Automatic detection of spectral lines and sources in 3D cubes

Maxim Voronkov

(Maxim.Voronkov@csiro.au)

CSIRO - Australia Telescope National Facility

Mopra workshop - Swinburne, 13.06.2008

ASAP line search in 1D spectra

- Large volume of data encourages automatic data reduction
- ATNF Spectral Analysis Package (ASAP) has a routine for automatic search of spectral lines in its standard distribution.
  - Class asaplinefind contains the routine

    Just type help asaplinefind in ASAP to get information about the interface and an example.
  - The algorithm involves a simple threshold criterion:

    A detection is claimed if a specified number of spectral channels deviates by more than a given threshold from the local baseline estimate.
  - Main challenges for the algorithm are:

    Bandpass shape vs. broad lines

    Off-line noise estimates if we don't know where the lines are

    Strong lines in the spectrum affect statistics and cause spurious detections

    Line wings can be below threshold

    Broad lines can be significantly oversampled

ASAP line search in 1D spectra

- Large volume of data encourages automatic data reduction
- ATNF Spectral Analysis Package (ASAP) has a routine for automatic search of spectral lines in its standard distribution.
  - Class asaplinefind contains the routine

    Just type help asaplinefind in ASAP to get information about the interface and an example.
  - The algorithm involves a simple threshold criterion:

    A detection is claimed if a specified number of spectral channels deviates by more than a given threshold from the local baseline estimate.
  - Main challenges for the algorithm are:

    Bandpass shape vs. broad lines → configurable parameter

    Off-line noise estimates if we don't know where the lines are

    Strong lines in the spectrum affect statistics and cause spurious detections

    Line wings can be below threshold

    Broad lines can be significantly oversampled

ASAP line search in 1D spectra

- Large volume of data encourages automatic data reduction
- ATNF Spectral Analysis Package (ASAP) has a routine for automatic search of spectral lines in its standard distribution.
  - Class asaplinefind contains the routine

    Just type help asaplinefind in ASAP to get information about the interface and an example.
  - The algorithm involves a simple threshold criterion:

    A detection is claimed if a specified number of spectral channels deviates by more than a given threshold from the local baseline estimate.
  - Main challenges for the algorithm are:

    Bandpass shape vs. broad lines

    Off-line noise estimates if we don't know where the lines are

    Strong lines in the spectrum affect statistics and cause spurious detections

    Line wings can be below threshold

    Broad lines can be significantly oversampled

multiple passes

ASAP line search in 1D spectra

- Large volume of data encourages automatic data reduction
- ATNF Spectral Analysis Package (ASAP) has a routine for automatic search of spectral lines in its standard distribution.
  - Class asaplinefind contains the routine

    Just type help asaplinefind in ASAP to get information about the interface and an example.
  - The algorithm involves a simple threshold criterion:

    A detection is claimed if a specified number of spectral channels deviates by more than a given threshold from the local baseline estimate.
  - Main challenges for the algorithm are:

    Bandpass shape vs. broad lines

    Off-line noise estimates if we don't know where the lines are

    Strong lines in the spectrum affect statistics and cause spurious detections

    Line wings can be below threshold $\longrightarrow$ wing search procedure

    Broad lines can be significantly oversampled

ASAP line search in 1D spectra

- Large volume of data encourages automatic data reduction
- ATNF Spectral Analysis Package (ASAP) has a routine for automatic search of spectral lines in its standard distribution.
  - Class asaplinefind contains the routine

    Just type help asaplinefind in ASAP to get information about the interface and an example.
  - The algorithm involves a simple threshold criterion:

    A detection is claimed if a specified number of spectral channels deviates by more than a given threshold from the local baseline estimate.
  - Main challenges for the algorithm are:

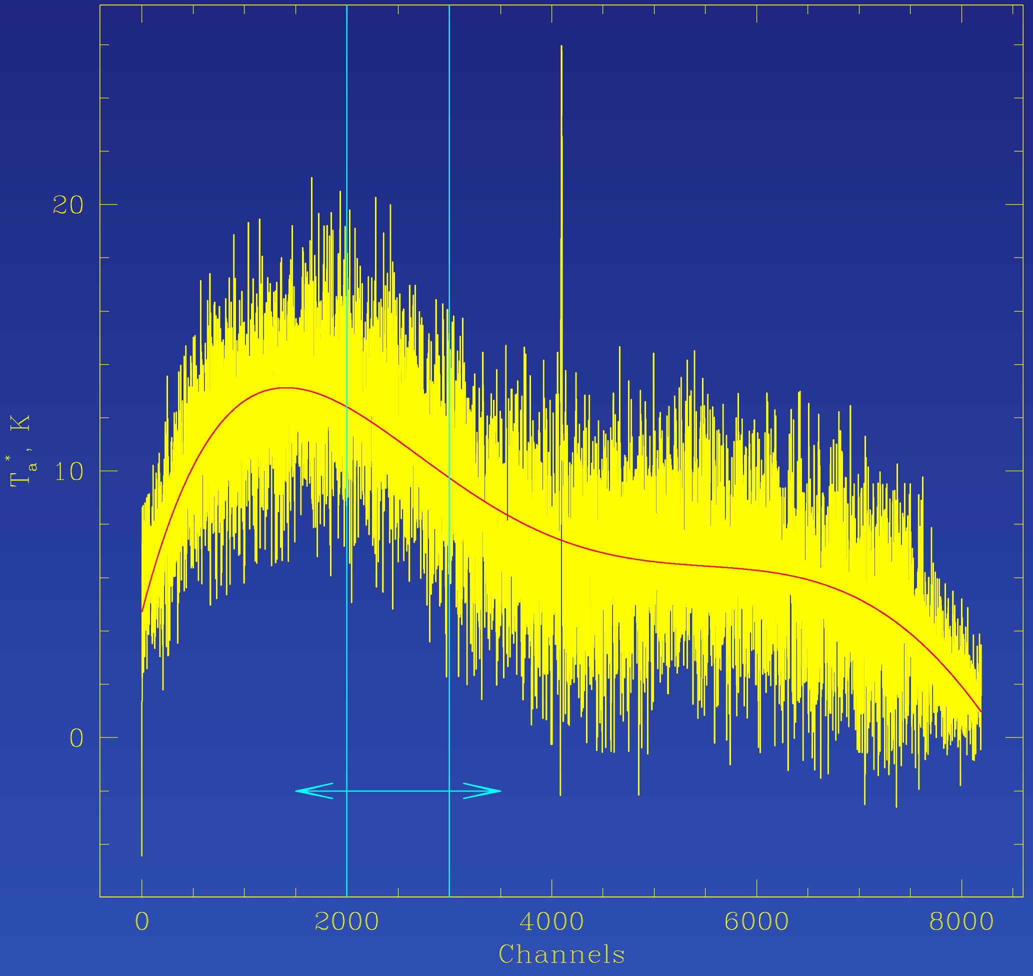
    Bandpass shape vs. broad lines

    Off-line noise estimates if we don't know where the lines are

    Strong lines in the spectrum affect statistics and cause spurious detections

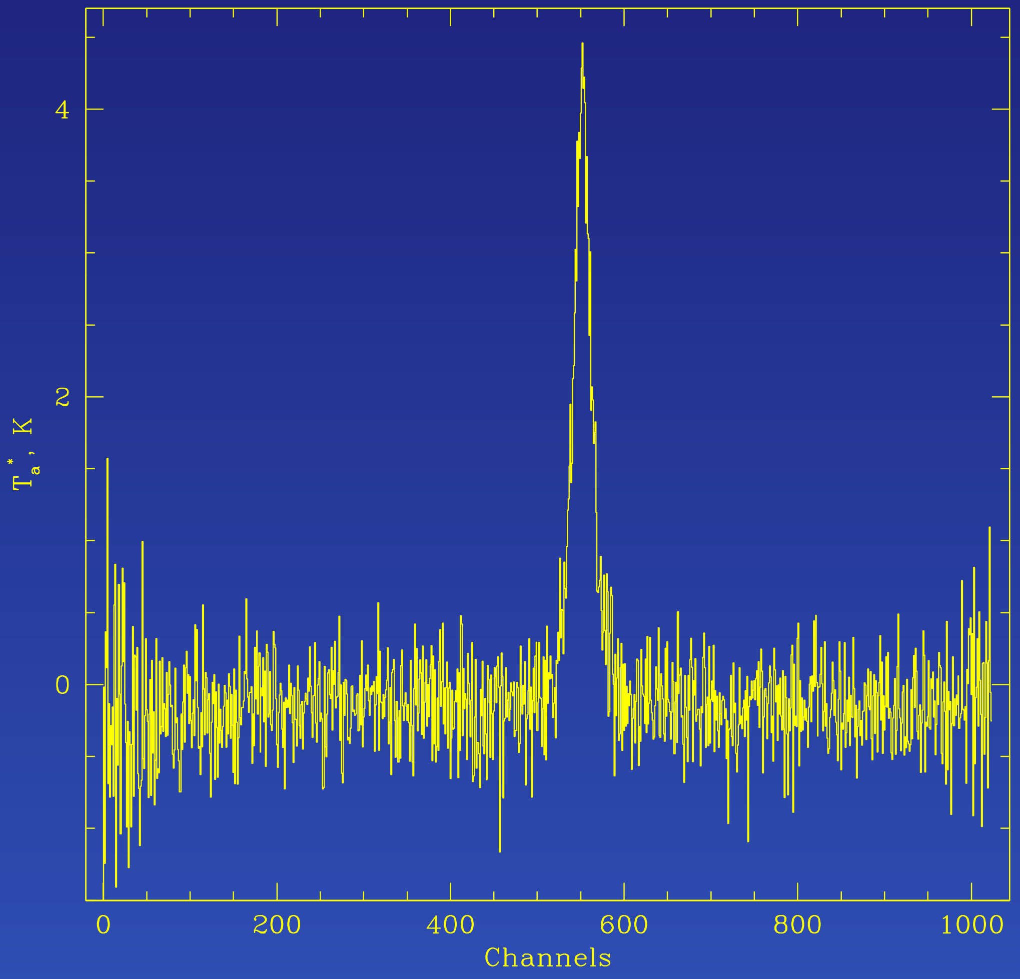
    Line wings can be below threshold

    Broad lines can be significantly oversampled $\longrightarrow$ averaging



- Statistics are calculated for the sample box
  - Linear fit (slope and offset)
  - Noise rms (using 80% of the samples)
- Channel has a signal, if
  - deviation from the fit is greater than a given SNR threshold
- Box is moved to keep the tested channel centred
- Known lines are excluded from statistics
- Edge channels can be rejected

Examples: strong line (old Mopra system)

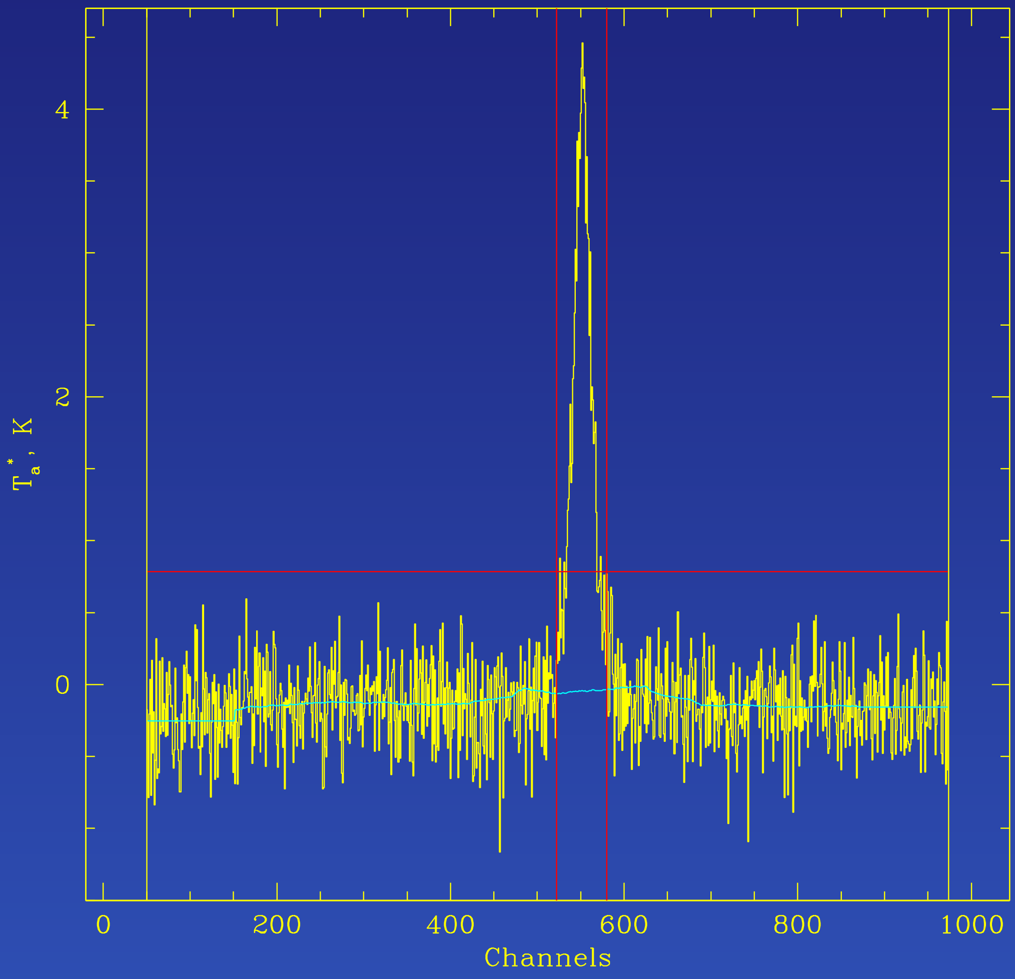


- Rejection of 50 edge channels from each side
- 3σ detection limit

$\downarrow$

- Line is found at 522–580 channels

Examples: strong line (old Mopra system)

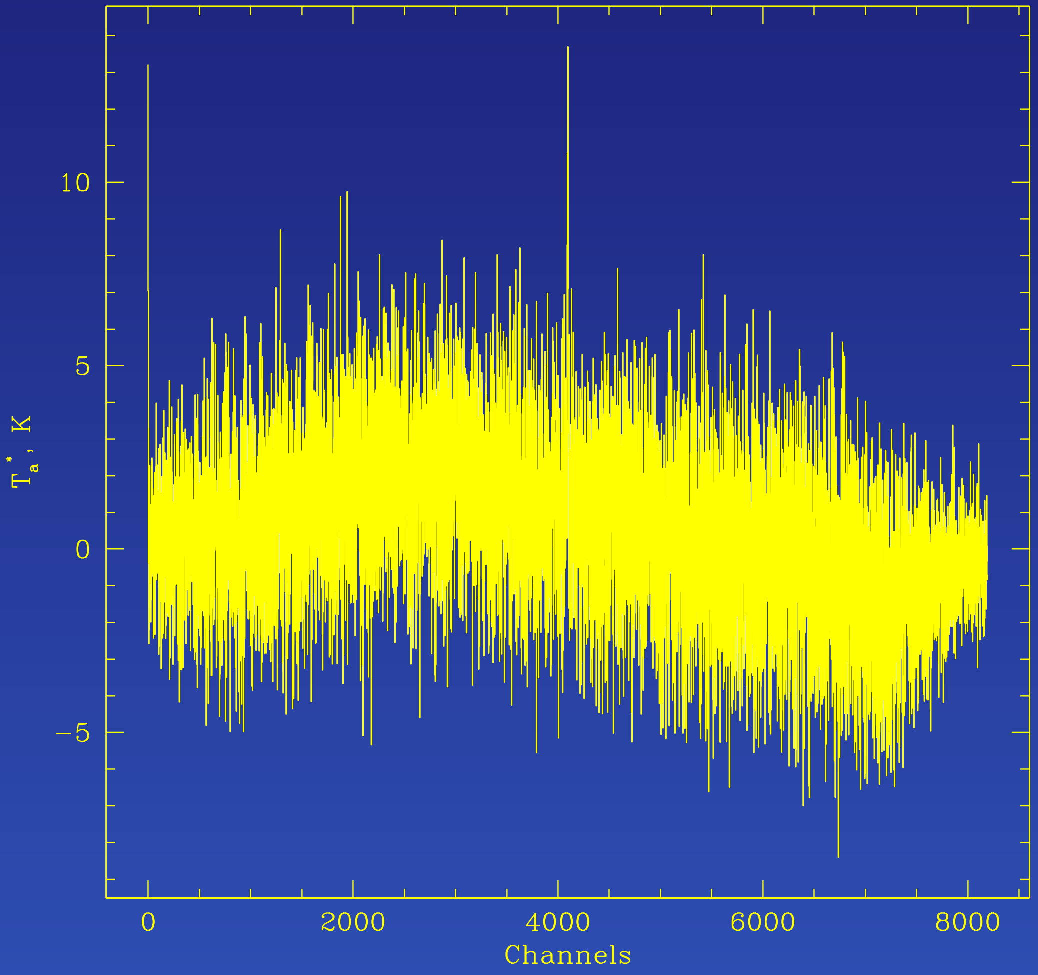


- Rejection of 50 edge channels from each side
- 3σ detection limit

$\downarrow$

- Line is found at 522—580 channels

Examples: weak line

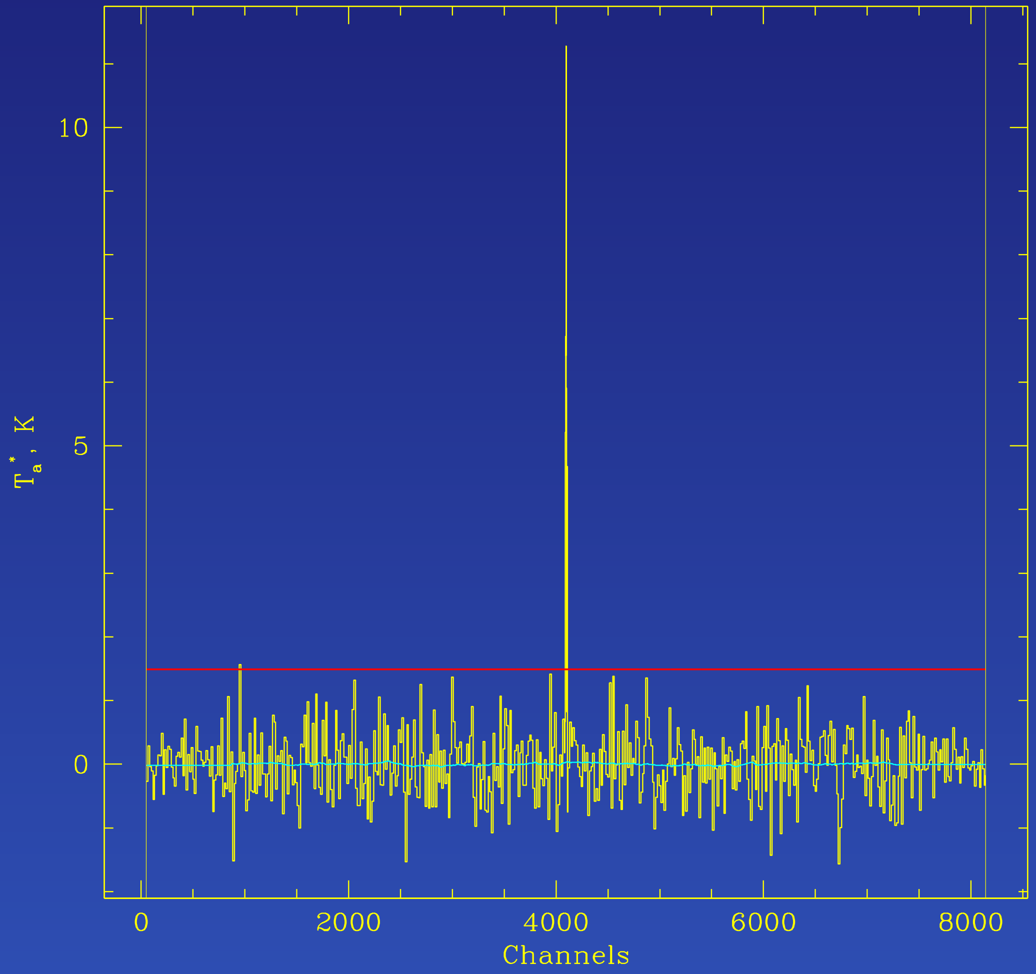


- This example demonstrates averaging of adjacent channels
- Without averaging the number of spectral channels above the threshold does not qualify for a detection

$\downarrow$

- Line is found at 4086−4108 channels

Examples: weak line

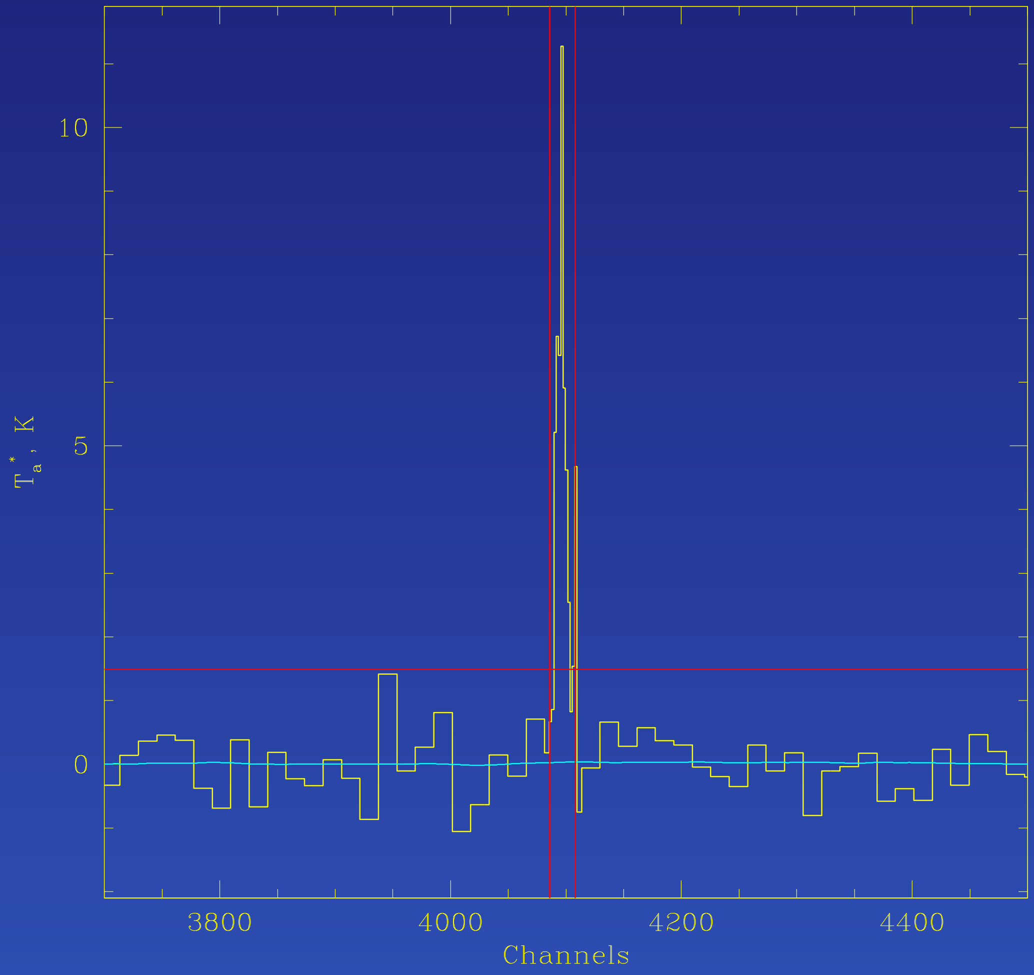


- This example demonstrates averaging of adjacent channels
- Without averaging the number of spectral channels above the threshold does not qualify for a detection

↓

- Line is found at 4086–4108 channels

Examples: weak line

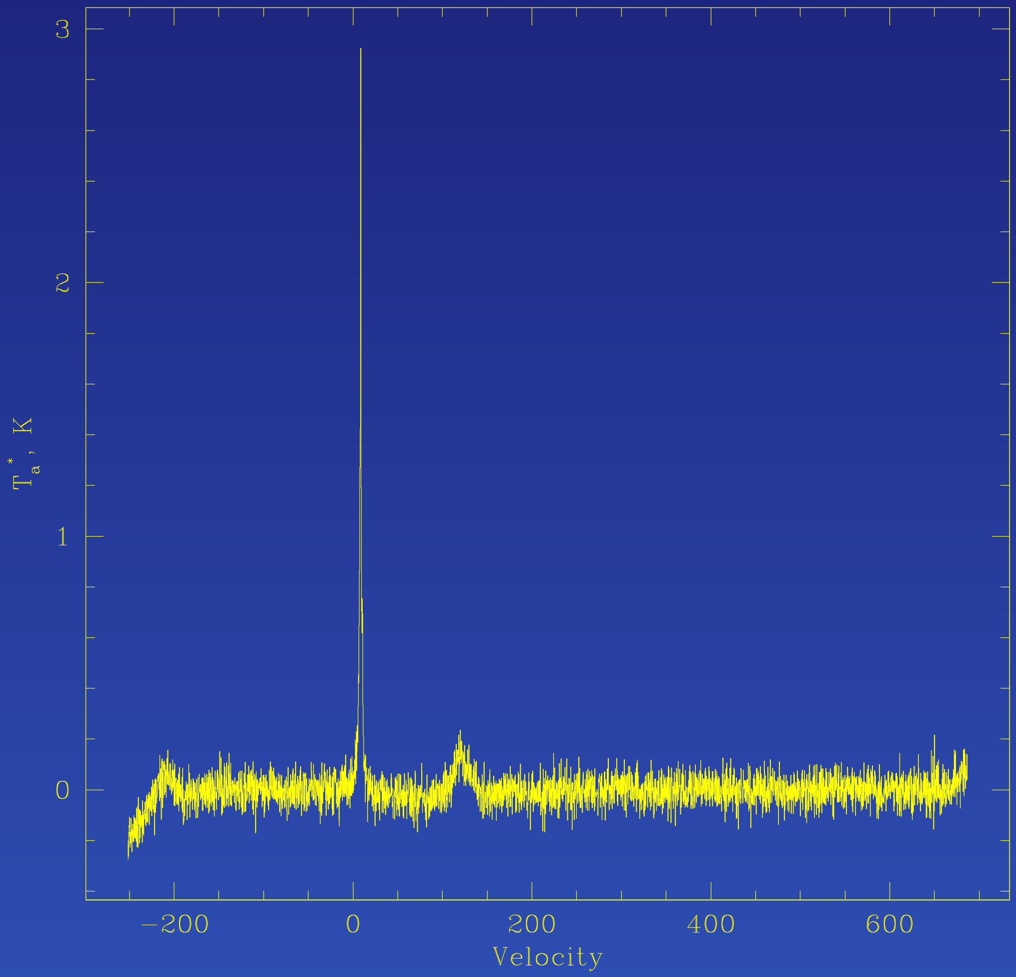


- This example demonstrates averaging of adjacent channels
- Without averaging the number of spectral channels above the threshold does not qualify for a detection

$\downarrow$

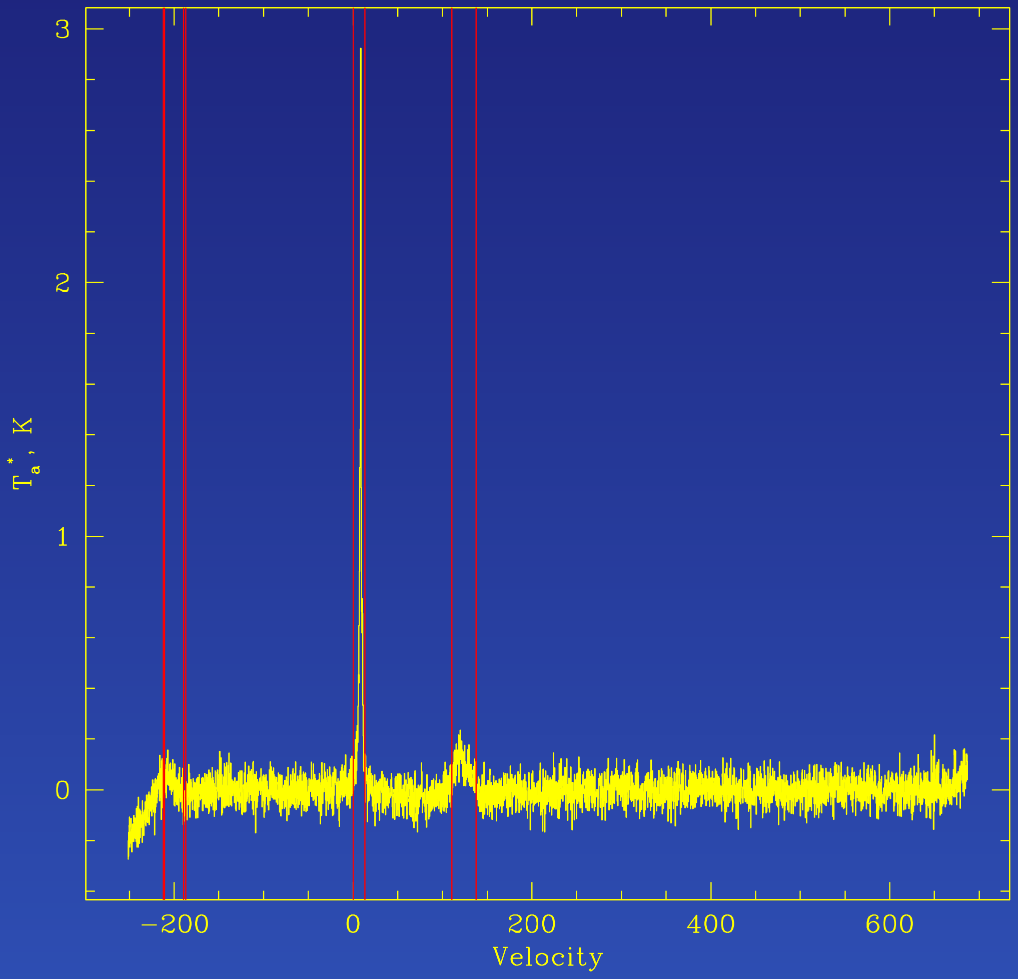
- Line is found at 4086–4108 channels

Examples: MOPS (one zoom band)



- Rejection of 100 edge channels from each side

Examples: MOPS (one zoom band)

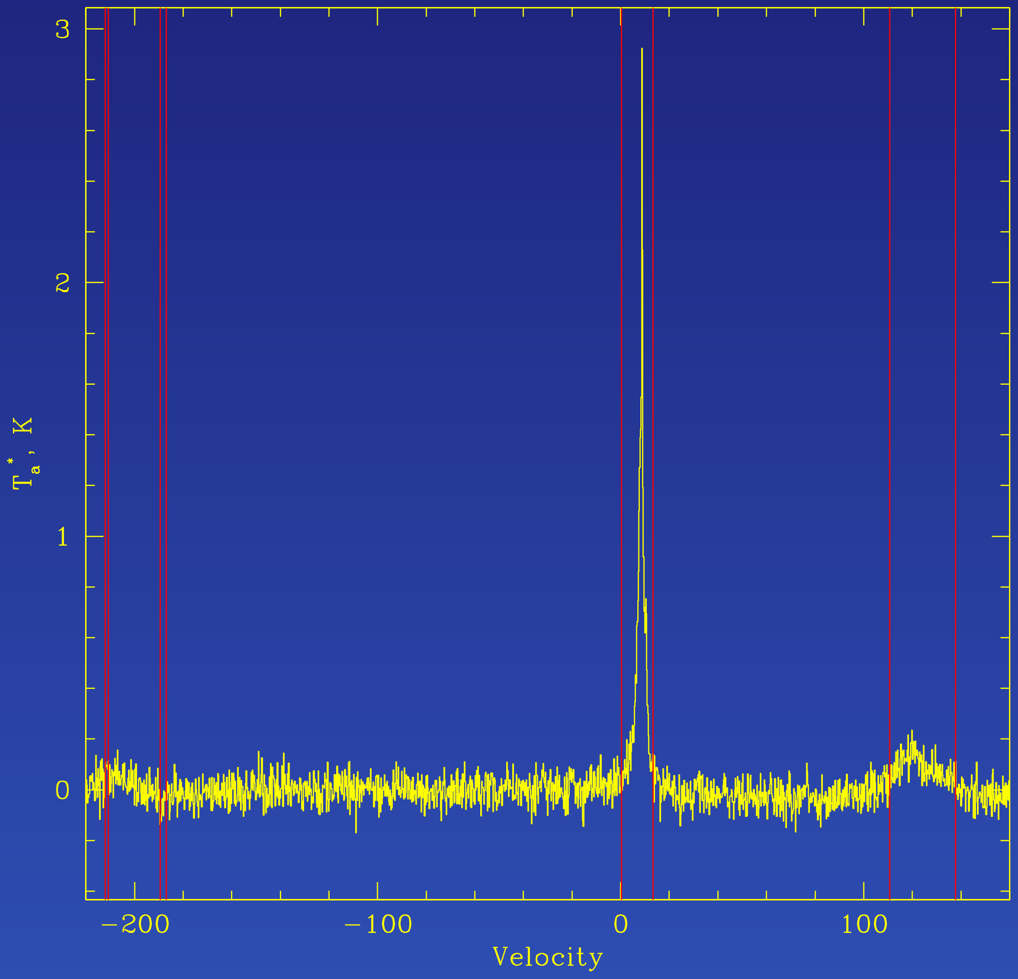


- Rejection of 100 edge channels from each side
- 3σ detection limit, no averaging

$\downarrow$

- Spurious detections near the edge

Examples: MOPS (one zoom band)

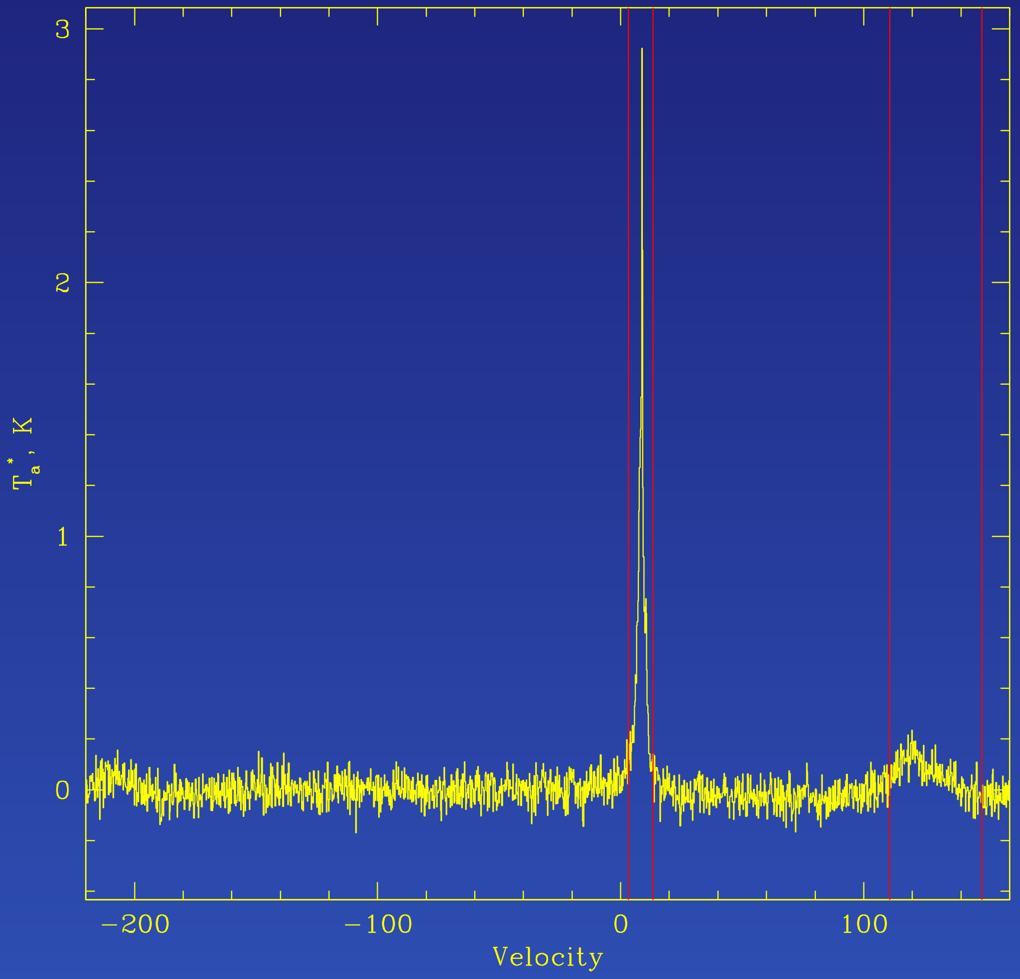


- Rejection of 100 edge channels from each side
- 3σ detection limit, no averaging

$\downarrow$

- Spurious detections near the edge

Examples: MOPS (one zoom band)



- Rejection of 100 edge channels from each side
- 5.2σ detection limit, averaging up to 8 channels

$\downarrow$

- Just two real lines left

Python interface (main asaplinefind methods)

- set_options (threshold, min_nchan, avg_limit, box_size)
  - $\longrightarrow$ threshold: single channel S/N ratio. Default is $\sqrt{3}$
  - $\longrightarrow$ min_nchan: minimum number of consecutive channels deviating more than the threshold required for detection. Default is 3.
  - $\longrightarrow$ avg_limit: maximum number of channels to average during the search for broad lines. Default is 8.
  - $\longrightarrow$ box_size: size of the sample box as a fraction of the bandwidth. Default is 0.2.
- set_scan (scan)
  - $\longrightarrow$ scan: scantable to use. Note, required Beam/IF/Polarisation should have already been selected before calling this method if the scantable has more than one.
- find_lines (nRow, mask, edge)
  - $\longrightarrow$ nRow: data row in the scantable to work with. Default is 0.
  - $\longrightarrow$ mask: optional mask (parts of the spectrum to ignore)
  - $\longrightarrow$ edge: number of edge channels to reject. Default is (0,0).
  - $\longleftarrow$ return: number of lines found
- get_ranges (defunits)
  - $\longrightarrow$ defunits: if True scantable units are used in the output, otherwise channels.
  - $\longleftarrow$ return: list of first and last channel/velocity for each line

Python example

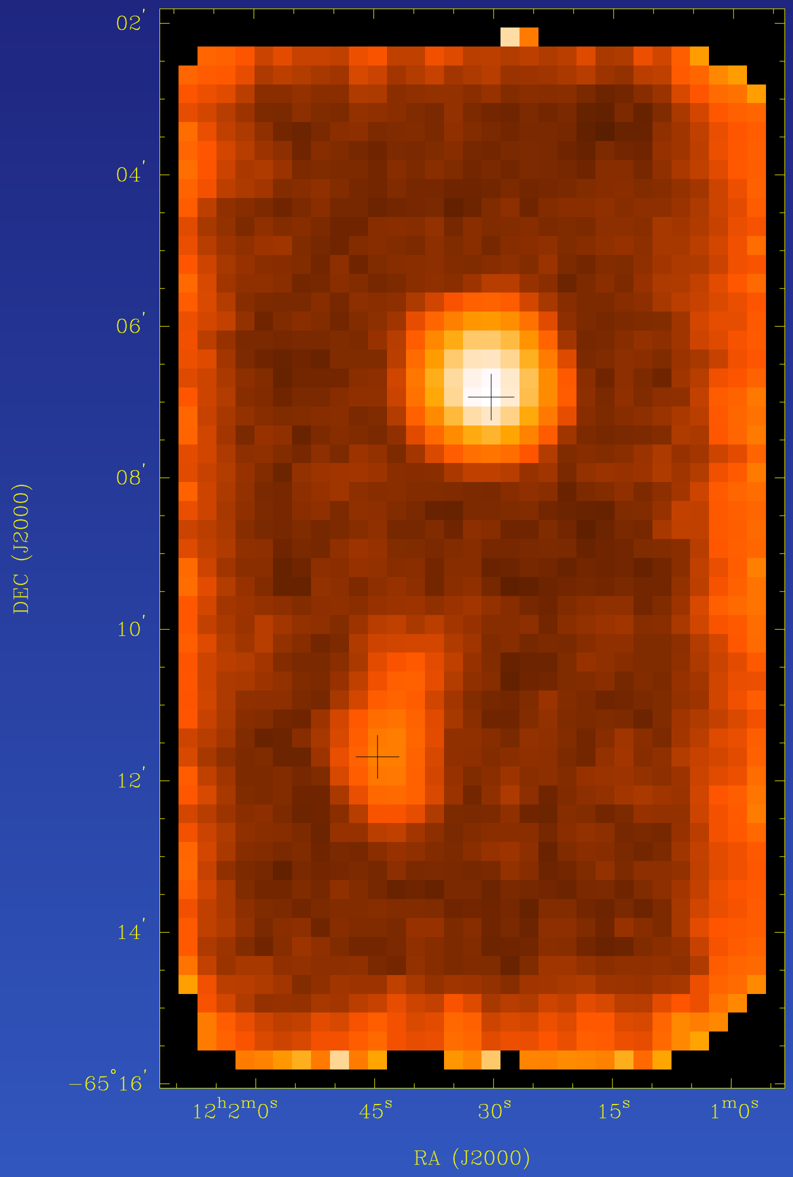
```
# Line search
fl=asaplinefind.linefinder()
fl.set_scan(scan)
fl.set_options(threshold=3)
nlines=fl.find_lines(edge=(200,100))
if nlines!=0:
      print "Found",nlines,"spectral lines:", fl.get_ranges()
else:
      print "No lines found!"

# automatic baselining
scan.auto_poly_baseline(order=3)
```

Search for spectral line sources in 3D cubes

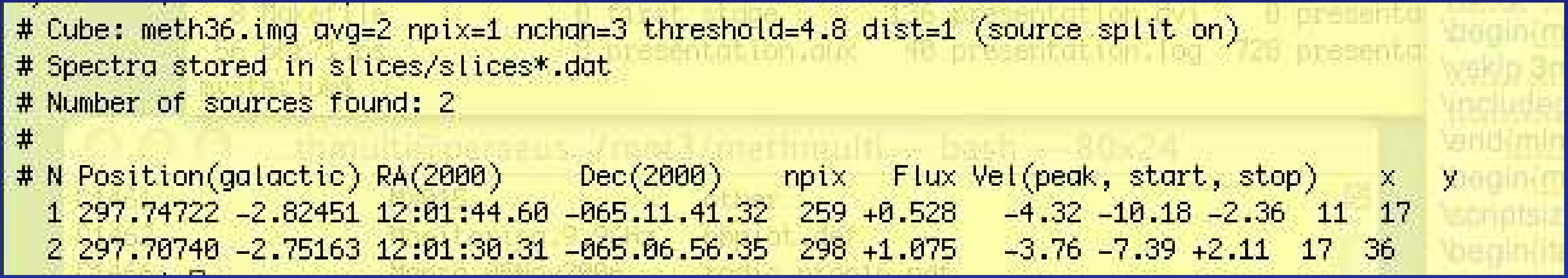
- Separate program written for Parkes Methanol Multi-beam Survey
- ASAP 1D routine is executed for every spatial pixel in the cube.
  - If any lines are detected, adjacent pixels are tested until the edge of the image is reached or there is no detection
  - Such isolated groups of pixels are combined into sources by merging overlapping velocity ranges
  - If there is a notable spatial offset between the peak positions, two or more sources will be formed even if they have close velocities.
  - The output includes peak positions, fluxes and velocities for each source as well as the lowest and highest velocity.
  - Optionally, the velocity ranges for all spectral components can be listed and the slices taken at the peak spatial pixel of each source can be exported
- The program does not try to decompose sources into components!
- Depending on the threshold there is always a number of spurious detections due to statistics

Example: Mopra OTF map of BHR71



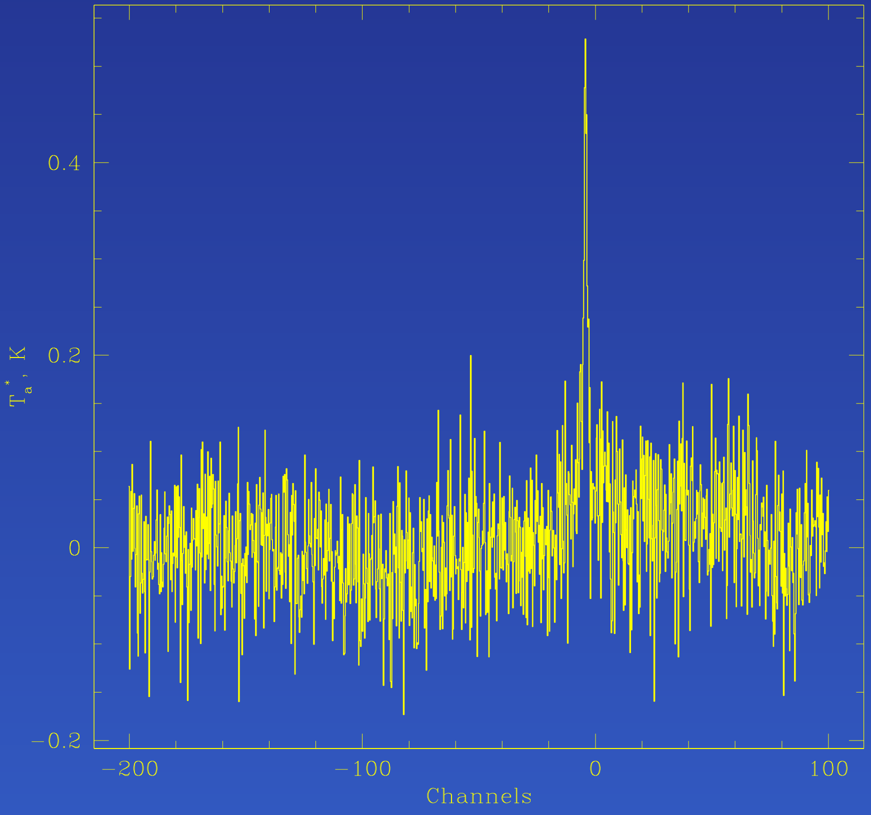
- Program found just two sources
- The southern source definitely has some structure, but was found as a single source because there was no significant velocity dispersion.

Example: Mopra OTF map of BHR71

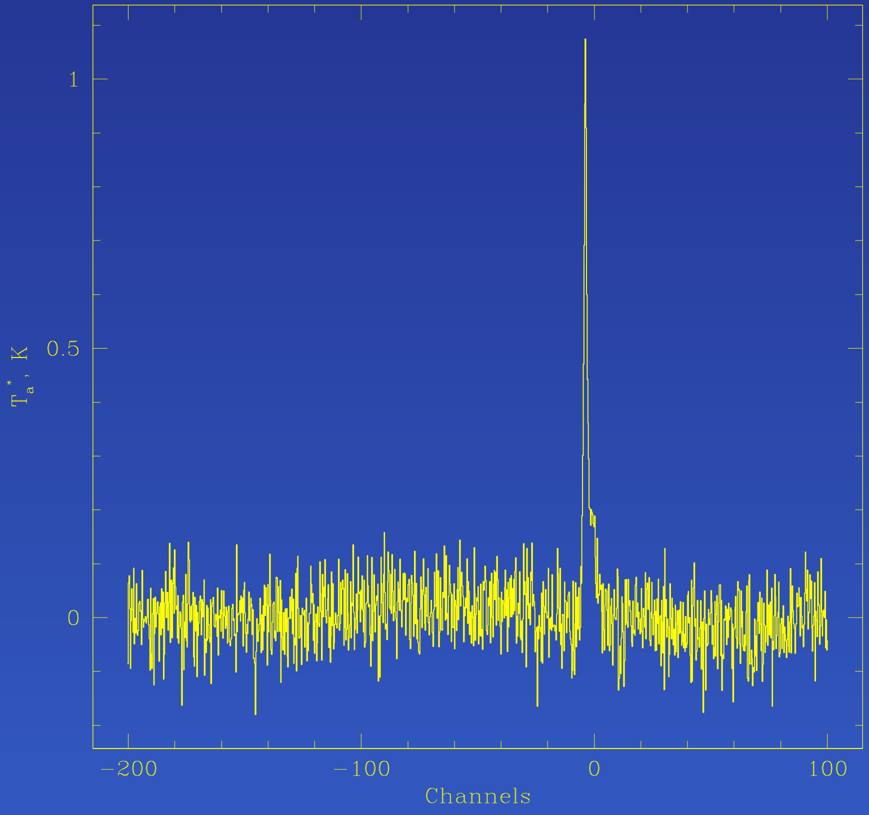


```
# Cube: meth36.img avg=2 npix=1 nchan=3 threshold=4.8 dist=1 (source split on)
# Spectra stored in slices/slices*.dat
# Number of sources found: 2
#
# N Position(galactic) RA(2000)     Dec(2000)     npix   Flux Vel(peak, start, stop)    x    y
  1 297.74722 -2.82451 12:01:44.60 -065.11.41.32  259 +0.528   -4.32 -10.18 -2.36   11   17
  2 297.70740 -2.75163 12:01:30.31 -065.06.56.35  298 +1.075   -3.76 -7.39 +2.11   17   36
```

Source 1



Source 2



Summary

- A number of solutions exists to search for spectral lines in on-off spectra and spectral line sources in 3D cubes
- Spectral line finder is a part of standard ASAP distribution. Just type help asaplinefind to find how to use it
- Source finder is not (yet) released outside of the Methanol Multibeam group, but I am open for collaboration if there is an interest in this software for other projects
- Other source finders exist, e.g. duchamp, which will be used for ASKAP