Aubio Study Results

Domenico Stefani

26 February 2021

Abstract

This brief report presents the results of a comparison study on all the methods of the Aubio suite for onset detection.

Some choices are strongly affected by the target application for the onset detector.

1 Introduction

The study is performed on sounds recorded from 6 combinations of different acoustic guitars and experienced guitarists. 8 different techniques were used to play the sounds, 4 of them being conventional techniques played with a pluck/pick on the strings, while the remaining 4 are percussive sounds produced by hitting the body of the guitar.

Aubio onset offers 8 different detection methods and different parameters. The methods are:

1. hfc.
2. energy.
3. complex.
4. phase.
5. specdiff.
6. kl.
7. mkl.
8. mkl (without adaptive whitening).
9. specflux.

The parameters are:

- *Hop size:* the value used was **64 samples** since this is a value for the audio block size for the vst plugin that offers a good trade off between latency and computational resources requirements on Elk Audio Os (Running on Raspberry Pi 4).
- *Minimum Inter-Onset Interval*: Debounce parameter which was set to **20ms** as it is the target latency for the whole classification pipeline in mind for this study.
- *Buffer Size*: This represents the size of the analysis window for which the onset detection function is computed. Different values were tested: **64,128,256,512,1024,2048 samples**. This affects in minor way the detection latency.
- *Silence Threshold*: It is used to cut low amplitude onsets. It's expressed in dB and values between **-60dB** and **-40dB** were used.
- *Onset Threshold*: A value used in dynamic thresholding to select onsets. Values between **0** and 4 were used.

2 Test

For each method, different combinations of the parameters were tested to optimize the f1-score. Since the cardinality of examples for each class does not reflect the one of the full dataset and it is not equal between classes, the metric optimized was the **macro average f1-score**, which is the mean of the f1-scores for each individual class. Similarly, all the latency metrics reported are averaged across all classes. For latency, upper and lower Tukey fences are reported (k = 1.5) along with the sample mean of the distribution.

The 2 objectives to optimize, in order to choose the combination of method and parameters to use, are the f1-score (to maximize) and the variability of the latency, represented by the Interquartile Range (to minimize). A flexible constraint can be posed on the upper fence which should not exceed a set value. In this regard, the pareto front is found with the 2 main objectives first, and then with the f1-score and upper fence as objectives.

3 Improvements and Adaptive Whitening

Once all the methods were tested and optimized, with their default Aubio initialization and several combinations silence threshold and onset threshold values, more research was devoted to improving the best performance, which was around 95% f-measure when using the *specdiff* method and buffer sizes of 256 and 512 samples.

Several advanced methods involved end-to-end deep neural network training or the use of DNNs only for the onset detection part: both approaches are interesting but beyond the aim of this research, since severe testing will need to be performed in order to ensure that these networks can work with very tight time constraints on an embedded device (with satisfactory classification performances). Moreover, many of the solutions presented as state of the art in the field of *online* onset detection accept very loose time constraint (±25ms or even ±50ms between labels and detected onsets). Similar approaches will be studied and applied to this problem in future studies.

3.1 Failed attempts

Different naive approaches were tested in order to improve the f-measure, including the use of a **Noise Gate** (to cut signal noise in silent sections), **Dynamic Range Compression** (to even out performances across different playing dynamics) and **Highpass filtering** (with and without mixing the filtered signal with the original one). Different parameters where used for all the aforementioned approaches, however performances only improved in the worst cases (cases where f1 was 85% reached even 90 or 92%) while the best cases were not improved (in some case the effects were detrimental). An exhaustive search in the space of effect parameters could entail better results, however the results with all the combinations devised were not promising.

3.2 Adaptive Whitening

Adaptive whitening[1] is an interesting technique for online normalization of frequency-bin-magnitude that was successfully applied to the family of Onset Detection methods that is also implemented in Aubio. The creators of the method show performance improvements of different entity depending on the application domain and the method used. Improvements can be classified from mild to significant, with the exception of the Modified Kullback-Leibler divergence (MKL) method [2], which was show by the authors to perform the best in many cases **without Adaptive Whitening** (while using AW entailed worse performances).

Looking into possible open source implementations of the method proposed by the authors, it was discovered that Brossier already implemented it in version **0.4.5** of Aubio, however it's applied by default on initialization of the onset object depending only on the method used. This setting is only available from the library version of aubio, while the aubioonset program offers no way of controlling this. Most importantly, from version 0.4.5 to the latest stable version of aubio[1], Adaptive Whitening is applied by default to the **MKL** method, which was shown to perform worse with it.

After modifying the aubio version to avoid using AW with MKL, the same previous tests were performed with this version of MKL with produced the best results yet, already reaching 95.11% f-measure with very low latency at 64 samples for the buffer size. In the other cases performances were improved of 2 and 5 percentage points over the previous best. Other methods didn't show these improvements when disabling AW.

[1] As of March 8th 2021

4 Results

The best f1-score results along with the latency metrics connected to them are presented in tables 1 and 2.

Table 1: The best f1-score avg. values are shown. Different combinations of Buffer size and Method produce different latency values, which are reported in the following tables. Bold values represent the points in the Pareto front defined by the points in the space of 2 objectives: the f1-score (to maximize) and the Inter Quartile Range (to minimize). More info in fig. 1 and table 3.

		Buffer size					
		64	128	256	512	1024	2048
Method	hfc	0.9341	0.9229	0.9120	0.8962	0.8993	0.8775
	energy	0.9364	0.9419	0.9470	0.9444	0.9533	0.9164
	complex	0.8351	0.8422	0.8579	0.8720	0.8623	0.7909
	phase	0.7587	0.8227	0.8742	0.8178	0.7421	0.7160
	specdiff	0.8524	0.9330	0.9511	0.9523	0.9528	0.9294
	kl	0.8532	0.8610	0.8658	0.8702	0.8919	0.9079
	mkl	0.8482	0.8522	0.8718	0.8661	0.8690	0.8706
	mkl (No whitening)	**0.9511**	0.9702	**0.9764**	0.9731	0.9732	0.9613
	specflux	0.8446	0.9185	**0.9255**	0.9067	0.8799	0.8659

Table 2: The results of the latency recorded on the examples which f1-score is reported in table 1 are shown here. Each cell contains 3 values: the first and the last are the lower and upper Tukey fences with $k = 1.5$, which are defined starting from the Interquartile range and are commonly used to define outliers of a distribution, while the central value is the sample mean of the latency distribution.

		Buffer size					
		64	128	256	512	1024	2048
Method	hfc	2.4/4.6/6.6	2.7/5.0/7.1	3.7/6.0/8.0	5.0/7.4/9.6	9.1/11.6/14.1	11.2/13.9/16.4
	energy	1.7/3.9/6.1	2.7/4.9/7.0	3.9/6.0/8.0	5.7/7.8/9.8	9.0/11.6/14.1	12.0/15.4/18.9
	complex	2.6/5.0/7.2	3.1/5.4/7.5	3.5/6.0/8.3	4.0/6.6/9.0	4.6/7.5/10.3	5.9/8.7/11.4
	phase	0.5/2.9/4.7	2.1/4.3/6.3	2.8/4.7/6.6	3.4/5.2/7.1	3.9/5.9/7.9	4.5/7.2/10.0
	specdiff	2.4/4.5/6.4	2.8/4.7/6.6	3.8/5.9/7.9	4.9/7.1/9.2	6.6/9.1/11.8	7.4/12.6/18.1
	kl	1.9/4.2/6.3	2.4/4.6/6.7	3.3/5.6/7.8	5.2/8.0/10.6	8.3/12.2/16.4	13.3/16.4/19.5
	mkl	2.3/4.6/6.7	2.9/5.3/7.5	3.8/6.4/8.8	5.4/8.2/11.0	8.5/11.0/13.6	11.4/14.5/18.1
	mkl (No whitening)	2.8/4.6/6.3	3.2/5.1/6.8	4.2/6.0/7.7	5.3/7.4/9.2	8.3/10.9/13.3	10.5/13.7/16.5
	specflux	2.1/3.8/5.3	2.5/4.3/5.8	3.1/4.8/6.3	3.8/5.6/7.3	4.4/6.3/8.1	5.3/7.5/9.5

Pareto front results are shown in table 3 and fig. 1 for the first analysis, and in table 4 and fig. 2 for the second one.

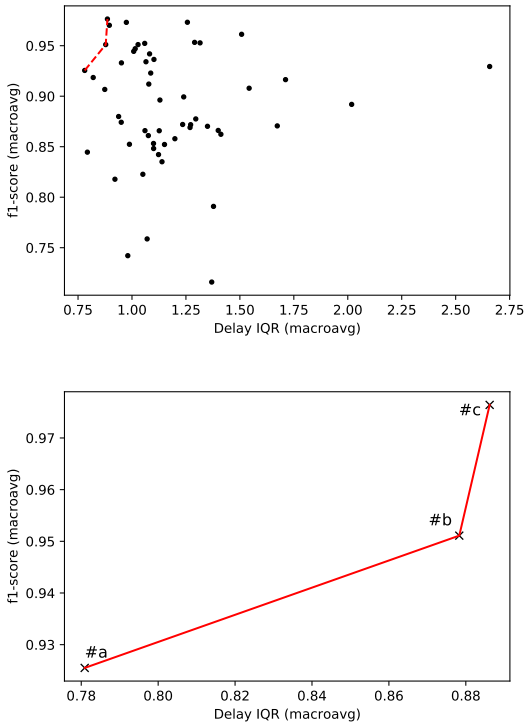

Figure 1: Pareto front computed for f1-score and the Interquartile Range of the latency distribution. The upper plot shows all the solution while the lower plot represents only the points in the front. The labels refer to the detailed information that can be found in table 3.

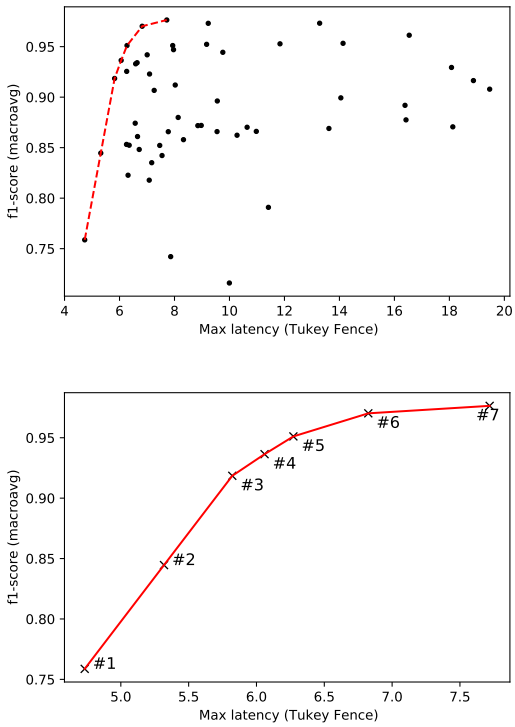

Figure 2: Pareto front computed for f1-score and upper Tukey fence. The upper plot shows all the solution while the lower plot represents only the points in the front. The labels refer to the detailed information that can be found in table 4.

Table 3: Pareto front solution with f1-score (macro average over all techniques) as the first objective and Interquartile Range of latency as the second.

#	Method	F1-score	Low Tukey fence (ms)	Delay mean (ms)	High Tukey fence (ms)	Onsets inside fences (%)	MAvg_t IQR
a	specflux	0.9255	3.1375	4.7744	6.2610	94.64	0.7809
b	mkl (No whitening)	0.9511	2.7591	4.5746	6.2722	96.90	0.8783
c	mkl (No whitening)	0.9764	4.1744	6.0215	7.7192	95.70	0.8862

5 Solution Choice

At this point, all the solutions in the Pareto front computed for IQR and f1-score (fig. 1 and table 3) are viable non-dominated solutions which offer different trade-offs between f1 performance and latency variance. In particular, solution #c from the first pareto front (which corresponds to #7 in the second) provides the best f-measure value, while still having low variance and a low maximum in the distribution of detection latency.

Because of this, the MKL method (without whitening) and with buffer size 256 was considered the best solution.

Table 4: Pareto front solution with f1-score (macro average over all techniques) as the first objective and maximum latency as the second, in the form of upper Tukey fence.

#	Method	F1-score	Low Tukey fence (ms)	Delay mean (ms)	High Tukey fence (ms)	Onsets inside fences (%)
1	phase	0.7587	0.4530	2.8542	4.7337	93.90
2	specflux	0.8446	2.1449	3.7673	5.3183	95.44
3	specflux	0.9185	2.5403	4.2609	5.8217	95.41
4	energy	0.9364	1.6527	3.9025	6.0597	97.73
5	mkl (No whitening)	0.9511	2.7591	4.5746	6.2722	96.90
6	mkl (No whitening)	0.9702	3.2442	5.0959	6.8244	96.74
7	mkl (No whitening)	0.9764	4.1744	6.0215	7.7192	95.70

Table 5: Non dominated solution of choice.

#	Method	Buffer Size	Hop size	Min IOI (s)	Silence Thresh. (dB)	Onset Thresh.	F1-score	Low Tukey fence (ms)	Delay mean (ms)	High Tukey fence (ms)	IQR (ms)	Stdev (ms)
c/7	mkl (No whitening)	256	64	0.02	-53	1.21	0.9764	4.2	6.0	7.7	0.89	0.92

6 Appendix

Here the Pareto plots are reported with solutions from each method highlighted.

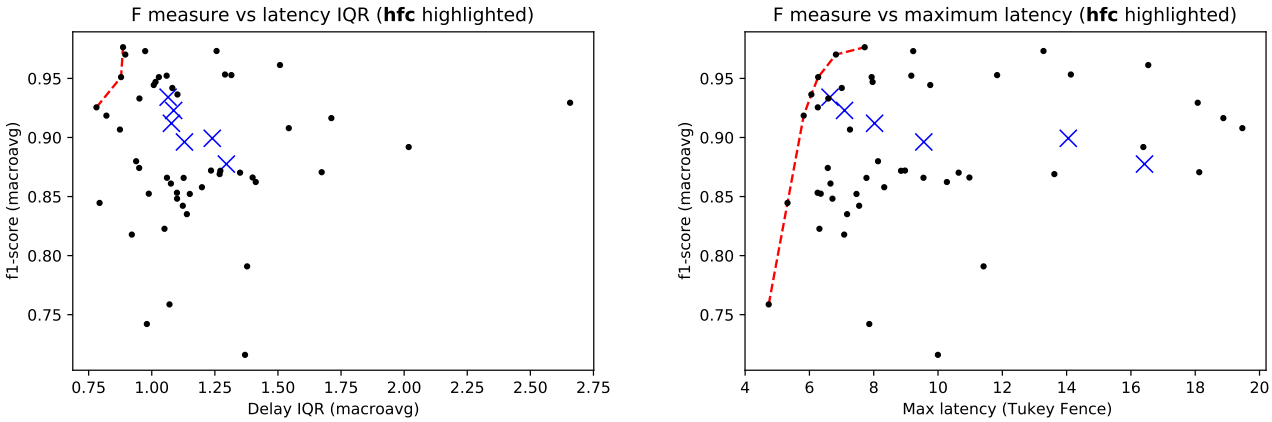

Figure 3: Solutions with hfc

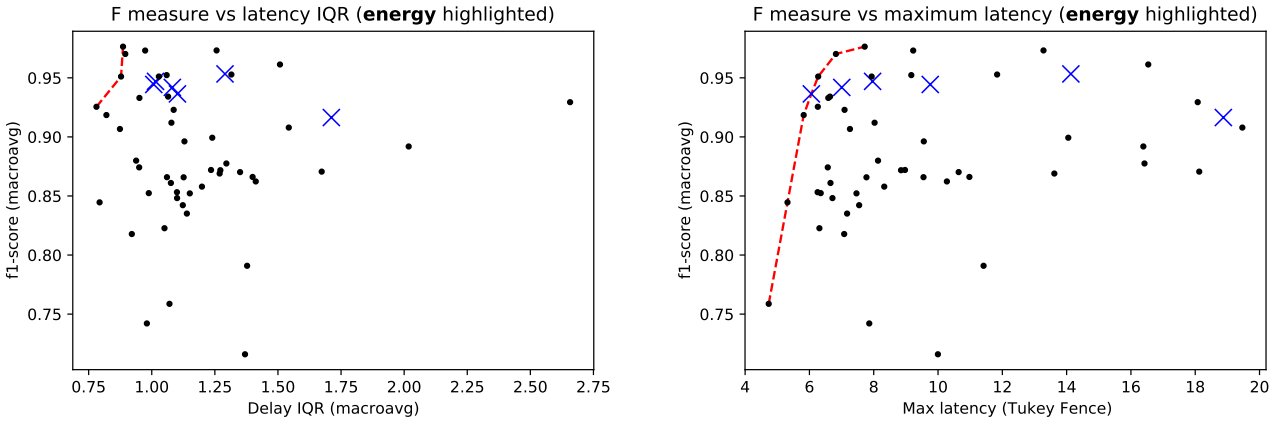

Figure 4: Solutions with energy

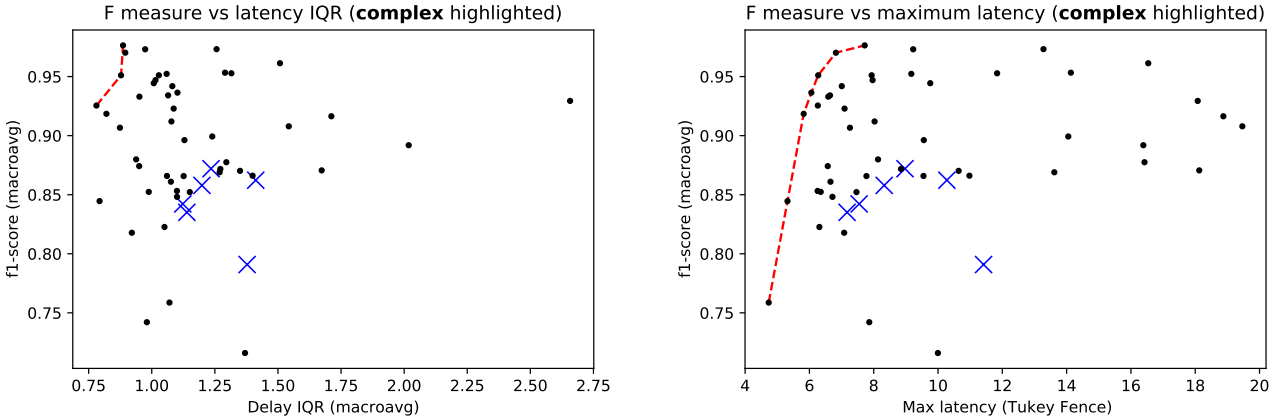

Figure 5: Solutions with complex

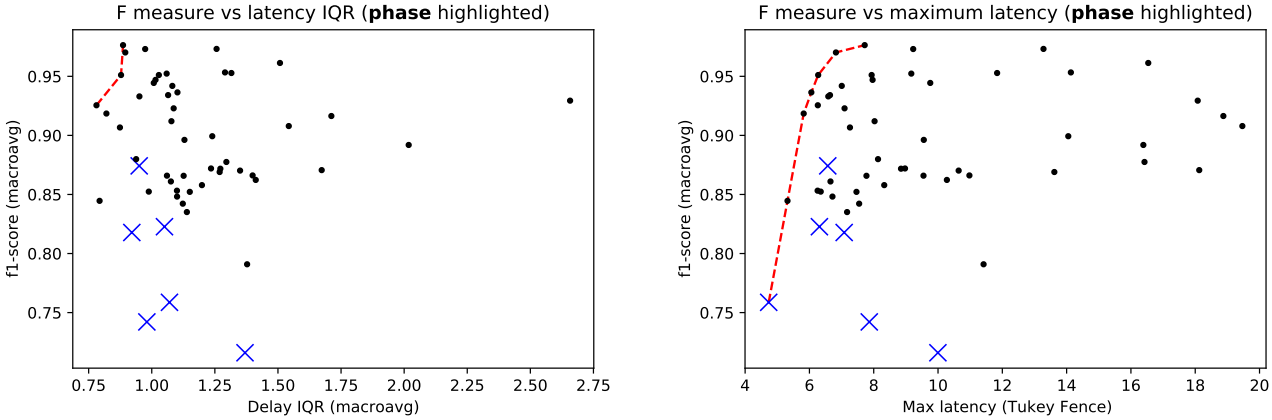

Figure 6: Solutions with phase

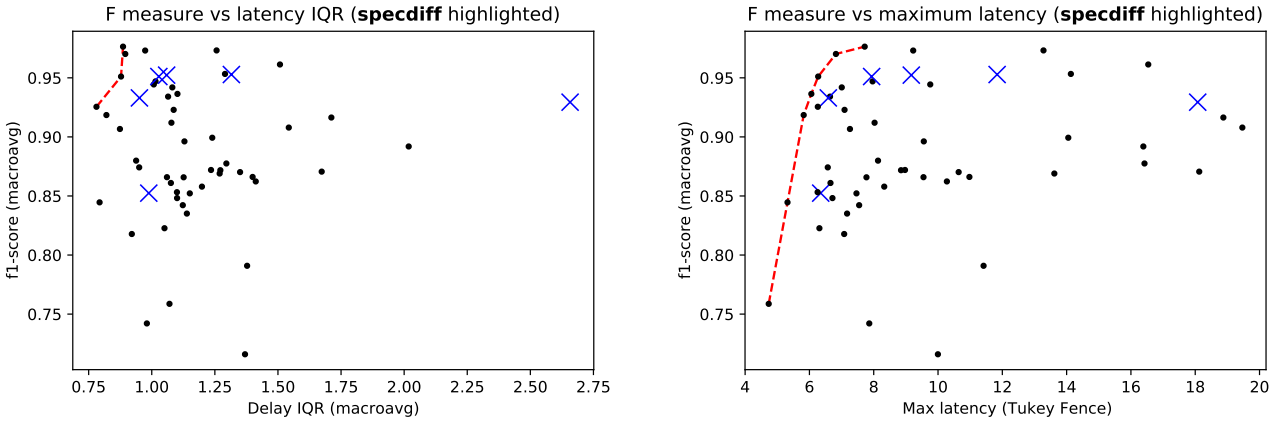

Figure 7: Solutions with specdiff

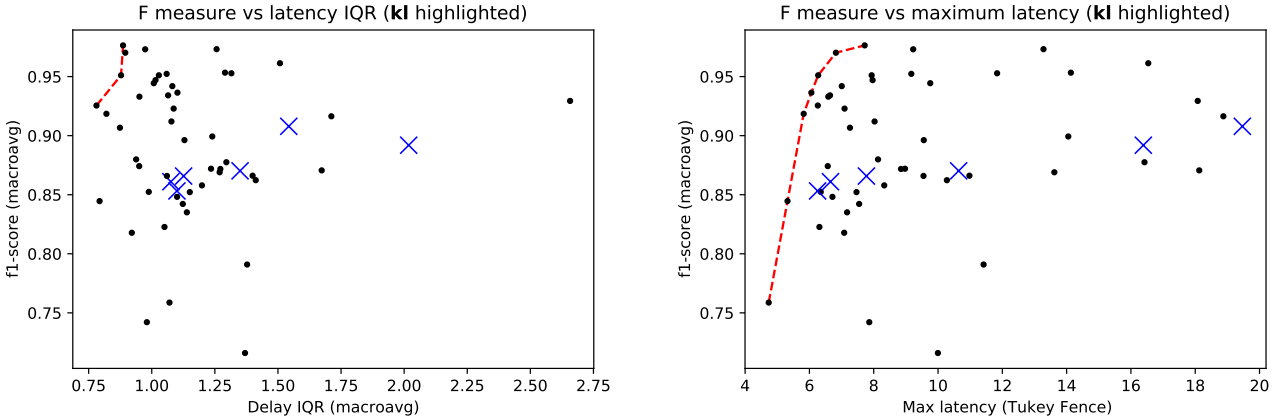

Figure 8: Solutions with kl

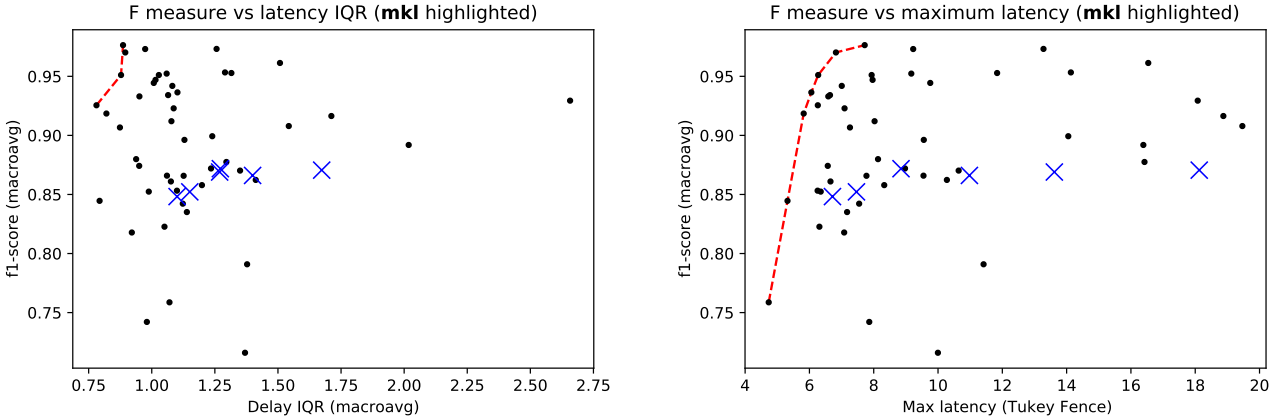

Figure 9: Solutions with mkl

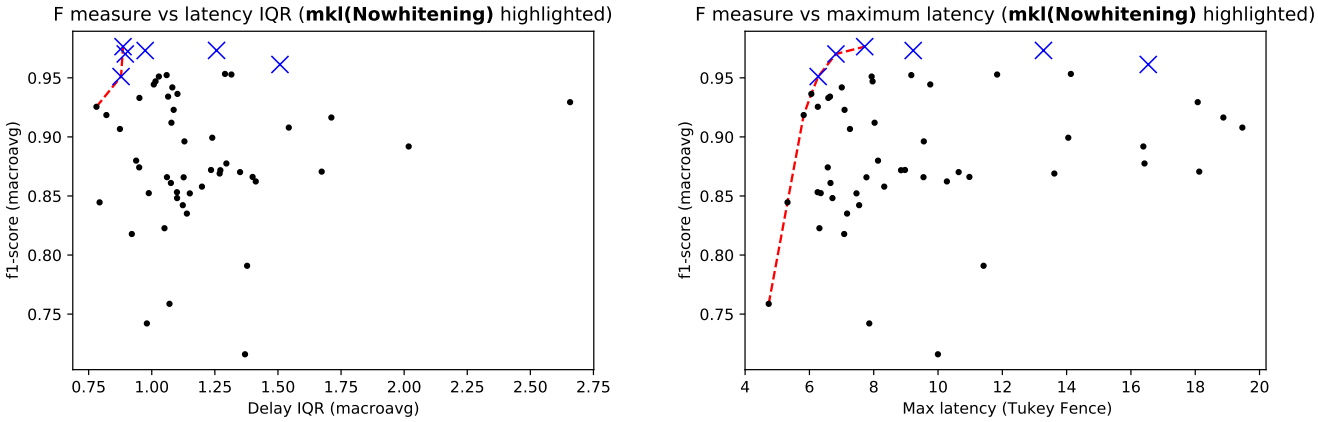

Figure 10: Solutions with mkl (No whitening)

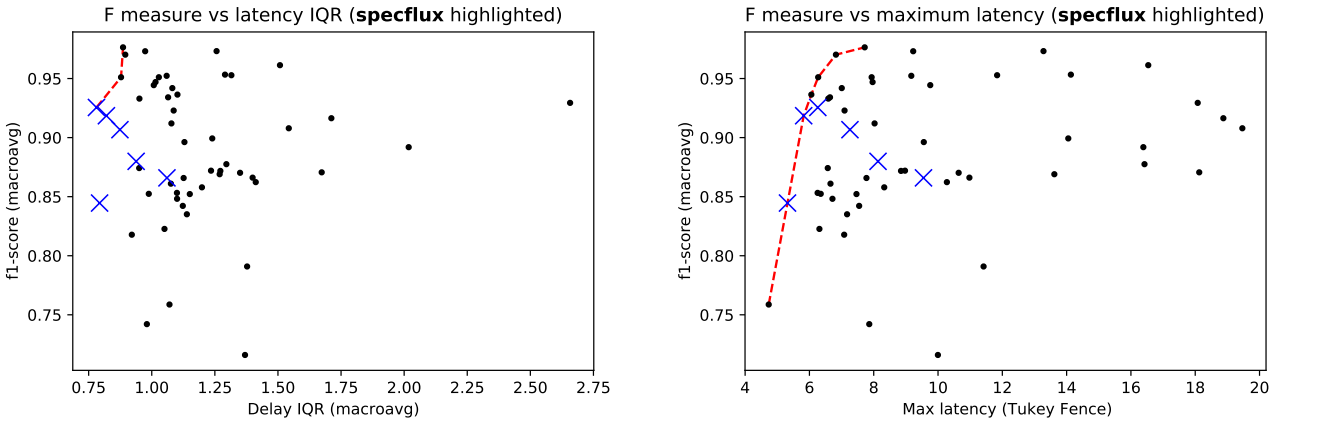

Figure 11: Solutions with specflux

References

[1] Dan Stowell and Mark Plumbley. Adaptive whitening for improved real-time audio onset detection. In *Proceedings of the 2007 International Computer Music Conference, ICMC 2007*, pages 312–319, 2007.

[2] P. Brossier. Automatic annotation of musical audio for interactive applications. 2006.