Digital PD-L1 CPS score: A workflow for Positive cell detection and classification using QuPath

Synnøve Yndestad[1,2]

[1]K.G. Jebsen Center for Genome-Directed Cancer Therapy, Department of Clinical Science, University of Bergen; [2]Department of Oncology, Haukeland University Hospital

Synnove.Yndestad@uib.no

ABSTRACT

Identifying patients that can benefit from specific drugs is essential in cancer treatment.

The KEYNOTE 355 trial[1] established a survival benefit for patients with metastatic Triple Negative Breast Cancer (TNBC) receiving the PD-L1 targeting drug pembrolizumab in patients with a PD-L1 **Combined Positive Score (CPS) >= 10**.

CPS score is calculated by counting the number of PD-L1 positive tumor cells, the number of PD-L1 positive immune cells, divided by the total number of tumor cells. Manually counting positive and negative cells is time consuming and may be prone to variability. Using digital pathology to estimate CPS score can increase precision, accuracy and reproducibility. It may also offer the opportunity of saving time by batch processing a large number of slides.

Calculating CPS score digitally requires a workflow that can detect positive and negative stained cells, as well as identifying tumor cells and immune cells, while ignoring stromal cells and artefacts. This can be achieved by training a classifier, which then can be applied for batch processing of digital slides. Here I present a workflow for how to calculate CPS score using QuPath[2] as performed in our recent paper[3]. QuPath is an open-source digital pathology platform designed for the analysis of histopathological images. QuPath supports a range of image analysis tasks, making it a valuable tool in the field of pathology and biomedical research.

Goal:

➔Detect positive/negative PD-L1 staining pr cell

➔Identify tumor and immune cells

➔Calculate CPS score

How to score 200+ whole slides on my laptop?

Make 10 smaller projects and run the same workflow for all.

- QuPath is **Free Open Source** software
- Easy to generate scripts (coding not required)
- Consistent, Reproducible, Batch processing

Train classifier and generate scripts

1-Image pre-processing

-Create QuPath project-folder “PDL1_train”

-Import multiple randomly selected images of PD-L1 stained IHC slides, representing a variety of the type of tissue for training.

- Set image type to **Brightfield-DAB**

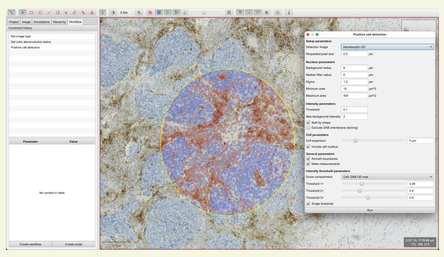
Go to **Analyze/Preprocessing/Estimate stain vectors**

- **Estimate stain vector**

Select a small representative area containing positive and negative staining, as well as slide background. The stain vector will help colour deconvolution to digitally separate Haematoxylin, DAB and residuals/background. Saving the stain vector used when training a classifier will ensure similar background-subtraction when applied on new images with same antibody stain.

Go to workflow and save as script “PDL1_EstimateStainVector.groovy”

2-Positive cell detection



Go to **Analyze/Cell detection/Positive cell detection**
Select appropriate settings for positive cell detection.
PD-L1 stains the cell membrane. There is no current reliable/easy method for selecting membrane staining in QuPath, so the most practical method is to set the **max intensity for the cell** as threshold for positive cell detection.

– Intensity threshold parameters: Score compartment -> Cell: DAB OD max

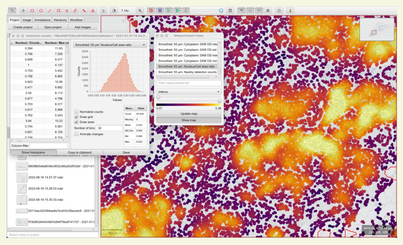
Select an appropriate threshold that balances detection of false positives and false negatives among the samples in the training set with the selected stain vector.

Test positive cell detection on negative slides, positive slides and slides with artefacts. In my dataset, standard settings and **Score compartment Cell: DAB OD max single intensity threshold 0.45** balanced the detection of weak positives with minimal false positives from background.

3-Add smooth features to cell detection

Go to **Analyze/Calculate features /Add smooth features**
In “Smooth object features” window select
Radius (FWHM) = 50 um

Smoothing features help the classification algorithm by supplementing all the measurements with the weighted sum of the corresponding measurements of the neighbouring cells.
Tumor cells correspond well with the Nucleus/cell area, but no single measurement is accurate enough, so we use machine learning to classify the detected cells into “Tumor” (Epithelial like) and “Stroma”.

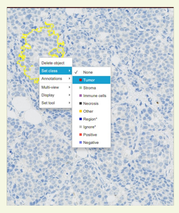


4-Classify cells and train classifier

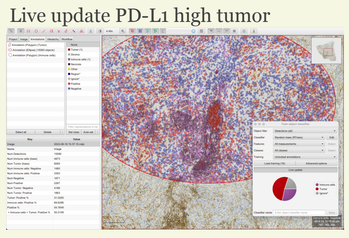
Go to **Classify/Object classification/Train object classifier**
Open the **Train Object Classifier** with live update. Select all images in project for training.
Available classifiers includes i.e Artificial neural networks, Random trees, and K nearest neighbor.
Choose one that works for your project. Random trees performed well in this setting.

Add Tumor/Stroma/immune cell annotation
Use brush-tool or wand/polygon tool to annotate an area - Control-click -> Set class
Repeat for all images in PDL1_train project.
Add multiple annotations for
- Tumor - Immune cell - Ignore (Red blood cells, debree, necrosis, artefacts etc)

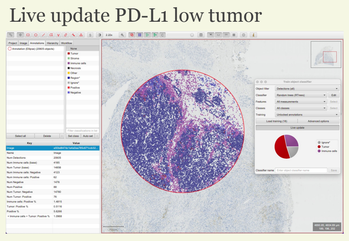
Train classifier



Live update PD-L1 high tumor



Live update PD-L1 low tumor



Do manual checks for correct annotations, and add more annotations where needed. Pay special attention to i.e. red blood cells and artefacts, add them to the “ignore” category.
Save the classifier when satisfied and load the saved classifier when classifying images in a new project.

Saved classifier as “PDL1_TumorImmuneIgnore_cellClassifier.json” in the project folder:

-> classifiers -> object_classifiers

All steps are logged in the workflow tab.
Save the steps –**Positive cell detection**, -**Smooth feature**s and -**Object classifier** as a groovey script.

Workflow for CPS Score

Create a QuPath project for images to be batch processed.
Add pre-trained classifier to appropiate folder in project.

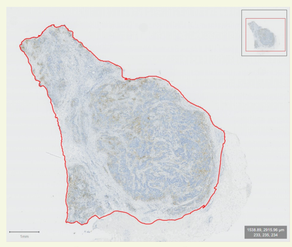
1-Image pre-processing

Run for **Project** :
Script: PDL1_EstimateStainVector.groovy

```
setImageType('BRIGHTFIELD_H_DAB');
setColorDeconvolutionStains('{"Name" : "H-DAB-PDL1", "Stain 1" : "Hematoxylin",
                            "Values 1" : "0.77475 0.55533 0.30229",
                            "Stain 2" : "DAB",
                            "Values 2" : "0.20981 0.43913 0.87358",
                            "Background" : " 232 233 233 "}');
```

Sets the image type to Brightfield HE with DAB.
Then a previously defined stain vector is applied to help colour deconvolution and disregard background.

2-Slide QC check and annotate area



For each image in project, use manual selection with wand tool/ brush to annotate areas to count.
Remove areas with artefacts, necrosis, dust i.e. and normal tissue.

Check slide for proper staining or artefacts:

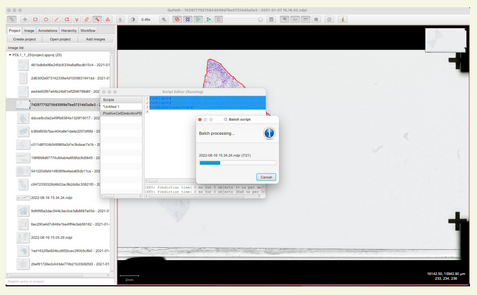
- Check integrity of slide, size and if enough tumour cells (minimum 100).
- Check DAB staining and artefacts. PD-L1 should stain membrane and not cytoplasm/nucleus.
- Annotate area, exclude areas with normal tissue, artefacts and necrosis.

3-Detect positive cell and classify cells

Run for Project:
Script: PositiveCellDetectionPDL1_w_CellClassificationTumorImmuneOther.groovy

Select annotations, Positive Cell Detection, Smooth features, Object classifier

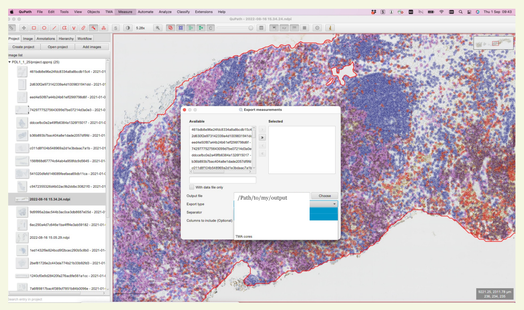
```
selectAnnotations() ;
runPlugin('qupath.imagej.detect.cells.PositiveCellDetection',
          '{"detectionImageBrightfield": "Hematoxylin OD",  "requestedPixelSizeMicrons": 0.5,
          "backgroundRadiusMicrons": 8.0,  "medianRadiusMicrons": 0.0,  "sigmaMicrons": 1.5,
          "minAreaMicrons": 10.0,  "maxAreaMicrons": 500.0,  "threshold": 0.1,  "maxBackground": 2.0,
          "watershedPostProcess": true,  "excludeDAB": false,  "cellExpansionMicrons": 5.0,
          "includeNuclei": true,  "smoothBoundaries": true,  "makeMeasurements": true,
          "thresholdCompartment": "Cell: DAB OD max",  "thresholdPositive1": 0.45,
          "thresholdPositive2": 0.4,  "thresholdPositive3": 0.6000000000000001,
          "singleThreshold": true}');
runPlugin('qupath.lib.plugins.objects.SmoothFeaturesPlugin',
          {"fwhmMicrons": 50.0,  "smoothWithinClasses": false}');
runObjectClassifier("PDL1_TumorImmuneIgnore_cellClassifier");
```



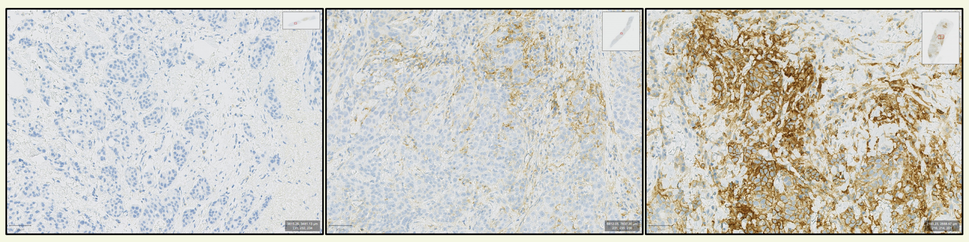
This will select the annotations in each image, do positive cell detection, add smooth features as defined, and finally runs the object classifier to classify cells as tumour/immune/ignore.

4-Export results

Export pr project all Annotations:
Will export summarized measurements in annotations.
Calculate CPS score i.e. in R.



$$\text{CPS} = \frac{\#\text{ Positive tumour cells} + \#\text{ positive immune cells}}{\text{Total viable tumour cells}} \times 100$$



CPS = 0 | CPS = 1-10 | CPS >= 10

“All models are wrong, but some are useful.”
George Box

REFERENCES

1. Cortes J, Rugo HS, Cescon DW, et al: Pembrolizumab plus Chemotherapy in Advanced Triple-Negative Breast Cancer. N Engl J Med 387:217-226, 2022
2. Bankhead P, Loughrey MB, Fernandez JA, et al: QuPath: Open source software for digital pathology image analysis. Sci Rep 7:16878, 2017
3. Yndestad, S., C. Engebrethsen, A. Herencia-Ropero, O. Nikolaienko, O. K. Vintermyr, R. K. Lillestol, L. Minsaas, B. Leirvaag, G. T. Iversen, B. Gilje, E. S. Blix, H. Espelid, S. Lundgren, J. Geisler, H. S. Aase, T. Aas, E. G. Gudlaugsson, A. Llop-Guevara, V. Serra, E. A. M. Janssen, P. E. Lonning, S. Knappskog and H. P. Eikesdal : Homologous Recombination Deficiency Across Subtypes of Primary Breast Cancer. JCO Precis Oncol 7:e2300338, 2023

Haukeland University Hospital

UNIVERSITY OF BERGEN