

```

import qupath.lib.gui.tools.MeasurementExporter
import qupath.lib.objects.PathAnnotationObject
import qupath.lib.roi.ShapeSimplifier

// Create shape simplifier
def simplifier = new ShapeSimplifier()

// Separate each measurement value in the output file with a tab ("\t")
def separator = ";"

// Choose the columns that will be included in the export
// Note: if 'columnsToInclude' is empty, all columns will be included
def columnsToInclude = new String[]{"Image","Name","Num Detections","Num Negative","Num Positive","Positive
%", "Num Positive per mm^2", "Area Åµm^2"}

// Choose the type of objects that the export will process
// Other possibilities include:
// 1. PathAnnotationObject
// 2. PathDetectionObject
// 3. PathRootObject
// Note: import statements should then be modified accordingly
def exportType = PathAnnotationObject.class

def imagesToExport = [getProjectEntry()]
def imageData = getCurrentImageData()
def hierarchy = imageData.getHierarchy()
def annotations = hierarchy.getAnnotationObjects()
float elevation = 10.0

// Choose your *full* output path
def name = GeneralTools.getNameWithoutExtension(imageData.getServer().getMetadata().getName())
def outputPath = "/nb_projects/qupath_Projects/CD8CellDetection/measurements/" + name +
"_CD8_measurements.csv"
def outputFile = new File(outputPath)

hierarchy.getSelectionModel().clearSelection()
for (annotation in annotations) {
    annotation.setROI(ShapeSimplifier.simplifyShape(annotation.getROI(), elevation))
    hierarchy.getSelectionModel().setSelectedObject(annotation)
    runPlugin('qupath.lib.plugins.objects.RefineAnnotationsPlugin', '{"minFragmentSizeMicrons": 120.0,
"maxHoleSizeMicrons": 120.0}')
    print annotation.getPathClass()
    if ([ "Zone 1", "Zone 2", "Thrombus" ].contains(annotation.getPathClass().name) ) {
        runPlugin('qupath.imagej.detect.cells.PositiveCellDetection', '{"detectionImageBrightfield": "Hematoxylin OD",
"requestedPixelSizeMicrons": 0.5, "backgroundRadiusMicrons": 8.0, "medianRadiusMicrons": 0.0, "sigmaMicrons":
1.0, "minAreaMicrons": 8.0, "maxAreaMicrons": 100.0, "threshold": 0.1, "maxBackground": 2.0,
"watershedPostProcess": true, "excludeDAB": false, "cellExpansionMicrons": 2.0, "includeNuclei": true,
"smoothBoundaries": true, "makeMeasurements": true, "thresholdCompartment": "Cytoplasm: DAB OD mean",
"thresholdPositive1": 0.2, "thresholdPositive2": 0.4, "thresholdPositive3": 0.6000000000000001, "singleThreshold":
true}');
    }
}
// Save the detections before exporting

```

```
getProjectEntry().saveImageData(imageData)
```

```
// Create the measurementExporter and start the export
```

```
def exporter = new MeasurementExporter()
```

```
    .imageList(imagesToExport)          // Images from which measurements will be exported
```

```
    .separator(separator)              // Character that separates values
```

```
    .includeOnlyColumns(columnsToInclude) // Columns are case-sensitive
```

```
    .exportType(exportType)            // Type of objects to export
```

```
    .exportMeasurements(outputFile)     // Start the export process
```

```
print "Done!"
```