Epithelium Pixel Width Method
By Eli Nacar

This method goes through every row of pixels in the image of the epithelium mask to determine the width of the epithelium in that row. It then places patches where they do not overlap with others until a certain percentage of the epithelium mask is covered.

There seems to be a discrepancy between the readME and the code itself.

As it stands, the readME outlines a method in which the epithelium width is used to calculate the height of any given patch, then calculates the patch width based on the total epithelium area. This ensures the patch is large enough to provide a meaningful sample of the epithelium.

The code that is supposed to execute this seems to do something slightly different. The code defines distinct patch sizes, then uses the epithelium width to fit appropriate patches to a given region. It iterates through the defined patch sizes, starting with the smallest size.

Pros:

- Computationally inexpensive
- Utilizes a lot of vectorized operations to carry out the task
- Seems to be the simplest algorithm we have

Cons:

- Does not touch on tilting the patches at all, and relies a lot on the grid of pixels
- Not a clear used algorithm
- Slice orientation

My opinion:

- I think this idea is a good one, and potentially simpler than the other stuff we've been looking at. There is the question of orientation of the slice, but in our last class, someone seemed to solve the orientation problem with a simple height and width of the image. This algorithm seems to work best on vertically oriented slices. If we can treat patching and tilting as separate tasks, this could be a good jumping off point before we tilt the patches.