# Weeds

Documentation Manual
June 9th, 2016

**Sébastien Maret**
Institut de Planétologie et d'Astrophysique de Grenoble

**Pierre Hily-Blant**
Institut de Planétologie et 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

**Mathieu Lonjaret**
Institut de Radio-Astronomie Millimétrique

**Arnaud Belloche**
Max-Planck-Institut für Radioastronomie

This manual documents Weeds, an extension of the CLASS radio astronomy software to reduce and analyze spectral surveys or spectral line observation with large bandwidths. Please report any errors in this manual to `gildas@iram.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 lines in crowded spectra, 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 accesses 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 makes 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 spectrum at millimeter and sub-millimeter wavelengths. Often, in order to analyze a spectral survey, one needs 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 analyze a spectral survey. It is written as a tutorial; each of step of the data analysis is explained through examples.

# 2 Getting started

Weeds has been distributed with GILDAS since March 2010. 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
GILDAS Version: dev (06jun16 15:32) (x86_64-darwin-gfortran-openmp) executable tree

 * 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 (S.Bardeau, J.Pety, P.Hily-Blant, S.Guilloteau)

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

(...)
LAS>
```

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

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

```
LAS90> help weeds\

    Weeds is a CLASS extension to analyze 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:
        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 commands provided by Weeds has 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 `use in`, followed by the name of the database[2]. For example:

```
LAS90> use in cdms
I-USE,  cdms (online) selected
```

or:

```
LAS90> use in jpl
I-USE,  jpl (online) selected
```

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

```
LAS90> use in cdms
I-USE,  cdms (online) selected
LAS90> use out mycdms.db
LAS90> select /freq 80e3 130e3
I-SELECT,  128957 lines found in the frequency range 80000.0 to 130000.0 MHz
LAS90> insert
```

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

Once you have made a local copy of a database, you may select it with the `use in` command, followed by the name of the local database file (`.db`):

```
LAS90> use in mycdms.db
I-USE,  mycdms.db (offline) selected
```

The `use in` and `use out` commands can be used to build a local database containing lines from different sources (e.g. the CDMS and the JPL), and even lines from a private catalog. See the online documentation for more information.

[2]The `use` command, as well as the `select` and `insert` commands that we use later on, are part of the `linedb\` language. Please type `help linedb\` for the full documentation of this language.

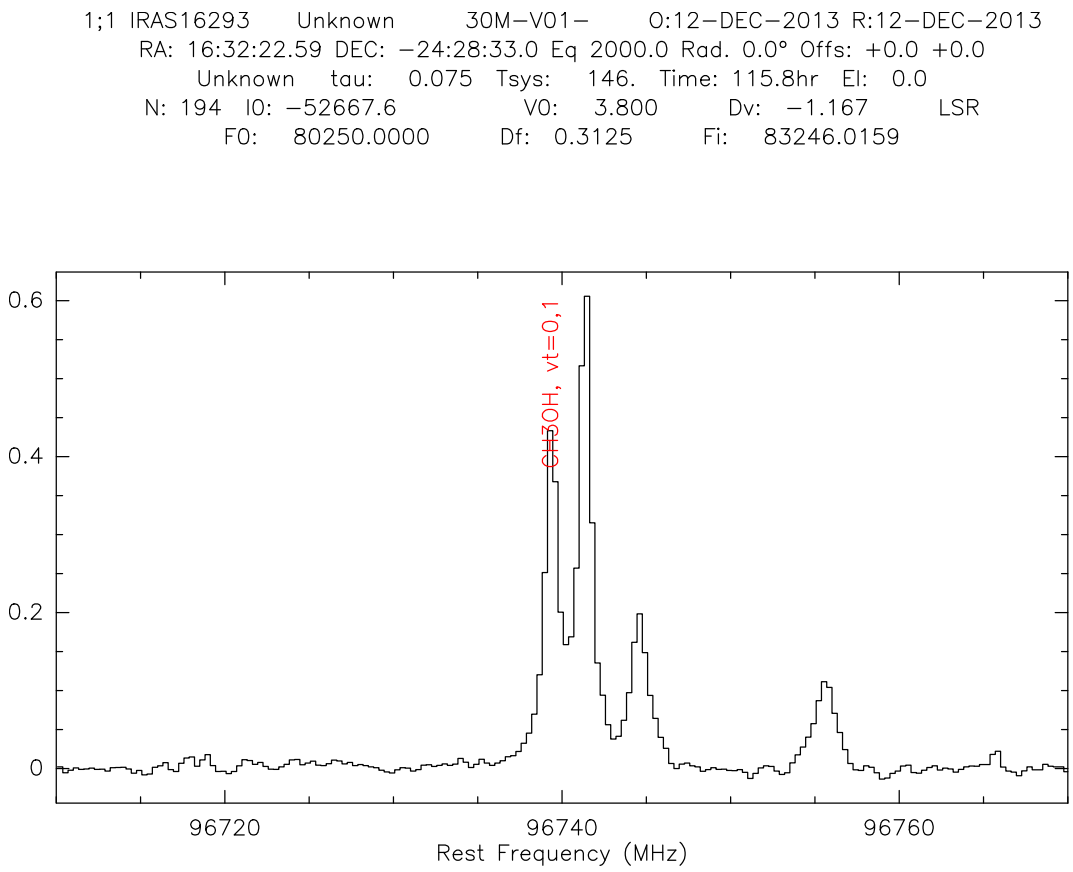



Figure 1: Example of a spectrum obtained with the IRAM-30m telescope towards the IRAS 16293 protostar. One of the methanol transition has been labeled with lid.

# 4 Identifying lines in a spectrum

Fig. 1 shows a part of a spectral survey[3]. The spectrum on the figure shows several well detected lines that we want to identify. This can be done with `lid` which allows us to select a line on the spectrum 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 `use in` command, as explained above) for lines within 10 channels of the selected region(s) of the spectrum[4]. For each line found, the command marks the rest frequency of the line on the spectrum, and a table displaying the line frequency, upper level energy, Einstein coefficient and quantum numbers and is printed:

```
LAS90> lid
I-SELECT,  1 lines found in the frequency range 96737.8185473 to 96740.9437312 MHz
  # Species               Freq[MHz] Err[MHz] Eup[K]  Gup  Aij[s-1]    Upper level -- Lower level    Origin
  1 CH3OH, vt=0,1         96739.362   0.005    12.5    5  2.56e-06     2 -1  2  0 -- 1 -1  1  0     cdms
```

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
I-SELECT,  4 lines found in the frequency range 96739.7756444 to 96742.9008283 MHz
  # Species               Freq[MHz] Err[MHz] Eup[K]  Gup  Aij[s-1]    Upper level -- Lower level    Origin
```

[3]The spectrum used in this tutorial is included in the Gildas distribution (courtesy of Emmanuel Caux), and can be copied in the current directory with the following CLASS command: `sic copy gag_demo:demo-weeds-1.30m demo-weeds-1.30m`.

[4]This frequency interval can be changed with the `/delta` option.

```
  1 n-C3H7CN                96741.208   0.009   158.0   50  7.60e-06       12 310 1 -- 11 2 9 1       cdms
  2 gGG'g-1,3-Propanediol   96741.245   0.030    40.1   31  1.58e-05       15 213 1 -- 14 212 0       cdms
  3 C4H, v7=1               96741.314   0.092   214.2   44  3.74e-06         10 111 -- 9-110          cdms
  4 CH3OH, vt=0,1           96741.375   0.005     7.0    5  3.41e-06     2  0  2 +0 -- 1  0  1 +0     cdms
```

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 allows to restrict searches to a given species.

It should be noted that `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 works 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 the spectrum (which covers frequencies between 90 and 100 GHz). The methanol line index is built with:

```
LAS90> lfind "CH3OH, vt=0,1"
I-SELECT,  36 lines found in the frequency range 89999.6813201 to 100000.269737 MHz
```

and then listed with:

```
LAS90> llist
  # Species         Freq[MHz] Err[MHz] Eup[K]  Gup  Aij[s-1]    Upper level -- Lower level    Origin
  1 CH3OH, vt=0,1   90295.864   1.754  1234.0   47  1.12e-08    23 +8 15  1 -- 23 +6 17  1    cdms
  2 CH3OH, vt=0,1   90812.387   0.232   808.3   41  2.81e-06    20 -3 17  1 -- 19 -2 17  1    cdms
  3 CH3OH, vt=0,1   91254.686   0.010   514.3   41  5.31e-08    20 -2 19  0 -- 20 +2 18  0    cdms
  4 CH3OH, vt=0,1   92409.579   0.019    44.3    9  5.10e-10     4 +1  3  0 -- 3 -2  2  0     cdms
(...)
 36 CH3OH, vt=0,1   99776.775   0.649   902.3   41  4.98e-06    20 +3 18  1 -- 21 +4 18  1    cdms
```

The command prints all methanol lines between 90 and 100 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 appears 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 is 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 `/sortby e` option:

```
LAS90> lfind "CH3OH, vt=0,1" /sortby e
I-SELECT,  36 lines found in the frequency range 89999.6813201 to 100000.269737 MHz
LAS90> llist
  # Species         Freq[MHz] Err[MHz] Eup[K]  Gup  Aij[s-1]    Upper level -- Lower level    Origin
  1 CH3OH, vt=0,1   96741.375   0.005     7.0    5  3.41e-06     2  0  2 +0 -- 1  0  1 +0     cdms
  2 CH3OH, vt=0,1   96739.362   0.005    12.5    5  2.56e-06     2 -1  2  0 -- 1 -1  1  0     cdms
  3 CH3OH, vt=0,1   96744.550   0.005    20.1    5  3.41e-06     2 +0  2  0 -- 1 +0  1  0     cdms
  4 CH3OH, vt=0,1   95914.309   0.005    21.4    5  2.49e-06     2  1  2 +0 -- 1  1  1 +0     cdms
(...)
 36 CH3OH, vt=0,1   95523.388   1.982  1289.5   49  1.74e-08    24 +8 16  1 -- 24 +6 18  1    cdms
```

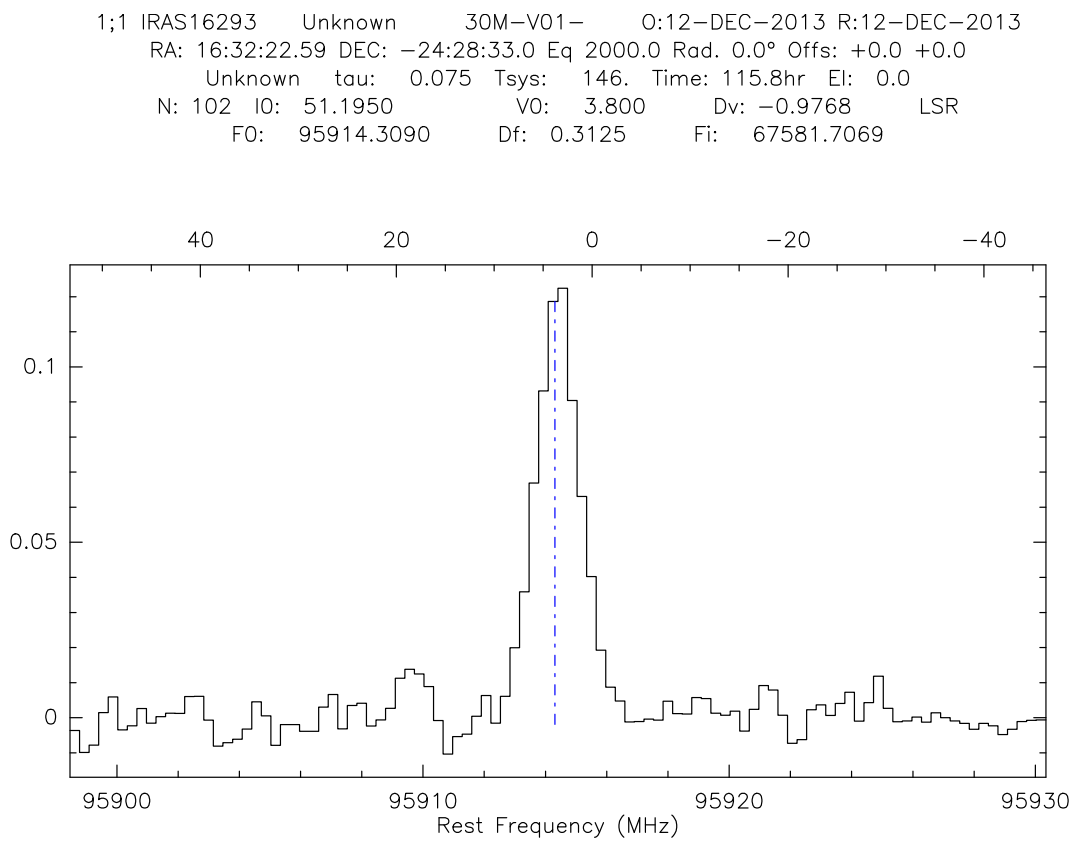



Figure 2: Line displayed with lplot command. The blue vertical line shows the rest frequency of the line. The upper x-axis shows the velocity offset from the line rest frequency.

Since we have already identified the first three lines in the index, let's have a look at another one:

```
LAS90> lget 4
I-LGET,  Found line frequency in the current scan
  # Species         Freq[MHz] Err[MHz] Eup[K]  Gup  Aij[s-1]    Upper level -- Lower level    Origin
  1 CH3OH, vt=0,1   95914.309   0.005    21.4    5  2.49e-06     2  1  2 +0 -- 1  1  1 +0     cdms
```

The `lget` command modifies the scan frequency so that the velocity axis is centered at the line rest frequency. We can then use:

```
LAS90> lplot
```

which gives the spectrum shown on Fig. 2.

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 line 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 model of the candidate line. For example, you may want to check if the relative intensities of each candidate line 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. The source is assumed to have, for each species, a given column density, excitation temperature, line FWHM, 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. More details on the formulas used can be found in § **??**.

Let us go back to the methanol lines that we have identified in our spectrum (see Fig. 2). The emission can be modeled using the `modsource` command, which takes two arguments: the name of the 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 CDMS does not provide the partition function for this species:

```
LAS90> use in jpl
I-USE,  jpl (online) selected
```

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

```
! species        Ntot        Tex  source_size v_off        width
!                 (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:

```
LAS> modsource iras16293.mdl 30
I-SELECT,  4 lines found in the frequency range 96710.0 to 96770.0 MHz
I-MODSOURCE,   4 CH3OH lines found in the frequency range
I-MODSOURCE,  log10 of the partition function at 10.0 K from jpl is 1.3419
  # Species        Freq[MHz] Err[MHz] Eup[K]  Gup  Aij[s-1]    Upper level -- Lower level    Origin Tau
  1 CH3OH          96739.358   0.002    12.5    5  2.56e-06        2-1   0 -- 1-1   0           jpl 7.25e-01
  2 CH3OH          96741.371   0.002     7.0    5  3.41e-06        2 0 + 0 -- 1 0 + 0          jpl 1.69e+00
  3 CH3OH          96744.545   0.002    20.1    5  3.41e-06        2 0   0 -- 1 0   0          jpl 4.56e-01
  4 CH3OH          96755.501   0.002    28.0    5  2.62e-06        2 1   0 -- 1 1   0          jpl 1.59e-01
I-MODEL,   Blanking value:   -1000.00000
I-RESAMPLE,  Frequency resolution: .31252 MHz (observatory), .31252 MHz (rest frame)
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 are found in the frequency range. Note the `/verbose` option, that prints the frequency, 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 the observed and modeled spectrum into buffers. These buffers can be listed with:

```
LAS90> memorize
I-MEMORIZE,  Current memories:
OBS           TB_MODEL
```

The `TB_MODEL` buffer contains the modeled brightness temperature. The buffer can be retrieved with the `retrieve` command. The `OBS` buffer contains the observed spectrum, 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.

[5]Alternatively, one may retrieve the predicted spectrum with `retrieve TB_MODEL`, and plot it over the observed spectrum with `spectrum`. However, it is important to fix the 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 ensures that both the observed and predicted spectra are on the same scale.

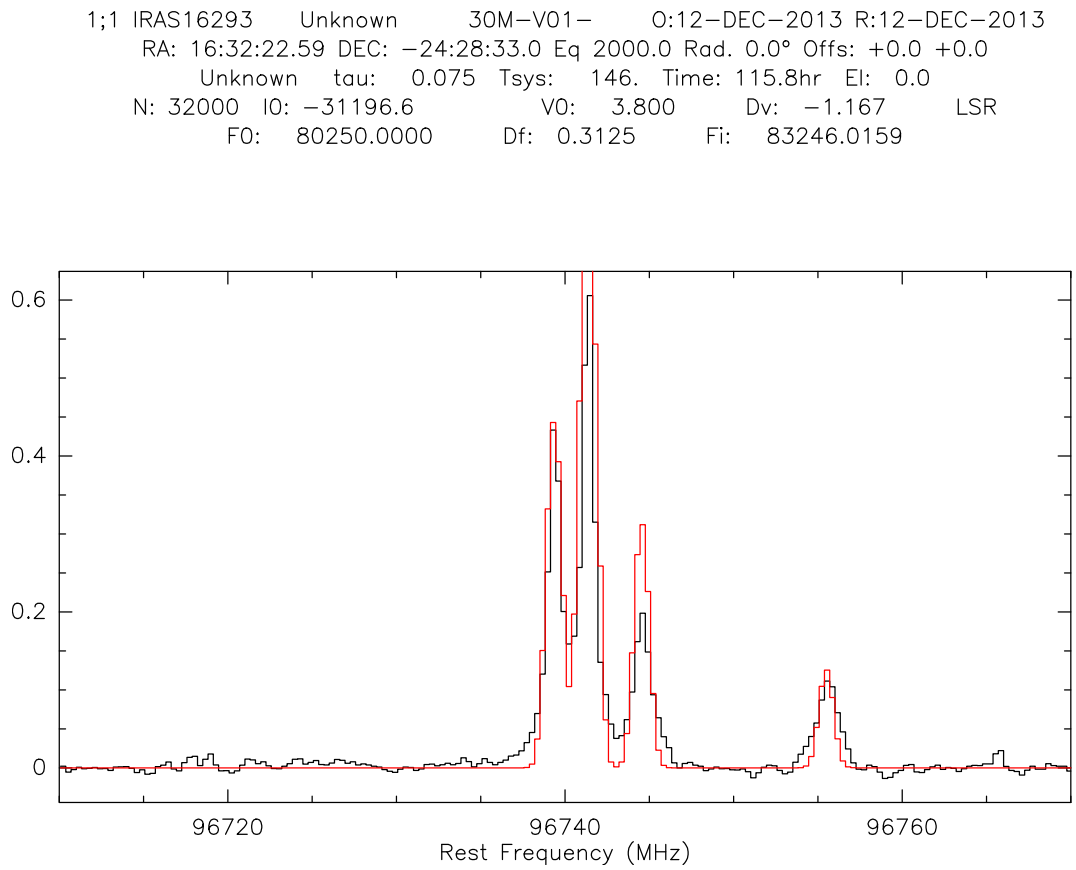



Figure 3: Observed (black histogram) and predicted spectra (in red) displayed with the modshow command.

Line indexes can be used together with `modshow` to examine other lines between 90 and 100 GHz, and to check whether our model can also reproduce them. For example:

```
LAS90> lfind "CH3OH" 90e3 100e3 /sortby e
I-SELECT,  142 lines found in the frequency range 90000.0 to 100000.0 MHz
LAS90> lget 4
LAS90> lplot
LAS90> modsource iras16293.mdl 30
I-SELECT,  1 lines found in the frequency range 95898.6840807 to 95929.9359193 MHz
I-MODSOURCE: 1 CH3OH lines found in the frequency range
I-MODEL,   Blanking value:   -1000.00000
I-RESAMPLE,  Frequency resolution: .31252 MHz (observatory), .31252 MHz (rest frame)
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. We must also rebuild the line index after each `modsource`, because the commands builds its own line index, containing only lines in the frequency range covered by the current window. Scripts can be easily created to loop over line index and examine each of the observed and predicted lines.

# 7 More information on modsource

`modsource` computes the spectrum emitted by a uniform source. The spectrum consists of continuum emission and spectral line emission. The line emission is computed under the assumption of local thermodynamic equilibrium (LTE). `modsource` takes into account the optical depth of the lines, the finite angular resolution of the telescope, and the presence of the cosmic microwave background (CMB) to compute the radiative transfer. Depending on the continuum and line excitation temperatures, the lines can appear in emission or in absorption in the synthetic spectrum. Several molecules and/or components can be modeled at the same time. The spectra of these components are computed separately, and then added linearly. It is a correct treatment for components that do not overlap spectrally or spatially within the beam, or for components with optically thin emission. However, such a treatment is not correct in the case of spatially-overlapping components with optically thick emission. In such cases, the intensities of the lines will be overestimated.

The components belong by default to the "main group" of components. The components marked with the option `/absorption` belong to the "foreground" group of components. For the "main group" components, the background used to compute the radiative transfer is the sum of the CMB and the source continuum emission. For the "foreground" components, the spectrum is computed recursively and the background initially consists of the spectrum of the "main group" (that already contains the CMB and the source continuum).

The source continuum emission is assumed to be optically thin (i.e. transparent to the CMB) and spatially uniform, to fill the beam of the single-dish telescope or the synthesized beam of the interferometer, and to come from a location behind the medium that contains the molecules modeled with modsource (i.e. lines and continuum are not mixed). The assumption of a uniform continuum emission filling the beam will produce inacurrate results in the case of a centrally peaked continuum emission and a molecule component with a beam filling factor smaller than 1. In such a case, the molecule should "see" a stronger continuum than what modsource assumes. Given that different components may have different beam filling factors, properly taking into account the small-scale structure of the continuum emission would not be trivial and is beyond the scope of `modsource`.

Let's call $T_{\mathrm{cont},\nu}$ the effective radiation temperature of the source continuum emission. It corresponds to the level of the "baseline" in the spectrum if this "baseline" represents the true continuum level of the source (case of, e.g., interferometric or some single-dish wobbler spectra), or the user has to assume its value if the baseline cannot be trusted and has been removed (case of, e.g., single-dish position-switching spectra). $T_{\mathrm{cont},\nu}$ can be a frequency-dependent function.

For a component with a beam filling factor of 1, the brightness temperature of the spectrum is computed as:

$$T_{\rm B}(\nu) = J_\nu(T_{\rm ex})(1 - e^{-\tau_\nu}) + T_{{\rm cont},\nu} e^{-\tau_\nu} + J_\nu(T_{\rm cmb}) e^{-\tau_\nu} - J_\nu(T_{\rm cmb}) \ , \quad (1)$$

where $\tau_\nu$ is the opacity of the transition, $T_{\rm ex}$ the excitation temperature of the transition, $T_{\rm cmb}$ the temperature of the CMB, and $J_\nu(T) = \frac{h\nu/k}{e^{h\nu/kT}-1}$, with $h$ the Planck constant, and $k$ the Boltzmann constant. The last term in Eq. 1 represents the subtraction of the spectrum of the off-source position in the case of single-dish observations or the spatial filtering of the uniform CMB radiation in the case of interferometric observations. This equation can be rewritten as:

$$T_{\rm B}(\nu) = T_{{\rm cont},\nu} + (J_\nu(T_{\rm ex}) - J_\nu(T_{\rm cmb}) - T_{{\rm cont},\nu})\,(1 - e^{-\tau_\nu}) \quad (2)$$

For a component with an arbitrary beam filling factor, $\eta$, modsource computes the spectrum as:

$$T_{\rm B}(\nu) = T_{{\rm cont},\nu} + \eta\,(J_\nu(T_{\rm ex}) - J_\nu(T_{\rm cmb}) - T_{{\rm cont},\nu})\,(1 - e^{-\tau_\nu}) \quad (3)$$

For $N_1$ components in the "main group" (mg), the output spectrum is:

$$T_{\rm B}^{\rm mg}(\nu) = T_{{\rm cont},\nu} + \sum_{i=1}^{N_1} \eta_i\,(J_\nu(T_{{\rm ex},i}) - J_\nu(T_{\rm cmb}) - T_{{\rm cont},\nu})\,(1 - e^{-\tau_{\nu,i}}) \quad (4)$$

For $N_2$ additional components in the "foreground group", the output spectrum is computed with the following recursive equation:

$$T_{\rm B}^{j}(\nu) = T_{\rm B}^{j-1}(\nu) e^{-\tau_{\nu,j}} + \eta_j\,(J_\nu(T_{{\rm ex},j}) - J_\nu(T_{\rm cmb}))\,(1 - e^{-\tau_{\nu,j}}) \quad (5)$$

with $j$ varying from 1 to $N_2$ and $T_{\rm B}^0(\nu) = T_{\rm B}^{\rm mg}(\nu)$. Note that this equation is correct only if all foreground components have a beam filling factor, $\eta_j$, of 1.

By default, the final output spectrum is

$$T_{\rm B}(\nu) = T_{\rm B}^{N_2}(\nu) - T_{{\rm cont},\nu} \quad (6)$$

or, if there is no "foreground group" component:

$$T_{\rm B}(\nu) = T_{\rm B}^{\rm mg}(\nu) - T_{{\rm cont},\nu} \quad (7)$$

If the user selects the option `/keepcont`, then the output spectrum is $T_{\rm B}(\nu) = T_{\rm B}^{N_2}(\nu)$ in the first case and $T_{\rm B}(\nu) = T_{\rm B}^{\rm mg}(\nu)$ in the second case. More details about modsource can be found in Maret, Hily-Blant, Pety et al., *Astronomy & Astrophysics* 526, A47 (2011).

# 8 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.:

```
LAS90> help modsource
WEEDS\MODSOURCE = "python weeds/modsource.py"
        [WEEDS\]MODSOURCE  MODFILE  ANTSIZE  [/TCONT  TCONT]  [/OPACITY   I]
    [/KEEPCONT] [/VERBOSE]

    Model  the emission of a source assuming 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 brightness temperature is stored  in  the  TB_MODEL
    memory. For convenience the current spectrum is also stored in a memory,
    OBS, so one can easily overplot the result of the model on the  observed
    spectrum.  See the documentation manual for more details on the computa-
    tion.
```

```
Arguments:
  MODFILE     the name of the file containing the parameters.  The  file
              must  have  one line per species with at least six columns
              containing respectively:

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

              One additional column may be used to  select  a  different
              origin for each species:

              7)  The origin of this species (e.g. JPL or CDMS). If emp-
              ty, species with any origin are selected.

              The following options may be used at the end of the line:

              /ABSORPTION  If this option is set, the component  is  as-
              sumed to be a foreground layer.

              /PARTFUNC  Q   Set  the log10 of the partition function at
              the excitation temperature to Q. If not set, the partition
              function  values (log-log interpolated/extrapolated at the
              excitation temperature) from  the  selected  database  are
              used.

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

  ANTSIZE     the antenna size in meters

Options:
  /TCONT      set the continuum level to TCONT, in K. Default is 0.

  /OPACITY  I  store  the  opacity of the Ith component in the TAU_MODEL
              memory. Components are numbered in the same  order  as  in
              MODFILE, starting from 1.

  /KEEPCONT   do not remove the continuum from the final spectra.

  /VERBOSE    print  partition  function,  line frequencies, upper level
              energies and degeneracies, Einstein coefficients and opac-
              ities at the line centers.
```

For a more detailed description of the Weeds extension, we refer the interested reader to Maret, Hily-Blant, Pety et al., *Astronomy & Astrophysics* 526, A47 (2011). This paper describes the implementation of the extension, and gives more details about the spectroscopic databases that it uses, and explains how spectra are modeled.

Weeds developers can be contacted by email at `gildas@iram.fr`. Bug reports should also be sent to this address.

# A List of contributing authors

The Weeds project started on mid-2009 at Laboratoire d'Astrophysique de Grenoble (now known as Institut de Planétologie et d'Astrophysique de Grenoble). Soon after Institut de Radio Astronomie Millimétrique started to support its development.

The following authors have contributed to the project:

- Sébastien Maret (Institut de Planétologie et d'Astrophysique de Grenoble)
- Pierre Hily-Blant (Institut de Planétologie et 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)
- Mathieu Lonjaret (Institut de Radio Astronomie Millimétrique)
- Arnaud Belloche (Max-Planck-Institut für Radioastronomie)

Weeds relies on PyGILDAS, a Python binding for GILDAS developed by Sébastien Bardeau, Jérôme Pety, Emmanuel Reynier (Institut de Radio Astronomie Millimétrique) and Stéphane Guilloteau (Observatoire de Bordeaux).

# B Acknowledging Weeds developers and spectral line databases in your publications

If you use Weeds in your research work, please cite Maret, Hily-Blant, Pety et al., *Astronomy & Astrophysics* 526, A47 (2011) in your publications. Please also acknowledge the use of the CDMS and JPL databases by citing the following papers:

- H. M. Pickett, R. L. Poynter, E. A. Cohen, M. L. Delitsky, J. C. Pearson, and H. S. P. Müller, "Submillimeter, Millimeter, and Microwave Spectral Line Catalog", J. Quant. Spectrosc. & Rad. Transfer 60, 883-890 (1998).
- H. S. P. Müller, F. Schlöder, J. Stutzki, and G. Winnewisser, "The Cologne Database for Molecular Spectroscopy, CDMS: a useful tool for astronomers and spectroscopists" J. Mol. Struct. 742, 215–227 (2005)

# C List of major changes

## C.1 April 1st, 2016 release

The previous versions of modsource did not have a proper implementation of the background emission in the cases where both the cosmic microwave background (CMB) and another source of continuum emission are present. The new version deals with such a situation in an improved way. In addition, the parameter defining the source continuum emission now corresponds to the level of continuum emission that can be measured in the spectrum of the source. In the previous versions, the continuum was assumed to be a blackbody and the parameter was defined as the temperature of this blackbody, i.e. it was not directly related to the level of continuum emission seen in the spectrum. The new behaviour is particularly convenient for interferometric observations, space observations, or in some cases single-dish wobbler observations that usually deliver spectra with reliable levels

of continuum emission. Finally, the user now has the possibility to define a frequency-dependent source continuum emission by providing an array of continuum temperatures with the same size as the spectrum. By default, modsource still produces a baseline-removed synthetic spectrum. A new option has now been added to produce a synthetic spectrum that includes the source continuum emission.

## C.2 March 1st, 2014 release

Weeds used to rely on a Python library, `linedb`, for querying spectral line databases. This library has been developed as part of Weeds, but it has been realized that it could be useful in other GILDAS programs as well. Therefore it has been moved to the GILDAS kernel, after some important modifications to make it more generic and versatile. It is now a separate language (see `help linedb\` for more information). As a consequence, a couple of Weeds commands have been modified:

- The `dbselect` has been removed. The database selection is now made with the `use in` command, which is part of the `linedb\` language.
- The `dbcache` command has also been removed. Users can now build a local copy of a database with a combination of `use in`, `use out`, `select` and `insert`. See section 3 for an example of this.

In addition, the following changes have been implemented in this release:

- `modsource` now allows to model sources with one or several foreground absorbing layers. This is set with the `/absorption` option in the model file. See the `modsource` command help for more information.
- A bug in the interpolation of partition functions as a function of the temperature in `modsource` has been corrected. The interpolation is now done in log-log.
- `lid` now prints the uncertainty on the line frequencies

# D WEEDS Language Internal Help

## D.1 Language

Weeds is a CLASS extension to analyze 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:

| | |
|---|---|
| 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 assuming LTE |
| MODSHOW | Show the results of MODSOURCE |

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

## D.2 LFIND

[WEEDS\]LFIND [SPECIES [FMIN FMAX]] [/SORTBY ORDER] [/ENERGY ENERGY] [/AIJ AMIN] [/ORIGIN OriginName]

Find the lines within a frequency range and create a line index. The command looks for the lines between FMIN and FMAX (default is the frequency range covered by the current R spectrum) in the database selected with the USE IN command.

Arguments:

| | |
|---|---|
| SPECIES | the name of the species. Default is "All". |
| FMIN | the minimum frequency in MHz. Default is the current R spectrum minimum limit. |
| FMAX | the maximum frequency in MHz. Default is the current R spectrum maximum limit. |

Options:

| | |
|---|---|
| /SORTBY | sort the lines according to ORDER, which can be any combination of "frequency", "energy" and "aeinstein", and all the subsequent permutations. |
| /ENERGY | refine the search to lines with upper level energy lower than ENERGY, expressed in Kelvins. |
| /AIJ | refine the search to lines with einstein A coefficient greater than AMIN, expressed s-1. |
| /ORIGIN | refine the search to lines originally from the named database only. |

As an output, the structure LINES% will contain the content of the cur-

rent index. See HELP SELECT for details.

## D.3 LGET

```
[WEEDS\]LGET [Key|Num] [/INDEX]
```

Get a line from the line index. The commands checks if the line frequency is covered by the current observation. If yes, the observation is modified so that its rest frequency corresponds to the line frequency. 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 /SORTBY e
```

and then examine each of the line candidates with:

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

Options:

/INDEX search for the line in all the observations of the current observation index. If this option is set, LGET begins by searching for the first observation (from the last FIND command) that covers the line frequency. It is useful when working with line surveys that consists in several observations, each covering a different frequency band.

Argument: it can be either

1) One of the following key 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

## D.4 LID

```
[WEEDS\]LID [/SPECIES Name1 Name2 ...] [/ENERGY Value] [/AIJ Amin]
[[/DELTA Df] | [/FULL]] [/IMAGE] [/LOOP] [/COLOR Name] [/ANGLE Value]
```

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 USE IN command.

Search options:

/SPECIES refine the search to lines of one or more species. The species name should match exactly that in the database. Wildcards '*' are accepted. Default is "All".

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

/AIJ refine the search to lines with Einstein A coefficient greater than AMIN, expressed in s-1.

/DELTA search for lines within DFREQ, expressed in MHz, from the cursor position. Default is 10 times the channel width.

/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).

/LOOP by default, the cursor is invoked only once, but you can iterate indefinitely with the option /LOOP. Type 'E' to exit the loop.

Plot options:

/COLOR customize the color of the line and its label. See PEN /COLOUR ? or Menu > Demos > Greg > PENCIL color names for possible colors. By default, the lines are plotted in red for signal band and blue for image band.

/ANGLE customize the angle of the label (degrees). Default is 90.

## D.5 LLIST

[WEEDS\]LLIST

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

## D.6 LPLOT

[WEEDS\]LPLOT [Nchan]

Plot the current line selected from the last call to LGET. A vertical line is drawn at the rest frequency of the line.

An optional Nchan parameter may be given to define the frequency interval to be plotted around the line, expressed in number of channels (de-

fault 100).

## D.7 MODSHOW

[WEEDS\]MODSHOW [/PEN Ipen] [/SCALE AUTO|KEEP]

Plot the results of MODSOURCE over the current spectrum.

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 spectrum. The current Y scale is kept by default (/SCALE KEEP), but it can be automatically recomputed if the model requires a larger range (/SCALE AUTO).

The model is drawn with the pencil 1 by default (plain line, red color at start), but this can be changed with the /PEN option.

## D.8 MODSOURCE

[WEEDS\]MODSOURCE MODFILE ANTSIZE [/TCONT TCONT] [/OPACITY I] [/KEEPCONT] [/VERBOSE]

Model the emission of a source assuming 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 brightness temperature is stored in the TB_MODEL memory. For convenience the current spectrum is also stored in a memory, OBS, so one can easily overplot the result of the model on the observed spectrum. See the documentation manual for more details on the computation.

Arguments:

MODFILE the name of the file containing the parameters. The file must have one line per species with at least six columns containing respectively:

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

One additional column may be used to select a different origin for each species:

7) The origin of this species (e.g. JPL or CDMS). If empty, species with any origin are selected.

The following options may be used at the end of the line:

/ABSORPTION If this option is set, the component is assumed to be a foreground layer.

/PARTFUNC Q Set the log10 of the partition function at the excitation temperature to Q. If not set, the partition function values (log-log interpolated/extrapolated at the excitation temperature) from the selected database are used.

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

ANTSIZE the antenna size in meters

Options:

/TCONT set the continuum level to TCONT, in K. The value can be scalar (one value for the whole frequency range) or an array name (same size as RY, one value per channel). Default is 0.

/OPACITY I store the opacity of the Ith component in the TAU_MODEL memory. Components are numbered in the same order as in MODFILE, starting from 1.

/KEEPCONT do not remove the continuum from the final spectra.

/VERBOSE print partition function, line frequencies, upper level energies and degeneracies, Einstein coefficients and opacities at the line centers.

## D.9 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.