Module_2:

Team Members:

Yaseen Noori and Marielle Miranda

Project Title:

From Image to Insight: Interpolative Prediction of Fibrotic Severity Across Lung Depths

Project Goal:

This project seeks to *develop an image analysis pipline that will predict the extent of fibrosis in the lung at different biopsy depths from the top of the lung*

Two Main Questions:
- How does the extent of fibrosis vary throughout the depth of a fibrtic lung?
- Using interpolation, can you predict the amount of fibrosis at a specific depth into 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: *In the United States there are around 140,000 patients living with idiopathic lung fibrosis and around 30,000 to 50,000 new cases diagnosed annually. Furthermore, worldwide, it is reported that 17.7 per 100,000 people are affected with idiopathic lung fibrosis and there are around 5.8 new cases per 100,000 people a year.*
  - https://medlineplus.gov/genetics/condition/idiopathic-pulmonary-fibrosis/#:~:text=Frequency,appear%20to%20run%20in%20families.
  - https://pmc.ncbi.nlm.nih.gov/articles/PMC12330001/#:~:text=Pooled%20global%20incidence%20(5.8%20per compared%20with%20Europe%20and%20Asia.
- Risk factors (genetic, lifestyle): *GENETIC factors that can lead to idiopathic lung fibrosis is a MUC5B gene variant (affects mucus clearing in lungs), surfactant genes (affects lung stability, such as SFTPC and SFTA1/2), telomere-related genes (can cause cell aging and poor lung repair. i.e. TERT, TERC, RTEL1), and family history (having family members with idiopathic lung fibrosis can increase risk). LIFESTYLE choices that can increase of getting idiopathic lung fibrosis are smoking, exposure to dust/fume, acid reflux/GERD (tiny amounts of acid that enter the lungs may cause injuries), viral infections (may trigger fibrosis), older age, and being a male (men have a greater risk of getting idiopathic lung fibrosis compared to women)*
  - ChatGPT Prompt: "What are the risk factors (genetic and lifestyle) of idiopathic lung fibrosis?
- Symptoms: *Common symptoms for idiopathic lung fibrosis are the following: fatigue, difficulty breathing, persistent dry cough, and crackling sounds in the lungs (also known as "rails").*

  - https://youtu.be/s2owdwl5Vac?si=7ZRDmlUugWh6iE28
- Standard of care treatment(s): *Common standard of care treatments for idiopathic lung fibrosis are antifibrotic medications, such as Nintedainb and Pirfenidone, which helps slow the disease progression by decreasing the formation of scar tissue through targeting fibroblasts and fibrocytes and interfering with signaling pathways. Another one is lifestyle adjustments, such as exercising and smoking cessation. If the progression of the disease is advanced, lung transplant may be an option or a mechanical ventilation.*
  - https://www.mayoclinic.org/diseases-conditions/pulmonary-fibrosis/diagnosis-treatment/drc-20353695#:~:text=Physical%20exercise%20to%20improve%20how,treatment%20option%20for%20your%20condition.
- Biological mechanisms (anatomy, organ physiology, cell & molecular physiology): *ANATOMY: Mainly, idiopathic lung fibrosis causes damage to the interstitial tissue (the thin layer between the alveoli and capillaries.) Over time, this tissue will thicken in the lower and outer regions of the lung (subpleural and basal). Furthermore, as the disease progresses, the normal lung structure becomes replaced with scar tissue and cystic spaces which create a "honeycomb" look and reduces the overall size and elasticity of the lungs. ORGAN PHYSIOLOGY: With idiopathic lung fibrosis, the scar tissue causes the lungs to become stiff and less elastic, which makes it harder for the lung to expand when breathing. Furthermore, gas exchange becomes inefficient due to oxygen having to cross thickened and scared walls. Overall, this results in low blood oxygen, shortness of breath, and fatigue. If the progression gets worse, this may lead to the lung capacity decreasing and causing respiratory failure. CELL AND MOLECULAR PHYSIOLOGY: At the cellular level, idiopathic lung fibrosis is a result of abnormal wound healing in the alveoli after repeated microscopic injury. The damaged alveolar epithelial cells then release signals, such as TGF-β, PDGF, and IL-13, which activate fibroblasts to produce excess collagen and scar tissue. Furthermore, due to idiopathic lung fibrosis, fibroblasts will keep multiplying while epithelial cells fail to regenerate. This leads to continuous fibrosis. Additionally, genetic factors (i.e. shortened telomeres and mitochondrial dysfunction) can lead to increased oxidative stress and can cause lung cells to be more vulnerable, while abnormal blood vessel growth and immune activation can further worsen the scarring process.*
  - ChatGPT Prompt: "What are the biological mechanisms, such as anatomy, organ physiology, cell & molecular physiology, of idiopathic lung fibrosis?"

Data-Set:

- Describe the data set(s) you will analyze: *The data is 78 black and white images that are slices of mouse lungs at various depths. The black in the image is considered the healthy lungs, while the white is the fibrotic lesion that was caused from injecting the mouse's trachea with bleomycin to create lung fibrosis.*
- Cite the source(s) of the data: *Module 2_Lecture 1 - 2025.pptx*
- Describe how the data was collected -- What techniques were used?: *The data was collected through injecting bleomycin, an antibiotic known for being used in chemotherapy that can also cause lung fibrosis, into a mouse's trachea. After three weeks, the lung of the mouse is harvested and fixed in paraformaldehyde and mounted in gel or wax. Then using a cryotome or a microtome, the lungs are sliced into sections.*

The sections are then placed onto a glass microscopic slide and is immunostain by adding a fluorescent-labeled antibody that binds to the proteins (desmin smooth muscle alpha actin, and CD-31). Finally, the slices are viewed under a microscope and a digital image is taken. Important things to note is that the desmin signals are converted to black and white (white = fibrotic lesion and black = healthy lung) and there are a total of 78 images collected at various depths.

- What units are the data measured in?: *The data is measured in microns for the depth of the image.*
- What subjects the data was obtained from?: *The subject the data was obtained from was one mouse that was injected with bleomycin in their trachea whose lungs were harvested after 3 weeks.*
- Was there bias in how the data was collected?: *No, there are no biases in how the data was collected.*
- What were the limitations/assumptions in the data set: *Yes, there are limitations and assumptions in the data set. One major limitation is that the study uses mouse lungs to model fibrosis in humans. Mouse lungs have a horizontal orientation, whereas human lungs have a vertical orientation. This difference alters gravitational, mechanical, and perfusion gradients, which can affect how fibrosis develops and distributes throughout the lung. Therefore, the results from mice may not fully replicate the spatial patterns or progression of fibrosis seen in humans.*

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.)

The following code is used to create a .csv file that contains the filename, depths, and percent white pixels for our 6 speciifc images that we chose: ch010017, ch010018, ch010019, ch010021, ch010022, and ch010159

```
'''Module 2: count black and white pixels in a .jpg and extrapolate
points'''

from termcolor import colored          # lets us print colored text in
the terminal (just for nicer output)
import cv2                             # reads images and does basic
image processing (we'll use thresholding)
import numpy as np                     # helps with fast math on image
arrays (counting pixels)
import matplotlib.pyplot as plt        # makes plots (we'll show
scatter plots)
from scipy.interpolate import interp1d # makes a function that can
estimate values between known data points
import pandas as pd                    # makes a small table and
```

```
writes it to a CSV file

# Load the images you want to analyze
# We'll point to each image by its full path. Order matters because it
must match the depths list below.

filenames = [
    r"/Users/mariellemiranda/Downloads/Module 2 - Lungs/MASK_Sk658
Llobe ch010017.jpg",
    r"/Users/mariellemiranda/Downloads/Module 2 - Lungs/MASK_Sk658
Llobe ch010018.jpg",
    r"/Users/mariellemiranda/Downloads/Module 2 - Lungs/MASK_Sk658
Llobe ch010019.jpg",
    r"/Users/mariellemiranda/Downloads/Module 2 - Lungs/MASK_Sk658
Llobe ch010021.jpg",
    r"/Users/mariellemiranda/Downloads/Module 2 - Lungs/MASK_Sk658
Llobe ch010022 (1).jpg",
    r"/Users/mariellemiranda/Downloads/Module 2 - Lungs/MASK_SK658
Slobe ch010159.jpg",
]

# Depths for each image in microns. The 1st number goes with the 1st
filename, etc.
# We'll use these as the x-axis values and for interpolation.
depths = [
    45,
    90,
    60,
    30,
    80,
    860
]

# Empty lists we will fill as we go.
images = []          # will store each image after we load it in
grayscale
white_counts = []    # number of white pixels (value 255) for each
image
black_counts = []    # number of black pixels (value 0) for each image
white_percents = []  # percent of the image that is white (we'll
compute this)

# Read each image file in grayscale (0 = grayscale). Add it to our
images list.
for filename in filenames:
    img = cv2.imread(filename, 0)   # if path is wrong, this could be
None
    images.append(img)

# Now, for each image:
```

```
# 1) Convert to pure black-and-white using a threshold
# 2) Count white pixels and black pixels
for x in range(len(filenames)):
    # Any pixel >= 127 becomes 255 (white). Anything below 127 becomes
0 (black).
    # We ignore the first returned value (the used threshold) with the
underscore.
    _, binary = cv2.threshold(images[x], 127, 255, cv2.THRESH_BINARY)

    # Count how many pixels are white and black by checking equality
in the binary image.
    white = np.sum(binary == 255)
    black = np.sum(binary == 0)

    # Save those counts for this image.
    white_counts.append(white)
    black_counts.append(black)

# Print out the raw counts so you can see them image by 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()  # just a blank line for readability

# Turn counts into percentages: % white = 100 * white / (white +
black).
# We do this for every image and store the results.
for x in range(len(filenames)):
    white_percent = (100 * (white_counts[x] / (black_counts[x] +
white_counts[x])))
    white_percents.append(white_percent)

# Show each filename and its % white along with its depth.
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'''

# Make a small table (DataFrame) that lines up each filename with its
depth and percent white.
df = pd.DataFrame({
    'Filenames': filenames,
    'Depths': depths,
    'White percents': white_percents
```

```
})

# Save that table as Percent_White_Pixels.csv in whatever your current
working folder is.
# (If you don't see it, check the folder where you're running Python
from.)
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'''

# Ask the user for a depth value. We'll estimate (interpolate) the %
white at that depth.
# The input comes in as text, so we convert to float. 'colored' just
makes the prompt yellow.
interpolate_depth = float(input(colored("Enter the depth at which you
want to interpolate a point: ", "yellow")))

# x = known depths, y = known % white values. We'll build an
interpolation function from these.
x = depths
y = white_percents

# Make a simple straight-line (linear) interpolation function.
# You could switch to 'quadratic' or 'cubic' if you had more data and
wanted a curve.
i = interp1d(x, y, kind='linear')

# Use that function to estimate the % white at the depth the user
typed in.
interpolate_point = i(interpolate_depth)
print(colored(f'The interpolated point is at the x-coordinate
{interpolate_depth} and y-coordinate {interpolate_point}.', "green"))

# For plotting, we make copies of the data lists and add the new
(depth, %white) point to the end.
depths_i = depths[:]
depths_i.append(interpolate_depth)
white_percents_i = white_percents[:]
white_percents_i.append(interpolate_point)

# Make a figure with 2 rows:
# Top plot = only the original measured data
# Bottom plot = original data + the new interpolated point highlighted
in red
fig, axs = plt.subplots(2, 1)

# Plot 1: depth vs % white using your measured images
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)

# Plot 2: same points, but also show the interpolated point at the end
in red
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 point (red)')
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],                # x-value of the last
point (the new depth)
    white_percents_i[len(white_percents_i)-1], # y-value of the last
point (the new % white)
    color='red', s=100, label='Highlighted point'
)

# Make spacing look nicer and show the plots.
plt.tight_layout()
plt.show()
```

Verify and Validate Your Analysis

Verification (checking that the analysis ran correctly):
We verified our analysis by confirming that each image was successfully loaded, converted to grayscale, and processed using OpenCV's `cv2.threshold()` function. For each image, we checked that both white and black pixel counts were non-zero and reflected visible differences in fibrosis severity. The white-pixel percentages printed in the terminal matched the values written to the generated `Percent_White_Pixels.csv` file, confirming that the output data was exported correctly.

We also tested the interpolation by entering known and intermediate depth values (for example, 200 µm and 500 µm) and verified that the predicted percent white pixels fell between the minimum and maximum measured values. The plots displayed clear data trends, and the interpolated point appeared in red at the correct position on the graph—showing that the interpolation and plotting functions worked as intended.

Validation (checking that the results make biological sense):
To validate our results, we compared our analysis trends with known biological findings from studies on bleomycin-induced pulmonary fibrosis in mouse models. Published research shows that fibrotic lesions increase in the lower and outer (subpleural and basal) regions of the lung, while upper regions remain relatively less affected. In our dataset, images taken at deeper depths of the lung showed higher percentages of white pixels—representing more fibrosis—while shallower slices showed lower percentages. This pattern aligns with the expected

physiological distribution of fibrosis, supporting that our analysis and interpolation pipeline produced results consistent with established biological evidence.

Sources:
Moeller A, Ask K, Warburton D, Gauldie J, Kolb M. The bleomycin animal model: a useful tool to investigate treatment options for idiopathic pulmonary fibrosis? Int J Biochem Cell Biol. 2008;40(3):362-82. doi: 10.1016/j.biocel.2007.08.011. Epub 2007 Aug 30. PMID: 17936056; PMCID: PMC2323681.

Conclusions and Ethical Implications

Conclusions:
Our image analysis and interpolation pipeline successfully quantified the extent of fibrosis across different lung depths. The results showed a clear pattern: deeper lung sections contained a higher percentage of white pixels, indicating more fibrosis, while shallower sections displayed less scarring. This supports our overarching question by demonstrating that fibrosis severity increases with tissue depth. The interpolation model also proved useful for estimating fibrosis at unmeasured depths, allowing for predictions without analyzing every physical tissue slice. Overall, this analysis shows that computational imaging can provide meaningful biological insight into disease progression.

Ethical Implications:
Although our data came from animal experiments, it is important to acknowledge the ethical responsibilities involved. The induction of fibrosis using bleomycin must follow approved animal welfare and ethical research protocols (such as IACUC standards). On a positive note, developing digital image analysis tools like this can help reduce the need for additional animal testing in future studies.

Limitations and Future Work

Limitations:

- Only six image slices were analyzed out of the full dataset of 78, which limits the accuracy and smoothness of the interpolation curve.
- The interpolation used a linear model, which assumes a straight-line relationship between fibrosis percentage and lung depth. In reality, fibrosis progression is likely nonlinear and may vary by region.
- The threshold value (127) used to separate black and white pixels was fixed and may not perfectly account for variations in lighting, staining, or image brightness, which can slightly affect pixel classification.
- The analysis only considered one channel (black and white) and did not include other staining information that could help differentiate tissue types or identify specific molecular markers.

Future Work:

- Incorporate all 78 available images to build a more complete and detailed fibrosis-depth curve.

- Experiment with nonlinear interpolation methods (e.g., quadratic or cubic) to better capture complex changes in fibrosis severity.

- Use adaptive thresholding or machine learning–based segmentation to improve accuracy under varying image conditions.

- Extend the analysis to include additional color or fluorescent channels to study relationships between fibrosis and other biological markers.

- Compare the interpolated fibrosis predictions with histopathological scores or published quantitative datasets to further validate the model's accuracy.

NOTES FROM YOUR TEAM:

- *October 2 - We were given the data set and ask to create a .csv file for the percent of white pixels*
- *October 7 - We found out how to convert a .ipynb to a .pdf and turned in our first notebook cheek*

QUESTIONS FOR YOUR TA:

These are questions we have for our TA.

- Is there a scientific paper, like the last module, to learn more about the data set (i.e. data collection and the units the data is measured in)?