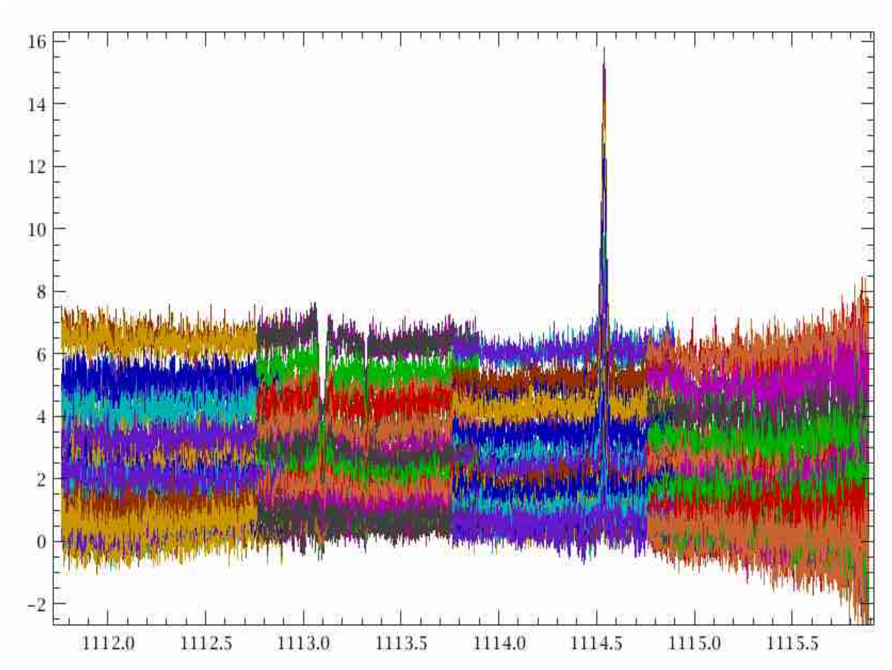
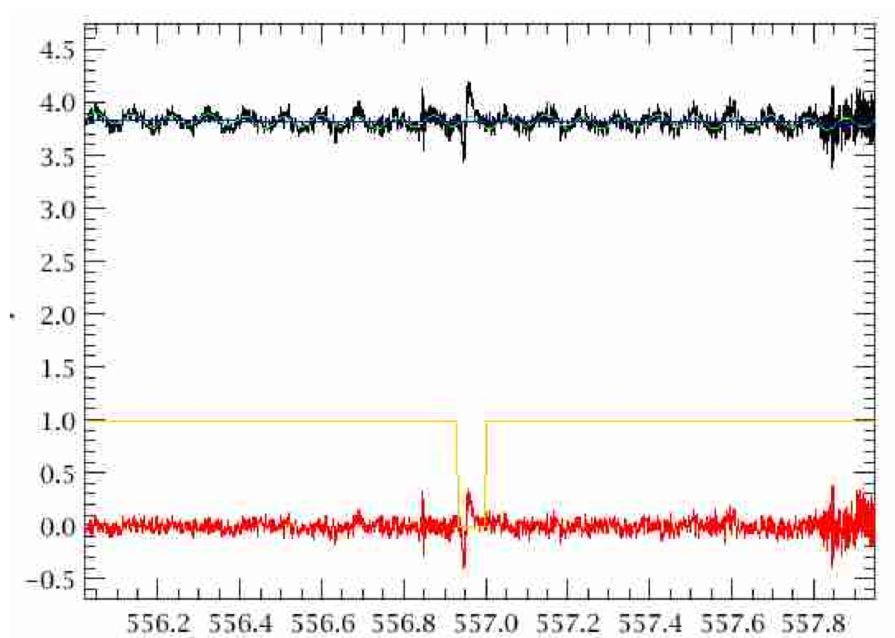
Removing Artifacts from HIFI Spectra

Adwin Boogert
NHSC/IPAC, Caltech





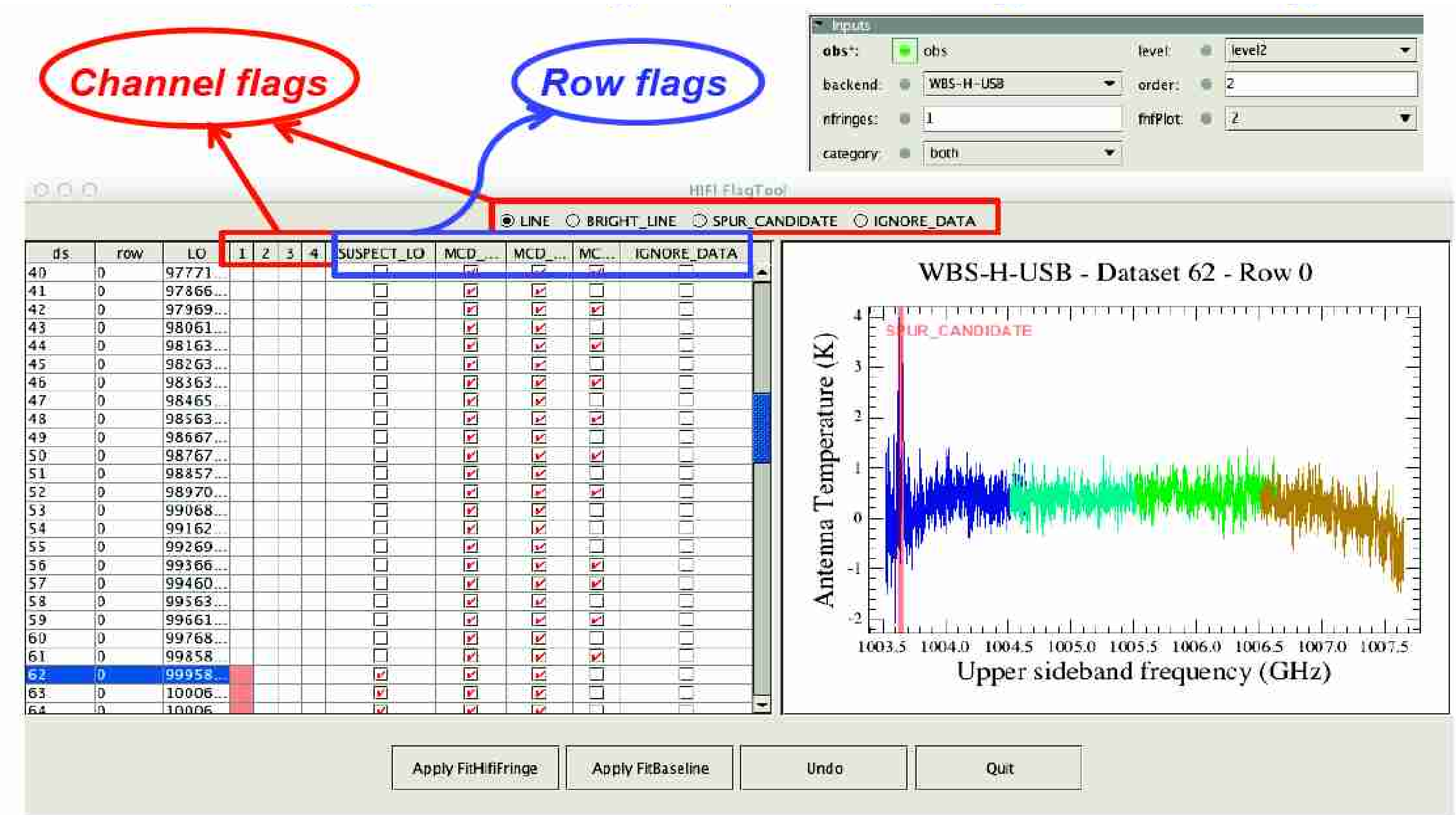
Typical HIFI Post-Pipeline Data Flow

Subset of following steps needed to improve pipeline level 2/2.5 spectra:

1. Overall data inspection: SpectrumExplorer *[demo-ed earlier]*
2. Flag spurs and other bad data: flagTool or script
3. Remove standing waves: fitHifiFringe or more advanced doFilterLoads and HEBStWvCatalogCorrection.py pipeline methods
4. Remove baseline offsets and slopes: fitBaseline
5. Maps: reconstruction cube with doGridding *[demo-ed later]*
6. SScans: sideband separation with doDeconvolution *[demo-ed later]*
7. Averaging H and V-spectra (polarPair) or cubes [*demo-ed later*]
8. Correct for beam efficiency (e.g., T_A to T_{MB})
9. Exporting data in other formats (ASCII, FITS, CLASS, VO)

Flag Bad Data

Instrumental spikes (spurs) and other pipeline flags visualized in SpectrumExplorer and flagTool. Latter is best to modify flags. Spectral lines can also be flagged with special line flags, useful in baseline fitting.



Flags that Matter

Flag definitions are given in Chapter 8 in HIFI Users Manual.

Row flags apply to entire spectrum, channel flags to individual channels.

Most tools will ignore data that have these channel flags:

bit	meaning
0	BAD_PIXEL
1	SATURATED
3	NOT_CALIBRATED
6	DARK_PIXEL
7	SPUR_CANDIDATE *check if there is indeed a spur in the data!*
28	LINE *set by user, used for line masking fitBaseline, fitHifiFringe*
29	BRIGHT_LINE *set by user, used by doDeconvolution to avoid ghosts*
30	IGNORE_DATA *set by user*

Some row flags may indicate serious issues with the data. Consult the quality report in the ObsContext to see if they are truly severe. Also bit 20 (IGNORE_DATA) can be set by the user to flag an entire spectrum as bad.

Baseline Artifacts

Similar to ground-based heterodyne instruments, HIFI instabilities and "imperfect" AOT or pipeline design leave instrumental signatures in Level 2 and 2.5 data:

- Standing waves with different periods, shapes, amplitudes
- Slopes
- Offsets
- Jumps between sub-bands

HIPE offers basic tools to correct for this:

1) fitHifiFringe: HIFI-optimized sine-wave fitting tool
2) fitBaseline: HIFI-optimized polynomial fitting tool

As well as more advanced tools:

1) DoFilterloads: optional pipeline step to remove particular load waves
2) HEBStWvCatalogCorrection.py: removal band 6+7 HEB electronic waves

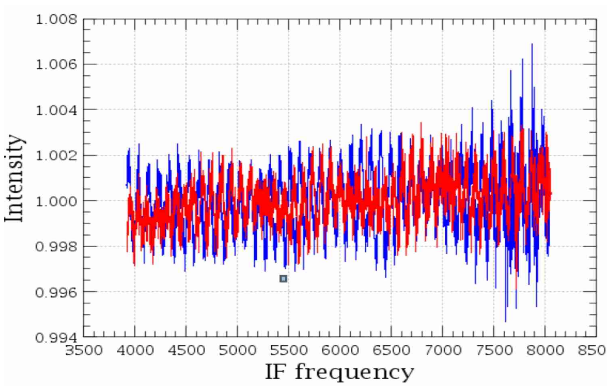
Baseline Problems: Standing Waves

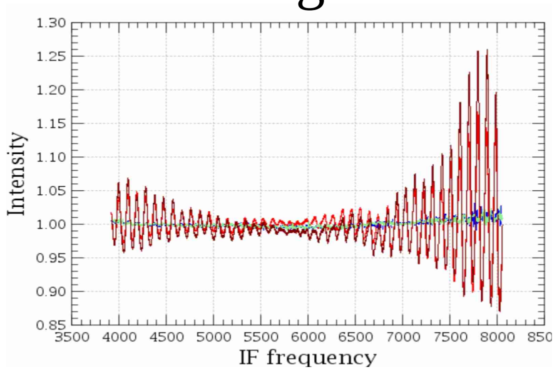
Standing waves produced by optical and electronic components. Pipeline removes them by subtracting reference sky or load spectra.

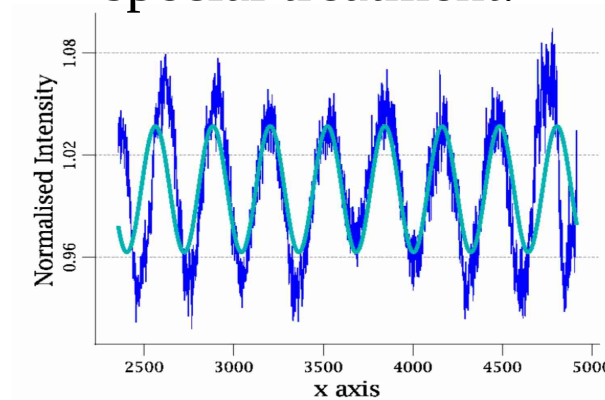
Standing wave residuals sometimes seen in Level 2/2.5 data. Strength in agreement with HSPOT predictions.

Wave-type is HIFI-band dependent:

- Beamsplitter bands 1, 2, and 5 show sine waves
- Diplexer bands 3 and 4 show sine waves with amplitude increasing to IF band edges
- HEB bands 6 and 7 waves are not sine waves. Requires special treatment.







Optical Waves in Level 2 Data

Origin (Mixer to..)	Period [MHz]	Amplitude Bands 1-2	Amplitude Bands 3-4	Amplitude Band 5	Amplitude Bands 6-7
Cold Black Body	98	3-4%	1-2%	1%	<1%
Hot Black Body	92	3-4%	1-2%	1%	<1%
Local Oscillator	100	<1%	2-4%	3%	3-25%
Roof Top Mirror	620	n.a.	1-2%	n.a.	<1%

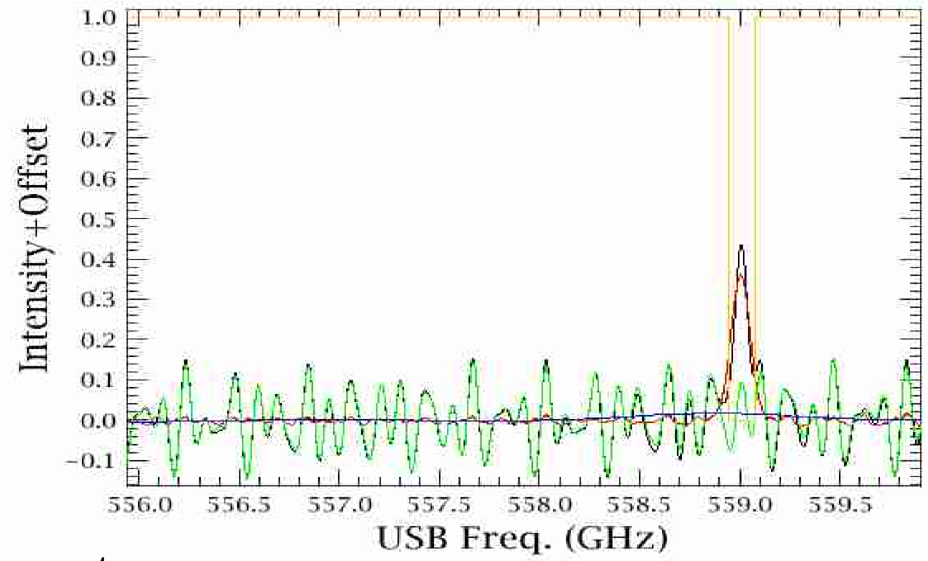
In this workshop we will discuss removal residual waves with fitHifiFringe.

Alternatively, the Cold and Hot Black Body load waves can be removed with the DoFilterloads pipeline step or in combination with fitHifiFringe.

Bands 6 and 7 also have a strong electronic standing wave with a period of 320 MHz. Special “pattern match method” to remove these waves is under development, but advanced user script is available.

Standing Wave Removal with Sine Wave Fits: *fitHifiFringe*

- In order to fit sine waves to the standing waves, the standing waves need to be separated from:
  - other baseline fluctuations
  - emission/absorption lines



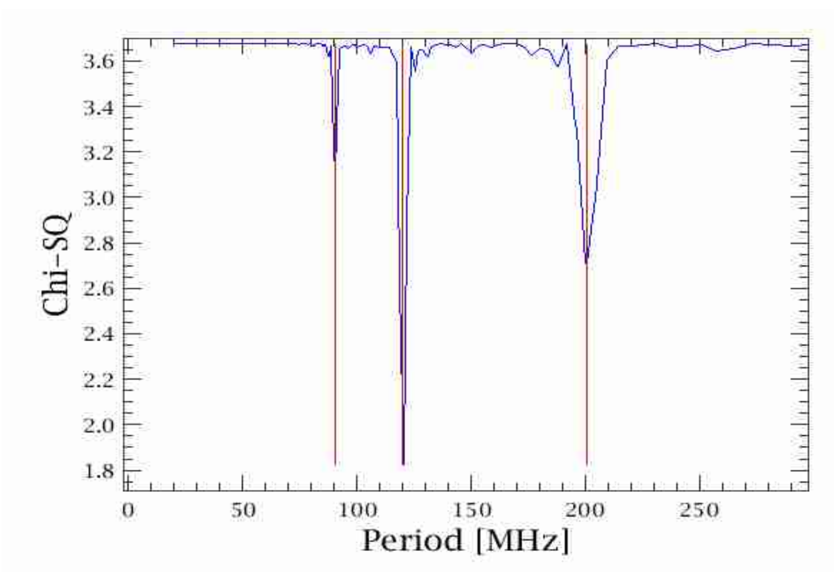
- Fit N sine waves with different periods, amplitudes, phases to baseline-subtracted, line-masked spectrum.

- Subtract SW fit from original input data.
- fitHifiFringe does this by default automatically. But it is not always perfect, and the user has to inspect each plot and may need to set line masks by hand.

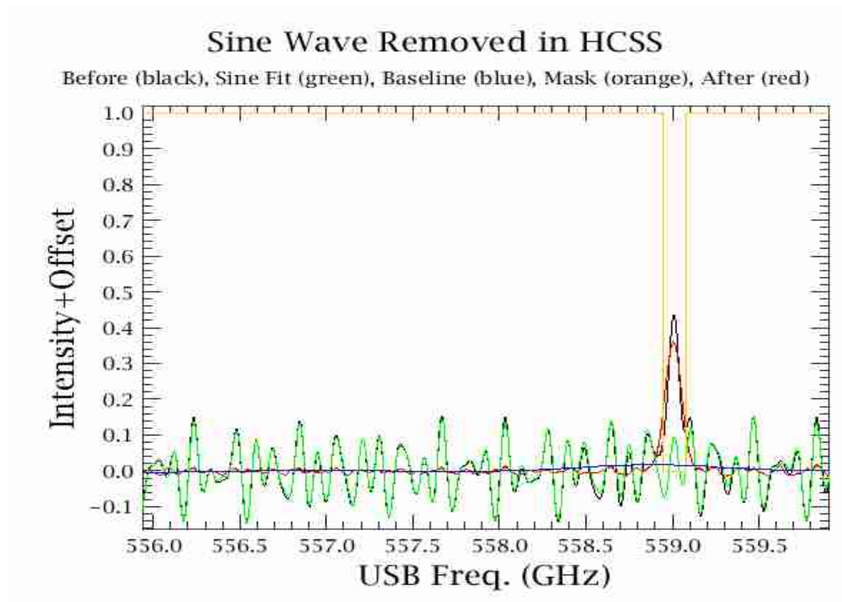
Standing Wave Removal with fitHifiFringe

fitHifiFringe shows fitted periods, amplitudes, phases in HIPE console, stores them in ObsContext, and by default produces 2 plots:

- χ^2 as function of period. Minima found are indicated with red vertical lines



- Result plot: input data, sine wave fit, baseline, mask, sine-wave subtracted spectrum



Standing Wave Removal with fitHifiFringe

Most important fitHifiFringe input:

•nfringes: number of sine waves to fit
•start_period: shortest period SW to search for
•end_period: longest period SW to search for
•typical_period: typical SW expected in data. Longer period structures are assumed to be baseline or sky features.
•doglue: determine SW on combined sub-bands
•usermask: user-defined mask
•sub_base: subtract smooth baseline as well
•averscan: determine SW from average of all scans, and subtract that from all.

fitHifiFringe output: sine wave(s)-subtracted data (obs, htp, sds) and list of sine wave parameters fitted

Standing Wave Removal with fitHifiFringe: *Limitations*

Standing waves were successfully removed with fitHifiFringe in all bands, but it cannot be guaranteed for every observation:

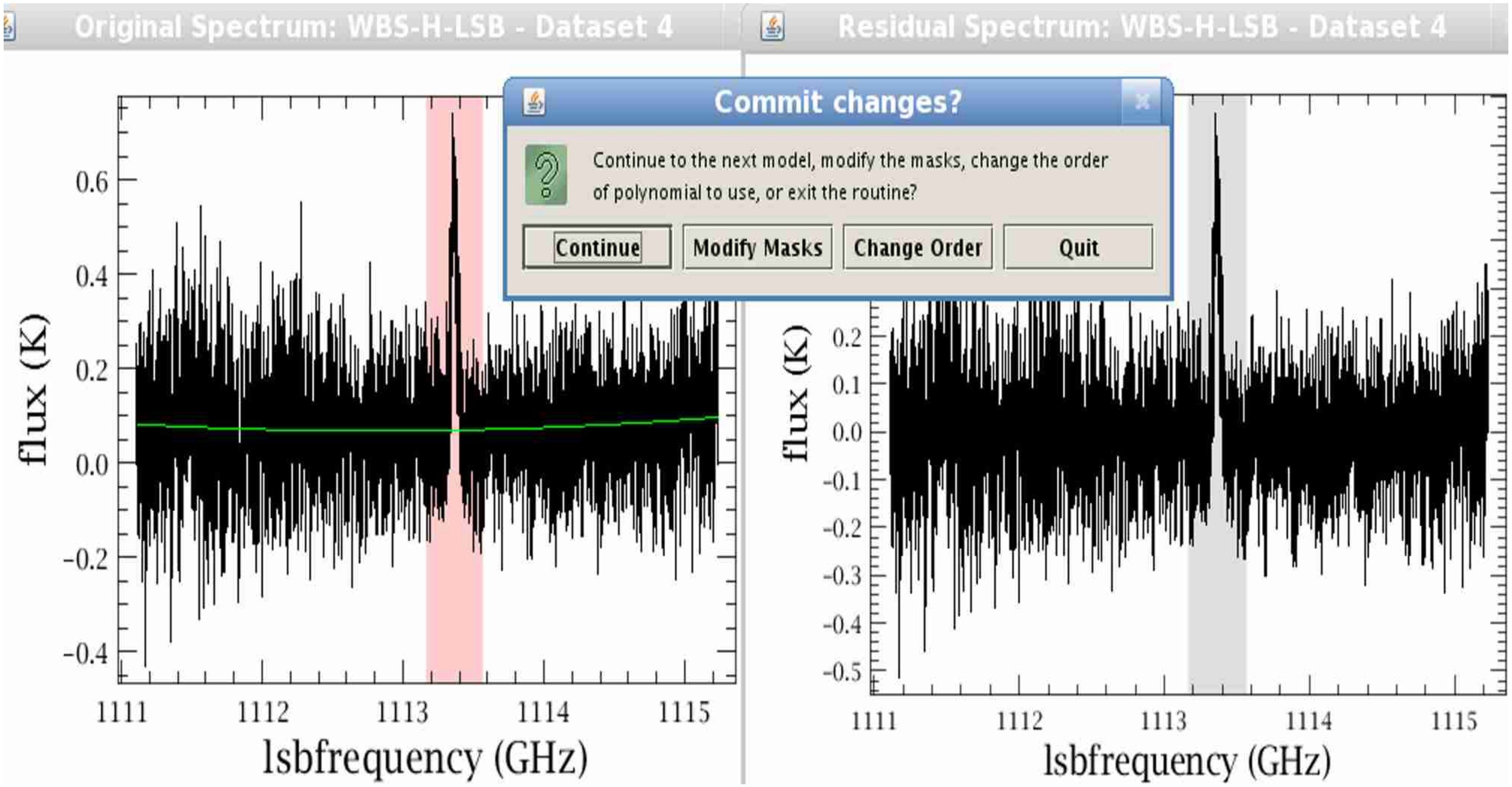
- If there are so many lines, that little 'clean' baseline is left
- Bands 3 and 4 'diplexer' waves are not pure sine waves. It helps if lines are near middle of band, where amplitudes are lowest.
- Band 6 and 7 'electronic' waves are not pure sine waves. An alternative 'pattern-matching' method is available in the advanced HEBStWvCatalogCorrection.py script.
- In specific cases, esp. for deep integrations in bands 1 and 2, the doFilterLoads method mentioned in next slides is better alternative

Baseline Correction with fitBaseline

- fitBaseline: user-friendly tool for polynomial baseline fitting and subtraction or division

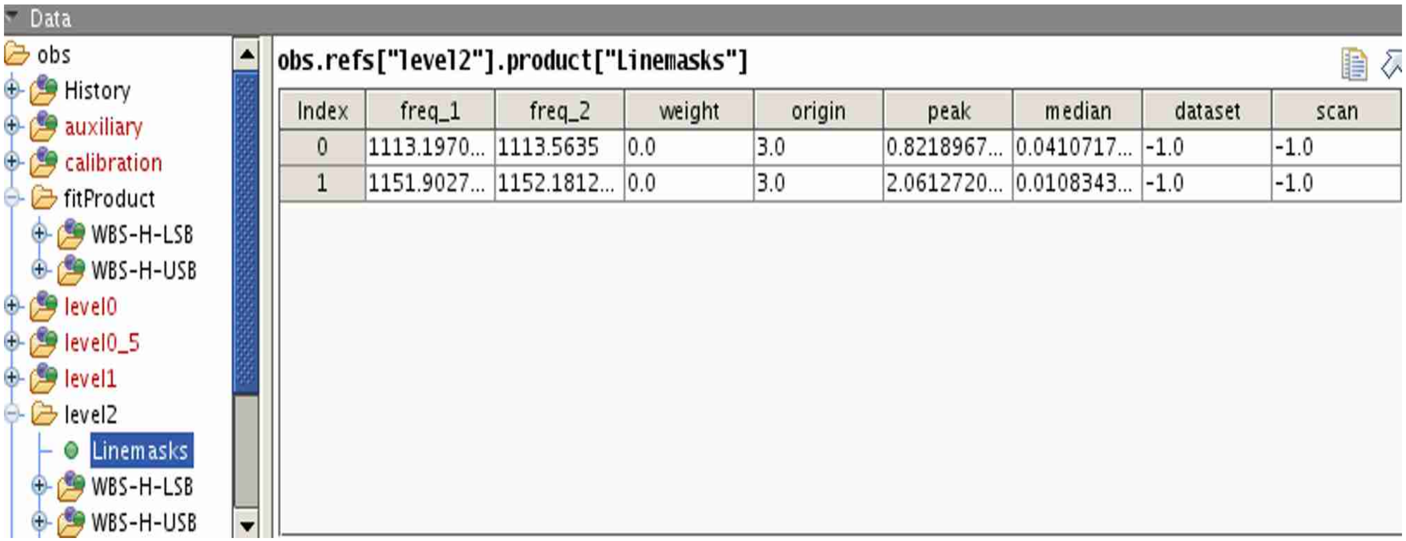
- Features include:
  - Mask lines or spurs by clicking or use automated masking procedure
  - Disable mask, change polynomial order iteratively by user
  - In SScans, automatically determine sideband of line, and then propagate masks to other LO settings
  - Line mask is stored in table in ObsContext
  - Before/after spectra and mask spectra are all stored
  - Re-do fits using stored masks and polynomial orders. Useful after new pipeline processing.
  - Subtract WBS polynomials from HRS spectra (useful if they have little 'clean' baseline)

Baseline Removal with Polynomial Fits: *FitBaseline*



For polynomial baseline fitting with FitBaseline one may need to inspect each plot and adjust the line masks by hand.

Baseline Correction with FitBaseline
Line Mask



Index	freq_1	freq_2	weight	origin	peak	median	dataset	scan
0	1113.1970...	1113.5635	0.0	3.0	0.8218967...	0.0410717...	-1.0	-1.0
1	1151.9027...	1152.1812...	0.0	3.0	2.0612720...	0.0108343...	-1.0	-1.0

Masked frequency ranges stored in a table in the ObsContext.
This is in fact a line list.

Advanced Topic 1: Removing Optical Waves with doFilterLoads Step

Removing CBB and HBB Waves

The 92 and 98 MHz hot and cold load waves are divided into the sky spectra during passband calibration:

$J_{ON}-J_{OFF}=(C_{ON}-C_{OFF})/(C_{HBB}-C_{CBB})*(J_{HOT}-J_{COLD})$

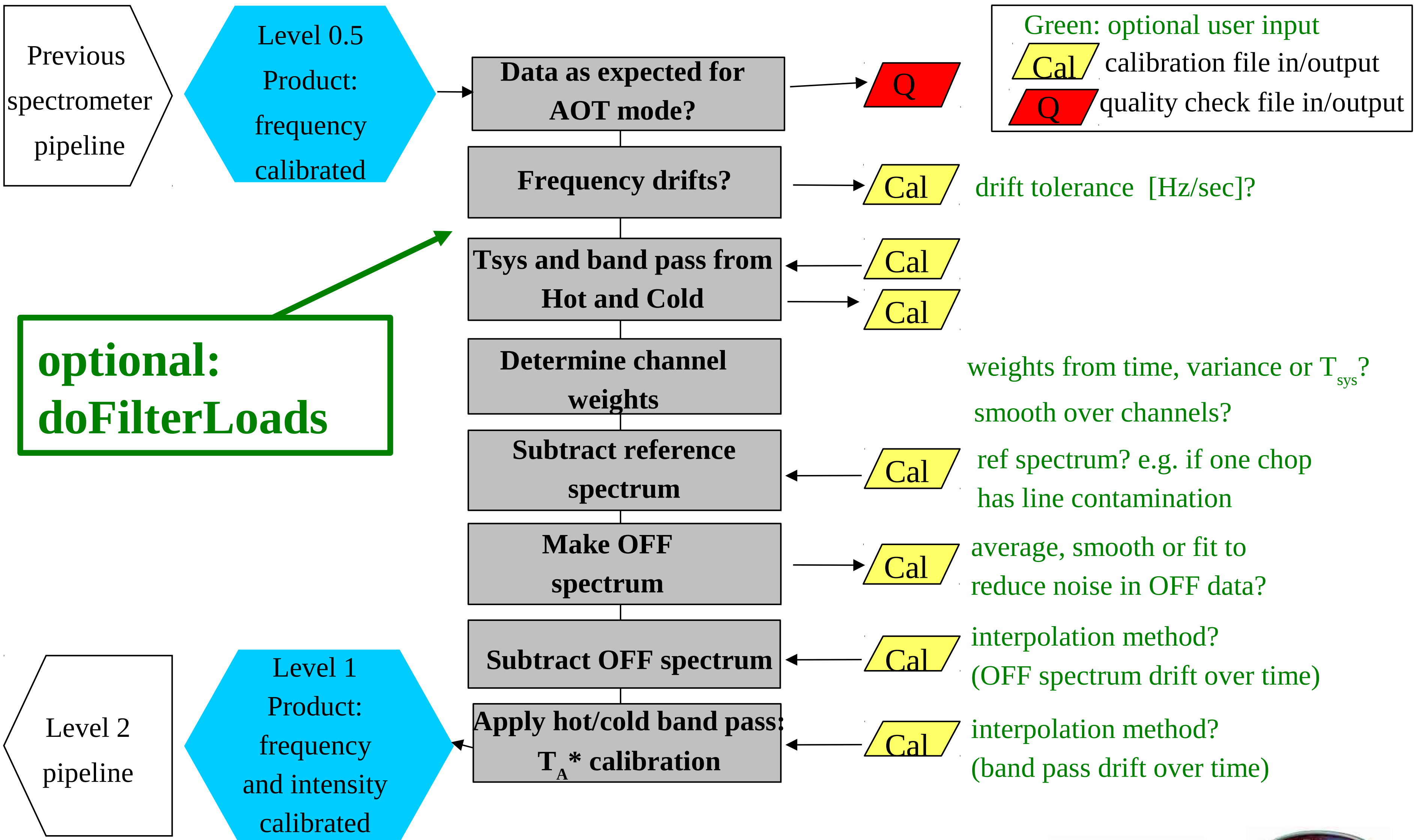
The optional pipeline step doFilterLoads is intended to remove the 92 and 98 MHz waves.

To isolate these waves from the other waves (most importantly the 100 MHz LO-mixer wave), the load spectra are divided by sky spectra.

DoFilterLoads works as follows:

- Compute C_{HBB}/C_{OFF} and C_{CBB}/C_{OFF}
- Remove waves using cubic spline or FFT
- Multiply C_{OFF} back in to get modified C_{HBB} and C_{CBB}

Generic Pipeline (Level 0.5 -> 1)



Removing CBB and HBB Waves

Advantage FilterLoads method over sine wave fitting level 2 spectra:

- Load spectra not 'contaminated' by emission or absorption lines, making the method more reliable and objective.

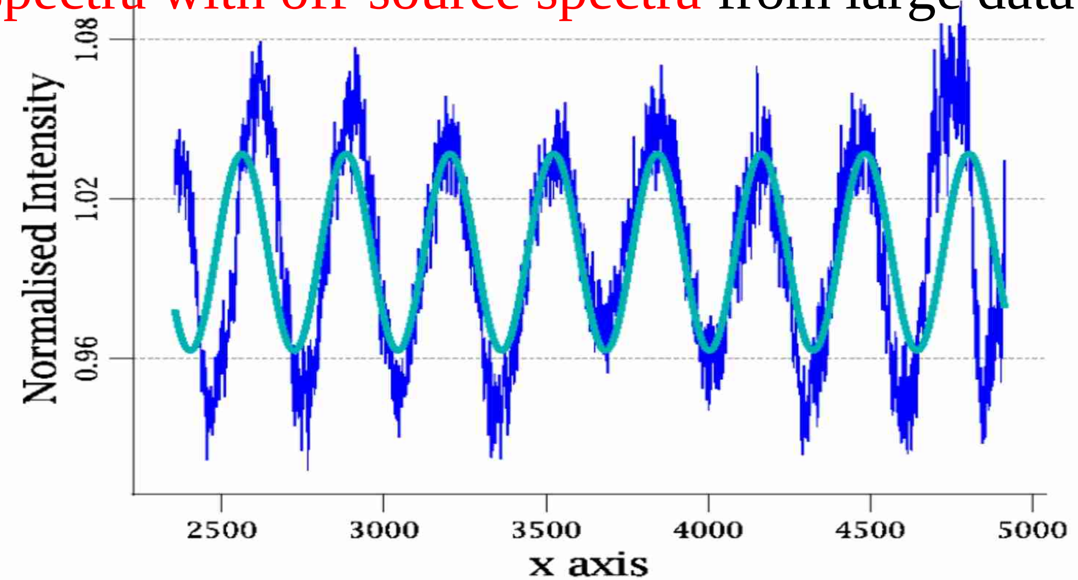
Caveats and side effects:

- Method only works if off sky measurements available.
- Outer edges of sub-bands will show artifacts because of smoothing
- Method only works if load waves stronger than 100 MHz LO-mixer wave, i.e.,
  - For bands 1 and 2
  - Central part bands 3 and 4 (LO-mixer wave dominates in the outer parts)
  - Little effect on band 5-7
  - For any remaining waves fitHifiFringe can be applied at level 2/2.5

Advanced Topic 2:
HEB bands 6 and 7 Standing Wave Removal

HEB Standing Waves

- Strongest standing waves in level 2 spectra HEB bands 6 and 7 have electronic, not optical origin: impedance difference mixer to amplifier.
- Optical waves also present in bands 6 and 7, but are weaker.
- HEB waves are not sine waves. Exact shape and amplitude depends on power on detectors.
- fitHifiFringe can approximate HEB waves with 2-3 sine waves (task has band-dependent defaults)
- More reliable removal of HEB waves requires special treatment: match on-source spectra with off-source spectra from large database

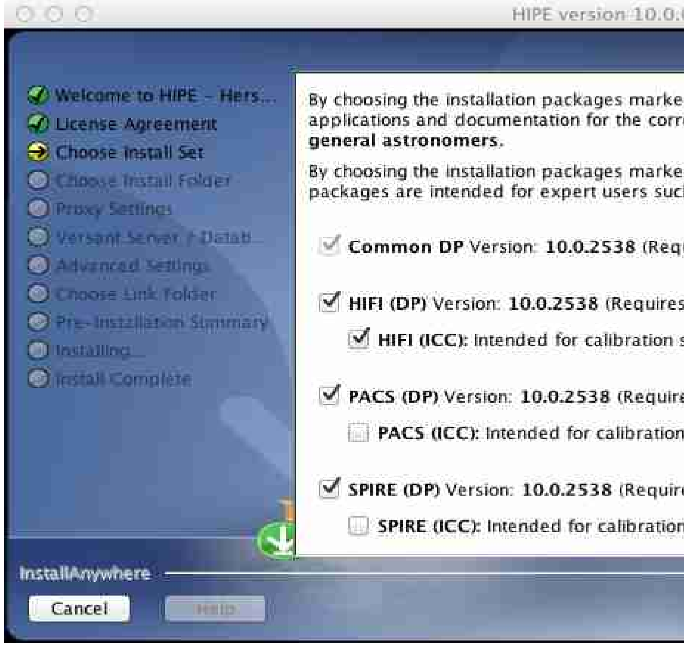


HEB Matching Technique

A script is available that subtracts best-matching off spectra, i.e., it modifies the standard HIFI pipeline:

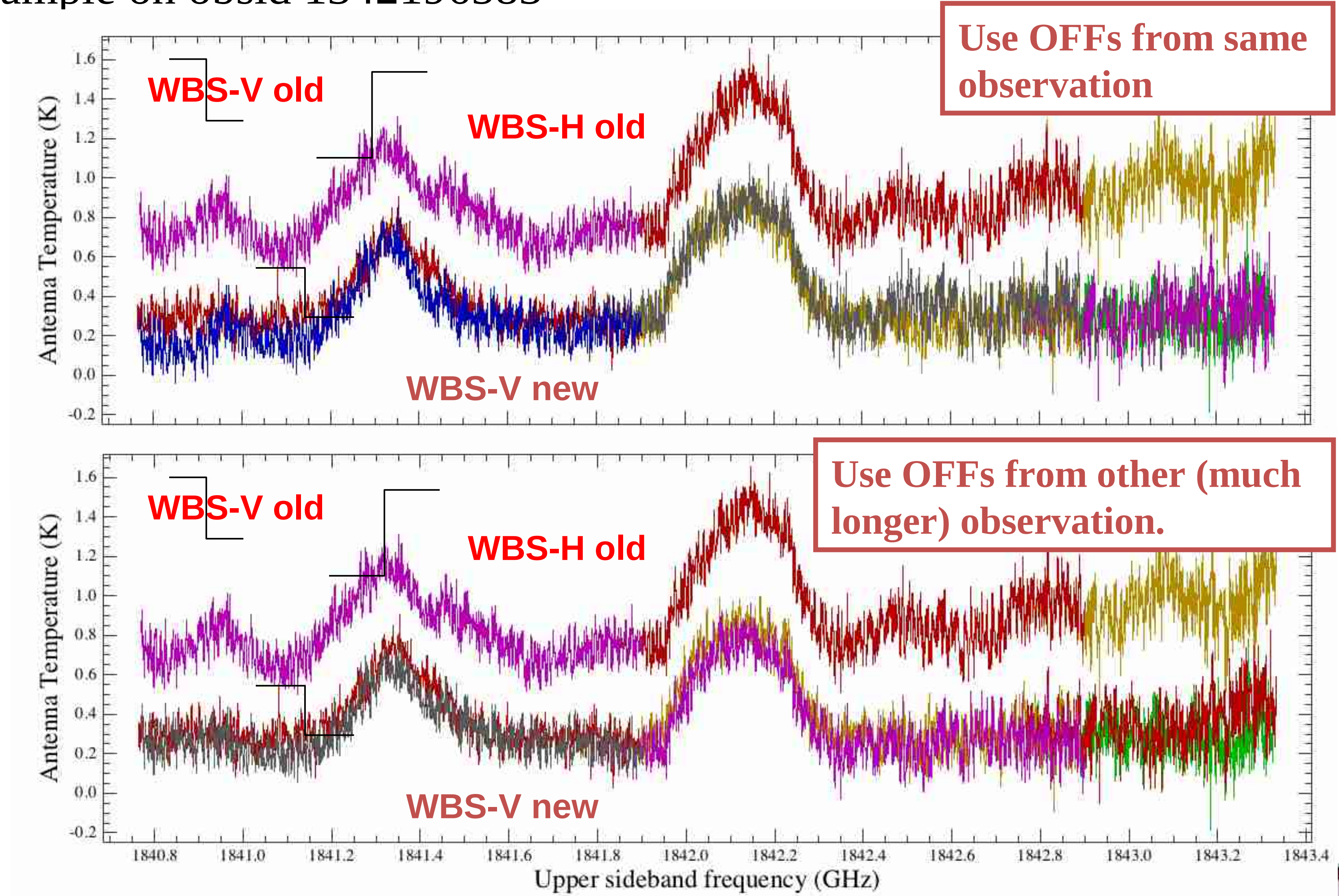
scripts/hifi/scripts/users/engineering/HEBStWvCatalogCorrection.py

- Not a standard task. Only available in expert branch of HIPE.
- This is advanced users script!
- Currently covers only DBS mode. Script for OTF mode available on request.
- Continuum is conserved
- Version 0.53 available in HIPE 11.0
- Possible to use ad hoc public obsids to feed a larger set of OFF spectra – use data from the same mixer band ! (6/7). Band 6A: obsid 1342190743 was made public.



HEB Matching Technique

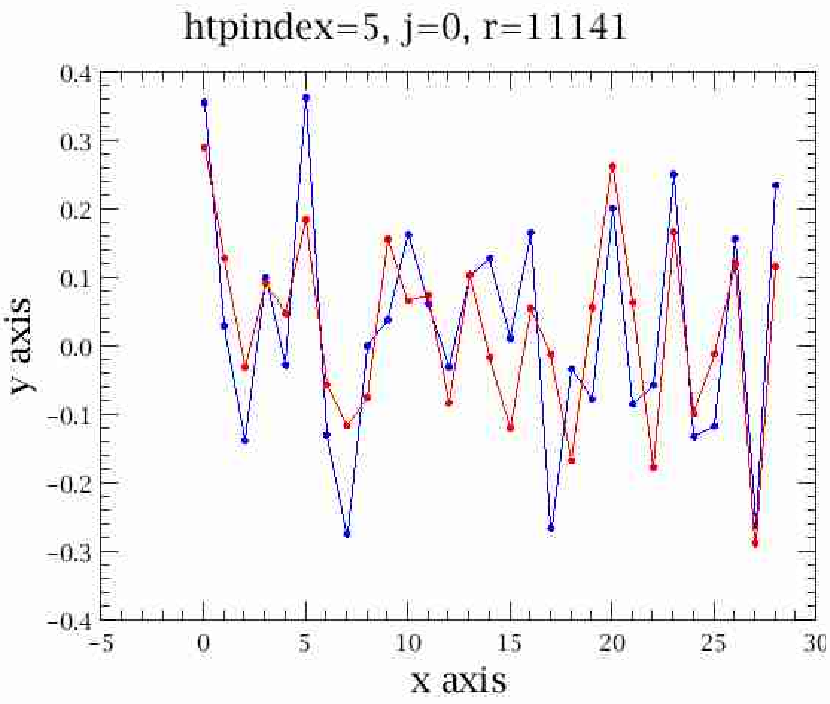
Example on obsid 1342196583



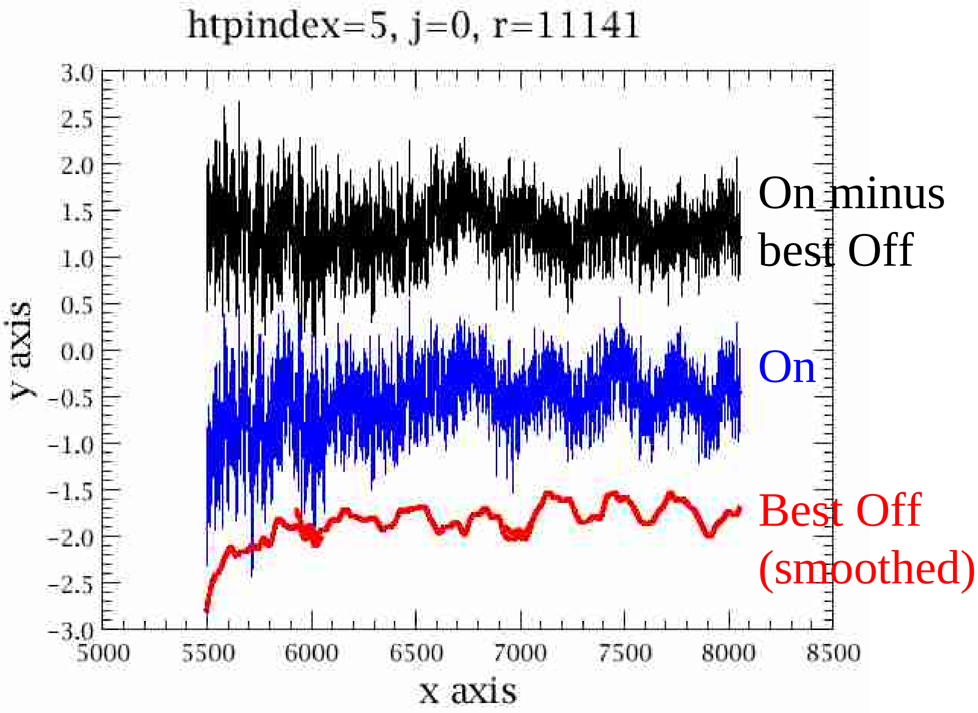
HEB Matching Technique

For every on-source spectrum, these plots are produced (if option plot=True):

Heavily smoothed on and off spectra (smoothing speeds up processing)



Before and after spectra



HEB Matching Technique

Tips on how to run script:

- This script is for advanced users! Contact NHSC if you need help!
- Copy script to local directory: scripts/hifi/scripts/users/engineering/HEBStWvCatalogCorrection.py
- There is no manual, but script contains point-by-point instructions, including which parameters to modify:
  - Which observation to process
  - Which observation to use for reference OFF spectra
  - Create or use calibration database?
  - Process WBS-H or WBS-V (not both!)
  - Plot intermediate results?
- Script runs pipeline, which requires local calibration database hifi-cal
- Run script line by line using HIPE's single green arrow

HEB Matching Technique

Future developments:

- Make script work for all observing modes
- Data mining for more standing wave templates
- Speed up processing and reduce memory usage by storing spline points in table instead of spectra.
- Make task more user-friendly