jClustering v1.2.4 user manual

José María Mateos
jmmateos@mce.hggm.es

August 6, 2013

Contents

1 Introduction

jClustering is an ImageJ plugin for dynamic (3D + time or 2D + time) segmentation. It groups image voxels by similarity of their temporal behavior. This is a brief user manual that explains how to use jClustering within a working ImageJ installation.

For further information regarding jClustering, please refer to *jClustering, an open framework for the development of 4D clustering algorithms*, PLOS ONE, 2013.

This manual refers to jClustering v1.2.4 and beyond. It will not be edited just to update this version number. Unless there are significative changes in a new version, do not expect this number to match the latest release.

2 Download and installation

The latest version of jClustering can be downloaded from the *Releases* section of the main github repository for this project at `https://github.com/HGGM-LIM/jclustering/releases`.

The installation procedure is straightforward, you just need to follow these steps:

1. Download the latest jClustering release from `https://github.com/HGGM-LIM/jclustering/releases` and copy the `jClustering_.jar` file into the `plugins/` directory of your ImageJ installation.
2. Download the latest Apache Commons Math library from `http://commons.apache.org/math/download_math.cgi` (it should be a file with a name in the form of `commons-math3-3.X-bin.zip`, where X is the most recent version number, open it with any unzip program and copy the file `commons-math3-3.X/commons-math3-3.X.jar` to the `plugins/jars` directory of your ImageJ installation.
3. Same for the fastICA library. Go to its webpage (`http://sourceforge.net/projects/fastica/`) and store the .jar file in the `plugins/jars` directory of your ImageJ installation.

jClustering has been tested on ImageJ versions newer than 1.46r (included) and JRE 6.

3 Main window

When you run jClustering (Plugins > Clustering > jClustering) with an ImageJ hyperstack open, you will get the main window (figure 1).

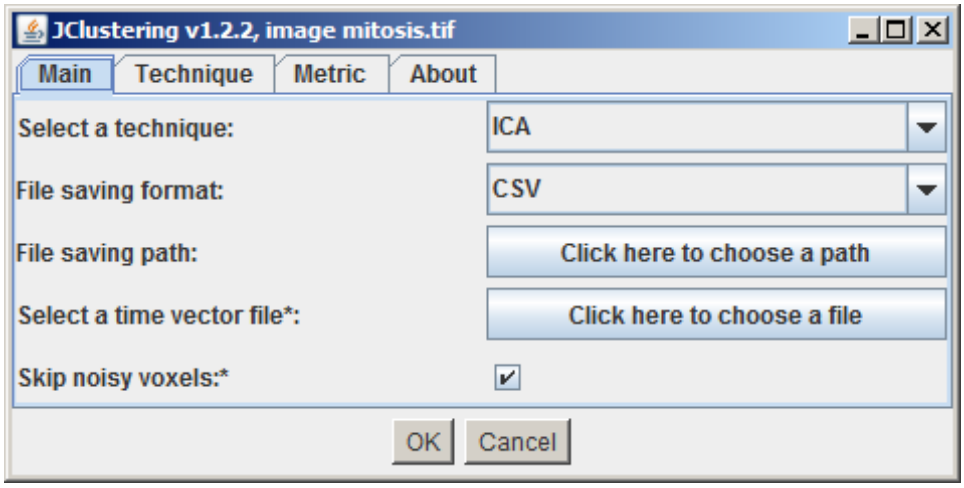


Figure 1: Main jClustering window.

If a hyperstack is not present, an error dialog will pop-up. jClustering works on images that have already been opened and displays the name of the image it is acting upon on the main window title bar. If you wish to run clustering operations on another image, select it and run another instance of jClustering.

The different options are:

- *Select a technique:* select a clustering technique. Please refer for the next sections for information on each one of them.
- *File saving format:* choose the format in which the text files containing the time-activity curves for each cluster will be stored. Current values are CSV (comma-separated values), PMOD (for the PMOD software, `http://www.pmod.com`) and tab-separated.
- *File saving path:* if you specify a path here, the text file will be saved in that path. The name of the text file begins with `jclustering` and includes a timestamp of the time of the analysis.
- *Select a time vector file:* ImageJ does not read different frame length information from the image header. If you want the frame start and end times to appear in your results text file, you need to specify it here. The format of this file is a space or tab-separated file with two columns: frame start and frame end time. For instance:

```
0.0 2.0
2.0 4.0
4.0 6.0
...
```

- *Skip noisy voxels:* This is an experimental function that skips voxels with a time-activity curve that can be considered noise. As the *ImagePlusHypIterator* object now does not return voxels that have been masked, this functionality is deprecated and unchecked by default but has not yet been removed

4 Implemented clustering techniques

The options for the different clustering techniques implemented in jClustering are shown below.

4.1 Independent Component Analysis (ICA)

The ICA technique tab is shown in figure 2.

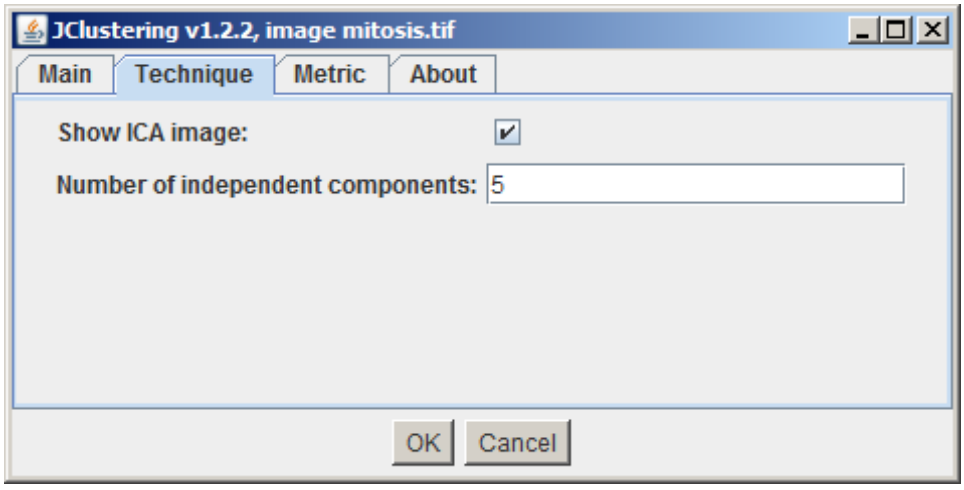


Figure 2: Independent Component Analysis window.

The different options are:

- *Show ICA image:* as jClustering shows the final clustering result based on a *winner-takes-it-all* approach, it may be useful to show on screen the resulting ICA analysis. This image will show up if this option is checked (it is by default).
- *Number of independent components:* the estimated number of independent components. Defaults to 5.

This clustering technique also outputs the independent components found on the output directory, if this has been chosen.

4.2 K-means

The k-means technique tab is shown in figure 3.

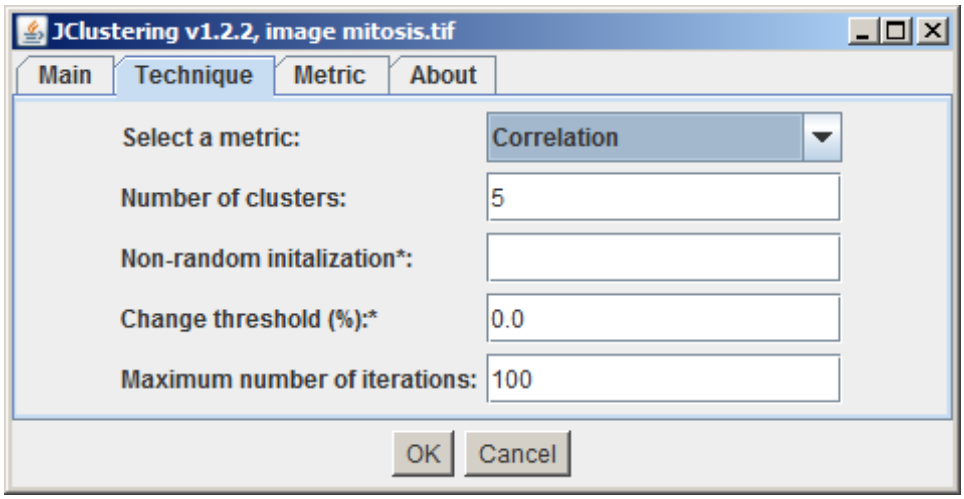


Figure 3: K-means window.

The different options are:

- *Select a metric:* selects one of the implemented metrics. Please refer to section 5 for more information.
- *Number of clusters:* the initial number of clusters to create. Defaults to 5. The final result may yield fewer clusters, as some might be emptied during the iterative process.
- *Non-random initialization:* if you wish to select all or some initial centroids you can write the coordinates here with the following format:

`x1,y1,z1;x2,y2,z2`, where x, y and z are the 3D coordinates of the centroid. If you define fewer points than the configured number of clusters, the rest will be randomly chosen. If you write `++` here, a k-means++ initialization algorithm will be used. Use `det++` for a *deterministic* k-means initialization, in which the first centroid corresponds to the time-activity curve with the highest maximum amplitude.

- *Change threshold (%):* the amount of change allowed between clusters to end the iterative process. By default, all clusters must remain the same between two given iterations to stop.
- *Maximum number of iterations:* the maximum number of iterations allowed. If no convergence is achieved and this number of iterations is reached, the program finishes with the clusters currently in memory.

4.3 Leader-follower

The leader-follower technique tab is shown in figure 4.

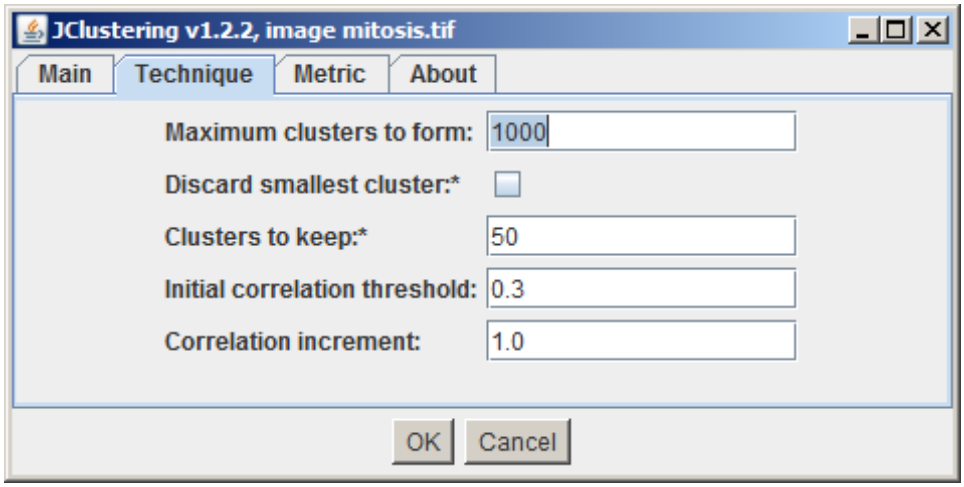


Figure 4: Leader-follower window.

The different options are:

- *Maximum clusters to form:* the maximum number of clusters the algorithm will create.
- *Discard smallest cluster:* if the maximum number of clusters is reached

and a new one needs to be created, the smallest one (the one with fewer voxels and lower mean amplitude) is discarded and a new one is created when this box is checked. When unchecked, voxels that need to create a new cluster are discarded.

- *Clusters to keep:* the number of clusters that will be shown on screen when the algorithm finishes. As it normally creates a great number of clusters, it is better to keep this number low. Clusters are shown on screen ordered by the number of voxels they contain, with the biggest ones first.
- *Initial correlation threshold:* the initial correlation threshold use to create new clusters.
- *Correlation increment:* a multiplicative variable that increments the correlation threshold of a given cluster. When set to a value greater than 1.0, the correlation threshold for a given cluster is incremented every time a new voxel is added. No increment is set by default.

4.4 Principal Component Analysis (PCA)

The PCA technique tab is shown in figure 5.

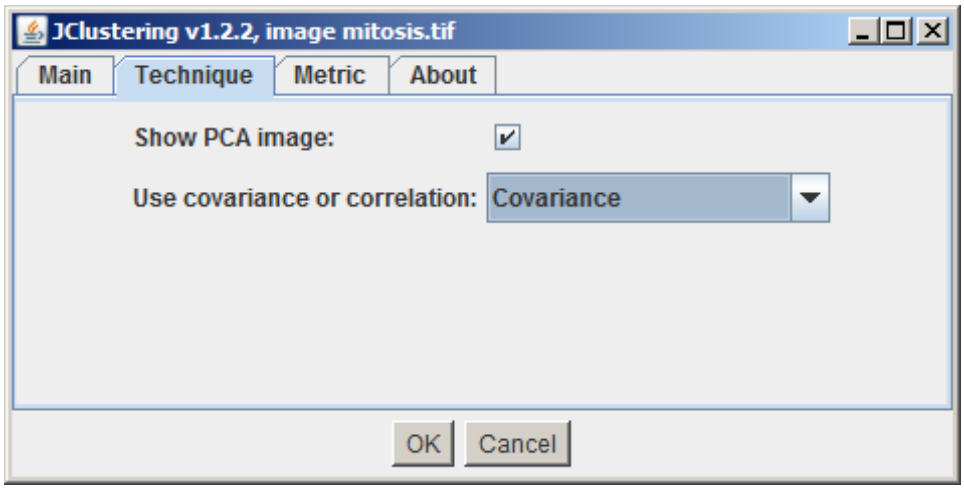


Figure 5: PCA window.

The different options are:

- *Show PCA image:* same as the ICA case. If you wish to visualize the final PCA result, check this box.
- *Use covariance or correlation:* PCA is computed by applying SVD to the covariance matrix. In some cases, the correlation matrix may also be used. This dialog allows you to select either one. Defaults to covariance.

This clustering technique also writes to file the values of the principal components.

4.5 Singular Value Decomposition (SVD)

The SVD technique tab is shown in figure 6.

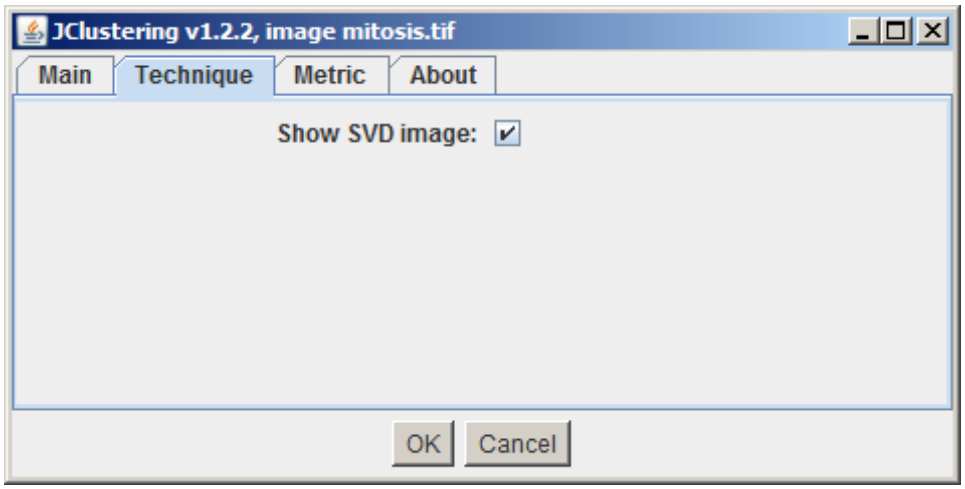


Figure 6: SVD window.

The only option is to show the SVD image, as in the ICA and PCA cases. SVD computation is done by applying SVD to the whole image matrix, not to the correlation or covariance one (PCA).

This clustering technique also writes to file the values of the eigenvectors found.

5 Implemented metrics (distances)

The different metrics are currently used only the the k-means clustering technique. For more information on the particular metrics implemented, please refer to the original jClustering paper on PLOS ONE.

6 Results presentation and interpretation

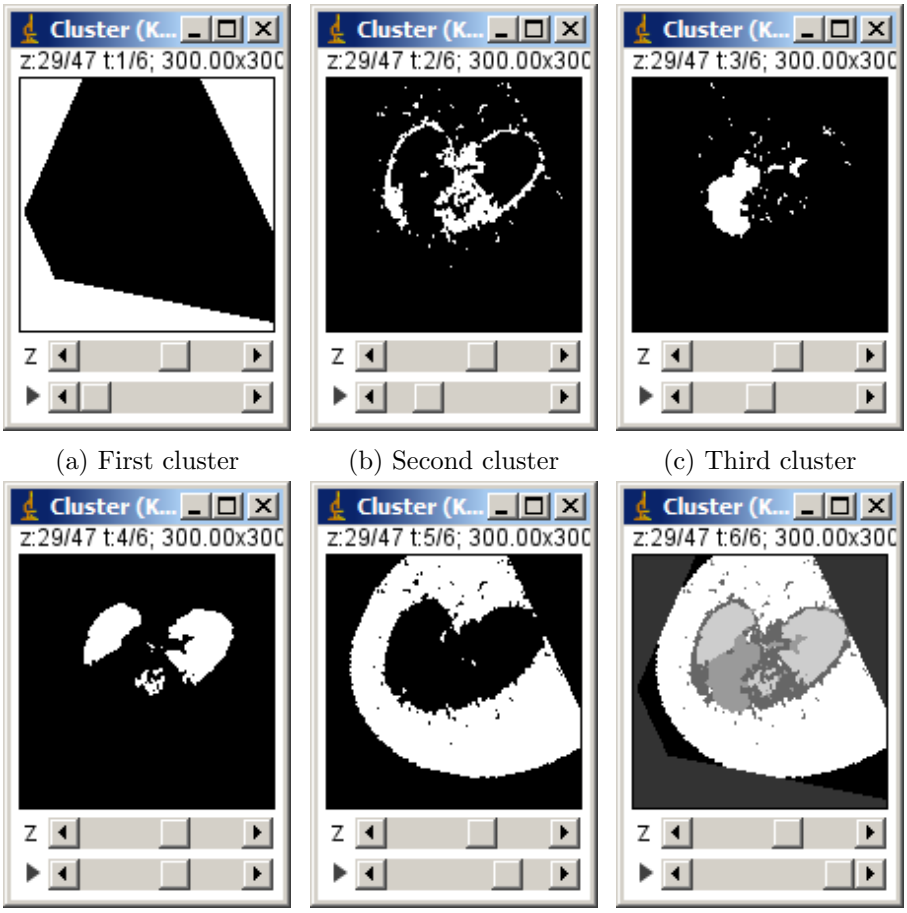
This section details how jClustering shows the results on screen and how they should be interpreted. The reader will also find here details on how the text files are stored.

6.1 Graphical display

jClustering shows the resulting segmentation clusters in a new 4D image. This image has $n + 1$ frames, with n being the total number of clusters formed. Each frame of this image represents each individual cluster, with the voxels belonging to it with a non-zero value. The last frame is useful for visualizing all clusters at the same time; in this representation, each one is displayed with a different grayscale value.

In figure 7 the results from a 5 clusters k-means are shown.

It is important to note that this convention is used then *deterministic* clusters are used. Some techniques, such as PCA, also have the possibility of displaying the computed principal components. These techniques (currently: ICA, PCA and SVD) generate their clusters by using a winner-takes-it-all approach, in which the independent component / principal component or singular value with the greatest value is used to assign each voxel. In any case, the mathematical representation of the images may be useful if the user wishes to do some more elaborated processing. For instance, when the PCA cluster is set to show the PCA image, the same approach is used: a HyperStack is returned (in addition to the original clusters) and each principal component is shown in a different frame.



(a) First cluster (b) Second cluster (c) Third cluster

(d) Fourth cluster (e) Fifth cluster (f) All clusters at once

Figure 7: Example of final clustering displaying by jClustering using ImageJ's HyperStacks

6.2 Text files

More important than the visual representation of the formed clusters are the values of the time-activity curves of the different clusters. These values will be stored to a file if the *File saving path* option has been set in the main tab; the TAC for each cluster will be stored as a column in that file, every frame is a row.

The format for this file must be chosen by the user (through the *File saving format* option and also depends on whether a time file vector with the frame starting and end times has been specified. If this has been the case, the first two columns are taken from that vector file. In the case of the PMOD format, as the frame information is mandatory, if no time vector file has been provided, hard-coded frame information (each frame lasting 1 second) will be used. That is hardly optimal, but that is what that specific format requires.