Batch Processing Tutorial

Douglas Gillespie June 2024

This tutorial will work with any recent version of PAMGuard (V2.02.10 or above) though we're assuming you're using it with V2.02.11c released for this workshop.

Before starting, you must install the batch processing plugin, following the instructions in PAMGuard Tutorial Installation.docx.

Contents

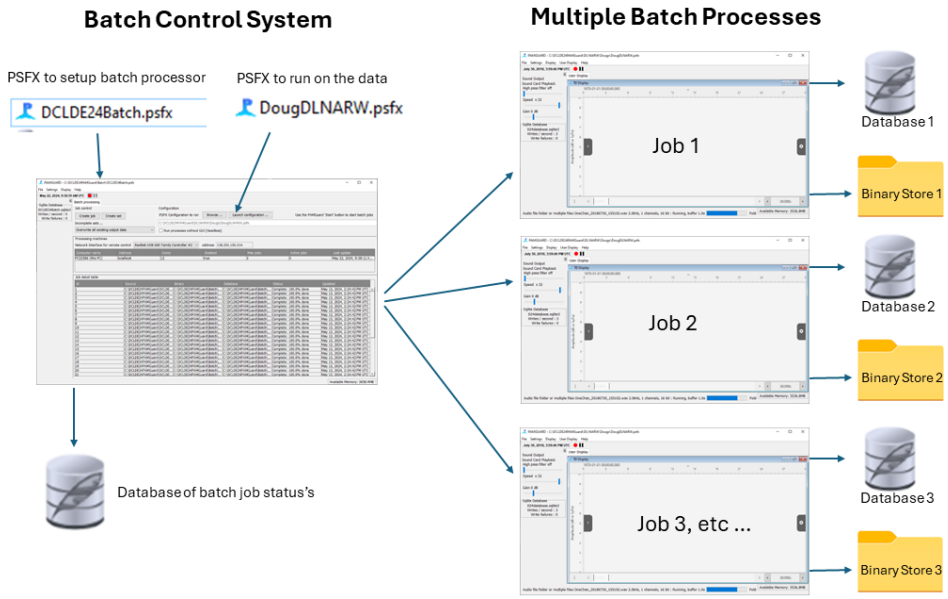
1 Introduction

Particularly with the availability of reasonably prices autonomous data recorders, we often want to process multiple data sets in exactly the same way. Similarly, with all the exciting new Deep Learning detectors / classifiers that are becoming available, you may want to reprocess data from multiple old cruises.

PAMGuard users will know that setting up the same configurations (PAMGuard configurations are held in psfx files) on multiple datasets is tiresome. For each dataset, you need to copy the psfx, then change the input folder for your sound files, the output folder for the binary data and the output database. If you get this wrong, then you might overwrite some data. Then, when you decide that those weren't exactly the detector settings you wanted, you have to do it all again for all your data sets.

The PAMGuard batch processing module addressed this problem by running the same configuration on as many data sets as you want. You set up a series of jobs, and it will work through them, using the same configuration, until they are complete. Generally, it will run

multiple jobs concurrently on a single machine. (In the future I hope to get it running jobs on multiple machines on a local area network.)



1.1 Setting up

The batch processor runs as a PAMGuard module. This will be in a SEPARATE instance of PAMGuard to the PAMGuard's doing the actual processing. It's all shown in Figure 1.

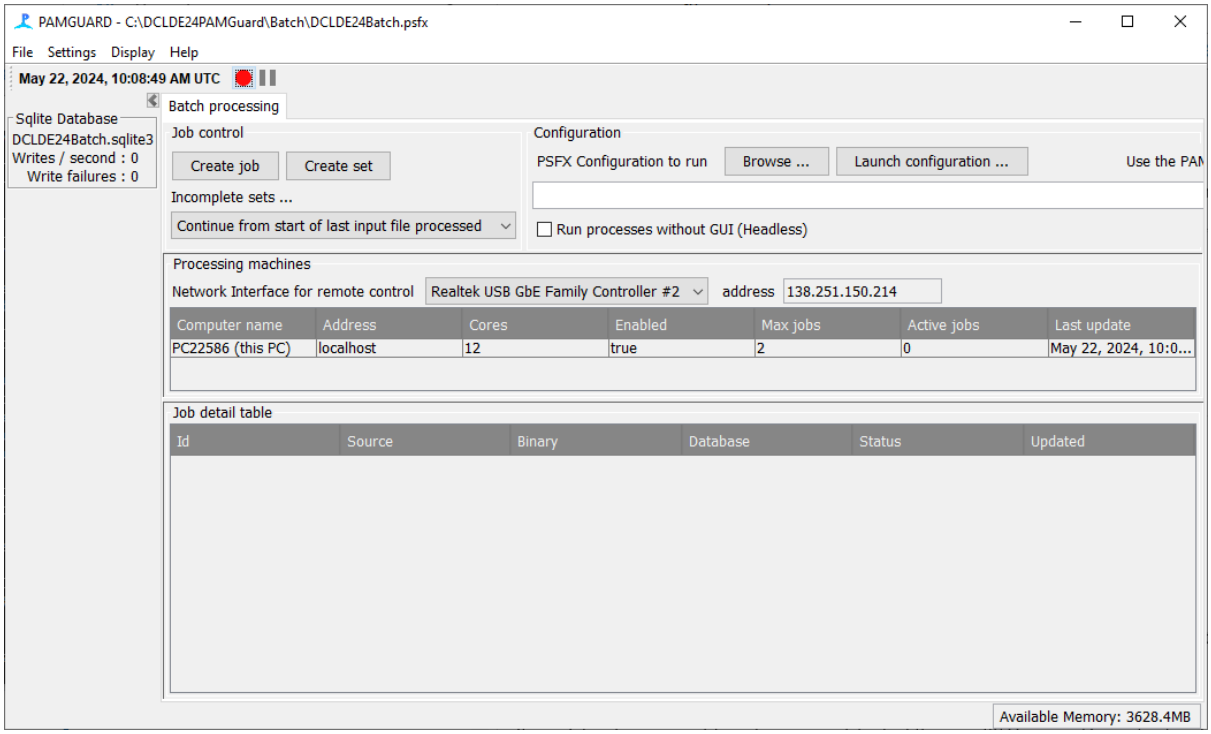
The Batch psfx will contain only two modules: The batch processor module and a database module, which will store the progress of the batch jobs. The batch psfx will also 'know' about another psfx: this is the one with the configuration that you want to run on all your datasets. You'll already have developed and tested this on sections of your data – for this exercise, we're going to use the exact same one that we developed for the Deep Learning tutorial earlier, so below we'll concentrate on getting the batch control module setup.

1.2 Create your batch psfx

First, start a new PAMGuard configuration. To do this, launch PAMGuard, and when it asks for a configuration, select 'Browse/Create New ...' and create a new psfx file somewhere on your system.

From the 'File / Add Modules / Utilities' menu add a database module and then add a batch processing module. Then go back to the file menu, down to 'Database' and 'Database Selection' and 'Browse / Create' to make a new sqlite database on your computer (I'd put this in the same folder as the psfx you just created).

You should now see an empty batch processor display like this:



If you can't add the batch processing module, it's likely you've not installed the plugin correctly. Take another look at the installation instructions or ask for help.

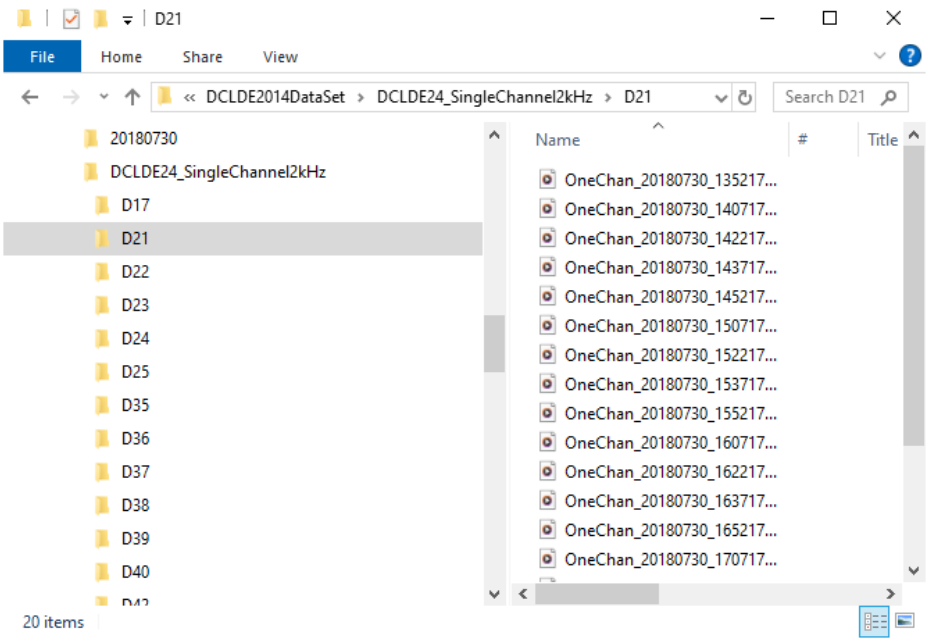
1.3 Select the psfx you plan to run on your data

In the 'Configuration' section of the batch processing control panel, press 'Browse' and select the psfx file that you were using earlier for the Deep Learning exercise. Remember this is a different psfx file to the one you created for the batch processing module. Once you've selected it, you can press the 'Launch Configuration' to check what you're about to run (this will launch another instance of PAMGuard) and you can modify the configuration now if you want to (there is no need to do this for this exercise). Remember to either close that configuration, or at least save it before starting the back processes.

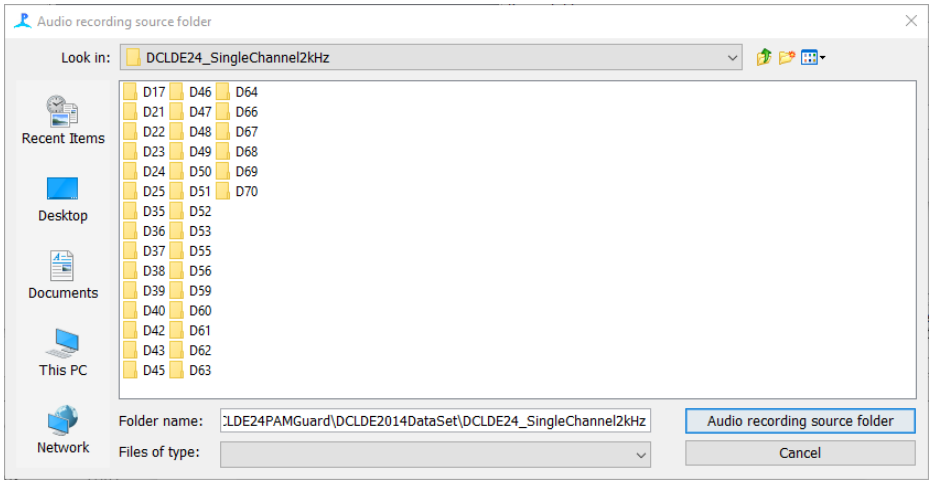
This psfx will probably still be set up to take the one hour of audio data you used for the deep learning exercise and will be outputting data to that same database and binary store. That's fine. The batch processor will change those settings for each job and nothing will be overwritten. If you're worried though, you can change them to a new blank database and blank binary store folder.

1.4 Set up batch jobs.

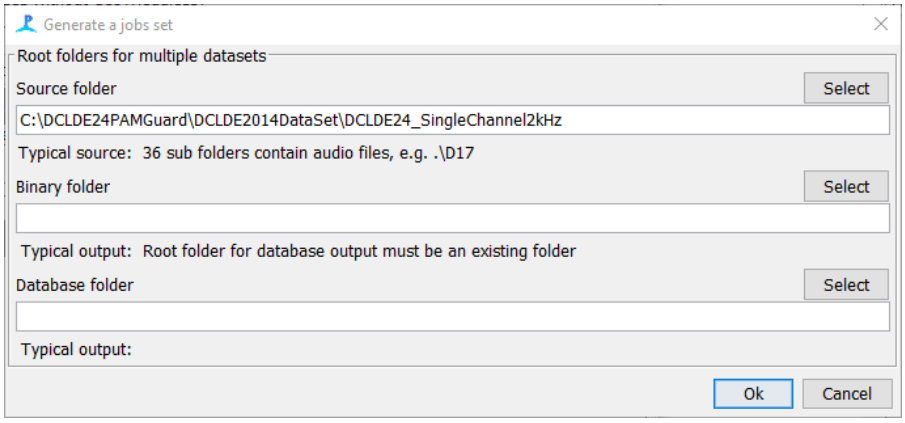
For this exercise, we've taken 36 folders of data which have been modified slightly from the sonobuoy DCLDE 2024 dataset. Data from each sonobuoy are stored in a separate sub folder. Use Windows Explorer to take a look at the raw data and see how it's organised. Here's how it looks on my own laptop



To set up batch jobs, you can either press the ‘Create job’ button (try it now) which will allow you to set up a single job, specifying where the raw data (i.e. the wav files) for that job are, where you want the binary data to go and which output database. However, the whole purpose of the batch processor is to avoid doing the same thing 36 times, so ‘Cancel’ that and press the ‘Create Set’ button. In the top right, press the ‘Select’ button for the ’Source Folder’ and navigate to the folder that contains the 36 folders of audio data. On my laptop this is ‘C:\DCLDE24PAMGuard\DCLDE2014DataSet\DCLDE24_SingleChannel2kHz’



When you select this folder, the batch module will analyse the directly structure and establish that there are 36 folders containing audio folders. If you don’t get this, then you’ve selected the wrong root folder so try again.



Now select folders for the binary output and for the databases that will get created. These will probably default to the folder containing your batch processing psfx, which is fine, but feel free to put them somewhere else.

> Hopefully, if you're doing this for real, you'll have thought about how much data you're about to generate and will select folder on a drive with enough space.

Once all three fields in the 'Generate a job set' dialog are complete, press Ok and the 36 jobs will be added to the table on the main display.

1.5 When to set the jobs up individually

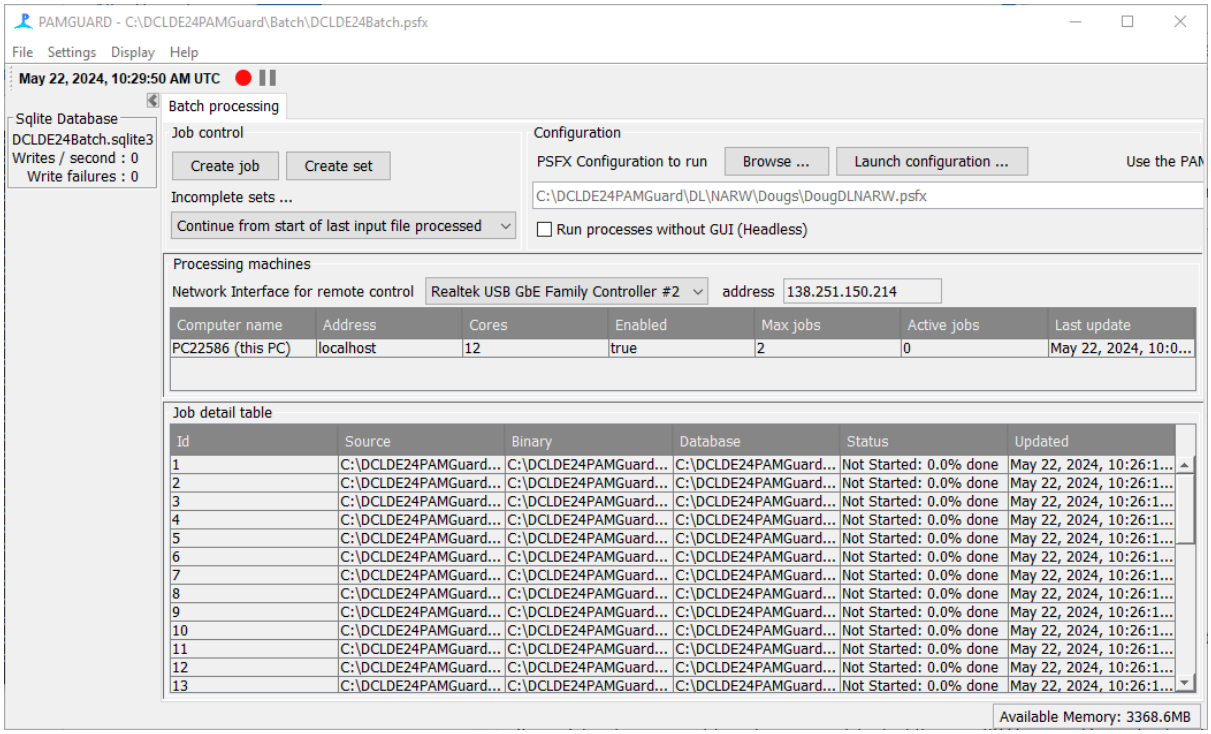
Sometimes, it's not possible to have all of the data you want to process on a single drive, in a single folder structure. In this case you may have to set up jobs individually. For instance, I recently reprocessed 12 sets of data spread across 12 different 10 Terabyte external hard drives on a single desktop. Since you don't know which letter each drive will be assigned until you plug it in, I proceeded as follows:

- First, I checked for Windows updates. There were some, so I rebooted the computer, then disabled further updates for a two-week period which is the maximum our university admin will allow.
- I then plugged in six of the hard drives and set up six jobs, one of the data on each drive.
- I set it to run three jobs at a time and went away for the weekend.
- On Monday, the first three jobs had completed, so I unplugged those drives (careful to get the right ones!) and plugged in three more. I set up the jobs for those three while it was still processing the rest of the first six and got on with other things.
- Every day or so, when I noticed jobs had completed, I remove the relevant drive, put a new one on and set up a job for it.

I got through 80 Terrabytes of data in a week, all with the same psfx and a minimum of interaction with the PC. (Of course it would have been easier if all my data were on a server, but that's another story).

2 Run the jobs

The only instance of PAMGuard open on your computer now should be the one with the batch processor module and the database and it should look something like this:



By default, it's going to run two jobs at a time. Leave it at that for now.

> All modern computers have processors with multiple 'Cores' or 'logical processors'. These are effectively separate processors, which can run in parallel, but share the same memory space. The number of cores on your machine should be showing on the batch processor display. My laptop has 12, but my (faster) desktop has 20. PAMGuard already spreads it's workload across multiple cores (more-or-less using one per detector), but generally doesn't use all of them. I generally find that I can run three of four jobs at a time before they start slowing down, but this can vary a lot on different machines and with things like the complexity of your PAMGuard configuration, data storage (is it a local disk, a network drive) etc.

Press the PAMGuard red start button and watch ....

You should see it pretty immediately start two PAMGuard jobs. These will run through the files in the appropriate folders and automatically create the required output folders and database files. Use Windows explorer to check this is happening as you'd expect it to. As each job completes, that instance of PAMGuard will close and a new one, for the next job, will launch.

If you can, open the task manager to see how much of the computers CPU you are using.

2.1 Increase the number of concurrent jobs

If you want to, try increasing the number of jobs that can run concurrently. To do this, simply right click on the single row of the computers table and select 'Increase max jobs to 3'. Do the other jobs slow down ? Did the amount of CPU in the task manager increase ?

You can also decrease the number of jobs. If you do this, it won't actually stop any jobs that are currently running, but it won't start new ones until the number running drops below the set maximum.

2.2 Stop

We probably won't have time for your laptop to get through all the data. When you've had enough press the 'stop' button on the PAMGuard batch module. Only then can you either wait for running jobs to complete, or stop them individually.

3 DIY batch control (for experts)

The functionality to make the batch processing work is all built into PAMGuard as command line options. For instance, the command line issued by the batch processor to start the first of the jobs in this example was

"C:\Program Files\Pamguard\Pamguard.exe" "-psf" "C:\DCLDE24PAMGuard\DL\NARW\Dougs\DougDLNARW.psfx" "-wavfilefolder" "C:\DCLDE24PAMGuard\DCLDE2014DataSet\DCLDE24_SingleChannel2kHz\D17" "-binaryfolder" "C:\DCLDE24PAMGuard\Batch\D17binary" "-databasefile" "C:\DCLDE24PAMGuard\Batch\D17database" "-autostart" "-reprocessoption" "CONTINUECURRENTFILE" "-multicast" "230.1.1.1" "12346" "-netSend.id1" "1" "-netSend.id2" "3055"

Most of the content of this command line are reasonably guessable, but there is some documentation at https://github.com/douggillespie/PAMGuard/wiki.

The code below is something I wrote in Matlab to a) test the functionality and b) process the DCLDE24 dataset before I'd completed the plugin module that you've learned about above:

```
root = 'C:\ProjectData\DCLDE2024\';
psfx = 'C:\ProjectData\DCLDE2024\dclde2024rw.psfx';
pgExe = 'C:\Program Files\Pamguard\Pamguard.exe'
subdirs = dir(root);
% parfor (i = 1:numel(subdirs), 2)
for i  = 1:numel(subdirs)
    if (subdirs(i).isdir == false)
        continue
    end
    subPath = fullfile(root, subdirs(i).name);
    wavs = dir([subPath, '\*.wav']);
    if numel(wavs) == 0
        continue
    end
    % now make a new binary folder name and database name based on the path
    binName = fullfile(root,  [subdirs(i).name, 'binary']);
    dbName = fullfile(root,  [subdirs(i).name, 'database.sqlite3']);
    % see https://github.com/douggillespie/PAMGuard/wiki/Command-Line
    commandOpts = sprintf('-psf "%s" -wavfilefolder "%s" -binaryfolder "%s" -
databasefile "%s" -autostart -autoexit', ...
       psfx, subPath, binName, dbName);
    fullCmd = sprintf('"%s" %s', pgExe, commandOpts);
   pgOut{i} = system(fullCmd);
    fprintf('completed PAMGuard run on %s\n', subdirs(i).name)
end
```

In principle, it should be possible to set up lists of jobs to run in the background on a server. This would use the command line options built into the main PAMGuard program, but would not use the PAMGuard batch processing module. I'm not aware of anyone who's actually done this yet, but would love to hear from anyone who wants to try.

4 What's Next ?

There are two main developments we're hoping to work on in the batch processing module in the remainder of 2024. The first development, batch processing of Offline Tasks, is a high priority since I know we all often need to do that to our data. The second, running on multiple machines, is lower priority since it may be used by very few people. We welcome any views on the priority that should be given to both these tasks and any input on how you would like them to operate.

4.1 Offline Tasks

Many of you will be familiar with some of the 'offline tasks' available in PAMGuard viewer mode. These include things like re-running the click classifier which attempts to assign individual clicks to different species classes, or re-running the whistle classifier algorithm on pre-detected whistles. Setting this up, is proving a lot more difficult since the PAMGuard viewer databases already contain a configuration, but the batch controller will need to update parts of that configuration, but not necessarily all of it. Work is ongoing and we welcome examples of data and input as to what your priorities are moving forwards.

4.2 Multiple Machines

It's always been our intention to make the batch processor spread tasks out across multiple computers. There are some quite significant challenges in doing this since both configurations and data will need to be moved between machines. Key is now data are stored: It's only going to work if data are on a shared Network storage system. We value input on how people might like to see this work.