SoilJ – Technical Manual

Version 0.0.4   2019-10-11

John Koestel

Introduction

SoilJ is a plugin for the JAVA-based, free and open image processing software ImageJ (Schneider, Rasband, et al., 2012). It is planned to be published in the framework of the FIJI plugin-bundle distribution (Schindelin, Arganda-Carreras, et al., 2012) on GitHub.

SoilJ is tailor-made for semi-automated image processing and analyses of 3-D X-ray images of soil columns, which allows for the rapid analyses of a large number of images. It includes modules for

- column outline detection,
- intensity-bias correction,
- image segmentation,
- detection of the top and bottom topography of the soil column,
- extraction of the particulate organic matter and roots,
- extraction of the pore-size distribution and
- pore-space morphology analyses.

The morphology analyses module makes abundant use of the ImageJ plugin BoneJ (Doube, Klosowski, et al., 2010).

SoilJ is published under the GNU General Public License:

SoilJ is a collection of ImageJ plugins for the semi-automatized processing of 3-D
X-ray images of soil columns
Copyright 2014 2015 2016 2017 John Koestel

This program is free software: you can redistribute it and/or modify it under the
terms of the GNU General Public License as published by the Free Software Foundation,
either version 3 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY
WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A
PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this
program. If not, see <http://www.gnu.org/licenses/>.

If you are using SoilJ in a scientific study, please take care that the used software components are properly cited, i.e. as indicated in the plugin module.

Installation

Install ImageJ using the Fiji distribution, which is available on http://fiji.sc.

Automatic installation

(will be added, maybe soon)

Manual installation

SoilJ requires several ImageJ/Fiji plugins to work properly. Most of them are installed using the in-built Fiji Updater. To install the required plugins, go to *Help - > Update*. Then wait that Fiji has updated itself. This may require that you restart Fiji in order to complete the base update. In this case, restart Fiji and again click on *Help - > Update.* In the *ImageJ Updater* window that is now open, click *Manage update sites.* In the next window, make sure that the following plugins are checked:

ImageJ	http://update.imagej.net/
Fiji	http://update.fiji.sc/
Java-8	http://sites.imagej.net/Java-8/
3D ImageJ Suite	http://sites.imagej.net/Tboudier/
IJPB-plugins	http://sites.imagej.net/IJPB-plugins/

When all necessary plugins are checked, click on C*lose* and *Apply changes*.

Moreover, some SoilJ routines require the old BoneJ plugin (note that BoneJ2 will not work). You install this plugin by visiting bonej.org. Scroll down to "Install BoneJ" and download bonej_.jar into your ImageJ plugin folder.

Then, download or copy the SoilJ_-x.x.x-SNAPSHOT.jar file into your ImageJ plugin folder. You may obtain this file on https://github.com/johnkoestel/soilj.

**Eventually, restart ImageJ!**

Setting up Eclipse for SoilJ

Download and install Eclipse. Open Eclipse and close the welcome screen. Download and unzip the latest SoilJ source code into a folder named SoilJ_. Also download the old BoneJ_.jar plugin and save it at a location of your choice.

Set the workspace to the folder in which your SoilJ_ folder is located. For example, if your SoilJ_ folder is located in **E:Eclipse/MySoilJWorkspace/SoilJ_,** choose **E:Eclipse/MySoilJWorkspace** as your workspace location. Then import Soil as existing Maven Project.
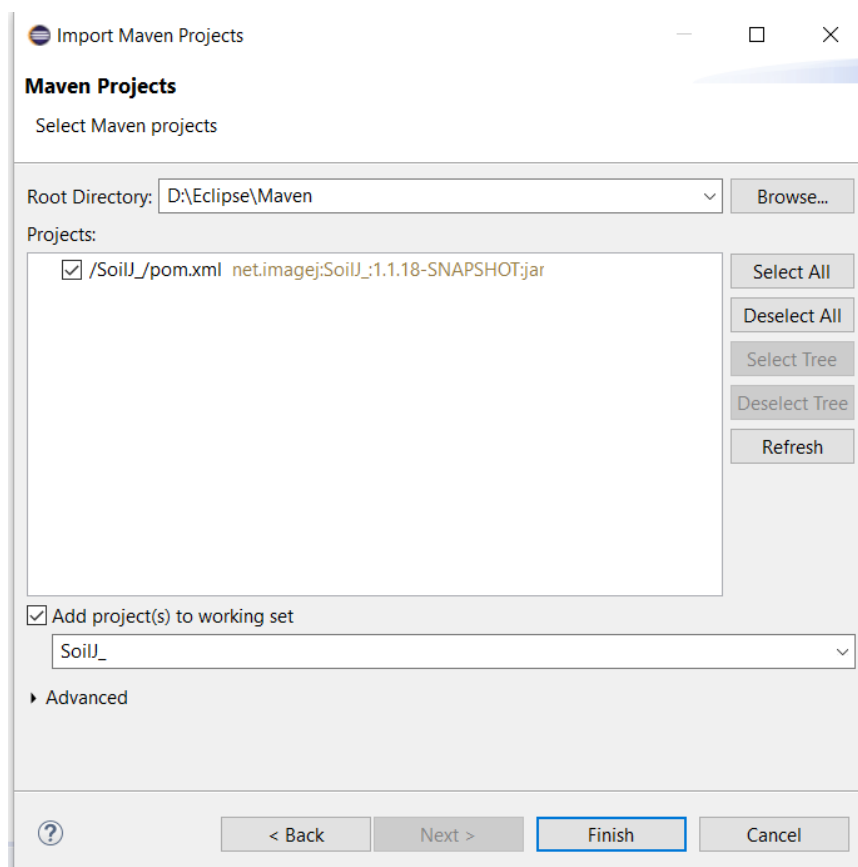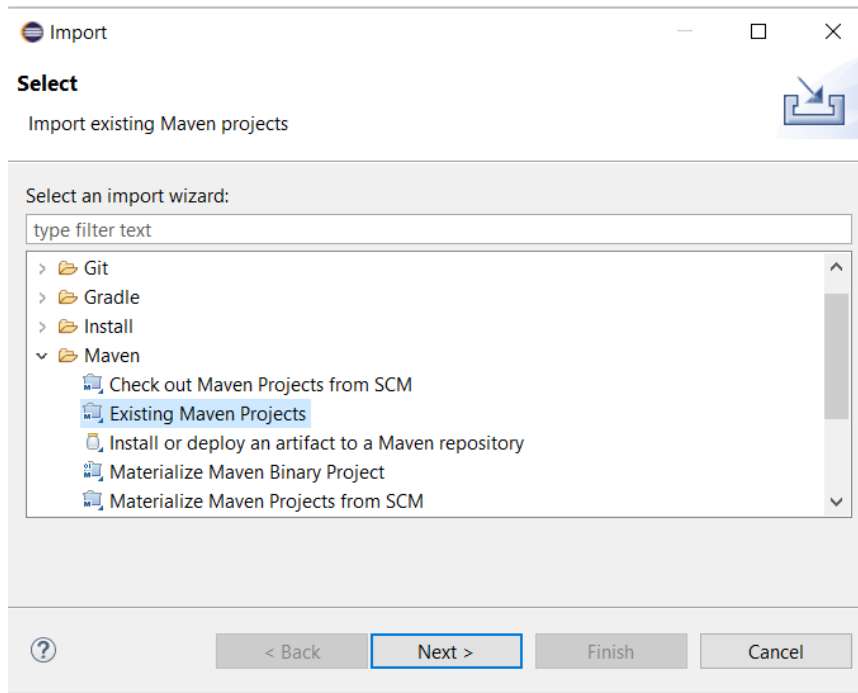
Note that the root directory will of course be different in your case. Important is that the POM.xml file is discovered and selected.

After all dependent files have been downloaded, add the old BoneJ_.jar to the **Java Built Path / Libraries**. You can get there by ***Project / Properties / Java Build Path / Libraries.*** Here, click on Add External JARs and select the BoneJ_.jar file that you have been downloading earlier.
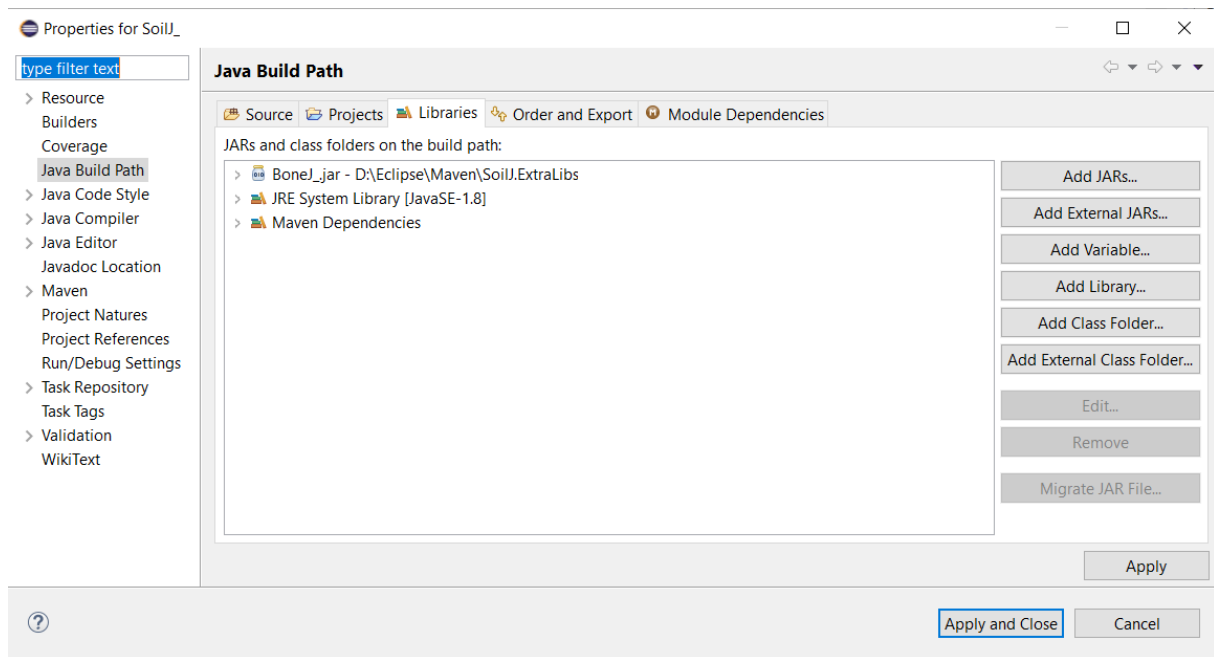
Then, make sure that SoilJ_/src/main/java and SoilJ_/src/main/java are included in the Java Build Path. You can check by *Project / Properties / Java Build Path*.
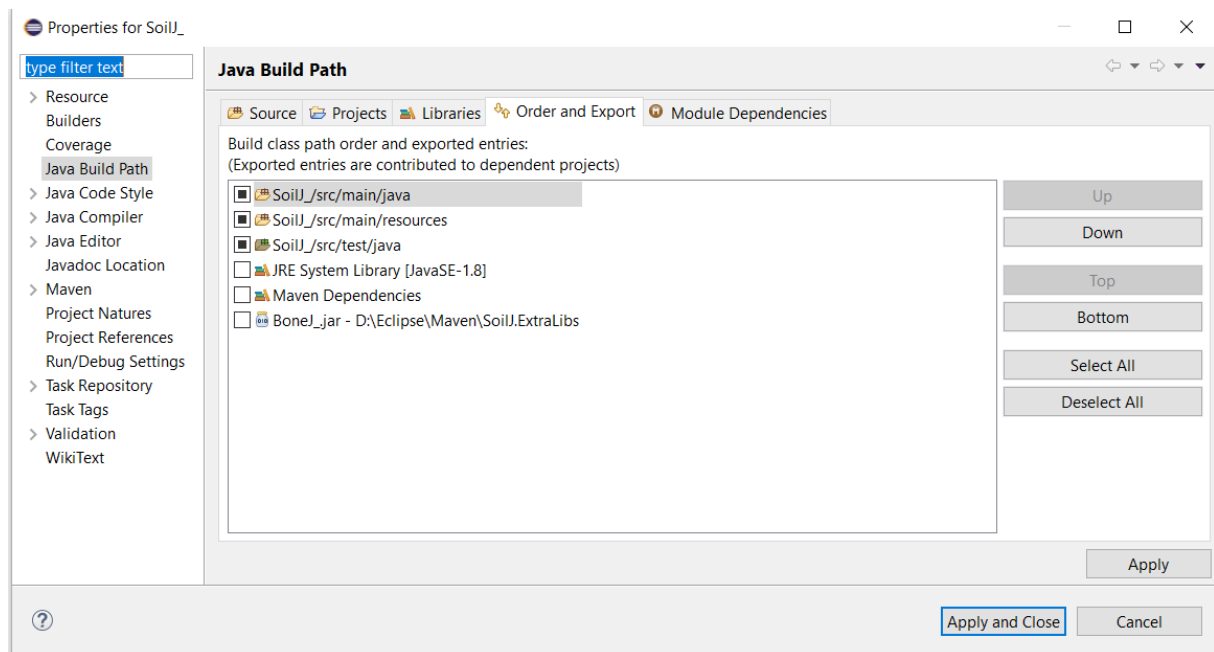
Then modify the location of your virtual plugin folder in SoilJ_.java to your workspace folder, e.g. E:Eclipse/MySoilJWorkspace.
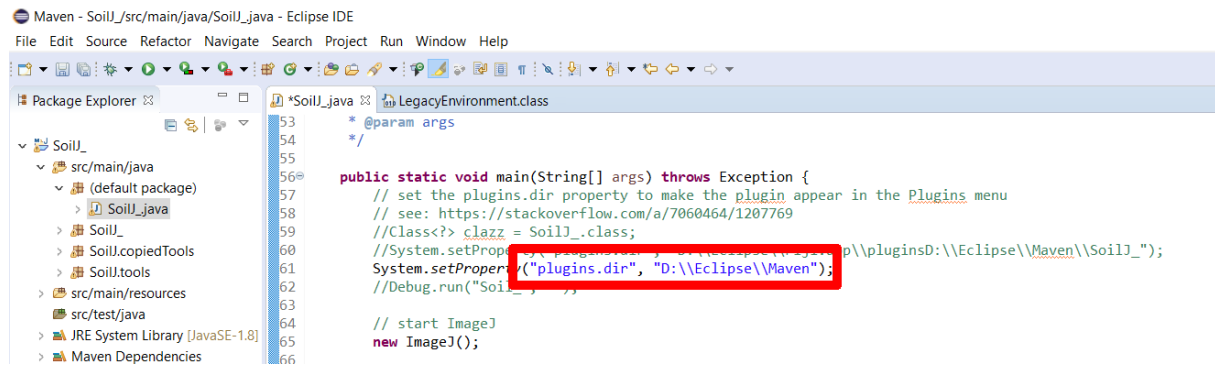
You can run SoilJ now inside ImageJ by right-clicking on the SoilJ_.jar file and selecting **Debug As / 1 Java Application**. SoilJ should now be visible in the plugin folder. If you set break points in the java source code in Eclipse, SoilJ will pause at them und you can inspect the respective variable values.

Eventually, set up your Maven to create the SoilJ_ jar files. Do this again by right-clicking on the SoilJ_.jar file, but this time select **Debug As / Debug Configurations…**. In the next menu double click on **Maven Build**. Choose the workspace directory as base directory as below.
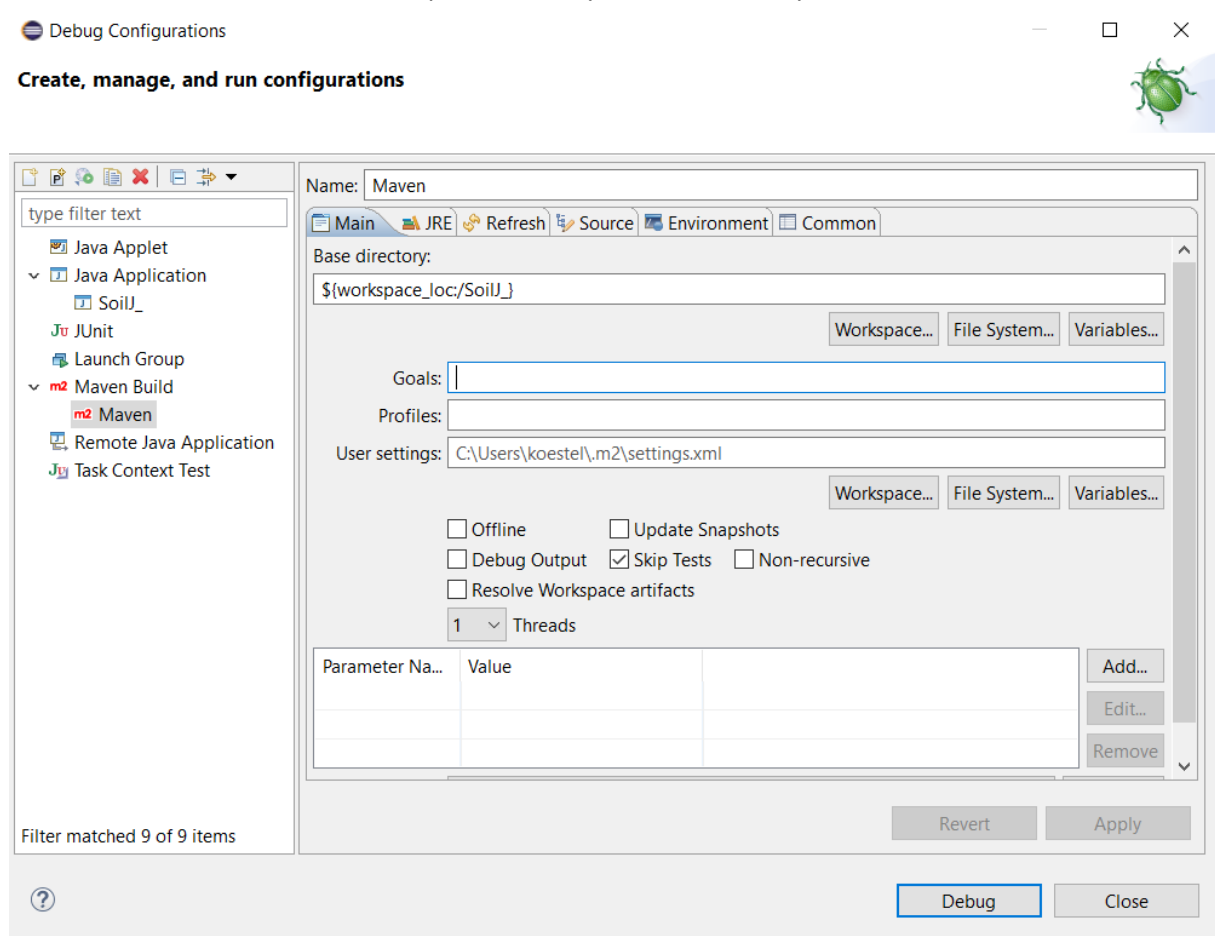
Modules

From version 1.2.0 onwards, you need to select TIFF files directly. You are then given the choice to either process only the selected image or to process all images in the folder of the selected image.

CombineTiffStack2Tiff
Required input: a folder containing one or several folders with 2-D TIFF image sequences.

Several individual single-layer TIFF-files are combined into one single multi-layer TIFF-file. The output file is saved under a newly created directory "3D". The program will crash in the selected TIFF files are multi-layer TIFF-files or if at least two TIFF-file deviate in image width or height.

A directory needs to be chosen that does not contain any TIFF-files itself but instead contains sub-directories with TIFF-files, the TIFF-files in each individual sub-directory will be merged to individual multi-layer TIFF-files. The names of the output TIFF-files are set to the names of the corresponding sub-directories.

The file or directory selection mechanism of all following modules is identical to the above described, with the exception of different output locations.

StraightenAndCenter
Required input: a folder containing one or several 3-D TIFF images.

The location of the outer hull of the sample column is searched in 50 horizontal image cross-sections. The 50 horizontal cross-sections are equidistantly distributed over the two central quartiles of the image's length along the Z-coordinate. Ellipses are fitted to the found location of the outer column perimeter in each cross-section. A straight line is fitted through the 50 ellipse centers from which the inclination and location of the column is deduced. Using this information, the column is rotated into an upright position and moved to the center of the image canvas. Unused parts of the canvas are removed from the image, apart from an at least 25 voxels thick fringe around the outer hull.

FindColumnOutlines
Required input: a folder containing one or several 16-bit 3-D TIFF images.

A plugin for automatically detecting column wall outlines of circular soil columns. The perimeter of the outer hull of the sample column is searched in 60 equidistantly spaced horizontal cross-section of the 3-D image. Ellipses are fitted to the found location of the outer column perimeter in each cross-section.

Various test indices are calculated upon which it is decided whether the column was found in the investigated cross-section or not. The topmost and bottommost cross-sections in which the column has been found are labelled as top and bottom surfaces of the sample column (not necessarily the surface of the soil contained in the column).

All image cross-section above and below the detected top and bottom of the sample column are removed from the 3-D image. The ellipse parameters for the cross-sections with missing column detection in between the top and bottom of the column are filled by linear interpolation using the ellipse parameters corresponding to the nearest cross-sections with detected column perimeters.

Optionally, the inner perimeter of the sampling column may be searched, i.e. the outer perimeter of the actual soil sample. Note that this option is only useful if a sufficiently large density contrast between sampling column and soil matrix exists. Alternatively, the inner perimeter of the sampling column may be calculated by subtracting the column's wall thickness from the outer perimeter. In this case, the wall thickness will be queried in the input mask.

The ellipses' parameters (center, major and minor radii, angle of major radius from y-direction, goodness of ellipse fit) are saved in a newly created sub-folder named 'InnerCircle'.

CalibrateGrayValues
Required input: a folder containing one or several 16-bit 3-D TIFF images and the location of the column outlines saved in the InnerCircle folder.

This module helps calibrating a series of 3-D images to one common gray-scale. Two reference gray values are selected that correspond to objects of known or at least constant density. Typically, one of the reference values corresponds to the gray value of the column walls.

The second gray value is chosen as a quantile of the histogram of the gray-values within the individual horizontal cross-sections, respectively. A very low quantile, e.g. 0.001, maybe be chosen to represent the least dense phase in the imaged object, e.g. air. It must be specified where the quantile is sampled: inside the inner wall perimeter, i.e. inside the soil, or outside the outer wall perimeter. For the latter case, two option are offered: sampling close to the outer wall perimeter (within one wall thickness) or distant to the outer wall perimeter (more than one wall thickness distance from the outer wall perimeter).

The gray values of the original image are then scaled according to the reference and target values. The images with the standardized gray values are saved in a folder named: 'Standard_<props>' where <*props*> stands for the standardization choices made in the plugin. If the lower reference values is chosen as the 0.001 quantile of the gray values inside the column and the upper reference value as the column wall, the output folder's name will be 'Standard_Quantile001InsideAndWall'.

ImageSegmentation
Required input: a folder containing one or several 16-bit 3-D TIFF images

The image is segmented into two phases: a denser and a less dense one. A constant, global threshold determined by e.g. using the SoilJ JointThresholdDetection module, can be applied to a dataset of calibrated images.

Alternatively, the images may be segmented into two phases using either one or two sequential global thresholding methods. As of March 2017, SoilJ does not incorporate local thresholding approaches. However, such approaches can be carried out independently from SoilJ by using other ImageJ plugins or third party software.

The use of InnerCircle files is optional.

ImageSegmentation provides the option to only save sample cross-sections to review the segmentation quality before 3-D binary files are saved.

SurfaceDetection

Required input: a folder containing one or several 16-bit 3-D TIFF images and the location of the column outlines saved in the InnerCircle folder.
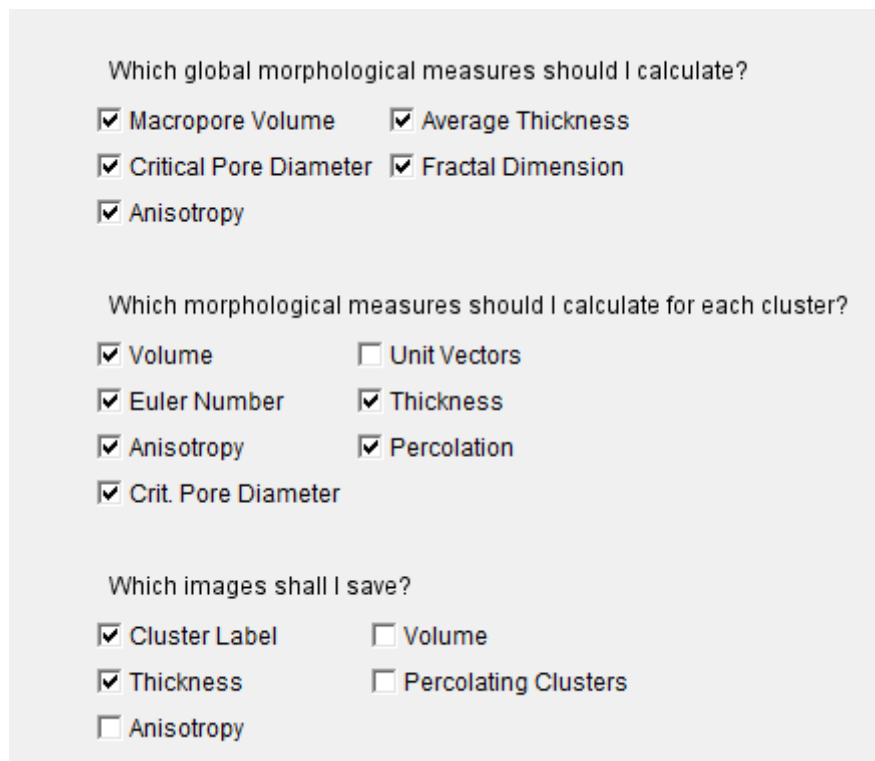
The surface topography at the top and the bottom of the column is detected and saved in a two-layer TIFF under the folder "SurfaceOfColumn".

PoreSpaceAnalyzer

Required input: a folder containing one or several 8-bit binary 3-D TIFF images (the gray values of the two phases being 0 and 255)

The morphological properties of the brighter phase of a binary image are analyzed. Most of the morphological measures are calculated by making use of plugins collected in the BoneJ bundle (Boube et al., 2010).

Additionally, options to calculate the percolation properties of the sample are available, including connection of a pore cluster to the top and/or bottom boundary of the soil column as well as the critical pore diameter.

You may choose to save up to 5 images of properties of the soil column images located in the selected folder. The morphological measures will be saved as ASCII files in a folder named *Stats*.

Optionally, the column outlines saved in the *InnerCircle* folder maybe be employed. Likewise, the surface topographies of the column saved in the *SurfaceOfColumn* may be taken advantage of.

BeamDeHardening (experimental)

Required input: a folder containing one or several 16-bit 3-D TIFF images and the location of the column outlines saved in the InnerCircle folder.

A plugin to remove beam-hardening artifacts from 3-D TIFF images of circular soil columns.

Work on a more user friendly interface and a technical documentation is in progress.

MedianFilterAndUnsharpMask3D
The 3-D median filter and a 3-D unsharp mask.

JointThresholdDetection
Required input: a folder containing one or several 16-bit 3-D TIFF

A tool that creates histograms of all TIFF images in the selected folder. The joint histogram is calculated and several of-the-shelf thresholding algorithms are applied to it.

The color codes in the output figure stand for:

color code	thresholding algorithm designation
RED	default (IJ_isodata)
BLUE	Otsu
GREEN	Huang
CYAN	maximum entropy
MAGENTA	minimum
ORANGE	minimum error
YELLOW	Renyi entropy
PINK	triangle
GRAY	isodata (normal version)

The numerical values of the thresholding results can be obtained from the table associated with the output figure. The values are listed in the same order as in the table above.

Optionally, the column outlines saved in the *InnerCircle* folder maybe be employed. Likewise, the surface topographies of the column saved in the *SurfaceOfColumn* may be taken advantage of.

ExtractPoreSizeDistribution
Required input: a folder containing one or several 32-bit 3-D thickness TIFFs

Extracts the pore size distribution from all thickness TIFFs located in the selected folder and saves the information in ASCII files in the selected folder.

ExtractPOMAndRoots
Required input: a folder containing one or several 16-bit 3-D TIFF

Extracts all regions within a chosen gray value range with sufficiently small gradients, i.e. partial volume voxels which typically are typically associated with large first derivatives in gray values are filtered out. The plugin is therefore suited to extract fresh organic matter, roots or water phases from the image.

PlotVerticalProfile
Required input: a folder containing one or several 16-bit 3-D TIFF

Calculates and plots statistics along the vertical axis of the column. This is the only plugin within SoilJ that requires the selection of an individual TIFF files instead of a folder containing several TIFFs.

GenerateRandomPoreClusters
Required input: none

Generates random pore networks by sequentially assigning random voxels to the pore-phase until a predefined porosity is reached.

SubScaleAnalyzer
Required input: a folder containing one or several 8-bit binary 3-D TIFF images (the gray values of the two phases being 0 and 255)

A tool for analyzing a series of sub-regions of interest for various morphological properties within binary images.

References

Doube, M., M.M. Klosowski, I. Arganda-Carreras, F.P. Cordelieres, R.P. Dougherty, J.S. Jackson, et al. 2010. BoneJ Free and extensible bone image analysis in ImageJ. Bone 47: 1076-1079. doi:10.1016/j.bone.2010.08.023.

Koestel, J. 2017 SoilJ: An ImageJ plugin for semi-automatized image-processing of 3-D X-ray images of soil columns. Sbmitted to Vadose Zone Journal.

Schindelin, J., I. Arganda-Carreras, E. Frise, V. Kaynig, M. Longair, T. Pietzsch, et al. 2012. Fiji: an open-source platform for biological-image analysis. Nature Methods 9: 676-682. doi:10.1038/nmeth.2019.

Schneider, C.A., W.S. Rasband and K.W. Eliceiri. 2012. NIH Image to ImageJ: 25 years of image analysis. Nature Methods 9: 671-675. doi:10.1038/nmeth.2089.