



SOFTWARE TOOL ARTICLE

REVISED Fiji plugins for qualitative image annotations: routine analysis and application to image classification [version 2; peer review: 2 approved, 1 approved with reservations]

Laurent S. V. Thomas ¹⁻³, Franz Schaefer³, Jochen Gehrig ^{1,2}

¹Acquifer Imaging GmbH, Heidelberg, Germany

²DITABIS, Digital Biomedical Imaging Systems AG, Pforzheim, Germany

³Department of Pediatrics, University Children's Hospital, Heidelberg, Germany

v2 First published: 15 Oct 2020, 9:1248
<https://doi.org/10.12688/f1000research.26872.1>

Latest published: 12 Feb 2021, 9:1248
<https://doi.org/10.12688/f1000research.26872.2>

Abstract

Quantitative measurements and qualitative description of scientific images are both important to describe the complexity of digital image data. While various software solutions for quantitative measurements in images exist, there is a lack of simple tools for the qualitative description of images in common user-oriented image analysis software. To address this issue, we developed a set of Fiji plugins that facilitate the systematic manual annotation of images or image-regions. From a list of user-defined keywords, these plugins generate an easy-to-use graphical interface with buttons or checkboxes for the assignment of single or multiple pre-defined categories to full images or individual regions of interest. In addition to qualitative annotations, any quantitative measurement from the standard Fiji options can also be automatically reported. Besides the interactive user interface, keyboard shortcuts are available to speed-up the annotation process for larger datasets. The annotations are reported in a Fiji result table that can be exported as a pre-formatted csv file, for further analysis with common spreadsheet software or custom automated pipelines. To illustrate possible use case of the annotations, and facilitate the analysis of the generated annotations, we provide examples of such pipelines, including data-visualization solutions in Fiji and KNIME, as well as a complete workflow for training and application of a deep learning model for image classification in KNIME. Ultimately, the plugins enable standardized routine sample evaluation, classification, or ground-truth category annotation of any digital image data compatible with Fiji.

Keywords

ImageJ, Fiji, KNIME, image annotation, image classification, ground-truth labelling, qualitative analysis, bioimage analysis

Open Peer Review

Reviewer Status

	Invited Reviewers		
	1	2	3
version 2			
(revision)			
12 Feb 2021		report	report
		↑	
version 1			
15 Oct 2020	report	report	

1. **Jan Eglinger** , Friedrich Miescher Institute for Biomedical Research (FMI), Basel, Switzerland
2. **Aliaksandr Halavaty**, European Molecular Biology Laboratory, Heidelberg, Germany
Christian Tischer , European Molecular Biology Laboratory (EMBL), Heidelberg, Germany
3. **Elisabet Teixido** , University of Barcelona, Barcelona, Spain

Any reports and responses or comments on the article can be found at the end of the article.



This article is included in the **NEUBIAS - the Bioimage Analysts Network gateway**.

Corresponding author: Laurent S. V. Thomas (l.thomas@acquifer.de)

Author roles: **Thomas LSV:** Conceptualization, Software, Validation, Visualization, Writing – Original Draft Preparation, Writing – Review & Editing; **Schaefer F:** Writing – Review & Editing; **Gehrig J:** Conceptualization, Funding Acquisition, Project Administration, Supervision, Validation, Writing – Review & Editing

Competing interests: Laurent Thomas and Jochen Gehrig are employees of ACQUIFER Imaging GmbH, Heidelberg, Germany

Grant information: This project has received funding from the European Union's Horizon 2020 research and innovation programme to Ditabis under the Marie Skłodowska-Curie grant agreement No 721537 "ImageInLife".

The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.

Copyright: © 2021 Thomas LSV *et al.* This is an open access article distributed under the terms of the [Creative Commons Attribution License](#), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

How to cite this article: Thomas LSV, Schaefer F and Gehrig J. **Fiji plugins for qualitative image annotations: routine analysis and application to image classification [version 2; peer review: 2 approved, 1 approved with reservations]** F1000Research 2021, 9:1248 <https://doi.org/10.12688/f1000research.26872.2>

First published: 15 Oct 2020, 9:1248 <https://doi.org/10.12688/f1000research.26872.1>

REVISED Amendments from Version 1*Manuscript*

- the introduction includes a brief review of existing annotation solutions and their limitations
- [Figure 1–Figure 3](#) have been updated to reflect the new plugin interfaces
- [Figure 4](#) has been replaced with an overview figure of the possible data-visualizations and applications
- similarly the *Uses cases* section was simplified, there is no more dedicated paragraphs for the sunburst chart and deep learning
- Previous [Figure 4](#) is now available as [Supplementary Figure 4](#) on Zenodo
- A new data-visualization Fiji plugin for pie chart visualization was implemented (see new supplementary [Figure 2](#))
- the DOI link to Zenodo was updated to always point to the latest version
- The ‘competing interest’ statement was updated to reflect the current positions of Jochen Gehrig and Laurent Thomas. Both authors are former employees of DITABIS AG, Pforzheim, Germany and as of 2021 employees of ACQUIFER Imaging GmbH, Heidelberg, Germany exclusively.

Plugins

Several enhancements of the plugins previously suggested were implemented :

- option to browse images in a directory
- selection of the dimension for hyperstack browsing
- “Add new category” button for the “button” and “checkbox” plugins
- pop-up with keyboard shortcut message for buttons in the “button” plugin
- “run measure” checkbox was moved to the initial configuration window
- annotations are appended to any active table for ImageJ > 1.53g
- Add a checkbox option to recover categories from an active table

Any further responses from the reviewers can be found at the end of the article

Introduction

A common requirement of most imaging projects is to qualitatively describe images, either by assigning them to defined categories or by selecting a set of descriptive keywords. This routine task is shared by various scientific fields, for instance in biomedical research for the categorization of samples, in clinical imaging for image-based diagnostics, or in manufacturing for the description of object-properties.

Qualitative descriptors, or keywords, can correspond to the presence or discrete count of features, the evaluation of quality criteria, or the assignment of images to specific categories. While automated methods for such qualitative description may exist, they usually require substantial effort for their implementation and validation. Therefore, for routine image data analysis and inspection, the qualitative description is usually performed manually. Similarly, for the training of machine learning models, manual annotations by experts are typically used as ground truth material.

For small datasets of a few dozen images, manual description of images can be performed by reporting the image identifier and qualitative descriptors in a simple spreadsheet. However, for larger datasets or large number of descriptors, this becomes quickly overwhelming and error prone as one needs to inspect a multitude of images while appending information to increasingly complex tables. Several software tools have been reported for the annotation of images or regions of interest (ROIs), mostly targeting ground-truth annotations for automated classification and object-detection (see [Hollandi et al., 2020](#) for a comparison of available solutions). However, most of these software tools have been initially designed for the annotation of real-life photographs, and thus have limited compatibility with scientific image formats (e.g. 16-bit tiff), besides requiring specific installation and configuration. Most bio-image analysis software packages similarly support annotations in the form of regions of interests (ROIs) associated to a category label (e.g. ImageJ/Fiji ([Schindelin et al., 2012](#)); [Schneider et al., 2012](#)), QuPath ([Bankhead et al., 2017](#)), Ilastik ([Berg et al., 2019](#)), ICY ([de Chaumont et al., 2012](#)), KNIME ([Berthold et al., 2009](#))), with applications for classification or segmentation. Typically, with those existing solutions, only a single category descriptor can be associated to an image or ROI. There is surprisingly no widespread solution available in common user-oriented scientific image analysis software, for the assignment of multiple descriptive keywords to images or ROIs. We previously proposed a standalone python annotation tool for this purpose, illustrated with the annotation of zebrafish morphological phenotypes ([Westhoff et al., 2020](#)). However, we believe that a similar implementation integrated within a widespread scientific image-analysis software would improve compatibility with image formats, software distribution, long-term support and adoption by the community. Therefore, we developed a set of plugins for Fiji, to facilitate and standardize routine qualitative image annotations, in particular for large image datasets. We also illustrate possible applications of the resulting standardized qualitative description for the visualization of data-distribution, or the training of supervised image classification models.

Methods**Implementation**

We developed a set of Fiji plugins for the assignment of single or multiple descriptive keywords to images, or image-regions outlined by ROIs. The plugins provide an intuitive graphical user-interface (GUI) consisting of either buttons, checkboxes, or dropdown menus for the assignment of user-defined keywords ([Figure 1–Figure 3](#)). The GUI is automatically generated from a set of keywords, defined at the beginning of the annotation session. Additional keywords and associated GUI elements can be later added to the plugin interface during the annotation to account for new descriptors, by clicking the “Add category” button. In addition to the pre-defined set of keywords, arbitrary image-specific comments can also be entered via a text input field. Furthermore, if *run Measure* is selected in the graphical interface, quantitative measurements as defined in Fiji’s menu *Analyze>Set Measurements* are reported in addition to the selected keywords. By default, the annotations and measurements are assigned to the entire image but can also

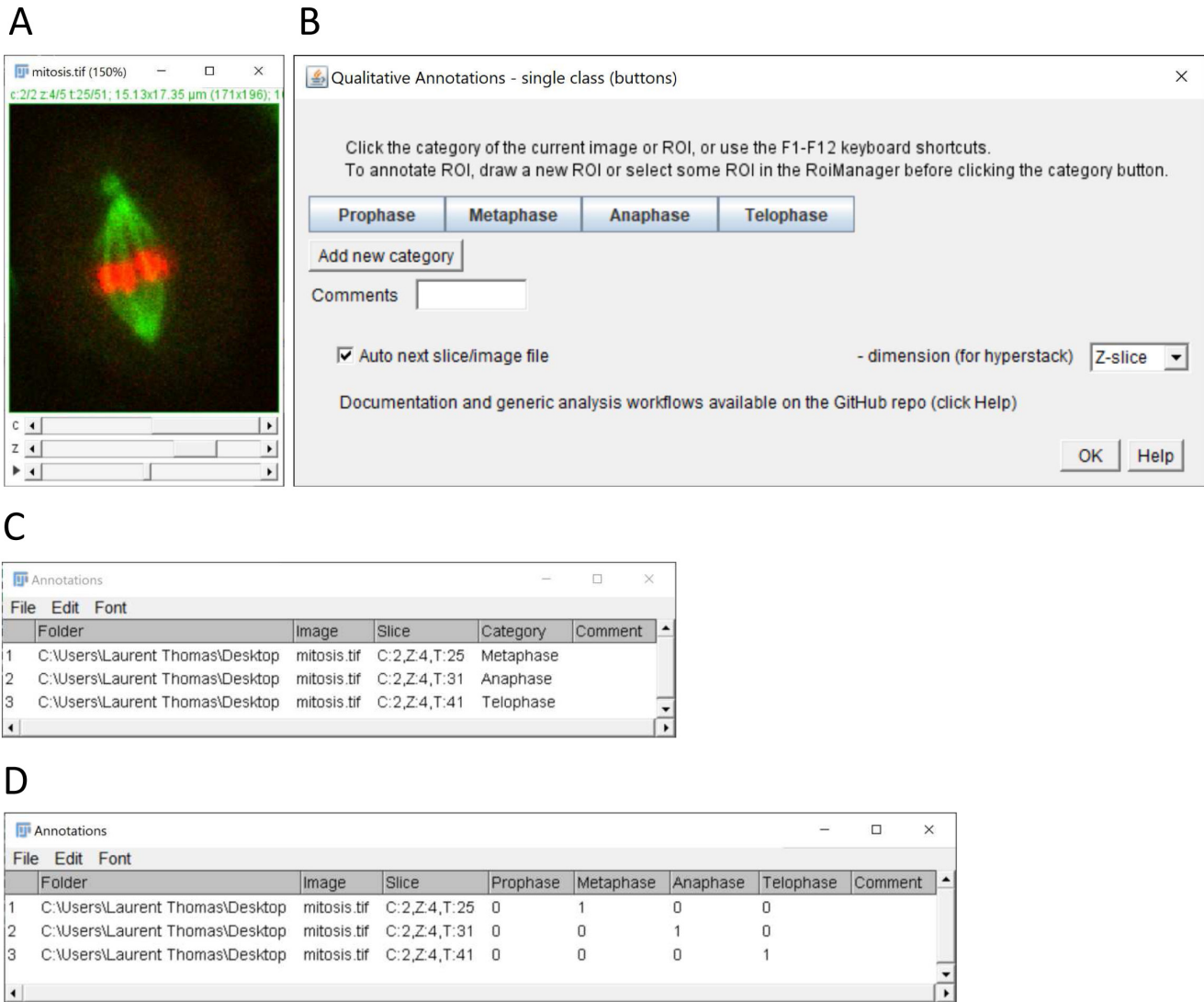


Figure 1. Single-category annotation of images in multi-dimensional stacks. (A) Example of multi-dimensional image stack used to annotate mitotic stages in time-lapse data (source: ImageJ) example image “Mitosis” – image credit NIH). (B) Graphical interface of the *single class (buttons)* plugin configured for annotation of 4 mitotic stages. (C) Results table with annotated categories (column *category*) generated by the plugin after selecting the *single category column* option in the plugin configuration window (not shown). (D) Alternative results table output using 1-hot encoding after selecting the option *1 column per category*. The resulting 1-hot encoding of categories can be used for the training of classification algorithms.

describe image-regions outlined by ROI. The latter is simply achieved by drawing a new ROI on the image or selecting existing ROIs in the ROI Manager before assigning the keywords (Figure 3). Newly drawn ROIs are automatically added to the ROI Manager upon annotation with the plugins.

The selected keywords, comments, measurements, and the image filename and directory are reported in a Fiji result table window with one row per annotation event (Figure 1C, D). An annotation event is triggered when a button is clicked, or a keyboard shortcut is pressed. With ROIs, the identifiers of the ROIs are also reported in the table, and the descriptors are

saved as properties of the ROI objects. The information can then be retrieved from the ROIs using the Fiji scripting functions. The result table is updated row by row, as the user progresses with the annotations. Table rows can be deleted within Fiji if some annotations should be corrected. The result table can be saved as a csv file at any time and edited in a spreadsheet software or text editor, for instance to update the image directory column when images have been transferred to a different location or workstation.

Three plugins are provided to accommodate for different annotation modalities. The *single-class (buttons)* plugin

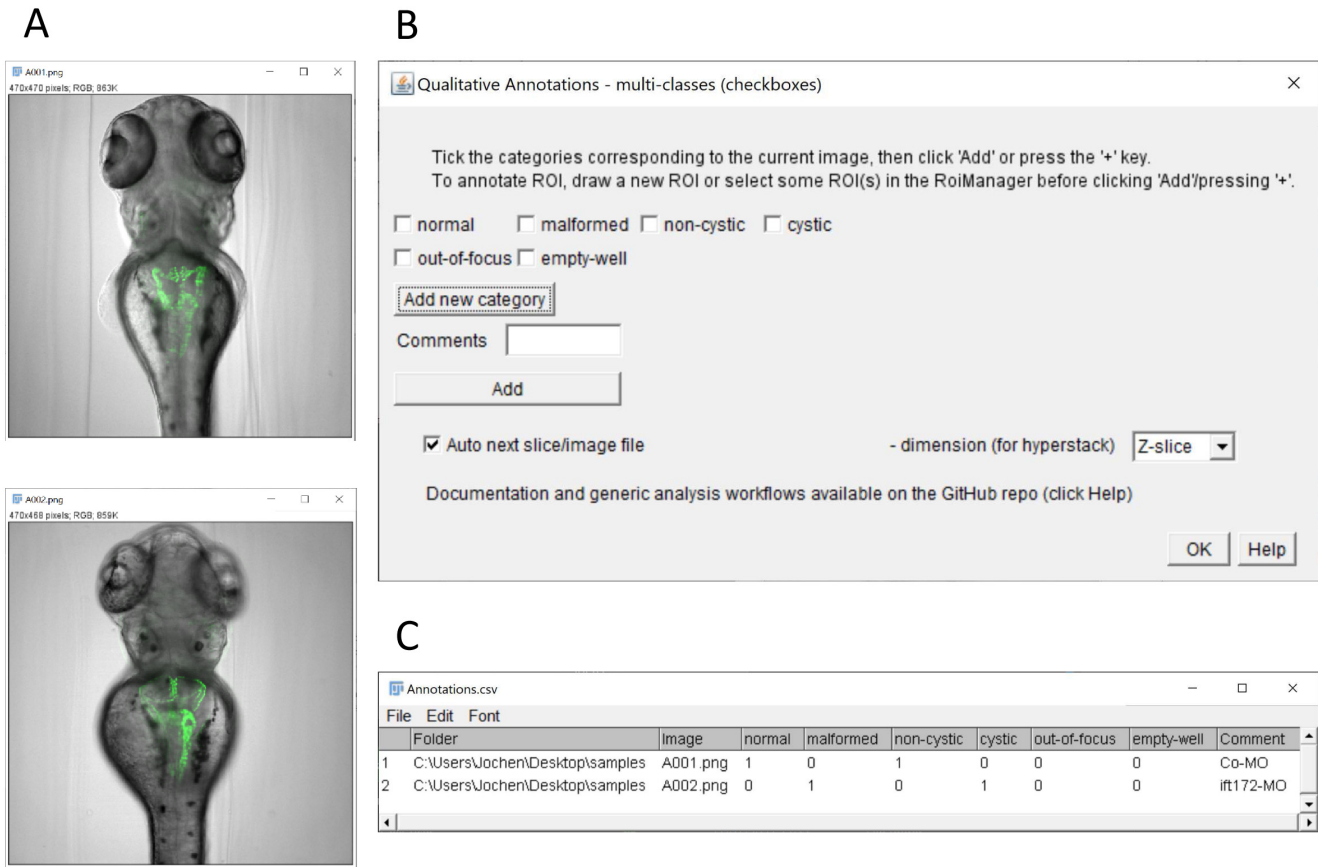


Figure 2. Annotation of multiple categories using the multi-class (checkboxes) plugin. (A) Example images of transgenic zebrafish larvae of the *Tg(wt1b:egfp)* transgenic line after injection with control morpholino (upper panel) or with *ift172* morpholino (lower panel) inducing pronephric cysts. In this illustration, the plugin is used to score overall morphology and cyst formation. It could also be used to mark erroneous images (such as out-of-focus or empty wells). Images are from (Pandey *et al.*, 2019). (B) Graphical interface of the checkbox annotation plugin configured with 2 checkboxes for overall morphology, 2 checkboxes for pronephric cysts, and checkboxes to report out-of-focus and empty wells. Contrary to the *single class (button)* plugin, multiple categories can be assigned to a given image. (C) Resulting multi-category classification table with binary encoding of the annotations (True/False).

(Figure 1) allows the assignment of a single keyword per image by clicking the corresponding button. With this plugin, the user can decide if the result table should contain a single category column containing the clicked category keyword for each image (Figure 1C), or one column per category with a binary code (0/1) depicting the assignment (Figure 1D). The latter option is particularly suitable for the training of supervised classification algorithms, which typically expect for their training an array of probabilities with 1 for the actual image-category and 0 for all other categories (also called 1-hot encoding, see Müller & Guido, 2016).

The *multi-class (checkboxes)* plugin (Figure 2) allows multiple keywords per image, which are selected via associated checkboxes (Figure 2B). This yields a result table with one column per keyword, and a 0/1 code if the keywords apply or not (Figure 2C). In this case, the table structure is similar to

Figure 1D except that multiple keywords might be selected for a given image (i.e. multiple 1 for a given table row, as in row 1).

The *multi-class (dropdown)* plugin (Figure 3) allows choosing keywords from distinct lists of choices using dropdown menus (Figure 3B). The labels and choices for the dropdown menus are defined by the user in a simple csv file (Extended data, Supplementary Figure 1) (Thomas, 2020). This is convenient if multiple image features should be reported in separate columns (content, quality, etc.) with several options for each feature.

Operation

The plugins run on any system capable of executing Fiji. Executing one of the plugins will first display a set of configuration windows to define the keywords, image browsing

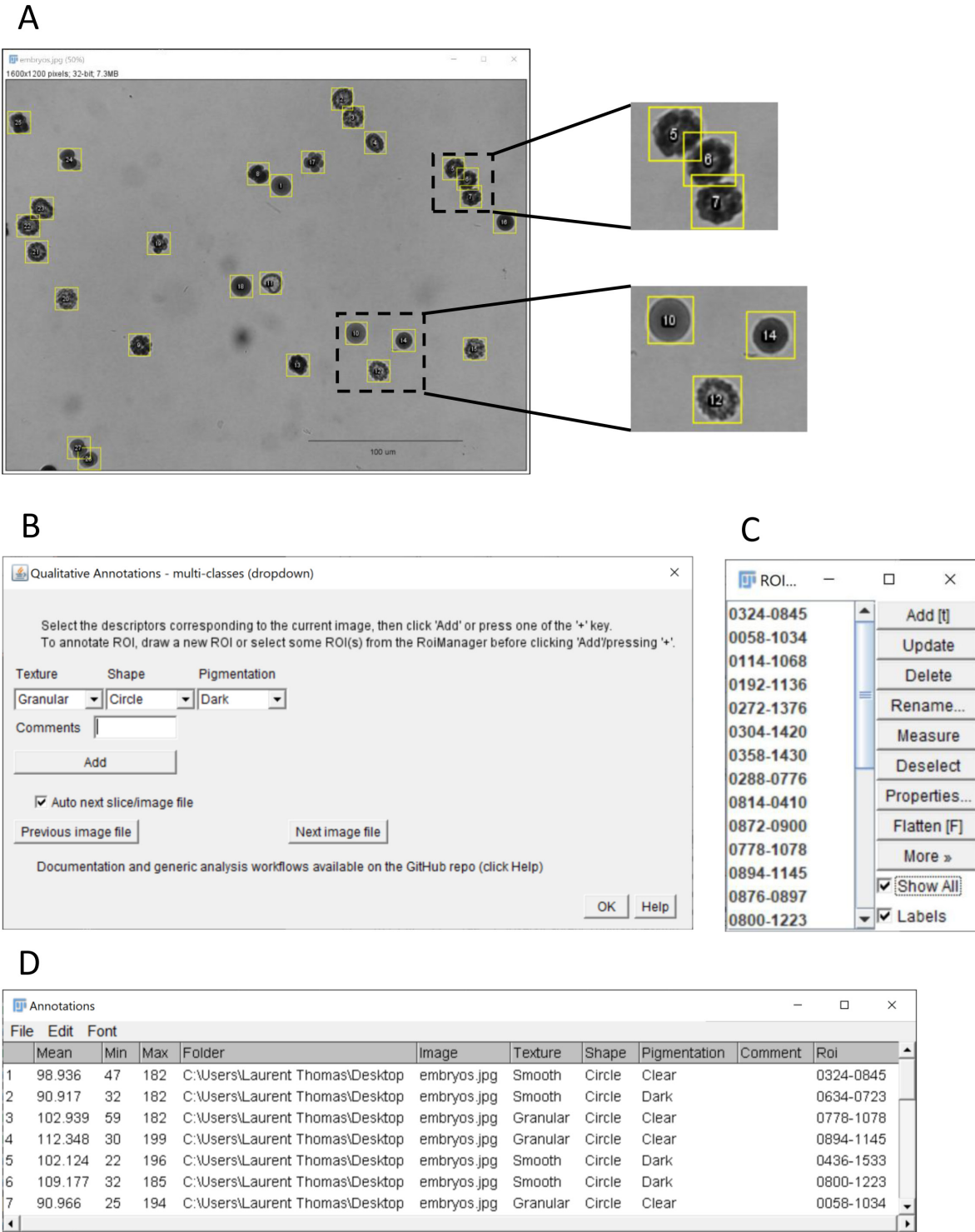


Figure 3. Qualitative and quantitative annotations of image regions using the multi-class (dropdown) plugin. (A) ImageJ's sample image "embryos" after conversion to grayscale using the command *Image > Type > 32-bit* (image credit: NIH). The embryos outlined with yellow regions of interest were annotated using the "multi-class (dropdown)" plugin. The insets at the top shows the annotation of overlapping ROIs, here corresponding to embryos with phenotype granular texture, dark pigmentation and elliptic shape. The inset at the bottom shows other embryos with different phenotypes (10: smooth/clear/circular, 12: granular/clear/elliptic, 14: smooth/dark/circular). **(B)** Graphical interface of the multi-class (dropdown) plugin. Three exemplary features are scored for each embryo: texture (granular, smooth), shape (circle, ellipse) and pigmentation (dark, clear). Quantitative measurements as selected in the *Analyze > set Measurements* menu (here Mean, Min and Max grey level) are also reported for each embryo, when the *run Measure* option is ticked. **(C)** ROI Manager with ROIs corresponding to annotated regions. **(D)** Resulting classification table with the selected features, qualitative measurement and associated ROI identifier for the outlined embryos.

mode (stack or directory) and if quantitative measurements should be reported (“Run measure”). Upon validation of the configuration, the actual annotation interface as in [Figure 1–Figure 3](#) is displayed and ready to use for annotation. For the *single-class (buttons)* plugin, annotations can be recorded by clicking the corresponding category button, or by pressing one of the F1-F12 keyboard shortcuts. The shortcuts are automatically mapped to the categories in the order of their respective buttons, e.g. pressing F1 is equivalent to clicking the leftmost button (see pop-up message when hovering the mouse over a button). For the *multi-class* plugins, the annotations can be recorded by clicking the *Add* button or pressing one of the + keys of the keyboard. For every plugin, an annotation event updates the results table as described in the implementation section, stores any newly drawn ROI into Fiji’s ROI Manager, and if the corresponding option is selected in the graphical user interface, the next image slice of the selected dimension is displayed in “stack” browsing mode when a stack is annotated. Similarly, in “directory” browsing mode the next image file is loaded. The annotations are automatically appended to an active Fiji table window if available. For Fiji installations with ImageJ versions below 1.53g, annotations will be appended to a table window entitled “Annotations” or “Annotations.csv” if available.

Use cases

The described plugins allow the rapid and systematic description of single images, image-planes within multi-dimensional images or image-regions with custom keywords. Rich qualitative descriptions can thus be reported by combining multiple keywords, although the plugins can also be used for ground-truth category annotation, for which typically a single label is reported for each image instance. Additionally, by activating the measurement option, the qualitative description can be complemented by any of ImageJ’s quantitative measurements. The annotation tools can be used for routine image evaluation e.g. for the assignment of predefined categories, to identify outliers or low-quality images, or to assess the presence of a particular object or structure. Examples annotations are illustrated in [Figure 1](#) (single-cell mitotic stage), [Figure 2](#) (pronephric morphological alterations in transgenic zebrafish larvae ([Pandey et al., 2019](#))) and [Figure 3](#) (phenotypic description of multicellular embryos). Images and annotations are available as *Underlying data* ([Pandey et al., 2020](#)).

For the annotation of ROIs, the presented plugins can be used in combination with our previously published *ROI 1-click tools* ([Thomas & Gehrig, 2020](#)), which facilitate the creation of ROIs of predefined shapes, and the automated execution of custom commands for these ROIs. The generated ROIs can then be described with qualitative features using the hereby presented plugins, either for one ROI at a time, or by simultaneously selecting multiple ROIs. Besides facilitating qualitative annotations, the plugins have the advantage to generate tables with standardized structures that can potentially facilitate the visualization and analysis of the annotations by automated workflows.

To illustrate and expand on the potential of such annotations, we provide a set of generic workflows which directly operate on tables generated by one of the presented plugins. The example workflows demonstrate classical scenarios for the exploitation of qualitative descriptors: (i) the visualization of data-distribution using a pie chart (Fiji plugin, [Figure 4.B](#), Supplementary Figure 2), (ii) interactive sunburst chart visualization (*Extended data KNIME workflow*, [Figure 4.C](#), Supplementary Figure 3) ([Thomas, 2020](#)), and (iii) the training of a deep learning model for image classification (KNIME workflows, [Figure 4.D](#), Supplementary Figure 4, 5). We developed a dedicated Fiji plugin for the pie chart visualization as part of the *Qualitative Annotations* update site. The plugin relies on the [JFreeChart library](#), providing advanced customisation options and readily available for scripting in Fiji. The plugin allows representing the data-distribution from a table column and is macro-recordable. It is not limited to results tables generated with the annotation plugins, and thus offer a novel plotting option for end-users in Fiji. While the pie chart represents the distribution for a single data column, the sunburst chart visualization in KNIME allows visualizing and relating the distribution for multiple feature columns, as each column is represented as an additional level in the chart. Besides, the plot is generated by a javascript view node, which offers enhanced interactivity for the exploration of the data-distribution. Finally, for the training of a deep learning model for image-classification, we adapted an existing [KNIME example workflow](#) for transfer-learning of a pretrained model. We could demonstrate rapid model training and high classification accuracy with a moderate number of training images (116) for the classification of microscopy images representing kidney morphologies in developing zebrafish larvae ([Figure 4D](#), Supplementary Figure 4D). The KNIME workflows do not require advance knowledge of KNIME and can be readily used without major adaptation. By providing those examples, we also wish to facilitate and spread those advanced data-processing tools, by drastically reducing the need for custom development. We also provide detailed documentation about the workflows and required software dependencies on the GitHub repository (<https://github.com/LauLauThom/Fiji-QualiAnnotations>).

Conclusion

Here, we propose a set of plugins for the qualitative annotations of images or image regions, designed for the popular image analysis software Fiji. The annotations comprise user-defined keywords, as well as optional quantitative measurements as available in Fiji. The keywords can describe categorical classification, the evaluation of quality metrics or the presence of particular objects or structures. The plugins are easy to install and to use via an intuitive graphical user interface. In particular, the tools facilitate tedious qualitative annotation tasks, especially for large-datasets, or for the evaluation of multiple features. The annotations are recorded as standardized result tables, to facilitate automated analysis by generic workflows. To this extent, example workflows for data-visualization and supervised data classification are provided, which can be directly executed with the resulting annotation table without further

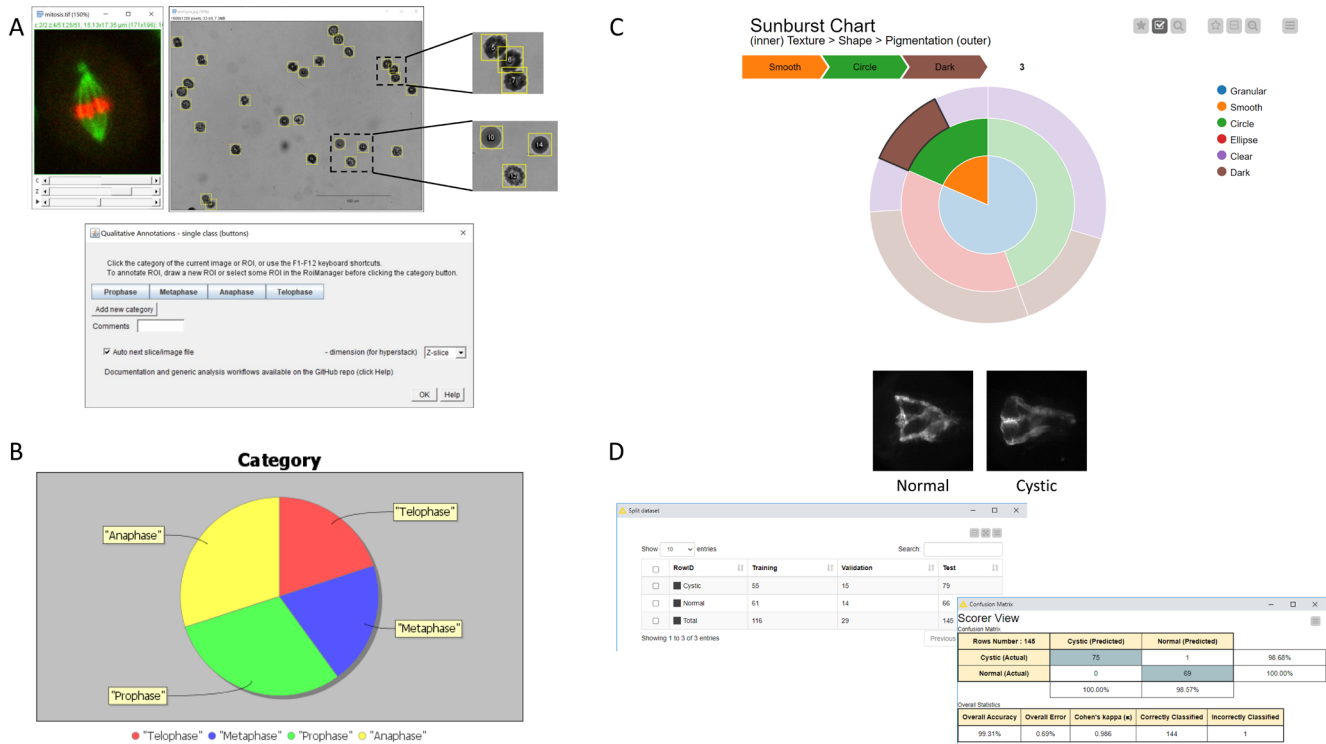


Figure 4. Overview of the annotation tools and possible use cases of the annotations. (A) The qualitative annotation plugins provide simple graphical user interfaces for the annotations of images or image regions outlined by ROIs (see Figure 1–Figure 3). (B) Pie chart visualization of the data-distribution from a single table column, here illustrated with the distribution of the mitotic stage in a population of cells (fictive distribution). The plot is generated in Fiji by the plugin “Pie chart from table column”, provided with the annotation plugins (see Supplementary Figure 2). (C) Interactive sunburst chart visualization in KNIME, illustrated with the distribution of morphological phenotypes of multi-cellular embryos (as in Figure 3). Distinct data-columns are represented as successive levels of the chart (see Supplementary Figure 3). (D) Training of a deep-learning model for image-classification in KNIME with representative images of the custom categories (kidney morphology in zebrafish larvae, left: normal, right: cystic), distribution of the annotated images between training, validation and test fraction, and result of the classification shown as a confusion matrix (see Supplementary Figure 4,5).

customization effort (Figure 4). Finally, video tutorials about the plugins and analysis workflows are available on YouTube.

Data availability

Underlying data

Zenodo: Fluorescently-labelled zebrafish pronephroi + ground truth classes (normal/cystic) + trained CNN model. <https://doi.org/10.5281/zenodo.3997728> (Pandey et al., 2020).

This project contains the following underlying data:

- Annotations-multiColumn.csv. (Ground-truth category annotations.)
- Annotations-singleColumn.csv. (Ground-truth category annotations.)
- images.zip. (Images of fluorescently labelled pronephroi in transgenic *Tg(wt1b:EGFP)* zebrafish larvae used for the training and validation of the deep learning model for classification.)
- trainedModel.zip (Pretrained model.)

Extended data

Zenodo: Qualitative image annotation plugins for Fiji - <https://doi.org/10.5281/zenodo.4063891> (Thomas, 2020).

This project contains the following extended data:

- Supplementary Figure 1: Detail of the input for the multi-class (dropdown) plugin.
- Supplementary Figure 2: Custom plugin for data-visualization as a pie chart in Fiji.
- Supplementary Figure 3: Visualizing data-distribution using sunburst charts in KNIME.
- Supplementary Figure 4: Training a deep learning model for image classification in KNIME using the generated annotations.
- Supplementary Figure 5: Detail of the Keras network learner KNIME node.

Data are available under the terms of the Creative Commons Attribution 4.0 International license (CC-BY 4.0).

Software availability

Source codes, documentation and example workflows are available at: <https://github.com/LauLauThom/Fiji-Quali-Annotations>.

Archived source code at time of publication: <https://doi.org/10.5281/zenodo.4064118> (Thomas, 2020).

License: GNU General Public License v3.

The plugins can be installed in Fiji by simply activating the *Qualitative Annotations* update site. Then the plugins are listed under the menu *Plugins > Qualitative Annotations*.

A pre-configured Fiji installation bundle for windows is also archived in the release section of the repository.

The following KNIME workflows and associated documentation README files are available in the subdirectory *KNIMEworkflows* under a [Creative Commons Attribution 4.0 International License \(CC-BY\)](https://creativecommons.org/licenses/by/4.0/):

- View-Images-And-Annotations workflow
- Sunburst-chart-workflow (with csv of annotations for multi-cellular embryos)
- Deep-Learning – binary classifier – training workflow
- Deep-Learning – binary classifier – prediction workflow
- Deep-Learning – multi-class classifier – training workflow
- Deep-Learning – multi-class classifier – prediction workflow

Acknowledgements

We thank Gunjan Pandey (ACQUIFER and Children's Hospital) for generating images and Jens Westhoff (Children's Hospital, Heidelberg) for general support.

This publication was supported by COST Action NEUBIAS (CA15124), funded by COST (European Cooperation in Science and Technology).

References

- Bankhead P, Loughrey MB, Fernández JA, et al.: **QuPath: Open source software for digital pathology image analysis**. *Sci Rep*. 2017; **7**(1): 16878.
[PubMed Abstract](#) | [Publisher Full Text](#) | [Free Full Text](#)
- Berg S, Kutra D, Kroeger T, et al.: **ilastik: interactive machine learning for (bio)image analysis**. *Nat Methods*. 2019; **16**(12): 1226–1232.
[PubMed Abstract](#) | [Publisher Full Text](#)
- Berthold MR, Cebon N, Dill F, et al.: **KNIME - the Konstanz information miner: version 2.0 and beyond**. *ACM SIGKDD Explor Newsl*. 2009; **11**(1): 26–31.
[Publisher Full Text](#)
- de Chaumont F, Dallongeville S, Chenouard N, et al.: **Icy: an open bioimage informatics platform for extended reproducible research**. *Focus on bioimage informatics*. *Nat Methods*. 2012; **9**(7): 690–696.
[PubMed Abstract](#) | [Publisher Full Text](#)
- Hollandi R, Diósdí A, Hollandi G, et al.: **AnnotatorJ: an ImageJ plugin to ease hand annotation of cellular compartments**. Ed. Lippincott-Schwartz, J. *Mol Biol Cell*. 2020; **31**(20): 2179–2186.
[PubMed Abstract](#) | [Publisher Full Text](#) | [Free Full Text](#)
- Müller AC, Guido S: **Introduction to machine learning with Python: a guide for data scientists**. First edition. ed. O'Reilly Media, Inc, Sebastopol, CA. 2016.
[Reference Source](#)
- Pandey G, Gehrig J, Thomas L: **Fluorescently-labelled zebrafish pronephroi + ground truth classes (normal/cystic) + trained CNN model (Version 1.0)** [Data set]. *Zenodo*. 2020.
<http://www.doi.org/10.5281/zenodo.3997728>
- Pandey G, Westhoff J, Schaefer F, et al.: **A Smart Imaging Workflow for Organ-Specific Screening in a Cystic Kidney Zebrafish Disease Model**. *Int J Mol Sci*. 2019; **20**(6): 1290.
[PubMed Abstract](#) | [Publisher Full Text](#) | [Free Full Text](#)
- Schindelin J, Arganda-Carreras I, Frise E, et al.: **Fiji: an open-source platform for biological-image analysis**. *Nat Methods*. 2012; **9**(7): 676–682.
[PubMed Abstract](#) | [Publisher Full Text](#) | [Free Full Text](#)
- Schneider CA, Rasband WS, Eliceiri KW: **NIH Image to ImageJ: 25 years of image analysis**. *Nat Methods*. 2012; **9**(7): 671–675.
[PubMed Abstract](#) | [Publisher Full Text](#) | [Free Full Text](#)
- Thomas L: **Qualitative image annotation plugins for Fiji (Version 1.0.2bis)**. *Zenodo*. 2020.
<http://www.doi.org/10.5281/zenodo.4064118>
- Thomas LS, Gehrig J: **ImageJ/Fiji ROI 1-click tools for rapid manual image annotations and measurements**. 2020.
[Publisher Full Text](#)
- Westhoff JH, Steenbergen PJ, Thomas LSV, et al.: **In vivo High-Content Screening in Zebrafish for Developmental Nephrotoxicity of Approved Drugs**. *Front Cell Dev Biol*. 2020; **8**: 583.
[PubMed Abstract](#) | [Publisher Full Text](#) | [Free Full Text](#)

Open Peer Review

Current Peer Review Status:



Version 2

Reviewer Report 31 March 2021

<https://doi.org/10.5256/f1000research.54338.r81499>

© 2021 Teixido E. This is an open access peer review report distributed under the terms of the [Creative Commons Attribution License](#), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.



Elisabet Teixido 

Toxicology Unit, University of Barcelona, Barcelona, Spain

The authors present a new set of Fiji plugins for the qualitative annotation of images in a single or multiple descriptive keywords. They also include the possibility to annotate outlined ROIs of the image and include some quantitative measurements using the defined menu in Fiji.

There is an increasing need for future development of annotation tools in order to make the annotation process more efficient and they design a very useful tool for the annotation of 2D images.

The publication is well written and describes the plugin in detail, in addition there are some youtube tutorials that can help you understand and install the plugin in Fiji. I really appreciate the inclusion of some examples for the generated annotations, they include some analysis in ImageJ itself and also presented pipelines for data-visualization and a deep learning model for classification in KNIME.

I just have some comments and future suggestions for the tool.

Biomedical researchers are looking to use ontologies to support cross-laboratory data sharing and integration. It would be useful to be able to fetch categories from a file and not from a table of a set of images previously analysed. That way we may create common vocabulary annotation sets following ontology terms that can be shared across labs.

The only current limitation of the tool is not been able to save coordinates of the ImageJ ROIs. It would be useful for shape analysis and also for building machine learning models to automatically identify for instance anatomical parts of an image.

Is the rationale for developing the new software tool clearly explained?

Yes

Is the description of the software tool technically sound?

Yes

Are sufficient details of the code, methods and analysis (if applicable) provided to allow replication of the software development and its use by others?

Yes

Is sufficient information provided to allow interpretation of the expected output datasets and any results generated using the tool?

Yes

Are the conclusions about the tool and its performance adequately supported by the findings presented in the article?

Yes

Competing Interests: No competing interests were disclosed.

Reviewer Expertise: Bioimage analysis, Toxicology

I confirm that I have read this submission and believe that I have an appropriate level of expertise to confirm that it is of an acceptable scientific standard.

Reviewer Report 26 February 2021

<https://doi.org/10.5256/f1000research.54338.r79507>

© 2021 Tischer C et al. This is an open access peer review report distributed under the terms of the [Creative Commons Attribution License](#), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.



Aliaksandr Halavatyi

European Molecular Biology Laboratory, Heidelberg, Germany

Christian Tischer 

European Molecular Biology Laboratory (EMBL), Heidelberg, Germany

We now approve this article for indexing.

Is the rationale for developing the new software tool clearly explained?

Yes

Is the description of the software tool technically sound?

Yes

Are sufficient details of the code, methods and analysis (if applicable) provided to allow replication of the software development and its use by others?

Yes

Is sufficient information provided to allow interpretation of the expected output datasets and any results generated using the tool?

Yes

Are the conclusions about the tool and its performance adequately supported by the findings presented in the article?

Yes

Competing Interests: No competing interests were disclosed.

We confirm that we have read this submission and believe that we have an appropriate level of expertise to confirm that it is of an acceptable scientific standard.

Version 1

Reviewer Report 26 November 2020

<https://doi.org/10.5256/f1000research.29676.r74544>

© 2020 Tischer C et al. This is an open access peer review report distributed under the terms of the [Creative Commons Attribution License](#), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.



Christian Tischer 

European Molecular Biology Laboratory (EMBL), Heidelberg, Germany

Aliaksandr Halavatyi

European Molecular Biology Laboratory, Heidelberg, Germany

The authors present an ImageJ plugin suite for the annotation of image planes and regions of interest (ROI) within image planes, using standard ImageJ technology such as results tables and the ROI manager. We think this is a very valuable addition to the ImageJ ecosystem as it is easy to use and very well integrated with current functionality that is already known to many users and developers.

Below we have some specific comments which we hope could be helpful to further improve the publication and the plugins.

We thank the authors for their efforts in implementing this great addition to the ImageJ ecosystem and hope that below suggestions are helpful!

With kind regards,
Christian Tischer and Aliaksandr Halavatyi

[Suggestions for the publication](#)

We think the publication is nicely written and describes their plugins very well.

The authors describe how the output of their plugins could be consumed by a machine learning workflow in KNIME. While the workflow in KNIME is very interesting and relevant we feel at the same time that it is slightly out of scope and we thus suggest moving this to the supplemental material. The reason is that we would expect the main readers of this publication and also the main users of the plugins to be familiar with the ImageJ ecosystem, but not necessarily with KNIME. In addition, if the final aim of the users is to execute a workflow in KNIME there may be other annotation tools available and the authors should clarify how their tool is superior to potential other solutions for image annotation.

Along those lines, it would be great to also present (maybe also in the supplemental material) a follow up workflow that can be executed entirely within ImageJ. Specifically, it would be great to see an example of how the annotated ROIs could be used, e.g. to train a machine learning model. However, we don't know how feasible this is and it is really just a suggestion.

Technical issues and suggestions for the plugins

There are hotkeys (F1, F2, ...) available for the single class plugin, which is great as it speeds up annotation. However, we did not manage to have them working on MacOS (10.14.6). In addition, if one has many classes one would have to count the buttons to figure out which hotkey belongs to which class. We thus suggest that the hotkeys are either written on the category buttons or appears as pop-up messages when a user hovers over the corresponding button.

Currently, one cannot add additional categories during the labelling process. We think it would be great to be able to (e.g., via a new button: [Add category]), because images may contain phenotypes that one did not think about upfront.

One can annotate the same image, slice and ROI multiple times, where each annotation will add another row to the table. As a consequence, the table contains conflicting annotations for the same image region. We think it would be (much) better if the annotation was replaced, at least optionally.

The possibility to continue an annotation by loading a table and renaming it to "Annotations" is a useful feature, but we think the current implementation has a limitation in a sense that the class names are not read from the table, but rather taken from the last annotation session. This can be an issue if users work on several annotations. We suggest adding an option to fetch the set of category names from the table.

The logic of jumping to the next image slice in case of hyperstacks (4D or 5D data) is not completely obvious. We suggest adding a drop-down menu next to the "auto next slice" checkbox, where the user can select the dimension (z, c, or t) along which to automatically progress.

Currently, annotating a set of images that cannot be loaded into a hyperstack, e.g. because they may have different sizes or dimensionality, is possible but a bit tedious. We suggest adding a modality where the user would select one folder on the disk and then the plugin would automatically open and close the images in the folder one-by-one, allowing them to be annotated one-by-one. The additional advantage of this modality also is that loading different images into a

hyperstack is not something that every user may know how to achieve. We suggest adding [previous] and [next] buttons to this modality, allowing one to correct a previous annotation in case a mistake has been made.

ImageJ ROIs are not a “cross platform standard”. In order to enable follow up workflows outside ImageJ it may be thus useful to store the bounding box coordinates for rectangular ROIs in the table.

When clicking the Help button on one of the annotation UIs the UI was closing, which we think is probably a bug (tested on Mac).

Is the rationale for developing the new software tool clearly explained?

Yes

Is the description of the software tool technically sound?

Yes

Are sufficient details of the code, methods and analysis (if applicable) provided to allow replication of the software development and its use by others?

Yes

Is sufficient information provided to allow interpretation of the expected output datasets and any results generated using the tool?

Partly

Are the conclusions about the tool and its performance adequately supported by the findings presented in the article?

Yes

Competing Interests: No competing interests were disclosed.

Reviewer Expertise: Bioimage analysis

We confirm that we have read this submission and believe that we have an appropriate level of expertise to confirm that it is of an acceptable scientific standard, however we have significant reservations, as outlined above.

Author Response 29 Jan 2021

Laurent Thomas, Acquirer Imaging GmbH, Heidelberg, Germany

We thank the reviewers for their constructive criticism and helpful suggestions. Since the first manuscript version we implemented most of the previous suggestions to improve the plugins, which are readily available by updating the plugins in Fiji. We further address each point in details below.

- *The authors describe how the output of their plugins could be consumed by a machine*

learning workflow in KNIME. While the workflow in KNIME is very interesting and relevant we feel at the same time that it is slightly out of scope and we thus suggest moving this to the supplemental material. The reason is that we would expect the main readers of this publication and also the main users of the plugins to be familiar with the ImageJ ecosystem, but not necessarily with KNIME. In addition, if the final aim of the users is to execute a workflow in KNIME there may be other annotation tools available and the authors should clarify how their tool is superior to potential other solutions for image annotation.

Following this observation, we have simplified the article such that the examples of visualization and analysis (including the deep learning workflows) are mentioned in the “Use cases” and summarized in the new Figure 4. Details about the deep learning implementation and individual workflow requirements is covered in Supplementary Figures (available on the Zenodo repository) and in the online documentation of the GitHub repository.

Regarding annotation tools, there are several software packages available for this specific purpose of ground-truth annotations for supervised machine learning. However, as now explained in introduction they are sometimes not adapted to scientific image formats, while most of them are not integrated into user-oriented scientific image analysis software, and thus require additional installation or configuration. The qualitative annotations plugins instead can be easily installed in ImageJ/Fiji, which supports a large variety of scientific image formats, and is a familiar software environment for most life-scientists. Another advantage of the presented plugins is the support for multiple descriptive keywords per image instance, which is not systematically available with other software packages. Ground-truth category annotations for image-classification or object-detection is one possible application of the qualitative annotation tools. We chose KNIME to illustrate the training of an image-classifier, because it is a relatively accessible data-analysis platform for non-expert users, compared to a programming language. It supports advanced image-processing functionalities, was readily used for bioimage analysis and allows rapid prototyping and customization thanks to its interactive graphical user interface. However, more advanced users can surely reproduce this workflow in a programming language of their choice.

- *Along those lines, it would be great to also present (maybe also in the supplemental material) a follow up workflow that can be executed entirely within ImageJ. Specifically, it would be great to see an example of how the annotated ROIs could be used, e.g. to train a machine learning model. However, we don't know how feasible this is and it is really just a suggestion.*

It is potentially possible to train a machine learning or deep learning model directly in ImageJ/Fiji e.g. thanks to respectively the Weka or TensorFlow integration in Fiji. However, this would represent a significant amount of work, and would not offer as much flexibility as the reported KNIME workflows for deep learning, which can be rapidly adapted in KNIME's graphical user-interface.

As an illustration of use case in Fiji, we propose instead an additional plugin for data-visualization relying on the JFreeChart library (available with Fiji). This new plugin generates a pie chart from a table column, to visualize the category distribution. It is available in the same menu “Qualitative Annotation” under “PieChart from category column”– (see Figure

5.B and supplementary figure 2).

The plugins support any column of data from a table opened in Fiji and is macro recordable. It is thus not limited to annotation tables generated with the presented plugins. To our knowledge, the pie chart visualization was not previously available to users via the Fiji menus or plugins. Therefore, we believe that it can be an interesting complement to the annotation plugins.

Technical issues and suggestions for the plugins

- *There are hotkeys (F1, F2, ...) available for the single class plugin, which is great as it speeds up annotation. However, we did not manage to have them working on MacOS (10.14.6)*

At the time of the first manuscript version, the plugins were only tested on windows but are expected to run similarly across platforms. Hotkeys might be an exception, for instance on Linux we also observe a different behaviour (see [Issue #9 · LauLauThom/Fiji-QualiAnnotations \(github.com\)](#)).

We don't have a MacOS system at hand for testing, but we invite the reviewers to follow up in the dedicated issue thread on GitHub mentioned above.

The hotkeys are sometimes not responsive also on Windows. Clicking the plugin window or one of the buttons usually allows reactivating the hotkey functionality.

That said, we believe this is not a major issue preventing the use of the plugins, and we hope to fix it in later versions of the plugins.

- *[..] if one has many classes one would have to count the buttons to figure out which hotkey belongs to which class. We thus suggest that the hotkeys are either written on the category buttons or appears as pop-up messages when a user hovers over the corresponding button.*

We have updated the button plugin with buttons from the java swing package (previously java awt buttons), which support pop-up messages. The associated hotkey is thus now displayed when the mouse is hovered over a category button. With this new button class, the layout of the GUI might be slightly impaired when a new button is added by clicking the "Add new category". When this happens, the window can be resized manually to make sure all GUI elements fits in the window. We will fix this issue in later plugin versions if we can identify the source of the problem.

- *Currently, one cannot add additional categories during the labelling process. We think it would be great to be able to (e.g., via a new button: [Add category]), because images may contain phenotypes that one did not think about upfront.*

We added a "Add new category" button to the *single class (button)* and *multi-class (checkboxes)* plugins, which allows updating the plugin interface with new categories on the fly.

We did not add this functionality to the dropdown plugin, which would require providing multiple items (label and list of choices). However, with any of the annotation plugins, the plugin interface can be closed and restarted while keeping the current annotation table opened in ImageJ/Fiji. Although not as straightforward as the "Add new category" option, the annotation can thus be interrupted to update the graphical interface and resumed at a later time without losing information.

- *One can annotate the same image, slice and ROI multiple times, where each annotation will add another row to the table. As a consequence, the table contains conflicting annotations for the same image region. We think it would be (much) better if the annotation was replaced, at least optionally.*

We have considered the possibility to add an option (checkbox) to replace previous annotations of identical image or ROI (see dedicated [branch](#) on GitHub). However, when “run Measure” is selected, a table row is anyway added to the result table with the quantitative measurements, before the descriptive keywords are added to the table row. Accounting for this alternative behaviour would further complicate the code (see function [defaultActionSequence\(\)](#)). A more elegant solution would be to have a way to store the measurement into a variable first, which is apparently not the case with the current ImageJ1 package.

We thus might address this issue in later releases of the tools.

- *The possibility to continue an annotation by loading a table and renaming it to “Annotations” is a useful feature, but we think the current implementation has a limitation in a sense that the class names are not read from the table, but rather taken from the last annotation session. This can be an issue if users work on several annotations. We suggest adding an option to fetch the set of category names from the table.*

We added an option to fetch the categories names from a table currently opened in ImageJ/Fiji.

If the table contains a column name “Category” then the set of categories is initialized from this column. Otherwise, the set of categories is initialized from the column headers, by excluding the headers for the measurement and file information columns. However, the number of categories will still be taken into account.

Besides, with ImageJ versions above 1.53g, the annotations will be appended to/read from any active table window. Below this version, the annotations are appended to/read from table windows entitled “Annotations” or “Annotations.csv”.

- *The logic of jumping to the next image slice in case of hyperstacks (4D or 5D data) is not completely obvious. We suggest adding a drop-down menu next to the “auto next slice” checkbox, where the user can select the dimension (z, c, or t) along which to automatically progress.*

As suggested, we added a dropdown menu next to the “Auto next slice” checkbox to specify the dimension to explore with hyperstacks. This dropdown menu is shown when “stack” is selected as browsing mode in the initial configuration window (see below).

- *Currently, annotating a set of images that cannot be loaded into a hyperstack, e.g. because they may have different sizes or dimensionality, is possible but a bit tedious. We suggest adding a modality where the user would select one folder on the disk and then the plugin would automatically open and close the images in the folder one-by-one, allowing them to be annotated one-by-one. The additional advantage of this modality also is that loading different images into a hyperstack is not something that every user may know how to achieve. We suggest adding [previous] and [next] buttons to this modality, allowing one to correct a previous annotation in case a mistake has been made.*

We added a new setting “Browsing mode” in the initial configuration window, which can be set to “stack” or “directory”. The “stack” mode corresponds to the previously available

modality and is well adapted to the annotations of image slices in stacks or hyperstacks. With the “directory” mode, ticking the “Auto next slice/image file” will switch to the next image file in the directory of the currently opened image file. This allows annotating a list of files in a directory as proposed. In “directory” browsing mode the annotation interface also has “previous” and “next image file” buttons, as suggested.

- *ImageJ ROIs are not a “cross platform standard”. In order to enable follow up workflows outside ImageJ it may be thus useful to store the bounding box coordinates for rectangular ROIs in the table.*

Bounding box coordinates for ROIs can be readily recovered by activating the “Bounding Rectangle” option in Analyze > Set measurements and activating the “run Measure” option in the plugin configuration.

For other ROI shapes, such as polygon and free shapes, the number of summits/coordinates is variable. There is thus the option to have on column per coordinates, which is not ideal as it can potentially yield lots of columns. The second option is to have a single column containing the coordinates as a list or another type of collection. However, there is no real standard convention for polygons and other free ROI shapes coordinates encoding. Some further discussion would thus be needed before implementing such functionality (in a dedicated GitHub issue for instance).

- *When clicking the Help button on one of the annotation UIs the UI was closing, which we think is probably a bug (tested on Mac).*

It is expected that clicking the help button should close the UI, but it should also open the page of the GitHub repository in the default browser. If this is not the case this might be an OS-dependant error, like for the hotkeys. We have opened a dedicated issue thread on GitHub ([Issue #22 · LauLauThom/Fiji-QualiAnnotations \(github.com\)](#)).

Competing Interests: No competing interests were disclosed.

Reviewer Report 13 November 2020

<https://doi.org/10.5256/f1000research.29676.r73675>

© 2020 Eglinger J. This is an open access peer review report distributed under the terms of the [Creative Commons Attribution License](#), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.



Jan Eglinger

Friedrich Miescher Institute for Biomedical Research (FMI), Basel, Switzerland

The authors introduce a new set of Fiji plugins that allows user-friendly annotation of images in two ways: 1) tagging entire images with one or several keywords/classes, and 2) tagging selected regions within an image.

Thanks to the integration with the Fiji/ImageJ framework (installation via an ImageJ update site, graphical user interface using the built-in functionality provided by Fiji), the plugins are indeed

simple to install and use, and simplify the process to get tabular data from a manual annotation task.

In the introduction, the authors state that there is "no widespread solution available [...] for the assignment of multiple descriptive keywords to images". In my opinion, this statement demonstrates a lack of research, as a simple online search for "image annotation tool" reveals many solutions used in the field of Computer Vision to annotate ground truth for machine learning datasets (both for image classification and for bounding box annotations). While some of these tools might be more commonly used than others, I believe they should have at least included a few of them in a thorough comparison, such as e.g. LabelMe [1], MakeSense.AI [2], LabelImg [3], VGG Image Annotator [4], or Scalabel [5]. The only other available tool they cite is ACQUIFER Manual Annotation Tool, a software created by (partially) the same authors, available on request only.

Nevertheless, the set of plugins introduced here certainly fills a gap for users of Fiji/ImageJ in particular, which means a lot of scientists using it to answer biological questions with (microscopy) image data, who will welcome a way to generate annotation data from within a framework that they already use and know.

Specifically for biological applications however, I see a potential limitation of the use of this plugin. With modern microscopy datasets often being multi-dimensional, region annotations (bounding-box or pixel-wise) would benefit from being 3-dimensional to reflect a volume of interest. By being tied very much to the legacy user interface of Fiji (built-in ROI support is 2D only), the Qualitative Annotations plugin will be of very limited use for true 3D applications, since any annotation can only be done slice-wise, in the current state. Other plugins, such as the 3D ImageJ Suite, offer support for 3-dimensional object annotations, but lack an easy way to add keywords/class annotations.

Another current limitation of the Qualitative Annotations plugin is the absence of ways to review and/or refine an annotation after it has been added to the results table.

Lastly, since the authors include KNIME workflows in their article to illustrate further processing of the annotation data (e.g. for training of machine learning models), I am surprised that they didn't also explore ways to do the manual annotation within a KNIME workflow, in order to include the annotation process in their "complete pipeline for image classification".

The KNIME Image Processing [6] extension offers the 'Interactive Annotator' and 'Interactive Labeling Editor' nodes that can be used to annotate image regions. For interactive sequential classification of images, the 'Active Learning Loop' nodes offer an excellent alternative. These options should be mentioned for comparison when discussing the performance of the Qualitative Annotations tools introduced here.

Is the rationale for developing the new software tool clearly explained?

Partly

Is the description of the software tool technically sound?

Yes

Are sufficient details of the code, methods and analysis (if applicable) provided to allow

replication of the software development and its use by others?

Yes

Is sufficient information provided to allow interpretation of the expected output datasets and any results generated using the tool?

Yes

Are the conclusions about the tool and its performance adequately supported by the findings presented in the article?

Partly

Competing Interests: No competing interests were disclosed.**Reviewer Expertise:** Image Processing, Software Development, Microscopy**I confirm that I have read this submission and believe that I have an appropriate level of expertise to confirm that it is of an acceptable scientific standard, however I have significant reservations, as outlined above.**

Author Response 29 Jan 2021

Laurent Thomas, Acquifer Imaging GmbH, Heidelberg, Germany

Dear Jan, thank you for your feedback. The tools are indeed mostly targeting the annotation of 2D images, or from image slices in multi-dimensional images. They are not designed to address qualitative annotations of 3D volumes, which as you pointed out have limited support in ImageJ/Fiji.

In the introduction, the authors state that there is "no widespread solution available [...] for the assignment of multiple descriptive keywords to images". In my opinion, this statement demonstrates a lack of research, as a simple online search for "image annotation tool" reveals many solutions used in the field of Computer Vision to annotate ground truth for machine learning datasets (both for image classification and for bounding box annotations). While some of these tools might be more commonly used than others, I believe they should have at least included a few of them in a thorough comparison, such as e.g. LabelMe [1], MakeSense.AI [2], LabelImg [3], VGG Image Annotator [4], or Scalabel [5].

Regarding other image annotations solutions mentioned above, most of them target RGB colour images and thus are not well adapted to scientific images or file format (e.g 16-bit multi-dimensional tiff). Besides, they mostly target single-category annotations of images or images regions for classification, object-detection or semantic segmentation and usually do not allow the assignment of multiple qualitative keywords to a single image instance. Moreover, these are often standalone software, and thus for non-experts likely not as accessible as integrated solutions in bioimage analysis software such as ImageJ/Fiji. We have reformulated the introduction to include a brief comparison with existing solutions and their limitations. Although the plugins presented in this article can indeed be used for the tasks mentioned above, we still believe that the annotations with multiple keywords was

a truly missing functionality and can be of major interest to describe complex biological phenotypes.

Another current limitation of the Qualitative Annotations plugin is the absence of ways to review and/or refine an annotation after it has been added to the results table.

Regarding this limitation, again due to its tight integration in ImageJ/Fiji, individual table cells of the result table cannot be edited interactively. However, if a mistake was done while annotating, the concerned table rows can be selected and deleted, and the annotation repeated for the images. We have added a sentence in the paragraph implementation mentioning this possibility. The table could also be edited using the macro language, although it is likely not a probable use-case.

After annotations, the csv file containing the results can be edited in any software supporting csv such as spreadsheet software or text editors.

Lastly, since the authors include KNIME workflows in their article to illustrate further processing of the annotation data (e.g. for training of machine learning models), I am surprised that they didn't also explore ways to do the manual annotation within a KNIME workflow, in order to include the annotation process in their "complete pipeline for image classification".

The KNIME Image Processing [6] extension offers the 'Interactive Annotator' and 'Interactive Labeling Editor' nodes that can be used to annotate image regions. For interactive sequential classification of images, the 'Active Learning Loop' nodes offer an excellent alternative. These options should be mentioned for comparison when discussing the performance of the Qualitative Annotations tools introduced here.

KNIME indeed offers powerful functionalities for ground-truth annotations of image regions or image category, however the annotations are here also limited to a single category per ROI or image. Besides, despite its advanced data and image-analysis functionalities, we believe that it is not the most intuitive software for interactive data-annotation, especially for users not already familiar with KNIME. Still, we have included KNIME in the list of existing annotation solutions compatible with bioimages, in the introduction.

Competing Interests: No competing interests were disclosed.

The benefits of publishing with F1000Research:

- Your article is published within days, with no editorial bias
- You can publish traditional articles, null/negative results, case reports, data notes and more
- The peer review process is transparent and collaborative
- Your article is indexed in PubMed after passing peer review
- Dedicated customer support at every stage

For pre-submission enquiries, contact research@f1000.com

F1000Research