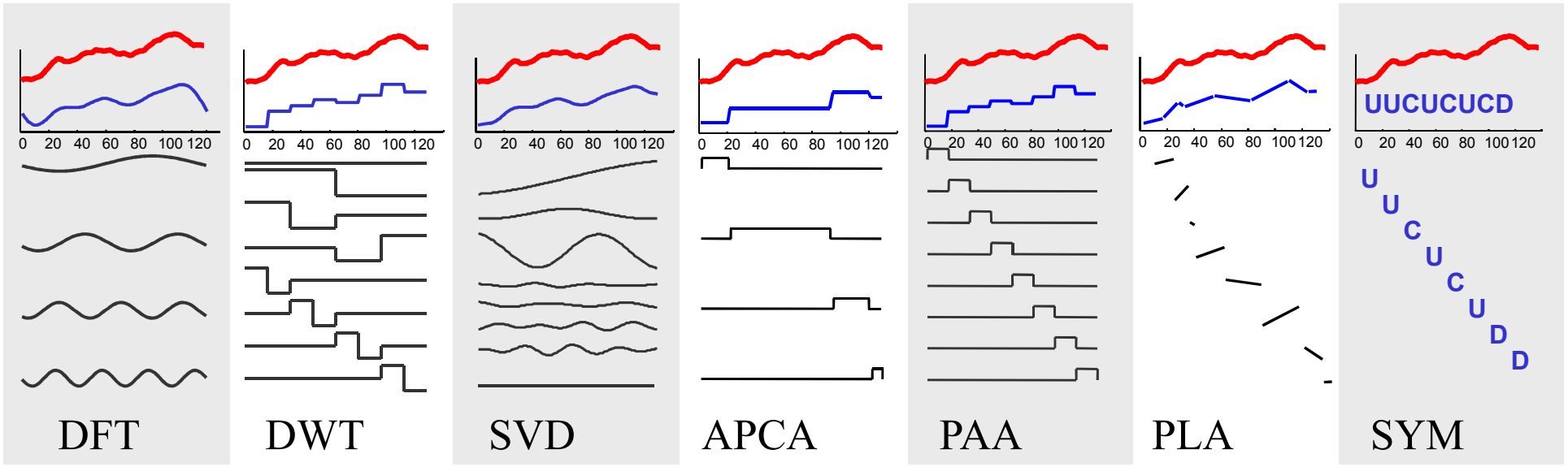
• Create an *approximation* of the data, which will fit in main memory, yet retains the essential features of interest

• 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
Data Adaptive
Non Data Adaptive
Sorted Coefficients
Piecewise Polynomial
Singular Value Decomposition
Symbolic
Trees
Piecewise Linear Approximation
Adaptive Piecewise Constant Approximaton
Natural Language
Strings
Interpolation
Regression
Wavelets
Random Mappings
Spectral
Piecewise Aggregate Approximation
Orthonormal
Bi-Orthonormal
Discrete Fourier Transform
Discrete Cosine Transform
Haar
Daubechies dbn n > 1
Coiflets
Symlets


0 20 40 60 80 100 120
UUCUCUCD
DFT
DWT
SVD
APCA
PAA
PLA
SYM

The Generic Data Mining Algorithm (revisited)

• Create an *approximation* of the data, which will fit in main memory, yet retains the essential features of interest

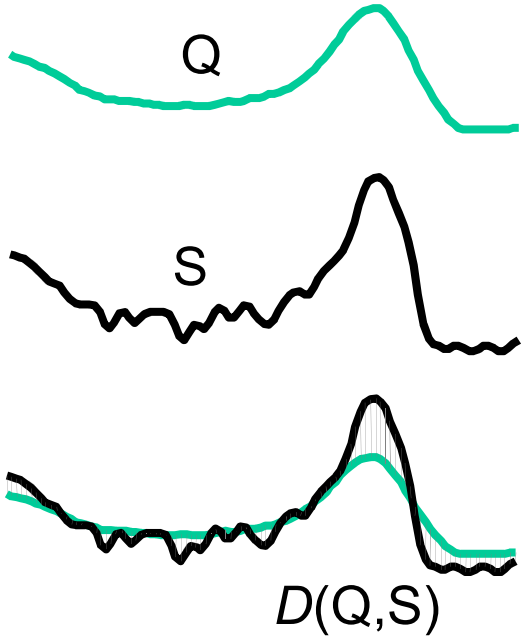
• Approximately solve the problem at hand in main memory

• 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

This *only* works if the approximation allows **lower bounding**

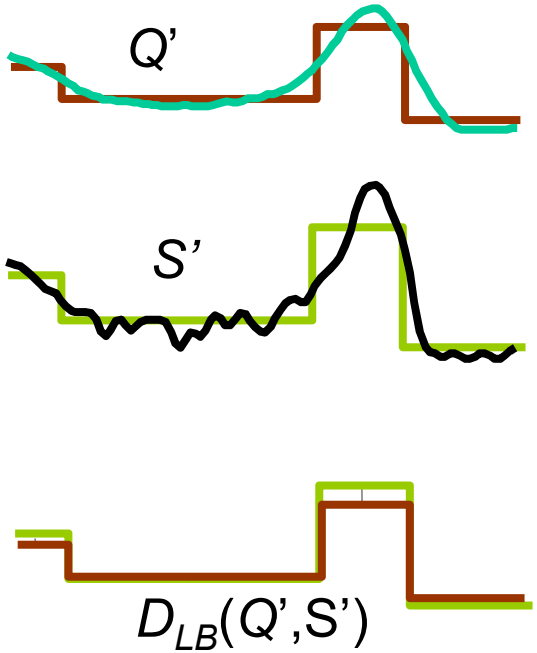
What is lower bounding?

Exact (Euclidean) distance D(Q,S)



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

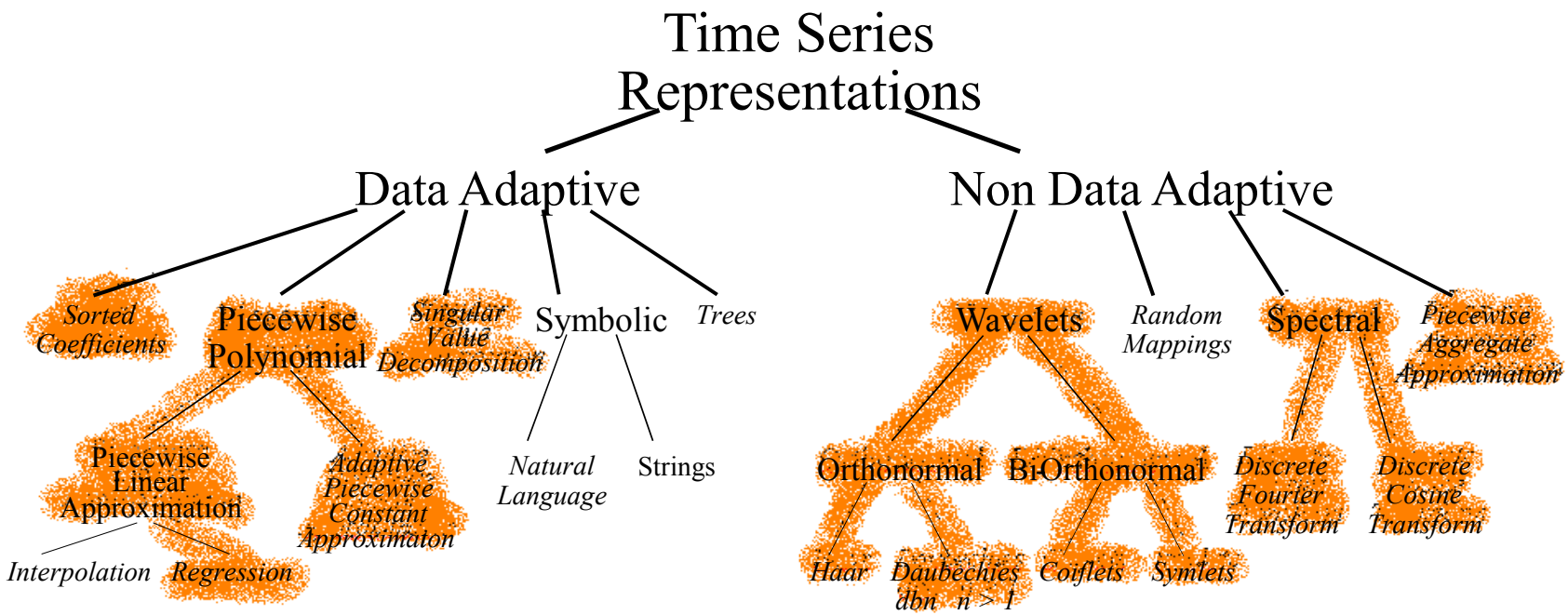
Lower bounding distance $D_{LB}(Q,S)$



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

Lower bounding means that for all Q and S, we have…

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



We can live without "*trees*", "*random mappings*" and "*natural language*", but it would be nice if we could lower bound *strings (symbolic or discrete approximations)*…

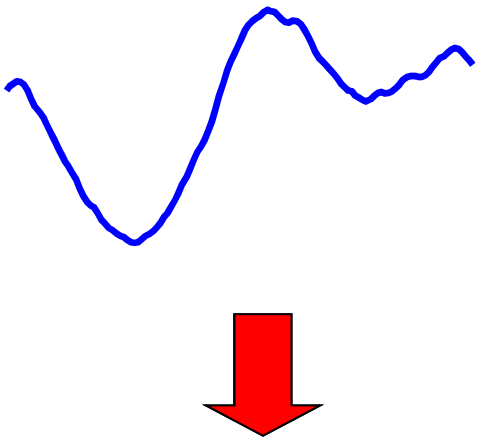
A lower bounding symbolic approach would allow data miners to…

- Use suffix trees, hashing, markov models etc
- Use text processing and bioinformatic algorithms

We have created the first symbolic representation of time series, that allows…

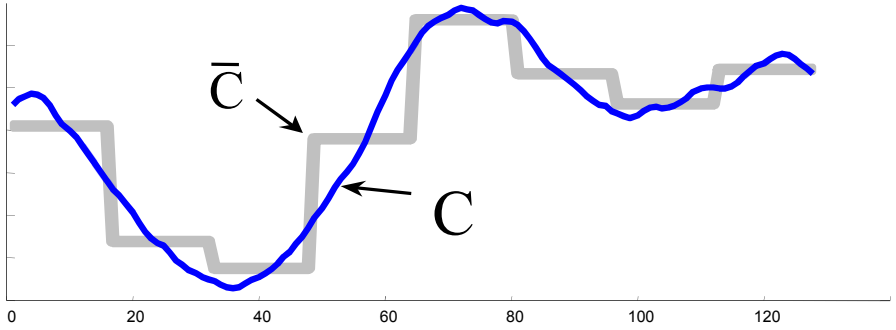
- Lower bounding of Euclidean distance
- Dimensionality Reduction
- Numerosity Reduction

We call our representation SAX
Symbolic **A**ggregate Appro**X**imation



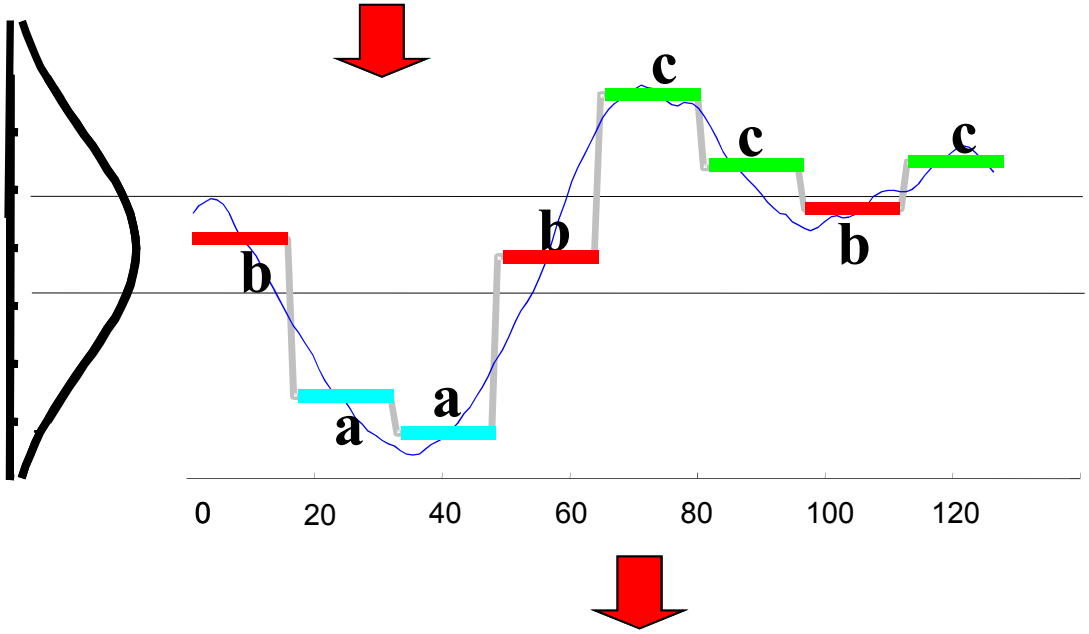
baabccbc

How do we obtain SAX?

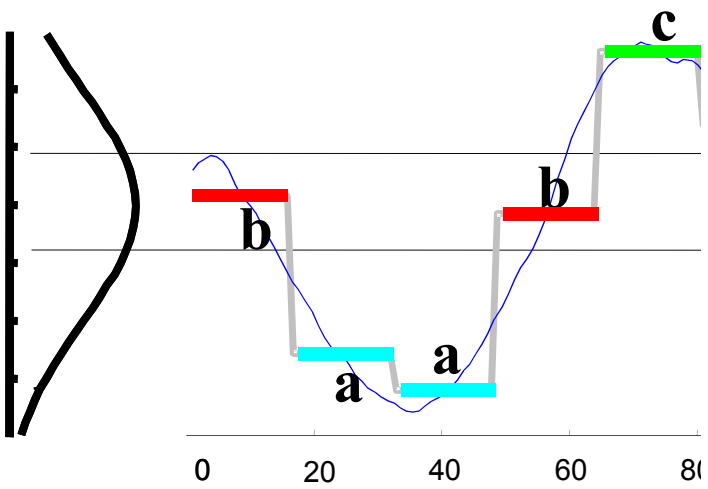


First convert the time series to PAA representation, then convert the PAA to symbols

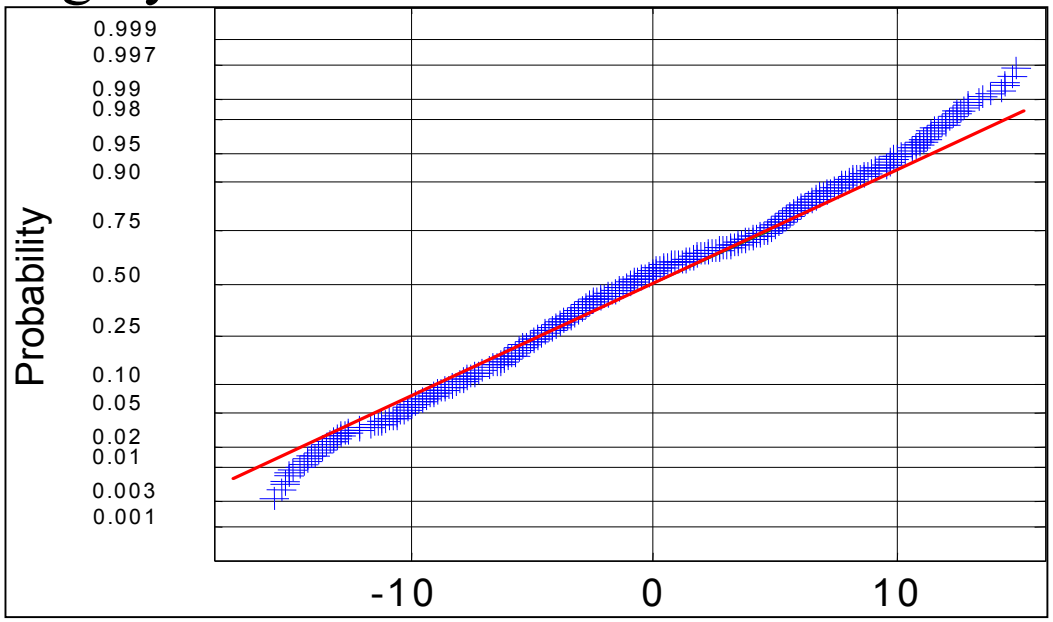
It take linear time



baabccbc

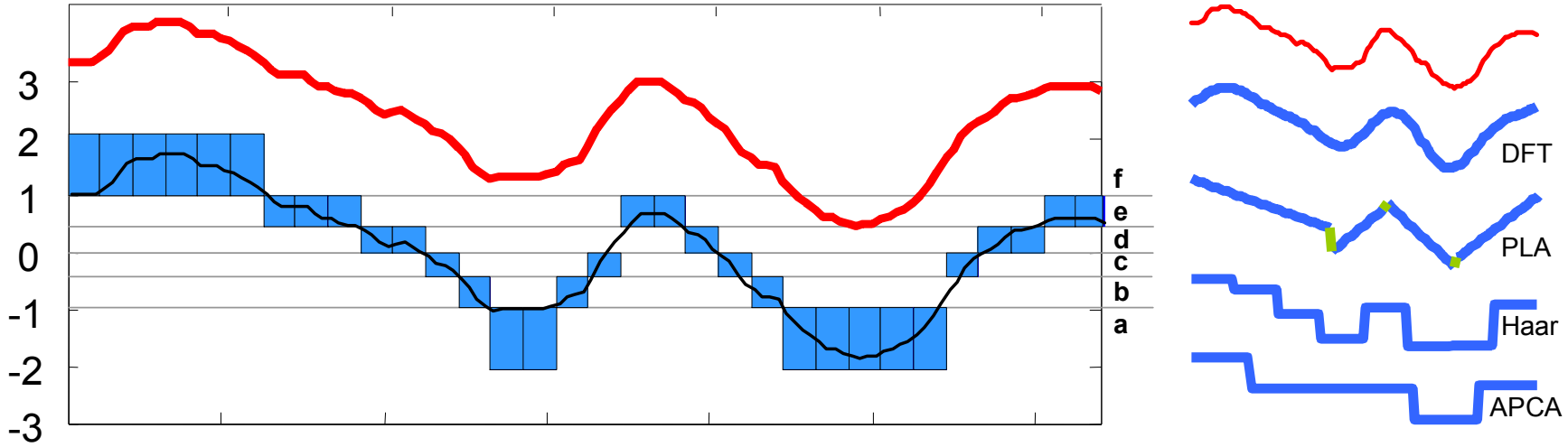


Time series subsequences tend to have a highly Gaussian distribution



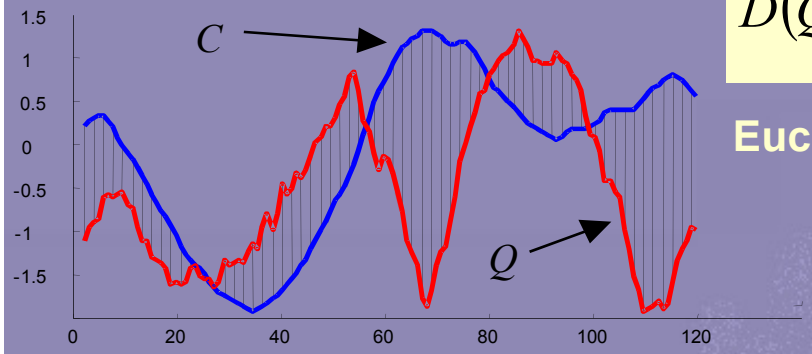
A normal probability plot of the (cumulative) distribution of values from subsequences of length 128.

Visual Comparison



A raw time series of length 128 is transformed into the word "**fffffeeeddcbaabceedcbaaaaacddee**."

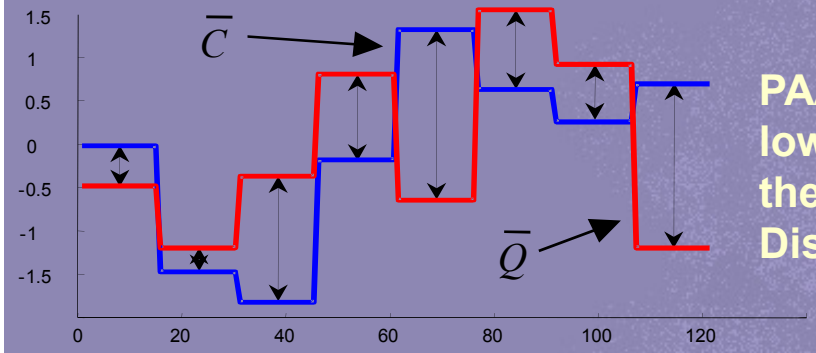
- We can use more symbols to represent the time series since each symbol requires fewer bits than real-numbers (float, double)



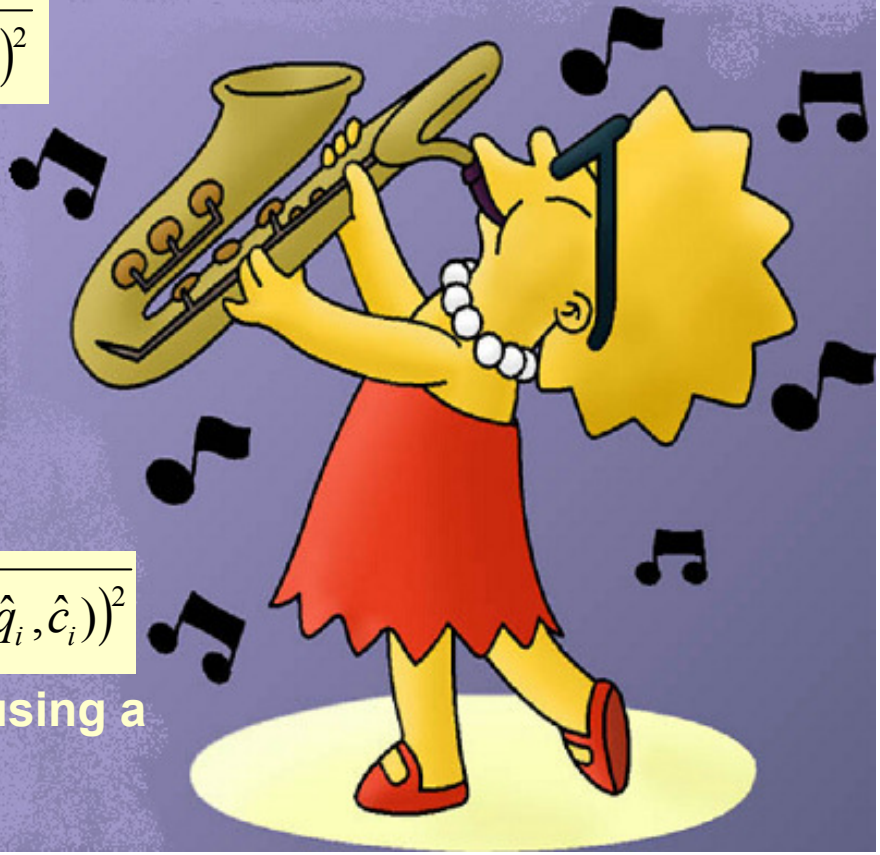
$$D(Q,C) \equiv \sqrt{\sum_{i=1}^{n} (q_i - c_i)^2}$$

Euclidean Distance

$$DR(\overline{Q},\overline{C}) \equiv \sqrt{\frac{n}{w}} \sqrt{\sum_{i=1}^{w} (\overline{q}_i - \overline{c}_i)^2}$$



PAA distance lower-bounds the Euclidean Distance



$$\hat{C} = \mathbf{baabccbc}$$

$$\hat{Q} = \mathbf{babcacca}$$

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

dist() can be implemented using a table lookup.

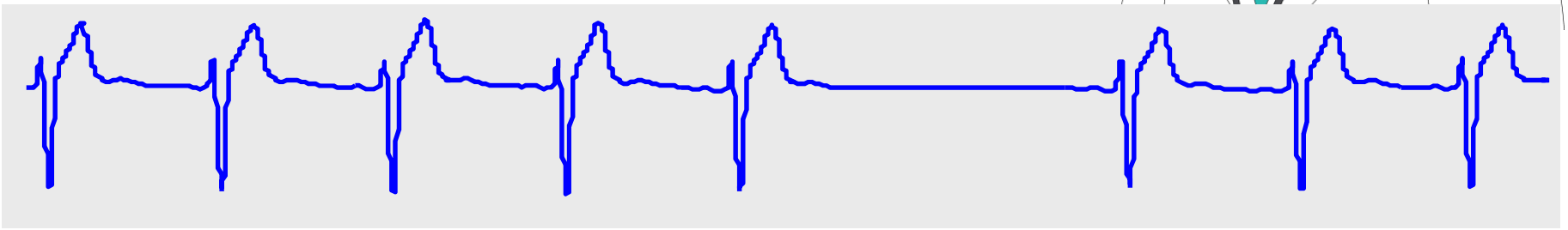
SAX is just as good as other representations, or working on the raw data for most problems **(Slides shown at the end of this presentation)**

Now let us consider SAX for two hot problems, novelty detection and motif discovery

We will start with novelty detection…

Novelty Detection

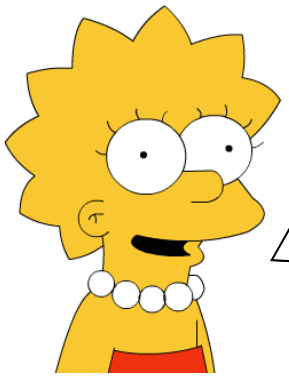
- Fault detection
- Interestingness detection
- Anomaly detection
- Surprisingness detection



…note that this problem should not be confused with the relatively simple problem of outlier detection. Remember Hawkins famous definition of an outlier…

… an outlier is an observation that deviates so much from other observations as to arouse suspicion that it was generated from a different mechanism…

Douglas M. Hawkins



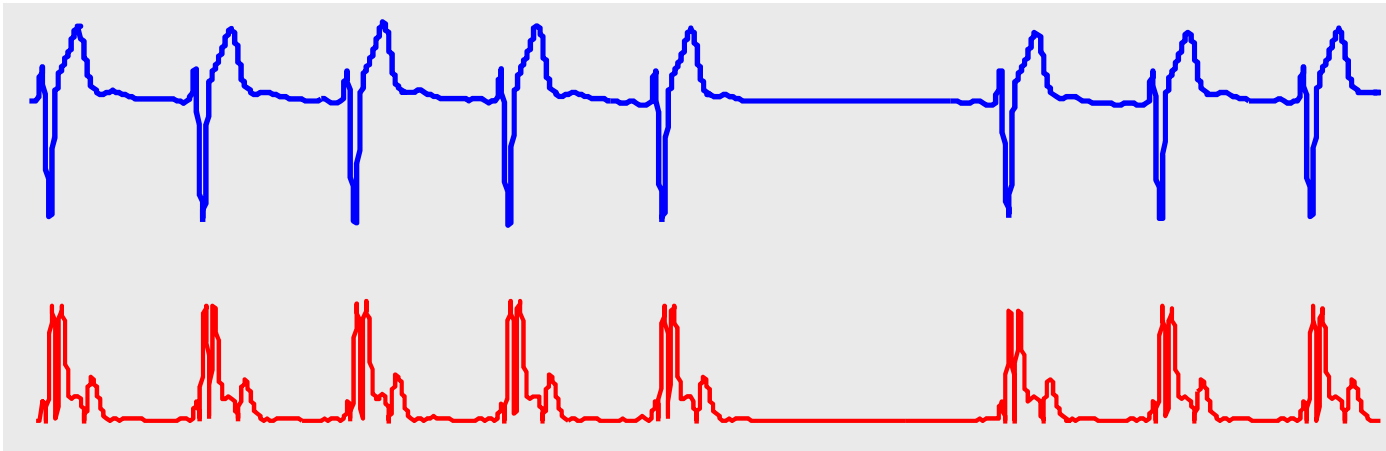
Thanks Doug, the check is in the mail.

We are **not** interested in finding individually surprising *datapoints*, we are interested in finding surprising *patterns*.

Lots of good folks have worked on this, and closely related problems. It is referred to as the detection of "Aberrant Behavior[1]", "Novelties[2]", "Anomalies[3]", "Faults[4]", "Surprises[5]", "Deviants[6]" ,"Temporal Change[7]", and "Outliers[8]".

1. Brutlag, Kotsakis et. al.
2. Daspupta et. al., Borisyuk et. al.
3. Whitehead et. al., Decoste
4. Yairi et. al.
5. Shahabi, Chakrabarti
6. Jagadish et. al.
7. Blockeel et. al., Fawcett et. al.
8. Hawkins.

Arrr... what be wrong with current approaches?



The blue time series at the top is a normal healthy human electrocardiogram with an artificial "*flatline*" added. The sequence in red at the bottom indicates how *surprising* local subsections of the time series are under the measure introduced in Shahabi et. al.

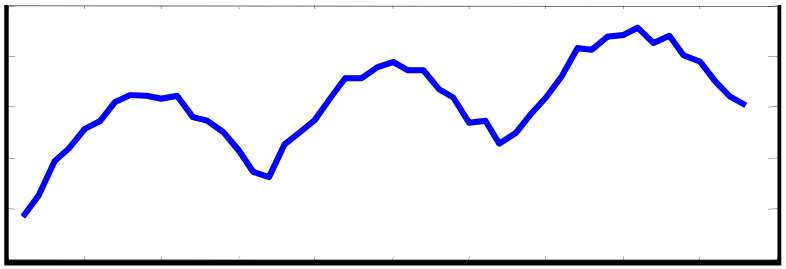
Our Solution

Based on the following intuition, a pattern is surprising if its frequency of occurrence is greatly different from that which we expected, given previous experience…

This is a nice intuition, but useless unless we can more formally define it, and calculate it efficiently

Note that unlike all previous attempts to solve this problem, our notion surprisingness of a pattern is not tied exclusively to its shape. Instead it depends on the difference between the shape's expected frequency and its observed frequency.

For example consider the familiar *head and shoulders* pattern shown below...



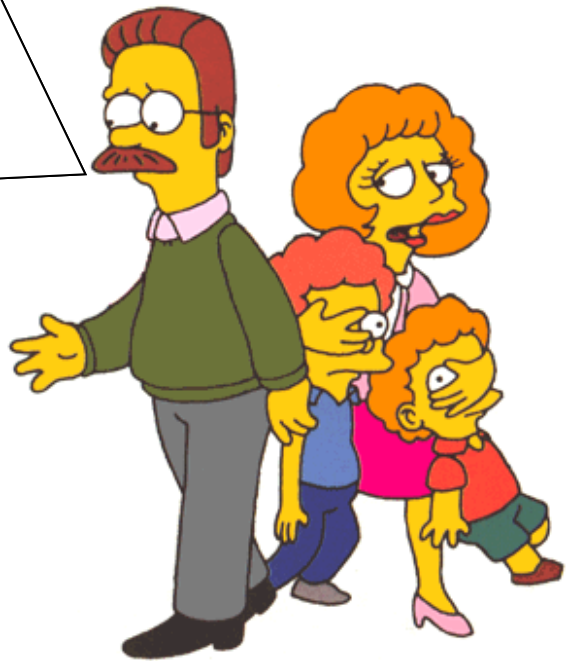
The existence of this pattern in a stock market time series should not be consider surprising since they are known to occur (even if only by chance). However, if it occurred ten times this year, as opposed to occurring an average of twice a year in previous years, our measure of surprise will flag the shape as being surprising. Cool eh?

The pattern would also be surprising if its frequency of occurrence is less than expected. Once again our definition would flag such patterns.

We call our algorithm… Tarzan!

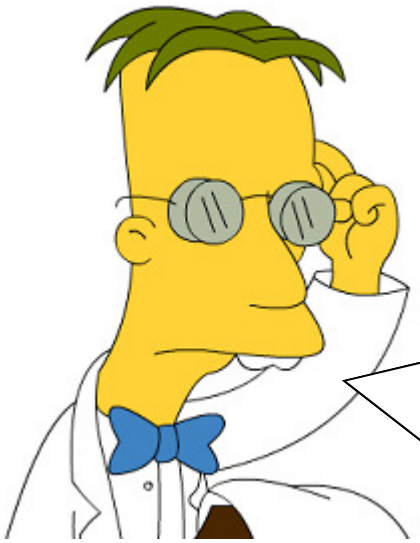
"Tarzan" is not an acronym. It is a pun on the fact that the heart of the algorithm relies comparing two suffix trees, "tree to tree"!

Homer, I hate to be a fuddy-duddy, but could you put on some pants?

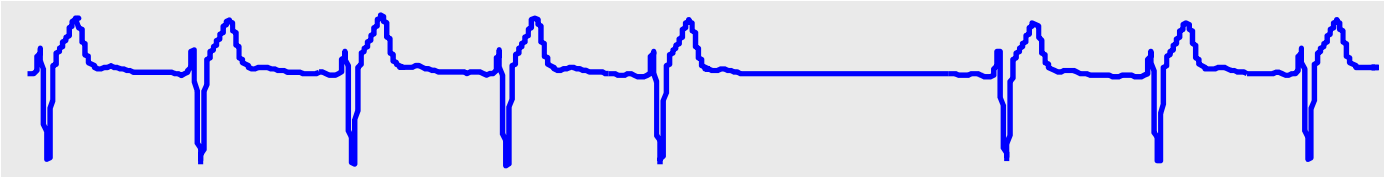


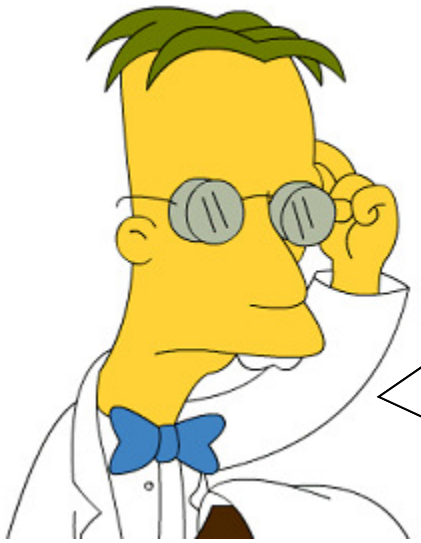
Tarzan (R) is a registered trademark of Edgar Rice Burroughs, Inc.

We begin by defining some terms… Professor Frink?



Definition 1: A time series pattern P, extracted from database X is surprising relative to a database R, if the probability of its occurrence is greatly different to that expected by chance, assuming that R and X are created by the same underlying process.



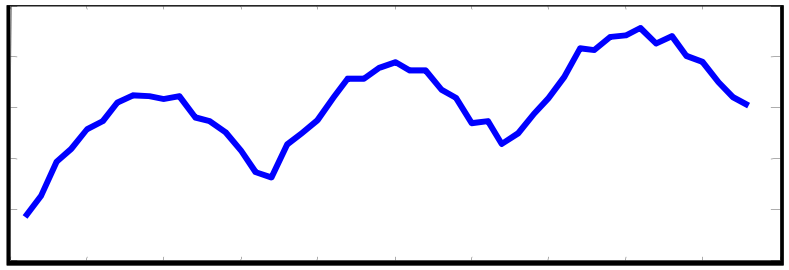


Definition 1: A time series pattern P, extracted from database X is surprising relative to a database R, if the probability of occurrence is greatly different to that expected by chance, assuming that R and X are created by the same underlying process.

But you can never know the probability of a pattern you have never seen!

And probability isn't even defined for real valued time series!

We need to discretize the time series into symbolic strings… SAX!!



aaabaabcbabccb

Once we have done this, we can use Markov models to calculate the probability of any pattern, including ones we have never seen before

If x = **principalskinner**

Σ is
{**a**,**c**,**e**,**i**,**k**,**l**,**n**,**p**,**r**,**s**}

$|x|$ is 16

skin is a substring of x

prin is a prefix of x

ner is a suffix of x

If y = **in**, then $f_x(y) = 2$

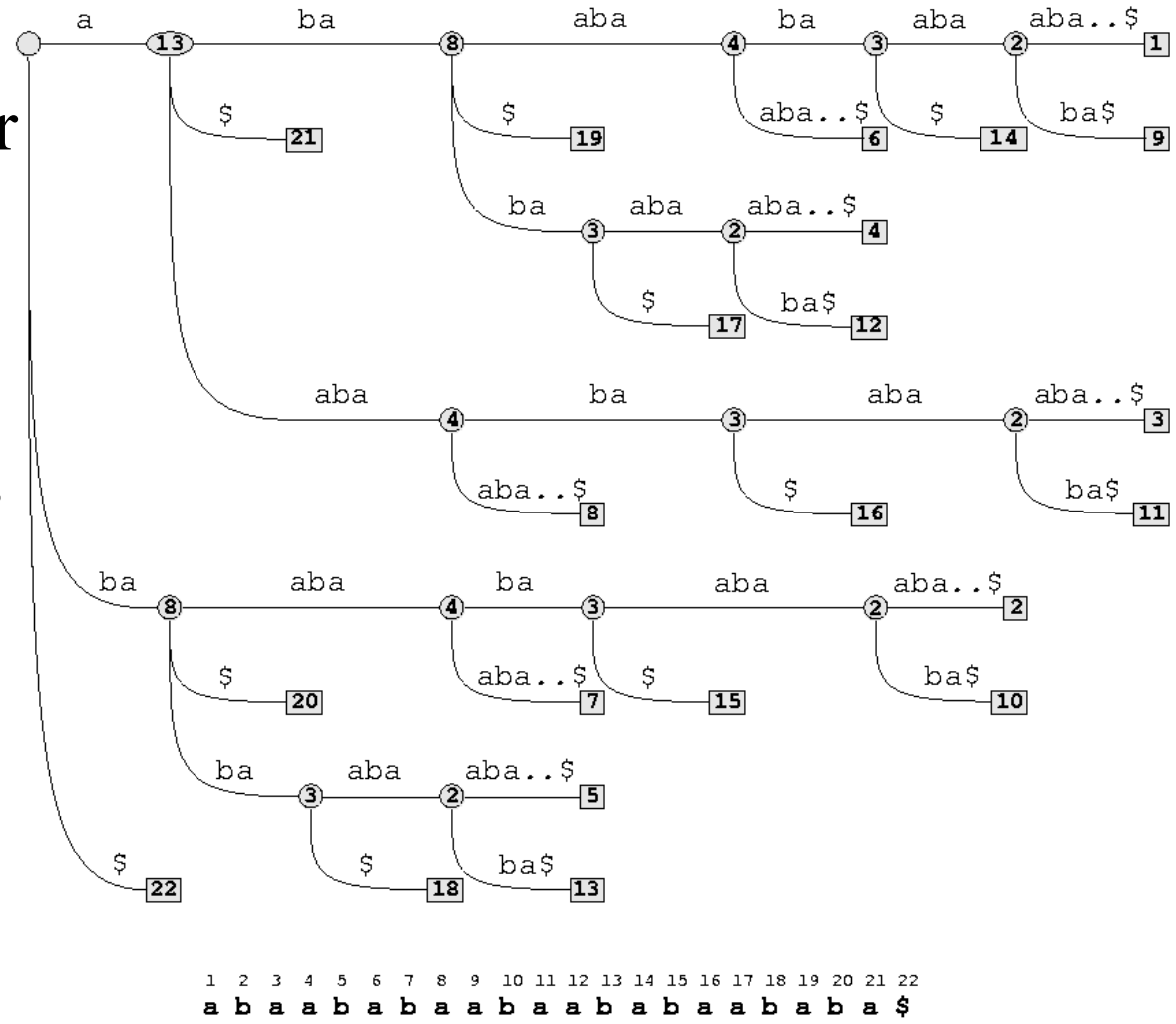
If y = **pal**, then $f_x(y) = 1$

principalskinner

Can we do all this in linear space and time?

Yes! Some very clever modifications of suffix trees (Mostly due to Stefano Lonardi) let us do this in linear space.

An individual pattern can be tested in constant time!



Experimental Evaluation

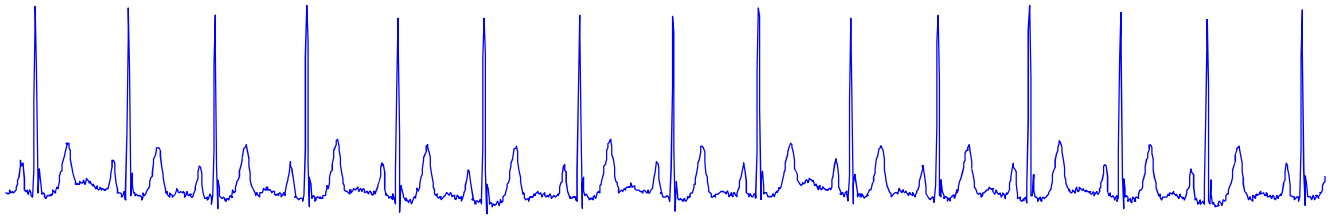
We would like to demonstrate two features of our proposed approach

• **Sensitivity (High True Positive Rate)**
The algorithm can find truly surprising patterns in a time series.

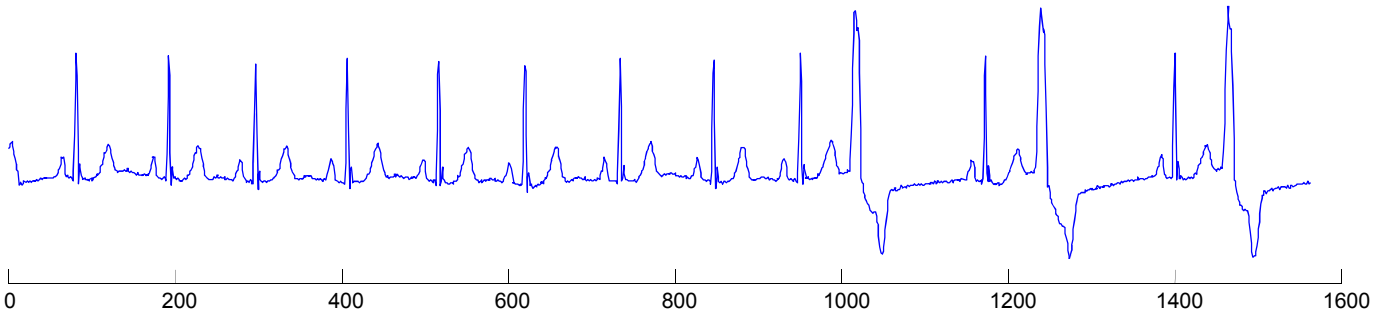
• **Selectivity (Low False Positive Rate)**
The algorithm will not find spurious "surprising" patterns in a time series

Experiment 1: Shock ECG

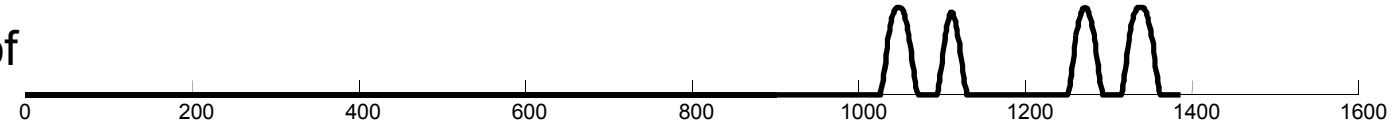
Training data



Test data
(subset)

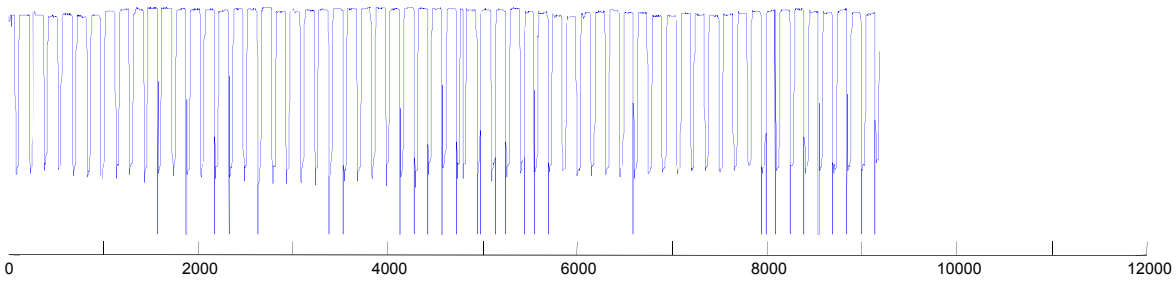


Tarzan's level of
surprise

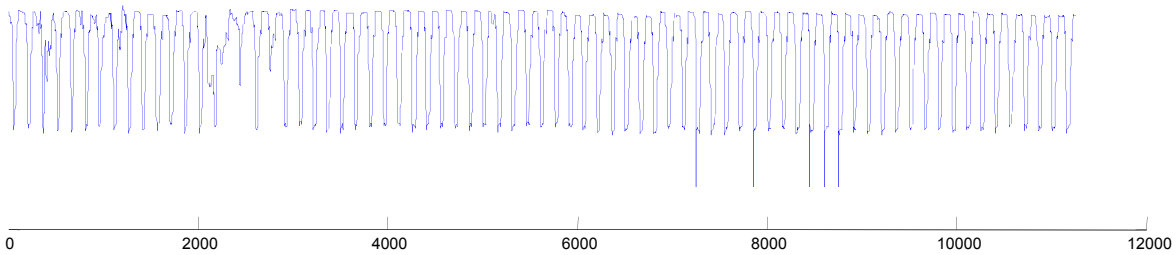


Experiment 2: Video (Part 1)

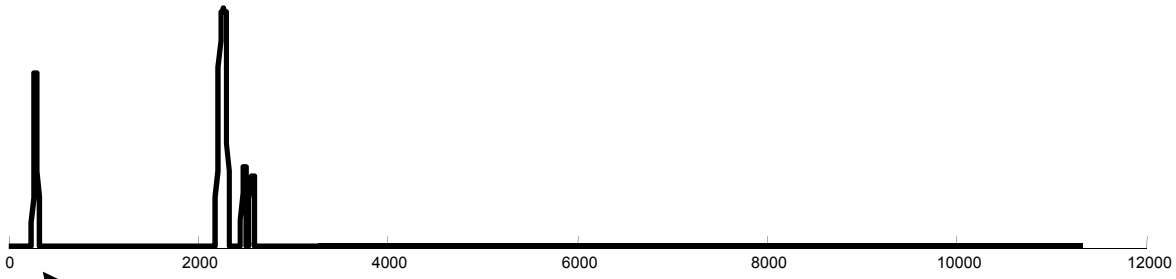
Training data



Test data
(subset)

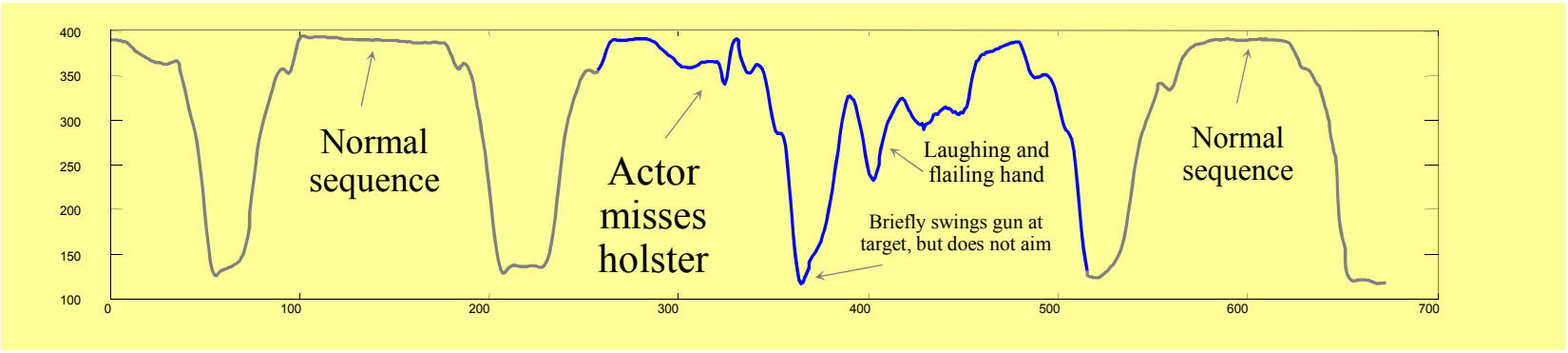


Tarzan's level of surprise



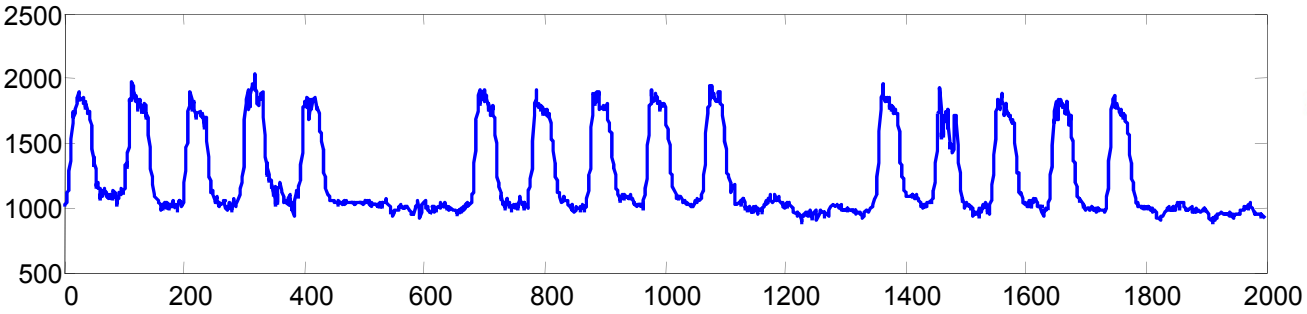
We zoom in on this section in the next slide

Experiment 2: Video (Part 2)



Experiment 3: Power Demand (Part 1)

We consider a dataset that contains the **power demand** for a Dutch research facility for the entire year of 1997. The data is sampled over 15 minute averages, and thus contains 35,040 points.



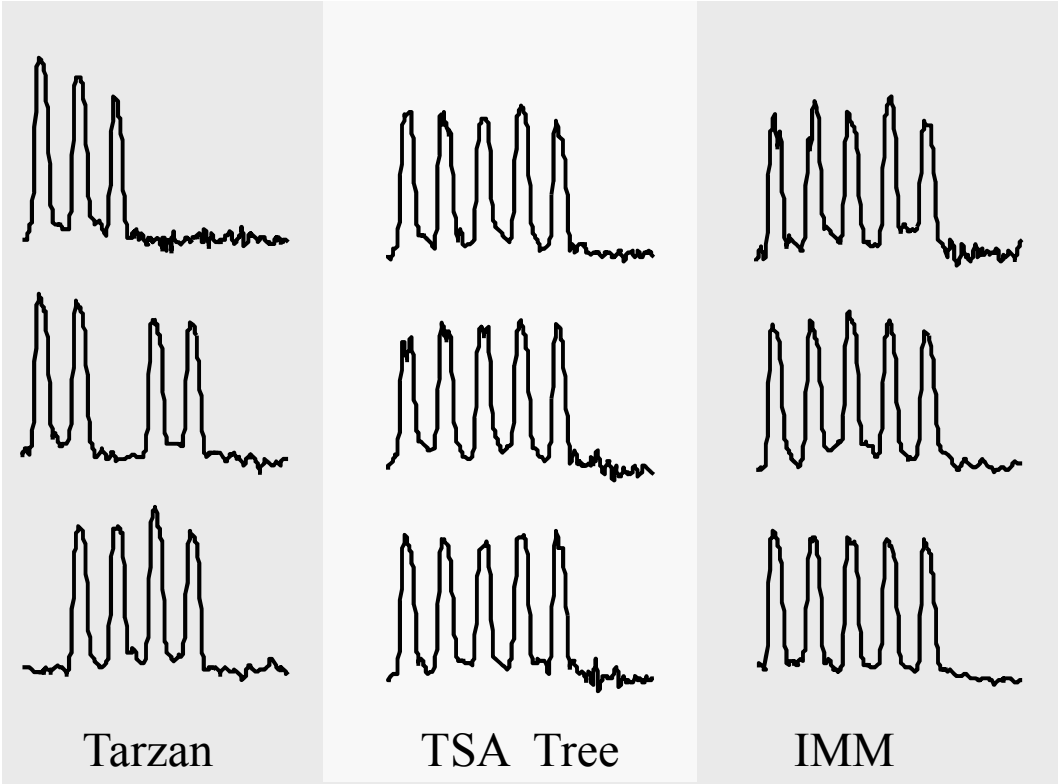
The first 3 weeks of the power demand dataset. Note the repeating pattern of a strong peak for each of the five weekdays, followed by relatively quite weekends

Experiment 3: Power Demand (Part 2)

We used from Monday January 6th to Sunday March 23rd as reference data. This time period is devoid of national holidays. We tested on the remainder of the year.

We will just show the 3 most surprising subsequences found by each algorithm. For each of the 3 approaches we show the entire week (beginning Monday) in which the 3 largest values of surprise fell.

Both TSA-tree and IMM returned sequences that appear to be normal workweeks, however Tarzan returned 3 sequences that correspond to the weeks that contain national holidays in the Netherlands. In particular, from top to bottom, the week spanning both December 25th and 26th and the weeks containing Wednesday April 30th (Koninginnedag, "Queen's Day") and May 19th (Whit Monday).



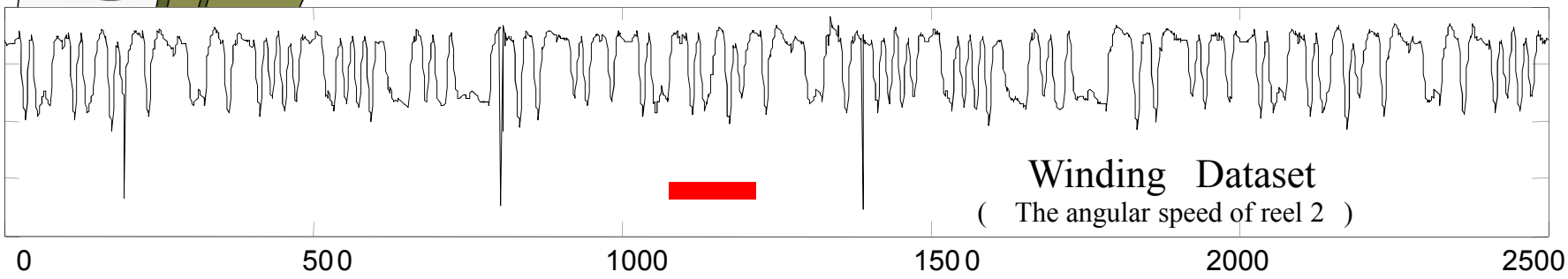
NASA recently said "*TARZAN holds great promise for the future**".

There is now a journal version of TARZAN (under review), if you would like a copy, just ask.

In the meantime, let us consider motif discovery…

* Isaac, D. and Christopher Lynnes, 2003. Automated Data Quality Assessment in the Intelligent Archive, White Paper prepared for the Intelligent Data Understanding program.

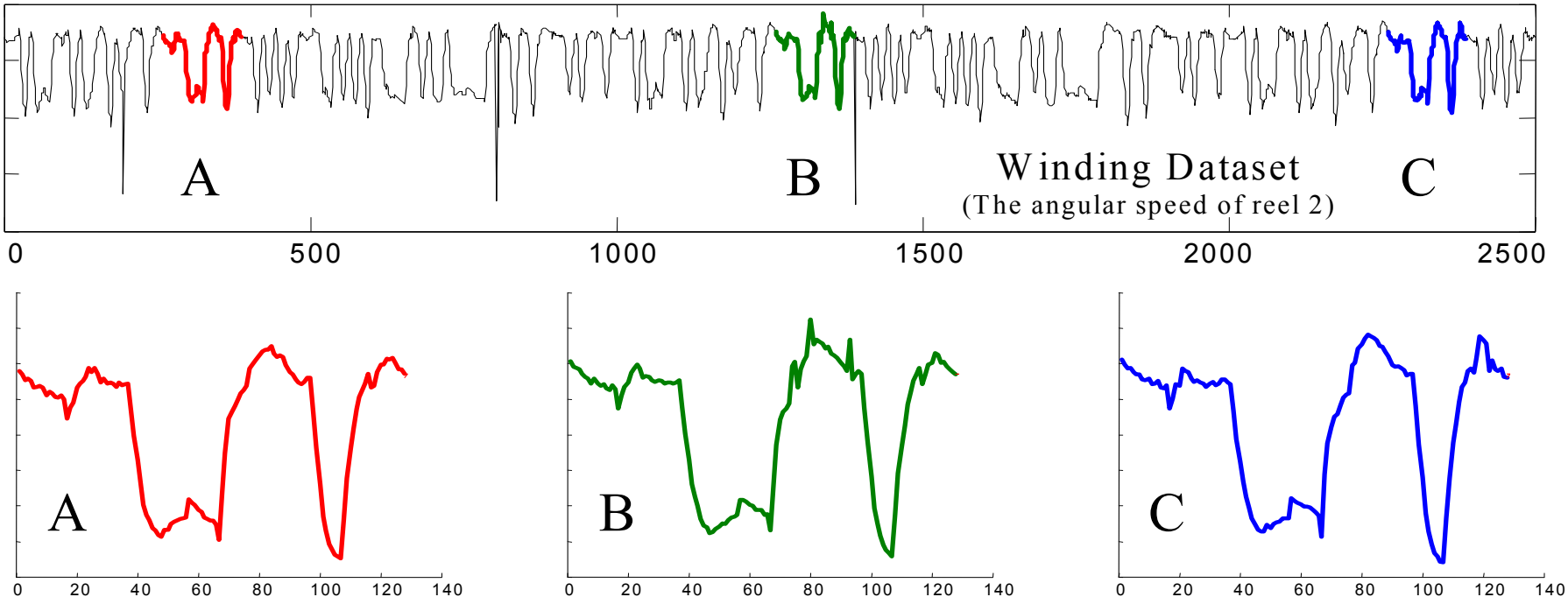
SAX allows Motif Discovery!



Informally, motifs are reoccurring patterns…

Motif Discovery

To find these 3 motifs would require about 6,250,000 calls to the Euclidean distance function.



Why Find Motifs?

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

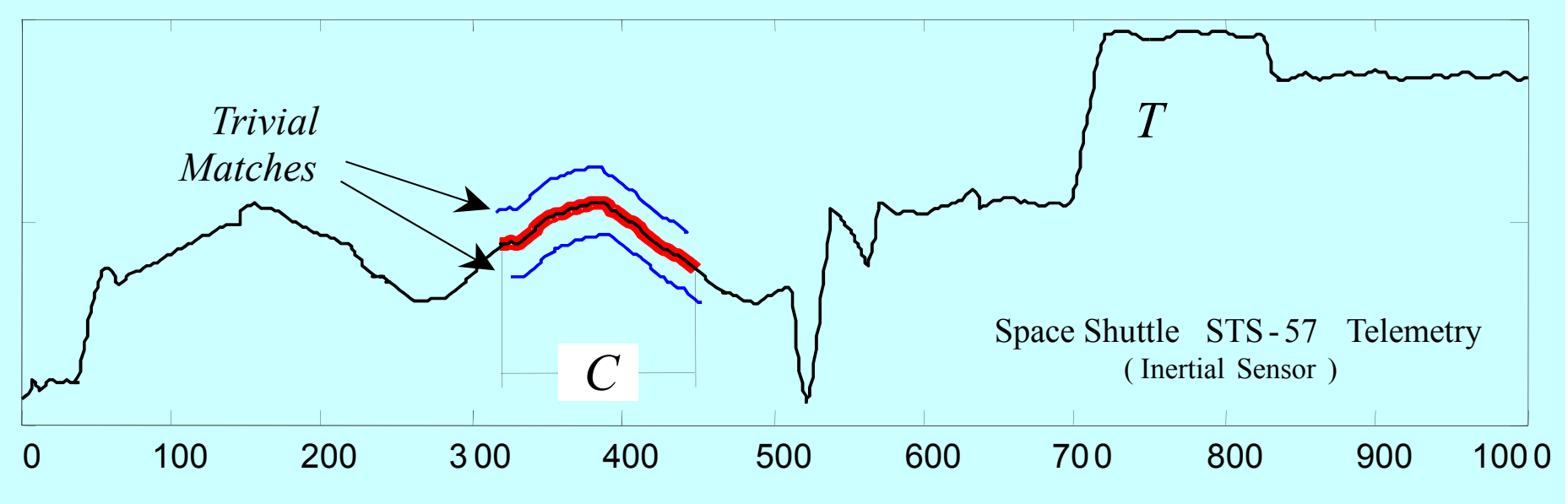
· Several time series **classification algorithms** work by constructing typical prototypes of each class. These prototypes may be considered motifs.

· 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)



Definition 1. *Match*: Given a positive real number R (called *range*) and a time series T containing a subsequence C beginning at position p and a subsequence M beginning at q, if $D(C, M) \leq R$, then M is called a *matching* subsequence of C.

Definition 2. *Trivial Match*: Given a time series T, containing a subsequence C beginning at position p and a matching subsequence M beginning at q, we say that M is a *trivial match* to C if either $p = q$ or there does not exist a subsequence M' beginning at q' such that $D(C, M') > R$, and either $q < q' < p$ or $p < q' < q$.

Definition 3. *K-Motif*(n,R): Given a time series T, a subsequence length n and a range R, the most significant motif in T (hereafter called the *1-Motif*(n,R)) is the subsequence C_1 that has highest count of non-trivial matches (ties are broken by choosing the motif whose matches have the lower variance). The K^{th} most significant motif in T (hereafter called the *K-Motif*(n,R)) is the subsequence C_K that has the highest count of non-trivial matches, and satisfies $D(C_K, C_i) > 2R$, for all $1 \leq i < K$.

OK, we can define motifs, but how do we find them?

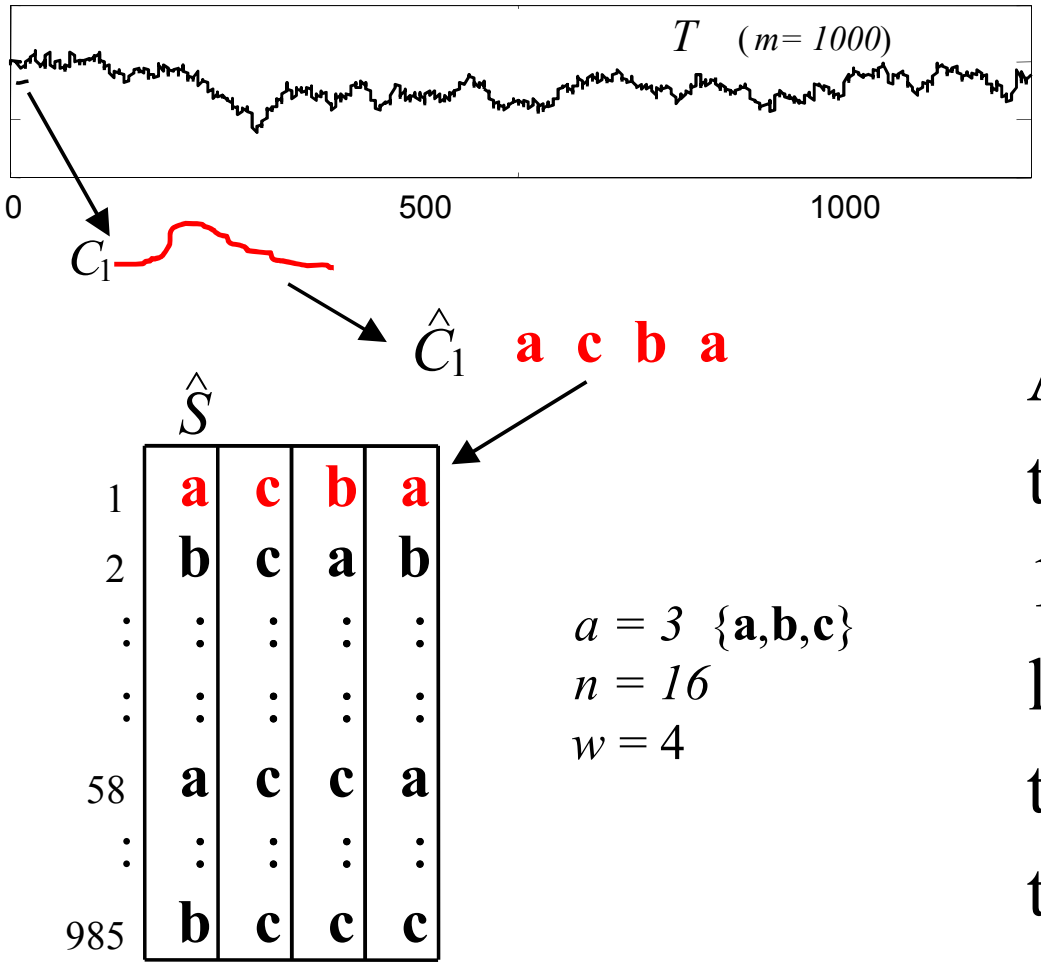
The obvious brute force search algorithm is just too slow…

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

*** J Buhler and M Tompa. *Finding motifs using random projections*. In RECOMB'01. 2001.**

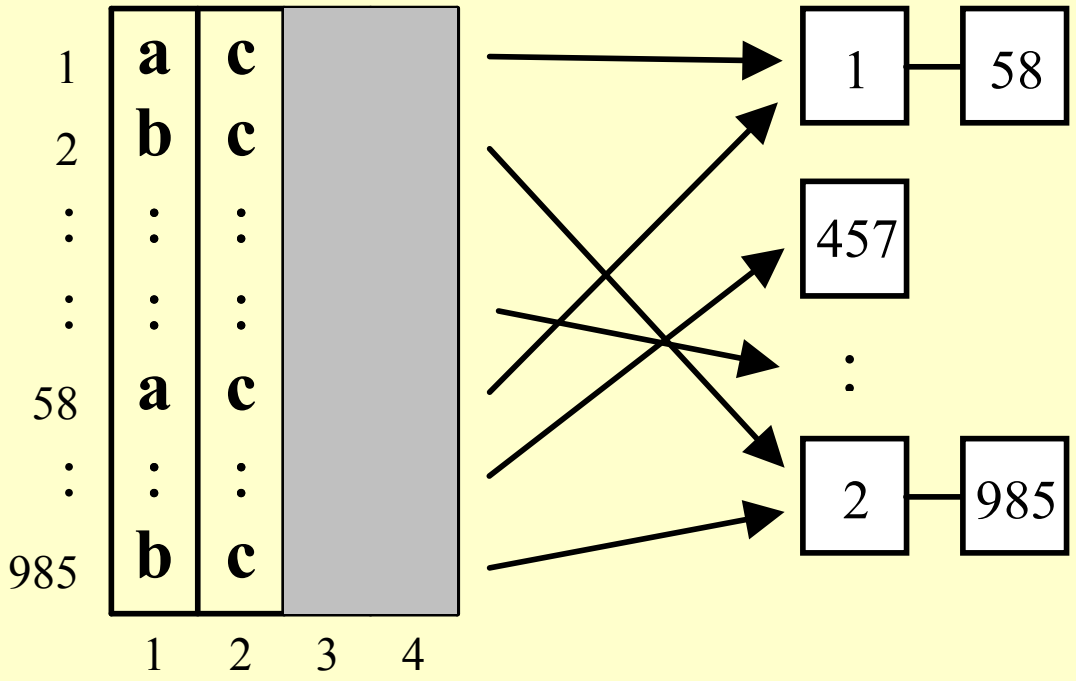
A simple worked example of our motif discovery algorithm

The next 4 slides

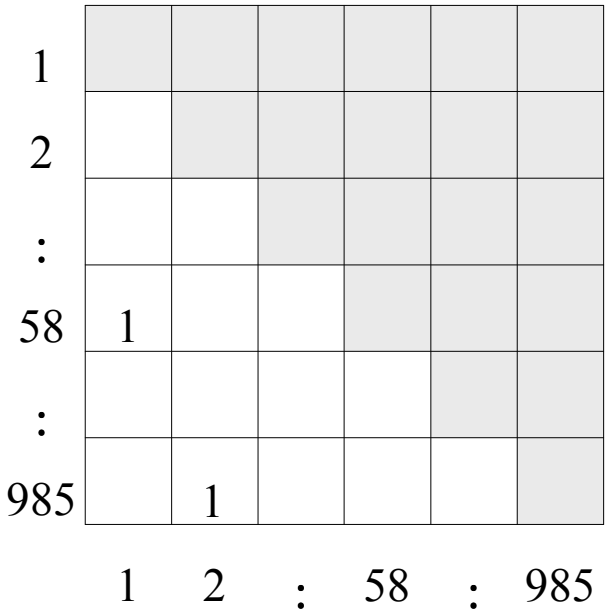


Assume that we have a time series T of length 1,000, and a motif of length 16, which occurs twice, at time T_1 and time T_{58}.

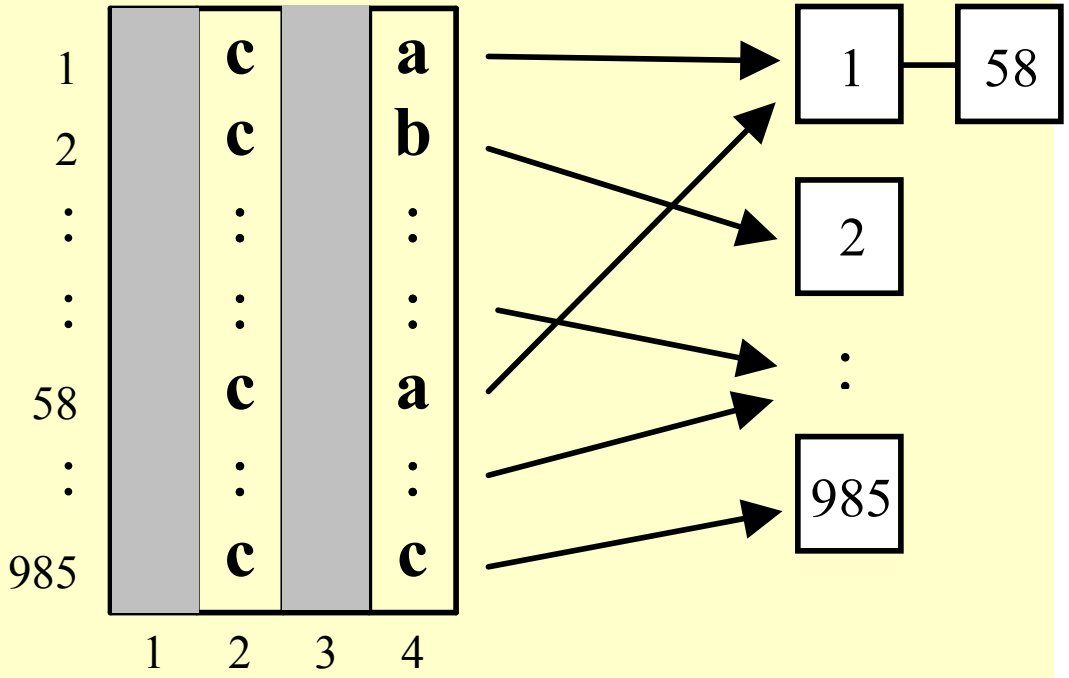
A mask {1,2} was randomly chosen, so the values in columns {1,2} were used to project matrix into buckets.



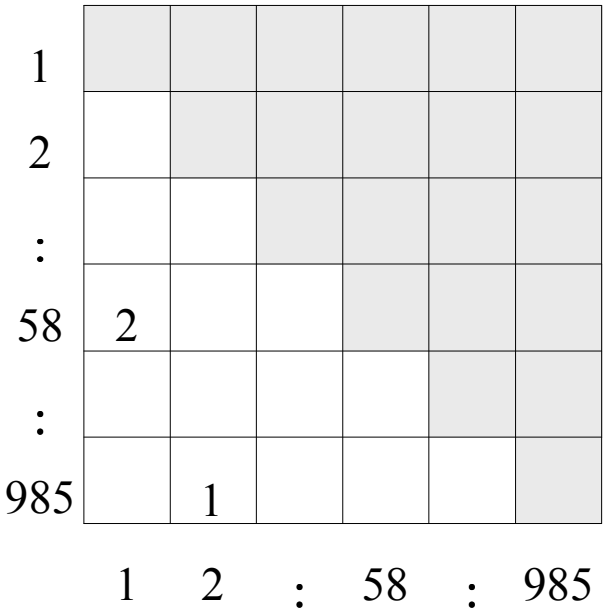
Collisions are recorded by incrementing the appropriate location in the collision matrix



A mask {2,4} was randomly chosen, so the values in columns {2,4} were used to project matrix into buckets.



Once again, collisions are recorded by incrementing the appropriate location in the collision matrix



We can calculate the expected values in the matrix, assuming there are NO patterns…

$$\mathrm{E}(k,a,w,d,t) = \binom{k}{2} \sum_{i=0}^{d} \left(1 - \frac{i}{w}\right)^{t} \binom{w}{i} \left(\frac{a-1}{a}\right)^{i} \left(\frac{1}{a}\right)^{w-i}$$

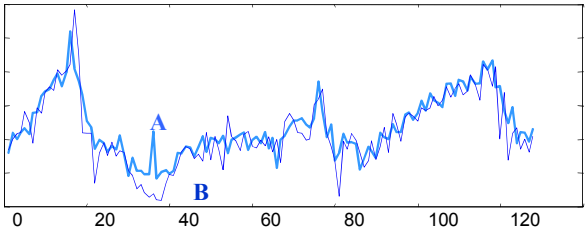
Suppose

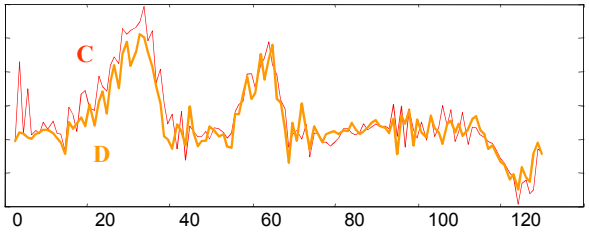
$\mathrm{E}(k,a,w,d,t) = 2$

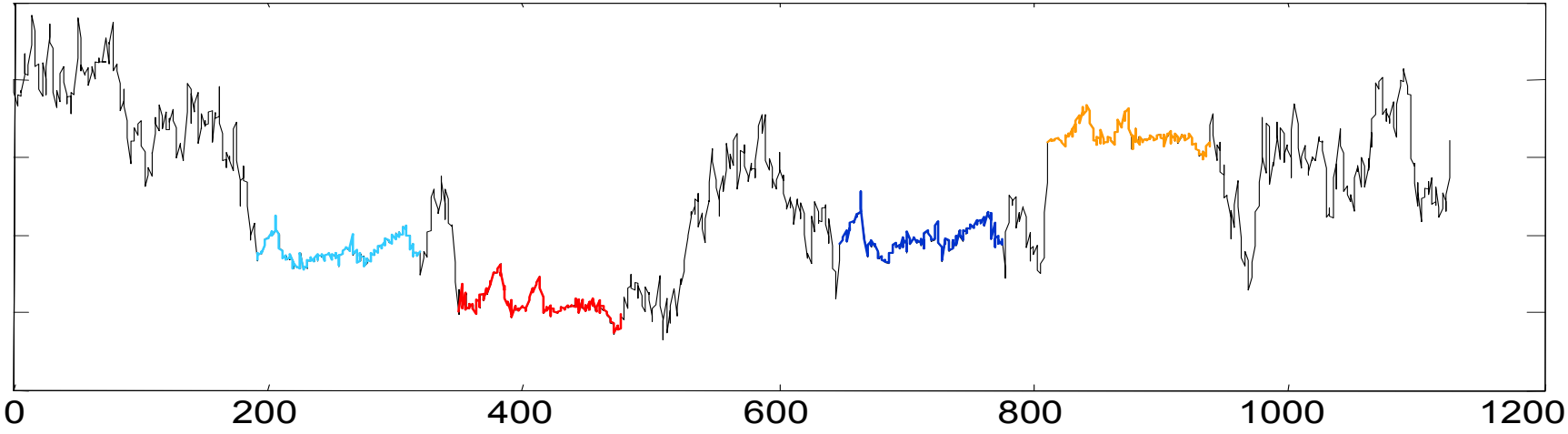
1						
2	2					
:	1	3				
58	**27**	2	1			
:	3	2	2	1		
985	0	1	2	1	3	
	1	2	:	58	:	985

A Simple Experiment

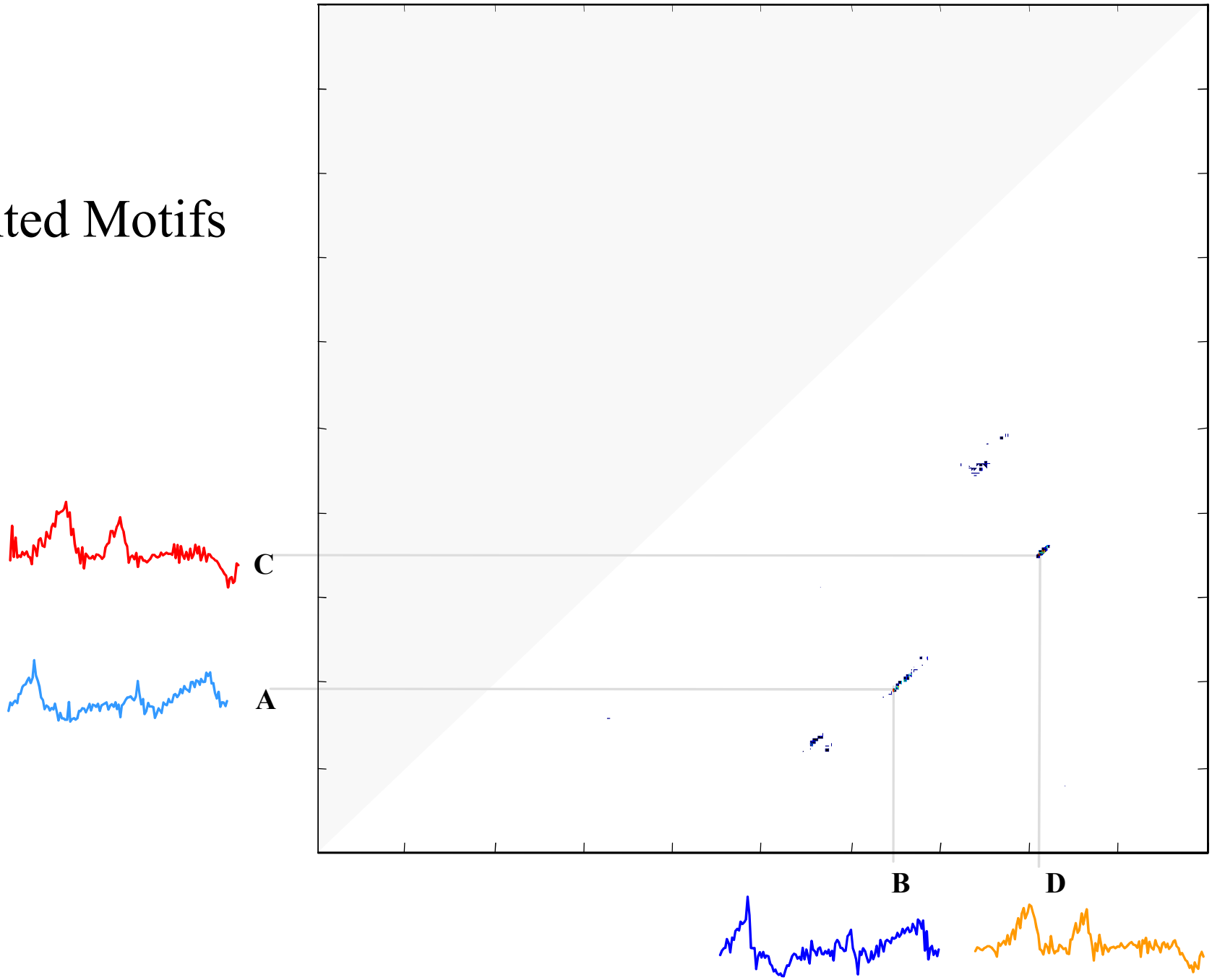
Lets imbed two motifs into a random walk time series, and see if we can recover them



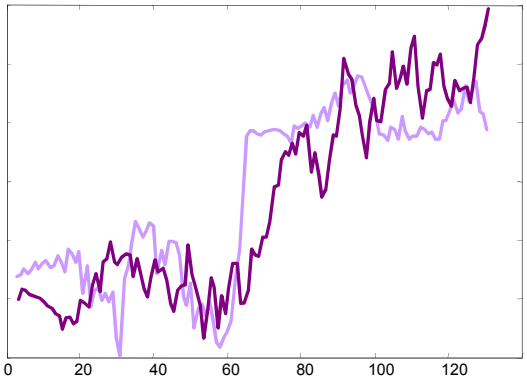


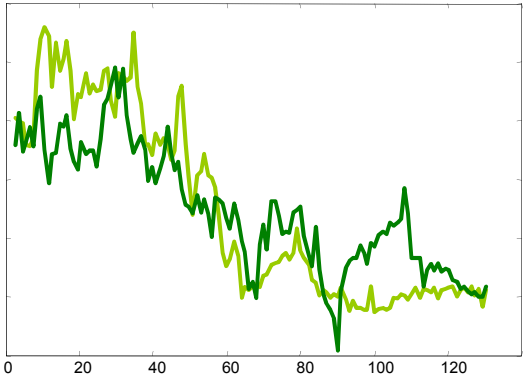


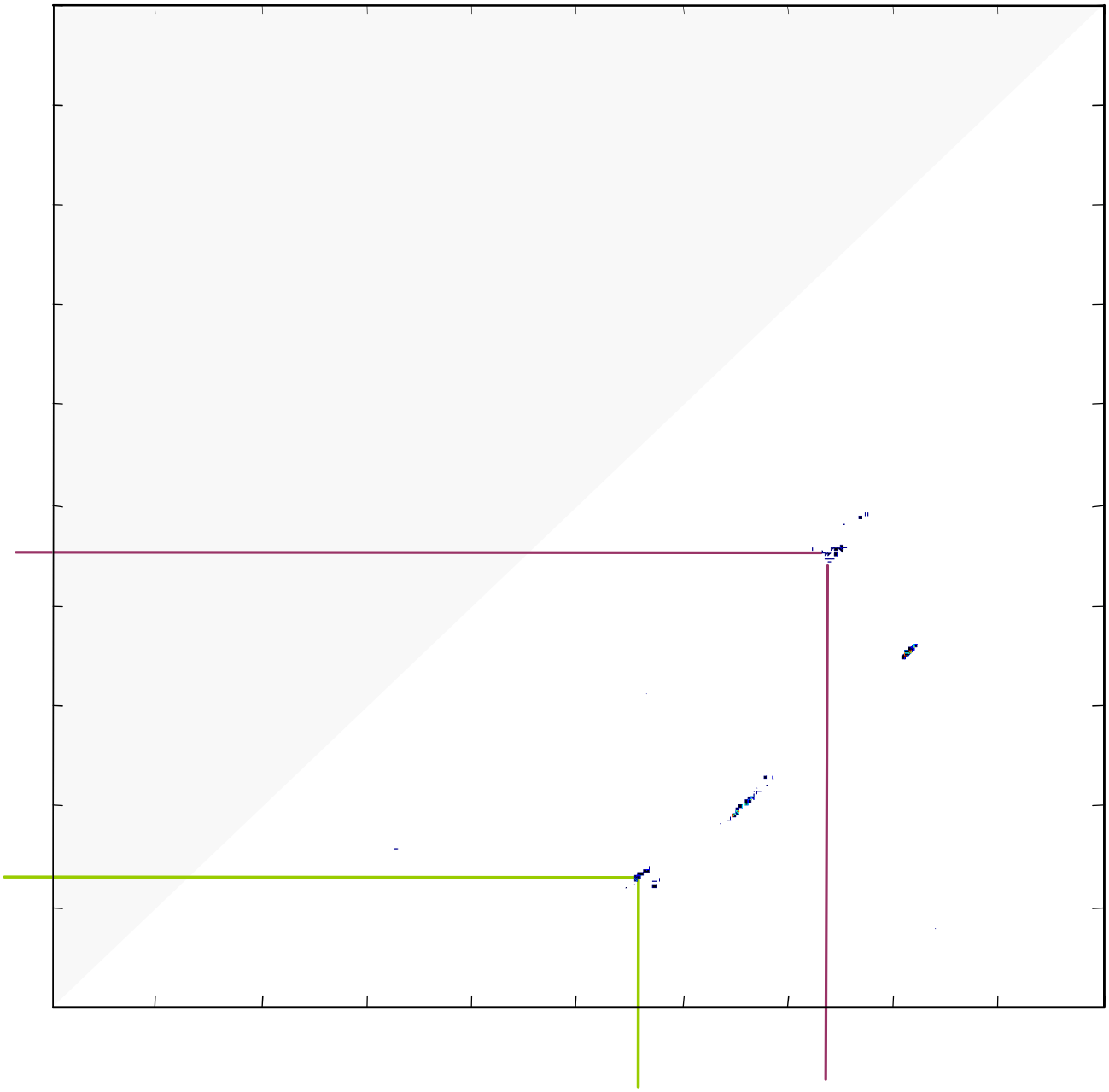
Planted Motifs



“Real” Motifs

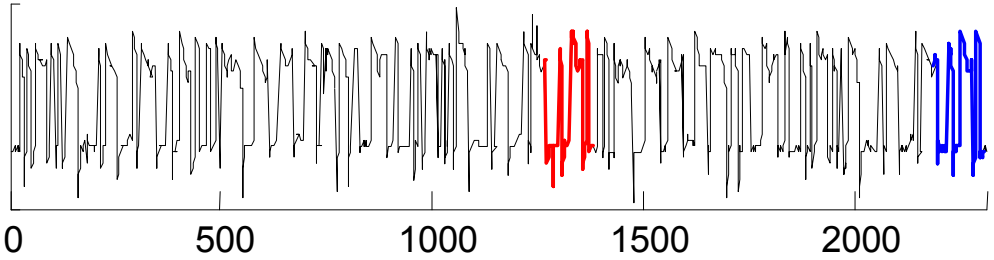


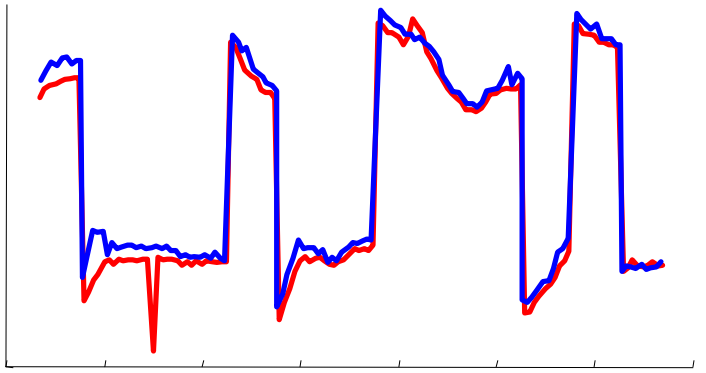




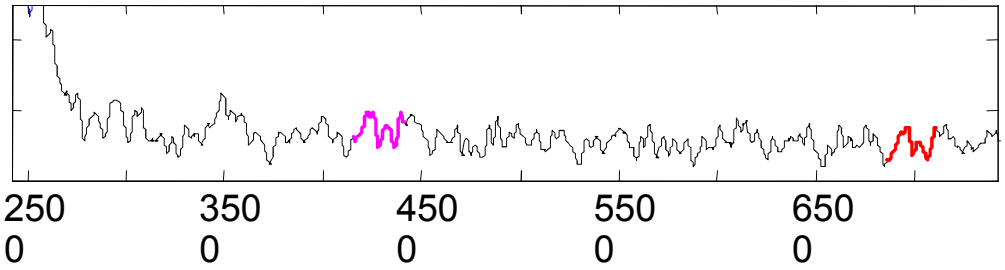
Some Examples of Real Motifs

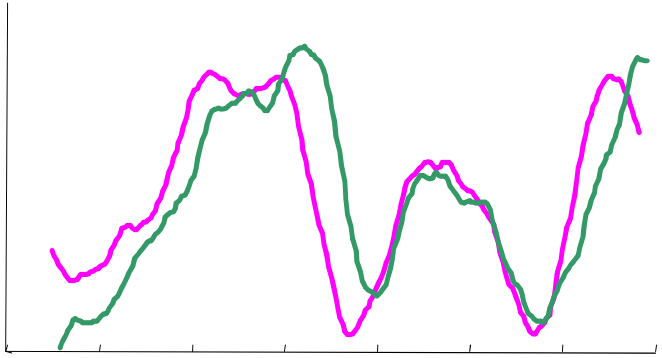
Motor 1 (DC Current)



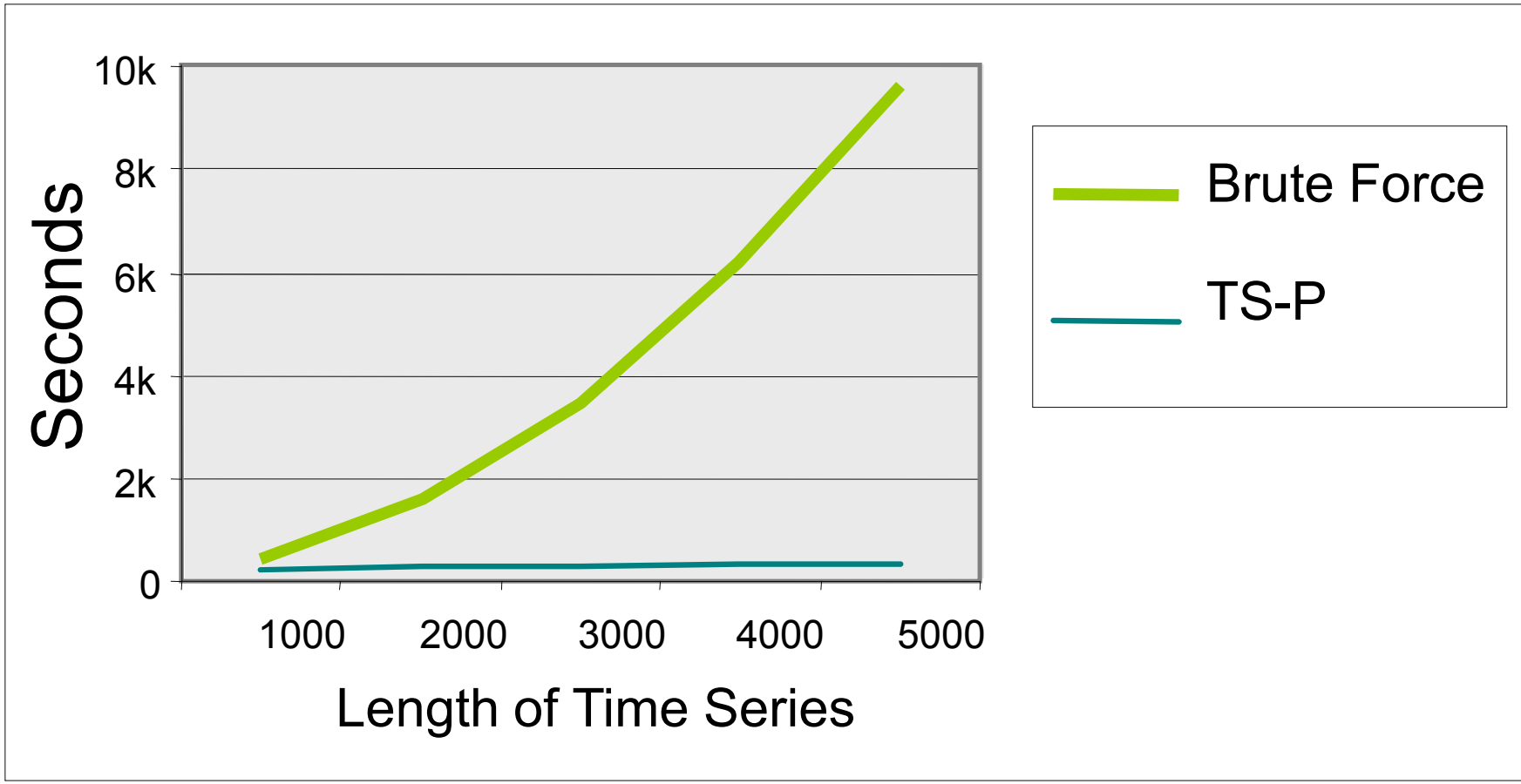


Astrophysics (Photon Count)





How Fast can we find Motifs?



Let us consider the utility of SAX for visualizing time series. We start with an apparent digression, visualizing DNA….

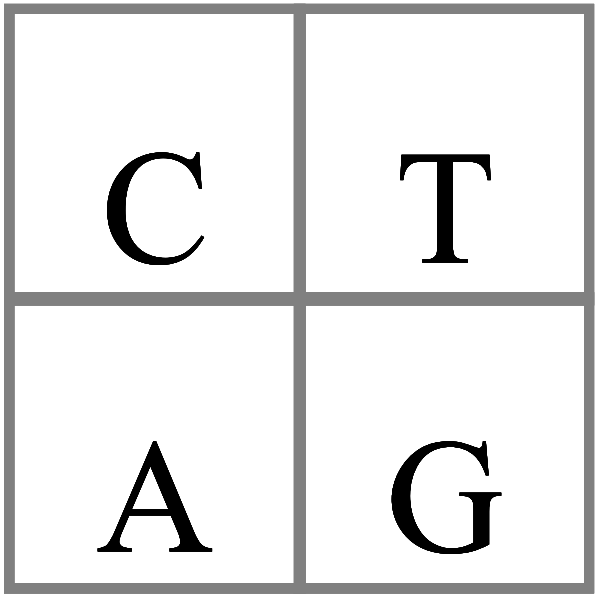
The DNA of two species…

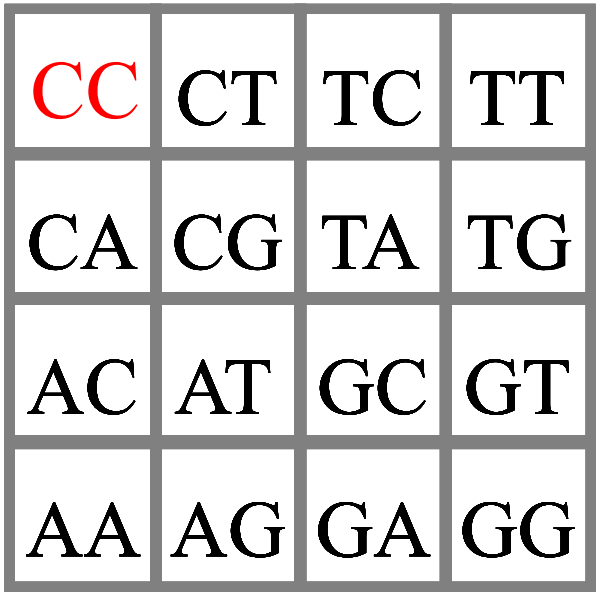
Are they similar?

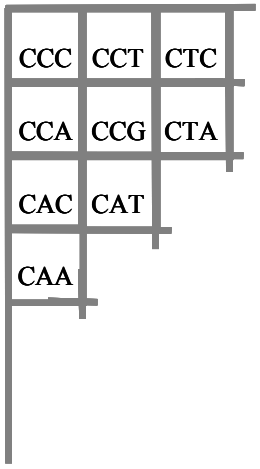
C	T
A	G

0.20	0.24
0.26	0.30

CCGTGCTAGGGCCACCTACCTTGGTCC
CCGCAAGCTCATCTGCGCGAACCAGAA
GCCACCACCTTGGGTTGAAATTAAGGA
GCGGTTGGCAGCTTCCAGGCGCACGTA
CTGCGAATAAATAACTGTCCGCACAAG
AGCCGACGATAAAGAAGAGAGTCGACC
CTCTAGTCACGACCTACACACAGAACC
GTGCTAGACGCCATGAGATAAGCTAAC



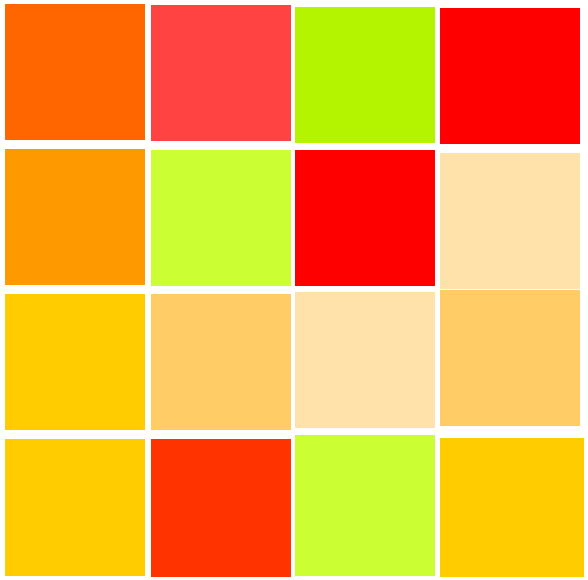




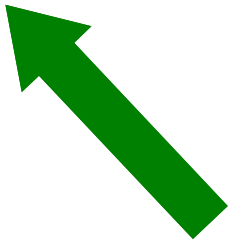
CCGTGCTAGGGCCACCTACCTTGGTCC
CCGCAAGCTCATCTGCGCGAACCAGA
GCCACCACCTTGGGTTGAAATTAAGGA
GCGGTTGGCAGCTTCCAGGCGCACGT
CTGCGAATAAATAACTGTCCGCACAAG
AGCCGACGATAAAGAAGAGAGTCGAC
CTCTAGTCACGACCTACACACAGAACC
GTGCTAGACGCCATGAGATAAGCTAAC

0.02	0.04	0.09	0.04
	0.03	0.07	0.02
		0.11	0.03

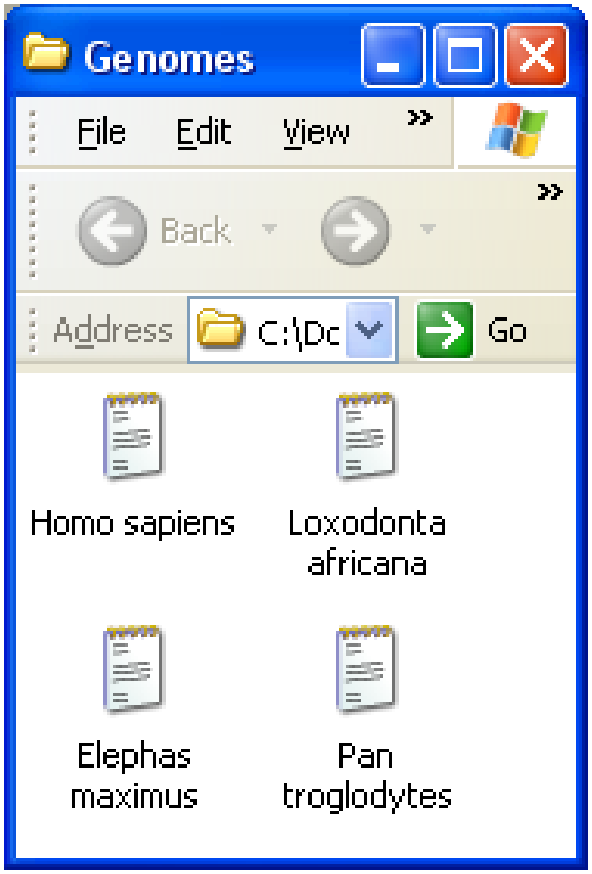
CCGTGCTAGGCCCCACCCCTACCTTGC
GTCCCCGCAAGCTCATCTGCGCGAACC
GAACGCCCACCACCCTTGGGTTGAAAT
AAGGAGGCGGTTGGCAGCTTCCCAGG
CACGTACCTGCGAATAAATAACTGTCC
ACAAGGAGCCCGACGATAGTCGACCC
TCTAGTCACGACCTACACACAGAACCT
TGCTAGACGCCATGAGATAAGCTAACA

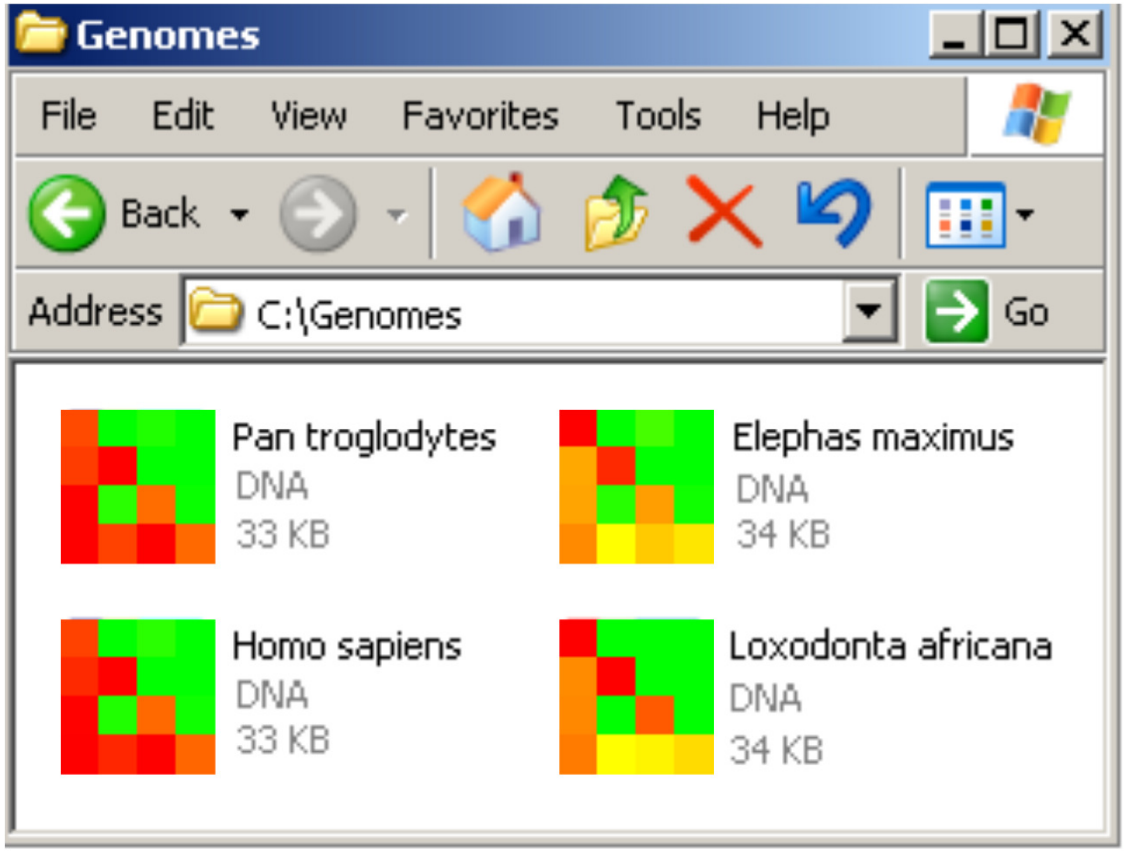


OK. Given any DNA string I can make a colored bitmap, so what?



CCGTGCTAGGCCCCACCCCTACCTTGC
GTCCCCGCAAGCTCATCTGCGCGAACC
GAACGCCCACCACCCTTGGGTTGAAAT
AAGGAGGCGGTTGGCAGCTTCCCAGG
CACGTACCTGCGAATAAATAACTGTCC
ACAAGGAGCCCGACGATAGTCGACCC
TCTAGTCACGACCTACACACAGAACCT
TGCTAGACGCCATGAGATAAGCTAACA



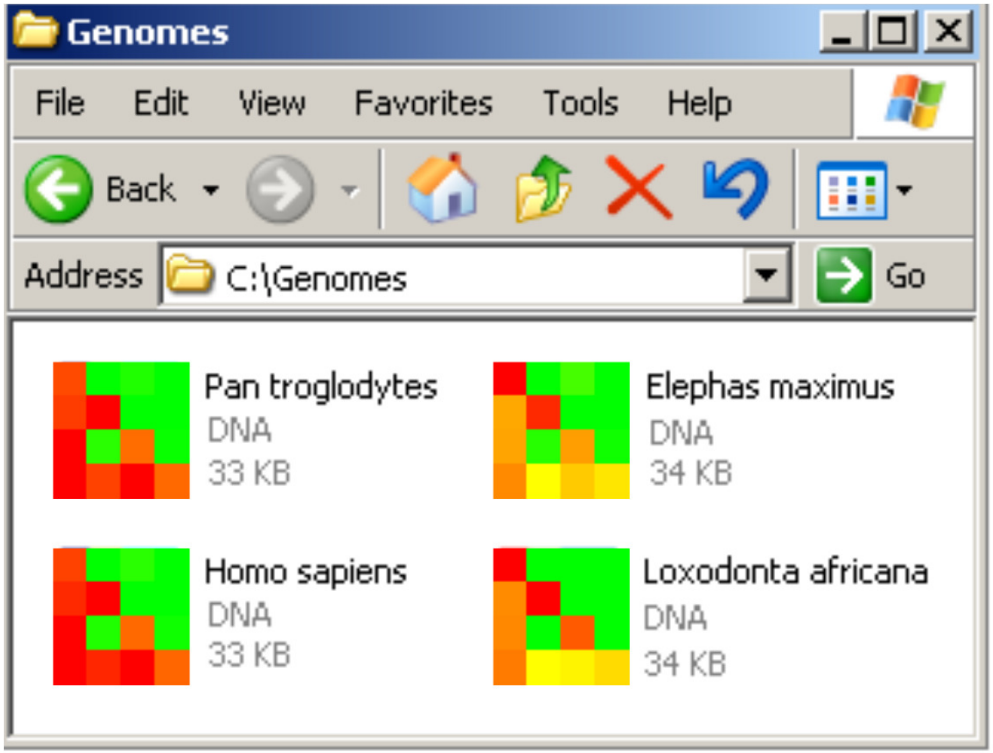


Note *Elephas maximus* is the Indian Elephant, *Loxodonta africana* is the African elephant and *Pan troglodytes* is the chimpanzee.

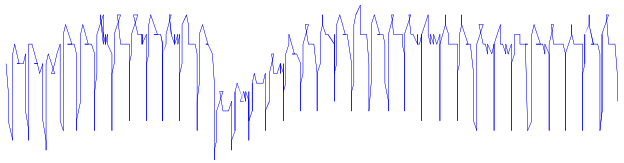
Two Questions

- Can we do something similar for time series?

- Would it be useful?



Can we do make bitmaps for time series?

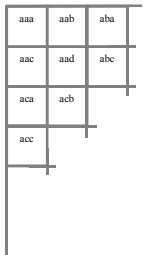


Yes, with SAX!

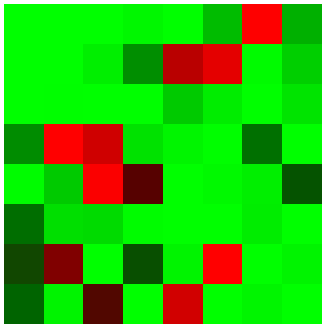
accbabcdbcabdbcadbacbdbdcadbaacb…

a	b
c	d

aa	ab	ba	bb
ac	ad	bc	bd
ca	cb	da	db
cc	cd	dc	dd



Time Series Bitmap



Time Series Bitmaps

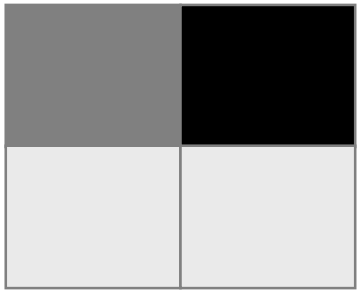
Imagine we have the following SAX strings…

abcdba
bdbadb
cbabca

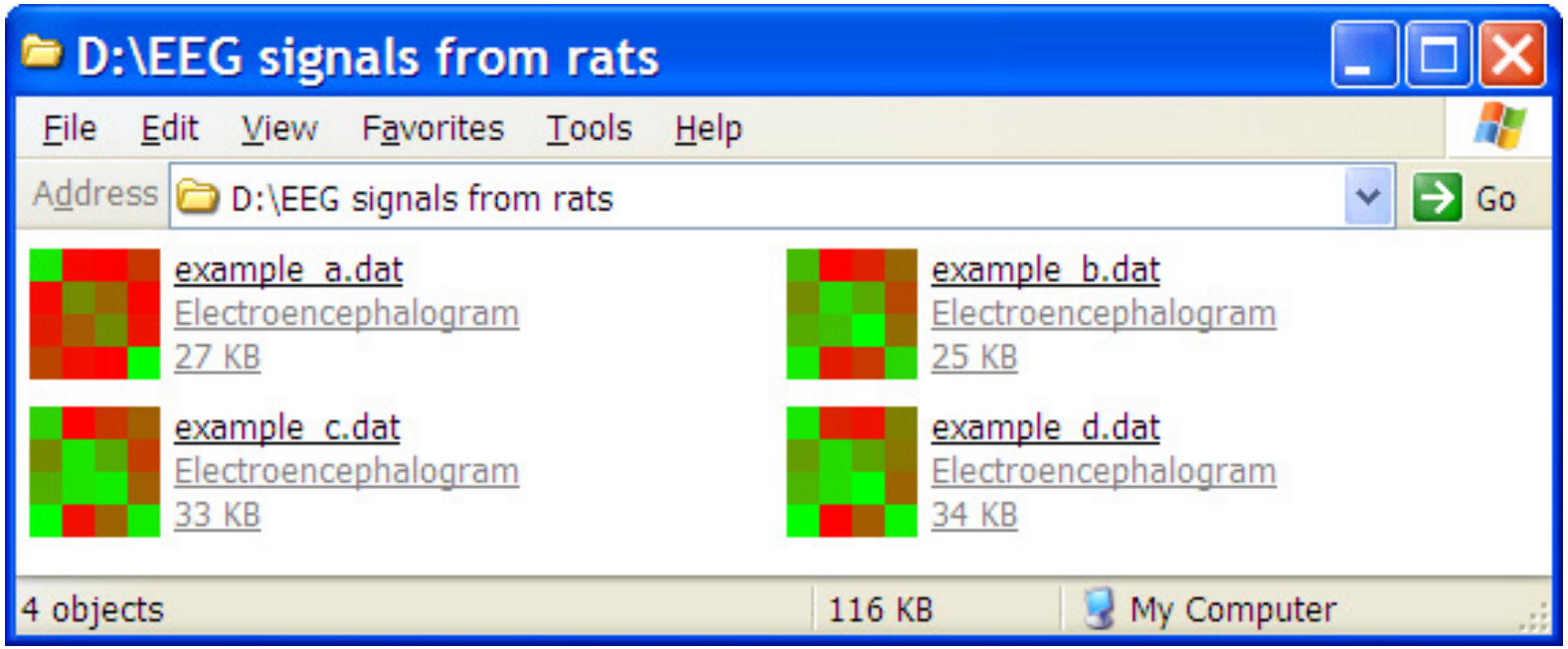
There are 5 “a”
There are 7 “b”
There are 3 “c”
There are 3 “d”

a	b
c	d

5	7
3	3



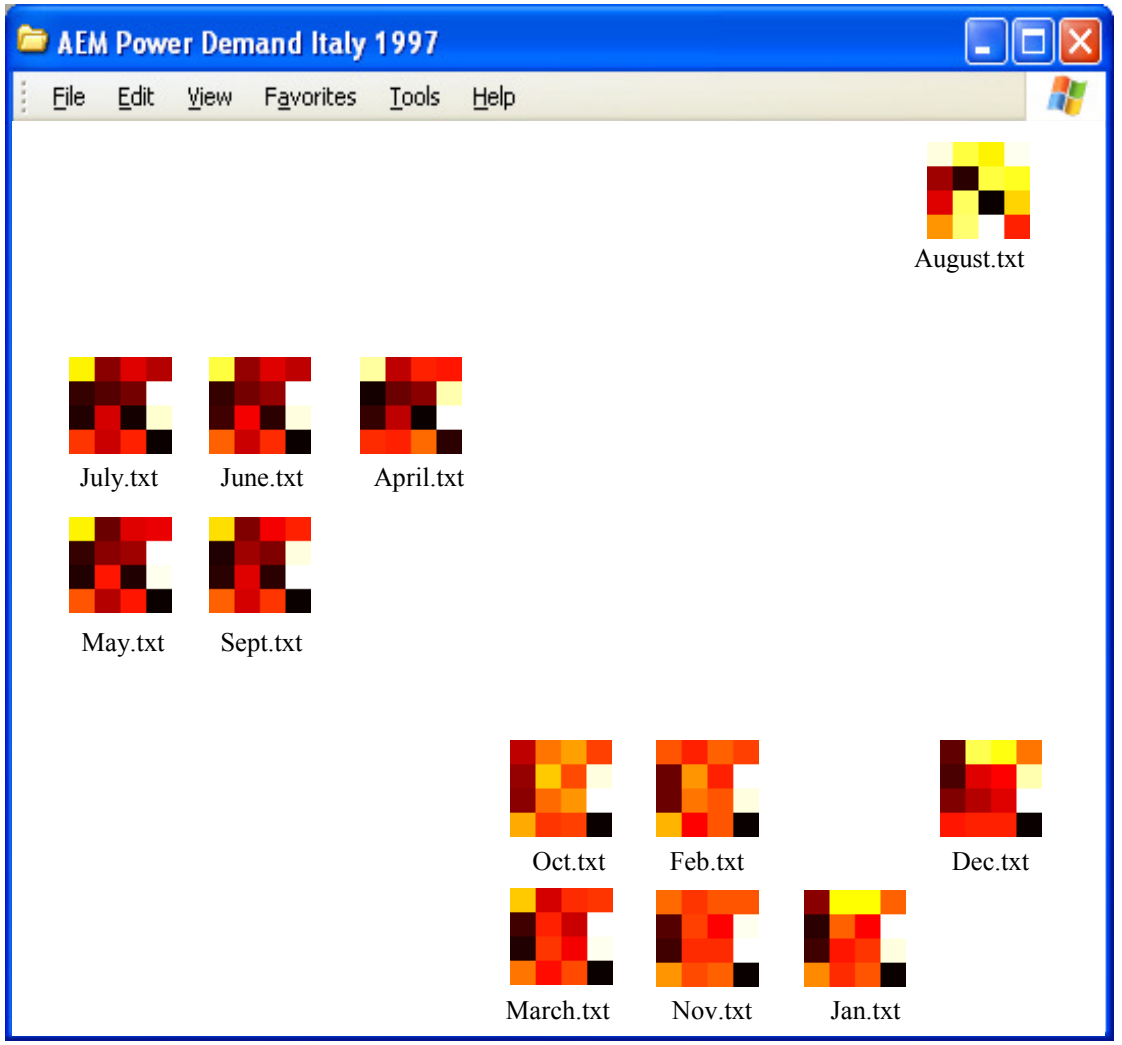
We can paint the pixels based on the frequencies

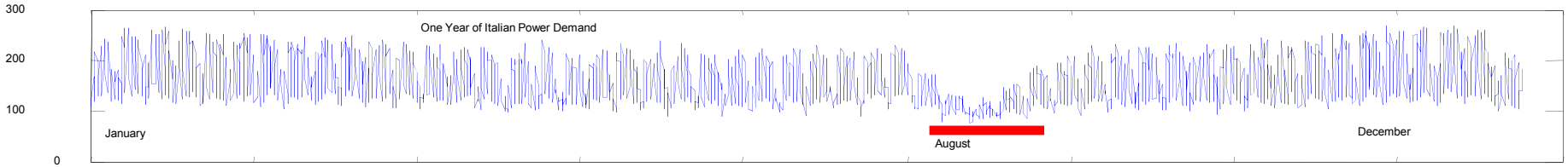


While they are all example of EEGs, *example_a.dat* is from a normal trace, whereas the others contain examples of spike-wave discharges.

We can further enhance the time series bitmaps by arranging the thumbnails by "*cluster*", instead of arranging by *date*, *size*, *name* etc

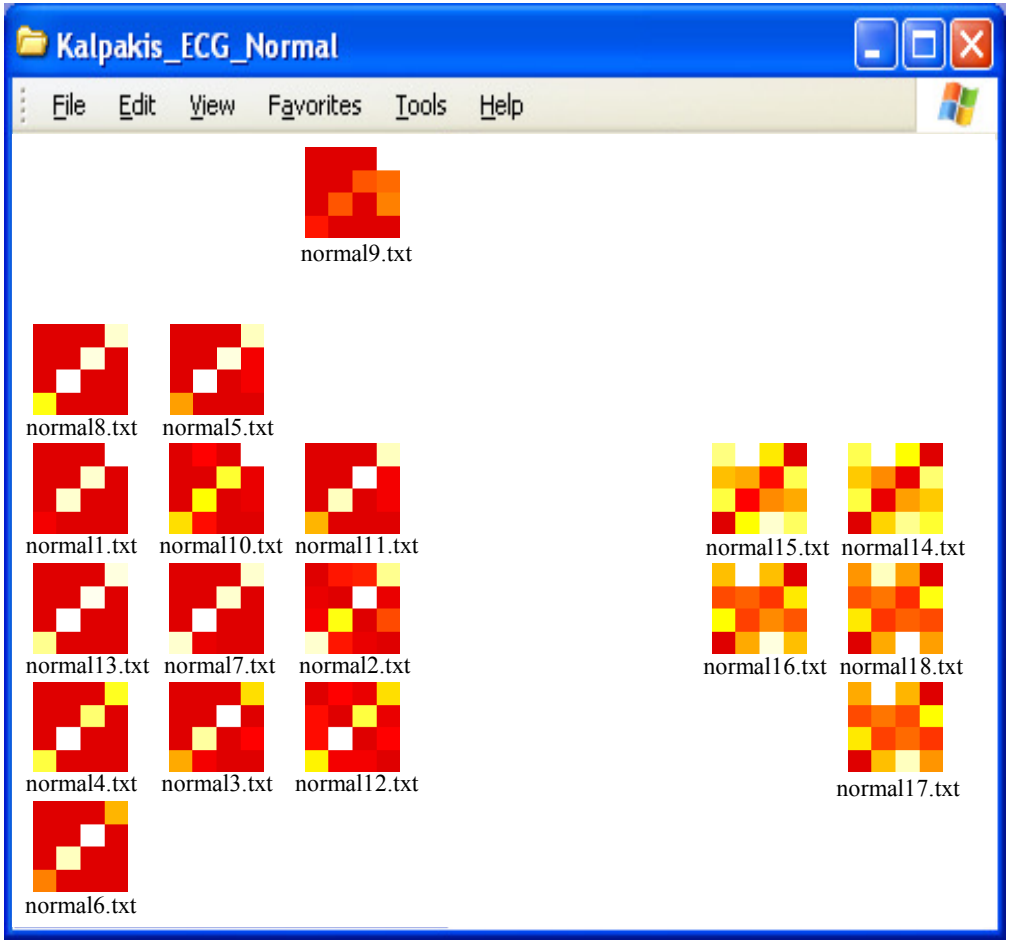
We can achieve this with MDS.

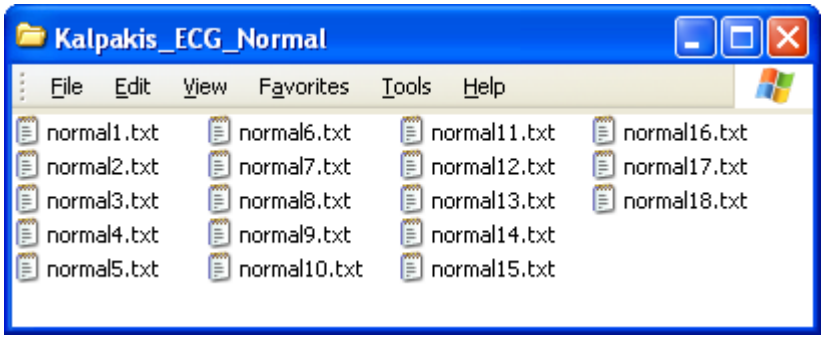


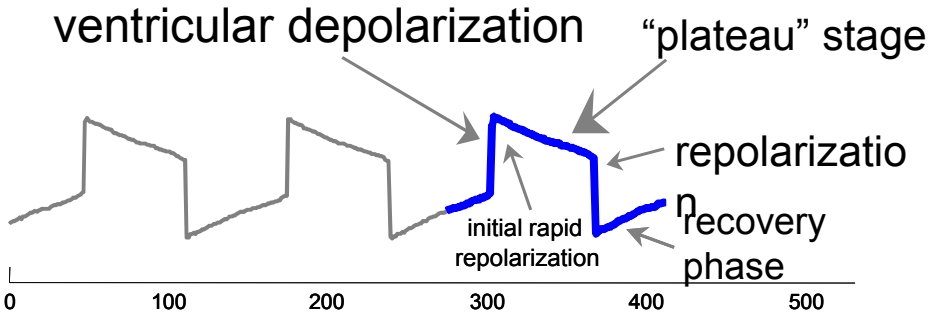


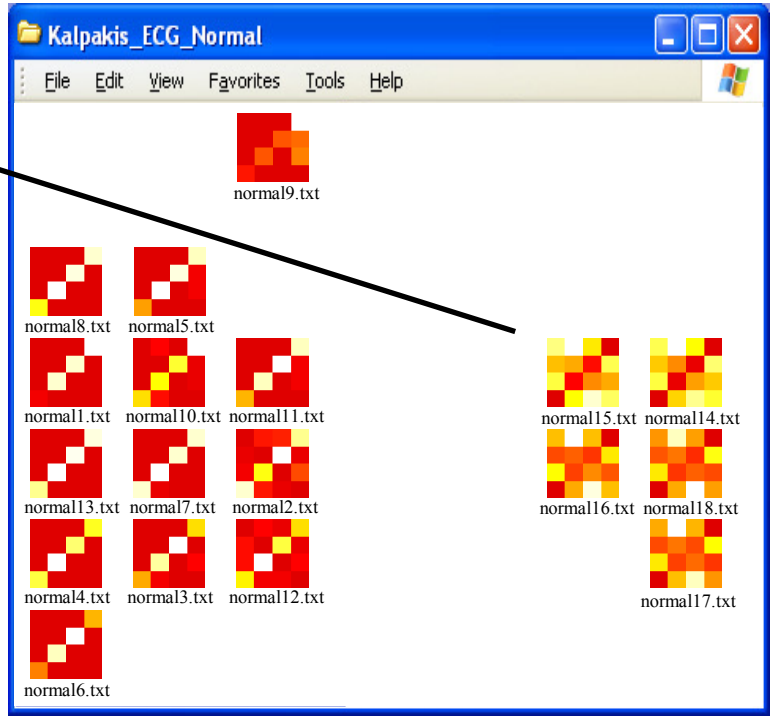
A well known dataset *Kalpakis_ECG*, allegedly contains 70 ECGS

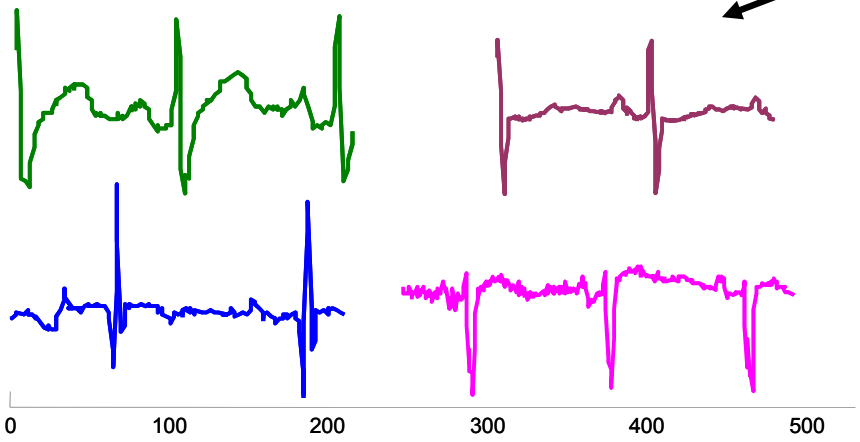
If we view them as time series bitmaps, a handful stand out…





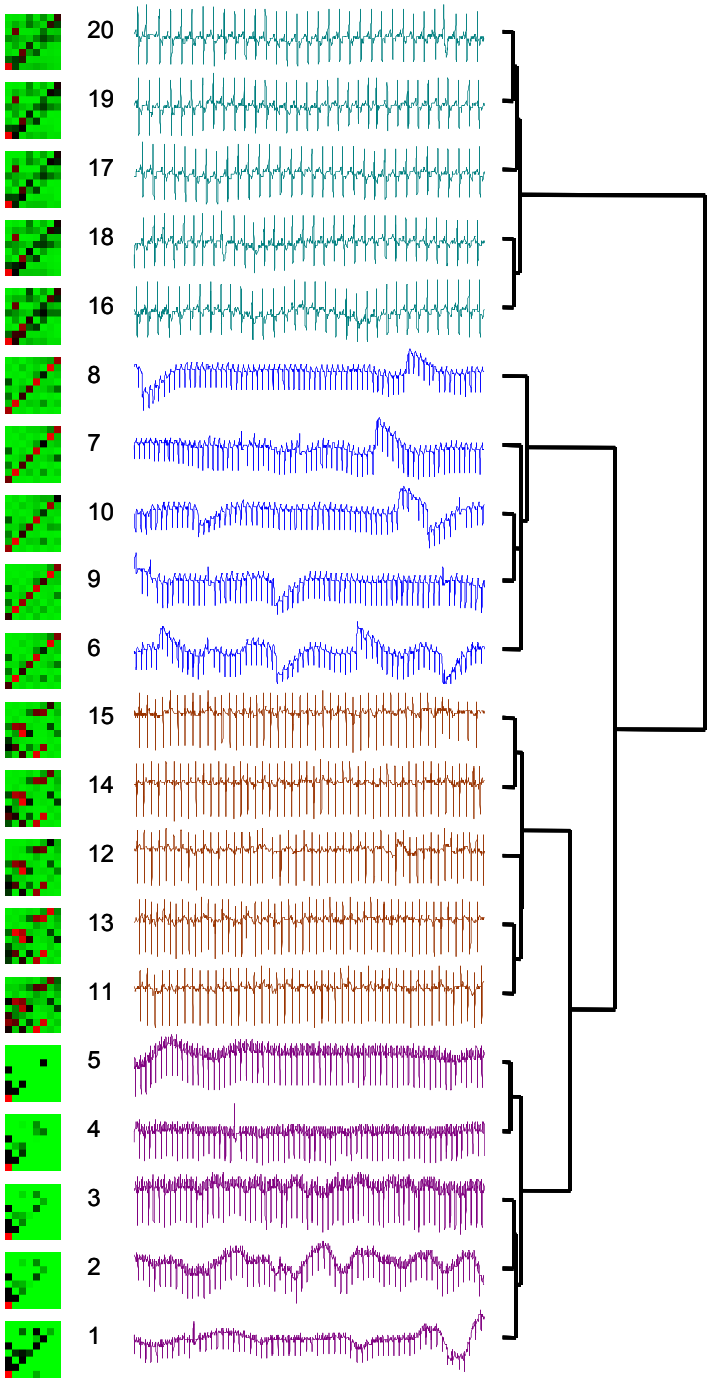






Some of the data are not heartbeats! They are the action potential of a normal pacemaker cell

We can test how much useful information is retained in the bitmaps by using *only* the bitmaps for clustering/classification/anomaly detection



We can test how much useful information is retained in the bitmaps by using *only* the bitmaps for clustering/classification/anomaly detection

Data Key

Cluster 1 (datasets 1 ~ 5):

BIDMC Congestive Heart Failure Database (chfdb): record chf02

Start times at 0, 82, 150, 200, 250, respectively

Cluster 2 (datasets 6 ~ 10):

BIDMC Congestive Heart Failure Database (chfdb): record chf15

Start times at 0, 82, 150, 200, 250, respectively

Cluster 3 (datasets 11 ~ 15):

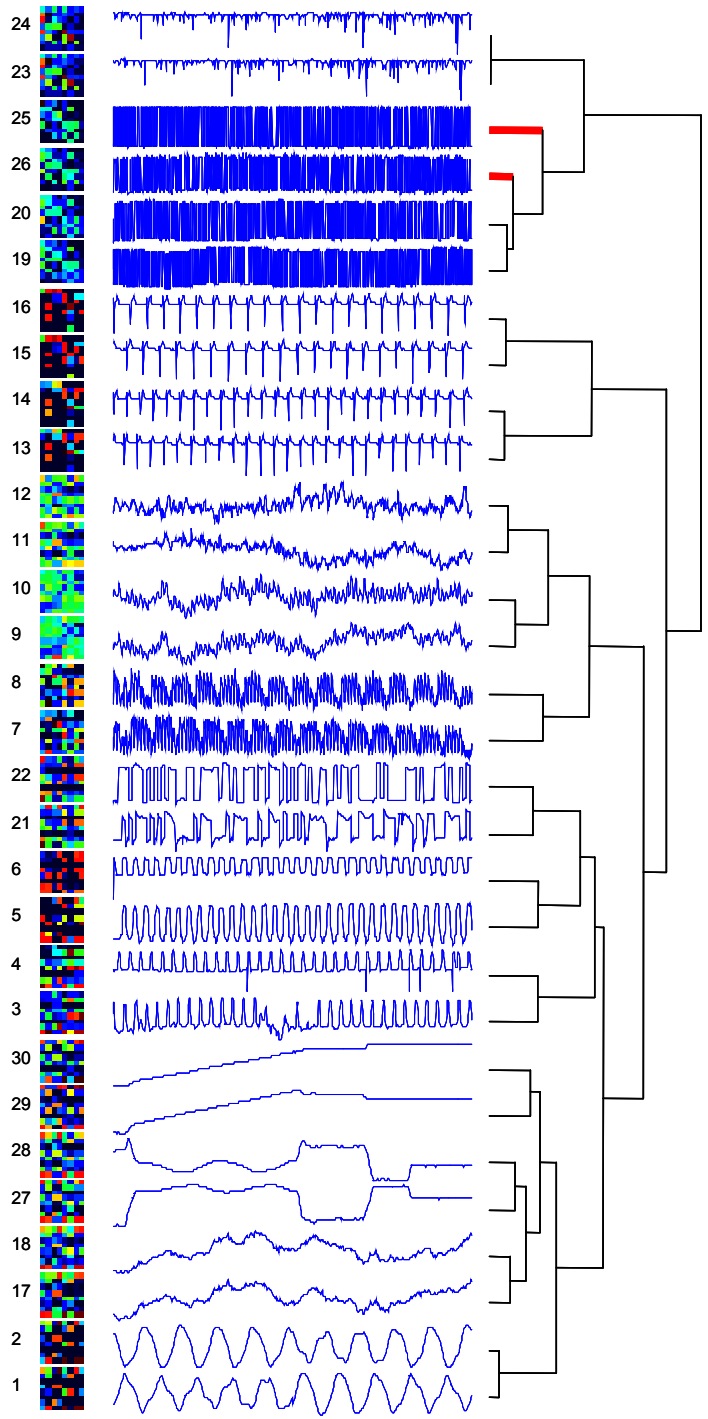
Long Term ST Database (ltstdb): record 20021

Start times at 0, 50, 100, 150, 200, respectively

Cluster 4 (datasets 16 ~ 20):

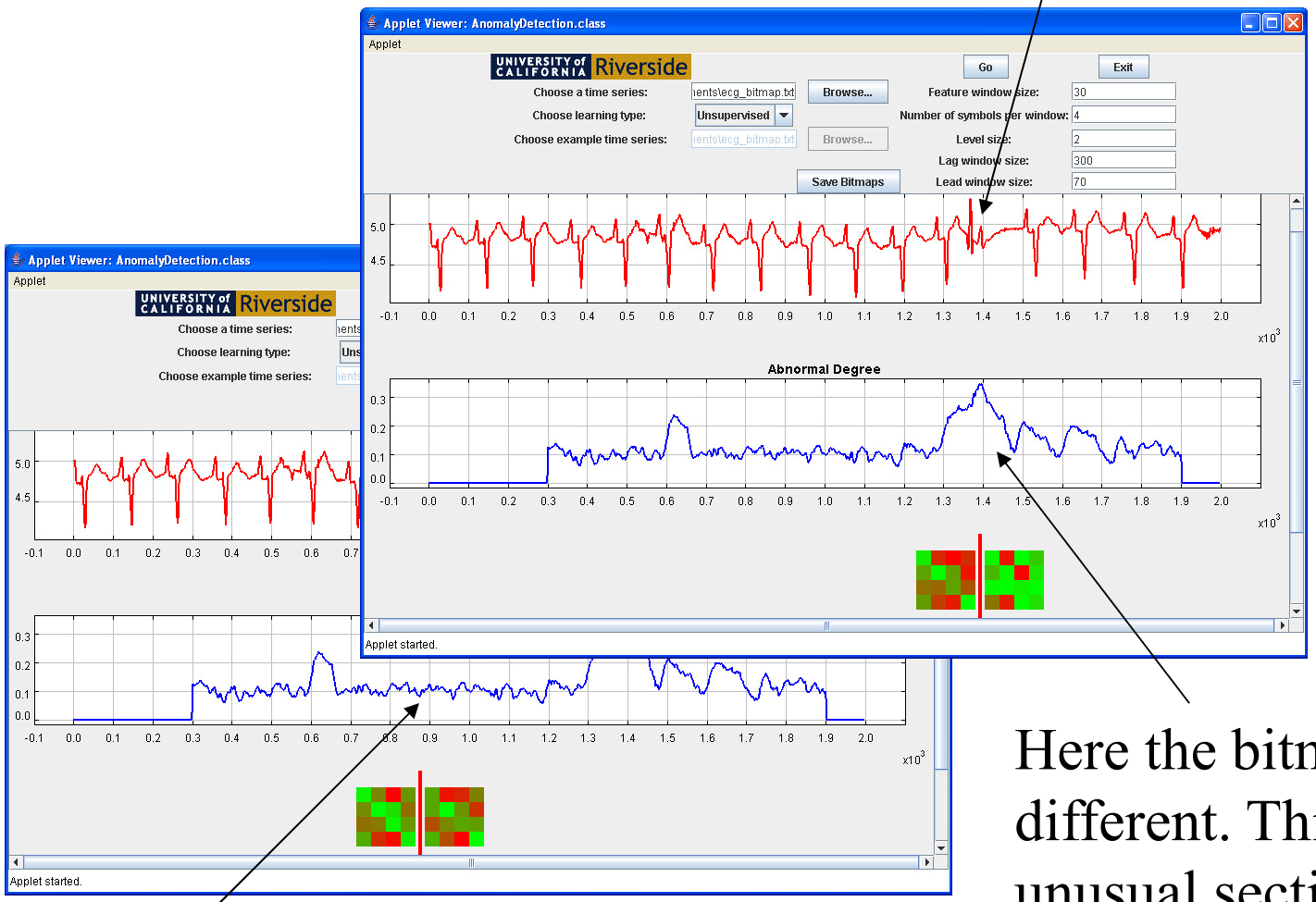
MIT-BIH Noise Stress Test Database (nstdb): record 118e6

Start times at 0, 50, 100, 150, 200, respectively



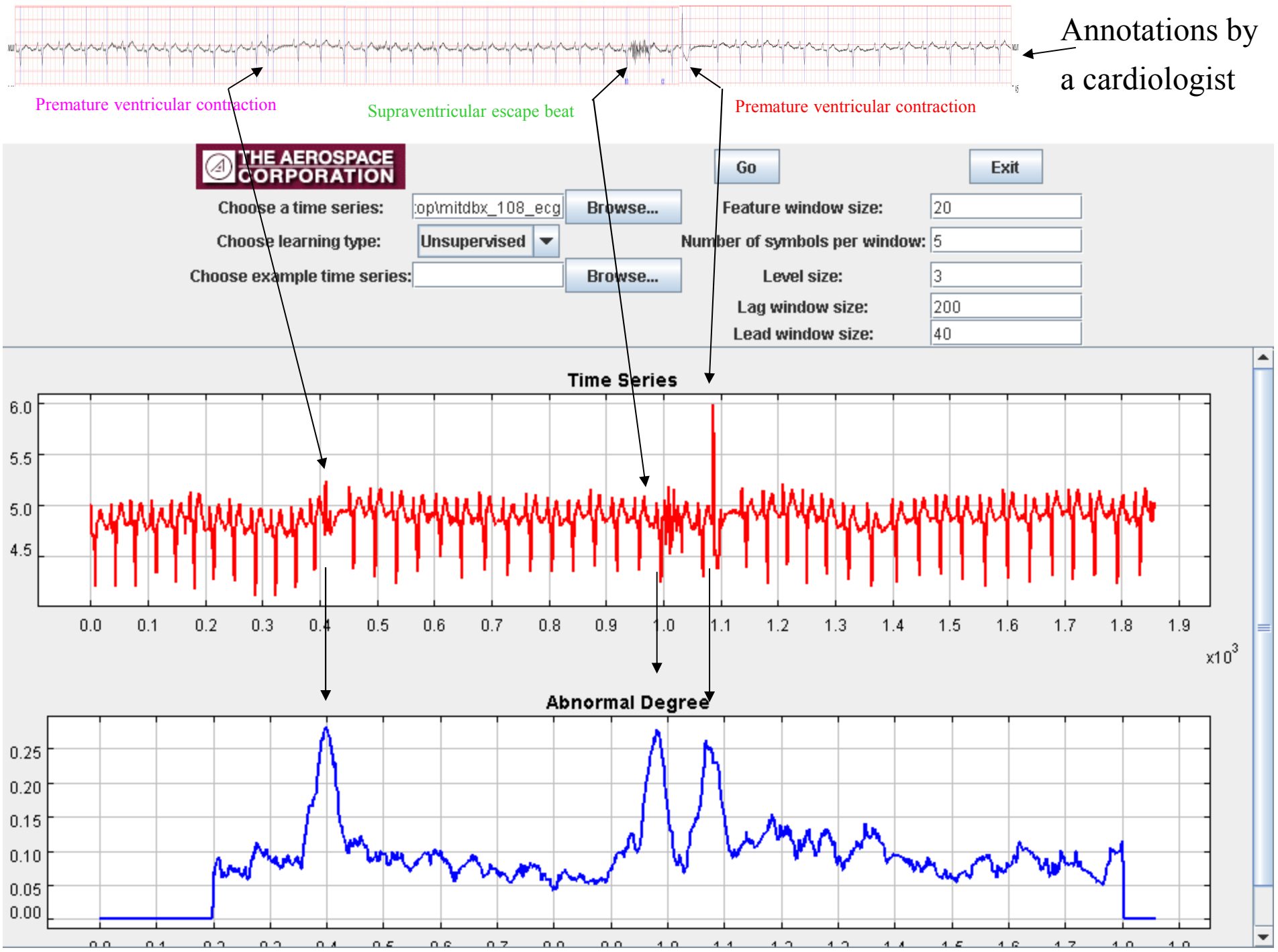
We can test how much useful information is retained in the bitmaps by using *only* the bitmaps for clustering/classification/anomaly detection

Here is a Premature Ventricular Contraction (PVC)



Here the bitmaps are almost the same.

Here the bitmaps are very different. This is the most unusual section of the time series, and it coincidences with the PVC.


Annotations by a cardiologist
Premature ventricular contraction
Supraventricular escape beat
Premature ventricular contraction
THE AEROSPACE CORPORATION
Go
Exit
Choose a time series:
:op\mitdbx_108_ecg
Browse...
Feature window size:
20
Choose learning type:
Unsupervised
Number of symbols per window:
5
Choose example time series:
Browse...
Level size:
3
Lag window size:
200
Lead window size:
40
Time Series
6.0
5.5
5.0
4.5
0.0 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1.0 1.1 1.2 1.3 1.4 1.5 1.6 1.7 1.8 1.9
x10^3
Abnormal Degree
0.25
0.20
0.15
0.10
0.05
0.00


The Last Word
The sun is setting on all other symbolic representations of time series, SAX is the *only* way to go

SAX Summary

• For most classic data mining tasks (classification, clustering and indexing), SAX is at least as good as the raw data, DFT, DWT, SVD etc.

• SAX allows the best anomaly detection algorithm.

• SAX is the *engine* behind the only realistic time series motif discovery algorithm.


The Last Word
The sun is setting on all other symbolic representations of time series, SAX is the *only* way to go

Conclusions

- SAX is posed to make major contributions to time series data mining in the next few years.

- A more general conclusion, if you want to solve you data mining problem, think representation, representation, representation.

The slides that follow demonstrate that SAX is as good as DFT, DWT etc for the classic data mining tasks, this is important, but not very exciting, thus relegated to this appendix.

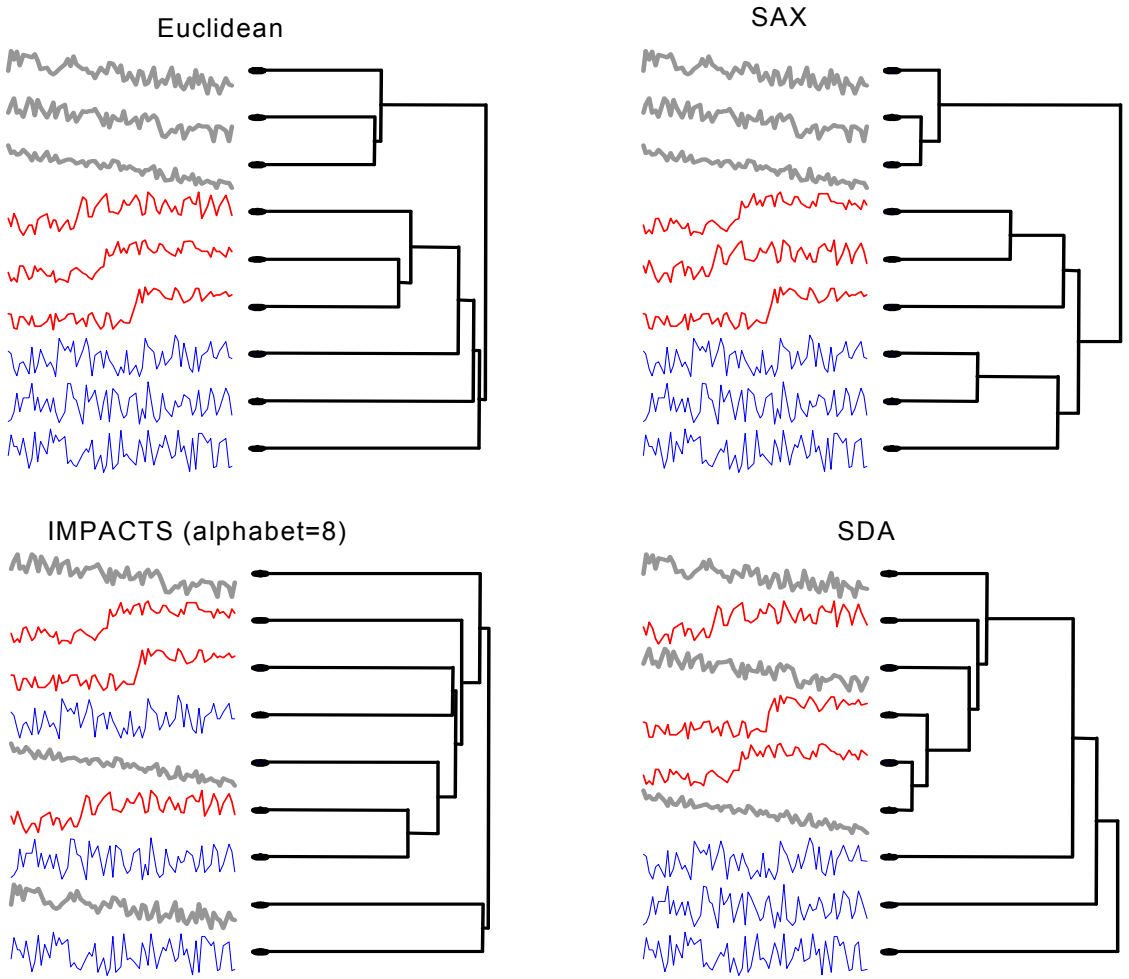
Experimental Validation

- Clustering
  - Hierarchical
  - Partitional
- Classification
  - Nearest Neighbor
  - Decision Tree
- Indexing
  - VA File
- Discrete Data only
  - Anomaly Detection
  - Motif Discovery

Clustering

- Hierarchical Clustering
  - Compute pairwise distance, merge similar clusters bottom-up
  - Compared with Euclidean, IMPACTS, and SDA

Hierarchical Clustering

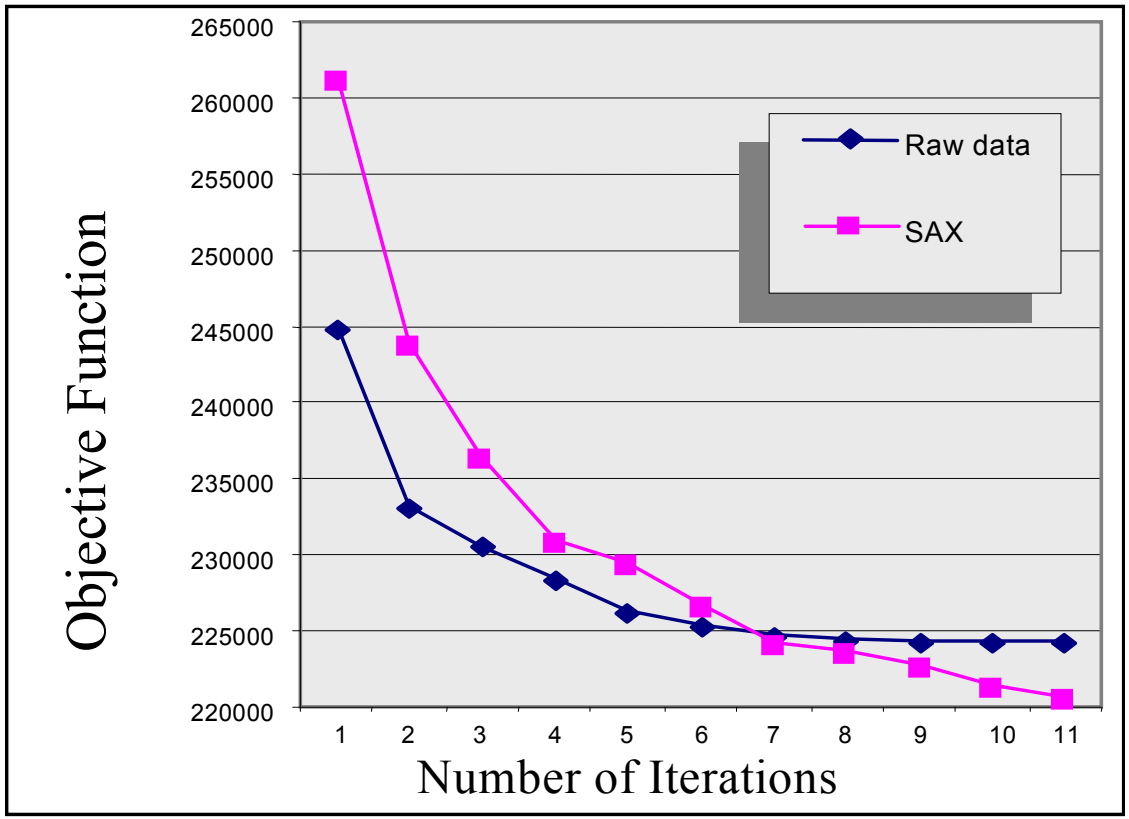


Hierarchical Clustering

Clustering

- Hierarchical Clustering
  - Compute pairwise distance, merge similar clusters bottom-up
  - Compared with Euclidean, IMPACTS, and SDA
- Partitional Clustering
  - K-means
  - Optimize the objective function by minimizing the sum of squared intra-cluster errors
  - Compared with Raw data

Partitional (K-means) Clustering

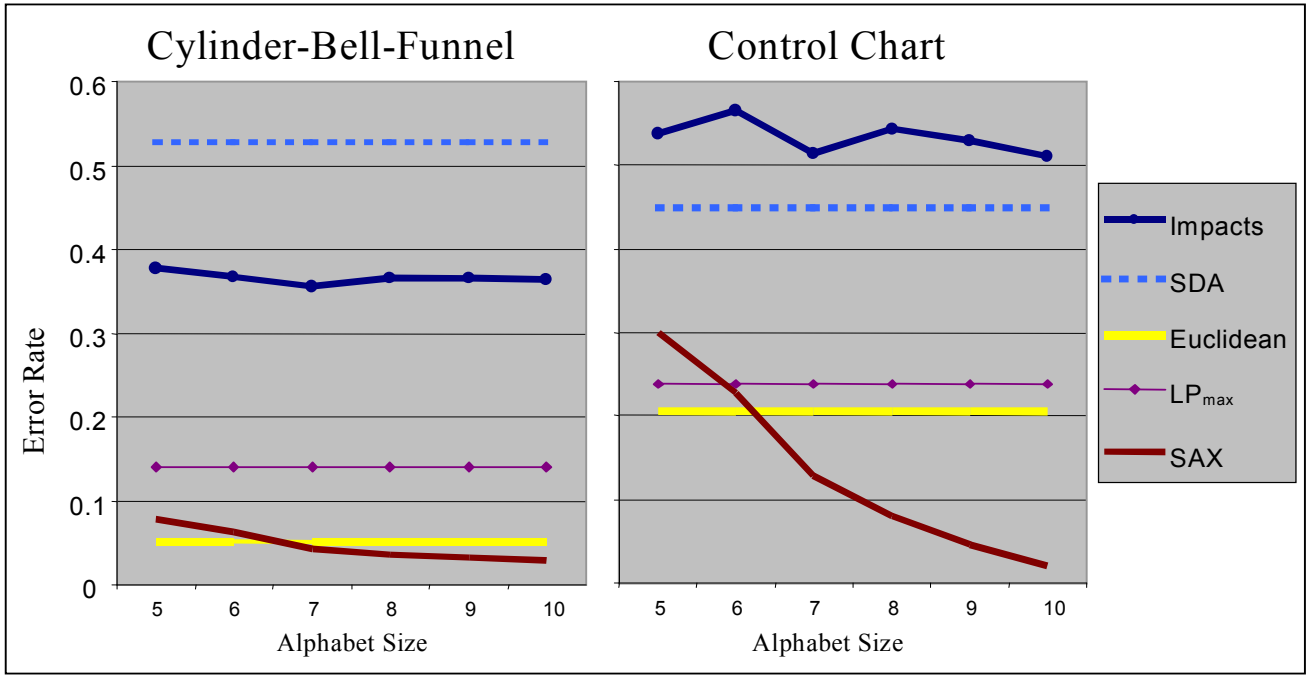


Partitional (k-means) Clustering

Classification

- Nearest Neighbor
  - Leaving-one-out cross validation
  - Compared with Euclidean Distance, IMPACTS, SDA, and LP∞
  - Datasets: Control Charts & CBF (Cylinder, Bell, Funnel)

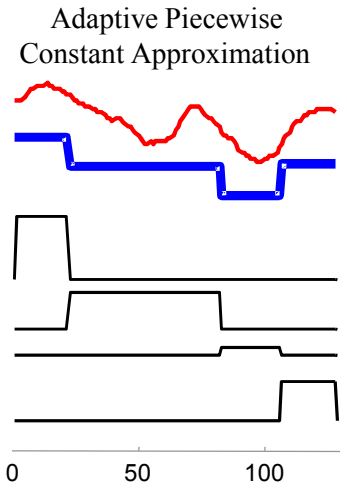
Nearest Neighbor



Nearest Neighbor

Classification

- Nearest Neighbor
  - Leaving-one-out cross validation
  - Compared with Euclidean Distance, IMPACTS, SDA, and LP∞
  - Datasets: Control Charts & CBF (Cylinder, Bell, Funnel)
- Decision Tree
  - Defined for real data, but attempting to use DT on time series raw data would be a mistake
    - High dimensionality/Noise level would result in deep, bushy trees
  - Geurts (’01) suggests representng time series as Regression Tree, and training decision tree on it.



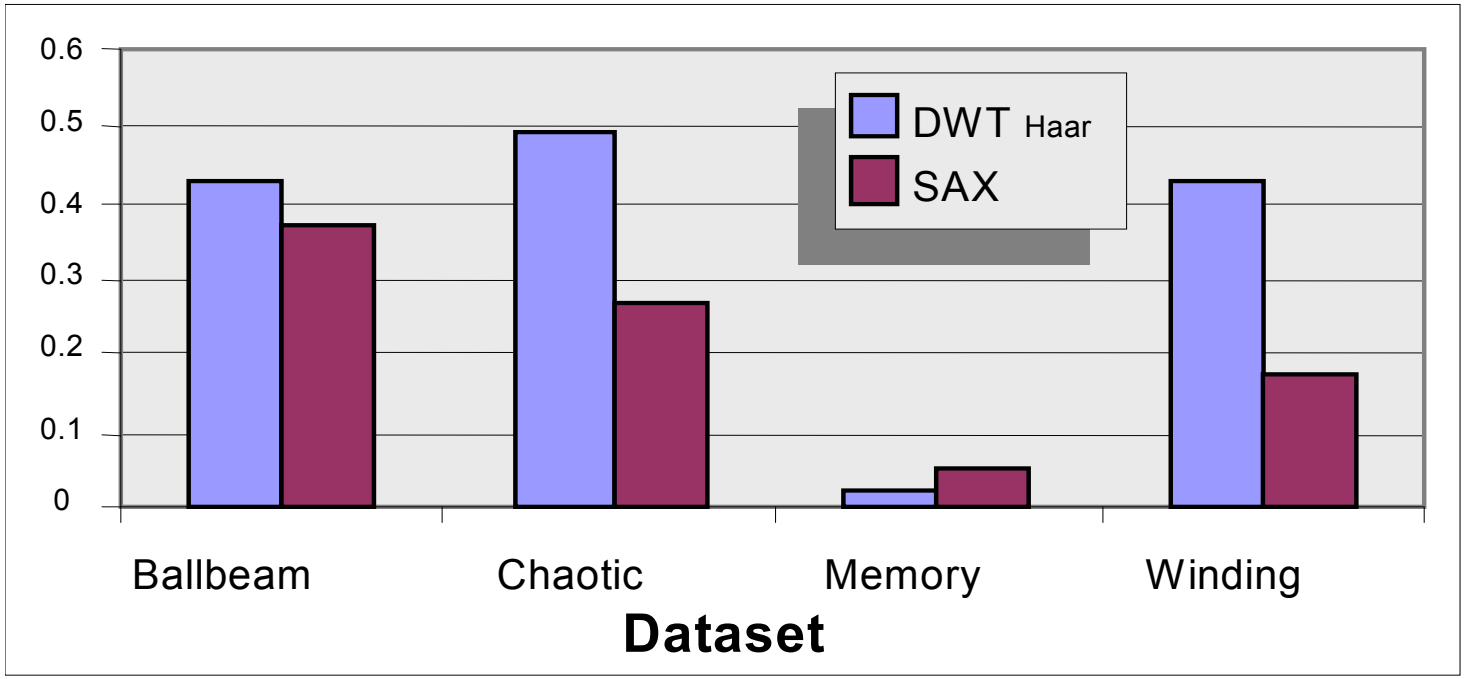
Decision (Regression) Tree

Dataset	SAX	Regression Tree
CC	3.04 ± 1.64	2.78 ± 2.11
CBF	0.97 ± 1.41	1.14 ± 1.02

Indexing

- Indexing scheme similar to VA (Vector Approximation) File
  - Dataset is large and disk-resident
  - Reduced dimensionality could still be too high for R-tree to perform well
- Compare with Haar Wavelet

Indexing


0.6
0.5
0.4
0.3
0.2
0.1
0
DWT Haar
SAX
Ballbeam
Chaotic
Memory
Winding
Dataset