Report of access of simulated supernovas data at LSST DC2

Laura Margarita Rodriguez Ortiz[1], Luis Henry Quiroga Nuñez[2]
Universidad de America, Bogota D.C, Colombia[1]
Florida Institute of Technology[2]

Corresponding author(s). E-mail(s): laura.rodriguez@estudiantes.uamerica.edu.co[1];
lquiroganunez@fit.edu[2]

Abstract

This report intends to provide guide about the access management in the LSST project to the public, and supply information about its structure. This is one of the differents challenges when one faces to access to the properly data in the Rubin Science Platform and LSST DESC Data Portal. We particularly focus on testing the photometry of the Vera Rubin Telescope for simulated supernovas data at LSST DC2.

Keywords: Access, LSST Project, challenges, photometry, simulated supernovas.

Introduction

Rubin Observatory's mission is to build a well-understood system that will produce an unprecedented astronomical data set for studies of the deep and dynamic universe. The aims is to produce the data widely accessible to a diverse community of scientists, and engage the public to explore the Universe with all the scientific staff .

Given the initiative to investigate the entire project that frames the Vera Rubin Observatory, RECA (The Network of Astronomy Students in Colombia) initiated the summer internship project in which undergraduate students of careers related to Astronomy contribute to the scientific development of this project.

This report aims to describe the steps taken to access LSST data, since it was not easily accessible, and therefore, it is important to keep in mind the difficulties presented. So, the scientific community has greater ease in accessing and continuing to generate science.

The Python notebook here generated handle data from some simulated supernovas from the LSST DESC 2 data portal. Specifically, when analyzing variable information, aspects as the wavelengths range for the LSST Telescope (ultraviolet to near-infrared, divided into spectral bands labeled u, g, r, i, z, and y) should be taken such. These features help to understand the model code that shows the bandpass data of each analyzed supernova.

Methodology

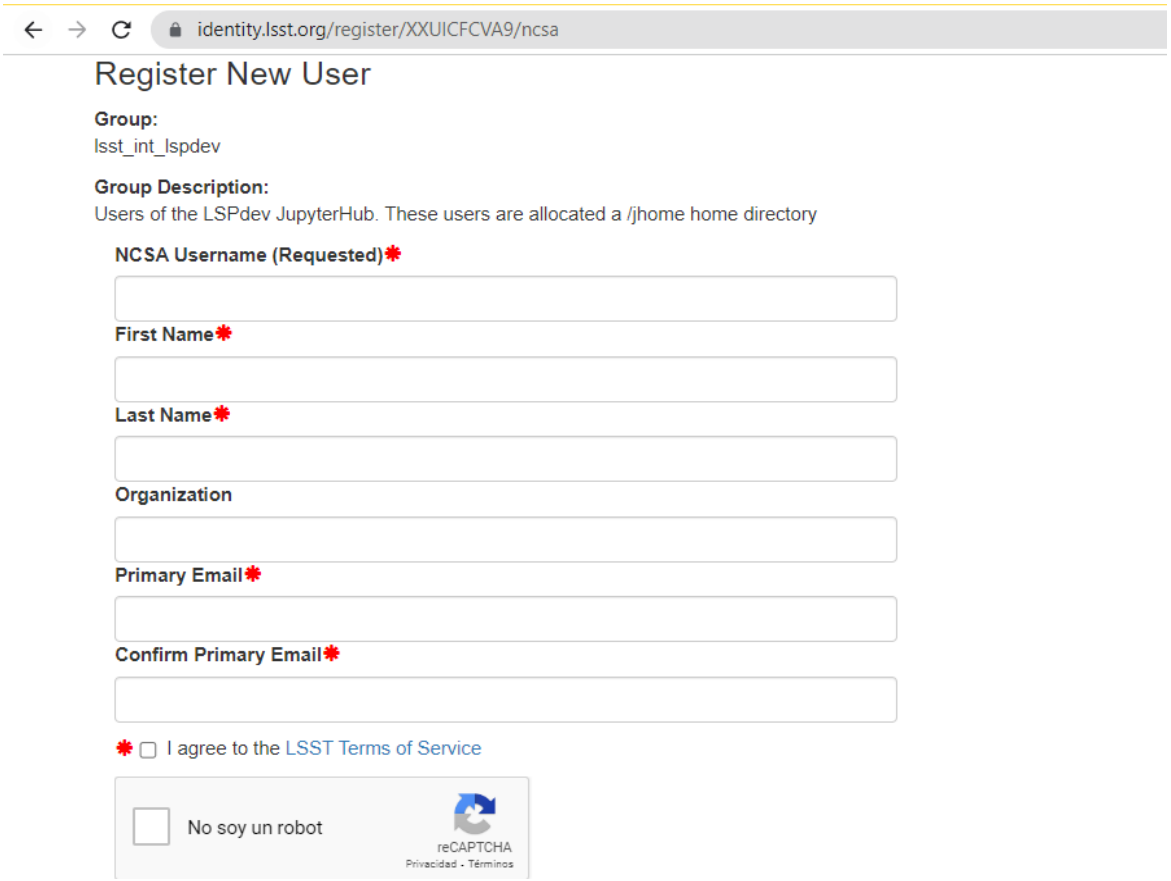
Hereby, we describe how we approach to the simulated data following teh current guides published in the website.

GalSim

Following the instructions of the repository https://github.com/GalSim-developers/GalSim, we proceeded to install the software on the Spyder with a Windows operating system but it was not possible. Neverthless, it was tested on a computer with an OS operating system running initially successfully. Afterwards, the user is asked to acess the LSST platform to manage the database directly.

LSST Platform

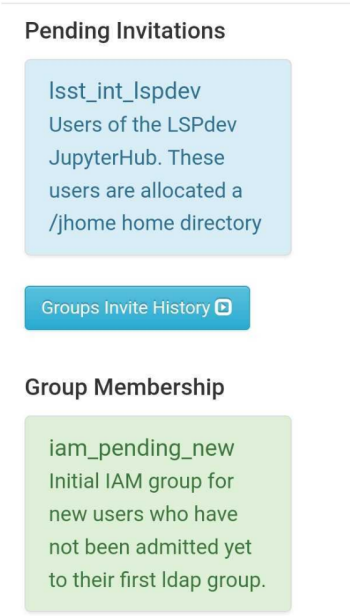
To access the LSST platform it is necessary to create an account at https://identity.lsst.org/login, registering with an account associated with the NCSA (The National Center for Supercomputing Applications), that requires completing this form:



Which will be the request to the join the group "lsst_int_lspdev" and then will give access to the data that is needed from the LSST. Once the form is filled out, you have to

wait a couple of days for said request to be approved, the follow-up of the its status is displayed on the registration platform:

During this waiting process, the VPN and the DUO application are installed on a cellphone to be able to log in to the platforms that involve the NCSA.



DUO

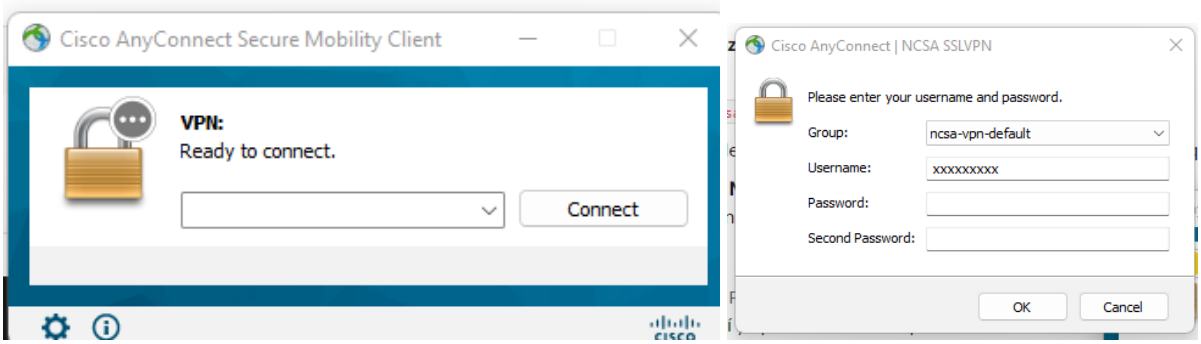
It is necessary to download the DUO authentication control application, this will allow access to the VPN and to the LSST pages and platform that will be used for data management. Once the registration data is entered, in one of the options described above, a "push" will arrive to the cellphone linked to the user.

It is important to mention that there are two options for entering the platform, one is through the "push" described in the previous paragraph and the other is with a code that is generated in the link where the account is created. According to the NCSA, these codes must be kept secure and confidential since they are changed periodically and are one-time use, as are all personal data, passwords and permissions to access the data.

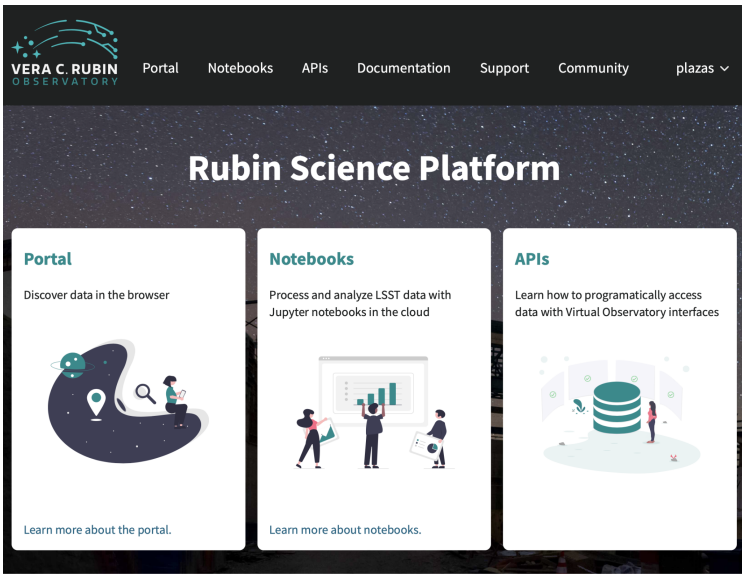
VPN

In the case of VPN, Cisco AnyConnect (https://its.gmu.edu/knowledge-base/how-to-install-cisco-anyconnect-on-a-windows-computer/) allows the connection to the platform that contains the data to be used (https://lsst-lsp-stable.ncsa.illinois.edu). It is important to emphasize that this last link only works through the characteristic user that the NCSA gives for the VPN.

Once the VPN is installed, you have to proceed to copy the connection link (https://sslvpn.ncsa.illinois.edu/) and click on “Connect” and this will open a new window where “group” is ncsa-default. A user is created and in password the word “push” must be used to activate the push notification in the DUO app. The user has to accept conditions from their cellphone in order to activate the VPN connection.



Once the request to access the data is accepted (and taking into account the steps to install the VPN and user management through the DUO app describe above), you can access the LSST platform at the link: https://lsst -lsp-stable.ncsa.illinois.edu.



On this page you can access the "Notebooks" option, where you can see a "Kubernetes Pod" in which you can choose "recommended" and "medium" for greater user convenience. Once inside, you can access Jupyter notebooks or terminal. In this tool you will find several package options and one of the packages you have access to is GalSim.

After exploring this option, the "Portal" was explored taking the audiovisual tool as a reference. We reported to the (https://community.lsst.org/) that the "Crash Course on LSST and Rubin Observatory", has a problem whe one is trying to access.

A response was obtained from staff members working on the platform, but it was concluded by a developer from there that it is better not to use the NCSA portal and only use Notebooks. So the next option to work with data was to try "API".

API - TOPCAT

TOPCAT (Tool for Operations on Catalogs And Tables) is an interactive graphical viewer and editor for tabular data. Its aim is to provide most of the facilities that astronomers need for analysis and manipulation of source catalogs and other tables, though it can be used for non-astronomical data as well (http://www.star.bris.ac.uk/~mbt/topcat/).

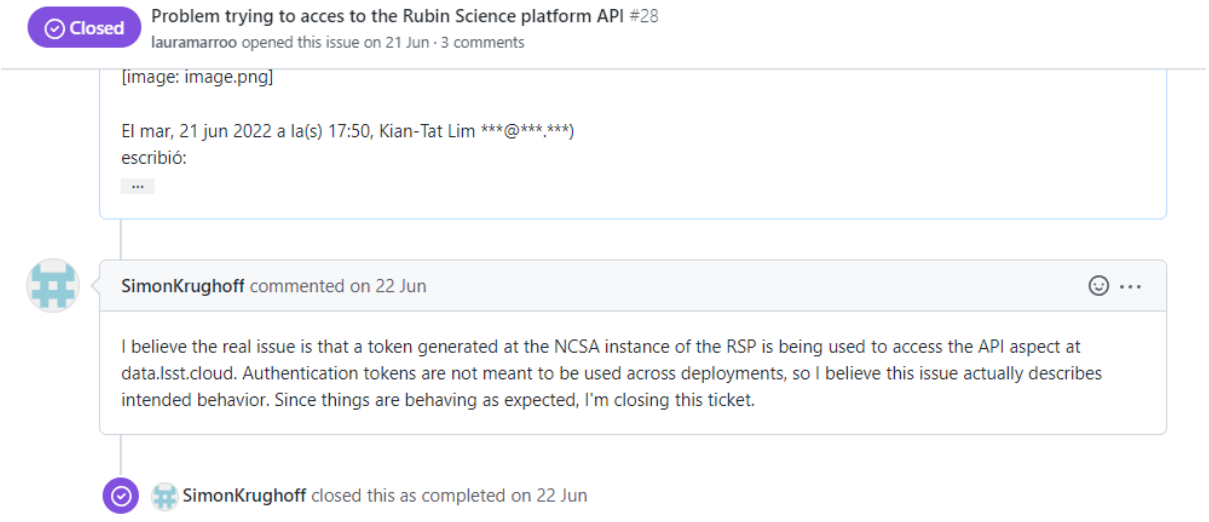
This graphical viewer is written in the Java language, so the proper option to install TOPCAT in a Windows operating system is starting to install Java and then, directly from the TOPCAT web in the "Downloads-Full Starjava Installation" section install TOPCAT..

When the editor is installed, the option "Query remote databases" brings us a multiple series of astronomical tables. We have to get the API from the LSST to use this information in TOPCAT. When we were trying to access the Rubin science platform using TAP through TOPCAT, we used the https://data.lsst.cloud/api/tap as the TAP endpoint. When TOPCAT tries to connect to the server, it requires a username and password. According to the https://lsst-lsp-stable.ncsa.illinois.edu/api-aspect, the username is x-oauth-basic and the password is the token. However, it keeps telling that the authentication failed.

In order to reproduce this error please follow:

1. Open TOPCAT
2. Click on Query remote database button
3. Enter the TAP URL: https://data.lsst.cloud/api/tap
4. Enter username (x-oauth-basic) and password (token)
5. See error: https://data.lsst.cloud/api/tap/tables error: java.io.IOException: Table resource access failure (401 Unauthorized)

So, we request help in @rubin-dp0 in GitHub, for repositories related to Rubin Observatory's Data Preview, and we obtained this answer:



GCRCatalogs

GCRCatalogs is a Python package that serves as a repository of various galaxy catalogs and sky catalogs for the LSST Dark Energy Science Collaboration (DESC). It provides a

unified user interface to access all catalogs by using the Generic Catalog Reader (GCR) base class (https://github.com/LSSTDESC/gcr-catalogs).

The LSST DESC Data Portal allows all users to transfer the DC2 Public Release data using Globus Connect. In this web, we had to create a GlobusID and set up the Globus Personal Connect in a local machine to download the files we will use. In the "FILE MANAGER" option, we had to search the catalogs:

- ➔ lsstdesc-public/dc2/run2.2i-dr6-v4/truth_sn/truth_sn_summary_v1-0-0.parquet
- ➔ lsstdesc-public/dc2/run2.2i-dr6-v4/truth_sn/truth_sn_variability_v1-0-0.parquet

Those files contain many information, and for this reason ,you should select only the data that you will use to avoid download extremely heavy files. Once the above has been done, we proceed to transfer those files to a local machine with the "Transfer" option and then, select the location in the local machine that we wanted.

Finally in Python, we loaded the supernova summary table, but Python didn't recognize the files that we called. So, we realize that we had to organize the files like the file path shown in the repository:

(lsstdesc-public/dc2/run2.2i-dr6-v4/truth_sn/truth_sn_summary_v1-0-0.parquet,
lsstdesc-public/dc2/run2.2i-dr6-v4/truth_sn/truth_sn_variability_v1-0-0.parquet).

Once those steps were finished, we could start working on the supernovas code.

Development: Supernova Code

For the analysis of simulated data, the code was created with the necessary steps to import the libraries and the data management, where the principal variables are the luminosity fluxes and the time characteristics of each supernova.

This code is in a code hosting platform (GitHub) with the aim to improve and to ease the scientific investigations that could come and also will be used for the real Rubin Observatory's data .

GitHub link: https://github.com/lauramarroo/LSST_Supernova_simulations.