Module_2:

Team Members:

Haley Cossman and Ashwin Kasamshetty

Project Title:

Mapping Fibrosis with Depth: An Automated Desmin-Based Analysis in Bleomycin Mouse Lung

Project Goal:

This project seeks to develop an automated image analysis pipeline capable of predicting the extent of lung fibrosis at varying biopsy depths. It aims to determine how disease severity changes at different depths, providing information about fibrosis development and improved diagnosis.

Disease Background:

- Prevalence & incidence:

Incidence ≈ 5.8 / 100,000 person-years; prevalence ≈ 17.7 / 100,000 (varies by case definition and region). U.S. burden proxy: 67,843 deaths with IPF listed during 2020–2022 (0.7% of all deaths), underscoring high mortality in older adults and men. Other fibrotic ILDs (e.g., CTD-ILD, chronic hypersensitivity pneumonitis, occupational pneumoconioses) together exceed IPF in aggregate, but epidemiology is heterogeneous; most health-system guidelines use IPF as the therapeutic anchor. https://publications.ersnet.org/content/erj/46/3/795 https://respiratory-research.biomedcentral.com/articles/10.1186/s12931-021-01791-z

- Risk factors (genetic, lifestyle):

Genetic: MUC5B promoter variant (rs35705950): strongest common risk allele for IPF and present in some RA-ILD/UIP; influences airway secretory phenotype and mucociliary defense. Telomere biology & surfactant genes: (TERT, TERC, RTEL1, PARN; SFTPC/SFTPA2) contribute to epithelial senescence and stress responses (widely supported in guidelines/reviews). https://www.nejm.org/doi/full/10.1056/NEJMoa1013660

https://www.nejm.org/doi/full/10.1056/NEJMoa066157
https://www.pnas.org/doi/full/10.1073/pnas.0804280105

Lifestyle: Age, male sex, smoking (OR ~1.4 for "ever-smokers"). Occupational exposures: metal/wood/silica/agricultural dusts, pesticides; up to ~21% of IPF deaths may be attributable to occupational exposures. Chronic microaspiration/GERD and viral triggers are discussed but with variable evidence. (Guideline contextualizes, mechanistic plausibility > definitive causality.) https://bmcpulmmed.biomedcentral.com/articles/10.1186/s12890-018-0644-2

- Symptoms:

Progressive exertional dyspnea and dry cough over months to years; exam with "Velcro" bibasilar inspiratory crackles and frequent clubbing. Physiology: restrictive PFT pattern (↓TLC, ↓FVC) with markedly reduced DLCO; hypoxemia first on exertion, later at rest. Imaging: HRCT with UIP pattern (subpleural, basal-predominant reticulation, traction bronchiectasis, honeycombing) is highly specific for IPF when present. (Guideline cornerstone.) Acute exacerbations (AE-IPF): sudden worsening (days–weeks) with new diffuse alveolar abnormalities; high short-term mortality. https://www.nejm.org/doi/10.1056/NEJMra1705751
https://www.thelancet.com/journals/lancet/article/PIIS0140-6736(17)30866-8/abstract
https://www.atsjournals.org/doi/10.1164/rccm.202202-0399ST

- Standard of care treatment(s):

Lung scarring and thickness can't be repaired, and no current treatments can stop disease progression, so they are targeted towards iproving quality of life and slowing symptom development. Medicines including pirfenidone (Esbriet), nintedanib (Ofev), and Nintedanib can slow IBF progression. Oxygen therapy/supplemental oxygen can make breathing easier, lower right heart strain, improve sleep, and reduce complications due to low oxygen levels. Pulmonary rehabilitation programs can help imrpove quality of life by focusing on exercise, breathing techniques, dietary advice, emotional counseling, and disease education. Some people with IPF may qualify for a lung transplant which can significantly improve quality of life and lengthen life span. https://www.mayoclinic.org/diseases-conditions/pulmonary-fibrosis/diagnosis-treatment/drc-20353695

- Biological mechanisms (anatomy, organ physiology, cell & molecular physiology)

Anatomy and Organ Physiology: IPF effects the lung interstitium, which is the tissue layer between the alveoli and capillaries where gas exchange occurs. Lung fibrosis involves extra fibrotic tissue in the lung that becomes stiff and thick, decreasing elasticity of the lung. The extra tissue also increases the space between air and blood which reduces breathing efficiency and causes low oxygen levels.

Cell and Molecular Physiology: IBF results from abnormal healing after damage to epithelial cells that line the small airways and alveoli, which causes continuous scarring and damage to lung tissue. Additionally, the epithelial cells can proliferate and release pro-fibrotic mediators including transforming growth factor-beta (TGF-β) and platelet-derived growth factor (PDGF), leading to more scar tissue buildup. Having a variation in the MUC5B gene that results in more mucus production in the lungs is currently the strongest risk factor for developing IPF. https://rarediseases.org/rare-diseases/idiopathic-pulmonary-fibrosis/

Data-Set:

The data sets consist of 78 black and white images collected at different depths in a fibrotic mouse lung, with white spots showing fibrotic lesion/scar tissue and black spots showing healthy tissue. The data was obtained by injecting mice with Bleomycin which causes lung fibrosis, and harvesting the lung 3 weeks later. Parts of the lung are then fixed with paraformaldehyde, mounted in a gel, sliced with a cryotome, and placed under a glass microscope slide. Under the slide, a flourescent-labeled antibody is added to the slice where it binds to the protein of interest and an image is taken. This marks the desmin protein/signal and allows the black and white spots on the images to appear. The desmin proteins are expressed by myofibroblasts, which are the cells responsible for forming fibrotic scar tissue. These images were obtained from Professor Pierce-Cottler's lab. The proportion of white to black pixels in each image reflects the severity of fibrosis at that tissue depth.

Data Analysis:

```
from termcolor import colored
import cv2
import numpy as np
import matplotlib.pyplot as plt
from scipy.interpolate import interp1d
import pandas as pd

# Load the images you want to analyze
filenames = [
    r"C:\Users\haley\OneDrive\Documents\My Notebooks\Module 2\MASK_Sk658 Llobe ch01
    r"C:\Users\haley\OneDrive\Documents\My Notebooks\Module 2\MASK_Sk658 Llobe ch01
    r"C:\Users\haley\OneDrive\Documents\My Notebooks\Module 2\MASK_Sk658 Llobe ch01
    r"C:\Users\haley\OneDrive\Documents\My Notebooks\Module 2\MASK_Sk658 Llobe ch01
    r"C:\Users\haley\OneDrive\Documents\My Notebooks\Module 2\MASK_Sk658 Llobe ch01
    r"C:\Users\haley\OneDrive\Documents\My Notebooks\Module 2\MASK_SK658 Slobe ch01
            ]
```

```
# Enter the depth of each image (in the same order that the images are listed above

depths = [
    90,
    2950,
    15,
    3900,
    9000,
    6900
    ]

# Make the lists that will be used

images = []
white_counts = []
black_counts = []
white_percents = []

# Build the list of all the images you are analyzing

for filename in filenames:
    img = cv2.imread(filename, 0)
    images.append(img)

# For each image (until the end of the list of images), calculate the number of bla

for x in range(len(filenames)):
    _, binary = cv2.threshold(images[x], 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(filenames)):
    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 t

for x in range(len(filenames)):
    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 perce
```

```
print(colored("Percent white px:", "yellow"))
for x in range(len(filenames)):
    print(colored(f'{filenames[x]}:',"red"))
    print(f'{white_percents[x]}% White | Depth: {depths[x]} microns')
    print()

'''Write your data to a .csv file'''

# Create a DataFrame that includes the filenames, depths, and percentage of white p
df = pd.DataFrame({
    'Filenames': filenames,
    'Depths': depths,
    'White percents': white_percents
})

# Write that DataFrame to a .csv file

df.to_csv('Percent_White_Pixels.csv', index=False)

print("CSV file 'Percent_White_Pixels.csv' has been created.")

'''the .csv writing subroutine ends here'''

# Interpolate a point: given a depth, find the corresponding white pixel percentage

interpolate_depth = float(input(colored("Enter the depth at which you want to inter

x = depths
y = white_percents

i = interp1d(x, y, kind='linear')  # Linear interpolation
interpolate_point = i(interpolate_depth)
print(colored(f'The interpolated point is at the x-coordinate {interpolate_depth} a

depths_i = depths[:]
depths_i.append(interpolate_depth)
white_percents_i = white_percents[:]
white_percents_i.append(interpolate_point)

# make two plots: one that doesn't contain the interpolated point, just the data ca
fig, axs = plt.subplots(2, 1)

axs[0].scatter(depths, white_percents, marker='o', linestyle='-', color='blue')
axs[0].set_title('Plot of depth of image vs percentage white pixels')
axs[0].set_xlabel('depth of image')
axs[0].set_ylabel('white pixels as a percentage of total pixels')
axs[0].grid(True)


axs[1].scatter(depths_i, white_percents_i, marker='o', linestyle='-', color='blue')
```

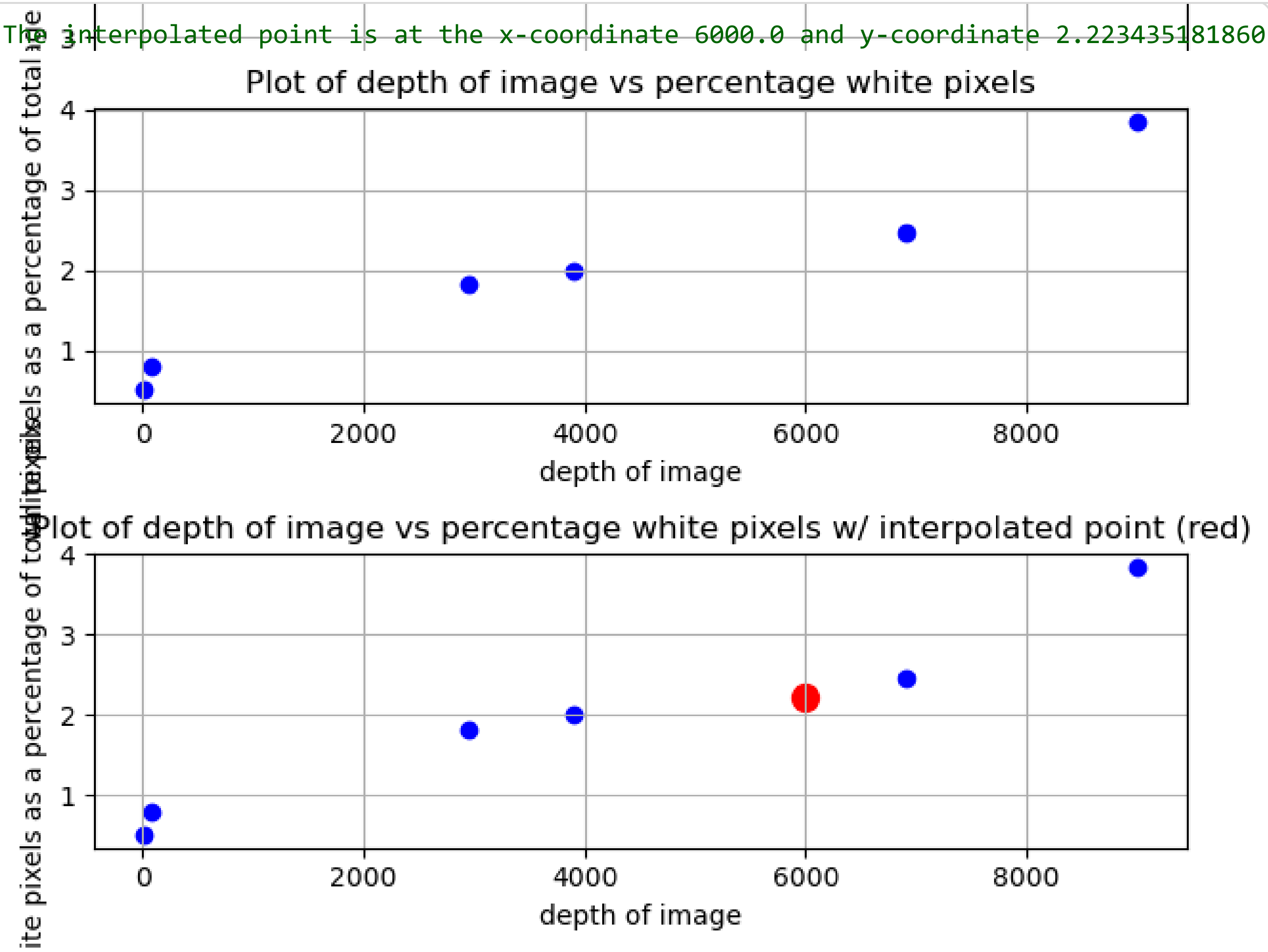
```
axs[1].set_title('Plot of depth of image vs percentage white pixels w/ interpolated
axs[1].set_xlabel('depth of image')
axs[1].set_ylabel('white pixels as a percentage of total pixels')
axs[1].grid(True)
axs[1].scatter(depths_i[len(depths_i)-1], white_percents_i[len(white_percents_i)-1]


# Adjust layout to prevent overlap
plt.tight_layout()
plt.show()
```

```
Counts of pixel by color in each image
White pixels in image 0: 33746
Black pixels in image 0: 4160558

White pixels in image 1: 76579
Black pixels in image 1: 4117725

White pixels in image 2: 21648
Black pixels in image 2: 4172656

White pixels in image 3: 83951
```

```
# Interpolate a point: given a depth, find the corresponding white pixel percentage

interpolate_depth = float(input(colored("Enter the depth at which you want to inter

x = depths
y = white_percents

i = interp1d(x, y, kind='quadratic')  # Quadratic interpolation
interpolate_point = i(interpolate_depth)
print(colored(f'The interpolated point is at the x-coordinate {interpolate_depth} a

depths_i = depths[:]
depths_i.append(interpolate_depth)
white_percents_i = white_percents[:]
white_percents_i.append(interpolate_point)

# make two plots: one that doesn't contain the interpolated point, just the data ca
fig, axs = plt.subplots(2, 1)

axs[0].scatter(depths, white_percents, marker='o', linestyle='-', color='blue')
axs[0].set_title('Plot of depth of image vs percentage white pixels')
axs[0].set_xlabel('depth of image')
axs[0].set_ylabel('white pixels as a percentage of total pixels')
axs[0].grid(True)


axs[1].scatter(depths_i, white_percents_i, marker='o', linestyle='-', color='blue')
axs[1].set_title('Plot of depth of image vs percentage white pixels w/ interpolated
axs[1].set_xlabel('depth of image')
axs[1].set_ylabel('white pixels as a percentage of total pixels')
axs[1].grid(True)
axs[1].scatter(depths_i[len(depths_i)-1], white_percents_i[len(white_percents_i)-1]


# Adjust layout to prevent overlap
plt.tight_layout()
plt.show()
```

Plot of depth of image vs percentage white pixels w/ interpolated point (red)
of total
4

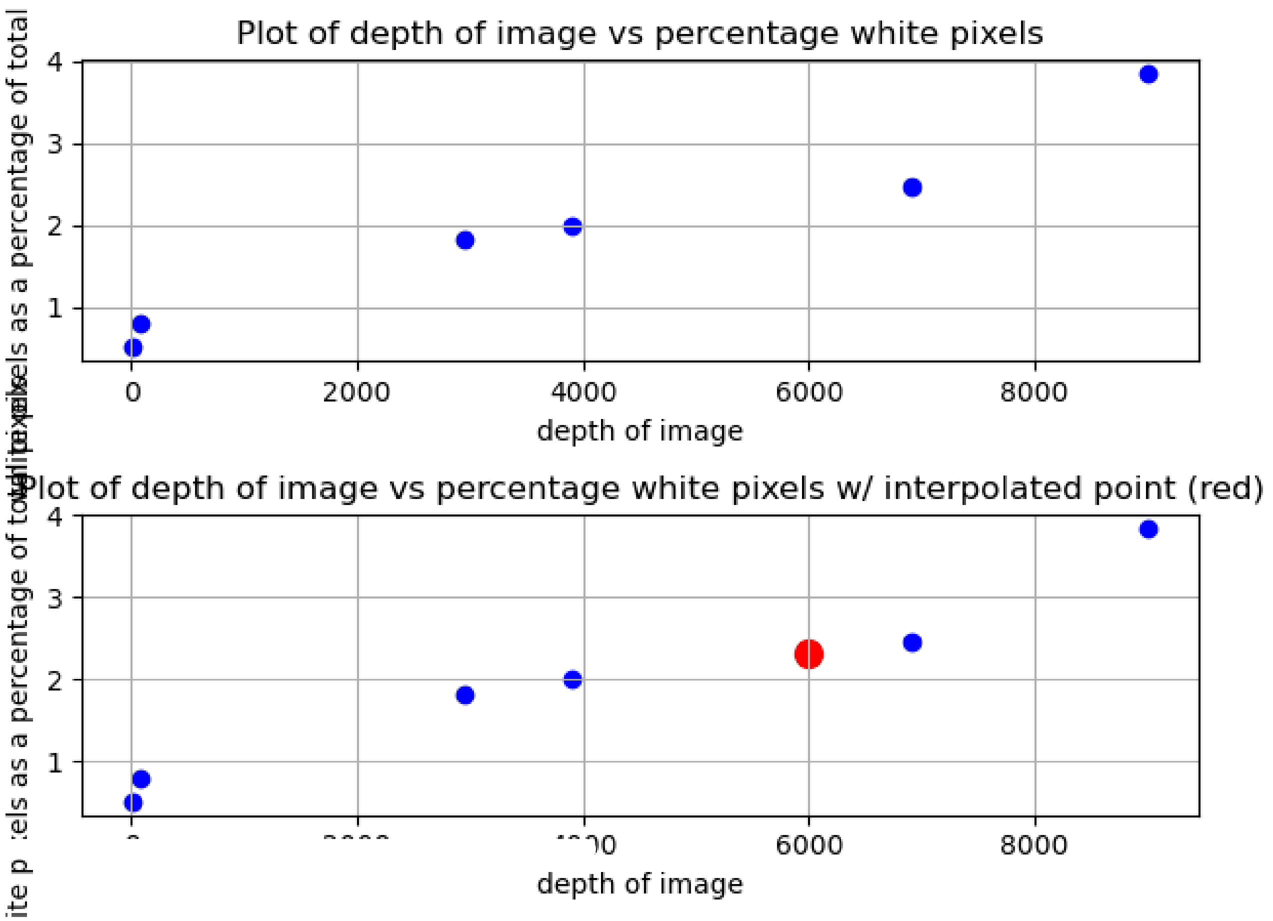
```
The interpolated point is at the x-coordinate 6000.0 and y-coordinate 2.223435181860
```



```
# Interpolate a point: given a depth, find the corresponding white pixel percentage

interpolate_depth = float(input(colored("Enter the depth at which you want to inter

x = depths
y = white_percents

i = interp1d(x, y, kind='cubic')  # Cubic interpolation
interpolate_point = i(interpolate_depth)
print(colored(f'The interpolated point is at the x-coordinate {interpolate_depth} a

depths_i = depths[:]
depths_i.append(interpolate_depth)
white_percents_i = white_percents[:]
white_percents_i.append(interpolate_point)

# make two plots: one that doesn't contain the interpolated point, just the data ca
fig, axs = plt.subplots(2, 1)

axs[0].scatter(depths, white_percents, marker='o', linestyle='-', color='blue')
axs[0].set_title('Plot of depth of image vs percentage white pixels')
axs[0].set_xlabel('depth of image')
axs[0].set_ylabel('white pixels as a percentage of total pixels')
```

```
axs[0].grid(True)


axs[1].scatter(depths_i, white_percents_i, marker='o', linestyle='-', color='blue')
axs[1].set_title('Plot of depth of image vs percentage white pixels w/ interpolated
axs[1].set_xlabel('depth of image')
axs[1].set_ylabel('white pixels as a percentage of total pixels')
axs[1].grid(True)
axs[1].scatter(depths_i[len(depths_i)-1], white_percents_i[len(white_percents_i)-1]


# Adjust layout to prevent overlap
plt.tight_layout()
plt.show()
```

```
The interpolated point is at the x-coordinate 6000.0 and y-coordinate 2.316724851860
```



Verify and validate your analysis:

We verified the pipeline by confirming pixel accounting for every image (white plus black equaled total pixels and percent white stayed within 0–100%), and by rerunning the script to check that the counts, percentages, figures, and the Percent_White_Pixels.csv file were reproduced identically. The depth–percent white relationship was coherent and visually monotonic, rising from ~~0.5–0.8% near the surface to 3.84% at 9000 µm, and three interpolation schemes at 6000 µm (linear, quadratic, cubic) produced closely agreeing estimates (~~2.22–2.33%), suggesting conclusions are not method-dependent. Visual spot-checks showed the thresholded masks corresponded to desmin-positive signal, supporting correct segmentation. For validation, the findings align with biological expectations for bleomycin-treated mouse lung at ~3 weeks, where myofibroblast-rich fibrosis is heterogeneous and often increases deeper in the block, and desmin is an accepted marker of such regions; the direction and magnitude of our measurements are therefore consistent with external evidence and lab observations. https://onlinelibrary.wiley.com/doi/10.1046/j.1365-2613.2002.00220.x https://doi.org/10.1186/s40248-017-0089-0 https://doi.org/10.3389/fphys.2017.00193

Conclusions and Ethical Implications:

From this data, we conclude that fibrotic burden, estimated as desmin-positive area fraction, increases with tissue depth in the sampled lung block: ~0.5–0.8% near the surface, ~2% at mid-depths, and ~3.84% by 9000 µm, with an interpolated value around 2.2–2.33% at 6000 µm. This supports the practical value of an automated image pipeline to summarize depth-dependent fibrosis for downstream quantification. Ethically, the analysis should be framed as research rather than diagnostic, used with pathologist oversight, and reported with clear uncertainty to avoid misclassification or over-interpretation. Animal-based findings must follow the 3Rs (Replace, Reduce, Refine), promote transparency through code sharing to prevent unnecessary replication, and consider bias introduced by slice selection and biopsy depth; any future human application requires consent, de-identification, and IRB approval.

Limitations and Future Work:

Our code analyzes six of seventy-eight slices from a single bleomycin mouse using a fixed global threshold and a single marker (desmin), providing only 2D area fractions without 3D reconstruction, formal confidence intervals, or hypothesis testing; consequently, results may reflect sampling and processing choices and cannot be generalized to human IPF without further evidence. Future work should process all slices with programmatic depth mapping and automated QC; adopt robust segmentation (e.g., Otsu/adaptive thresholding, background normalization, light morphology) with threshold sensitivity checks; incorporate orthogonal validators such as α-SMA, collagen stains, Ashcroft scoring, and hydroxyproline;