SPECIES: Spectroscopic Parameters and atmosphEric ChemIstriEs of Stars

Manual

Maritza Soto

October 31, 2017

Contents

Chapter 1

Introduction

SPECIES is a code written mostly in Python, that computes stellar parameters from high resolution echelle spectra. The code measures the equivalent width of the given spectra (more details on the accepted spectra in 2), and solves the local thermodynamic equilibrium equation for a set of temperature, metallicity, surface gravity and microturbulent velocity. It then uses those values to compute the broadening, macroturbulent and rotational velocities, and the mass and age, as well as uncertainties for each parameter.

This present manual is intended to help users understand how to perform the calculation of the parameters with SPECIES, and understanding the results and corresponding files. For an explanation of the methods used to compute the parameters, please read Soto et al. 2017.

(SPECIES went through some major changes. If you installed SPECIES before November 2017, please update all the packages and make sure you include the MOOGPATH in your bash file (section 1.1.4). It is not necessary to reinstall ARES nor MOOG).

1.1 Installation

SPECIES is publicly available in GitHub[1]. In order to install it, it is necessary to just place the corresponding folder in your preferred location. Before using it, make sure you have installed the python packages listed in the following section, as well as the ARES and MOOG codes (section 1.1.3, 1.1.4).

1.1.1 Contents

Before installing SPECIES, make sure all the necessary files and folders are included. The files are: SPECIES.py, atmos.py, calc_broadening.py, calc_errors.py, calc_params.py, ccf.py, photometric_relations.py, find_mass.py, interpol_function.py, instruments.py, spectra.py, and run_ARES.bash.

The folders, and its contents, are:

- `ARES2`
- `EW/`: `plots_spectra/`
- `MOOGFEB2017/`: MOOG files, abfind.par.

[1] `https://github.com/msotov/SPECIES/`

- MOOG_linelist/
- Spectra/: linelist.dat, lines_ab.dat, mine.opt, plots_ccf/, plots_fit.
- atm_models/: atlas9/
- binary_masks/: G2.mas, K0.mas, K5.mas, M2.mas.
- debugging_files/
- dotter_isochrones/: plots/
- output/: MOOG_output_files/, coeffs_error_files/
- plots_broadening/

1.1.2 Requirements

SPECIES uses a large number of python packages in order to optimize the calculation of the parameters. Most of them are included in your average python installation, but some of them have to be installed separately. These are:

- numpy (http://www.numpy.org)
- scipy (https://www.scipy.org)
- matplotlib (http://matplotlib.org)
  For SPECIES, it is necessary that the default backend for matplotlib is Agg, in order to avoid problems with the multiprocessing package. In order to change that, first locate where your matplotlibrc file is. Do the following:

```
import matplotlib
matplotlib.matplotlib_fname()
```

  and in the matplotlibrc file change the backend:

```
backend      : Agg
```

- astropy (http://www.astropy.org)
- astroquery (http://astroquery.readthedocs.io/en/latest/)
- PyAstronomy (http://www.hs.uni-hamburg.de/DE/Ins/Per/Czesla/PyA/PyA/index.html)
- isochrones (http://isochrones.readthedocs.io/en/latest/)
  We find that with MultiNest/PyMultiNest (http://astrobetter.com/wiki/MultiNest+Installation+Notes) we obtain more accurate results for the mass and age computation than with emcee (http://dan.iel.fm/emcee/current/), therefore we recommend having MultiNest installed in your computer. After installing isochrones, open your terminal and type:

```
nosetests isochrones
```

That should download the grid data. In case it gets interrupted, delete the `~/.isochrones` directory and try again.

- `uncertainties` (`https://pythonhosted.org/uncertainties/`)

1.1.3 ARES

We use the `Ares` code (Sousa et al. 2008) to compute the equivalent width for a set of lines in our spectra. It can be obtained from `http://www.astro.up.pt/~sousasag/ares/`, but a version of it is also included in SPECIES, with some minor changes from the original files. The requirements and installation are explained in the README file inside the ARES2 folder. If you're going to use the ARES version included in SPECIES, compile `ARES_v2_test.c` instead of `ARES_v2.c`. Also, call the compiled file as `ARES_v2`. This is necessary for running SPECIES, because the script that executes ARES calls the executable `ARES_v2`. All the other installation steps should remain the same.

We had some problems when using the latest version of `gsl`, so if you encounter any problem when installing ARES, we recommend changing the version of `gsl` to the previous one (for example, we use `gsl-1.16`).

Note: Please make sure that you created a link into your link directory for ARES, by including

`$ln -s compiledfile ~/bin/ARES_v2`

in your `bash` file. This is necessary for later executing ARES.

1.1.4 MOOG

We use MOOG (`http://www.as.utexas.edu/~chris/moog.html`) in order to solve the local thermodynamic equation, for a set of parameters. A version of MOOG is included in SPECIES, from the February 2017 release. The version included in SPECIES does not require for *SM* to be installed (MOOGSILENT), and it is the one used in SPECIES. In order to install it, first delete all the .o files (executables), and modify the following files:

- Makefile.XXsilent, where XX is the type of machine you are using (ie. Makefile.rh64silent or Makefile.macdesksilent).

  Change the compilers and path to the libraries:

```
CC = cc
FC = gfortran -w

# the following lines point to some needed libraries
X11LIB = /usr/X11R6/lib
SMLIB = /Applications/scisoft/i386/Packages/sm/lib
AQLIB = /usr/local/scisoft/lib
```

  If you don't have *SM* installed, then ignore the paths to `SMLIB` and `AQLIB`.

- Moogsilent.f

  Change the path of MOOG

```
moogpath = '/path/to/the/MOOG/folder'
```

  Change the type of machine (possible values are mac, pcl and uni)

```
machine = "mac"
```

Once that is changed, type: `make -f Makefile.XXsilent`, where XX is the type of machine.

Working with another version of MOOG

If you want to use a version of MOOG different from the one included in SPECIES, then in order to use it you will have to modify some other files, and check if some quantities agree with the values in SPECIES.

Files in MOOG

If you don't have *SM* installed, you'll have to modify the following files in the MOOG directory:

- `Abundplot.f`, `Binplot.f`, `Cogplot.f`, `Defcolor.f`, `Drawcurs.f`, `Fluxplot.f`, `Makeplot.f`, `Pointcurs.f`, `Specplot.f`

  Copy them into other files with different names (ie. `file_old.f`), and modify the originals so that they all look like:

```
      subroutine XXX
c*****************************************************
c     Description of the subroutine
c*****************************************************

      end
```

  with `XXX` is the name of each subroutine mentioned above. If you have questions, check those files in the `MOOG` file included in SPECIES.

- In `Makefile.XXsilent` (with XX the type of machine), reaching the end of the file, comment the line that says
  `-L$(SMLIB) -lplotsub -ldevices -lutils -L$(AQLIB) -laquaterm`, so that it will look like:

```
MOOGSILENT:  $(OBJECTS);
    $(FC) $(OBJECTS) -o MOOGSILENT -L$(X11LIB) -lX11 \
#   -L$(SMLIB) -lplotsub -ldevices -lutils -L$(AQLIB) -laquaterm
```

Files outside the MOOG directory

- `calc_params.py`: In the function `runMOOG_ab`, close to the end of the file, check that the abundances being subtracted to each elements agree with the ones in `Batom.f`, in the MOOG directory. The ones used are from Asplund et al. 2009. Currently, it looks like this:

```
ab_NaI = abund['ab']['NaI'] - 6.24
ab_MgI = abund['ab']['MgI'] - 7.60
ab_AlI = abund['ab']['AlI'] - 6.45
ab_SiI = abund['ab']['SiI'] - 7.51
ab_CaI = abund['ab']['CaI']- 6.34
ab_TiI = abund['ab']['TiI']- 4.95
ab_TiII = abund['ab']['TiII']- 4.95
ab_CrI = abund['ab']['CrI'] - 5.64
ab_MnI = abund['ab']['MnI'] - 5.43
ab_FeI = abund['ab']['FeI'] - 7.50
ab_FeII = abund['ab']['FeII'] - 7.50
ab_NiI = abund['ab']['NiI'] - 6.22
ab_CuI = abund['ab']['CuI'] - 4.19
ab_ZnI = abund['ab']['ZnI'] - 4.56
```

- `calc_params.py`: In the function `runMOOG` (at line $\sim$ 200), check that the abundance being subtracted to Fe agrees with the abundance for iron in `Batom.f`, in the MOOG directory. Currently, it looks like this:

```
ab = ab - 7.50
```

- `interpol_function.py`: Close to the end of the file, check that the value being added to the iron abundance agrees with the one being subtracted in the `runMOOG` function, included in `calc_params.py`. Currently, it looks like this:

```
logefe = 7.50 + xmetal
```

- Verify the path to the MOOG directory in the functions `run_iteration` and `calc_ab`, both included in `calc_params.py`, and in `run_moo.bash` and `run_moog_ab.bash`.

Finally, do not forget to copy `abfind.par` from the MOOG directory in SPECIES, to your new MOOG directory, with its latest release.

The distribution in the results might also vary from the ones presented in Soto et al. 2017, so be careful when applying any kind of correction from the paper into your new results. The benchmark values we used to calibrate our results might also change, therefore if you're not sure about how the results will change when using another MOOG version, send me an email and I'll try to hep.

Create the MOOGPATH

This new version of SPECIES does not use the .bash files from the previous version. Instead, the MOOG executable can be called from whatever directory you are working from. In order to do that, it is necessary to include in your $\sim$/.bash file the path to the MOOG directory. In my case, that bash file is called $\sim$/.bash_profile (Mac Desktop machine, for Linux should be something like .bash or .bashrc). Include the following line:

```
export PATH="/full/path/to/MOOG:$PATH"
```

Then, in the terminal type `source ~/.bash_file`, where `~/.bash_file` is your .bash file.

1.2 Checking the installation

In order to test if SPECIES was correctly installed, we have included one of the sun's spectra used in our calculus. It's located in `Spectra/sun01_harps.fits`. The idea is to execute SPECIES using that spectra, and see if it runs smoothly, or if there was any error or missing file. Move to the SPECIES directory and type the following in your terminal:

```
python SPECIES.py -starlist sun01 -output sun01
```

If you don't encounter problems, then the files `output/sun01.fits` and `output/sun01.dat` should have been created. They included the final results from SPECIES for that certain spectrum. A description of the input commands is described in the next section, and the files received and created by SPECIES are described in nest chapters.

If you do encounter problems, please check if MOOG and ARES were correctly installed. In your terminal, you could type `ARES_v2`, and it should recognize the command if ARES is installed (that can be typed in any directory). For MOOG, type `MOOGSILENT` in the command line. If it recognizes the command, then MOOG is correctly installed. If you cannot solve your problem, please post it on the GitHub repository for SPECIES, or send an email to `maritsoto@ug.uchile.cl`.

1.3 Running SPECIES

In order to execute SPECIES, you need a high resolution echelle spectra, plus other auxiliary files, which will be better explained in the next chapter.

The Spectra has to be placed in the `Spectra/` folder, with the following name format: `starname_inst.fits`, where `inst` is the name of the instrument used to obtain the spectra, in non-capital letters.

Once that is ready, along with the auxiliary files (if needed), then execute SPECIES by typing

```
python SPECIES.py
```

You can also add or modify some options for SPECIES, by adding them on the command line. These are:

- `-starlist`: Name of the stars to use, separated by commas. If the list is too long, retrieve it from inside the code, in the part that says

  ```
  # List of stars for which to compute the parameters
  ```

- `-inst`: Names of the instruments you want to use, separated by commas and in non-capital letters. If "all" is included, spectra from all accepted instruments will be used. Default is "all".
- `-ncores`: Number of cores to use. Default is 3.
- `-output_format`: Format of the output file. Choices are ['Both', 'ASCII', 'FITS']. Default is 'Both'.
- `-output`: Name of the output file, without extension. Default is 'summary_output'.

- `-hold_params`: Option to set the atmospheric parameters to a certain value. When used, no argument has to be given.
- `-file_params`: Name of the file with the values for the parameters to hold. Will be used only if `-hold_params` was activated.
- `-hold_mass`: Option to set the mass, age, and photometric logg to a certain value. When used, no argument has to be given.
- `-file_mass`: Name of the file with the hold values for the mass, age and photometric logg. Will only be used if `-hold_mass` was activated.
- `-set_boundaries`: Option to change the boundaries of the atmospheric parameters. When used, no argument has to be given.
- `-file_with_boundaries`: Name of the file with the boundaries for the atmospheric parameters. Will only be used if `-set_boundaries` was activated.
- `-no_abundance`: Option to avoid the computation of atomic abundances. When used, no argument has to be given.
- `-no_error`: Option to avoid the computation of uncertainties to the atmospheric parameters. When used, no argument has to be given.
- `-no_debug`: Option to avoid the creation of a debug file for each star, where the details of the computation will be stored. When used, no argument has to be given.
- `-no_check_t`: Option to avoid checking the temperature from SPECIES with the one obtained from photometric relations. The temperature from photometry will still be computed, but it won't be compared with the one obtained from SPECIES. When used, no argument has to be given.
- `-file_colors`: Name of the file with photometric colors. If they are not given, the colors will be retrieved from Vizier.
- `-use_coords`: Option to use the coordinates of the star, instead of its name, to look for the star colors. When used, no other arguments has to be given.
- `-file_coords`: Name of the file with the coordinates of the stars (in deg). Used only if `-use_coords` was activated.
- `-mass_from_file`: Option to load the chains created during a previous mass calculation, stored in `dotter_isochrones/$\ast$.h5`
- `-vt_hold`: Option to set the microturbulence = 1.2 km/s when there is no convergence in the atmospheric parameters. It is only done when a set value for the microturbulence wasn't given previously, or the number of Fe I lines detected for the spectrum is larger than 10.

For example, if I want to run SPECIES for the star HD10700, using only the HARPS spectra available, the coordinates listed in the fits header, the mass and age measurements from a file, and the output file to be called 'summary_HD10700', then I'd type:

```
python SPECIES.py -output out_HD10700 -use_coords -starlist HD10700 -inst
    harps -hold_mass -file_mass HD10700_mass.txt
```

Chapter 2

Input files

As mentioned in the previous chapter, the input files needed to run SPECIES are a high resolution echelle spectra (.fits file) and a set of auxiliary file, which are not mandatory.

2.1 Input spectra

SPECIES currently accepts spectra from HARPS, FEROS, UVES, HIRES, AAT and CORALIE. The files have to be placed in the `Spectra/` directory, and are explained as follow:

- **HARPS**: The HARPS spectra can be retrieved from the Phase 3 data retrieval in the ESO archive. SPECIES uses the s1d fits file, for the A fibre, and consist of the spectra from all the orders already combined. The file has to be called `starname_harps.fits`. It also accepts spectra reduced using the `ceres`[1] pipeline. In those cases, the spectra will be divided by orders, which SPECIES will identify and treat the data accordingly (see the **FEROS** file handling).
- **FEROS**: The FEROS spectra can be retrieved from the Phase 3 data retrieval in the ESO archive, reduced by the ESO pipeline, or can be reduced data from other pipeline. The difference between both files is that, for the case of the data processed with the ESO pipeline, its format is very similar to the HARPS one, that is, all the orders are already combined. The spectra from other pipelines usually comes with each spectral order separated, in which case SPECIES has to combine them to create the 1D spectrum used by ARES. When combining the spectral orders, if two consecutive orders overlap in wavelength, then SPECIES will select the data from the red order (that is because the performance of FEROS improves in the red part of the spectrum, increasing the signal-to-noise of the data). If the input FEROS spectra has the spectral orders already combined, then the file has to be called `starname_feros.fits`. In the case when the spectral orders are separated, then the file has to be called `starname_feros_orders.fits`.
- **UVES**: The UVES spectra can be retrieved from the Phase 3 data retrieval in the ESO archive. Ii has to consist of two files, one for the blue part of the spectrum, and one for the red part. SPECIES will take both files and combine the spectra in order to create the file accepted by ARES. If there is overlap in wavelength between the red and blue parts, then SPECIES will select the data from the highest signal-to-noise (S/N) part. The files have to be called `starname_uves_red.fits` and `starname_uves_blue.fits`.

[1] `https://github.com/rabrahm/ceres`

- **HIRES**: The HIRES spectra can be obtained from the KOA data archive[2]. The spectra has to be already reduced, and without the iodine cell. Before downloading the files, make sure that the spectra is present in most of the spectral orders. It is not necessary to include the calibration files. SPECIES will combine the spectra from the different spectral orders, taken the data with the highest signal to noise (S/N) for the case when the orders overlap. The resulting spectra will be placed in `Spectra/starname_hires.fits`. The folder with the files has to be called `starname_hires/`, and be placed in the `Spectra/` directory.

2.2 Auxiliary files

These files contain additional information for SPECIES, consisting of extra information for the stellar parameters. They are described as follows:

- **file params**: This file contains information about the atmospheric parameters, as well as broadening, macroturbulence and rotational velocities. You can specify initial values for the parameters, and whether you want to keep that value uncharged through the computation of not. The file must have a header with the names of the columns, and each column has to be separated by spaces or tabs. They are as follow:

  1. `Starname`: `Starname_inst`, with `inst` is the instrument used to obtain the spectra.
  2. `hold`: Name (or list of names) of the parameters you wish to hold to a certain value during the computation. The names have to be comma-limited and with no spaces in between. Accepted names are `temperature`, `metallicity`, `pressure`, and `velocity`.
  3. `T`: value for the temperature. If `temperature` is not in hold, then this will be taken as the initial value. If none if given, replace the number with `no`.
  4. `logg`: value for the surface gravity. If `pressure` is not in hold, then this will be taken as the initial value. If none if given, replace the number with `no`.
  5. `met`: value for the metallicity. If `metallicity` is not in hold, then this will be taken as the initial value. If none if given, replace the number with `no`.
  6. `micro`: value for the microturbulence velocity. If `velocity` is not in hold, then this will be taken as the initial value. If none if given, replace the number with `no`.
  7. `broadening`: no longer used, replace with `no`.
  8. `err_broad`: no longer used, replace with `no`.
  9. `vsini`: value for the rotational velocity. If none is given, replace the number with `no`.
  10. `err_vsini`: error for the rotational velocity. If none is given, replace the number with `no`.
  11. `vmac`: value for the macroturbulence velocity. If none is given, replace the number with `no`.

[2] `http://www2.keck.hawaii.edu/koa/public/koa.php`

12. `err_vmac`: error for the macroturbulence velocity. If none is given, replace the number with `no`.

For example:

```
Starname    hold                        T       logg met    micro
   broadening   err_broad    vsini   err_vsini  vmac  err_vmac
HD55_harps  temperature,metallicity  4760.1  no   -0.72  1.123
   no          no          1.38    0.44       2.33  0.54
```

- **file mass**: File with information regarding the mass, age, and photometric logg for each star. The file must have a header with the names of the columns, and each column has to be separated by spaces or tabs. They are as follow:

1. `Starname`: `Starname_inst`, with `inst` is the instrument used to obtain the spectra.
2. `mass`: Value for the mass (in $M_{\odot}$).
3. `err_mass`: Uncertainty for the mass.
4. `age`: Value for the age (in Gyr).
5. `err_age`: Uncertainty for the age.
6. `logg_p`: Value for the photometric logg.
7. `err_logg_p`: Uncertainty for the photometric logg.

For example:

```
Starname    mass  err_mass    age   err_age  logg_p  err_logg_p
HD55_harps  0.63  0.021       4.94  5.5      4.7     0.02
```

- **file with boundaries**: File with the ranges accepted for the atmospheric parameters. The file must have a header with the names of the columns, and each column has to be separated by spaces or tabs. They are as follow:

1. `Starname`: `Starname_inst`, with `inst` is the instrument used to obtain the spectra.
2. `temperature`: Minimum and maximum accepted values for the temperature, separated by commas. If none is given, replace it with `no`.
3. `metallicity`: Minimum and maximum accepted values for the metallicity, separated by commas. If none is given, replace it with `no`.
4. `gravity`: Minimum and maximum accepted values for the surface gravity, separated by commas. If none is given, replace it with `no`.
5. `velocity`: Minimum and maximum accepted values for the microturbulence velocity, separated by commas. If none is given, replace it with `no`.

For example:

```
Starname  temperature  metallicity  gravity  velocity
NG021676  no           no           4.0,5.0  no
```

- **file colors**: File with photometric information for each star. The file must have a header with the names of the columns, and each column has to be separated by spaces or tabs. They are as follow:

  1. `Starname`: Name of the star, without the instrument.
  2. Apparent magnitudes: Magnitude in several bands for each star. If no magnitude is known, replace it with `no`. Magnitudes are: B, V, R, I, J, H, K, B_t, V_t, b, y.

  For example:

```
Starname     B    V    R    I    J    H    K    Bt   Vt   b    y
star01   12.19    11.60    no   no   10.51    10.27    10.22    no   no   no   no
```

- **file coords**: File with coordinates for each star, in `deg`. The columns are `Starname`, `RA`, and `DEC`, all separated by spaces or tabs. The file header must include the name of the columns.

Chapter 3

Output files

This chapter explains all the files created by SPECIES, including plots and final results.

3.1 Spectral files

These are the files that involve working with the spectra alone, before computing the equivalent width for the spectral lines.

1. Combined spectra: These are the files created when it is necessary for SPECIES to combine spectral orders into only one spectrum. This is done for the feros, uves, and hires data. For the case of FEROS, the combined spectra will still be called `starname_feros_orders.fits`, and the original file will adopt the name `starname_feros_orders_original.fits`. For UVES and HIRES, the combined spectra will be called `starname_uves.fits` and `starname_hires.fits`, respectively. These files will be stored in the `Spectra/` directory.

2. Restframe spectra: This is the spectra corrected to restframe, and corresponds to the input given to ARES. The velocity correction to apply is computed by performing a cross-correlation function (CCF) between the spectra and a binary mask, consisting on all the lines between 5500 and 6050Å, for the G2 spectral type mask. The files are called `starname_inst_res.fits`, and are stored in the `Spectra/` directory.

3. Plot CCF: These plots show the CCF between the spectra and the binary mask, used to correct the spectra to restframe. The top panel shows three spectral lines (6021.8, 6024.06 and 6027.06 Å) with its position in the original spectra (grey), and in the corrected spectra (black), along with three vertical lines representing the correct restframe position (red). The bottom panel show the corresponding CCF, with its computed velocity. They are stored in `Spectra/plots_ccf/`

4. Plot lines: These plots show four spectral lines (6021.8, 6024.06, 6027.06 and 6562.8 Å) in the restframe-corrected spectrum, along with red lines corresponding to their restframe position. They are stored in `Spectra/plots_fit/`

3.2 Equivalent width files

These are the files produced by ARES, with the equivalent width (EW) information.

1. EW for the iron lines: File with the fit parameters and EW for the iron lines from the file `linelist.dat`. They are called `starname_inst.ares`, and are stored in `EW/`.

2. EW for the atomic abundances: File with the fit parameters and EW for the spectral lines from the file `lines_ab.dat`, which are used to compute the abundance of several elements. They are called `starname_inst_ab.ares`, and are stored in `EW/`.

3. Plot spectra: Plots of the 1D spectra for each star, with red lines showing the position of the lines used to obtain the EW. The inset shows the position of the H-alpha line. They are stored in the directory `EW/plots_spectra/`

3.3 Debugging files

These files describe each step of the parameters computation. They are useful for the cases when there is a problem with the code, and one wants to know in which part of the computation it happened. SPECIES is intended to be completely automatic, therefore it is possible to follow the computation for each star by studying the debugging files. These are stored in `debugging_files/`

3.4 MOOG

These files correspond to the MOOG output, when assuming an atmospheric model with the correct parameters. They correspond to:

1. `output/MOOG_output_files/starname_inst_out.test`

2. `output/MOOG_output_files/starname_inst.test`

3. `output/MOOG_output_files/starname_inst_ab_out.test`

4. `output/MOOG_output_files/starname_inst_ab.test`

3.5 Rotational and macroturbulent velocity

These are plots created during the computation of the rotational and macroturbulent velocity. They are located in `plots_broadening/`.

1. `starname_inst_a_vsini.pdf`: Shows the change of S with the abundance (left panels) and rotational velocity (right panels). Each line represents a different line.

2. `starname_inst_vsini_new.pdf`: The the line fits, created with the final values for $v \sin i$ and v_{mac}. The panels with red background are the lines that were not included in the final values, due to them not representing a good fit to the observed line.

3. `starname_inst_vsini_paper.pdf`: Combination of the two previous files, for the purposes of including it in a publication.

For a better understanding of the quantities being plotted please refer to the project paper.

3.6 Mass and age calculation

These are files produced during the computation of the mass, age, and photometric logg.

1. Chain files: These files contain the chains performed to obtain the mass, age and photometric logg. Before performing new chains, SPECIES looks for these files and uses the already computed information. The files are in `dotter_isochrones/starname_inst_samples.h5`

2. plots CMD + distribution: These plots show the position of the star in the logg vs T_{eff} diagram, using the values already computed by SPECIES. The lines represent isochrones with the same metallicity as our star, but with different masses. The black line is the isochrone corresponding to the mass obtained in SPECIES. The other panels show the distribution of mass, age, and photometric logg obtained. These plots are found in `dotter_isochrones/plots/starname_inst_CMD_dist.pdf`

3. plots distribution: These plots show the distribution of mass, age and photometric logg found by SPECIES. The blue lines are the median for each distribution. The correlation between each quantity is also shown. These plots are found in `dotter_isochrones/plots/starname_inst_mass_age.pdf`

3.7 Final output

This file contains the final values for the parameters computed by SPECIES. It can be found in the `output/` directory, under the name and extension provided by the user. It has the following columns:

1. `Starname`
2. `Instrument`
3. `[Fe/H]`
4. `err_[Fe/H]`
5. `Temperature`
6. `err_T`
7. `logg`
8. `err_logg`
9. `vt`
10. `err_vt`
11. `nFeI`
12. `nFeII`
13. `exception`
14. `vsini`

15. err_vsini
16. vmac
17. err_vmac

18-51. [M/H], e_[M/H], nM

52. exception_Fe
53. exception_Ti
54. [FeI/H]
55. [FeII/H]
56. Mass
57. err_mass
58. Age
59. err_age
60. Photo_logg
61. err_photo_logg
62. Radius
63. err_radius
64. use_Tc
65. use_vt
66. err_vt2
67. err_T2

3.8 Temporary files

These are files created and used by SPECIES during the computation of the parameters, but should be deleted once the final results have been found. These are:

1. MOOG/abfind_starname_inst.par
2. MOOG/abfind_starname_inst_2.par
3. atm_models/starname_inst_n.atm
4. output/starname_inst_out.test
5. output/starname_inst.test
6. MOOG_linelist/lines.starname_inst.ares

7. MOOG/abfind_starname_inst_ab.par
8. MOOG/abfind_starname_inst_ab_2.par
9. atm_models/starname_inst_ab.atm
10. output/starname_inst_ab_out.test
11. output/starname_inst_ab.test
12. MOOG_linelist/lines.starname_inst_ab.ares