Introduction

The goal of this project is to be able to automatically identify potential beaked whale encounters in recordings. To achieve this, we use Pamguard's matched template classifier with a suite of beaked whale templates to try and identify potential beaked whale echolocation clicks.

Event creation and labeling

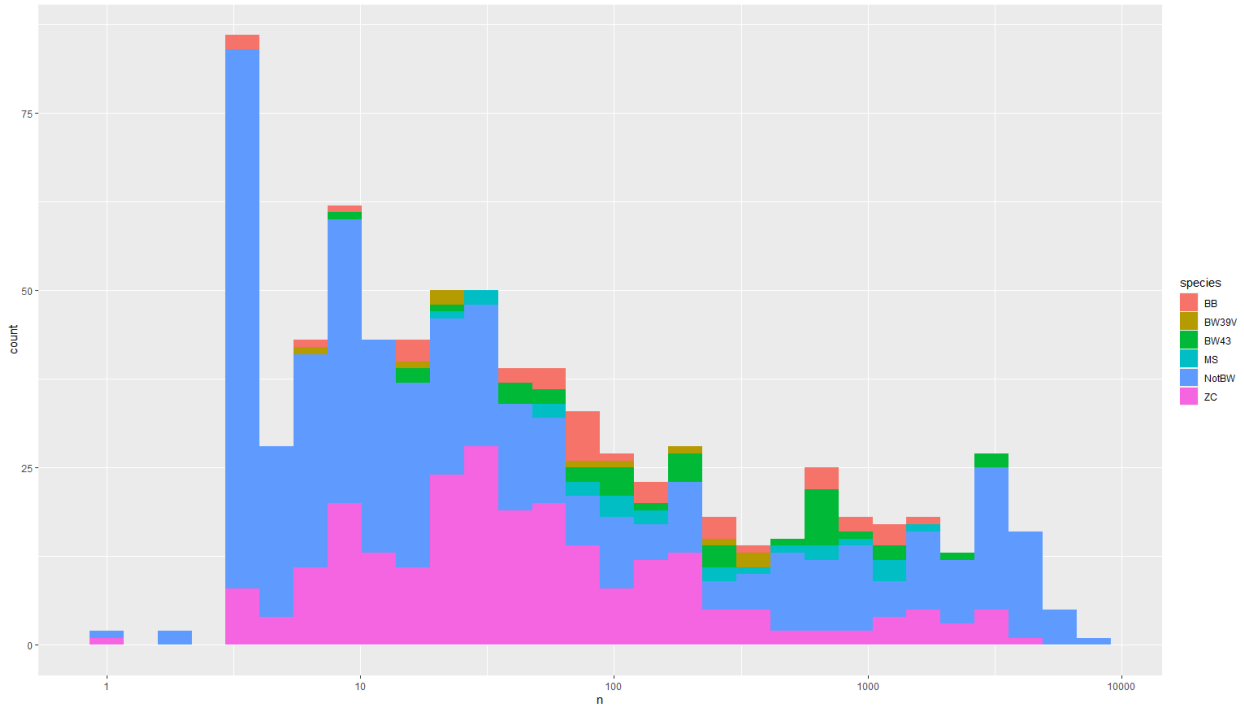
Candidate beaked whale events were identified in 4 steps:

1) The matched template classifier was run using PAMGuard version 2.02.09f (modified version from Doug & Jamie) using 6 different beaked whale templates including ZC, BW43, BW39V, MS, BB, BWC
2) Match template scores were read in from the binary files for all click detections
3) Potential beaked clicks were marked if the template scores were greater than c(.55, .45, .6, .5, .25, .6)
4) A match template classifier (MTC) event was created if there were at least 3 potential beaked whale clicks within a 2 minute window. Events were limited to a maximum duration of 2 minutes, or if it had been more than 2 minutes since the last potential beaked whale click

MTC events were created for a set of drifts from PASCAL, CCES, and ADRIFT that had already been manually reviewed by an analyst so that the automatically created MTC events could be compared to known beaked whale events. Unfortunately the drifts had to be re-run on a newer version of PAMGuard to incorporate updates to the matched template classifier module, so it was not possible to directly compare the MTC events to the manual events on a click-by-click basis. Instead, the times of the MTC events were compared to the times of the manual events, and if there was any overlap in the start and end times then the MTC event was considered to be a beaked whale event. Any events that did not overlap in time with a manually labeled event were marked as "NotBW."

A positive result from this process was that almost every single manually labeled event was covered by a MTC event **(**two events were missed in one drift, including one Cuvier's event with eight clicks and one potential beaked whale event with 7 clicks**)**, so we were hopeful that this approach could be useful for identifying future beaked whale events. The downside of this was that many drifts had two thousand or more total MTC events created, where only around 100 of those were labeled as beaked whale events. Additionally, many of these events contained thousands of clicks, an unrealistic number for a 2 minute beaked whale encounter. The figure below shows the distribution of the number of clicks in the MTC events, colored by species (note the log scale of the x-axis). The MTC events were to be added to PAMGuard databases

using the *PAMmisc::addPgEvent* so that they could be viewed in PAMguard and processed with PAMpal, but the overwhelming number of “NotBW” events proved to be too time consuming for this. As a compromise, instead of adding *all* MTC events to each drift’s database, we instead added all beaked whale MTC events and a random subset of 200 of the “NotBW” events.



Initial BANTER model

The hope for this project is that we could use the now labeled MTC events to train a BANTER model to predict on future MTC events, fully automating the event-creation and classification process (with hopefully minimal manual validation required at the end). As part of this a few event-level measures were added to the MTC events:

- **PeakVar** - the variance of the peak frequency of clicks in the event
- **AngleVar** - the variance of the received angle of clicks
- **Angle10** - the 10% quantile of the received angle
- **BwVar** - the variance of the 3dB bandwidth
- **ICI10** - the 10% quantile of the “All_ici” measure
- **Trough10** - the 10% quantile of the non-zero trough measures

And two additional event level measures were calculated from the average spectra for each event:

- **slope20_30** - the average slope of the average spectrum between the 20kHz and 30kHz points

- **slope30_40** - the average slope of the average spectrum between the 30kHz and 40kHz points

Initially these measures were calculated using all available data, but they appeared to have minimal impact on results, so all 8 above event-level measures were instead calculating using a filtered subset of data (all data were still used to train the models). Filtering was done to hopefully remove some of the undesired “noise” MTC detections, filtering applied was:

filter(data, BW_10dB < 35, duration < 1000, angle > 1.5)

Initial BANTER results

Results from initial model were roughly as shown below. Overall accuracy is fairly poor, even if accuracy for some classes is decent. The bigger problem is that precision is very poor, for all non-Cuvier’s beaked whales we have more incorrect predictions than correct predictions.

```
        OOB estimate of  error rate: 31.6%
Confusion matrix:
      BB BW39V BW43 MS NotBW  ZC class.error
BB    28     1    3  1     1   0   0.1764706
BW39V  0     9    1  0     0   0   0.1000000
BW43   0     1   25  0     2   0   0.1071429
MS     0     0    0 23     0   0   0.0000000
NotBW 41    31   53 28   259  34   0.4192825
ZC     6     2   26  0    24 208   0.2180451
```

These results in of themselves are not necessarily bad if it is possible to easily identify which of these events might need manual review based on the model’s output prediction scores. A summary of the predicted probabilities from the Banter model is shown below - the image is faceted by the predicted class, the color is whether or not the prediction is correct. The top half show the distribution of the highest prediction score, the bottom 6 are the difference between the highest and next-highest class. The main problem here is that not all correct predictions have high scores, and not all incorrect predictions have low scores. If this were this case then we could easily come up with a framework like “Review all events with scores less than XXX” and this could represent a significant time savings over manual review. However, since a large number of the “NotBW” and “ZC” events have low scores (and these are by far the most numerous classes), any score-based review scheme will still end up manually reviewing a vast majority of the data.

We believe that the main reason for these poor results is that BANTER works by aggregating many measures at the event-level, but we know (based on the number of detections in the MTC events and from visual confirmation) that many of the events contain mostly noise with a much smaller proportion of actual beaked whale clicks. If our training data is known to be noisy, then there is only so much that the model can learn.

One approach we tried is using only angles higher than 90 degrees (so detections at or below the array) for model training. From visual inspection, many of the events with thousands of clicks contain a large number of dolphin detections above the array. Most beaked whales (with the exception of Baird's) should occur below the array, so filtering by angle should hopefully reduce the amount of noise included in training. Results remained largely the same, with some classes improving and others getting worse. The model was still not good enough to save time over manual validation.

CV model - Selective Net background

Another possible approach to removing non-beaked whale clicks from the MTC events is using the computer vision based model developed by Taiki at the Caltech CV4Ecology workshop. This model was initially trained on only the manually validated data, so directly applying it to this project is not feasible since the CV model has not seen the type and variety of non-beaked whale detections that are present in the MTC data. However, the last addition to the CV model was something called "SelectiveNet", which allows the model to decide whether or not to make a prediction on a given input image. The way this works is that the SelectiveNet model additionally outputs a "selection" score from 0 to 1 for each detection, where 0 means "Make no

prediction" (and thus no penalty for being wrong), and 1 means "Make a prediction." The model is trained on Wigner transform images, which can be quite noisy. So even for known beaked whale detections, the Wigner image may not be suitable for a human analyst to use for classification purposes. Since these images were included in the training data, the CV model will by default attempt to learn their features. The SelectiveNet feature allows the model to instead choose to not make a prediction for noisy images that it frequently gets wrong. Visual analysis of the SelectiveNet based models seemed to validate this - clicks given low selection scores seemed to be the types of noisy Wigner images that a manual analyst would not use for classification. We hoped that if we used this model to predict on our MTC events that the non-beaked whale detections present would generally get lower selections scores, and the good beaked whale clicks that we are most interested in would get high selection scores.

CV model - Logistics / processing requirements

A main challenge of using the computer vision model for this project is that it requires a significant amount of data processing on top of an already processing-heavy workflow. Individual Wigner transform images must be made and stored for each click detection (hundreds of thousands in the training data). Then these Wigner images must be run through the CV model, which is a Python based program that was not designed to easily accommodate new data or be user friendly since it was a one-off project. The Python model additionally runs much faster on a computer with a GPU and Cuda installed to speed up the prediction process. These hurdles result in many hours of data transfer and processing for each drift, but we were hopeful that the end result would provide enough performance gain that it would balance out in saving manual review time.

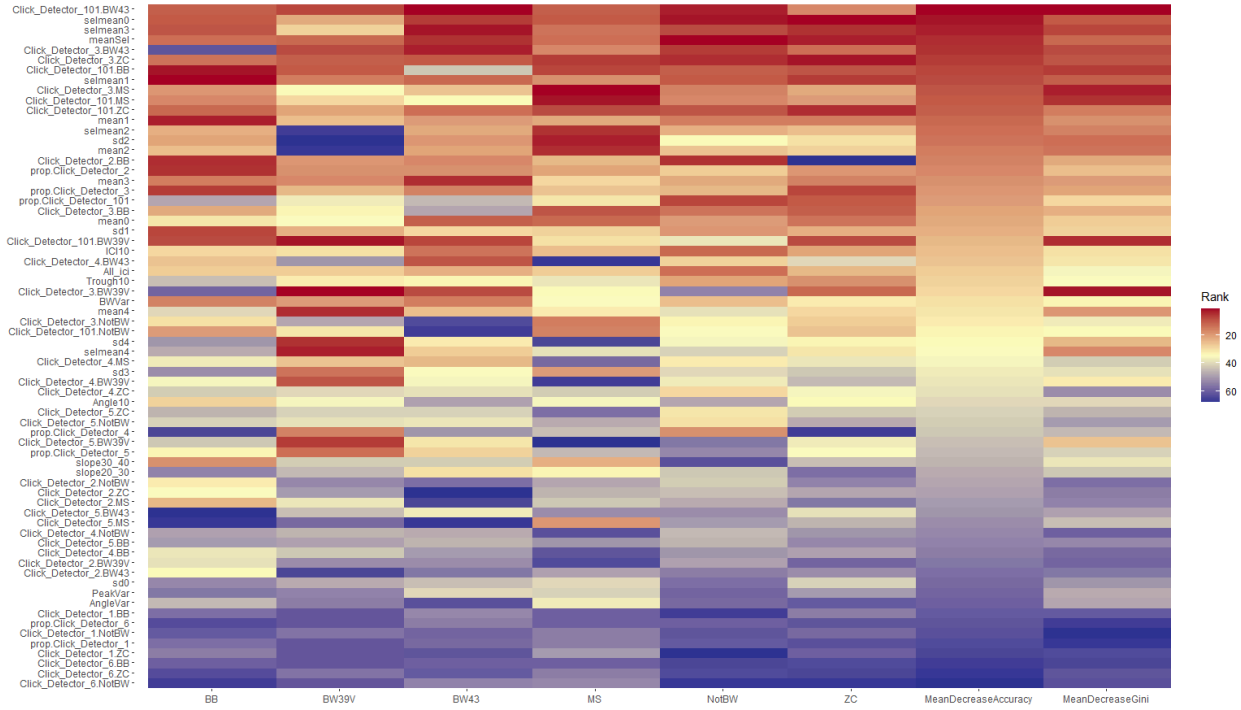
CV model - Integration with BANTER

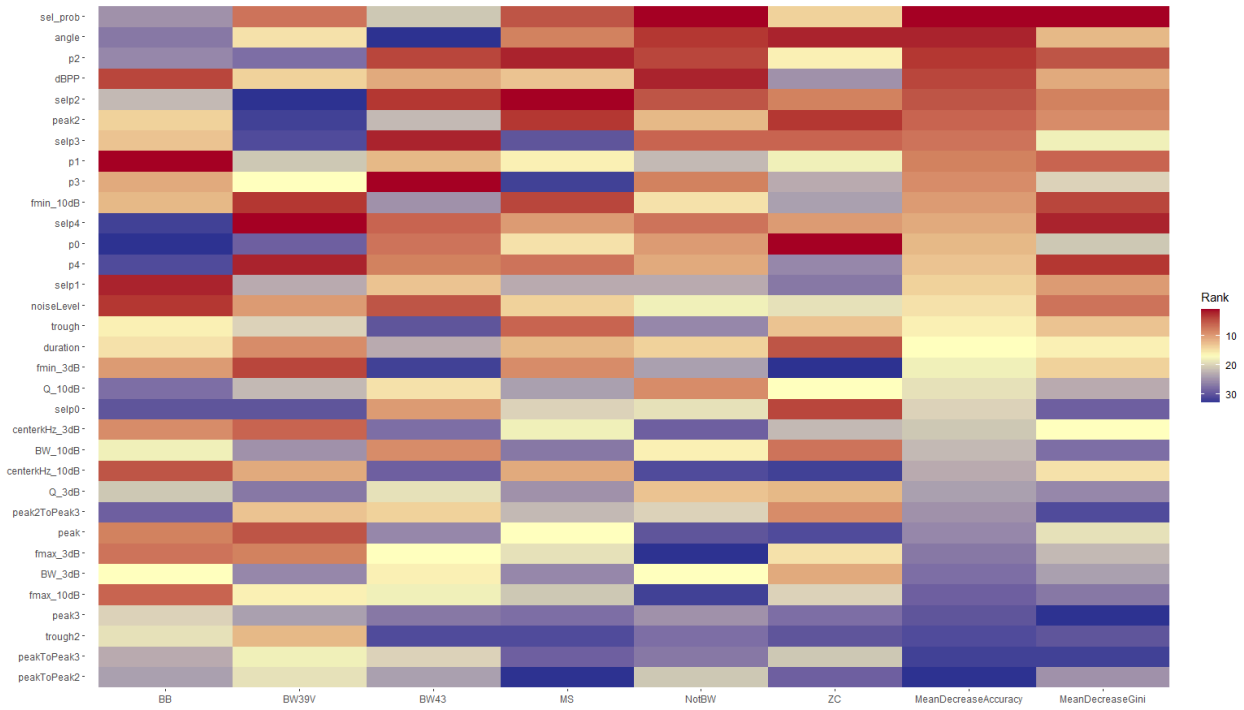
The computer vision model outputs 6 main results for each detection. p0 to p4 are the predicted probability of classes the model was trained on, corresponding to ZC, BB, MS, BW43, and BW39V. Sel_prob is the selection score from the SelectiveNet model. These outputs were used to add both detection- and event-level metrics to the BANTER training data. At the detection-level, the 6 basic outputs were all added. Additionally values selp0 to selp4 were added, these were calculated as sel_prob * (p0, p1, … , p4). 16 event-level metrics were also added:

- mean0, …, mean4 - the mean of p0, …, p4
- sd0, …, sd4 - the standard deviation of p0, .., p4
- selmean0, …, selmean4 - the mean of selp0, …, selp4
- meanSel - the mean of sel_prob

Results for the model including these new measures are shown below. Error is down to 14% overall error, and precision is greatly improved for all classes other than BB. Additionally looking at the importance plot for the final event-level and detector models confirms that many of the new measures are among the most powerful predictors.

```
        OOB estimate of  error rate: 13.94%
Confusion matrix:
       BB BW39V BW43 MS NotBW  ZC class.error
BB     31     1    0  0     3   1  0.13888889
BW39V   0     8    0  0     2   0  0.20000000
BW43    1     0   35  0     1   1  0.07894737
MS      0     0    1 22     0   0  0.04347826
NotBW  31     1   11  0   377  15  0.13333333
ZC      5     0    4  0    31 200  0.16666667
```





However, looking at the prediction scores shows that this model is likely still not good enough to reduce manual workloads. While the plot below shows that there are many improvements over the original model, it also shows that there are many ZC and NotBW events with low scores.

A hypothetical manual review score threshold looking at these data might be to manually review NotBW and ZC events that have a diff12 score under 0.25 (this appears to capture most errors), and all diff12 scores under 0.5 for all other species. This would allow few unintentional errors through while attempting to minimize the amount of review necessary, but even under this plan

we would be reviewing 300 events out of the total of almost 800. This is a significant proportion of the data that would still need to be manually reviewed, especially when considering the extra processing time required to get these predictions AND the fact that this training data contains only a subset of the actual total number of NotBW events. Additionally, it should be noted that these scores are likely to be very optimistic since they are the “out of bag” scores from the data used to train the model. Actual performance on novel data is likely to be notably worse, with more prediction errors and lower prediction scores, along with hundreds more “NotBW” events being present in new drifts.

CV model - angle/selection cutoffs

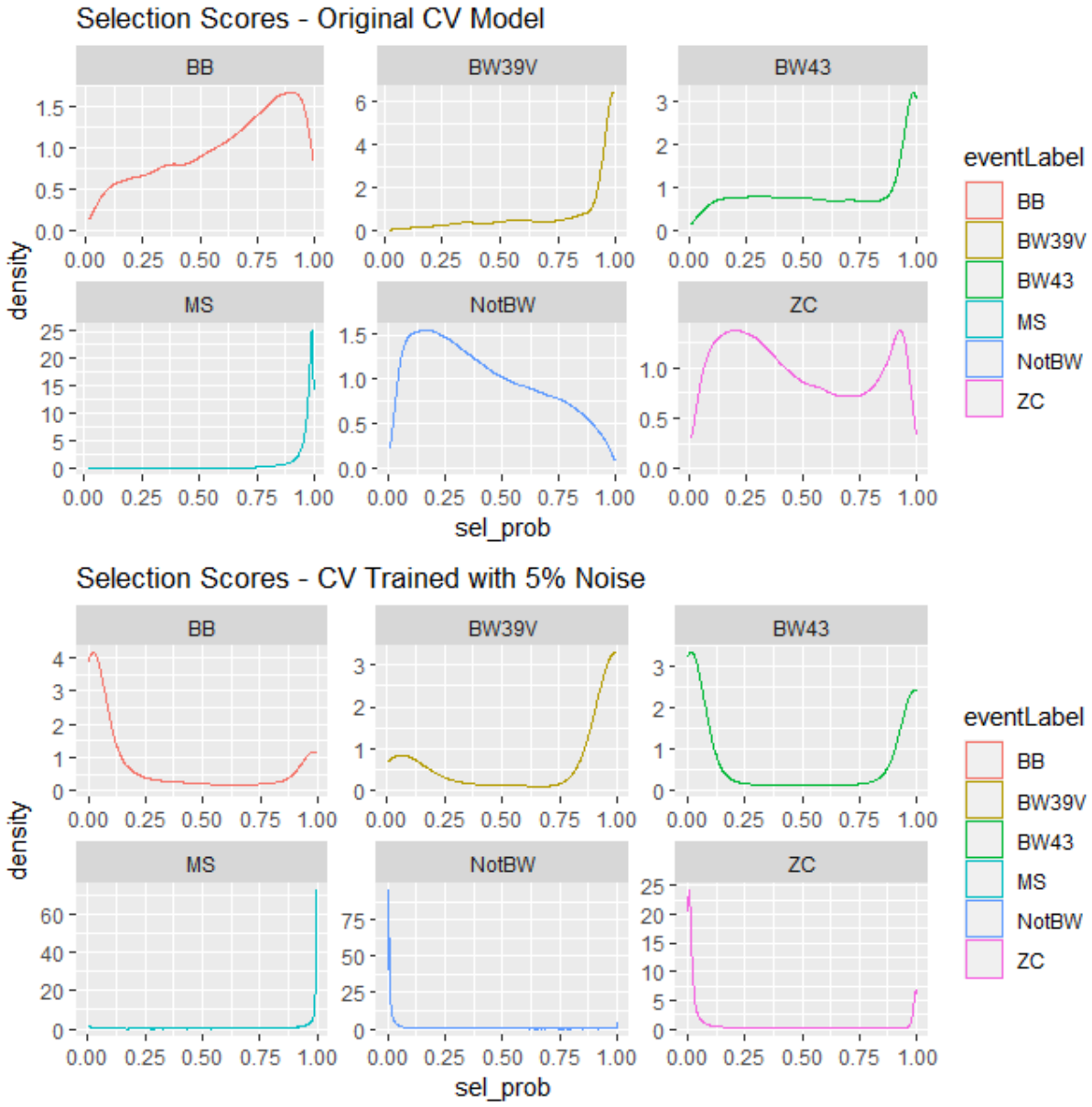
We tried two other variations of the model that we thought would have improved results over simply adding the CV model based detection and event measures to the baseline BANTER model. First we thought that filtering out all detections with a selection score under a certain threshold would improve results by allowing the model to better learn the characteristics of actual beaked whale sounds rather than noise. Cutoffs from 0.3 to 0.5 were tried, and in all cases model performance was worse than a no cutoff model. We tried a similar idea of removing all detections with received angles less than 90 degrees, since these should mostly be dolphin or noise detections (with the exception of some Baird’s detections). The effect of this was mostly that some “NotBW” events get removed entirely (all < 90 degrees), performance was generally similar across the board except that precision for Baird’s events was notably worse. One advantage of this model is that training time is significantly faster since approximately ¾ of the detections were below 90 degrees.

CV model - noisy retraining

We were surprised that the selection score cutoff model performed significantly worse than the no cutoff model, and hypothesized that maybe the selection model just needed to better learn the specific types of “noise” present in our data that we wished to cut out. The plan for this was to add a number of detections to the original model’s training data that were examples of the noise we wanted to get rid of. The species labels of these new noisy detections would be randomized from the original set of 5 species the model was trained on, and kept in equal proportion to the original species distribution. The idea is that since the labels are randomized, the model should not be able to decide on a particular species for the noise category. The easiest path to reducing the loss function in this case should be to learn the pattern of the noise in the SelectiveNet model, and assign them all low selection scores. We could then use these low selection scores to root out the noise detections in our MTC data.

One tricky part of this approach is figuring out the actual detections that should comprise the added noise data. We took random samples from events that had a large number of detections (> 1000), and only sampled from detections that were less than 90 degrees. We thought that these would be very likely to be non-beaked whale detections included by the MTC scores that we did not want in the training data.

We started by adding a number of noisy detections equal to 3% of the original number of detections, then also trained models with 5% and 8% noise data. Below is a graphic showing the distribution of selection scores in the original CV model (top) and the retrained model (bottom).



It is clear from the image that the new model produces significantly different selection scores, and seems to be doing what we want. The vast majority of “NotBW” scores being near zero is a good sign, and we know that many of the other beaked whale events contain a large number of non-beaked whale detections, so those distributions having bimodal distributions around 0 and 1 make sense. However, results using predictions from any of the new models using 3, 5 or 8% noise data were worse than the original model. Attempts were made for all variations using both selection score cutoffs and angle cutoffs, and in all cases results were worse than the original model with no cutoffs.

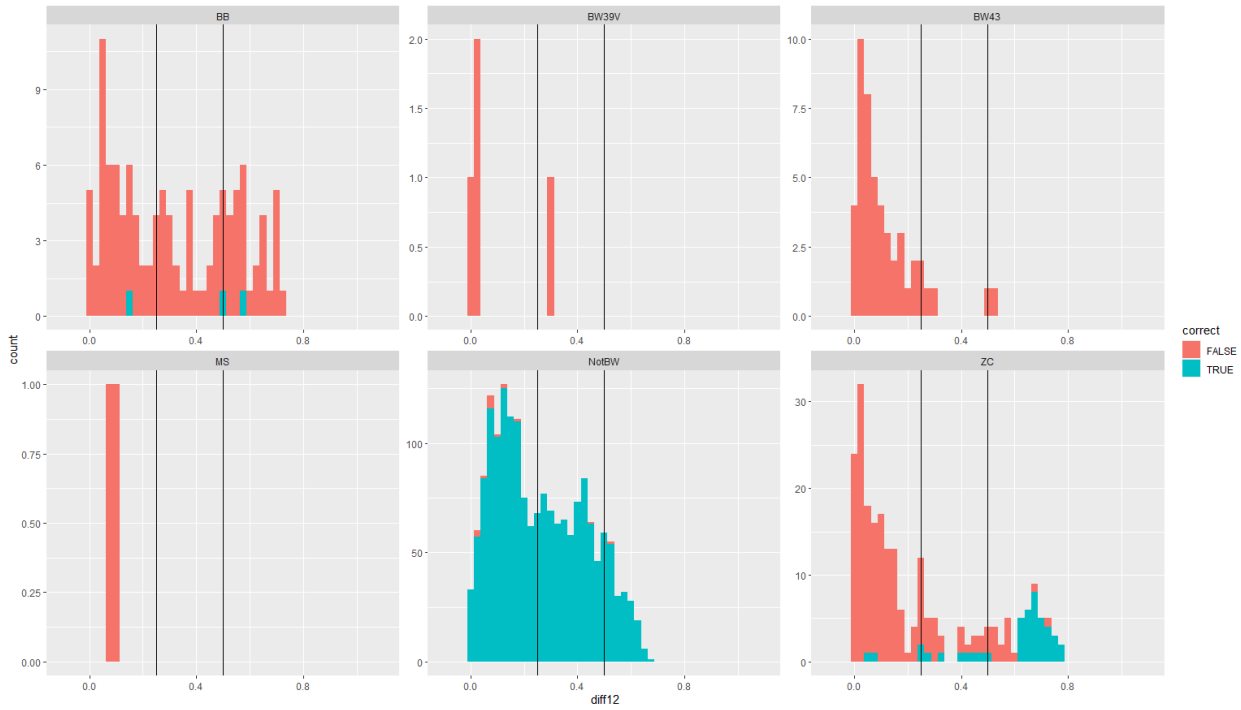
CV model - new data testing

At this point in the project, the time invested has not improved results enough to warrant continued investigation. The purpose of the project was to reduce the amount of manual review required for the remainder of the ADRIFT deployments, but as time goes on it is becoming clear that it is most efficient at this point to just manually review the drifts. As one last test before moving on, we use the current best model to predict on a new drift (ADRIFT_024) that Anne has manually validated as a way to confirm whether or not the model is potentially useful as it exists.

The total time taken (roughly) to transfer, process, and predict on the new data was more than two full workdays (sometimes processing / transferring occurred overnight). The total number of detections used in the training data for this model was around 350k (10 drifts), but the total number of detections processed for ADRIFT_024 was nearly 900k. This giant increase is because the training data only included a maximum of 50 “NotBW” events per training drift, but when predicting on new data we must obviously include all events, which in this case was over 2000. “NotBW” events are also more likely than others to contain a large number of detections.

Results for the best model with no angle or selection cutoffs are shown below. The precision for classes other than NotBW and MS is terrible, and the picture is worse when looking at the distribution of prediction scores. Vertical black lines are drawn at the hypothetical 0.25 and 0.5 review score thresholds from earlier, which reveals two major problems. First is that there are far too many events below the review thresholds (1200 out of 2200 events using the same scores as previous). Second is that there are many incorrect predictions (especially BB) with high scores, so it is very difficult see any review-score scheme being able to capture this (outside of deciding to review *all* beaked whale predictions).

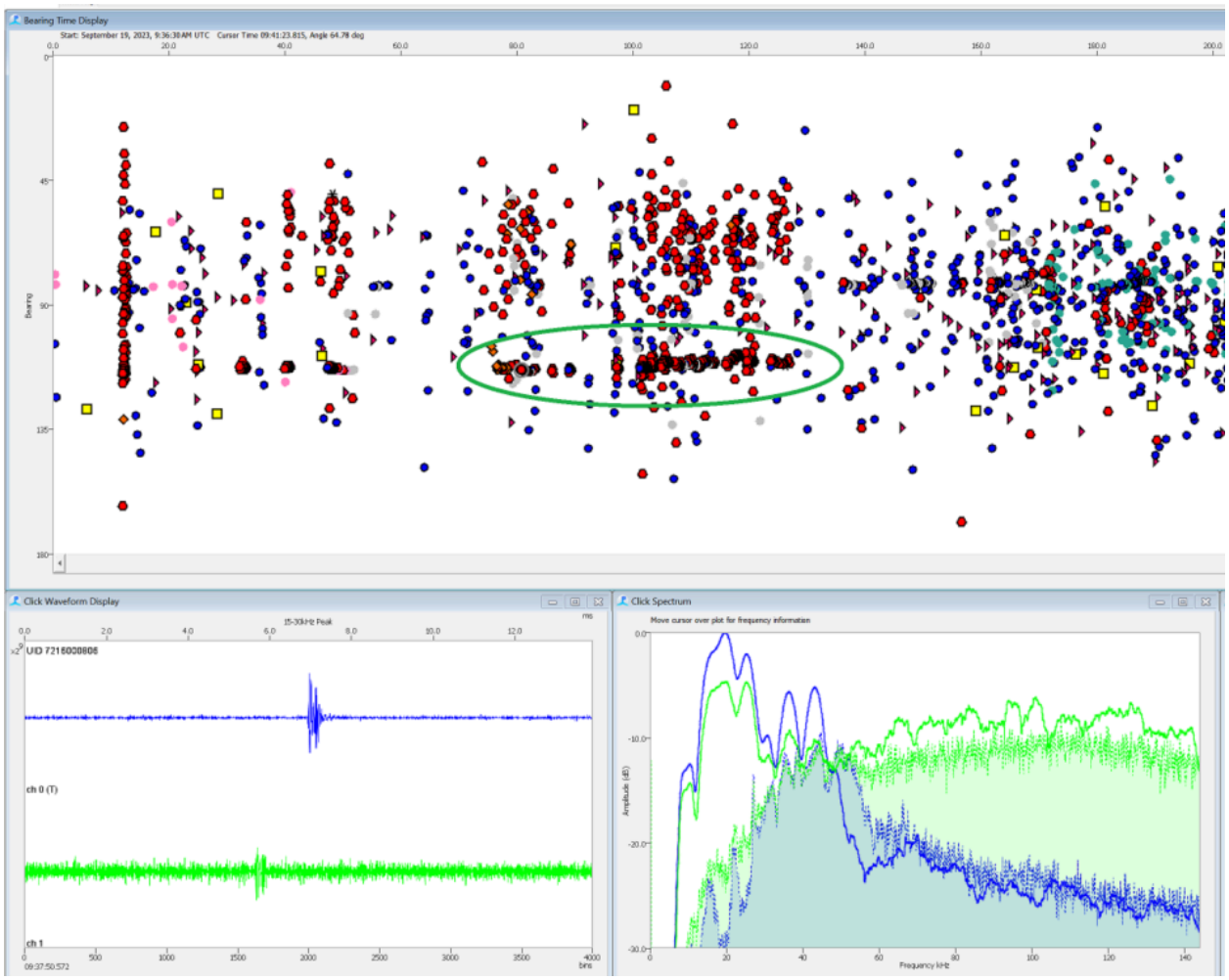
	Banter					
Manual	BB	BW39V	BW43	MS	NotBW	ZC
BB	3	0	0	0	10	2
NotBW	107	4	47	2	1839	186
ZC	1	0	1	0	6	44



At this point the poor performance on this novel drift made it clear that this project would not result in time savings for the rest of the ADRIFT deployments.

Known Issues

We believe that the main culprit for poor performance is with the actual MTC event definition and click detection process. The issue with detecting too many non-beaked whale clicks being detected could potentially be alleviated with the techniques we attempted here, but the bigger problem is that we found many instances (especially with BB events) where it was not detecting very many of the actual beaked whale clicks. So no matter how good the logic is for removing the non-BW detections, there are so few BW detections left in these events that it does not matter. The effect of this for current iterations of the model is that many of our events labeled “BB” are actually entirely noise with the exception of a small handful of BB detections, so the model is forced to learn that “Noise == BB”, and thus many misclassifications of NotBW as BB. In the image below, the click train circled in green should be the Baird’s detections that we hoped the MTC event would capture. In this part of the image, the MTC clicks are colored gray, so we would hope that most of the clicks in that click train would be gray, but we see that very few actually are.



Future Work

Current ideas to improve performance all rely on re-running the MTC module, which would mean new binaries and then re-running all the processing required to deal with those. Since the main issue seems to be that the MTC is not able to detect the majority of many BB calls, we would like to try and find a new BB template that is able to better capture these clicks. If the templates are performing better, this may also allow us to raise the required template score threshold so that we can eliminate more non-BW detections.

We also have a few potential avenues for reducing the number of detections that need to be processed for this workflow. The first is using an angle threshold at the initial MTC event creation process, and only working with detections that are greater than 90 degrees. The potential downside of this is that we may lose some Baird's events since they are known to make clicks higher than the array, but Baird's are already problematic so this is a tradeoff we may be willing to accept. From looking at our training data, this angle cutoff drastically reduces

the number of detections we need to process, so could represent a massive savings in processing time.

Second, it may be worthwhile looking into if there is a pattern to which of the MTC templates is most often triggering on the non-BW detections. If just one or two of the templates is most often the culprit for allowing in a majority of the noise, then it is possible that either by updating the template or increasing the threshold that we may be able to reduce some of the non-BW detections.

Finally, there are improvements that could be made to streamline the processing workflow. As it stands, it is a bunch of somewhat disconnected pieces developed for previous one-off projects. The main inefficiency and processing time cost is that the binary files need to be opened and read at least four separate times - once for initial MTC event definition(all binaries), again for inserting events into databases, again for PAMpal processing, and a final time for creating the Wigner images. The last two could likely be combined (create Wigners during PAMpal processing by adding a function), and event insertion could likely be reworked to use the data already loaded from the first MTC step.

Recreating This Analysis

Here is a rough outline of the data needs and steps required if someone wanted to recreate this project on a different dataset:

1. The starting point needs to be a manually labeled set of beaked whale events. The exact size required for success is uncertain, but if trying to use this model in a locale other than the California coast then the computer vision model will need to be re-trained. In an ideal scenario, with sufficient training data available, a validation set would be held out comprised entirely of drifts or tracks that are not part of the training dataset. While random forest models do not technically require validation data, the purpose of this held out data would be to validate any manual review thresholds for future unseen datasets.
2. A set of beaked whale templates must be identified, and PAMguard run using these templates on that same data. Ideally these would have occurred in the same version of PG as the original manually labeled events so that UIDs could be directly matched between the two, second best would be that UIDs are changed in the binaries but no other detection related parameters are changed so that detections could at least be paired by times. At worst, both these things have changed so that pairing of MTC events to manual labels can only be done by overlapping time (as was the case with this project).
3. All binaries must be loaded, and thresholds for candidate BW clicks/events need to be decided. Then these candidate MTC events are inserted into a copy of the Pamguard database ***that has no existing events or event detections***
4. These databases are processed with PAMpal using *mode='db'*
5. The *AcousticStudy* is then used to create the Wigner images for processing, and a CSV of those Wigner image filenames and statistics is saved

6. If necessary (due to changing species or locale), a new computer vision model is trained using these Wigner images. Then predictions are made using the computer vision model on the Wigner images
7. A BANTER training dataset is created using *export_banter* , and additional computer vision model-based detection- and event-level parameters are added to the BANTER training data.
8. The BANTER model is trained, and performance is reviewed by predicting on the validation set (or if not possible, using the out-of-bag prediction scores from the model).

Code Links

Code for dealing with banter model data prep, training, CV model incorporation
https://github.com/TaikiSan21/PAMscapes/blob/main/devel/selPredBanterExample.R
https://github.com/TaikiSan21/PAMscapes/blob/main/devel/selPredBanterFunctions.R
Original functions for creating wigner images for CV model from AcousticStudy
https://github.com/CV4EcologySchool/beaker-wigner-class/blob/main/R/CV4EProcessingFunctions.R
Code for creating new AcousticStudy and Wigner images using above functions, as well as Steps for creating the noise-added datasets
https://github.com/TaikiSan21/PAMpal/blob/main/devel/wiggyBeakersDataProcessing.R