

# InversionPipeline

Version 1.0

Daniel Reese  
and  
Gaël Buldgen

October 2015

# Contents

<b>1</b>	<b>Getting started</b>	<b>3</b>
1.1	Running the program . . . . .	3
1.2	Using the program . . . . .	3
1.3	Preliminary remarks . . . . .	4
<b>2</b>	<b>Technical aspects</b>	<b>14</b>
2.1	Treatment of pulsation modes . . . . .	14
2.2	Inversions . . . . .	14
2.3	Combining inversion results from multiple reference models . . . . .	15
2.3.1	Different strategies for obtaining weights . . . . .	16
<b>3</b>	<b>File formats</b>	<b>16</b>
3.1	Script files . . . . .	16
3.2	Observed frequencies and rotational splittings . . . . .	17
3.3	Inversion types and parameters . . . . .	18
3.4	Constraints . . . . .	19
3.5	Result files . . . . .	21
3.6	Stellar models . . . . .	21
3.6.1	MOD . . . . .	22
3.6.2	CLES format . . . . .	22
3.6.3	LOSC format . . . . .	23
3.7	Grid of models . . . . .	24
3.8	GZIP compression . . . . .	24
<b>4</b>	<b>Known bugs</b>	<b>25</b>
<b>5</b>	<b>Copyright notices</b>	<b>25</b>
5.1	The rocket icon on the “Launch” button . . . . .	25
5.2	Source code for reading fortran binary files . . . . .	25
5.3	Supplementary notices . . . . .	25

## Acknowledgements

This program was written by Daniel Reese, whose work was supported by the SPACEINN network, a major international collaboration funded by the European Commission’s Seventh Framework Programme (FP7/2007-2013) under grant agreement no. 312844. Contributions by Gaël Buldgen have also been incorporated in this code.

# 1 Getting started

## 1.1 Running the program

`InversionPipeline` runs under Java 7.0 or later versions. If Java is not installed on your computer, or is not sufficiently up-to-date, it can be downloaded from:

`http://www.java.com/en/`

JRE (Java Runtime Environment) allows you to run Java programs but not to compile your own. JDK (Java Development Kit) allows you to run and compile Java programs.

To run the program download and extract the file `InversionPipeline.jar` from the following website:

`http://bison.ph.bham.ac.uk/spaceinne/inversionpipeline/index.html`

then type the following command in a command window, in the directory that contains `InversionPipeline.jar`:

```
java -jar InversionPipeline.jar
```

If you want to to run `InversionPipeline` with a script file, simply type:

```
java -jar InversionPipeline.jar script_file
```

where `script_file` is the name of your script file (its format will be described in a later section). Typing:

```
java -jar InversionPipeline.jar -help
```

or

```
java -jar InversionPipeline.jar --help
```

displays a very simple help text.

If you are planning to do calculations involving large data and kernels sets, you may need to allocate a larger amount of memory to run the program. To allocate, for example, 500 MB of memory, use the following command:

```
java -Xmx500m -jar InversionPipeline.jar
```

*Note:* the option `-Xmx` is nonstandard and may change according to the release installed on your computer.

## 1.2 Using the program

`InversionPipeline` is a program which calculates a series of inversions by automatically choosing models from a grid. Once the program is running, the user typically goes through the following steps:

1. read a set of frequencies (and optionally rotational splittings, although these are not currently used by `InversionPipeline`).
2. (optional) plot an echelle diagram of the above frequencies
3. load a set of inversion parameters for various inverted quantities
4. load a set of observational constraints
5. preview the models selected via the above observational constraints (to make sure a reasonable number of models is selected). Models are selected if they are within  $1\sigma$  of the observational constraints.
6. launch a series of inversions by automatically selecting models from one or more local or remote grids, using the above observational constraints

These operations can be done either automatically using a script file, or interactively using the different tabs within the program. These tabs are:

- **Frequency data:** this allows the user to load and edit observed pulsation frequencies (and optionally rotational splittings)
- **Echelle diagram:** this constructs an echelle diagram based on the observed frequencies
- **Inversion parameters:** this allows the user to set the types of inversion he or she would like to carry out, as well as relevant parameters such as the regularisation parameter
- **Launch page:** this allows the user to load/specify observational constraints, preview the selected models, and launch a series of inversions by automatically selecting reference models based on the above constraints
- **Physical constants:** this allows the user to modify physical constants (such as the gravitational constant) as well as astronomical constants (such as the solar mass)

The next few pages show a set of screen captures along with a brief description of the various buttons and options which appear in the different tabs. Also included are some screen captures of pop-up windows which appear, for instance, when using the interactive results or model legends. The actual appearance of these tabs and windows may vary from one platform to another depending on the Java installation. This section can therefore be viewed as a quick and easy guide for `InversionPipeline`. For more detailed information on the relevant file formats as well as some technical aspects, we refer the reader to the following sections.

### 1.3 Preliminary remarks

Some of the operations can take some time. For example, carrying out a series of inversions on a hundred or so reference models may take a couple of minutes. When the program is calculating, it is best just to wait and let it finish what it is doing.

InversionPipeline 1.0

Frequency data

Echelle diagram

Inversion parameters

Launch page

Physical constants

$l$	$\nu$ (in $\mu\text{Hz}$ )	$\sigma(\nu)$ (in $\mu\text{Hz}$ )	Rot. Splittings	$\sigma(\text{R.S.})$
0	2 847.218	0.300	***	***
0	3 019.975	0.300	***	***
0	3 191.840	0.300	***	***
0	3 362.618	0.300	***	***
0	3 533.846	0.300	***	***
0	3 705.705	0.300	***	***
0	3 878.039	0.300	***	***
0	4 050.947	0.300	***	***
0	4 223.775	0.300	***	***
0	4 396.860	0.300	***	***
0	4 570.395	0.300	***	***
1	2 928.097	0.300	***	***
1	3 100.339	0.300	***	***
1	3 271.838	0.300	***	***
1	3 443.007	0.300	***	***
1	3 614.436	0.300	***	***
1	3 786.891	0.300	***	***
1	3 959.635	0.300	***	***
1	4 132.552	0.300	***	***
1	4 305.784	0.300	***	***
1	4 479.050	0.300	***	***
1	4 652.857	0.300	***	***
2	3 003.152	0.300	***	***
2	3 175.519	0.300	***	***
2	3 346.788	0.300	***	***
2	3 518.465	0.300	***	***
2	3 690.636	0.300	***	***
2	3 863.397	0.300	***	***
2	4 036.708	0.300	***	***
2	4 209.894	0.300	***	***
2	4 383.385	0.300	***	***
2	4 557.233	0.300	***	***
2	4 731.312	0.300	***	***

Add row

Remove selected row(s)

Clear

Sort

Load file

Append file

Write file

Number of frequencies: 33

Number of R.S.: 0

add a row with frequency data

remove the selected rows

clear all frequency data

sort according to l value, than frequency

load frequency from a file (replace any data in table)

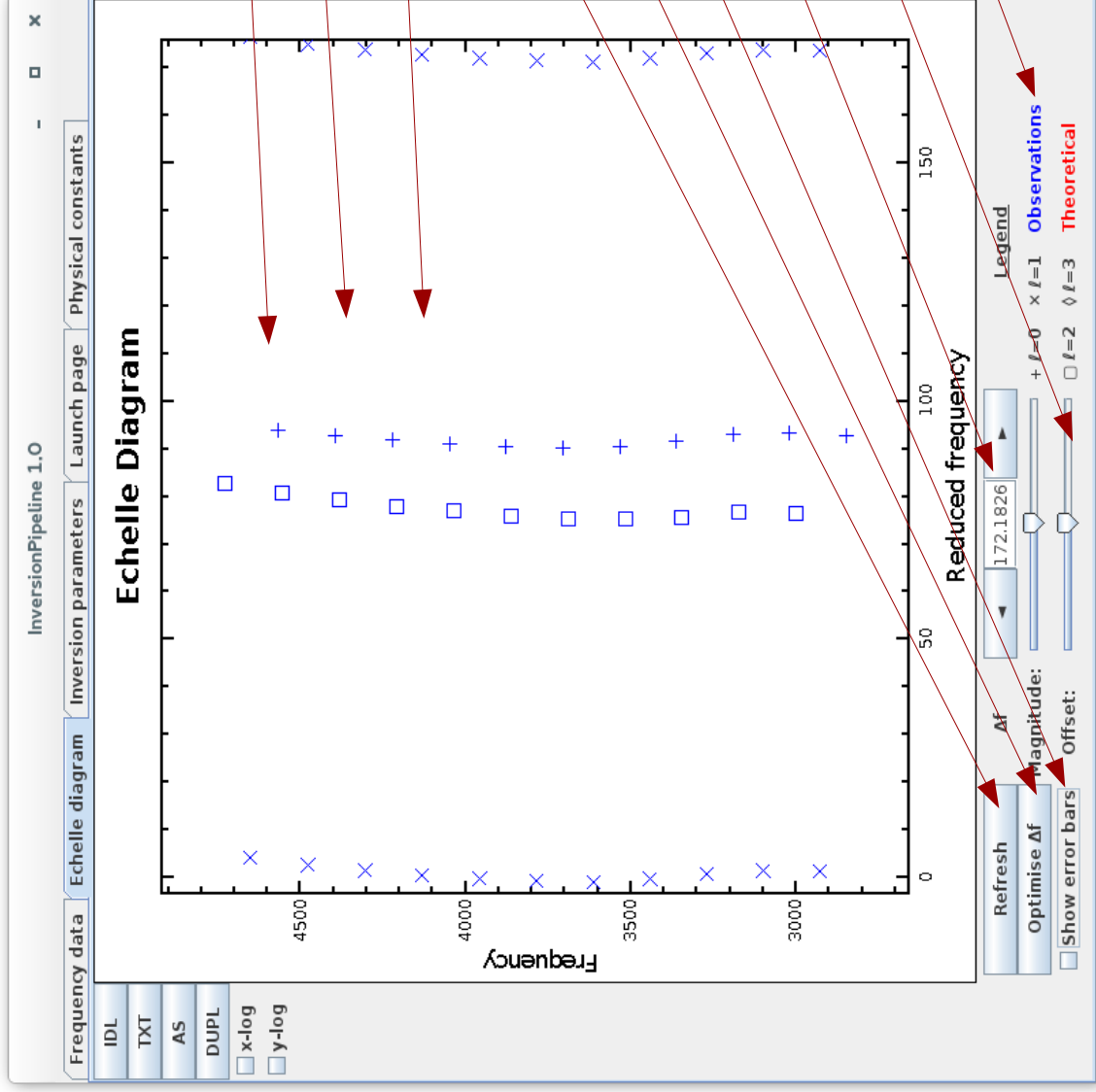
load frequency from a file (append information to end of table)

write a file based on data in table

number of frequencies in table

number of rotational splittings (not used for the moment)

each cell can be edited



interactive echelle diagram

left click and drag to zoom in

middle click and drag to  
move around in the plot

refresh echelle diagram  
(updates with new data)

find large frequency separation

show error bars on frequencies

modify large frequency  
separation

change offset in echelle  
diagram

legend which explains the  
different types of symbols

InversionPipeline 1.0

Frequency data

Echelle diagram

Inversion parameters

Launch page

Physical constants

Method	$n_{\text{surf}}$ or $n_{\text{min}}$	$\beta$ or $b$	$\theta$	Color
RHO.DNU	0	---	---	
RHO.KBCD	0	4.900	---	
RHO.SOLA	0	1.0000e-08	1.0000e-04	
RHO.SOLA	0	1.0000e-08	1.0000e-02	
RHO.SOLA	0	1.0000e-08	1.000	
RHO.SOLA	1	1.0000e-08	1.0000e-04	
RHO.SOLA	1	1.0000e-08	1.0000e-02	
RHO.SOLA	1	1.0000e-08	1.000	
AR1.NU	0	---	---	
AR1.DNU	0	---	---	
AR1.KBCD	0	4.900	---	
AR1.SOLA	0	1.0000e-06	1.0000e-04	
AR1.SOLA	0	1.0000e-06	0.100	
AR1.ISOLA	0	1.0000e-06	1.0000e-04	
AR1.ISOLA	0	1.0000e-06	0.100	
AR1.ISOLA2	0	1.0000e-06	1.0000e-04	
AR1.ISOLA2	0	1.0000e-06	0.100	
AR2.DNU	0	---	---	
AR2.KBCD	0	4.900	---	
AR2.SOLA	0	1.0000e-08	1.0000e-04	
AR2.SOLA	0	1.0000e-08	0.100	
AR2.ISOLA	0	1.0000e-08	1.0000e-04	
AR2.ISOLA	0	1.0000e-08	0.100	
SSEP.DNU	0	---	---	
SSEP.SOLA	0	1.0000e-04	1.0000e-06	
SSEP.SOLA	0	1.0000e-04	1.0000e-05	
SSEP.SOLA	0	1.0000e-04	1.0000e-04	
SSEP.SOLA	0	1.0000e-03	1.0000e-04	
SSEP.ISOLA	0	1.0000e-03	1.0000e-08	
SSEP.ISOLA	0	1.0000e-02	1.0000e-08	
SSEP.ISOLA2	0	1.0000e-03	1.0000e-08	
SSEP.ISOLA2	0	1.0000e-02	1.0000e-08	

LEGEND

Quantities

RHO	mean density
AR1	acoustic radius ( $c^2 + \rho$ )
AR2	acoustic radius ( $\rho + \Gamma_1$ )
SSEP	stellar age proxy

Methods

NU	direct fit to $\nu$
DNU	$\Delta\nu$ or $\delta\nu$
KBCD	Kjeldsen et al. (2008)
SOLA	SOLA: $K_{\text{avg}}$ & $K_{\text{cross}}$
ISOLA	SOLA: $JK_{\text{avg}}$ & $JK_{\text{cross}}$
ISOLA2	SOLA: $JK_{\text{avg}}$ & $JK_{\text{cross}}$

Parameters

$n_{\text{min}}$	lowest radial order
$b$	KBCD surface exponent
$n_{\text{surf}}$	# surface polynomials
$\beta$	tradeoff: $K_{\text{avg}}$ vs. $K_{\text{cross}}$
$\alpha$	tradeoff: $K$ vs. $\sigma$

Add row

Remove selected row(s)

Clear

Load file

Append file

Write file

add a row inversion parameters

remove the selected rows

clear all rows in table

load inversion parameters from a file (replace any data in table)

load inversion parameters from a file (append information to end of table)

write a file with inversion parameters

the cells in the table can be edited (including the colours)

a legend which explains the different types of inversions and relevant parameters

set of constraints, typically involving a central value and an error bar

check boxes used to select particular constraints

different stellar evolution phases can be selected (if these have been identified in the grid of models)

- load constraints from file

save constraints to file

use value of gravitational constant  
either from grid or from  
InversionPipeline (see Physical  
constants tab)

only use radial modes

quadratically add variational error  
on theoretical pulsation frequencies  
to observed errors

- select order pulsation calculations

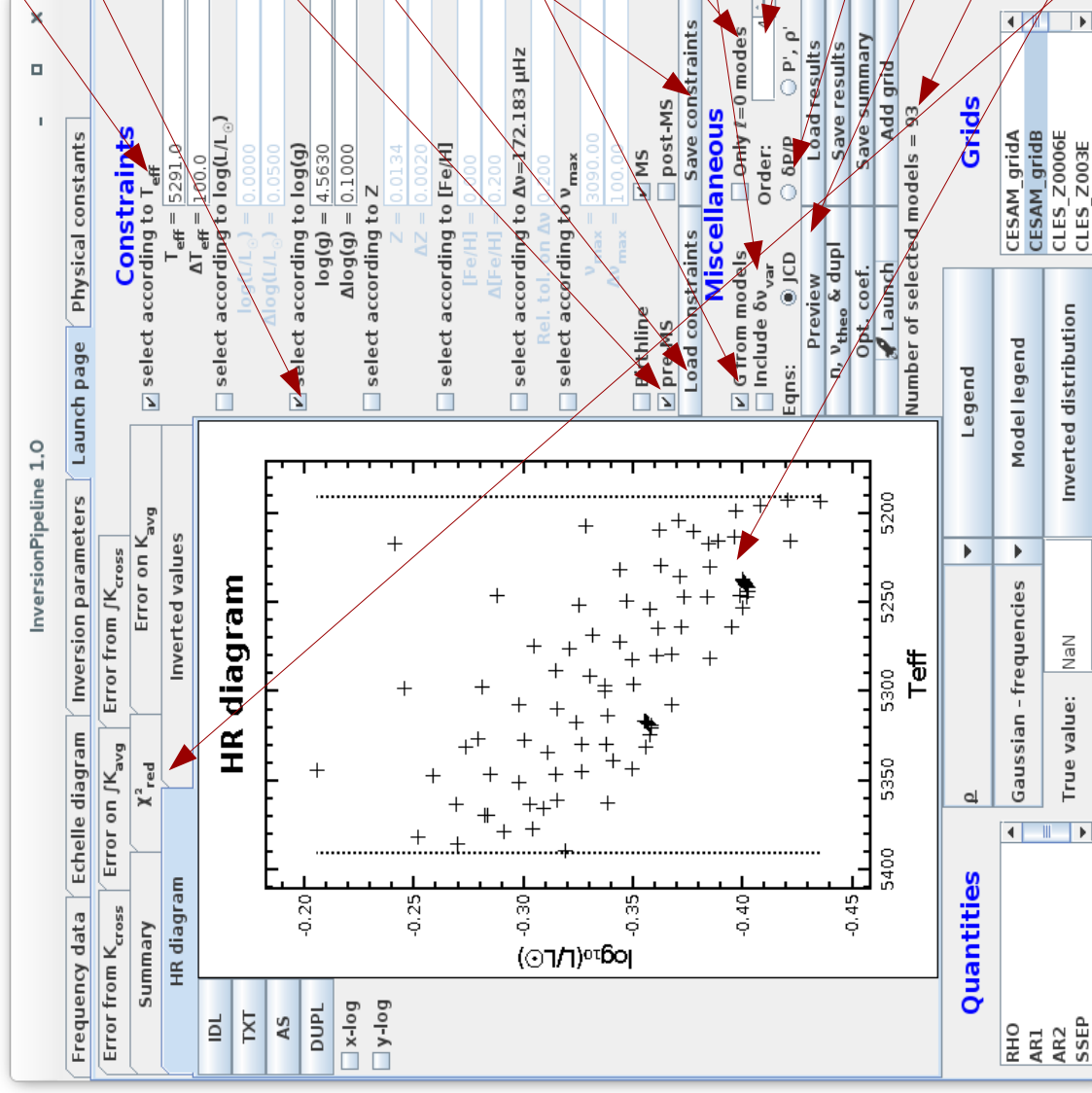
select type of equations in pulsation calculations

- preview selected models

- number of selected models

selected models in an HR diagram  
(right clicks yield model properties)

tabs with different results/diagnostics





InversionPipeline 1.0

Frequency data
Echelle diagram
Inversion parameters
Launch page
Physical constants

Summary

Error from  $K_{\text{cross}}$

Error on  $fK_{\text{avg}}$

$\chi^2_{\text{red}}$

HR diagram

Inverted values

☒ select according to  $T_{\text{eff}}$

$T_{\text{eff}} = 5291.0$   
 $\Delta T_{\text{eff}} = 100.0$

☐ select according to  $\log(L/L_{\odot})$

$\log(L/L_{\odot}) = 0.0000$   
 $\Delta \log(L/L_{\odot}) = 0.0500$

☒ select according to  $\log(g)$

$\log(g) = 4.5630$   
 $\Delta \log(g) = 0.1000$

☐ select according to  $Z$

$Z = 0.0134$   
 $\Delta Z = 0.0020$

☐ select according to  $[\text{Fe}/\text{H}]$

$[\text{Fe}/\text{H}] = 0.000$   
 $\Delta [\text{Fe}/\text{H}] = 0.200$

☐ select according to  $\Delta v = 172.183 \mu\text{Hz}$

Rel. tol. on  $\Delta v$  0.200

☐ select according to  $v_{\text{max}}$

$v_{\text{max}} = 3750.00$   
 $\Delta v_{\text{max}} = 700.00$

☐ Birthline

☒ pre-MS

☐ MS

☐ post-MS

Load constraints

Save constraints

Miscellaneous

☒ G from models

☐ Only  $T=0$  modes

☐ Include  $\delta v_{\text{var}}$

Order:

Eqns: ☒  $f$ CD ☐  $\delta p/p$  ☐  $p/\rho$  ☐  $\Delta f/f$

Preview

$n, v_{\text{thet}}$  & dupl

Load results

Save results

Opt-coor

Save summary

Launch

Add grid

Number of selected models = 93

Quantities

RHO

ARI

AR2

SSEP

Legend

Model legend

Inverted distribution

Grids

CESAM\_gridA

CESAM\_gridB

CLES\_Z0006E

CLES\_Z0036E

Frequency data

Error from  $K_{\text{cross}}$

Error on  $fK_{\text{avg}}$

$\chi^2_{\text{red}}$

HR diagram

Inverted values

☒ select according to  $T_{\text{eff}}$

$T_{\text{eff}} = 5291.0$   
 $\Delta T_{\text{eff}} = 100.0$

☐ select according to  $\log(L/L_{\odot})$

$\log(L/L_{\odot}) = 0.0000$   
 $\Delta \log(L/L_{\odot}) = 0.0500$

☒ select according to  $\log(g)$

$\log(g) = 4.5630$   
 $\Delta \log(g) = 0.1000$

☐ select according to  $Z$

$Z = 0.0134$   
 $\Delta Z = 0.0020$

☐ select according to  $[\text{Fe}/\text{H}]$

$[\text{Fe}/\text{H}] = 0.000$   
 $\Delta [\text{Fe}/\text{H}] = 0.200$

☐ select according to  $\Delta v = 172.183 \mu\text{Hz}$

Rel. tol. on  $\Delta v$  0.200

☐ select according to  $v_{\text{max}}$

$v_{\text{max}} = 3750.00$   
 $\Delta v_{\text{max}} = 700.00$

☐ Birthline

☒ pre-MS

☐ MS

☐ post-MS

Load constraints

Save constraints

Miscellaneous

☒ G from models

☐ Only  $T=0$  modes

☐ Include  $\delta v_{\text{var}}$

Order:

Eqns: ☒  $f$ CD ☐  $\delta p/p$  ☐  $p/\rho$  ☐  $\Delta f/f$

Preview

$n, v_{\text{thet}}$  & dupl

Load results

Save results

Opt-coor

Save summary

Launch

Add grid

Number of selected models = 93

Quantities

RHO

ARI

AR2

SSEP

Legend

Model legend

Inverted distribution

Grids

CESAM\_gridA

CESAM\_gridB

CLES\_Z0006E

CLES\_Z0036E

Frequency data

Error from  $K_{\text{cross}}$

Error on  $fK_{\text{avg}}$

$\chi^2_{\text{red}}$

HR diagram

Inverted values

☒ select according to  $T_{\text{eff}}$

$T_{\text{eff}} = 5291.0$   
 $\Delta T_{\text{eff}} = 100.0$

☐ select according to  $\log(L/L_{\odot})$

$\log(L/L_{\odot}) = 0.0000$   
 $\Delta \log(L/L_{\odot}) = 0.0500$

☒ select according to  $\log(g)$

$\log(g) = 4.5630$   
 $\Delta \log(g) = 0.1000$

☐ select according to  $Z$

$Z = 0.0134$   
 $\Delta Z = 0.0020$

☐ select according to  $[\text{Fe}/\text{H}]$

$[\text{Fe}/\text{H}] = 0.000$   
 $\Delta [\text{Fe}/\text{H}] = 0.200$

☐ select according to  $\Delta v = 172.183 \mu\text{Hz}$

Rel. tol. on  $\Delta v$  0.200

☐ select according to  $v_{\text{max}}$

$v_{\text{max}} = 3750.00$   
 $\Delta v_{\text{max}} = 700.00$

☐ Birthline

☒ pre-MS

☐ MS

☐ post-MS

Load constraints

Save constraints

Miscellaneous

☒ G from models

☐ Only  $T=0$  modes

☐ Include  $\delta v_{\text{var}}$

Order:

Eqns: ☒  $f$ CD ☐  $\delta p/p$  ☐  $p/\rho$  ☐  $\Delta f/f$

Preview

$n, v_{\text{thet}}$  & dupl

Load results

Save results

Opt-coor

Save summary

Launch

Add grid

Number of selected models = 93

Quantities

RHO

ARI

AR2

SSEP

Legend

Model legend

Inverted distribution

Grids

CESAM\_gridA

CESAM\_gridB

CLES\_Z0006E

CLES\_Z0036E

Frequency data

Error from  $K_{\text{cross}}$

Error on  $fK_{\text{avg}}$

$\chi^2_{\text{red}}$

HR diagram

Inverted values

☒ select according to  $T_{\text{eff}}$

$T_{\text{eff}} = 5291.0$   
 $\Delta T_{\text{eff}} = 100.0$

☐ select according to  $\log(L/L_{\odot})$

$\log(L/L_{\odot}) = 0.0000$   
 $\Delta \log(L/L_{\odot}) = 0.0500$

☒ select according to  $\log(g)$

$\log(g) = 4.5630$   
 $\Delta \log(g) = 0.1000$

☐ select according to  $Z$

$Z = 0.0134$   
 $\Delta Z = 0.0020$

☐ select according to  $[\text{Fe}/\text{H}]$

$[\text{Fe}/\text{H}] = 0.000$   
 $\Delta [\text{Fe}/\text{H}] = 0.200$

☐ select according to  $\Delta v = 172.183 \mu\text{Hz}$

Rel. tol. on  $\Delta v$  0.200

☒ select according to  $T_{\text{eff}}$   
 $T_{\text{eff}} = 5291.0$   
 $\Delta T_{\text{eff}} = 100.0$   
☐ select according to  $\log(t/L_c)$

☒ select according to  $\log(q)$ 

$\Delta \log(g) = 0.1000$

☐ select according to 7

 $\Delta Z = 0.0020$ 
$$\Delta[\text{Fe}/\text{H}] = 0.200$$
☐ select according to  $v_{\max}$ 

☐ Birthline ☒ MS ☐ MS

Load constraints	Save constraints
<b>Miscellaneous</b>	

☐ Include  $\delta v_{\text{var}}$  Order: 4

Prev. tw	Save results
$n, v_{\text{theo}}$ & dupl	Save results
Opt. coef.	Save summary
Launch	Add

Legend  CESAM grida 




Interactive Legend

RHO

AR1

AR2

SSEP

Method

n

$\beta$  or b

$\theta$

Color

<input checked="" type="checkbox"/> R-U	0			
<input checked="" type="checkbox"/> KBCD	0	4.90e+00		
<input checked="" type="checkbox"/> SOLA	0	1.00e-08	1.00e-04	
<input checked="" type="checkbox"/> SOLA	0	1.00e-08	1.00e-02	
<input checked="" type="checkbox"/> SOLA	0	1.00e-08	1.00e+00	
<input checked="" type="checkbox"/> SOLA	1	1.00e-08	1.00e-04	
<input checked="" type="checkbox"/> SOLA	1	1.00e-08	1.00e-02	
<input checked="" type="checkbox"/> SOLA	1	1.00e-08	1.00e+00	
				<div>All</div> <div>None</div>

interactive legend for inversion results

select inversion results to be displayed

parameters used in various inversions (cannot be edited)

edit colours used for various inversion results

select all inversion results

select none of the inversion results

different tabs for different inverted quantities

Interactive model legend

Model	$\rho$	$\chi^2_{red}$	Prob.	ID set	Dupl.
<input type="checkbox"/> Model 92	1.62e+00	1.3e+01	1.0e-05	1	1
<input type="checkbox"/> Model 93	1.65e+00	2.4e+01	3.2e-10	1	1
<input type="checkbox"/> Model 79	1.76e+00	1.6e+01	8.0e-07	1	1
<input type="checkbox"/> Model 5	1.77e+00	2.7e+02	7.7e-118	1	1
<input type="checkbox"/> Model 13	1.78e+00	2.2e+02	4.2e-97	1	1
<input type="checkbox"/> Model 24	1.85e+00	1.5e+02	3.0e-66	1	1
<input type="checkbox"/> Model 12	1.86e+00	1.8e+02	1.2e-79	1	1
<input type="checkbox"/> Model 4	1.87e+00	2.2e+02	1.4e-96	1	1
<input type="checkbox"/> Model 56	1.88e+00	9.8e+00	3.3e-04	1	1
<input type="checkbox"/> Model 23	1.92e+00	1.2e+02	3.9e-54	1	1
<input type="checkbox"/> Model 11	1.94e+00	1.5e+02	1.6e-65	1	1
<input type="checkbox"/> Model 3	1.96e+00	1.8e+02	1.5e-79	1	1
<input type="checkbox"/> Model 57	1.97e+00	1.5e+01	2.6e-06	1	1
<input type="checkbox"/> Model 22	1.99e+00	1.0e+02	8.8e-44	1	1
<input type="checkbox"/> Model 10	2.02e+00	1.2e+02	1.1e-53	1	1
<input type="checkbox"/> Model 40	2.02e+00	6.1e+00	1.3e-02	1	1
<input type="checkbox"/> Model 39	2.02e+00	6.5e+01	3.4e-28	1	1
<input type="checkbox"/> Model 2	2.04e+00	1.5e+02	9.4e-66	1	1
<input type="checkbox"/> Model 21	2.05e+00	8.1e+01	2.5e-35	1	1
<input type="checkbox"/> Model 38	2.08e+00	5.1e+01	3.8e-22	1	1
<input type="checkbox"/> Model 9	2.09e+00	1.0e+02	6.6e-44	1	1
<input type="checkbox"/> Model 1	2.12e+00	1.3e+02	2.5e-54	1	1
<input type="checkbox"/> Model 58	2.12e+00	1.2e+01	2.8e-05	1	1
<input type="checkbox"/> Model 20	2.12e+00	6.5e+01	3.3e-28	1	1
<input type="checkbox"/> Model 41	2.13e+00	8.9e+00	8.3e-04	1	1
<input type="checkbox"/> Model 37	2.14e+00	3.9e+01	4.6e-17	1	1
<input type="checkbox"/> Model 8	2.16e+00	8.2e+01	9.3e-36	1	1
<input type="checkbox"/> Model 19	2.18e+00	5.2e+01	2.2e-22	1	1
<input type="checkbox"/> Model 36	2.20e+00	3.0e+01	4.8e-13	1	1
<input type="checkbox"/> Model 55	2.20e+00	1.6e+01	1.1e-06	1	1

interactive legend for models

select individual models

relevant parameter values

reduced  $\chi^2$  (based on frequencies)

probability (or weight) used for combining inversion results

index such that models with the same value use the same mode identification (same radial orders and harmonic degrees)

indicates if modes are duplicate (i.e. if two observed modes are matched to the same theoretical mode)

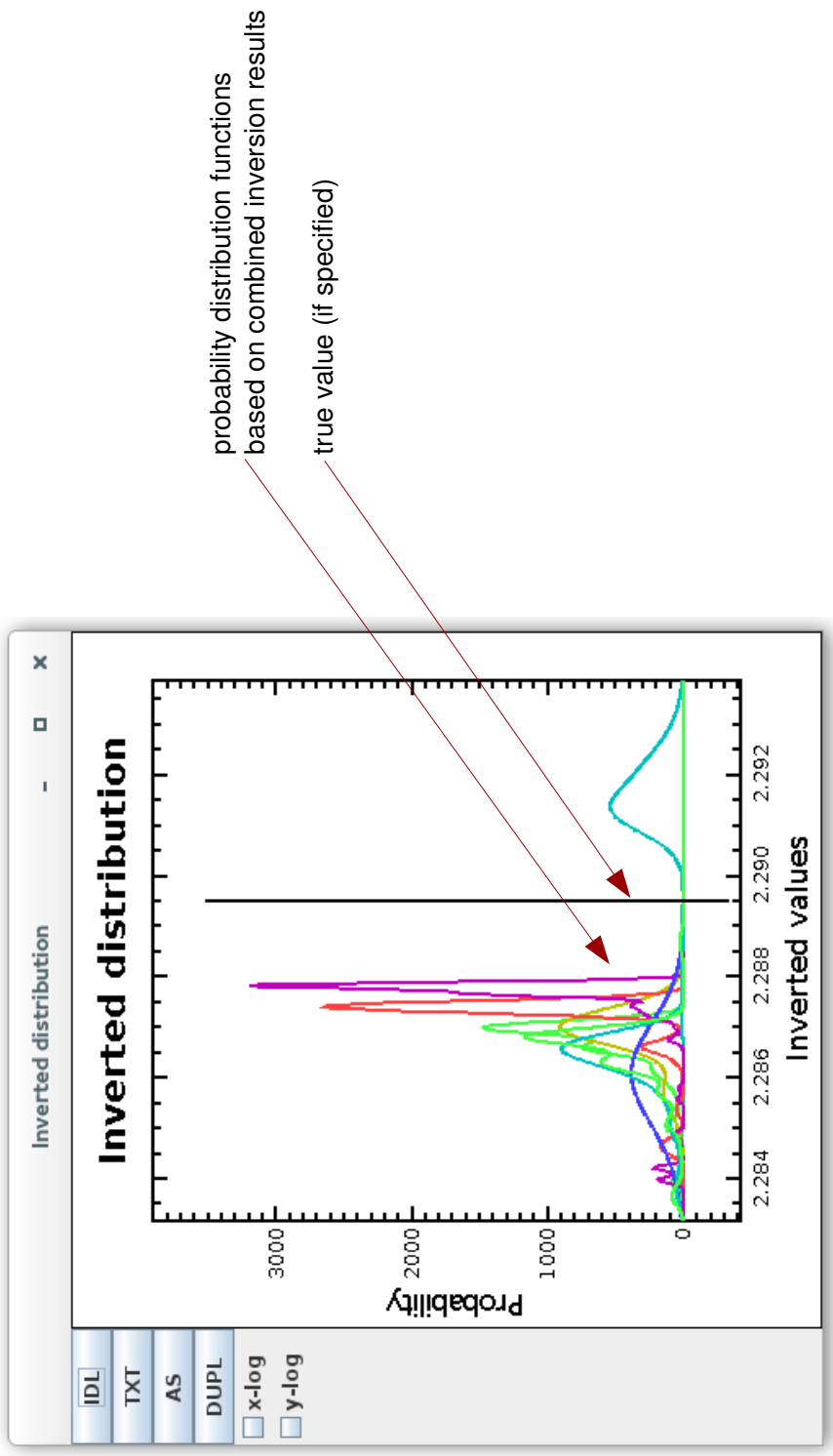
select all models

select none of the models

select the best model (in terms of the reduced  $\chi^2$  value)

select models with duplicate modes

select models with a particular set of mode identifications



InversionPipeline 1.0

Frequency data

Echelle diagram

Inversion parameters

Launch page

Physical constants

Gravitational constant,  $G$

Current value:  $6.6716823\text{e-}08 \text{ cm}^3 \cdot \text{g}^{-1} \cdot \text{s}^{-2}$

6.6716823e-08

Solar effective temperature,  $T_{\text{eff}, \odot}$

Current value: 5777.00 K

5777.00

$G = 6.6716823 \times 10^{-8} \text{ cgs (CoRoT/ESTA models)}$

$T_{\text{eff}, \odot} = 5777 \text{ K}$

$G = 6.67232 \times 10^{-8} \text{ cgs (Model S)}$

Solar large separation,  $\Delta\nu_{\odot}$

$G = 6.67259 \times 10^{-8} \text{ cgs (1986 CODATA)}$

Current value: 135.1  $\mu\text{Hz}$

135.1

$G = 6.67384 \times 10^{-8} \text{ cgs (2010 CODATA)}$

$\Delta\nu_{\odot} = 135.1 \text{ } \mu\text{Hz (Huber et al. 2009, 2011)}$

$G = 6.67428 \times 10^{-8} \text{ cgs (2006 CODATA)}$

Solar  $v_{\text{max}, \odot}$

Current value:  $5.670373\text{e-}05 \text{ g} \cdot \text{s}^{-2} \cdot \text{K}^{-4}$

Current value: 3090.0  $\mu\text{Hz}$

3090.0

Stefan-Boltzmann contant,  $\sigma$

$v_{\text{max}, \odot} = 3090.0 \text{ } \mu\text{Hz (Huber et al. 2009, 2011)}$

$5.670373\text{e-}05$

Solar mass,  $M_{\odot}$

$\sigma = 5.670373 \times 10^{-5} \text{ cgs (2010 CODATA)}$

Current value:  $1.98919\text{e+}33 \text{ g}$

1.98919e+33

Solar composition

Current values:  $X_{\odot}=0.7381$ ,  $Z_{\odot}=0.0134$

$M_{\odot} = 1.98919 \times 10^{33} \text{ g (CoRoT/ESTA models)}$

$X_{\odot} = 0.7381$

$Z_{\odot} = 0.0134$

$M_{\odot} = 1.989 \times 10^{33} \text{ g (Model S)}$

$X_{\odot}=0.7336$ ,  $Z_{\odot}=0.0179$  (Grevesse & Noels 1993)

$M_{\odot} = 1.98855 \times 10^{33} \text{ g (Cox 2000 + 2010 CODATA)}$

$X_{\odot}=0.7345$ ,  $Z_{\odot}=0.0169$  (Grevesse & Sauval 1998)

Solar radius,  $R_{\odot}$

$X_{\odot}=0.7392$ ,  $Z_{\odot}=0.0122$  (Asplund et al. 2005)

Current value:  $6.95990\text{e+}10 \text{ cm}$

$X_{\odot}=0.7381$ ,  $Z_{\odot}=0.0134$  (Asplund et al. 2009)

$6.95990\text{e+}10$

Derived solar parameters

$R_{\odot} = 6.9599 \times 10^{10} \text{ cm (CoRoT/ESTA models, Model S)}$

Solar mean density:  $\rho_{\odot} = 1.4085707 \text{ g} \cdot \text{cm}^{-3}$

$R_{\odot} = 6.95613 \times 10^{10} \text{ cm (Haberreiter et al. 2008)}$

Solar luminosity:  $L_{\odot} = 3.8444743\text{e+}33 \text{ g} \cdot \text{cm}^2 \cdot \text{s}^{-3}$

Save constants

page where different physical/  
astronomical constants can be  
adjusted

text field where a value can be  
entered manually

buttons with different standard  
values from the literature

derived values

save the constants in a configuration  
file called ~/.InversionPipeline

## 2 Technical aspects

The following subsections briefly describes some of the technical aspects of what calculations are carried out `InversionPipeline`. Given the existing documentation which can be found elsewhere, relevant references have been given rather than a full documentation in a number of cases.

### 2.1 Treatment of pulsation modes

The pulsation calculations is very similar between `InversionPipeline` and `InversionKit`, except that interface conditions (if the star has multiple domains) is not implemented in the current version `InversionPipeline`. Accordingly, we refer the reader to the documentation of `InversionKit` for the technical aspects of calculating pulsations modes, including the various choices of pulsation equations.

It is nonetheless important to point out some key aspects in the pulsation calculations. In `InversionPipeline`, pulsation modes are calculated automatically and cannot be loaded from a file. Given that `InversionPipeline` is trying to fit a set of observed frequencies, it will use the frequencies as input values (or shifts) or the shift-invert strategy for calculating modes – the user cannot specify another choice of modes to be calculated. Accordingly, the calculated modes will be the closest modes to the observed frequencies. However, if the reference model is a poor match to the actual star, then it is possible that the same theoretical mode will be the closest match to two observed modes, meaning a theoretical mode will be duplicated. `InversionPipeline` keeps track of duplicated modes, as well as the mode-identification obtained using each reference model. This information can be viewed using the “ $n$ ,  $\nu_{\text{theo}}$  & dupl” button on the Launch page. A more succinct version of this information can be seen in the interactive model menu where one column indicates whether there are duplicate modes, and a different column gives the “ID set”, an arbitrary index such that models with the same “ID set” use the same mode-identification and models with different “ID sets” use different mode-identifications.

### 2.2 Inversions

`InversionPipeline` is able to carry out SOLA type inversions for the mean density as described in Reese et al. (2012, A&A 539, A63), as well as inversions for the acoustic radius with two different choices of structural kernels, and inversions for an age indicator based on the small frequency separation, as described in Buldgen et al. (2015, A&A 574, A42). In addition, it is possible to obtain estimates of the mean density as well as the acoustic radius using an approach derived from Kjeldsen et al. (2008, ApJL 683, 175). Average large frequency separations have been implemented for both of these quantities, as well as the small frequency separation in the case of the age indicator. Finally, it is also possible to apply a frequency scaling law to obtain the mean density and the acoustic radius, based on the minimisation of the following cost function:

$$J = \sum_l \frac{(s\nu_l^{\text{ref}} - \nu_l^{\text{obs}})^2}{\sigma^2} \quad (1)$$

where  $s$  is a scale factor,  $\nu$  a frequency,  $\ell$  shorthand for  $(n, \ell)$ , and  $\sigma$  the observational error bar on  $\nu$ . Hence, this scale factor can be used to correct the mean density of the reference model and/or its acoustic radius so as to reproduce those of the observed star

(applying this approach for the acoustic radius seems somewhat questionable and a simple large frequency separation seems more appropriate).

## 2.3 Combining inversion results from multiple reference models

For a given inversion method, a set of  $n$  inversion results,  $\bar{A}_i$ , and associated uncertainties,  $\sigma_{A_i}$ , deriving from the uncertainties on the frequencies, will be obtained. These correspond to  $n$  probability distribution functions which `InversionPipeline` subsequently combines into a single probability distribution function via a weighted sum:

$$f(A) = \sum_{i=1}^n p_i f_i(A) \quad (2)$$

where  $p_i$  are the different weights,  $f_i$  the individual probability distribution functions and  $f$ , the final distribution. The weights involved can be thought of as a probability for each model. Accordingly they need to be positive and add up to 1, which ensures that  $f$  is positive and has an integral equal to 1.

The characteristics of this combined probability distribution function are easy to derive, and lead to an interesting interpretation:

**Average value** The average value of  $\bar{A}$ , is by definition:

$$\bar{A} = \int A f(A) dA = \int A \sum_{i=1}^n p_i f_i(A) dA = \sum_{i=1}^n p_i \int A f_i(A) dA = \sum_{i=1}^n p_i \bar{A}_i \quad (3)$$

where  $\bar{A}_i = \int A f_i(A) dA$  is the average  $A$  value for each model. Hence,  $\bar{A}$  is simply a weighted average of the  $\bar{A}_i$ .

**Variance** By definition, the variance of  $A$  is:

$$\begin{aligned} \text{Var}(A) &= \int (A - \bar{A})^2 f(A) dA = \sum_{i=1}^n p_i \int (A - \bar{A})^2 f_i(A) dA \\ &= \sum_{i=1}^n p_i \int (A - \bar{A}_i + \bar{A}_i - \bar{A})^2 f_i(A) dA \\ &= \sum_{i=1}^n p_i \int [(A - \bar{A}_i)^2 + 2(A - \bar{A}_i)(\bar{A}_i - \bar{A}) + (\bar{A}_i - \bar{A})^2] f_i(A) dA \\ &= \sum_{i=1}^n p_i [\sigma_{A_i}^2 + 2(\bar{A}_i - \bar{A})(\bar{A}_i - \bar{A}) + (\bar{A}_i - \bar{A})^2] \\ &= \sum_{i=1}^n p_i [\sigma_{A_i}^2 + (\bar{A}_i - \bar{A})^2]. \end{aligned} \quad (4)$$

Hence, the variance is split up into two parts which add up quadratically:

$$\sum_{i=1}^n p_i \sigma_{A_i}^2 = \begin{cases} \text{the square of the uncertainties which} \\ \text{propagate from the error bars on} \\ \text{the observed frequencies} \end{cases} \quad (5)$$

$$\begin{aligned} \sum_{i=1}^n p_i (\bar{A}_i - \bar{A})^2 &= \left( \sum_{i=1}^n p_i \bar{A}_i^2 \right) - \bar{A}^2 \\ &= \begin{cases} \text{the square of the uncertainties} \\ \text{coming from the scatter between} \\ \text{the different models} \end{cases} \end{aligned} \quad (6)$$

### 2.3.1 Different strategies for obtaining weights

InversionPipeline offers 3 different strategies for determining the weights  $p_i$ .

1. **Uniform:** this strategy consists simply in giving the same weight to each model, *i.e.*  $1/n$ .
2. **Gaussian – frequencies:** this strategy first involves calculating a *reduced*  $\chi^2$  as follows:

$$\chi_{\text{seism}}^2 = \frac{1}{K} \sum_{k=1}^K \left( \frac{\nu_k^{\text{theo.}} - \nu_k^{\text{obs.}}}{\sigma_{\nu_k}} \right)^2 \quad (7)$$

where  $K$  is the number of *non-duplicate* modes,  $\nu_k^{\text{theo.}}$  the theoretical frequency,  $\nu_k^{\text{obs.}}$  the observed frequency, and  $\sigma_{\nu_k}$  associated error bar. The probability is then calculated as follows:  $p_i = A \exp(-\chi_{\text{seism}}^2/2)$ , where  $A$  is a normalisation constant set so that  $\sum_{i=1}^n p_i = 1$ .

3. **Gaussian – constraints:** this strategy first involves calculating an *un-reduced*  $\chi^2$  on the constraints (including the large frequency separation and  $\nu_{\text{max}}$  if specified). The probability is then calculated as follows:  $p_i = A \exp(-\chi_{\text{constraints}}^2/2)$ , where  $A$  is a normalisation constant set so that  $\sum_{i=1}^n p_i = 1$ .

## 3 File formats

### 3.1 Script files

The script file takes on the following format:

- On a given line, anything following a “#” is treated like a comment and ignored.
- Each line starts with a case-insensitive “command”. These commands come in several types, based on the number and type of arguments which follow. The commands will be executed sequentially in the order they appear in the file.
- The commands “FREQUENCIES”, “PARAMETERS”, “CONSTRAINTS”, and “RESULTS” allow the user to specify files which correspond to the set of observed frequencies, inversion parameters, observational constraints, and results, respectively. These commands take two arguments: the first is either “IN” (which specifies



that the file will be used as an input) or “OUT” (meaning `InversionPipeline` will write to the file), and the second is the filename.

- The command “SUMMARY” can be used to write a summary of the results to the specified filename. It only takes one argument, the name of the file to which to write the result summary.
- The command “PROBABILITIES” is followed by one argument which specifies the weighting on reference models when combining their inversion results. The three options are: “UNIFORM” (all reference models have the same weight), “FREQUENCIES” (apply an  $\exp(-\chi_{\text{seism}}^2/2)$  weighting, where  $\chi_{\text{seism}}^2$  is a *reduced*  $\chi^2$  value based on the individual frequencies), and “CONSTRAINTS” (apply an  $\exp(-\chi_{\text{constraints}}^2/2)$  weighting, where  $\chi_{\text{constraints}}^2$  is an *un-reduced*  $\chi^2$  value based on the various constraints, most of which are classical).
- The command “LOAD\_GRID” allows the user to load an additional grid of models. This command only takes one argument, the name of the file with the grid of models.
- The command “LAUNCH” takes no arguments, and launches the inversions
- The command “NOGUI” takes no arguments and prevents the Graphical User Interface (GUI) from being launched. It makes no difference where this line is located in the script file.

The following shows an example of a script file.

```
# This is a comment ...
NOGUI
FREQUENCIES IN Reese2012/freq_Aprime
PARAMETERS IN Reese2012/inversion_parameters
CONSTRAINTS IN Reese2012/constraints
PROBABILITIES FREQUENCIES
LOAD_GRID ../../list # a comment can start in the middle of a line
SELECT_GRID CESAM_gridB CLES_Z002E Interpolated_models
LAUNCH
RESULTS OUT results_Aprime
SUMMARY summary_Aprime
```

### 3.2 Observed frequencies and rotational splittings

`InversionPipeline` reads the observed frequencies from which it calculates the frequency shifts. `InversionPipeline` may also read rotational splittings if included in the same file, although these are not used for the moment (the original code was extracted from `InversionKit` where rotational splittings are used, and left in in case `InversionPipeline` evolves to handle inversions related to the rotation profile). This file should obey the following formatting rules:

- On a given line, anything following a “#” is treated like a comment and ignored.
- Lines with fewer than 3 entries are discarded and provoke a warning message.

- Lines with 3 or 4 entries should contain:
  - an integer entry with the harmonic degree (or order)  $\ell$
  - the observed frequency,  $\nu_{\text{obs}}$  (in  $\mu\text{Hz}$ )
  - the error bar on the observed frequency,  $\sigma\nu_{\text{obs}}$  (in  $\mu\text{Hz}$ )
  - if a fourth parameter or entry is present, it is discarded
- Lines with at least 5 entries should contain:
  - $\ell$ ,  $\nu_{\text{obs}}$ , and  $\sigma\nu_{\text{obs}}$  as the first three entries
  - the rotational splitting and its error bar as the next two entries
  - further entries are discarded
- For the floating point entries, non-numerical entries can be used instead, such as “null”, to indicate an unknown value (this will appear as a blank entry in the table)

The following shows an example of a file with frequency data:

#	1	frequency	error
0		2847.2175635	0.3
0		3019.9748235	0.3
0		3191.8396392	0.3
1		2928.0967091	0.3
1		3100.3389662	0.3
1		3271.8382334	0.3
2		3003.1517125	0.3
2		3175.5186238	0.3
2		3346.7880266	0.3

### 3.3 Inversion types and parameters

The file with the inversion types and parameters contains four columns:

1. This column contains the type of inversion. It will be a composed name consisting of two parts. The first part specifies the quantity being inverted (for example “RHO” for the mean density) and the second part specifies the type of “inversion” (for example “DNU” for a method based on the large frequency separation, or “SOLA” for SOLA inversions). A complete list may be found in the legend on the Inversion parameters tab. These two parts are then joined by a period, “.”. We note that “SSEP” (*i.e.* a quantity derived from the small separation) are not compatible with “NU” or “KBCD” type methods and that the “DNU” method corresponds to the small separation in this particular case. The pull down menus within the cells of the table in the Inversion parameters tab automatically take these limitations.
2. The second column is an integer which either represents a minimal radial order (in the case of “NU”, “DNU” and “KBCD” type “inversions”) or the number of surface terms (in “SOLA”, “ISOLA”, “ISOLA2” type inversions).

3. The third column contains a floating point number with either  $b$ , the exponent in the Kjeldsen et al. (2008, ApJ 683, L175) recipe (the “KBCD” method), or the  $\beta$  parameter in SOLA type inversions that regulates the trade-off between optimising the averaging kernel and minimising the cross-term kernel (the “SOLA”, “ISOLA”, and “ISOLA2” type inversions).
4. The fourth column contains a floating point number with  $\theta$ , the parameter that regulates between regularising the solution (by reducing the propagated errors) and optimising the averaging kernels (it applies to “SOLA”, “ISOLA”, and “ISOLA2” inversions).

Furthermore, anything that follows a hash, “#”, is treated as a comment and removed. There is no fifth column to specify the colour as that is automatically determined by `InversionPipeline` (we do note, however, that the user can modify this colour by editing cell or using the interactive results legend).

The following shows an example of a file with inversion types and parameters:

#	method	n	beta	theta
	RHO.DNU	0	null	null
	RHO.KBCD	0	4.900e+00	null
	RHO.SOLA	0	1.000e-08	1.000e-02
	RHO.SOLA	1	1.000e-08	1.000e-02
	AR1.NU	0	null	null
	AR1.DNU	0	null	null
	AR1.KBCD	0	4.900e+00	null
	AR1.SOLA	0	1.000e-06	1.000e-04
	AR1.ISOLA	0	1.000e-06	1.000e-04
	AR1.ISOLA2	0	1.000e-06	1.000e-04
	AR2.ISOLA	0	1.000e-08	1.000e-01
	SSEP.DNU	0	null	null
	SSEP.SOLA	0	1.000e-04	1.000e-06
	SSEP.ISOLA	0	1.000e-03	1.000e-08
	SSEP.ISOLA2	0	1.000e-02	1.000e-08

### 3.4 Constraints

The file with observational constraints follows the following format:

- On a given line, anything following a “#” is treated like a comment and ignored.
- Each line starts with a case-insensitive “command”. These commands come in several types, based on the number and type of arguments which follow. None of the lines are obligatory, and later lines will take precedence over previous lines if they are contradictory (for instance, if one tries to specify two different ranges for the effective temperature).
- The commands “TEFF”, “LOGL”, “LOGG”, “Z”, “FEH”, and “NUMAX” are followed by two real numbers. The first gives the value of the parameter, and the second the  $1\sigma$  error bar.

- The command “DNU” is followed by one real number which corresponds to the  $1\sigma$  error bar. The value of this parameter is not specified as it is deduced from the set of observed frequencies.
- The commands “MODELG” (*i.e.* should `InversionPipeline` use the value of  $G$  from the model rather than from the Constants page?), “RADIAL” (should `InversionPipeline` only use radial modes?), and “ERRORVAR” (should `InversionPipeline` include the theoretical errors on the frequencies, as deduced via the variational principle, by quadratically adding them to the error bars on the observed frequencies?) are followed by “TRUE” or “FALSE”. We note that the “RADIAL” argument may lead to some technical issues later on in running `InversionPipeline`, so an alternative (and easier) option is to only include radial modes in the set of observed frequencies.
- The command “EQUATIONS” specifies what system of equations should be used when calculating the pulsation modes of the various models. It is followed by one of the three following options: “LAGRANGE” (this uses the Lagrangian pressure perturbation), “EULER” (this uses the Eulerian pressure perturbation), and “JCD” (this uses the system of equations from ADIPLS with a few minor adaptations). The default option (*i.e.* if there are no lines beginning with “EQUATIONS”) is “JCD”.
- The command “ORDER” is used to specify the numerical order of the finite differences used to discretise the pulsation equations. It is followed by an even integer between 2 and 20.
- The command “STAGE” allows the user to specify the evolutionary stage(s). It is followed by zero to several case-insensitive arguments which can be any of the following: “BIRTH” (*i.e.* the birth-line), “PREMS” (the pre-main sequence), “MS” (the main sequence), “POSTMS” (the post-main sequence). Stellar models which do not fall into one of the categories specified by the list of arguments are rejected (we note that models with an unspecified evolutionary stage (*i.e.* “noStage”) are never rejected via this mechanism.
- The command “GRID” allows the user to specify which grids of models to use when searching for reference models. The arguments which follow correspond to the names of the selected grid.
- The command “PROBABILITIES” is followed by one argument which specifies the weighting on reference models when combining their inversion results. The three options are: “UNIFORM” (all reference models have the same weight), “FREQUENCIES” (apply an  $\exp(-\chi^2_{\text{seism}}/2)$  weighting, where  $\chi^2_{\text{seism}}$  is a *reduced*  $\chi^2$  value based on the individual frequencies), and “CONSTRAINTS” (apply an  $\exp(-\chi^2_{\text{constraints}}/2)$  weighting, where  $\chi^2_{\text{constraints}}$  is an *un-reduced*  $\chi^2$  value based on the various constraints, most of which are classical).

The following shows an example of a file with observational constraints, as well as specifications on the evolutionary status and the grid(s) of models to be used:

```
TEFF  5.291000000000e+03  1.000000000000e+02
LOGG  4.563000000000e+00  1.000000000000e-01
STAGE PREMS MS
GRID  CESAM_gridB
```

### 3.5 Result files

There are two types of result files: the main result files which contain all of the inversion, and a summary file with a restricted amount of information. The main result file follows a fairly complex format, and includes a succession of different types of results. `InversionPipeline` both writes such files and reads them for the purposes of visualisation. The summary file, in contrast, has a much simpler format, but doesn't contain enough information to be visualised using `InversionPipeline`. Accordingly, `InversionPipeline` only writes such files, but it doesn't read them. The first three lines of the summary file give the number of reference models, the number of methods or inversion, and the number of modes. This is followed by a set of lines which describe each method or inversion. The next (and last) section is a 5 column table with one line per method/inversion. These columns represent:

1. the index of the method/inversion
2. the average value
3. the  $1\sigma$  error bar from propagated observational errors (see Eq. 5)
4. the  $1\sigma$  error bar from the scatter between the results from different reference models (see Eq. 6)
5. the total  $1\sigma$  error bar

### 3.6 Stellar models

`InversionPipeline` accepts the following file formats for stellar models:

1. **AMDL**: FORTRAN binary files generated by ASTEC
2. **FAMDL**: text (or ascii) version of the AMDL files generated by ASTEC
3. **FGONG**: exchange format under the GONG model comparison scheme, and official output format from ASTEC
4. **CESAM**: .osc text files generated by CESAM (including CESAM2k)
5. **MOD**: FORTRAN binary file from CLES
6. **CLES**: text files from CLES
7. **LOSC**: a text file from LOSC but in which the model is embedded

The main difference with `InversionKit` is that the text version of the **MOD** format is not currently implemented.

`InversionPipeline` will attempt to distinguish between models generated by CESAM2k and earlier versions of CESAM by searching for “CESAM2k” in the header. If

need be, MODCONV can be used to convert models from one format to another. This tool is available at:

<http://www.astro.up.pt/corot/ntools/modconv/>

A description of the first four file formats can be found at:

[http://www.astro.up.pt/corot/ntools/docs/CoRoT\\_ESTA\\_Files.pdf](http://www.astro.up.pt/corot/ntools/docs/CoRoT_ESTA_Files.pdf)

and within the instructions to the ADIPLS pulsation code:

[http://www.phys.au.dk/~jcd/adipack.n/notes/adiab\\_prog.ps.gz](http://www.phys.au.dk/~jcd/adipack.n/notes/adiab_prog.ps.gz)

The last three file formats will be briefly described in the following sections. This will then be followed by a table which summarises the different structural variables available in the various formats, some useful formulas which relate these variables, and finally a brief description of the format of the file with a list of models for the HR diagram.

### 3.6.1 MOD

These contain a 5 line header made up of 80 characters in the binary version, followed by a line which contains either 3 or 4 parameters. These are:

- the number of grid points
- the stellar radius (in cm)
- the stellar mass (in g)
- if present, the gravitational constant (in cgs)

This is then followed by a list of entries with 5 or 6 values which contain the following quantities:

- the radial coordinate (in cm)
- the cumulative mass (in g)
- the pressure (in cgs)
- the density (in g/cm<sup>3</sup>)
- the  $\Gamma_1$  profile
- if present, an unidentified quantity closely related to the adiabatic discriminant (`InversionPipeline` does not make use of this column)

### 3.6.2 CLES format

These contain a header which ends with the expression “%%beginoscddata”. This is followed by a line which contains the following parameters:

- the stellar radius (in cm)

- the stellar mass (in g)
- the gravitational constant (in cgs)

This is followed by a line with the number of domains (in case there are double points),  $n_{\text{domains}}$ , as well as a section made up of  $n_{\text{domains}}$  lines which gives information on these domains. The following line contains the number of grid points. This is then followed by a section with 6 columns which contain the following quantities:

- the radial coordinate (in cm)
- the quantity  $m/r^3$  (in g/cm<sup>3</sup>)
- the pressure (in cgs)
- the density (in g/cm<sup>3</sup>)
- the  $\Gamma_1$  profile
- the quantity  $-\frac{A}{r^2}$

### 3.6.3 LOSC format

These contain a 5 line header which is skipped. This is followed by a line with the following parameters:

- the number of grid points
- the stellar radius (in cm)
- the stellar mass (in g)
- the gravitational constant (in g)

The next three lines contain a header with column names and a list of various quantities related to the pulsation mode contained with the file. This is followed by a section with many columns. The first six columns are:

- the radial coordinate normalised by the radius
- the quantity  $\eta = \frac{R^3 m}{M r^3}$
- the quantity  $\frac{R}{GM} \frac{P}{\rho}$
- a normalised density,  $\frac{4\pi R^3}{M} \rho$
- the  $\Gamma_1$  profile
- the quantity  $-\frac{A}{r^2}$

### 3.7 Grid of models

`InversionPipeline` is able to read in supplementary grids of models via the “Add grid” button on the Launch page. The file which contains the list of models obeys the following format:

- the first line is a header with 4 entries: the name of the grid, the value of the gravitational constant (in cgs), the format of models, and a URL-style prefix for the root folder with the grid of models. This allows the user to use either local files (by starting the path with `file:///`) or remote files (beginning with `http://`).
- a succession of lines (one per model) with the following information:
  1. the second half of the filename of the model (*i.e.* when combined with the prefix on the first line, this produces a complete URL-style path)
  2. the mass in g,
  3. the radius in cm,
  4. the luminosity in erg/s,
  5. the initial metallicity,
  6. the initial hydrogen content,
  7. the age in Myrs,
  8. the temperature in K,
  9. and the evolutionary stage (the options are case-insensitive and include “NOSTAGE” if the evolutionary is unknown, “BIRTH”, “PREMS”, “MS”, and “POSTMS”).

The following is an example of a file with a grid of models.

```
Interpolated_models 6.67428E-8 MOD file:///home/dreese/interpolated_models/  
Model0.mod 2.167e+33 7.778e+10 4.988e+33 0.048 0.609 3.160e+03 5832.4 noStage  
Model1.mod 2.159e+33 7.768e+10 4.937e+33 0.049 0.604 3.222e+03 5821.3 noStage  
Model2.mod 2.168e+33 7.779e+10 4.994e+33 0.047 0.610 3.167e+03 5833.8 noStage  
Model3.mod 2.149e+33 7.755e+10 4.870e+33 0.052 0.597 3.311e+03 5806.1 noStage
```

*Note:* in general it is better to have a higher number of significant digits in order to retain a higher numerical accuracy. Here, the numbers have been truncated so the example can fit more easily on the page.

### 3.8 GZIP compression

`InversionPipeline` can read “gzipped” text files. In order to determine whether or not a file is gzipped, `InversionPipeline` looks at the filename to see if it ends with “.gz”. The same rule also applies when saving a model from the HELAS website <http://www.astro.up.pt/helas/> onto the hard disk – choosing a filename which ends with “.gz” causes `InversionPipeline` to save a compressed file, whereas any other ending produces an uncompressed file.



## 4 Known bugs

Here is a list of known bugs. If you find any other, please let us know by sending us an email ([daniel.reese@obspm.fr](mailto:daniel.reese@obspm.fr)).

- excessive zooming on plots can produce irregular behaviour
- the pulsation calculations becomes unreliable beyond  $\ell = 20$  due to the fact that `InversionPipeline` uses scaled variables in the pulsation equations.

## 5 Copyright notices

Below is the copyright notice that goes with `InversionPipeline`.

Copyright (c) Daniel Reese, Gael Buldgen, 2015

This file is part of `InversionPipeline`.

`InversionPipeline` is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version.

`InversionPipeline` is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with `InversionPipeline`. If not, see <http://www.gnu.org/licenses/>.

### 5.1 The rocket icon on the “Launch” button

The rocket icon was downloaded from the following website: <https://icons8.com/web-app/for/all/rocket>

### 5.2 Source code for reading fortran binary files

The source code for reading fortran binary files comes from the following web-pages:  
<http://docjar.com/docs/api/org/fudaa/dodico/fortran/NativeBinaryInputStream.html>  
<http://docjar.com/docs/api/org/fudaa/dodico/fortran/NativeBinaryOutputStream.html>  
<http://docjar.com/docs/api/org/fudaa/dodico/fortran/FortranBinaryInputStream.html>  
<http://docjar.com/docs/api/org/fudaa/dodico/fortran/FortranBinaryOutputStream.html>  
and are covered by the GNU GPL2 License. They have been corrected and modified so as to meet the needs of `InversionPipeline`.

### 5.3 Supplementary notices

Some of the code comes from other sources. The corresponding copyright notices are reproduced below:

## Notice number 1

@(#)OptionPaneDemo.java 1.9 04/07/26

Copyright (c) 2004 Sun Microsystems, Inc. All Rights Reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- Redistribution of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- Redistribution in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

Neither the name of Sun Microsystems, Inc. or the names of contributors may be used to endorse or promote products derived from this software without specific prior written permission.

This software is provided "AS IS," without a warranty of any kind. ALL EXPRESS OR IMPLIED CONDITIONS, REPRESENTATIONS AND WARRANTIES, INCLUDING ANY IMPLIED WARRANTY OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT, ARE HEREBY EXCLUDED. SUN MICROSYSTEMS, INC. ("SUN") AND ITS LICENSORS SHALL NOT BE LIABLE FOR ANY DAMAGES SUFFERED BY LICENSEE AS A RESULT OF USING, MODIFYING OR DISTRIBUTING THIS SOFTWARE OR ITS DERIVATIVES. IN NO EVENT WILL SUN OR ITS LICENSORS BE LIABLE FOR ANY LOST REVENUE, PROFIT OR DATA, OR FOR DIRECT, INDIRECT, SPECIAL, CONSEQUENTIAL, INCIDENTAL OR PUNITIVE DAMAGES, HOWEVER CAUSED AND REGARDLESS OF THE THEORY OF LIABILITY, ARISING OUT OF THE USE OF OR INABILITY TO USE THIS SOFTWARE, EVEN IF SUN HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

You acknowledge that this software is not designed, licensed or intended for use in the design, construction, operation or maintenance of any nuclear facility.

## Notice number 2

Copyright (c) Ian F. Darwin, <http://www.darwinsys.com/>, 1996-2002.  
All rights reserved. Software written by Ian F. Darwin and others.  
\$Id: LICENSE,v 1.8 2004/02/09 03:33:38 ian Exp \$

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE AUTHOR AND CONTRIBUTORS ‘‘AS IS’’ AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Java, the Duke mascot, and all variants of Sun’s Java "steaming coffee cup" logo are trademarks of Sun Microsystems. Sun’s, and James Gosling’s, pioneering role in inventing and promulgating (and standardizing) the Java language and environment is gratefully acknowledged.

The pioneering role of Dennis Ritchie and Bjarne Stroustrup, of AT&T, for inventing predecessor languages C and C++ is also gratefully acknowledged.