



PROJECT:
Segmenting Immune Cells
in Breast Cancer Tissue

Anna, Hanson, Lauren, Lisa, Stanley

Mittal Lab



BACKGROUND

Context: Breast cancer – the most prevalent cancer in the world.

Currently: Breast examination, mammography, breast ultrasound, MRI, and other imaging modalities.

Overall problem: Early diagnosis is key for successful cancer treatment. However, interpretations of screenings can vary from doctor to doctor → misdiagnosis

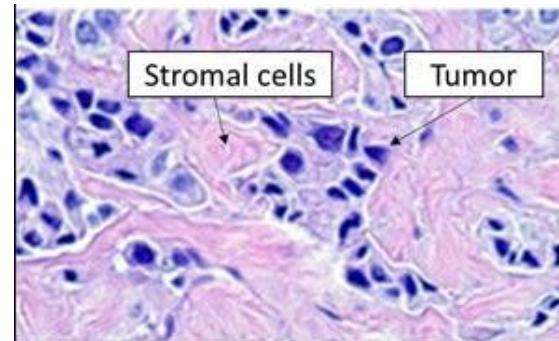
PROJECT SCOPE

Data: Utilize H&E stained images from cancerous tissue to segment tumor infiltrating lymphocytes (**TILs**) in the stroma of breast cancer tissue
→ have clinical significance

Outcome: Better visualize the spatial organization of immune cells in the tumor microenvironment.



TIL

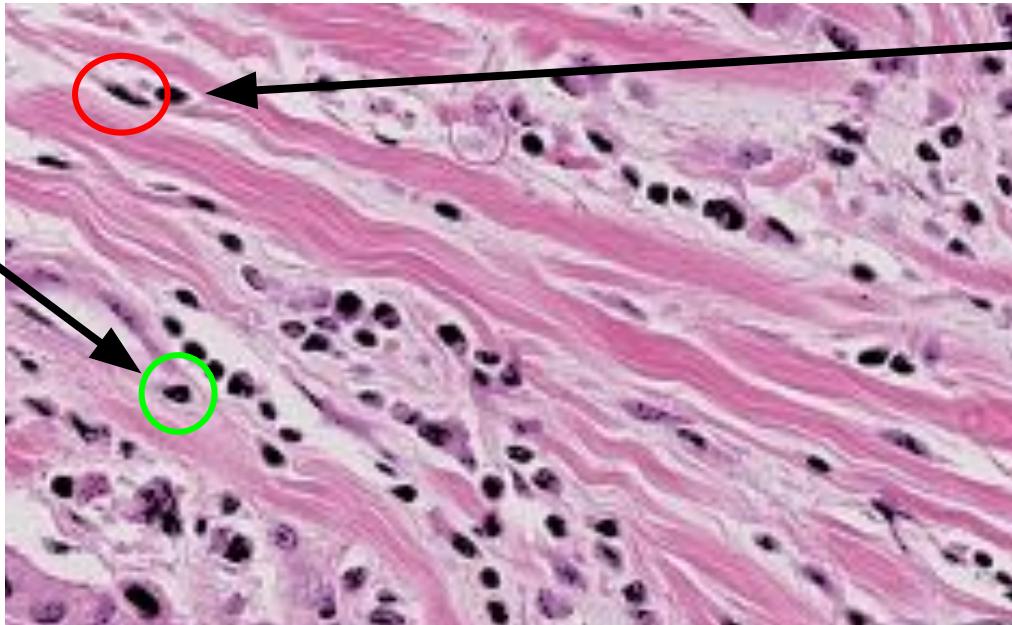


Note: no scale bars available for images

Sakiyama, Nobuyuki & Nagayama, Katsuya (2018)

PROJECT SCOPE

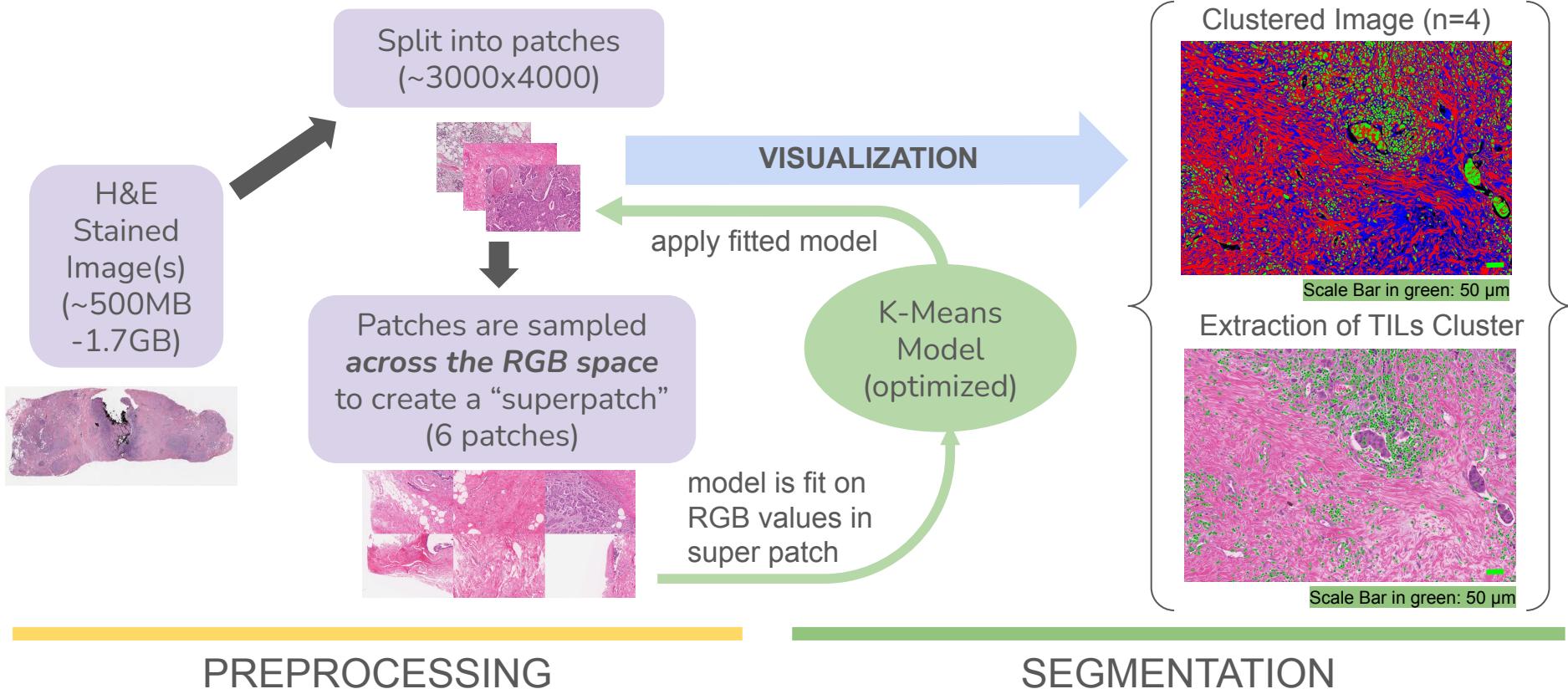
TIL



Note: no
scale bars
available for
images



WHAT WE WERE GIVEN



PREPROCESSING

1

Creating Patches from .Svs Image

1	2	3	4	5
6	7	8	9	10
11	12	13	14	15
16	17	18	19	20
21	22	23	24	25

2

Remove White
Background Patches

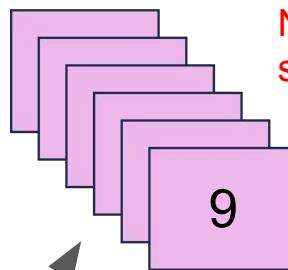
Poorly
documented
method - hard
to make
changes to

1	2	3	4	5
6	8	9	12	13
14	15	16	18	20
21	22	23	24	25

3

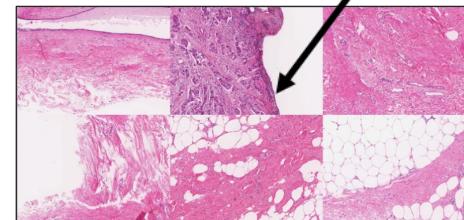
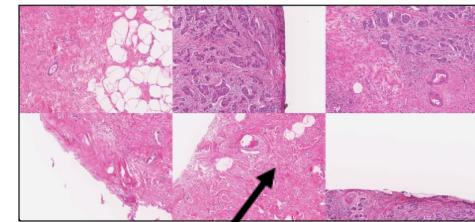
Determine patches to
use in superpatch

No random seed
specified in sampling



4

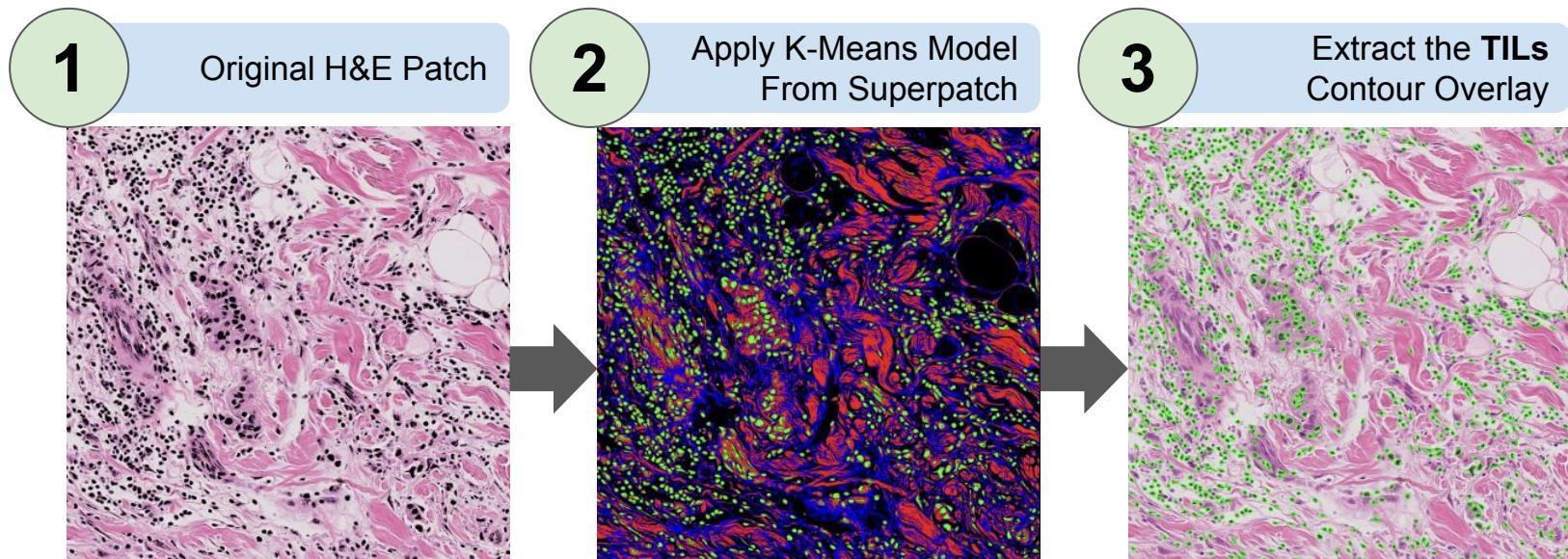
Stitch patches
into superpatch



Different superpatch output
each run and patches still
contain a lot of white



SEGMENTATION: K-MEANS



Currently no way to extract the cluster mask associated with the green TILs and model cannot be performed easily on a single image with no superpatch. The TILs are also not all of the green, how can we further segment them out?



GOALS

Previously: Optimized a K-Means model that can segment TILs from an H&E stained image.

Our Initial Goals:

- Fix previous issues
- Quantify/qualify how “good” (i.e. representative) a superpatch is
 - Explore sampling methods with more images
- Incorporate spatial algorithms to further cluster TILs
- Focus segmentation on just the stroma (the pink parts!)



PROGRESS: DEBUGGING

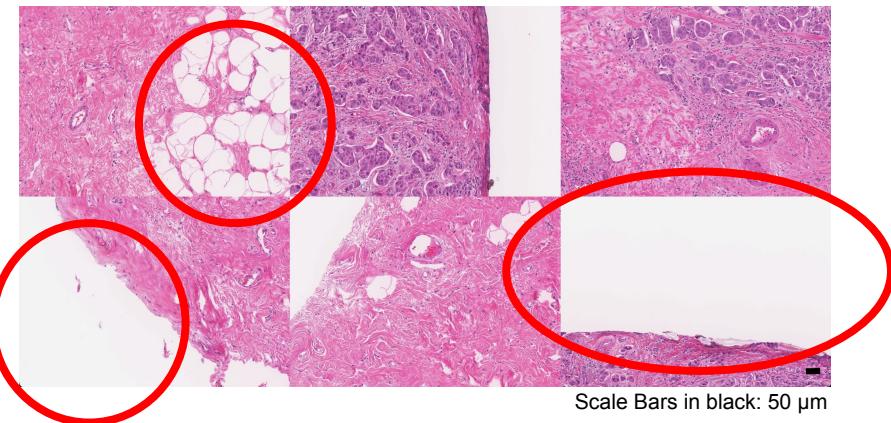
Problem: How do we effectively integrate code from another team within our own project?

- **Random state arguments:**
 - Produces the same superpatch for individual .svs image
- **Background Filtering:**
 - Accurately removes white background within patches
- **Miscellaneous:**
 - Correct os file handling
 - Running previously written unit tests were failing

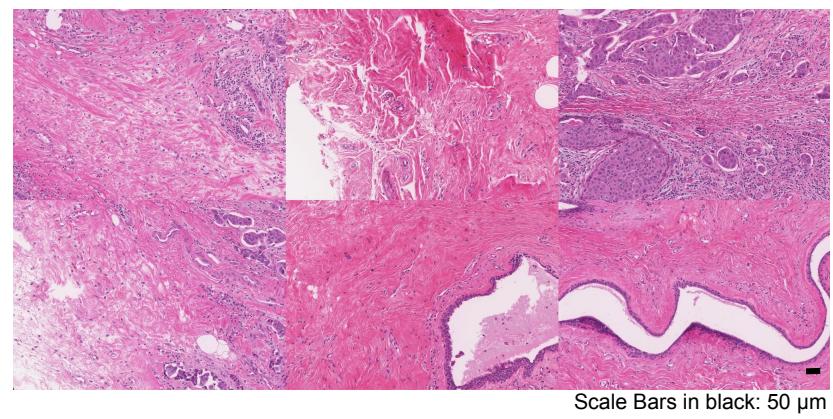
PROGRESS: BACKGROUND FILTERING

Problem: the previous code did not properly filter out the background in the preprocessing step

Super Patch: Previous Code



Super Patch: Gaussian Mixture Model
(2 Peaks)

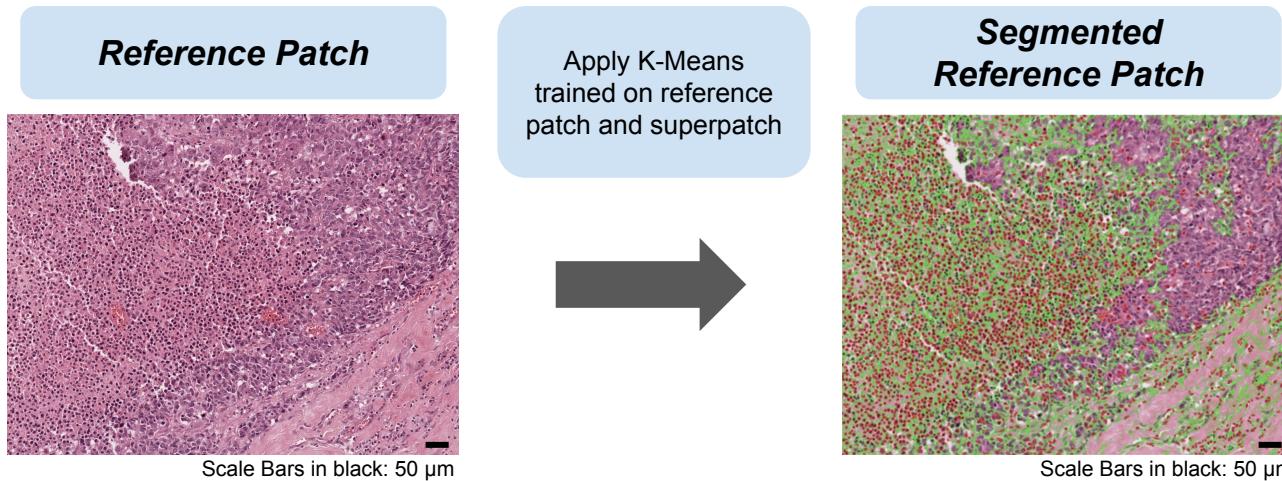


PROGRESS: SUPERPATCH

Problem: How “good” is the superpatch?

- Creating a score + visualizing to determine how good a superpatch is:

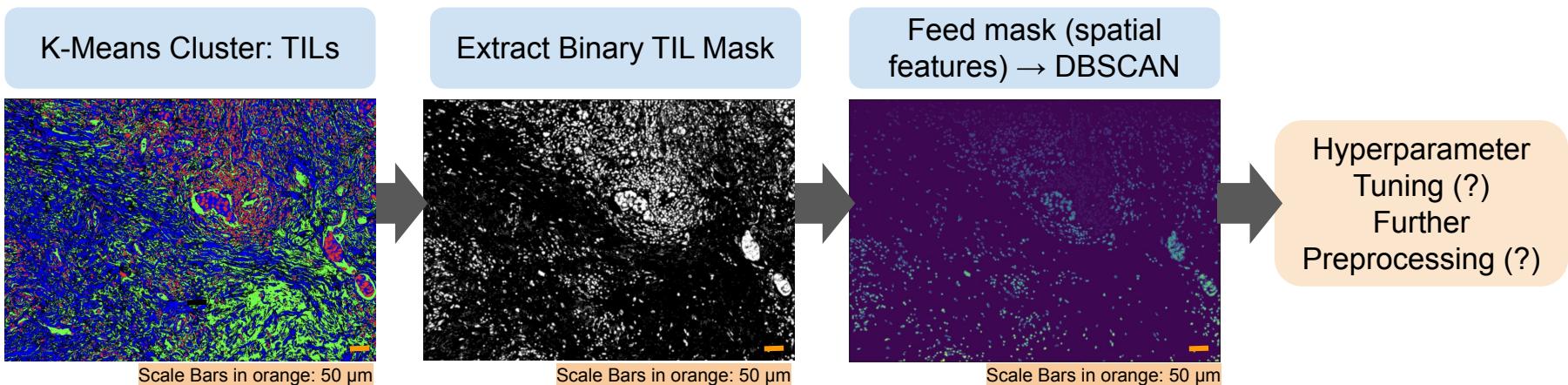
(Quantified with mean squared error)



PROGRESS: DBSCAN

Problem: Is K-Means enough?

- **Implementing DBSCAN:** Integrating a pipeline from K-Means → Spatial algorithms (e.g. DBSCAN)





DOCUMENTATION

- Previous jupyter notebooks outlining process flow used local file paths
 - Added new notebooks for future users for both similarity scoring and DBSCAN/-Means models
- Previous functions missing some docstrings and inconsistent variable naming – updated these in the code and in the Components.md file

Kmeans-Dbscan Segmentation

How to Run Notebook

1. Set up virtual conda environment if you have not already done

Similarity Use

This part utilizes KMeans_superpatch_fit and segment_TILs from seg.py reference patch and a superpatch, and then used to predict clusters on the gives a dictionary containing a masks of the TILs, which is then converted computes the mean squared error between the masks trained on two different mask (red) on top of the reference image.

```
# use the github cloned folder path
import os
directory_path = os.getcwd()
repository_path = os.path.dirname(directory_path)
```

README

TILseg README File

Last Updated: March 13th, 2024

SINCE PREVIOUS UPDATE (March 15th, 2023)

DEVELOPMENTS

- *K-Means to DBSCAN:* Previously, K-Means was found to be the best at finding clusters in TILs. We have built a pipeline in `refine_kmeans.py` to apply K-Means to patches of a whole slide image and then uses this for further clustering.



CHALLENGES

Technical/Logistical Difficulties:

- Lag time in acquiring data
- Long run times (massive amount of data)
- Creating a working environment
- Understanding 1000s+ lines of code in 2 weeks...

However, good documentation!

- Detailed README File
- Docstrings for each function
- Line by line explanation of code



CONCLUSIONS

Our Goals:

- Quantify/qualify how “good” (i.e. representative) a superpatch is
 - Explore sampling methods with ~~multiple images~~
- Incorporate spatial algorithms to further cluster TILs
- ~~Focus segmentation on just the stroma (the pink parts!)~~

Future Directions:

- Improve scoring method
- Test methods on a larger dataset → robustness?
- Hyperparameter tuning (incorporate metrics and others)
- Integrate with the pipeline currently used in the Mittal lab