

Module_2: Lung Fibrotic Disease

Team Members:

Nolan Nguyen & Gabby Holohan

Project Title:

Predicting the Extent of Lung Fibrosis at Different Biospy Depths via Interpolation

Project Goal:

This project seeks to develop an image analysis pipeline to predict the extent of lung fibrosis at different biopsy depths from the top of the lung.

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
 - Globally, the prevalence and incidence vary widely across demographic regions. Incidence estimates in Asia-Pacific countries were between .35 to 1.3 per 10,000 people, 0.09 to .49 in Europe, and .75 to .93 in North America.
 - Estimates for prevalence were between .57 to 4.51 in Asia-Pacific countries, .33 to 2.51 in European countries, and 2.4 to 2.98 in North America.
 - Considered a rare disease in all countries studied besides South Korea
 - <https://pubmed.ncbi.nlm.nih.gov/34233665/>
 - Prevalence in the USA is between .14 and .279 per 10,000 using narrow definitions and .427 to .63 using broad definitions.
 - Prevalence in Europe is between .125 and .234
 - Annual incidence in the USA is between .68 and .88 using narrow definitions and .163 and .174 using broad definitions
 - Annual incidence in Europe is between .022 and .74
 - Prevalence and incidence increase with age, are higher in males, and seem to be increasing in recent years.
 - <https://pubmed.ncbi.nlm.nih.gov/23204124/>
- Risk factors (genetic, lifestyle)
 - Age: risk of IPF increases with age; diagnosed most often in patients aged 60-80
 - Smoking is a common risk factor
 - IPF is more common in men
 - Genetics: Mutations in genes responsible for producing surfactant and mucus, such as MUC5B, increases risk for IPF
 - MUC5B makes a mucus protein responsible for clearing bacteria and other substances from the lung

- Mutations in TERT and TERC, genes that produce telomerase (an enzyme protecting DNA during replication), are also more prevalent in those with IPF
- More likely to get IPF if a first-degree relative has IPF
- <https://www.nhlbi.nih.gov/health/idiopathic-pulmonary-fibrosis/causes>
- Ephysema
- Working in mining, farming, or construction or exposure to pollutants that damage the lungs
- Cancer treatments like radiation treatments or some chemotherapy medications
- <https://www.mayoclinic.org/diseases-conditions/pulmonary-fibrosis/symptoms-causes/syc-20353690>
- Symptoms
 - Shortness of breath
 - Dry cough
 - Extreme tiredness
 - Unintended weightloss
 - Aching muscles and joints
 - Clubbing: widening/rounding of fingertips and toes
 - Severity may increase rapidly over time
 - Rapid progression of symptoms, shortness of breath specifically, is called acute exacerbation. Acute exacerbation can be life-threatening.
 - Related complications: pulmonary hypertension, right-sided heart failure, respiratory failure, lung cancer, and other lung problems (blood clots, collapsed lung, lung infections, etc)
 - <https://www.mayoclinic.org/diseases-conditions/pulmonary-fibrosis/symptoms-causes/syc-20353690>
- Standard of care treatment(s)
 - There is no cure for lung fibrosis, but treatments can slow progression and improve quality of life.
 - FDA-approved anti-fibrotic drugs: pirfenidone and nintedanib slow decline in lung function for idiopathic pulmonary fibrosis (IPF).
 - Supportive therapies include supplemental oxygen, pulmonary rehabilitation, symptom management, and prevention/management of complications (such as vaccinations and prompt treatment of infections).
 - In eligible patients with advanced disease, lung transplantation is considered.
 - Corticosteroids and immunosuppressants may be used in some non-IPF interstitial lung disease but are not effective for IPF and can be harmful if misapplied.
 - <https://www.sciencedirect.com/science/article/pii/S2772558825000052>
- Biological mechanisms (anatomy, organ physiology, cell & molecular physiology)
 - Lung fibrosis is characterized by the accumulation of extracellular matrix (mainly collagen), leading to thickening and stiffening of the alveolar walls.
 - Injury to the alveolar epithelium triggers chronic, dysregulated repair responses involving the activation and proliferation of fibroblasts and differentiation into myofibroblasts.

- Myofibroblasts secrete excessive collagen and other matrix proteins, which replace normal lung tissue and reduce lung compliance.
- Dysregulated signaling pathways (e.g., TGF- β , PDGF, and connective tissue growth factor) play key roles; abnormal epithelial-mesenchymal interactions perpetuate fibrosis.
- Loss of normal lung architecture impairs gas exchange, decreases vital capacity and compliance, and leads to progressive respiratory dysfunction.
- <https://pmc.ncbi.nlm.nih.gov/articles/PMC2675823/>

Data-Set:

Unpublished data was 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 data consists of 78 black and white images at different depths of a fibrotic mouse lung. Bleomycin, an antibiotic primarily used to treat cancer that causes lung fibrosis, is injected into the mice's tracheas. After three weeks, the fibrotic mouse lungs are harvested to be biopsied. The lung was fixed, mounted in gel, and sliced with a microtome. Three immunostains, desmin, smooth muscle alpha actin, and CD-31, were added to each lung sample to identify myofibroblasts, large blood vessel smooth muscle cells, and endothelial cells respectively. The samples were placed under a microscope, capturing an image and the desmin signal was converted to a black and white image. The white parts of this image represent fibrotic lesions. The depths of the slices were measured in micrometers.
<https://uvahealth.com/findadoctor/John-Kim-1407155682>

Data Analysis:

(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, generate your .csv file, etc.)

How Data Was Analyzed

Data Inspection & Understanding: We began by loading all image files and their corresponding depth values from a CSV, ensuring matched filenames and correct metadata association.

Image Analysis: For each image, the algorithm thresholds pixel values to distinguish between white (fibrosis) and black (background), then counts these pixels to quantify tissue areas.

Metric Calculation: The percentage of white pixels per image is computed as a quantitative fibrosis metric, paired to each biopsy depth.

Visualization & Inquiry: We plotted percentage white pixels versus biopsy depth for the full set and a 6-point subset, then used interpolation to estimate fibrosis for arbitrary depths.

Question Formation & Interpretation: By visualizing both raw and interpolated data, we were able to ask and answer: "How does fibrosis extent vary with lung depth?". Allowing us to demonstrate effective dataset interrogation and scientific reasoning.

```
from termcolor import colored
import cv2
```

```

import numpy as np
import matplotlib.pyplot as plt
from scipy.interpolate import interp1d
from scipy import stats
import pandas as pd
import os

# Read images from your folder
def read_images_from_folder(folder_path, extensions=['.jpg', '.jpeg', '.png']):
    files = []
    for file in os.listdir(folder_path):
        if any(file.lower().endswith(ext) for ext in extensions):
            files.append(os.path.join(folder_path, file))
    files.sort()
    return files

# Custom function to read your CSV file with full file paths and depths
# Each line format: full_path_to_file,depth
def read_depths_from_custom_csv(filepath):
    df = pd.read_csv(filepath)
    # Extract just the basename for filename matching
    df['Filename'] = df['FileNames'].apply(lambda x:
os.path.basename(x.strip()))
    # Rename the depth column to a simple name
    df = df.rename(columns={'Depth from lung surface (in micrometers)
where image was acquired': 'Depth'})
    return df[['Filename', 'Depth']]

folder_path = r"C:\Users\NTNgu\Desktop\Module #2\Depth Images"
depths_csv_path = r"C:\Users\NTNgu\Desktop\Module #2\FileNames and
Depths for Students.csv"

# Read images and depths DataFrame
image_files = read_images_from_folder(folder_path)
depths_df = read_depths_from_custom_csv(depths_csv_path)

# Match each image to its depth by filename (basename)
matched_files = []
matched_depths = []

for img_path in image_files:
    basename = os.path.basename(img_path).strip().lower()
    match = depths_df[depths_df['Filename'].str.strip().str.lower() ==
basename]
    if not match.empty:
        matched_files.append(img_path)
        matched_depths.append(match['Depth'].values[0])

```

```

# Prepare lists
images = []
white_counts = []
black_counts = []
white_percents = []

# Build the list of all the images you are analyzing
for filename in matched_files:
    img = cv2.imread(filename, 0)
    images.append(img)

# For each image, calculate the number of black and white pixels
for x in range(len(matched_files)):
    _, 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 pixel counts
print(colored("Counts of pixel by color in each image", "yellow"))
for x in range(len(matched_files)):
    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 percentage of white pixels in each image
for x in range(len(matched_files)):
    white_percent = (100 * (white_counts[x] / (black_counts[x] +
white_counts[x])))
    white_percents.append(white_percent)

# Print filename, percent white pixels, and depth
print(colored("Percent white px:", "yellow"))
for x in range(len(matched_files)):
    print(colored(f'{matched_files[x]}:', "red"))
    print(f'{white_percents[x]}% White | Depth: {matched_depths[x]}
microns')
    print()

# Create a DataFrame and save CSV
df = pd.DataFrame({
    'FileNames': matched_files,
    'Depths': matched_depths,
    'White percents': white_percents
})
df.to_csv('Percent_White_Pixels.csv', index=False)
print("CSV file 'Percent_White_Pixels.csv' has been created.")

```

```

# Ask user for interpolation depth
interpolate_depth = float(input(colored("Enter the depth at which you
want to interpolate a point: ", "yellow")))

# Full data interpolation
i_full = interp1d(matched_depths, white_percents, kind='linear')
interpolate_point_full = i_full(interpolate_depth)
print(colored(f'The interpolated point (full 78 images) is at depth
{interpolate_depth} microns: {interpolate_point_full}', "green"))

# Subset interpolation with 6 points
subset_indices = [0, 1, 2, 3, 4, 5]
subset_depths = [matched_depths[i] for i in subset_indices]
subset_white_percents = [white_percents[i] for i in subset_indices]

i_subset = interp1d(subset_depths, subset_white_percents,
kind='linear')
interpolate_point_subset = i_subset(interpolate_depth)
print(colored(f'Interpolated point (subset of 6) at depth
{interpolate_depth} microns: {interpolate_point_subset}', "cyan"))

# Append interpolation points
depths_full_i = matched_depths[:]
depths_full_i.append(interpolate_depth)
white_percents_full_i = white_percents[:]
white_percents_full_i.append(interpolate_point_full)

subset_depths_i = subset_depths[:]
subset_depths_i.append(interpolate_depth)
subset_white_percents_i = subset_white_percents[:]
subset_white_percents_i.append(interpolate_point_subset)

# Increase subplot count to 6 for new sixth plot
fig, axs = plt.subplots(6, 1, figsize=(8, 24))

# 1. All images no interpolation
axs[0].scatter(matched_depths, white_percents, marker='o',
linestyle='-', color='blue')
axs[0].set_title('All Images: Depth of image vs Percent white pixels')
axs[0].set_xlabel('Depth (microns)')
axs[0].set_ylabel('Percent White Pixels')
axs[0].grid(True)

# 2. All images with interpolated point
axs[1].scatter(matched_depths, white_percents, marker='o',
linestyle='-', color='blue')
axs[1].scatter(interpolate_depth, interpolate_point_full, color='red',
s=100, label='Interpolated Point')
axs[1].set_title('All Images with Interpolated Point (red)')

```

```

axs[1].set_xlabel('Depth (microns)')
axs[1].set_ylabel('Percent White Pixels')
axs[1].grid(True)
axs[1].legend()

# 3. Subset (6 points) no interpolation
axs[2].scatter(subset_depths, subset_white_percents, marker='o',
               linestyle='--', color='green')
axs[2].set_title('Subset of 6 Images: Depth vs Percent white pixels')
axs[2].set_xlabel('Depth (microns)')
axs[2].set_ylabel('Percent White Pixels')
axs[2].grid(True)

# 4. Subset + interpolated point
axs[3].scatter(subset_depths_i, subset_white_percents_i, marker='o',
               linestyle='--', color='green')
axs[3].scatter(interpolate_depth, interpolate_point_subset,
               color='red', s=100, label='Interpolated Point')
axs[3].set_title('Subset + Interpolated Point (red)')
axs[3].set_xlabel('Depth (microns)')
axs[3].set_ylabel('Percent White Pixels')
axs[3].grid(True)
axs[3].legend()

# 5. All 78 images with linear regression
axs[4].scatter(matched_depths, white_percents, color='blue',
               marker='o', label='Data Points')
slope_all, intercept_all, r_value_all, p_value_all, std_err_all =
stats.linregress(matched_depths, white_percents)
x_line_all = np.linspace(min(matched_depths), max(matched_depths),
                           100)
y_line_all = slope_all * x_line_all + intercept_all
axs[4].plot(x_line_all, y_line_all, color='red', linewidth=2,
            label=f'Linear Fit (R²={r_value_all**2:.3f})')
axs[4].set_title('All 78 Images: Depth vs Percent White Pixels with
Linear Regression')
axs[4].set_xlabel('Depth (microns)')
axs[4].set_ylabel('Percent White Pixels')
axs[4].grid(True)
axs[4].legend()

# 6. Subset of 6 images with linear regression
axs[5].scatter(subset_depths, subset_white_percents, color='green',
               marker='o', label='Subset Points')
slope_sub, intercept_sub, r_value_sub, p_value_sub, std_err_sub =
stats.linregress(subset_depths, subset_white_percents)
x_line_sub = np.linspace(min(subset_depths), max(subset_depths), 100)
y_line_sub = slope_sub * x_line_sub + intercept_sub
axs[5].plot(x_line_sub, y_line_sub, color='red', linewidth=2,
            label=f'Linear Fit (R²={r_value_sub**2:.3f})')

```

```

axs[5].set_title('Subset of 6 Images: Depth vs Percent White Pixels
with Linear Regression')
axs[5].set_xlabel('Depth (microns)')
axs[5].set_ylabel('Percent White Pixels')
axs[5].grid(True)
axs[5].legend()

# Print regression outputs for both full and subset data
print(colored("Linear Regression Results (All 78 Images):",
"magenta"))
print(colored(f"Slope: {slope_all:.4f}", "magenta"))
print(colored(f"Intercept: {intercept_all:.4f}", "magenta"))
print(colored(f"R-squared: {r_value_all**2:.4f}", "magenta"))
print(colored(f"P-value: {p_value_all:.4e}", "magenta"))

print(colored("Linear Regression Results (Subset of 6 Images):",
"cyan"))
print(colored(f"Slope: {slope_sub:.4f}", "cyan"))
print(colored(f"Intercept: {intercept_sub:.4f}", "cyan"))
print(colored(f"R-squared: {r_value_sub**2:.4f}", "cyan"))
print(colored(f"P-value: {p_value_sub:.4e}", "cyan"))

plt.tight_layout()
plt.show()

```

Verify and validate your analysis:

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

We verified our analysis by carefully matching each of the 78 lung biopsy images with their respective depths using the filenames and depth metadata. We checked that the pixel counting algorithm ran consistently and produced expected counts across images. The continuity and smoothness of the interpolation curves were assessed visually to ensure no outlier or erroneous data points compromised the interpolation function.

To validate our results against existing evidence, we compared the trends we observed—specifically, the increase in white pixel percentage with biopsy depth as a surrogate for fibrosis progression—to findings in published literature. A prior study in pulmonary fibrosis pathology and imaging (<https://pmc.ncbi.nlm.nih.gov/articles/PMC9872080/>) reports similar relationships between biopsy depth and fibrotic tissue presence. It states that fibrotic changes are "more widespread in the subpleural region of the lower lobes," This agreement supports the biological plausibility and reliability of our image-based quantification and interpolation approach.*

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.

We found that the percent of white pixels, representing extent of pulmonary fibrosis, increased with depth. Therefore, our data suggest that the extent of fibrosis is directly related to the depth of the sample from the top of the lung: the extent of fibrosis is greater in deeper pulmonary regions. With this conclusion, a medical device constructed to evaluate pulmonary fibrosis at different biopsies must have precise and accurate depth perception and must be able to collect biopsy samples at greater depths of the lung.

Some ethical implications of this finding involve equity in healthcare, diagnoses procedures, and responsibility for patients' wellbeing. Medical devices or procedures involving more invasive technology may pose a health risk to patients both in clinical trials and as treatment. Additionally, invasive procedures may be more expensive and therefore less accessible to patients. This inequity may result in diagnoses gaps and underestimating the severity of IPF if less invasive technology collects biopsies from shallower regions of the lung. Healthcare providers must be aware that biopsy samples from shallower regions may not accurately predict the progression of IPF.

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.)

Our data were gathered from biopsies of mice lungs, which differ in size and orientation from human lungs. Mouse lungs are much smaller than human lungs, and mice are quadrupedal while humans are bipedal, potentially impacting the extent of fibrosis in different regions of the lung. Therefore, we cannot be entirely certain that our findings apply to the human lung until human samples are collected and analyzed.

Future work may include running human biopsy samples through our program to better predict degree of fibrosis at different lung depths in humans or altering the computational model to perform interpolations with other shapes such as quadratic or cubic. The reasoning behind an increasing degree of fibrosis in deeper regions of the lung is another question delivered by our data analysis. Researching why this is the case could further aid medical device design and potential treatment for IPF.

NOTES FROM YOUR TEAM:

- 10/7/2025 - Both team members worked to debug code for creating the .csv file and interpolating data points.
- 10/9/2025 - Both team members explored ways to expand on the current project model (including all data points instead of 6, messing with different curves of best fit)
- 10/14/2025 - Library meeting to plan presentation and sort out jupyter notebook logistics. Divided responsibilities for slides on the presentation: Analysis and Verification (Nolan), Goal, Background and Ethical Implications (Gabby). For notebook logistics:
- Nolan: finishing analysis code for 78 images, ensure output plots are accurate, finish verification and validation.
- Gabby: finish ethical implications, conclusions, future work and limitations.

QUESTIONS FOR YOUR TA:

No questions at this time.