**Response to reviewers**

Associate Editor Comments to Author:

I appreciate the authors' patience, and I have now received two reviews of the manuscript and package. On whole they both see the potential value and interest of the manuscript and package, as do I. There is some significant work to be done, however, before the manuscript might be found suitable for publication. Both reviews are quite extensive and offer a suite of valuable suggestions which I would encourage the authors to consider carefully. I won't reiterate their specifics, then, but would emphasise and add a few additional points:

1. I agree with both reviewers that it is important to further emphasise what distinguishes ohun from related packages, of which there are quite a few. There is clearly novelty and methodological advancements presented here, but at present it is difficult to get a sense of where this package sits in the broader ecosystem, and hence when users might reach for it.

*Response:*

1. I appreciate the manuscript was (presumably) written in RMarkdown, but the inclusion of code and its output as images makes for difficult reading, and isn't in keeping with the typical style of an applications article. Code can simply be included as specially-formatted text, either inline or on its own line, while any images should be true figures, complete with legend. This will necessitate some re-organisation of the manuscript as there are quite a few such blocks, but feel free to take a look at some recent examples for ideas on how it might proceed.

*Response: We have fixed the format of code and its output so it can be copied. As most outputs were in the form or data frames we included them as microsoft word tables.*

1. Regarding the package, I agree with the reviewer that the code is tricky to read (which people will), owing to some very long lines. Typically it's recommended that lines not exceed ~80 characters for readability. You might consider using a package like styler() to do the heavy lifting.

*Response: This is a great suggestion. We have formatted the code using the styler package. That said, some long lines of codes were necessary as we wanted to avoid duplicated large objects within a function call (for instance wave objects). Long lines of codes can be split into shorter codes in which the output is saved and then used by the following code. However, this common way to make code more readable can be problematic when dealing with large R objects. These was found to be an important limitation in previous versions when working with relatively large sound files. Therefore the solution was to apply all modifications into a single line of code. We agree this long lines are not very friendly for users inspecting the code, but not doing it would severely affect the memory efficiency of the package.*

Reviewer: 1

1. Line 171: is mean the most useful here, rather than median or mode? Examples of how a user would use this metric would help to justify the choice of mean.

*Response: Under our experience the amount of overlap tend to be normally distributed, which makes the mean a good descriptor. We have added that “'Split positives' and 'merge positives' allow users to infer whether target signals are being detected as discrete units. An optimal routine should present no split and merge positives. This index is mostly relevant for energy-based detection. 'Overlap to true positives' is useful when trying to precisely identify the start and end of target signals. A perfect detection must result in complete overlap.”*

1. Line 173: Typo > ‘tuning\_parameters’

*Response: fixed.*

1. Line 173: ‘Here’ – should there be a figure reference?

*Response: The ‘Here’ referred to the following chunk of code. We have reworded that to make that clearer: “ The following code show the first ten files detailed by the ...”*

1. Line 174: Typo>’combined’

*Response: fixed.*

1. Line 171-174: I don’t really understand what is being described here

*Response: We tried to explain that the indices can be computed for the entire acoustic data set of independently for each sound file. We have clarified this in the text: “*By default the function computes indices across all sound files in the data set. However, the function also allows detailing those indices separately for each sound file (argument ‘by.sound.file’)”.

1. Line 236: Is it possible to indicate what a 'good' division of space (and therefore templates?) and a bad one might look like? Why would I want/need to provide my own dimensions?

*Response: Space division is fixed so there is no good or bad divisions. That said, this makes more sense when sounds are homogeneously spread across the acoustic space, so the templates represent an similar portion of the overall population of sounds. We have added the following sentences to the text: “This method is better fit for acoustic spaces in which sounds are homogeneously distributed as templates might represent similar portions of the overall population of sounds in the data. However, it can still be useful for identifying structurally diverse templates in unequally distributed acoustic spaces.*”

Providing your own acoustic space allows user to select which acoustic features are used for creating that space. We have made this explicit in the text: “ Users can also provide their own acoustic space dimensions (argument 'acoustic.space'), which allows users to customize the acoustic space by specifying the features used for projecting it.”

1. Line 248: highest performance being defined as highest f1?

*Response: Yes. We have clarified it in the text: “Table 1 shows the two highest performance runs (identified as the highest f1 score) for each template.”*

1. Line 293: needs additional explanation, not sure what hold time values are, or why you need 3 max/min values

*Response: We have clarified that “This function runs a detection for all possible combinations of tuning parameters.* ***Several values for each tuning parameter can be evaluated in a single run.*** *The code below tries three minimum duration and maximum duration values and two hold time values”. We added the explanation of hold time as a in-line comment to the code as suggested by the 2nd reviewer (“ # time range in which to merge detections into a single one (ms)”)*

1. Line 305: Typo? Maybe should be 0.96 as in figure?

*Response: fixed.*

1. Line 315: is this an additional tool? More of a tip/trick?

*Response:*

1. Line 317: Typo > ‘stereotype’

*Response: fixed.*

1. Line 325: Typo > ‘addition’

*Response: fixed.*

1. Line 327: but most/all of these functions are available in large ML packages like caret

*Response:*

We agree that functions to compute signal detection theory indices are found in other packages like caret. However, there is a lot of coding needed for converting the output of an sound detection routine into a caret friendly format. Overlapping signals need to be determined and then bipartite matching graphs are used to pick up the best match for each detected signal based on some criteria (e.g. highest correlation or highest amplitude). Then true/false positives and false negatives need to be labeled. Calculating the indices is the easiest part. In fact the indices are so straightforward to calculate that you don’t really need a dependency for this (e.g. caret is not a dependency in ohun). Therefore, we think there was a true need for tools like ohun or sed-eval (that's why we ended up creating the package, there was no other way to do it).

1. Line 327: What platform – R?

*Response: sed\_eval is a Python. We have added this to the text: “* in addtion to *ohun* there is only one other performance-evaluating software developed in a free, open-source platform (**Python package sed\_eval**, Mesaros 2016)”

1. Line 344-346: citation needed

*Response: fixed: “ Deep learning methods tend to require greater computational power, larger training data sets* ***(Stowell 2022)****, and, in some cases, more complex training routines (****e.g. data augmentation,*** *but see transfer learning approaches).”*

1. Line 354 on: These feel a bit out of place. Can they be in the 'Other tools' sections which could be expanded to 'Other tools and tips', or maybe moved to an appendix?

*Response:*

1. Line 378: Don't fully follow here. Again feels out of place in discussion

*Response:*

Reviewer: 2

1. It is not clear from the manuscript how the package can be installed. The link to GitHub is missing. There is also no mention of CRAN as far as I can see. In the code-check-file there is the suggestion to install from GitHub (why not CRAN?), but this will not be available to readers after review. It would be good to include this under Supporting Information.

*Response: We have added links to the github CRAN repositories and to the package vignette in the main text. We also included the installation code for both CRAN and github on both manuals in the supplementary materials.*

1. Generally the code on GitHub has very messy indentation. It makes it difficult to read the if statements in the functions. This can be solved by using ‘reindent code’ or ctrl/cmd + I.

*Response: Thanks. We have followed the suggestion by using the package styler to fix indentation.*

1. The code blocks in the paper are screenshots, which makes it impossible to copy paste the content. Also, no scripts are supplied (or I have not been able to find them), so users have to copy paste every chunk of code out of the html files. It would be much more useful if there was an .rmd or .R file with all code. I did finally figure out there was an rmd file in the vignettes folder, maybe make this more clear.

*Response: We have included the code as text in this version.*

1. In the discussion I’m missing an overview of which methods from the literature can be useful. The package presented here is very limited in the signal detection part, and it’s strength really is the comparison. I think a guideline in the style of “if you have PAM data with multiple species and you want to count vocal activity for specific species, this methods (REF) might be useful; but if you have…” is very much needed. Stowell reviewed existing methods, but a table or paragraph summarising their conclusions would make this paper much stronger.

*Response:*

1. When running devtools::check() on the GitHub package, it takes a very long time to compile the vignettes. This makes checking the code very difficult. I gave up after several hours and could therefore not complete all the checks. devtools::test() passed all tests, and obviously you already tested this on multiple systems to submit to CRAN.

*Response: Yes the code was tested on multiple systems. devtools::check() work fine for me so not sure what is going on. Note that the package is currently under review in RopenSci (https://github.com/ropensci/software-review/issues/568#issuecomment-1380211327) which requires adding code coverage and continuous integration to the package.*

1. l. 95 there is a space missing

*Response: fixed.*

1. l. 118-119 this sentence is not clear to me

*Response:*

1. l. 160 these values seem very arbitrary and suggest authors have optimised performance (which is not stated in the text), or is this because the times correspond to standard sample intervals (256, 512, etc.)?

*Response: Yes, those values were obtained through manual optimisation. We have explicitly stated that in the text: “For instance, the following code evaluates a routine run in Raven Pro 1.6 ([Charif et al. 2010](#ref-Charif2010)) using the “band limited energy detector” option (minimum frequency: 0.8 kHz; maximum frequency: 22 kHz; minimum duration: 0.03968 s; maximum duration: 0.54989s; minimum separation: 0.02268 s;* ***values obtained through manual optimization****) on a subset of the zebra finch recordings described below”*

1. l. 161 why are there two code blocks? Could you explain the argument `by = “turning\_parameters”`? It might be useful to show the `head()` of the csv files so that readers know what goes into the function.

*Response: We removed one of the code blocks. We have added that “The argument 'by' in the code above allows users to defined a column indicating which detections (i.e. rows) belong to the same run.”. We also added a call to head() to show what input data goes into the function as suggested by the reviewer.*

1. l. 163 it would be useful to explain “bipartite matching graphs”

*Response: We have added that: “The diagnose\_detection function employs the maximum bipartite matching algorithm (Csardi & Nepusz, 2006) with optimization achieved through the push-relabel algorithm (Goldberg & Tarjan, 1988). The matching process takes into account the proportion of overlap between candidate detections, using it as a weighting factor. This algorithm aims to maximize the assignment of detections to target sounds, optimizing the matching between detected and reference events. As a result, each reference sound is exclusively associated with a single detection (Lostanlen et al. 2019).*”

1. l. 166 it would be useful to include how recall etc. is computed for partially overlapping detection/reference. If a reference is overlapped partially by one or multiple signals, does that count as TP (for both)? What if the most of the detection is outside the reference?

*Response:*

1. l. 169 I don’t understand the “merged positives”. How can multiple detections overlap? Or is the section mention of detection referring the reference base?

*Response:*

*We have clarified the description of merged positives: “ 'merged positives' is the number of detections that overlap more than one reference sound”*

1. l. 183 I think it would actually be useful to do a direct comparison for both methods. Now users are left to wonder if these methods are better or worse than existing implementations.

*Response: We appreciate the reviewer's suggestion to include a direct comparison between the methods used in our study and existing implementations. We agree that such a comparison would provide valuable insights for readers. However, we encountered challenges in conducting a fair and comprehensive comparison due to our limited experience with automatic detection in Raven. While we acknowledge the importance of benchmarking our methods against existing implementations, we believe that our focus in this study was primarily on the novel aspects of our approach and its potential contributions to the field.*

1. l. 189 it would be useful to give suggestions for these methods

*Response: We have pointed out readers to “Stowell 2022 for a detailed review of available methods”.*

1. l. 205 it is not clear that the “time window” refers to the spectrogram, maybe clarify this

*Response: We have added that “Annotations were made* ***on the spectrograms*** *with a time window of 200 samples and 70% overlap and were then imported into R using the package Rraven (Araya-Salas 2020).”*

1. Figure 1. it would be useful to add the scales on the figure, now readers have to calculate which frequency is in the middle

*Response: I agree with the reviewer. However, when adding the axis to all spectrograms they look way smaller. So we prefered to keep it as it is.*

1. l. 217 maybe elaborate a bit more. I would also move the explanation before the justification. Now readers that don’t know the methods, don’t understand the justification.

*Response: We have moved the justification after the explanation of the method and elaborated more on its details: “Template-based detection uses spectrographic cross-correlation to find sounds resembling an example target sounds (i.e., template) across sound files. The method produces vectors of correlation values through time, in which a correlation threshold can be applied to separate detections from background noise.”*

1. l. 222 this is a very useful function!

*Response: thanks!*

1. l. 242 I would suggest reformatting the code so that lines align. Also, it might be useful to add comments after arguments with short explanation in the chunk, so that users can copy paste and do not have to go back to the manuscript to understand them.

*Response: This is a great suggestion. We have standardized indentation of code and added in-line comments to must of the code.*

1. l. 259 for this method it would be useful to generate a consensus detection table. The advantage of multiple templates is that you can use all, and then raise the detection threshold. Outlier calls would still be detected by one (but not all) of the templates.

*Response:*

1. Figure 2. I would suggest to export the figure as PDF. Now the resolution is very low.

*Response: We included a higher resolution version this time.*

1. l. 268 I would suggest to explain the method. Did the authors use an envelope and amplitude threshold, or how was the “energy” used for detections?

*Response: We have added that: “This method applies a threshold to amplitude envelopes to infer the position of sound events.”*

1. l. 294 I would explain the argument in the code chunk

*Response: We have added in-line comments explaining the arguments in the code chunk (“*hold.time = c(0, 5), # time range in which to merge detections into a single one (ms)”)

1. l. 307 I would suggest to add an option that ensures training and testing data is representative. E.g., does the code take into account that some recordings are from the same day or same individual? These should be equally present in training and testing data. On the other hand, if the test performance is a metric of general performance, I would suggest training on a subset of individuals/days and testing on another subset. This way, researchers know the performance even when they add new individuals to the dataset later.

*Response: This is a great suggestion. We have added the argument “macro.averaging” to the function diagnose\_detection() which gives equal weight to the different sound files for estimating performance indices. Stratified sampling is also a desired approach as mentioned by the reviewer.However, instead of implementing the stratified sampling method ourselves, we decided to recommend the use of the caret package, which offers a straightforward and well-established implementation of stratified sampling. The section reads now as follows:*

*“Additional measures might be needed when working with unbalanced data sets. The function diagnose\_detection can use macro-averaging for estimating performance indices to give equal weight to each sound file (Mesaros et al. 2016):*

diagnose\_detection(

reference = test\_ref, # data frame with reference annotations

detection = det\_test, # detection data frame to be diagnosed

by.sound.file = FALSE # summarize across sound files,

macro.averaging = TRUE # calculates average of within sound file averages

)

*Stratified sampling that accounts for additional structure in the data (e.g. several individuals, populations, days) might also help to deal with unbalanced data and ensure similar performance on unseen data (see createDataPartition in the R package caret for creating stratified training samples).*

1. l. 326 there is a typo in sed\_eval (sed-eval) which makes it hard to find if you Google it, otherwise I really wouldn’t mind. Also, many machine learning implementations are open source and do report performance metrics, but it is still hard to compare metrics on your own dataset.

*Response: fixed.*

1. l. 342 without a comparison this is hard to judge. The reader has no way of judging how difficult the datasets that are presented really are.

*Response: We agree with the reviewer. However notice that we emphasized that our examples show the methods can work well under “appropiate conditions”. So we are not claiming the data sets are difficult or that the methods would always perform fine on difficult data sets. We are only saying that under some conditions it can work well.*

1. l. 358 is this really true? I would think SPCC is very time-consuming.

*Response:*

1. l. 365 this is a good point! On that note, this study used 500 kHz for calls up to 44 kHz. It would be great to show an example of how to downsample these before running the template based analysis (which was run on a narrower range, but computing the spectrogram will still have taken much longer).

*Response: I agree with the reviewer. However, due to the constraints of the limited space available for application manuscripts, we have to reduce the number of example code chunks in order to accommodate other essential components.*

Example code for Thyroptera:

1. It would be useful if functions also work when the audio files are not in the working directory. Now I get the error: “Error: The sound files are not in the working directory”. There is no indication that the working directory needs to be set (which I think would also make the package less flexible).

*Response: We agree with the reviewer. Now all function have the argument “path” that allow users to set the location of the files.*

1. catalog() outputs a jpeg in the working directory. It would be more flexible if the user can specify the output location. Otherwise it’s not possible to keep the working directory organised with a folder structure. The default can still be `getwd()`.

*Response: We agree with the reviewer. We have added an explanation to let readers know how to find the output of the function.*