

SOFTWARE METAPAPER

MiToBo – A Toolbox for Image Processing and Analysis

Birgit Möller¹, Markus Glaß², Danny Misiak² and Stefan Posch¹

¹ Institute of Computer Science, Faculty of the Natural Sciences III, Martin Luther University Halle-Wittenberg, Halle (Saale), Germany
birgit.moeller@informatik.uni-halle.de

² Institute of Molecular Medicine, Medical Faculty, Martin Luther University Halle-Wittenberg, Halle (Saale), Germany

Corresponding author: Birgit Möller

MiToBo is a toolbox and Java library for solving basic as well as advanced image processing and analysis tasks. It features a rich collection of fundamental, intermediate and high-level image processing operators and algorithms as well as a couple of sophisticated tools for specific biological and biomedical applications. These tools include operators for elucidating cellular morphology and locomotion as well as operators for the characterization of certain intracellular particles and structures.

MiToBo builds upon and integrates into the widely-used image analysis software packages ImageJ and Fiji [11, 10], and all of its operators can easily be run in ImageJ and Fiji via a generic operator runner plugin. Alternatively MiToBo operators can directly be run from command line, and using its functionality as a library for developing own applications is also supported. Thanks to the Alida library [8] forming the base of MiToBo all operators share unified APIs fostering reusability, and graphical as well as command line user interfaces for operators are automatically generated. MiToBo is available from its website <http://www.informatik.uni-halle.de/mitobo>, on Github, via an Apache Archiva Maven repository server, and it can easily be activated in Fiji via its own update site.

Keywords: Image Analysis; Image Processing; Toolbox; Java Library; ImageJ; Fiji; Microscope Images; Biomedical Imaging; Wound Closure; Actin Microfilament; Morphology; Protein Profile

(1) Overview

Introduction

The Microscope Image Analysis Toolbox MiToBo is an extendable collection of basic as well as more sophisticated image processing operators. Additionally, it offers several pipelines for analyzing specific microscopy based experimental setups [1, 2, 5, 6].

Although the focus is on biomedical image processing, it is also applicable to other image analysis tasks. Since it integrates into the popular image processing platforms ImageJ and Fiji, respectively, it automatically benefits from the capability of these programs to process a huge variety of input image types. MiToBo provides an elaborated infrastructure for development and rapid prototyping. Besides its usage in biomedical research, e.g. [4, 12, 13], it is also actively used for teaching. Its implementation focuses on easy and unified (re-)usability as well as modularity. MiToBo operators can be configured to be executed via a graphical user interface as well as from command line. Furthermore, the integrated graphical programming platform Grappa [3] can be used to create complex analysis pipelines by simply combining existing operators inside a workbench area.

Algorithms and Applications Basic image processing algorithms implemented in MiToBo comprise, for example, linear and non-linear filters, morphological operations, image arithmetics, connected component labeling or several automatic thresholding methods.

Among the more complex operators are segmentation algorithms like snakes or level set methods, which are highly configurable, e.g., by offering different energy functionals or initialization methods [7]. Furthermore, MiToBo contains operators for contrast enhancement, illumination correction or feature extraction as well as operators to segment cell nuclei or boundaries, for example. Some processing pipelines tailored for the analysis of specific biomedical experiments are also included in MiToBo and its number increases constantly. The ScratchAssayAnalyzer [2] is designed for quantifying the development of the cell-free area in scratch/wound closure assays. With the ActinAnalyzer2D [6] the distribution of actin microfilament patterns inside cells can be quantified and compared. The NeuronAnalyzer2D [5] segments fluorescently labeled neurons, characterizes the cell morphology and extracts profiles from labeled molecules within the cell. The MTBCellCounter [1] is dedicated to semi-automatic

labeling and counting of spot-like structures and small cells. The migratory behavior of fluorescently labeled cells can be analyzed using the CellMigrationAnalyzer. The MiToBo Cell Image Analyzer (MiCA) [9] is conceptualized for segmenting cell boundaries and sub-cellular particles in fluorescence microscopy images. In total, the current release (MiToBo 1.7) contains more than 150 distinct operators.

General features MiToBo's capacities to supply easy usability, generic user interfaces and graphical programming for image processing applications are mainly taken from Alida (see Section Implementation). Building on Alida's concept, each image data analysis action is realized in terms of an operator where each operator implements a common interface definition. As furthermore their input and output parameters are accessible in a standardized manner each operator can be invoked in a uniform way. This is exploited to automatically generate command line and graphical user interfaces for each operator implemented in MiToBo without any additional burden on the programmer (see **Figs. 1** and **2** for examples). The graphical user interface includes the option for batch mode operation which, e.g., allows to apply an operation to all files of a given directory or for a range of parameter values.

While developing image analysis pipelines and also with regard to the reproducibility of image analysis results it is usually of interest to thoroughly document parameter settings and analysis steps. To simplify such logging procedures MiToBo offers an automatic process documentation where the invocation of an operator not only produces image analysis results, but at the same time registers within the framework along with all input and output objects as well as parameters settings. These data acquired during an image analysis process and the order of operator calls form a directed graph containing all relevant information for later reconstruction or verification of the analysis procedure. This processing graph is made explicit in terms of an XML representation which can be visually explored with appropriate graphical frontends like Chipory (see the Alida home page) or might be stored in data bases for archival purposes. **Fig. 3** gives an example processing graph for an image processing pipeline.

In most cases several image analysis operations, e.g., filters, thresholding, or segmentation algorithms, have to be applied in a sequential, parallel or nested fashion to achieve desired image processing results. The graphical programming editor Grappa included in MiToBo facilitates the design of such processing pipelines or workflows

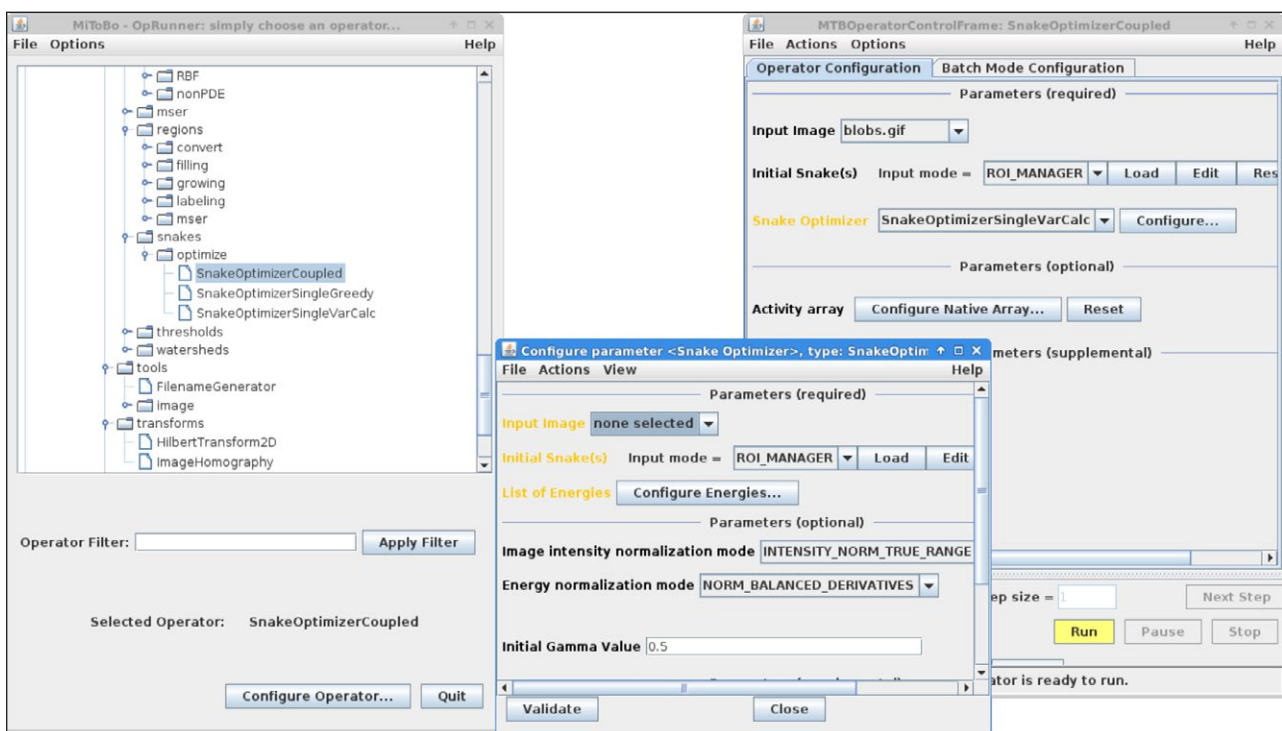


Figure 1: The main window of Alida's operator runner which allows to select any of the available MiToBo image processing operators (top left), and two automatically generated control and configuration windows for operators (right and bottom).

```
java de.unihalle.informatik.Alida.tools.ALDOpRunner GaussFilter \
  inputImg=cell.tif sigmaX=1.4 resultImg=cell-g3.tif
```

Figure 2: Example call of an image analysis operator from command line. The operator GaussFilter is executed to apply a Gaussian filter with standard deviation of 1.4 to the input image cell.tif, generating the filtered result image cell-g3.tif.

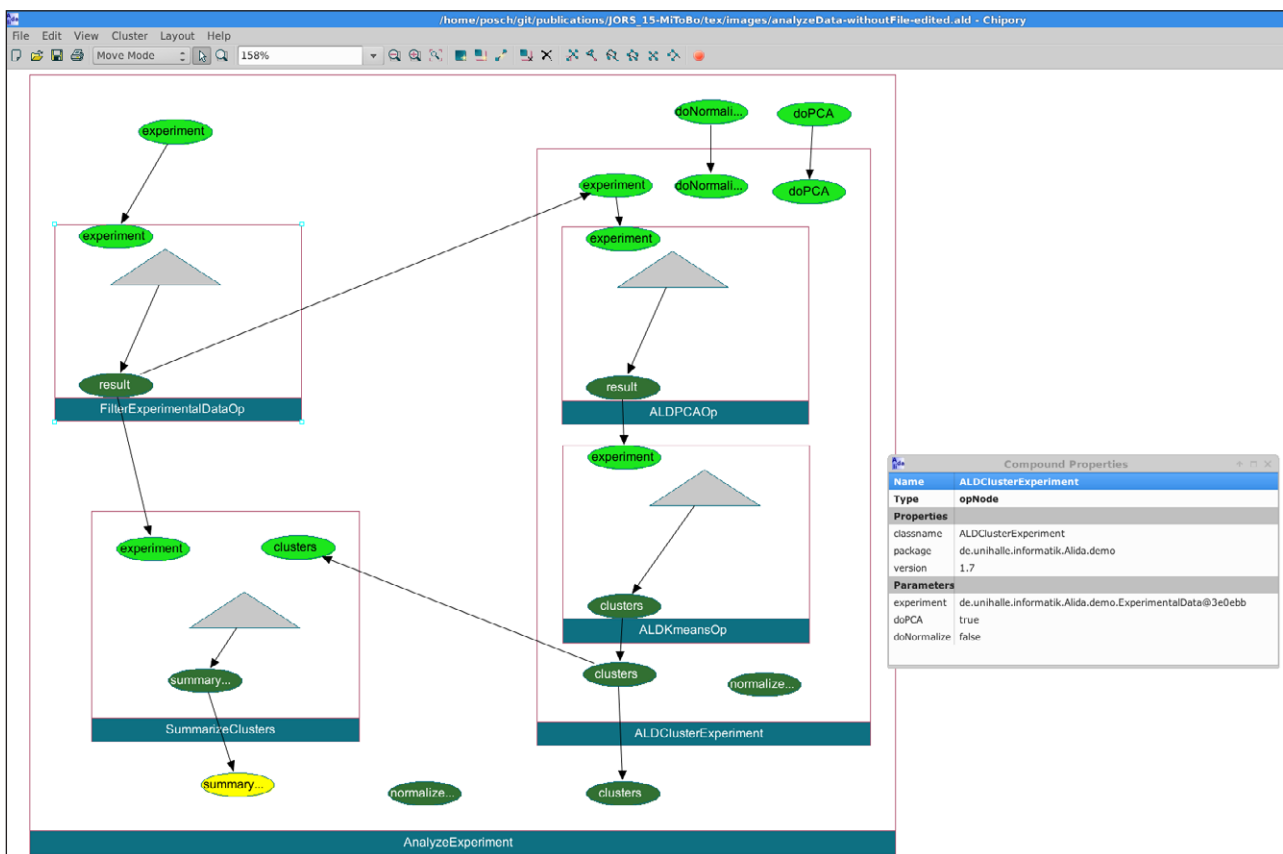


Figure 3: Example processing graph representing the history of operations for producing the image data object shown as yellow ellipse. Each operator invocation is represented by a blue rectangle. Light and dark green ellipses are input and output ports of operators, gray triangles depict data ports representing newly generated data. The rightmost window displays the parameters and software version for the operator `ALDClusterExperiment` during execution.

in an intuitive way. All image processing algorithms implemented as an Alida operator are right away available as operator nodes in Grappa and form the building blocks for workflows. Their input and output parameters may be connected interactively by user-friendly mouse actions where compatibility of data types is enforced and automatic data type conversion is applied if appropriate converts are available (see **Fig. 4** for an example).

Implementation

The MiToBo library, implemented in Java, uses and strongly builds on Alida, an *Advanced Library for Integrated Development of Data Analysis Applications*, which defines and implements a concept for designing libraries and toolkits in data analysis (see [8] and <http://www.informatik.uni-halle.de/alida>).

On the other hand MiToBo builds upon ImageJ, a Java toolkit for image processing and analysis [11] widely used, but not confined to biomedical applications. ImageJ provides the user with flexible means to interactively process, modify, and display images, with a large variety of basic built-in image processing operations, and also with a huge collection of optional plugins downloadable from the web. While MiToBo is compatible with ImageJ it significantly improves data access and exchange between more complex processing modules exploiting Alida's operator

concept. Furthermore, MiToBo extends ImageJ's image data types and provides more sophisticated data structures for, e.g., regions and contours. MiToBo's facilities are available within ImageJ either via the MiToBo OpRunner plugin or as dedicated ImageJ plugins. The former allows to configure and run each data analysis procedure implemented as an Alida operator (c.f. Subsection General features and **Fig. 1**).

Quality Control

The MiToBo core library and the plugin collection currently consist of about 500 Java class files. Approximately 10% of them regularly undergo automatic tests based on JUnit test classes, while the bigger part of the testing is done by manual verification. MiToBo is developed on different operating systems (e.g., Ubuntu Linux 12.04 and 14.04, Microsoft Windows 7 and 8). While the officially supported Java version is still 1.6 (for compatibility reasons with ImageJ/Fiji), developers and users also run MiToBo frequently with additional Java versions (OpenJDK and Oracle Java 1.7/1.8) to ensure a maximum degree of portability.

MiToBo's operators and applications are usually developed in close collaboration with users mainly from Computer Science, Biology and Medicine. They particularly provide feedback on software usability and the plausibility of computed results. Thanks to the update

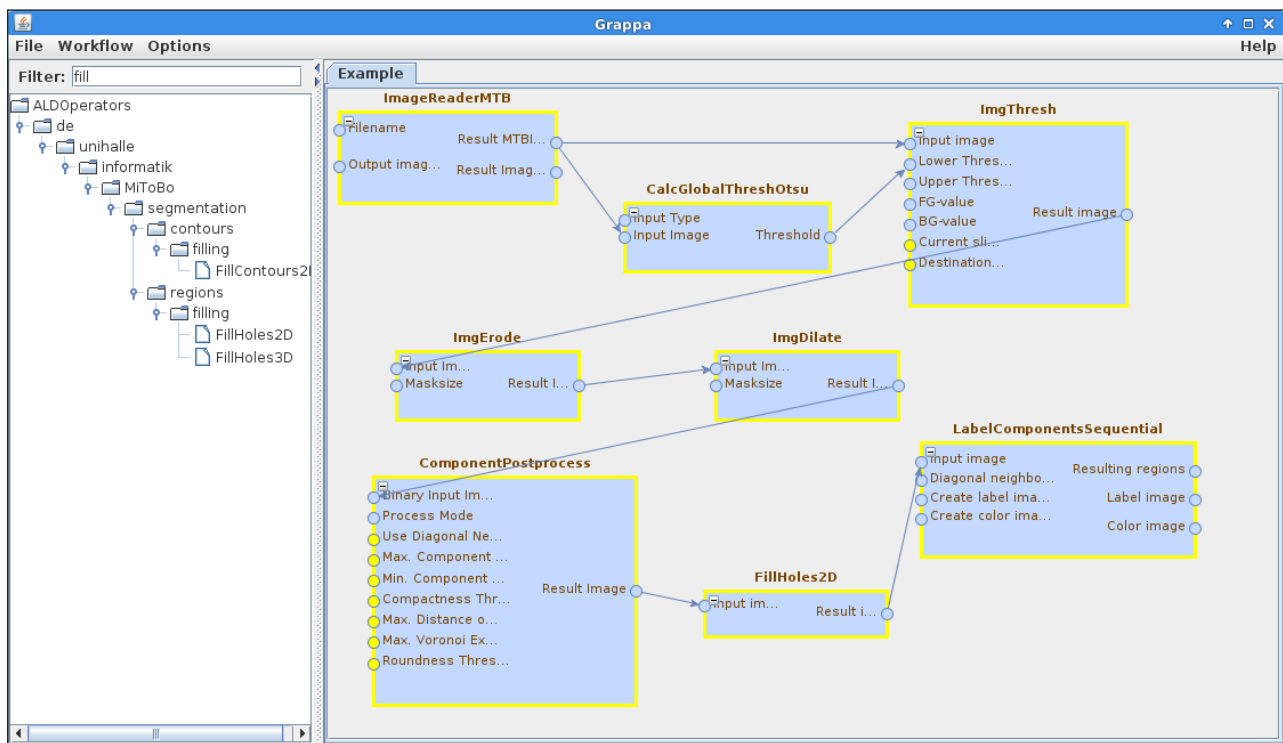


Figure 4: Screen shot of Alida's graphical programming editor Grappa. The operator chooser panel on the left allows to choose from all available operators. The main panel shows a workflow composed of various operators connected by edges defining the data flow. Additional parameters may be set via the same automatically generated GUIs available within the operator runner.

mechanisms integrated in Fiji, such feedback is promptly available resulting in rather short prototyping, software development and bugfix cycles.

(2) Availability

Project Homepage

<http://www.informatik.uni-halle.de/mitobo>

Operating System

MiToBo and ImageJ/Fiji run on different versions of Unix, Linux, Mac OS, and Windows.

Programming Language

Java, version 1.6

Additional system requirements

None.

Dependencies

MiToBo builds upon ImageJ/Fiji and its operators can best be run in ImageJ or Fiji, but usage as a library and on command line is also possible. For usage as a library or with ImageJ all required dependencies are shipped as part of the MiToBo distribution. In Fiji MiToBo is available via an update site which automatically resolves the dependencies (<http://sites.imagej.net/MiToBo/>). To activate MiToBo in Fiji the entry of MiToBo's update site in Fiji's built-in list of available sites just needs to be checked and the updater run (for further details refer to the Fiji documentation at http://fiji.sc/Update_Sites).

List of Contributors

Markus Glaß

Oliver Greß

Danny Misiak

Birgit Möller

Stefan Posch

Software Locations:

• Archive:

Name: Zenodo Research Archive

Title: MiToBo – A Microscope Image Analysis Toolbox: v1.7

URL: <https://zenodo.org/record/31364>

Persistent identifier: <http://dx.doi.org/10.5281/zenodo.31364>

Licence: GNU General Public License, Version 3

Publisher: Birgit Möller, Stefan Posch, Markus Glaß, Danny Misiak

Artifact Version: 1.7

Date published: 09/24/2015

• Code repository:

Name: Github

Titles: mitobo / mitobo-plugins

Identifiers: <https://github.com/mitobo-hub/mitobo>

<https://github.com/mitobo-hub/mitobo-plugins>

Licence: GNU General Public License, Version 3

Publisher: Birgit Möller, Stefan Posch, Markus Glaß, Danny Misiak

Date published: 06/28/2015

- Maven repository server:
Name: Apache Archiva Repository Server
Title: de.unihalle.informatik.MiToBo
Identifier: <https://moon.informatik.uni-halle.de/archiva/>
Licence: GNU General Public License, Version 3
Publisher: Birgit Möller, Stefan Posch, Markus Glaß, Danny Misiak
Date published: snapshots and releases are continuously published

Language

English.

(3) Reuse Potential

One of the explicit goals of MiToBo is reusability. As pointed out in Section 1 MiToBo includes implementations of algorithms and operators with varying complexity. Most algorithms and concepts are independent of concrete applications. Expert as well as non-expert users can easily apply them to image processing and analysis problems in a large variety of domains, not necessarily restricted to microscopy images or biomedical scenarios. In addition, also the more specialized operators built on top of these basic ones and being integral part of MiToBo demonstrate the reuse potential. They cover applications from biology, biomedicine and ecology. Complementary the set of different user interfaces which Alida and MiToBo offer, i.e., graphical and command line interfaces, support efficient reuse of MiToBo not only with regard to concrete image processing problems, but also in different usage scenarios ranging from rapid-prototyping and development with tied user-developer feedback loops to scripting and high-throughput processing.

Finally, reusability of the MiToBo toolbox and its operators is not restricted to the user side. Also for developers reusing operators and implemented algorithms is easy. Based on Alida's concepts for generic implementation of functionality and user interfaces [8] all operators provide unified interfaces for reusing them on the code level in a modular fashion. Likewise new operators and also data types can easily be added to MiToBo's core and take immediate advantage of the rich set of built-in features in Alida and MiToBo. To ease the first steps with MiToBo for own developments a pre-configured Maven project containing a demo operator can be found on MiToBo's project website.

Competing Interests

The authors declare that they have no competing interests.

References

1. Franke, L, Storbeck, B, Erickson, J L, Rödel, D, Schröter, D, Möller, B and Schattat, M H 2015 The 'MTB Cell Counter' a versatile tool for the semi-automated quantification of sub-cellular phenotypes in fluorescence microscopy images. A case study on plastids, nuclei and peroxisomes. *Journal of Endocytosis and Cell Research*, 26: 31–42.
2. Glaß, M, Möller, B, Zirkel, A, Wächter, K, Hüttelmaier, S and Posch, S 2012 (September) Cell migration analysis: segmenting scratch assay images with level sets and support vector machines. *Pattern Recognition*, 45(9): 3154–3165. DOI: <http://dx.doi.org/10.1016/j.patcog.2012.03.001>
3. Kirchner, S, Posch, S and Möller, B 2012 (October) Graphical programming in Alida and ImageJ 2.0 with Grappa. In *Proc. of ImageJ User & Developer Conference*. Mondorf-les-Bains, Luxembourg, pp. 138–143.
4. Komaravolu, R K, Adam, C, Moonen, J-R A J, Harmsen, M C, Goebeler, M and Schmidt, M 2015 Erk5 inhibits endothelial migration via KLF2-dependent down-regulation of PAK1. *Cardiovascular Research*, 105(1): 86–95. DOI: <http://dx.doi.org/10.1093/cvr/cvu236>
5. Misiak, D, Posch, S, Lederer, M, Reinke, C, Hüttelmaier, S and Möller, B 2014 (March) Extraction of protein profiles from primary neurons using active contour models and wavelets. *Journal of Neuroscience Methods*, 225: 1–12. DOI: <http://dx.doi.org/10.1016/j.jneumeth.2013.12.009>
6. Möller, B, Piltz, E and Bley, N 2014 (August) Quantification of actin structures using unsupervised pattern analysis techniques. In *Proceedings of Int. Conf. on Pattern Recognition (ICPR)*. Stockholm, Sweden, IEEE, pp. 3251–3256. DOI: <http://dx.doi.org/10.1109/icpr.2014.560>
7. Möller, B and Posch, S 2012 (May) Comparing active contours for the segmentation of biomedical images. In *Proc. of IEEE International Symposium on Biomedical Imaging: From Nano to Macro (ISBI)*, IEEE Catalog No.: CFP12BIS-CDR ISBN: 978-1-4577-1856-4, pp. 736–739, Barcelona, Spain. DOI: <http://dx.doi.org/10.1109/isbi.2012.6235653>
8. Möller, B and Posch, S 2013 (May) A framework unifying the development of image analysis algorithms and associated user interfaces. In *Proc. of 13th IAPR International Conference on Machine Vision Applications (MVA)*. Kyoto, Japan, pp. 447–450.
9. Möller, B, Stöhr, N, Hüttelmaier, S and Posch, S 2010 (August) Cascaded segmentation of grained cell tissue with active contour models. In *Proceedings of International Conference on Pattern Recognition*. Istanbul, Turkey, IEEE, pp. 1481–1484. DOI: <http://dx.doi.org/10.1109/icpr.2010.366>
10. Schindelin, J, Arganda-Carreras, I, Frise, E, Kaynig, V, Longair, M, Pietzsch, T, Preibisch, S, Rueden, C, Saalfeld, S, Schmid, B, et al. 2012 Fiji: an open-source platform for biological-image analysis. *Nature methods*, 9(7): 676–682. DOI: <http://dx.doi.org/10.1038/nmeth.2019>
11. Schneider, C A, Rasband, W S and Eliceiri, K W 2012 NIH Image to ImageJ: 25 years of image analysis. *Nature Methods*, 9(7): 671–675. DOI: <http://dx.doi.org/10.1038/nmeth.2089>

12. **Wolf, A, Rietscher, K, Glaß, M, Hüttelmaier, S, Schutkowski, M, Ihling, C, Sinz, A, Wingenfeld, A, Mun, A and Hatzfeld, M** 2013 Insulin signaling via Akt2 switches plakophilin 1 function from stabilizing cell adhesion to promoting cell proliferation. *Journal of Cell Science*, 126(Pt 8): 1832–1844. DOI: <http://dx.doi.org/10.1242/jcs.118992>
13. **Zirkel, A, Lederer, M, Stöhr, N, Pazaitis, N and Hüttelmaier, S** 2013 IGF2BP1 promotes mesenchymal cell properties and migration of tumor-derived cells by enhancing the expression of LEF1 and SNAI2 (SLUG). *Nucleic Acids Research*, 41(13): 6618–6636. DOI: <http://dx.doi.org/10.1093/nar/gkt410>

How to cite this article: Möller, B, Glaß, M, Misiak, D and Posch, S 2016 MiToBo – A Toolbox for Image Processing and Analysis. *Journal of Open Research Software* 4: e17, DOI: <http://dx.doi.org/10.5334/jors.103>

Submitted: 06 November 2015 **Accepted:** 08 March 2016 **Published:** 28 April 2016

Copyright: © 2016 The Author(s). This is an open-access article distributed under the terms of the Creative Commons Attribution 4.0 International License (CC-BY 4.0), which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited. See <http://creativecommons.org/licenses/by/4.0/>.