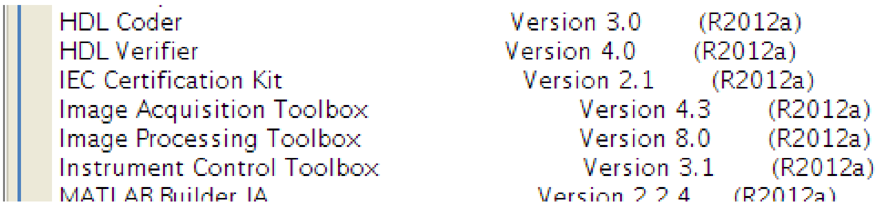
GPDQ: Gold Particle Detection and Quantification v0.2

User manual

Introduction

This document presents all the information required to run the software QPDQ for gold particle detection and quantification in michroscopical images.

The software is written in Matlab, therefore it is necessary to have this tool installed. Moreover, it uses the *Image Processing Toolbox*, which must be installed as well. The command `ver` allows checking the packages installed in Matlab. Figure 1 shows part of the output produced.



HDL Coder	Version 3.0	(R2012a)
HDL Verifier	Version 4.0	(R2012a)
IEC Certification Kit	Version 2.1	(R2012a)
Image Acquisition Toolbox	Version 4.3	(R2012a)
Image Processing Toolbox	Version 8.0	(R2012a)
Instrument Control Toolbox	Version 3.1	(R2012a)
MATLAB Builder JA	Version 2.2.4	(R2012a)

Figure 1: Output of command `ver`

If the toolbox is not installed, then it has to. It can be done by reinstalling Matlab and selecting this package.

Preparing the images

On its *current state* the software is prepared to deal with a specific kind of images. In particular, the extension of the images must be **`.tif`**. Moreover the size of the main picture (without information) must be *1024 x 1024 pixels*, and it must be located in the top left corner.

For each image `file.tif`, where “`file`” can be any name, there must be another image called `file_mod.tif` containing the section of interest. In this second picture the discarded areas are white. Figure 2 shows a pair of images. The first one contains the original picture, whereas the second contains the section. The white area in the bottom contains all the information related to the parameters of the photo, and is discarded by the program.

Sections can be obtained by any software available, such as imageJ.

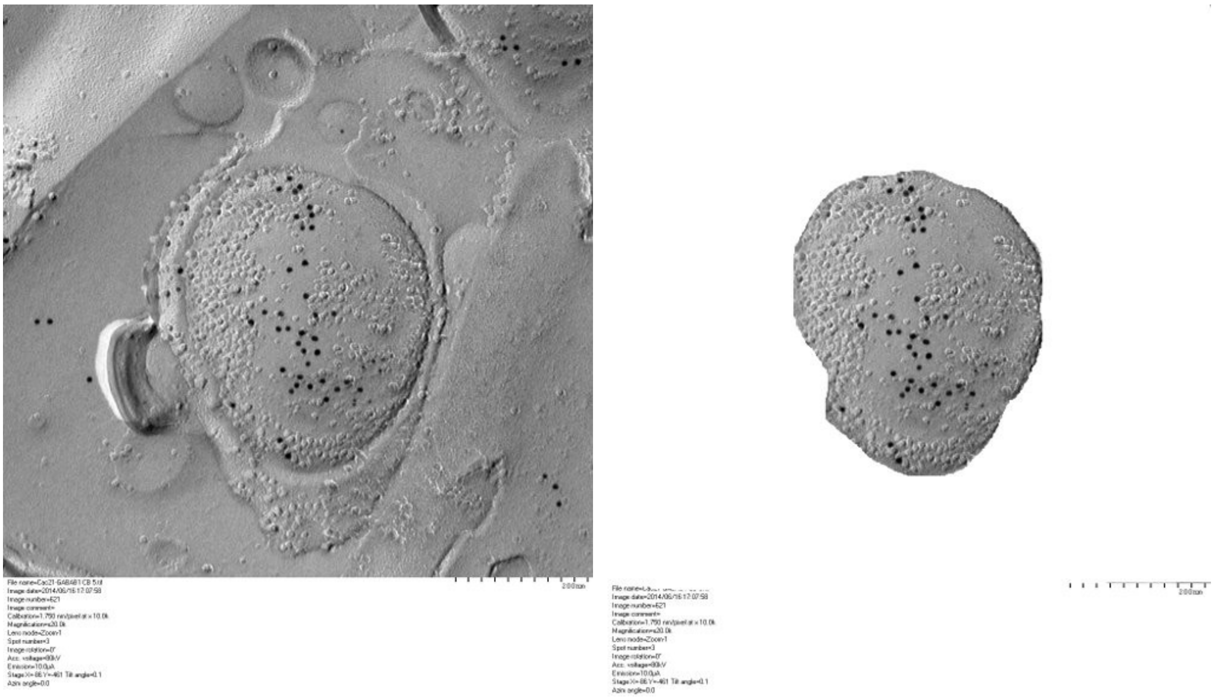


Figure 2: Original image and image with section of interest.

List of images

There are some parameters related to the images that can't be read automatically: Group, calibration and magnification. Because of that, it is necessary to elaborate a text file containing the data related to all the images in the study. This file is a `.csv` (*comma separated values*) which can be edited with any text editor.

Figure 3 shows an example of file. The header must contain the name of each one of the four fields in this precise order. Then, each row contains the values for each field sepparated by comma. The route to the image is the name of the image without extension. Thus the first row refers to the images `1.tif` and `1_mod.tif` which are in folder `AXON`, belong to the group `AXON1`, and has a calibration *1.750* and a magnification of *12*.

```
1 GROUP,  ROUTE,   CALIBRATION,    MAGNIFICATION
2 AXON1,  AXON/1,  1.750,          12
3 AXON1,  AXON/2,  1.750,          20
4 AXON2,  AXON/3,  1.750,          20
5 AXON2,  AXON/4,  1.750,          25
6 AXON,   AXON/5A,         1.750,  20
```

Figure 3: Example of configuration file

Using the software

Once the images and the image list have been prepared, the software can be used. In order to do that, the working folder of Matlab must be set to QPDQ. Then the software can be used.

particleLabeling

The command `particleLabeling` , which must be typed in the Matlab console, runs the program which allows labeling images. This program asks for a configuration file with the list of the images which is shown in Figure 4. When loaded, it allows processing each image separately.

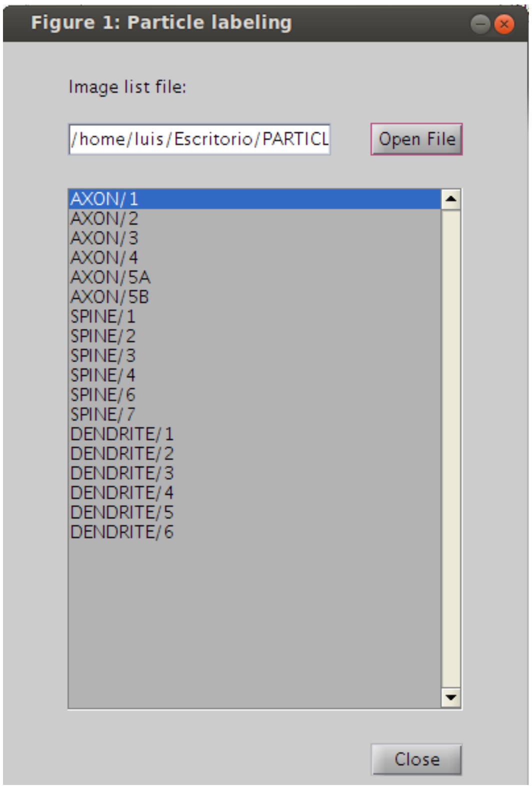


Figure 4: Selection of the image to be labeled

When one image is clicked, then it opens the window which allow marking the points. It is shown in Figure 5. There are two images. The left one shows the entire image. When clicking on it of moving the rectangle, the program shows an ampliation in the right image, and allows selecting the particles. Although selection is manual, **the exact positions of the circles are detected automatically**. Therefore, it is not necessary to click over the particle, and the labeling is fast.

Once the diameter of the points is detected, they can be labeled. As they are so small, it is difficult for the *hough transform* to detect them. Because of that, there is a margin in the expected diameter. This is set to 30%, but it is better sometimes to fix it to 40%. In relation with the sensitivity, higher values allow detecting more circles, but with the cost of introducing many false positives.

Last, it is possible to clear the image (with the button). Save and close saves all the marked dots, whereas if closing the window it is shown a dialog which allows discarding the changes.

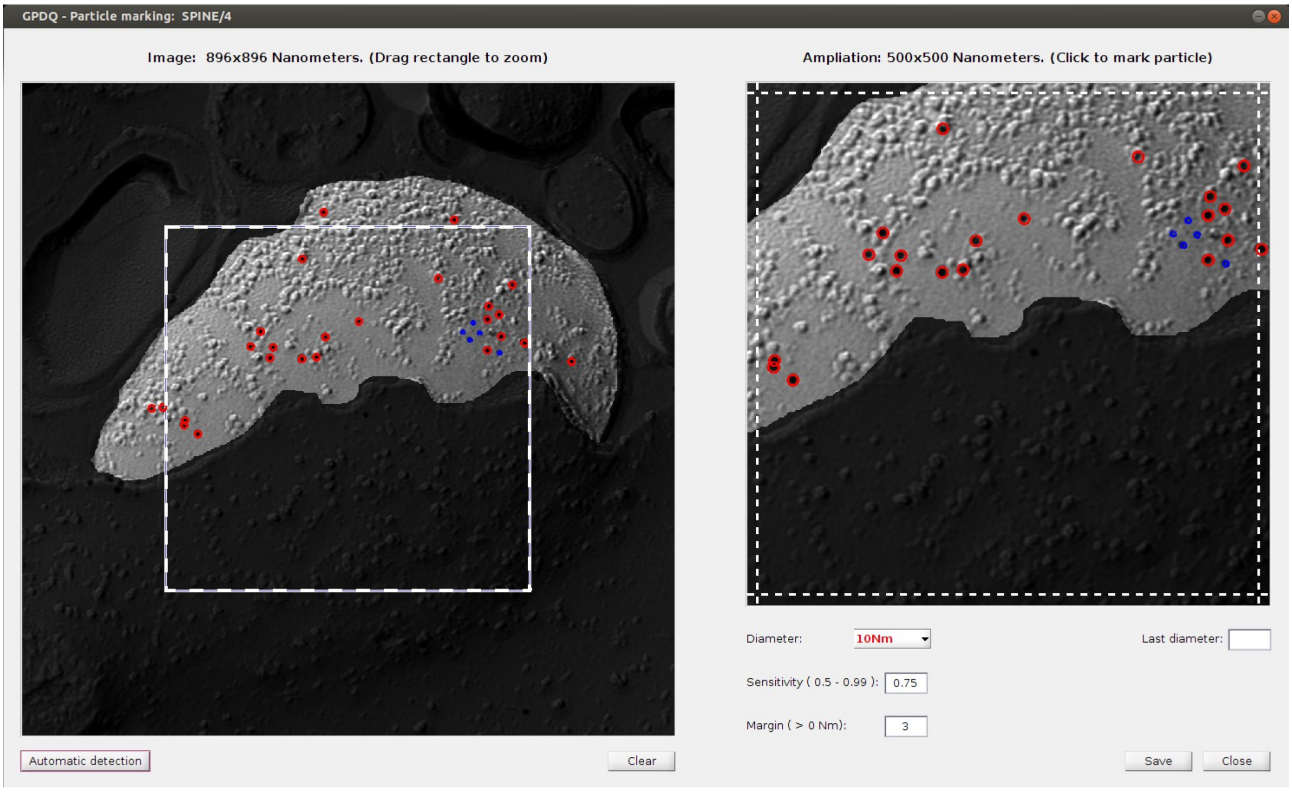


Figure 5: Labeling

If the route to the image being labeled is `AXON/1` it refers to both `AXON/1.tif` and `AXON/1_mod.tif`. The dots are automatically stored in a file called `1dots.csv` which contains their locations (in nanometers with respect to the top left corner) their actual radii and the expected radii (2.5 or 5 nanometers).

Automatic Labeling

For 10 Nm particles, the software allows fully automatic labeling. When clicking the button in the bottom left corner, a new window, as the one shown in Figure 6, shows the process of automatic detection. The detected particles can be either added or discarded. If added, the software adds those particles which have not been already selected.

Parameters for automatic labeling are the same used for manual labeling. Therefore, it is possible to test them before adding new particles.

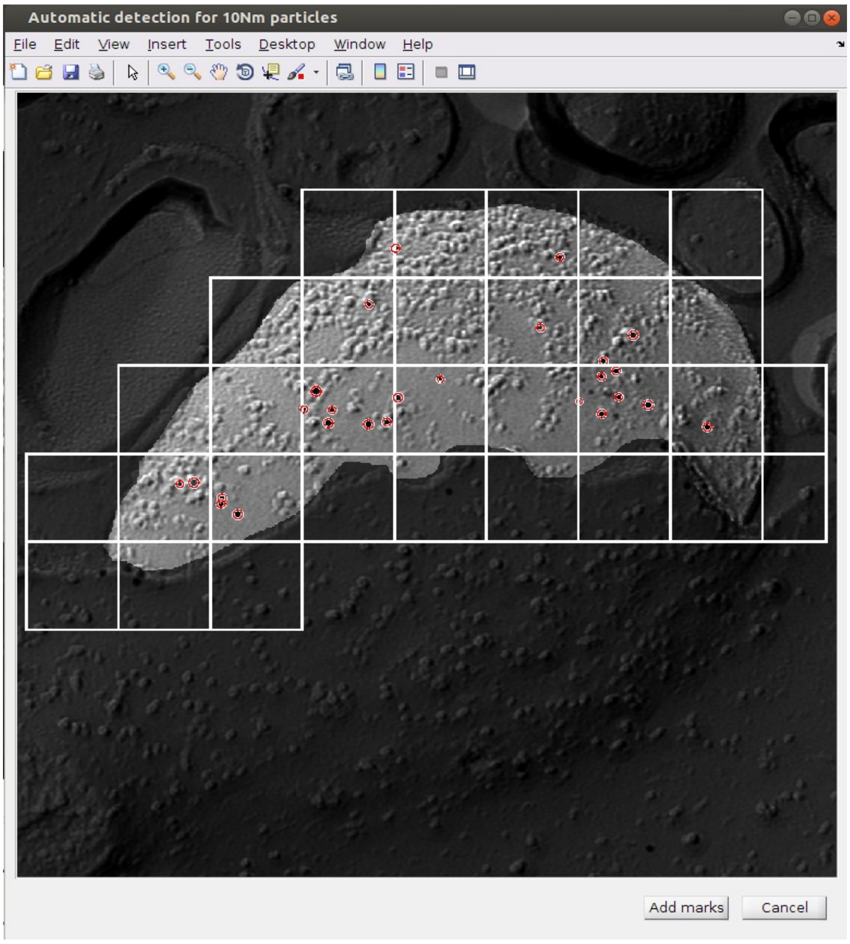


Figure 6: Automatic labeling

generateReports

This command **`generateReports`** generates the reports with all the information. It shows a dialog which allows selecting the file containing the data of the list of images and the folder where the reports will be generated. It is shown in Figure 7.

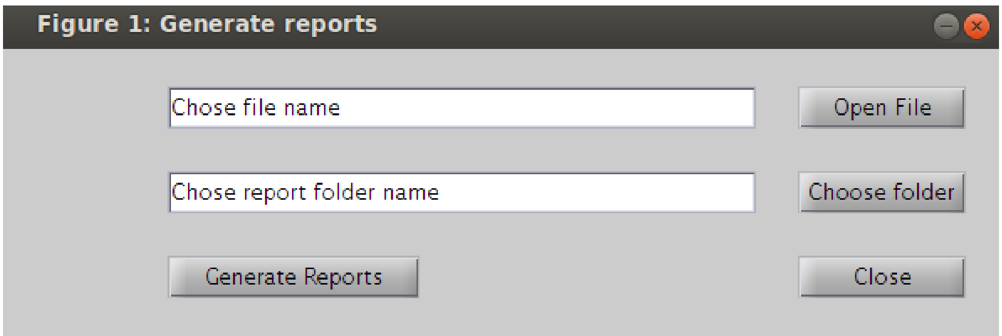


Figure 7: Report generation dialog

The reports are csv files than can be imported with excel. All of them, and their columns, are described next.

`images.csv`: Data relative to the images.

- `IMG_ID`: Numerical identifier of the image.
- `ROUTE`: Base name of the image (including folder).
- `GROUP`: Group the image belongs to.
- `SCALE`: Scale of the image (nanometers / pixel).
- `AREA`: Area of interest in the image (squared nanometers)
- `TOTAL_NUM_PARTICLES`: Total number of particles.

`partCounting.csv`: Counts the number of particle by size.

- `IMG_ID`: Numerical identifier of the image.
- `ROUTE`: Base name of the image (including folder).
- `GROUP`: Group the image belongs to.
- `NUM_PART_2.50 Nm`: Number of particles with radius 2.5Nm in the image.
- `NUM_PART_5.00 Nm`: Number of particles with radius 5Nm in the image.

`distNearestPart.csv`: For each particle, returns the distance to the nearest neighbor (does not distinguish particles).

- `IMG_ID`: Numerical identifier of the image.
- `ROUTE`: Base name of the image (including folder).
- `GROUP`: Group the image belongs to.
- `PART_NUMBER`: Id of the particle.
- `RADIUS`: Radius of the particle.
- `DIST_NEAREST_PART`: Distance to the nearest particle.

`distNearest5NmPart.csv`: For each particle, returns the distance to the nearest 5Nm neighbor.

- `IMG_ID`: Numerical identifier of the image.
- `ROUTE`: Base name of the image (including folder).
- `GROUP`: Group the image belongs to.
- `PART_NUMBER`: Id of the particle.
- `RADIUS`: Radius of the particle.
- `DIST_NEAREST_PART`: Distance to the nearest particle with radius 5Nm.

`DistNearest2.5NmPart.csv`: For each particle, returns the distance to the nearest 2.5Nm neighbor.

- `IMG_ID`: Numerical identifier of the image.
- `ROUTE`: Base name of the image (including folder).
- `GROUP`: Group the image belongs to.
- `PART_NUMBER`: Id of the particle.
- `RADIUS`: Radius of the particle.
- `DIST_NEAREST_PART`: Distance to the nearest particle with radius 2.5m.

`clusteringAll.csv`: For each image, contains the results of the clustering. (Does not distinguish particles).

- `IMG_ID:` Numerical identifier of the image.
- `CLUSTER_ID:` Identifier of the cluster.
- `NUM_PARTICLES:` Number of particles in the cluster (considers All).
- `NUM_PART_2.50:` Number of particles with radius 2.5Nm in the cluster.
- `NUM_PART_5:` Number of particles with radius 5Nm in the cluster.
- `AREA:` Area of the cluster (Convex Hull).
- `DENSITY:` Density of particles in the area of the cluster.
- `RIPLEY'S L:` Stabilized Ripley's K function in the cluster.
- `DISTANCE_TO_NEAREST:` Distance to the nearest cluster.

`clustering5.csv`: For each image, contains the results of the clustering. (Only considers particles with radius 5Nm).

- `IMG_ID:` Numerical identifier of the image.
- `CLUSTER_ID:` Identifier of the cluster.
- `NUM_PARTICLES:` Number of particles in the cluster (radius 5Nm).
- `AREA:` Area of the cluster (Convex Hull).
- `DENSITY:` Density of particles in the area of the cluster.
- `RIPLEY'S L:` Stabilized Ripley's K function in the cluster.
- `DISTANCE_TO_NEAREST:` Distance to the nearest cluster of particles with radius 5Nm.

`Clustering2.5.csv`: For each image, contains the results of the clustering. (Only considers particles with radius 2.5Nm).

- `IMG_ID:` Numerical identifier of the image.
- `CLUSTER_ID:` Identifier of the cluster.
- `NUM_PARTICLES:` Number of particles in the cluster (radius 2.5Nm).
- `AREA:` Area of the cluster (Convex Hull).
- `DENSITY:` Density of particles in the area of the cluster.
- `RIPLEY'S L:` Stabilized Ripley's K function in the cluster.
- `DISTANCE_TO_NEAREST:` Distance to the nearest cluster of particles with radius 2.5Nm.

Visualization.

The command **`visualize`** allows visualizing the list of the images in a configuration file. Besides the image, it shows some basical information. Moreover, it allows showing the marks for particles and clusters. Figure 7 shows an screenshot of this function. As can be seen, it shows the marks for both kinds of particles, and the clusters for small particles (two clusters).

The button '`Export`' under the image allows exporting the image which is currently shown to a '`.png`' file.

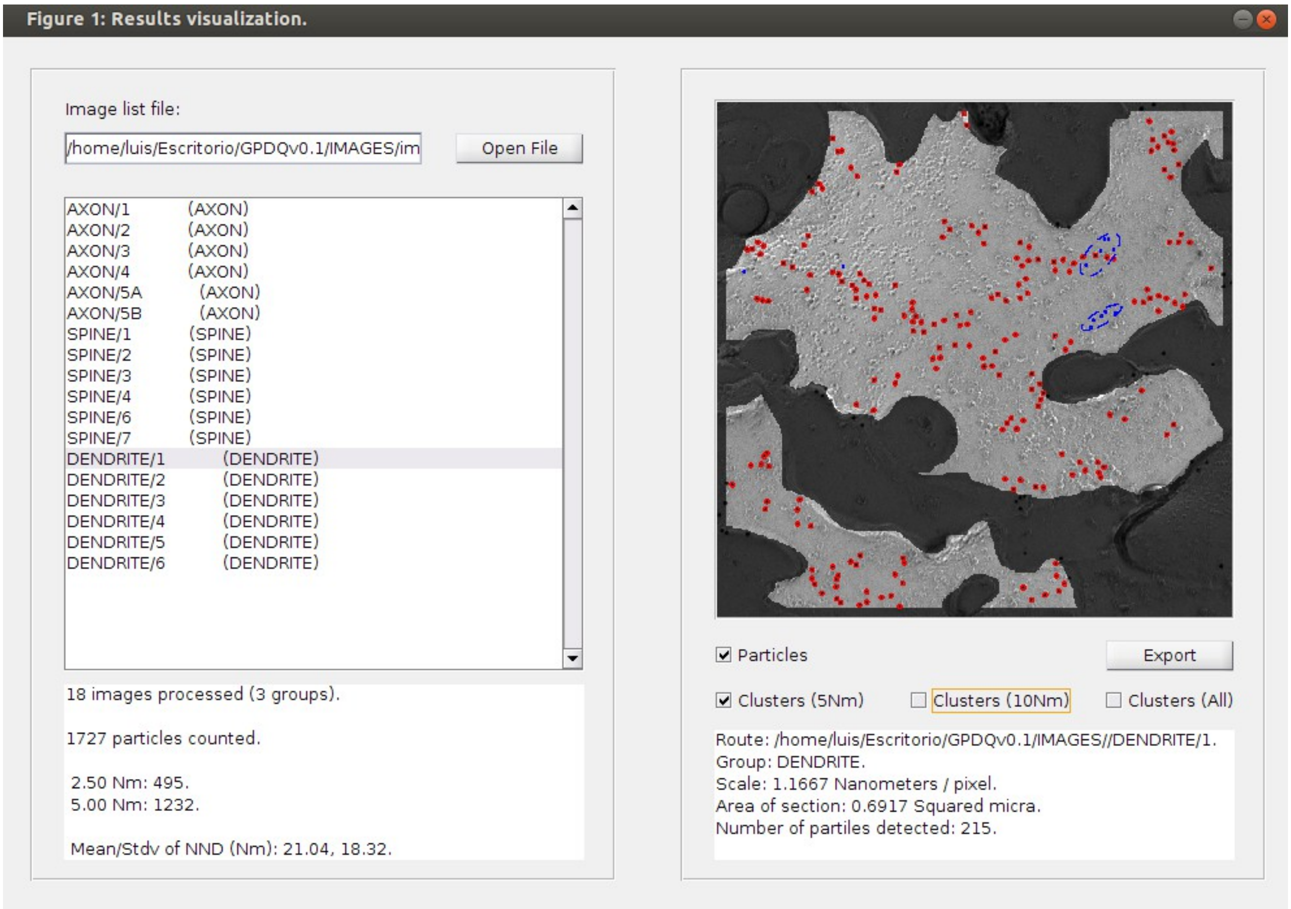


Figure 8: Visualization