Plan

Introduction

By creating DeteClic, we provide a user-friendly click detector, based on robust analysis of the acoustic signal. This project arises from a common need for an easy-use detector allowing both the detection and its treatment.

Automatically detecting clicks

1 Autodetection methods

We tested for different thresholds (alpha) tested and chose the optimum ones

passe en log d'abord plus avantageux

Modify these parameters in preferences

1.1 TKg

1.2 Spectrogram

1.3 Kurtosis

1.4 Intercorrelation

1.5 Combining results

The four detection methods show different precision and recall results (see the Performances modeling section). They have advantages and flaws, e.g. Kurtosis do find most clicks but it is often paired with an important rate of false alarms. Combining the detections of these four methods then allows for a greater precision, eliminating all false alarms detected by one or two methods. We here offer the user to choose to combine the results from two to four methods, that is three additional detection methods. This means that to be accounted for, a signal must be detected by at least two to four methods. For example, combining two methods implies that only signals detected by at least two methods will be considered (6 possibilities, Table 1). These combination methods will be referred to as "comb2", "comb3", and "comb4" (the digit indicated the minimum number of methods that must have detected a signal for it to be considered).

Number of methods	Combination possibilities			
2	K			IntC
	K			EF
	K			TKg
	EF			IntC
	TKg			IntC
	EF			TKg
3	K	TKg		IntC
	K	TKg		EF
	K	EF		IntC
	TKg	EF		IntC
4	K	TKg	EF	IntC

1.6 Dead time

DeteClic uses vectors for each detection method, which are filled with ones when a signal is detected. To allow for a uniform detection, avoiding the same click to be detected several times, we considered a ”dead time” after all detection. Once a click is detected, no detection will be considered for a given period of time (i.e. what we call the dead time). That is, for this period, all samples will be filled with ones. The user can choose this period of time (in the Preferences settings).
For example, the default setting for sperm whale click detection is 75ms (close to Whitehead 1990) but this is adjustable to your own data set and species. One should note that, for any species, considering any dead time prevents detection of clicks with a lower ICI. In the discussed example of sperm whales, clicks with ICI$\leq$50ms or even $\leq$15ms have been described among creaks (Jaquet 2001) and will be excluded from the detection.

table for preset dead time by species

2 Launching detection

preferences species presets reference clicks

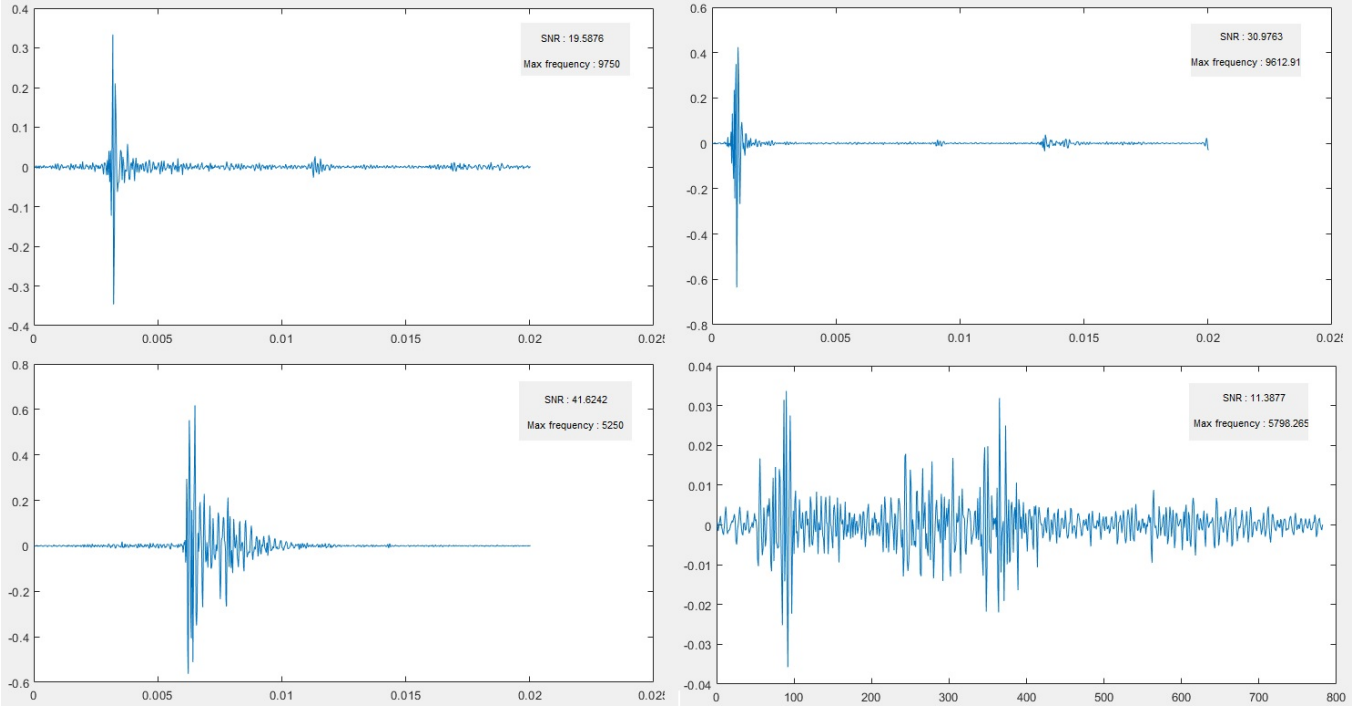


Figure 1

then see the ”outputs” doc

Automatic detection results

1 Detection outputs

One can modify the outputs described below and the train detection settings in the following tab: Edit▷Preferences▷Detection▷Metric.

1.1 Clicks

For each file analysed, the **number of clicks** is extracted as the number of signals detected by the selected method(s). From these, the **average click rate per minute**, the **average ICI** (in seconds) and **ICI standard deviation** are calculated. The number of minutes where clicks were detected ("**dpm**" , detection positive minutes) illustrates if signals were detected all along the recording (as done for porpoise (Brookes 2013, SAMBAH report). It is an indication of click intensity (Hernandez Milian 2008). As described below, echolocation clicks could be considered as part of a longer event, a click train. Therefore, we also provide the number of clicks found within trains (all train included).

1.2 Click trains

1.2.1 How to define click trains ?

Because parasite transient noises are very likely to be detected as clicks (false positive detections), we grouped detection events to detect click trains. This allows for a greater precision (specificity) in event detection but costs in terms of resolution. Clicks among trains are discriminated using 1) **ICI** and 2) **train thresholds**.
1) We considered that parasite transient noises are randomly distributed while clicks of an echolocation event are regular (Møll 2000, Jaquet 2001, and this was visually verified on a real dataset). They can be discriminated by the duration separating 2 signals, which should be regularly short within an echolocation train (generally ICI $\leq$2sec, Whitehead 1990, Goold 1995). One should keep in mind that clicks emitted in other context (e.g. social interactions, such as slow clicks (Mullins 1988), codas (Watkins and Schevill 1977) or clangs (Gordon 1987)) would be excluded from such analysis. By default, any detection separated from others by 5 sec or more would be excluded. This threshold was set using the broken sticks method (SiZer R package, Sonderegger 2009) but any user can adjust it.
2) The user can also define a minimum duration of a train. By default, trains are at least 20 seconds long.

1.2.2 Train analysis

DeteClic calculates the **number of trains** for a given audio and provides their **starting and ending times** (from which their duration time can be calculated). For each train, it calculates **the average ICI**, **ICI standard deviation**, and the **number of clicks contained in the event**. We use the detection of click trains to establish if there were animals emitting around the recorder (referred to as a **presence event**, based on a minimum number of trains).

2 Data management

Once the click detection and analysis done, DeteClic saves 2 complementary .xlsx files. The saving location can be selected in: Edit▷Preferences▷General.

2.1 Detection general outputs

First, " file_result.xlsx" records the name and duration of the audio files analysed, and if any animal were detected (presence: yes/no). For all the recordings, the file contains the number of trains, number of dpm, the click rate, average and standard deviation in ICI. We distinguish all detections found ("click number") from those part of a detected train (as defined above, "click in train"). If a file has not yet been analysed, the variables display "not analysed", which avoids confusion with analysed files without any clicks. One .xlsx file is created for all recordings analysed and it contains one sheet per selected method (plus the combinations).

file_result.xlsx

	A	B	C	D	E	F	G	H	I	J
1	file name	duration	presence	train number	dpm	click rate	average ICI	ICI variance	click number	click in train
2	cachalot	106,6916875	yes	1	3	93,91547022	0,634939759	0,346911476	167	167
3	1_H00001	224,9997245	not analysed	not analysed	not analysed	not analysed	not analysed	not analysed	not analysed	not analysed
4										

Figure 1: file_result.xlsx

2.2 Train events outputs

"train_result.xslx" displays all trains from all files analysed (i.e. each line computes all outputs for a train, Figure 2). All trains are numbered within a file. 2 trains can be numbered "2" but they belong to different files (seen in "file name"). For each train, its beginning and ending time (in sec) are indicated, along with the clicks and their ICI characteristics detailed above. One can also choose to display and save a graph with the signal amplitude and the train automatically and manually detected (Figure 3, see also Performances section). To save the graphs, tick the frame "save figure of train" (in Edit▷Preferences▷Detection▷Metric).

train_result.xlsx

	A	B	C	D	E	F	G
1	file name	train	beginning	ending	average ICI	ICI std	number of clicks
2	cachalot	1	0,04529167	105,158333	0,63493976	0,34691148	165
3							

Figure 2: train_result.xslx

2.3 Detection parameters

All the detection parameters are saved in a supplementary sheet in those two files. is it done ?? capture ecran

2.4 Noise

DeteClic calculates (i) the peak of intensity for three frequency intervals and (ii) the average intensity in these ranges. The user can adjust these intervals in: Edit ▷ Preferences ▷ Detection ▷ Detection variables ▷ Frequency intervals for noise estimation. This is further discussed in the Results section. The two variables are saved in a .mat format for each analyzed file.saved for all files?
We do not provide SNRs of every detection, but as DeteClic is an open-source detector, it is easily extractable.

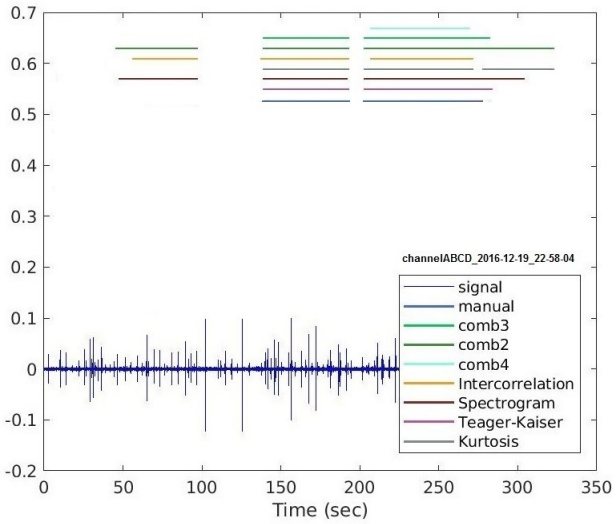


Figure 3: Displaying the signal amplitude over time and the trains labelled manually and detected automatically allows for a visual verification of DeteClic outputs.

Detection performances

1 Launching the performance tool

1.1 perf_launcher

calling files2perf and pathname loop for all files calling for audio and loading csv files (manuel+auto)

1.2 Position function

Filling the gaps - Manual detections
We first implemented a pre-filling of 20 ms (considered to be the length of a regular click Whitehead 1990, Goold 1995) for each manually detected click in the detected_man vector (i.e. 0.020*fe samples filled with 1, followed by a 0). This avoids any closely-located clicks to be skipped, as they could be considered as one instead of two clicks
Position vectors
As for automatic detections (see the appropriate section) we do not consider any detection within the dead time following a detected click (as set for the studied species or adapted by the user). As previously explained this filling interval can be adjusted. Based on these detections, we create a position vector for all selected methods (except if no click was found).
ADD ECHOES

2 Performance analysis

2.1 Comparing automatic and manual detections

We based our performance tool on a simple comparison on what is detected by the automatic methods and what is manually annotated. Each manually-detected click is compared the automatically-detected clicks: is the manual click included in an automatic one ? This is done with a tolerance interval (k). For instance, an automatically-found click is included in the sample interval [230, 800], it would look for any manual click found in the interval [230-k, 800+k].
If a manually-detected click corresponds to an automatic one (i.e. it is included in it), the comparison is stopped and the corresponding automatic click is excluded from the analysis. This avoids any automatic click to be twice counted as correct. It then pass on comparing the next manual click.

2.2 Performance indicators

Two outputs describe the automatic detections quality: their precision and recall rates. We calculate them as (Yang 2017,Roch 2013, Gillespie 2013, Towsey 2012):

$$\text{precision} = \frac{\text{number of correct clicks}}{\text{number of clicks found by the automatic method}}$$

$$\text{recall} = \frac{\text{number of correct clicks}}{\text{number of clicks manually found}}$$

The precision gives the false alarm rate while the recall indicates if the method found most of the manually-detected clicks.

2.3 Click trains and presence events

As detailed in the outputs session, a higher level of analysis allows for a better precision, based on a pattern recognition, here click trains. We consider regular and close clicks (low duration between two events) to form a train. The train must last some time. Comparing manual and automatic detection of trains would be challenging, DeteClic can display graphs allowing visual check. One can see if the trains found manually and automatically correctly overlap. This can also be used to verify if all clicks found are not parasite transient noise, if they do not belong to a train event (while only considering echolocation regular clicks, e.g. excluding codas).

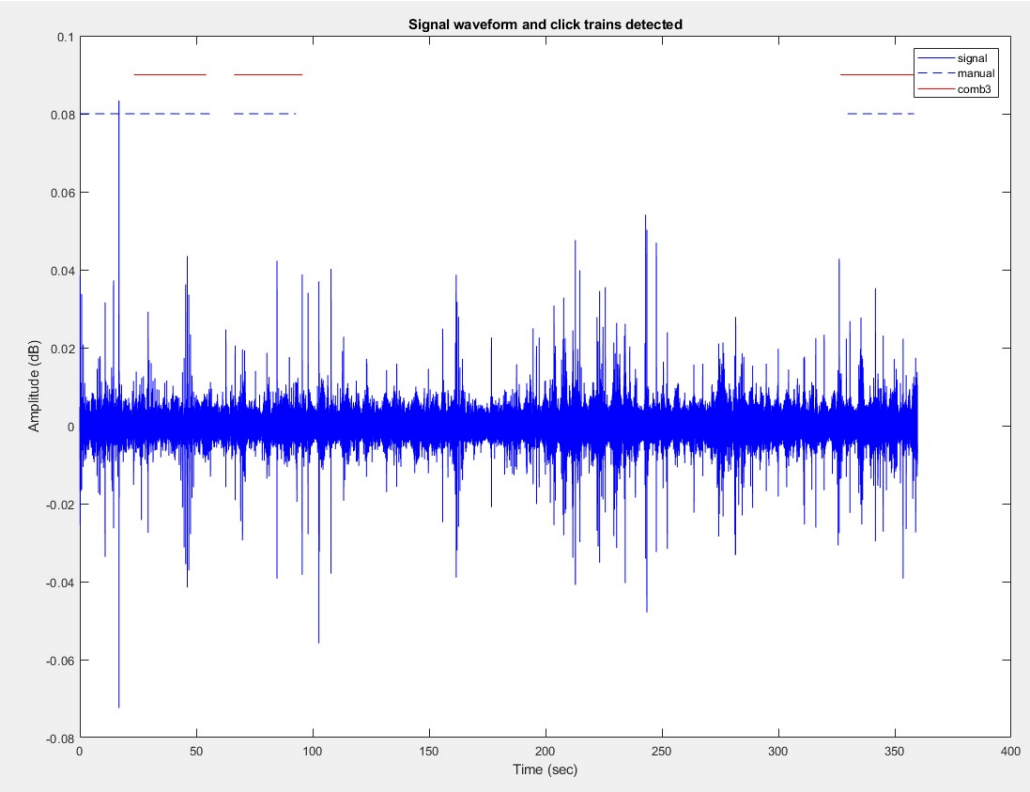


Figure 1: Displaying the signal amplitude over time and the trains labelled manually and detected automatically allows for a visual verification of DeteClic outputs.

3 Results display and record

3.1 Tables

A .csv file is created for each file analyzed (Figure 2). It includes the number of clicks found automatically and manually, with the associated precision and recall for audio files analyzed at once and selected methods.

3.2 Plots

Once the performance analysis finished and the tables saved. A window shows the average precision and recall rates with their standard error for all analyzed files. On the same window, DeteClic also produces visual display of the results from the comparison with manual labeling. Three plots are

17_12_14_47_files_used.csv

	A	B	C	D	E	F
1	methods	file	clicks_number	recall	precision	clicks_manuel
9	Intercorrelation	channelAB_2017-01-16_06-53-55	411	0.97	0.61	259
10	Spectrogram	channelAB_2017-01-16_06-53-55	215	0.57	0.69	259
11	Teager-Kaiser_g	channelAB_2017-01-16_06-53-55	180	0.48	0.69	259
12	Kurtosis	channelAB_2017-01-16_06-53-55	450	0.99	0.57	259
13	Combination 2	channelAB_2017-01-16_06-53-55	502	0.96	0.49	259
14	Combination 3	channelAB_2017-01-16_06-53-55	250	0.67	0.69	259
15	Combination 4	channelAB_2017-01-16_06-53-55	141	0.38	0.7	259

Figure 2: The performance analysis produces one .csv file for all .wav files analyzed. For all recordings, it indicates the number of clicks both found manually and by each automatic methods (clicks_number), with the associated recall and precision.

produced: (i) the number of manually- on automatically-detected clicks for all possible methods (Teager-Keiser on the Figure 3 example), (ii) boxplot of the precision and (iii) the recall rates for all selected methods (Figure 4).The user can choose between the three to display, and if they are to be saved (as .jpeg files).

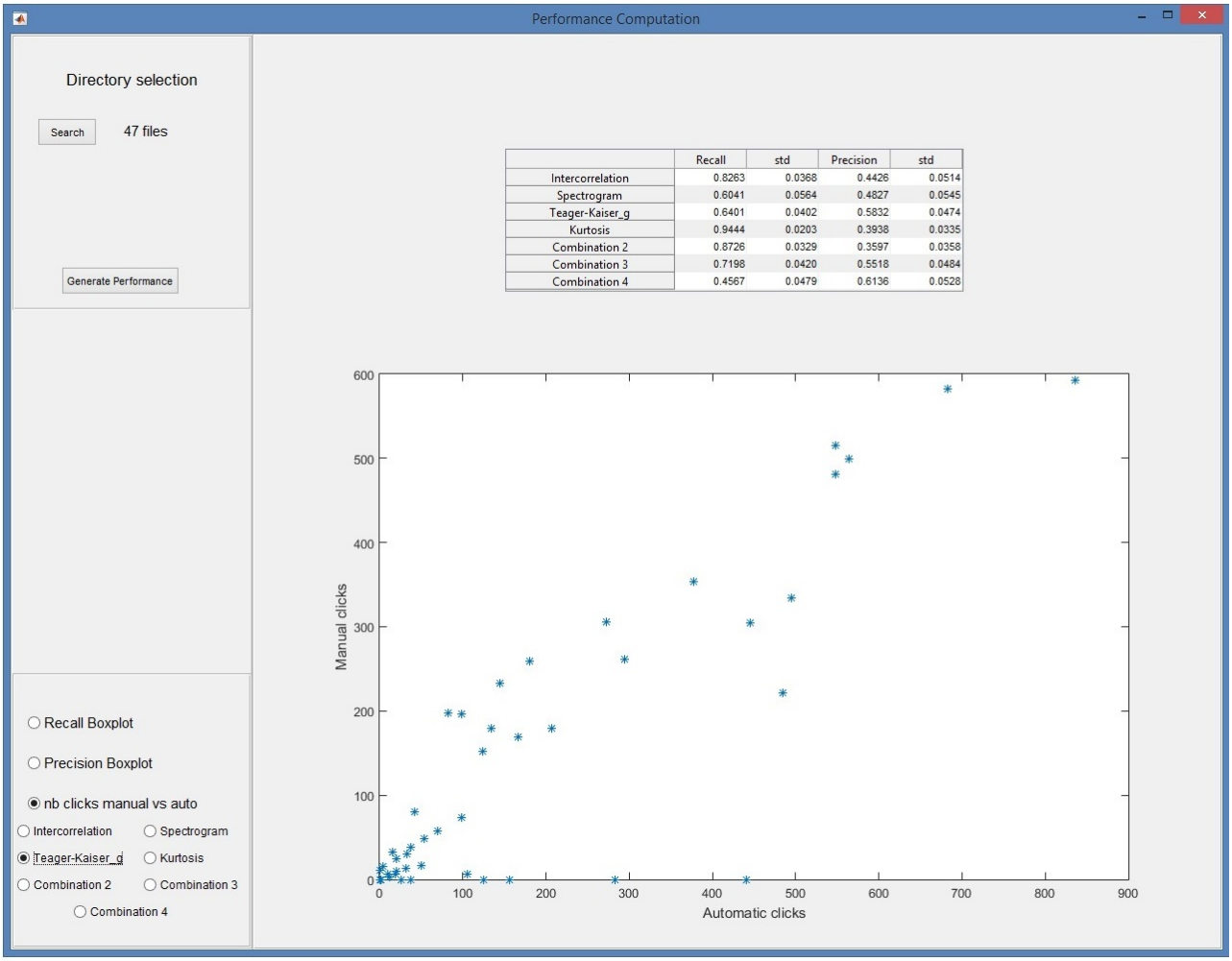


Figure 3: In addition to the rates and their variation, the user can chose to display the number of clicks found manually against that automatically detected.

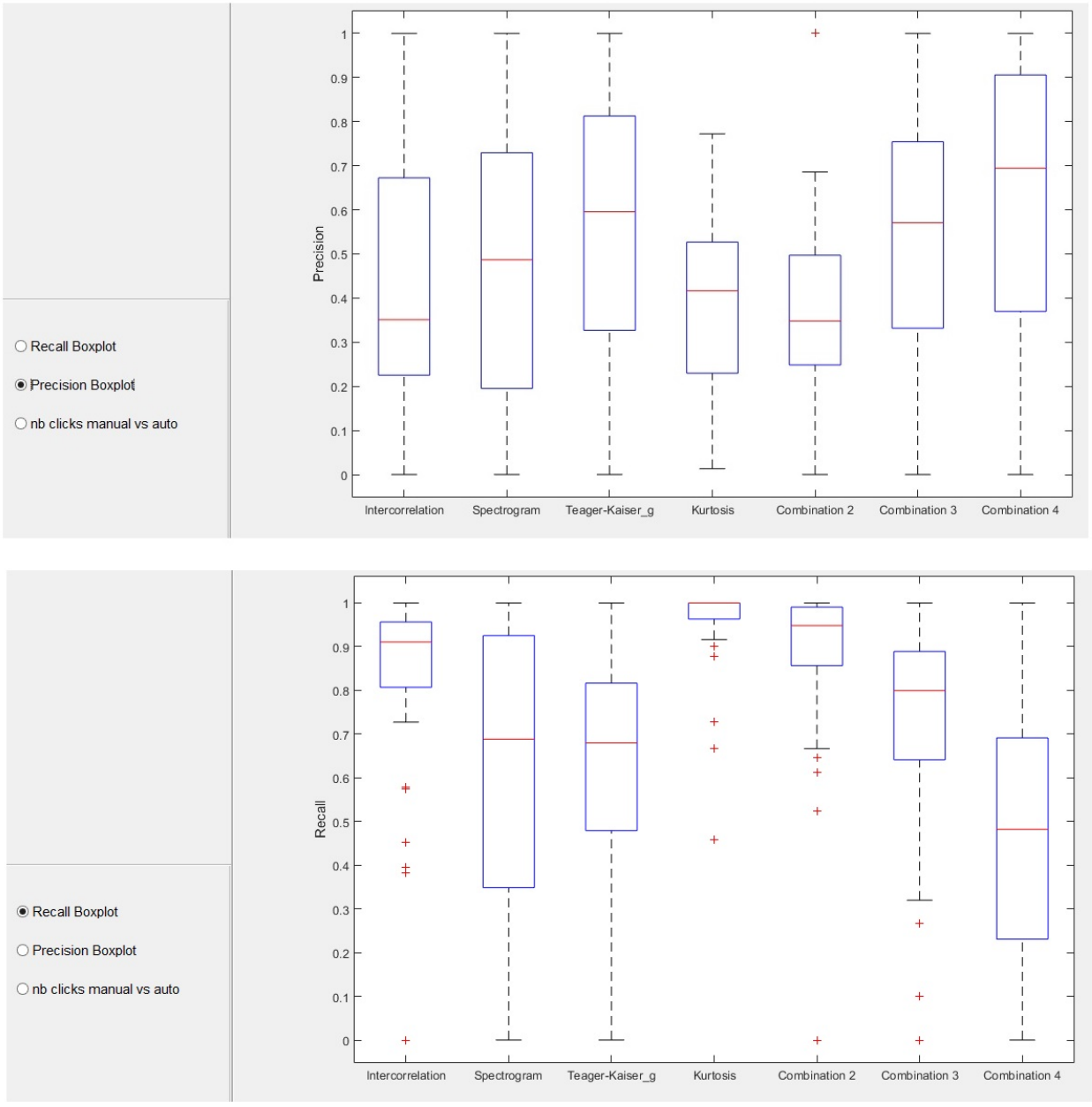
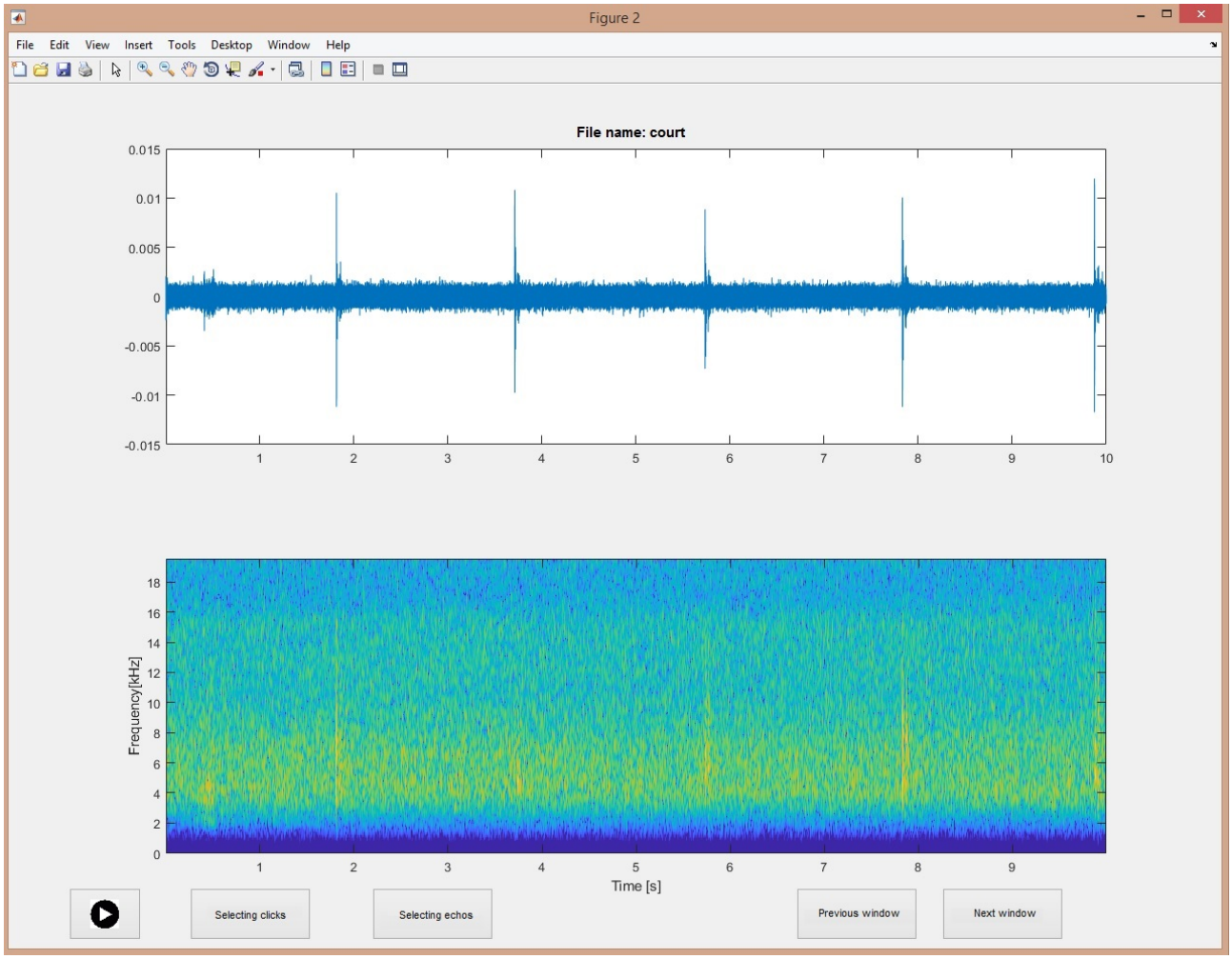


Figure 4: The user can also display and save boxplots of the recall and precision rates, as these plots are automatically printed at the end of the performance analysis.

Manually labeling clicks



0.1 Launching labeling

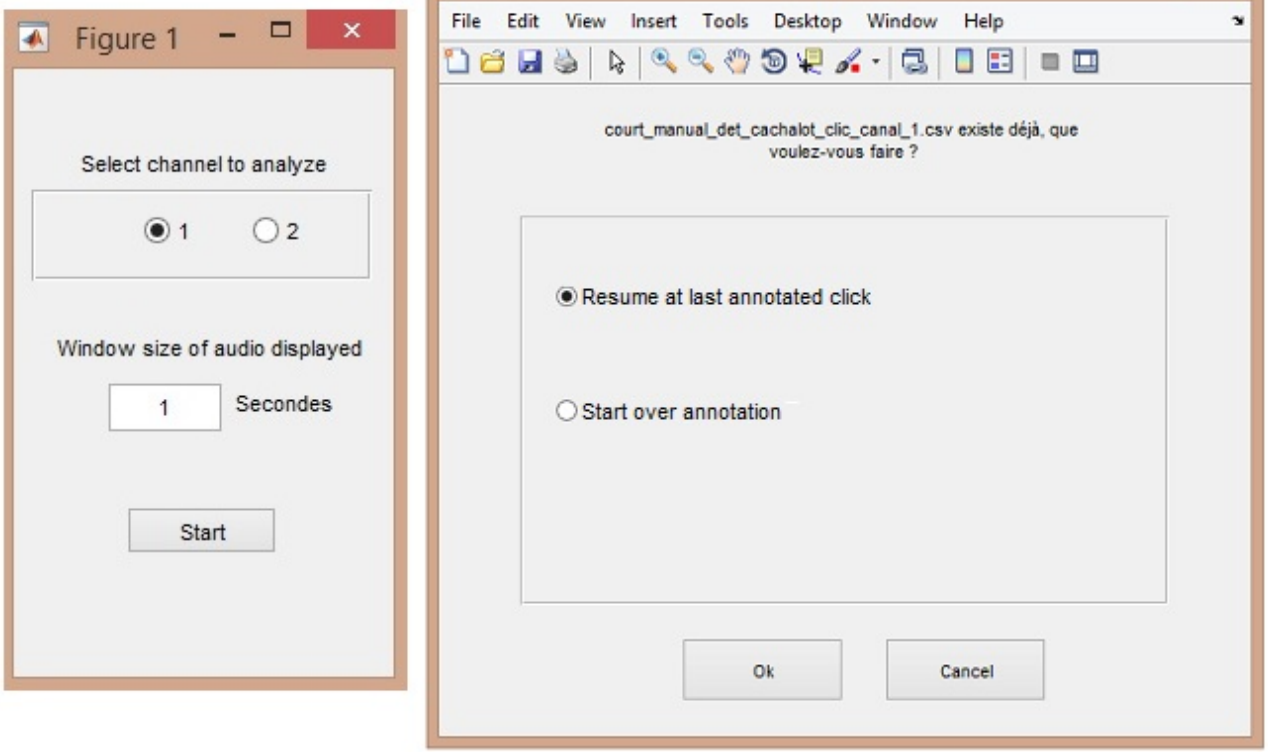
When launching DeteClic for a manual labeling operation, the user must select one audio file (selecting more would cause an error). The user is then asked to select the wanted channel and the appropriate display window (in seconds, Figure 2). Then, one can point at clicks on two graphs: the signal waveform and its spectrogram (Figure 1 and 3). One can differentiate clicks and their echoes. While doing so, it is possible to listen to the signal sample defined by the current window of analysis.

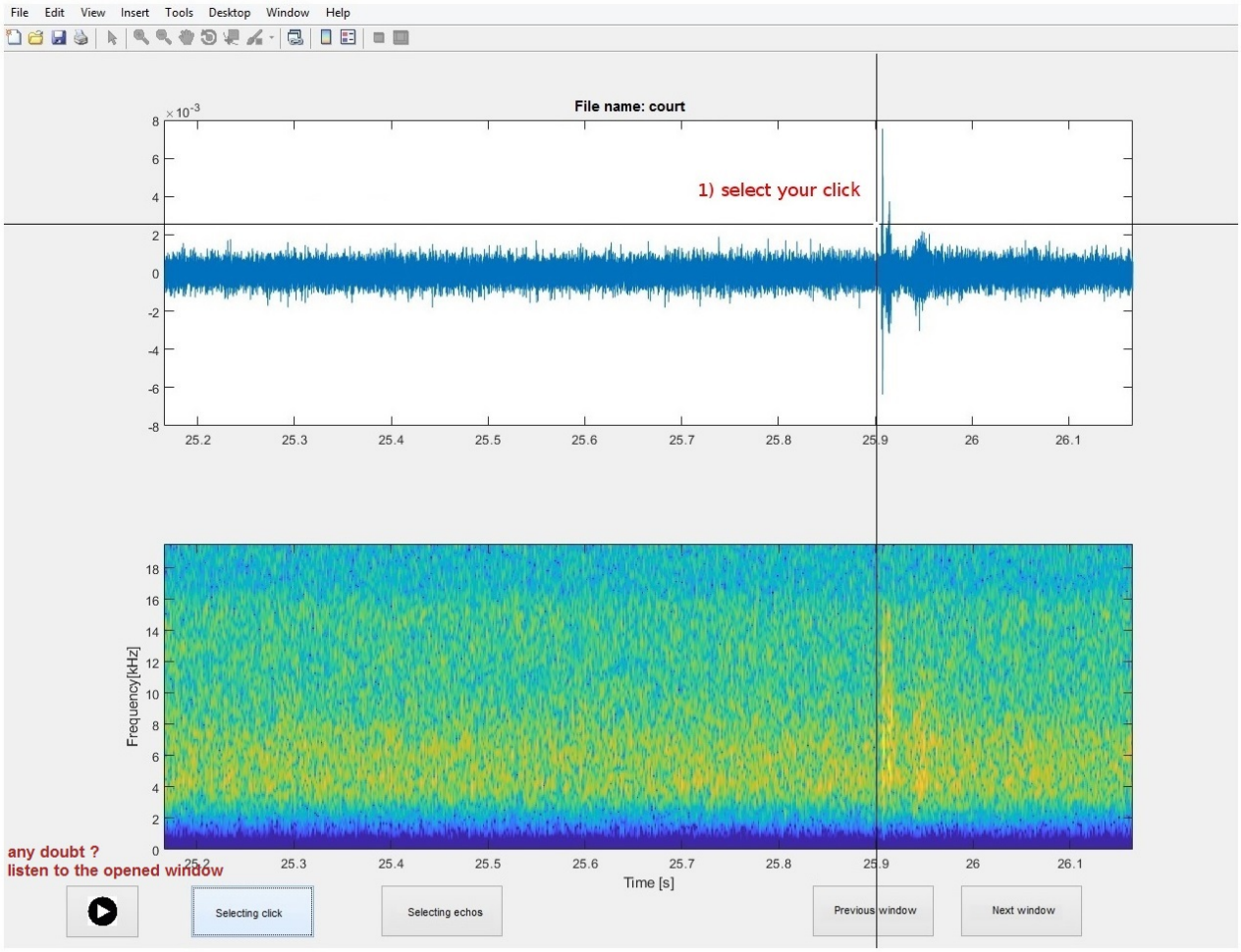
0.2 Saving labeling

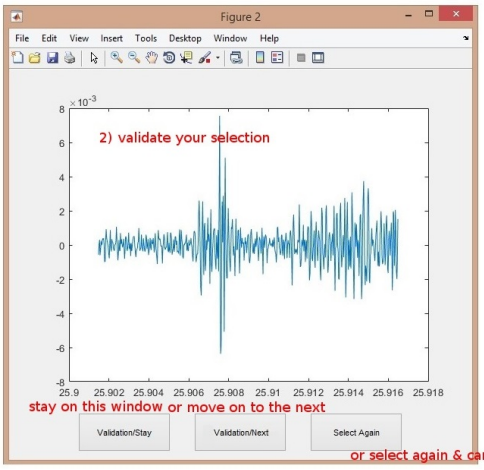
Labelled clicks and echoes are saved on two different .csv files in a ”manual detection” folder. The folder is created if it does not exist yet in the path folder. The two files are named:
”$filename$”$\_manual\_det\_spw\_click\_channel\_1.csv$
”$filename$”$\_manual\_det\_spw\_echoes\_channel\_1.csv$
One should be careful not to rename them, as they are needed for any comparison with automatic detections to establish the performance of the detector.

If there is already an existing manual detection, one will have the choice either to rewrite it, by starting over the detection, or to continue the existing (from the last labeled-click).

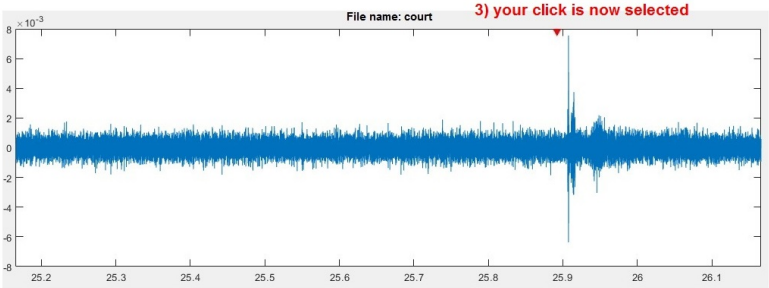
figure for already existing files add translation to resume annotation







or select again & cancel this selection



Our performance tests - detection results on a real data set and advised precautions

A- Detection quality

0.1 Audio samples

DeteClic performance analysis was conducted on 51 audio files (270.3 min). These were collected by Gaëtan Richard, on longline fishing vessels within the Kerguelen Island Exclusive Economic Zone. Hydrophones were either deployed on fishing lines or attached to a buoy and a ballast (referred to as "antenna files") from December 2016 to late January 2017. We randomly selected groups of four contiguous recordings among all deployments. Among these, 39 files were suitable for the quality description, while 33 contained manually-found clicks and were used for statistical analysis described below. The 12 remaining files were used for primary exploration and detection settings adjustments. These include regular files or with no clicks, made before the line lowering in water, or antenna files with strong transient parasite noises. All these recordings were filtered with a 4000 Hz high-pass filter.
All qualitative analysis was ran either using Audacity 2.1.1 or DeteClic (ran on Matlab R2017b).

0.2 Precision and recall rates

The precision and recall rates calculation is presented in the Performances section. The detection quality of the different methods, obtained on 41 audio files, are presented in Figure 1.a). Visual representations of these metrics are found in Figure 2 and 3. We found that combining the results of at least three detection methods is the best approach to estimate clicks number and robust presence-event analysis. This method finds most clicks (73%$\pm$4) with a reasonable precision rate (56%$\pm$5).
Still, all methods analysis showed a correlation between the number of clicks manually- and automatically-detected (Figure 1.b)).
We would like to point out these estimations might be underestimated due to the comparison itself (see the Perfomances section). We compare the automatic detection to the manual labelling, which is subjective and not flawless (e.g. some missed clicks). Echoes and the click's multipulse structure might induce some mistakes in the comparison: if the user points at the second pulse and the detector at the first, if they are separated by several milliseconds, the detection will be counted as a false alarm (while missing the click). This can be adjusted with the tolerance threshold, but increasing it will allow for more mistakes.

0.3 Processing time

lalala we havent done it yet

a)

Rates (%)	Recall (±SD)	Precision (±SD)
TKg	64±5	52±5
IntC	81±4	45±5
EF	65±6	38±6
K	88±3	47±4
comb2	84±4	36±4
comb3	71±5	54±5
comb4	49±5	57±6

b)

	r^2	t stat	p-value	DF	Slope	intercept
TKg	0.90	18.652	<2.10^-16	37	0.72	12.66
IntC	0.24	3.37	0.00179	37	0.15	69
EF	0.28	3.82	0.000497	37	0.1	74.54
K	0.72	9.82	7.5.10^-12	37	0.41	26.42
comb2	053	6.45	1.54.10^-7	37	0.19	36.1
comb3	0.81	12.48	7.85^-15	37	0.45	36.19
Comb4	0.85	14.45	2.10^-16	37	0.89	33.25

Figure 1: Results of a) recall and precision rates, and b) correlation of the number of clicks found manually or by the different methods.

B- Testing what factors influence our performances

1 Introduction

The following modeling is useful to understand how the audio characteristics, the methods or the detection parameters selected would influence the outputs of DeteClic automatic detections. We considered two outputs for each 33 audio files tested: the precision and recall rates.
We attempted to find an indicator illustrating both precision and recall rates, you can find further details in the blabla section.

1.1 Explanatory variables

We hypothesized that results would differ between (1) files, (2) detection parameters and (3) detection methods.

1. We decided to specify audio file characteristics (rather than just taking a ”file” variable). It was not possible to consider a variable describing transient parasite noises, which are the main source of errors of DeteClic. Regarding the ambient background noise, we only considered the peak of intensity for the frequency range between 4000 and 25000Hz (the file was first filtered to remove all clicks detected by the combination of at least two methods (see the Detection methods section and 5. Noise pollution below). We also accounted for the presence of audible clicks using the number of clicks manually labelled.

2. The detection parameters that could influence the results are the needed inputs for the different methods, that is: a reference click and a noise interval (see the Autodetection section). Here, we took into account if a given file was used as reference. As discussed later, some methods do not use the reference click.

3. Finally we considered that the detection methods would differ in their results.

1.2 Hypotheses

We can summarize the variables and the associated hypotheses for the results (i.e. recall and precision rates) as:

- X_{i1} = indicates if the file is used to select noise and click references, binary variable of yes/no form.
  We hypothesized that files used as the reference would show greater results.

- X_{i2} = number of manually-labelled clicks, continuous variable. It was standardized through a centred-reduction, defined as: (X-mean(X))/standard deviation(X).
  We hypothesized that this number is close (at least proportional) to the number of existing clicks and that an important number of clicks would be easier to detect.

- X_{i3} = peak of intensity for ambient signals with a frequency between 4000 and 25000Hz, continuous variable, also standardized as X_{i2}.
  Background noise of strong intensity could be mistaken as clicks, we therefore expected that a high value of X_{i3} would induce a lower quality of the detection.

- X_{i4} = detection methods, nominal variable with 7 levels.
  We expected the combination of at least 3 methods to be the best method.

- Because all methods are not based on the click and noise references, we expect a positive effect of the X_{i1} variable for some methods (some levels of X_{i4}). This was modelled through an interaction of these two variables.

Continuous variables (X_{i2} and X_{i3}) were centered-reduced to allow for an easier comparison of the resulting coefficients and it avoids any convergence issues.

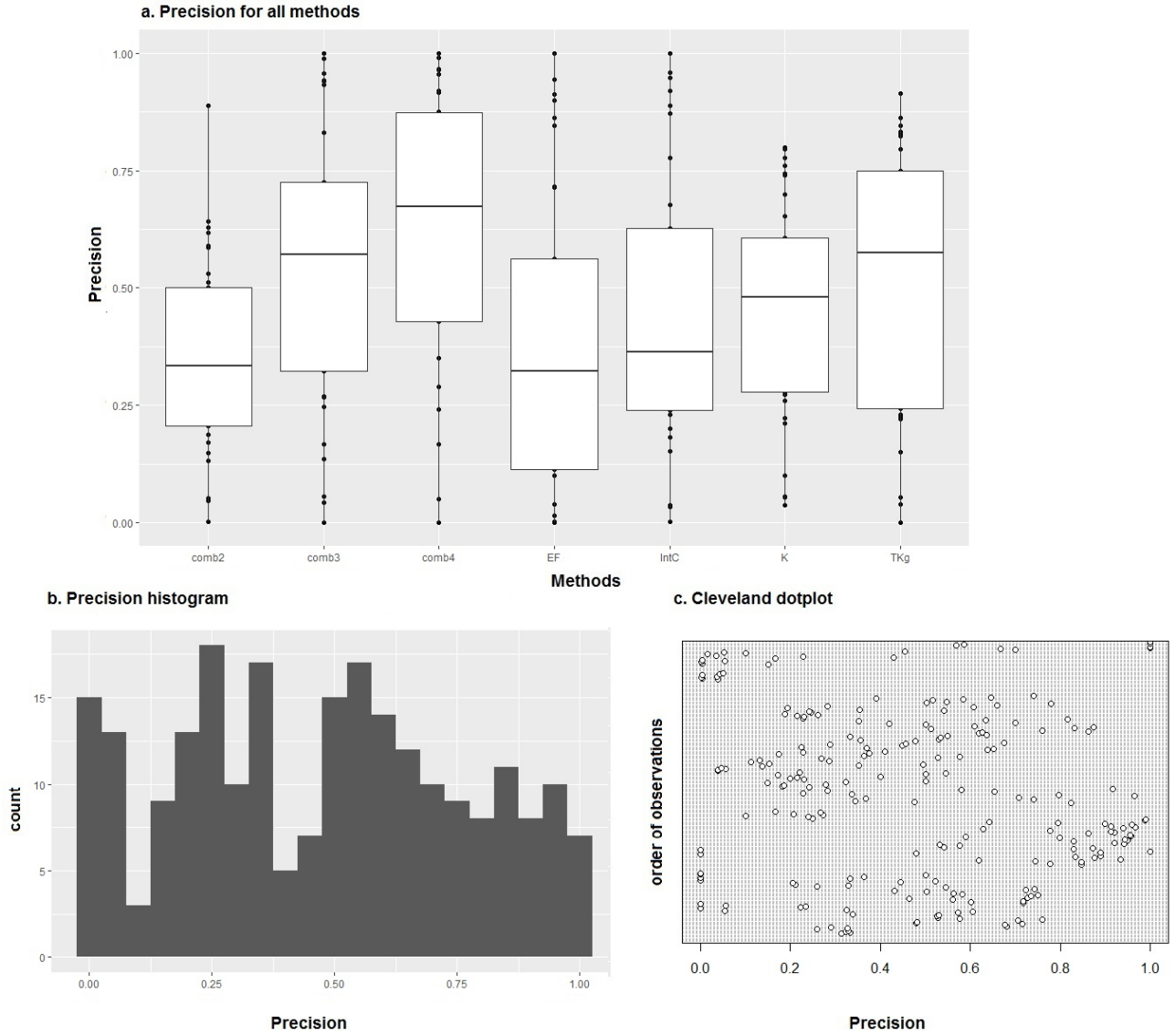


Figure 2: Data exploration of the precision rates depending on a. methods, b. for all the observations (all files and methods confound), and c. the associated dotchart.

1.3 Data exploration

We followed the instructions given by Zuur et al. (2010) to ensure a proper data exploration. For both recall and precision rates, we produced a set of graphs prior and after the model analysis, allowing visual check for tendencies and validation of the model choice (e.g. no zero inflation). Cleaveland dotplots of the two rates allowed for visual exclusion of outliers (there was none, Figures 2 and 3). Response variables were plotted against any time variables (order of the files and recording date) as well as spatial variables (boats) to check for any pattern, any interaction between variables (data not shown). We also produced coplots and pairplots of all variables against each other (Figure 4).

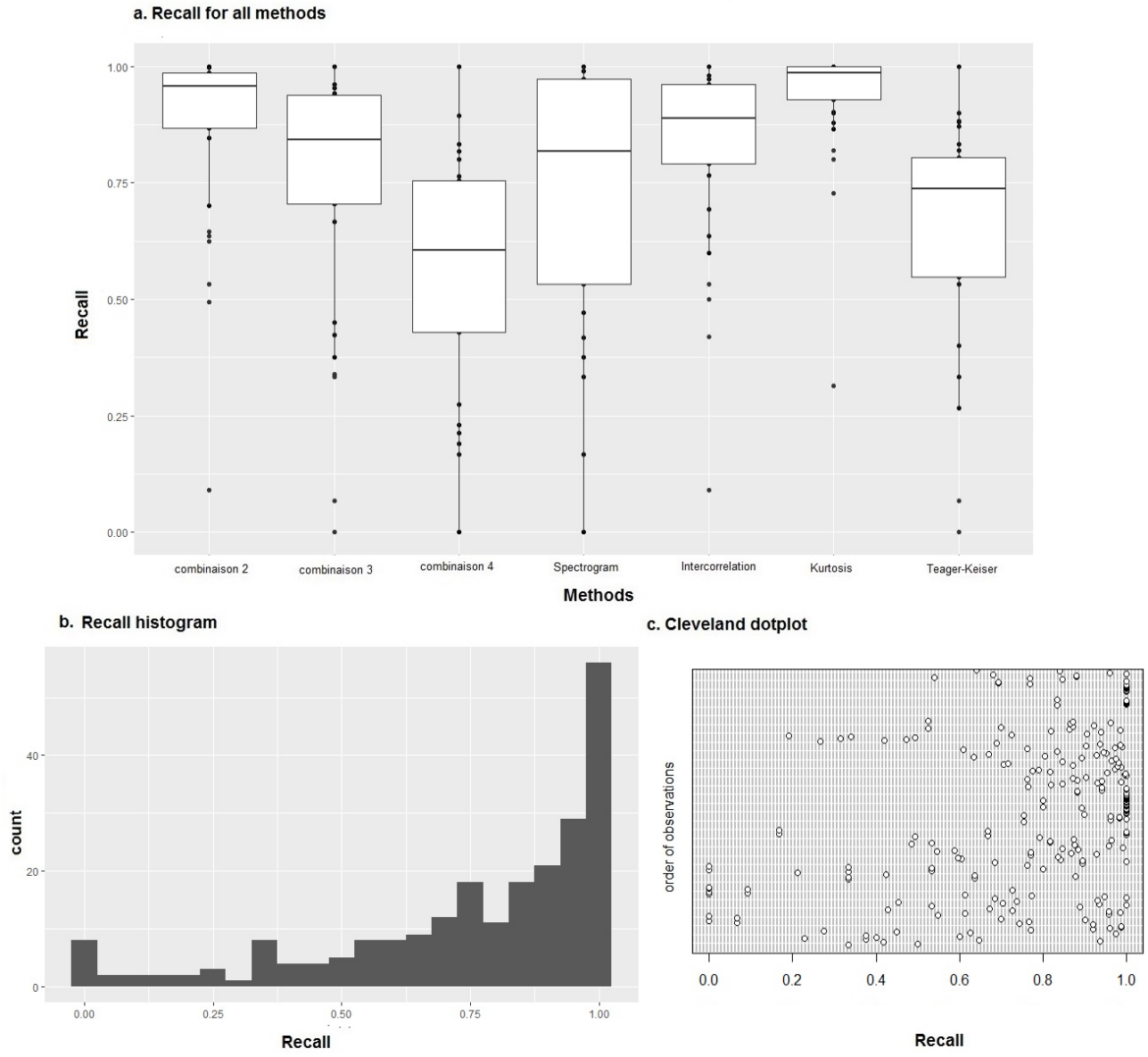


Figure 3: Data exploration of the recall rates depending on a. methods, b. for all the observations (all files and methods confound), and c. the associated dotchart.

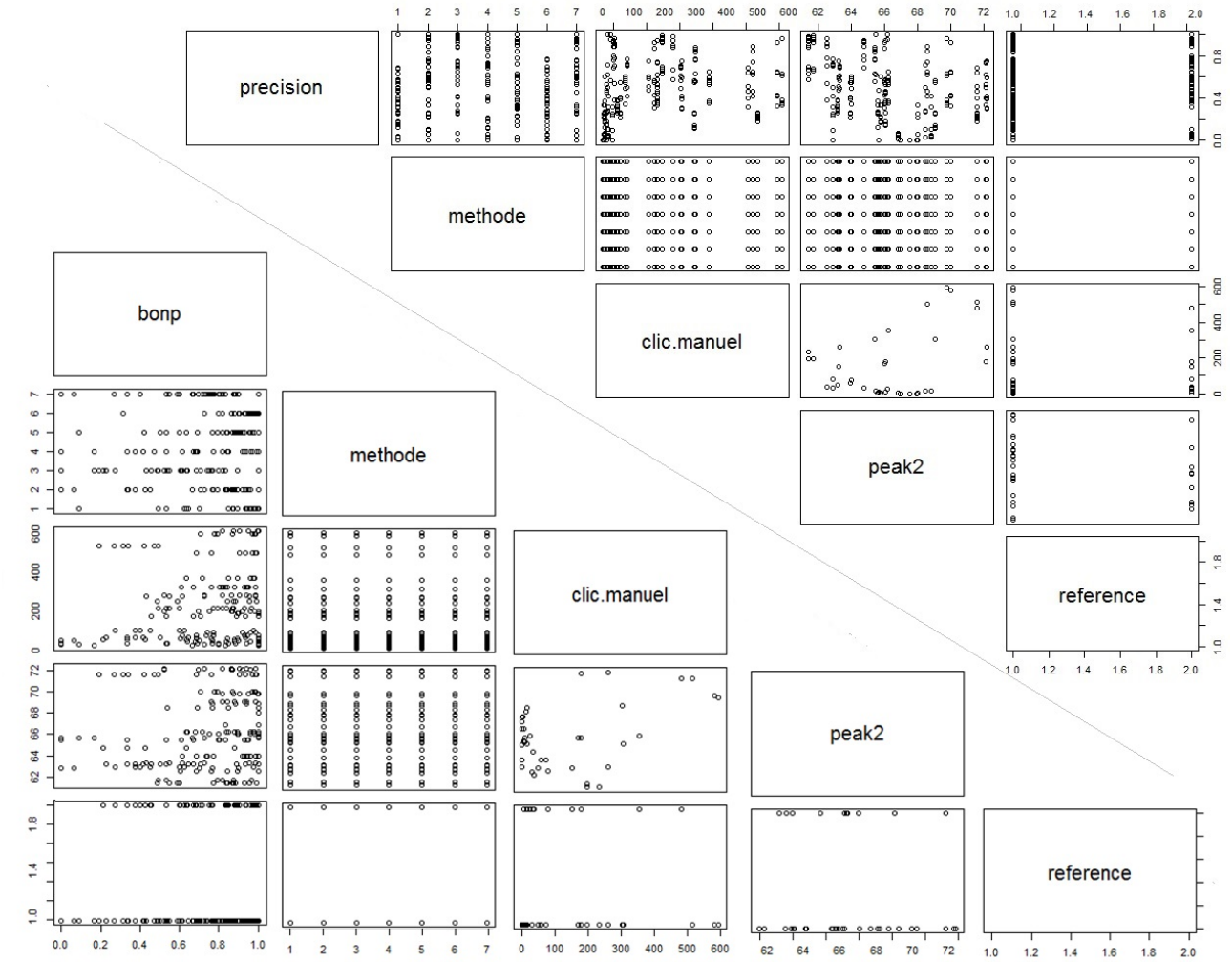


Figure 4: Plairplots produced for both recall and precision rates. This allowed to check for any tendencies and if there were unexpected interactions to account for.

2 Modeling

2.1 Models

2.1.1 General linear modeling

We considered two response variables:

Y_i: the recall rate for the file i
Z_i: the precision rate for the file i

These variables are proportional data (comprised between 0 and 1) and can be modelled using a logistic regression (Zuur et al. 2007, 2009). They can be written as:

$Y_i \sim B(n_i , \pi_i)$ $\quad E(Y_i)= n_i * \pi_i$ $\quad Var(\pi_i)= n_i * \pi_i (1-\pi_i)$

with π_i the probability that a detected signal is a correct click (i.e. manually labelled), so $1-\pi_i$ the probability that a click is missed.
The same applies to Z_i, where π_i is the probability that the click is detected both manually and automatically, but $1-\pi_i$ is the probability that a click is a false alarm.

The complete models tested were of the form :

$Y_i= \text{logit}(\pi_i)$
$Y_i= \alpha+ \beta_1 * X_{i1} + \beta_2 * X_{i2} + \beta_3 * X_{i3} + \beta_4 * X_{i4} + \beta_5 * X_{i1} * X_{i4} + \varepsilon_i$

$$Z_i = \alpha + \beta_1 * X_{i1} + \beta_2 * X_{i2} + \beta_3 * X_{i3} + \beta_4 * X_{i4} + \beta_5 * X_{i1} * X_{i4} + \varepsilon_i$$

with α the intercept, ε the residuals, X_{ip} the covariates (with p $\in [1,4]$) and β_p the associated slopes, for a given file i. We hypothesized that the parameters (β and residuals), as well as the 33 files, are independent. Because we use a logistic regression, there is no underlying assumption on the residuals distribution.
These models were tested using R Studio 1.1.383, the glm function and a binomial family. Weights selected for in the glm formula depended on the response variable: Z_i was weighted by the number of clicks automatically detected while the number of manually-labelled clicks were used for Y_i.

2.1.2 General linear mixed modeling

Each observation (i.e. detection) is made for a given file (i $\in [1,33]$ and a detection method (j $\in [1,7]$). Because methods can be seen as nested in the file level, considering a random effect of this variable (with j levels) could be more realistic. Therefore we tested models with mixed effect (random or fixed) based on the same variables and a logistic regression. The complete mixed models also accounted for an interaction between X_{i1} and X_{i4}. For both recall (Y_{ij}) and precision (Z_i) rates, models can be written as:

$$Z_{ij} = \alpha + \alpha_j + (\beta_1 + z_j) * X_{ij1} * X_{i4} + \beta_2 * X_{ij2} + \beta_3 * X_{ij3} + z_j * X_{i4} + \varepsilon_{ij}$$

where α the intercept of the fixed effects, α_j that of the random part, ε the residuals, X_{ip} the covariates (with p $\in [1,4]$), β_p the slopes of the variables with a fixed effect, and z_j the slopes associated with random effects, for a given file i and detection method j.
Which in R was translated to:

$$Precision_{ij} \sim X_{ij1} + X_{ij2} + X_{ij3} + (1 \mid X_{ij4}) + (1 \mid X_{ij1}{:}X_{ij4}))$$
$$Recall_{ij} \sim X_{ij1} + X_{ij2} + X_{ij3} + (1 \mid X_{ij4}) + (1 \mid X_{ij1}{:}X_{ij4}))$$

and tested using the glmer function R, again with a binomial family.

2.2 Model selection

a. Precision modeling

Models	AIC	Null deviance	Residual deviance	VIF	drop1
1. $Z_i \sim \alpha + \beta_1 X_{i1} + \beta_2 X_{i2} + \beta_3 X_{i3} + \beta_4 X_{i4} + \beta_5 X_{i1} * X_{i4} + \varepsilon_i$	14836	27573 Df=225	13721 Df=210	X_{i2} : VIF=1.81, Df=1 X_{i3} : VIF=1.63, Df=1	X_{i2}: Df= 1, AIC=18423, LRT=3588.7, pvalue<0.0001 X_{i3} : Df= 1, AIC=19794, LRT=4960, pvalue<0.0001 $X_{i1} * X_{i4}$: Df= 6, AIC=14984, LRT=159.7, pvalue<0.0001
2. $Z_{ij} \sim \alpha + \alpha_j + (\beta 1 + z_j) X_{ij1} + \beta_2 X_{ij2} + \beta_3 X_{ij3} + \beta_4 X_{ij4} + z_j X_{ij4} + \varepsilon_{ij}$	14900	24386	14888 Df=225		X_{i2}: Df= 1, AIC =18492, LRT=3593.5, pvalue<0.0001 X_{i3}: Df= 1, AIC=19860, LRT=4961.6, pvalue<0.001 X_{i1} : Df= 1, AIC =14902, LRT=4.4, pvalue=<0.05
3. $Z_{ij} \sim 1$	28658	27573	27573 Df=225		

b. Recall modeling

Models	AIC	Null deviance	Residual deviance	VIF	drop1
1. $Y_i \sim \alpha + \beta_1 X_{i1} + \beta_2 X_{i2} + \beta_3 X_{i3} + \beta_4 X_{i4} + \beta_5 X_{i1} * X_{i4} + \varepsilon_i$	8624.9	10739.2 Df=230	7808.6 Df=215	X_{i2} : VIF=1.69, Df=1 X_{i3} : VIF=1.67, Df=1	X_{i2}: Df= 1, AIC=8714, LRT=91.4, pvalue<0.0001 X_{i3}: Df= 1, AIC=8673, LRT=50.1, pvalue<0.0001 $X_{i1} * X_{i4}$: Df= 1, AIC=8713, LRT=99.59, pvalue<0.0001
2. $Y_{ij} \sim \alpha + \alpha_j + (\beta 1 + z_j) X_{ij1} + \beta_2 X_{ij2} + \beta_3 X_{ij3} + \beta_4 X_{ij4} + z_j X_{ij4} + \varepsilon_{ij}$	9850	10739	8667 Df=225		X_{i1}: Df= 1, AIC =8769, LRT=91.2, pvalue<0.0001 X_{i2}: Df= 1, AIC=8728, LRT=50.13, pvalue<0.0001 X_{i3} : Df= 1, AIC =8687, LRT=9.22, pvalue<0.01
5. $Y_{ij} \sim 1$	8680	10739	10739 Df=230		

Figure 5: Model selection results for a. precision and b. recall rates. Most parsimonious models are written in blue.

We followed the top-down selection described by Zuur et al. (2007). First, we implemented the complete models, i.e. with all covariates presented above, based on our hypotheses. Then, we removed all continuous variables showing variance inflation factors (VIF, vif function from the car package) greater than 3 (Zuur et al. 2010). In addition, we used the drop1 function (with a Chi^2 test) to test if any other variables (including continuous) should be dropped, that is if dropping a covariate induced no significant change. The last step was to check for non-significant variables. These three steps resulted in a set of models, that we discriminated on their AIC (Akaike Information Criterion), and we selected the model with the lower estimator.

3 Results

3.1 Model validation

The most parsimonious models are presented in Figure 5. It was best to consider methods as a fixed-effects factor, for both precision and recall rates modeling.
We studied the distribution of different types of residuals for the two models showing highest likelihood (using the anova and resid functions, Figure 6). Because none showed any pattern, we could interpret the results.

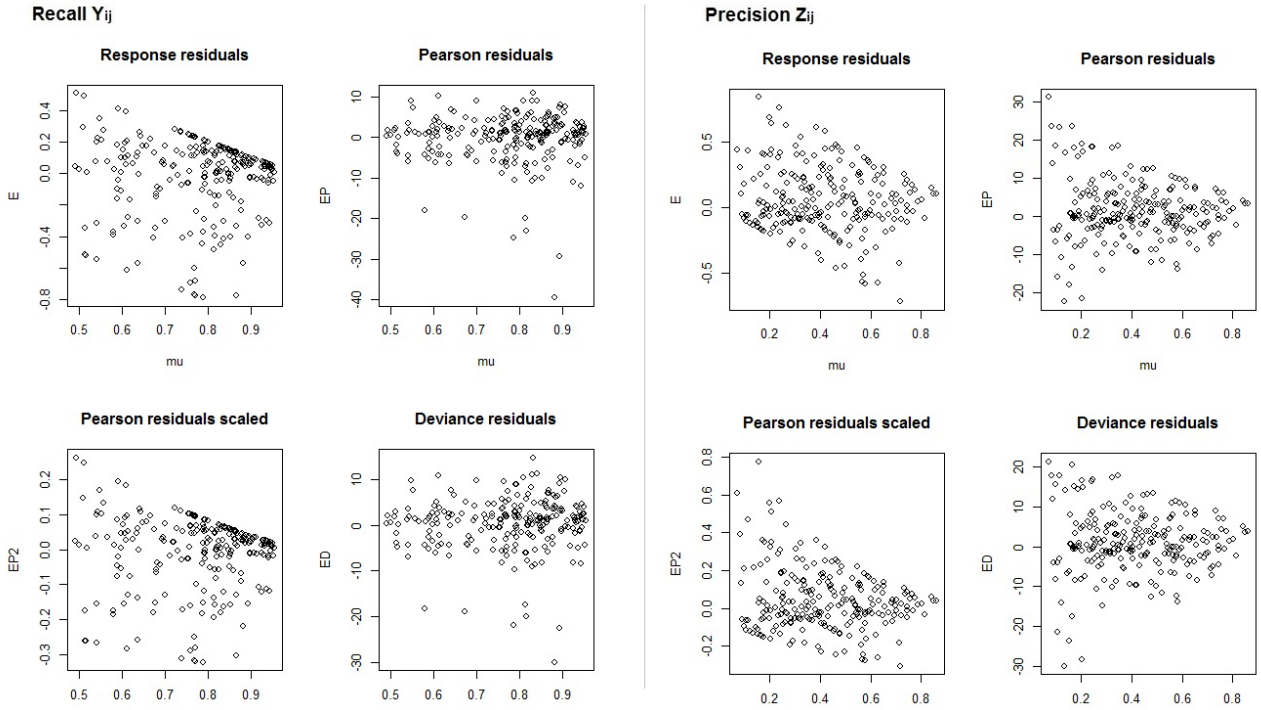


Figure 6: Residuals for the models Y1 and Z1.

3.2 Precision and recall rates modeling

We tested the effect of the same variables on the precision and recall rates (see above). The results presented in Figures 7 and 8 used for reference: (i) the combination of at least two methods (ii) on a file that was not used as a reference (X_{i1}="No" and X_{i4}="combination 2"). Both chosen models explained an non-negligible part of the observed variance (27% for Y_{i4} and 50% for Z_{i4}, Figure 5).

- **Precision rates**
  Modeling results are shown in Figure 7. Combining four methods appeared to give the best precision rates (followed by Teager-Keiser and the combination of three methods, i.e. they

allowed the lower rates of false alarms). Precision rates were significantly better for a higher number of real clicks and a lower background noise intensity. Precision was optimized for files used as reference for all methods, except the Intercorrelation. This result is puzzling, as the latter is the only method to use the reference click.

- **Recall rates**
  Modeling results are shown in Figure 8. Among all 7 methods, Kurtosis and combining two methods showed the best recall rates (i.e. lower false negative rates), and they showed equivalent recall rates. Using a file as the reference increased the overall recall, but the effect varied between methods. Files used as reference showed better recall rates for Intercorrelation and Kurtosis (but because other methods showed lower recall results than combination 2, it would explain the absence of visible effect when comparing to combination 2). On the other hand, recall rates were higher for higher number of clicks. In general, even when using the combination of three methods, the precision rates are low and the train analysis prevent wrong inference.
  The variable X_{i3} we used to approach parasite noise appeared to influence both recall precision rates (Figure 7, 8). This means that higher noise intensity would increase the number of false alarm (false positive) and the probability to miss clicks (false negative).

Only the intercorrelation method uses the click reference (it positively affects its recall rates and negatively affects its precision). Also, because much more files were not used as reference, it might mask any effect. Thus the effect that we found here on other methods could illustrate that background noise levels can change significantly between four contiguous audio files, leading to a decrease in the detection quality. Therefore, one should be careful not to analyze too many files based on only one click and noise reference.

a. Precision rates modeling

$Z_i \sim \alpha + \beta_1 X_{i1} + \beta_2 X_{i2} + \beta_4 X_{i4} + \beta_5 X_{i1} * X_{i4} + \varepsilon_i$:

	Estimate	Std error	Z value	p-values
Intercept	-1.23	0.02	-55.21	< 0.0001
X_{i4} combinaison 3	0.95	0.03	29.33	< 0.0001
X_{i4} combinaison 4	1.63	0.04	39.53	< 0.0001
X_{i4} Spectrogram	-0.19	0.03	-6.716	< 0.0001
X_{i4} Intercorrelation	0.43	0.03	14.33	< 0.0001
X_{i4} Kurtosis	0.75	0.03	24.43	< 0.001
X_{i4} Teager Keiser	1.29	0.04	35.67	< 0.0001
X_{i1} Yes	0.07	0.04	1.732	0.08
X_{i2} number of clicks manually-detected	0.5	0.01	57.5	< 0.0001
X_{i3} intensity peak Hz[4000,25000]	-0.7	0.01	-67.8	< 0.0001
X_{i4} combinaison 3 : X_{i1} Yes	0.44	0.07	6.8	0.11
X_{i4} combinaison 4 : X_{i1} Yes	0.34	0.08	4.03	< 0.0001
X_{i4} Spectrogram : X_{i1} Yes	0.1	0.06	1.91	0.08
X_{i4} Intercorrelation : X_{i1} Yes	-0.23	0.06	-4.17	< 0.01
X_{i4} Kurtosis : X_{i1} Yes	0.35	0.06	5.81	< 0.0001
X_{i4} Teager Keiser : X_{i1} Yes	0.01	0.07	0.09	< 0.0001

Figure 7: Precision modeling results

b. Recall rates modeling

$Y_i \sim \alpha + \beta_1 X_{i1} + \beta_2 X_{i2} + \beta_4 X_{i4} + \beta_3 X_{i3} + \beta_5 X_{i1} * X_{i4} + \varepsilon_i$

	Estimate	Std error	Z value	p-values
Intercept	1.96	0.05	36.9	< 0.0001
X_{i4} combinaison 3	-0.82	0.06	-12.9	< 0.0001
X_{i4} combinaison 4	-1.8	0.06	-30.15	< 0.0001
X_{i4} Spectrogram	-0.65	0.06	-10.14	< 0.0001
X_{i4} Intercorrelation	-0.64	0.07	-9.94	< 0.0001
X_{i4} Kurtosis	-0.13	0.07	-1.81	0.07
X_{i4} Teager Keiser	-1.41	0.06	-23.24	< 0.0001
X_{i1} Yes	0.77	0.13	5.96	< 0.0001
X_{i2} number of clicks manually-detected	0.16	0.02	9.61	< 0.0001
X_{i3} intensity peak Hz[4000,25000]	-0.1	0.01	-7.101	< 0.0001
X_{i4} combinaison 3 : X_{i1} Yes	-0.22	0.16	-1.41	0.16
X_{i4} combinaison 4 : X_{i1} Yes	-0.55	0.14	-3.81	< 0.01
X_{i4} Spectrogram : X_{i1} Yes	-0.55	0.15	-3.56	< 0.001
X_{i4} Intercorrelation : X_{i1} Yes	0.38	0.17	2.23	< 0.05
X_{i4} Kurtosis : X_{i1} Yes	0.23	0.19	1.24	0.22
X_{i4} Teager Keiser : X_{i1} Yes	-0.52	0.15	-3.58	< 0.05

Figure 8: Recall modeling results.

C- Optimizing DeteClic

4 Noise and reference click selection

For the reasons explained below, we recommend to analyze a reasonable set of audio files with the same references to optimize the detection quality (we chose to run DeteClic on five-files sets).

4.1 Selecting noise for detection threshold

Analyzing audio files with important background noise can be challenging for DeteClic: many transient noises can be mistaken as clicks. We first ran several qualitative analysis (data not shown). Then, we tested for this on recordings showing important parasite noises, such as antenna files (chains of the hydrophone setting produced continuous transient signals) and recordings made before the hydrophone reached water.

For this analysis, we selected three different noise intervals (on the same file, Figure 9), including transient noise with different amplitudes. The detections were executed with different noise thresholds (see the Autodetection section). The noise intervals used are shown in Figure 9, these noises were tested on four files (with the same reference click from the set provided with DeteClic). The results are shown in Figure 10.

Including a powerful noise (Figure 10.A) see 3) of all four files) (i) leads to lower false alarm rates

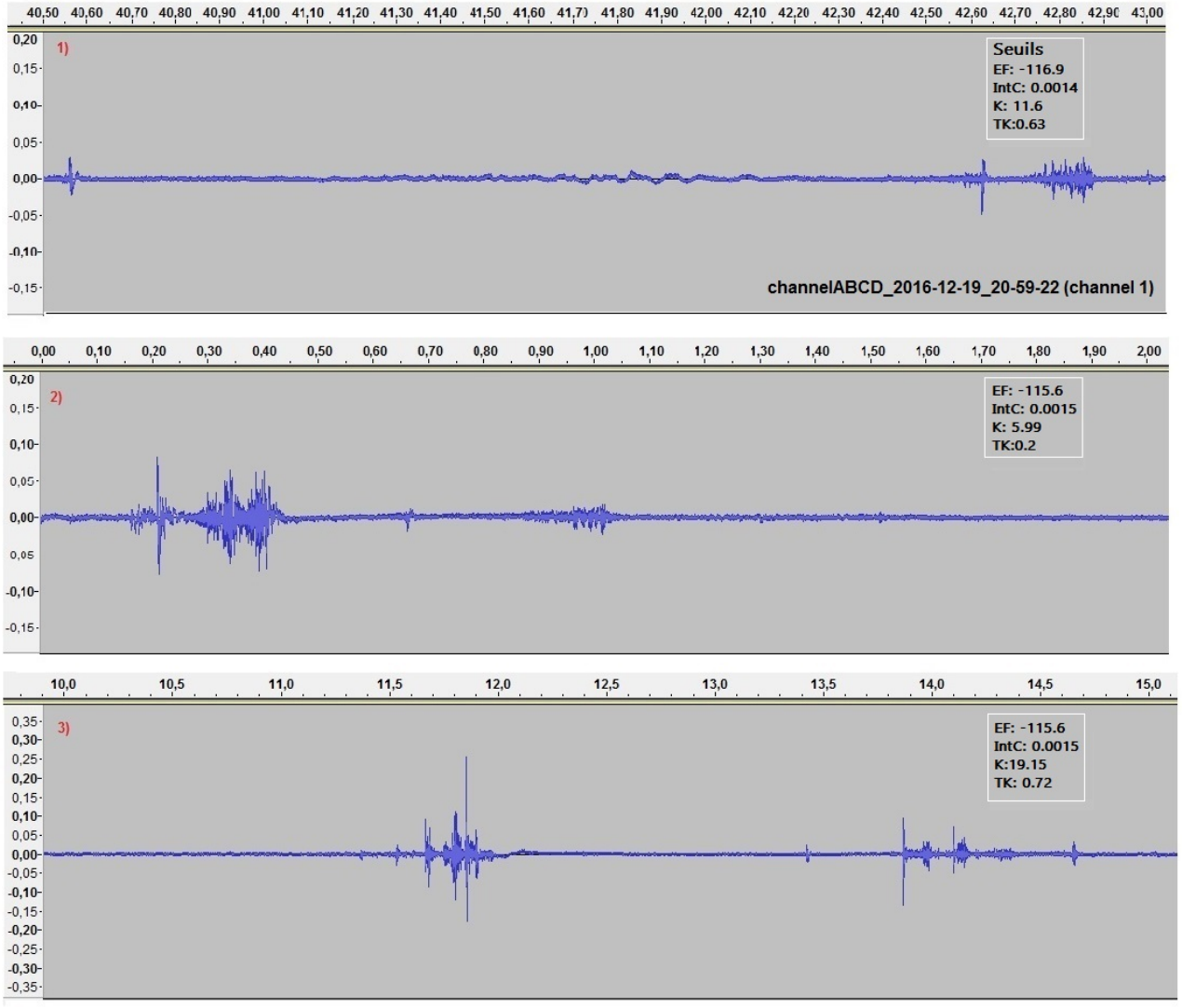


Figure 9: Noise selection, add thresholds!

(in terms of signal mistaken as clicks), and (ii) reduces the number of trains wrongly detected. For instance, considering a stronger noise allowed a correct discrimination of what was otherwise considered to be trains (Figure 10.B), file 22-58-04).Trains are here detected because these antenna files are really noisy, the chains regularly chocked on the hydrophone structure.

The figure 10 also illustrates the strength of DeteClic, that is to combine results of three detection methods allows for a robust analysis. Still, to properly launch the automatic detection, one should first have a look to the audio files to select a representative noise interval. In our example of the antenna files, because we knew there was an important quantity of pulsed parasite noises, we could include them when selecting the reference noise.

We also recommend not to select a small noise interval (at least 20 000 samples).

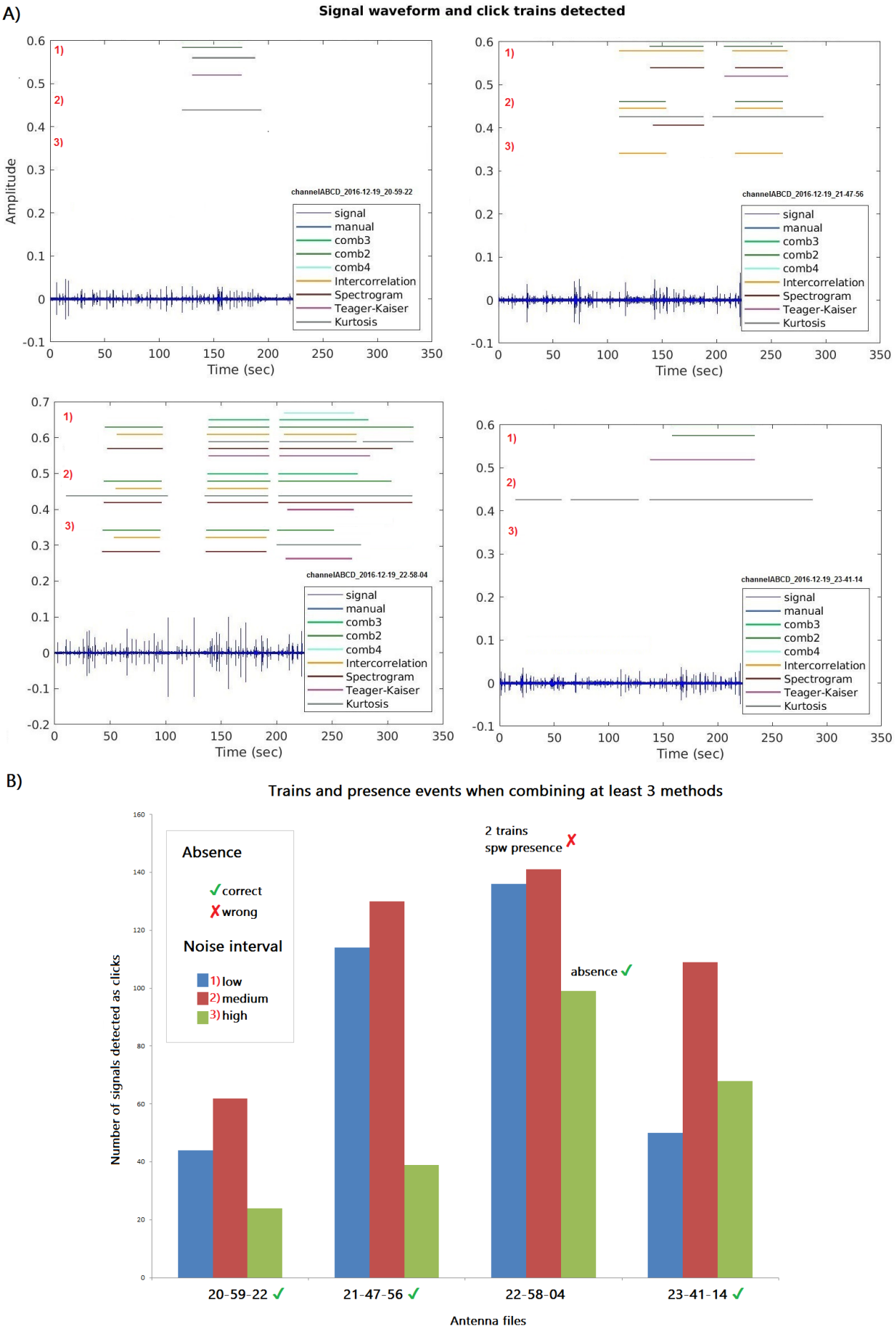


Figure 10: A) Train outputs differ when selecting different noise intervals. Each square shows the signal (blue wave shape) for one file. 1), 2), and 3) shows the trains detected (colored lines) for the three noise intervals presented in Figure 9. B) Number of signals mistaken as clicks for four antenna files, for the three noise intervals (Figure 9).

4.2 What click to choose?

4.2.1 Click pulse structure

Cetacean species show different click structures. For example, it has been argued whether sperm whales show a mono-pulsed or multi-pulsed click structure (Figure 11, Møhl 2003, Schulz 2009, Goold 1995, Drouot 2004, Whitehead 1990). This structure is changed by the signal transmission and depends on the relative position of the animal (Antunes 2010). This signal distortion is very likely to be found when studying other cetaceans.
To launch the spectrogram detection method implemented in DeteClic, the user must select a click that will be used as a reference (see the Autodection section). How you select this click influences the results. We found that starting a same reference click at the 1st or next pulse (Figure 11) resulted in low differences in click detection. However, this might not be the case when the multi-pulsed structure is very marked (if you select a reference click showing only one pulse).

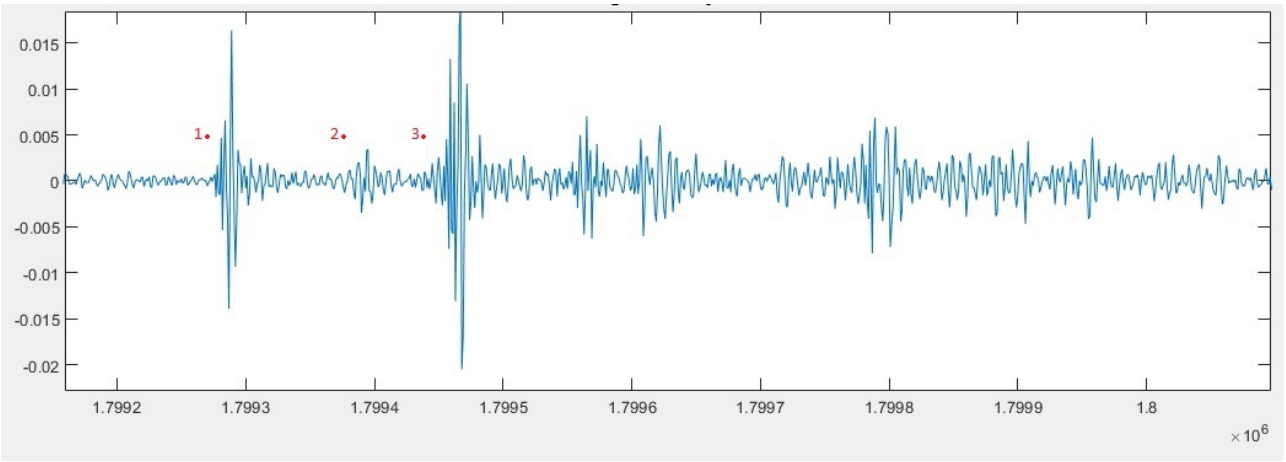


Figure 11: Multipulse structure of a click (Zimmer 2005). Starting the reference click at one of the three red points will induce slightly different detection results.

4.2.2 Reference click

As stated above, the click structure is influenced by several factors (Figure 11 and 12). We advise to select a click with a similar structure from those of the recordings analysed, especially when using one of the reference clicks provided (see the Autodetection section). When launching the DeteClic autodetection, we offer to chose a reference among four different clicks that we have found in our own data set. We tested the difference in quality detection between these four clicks on a 10 files sample (recordings of about 5 minutes). We compared the average precision and recall obtained for these 10 files (Figures 13 and 14).
Choosing an appropriate click will enhance the results of the intercorrelation analysis (its recall rates, Figure 13). But the click selection will poorly affect the results of other methods and their combination (Figure 13 and 14).

5 Noise pollution

5.1 Describing background noise

Our main issue was to differentiate transient parasite noise from clicks, which are quite similar. Manually, we are most of the time able to discriminate them (combining earing, spectrogram, wave shape), but how? How to characterize these parasite noise?

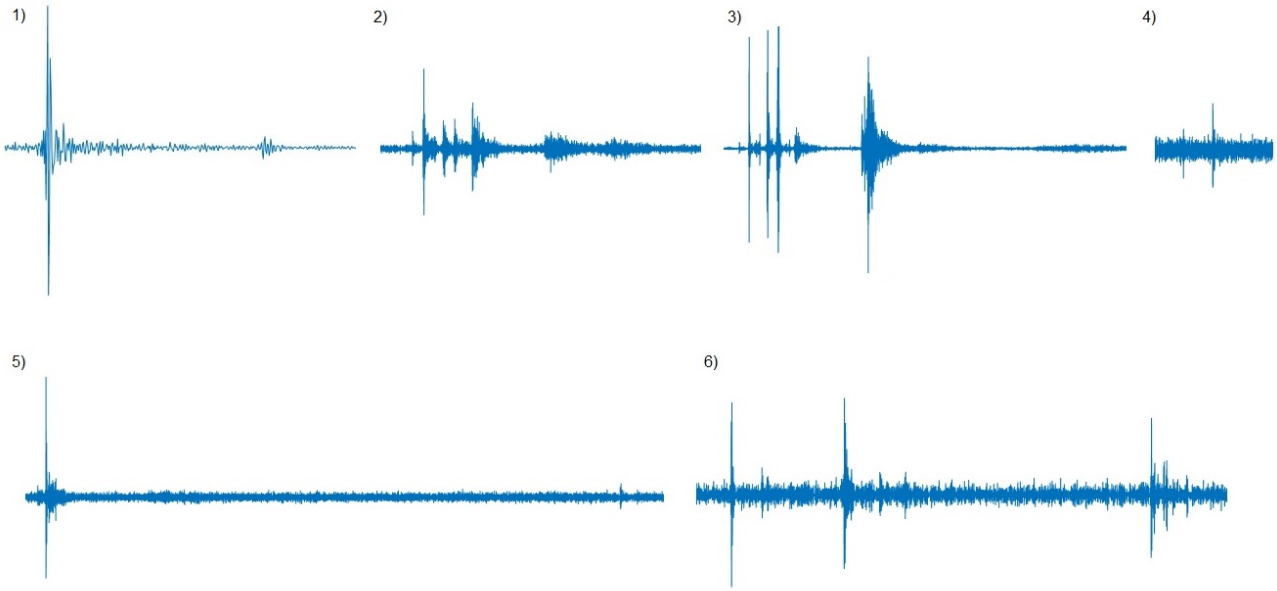


Figure 12: Examples of click found in our data set. They show really different SNRs and wave shapes.

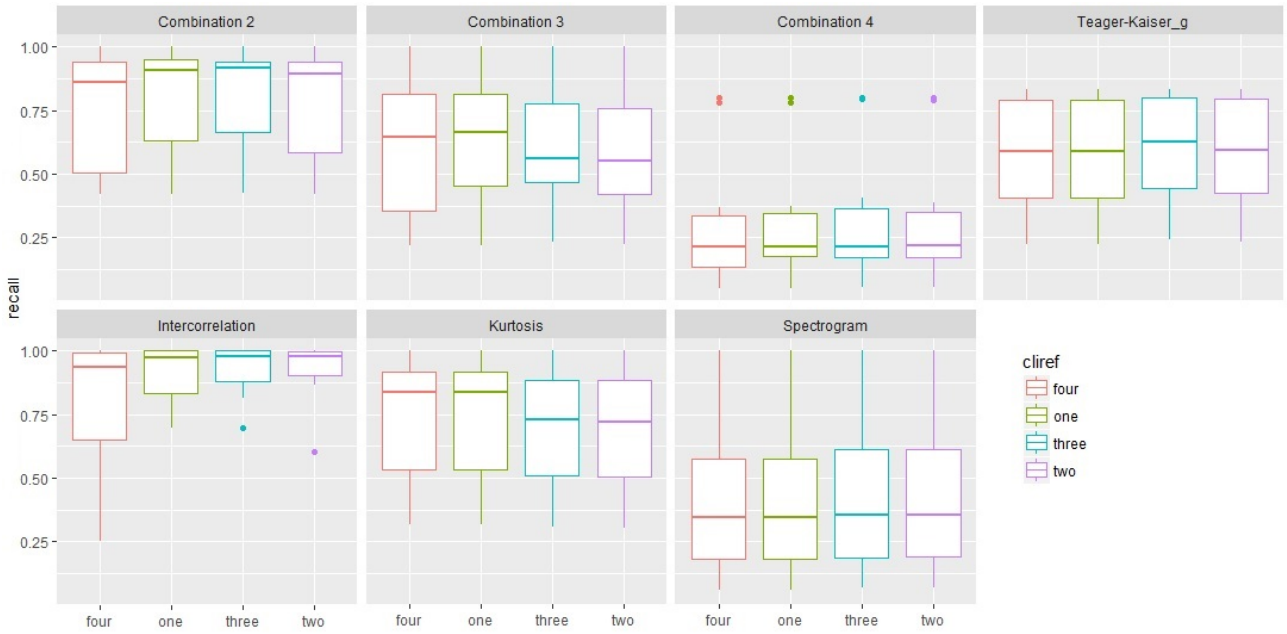


Figure 13: Recall rates obtained when using different reference clicks (described in the Autodetection section).

This is a very complex question, and we struggled in finding specific research on this topic. We aimed at describing a proper ”transient noise” variable, to quantify what is detected as clicks and why. We could not find any in the literature, and counting them manually would be to subjective, so we did not investigate this issue any further.
We strongly advise to filter your recordings (we used a 4000Hz high-pass filter) to eliminate most of the continuous noises, such as sounds produced by boats (McKenna 2012). We considered the ocean ambient noise mostly found in three frequency intervals (Hildebrand 2009, Richardson 2013):

- Low [10-500Hz], anthropogenic sources.
- Medium [500-25kHz], sea surface agitation and anthropogenic noises.

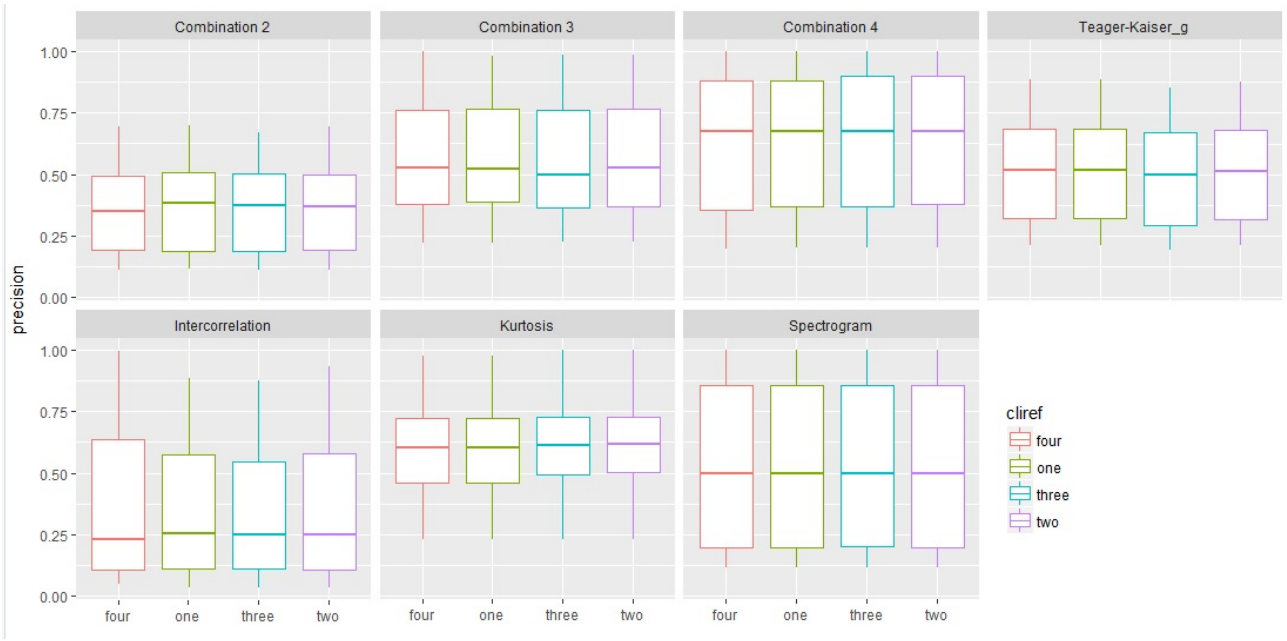


Figure 14: Precision rates obtained when using different reference clicks.

- High [≥25kHz], thermal noises.

The user can adjust these intervals in: Edit ▷ Preferences ▷ Detection ▷ Detection variables ▷ Frequency intervals for noise estimation (see the Outputs section).
We (i) studied the SNR of the signals automatically detected as clicks (comb3). We then wanted to describe the ambient noise. We removed all signal detected by at least two methods, so we could extract (ii) the peak of intensity for the three frequency intervals described above, and (iii) the average intensity in these frequency bands.

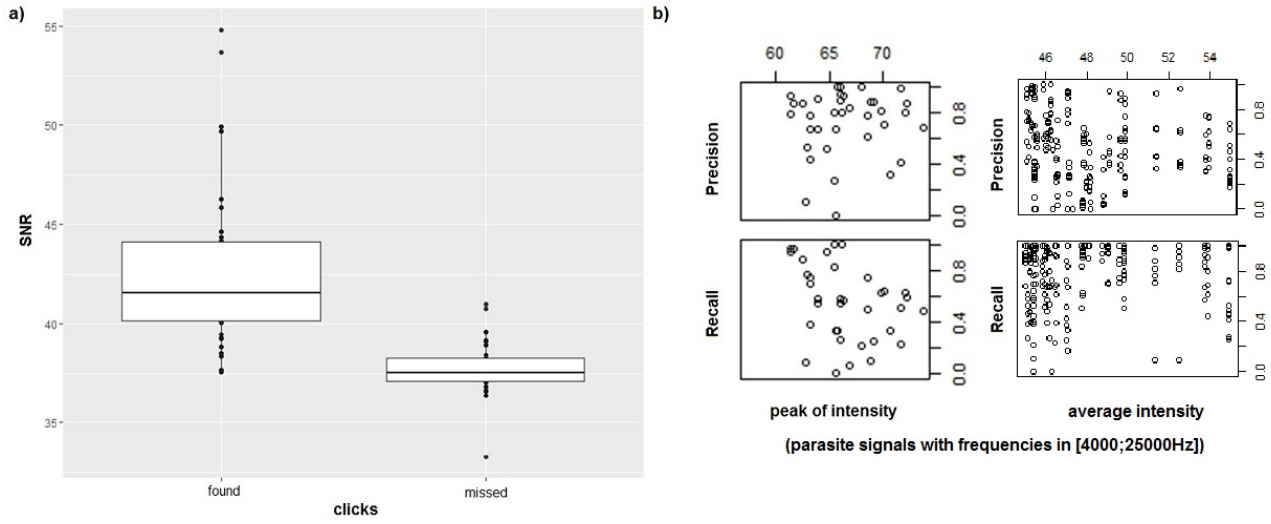


Figure 15: Effect of ambient noise on the detection quality. a) Differences in SNR for clicks that were automatically detected or missed. b) Relationships between the intensity peak of parasite signals and the precision or recall rates.

5.2 Interfering with click detection

(i) First, we found that missed clicks (false negative) showed a lower SNR compared to those that were correctly detected (Figure 15.a). This indicates that clicks with a low SNR (e.g. because of a

high distance from the recorder) would not be detected.
We qualitatively investigated how the background noise (transient or continuous) would influence DeteClic results. The results for the antenna files, showing an important noise pollution, are shown above (Figure 10). It clearly appeared that false alarms were either strong pulse parasite signals or remained unexplained (because we studied combinations of methods with different signal analysis). However, the two variables ((ii) peak and (iii) average intensity of the frequencies in [4000;25000Hz] because of the 4000Hz filter) do not clearly correlate with the detection quality (Figure 15.b.). Again, this illustrates a need for a proper noise description.

D- Comparing DeteClic

We ran both DeteClic and Pamguard 1.15.13 on 46 of the previously presented audio files (see the Audio sample section above). One should note that these two detectors are not to be used for the same type of analysis: Pamguard was primarily implemented for real-time analysis while DeteClic must be used on previously-made recordings. We ran Pamguard using most of the default settings, because we could not find any equivalents of the detection parameters specified in DeteClic. We only changed the minimum click separation to 2800 samples (which corresponds to the dead time we used in DeteClic). We noted the number of clicks found by both detector. Because Pamguard do not offer a tool to compare its automatic detection to any manual labeling, we did not assess precision and recall rates for Pamguard.
We compared Pamguard to the combination of at least three methods in DeteClic (comb3). We first considered the difference between automatic and manual labeling for both detectors:

$$\text{comb3 or pamguard indicator} = |\ \text{number of manually-labeled clicks} - \text{number of automatically-found clicks}\ |$$

We considered the absolute value of differences in number of clicks, and this does not indicate if these correspond to real clicks (and both detectors mostly found more clicks than there were, 70% of these differences were negative for comb3 and 80% for Pamguard). However, if the indicator shows a high value, an important part of these signals are not clicks, so it roughly indicates the detection quality.
Then we calculated the gap between the two methods:

$$\text{gap} = \text{comb3 indicator} - \text{pamguard indicator}$$

A negative value indicates that the number of clicks found by Pamguard differs more from manual labeling than what is found by the combination of three methods in DeteClic.
Both methods showed a significant correlation between number of clicks automatically- and manually-found (Figure 16). Still, Pamguard showed greater deviations from the number of manually-labeled clicks (Figure 17). It found more clicks (false alarms) than DeteClic did. Yet, when no clicks were manually detected, Pamguard was very likely not to detected any false alarm, contrary to DeteClic. One can then argue that this default is rectified through DeteClic train analysis (see the Outputs section).

Finally, we found that using DeteClic is more user-friendly that Pamguard is. DeteClic allows for an easy first handle, results comparison, or easy record. This comparison with Pamguard therefore shows the usefulness and value of DeteClic.

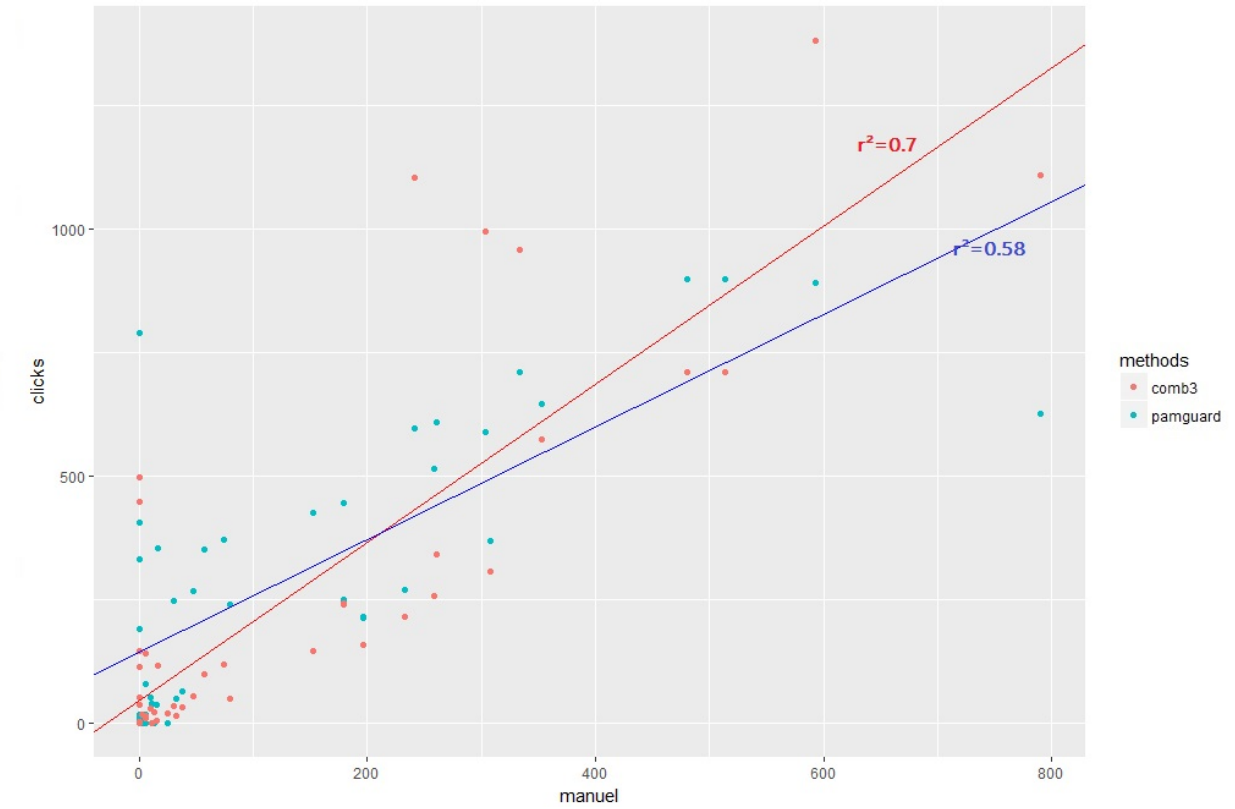


Figure 16: Correlation of the number of clicks found manually or by the two detectors. The regression lines are based on: Pamguard: intercept=-19, slope=0.52,t=8.3, pvalue$\leq$0.0001, DF=44; DeteClic: intercept=19, slope=0.56,t=9.2, pvalue$\leq$0.0001, DF=44)

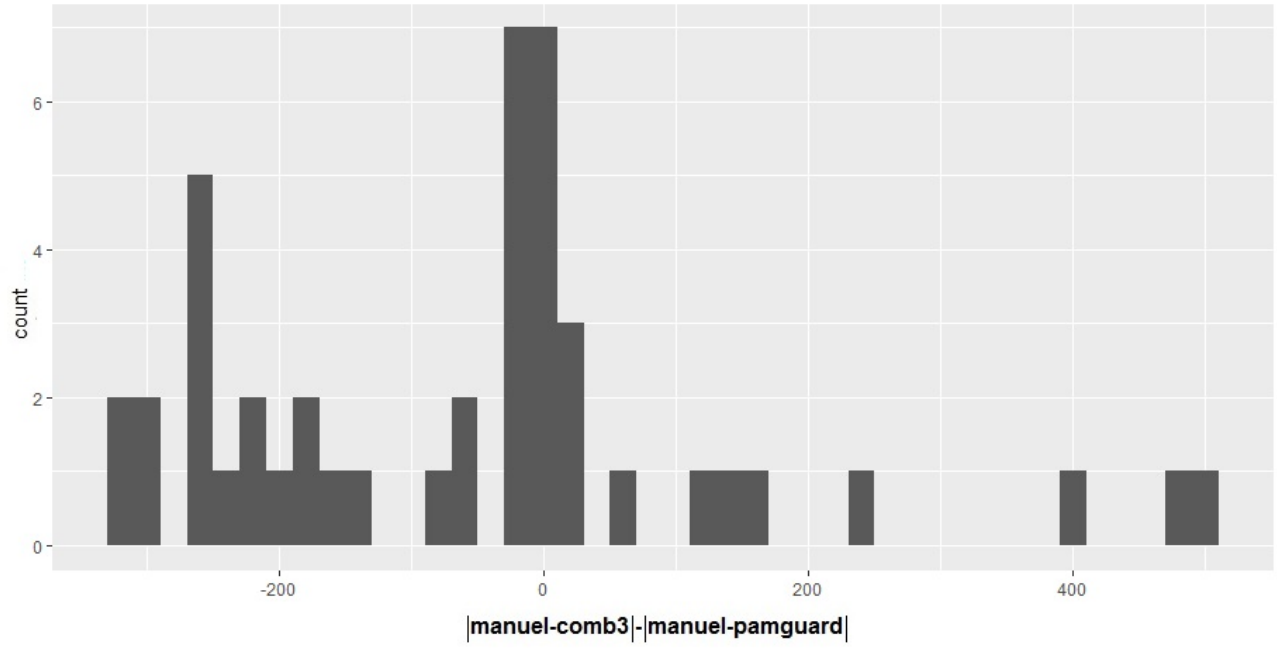


Figure 17: Histogram representation of the ”gap” between detections of Pamguard and DeteClic (i.e. the ordinate axis unity is an audio file). Pamguard detections with a higher number of false alarm will give highly negative gap values. The very high positive value (466) was obtained for a file with high noise pollution but no clicks found.

Detection performances

1 Launching the performance tool

1.1 perf_launcher

calling files2perf and pathname loop for all files calling for audio and loading csv files (manuel+auto)

1.2 Position function

Filling the gaps - Manual detections
We first implemented a pre-filling of 20 ms (considered to be the length of a regular click Whitehead 1990, Goold 1995) for each manually detected click in the detected_man vector (i.e. 0.020*fe samples filled with 1, followed by a 0). This avoids any closely-located clicks to be skipped, as they could be considered as one instead of two clicks
Position vectors
As for automatic detections (see the appropriate section) we do not consider any detection within the dead time following a detected click (as set for the studied species or adapted by the user). As previously explained this filling interval can be adjusted. Based on these detections, we create a position vector for all selected methods (except if no click was found).
ADD ECHOES

2 Performance analysis

2.1 Comparing automatic and manual detections

We based our performance tool on a simple comparison on what is detected by the automatic methods and what is manually annotated. Each manually-detected click is compared the automatically-detected clicks: is the manual click included in an automatic one ? This is done with a tolerance interval (k). For instance, an automatically-found click is included in the sample interval [230, 800], it would look for any manual click found in the interval [230-k, 800+k].
If a manually-detected click corresponds to an automatic one (i.e. it is included in it), the comparison is stopped and the corresponding automatic click is excluded from the analysis. This avoids any automatic click to be twice counted as correct. It then pass on comparing the next manual click.

2.2 Performance indicators

Two outputs describe the automatic detections quality: their precision and recall rates. We calculate them as (Yang 2017,Roch 2013, Gillespie 2013, Towsey 2012):

$$\text{precision} = \frac{\text{number of correct clicks}}{\text{number of clicks found by the automatic method}}$$

$$\text{recall} = \frac{\text{number of correct clicks}}{\text{number of clicks manually found}}$$

The precision gives the false alarm rate while the recall indicates if the method found most of the manually-detected clicks.

2.3 Click trains and presence events

As detailed in the outputs session, a higher level of analysis allows for a better precision, based on a pattern recognition, here click trains. We consider regular and close clicks (low duration between two events) to form a train. The train must last some time. Comparing manual and automatic detection of trains would be challenging, DeteClic can display graphs allowing visual check. One can see if the trains found manually and automatically correctly overlap. This can also be used to verify if all clicks found are not parasite transient noise, if they do not belong to a train event (while only considering echolocation regular clicks, e.g. excluding codas).

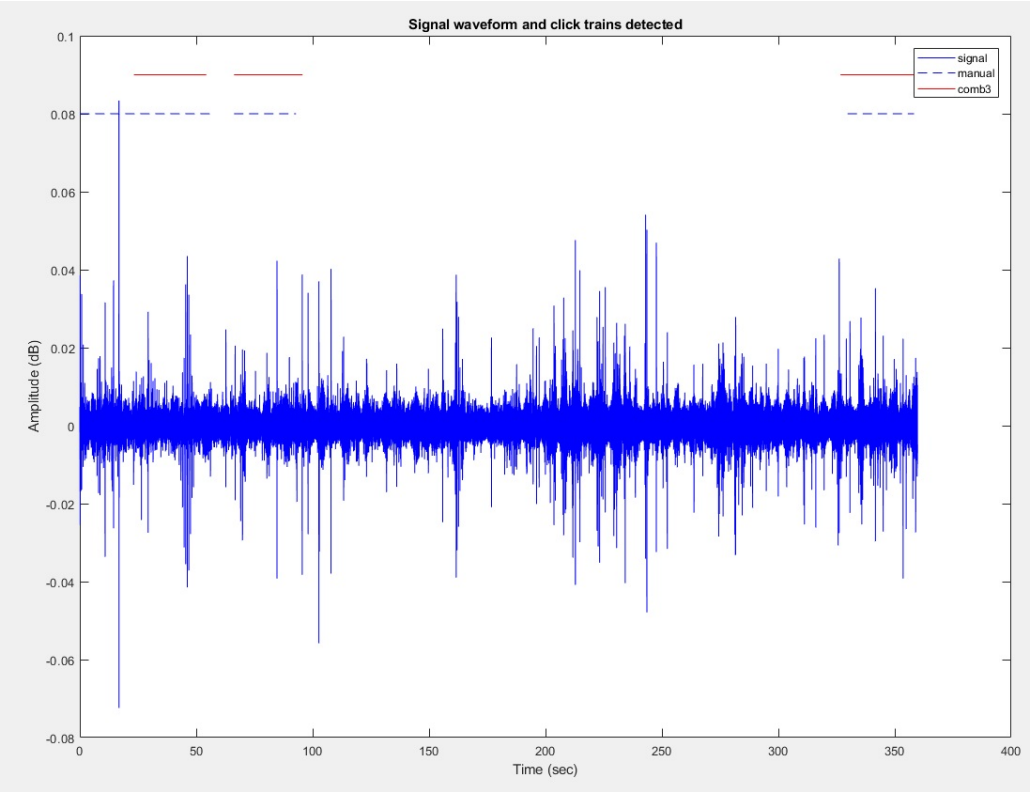


Figure 1: Displaying the signal amplitude over time and the trains labelled manually and detected automatically allows for a visual verification of DeteClic outputs.

3 Results display and record

3.1 Tables

A .csv file is created for each file analyzed (Figure 2). It includes the number of clicks found automatically and manually, with the associated precision and recall for audio files analyzed at once and selected methods.

3.2 Plots

Once the performance analysis finished and the tables saved. A window shows the average precision and recall rates with their standard error for all analyzed files. On the same window, DeteClic also produces visual display of the results from the comparison with manual labeling. Three plots are

17_12_14_47_files_used.csv

	A	B	C	D	E	F
1	methods	file	clicks_number	recall	precision	clicks_manuel
9	Intercorrelation	channelAB_2017-01-16_06-53-55	411	0.97	0.61	259
10	Spectrogram	channelAB_2017-01-16_06-53-55	215	0.57	0.69	259
11	Teager-Kaiser_g	channelAB_2017-01-16_06-53-55	180	0.48	0.69	259
12	Kurtosis	channelAB_2017-01-16_06-53-55	450	0.99	0.57	259
13	Combination 2	channelAB_2017-01-16_06-53-55	502	0.96	0.49	259
14	Combination 3	channelAB_2017-01-16_06-53-55	250	0.67	0.69	259
15	Combination 4	channelAB_2017-01-16_06-53-55	141	0.38	0.7	259

Figure 2: The performance analysis produces one .csv file for all .wav files analyzed. For all recordings, it indicates the number of clicks both found manually and by each automatic methods (clicks_number), with the associated recall and precision.

produced: (i) the number of manually- on automatically-detected clicks for all possible methods (Teager-Keiser on the Figure 3 example), (ii) boxplot of the precision and (iii) the recall rates for all selected methods (Figure 4).The user can choose between the three to display, and if they are to be saved (as .jpeg files).

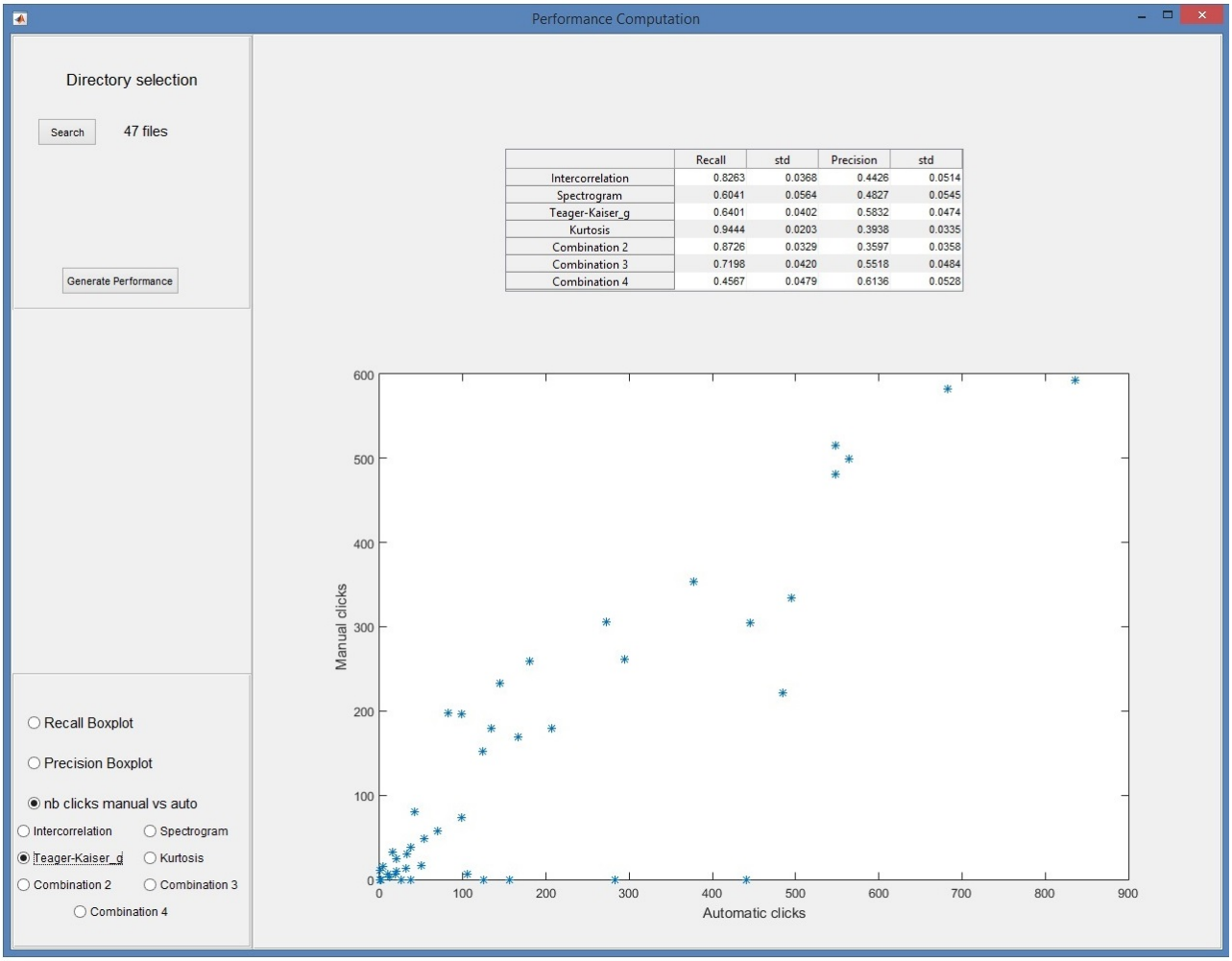


Figure 3: In addition to the rates and their variation, the user can chose to display the number of clicks found manually against that automatically detected.

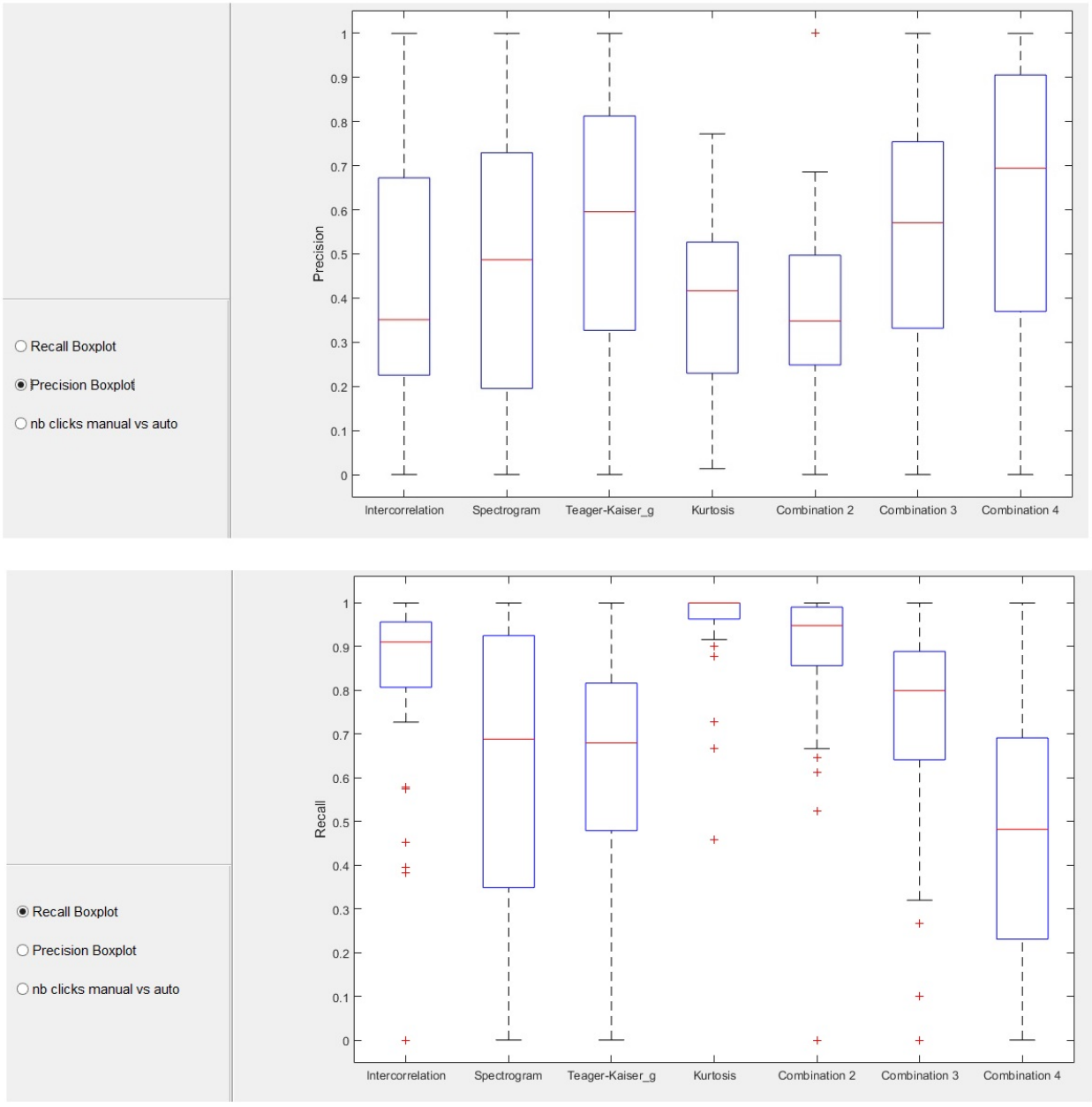


Figure 4: The user can also display and save boxplots of the recall and precision rates, as these plots are automatically printed at the end of the performance analysis.

DCL 2018 Abstract Submission Form

Please use this form for all abstract submissions. If multiple affiliations are required, then please indicate the affiliations of each author.

Title Introducing DeteClic: a user-friendly and comprehensive automated click detector to monitor odontocetes

Preferred Format (Oral / Poster)

Authors Fabio Cassiano Juliette Biquet Flore Samaran Angélique Drémeau Christophe Guinet
Affiliations Laboratoire Lab-STICC (UMR 6285, ENSTA Bretagne) Laboratoire UMR CEBC (Chizé)
Contact Email fabio.cassiano@hotmail.fr juliette.biquet@u-psud.fr
Presenter Fabio Cassiano Juliette Biquet

Uses workshop dataset? no

Abstract (300 words maximum)

Passive acoustic monitoring is now intensively used to study cetaceans. Given the growing amount of collected data, it is of interest to implement automatic methods to monitor cetaceans' sounds. In that aim, we developed DeteClic, a transparent and user-friendly odontocetes click detector on Matlab. This detector takes the form of an intuitive interface to analyze pulsed and loud acoustic signals. DeteClic can be used along the entire bioacoustics analyzing process, from the display of graphic representations to the click detection and their understanding. Our detector allows for (i) manual labeling, based on both listening and visualization, and (ii) automatic detection. DeteClic includes 4 different automatic click detection approaches: (i) a Teager-Kaiser energy operator, (ii) an intercorrelation computation with a given reference signal, (iii) a spectrogram analysis, and (iv) a kurtosis-based statistical detection. Combining the results of these 4 methods enhances the automatic detection robustness. To assess its performances, we equipped DeteClic with an evaluation tool, relying on automatic comparison of manually- and automatically-detected clicks. DeteClic can therefore be used to compare the results of the methods (i.e. recall and precision rates) depending on acoustic environments and recordings. In addition to the clicks detection and based on how the user defines a train of clicks (using criteria in time between clicks and train duration), DeteClic enables a presence event analysis. Finally, we provide a series of outputs easily reusable for any further statistical or visual analysis. We first tested DeteClic on sperm whale clicks, recorded during longline fishing campaigns around the Kerguelen Islands (southern Indian Ocean). Considering clicks detected by at least three methods was the most efficient approach (average recall of 73+-4% and precision of 56+-5%). Discriminating click trains allowed for a greater precision in detecting echolocation events and an accurate presence event analysis.

Notes Any additional information to workshop organisers ?

Analysis parameters

1 Pattern event recognition

We decided to base our presence analysis on the detection of click trains. We used two criteria (see Automatic detection results section) 1) duration between clicks (ICI) and 2) train duration. For this analysis, we used the combination of at least three automatic detection methods (here this will be referred to as automatic detection).

1.1 Interclick intervals

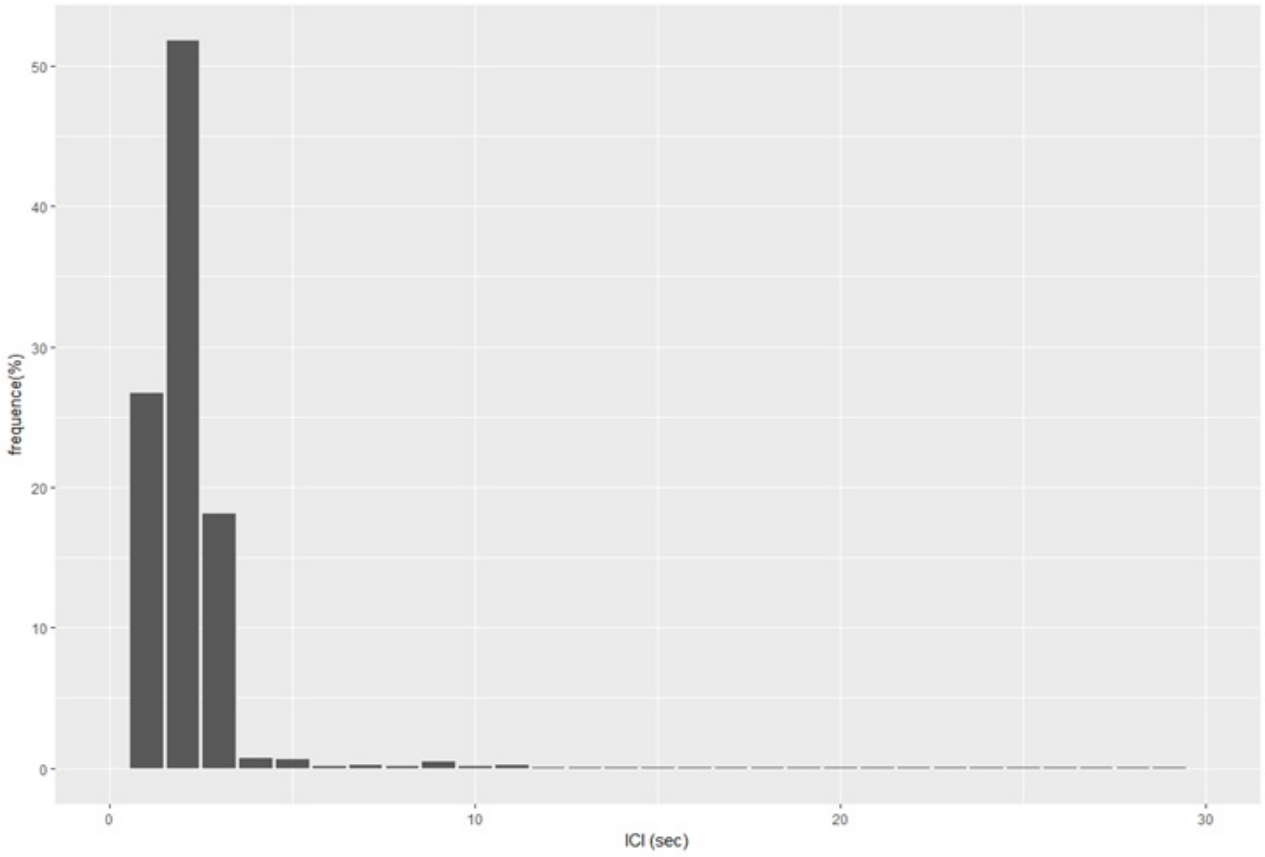


Figure 1

Interclick intervals are commonly described between 0.5 and 2sec (they can be lower within creaks). Here, we allowed for more time between clicks to account for any missed ones (as it would interrupt the train otherwise).
We used the broken stick method (as Sonderegger et al (2009) did). Over 9000 ICI were extracted from 41 files. The analysis was ran on manually- and automatically-labelled clicks. We used the SiZer R package to calculate the ICI threshold for (i) 10 individual files and (ii) all the files.
Files with no manually-detected clicks showed great ICI variance and high ICI thresholds. Visual analysis also confirmed any event of presence could be discriminated by ICIs (as false alarms showed

random ICIs). For all the files, the manual detection showed an ICI interval of 4.67sec while it was 5.35sec for the automatic detection. Therefore, we chose a 5sec threshold.

1.2 Train duration

CHANGE ALL THIS FOR NEW STUDY
REDO FIGURE WITH TRAIN SIZE IN THE PROPER ORDER
Considering a 5sec ICI threshold led to a constant correlation between the number of trains manually and automatically found. We could not find any description of natural train duration, which of course greatly varies. Still, it would have been interesting to have an average minimum of train duration.
We tested what train duration (5, 10, 20, 30, and 40sec) gave the best results on the same audio samples. We also tested whether we should consider ICI $\leq$ 2.5sec or $\leq$ 5sec. As confirmed by the calculation of an ICI threshold, 5sec gave a more relevant train analysis (these results are shown here, Figure 2). All duration tested except 40sec showed train false alarms. For all, the number of trains was strongly correlated with the number of manually-detected clicks. Again, 40sec showed the best correlation between the number of trains automatically- and manually-detected (Figure 2).

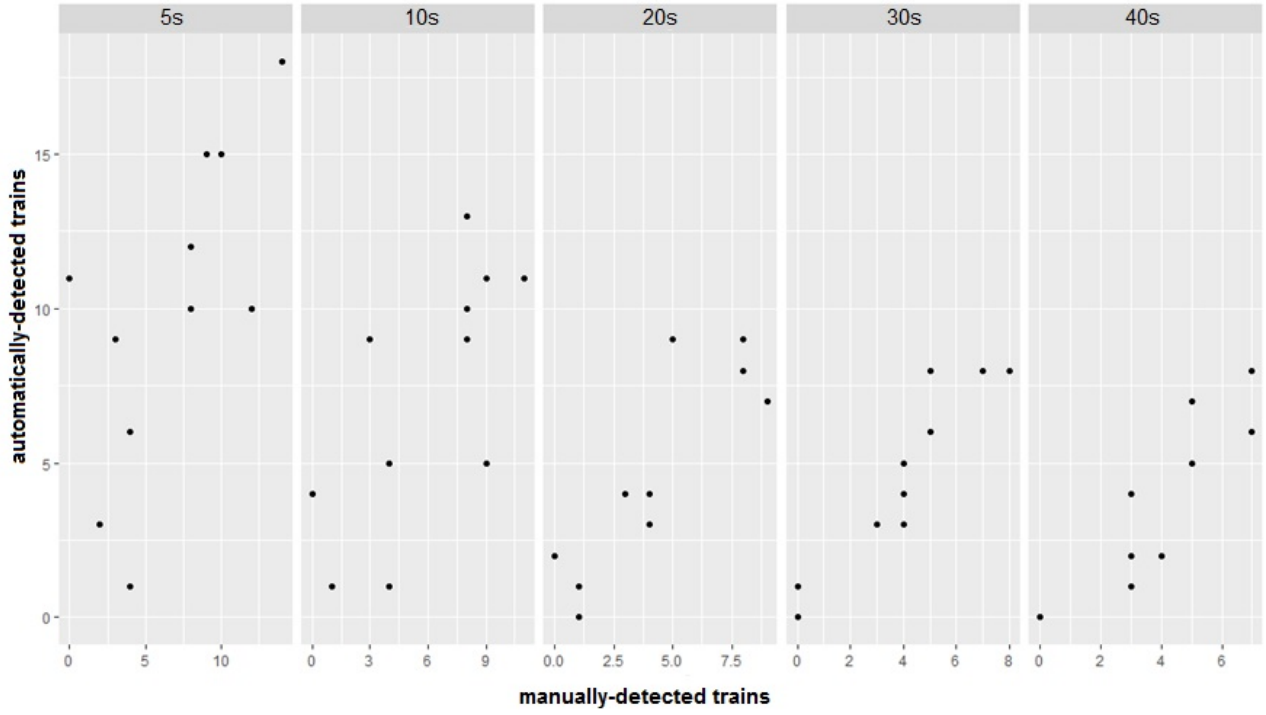


Figure 2

Sperm whale detection protocol

1. ref: pour ce jeu, tous les 5 fichiers, prendre celui du milieu comme ref, soit une ref toutes les 30min
   clic ref : un pour le jeu de données
   Attention, le clic de référence doit avoir la même fréquence d'échantillonnage que le fichier
   bruit ref: sur le fichier du milieu, 1sec sur la 1ère fen ou suivante si besoin, avec bruit transitoire si possible

2. trains
   ICI min: 5sec
   train duration: 20sec
   dpdm:20sec

3. presence
   1 train

4. Filtrage filtre passe-haut, 4000
   filtre passe-bande, 20 000