**Thyroid Project Outline**

James Dolezal, December 2018

**Aim**

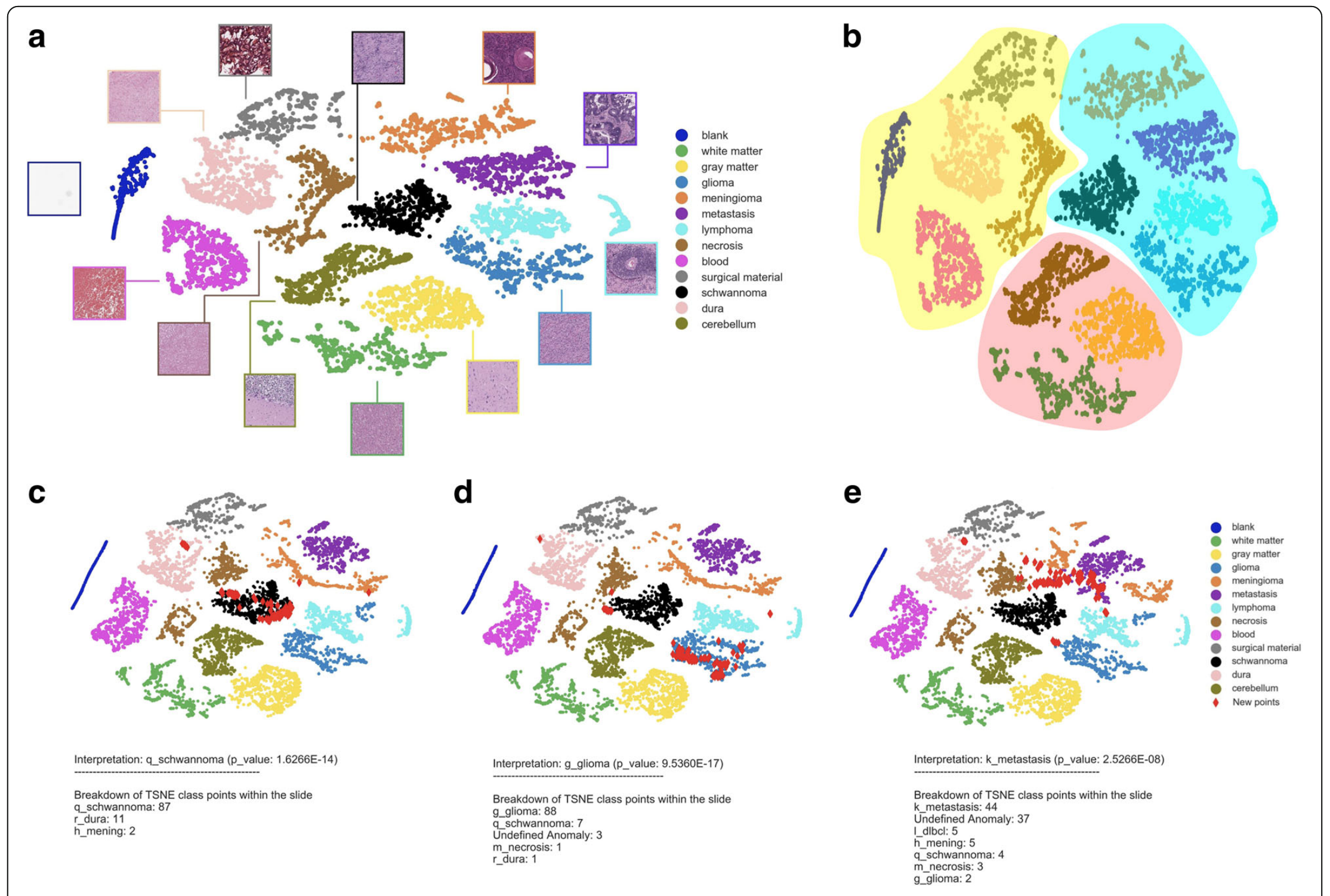
To distinguish noninvasive follicular thyroid neoplasm with papillary-like nuclear features (NIFTPs) from follicular adenomas and papillary carcinomas with extensive follicular growth (PTC-EFGs) using machine-learned based histologic image analysis, and to examine the similarities and variability of visual features among these classes.

**Approach**

A convolutional neural network will be trained to distinguish five histologic classes: benign adenoma, follicular thyroid carcinoma (FTC), NIFTP, PTC-EFG, and classic papillary thyroid carcinoma (PTC-classic). The neural network model will then be applied to representative sections of slides from each of the classes, and the final layer weights will be visualized using a machine learning based dimensionality reduction algorithm, demonstrating the relationships between the histologic features of the classes. An example result of a similar analysis is provided in **Figure 1**.

**Hypothesis**

When the visual features from the final layer of the trained neural network are mapped using t-SNE, follicular adenomas and classic PTCs will cluster distinctly, while NIFTPs and PTC-EFGs will cluster near one another, supporting the hypothesis that these two diseases represent two sides of a morphologic spectrum.



***Figure 1.*** *Sample results from an analysis of brain tissue subtypes using a neural network and t-SNE for final layer visualization. Faust, 2018. https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5956828/pdf/12859\_2018\_Article\_2184.pdf*

**Methods**

***Preparing and annotating whole-slide images***

Using a \*\*\* slide scanner with capture resolution of 0.275 microns per pixel (viewing magnification: 40X), pathologist-annotated slides of 7 benign adenomas, 2 FTCs, 42 NIFTPs, 8 classic PTCs, and 13 PTCs with follicular growth were obtained in Aperio (\*.svs) format. Using Aperio, slides were exported into JPG format in chunks, with a maximum size of 65535 x 65535 pixels. JPG chunks were annotated at 10% resolution using [LabelMe](https://github.com/wkentaro/labelme).

***Preparing slide tiles for the convolutional neural network***

Using the 10% resolution annotations as a guide, full-resolution JPG chunks were sectioned into tiles of 1024 x 1024 pixels (height/width of 280µm, apparent magnification of 40X) and 512 x 512 pixels (height/width of 140µm, apparent magnification of 40X). The 1024 x 1024 sections were then down-scaled to a resolution of 512 x 512 with a height/width of 280µm and apparent magnification of 20X. This provided three sets of tiles for use in determining which image size and apparent magnification would be optimal for training. All tiles were then flipped and rotated to increase yield.

***Neural network analysis***

A neural network based on Inception-v4 with 2048 output features was built using Python and Tensorflow. The analysis was performed using GPU acceleration from an Nvidia GeForce 1080 Ti. The general structure of the neural network pipeline is as follows:

1. Data augmentation: input tiles are processed, normalized, and assembled into batches;
2. Feature extraction: a series of convolution, normalization, and pooling layers, assembled according to the Inception-v4 architecture, with 2048 output features;
3. Logits: final linear, logistic neural network layer which connects the output features to a number of tensors equal to the number of output classes

Training was first performed in the test cases listed below for hyperparameter tuning. Training was accomplished using a modified gradient descent algorithm with exponential learning rate decay at 16-bit floating point resolution. Once training had completed, performance was evaluated by applying the trained model in a sliding-window fashion across the whole slide images used for the test case, visualizing model prediction with a heatmap overlay.

Initial hyperparameters, including tile size, apparent magnification, and batch size were determined through iterative testing of Case 1 using this method. Additional hyperparameters, including initial learning rate and total number of epochs, were determined through iterative testing of Case 2. Hyperparameters were finalized through iterative testing of Case 3.

**Case 1**: Two classes; one benign adenoma v. one NIFTP

**Case 2:** Two classes; **f**ive benign adenomas v. five NIFTPs

**Case 3**: Three classes; three benign adenomas v. three NIFTPs v. three classic PTCs

Once hyperparameters are tuned, 70% of tiles from each whole slide image will be used to train a final model. Performance will be evaluated in three manners; first, by determining model accuracy when applied to the remaining 30% of the tiles. Second, the final model will be applied to all whole-slide images in a sliding-window fashion, with a heatmap overlay visualizing model predictions. Finally, the 30% of tiles reserved for evaluation will be analyzed using the dimensionality reduction visualization technique described below.

***Visualizing histopathologic variation with dimensionality reduction***

For the aforementioned evaluation tiles, final layer output values will be exported for later use. These output values will then be visualized using a machine learning based dimensionality reduction algorithm, either via t-SNE or UMAP.