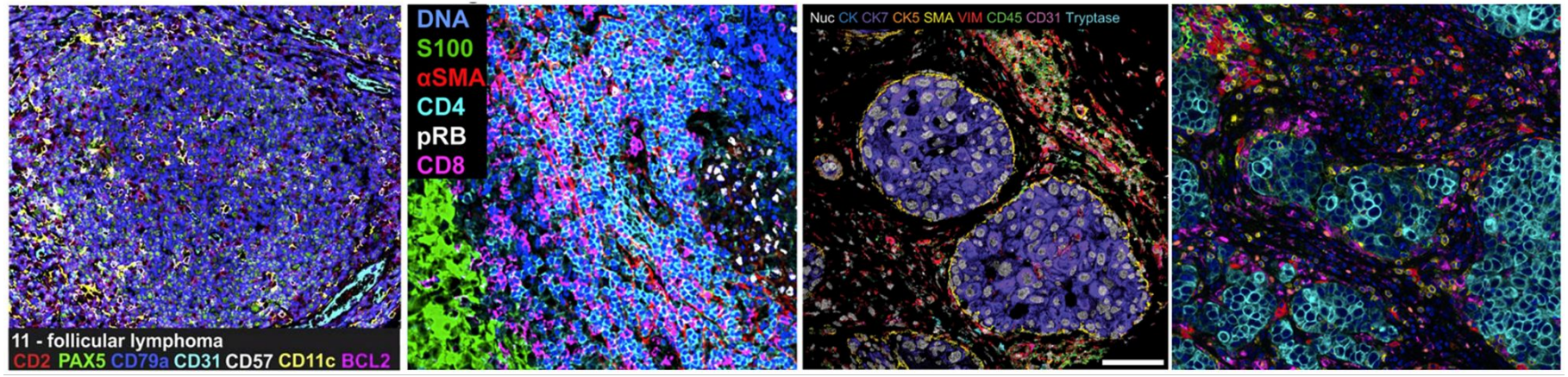
Using vision transformers to predict outcome in ductal carcinoma in situ (DCIS)

IMM310

Candace Liu, Eila Arich-Landkof, Joshua Gillard, Yury Goltsev, John Hickey

Background

- Multiplexed imaging techniques allow us to measure 40+ antibodies on the same tissue section
- MIBI, CODEX, IMC, CyCIF, etc.

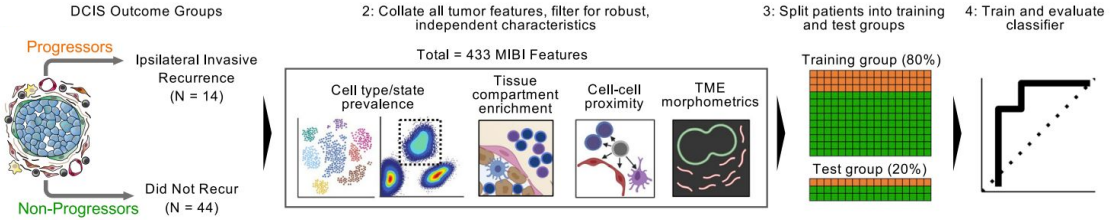


CODEX CyCIF MIBI Vectra

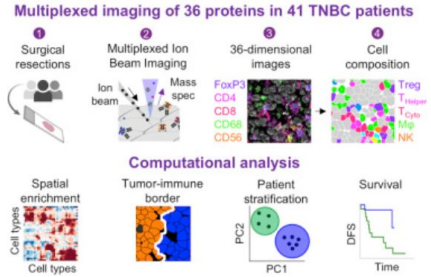
Background

- In multiplexed image analysis, we will typically do feature extraction steps (e.g. cell classifications, extract spatial features, etc.) and use these to try to predict some outcome variable

Risom et al., 2022



Keren et al., 2018

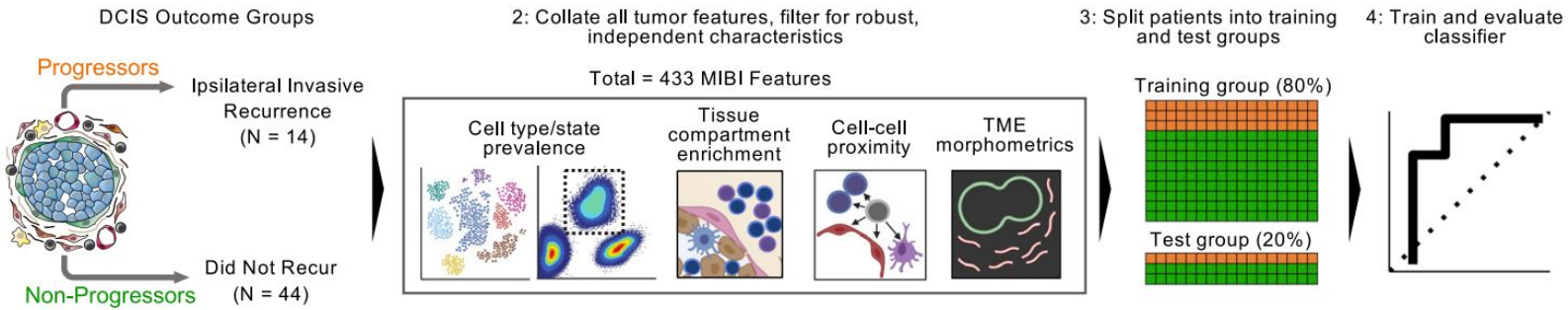


Background

- In multiplexed image analysis, we will typically do feature extraction steps (e.g. cell classifications, extract spatial features, etc.) and use these to try to predict some outcome variable
- Can we skip this “feature extraction” step and use a transformer to predict the outcome variable?
    - Let the model tell us what the “interesting” features are

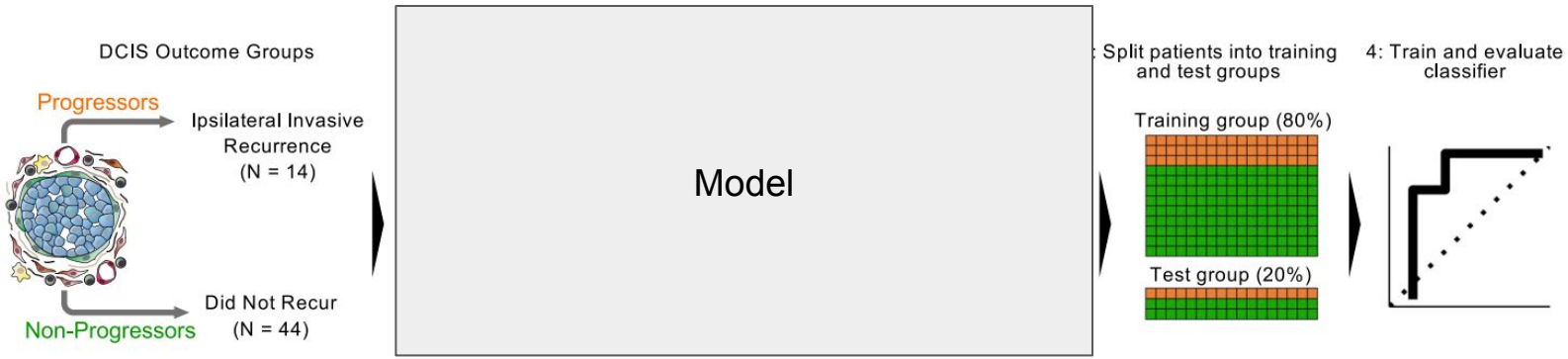
Background

- In multiplexed image analysis, we will typically do feature extraction steps (e.g. cell classifications, extract spatial features, etc.) and use these to try to predict some outcome variable
- Can we skip this “feature extraction” step and use a transformer to predict the outcome variable?
    - Let the model tell us what the “interesting” features are



Background

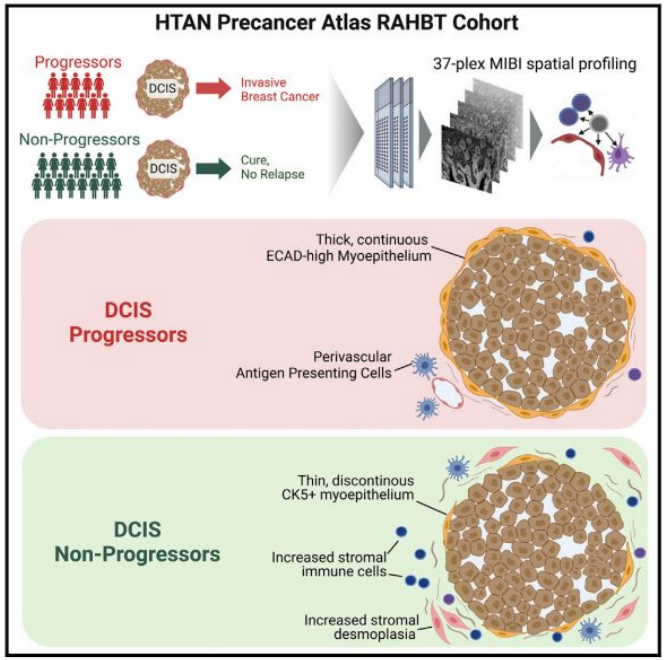
- In multiplexed image analysis, we will typically do feature extraction steps (e.g. cell classifications, extract spatial features, etc.) and use these to try to predict some outcome variable
- Can we skip this “feature extraction” step and use a transformer to predict the outcome variable?
    - Let the model tell us what the “interesting” features are



Cell

Transition to invasive breast cancer is associated with progressive changes in the structure and composition of tumor stroma

Graphical abstract



Authors

Tyler Risom, David R. Glass, Inna Averbukh, ..., Graham A. Colditz, Robert B. West, Michael Angelo

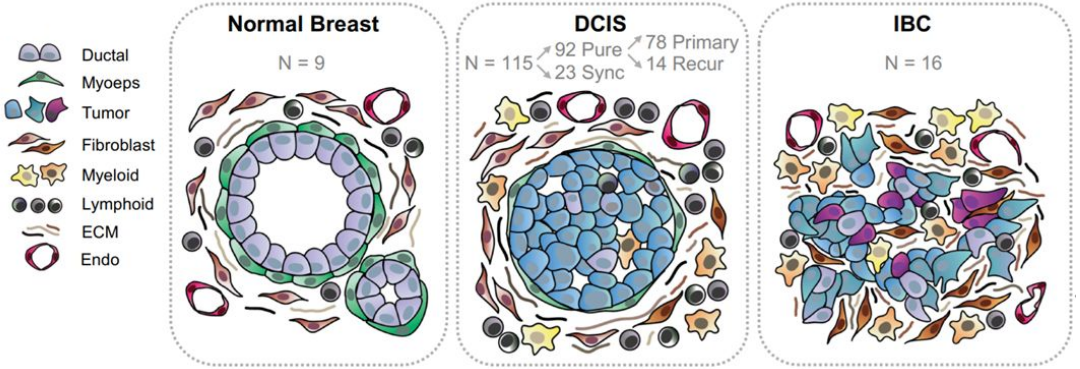
Correspondence

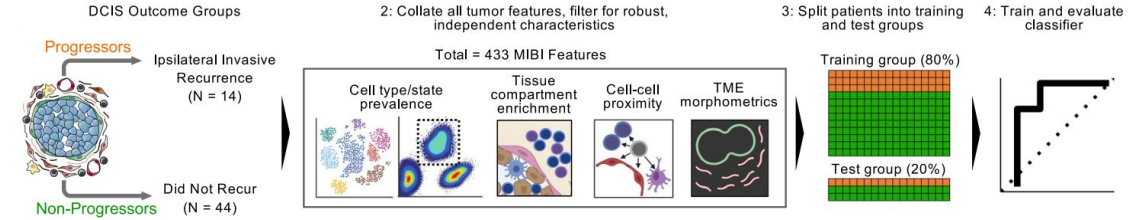
rbwest@stanford.edu (R.B.W.), mangelo0@stanford.edu (M.A.)

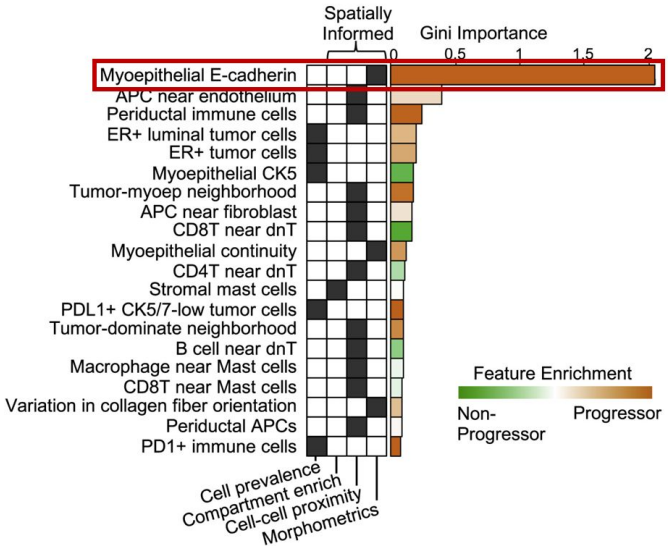
In brief

A spatial imaging atlas of patient-matched ductal carcinoma *in situ* and invasive breast cancer depicts coordinated changes in the tumor microenviroment associated with invasive relapse, suggesting a potential protective role of myoepithelial disruption against invasive progression.

Background


Ductal
Myoeps
Tumor
Fibroblast
Myeloid
Lymphoid
ECM
Endo
Normal Breast
N = 9
DCIS
N = 115
92 Pure
23 Sync
78 Primary
14 Recur
IBC
N = 16

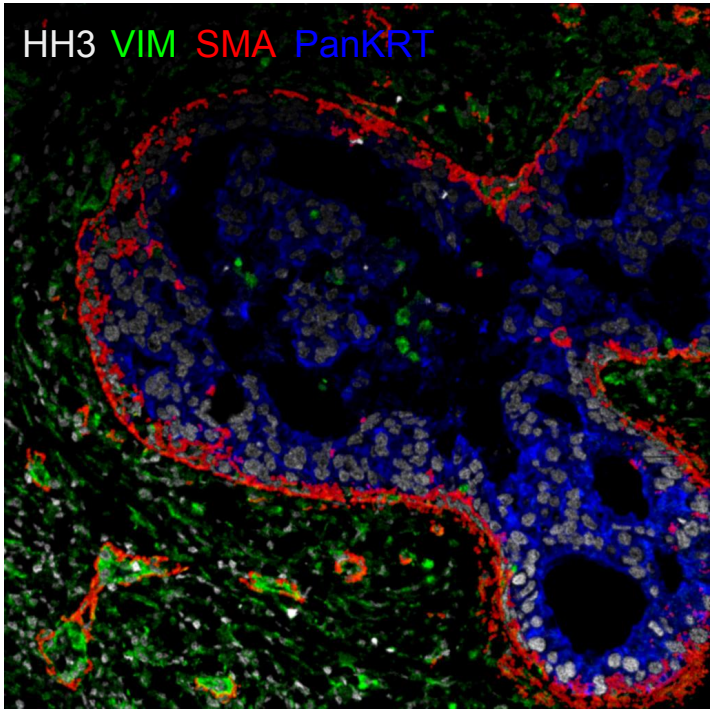

DCIS Outcome Groups
Progressors
Ipsilateral Invasive Recurrence (N = 14)
Did Not Recur (N = 44)
Non-Progressors
2: Collate all tumor features, filter for robust, independent characteristics
Total = 433 MIBI Features
Cell type/state prevalence
Tissue compartment enrichment
Cell-cell proximity
TME morphometrics
3: Split patients into training and test groups
Training group (80%)
Test group (20%)
4: Train and evaluate classifier


Spatially Informed
Gini Importance
0
0.5
1
1.5
2
Myoepithelial E-cadherin
APC near endothelium
Periductal immune cells
ER+ luminal tumor cells
ER+ tumor cells
Myoepithelial CK5
Tumor-myoep neighborhood
APC near fibroblast
CD8T near dnT
Myoepithelial continuity
CD4T near dnT
Stromal mast cells
PDL1+ CK5/7-low tumor cells
Tumor-dominate neighborhood
B cell near dnT
Macrophage near Mast cells
CD8T near Mast cells
Variation in collagen fiber orientation
Periductal APCs
PD1+ immune cells
Cell prevalence
Compartment enrich
Cell-cell proximity
Morphometrics
Feature Enrichment
Non-Progressor
Progressor

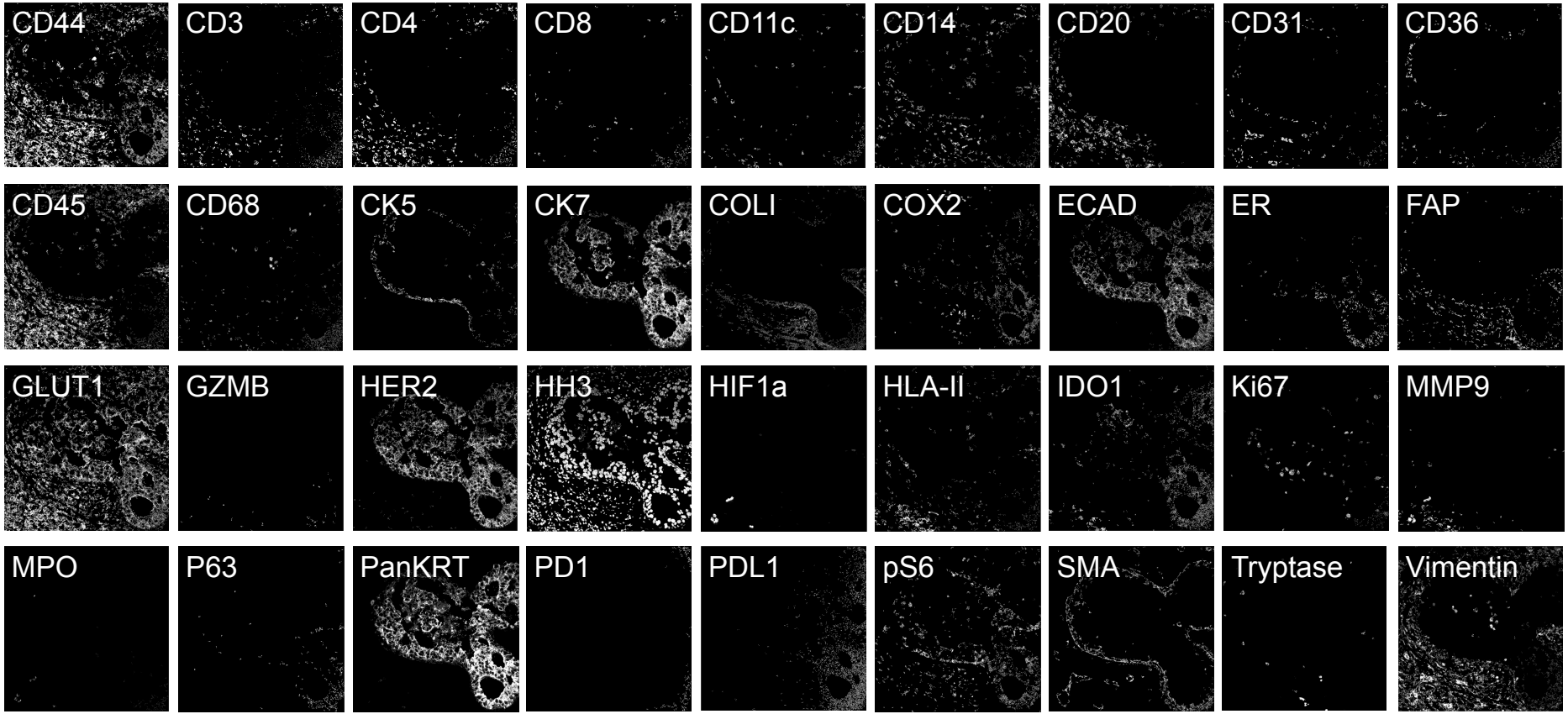
Dataset details

- 58 primary DCIS samples
  - 44 non-progressors, 14 progressors
- Imaged 41 proteins per sample using MIBI-TOF
- **Goal**: stratify non-progressors and progressors using MIBI-TOF images

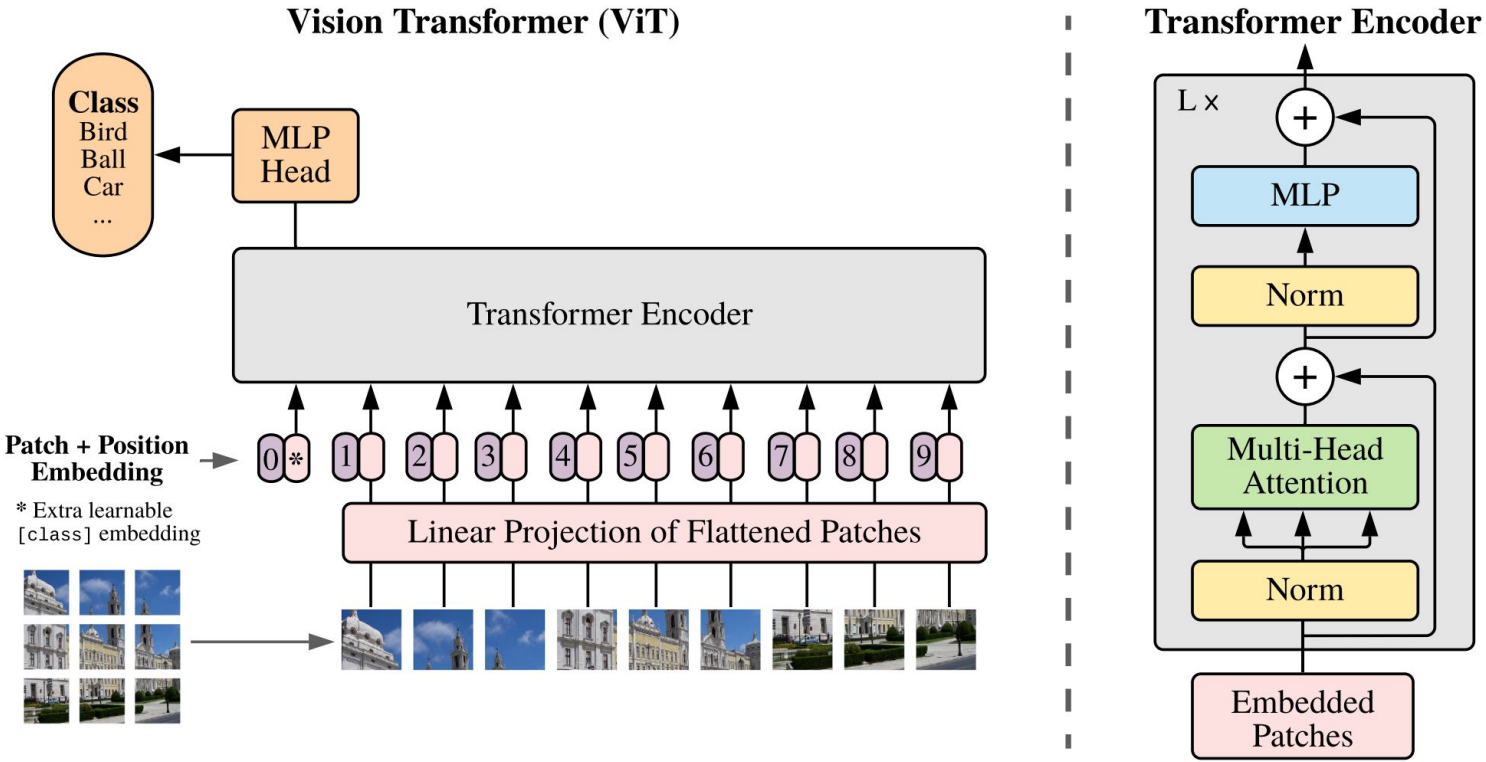
41-plex MIBI images



41-plex MIBI images



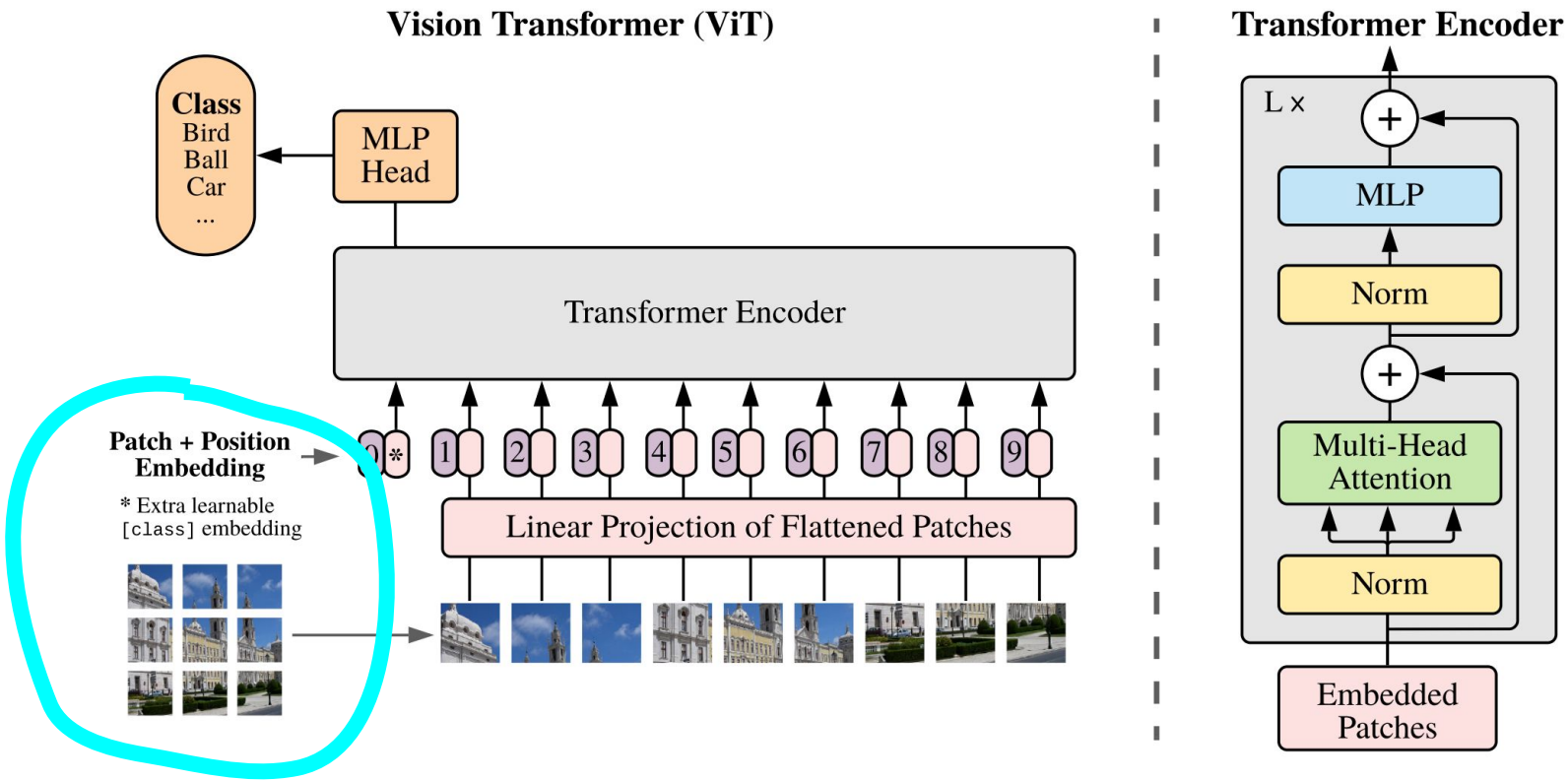
Vision transformer (ViT)



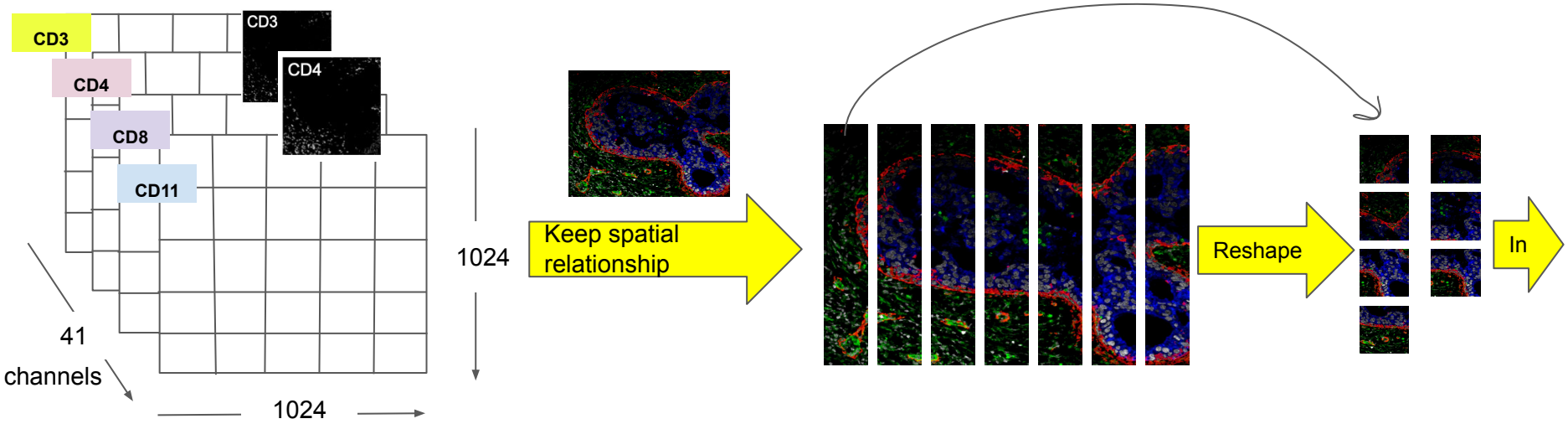
The ViT model details

- Pretrained model: 'google/vit-base-patch16-224-in21k'
  - patch16 => Patches the input images to 16x16
  - 224 => Requires images with dimensions 224x244x3 as input
  - in => Trained on ImageNet dataset
  - 21k => Train on over 21,000 classes (21,834 classes)

Vision transformer (ViT)



Data preprocessing Vision Transformers ViT



Data challenges:

- Spatial dependencies
- Unusual format on input data. 41 channels that need to be collapse
- Large “black” spaces
- Imbalance

Preprocess step 1:

- Collapse data vertically
- Vectorization (for each column, for all rows x channels)
- Remove all 0’z vectors

Addressed the challenges while keeping the spatial relationship

Preprocess step 2:

- Reshape to RGB
- Prepare input for the model (224x224x3)

Data preprocessing output

```
1  ds
```

```
{'train': Dataset({
    features: ['image_path', 'labels', 'pil_image'],
    num_rows: 30519
}),
'test': Dataset({
    features: ['image_path', 'labels', 'pil_image'],
    num_rows: 6542
}),
'validation': Dataset({
    features: ['image_path', 'labels', 'pil_image'],
    num_rows: 6539
})}
```

Training 1 - parameters & loss

```
lr = 5e-3
num_train_epochs = 50
batch_size = 16
early_stopping_callback = EarlyStoppingCallback(early_stopping_patience=5)
```

```
inputs = feature_extractor([Image.open(x) for x in example_batch['image_path']],
                           do_normalize=do_normalize,
                           do_rescale=do_rescale,
                           do_resize=do_resize,
                           augmentation_fn=augmentation_
                           return_tensors='pt')
```

```
  warnings.warn(
```

[8286/9540
4.34/50]

Epoch	Training Loss	Validation Loss	Accuracy
1	0.637200	0.633217	0.671509
2	0.632100	0.627364	0.671509
3	0.628800	0.624400	0.671509
4	0.626000	0.622007	0.671509

Training 2 - parameters & loss

```
inputs = feature_extractor([Image.open(x) for x in example_batch['image_path']],
                           do_normalize=do_normalize,
                           do_rescale=do_rescale,
                           do_resize=do_resize,
                           augmentation_fn=augmentation_fn,
                           return_tensors='pt')
```

```
 8 lr = 5e-2
 9 num_train_epochs = 50
10 batch_size = 8
11 # early_stopping_callback = Early
```

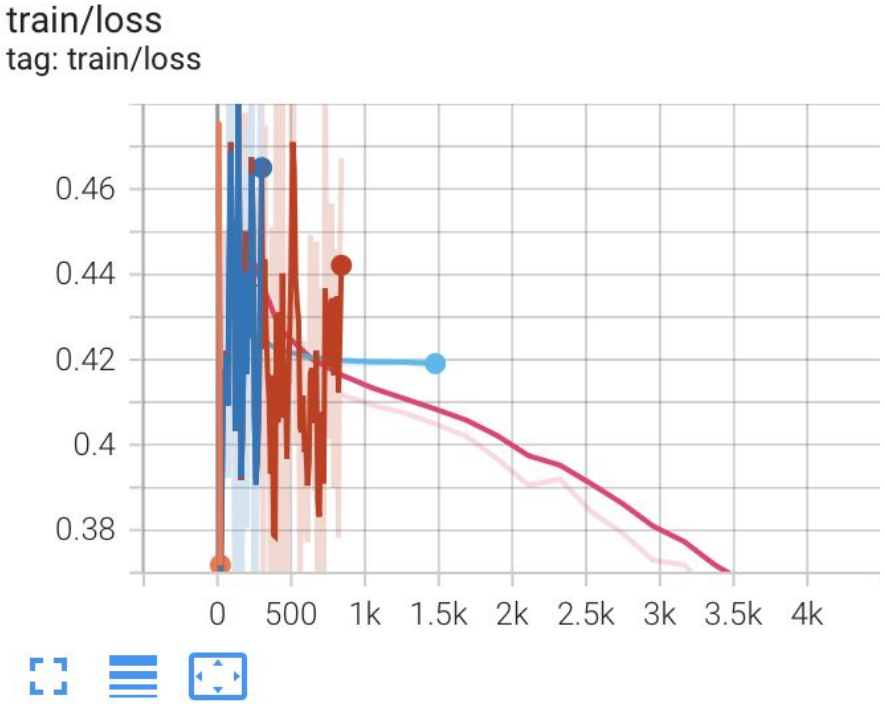
```
1 train_results = trainer.train()
2
```

```
/usr/local/lib/python3.10/dist-packages/transformers/optimiza
  warnings.warn(
[ 201/191000 1:43:10 < 1648:50:2
```

Step	Training Loss	Validation Loss	Accuracy
50	0.729300	0.708615	0.328491
100	0.631600	0.638786	0.671509
150	0.686900	0.643059	0.671509
200	0.829300	0.642763	0.671509

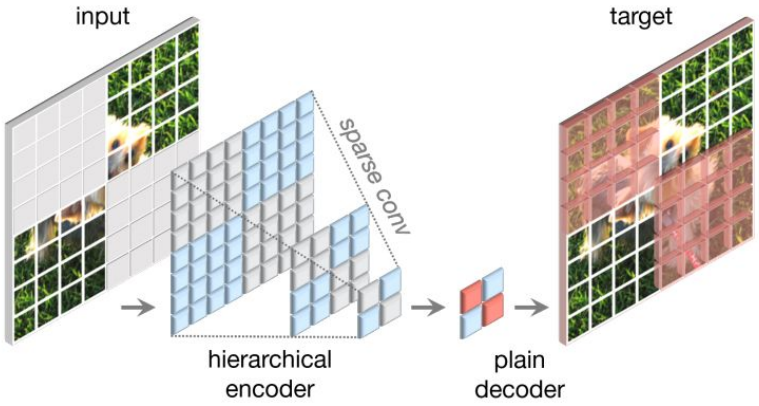
Multiple executions view & insights

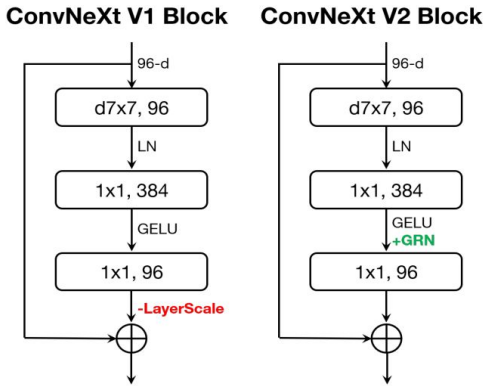
- It is possible to predict progression from the images
- More work need to be done
  - Additional images and execution tuning will be required to make it clinically relevant
- Code for all steps is public: https://github.com/ort-eila/csiseminar



ConvNeXt V2

- Convolutional model inspired by the design of vision transformers
- Successor of ConvNeXt
- Fully convolutional masked autoencoder framework and a new Global Response Normalization (GRN) layer





Preprocessing

- Similar for ViT, but did not reshape to 3 channels - altered the first layer to accept 41 channels
- Model_checkpoint = "facebook/convnextv2-tiny-1k-224"
- Normalized using mean and standard deviation for each channel across all images

Downloaded model:

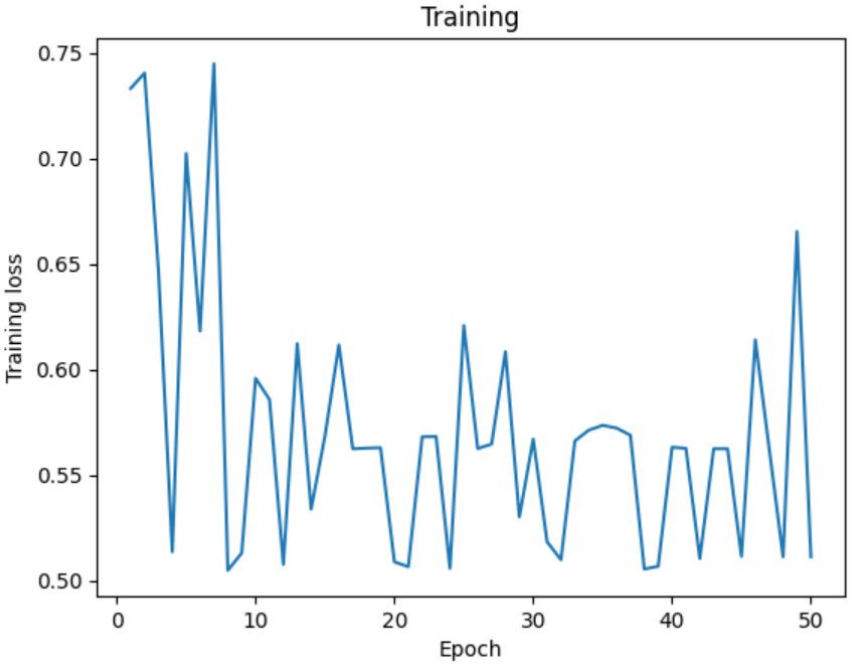
```
ConvNextV2ForImageClassification(
  (convnextv2): ConvNextV2Model(
    (embeddings): ConvNextV2Embeddings(
      (patch_embeddings): Conv2d(3, 96, kernel_size=(4, 4), stride=(4, 4))
      (layernorm): ConvNextV2LayerNorm()
    )
```

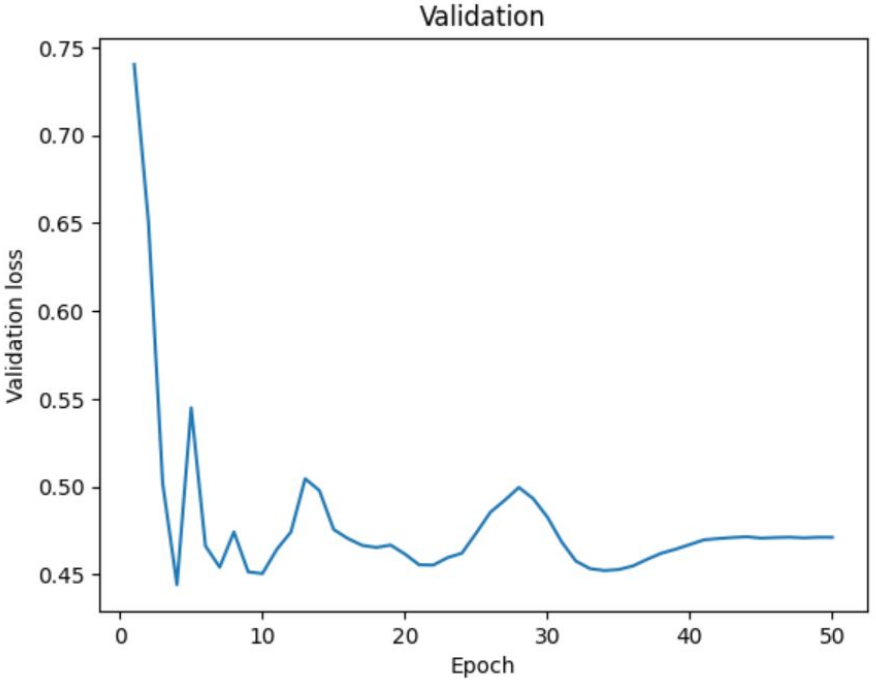
Changed to:

```
ConvNextV2ForImageClassification(
  (convnextv2): ConvNextV2Model(
    (embeddings): ConvNextV2Embeddings(
      (patch_embeddings): Conv2d(41, 96, kernel_size=(4, 4), stride=(4, 4))
      (layernorm): ConvNextV2LayerNorm()
    )
```

```
train_transforms = Compose(
        [
            ToTensor(),
            normalize,
            RandomResizedCrop(crop_size),
            RandomHorizontalFlip(),
        ]
    )
```

Results (kind of)





Validation accuracy = 0.8333

Future directions

- More work to be done
- Acquire more samples
- Data augmentation
- Pre-train or fine-tune the model using multiplexed images from other technologies and disease states so the model can learn what these images look like
- Try different model architectures
  - ImageGPT
  - Swin Transformer
  - Pyramid Vision Transformer
  - Many more…
- Encouraging results!

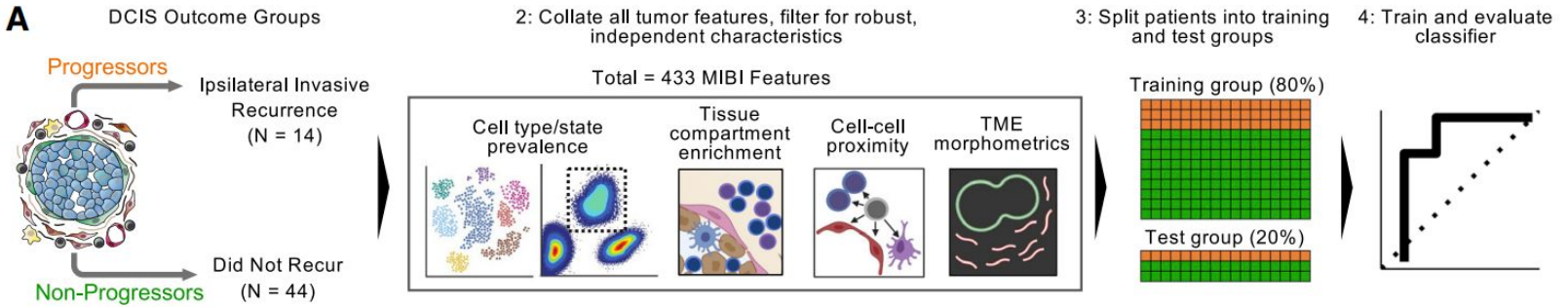
Challenges

- Most pre-trained image classification models expect inputs with 3 channels (RGB)
  - Had to adapt existing models to accept 41 channels
- Small sample size

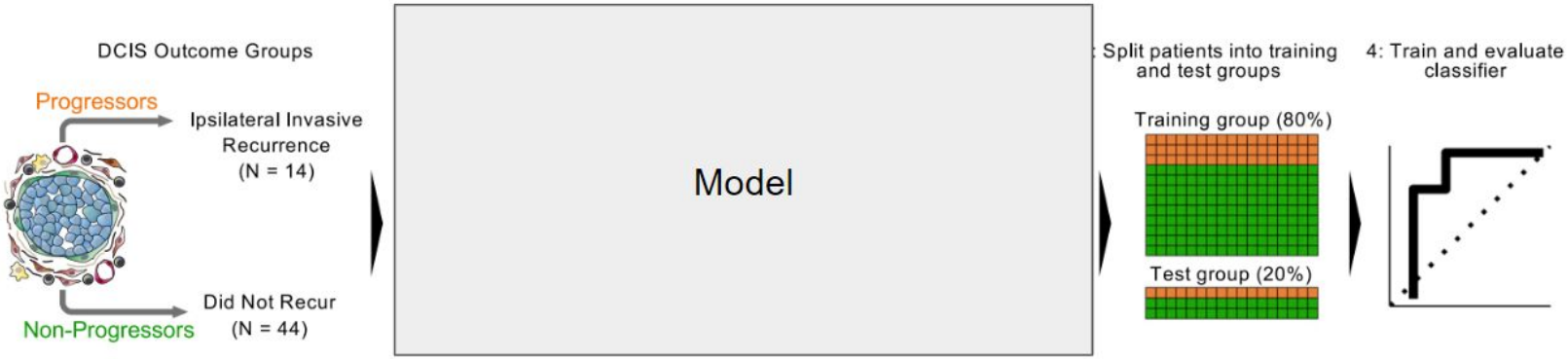
Future options

- Data augmentation
- Use other MIBI images of other types of patients to help with variety of the data and
  - Can be also helpful with non-progressors by reasoning the progressors.

a)


A
DCIS Outcome Groups
Progressors
Ipsilateral Invasive Recurrence (N = 14)
Non-Progressors
Did Not Recur (N = 44)
2: Collate all tumor features, filter for robust, independent characteristics
Total = 433 MIBI Features
Cell type/state prevalence
Tissue compartment enrichment
Cell-cell proximity
TME morphometrics
3: Split patients into training and test groups
Training group (80%)
Test group (20%)
4: Train and evaluate classifier

b)


DCIS Outcome Groups
Progressors
Ipsilateral Invasive Recurrence (N = 14)
Non-Progressors
Did Not Recur (N = 44)
Model
Split patients into training and test groups
Training group (80%)
Test group (20%)
4: Train and evaluate classifier