protocols.io

APR 01, 2024

🌐 CODA (part 5): nuclear coordinate generation | HuBMAP | JHU-TMC

Kyu Sang Han[1], Pei-Hsun Wu[1], Joel Sunshine[2], Ashley Kiemen[2],
Sashank Reddy[2], Denis Wirtz[1,2]

[1]Johns Hopkins University; [2]Johns Hopkins Medicine

Human BioMolecular Atlas Program (HuBMAP) Method Development Community

TMC - Johns Hopkins University

Kyu Sang Han
Johns Hopkins University

ABSTRACT

In this section, generate nuclear coordinates on the
high-resolution H&E images through color deconvolution and identification
of two-dimensional intensity minimums in the hematoxylin channels of the images
(corresponding to the dark blue nuclei). This calculation is done on the
high-resolution tif images saved inside pth10x.

Note on cell detection: It is important that the cell
coordinates are generated on unregistered images, as warping caused by the
registration process can cause inaccuracies in nuclear detection performed on
the registered images. Instead, detect coordinates on the unregistered images,
then apply the registration transformations to the cell coordinates to
determine their positions in registered space.

Nuclear coordinate generation

1   generate a mosaic image containing tiles from nine randomly chosen high-resolution images. This will be the image used to optimize the parameters for the cell detection algorithm. Given the path to the high-resolution images, the function will randomly choose nine files (filenames can overlap if the folder contains fewer than nine files). The low-resolution version of each image will be loaded and displayed to the user, prompting the user to click on a region containing tissue. This will be repeated nine times, after which the high-resolution mosaic image will be generated.

     Note: When manually selecting regions for the mosaic image, select regions with various morphologies to ensure that your cell detection algorithm is robust.

2   **make_cell_detection_mosaic(pth10x);**
     This function will create a subfolder containing the mosaic image:
     pthmosaic=[pth10x,'cell_detection_validation'];

3   Manually count the nuclei on the mosaic image to generate ground-truth coordinates. Given the path to the mosaic image, this function will display the image to the user and prompt the user to zoom in. Zoom to a region where nuclei are clear, then press 'spacebar.' Click on each nucleus in the zoomed region, then press 'z' to zoom or scroll to another region in the image. Continue until all nuclei in the mosaic have been annotated. At any time, exit the code by pressing 'z' and selecting continue later. When you recall the code it will automatically continue from where you left off. When you are finished, press 'z' then select 'Quit'.

For a mosaic image comprised of 9 200x 200 micron^2
10x magnification tiles, manual annotation of nuclei for a medium dense tissue should take 30 – 45 minutes for a trained user.

manual_cell_count(pthmosaic);

This will create a subfolder named 'manual detection' containing the manually identified coordinates inside a mat file in a variable named 'xym.'

pth_mosaic_manul_coords=[pthmosaic,manual detection'];

4   Next, determine the optimal intensity cutoff and minimum spacing between nuclei to most maximize the true positives and minimize the false positives and false negatives in automatic determination of nuclear coordinates. This is done through comparison of the manually generated cell coordinates to several variations of automatically generated coordinates.

5   First, deconvolve the mosaic image to get its hematoxylin channel.
If the images are H&E, use the ImageJ default optical densities for H&E images.

This function generates output images containing the hematoxylin and eosin (or DAB) channel images:
pthmosaicH=[pth10x,'Hchannel'];
pthmosaicE=[pth10x,'Echannel'];

6   Next, optimize the parameters by iteratively generating nuclear coordinates on the mosaic image and comparing them to the manual cell detection.

get_nuclear_detection_parameters(pthmosaic)

This will generate a subfolder named 'automatic detection' containing the optimal automatically generated cell coordinates, and a subfolder named 'optimization params' containing the determined parameters.

paramsfile=[pthmosaic,'automatic_detection\optimized_params.mat'];

7   Now, apply the color deconvolution to the entire high-resolution dataset using the calculated parameters. We first deconvolve the images to get their hematoxylin channel. If the images are H&E, use the ImageJ default optical densities for H&E images:

stain_type=1;
If the images are IHC, use the ImageJ default optical densities for H DAB images.

stain_type=2;
deconvolve_histological_images(pth10x,stain_type);

This function generates output images containing the hematoxylin and eosin (or DAB, for IHC) channel images:

pthH=[pth10x,'Hchannel'];
pthE=[pth10x,'Echannel'];

8   Finally, apply the cell detection to the hematoxylin channel of the high-resolution images using the optimized parameters:

pthparams=[pth10x,'cell_detection_validation\automatic_detection'];
cell_detection(pthH,pthparams);

9   The workflow in this section will create a subfolder inside the high-resolution hematoxylin-channel tif image folder named 'cell_counts'. Within 'cell_counts' will be a mat file corresponding to each tif image. Inside each mat file will be a variable named 'xy' with coordinates for each cell found in the corresponding image.

pthcoords=[pth10x,'cell_counts\'];