



JAN 24, 2024

OPEN  ACCESS**DOI:**

[dx.doi.org/10.17504/protocols.io.e6nvwdmnwlmk/v1](https://dx.doi.org/10.17504/protocols.io.e6nvwdmnwlmk/v1)

**Protocol Citation:** Patricia Lumanto, Diana Nakib, Xinle Wang, Sonya Macparland 2024. QuPath Immunofluorescence Cell Detection and Co-localization Protocol. [protocols.io](#)  
<https://dx.doi.org/10.17504/protocols.io.e6nvwdmnwlmk/v1>

**License:** This is an open access protocol distributed under the terms of the [Creative Commons Attribution License](#), which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited

**Protocol status:** Working

**Created:** Nov 13, 2023

## QuPath Immunofluorescence Cell Detection and Co-localization Protocol

Patricia Lumanto<sup>1,2</sup>, Diana Nakib<sup>1,2</sup>, Xinle Wang<sup>1,2</sup>, Sonya Macparland<sup>1,2</sup>

<sup>1</sup>University of Toronto; <sup>2</sup>University Health Network

University Health Network, MacParland Lab



Patricia Lumanto

University of Toronto, Scarborough, University Health Netw...

### ABSTRACT

QuPath is an open-source imaging software that provides a user-friendly platform for digital pathology and whole slide image analysis. This protocol aims to provide a step-by-step introduction to optimizing and running total cell detection and positive cell detection on QuPath for Immunofluorescence (IF) images. The protocol offers the possibility of assessing stain localization and co-localization within the liver immune environment and can be further applied to other tissues. Positive cell detection allows for the quantification of marker-specific positive cells across the tissue sample, and stain co-localization allows for the characterization of cells that are positive for two markers of interest, representing specific cell-cell interaction within the tissue sample, both providing a better understanding of the liver microenvironment.

This protocol was used to quantify and characterize the localization of different liver immune cell subpopulations (presence of liver sinusoidal endothelial cells, Kupffer cells, cholangiocytes, and bipotent hepatocytes) as seen in Andrews & Nakib et al. (J Hepatol 2024 doi:[10.1016/j.jhep.2023.12.023](https://doi.org/10.1016/j.jhep.2023.12.023)).

### MATERIALS

QuPath v0.4.3

Last Modified: Jan 24, 2024

PROTOCOL integer ID: 90862

## Image Brightness and Contrast Adjustment

- 1 Click on **File: Project** to start a project.

**Note:** a project must be created in order for annotations and measurements to be exported.

**Tip:** prepare an empty folder with an appropriate folder name prior to starting the project.

- 2 Import the desired image by clicking on **File: Open**.

**Note:** scanned image files (e.g. .czi, .tiff) may take up quite a bit of space on your internal memory, make sure sufficient space is available on your local drive.

- 3 Select **View: Brightness/Contrast**

- Select the desired channels for visualization (e.g. DAPI, AF568)



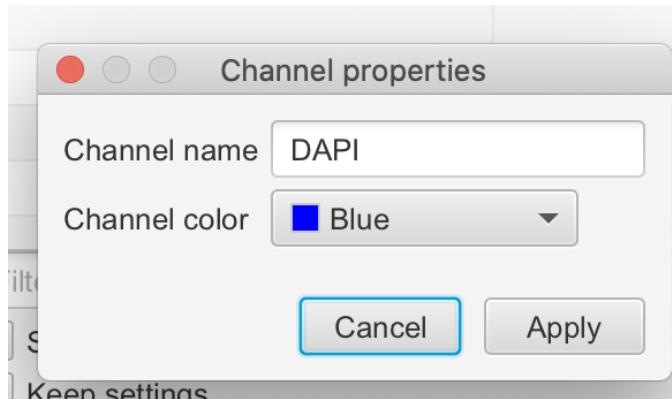
Color	Selected
DAPI (C1)	<input checked="" type="checkbox"/>
AF568 (C2)	<input checked="" type="checkbox"/>
AF647 (C3)	<input type="checkbox"/>

Table containing the different stains used for this experiment and their respective colors.

- 4 Double click on any channel to change its name and color.

**Note:** the original scan colors are not permanent and can be changed to make visualization easier.

**Tip:** complementary colors (e.g. red and green, yellow and purple) will be easiest to distinguish for visualization.

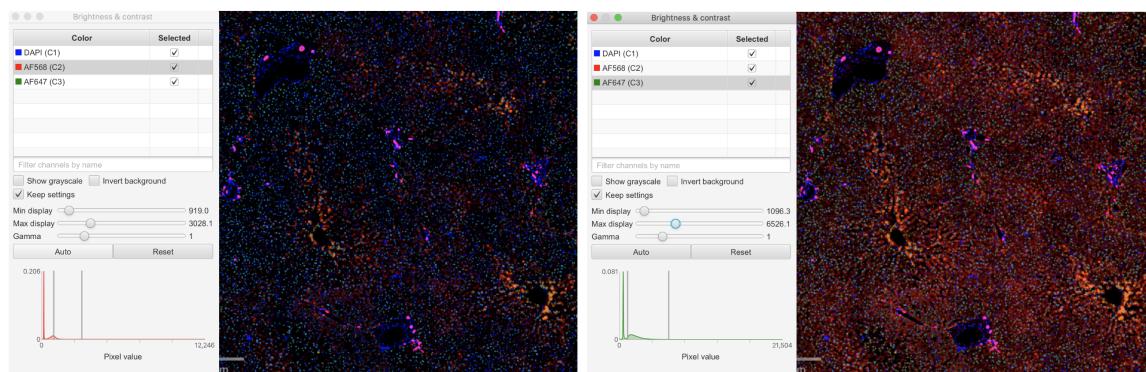


Pop up box to be used to change the channel name and channel color, through the drop down menu.

- 5** Adjust the brightness and contrast of the image by changing the min and max display values using the slider.

**Note:** The default gamma value is 1.

**Tip:** Refer to IHC experiments (e.g. from The Human Protein Atlas) to help visualize expected stain localization and intensity.



Comparison of an optimally adjusted image (left) vs oversaturated image (right).

## Region of Interest Selection and Annotation

- 6** Use **rectangle tool** to select region of interest.

- 7** Rotate the rectangle by using the transform tool (**Objects: Annotations: Transform annotations**).

- 8 Select the rectangle and run the following script (**Automate: Show script editor**).

**Note:** if lengthwise sections are preferred, change "d1 < d2" to "d1 > d2" (on line 45 of the script).

**Tip:** change the name for each region of interest by changing the ROI number ("ROI 1 Layer \${i+1}") on the last line of the script.

```
-----  
-----  
  
/**  
 * Split a rectangle into equal parts along its longest dimension.  
 * Rotated rectangles are supported.  
 *  
 * Written for https://forum.image.sc/t/how-to-divide-annotation-roi-into-  
equal-dimension-bins/51563/8  
 *  
 * @author Pete Bankhead  
 * Taken from  
https://gist.github.com/petebankhead/e177b07784460a76d70f89347edd988d  
 */  
  
// Number of regions to create  
int nRegions = 10  
  
// Get selected object  
def selected = getSelectedObject()  
if (selected == null) {  
    println 'No object selected!'  
}  
  
// Get points, removing duplicates  
def roi = selected.getROI()  
def points = new LinkedHashSet<>(roi.getAllPoints()) as List  
if (points.size() != 4) {  
    println 'I need a ROI with exactly 4 points'  
    return  
}  
  
// Get the side lengths  
double d1 = points[1].distance(points[0])  
double d2 = points[2].distance(points[1])  
double d3 = points[3].distance(points[2])  
double d4 = points[0].distance(points[3])  
  
// Check we really have a rectangle  
// (Set this to a higher number of it's too strict)  
double eps = 0.01  
if (Math.abs(d1 - d3) > eps || Math.abs(d4 - d2) > eps) {  
    println 'Points do not appear to form a rectangle!'  
    return
```

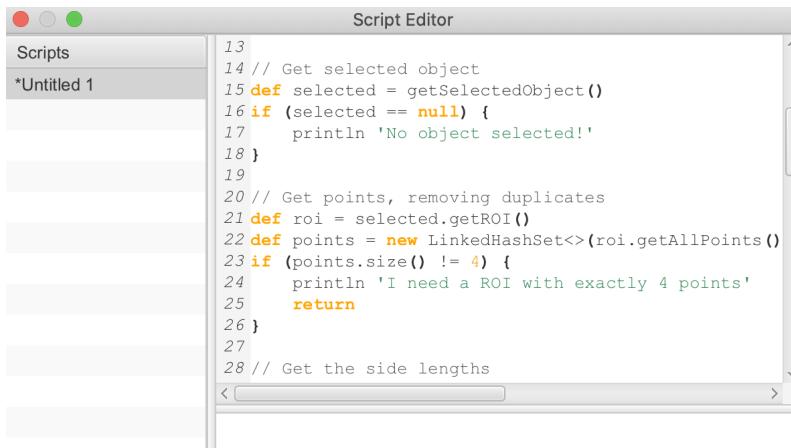
```
}

// Get starting point based on longest side
int ind = 0
if (d1 < d2) {
    points.add(0, points.remove(3))
}
double x = points[ind].x
double y = points[ind].y
double dx = (points[ind+1].x - x) / nRegions
double dy = (points[ind+1].y - y) / nRegions
double dx2 = (points[ind+2].x - points[ind+1].x)
double dy2 = (points[ind+2].y - points[ind+1].y)

// Add annotations
def annotations = []
for (int i = 0; i < nRegions; i++) {
    double originX = x + dx*i
    double originY = y + dy*i
    def polygon = ROIs.createPolygonROI(
        [originX, originX+dx, originX+dx+dx2, originX+dx2] as double[],
        [originY, originY+dy, originY+dy+dy2, originY+dy2] as double[],
        roi.getImagePlane()
    )
    def newAnnotation = PathObjects.createAnnotationObject(polygon)
    newAnnotation.setName("ROI 1 Layer ${i+1}") //To change ROI number
    annotations << newAnnotation
}
addObjects(annotations)
```

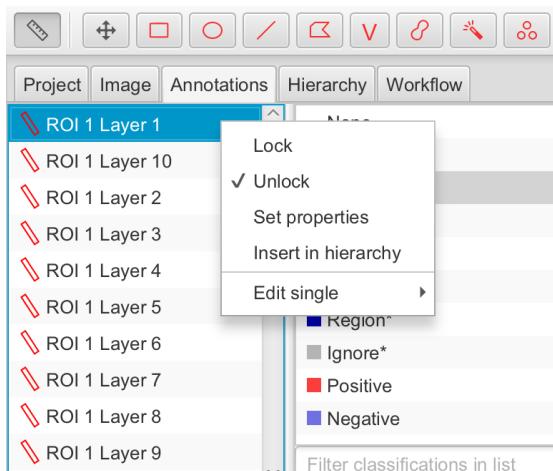
---

---



Script editor pop up box to be used when a script is required.

- 9** Rename **Annotations** by right clicking on the selected layer then select **Set properties**.



List containing the different layers and regions of interest along with their given names.

## Positive Cell Detection and Measurements

- 10** Select the region (largest rectangle) and click on **Analyze: Cell detection: Positive Cell Detection**.

- 10.1** Select the desired stain for cell detection by clicking on the drop down menu next to **Detection channel**.

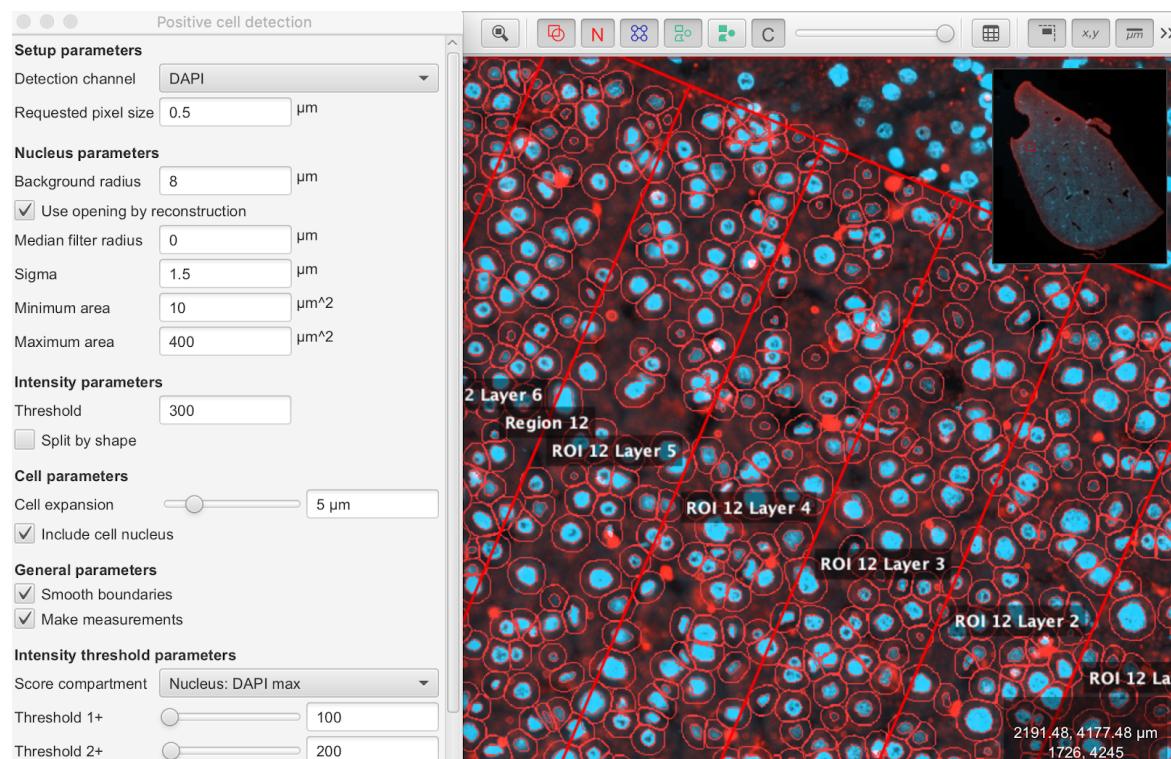
**10.2** Modify the **Threshold** value such that all positive cells are accounted for.

**Note:** a threshold of 300-400 would be ideal for a DAPI nuclei stain.

**10.3** Change the **Score compartment** by clicking on the dropdown menu next to it such that the stain selected matches its localization within the cell.

**Note:** it is easier to adjust the threshold when stain localization is known (e.g. Nucleus: DAPI max)

**Tip:** if unsure about stain localization, select the **cell** option for easiest detection.



Positive cell detection pop up box containing the different adjustable parameters.

**11** Click **Run** to start positive cell detection.

- Keep adjusting the **Threshold** value until the correct cells are detected.
- Note:** it may take several adjustments to optimize the threshold value, zoom in and out of the image, if necessary, to help determine appropriate detection.
- Repeat for other regions

## Cell Detection and Co-localization

**12** Select the annotation and run cell detection by clicking **Analyze: Cell detection: Cell detection**.

**12.1** Select the desired stain for cell detection by clicking on the drop down menu next to **Detection channel**.

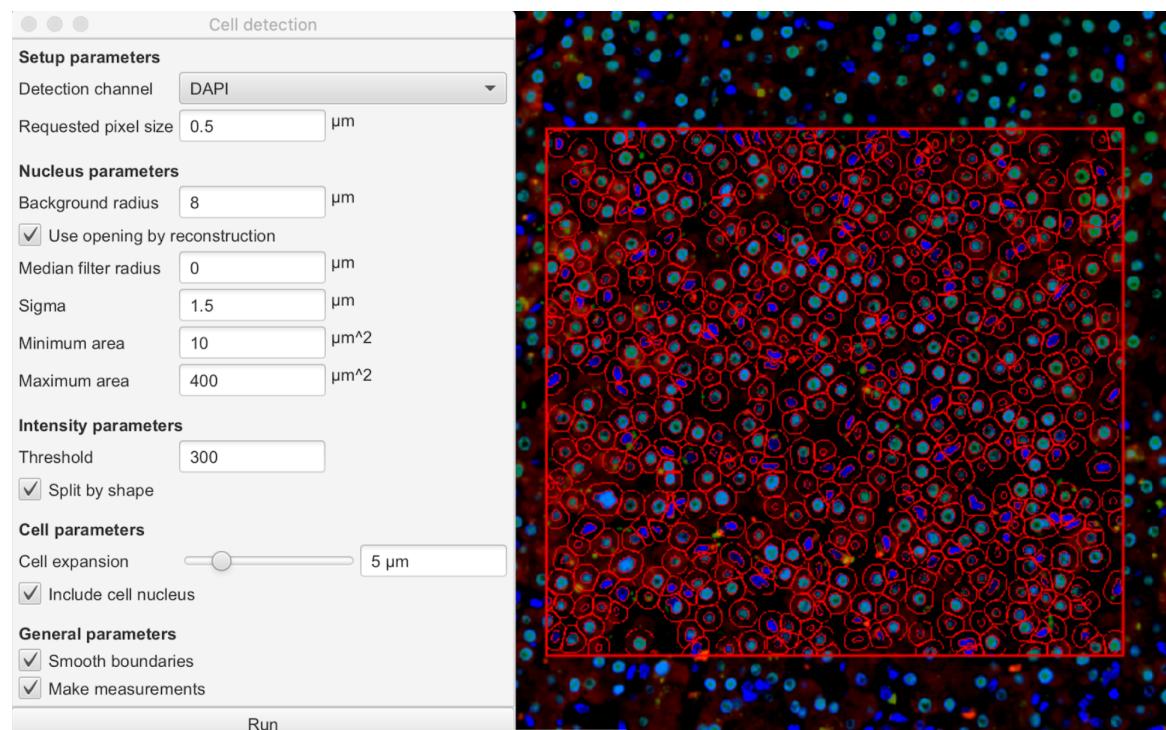
**12.2** Modify the **Threshold** value such that all positive cells are accounted for.

**Note:** a threshold of 300-400 would be ideal for a DAPI nuclei stain.

**12.3** Change the **Score compartment** by clicking on the dropdown menu next to it such that the stain selected matches its localization within the cell.

**Note:** it is easier to adjust the threshold when stain localization is known (e.g. Nucleus: DAPI max)

**Tip:** if unsure about stain localization, select the **cell** option for easiest detection.

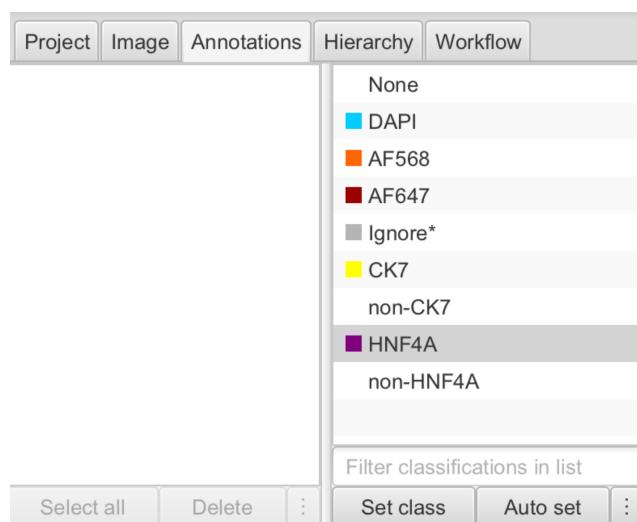


Cell detection pop up box containing the different adjustable parameters.

**12.4**

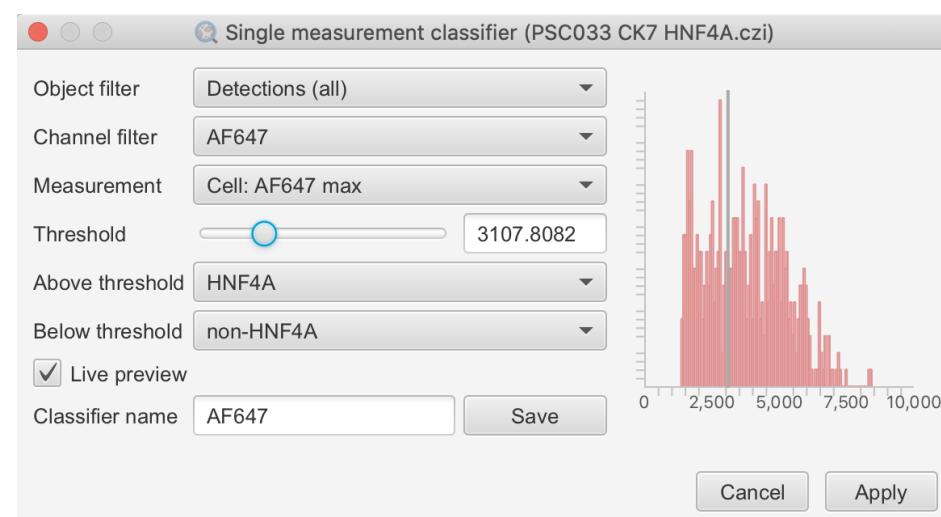
- If only cell detection or cell count is required, skip steps 12 to 16 and move forward with the file export.
- If co-localization assessment is required proceed to step 12.

- 13 Create new classes by going to **Annotations**, and right clicking on the class list to add a new class.  
**Tip:** creating classes for both positive and negative cells makes classification easier (e.g. HNF4A and non-HNF4A).



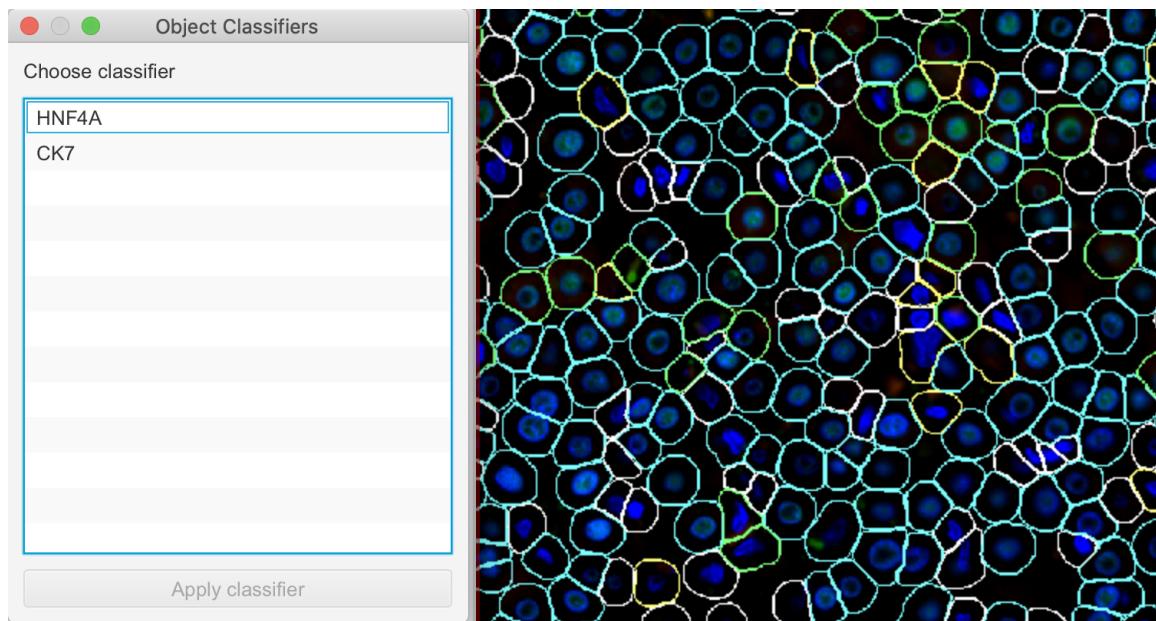
Class list containing positive and negative cell classes for a particular stain.

- 14 Create cell classifications by clicking **Classify: Object classification: Create single measurement classifier**.
- 14.1 Select the desired channel filter (e.g. DAPI, AF647) by clicking on the dropdown menu next to **Object filter**.
- 14.2 Adjust the **Threshold** using the slider.  
**Note:** turn on live preview for better visualization.
- 14.3 Save the classifier using the stain name.



Single measurement classifier window with the different adjustable parameters.

- 15 Select the region of interest (previously selected using the rectangle tool).
- 16 Classify object by clicking **Classify: Object classification: Load object classifier**.
- 17 Apply object classification by clicking on the name of each classifier then clicking **Apply classifier**.  
*Tip: To apply more than one classifier to the same region select both using the command or control key and click **Apply classifier sequentially**.*



Classifier window showing the different classes created applied to an immunofluorescence image. Colors shown corresponds to the same color as previously picked when creating classes on the **Annotations** window. The blue and yellow circles represent single positive cells, respective to stain; while the green and white circles represent double positive and negative cells, respectively.

## File Export

18

Click on **Measurements: Export measurements** to export data into a tsv file.

- Click on export annotations.
- Click on browse to select a name and file path for the export.

**Note:** the generated tsv file can be opened on spreadsheet programs such as Microsoft Excel and copied to data analyzing softwares.