

RESEARCH ARTICLE

pyKNEEr: An image analysis workflow for open and reproducible research on femoral knee cartilage

Serena Bonaretti^{1,2*}, Garry E. Gold¹, Gary S. Beaupre^{2,3}

1 Department of Radiology, Stanford University, Stanford, CA, United States of America, **2** Musculoskeletal Research Laboratory, VA Palo Alto Health Care System, Palo Alto, CA, United States of America, **3** Department of Bioengineering, Stanford University, Stanford, CA, United States of America

✉ Current address: Transparent MSK Research, Maastricht, The Netherlands

* serena.bonaretti.research@gmail.com



OPEN ACCESS

Citation: Bonaretti S, Gold GE, Beaupre GS (2020) pyKNEEr: An image analysis workflow for open and reproducible research on femoral knee cartilage. PLoS ONE 15(1): e0226501. <https://doi.org/10.1371/journal.pone.0226501>

Editor: Ivan Olier, Liverpool John Moores University, UNITED KINGDOM

Received: May 8, 2019

Accepted: November 27, 2019

Published: January 24, 2020

Copyright: This is an open access article, free of all copyright, and may be freely reproduced, distributed, transmitted, modified, built upon, or otherwise used by anyone for any lawful purpose. The work is made available under the [Creative Commons CC0](https://creativecommons.org/licenses/by/4.0/) public domain dedication.

Data Availability Statement: - Raw data from the Osteoarthritis Initiative are here: <https://nda.nih.gov/oai> - Processed data are on Zenodo: <https://www.doi.org/10.5281/zenodo.2530608> - Data analysis is on GitHub: <https://github.com/sbonaretti/pyKNEEr/tree/master/publication/data>.

Funding: S.B. and G.S.B. were supported by Merit Review award I01 RX001811 from the United States (U.S.) Department of Veterans Affairs, Rehabilitation Research & Development Service (<https://www.rehab.research.va.gov>). G.E.G. was supported by awards R01 EB002524 and K24

Abstract

Transparent research in musculoskeletal imaging is fundamental to reliably investigate diseases such as knee osteoarthritis (OA), a chronic disease impairing femoral knee cartilage. To study cartilage degeneration, researchers have developed algorithms to segment femoral knee cartilage from magnetic resonance (MR) images and to measure cartilage morphology and relaxometry. The majority of these algorithms are not publicly available or require advanced programming skills to be compiled and run. However, to accelerate discoveries and findings, it is crucial to have open and reproducible workflows. We present pyKNEEr, a framework for open and reproducible research on femoral knee cartilage from MR images. pyKNEEr is written in python, uses Jupyter notebook as a user interface, and is available on GitHub with a GNU GPLv3 license. It is composed of three modules: 1) image preprocessing to standardize spatial and intensity characteristics; 2) femoral knee cartilage segmentation for intersubject, multimodal, and longitudinal acquisitions; and 3) analysis of cartilage morphology and relaxometry. Each module contains one or more Jupyter notebooks with narrative, code, visualizations, and dependencies to reproduce computational environments. pyKNEEr facilitates transparent image-based research of femoral knee cartilage because of its ease of installation and use, and its versatility for publication and sharing among researchers. Finally, due to its modular structure, pyKNEEr favors code extension and algorithm comparison. We tested our reproducible workflows with experiments that also constitute an example of transparent research with pyKNEEr, and we compared pyKNEEr performances to existing algorithms in literature review visualizations. We provide links to executed notebooks and executable environments for immediate reproducibility of our findings.

Introduction

Open science and computational reproducibility are recent movements in the scientific community that aim to promote and encourage transparent research. They are supported by

AR062068 from the U.S. National Institutes of Health (NIH) (<https://www.nih.gov/>). The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.

Competing interests: The authors have declared that no competing interests exist.

national and international funding agencies, such as the United States National Institutes of Health (NIH) [1] and the European Commission [2]. Open science refers to the free availability of data, software, and methods developed by researchers with the aim to share knowledge and tools [3]. Computational reproducibility is the ability of researchers to duplicate the results of a previous study, using the same data, software, and methods used by the original authors [4]. Openness and reproducibility are essential to researchers to assess the accuracy of scientific claims [5], build on the work of other scientists with confidence and efficiency (i.e. without “reinventing the wheel”) [6], and collaborate to improve and expand robust scientific workflows to accelerate scientific discoveries [7–9]. Historically, research data, tools, and processes were rarely openly available because of limited storage and computational power [9]. Nowadays, there are several opportunities to conduct transparent research: data repositories (e.g. [Zenodo](#) and [FigShare](#)), code repositories (e.g. [GitHub](#), [GitLab](#), and [Bitbucket](#)), and platforms for open science (e.g. [The European Open Science Cloud](#) and [Open Science Framework](#)). In addition, there exist computational notebooks that combine narrative text, code, and visualization of results (e.g. [Jupyter notebook](#) [10, 11] and [R markdown](#) [12]), allowing researchers to create workflows that are computationally transparent and well documented [6]. Finally, it is possible to recreate executable environments from repositories to run notebooks directly in a browser and thus make code immediately reproducible (e.g. [Binder](#) [13]).

In the evolution of research practice, the structure of scientific papers, intended as vehicles to communicate methods and results to peers, is changing. In 1992, Claerbout was among the first to envision interactive publications: “[...] an author attaches to every figure caption a pushbutton or a name tag usable to recalculate the figure from all its data, parameters, and programs. This provides a concrete definition of reproducibility in computationally oriented research” [14]. Following this vision, papers are transforming from static to interactive. They will progressively integrate data and code repositories, metadata files describing data characteristics (e.g. origin, selection criteria, etc.), and computational notebooks used to compute results and create graphs and tables [15, 16] for more transparent research.

Transparency in image-based research is crucial to provide meaningful and reliable answers to medical and biological questions [17]. In the musculoskeletal field, quantitative analysis from magnetic resonance (MR) imaging has assumed an increasingly important role in investigating osteoarthritis (OA) [18]. OA is the most common joint disease worldwide, affecting about 2 in 10 women and 1 in 10 men over 60 years of age [19]. It causes structural changes and loss of articular cartilage, with consequent pain, stiffness, and limitation of daily activities [20]. OA of the knee is one of the main forms of OA, affecting 1/3 of the adults with OA [21] and accounting for 83% of the total OA economic burden [22]. To investigate knee OA, scientists have developed algorithms to preprocess MR images, segment femoral knee cartilage, and extract quantitative measurements of morphology, such as thickness [23] and volume [24], and relaxation times, such as $T_{1\rho}$ and T_2 [25].

In the image analysis pipeline, segmentation constitute a major challenge. Researchers still tend to segment femoral knee cartilage manually or semi-automatically, using commercial or in-house software, in a tedious and non-reproducible manner [26, 27]. However, there exist several algorithms that researchers have developed to automatically segment knee cartilage. In the literature and in published reviews [28–30], we have found 29 relevant publications that propose new algorithms to segment femoral knee cartilage. These algorithms are based on different principles, namely active contours, atlas-based, graph-based, machine and deep learning, and hybrid combinations, and were developed by various research groups worldwide, as depicted in the literature review visualization in [Fig 1](#). Of these, only the implementations by Wang et al. [31] and by Shan et al. [32] are open-source and hosted in public repositories (see

Wang's repository and Shan's repository). These two implementations, however, have some limitations: in the first case, documentations of code and usage are not extensive, while in the second case the code is written in C++ and requires advanced programming skills to be compiled and run. Other communities, such as neuroimaging, largely benefit from robust, open-source, and easy-to-use software to segment and analyze images (e.g. ANTs [33], FreeSurfer [34], Nipype [35]). Because of these open-access tools, researchers do not need to re-implement algorithms for basic processing and can focus on further statistical analyses [36–38]. To accelerate discoveries and findings, it is fundamentally important to have not only open-source tools, but also workflows that are computationally reproducible, and thus enhance scientific rigor and transparency [8].

In this paper, we present `pyKNEEr`, an automatic workflow to preprocess, segment, and analyze femoral knee cartilage from MR images specifically designed for open and reproducible research. The main characteristics of `pyKNEEr` are embedded in its name: *py* is for *python*, to indicate openness, *KNEE* is for femoral knee cartilage, and *r* is for *reproducibility*. `pyKNEEr` is written in python with Jupyter notebooks as graphical user-interface, is shared on GitHub, and has a documentation website. In addition, we provide an example of transparent research with `pyKNEEr` through our validation study, implemented using images from the Osteoarthritis Initiative (OAI) [68] as well as in-house images. Finally, to be compliant with recommendations for interactive publications, throughout the paper we provide links to data files and repositories, software repositories, specific code and Jupyter notebooks, executable environments, metafiles and web documentation, and websites [69]. In addition, we provide an example of transparent research with `pyKNEEr` through our validation study, implemented using images from the Osteoarthritis Initiative (OAI) [68] as well as in-house images. Finally, to be compliant with recommendations for interactive publications, throughout the paper we provide links to data files and repositories, software repositories, specific code and Jupyter notebooks, executable environments, metafiles and web documentation, and websites [69].

Characteristics and structure of `pyKNEEr`

Openness: Python, file formats, code reuse, and GitHub

Characteristics and structure of `pyKNEEr` are based on recommendations for open scientific software in the literature, such as usage of open language and file formats, code reuse, and licensed distribution [6, 7, 70]. We wrote `pyKNEEr` in the open language *python*, using open-access libraries to perform computations, such as *NumPy* for linear algebra [71, 72], *pandas* for data analysis [73], *matplotlib* for visualizations [74], *SimpleITK* for medical image processing and analysis [75], and *itkwidgets* for 3D rendering. We used widespread open-source formats for input and output data, such as text files (`.txt`) for input image lists, *dicom* (`.dcm`) and *metafile* (`.mha`) for images, and *tabular files* (`.csv`) for tables. To favor our code reuse, we organized `pyKNEEr` in three modules: 1) image preprocessing; 2) femoral knee cartilage segmentation; and 3) morphology and relaxometry analysis. Modularity will allow us and other researchers to test, enhance, and expand the code by simply modifying, substituting, or adding Jupyter notebooks. At the same time, we reused open-source code developed by other scientists, such as preprocessing algorithms developed by Shan et al. [32] and *elastix* for atlas-based segmentation [76]. Finally, we released `pyKNEEr` on GitHub with a GNU GPLv3 license, which requires openness of derivative work. For citation, we assigned `pyKNEEr` a digital object identifier (DOI), obtained through the merge of the GitHub release to Zenodo (Table 1).

Literature map of femoral knee cartilage segmentation

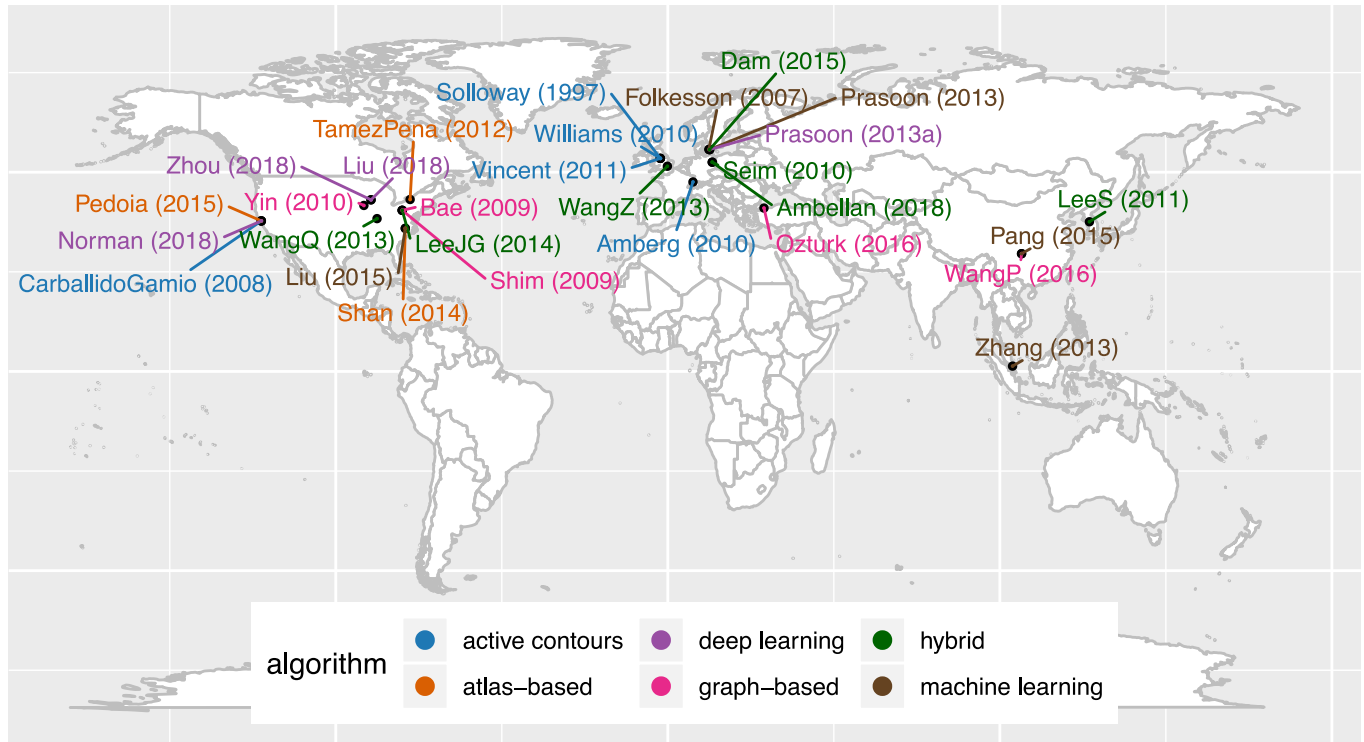


Fig 1. Literature map of femoral knee cartilage segmentation. The visualization shows name of first author, year of publication, affiliation of last author, and segmentation method for 29 relevant publications on femoral knee cartilage segmentation from 1997 to 2018. Publications by segmentation method and in alphabetical order are: Active contours: Amberg(2010) [39], Carballido-Gamio(2008) [40], Solloway(1997) [41], Vincent(2011) [42], Williams(2010) [43]; Atlas-based: Padoia(2015) [44], Shan(2014) [32], Tamez-Pena(2012) [45]; Deep-learning: Liu(2018) [46], Norman(2018) [47], Prasoorn(2013a) [48], Zhou(2018) [49]; Graph-based: Bae(2009) [50], Ozturk(2016) [51], Shim(2009) [52], WangP(2016) [53], Yin(2010) [54]; Hybrid: Ambellan(2018) [55], Dam(2015) [56], LeeJG(2014) [57], LeeS(2011) [58], Seim(2010) [59], WangQ(2013) [31], WangZ(2013) [60]; Machine learning: Folkesson(2007) [61], Liu(2015) [62], Pang(2015) [63], Prasoorn(2013) [64], Zhang(2013) [65]. This graph and graphs in Figs 4 and 5 were made in Jupyter notebook using ggplot2 [66], an R package based on the grammar of graphics [67]. (See [data](#), [code](#), [executable environment](#)).

<https://doi.org/10.1371/journal.pone.0226501.g001>

Table 1. Openness and reproducibility of pyKNEER code and experimental data.

	Repository	Metadata / Documentation	Language / Format	License	DOI	Citation
<i>Software Used</i>						
Preprocessing	Bitbucket	Wiki	C++, ITK	Apache	https://doi.org/10.1016/j.media.2014.05.008 *	[32]*
elastix 4.8	GitHub	Github Wiki	C++, ITK	Apache	https://doi.org/10.1109/TMI.2009.2035616 *	[76]*
<i>Developed</i>						
pyKNEER	GitHub	Website	python, Jupyter notebook	GNU GPLv3	https://doi.org/10.5281/zenodo.2574172	Bonaretti S. et al. "pyKNEER" (v0.0.1). Zenodo. 2019. 10.5281/zenodo.2574172
<i>Data</i>						
Original	OAI	Website	.dcm	Data user agreement	https://doi.org/10.1016/j.joca.2008.06.016 *	[68]*
Derived (results)	Zenodo	Jupyter notebook	.mha, .txt, .csv	CC-BY-NC-SA	https://doi.org/10.5281/zenodo.2530608	Bonaretti S. et al. Dataset used in (Bonaretti et al. 2019). Zenodo. 2019. 10.5281/zenodo.2530608

*DOIs and citations of used software and original data refers to their corresponding publication.

<https://doi.org/10.1371/journal.pone.0226501.t001>

Reproducibility: Jupyter notebooks with computational narratives and dependencies

We designed `pyKNEEr` as a tool to perform and support computational reproducible research, using principles recommended in the literature [5, 6]. For each module of the framework, we used one or more Jupyter notebooks as a user-interface, because of their versatility in combining code, text, and visualization, and because they can be easily shared among researchers, regardless of operating systems.

Across `pyKNEEr` modules, we used the same notebook structure for consistent computational narratives (Fig 2). Each notebook contains:

1. Link to the GitHub repository: The repository contains code and additional material, such as source files of documentation and publication;
2. Link to documentation: Each notebook is associated with a webpage containing instructions on how to create input text files, run notebooks, and evaluate outputs. Explanations include step-by-step instructions for a [demo dataset](#), provided to the user to become familiar with the software. Single webpages are part of a documentation website, comprehensive of installation instructions and frequently asked questions. We created the website using [sphinx](#), the python documentation generator;
3. Introduction: Brief explanation of the algorithms in the notebook;
4. User inputs: The input of each notebook is a text file (`.txt`) with folder paths and file names of images to process or analyze. Additional inputs are variables to customize the process, such as number of cores and algorithm options;
5. Commands with narrative: Titles and subtitles define narrative units of computations, and additional texts provide information about command usage and outcome interpretation. Commands in the notebook call functions in python files associated with that notebook (e.g. in the preprocessing module, the notebook `preprocessing.ipynb` calls the python file `preprocessing_for_nb.py`). In turn, associated python files call functions in core files (e.g. the python file `preprocessing_for_nb.py` calls `sitk_functions.py`, containing image handling functions);
6. Visualization of outputs: Qualitative visualizations include sagittal slices with superimposed cartilage mask or relaxometry map, 2D thickness maps, and 3D relaxometry maps, to allow users a preliminary evaluation of outputs. Quantitative visualizations include scatter plots and tables with numerical values and descriptive statistics (Fig 2b), which are also saved in `.csv` files to allow researcher subsequent analysis;
7. References: List of main references used in notebook algorithms;
8. Dependencies: Code dependencies (i.e. version of python, python packages, and computer operating systems and hardware) to allow researchers to recreate the current computational environment and thus reproduce findings. To print dependencies, we used the python package `watermark`.

Algorithms in `pyKNEEr`

`pyKNEEr` contains specific algorithms to preprocess, segment, and analyze femoral knee cartilage from MR images.

Image preprocessing. Spatial and intensity preprocessing provide standardized high quality images to the segmentation algorithm [77]. In spatial preprocessing, we transform images

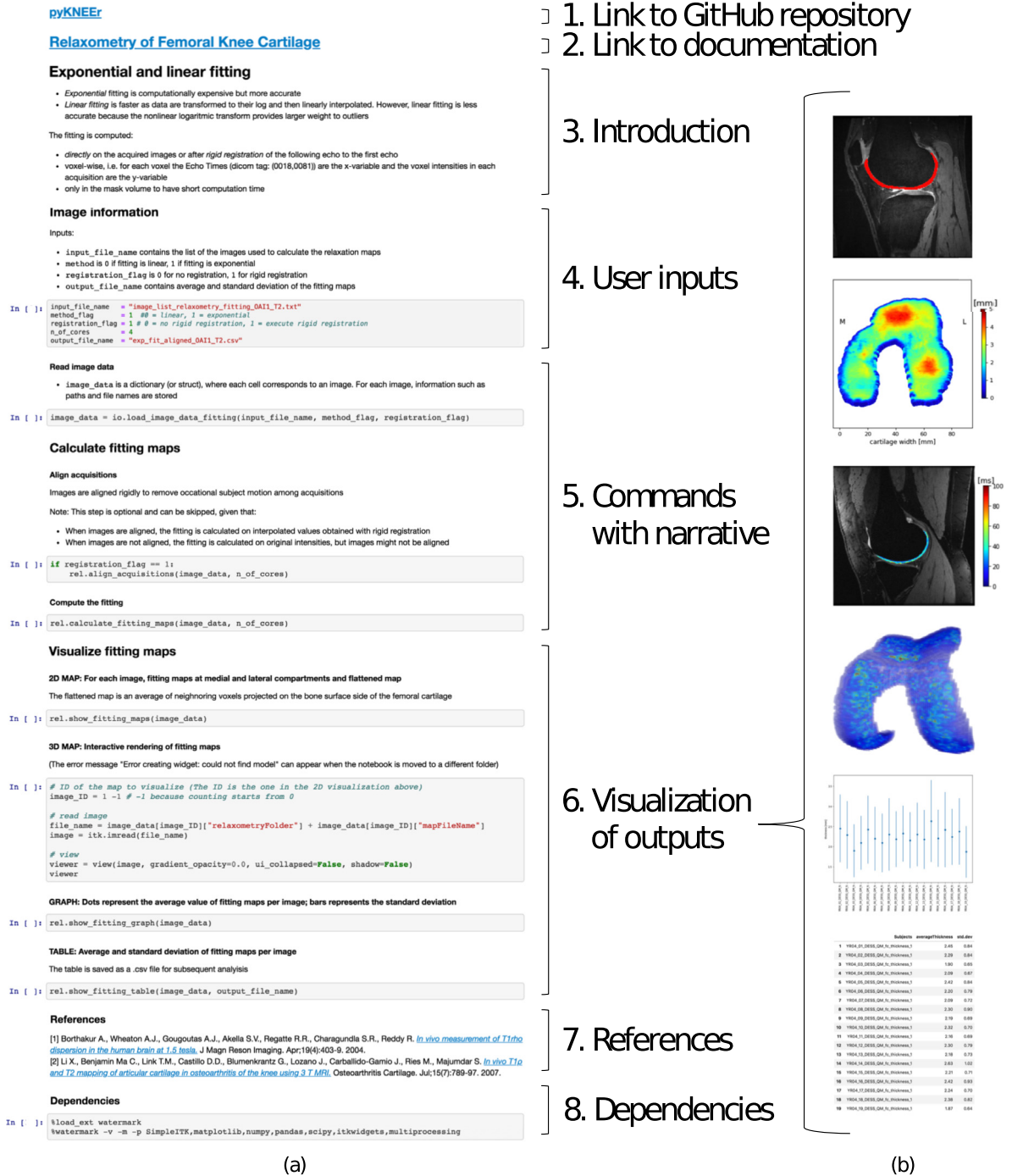


Fig 2. User-interface of modules in pyKNEEr. (a) Structure of Jupyter notebooks and (b) qualitative and quantitative visualization of outputs (from top: cartilage segmentation on image slice, flattened map of cartilage thickness, relaxation map on image slice, 3D relaxation map, and plot and table with average and standard deviation of thickness values).

<https://doi.org/10.1371/journal.pone.0226501.g002>

to right-anterior-inferior (RAI) orientation, we flip right knees (when present) to the left laterality, and we set image origin to the origin of the cartesian system (0,0,0). In intensity preprocessing, we correct image intensities for the inhomogeneities of the static magnetic field (B_0) [78], we rescale intensities to a common range [0–100], and we enhance cartilage edges with

edge-preserving smoothing using curvature flow [79] (Fig 3I). Implementation of intensity preprocessing is a translation of the open access code by Shan et al. [32] from C++ to python.

Femoral knee cartilage segmentation. Three steps comprise femoral cartilage segmentation: 1) finding a reference image; 2) segmenting femoral cartilage; and 3) evaluating segmentation quality (Fig 3II). Finding reference image and evaluating segmentation quality are possible only when ground truth segmentations are available.

Finding a reference image. We propose a convergence study to find the reference image, i.e. a segmented image used as a template, or atlas, for the segmentation. First, we randomly select an image as a reference image. Then, we register all images of the dataset to the reference using rigid, similarity, and spline transformations, as explained in the following paragraph. Next, we average the vector fields that result from the registrations. Finally, we choose the image whose vector field is the closest to the average vector field as the new reference for the following iteration. We repeat until two consecutive iterations converge to the same reference image or after a fixed number of iterations. It is possible to execute the search for the reference several times using different images as the initial reference to confirm the selection of the reference image. This algorithm requires femur masks because the comparison among vector fields and their average is calculated in the femur volume, as cartilage volume is limited.

Atlas-based segmentation. Initially, we register a moving image (i.e. any image of the dataset) to a reference image, by transforming first the femur and then the cartilage. Then, we invert the transformation describing the registration. Finally, we apply the inverted transformation to the cartilage mask of the reference image to obtain the cartilage mask of the moving image. The images to segment can be new subjects (intersubject), images of the same subject acquired at different time points (longitudinal), or images of the same subject acquired with different protocols (multimodal). To segment intersubject images, we use rigid, similarity, and spline registration, to segment longitudinal images only rigid and spline registration, and to segment multimodal images only rigid registration. We perform image registration and mask warping with `elastix` and `transformix`, respectively [44, 76], using a multiresolution approach with smoothing image pyramid, random coordinate sampler, adaptive stochastic gradient descent optimizer, and B-spline interpolators [44]. Detailed parameters are in the code repository (GitHub).

Evaluating segmentation quality. We quantitatively evaluate quality of segmentation using the Dice Similarity Coefficient (DSC) and average surface distance (ASD) for the whole mask region of interest. DSC is a measure of the overlap between a newly segmented mask and the corresponding ground truth segmentation [80]. The Dice Similarity Coefficient is calculated as:

$$DSC = \frac{2|NM \cap GT|}{|NM| + |GT|} \tag{1}$$

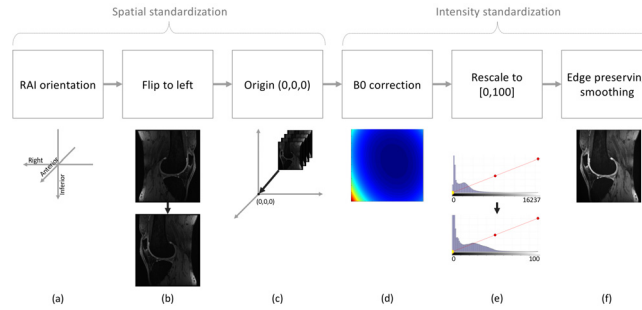
where NM is the newly segmented mask, and GT is the ground truth. ASD is the average of the Euclidean distances between the newly segmented mask and the ground truth mask, calculated as:

$$ASD = \frac{1}{n_{nm} + n_{gt}} \left(\sum_{i=1}^{n_{nm}} \min_{gt \in edge_{GT}} \|nm_i - gt\|_2 + \sum_{j=1}^{n_{gt}} \min_{nm \in edge_{MN}} \|gt_j - nm\|_2 \right) \tag{2}$$

where $edge_{NM}$ is the edge the newly segmented mask, nm is any voxel, and n_{nm} is the number of voxels; similarly, $edge_{GT}$ is the edge the ground truth mask, gt is any voxel, and n_{gt} is the number of voxels. [55].

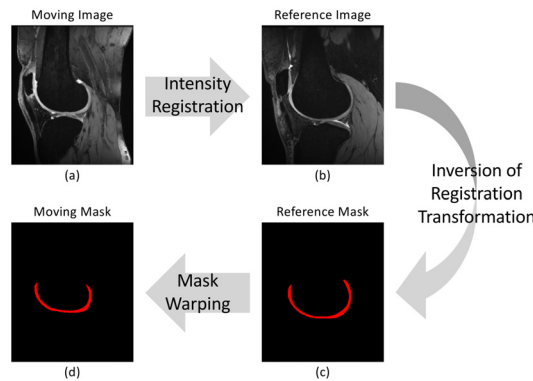
MAIN ALGORITHMS IN pyKNEEr

I. Preprocessing preprocessing.ipynb



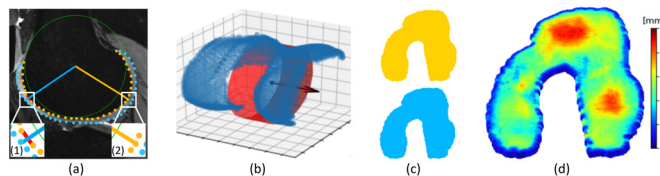
Spatial standardization: (a) image set to RAI orientation (right-anterior-inferior with respect to the subject’s coordinate system), (b) right knee (when present) flipped to left, (c) image origin set to (0,0,0). Intensity standardization: (d) correction of B_0 inhomogeneity, (e) intensity rescaling to [0,100], and (f) edge preserving smoothing.

II. Segmentation segmentation.ipynb (find_reference.ipynb) (segmentation_quality.ipynb)



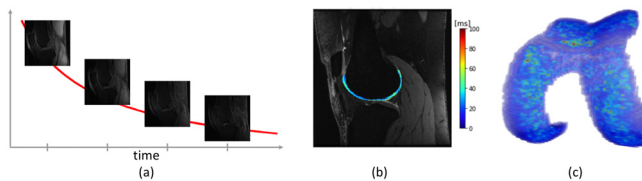
Atlas-based segmentation. We register a moving image (a) to a reference image (b). We invert the computed transformation and we apply it to the mask of the reference image (c) to obtain the mask of the moving image (d).

III. Analysis morphology.ipynb



Calculating cartilage thickness. (a) To separate cartilage sides, we interpolate a circle (green) in the cartilage and we connect its center to cartilage points through rays. When a ray (blue) intersects a segment (red) connecting two consecutive points, the point belongs to articular cartilage (1), otherwise to subchondral cartilage (2). (b) Interpolation to a cylinder to flatten the cartilage. (c) Visualization of subchondral (yellow) and articular (blue) surfaces. (d) 2D map of cartilage thickness.

relaxometry_fitting.ipynb (relaxometry_EPG.ipynb)



Calculating relaxometry maps using exponential fitting. (a) Voxel-wise interpolation of images at subsequent TE or TSL. (b) Relaxation map superimposed on one slice of the image. (c) Relaxation map in 3D.

Fig 3. Main algorithms in pyKNEEr modules. I. Image preprocessing; II. Femoral cartilage segmentation; and III. Analysis of morphology and relaxometry. Left: Names of Jupyter notebooks. In parenthesis, notebooks in the module not depicted here. Middle: Graphic summary of algorithms. Right: Algorithm descriptions.

<https://doi.org/10.1371/journal.pone.0226501.g003>

Morphology and relaxometry analysis. In pyKNEEr, cartilage analysis includes morphology and relaxometry (Fig 3III).

Cartilage morphology. Morphology quantifications are cartilage thickness and cartilage volume. To calculate cartilage thickness, first we extract contours from each slice of the cartilage

mask as a point cloud. Then, we separate the subchondral side of the cartilage from the articular side, we interpolate each cartilage side to a cylinder that we unroll to flatten cartilage [26], and we calculate thickness between the two cartilage sides using a nearest neighbor algorithm in the 3D space [40, 81]. Finally, we associate thicknesses to the subchondral point cloud to visualize them as a 2D map. We compute cartilage volume as the number of voxels of the mask multiplied by the voxel volume.

Cartilage relaxometry. We implemented two algorithms to calculate relaxometry maps: Exponential or linear fitting and Extended Phase Graph (EPG) modeling. We use exponential or linear fitting to compute $T_{1\rho}$ maps from $T_{1\rho}$ -weighted images and T_2 maps from T_2 -weighted images. We calculate exponential fitting by solving a mono-exponential equation voxel-wise using a Levenberg-Marquardt fitting algorithm [25]:

$$S(T_a) = K \cdot \exp(-T_a/T_b) \quad (3)$$

where: for $T_{1\rho}$ -weighted images, T_a is time of spin-lock (TSL) and T_b is $T_{1\rho}$; for T_2 -weighted images, T_a is echo time (TE) and T_b is T_2 ; and K is a constant. We compute linear fitting by transforming the images to their logarithm and then linearly interpolating voxel-by-voxel. Linear fitting is not recommended when signal-to-noise ratio is high because the logarithm transformation alters the normality of noise distribution, but it is fast and computationally inexpensive [82]. Before calculating exponential or linear fitting, the user has the option to register the images with lowest TE or TSL to the image with the highest TE or TSL to correct for image motion during acquisition [83]. We use EPG to calculate T_2 maps from DESS acquisition. The implementation in `pyKNEEr` is the one proposed by Sveinsson et al. [84], which is based on a linear approximation of the relationship between the two DESS signals.

Computational costs. Computation time for one image through the whole pipeline is about 45 minutes on one core on a MacOS laptop with a processor Intel Core i5 at 2.3 GHz and memory of 16 GB at 2133 MHz. On average, the computation times for the workflow steps are: preprocessing: 20 minutes, segmentation: 15 minutes, morphology analysis: 5 minutes, and relaxometry analysis: 5 minutes. To optimize computational effort, we used the [multiprocessing](#) python package to process images on separate cores. Therefore, computation time for a whole dataset is linearly dependent on the number of cores. As an example, computation time for 2 images on one core is about 90 (45x2) minutes, on two cores is 45 minutes.

Open and reproducible research with `pyKNEEr`: Our validation study

We validated `pyKNEEr` with experiments that also constitute an example of open and reproducible research with `pyKNEEr`.

Image data

We used three datasets that we named OAI1, OAI2, and inHouse (Table 2). OAI1 contained 19 Double-Echo in Steady-State (DESS) images and T_2 -weighted (T_2 -w) spin-echo images acquired at year 4 of the OAI study. Ground truth segmentations were created using an atlas-based method (Qmetrics Technologies, Rochester, NY, USA) [45] for a previous study [85]. OAI2 consisted of 88 DESS images acquired at baseline and at 1-year followup. Ground truth segmentations were computed using an active appearance model (imorphics, Manchester, UK) [42]. Finally, inHouse contained 4 images acquired at Stanford University using DESS and CubeQuant protocols. For clarity in the following, OAI1 will be split in OAI1-DESS and OAI1-T2, OAI2 in OAI2-BL (baseline) and OAI2-FU (followup), and inHouse in inHouse-DESS and inHouse-CQ (CubeQuant). Details of the acquisition parameters are in Table 2I.

Table 2. Datasets used to evaluate pyKNEER. *I. Acquisition parameters:* Parameters of the protocols used to acquire the images. Images of OAI1-DESS, OAI2-BL, and OAI2-FU were acquired with the same DESS protocol, consisting of 2 echos, although only their average was available (◊). Images of one subject of the dataset OAI1 had different slice spacing and thickness (*). Data queries to obtain acquisition parameters are in a Jupyter notebook ([here](#)). The original identification numbers (IDs) of the OAI images are in a Jupyter notebook used as a metafile ([here](#)). *II. Ground truth segmentation:* The datasets OAI1 and OAI2 have ground truth segmentations. They differ for computational method, segmented anatomy, and label type. *III. Experimental results:* Details of the steps in pyKNEER for each dataset. Full circle (●) indicates processing of the dataset, while empty circle (◊) indicates processing of ground truth segmentations. The numbers in “Find reference” indicated the ID of the seed images used in the convergence study. Links are to the executed notebooks on GitHub.

Dataset	OAI1-DESS	OAI1-T ₂	OAI2-BL	OAI2-FU	inHouse-DESS	inHouse-CQ
Number of subjects	19	19	88	88	4	4
I. Acquisition parameters						
Acquisition protocol	DESS	T ₂ -w	DESS		DESS	CubeQuant
Acquisition plane	sagittal	sagittal	sagittal		sagittal	sagittal
Number of images in series	2 (1 available)◊	7	2 (1 available)◊		2	4
In-plane spacing [mm]	0.3646 x 0.3646 (0.4270 x 0.4270)*	0.3125 x 0.3125 (0.4296 x 0.4296)*	0.3646 x 0.3646		0.3125 x 0.3125	0.3125 x 0.3125
Slice thickness [mm]	0.7 (0.75)*	3 (3.5)*	0.7		1.5	3
Echo time (TE) [ms]	4.7	10, 20, 30, 40, 50, 60, 70	4.7		42.52	-
Spin-lock time (TSL) [ms]	-	-	-		-	1, 10, 30, 60
Repetition time (TR) [ms]	16.32	2700 (2900)*	16.32		25	1302
Flip angle [°]	25	180	25		30	90
II. Ground truth segmentation						
Method	atlas-based		active models		-	-
Anatomy	femur, femoral cartilage		femoral cartilage		-	-
Type	mask		contour		-	-
III. Experimental results						
Image number in series	1	1	2-7	1	1	1
Preprocessing						
Spatial standardization	●	●	●	●	●	●
Intensity standardization	●	●	-	●	-	●
Segmentation						
Find reference	4, 8, 10, 13, 16	-	-	-	-	-
Intersubject	●	-	-	●	-	●
Longitudinal	-	-	-	-	●	-
Multimodal	-	●	-	-	-	●
Segmentation quality						
Dice coefficient	●	●	-	●	●	-
Analysis						
Morphology	●◊	●◊	-	●◊	●◊	●
Relaxation	-	-	●◊	-	-	●

<https://doi.org/10.1371/journal.pone.0226501.t002>

Results

We preprocessed, segmented, and analyzed all the dataset using different options in pyKNEER, according to dataset characteristics and availability of ground truth segmentation (link to dataset [here](#)) (Table 2III).

Preprocessing. We executed spatial preprocessing for all images of the datasets and intensity preprocessing only for the images directly involved in segmentation.

Finding reference. We selected the reference mask from the dataset OAI1-DESS because of the availability of ground truth segmentations of the femur, which is the volume where we compare velocity fields. We picked 5 images as initial reference for our parallel investigation

using a python random function (random seed = 4; see [code](#)). For all the studies, we found the reference as the subject whose vector field distance to the average vector field was the minimum (subject ID = 9).

Segmenting intersubject, longitudinal, and multimodal images. We segmented images from OAI1-DESS, OAI2-BL, and inHouse-DESS as new subjects. Segmentation failure were 1 for OAI1-DESS (ID = 6, DSC = 0.05, ASD = 28.14mm; see [code](#)), 3 for OAI2-BL (ID = 6, DSC = 0.01, ASD = 60.98mm; ID = 24, DSC = 0.34, ASD = 5.49mm; ID = 31, DSC = 0.57, ASD = 8.18mm; see [code](#)), and none inHouse-DESS (see [code](#)). We excluded the failed registrations from the following analysis of segmentation quality, cartilage morphology, and cartilage relaxometry. We segmented the first acquisition of OAI1-T2 images (see [code](#)) and inHouse-CQ images (see [code](#)) using the multimodal option in pyKNEEr, and OAI2-FU images (see [code](#)) using the longitudinal option.

Segmentation quality. We evaluated segmentation quality for the datasets OAI1 and OAI2 because they had ground truth segmentations of femoral cartilage. The Dice similarity coefficients were 0.86 ± 0.02 (mean \pm standard deviation) for OAI1-DESS, 0.76 ± 0.04 for OAI1-T2, 0.73 ± 0.04 for OAI2-BL, and 0.72 ± 0.04 for OAI2-FU ([Fig 4a](#)). The ASD measures were $0.60 \pm 0.12\text{mm}$ for OAI1-DESS, $0.54 \pm 0.11\text{mm}$ for OAI1-T2, $1.33 \pm 0.33\text{mm}$ for OAI2-BL, and $1.38 \pm 0.33\text{mm}$ for OAI2-FU (see [code](#)).

Morphology. We calculated cartilage thickness and volume for all datasets, including ground truth segmentations. We computed correlations of cartilage thickness calculated from pyKNEEr's segmentation and ground truth segmentation, and we found that Pearson coefficients were 0.958 for OAI1-DESS, 0.668 for OAI1-T2, 0.654 for OAI2-BL, and 0.667 for OAI2-FU ([Fig 4b](#)). Similarly, we computed correlations for cartilage volume, and we found that Pearson coefficients were 0.983 for OAI1-DESS, 0.847 for OAI1-T2, 0.891 for OAI2-BL, and 0.885 for OAI2-FU ([Fig 4c](#)) (see [code](#)).

Relaxometry. Before calculating relaxometry maps for OAI1-T2, we rigidly registered the images with shortest TE to the image with longest TE. Similarly, before calculating $T_{1\rho}$ maps for inHouse-CQ, we rigidly registered the images with shortest TSL to the image with longest TSL. Then, we calculated T_2 maps for OAI-T2 images extracting values in pyKNEEr's masks (see [code](#)) and ground truth masks (see [code](#)), and we compared them, obtaining a Pearson's coefficient of 0.513 ([Fig 4d](#)) (see [code](#)). Finally, we computed relaxometry maps using exponential fitting for inHouse-CQ (see [code](#)) and EPG modeling for inHouse-DESS (see [code](#)) to show feasibility of the methods.

Discussion

To test possible reproducible workflows with pyKNEEr, we ran experiments with three different datasets. Image preprocessing was successful in all cases, while image segmentation failed in 4 cases. Average DSC were 0.81 for dataset OAI1 and 0.73 for dataset OAI2. These values are in the range of published values, as depicted in the literature review visualization of DSC ([Fig 5](#)). Discrepancies of DSC between OAI1 and OAI2 can be due to the different characteristics of ground truth segmentations. OAI1 ground truth segmentations were created using an atlas-based method with $DSC = 0.88$ [45] (see "TamezPena (2012)" in [Fig 5](#)), whereas OAI2 ground truth segmentations were created using an active appearance model with $DSC = 0.78$ [42] (see "Vincent (2011)" in [Fig 5](#)). In addition, to calculate DSC we transformed OAI2 ground truth segmentations from contours to volumetric masks, potentially adding discretization error. Quality of segmentation had a direct impact on morphology and relaxometry analysis. Pearson's coefficient was higher for cartilage volume than cartilage thickness, suggesting higher preservation of volume, and it was low for T_2 relaxation times,

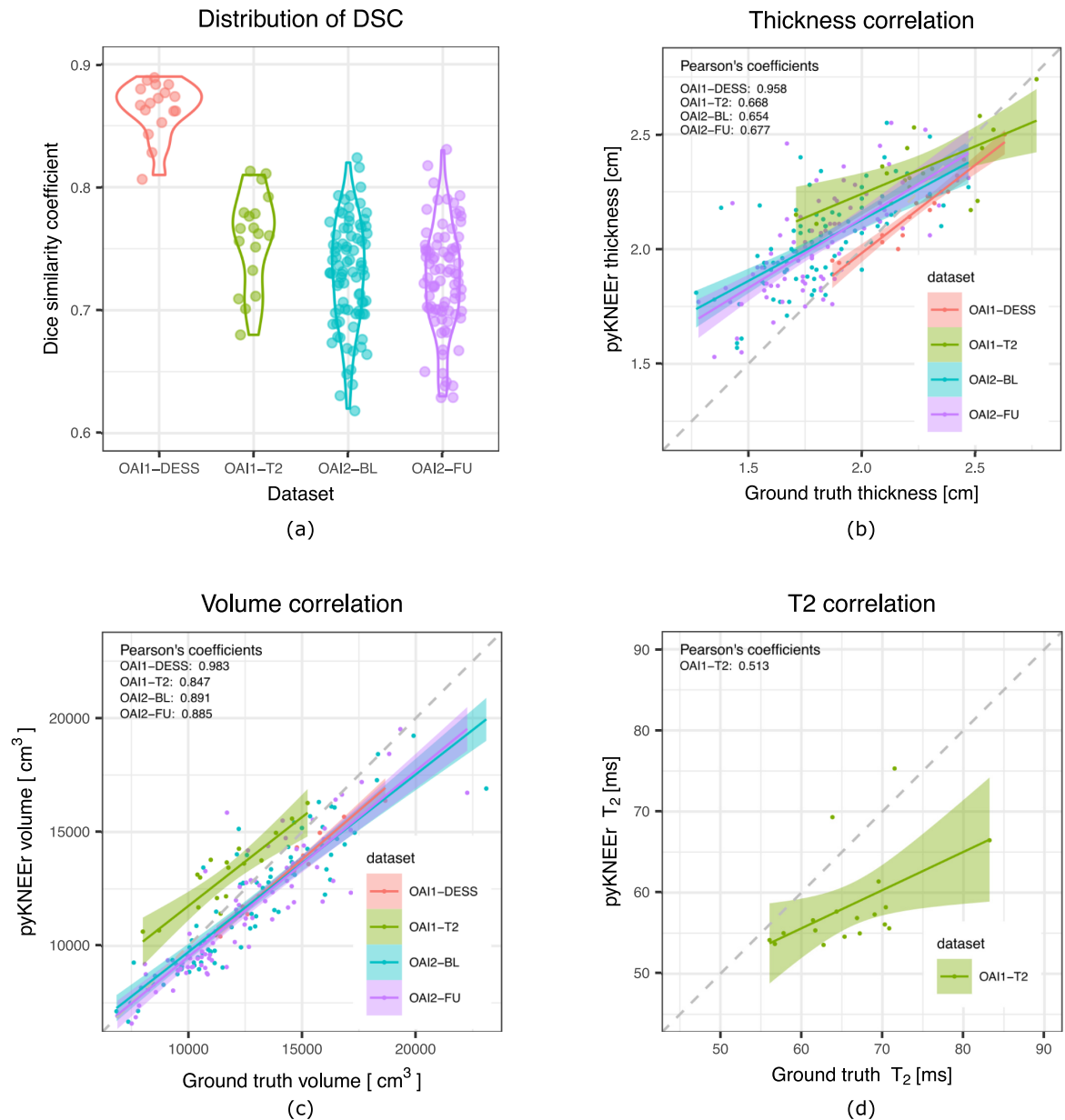


Fig 4. Results for the datasets OAI1-DESS (red), OAI1-T2 (green), OAI2-BL (cyan), and OAI2-FU (purple). (a) Violin plots describing the distribution of the DSC within each dataset. The dots represent DSC values spread around the y-axis to avoid visualization overlap. (b-d) Correlation between measurements derived from ground truth segmentations and pyKNEER's segmentations, i.e. cartilage thickness (b), cartilage volume (c), and T₂ maps (d). (See [data](#), [code](#), [computational environment](#)).

<https://doi.org/10.1371/journal.pone.0226501.g004>

suggesting higher dependency on segmentation quality for intensity-based measurements. Finally, regression lines show that measurements from pyKNEER segmentation overestimated small thicknesses and underestimated large volumes and T₂ values (Fig 4). We implemented atlas-based segmentation because it has the advantage to provide byproducts for further analysis. Image correspondences established during the registration step can be used for intersubject and longitudinal comparison of cartilage thickness and relaxation times, and voxel-based morphometry and relaxometry [44].

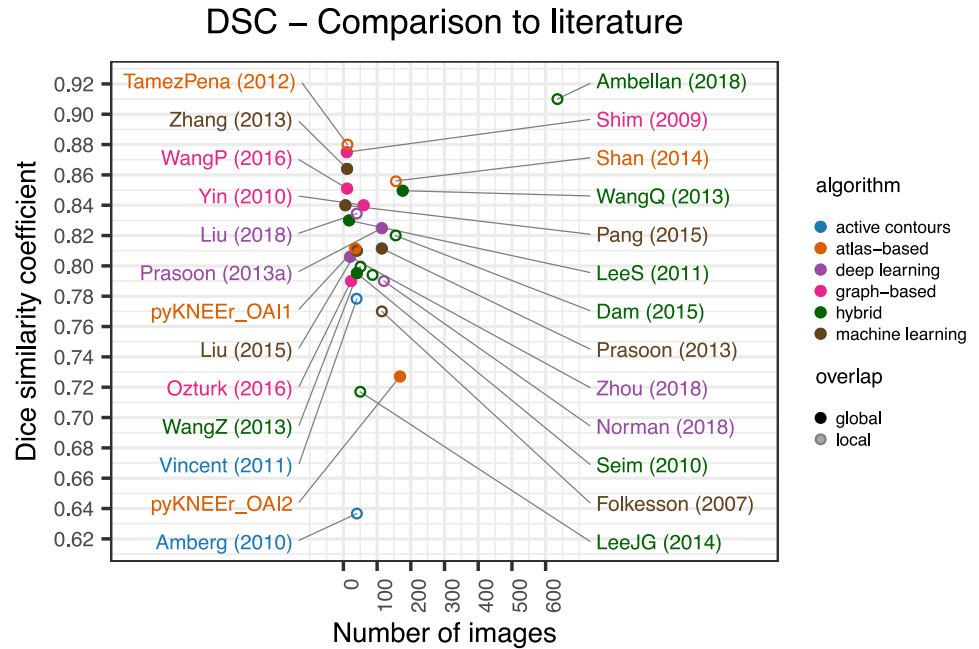


Fig 5. Performances of the segmentation module of pyKNEEr, compared with 24 studies in literature that report it. Full dots represents studies where DSCs were calculated on the whole mask, whereas empty dots represent studies where DSCs were calculated in specific parts of the cartilage, e.g. the weight-bearing area [28]. (See [data](#), [code](#), [executable environment](#)).

<https://doi.org/10.1371/journal.pone.0226501.g005>

We designed pyKNEEr to facilitate transparent research on femoral knee analysis from MR images. Traditionally, medical image analysis workflows are in ITK, VTK, and Qt, requiring advanced computational skills in C++ to build, run, and extend code. We wrote pyKNEEr in python because of its ease of use, compatibility with various operating systems, and extensive computing support through packages and open code. As a consequence, pyKNEEr can be easily installed as a package in the python environment and does not require advanced programming skills. In addition, we used Jupyter notebooks as a user-interface because of their ease of use, literate computing approach [86], versatility for publications, and sharing among researchers. In pyKNEEr, Jupyter notebooks can be simply downloaded from our [GitHub repository](#) to a local folder. Researchers have to input an image list and optionally set a few variables, and after automatic execution of the notebook, they directly obtain visualizations, graphs, and tables for further analysis. In addition, researchers can link the executed notebook directly to papers (similarly to [Table 2](#)) and thus create an interactive publication with reproducible analysis. In the medical image analysis community, other examples of combined use of python and Jupyter notebooks are mainly for educational and general research purpose (e.g. [SimpleITK notebooks](#) [87]), while usage of python as a programming language is rapidly gaining popularity in neuroimaging (e.g. [Nipype](#) [35]).

Several extensions of pyKNEEr could be imagined, due to the modularity of its structure. In the segmentation module, the current notebook implementing atlas-based segmentation ([segmentation.ipynb](#)) could be substituted by notebooks with hybrid machine or deep learning algorithms, which can provide higher DSC [55] ([Fig 5](#)). In the morphology module ([morphology.ipynb](#)), the code structure already includes a flag (`thickness_algo`) to integrate additional algorithms for cartilage thickness, such as surface normal vectors, local thickness, and potential field lines [81]. Finally, new notebooks could be added to the workflow to segment and analyze more knee tissues, such as tibial cartilage, patellar cartilage, and the

menisci. Extensions will require a limited amount of effort because of the popularity and ease of `python`, the free availability of a large number of programming packages, and the flexibility of Jupyter notebooks [87]. In addition, standardized file format and computational environment will facilitate comparison of findings and performances of new algorithms.

In conclusion, we have presented `pyKNEEr`, an image analysis workflow for open and reproducible research on femoral knee cartilage. We validated `pyKNEEr` with three experiments, where we tested preprocessing, segmentation, and analysis. Through our validation test, we presented a possible modality of conducting open and reproducible research with `pyKNEEr`. Finally, in our paper we provide links to executed notebooks and executable environments for computational reproducibility of our results and analysis.

Acknowledgments

We would like to thank Uche Monu for her insights on cartilage segmentation, Bragi Sveinsson for sharing his code on EPG modeling, Valentina Mazzoli for her insight on relaxometry, Christof Seiler for software engineering support, Amy Silder, Julie Kolesar, and Scott Uhlrich for beta testing, Susan Holmes, Felix Ambellan, and Yolanda Gil for their feedbacks on the first preprint version of this paper, and all the researchers that make their code open-source, on whose effort we could create `pyKNEEr`.

Author Contributions

Conceptualization: Serena Bonaretti.

Data curation: Serena Bonaretti.

Formal analysis: Serena Bonaretti.

Funding acquisition: Gary S. Beaupre.

Methodology: Serena Bonaretti.

Project administration: Gary S. Beaupre.

Resources: Garry E. Gold, Gary S. Beaupre.

Software: Serena Bonaretti.

Supervision: Garry E. Gold, Gary S. Beaupre.

Validation: Serena Bonaretti.

Visualization: Serena Bonaretti.

Writing – original draft: Serena Bonaretti.

Writing – review & editing: Serena Bonaretti, Garry E. Gold, Gary S. Beaupre.

References

1. Collins FS, Tabak LA. NIH plans to enhance reproducibility. *Nature*. 2014; 505(7485):612–613. PMID: [24482835](https://pubmed.ncbi.nlm.nih.gov/24482835/)
2. Commission TE. Commission recommendations of 17 July 2012 on access to and preservation of scientific information; 2012.
3. Woelfle M, Olliaro P, Todd MH. Open science is a research accelerator. *Nature Chemistry*. 2011; 3(10):745–748. <https://doi.org/10.1038/nchem.1149> PMID: [21941234](https://pubmed.ncbi.nlm.nih.gov/21941234/)
4. Bollen K, Cacioppo JT, Kaplan R, Krosnick J, Olds JL. Social, behavioral, and economic sciences perspectives on robust and reliable science; 2015.

5. Sandve GK, Nekrutenko A, Taylor J, Hovig E. Ten simple rules for reproducible computational research. *PLoS Computational Biology*. 2013; 9(10):1–4. <https://doi.org/10.1371/journal.pcbi.1003285>
6. Rule A, Birmingham A, Zuniga C, Altintas I, Huang SC, Knight R, et al. Ten simple rules for reproducible research in Jupyter notebooks. *arXiv:181008055*. 2018.
7. Prlić A, Procter JB. Ten simple rules for the open development of scientific software. *PLoS Computational Biology*. 2012; 8(12):8–10.
8. Donoho DL, Maleki A, Rahman IU, Shahram M, Stodden V. Reproducible research in computational harmonic analysis. *Comput Sci Eng*. 2009; 11(1):8–18. <https://doi.org/10.1109/MCSE.2009.15>
9. Munafò MR, Nosek BA, Bishop DVM, Button KS, Chambers CD, Percie N, et al. A manifesto for reproducible science. *Nature Publishing Group*. 2017; 1(January):1–9.
10. Pérez F, Granger BE. IPython: a System for Interactive Scientific Computing. *Computing in Science and Engineering*. 2007; 9(3):21–29. <https://doi.org/10.1109/MCSE.2007.53>
11. Kluyver T, Ragan-Kelley B, Pérez F, Granger B, Bussonnier M, Frederic J, et al. Jupyter Notebooks—a publishing format for reproducible computational workflows. In: Loizides F, Schmidt B, editors. *Positioning and Power in Academic Publishing: Players, Agents and Agendas*. IOS Press; 2016. p. 87–90.
12. R Core Team. R: A Language and Environment for Statistical Computing; 2013. Available from: <http://www.R-project.org/>.
13. Jupyter P, Bussonnier M, Forde J, Freeman J, Granger B, Head T, et al. Binder 2.0—Reproducible, interactive, sharable environments for science at scale. In: *Proceedings of the 17th Python in Science Conference*. Scipy; 2018. p. 113–120. Available from: https://conference.scipy.org/proceedings/scipy2018/project/_jupyter.html.
14. Claerhout JF, Karrenbach M. Electronic documents give reproducible research a new meaning. *SEG Technical Program Expanded Abstracts* 1992. 1992; 11(1):601–604. <https://doi.org/10.1190/1.1822162>
15. Gil Y, David CH, Demir I, Essawy BT, Fulweiler RW, Goodall JL, et al. Toward the geoscience paper of the future: Best practices for documenting and sharing research from data to software to provenance. *Earth and Space Science*. 2016; 3(10):388–415. <https://doi.org/10.1002/2015EA000136>
16. Gundersen OE, Gil Y, Aha DW. On reproducible AI: Towards reproducible research, open science, and digital scholarship in AI publications. *AI Magazine*. 2017; 39(3):56–68. <https://doi.org/10.1609/aimag.v39i3.2816>
17. Poldrack RA, Baker CI, Durnez J, Gorgolewski KJ, Matthews PM, Munafò MR, et al. Scanning the horizon: Towards transparent and reproducible neuroimaging research. *Nature Reviews Neuroscience*. 2017; 18(2):115–126. <https://doi.org/10.1038/nrn.2016.167> PMID: 28053326
18. Hafezi-Nejad N, Demehri S, Guermazi A, Carrino JA. Osteoarthritis year in review 2017: updates on imaging advancements. *Osteoarthritis and Cartilage*. 2018; 26(3):341–349. <https://doi.org/10.1016/j.joca.2018.01.007> PMID: 29330100
19. Woolf AD, Pfleger B. Burden of major musculoskeletal conditions. *Bulletin of the World Health Organization*. 2003; 81(9):646–656. PMID: 14710506
20. Palazzo C, Ravaud JF, Papeard A, Ravaud P, Poiraudou S. The burden of musculoskeletal conditions. *PLoS ONE*. 2014; 9(3):e90633. <https://doi.org/10.1371/journal.pone.0090633> PMID: 24595187
21. Martel-Pelletier J, Barr AJ, Cicuttini FM, Conaghan PG, Cooper C, Goldring MB, et al. Osteoarthritis. *Nature Reviews Disease Primers*. 2016; 2:1–18. <https://doi.org/10.1038/nrdp.2016.72>
22. Hunter DJ, Schofield D, Callander E. The individual and socioeconomic impact of osteoarthritis. *Nature Reviews Rheumatology*. 2014; 10(7):437–441. <https://doi.org/10.1038/nrrheum.2014.44> PMID: 24662640
23. Eckstein F, Boudreau R, Wang Z, Hannon MJ, Duryea J, Wirth W, et al. Comparison of radiographic joint space width and magnetic resonance imaging for prediction of knee replacement: A longitudinal case-control study from the Osteoarthritis Initiative. *European Radiology*. 2016; 26(6):1942–1951. <https://doi.org/10.1007/s00330-015-3977-8> PMID: 26376884
24. Schaefer LF, Sury M, Yin M, Jamieson S, Donnell I, Smith SE, et al. Quantitative measurement of medial femoral knee cartilage volume – analysis of the OA Biomarkers Consortium FNIH Study cohort. *Osteoarthritis and Cartilage*. 2017; 25(7):1107–1113. <https://doi.org/10.1016/j.joca.2017.01.010> PMID: 28153788
25. Li X, Benjamin Ma C, Link TM, Castillo DD, Blumenkrantz G, Lozano J, et al. In vivo T1ρ and T2 mapping of articular cartilage in osteoarthritis of the knee using 3 T MRI. *Osteoarthritis and Cartilage*. 2007; 15(7):789–797. <https://doi.org/10.1016/j.joca.2007.01.011> PMID: 17307365
26. Monu UD, Jordan CD, Samuelson BL, Hargreaves BA, Gold GE, McWalter EJ. Cluster analysis of quantitative MRI T2 and T1rho relaxation times of cartilage identifies differences between healthy and

- ACL-injured individuals at 3T. *Osteoarthritis and Cartilage*. 2017; 25(4):513–520. <https://doi.org/10.1016/j.joca.2016.09.015> PMID: 27720806
27. Liukkonen MK, Mononen ME, Tanska P, Saarakkala S, Nieminen MT, Korhonen RK. Application of a semi-automatic cartilage segmentation method for biomechanical modeling of the knee joint. *Computer Methods in Biomechanics and Biomedical Engineering*. 2017; 20(13):1453–1463. <https://doi.org/10.1080/10255842.2017.1375477> PMID: 28895760
 28. Heimann T, Morrison B. Segmentation of knee images: A grand challenge. *Proc Medical Image Analysis for the Clinic: A Grand Challenge Beijing, China*. 2010; p. 207–214.
 29. Pedoia V, Majumdar S, Link TM. Segmentation of joint and musculoskeletal tissue in the study of arthritis. *Magnetic Resonance Materials in Physics, Biology and Medicine*. 2016. <https://doi.org/10.1007/s10334-016-0532-9>
 30. Zhang B, Zhang Y, Cheng HD, Xian M, Gai S, Cheng O, et al. Computer-aided knee joint magnetic resonance image segmentation—A survey. *bioRxiv* = 1180204894v1. 2018.
 31. Wang Q, Wu D, Lu L, Liu M, Boyer KL, Zhou SK. Semantic context forests for learning-based knee cartilage segmentation in 3D MR images. In: Springer, editor. *Medical Computer Vision. Large Data in Medical Imaging Lecture Notes in Computer Science*. New York; 2013. p. 105–115. Available from: <http://arxiv.org/abs/1307.2965> http://dx.doi.org/10.1007/978-3-319-05530-5_11.
 32. Shan L, Zach C, Charles C, Niethammer M. Automatic atlas-based three-label cartilage segmentation from MR knee images. *Medical Image Analysis*. 2014; 18(7):1233–1246. <https://doi.org/10.1016/j.media.2014.05.008> PMID: 25128683
 33. Avants BB, Tustison NJ, Song G, Cook PA, Klein A, Gee JC. A reproducible evaluation of ANTs similarity metric performance in brain image registration. *NeuroImage*. 2011; 54(3):2033–2044. <https://doi.org/10.1016/j.neuroimage.2010.09.025> PMID: 20851191
 34. Fischl B, Salat DH, Busa E, Albert M, Dieterich M, Haselgrove C, et al. Whole brain segmentation: automated labeling of neuroanatomical structures in the human brain. *Neuron*. 2002; 33(3):341–355. [https://doi.org/10.1016/s0896-6273\(02\)00569-x](https://doi.org/10.1016/s0896-6273(02)00569-x) PMID: 11832223
 35. Gorgolewski K, Burns CD, Madison C, Clark D, Halchenko YO, Waskom ML, et al. Nipype: A flexible, lightweight and extensible neuroimaging data processing framework in python. *Frontiers in Neuroinformatics*. 2011; 5(August). <https://doi.org/10.3389/fninf.2011.00013> PMID: 21897815
 36. Van Erp TGM, Hibar DP, Rasmussen JM, Glahn DC, Pearlson GD, Andreassen OA, et al. Subcortical brain volume abnormalities in 2028 individuals with schizophrenia and 2540 healthy controls via the ENIGMA consortium. *Molecular Psychiatry*. 2016; 21(4):547–553. <https://doi.org/10.1038/mp.2015.63> PMID: 26033243
 37. Lawson GM, Duda JT, Avants BB, Wu J, Farah MJ. Associations between children's socioeconomic status and prefrontal cortical thickness. *Developmental Science*. 2013; 16(5):641–652. <https://doi.org/10.1111/desc.12096> PMID: 24033570
 38. Doehrmann O, Ghosh SS, Polli FE, Reynolds GO, Horn F, Keshavan A, et al. Predicting treatment response in social anxiety disorder from functional magnetic resonance imaging. *Archives of General Psychiatry*. 2013; 70(1):87–97.
 39. Amberg M, Luthi M, Vetter T. Fully automated segmentation of the knee using local deformation-model fitting. In: *MICCAI 2010 Workshop Medical Image Analysis for the Clinic—A Grand Challenge (SKI10)*; 2010. p. 251–260. Available from: <http://www.diagnijmegen.nl/~bram/grandchallenge2010/251.pdf>.
 40. Carballido-Gamio J, Bauer JS, Stahl R, Lee KY, Krause S, Link TM, et al. Inter-subject comparison of MRI knee cartilage thickness. *Medical Image Analysis*. 2008; 12(2):120–135. <https://doi.org/10.1016/j.media.2007.08.002> PMID: 17923429
 41. Solloway S, Hutchinson CE, Waterton JC, Taylor CJ. The use of active shape models for making thickness measurements of articular cartilage from MR images. *Magnetic resonance in medicine*. 1997; 37(6):943–952. <https://doi.org/10.1002/mrm.1910370620> PMID: 9178247
 42. Vincent G, Wolstenholme C, Scott I, Bowes M. Fully automatic segmentation of the knee joint using active appearance models. *MICCAI 2010 Workshop Medical Image Analysis for the Clinic—A Grand Challenge (SKI10)*. 2011.
 43. Williams TG, Holmes AP, Waterton JC, Maclewickz RA, Hutchinson CE, Moots RJ, et al. Anatomically corresponded regional analysis of cartilage in asymptomatic and osteoarthritic knees by statistical shape modelling of the bone. *IEEE Transactions on Medical Imaging*. 2010; 29(8):1541–1559. <https://doi.org/10.1109/TMI.2010.2047653> PMID: 20378463
 44. Pedoia V, Li X, Su F, Calixto N, Majumdar S. Fully automatic analysis of the knee articular cartilage T_{1ρ} relaxation time using voxel-based relaxometry. *Journal of Magnetic Resonance Imaging*. 2015; 43:970–980. <https://doi.org/10.1002/jmri.25065> PMID: 26443990

45. Tamez-Peña JG, Farber J, González PC, Schreyer E, Schneider E, Totterman S. Unsupervised segmentation and quantification of anatomical knee features: Data from the osteoarthritis initiative. *IEEE Transactions on Biomedical Engineering*. 2012; 59(4):1177–1186. <https://doi.org/10.1109/TBME.2012.2186612> PMID: 22318477
46. Liu F, Zhou Z, Jang H, McMillan A, Kijowski R. Deep convolutional neural network and 3D deformable approach for tissue segmentation in musculoskeletal magnetic resonance imaging. *Magnetic Resonance in Medicine*. 2018; 79:2379–2391. <https://doi.org/10.1002/mrm.26841> PMID: 28733975
47. Norman B, Pedoia V, Majumdar S. Use of 2D U-Net convolutional neural networks for automated cartilage and meniscus segmentation of knee MR imaging data to determine relaxometry and morphometry. *Radiology*. 2018; 288(1):177–185. <https://doi.org/10.1148/radiol.2018172322> PMID: 29584598
48. Prasoon A, Petersen K, Igel C, Lauze F, Dam E, Nielsen M. Deep feature learning for knee cartilage segmentation using a triplanar convolutional neural network. In: *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2013*. 09; 2013. p. 246–253.
49. Zhou Z, Zhao G, Kijowski R, Liu F. Deep convolutional neural network for segmentation of knee joint anatomy. *Magnetic Resonance in Medicine*. 2018; 80(6):2759–2770. <https://doi.org/10.1002/mrm.27229> PMID: 29774599
50. Bae KT, Shim H, Tao C, Chang S, Wang JH, Boudreau R, et al. Intra- and inter-observer reproducibility of volume measurement of knee cartilage segmented from the OAI MR image set using a novel semi-automated segmentation method. *Osteoarthritis and Cartilage*. 2009; 17(12):1589–1597. <https://doi.org/10.1016/j.joca.2009.06.003> PMID: 19577672
51. Öztürk CN, Albayrak S. Automatic segmentation of cartilage in high-field magnetic resonance images of the knee joint with an improved voxel-classification-driven region-growing algorithm using vicinity-correlated subsampling. *Computers in Biology and Medicine*. 2016; 72:90–107. <https://doi.org/10.1016/j.compbiomed.2016.03.011> PMID: 27017069
52. Shim H, Chang S, Tao C, Wang JH, Kwok CK, Bae KT. Knee cartilage: Efficient and reproducible segmentation on high-spatial-resolution MR images with the semiautomated graph-cut algorithm method. *Radiology*. 2009; 251(2):548–556. <https://doi.org/10.1148/radiol.2512081332> PMID: 19401579
53. Wang P, He X, Li Y, Zhu X, Chen W, Qiu M. Automatic knee cartilage segmentation using multi-feature support vector machine and elastic region growing for magnetic resonance images. *Journal of Medical Imaging and Health Informatics*. 2016; 6(4):948–956. <https://doi.org/10.1166/jmihi.2016.1748>
54. Yin Y, Zhang X, Williams R, Wu X, Anderson D, Sonka M. LOGISMOS—Layered optimal graph image segmentation of multiple objects and surfaces: Cartilage segmentation in the knee joint. *IEEE Trans Med Imaging*. 2010; 29(12):2023–2037. <https://doi.org/10.1109/TMI.2010.2058861> PMID: 20643602
55. Ambellan F, Tack A, Ehke M, Zachow S. Automated segmentation of knee bone and cartilage combining statistical shape knowledge and convolutional neural networks: Data from the Osteoarthritis Initiative. *Medical Image Analysis*. 2018.
56. Dam EB, Lillholm M, Marques J, Nielsen M. Automatic Segmentation of High- and Low-Field Knee MRIs Using Knee Image Quantification with Data from the Osteoarthritis Initiative. *Journal of Medical Imaging*. 2015; 2(2):1–13. <https://doi.org/10.1117/1.JMI.2.2.024001>
57. Lee JG, Gumus S, Moon CH, Kwok CK, Bae KT. Fully automated segmentation of cartilage from the MR images of knee using a multi-atlas and local structural analysis method. *Medical Physics*. 2014; 41(9):092303. <https://doi.org/10.1118/1.4893533> PMID: 25186408
58. Lee S, Park SH, Shim H, Yun ID, Lee SU. Optimization of local shape and appearance probabilities for segmentation of knee cartilage in 3-D MR images. *Computer Vision and Image Understanding*. 2011; 115(12):1710–1720. <https://doi.org/10.1016/j.cviu.2011.05.014>
59. Seim H, Kainmueller D, Lamecker H, Bindernagel M, Malinowski J, Zachow S. Model-based auto-segmentation of knee bones and cartilage in MRI data. In: *Proc. Medical Image Analysis for the Clinic: A Grand Challenge*. Beijing, China; 2010. p. 215–223. Available from: <http://www.zib.de/visual/medicalhttp://www.diagnijmegen.nl/~jbram/grandchallenge2010/215.pdf>.
60. Wang Z, Donoghue C, Rueckert D. Patch-based segmentation without registration: Application to knee MRI. In: *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*. vol. 8184 LNCS; 2013. p. 98–105.
61. Folkesson J, Dam EB, Olsen OF, Pettersen PC, Christiansen C. Segmenting articular cartilage automatically using a voxel classification approach. *IEEE Transactions on Medical Imaging*. 2007; 26(1):106–115. <https://doi.org/10.1109/TMI.2006.886808> PMID: 17243589
62. Liu Q, Wang Q, Zhang L, Gao Y, Shen D. Multi-atlas context forests for knee MR image segmentation. In: *International Workshop on Machine Learning in Medical Imaging*. June 2016; 2015. p. 186–193. Available from: <http://arxiv.org/abs/1701.05616>.

63. Pang J, Li PY, Qiu M, Chen W, Qiao L. Automatic articular cartilage segmentation based on pattern recognition from knee MRI images. *Journal of Digital Imaging*. 2015; 28(6):695–703. <https://doi.org/10.1007/s10278-015-9780-x> PMID: 25700618
64. Prasoon A, Igel C, Loog M, Lauze F, Dam EB, Nielsen M. Femoral cartilage segmentation in knee MRI scans using two stage voxel classification. *Proceedings of the Annual International Conference of the IEEE Engineering in Medicine and Biology Society, EMBS*. 2013; p. 5469–5472.
65. Zhang K, Lu W, Marzilano P. Automatic knee cartilage segmentation from multi-contrast MR images using support vector machine classification with spatial dependencies. *Magnetic Resonance Imaging*. 2013; 31(10):1731–1743. <https://doi.org/10.1016/j.mri.2013.06.005> PMID: 23867282
66. Wickham H. *ggplot2: Elegant graphics for data analysis*. Springer-Verlag New York; 2016. Available from: <http://ggplot2.org>.
67. Wilkinson L. *The grammar of graphics (statistics and computing)*. Berlin, Heidelberg: Springer-Verlag; 2005.
68. Peterfy CG, Schneider E, Nevitt M. The osteoarthritis initiative: report on the design rationale for the magnetic resonance imaging protocol for the knee. *Osteoarthritis and Cartilage*. 2008; 16(12):1433–1441. <https://doi.org/10.1016/j.joca.2008.06.016> PMID: 18786841
69. Luger R, Foreman-Mackey EAD, Fleming DP, Lustig-Yaeger J, Deitrick R. STARRY: Analytic occultation light curves. arXiv:181006559v1 [astro-phIM]. 2018.
70. Jiménez RC, Kuzak M, Alhamdoosh M, Barker M, Batut B, Borg M, et al. Four simple recommendations to encourage best practices in research software. *F1000Research*. 2017; 6:876. <https://doi.org/10.12688/f1000research.11407.1>
71. Oliphant TE. *A guide to NumPy*. vol. 1. Trelgol Publishing USA; 2006.
72. van der Walt S, Colbert SC, Varoquaux G. The NumPy Array: A Structure for Efficient Numerical Computation. *Computing in Science Engineering*. 2011; 13(2):22–30. <https://doi.org/10.1109/MCSE.2011.37>
73. McKinney W. *Data Structures for Statistical Computing in Python*. In: van der Walt S, Millman J, editors. *Proceedings of the 9th Python in Science Conference*; 2010. p. 51–56.
74. Hunter JD. Matplotlib: A 2D graphics environment. *Computing in Science & Engineering*. 2007; 9(3):90–95. <https://doi.org/10.1109/MCSE.2007.55>
75. Lowekamp BC, Chen DT, Ibáñez L, Blezek D. The Design of SimpleITK. *Frontiers in Neuroinformatics*. 2013; 7(December):1–14.
76. Klein S, Staring M, Murphy K, Viergever MA, Pluim J. elastix: A Toolbox for intensity-based medical image registration. *IEEE Transactions on Medical Imaging*. 2010; 29(1):196–205. <https://doi.org/10.1109/TMI.2009.2035616> PMID: 19923044
77. Esteban O, Markiewicz CJ, Blair RW, Moodie CA, Ayse I, Erramuzpe A, et al. FMRIPrep: A robust pre-processing pipeline for functional MRI. *Nature Methods*. 2019; 16(January):1–20.
78. Sled JG, Zijdenbos AP, Evans AC. A nonparametric method for automatic correction of intensity nonuniformity in MRI data. *IEEE Transactions on Medical Imaging*. 1998; 17(1):87–97. <https://doi.org/10.1109/42.668698> PMID: 9617910
79. Sethian J. *Level set methods and fast marching methods*. Cambridge Press; 1999.
80. Dice L. Measures of the amount of ecologic association between species. *Ecology*. 1945; p. 297–302. <https://doi.org/10.2307/1932409>
81. Maier J, Black M, Bonaretti S, Bier B, Eskofier B, Choi JH, et al. Comparison of different approaches for measuring tibial cartilage thickness. *Journal of integrative bioinformatics*. 2017; 14(2):1–10. <https://doi.org/10.1515/jib-2017-0015>
82. Chen W. Errors in quantitative T1rho imaging and the correction methods. *Quantitative imaging in medicine and surgery*. 2015; 5(4):583–91. <https://doi.org/10.3978/j.issn.2223-4292.2015.08.05> PMID: 26435922
83. van Tiel J, Kotek G, Reijman M, Bos PK, Bron EE, Klein S, et al. Is T1ρ mapping an alternative to delayed gadolinium-enhanced MR imaging of cartilage in the assessment of sulphated glycosaminoglycan content in human osteoarthritic knees? An in vivo validation study. *Radiology*. 2016; 279(2):523–531. <https://doi.org/10.1148/radiol.2015150693> PMID: 26588020
84. Sveinsson B, Chaudhari AS, Gold GE, Hargreaves BA. A simple analytic method for estimating T2 in the knee from DESS. *Magnetic Resonance Imaging*. 2016; 38:63–70. <https://doi.org/10.1016/j.mri.2016.12.018> PMID: 28017730
85. Hallaj E, Hastie TJ, Gold GE, Delp SL. Physical activity is associated with changes in knee cartilage microstructure. *Osteoarthritis and Cartilage*. 2018; 26(6):770–774. <https://doi.org/10.1016/j.joca.2018.03.009> PMID: 29605382

86. Millman KJ, Pérez F. Developing Open Source Practices. In: Stodden V, Leisch F, Peng RD, editors. *Implementing Reproducible Research*. Taylor & Francis; 2014. p. 1–29. Available from: <https://osf.io/h9gsd/>.
87. Yaniv Z, Lowekamp BC, Johnson HJ, Beare R. SimpleITK image-analysis notebooks: A collaborative environment for education and reproducible research. *Journal of Digital Imaging*. 2018; 31(3):290–303. <https://doi.org/10.1007/s10278-017-0037-8> PMID: 29181613