Supplementary Material For:

Use and validation of epithelial recognition and fields of view algorithms on virtual slides to guide TMA construction

Sanford H. Barsky¹,², Lynda Gentchev³, Amitabha S. Basu⁴, Rafael E. Jimenez³, Kamel Boussaid⁴, and Abhi S. Gholap⁴

¹Department of Pathology, University of Nevada School of Medicine, Reno, NV, USA ²Nevada Cancer Institute, Las Vegas, NV, USA, ³Ohio State University College of Medicine, Columbus, OH, USA, and ⁴BioImagene, Inc., Cupertino, CA, USA

BioTechniques 47:927-938 (November 2009) doi 10.2144/000113207
Keywords: epithelial recognition algorithms; fields of view; tissue microarrays; virtual slides; automation of construction

Tables

Table S1: Gridding of whole slides

1. Get a single bounding box around the tissue area in the input image.

2. Perform gridding within this bounding box using these conditions:
   a. All the FOV will have same dimensions.
   b. The dimensions of all the FOV would be less than or equal to 2000 (2000X2000 pixels).

3. Within these areas of identification or interest (AOI), mark those FOV which don’t have any tissue information. These no tissue FOV can create different regions within the identified AOI. So find these multiple AOI if any, which are formed after deleting no tissue FOV.

4. Merge overlapped AOI if any.

5. Make a list of the independent AOI in the image and perform indexing of the FOV within them in terms of row number and column numbers.

6. Return with this information to application.
Table S2: Flowchart for epithelial cell detection

Block 2.0: Cell Detection

Block 2.1: Blur the image using Gaussian kernel.
1. Sigma = 3.
2. Compute kernel size = \(1 + \frac{2 \times \text{cell}}{(2.5 \times \text{Sigma})}\).
3. Compute Gaussian kernel \(f(x) = \frac{\text{power}(2.71828, -\frac{0.5 \times x^2}{(\text{Sigma} \times \text{Sigma})})}{(\text{Sigma} \times \text{sqr}(2 \pi))}\).
4. Convolve Gaussian kernel on image

Block 2.2: Mark centers & axes of the cells -
1. Mark the center pixels of the dark cells - The center pixel intensity should be less than the 5 neighboring pixel intensities in 4 directions (left, top, right, and bottom).
2. Find horizontal axis pixels - If the pixel intensity is less than 5 neighboring pixels intensities in top direction & 5 neighboring pixels in bottom direction then pixel is marked as horizontal axis pixel.
3. Find horizontal axis pixels - If the pixel intensity is less than 5 neighboring pixels intensities in left direction & 5 neighboring pixels in right direction then pixel is marked as vertical axis pixel.

Block 2.3: Detect cells -
1. Label center & connected axes.
2. Disconnect horizontal & vertical axes joining centers of cells.
3. Identify cells bounding box from labeled objects.
4. Compute mean & standard deviation for each object within bounding box.
5. Extend cell from center up to the value (mean - standard deviation) within the bounding box.

Table S3: Flowchart for image enhancement

Block 1.0: Image Enhancement

Block 1.1: Preprocess
1. Compute histogram of each color plane.
2. Compute mean & standard deviation of each color plane.
3. If standard deviation of each plane is less than 5, image is uniform & do not apply following steps.
4. Compute cumulative histogram for each plane.
5. Get minimum intensity value at 0.5 percentage & maximum intensity value at 99.5% of cumulative histogram for each color plane.
6. Set pixel intensities below minimum intensity value to 0 & pixel intensities above maximum intensity value to 255 for each color plane independently.
7. Stretch the pixel intensity values between minimum to maximum between 0 to 255 for each color plane.

Block 1.2: Background Removal
1. Compute mean & standard deviation of each color plane.
2. Compute minimum background intensity for each color plane using formula
   \[
   \text{minimum background intensity} = \text{mean} - \frac{(\text{standard deviation})}{(\text{log (standard deviation)} + 1)}
   \]
3. Set pixel intensities of all color plane to 255 as background pixel if pixel intensity is more than “minimum background intensity” for any plane.
Table S4: Flowchart for Area of Identification (AOI) algorithm
Table S5: Flowchart for identification of AOI at 4x

- Convert Input Image to Gray raster (A1)
- Apply Auto Contrast on Gray Raster (A2)
- Segment Black regions on Gray value, RedGreen% & BlueGreen% (A3)
- Label Black Objects and mark as TissueArea or Non TissueArea (A4)
- Remove Black Objects regions from I/p image (A5)
- Get Gray Raster Image (A6)
- Apply median Filter (Mask size = 5) (A7)
- Apply Auto Contrast (A8)
- Compute Gray Mean and Std. Deviation (A9)
- Segment Image for AOI (uThresholdVal) (A10)
- Mark AOI by BLUE COLOR (A11) pcxOut
- Identify and Filter Folded Artifact
Table S6: Flowchart for identification of AOI at 10x

1. Convert Input Image to GrayRaster puGray (B1)
2. Apply Median Filter on GrayRaster puGray (B2)
3. Apply Auto-Contrast puGray (B3)
4. Compute Gray Mean-Std. Deviation of the image puGray (B4)
5. Threshold Image with Value Thresh = mean – Std. Dev. puGray (B5)
6. Mark the Threshold image by Blue color as AOI pcxOutput (B6)

Table S7: Calculation of epithelial percentages and correction for minimal tissue within FOV

1. Count the total number of blue pixels of the segmented image after step A11 (for image at 4x) (Table 5) or after step B6 (for image at 10x) (Table 6) - This is total epithelial pixels.
2. Count the total number of AOI pixels (non background pixels).
3. Count the total number of stromal pixels (Total AOI pixel-Total Blue pixel)
4. Calculate the percentages of epithelial and stromal pixels with respect to total AOI pixel.
5. Some FOV may contain minimal tissue compared to grid area. In that case epithelial % may be artifactually elevated. To deal with this issue grids with total tissue < 10% of the grid area are automatically removed from the analysis.
Details of image enhancement and ERA

Image enhancement

Verifying the content of the image. The cell detection process could be simplified if we skipped processing fields or areas of slides that did not have tissue. This was achieved by computing the mean and standard deviations of the red, blue, and green planes of the image. In the case of the areas without tissue, there would be expected to be little or no variations in colors. Standard deviations, which reflected variation, would be low.

Mask removal. The mask or background in an image was represented by determining the mean of pixel values. By mapping the mean of pixel values to the mid-value of the pixel value range, we achieved a mask removal effect or normalization of the background to a standard value.

Contrast enhancement. For contrast enhancement, the algorithms first differentiated the objects of interest from the background. We transformed the image in such a manner that objects that were darker than the background became even darker. If the background was bright (which normally was the case), the pre-processing algorithms caused the background to become even brighter. Contrast in a digital image referred to the difference in color values between any two given pixels. Color values at a given pixel were independently computed from red, green, and blue components of the given color image. The first step in this preprocessing was the determination of the active range of intensities in each of the colors. We computed the histogram of all color planes (red, green, and blue) of the input image. We used these histograms to compute a minimum intensity such that, starting from lowest intensity, the cumulative pixels up to minimum intensity was equal to 0.5% of total pixels in the image. Next, we mapped the active range to (0, 255). All pixels with values less than minimum intensity were discarded (see Table 1S).

Background removal: The region of interest (ROI) was detected based on two distinctive features of the epithelial area. The epithelial area was darker compared with the stromal area and the epithelial cells were more densely packed than the stromal cells. We computed the minimum background intensity using mean and standard deviation. Equation (1) is used to compute minimum background intensity:

\[
\text{Minimum background intensity} = M - (D/(\log(D)+1)),
\]

where \(D\) is the standard deviation and \(M\) is the mean. The minimum background intensity was computed independently for each color plane. If any color component of a pixel was greater than the respective minimum background intensity, then the pixel was treated as a background pixel. All background pixels were given 255 values in red, green, and blue planes.

Epithelial recognition algorithms (ERAs)

ERAs employed the Gaussian kernel. The Gaussian kernel is a well-known imaging concept for weighted averaging of pixels in a small window centered around a given pixel. Keeping the window size equal to the width of two typical epithelial cells, we were able to differentiate between the densely packed epithelial area and the less densely packed stromal area. In this analysis, the weighted average was higher in the stromal area.

The detailed mathematics of this imaging maneuver were as follows: For epithelial cell detection, the Gaussian kernel of sigma 3 was used. The Gaussian kernel was computed using the following equation

\[
\text{Gaussian kernel } f_x = \text{power } (2.71828, \frac{-0.5x^2}{(\text{Sigma}\times\text{Sigma})})/((\text{Sigma}\times\text{squ}(2\times\pi))).
\]

[Eq. 2]

The Gaussian kernel was used for convolution of the modified image using the equation

\[
\text{kernel size} = 1 + 2\times\text{cell}(2.5\times\text{Sigma}).
\]

[Eq. 3]

\[
G = \sum_{x=-\text{kernel size}/2}^{x=\text{kernel size}/2} f_x \times I_x
\]

[Eq. 4]

where \(G\) is the Gaussian value at \(c\) position, and \(I_x\) is the pixel value at \(x\).

The next step was marking the extent of each cell. Marking a cell was done by cropping the image starting from the cell center. Cropping was stopped once the pixel intensity was brighter than a limit or the bounding box of the cell was reached. The bounding box of a cell was determined based on the two symmetry curves.

The final step was removing the stromal cells. In this step, we used four stages for removing stromal cells. First, we processed the image to detect all cells, that is stromal, epithelial, and other cells. We then segmented the image by thresholding using image statistics, at (mean-standard deviation). In the second stage, we computed the elongation ratio of each of the segmented objects. The elongation
ratio was the ratio of major axis over minor axis. In the next stage we tried to identify stromal cells that were elongated and in the range of 45–500 pixels. These limits were derived after a detailed study of images at the standard resolution. We considered a cell as stromal if the elongation ratio was more than factor $F$, where

$$ F = \max(\log(\text{Object Area/(float)12}),1.5) $$

[Eq. 5]

Finally, we removed all cells overlapping with any one of the detected stromal cells.

### Abbreviated synoptic code

#### Detailed code of the algorithms

The detailed source code of the algorithms is too lengthy to provide. However, an abbreviated synoptic code, which provides the essence of the source code, is as follows:

**Identify tissue area of interest (AOI)**

Once an image is brought into the system, the tissue area is separated out from the background using the intensity threshold criterion.

- **Input:** Input image
- **Output:** AOI image
- **Parameters:** Red intensity threshold, Green intensity threshold, Blue intensity threshold, and Gray intensity threshold

```plaintext
FOR Row index = 0 to Image Height
    FOR Column index = 0 to Image Width left to Right Scan
        Read Blue Value of Pixel at (Row, Column)
        Read Green Value of Pixel at (Row, Column)
        Read Red Value of Pixel at (Row, Column)
        Compute Gray Value of Pixel
        IF Blue Value > Blue Intensity Threshold And
        Red Value > Red Intensity Threshold And
        Green Value > Green Intensity Threshold Or
        Gray Value < Gray Intensity Threshold
            Set AOI Pixel Black color (Non Tissue Area)
        Else
            Break (exit out of the first loop construct)
        End IF
    ENDFOR
    FOR Column index = Image Width to 0 Right to left Scan
        Read Blue Value of Pixel at (Row, Column)
        Read Green Value of Pixel at (Row, Column)
        Read Red Value of Pixel at (Row, Column)
        Compute Gray Value of Pixel
        IF Blue Value > Blue Intensity Threshold And
        Red Value > Red Intensity Threshold And
        Green Value > Green Intensity Threshold Or
        Gray Value < Gray Intensity Threshold
            Set AOI Pixel Black color (Non Tissue Area)
        Else
            Break (exit out of the first loop construct)
        End IF
    ENDFOR
ENDFOR
```

**Note:** The above procedure is repeated for top-to-bottom and bottom-to-top scan of the image.
Preprocess image
The preprocessing algorithms perform two activities.

Mask removal. If the image has any mask at the time it is brought into the system, it is removed at this stage.

Input: Input image
Output: Mask removed image
Parameters: Histogram clip value

<table>
<thead>
<tr>
<th>COMPUTE Mean, Standard Deviation and Histogram for R, G, and B planes</th>
</tr>
</thead>
<tbody>
<tr>
<td>IF Red Standard Deviation &lt; 5 AND</td>
</tr>
<tr>
<td>Green Standard Deviation &lt; 5 AND</td>
</tr>
<tr>
<td>Blue Standard Deviation &lt; 5</td>
</tr>
<tr>
<td>EXIT FUNCTION</td>
</tr>
<tr>
<td>END IF</td>
</tr>
<tr>
<td>// Find minimum &amp; maximum values of each color plane</td>
</tr>
<tr>
<td>For Index = 0 to 255</td>
</tr>
<tr>
<td>Red Value Percentage + = (100 * RedHistogram [index] / Number of Pixels)</td>
</tr>
<tr>
<td>IF Red Value Percentage &lt; = Histogram Clip Value</td>
</tr>
<tr>
<td>Minimum Red Value = index</td>
</tr>
<tr>
<td>END IF</td>
</tr>
<tr>
<td>IF Red Value Percentage &lt;= 100 - Histogram Clip Value</td>
</tr>
<tr>
<td>Maximum Red Value = index</td>
</tr>
<tr>
<td>END IF</td>
</tr>
<tr>
<td>// Similarly Minimum Green value, Maximum Green Value, Minimum Blue Value, Maximum Blue value is computed.</td>
</tr>
<tr>
<td>End For</td>
</tr>
<tr>
<td>// Stretch Histogram for Red Channel</td>
</tr>
<tr>
<td>For Index = 0 to 255</td>
</tr>
<tr>
<td>Stretch = Maximum Red Value - Minimum Red Value</td>
</tr>
<tr>
<td>IF Index &lt; Minimum Red Value</td>
</tr>
<tr>
<td>RedHistogram [Index] = 0</td>
</tr>
<tr>
<td>ELIF Index &gt; Maximum Red Value</td>
</tr>
<tr>
<td>RedHistogram [Index] = 255</td>
</tr>
<tr>
<td>ELSE</td>
</tr>
<tr>
<td>RedHistogram [Index] = (255 * ((Index - Minimum Red Value) / Stretch))</td>
</tr>
<tr>
<td>END IF</td>
</tr>
<tr>
<td>End For</td>
</tr>
<tr>
<td>// Similarly Stretch Histogram for Green Channel and Blue Channel</td>
</tr>
<tr>
<td>Apply Red Histogram to Red channel of Input Image</td>
</tr>
<tr>
<td>Apply Green Histogram to Green channel of Input Image</td>
</tr>
<tr>
<td>Apply Blue Histogram to Blue channel of Input Image</td>
</tr>
</tbody>
</table>
**Pixel classification.** Once the mask is removed, the pixels are classified into ‘Stained’ and ‘Non-stained’ pixels.

**Input:** Preprocessed image  
**Output:** Pixel classified image  
**Parameters:** Staining threshold

```
// Enhance Colors of Input Image
FOR Row index = 0 to Image Height
    FOR Column index = 0 to Image Width
        Read Blue Value of Pixel at (Row, Column)
        Read Green Value of Pixel at (Row, Column)
        Read Red Value of Pixel at (Row, Column)
        Blue Percent = 255 * (Blue Value – Red Value)/(Blue Value – Red Value + 1)
        New Blue Value = Maximum of Blue Percent and 0
        Red Percent = 255 * (Red Value – Blue Value)/(Red Value – Blue Value + 1)
        New Red Value = Maximum of Red Percent and 0
        IF Red Percent > Blue Percent
            Set New Red Value at (Row, Column)
        ELIF
            Set New Blue Value at (Row, Column)
        END IF
    END FOR
END FOR
// Classify Pixels
FOR Row index = 0 to Image Height
    FOR Column index = 0 to Image Width
        Read Blue Value of Pixel at (Row, Column)
        Read Green Value of Pixel at (Row, Column)
        Read Red Value of Pixel at (Row, Column)
        Stain Percent = 100 * (Red Value – Blue Value)/(Red Value – Blue Value + 1)
        IF Stain Percent < Staining Threshold
            Set Blue Color at (Row, Column) // Non Stained Pixel
        ELSE
            Set Red Color at (Row, Column) // Stained Pixel
        END IF
    END FOR
END FOR
```

**Image segmentation**
The image segmentation algorithms perform four activities: *(a)* Split preprocessed image into stained and non-stained images, *(b)* Calculate the image statistics for both the images, *(c)* Calculate the threshold values for both the images using image statistics, *(d)* Threshold both types of images using the threshold values, and combine the segmented images into single image

**Input:** Preprocessed image  
**Output:** Pixel classified image  
**Parameters:** Staining threshold

```
// Split Preprocessed Image into Stained and Non Stained Images
Copy Pre processed Image to Non Stained Image
Copy Pre processed Image to Stained Image
FOR Row index = 0 to Image Height
    FOR Column index = 0 to Image Width
        Read Color Value of Pixel at (Row, Column) from Pixel Classified Image
        IF Color Value = RED
            Set White Color at (Row, Column) in Non Stained Image
        ELIF Color Value = BLUE
            Set White Color at (Row, Column) in Stained Image
        END IF
    END FOR
END FOR
```
Compute gradient of segmented objects
Here, we find out the contrast between the background and the edge of the segmented object.

Input: Preprocessed image
Output: Pixel classified image
Parameters: Staining threshold

// Gradient is computed along Hue Plane
Each Pixel in the Image Consider its 3 X 3 Neighborhood
a1 a2 a3
a4 a5 a6 3x3 Neighborhood
a7 a8 a9
Where a1…a9 are Hue values of Pixels and a5 is current pixel
X = maximum of (a4 - a5), (a5 - a6), (a1 - a5), (a5 - a9)
Y = maximum of (a2 - a5), (a8 - a5), (a3 - a5), (a7 - a5)
Gradient = X+Y

// similarly Gradient is computed along Saturation and Intensity Plane
Compute Maximum and Minimum Gradient Values for Each Plane
Maximum Hue Gradient, Minimum Hue Gradient
Maximum Saturation Gradient, Minimum Saturation Gradient
Maximum Intensity Gradient, Minimum Intensity Gradient

// Actual Gradient is computed along Hue Plane
FOR Row index = 0 to Image Height
FOR Column index = 0 to Image Width
Read Gradient Value of Pixel at (Row, Column)
Hue Gradient = 255 * (Gradient – Minimum Hue Gradient) / Maximum Hue Gradient
END FOR
END FOR

//Similarly Actual Gradient is computed along Saturation and Intensity Plane

Validate segmented objects
First, large and blurred objects are filtered out as artifacts. This is done using the gradient information at the object boundary. Then, the objects having gradients below a certain threshold at the boundary are rejected.

Identify epithelial nuclei
A. Identify features of segmented objects. Various geometrical features of object boundary are computed. These features are given differential weights and are used in classifying the objects into epithelial and not-epithelial, and stromal and lymph cells.

Input: Segmented image
Output: Segmented image and geometric values of objects
Parameters: Lymph and Stromal cell filter threshold

//Compute Boundary Irregularity for each object.
FOR boundarypoint= 1 to End boundarypoint
BoundaryPoint1=boundarypoint - 3
BoundaryPoint2=boundarypoint
BoundaryPoint3=boundarypoint + 3
Compute Angle between BoundaryPoint1 and BoundaryPoint3 at BoundaryPoint2
IF Angle < Angle Threshold
Increment Irregularity Count //For each Object
END IF
END FOR

// Compute Object Circularity, Elongation Ratio and Normalized Elongation Ratio.
Compute Thickness of Object
Compute Alignment of Boundary Pixels.
Compute Convex Hull Ratio.
Compute Staining Percentage of Object.
Compute Gradient of Object Boundary Pixels.
Compute Slope of Object Boundary Pixels.
Compute Darkness of Object.
B. Identify Epithelial Area as Mask for Segmented Image. This function includes (1) Identifying nuclei-like objects in the image, (2) expanding these nuclei up to their boundaries, and (3) Using this image with nuclei as a mask on the segmented image, breaking connected nuclei and marking their centers.

### Input: Preprocessed image
### Output: Mask image

```plaintext
// Gaussian Blur the Image
// Gaussian kernel of sigma 3 is used
Compute Gaussian Kernel
FOR all Pixels
    IF intensity of Pixel < Mean + Standard Deviation
        Pixel belongs to Region of Interest
    END IF
    IF ROI Pixel Intensity < 5 neighboring pixels in the top direction and 5 neighboring
        pixels in the bottom direction
        Pixel is on the curve of symmetry (horizontal)
    END IF
    // Similarly, a selected pixel will be considered to be on the curve of symmetry
    // (vertical) only if the pixel intensity is less than 5 neighboring pixels in the left
    // direction and 5 neighboring pixels in the right direction.
END FOR
Identify cell boundaries using the cell’s center and the two symmetry curves.
// In case of Overlapping Cells the mid-point on the symmetry curve joining the two
// cell centers (adjacent cells) is considered for identifying cell boundaries.
```

### Classify nuclei on stain
Pixel classified image is used for classification of nuclei into stained and non-stained nuclei. Two parameters control this classification: (a) the difference between red value and blue value of the pixel, and (b) the percentage of pixels having that difference in the nuclei.

### Input: Segmented image, Pixel classified image
### Output: Segmented image and geometric values of objects
### Parameters: Lymph and Stromal cell filter threshold

```plaintext
// Staining Color control- difference between Red value and Blue value of the pixel
// Staining Percentage- percentage of pixels having that difference FOR all Nuclei
Compute Nuclei Staining Percentage
(Red Pixel count from Pixel Classified Image having intensity difference as Specified by
Stain Color Control/Area of Nuclei)
IF Nuclei Staining Percentage >= Staining Percentage
    Nuclei is stained
    Increment Stained Cell Count
ELSE
    Nuclei is not stained
    Increment Non Stained Cell Count
END IF
END FOR
```

### Results calculation
The results generated are as follows: (a) non-stained cell count, (b) stained cell count, and (c) percentage positivity, given by expression: \[ \frac{\text{Stained Cell Count}}{\text{Non-stained Cell Count} + \text{Stained Cell Count}} \].
Demonstration of web site link

The web site for the demonstration of the epithelial recognition algorithms (ERAs) has been created which will allow the reader to both view these algorithms in action and also afford an opportunity to upload images either in the form of photomicrographs or scanned images and have these images analyzed with the ERAs.

Instructions
2. Click sign up now.
3. Fill out screen.
4. In 1–2 days, reader will receive E mail that application is being reviewed.
5. In 1–2 days, reader will receive an additional E mail about logging in and setting their password.
6. After this has all been completed, reader should go to site: http://www.pathxchange.org and log in.
7. On the screen, under All Categories, go to Other and double click Epithelial % area detection using software in H&E stained breast tissue.
8. The Image Gallery depicted contains hematoxylin- and eosin-stained images of varying epithelial percentages in the left column. Pointing to the left image shows the TMA location and double clicking the image enlarges it.
9. On the right column is the corresponding ERA-generated epithelial identification that corresponds to the H&E image on the left. Pointing to the image shows the calculated epithelial percentage. Double clicking the image will allow the reader to see the demonstration of the applied epithelial recognition algorithm, the basis of the epithelial percentage calculation and the actual calculation.
10. A group of breast lesions (10 in number) with varying epithelial percentages is depicted so that the viewer can see the versatility and robustness of the epithelial recognition algorithm being applied. These areas are not a pseudocolor overlay but actually generated by the ERA which provide a “demo” version of the software.
11. In addition, examples of a false positive (#11) and a false negative (#12) analysis generated by the ERA are also depicted. Please also see the “Results” section in the main body of the paper.
12. The reader can upload a few of their own images so that the ERAs can be applied to them and the reader can view the results online. The uploaded images must be in JPEG or JPEG2 format. However, because the ERAs are housed in a separate server, which cannot be run automatically on this URL, the images that the reader sends in will be analyzed in 48 h and the analyzed image will then be uploaded to the URL so that the reader can view the analysis.
13. To upload images, click Add a Case and then under Create Case, in the Title box, type “Epithelial recognition algorithm or ERA”; check Category corresponding to tissue of origin; in Description, type “Attention: Ellen Suzue;” for Privacy, check either the Community or Public box. Do not check the Private box. The Community and Public box selections are designed to enable the running of the ERAs on the uploaded image (selecting the Private box will not enable the ERA to run); the Diagnosis box is optional; under File Attachment, browse and attach the photomicrograph or scanning file; press Attach and then Save. You may click on the image to view. You may repeat the above steps to load as many images as you would like. Then log off.
14. In 48 h, log on again, under All Categories, go to Other and double click Epithelial % area detection using software in H&E stained breast tissue. Scroll down to see your image(s). The Image Gallery depicted contains the hematoxylin- and eosin-stained images of varying epithelial percentages in the left column. Double clicking the image enlarges it. On the right column is the corresponding algorithmic generated epithelial recognition algorithm that corresponds to the H&E image on the left. Pointing to the right image shows the calculated epithelial percentage. Double clicking the image will allow the reader to see the demonstration of the applied epithelial recognition algorithm, the basis of the epithelial percentage calculation and the actual calculation on the images the user has uploaded.
Demonstration of TMAker

Figure S1. TMAker device for constructing robotic TMA. (A,B) Schematic diagram of entire device (A) and automated block image analysis and coring station (B). (C) Photograph of TMAker. [Q21]