Clustering of Time Series Data Using Consensus Motifs

Achyuth Madhav Diwakar

Advisor: Prof. Eamonn Keogh

Abstract

The ubiquitous nature of time series and the need to effectively analyze time series data has provided the motivation for this project. The consensus motif search algorithm is a powerful tool that can retrieve the best repeated structure in a dataset of time series. The objective of this project is to use consensus motifs to cluster the time series into the classes to which they belong. Using the properties of consensus motifs and a forward selection approach a clustering algorithm was developed. Experiments performed on synthetic and real world data provide further evidence that consensus motifs can be used for time series clustering. The MATLAB application developed for this purpose, successfully clustered time series of different classes with high accuracy. The properties of the algorithm and scope for future work has also been covered in this research endeavor.

Overview

- Get acquainted with time series, matrix profile and consensus motifs
- Results of the consensus motif search algorithm obtained from single class and multi-class datasets
- Exploit the results to develop an algorithm to cluster the time series
- Test the algorithm on synthetic and real world data
- Challenges and conclusions
- Future work

Background: What is a Time Series?

- A time series is a sequence of numerical data points in successive order.
- This data is usually recorded at fixed intervals of time.
- Example: The plot represents a time series with the average share prices of Apple Inc. recorded once per day, from 02/21/2019 to 03/20/2019.

Background: Matrix Profile and Motifs

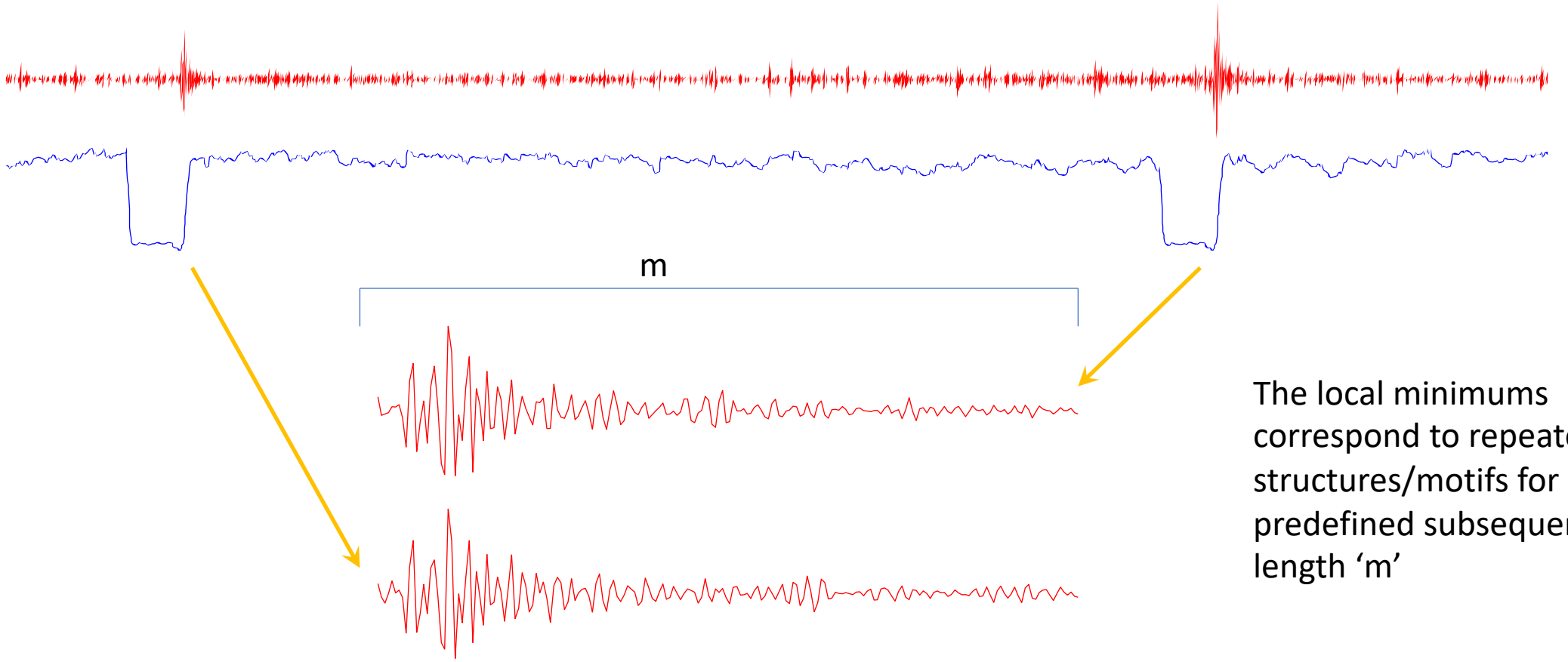
- A Matrix Profile is a vector, which stores the Euclidean distance between a subsequence and its nearest neighbor in another time series.
- The matrix profile of a time series can also be calculated on itself thus finding distances of nearest neighbors within the time series.

Ref: Prof. Keogh's slides on Matrix Profile: https://www.cs.ucr.edu/%7Eeamonn/matrix_profile_i.pptx

UCR

Background: Matrix Profile and Motifs

- Seismologists are interested in finding repeated earthquakes in long sequence of seismometer readings, as the repeated earthquakes could be years apart



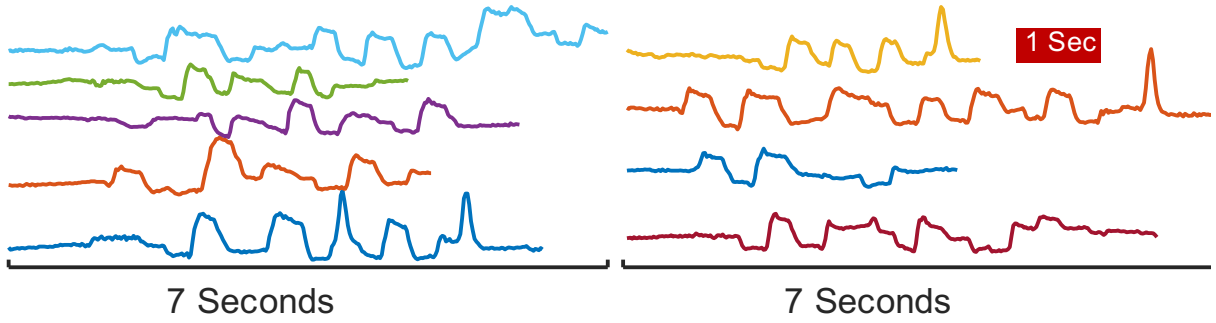
The local minimums correspond to repeated structures/motifs for a predefined subsequence length ‘m’

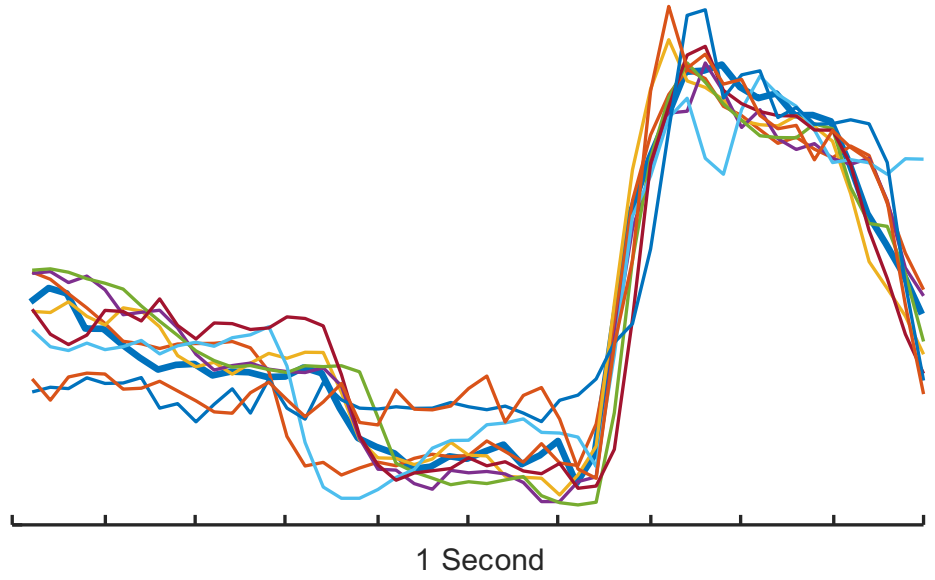
Ref: Prof. Keogh’s slides on Matrix Profile: https://www.cs.ucr.edu/%7Eeamonn/matrix_profile_i.pptx

UCR

Background: Consensus Motifs

- Given 9 time series corresponding to different sentences (in Japanese) spelled out by the eye movements(along vertical axis) of an individual modeling locked-in syndrome.
- Subsequence length selected = 1 s
- The algorithm then finds a repeated structure, of length 1 second, in the set of time series.
- This repeated pattern is called a consensus motif.



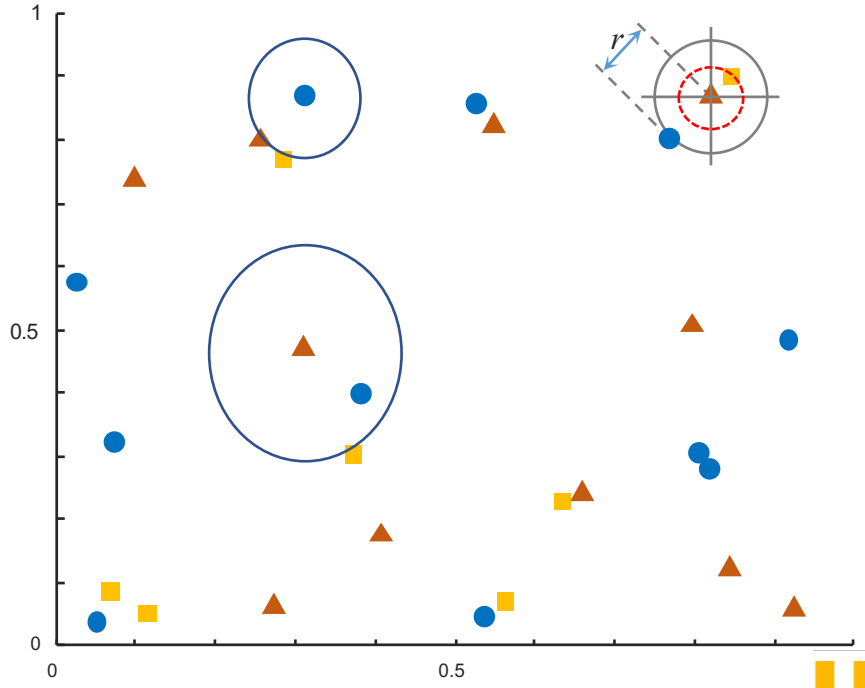


Ref: Finding and Exploiting Time Series Consensus Motifs – Prof. Eamonn Keogh Research Group, UC Riverside

Background: Consensus Motifs - Radius

- The *radius r* of a subsequence *Tij,m* of time series *Ti* with respect to a sequence of time series *T1...Tk* is the maximum distance between *Tij,m* and its nearest neighbor in each of *T1...Tk*.

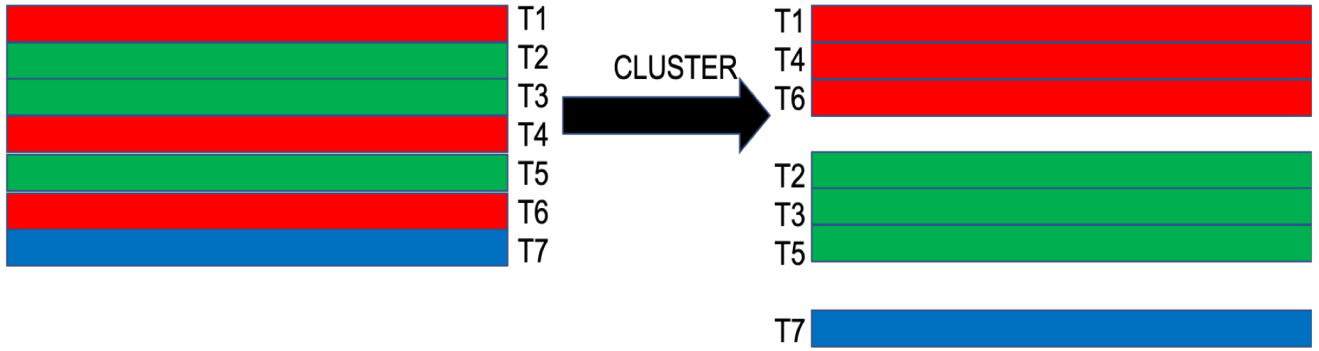
The subsequences of three time series, A●, B▲and C■ exist as points in a *m*-dimensional space. A hypersphere can be centered at each of the subsequences and have its radius *r* expanded until it includes at least one of each of the time series. The subsequence that has the smallest such *r* is the consensus motif.



Ref: Finding and Exploiting Time Series Consensus Motifs – Prof. Eamonn Keogh Research Group, UC Riverside

Objective

- Given a set of N time series, each belonging to one of k classes, cluster them into their corresponding classes using the consensus motif search algorithm.



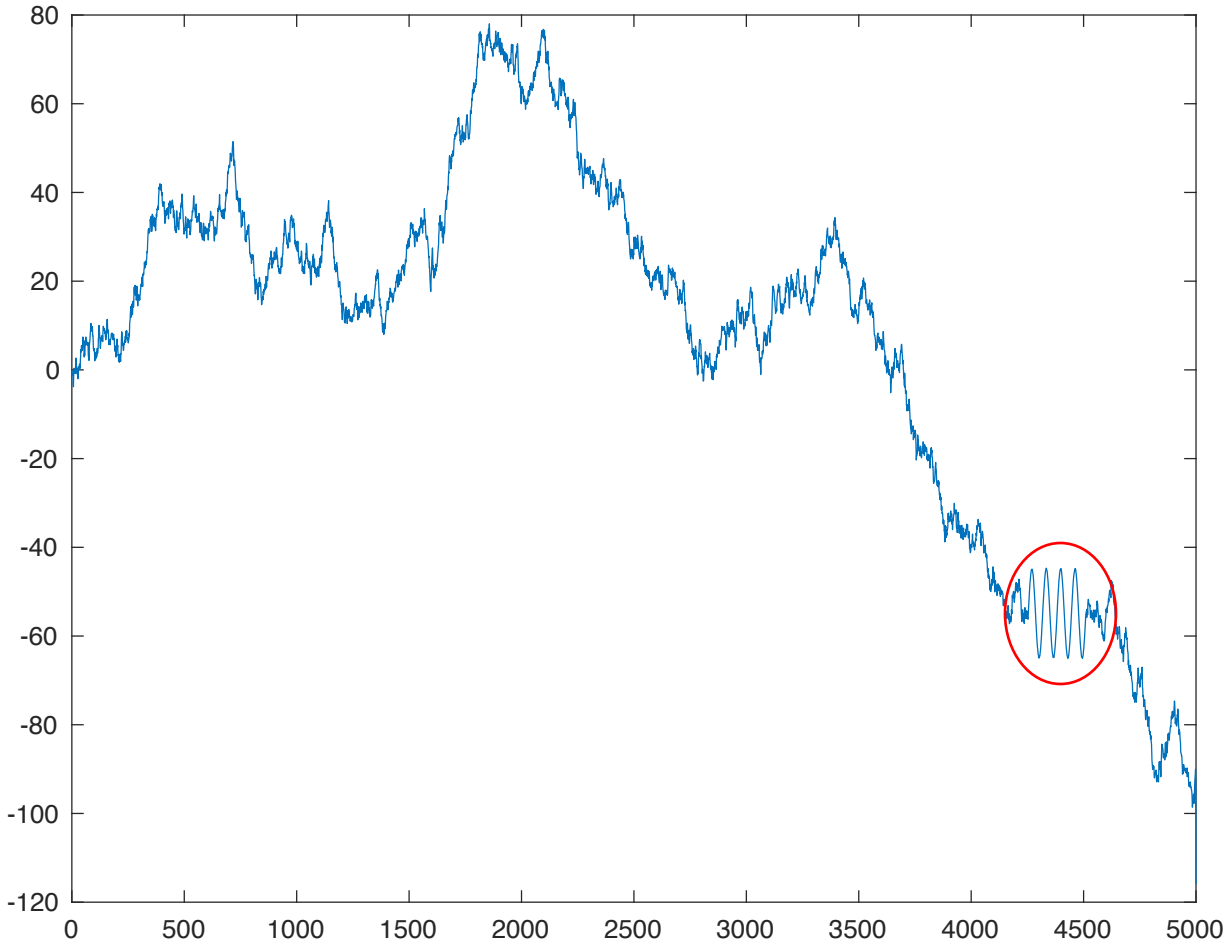
Consensus Motifs in Action

- First, we perform experiments with consensus motifs on datasets of different composition.
- The datasets used initially are synthetic and experiments are performed on 1-class and 2-class datasets.
- The results from the consensus motif search algorithm would provide insights to develop a clustering technique using consensus motifs.

UCR

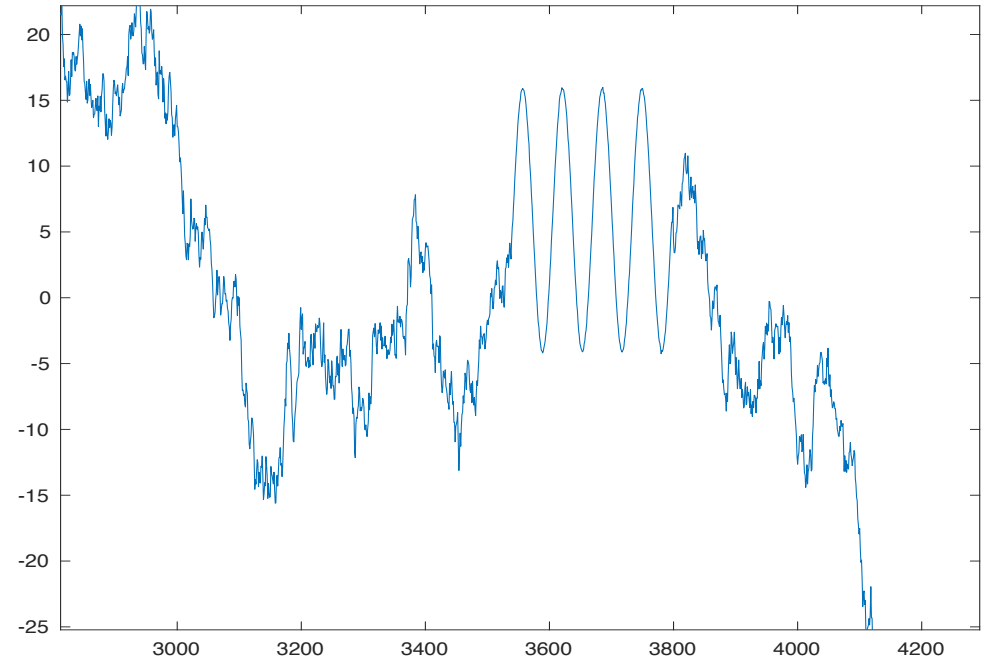
Synthetic data – Sine wave

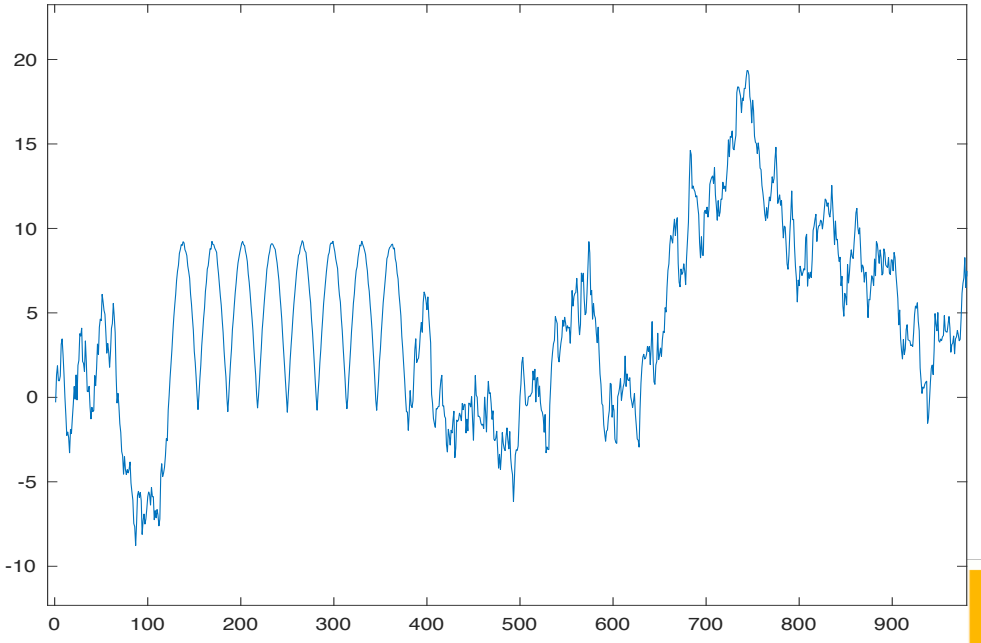
- A synthetic time series is created by generating a random walk or just random data and incorporating some meaningful data within.
- Generated a random walk of length 5000.
- Sine wave of length 256 inserted at a random location inside it.



Synthetic data – Sine wave and abs(Sine wave)

- We now have two classes of time series, each representing a sine wave or the absolute value of a sine wave.
- The size of meaningful data in both classes is 256 points.
- The zoomed in graphs of the time series show the different motifs inserted in the random walk.

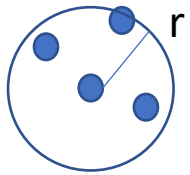


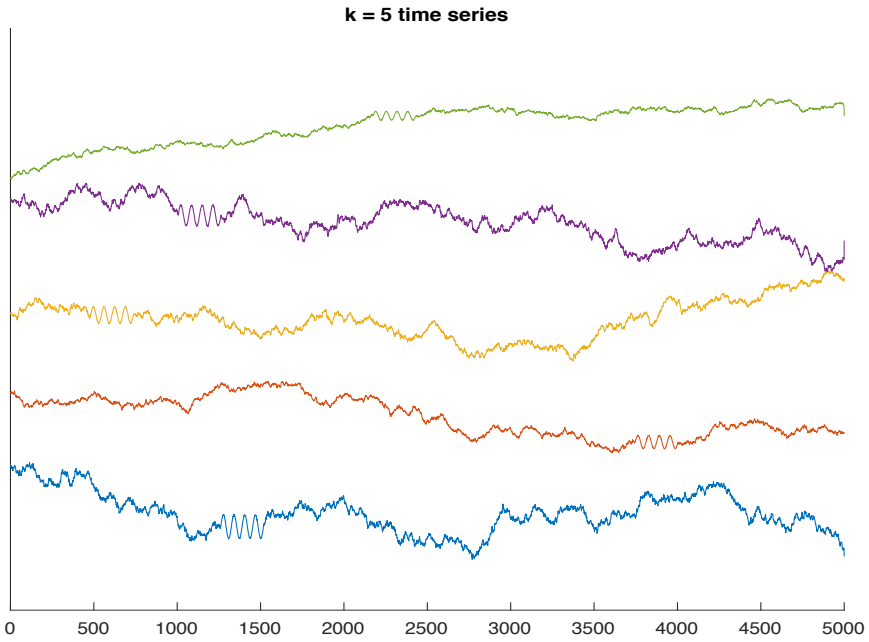


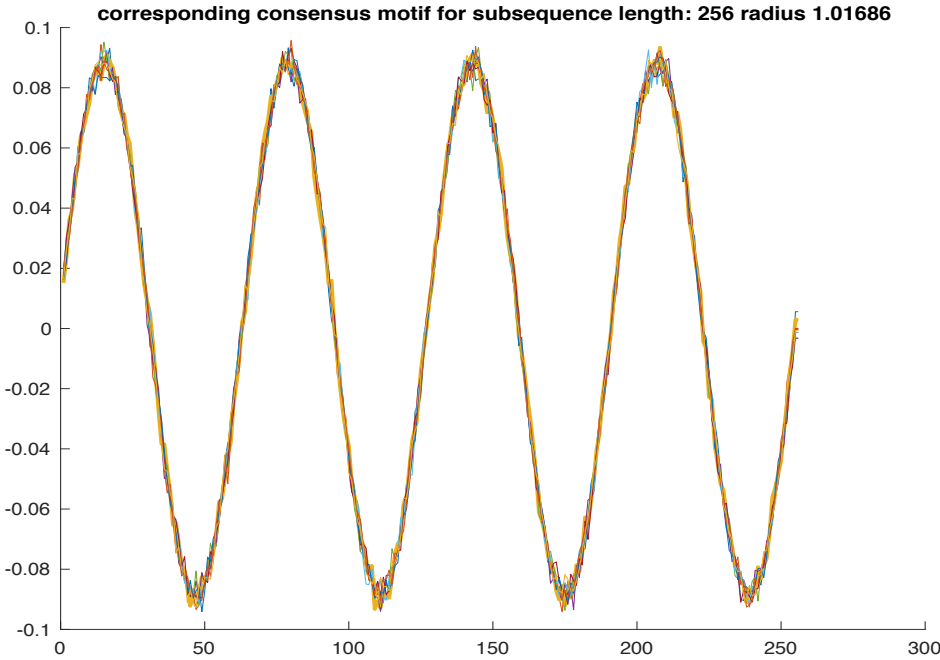
Consensus Motif: Homogenous dataset

- 5 time series of length 5000 each
- A sine wave of length 256 inserted at a random position in each time series
- Sine wave has a 10% white gaussian noise
- The consensus motif found is the sine wave
- Subsequence length: 256

- Radius observed = 1.01

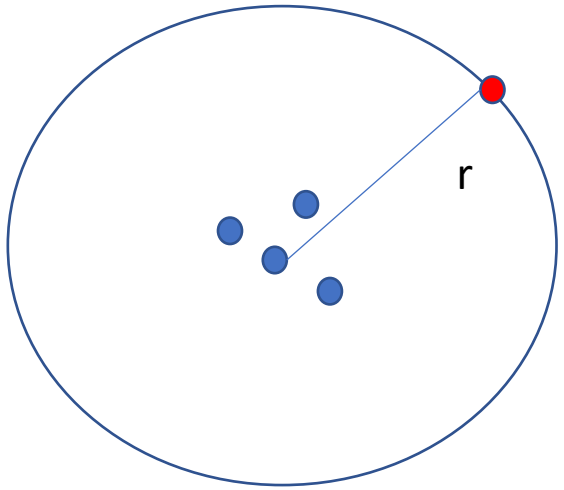


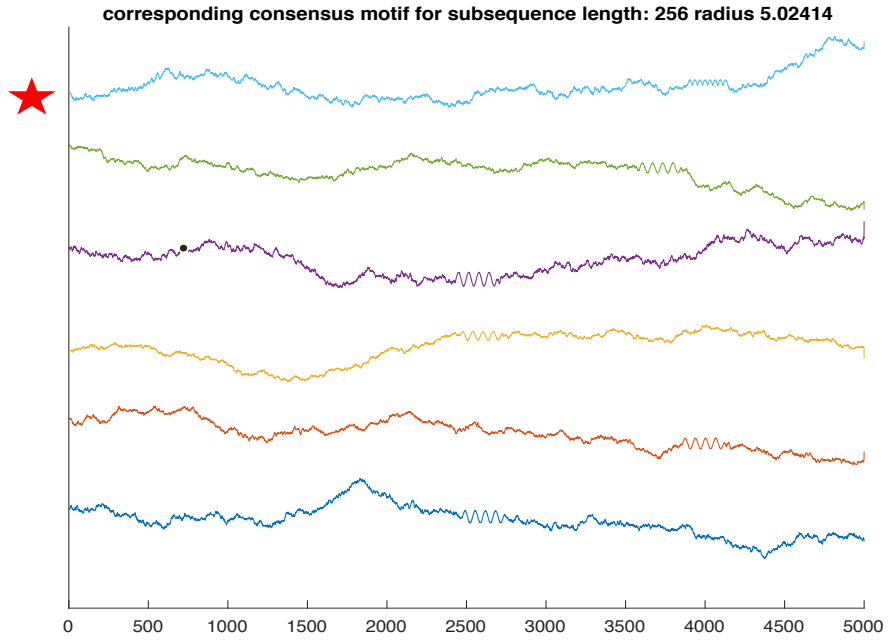


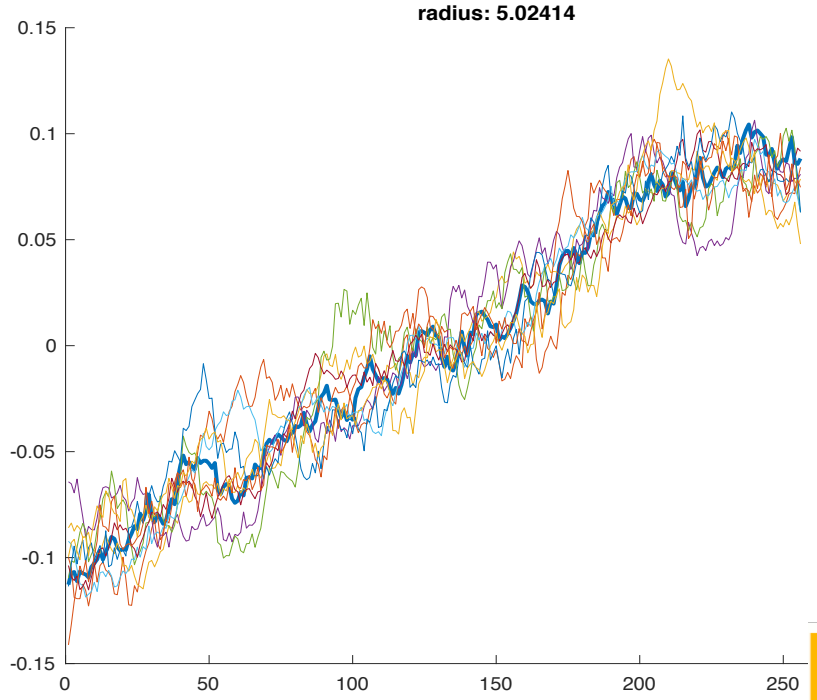


Introduce a different class into the dataset

- A sixth time series of length 5000 containing abs(sine) data of length 256 is added
- The consensus motif search now reveals a motif that is noisy
- **A drastic increase in radius is observed, with the introduction of a series belonging to a different class. (radius = 5.02)**

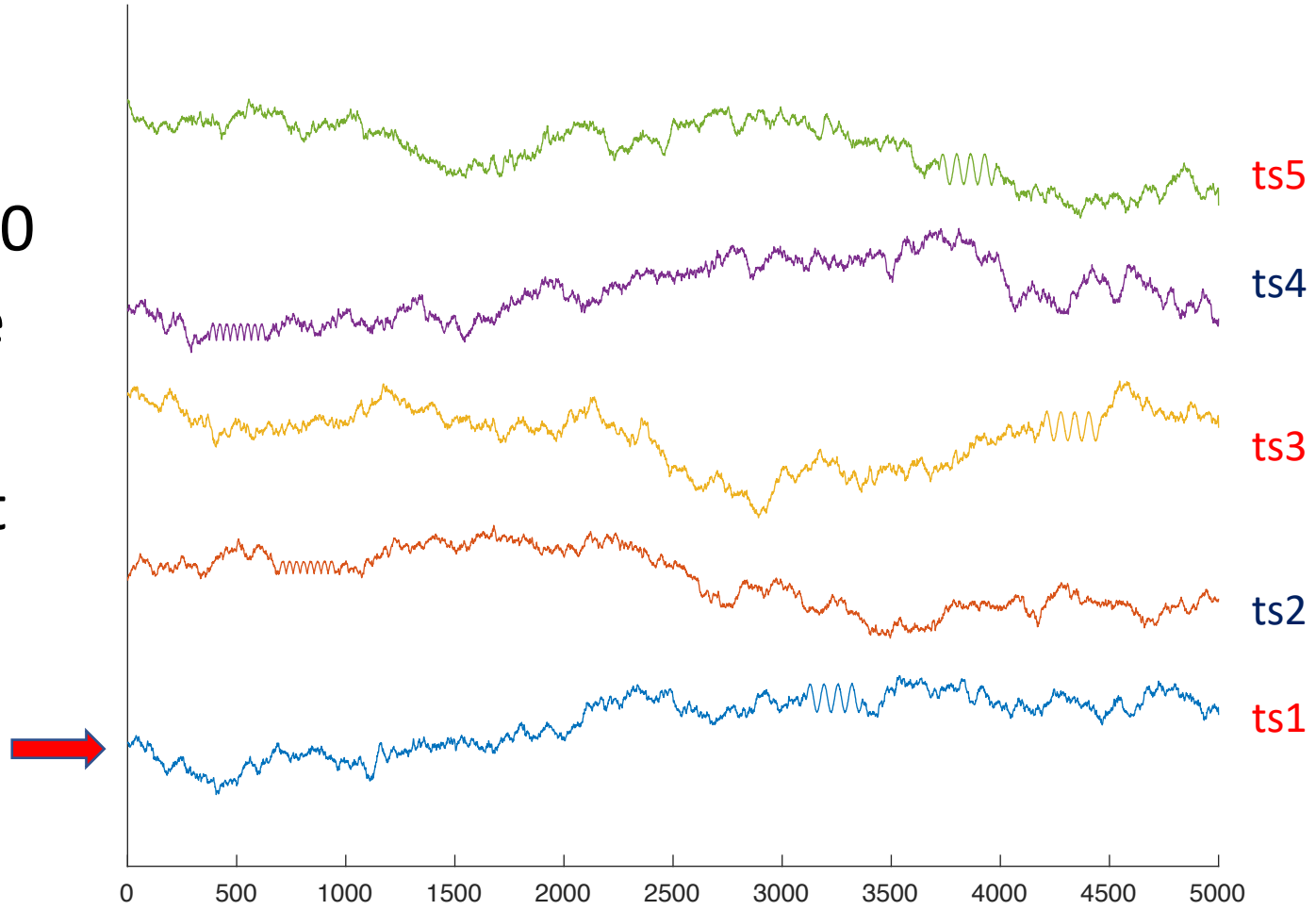






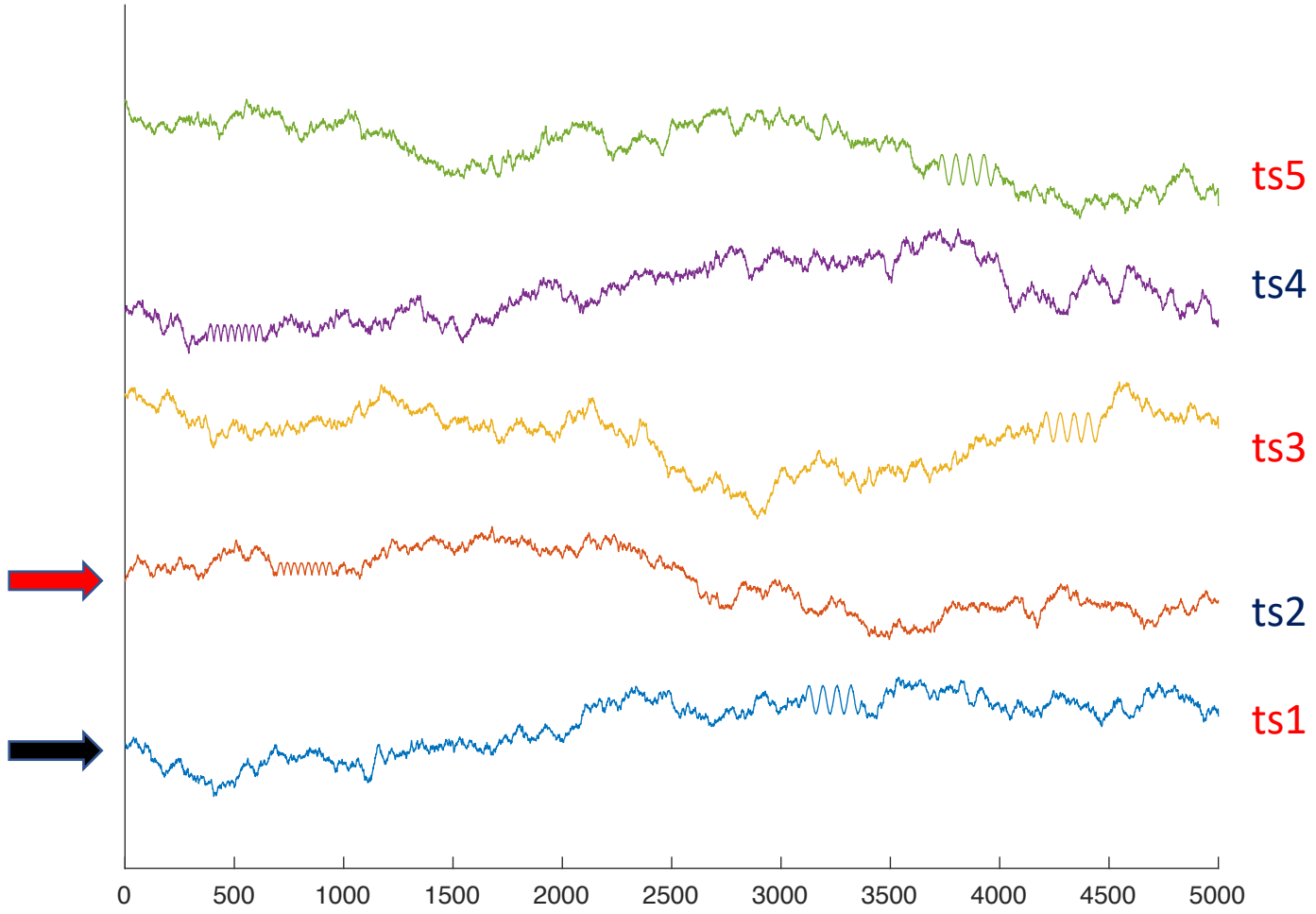
Clustering Algorithm

- First, the structure of the dataset is established
- 5 time series of length 5000
- ts1, ts3 and ts5 : sine wave
- ts2 and ts4 : abs(sine)
- ts1 is selected first (default start point)
- The radius obtained with each selection is plotted.
- Current radius = 0



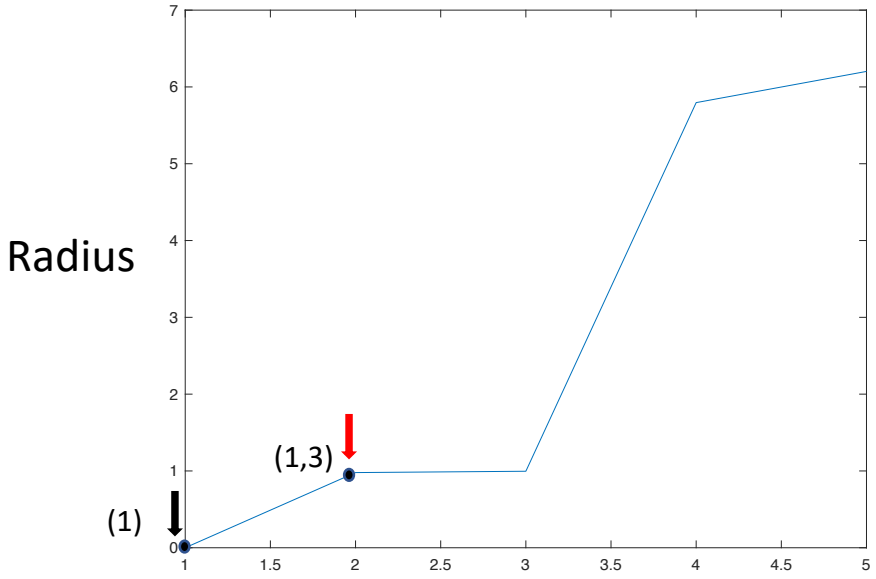
Clustering Algorithm

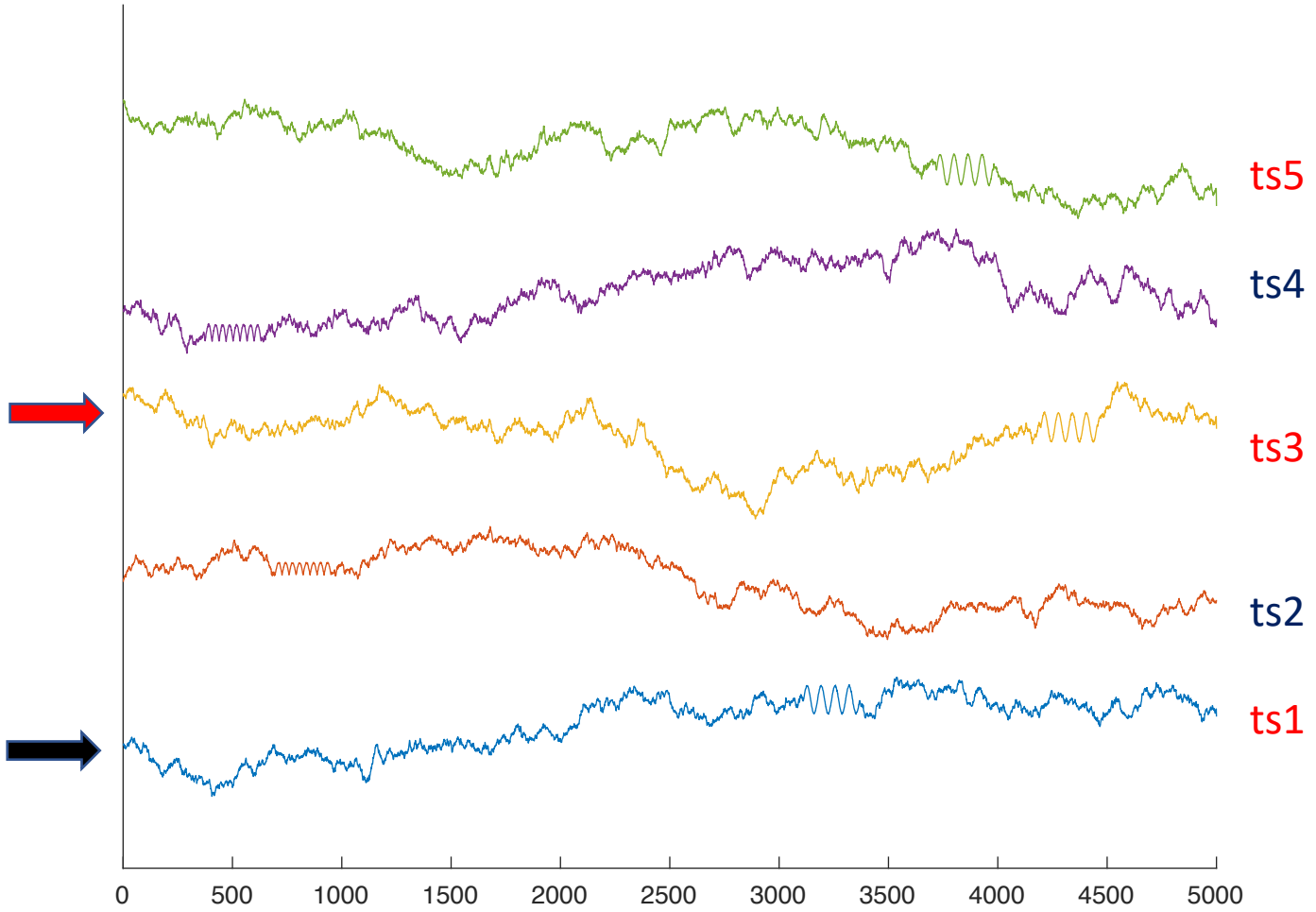
- The consensus motif search is run with ts1 and **each of the other time series**.
- The minimum consensus radius observed in all combinations can be clustered first



Clustering Algorithm

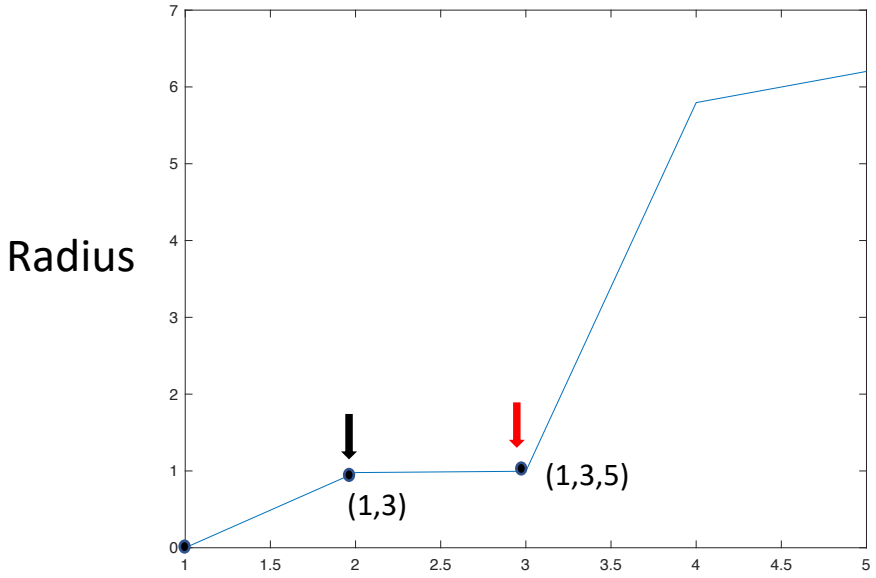
- The consensus motif search is run with ts1 and **each of the other time series**
- The smallest radius is observed for (ts1,ts3).

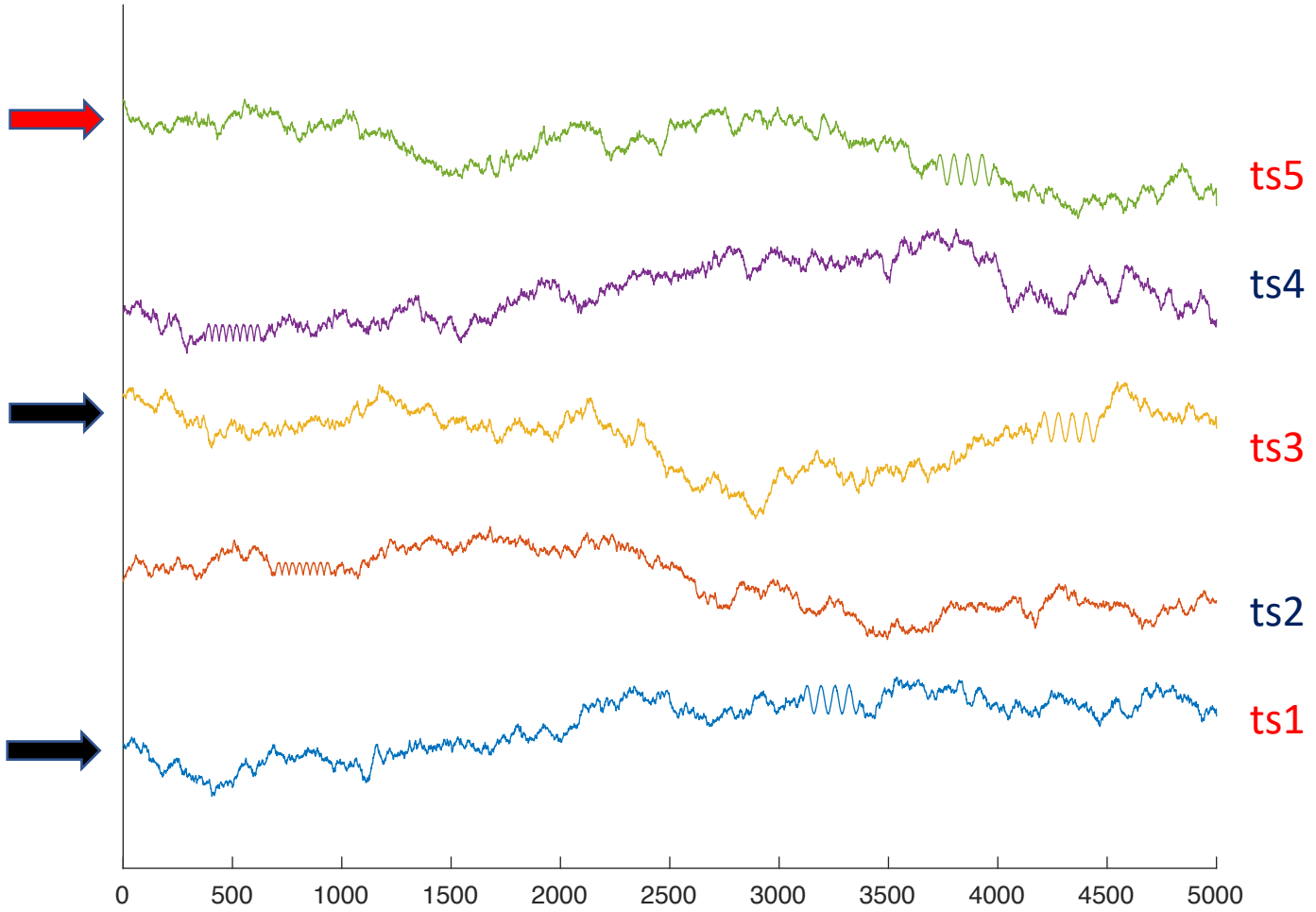




Clustering Algorithm

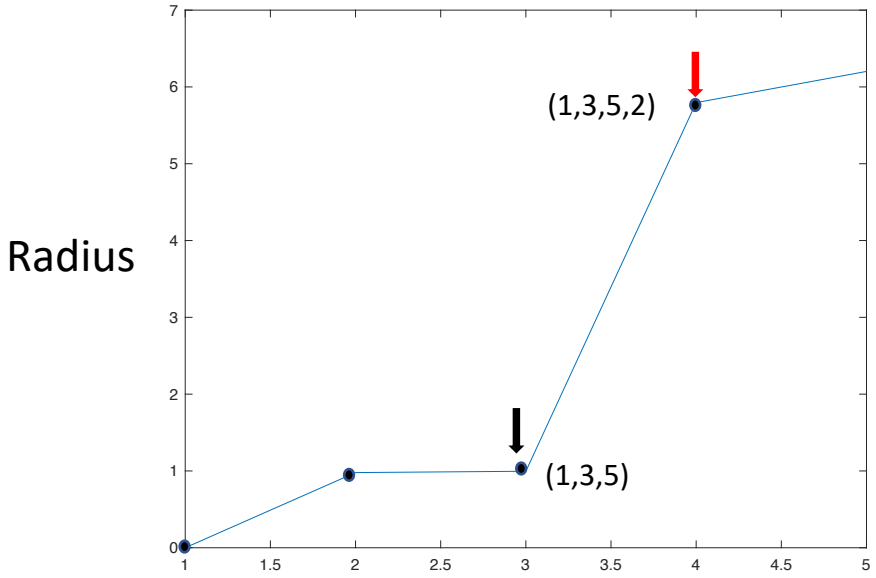
- Now we look for a third time series such that it has the smallest radius with (ts1, ts3).
- The minimum radius is found when ts5 is picked.

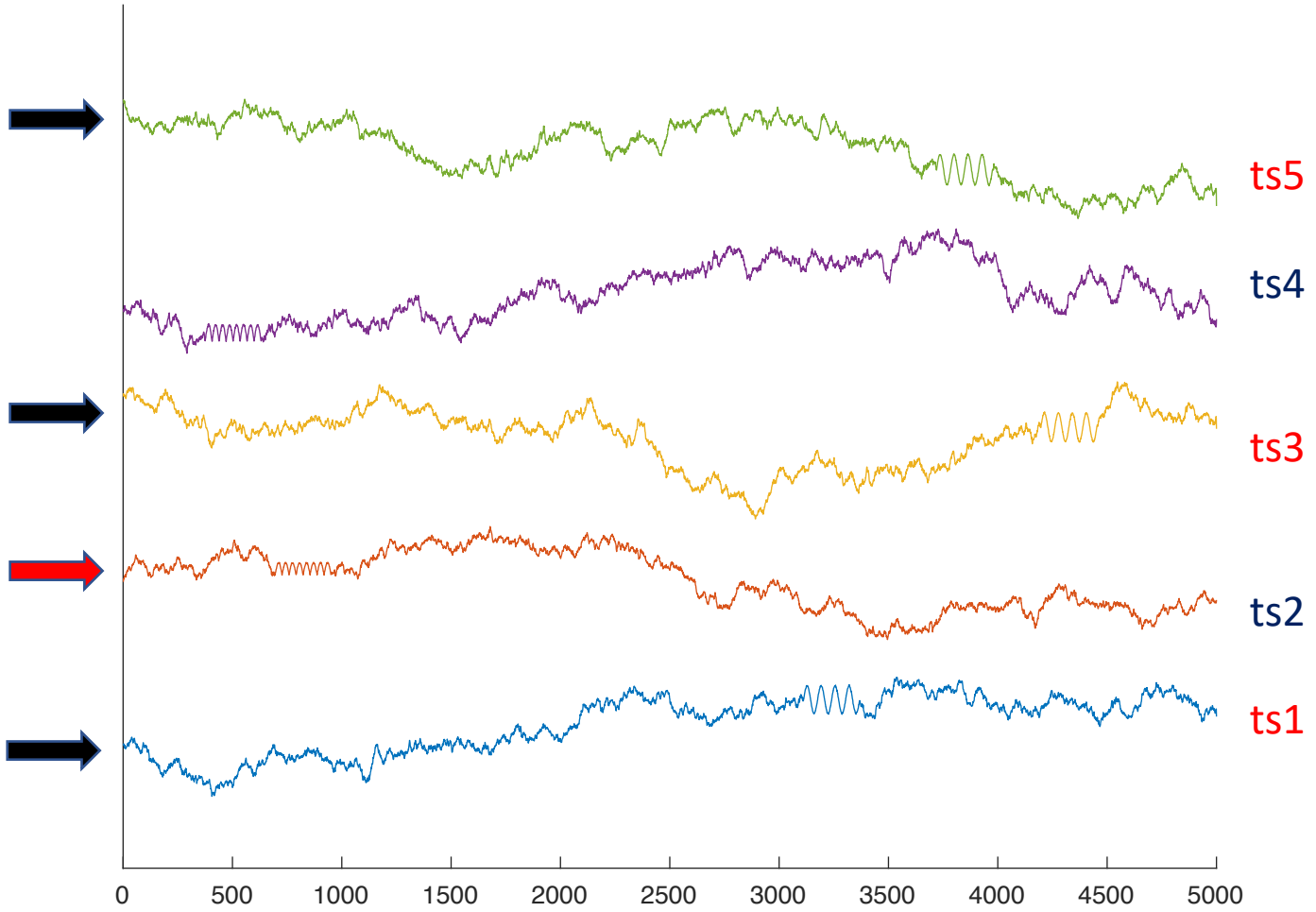




Clustering Algorithm

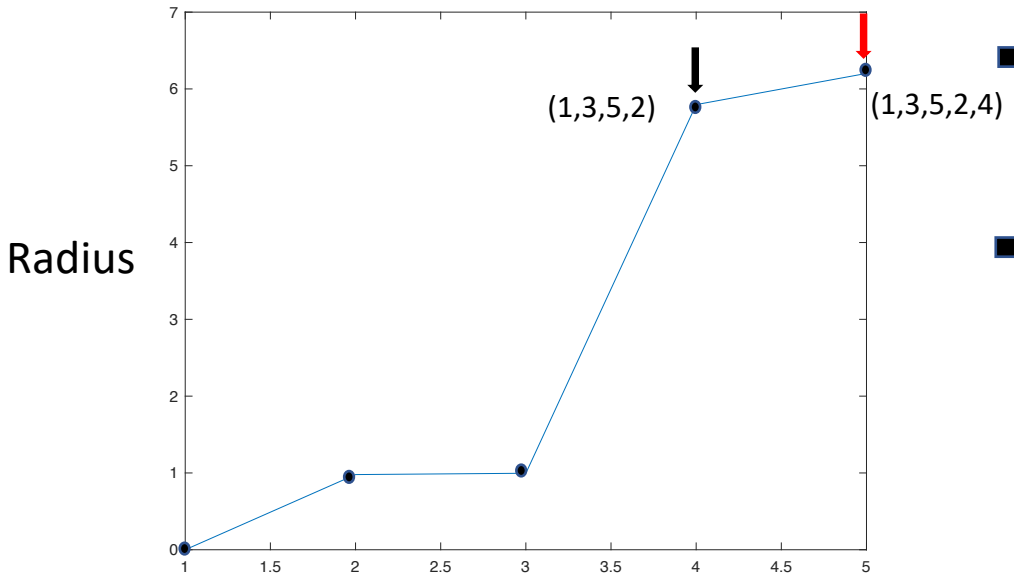
- Now, it is found that the smallest radius is obtained between (ts1,ts3,ts5) and ts4
- The **radius suddenly increases**.

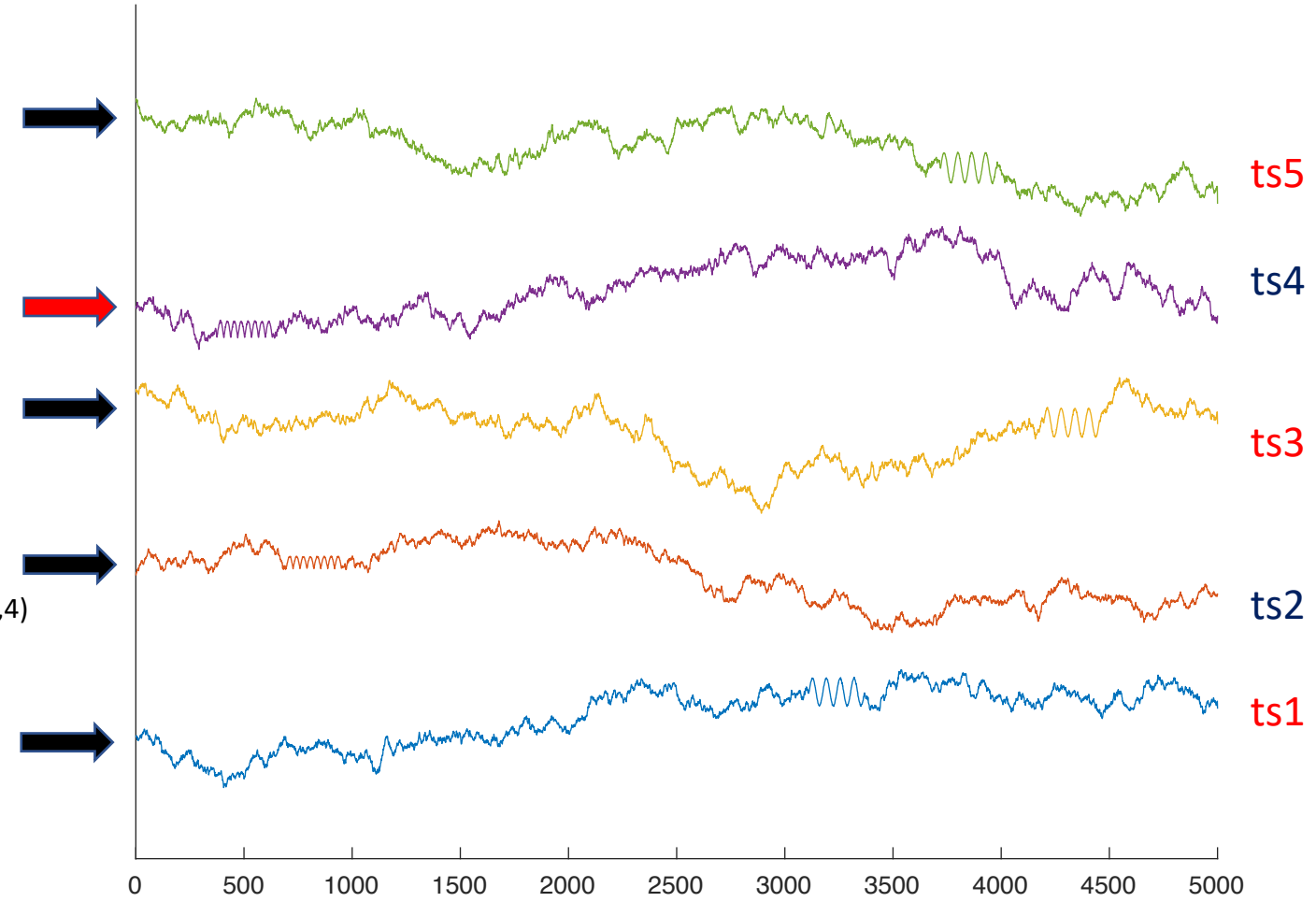




Clustering Algorithm

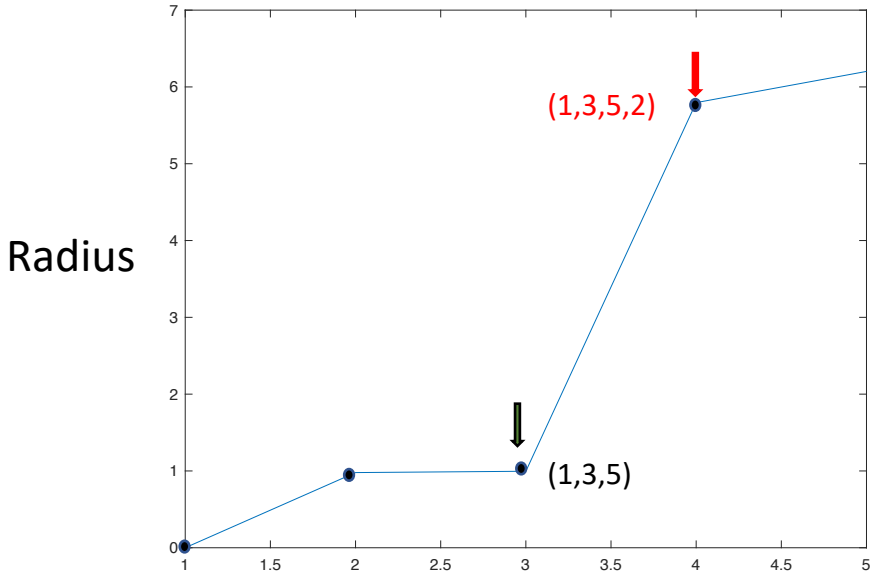
- Finally, ts4 is selected.
- The radius increases slightly.

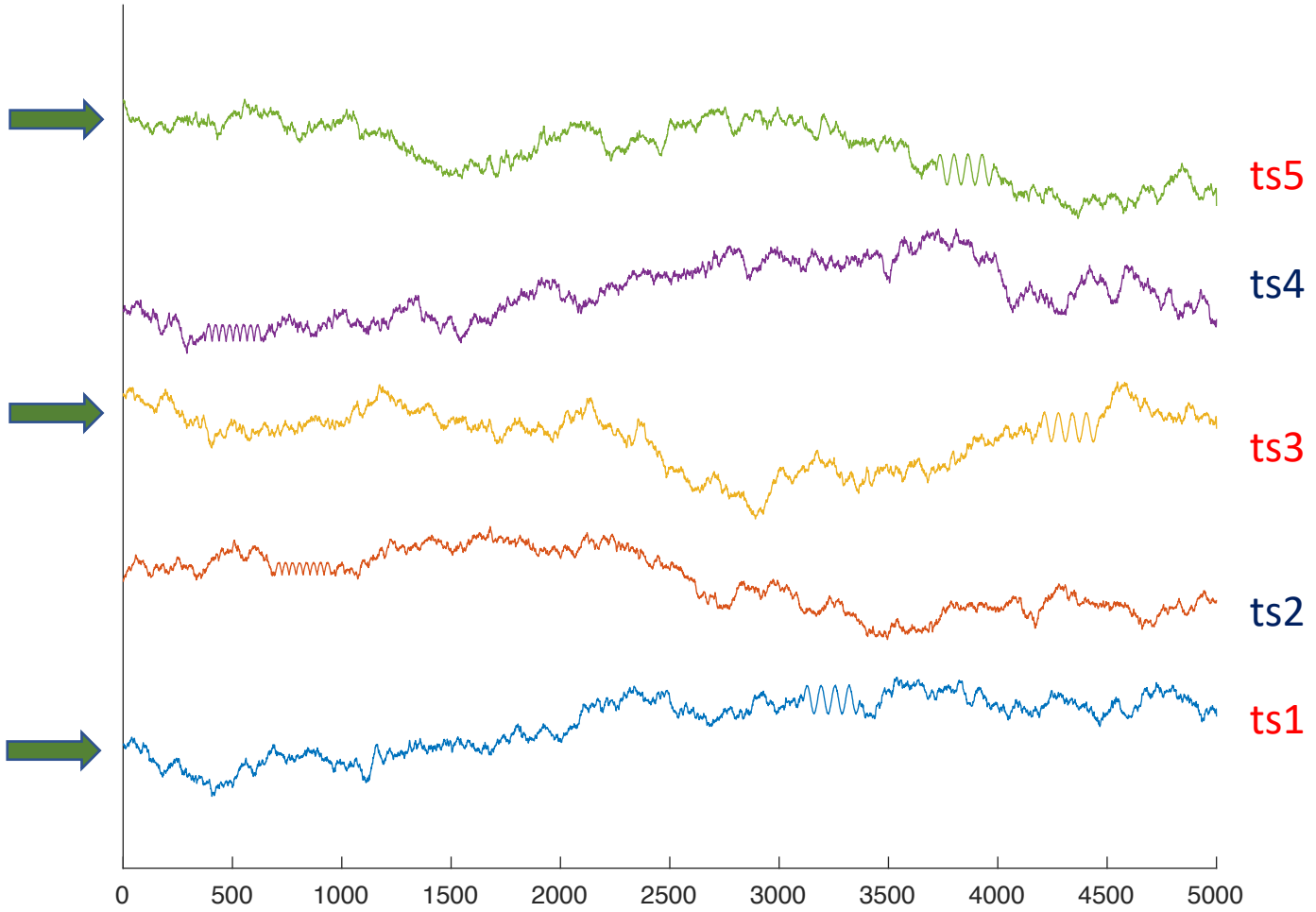




Clustering Algorithm

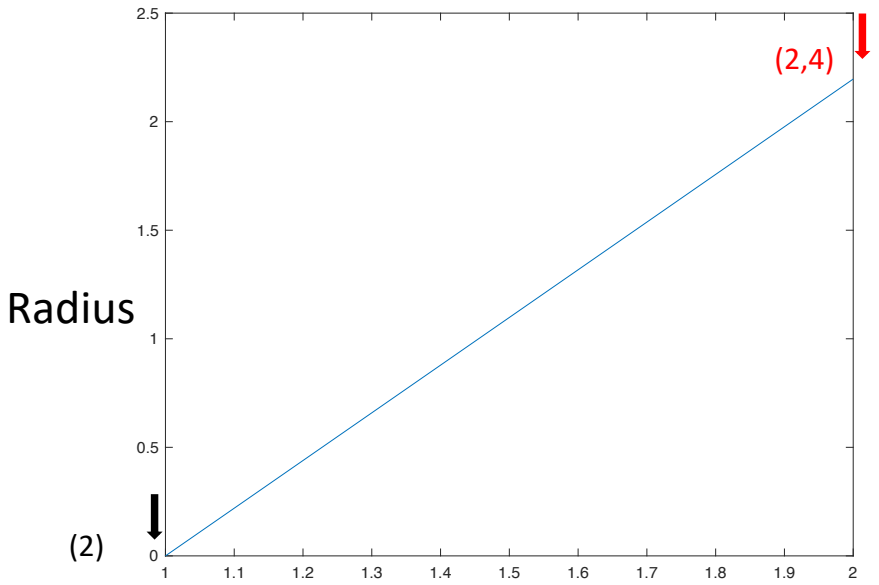
- The radius jump was observed when ts2 was selected.
- Therefore, (ts1,ts3,ts5) is clustered.

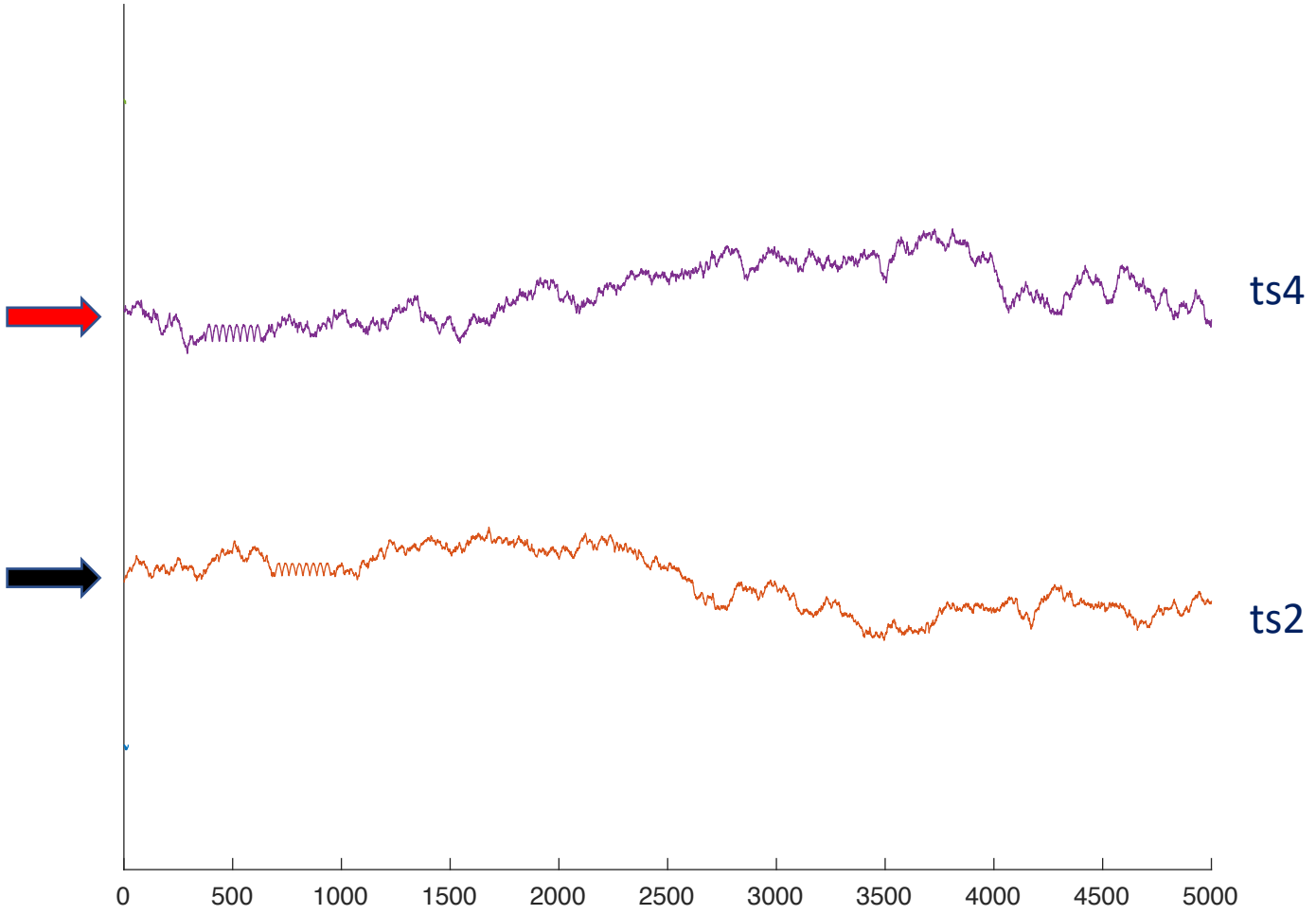




Clustering Algorithm

- The algorithm now runs on the remaining series
- ts2 and ts4 get selected





Pseudocode – consensus_cluster

Function:	consensus_cluster((T1,….Tn), m)
Input:	[T1, T2, T3,…….,Tn]; //matrix of n time series
	m = subsequence_length
Output:	cluster_1: (Ti…Tj),……,cluster_k: (Ta….Tb)

```
1) unselected <- [1,2,3,……,n]; // as initially nothing is selected
2)  while (unselected not empty)
3) [selections, radii] <- find_radii(dataset, unselected)
4) change_index <- isChange(radii) // find the point at which radius increases sharply
5) cluster <- selections[1:change_index]  // pick the selections until drastic radius change
6) removed <- remove cluster from unselected // remove the cluster from unselected
7) count <- increment count; // increase cluster count
8)  end-while
```

Pseudocode - find_radii

```
Function:   consensus_cluster((T1,….Tn), m)
Input:      [T1, T2, T3,…….,Tn];          //matrix of n time series
            [1, 2,…k];                    //IDs of unselected time series
            m = subsequence_length
Output:     selections : [order of selection of time series]
            radii: [r1,r2….,rk]           //radius at each selection
```

```
1.  start_point <- unselected(1);                          //first unselected time series ID
2.  selected <- dataset(start_point);                      //selected time series data
3.  remove start_point from unselected                     //remove the first series ID from unselected
4.  for count <- 1,2….,size(unselected)                    //number of selections possible
5.    new_selection <- true;                               //flag to assign new min_radius
6.    for ts_id <- unselected:                             //ID of time series to try
7.      trial <- dataset(ts_id);                           //pull the time series from dataset
8.      current <- concatenate(trial, selected);           //join the selected and trial time series to form current series with NaN
9.      [sol,~] <- consensus_search. (current,subsequence_len,false); //get radius from consensus motif search
10.     if new_selection:                                  //for every new selection reset the min-radius
11.       [min_sofar, best_selection, new_selection] <- [sol.radius, ts_id, false];
12.     end-if
13.     if min_sofar > sol.radius:                         //set min-radius if new minimum is found
14.       [min_sofar, best_selection] <- [sol.radius, ts_id];
15.     end-if
16.     end-for
17. selections <- append(best_selection);                  //append the ID of time series with minimum radius
18. radii <-  append(min_sofar);                           //append the minimum radius
19. selected <- append(dataset(best_selection));           //append the selected time series to previous selections
20. remove best_selection from unselected                  //delete selected time series from unselected
21. end-for
```

What is the right subsequence length?

- To select a subsequence length, it is useful to have some information on the domain in which data mining is being performed.
- Example: The informal word for “Father” is similar in so many different languages. To find such a pattern in a dataset of speeches in different languages, the length of the word ”papa” could be selected as a subsequence length.

* Bengali: Bābā	* Norwegian : papa
* Mandarin : baba	* Spanish : papá
* Polish : tata	* Swahili : baba
* Swahili : baba	* English : papa
* Turkish : baba	* Hindi : papa
* Xhosa: -tata	* Indonesian : bapa

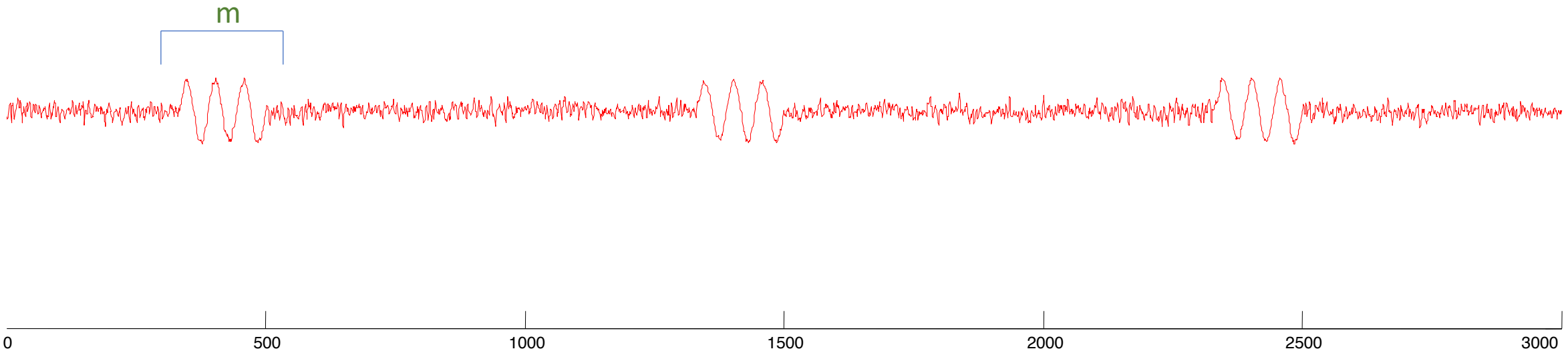
en.wikipedia.org/wiki/Mama_and_papa

Ref: Prof. Keogh’s slides on Matrix Profile: https://www.cs.ucr.edu/%7Eeamonn/matrix_profile_i.pptx

UCR

What is the right subsequence length?

- Another way of selecting a subsequence length is to visually observe a time series plot and look for a repeating structure within.
- The length of that structure could be used as a subsequence length to get meaningful results from the consensus motif algorithm



Ref: Prof. Keogh's slides on Matrix Profile: https://www.cs.ucr.edu/%7Eeamonn/matrix_profile_i.pptx

UCR

Tests on synthetic data: Trace Dataset

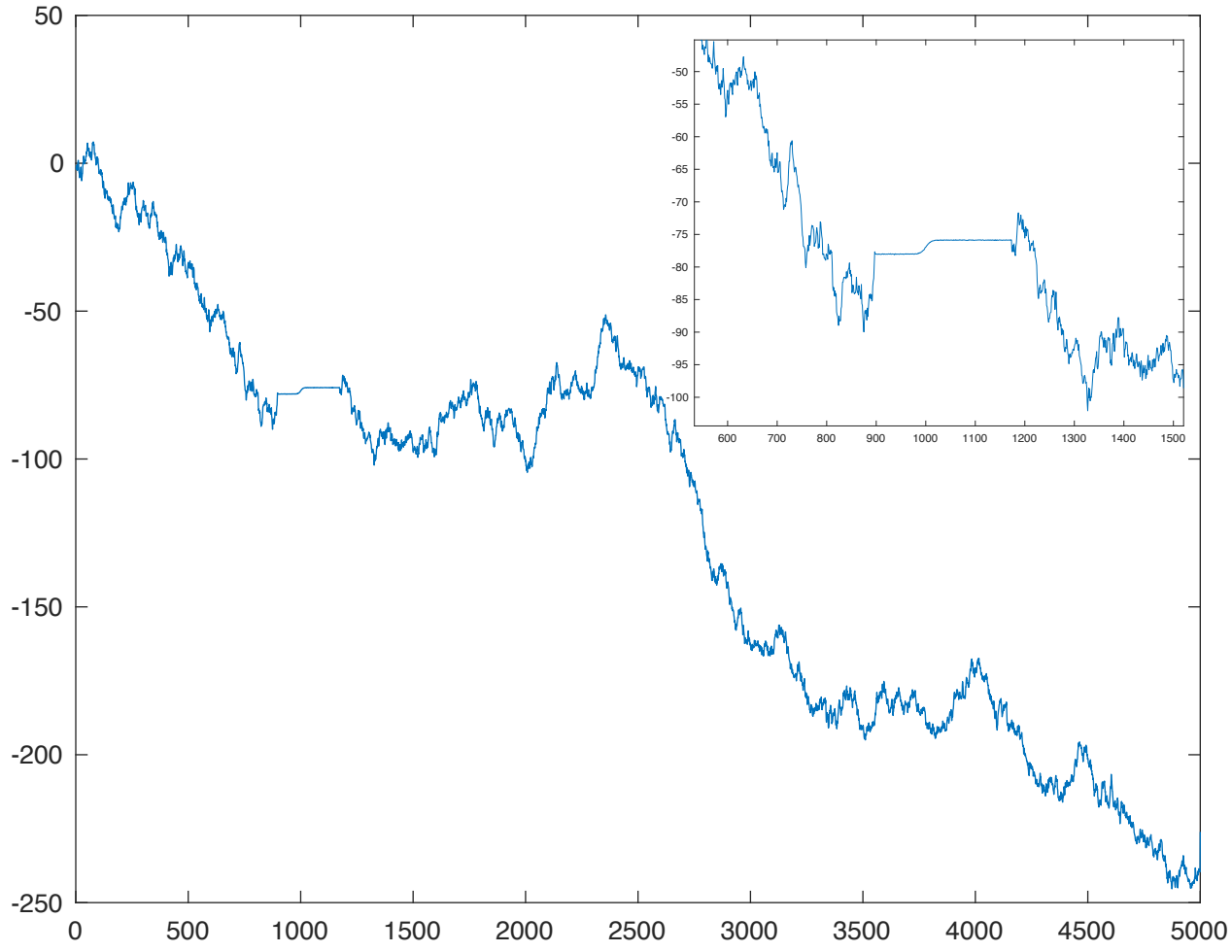
- The Trace dataset is a subset of the Transient Classification Benchmark (Trace project), an initiative to collate data from the application domain of the process industry (e.g. nuclear, chemical, etc.). It is a synthetic dataset designed to simulate instrumentation failures in a nuclear power plant.
- Consists of 50 instances for each of 4 classes.
- Size of each instance = 275

Ref: UCR Time series classification archive

Ref: http://www.timeseriesclassification.com/description.php?Dataset=Trace

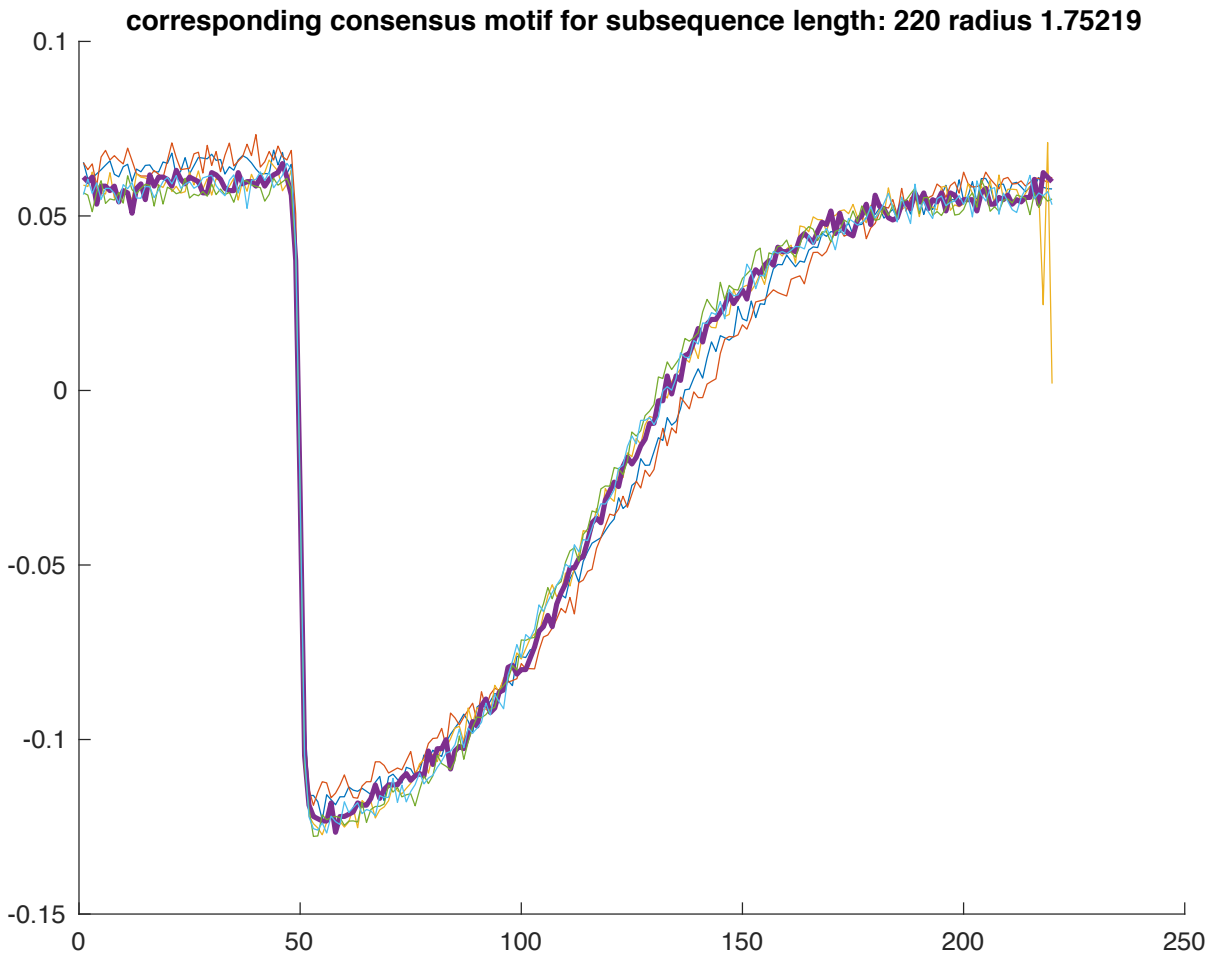
Trace dataset structure

- The data from the Trace dataset is embedded in a random walk of length 5000
- Size of meaningful data = 275
- Class1: 1 2 6 9
- Class 2: 3 4 5 7 8 10

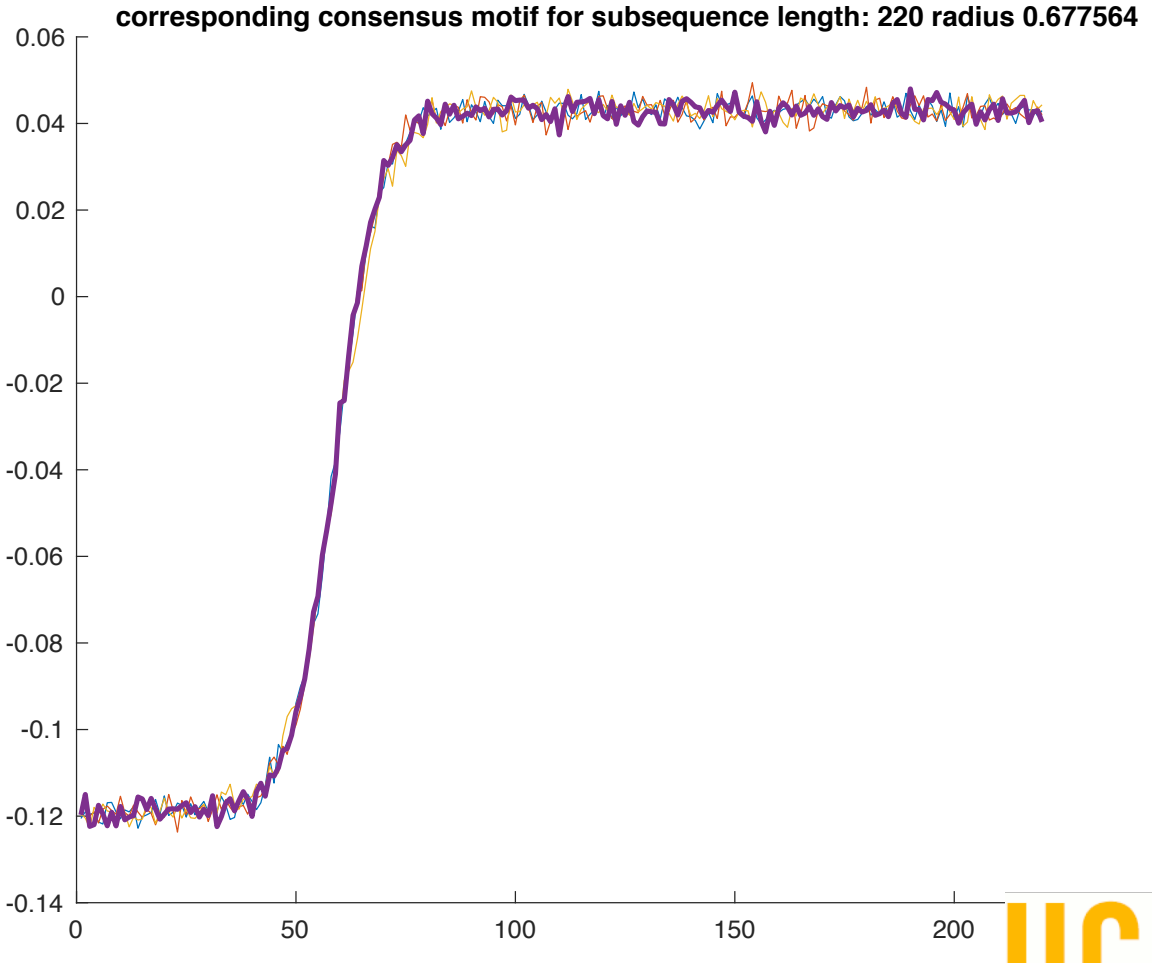


Trace Dataset (Class 1 v Class 2)

Consensus Motif – Class 1

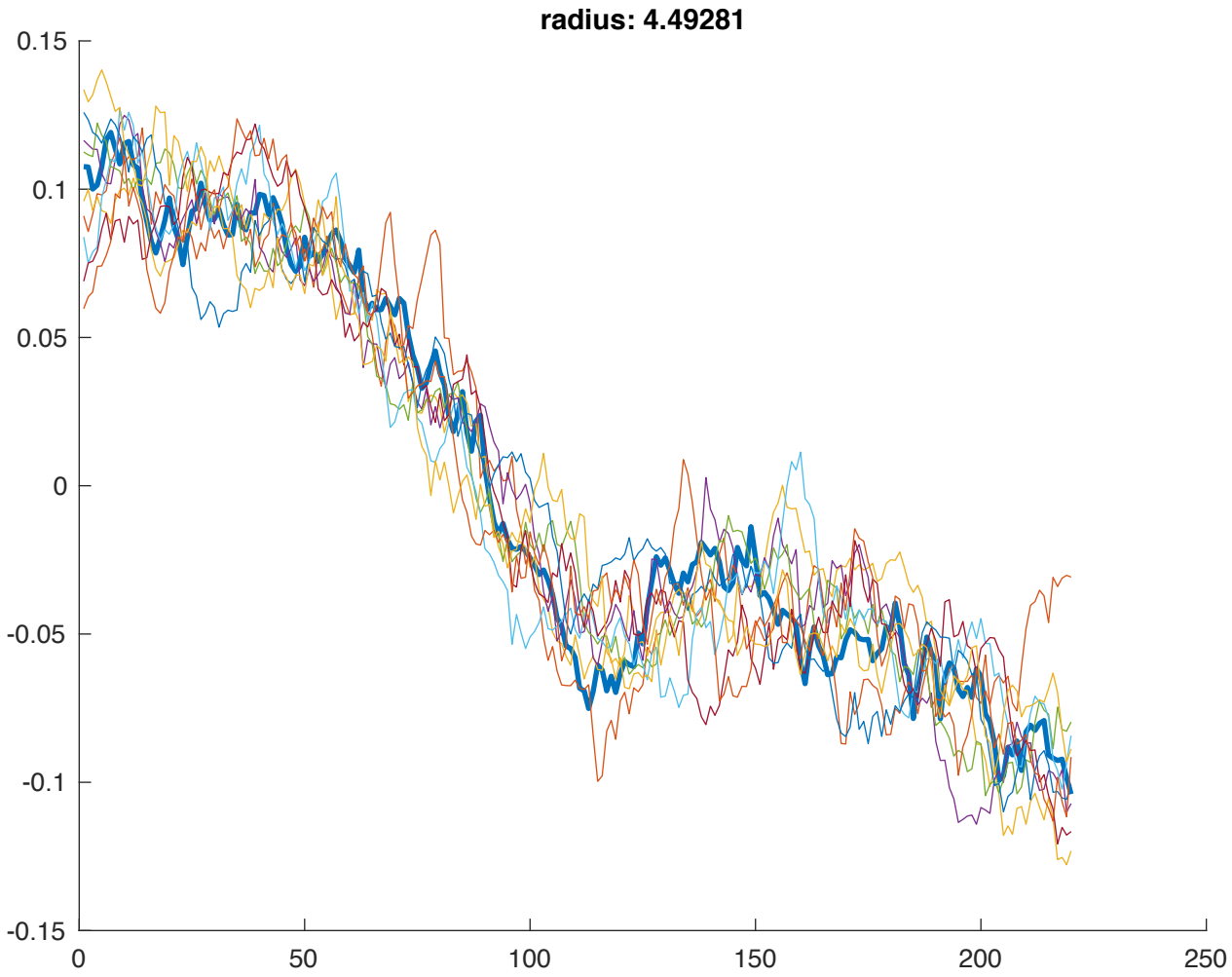


Consensus Motif – Class 2



Trace Dataset

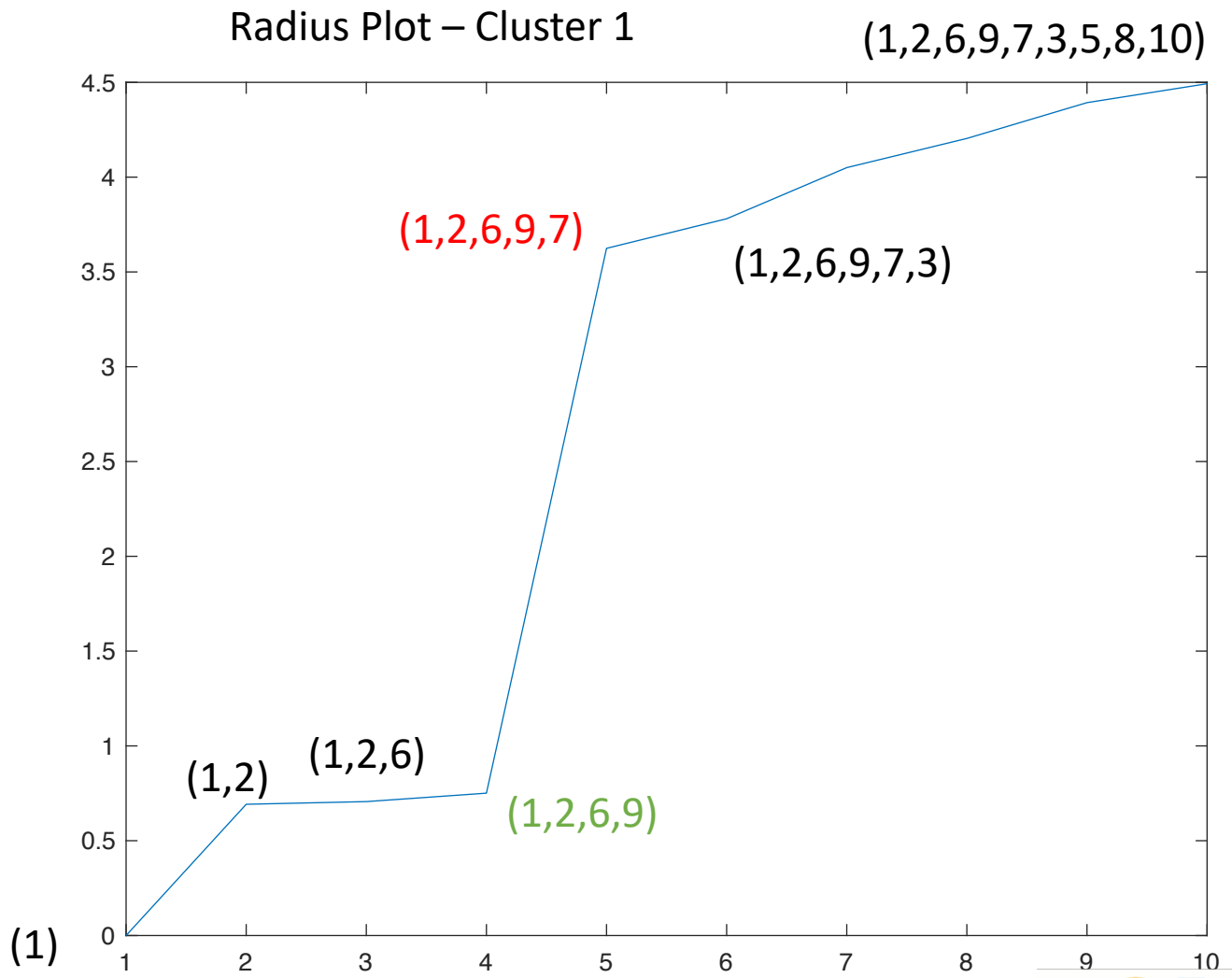
- Class1: 1 2 6 9
- Class 2: 3 4 5 7 8 10
- Subsequence length : 220
- Radius for the entire dataset = 4.49



Consensus Motif for entire dataset

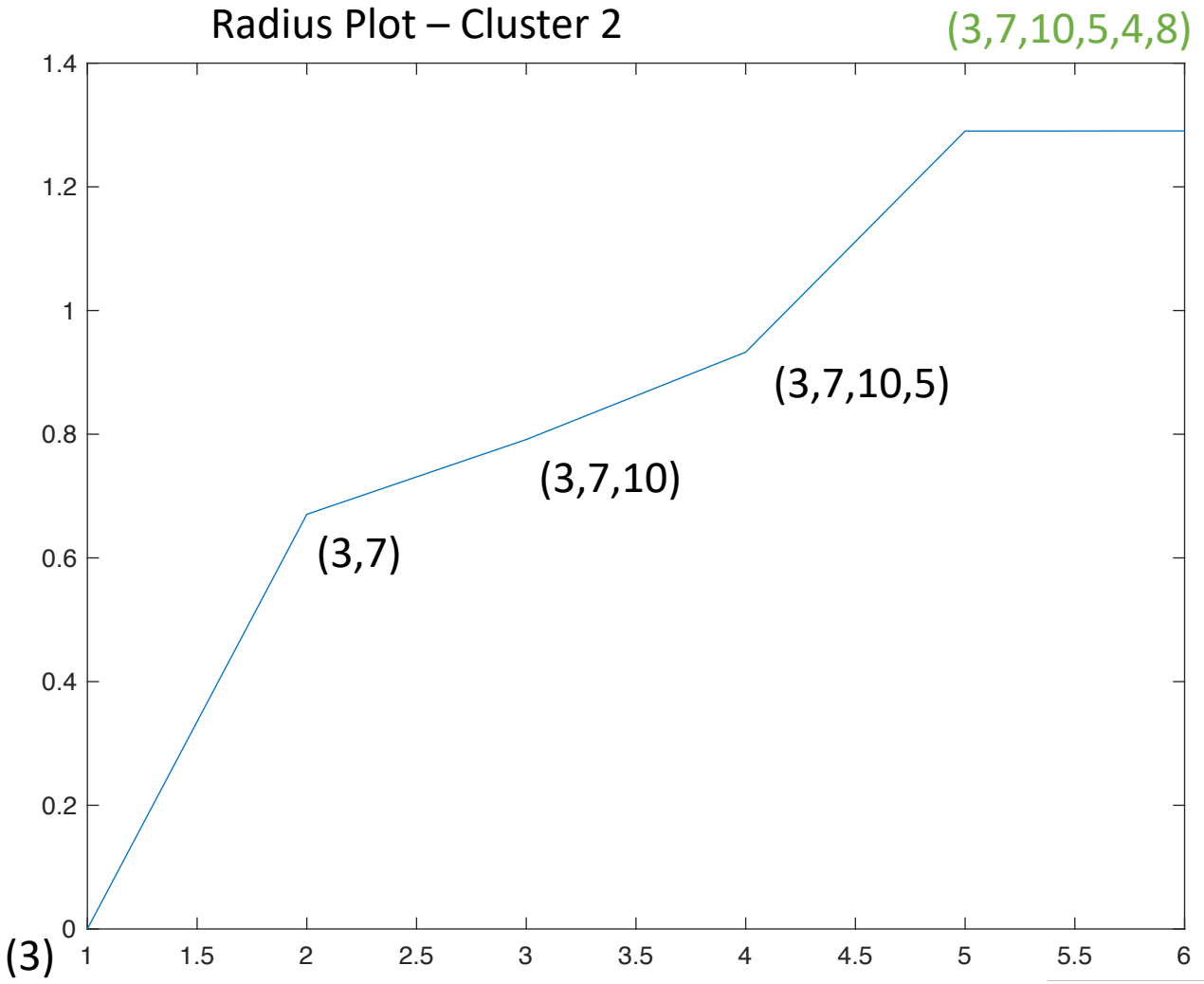
Trace Dataset: Cluster - 1

- Actual Distribution:
  - Class 1: 1 2 6 9
  - Class 2: 3 4 5 7 8 10
- Obtained Result:
  - Cluster 1: 1 2 6 9
  - Unselected : 3 4 5 7 8 10



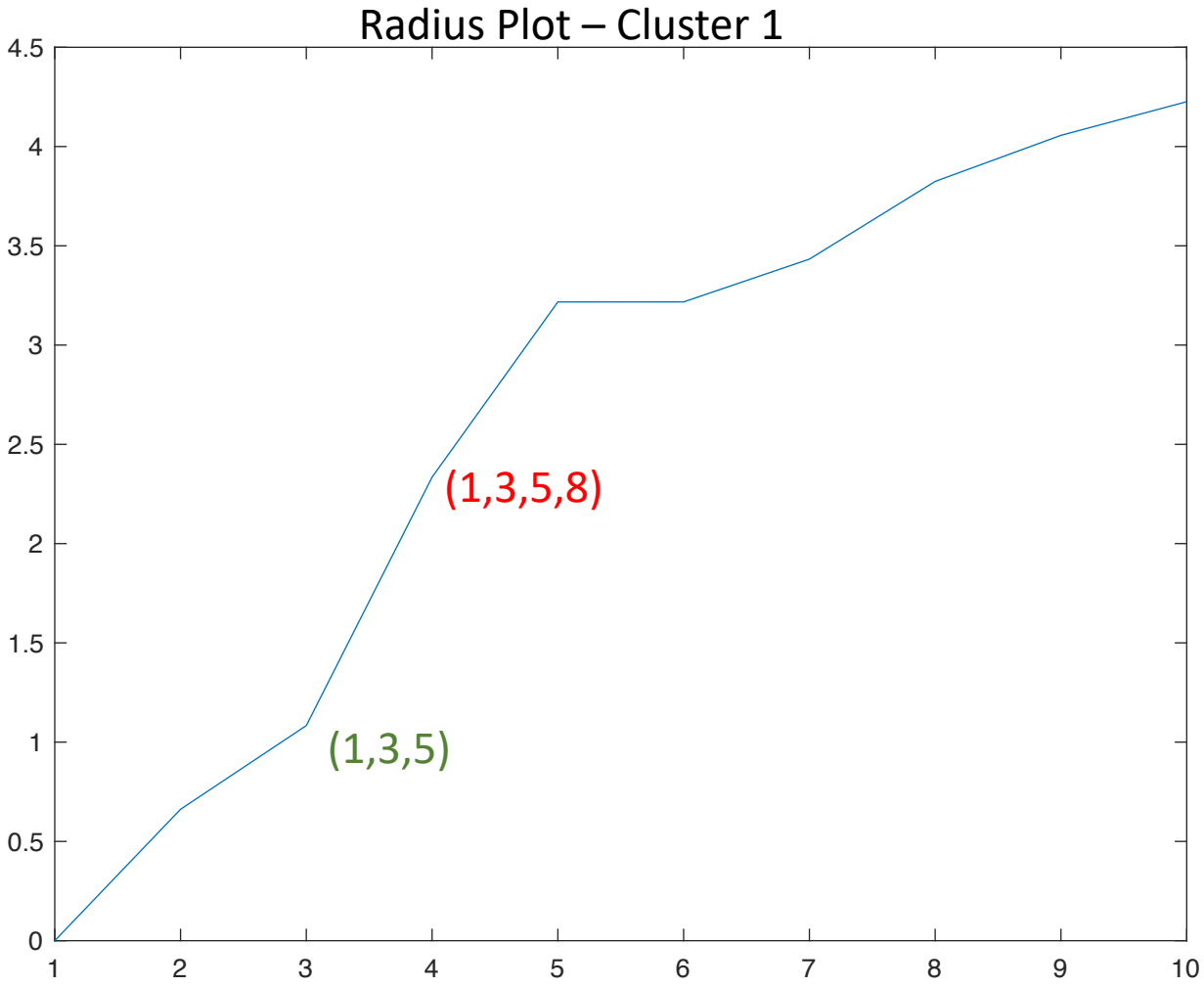
Trace Dataset: Cluster - 2

- Actual Distribution:
  - Class 1: 1 2 6 9
  - Class 2: 3 4 5 7 8 10
- Obtained Result:
  - Cluster 1: 1 2 6 9
  - Cluster 2: 3 7 10 5 4 8
- Accuracy : 100%



Trace dataset: 3 classes

- A new class is added to the dataset.
  - Class 1: 1 3 5 8
  - Class 2: 2 4 6
  - Class 3: 7 9 10
- Clusters Found:
  - Cluster 1: 1 3 5
  - Cluster 2: 2 4 6
  - Cluster 3: 7 9 10
  - Cluster 4: 8
- Accuracy: 90%
- Though 8 is selected after 5, the radius difference causes the clustering to stop at (1,3,5)



Experiments on Real Data

Bird songs

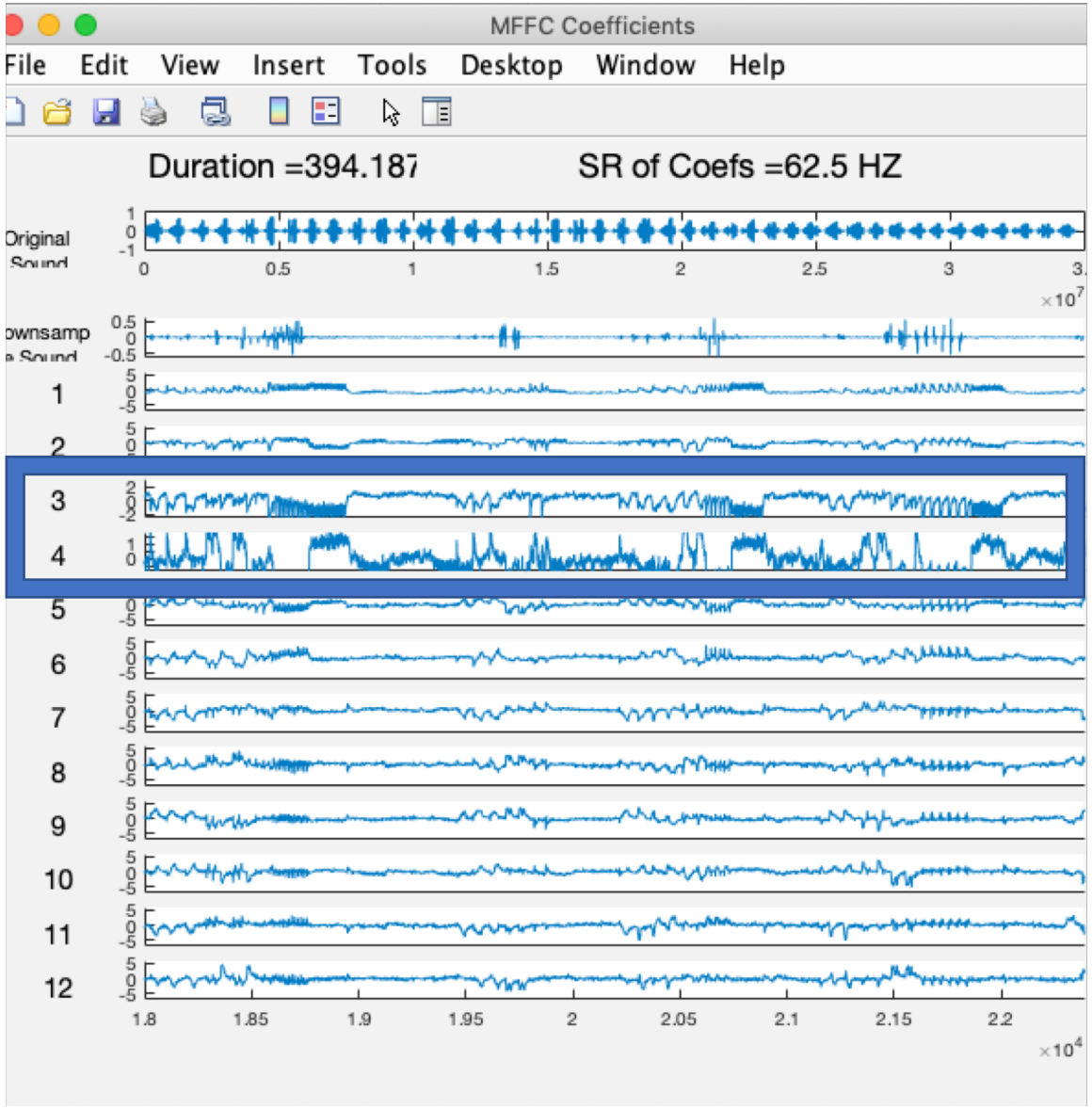
XC134265 - Thrush Nightingale - Luscinia luscinia
Song duration: 06m:34s

XC411608 - Common Nightingale - Luscinia megarhynchos
Song duration: 05m:22s

Ref: http://xeno-canto.org

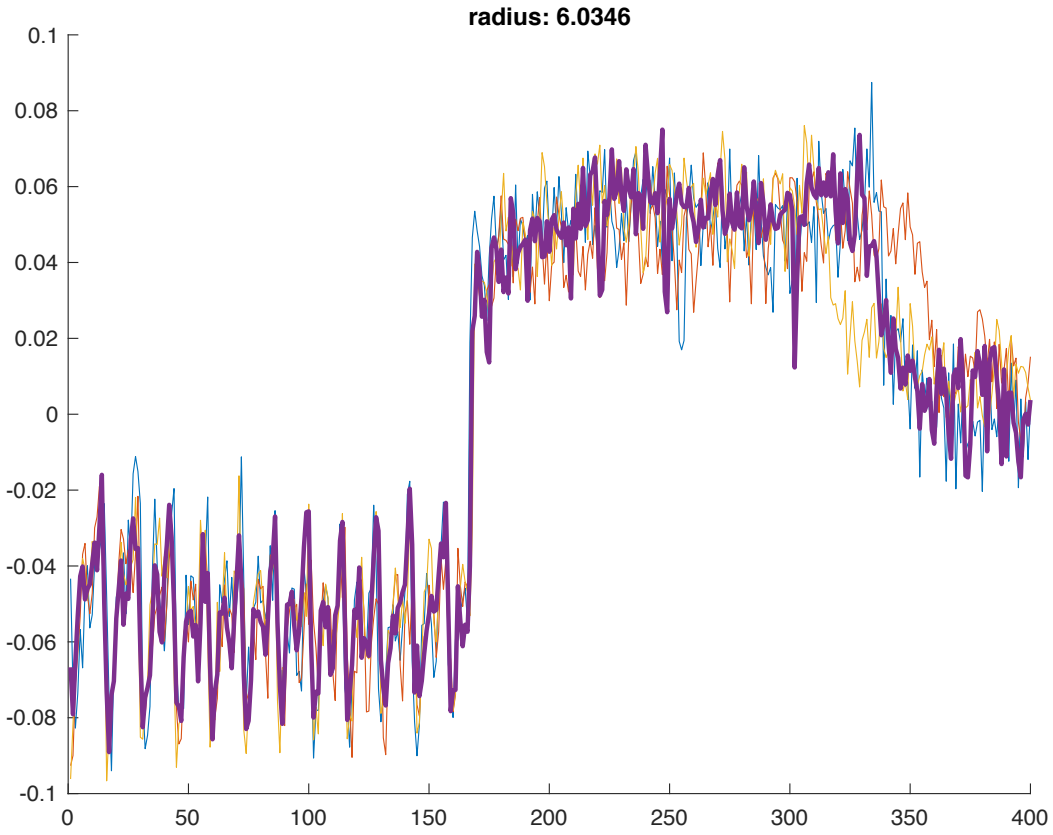
Dataset Creation

- Each of the 12 MFCC time series is of size 49244 (06m:34s).
- The MFCC time series **4** was cut into 4 equal parts and cropped to 10000 (80s) each.
- Now we have four time series representing one class. i.e. The Thrush Nightingale
- Repeat the process for the **common nightingale.**
- Dataset: 8 time series of length 10000(80s), 4 of each class.

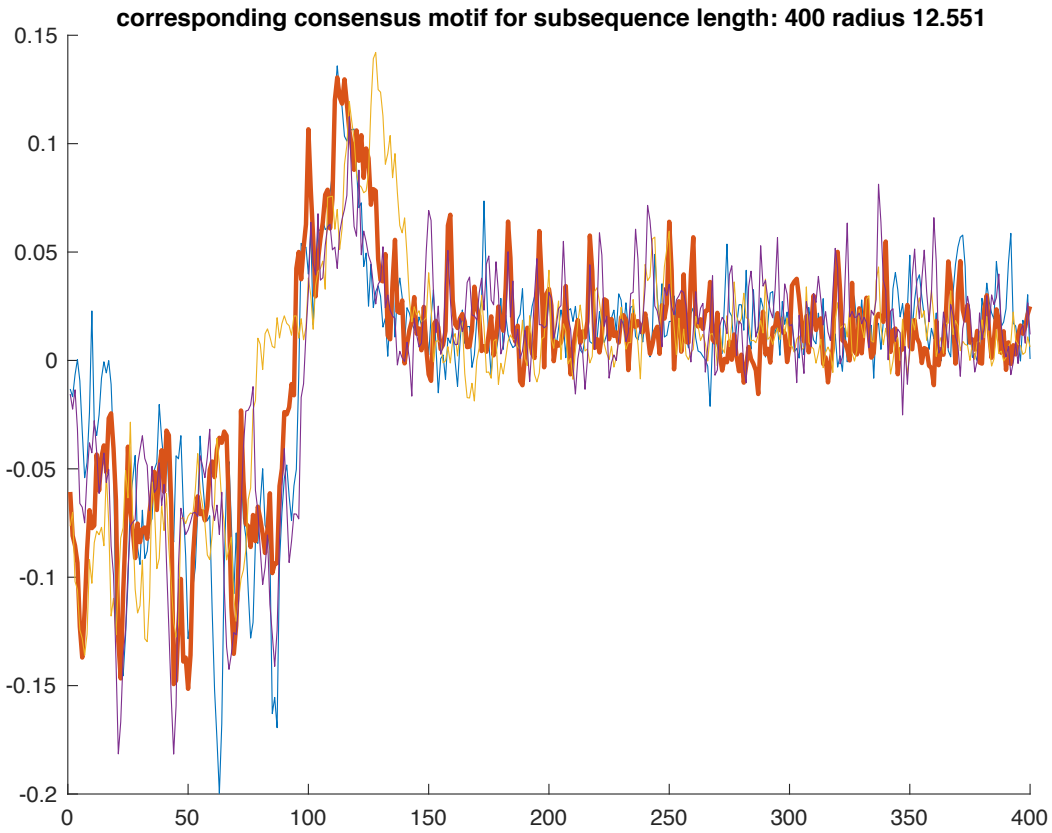


Consensus Motifs (Thrush N. vs Common N.)
Subsequence Length: 400 (3.2s)

Thrush N.

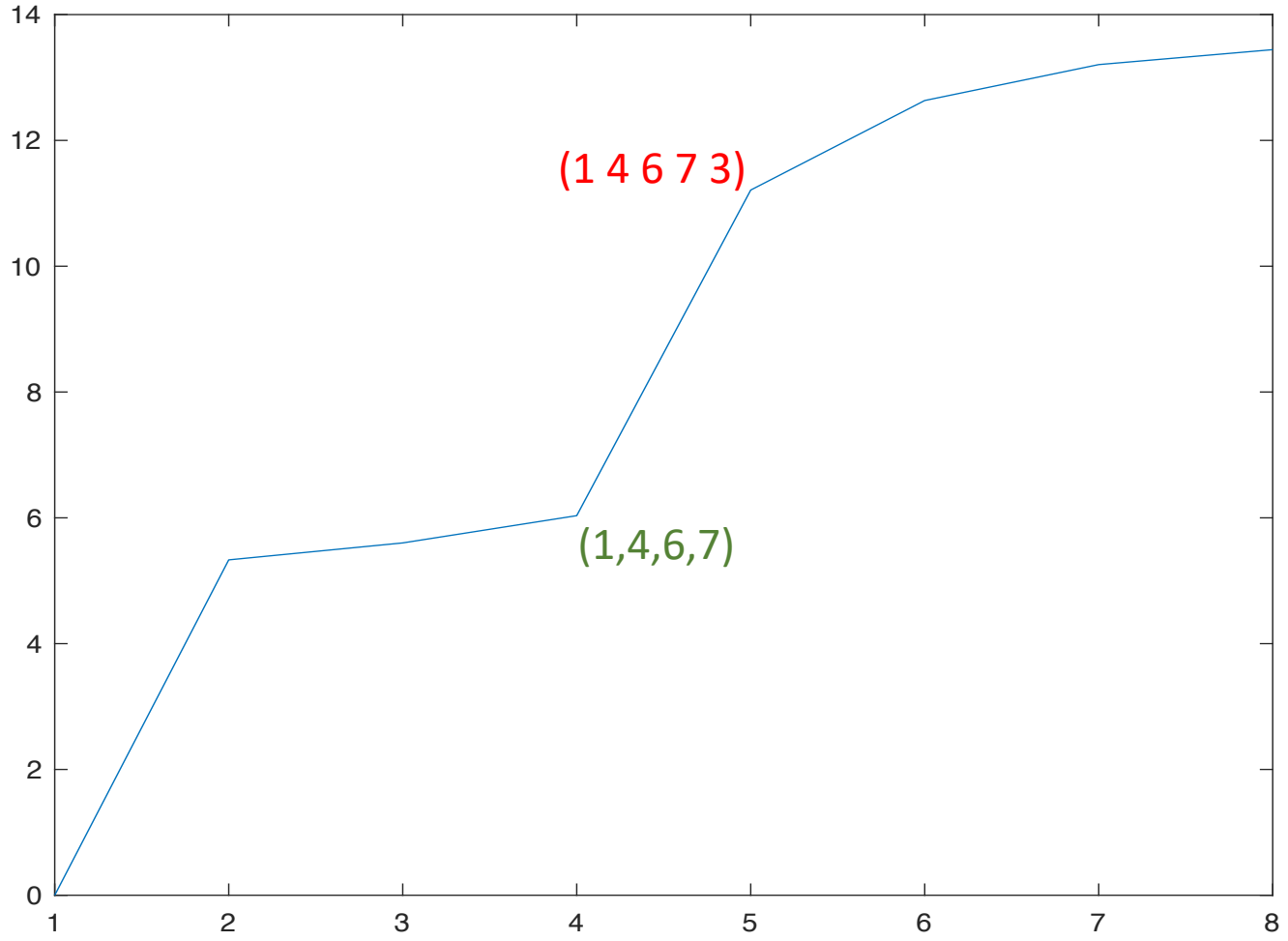


Common N.



Results: Thrush N. and Common N.

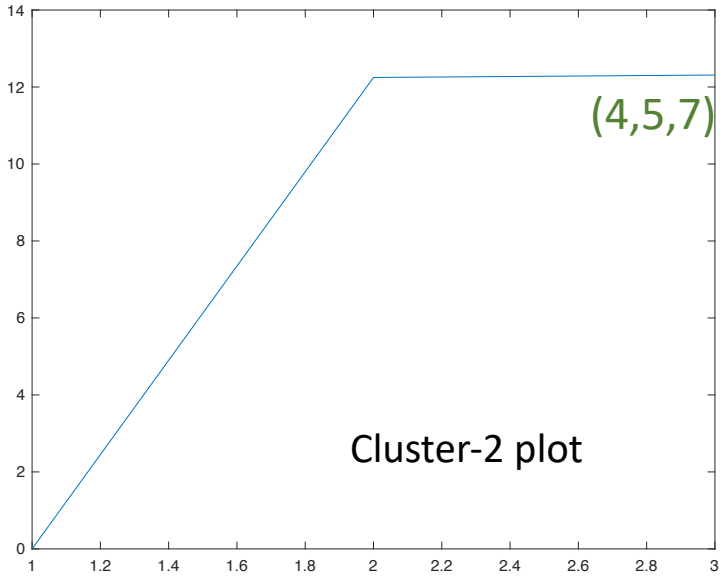
- Actual Data:
  - Thrush Nightingale: 1 4 6 7
  - Common N. : 2 3 5 8
- Thrush Nightingale clustered accurately.
- One outlier in common nightingale.
- Clusters Obtained:
  - Cluster 1: 1 4 6 7 (Thrush N.)
  - Cluster 2: 2 3 5 (common N.)
  - Cluster 3: 8 (common N.)
- Accuracy : 80%

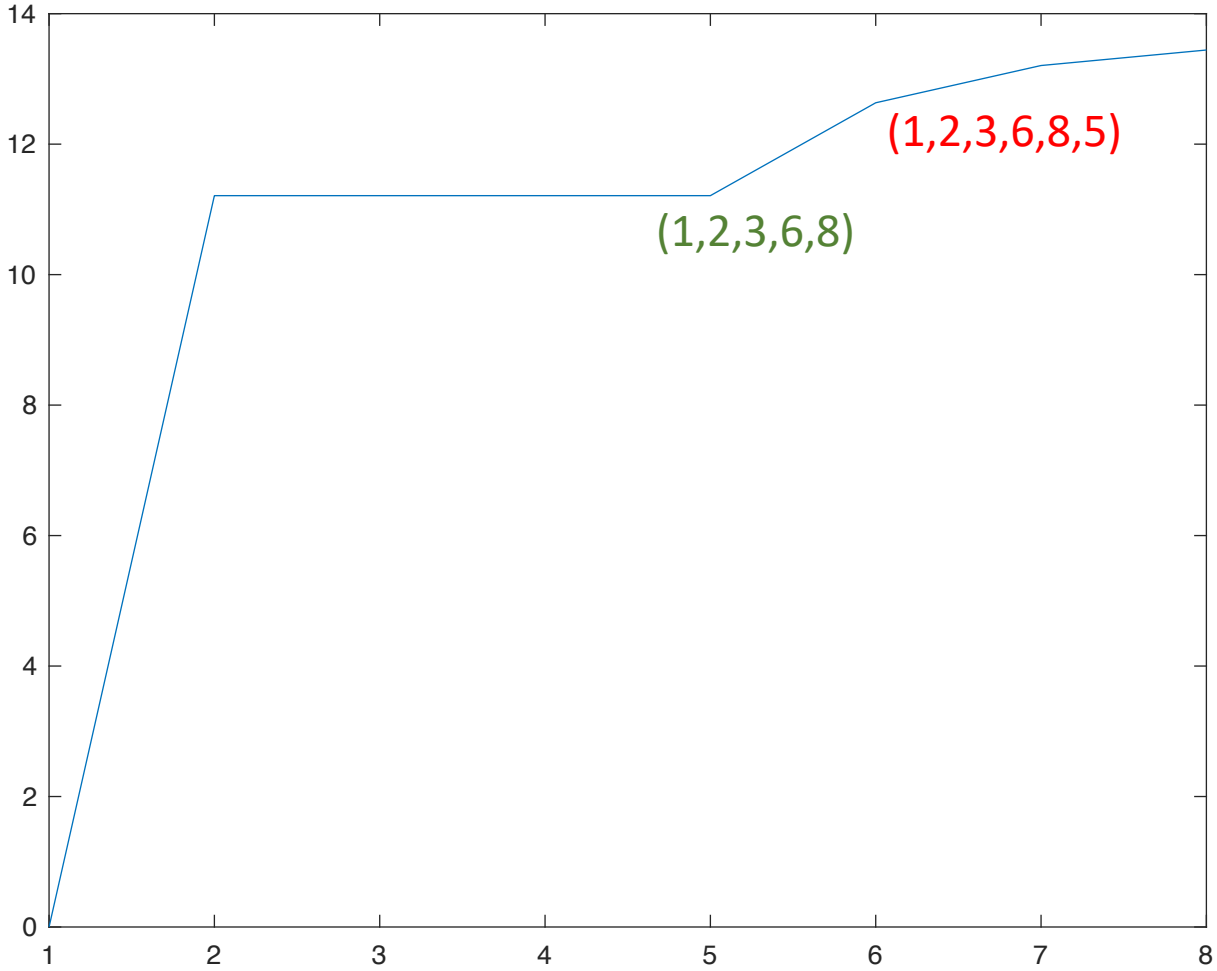


Radius Plot (X-axis: Selections, Y-axis: radius)

Dataset is Shuffled

- Actual Data:
  - Common Nightingale : 1 4 5 7
  - Thrush Nightingale: 2 3 6 8
- Clusters Obtained:
  - Cluster 1: 1 2 3 6 8(Thrush N.)
  - Cluster 2: 4 5 7 (common N.)
- Accuracy : 80%

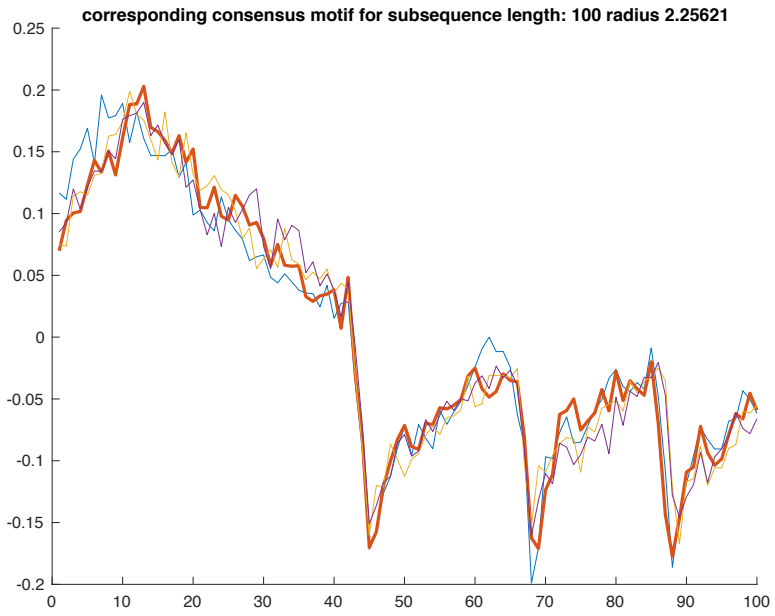




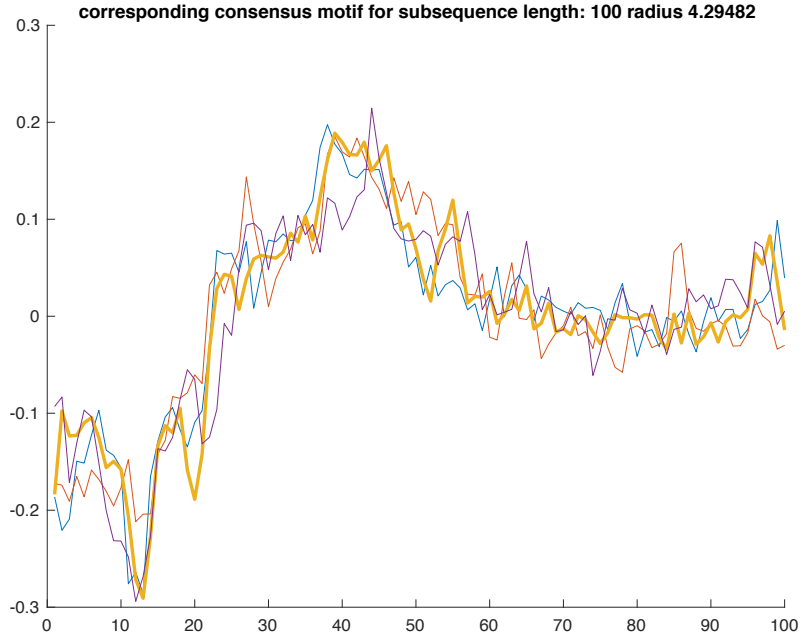
Radius Plot (X-axis: Selections, Y-axis: radius)

Subsequence Length Lowered

Thrush Nightingale



Common Nightingale



- The consensus motif for Common Nightingale is still noisy (radius is high)
- Subsequence length used is 100 (0.8s)
- Clusters Obtained:
  - Cluster 1: 1 2 3 6 8(Thrush N.)
  - Cluster 2: 4 5 7 (common N.)

Actual Data:

Common Nightingale : 1 4 5 7

Thrush Nightingale: 2 3 6 8

Bird Songs

XC444371 - Thrush Nightingale - Luscinia luscinia
Song duration: 06m:44s

CLASS 1

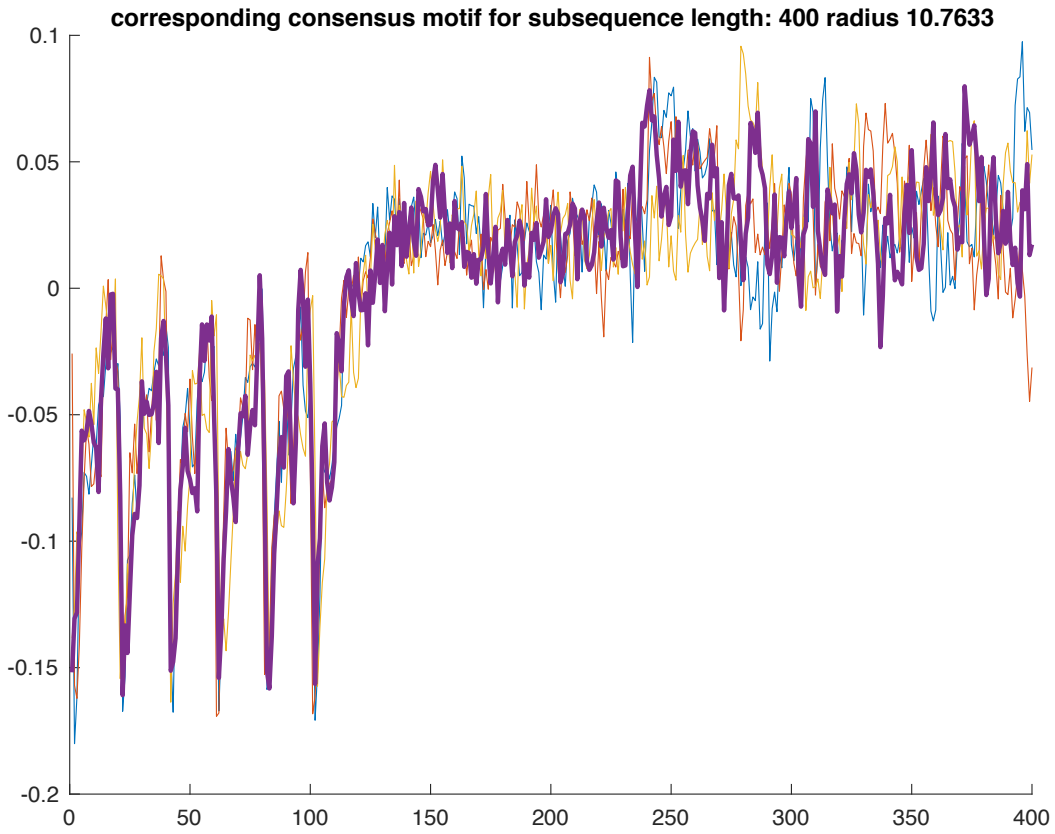
XC134265 - Thrush Nightingale - Luscinia luscinia
Song duration: 06m:34s

CLASS 2

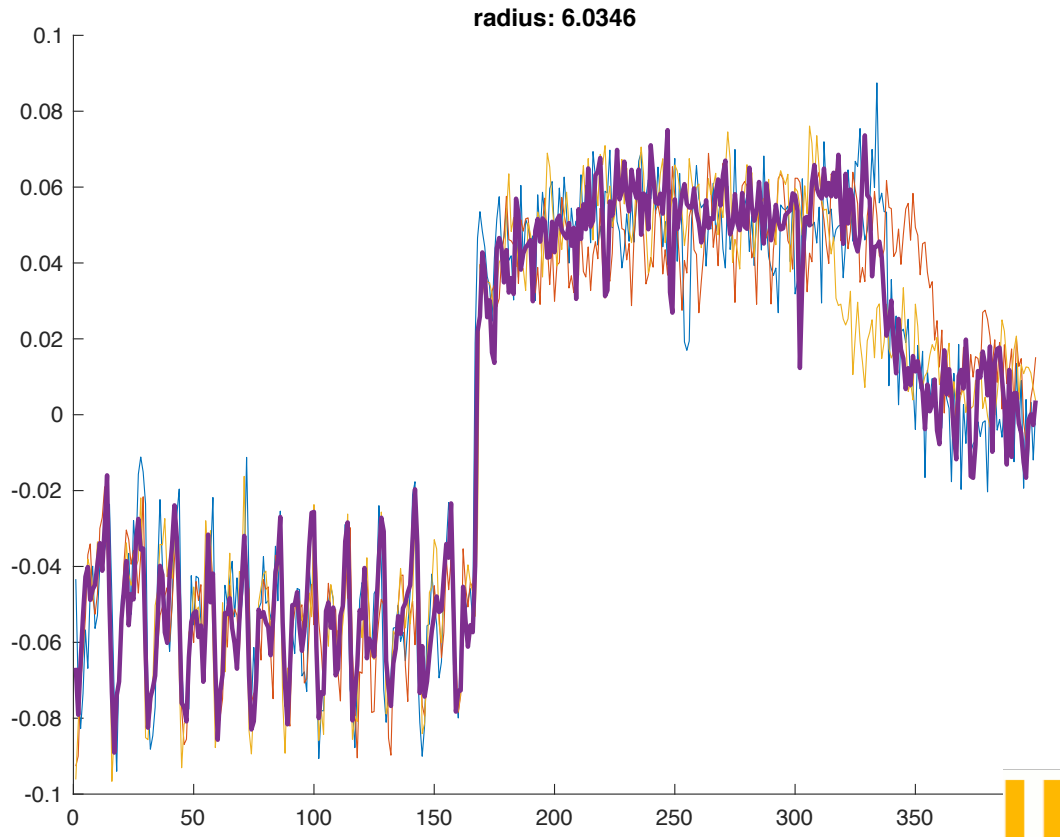
Ref: http://xeno-canto.org

Motifs (Class1 vs. Class 2) both Thrush Nightingale

Class1 (4 time series of length 80s, subsequence length: 3.2s



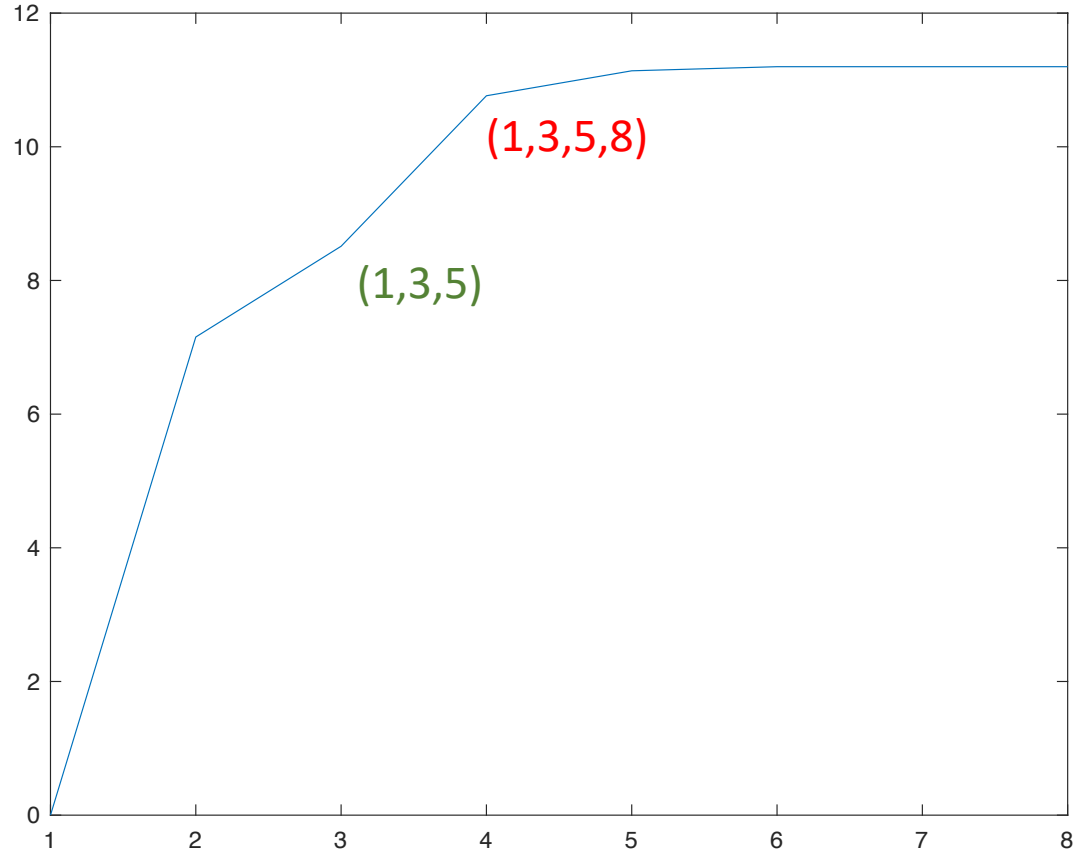
Class2 (4 time series of length 80s, subsequence length: 3.2s



Results - Clustering class 1

- Actual Data:
  - Class 1: 1 3 5 8
  - Class 2: 2 4 6 7
- Results obtained:
  - Cluster 1: 1 3 5

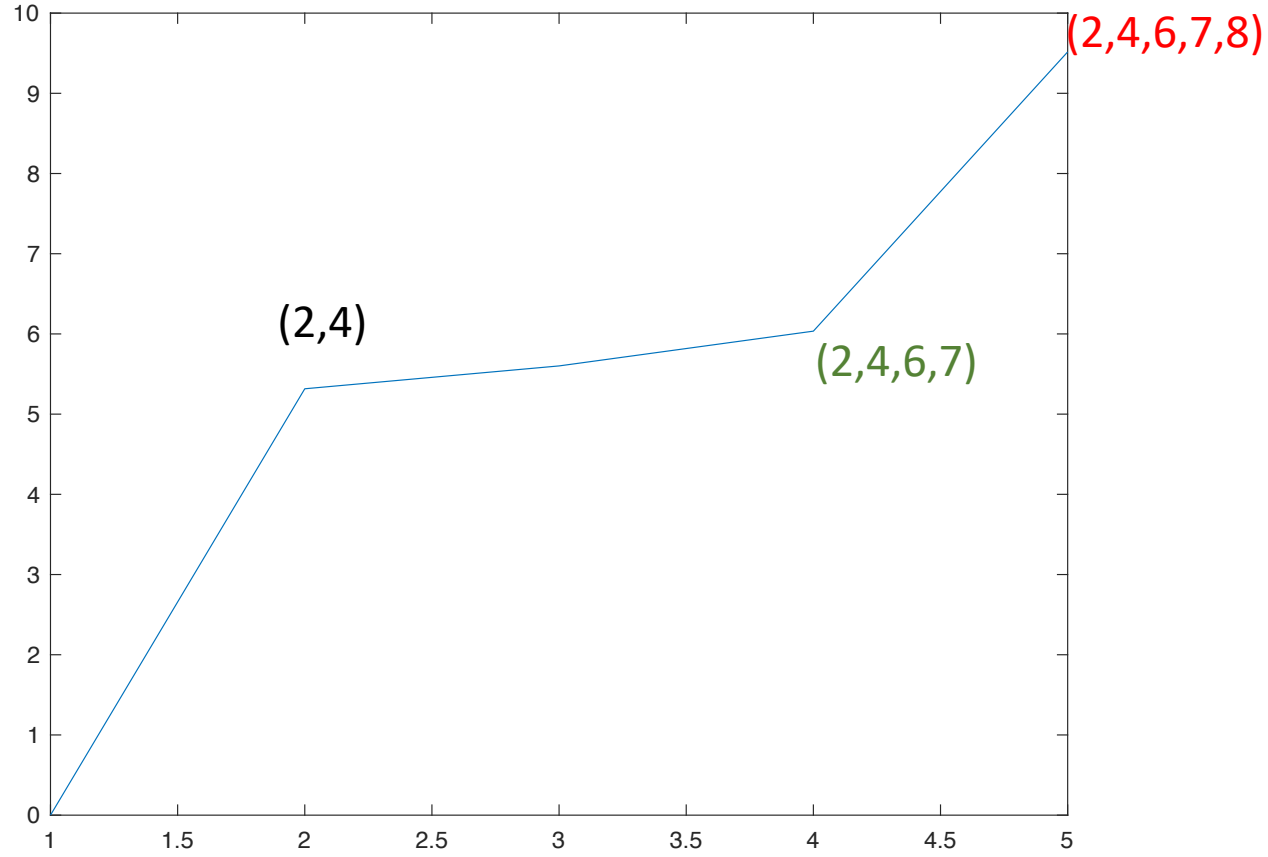
Radius Plot (X-axis: Selections, Y-axis: radius)



Results – Clustering class 2

- Actual Data:
  - Class 1: 1 3 5 8
  - Class 2: 2 4 6 7
- Results obtained:
  - Cluster 1: 1 3 5
  - Cluster 2: 2 4 6 7
  - Cluster 3: 8 (Obtained in 3rd iteration)

Radius Plot (X-axis: Selections, Y-axis: radius)



Conclusions

- Given a set of **n** time series(of **k** classes), the algorithm forms **c** clusters using the radius obtained from the consensus motif search.
- A forward selection approach is followed to cluster the data.
- The algorithm is **not order-invariant**.
  - Displays order invariant behavior for some types of datasets.
- The algorithm is **deterministic**.
  - Using the same input, multiple runs of the program returned the same output.

Future Work

- Upgrade the algorithm to accept time series with varying sizes
- Explore optimization (pruning)
- Explore other methods of clustering like Backward Elimination
- Develop an algorithm that can train the clustering tool to improve accuracy