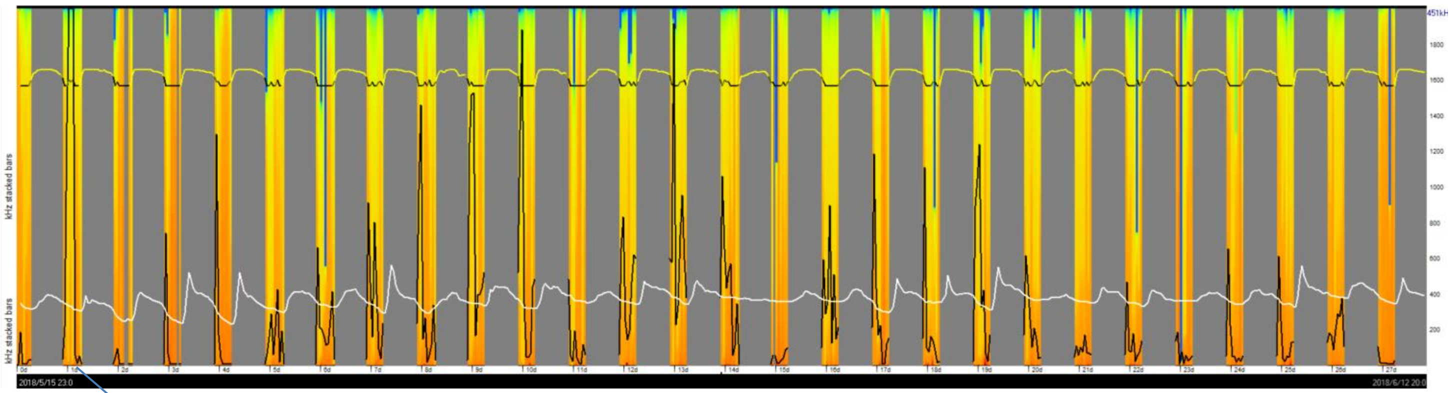
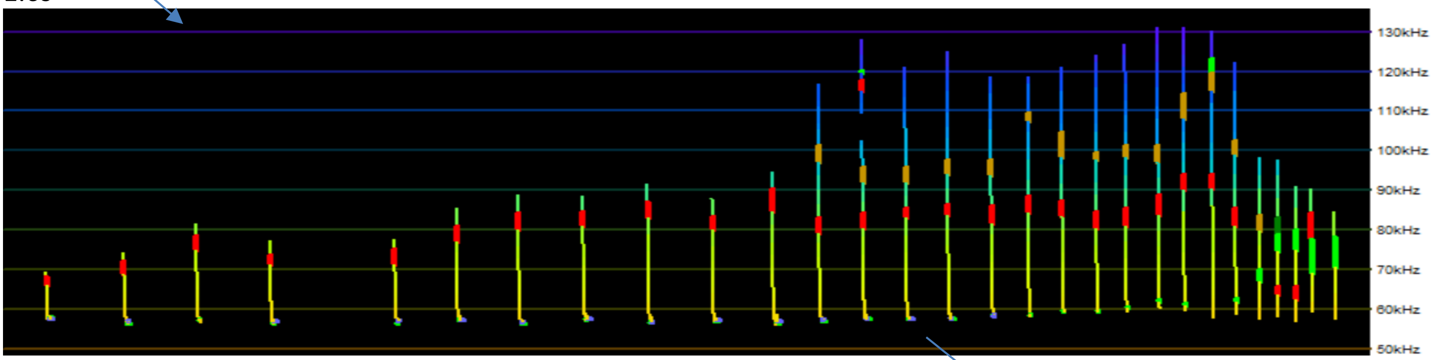
BatBug User Guide

draft July 2024

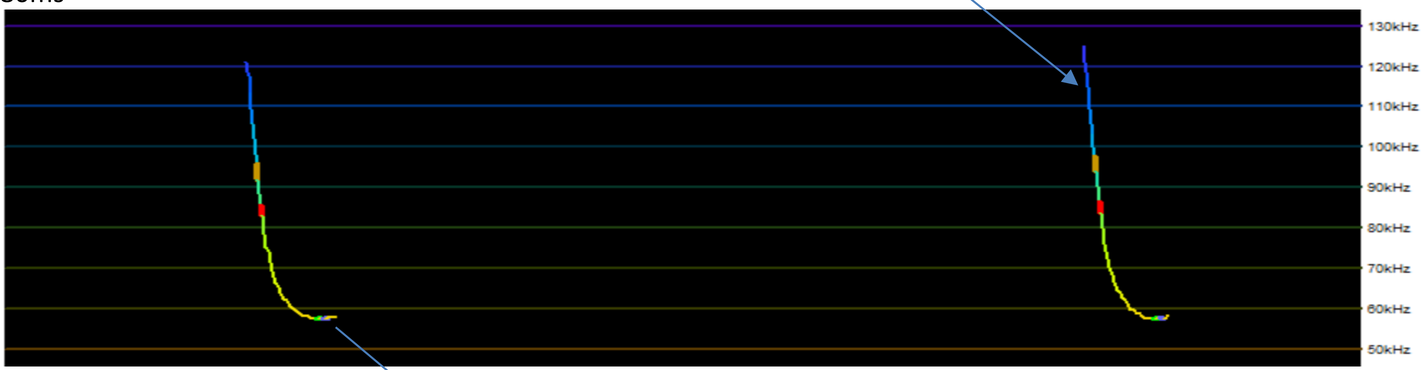
28 days of data: frequency distribution, light level, and temperature



1.6s



80ms



features of 1 call:

time	5uS	CallN	FlightN	Seq	CF_FM	Buzz	Nwave groups	avScore	Reverb	Duration	Fmin	Fmax	Fstart	Fend	Frange	ModalKHZ	MedianKHZ	Fav	FbodyHi	FbodyLo	MinSlopeKHZ	MinSlope	UpBend@	UpSharp	DownBend@	AmpAv	Nmissing
16/05/2018 23:23	6964867	84	111	UFMR	tcQCF	FALSE	45	8	FALSE	52	57	124	124	58	67	58	65	75	90	58	57	77	70	7	94	130	0

45 *8-cycle groups* in call:

time	5uS	avF	SPL	score	Pk1+3	Pk2+4	Pk5+7	Pk6+8	KHZ1+3	KHZ2+4	KHZ5+7	KHZ6+8	tcCF	CallN
16/05/2018 23:23	6964867	125	93	5	85	78	83	93	131	123	123	123	0	84
16/05/2018 23:23	6964881	121	122	7	122	110	105	89	119	123	127	116	0	84
16/05/2018 23:23	6964895	119	117	9	103	103	140	123	119	119	116	119	0	84
16/05/2018 23:23	6964908	118	180	9	150	168	206	198	119	116	116	119	0	84
16/05/2018 23:23	6964923	115	246	8	246	238	220	212	116	116	113	116	0	84
16/05/2018 23:23	6964938	110	172	7	172	168	112	120	113	107	110	110	0	84
16/05/2018 23:23	6964953	106	150	7	117	150	96	106	104	113	101	107	0	84
16/05/2018 23:23	6964969	103	184	3	126	146	182	184	107	99	110	99	0	84

4weeks of bat calls, light levels, temperatures, and call (*Pipistrellus pygmaeus*) details from a single BatBug deployment. The wavelength of every cycle in every detected call is logged.

Chelonia Limited

The Barkhouse, North Cliff, Mousehole, Cornwall, TR19 6PH, UK

Feedback

We're keen to know how you get on. Write to nick.tregenza@chelonia.co.uk.

Specification

The aim of the BatBug is:

To facilitate accurate ***long-term acoustic monitoring of bat populations*.**

To achieve this it aims to combine:

- wide coverage of frequencies – 17 to 230kHz.
- compact data so that 32GB will hold 1 year of data.
- solar power
- rapid automated analysis - this is partly developed and awaits more extensive 'training data'.
- low false positive rates – this is more important than maximum sensitivity as significant FP rates force non-automated analysis.
- weather resistance.

Compact data – the BatBug holds over one year of bat data on one 32GB SD card. To achieve this continuous real time sound analysis is used to select only actual calls, plus any ambient sounds that resemble calls. This 'micro-triggering' is carried out by a 'wavelet transform' algorithm that triggers on short segments of sound that have a low bandwidth. Some sound from before and after any such tonal event is also selected.

Micro-triggering also reduces the complexity of the 'set up' process, as it implicitly finds the optimum threshold in quiet and noisy conditions so there is no need to adjust and detection settings to match some prediction of what will happen. Triggering is on *bandwidth* not frequency, so it works for bats of all frequencies. The bandwidth is measured over a short time window, not over the whole call.

Automated call detection is carried on in post-processing on a PC and is very fast – less than 1 minute per day of data – because of the computation already done in real time. Data from known species is needed to develop this further.

Environmental data: Light and temperature are recorded each minute.

Waterproofing: the instrument needs to be at least 30degrees from facing vertically upwards. A vertical 'view' is possible by pointing a horizontal BatBug at a smooth flat surface sloping at 45 degrees.

Frequency range - 9kHz to 230kHz.

Directionality - The zone of sensitivity is a cone in the long axis of the BatBug and out to about 45degrees from this.

Audio -The BatBug does not make any sound for you (or bats) to hear, but you can replay calls later from a PC.

Batteries/power - The **Solar Powered BatBug** runs indefinitely from a solar panel + night battery unit.

Alternatively the **BatBug** can use up 3 internal stacks of 6 AA cells. Any type of AA cell can be used, with only one type in a stack. On batteries it can for more than 21 * 24hours. 'Night only' can be set using the light sensor to switch it on and off at a chosen light level. No times need to be picked and this greatly extends the number of nights that can be logged, especially in summer.

Quick start

Solar: Put a blank SD card into the BatBug. Connect the BatBug lead to the Solar Unit lead, and it starts running. Various LED flashes occur ending in 4 green flashes when it starts. These can be seen

Internal AA cells:

1. Put batteries in the BatBug and fit the battery pack lid - this is a metal plate with two guide pins that both run down *within* the middle battery compartment. Push it down until the spring clip protrudes through the square hole and moves across to retain it.
2. Put in a blank 32GB micro-SD card. You will see various flashes as each battery pack is tested, then the green LED flashes briefly four times – the 'happy flashes', and the BatBug **starts**. If you get red flashes you need to erase or re-format the card (The 'quick format' option in Windows) on a PC.
3. Fit the lid to the tube, taking care that the retaining cord is entirely inside the tube.
4. Deployment: preferably 30degrees or more from vertical pointing into the space you wish to monitor. See Deployment for more ideas…
5. Getting your data: Press the **'Save File Button'** next to the SD card. Wait for 10 seconds or until and until any LED flashes have stopped. Then remove the card from the BatBug. Read the SD card with **BATBUG.exe by going to the 'Get Data' page of the menu** and reading the file 'BATBUG0.CHE' from the SD card. Automated call extraction follows and one or more '.BB1' files are created.
6. A deployment gives a series of one or more files that have a maximum size of just under 4GB. One file may cover more than a year. If the card, by mistake, contained old data then the BatBug will save the new data with a different series of file names. These raw data files and named BATBUG0.CHE, BATBUG1.CHE etc, and if there has been a restart further series CATBUG0.CHE, DATBUG0.CHE, EATBUG.CHE etc!!! are written. You only have to open BATBUG0.CHE, and the rest follow automatically.
7. The longest .BB1 file will be opened automatically, with a .BB2 file containing the calls found in the .BB1 data.

Batteries: Rechargeable NI-MH battery stack rules:

- Get high quality new batteries e.g. *Eneloop*, and keep the same set of 6 together. fro
- Use an intelligent charger e.g. the *Opus BT-C3100 v 2.2* that allows you to charge/discharge/refresh and perform internal resistance tests whilst viewing the charging progress.
- After more than 10 charge cycles some cells may be impaired so
- test their capacity, internal resistance and voltage. Then group together only similar batteries of the same brand and age.
- Ni-MH batteries need periodic full discharge and recharge.
- In the BatBug the weakest stack is selected and exhausted first, to avoid throwing away partly discharged non-rechargeable batteries.

Raw Data:

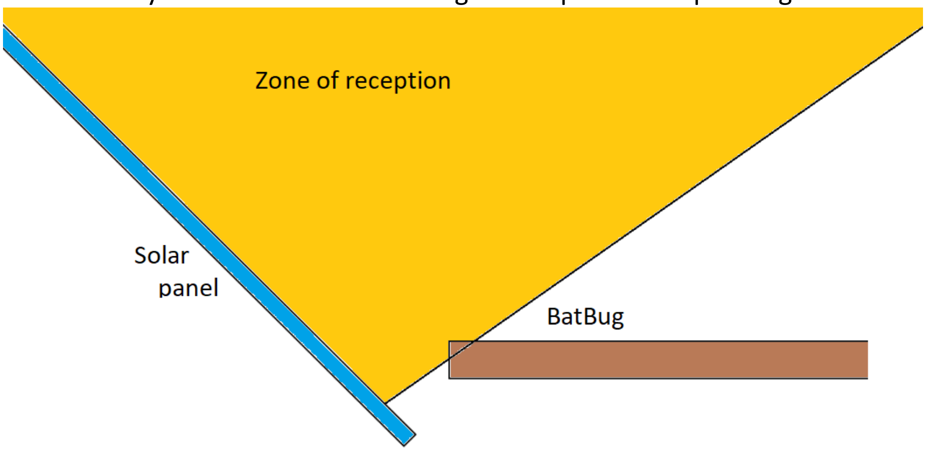
- Frequency range 9kHz to 300kHz. The wavelength (duration) of each cycle in a call is logged. This gives a precise view of the 'wave frequency' of the ends of calls.
  - As a result these frequencies are often (correctly) shown as being higher than the 'traditional' values from .wav file loggers. Those instruments (sometimes called 'time expansion' loggers) get the frequency from the Discrete Fourier Transform which looks a period of time that includes multiple waves in a call, so the peak frequency is, 'blurred' or 'averaged' to some degree.
- Calls are recorded as the wavelength (a period of time) of every wave (cycle) plus interleaved amplitudes i.e. pairs 1+3, 2+4, etc. This format allows the presence of a key harmonic configuration to be more clearly

identified - calls in which there is a dominant second harmonic plus a third harmonic. This call structure is particularly valuable in species identification as Rhinophids use this and some (all?) long-eared bat calls move from a dominant first harmonic (fundamental) to a dominant second harmonic. In both case the harmonic structure can be captured in the BatBug data.

- Selected data is recorded in timed 8-cycle segments, called 'wave groups'. The start time of each wave group is logged to 5microsecond resolution, and the duration of individual cycles is logged in 250nanosecond units (4million of these units in one second). A bandwidth measurement for the wave group is made in real-time by the BatBug and is also stored.
- Temperature and light level are logged every minute.
- Daytime: Call logging can be switched on and off automatically in response to light to prolong the running time.

Housing:

- Robust housing, waterproof at any angle greater than 30degrees.
    - For a directly vertical 'view' the BatBug can be positioned pointing into the solar panel :



- Single point deployment: it can easily be hung from high branches, light fittings, etc.
- buoyancy collar can be fitted to enable it to be deployed in lakes with only the microphone housing showing above the water. This avoids echo-interference.
- static proof (not susceptible to electrostatic damage).

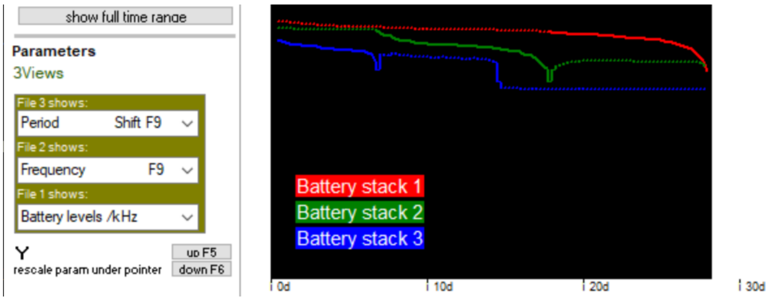
Solar power:

- The small solar panel (max 4W) can power a BatBug indefinitely. A direct view of the sun is required at some point in sunny days (i.e. not in unbroken shade) A 20W panel can power the BatBug in many locations without any direct view of the sun.
- The external sealed lead acid battery in the box can run the BatBug for a few days to carry it through the nights and cloudy periods.
- The charge controller measures the battery temperature and prevents damage due to over-charging or over-discharge.
- Extension leads of any length can be provided.

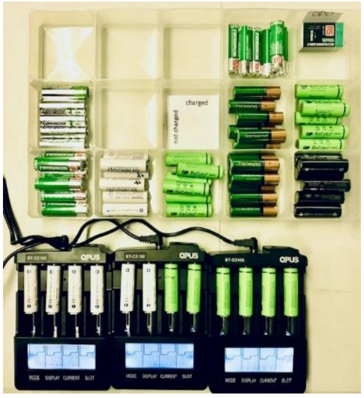
AA Batteries:

- These require a different lid, that does not have the connection for the solar panel.
- 1, 2 or 3 sets of AA cells - all go in **negative end first**.

- Do **not** drop batteries in to a vertical BatBug. Push them into a nearly horizontal BatBug.
- NiMH rechargeable / Alkaline / 1.5v Lithium cells may be used. Each stack must be composed of only one type of cell.
- Low / zero battery wastage: The weakest stack is used up first, so that it can be replaced and/or recharged. This means that unless your deployments are long you only need to replace or recharge completely flat batteries – the better stack is kept in reserve.
- Running time: 18 AA cells: continuously for 28+ days; longer if ON during low light only; indefinite for BatBug Solar.
- The battery history can be viewed as shown here.



- Managing AA rechargeables: we use “Opus BT-C3100 v 2.2” chargers and a plastic tray to keep sets of batteries together. Use different brands in each stack; put them in ‘portrait’ orientation in the plastic tray compartment when waiting to be charged, and in ‘landscape’ orientation when charged.



Running / Stopping

The SD card switches the BatBug on and off. It ‘snoozes’ – a very low power mode – through the day if you set it to be ON only in the dark.

With batteries present:

Insert an SD card. If it is recognised the BatBug starts, checks the batteries and finally the green LEDs flash 4 times, and you can close the lid. If the SD card is not recognised the red LEDs flash. In that case you need to re-format the card (see SD cards on p8) on a PC.

If the batteries run down during logging the BatBug saves the file, switches off and the LEDs go off. Until that happens the LEDs flash briefly every time data is stored. If a file is already on the card, or the BatBug restarts a new file with a modified name is created.

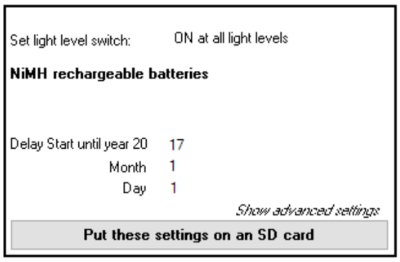
At the end of deployment, **always press the ‘Save File Button’,** situated next to the SD card slot**.**

If there is data to save you will see rapid green flashing. Wait for a long green light to end before removing the SD card. If there is no LED activity within 20 seconds remove the SD card – the BatBug may have already stopped. With no SD card the BatBug is OFF.

You may find you can remove and replace the SD card using your fingers, or you may need needle-nosed pliers:

Settings

It’s usually best to use the default settings, which you do with a new BatBug simply by recording on any blank SD card.



To extend the running time: The main reason for making a change to the settings is to extend the running time if data has been lost in your deployments from battery exhaustion or filling the memory.

Memory use: is managed in sites with very high levels of bat activity by progressively limiting the amount of data that is stored each day. This is done automatically if needed and usually does not happen at all. You can override this via the ‘advanced settings’.

Battery life: Confine logging to times when light levels are low. This doesn’t help in caves where light levels are always low. You will miss day-flying bats, of course, and they are biologically interesting.

To give a BatBug new operational defaults you create a BATSETS.txt file using BatBug.exe. The BatBug will read this when it starts, and these settings will then become the normal settings and will be used subsequently. Then remove the SD card in the normal way and insert a blank SD card.

Software in brief

‘BatBug.exe’ is dedicated **freeware.** It will remain free, including future upgrades, species/guild identification etc. Download from the Chelonia/BatBug website. Features include:

- fast navigation though time; zoom from microseconds to days.
- synchronous frequency and amplitude displays.
- automatic marking of key frequency features of calls.
- instant diel pattern graphs.
- data from up to 6 BatBugs can be viewed in sync in the same display window.
- species identification will be added as we acquire sufficient data. Help with obtaining that data is required, and if you can help then your help will be acknowledged in the software. What we require is named-species data – either from a known single species roost (or mixed, but easily distinguished, species) or where there is some other basis for a confident identification.
- Third party access: the data format is open source, and the full data can be exported so that third parties can develop species identification methods, and write back their results into the BatBug files, so that the viewing, filtering and exporting functions are available.

Useful information on the item under the mouse pointer is shown in the blue bar at the top of the screen.

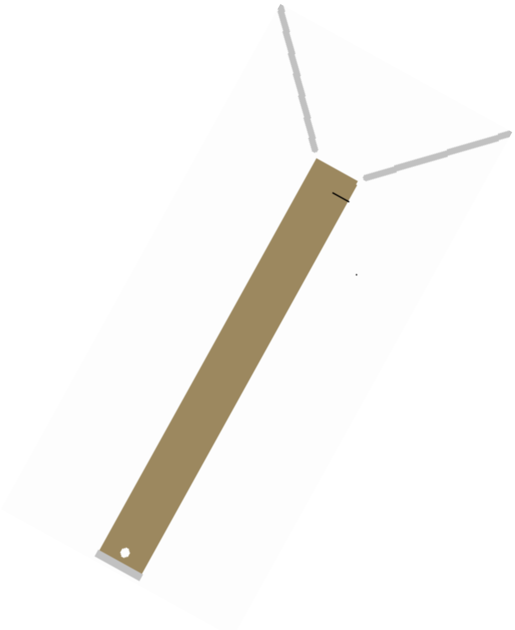
##

Deployment … spot the BatBug:









Some notes on deployment:

Orientation*.* The angle of reception is approximately 45°

Optimal siting. BatBugs can be suspended from a single point, by a deployment hook, that allows the instrument to be lifted at the end of an extensible pole of the kind used by window cleaners. To avoid rain pooling the BatBug should not be vertical but 30 degrees or more off vertical and facing into the space you wish to monitor.

The deployment hook can be used to hang a BB from a ledge in a cave. Pipes or gutters existing, or added to, buildings allow good concealment and consistent siting. It can be moored in water. This requires a flotation collar and small righting weight. The mooring line can be black, non-buoyant, and terminated at a stone placed in shallow water where you can find it.

Reflected sound from flat surfaces:

If a BatBug is laid flat on a smooth surface it is a good idea to lay it on a pad of folded old blanket or similar fabric or felt, 4cm or thicker, to eliminate echoes from the surface.

Mooring the BatBug in the water, or pointing up at 45degrees at the edge of the water will avoid echoes from the water surface.

Weather:

The BatBug is waterproof so rain, or water running down the fixing, is not a problem and does not cause any sounds to be recorded. Keeping the BatBug off vertical is necessary where it may be hit by rain drops as it helps water entering the microphone compartment to drain out through the small holes.

Where snow, falling leaves etc. may tend to accumulate on an upward facing BatBug, a conical mesh ‘witch’s hat’ can be fitted above to shed these.

Theft:

A major challenge in collecting years of continuous data is loss of instruments. Protected places like gardens are easy. For harder sits consider:

Inaccessibility: handing BatBugs in high places, mooring in water.

Natural disguise: stretchy ‘bark effect’ neoprene skins are available so that BatBugs can be made to resemble pieces of tree branches, and equipped with real twigs with dead leaves etc.… The cylindrical shape of BatBugs is useful here.

Rubbish disguise: BatBugs can be encased in artfully designed scraps of old, torn clothing to look like rubbish/litter – easy to spot, but uninteresting.

Concealment: BatBug holders resembling minor features of buildings (alarm boxes, water pipes etc.) can be attached to buildings, bridges etc. to hold BatBugs for long term studies.

Camouflage: traditional camouflage exteriors use natural disguise to reduce the chance of being spotted, but are not recommended as they reveal the object, once spotted, as definitely worth investigating…!

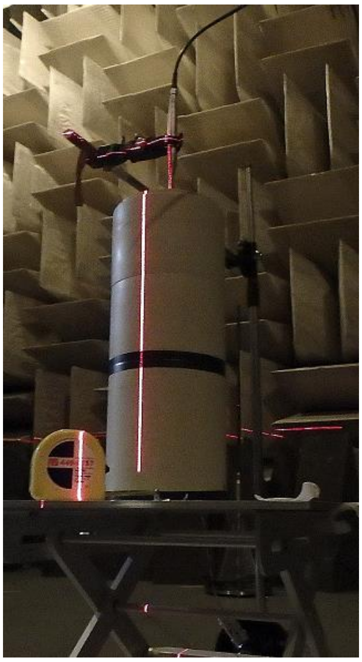
Vandalism: mostly the same issues as theft, but:

Inaccessibility: BatBugs hanging in a high spot generally survive stones thrown at them!

Send us your solutions to these problems!

Calibration

All BatBugs are tested against a calibrated ultrasound sound source, shown, right, undergoing its own calibration process at the National Physical Laboratory, UK.



SD cards

Any micro SD card up to 32GB will work in the BatBug.

The data goes into BATBUG*n*.CHE files, with a maximum size of 4GB, starting with BATBUG0.CHE. If the card fills up no further data is recorded. If files are already present or the BatBug restarts, then a series 0ATBUG*n*.CHE is created.

After reading the file into your computer, delete all the data files from the SD card.

If you use a blank SD card the last settings are used. The default settings (as supplied) are good for nearly all projects.

We strongly recommend installing this SD card formatter: https://www.sdcard.org/downloads/formatter/ from the SD Association. It is more reliable than others and only ever formats SD cards!

File types

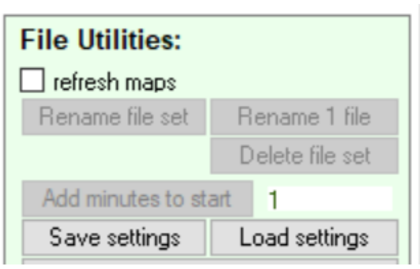
On the PC:

*.BB1 files contain the raw data and are the same as the SD card file except that they have a larger header that stores various items of information you add at the time the SD card is read, or later. In BB1 files every wave period (duration) is stored.

*.BB2 files contain only the data from those wave groups (segments of sound 8 cycles long) that have been automatically identified as belonging to calls. They also include information on call features and markers on wave groups to indicate a number of call features. These are listed below.

*.bm1, *.bm2 These are map files that enable rapid navigation of large data files. If you delete them and the software needs them, it re-maps the file and recreates the map file. If you manually rename, delete or move a data file you should delete or move the associated map files (that have the same name) to save disk space. If in doubt, delete!

.bbf These are small text files that you can create via the Files/filters page:



They enable you to return to any point of interest in any file.

On the SD card:

BATBUG*n*.CHE –data files.

BATSETS.txt – a settings file made using the BatBug software. These settings are read when the card is inserted (if batteries are present) and the BatBug will continue to use these settings until different ones are loaded.

BATTIME.txt – this can be made using the BatBug software to reset the clock, and must be erased after that, as it would reset the clock next time it's used.

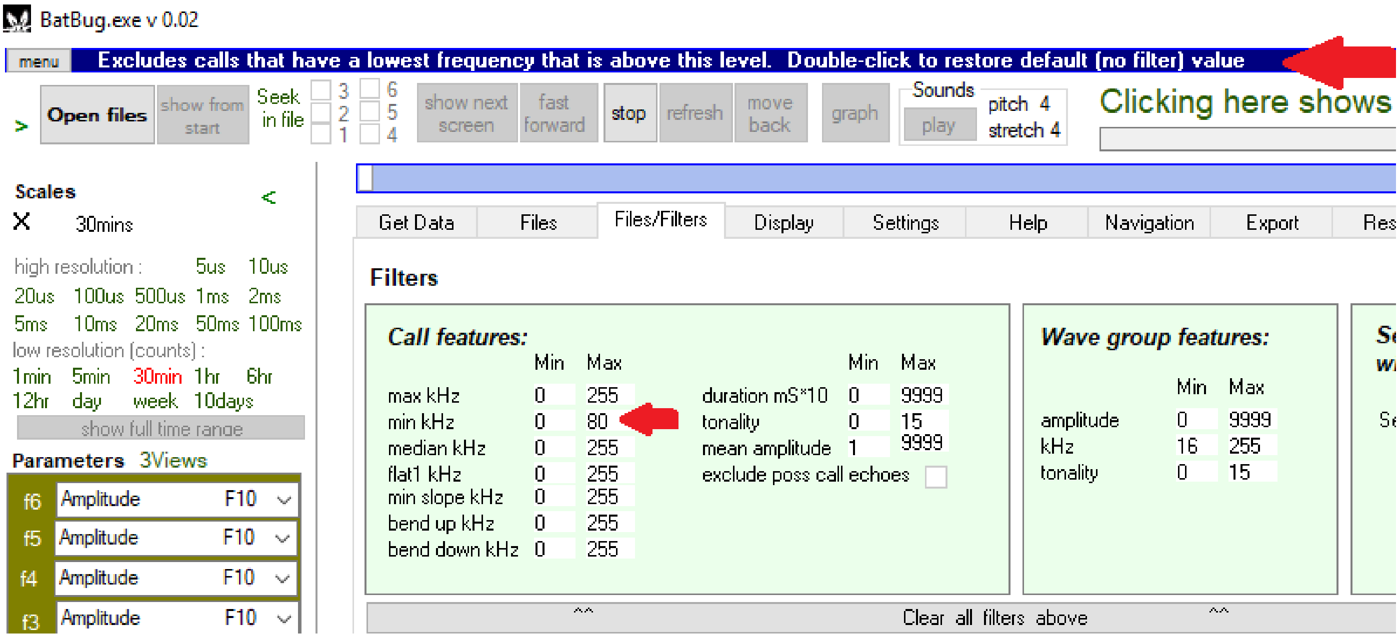
Software guide

BatBug.exe is freeware and can be downloaded from the Chelonia website and run on any Windows PC. The program is in portable format and does not require installation. Deleting the BatBug.exe file removes it completely. The screen needs to be at least 1080 pixels vertically. 4k screens make data inspection a joy.

A speaker or headset is needed to hear the playback of calls.

Hints

In BatBug.exe when the pointer goes over significant items on the screen useful hints are shown in the blue bar at the top of the screen. This one describes a filter:

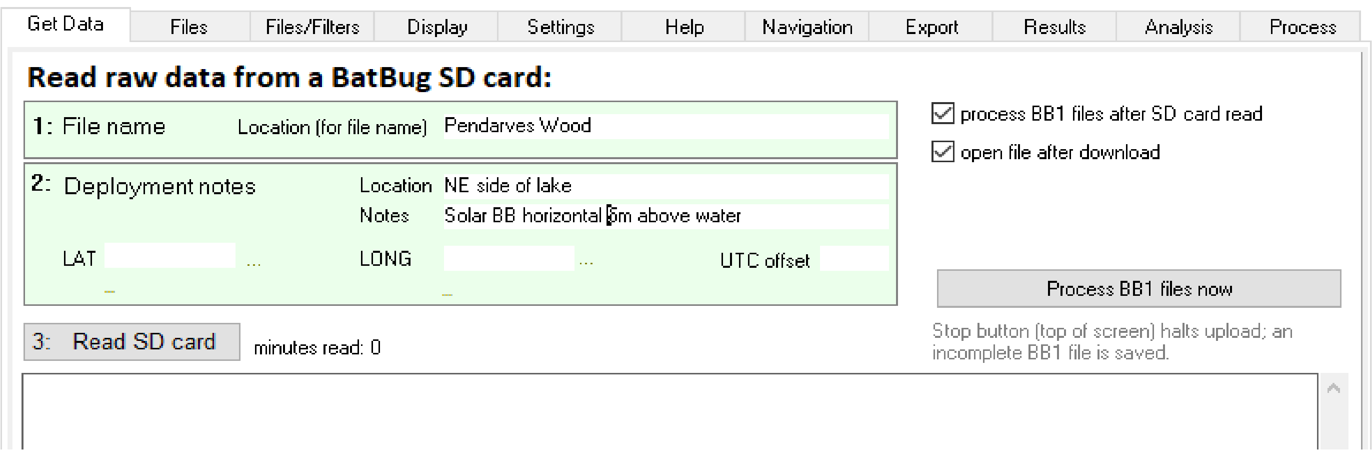


The **main MENU** is shown or hidden when the mouse pointer enters the blue bar, or when the large green text (Clicking here..' in this illustration) is clicked.

Reading data from the SD card

Go to the 'Get Data' page of the menu and enter the location of the BatBug in (1).

Add deployment notes if you wish and click 'Read SD card'. After the file is read it will be processed to make a BB2 file.



Keep all your files in one directory! The file name will have the location, date, and POD number, so you can sort files instantly using all these, and can carry out batch export etc. which will be much more time consuming if you put them in lots of directories.

Viewing BatBug Data

Data from BatBugs is in *.BB1 files, which are processed to produce *.BB2 files in which calls are identified. *.BB3 summary files in which bat passes (and, later, species) are identified. The process to create BB3 files is not yet developed. The SD card data cannot be viewed directly, but is converted to the BB1 file when the card is read.

A ‘file set’ consists of the BB1 and BB2 files.

Zooming

BATBUG.exe gives you a zoom range of 1 to 1 billion. So you can view many months of data on one screen and zoom in wherever you like to see the data on individual calls.

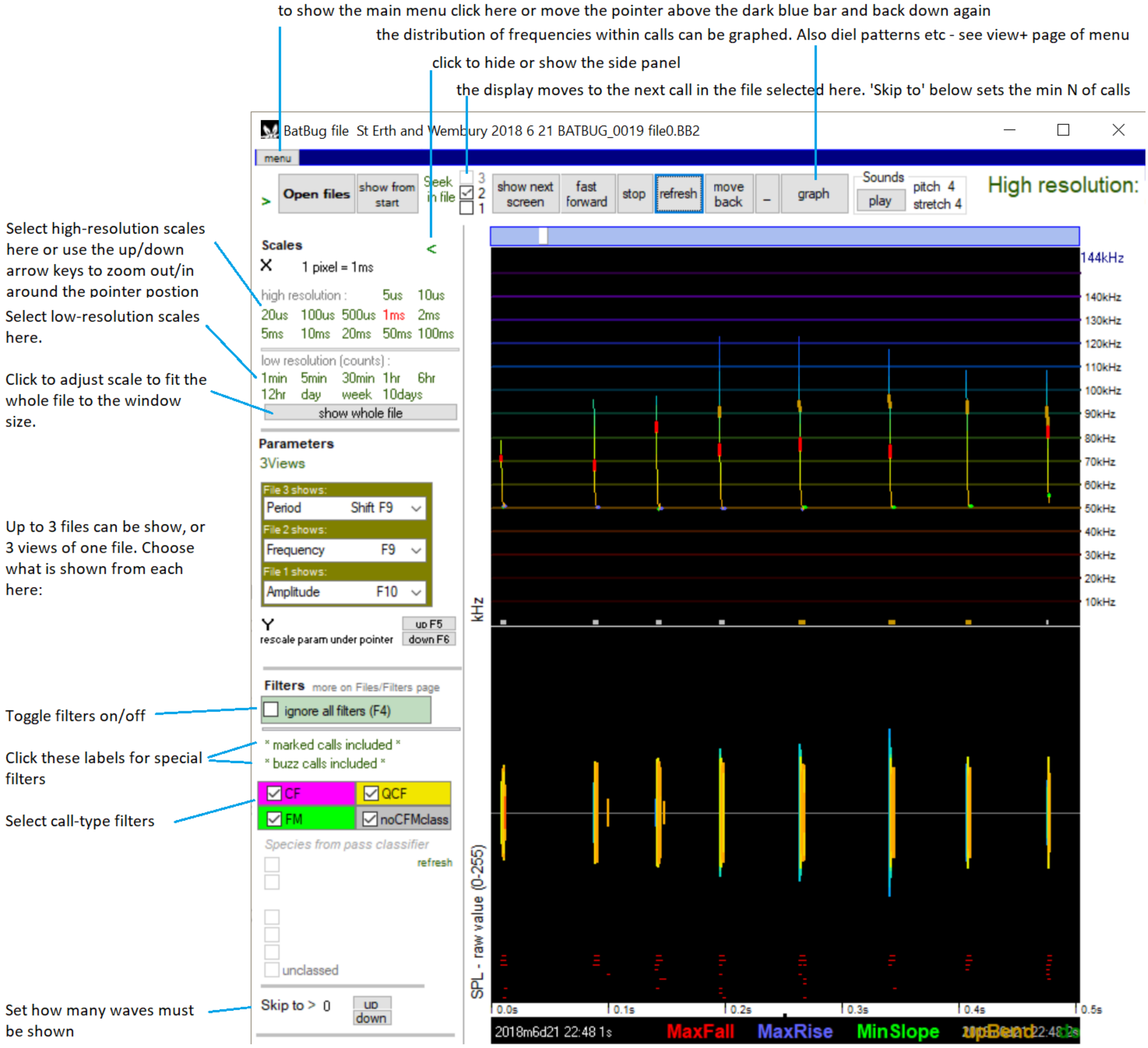
‘Low-res’ displays show counts of how many wave groups of bat data were logged, plus the distribution of frequencies, amplitudes or call types, plus temperature and light level.   The Graphs button shows the diel pattern - all the data on display is analysed by time of day so that you can see the average pattern of activity at different frequencies through 24 hours from noon to noon.

‘High-resolution’ displays show the frequency (or wavelength if you prefer) profile of the calls, and matches the kind of spectrum display you would see from a .wav file recorder, but with less information on harmonics - however the most significant harmonic structure, a dominant second harmonic plus higher harmonics, may be shown.

Data is viewed in high resolution when you zoom in, showing the features of each call. When you zoom out it switches to **low resolution**, showing you averages of frequencies in calls over minutes or longer periods.

When several days are displayed the average daily pattern of activity can be graphed.

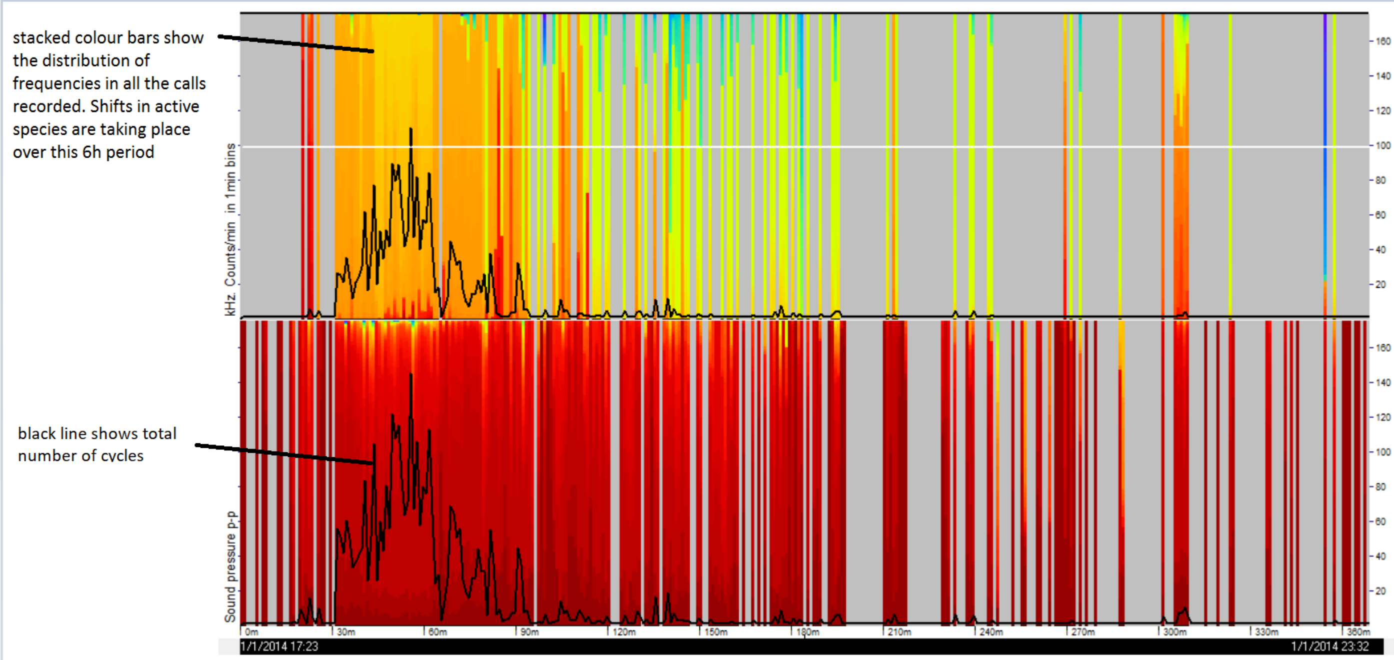
‘**High-resolution’** display:



Batbug.exe can now display 6 files in synchrony.

Low resolution display

… does not show individual calls, but shows averages of all those calls that pass any filters that have been set:

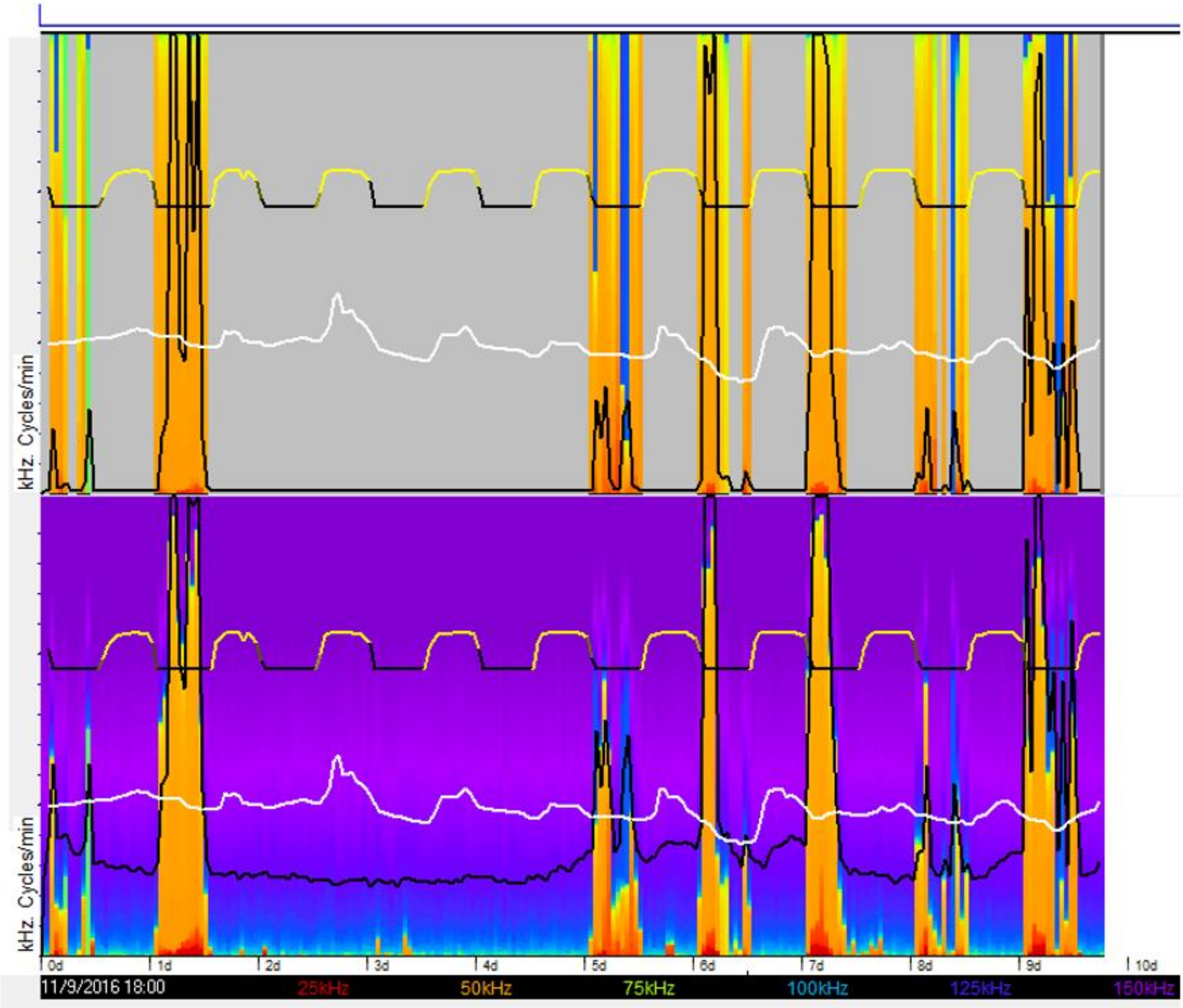


Here 10 nights are shown:

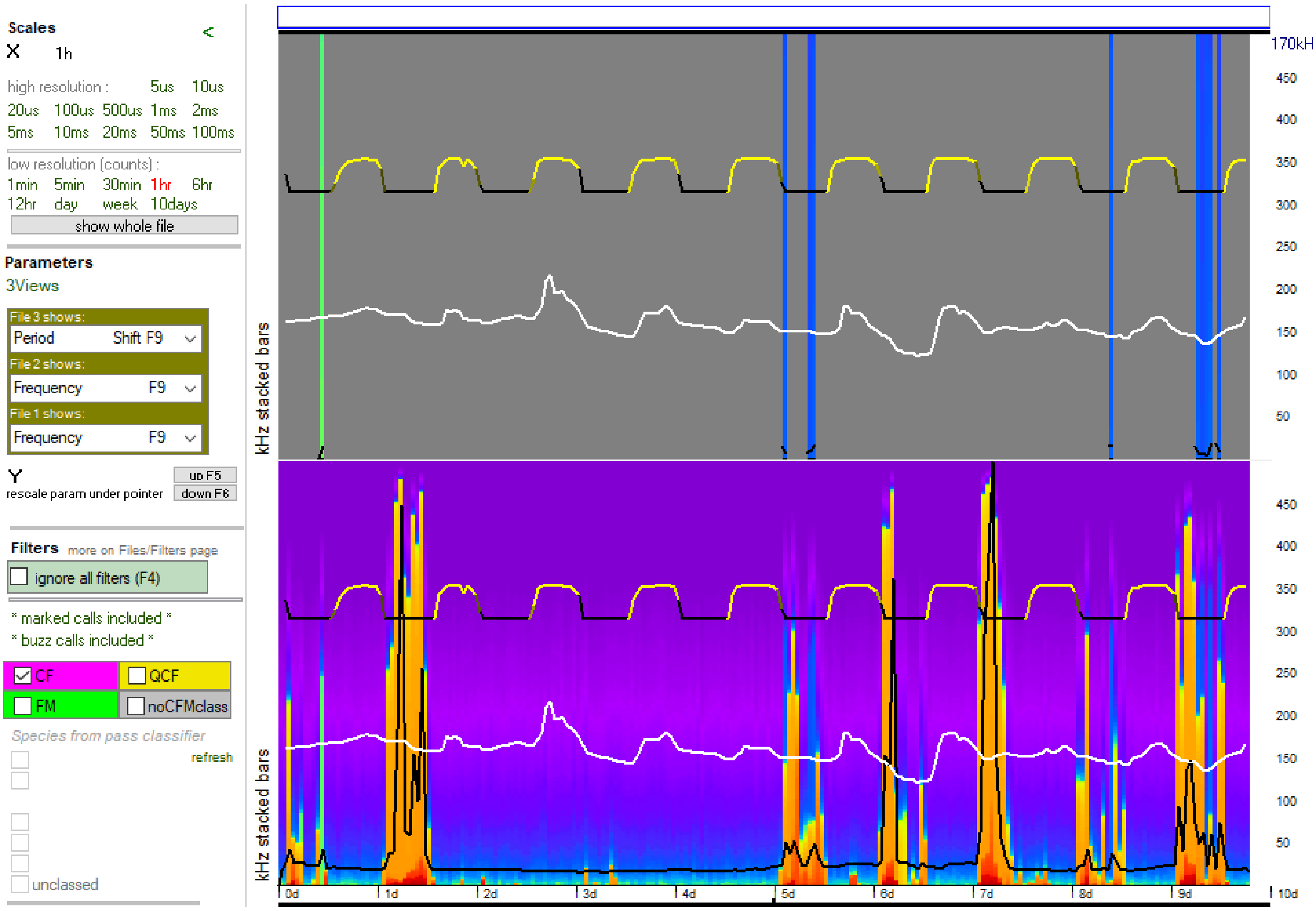
The yellow/black line is light intensity. The white line is temperature. The black line is number of 8-cycle groups.

In the BB1 file – lower panel – there is a lot of high frequency noise, shown in purple, that does not appear in the BB2 file – upper panel – as it is not in the form of calls.

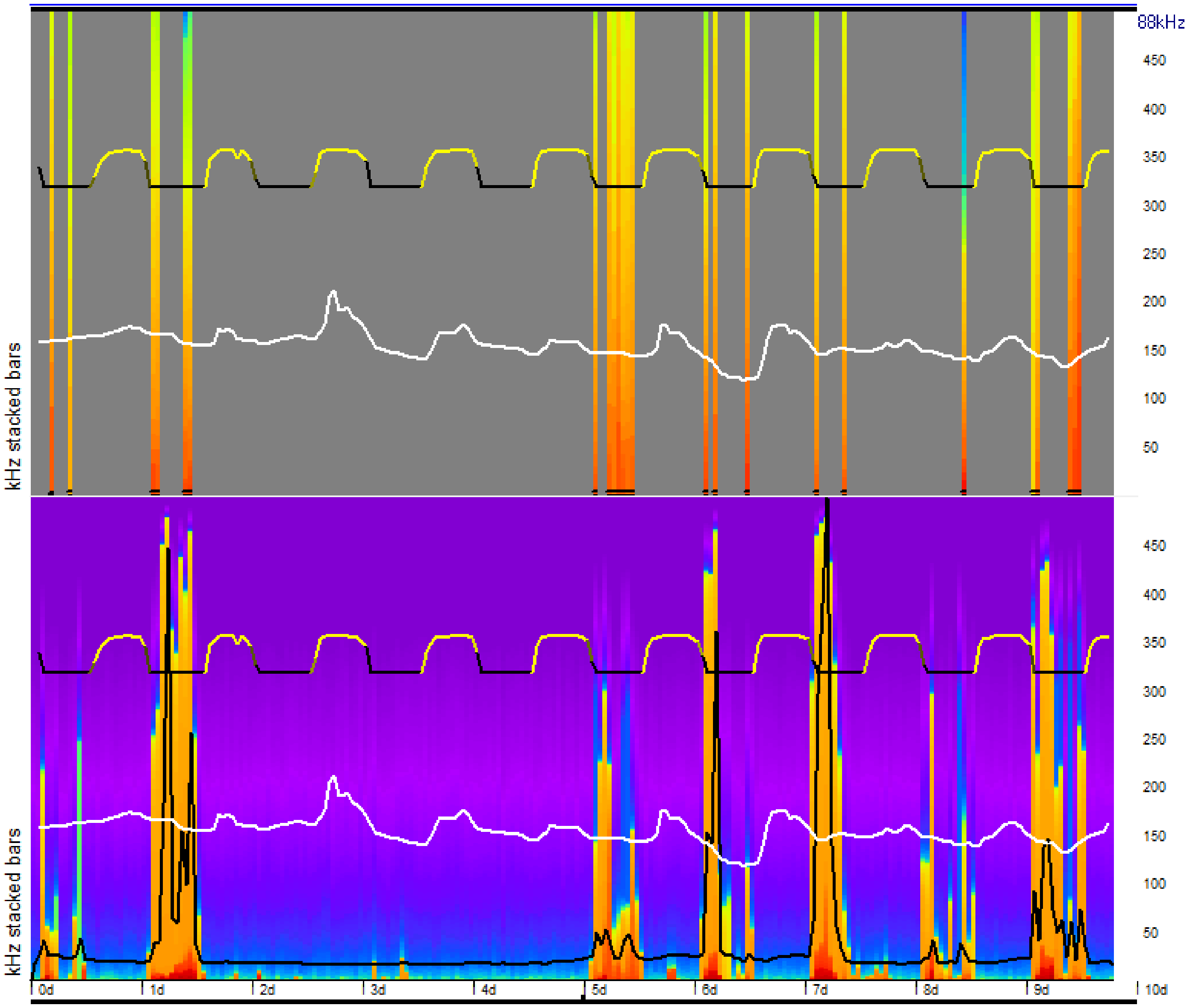
Three nights had no bats although the nights before and after had many. The bat calls show a range of frequency, with the blue being Lesser Horseshoes, the light green being Greater Horseshoes, and the orange being Pipistrelles.



Here the data is filtered to show CF – constant frequency – calls, so the Greater Horseshoe and Lesser Horseshoe calls are selected:

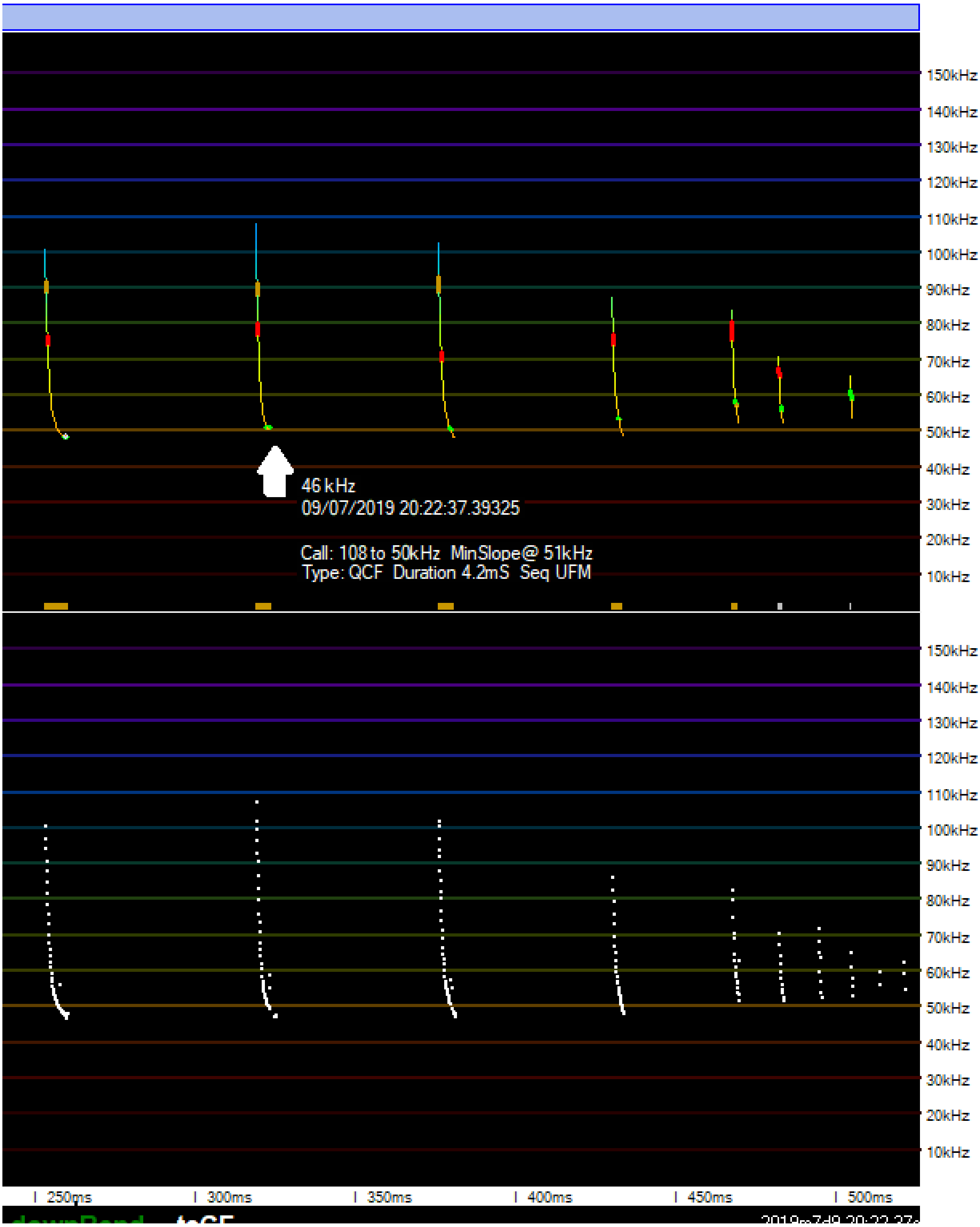


Here the filter is set to FM bats:



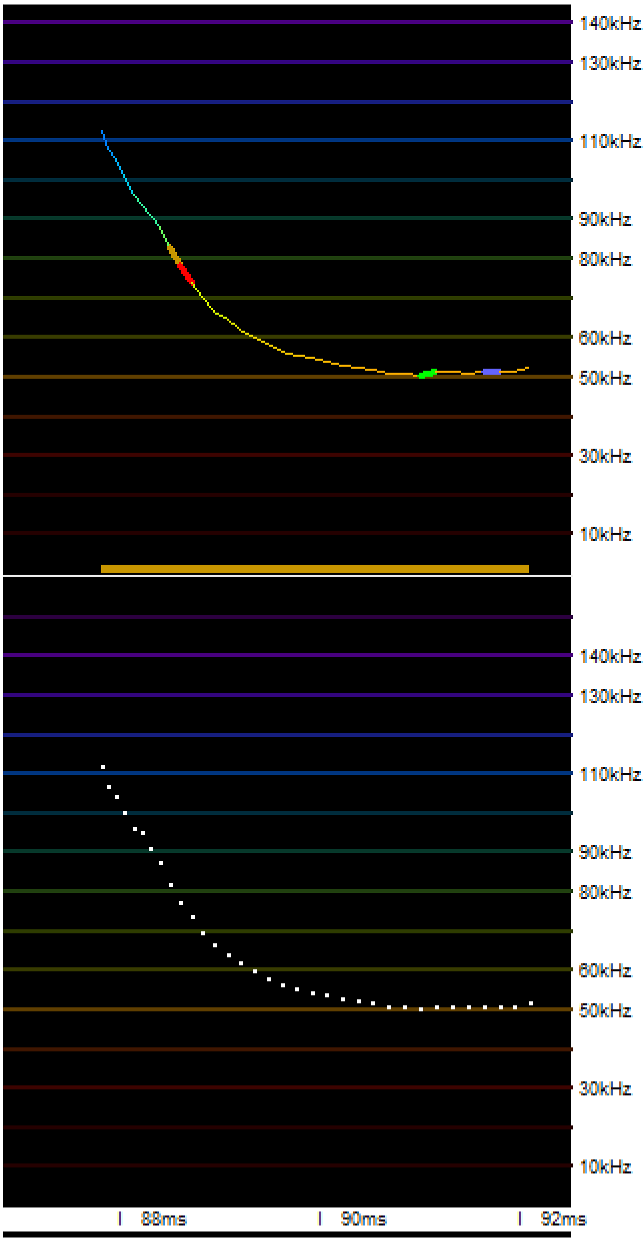
The encounter on the 8th night is conspicuous for the high frequency content, shown as the blue end of the spectrum.

Zooming in to high res shows more detail. A brief summary of call features is shown from BB2 files when the mouse pointer is in or near their time range :

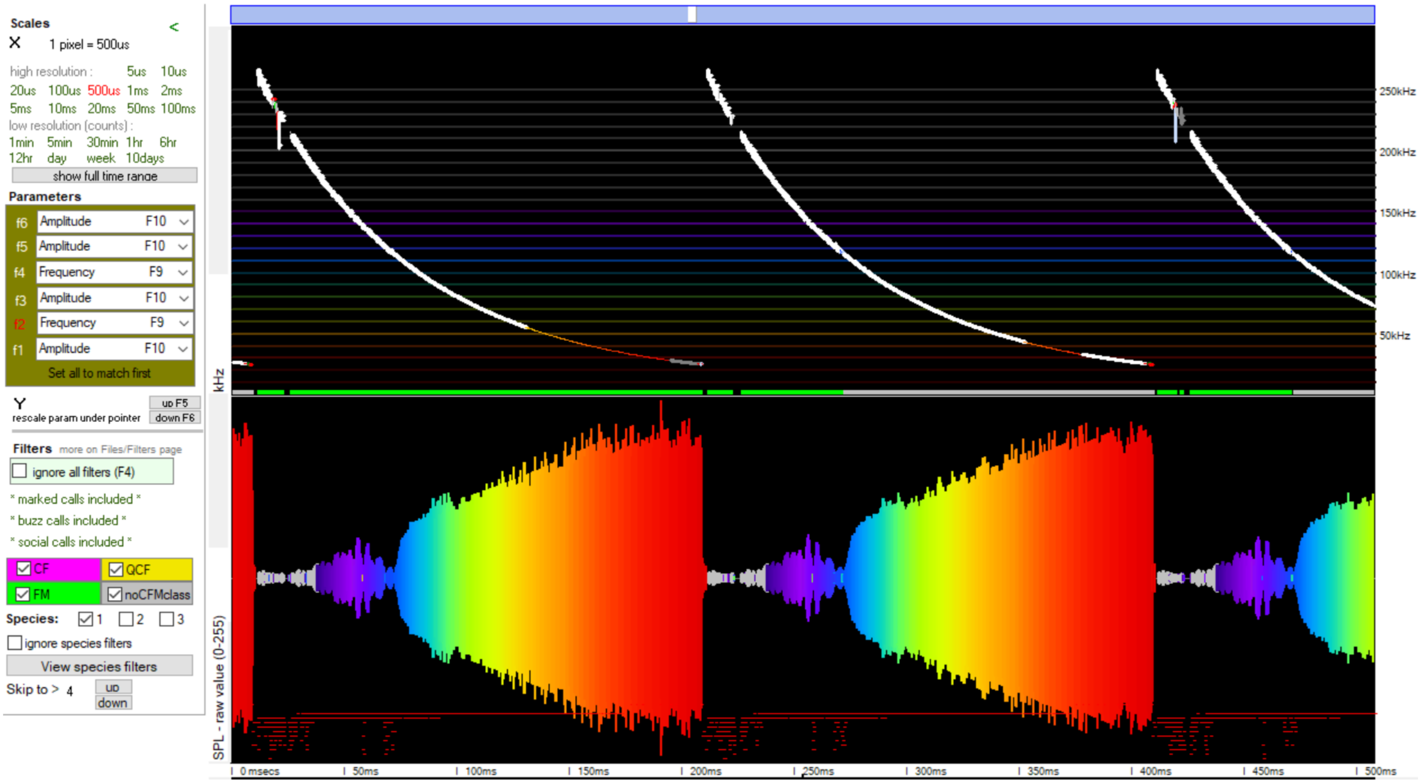


The small green mark – minimum slope – on these calls is a useful indication, in lower time-resolution views ,that this is a QCF call.

Zooming in further shows:



This view shows the sound pressure profile in the lower panel, the frequency (kHz) in the next panel. This is test data – there are no bats with frequencies at the level (270kHz) detected here.

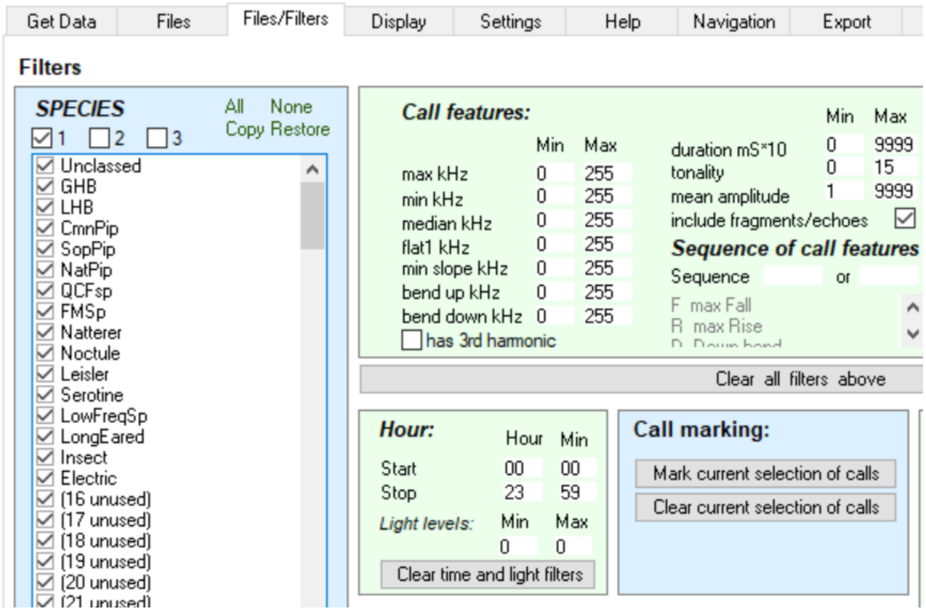


Editing

In High resolution mode it is possible to 'mark' the call that is to the right of the mouse pointer via the right click menu. This can be used to remove any misclassified calls.

Clicking the '* marked clicks included' label steps through 'marked clicks excluded' and 'marked calls only'.

Any selection of calls can be marked, or cleared, using the buttons bottom right, on the Files/Filters page:



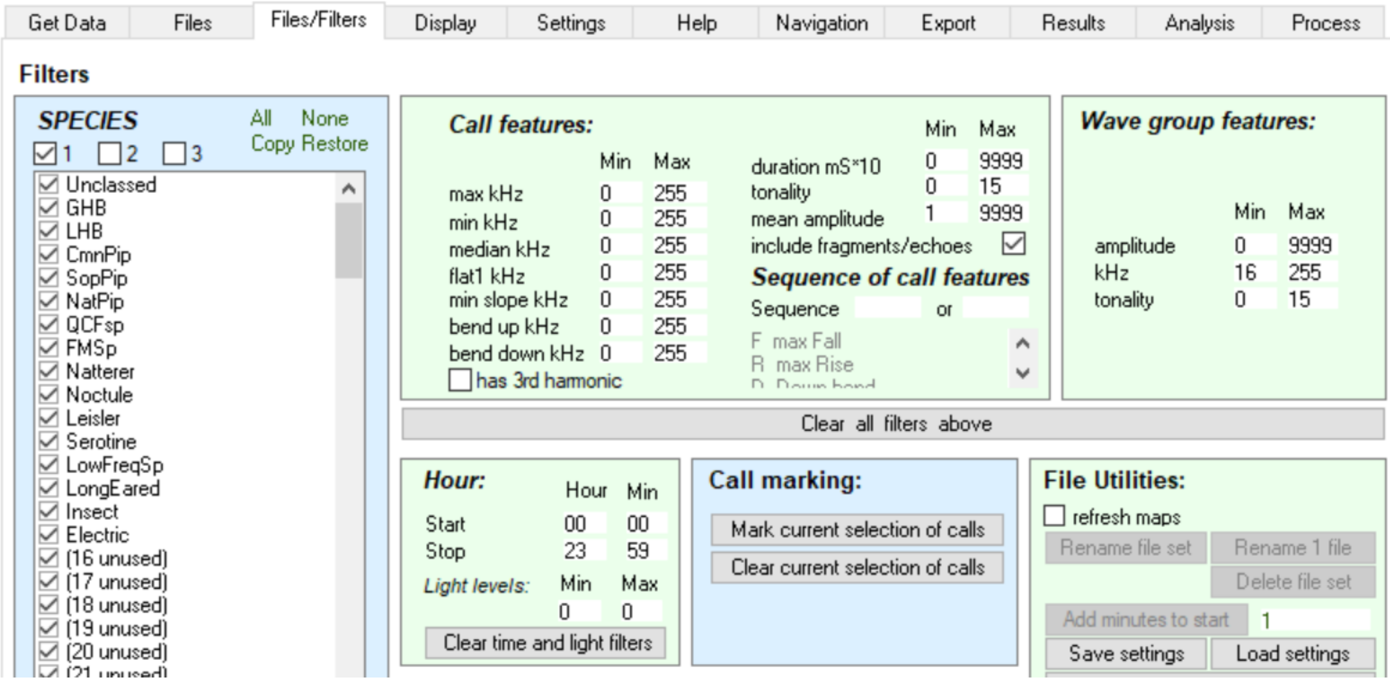
Navigation

You can rapidly navigate to any particular time and zoom in to see details of individual calls at that time. The main methods are:

- Drag the slider at the top of the display. As you do that the slider will expand to show the date and time that will be displayed if you stop dragging it.
- Use the up and down arrows to zoom in and out around the current position of the pointer.
- Use the right and left arrows to show data later than or earlier than the pointer position.
- Hold the mouse button down and move the pointer left or right to highlight a small section of data. The display will zoom in to approximately that section.
- Select a display resolution on the panel to the left of the graphical display.
- Mark any single point of interest via the pop-up menu that appears when you right-click within the display area. You can then return to it via the right-click menu the next time you open that file.

Saving settings and source details

The buttons, bottom right, below, are really useful. When you see something interesting, or have a good set of filters, hit 'Save settings' and give it an informative name 'Day flying pip' or 'Leisler filters'. A .bbf file is created and you can return to the same place in the same file very quickly, with all your filters and display set up, or you can choose to restore the filters but not re-open that file:



File listing

The Files page of the menu lists the six files that can be open at one time, in their vertical order on screen:

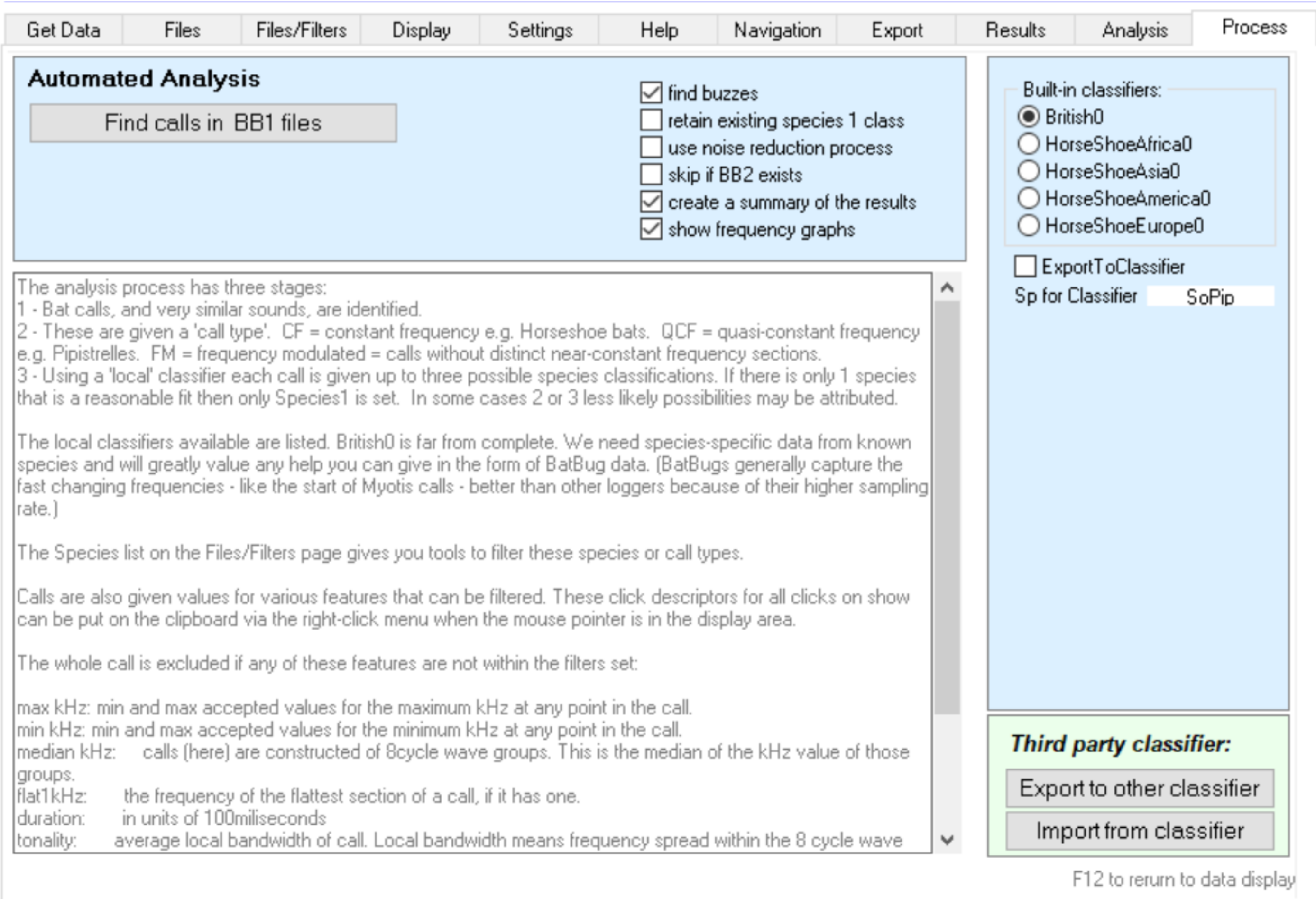
Data is read from the SD card into a *.BB1 file on your PC that has a name like 'Bostraze 2014 08 14.BB1' It's a really good idea to use this name format and keep all the files from one project in one directory. The filename is constructed from:

- Location, that you enter in the software e.g. 'Bostraze'.
- the start date from the BatBug clock.
- .BB1 = raw data file. Archive these files and delete the BATBUGn.CHE files from the SD card.

With this name structure the files can, in Windows, be sorted by location and then date.

*Tip: keep all files from a project **in the same directory** for easy selection display and processing.*

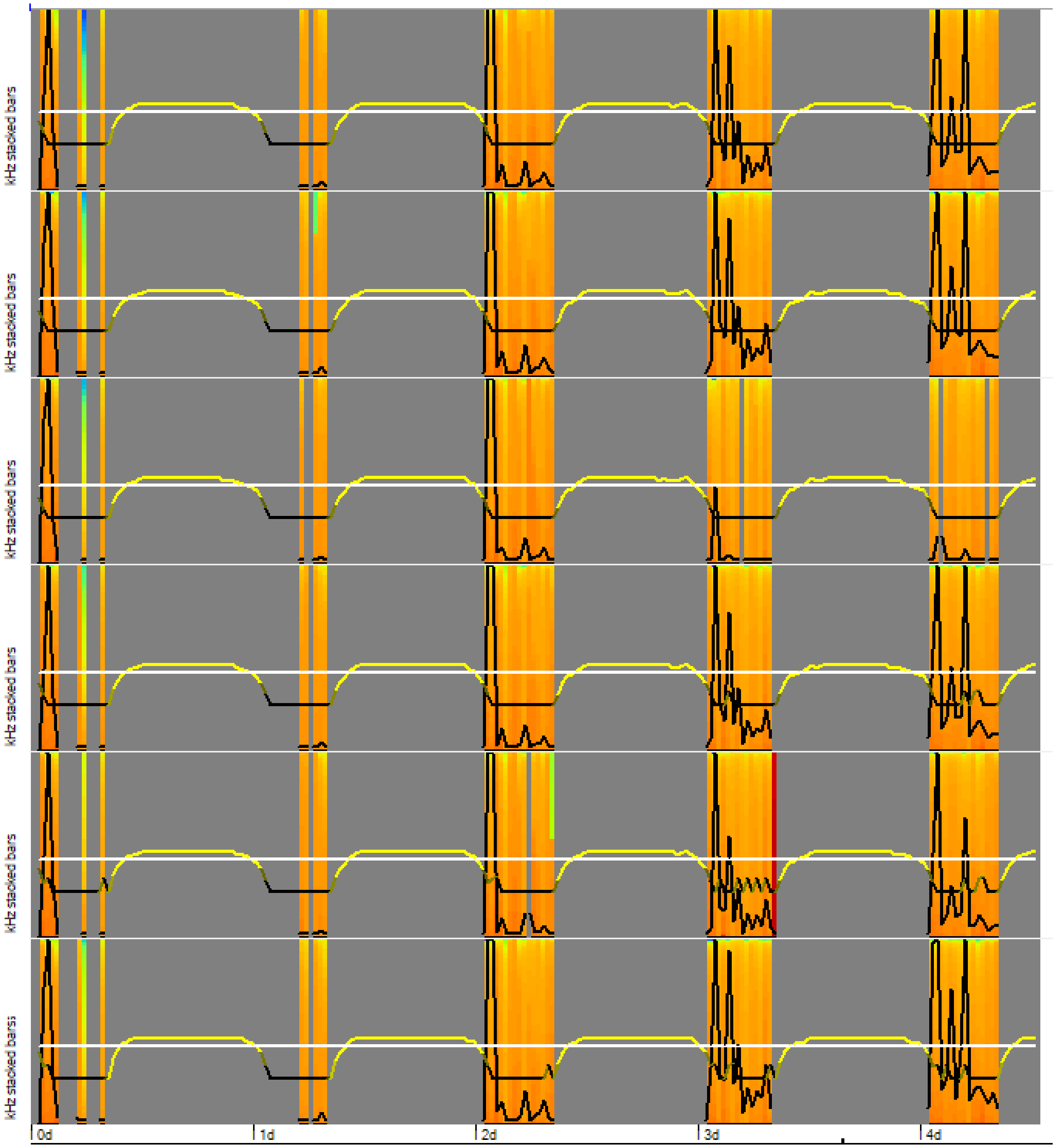
BB1 files can cover over a year, and contain the raw sound data and the temperature + light level at 1 minute intervals, including when the BatBug is on standby. BB1 files are processed to make slightly shorter BB2 files, either automatically after reading the SD card, or when you press the 'Find calls in a batch of BB1 files' button on the Process page of the menu:



Not all these classifiers or functions are currently available. In most cases some information is shown at the top of the BatBug window when the mouse pointer is over the item.

BB2 files show the results of the software's process for identifying calls from the mass of wave groups recorded, and include only those wave groups that are in calls.

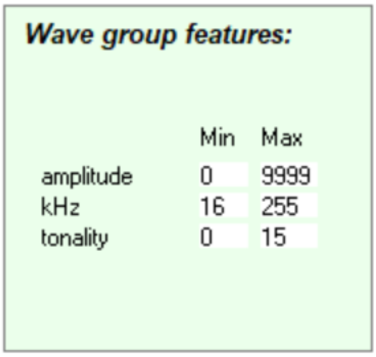
Here 6 files from simultaneous tests of different hardware are shown together (the temperature sensors have been disabled in these instruments)



… number 4 is not doing at all well!

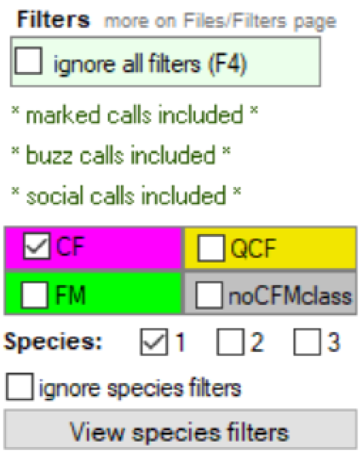
Filters

In BB1 files data is 'wave groups' each 8 cycles (waves) long. These can be filtered by amplitude, frequency, and bandwidth.



In BB2 files the wave groups are grouped into calls. These can also be filtered by features of the whole call.   An alarmingly large choice of filters is possible, but you don't have to master all of these!

Unless 'ignore all filters' is in use then all the filters set must be passed by any call for it to be displayed or exported. These include the guild or group filters shown here:



'marked' calls are those you have marked either using the options shown in the right-click popup menu, or by marking a whole set of clicks selected by filtering ... see below.

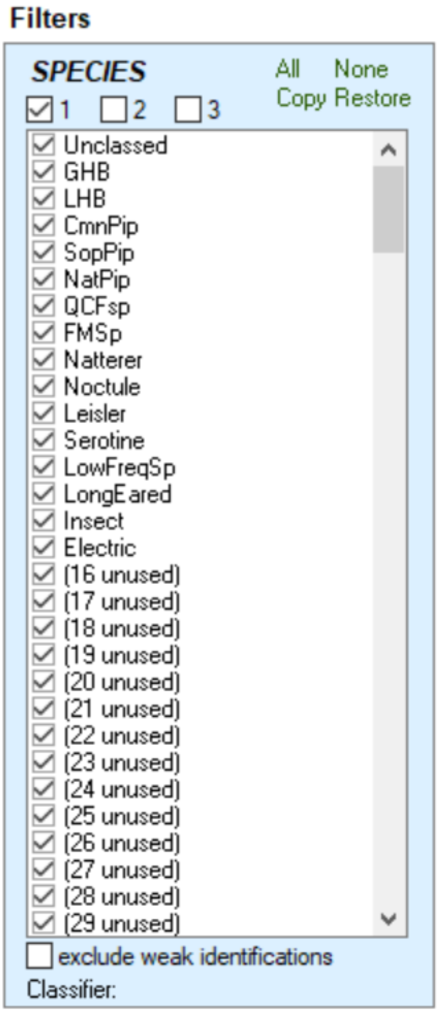
'buzz calls' are possible feeding buzzes found automatically by the process that finds calls and creates the .BB2 file.

'social calls' are not yet identified.

CF = constant frequency – horseshoe bats.

QCF = quasi-constant frequency = calls like those from Pipistrelles that have a section that changes frequency only slowly, but it usually far from 'constant' frequency.

FM = frequency modulated calls, often rapid frequency downsweeps.



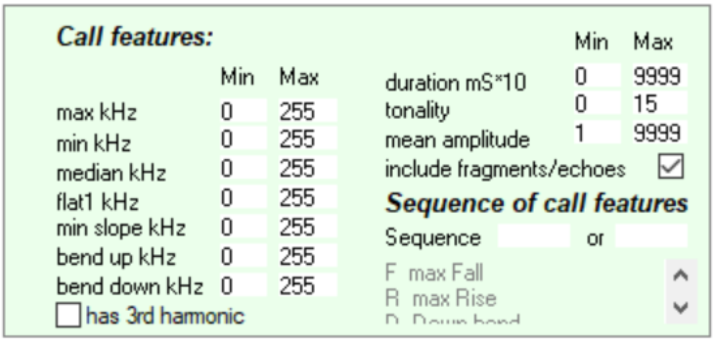
In addition to these very broad filters there is, or will be, a set of species filters associated with a set of species or species groups (guilds) identified by the specific classifier used. Each call has up to 3 classifications attached to it, in decreasing order of confidence. They are selected by the 1,2,3 boxes at the top on the Filters page.

To view, say, Soprano Pipistrelle detections you click 'None' to remove all selected species, and then click SopPip, so this is now the only species selected. Then view the data in low or high resolution to see those detections if any.

To exclude, say, Common Pipistrelle detections, click 'All' and then click CmnPip to deselect it.

Copy and *Restore* allow you to temporarily save a set of options and go back to it easily.

Independently of the species classifications you can filter by the features of each call:



‘Sequence’ allows you to find similar calls. The stepped CF call shown above would be found by entering ‘CFC’ in the sequence filter box.

Data can also be filtered by time and light levels:

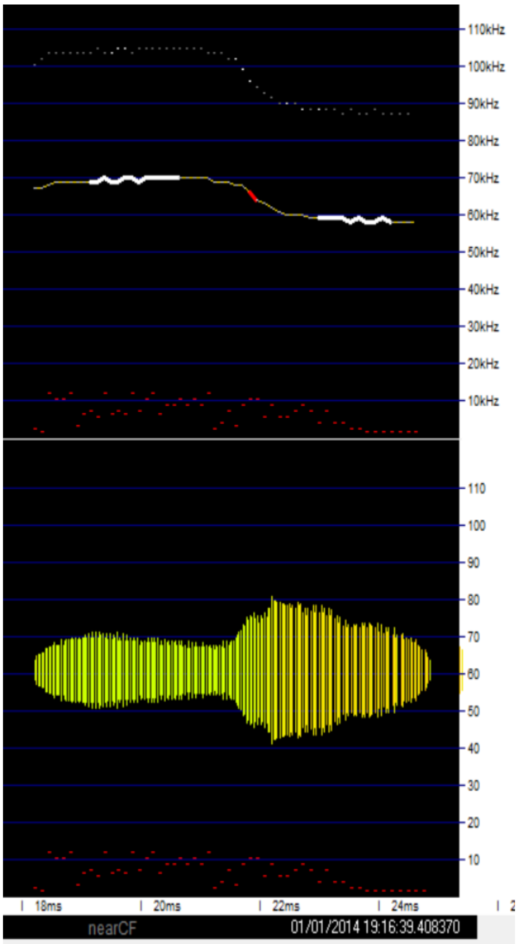
Data, filtered or unfiltered, can be exported from files or huge sets of files via the Export page of the menu, or just from the data shown on screen, via the right-click menu when the pointer is over the display.

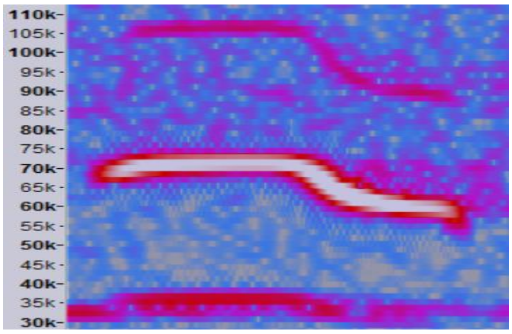
BatBug and Wave File data

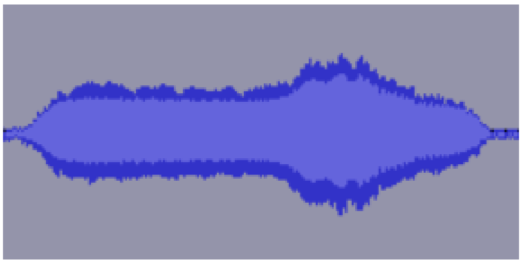
Wave files *.wav are the standard format for sound recording, and bat detectors that make this kind of record are sometimes called ‘time expansion’ detectors as they make calls audible to humans by playing them back more slowly than they were recorded. Some such files can be converted to ‘virtual BatBug’ files as here:

<table>
<tr><td></td><td>BatBug data</td><td>.Wav file data</td></tr>
<tr>
<td>In the example, right, the lower panel shows the BB1 file amplitude data of a call from a Central American bat. The upper panel shows the BB2 data, in which calls are identified, and the frequency displayed.

The frequency contour shows three colour-coded sections. The two thick white line sections have a constant frequency, although the second is characteristically lower in this species, while the red section identifies the frequency of the maximum rate of fall of the frequency.

The dotted line in the upper part of the display shows a third harmonic. This can be identified where the second harmonic is dominant, as in many constant frequency bat calls.</td>
<td>A single bat call:


BatBug data allows the fine structure of CF calls to be explored.</td>
<td>

The BB1 data in this case were generated from the .wav file. The spectrum is shown above and the amplitude below:

Large sets of .wav files can be converted to BatBug files by the BatBug.exe freeware.</td>
</tr>
<tr>
<td colspan="2">The red dots in the lower part of each panel show a measure of the tonality or bandwidth of that part of the call. Parts of the call that contain a very narrow range of frequencies show as high on this scale. Very quiet calls usually show as low values because reverberation, ambient noise and electronic noise increase the bandwidth i.e. make it a less pure tone. Even numbered harmonics do not contribute to this.</td>
<td></td>
</tr>
</table>

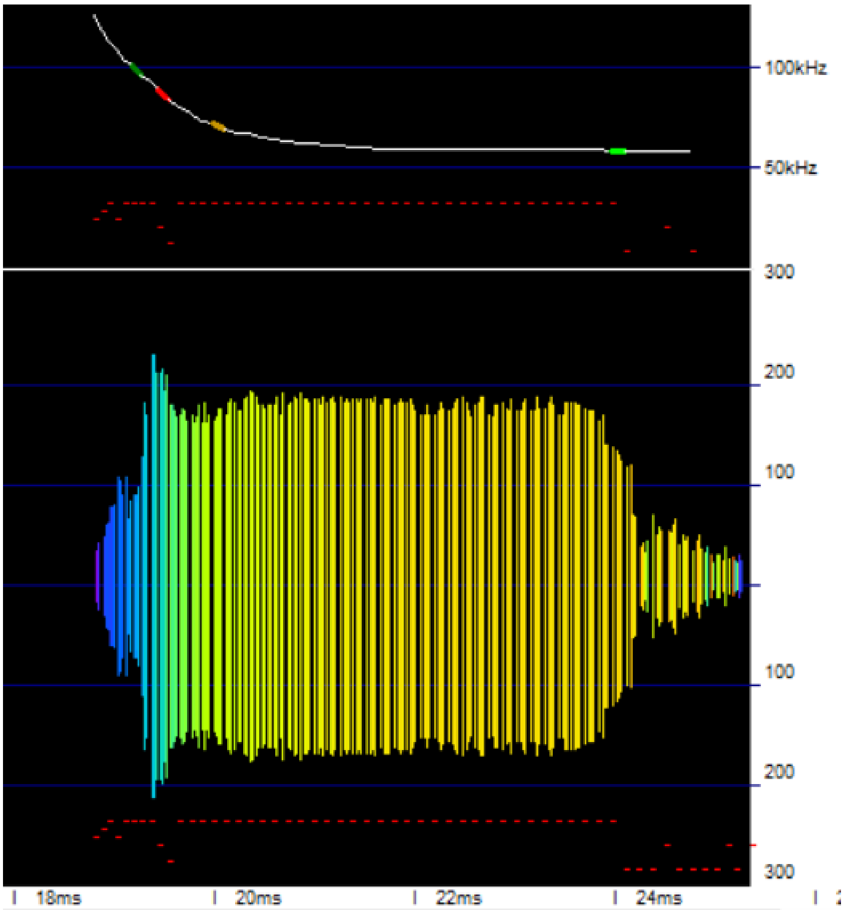
Bat Calls and their Features

Call descriptions. These contain 30+ different measures of the call that are potentially useful in the process of identifying species. For any one call several will be valuable measures and several will have little value.

Time	Date and time
5us	Time within minute. 5 microseconds precision.
avF	Frequency of wave group in kHz
SPL	Average sound pressure level of wave group
Q	Bandwidth – higher values = lower bandwidth
ErrN	
Pk1+3 etc.	Average amplitude of first and third wave peaks in wave group
IPI1+3 etc.	Average interpeak interval preceding of first and third waves in wave group
Call descriptors	
CallN	Starting at 1 in each new minute.
Buzz	Identified buzzes and these can be filtered via the clickable label in the panel at the left of the display * marked calls included * * buzz calls included *
CF_FM	Calls are put into one of these categories when the BB2 file is made: call type: ☐ FM ☐ noCFclass ☑ CF ☐ QCF ☐ possCF By filtering for, say, 75-85kHz and 'CF' – constant frequency – Greater Horseshoe bat calls are picked out.
Hmic2	This identifies calls that have a dominant second harmonic, plus a third harmonic. This pattern is seen in the calls of Rhinolophid (Horseshoe) bats and in the later part of the call of Plecotus (Long Eared) bats.
Fmin	Minimum kHz in the call.
Fmax	Maximum kHz in the call.
Fstart	Initial kHz of call.
Fend	kHz of end of the call.
Frange	Max kHz – Min kHz
Fmedian	Frequency of 50^{th} centile of wavelengths in the call. This is preferred to sound amplitude measures as these are strongly affected by the relative direction of bat and microphone, by microphone characteristics and by the distance of the bat from the microphone, and by the humidity which increases the absorption of higher frequencies.
Fav	Reciprocal of average of all wavelengths in the call.
FbodyHi	Frequency of 75^{th} centile of wavelengths.
FbodyLo	Frequency 25^{th} centile of wavelengths.
MinSlope	Frequency of lowest rate of change of wave period
MinSlopeKHZ	kHz at time point of lowest slope of wavelengths
'UpBend@	The frequency of the fastest slowing of a frequency fall.
UpSharp	Measure of the rate of deceleration at the UpBend
'DownBend@	the frequency of the fastest slowing of a frequency rise.

DownSharp	Measure of the rate of deceleration at the DownBend
Flat1Khz	kHz at lowest rate of frequency change if low.
Flat1Slope	Rate of change of frequency at Flat1KHz
F1flatness	Low values indicate minimal frequency change through flat section. Arbitrary scale.
CFdur	Duration of flat section as number of cycles/8
AmpAv	Average amplitude through whole call. Not currently corrected for microphone response.
Nmissing	Number of 8 cycle sections of a call that were not recorded or rejected.
nOct	Number of 8 cycle sections recorded in a call

This is a Soprano pipistrelle call:



The right-click menu in the display area allows you to put the call parameters into the clipboard and paste them into a spreadsheet:

Fmin	Fmax	Fstart	Fend	Frange	FmaxEnergy	Fav	FbodyHi	FbodyLo
58	126	126	58	68	59	68	60	58
MinSlope	MinSlopeKHZ	UpBend@	UpSharp	DownBend@	DownSharp			
0	58	69	5	96	3			

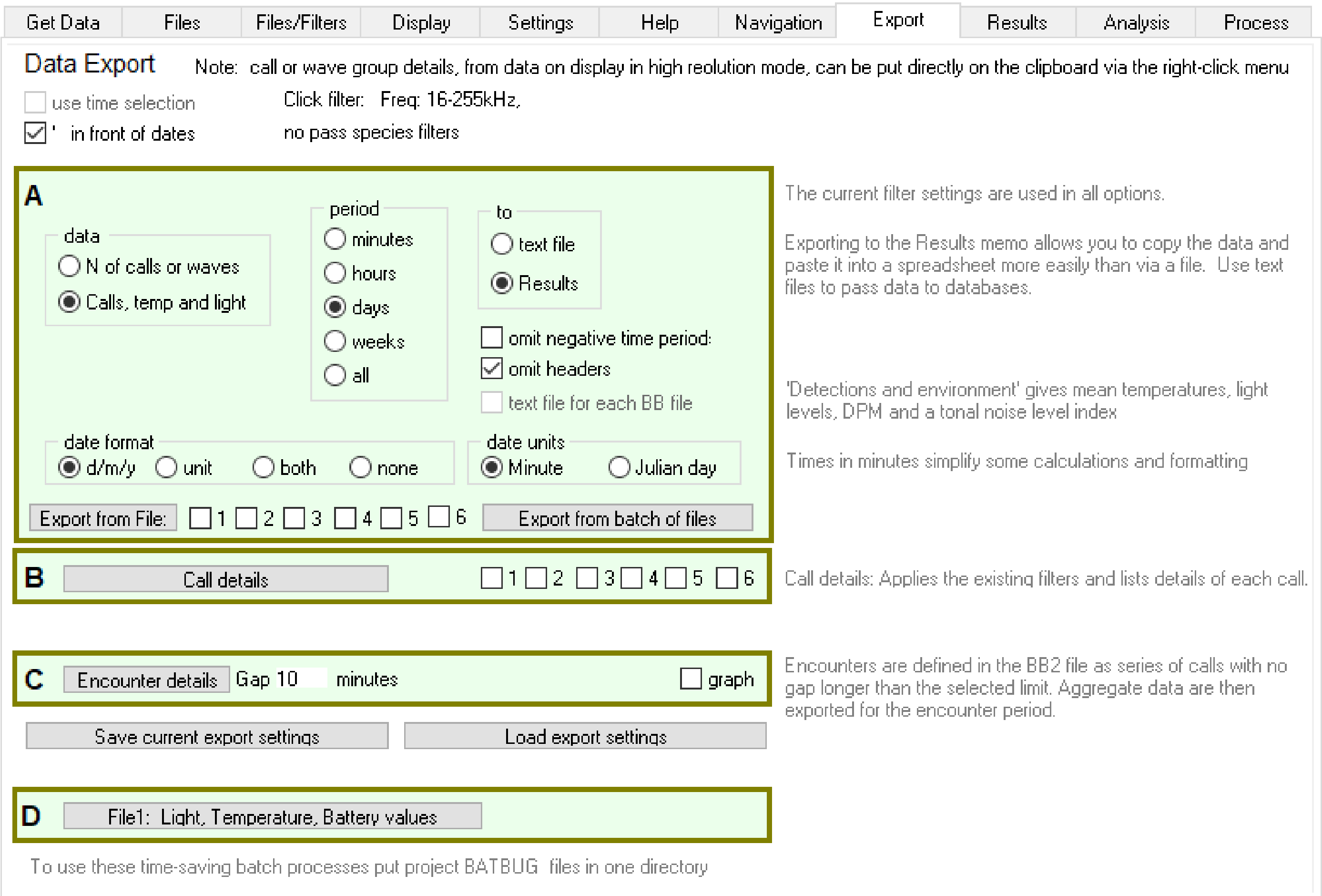
Call feature marks – these are shown with colour codes below the display panel.

Without these markers the last two features can be surprisingly hard to see. We are very good at seeing bends in straight lines, but not very good at seeing bends in curves, although they are easy to identify mathematically. Often switching the display to show the wave period (the time duration of a single wave in the call) instead of the frequency makes the position of the bend clearer by straightening the curve that is shown in the frequency display.

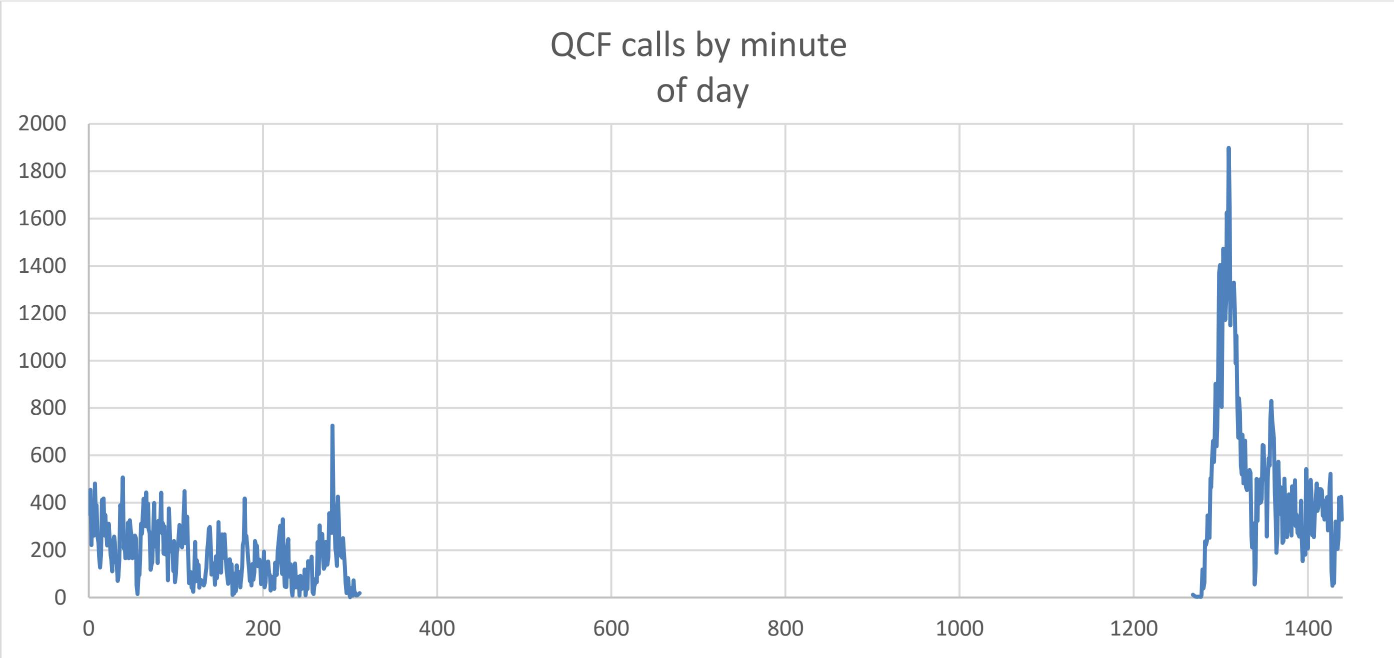
This is because bats more often modulate the wavelength, rather than the frequency, in a linear fashion. Because of this all the calculations in BatBug.exe are done on wavelength rather than frequency, and this does give a somewhat different, but still consistent, result on the timing, in a call, of 'bends' in the frequency trend.

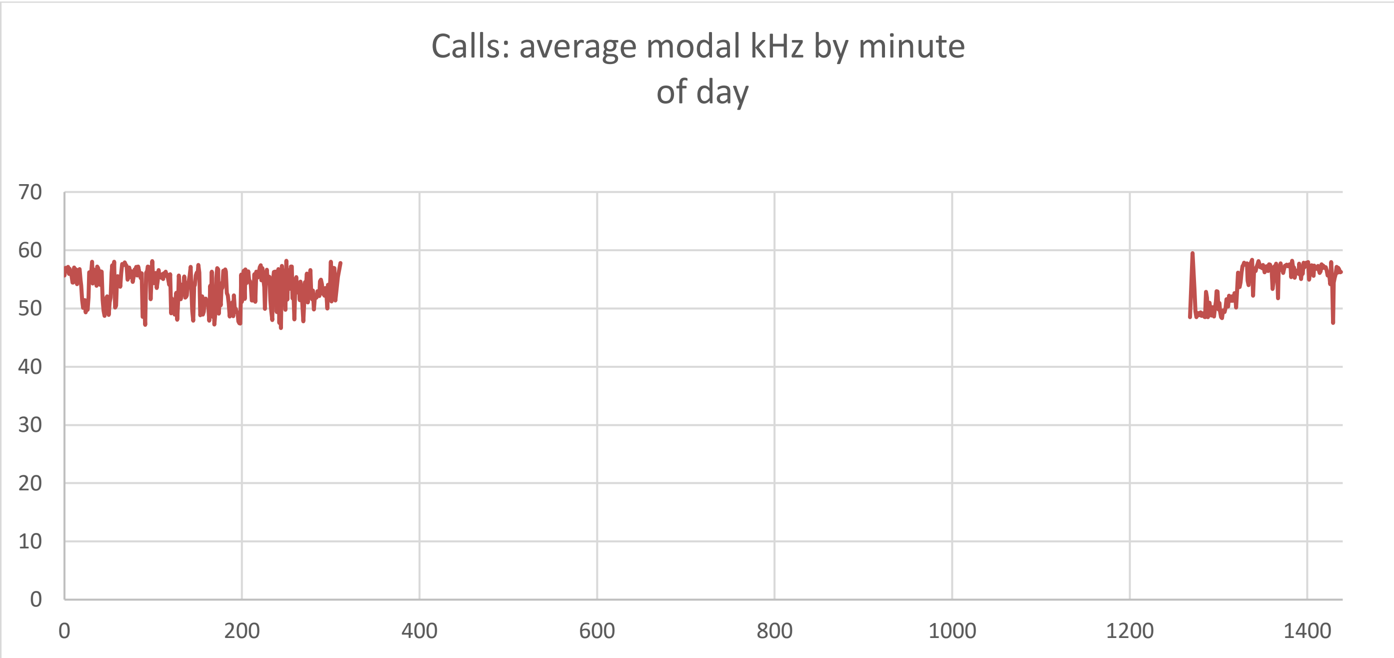
The call features that can be marked graphically, and are shown in the filters menu.

Data export



Call details can be imported directly into a database, such as access. Within 20minutes of reading the data it is easy to obtain these graphs of 140k calls, having run the automated analysis which took less than 1 minute for 28 days of data:





It seems that the early frequencies are rather lower, than later. Is that due to a different mix of species, a change in the actual calls used, or ???

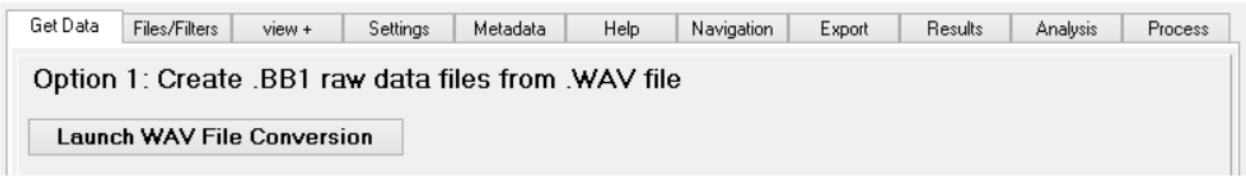
Wav file conversion

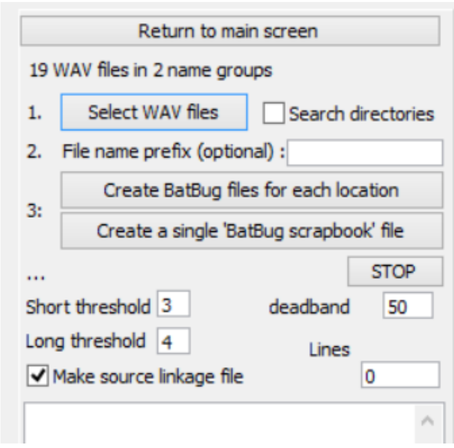
Software / converting wav files to BatBug files : **currently not available:**

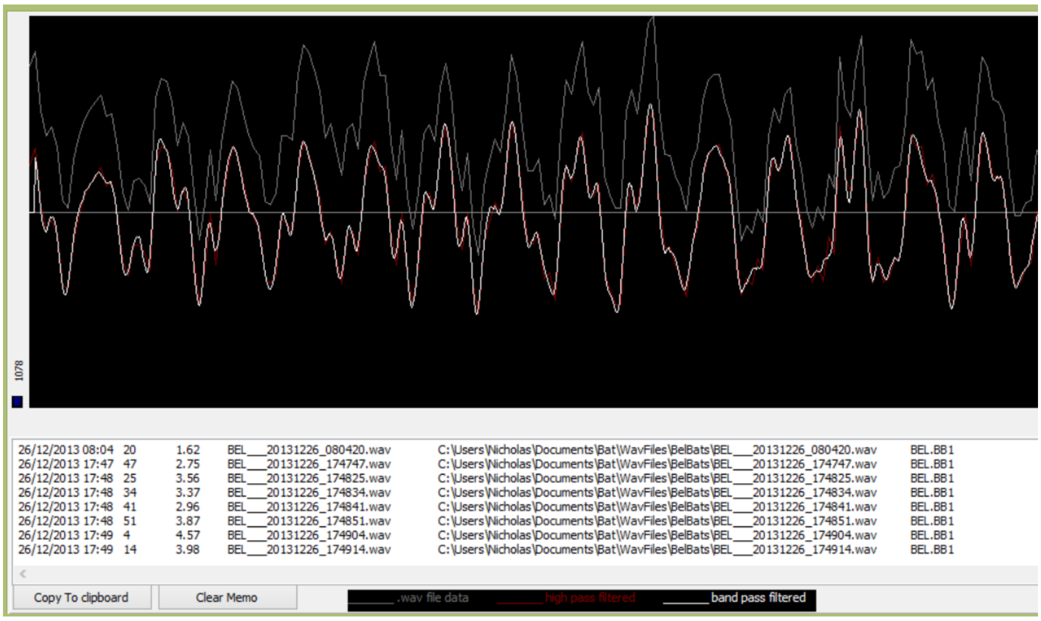
A utility is included in BatBug.exe that allows large numbers of *.wav files recorded at 384k samples per second to be converted to single 'virtual BatBug' files, in which they are grouped together by the place-name in the file names, with the times of the files used to construct a single *.BB1 file. In this format the data from a year or more can be seen on a single screen, graphs of daily patterns of activity can be shown etc. No temperature or light measures will be available and, in comparison with a native BatBug file there will be a smaller frequency range, and uneven frequency response, particularly in the case of SM2Bat data collected with MEMS microphones (UXS and UXT). The problems with 'clipping', where the SM2Bat is overloaded by a bat at very close range, are largely corrected during the conversion – graphs are shown of this filtering process.

Conversion typically reduces the file volume by a factor of more than 100 and often more than 1000, so 1 TB of files can become a single file of less than 1MB. In BatBug.exe it is possible to see any period in a year at which the frequency profile is unusual and zoom in instantly to see the call profiles. The name of the *.wav file from which a particular section of data came can be obtained from BatBug.

Go to the menu, select the 'Get Data' page, click the 'Launch WAV File Conversion' button,







Data: third party support

Chelonia is happy to support any research using BatBugs by facilitating data access or handling and is willing to create data export and re-import utilities for this. Please let us know what might be useful to you.

Signal processing / how it works

Background:

The BatBug has been developed using experience and methods from the development of Chelonia's 'F-POD', a dolphin and porpoise echo-location logger. It is the successor to the C-POD which is the industry standard for cetacean monitoring, and has been used successfully to monitor endangered species, such as the Vaquita, to study fishery by-catch, to assess the distribution of Harbour Porpoises across the Baltic Sea etc. That work has contributed a wealth of experience to the development of the BatBug, particularly because cetacean echolocation is much harder to identify than the calls of most bats as the clicks are much shorter and far less distinctive

The analysis of large data sets can be prohibitively expensive if visual inspection of detections is required. The C-POD has been successfully used in very large projects, e.g. to measure the population size of the Baltic Sea Harbour Porpoise. In the SAMBAH project 400 logger-years of data were obtained from a grid array of 300 C-PODs across the Baltic from Denmark to Finland, capturing echolocation clicks at up to 160kHz. Highly efficient data selection and compression, plus automated processing with very low false positive rates, were a key to the success of this project, as recording continuous .wav files would have generated so much data that reading the SD cards would, alone, have taken a year! The automated post-processing achieved an error rate of less than one false-positive second per year of logging.

Data collected:

The BatBug samples sound at 1 million samples per second. This is up-sampled to 4MHz to provide the minimum time resolution for the wavelet transform.

A compressed record of every narrowband sequence of 8 cycles is stored. This consists of the wavelengths of each waves. Amplitudes of pairs of waves are also stored. The pairs are not recorded as a simple sequence 1,2: 2,3: etc. but are interleaved 1,3: 2,4: etc. as this helps in the recognition of a distinctive harmonic structure – a dominant 2nd harmonic with significant energy in the 3rd harmonic that is a useful signature of some bat species.

This data set is substantially richer in information than the more highly compressed data stored by the brilliantly innovative ANABAT recorder, but is not as rich as that from some .wav file recorders. However, because of the BatBug's high sampling rate it can deliver more detailed information on, for example, the fine structure of constant-frequency bat calls than .wav file records give. Calls can be reconstructed in .wav file format and the Discrete Fourier Transform can then be used to produce a conventional spectral plot, but this is not currently implemented in the software, which, instead, shows the frequency structure of the calls using the wavelength data. This is both much faster and gives higher frequency resolution.

The advantage of this approach to the signal processing is that the data volume is typically reduced to below 1% of that produced by a sound-level-triggered .wav file recording system. This makes long deployments possible, but only if the instrument's power consumption is low.

Power consumption:

The BatBug uses a low-power FPGA (a field programmable gate array) as its data processor, in a system managed by a PIC microcontroller. The FPGA is capable of parallel computing which keeps clock speeds down and saves power. However it constrains the signal processing which has been develop don the basis of dedicated field and research to maximise the computational efficiency of the algorithm used.

Species identification:

The variability of calls within many species is large enough to create overlap in call features between species. False positive rates from automated detection systems vary depending on what other species are present. False negative rates are also affected because classification to species may need to be more conservative where similar species are present.

The approach implemented in the BatBug software is to:

Pick out and characterise calls in the data.

Pick out buzzes that may be composed of weak and ill-defined calls that would not be picked out in (1) above.

Classify calls into 'guilds'. At present the only classifier is 'BritGuilds' that classifies calls into:

CF – constant frequency – Two frequency bands corresponding to Greater and Lesser Horseshoe Bats.

QCF – quasi-constant frequency – Pipistrelle calls that show a downsweep followed by a nearly flat segment of the call. The terminal rise in frequency is not a required feature for this class.

FM – frequency modulated – calls showing a rapid downsweep with no flattening off, as produced by Myotis bats.

Unclassed – everything that doesn't fit the other classes.

This classifier is rudimentary as we don't have enough data from known species to develop it further. Any BatBug data that has known species will help develop this and will be gratefully received!