



Spectrum Lab is one of many applications that can display a signal in the frequency domain, i.e. showing the make up of its frequency spectrum.

Many of these other applications are aimed at SDR-based receivers so that you can operate them in a similar fashion to a traditional radio receiver. Examples are SDR Sharp (SDR#) and HDSDR. Spectrum Lab is not intended to do this so comparisons miss the point.

The secret of Spectrum Lab is in the name – it's a laboratory for analysing signals.

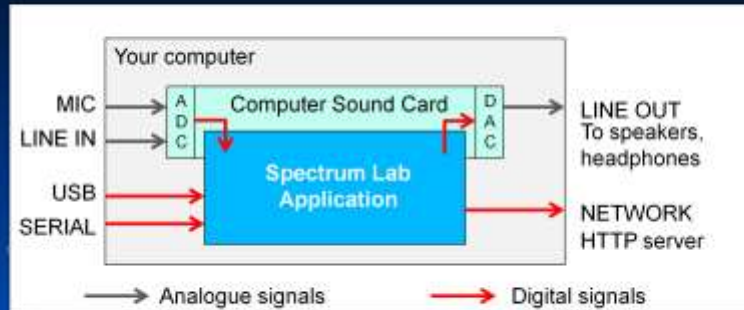
## AGENDA

1. What is Spectrum Lab?
2. The Fast Fourier Transform (FFT)
3. Spectrum Lab in Meteor Scatter work
4. Setting up Spectrum Lab
5. Further information



# 1. What is Spectrum Lab?

- An audio analysis application developed by Wolfgang Buescher
- Runs on Windows and Linux (under Wine)
- A flexible, configurable, development tool



Spectrum Lab is an application that works in conjunction with the sound card in your computer.

It uses the Analogue to Digital Converter on your sound card to convert audio signals into digital data which it then processes.

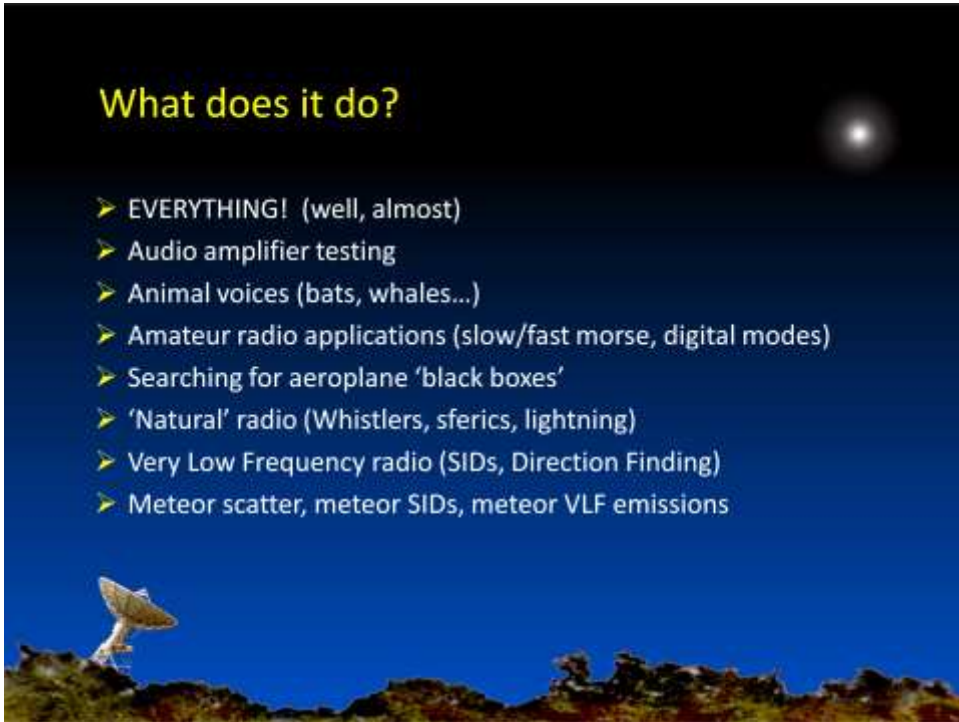
At the end of this processing Spectrum Lab displays the results on the computer screen and perhaps saves them as files on the Hard Disk.

If needed, Spectrum Lab uses the Digital to Analogue Converter on your sound card to convert the processed signal back to analogue for playing over headphones or a loudspeaker.

Spectrum Lab can also work with external devices such as an external sound card or an SDR receiver, via the USB port, or audio data delivered via the Serial and Network ports

## What does it do?

- EVERYTHING! (well, almost)
- Audio amplifier testing
- Animal voices (bats, whales...)
- Amateur radio applications (slow/fast morse, digital modes)
- Searching for aeroplane 'black boxes'
- 'Natural' radio (Whistlers, sferics, lightning)
- Very Low Frequency radio (SIDs, Direction Finding)
- Meteor scatter, meteor SIDs, meteor VLF emissions

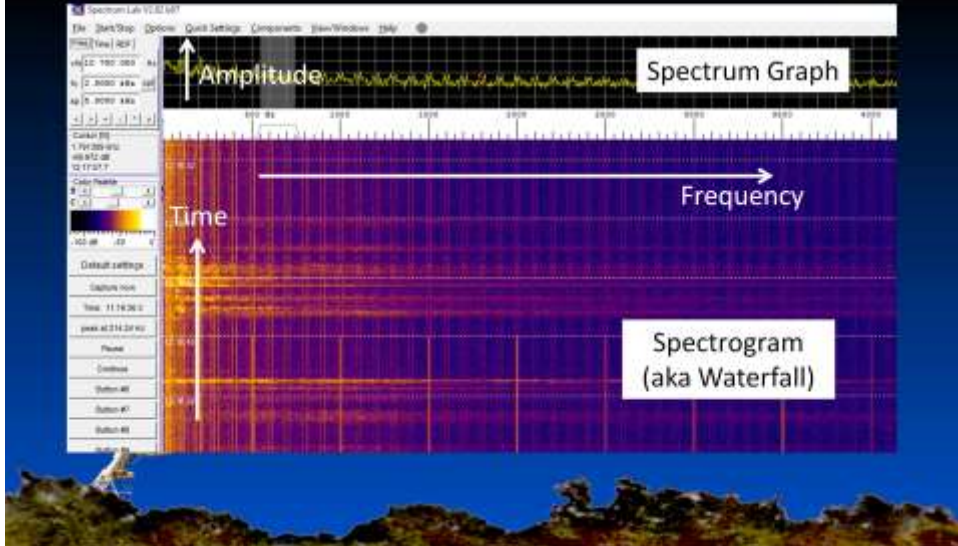


Spectrum Lab started life as an application to experiment with sound signals and analyse the performance of audio amplifiers.

It has grown (and continues to grow) organically (= chaotically) as people have suggested new applications.

This makes it a difficult application to use for the beginner, but it does have a configuration facility that enables you to set it up for a particular purpose without getting into the detail.

## Default screen layout

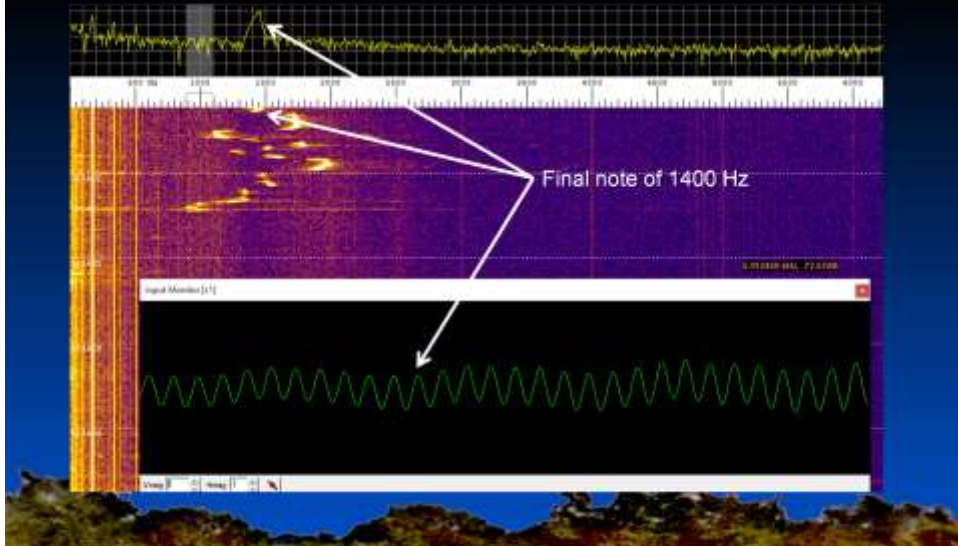


This is the factory default configuration that you will see when you first load Spectrum lab. If you are using a laptop it will probably be displaying the signal from the microphone and the output will be going to your loudspeaker, so if you don't have the speaker muted you will quickly get a howl and a white screen as you get increasing feedback from one to the other.

The Spectrum Graph shows the spectrum as a line plot. One axis of this graph is the frequency, the other is the amplitude (linear or logarithmic)

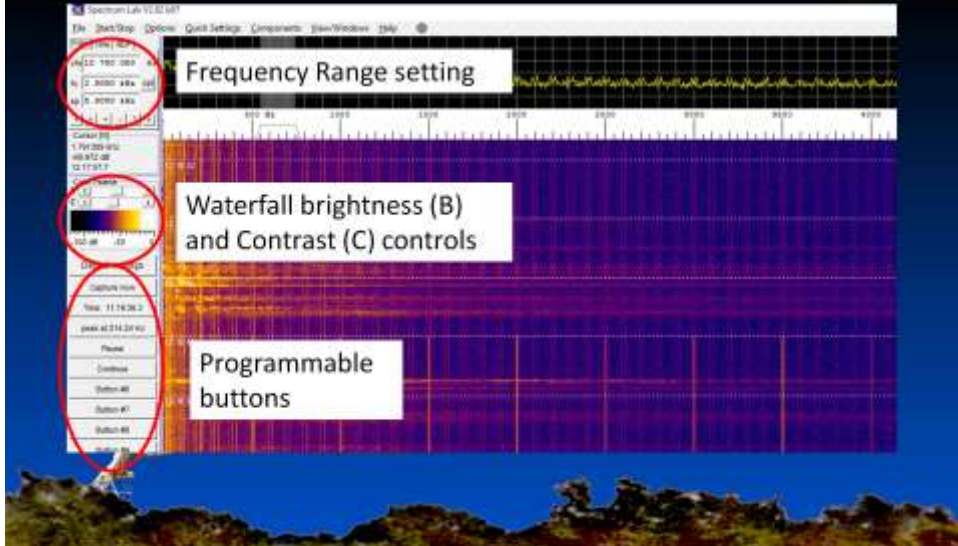
The Spectrogram screen is a scrolling bitmap showing the "history" of the last recorded spectra. As time passes the old samples will be scrolled out of view.

## Showing a time varying signal



The spectrum of a time varying signal in the Waterfall screen, produced by me whistling into the microphone.

## Default screen layout



The frequency controls at the top left set the centre frequency and width of the two screens.

The buttons underneath allow you to scroll up and down in frequency and zoom in or out.

The slider controls provide FINE adjustment of the brightness and contrast of the Waterfall screen. They do not affect the Spectrum Graph screen.

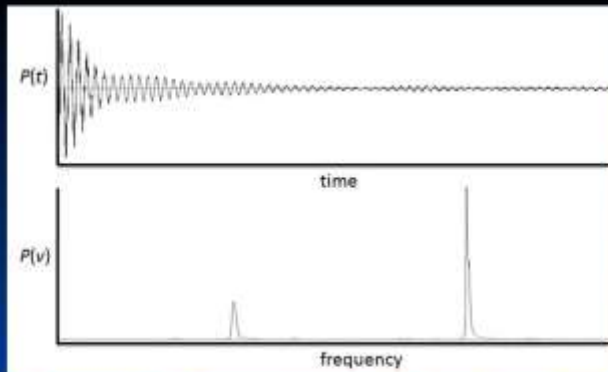
For COARSE adjustment of the brightness you need to use the Spectrum Display Settings (Part 2) found under the Options menu tab. You would usually use these coarse settings when switching between LOG and LIN display of amplitude, or changing the source, e.g. from a microphone input to a Line In input.

The Coarse settings drive the y-axis scale in the Spectrum Graph window.

The functions of the programmable buttons are set up in the Configuration File that is loaded when you start Spectrum Lab.



## 2. The Fast Fourier Transform (FFT)



Two ways  
of looking  
at the  
same  
signal

Credit: Time domain to frequency domain by Pbchem

The Fourier Transform is a mathematical algorithm but a fundamental component of modern life, for instance the Digital Signal Processing used in modern radio and television equipment, and the compression systems used in audio and image systems.

The basis is that any

The same principle applies to spatial data and is the basis of the aperture synthesis technique used in radio telescope arrays.

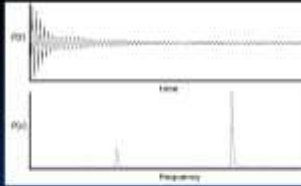
The Fast Fourier Transform is a derivation that is suited to computer processing.

The top graph shows how a signal amplitude varies with time: the bottom graph shows the how the amplitude varies with frequency.

These are two ways of looking at the same thing.



## Limitations of the FFT



These are **analogue** waveforms,  
but computers are **digital** machines  
dealing with discrete values

- The incoming analogue signal is sampled and each sample is measured/given a value in the Analogue to Digital Converter
- A block of these sample values is used in each FFT calculation
- The FFT calculation tells you how much energy is present in individual frequency 'bins'

In the analogue world the Fourier Transform works perfectly, but when it is implemented as the digital Fast Fourier Transform several issues arise which can distort the output. You need to be aware of these if you are not to be misled.

Computers only work with numbers so the incoming analogue signal has to be sampled and each sample converted to a value. If the signal is not sampled often enough this process will introduce false signals.

The FFT algorithm is then applied to a series or block of samples, typically thousands at a time. The result of the FFT calculation tells you the amount of energy in discrete frequency 'bins' of a few Hertz wide.

The more samples that are used in the FFT calculation, the narrower the bins, i.e. the frequency resolution goes up.

## Limitations of the FFT

### RESOLUTION

- Frequency resolution (width of each bin in Hz) is driven by the number of samples used in each FFT calculation
- More samples = better resolution (fewer Hz per bin)

### BUT

- It takes time to collect the extra samples
- So better frequency resolution means poorer time resolution

FFT Block size	Bin width	Window size
4,096 (4k) samples	2.92 Hertz	0.34 seconds
8,192 (8k)	1.46 Hertz	0.68 seconds
16,384 (16k)	0.73 Hertz	1.36 seconds

There is a trade-off between frequency and time resolution.

The result coming out of the FFT calculation is for the entire period of samples – the ‘Window size’.

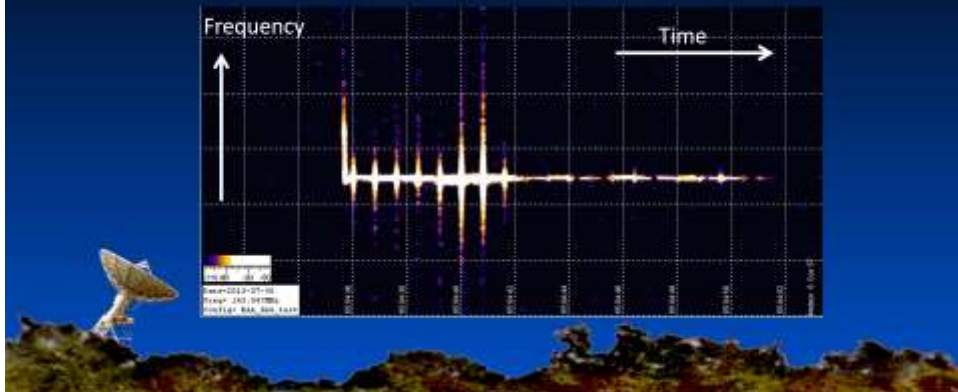
So for a 4k block of samples at a sample rate of 12 kHz, it takes a third of a second to collect enough samples for a single FFT calculation, and the spectrum that comes out of that calculation represents the whole period.

But what if the signal changes in that time?

## Limitations of the FFT

### ADDING ARTEFACTS

- The FFT calculation is only valid if the signal is unchanged during the block of samples (SL calls this the 'Window Time')
- Changes within the FFT window time introduce artefacts



Note that Spectrum Lab has now been configured to scroll horizontally rather than vertically, to maximise visibility of the echo detail.

If the signal changes within the period covered by the block of samples, the FFT calculation will 'invent' spurious frequency components – artefacts.

The GRAVES transmitted beam steps across the sky at 0.8 second intervals, illuminating different sections of the meteor trail.

So, every 0.8 seconds both the amplitude and phase of the reflected signal will change, causing the FFT process to create vertical, symmetrical spikes. The stronger the signal, the stronger the spikes.

QUESTION: Is this a problem, or is it a mechanism for synchronising observations made by multiple observers?

## Impact of the FFT on meteor counts

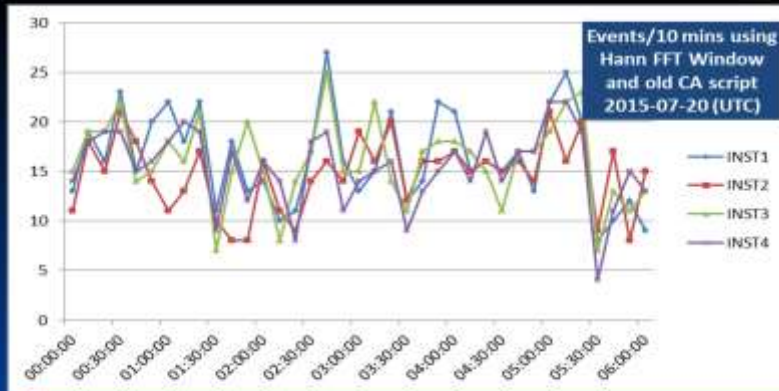


You can run multiple instances of Spectrum Lab at the same time and from the same input, limited by your PC cpu capability.

These four instances have been set up with the same configuration file, apart from having different file names for saving data.

But they do not agree on the number of meteor triggers – why not?

## Discrepancies from the same input



If we can't get our own computer to give consistent results, what's the point of comparing data with other observers?

Prolonged monitoring shows inconsistencies between identical instances of Spectrum Lab fed by the same data stream!

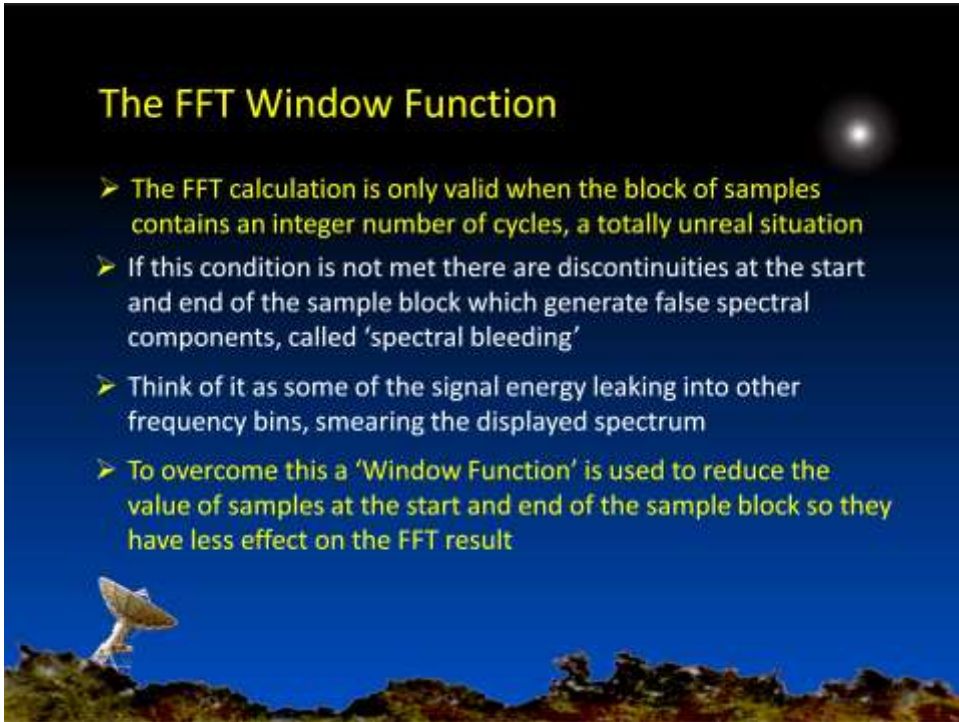
The individual graphs converge for longer periods, e.g. for counts per hour, but with identical configurations and using the same data stream they should produce identical results across all time periods.

The answer is down to the time relationship (synchronisation) between the FFT sample blocks and short meteor events.

The individual FFT calculations for the four instances are not time synchronised and the FFT 'Window' function is coming into play.

## The FFT Window Function

- The FFT calculation is only valid when the block of samples contains an integer number of cycles, a totally unreal situation
- If this condition is not met there are discontinuities at the start and end of the sample block which generate false spectral components, called 'spectral bleeding'
- Think of it as some of the signal energy leaking into other frequency bins, smearing the displayed spectrum
- To overcome this a 'Window Function' is used to reduce the value of samples at the start and end of the sample block so they have less effect on the FFT result



The FFT calculation is only valid when the block of samples contains an integer number of cycles. So it only produces the correct answer if we are looking at one specific frequency and the time duration of the FFT block is equal to a whole number of cycles.

But we are dealing with a range of frequencies so this condition can never be met for all of them.

The Window Function forces the samples at each end of the block to zero, to impose a false appearance of continuity.



## The FFT Window Function

- Spectral bleeding can be significantly reduced by forcing the samples at the beginning and end of the block to zero by multiplying the block by a Window function
- There are many types of Window Function, each having its own strengths and weaknesses, but the most commonly used is the Hann (aka Hanning) function
- But the Window Function has serious consequences for short duration signals, such as meteor head echoes. A Head Echo that happens to fall in the middle of a sample block is unaffected by the Window Function, but if it occurs towards the start or end of the block it is attenuated and will not trigger a count



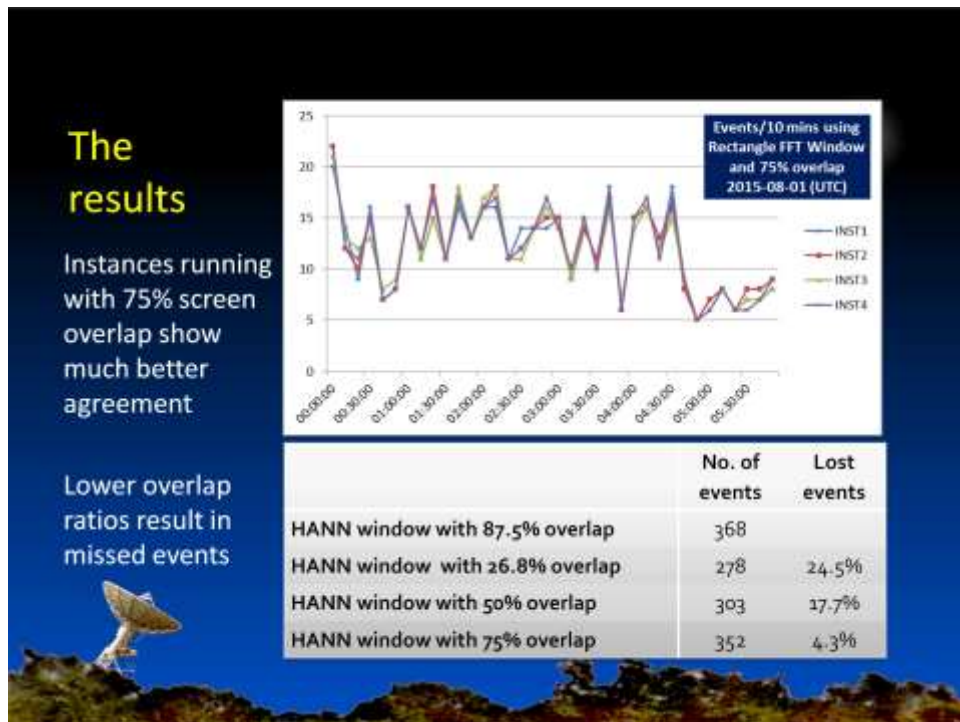


## Using Window Overlap

- If the FFT calculation is set to run every 100 milliseconds, but uses 200 milliseconds worth of samples, it will be including samples used in previous/succeeding calculations— so called Window Overlap
- With a 75% overlap any short duration event will fall within the peak sensitivity of one of the FFT calculations. But for precision amplitude measurements we should consider an 87.5% overlap.
- Using high screen overlaps is practical for fast scroll rates (revealing detail) but creates issues for screen shots covering 5 or 10 minutes.



For slow scrolling screens, achieving a high Window Overlap limits the options for frequency and time resolutions.



An instance running at 27% overlap was found to record 25% less meteor events than one running at 87.5% overlap.

This effect discriminates against short duration head echo events. Longer duration events will span multiple FFT calculations and hence trigger a count.

A 75% overlap is perhaps the best compromise between accuracy and processing requirements.

## What makes Spectrum Lab so different?

- ANSWER a) You can configure the system to suit your needs  
b) The Conditional Actions function

If	Then	Comment
always	A=noise(low,high); Sig=peak_a(low,high); h2=str("hh:mm:ss",now)	Continually monitor FFT frequency bins between the specified 'low' and 'high' frequencies and capture the current timestamp
Sig>{A+17}	C=C+1; timer0.restart(2)	If trigger condition is met, increment a duration counter and restart a 2 second 'timeout' timer
C=1	Mtime=h2:MName="Meteor"+h1:MSig=Sig	Transfer Event start time, label, and signal level to holding registers
timer0. expired(1)	fopen2("c:\\Spectrum\\"+"event_log"+ str("YYYYMMDD",now)+".txt",a): fp2[Mtime+"."+str(MSig)+",""+str(C)]	If trigger condition not met for 2 secs, save the initial start conditions and the event details to a log file

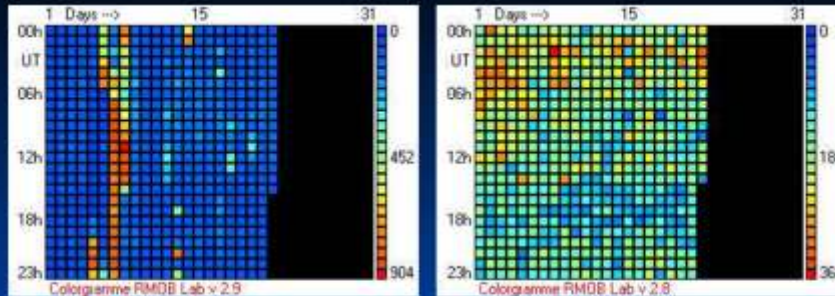
As previously noted, Spectrum Lab is configurable, so you can try different settings and signal processing techniques to improve event detection and extract more information.

The Conditional Actions (CA) uses Spectrum Lab's built in Command Interpreter function to apply tests to the internally generated data and to use the results to trigger other actions.

The Command Interpreter can also be used to configure and control Spectrum Lab from a remote application, using the built-in HTTP server, and to invoke Macros for internal operations.

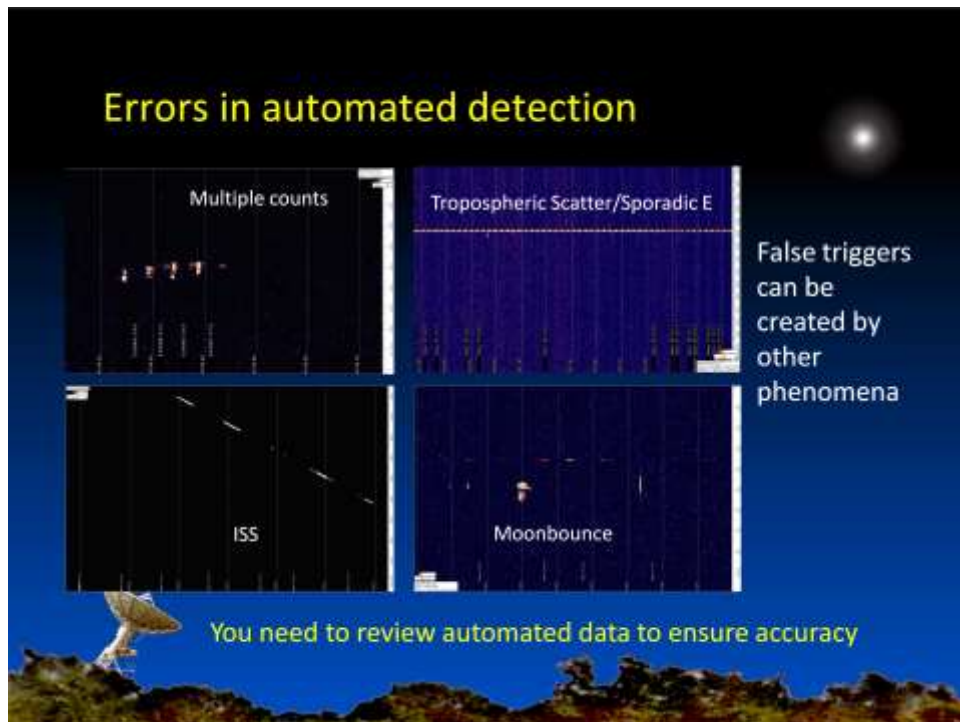
## Automated detection is problematic

Two very different reports for the same month!



Images courtesy of the Radio Meteor Observing Bulletin

An examination of the RMOB website reveals many cases of inconsistent reports. Poor quality data hinders attempts to identify subtle changes in meteor activity caused by small outbursts or by minor meteor streams.



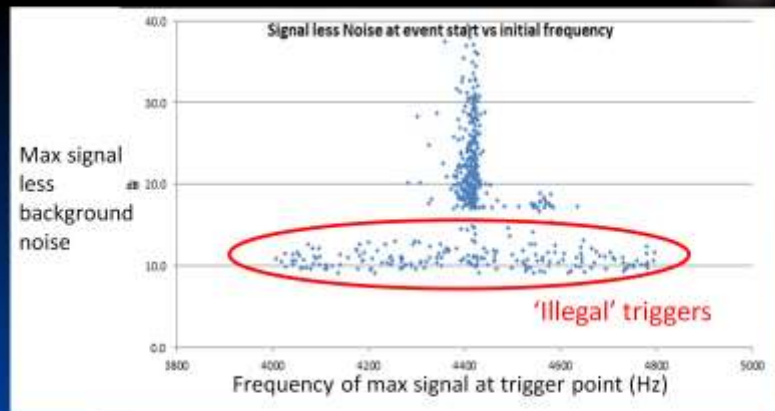
The 'Signal > Noise + 17 db' trigger criteria is simple and copes with the majority of interference spikes but there are several other phenomena that will register as false meteor echoes.

This is a significant problem for the automated detection and reporting of meteor activity and results in poor quality data.

At present the only way of removing these false triggers is to review the event log against a visual record and delete false triggers.

We need better detection algorithms!

## A problem with the standard trigger test



The trigger level is set at 17 dB – what are all these other triggers doing here?

Plotting the data saved in the .txt file can reveal interesting (alarming!) patterns.

The 'legal' triggers have 17 dB and more difference between the signal and noise levels, as defined in the CA script.

They are also clustered around the GRAVES frequency (approx. 4400 Hz for my installation)

The outliers at around 4600 Hz represent triggers from events with high Doppler shifts, so these all make sense.

BUT - there should not be any triggers recorded with signals less than 17 dB above the noise level.

These 'illegal' triggers are recorded as having a random frequency across the Low to High frequency range defined in the CA script.

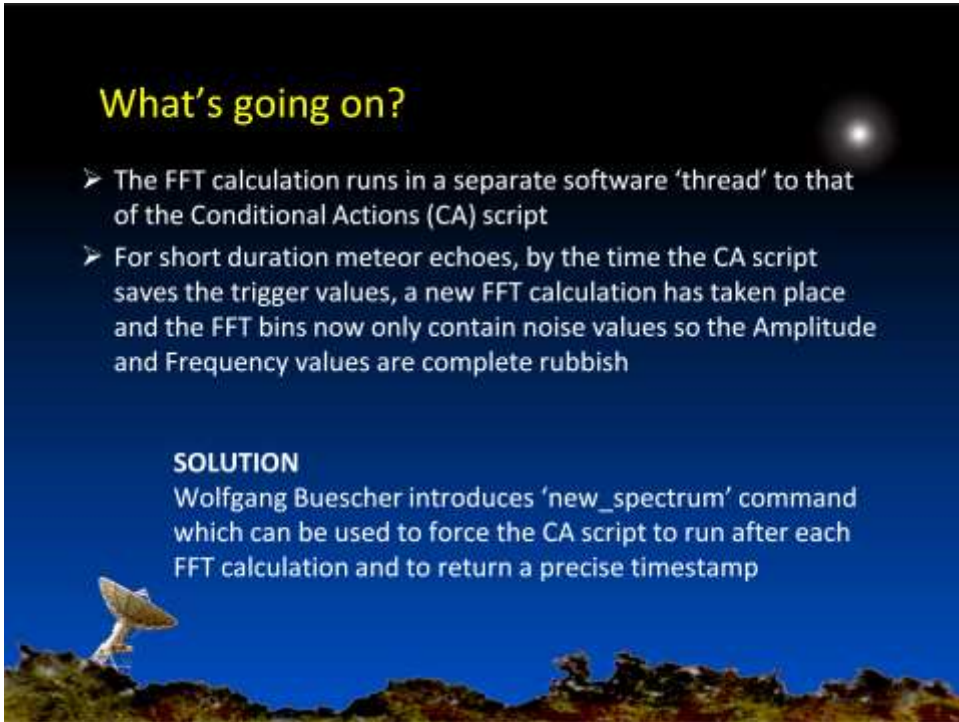


## What's going on?

- The FFT calculation runs in a separate software 'thread' to that of the Conditional Actions (CA) script
- For short duration meteor echoes, by the time the CA script saves the trigger values, a new FFT calculation has taken place and the FFT bins now only contain noise values so the Amplitude and Frequency values are complete rubbish

### SOLUTION

Wolfgang Buescher introduces 'new\_spectrum' command which can be used to force the CA script to run after each FFT calculation and to return a precise timestamp



This problem is caused by a lack of synchronisation between the FFT calculation and the Conditional Actions script.

This affects short duration events lasting less than the FFT Window time. Longer events mean that consecutive FFT calculations return similar spectra so the bin values are still representative when the CA script saves them.

This partly explains why I could not compare Horizontal and Vertical Polarisation measurements for individual meteor events.



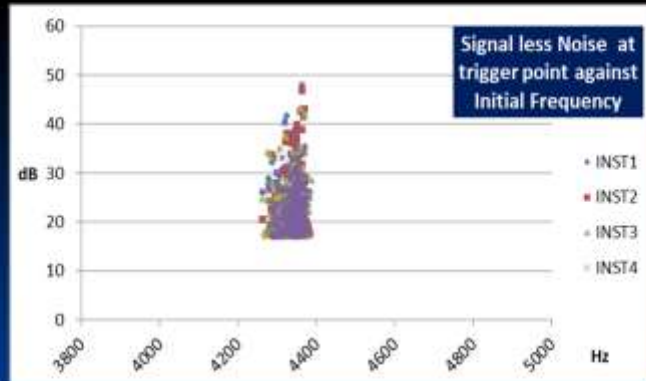
## Revised CA using new\_spectrum command

If	Then	Comment
new_spectrum	A=noise(Low,High): Sig=peak_a(Low,High):Trig=(A+17)	Use new command to force CA script to run after each FFT calculation
Sig>Trig	then inc(MCount):t1=time:	New event – use 'time' to record FFT time rather than 'now'
continuation	Event=MCount: TrigTime=str("hh:mm:ss",time): TrigSig=Sig:TrigNoise=A: TrigFreq=peak_f(Low,High)	Transfer trigger values to 'holding' registers
Sig<=Trig	then t2=(time-t1)	Signal falls below trigger point so calculate duration

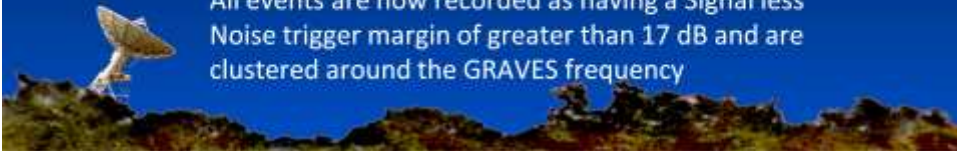
'new\_spectrum' ensures correct Signal and Noise levels are saved to file, whilst using 'time' rather than 'now' saves the precise timestamp of the FFT calculation

A further advantage of this script is that it reports the duration of the event as seconds. The previous method measured the duration in terms of a loop counter, the loop time of which will be dependent upon the hardware (cpu) specification of the host PC.

## Results using revised CA script



All events are now recorded as having a Signal less Noise trigger margin of greater than 17 dB and are clustered around the GRAVES frequency



## Outputs from Spectrum Lab

### 1. The Event Log

Format and destination folder defined in Conditional Actions script

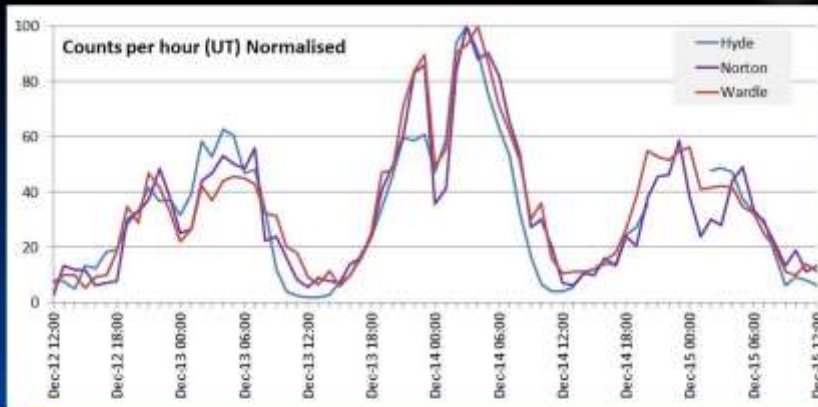
If	Then
timero. expired(x)	fopen2("c:\\Spectrum\\"+"event_log"+str("YYYYMMDD",now)+".txt",a); fp2(Mtime+", "+str(E)+", "+str(MSig)+", "+str(MNoise)+", "+str(MFreq)+", +str(C)); fclose2

Time(UT)	Event#	Sig(dB)	Noise(dB)	Freq(Hz)	Durn(secs)
15:08:12	1	-65.5	-87.6	3409	2.39
15:08:19	2	-69.5	-86.6	3309	0.34
15:08:24	3	-69.9	-87.2	3500	0.34
15:08:26	4	-68.7	-85.8	3307	0.34
15:08:57	5	-69.1	-86.8	3500	0.34

Once an event has finished (trigger test has failed for more than 2 seconds) Spectrum Lab opens the event log and adds an extra record listing a timestamp and event number, plus whatever other parameters are defined in the Conditional Actions script.

The event log can be opened using Notepad or similar, or imported into a spreadsheet for further analysis.

## Outputs from Spectrum Lab



Excel chart uses event log data to compare observer reports for 2014 Geminids

An example of how the Event Logs from 3 observers have been imported into a spreadsheet for comparison.

A moment of revelation was experienced when looking at the Planisphere for 23:00 on the 13<sup>th</sup> to 02:00 on the 14<sup>th</sup> December!

(Hint1: GRAVES is South East of the observers)

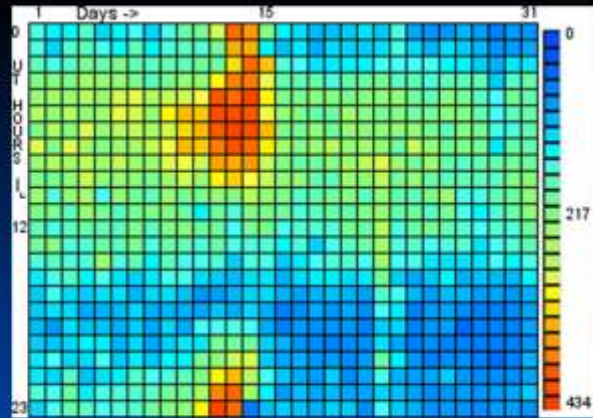
(Hint2: What shower is this?)

## Analysing meteor data

... or use it to  
chart activity over  
longer durations

December 2011

Clearly shows  
pattern of  
Sporadic and  
Shower activity



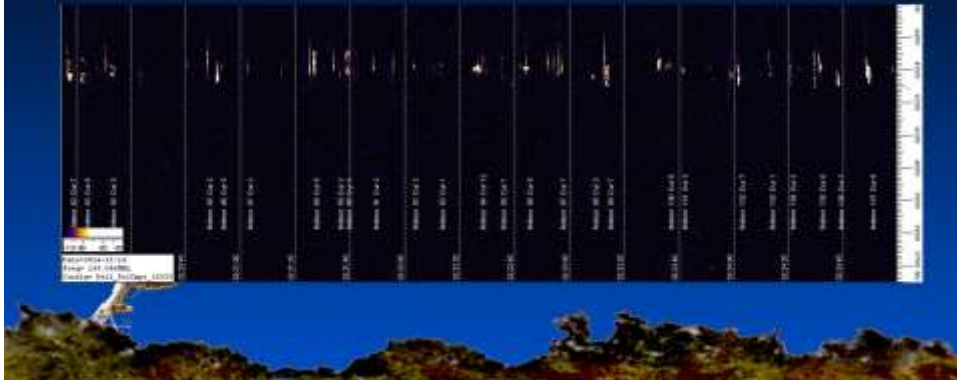
The data from Spectrum Lab can also be used to derive activity charts for longer periods of time.

## Outputs from Spectrum Lab

### 2. Screen shots

Conditional Actions script defines when a screen shot is captured

- Every 5 or 10 minutes for later reconciliation against event log
- After a significant event, e.g Duration longer than 0.5 seconds



You can automate the saving of screen shots using commands in the Conditional Actions script.

As an example, using a slow scroll rate and saving a screen every 5 minutes provides a complete visual record for checking for invalid events on the event log file, increasing the accuracy and value of the data.

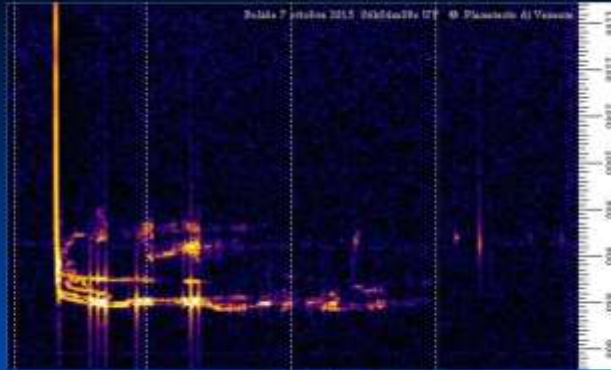
## Outputs from Spectrum Lab

### 2. Screen shots

... or perhaps after a significant event, e.g Duration > 0.5 seconds

Meteor event of  
2105 Oct 7

Observed by  
Enrico Stomeo  
Venice  
Planetarium



Alternatively you can run a second instance of Spectrum Lab with a fast scroll rate and a Conditional Actions script that triggers a screen capture for longer events, to capture more detail on individual events.

**REMEMBER:** If you do run multiple instances of Spectrum Lab they should save event logs and screenshots with different names!



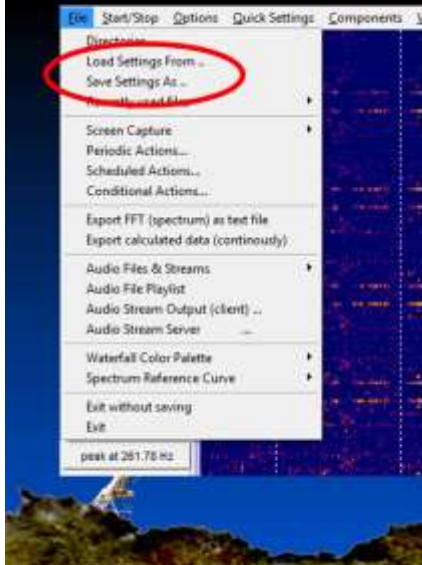
## Outputs from Spectrum Lab

### 3. SAVING .wav FILES

- Spectrum Lab can be configured to automatically save audio recordings of meteor 'pings'
- SL incorporates a large input data buffer so you can capture a recording from 3 or 4 seconds before the event
- Analysing meteor pings in the time domain (e.g. using Audacity or similar) is essential for some applications
- Audio files are very large and need discipline in archiving old files or the Hard Drive will fill quickly



## 4. Setting up Spectrum Lab

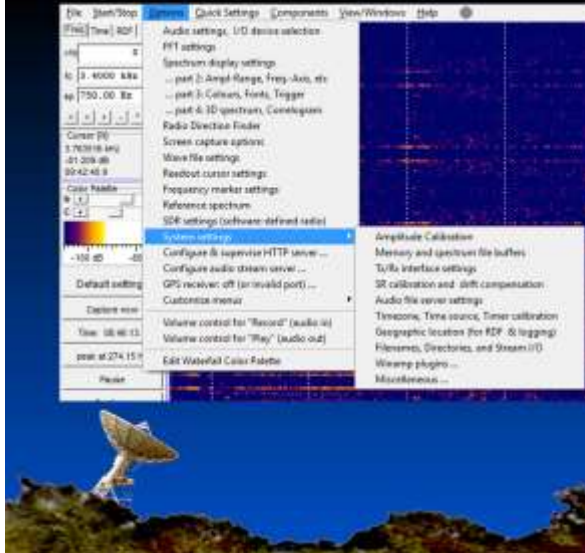


### Spectrum Lab Config file

- Configures Spectrum Lab to a defined starting state
  - Screen sizes and layouts
  - Frequency settings
  - Spectrum analyser settings
  - Conditional Actions
- Any changes are saved to a temporary file when the application is closed and used when restarted
- Keeps track of different configurations for instances running at the same time

Most of the parameters controlling the operation of Spectrum Lab are loaded from a Configuration (User .usr) file when the application starts. Changes to this are saved when you close the application unless you use the 'Exit without saving option'.

## The exception to the rule



### System Settings

➤ Specific to your PC so not part of the Config file

- Timezone settings
- Filenames, Directories

## 5. Further information

BAA RAG website	<a href="http://www.britastro.org/radio/">www.britastro.org/radio/</a>
BAA RAG Coordinator e-mail	<a href="mailto:radiogroup@britastro.org">radiogroup@britastro.org</a>
International Meteor Organisation	<a href="http://imo.net/radio">imo.net/radio</a>
Spectrum Lab download	<a href="http://www.qsl.net/dl4yhf/spectra1.html">www.qsl.net/dl4yhf/spectra1.html</a>
Radio Meteor Observing Bulletin	<a href="http://www.rmob.org/index.php">www.rmob.org/index.php</a>

