The Journal of Open Source Software

AtlasReader: A Python package to generate coordinate tables, region labels, and informative figures from statistical MRI images

Michael Philipp Notter[1], **Dan Gale**[2], **Peer Herholz**[3, 4, 5], **Ross Markello**[5], **Marie-Laure Notter-Bielser**[6], **and Kirstie Whitaker**[7]

1 The Laboratory for Investigative Neurophysiology (The LINE), Department of Radiology and Department of Clinical Neurosciences, Lausanne, Switzerland; Center for Biomedical Imaging (CIBM), Lausanne, Switzerland **2** Centre for Neuroscience Studies, Queen's University, Kingston, Canada **3** Laboratory for Multimodal Neuroimaging, Philipps-University Marburg, Hesse, Germany **4** International Laboratory for Brain, Music and Sound Research, Université de Montréal & McGill University, Montréal, Canada **5** McConnell Brain Imaging Centre, Montréal Neurological Institute, McGill University, Montréal, Canada **6** Centre Leenaards de la Mémoire, Centre Hospitalier Universitaire Vaudois, Lausanne, Switzerland **7** Alan Turing Institute, London, UK; Department of Psychiatry, University of Cambridge, Cambridge, UK

DOI: 10.21105/joss.01257

Software

- Review
- Repository
- Archive

Submitted: 05 February 2019
Published: 24 February 2019

Summary

A major advantage of magnetic resonance imaging (MRI) over other neuroimaging methods is its capability to noninvasively locate a region of interest (ROI) in the human brain. For example, using functional MRI, we are able to pinpoint where in the brain a cognitive task elicits higher activation relative to a control. But just knowing the Cartesian coordinate of such a ROI is not useful if we cannot assign it a neuroanatomical label. For this reason, MRI images are usually normalized into a common template space (Fonov et al., 2011), where well-established atlases can be used to associate a given coordinate with the label of a brain region. Most major neuroimaging software packages provide some functionality to locate the main peaks of an ROI but this functionality is often restricted to a few atlases, frequently requires manual intervention, does not give the user much flexibility in the output creation process, and never considers the full extent of the ROI.

To tackle those shortcomings, we created AtlasReader, a Python interface for generating coordinate tables and region labels from statistical MRI images. With AtlasReader, users can use any of the freely and publicly available neuroimaging atlases, without any restriction to their preferred software package, to create publication-ready output figures and tables that contain relevant information about the peaks and clusters extent of each ROI. To our knowledge, providing atlas information about the full extent of a cluster, i.e. over which atlas regions does a ROI extent, is a new feature that is not available in any other, comparable neuroimaging software package.

Executing AtlasReader on an MRI image will create the following four outputs:

1. An **overview figure** showing all ROIs throughout the whole brain (Fig. 1).
2. For each ROI, an **informative figure** showing the sagittal, coronal and transversal plane centered on the main peak of the ROI (Fig. 2).
3. A **table** containing information about the main **peaks** in each ROI (Fig. 3).
4. A **table** containing information about the **cluster extent** of each ROI (Fig. 4).

Users have many parameters available to guide the creation of these outputs. For example, with `cluster_extent` a user can specify the minimum number of contiguous voxels

The Journal of Open Source Software

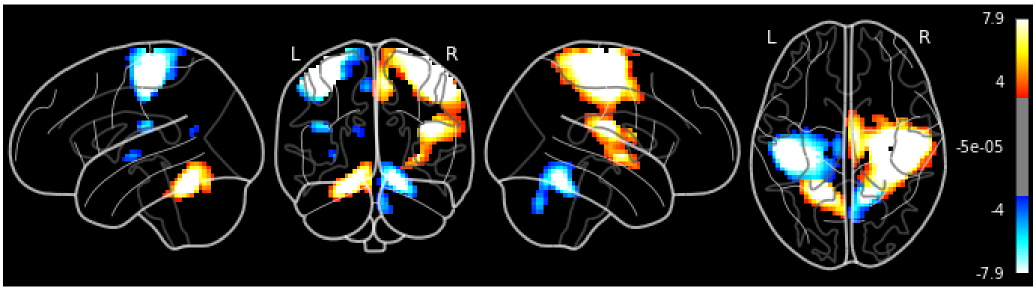

Figure 1: Overview figure showing the ROIs throughout the whole brain at once.

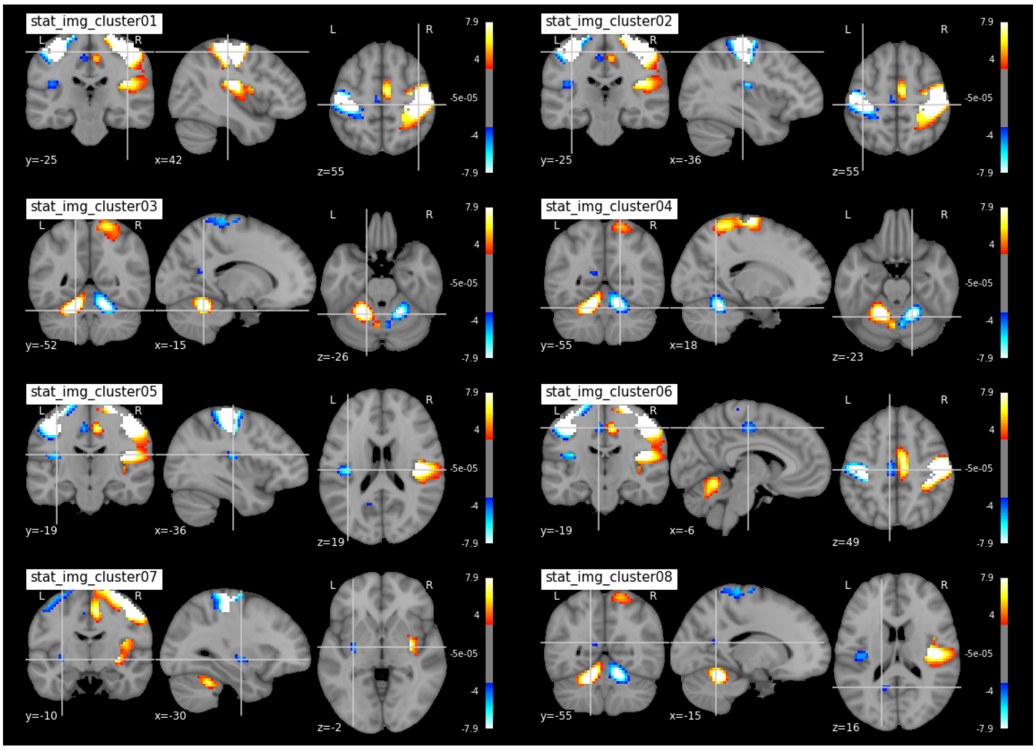

Figure 2: Eight cluster figures, each centered on the main peak of the ROI, showing the sagittal, coronal and transversal plane of the ROI.

required for a ROI to be shown in the output, `min_distance` can be used to extract information from multiple peaks within a given ROI, and `atlas` can be used to specify which atlases should be used for the output creation. By default, AtlasReader uses the AAL, the Desikan-Killiany, and the Harvard-Oxford atlases (Fig. 5). In the current version, users also have access to the Aicha, the Destrieux, the Juelich, the Marsatlas, the Neuromorphometrics, and the Talairach atlas. Further details about the individual atlases, how to acknowledge them, and their license requirements are detailed in the atlasreader/data directory.

AtlasReader is licensed under the BSD-3 license and depends on the following python libraries: `matplotlib` (Hunter, 2007), `nibabel` (Brett et al., 2018), `nilearn` (Abraham et al., 2014), `numpy` (Oliphant, 2007), `scipy` (Jones, Oliphant, Peterson, & others, 2001), `scikitlearn` (Pedregosa et al., 2011) and `scikitimage` (Van der Walt et al., 2014).

For a more detailed explanation about how AtlasReader works and instructions on how to install the software on your system, see https://github.com/miykael/atlasreader.

cluster_id	peak_x	peak_y	peak_z	peak_value	volume_mm	aal	desikan_killiany	harvard_oxford
1	6	-10	52	7.94135	58563	Supp_Motor_Area_R	ctx-rh-paracentral	53% Right_Juxtapositional_Lobule_Cortex_(forme...
1	45	-19	16	7.94135	58563	Rolandic_Oper_R	Unknown	43% Right_Central_Opercular_Cortex; 40% Right_...
1	42	-25	58	7.94135	58563	Postcentral_R	ctx-rh-postcentral	43% Right_Postcentral_Gyrus; 17% Right_Precent...
1	33	-7	-2	7.90531	58563	Putamen_R	Right-Putamen	71% Right_Putamen
1	42	-1	13	5.47070	58563	Rolandic_Oper_R	Unknown	74% Right_Central_Opercular_Cortex; 7% Right_I...
1	9	2	73	3.56015	58563	Supp_Motor_Area_R	ctx-rh-superiorfrontal	49% Right_Superior_Frontal_Gyrus; 6% Right_Jux...
2	-30	-19	67	-7.94144	19089	Precentral_L	Left-Cerebral-White-Matter	46% Left_Precentral_Gyrus
3	-15	-52	-26	7.94135	9612	no_label	Left-Cerebellum-Cortex	0% no_label
4	18	-55	-23	-7.94144	8505	Cerebelum_6_R	Right-Cerebellum-Cortex	0% no_label
4	6	-70	-38	-5.30572	8505	Vermis_8	Right-Cerebellum-Cortex	0% no_label
5	-36	-19	19	-6.21808	1161	Insula_L	Unknown	37% Left_Central_Opercular_Cortex; 37% Left_In...
6	-6	-19	49	-5.03538	1134	Cingulate_Mid_L	ctx-lh-paracentral	50% Left_Precentral_Gyrus; 9% Left_Juxtapositi...
7	-30	-10	-2	-4.65454	378	Putamen_L	Left-Putamen	98% Left_Putamen
8	-15	-55	16	-3.57240	243	Precuneus_L	Left-Cerebral-White-Matter	35% Left_Precuneous_Cortex

Figure 3: Example of a peak table showing relevant information for the main peaks of each ROI. This table contains the cluster association and location of each peak, its signal value at this location, the cluster extent (in mm, not in number of voxels), as well as the membership of each peak, given a particular atlas.

cluster_id	peak_x	peak_y	peak_z	cluster_mean	volume_mm	aal	desikan_killiany	harvard_oxford
1	42	-25	55	5.80230	58563	29.09% Postcentral_R; 15.17% Precentral_R; 9.1...	31.21% Unknown; 27.43% Right-Cerebral-White-Ma...	28.54% Right_Postcentral_Gyrus; 19.59% Right_P...
2	-36	-25	55	-5.96750	19089	60.82% Postcentral_L; 26.45% Precentral_L; 5.9...	47.81% Left-Cerebral-White-Matter; 19.09% ctx-...	61.10% Left_Postcentral_Gyrus; 35.08% Left_Pre...
3	-15	-52	-26	5.42533	9612	44.10% Cerebelum_6_L; 33.99% Cerebelum_4_5_L; ...	75.84% Left-Cerebellum-Cortex; 19.94% Left-Cer...	78.37% no_label; 10.39% Left_Lingual_Gyrus; 5....
4	18	-55	-23	-5.04111	8505	32.70% Cerebelum_6_R; 32.70% Cerebelum_4_5_R; ...	76.51% Right-Cerebellum-Cortex; 12.38% Right-C...	81.90% no_label; 16.19% Right_Lingual_Gyrus
5	-36	-19	19	-4.36624	1161	72.09% Rolandic_Oper_L; 27.91% Insula_L	48.84% Unknown; 30.23% ctx-lh-supramarginal; 1...	58.14% Left_Central_Opercular_Cortex; 25.58% L...
6	-6	-19	49	-3.82011	1134	50.00% Cingulate_Mid_L; 40.48% Supp_Motor_Area...	40.48% ctx-lh-paracentral; 30.95% Unknown; 14....	71.43% Left_Precentral_Gyrus; 16.67% Left_Juxt...
7	-30	-10	-2	-3.67586	378	92.86% Putamen_L; 7.14% no_label	100.00% Left-Putamen	100.00% Left_Putamen
8	-15	-55	16	-3.28974	243	77.78% Precuneus_L; 11.11% Cuneus_L; 11.11% Ca...	44.44% Left-Cerebral-White-Matter; 33.33% Unkn...	100.00% Left_Precuneous_Cortex

Figure 4: Example of a cluster table showing relevant information for the cluster extent of each ROI. This table contains the cluster association and location of each peak, the mean value within the cluster, the cluster extent (in mm, not in number of voxels), as well as the membership of each cluster, given a particular atlas.

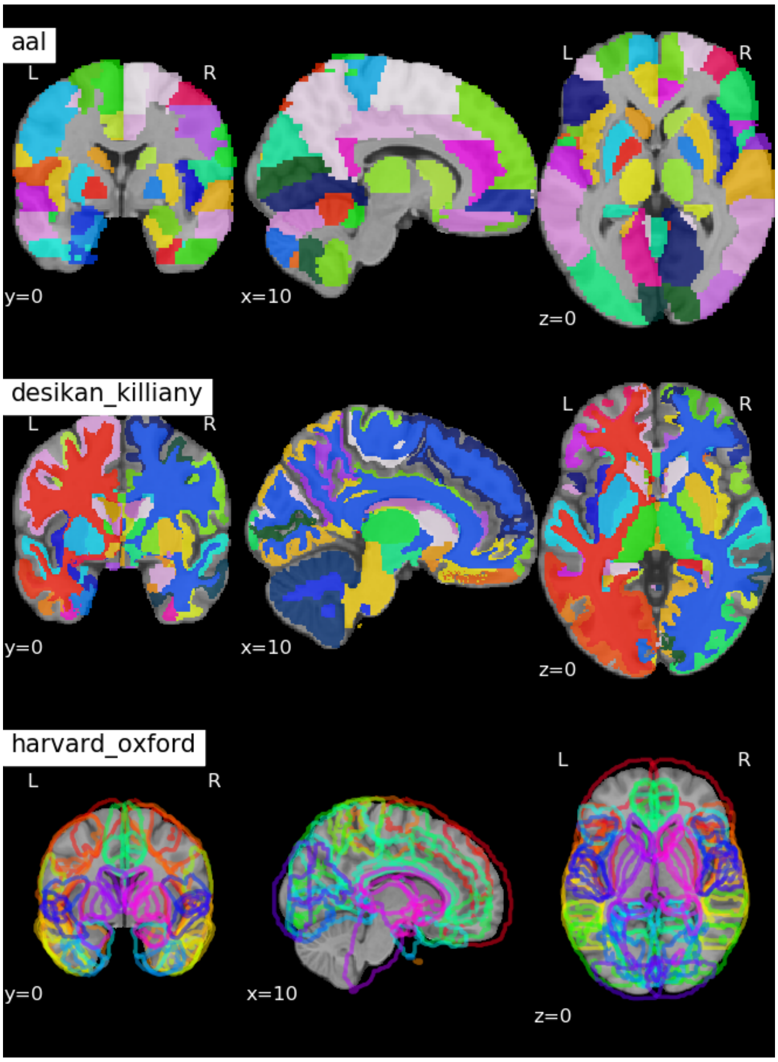

Figure 5: Depiction of AtlasReader's default atlases. Individually colored label of the three default atlases, AAL, Desikan-Killiany and Harvard-Oxford, overlaid on the ICBM 2009c nonlinear asymmetric atlas. The Harvard-Oxford atlas is visualized differently because it is a probability atlas and therefore has overlapping regions.

Acknowledgement

Michael Philipp Notter is grateful for the support of the Swiss National Science Foundation grants, awarded to M. Murray (320030-149982, 320030-169206, and 51NF40-158776). Kirstie Whitaker is funded through an Alan Turing Institute Research Fellowship under EPSRC Research grant (TU/A/000017). This project was started at Neurohackademy 2018 which is funded by the National Institute of Mental Health through a grant to Ariel Rokem and Tal Yarkoni (R25MH112480).

References

Abraham, A., Pedregosa, F., Eickenberg, M., Gervais, P., Mueller, A., Kossaifi, J., Gramfort, A., et al. (2014). Machine learning for neuroimaging with scikit-learn. *Frontiers in neuroinformatics*, *8*, 14. doi:10.3389/fninf.2014.00014

Brett, M., Hanke, M., Markiewicz, C., Côté, M.-A., McCarthy, P., Ghosh, S., Wassermann, D., et al. (2018). Nibabel: Access a cacophony of neuro-imaging file formats, version 2.3.0. doi:10.5281/zenodo.1287921

Fonov, V., Evans, A. C., Botteron, K., Almli, C. R., McKinstry, R. C., Collins, D. L., Group, B. D. C., et al. (2011). Unbiased average age-appropriate atlases for pediatric studies. *Neuroimage*, *54*(1), 313–327. doi:10.1016/j.neuroimage.2010.07.033

Hunter, J. D. (2007). Matplotlib: A 2D graphics environment. *Computing In Science & Engineering*, *9*(3), 90–95. doi:10.1109/MCSE.2007.55

Jones, E., Oliphant, T., Peterson, P., & others. (2001). SciPy: Open source scientific tools for Python. Retrieved from http://www.scipy.org/

Oliphant, T. E. (2007). Python for scientific computing. *Computing in Science & Engineering*, *9*(3). doi:10.1109/MCSE.2007.58

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., et al. (2011). Scikit-learn: Machine learning in python. *Journal of machine learning research*, *12*(Oct), 2825–2830. Retrieved from https://scikit-learn.org/

Van der Walt, S., Schönberger, J. L., Nunez-Iglesias, J., Boulogne, F., Warner, J. D., Yager, N., Gouillart, E., et al. (2014). Scikit-image: Image processing in python. *PeerJ*, *2*, e453. doi:10.7717/peerj.453