Vera Rubin Observatory/Zooniverse

Deliverable: Software for transferring data from Rubin Science Platform to Zooniverse

1. Aim:

In order to facilitate the easiest possible set up for Rubin data rights holders who wish to run citizen science projects, the aim of this piece of work is to deliver code which can transfer data from the RSP to the Zooniverse citizen science platform. It is anticipated that rights holders would be able to transfer, in the first instance, small volumes of data (~300 images) to allow project design and demonstration, later transferring larger volumes after the project is formally approved.

1. Definitions

We adopt the Zooniverse terminology:

Classification Information provided via a task on the project website e.g. a marked feature in a light curve. A classification consists of a subject, task answer and user.

Subject An image or dataset (e.g. a time series) to be presented to the user.

Task A single interaction (e.g. a question which is asked of a user)

Workflow An ordered series of tasks

1. Approach and implementation

Assuming that users will work with notebooks within the Rubin Science Platform, we have assumed that the code we develop will need to run within the hosted Notebook Aspect at <http://data.lsst.cloud>. For more direct access to Zooniverse, users are directed to use the existing Panoptes Python Client (<https://github.com/zooniverse/panoptes-python-client>, <https://panoptes-python-client.readthedocs.io/en/latest/>)

Following discussions with the Rubin EPO team, it was decided that there was no need to incorporate enforcement of data rights by, for example, checking on the number of transfers initiated by a particular user. This mirrors behaviour elsewhere on the platform, which will allow users to download data for their own use and greatly simplifies the system which needs to be built.

The core functionality is contained within the rubin.citsci package, which can be installed via pip (but which will be preinstalled in the RSP). This code is contained in the following repository: <https://github.com/lsst-epo/citsci-notebook-core-pipeline>, and interacts with the Zooniverse Panoptes API, documented here: <https://github.com/zooniverse/panoptes>

Users are required to create a Zooniverse account, and a blank project, which they must set to public (either on the Zooniverse site at lab.zooniverse.org or using the Python client, above). They supply credentials to the RSP notebook, along with the project ‘slug’ (e.g zookeeper/galaxy-zoo) which identifies the Zooniverse project. (Note: It is anticipated that most users will start not with a blank project, but with a template which includes at least basic project information).

Users are expected to prepare data – for example, cutout images – within the RSP, using existing tools. Along with the images, Zooniverse expects a manifest file (format: csv) which contains metadata for the subject set as well as controlling many project options. Detailed descriptions of the Zooniverse manifest format are given here: <https://about.pfe-preview.zooniverse.org/lab-how-to>

We note in particular the utility of including metadata in columns beginning with #, which will not then be displayed to the user.

The tools allow for the automatic creation of a manifest file, but we have also allowed the RSP user to modify or make their own. The Zooniverse back end expects manifests to conform with RFC4180 <https://datatracker.ietf.org/doc/html/rfc4180.html> - essentially, comma separated files with empty columns where data is missing.

A zipped file can then be created and passed to Zooniverse via the cit\_sci\_pipeline.send\_image\_data command. This will create a new subject set associated with the project specified earlier in the process. At present, it is anticipated that the matching of a single subject set to each transfer will make the process easy to understand; commands exist, instead, for appending data to existing subject sets should the user prefer.

1. Further work

The design and implementation of a Rubin template project or projects, which at minimum would include observatory branding but which might also specify particular content or even specific workflows for, for example, handling light curve data. The design and development of these templates will take place as we build example projects.

We have also identified the need for more verbose error reporting from Zooniverse in the event that malformed or otherwise incompatible data is transferred. We will look to schedule this work in the next year.