

TANGO: a customizable tool for high-throughput image analysis of nuclear signals

Jean Ollion^a, Julien Cochenec^a, François Loll^a, Christophe Escudé^a and Thomas Boudier^b

^aCNRS UMR 7196, Muséum National d'Histoire Naturelle, Paris, France.

^bUniversité Pierre et Marie Curie, IFR83, Paris, France.

ABSTRACT

The study of nuclear architecture has become a very important field of cellular biology. When studied with an imaging approach, a limiting step is the automatic processing and quantification of the images. Here we present TANGO (Tools for Analysis of Nuclear Genome Organization), an ImageJ plug-in for high-throughput quantitative analysis of 3D fluorescence microscopy images, targeted towards analysis of nuclear architecture. Several efficient programs for processing and quantification of 3D fluorescence signal already exist in the ImageJ community, but those programs are often difficult to use on a large number of images without programming competences. TANGO takes advantage of this rich community by allowing the user to easily parametrize, assemble and run those programs on large sets of images. Thanks to this modularity, TANGO offers a very large spectrum of applications. It also provides functionalities to manage large datasets of images, visualize them in 3D and rapidly assess the quality of processing. Additionally, it can be used for automatic analysis in fields other than nuclear architecture. TANGO is free, open-source and available at <http://www.mnhn.fr/usm503/tango>.

Keywords: ImageJ, 3D, plug-in, High-Throughput, Nuclear Architecture

1. INTRODUCTION

The nucleus is a highly structured and organized cellular organelle. It is composed of various distinct compartments, defined by the presence of specific DNA sequences and/or the concentration of a given protein. The best known is the nucleolus, but there are many others such as PML bodies, chromocenters (pericentric heterochromatin foci, which display as brighter spots within the DAPI nuclear signal), etc. The spatial localization of these structures within the nucleus space has been linked with major cell functions such as regulation of gene expression. Several techniques exist to study their location, for example Immunocytochemistry and FISH (Fluorescence *In Situ* Hybridization). They allow selective fluorescent labelling of multiple compartments within the nucleus, with few disruptions of spatial information. These experiments can provide much quantitative information on these structures, at the single-cell level. Though recent microscopes facilitate the acquisition of large sets of images, one limiting step is the automatic processing and quantification of images. Furthermore, different labelling experiments can mark different structures, each of them potentially needing different processing settings. Quantification also depends on the experiment and the biological question. In this context, TANGO (Tools for Analysis of Nuclear Genome Organization) was developed as an ImageJ (see appendix B1) plug-in for high-throughput quantitative analysis of 3D (three-dimensional) fluorescence microscopy images. It can be adapted to various imaging experiments and provide data to answer different biological interrogations. Its genericity relies on two main features:

- Flexibility: a user can easily parametrize the analysis to match specific labelling experiment architecture
- Modularity: processing and quantification of labelled structures is carried-out using plug-ins, thus they are not limited to existing programs and can be extended for specific needs.

In order to gain statistical significance, studies should integrate data on a large population of cells. For this reason, TANGO has been designed to automatically process substantial sets of images, with the possibility of a rapid manual intervention and visualization at several steps in the processing.

Further author information: (Send correspondence to T. Boudier)

T. Boudier: E-mail: thomas.boudier@snv.jussieu.fr

J. Ollion: E-mail: ollion@mnhn.fr

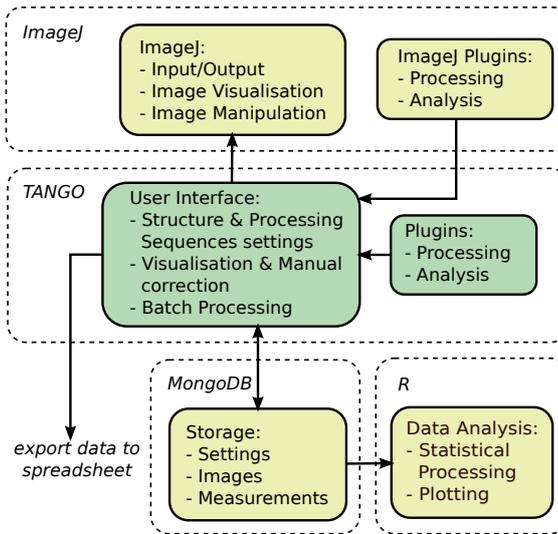


Figure 1. TANGO's architecture : ImageJ is used for image management and its plug-ins; MongoDB is used for storage; R is used for data analysis.

2. MATERIALS AND METHODS

2.1 Implementation

TANGO has been implemented as a java plug-in for ImageJ (see Figure 1), therefore benefiting from ImageJ functionalities such as opening of images, visualization and standard manipulation (i.e. brightness and contrast adjustment), or ROI (Region Of Interest) editing. This plug-in also interacts with other plug-ins like Image5D (B4) and 3D Viewer (B3) for volumic visualization. Another characteristic of TANGO is that it is linked to a database. MongoDB has been chosen (B2) because it is a powerful, easy-to-use, multi-platform object-oriented database system. It integrates the gridFS framework which allows efficient storage of files with no size limitation. MongoDB is used to store raw and processed images, which enables TANGO to manage voluminous sets of images. It also allows storage of data resulting from the quantification step, and user settings.

2.2 Data Architecture

Data architecture of TANGO has a hierarchical organization: an experiment contains *fields*, that contain *cells*, that are composed of one or several *structures*, each of them possibly having several segmented *objects*. This architecture corresponds to the underlying biological structure visible in acquired images, and is also used in the database and in the user interface. In this way, every element of the architecture has a biological significance.

Fields are composed of channel images. Each channel image (2D or 3D grayscale image) is acquired with particular microscopy settings, and the resulting signal corresponds to the localization of a labelled marker (e.g: probe or antibody). Typically, one of the channel images is used to label nuclei, usually a DNA staining, like DAPI.

Cells are composed of the cropped channel images around the nucleus bounds, and their associated *structures*.

A *structure* is a biological compartment of interest in the labelling experiment. There are several ways to define structures: in most cases, structures are extracted in each nuclei from a channel image. A given channel image may be used to extract information about one or more nuclear structures. For instance, DAPI staining of mouse nuclei can be used to segment both nuclei and chromocenters. For that reason, when defining an experiment in TANGO's interface, a user first defines channel images, then the structures, and associates a channel image to each structure. Another possibility is that a structure is defined by more than one channel, as in multi-FISH experiments. Those *structures* are termed *virtual structures* and are generated by ensemblist operations on several pre-existing structures. They do not have an associated channel image, thus their information is contained in the labelled image only. For instance, if probe 1 detects the biological compartments A and B and

probe 2 detects the biological compartments B and C, is it possible to create 3 virtual structures corresponding to the biological compartments A, B, and C only. Lastly, in some cases, it is irrelevant to segment some biological compartments: the corresponding *structures* are only represented by their associated grayscale signal.

The last elements of the architecture are the *objects* extracted from segmented *structures*. Hence, each object is identified by a value in the labelled image resulting from the processing sequence.

2.3 Processing Sequences

Processing sequence is termed as a set of elementary operations applied to a grayscale channel image to segment the structures it contains. It is composed of three steps: *pre-processing*, *segmentation* and *post-processing*. For simplicity, sequences to segment nuclei have been separated from sequences that segment other structures. Up-to-date algorithms for processing and segmentation have been implemented in order to deal with the large variety of structures encountered in the nucleus, such as the nucleus itself, diffuse regions such as chromosomal territories and spot-like objects. Template processing sequences can be defined once for each kind of structure, by the user, and can be re-used in different experiments.

The *pre-processing* step is optional and can contain a sequence of several filters. The aim of this step is usually to correct common imaging artifacts such as diffraction artifacts, nonuniform illumination, bleed-through of signals between imaging channels, non specific labelling, and noise. Different filters have been implemented, such as PureDenoise (A10), 3D Gaussian or 3D median. The pre-processing step should also prepare the image for the segmentation step by enhancing the signal to noise ratio, using for instance 3D band-pass, topHat (A3) or Laplacian of Gaussian (LoG) filtering (A9) to enhance spot-like objects of specific size.

Segmentation is the process of identifying *objects* in the pre-processed image. It generates a labelled image, in which a value (label) is assigned to each object. Segmentation algorithms are usually dependent on the shape of the objects (e.g spots *versus* chromosome territories), and sometimes on the type of image and noise (e.g: confocal *vs* epifluorescence). Robust algorithms for segmentation of nuclear objects have been implemented, for example seed-based 3D watershed algorithm.

Like the pre-processing step, the *post-processing* step is optional and can be composed of several filters. The aim of this step is to correct the result of segmentation and to produce a corrected labelled image. Classical operations are removing of small objects and morphological operations such as Closing. Those filters work on a labelled image and generate a new corrected labelled image.

2.4 Quantification

After each structure is segmented, TANGO performs the quantification step to extract features from each nuclei. The quantification is carried-out via particular plug-ins, termed here *Measurements* plug-ins. They can use information contained in segmented objects, as well as in raw grayscale images. In order to cover the maximum range of encountered cases, two main category of measurements have been defined:

1. Measurements on individual objects: a value is computed for each object of a given structure, for instance shape descriptors such as volume, surface, compacity, etc.
2. Measurements between one or more structures:
 - Measurements between two objects from same or different structures: for instance distance between centers of two objects.
 - More generic measurements between one or more structures. The resulting value can be either a scalar or an array. For instance measurement of a texture parameter on a grayscale image.

A database structure has been designed that fits these measurement categories, so that they can be easily and efficiently accessed. Basic quantifications have been implemented such as distance measurements, shape descriptors, quantification of fluorescence signal, etc. Some ImageJ plug-ins are already available in TANGO, such as JACOP (A8) for voxel and object-oriented colocalization or Texture analysis (A11).

2.5 Implementing plug-ins

We have extended the notion of plug-ins already available in ImageJ, to define the characteristics of the elementary operations assembled by TANGO. The implementation of plug-ins is made via Java interfaces that constrain inputs, outputs and parametrization. There are several types of TANGO plug-ins, the most important being: pre-processing, post-processing, segmentation and measurements. As described in the previous sections, each have specific functions, thus specific inputs and outputs. ImageJ plug-ins can be easily modified to be used by TANGO, a more detailed explanation and examples are available on the website. Once implemented, TANGO lists the available plug-ins in the appropriate sections, using the same plug-in system as ImageJ.

Like ImageJ plug-ins, TANGO plug-ins need input parameters. To obtain a generic and standardized parametrization of plug-ins, a new class of parameters has been designed. Parameters needed by the plug-in are specified by the developer when implementing a TANGO plug-in interface. To help parametrization, developers can enter default values, and limit some parameters to a range of values. When parametrizing the plug-in, a panel will be generated in the interface, where users can set the parameters. The settings will be saved in the database, then retrieved and made accessible to the plug-in before it is run.

The parameter class has several subtypes. There are canonical parameters (numbers, text, boolean) and more sophisticated parameters such as parameters that call an external plug-in. For instance, thresholders, another type of TANGO plug-in that compute a threshold value out of a grayscale image, can be used as a parameter. For segmentation, a threshold value is often required. Instead of a being constant value, a thresholder can be used so that the threshold value will depend on input image. When parametrizing the plug-in, the user can choose among any available thresholders in TANGO, that includes ImageJ thresholding methods. The list of thresholders can be extended like any other TANGO plug-in. From a development perspective, those parameters are as easy to use as canonical parameters: the threshold value is computed with a single method call. This allows a very generic parametrization of plug-ins, which can increase their range of application. The list of available parameters is detailed on the website.

2.6 Documentation

A interactive form of documentation has been integrated in TANGO. When a user mouses over documented element in the interface, the associated documentation is displayed in a separate window that can be easily accessed through the interface. There are two sources of documentation: internal documentation and plug-in documentation. The internal documentation of TANGO is associated to each part of the interface. Documentation is also synchronized between the interface and the website, so that if the help is updated on the website, it is also updated in the interface. Developers can write general documentation on their plug-in, and it is associated to the plug-in text. Each parameter needed by the plug-in can also be documented.

3. RESULTS

When using TANGO's interface the user first connects to the database, locally or on a remote server. Then existing experiments can be browsed or new ones created. The first step in creating an experiment is defining its structure: number of channel images, structures and measurements. Images corresponding to microscopy fields are added to the experiment, the program will then automatically save the images to the database, create thumbnails and display the list of fields. All the formats compatible with Bio-formats (B6)(over 100 image formats supported) can be imported. For each structure, a pre-defined processing sequence can be associated and easily tested on images. The user can then run the whole process, or only parts of it if so desired (see Figure 2):

- 3D-Segmentation of nuclei in each microscopy field. Usually an image corresponding to DNA staining is used for this step.
- Segmentation of structures within each nucleus.
- Quantification of all structures inside nuclei, potentially using all the corresponding segmented and raw images.

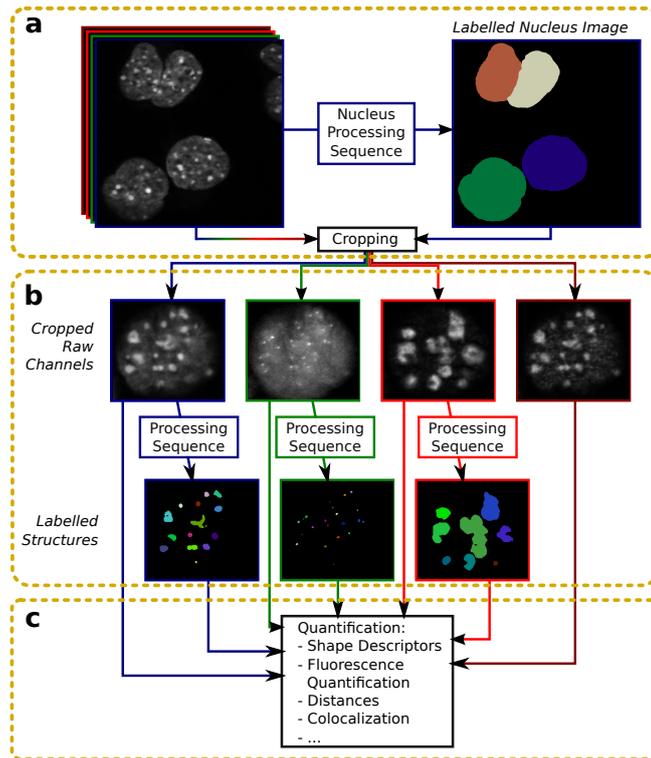


Figure 2. Data processing workflow is composed of three main steps: (a) Segmentation of nuclei in each field, and cropping of channel images around the bounding box of each nucleus. (b) Segmentation of structures in each nucleus. (c) Quantification (feature extraction) in each nucleus, potentially using all raw channel images and labelled images.

The results can then be visualized in different ways. Raw and segmented data can be opened with standard ImageJ visualization, Image5D for an overlay visualization or 3D Viewer for 3D visualization. To assess the quality of segmentation, ROIs of segmented objects can be displayed on opened images, and will be automatically updated when changing the Z plane. All segmented data, nuclei and structures, can be manually corrected using semi-manual segmentation plug-ins. Quantitative results are saved in the database and can be displayed in the interface. An R-package was developed, named *rTango* (available on our website), following the same data structure as TANGO, allowing easy extraction of all types of data under R, directly from the database. For compatibility with other statistical processing softwares, it is also possible to extract data and generate spreadsheet-compatible files, like Results Table in ImageJ.

3.1 User Interface

The interface is composed of five tabs (see Figure 3). The *Connect* tab is for connection with the database, import/export of data and general options. The *Edit Experiment* is where the architecture of the experiment is set (channel images, structures and measurements). The *Nucleus* and *Structure Processing sequence* tabs allow the processing sequences to be defined. The same template processing sequence can be used in different structures. They can be imported and exported. The *Data* tab is the data manipulation tab, where images are imported in the database and then processed. It permits a rapid browsing of images, which conveniently enables assessment of processing quality, as described in Figure 3. Automatically generated thumbnails allow an overall visualization of images. The architecture of this tab reflects the hierarchical architecture of data in TANGO. When a selection on a container changes, the list of contained objects is updated. The user interface is used for parametrizing experiments and processing sequences, however, every action can be run either via the interface or via command lines, for instance in the case of remote usage.

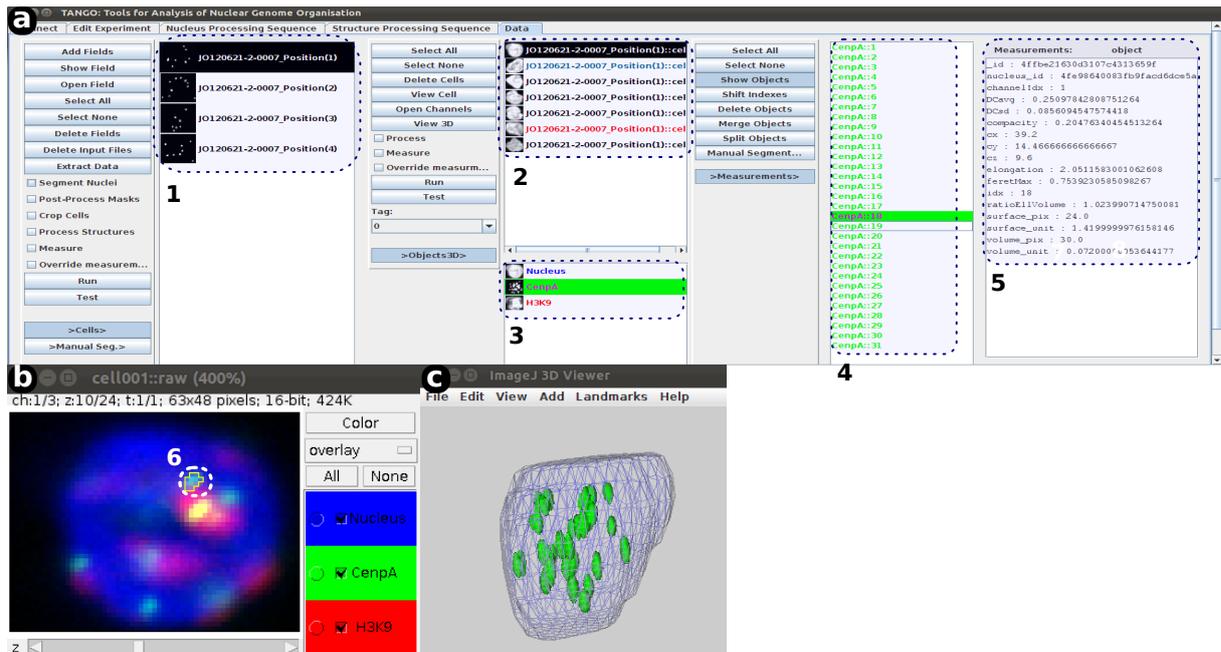


Figure 3. (a) User Interface: The User has access to fields (1), cells within fields (2), structures within cells (3) and objects within structures (4). Measurements are displayed in (5) and will automatically be updated when the selected structures and objects change. ROIs of objects are displayed on the active image (6). It is possible to visualize fields and cells with a color-combined stack view (using the plug-in image5D)(b), or in 3D (using the plug-in 3D viewer)(c).

4. CONCLUSIONS

TANGO is an integrated analysis platform that performs segmentation and various quantifications on labelling experiments, designed to manage large sets of images. The interface also provides tools for manual and semi-automatic segmentation, and rapid assessment of segmentation quality. Multidimensional data resulting from quantification is formatted to be easily processed by statistical softwares. Integration of various existing processing and quantification ImageJ plug-ins, makes TANGO a very versatile tool. TANGO has been designed for analysis of nuclear architecture: it performs the analysis of different structures within nuclei. It could also be extended to other applications with a compatible data structure: as long as input images are 2D or 3D grayscale images, with one or several channels and they may contain at most one level of sub-compartments (nuclei in our case).

APPENDIX A. IMAGEJ PLUGINS USED IN TANGO

- A1. 3D Analysis: http://imagejdocu.tudor.lu/doku.php?id=plugin:stacks:3d_roi_manager:start
- A2. 3D Euclidean Distance Map: http://www.optinav.com/Local_Thickness.htm
- A3. 3D Filters: http://imagejdocu.tudor.lu/doku.php?id=plugin:filter:3d_filters:start
- A4. 3D Spots Segmentation: http://imagejdocu.tudor.lu/doku.php?id=plugin:segmentation:3d_spots_segmentation:start
- A5. BoneJ: <http://bonej.org/>
- A6. DropletFinder: http://imagejdocu.tudor.lu/doku.php?id=plugin:analysis:droplet_counter:start
- A7. FeatureJ: <http://www.imagescience.org/meijering/software/featurej/>

- A8. JACOP: http://imagejdocu.tudor.lu/doku.php?id=plugin:analysis:jacop_2.0:just_another_colocalization_plugin:start
- A9. LoG3D (Laplacian of Gaussian): <http://bigwww.epfl.ch/sage/soft/LoG3D/>
- A10. PureDenoise: <http://bigwww.epfl.ch/algorithms/denoise/>
- A11. Texture analysis: <http://rsbweb.nih.gov/ij/plugins/texture.html>

APPENDIX B. SOFTWARES/TOOLS USED BY TANGO

- B1. ImageJ: <http://imagej.nih.gov/ij/>
- B2. MongoDB: <http://www.mongodb.org/>
- B3. 3D Viewer: <http://3dviewer.neurofly.de/>
- B4. Image5D: <http://imagej.nih.gov/ij/plugins/image5d.html>
- B5. Jericho: <http://jericho.htmlparser.net/docs/index.html>
- B6. Loci-tools: <http://loci.wisc.edu/software/bio-formats>