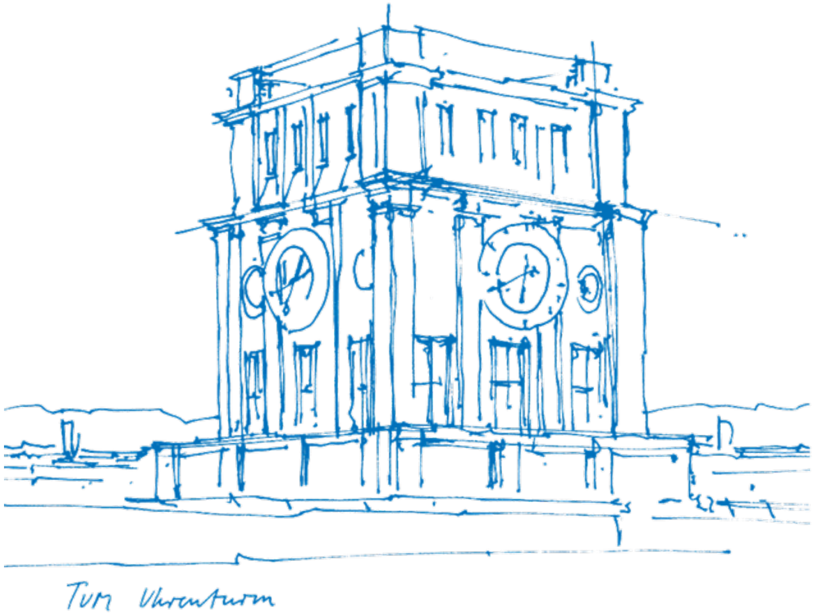
On-The-Fly pattern recognition for satellite time series data

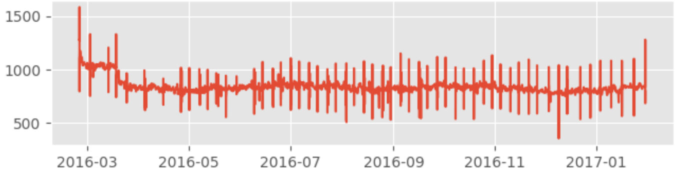
Lucas Lincoln | Markus Steinbach | Lukas Dreier

Munich, 6th August 2019

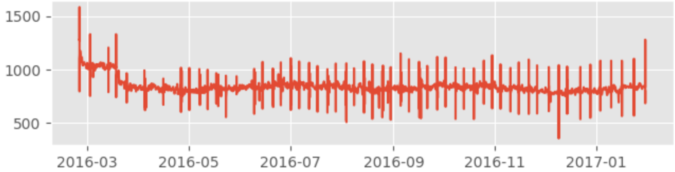


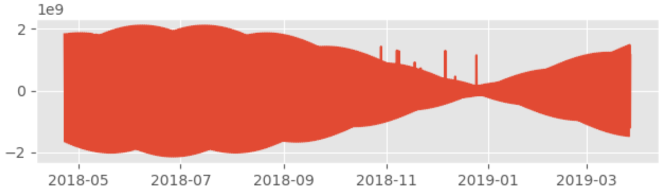
Goal of the project

Goal of the project

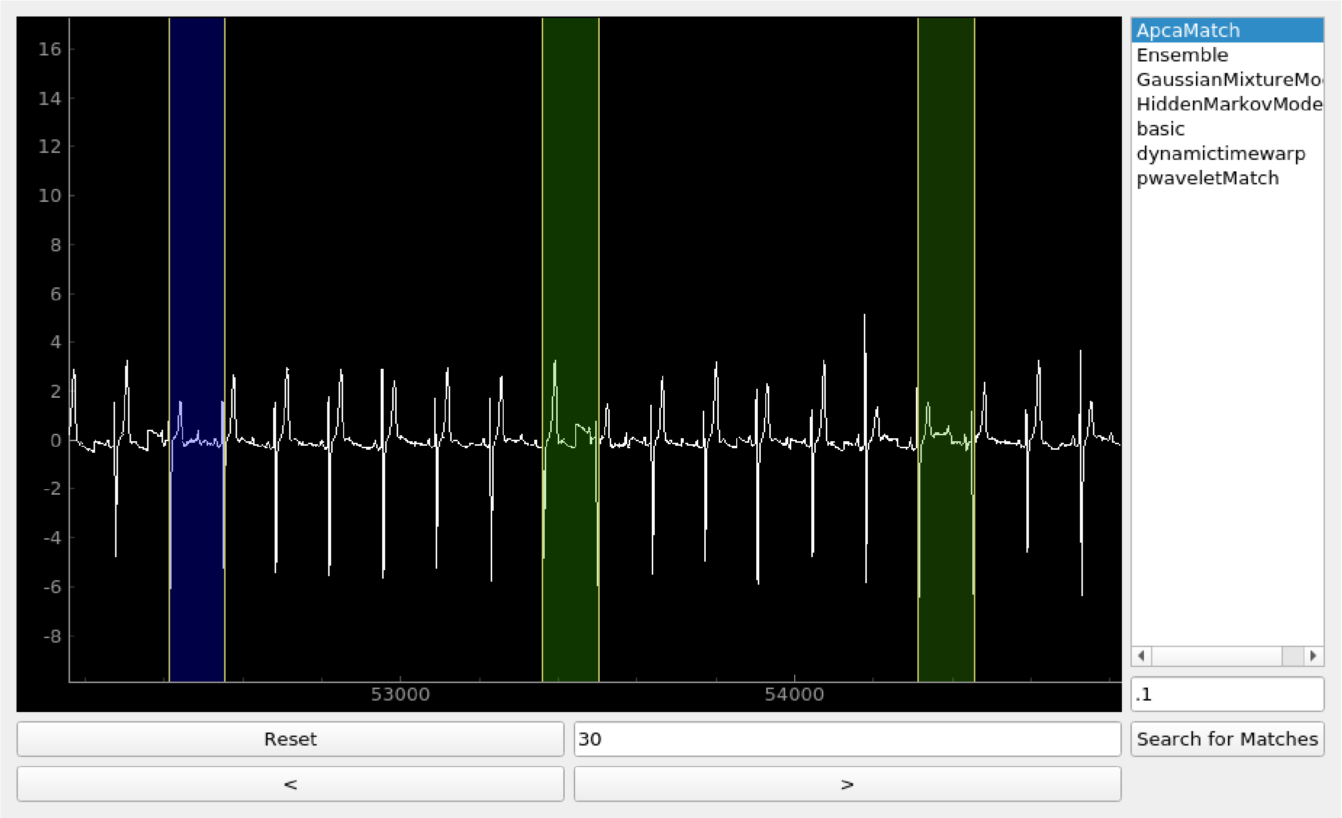


Goal of the project



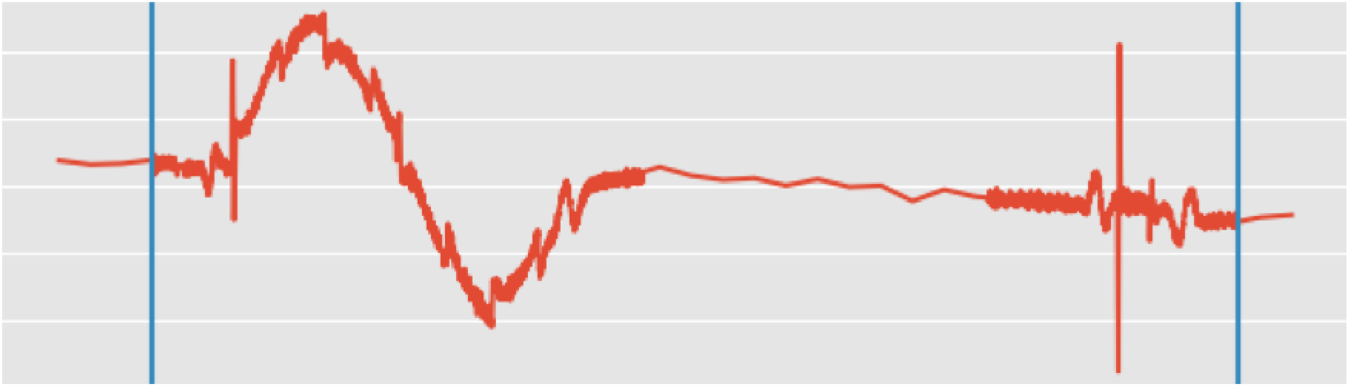


Demo of Matching Tool



Key Challenges

1. **Inconsistent Sampling**
2. Instantaneous events in otherwise predictable signals
3. Low Pattern-to-noise ratio
4. Discrete or Continuous signals

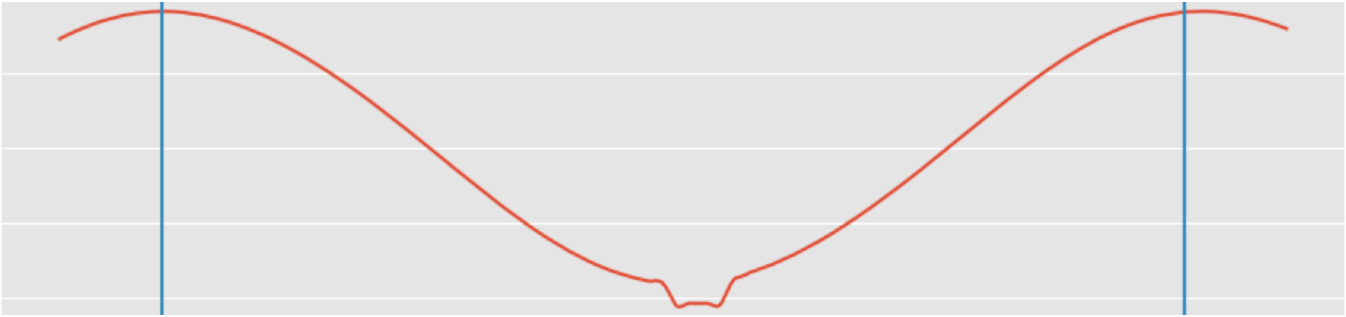


Key Challenges

1. Inconsistent Sampling
2. **Instantaneous events in otherwise predictable signals**
3. Low Pattern-to-noise ratio
4. Discrete or Continuous signals

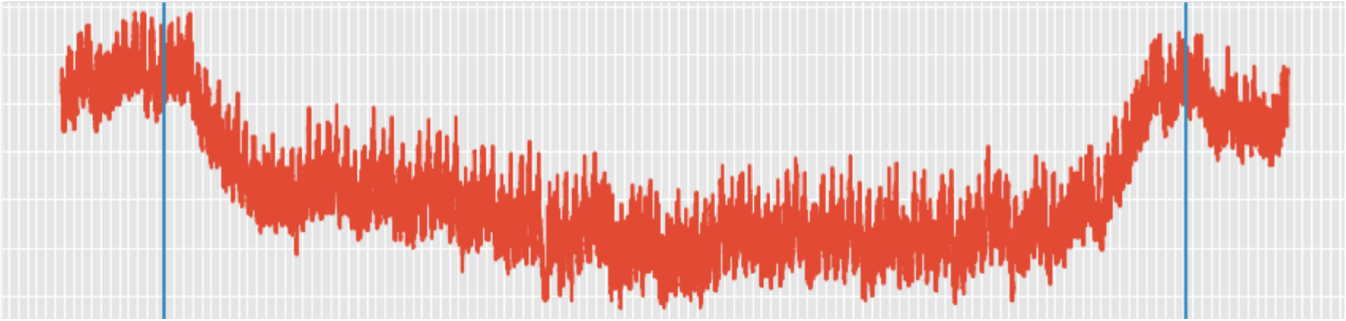
Key Challenges

1. Inconsistent Sampling
2. **Instantaneous events in otherwise predictable signals**
3. Low Pattern-to-noise ratio
4. Discrete or Continuous signals



Key Challenges

1. Inconsistent Sampling
2. Instantaneous events in otherwise predictable signals
3. **Low Pattern-to-noise ratio**
4. Discrete or Continuous signals

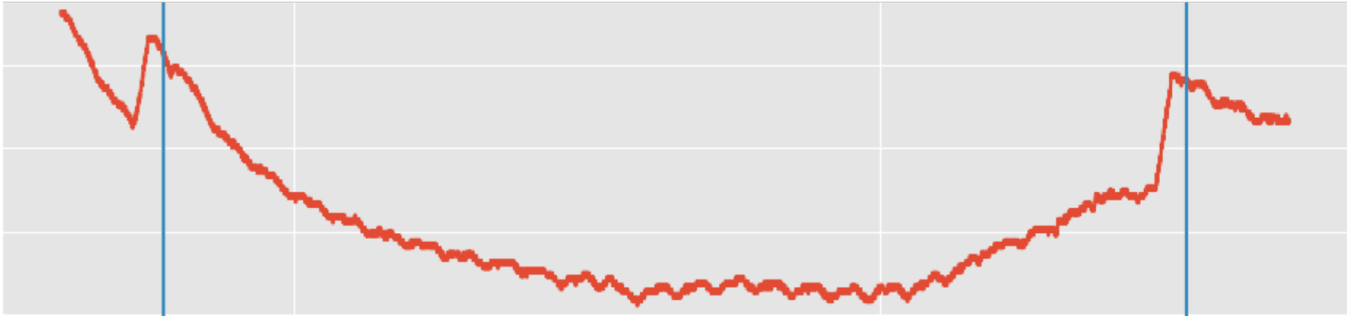


Key Challenges

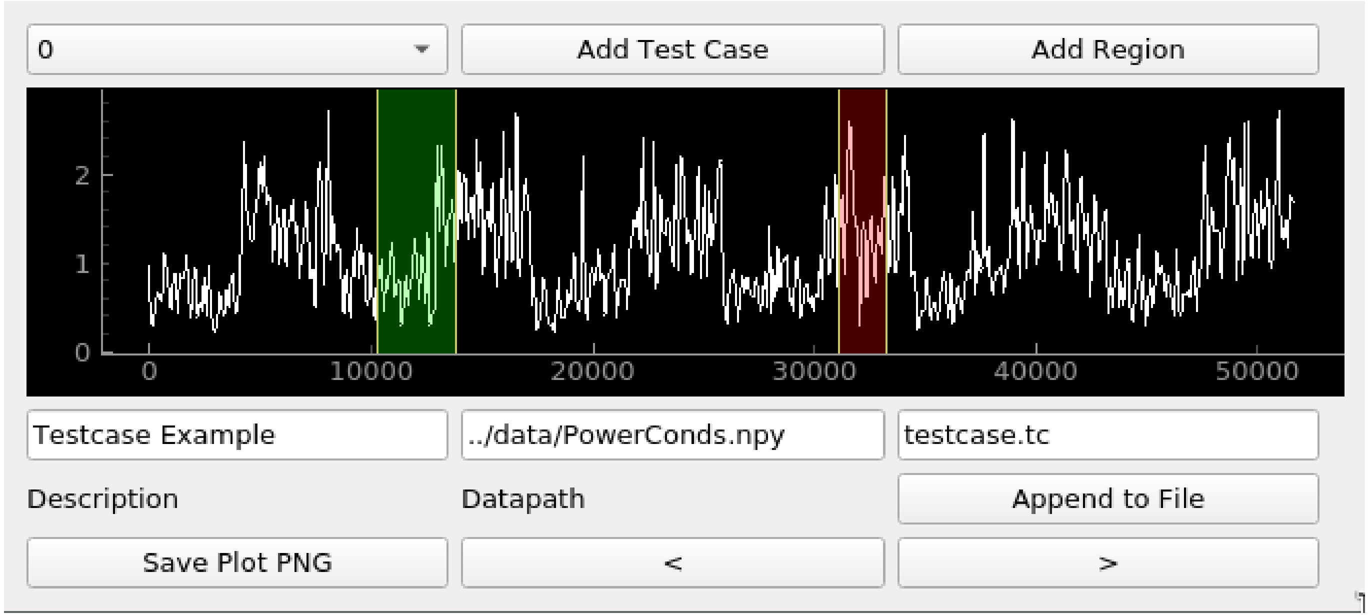
1. Inconsistent Sampling
2. Instantaneous events in otherwise predictable signals
3. Low Pattern-to-noise ratio
4. **Discrete or Continuous signals**

Key Challenges

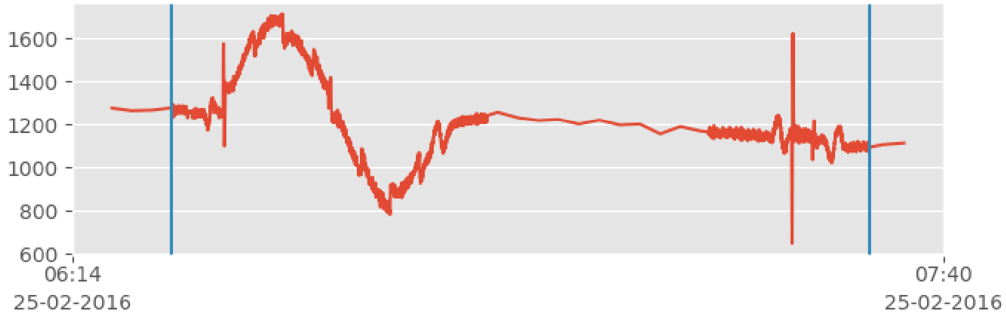
1. Inconsistent Sampling
2. Instantaneous events in otherwise predictable signals
3. Low Pattern-to-noise ratio
4. **Discrete or Continuous signals**

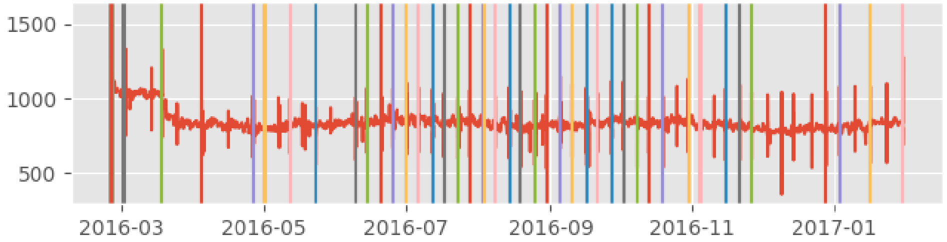


Labeling tool



Testcase Overview





Occurences	42
Duration (m)	68.8
Average match duration (m)	84.5
Discrete/continuous	continuous
StdDv during match	205.5

Notes
Sinus wave and another pattern; In between a period with lower sampling

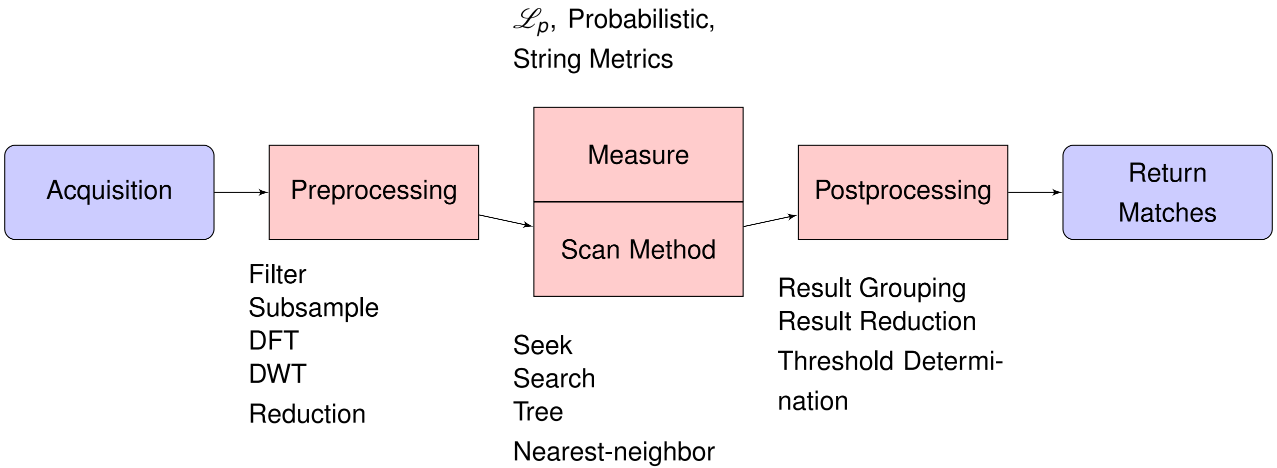
UCR Data

- The great time series classification bake off - Bagnall, A., Lines, J., Bostrom, A. et al. Data Min Knowl Disc (2017) 31: 606.
- 128 (labeled) time series datasets
- Created a routine to convert to our testcase format
- Allows us to use external (eg: LRZ) compute resourses to perform iterative studies

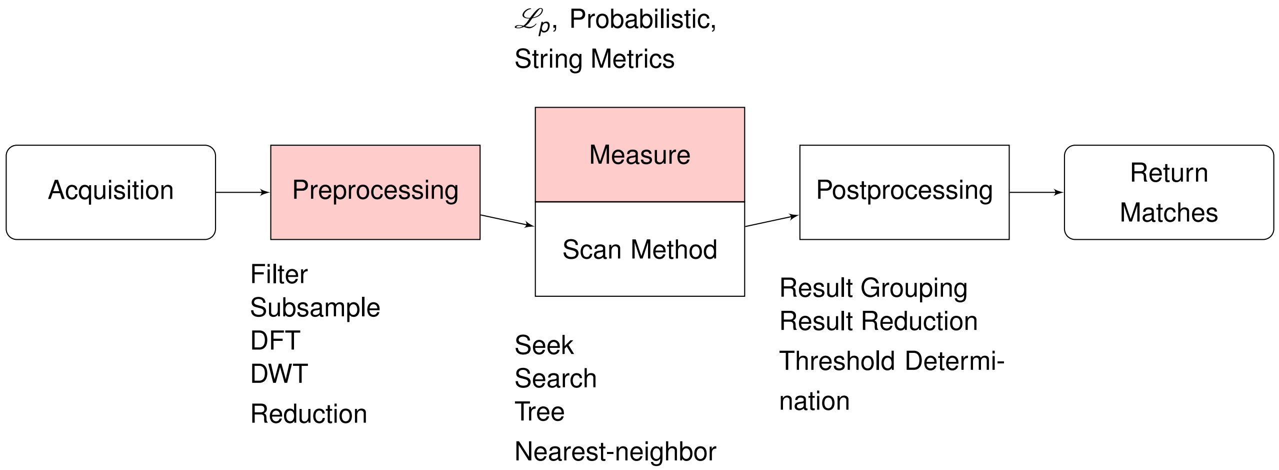
Datasets currently used:

- **ECG200**
- **ECGFiveDays**
- **FordA**
- **OliveOil**
- **PowerConds**
- **Computers**

Basic Structure of Time Series Matching



Basic Structure of Time Series Matching



Algorithms - Overview

- Characterization mainly depends on the preprocessing and the measure part of time series matching
- Deep learning approaches should not be taken into account

Distance

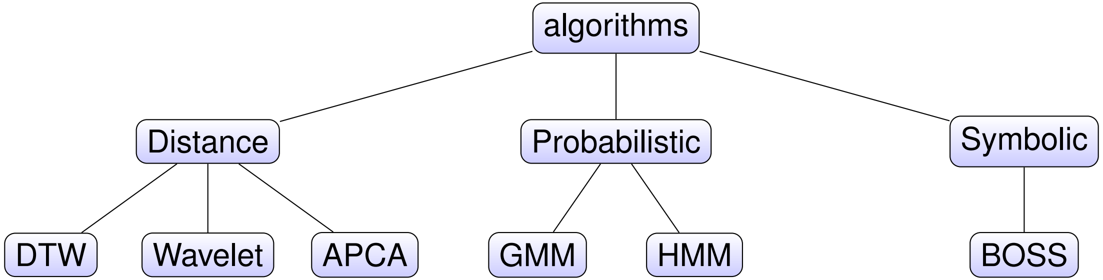
- Comparing euclidean distance or more complex routines such as DTW
- Transforming data to Wavelet representation and measure distance in Wavelet subspace

Probabilistic

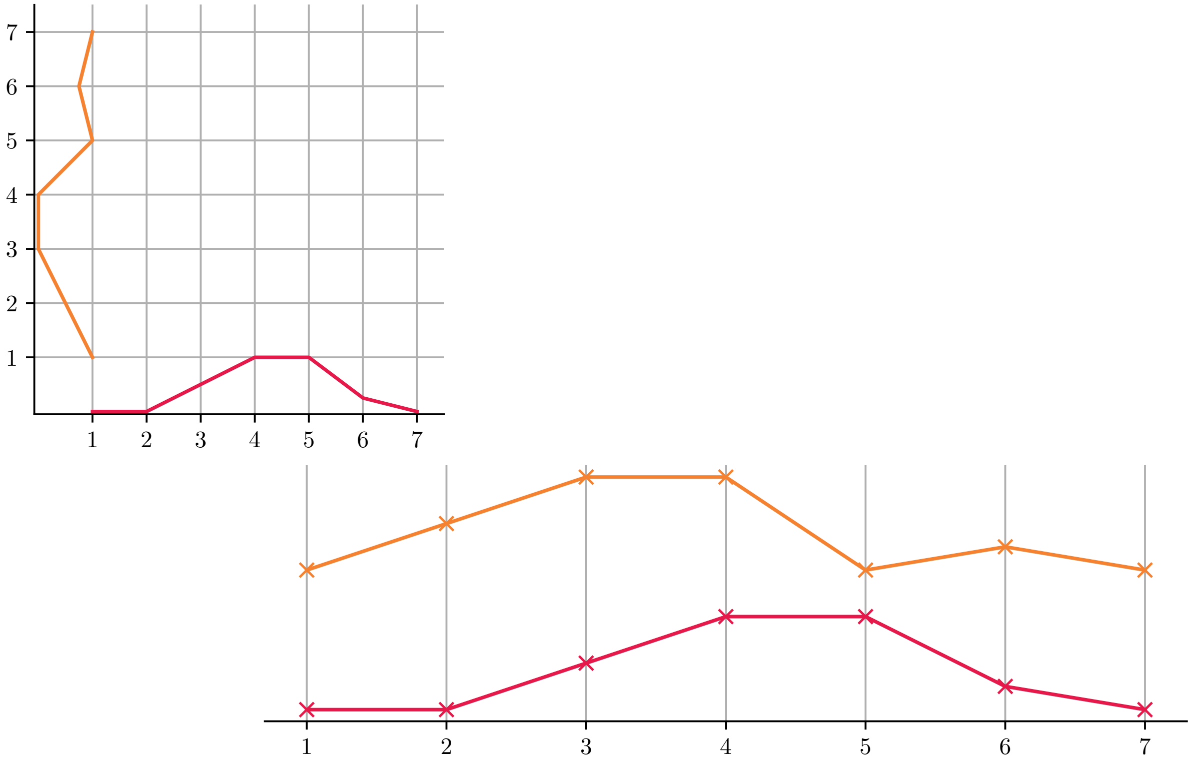
- Represent time series as probabilistic model
- Compare likelihoods for similarity

Symbolic

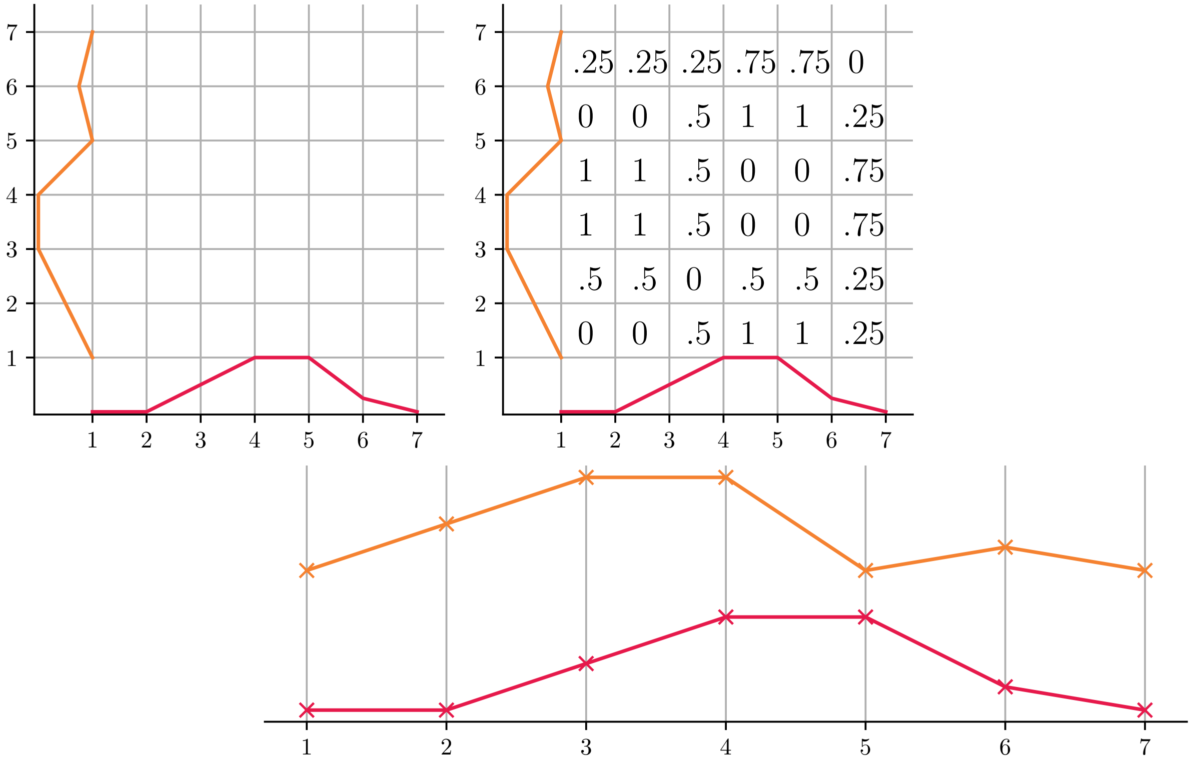
- Transform data to a symbolic representation
- Measure the symbolic distance, i.e. similarity of strings



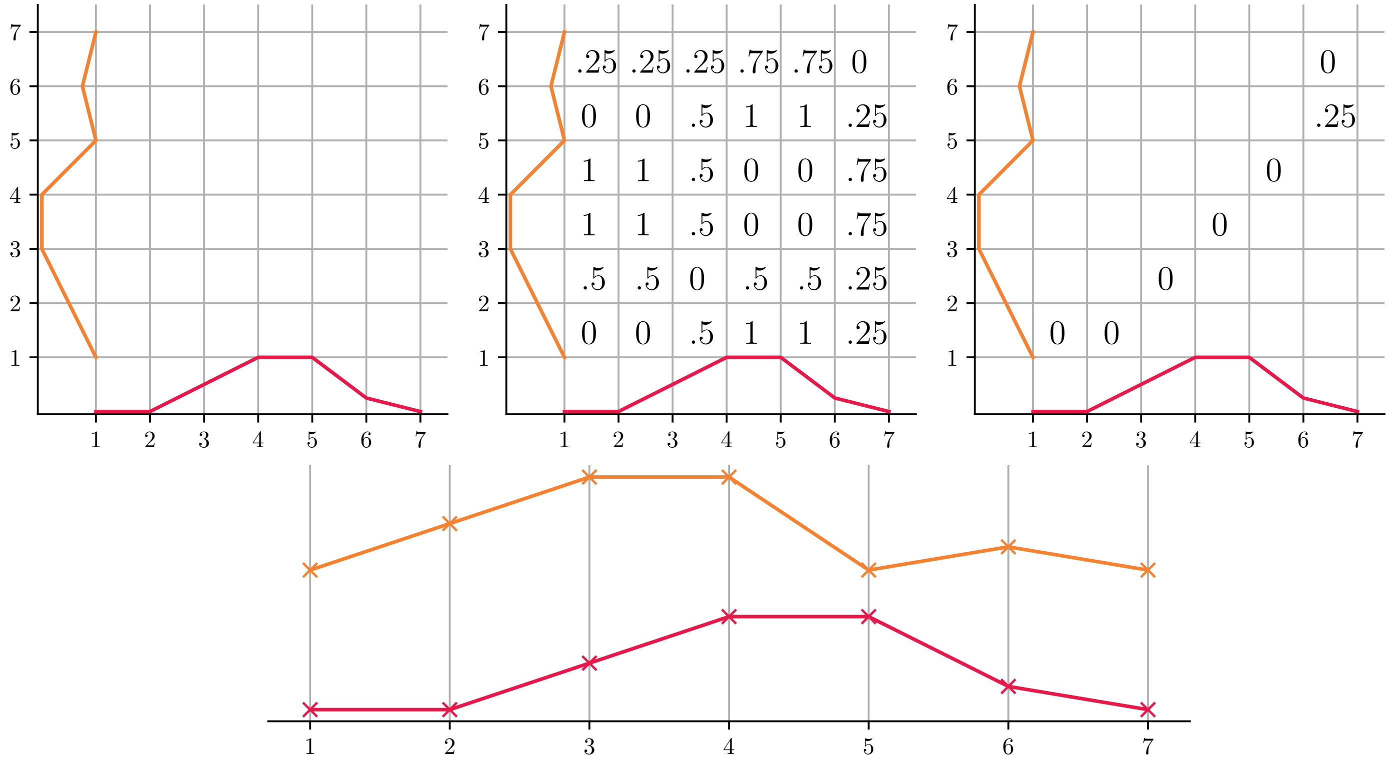
Algorithms - Dynamic Time Warping



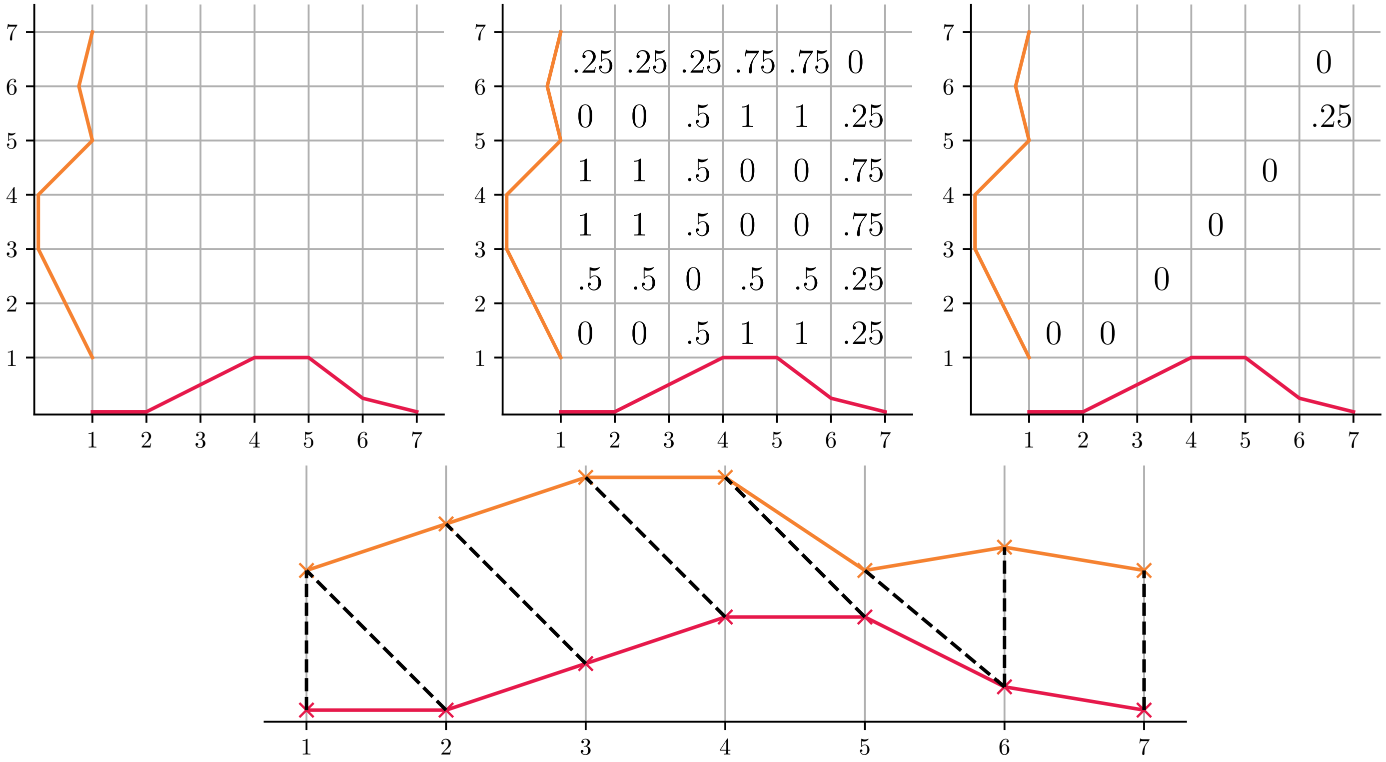
Algorithms - Dynamic Time Warping



Algorithms - Dynamic Time Warping

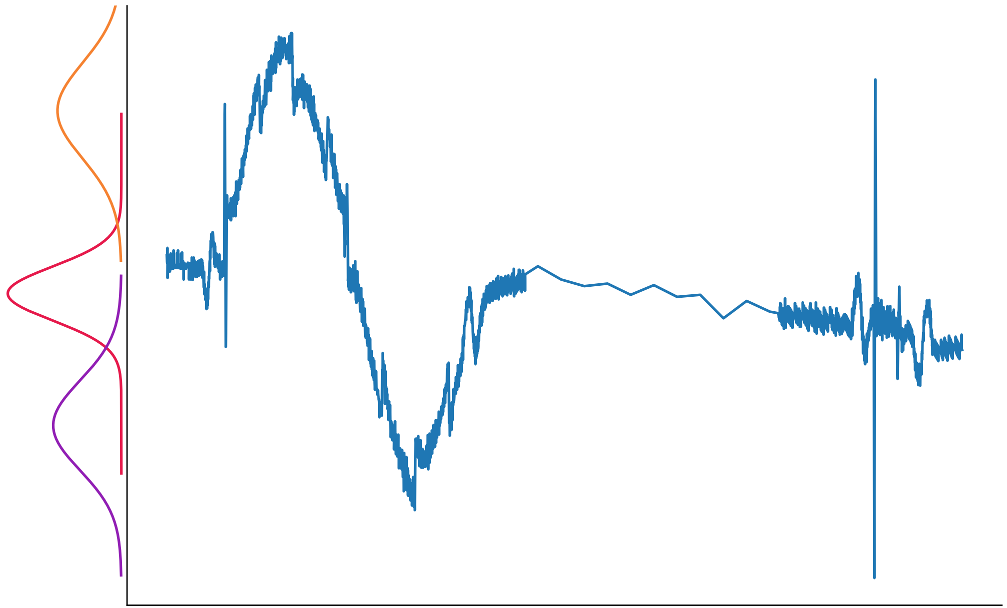


Algorithms - Dynamic Time Warping



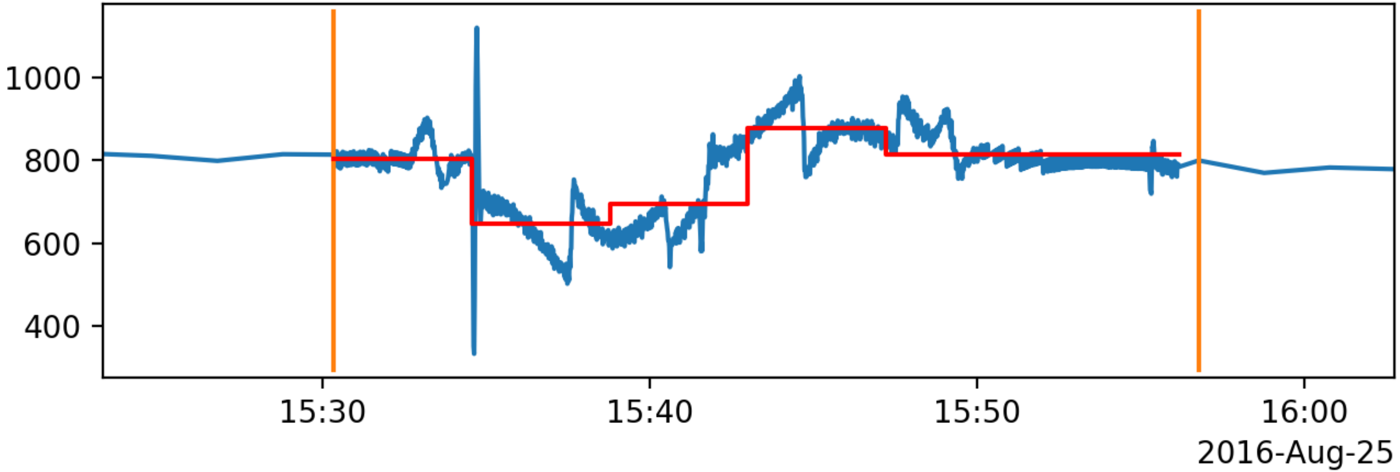
Algorithms - Gaussian Mixture Model

- Represent time series as a Gaussian Mixture Model
- Use Likelihoods to determine the similarity of two time series
- Different probabilistic models can be used as well



Algorithms - APCA

- Approximation of a time series with *n* (is a hyperparameter) segments of variable length
- Is based on a discrete Haar wavelet transformation
- Preserves only the essential structure of the time series
- Euclidean distance is measured between two APCA approximations



Algorithms - Ensemble

Every algorithm has different strengths and weaknesses

Idea: Combine them to one strong algorithm

- Common technique in time series forecasting
- Decreases the variance and often leads to better results

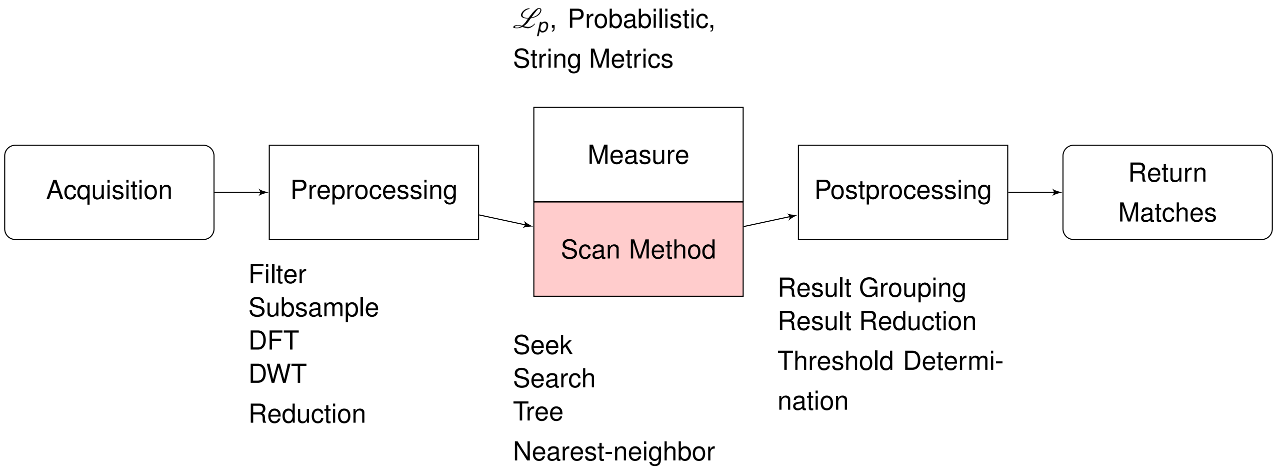
Functionality

- Execute DTW, GMM and APCA independently on a query time series
- Return a match if at least one algorithm matches the time series
- Determine quality by scaling all qualities to their best threshold

Further ideas

- Determine weights depending on the shape of the query time series
- More sophisticated approaches can lead to overfitting

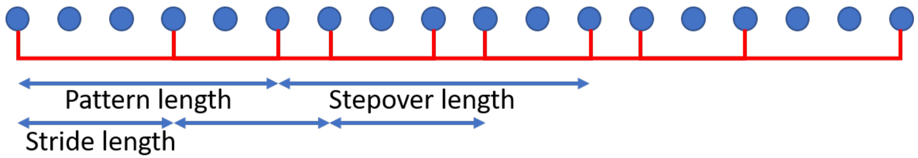
Basic Structure of Time Series Matching



Scan Method

Problem: Searching through a time series can be done in several ways and directly affects the return of matches

General idea: Based on a stride length and a stepover length one walks through the time series



Fixed Walk

Walk through time series in fixed stride length

Potential problems

Can return a lot of overlaps

Smart Walk

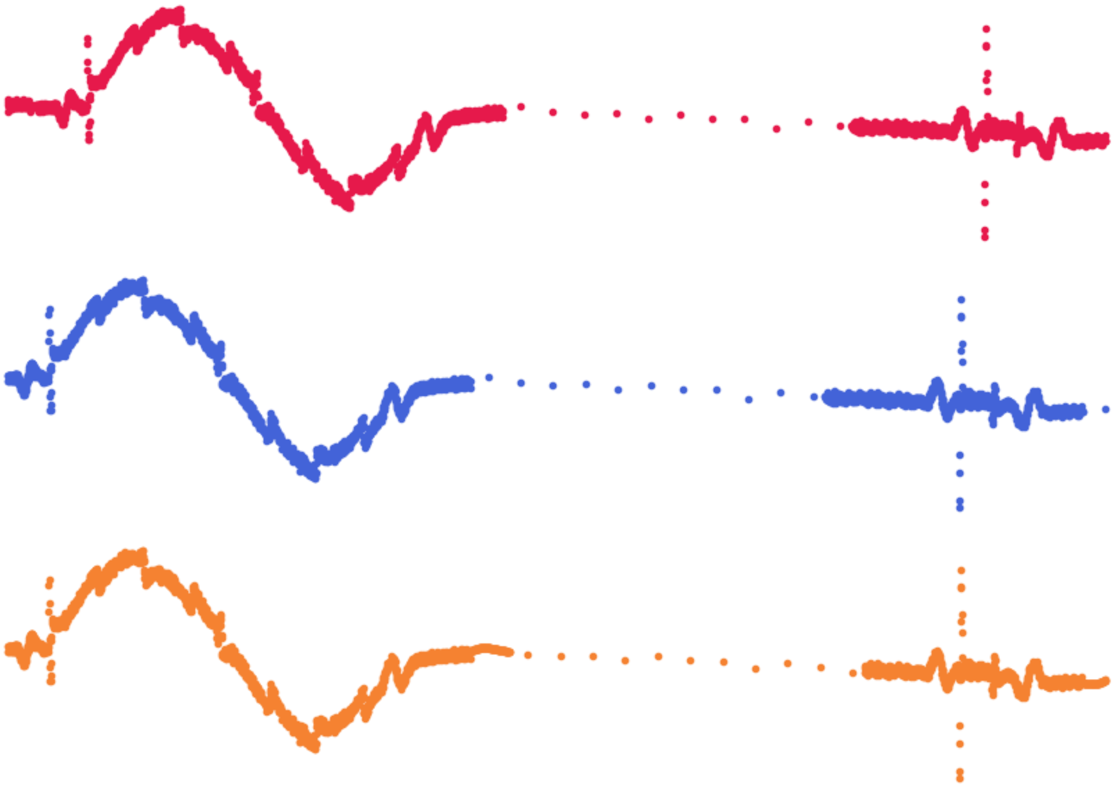
Walk through time series in fixed stride length until a match occurs

If a match occurs add the stepover length to avoid any overlaps

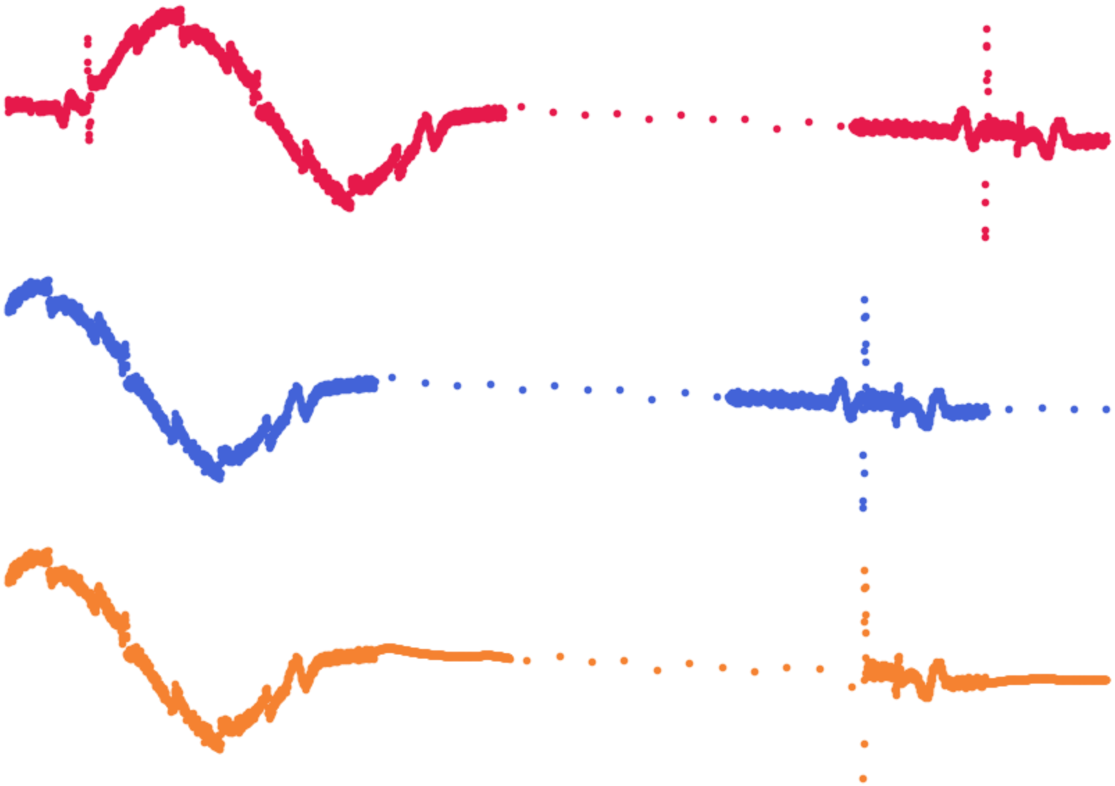
Potential problems

Since it returns the first match of a potential series of matchs this could be an edge match

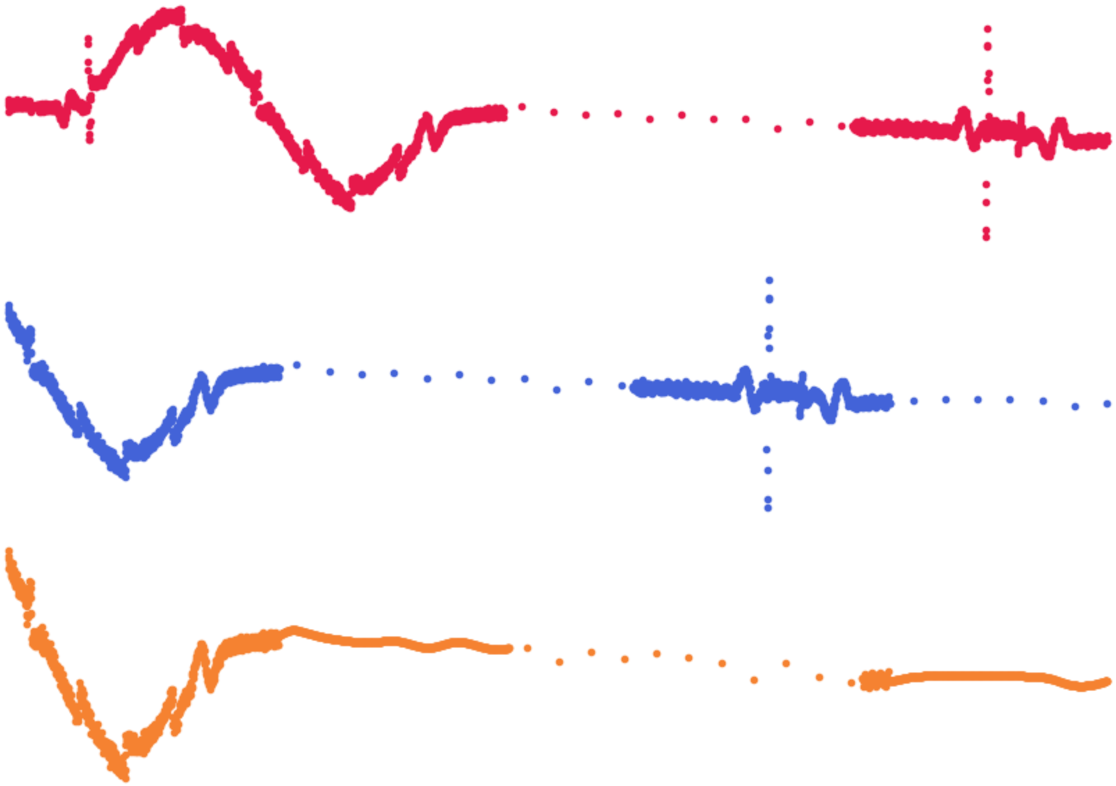
Resampling



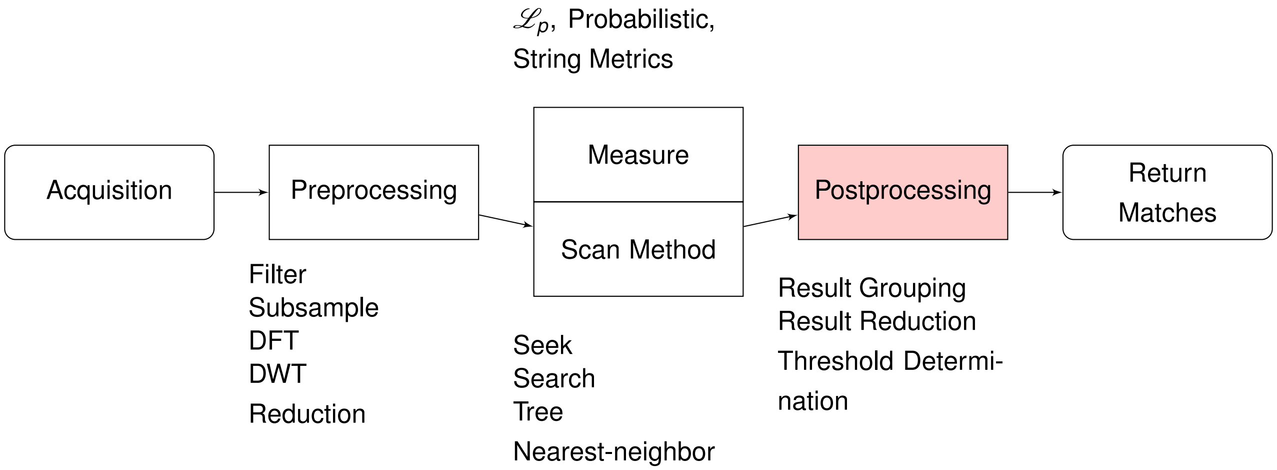
Resampling



Resampling

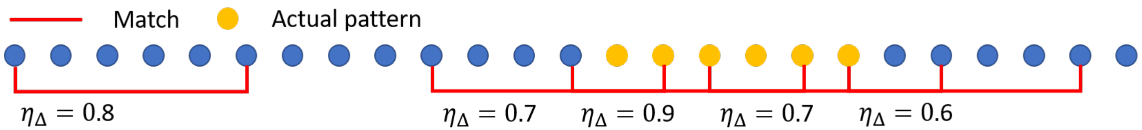


Basic Structure of Time Series Matching



Postprocessing methods

Problem: Depending on the choice of our threshold we get a lot of overlapping matches which negatively affect the user experience



Combine-group

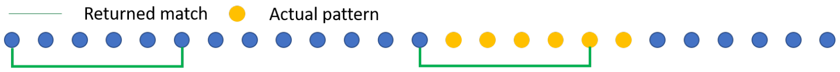
Idea: Merge all overlapping regions together

- Quality of best match in group is returned
- Can potentially combine different unique matches

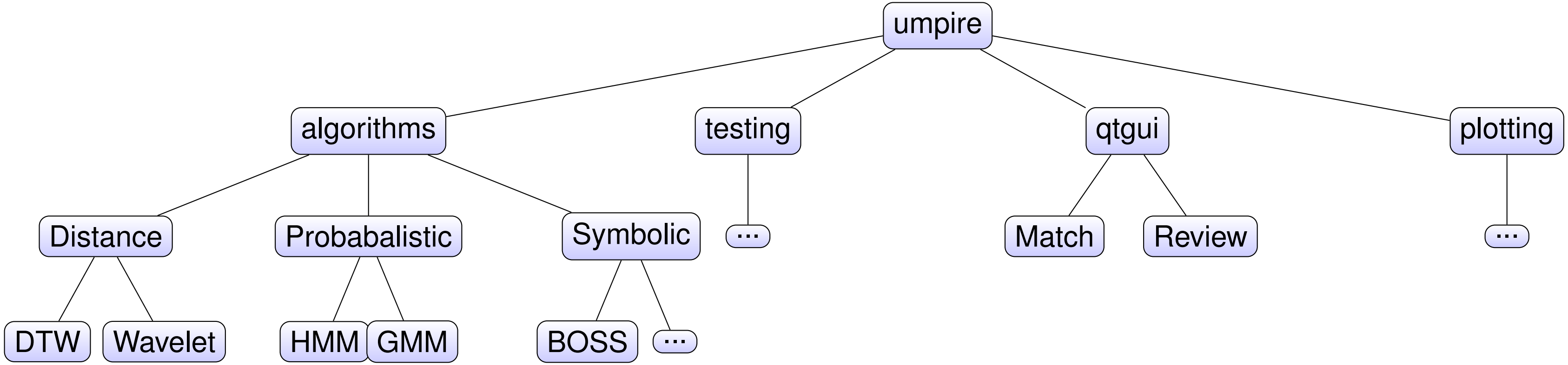
Best-in-group

Idea: Choose best quality within all overlapping regions

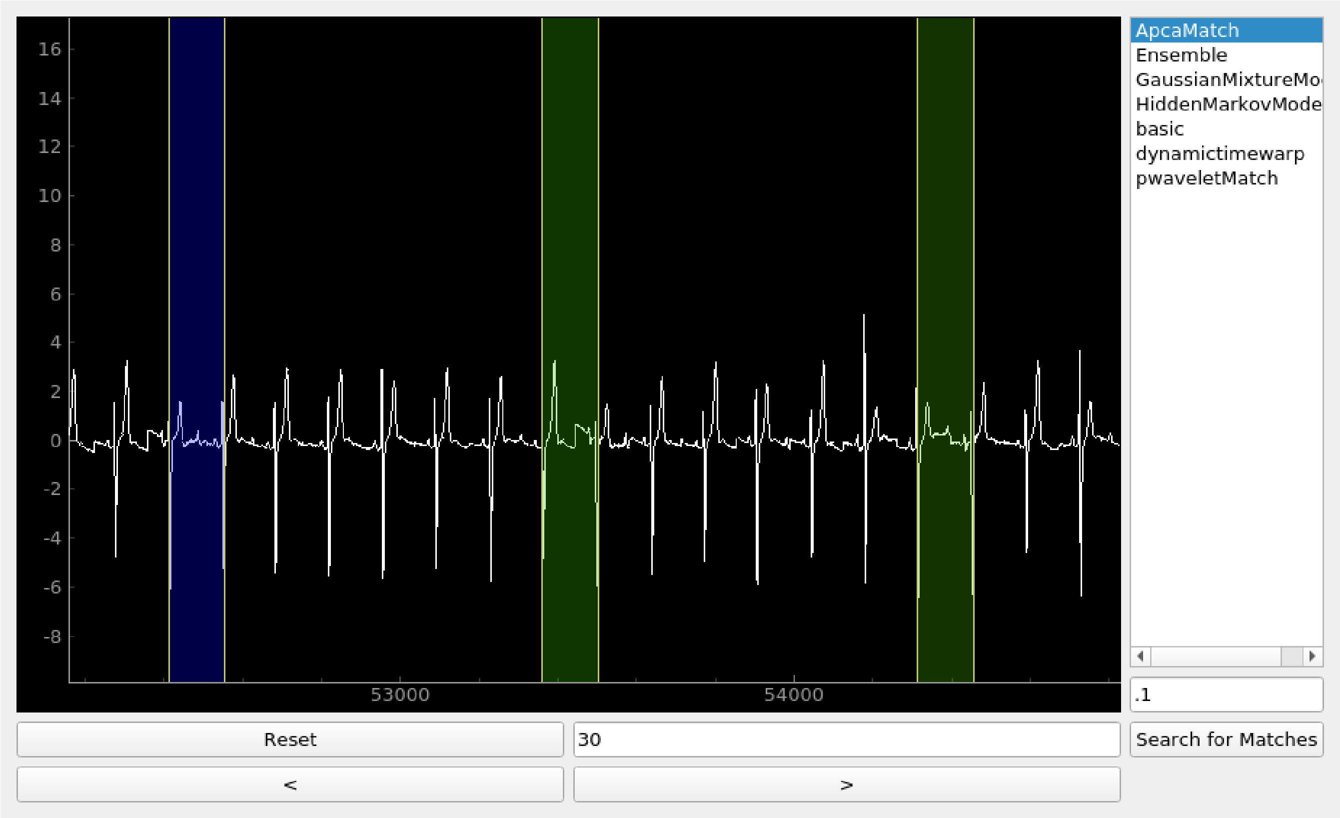
- Returns only a single region
- Can lose some possible matches

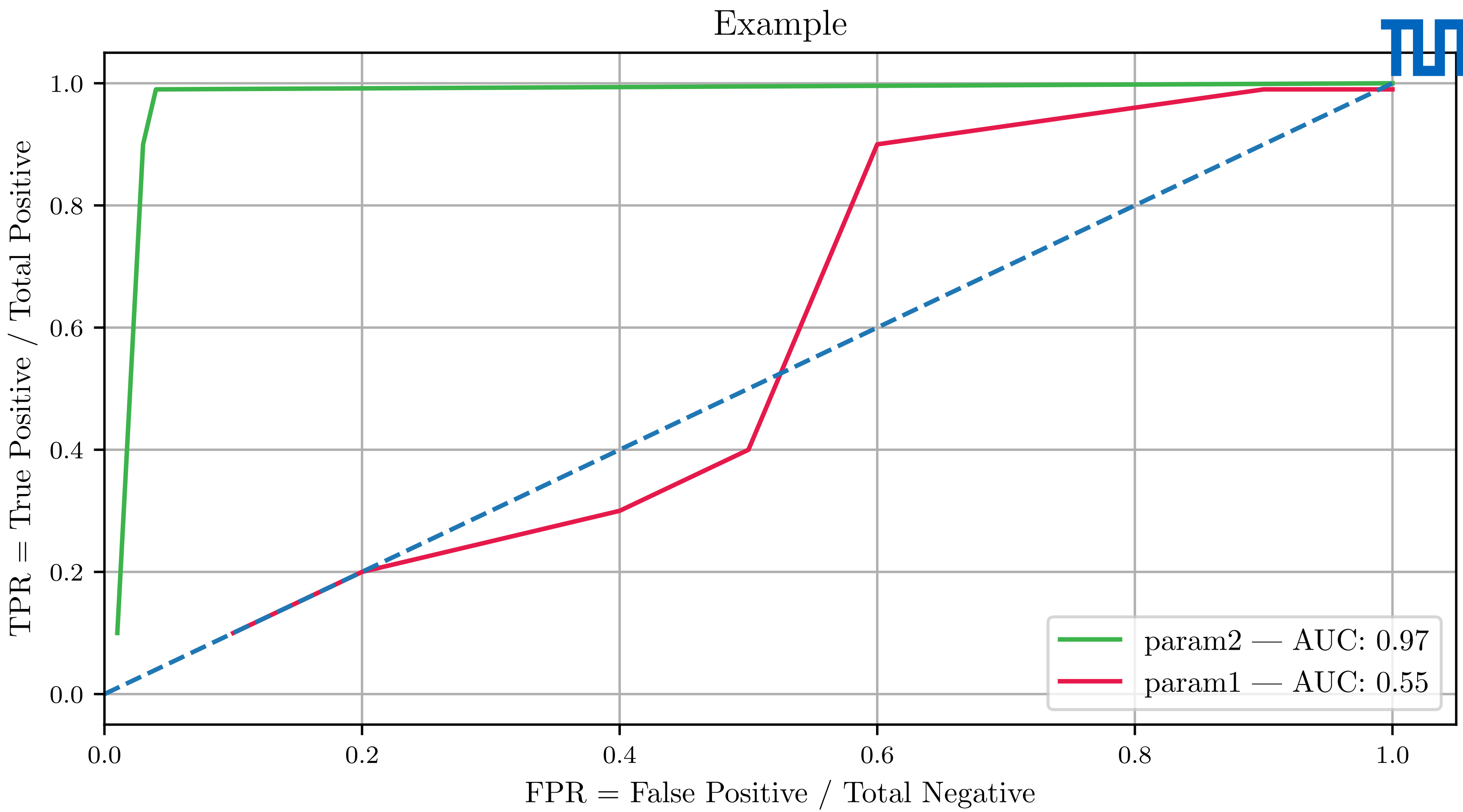


Implementation



Matching tool




Example
TPR = True Positive / Total Positive
FPR = False Positive / Total Negative
0.0
0.2
0.4
0.6
0.8
1.0
param2 — AUC: 0.97
param1 — AUC: 0.55

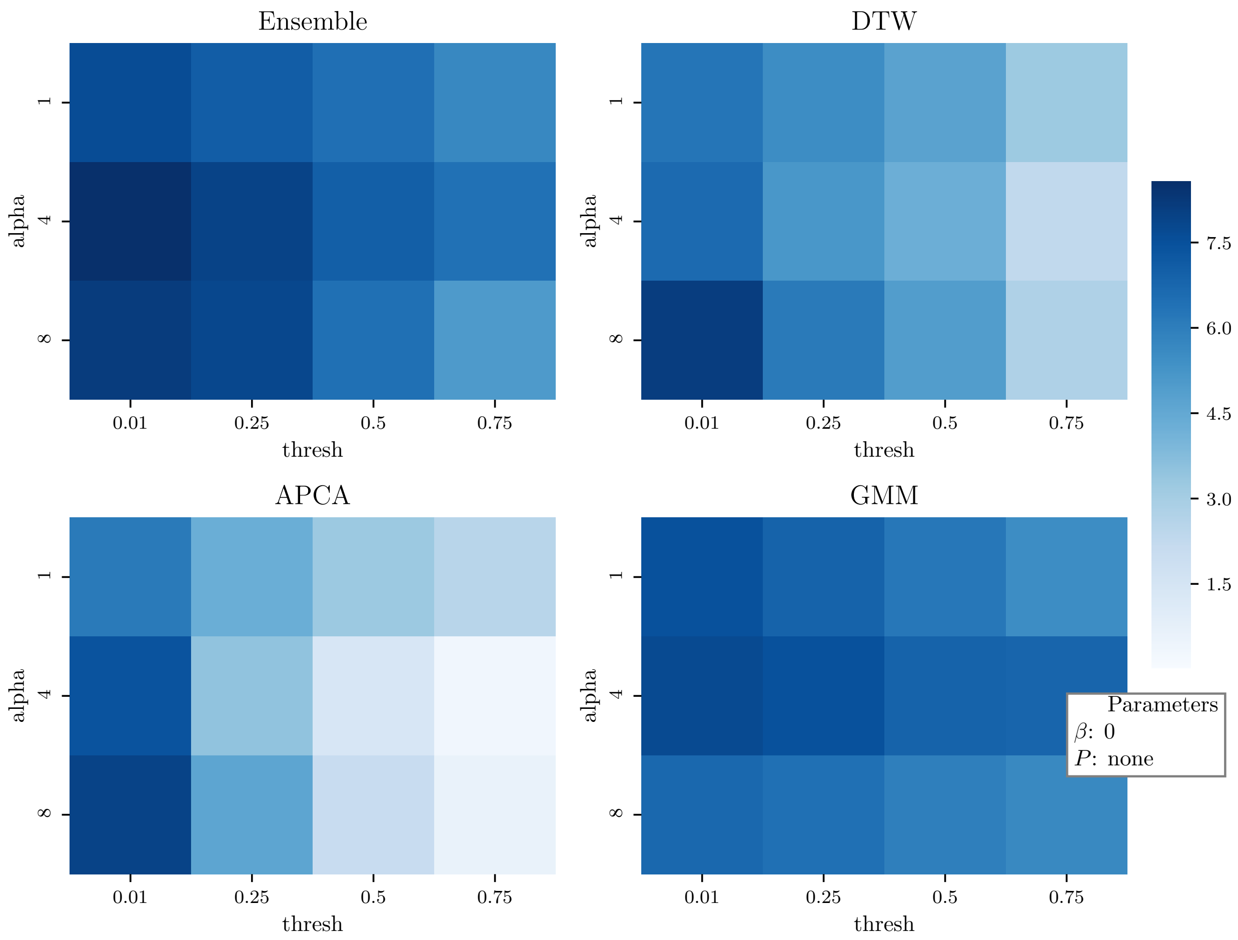
Parameter optimization study

All algorithms are affected by following parameters:

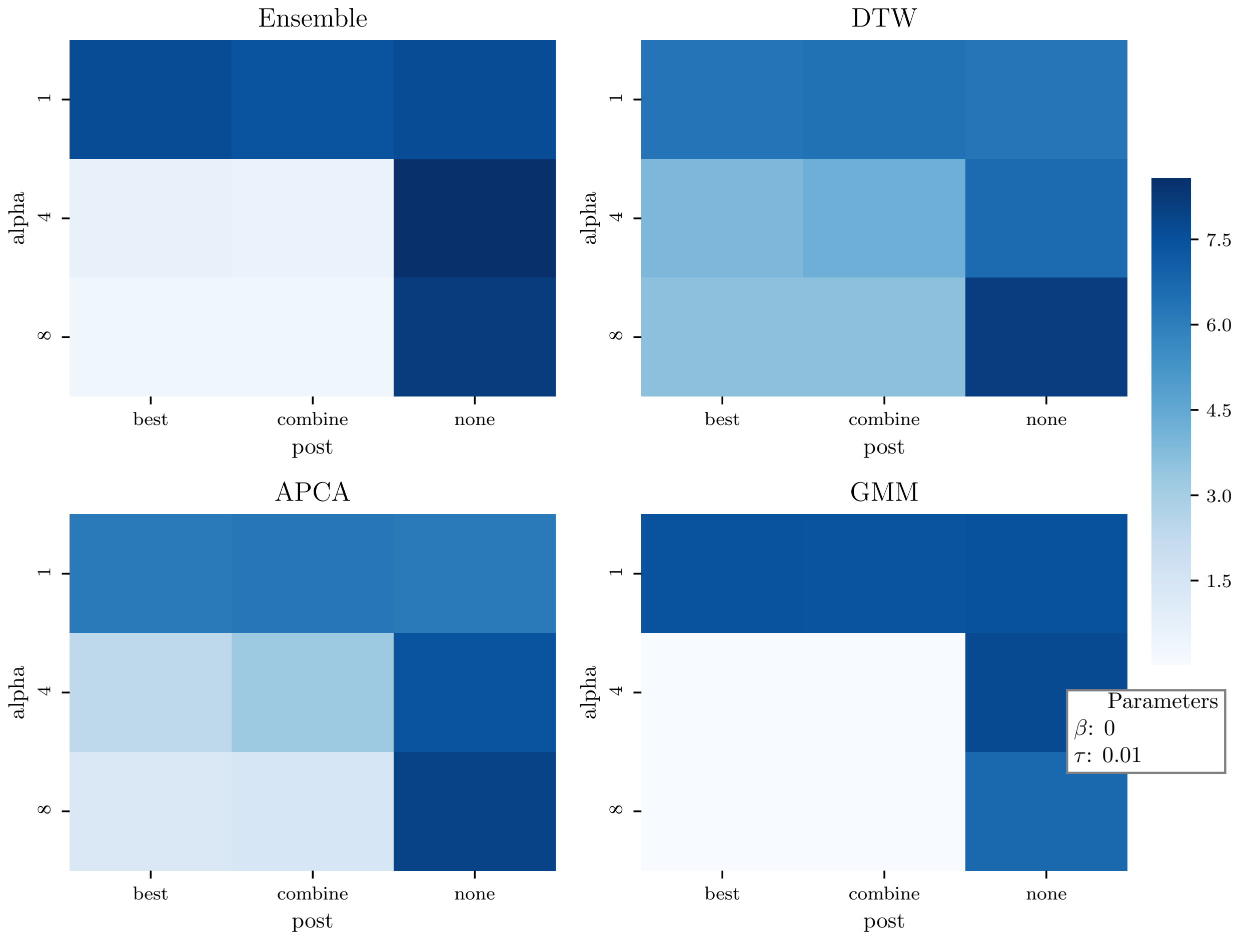
Symbol	Description	range
τ	**Quality measure threshold**, below which a potential match is discarded.	0.1 - 0.75
α	**Step size fraction**. The fraction of the query length (in time) which is used as the stride for a seek	$1, \frac{1}{2}, \frac{1}{4}, \frac{1}{8}$
β	**Stepover fraction**. The fraction of the query length to step over in the case of a smart walk. $\beta = 0$ is defined to be a fixed-walk.	off, $1, \frac{1}{2}$
P	**Postprocessing method**	none, best, combine

- Parameter study was conducted with the UCR data set (5 data sets with a total of 14 test cases)
- Performance measure was the AUC metric
- AUC evaluates the performance of an algorithm on a test case with a single value from [0,1] (1 = perfect classifier)
- We will present the results in aggregate: At each combination of parameters, the sum of all AUC scores across test cases is used as the final score (in our case, a score of 14 would represent a perfect classifier)

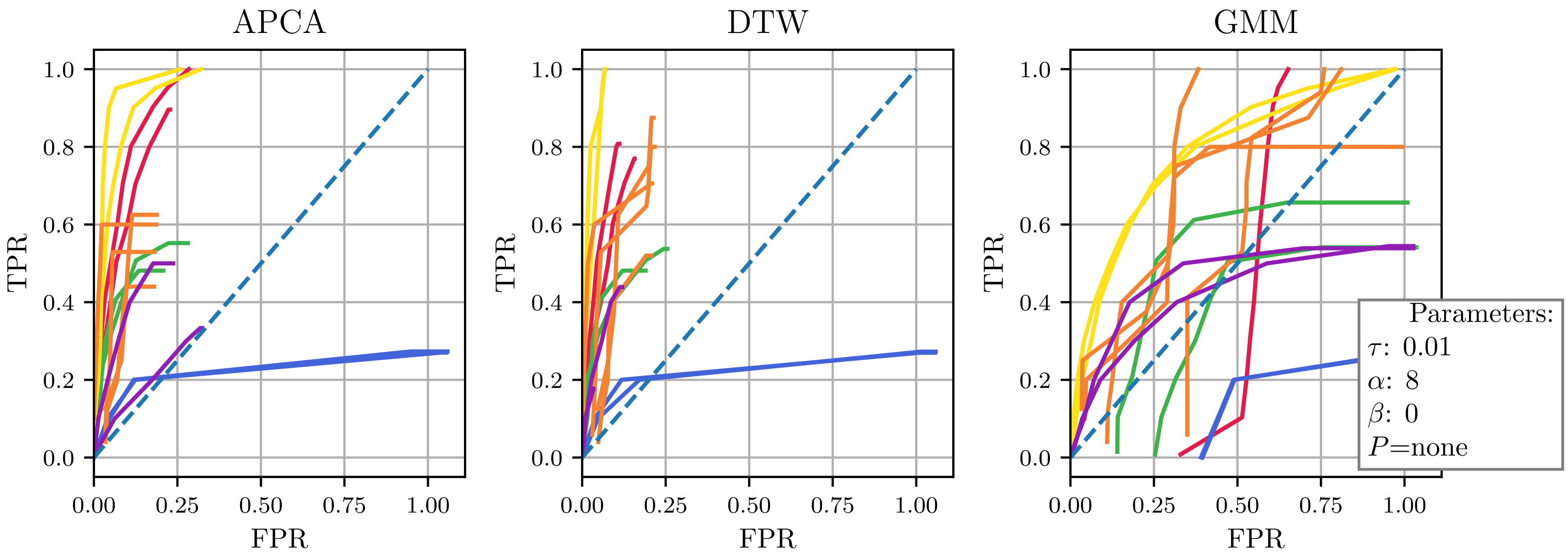
Parameter combinations of the step size fraction (alpha) and the quality measure threshold



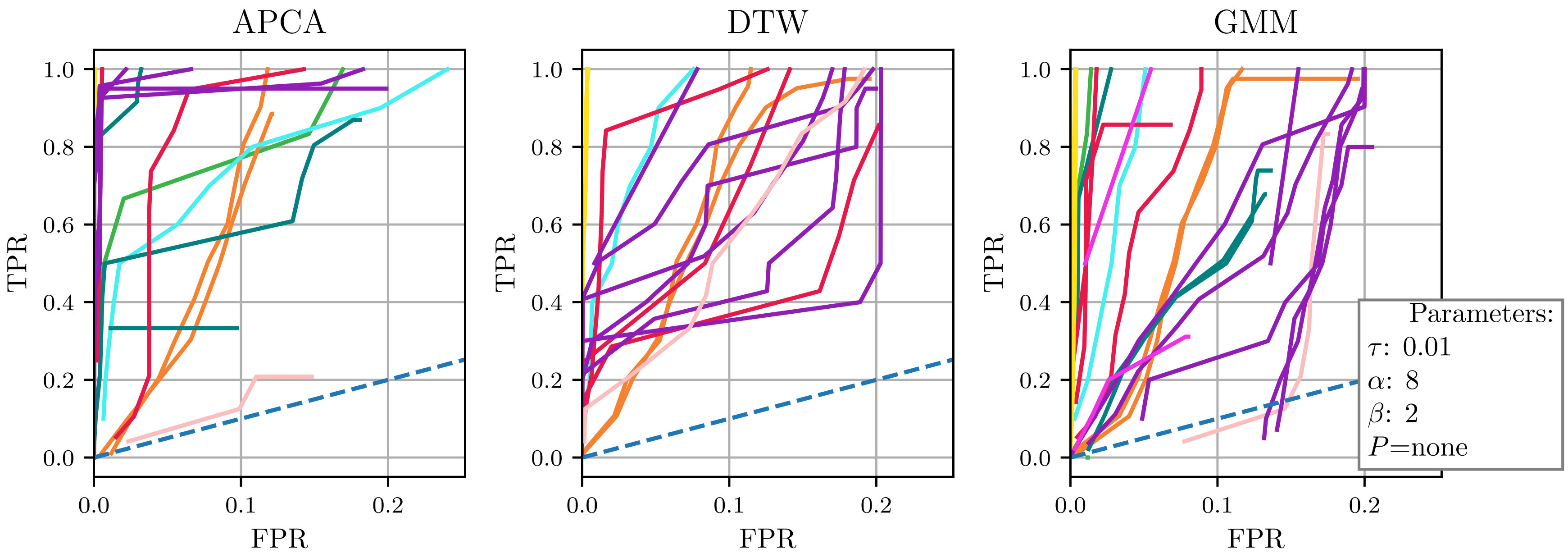
Parameter combinations of the step size fraction (alpha) and postprocessing method



AUC metric with UCR data

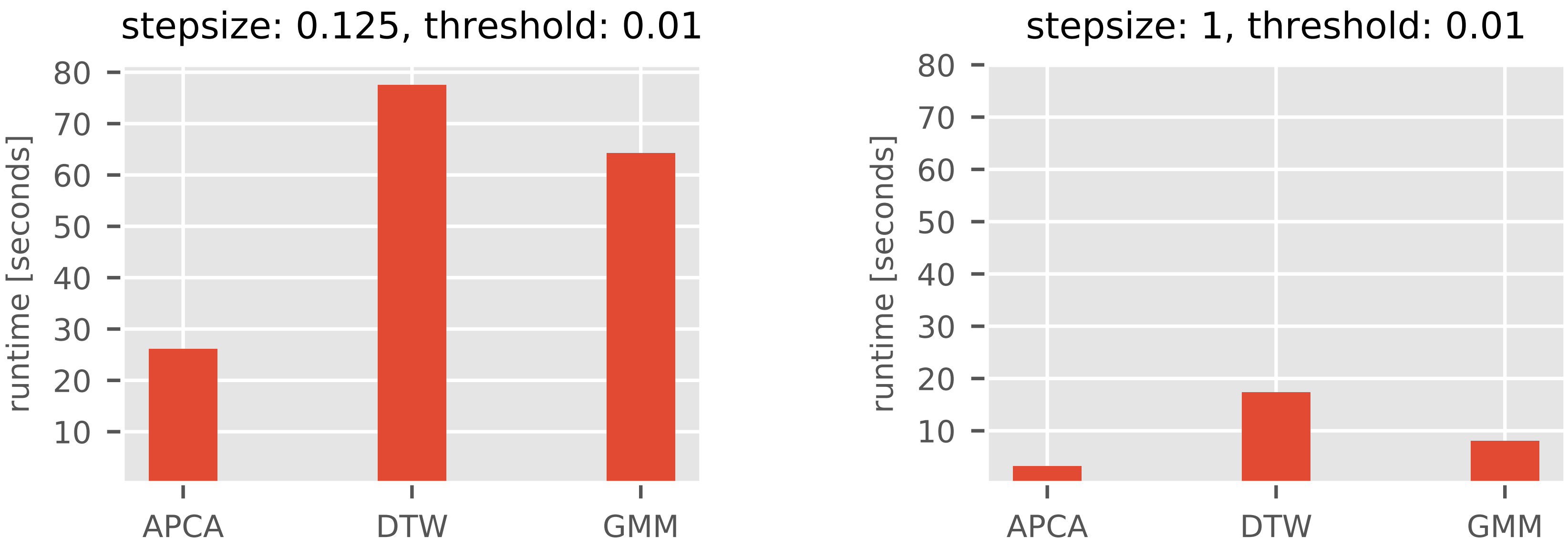


AUC metric with DLR data



Example Runtime

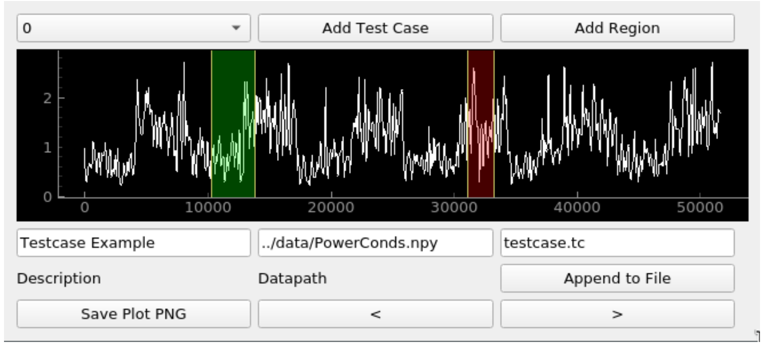
- Runtime on parameter 12 of the DLR data set
- Parameter consists of 37,673,423 data points

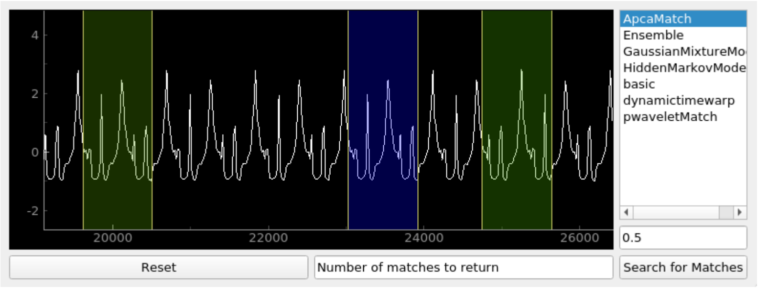


Conclusion

Achievements

- Authored a software suite
- User-selectable region matching from time-series data
- 5 algorithms included
- Easily extendable with additional algorithms, performance measures, postprocessing functions, or combinations thereof
- GUI tool for manual time-series pattern labeling or direct review of algorithm performance on a particular test pattern





Thank you for your attention!

Any questions?

Algorithms - Dynamic Time Warping

Complexity considerations:

- In general, $O(NM)$ space and time complexity. For N, M length of search and test query, respectively.
- In many cases, $N \approx M$ and complexity is $O(N^2)$
- Note that this is $O(N^2)$ per potential match! Scanning through the entire time history is then worse than $O(N^3)$

Improvements:

- Early termination due to minumum error accumulation
- Enforce matrix *bandwidth* (maximum point-to-point scaling) maximum
- Enforce maximum path slope conditions (limit on severity of nonlinear scaling)

Algorithms - Gaussian Mixture Model

Definition of Gaussian Mixture Models

K number of mixture components, N number of observatons

$\phi_{i=1,\dots,K}$ weights for distributions

$\theta_{i=1,\dots,K} = \{\mu_{i=1,\dots,K}, \sigma_{i=1,\dots,K}\}$ with $\mu_{i=1,\dots,K}$ mean and $omega_{i=1,\dots,K}$ variance for every component

$z_{i=1,\dots,N}$ is a categorial variable representing the mixture component

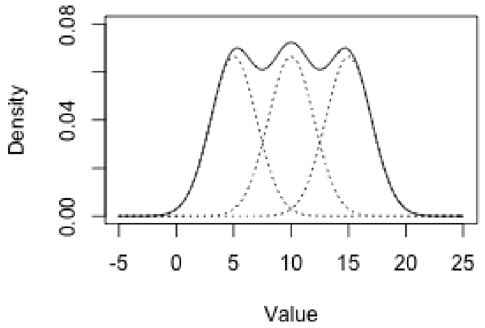
$x_{i=1,\dots,N}$ is $\mathcal{N}(\mu_{z_i}, \sigma_{z_i})$ distributed

Idea

- Mixture components represent the different states of the time series
- Different plateaus vary
  in mean and different volatilities can be characterized with the different variances
- Frequency of different states is represented by the weights

Advantages

- Extension to multidimensional pattern possible
- Combination with different preprocessing methods (Wavelet etc.) possible
- Capturing of structured data



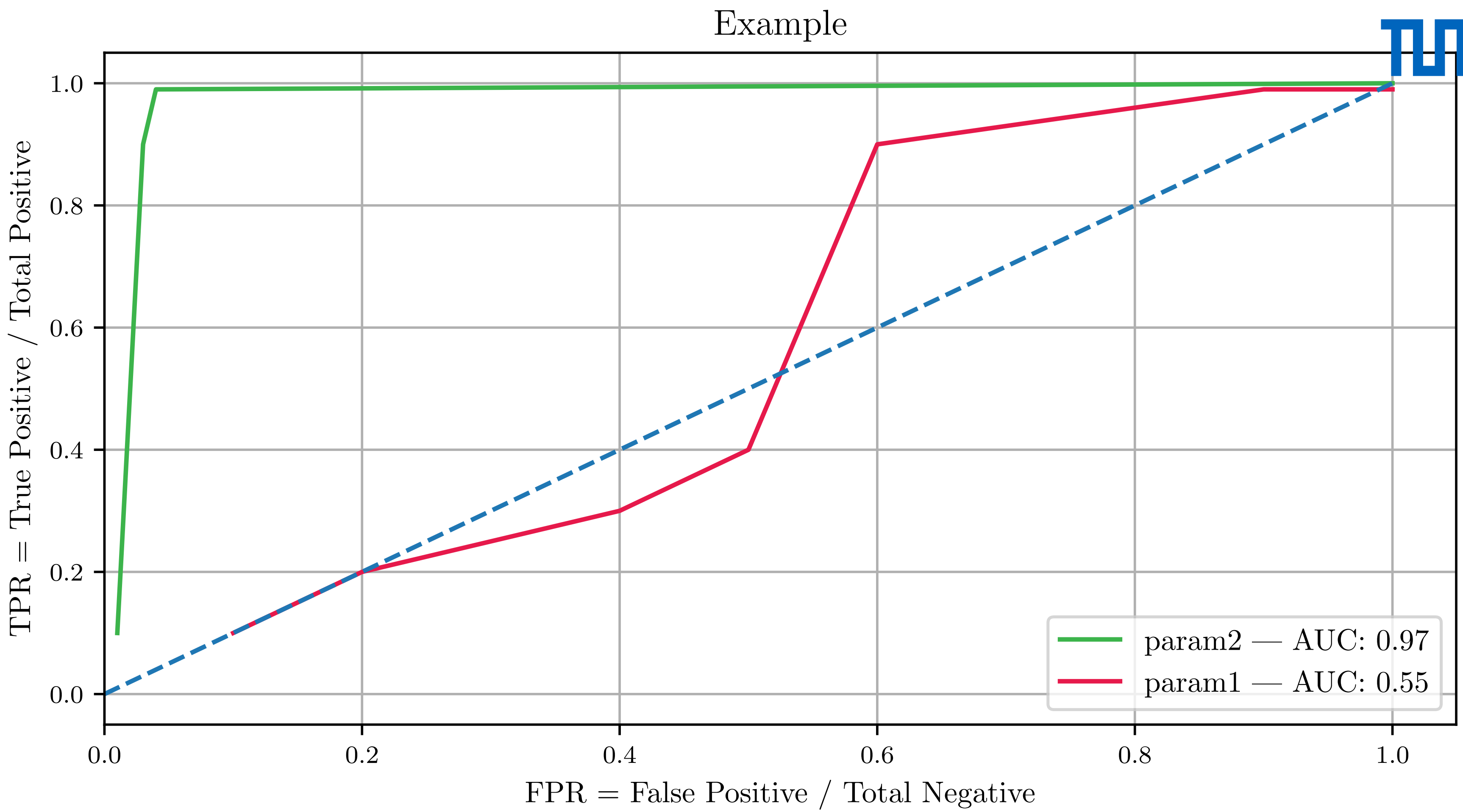
[1]

[1] https://brilliant.org/wiki/gaussian-mixture-model/

Performance Measure

Receiver Operating Characteristic

- Measure for performance of binary classifier across different algorithms
- Comparing true-positive rate against false-positive rate
- Step function to 1 for perfect classifier and diagonal for random classifier


Example
1.0
0.8
0.6
0.4
0.2
0.0
TPR = True Positive / Total Positive
0.0
0.2
0.4
0.6
0.8
1.0
FPR = False Positive / Total Negative
param2 — AUC: 0.97
param1 — AUC: 0.55

Performance Measure

Own implementation

- True-positive means that at least 40% of a return overlap a labeled region; false-positive if it does not; false-negative if it does not return a match
- Choose $\frac{8 \cdot \text{Total Timeseries Duration}}{\text{Query Duration}}$ as overall number of regions to be classified.