RFI Identification and Automatic Flagging

Urvashi Rau

VLA Data Reduction Workshop

8 – 12 April 2013

NRAO, Socorro

Outline :

RFI at the VLA + Online Flags

Automatic Flagging Options and Strategies

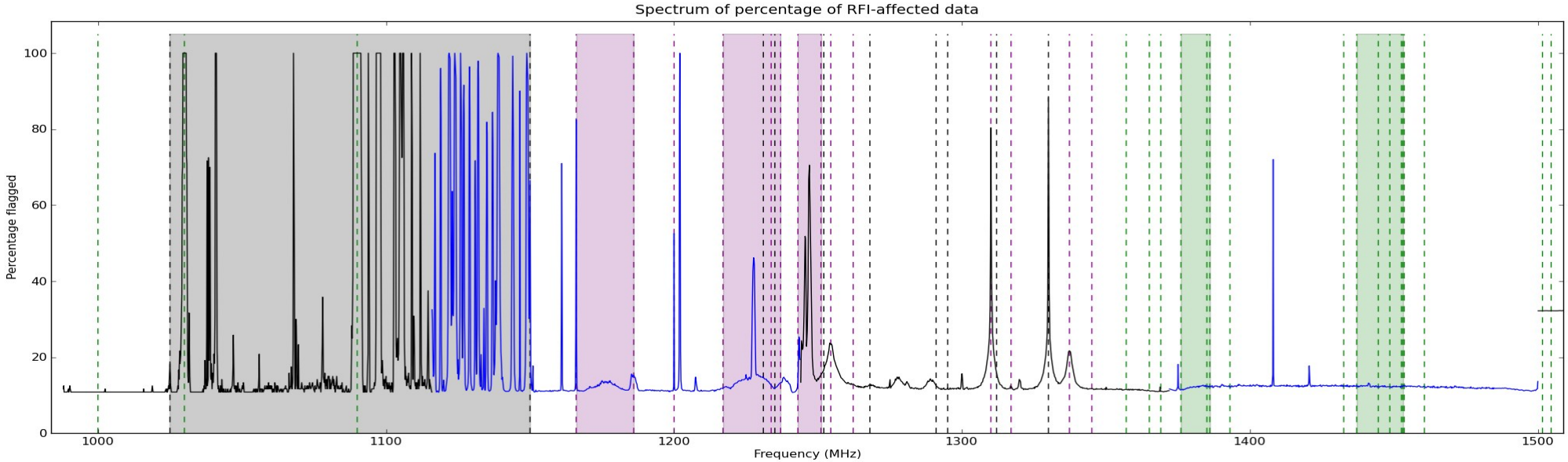
Some examples of what works when....

(31 slides)

RFI at the VLA + Online flags

Radio Frequency Interference : Terrestrial, Satellites, etc..

At L-Band, can use ~500 MHz with very rough flagging, ~800 MHz if done carefully.
RFI is less at higher frequencies. At X-band and beyond, RFI is minimal



https://science.nrao.edu/facilities/vla/observing/RFI

Online + Deterministic Flags :

– operator logs of known bad antennas and time-ranges / slew-time, etc...
– shadowing between antennas (elevation-dependent)
– elevation-dependent flags
– known frequency ranges with bad RFI
– exact zeros (from the correlator)

Flagging Tasks - flagdata + flagcmd

mode='manual' or mode='unflag' # Use MS-selection syntax to pick subsets to flag/unflag

mode = 'quack' # Flag data at the beginning or end of a scan (operate on selected data).

mode = 'elevation' # Flag data between specified elevation limits (operate on selected data)

mode='shadow' # Flag baselines with shadowed-antennas

---

mode = 'clip' # Threshold-based flagging on data-expressions (ABS_RR, ABS_I, REAL_I, etc)

mode='tfcrop' # Automatic flagging : Find outliers on the 2D time-frequency plane

mode='rflag' # Find outliers based on statistics of real and imag visibilities over time

mode='extend' # Grow/extend flags around existing ones.

---

mode='summary' # Count existing flags and return a python-dictionary of counts per antenna, spw, etc

mode = 'list' # Supply a list of flag commands, built out of parameters of any other mode.

Run-modes : " apply / calculate " + runtime "display" of data before and after flagging, and reports.

Flagging Modes : tfcrop

Detect outliers on the 2D time-freq plane.

For each baseline,

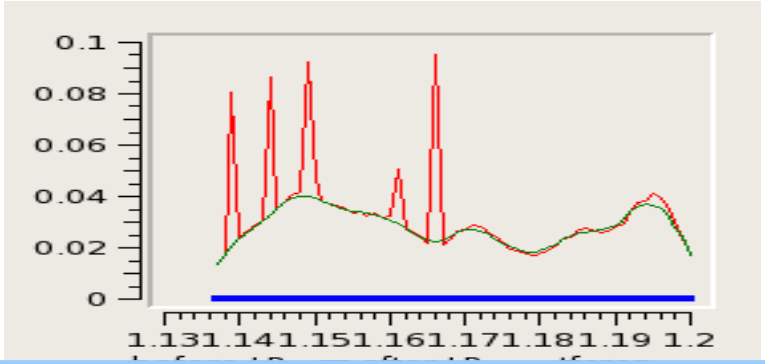
(1) Average visibility amplitudes along time dimension to form an average spectrum

(2) Calculate a robust piece-wise polynomial fit for the band-shape at the base of RFI spikes

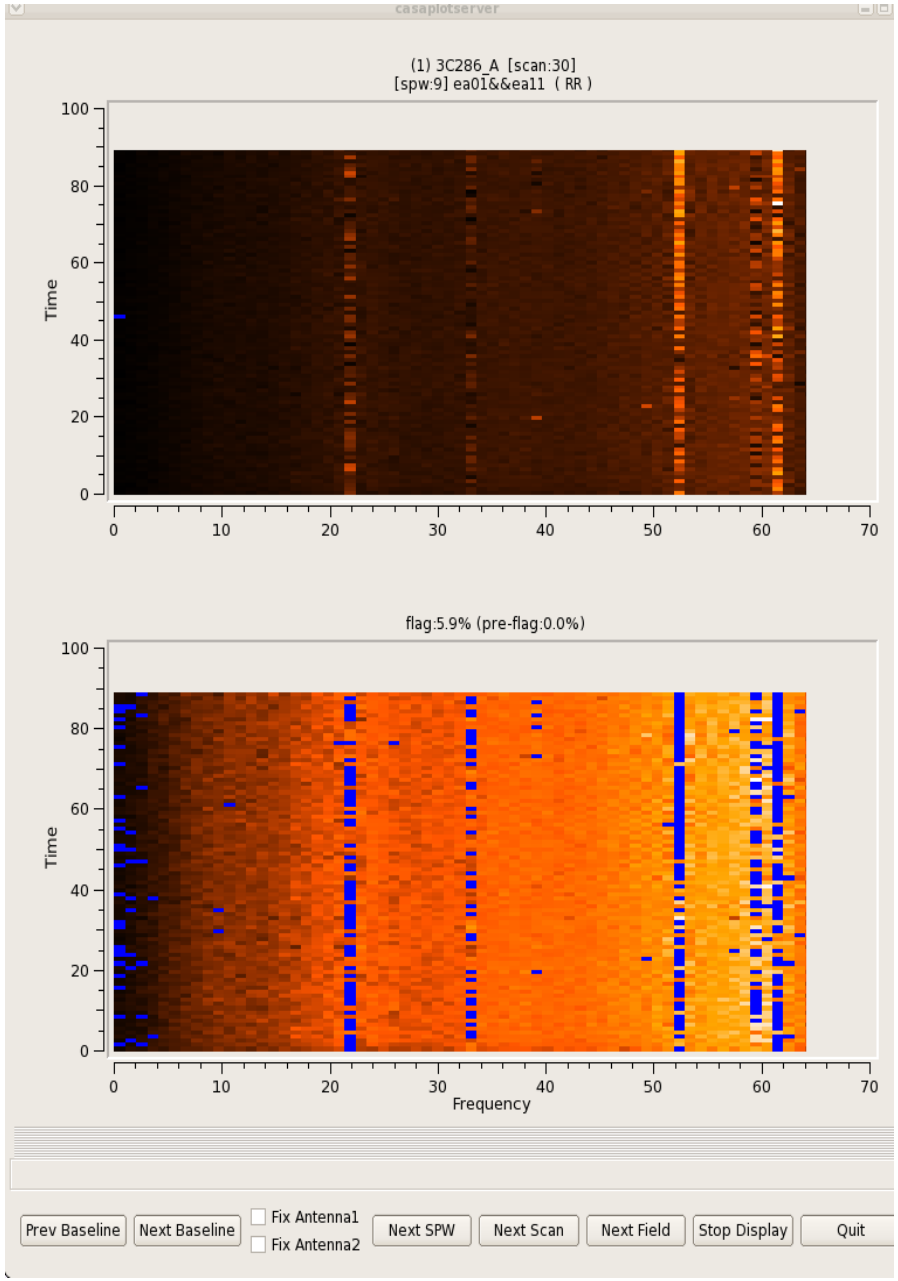
– Calculate 'sigma' of (data - fit).

(3) Flag points deviating from the fit by more than N-sigma

(4) Repeat (1-3) along the other dimension.



- Calculates and applies flags in one pass through the MS
- Can operate on un-calibrated data and bandshapes
- Flag extension can be done via mode='extend'



Flagging Modes : tfcrop

ntime = 'scan' or '2.0min' or 120.0 # Chunks of time across which to extend flags.

combinescans = True/False # if ntime > scan, join multiple scans

datacolumn = 'data', 'corrected', 'model', 'residual'

correlation = 'ABS_ALL' # Correlation expression. Augmented MS-selection syntax.
Expressions : 'ALL', 'RR','LL', 'I', 'Q','U','V', 'WVR'
Functions : 'ABS', 'ARG', 'RE', 'IM', 'NORM'
Ex : 'ABS_RR,LL,V'

timecutoff, freqcutoff = 3.0 # Scale-factor for deviation from fit, to compute thresholds

timefit, freqfit = 'line' or 'poly' # Fit a straight line, or piecewise polynomial (coming : only mean)

maxnpieces = 7 # maximum number of pieces in the polynomial fit (only for 'poly')

flagdimension = 'freqtime' # Directions in which to calculate stats 'timefreq','time','freq'

usewindowstats , halfwin # Options to use sliding-window statistics 'std' and 'sum'

Flagging Modes : rflag

(A close copy of RFLAG from AIPS (E.Greisen, 2011))

Detect outliers based on a sliding-window rms

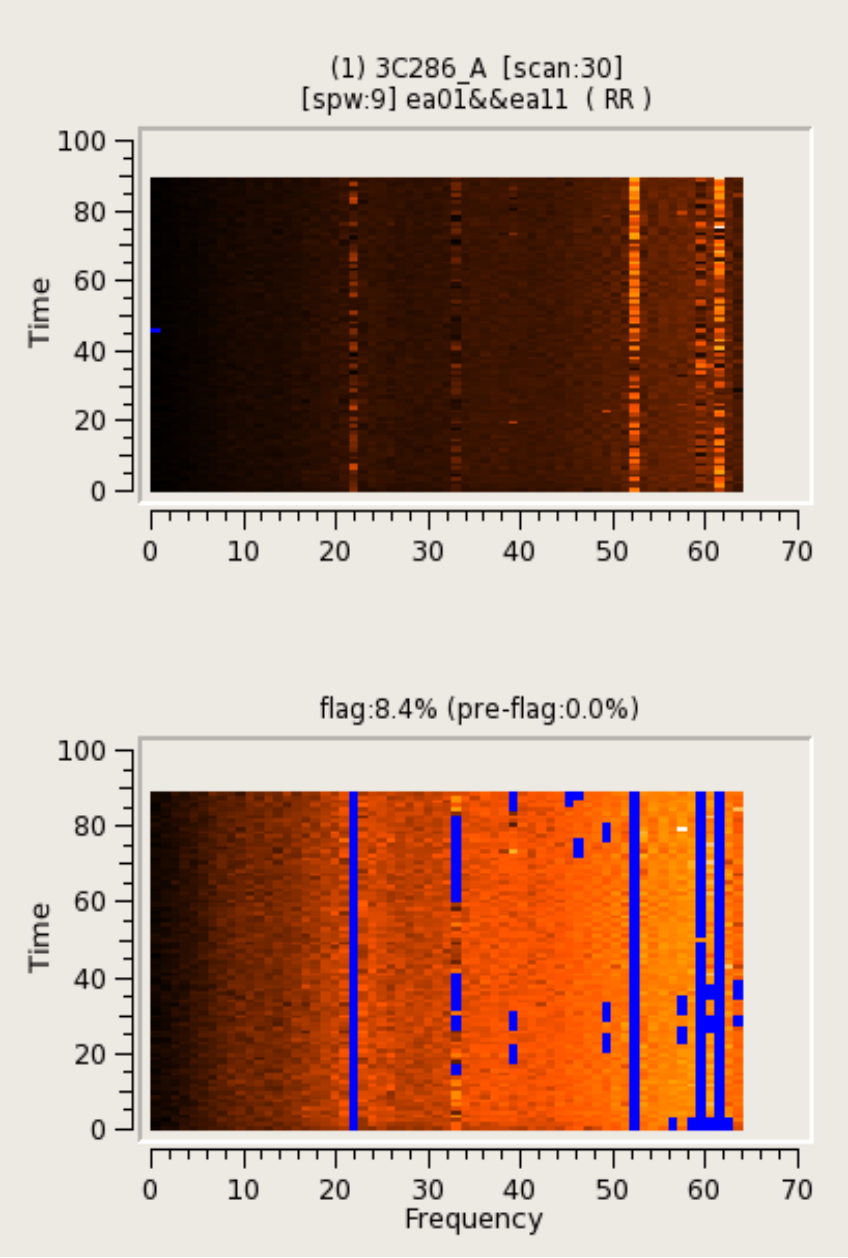
(1) Time-Analysis (for each channel)

(1a) Calculate local RMS of real and imag parts of visibilities within a sliding time window.
(1b) Calculate the median RMS across time windows, deviations of lo cal RMS from this median, and the median deviation
(1c) Flag if local RMS > N x (medianRMS + medianDev)

(2) Spectral-Analysis (for each timestep)

(2a) Calculate Avg of real and imag pars of visibilities and their RMS across channels.
(2b) Calculate the deviation of each channel from this Avg, and the median-deviation.
(2d) Flag if deviation > N x medianDev

- Can be run with 1 or 2 passes through the data (calc + flag)
- Requires calibrated data.
- Flag extension can be done via mode='extend'
- Can flag short baselines if there is extended emission
- Good for low level noisy RFI



Flagging Modes : rflag

```
ntime  = 'scan' or '2.0min' or 120.0   #   Chunks of time across which to extend flags.

combinescans = True/False              #    if ntime > scan, join multiple scans

datacolumn = 'data', 'corrected', 'model', 'residual'

correlation = 'ABS_ALL'                # Correlation expression. Augmented MS-selection syntax.
                                         Expressions : 'ALL', 'RR','LL',  'I', 'Q','U','V', 'WVR'
                                         Functions : 'ABS', 'ARG', 'RE', 'IM', 'NORM'
                                         Ex : 'ABS_RR,LL,V'

winsize  = 3                           #   size of sliding time window [ fparm(1) ]

timedev = [ [ 0, 2, 0.0534 ] ]         #   time-series noise estimate [ noise ]
freqdev  = [ ]                         #   spectral noise estimate [ scutoff ]

timedevscale = 5                       #   threshold scaling for timedev [ fparm(9) ]
freqdevscale = 5                       # threshold scaling for freqdev [ fparm(10) ]

spectralmin = 1e+6                     # flag whole spectrum if freqdev > spectralmin [ fparm(6) ]
spectralmax = 0.0                      # flag whole spectrum if freqdev < spectralmax [ fparm(5)] ]

outfile  = 'thresholds.txt'            #  text file to hold output thresholds from rflag.
```

Flagging Modes : extend

Example :

Flag only RR,

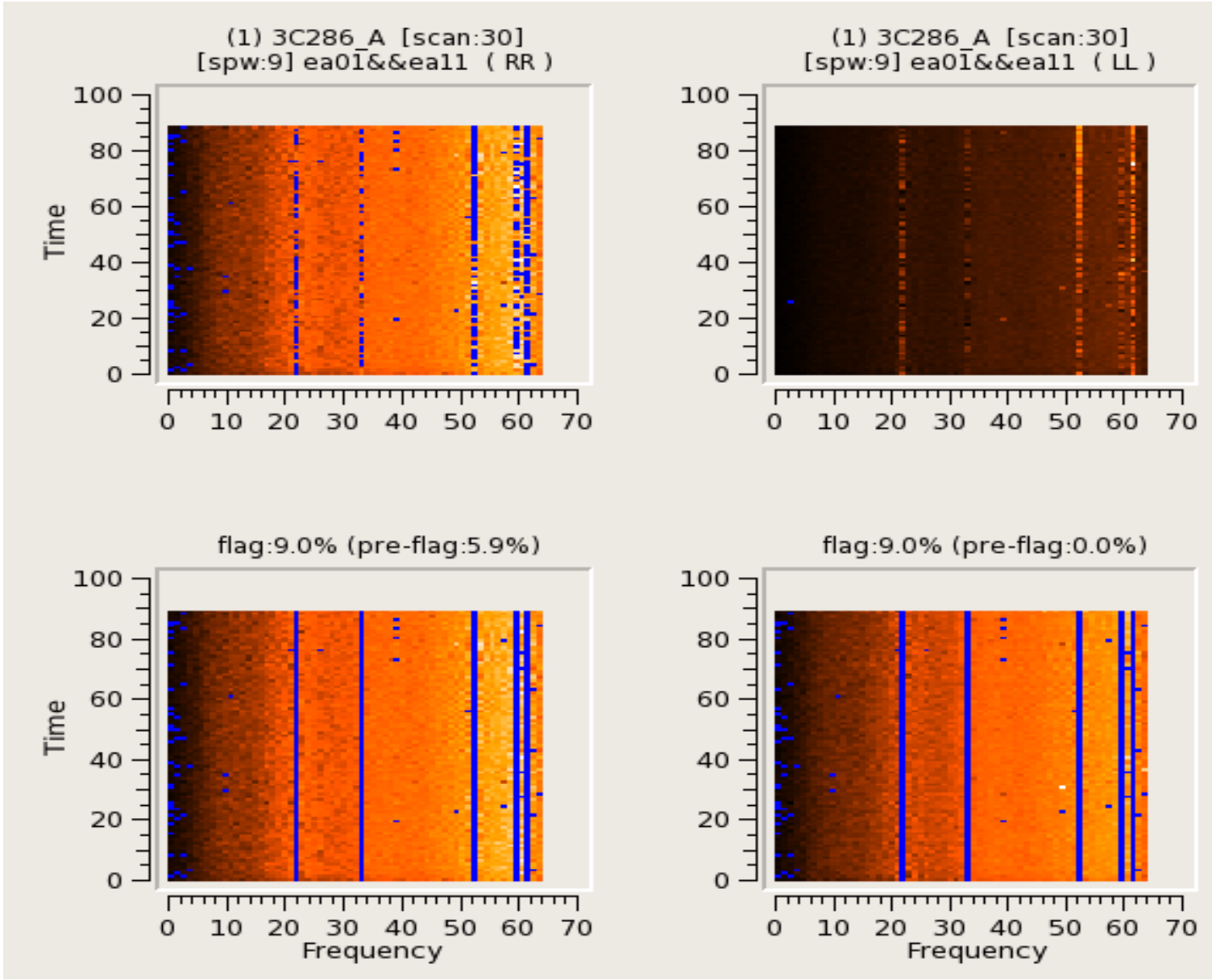
+

Growtime = 30.0

+

Extendpols = True

Useful as a follow-up option after rflag or tfcrop.



Flagging Modes : extend

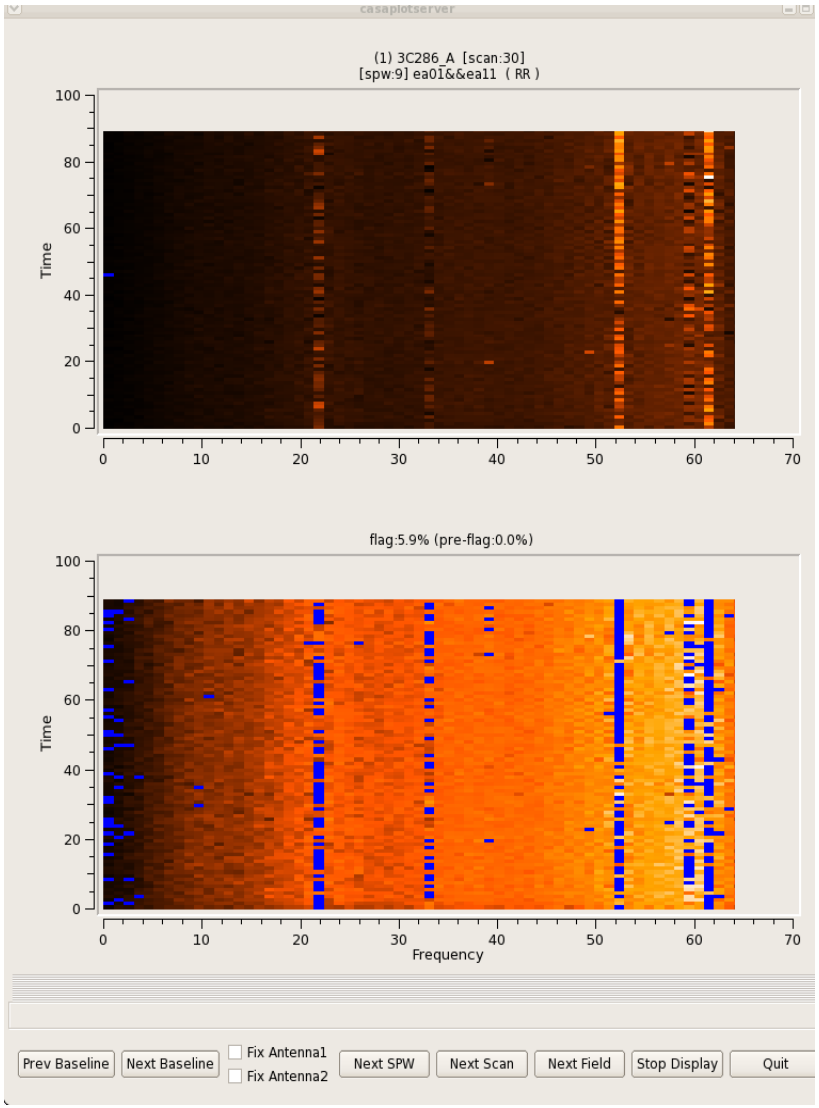
Extend flags along time, frequency, polarizations (and baselines). Operates on selected data

```
ntime  = 'scan' or '2.0min' or 120.0   #   Chunks of time across which to extend flags.

combinescans = True/False              #    if ntime > scan, join multiple scans

extendpols = True/False                #    Extend across all selected correlations

growtime = 80.0                        #   For each channel, flag entire timerange (ntime)
                                             if more than 80% of the data is flagged.

growfreq = 80.0                        #   For each time, flag selected channels (in current
                                             SPW)  if more than 80% of the data is flagged.

growaround = True/False                #   On the 2D time/freq plane, if more than 4
                                             surrounding points are flagged, flag it.

flagneartime = True/False              #   Flag points before after every flagged point in time
flagnearfreq  = True/False             #   Flag channels before/after every flagged one.

extendants = 80%  (coming soon)        #   For each channel/time, flag all baselines if more
                                             than 80% are flagged.
```

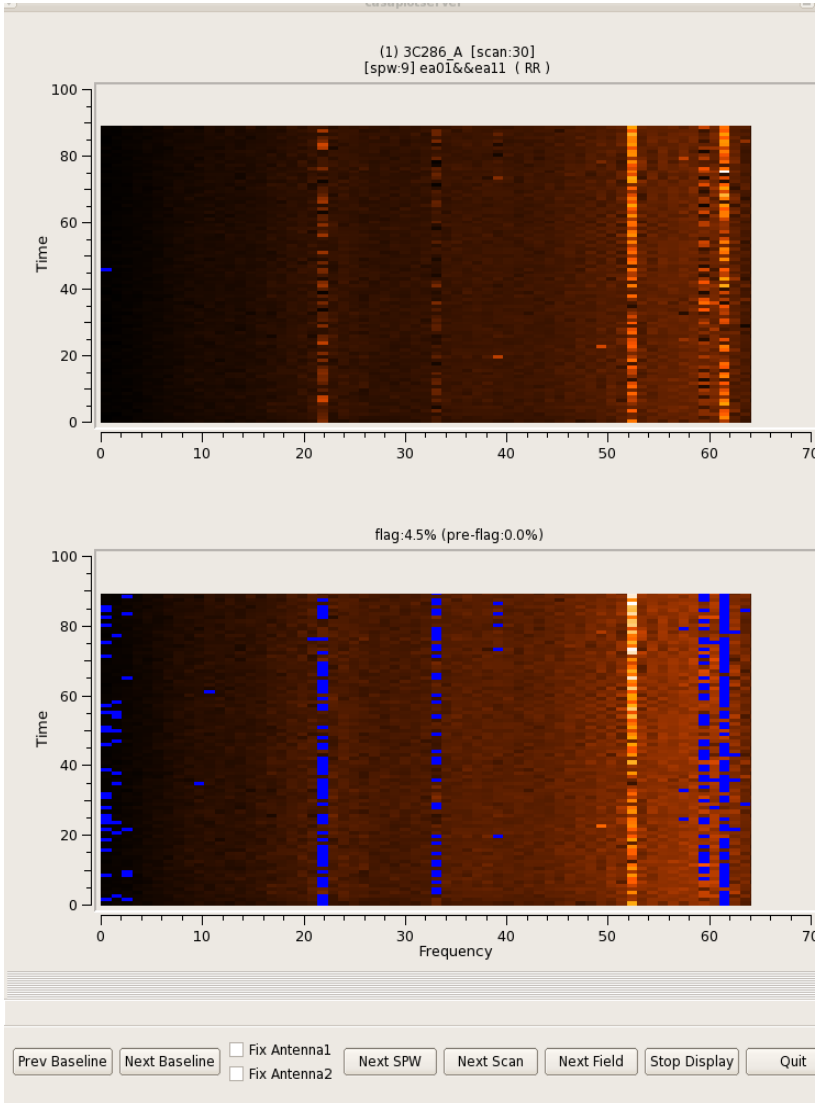
- Protect your spectral-lines by de-selecting them in the spw-selection parameters for 'extend'.

Protecting spectral-lines, during automatic flagging

spw = ' 9 '



spw = ' 9 : 0~45; 53~63 '



This is relevant to 'clip' , 'extend' , 'tfcrop' , 'rflag'

Flagging Modes : list (flagdata, flagcmd)

Supply a list of flag commands, to be run in one single pass through the data.

- Apply large numbers of online flags (all mode='manual') together.
- Combine modes that read the same data, to save on I/O
- flagcmd can read flags from XML files and read/write to the FLAG_CMD MS subtable

Example list of commands (in a text file) :

```
flagdata( vis='xxx.ms', mode='list', inpfile='cmds.txt' )

  mode='manual'   antenna='1&&2'    spw='3'    timerange='12:23:02.3~12.24:45.2'
  mode='manual'   antenna='ea05'    spw='5:50~60'   scan='2~10'
  mode='manual'   antenna='ea01,ea24'    spw='9'    correlation='RR,RL'
  mode='quack'    quackinterval=5.0
  mode='shadow'  tolerance=5.0
```

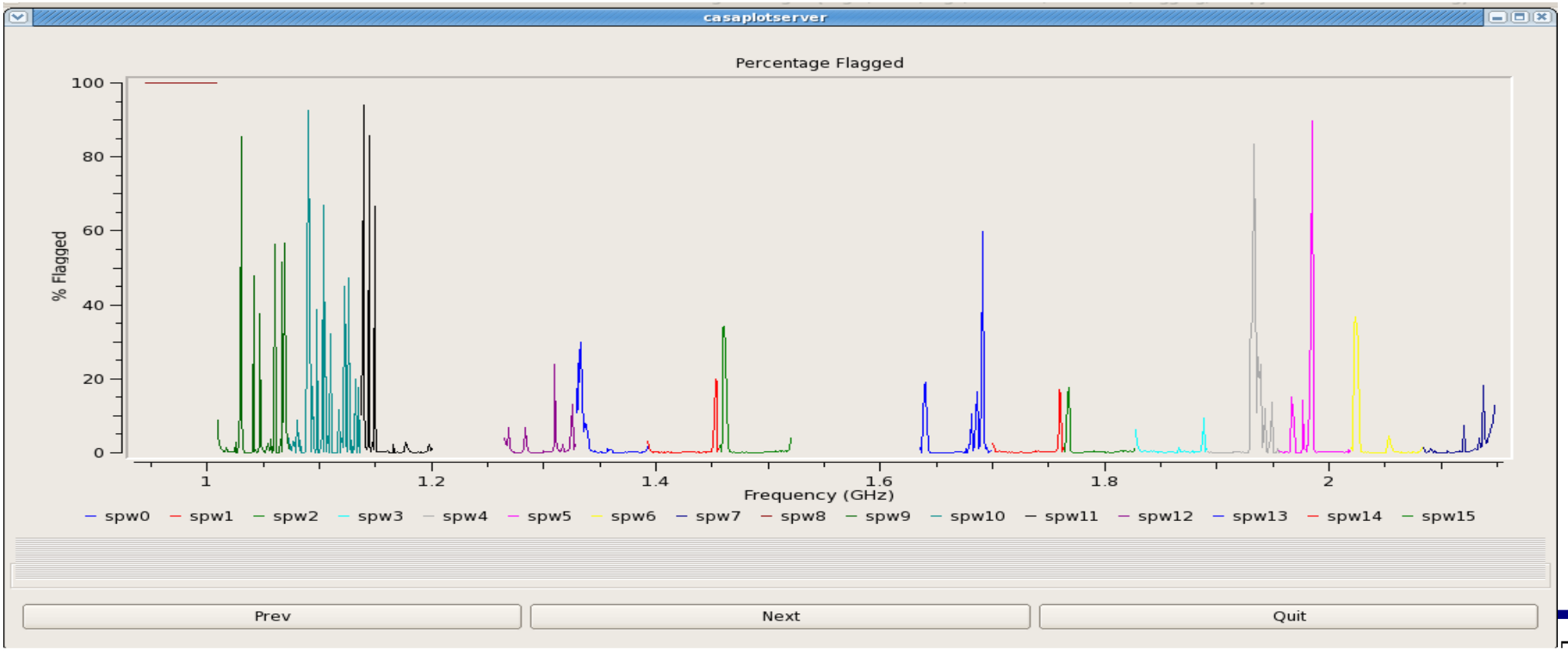
Example command list (interactive input) :

```
flagdata( vis='xxx.ms', mode='list', inpfile=cmdlist )

cmdlist = [ “mode='manual' antenna='1&&2' spw='3'” , “mode='shadow' tolerance=5.0” ]
```

Summary mode - output python dictionary of flag counts

Output : Dictionary of flag counts (vs) antenna, field, spw, channel, correlation, etc...

```
counts = flagdata(vis='xxx.ms' , mode='summary' , display='report' , spwchan=True );

print “Counts per spw : “, counts['spw'];

print “Total % flagged : “ , 100.0 * counts['flagged'] / counts['total']

print “Number of unflagged visibilities in spw 9 : “, counts['spw']['9']['total'] – counts['spw']['9']['flagged']
```



Auto-flagging strategies – all methods need tuning

In general, examine pieces of your data visually, to decide auto-flagging strategy

--- one scan --- few baselines --- all spws ---

All current autoflag algorithms require tuning for best results.

– Rflag calculates statistics across all times, channels, baselines and pols

Tunable parameters : N-sigma thresholds, sliding-window sizes

– TFCrop operates on the time-freq plane, separately per baseline and correlation.

Tunable parameters : N-sigma thresholds, window sizes, types of time/freq fits.

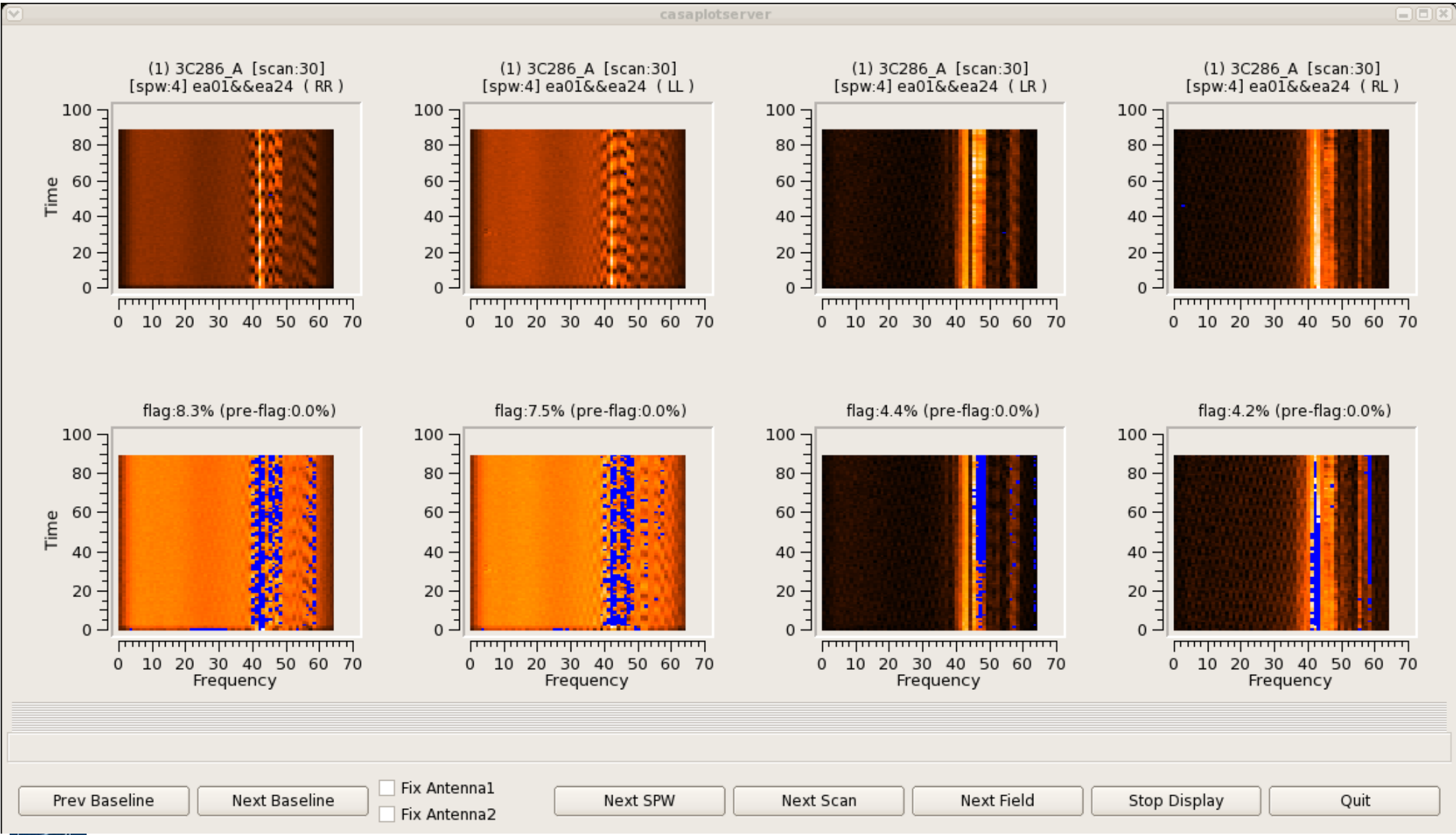
– Flag Extension

Tunable parameters : Grow flags in time and frequency or both,
extend across correlations, etc....

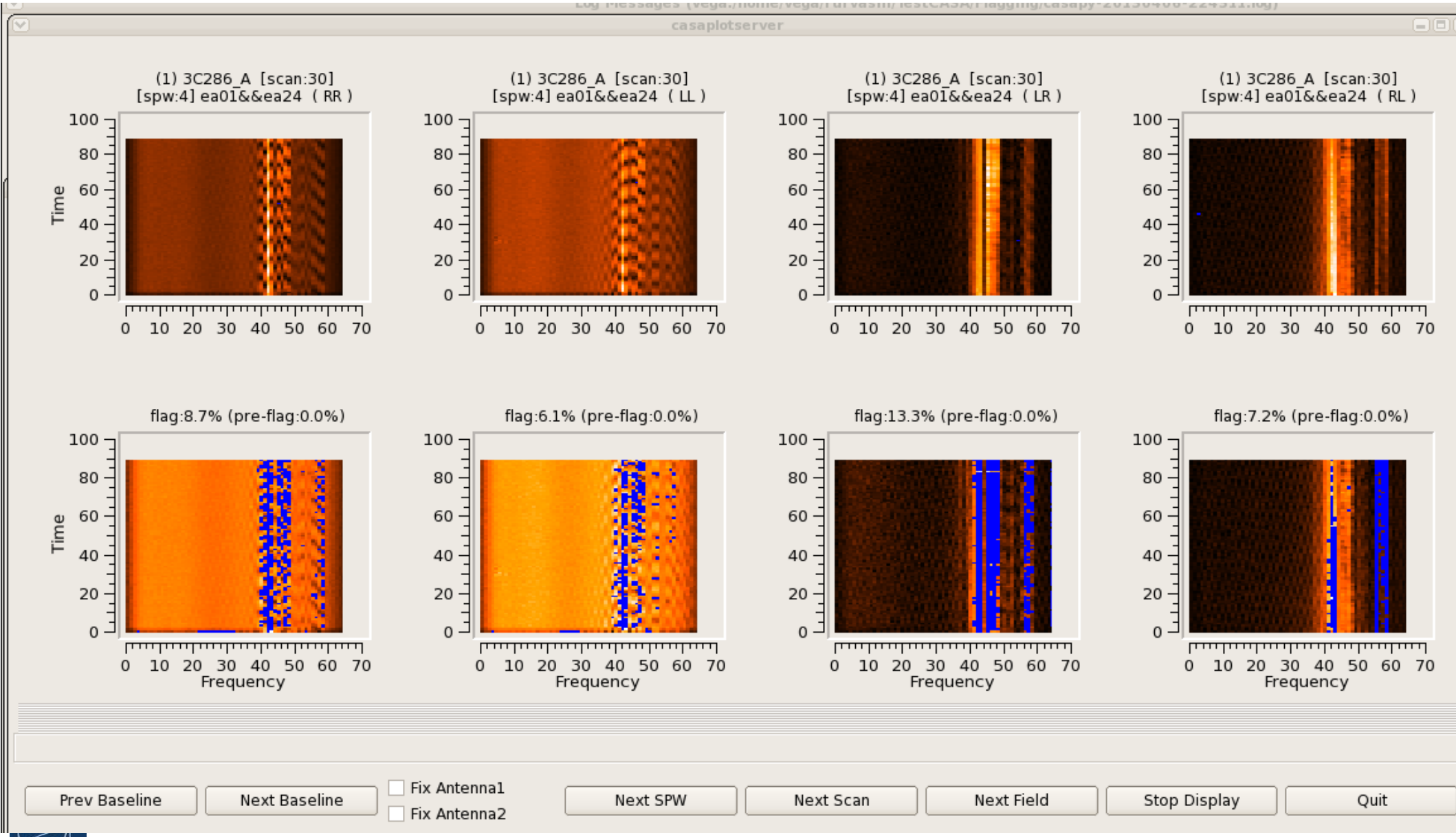
– AO-Flagger from LOFAR : http://sourceforge.net/p/aoflagger/wiki/Home/

Operates on the 2D time-freq plane, based on a robust 2D rubber-sheet fit

```
flagdata(vis='xxx.ms', spw='4'  mode='tfcrop', action='calculate', display='data')

cmdlist = [ “ spw='4'  mode='tfcrop' “ ]
flagdata(vis='xxx.ms', mode='list', inpfile=cmdlist, action='calculate', display='data')
```

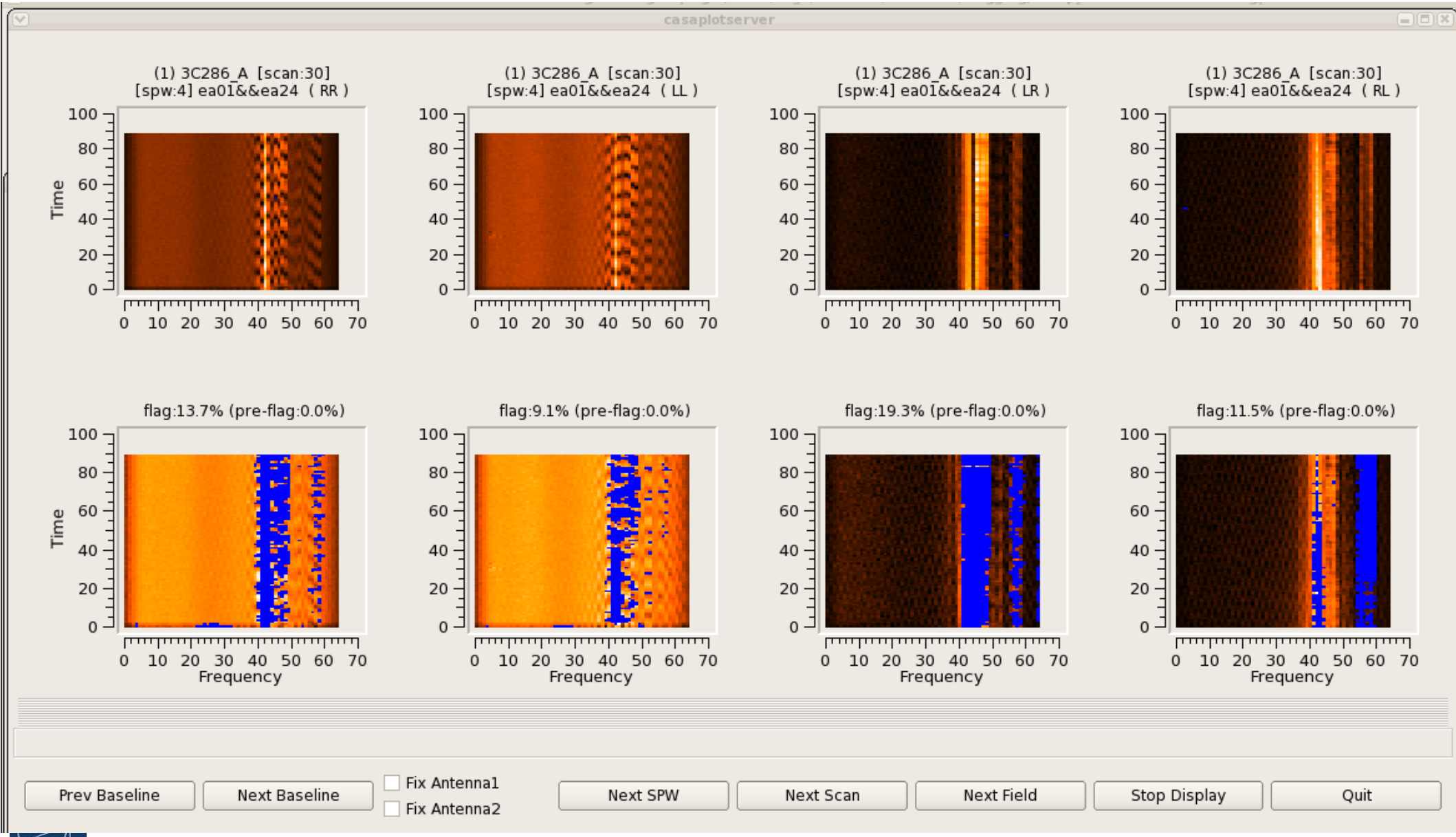


```
flagdata(vis='xxx.ms', spw='4' mode='tfcrop', maxnpieces=4, action='calculate', display='data')

cmdlist = [ " spw='4'  mode='tfcrop'  maxnpieces=4 " ]
flagdata(vis='xxx.ms', mode='list', inpfile=cmdlist, action='calculate', display='data')
```

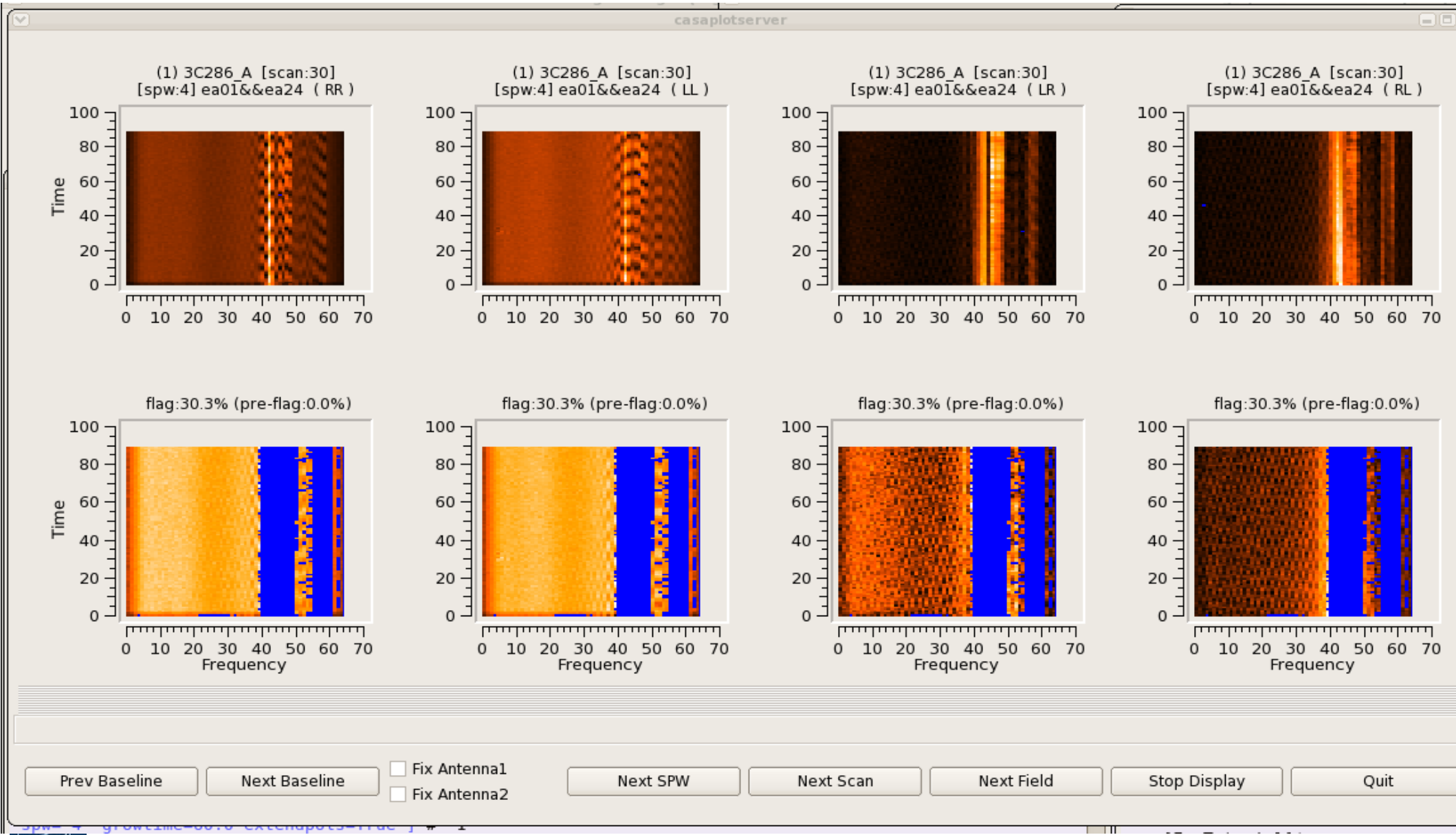


cmdlist = [“ spw='4' mode='tfcrop' maxnpieces=4 usewindowstats='std' halfwin=2 “]

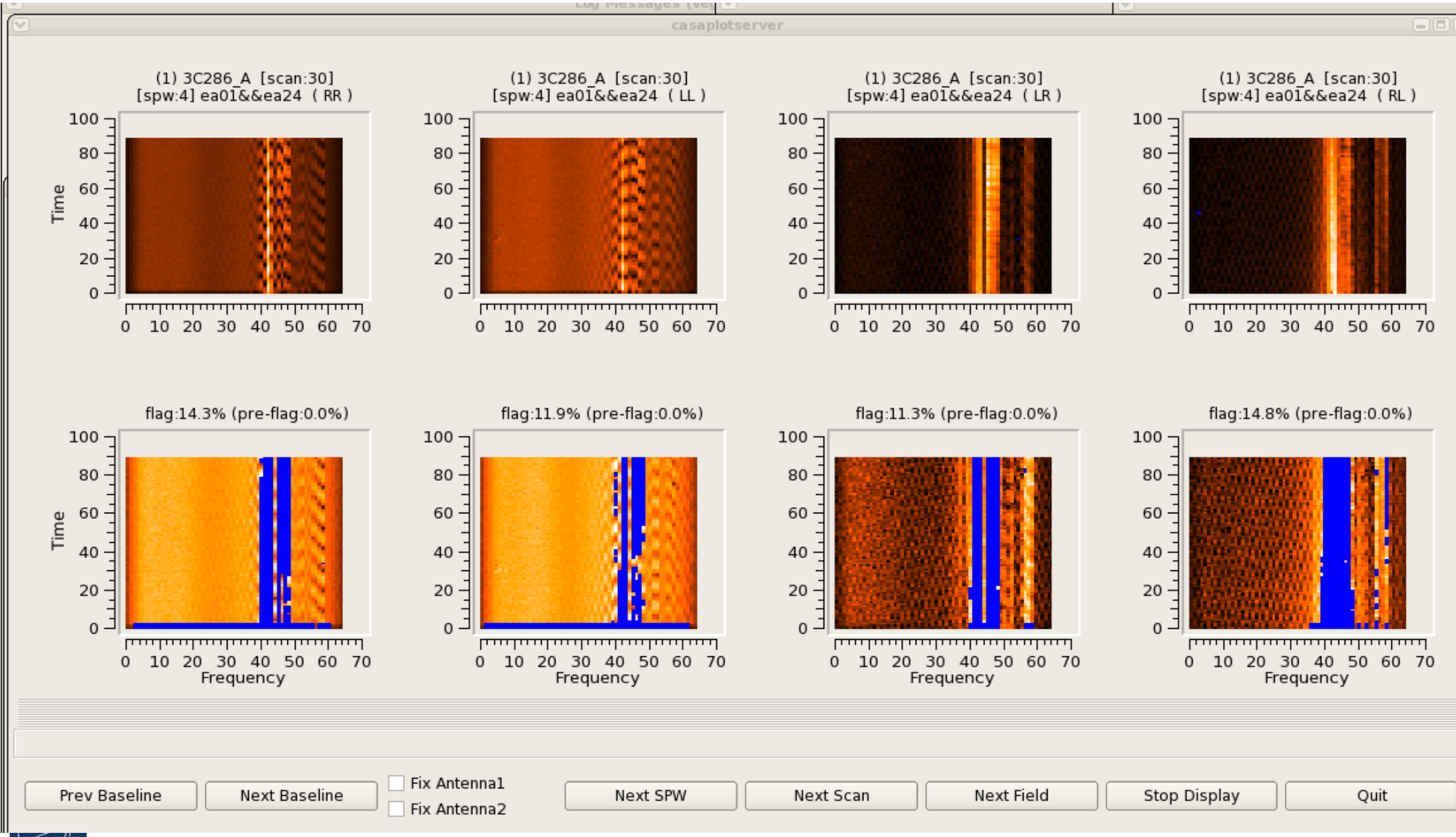
flagdata(vis='xxx.ms', mode='list', inpfile=cmdlist, action='calculate', display='data')



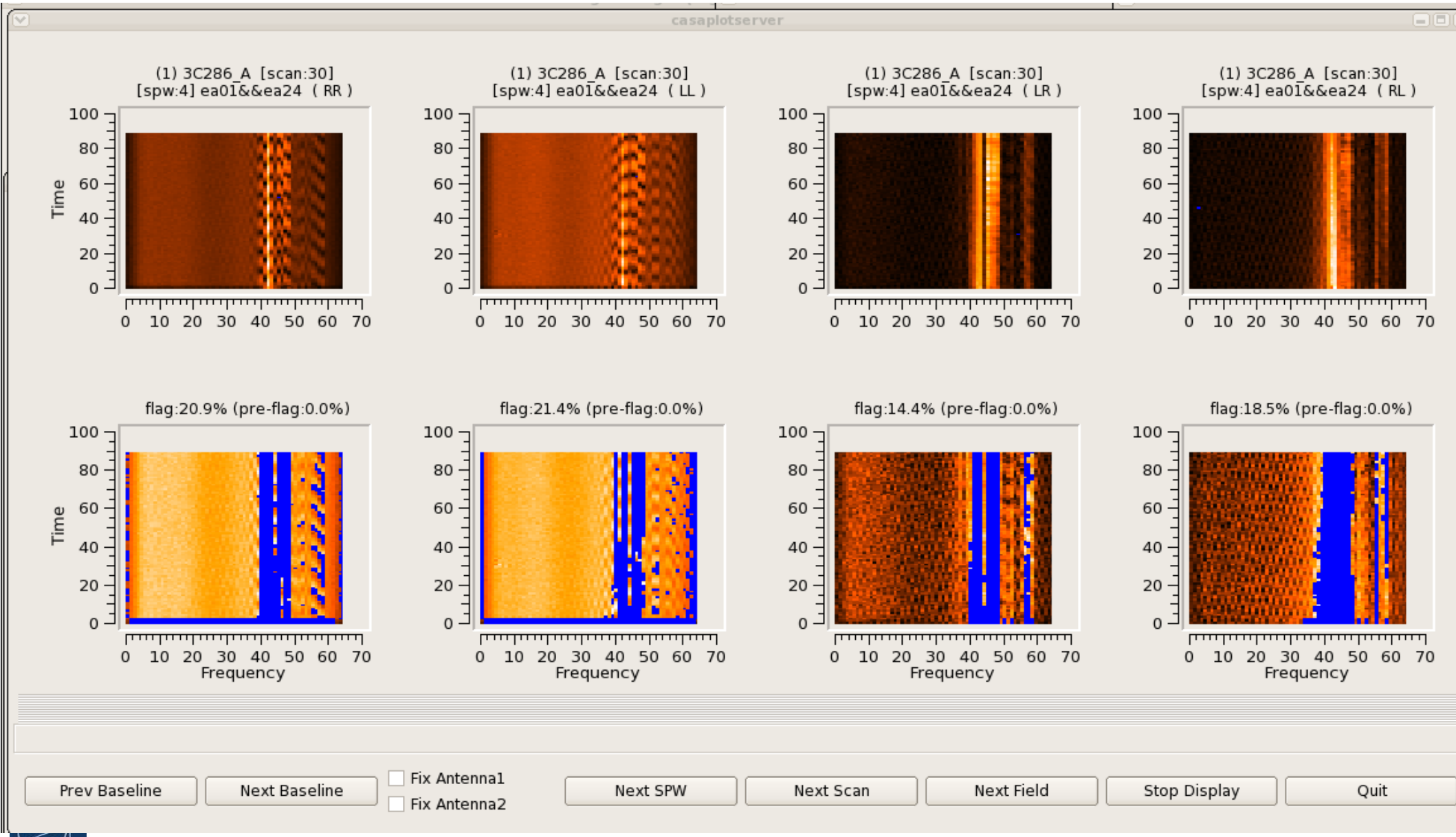
```
cmdlist = [ “ spw='4' mode='tfcrop'  maxnpieces=4 usewindowstats='std' halfwin=2 “,
            “ spw='4' mode='extend' growtime=60.0  extendpols=True “ ]

flagdata(vis='xxx.ms', mode='list', inpfile=cmdlist, action='calculate', display='data')
```



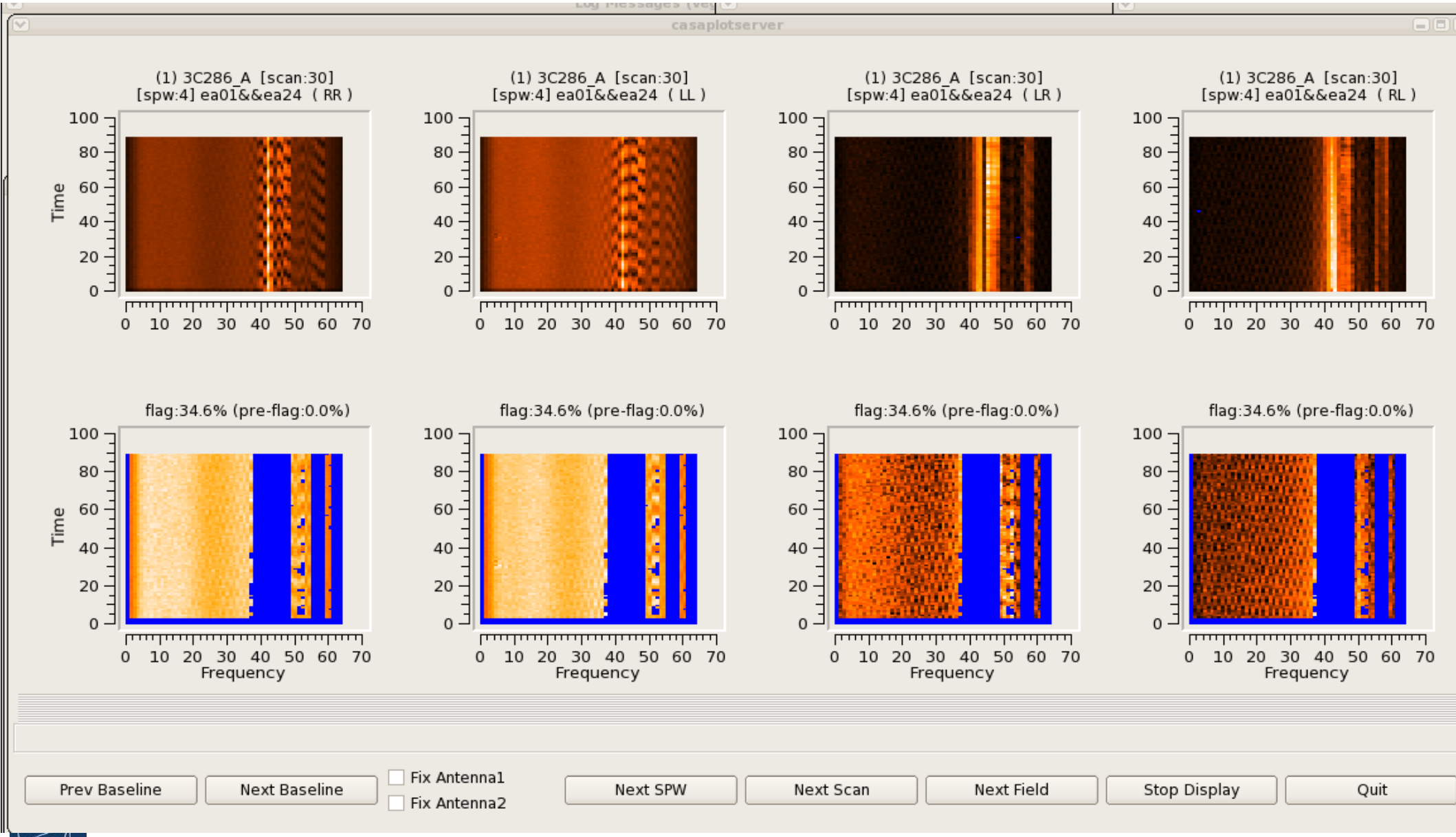
```
cmdlist = [ “ spw='4' mode='rflag' “ ]

flagdata(vis='xxx.ms', mode='list', inpfile=cmdlist, action='calculate', display='data')
```



```
cmdlist = [ “ spw='4' mode='rflag' freqdevscale=4.0 “ ]

flagdata(vis='xxx.ms', mode='list', inpfile=cmdlist, action='calculate', display='data')
```

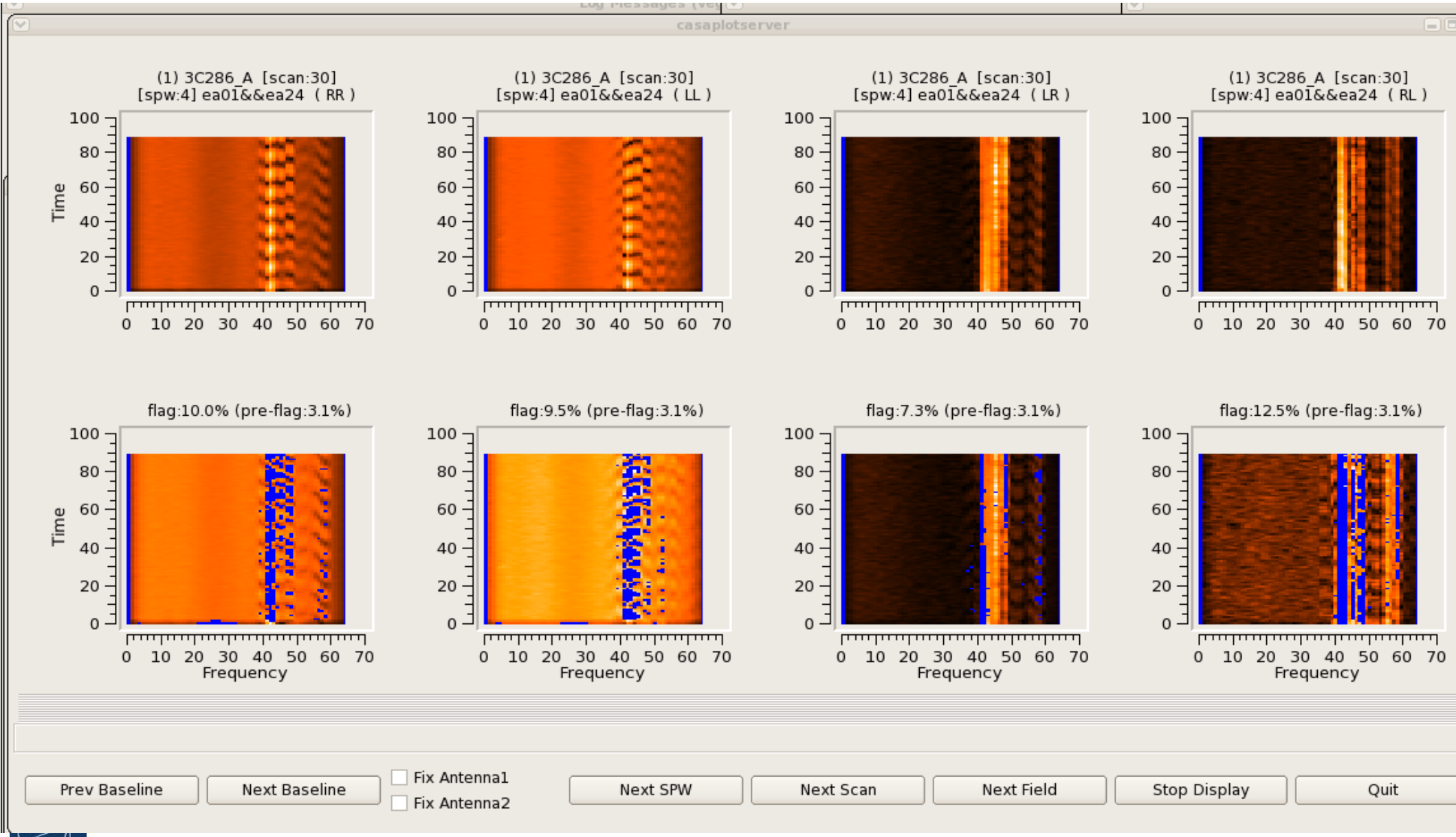


```
cmdlist = [ “ spw='4'  mode='rflag'   freqdevscale=4.0 “,
            “ spw='4'  mode='extend' growtime=30.0  extendpols=True  ]

flagdata(vis='xxx.ms', mode='list', inpfile=cmdlist, action='calculate', display='data')
```



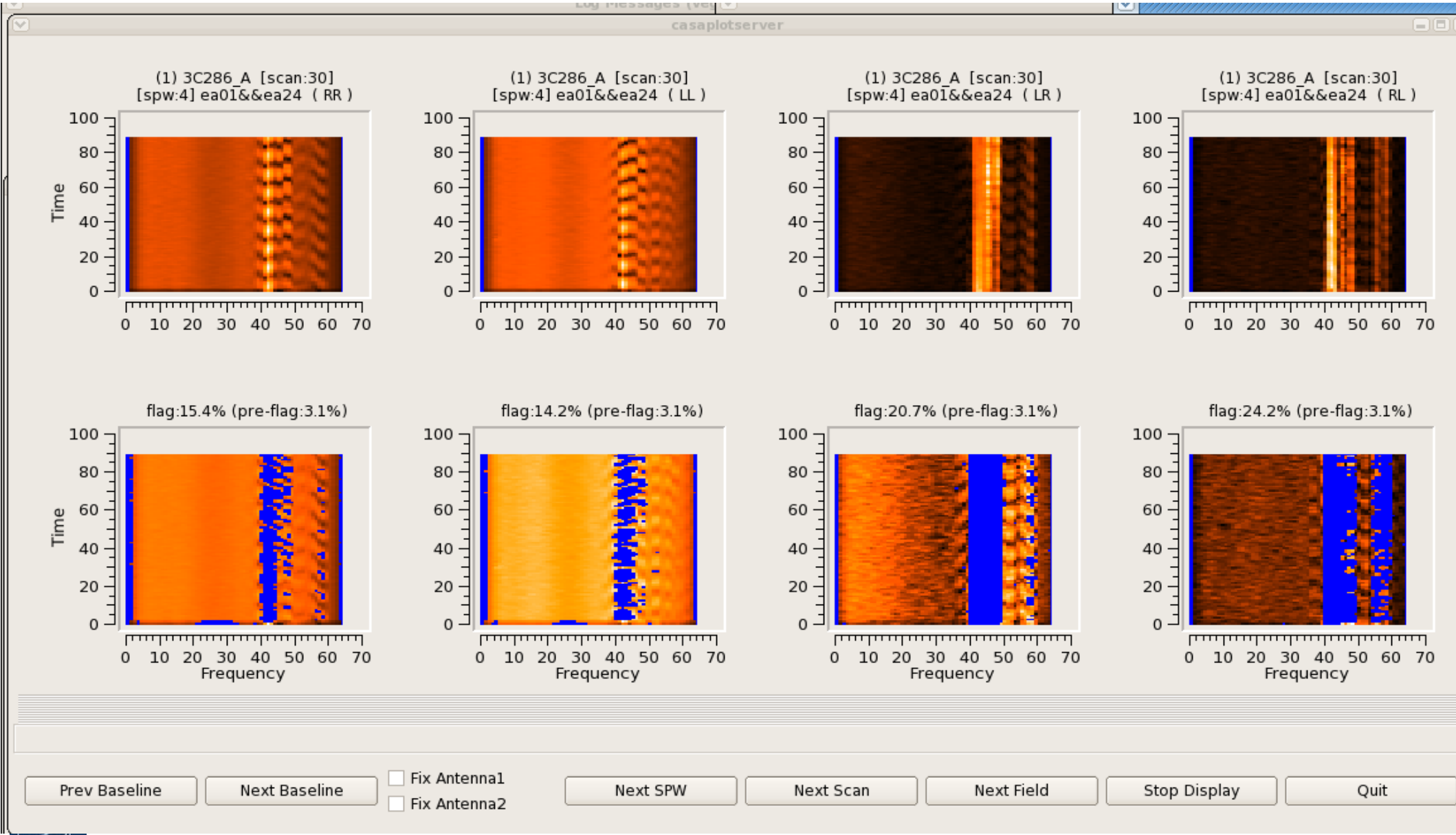
cmdlist = [“ spw='4' mode='tfcrop' “] On Hanning Smoothed Data

flagdata(vis='xxx.ms', mode='list', inpfile=cmdlist, action='calculate', display='data')



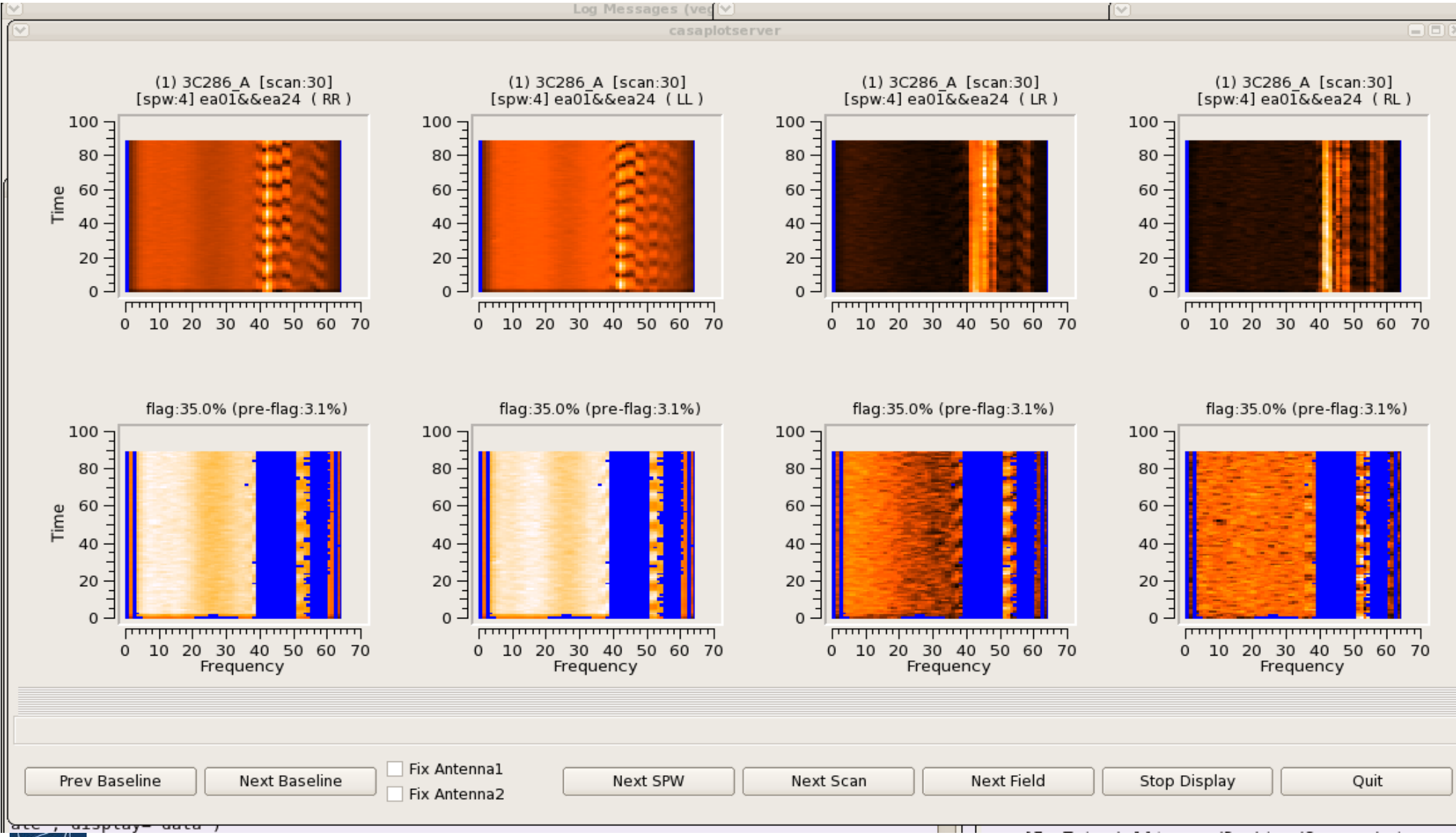
cmdlist = [“ spw='4' mode='tfcrop' usewindowstats='sum' “] On Hanning Smoothed Data

flagdata(vis='xxx.ms', mode='list', inpfile=cmdlist, action='calculate', display='data')



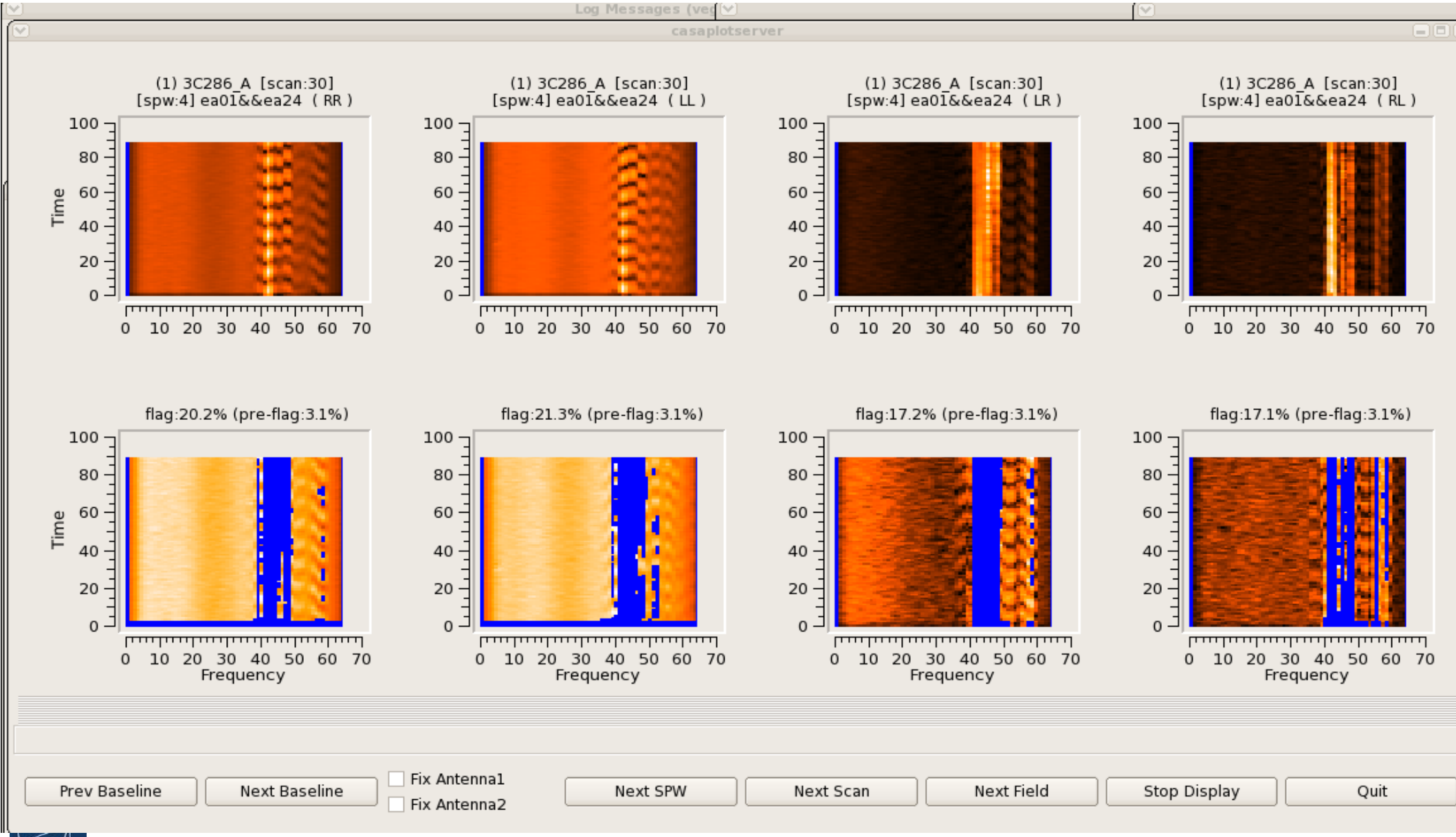
```
cmdlist = [ “ spw='4'  mode='tfcrop' usewindowstats='sum'  “,
            “ spw='4'  mode='extend'  growtime=30.0 extendpols=True “ ]

flagdata(vis='xxx.ms', mode='list', inpfile=cmdlist, action='calculate', display='data')
```

On Hanning Smoothed Data



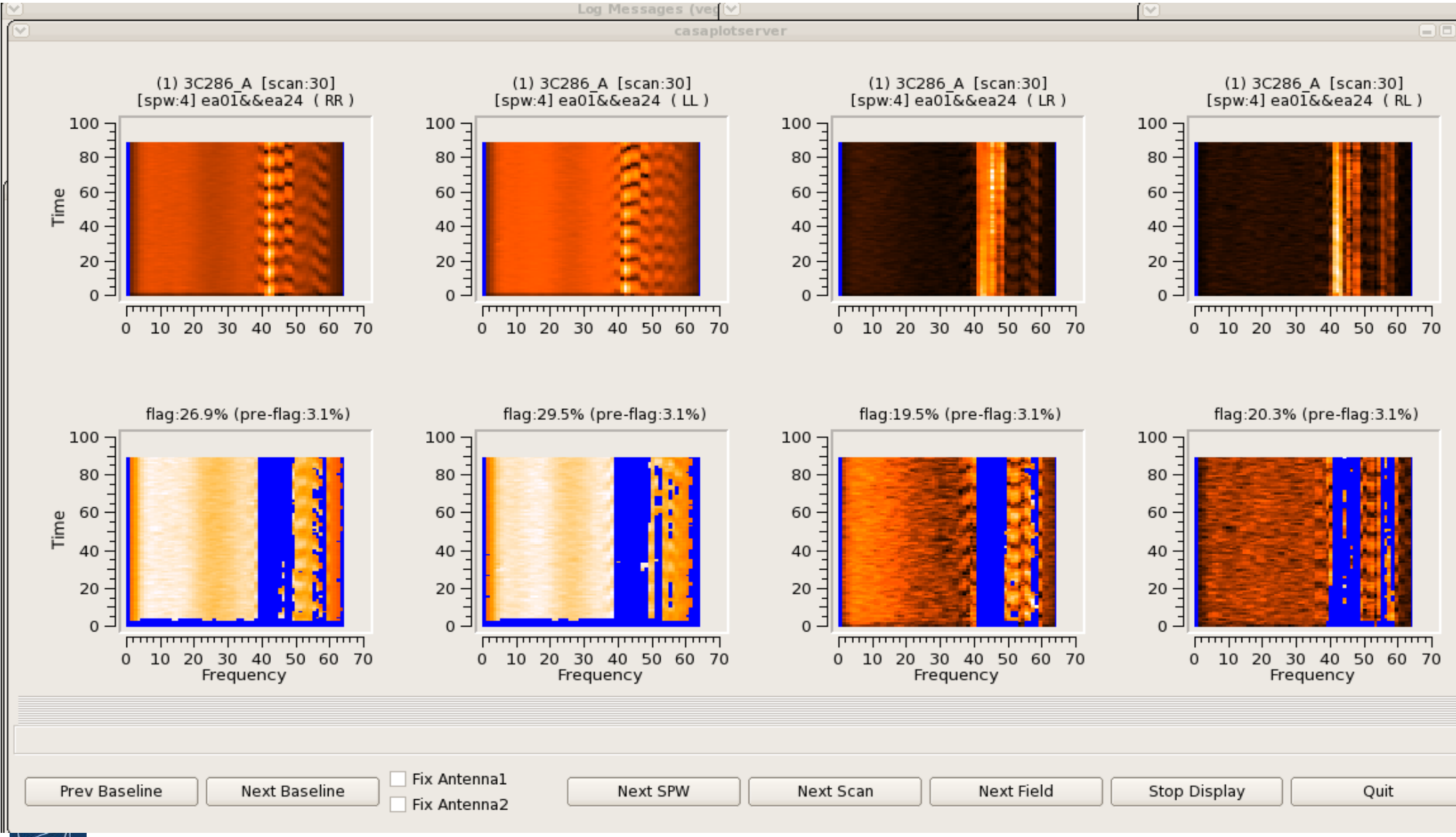
cmdlist = [“ spw='4' mode='rflag' “] On Hanning Smoothed Data

flagdata(vis='xxx.ms', mode='list', inpfile=cmdlist, action='calculate', display='data')



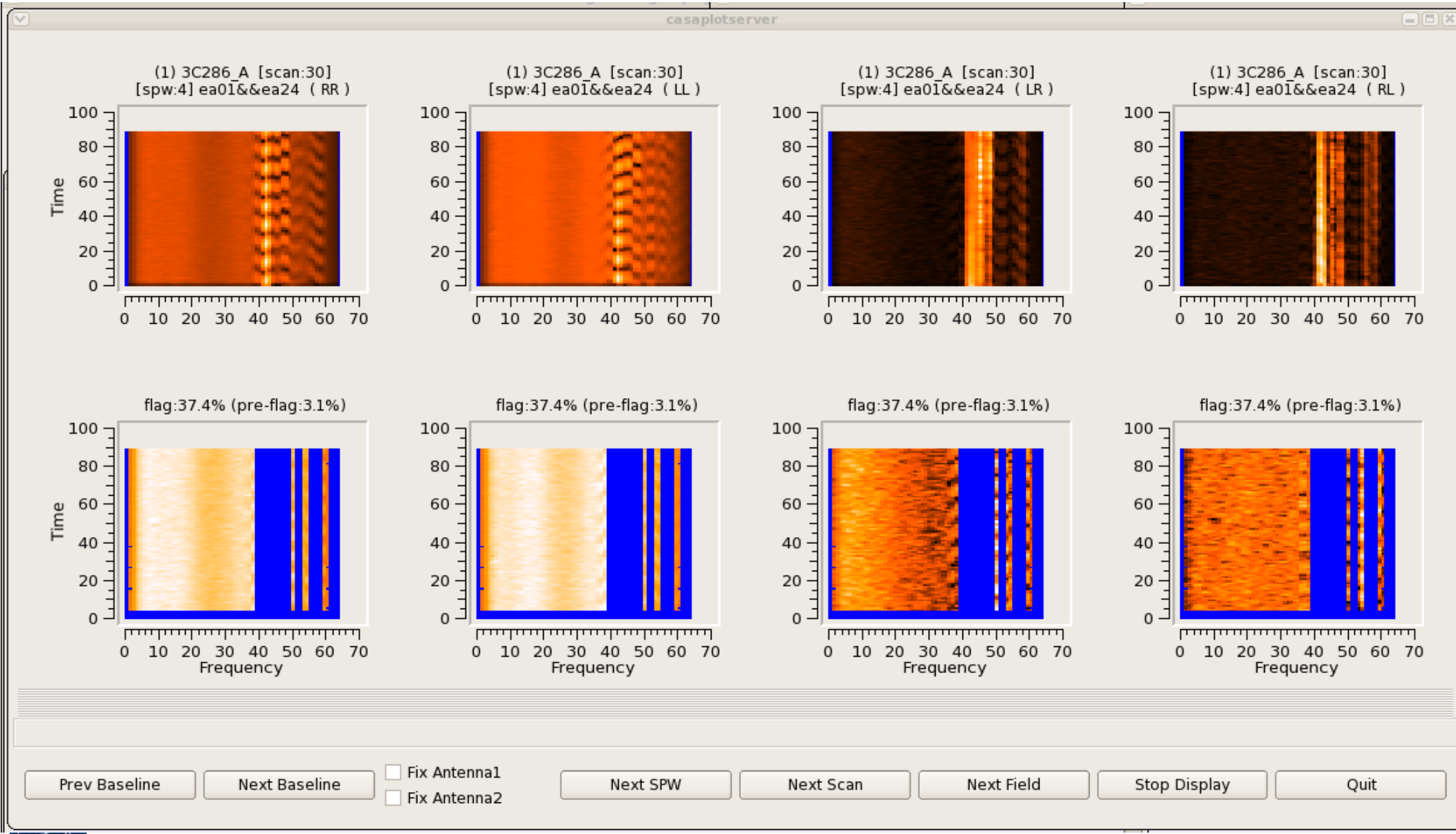
cmdlist = [“ spw='4' mode='rflag' freqdevscale=3.0 “] On Hanning Smoothed Data

flagdata(vis='xxx.ms', mode='list', inpfile=cmdlist, action='calculate', display='data')



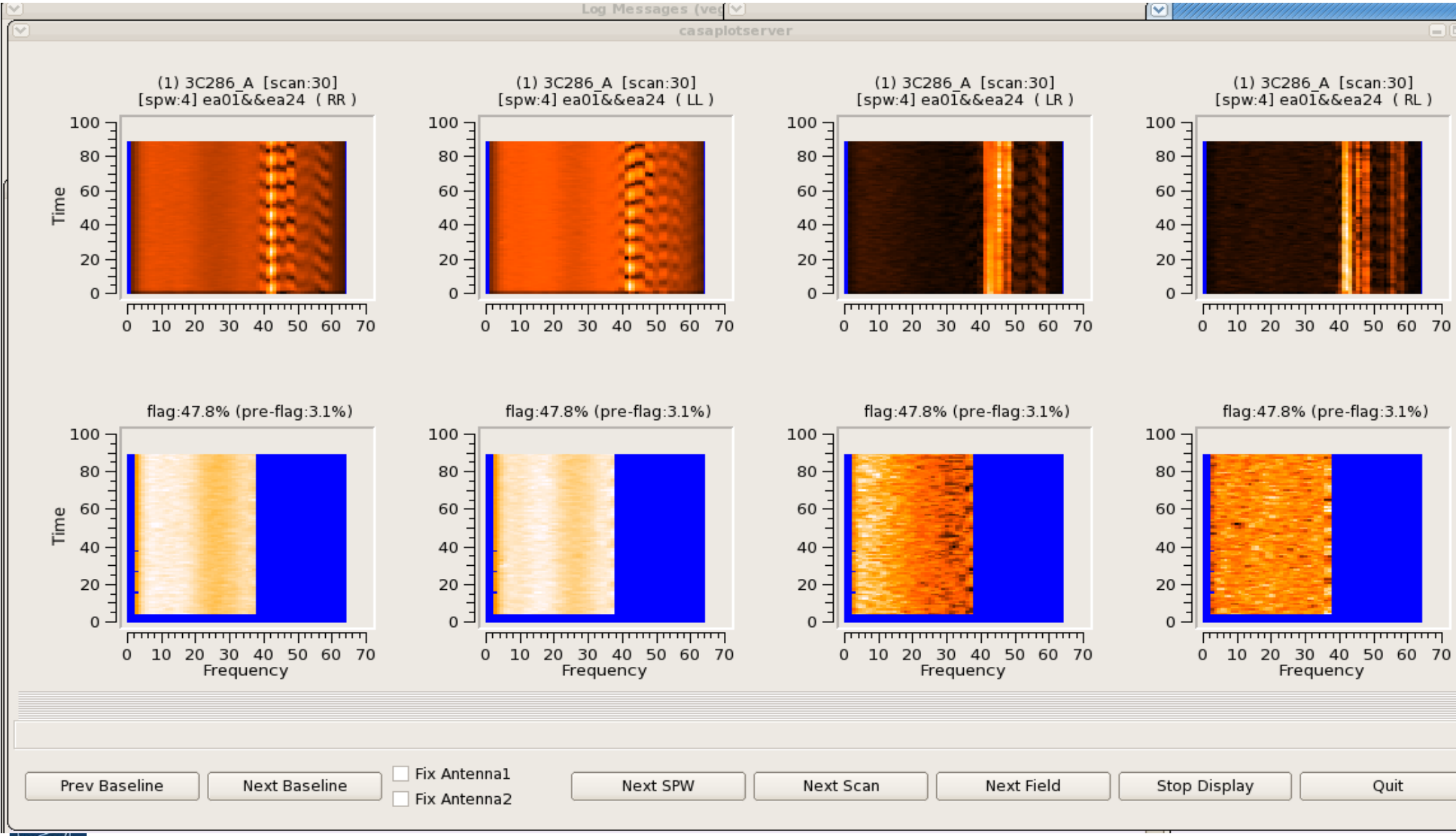
```
cmdlist = [ “ spw='4'  mode='rflag'  freqdevscale=3.0  “,
            “ spw='4'  mode='extend'  growtime=30.0 extendpols=True “ ]

flagdata(vis='xxx.ms', mode='list', inpfile=cmdlist, action='calculate', display='data')
```

On Hanning Smoothed Data

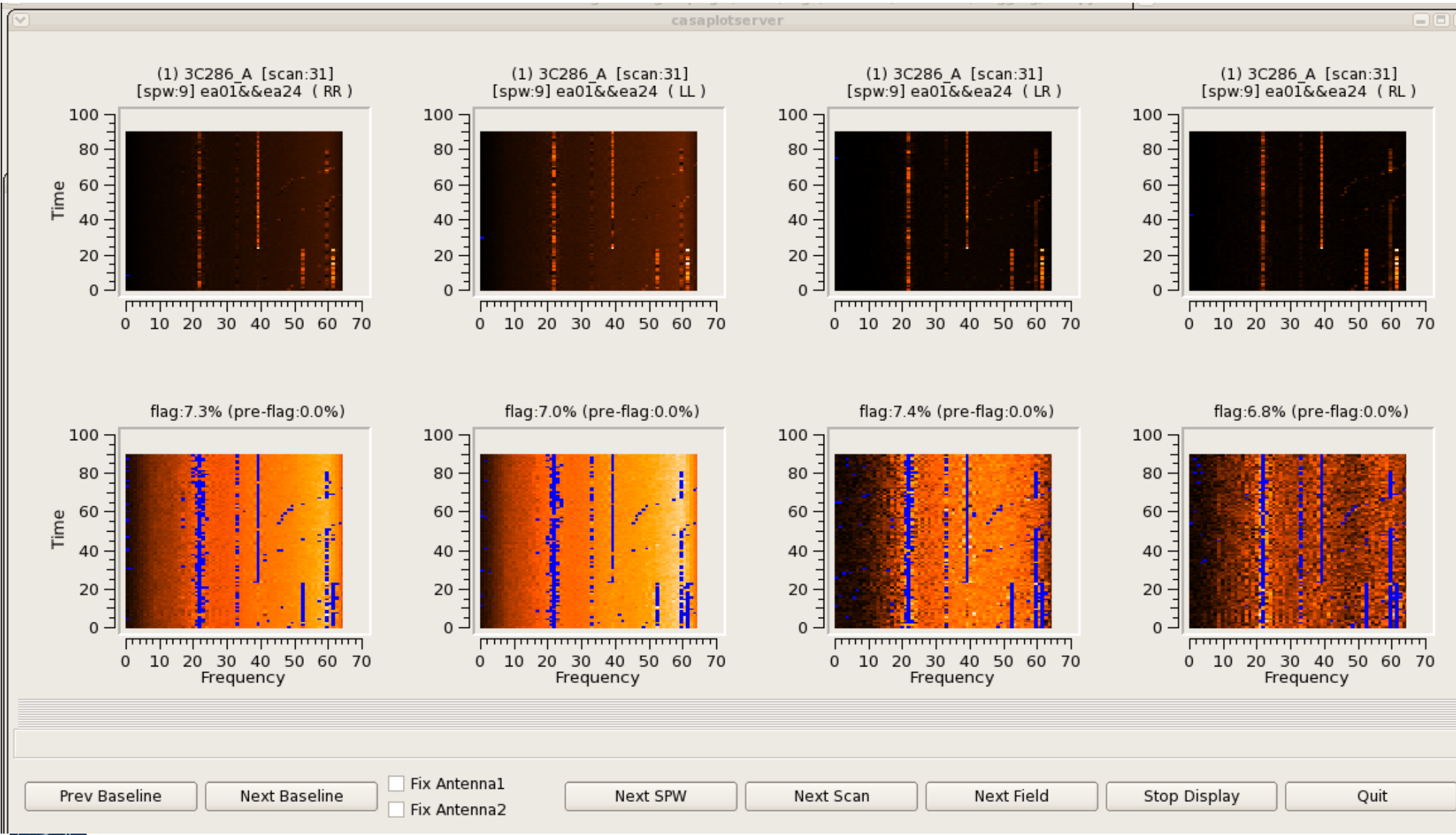


```
cmdlist = [ “ spw='4'  mode='rflag'  freqdevscale=3.0  “,
            “ spw='4'  mode='extend'  growtime=30.0 extendpols=True  flagnearfreq=True “ ]

flagdata(vis='xxx.ms', mode='list', inpfile=cmdlist, action='calculate', display='data')
```

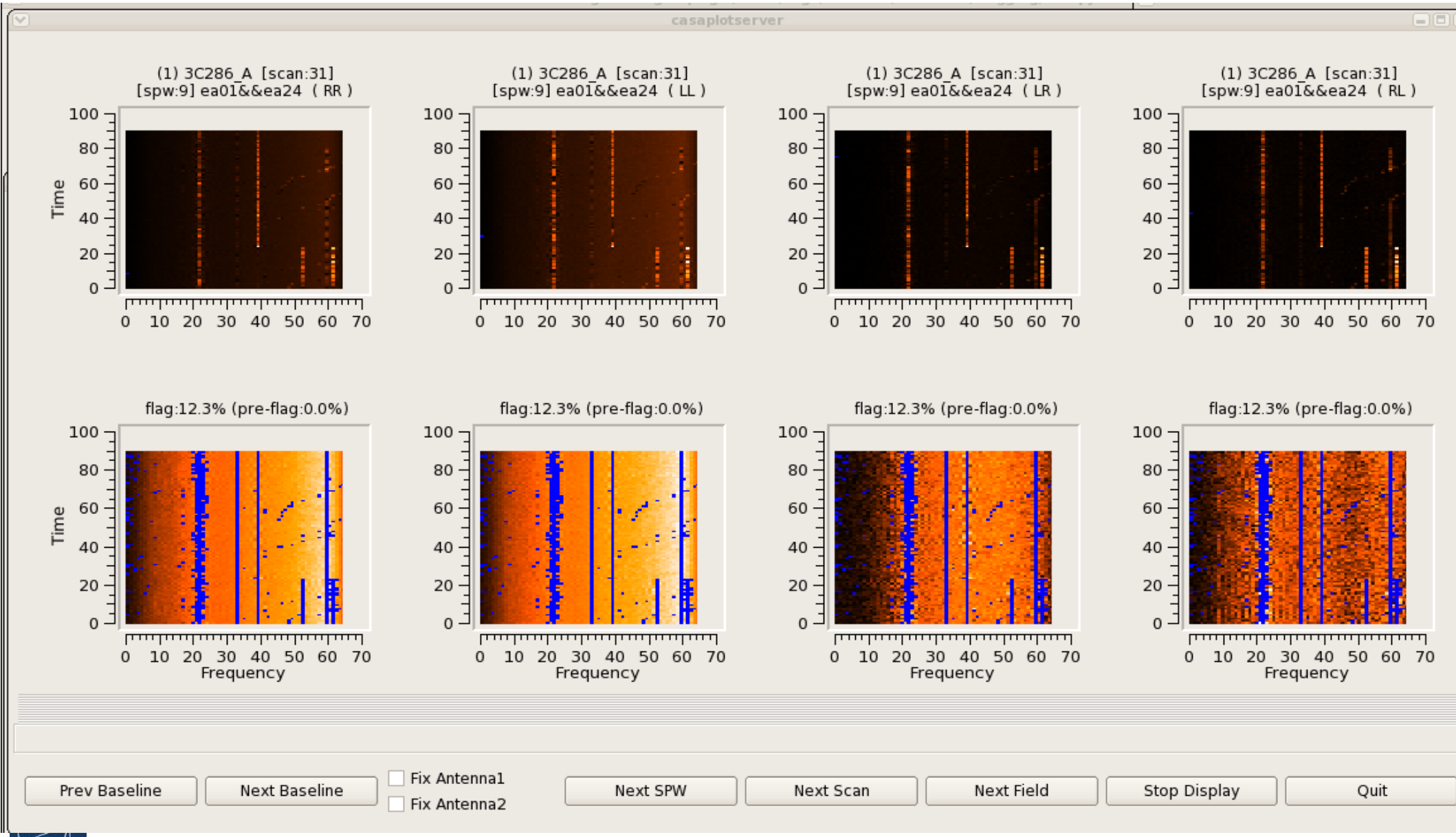
On Hanning Smoothed Data



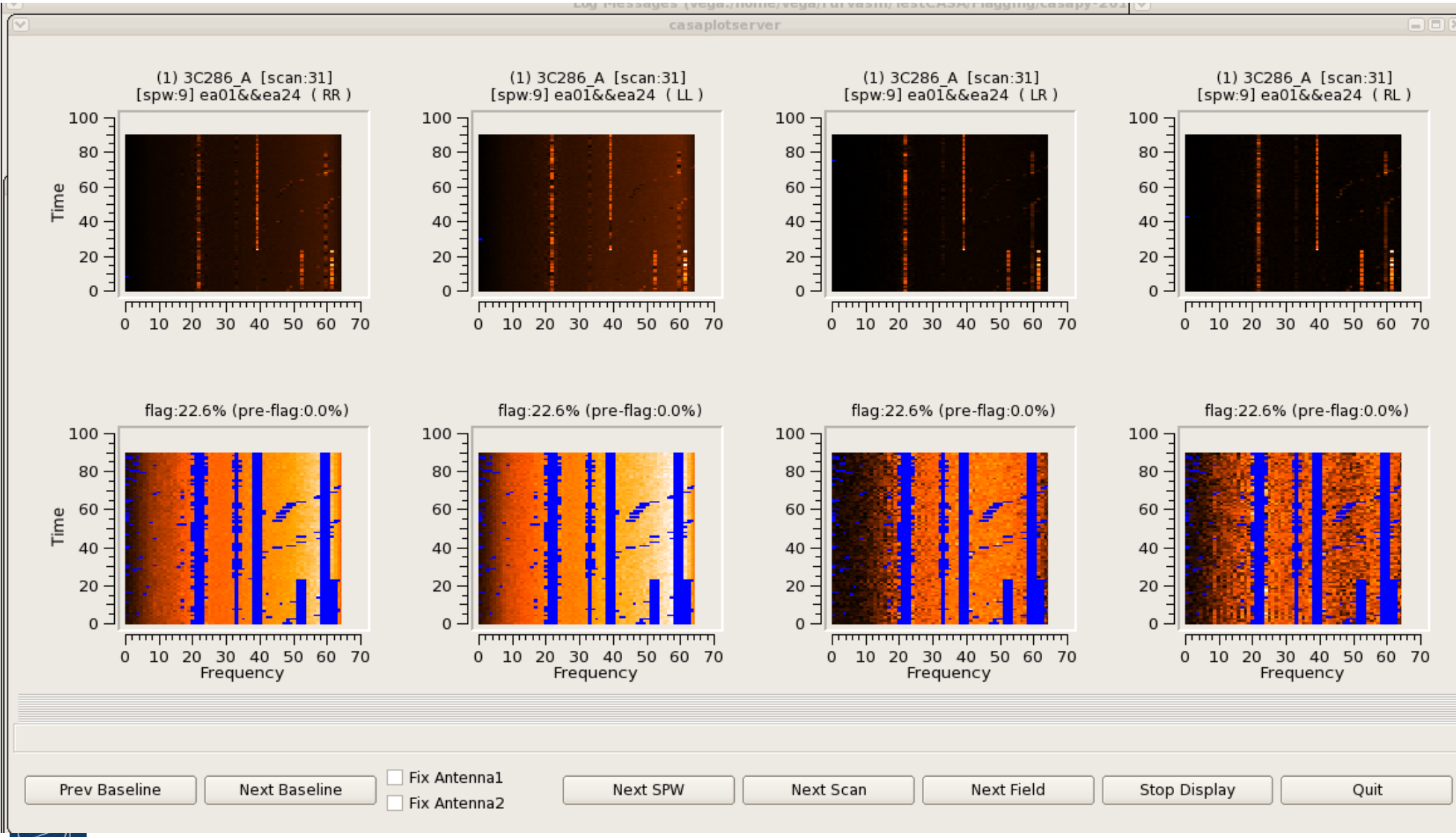
```
cmdlist = [ “ spw='9'  mode='tfcrop' “ ]

flagdata(vis='xxx.ms', mode='list', inpfile=cmdlist, action='calculate', display='data')
```



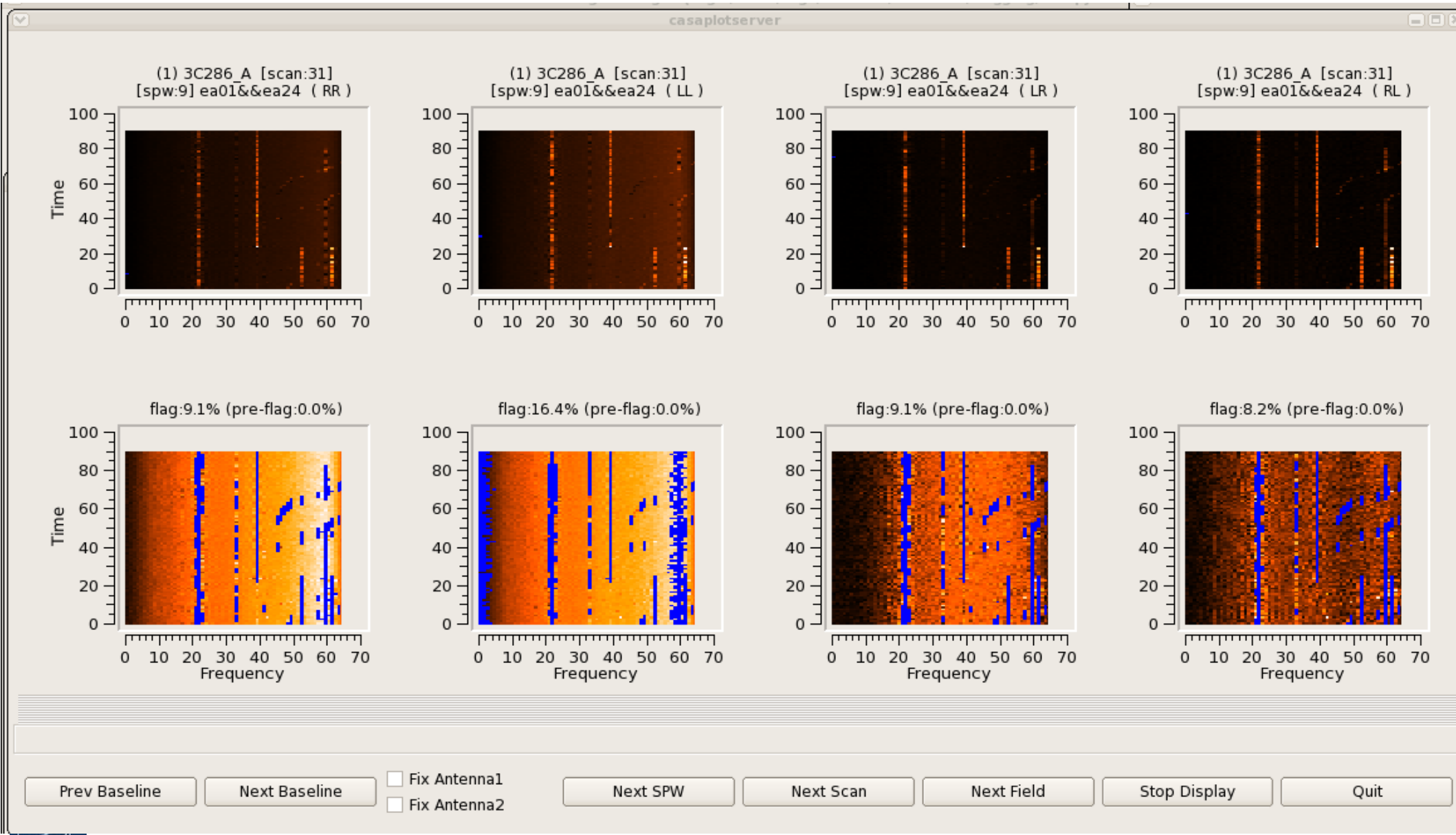
```
cmdlist = [ “ spw='9'  mode='tfcrop'  “,
            “ spw='9'  mode='extend'  growtime=50.0 extendpols=True ]

flagdata(vis='xxx.ms', mode='list', inpfile=cmdlist, action='calculate', display='data')
```



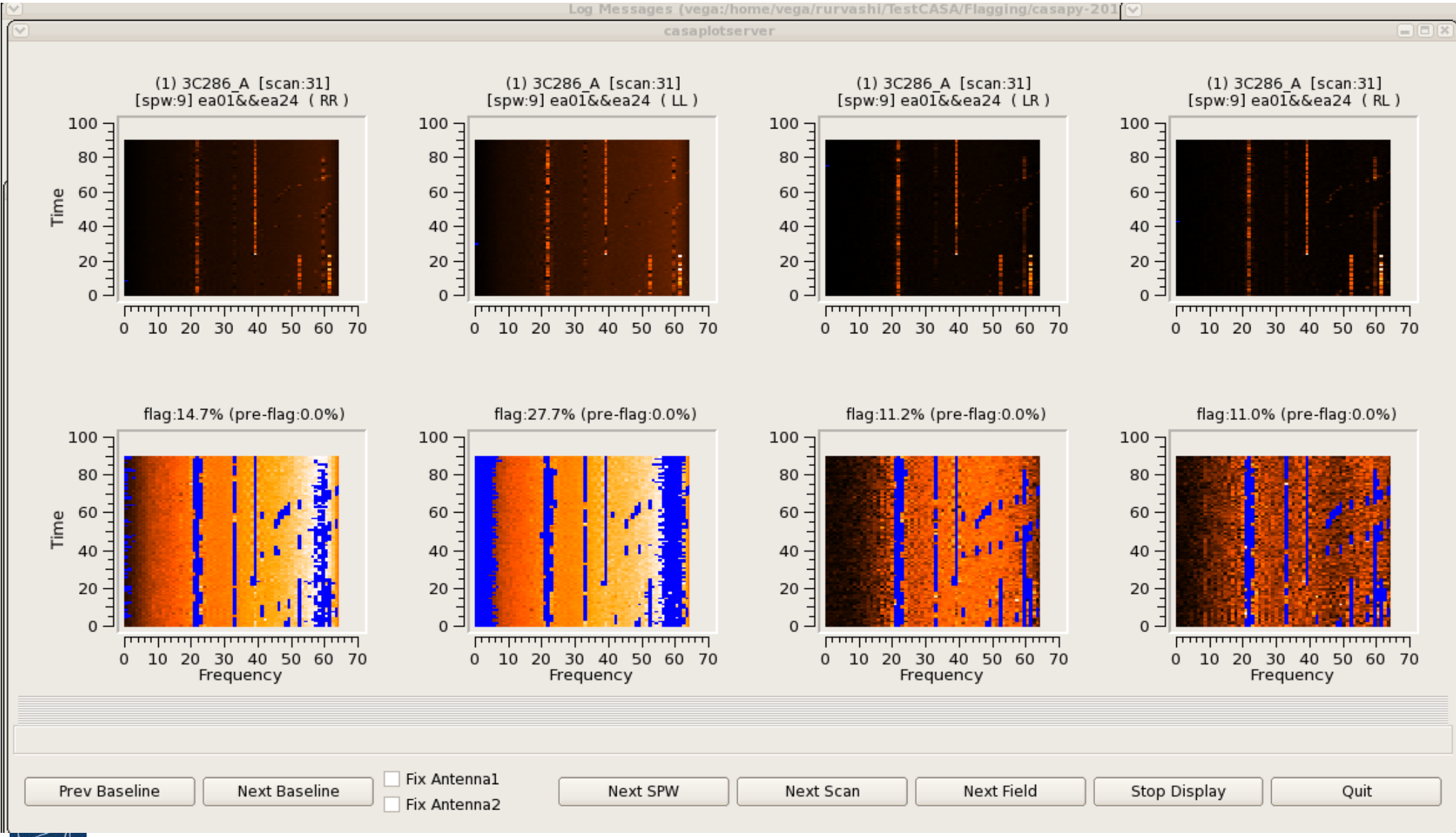
```
cmdlist = [ “ spw='9'  mode='tfcrop'  usewindowstats='sum'  “ ,
            “ spw='9'  mode='extend'  growtime=50.0  extendpols=True “ ]

flagdata(vis='xxx.ms', mode='list', inpfile=cmdlist, action='calculate', display='data')
```



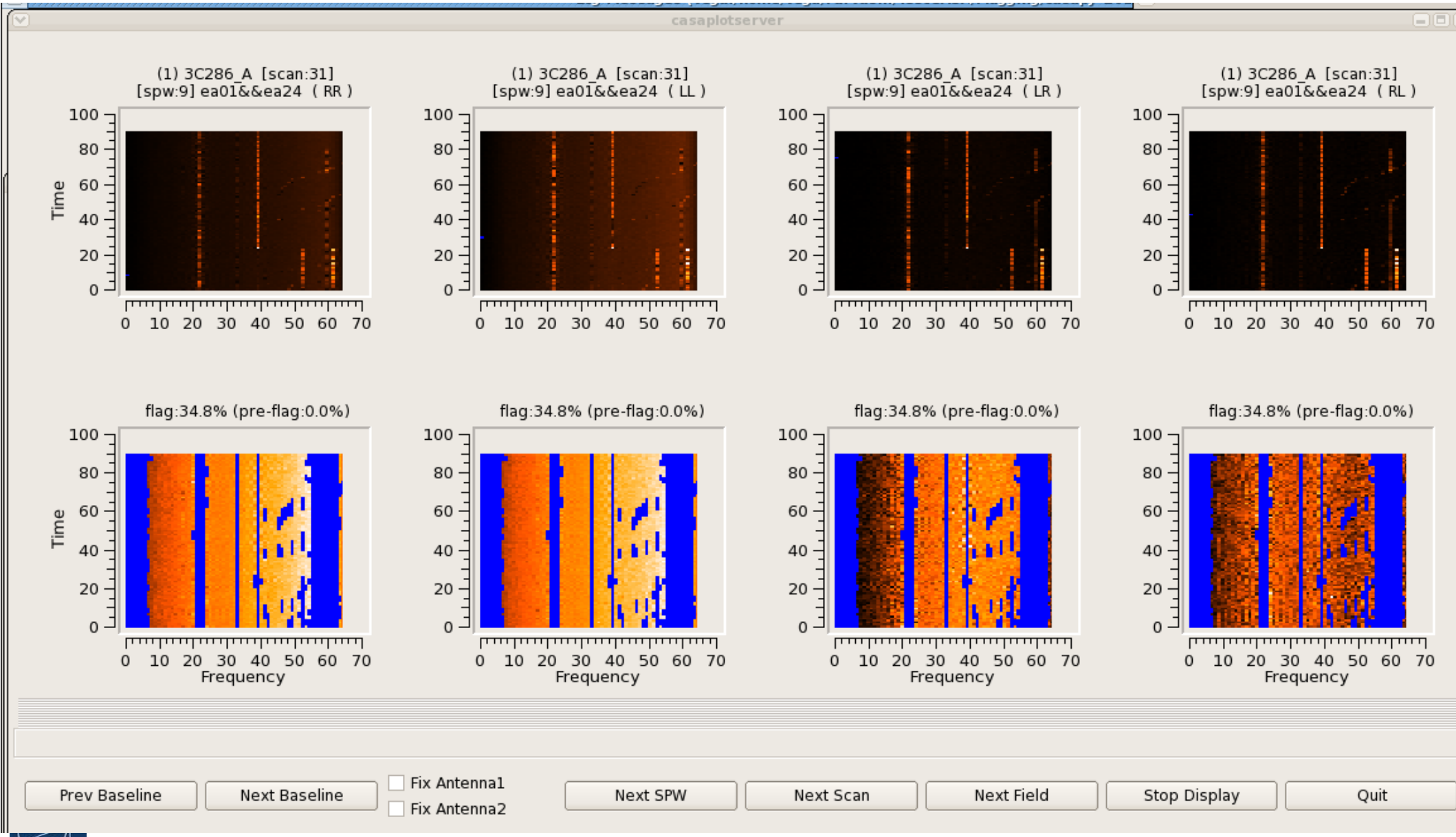
```
cmdlist = [ “ spw='9'  mode='rflag' “ ]

flagdata(vis='xxx.ms', mode='list', inpfile=cmdlist, action='calculate', display='data')
```



```
cmdlist = [ “ spw='9'  mode='rflag'  freqdevscale=4.0 “ ]

flagdata(vis='xxx.ms', mode='list', inpfile=cmdlist, action='calculate', display='data')
```

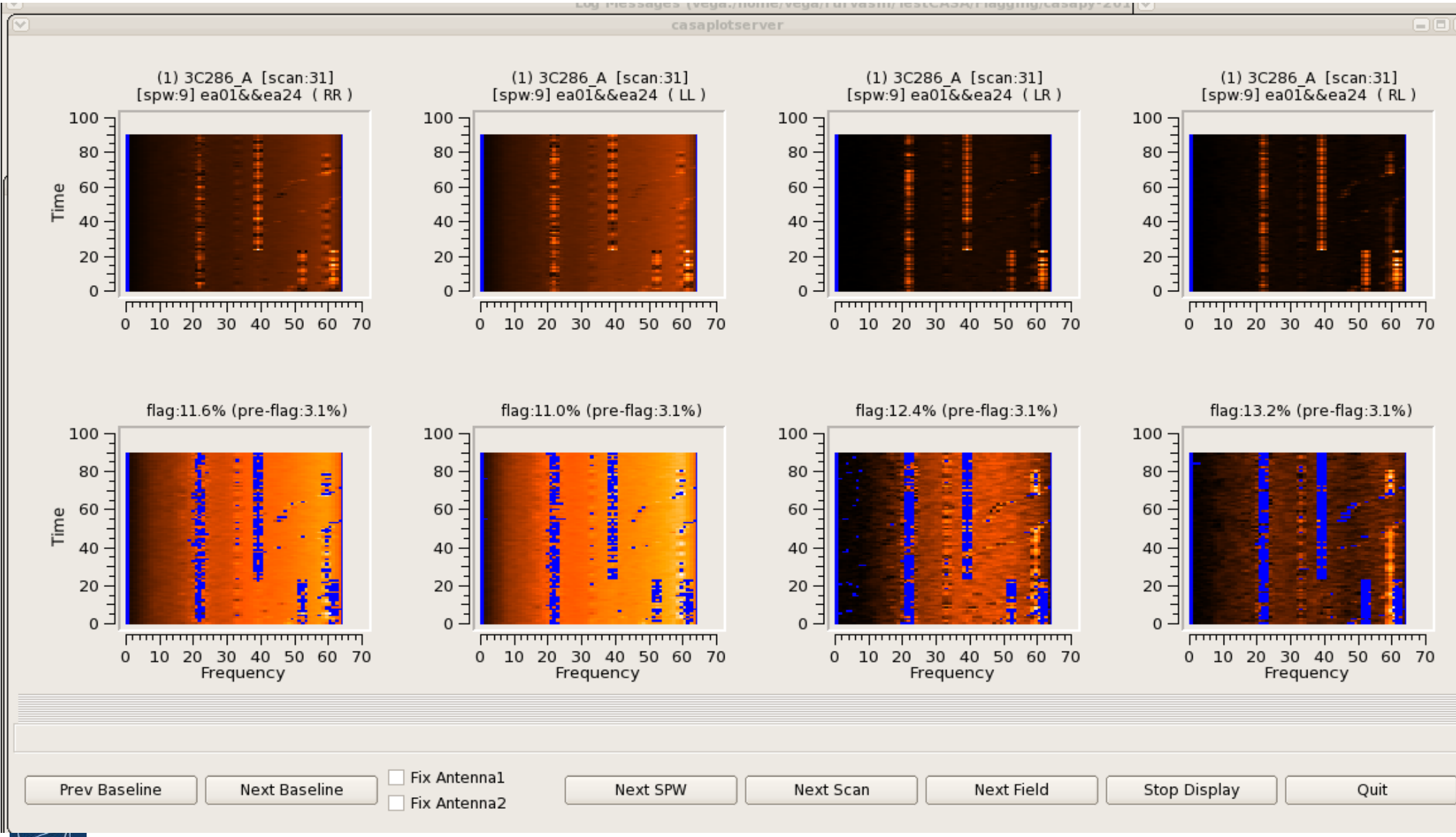


```
cmdlist = [ “ spw='9'  mode='rflag'  freqdevscale=4.0 “ ,
            “ spw='9'  mode='extend'  growtime=50.0   flagneartime=True  extendpols=True ]

flagdata(vis='xxx.ms', mode='list', inpfile=cmdlist, action='calculate', display='data')
```



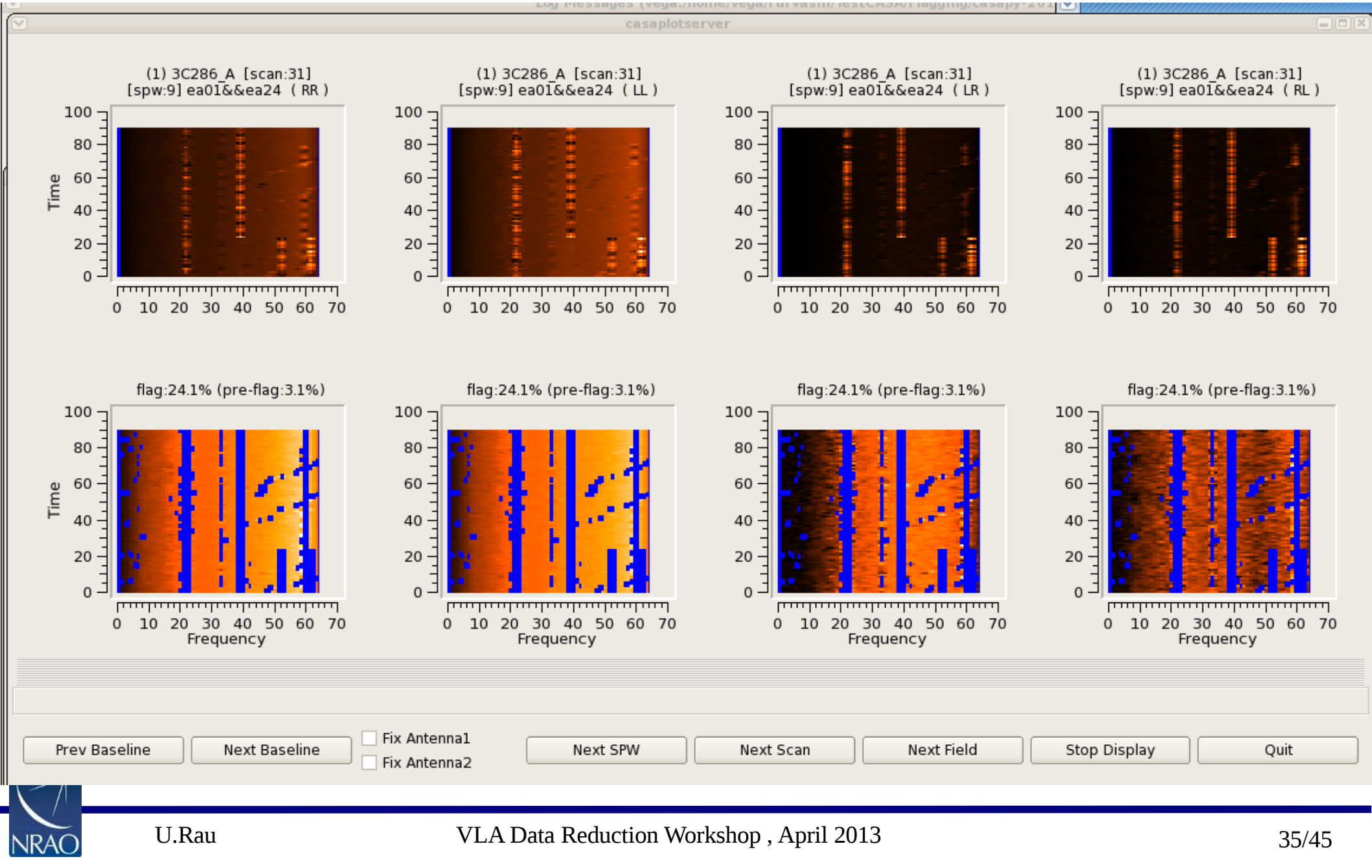
cmdlist = [“ spw='9' mode='tfcrop' “] On Hanning Smoothed Data

flagdata(vis='xxx.ms', mode='list', inpfile=cmdlist, action='calculate', display='data')



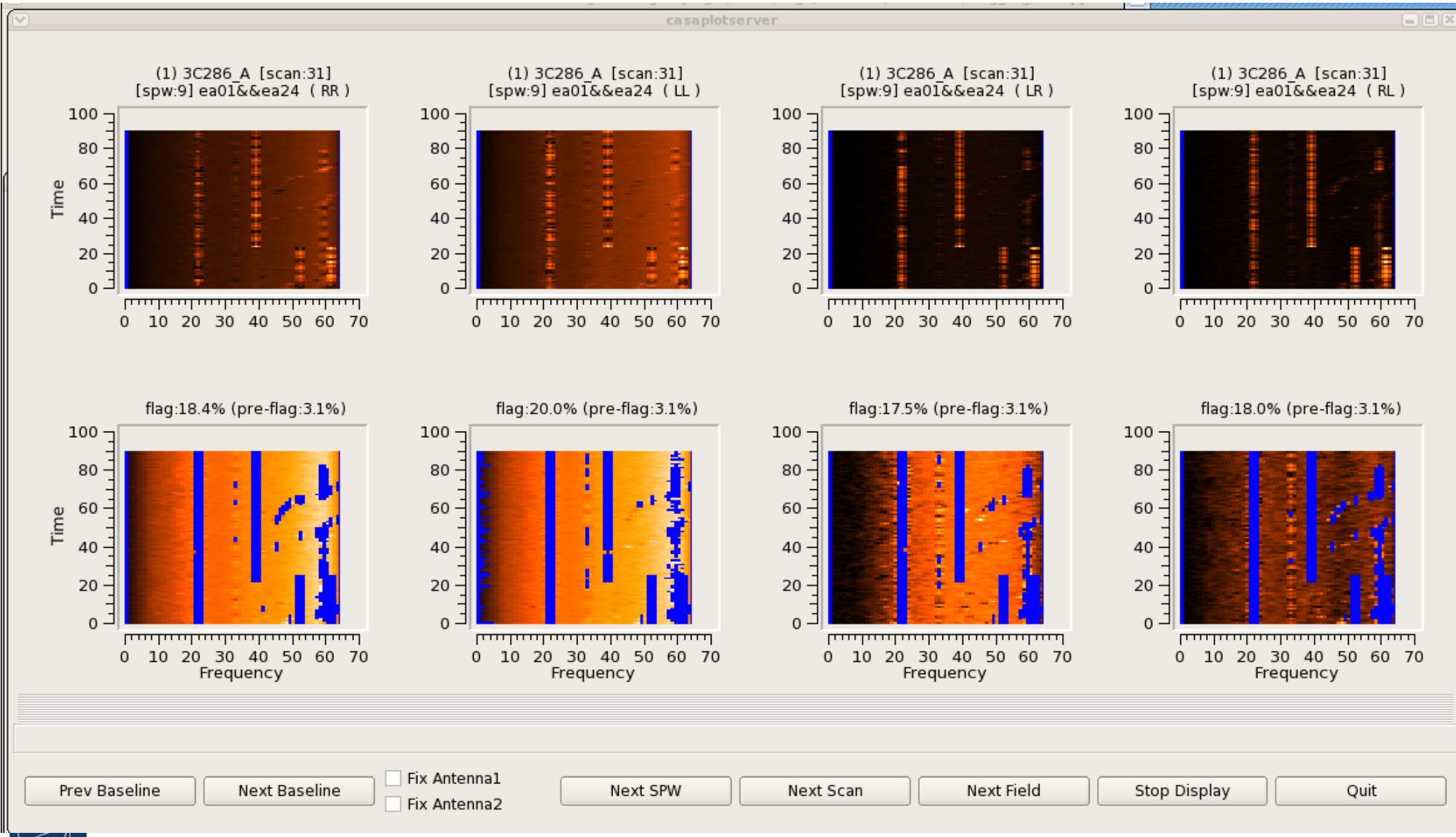
```
cmdlist = [ “ spw='9'  mode='tfcrop' “ ,
            “ spw='9'  mode='extend'  growtime=50.0 extendpols=True  flagneartime=True“ ]

flagdata(vis='xxx.ms', mode='list', inpfile=cmdlist, action='calculate', display='data')
```

On Hanning Smoothed Data



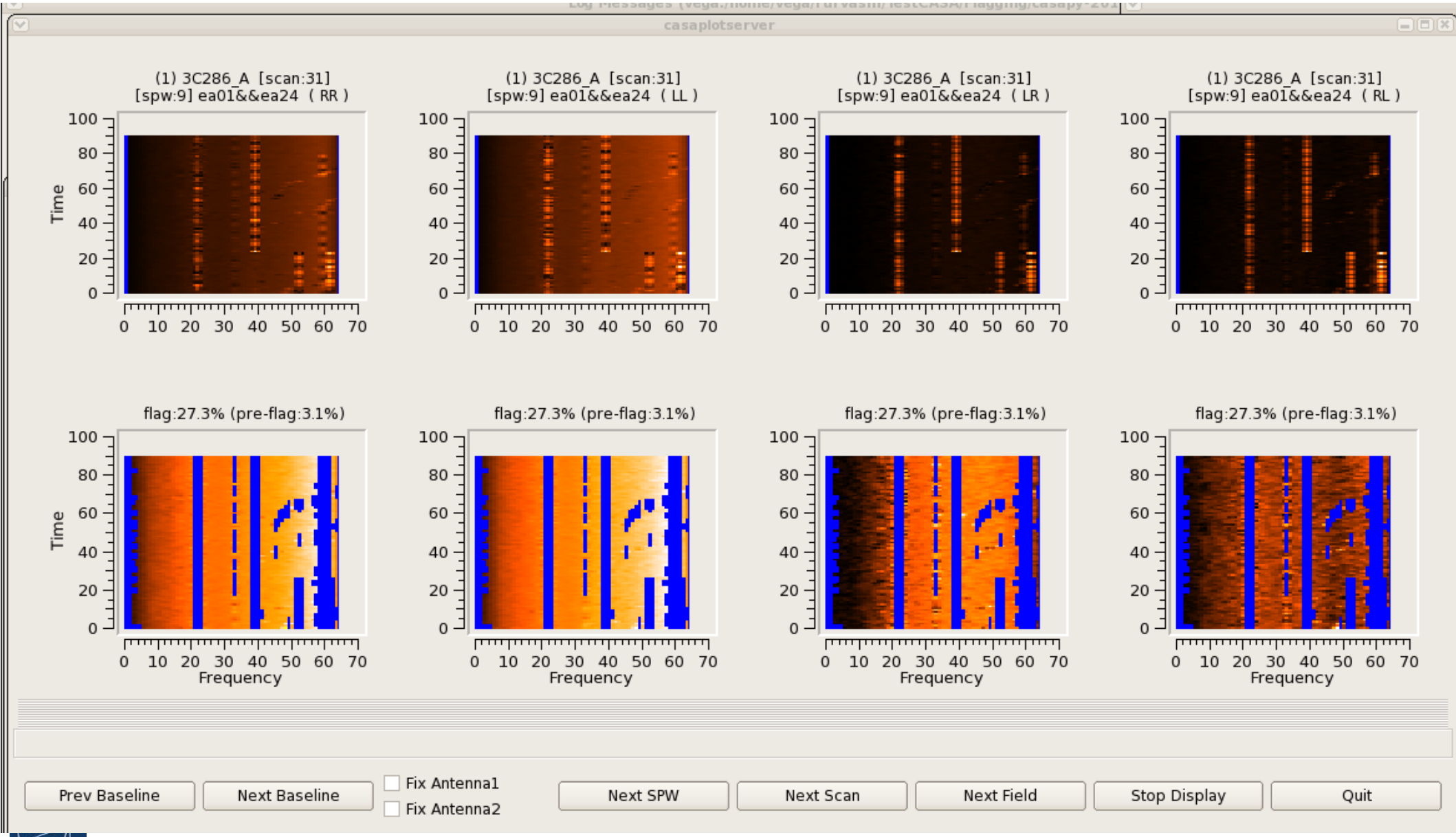
cmdlist = [“ spw='9' mode='rflag' “] On Hanning Smoothed Data

flagdata(vis='xxx.ms', mode='list', inpfile=cmdlist, action='calculate', display='data')



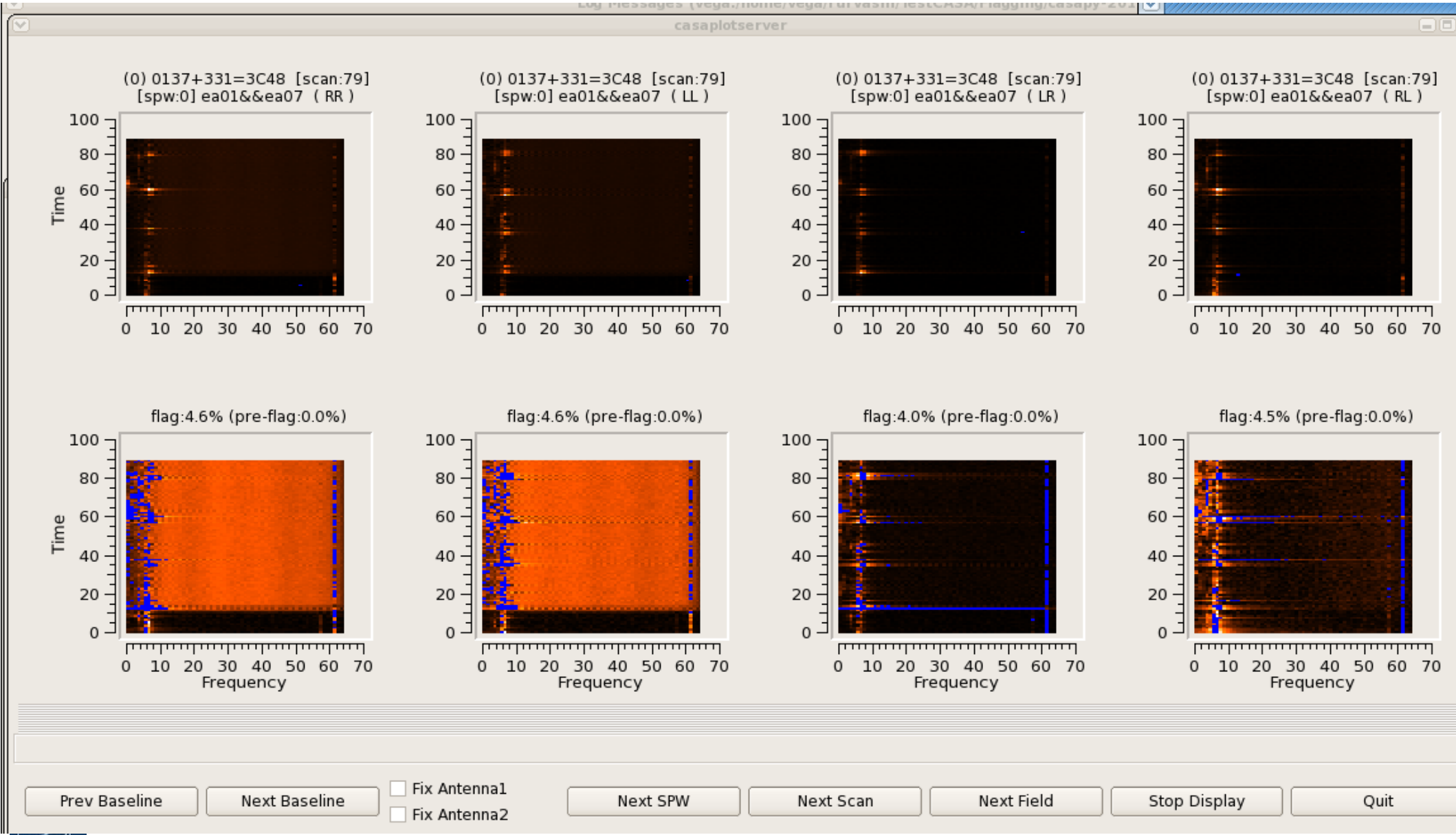
```
cmdlist = [ “ spw='9'  mode='rflag' “ ,
            “ spw='9'  mode='extend'  growtime=50.0 extendpols=True  flagneartime=True“ ]

flagdata(vis='xxx.ms', mode='list', inpfile=cmdlist, action='calculate', display='data')
```

On Hanning Smoothed Data

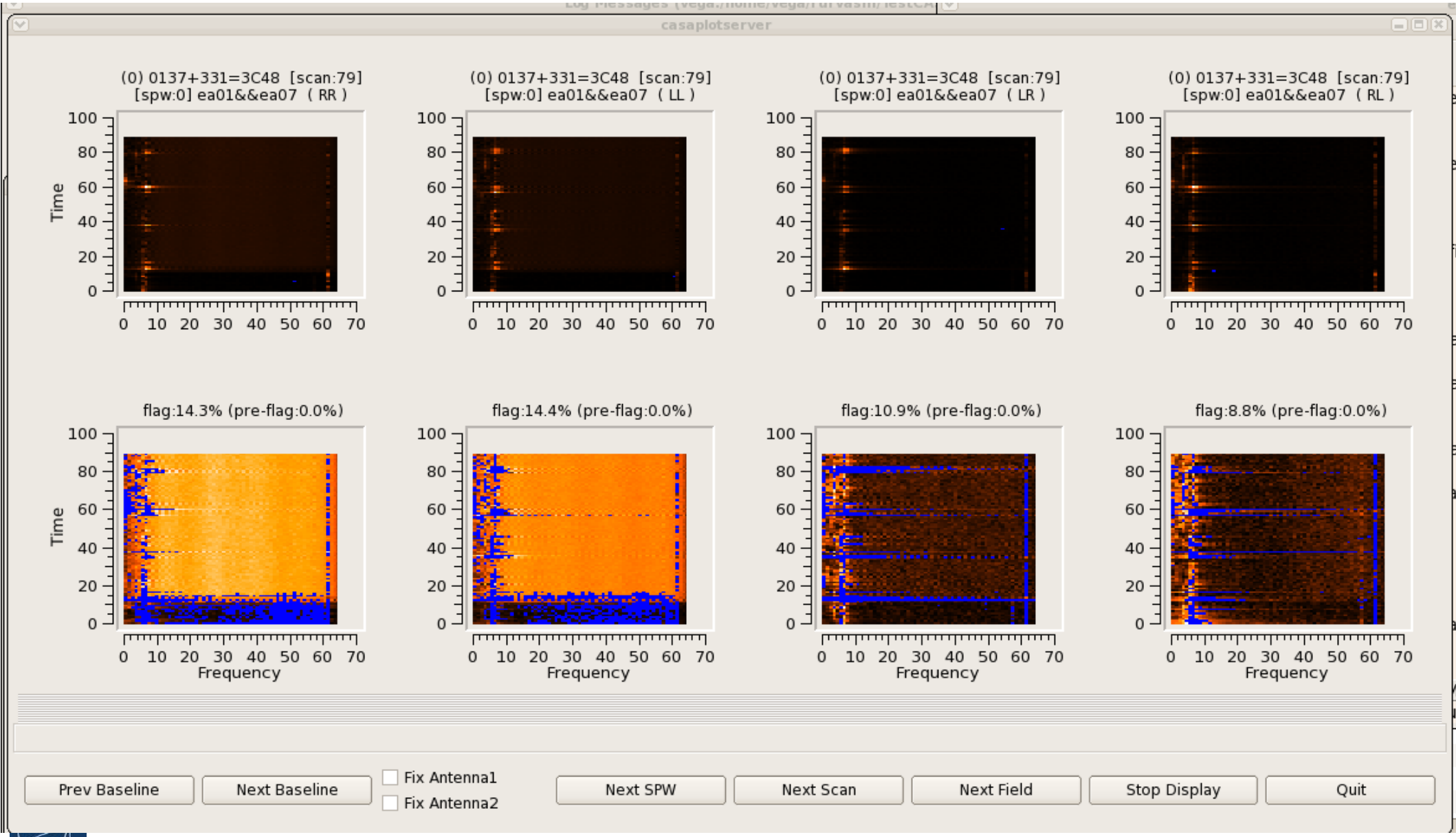


cmdlist = [“ spw='9' mode='tfcrop' “]

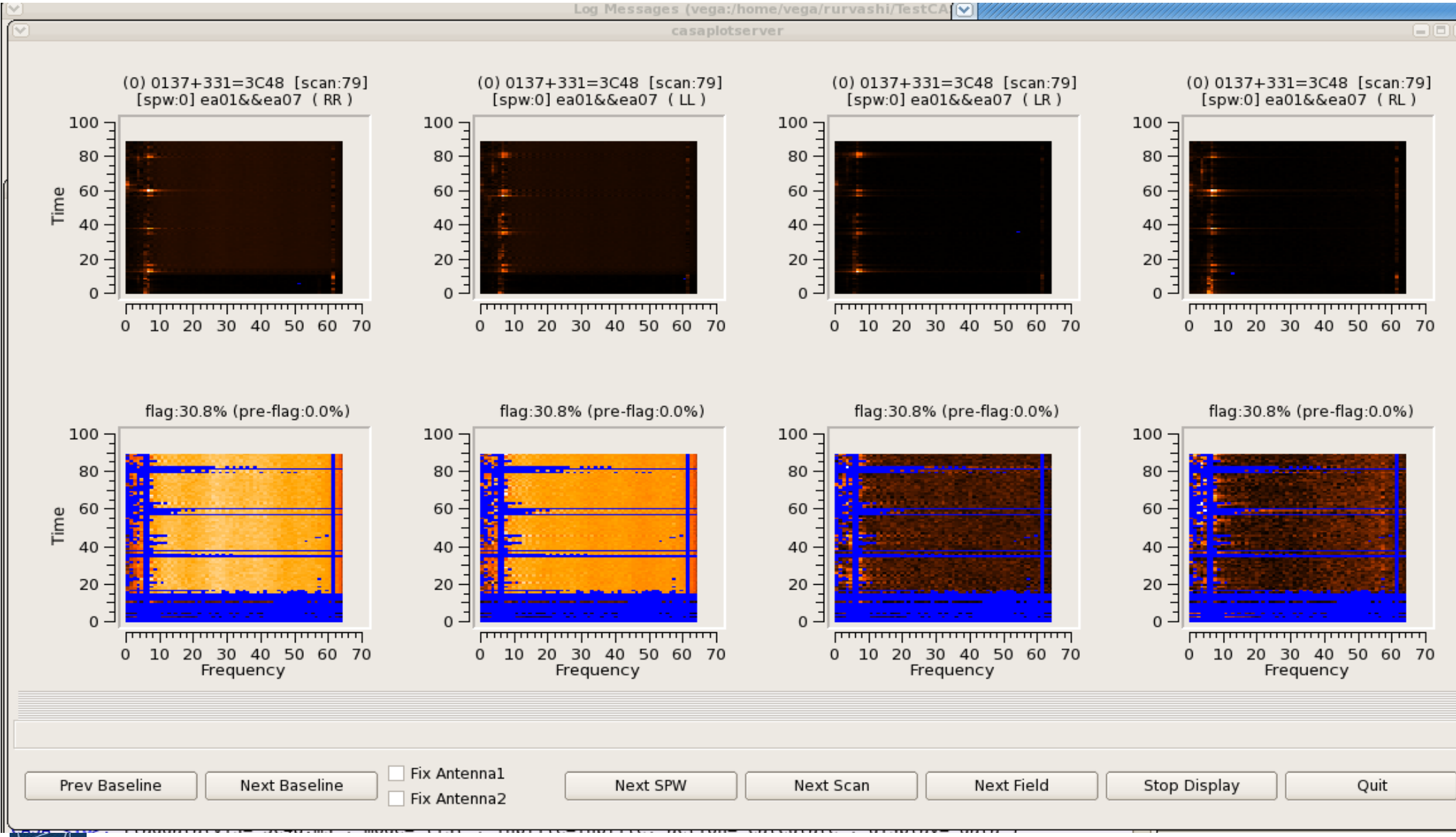
flagdata(vis='xxx.ms', mode='list', inpfile=cmdlist, action='calculate', display='data')



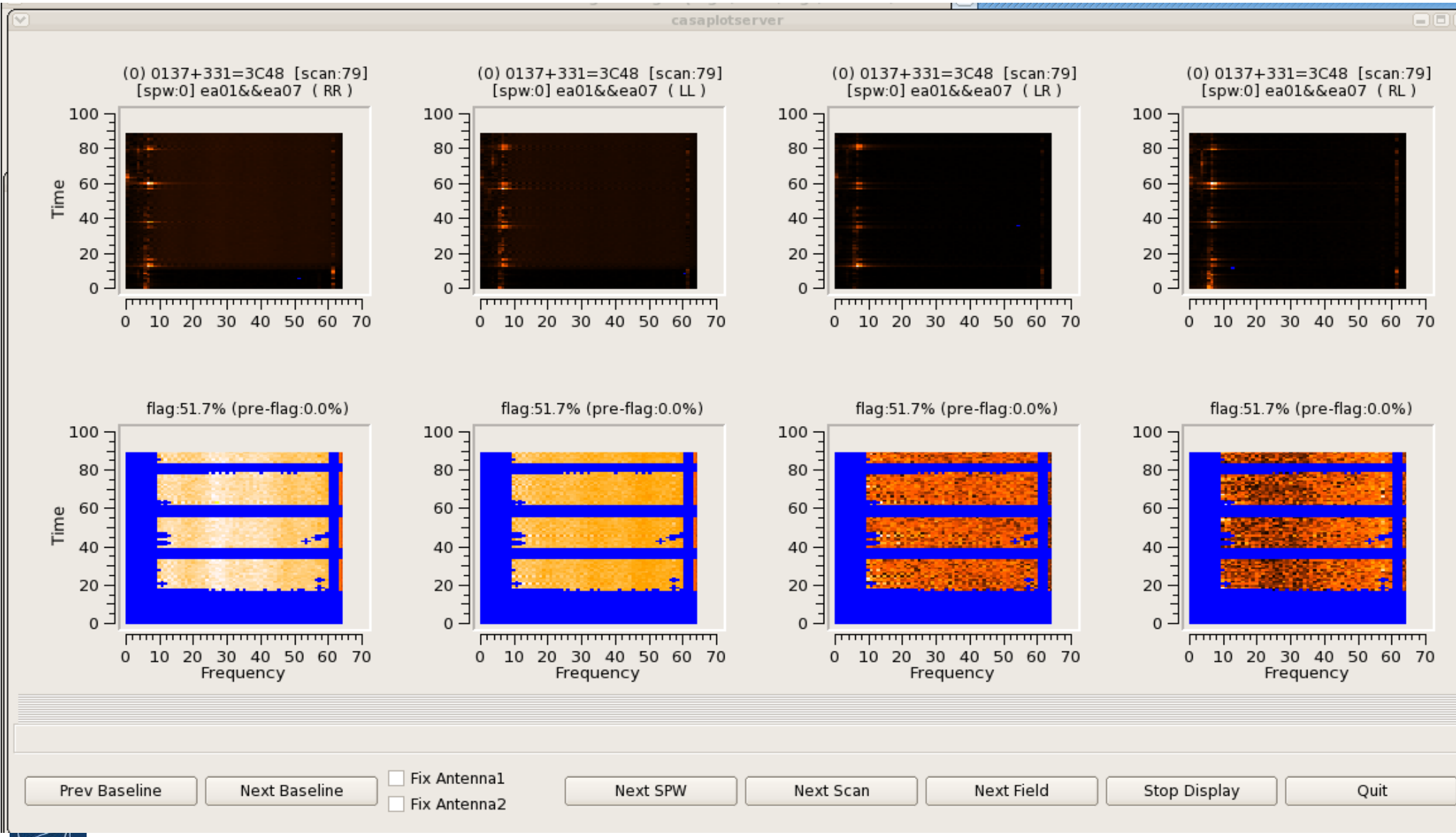
```
cmdlist = [ “ spw='9'  mode='tfcrop' maxnpieces=4 timecutoff=2.5 freqcutoff=3.0 timefit='poly' “ ]

flagdata(vis='xxx.ms', mode='list', inpfile=cmdlist, action='calculate', display='data')
```



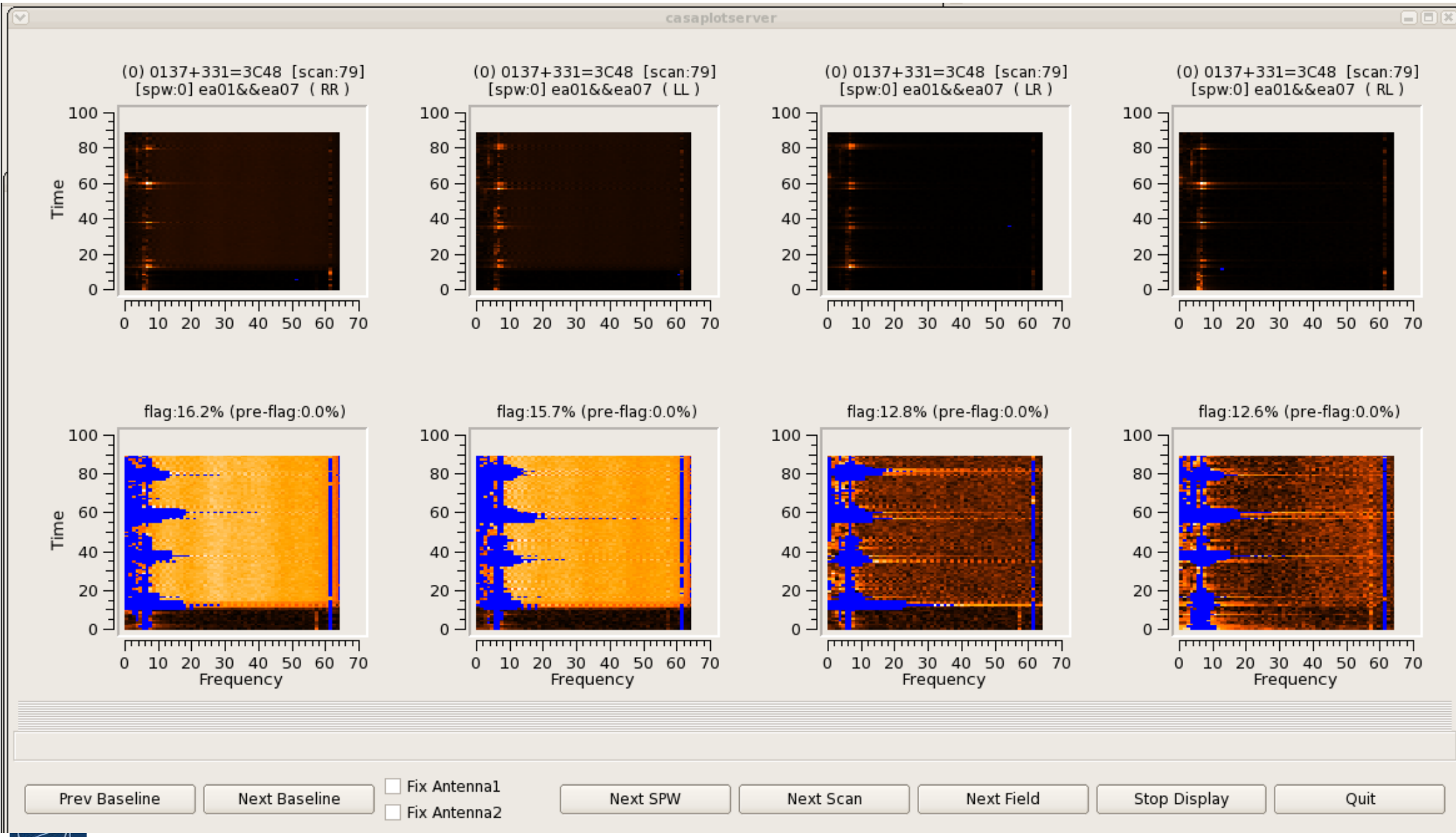
```
cmdlist = [ “ spw='9'  mode='tfcrop' maxnpieces=4 timecutoff=2.5 freqcutoff=3.0 timefit='poly' “,
            “ spw='9'  mode='extend' growtime=50.0  extendpols=True  growfreq=50.0 “ ]

flagdata(vis='xxx.ms', mode='list', inpfile=cmdlist, action='calculate', display='data')
```



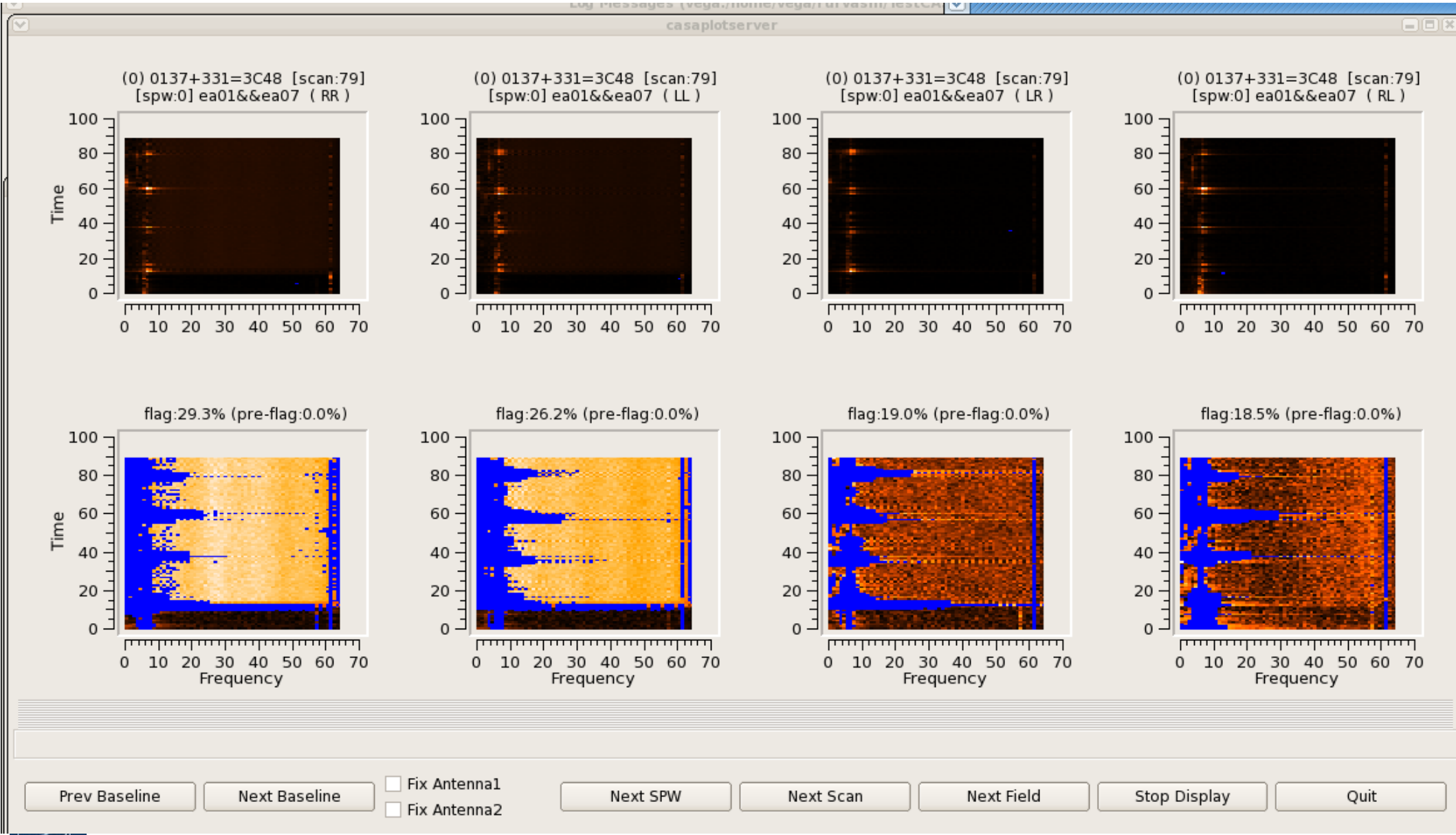
```
cmdlist = [ “ spw='9'  mode='tfcrop' maxnpieces=4 timecutoff=2.5 freqcutoff=3.0 timefit='poly' “,
            “ spw='9'  mode='extend' growtime=50.0  extendpols=True  growfreq=50.0
                                                    flagnearfreq=True  flagnearfreq=True “ ]
flagdata(vis='xxx.ms', mode='list', inpfile=cmdlist, action='calculate', display='data')
```



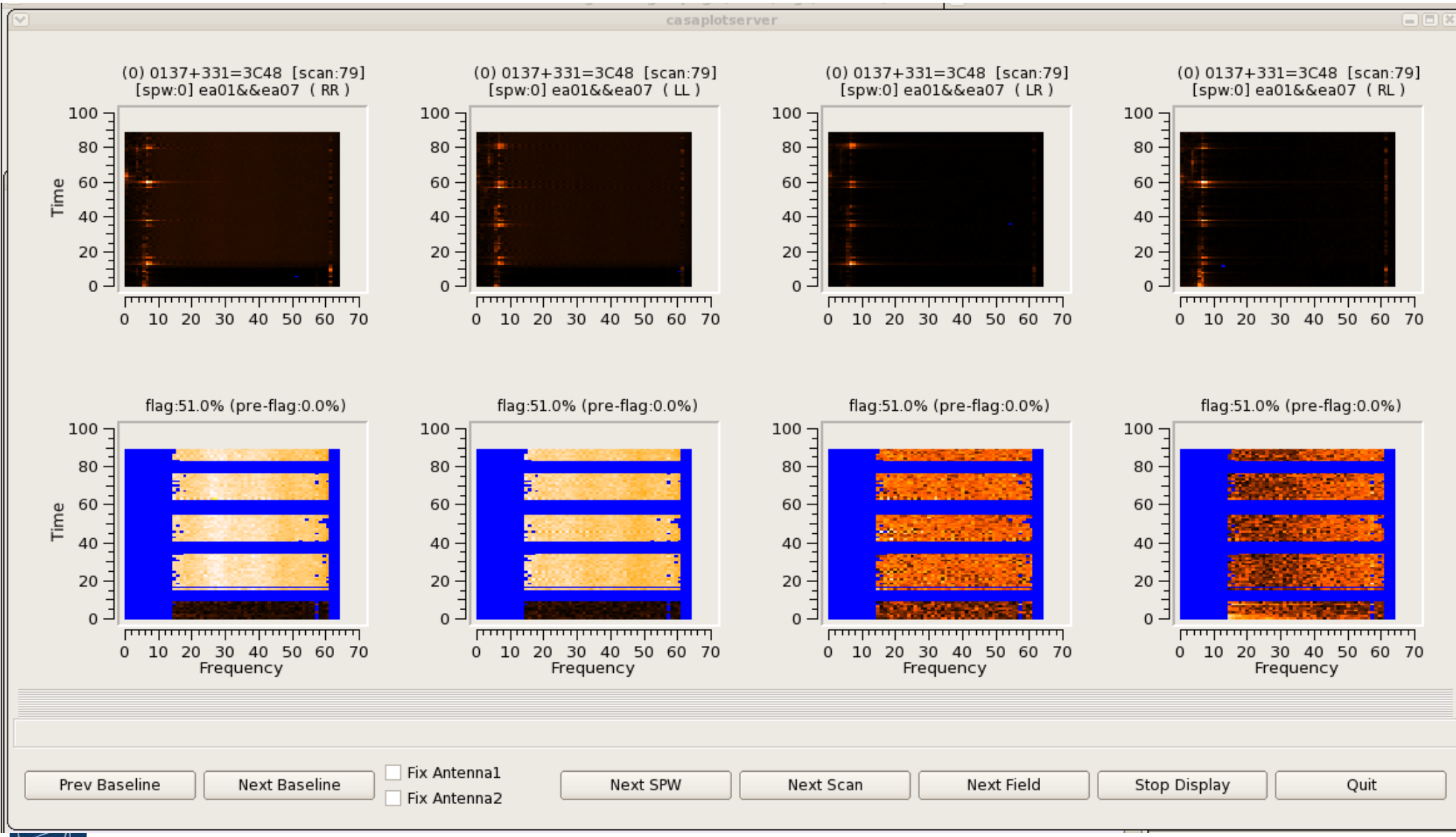
```
cmdlist = [ “ spw='9'  mode='rflag' “ ]

flagdata(vis='xxx.ms', mode='list', inpfile=cmdlist, action='calculate', display='data')
```



```
cmdlist = [ " spw='9'  mode='rflag'  freqdevscale=3.0  timedevscale=3.0 " ]

flagdata(vis='xxx.ms', mode='list', inpfile=cmdlist, action='calculate', display='data')
```



```
cmdlist = [ “ spw='9'  mode='rflag'  freqdevscale=3.0  timedevscale=3.0 “ ,
            “ spw='9'  mode='extend'  growtime=50.0  growfreq=30.0  extendpols=True “ ]

flagdata(vis='xxx.ms', mode='list', inpfile=cmdlist, action='calculate', display='data')
```



Summary

– Automatic Flagging options exist.

– They all need tuning. Usually, one setup per SPW or band

=> Look at small pieces of your data, and decide flagging strategy

=> Use plotms or viewer or flagdata(action='calculate', display='data') and try different flagging setups. AOflagger has its own displays.

=> Defaults will not suffice for all cases.

=> Modify thresholds, sliding-window sizes, and extend flags.

– Use batch-modes in flagdata/flagcmd when relevant, to run multiple flag commands in sequence, using only one pass through the data.

– Documentation : CASA – Using Casa – Other Documentation – Flagging