

Supplementary Materials

A Reconstruction of DAPI Channels

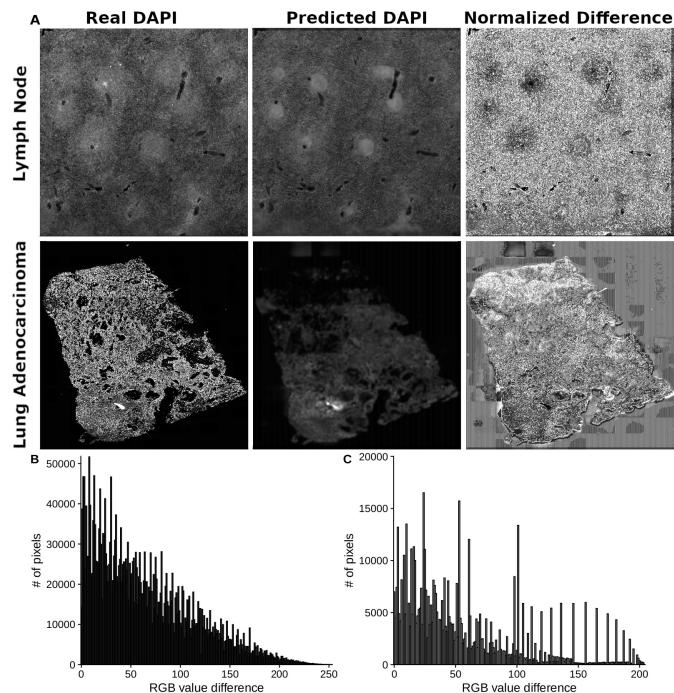


Fig. 1. Predictions of DAPI channels in HuBMAP and NCI data multiplex protein samples yielded reducer per-pixel variation. (A) Normalized predicted and real DAPI channels from photogenic histologies in HuBMAP lymph node tissue and adenocarcinoma samples were overlaid by difference estimates. Images were brightness corrected to highlight morphological similarities. Real and generated channels down sampled to 1024x1024 were normalized by mean histogram equalization and variance was calculated by absolute difference of matching pixels. Increased value of pixels represents high variation in predicted channel. Frequency of pixels at different threshold of variants in (B) HuBMAP and (C) NCI images was reported for normalized difference estimates.

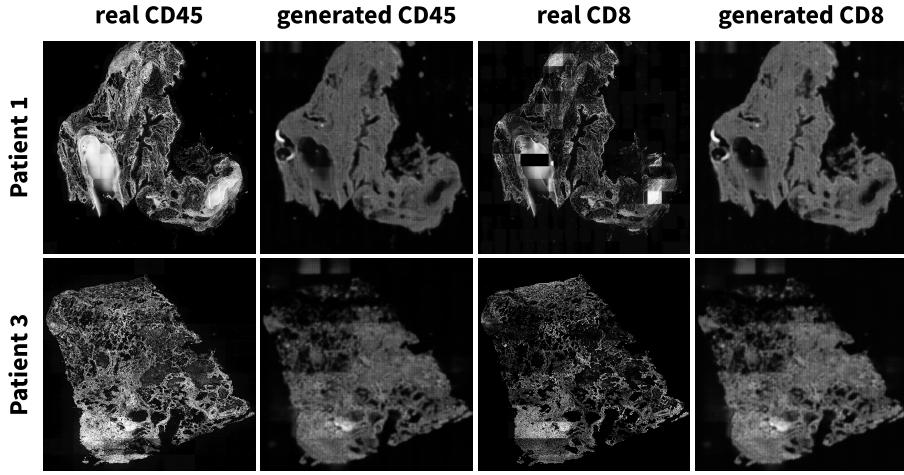


Fig. 2. CD45 and CD8 marker channels are compared to generated channels between samples from two lung adenocarcinoma patients. Images were brightness corrected to highlight morphological similarities.

B Generalization of SSIM Guided Generative Model to Lung Adenocarcinoma Cancer Tissue

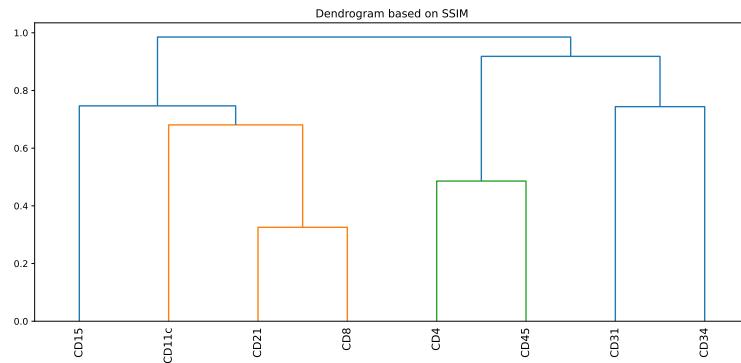
To evaluate the ability for our model to predict protein channels in generalized tissue images (i.e. from different organs and disease state than the HuBMAP model was trained on), we generated channels from lung adenocarcinoma tissue sections using training data from HuBMAP samples. We took sections of lung tissue taken from 5 patients (Table 1) and stained them for CD11c, CD15, CD21, CD31, CD4, CD8, CD45 and DAPI using the CODEX/Phenocycler-fusion platform. The per-pixel variance between real and generated DAPI channels in lung adenocarcinoma sections was comparable to that predicted for HubMAP (Supplementary Figure 1). Predicted CD45 and CD8 channels in two lung adenocarcinoma patients showed high morphological similarity to original stains and the ability to remove artifacts from downsampling (Figure 2).

C Cluster vs. Random Experiment for Experimental Data Set

Figure 3 depicts the hierarchical clustering that was used to evaluate the clustering effectiveness on another experiment in order to select the best model to apply to our experimental spatial proteomics data set.

Table 1. Lung adenocarcinoma patient information

Patient #	Images	Tissue #	Channels per Image	Comment
1	2	Lung	38;39	
2	3	Lung	36;25;18	Large immune infiltrate
3	1	Lung	35	Immune infiltrate within the tumor and at the tumor edge
4	2	Lung	18;23	Few cells, especially in the bottom part of the tissue, mostly connective tissue
5	2	Lung	39;19	Largest tumor with significant immune infiltration

**Fig. 3.** SSIM based Clustering

We have used all possible combination of channels 5 channels to predict 3 channels for this experiment.

- source channels: all combination of ['CD11c', 'CD15', 'CD21', 'CD31', 'CD4']
- target channels ['CD45', 'CD8', 'CD34']
- best performing model (on validation loss) for the source channel of size four ['CD11c', 'CD15', 'CD21', 'CD31']

Figure 4 and Figure 5 represent the generator loss (training) generator loss (validation) for this experiment respectably.

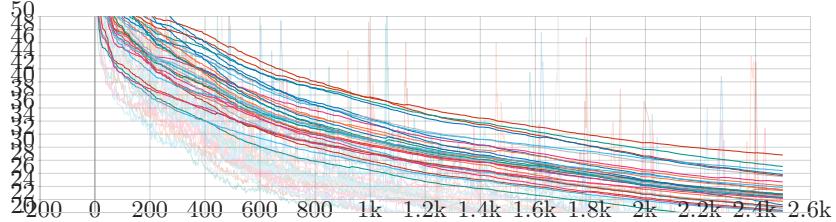


Fig. 4. Generator Training Loss

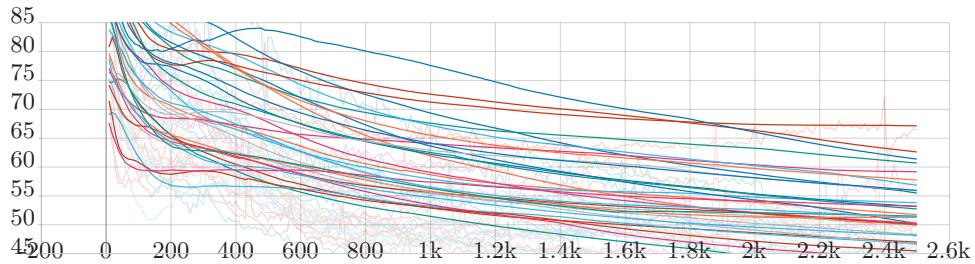


Fig. 5. Generator Validation Loss

D Loss Comparison Varying Number of Input Channel

We discovered that increasing the source channel improves model performance in one of our experiments. We observed that the validation loss decreases as the number of source channels increases. Figure 6, and Figure 7 represent the generator loss, and validation loss for predicting N_{pr} channel based on N_{ip} channel. where N_T , N_{ip} , N_{pr} stand as below.

$$N_T = \text{Number of total channel in the image}$$

$$N_{ip} = \text{Number of channels selected as source image}$$

$$N_{pr} = \text{Number of channels selected for prediction}$$

$$N_{pr} = N_T - N_{ip}$$

For Training, the loss for varying number of input channels in the final step are as follow, $16 > 17 > 15 > 18 > 19 > 20 > 23 > 21 > 22 > 24 > 25$ [Sorted based on loss]

For Validation, the loss for varying number of input channels in the final step are as follow, $17 > 16 > 19 > 15 > 18 > 20 > 23 > 21 > 24 > 22 > 25$ [Sorted based on loss]

We have also found that some channels are hard to predict. Below is the loss plot for Generating 4 different channels, Vimentin > PanCK > Podoplanin > SMAActin [Sorted based on loss]

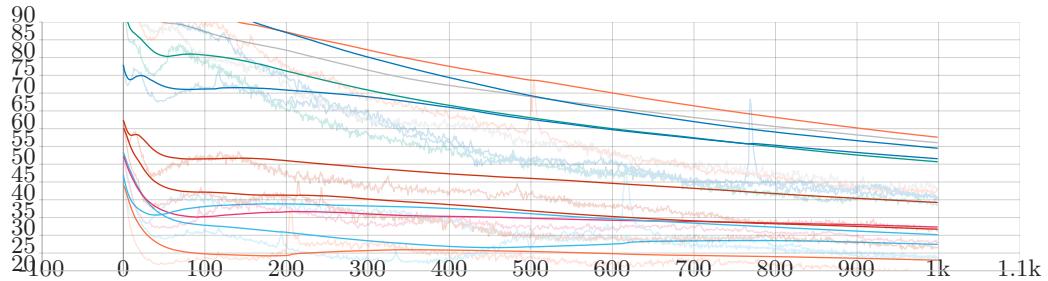


Fig. 6. Generator (Train) Loss on Different Number of input channel

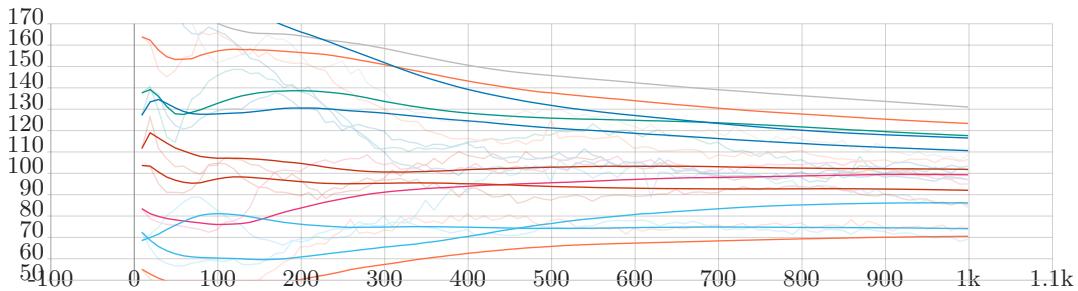


Fig. 7. Generator (Validation) Loss on Different Number of input channel

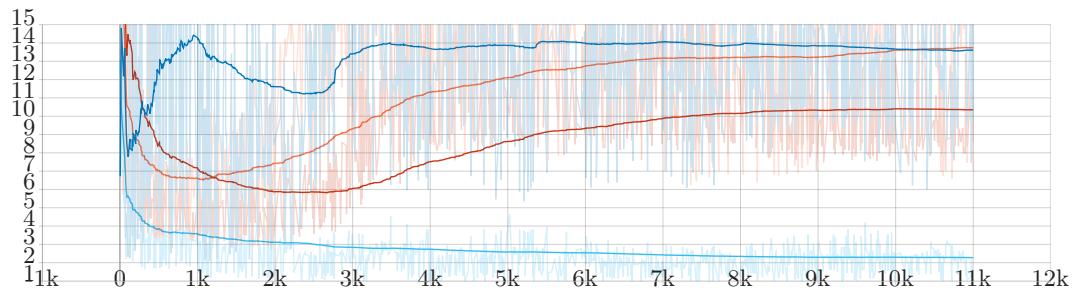


Fig. 8. Generator Loss for Different Channel

E Training Guideline

Step 1- Coning and Library installation First you have to clone the repository and install the required libraries using the following commands (GPU required).

```
git clone https://github.com/aauthors131/multiplexed-image-synthesis
pip install -r requirements.txt
```

Step 2 - Channel Selection: Use the below script to get source and target input channel based on SSIM.

```
script: select_channel_from_cluster.py

python3 select_channel_from_cluster.py --data_dir Data/ \
→ --channel_names channel_names.txt --num_of_cluster 6 \
→ --percentage_from_each_cluster 0.8
```

It will generate a json file for source and target channel (channel indices of given tiff image) like below. You can manually create the files if you want.

```
channel_id_for_training = {"uid": "1", "source_channel_ids": [
→ [1,2,3,4,5], "target_channel_ids": [6, 7]}
```

Step 3- Training:

```
python3 train.py
--raw_data_dir "Data/" \
--tb_logger_name "tb_logger" --dataset_name "codex" \
--channel_ids "channel_id_distribution/cids.json"
```

You can also pass more parameter for network configuration

Table 2. Important Parameters for Network Configuration

-n_gf	the number of channels in the first layer of G
-n_downsample	how many times you want to downsample input data in G
-n_residual	the number of residual blocks in G
-n_D	how many discriminators in different scales you want to use

Running this script will begin training a model and attempt to generate the target channels of a given tiff image based on the source channels. It will save the two best models based on validation loss. The tensorboard will save related plots.

Step 4- Testing: You can use a pre-trained model to run on different test sets using the below script.

```
python3 test.py --raw_data_dir "Data/"
--trained_model_dir
↪ "checkpoints/test/Model/epoch=1699-step=3400.ckpt" \
↪ --channel_ids "channel_id_distribution/cids.json"
```

Step 5- Inference: You can use a trained model to generate target channel images based on source channels. To make inference, you can use the following script. You'll pass empty list for "target_channel_ids" in json.

```
python infer.py --trained_model_dir
↪ checkpoints/test/Model/epoch=1699-step=3400.ckpt
↪ --channel_ids cids.json --raw_data_dir "Data/"
```

Pre-trained Model: A pretrained model is included in the repository named "epoch=1699-step=3400.ckpt" that trained on ['CD11c', 'CD21', 'CD15', 'CD4', 'CD31', 'CD34'] channels to predict ['CD8', 'CD45'] channels. So, with an image of a given tissue that contains the expression of the first set of channels, you can use this model to synthetically generate the expression of the second set of channels.

F Generated Channel Samples

G Clinically Relevant Generative Images

Table 3. CD8 Density

Biomarker	Highly expressed (pixel count)	Highly expressed CD8 over the sample (ratio)
CD8 (Real)	15753	0.146
CD8 (Synthesized)	15299	0.129

H Data and Code Availability

For review, we have shared all code and new data generated. The new CODEX lung adenocarcinoma dataset that we generated from human patients to validate the novel computational methods in this paper is available here. All of our code to recreate our results is available here. We also utilized previously published CODEX data from the HuBMAP consortium available here. Specifically, we included the following HuBMAP samples in our training:

HBM347.PSLC.425, HBM863.FDNH.844, HBM992.RHJW.288, HBM622.JXWQ.554, HBM495.VWBD.428, HBM857.ZBDC.975, HBM496.ZJFC.554, HBM465.HZHH.676, HBM887.SHVF.747, HBM268.NKXB.243, HBM938.TNNT.879, HBM597.KZXW.469,

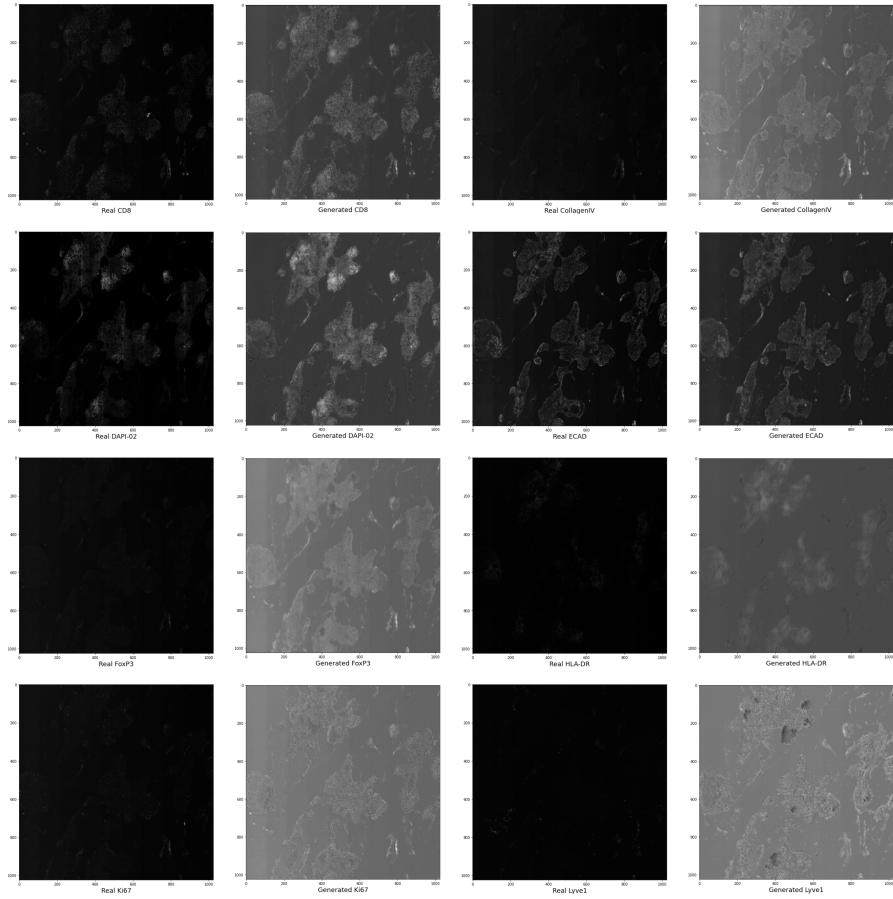


Fig. 9. Sample output on the Test set (trained on 17 channels to predict 12 other channels on HuBMAP data

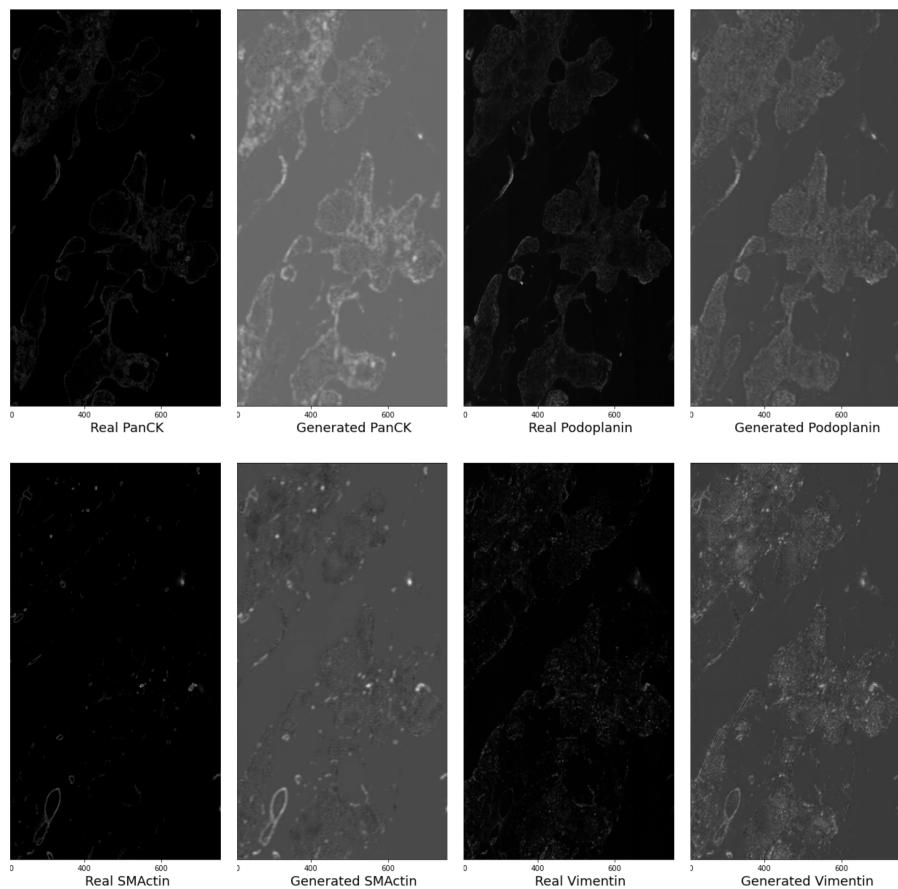


Fig. 10. Sample output on the Test set (trained on 17 channels to predict 12 other channels on HuBMAP data)

HBM373.LDGF.766, HBM522.BSZT.385, HBM528.NMQV.274, HBM396.BXSQ.568,
HBM825.KFFT.669, HBM557.SGTC.262, HBM893.CCKX.496, HBM997.PVCF.629,
HBM685.TBGN.663.