Weeds

Documentation Manual
August 24, 2010

Sébastien Maret
Laboratoire d'Astrophysique de Grenoble

Pierre Hily-Blant
Laboratoire d'Astrophysique de Grenoble

Jérôme Pety
Institut de Radio-Astronomie Millimétrique

Sébastien Bardeau
Institut de Radio-Astronomie Millimétrique

Emmanuel Reynier
Institut de Radio-Astronomie Millimétrique

This manual documents Weeds, an extension of the CLASS radio astronomy software to reduce and analyse spectral surveys or spectral line observation with large bandwidths. Please report any errors in this manual to `weeds-devel@obs.ujf-grenoble.fr`. Related information is available in the CLASS manual.

1 What is Weeds?

Weeds is an extension of the CLASS radio astronomy software which is intended to ease the analysis of spectral surveys or spectral lines observations with large bandwidths. Weeds provides several commands to identify lines on a spectrum using spectral catalogs, which are accessed on-line using the VO-compliant SLAP protocol. In addition, Weeds can perform a simple modeling of the observed spectra, under the assumption of local thermodynamic equilibrium. Together with the catalog query commands, the modeling tool allows to efficiently identify line on crowded spectrum, where the line overlap can be important.

Weeds development was inspired by other packages, such as XCLASS and CASSIS. Both Weeds and XCLASS are extensions of the widely used CLASS spectra reduction software, which is part of GILDAS; this allows to reduce and analyze spectral surveys within the same environment. However, unlike XCLASS, Weeds is distributed with GILDAS as a contributed extension, and it uses the most recent CLASS version (CLASS90). Another difference between Weeds and both XCLASS and CASSIS is that Weeds access to spectral catalogs on-line. Consequently any changes in these catalogs (e.g. line additions) are readily available in Weeds. Nevertheless, Weeds can make a local copy a spectral line catalog, for offline searches. This is useful when you want to use Weeds with no (or a slow) internet connection. All this make Weeds a lightweight, easy-to-install and efficient alternative to other spectral line analysis packages.

Weeds is named after some species that are detected in the interstellar medium, such as methanol, methyl formate or dimethyl ether. Because of internal motions, these species have an extremely rich line spectra at millimiter and sub-millimeter wavelengths. Often, in order to analyze a spectral survey, one need to identify the weeds first before picking-up the flowers (that is the lines of other species of interest). This is precisely the purpose of this extension.

This document describes how to use Weeds to analyse a spectral survey. It is written as a tutorial; each of step of the data is explained through examples.

2 Getting started

Starting from March 2010, Weeds is distributed with GILDAS. Provided that you have installed GILDAS with the Python extension (named PyGILDAS)[1] you should have Weeds already installed on your computer. If this is not the case, please refer to the GILDAS documentation for instructions on how to install PyGILDAS.

Let us first start CLASS from a terminal:

```
% class
% class
GILDAS Version: dev (28jul10 11:09) (x86_64-darwin-gfortran) source tree
Python 2.6.5 (r265:79063, Jul 20 2010, 00:26:30)
[GCC 4.2.1 (Apple Inc. build 5659)] on darwin
Importing all SIC variables into Python...
... done.

 * Welcome to CLASS

 * Loaded modules
    atm
    sic (J.Pety, S.Bardeau, S.Guilloteau, E.Reynier)
    greg (J.Pety, S.Bardeau, S.Guilloteau, E.Reynier)
    ephem (F.Gueth, J.Pety)
    class (J.Pety, P.Hily-Blant, S.Guilloteau)
```

[1] Weeds requires Python version $\geq$ 2.6 or newer.

```
 * Loaded extensions
    weeds (S.Maret, P.Hily-Blant, J.Pety, S.Bardeau, E.Reynier)

(...)
LAS>
```

Note that the welcome message mention that the Weeds extension as been loaded. The extension defines a language, named `weeds\`. This language has a number of command, whose list can be obtained by typing:

```
LAS90> help weeds\
USER\WEEDS = "

    Weeds is a CLASS extension to analyse spectral surveys or
    spectral lines observations with large bandwidths. It provides
    several commands to identify lines on a spectrum and to model it.

    Available commands:
        DBSELECT  Select an atomic and molecular line database
        DBCACHE   Cache an atomic and molecular line database
        LID       Identify lines on the current spectra
        LFIND     Find lines from a species within a frequency range
        LLIST     List lines from the line index
        LGET      Get a line from the line index
        LPLOT     Plot a line from the current line index
        MODSOURCE Model the emission of a source at the LTE
        MODSHOW   Show the results of MODSOURCE

   For more information of each command, type 'help <command>'
```

Each of the command provided by Weeds as an extensive documentation. Type `help` followed by the command name to read it.

3 Selecting a spectral database

The first step to analyze our spectra is to select a spectral line database. Weeds can access both the JPL database for molecular spectroscopy and the Cologne Database for Molecular Spectroscopy (CDMS). To select a database, type `dbselect`, followed by the name of the database. For example:

```
LAS90> dbselect jpl
I-DATABASE, jpl database selected (online).
```

or:

```
LAS90> dbselect cdms
I-DATABASE, cdms database selected (online).
```

By default, the CDMS database is selected.

As mentioned already, Weeds makes online queries to the molecular databases. However, it is sometime helpful to make offline searches in a database, for example if you are on travel with no (or a slow) internet connection. Weeds allow us to download a part of – or even the entire – spectral catalog from a database. This catalog is stored in a cache file, and can be used later for offline searches. Caching the CDMS database between 80 and 130 GHz is done as follows:

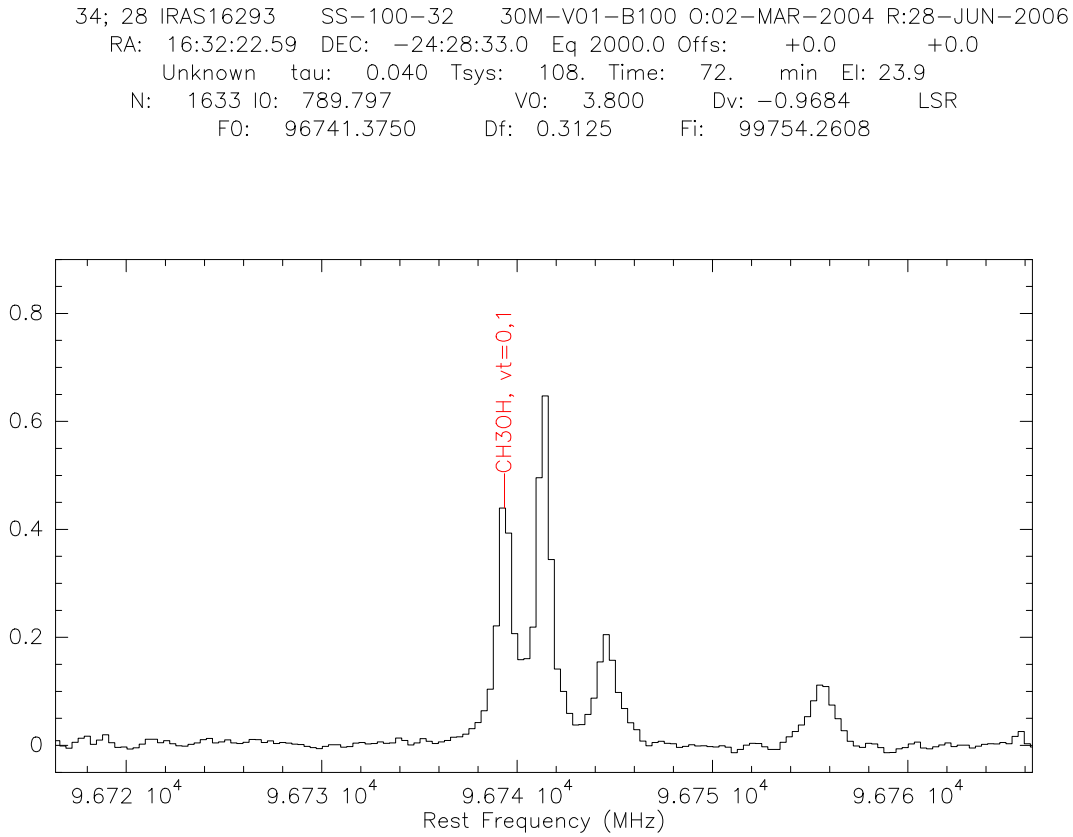

Figure 1: Example spectra obtained with the IRAM-30m telescope towards the IRAS 16293 protostar. One of the methanol transition has been labeled with lid.

```
LAS90> dbcache cdms /lower 80e3 /upper 130e3
I-DBCACHE, Caching cdms database...
Downloading lines:      100% |############################################################| Time: 00:00:18
Downloading part.func.: 100% |############################################################| Time: 00:00:54
I-DBCACHE, Done.
I-DBCACHE, cdms database selected (cached).
```

Depending on the speed of your internet connection and the frequency range you have selected, caching a database may take from a few seconds to a few tens of minutes, so please be patient! In particular, caching the JPL database may take a while, because of technical limitations with this database; caching the CDMS database is much faster.

Once you have cached a database, you may select it with the `/cache` option of the `dbselect` command[2]:

```
LAS90> dbselect cdms /cache
I-DBSELECT, cdms database selected (cached).
```

A final note on caching a database: every time the same database is cached, the older cache file is overwritten; there is no update of an old cache file. However, caches of different databases are stored in different files, so caching the JPL database will not overwrite an older cache of the CDMS.

4 Identifying lines on a spectrum

Fig. 1 show an example scan from a spectral survey. Spectral surveys typically have many different scans of this kind, each corresponding to a different frequency setting. The scan on the figure show

[2]After caching a database, Weeds switches automatically to the cached version.

several well detected lines that we want to identify. This can be done with `lid` which allow us to select a line on the scan with the mouse. Clicking on the left button of the mouse over some part of the spectrum selects a line. Then the command searches in the database (which has been selected previously with the `dbselect` command, as explained above) for lines within 1 MHz of the selected region(s) of the spectrum. For each line found, the command mark the rest frequency of the line on the scan, and a table displaying the line frequency, upper level energy, Einstein coefficient and quantum numbers and is printed:

```
LAS90> lid
-------------------------------------------------------------------------------------------
| Species          |  Freq. (GHz) | Eup (K) | Aul (s-1) | Upper level    -- Lower level    |
-------------------------------------------------------------------------------------------
| CH3OH, vt=0,1    |    96.739362 |    12.5 |  2.56e-06 | 2 -1  2  0     -- 1 -1  1  0     |
-------------------------------------------------------------------------------------------
```

Most of the time, several line candidates are found for a given frequency. For example, using `lid` to search for lines in the catalog around the brightest line in Fig. 1, we obtain:

```
LAS90> lid
-------------------------------------------------------------------------------------------
| Species          |  Freq. (GHz) | Eup (K) | Aul (s-1) | Upper level    -- Lower level    |
-------------------------------------------------------------------------------------------
| n-C3H7CN         |    96.741208 |   158.0 |  7.60e-06 | 12 310 1       -- 11 2 9 1       |
| C4H, v7=1        |    96.741314 |   214.2 |  3.74e-06 | 10 111         -- 9-110          |
| CH3OH, vt=0,1    |    96.741375 |     7.0 |  3.41e-06 | 2  0  2 +0     -- 1  0  1 +0     |
-------------------------------------------------------------------------------------------
```

Searches in the database may be refined to lines with an upper level energy lower than a given value using the `/energy` option, or to lines with an Einstein coefficient greater than a given value with the `/aeinstein` option. The `/species` option allow to restrict searches to a given species.

It should be noted `lid` prints the line upper and lower level quantum numbers exactly as they appear in the selected database. We refer the user to the documentation of these databases for the meaning of these quantum numbers.

5 Using spectral line indexes

Once we have identified a line from some species in our spectral survey, we may want to search for other lines of that species at different frequencies. For this, Weeds allows to create a line index, which work in a similar way as CLASS scan indexes. Scan indexes are created by the CLASS `find` command and they are listed by the `list` command. A given scan from the current index can then be loaded and plotted using the `get` and `plot` commands. Likewise, line indexes are created with the `lfind` command, listed with the `llist` command, and a given line may be loaded and plotted using the `lget` and `lplot` commands.

For example, let's assume that we have identified a methanol line on the spectrum shown on Fig. 1, and that we want to see if other methanol lines are present in other scans. The methanol line index is built with:

```
LAS90> lfind "CH3OH, vt=0,1" 80e3 120e3
I-LFIND, 137 lines found.
```

and then listed with:

```
LAS90> llist
-------------------------------------------------------------------------------------------
```

```
| Index | Species          |  Freq. (GHz) | Eup (K) | Upper level    -- Lower level    |
-------------------------------------------------------------------------------------------
|     1 | CH3OH, vt=0,1    |    80.993241 |   102.8 | 7  2  6 -0     -- 8  1  7 -0     |
|     2 | CH3OH, vt=0,1    |    81.147462 |  1130.8 | 21 +8 13  1    -- 21 +6 15  1    |
|     3 | CH3OH, vt=0,1    |    81.318424 |   381.8 | 17 -2 16  0    -- 17 +2 15  0    |
|     4 | CH3OH, vt=0,1    |    81.653080 |   493.1 | 18 +4 14  0    -- 19 +3 16  0    |
(...)
|   137 | CH3OH, vt=0,1    |   119.271820 |   481.5 | 12 +0 12  1    -- 11 -3  8  1    |
-------------------------------------------------------------------------------------------
```

The command print all methanol lines comprised between 80 and 120 GHz, ordered by increasing rest frequencies. It also builds an internal index containing all these lines. Note that the species name must be typed exactly as it appear in the database and between double quotes, i.e. `"CH3OH, vt=0,1"` in this case.

Like for scans, each line is associated with an entry number, that we can use to load and to plot the line. Since, by default, lines are ordered by frequency, subsequent entry numbers will correspond to lines with increasing frequencies. However, if we search for lines of a given species, it usually a good idea to look for lines with the lowest upper energy levels, because we expect them to be brighter than higher lying lines. For this, we can re-order the lines by increasing upper level energies using the `/order e` option:

```
LAS90> lfind "CH3OH, vt=0,1" 80e3 120e3 /order e
I-LFIND, 137 lines found.
LAS90> llist
-------------------------------------------------------------------------------------------
| Index | Species          |  Freq. (GHz) | Eup (K) | Upper level    -- Lower level    |
-------------------------------------------------------------------------------------------
|     1 | CH3OH, vt=0,1    |    96.741375 |     7.0 | 2  0  2 +0     -- 1  0  1 +0     |
|     2 | CH3OH, vt=0,1    |    96.739362 |    12.6 | 2 -1  2  0     -- 1 -1  1  0     |
|     3 | CH3OH, vt=0,1    |   108.893963 |    13.1 | 0 +0  0  0     -- 1 -1  1  0     |
|     4 | CH3OH, vt=0,1    |    96.744550 |    20.1 | 2 +0  2  0     -- 1 +0  1  0     |
(...)
|   137 | CH3OH, vt=0,1    |    89.801138 |  1665.5 | 25+10 15  1    -- 25+11 14  1    |
-------------------------------------------------------------------------------------------
```

Since we have already identified the two first lines on the index, let's have a look at the third one:

```
LAS90> lget 3
I-SCANFIND, Found scan #7182
I-LGET, Found line frequency in the current scan index
-------------------------------------------------------------------------------------------
| Index | Species          |  Freq. (GHz) | Eup (K) | Upper level    -- Lower level    |
-------------------------------------------------------------------------------------------
|     3 | CH3OH, vt=0,1    |   108.893963 |    13.1 | 0 +0  0  0     -- 1 -1  1  0     |
-------------------------------------------------------------------------------------------
```

The `lget` command does several things. It first searches in the scan index for the first scan that covers the reset frequency of the line[3]. If it finds one, the scan is loaded in the CLASS buffers, and the scan frequency is then modified so that the velocity axis is centered at the line rest frequency. We can now plot the line:

```
LAS90> lplot
```

which gives the spectrum shown on Fig. 2.

[3]It is important to note that `lget` searches for the first scan covered by the rest frequency. Other scans covering the same frequency are ignored. Thus if you have a spectral survey with several scans covering the same frequency, it is recommended to average these scan together with the CLASS `average` beforehand.

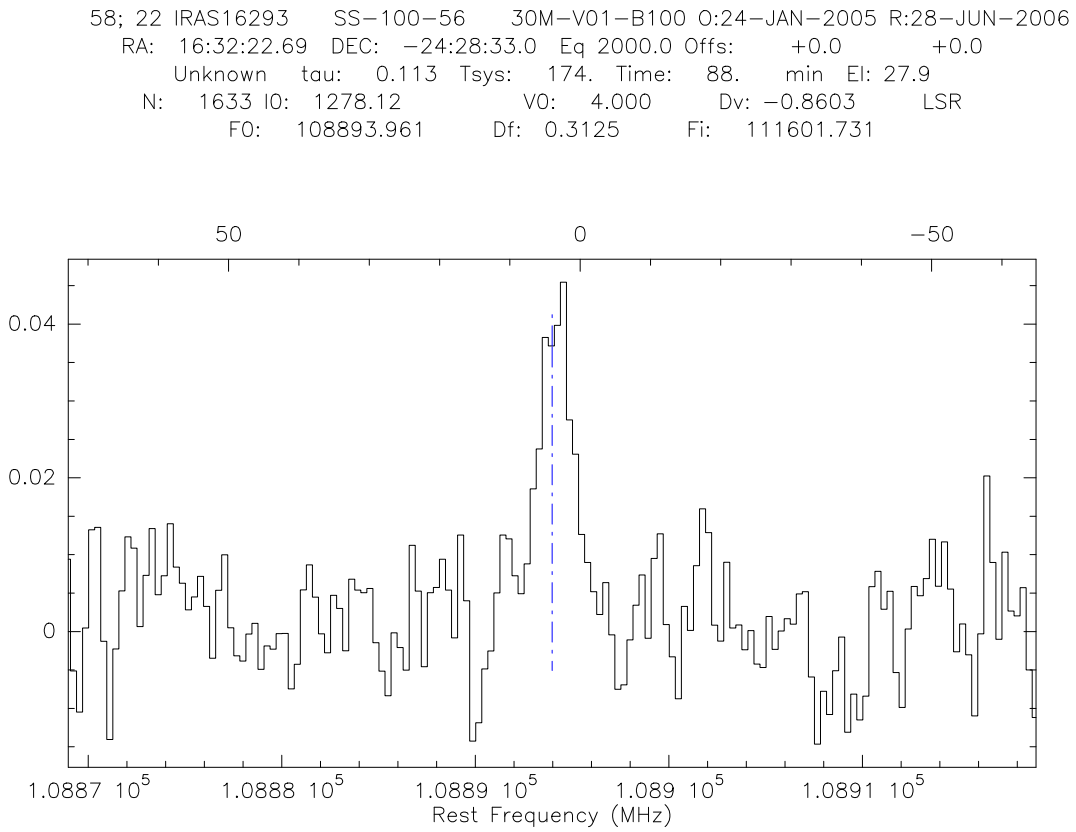

Figure 2: Line displayed with lplot command. The blue vertical line show the rest frequency of the line. The upper x-axis show the velocity offset from the line rest frequency.

The `lget` and `lplot` commands allow to quickly "navigate" in a spectral survey to look for the different lines of a given species. `lget f` will load the first scan of the index, then `lget n` will get the next one. `lget p` will get the previous one, etc. If you get lost at some point, you can always type `llist` to display the line index again.

6 Modeling a spectrum

Experience shows that in order to identify a species in a spectral survey securely, it is often necessary to make a basic modeling of the line candidates. For example, you may want to check if the relative intensities of each line candidates imply a reasonable kinetic temperature. You may also want to check that other lines with similar predicted intensity are also detected.

Weeds allows to compute the emission of a source under the assumption of local thermodynamical equilibrium[4]. The source is assumed to have, for each species, a given column density, excitation temperature, turbulent velocity, systemic velocity and size. Several components (with e.g. different temperatures and/or sizes) can be added. Weeds will compute the emission of these various components – taking into account the line opacity and the beam dilution factor – and it will display them on the observed spectrum. The different source parameters can be adjusted until a good match between the model and the observation is obtained.

Let us go back to the methanol lines that we have identified on our spectrum (see Fig. 2). The emission can be modeled using the `modsource` command, which takes two arguments: the name of file containing the source parameters, and the size of the antenna that we have used for the observations, in meters. To model the emission, we need to use the JPL database, because the

[4]See Maret, Hily-Blant, Pety et al., *Astronomy & Astrophysics*, in prep. (2010) for the formula used.

CDMS does not provides partition functions:

```
LAS90> dbselect jpl
I-DBSELECT, jpl database selected (online).
```

Our model file, that we have named `iras16293.mdl`, looks like this:

```
# Source model for IRAS16293
#
# species       Ntot    Tex   theta   v_off   delta_v
#              (cm-2)  (K)   (‘‘)    (km/s)  (km/s)
#
CH3OH          2.0e15  10    10      0       3.0
```

Lines that start with a `#` are comments: they are ignored. The last line gives the name of the species, its column density, excitation temperature, the size of the emission in arc seconds, the offset velocity (with respect to the source velocity in the class file header), and the line width. The spectrum, as it would be observed with the IRAM-30m antenna, can be computed with:

```
LAS90> modsource iras16293.mdl 30 /verbose
I-MODSOURCE: 4 CH3OH lines found in the frequency range
I-MODSOURCE: Partition function at 10.0 K is 41.1
-----------------------------------------------------------
| Freq. (GHz)  | Eup (K) |  gu  | Aul (s-1) |    tau    |
-----------------------------------------------------------
|    96.739358 |    12.5 |    5 |  3.79e-06 |  5.73e-01 |
|    96.741371 |     7.0 |    5 |  5.05e-06 |  1.34e+00 |
|    96.744545 |    20.1 |    5 |  5.05e-06 |  3.61e-01 |
|    96.755501 |    28.0 |    5 |  3.89e-06 |  1.26e-01 |
-----------------------------------------------------------
I-MODEL,   Blanking value:   -1000.0000
I-MODSOURCE, Model has been stored in memory
```

For efficiency reasons, the command computes the spectrum over the frequency range covered by the current window only. In our case, four methanol lines have found in the frequency range. Note the `/verbose` option, that prints the frequencies, upper level energy and statistical weight, Einstein coefficient and computed opacity at the line center. The `modsource` command itself does not plot anything; it just stores modeled spectrum is stored into buffers. These buffers can be listed with:

```
LAS90> memorize
I-MEMORY,  Current memories
 OBS
 TA_MODEL
```

The `TA_MODEL` buffer contains the predicted antenna temperature. The buffer can be retrieved with the `retrieve` command. The `OBS` buffer contains the observed spectra, which is saved automatically by `modsource`.

The predicted spectrum can be drawn over the observed spectrum using the `modshow` command[5]. This command gives the spectra shown on Fig. 3. As it can be seen on this figure, our model is in quite good agreement with the observed spectrum.

The `lget` command can be used together with `modshow` to examine other lines, and to check whether our model can also reproduce them. For example, we can examine the third line of the line index with:

[5]Alternatively, one may retrieve the predicted spectrum with `retrieve TA_MODEL`, and plot it over the observed spectrum with `spectrum`. However, it is important to fix y axis range; if the axis is set to `auto`, then the predicted and modeled spectrum will not be on the same scale, and the comparison will be meaningless. The `modshow` command ensure that both the observed and predicted spectrum are on the same scale.

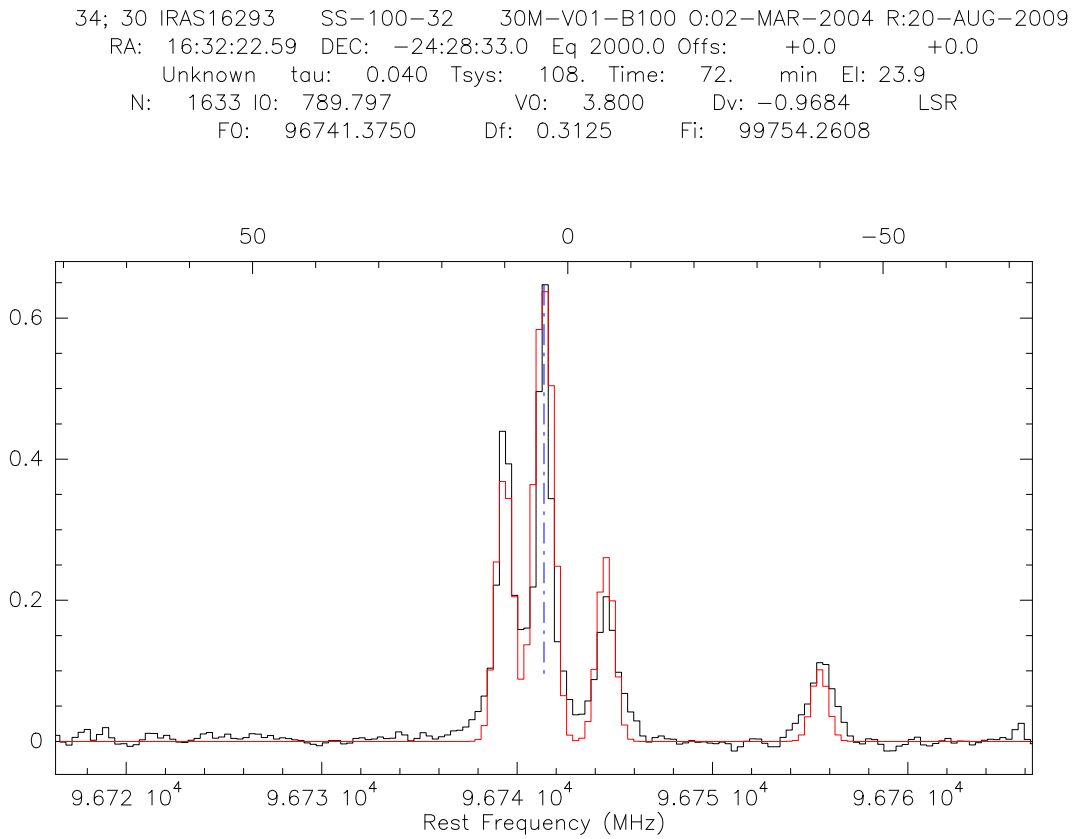

Figure 3: Observed (black histogram) and predicted spectrum (in red) displayed with the modshow command.

```
LAS90> lget 2
LAS90> lplot
LAS90> modsource iras16293.mdl 30
I-MODSOURCE: 3 CH3OH, vt=0,1 lines found in the frequency range
I-MODEL,   Blanking value:   -1000.0000
I-MODEL,   Blanking value:   -1000.0000
I-MODSOURCE, Model has been stored in memory
LAS90> modshow
```

Note that we must use the `modsource` command again, because the command computes a spectrum only over the frequency range covered by the current scan. Scripts can be easily created to loop over line index and examine each of the observed and predicted lines.

7 For more information

This documentation manual covers the basic usage of the Weeds extension. More detailed information on each command is available using the Gildas online help, e.g.:

```
WEEDS\MODSOURCE = "python weeds/modsource.py"
        [WEEDS\]MODSOURCE MODFILE ANTSIZE [/BACKGROUND TBG] [/VERBOSE]

    Model the emission of a source at the  local  thermodynamic  equilibrium
    (LTE). The model paramaters (column density, kinetic temperature, source
    size, line width and velocity offset for each species) are read from MOD-
    FILE.  The modeled antenna temperature is stored in the TA_MODEL memory.
    For convenience the current spectra is also stored in a memory, OBS,  so
```

```
one can easily overplot the result of the model on the observed spectra.

Arguments:
  MODFILE     the name of the file containing the parameters.  The  file
              should one line per species with six columns containing re-
              spectively:

              1) The species name
              2) The column density for this species in cm-2
              3) The exitation temperature in K
              4) The source size in arcsecs
              5) The offset velocity from the source VLSR, in km/s
              6) The line FWHM in km/s

              Lines that start with a "#" are treated as comments.

  ANTSIZE     the antenna size in meters

Options:
  /BACKGROUND set the background temperature to TBG, in  K.  Default  is
              2.73.

  /VERBOSE    prints  partition function, lines frequencies, upper level
              energies and degeneracies, Einstein coefficients and opac-
              ity at the line center.
```

For a more detailed description of the Weeds extension, we refer the interested reader to Maret, Hily-Blant, Pety et al., *Astronomy & Astrophysics*, in prep. (2010). This paper describes the implementation of the extension, and gives more details on the spectroscopic databases that it uses, and explains how spectrum are modeled.

The users interested in the development of the Weeds extension may consult the Weeds developers' wiki[6]. Weeds developers can be contacted by email at `weeds-devel@obs.ujf-grenoble.fr`. Bug reports should also be sent to this list. The list as a public archive[7].

[6] `http://www-laog.obs.ujf-grenoble.fr/twiki/bin/view/Ipag/Collaborations/Weeds`
[7] `http://listes.ujf-grenoble.fr/wws/arc/weeds-devel`

A List of contributing authors

The Weeds project started on mid-2009 at Laboratoire d'Astrophysique de Grenoble. Soon after Institut de Radio Astronomie Millimetrique started to support its development. The following authors have contributed to the project:

- Sébastien Maret (Laboratoire d'Astrophysique de Grenoble)
- Pierre Hily-Blant (Laboratoire d'Astrophysique de Grenoble)
- Jérôme Pety (Institut de Radio Astronomie Millimetrique)
- Sébastien Bardeau (Institut de Radio Astronomie Millimetrique)
- Emmanuel Reynier (Institut de Radio Astronomie Millimetrique)

Weeds is a volunteer project; contributions from other authors are of course welcome. If you are interested in participating in the Weeds development, please go to the developers wiki or contact the developers using the mailing list.

B Acknowledging Weeds developers in your publications

If you use it in your research work, you may acknowledge the developers by citing the following paper Maret, Hily-Blant, Pety et al., *Astronomy & Astrophysics*, in prep. (2010) in your publications.

C WEEDS Language Internal Help

C.1 Language

```
Weeds is a CLASS extension to analyse spectral surveys or spectral lines
observations with large bandwidths. It provides several commands to
identify lines on a spectrum and to model it.

Available commands:
    DBSELECT  Select an atomic and molecular line database
    DBCACHE   Cache an atomic and molecular line database
    LID       Identify lines on the current spectra
    LFIND     Find lines from a species within a frequency range
    LLIST     List lines from the line index
    LGET      Get a line from the line index
    LPLOT     Plot a line from the current line index
    MODSOURCE Model the emission of a source at the LTE
    MODSHOW   Show the results of MODSOURCE

For more information of each command, type 'help <command>'
```

C.2 DBCACHE

```
    [WEEDS\]DBCACHE DATABASE [/LOWER FREQ] [/UPPER FREQ] [/STEP DFREQ]

Make a local copy (cache) of a database. This is done by fetching the
entire (or some part of) the database, and storing its content on the
disk.

Arguments:
    DATABASE  the database name. Currently supported database are:
              1) "cdms":  the Cologne Database for Molecular Spectrocopy
                 (CDMS)
              2) "voparis: the Paris VO observatory interface to the
                 CDMS database
              3) "jpl": the JPL molecular spectroscopy database

Options:
    /LOWER    minimum line frequency to be stored in the cache, in MHz
              (default 0)

    /UPPER    maximum line frequency to be stored in the cache, in MHz
              (default 2e6 MHz, i.e. 2 THz)

    /STEP     frequency interval for each database query. If one obtain
              a timeout error or a message saying that the database re-
              turned two many lines, one can set this to a lower value
              (default 1e4 MHz)
```

C.3 DBSELECT

```
[WEEDS\]DBSELECT DATABASE [/CACHE]
```

Select the database to use for line identification and modeling.

Arguments:

DATABASE the database name. Currently supported database are:
1) "cdms": the Cologne Database for Molecular Spectrocopy (CDMS)
2) "voparis: the Paris VO observatory interface to the CDMS database
3) "jpl": the JPL molecular spectroscopy database

Options:

/CACHE use the local copy (cache) of the database, if it exists. A cache may be created with DBCACHE.

C.4 LFIND

```
[WEEDS\]LFIND [SPECIES [FMIN FMAX]] [/ORDER f|e]
```

Find the lines from a species within a frequency range and create a line index. The command look the lines in the database selected with the DATABASE SELECT command. Lines in the index may be ordered by frequency or by upper level energy.

Arguments:

SPECIES the name of the species. It can include the * wildcard, but only when using the database cache.

FMIN the minimum frequency in MHz.

FMAX the maximum frequency in MHz

Options:

/ORDER sort the lines in the index according to this criterion. Accepted values are:

"f" order the lines by increasing frequency. This is the default.
"e" order the lines by increasing upper level energy.

C.5 LGET

```
[WEEDS\]LGET [OPTION] [INDEX]
```

Get a line from the line index, and find the corresponding scan in the index. The command works in a similar fashion than GET: for example LGET F will get the first line in the line list, LGET N will get the next one, etc. It is especially useful when one want to search for lines of a given species. For example, one can list all methanol lines in the 3 mm

band, ordered by increasing upper level energy, with:

```
LAS> LLIST "CH3OH, vt=0,1" 80000 110000 /ORDER e
```

and then examine each of the line candidates with:

```
LAS> LGET F
LAS> LPLOT
LAS> LGET N
LAS> LPLOT
LAS> ...
```

Arguments:

INDEX — the line index. It can be either:

1) One of the following letters:
   - "f" get the first line of the index
   - "l" get the last line of the index
   - "p" get the previous line in the index
   - "n" get the next line of the index

2) An integer

3) Unspecified, in which case the last line is plotted again.

C.6 LID

[WEEDS\]LID [/SPECIES SPECIES] [/ENERGY ENERGY] [/AEINSTEIN AEINSTEIN] [[/DELTA DFREQ] | [/FULL]] [/IMAGE]

Search for lines in the database around a central frequency and mark them on the current spectra. By default, the central frequency is selected with the mouse left button. The search is done in an interval of width DFREQ around the selected frequency; alternatively, the /FULL option may be used to search for lines over the entire frequency range covered by the current spectra. The line database in which the search is performed may be selected with the DBSELECT command.

Options:

/SPECIES — refine the search to lines of a given species. The species name should match exactly that in the database. Default is "All".

/ENERGY — refine the search to lines with upper level energy lower than ENERGY, expressed in Kelvins.

/AEINSTEIN — refine the search to lines with einstein A coefficient greater than AEINSTEIN, expressed in Kelvins.

/DELTA — search for lines within DFREQ, expressed in MHz, from the

cursor position. Default is 1 MHz.

/FULL search for lines in the full frequency range covered by the current spectra.

/IMAGE search for lines in the image sideband. Note that this is the default if the spectra lower axis unit is I (image).

C.7 LLIST

[WEEDS\]LLIST

List the lines from the line index. The line index is built with the LFIND command.

C.8 LPLOT

[WEEDS\]LPLOT [DFREQ]

Plot the line from the current line index (built from the last FINDLINE command). A vertical line is drawn at the rest frequency of the line. An optional DFREQ parameter may be given to define the frequency interval to be plotted around the line (default 50 MHz)

Arguments:

DFREQ the frequency interval to be plotted, in MHz (default 50 MHz)

C.9 MODSHOW

[WEEDS\]MODSHOW

Plot the results of MODSHOW over the current spectra. The command ensures that the x-axis range and y-axis range are fixed so one can compare the predictions of the model with the observed spectra.

C.10 MODSOURCE

[WEEDS\]MODSOURCE MODFILE ANTSIZE [/BACKGROUND TBG] [/VERBOSE]

Model the emission of a source at the local thermodynamic equilibrium (LTE). The model paramaters (column density, kinetic temperature, source size, line width and velocity offset for each species) are read from MODFILE. The modeled antenna temperature is stored in the TA_MODEL memory. For convenience the current spectra is also stored in a memory, OBS, so one can easily overplot the result of the model on the observed spectra.

Arguments:

MODFILE the name of the file containing the parameters. The file should one line per species with six columns containing respectively:

1) The species name
2) The column density for this species in cm-2
3) The exitation temperature in K
4) The source size in arcsecs
5) The offset velocity from the source VLSR, in km/s
6) The line FWHM in km/s

Lines that start with a "#" are treated as comments.

ANTSIZE the antenna size in meters

Options:

/BACKGROUND set the background temperature to TBG, in K. Default is 2.73.

/VERBOSE prints partition function, lines frequencies, upper level energies and degeneracies, Einstein coefficients and opacity at the line center.

C.11 SCANFIND

[WEEDS\]SCANFIND RESTFREQ

This function loops over the scan index until it finds one that covers the given rest frequency. When the scan is found, it is loaded. Note that if several scans covers the same frequency, the command will find only the first one; other will be ignored.

Arguments:

RESTFREQ the rest frequency to search, in MHz.