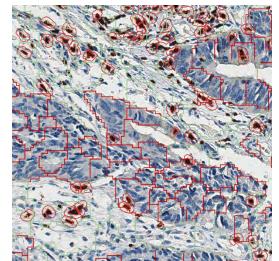


Jan 18, 2025 Version 2

## Quantitative whole-slide image analysis of CD3 and CD8 immunohistochemical stains V.2

DOI

[dx.doi.org/10.17504/protocols.io.yqvfvw6](https://dx.doi.org/10.17504/protocols.io.yqvfvw6)



Seungyeon Yoo<sup>1</sup>

<sup>1</sup>Department of Pathology, Seoul National University Hospital, Seoul National University College of Medicine, Republic of Korea



Seungyeon Yoo

OPEN  ACCESS



DOI: [dx.doi.org/10.17504/protocols.io.yqvfvw6](https://dx.doi.org/10.17504/protocols.io.yqvfvw6)

**Protocol Citation:** Seungyeon Yoo 2025. Quantitative whole-slide image analysis of CD3 and CD8 immunohistochemical stains.

**protocols.io** <https://dx.doi.org/10.17504/protocols.io.yqvfvw6>

**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:** March 01, 2019

**Last Modified:** January 18, 2025

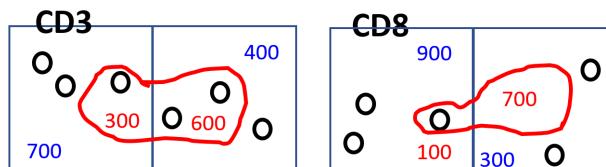
**Protocol Integer ID:** 20981

## Abstract

Given the virtual slide images of CD3 and CD8 immunohistochemical stains, this protocol provides the following quantitative measures of tumor immune microenvironment:

- Density of intraepithelial/stromal tumor-infiltrating lymphocytes (TIL)
- Proportion of stroma within an annotated tumor area
- Ratio of intraepithelial and stromal TIL
- Ratio of CD8 and CD3 TIL

Once you annotate the tumor area manually, it segments the area into 1000 x 1000-pixel tiles. This enables you to assess not only the **overall** density of TIL and stroma, but also their mean, minimum, median, maximum and variance over the tiles.



- **Overall density of TIL (/mm<sup>2</sup>)**

	CD3	CD8
Intraepithelial TIL (ieTIL)	(1+2) (0.3+0.6)	(1+0) (0.1+0.7)
Stromal TIL (strTIL)	(2+1) (0.7+0.4)	(2+2) (0.9+0.3)

- **Overall density of stroma**

$$= \text{Mean}(1100/2000, 1200/2000)$$



**Mean, Min, Median, Max**

: representative value

**Coefficient of variation**

**Quartile coefficient of dispersion**

: degree of intratumoral heterogeneity

**Ratio of ieTIL to strTIL**

**Ratio of CD8 to CD3**

: degree of effective immunity

**TIL count at tumor center and invasive margin**

## Materials

### 1. Computer (of course)

This protocol has been tested on the machines with the following specification:

- Windows 10, AMD Ryzen 7 1700 3.00 GHz (RAM 32GB)
- Windows 7, Intel i7-4790 3.60 GHz (RAM 32GB)
- Ubuntu 14.04 LTS, Intel Xeon 3.40 GHz (RAM 32GB)

In one word: no supercomputer needed. I have nothing to say about Mac machines since I have never used them.

### 2. Software

#### ▪ Python 2.7 (NOT 3.X)

As a windows user, I used Anaconda (<https://www.anaconda.com/download/>) because I needed numpy.. you might try other options here (<https://scipy.org/install.html>)

#### ▪ R

'**caret**', '**readr**' and '**randomForest**' packages have to be installed in advance and **the directory where Rscript.exe resides** has to be included in the PATH system variable. For those who are not sure what you have to do, please google with keywords such as 'how to install a package in R' or 'how to add a new folder to system path'. The directory where Rscript.exe resides in my computer was C:\Program Files\R\R-3.5.2\bin. Yours can be different, so please check it.

#### ▪ QuPath ([Link](#))

It seems that the availability of a machine totally depends on the operability of QuPath; I discovered that QuPath is not as stable on Linux as it is on Windows. In fact, I failed to run this algorithm on a CentOS linux 7 supercomputer. It would be helpful to check the googlegroup of QuPath users ([Link](#)) when you have troubles with QuPath.

### 3. Your own virtual slides

For each patient case, a representative tumor block was selected, immunohistochemistry of CD3 and CD8 was performed and the slides were scanned by **Aperio AT2** slide scanner at **20x** magnification. Why not 40x? just storage issues. I've also tried some images scanned at 40x magnification but had no problem.

### 4. Codes + a

Check [here](#)

You need to create a working directory, create four subdirectories and distribute files as follows:

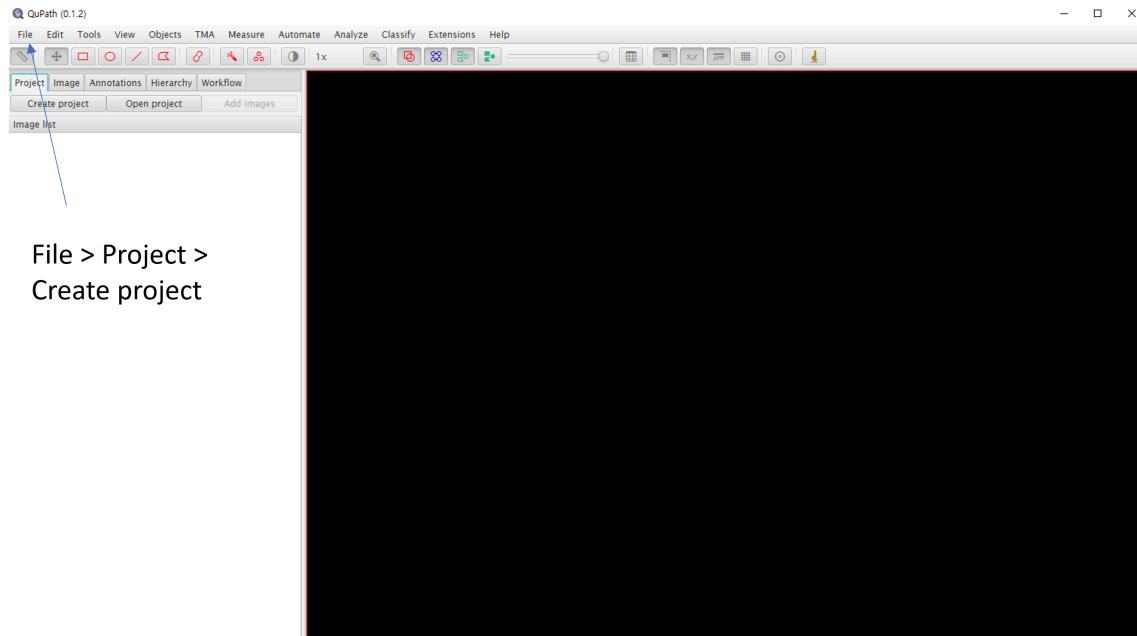
C &gt; HDD (G:) &gt; Work

이름	수정한 날짜	유형	크기
code Put the codes here	2018-12-30 오전 10:...	파일 폴더	
Data ← At this point, just leave them	2018-12-30 오전 10:...	파일 폴더	
Project	2018-12-30 오전 10:...	파일 폴더	
SVS Put the virtual slides here	2018-12-30 오전 8:44	파일 폴더	

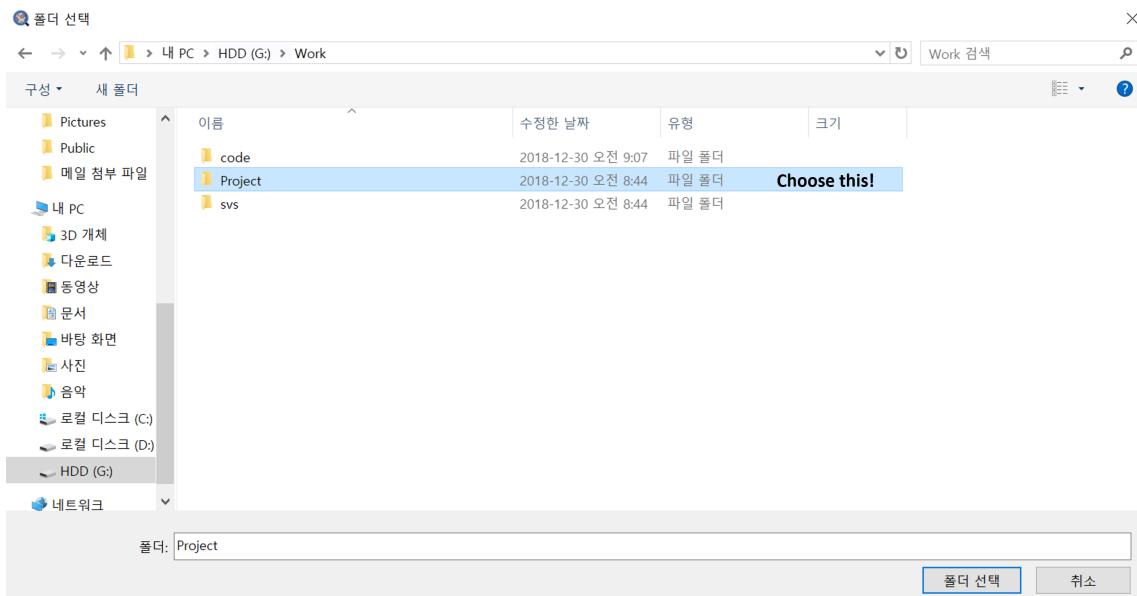


## Create a new project

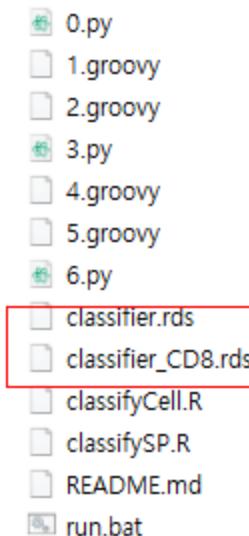
1



File > Project >  
Create project



After creating the project, open 'code' folder and execute 'run.bat' by double-clicking. It will automatically create the script files according to your directory setting.



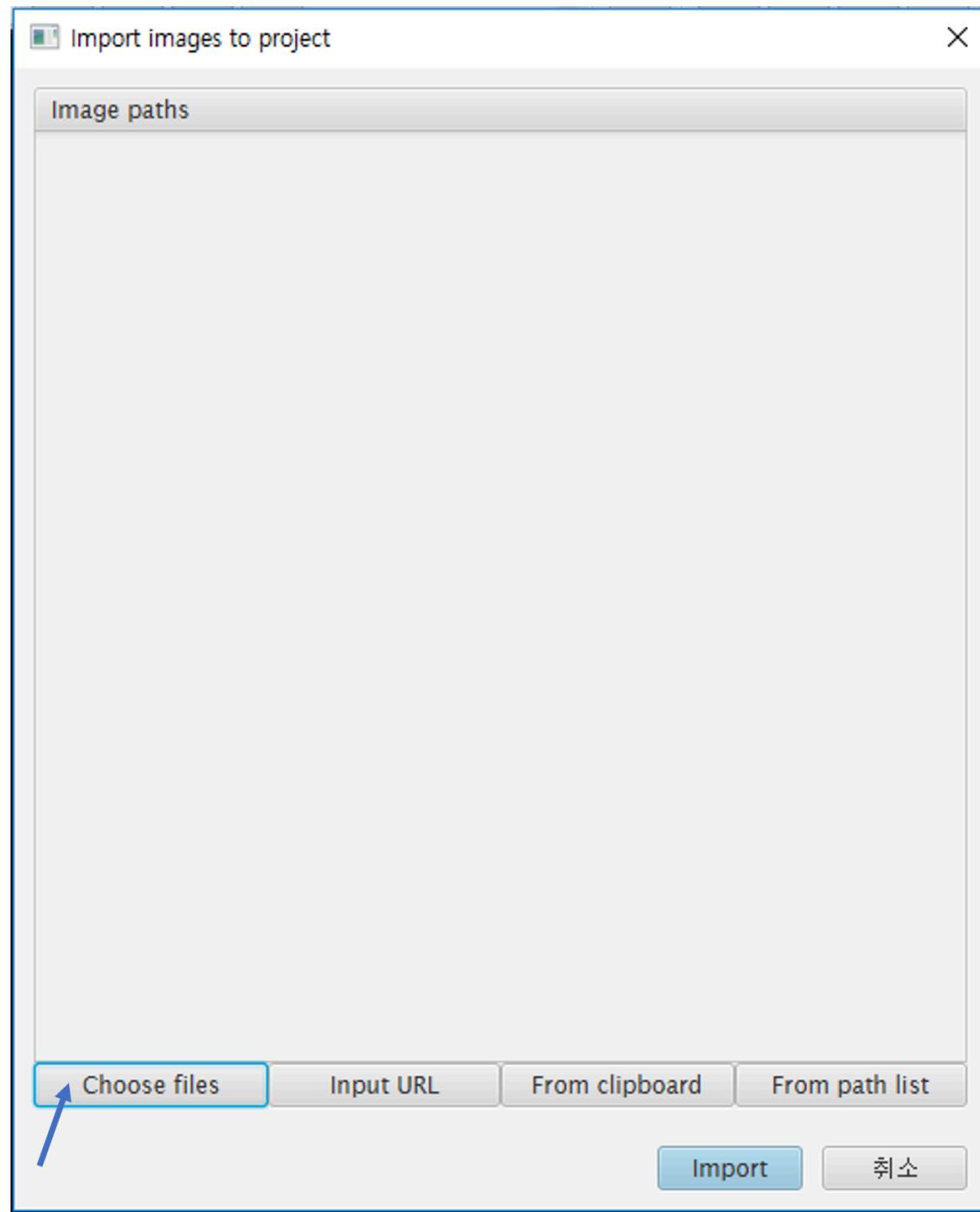
These will appear by unzipping .zip files (I had to segment them due to upload size limit of GitHub)

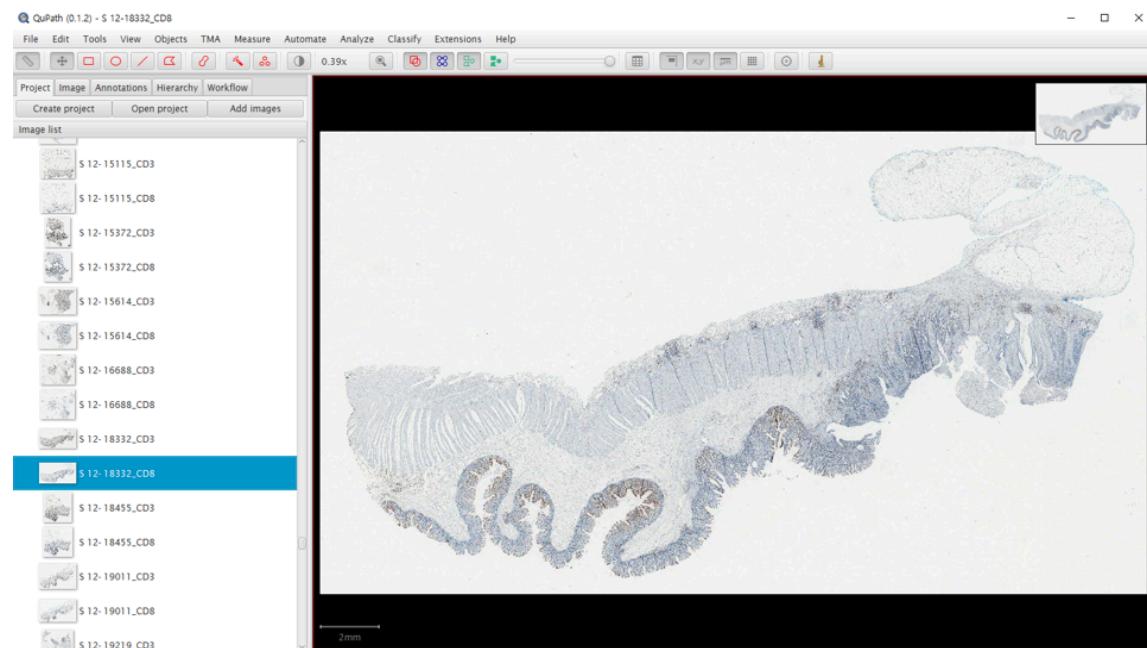
This script do the following substitution and move all the above files to Project/Scripts (this /Scripts subfolder is generated by creating a new project): for non-windows users who might not be able to run .bat file, you can just do that by yourself.

\_REPLACE\_DIR\_ → <Your work folder>/Data/ (i'll call it 'data folder' from now on)  
\_REPLACE\_PROJECT\_DIR\_ → <Your work folder>/Project/ (i'll call it 'project folded' from now on)

## Populate the project with your virtual slides

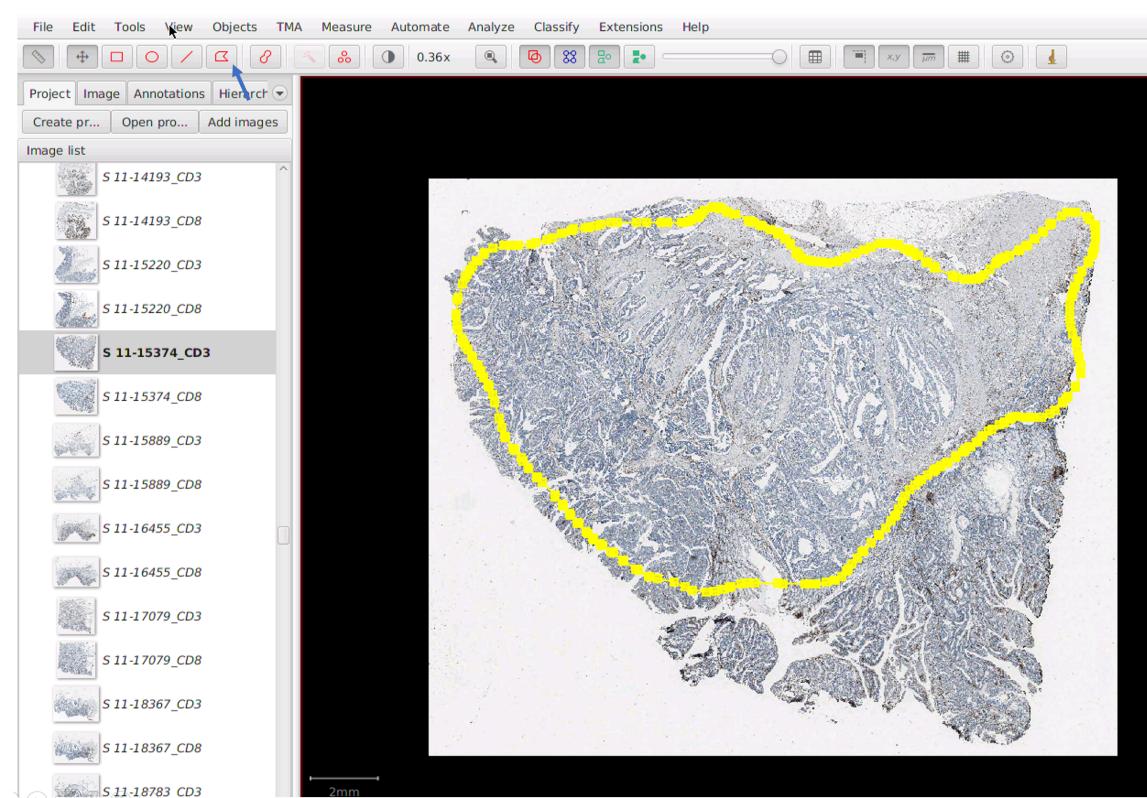
- 2 File > Project > add images





## Annotation of the tumor area

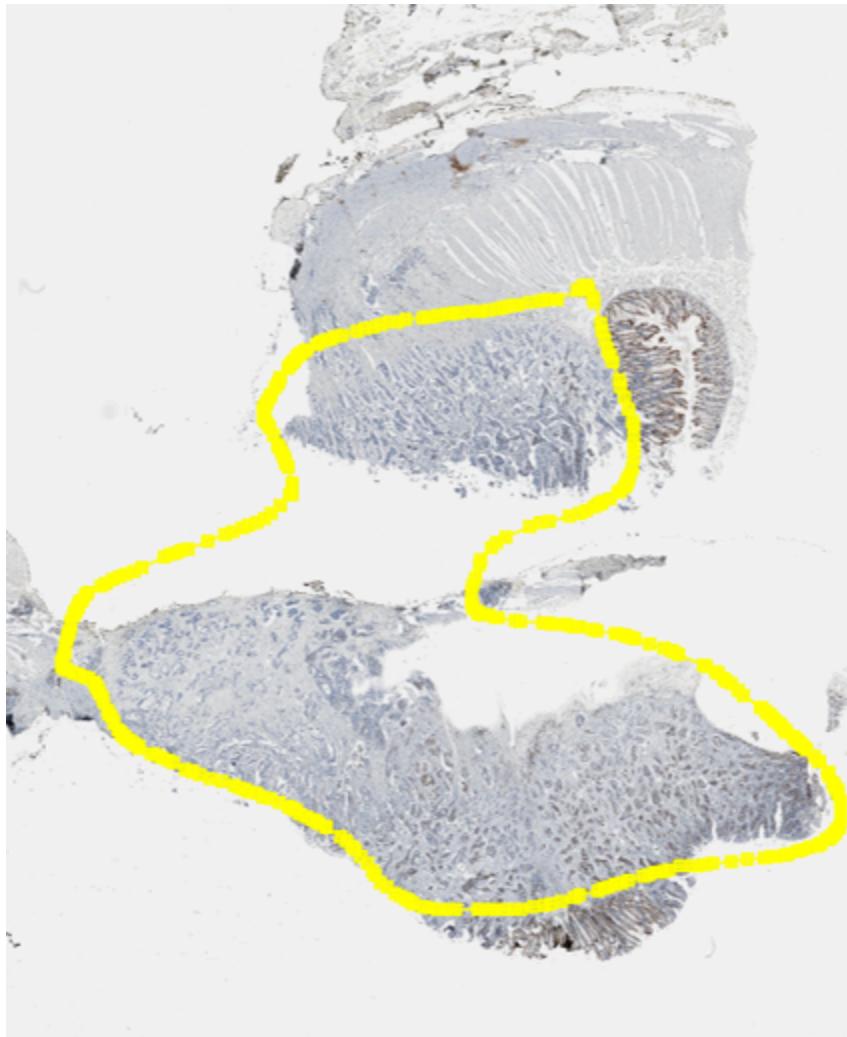
3



After selecting the polygon tool, click one point and drag along the contour of the tumor area while not releasing the button. When the contour is finished, double-click the finishing point.

I don't think there is a general rule of tumor area selection, but I tried to keep the following rules:

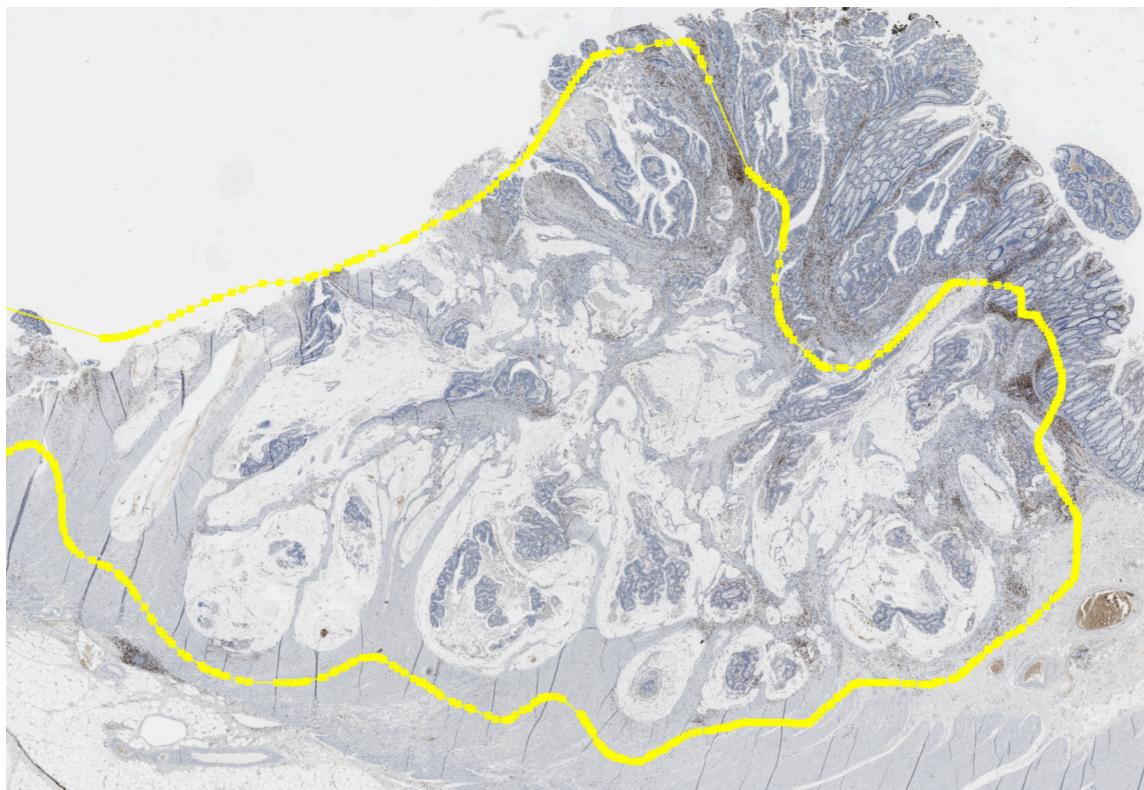
- I tried to have one uninterrupted field of tumor area per each image, but there were some cases with scanty tumor which inevitably forced me to select several separated pieces.



- I tried to exclude the adenomatous area, but there were some ambiguous cases.



- In the case of mucinous carcinoma, the area with acellular mucin pool was also included.
- The area of normal mucosa, necrotic debris and artifacts such as tissue folding, speckles were avoided.
- Normal lymphoid structures such as peritumoral lymph node and mucosa-associated lymphoid tissue near the mucosa were avoided.



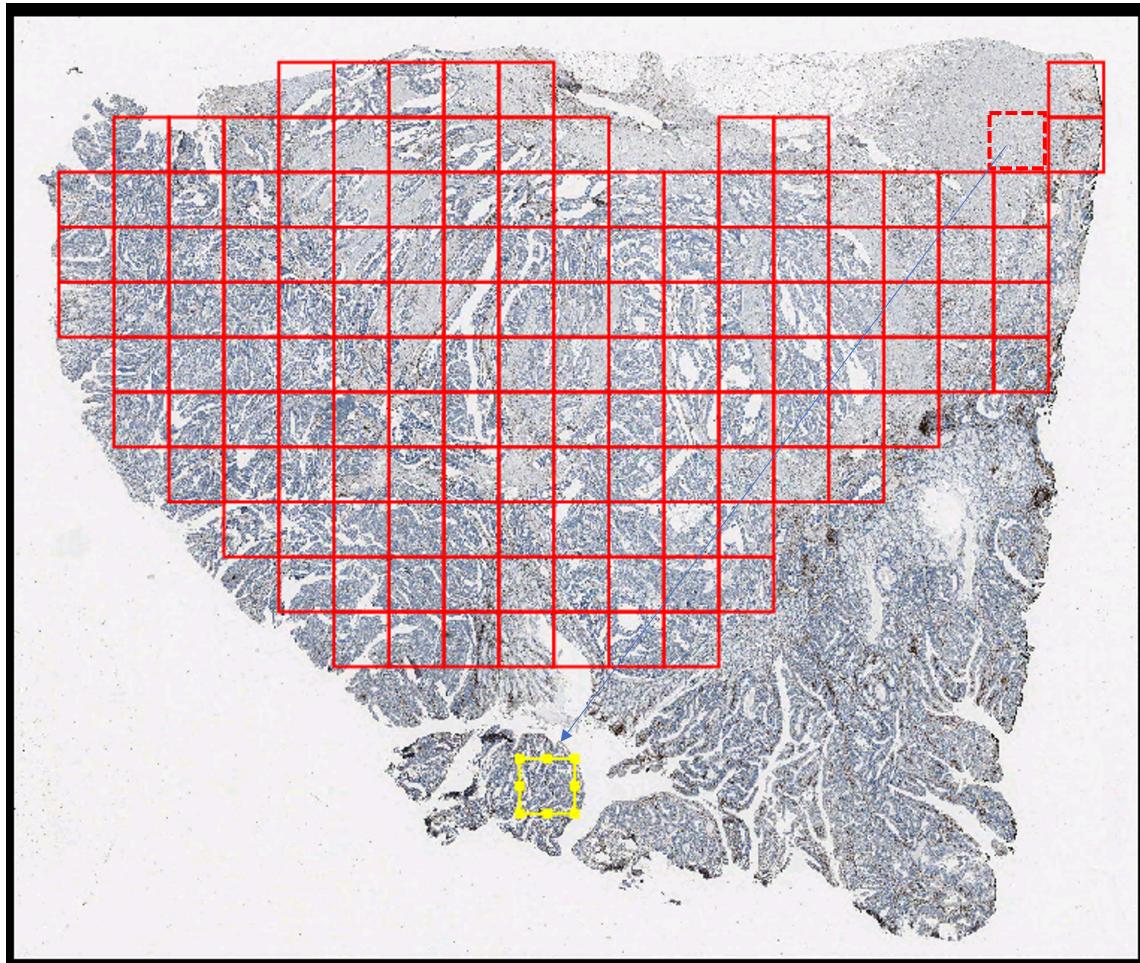
You don't need to be too precise at this point, as tumor area can be more precisely defined in the following step.

## Segmentation

- 4     Analyze > region identification > create tiles

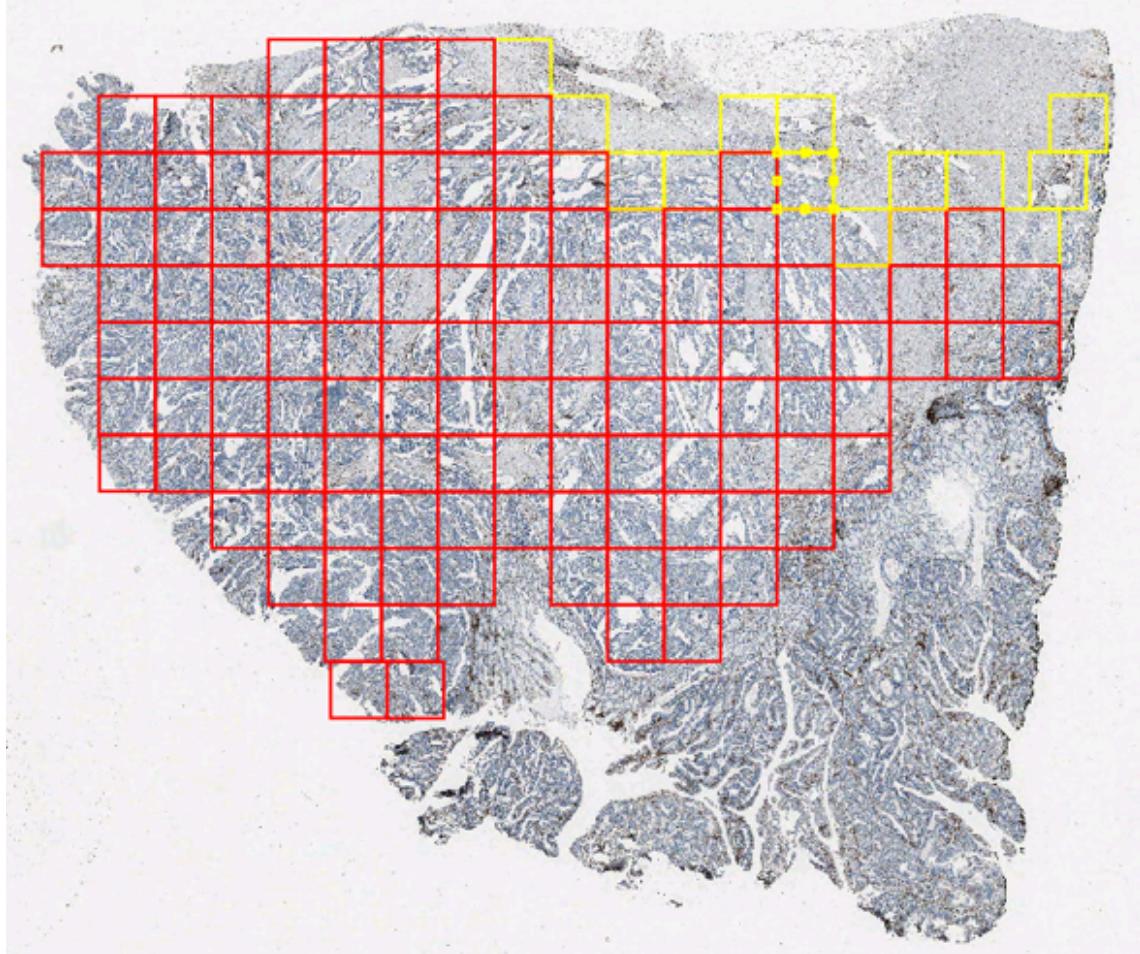


Once the tiles are created, it is free to delete or move them to elaborate the tumor area.



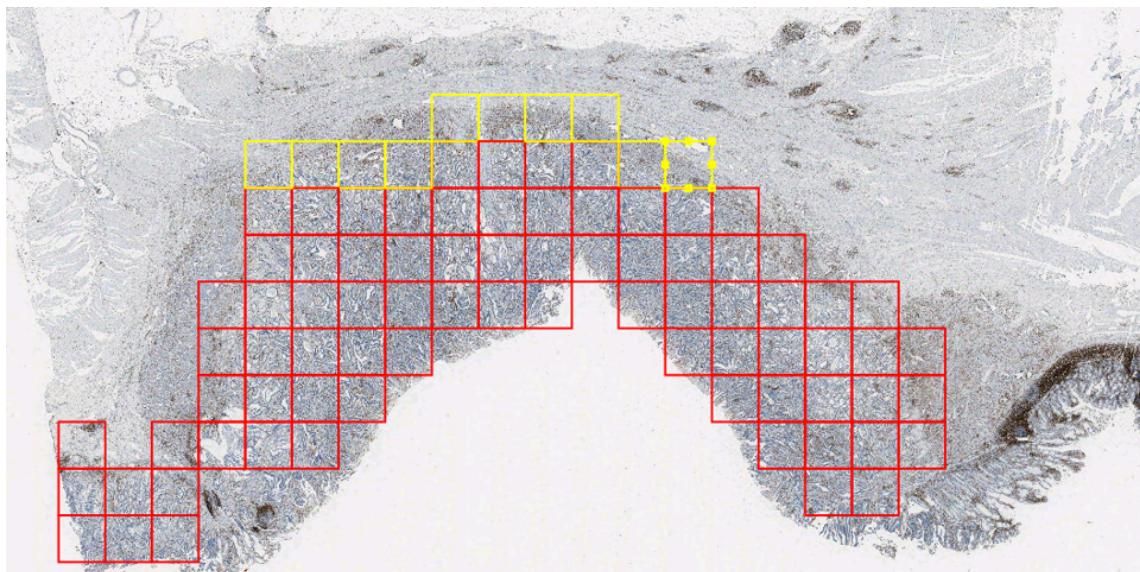
## Designate the invasive margin (IM)

- 5 While pushing the control key (Ctrl), click the tiles that constitute the IM.

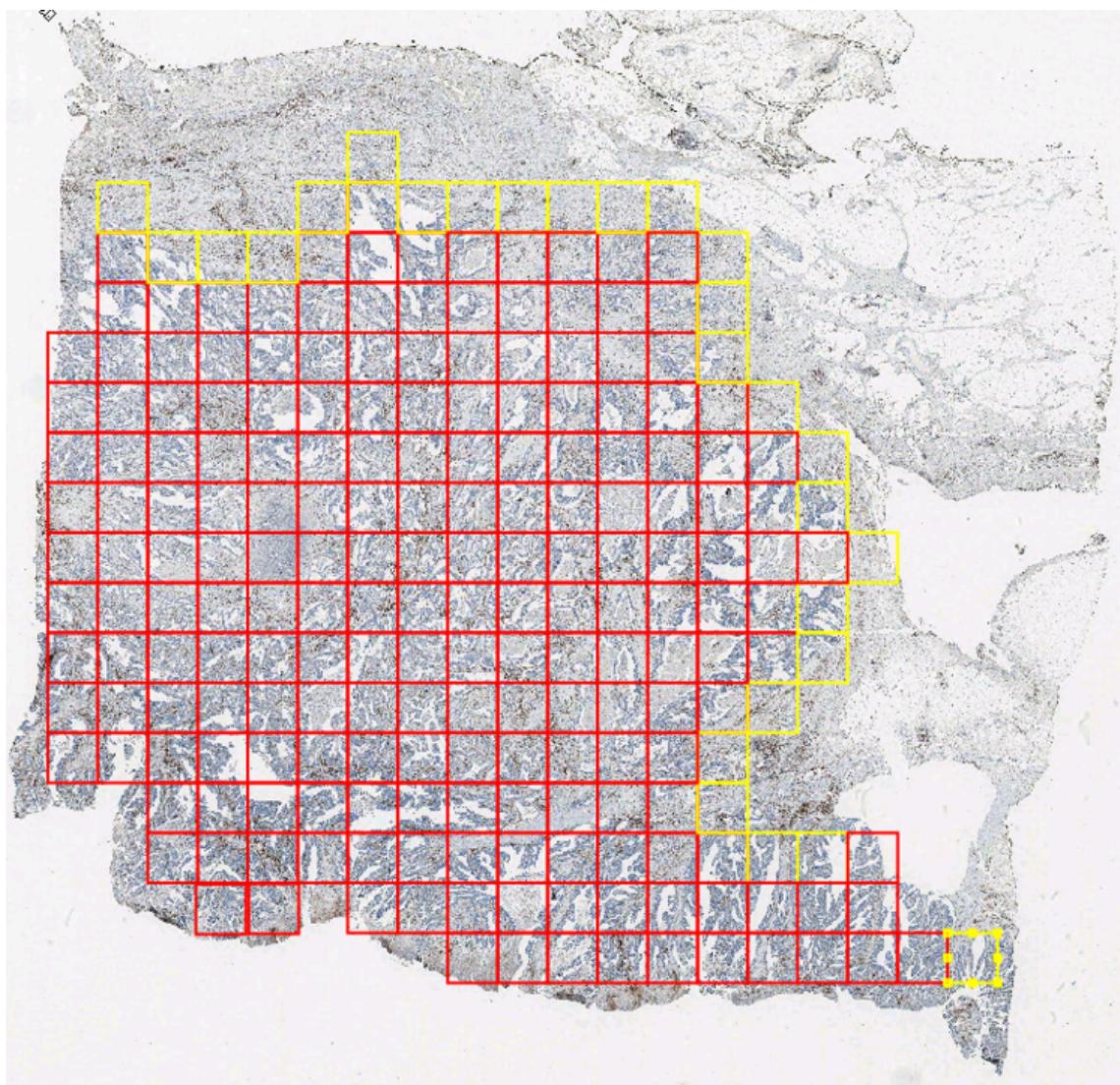


Again, I don't think there is no general definition of the IM, but I tried to keep the following rules:

- When the mucosa-to-serosa axis is evident, the IM is defined as the "frontmost" files. In other words, the tiles approaching the mucosa are not considered as IM.



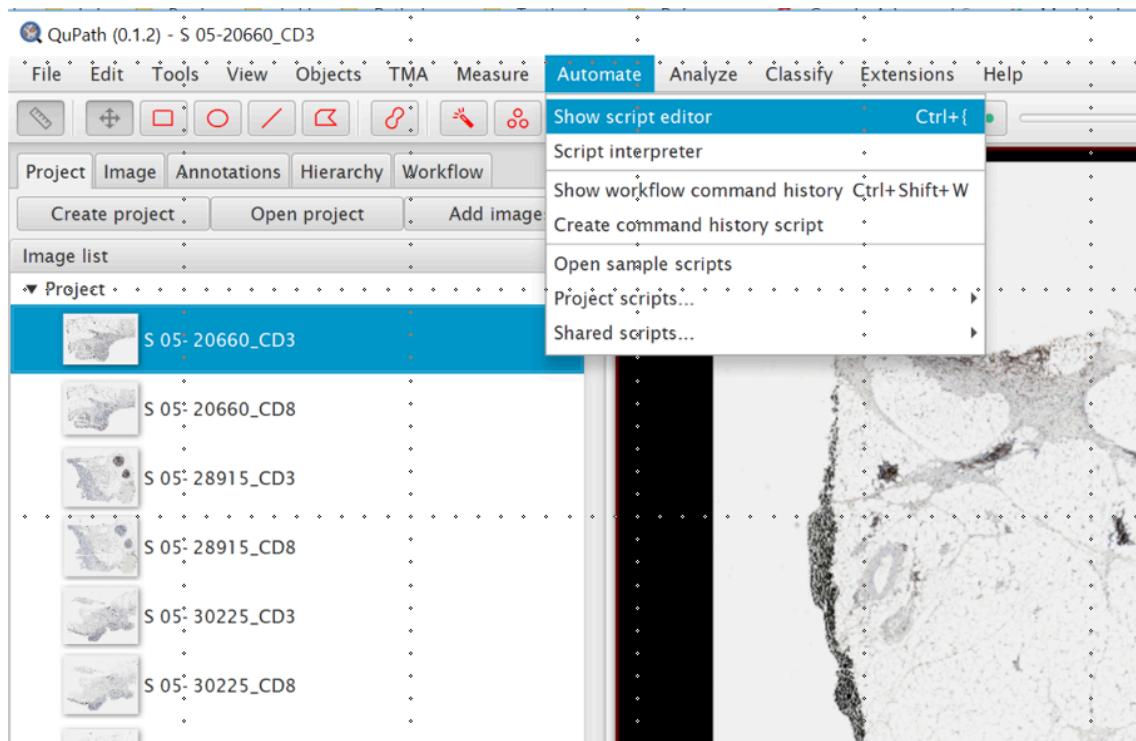
- When the axis is not evident, any border with the non-neoplastic tissue was considered as IM.



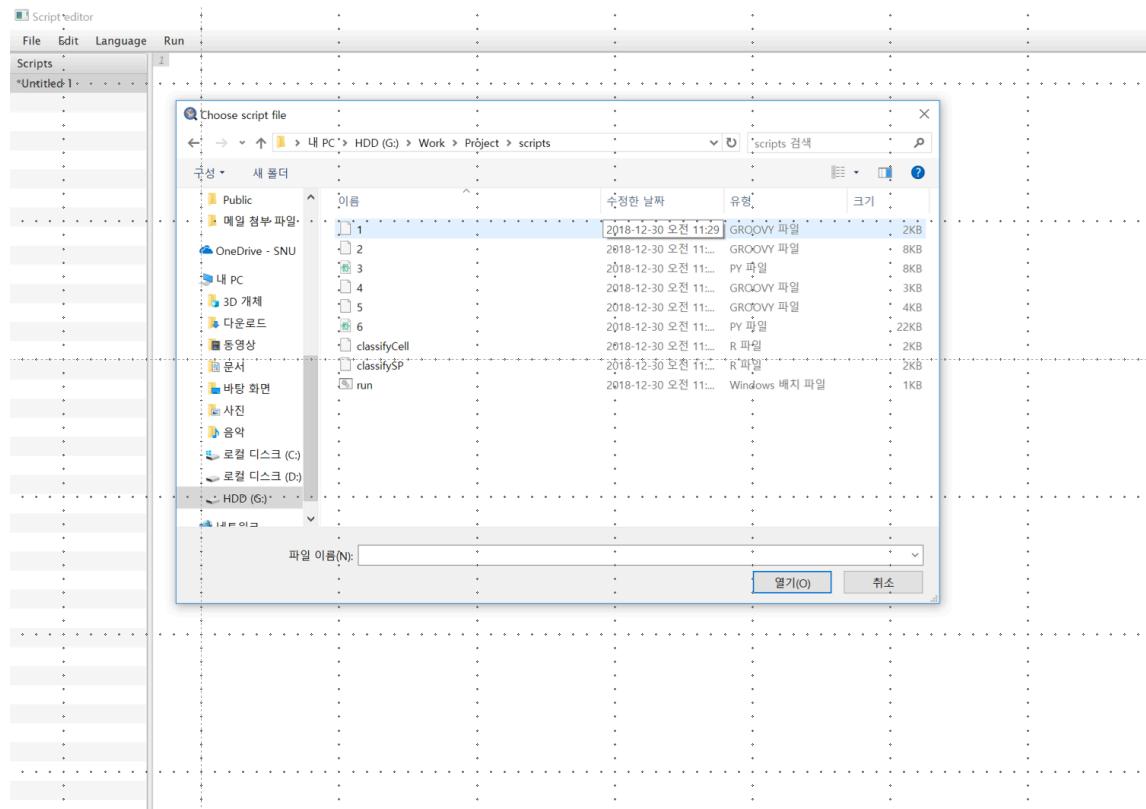
- The area bordering with abscess, necrosis or lymph node is NOT invasive margin.
- The IM has to contain tumor cells. As drawing tumor area in step 3 produces somewhat crude outline, some tiles at the border of tumor and normal tissue might not contain tumor cells. In that case, such tiles need to be deleted.

## Run 1.groovy

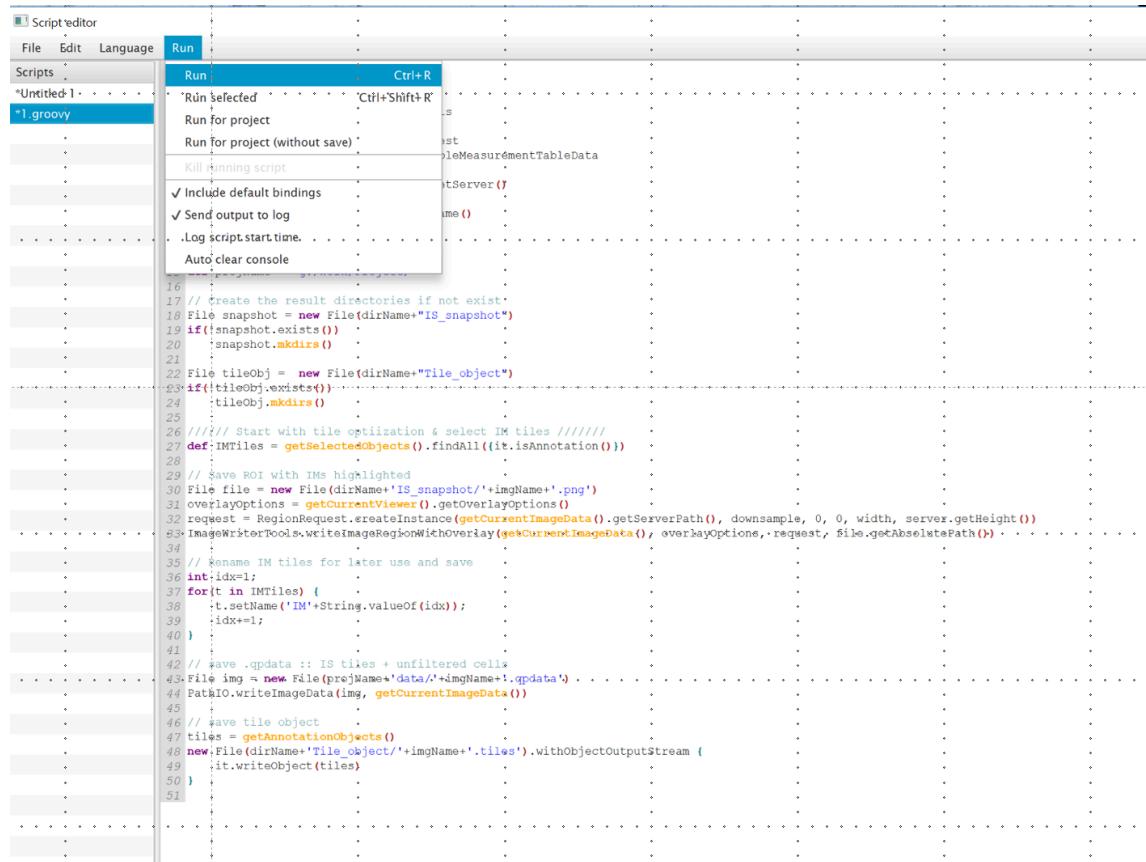
- 6 This script saves the screenshot of tiles with IM highlighted (just like the one you've just seen in the previous step) and saves tile objects for later use.



Select Automate > Show script editor, then you'll see a new window.



Select File > Open, go to your project folder's script folder then you'll find script files. Select 1.groovy and open it.



The screenshot shows a 'Script editor' window with a context menu open over a script file named '1.groovy'. The 'Run' option is highlighted in blue, indicating it is selected. Other options in the menu include 'Run selected', 'Run for project', 'Run for project (without save)', 'Kill running script', 'Include default bindings', 'Send output to log', 'Log script start time', and 'Auto-clear console'. The script code itself is visible below the menu, starting with some imports and comments about creating result directories.

```
16
17 // create the result directories if not exist
18 file snapshot = new File(dirName+"IS_snapshot")
19 if(!snapshot.exists())
20     snapshot.mkdirs()
21
22 File tileObj = new File(dirName+"Tile_object")
23 if(!tileObj.exists())
24     tileObj.mkdirs()
25
26 // Start with tile optimization & select IM tiles //////
27 def IMTiles = getSelectedObjects().findAll({it.isAnnotation()})
28
29 // Save ROI with IMs highlighted
30 File file = new File(dirName+"IS_snapshot"+imgName+".png")
31 overlayOptions = getCurrentViewer().getOverlayOptions()
32 request = RegionRequest.createInstance(getCurrentImageData().getServerPath(), downsample, 0, 0, width, server.getHeight())
33 ImageWriterTools.writeImageRegionWithOverlay(getCurrentImageData(), overlayOptions, request, file.getAbsolutePath())
34
35 // Rename IM tiles for later use and save
36 int idx=1;
37 for(t in IMTiles) {
38     t.setName('IM'+String.valueOf(idx));
39     idx++;
40 }
41
42 // Save .gpdata :: IS tiles + unfiltered cells
43 File img = new File(preName+'data/'+imgName+'.gpdata')
44 PathIO.writeImageData(img, getCurrentImageData())
45
46 // Save tile object
47 tiles = getAnnotationObjects()
48 new File(dirName+"Tile_object/"+imgName+".tiles").withObjectOutputStream {
49     it.writeObject(tiles)
50 }
```

To run the script, select Run > Run or just push Ctrl+R.

## Repeat step 3 through 6 for all the slide files

- 7 The annotation of tumor area is the only step human intervention is needed. All the other steps can be done automatically. I thought it was better for me to stay in front of the computer only when I was needed and let the computer do all the other jobs by itself while I was away, since the other jobs would be time-consuming. That was the reason I separated 1.groovy from others.

So suppose you started with  $N$  slide files. If you have successfully finished the step 7, your data folder would have the two subfolders as follows, each of them containing  $N$  files.

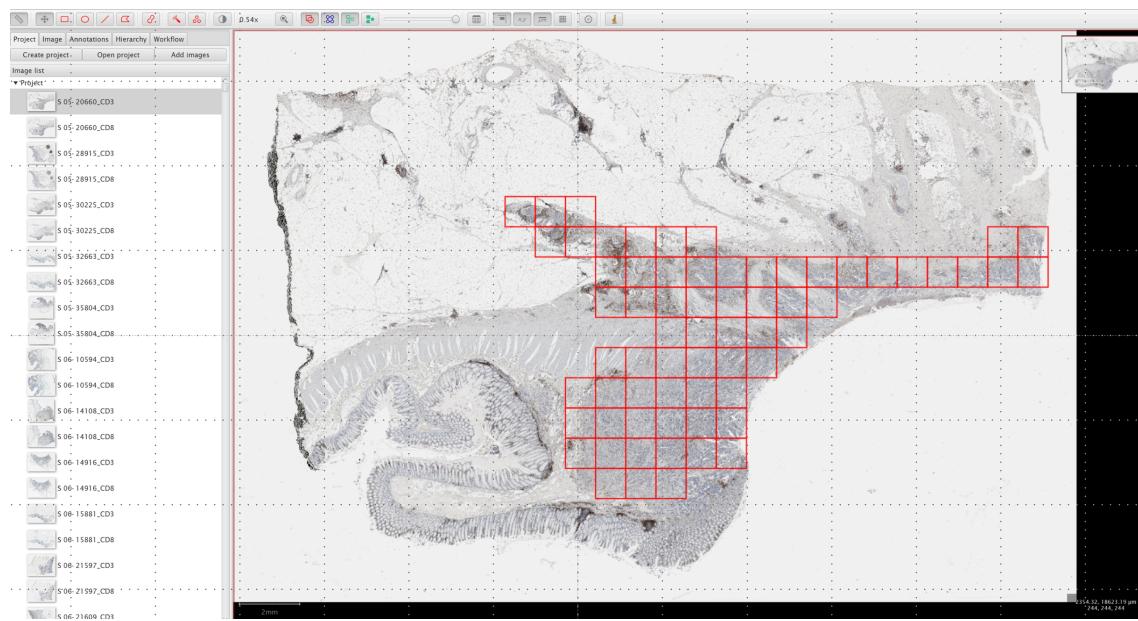
보기	PC > HDD (G) > Work > Data >	수정한 날짜	유형	크기
이름	IS_snapshot	2019-02-06 오후 2:27	파일 폴더	
	Tile_object	2019-02-06 오후 2:28	파일 폴더	

and your project folder's subfolder named [data] would also have N qpdata files.

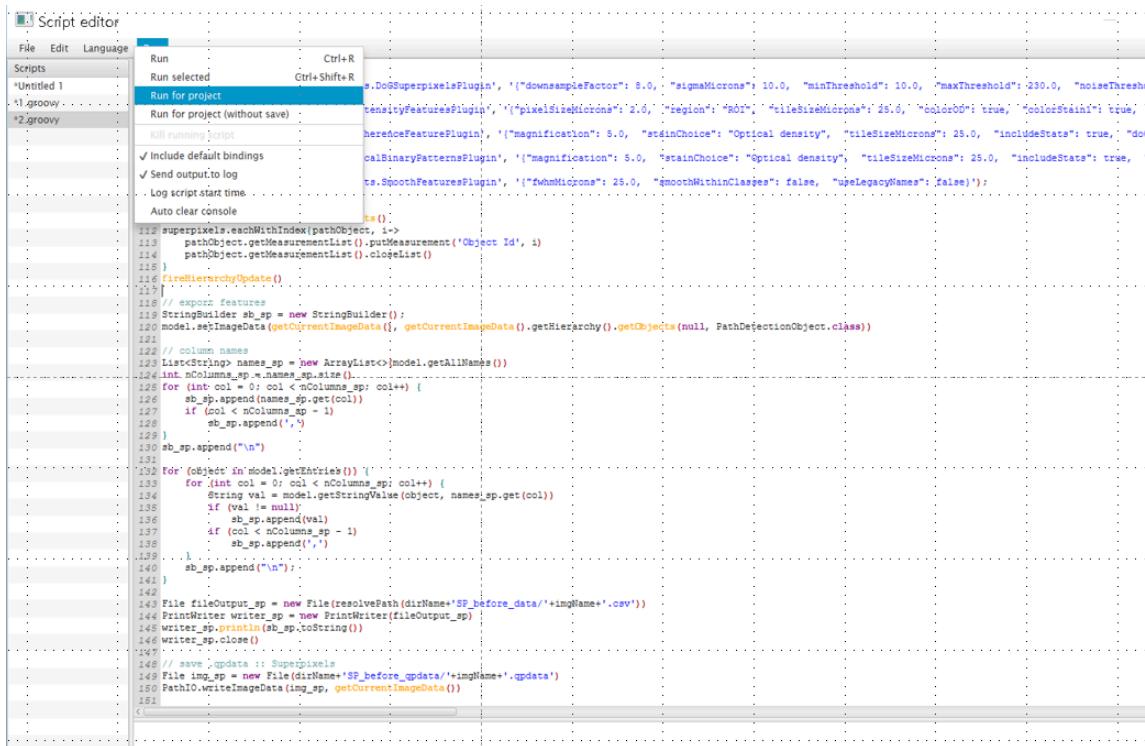
 S 07-1837_CD3	2018-12-22 오후 11:19	QuPath data file
 S 07-1837_CD8	2018-12-22 오후 11:21	QuPath data file
 S 07-2171_CD3	2018-12-22 오후 11:34	QuPath data file
 S 07-2171_CD8	2018-12-22 오후 11:36	QuPath data file
 S 07-2694_CD3	2018-12-23 오전 12:41	QuPath data file
 S 07-2694_CD8	2018-12-23 오전 12:43	QuPath data file
 S 07-3030_CD3	2018-12-23 오전 1:04	QuPath data file
 S 07-3030_CD8	2018-12-23 오전 1:09	QuPath data file
 S 07-3139_CD3	2018-12-23 오전 1:14	QuPath data file
 S 07-3139_CD8	2018-12-23 오전 1:16	QuPath data file
 S 07-3199_CD3	2018-12-23 오전 1:22	QuPath data file
 S 07-3199_CD8	2018-12-23 오전 1:25	QuPath data file
 S 07-4446_CD3	2018-12-23 오전 1:31	QuPath data file
 S 07-4446_CD8	2018-12-23 오전 1:34	QuPath data file
 S 07-5395_CD3	2018-12-23 오전 1:44	QuPath data file
 S 07-5395_CD8	2018-12-23 오전 1:54	QuPath data file
 S 07-7066_CD3	2018-12-23 오전 1:56	QuPath data file
 S 07-7066_CD8	2018-12-23 오전 1:58	QuPath data file
 S 07-8427_CD3	2018-12-23 오전 2:02	QuPath data file
 S 07-8427_CD8	2018-12-23 오전 2:06	QuPath data file
 S 07-8557_CD3	2018-12-23 오전 2:15	QuPath data file
 S 07-8557_CD8	2018-12-23 오전 2:24	QuPath data file
 S 07-9198_CD3	2018-12-23 오전 2:32	QuPath data file
 S 07-9198_CD8	2018-12-23 오전 2:41	QuPath data file
 S 07-10940_CD3	2018-12-22 오후 10:59	QuPath data file
 S 07-10940_CD8	2018-12-22 오후 11:03	QuPath data file
 S 07-12349_CD3	2018-12-22 오후 11:04	QuPath data file
 S 07-12349_CD8	2018-12-22 오후 11:04	QuPath data file
 S 07-12631_CD3	2018-12-22 오후 11:09	QuPath data file
 S 07-12631_CD8	2018-12-22 오후 11:13	QuPath data file
 S 07-15744_CD3	2018-12-22 오후 11:15	QuPath data file
 S 07-15744_CD8	2018-12-22 오후 11:16	QuPath data file
 S 07-19568_CD3	2018-12-22 오후 11:25	QuPath data file
 S 07-19568_CD8	2018-12-22 오후 11:29	QuPath data file
 S 07-20836_CD3	2018-12-22 오후 11:31	QuPath data file
 S 07-20836_CD8	2018-12-22 오후 11:33	QuPath data file
 S 07-22041_CD3	2018-12-22 오후 11:38	QuPath data file

## Run 2.groovy

- 8 Check any image using project image list and make sure you see a image with tiles overlaid, like the one below. Since this is what qpdata file stored at /project/data does, any image should be seen with tiles overlaid if you have proper N qpdata files generated.



Then go to the script editor and open 2.groovy. Since N files are ready, you'd want the computer to process all the files by itself. To do so,



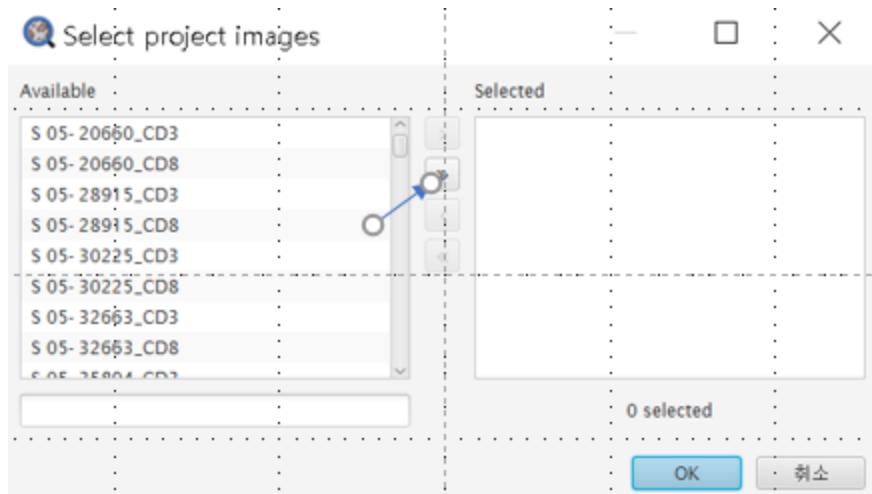
```

File Edit Language
Scripts
*Untitled 1
+1.groovy
+2.groovy

Run selected Ctrl+Shift+R
Run for project Ctrl+R
Run for project (without save)
Kill running script
✓ Include default bindings
✓ Send output to log
Log script start time
Auto clear console
  *()
1.1 superpixels.eachWithIndex(pathObject, i->
1.2   pathObject.getMeasurementList().putMeasurement('Object Id', i)
1.3   pathObject.getMeasurementList().closeList()
1.4 )
1.5 )
1.6 fireHierarchyUpdate()
1.7 // export features
1.8 StringBuilder sb_sp = new StringBuilder();
1.9 model.setImageData(getCurrentImageData(), getCurrentImageData().getHierarchy().getObjects(null, PathDefinitionObject.class))
1.10
1.11 // column names
1.12 List<String> names_sp = new ArrayList<>(model.getAllNames())
1.13 int nColumns_sp = names_sp.size()
1.14 for (int col = 0; col < nColumns_sp; col++) {
1.15   sb_sp.append(names_sp.get(col))
1.16   if (col < nColumns_sp - 1)
1.17     sb_sp.append(',')
1.18 }
1.19 sb_sp.append("\n")
1.20
1.21 Tdr'(object: In[model.getEntries()]) {
1.22   for (int col = 0; col < nColumns_sp; col++) {
1.23     String val = model.getStringValue(object, names_sp.get(col))
1.24     if (val != null)
1.25       sb_sp.append(val)
1.26     if (col < nColumns_sp - 1)
1.27       sb_sp.append(',')
1.28   }
1.29   sb_sp.append("\n");
1.30 }
1.31
1.32 File fileOutput_sp = new File(resolvePath(dirName+'SP_before_data/'+imgName+'.csv'))
1.33 PrintWriter writer_sp = new PrintWriter(fileOutput_sp)
1.34 writer_sp.println(sb_sp.toString())
1.35 writer_sp.close()
1.36
1.37 // save .pdata :: Superpixels
1.38 File img_sp = new File(dirName+'SP_before_pdata/'+imgName+'.pdata')
1.39 PathIO.writeImageData(img_sp, getCurrentImageData())
1.40
1.41

```

Select Run > Run project (**without save**), then a new window will appear.



Push >> to select all the images in the project to be processed and click [OK].

It will take some time to finish... for example, processing 1158 files took more than one day. The program might seem to be halted, but you will see it's not by checking the data folder. There will be four subfolders (Cell\_before\_data, Cell\_before\_qpdata, SP\_before\_data, SP\_before\_qpdata) created and new output files being generated.

When each of the four subfolders is populated with N files, step 8 is over.

Take time to check contents of SP\_before\_data and SP\_before\_qpdata to see whether there are some output files with oddly small size. It happened because superpixel segmentation sometimes fails for unknown reason. Since there's no superpixel, the output file containing information of superpixels should be small. In my case, the problem disappeared by repeating step 3 through 6 and running 2.groovy only for those files.

## Run 3.py

- 9 In fact, 2.groovy identified putative cells and segmented images into superpixels. Then the next script, 3.py, filters true lymphocytes and classifies superpixels into tumor or stroma. It should be run outside QuPath because I used machine learning classifier implemented by R and used python to process QuPath output files ready to be classified. QuPath does have machine learning classifier, which can be replaced with this step :)

Folded tissue fragments or ink particles can be falsely recognized as lymphocytes, so try to avoid the artifacts as possible. I tried to make classifiers to filter them out, but realized that so many false negatives happen by doing so.

As mentioned in my article, lots of my CD8 images suffered varying degree of nonspecific membranous staining. As a result, I had to apply a lymphocyte classifier for CD8 images to filter out artifacts generated by the nonspecific staining. For CD3 images, no filtering was needed. So my script was designed to classify every cell as "positive" for CD3 stains and to retrieve a classifier for CD8 stains as follows:

```
if(nrow(nuc)!=0) {  
  if(grep('CD3', fName))  
    prediction <- rep(as.factor('Positive'), nrow(nuc)) # No filtering for CD3!  
  else  
    prediction <- predict(CD8_model, nuc)  
  
  write.csv(data.frame(total[,1], prediction), file=paste(resDir, fName, '.csv', sep=''))  
}
```

You might have different kinds of problems on your images. In that case, you might want to create your own classifiers and modify the above part of classifyCell.R to selectively apply the classifier.

After do your modification, run 3.py.

**For the script to run with no problem, two things to be done.**

- 1) the directory of R belongs to the system environmental variable**
- 2) package readr, caret and randomForest have to be installed in your R.**

For Windows users who have installed Anaconda, you can run the script by **executing Anaconda Powershell Prompt** and type as follows:

```
> python <path to the script file>/3.py
```

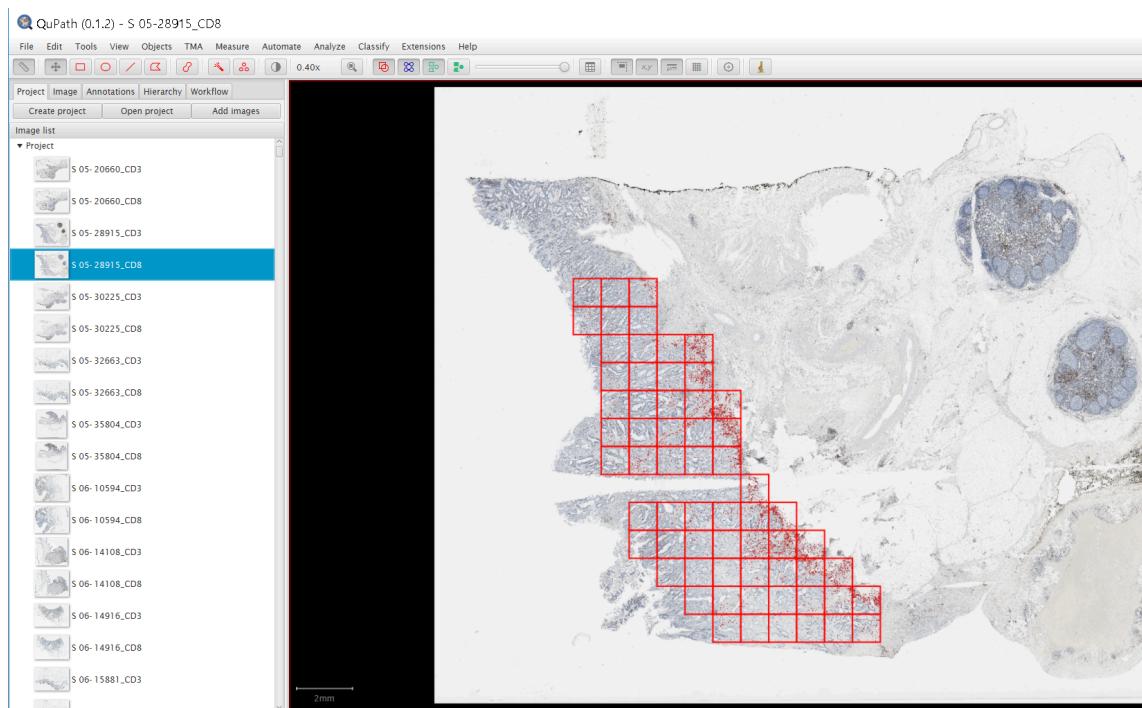


A screenshot of a terminal window titled "Anaconda Powershell Prompt". The command "python g:\work\project\scripts\3.py" is typed into the prompt. The terminal shows the path "C:\Users\Niki Yoo" and the command is highlighted in yellow.

for me, it took several hours (12-24h) to process 1158 files.

## Run 4.groovy

- 10 Go back to QuPath and check any image using project image list. If what you see is an image with tiles and putative cells like the one below, that means everything so far has been fine.

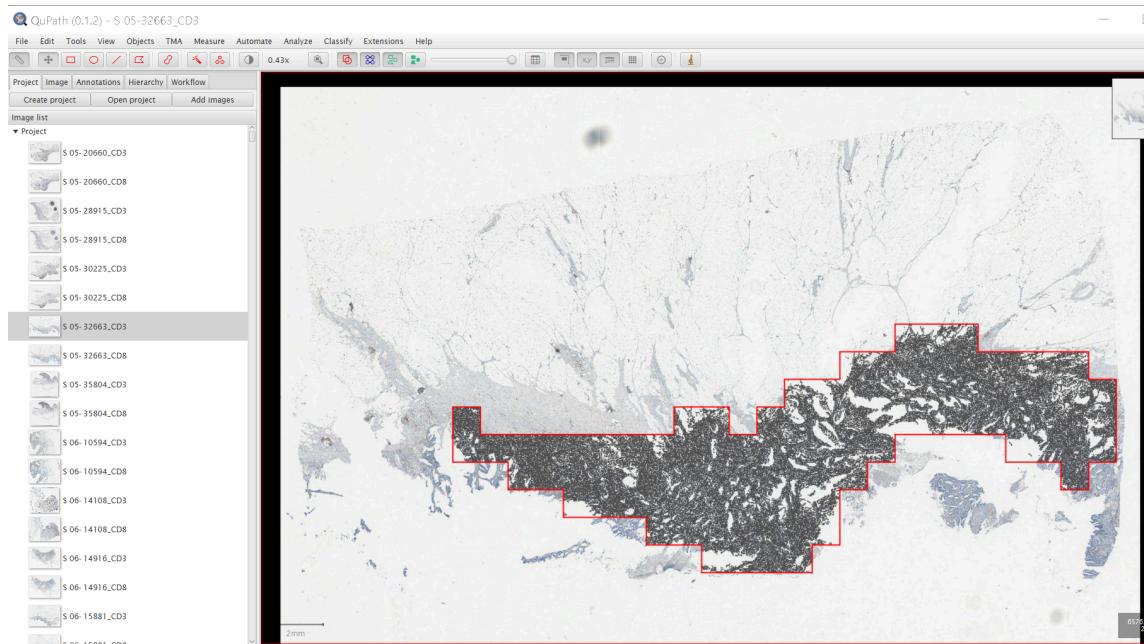


Then open 4.groovy and Run > Run for project (**without save**).

This script applies the result of cell filtering to QuPath.

## Run 5.groovy

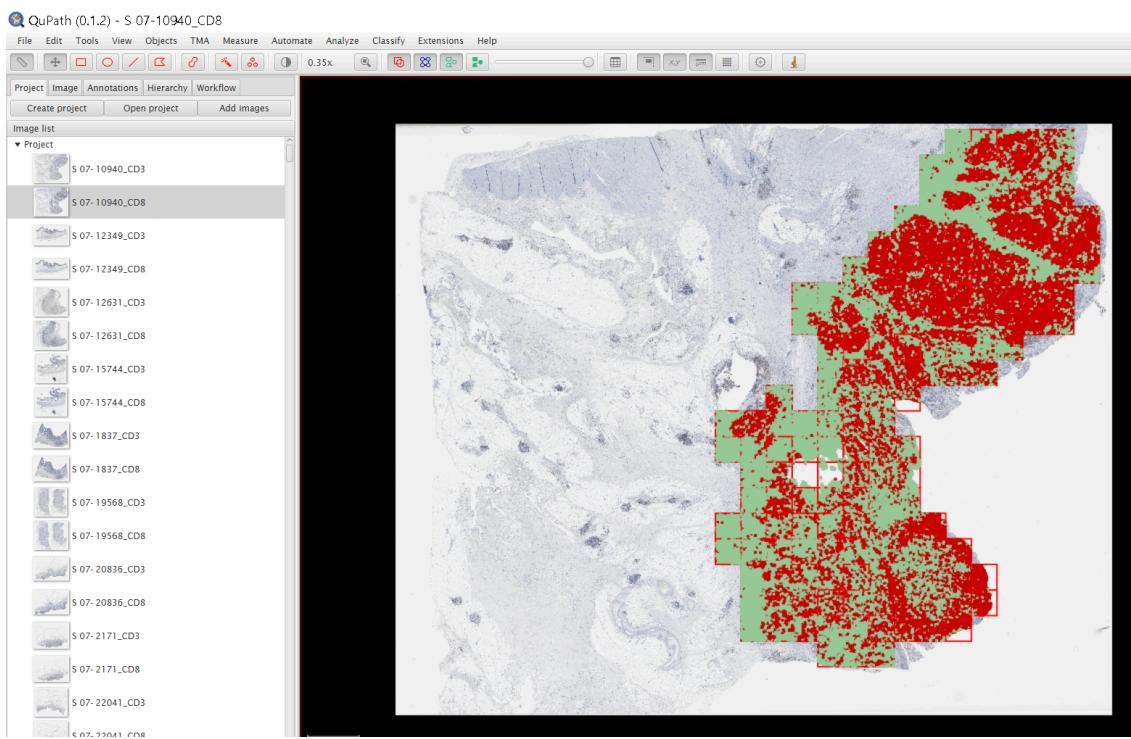
- 11 Check any image using project image list and make sure you see a image like the one below.  
These are superpixels!



Then open 5.groovy and Run > Run for project (**without save**).

This script applies the result of superpixel classification and retrieve the filtered cells to obtain intraepithelial (true lymphocytes in tumor superpixels) and stromal (true lymphocytes in stroma superpixels) TIL.

To see the result after the execution is finished, select File > Open > go to the data folder > select 'ES\_Tilewise\_qpdata' subfolder > select any qpdata file



## Run 6.py

**12 This is the final step. It produces a file named count.csv, which contains values of 207 measures for each case.**

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z		
1	Case Id	CD3_le_de	CD3_str_d	CD3_le_str	CD3_le_de	CD3_le_d	CD3_le_CD3_le																				
2	S 10-1885	58.96181	76.08653	0.774931	63.84064	81.96985	2.71712	0	1.043694	0	49.01749	52.76623	1.019154	46.6593	61.6349	57.72416	0.493345	0.613564	0.4848512	0.926769	2.450804	132.8082	119.1736	1.114409	132.442		
3	S 11-2324	136.9342	323.0564	0.423871	128.5459	324.4104	0.369967	22.94469	63.58527	0.029448	77.7712	268.6894	0.322834	496.0017	951.8046	1.06325	0.524318	0.419293	0.411979	0.965287	0.582236	0.626602	147.17674	210.938	0.351652	72.70691	
4	S 07-2083	52.8455	194.7947	0.271289	56.3462	193.6317	0.271198	0	2428709	0	33.2368	188.0937	0.208714	196.0819	65.9549	0.807015	0.374811	0.510145	0.584844	1.255042	0.680741	0.872317	38.77952	207.4444	0.186939	192.346	
5	S 08-4419	73.1819	277.639	0.263556	95.19225	278.541	0.453232	0	12.37833	0	70.7748	235.5004	0.334579	490.3377	597.2272	3.73009	0.453791	0.33922	0.404734	0.89152	0.492579	1.219181	75.68988	215.9732	0.356014	92.9824	
6	S 11-1277	82.1324	102.5614	61.58763	146.9977	0.485779	0	0	49.33059	112.1847	0	640479	0	74.74694	0	0	0	0	0	0	0	0	0	0	0	0	
7	S 09-2466	181.679	399.5787	0.045461	195.0593	382.0111	0.460236	0	20.63061	0	0	29.9795	354.532	0.343354	173.877	127.1064	0.329867	0.051845	0.89892	0.948977	1.404258	0.595202	0.963573	384.813	380.4135	0.405562	151.650
8	S 09-2467	23.4602	100.29153	0.260859	0	0	0	0	0	0	0	29.9745	0	108.1064	0.257453	139.1625	0.329862	0.897171	0.525387	0.950707	0.070142	0.895124	0.101558	0.10151	0.346816	52.102	
9	S 09-2468	29.29153	158.67678	0.260859	0	0	0	0	0	0	0	50.01446	0	23.9745	108.1064	0.252701	107.042	0.895124	0.525387	0.950707	0.070142	0.895124	0.101558	0.10151	0.346816	52.102	
10	S 08-3891	47.20887	124.426	0.3787413	38.39187	141.435	0.801427	0	0	6.94357	0	26.44708	93.9556	0.291618	23.89564	66.0045	0.89514	0.89796	0.8957	0.491981	1.082529	0.899116	0.651098	42.64486	19.98873	0.357079	40.47673
11	S 07-2171	126.10285	142.47891	0.378987	21.03808	78.86708	0.088351	116.1927	313.3064	0.399412	0	24.0201	567.7207	0.655105	0.40207	0.186169	0.60115	0.862478	0.960911	0.893114	0.95528	66.685393	12.7287	0.591064	52.3517		
12	S 10-1369	154.1558	142.1781	0.0107667	13.08884	140.5977	0.884152	0	10.05597	0	5.233311	124.6918	0.044527	94.2015	414.1961	0.0518132	1	0.474881	1	1.478101	0.446057	1.2399	10.040937	106.3844	1.032411	37.7778	
13	S 08-2567	77.65958	71.5491	1.085402	46.39071	81.20488	0.518829	0	0	0	5.464744	33.3329	0.321324	52.52737	401.3721	0.4515107	1	0.912834	1	1.687734	1.234998	1.37742	57.12356	12.1574	0.509332	39.042	
14	S 08-2120	35.31028	184.0753	0.0212429	41.71483	173.742	0.279081	0	13.98505	0	28.74091	145.4093	0.195153	120.3049	682.8874	1.357219	0.790785	0.392534	0.0713185	1.143269	0.664935	1.020654	57.24238	160.3211	0.357048	56.9068	
15	S 11-3184	121.1308	169.59441	0.0715411	87.40359	104.9697	0.374285	0	1.667997	0	21.56665	120.674	0.357857	666.0727	821.2242	2.226725	0	1.0803187	1	1.470999	1.055098	0.111116	47.74688	78.8461	0.610694	44.0931	
16	S 11-2550	157.5937	337.0271	0.647599	146.4884	308.2404	0.533339	0	53.18196	0	162.0587	297.424	0.5107	507.647	608.1854	2.132981	0.730698	0.224494	0.671531	0.789853	0.047066	0.895105	94.96708	201.2554	0.471873	92.7373	
17	S 12-4557	119.5075	326.81	0.366582	144.9492	305.51525	0.949074	19.74301	69.37658	0.063732	108.4961	282.5313	0.410168	698.843	778.7598	1.297794	0.399878	0.041416	0.338847	0.87072	0.511705	0.577451	89.55984	173.2239	0.517017	88.98921	
18	S 09-1712	55.01075	182.459	0.301492	56.38571	187.6218	0.343803	3.727515	0	0.028753	40.35856	137.0515	0.305555	148.2407	91.2341	1.186047	0.433289	0.530887	0.241794	0.964731	0.899762	0.572258	11.60143	54.45504	0.21305	11.9181	
19	S 09-4320	80.43459	178.2575	0.451227	77.72041	145.2526	0.522691	153.1694	9.11179	0.070166	60.85106	135.2003	0.438505	371.7418	85.17736	1.84744	0.556952	0.606957	0.411919	0.846492	0.842979	0.612343	51.95584	129.3597	0.401639	52.90695	
20	S 09-3002	104.865	194.9525	0.057385	67.70438	245.127	0.299857	0	11.6343	0	42.96269	169.1965	0.266831	250.5431	830.1937	1.63119	0.796214	0.653107	0.956811	0.895058	0.362	74.80797	139.3031	0.537016	50.75711		
21	S 12-1709	130.6545	188.629	0.094116	165.0799	188.8079	0.144116	0	6.036883	0	7.949749	94.44249	0.091768	107.0506	46.3464	1.405719	1	0.565396	1	1.747865	0.85959	129.5538	262.1656	66.65059	0.391613	23.8991	
22	S 08-2296	430.6397	399.0264	1.393538	393.3719	306.6236	1.571048	0	22.41219	0	377.1351	291.751	1.23079	128.92	841.6193	202.6203	0.635739	0.334393	0.267779	0.556607	0.511899	202.0939	178.9567	222.0841	0.625596	141.385	
23	S 09-1059	58.16575	104.124	0.058618	86.51379	93.85529	1.515167	0	0	0	45.20182	50.37255	0.670454	99.635	824.9776	9.124465	0.572714	0.744174	0.944584	173.9621	134.0045	1.191549	25.78184	87.01447	0.307788	37.7355	
24	S 10-1164	232.1801	239.5275	0.32625	60.8204	239.5275	0.32625	0	0	0	3.0000	165.8204	0.21166	92.6201	1.27446	0.682197	0.530887	0.384322	0.094441	0.964731	0.724238	0.857048	1.37674	0.346492	185.5104	1.232121	
25	S 10-1164	232.1801	239.5275	0.32625	60.8204	239.5275	0.32625	0	0	0	2.97751	0	1.064845	0.070454	0.086909	0.051155	0.052231	0.094441	0.094441	0.072438	0.072438	0.072438	0.072438	0.072438	0.072438	0.072438	
26	S 08-3494	4.912059	14.20285	0.439827	53.89414	14.15289	0.453108	0	0	0	2.35508	9.07055	0.089899	84.90143	72.627	52.88651	1	1.681302	1	1.904158	1.207818	1.757985	73.8792	2.318201	19.2073		
27	S 09-2764	50.31609	206.2827	0.24104	54.66054	235.56162	0.208119	0	0	0	48.47421	206.9398	0.275929	514.1918	1340198	0.348771	0.314612	0.040418	0.065159	0.044879	0.505405	22.9993	116.5841	1.06558	23.4861		
28	S 09-1780	29.95275	100.64466	0.29842	82.25051	116.81106	0.137749	0	0	0	6.595142	84.01216	0.086788	1.062315	1.062315	0.062315	1	1.709587	0.933317	1.656004	9.064003	73.84101	0.124095	5.10339			
29	S 10-2405	85.891852	37.69941	0.227922	88.79919	32.52325	0.185803	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
30	S 09-3957	115.90961	64.14394	0.180697	102.3845	59.41192	0.24152	0	0	0	0	0	0	0	168.5342	402.9609	3.905724	1	0.653891	1	1.8885	1.04261	2.139836	27.80538	82.8034	0.3358	28.0316

For Windows users who have installed Anaconda, you can run the script by **executing Anaconda**

**Powershell Prompt** and type as follows:

> **python <path to the script file>/6.py**