

## **Module\_2:**

### **Team Members:**

Kayla Chin and Alex Kremsreiter

### **Project Title:**

Deep Dive into Fibrosis: Predicting Lung Scarring by Depth

### **Project Goal:**

This project seeks to better understand how the severity of fibrosis changes throughout a fibrotic lung. From this, the goal is to use interpolation strategies to predict the amount of fibrosis when given a specified depth.

### **Disease Background:**

- Prevalence & incidence
  - Idiopathic pulmonary fibrosis has an estimated prevalence of 13 to 20 per 100,000 people worldwide
  - In US, 30,000 to 40,000 new cases are diagnosed each year
- Risk factors (genetic, lifestyle)
  - While the main cause of IPF is unknown, there are certain condition that can exacerbate symptoms and make you more prone to the disease
  - genetic mutations in TERC and TERP have been found in about 15% of familial cases (not idiopathic)
    - TERC and TERP are related to the development of telomerase, without them functioning properly it is suspected that lung tissue/ cells die prematurely leading to the fibrotic tissue
  - lifestyle contributions that are often seen in cases of IFP are exposure to wood/ metal dust, viral infections, and cigarette smoking.
- Symptoms
  - shortness of breath and persistent dry/ hacking cough are the most common symptoms
  - may also effect appetite leading to gradual weight loss
  - scarred lungs are the diagnosing feature of IFP
    - develops to the point in which lungs can no longer provide oxygen to the rest of the body

- can lead to lung cancer, blood clots, and pneumonia
- Standard of care treatment(s)
  - There are no current universal treatments/ cures for fibrosis due to its intricate nature in the lungs
  - Medications that attempt to improve quality of life and decrease preliminary symptoms are pirfenidone (Esbriet) or nintedanib (Ofev).
    - attempt to slow the progress of fibrotic tissue development
  - Due to its relationship with GERD, certain anti-acid medications may be prescribed in order to lessen complementary symptoms.
- Biological mechanisms (anatomy, organ physiology, cell & molecular physiology)
  - The primary effected mechanism is the pulmonary system, however the immune system also plays a major role in how fibrosis develops/ grows.
    - dysregulation of macrophages leads to the release of pro-fibrotic factors such as TGF- $\beta$  and PDGF.
  - Cell proliferation and increase in growth factors also leads to the build up of tissue in the lungs

## Work Cited

- NIH - Idiopathic pulmonary fibrosis
- Mayo Clinic - Pulmonary Fibrosis
- NIH - Interpreting Immunoregulation in Lung Fibrosis: A New Branch of the Immune Model
- BMC - Pulmonary fibrosis: from mechanisms to therapies

## Data-Set:

- Bleomycin was used to induce fibrosis in the lungs of mice and model IPF from Pierce-Cottler Lab
- The lungs were then harvested and sectioned transversely
  - gives multiple different verticle depths of the lung tissue
  - allows for studying how the different sample depths result in varying amount of fibrotic lung tissue
- Slides were then imgaged (78 in total)
  - white: fibrotic lesions
  - black: health lung
- Because this is a comparative model, it is important to note that these conditions are not identical to human IPF and there may be differences
  - Mouse lungs have horizontal orientation, while human's have vertical which could effect how these results are applied.

## Data Analysis:

Our team began by loading a series of lung tissue images at various depths with various amount of fibrosis at each depth. Then used python to count the number of black and white pixels in each image, and calculates the percentage of white pixels, which represent fibrotic tissue. The results include a table with the image file names with its associated depth and percentage of white pixels, this is then saved to a CSV file. Finally, the code plots the relationship between lung depth and percent white pixels to visualize fibrosis trends across tissue layers.

```
In [6]: ### '''Module 2: count black and white pixels in a .jpg and extrapolate points'''

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\akrem\OneDrive\Desktop\Computational BME\Project 2\MASK_Sk658 Llobe",
    r"C:\Users\akrem\OneDrive\Desktop\Computational BME\Project 2\MASK_Sk658 Llobe",
    r"C:\Users\akrem\OneDrive\Desktop\Computational BME\Project 2\MASK_SK658 Slobe",
    r"C:\Users\akrem\OneDrive\Desktop\Computational BME\Project 2\MASK_SK658 Slobe",
    r"C:\Users\akrem\OneDrive\Desktop\Computational BME\Project 2\MASK_SK658 Slobe",
    r"C:\Users\akrem\OneDrive\Desktop\Computational BME\Project 2\MASK_SK658 Slobe"
]

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

depths = [
    90,
    7500,
    5900,
    8000,
    7300,
    920
]

# Make the lists that will be used
# === Initialize lists ===
images = []
white_counts = []
black_counts = []
white_percents = []

# === Load all images ===
for filename in filenames:
    img = cv2.imread(filename, 0)
    if img is None:
        print(colored(f"Warning: Could not read image {filename}", "red"))
    images.append(img)
```

```

# === Count black and white pixels in each image ===
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 individual pixel counts ===
print(colored("\nCounts of pixels by color in each image:", "yellow"))
for x in range(len(filenames)):
    print(colored(f"Image {x+1}: {filenames[x]}", "red"))
    print(colored(f"White pixels: {white_counts[x]}", "white"))
    print(colored(f"Black pixels: {black_counts[x]}\n", "cyan"))

# === Calculate % white pixels ===
for x in range(len(filenames)):
    white_percent = 100 * (white_counts[x] / (black_counts[x] + white_counts[x]))
    white_percents.append(white_percent)

# === Print a clean summary table ===
print(colored("\nSummary Table", "yellow"))
print("=" * 65)
print(f"{'Image Name':<35} {'Depth (\mu m)':>12} {'% White Pixels':>15}")
print("-" * 65)
for i in range(len(filenames)):
    short_name = filenames[i].split("\\")[-1]
    print(f"{short_name:<35} {depths[i]:>12} {white_percents[i]:>15.2f}")
print("=" * 65)

# === Write results to CSV ===
df = pd.DataFrame({
    'Filenames': filenames,
    'Depths': depths,
    'White percents': white_percents
})
df.to_csv('Percent_White_Pixels.csv', index=False)
print(colored("\nCSV file 'Percent_White_Pixels.csv' has been created.", "green"))

# === Plot results ===
fig, ax = plt.subplots()
ax.scatter(depths, white_percents, color='blue')
ax.set_title('Depth vs % White Pixels')
ax.set_xlabel('Depth (\mu m)')
ax.set_ylabel('% White Pixels')
ax.grid(True)

plt.tight_layout()
plt.show()

```

Counts of pixels by color in each image:

Image 1: C:\Users\akrem\OneDrive\Desktop\Computational BME\Project 2\MASK\_Sk658 Llobe ch010018.jpg  
White pixels: 33746  
Black pixels: 4160558

Image 2: C:\Users\akrem\OneDrive\Desktop\Computational BME\Project 2\MASK\_Sk658 Llobe ch010149.jpg  
White pixels: 124833  
Black pixels: 4069471

Image 3: C:\Users\akrem\OneDrive\Desktop\Computational BME\Project 2\MASK\_SK658 Slobe ch010077.jpg  
White pixels: 99131  
Black pixels: 4095173

Image 4: C:\Users\akrem\OneDrive\Desktop\Computational BME\Project 2\MASK\_SK658 Slobe ch010087.jpg  
White pixels: 155019  
Black pixels: 4039285

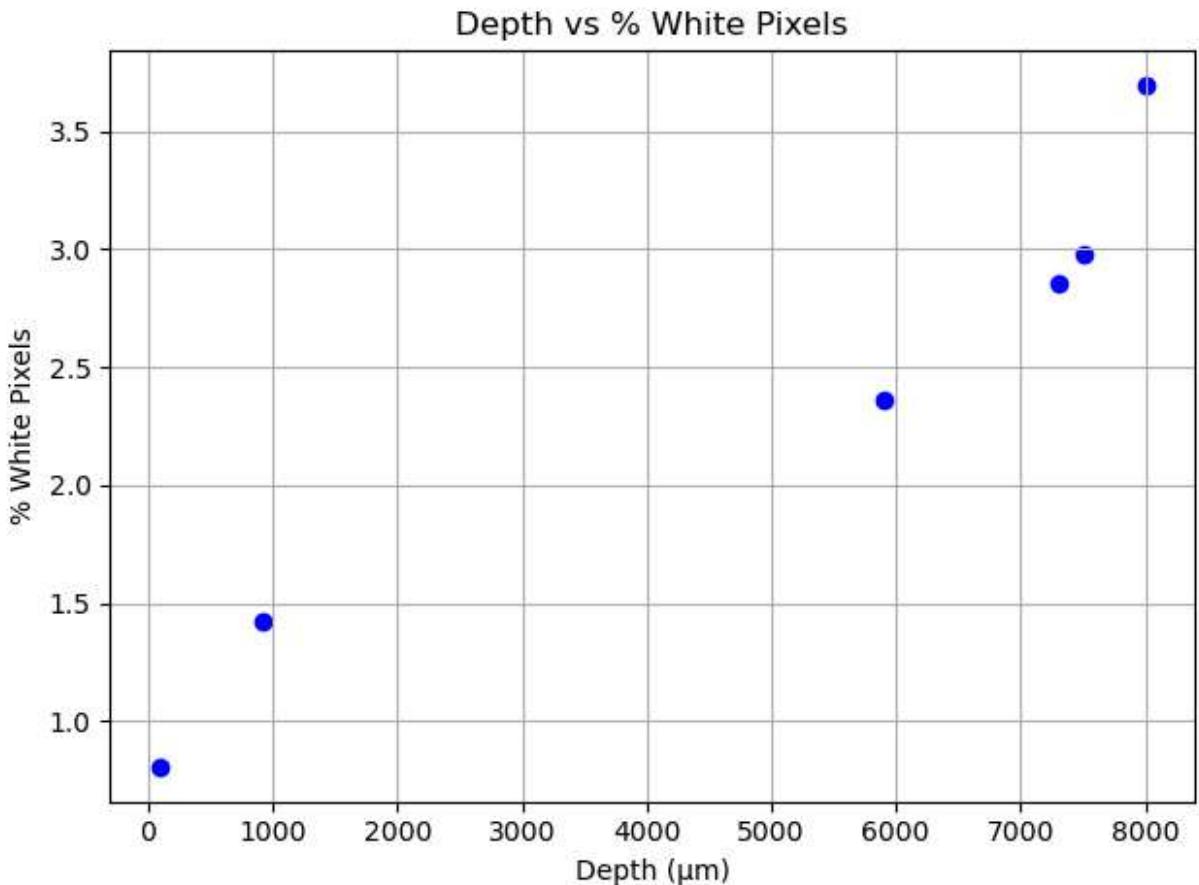
Image 5: C:\Users\akrem\OneDrive\Desktop\Computational BME\Project 2\MASK\_SK658 Slobe ch010113.jpg  
White pixels: 119938  
Black pixels: 4074366

Image 6: C:\Users\akrem\OneDrive\Desktop\Computational BME\Project 2\MASK\_SK658 Slobe ch010158.jpg  
White pixels: 59788  
Black pixels: 4134516

#### Summary Table

Image Name	Depth ( $\mu\text{m}$ )	% White Pixels
MASK_Sk658 Llobe ch010018.jpg	90	0.80
MASK_Sk658 Llobe ch010149.jpg	7500	2.98
MASK_SK658 Slobe ch010077.jpg	5900	2.36
MASK_SK658 Slobe ch010087.jpg	8000	3.70
MASK_Sk658 Slobe ch010113.jpg	7300	2.86
MASK_SK658 Slobe ch010158.jpg	920	1.43

CSV file 'Percent\_White\_Pixels.csv' has been created.



Our team then performed a linear interpolation to estimate the percentage of white pixels (fibrotic tissue) at any specified lung depth not directly measured within the dataset. The plot fits a linear regression model to the six images from before. The interpolation prediction calculates the interpolated value from surrounding data points. The program then displays two plots: the first shows the original linear trend of the data and the second shows the user-specified interpolated point to visualize how it fits in with the rest of the data points.

```
In [18]: # Interpolate a point: given a depth, find the corresponding white pixel percentage

import numpy as np
from scipy.interpolate import interp1d
from scipy import stats
from termcolor import colored
import matplotlib.pyplot as plt

# === Sort data by depth (important for interpolation) ===
x, y = zip(*sorted(zip(depths, white_percents)))

# === Interpolate user-provided depth ===
interpolate_depth = float(input(colored("Enter the depth at which you want to inter
i = interp1d(x, y, kind='linear', fill_value='extrapolate')
interpolate_point = float(i(interpolate_depth))
print(colored(f"The interpolated point is at x = {interpolate_depth:.2f} μm and y ="

# === Extend data for plotting ===
depths_i = list(x) + [interpolate_depth]
```

```

white_percents_i = list(y) + [interpolate_point]

# === Create subplots ===
fig, axs = plt.subplots(2, 1, figsize=(7, 10))

# --- First plot: original data ---
slope, intercept = np.polyfit(x, y, 1)
regression_line = slope * np.array(x) + intercept
r_squared_orig = stats.pearsonr(x, y)[0] ** 2

axs[0].scatter(x, y, color='blue', label='Data')
axs[0].plot(x, regression_line, color='red', label=f'y = {slope:.4f}x + {intercept:.4f}')
axs[0].set_title('Depth vs % White Pixels (Original Data)')
axs[0].set_xlabel('Depth (\mu m)')
axs[0].set_ylabel('% White Pixels')
axs[0].grid(True)
axs[0].legend()

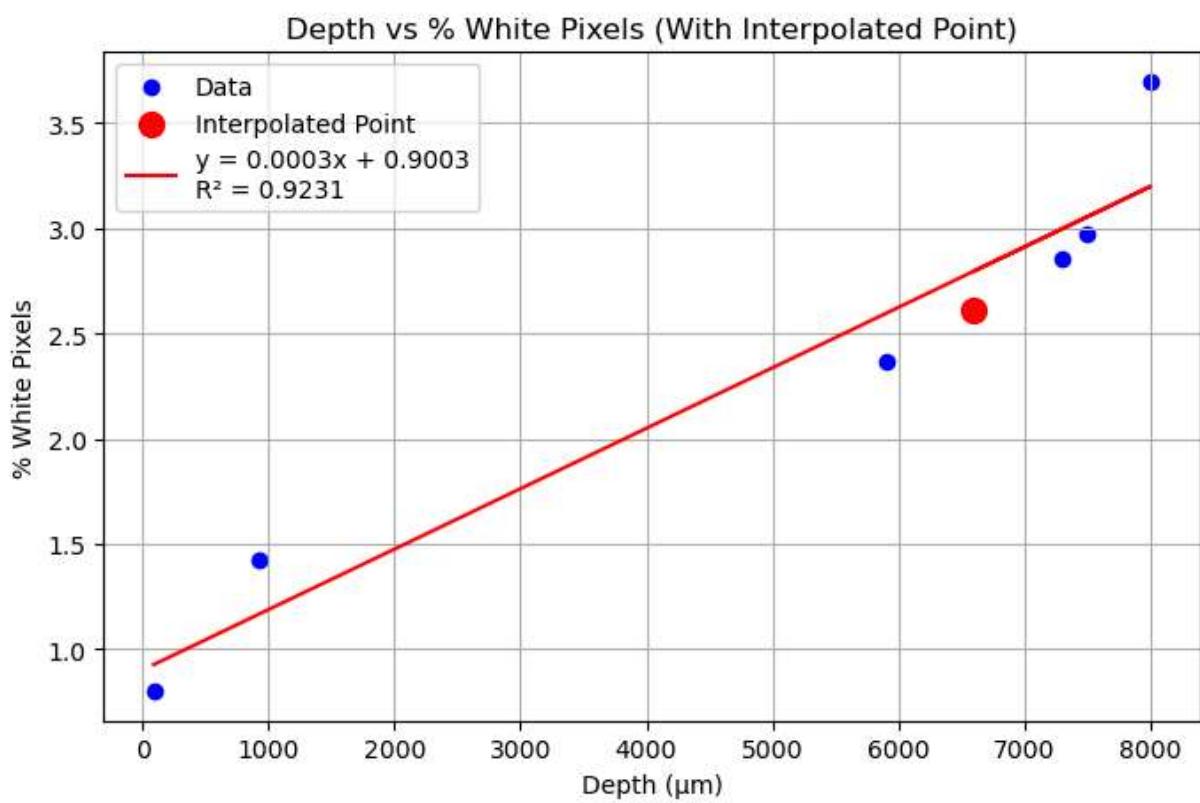
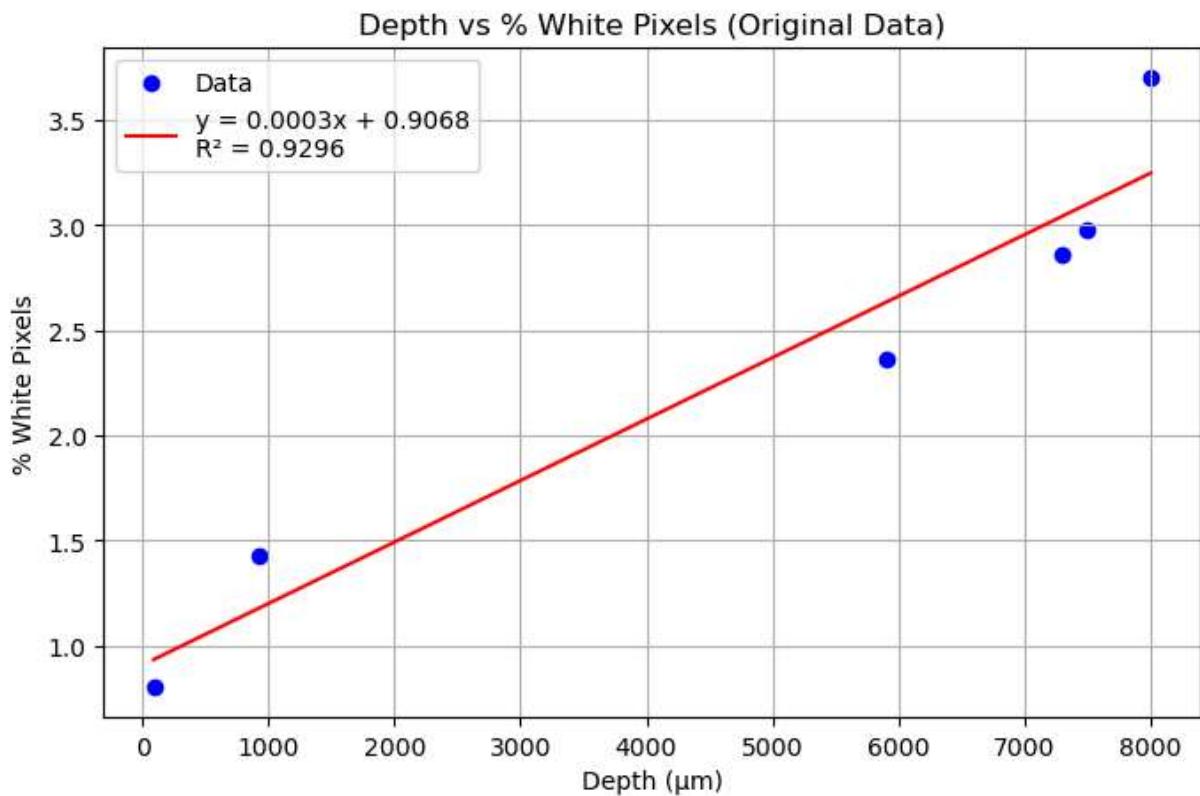
# --- Second plot: including interpolated point ---
slope_i, intercept_i = np.polyfit(depths_i, white_percents_i, 1)
regression_line_i = slope_i * np.array(depths_i) + intercept_i
r_squared_i = stats.pearsonr(depths_i, white_percents_i)[0] ** 2

axs[1].scatter(x, y, color='blue', label='Data')
axs[1].scatter(interpolate_depth, interpolate_point, color='red', s=100, label='Interpolated Point')
axs[1].plot(depths_i, regression_line_i, color='red', label=f'y = {slope_i:.4f}x + {intercept_i:.4f}')
axs[1].set_title('Depth vs % White Pixels (With Interpolated Point)')
axs[1].set_xlabel('Depth (\mu m)')
axs[1].set_ylabel('% White Pixels')
axs[1].grid(True)
axs[1].legend()

plt.tight_layout()
plt.subplots_adjust(hspace=0.4)
plt.show()

```

The interpolated point is at  $x = 6600.00 \mu\text{m}$  and  $y = 2.61\%$ .



Verify and validate your analysis:

To ensure image analysis was performed correctly, our team inspected the plot of percent white pixels versus lung depth. The positive linear trend ( $R^2 \approx 0.93$ ) indicates that the fraction of white pixels, which represents fibrotic tissue, increases as depth increases. This is consistent with the hypothesis that there is an increase in fibrosis in the deeper lung regions.

When our team interpolated the percent white pixel values at unmeasured depths using a linear methods, the estimated values closely followed the trend defined by the set data points. For instance, at a depth of 6600  $\mu\text{m}$ , the linear interpolation predicted ~2.61% white pixels. This is fairly consistent with the result given by running the same file at 6600 $\mu\text{m}$  which reads ~2.41% white pixels; this shows a 8.3% decrease from the interpolation prediction to the python reading. At a depth of 500 $\mu\text{m}$ , the interpolation prediction reads ~1.11%, while the python prediction predicts ~1.16%. This shows a 4.5% decrease from the interpolation prediction to the python reading. This suggests that the interpolation behaves reasonably and the data is fairly consistent.

## Conclusions and Ethical Implications:

- Given the data, we concluded that an increase in sample depth within a fibrotic lung, more fibrotic tissue would be found (increase in white pixels percentage)
  - We are also concluding that there is a linear relationship between the depth of the image and white pixels, allowing us to interpolate based on the linear regression line.
- From this conclusion a biopsy medical device should be able to collect samples deeper within the lung would be necessary to accurately find the full extent in which fibrotic tissue exists in a give IPF patient.
- The ethical implication of this conclusion effects how a diagnostic tool would be used
  - because we believe that the deeper the lung tissue, the more fibrotic it is the device would have to biopsy deeper within a patients lung and could be more invasive.

## Limitations and Future Work:

- The primary limitation of this conclusion is that it is a comparative model (mice lung's) and therefore there is the risk of translating it into human application.
- Future work can look towards obtaining human samples and seeing if these trends continue, if so then the development of this device would be of higher need and could more accurately help human cases.

## NOTES FROM YOUR TEAM:

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

## **QUESTIONS FOR YOUR TA:**

*Our team does not have any notes for the TA at this time.*