Module_2:

Team Members:

Max Calcoen, Jack O'Hearn

Project Title: TBD

idea: how do blood vessel summary statistics differ across left and superior lobes?

Project Goal: TBD

idea: this project seeks to find out how the blood vessels of left and superior lobes differ

Disease Background:

Fill in information and please note that this module is truncated and only has 5 bullets (instead of the 11 that you did for Module #1).

- Prevalence & incidence (source)
    - north america / europe: ~2.8 to 9.3 per 100,000 person-years (varies by definition)
    - usa, narrow definition: ~6.8 per 100,000; broader: up to ~16.3–17.4 per 100,000
    - prevalence in u.s. estimated ~10 to 60 cases / 100,000 (rare disease threshold)
- Risk factors (genetic, lifestyle) (source)
    - older age / aging (disease primarily presents in middle to older adults)
    - male sex bias
    - cigarette smoking / smoking history (strongest environmental risk)
    - genetic predispositions: mutations in telomerase genes, surfactant genes, shorter telomeres, MUC5B promoter variant (minor allele)
    - microbiome / microbial burden in lung (higher bacterial load in BAL, certain genera like Streptococcus, Staphylococcus)
    - comorbidities / cofactors: gastroesophageal reflux disease (GERD), environmental exposures (metal dusts, silica, wood dust), viral injury, lung injury agents
- Symptoms (source)
    - gradual onset progressive (shortness of breath) on exertion
    - dry nonproductive chronic cough
    - bibasilar “velcro” crackles on auscultation (fine inspiratory crackles)
    - reduced exercise tolerance, fatigue, weight loss (constitutional)
    - pulmonary function test: restrictive pattern, decreased forced vital capacity (FVC), decreased DLCO (diffusing capacity)
- Standard of care treatment(s) (source)
    - antifibrotic therapy: Pirfenidone (slows decline in lung function, reduces fibrosis mediators)
    - antifibrotic therapy: Nintedanib (tyrosine kinase inhibitor, slows FVC decline)
    - supportive therapies: supplemental oxygen, pulmonary rehabilitation, symptom control (cough management)
    - management of comorbidities (GERD, pulmonary hypertension)
    - lung transplantation
    - clinical trials / emerging therapies (targeting profibrotic pathways, biomarkers)
- Biological mechanisms (anatomy, organ physiology, cell & molecular physiology) (source)
    - impaired epithelial regeneration / aberrant repair, with persistent activation of epithelial cells and failed re-epithelialization
    - senescence, mitochondrial dysfunction, oxidative stress in epithelial cells leading to pro-fibrotic signaling
    - release of profibrotic mediators (TGF-β, connective tissue growth factor, PDGF, fibronectin, integrins) from injured epithelium & mesenchymal cells
    - recruitment, activation and differentiation of fibroblasts → myofibroblasts → excessive ECM (extracellular matrix) deposition (collagen, fibronectin, proteoglycans)
    - mechanical stress feedback: matrix stiffening, mechanotransduction, latent TGF-ß activation via contraction of myofibroblasts
    - cross talk with immune cells / inflammation: dysregulated wound healing, low grade chronic inflammation, altered macrophage / fibrocyte responses
    - epigenetic alterations, noncoding RNAs, altered gene expression networks in fibrotic lung cells

Data-Set:

(Describe the data set(s) you will analyze. Cite the source(s) of the data. Describe how the data was collected -- What techniques were used? What units are the data measured in? Etc.)

- The dataset is composed of Unpublished data collected by the Peirce-Cottler Lab (Dept. of Biomedical Engineering) and Kim Lab (Division of Pulmonary and Critical Care) at the University of Virginia School of Medicine.
- The dataset is composed of 78 black and white images (.jpg), collected at various depths of a fibrotic mouse lung.
- White in the images symbolizes fibrotic lesion, and black symbolizes healthy lung tissue.
- The images come from a Bleomycin-induced Lung Injury Model, where an antibiotic primarily used as chemotherapy (but also causes lung fibrosis) is administered to a mouse.
- 3 weeks later, the mice were harvested.
- The mouse lung tissue is then fixed, mounted, and sliced, then fluorescent microscopy was performed.
- The mouse lungs were immunostained for 3 proteins of interest:
    - desmin (myofibroblasts)
    - smooth muscle alpha actin (large blood vessel smooth muscle cells)
    - CD-31 (endothelial cells)

Data Analyis:

(Describe how you analyzed the data. This is where you should intersperse your Python code so that anyone reading this can run your code to perform the analysis that you did, generate your figures, etc.)

First, we pulled in each image as a cv2 image object. Then, we counted each white pixel in each image. We plotted this relationship, which revealed that higher image depths have a higher percentage of white pixels. Finally, we tried removing noise from the image using contour detection. First, we tried to simply filter out dots with small contours. Then, we tried to dilate each contour to see if it could help merge borders of larger blobs (vessels) with empty space inside. Finally, we combined the methods, first dilating the contours to join edges of broken blobs and then filtering out blobs with small contours.

Below is the analysis.

data folder setup (for running code below)

```
/data
    /imaging
        MASK_Sk658 Llobe ch010017.jpg
        ...here contains all the images
    depths.csv
    Filenames and Depths (old).csv
    pct_white_pixels.csv
```

In [8]:

```python
"""FILE: get_pct_white.ipynb"""

# %%
import pandas as pd
import numpy as np
import cv2
import os
from termcolor import colored
import matplotlib.pyplot as plt

images = pd.DataFrame()

data_path = r"data"
imaging_path = r"imaging"
filenames = os.listdir(os.path.join(data_path, imaging_path))
depths = pd.read_csv(os.path.join(data_path, "depths.csv"))

for i in filenames:
    img = cv2.imread(os.path.join(data_path, imaging_path, i), 0)
    try:
        depth = depths[depths["Filenames"].str.lower() == i.lower()][
            "Depth from lung surface (in micrometers) where image was acquired"
        ].values[0]
        # some files are named with SK658 and some with Sk658
    except IndexError:
        print(f"couldn't find depth for file {i}")
        continue
    images = pd.concat(
        [images, pd.DataFrame([{"filename": i, "image": img, "depth": depth}])],
        ignore_index=True,
    )
display(images.head())

# %%
white_counts = []
black_counts = []
white_percents = []


for x in range(len(images)):
    _, binary = cv2.threshold(images.iloc[x]["image"], 127, 255, cv2.THRESH_BINARY)

    white = np.sum(binary == 255)
    black = np.sum(binary == 0)

    white_counts.append(white)
    black_counts.append(black)

# print the number of white and black pixels in each image.

print(colored("Counts of pixel by color in each image", "yellow"))
for x in range(len(images)):
    print(colored(f"White pixels in image {x}: {white_counts[x]}", "white"))
    print(colored(f"Black pixels in image {x}: {black_counts[x]}", "black"))
    print()

# calculate the percentage of pixels in each image that are white and make a list that contains these percentages for each filename

for x in range(len(images)):
    white_percent = 100 * (white_counts[x] / (black_counts[x] + white_counts[x]))
    white_percents.append(white_percent)

# print the filename (on one line in red font), and below that line print the percent white pixels and depth

print(colored("Percent white px:", "yellow"))
for x in range(len(images)):
    print(colored(f"{images.iloc[x]['filename']}:", "red"))
    print(f"{white_percents[x]}% White | Depth: {images.iloc[x]['depth']} microns")
    print()

# %%
"""Write your data to a .csv file"""

# Create a DataFrame that includes the filenames, depths, and percentage of white pixels
df = pd.DataFrame(
    {"filename": images["filename"], "depth": images["depth"], "white_percent": white_percents}
)

# Write that DataFrame to a .csv file

df.to_csv(os.path.join(data_path, "pct_white_pixels.csv"), index=False)

# %%
# display graph
plt.scatter(images["depth"], white_percents, marker="o", linestyle="-", color="blue")
plt.title("Plot of depth of image vs percentage white pixels")
plt.xlabel("depth of image")
plt.ylabel("white pixels as a percentage of total pixels")
plt.grid(True)
```

```
couldn't find depth for file MASK_Sk658 Llobe ch010053.jpg
```

	filename	image	depth
0	MASK_SK658 Slobe ch010129.jpg	[[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...	3250
1	MASK_SK658 Slobe ch010115.jpg	[[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...	8800
2	MASK_SK658 Slobe ch010114.jpg	[[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...	9900
3	MASK_SK658 Slobe ch010060.jpg	[[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...	8400
4	MASK_SK658 Slobe ch010048.jpg	[[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,...	540

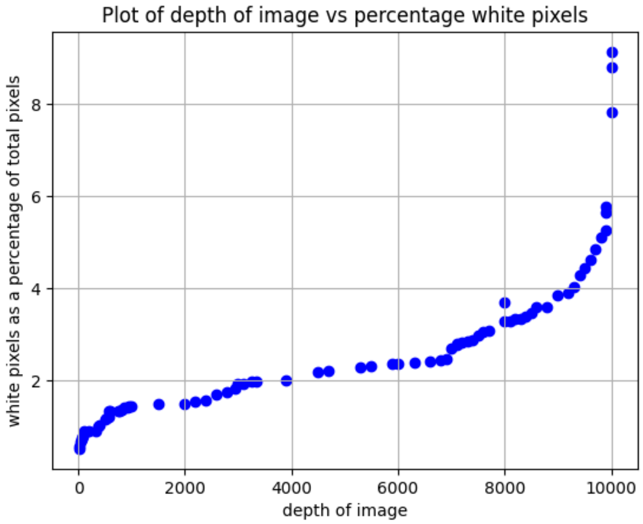
```
Counts of pixel by color in each image
White pixels in image 0: 82784
Black pixels in image 0: 4111520

White pixels in image 1: 151132
Black pixels in image 1: 4043172

White pixels in image 2: 242340
Black pixels in image 2: 3951964

White pixels in image 3: 142359
Black pixels in image 3: 4051945

White pixels in image 4: 49491
Black pixels in image 4: 4144813

White pixels in image 5: 369855
Black pixels in image 5: 3824449

White pixels in image 6: 56003
Black pixels in image 6: 4138301

White pixels in image 7: 193743
Black pixels in image 7: 4000561

White pixels in image 8: 99131
Black pixels in image 8: 4095173

White pixels in image 9: 120910
Black pixels in image 9: 4073394

White pixels in image 10: 139549
Black pixels in image 10: 4054755

White pixels in image 11: 383479
Black pixels in image 11: 3810825

White pixels in image 12: 119938
Black pixels in image 12: 4074366

White pixels in image 13: 100236
Black pixels in image 13: 4094068

White pixels in image 14: 97032
Black pixels in image 14: 4097272

White pixels in image 15: 60715
Black pixels in image 15: 4133589

White pixels in image 16: 83951
Black pixels in image 16: 4110353

White pixels in image 17: 117756
Black pixels in image 17: 4076548

White pixels in image 18: 62508
Black pixels in image 18: 4131796

White pixels in image 19: 203857
Black pixels in image 19: 3990447

White pixels in image 20: 96012
Black pixels in image 20: 4098292

White pixels in image 21: 150556
Black pixels in image 21: 4043748

White pixels in image 22: 137688
Black pixels in image 22: 4056616

White pixels in image 23: 129607
Black pixels in image 23: 4064697

White pixels in image 24: 124833
Black pixels in image 24: 4069471

White pixels in image 25: 118409
Black pixels in image 25: 4075895

White pixels in image 26: 81014
Black pixels in image 26: 4113290

White pixels in image 27: 83416
Black pixels in image 27: 4110888

White pixels in image 28: 27561
Black pixels in image 28: 4166743

White pixels in image 29: 59788
Black pixels in image 29: 4134516

White pixels in image 30: 59426
Black pixels in image 30: 4134878

White pixels in image 31: 65021
```

```
Black pixels in image 31: 4129283

White pixels in image 32: 128537
Black pixels in image 32: 4065767

White pixels in image 33: 57531
Black pixels in image 33: 4136773

White pixels in image 34: 21648
Black pixels in image 34: 4172656

White pixels in image 35: 101283
Black pixels in image 35: 4093021

White pixels in image 36: 99350
Black pixels in image 36: 4094954

White pixels in image 37: 70715
Black pixels in image 37: 4123589

White pixels in image 38: 56419
Black pixels in image 38: 4137885

White pixels in image 39: 38068
Black pixels in image 39: 4156236

White pixels in image 40: 116432
Black pixels in image 40: 4077872

White pixels in image 41: 56534
Black pixels in image 41: 4137770

White pixels in image 42: 23900
Black pixels in image 42: 4170404

White pixels in image 43: 37508
Black pixels in image 43: 4156796

White pixels in image 44: 139818
Black pixels in image 44: 4054486

White pixels in image 45: 33151
Black pixels in image 45: 4161153

White pixels in image 46: 29677
Black pixels in image 46: 4164627

White pixels in image 47: 48667
Black pixels in image 47: 4145637

White pixels in image 48: 73224
Black pixels in image 48: 4121080

White pixels in image 49: 145829
Black pixels in image 49: 4048475

White pixels in image 50: 76579
Black pixels in image 50: 4117725

White pixels in image 51: 51040
Black pixels in image 51: 4143264

White pixels in image 52: 59908
Black pixels in image 52: 4134396

White pixels in image 53: 31331
Black pixels in image 53: 4162973

White pixels in image 54: 62913
Black pixels in image 54: 4131391

White pixels in image 55: 80534
Black pixels in image 55: 4113770

White pixels in image 56: 33746
Black pixels in image 56: 4160558

White pixels in image 57: 37799
Black pixels in image 57: 4156505

White pixels in image 58: 56360
Black pixels in image 58: 4137944

White pixels in image 59: 221302
Black pixels in image 59: 3973002

White pixels in image 60: 65633
Black pixels in image 60: 4128671

White pixels in image 61: 214773
Black pixels in image 61: 3979531

White pixels in image 62: 91653
Black pixels in image 62: 4102651
```

```
White pixels in image 63: 186078
Black pixels in image 63: 4008226

White pixels in image 64: 179894
Black pixels in image 64: 4014410

White pixels in image 65: 41943
Black pixels in image 65: 4152361

White pixels in image 66: 163638
Black pixels in image 66: 4030666

White pixels in image 67: 161111
Black pixels in image 67: 4033193

White pixels in image 68: 43424
Black pixels in image 68: 4150880

White pixels in image 69: 102335
Black pixels in image 69: 4091969

White pixels in image 70: 328891
Black pixels in image 70: 3865413

White pixels in image 71: 103576
Black pixels in image 71: 4090728

White pixels in image 72: 137592
Black pixels in image 72: 4056712

White pixels in image 73: 237289
Black pixels in image 73: 3957015

White pixels in image 74: 112613
Black pixels in image 74: 4081691

White pixels in image 75: 92260
Black pixels in image 75: 4102044

White pixels in image 76: 155019
Black pixels in image 76: 4039285

White pixels in image 77: 168932
Black pixels in image 77: 4025372

Percent white px:
MASK_SK658 Slobe ch010129.jpg:
1.973724365234375% White | Depth: 3250 microns

MASK_SK658 Slobe ch010115.jpg:
3.6032676696777344% White | Depth: 8800 microns

MASK_SK658 Slobe ch010114.jpg:
5.777835845947266% White | Depth: 9900 microns

MASK_SK658 Slobe ch010060.jpg:
3.3941030502319336% White | Depth: 8400 microns

MASK_SK658 Slobe ch010048.jpg:
1.179957389831543% White | Depth: 540 microns

MASK_SK658 Slobe ch010089.jpg:
8.81803035736084% White | Depth: 10000 microns

MASK_Sk658 Llobe ch010061.jpg:
1.3352155685424805% White | Depth: 585 microns

MASK_SK658 Slobe ch010103.jpg:
4.619193077087402% White | Depth: 9600 microns

MASK_SK658 Slobe ch010077.jpg:
2.363467216491699% White | Depth: 5900 microns

MASK_SK658 Slobe ch010063.jpg:
2.882719039916992% White | Depth: 7400 microns

MASK_SK658 Slobe ch010088.jpg:
3.3271074295043945% White | Depth: 8200 microns

MASK_SK658 Slobe ch010098.jpg:
9.142851829528809% White | Depth: 10000 microns

MASK_SK658 Slobe ch010113.jpg:
2.8595447540283203% White | Depth: 7300 microns

MASK_SK658 Slobe ch010107.jpg:
2.389812469482422% White | Depth: 6300 microns

MASK_SK658 Slobe ch010112.jpg:
2.3134231567382812% White | Depth: 5500 microns

MASK_SK658 Slobe ch010066.jpg:
1.4475584030151367% White | Depth: 1000 microns

MASK_Sk658 Llobe ch010065.jpg:
```

```
2.0015478134155273% White | Depth: 3900 microns

MASK_Sk658 Llobe ch010071.jpg:
2.8075218200683594% White | Depth: 7100 microns

MASK_Sk658 Llobe ch010067.jpg:
1.4903068542480469% White | Depth: 1500 microns

MASK_SK658 Slobe ch010104.jpg:
4.860329627990723% White | Depth: 9700 microns

MASK_SK658 Slobe ch010110.jpg:
2.289104461669922% White | Depth: 5300 microns

MASK_SK658 Slobe ch010111.jpg:
3.5895347595214844% White | Depth: 8600 microns

MASK_SK658 Slobe ch010105.jpg:
3.2827377319335938% White | Depth: 8100 microns

MASK_SK658 Slobe ch010059.jpg:
3.090071678161621% White | Depth: 7700 microns

MASK_SK658 Llobe ch010149.jpg:
2.976250648498535% White | Depth: 7500 microns

MASK_SK658 Llobe ch010160.jpg:
2.8230905532836914% White | Depth: 7200 microns

MASK_SK658 Llobe ch010174.jpg:
1.9315242767333984% White | Depth: 3100 microns

MASK_SK658 Slobe ch010149.jpg:
1.9887924194335938% White | Depth: 3350 microns

MASK_SK658 Llobe ch010017.jpg:
0.6571054458618164% White | Depth: 45 microns

MASK_SK658 Slobe ch010158.jpg:
1.4254570007324219% White | Depth: 920 microns

MASK_SK658 Slobe ch010159.jpg:
1.4168262481689453% White | Depth: 860 microns

MASK_SK658 Llobe ch010164.jpg:
1.5502214431762695% White | Depth: 2200 microns

MASK_SK658 Llobe ch010159.jpg:
3.064560890197754% White | Depth: 7600 microns

MASK_SK658 Llobe ch010171.jpg:
1.3716459274291992% White | Depth: 810 microns

MASK_SK658 Llobe ch010039.jpg:
0.5161285400390625% White | Depth: 15 microns

MASK_SK658 Llobe ch010034.jpg:
2.4147748947143555% White | Depth: 6600 microns

MASK_SK658 Llobe ch010168.jpg:
2.3686885833740234% White | Depth: 6000 microns

MASK_SK658 Slobe ch010143.jpg:
1.6859769821166992% White | Depth: 2600 microns

MASK_SK658 Slobe ch010157.jpg:
1.3451337814331055% White | Depth: 750 microns

MASK_SK658 Slobe ch010156.jpg:
0.9076118469238281% White | Depth: 330 microns

MASK_SK658 Slobe ch010142.jpg:
2.7759552001953125% White | Depth: 7100 microns

MASK_SK658 Llobe ch010035.jpg:
1.3478755950927734% White | Depth: 780 microns

MASK_SK658 Llobe ch010021.jpg:
0.5698204040527344% White | Depth: 30 microns

MASK_SK658 Llobe ch010023.jpg:
0.8942604064941406% White | Depth: 100 microns

MASK_SK658 Slobe ch010140.jpg:
3.3335208892822266% White | Depth: 8300 microns

MASK_SK658 Llobe ch010022.jpg:
0.7903814315795898% White | Depth: 80 microns

MASK_SK658 Llobe ch010036.jpg:
0.707554817199707% White | Depth: 55 microns

MASK_SK658 Llobe ch010032.jpg:
1.1603116989135742% White | Depth: 500 microns
```

```
MASK_Sk658 Llobe ch010026.jpg:
1.7457962036132812% White | Depth: 2800 microns

MASK_Sk658 Llobe ch010146.jpg:
3.476834297180176% White | Depth: 8500 microns

MASK_Sk658 Llobe ch010033.jpg:
1.8257856369018555% White | Depth: 2950 microns

MASK_Sk658 Llobe ch010025.jpg:
1.216888427734375% White | Depth: 570 microns

MASK_Sk658 Llobe ch010031.jpg:
1.4283180236816406% White | Depth: 955 microns

MASK_Sk658 Llobe ch010019.jpg:
0.7469892501831055% White | Depth: 60 microns

MASK_SK658 Slobe ch010146.jpg:
1.4999628067016602% White | Depth: 2000 microns

MASK_SK658 Slobe ch010147.jpg:
1.9200801849365234% White | Depth: 3000 microns

MASK_Sk658 Llobe ch010018.jpg:
0.8045673370361328% White | Depth: 90 microns

MASK_Sk658 Llobe ch010030.jpg:
0.9011983871459961% White | Depth: 200 microns

MASK_Sk658 Llobe ch010024.jpg:
1.3437271118164062% White | Depth: 600 microns

MASK_SK658 Slobe ch010097.jpg:
5.276250839233398% White | Depth: 9900 microns

MASK_SK658 Slobe ch010083.jpg:
1.5648126602172852% White | Depth: 2400 microns

MASK_SK658 Slobe ch010068.jpg:
5.120587348937988% White | Depth: 9800 microns

MASK_SK658 Slobe ch010134.jpg:
2.185177803039551% White | Depth: 4500 microns

MASK_SK658 Slobe ch010135.jpg:
4.436445236206055% White | Depth: 9500 microns

MASK_SK658 Slobe ch010096.jpg:
4.289007186889648% White | Depth: 9400 microns

MASK_Sk658 Llobe ch010040.jpg:
0.9999990463256836% White | Depth: 380 microns

MASK_SK658 Slobe ch010136.jpg:
3.9014339447021484% White | Depth: 9200 microns

MASK_Sk658 Llobe ch010121.jpg:
3.8411855697631836% White | Depth: 9000 microns

MASK_Sk658 Llobe ch010051.jpg:
1.035308837890625% White | Depth: 400 microns

MASK_SK658 Slobe ch010126.jpg:
2.43985652923584% White | Depth: 6800 microns

MASK_SK658 Slobe ch010092.jpg:
7.841372489929199% White | Depth: 10000 microns

MASK_SK658 Slobe ch010086.jpg:
2.4694442749023438% White | Depth: 6900 microns

MASK_SK658 Slobe ch010119.jpg:
3.2804489135742188% White | Depth: 8000 microns

MASK_SK658 Slobe ch010118.jpg:
5.657410621643066% White | Depth: 9900 microns

MASK_SK658 Slobe ch010130.jpg:
2.684903144836426% White | Depth: 7000 microns

MASK_SK658 Slobe ch010078.jpg:
2.1996498107910156% White | Depth: 4700 microns

MASK_SK658 Slobe ch010087.jpg:
3.6959409713745117% White | Depth: 8000 microns

MASK_SK658 Slobe ch010093.jpg:
4.027652740478516% White | Depth: 9300 microns
```

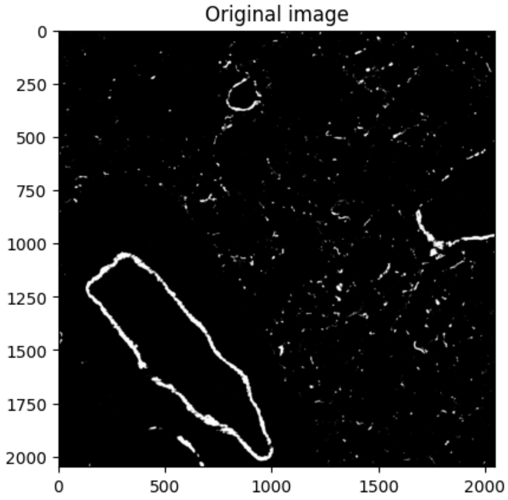


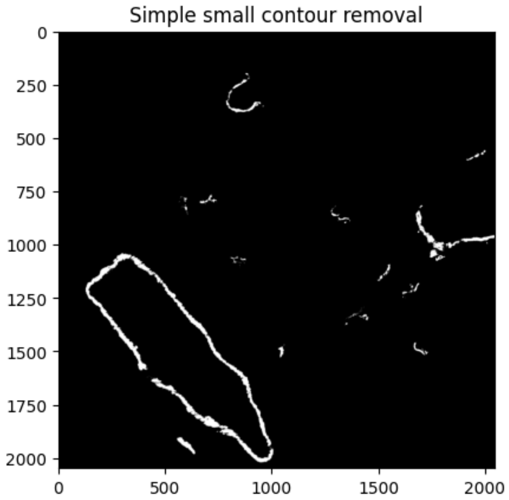
In [17]:

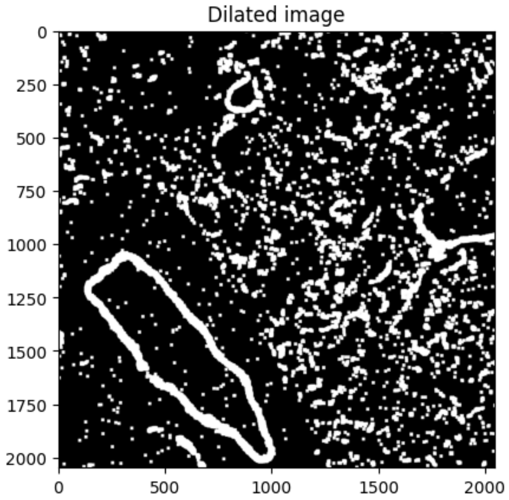
```
"""FILE: contour_detection.ipynb"""
# %%
import pandas as pd
import os
import cv2
import numpy as np
import matplotlib.pyplot as plt

images = pd.DataFrame()

data_path = r"data"
imaging_path = r"imaging"
filenames = os.listdir(os.path.join(data_path, imaging_path))
depths = pd.read_csv(os.path.join(data_path, "depths.csv"))

# display(filenames)
for i in filenames:
    img = cv2.imread(os.path.join(data_path, imaging_path, i), 0)
    try:
        depth = depths[depths["Filenames"].str.lower() == i.lower()][
            "Depth from lung surface (in micrometers) where image was acquired"
        ].values[0]
        # some files are named with SK658 and some with Sk658
    except IndexError:
        print(f"couldn't find depth for file {i}")
        continue
    images = pd.concat(
        [images, pd.DataFrame([{"filename": i, "image": img, "depth": depth}])],
        ignore_index=True,
    )

print(images.shape)

# %%
print(images["filename"])

# %%
img = (
    images[images["filename"] == "MASK_Sk658 Llobe ch010034.jpg"]["image"]
    .values[0]
    .copy()
)

# convert to rgb instead of grayscale
plt.imshow(img, cmap="gray")
plt.title("Original image")
plt.show()

# %%
# remove small contours

img_contour_simple = img.copy()

contours, _ = cv2.findContours(
    img_contour_simple, cv2.RETR_EXTERNAL, cv2.CHAIN_APPROX_SIMPLE
)

for cntr in contours:
    if cv2.contourArea(cntr) > 1000:
        continue
    convHull = cv2.convexHull(cntr)
    cv2.drawContours(
        img_contour_simple, [convHull], -1, (0, 0, 0), thickness=cv2.FILLED
    )
img_contour_simple = cv2.cvtColor(img_contour_simple, cv2.COLOR_GRAY2RGB)
plt.imshow(img_contour_simple)
plt.title("Simple small contour removal")
```

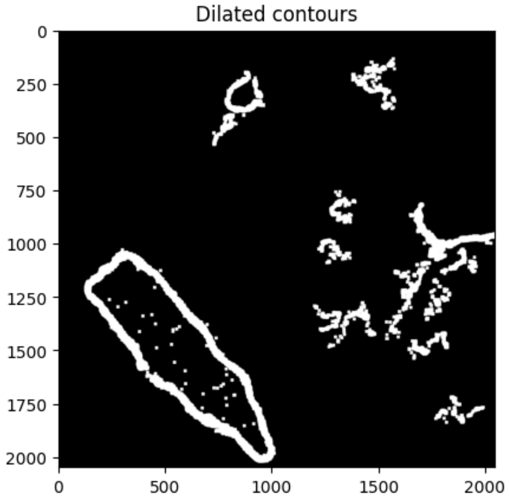
```python
plt.show()

# %%
img_dilate = img.copy()
img_dilate = cv2.dilate(img_dilate, np.ones((15, 15), np.uint8), iterations=1)

plt.imshow(img_dilate, cmap="gray")
plt.title("Dilated image")
plt.show()

# %%
# look at contours on dilated image
img_dilated_contour = img_dilate.copy()
contours, _ = cv2.findContours(
    img_dilated_contour, cv2.RETR_EXTERNAL, cv2.CHAIN_APPROX_SIMPLE
)

for cntr in contours:
    if cv2.contourArea(cntr) > 10000:
        continue
    convHull = cv2.convexHull(cntr)
    cv2.drawContours(
        img_dilated_contour, [convHull], -1, (0, 0, 0), thickness=cv2.FILLED
    )
img_dilated_contour = cv2.cvtColor(img_dilated_contour, cv2.COLOR_GRAY2RGB)
plt.imshow(img_dilated_contour)
plt.title("Dilated contours")
plt.show()

# %%
# combine
img_combine = img_contour_simple.copy()
img_combine = cv2.dilate(img_combine, np.ones((15, 15), np.uint8), iterations=1)
# Convert to grayscale before thresholding
img_combine_gray = cv2.cvtColor(img_combine, cv2.COLOR_RGB2GRAY)
_, img_combine_thresh = cv2.threshold(img_combine_gray, 128, 255, cv2.THRESH_BINARY | cv2.THRESH_OTSU)
contours, _ = cv2.findContours(img_combine_thresh, cv2.RETR_EXTERNAL, cv2.CHAIN_APPROX_SIMPLE)

for cntr in contours:
    if cv2.contourArea(cntr) > 50000:
        continue
    convHull = cv2.convexHull(cntr)
    cv2.drawContours(img_combine_thresh, [convHull], -1, (0, 0, 0), thickness=cv2.FILLED)
img_combine = cv2.cvtColor(img_combine_thresh, cv2.COLOR_GRAY2RGB)
plt.imshow(img_combine)
plt.title("Combined: removed small contours then dilated and\n removed small contours (with larger threshold) again")
plt.show()
```

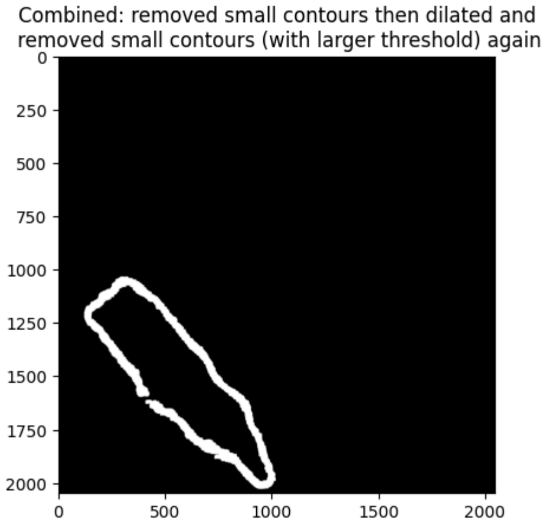
```
couldn't find depth for file MASK_Sk658 Llobe ch010053.jpg
(78, 3)
0     MASK_SK658 Slobe ch010129.jpg
1     MASK_SK658 Slobe ch010115.jpg
2     MASK_SK658 Slobe ch010114.jpg
3     MASK_SK658 Slobe ch010060.jpg
4     MASK_SK658 Slobe ch010048.jpg
                  ...
73    MASK_SK658 Slobe ch010118.jpg
74    MASK_SK658 Slobe ch010130.jpg
75    MASK_SK658 Slobe ch010078.jpg
76    MASK_SK658 Slobe ch010087.jpg
77    MASK_SK658 Slobe ch010093.jpg
Name: filename, Length: 78, dtype: object
```




Simple small contour removal
0
250
500
750
1000
1250
1500
1750
2000
0
500
1000
1500
2000


Dilated image
0
250
500
750
1000
1250
1500
1750
2000
0
500
1000
1500
2000


Dilated contours
0
250
500
750
1000
1250
1500
1750
2000
0
500
1000
1500
2000



Combined: removed small contours then dilated and removed small contours (with larger threshold) again

Verify and validate your analysis:

(Describe how you checked to see that your analysis gave you an answer that you believe (verify). Describe how your determined if your analysis gave you an answer that is supported by other evidence (e.g., a published paper).

Conclusions and Ethical Implications:

(Think about the answer your analysis generated, draw conclusions related to your overarching question, and discuss the ethical implications of your conclusions.

Limitations and Future Work:

(Think about the answer your analysis generated, draw conclusions related to your overarching question, and discuss the ethical implications of your conclusions.

NOTES FROM YOUR TEAM:

This is where our team is taking notes and recording activity.

none.

QUESTIONS FOR YOUR TA:

These are questions we have for our TA.

none.