Machine Learning used for event identification

6th Collaboration Meeting - Grenoble

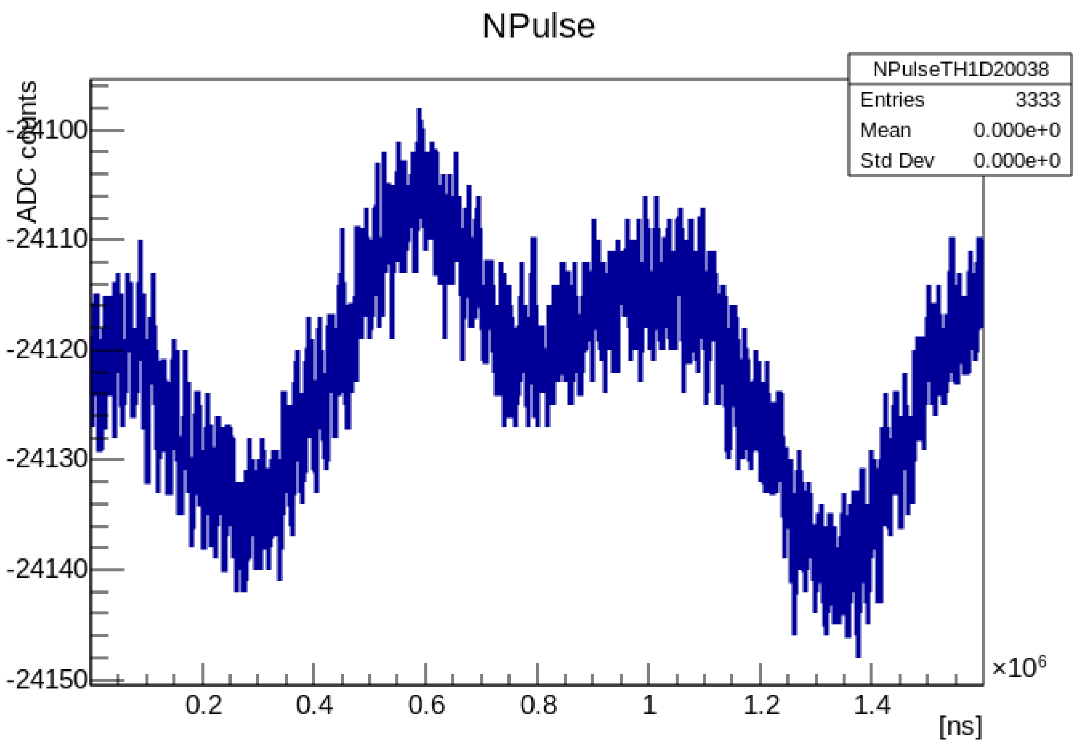
12th June 2019

Alex Rolland

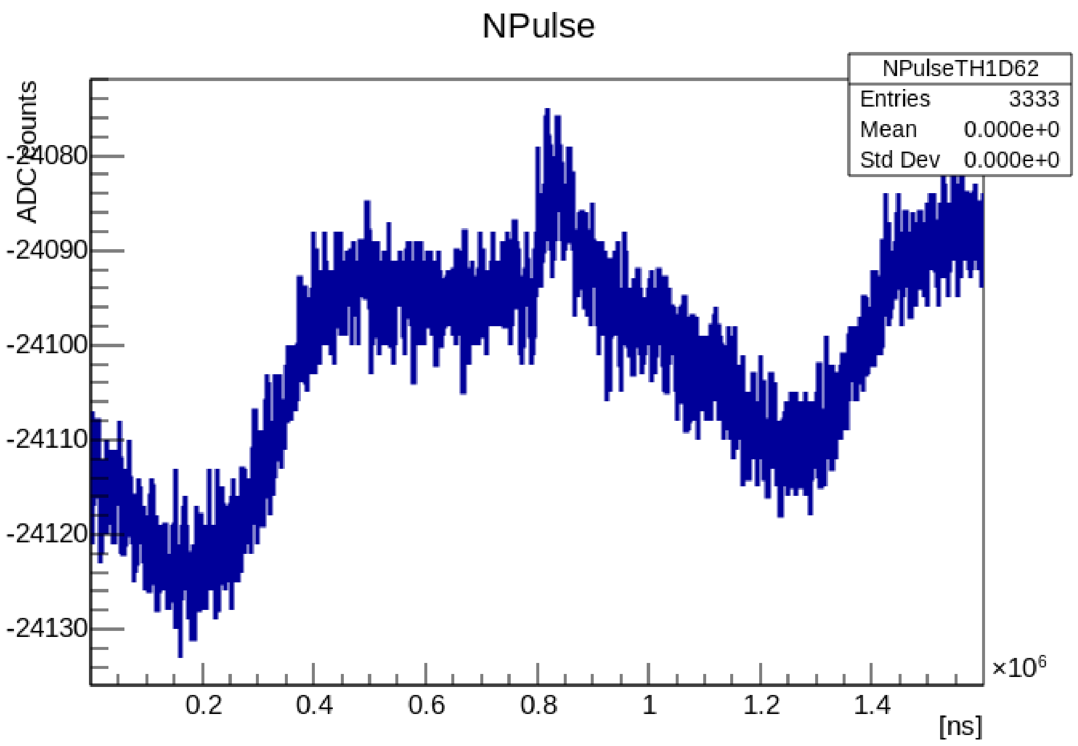
Outline

- Motivations
- Machine-learning methodology
- Results
  - Comparison with a treshold-based algorithm
  - Efficiency
  - Time & amplitude reconstruction
- Conclusion

Difficulty in identifying of low-energy events



Noise



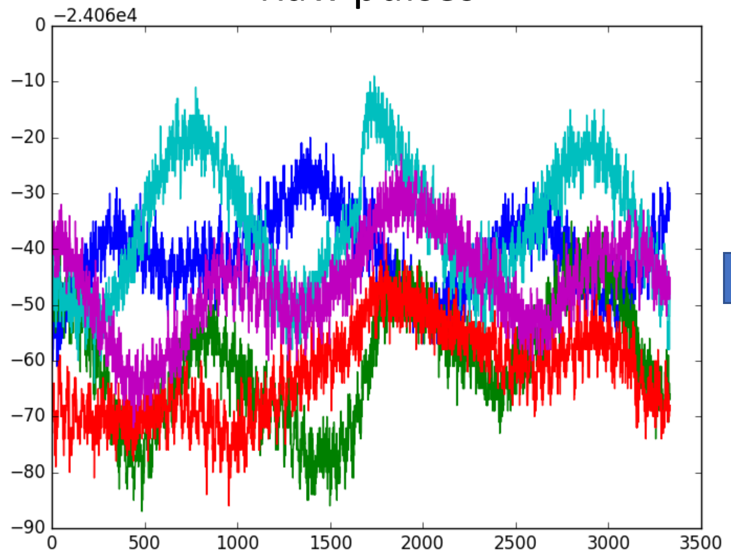
1 e- event

Machine learning methodology

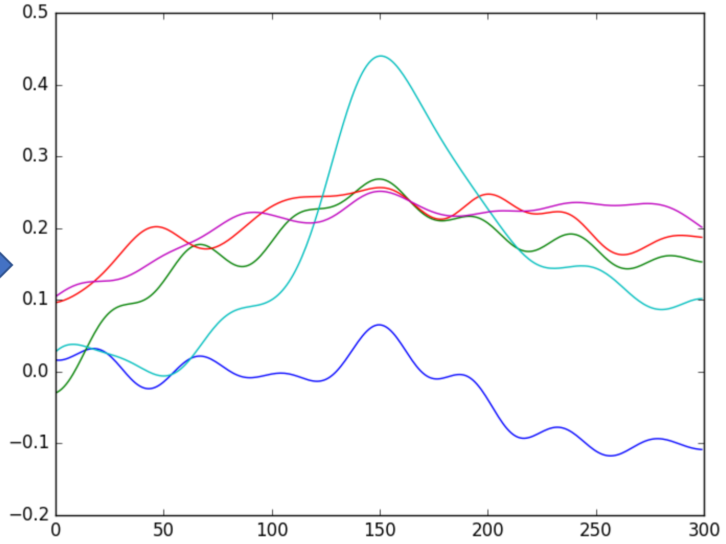
10 000 events of SEDINE 3 bar Ne simulations (~70% noise, 30% 1e-)

- Process the data :
  - Remove the baseline
  - Deconvolve
  - Filter
  - Center the pulse
  - Shorten the window

Raw pulses

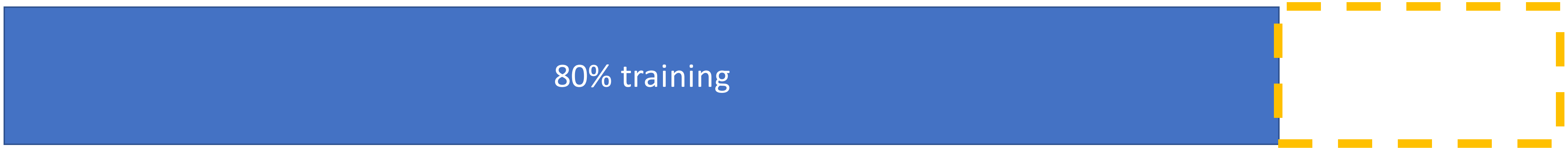


Processed pulses

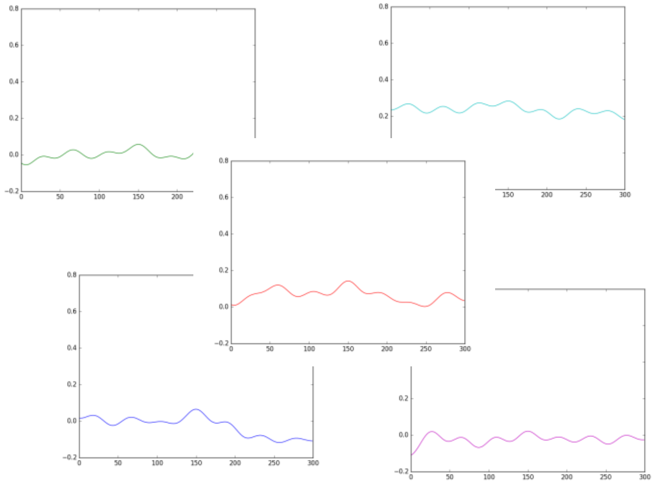


From 3333 to 300 bins

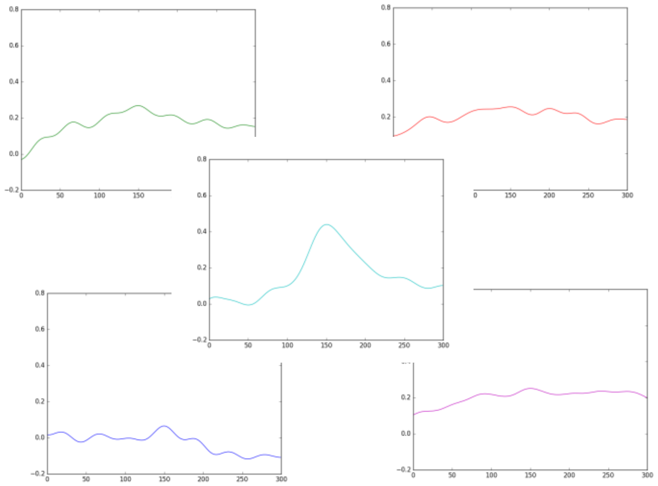
Machine learning methodology



- Use 80% of these data to **train** the model with a classification algorithm called **Support Vector Machine (SVM)**



Noise

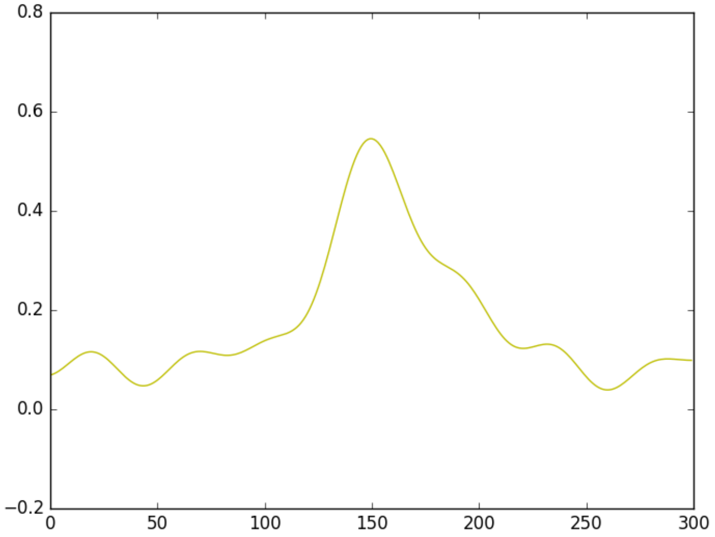


1 e- events

Machine learning methodology

80% training | 20% testing

- **Test** the model on the remaining 20%



Positive ✓

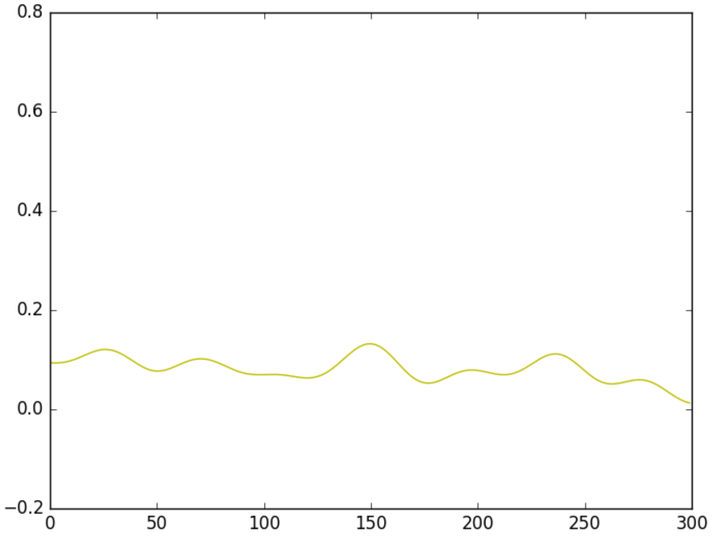
Confusion matrix

Found positive	Found negative	
True P	False N	Actually positive
False P	True N	Actually negative

Machine learning methodology

80% training | 20% testing

- **Test** the model on the remaining 20%



→ Negative ✓

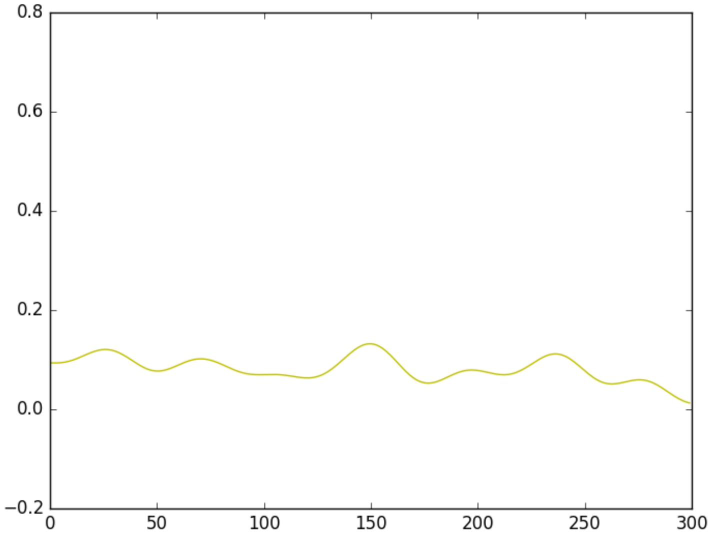
Confusion matrix

	Found positive	Found negative
Actually positive	True P	False N
Actually negative	False P	True N

Machine learning methodology

80% training	20% testing

- **Test** the model on the remaining 20%



→ Positive ✗

Confusion matrix

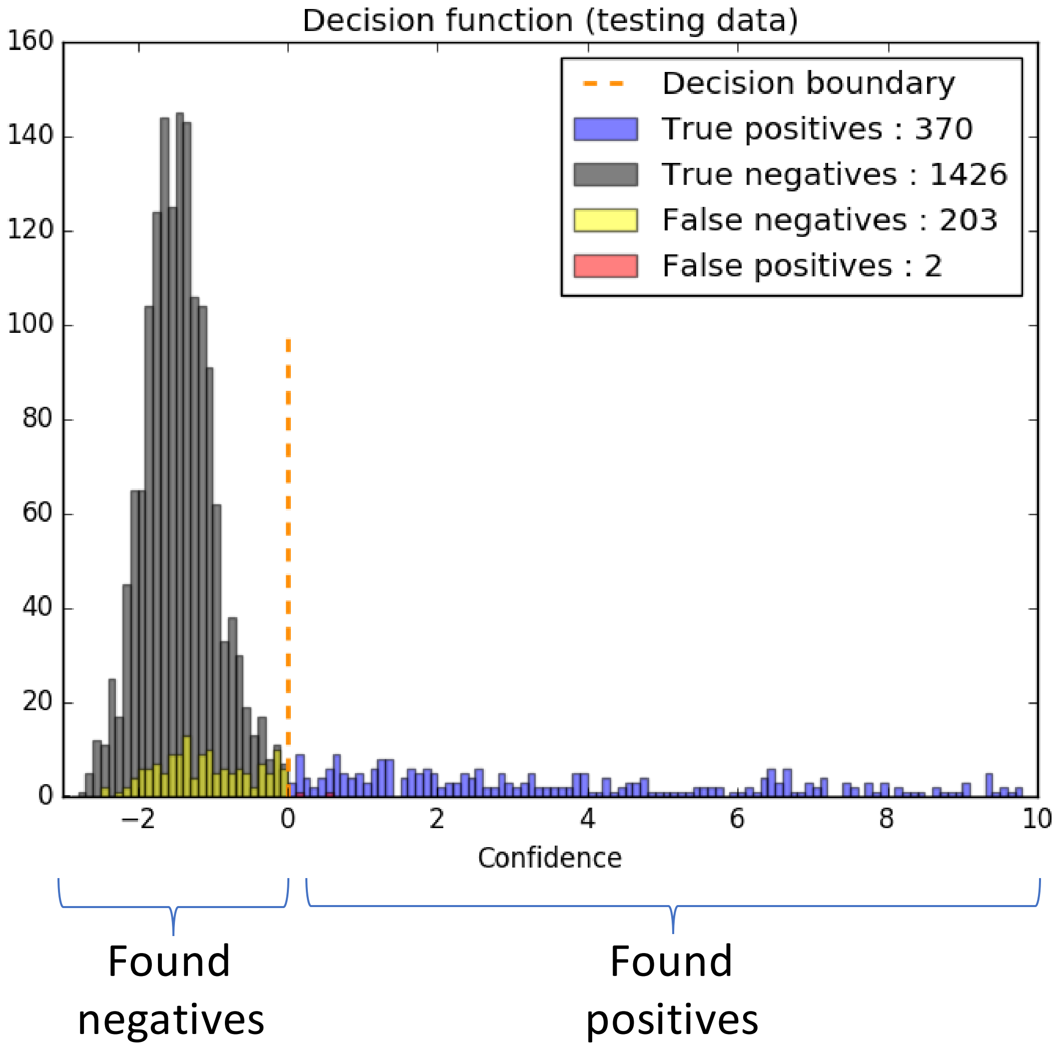
	Found positive	Found negative
Actually positive	True P	False N
Actually negative	False P	True N

Results on the 2000 testing samples

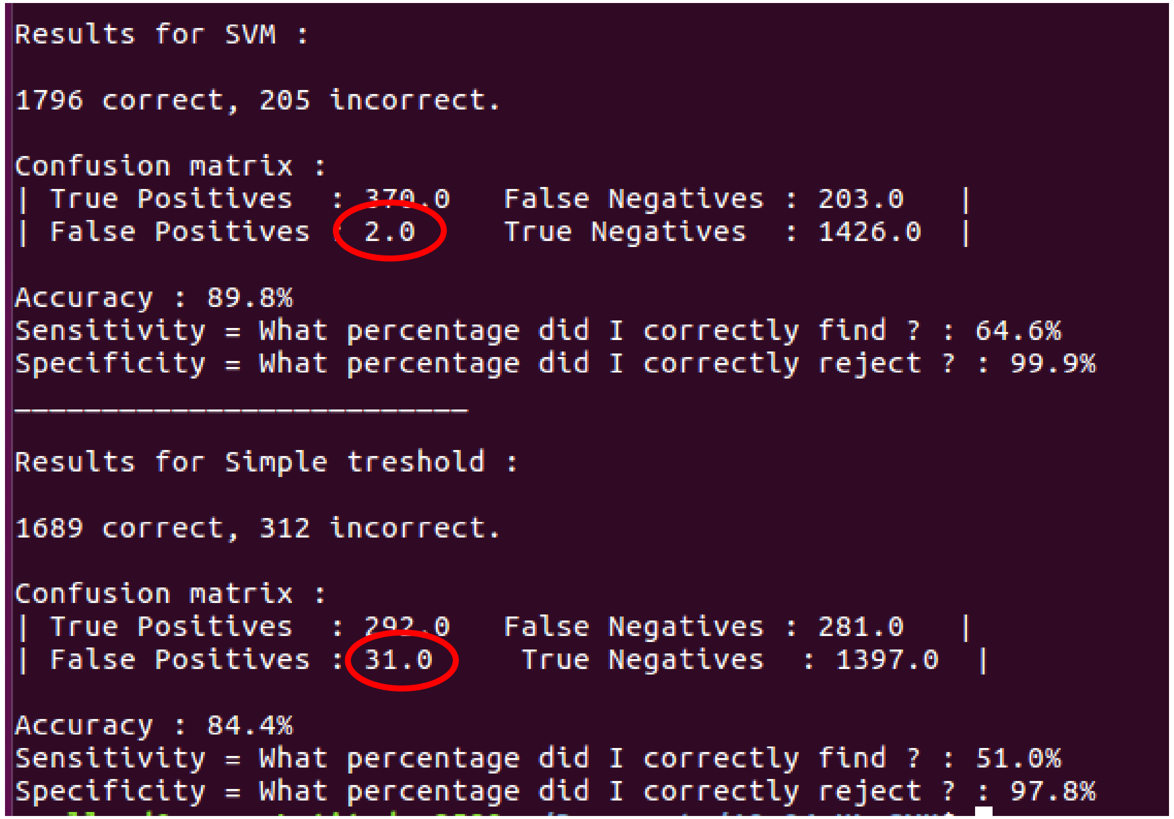
```
Results for SVM :

1796 correct, 205 incorrect.

Confusion matrix :
| True Positives  : 370.0   False Negatives : 203.0   |
| False Positives : 2.0     True Negatives  : 1426.0  |

Accuracy : 89.8%
Sensitivity = What percentage did I correctly find ? : 64.6%
Specificity = What percentage did I correctly reject ? : 99.9%
```

- Accuracy = (TP+TN)/(TP+FP+TN+FN)
- Sensitivity = TP/(TP+FN)
- Specificity = TN/(FP+TN)

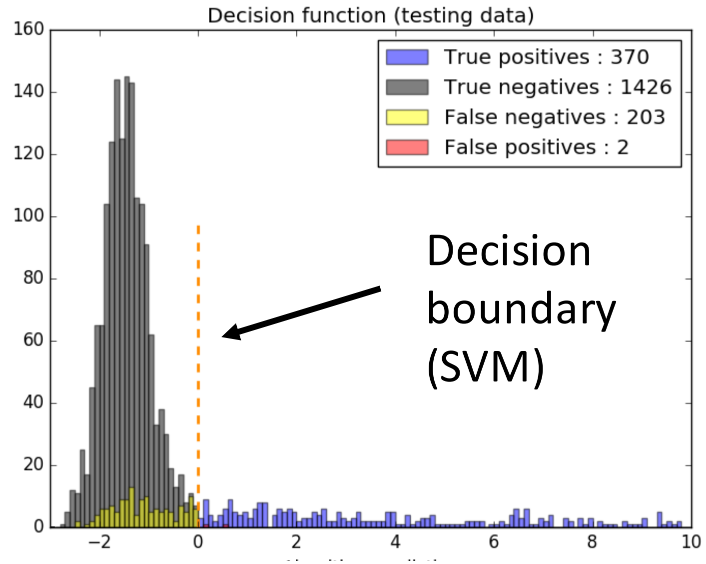
The algorithm is really good at identifying noise events!



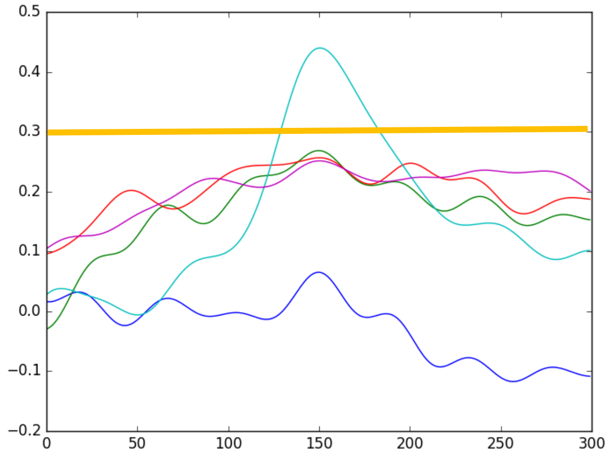
Results : comparing SVM and a treshold-based algorithm



```
Results for SVM :

1796 correct, 205 incorrect.

Confusion matrix :
| True Positives  : 370.0   False Negatives : 203.0   |
| False Positives : 2.0     True Negatives  : 1426.0  |

Accuracy : 89.8%
Sensitivity = What percentage did I correctly find ? : 64.6%
Specificity = What percentage did I correctly reject ? : 99.9%

--------------------------

Results for Simple treshold :

1689 correct, 312 incorrect.

Confusion matrix :
| True Positives  : 292.0   False Negatives : 281.0   |
| False Positives : 31.0     True Negatives  : 1397.0  |

Accuracy : 84.4%
Sensitivity = What percentage did I correctly find ? : 51.0%
Specificity = What percentage did I correctly reject ? : 97.8%
```



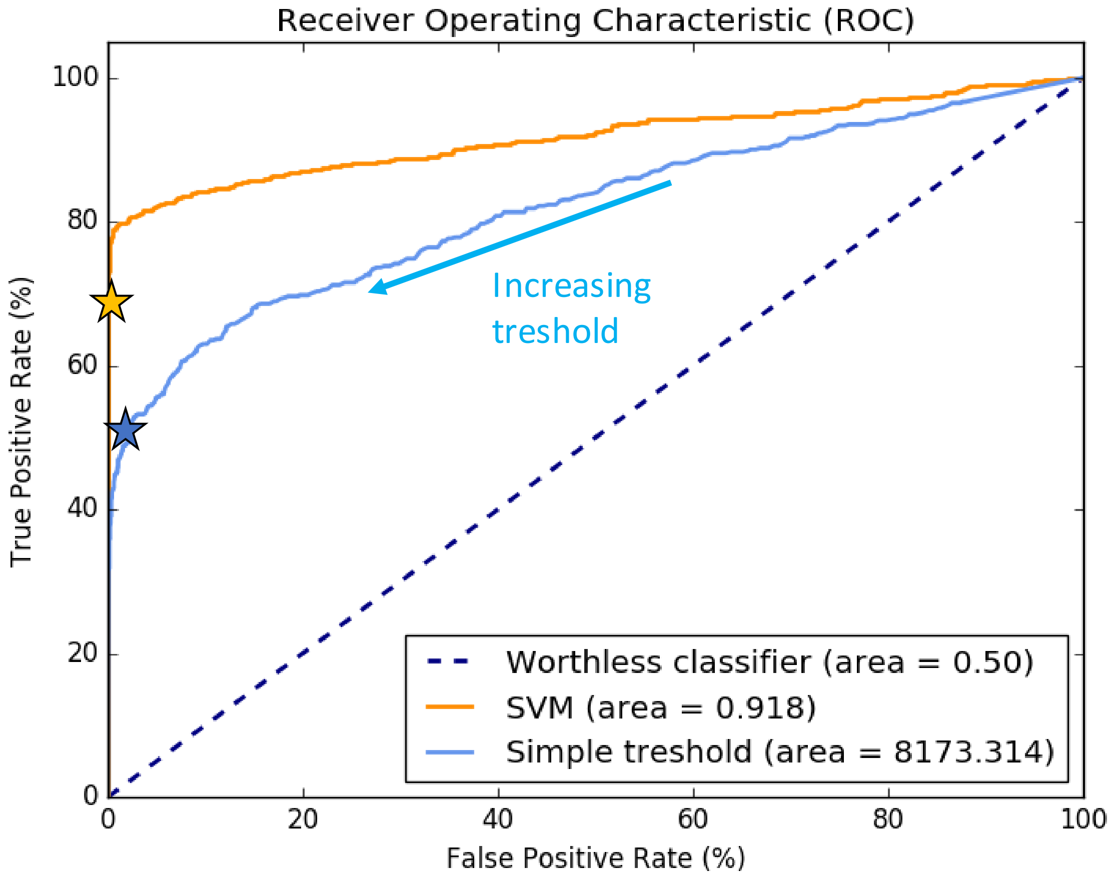
Decision boundary (SVM)



Treshold = 0.3

Results : comparing SVM and a treshold-based algorithm

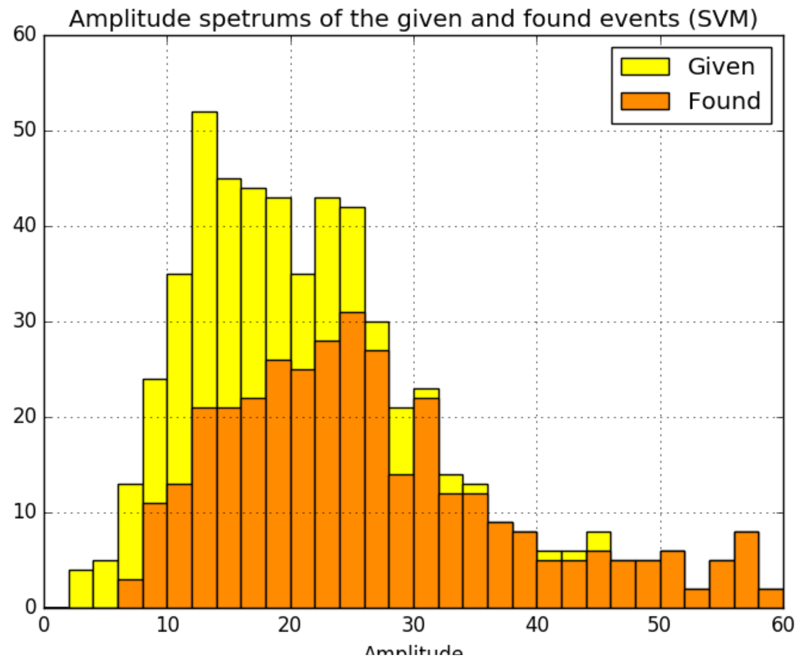
```
Results for SVM :

1796 correct, 205 incorrect.

Confusion matrix :
| True Positives  : 370.0   False Negatives : 203.0   |
| False Positives : 2.0     True Negatives  : 1426.0  |

Accuracy : 89.8%
Sensitivity = What percentage did I correctly find ? : 64.6%
Specificity = What percentage did I correctly reject ? : 99.9%

--------------------------

Results for Simple treshold :

1689 correct, 312 incorrect.

Confusion matrix :
| True Positives  : 292.0   False Negatives : 281.0   |
| False Positives : 31.0     True Negatives  : 1397.0  |

Accuracy : 84.4%
Sensitivity = What percentage did I correctly find ? : 51.0%
Specificity = What percentage did I correctly reject ? : 97.8%
```

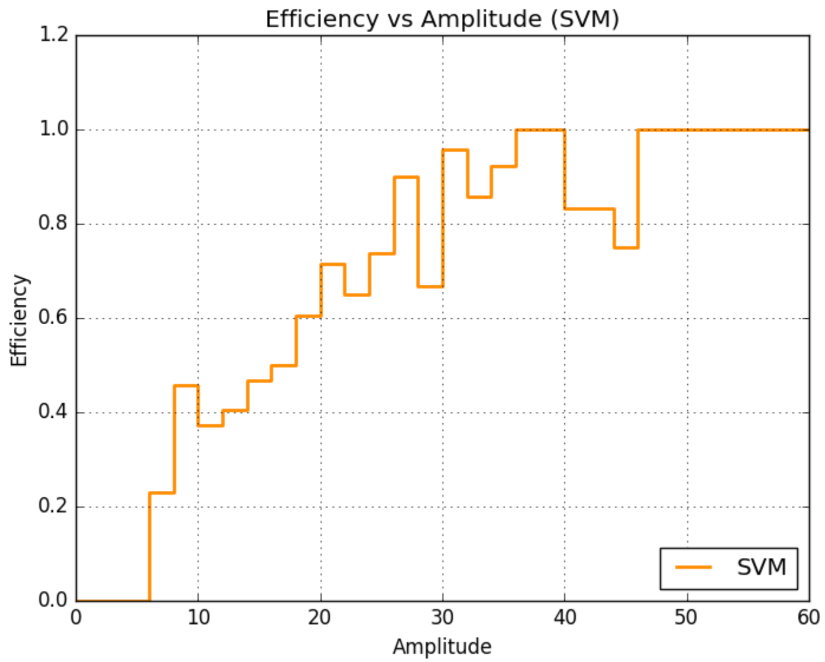


The performances of the treshold algorithm depend of course a lot on the processing so the gap could be smaller, but **the SVM algorithm gives good results as long as events from a same class (noise or 1 e-) behave the same way.**

Efficiency finding true events

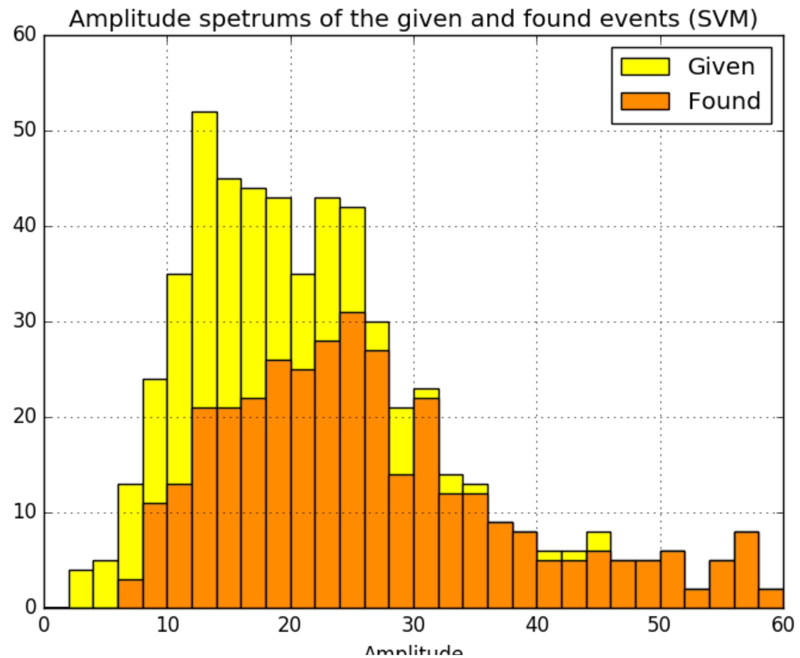
- Efficiency increasing with the amplitude

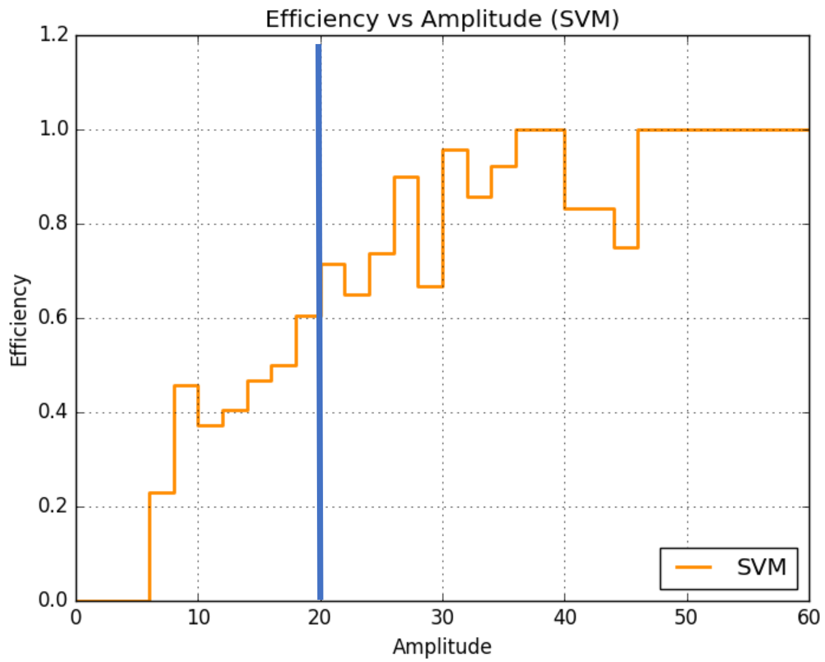




Efficiency finding true events

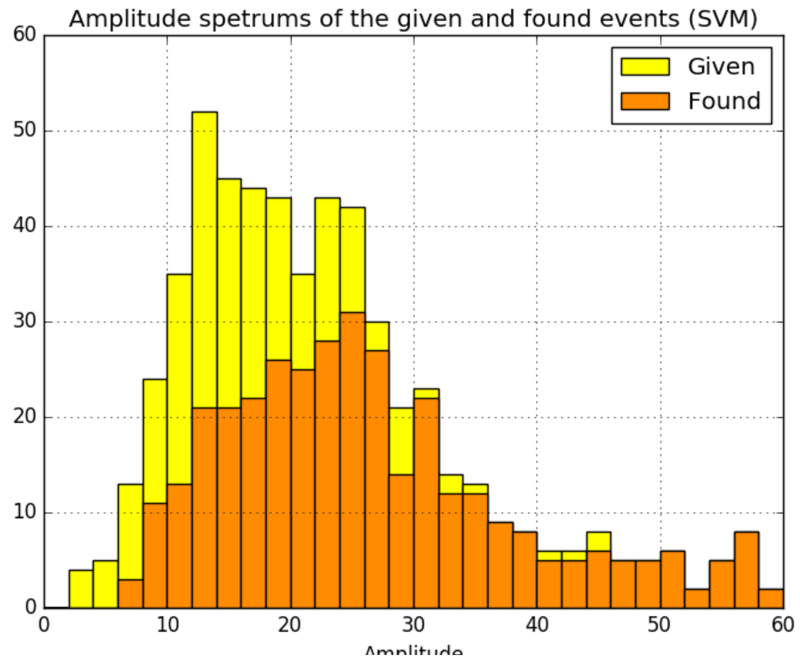
- Efficiency increasing with the amplitude
- Getting more than 80% of the events above 20 ADU

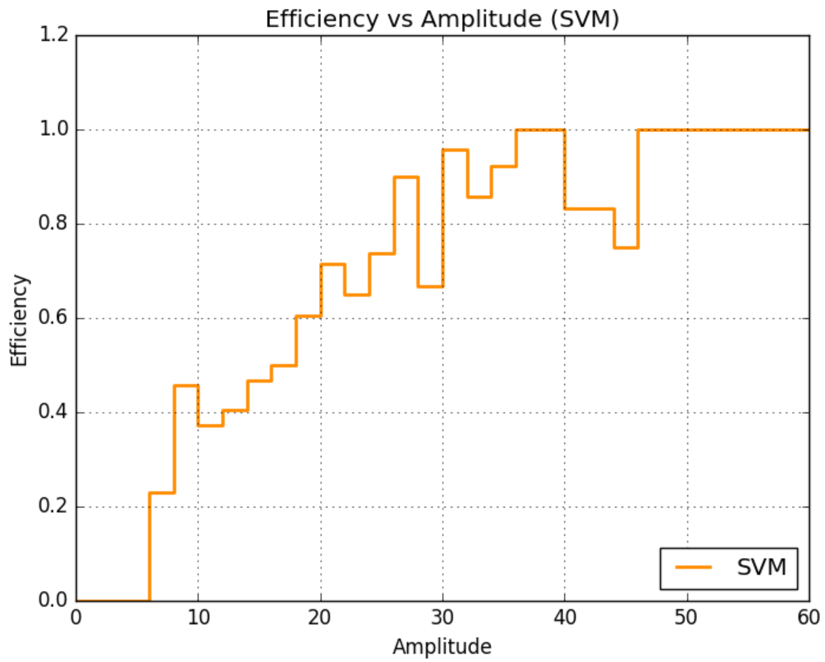


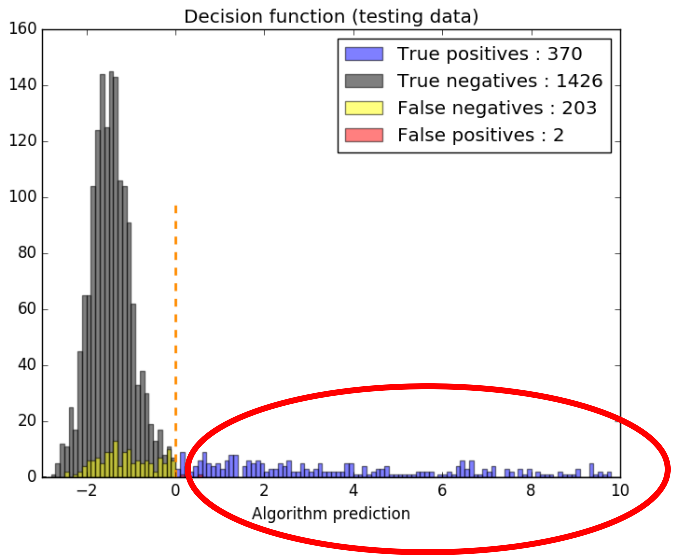


Efficiency finding true events

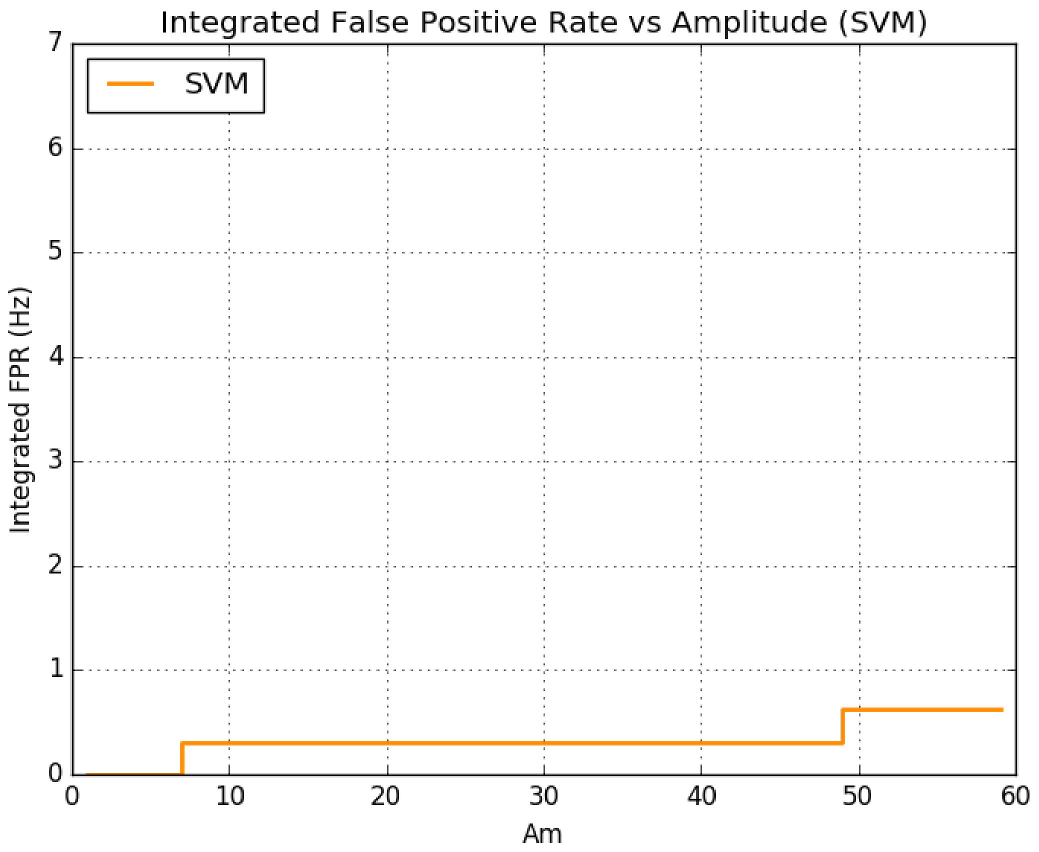
- Efficiency increasing with the amplitude
- Getting more than 80% of the events above 20 ADU
- Even if this algorithm is not based on amplitude calculation, events with higher amplitudes stand out more in the decision function







Efficiency rejecting noise events

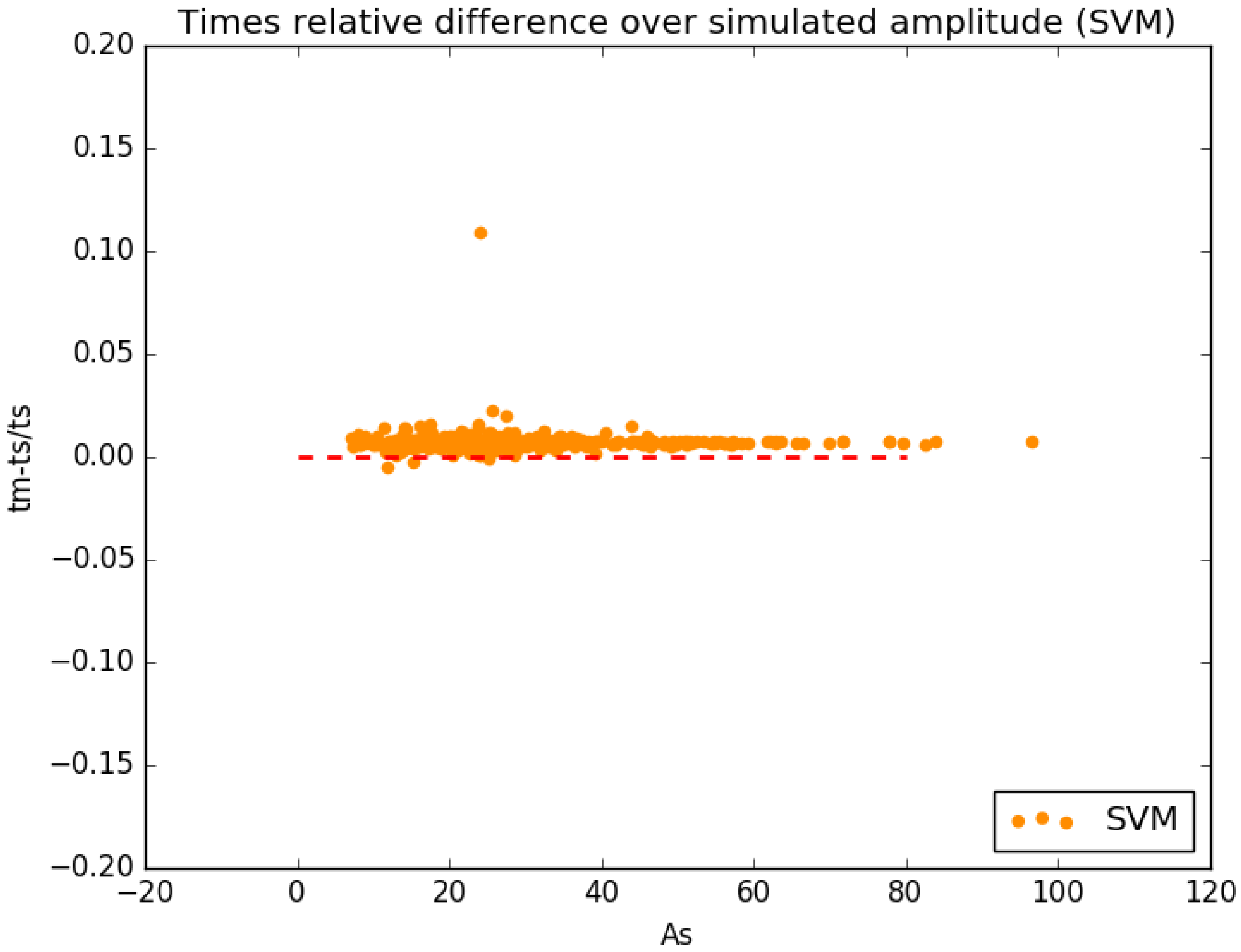


- Only 2 false positives (usually between 0 and 5) over 1428 noise events
- 2000 events of 1.6ms = 3.2s

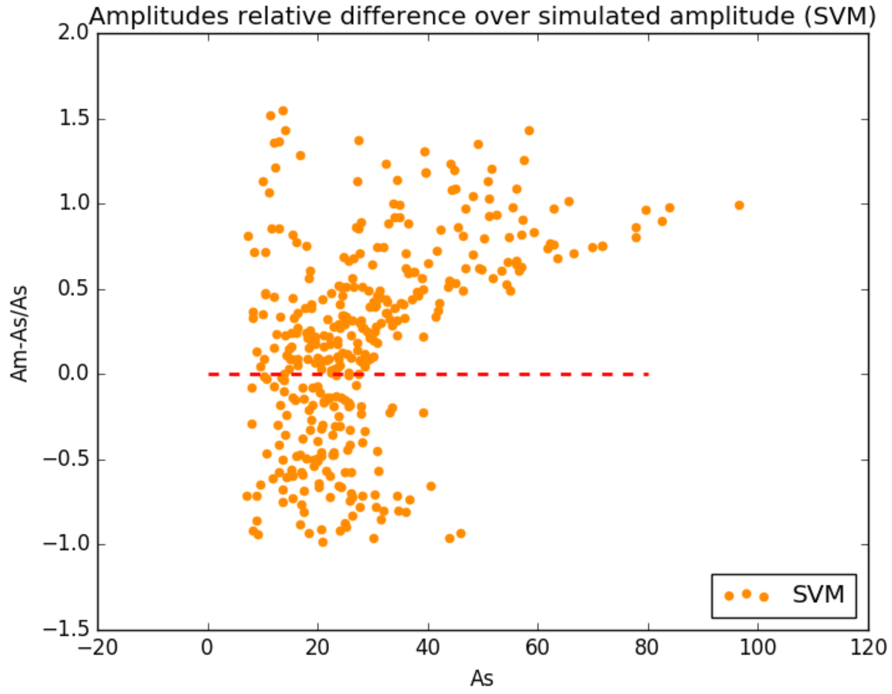
  => False positive rate ~ 1Hz
- This is an analysis classification algorithm, not a trigger

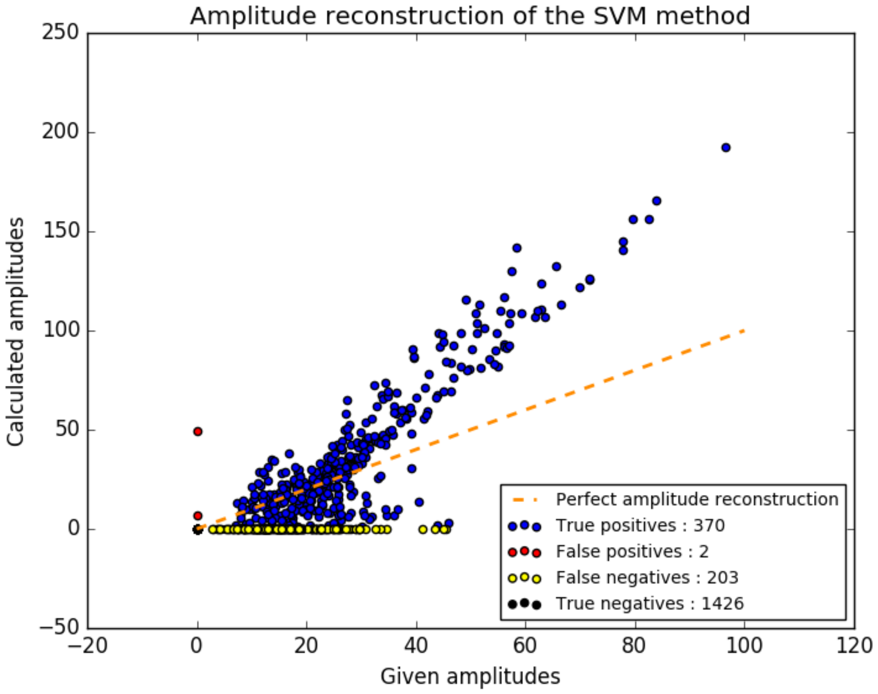
Time reconstruction

- **Very good time reconstruction**
- Proving that centering the pulse in the processing works well
- Could correct the little offset quickly



Amplitude reconstruction





The **amplitude reconstruction** is terrible because the SVM algorithm never needs to calculate amplitudes : it is just comparing the pulses shapes.

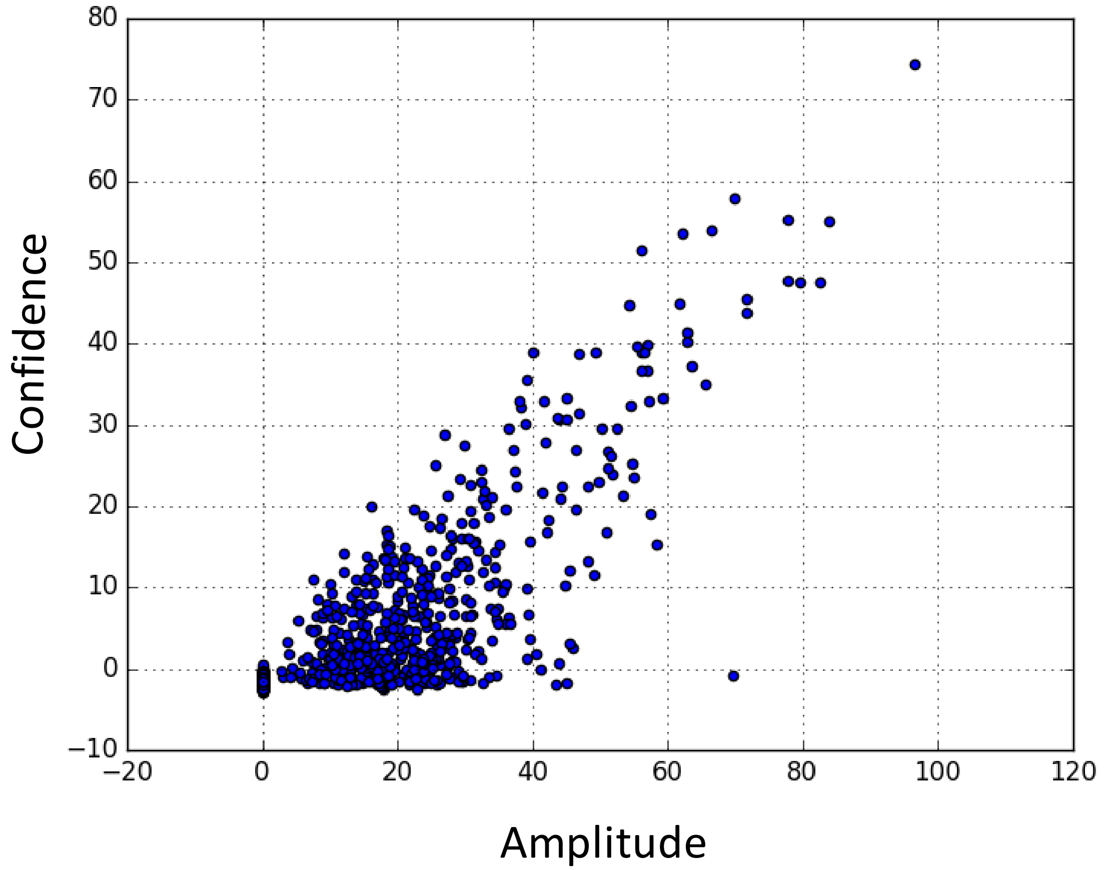
Conclusion

- The SVM algorithm is excellent at identifying noise events, therefore it's a very powerful analysis tool for low-energy studies.
- We manage to keep **~65%** of the true events when rejecting **99.9%** of the noise, which should allow us to **lower our threshold down to 20 ADU** (27eV on SEDINE 3bat Ne data) in the trigger algorithm, even if that makes us trigger on a lot of noise events.
- This is just a classification algorithm, not well suited for amplitude reconstruction. It would be better to process normally after sorting the events.
- Better results could be obtained with different improvements :
    - Training on more data (100k?). Need a more powerful computer.
    - Better processing and time reconstruction to make the events stand out more
    - Tune the decision boundary according to the desired false positive rate (also depends on the trigger algorithm!!)

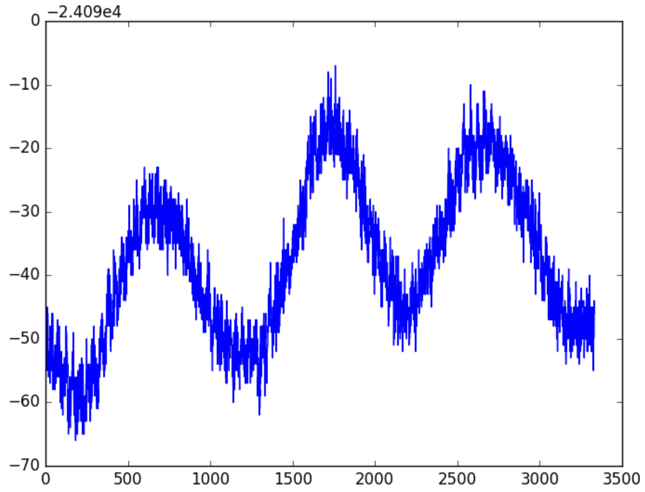
- Thank you!-

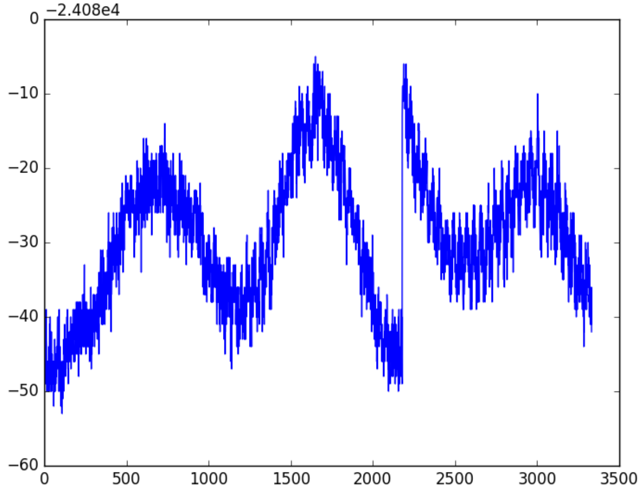
Backup slides

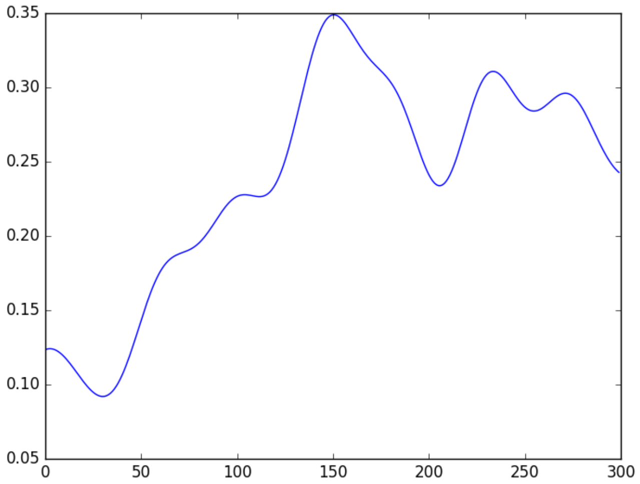
Amplitude vs Confidence correlation

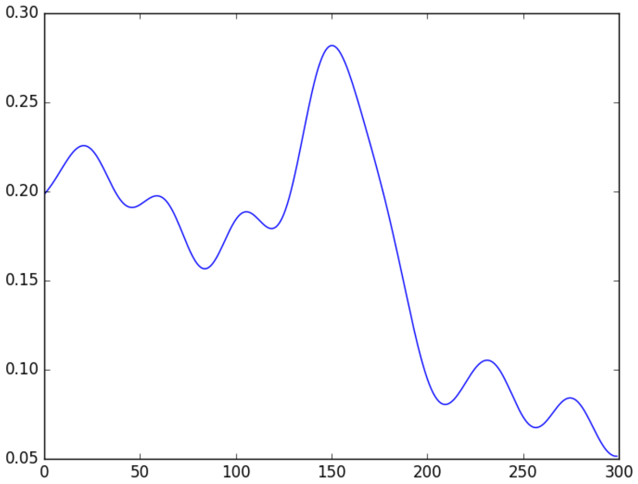


False positives events

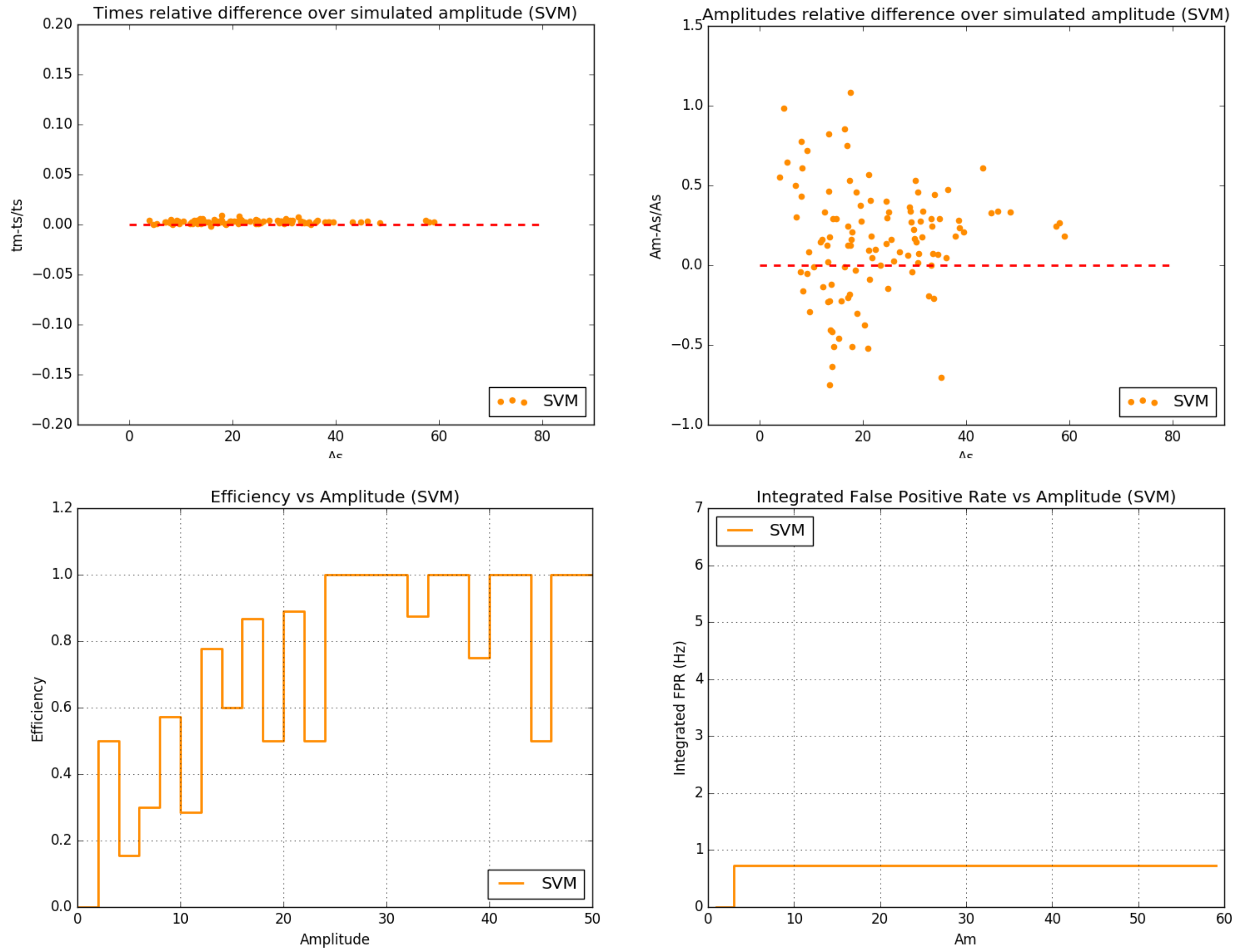






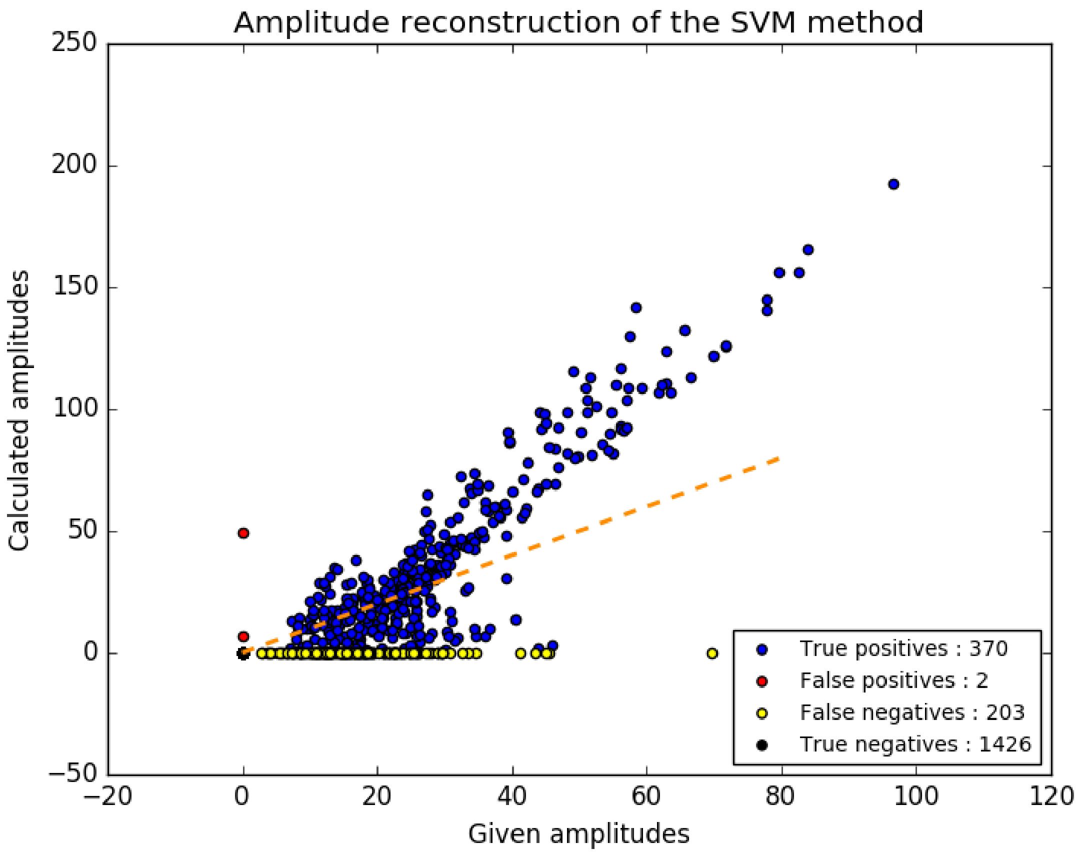


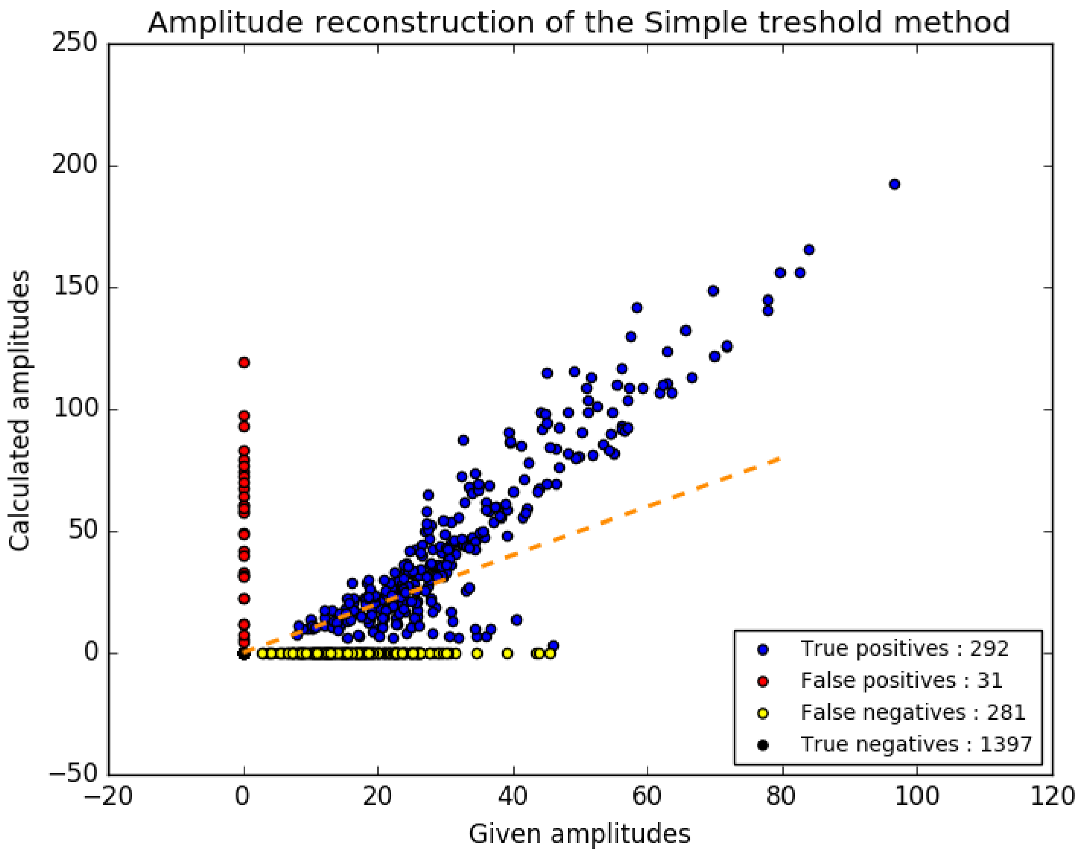
Results on the 3444 S30 events



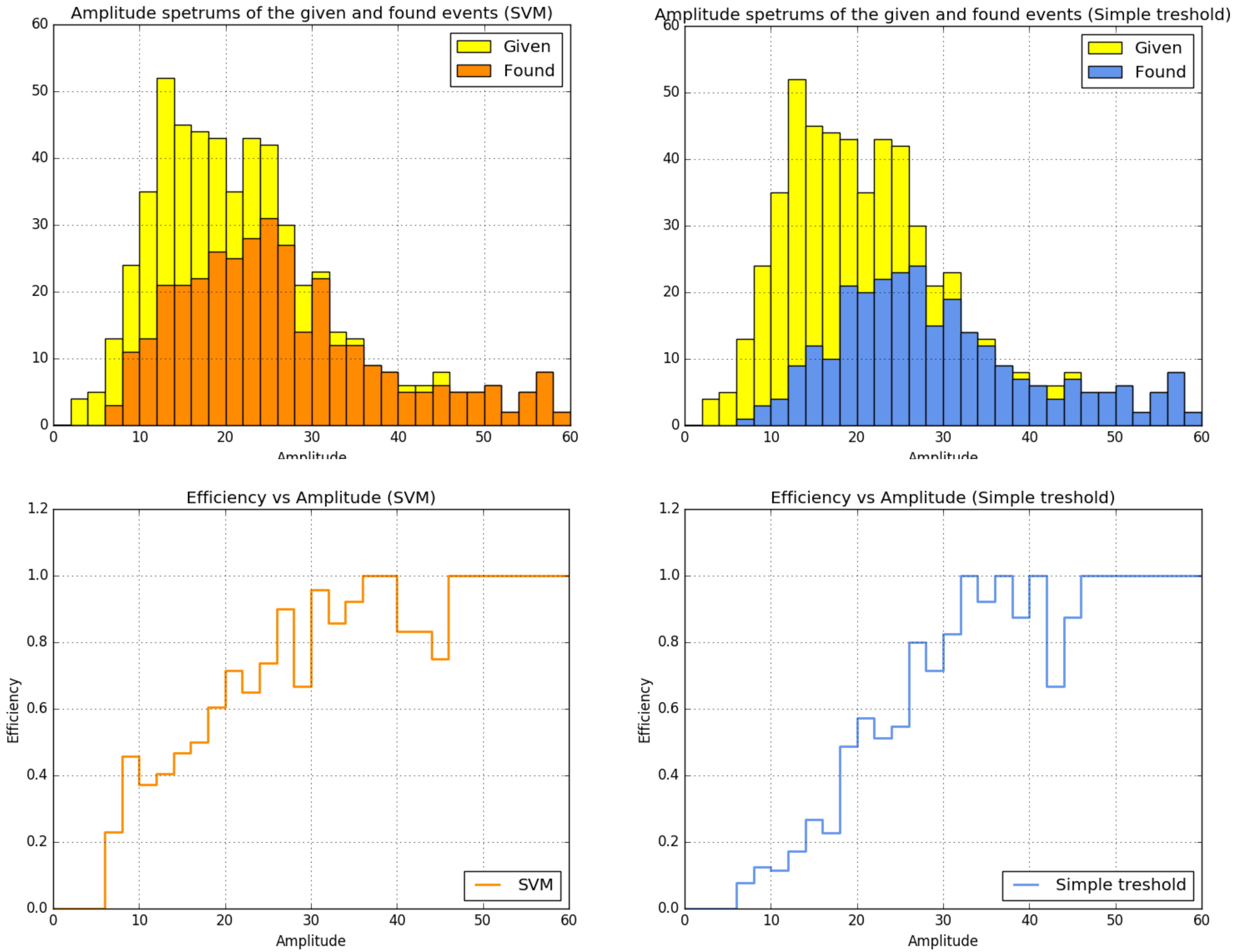
- Very good on time reconstruction
- Terrible on amplitude reconstruction
- Too little statistics for a clean Efficiency vs Amplitude plot
- Very low FPR

Comparison SVM vs treshold

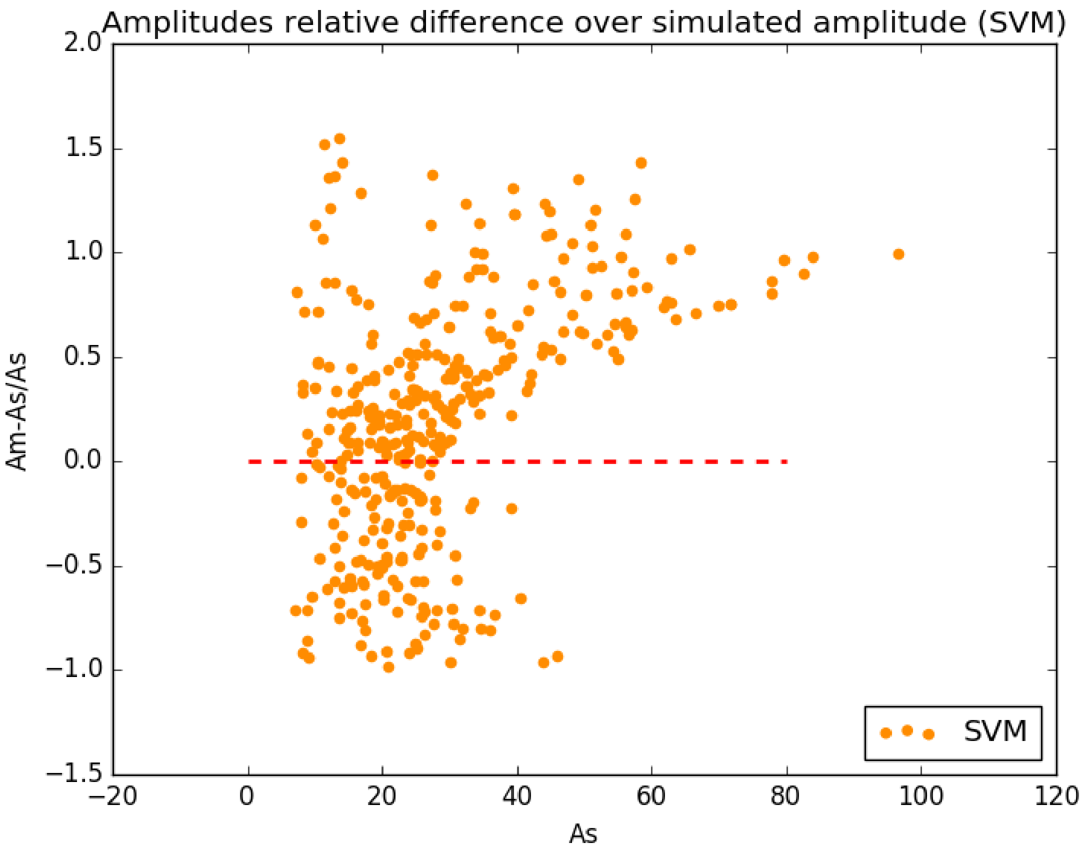


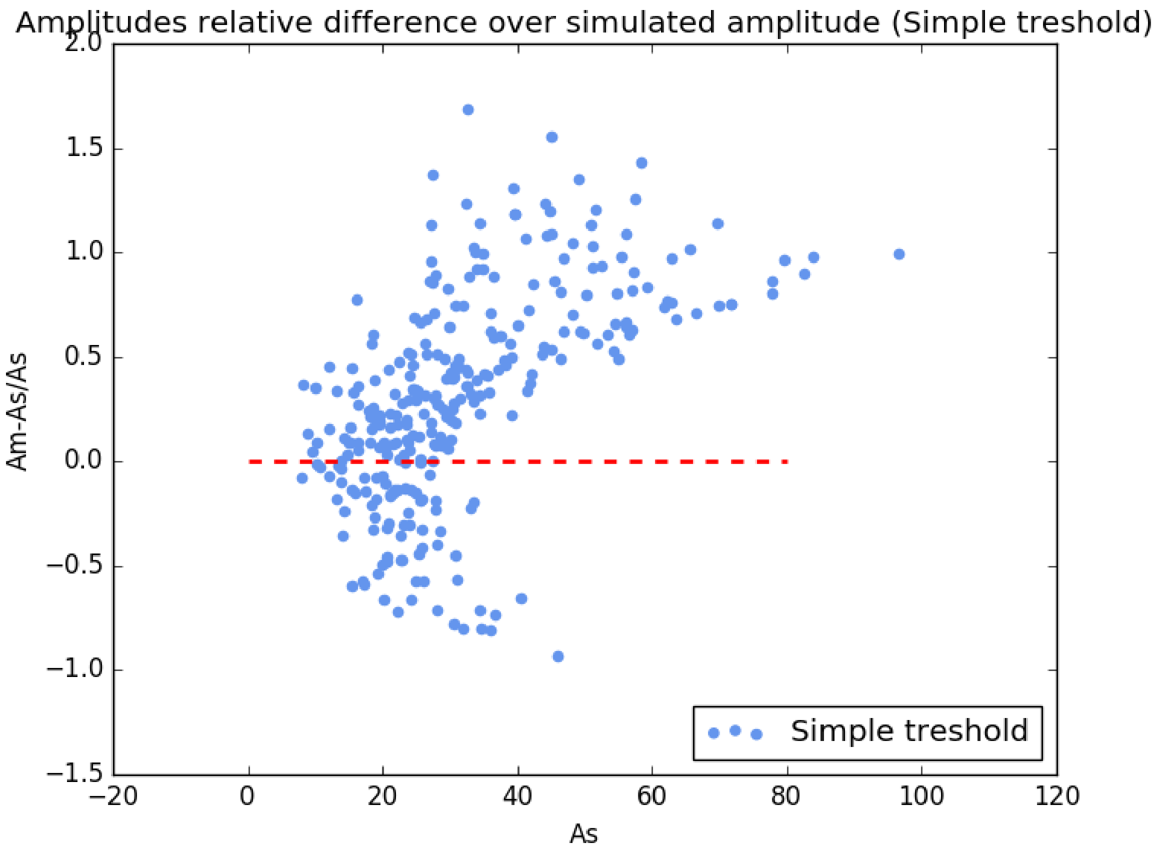


Results on the 3444 S30 events

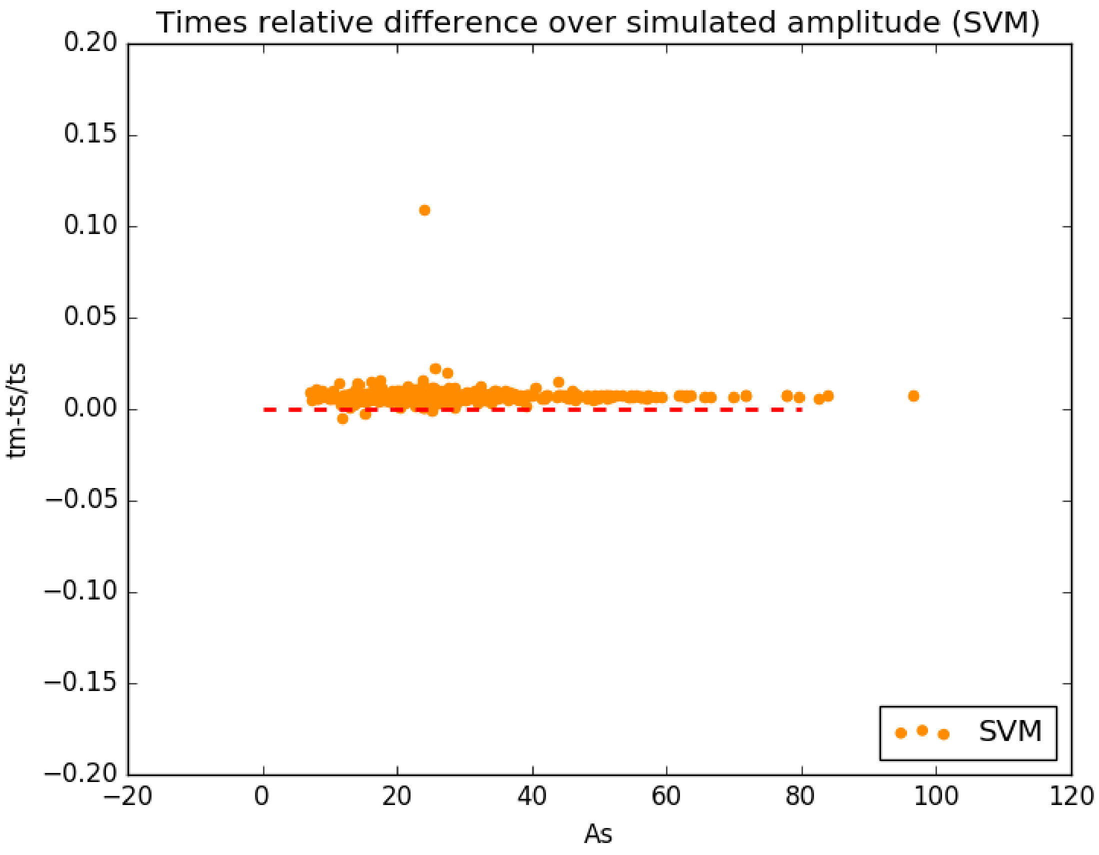


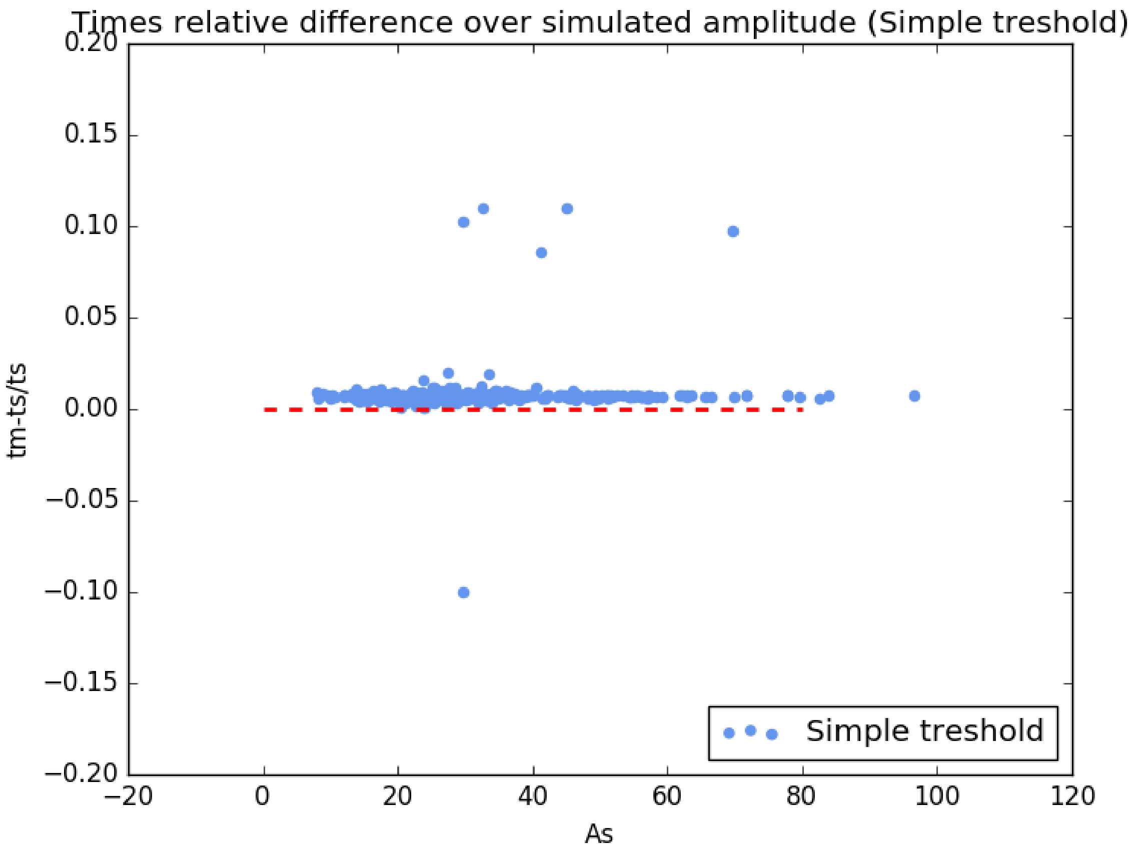
Comparison SVM vs treshold





Comparison SVM vs treshold





Comparison SVM vs treshold