




PlatiPy: Processing Library and Analysis Toolkit for Medical Imaging in Python

Phillip Chlap ^{1,3,4} and Robert N. Finnegan ^{2,3,5}

1 South Western Sydney Clinical School, University of New South Wales, Australia 2 University of Sydney, Australia 3 Ingham Institute for Applied Medical Research, Australia 4 Liverpool Cancer Therapy Centre, Liverpool Hospital, Australia 5 Northern Sydney Cancer Centre, Royal North Shore Hospital, Australia

DOI: [10.21105/joss.05374](https://doi.org/10.21105/joss.05374)

Software

- [Review](#) 
- [Repository](#) 
- [Archive](#) 

Editor: Samuel Forbes 

Reviewers:

- [@agisga](#)
- [@tomelse](#)

Submitted: 28 February 2023

Published: 26 June 2023

License

Authors of papers retain copyright and release the work under a Creative Commons Attribution 4.0 International License ([CC BY 4.0](https://creativecommons.org/licenses/by/4.0/)).

Summary

PlatiPy offers a comprehensive suite of tools and utilities for conducting medical image analysis research utilising Python. These tools include functions for converting data between the clinical standard DICOM format and the research-friendly NIfTI format, capabilities for image registration and atlas-based segmentation, and efficient image visualisation tools to facilitate rapid development in research. Additionally, the library includes auto-segmentation models developed through various research projects enabling their streamlined deployment for use in future projects by the research community.

Statement of need

Python has gained significant popularity in the field of medical image analysis research in recent years, due in part to its open-source nature and the support of a large community of third-party libraries. Libraries such as SimpleITK ([Lowekamp et al., 2013](#); [Yaniv et al., 2018](#)), scikit-learn ([Pedregosa et al., 2011](#)) and pydicom ([Mason et al., 2022](#)) offer a wide range of functionality for developing medical image analysis tools. However, researchers often face the challenge of writing code to prepare data, creating wrapper functions for common procedures, and visualising images with scalar, vector or structure overlays throughout the analysis pipeline. PlatiPy addresses these challenges by providing many of these functions, eliminating the need for researchers to reinvent the wheel.

Other libraries and toolkits exist which provide complimentary functionality to PlatiPy. For example the Medical Imaging Interaction Toolkit (MITK) ([MITK Team, 2023](#)) provides a C++ library and a graphical user interface to perform a wide range of medical image analysis tasks including reading DICOM data, performing image registration and producing visualisations. The MIPAV (Medical Image Processing, Analysis, and Visualization) application ([National Institutes of Health, 2022](#)), OnkoDICOM ([Maher, Ashley, 2022](#)) and dicompyler ([Panchal & Keyes, 2010](#)) also provide a graphical user interface for medical image analysis and analysis of radiotherapy data. While these tools have proven useful in many research projects, they do not provide Python bindings which is often desired in modern medical image analysis research software projects. Medical Open Network for AI (MONAI) is an open-source framework for deep learning in healthcare imaging ([Cardoso et al., 2022](#)). This is built on top of the pytorch ecosystem and would typically be used to develop an entire project. In comparison, PlatiPy provides many utility type functions which can be integrated as little or as much as needed for a research project. The Medical Image Processing in Python library (medpy) does provide many of these utility type functions however doesn't provide functionality in the way of image visualisation or image registration available in PlatiPy ([Maier, Oskar, 2019](#)).

Additionally, tools resulting from medical image analysis research projects are not always easily reusable outside of the original project context. Researchers may provide a GitHub repository with their tool, but it may lack documentation and require installation of various dependencies, making the process difficult and time-consuming. By incorporating these tools into PlatiPy and adhering to good software design principles, they can be more easily reused in future research projects.

Image Visualiser

In medical image analysis, the ability to produce visualisations of images at various stages of the analysis pipeline, from initial inspection of the dataset to presentation of final results, is crucial for effective analysis. However, researchers often avoid visually inspecting their output due to the significant amount of boilerplate code required to produce these visualisations using the standard Python libraries such as matplotlib library directly. The Image Visualiser included in PlatiPy wraps this functionality within an easy-to-use class which can display the cross sections of 3D medical images as well as overlay contours, scalar volumes and deformation vector fields. Figures are produced using matplotlib allowing users to easily extend the visualisation functionality (Hunter, 2007). An example of such a visualisation is provided in Figure 1.

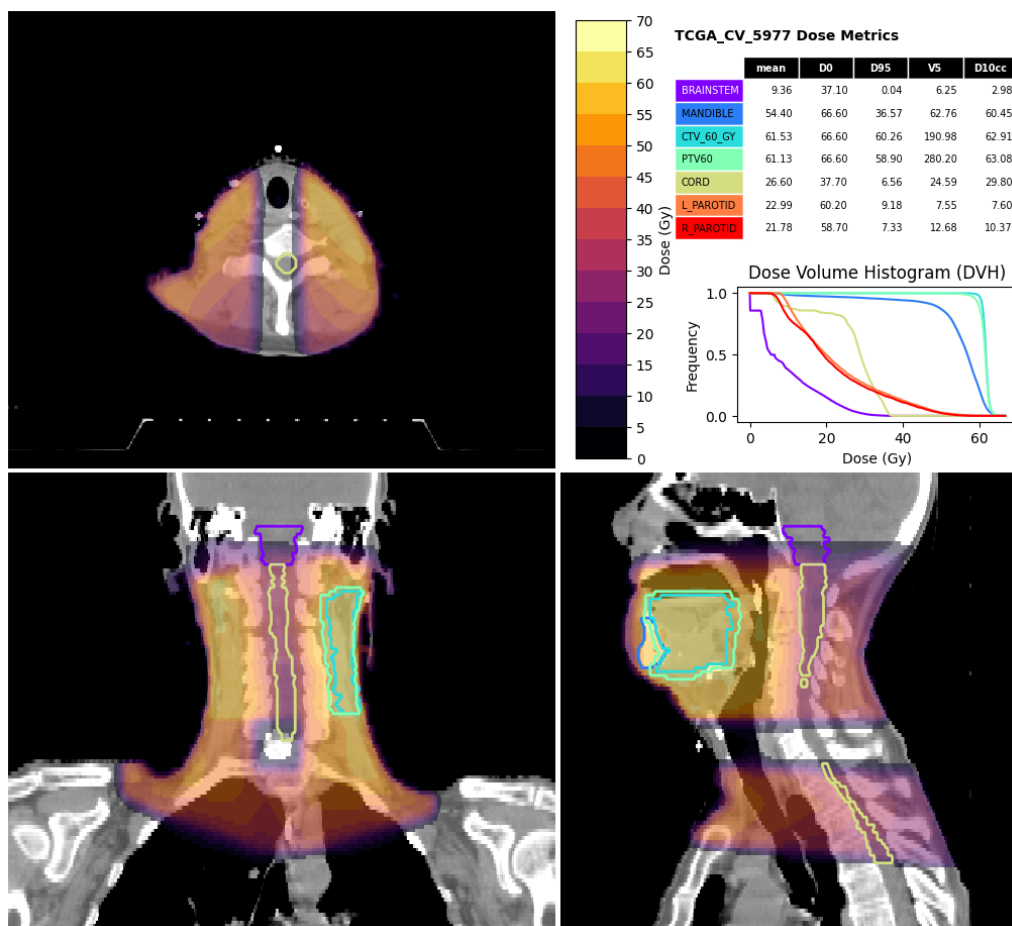


Figure 1: Visualisation of a patient's planned radiotherapy dose overlaid on the corresponding CT image produced using PlatiPy.

This tool has proven useful across many projects and is capable of producing visualisations fit for publication, such as those in (Finnegan et al., 2021, 2022).

DICOM Conversion

The DICOM standard is ubiquitous in the medical imaging field and is often the format in which research datasets are provided. While DICOM is useful in clinical scenarios, it can be challenging to work with and often researchers prefer other image formats, such as NIfTI, when developing code in context of a research project. The SimpleITK library can convert image volumes, such as CT and MRI, to NIfTI format, but conversion of other modalities specific to radiotherapy requires additional steps. To address this, PlatiPy provides functions for converting structure sets (RTSTRUCT) to separate NIfTI masks, as well as radiotherapy dose grids (RTDOSE) to a volume stored in NIfTI format.

Additionally, PlatiPy also provides functionality for converting data back to the DICOM format for analysis in clinical systems. This includes the ability to generate DICOM images, RTSTRUCT, and RTDOSE files from NIfTI.

Image Registration

PlatiPy offers several functions that make it easy to perform linear (rigid) and non-linear (deformable) registration, as well as apply these transformations to images [Figure 2]. These registration tools are put to use in the atlas-based segmentation projects allowing users to easily start performing this kind of segmentation using their own data.

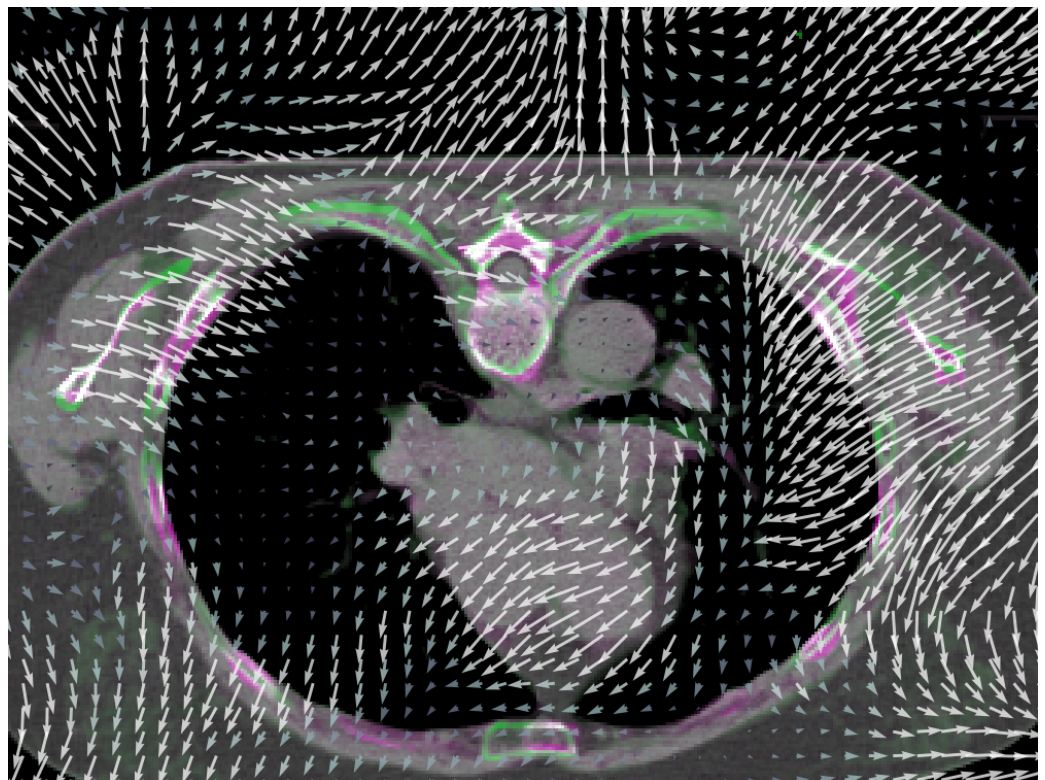


Figure 2: Registration between two CT images produced by PlatiPy and visualised with the ImageVisualiser to compare the registered images (magenta/green) as well as an overlay of the deformation vector field.

Auto-segmentation

PlatiPy provides a variety of tools and functionality that have been utilised in several research projects aimed at developing auto-segmentation models. These include models based on thresholding, atlases, and deep learning. The library includes various tools for performing pre-

and post-processing, image registration, and deep learning inference, making it a valuable resource for researchers working on auto-segmentation projects.

To date, two auto-segmentation models developed using PlatiPy have been deployed and made available through the library for direct use by researchers. The first is a cardiac sub-structure auto-segmentation model, which utilises a deep learning component to segment the whole heart, followed by an atlas-based segmentation and geometric definitions to segment 17 cardiac sub-structures on radiotherapy CT images (Finnegan et al., 2023). The second is a bronchial tree segmentation algorithm that employs threshold techniques to segment the lungs, followed by the airways in radiotherapy lung CT images (Ghandourh et al., 2021).

Metric computation

Computing similarity metrics between an auto-segmentation and a reference segmentation is a common task when working in the medical image analysis space. PlatiPy aims to provide a comprehensive collection of metrics to avoid the need for researchers to implement their own metrics or piece together functions from a range of other implementations. In addition, PlatiPy supplies functionality to compute several metrics at once and produce a visualisation of the results [Figure 3] which enables rapid development of medical image analysis research tools.

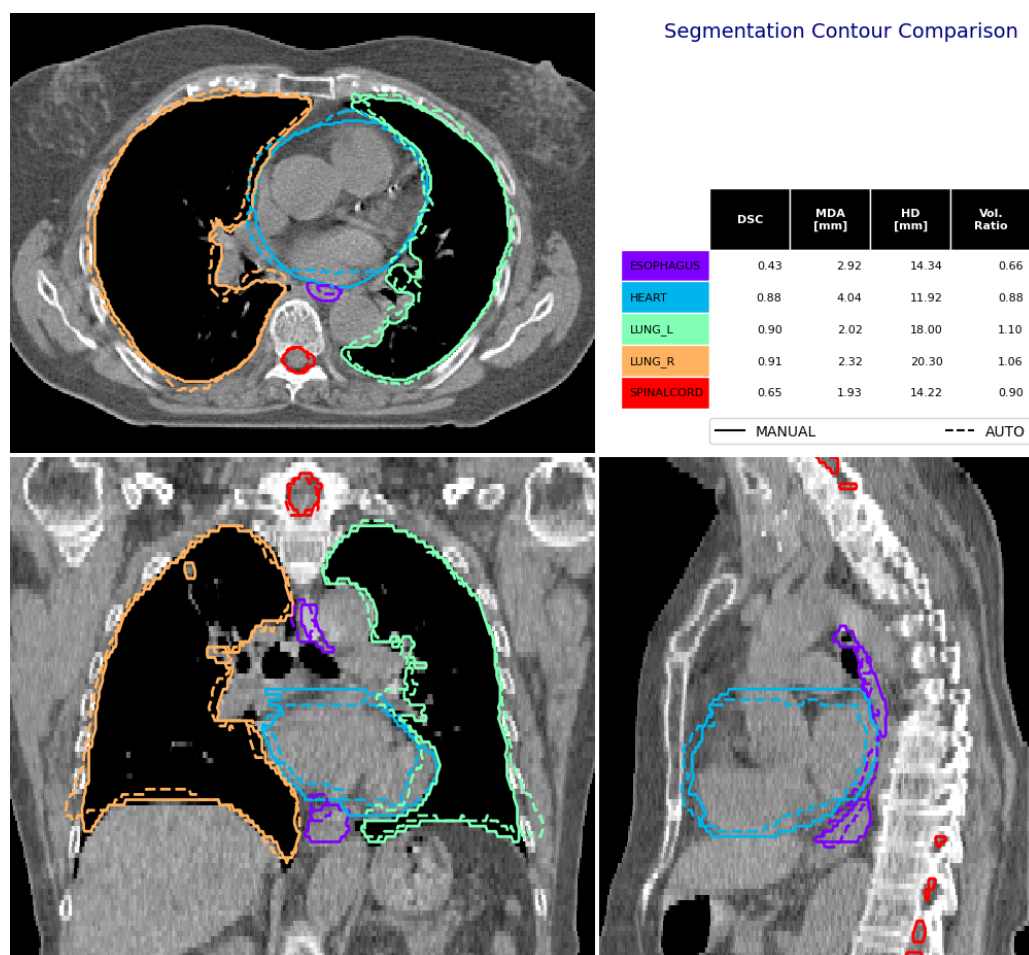


Figure 3: Example of visualisation produced by PlatiPy's contour comparison tool.

Acknowledgements

The authors of PlatiPy would like to thank the Medical Physics groups at the Ingham Institute and University of Sydney for their valuable contributions and for putting the library to use in their research. Their feedback and suggestions have played a crucial role in driving the development of PlatiPy. The authors also acknowledge the funding provided by the Australian Research Data Commons (ARDC) as part of the Australian Cancer Data Network (ACDN), which has been instrumental in supporting the development of the library. The authors would also like to thank Simon Biggs and the team at PyMedPhys for their inspiration and influence on the development of PlatiPy (Biggs et al., 2022).

Furthermore, the authors would like to acknowledge the use of data from The Cancer Imaging Archive in PlatiPy for examples, automated tests, and the figures in this paper (Clark et al., 2013).

References

- Biggs, S., Jennings, M., Swerdloff, S., Chlap, P., Lane, D., Rembish, J., McAloney, J., King, P., Ayala, R., Guan, F., Lambri, N., Crewson, C., & Sobolewski, M. (2022). PyMedPhys: A community effort to develop an open, Python-based standard library for medical physics applications. *Journal of Open Source Software*, 7(78), 4555. <https://doi.org/10.21105/joss.04555>
- Cardoso, M. J., Li, W., Brown, R., Ma, N., Kerfoot, E., Wang, Y., Murray, B., Myronenko, A., Zhao, C., Yang, D., Nath, V., He, Y., Xu, Z., Hatamizadeh, A., Zhu, W., Liu, Y., Zheng, M., Tang, Y., Yang, I., ... Feng, A. (2022). *MONAI: An open-source framework for deep learning in healthcare*. <https://doi.org/10.48550/arXiv.2211.02701>
- Clark, K., Vendt, B., Smith, K., Freymann, J., Kirby, J., Koppel, P., Moore, S., Phillips, S., Maffitt, D., Pringle, M., Tarbox, L., & Prior, F. (2013). The cancer imaging archive (TCIA): Maintaining and operating a public information repository. *Journal of Digital Imaging*, 26(6), 1045–1057. <https://doi.org/10.1007/s10278-013-9622-7>
- Finnegan, R. N., Chin, V., Chlap, P., Haidar, A., Otton, J., Dowling, J., Thwaites, D. I., Vinod, S. K., Delaney, G. P., & Holloway, L. (2023). Open-source, fully-automated hybrid cardiac substructure segmentation: development and optimisation. *Physical and Engineering Sciences in Medicine*. <https://doi.org/10.1007/s13246-023-01231-w>
- Finnegan, R. N., Orlandini, L., Liao, X., Yin, J., Lang, J., Dowling, J., & Fontanarosa, D. (2021). Feasibility of using a novel automatic cardiac segmentation algorithm in the clinical routine of lung cancer patients. *PLoS ONE*, 16(1 January), 1–14. <https://doi.org/10.1371/journal.pone.0245364>
- Finnegan, R. N., Reynolds, H. M., Ebert, M. A., Sun, Y., Holloway, L., Sykes, J. R., Dowling, J., Mitchell, C., Williams, S. G., Murphy, D. G., & Haworth, A. (2022). A statistical, voxelised model of prostate cancer for biologically optimised radiotherapy. *Physics and Imaging in Radiation Oncology*, 21(February), 136–145. <https://doi.org/10.1016/j.phro.2022.02.011>
- Ghandourh, W., Dowling, J., Chlap, P., Oar, A., Jacob, S., Batumalai, V., & Holloway, L. (2021). Assessing tumor centrality in lung stereotactic ablative body radiotherapy (SABR): the effects of variations in bronchial tree delineation and potential for automated methods. *Medical Dosimetry*, 46, 94–101. <https://doi.org/10.1016/j.meddos.2020.09.004>
- Hunter, J. D. (2007). Matplotlib: A 2D graphics environment. *Computing in Science & Engineering*, 9(3), 90–95. <https://doi.org/10.1109/MCSE.2007.55>
- Lowe, B. C., Chen, D. T., Ibáñez, L., & Blezek, D. (2013). The design of simpleITK. *Frontiers in Neuroinformatics*, 7(DEC), 1–14. <https://doi.org/10.3389/fninf.2013.00045>

- Maher, Ashley. (2022). *OnkoDICOM* (0.05.02-beta). <https://onkodicom.com.au>
- Maier, Oskar. (2019). *Medpy - medical image processing in python* (Version 0.4.0). <https://github.com/loli/medpy>
- Mason, D., Scaramallion, Mrbean-Bremen, Rhaxton, Suever, J., Vanessasaurus, Orfanos, D. P., Lemaitre, G., Panchal, A., Rothberg, A., Herrmann, M. D., Massich, J., Kerns, J., Korijn Van Golen, Robitaille, T., Biggs, S., Moloney, Bridge, C., Shun-Shin, M., ... Wada, M. (2022). *Pydicom/pydicom: Pydicom 2.3.1*. Zenodo. <https://doi.org/10.5281/ZENODO.7319790>
- MITK Team. (2023). *MITK* (Version v2023.04). <https://github.com/MITK/MITK>
- National Institutes of Health. (2022). *MIPAV (medical image processing, analysis, and visualization)* (Version 11.0.7). <https://mipav.cit.nih.gov>
- Panchal, A., & Keyes, R. (2010). SU-GG-t-260: Dicompyler: An open source radiation therapy research platform with a plugin architecture. *Medical Physics*, 37(6Part19), 3245–3245. <https://doi.org/10.1118/1.3468652>
- Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., & Duchesnay, Édouard. (2011). Scikit-learn: Machine learning in python. *Journal of Machine Learning Research*, 12(85), 2825–2830. <http://jmlr.org/papers/v12/pedregosa11a.html>
- Yaniv, Z., Lowekamp, B. C., Johnson, H. J., & Beare, R. (2018). SimpleITK Image-Analysis Notebooks: a Collaborative Environment for Education and Reproducible Research. *Journal of Digital Imaging*, 31(3), 290–303. <https://doi.org/10.1007/s10278-017-0037-8>