GPU access :

```
$ ssh revanth@192.168.130.148
Password : lfovia123
```

View -> Show document outline

Week 1: Read different articles on usage of machine learning in field of medicine

Article : Predicting non-small cell lung cancer prognosis by fully automated microscopic pathology image features

Notes:

Seven classifiers to conduct supervised machine-learning:

1) Naive Bayes
2) SVM with Gaussian kernel
3) SVM with linear kernel
4) SVM with polynomial kernel
5) Bagging for classification trees
6) Random forest utilizing conditional inference trees
7) Breiman's random forest

Process

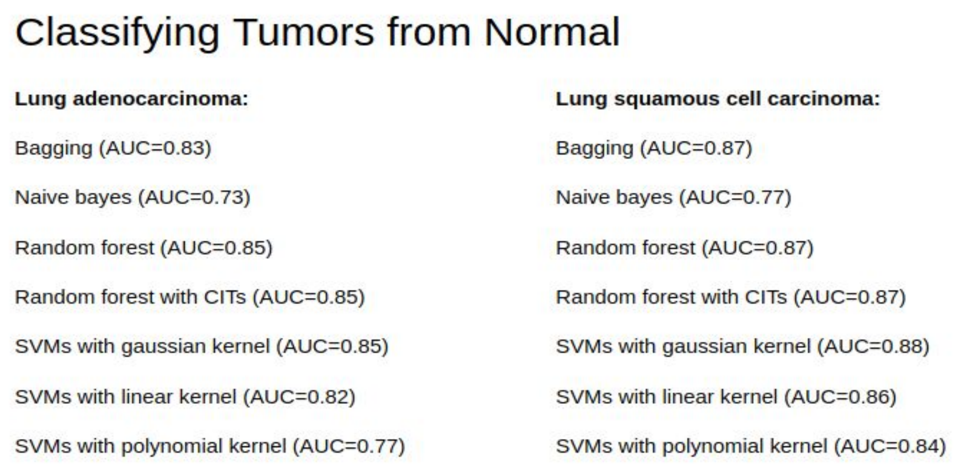
1. select the 10 densest tiles per image for further analysis.
2. fully automated image-segmentation pipeline to identifies the tumour nuclei and tumour cytoplasm from the histopathology images using the Otsu method and total of 9,879 quantitative features were extracted from each image tile with CellProfiler.
3. Types of image features included cell size, shape, distribution of pixel intensity in the cells and nuclei, as well as texture of the cells and nuclei.
4. net-Cox proportional hazards models to select the most informative quantitative image features which selected selected 15 features and calculated survival indices derived from H&E stained microscopic pathology images. Patients were categorized into longer-term or shorter-term survivors based on their survival indices.

Top Features for Classification

Top quantitative features were Haralick features of the nuclei, edge intensity of the nuclei, texture features of the cytoplasm and intensity distribution of the cytoplasm.

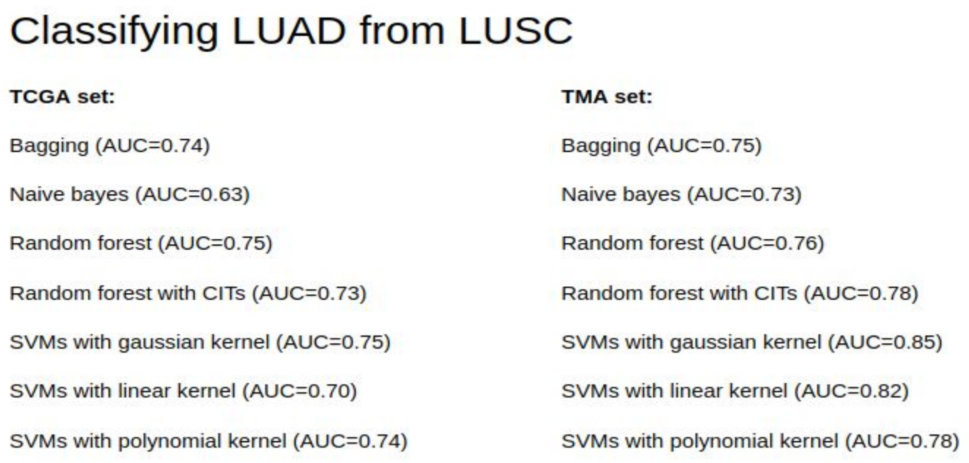
Top Features for Survival Index

Top features that facilitated classification of survival outcomes included texture of the nuclei, Zernike shape decomposition of the nuclei, and Zernike shape decomposition of the cytoplasm.



Classifying Tumors from Normal

Lung adenocarcinoma:	**Lung squamous cell carcinoma:**
Bagging (AUC=0.83)	Bagging (AUC=0.87)
Naive bayes (AUC=0.73)	Naive bayes (AUC=0.77)
Random forest (AUC=0.85)	Random forest (AUC=0.87)
Random forest with CITs (AUC=0.85)	Random forest with CITs (AUC=0.87)
SVMs with gaussian kernel (AUC=0.85)	SVMs with gaussian kernel (AUC=0.88)
SVMs with linear kernel (AUC=0.82)	SVMs with linear kernel (AUC=0.86)
SVMs with polynomial kernel (AUC=0.77)	SVMs with polynomial kernel (AUC=0.84)



Classifying LUAD from LUSC

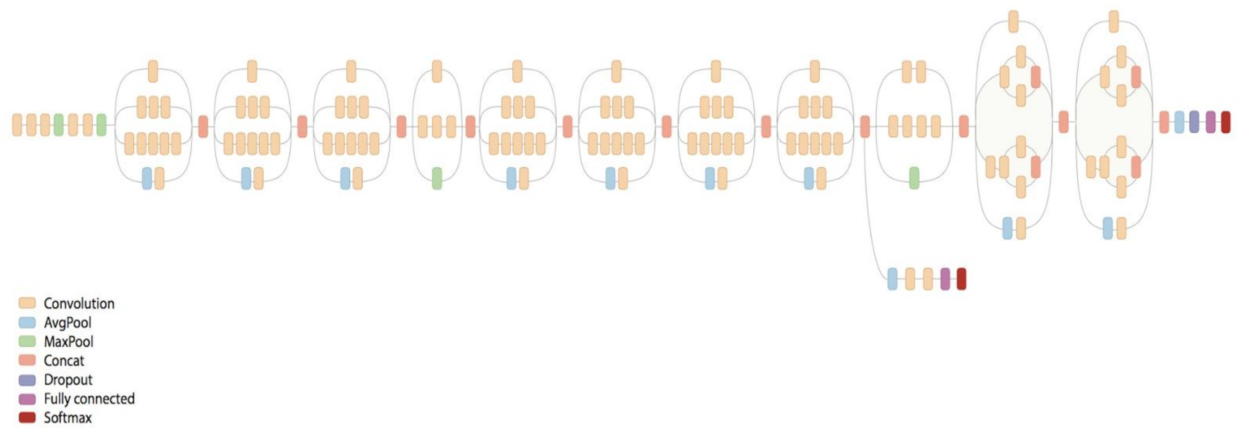
TCGA set:	**TMA set:**
Bagging (AUC=0.74)	Bagging (AUC=0.75)
Naive bayes (AUC=0.63)	Naive bayes (AUC=0.73)
Random forest (AUC=0.75)	Random forest (AUC=0.76)
Random forest with CITs (AUC=0.73)	Random forest with CITs (AUC=0.78)
SVMs with gaussian kernel (AUC=0.75)	SVMs with gaussian kernel (AUC=0.85)
SVMs with linear kernel (AUC=0.70)	SVMs with linear kernel (AUC=0.82)
SVMs with polynomial kernel (AUC=0.74)	SVMs with polynomial kernel (AUC=0.78)

Week 2: Read and understand the nature article

Article : Classification and mutation prediction from non–small cell lung cancer histopathology images using deep learning

Notes:

Inception v3 Network



The overall process is:

1. tile the svs images and convert into jpg
    a. SVS image is too large for the network
    b. Tile at different magnifications
2. sort the jpg images
    a. Sort tiles into train, test and validation sets with appending names
    b. Sort them into Solid_Normal_Tissue, TCGA-LUAD and TCGA-LUSC folders
3. convert each of the sets into TFRecord format
    a. TFRecord acts as input for Tensorflow networks
4. run training and validation
    a. Training can be done directly or through transfer learning from google datanet checkpoint
5. run testing
    a. Classifier testing is based on AUCs
6. run ROC curve & heatmap
    a. Heatmap gives a picture of how tiles of slides classified

Learn about working of convolution networks and Deep Neural Networks
https://www.youtube.com/playlist?list=PLkDaE6sCZn6Gl29AoE31iwdVwSG-KnDzF
https://www.youtube.com/playlist?list=PLZHQObOWTQDNU6R1_67000Dx_ZCJB-3pi

Week 3: Downloaded the images into Remote server

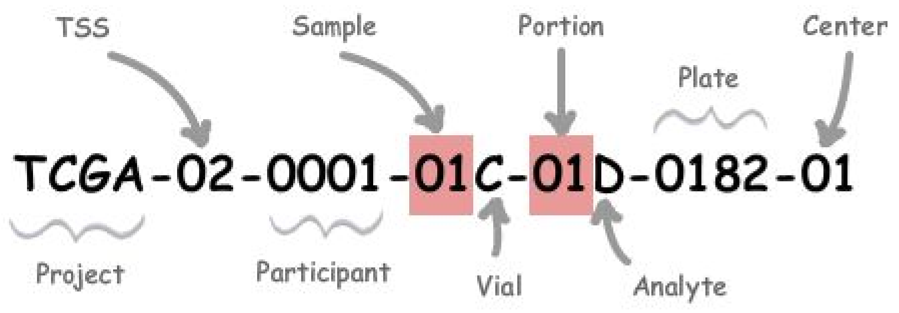
Visit https://portal.gdc.cancer.gov/legacy-archive/search

After adding filters
https://portal.gdc.cancer.gov/legacy-archive/search/f?filters=%7B%22op%22:%22and%22,%22content%22:%5B%7B%22op%22:%22in%22,%22content%22:%7B%22field%22:%22cases.project.primary_site%22,%22value%22:%5B%22Lung%22%5D%7D%7D,%7B%22op%22:%22in%22,%22content%22:%7B%22field%22:%22cases.project.program.name%22,%22value%22:%5B%22TCGA%22%5D%7D%7D,%7B%22op%22:%22in%22,%22content%22:%7B%22field%22:%22files.access%22,%22value%22:%5B%22open%22%5D%7D%7D,%7B%22op%22:%22in%22,%22content%22:%7B%22field%22:%22files.data_format%22,%22value%22:%5B%22SVS%22%5D%7D%7D%5D%7D

Process :
1) Install GDC Data Transfer Tool
2) Add few Slides to cart and download manifest
3) Use manifest as input for GDC Data Transfer Tool to download the selected slides
4) Download metadata (.json) file for the use of project
    a) It has all the details of a particular slide
        i) Cancer_type, stage
        ii) patient details etc

Breakdown of GDC naming of the slide:

TCGA-55-7573-11A-01-TS1.efb9eb21-c1f8-4abf-be2e-3ecfee65c153.svs



Project	Project name - TCGA
TSS	Which laboratory and study
Participant	Study participant
Sample	Sample type Tumor (01 - 09), normal (10 - 19) and control samples (20 - 29).
Vial	Order of sample in a sequence of samples
Portion	Order of portion in a sequence of 100 - 120 mg sample portions 1 The first portion of the sample 01-99

Analyte	Molecular type of analyte for analysis	D	The analyte is a DNA sample	
Plate	Order of plate in a sequence of 96-well plates	182	The 182nd plate	
Center	Sequencing or characterization center that will receive the aliquot for analysis	1	The Broad Institute GCC	**See Code Tables Report**

Week 4: Get the code running

00_preprocessing : Tiling, sorting and TFRecord conversion
01_training : Transfer learning or Training from random

00_preprocessing

Tiling :

```
$ python
/home/revanth/DeepPATH/DeepPATH_code/00_preprocessing/0b_tileLoop_dee
pzoom4.py  -s 299 -e 0 -j 32 -B 25 -o /home/revanth/output
"/home/revanth/images/*/*svs"
```

Inputs :

s (tile_size) : 299*299 , e (overlap): 0 , -j (num_of_threads): 32, -B (Percentage of Background allowed) : 25% , -o(path where output images are saved),

Output :

svs images are tiled in output/file_name/magnifications/*.jpeg files

Process :

1) Tiles the svs images at magnifications in multiples of 5
2) Converts the tile to grayscale and calculate percentage of background (White)
3) Discard the tile if it crosses the given value else retain it in original colour.
4) Named : valid_<Slide_name>_row_column

Sorting :

```
python
/home/revanth/DeepPATH/DeepPATH_code/00_preprocessing/0d_SortTiles.py
--SourceFolder=/home/revanth/output
--JsonFile=/home/revanth/Downloads/metadata.json --Magnification=20
--MagDiffAllowed=5 --SortingOption=3 --PercentTest=20
--PercentValid=15 --PatientID=12 --nSplit 0
```

Inputs

--SourceFolder(Directory of Tiled jpegs) : /home/revanth/output

--JsonFile (Directory to Json file) : /home/revanth/Downloads/metadata.json
--Magnification (Magnification of tiles to be selected) : 20
--MagDiffAllowed (If given magnification doesn't exist, take one at +/) : 5
--SortingOption (Based on what you wnat to classify) : 3 (Normal, LUAD, LUSC)
--PercentTest,PercentValid (Percentage of slides for Test and validation)
--PatientID (No. of digits representing a slide name) : 12 --nSplit 0

Output

Three folders named Solid_Tissue_Normal, TCGA-LUAD, TCGA-LUSC with tiles inside named <train/test/valid>_<Tile_name>

Process :

1) Select the tiles from given magnification
2) Sort the tiles into train, test and validation sets based on given percentages
3) Tiles of a single slide goes completely to one of the sets
4) Append the <train/test/valid>_ at the start

TFRecord conversion :

For Training and validation sets

```
python
/home/revanth/DeepPATH/DeepPATH_code/00_preprocessing/TFRecord_2or3_C
lasses/build_image_data.py --directory='/home/revanth/3_sort'
--output_directory='/home/revanth/3_tfrec' --train_shards=1024
--validation_shards=128 --num_threads=4
```

For Test set

```
python
/home/revanth/DeepPATH/DeepPATH_code/00_preprocessing/TFRecord_2or3_C
lasses/build_TF_test.py --directory='/home/revanth/3_sort'
--output_directory='/home/revanth/3_test' --num_threads=1
--one_FT_per_Tile=False --ImageSet_basename='test'
```

Inputs:

--directory (Directory of the tiles) : /home/revanth/3_sort
--output_directory (Directory to output TFReords) :
/home/revanth/<3_tfrec.3_test>

Output

1. TFReord would store the label, size, format etc of each tile
2. These would be in a format comfortable to tensorflow model

01_training

```
bazel build inception/imagenet_train
```

Bazel builds a make file for gpu optimisation.

Training : start with random values in kernels

```
   bazel-bin/inception/imagenet_train --num_gpus=1 --batch_size=10
--train_dir='/home/revanth/3_train' --data_dir='/home/revanth/3_tfrec'
--ClassNumber=3 --mode='0_softmax'
```

Transfer Learning : start with kernel values of imagenet database

```
bazel-bin/inception/imagenet_train --num_gpus=1 --batch_size=30
--train_dir='/home/revanth/transfer' --data_dir='/home/revanth/3_tfrec'
--pretrained_model_checkpoint_path="/home/revanth/DeepPATH/DeepPATH_code/01
_training/xClasses/inception-v3/model.ckpt-157585" --fine_tune=True
--initial_learning_rate=0.001  --ClassNumber=3 --mode='0_softmax'
```

Inputs :

--batch_size (No. of images for each batch)
--train_dir (Directory to store the checkpoint)
--data_dir (Directory to TFRecord of training tiles)
--ClassNumber (No of classes for final output) : 3 (Normal,LUAD and LUSC)
--mode (Error correction mode) : 0_softmax (softmax function)

Output:

Checkpoint will be stored with parameters of network at data_dir

Process :

1. Evaluate through the network
2. Backpropagate and add up the error of each parameter in each tower individually
3. Change the parameters after each batch considering all towers at a time
4. Store the checkpoint after every 5000 steps and at max_steps-1

Week 5: Work on errors and new approach

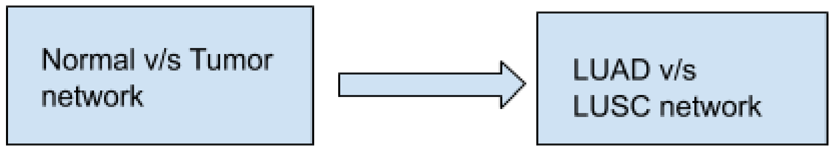
Error was due to abrupt stopping of training and corrupting the checkpoint
Retrained the network now without using transfer learning and setting max_steps in inception_train.py to 50000.

Memory of Remote system is completely filled
Solved after few days by Research students when they deleted some files

New Approach :

Article says a two step classifier (Normal vs cancer -> LUAD vs LUSC) proved to be less productive.

Our network is also a two-step classifier but during the training of the first network, not all the tiles are labelled same as their slides (tumour or normal) as in the paper but only the selected regions of the slide which have that behaviour are manually labelled and trained. Then we give an abnormal score for the whole slide and from there we eliminate the normal slide from the tumour slide and further train it for the LUAD v/s LUSC network.



Week 6: Final output

Testing include calculating per tile stats, per slide stats, AUCs and heatmaps for visualisation.

02_testing

```
python3
/home/revanth/DeepPATH/DeepPATH_code/02_testing/xClasses/nc_imagenet_eval.p
y --checkpoint_dir='/home/revanth/3_train'
--eval_dir='/home/revanth/3_output' --data_dir="/home/revanth/3_test"
--batch_size=10 --ImageSet_basename='test_' --run_once --ClassNumber 3
--mode='0_softmax' --TVmode='test'
```

Inputs :

--checkpoint_dir (Directory to the checkpoints)
--eval_dir (Directory to store the output of pertile stats)
--data_dir (Directory to TFRecord of test tiles)
--batch_size (No. of tiles to be tested at once befores calculating loss)
--ImageSet_basename (base name of test tiles)
--run_Once (Run only once not included while validating)
--ClassNumber (No of classes for final output) : 3 (Normal,LUAD and LUSC)
--mode (Error correction mode) : 0_softmax (softmax function)
--TVmode (Indicates validation or test)

Output:

Pertile statistics in the output directory

Process :

1. Evaluate through the network
2. Store the probabilities

A detailed description of the output we achieved after testing tiles for the direct 3-way classifier:

Output_stats_pertile.txt : (All the tiles sent for testing are reported in this file)
<u>Sample data values in the above files for four of the slides:</u>
<u>Actually normal labelled tiles</u> : (Code below for the slide is 11A)

Each line will be format

<TFRecord_name> <True/False> <[Prob. of background class, <Probabilities of each class>]> <probability of output class excluding background class> labels: <True label>

<TFRecord_name>
<True/False> : If Prediction is same as given
<[Prob. of background class, Probabilities of each class]>
<probability of output class excluding background class> : ignoring background class
<True label>

1) **test_TCGA-18-3417-11A-01-BS1.31d8bc6b-d57e-4f1d-9da8-f6baa090f882_16_22.dat False [0.02512684 0.13852204 0.32200715 0.514344] 0.14209237473191808 labels: 1**
   test_TCGA-18-3417-11A-01-BS1.31d8bc6b-d57e-4f1d-9da8-f6baa090f882_16_22.dat
   False (Corresponds to classification of true/false for the above tile)
   [0.02512684 0.13852204 0.32200715 0.514344]
   Output probabilities with 1st one being the inception's background class. And the remaining three probabilities for the types-(normal, LUAD, LUSC)
   0.14209237473191808
   corrected output probability for the true label - adjusted to ignore the background class
   labels: 1
   True label for the tile. Since the probability corresponding to the true label is not maximum,
   it says that our predictions are false.

2) **test_TCGA-18-3417-11A-01-BS1.31d8bc6b-d57e-4f1d-9da8-f6baa090f882_16_23.dat**
   True (Here it classifies as true)
   [0.02419432 0.56635374 0.13497548 0.27447653]
   Output probabilities same as above mentioned.
   0.580395987094186
   Correct output probability same as above mentioned.
   labels: 1
   True label for the file. Since the probability corresponding to the true label is maximum, it says that our predictions are true.

Actually cancer labelled tiles : (Code below for the slide is 01A)

1) **test_TCGA-97-8547-01A-01-TS1.cf6c29cd-1bf5-41f4-b008-d3e47aeff4ac_14_35.dat**
   False
   [0.02506102 0.18227097 0.30422345 0.4884446]
   0.3120435660785562
   labels: 2
   This tile true label is 2, but probability corresponding to the label 2 is not maximum.So prediction is false.

2) **test_TCGA-63-7020-01A-01-TS1.85055591-e148-4fe3-ae1f-2d55f84f04be_14_39.dat**
   True
   [0.0253623 0.11213965 0.3174582 0.5450399]
   0.5592230444732208
   labels: 3
   This tile true label is 3 and since probability corresponding to the label 3 is maximum,the prediction is true.

Code for ROC curves :

```
python3
/home/revanth/DeepPATH/DeepPATH_code/03_postprocessing/0h_ROC_MultiOutput_B
ootStrap.py  --file_stats  /home/revanth/3_output/out_filename_Stats.txt
--output_dir /home/revanth/3_roc --labels_names
/home/revanth/label_names.txt --ref_stats ' '
```

Inputs :
   --file_stats (Directory to the per tile stats file)
   --output_dir (Directory to store the ROC and perslide stats)
   --labels_names (Directory to file having names of the labels)
Output:
   Perslide statistics and ROC curves with respective AUCs
Process :
   Plot different ROCs with varying the threshold for classification with Average probability and Percentile selected and calculate AUCs and confidence interval.

The above command generates statistics in file names as follows :

```
out1_perTile_roc_data_AvPb_c1auc_0.9155_CIs_0.9102_0.9202_t0.221980.txt
out1_perTile_roc_data_AvPb_c2auc_0.4671_CIs_0.4565_0.4772_t0.267484.txt
out1_perTile_roc_data_AvPb_c3auc_0.7436_CIs_0.7352_0.7523_t0.555898.txt
```

```
out1_perTile_roc_data_AvPb_macro_auc_0.7088_CIs_0.7038_0.7137.txt
out1_perTile_roc_data_AvPb_micro_auc_0.7290_CIs_0.7237_0.7341.txt
out1_perTile_roc_data_PcSel_c1auc_0.7521_CIs_0.7432_0.7604.txt
out1_perTile_roc_data_PcSel_c2auc_0.4999_CIs_0.4998_0.5000.txt
out1_perTile_roc_data_PcSel_c3auc_0.6531_CIs_0.6472_0.6589.txt
out1_perTile_roc_data_PcSel_macro_auc_0.6351_CIs_0.6305_0.6392.txt
out1_perTile_roc_data_PcSel_micro_auc_0.6647_CIs_0.6595_0.6698.txt
out2_perSlideStats.txt
out2_roc_data_AvPb_c1auc_0.9356_CIs_0.8507_0.9968_t0.305405.txt
out2_roc_data_AvPb_c2auc_0.6272_CIs_0.4643_0.7857_t0.257720.txt
out2_roc_data_AvPb_c3auc_0.8305_CIs_0.6893_0.9467_t0.540061.txt
out2_roc_data_AvPb_macro_auc_0.8163_CIs_0.7291_0.8787.txt
out2_roc_data_AvPb_micro_auc_0.7658_CIs_0.6832_0.8432.txt
out2_roc_data_PcSel_c1auc_0.9211_CIs_0.8365_0.9824.txt
out2_roc_data_PcSel_c2auc_0.4868_CIs_0.4583_0.5000.txt
out2_roc_data_PcSel_c3auc_0.8629_CIs_0.7103_0.9715.txt
out2_roc_data_PcSel_macro_auc_0.7659_CIs_0.6953_0.8128.txt
out2_roc_data_PcSel_micro_auc_0.7148_CIs_0.6168_0.8122.txt
```

Each file except out2_perSlideStats.txt contains x y points with False positive rate on X and True positive rate on Y

It will generate files starting with "out1" for non aggregated per tile ROC, and files starting with "out2" for per slide aggregated ROC curve. AUC will be show in the filename. File names of the outputs:

- start with out1 if the ROC are per tile
- start with out2 if the ROC are per slide
- then contain if the per slide aggregation was done by averaging probabilities
- or if the aggregation was done by computing the percentage of tile selected
- then, the names end with something like ....c1auc_0.6071_CIs_0.6023_0.6121_t0.367.txt -> c1 (or c2 or c3...) means class 1 -> auc_0.6071. is the AUC for this class (if you have only 2 classes, the curves and AUC should be the same)
- the next two numbers are the Confidence Intervals.
- the last one with the "t" is the "optimal" threshold for this class (computed such as it's the nearest point on the ROC curve to the perfect (1,0) corner).

For example, one of the file is
`out1_perTile_roc_data_AvPb_c1auc_0.9155_CIs_0.9102_0.9202_t0.221980.txt`
The above file says it has got AUC value of 0.9155 for class1 and confidence interval for the mentioned class is 0.9102 to 0.9202.

ROC curve for multi output classification is calculated as below:

(1) Normal vs LUAD and LUSC,

(2) LUAD vs Normal and LUSC,
(3) LUSC vs Normal and LUAD.
Then the AUC is averaged over the three curves.

Each of the above file contain two values (x,y) co-ordinates which corresponds to mapping in the ROC curve. It also contains the out2_perSlideStats.txt in which one of the line is :

```
test_TCGA-66-2792-11A-01-TS1.fb255c48-b47f-45b1-9f04-5107b8c16e4e_0100
true_label: [1.0, 0.0, 0.0]
Percent_Selected: 0.301299    0.002597    0.696104
Average_Probability: 0.306132    0.245624    0.448244
tiles#: 385.000000
```

Above block is just for one slide and all the details regarding that slide which says that true label is Normal for above and percent tiles selected of each predicted type in the slide and average probability of the tiles of respective predicted types and the total number of tiles in the slide.
Every line looks line below :
<Slide_name>
true_label <[0/1, 0/1, 0/1]>
Percent_Selcted: No. of tiles predicting a class / Total tiles
Average_Probability: Averaging probabilities of each class over all the tiles
tiles# : No. of tiles for this slide

Heat maps:

```
python
/home/revanth/DeepPATH/DeepPATH_code/03_postprocessing/0f_HeatMap_nClasses.
py  --image_file '/home/revanth/3_sort' --tiles_overlap 0 --output_dir
'/home/revanth/3_heat' --tiles_stats
'/home/revanth/3_output/out_filename_Stats.txt' --resample_factor 10
--slide_filter 'TCGA-' --filter_tile '' --Cmap 'CancerType' --tiles_size
299
```

Inputs :
--image_file (Directory to the Sorted tiles)
--tiles_overlap (Overlapping pixels given while tiling)
--output_dir (Directory to store the Heatmap of the slides)
--tiles_stats (Directory to the file having per tile statistics)
--resample_factor (Factor to reduce the quality of the image of heatmap)
--slide_filter (name the slide should start with)
--filter_tile (used to eliminate a label)
--Cmap (cancer type / mutations)
--tiles_size (Size of the tiles given at tiling part)

Output:

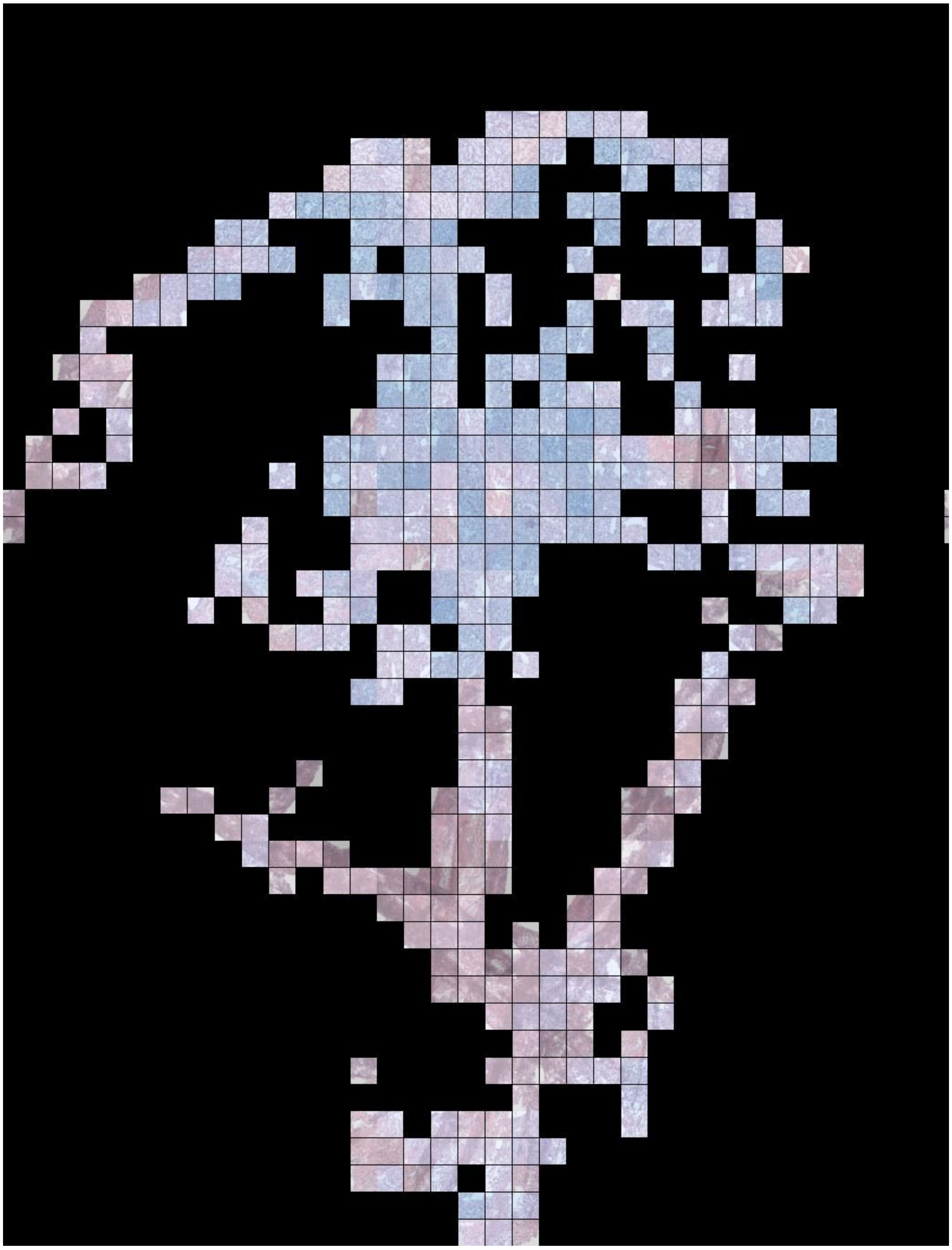
Image of the slides showing tiles tinted according to the labe;

Process :

1. Find the tile its position in the name
2. Its prediction is in tile_stats
3. Tint it with colour according to label
4. Generate the heatmap

The above command generates all the heatmaps for each of the slide and the colors are: black for class 1, red for class 2, blue for class 3, orange for class 4, green for class 5, purple otherwise

Here black is for Normal, red for LUAD and blue for LUSC, the tiles are painted with those colours proportional to the probabilities of those types which is evident below.



Winter break

https://drive.google.com/file/d/1L5Yudm5k4ZGejYWMgy8z6KYMGJl4y5mc/view

```
./gdc-client download -m gdc_manifest_20181204_064252.txt
```

```
python
/home/revanth/DeepPATH/DeepPATH_code/00_preprocessing/0b_tileLoop_deepzoom4.py  -s
299 -e 0 -j 32 -B 25 -o /home/revanth/new_tiles  "/home/revanth/new_svs/*/*svs"
```

```
python  /home/revanth/DeepPATH/DeepPATH_code/00_preprocessing/0d_SortTiles.py
--SourceFolder=/home/revanth/new_tiles
--JsonFile=/home/revanth/metadata.cart.2018-12-04.json --Magnification=20  --MagDiffAllowed=5
--SortingOption=3 --PercentTest=0 --PercentValid=0 --PatientID=12 --nSplit 0
```

Marking in Imagescope - https://youtu.be/nix1DMt9SIA