jClustering User Manual

José María Mateos
`jmmateos@mce.hggm.es`

June 27, 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.

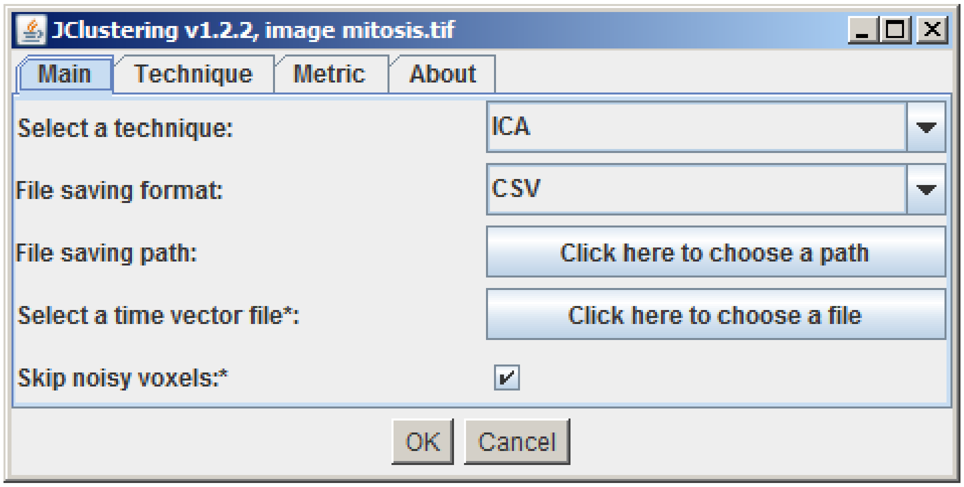
For further information regarding jClustering, please refer to *jClustering, an open framework for the development of 4D clustering algorithms*, PLOS ONE, 2013.

2 Installation

The installation procedure is detailed at `https://github.com/HGGM-LIM/jclustering#installation`. In any case, it is worth emphasizing here that **you need to rename the downloaded jClustering JAR file to jClustering_.jar, or you will get an error message**. This is due to the way the class autodetection code works.

3 Main window

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



JClustering v1.2.2, image mitosis.tif

Select a technique: ICA

File saving format: CSV

File saving path: Click here to choose a path

Select a time vector file*: Click here to choose a file

Skip noisy voxels:* ☑

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. The very simple implementation can be seen in the `isNoise` method in `https://github.com/HGGM-LIM/jclustering/blob/master/src/main/java/jclustering/ImagePlusHyp.java`. In the tested examples it removes most of the useless voxels and allows the algorithms to run much faster.

4 Implemented clustering techniques

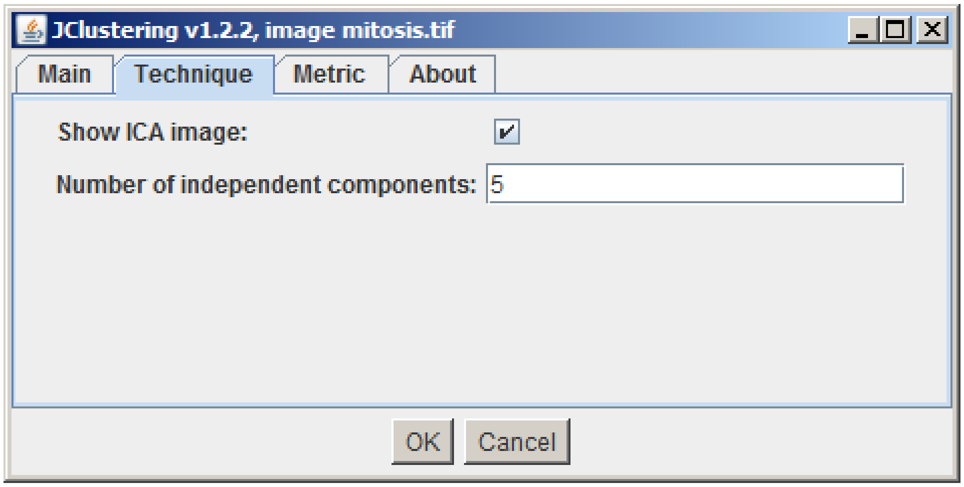
The options for the different clustering techniques implemented in jClustering are shown below.

4.1 Independent Component Analysis (ICA)

The ICA technique tab is shown in figure 2.

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



Show ICA image: ☑

Number of independent components: 5

Figure 2: Independent Component Analysis window.

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.

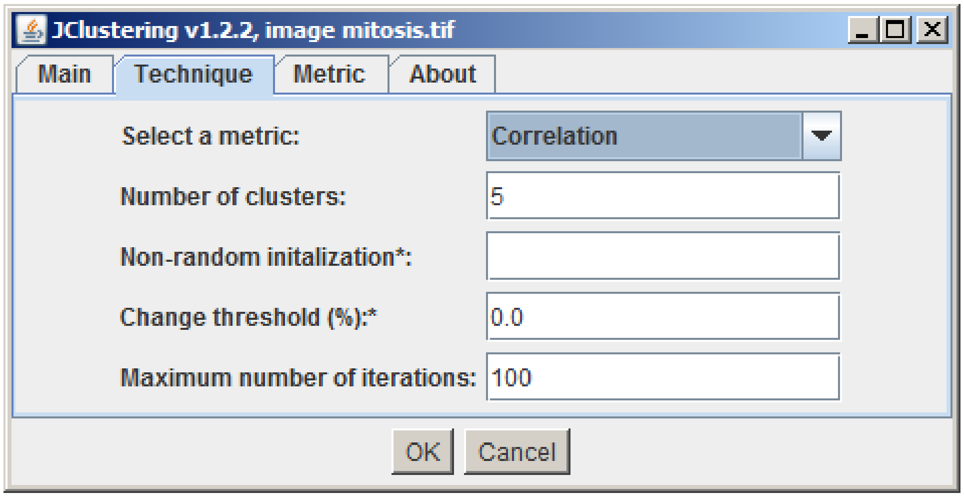
This clustering technique also outputs the independent components found on the output directory, if this has been chosen.

4.2 K-means

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

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 less clusters if one of them is 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



Select a metric: Correlation

Number of clusters: 5

Non-random initalization*:

Change threshold (%):* 0.0

Maximum number of iterations: 100

Figure 3: K-means window.

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.

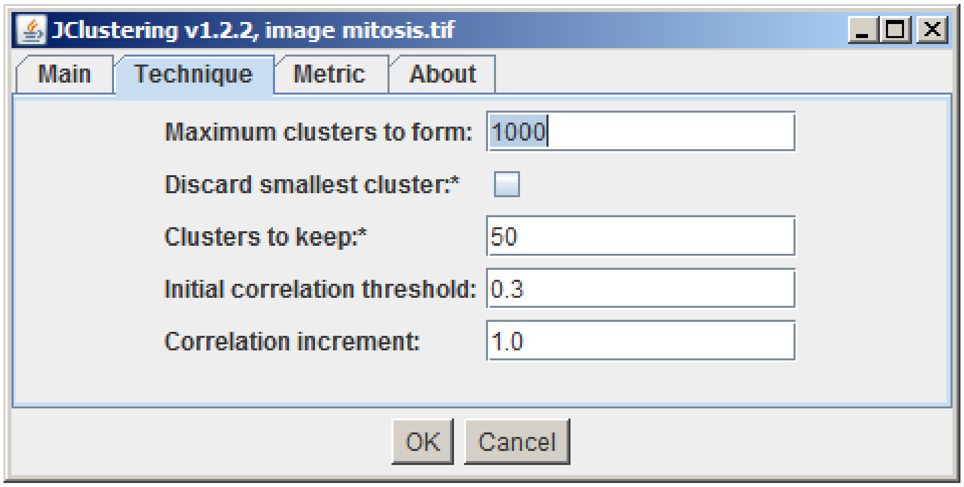
- *Change threshold (%):* the amount of change allowed between clusters to end the iterative process. By default, all clusters must remain the same between two given iterations to stop.
- *Maximum number of iterations:* the maximum number of iterations allowed. If no convergence is achieved and this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

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



Maximum clusters to form: 1000

Discard smallest cluster:* □

Clusters to keep:* 50

Initial correlation threshold: 0.3

Correlation increment: 1.0

Figure 4: Leader-follower window.

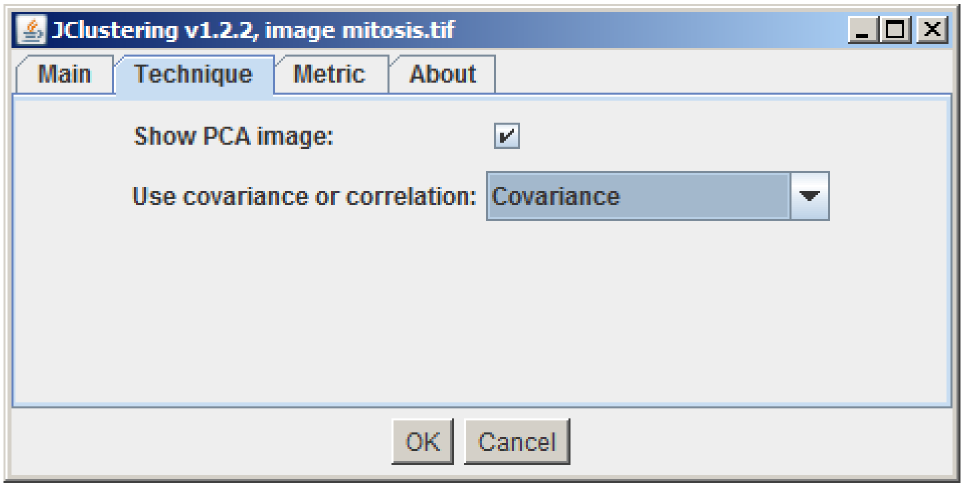
fewer voxels and lower mean amplitude) is discarded and a new one is created when this box is checked. When unchecked, voxels that need to create a new cluster are discarded.

- *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.

The different options are:



Show PCA image: ☑

Use covariance or correlation: Covariance

Figure 5: PCA window.

- *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.

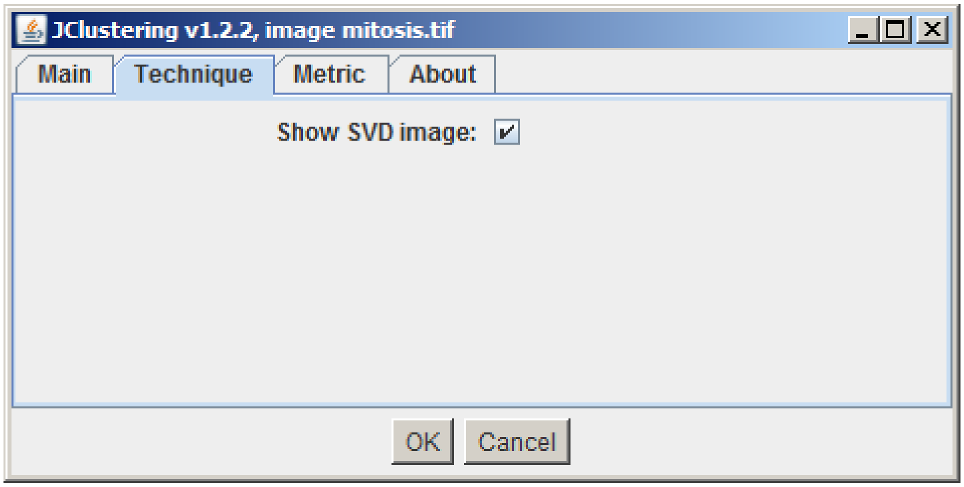
This clustering technique also writes to file the values of the principal components.

4.5 Singular Value Decomposition (SVD)

The SVD technique tab is shown in figure 6.

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.



JClustering v1.2.2, image mitosis.tif

Show SVD image: ☑

Figure 6: SVD window.

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.