Project proposal for Google Summer of Code 2019

Expansion of the TARDIS Atomic Database

EZEQUIEL A. PÁSSARO

Contact information

Full name: Ezequiel Alejandro Pássaro
Institution: Universidad Nacional de La Plata
E-mail: ???
Phone number: ???
Telegram user: ???
Adress: ???. La Plata, AR.

1 Overview

TARDIS is a Monte Carlo radiative transfer code whose primary goal is the calculation of theoretical spectra for supernovae based on a number of input parameters, such as the supernova brightness and the abundances of the different chemical elements present in the ejecta. The main idea for this procedure is that by finding a close match between theoretical and observed spectra the parameters that actually describe the supernovae can be identified.

In addition to the input parameters, TARDIS requires data for describing the structure of atoms from different elements. This data is not measured by astronomers, but is most often gathered in a lab by atomic physicists. While TARDIS developers have compiled a small dataset for their initial work, they would like to incorporate the acclaimed CMFGEN atomic database into theirs.

The result of this work will not only be of great value for TARDIS, but also for many researchers who require atomic measurements.

2 Early involvement with the project

As part of the selection process I was ordered to perform several tasks required to show competency in Python programming, problem solving and communication skills. This is a brief description of the pre-application work I have done and its results.

2.1 Packaging TARDIS for the Anaconda Distribution

The Anaconda Distribution represents a simple and convenient way to distribute scientific software and make it available to a broader userbase. Despite having experience in delivering software with recipes for Anaconda's virtual environments, building a package represented a challenge to me. The steps required to accomplish this task were:

- Get familiar with TARDIS and its dependencies.
- Research about how the conda packaging system works.
- Succesfully build a local package.
- Make a pull request to the TARDIS repository (ref.: **#895**).
- Merged by Wolfgang Kerzendorf.

Then I was asked to make TARDIS available for multiple platforms via the *conda-forge* repository. These steps were neccesary in order perform this duty:

- Become familiar with *conda-forge*'s guidelines.
- Fork the *staged-recipes* repository and make a pull request (ref.: **#7934**).
- Set communication with the *conda-forge*'s team.

- Act as a bridge between *conda-forge*'s team and TARDIS developers.
- Adjust the package recipe to meet both sides demands.
- Add proper documentation to the TARDIS repository.
- Waiting for merge.

Once merged I will have fulfilled one of the project's goals for GSoC before it starts. Also a fluent communication with TARDIS developers let me know more about the project structure, goals and the way their contributors work.

2.2 Reading tabulated data from CMFGEN

My solution for TEP004 first objective can be found at: `https://git.io/fjvV0`

3 Main objectives

3.1 Parsers for the CMFGEN atomic database

Understanding the way this database was built is a fundamental step in order to correctly extract its data. At first sight it seems messy: there are several different datasets for some species and some primitive version control system was implemented. It's organized in folders with the following hierarchy:

```
./ATOMIC/SPECIE/IONIZATION_STAGE/DATE/
```

Also, every DATE folder is populated with up to six different file types:

1. OSC (energy levels and oscillator strengths).
2. F_TO_S (super-level groups).
3. COL (collisional data).
4. PHOT (photoionization cross-sections for each level).
5. DIE (low temperature dielectronic recombination lines).
6. AUTO (autoionization level probabilities).

... adding up to a total of 2.374 files.

Each file type has a different structure (most of them contains more than a single table) and needs its own parser. Deal with the subtle differences between files of the same type is a challenge in itself. This work will be done with the data analysis library *Pandas*.

3.2 Output module

Atomic data must be passed to TARDIS in a proper way. This work is currently done with the affiliated package *Carsus* written by Mikhail Mishin.

3.3 Simulations for different data sources

The last objective of this proposal is running TARDIS multiple times to understand how the new ingested data affects the synthetic spectra (e.g., transitions between energy levels should lead to new absorption lines).

4 Timeline

- **Bonding period:** Read and discuss the CMFGEN documentation and README files. Analyze carefully each file type in order to anticipate issues that could arise.
- **Week 1:** Write parser for OSC files.
- **Week 2:** Write parser for COL files.
- **Week 3:** Write parser for PHOT files.
- **Week 4:** Write unit tests.
- **Week 5:** Implement a continuous deployment pipeline with Travis-CI.
- **Week 6:** Documentation. Start discussion about the upcoming tasks.
- **Weeks 7-8:** Output module for TARDIS.
- **Week 9:** Implement auto-citation feature.
- **Week 10:** Documentation and testing.
- **Weeks 11-12:** Run simulations with different data sets and discuss results.

Wishlist:

- Implement energy levels diagram (Grotrian diagram).
- Deploy TARDIS on Google Colab and GCP.
- Enable the Dask framework for parallel execution.

5 About me

I'm a last year astronomy student and open source enthusiast, mostly interested in the fields of machine learning, high performance computing, cloud computing and software programming for astrophysics.

Time commitment with GSoC will be 40 hours per week and reachable 24/7 through the agreed channels for the entire three months period. My time zone is UTC-03:00.

5.1 Relevant skills

- Strong background in physics and astronomy.
- Two years programming experience in Python including numerical, scientific and machine learning libraries.
- Two years programming experience in C/C++.
- Four years programming experience in Fortran.
- Ten years of experience with GNU/Linux and open source software.
- Bash scripting.

5.2 Open source projects

- **Pulsar Playground:** *"Data visualization and machine learning tools for the HTRU2 dataset"*

  `https://github.com/epassaro/pulsar-playground`

- **Pugliese:** *"Python reduction pipeline for the GRACES*[1] *spectrograph"*

  `https://github.com/fedecampu/pugliese`

[1] **G**emini **R**emote **A**ccess to **C**FHT **ES**PaDOnS

5.3 Scientific projects I'm involved[2]

- **PuMA:** Pulsar Monitoring in Argentina
  Institution: Instituto Argentino de Radioastronomía (UNLP - CONICET)
  Volunteer programmer.
- **Grupo de Ciencia de Datos**
  Institution: Facultad de Ciencias Astronómicas y Geofísicas (UNLP)
  Member.

5.4 Certifications

- **DataCamp:** Data Scientist with Python
  Cert.: *#66084*
- **DataQuest:** Data Analyst in Python
  Cert.: *BX1C1RBG7TAOW1EHMS0B*

[2] No commitments with projects listed above while GSoC lasts.