Below describes the steps and corresponding Matlab functions to create a composite image and subsequent image analysis by pixel intensity and pixel counting for a single slide using provided sample files in folder called ‘Testdata’. Matlab command lines will be shown with > and in purple for clarity. Functions were developed using Matlab version 2019b with Image Processing toolbox.

Images from the same slide must be saved as TIF and named using the following convention: SLIDEINFO\_XXdeg.TIF, where SLIDEINFO = user defined name to identify slide sample and XX = rotation angle. The rotation angle from the filename is used by different Matlab codes for various functions including the identifying the reference image and images for composite formation. For example filename labeled as ‘Sample1\_45deg.TIF’, identifies an image of histology slide identified as Sample1 and obtained after rotating the sample stage 45 deg from reference 0 (using identification markers as described in the manuscript). PSR images from the same sample and corresponding blank slide are saved in the same folder.

1. Perform background correction using BkgndCorrect\_histology.m. This function will move acquired images to ‘originals’ subfolder prior to background subtraction and save background corrected images to ‘BkCorr’ subfolder

Example:

> image\_dir = '\\Rootdirectory\Testdata';

>BkgndCorrect\_histology(image\_dir)

1. Co-register background corrected images against reference image, i.e. at 0°, using Coreg\_BkgndCorrect\_histology.m. Input image directory is directory with background corrected images from step 1. Co-registered images will be saved in ‘Rotated’ subfolder.

Example:

>image\_dir = '\\Rootdirectory\Testdata\BkCorr';

>Coreg\_BkgndCorrect\_histology(image\_dir)

1. Create a single composite image from images obtained over a range of rotation (ϴR) and specific angle of increment (∆ϴ) using Make\_composite.m. Input image directory is directory with co-registered background corrected images from step 2. The composite image will be saved in subfolder called ‘Composite\_yy\_zz’, where yy = minimum angle in the range of rotation and zz = maximum angle in the range of rotation.

Example: Combine images corresponding to rotation angle increment = 60° over rotation range of 0° to 358° (=1). Composite image will be saved in subfolder called ‘Composites\_0\_358’.

> image\_dir = ‘\\Rootdirectory\Testdata\Rotated';

> delta\_angle = 60;

> angle\_range = 1;

> Make\_composite(image\_dir, delta\_angle, angle\_range)

Alternate syntax angle\_range

> angle\_range = [0 358];

Example: Combine images corresponding to rotation angle increment = 60° over rotation range of 0° to 178° (=2). Composite image will be saved in subfolder called ‘Composites\_0\_178’.

> image\_dir = ‘\\Rootdirectory\Testdata\Rotated';

> delta\_angle = 60;

> angle\_range = 2;

> Make\_composite(image\_dir, delta\_angle, angle\_range)

Alternate syntax to angle\_range

> angle\_range = [0 178];

Example: Combine images corresponding to rotation angle increment = 60° over rotation range of 180° to 358° (=3). Composite image will be saved in subfolder called ‘Composites\_0\_178’.

> image\_dir = ‘\\Rootdirectory\Testdata\Rotated';

> delta\_angle = 60;

> angle\_range = 3;

> Make\_composite(image\_dir, delta\_angle, angle\_range)

Alternate syntax to angle\_range

> angle\_range = [180 358];

1. Co-register composite images against reference image by using Coreg\_composite.m. Input image directory is directory with composite images from step 3. Co-registered composite images are saved in the image directory. Prior to co-registration a copy of the composite images are saved in ‘’ subfolder

Reference image may be defined as:

1. the composite image in composite subfolder (from step 3) corresponding to rotation increment = 60°. This is the default if no variables or only ‘image\_dir’ variable is provided.
2. the composite image in composite subfolder (from step 3) corresponding to user defined rotation angle increment using ‘refangle’ variable.
3. an alternate image file using ‘fn\_ref’ variable. This can be the case when the image used to create the contour file is not the same images used to create composite images.

Example: Co-register all composite image in image\_dir against composite image from rotation increment = 60°.

> image\_dir = ‘Rootdirectory\Testdata\Composites\_0\_358’

> Coreg\_composite(image\_dir)

Example: Co-register all composite image in image\_dir against composite image from rotation increment = 12°.

> image\_dir = ‘Rootdirectory\Testdata\Composites\_0\_358’

> refangle = 12 ;

> Coreg\_composite(image\_dir,refangle)

Example: Co-register all composite image in image\_dir using a specific reference image file that in a different directory. It is necessary to provide refangle variable even if it is empty of value.

> image\_dir = ‘Rootdirectory\Testdata\Composites\_0\_358’

> fn\_ref = ‘Rootdirectory\Test Data\RefIM\DATA\_H006\_R1\_20\_PR\_002X\_0deg\_Ref.TIF’

> Coreg\_composite(image\_dir,[],fn\_ref)

1. Perform pixel intensity and pixel counting analysis on composite images using Call\_ImageAnalysis.m. Input image directory is directory with composite images from step 3 or 4. The function requires the mat file containing histology contours from the pathologist. The output is a mat file saved in ‘Analysis’ subfolder, with pixel indices for every pixel in the one degree polygon formed in the cap region (i.e. between the lumen and necrotic core contours) and corresponding total intensity and total pixel count per one degree polygon. This code can be used to perform pixel counting at different pixel intensity threshold as defined by variable pxthresh (default value = 600). Note that Call\_ImageAnalysis.m is dependent on two other functions: Comp\_ImageAnalysis.m and intersections.m.

Example: Analyze pixel intensity and count pixels above intensity = 600 in cap region of composite images in image directory.

> image\_dir = ‘Rootdirectory\Testdata\Composites\_0\_358’;

> contour\_fn = ‘Rootdirectory\Testdata\ContourData\DATA\_H006\_R1\_20\_PU\_002X\_CT.mat’;

> Call\_ImageAnalysis(image\_dirComp,contour\_fn)

Alternate syntax

> pxthresh = 600;

> Call\_ImageAnalysis(image\_dirComp,contour\_fn, pxthresh)

Results will be saved under .mat file: Rootdirectory\Test Data\Composites\_0\_358\Analysis\DATA\_H006\_R1\_20\_PR\_002X\_60deg\_600.mat

Example: Analyze pixel intensity and count pixels above intensity = 400 in cap region of composite images in image directory.

> image\_dir = ‘Rootdirectory\Testdata\Composites\_0\_358’;

> contour\_fn = ‘Rootdirectory\Testdata\ContourData\DATA\_H006\_R1\_20\_PU\_002X\_CT.mat’;

> pxthresh = 400;

> Call\_ImageAnalysis(image\_dirComp,contour\_fn, pxthresh)

Results will be saved under .mat file: Rootdirectory\Test Data\Composites\_0\_358\Analysis\DATA\_H006\_R1\_20\_PR\_002X\_60deg\_400.mat