

WORKING WITH SOLAR SYSTEM ELEMENTAL AND ISOTOPIC ABUNDANCES IN PYTHON.

Reto Trappitsch, Brandeis University, Department of Physics, 415 South St, Waltham, MA 02453, USA
(reto@brandeis.edu)

Motivation In recent years, python¹ has become an important programming language for data processing in the sciences. One clear example of this development is the prevalence of `astropy`,² a package that contains common tools required for astronomy and astrophysics with python. The two `astropy` papers [1] and [2] have at the time of this writing received 3833 and 1513 citations, respectively, as recorded in the ADS database.³

In cosmochemistry and astrophysics we are often faced with measurements and model calculations that need to be compared to the Solar System elemental and isotopic composition. Here we present a python package named `iniabu` (short for initial abundances) that can be used to query elemental and isotopic solar abundances. Currently implemented databases are Lodders et al. [3] (default) and Asplund et al. [4]. Furthermore, the NIST database⁴ is used to provide element and isotope masses. Abundances can be queried as linear number abundances (normalized to a silicon abundance of 10^6), logarithmic number abundances (normalized logarithmically to a hydrogen abundance of 12), and mass fractions (normalized to a total sum of one for all elements). Furthermore, `iniabu` allows the user to directly calculate element and isotope ratios, δ -values, internally normalized isotope ratios, and bracket notation as frequently used in astronomy.

The source code is released open source under the GNU Public License GPLv2 and is available for download from GitHub.⁵ Furthermore, the package is documented in detail on `readthedocs`⁶ and example usage files using Jupyter notebooks are available on the project's GitHub site as well.

Installation The currently available, stable version of the software is v1.0.0, which is available for download from the python packaging index PyPi.⁷ Alternatively, the most recent development version can also be directly installed from GitHub⁵ using `pip`. We are planning regular releases on PyPi as new features become available.

Currently, `iniabu` is compatible with python 3.6 and newer. The only required dependency is `numpy` [5].

Usage Here we briefly discuss some use cases that might be of interest for the cosmochemistry community. For more examples and a full discussion of all features, see the documentation.⁶ Note that subsequent examples depend on previous ones, e.g., importing the module is only shown in the first example.

For general usage, we recommend *importing* an instance of `iniabu` as following:

```
>>> from iniabu import ini
```

This will load the default database from Lodders et al. [3] with linear number abundances (abundance of silicon normalized to 10^6).

Various *package settings* are implemented as properties and can thus be queried and assigned in the same way as regular variables. For example, the set database and the set units can be queried in the following way:

```
>>> ini.database
'lodders09'
>>> ini.unit
'num_lin'
```

Elements and isotopes can be loaded into variables using the `ini.ele` and `ini.iso` proxy lists. The following is an example for helium:

```
>>> ele = ini.ele["He"]
>>> isos0 = ini.iso[["He-3", "He-4"]]
>>> isos1 = ini.iso["He"]
```

Here, the first line stores the element helium in a variable named `ele`. The second and third line are in fact identical and store all isotopes of helium in variables. Here, calling an isotope with an element name results in all isotopes of that specific element being returned. Element names must always be given using their abbreviation, isotopes using their abbreviation followed by a dash and the mass number of the isotope. The second line of above example shows that multiple entries can be queried simultaneously if passed as a list.

Having these variables defined, various *properties of elements and isotopes* can be queried. Some examples:

```
>>> ele.abu_solar
2511030000.0
>>> isos0.mass
array([3.01602932, 4.00260325])
```

The first line queries the solar abundance of helium. The second line queries the masses of the isotopes stored in

¹<https://python.org>

²<https://www.astropy.org>

³<https://ui.adsabs.harvard.edu/>

⁴<https://www.nist.gov/pml/atomic-weights-and-isotopic-compositions-relative-atomic-masses>

⁵<https://github.com/galactic-forensics/iniabu>

⁶<https://iniabu.readthedocs.io>

⁷<https://pypi.org/>

variable `isos0`. The output shows a feature of `iniabu`, namely that if more than one value is returned, we automatically return a `numpy array` instead of a regular python list. This enables the immediate usage of the output for subsequent mathematical operations.

δ -values for some data with respect to the Solar System initial abundances can be calculated. Let us assume we have two measurements of $^{29}\text{Si}/^{28}\text{Si}$ that are stored in a list called `msr`. The δ -values can then be calculated as following:

```
>>> msr = [0.06, 0.05]
>>> ini.iso_delta("Si-29", "Si-28",
                  msr)
array([182.05128205, -14.95726496])
```

Note here that δ -values are by default returned in permil. To change this behavior the keyword argument `delta_factor` can be used.

All functions of the code are extensively documented with docstrings. This means that if the user is running in `ipython` or from a Jupyter notebook, `help` can be obtained by simply adding a question mark after a command, e.g., `ini.iso_delta?`. This will bring up the docstring and describe briefly all arguments and keyword arguments. Full API documentation is also provided online.⁶

Database inconsistencies While linear number abundances are normalized to silicon equaling 10^6 as described above, this is only true within numerical precision for the Lodders et al. [3] database. For `iniabu` we used the data in table 10 of [3]. There, the silicon isotope abundances sum up to a total of 999,700. We did not re-normalize to 10^6 since within the available precision the two numbers are equal.

Furthermore, the abundance of ^{138}La in table 10 of [3] is given as 0.000. This number is reported with too few significant digits. Using the atom percentages for both lanthanum isotopes and the solar abundance for ^{139}La , we calculated a solar abundance for ^{138}La of 0.0004. This value is now used in `iniabu`.

Community The idea behind `iniabu` is to enable the whole community to easily interact with Solar System abundances in python. A small number of features, such as calculating δ -values, have so far been incorporated. Furthermore, various tests have been done in order to ensure the accuracy of the returned values. However, future development should also be inspired by the community.

The GitHub⁵ site allows for the possibility to raise issues. These issues are intended for bug reports, discussing problems that occur with the package, and to request new features and enhancements.

We also enabled a discussion board on GitHub.⁵ The idea is that users can discuss their experiences or provide examples for others to look at and study.

In the interest of fostering an open and welcoming environment we adopted a code of conduct for the `iniabu` project. The code of conduct can be found on the GitHub site.⁵

Contribution As an open source project we welcome contributions to `iniabu` from the community. The documentation⁶ contains a detailed developer's guide with instructions on how to get started. Planned contributions of features and enhancements for `iniabu` should, preferentially be discussed first by raising an issue.

To ensure maintainability, full coverage code testing using `pytest`⁸ is required. Test automation is provided with `nox`.⁹ The standard `nox` configuration will automatically be run using GitHub Actions when a pull request is submitted and merging into the main branch is blocked if these tests fail. This ensures that the package stays functional, backwards compatible, and simplifies maintainability. Automatic tests also run `xdoctest`¹⁰ to ensure that all docstring examples are functional.

To enhance code readability we enforce the `flake8`¹¹ style guide as well as automatic code formatting using `black`.¹² The latter will automatically take care of most style guide requirements and can, e.g., be used with the provided pre-commit hook.¹³

Acknowledgement We would like to thank Ondrea Clarkson, Falk Herwig, Maria Lugaro, and Marco Pignatari for helpful discussion on features and testing for v1.0.0 of `iniabu`.

References: [1] Astropy Collaboration et al. "Astropy: A community Python package for astronomy". In: *A&A* 558 (2013), A33. [2] Astropy Collaboration et al. "The Astropy Project: Building an Open-science Project and Status of the v2.0 Core Package". In: *AJ* 156.3 (2018), p. 123. [3] K. Lodders, H. Palme, and H.-P. Gail. "Abundances of the Elements in the Solar System". In: *Solar System*. Ed. by J. E. Trumper. Springer-Verlag Berlin Heidelberg, 2009. Chap. 4.4, pp. 560–630. [4] M. Asplund et al. "The Chemical Composition of the Sun". In: *Annual Review of Astronomy and Astrophysics* 47 (2009), pp. 481–522. [5] Charles R. Harris et al. "Array programming with NumPy". In: *Nature* 585 (2020), pp. 357–362.

⁸<https://docs.pytest.org>

⁹<https://nox.thea.codes/>

¹⁰<https://github.com/Erotemic/xdoctest>

¹¹<https://flake8.pycqa.org>

¹²<https://github.com/psf/black>

¹³<https://pre-commit.com/>