SDHDF spectral line processing

George Hobbs
V0.9: Dec 2019

NOTES:

- We need to take more data with only 1 or 2 pol. All testing so far for 4 pol!
- Must properly write out updated metadata information after modifying the file.
- Must fix centre frequency after zapping
- WORKING ON SAVING DATA FILES AFTER BASELINE REMOVAL **
- Should optimise the code -o when compiling
- Can I use “chunking” when reading the data?

General comments

The spectral line observing modes and software are still in development.

See the time-line of changes available from https://www.parkes.atnf.csiro.au/observing/History_Log.html for UWL observation updates. Key updates for the spectral line community are:

- **2018-10-30** Before this date: Band1 frequency range low->high; Band2 and band3 freq range high->low
- **2018-12-17** ~23:45 changes to Medusa chbw from -ve to +ve ; Medusa sub-band data freq, now range from low->high, Source J1644-4559 test
- **2019-02-11** Medusa updates, fixed negative bandwidth output
- **2019-03-28** Over the last couple of days it has come to light that the cal signal for spectral line observations made using the TOS system has not been properly synchronised with the medusa data. Recent testing has shown there was significant merging between the on and off labelled spectra (for example for some recent data the on-off cal spectra were about a factor of 10 smaller than they should have been). Over the course of the last couple of days the issue has been investigated and fixed, so all data from now (28/03) on should be fine. For data taken prior to today with a cal controlled by TOS, the absolute flux scaling will be in error, although we have yet to fully quantify the factor (potentially of the order of a few % TBD). We would also be concerned with the calibration of polarisation data taken prior to today"
- **2019-03-31** POL B missing from the UWL-Mid band. P977 observations from UTC_START=2019-03-30-17:33:51 onwards are affected till Sunday midday; FIX: restart FPGA data stream
- **2019-06-06** Medusa updated "Rather than discarding the final sub integration in Continuum mode, Medusa will now write out the partial sub-integration in the final file."
- **2019-07-16** Medusa UWL preprocessor modified to support production of calibration spectra with higher than 0.125 MHz resolution, with TOS templates and UI updated to implement Tsys calibration spectra range 0.001953125 - 1 MHz.
- **2019-07-18** 10dB pads fitted to all 6 UWL outputs to the Parkes Conversion System (PCS), translator cables 36, 37, 38, 39, 40, 41. This is to address Alex Dunning's theory that reflections from the PCS are causing ripple on the signal to the other full signal path

Medusa, Euryale and the SDHDF format

The Parkes UWL system is a flexible system for spectral line and continuum observations. The receiver and signal processor system divides the band into 26 subbands as described in Hobbs et al. (2019) and summarised in the table below.

Sub-band	RF band	Freq. range (MHz)	Central Freq. (MHz)	Median $T_{\rm sys}$ (K)	Median $S_{\rm sys}$ (Jy)	$\epsilon_{\rm ap}^{\rm sim}$	ϵ_B	FWHM (deg)	Band Use (%)
0	1	704–832	768	39	49	0.70	0.92	0.43	31
1	1	832–960	896	31	45	0.69	0.95	0.38	24
2	1	960–1088	1024	22	38	0.66	0.96	0.35	65
3	1	1088–1216	1152	22	38	0.60	0.97	0.32	58
4	1	1216–1344	1280	22	38	0.64	0.96	0.27	68
5	2	1344–1472	1408	23	34	0.65	0.95	0.25	91
6	2	1472–1600	1536	23	36	0.67	0.95	0.23	79
7	2	1600–1728	1664	22	33	0.67	0.95	0.20	85
8	2	1728–1856	1792	22	34	0.66	0.97	0.20	67
9	2	1856–1984	1920	22	36	0.67	0.94	0.18	75
10	2	1984–2112	2048	22	36	0.65	0.96	0.18	92
11	2	2112–2240	2176	19	36	0.64	0.95	0.17	76
12	2	2240–2368	2304	21	39	0.62	0.96	0.18	62
13	3	2368–2496	2432	21	39	0.62	0.96	0.16	27
14	3	2496–2624	2560	21	39	0.61	0.95	0.16	92
15	3	2624–2752	2688	21	41	0.65	0.96	0.15	84
16	3	2752–2880	2816	20	39	0.64	0.96	0.14	92
17	3	2880–3008	2944	20	39	0.62	0.96	0.13	92
18	3	3008–3136	3072	21	39	0.64	0.95	0.13	91
19	3	3136–3264	3200	20	43*	0.64	0.97	0.14	92
20	3	3264–3392	3328	20	45*	0.62	0.96	0.13	92
21	3	3392–3520	3456	21	44*	0.63	0.96	0.10	84
22	3	3520–3648	3584	23	48*	0.63	0.96	0.12	79
23	3	3648–3776	3712	23	53*	0.61	0.96	0.10	92
24	3	3776–3904	3840	25	70*	0.60	0.96	0.11	91
25	3	3904–4032	3968	27	72*	0.59	0.95	0.11	78

The data are channelised in the Medusa GPU processing system with up to 2^{21} = 2097152 channels per sub-band (equivalent to a frequency resolution of 61 Hz) forming one, two or four polarisation products:

- 1 product: AA*+BB* (pseudo-Stokes I)
- 2 products: AA*, BB*
- 4 products: AA*, BB*, Real(A*B), Imag(A*B)

The recorded spectra are written to disk (a "spectral dump") with a sampling interval between 0.25 and 60 seconds. The Euryale data staging server receives these spectral dumps, obtains further information about the observation such as telescope pointing positions and produces a Single Dish Hierarchical Data Format (SDHDF) data file for each observation. If available, calibration spectra are also stored in each output data file. Usually the calibration signal is switching throughout the observation. In such cases, an "ON CAL" spectrum and an "OFF CAL" spectrum are written into the SDHDF files with typically 1MHz channel resolution (and 4 polarisation products). Zoom bands are also supported enabling bands narrower than the native UWL 128-MHz subbands.

One restriction of the UWL system is that the digitiser sampling frequencies are fixed. It is therefore **not possible** to calibrate the bandpass by frequency switching.

Note that the spectral dump time (the integration time for each spectrum in the data file) depends upon the number of observed channels. Different sub-bands may have different number of channels and therefore the spectral dump times (and hence total integration time) may be different between sub-bands. For example, in one observation the problem here is the different channelisation for each of the sub-bands. Medusa was asked to produce 1.0 second sampling intervals, but due to the requested channelisations, it was not possible to produce exactly 1.0 second samples.

Medusa instead produced 0.98304s samples for sub-bands 00, 05 and 07 and 0.999424s samples for sub-bands 19 and 20.

Telescope positional information is obtained after the completion of the observation by querying a database. The database now updates all parameters (including RA, DEC, Az, El, Wind speed, etc.) once per second, but previously some parameters were updated with a significantly lower cadence. Software on Euryale interpolates the database information to correspond to the centre of each spectral dump.

Where is the code?

The code described in this document is written to provide a “stand-alone” package, but we note that SDHDF is relatively easy to read and process in many software languages including python. The code is currently being developed by George Hobbs and is available on his desktop computer (brahe) from /u/hob044/software/new_c/sdhdfProc. This code will be checked-in to a formal repository when basic testing has been completed. Note that this is currently a “Beta” version and the code has not been well tested. The code is written in C and relies on the hdf5 library available from https://www.hdfgroup.org/downloads/hdf5/.

Note that other code exists for viewing and manipulating HDF-format files:

1. **silx view <filename>** -- views the data file. Note that the frequency column is not written out with sufficient decimal places in silx. The actual data are fine.
2. **h5ls <filename>** -- lists the content of a data file
3. **h5dump <filename>** -- dumps the content of an HDF file.

SDHDF software

Identifying/inspecting SDHDF files		
sdhdf_describe	Show the primary metadata for a list of files (including project IDs, source names, observer, etc.)	v1.9 available
sdhdf_quickdump	Produce text files containing information within an SDHDF file	v1.9 available
sdhdf_identify	Identify a list of files with specified parameters (e.g., source names, project codes)	v1.9 available
Plotting/visualising SDHDF files		
sdhdf_plotSpectrum	Interactive visualisation of the spectra for a single band in a single SDHDF file.	v1.9 available
sdhdf_plotMultiSpec	Non-interactive plotting of spectra for a single band in one or more data files.	v1.9 available
sdhdf_plotWide	Interactive visualisation of the entire band (multiple sub-bands).	v1.9 available
sdhdf_plotScan	Plot of the signal strength as a function of	v1.9 available

	time-dependent parameters including pointing directions.	
sdhdf_plotMap	Interactive plot of a sky map built up from multiple scans	
Manipulating SDHDF files		
sdhdf_extract	Extract parts of an SDHDF file to produce a new (smaller) file (e.g., specific zoom bands)	v1.9 available
sdhdf_modify	Produce a new SDHDF file by modifying the contents of an existing file (i.e., averaging in time or frequency)	v1.9 available
sdhdf_sum	Produce a new SDHDF file by summing the spectral dumps in a set of existing files.	v1.9 available
Flagging		
sdhdf_flag	Interactive flagging of a data file	v1.9 available
sdhdf_autoFlag	Automatic flagging of a data file (including copying of flagged data from a different file)	v1.9 available
Conversion tools		
sdhdf_convertFrom	Convert to SDHDF-format from other formats (currently just text format)	
sdhdf_convertTo	Convert from SDHDF-format to other formats (currently just SDFITS)	
Processing tools		
sdhdf_onoff	Determine (ON-OFF)/OFF spectra from an ON-source file and an OFF-source file	v1.9 available
sdhdf_map	Produce a sky map from multiple input scans	
sdhdf_baseline	Determine (and remove) baselines from data files	
sdhdf_pointing	Produce pointing solutions using multiple scans	v1.9 only
sdhdf_gainCurve		v1.9 only

Calibration		
sdhdf_cal		
sdhdf_calProc	Processes the switching cal information to determine Tsys, differential gain and differential phase	****
Miscellaneous		
sdhdf_listLines	List the rest frequencies of spectral lines in a given frequency range.	v1.9 available

Bugs and feature requests

For feature requests/bug reports please leave a comment in this google doc or email george.hobbs@csiro.au.

Currently we have the following essential changes/bugs:

Code	**Issue**
sdhdf_baseline	Cannot save after removing baseline, cannot select subband
sdhdf_modify	Doesn't account for flags when frequency averaging
sdhdf_scan	ISSUE WITH POSITIONS(?) for uwl_200120_085636.hdf

And the following feature requests for future versions of the code:

Code	**Feature request**
quickdump	Not properly dealing with flags
identify	Find closest in position and/or time
modify	Fix metadata information (e.g., source names)
identify	Enable exact matching on searches

The following issues relate to the HDF5 v.1.9 format:

Request	Importance
5D arrays -> 3D - no beam and no bin	low

Inspecting and visualising spectral data

To demonstrate how to read metadata and visualise the spectra we have used the observation files that are available in /DATA/BRAHE_2/hob044/uwlData/v1.8/.

sdhdf_describe

We can first inspect the primary meta-data in the files (i.e., source names, chosen frequency bands etc.) using:

```
brahe% cd /data/BRAHE_2/hob044/uwlData/cenA
brahe% /u/hob044/software/new_c/sdhdfProc/sdhdf_describe uwl_*.hdf
```

Produces output like:

```
-----------------------------------------------------------------------------------------------------------------------------------------------------
File                         UTC                    SDHDF PID    Source               Tel        Observer   RCVR     Bands POL_TYPE   CAL   Time
-----------------------------------------------------------------------------------------------------------------------------------------------------
uwl_190604_090701.hdf        2019-06-04T09:07:01Z   1.8   P1017  3c273_cal_R          Parkes     CA         UWL      26    AABBCRCI   ON    114.00
uwl_190604_090821.hdf        2019-06-04T09:08:21Z   1.8   P1017  3c273_cal_R          Parkes     CA         UWL      26    AABBCRCI   ON    95.00
uwl_190604_090945.hdf        2019-06-04T09:09:45Z   1.8   P1017  3c273_cal_R          Parkes     CA         UWL      26    AABBCRCI   ON    114.00
uwl_190604_091118.hdf        2019-06-04T09:11:18Z   1.8   P1017  3c273_cal2_R         Parkes     CA         UWL      26    AABBCRCI   ON    133.00
uwl_190604_091921.hdf        2019-06-04T09:19:21Z   1.8   P1017  cenA_ra1_100         Parkes     CA         UWL      26    AABBCRCI   ON    437.00
uwl_190604_092328.hdf        2019-06-04T09:23:28Z   1.8   P1017  cenA_ra1_101         Parkes     CA         UWL      26    AABBCRCI   ON    437.00
uwl_190604_092736.hdf        2019-06-04T09:27:36Z   1.8   P1017  cenA_ra1_102         Parkes     CA         UWL      26    AABBCRCI   ON    437.00
```

If you do not wish to have the header banner then use the **-nohead** command line argument.

The columns are:

- Filename
- UTC of observation start
- SDHDF version number
- Project ID
- Observation ID (for Parkes observations this is the Dhagu scheduling block ID)
- Source name
- Telescope
- Observer
- Receiver
- Number of frequency bands
- Polarisation type
- Whether the calibrator was switching (ON) or OFF.
- Total integration time (seconds)

To obtain band information for a particular file (or set of files) use:

```
brahe% /u/hob044/software/new_c/sdhdfProc/sdhdf_describe -showb uwl_190604_090821.hdf
```

This gives:

```
--------------------------------------------------------------------------------------------------------------------------------------------
File                          UTC                   SDHDF PID    Source                Tel        Observer   RCVR    Bands POL_TYPE   CAL   Time
--------------------------------------------------------------------------------------------------------------------------------------------
uwl_190604_090821.hdf         2019-06-04T09:08:21Z   1.8   P1017  3c273_cal_R          Parkes     CA         UWL     26    AABBCRCI   ON    95.00
-------------------------------------------------------------------
       #   BandID      F0        F1       NCHAN    TDUMP NPOL  NDUMP
-------------------------------------------------------------------
 [Band] 000 band_SB00    704.00    832.00 4096    0.523 4     95
 [Band] 001 band_SB01    832.00    960.00 4096    0.523 4     95
 [Band] 002 band_SB02    960.00   1088.00 4096    0.523 4     95
 [Band] 003 band_SB03   1088.00   1216.00 4096    0.523 4     95
 [Band] 004 band_SB04   1216.00   1344.00 4096    0.523 4     95
 [Band] 005 band_SB05   1344.00   1472.00 4096    0.523 4     95
 [Band] 006 band_SB06   1472.00   1600.00 4096    0.523 4     95
 [Band] 007 band_SB07   1600.00   1728.00 4096    0.523 4     95
 [Band] 008 band_SB08   1728.00   1856.00 4096    0.523 4     95
 [Band] 009 band_SB09   1856.00   1984.00 4096    0.523 4     95
 [Band] 010 band_SB10   1984.00   2112.00 4096    0.523 4     95
 [Band] 011 band_SB11   2112.00   2240.00 4096    0.523 4     95
 [Band] 012 band_SB12   2240.00   2368.00 4096    0.523 4     95
 [Band] 013 band_SB13   2368.00   2496.00 4096    0.523 4     95
 [Band] 014 band_SB14   2496.00   2624.00 4096    0.523 4     95
 [Band] 015 band_SB15   2624.00   2752.00 4096    0.523 4     95
 [Band] 016 band_SB16   2752.00   2880.00 4096    0.523 4     95
 [Band] 017 band_SB17   2880.00   3008.00 4096    0.523 4     95
 [Band] 018 band_SB18   3008.00   3136.00 4096    0.523 4     95
 [Band] 019 band_SB19   3136.00   3264.00 4096    0.523 4     95
 [Band] 020 band_SB20   3264.00   3392.00 4096    0.523 4     95
 [Band] 021 band_SB21   3392.00   3520.00 4096    0.523 4     95
 [Band] 022 band_SB22   3520.00   3648.00 4096    0.523 4     95
 [Band] 023 band_SB23   3648.00   3776.00 4096    0.523 4     95
 [Band] 024 band_SB24   3776.00   3904.00 4096    0.523 4     95
 [Band] 025 band_SB25   3904.00   4032.00 4096    0.523 4     95
```

The band information is:

- Band identifier (starting from 0)
- Band label
- Lowest frequency in band (MHz)
- Highest frequency in band (MHz)
- Number of channels
- Spectral dump time (seconds)
- Number of polarisations
- Number of spectral dumps in the file

Information on the calibrator source can be obtained using -showc. This has the same output form as the sub-band information given above, e.g.:

```
uwl_190606_115350.hdf.T.lsr  2019-06-06T11:53:50Z   1.8   P1017  cenA_dec1_122        Parkes     CA         UWL     26    AABBCRCI   ON    589.00
 [CAL] 000 band_SB00   3904.00  4032.00 128   10.000 4     30
 [CAL] 001 band_SB01   3904.00  4032.00 128   10.000 4     30
 [CAL] 002 band_SB02   3904.00  4032.00 128   10.000 4     30
 [CAL] 003 band_SB03   3904.00  4032.00 128   10.000 4     30
```

It is also possible to obtain information about the observation pointing and time information for each spectral dump:

```
brahe% /u/hob044/software/new_c/sdhdfProc/sdhdf_describe -showd uwl_190604_090821.hdf
```

```
--------------------------------------------------------------------------------------------------------------------------------------------
File                          UTC                   SDHDF PID    Source                Tel        Observer   RCVR    Bands POL_TYPE   CAL   Time
--------------------------------------------------------------------------------------------------------------------------------------------
uwl_190604_090821.hdf         2019-06-04T09:08:21Z   1.8   P1017  3c273_cal_R          Parkes     CA         UWL     26    AABBCRCI   ON    95.00
 [SDUMP]      0.000000 58638.3807990 09:08:17.00 19:08:17.90  12:33:06.86 +02:03:08.60  18.191  53.624 292.231  64.543
 [SDUMP]      0.523000 58638.3808050 09:08:19.62 19:08:20.52  12:33:06.86 +02:03:08.60  18.180  53.627 292.231  64.543
 [SDUMP]      1.047000 58638.3808110 09:08:22.00 19:08:22.90  12:33:06.86 +02:03:08.60  18.171  53.629 292.231  64.543
```

```
[SDUMP]     1.570000 58638.3808170 09:08:22.00 19:08:22.90  12:33:06.86 +02:03:08.60  18.171  53.629 292.231  64.543
[SDUMP]     2.093000 58638.3808230 09:08:22.00 19:08:22.90  12:33:06.86 +02:03:08.60  18.171  53.629 292.231  64.543
[SDUMP]     2.616000 58638.3808290 09:08:22.00 19:08:22.90  12:33:06.86 +02:03:08.60  18.171  53.629 292.231  64.543
[SDUMP]     3.140000 58638.3808350 09:08:22.00 19:08:22.90  12:33:06.86 +02:03:08.60  18.168  53.630 292.231  64.543
[SDUMP]     3.663000 58638.3808410 09:08:22.00 19:08:22.90  12:33:06.86 +02:03:08.60  18.158  53.632 292.231  64.543
[SDUMP]     4.186000 58638.3808470 09:08:22.00 19:08:22.90  12:33:06.86 +02:03:08.60  18.152  53.634 292.231  64.543
```

Here the output format is:

- Elapsed time from observation start (seconds)
- MJD of spectral dump start time
- UTC of spectral dump start time
- AEST of spectral dump start (hh:mm:ss.s)
- Right ascension (hh:mm:ss.ss)
- Declination (dd:mm:ss.ss)
- Azimith (degrees)
- Elevation (degrees)
- Galactic longitude (degrees)
- Galactic latitude (degrees)

The user can determine the processing commands that have been applied to a given file using the **-history** option:

```
brahe-208% /u/hob044/software/new_c/sdhdfProc/sdhdf_describe -history uwl_191210_122529.hdf
2019-12-20-13:54:29 | sdhdf_writer | Write SDHDF format file
2019-12-20-13:54:29 | metadata_que | Query the observational
```

The last line gives the date, command and description of the command for every entry in the history table.

Software versions and descriptions used in producing the data file can be inspected using the -software command-line argument.

sdhdf_quickdump

To extract a spectrum from an SDHDF file use sdhdf_quickdump. For instance:

```
/u/hob044/software/new_c/sdhdfProc/sdhdf_quickdump -f uwl_190607_130055.hdf -sb 1 -sd 2
```

This command will read the file defined by “-f” and output the spectrum for sub-band 1 (defined by -sb) and the second spectral dump (defined by -sd). Note that the -sb argument can also be the name of the sub-band (e.g., -sb SB05). The output is similar to:

```
1 2 0 832.00000 2.63038e+06 3.02798e+06 111055 164287 0 0 0 0
1 2 1 832.03125 2.50802e+06 2.72595e+06 -9553.11 -25908 0 0 0 0
1 2 2 832.06250 2.49015e+06 2.70744e+06 12964.4 -47866.1 0 0 0 0
```

```
1 2 3 832.09375 2.42107e+06 2.72846e+06 25337.7 -30644 0 0 0 0
….
```

The columns are:

1. Subband number
2. Spectral dump number
3. Channel number
4. Channel topocentric frequency (MHz)
5. Pol 1
6. Pol 2
7. Pol 3
8. Pol 4
9. Flag for pol 1 (0 = not flagged, 1 = flagged)
10. Flag for pol 2
11. Flag for pol 3
12. Flag for pol 4

Other command line arguments include:

- -c1 first spectral channel
- -c2 last spectral channel
- -sum output the summation of the spectrum (between c1 and c2, if set)

sdhdf_identify

This code allows the user the identify SDHDF files in which a particular source was observed, or the observation covered a specific sky region:

% /u/hob044/software/new_c/sdhdfProc/sdhdf_identify -src cenA uwl*.hdf

will list the filenames containing the source “*cenA*”:

```
[SRC MATCH] uwl_190604_091921.hdf cenA_ra1_100
[SRC MATCH] uwl_190604_092328.hdf cenA_ra1_101
[SRC MATCH] uwl_190604_092736.hdf cenA_ra1_102
[SRC MATCH] uwl_190604_093144.hdf cenA_ra1_103
…
```

Use -noinfo if you only wish to obtain the filenames:

```
uwl_190604_091921.hdf
uwl_190604_092328.hdf
uwl_190604_092736.hdf
uwl_190604_093144.hdf
```

To identify all the files in which a pointing was within ‘D’ degrees of a specified RA/Dec coordinate (RA,DEC) then use

% /u/hob044/software/new_c/sdhdfProc/sdhdf_identify -coord <RA> <DEC> <D> uwl*.hdf

For instance:

% /u/hob044/software/new_c/sdhdfProc/sdhdf_identify -coord 198.068 -44.8243 0.1 uwl*.hdf

gives:

```
[DIST MATCH] uwl_190604_105832.hdf 318 0.0600269
[DIST MATCH] uwl_190604_105832.hdf 319 0.098256
[DIST MATCH] uwl_190604_110240.hdf 120 0.0882875
[DIST MATCH] uwl_190604_110240.hdf 121 0.0442946
```

The third column gives the spectral dump number corresponding to the angular offset (in degrees) given in the fourth column.

sdhdf_plotSpectrum

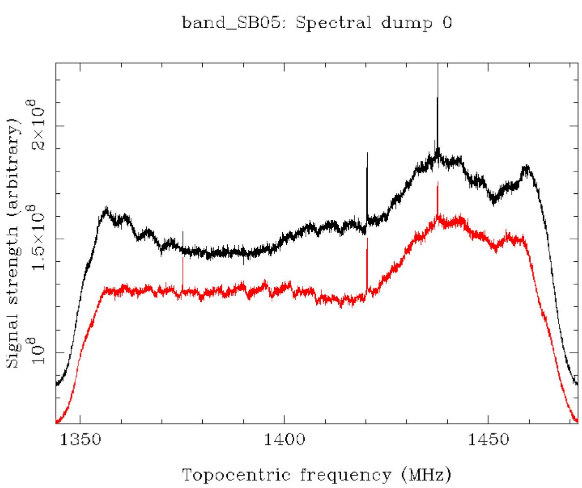
To make a simple spectral plot:

brahe-109% `/u/hob044/software/new_c/sdhdfProc/sdhdf_plotSpectrum -sb 5 -f uwl_190604_090821.hdf`

This produces a PGPLOT window like (black = pol A and red = pol B) (note that the default PGPLOT window has inverted-colours - to produce a white background use:

```
setenv PGPLOT_FOREGROUND black
setenv PGPLOT_BACKGROUND white
```

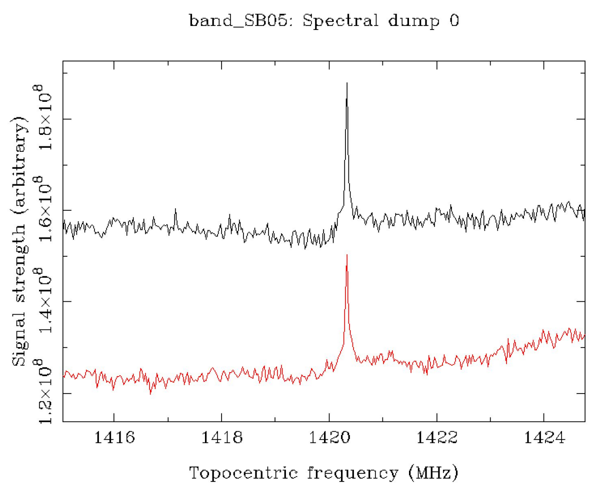
before running the plotSpectrum command)



Key-strokes include:
l = toggle plotting on logarithmic y-axis
m = toggle drawing spectral line rest frequencies
o = output spectrum (in the current zoom region) as a text file on the local disk
q = quit
z = use mouse clicks to define a zoom window - move the mouse to the bottom-left corner of the zoom window, then press ‘z’, then let go and move the mouse to the top-right corner and then click the mouse button.
u = unzoom
+ = increase spectral dump (sub-integration) number
- = decrease spectral dump (sub-integration) number
> = increase sub-band number

< = decrease sub-band number

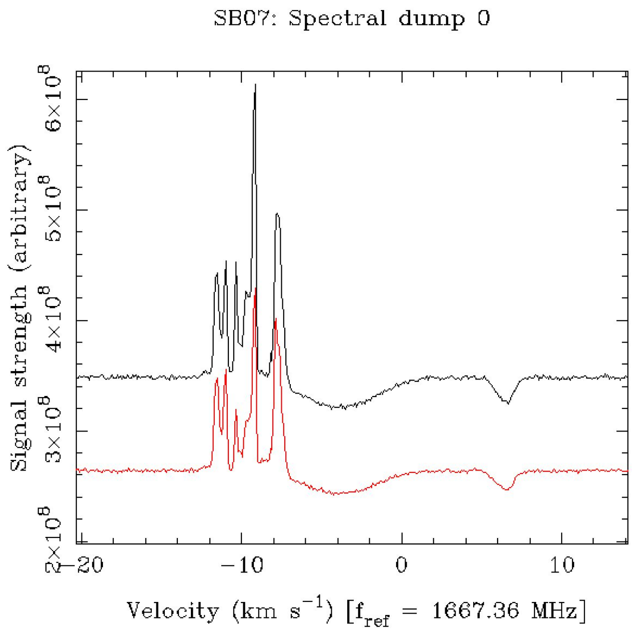
Zooming in around the HI line gives:



The x-axis can be converted into velocity by setting a reference frequency using -fref, e.g.:

```
/u/hob044/software/new_c/sdhdfProc/sdhdf_plotSpectrum -sb 0 -sd 0 -fref 1667.3590 -f uwl_191208_055418.hdf.T.lsr
```

Note that the user must ensure that the frequency axis has been converted to the local standard of rest (using sdhdf_modify). The result is similar to (this is an OH maser):



-yunit can be used to change the y-axis unit to e.g., “K” or “Jy”

sdhdf_plotMultiSpec

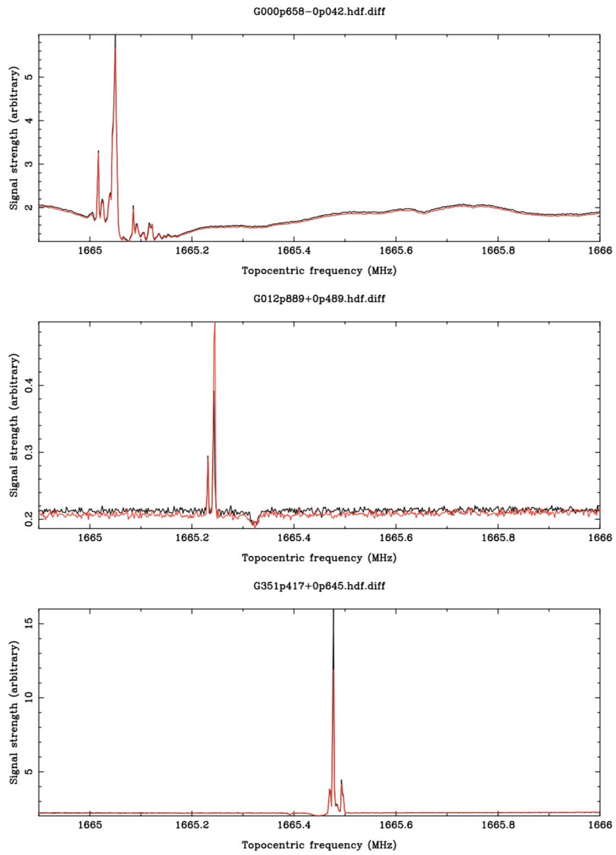
This software is used to make images (png/postscript) of spectra with specified zoom bands etc. Multiple spectra can be plotted simultaneously. Note that this plotting routine is not interactive.

```
/u/hob044/software/new_c/sdhdfProc/sdhdf_plotMultiSpec -sb <subband> -sd <dump> -f0 1666.5 -f1 1667.4
-p -av 8 -g <graphicalDevice> filenames.hdf
```

The options are:

- -f0 - lowest frequency to plot
- -f1 - highest frequency to plot
- -av <n> - average n frequency channels together
- -p - form Pol1 + Pol2
- -nx <N> - number of displays in the x-direction
- -ny <N> - number of displays in the y-direction
- -4pol - plots Pol1, Pol2, Pol3 and Pol4
- -g - set graphical device (e.g., /xs, plot.png/png, plot.ps/cps)
- -ch - set character height
- -join - make a joined plot
- -fref <REF> - set a reference frequency (in MHz). If this is set then velocity is plotted on the x-axis.

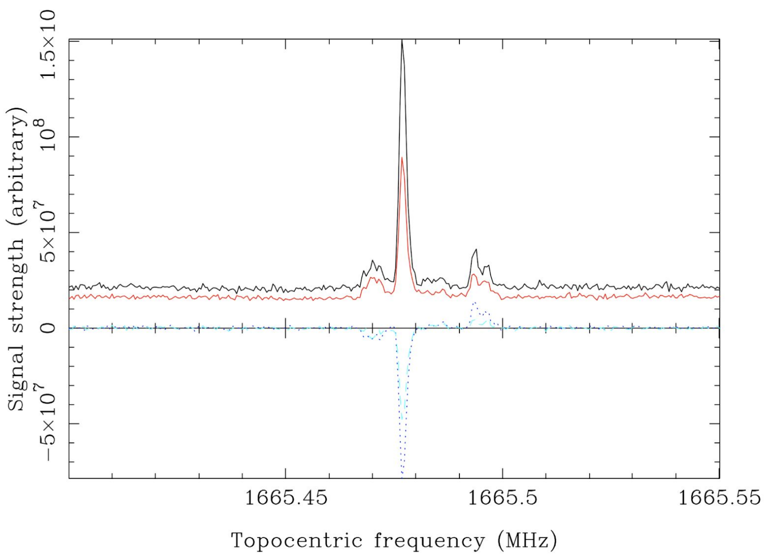
This can produce plots similar to (see Use Case 1):



Where the top plot represents (ON-OFF)/OFF, the central plot is ON and the bottom plot is OFF.

/u/hob044/software/new_c/sdhdfProc/sdhdf_plotMultiSpec -sb 0 -f0 1665.4 -f1 1665.55 -4pol -g maser_pol.ps/cps uwl_191208_055418.hdf

Produces a postscript file (maser_pol.ps) looking like:



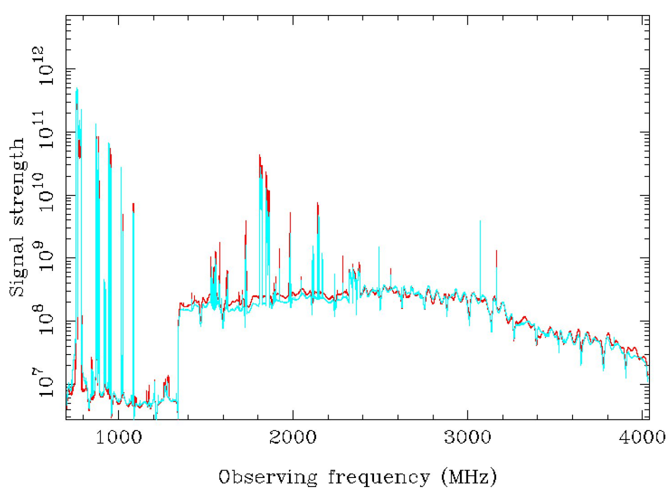
Pol 1 is in black, Pol 2 in red, Pol 3 in dashed light blue and Pol 4 in dotted dark blue.

sdhdf_plotWide

This program plots a wide-bandwidth spectrum (plotting all sub-bands simultaneously).

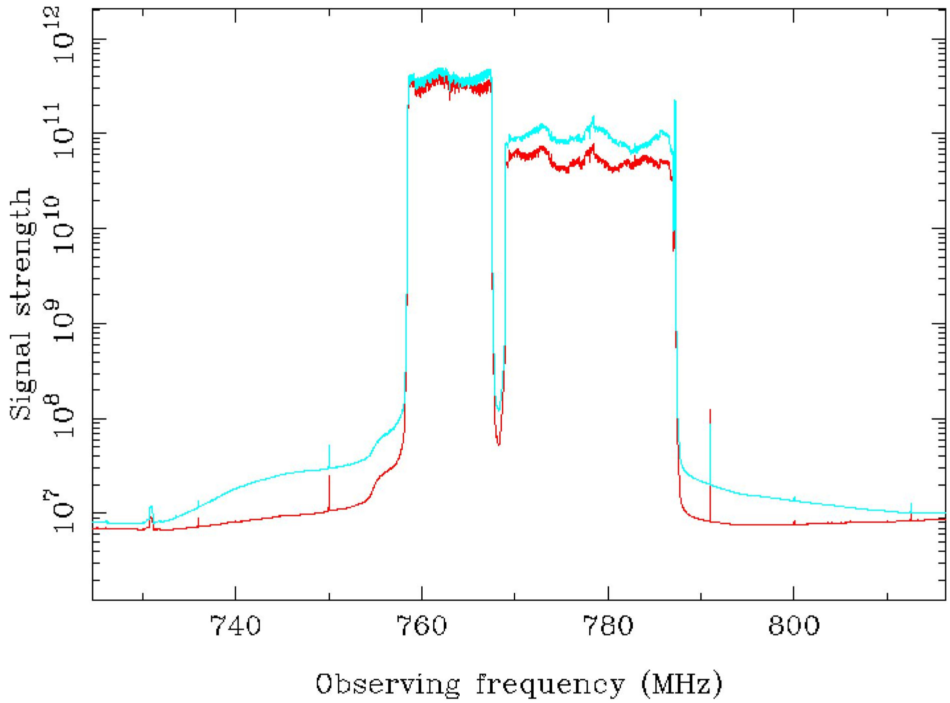
```
/u/hob044/software/new_c/sdhdfProc/sdhdf_plotWide uwl_190607_144518.hdf
```

Produces:

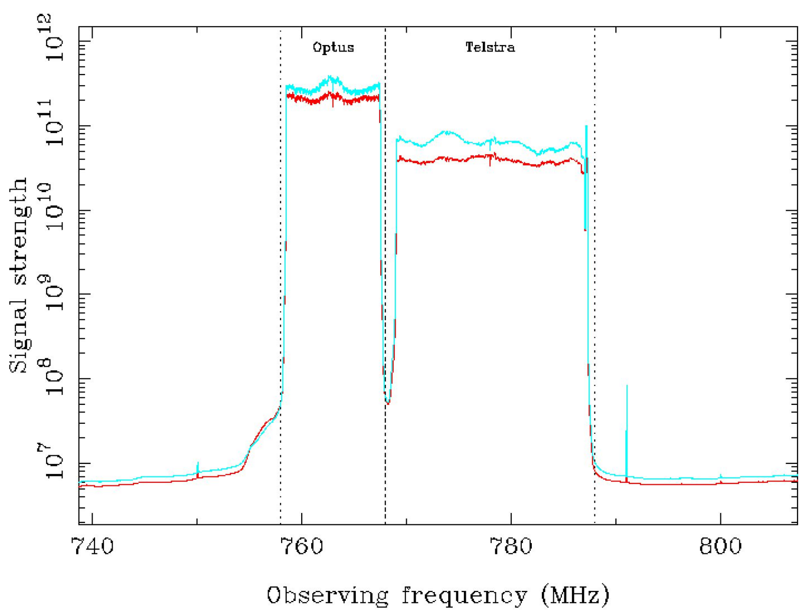


‘z’ zooms in on a region
‘u’ unzooms
‘Q’ quits

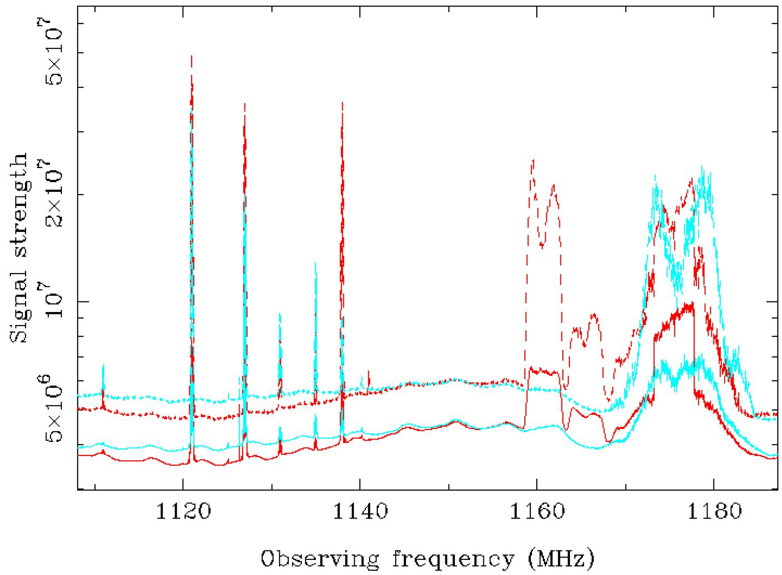
For instance, we can zoom in around the low frequencies:



Pressing ‘t’ highlights all the licensed transmitters in the band:



Pressing ‘m’ overlays a “max-hold” spectrum (with dashed lines):



Command line arguments are:

-minx <val>	Set minimum frequency (MHz)
-maxx <val>	Set maximum frequency (MHz)
-transmitters	Highlight licensed transmitters in the region
-maxhold	Show the max-hold spectra
-g <device>	Set the PGPLOT graphical device
-sb <val>	Set the minimum and maximum frequencies to correspond with sub-band “val” (starting from 0)

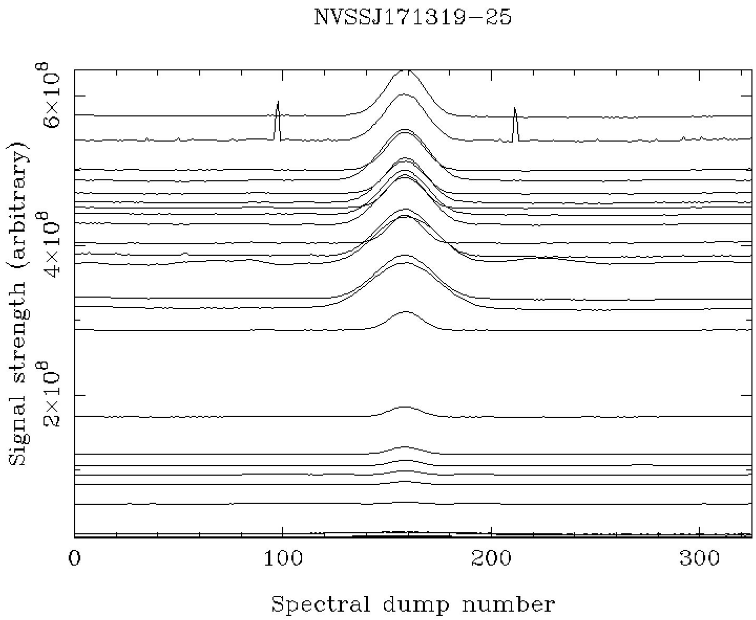
Note that if multiple SDHDF files are provided on the command line then the average and peak-hold spectra are calculated using all data files.

sdhdf_plotScan

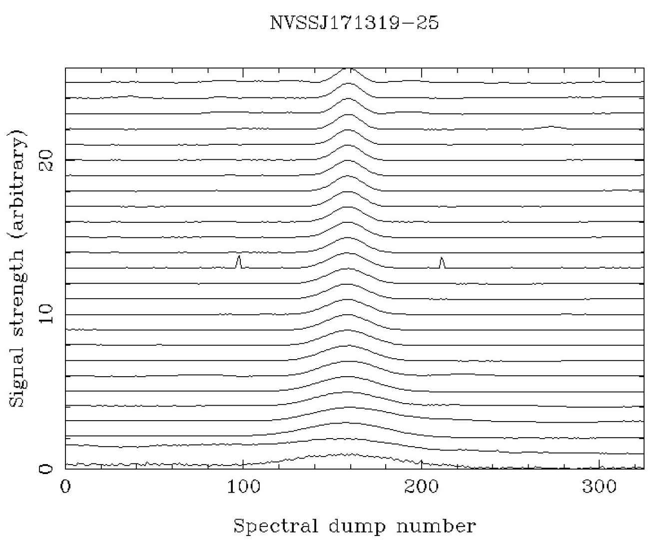
This is used to plot a scan across a source:

```
/u/hob044/software/new_c/sdhdfProc/sdhdf_plotScan  -f uwl_190607_152008.hdf.autoflag
```

produces:



Each line represents a frequency band and no scaling, nor normalisation has been applied. Pressing ‘n’ normalises the scan for each band and ‘o’ offsets each band:

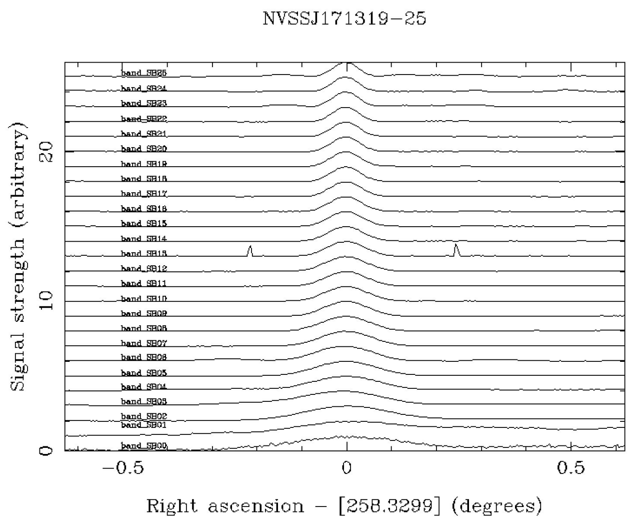


Press ‘L’ to overplot band labels and ‘l’ to toggle logarithmic scaling. The plot can be zoomed using ‘z’ and unzoomed using ‘u’. Pressing ‘x’ toggles the x-axis between:

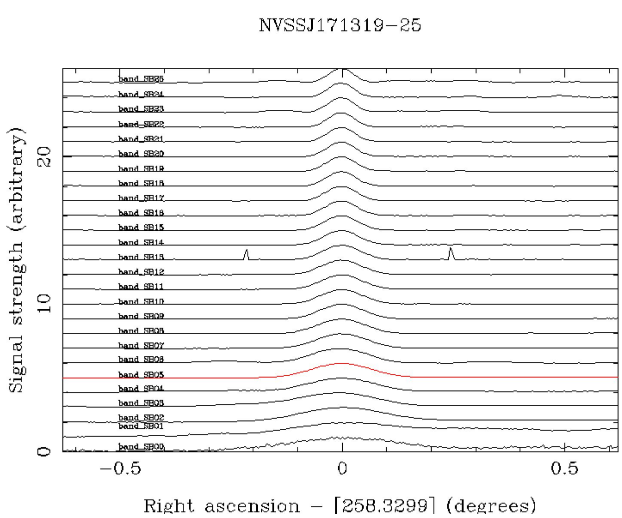
1. Spectral dump number
2. Time since observation start (seconds)
3. Right ascension (degrees)
4. Declination (degrees)
5. Azimuth (degrees)
6. Elevation (degrees)

7. Angular offset from (RA0, DEC0)
8. Right ascension - RA0
9. Declination - DEC0

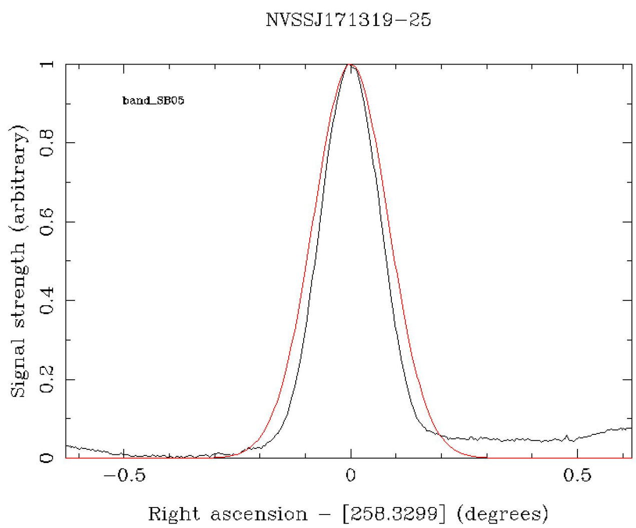
RA0 and DEC0 default to the position of the first spectral dump, but can be manually set using ‘p’. For instance:



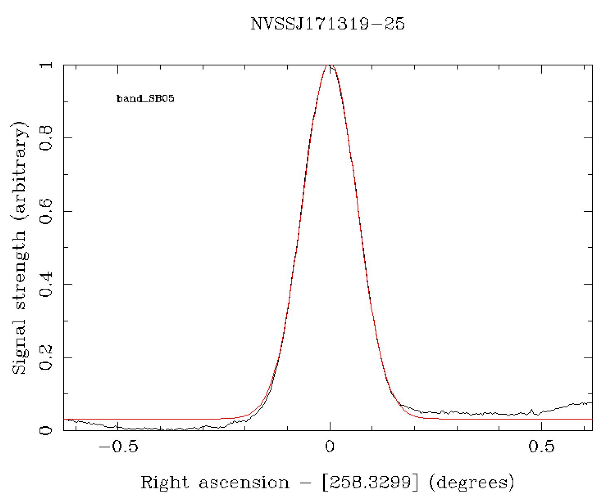
An individual band can be highlighted by pressing ‘+’ or ‘-’ (to scroll through the bands). In the example below SB05 has been highlighted:



That highlighted band can be plotted alone by pressing ‘s’:



The red curve is a Gaussian model of the beam shape. Press ‘f’ to fit the components of the model. This gives a better fit:



and a listing of the fitted parameters:

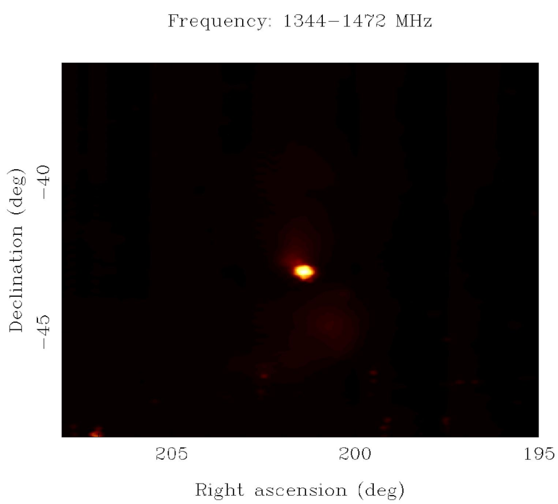
```
Band | Fc (MHz) | Amp | Angular offset | Width | Baseline
band_SB00 768.0000 0.776117 0.0108972 0.114944 0.168786
band_SB01 896.0000 0.8632 0.158046 0.346755 -0.0823061
band_SB02 1024.0000 0.925924 0.00548767 0.0912815 0.0776957
band_SB03 1152.0000 0.876202 -0.0151685 0.0908391 0.106226
band_SB04 1280.0000 0.897565 -0.00168524 0.0723519 0.0968157
band_SB05 1408.0000 0.972383 -0.000307199 0.0646752 0.0321847
band_SB06 1536.0000 0.96098 -0.0052142 0.058264 0.0586605
band_SB07 1664.0000 0.97918 -0.000345035 0.0543364 0.0276678
band_SB08 1792.0000 0.976251 -0.000919361 0.0507308 0.0308873
band_SB09 1920.0000 0.986014 -0.00023826 0.0466827 0.0218206
band_SB10 2048.0000 0.981377 -0.000233602 0.0456762 0.0219426
band_SB11 2176.0000 0.977862 -0.000379431 0.0442345 0.0266903
band_SB12 2304.0000 0.978881 -0.000640946 0.0434602 0.024009
```

```
band_SB13 2432.0000 0.971561 -0.000663769 0.0395886 0.0437241
band_SB14 2560.0000 0.975885 -0.00100852 0.0383559 0.0318558
band_SB15 2688.0000 0.981678 -0.00141107 0.0363431 0.0294878
band_SB16 2816.0000 0.989938 -0.00168522 0.0347653 0.0217726
band_SB17 2944.0000 0.989441 -0.00181886 0.0334649 0.0239982
band_SB18 3072.0000 0.984298 -0.00178864 0.0314896 0.0280998
band_SB19 3200.0000 0.984494 -0.00185576 0.0320129 0.0272024
band_SB20 3328.0000 0.984334 -0.00185476 0.0301215 0.0326871
band_SB21 3456.0000 0.976382 -0.00205249 0.0291086 0.0356294
band_SB22 3584.0000 0.975806 -0.00218435 0.0275619 0.0439998
band_SB23 3712.0000 0.96456 -0.00256303 0.0272785 0.0586242
band_SB24 3840.0000 0.951255 -0.00266011 0.0268112 0.0746141
band_SB25 3968.0000 0.930884 -0.00236068 0.0255899 0.0867758
```

sdhdf_plotMap

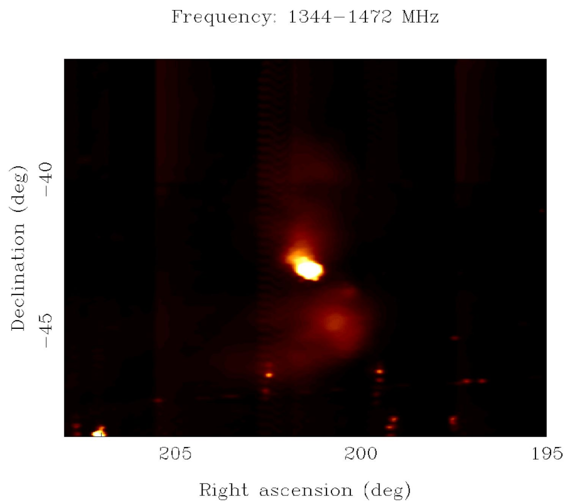
/u/hob044/software/new_c/sdhdfProc/sdhdf_plotMap -f map.store.hdf

Will show a map (created using sdhdf_map) such as:



‘z’ zooms in on a region of the map
‘u’ unzooms
‘q’ quits
‘m’ allows the user to select the minimum and maximum values for the colour range
‘s’ allows a box to the drawn on the image and the statistics of the values within that box are reported.
‘c’ toggles between colour and grey-scale images.

For instance using ‘m’ to reduce the range of the colour scheme produces:


Frequency: 1344−1472 MHz
Declination (deg)
−40
−45
205
200
195
Right ascension (deg)

Manipulating SDHDF files

It is possible to load in one, or more, input data files, carry out some processing steps and then output a new set of SDHDF files. The input files are listed on the command line and the output files have the same names, but with a user-defined file extension. For instance, the input file may be uwl_191208_055418.hdf, but after summing all the spectral dumps together the output file name may be uwl_191208_055418.hdf.T.

- sdhdf_extract: Extracts segments from a given file to produce a smaller output file. This routine can extract specific sub-bands or zoom-bands.
- sdhdf_modify: Modifies a data file by, e.g., averaging spectral dumps, forming Stokes parameters, or converting topocentric frequencies to barycentric or LSR frames.
- sdhdf_sum: Produces a single output file in which the spectra are the sum of all the spectra in the input files.

sdhdf_extract

Enables the copying of specific parts of an SDHDF file to a new SDHDF file. For instance, this code can be used to extract one or more sub-bands from a given file and output to a new file. In this example:

```
/u/hob044/software/new_c/sdhdfProc/sdhdf_extract -e extract -s SB05 -s SB12 input*.hdf
```

Subbands SB05 and SB12 are obtained from files containing input*.hdf in their name and written to the filename with extension .extract (defined by the -e option).

It is also possible to extract one or more zoom-bands from a data file (including over sub-band boundaries):

```
/u/hob044/software/new_c/sdhdfProc/sdhdf_extract -e extract -zoom 1420 1424 input*.hdf
```

Will extract a single zoom band covering frequencies from 1420 to 1424 MHz (defined by the -zoom <f0> <f1> option).

sdhdf_modify

Carries out simple modification/processing routines on an SDHDF file and produces a new one. Options:

-e	output file extension

-T	sum all spectral dumps
-bary	convert frequencies to barycentric frequencies
-lsr	convert frequencies to the local standard of rest
-fav <integer value>	average ‘value’ spectral channels together
-F	average to a single frequency channel
-p1	form total intensity: (Pol 1 + Pol 2)/2. (note dividing output by 2)
-p2	keep on Pol 1 and Pol 2 (not cross terms)
-p3	form total intensity: (Pol 1 + Pol 2) (note **not** dividing output by 2)
-S	transform to Stokes parameters (note, assuming already calibrated)

For example we wish to sum the spectral dumps in uwl_191208_055418.hdf:

```
/u/hob044/software/new_c/sdhdfProc/sdhdf_modify -e T -T uwl_191208_055418.hdf
```

The output is a new sdhdf file called uwl_191208_055418.hdf.T which can then be processed as normal (e.g., using plotting routines).

sdhdf_sum

Software to sum together (and produce an average) of the spectra in multiple files.

```
/u/hob044/software/new_c/sdhdfProc/sdhdf_sum -o outFile.hdf in*.hdf
```

Note that currently this only works with:

1. 1 spectral dump per file
2. All files with the same number of frequency channels and sub-bands
3. 4 pol data

Flags

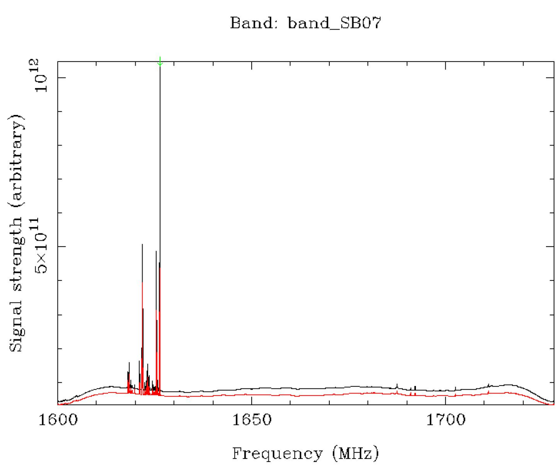
A flag can be set for each channel in a given observing band. A flag = 1 implies that the spectral channel should be discarded. A flag = 0 implies that the spectral channel should be used when plotting and processing. Note that the same flag is used for all spectral dumps and so cannot be used to de-weight bad spectral dumps.

sdhdf_flag

We wish to flag uwl_190608_102736.hdf.

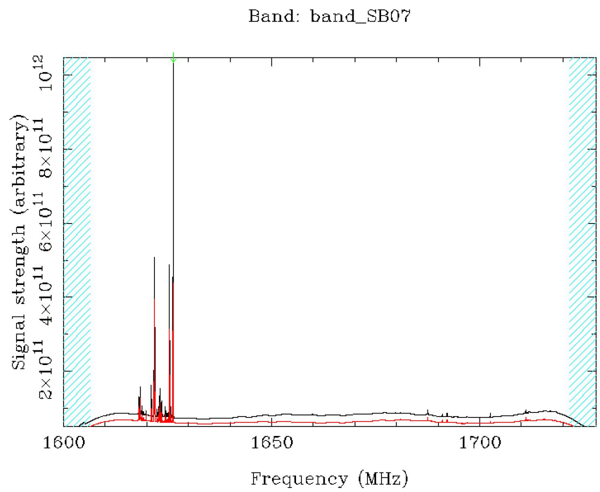
```
/u/hob044/software/new_c/sdhdfProc/sdhdf_flag -f uwl_190608_102736.hdf
```

The time-averaged spectra for a given sub-band are plotted:

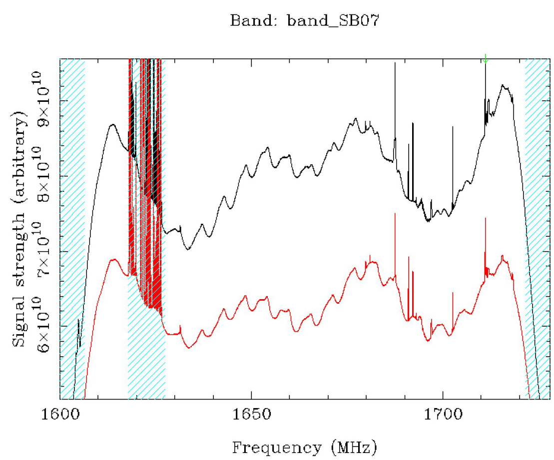


Key presses for general use are:
‘+’ move to next sub-band
‘-’ move to previous sub-band
‘z’ use the mouse to select a region to zoom-in on.
‘u’ unzoom
‘q’ quit

The user can also press ‘b’ which will automatically zap 5% of the Parkes UWL sub-band edges:



Flagged regions are shown using the light-blue, hashed regions. Other regions to flag can be selected manually be pressing ‘Z’ at the left-hand edge of the region, moving the mouse to the right-hand edge and then clicking the left-mouse button.



The plot automatically re-scales on the regions that have not been flagged.

Press ‘s’ to save a new output file (with extension .flag) that will contain the flagging information.

(Note to aid the user for manual flagging the strongest RFI peak in a given plot is identified with a green, downward-pointing arrow at the top of the plot.)

sdhdf_autoFlag

This program enables the user to flag a set of data files automatically. The only routine implemented so far copies the flag information from a specific file (specified using the -from command line argument) to all the other files listed on the command line.

% sdhdf_autoFlag -from cenA.sum.flag uwl*.hdf

NOT IMPLEMENTED YET

1. Automatically zap N% of band edges
2. Automatically zap known RFI
3. Read in lots of files and “OR” the flags.

Converting to and from SDHDF files

sdhdf_convertFrom

This routine converts spectral line data formats into the SDHDF format from a different format:

- Text files
- SDFITS (NOT IMPLEMENTED YET)

The text file format is simply two columns (currently only 1 pol data implemented so far):
<freq (MHz)> <signal>

Let's assume that the input text file is called psd.dat:

```
/u/hob044/software/new_c/sdhdfProc/sdhdf_convertTo psd.dat
```

This produces an output file psd.dat.hdf which can be processed as normal using the tools described in this document.

sdhdf_convertTo

The routine will convert SDHDF files into other data formats. It currently converts to SDFITS format (as defined by https://casa.nrao.edu/aips2_docs/notes/236/node14.html). ***NOTE THAT THIS CODE IS NOT WELL TESTED AND HAS QUITE A FEW HARDCODED PARAMETERS***.

```
/u/hob044/software/new_c/sdhdfProc/sdhdf_convertTo uwl_190608_102736.hdf
```

The input file names are given as command line arguments. The output file names are the same as the input with extension .fits.

Spectral line/continuum processing tools

sdhdf_onoff

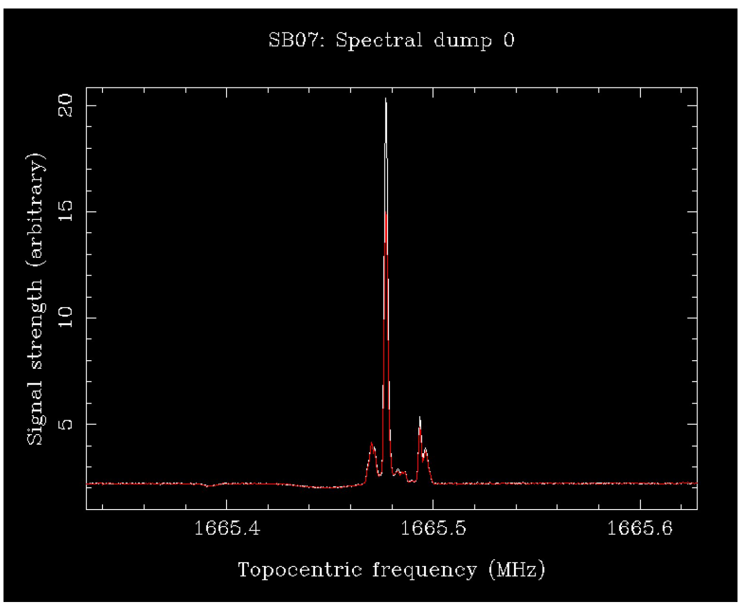
Takes a source observation (ON) and a reference observation (OFF) and produces a new file that contains (ON-OFF)/OFF.

Usage:

```
/u/hob044/software/new_c/sdhdfProc/sdhdf_onOff -on onFile.hdf.T -off offFile.hdf.T -o diff.hdf
```

This produces diff.hdf which contains the difference (ON-OFF)/OFF for each sub-band. The output can then be plotted using sdhdf_plotSpectrum:

```
/u/hob044/software/new_c/sdhdfProc/plotSpectrum -f diff.hdf -sb 0
```



Other sdhdf_onOff command line arguments include:

- -sclA [X] A scaling factor to multiple the output spectrum for Pol A: X(ON-OFF)/OFF
- -sclB [X] A scaling factor to multiple the output spectrum for Pol B: X(ON-OFF)/OFF

It is common to process multiple files simultaneously. To do this, produce a text file with three columns giving the ON, OFF and output filenames, e.g.:

```
uwl_191213_130258.hdf.T uwl_191213_123813.hdf.T s32_01.q
uwl_191214_093035.hdf.T uwl_191214_091041.hdf.T s32_02.q
uwl_191214_095559.hdf.T uwl_191214_101555.hdf.T s32_03.q
uwl_191214_124606.hdf.T uwl_191214_122628.hdf.T s32_04.q
```

```
uwl_191215_082415.hdf.T uwl_191215_085605.hdf.T s32_05.q
uwl_191216_075017.hdf.T uwl_191216_090232.hdf.T s32_06.q
uwl_191216_083041.hdf.T uwl_191216_090232.hdf.T s32_07.q
```

Then we use a command line similar to the following (where onOff_list_s32 is the filename of the file given above):

```
/u/hob044/software/new_c/sdhdfProc/sdhdf_onoff -batch onOff_list_s32
```

Current issues include:

- Requires only 1 spectral dump in both on and off files
- Does not use flags

sdhdf_map

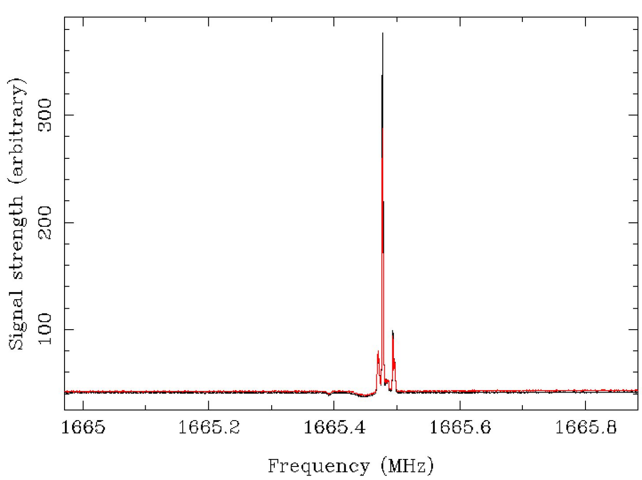
sdhdf_map is used to make a sky map from a set of scans.

```
/u/hob044/software/new_c/sdhdfProc/sdhdf_map -o map.hdf <filenames>
```

sdhdf_baseline

This code is used to determine (and remove) baselines from spectra.

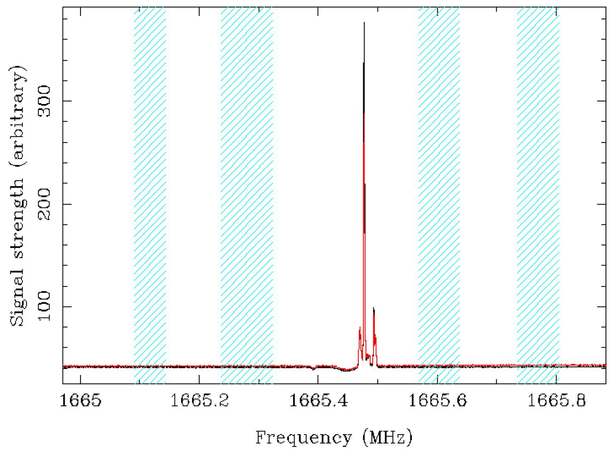
```
/u/hob044/software/new_c/sdhdfProc/sdhdf_baseline -f <filename.hdf>
```



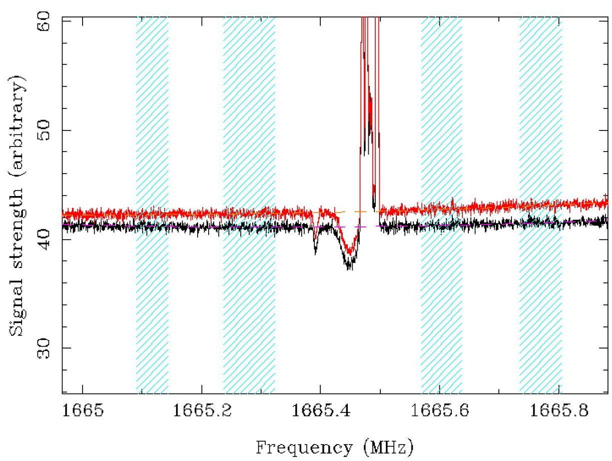
Standard key-strokes can be used to inspect the plot:
'z' - zoom in on a region
'u' - unzoom

‘q’ - quit

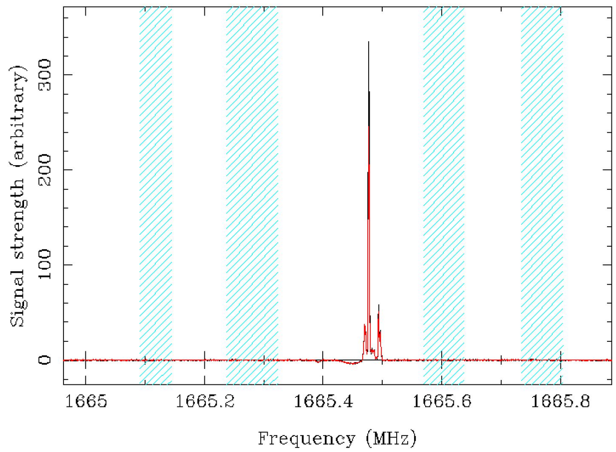
Pressing ‘b’ allows one or more baseline regions to be set. Press ‘b’ when the cursor corresponds to the lowest frequency in the baseline region and then click the left mouse button at the highest frequency. Press ‘b’ again to repeat for a different region. The selected regions will be highlighted:



Press ‘n’ to select the number of polynomial coefficients to include in the polynomial fit (1 = just mean, 2 = quadratic etc.). Then press ‘f’ to fit the polynomial coefficients:



The fitted function (for Pol A and Pol B) are shown as dashed lines. Press ‘r’ to remove the fitted function from the data.



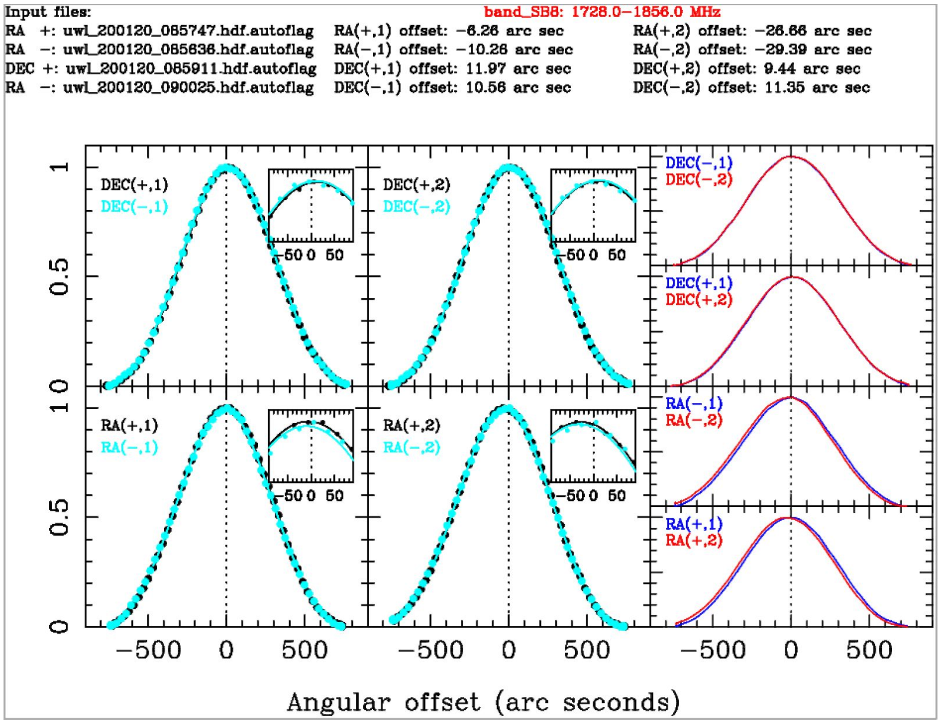
sdhdf_pointing

sdhdf_map is used to make pointing solutions using multiple scans across known sources. **Note that much of this code is currently hardcoded to the Parkes telescope**. The code inputs a scan across a known source in right ascension in the forward direction, then again in the backward direction and then the same in declination.

Typical usage is:

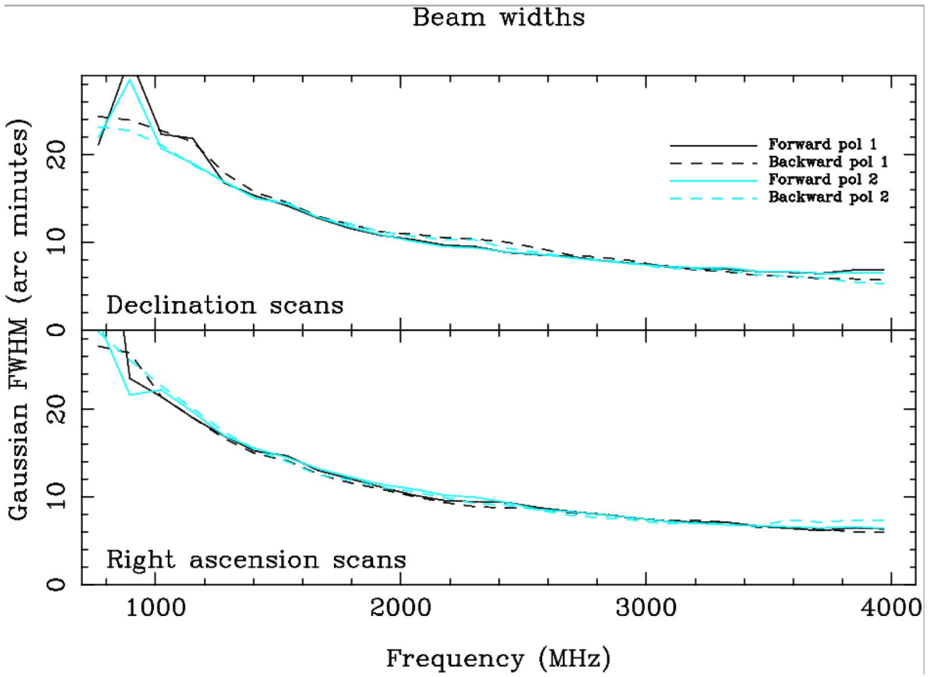
```
/u/hob044/software/new_c/sdhdfProc/sdhdf_pointing -raf uwl_200120_085747.hdf.autoflag -rab
uwl_200120_085636.hdf.autoflag -decf uwl_200120_085911.hdf.autoflag -decb
uwl_200120_090025.hdf.autoflag
```

Note that the data files have already been flagged for RFI (either manually or automatically). In each sub-band the code determines the signal strength versus time. The output is in sdhdf_pointing.ps. For the Parkes UWL receiver the first 26 pages in the output postscript file correspond to the 26 subbands:



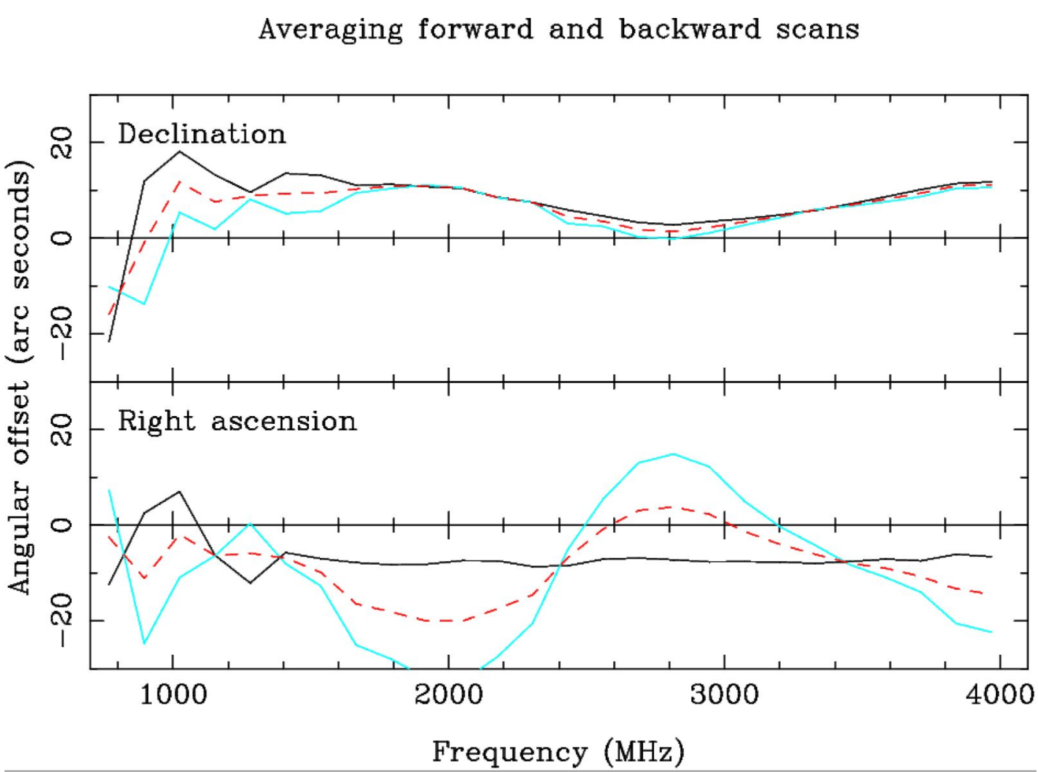
The panels show the normalised signal strength for two polarisations and for the forward and backward directions. The labels indicate RA/DEC(FWD/BACK, Pol1/2). A Gaussian function is fitted to the central regions and the offset from the known source position is listed in the values at the top of the page.

The beam widths are presented on page 27:



And these widths used to derive the antenna gain (in K/Jy) and the antenna efficiency:

The final page shows the pointing offsets:



A .pop file is written out for each sub-band to disk. These files contain the following columns:

dRA(s) dDEC('') HA_hh HA_mm DEC_hh DEC_mm Source FileName RCVR Para AXD ELD

Using the calibrator

The Parkes UWL calibrator can be “ON”, “OFF” or “SWITCHING”. If the calibrator is switching then the Medusa GPU cluster produces (1) a CAL-ON spectrum, (2) a CAL-OFF spectrum and (3) the astronomy spectrum. The CAL-ON and CAL-OFF spectra usually have 128 channels/sub-band (1MHz channels) and a spectral dump time of 5 seconds. The scaling factor between the values (V) in the spectra (measured in counts) for the CAL spectra and the astronomy spectra is:

$$V_{astro} = V_{cal} \frac{N_{chan,astro}}{N_{cha,cal}} \frac{1}{2} \times \frac{t_{astro}}{t_{cal}}$$

where t_astro and t_cal are the dump times for the astronomy spectrum and calibrator spectra respectively. (Note, I think that the factor of 0.5 comes from the fact that only half the signals are in the ON or the OFF spectra compared with the astronomy spectrum).

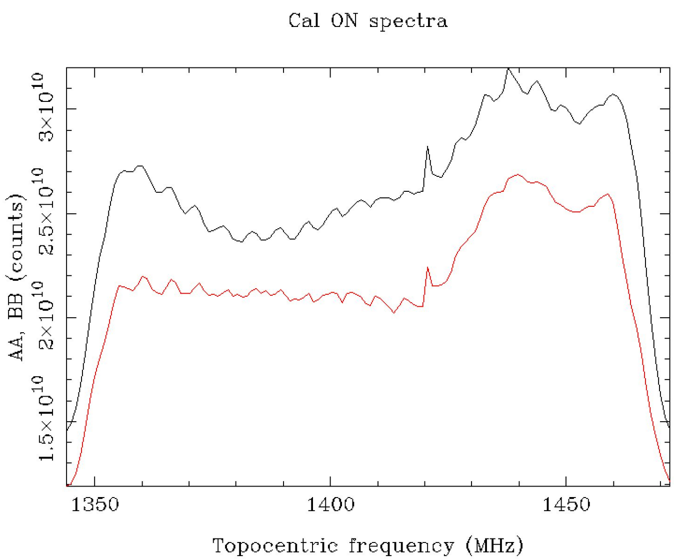
NEED TO HAVE THE METADATA CORRECT FOR SUBINT LENGTHS ETC. BEFORE APPLYING THE CAL AS NEED TO KNOW t_astro

sdhdf_cal

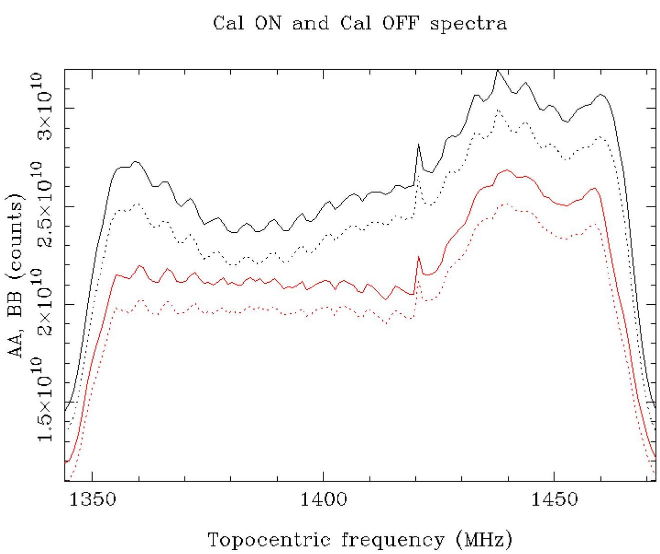
```
/u/hob044/software/new_c/sdhdfProc/sdhdf_cal -sb 1 -f uwl_191205_102837.hdf.T
```

This code produces various plots. Press the numbers 1 to 9 to select different plots. ‘z’ will zoom-in on a plot, ‘u’ will unzoom and ‘m’ will print out the coordinates of a given mouse click.

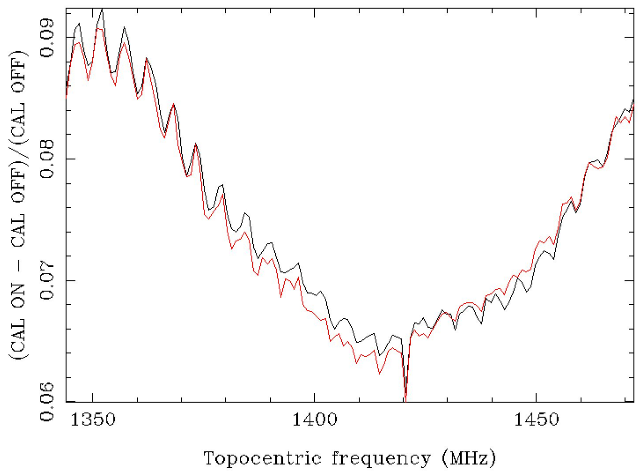
Plots 1 and 2 will show the CAL ON and CAL OFF spectra respectively:



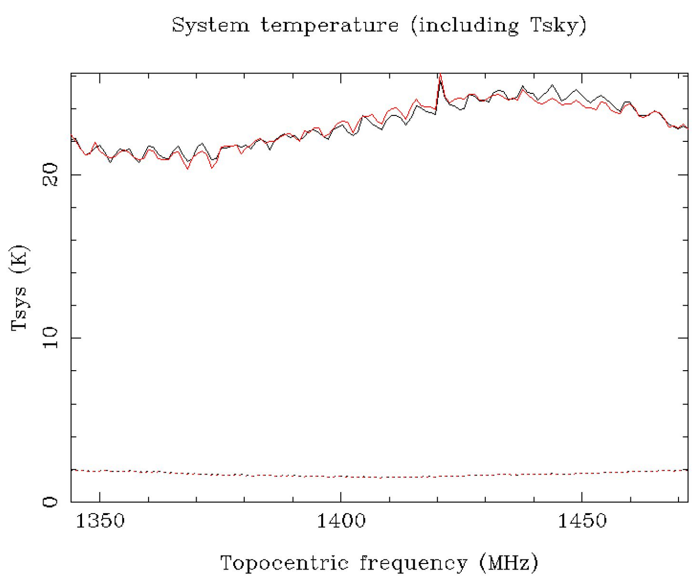
Plot ‘3’ plots the CAL ON (solid) and the CAL OFF (dotted) spectra on the same picture:



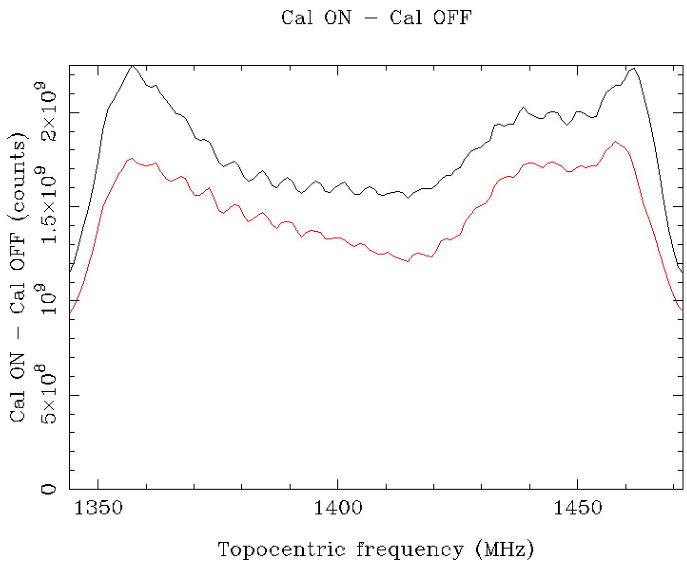
Plot ‘4’ shows (CAL ON-CAL OFF)/(CAL OFF):



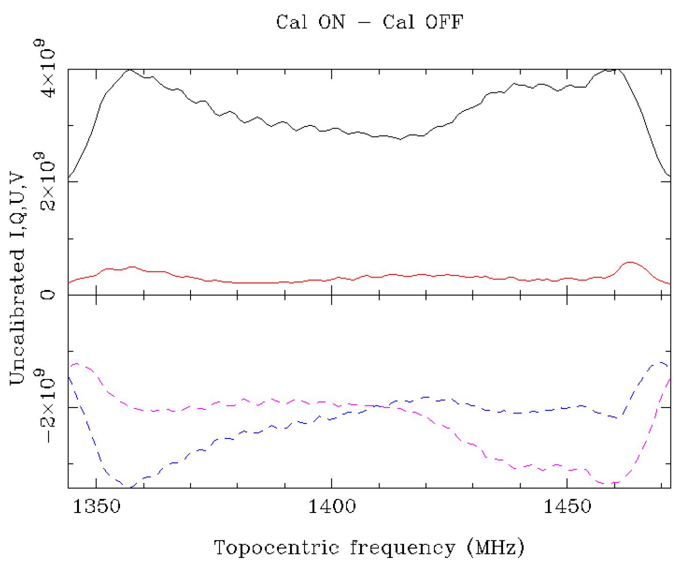
This is converted to Tsys, using knowledge of Tcal(f) which is shown as the dotted line in the bottom of the figure, in plot ‘5’:



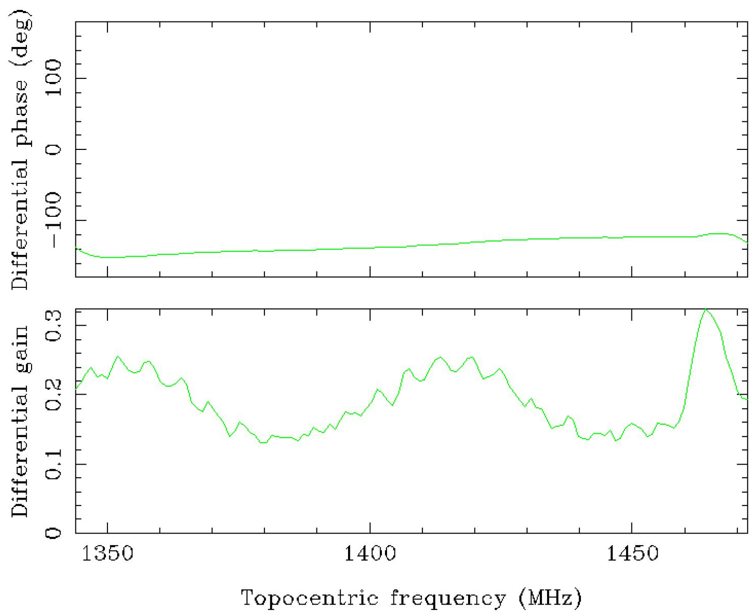
Plot ‘6’ shows (CAL ON - CAL OFF) in units of counts:



The calibrator signal is shown in “uncalibrated Stokes parameters” in plot ‘7’:



The differential gain and phase as measured from the calibrator is plotted in ‘8’:



Flux calibration

There are multiple ways to carry out flux calibration depending on the observation strategy:

1. **Type 1**: “Off source” measurement of system temperature and then (ON-OFF)/OFF determination. Use the known calibrator signal strength in units of K or Jy to scale the output spectrum. Note that this does not require the calibrator source running during the ON-Source measurement.
2. **Type 2:** Only “On source”, but switched calibrator running throughout the observation. Use the known calibrator signal strength in units of K or Jy to scale the astronomy-spectrum.

Type 1: “Off source” measurement of Tsys

Calculating (ON-OFF)/OFF scales the (ON-OFF) spectrum in units of the Tsys for the OFF-source. The switched calibrator can be used to determine Tsys for the “off source” position:

```
/u/hob044/software/new_c/sdhdfProc/sdhdf_cal -f uwl_191208_055751.hdf
```

(view relevant plot by pressing ‘5’ and then ‘m’)

The relevant scaling factors can then be applied to SDHDF_ONOFF:

```
/u/hob044/software/new_c/sdhdfProc/sdhdf_onoff -on uwl_191208_055418.hdf.T -off
uwl_191208_055751.hdf.T -o try.diff -sclA 27.65 -sclB 28.65
```

NOW NEED TO REMOVE A BASELINE AS THE ON-SRC HAS CONTINUUM EMISSION

Miscellaneous code

sdhdf_listLines

This code displays the rest frequencies of known lines in the UWL band:

```
sdhdf_listLines <f0 (MHz)> <f1 (MHz)>
```

This gives an output similar to:

```
1420.4058       HI: Neutral hydrogen
1538.1080       NH_2CHO
1538.6760       NH_2CHO
1539.2640       NH_2CHO
1539.5270       NH_2CHO
1539.8320       NH_2CHO
1540.9980       NH_2CHO
1570.8051       NH_2^13CHO
1584.2740       ^18OH
1610.2469       CH_3OCHO
1610.9000       CH_3OCHO
1612.2310       OH: Hydroxyl radical
1624.5179       ^17OH
1626.1610       ^17OH
1637.5640       ^18OH
1638.8051       HCOOH
1639.5031       ^18OH
1665.4019       OH: Hydroxyl radical
1667.3590       OH: Hydroxyl radical
1692.7950       ^18OH
```

Use case 1: On-off observations of a maser source (G351.417+0.645) (Stokes I)

Goal: To detect OH emission from G351.417+0645 and compare the total intensity signal with Green et al. XXX. The spectra should be plotted as a function of LSR velocity.

Data: uwl_191208_055418.hdf, uwl_191208_055751.hdf (from P737)

1. We start by checking that we have the correct files and we know which is the “ON”-source and which is the “OFF”-source:

   % sdhdf_describe *.hdf

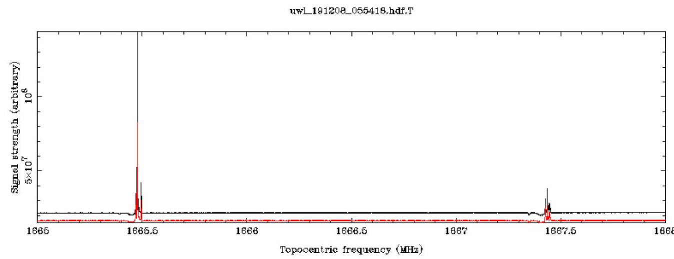
```
-------------------------------------------------------------------------------------------------------------------------------------------------
---
File                                    UTC                      SDHDF PID    Source               Tel        Observer   RCVR    Bands POL_TYPE
CAL   Time
-------------------------------------------------------------------------------------------------------------------------------------------------
---
uwl_191208_055418.hdf                   2019-12-08T05:54:18Z   1.8   P737   G351p417+0p645       Parkes     gre469     UWL     2     AABBCRCI
ON    182.00
uwl_191208_055751.hdf                   2019-12-08T05:57:51Z   1.8   P737   G351p417+0p645_OFF   Parkes     gre469     UWL     2     AABBCRCI
ON    181.00
```

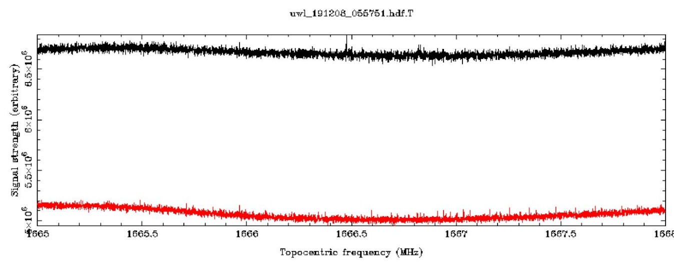
2. We can use the -showb option to determine how many frequency channels we have:

```
----------------------------------------------------------------------
        #   BandID      F0        F1       NCHAN    TDUMP  NPOL   NDUMP
----------------------------------------------------------------------
 [Band] 000 band_SB07   1600.00   1728.00  262144     0.983  4     182
 [Band] 001 band_SB08   1728.00   1856.00  262144     0.983  4     182
```

3. We can average the spectral dumps in each file:
   % sdhdf_modify -T -e T *.hdf

4. Let’s confirm that we see the maser line in the “ON” spectrum and not in the “OFF” spectrum:
   % sdhdf_plotMultiSpec -ny 2 -f0 1665 -f1 1668 -locky 0 1.5e8 *.T

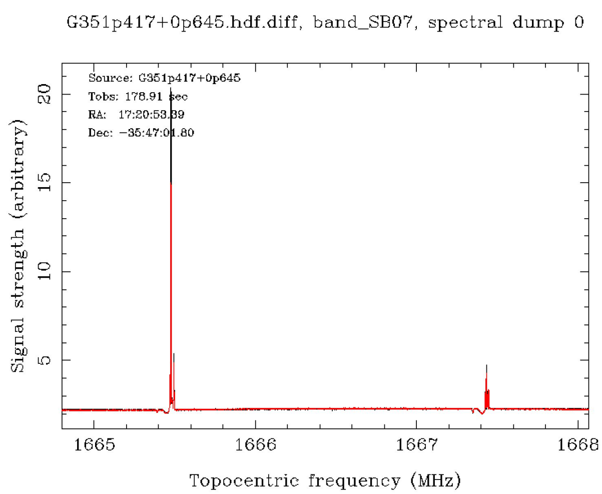




5. Now we form the (ON-OFF)/OFF spectra:

```
% sdhdf_onoff -on uwl_191208_055418.hdf.T -off
uwl_191208_055751.hdf.T -o G351p417+0p645.hdf.diff
```

6. This produces the output plot (using **plotSpectrum -f G351p417+0p645.hdf.diff**):

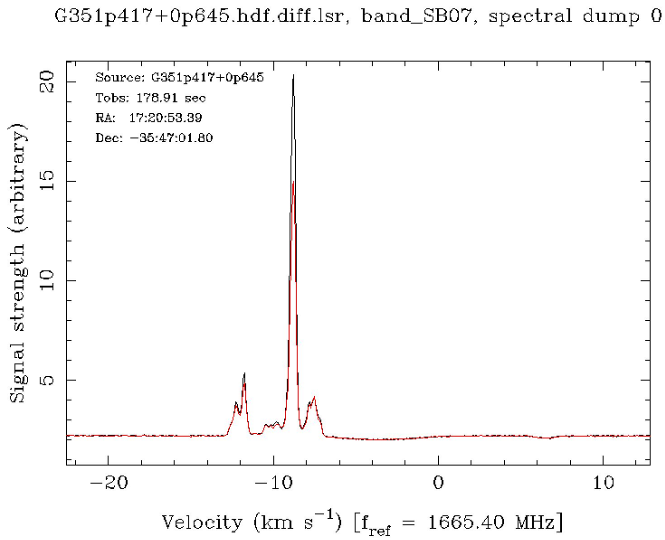


7. We can now convert the x-axis into LSR velocity:

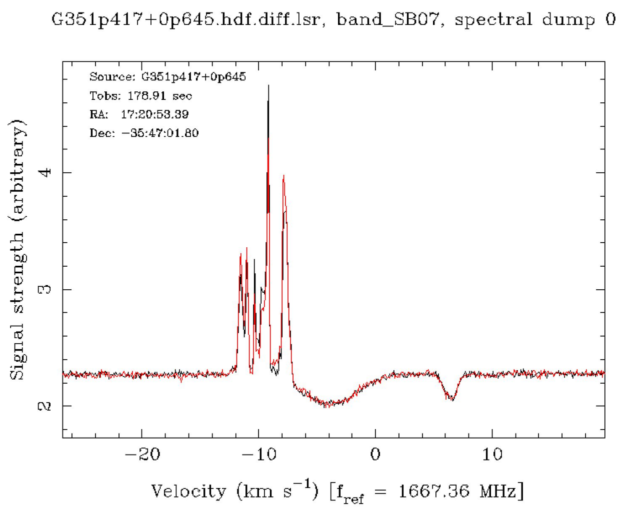
```
% sdhdf_modify -e lsr -lsr G351p417+0p645.hdf.diff
```

8. We can plot the two lines:

```
% sdhdf_plotSpectrum -fref 1665.4019 -f
G351p417+0p645.hdf.diff.lsr
```




```
% sdhdf_plotSpectrum -fref 1667.3590 -f
G351p417+0p645.hdf.diff.lsr
```



9. Note that the y-axis is in units of "Tsys". Next we should convert to Jy (or K).

(should extract the relevant bands near the start --- cannot do this yet as the extract code requires a single sub-int)

Use case 2: On-off observations of a maser source (Stokes I, Q, U and V)

Use Jimi's bright maser observation. Produce calibrated spectra (in velocity and Jy) for all Stokes parameters.

Need updated data files. Need calibration procedures
uwl_191208_055418.hdf P737   G351p417+0p645
uwl_191208_055751.hdf P737   G351p417+0p645_OFF
Also requires 0407 calibration solution.

Use case 3: Scan across bright, continuum source

Have scan across e.g., Hydra A. Demonstrate that we can remove RFI, measure flux density (in Jy) versus offset position. Fit a beam shape to the resulting scan.

Should be able to do most of this now - but need new data files

uwl_191127_195059.hdf: P737 HydraA_DCSCAN

Use case 4: HI observation

Calibrate an HI spectrum in K and LSR velocity. Demonstrate that we can remove the baseline and compare with GASS.   Have recent observation where the ripple will be lower.
Should be able to do now. 2020-01-04-14:15:15.hdf (P737).

Use case 5: Identifying spectral features

Use observation of Sgr region to identify multiple spectral lines. Calibrate in K.
Need suitable observation file.

Use case 6: Mapping

Goal: To create a map of the Cen A region
Data: uwl_190604_090701.hdf to uwl_190608_102736.hdf from P1017 (331 data files)

1. We first average all the spectral dumps within each file (COULD ALSO REMOVE CROSS-TERMS HERE):

> sdhdf_modify -T -e T uwl*.hdf

2. Let's check what files we have:
   > sdhdf_describe *.T

   The output is similar to:

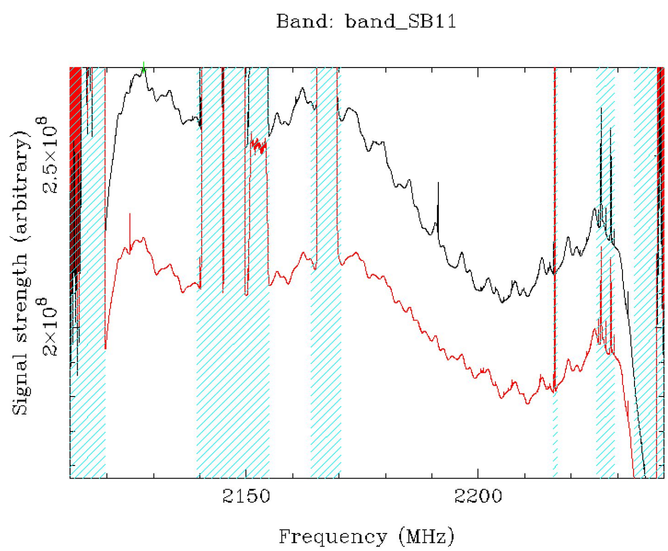
```
-------------------------------------------------------------------------------------------
-------------------------------------------------------
File                         UTC                    SDHDF PID    Source               Tel
Observer   RCVR    Bands POL_TYPE   CAL   Time
-------------------------------------------------------------------------------------------
-------------------------------------------------------
uwl_190604_090701.hdf.T      2019-06-04T09:07:01Z   1.8   P1017  3c273_cal_R
Parkes     CA         UWL     26    AABBCRCI   ON    114.00
uwl_190604_090821.hdf.T      2019-06-04T09:08:21Z   1.8   P1017  3c273_cal_R
Parkes     CA         UWL     26    AABBCRCI   ON    95.00
uwl_190604_090945.hdf.T      2019-06-04T09:09:45Z   1.8   P1017  3c273_cal_R
Parkes     CA         UWL     26    AABBCRCI   ON    114.00
uwl_190604_091118.hdf.T      2019-06-04T09:11:18Z   1.8   P1017  3c273_cal2_R
Parkes     CA         UWL     26    AABBCRCI   ON    133.00
uwl_190604_091921.hdf.T      2019-06-04T09:19:21Z   1.8   P1017  cenA_ra1_100
Parkes     CA         UWL     26    AABBCRCI   ON    437.00
uwl_190604_092328.hdf.T      2019-06-04T09:23:28Z   1.8   P1017  cenA_ra1_101
Parkes     CA         UWL     26    AABBCRCI   ON    437.00
uwl_190604_092736.hdf.T      2019-06-04T09:27:36Z   1.8   P1017  cenA_ra1_102
Parkes     CA         UWL     26    AABBCRCI   ON    437.00
...
```

3. To obtain an "average" spectrum for RFI identification we sum all the cenA observations together:
   > sdhdf_sum -o cenA.sum `sdhdf_identify -src cenA -noinfo uwl*.hdf`
   (this output file can be viewed using sdhdf_plotSpectrum).

4. Now let's flag the RFI:
   > sdhdf_flag -f cenA.sum

   Press 'b' to remove 5% of band edges
   'z' to zoom-in ('u' to unzoom)
   'Z' to flag a region.
   '+' to move to the next sub-band
   '-' to move to the previous sub-band

   After flagging, the plots should be similar to:



Press ‘s’ to save the flagged data file as cenA.sum.flag.

5. We wish to apply this flagging information to all our observations (note we need the scans here and so we use the original .hdf files):

   > sdhdf_autoFlag -from cenA.sum.flag uwl*.hdf

   This will produce multiple files with file extensions .autoflag

6. Should now set scalings based on closest primary calibrator (or using the cal etc.)

   The calibrator observations of 3C273 are:
   uwl_190604_090945.hdf
   uwl_190605_090916.hdf
   uwl_190606_092526.hdf
   uwl_190607_091754.hdf
   uwl_190608_091254.hdf

   Let’s average over all the spectral dumps in these files:

   > sdhdf_modify -T -e T uwl_190604_090945.hdf uwl_190605_090916.hdf uwl_190606_092526.hdf uwl_190607_091754.hdf uwl_190608_091254.hdf

   Now we scale the flagged spectra into units of the primary calibrator

   > sdhdf_scale -from uwl_190604_090945.hdf.T uwl_190604_*.hdf.autoflag
   > sdhdf_scale -from uwl_190605_090916.hdf.T uwl_190605_*.hdf.autoflag
   > sdhdf_scale -from uwl_190606_092526.hdf.T uwl_190606_*.hdf.autoflag

> sdhdf_scale -from uwl_190607_091754.hdf.T uwl_190607_*.hdf.autoflag
> sdhdf_scale -from uwl_190608_091254.hdf.T uwl_190608_*.hdf.autoflag
(should do 190609 and 190610 as well - no cals?)

7. Now let's make the map:
   > sdhdf_map -sb 5 uwl*.hdf.autoflag.scale
   (/u/hob044/software/new_c/sdhdfProc/sdhdf_map -sb 5 uwl_190604_*.hdf.autoflag.scale)
8. (/u/hob044/software/new_c/sdhdfProc/sdhdf_map -loadPlot mapGrid.dat)

Use case 7: Rise to set observations of PKS 1934

OLD: time variability of a continuum source

We have multiple hours of an observation of UV Ceti in /DATA/BRAHE_1/hob044/uvCeti and we wish to plot the total intensity signal as a function of time. For each sub-band we have 512 frequency channels. We first need to flag channels significantly affected by RFI:

```
brahe-111% /u/hob044/software/new_c/read_hdf2/sdhdf_flagRFI -f uwl_191122_122451.hdf
```

We now make a text file (uvCeti_onOff.dat) containing on-source pointings (left column) and off-source pointings (right column):

```
uwl_191122_100947.hdf uwl_191122_101221.hdf
uwl_191122_101457.hdf uwl_191122_101731.hdf
uwl_191122_102009.hdf uwl_191122_102246.hdf
uwl_191122_102524.hdf uwl_191122_102757.hdf
uwl_191122_103034.hdf uwl_191122_103313.hdf
uwl_191122_103553.hdf uwl_191122_103832.hdf
…
```

Future plans

Software

- **Code to fix metadata (i.e., change observer name etc.)**
- **Baseline tools. Choose file with HI, extract zoom-band around the HI. Define baseline regions and enable a fit (with various properties). Write the fit parameters somewhere in the file. Update other codes to remove the baseline if requested.**
- Update all code to enable batch processing
- Dealing with spectral ripples
- Simulating SDHDF spectral line files
- Pointing and focussing parameters
- Plotting flux time series
- 3-D cubes - spatial and velocity
- Calibration for K and Jy (check attributes)
- Calibration for Stokes parameters
- Code to output calibration data sets
- **Convert from SDFITS (archival data) to HDF5**
- Install formally and on different computers
- Get an RFI(Az,El) plot
- Update code for nbeams
- **Weights table**
- ATOA requirements
- Update so that arguments do not necessarily need to be before filenames
- **Pointing and focussing determination**
- Check definition of VLSR - taken from Carl Heiles' code
- Conversion to Left and Right circular polarisation
- Make the entire code less likely to fail if incorrect filenames etc.
- Should change info in obs_params to centre of dump
- Must ensure that cal_proc group gets passed around
- **Have a batch-style (sdhdf_zap) zapping tool**

1) **Auto-zap known RFI - have auxiliary/rfi table that contains**
    a) **Frequency range**
    b) **MJD date range for when RFI was on (could be 0 to 999999)**
    c) **Flag indicating whether it should be flagged automatically (i.e., always there)**
2) autoFlag code
3) NPOL = 1
4) Convert to Karma (kvis) format
5) Must increase integration time output from sdhdf_sum

6) Integration times etc. when writing a new file