XCLASS:
automatic line fitting of ALMA data

- Introduction and Tutorial -

Álvaro Sánchez-Monge
I. Physikalisches Intitut - Universität zu Köln
German ARC node - ALMA

XCLASS:
automatic line fitting of ALMA data

- Introduction -

Álvaro Sánchez-Monge
I. Physikalisches Intitut - Universität zu Köln
German ARC node - ALMA

WHAT

WHY

HOW

WHERE

WHAT is XCLASS ?

WHY do we need XCLASS ?

HOW does XCLASS work ?

WHERE can you find XCLASS ?

WHAT is XCLASS ?

WHAT is XCLASS ?

eXtended CASA Line Analysis Software Suite

XCLASS is a toolbox for the CASA* package
developed at the University of Cologne (within the German ARC node)
and aimed at identifying and fitting astronomical spectral line data

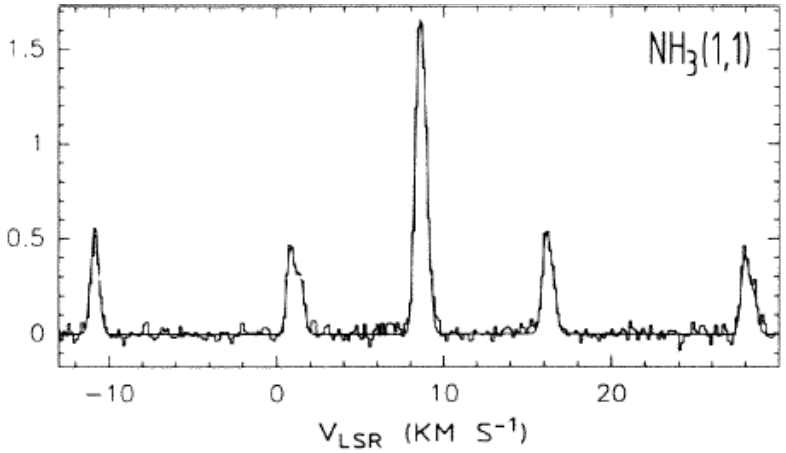
the toolbox contains an interface for the model optimizer MAGIX**
which helps to find the best description of the data using a certain model

* CASA = Common Astronomy Software Applications
** MAGIX = Modeling and Analysis Generic Interface for eXternal numerical codes

WHY do we need XCLASS ?

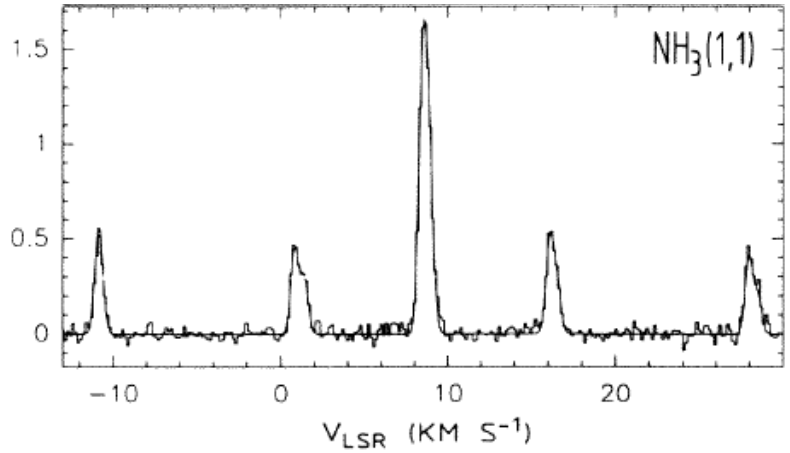
WHY do we need XCLASS ?

spectral lines from molecules represent a unique tool to characterize the nature of interstellar objects



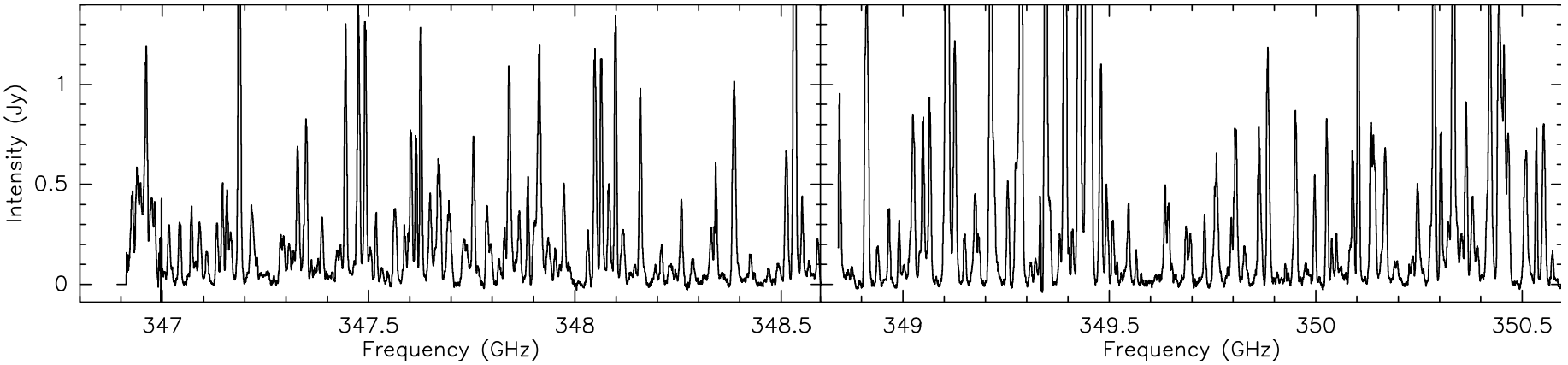
WHY do we need XCLASS ?

spectral lines from molecules represent a unique tool to characterize the nature of interstellar objects



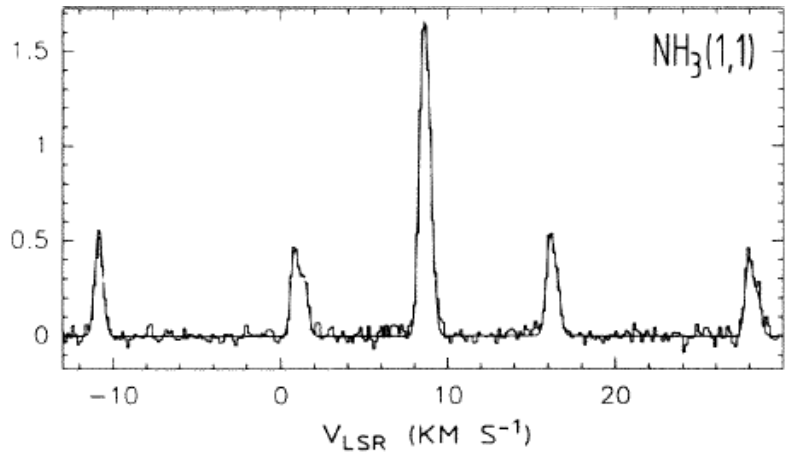
… with ALMA

high sensitivity: multitude of lines toward all (?) the studied sources



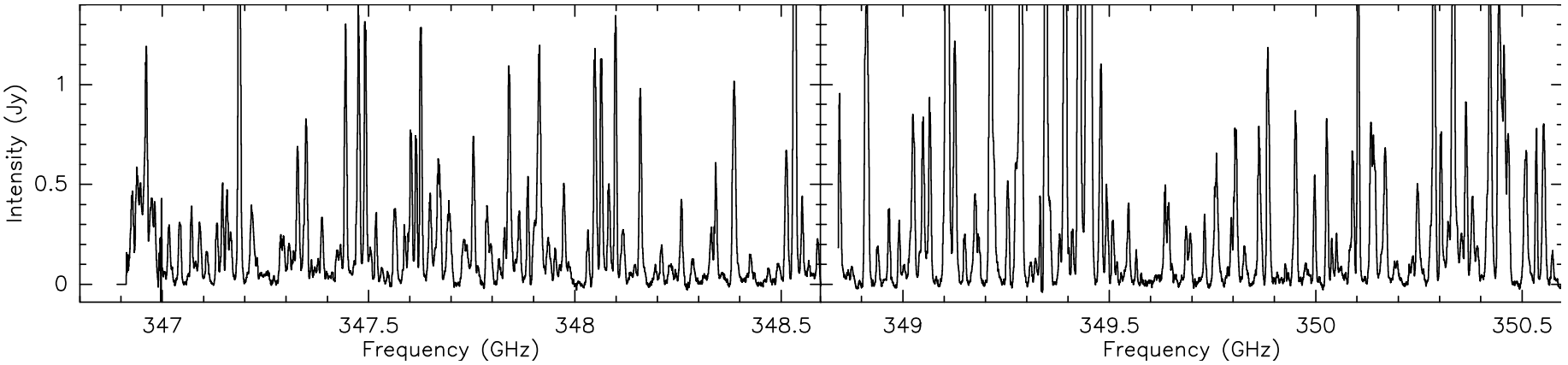
WHY do we need XCLASS ?

spectral lines from molecules represent a unique tool to characterize the nature of interstellar objects



… with ALMA

high sensitivity: multitude of lines toward all (?) the studied sources



HOW can we analyze such an amount of data? … using **XCLASS**

HOW does XCLASS work ?

HOW does XCLASS work ?

XCLASS generates **synthetic spectra** and compares it to real data

HOW does XCLASS work ?

XCLASS generates **synthetic spectra** and compares it to real data

considering LTE (local thermodynamic equilibrium)
taking into account opacity effects, and line blending

the modeled spectrum is a solution of the radiative transfer equation
for an isothermal object in one dimension

$$T_{\mathrm{mb}}^{\mathrm{core}}(\nu) = \sum_m \sum_c \left[\eta\left(\theta^{m,c}\right) \left[S^{m,c}(\nu) \left(1 - e^{-\tau_{\mathrm{total}}^{m,c}(\nu)}\right) + I_{\mathrm{bg}}^{\mathrm{core}}(\nu) \left(e^{-\tau_{\mathrm{total}}^{m,c}(\nu)} - 1\right) \right] \right] + \left(I_{\mathrm{bg}}^{\mathrm{core}}(\nu) - J_{\mathrm{CMB}}\right),$$

beam filling factor

source function

optical depth

sum goes over the indices *m* for molecule, *c* for component

HOW does XCLASS work ?

molecular parameters:
e.g. frequency of the transition, partition function, degeneracy of state taken from a database included in XCLASS

the database includes entries from the CDMS*, JPL and HITRAN catalogues using the VAMDC** portal

* CDMS = the Cologne Database for Molecular Spectroscopy
** VAMDC = Virtual Atomic and Molecular Data Center

HOW does XCLASS work ?

molecular parameters:
e.g. frequency of the transition, partition function, degeneracy of state taken from a database included in XCLASS

the database includes entries from the CDMS*, JPL and HITRAN catalogues using the VAMDC** portal

adjustable parameters:
only 5 input parameters:

gas temperature, molecular abundance (column density), velocity, linewidth, and size of the source

* CDMS = the Cologne Database for Molecular Spectroscopy
** VAMDC = Virtual Atomic and Molecular Data Center

HOW does XCLASS work ?

main functions of XCLASS to analyze spectroscopic data:

`myXCLASS`

function to model a spectrum by solving the radiative transfer equation

`myXCLASSFit`

the function uses MAGIX to fit experimental data using myXCLASS
(search for the set of parameters that best fits the data)

`myXCLASSMapFit`

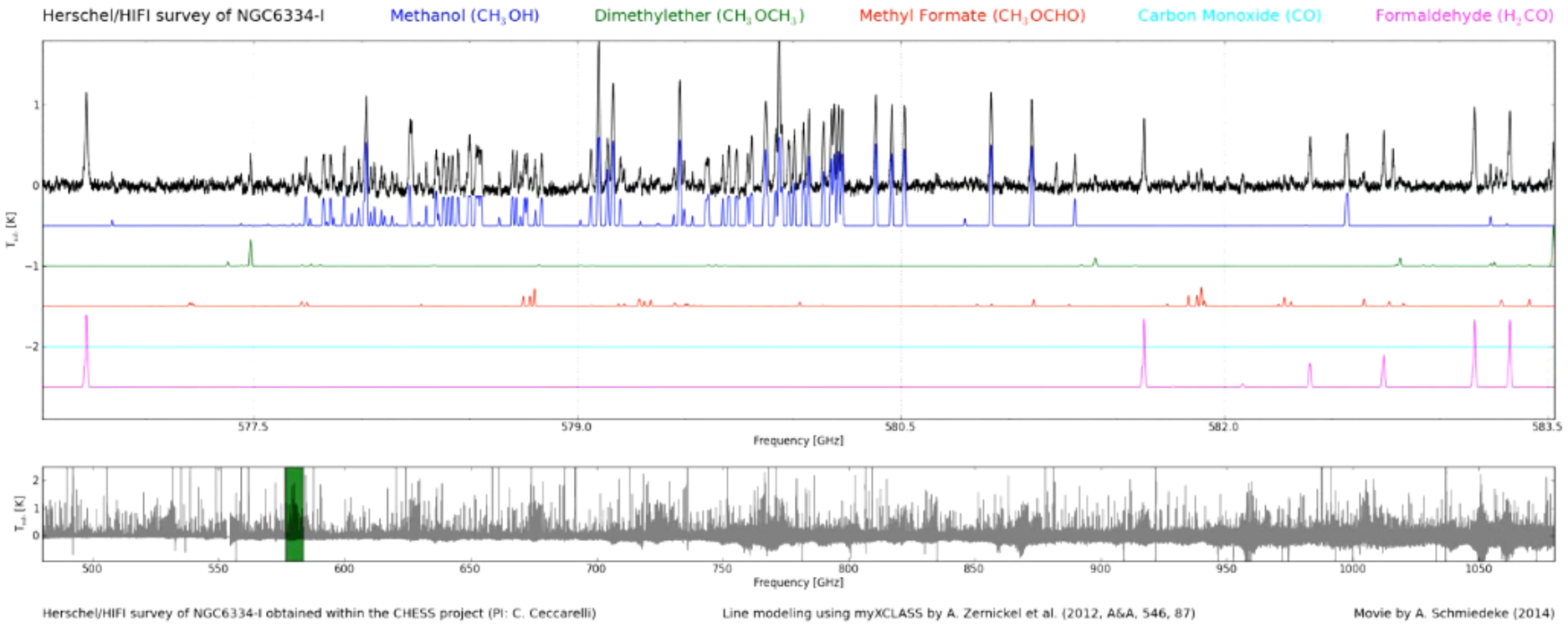
uses myXCLASS + MAGIX to fit a complete data cube
produces temperature maps, column density maps,...

`LineIdentification`

automatic routine to identify all the molecules in an observed spectrum

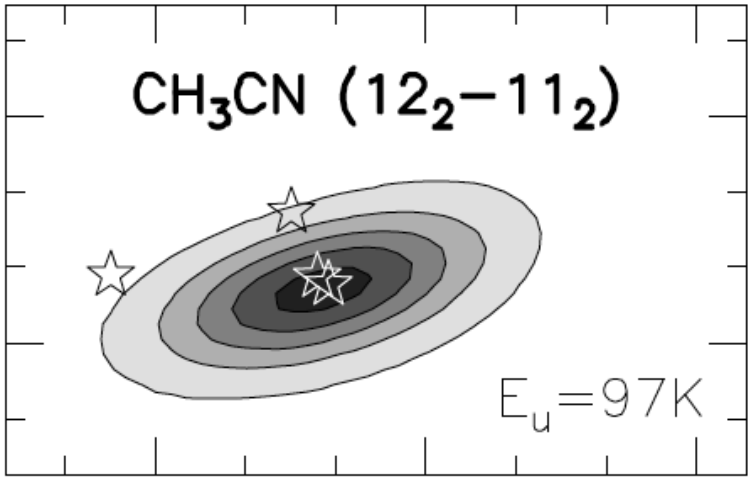
HOW does XCLASS work ?

example of **myXCLASSFit**:
- observational data in black
- fitted molecules in different colors



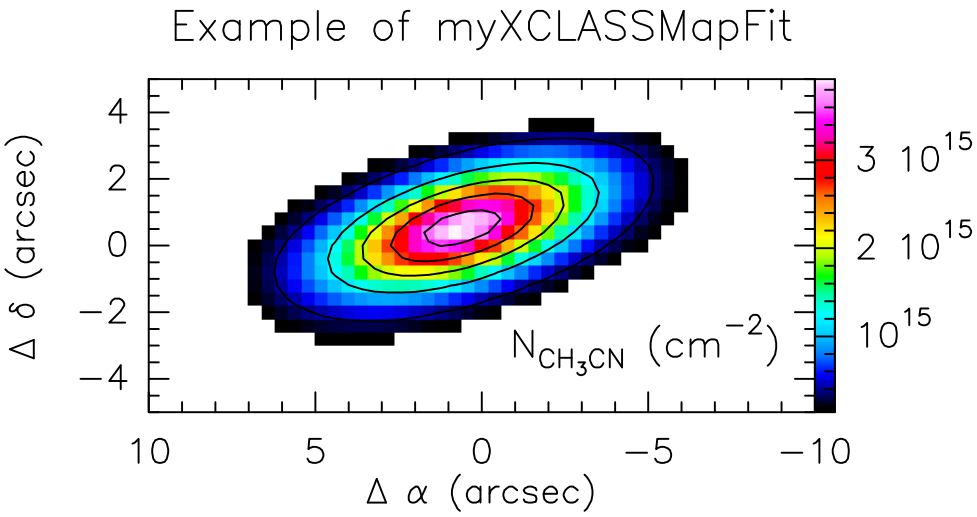
HOW does XCLASS work ?

example of **myXCLASSMapFit**:
- from CH_3CN line fitting



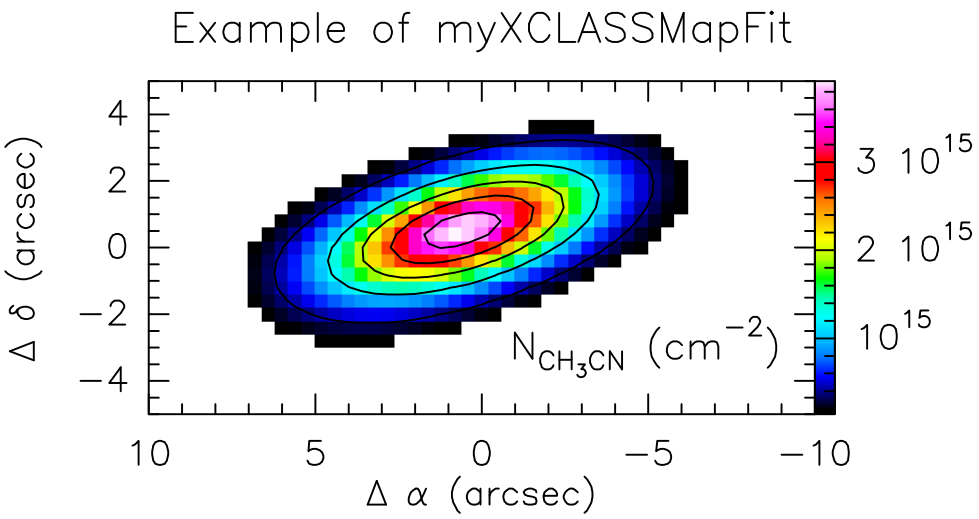
HOW does XCLASS work ?

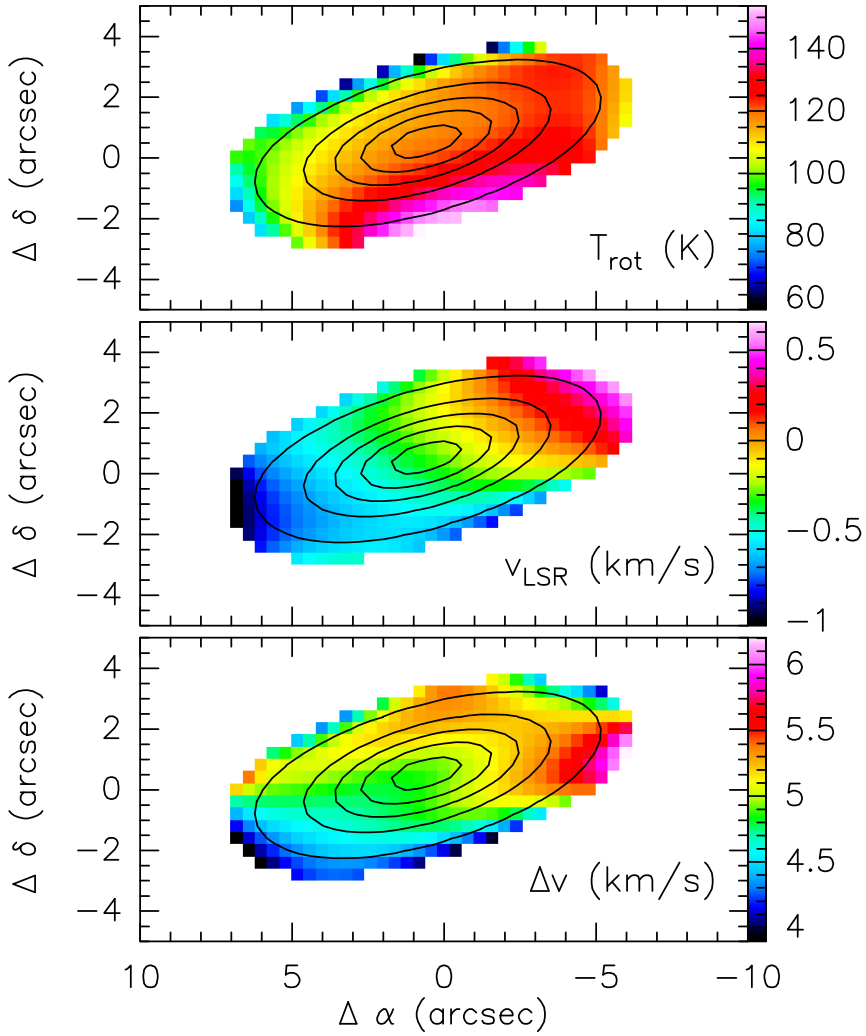
example of **myXCLASSMapFit**:
- from CH_3CN line fitting



HOW does XCLASS work ?

example of **myXCLASSMapFit**:
- from CH_3CN line fitting

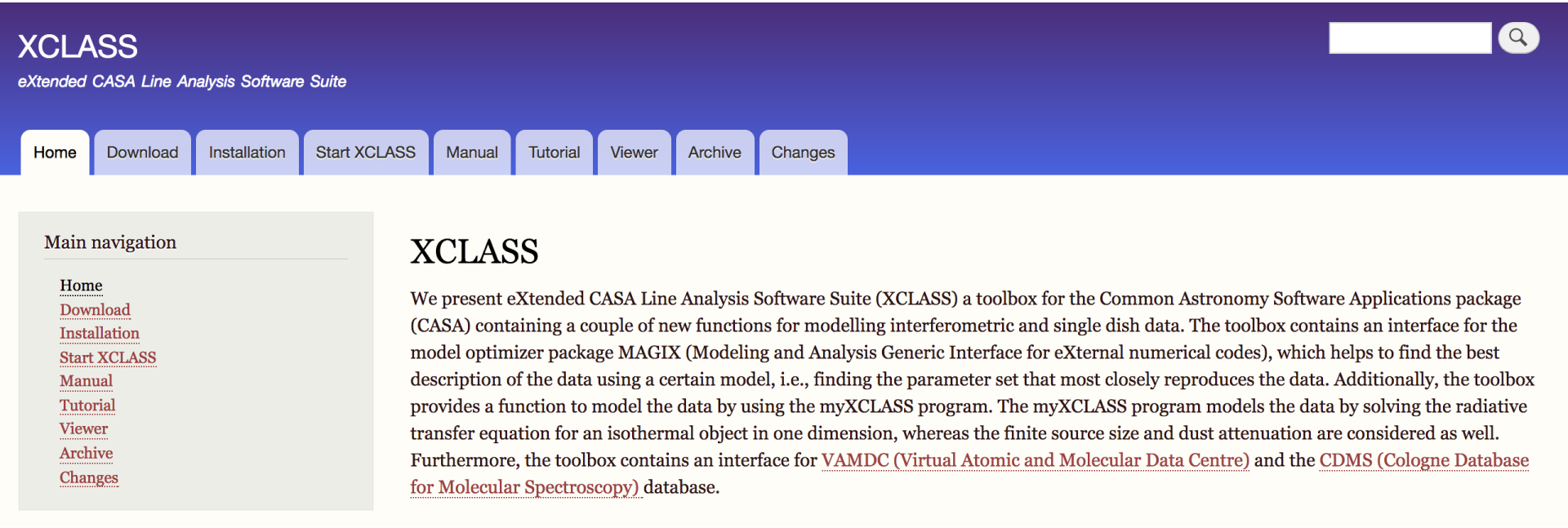




WHERE can you find XCLASS ?

WHERE can you find XCLASS ?

XCLASS for CASA can be downloaded from:
https://xclass.astro.uni-koeln.de/

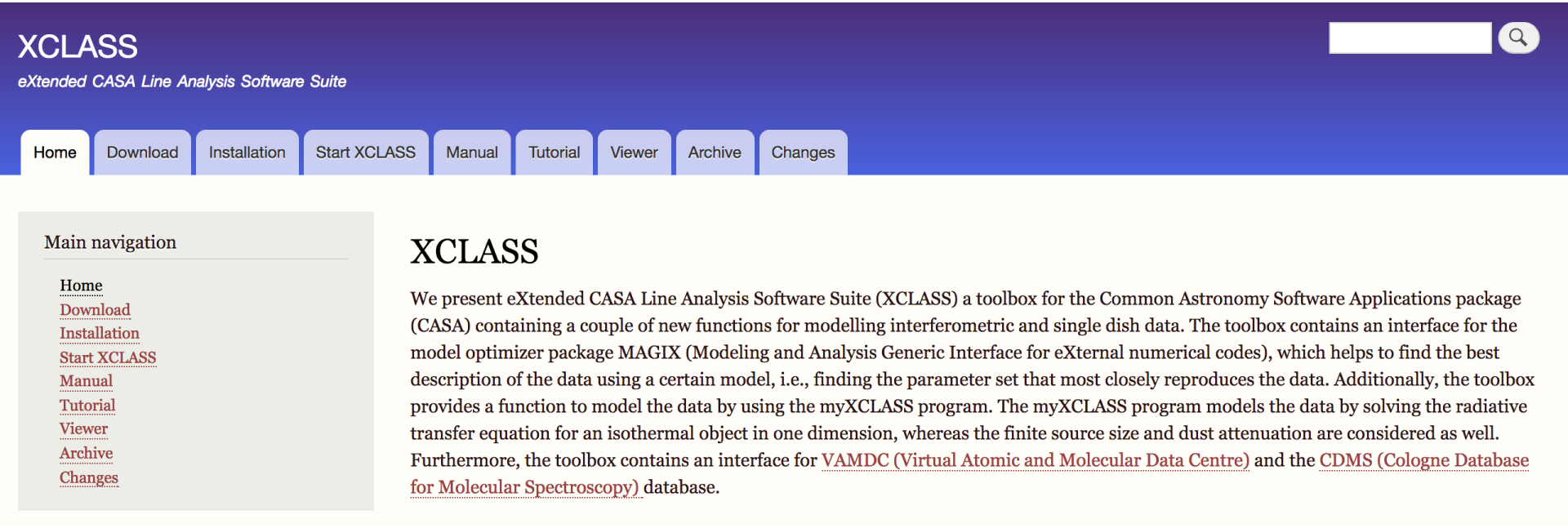


download the available zip file and execute

```
python install-in-casa.py --smp
```

WHERE can you find XCLASS ?

XCLASS for CASA can be downloaded from:
https://xclass.astro.uni-koeln.de/



download the available zip file and execute

```
python install-in-casa.py --smp
```

THANKS, questions?

XCLASS:
automatic line fitting of ALMA data

- Tutorial -

Álvaro Sánchez-Monge
I. Physikalisches Intitut - Universität zu Köln
German ARC node - ALMA

Download

XCLASS for CASA can be downloaded from:
https://xclass.astro.uni-koeln.de/Download

CASA can be downloaded from:
http://casa.nrao.edu/

XCLASS is available for Linux and MAC OS 10.11

tips:

- update your gfortran/gcc compilers
- XCLASS works smoothly in most Mac OS versions
- if you have problems, please contact: moeller@ph1.uni-koeln.de

Installation

First step:

install CASA following standard instructions
write in your .bashrc file the path to your CASA directory
execute CASA at least once (casa), before installing XCLASS

Second step:

unzip the downloaded XCLASS zip file
execute the installation script:

```
python install-in-casa.py --smp
```

tips:

- if you want the mpi version use: `python install-in-casa.py --mpi`

Starting XCLASS (within CASA)

within CASA:

```
$ casa
CASA$ tasklist

...
User defined tasks
------------------
myXCLASS
LoadASCIIFile
myXCLASSPlot
MAGIX
myXCLASSFit
myXCLASSMapFit
GetTransitions
ListDatabase
DatabaseQuery
UpdateDatabase
LineIdentification
myXCLASSMapRedoFit
```

Starting XCLASS (outside CASA)

outside CASA:

extend sys.path variable to include XCLASS directory

```
NewModulesPath = “path-of-XCLASS-Interface/build_tasks/”
sys.path.append(NewModulesPath)
```

then you can use XCLASS tasks as python packages

```
import task_ListDatabase
import task_UpdateDatabase
...
```

tips:

- you need the packages: numpy, scipy, pyfits, matplotlib and sqlite3
- check example files my_*task*_OUTcasa.py

UpdateDatabase()

description:

to download the latest version of the sqlite database from the CDMS/VAMDC server

UpdateDatabase()

description:

to download the latest version of the sqlite database from the CDMS/VAMDC server

usage/inputs:

- to download a new version of the database

```
CASA$ DBUpdateNew = "new"
CASA$ UpdateDatabase()
```

- to update the database

```
CASA$ DBUpdateNew = "update"
CASA$ UpdateDatabase()
```

tips:

- downloading a new database is fast (the update can take a long time)
- check example file my_update_INcasa.py

DatabaseQuery()

description:

to use a “query string” and search in the database

DatabaseQuery()

description:

to use a “query string” and search in the database

usage/inputs:

- to list all the species of the database

```
CASA$ QueryString="select PF_Name from Partitionfunctions"
CASA$ DatabaseQuery()
```

- to list species within a given frequency range

```
CASA$ QueryString = "select T_Name from transitions where
                     T_Frequency >= 10000 and
                     T_Frequency <= 12000"
CASA$ DatabaseQuery()
```

tips:

- check example file my_query_INcasa.py

ListDatabase()

description:

to read/list entries from the database

ListDatabase()

description:

to read/list entries from the database

usage/inputs:

```
CASA$ OutputDevice = ' '
CASA$ FreqMin = 210000.0
CASA$ FreqMax = 230000.0
CASA$ ElowMin = 0.0
CASA$ ElowMax = 300.0
CASA$ SelectMolecule = ["CO;v=0;", "C-13-O;v=0;"]

CASA$ ListDatabase()
```

tips:

- check example file my_list_INcasa.py

LoadASCIIFile()

description:

to read/import data from an ASCII file

LoadASCIIFile()

description:

to read/import data from an ASCII file

usage/inputs:

```
CASA$ FileName = "Observational-File.dat"
CASA$ NumHeaderLines = 0
CASA$ RestFreq = 0.0
CASA$ vLSR = 0.0
CASA$ LoadASCIIFile()
```

tips:

- you can save the information of the file on a variable, e.g.:

```
CASA$ expdata = LoadASCIIFile()
```

GetTransitions()

description:

to read/list entries from the database around a selected frequency

GetTransitions()

description:

to read/list entries from the database around a selected frequency

usage/inputs:

```
CASA$ expdata = data from previous step
CASA$ FreqMin = 210000.0
CASA$ FreqMax = 230000.0
CASA$ ElowMin = 0.0
CASA$ ElowMax = 300.0
CASA$ FrequencyWidth = 2.0
CASA$ GetTransitions()
```

tips:

- check example file my_transitions_INcasa.py

myXCLASS()

description:

to generate a synthetic spectrum

myXCLASS()

description:

to generate a synthetic spectrum

usage/inputs:

```
CASA$ FreqMin = 210000.0
CASA$ FreqMax = 230000.0
CASA$ FreqStep = 0.1
CASA$ t_back_flag = True/False
CASA$ tBack = 0.0
CASA$ tslope = 0.0
CASA$ nH_flag = True/False
CASA$ N_H = 1.e24
CASA$ beta_dust = 0.0
CASA$ kappa_1300 = 0.00
CASA$ iso__flag = True/False
CASA$ IsoTableFileName = "my_isotopologues_ratio.txt"
CASA$ MolfitsFileName = "molecules.molfit"
CASA$ myXCLASS()
[or CASA$ model, log, trans, IntOptical, JobDir = myXCLASS()]
```

The *MOLFIT* file

description:

indicates the parameters of the species to be considered

format:

```
% Number of molecules    1
%
% schema:
%
% name of molecule       number of components
% f  l  u  size    f  l  u  T_rot    f  l  u  N_tot    f  l  u  Dv    f  l  u  Vlsr    Aflag
%
CO;v=0;         1
  n  0  0  0.5     n  0  0  150.0    n  0  0  1.0e+16  n  0  0  5.0   n  0  0  -10.0   c
```

The *MOLFIT* file

description:

indicates the parameters of the species to be considered

format:

```
% Number of molecules    2
%
% schema:
%
% name of molecule       number of components
% f  l  u  size    f  l  u  T_rot    f  l  u  N_tot    f  l  u  Dv    f  l  u  Vlsr    Aflag
%
CO;v=0;         1
  n  0  0  0.5     n  0  0  150.0    n  0  0  1.0e+16  n  0  0  5.0   n  0  0  -10.0    c
CS;v=0;         1
  n  0  0  0.5     n  0  0  200.0    n  0  0  1.0e+15  n  0  0  8.0   n  0  0  -10.0    c
```

The *MOLFIT* file

description:

indicates the parameters of the species to be considered

format:

```
% Number of molecules    2
%
% schema:
%
% name of molecule       number of components
% f  l  u  size    f  l  u  T_rot    f  l  u  N_tot    f  l  u  Dv    f  l  u  Vlsr    Aflag
%
CO;v=0;         3
  n  0  0  0.5     n  0  0  150.0    n  0  0  1.0e+16  n  0  0  5.0   n  0  0  -10.0   c
  n  0  0  0.5     n  0  0   50.0    n  0  0  3.0e+15  n  0  0  3.0   n  0  0  -12.0   c
  n  0  0 10.0     n  0  0   30.0    n  0  0  1.0e+14  n  0  0  2.0   n  0  0   +8.0   f
CS;v=0;         1
  n  0  0  0.5     n  0  0  200.0    n  0  0  1.0e+15  n  0  0  8.0   n  0  0  -10.0   c
```

The *ISONAMES* file

description:

indicates the abundance ratios between isotopologues

format:

```
CH3CN;v8=1;          CH3CN;v=0;          1
C-13-H3CN;v=0;       CH3CN;v=0;          60
CH3C-13-N;v=0;       CH3CN;v=0;          60
CH3CN-15;v=0;        CH3CN;v=0;          300
SO;v=1;              SO;v=0;             1
S-34-O;v=0;          SO;v=0;             23
SO-18;v=0;           SO;v=0;             500
```

myXCLASSPlot()

description:

to plot the synthetic spectrum (with observational data)

myXCLASSPlot()

description:

to plot the synthetic spectrum (with observational data)

usage/inputs:

load an observational file with LoadASCIIFile() → expdata

generate a synthetic spectrum with myXCLASS()→ modeldata

```
CASA$ expdata = expdata
CASA$ modeldata = modeldata
CASA$ TransEnergies = trans
CASA$ xLowerLimit = 218000.0
CASA$ xUpperLimit = 230000.0
CASA$ yLowerLimit = -1.0
CASA$ yUpperLimit = 40.0
CASA$ myXCLASSPlot()
```

tips:

- check example file my_plot_INcasa.py

myXCLASSFit()

description:

to fit observational data with a synthetic spectrum (uses MAGIX)

myXCLASSFit()

description:

to fit observational data with a synthetic spectrum (uses MAGIX)

usage/inputs:

```
CASA$ NumberIteration = 100
CASA$ MolfitsFileName = 'my_molecules.molfit'
CASA$ experimentalData = 'my_observation.xml'
CASA$ AlgorithmXMLFile = 'my_algorithm.xml'
CASA$ newmolfit, modeldata, JobDir = myXCLASSFit()
```

tips:

- the `experimentalData` variable can be an ASCII file
- check example file my_fit_INcasa.py

The *my_observation.xml* file

description:

contains information on the observational data

usage:

check example: my_observation.xml

basically it contains:

- the ASCII file (xml format)
- frequency range covered
- background source
- hydrogen column density and dust (beta, kappa, N_H)
- isotopologues

tips:

- you can provide different observational ASCII files
- you can fit different frequency regimes within an ASCII file

The *my_algorithm.xml* file

description:

contains information on the algorithm that you want to use to fit data

usage:

check example: my_algorithm.xml

different options:

- Levenberg-Marquardt (local optimization, very fast)
- Simulated Annealing (local optimization, fast)
- Particle Swarm Optimization (global optimization, good convergence)
- Bees algorithm (global optimization, explore the landscape)
- Genetic algorithm (global optimization, good convergence)
- Nested Sampling (global optimization, good convergence)
- Interval Nested Sampling (global optimization, fast convergence)
- Error Estimation

The *my_algorithm.xml* file

description:

contains information on the algorithm that you want to use to fit data

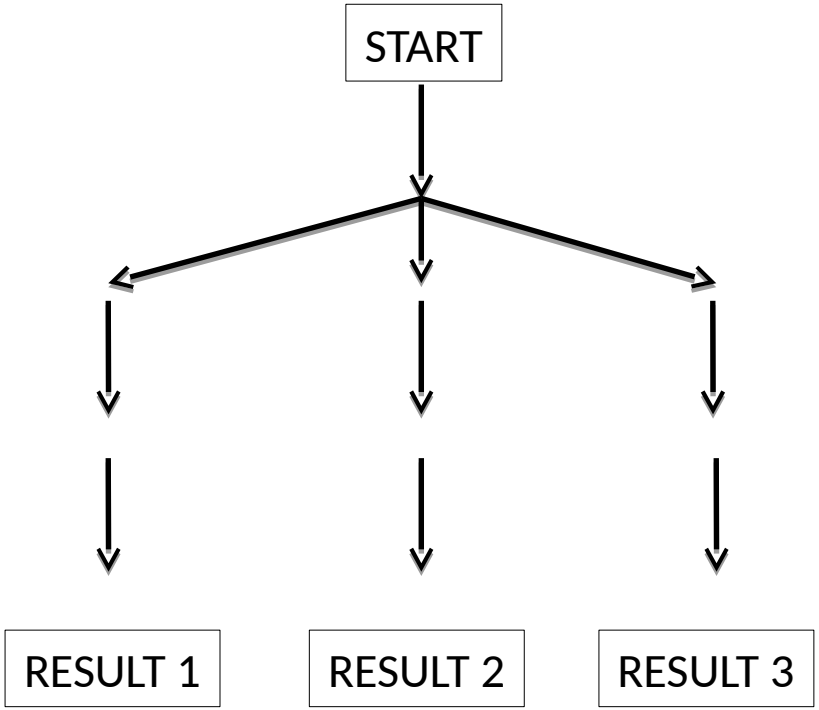
usage:

you can connect them!

1st algorithm: Bees

2nd algorithm: Levenberg-Marquardt

3rd algorithm: Error Estimation



myXCLASSMapFit()

description:

to fit a complete data cube instead of a single spectrum

myXCLASSMapFit()

description:

to fit a complete data cube instead of a single spectrum

usage/inputs:

```
CASA$ NumberIteration = 100
CASA$ MolfitsFileName = 'my_molecules.molfit'
CASA$ experimentalData = 'my_observation.xml'
CASA$ AlgorithmXMLFile = 'my_algorithm.xml'
CASA$ regionFileName = ' '
CASA$ UsePreviousResults = True/False
CASA$ Threshold = 0.1
CASA$ clusterdef = ' '
CASA$ myXCLASSMapFit()
```

tips:

- check example file my_mapfit_INcasa.py

LineIdentification()

description:

line identification routine

LineIdentification()

description:

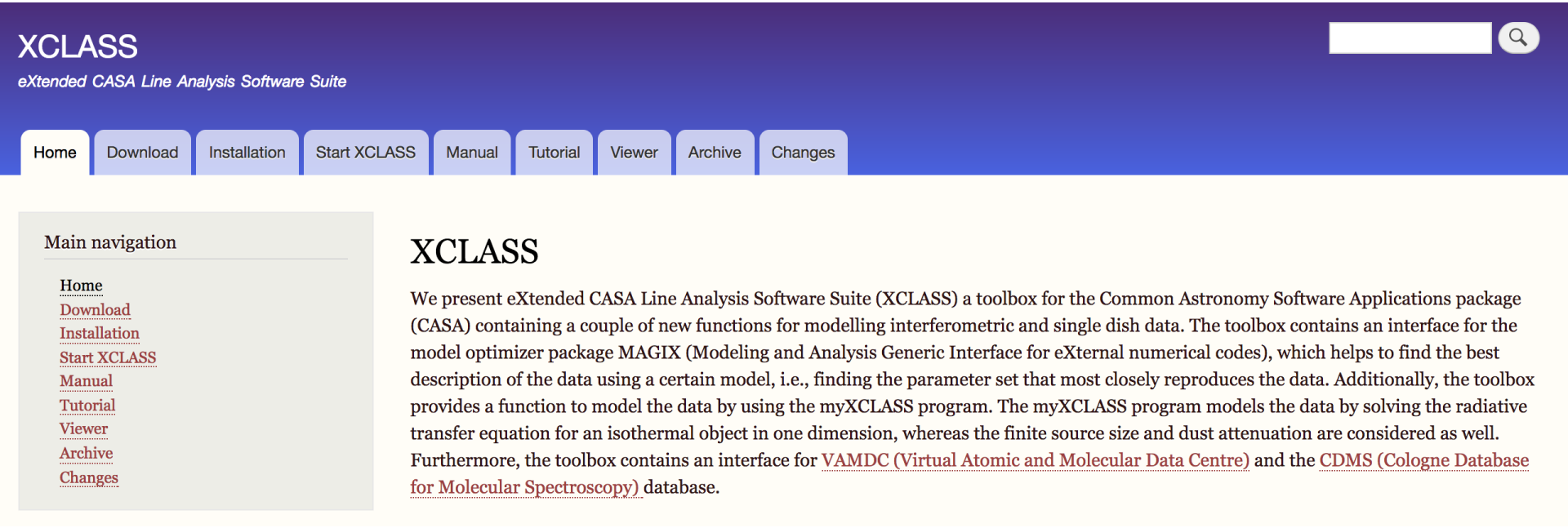
automatic line identification routine

usage/inputs:

```
CASA$ Noise = 0.5
CASA$ MaxOverestimationHeight = 500.0
CASA$ Tolerance = 65.0
CASA$ MinColumnDensityEmis = 0.0
CASA$ MinColumnDensityAbs = 0.0
CASA$ SourceName = ""
CASA$ DefaultMolfitFile = "my_LineID__default.molfit"
CASA$ SelectedMolecules = ["HCCCN;v=0;", "CH3OH;v=0;",
        "C2H5OH;v=0;", "CH3CN;v=0;", "SO;v=0;", "SO2;v=0;"]
CASA$ StrongMoleculeList = []
CASA$ NumberIteration = 10
CASA$ AlgorithmXMLFileSMF = ""
CASA$ AlgorithmXMLFileOverAll = ""
CASA$ experimentalData = "my_observation__LineID.xml"
CASA$ IdentifiedLines, JobDir = LineIdentification()
```

Remember

XCLASS for CASA can be downloaded from:
https://xclass.astro.uni-koeln.de/



download the available zip file and execute

```
python install-in-casa.py --smp
```

THANKS, questions?