

User's manual for JColloids.

P. Domínguez-García, M. Pancorbo, Francisco Ortega, M. A. Rubio

1. MOTIVATION

JChainsAnalyser is a Java-based software for the analysis of images of magneto-rheological fluids (MRF). A MRF consists on a suspension of micro-nanomagnetic particles on water or another organic liquid. When a magnetic field is applied to the suspension, the particles aggregate forming chains or clusters. This new version of **JChainsAnalyser** is named **JColloids** because it can be used not only with MRF but also with any kind of set of images of colloidal objects in a two-dimensional configuration. The basic usage and algorithms of this new version are the same as the older ones, and the differences are explained in the published manuscript.

This application uses as an external library the code of open-source software for image analysis called *ImageJ* (<http://rsb.info.nih.gov/ij/>). The main objectives of **JColloids** as an independent extension of *ImageJ* and as an update of **JChainsAnalyser** are the following:

1. To simplify the previous **JChainsAnalyser** by focusing on the detection of the colloidal objects using the minimum number of input parameters, and simplifying the code and the internal calculations.
2. To keep the main advantages of the previous version, such as the methodology for automatically analyzing an unlimited number of images, the algorithms used for the binarization of the images, and the functionalities of the application to be run on a desktop or remote computer for any operating system.
3. To update the external libraries used by the software, included *ImageJ*.

2. INSTRUCTIONS AND EXECUTION OF THE SOFTWARE

2.1 Software requirements

JColloids is a Java-based software, thus a JRE (Java Runtime Environment) for running this program has to be installed. You can download the latest version of the JRE here: <http://www.java.com/en/download>.

JColloids reads images stored on the following format:

time in miliseconds + term +.jpg.

For example, a working name for an input file is *25558800original.jpg*, where *25558800* is the time given on miliseconds, *original* is the term and *jpg* the image file format (*tiff* file formats can be also used). In the test directory, you can see some examples of experimental images, which can be used to test the application. Before running the application, please read the following:

JColloids can be used two-fold: i) Executing the graphical user interface. ii) Running the program directly using the command line.

The next subsection explains how to run the software and how to modify the input parameters.

2.2 Installation of the software

Unzip **JColloidsSoftware.zip** archive to the destination path. **JColloids** is a Java-based software and a JRE (Java Runtime Environment) for running this program has to be installed. You can download the last version of the JRE here: <http://www.java.com/en/download>. The files needed for the execution are contained in a directory named *JColloids*.

2.3 Executing the software

Here, we briefly describe the JColloids GUI and how to use the program. First, we need to run the GUI using the following command:

```
java -jar -Xms512m -Xmx512m -XX:-UseGCOverheadLimit JColloids.jar
```

This command line is provided inside the executable files *JColloids.bat* and *JColloids.sh* for Windows and Linux respectively. When any of these two files are executed or the command is run in a terminal, the following graphical user interface (GUI) appears:

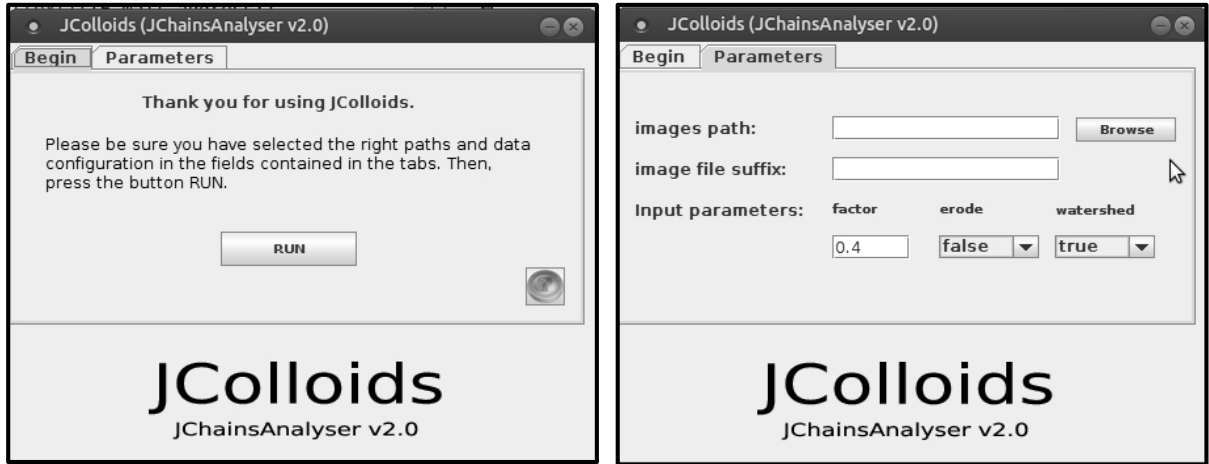


Figure 1. JColloids' GUI. The two available tabs are shown. The left one is for running the software, the right contains the directory paths for input files and the input parameters.

3. RUNNING THE APPLICATION

3.1 Using the GUI

First, run the executable files, *JColloids.bat* or *JColloids.sh* for Windows and Linux respectively.

Then, the GUI appears as shown in figure 1. Note that, in the images of fig. 1, there are two different tabs on the top part of the window. In the *Begin* tab (fig. 1 left), we can run the application by clicking the *RUN* button. By clicking on the help button on the right, this pdf will be shown. Before running this software, we must enter the paths of the input files and the required parameters. These input data will be stored and read from a **xml file**, called *input.xml*, which should be located in the installation folder, the same as the executable files and the JColloids.jar file.

By clicking on the *Parameters* tab, we will see an image like the one shown in fig. 1 on the right. The parameters are the following ones:

1. **images path:** Here, we have to enter the directory that contains the original images. There is no need to enter additional directory paths because the directories where the analysed images and the result files (output data) are saved will be automatically created. To change the directory path, we can enter the new one in the text field or click on the *Browse* button and manually select the directory. If the path directory is not selected or directly written in the xml file, an error message will appear after the execution.

2. **images file suffix:** The original images must be named in the following form: time in milliseconds + term + .jpg. For example, a correct name should be *25558800original.jpg*. Therefore, in this case, the suffix that must be written in the text field is *original*. If there is any text after the time string, this field should be empty. JColloids reads automatically the time values on the image file names, subtracts the minimum value, and then obtains the temporal value of each image. This is stored and saved on the *data.txt* output file.
3. Parameters for the filtering:
 - **factor:** Parameter for the segmentation of the images. It is usually fixed to 0.4. This is the only number to be modified if the filtering of the images does not provide the expected results.
 - **erode:** Boolean value (true or false). Performs a slightly eroding of the object profile. It can be useful for deleting small objects or to fix defects in the contour profiles.
 - **watershed:** Boolean value (true or false). Separates objects detected as only one. It can be useful to separate very close particles detected as one by the image analysis.

Note that, when a field is changed, it is highlighted on the text field (yellow color). Then, you need to press *ENTER* on the keyboard to confirm the change and save the selection in the xml file. The selection will be read and shown in the text field the next time the GUI will be executed.

Once we have introduced all the parameters into the xml file using the GUI or by writing directly into it, we can return to the *Begin* tab and press the *RUN* button. A window will appear containing a console with text messages. This console shows the messages related to the activity of the program execution. The two buttons at the bottom of the window can be used to clear the console messages (*Clear Console*) and to stop all the JColloids execution (*STOP ALL*).

When execution is over, we will find two new directories inside the selected path directory of the original images, named *analysed* and *statistics*. The *analysed* directory contains the binary analysed images, saved in gif image file format. The *statistics* directory contains a *data.txt* file which contains all the relevant data of the detected objects, and a *report.txt* file with a report of the process. If we run JColloids again, without changing the input path, all output files will be rewritten.

3.2 Using the command line

JColloids can be used without the GUI, so the programs can be run employing a remote computer without desktop access. In order to do that, we must include in the same directory all the jar files contained in the *jars* directory in the source code directory. The input xml file should be also in the directory (see *command* directory inside *JColloids* one). To change the input parameters, this xml file needs to be directly modified using an text editor. The command to be used in this case is the following one (in only one line):

```
java -Xms512m -Xmx512m -XX:-UseGCOverheadLimit -cp ij.jar:ij-ImageIO_.jar:
jdom-2.0.6.jar:l2fprod-common-directorychooser-6.9.1.jar:l2fprod-common-shared-6.9.1.jar:
Otsu_Thresholding.jar:JColloids.jar mains.PrincipalCommand
```

The *command* directory inside *JColloids* provides all the files needed for this kind of execution. Inside a terminal, simply run the file *JColloids-command.sh*.

4. FURTHER INFORMATION

In the following section, we would like to explain some details of JColloids about the libraries and code used.

4.1 Package structure

For more details about the structure of the source code, you can consult the documentation contained in the *doc* directory. This is a documentation automatically generated by Java using the command *javadoc*. There, you can see comments about the classes, objects and methods used. Basically, JColloids is composed by the following packages:

- **gui**: Contains the classes and objects for the GUI and the Console.
- **images**: A directory with the images used in the GUI.
- **interfaces**: The interfaces are designed to contain data, names and messages that are shared by the different objects in the code. This package also contains the objects related with the xml file which contains the input parameters.
- **jars**: Contains the external jars used by the code
- **main**s: This package contains the objects for running the main applications. The *Principal* object runs the GUI application, and the rest of the package elements are used for running the rest of the programs. These programs can be run in two ways. If we use the GUI, they are *Thread* objects, whereas if we use a command line execution, they are run using a classical static main method.
- **manipulations**: It contains the objects and the methods to manipulate images, making the image analysis, obtaining the output data and storing the results.
- **tracking**: The objects inside this package contain the classes for representing the properties of the colloids and the *Analyzer* for extracting the data from the filtered images.

4.2 External libraries

JColloids uses the methods and the code of *ImageJ* (v1.5.0) as a library, but we have also used other external libraries, updated from to the ones used in the original code of JChainsAnalyser. These additional external libraries are the following ones:

- **ij-imageIO** (v1.2.4): <http://ij-plugins.sourceforge.net/plugins/imageio/index.html> This is a plugin for *ImageJ*, which adds additional functionalities to image files formats.
- **Jdom2** (v2.0.6): <http://www.jdom.org/>. A very easy API to write/read xml files from Java.
- **L2FProd** (v.6.9.1): Used in JColloids for choosing path directories. They are two jar files which can be obtained from these repositories:
<https://mvnrepository.com/artifact/com.l2fprod.common/l2fprod-common-directorychooser/6.9.1> and <https://mvnrepository.com/artifact/com.l2fprod.common/l2fprod-common-shared/6.9.1>

Note that some of the classes contained in *ImageJ* have been adapted for JColloids to hide graphical elements or to change the visibility of objects and methods. For example, we use *BinaryFiller* in its version from *ImageJ* v1.40. We also use also a plugin for the Otsu thresholding algorithm for *ImageJ* as an external library (see <https://imagej.nih.gov/ij/plugins/otsu-thresholding.html>).