The Journal of Open Source Software

GALAssify: A Python package for visually classifying astronomical objects

Manuel Alcázar-Laynez [1¶], **Andoni Jimenez** [1], **Salvador Duarte Puertas** [1,2,3], **Cristian Pozo González**[1], **Jesús Domínguez-Gómez** [1], **Guillermo Blázquez-Calero** [4], **Maria del Carmen Argudo-Fernández** [1,2], **Simon Verley** [1,2], **Daniel Espada** [1,2], **Estrella Florido** [1,2], **Isabel Pérez** [1,2], **Tomás Ruiz-Lara** [1,2], **Laura Sánchez-Menguiano** [1,2], and **Almudena Zurita** [1,2]

1 Departamento de Física Teórica y del Cosmos, Campus de Fuentenueva, Universidad de Granada, 18071 Granada, Spain **2** Instituto Carlos I de Física Teórica y Computacional, Spain **3** Département de Physique, de Génie Physique et d'Optique, Université Laval, et Centre de Recherche en Astrophysique du Québec (CRAQ), Québec, QC, G1V 0A6, Canada **4** Instituto de Astrofísica de Andalucía, CSIC, Glorieta de la Astronomía s/n, E-18008 Granada, Spain. ¶ Corresponding author

DOI: 10.xxxxxx/draft

Software

- Review
- Repository
- Archive

Editor: Open Journals

Reviewers:

- @openjournals

Submitted: 01 January 1970
Published: unpublished

In partnership with

AMERICAN ASTRONOMICAL SOCIETY

This article and software are linked with research article DOI 10.3847/xxxxx, published in the Astrophysical Journal.

Summary

The visual classification of astronomical objects requires the use of tools that are simple and easily adaptable to the requirements of the user. In this context, we present `GALAssify`, a graphical tool that allows the user to visually inspect and characterise properties of astronomical objects in a simple way. In addition, `GALAssify` allows the user to save the results of the visual classification into a file using a list of previously defined tags based on the user's interests. `GALAssify` is available on GitLab (https://gitlab.com/astrogal/GALAssify).

For many classification problems faced in astrophysics, a graphical interface greatly facilitates the job. In the present work we focus on the classification of galaxies to present a software that helps in a customised and simple way to do this classification. `GALAssify` has been developed in Python using PyQt5 libraries. A priori, it has been initially developed to tackle astrophysical problems but, due to its versatility, it could be easily adapted. For instance, this tool can be used to classify microscopy images from biological studies or be used in any other discipline.

We provide instructions for the installation, usage and basic examples of how to use `GALAssify`. In the `GALAssify`'s GitLab we show the use of `GALAssify` for visual classification of galaxy morphology. However, the user can extrapolate this visual study to any image.

Statement of need

Python is currently one of the most widely used programming languages in the scientific community, particularly in astrophysics. We have developed `GALAssify` with the aim of facilitating scientists and collaborators the task of visually classify the desired properties of astronomical objects. Additionally, the results can be easily shared between collaborators for analysis and comparisons and can be used for scientific reporting.

`GALAssify` interface was initially designed to perform the galaxy sample selection in the CAVITY (Calar Alto Void Integral-field Treasury surveY) project. CAVITY is a survey aimed to study galaxies in voids using Integral Field Unit data (Pérez et al. 2024, submitted).

Usage

`GALAssify` is a tool to classify images from a list of user-defined tags. As an example, in Figure 1, we show an image of a galaxy from Sloan Digital Sky Survey (SDSS, York et al., 2000) in order to classify their morphology visually. To do so, we provide a list of galaxies, the equatorial coordinates right ascension and declination (ra and dec, respectively), the path to the figures, and the relevant tags for the classification. There are three types of buttons that can be selected, *radiobutton* (only one of the options in the list can be selected), *checkbox* (the desired number of options can be selected) and *comments*. This particular usage has been widely used and tested within the CAVITY collaboration.

In the example shown in Figure 1 we considered a sample of galaxies assignation. The left panel of the figure shows the list of galaxies, which is a table with the following columns: assignation, the name of the galaxy, an icon indicating whether it has been processed, and the coordinates ra and dec for each galaxy. In the upper part of the right panel, the image of the selected galaxy is displayed. Optionally, this panel can also display the corresponding FITS image of the selected galaxy, specified by the user. In the case of galaxies observed in the SDSS, the algorithm allows the user to provide a path to the figure (if the image is located on their computer) or download it from the SDSS website given its coordinates. The lower part of the right panel is divided into three sections, where we show the different classifying options. The classification of each galaxy can be edited or reset at any time. In the left section we can choose between the main morphological types: elliptical, spiral, irregular, or other in case it does not fit clearly into these categories. In the central section the tags considered in order to do the classification are shown. In our example we considered the following tags: large, tiny, face-on, edge-on, star, calibration, recentre, duplicated, member, HII regions, and finally a pair of tags "Yes" and "No" to choose if we want to consider or exclude the galaxy for our study, respectively. In the lower right section, comments can be added to each of the galaxies. Finally, to save the selection, one can simply click the "Save and next button" or press the "enter" key. The entire classification is saved in a comma-separated values (CSV) file, easily readable with any text editor, spreadsheet program or database manager.

We also provide additional support tools to:

- Download images from SDSS catalogue.
- Create an instructions pdf document to guide the user through the graphical user interface (GUI).

JOSS
The Journal of Open Source Software

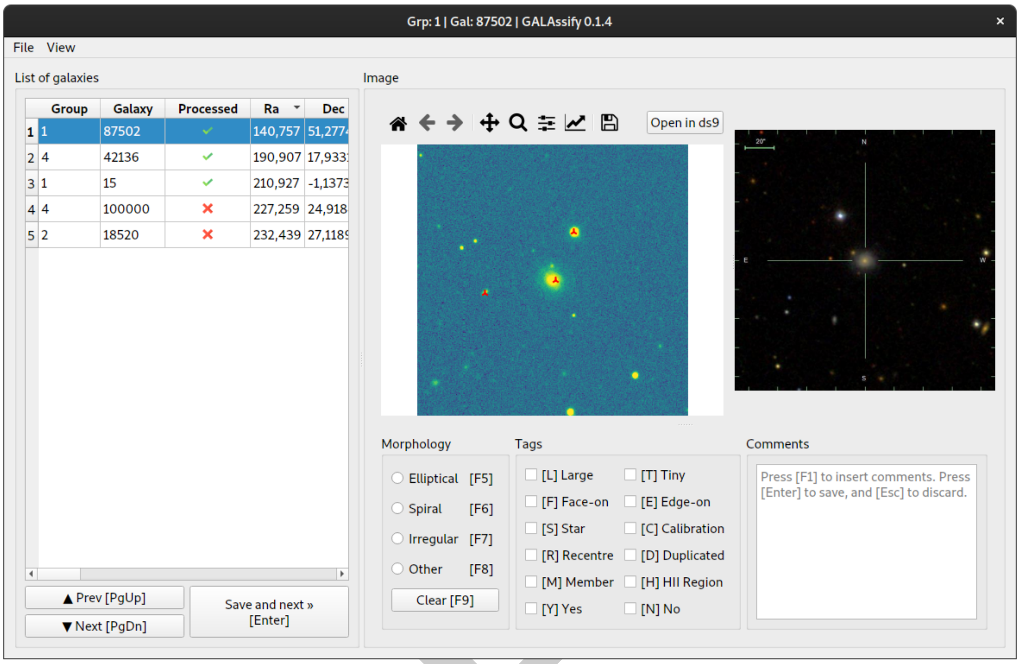

Figure 1: Interactive GUI: `GALAssify`.

Documentation

Package documentation is available on the `GALAssify`'s' GitLab page (https://gitlab.com/astrogal/GALAssify).

Software Citations

`GALAssify` should work with `python >= 3.9` and makes use of the following packages:

- Pyqt5 (https://www.riverbankcomputing.com/static/Docs/PyQt5/)
- Pandas (McKinney, 2010)
- NumPy (Harris et al., 2020; Walt et al., 2011)
- Matplotlib (Hunter, 2007)
- requests (https://requests.readthedocs.io/en/latest/)
- Pillow (https://python-pillow.org/)
- Astropy (Astropy Collaboration et al., 2013, 2018, 2022)
- pyds9 (https://github.com/ericmandel/pyds9)
- PyConsoleMenu (https://github.com/BaggerFast/PyConsoleMenu)

Only in case the user wants to display the images in fits format, it is necessary to have ds9 installed on the system.

The code is licensed under MIT License (MIT, https:/opensource.org/licenses/MIT) and is available on GitLab (https://gitlab.com/astrogal/GALAssify).

Acknowledgements

We acknowledge financial support by the research projects AYA2017-84897-P, PID2020-113689GB-I00, and PID2020-114414GB-I00, financed by MCIN/AEI/10.13039/501100011033, the project A-FQM-510-UGR20 financed from FEDER/Junta de Andalucía-Consejería de

Transforamción Económica, Industria, Conocimiento y Universidades/Proyecto and by the grants P20_00334 and FQM108, financed by the Junta de Andalucía (Spain).

M.A-F. acknowledges support from the Emergia program (EMERGIA20_38888) from Junta de Andaluc'ia.

G.B-C acknowledges financial support from grants PID2020-114461GB-I00 and CEX2021-001131-S, funded by MCIN/AEI/10.13039/501100011033, from Junta de Andalucía (Spain) grant P20-00880 (FEDER, EU) and from grant PRE2018-086111 funded by MCIN/AEI/10.13039/501100011033 and by 'ESF Investing in your future'.

SDP acknowledges financial support from Juan de la Cierva Formación fellowship (FJC2021-047523-I) financed by MCIN/AEI/10.13039/501100011033 and by the European Union 'NextGenerationEU'/PRTR, Ministerio de Economía y Competitividad under grants PID2019-107408GB-C44 and PID2020-113689GB-I00, from Junta de Andalucía Excellence Project P18-FR-2664, and SDP is grateful to the Natural Sciences and Engineering Research Council of Canada, the Fonds de Recherche du Québec, and the Canada Foundation for Innovation for funding.

TRL acknowledges support from Juan de la Cierva fellowship (IJC2020-043742-I), financed by MCIN/AEI/10.13039/501100011033.

References

Astropy Collaboration, Price-Whelan, A. M., Lim, P. L., Earl, N., Starkman, N., Bradley, L., Shupe, D. L., Patil, A. A., Corrales, L., Brasseur, C. E., Nöthe, M., Donath, A., Tollerud, E., Morris, B. M., Ginsburg, A., Vaher, E., Weaver, B. A., Tocknell, J., Jamieson, W., … Astropy Project Contributors. (2022). The Astropy Project: Sustaining and Growing a Community-oriented Open-source Project and the Latest Major Release (v5.0) of the Core Package. *935*(2), 167. https://doi.org/10.3847/1538-4357/ac7c74

Astropy Collaboration, Price-Whelan, A. M., Sipőcz, B. M., Günther, H. M., Lim, P. L., Crawford, S. M., Conseil, S., Shupe, D. L., Craig, M. W., Dencheva, N., Ginsburg, A., VanderPlas, J. T., Bradley, L. D., Pérez-Suárez, D., de Val-Borro, M., Aldcroft, T. L., Cruz, K. L., Robitaille, T. P., Tollerud, E. J., … Astropy Contributors. (2018). The Astropy Project: Building an Open-science Project and Status of the v2.0 Core Package. *156*(3), 123. https://doi.org/10.3847/1538-3881/aabc4f

Astropy Collaboration, Robitaille, T. P., Tollerud, E. J., Greenfield, P., Droettboom, M., Bray, E., Aldcroft, T., Davis, M., Ginsburg, A., Price-Whelan, A. M., Kerzendorf, W. E., Conley, A., Crighton, N., Barbary, K., Muna, D., Ferguson, H., Grollier, F., Parikh, M. M., Nair, P. H., … Streicher, O. (2013). Astropy: A community Python package for astronomy. *558*, A33. https://doi.org/10.1051/0004-6361/201322068

Harris, C. R., Millman, K. J., van der Walt, S. J., Gommers, R., Virtanen, P., Cournapeau, D., Wieser, E., Taylor, J., Berg, S., Smith, N. J., Kern, R., Picus, M., Hoyer, S., van Kerkwijk, M. H., Brett, M., Haldane, A., del Río, J. F., Wiebe, M., Peterson, P., … Oliphant, T. E. (2020). Array programming with NumPy. *Nature*, *585*(7825), 357–362. https://doi.org/10.1038/s41586-020-2649-2

Hunter, J. D. (2007). Matplotlib: A 2D graphics environment. *Computing In Science & Engineering*, *9*(3), 90–95.

McKinney, W. (2010). Data structures for statistical computing in python. In S. van der Walt & J. Millman (Eds.), *Proceedings of the 9th python in science conference* (pp. 51–56).

Walt, S. van der, Colbert, S. C., & Varoquaux, G. (2011). The NumPy array: A structure for efficient numerical computation. *Computing in Science & Engineering*, *13*(2), 22–30. https://doi.org/10.1109/MCSE.2011.37

York, D. G., Adelman, J., Anderson, J. E., Jr., Anderson, S. F., Annis, J., Bahcall, N. A., Bakken, J. A., Barkhouser, R., Bastian, S., Berman, E., Boroski, W. N., Bracker, S., Briegel, C., Briggs, J. W., Brinkmann, J., Brunner, R., Burles, S., Carey, L., Carr, M. A., … SDSS Collaboration. (2000). The Sloan Digital Sky Survey: Technical Summary. *120*, 1579–1587. https://doi.org/10.1086/301513