FASMA 2.0

Stellar spectral analysis package.

FASMA delivers the atmospheric stellar parameters (effective temperature, surface gravity, metallicity, microturbulence, macroturbulence, and rotational velocity) based on the spectral synthesis technique. The principle of this technique relies on the comparison of synthetic spectra with observations to yield the best-fit parameters under a χ^2 minimization process. FASMA 2.0 is now updated to deliver also chemical abundances of 13 elements (Li, Na, Mg, Al, Si, Ca, Sc, Ti, V, Cr, Mn, and Ni).

The python code runs with python 3 and is wrapped around the spectral synthesis package in fortran: MOOG version 2019 (Sneden et al. 1973).

The spectral synthesis technique requires model atmospheres for the calculation of the synthetic spectra by MOOG which is provided with FASMA. FASMA includes two grids of models in MOOG readable format, Kurucz and marcs that cover the parameter space for both dwarf and giant stars with metallicity limit of -5.0 dex. The instructions below are also included here.

Contact here for bugs: tsantaki@arcetri.astro.it

Installation

Installing FASMA requires 2 simple steps.

1) Run the installation script `install_fasma.sh` , in the `FASMA-synthesis` folder, in order to unzip the model atmospheres and install MOOG (MOOG uses gfortan, f77, or g77 compilers). During this installation, the user will be asked about system requirements: 'rh64' for 64-bit linux system, 'rh' for 32-bit linux system, 'maclap' for mac laptop, 'macdesk' for mac desktop.

```
$ ./install_fasma.sh
```

2) Then install FASMA with pip:

```
$ pip install .
```

Or, if you do not have sudo access, replace the pip command with:

```
$ pip install --user .
```

A list of basic python packages will be installed automatically with pip (see requirements).

To uninstall FASMA:

```
$ pip uninstall FASMA
```

Usage

FASMA by the configuration file

FASMA is so easy. You can run FASMA from the terminal by configuring the options in the `config.yml` file and then run in the working directory:

```
$ fasma
```

If the configuration file is called something else (e.g. myConf.cfg), then run:

```
$ fasma myConf.cfg
```

FASMA produces a log file, `fasma.log`, which catches errors in order to inform the user.

FASMA with scripts

For large lists of stars, it is preferable to use FASMA inside scripts. The configuration options are inserted from a dictionary. The main functions to use is `FASMA()` and the `result()` to obtain the output.

```
FASMA(cfgfile='config.yml', overwrite=None, **kwargs)

Input

cfgfile : str
  Configuration file (default: config.yml)
overwrite : bool
  Overwrite the synthresults.dat file (default: False)
**kwargs : dict
  Options to overwrite the default configuration options

Output

synthresults.dat : file
  Easy readable table with results (tab separated)



result()

Output

result : dict
  output parameters in dictionary
```

The expected output dictionary of `result()` includes the following:

```
if 'element' is True:
  'linelist': str
  'observations': str
  'teff': float
  'logg': float
  'feh': float
  'vt': float
  'vmac': float
  'vsini': float
  'element': str
  'abund': float
  'erabund': float
  'chi2': str
  'time': int
  'model': str
```

```
  'resolution': int
  'snr': int
  'spectrum': {
  'wave' : float,
  'flux' : float}


if 'minimize' is True:
  'linelist': str
  'observations': str
  'teff': float
  'erteff': float
  'logg': float
  'erlogg': float
  'feh': float
  'erfeh': float
  'vt': float
  'ervt': float
  'vmac': float
  'ervmac': float
  'vsini': float
  'ervsini': float
  'chi2': float
  'time': int
  'model': str
  'resolution': int
  'snr': int
  'spectrum': {
  'wave' : float,
  'flux' : float}
```

The above functions can be used as in this example:

```
from FASMA import FASMA

options = {'observations': '/home/FASMA-synthesis/FASMA/spectra/Sun_HARPS.fits',
           'minimize': True,
           'plot':True}
driver = FASMA(**options)
result = driver.result()
feh = result['feh']
```

The input options are presented in the next section and have the save terminology as in the configuration file. The result is a dictionary with the final parameters which are saved to a file (`synthresults.dat` or `synthresults_elements.dat`).

Configuration file

The complete list of options of the configuration file has the following format. These are the default values:

```
star:
  teff: 5777
  logg: 4.44
  feh: 0.0
  vmac: 3.21
  vsini: 1.9
  vt: 1.0
  MOOGv: 2019
  damping: 1
  element: False
  fix_feh: False
  fix_logg: False
  fix_teff: False
  fix_vmac: False
  fix_vsini: False
  fix_vt: False
  intervals_file: 'intervals.lst'
  limb: 0.6
  linelist: 'linelist.lst'
  minimize: False
  model: 'apogee_kurucz'
  observations: False
  plot: False
  plot_res: False
  refine: False
  resolution: null
  save: False
  snr: null
  step_flux: 3.0
  step_wave: 0.01
```

The configuration file (yaml format) is space sensitive and it can include only the values which

are other than the defaults. The user can check if the format is correct here. This file must be in the folder where FASMA runs.

There are various configuration combinations depending on what the user wants. **A mandatory requirement is to insert the complete path of the line list and the interval file. The option Number 2 below is recommended for the parameter determination.**

1. This is the simplest option for FASMA where a synthetic spectrum is created with solar values and plotted. The rest options not listed are set to default.

```
star:
 linelist: /home/FASMA-synthesis/FASMA/rawLinelist/linelist.lst
 intervals_file: /home/FASMA-synthesis/FASMA/rawLinelist/intervals.lst
```

The default line list is in the rawLinelist folder already provided within FASMA.

2. For a given line list select your options, e.g.:

```
star:
  linelist: /home/FASMA-synthesis/FASMA/rawLinelist/linelist.lst
  intervals_file: /home/FASMA-synthesis/FASMA/rawLinelist/intervals.lst
  fix_vmac: true
  fix_vt: true
  minimize: true
  refine: true
  observations: /home/FASMA-synthesis/FASMA/spectra/Sun_HARPS.fits
  resolution: 115000
  plot: true
```

This example indicates that for the observed spectrum `Sun_HARPS.fits` with resolution 115000, FASMA will initialize the minimization process starting from solar values, keeping the parameters vmac and vt fixed and using the `refine` option (see below). This option was used to derive parameters in Tsantaki et al. (2018). The spectral regions for the synthesis are defined in the `intervals.lst` file with the option intervals_file. The final spectrum will be plotted. For the not listed options, FASMA uses the default ones.

3. For a given line list and a set of initial values, a synthetic spectrum is creared, saved in the `results` folder, and plotted.

```
star:
 teff: 5777
 logg: 4.44
 vmac: 3.21
 vsini: 1.9
 vt: 1.0
 linelist: /home/FASMA-synthesis/FASMA/rawLinelist/linelist.lst
 intervals_file: /home/FASMA-synthesis/FASMA/rawLinelist/intervals.lst
 resolution: 115000
 plot: true
 save: true
```

4. This option will deliver the chemical abundance of Ti but can be adopted for any of the provided elements (Li, Na, Mg, Al, Si, Ca, Sc, Ti, V, Cr, Mn, Ni). Note that for this option the line list and intervals are different than for the stellar parameters. The options `minimize` and `element` contradict each other.

```
star:
 teff: 5777
 logg: 4.44
 vmac: 3.21
 vsini: 1.9
 vt: 1.0
 element: 'Ti'
 linelist: /home/FASMA-synthesis/FASMA/rawLinelist/elements.lst
 intervals_file: /home/FASMA-synthesis/FASMA/rawLinelist/intervals_elements.lst
 observations: /home/FASMA-synthesis/FASMA/spectra/Sun_HARPS.fits
 resolution: 115000
 plot: true
 save: true
```

5. For multiple stars:

```
HD105:
 element: 'Ti'
 linelist: /home/FASMA-synthesis/FASMA/rawLinelist/elements.lst
 intervals_file: /home/FASMA-synthesis/FASMA/rawLinelist/intervals_elements.lst
 observations: /home/FASMA-synthesis/FASMA/spectra/HD105_HARPS.fits
 resolution: 115000
```

```
HD1156:
 element: 'Ti'
 linelist: /home/FASMA-synthesis/FASMA/rawLinelist/elements.lst
 intervals_file: /home/FASMA-synthesis/FASMA/rawLinelist/intervals_elements.lst
 observations: /home/FASMA-synthesis/FASMA/spectra/HD1156_HARPS.fits
 resolution: 115000
```

Default Options

The default options of FASMA can be changed in the configuration file `config.cfg` or inserted in the `FASMA()` function.

1. The model atmospheres included in FASMA are: apogee_kurucz, and marcs.

```
model:        marcs
```

2. The version of MOOG currently is 2019 but can easily updated to newer versions. This is relevant only if the output files change for future updates of MOOG.

```
MOOGv:        2019
```

3. If save is True, the output synthetic spectrum is saved in the folder `results` . The output is saved in a .spec format and is read with the astropy.io fits module.

```
save:         False
```

4. If any of these options is set to True, then they are fixed during the minimization process.

```
teff:         False
logg:         False
feh:          False
vt:           False
vmac:         False
vsini:        False
```

5. These are plotting options of the synthetic/observed spectra (plot) and of their residuals in case of minimization (plot_res).

```
plot:          False
plot_res:      False
```

6. The damping option of MOOG to deal with the van der Waals broadening. The acceptable values are: 0, 1, 2.

```
damping:       1
```

7. The wavelength step for the synthetic spectrum to be created (step_wave) and the flux limit to consider opacity distributions from the neighboring lines (step_flux), both measured in Angstrom. These are the same options in MOOG.

```
step_wave:     0.01
step_flux:     3.0
```

8. If minimize appears in the options, the minimization process will start to find the best-fit parameters for the observed spectrum, therefore observations must be included in the options.

```
minimize:      False
```

9. The refine option performs corrections on the derived parameters after the minimization for a more detailed analysis. After a first run, a second minimization starts with optimized microturbulence and macroturbulence values according to the parameters of the first run. Refine will also remove bad flux points and missing lines.

```
refine:        False
```

10. The name of the spectrum is given here. The full path of the spectrum should be given. Note that the spectrum has to be in a specific format (see below).

```
observations: False
```

11. These files contain the lines with their atomic parameters (`linelist.lst`) and the intervals of the synthesis (`intervals.lst`). FASMA already has specific lists in the `rawLinelist` folder. The complete path should be given.

```
linelist:   linelist.lst
intervals_file:   intervals.lst
```

12. The signal-to-noise (SNR) value of the observed spectra can be given by the user (optional). It is used for the normalization process. Otherwise, the SNR is calculated by the PyAstronomy function within FASMA.

```
snr:          null
```

13. The resolution of the synthetic spectrum or the resolution of the observed spectrum. This value is necessary when minimization is on.

```
resolution:   null
```

14. The limb darkening coefficient for the vsini broadening calculations. It is set fixed to 0.6 but can have different values depending on luminosity class.

```
limb:         0.6
```

15. This option will start the minimization process to derive the chemical abundance of a specific element. The element has to be set from this list: Li, Na, Mg, Al, Si, Ca, Sc, Ti, V, Cr, Mn, Ni. In this case the line list and intervals are different than the standard derivation of the stellar parameters and should be changed to `elements.lst` and `intervals_elements.lst` , respectively located in the `rawLinelist` folder. For the derivation of the chemical abundance the minimize option should be False.

```
element:     False
```

Inputs

Input spectra

The main input is the stellar spectrum which has to be in a specific format.

1. a text file (.dat or .txt) with 2 columns for the wavelength and flux.
2. a fits file (.fits). FASMA reads fits files from the ESO archive (ESO SDP 1D spectrum) and also 1D files which include the key words 'CRVAL1' and 'CDELT1' in their header.
3. a table fits format (.spec). This is the format FASMA saves the output synthetic spectra if requested.

FASMA assumes the wavelength is in Angstrom and that the spectrum has been corrected for radial velocity shifts. This is an example on how to correct for such RV shifts. We advice the use of the fits format as reading them is much faster than the text files.

Input line lists

1. `linelist.lst` -- the line list for the derivation of stellar parameters as tested in Tsantaki et al. (2018)
2. `intervals.lst` -- the list of the spectral regions for the spectral synthesis
3. `elements.lst` -- the line list for the derivation of the chemical abundances from VALD
4. `intervals_elements.lst` the lines of the specific elements where the intervals are calculated from Adibekyan et al. (2015) and Delgado-Mena et al. (2015)

The user can adapt the above files (tab separated), such as adding more elements to the line lists.

Outputs

The delivered stellar parameters are saved in the `synthresults.dat` and the chemical abundances in the `synthresults_elements.dat`.

For a set of parameters (Teff, logg, [M/H]), a model atmosphere is obtained by interpolating from the model grid. The model atmosphere is created in the `out.atm` file. It is a tabulated form of the basic physical properties (temperature, pressure, etc.) at each layer of the atmosphere.

The raw synthetic spectrum without any broadening created by MOOG is saved in `summary.out` .

The `linelist.moog` contains the line list in the format readable by MOOG and `result.out` is a summary of all the parameters used for the synthesis of the spectrum.

The MOOG configuration file is `batch.par` and its options are set from the `config.yml` file.

AUTHORS

- M. Tsantaki
- D. Andreasen
- G. Teixeira