# 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.

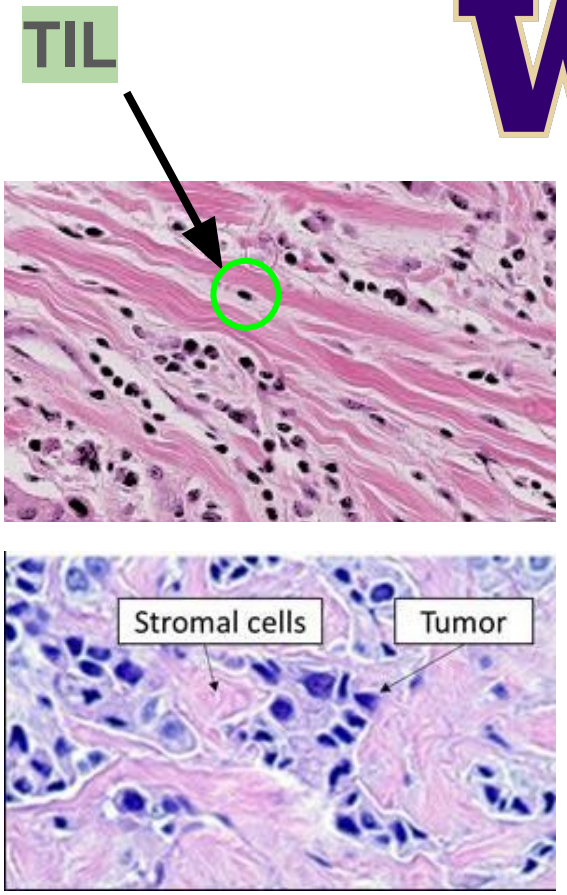
**Currently**: Breast examination, mammography, breast ultrasound, MRI, and other imaging modalities.

**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 (.svs) of cancerous breast tissue to segment tumor infiltrating lymphocytes (**TILs**) in the stroma → have clinical significance

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

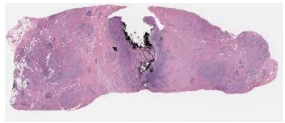



Sakiyama, Nobuyuki & Nagayama, Katsuya (2018)

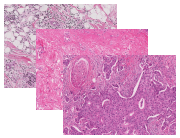
Note: no scale bars available for images

# PREVIOUS CODE (SEMDS WINTER 2023): WHAT WE WERE GIVEN

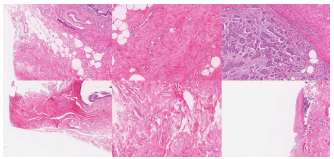
H&E Stained Image(s) (~500MB -1.7GB)



Split into patches (~3000x4000)



Patches are sampled ***across the RGB space*** to create a "superpatch" (6 patches)



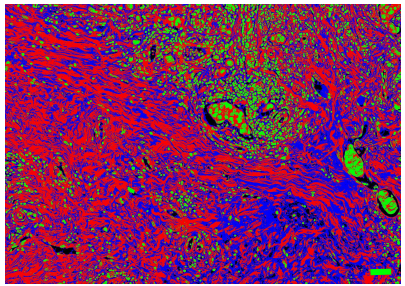
model is fit on RGB values in super patch

K-Means Model (optimized)

apply fitted model

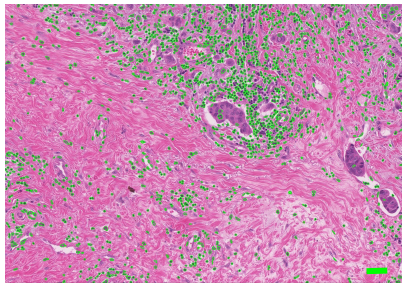
**VISUALIZATION**

Clustered Image (n=4)



Scale Bar in green: 50 µm

Extraction of TILs Cluster

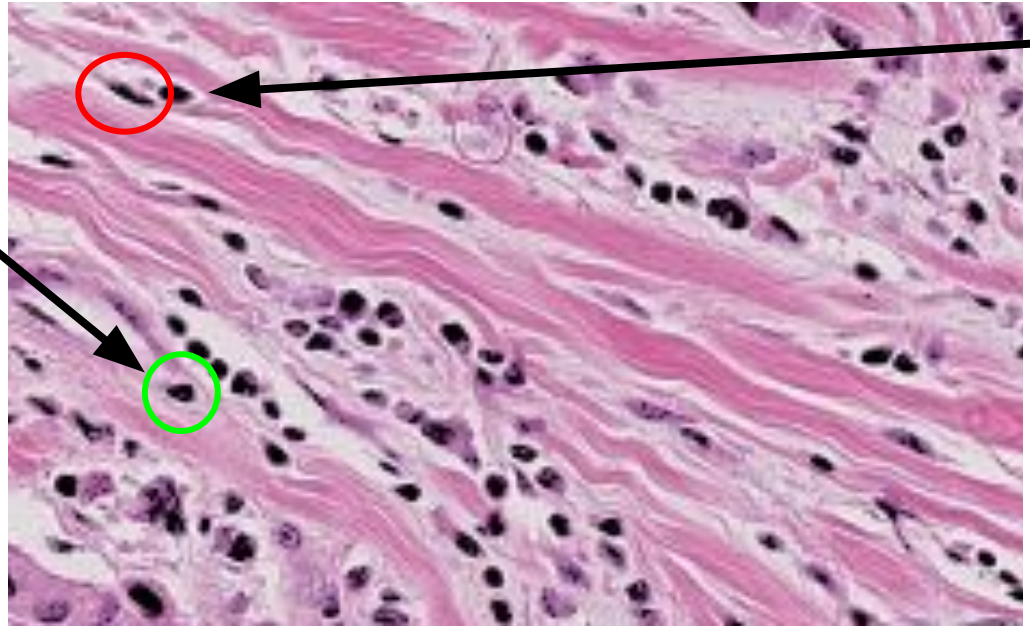


Scale Bar in green: 50 µm

PREPROCESSING

SEGMENTATION

# PROJECT SCOPE



**NOT** a TIL

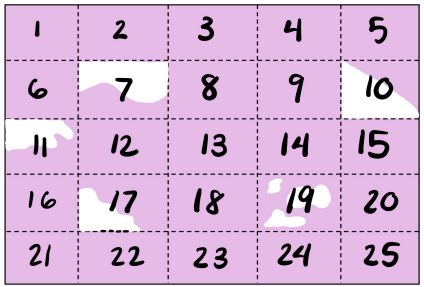
TIL

Note: no scale bars available for images

K-Means will cluster all purple/black cells together →
How to filter out just TILs?

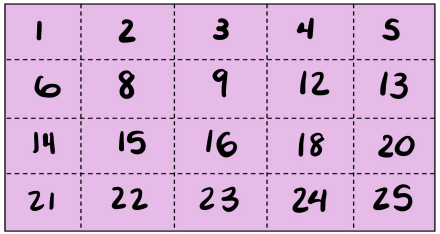
# PREPROCESSING

**1** Creating Patches from .svs Image

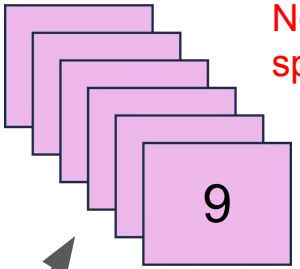



**2** Remove White Background Patches

Poorly documented method - hard to make changes to



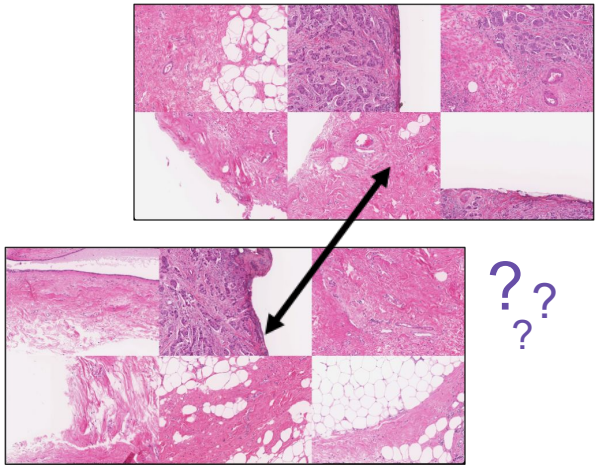

**3** Determine patches to use in superpatch




No random seed specified in sampling

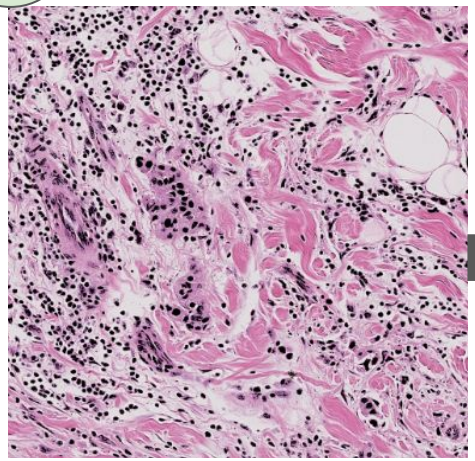
**4** Stitch patches into superpatch

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

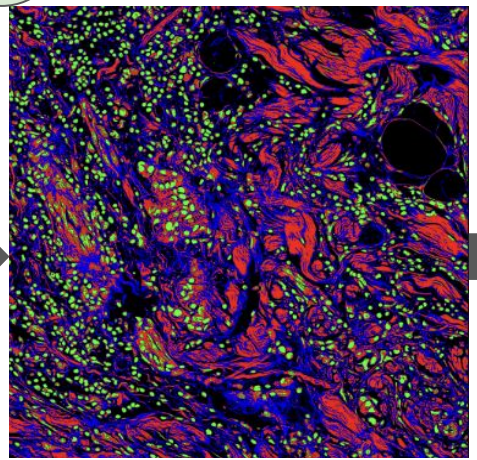


# SEGMENTATION: K-MEANS

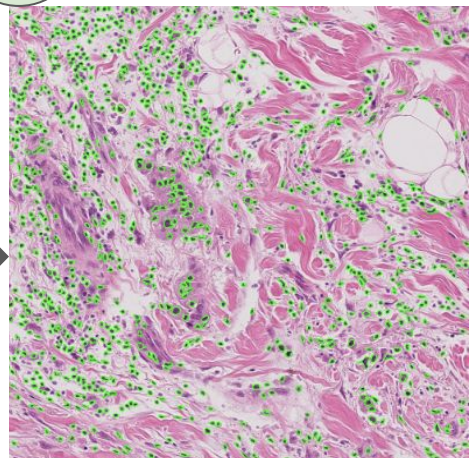
1 Original H&E Patch



2 Apply K-Means Model From Superpatch



3 Extract the **TILs** Contour Overlay



Note: no scale bars available for images from old repository

Currently the clustering 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 issues in the inherited code
- 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

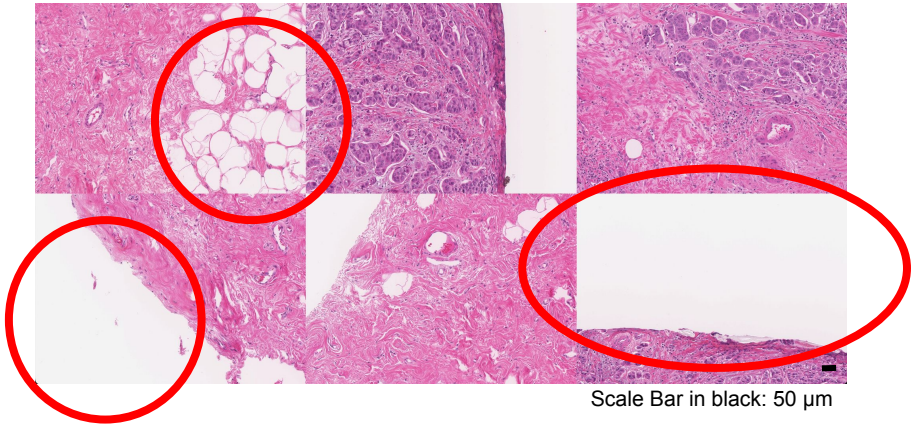
**Problem**: How do we effectively integrate code from another team within our own project?

- **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

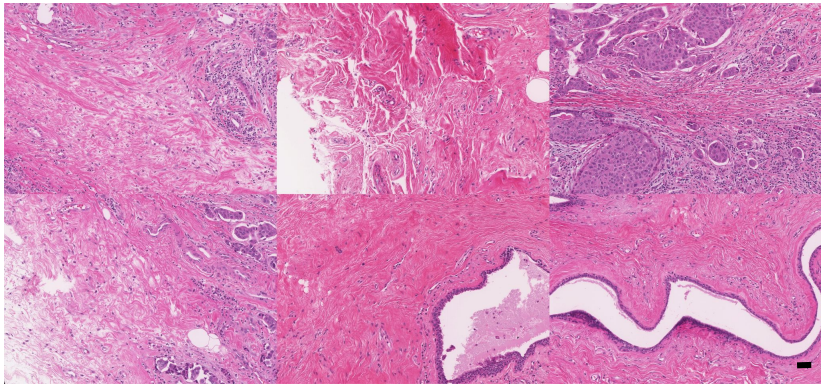
**Problem**: the previous code did not properly filter out the background in the preprocessing step

**Super Patch: Previous Code**



Scale Bar in black: 50 μm

**Super Patch: Gaussian Mixture Model (2 Peaks)**



Scale Bar in black: 50 μm

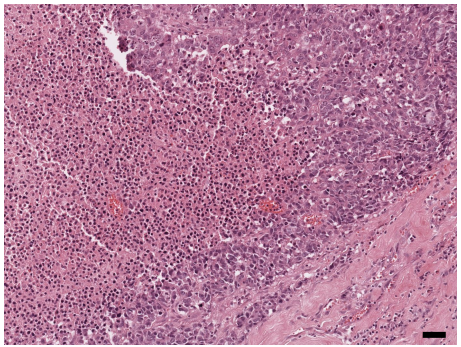
# PROGRESS: SUPERPATCH

**Problem**: How “good” is the superpatch?

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

(Quantified with mean squared error)

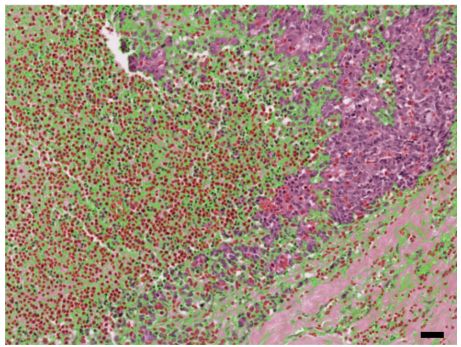
***Reference Patch***



Scale Bar in black: 50 μm

**1.** Fit and apply K-Means on reference patch
**2.** Apply pre-fitted superpatch model on reference patch

***Segmented Reference Patch***



Scale Bar in black: 50 μm

**Green = ‘ground truth’**

**Red = disagreement with superpatch model fit**

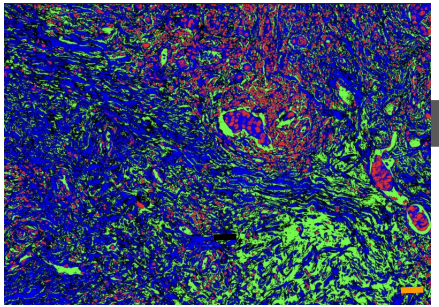
**MSE→ difference**

# PROGRESS: DBSCAN

**Problem**: Is K-Means enough?

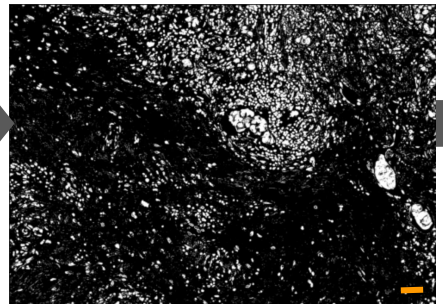
- **Implementing DBSCAN:** Integrating a pipeline from K-Means → Spatial algorithms (e.g. DBSCAN)

K-Means Cluster: TILs



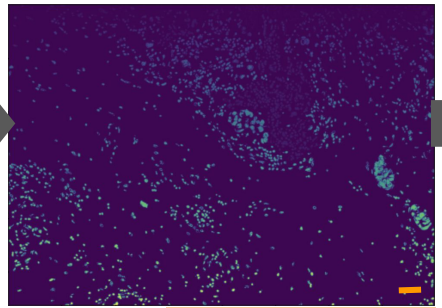
Scale Bars in orange: 50 μm

Extract Binary TIL Mask



Scale Bars in orange: 50 μm

Feed mask (spatial features) → DBSCAN

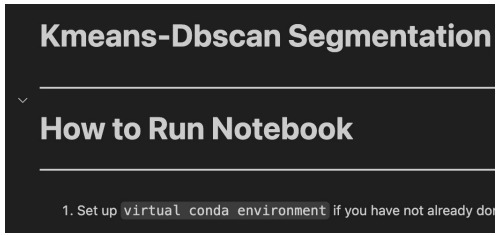


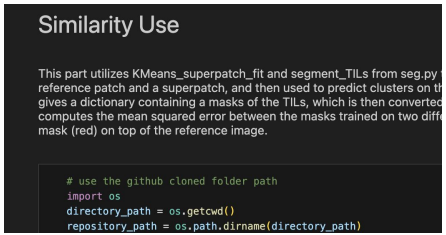
Scale Bars in orange: 50 μm

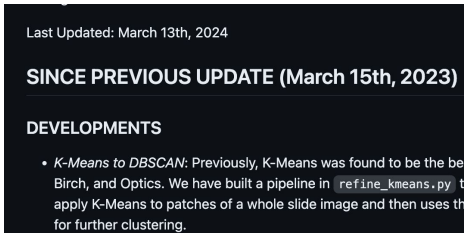
Hyperparameter Tuning (?)
Further Preprocessing (?)

# 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
- Updating documentation → make code more modifiable and understandable for future users/researchers
  - Including bug fixes
  - Updates + future progress sections



# CHALLENGES

**Technical/Logistical Difficulties**:

- Lag time in acquiring data
- Long run times (massive amount of data + complex modeling)
- 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

**Progress:**

- ☑ Creating pipelines to…
  - ☑ Quantify/qualify how “good”/representative) a superpatch is
  - ☑ Incorporate DBSCAN to further cluster TILs from K-Means
- ☑ Debug previous code
- ☑ Improved documentation

**Future Directions:**

- Test methods on a larger dataset → robustness
- Explore hyperparameter tuning + preprocessing
- Integrate with the pipeline currently used in the Mittal lab

# Thank you!