BIOINFORMATICS

Vol. 00 no. 00 2005
Pages 1–11

Supplementary Document - Data Distributions, Computing Time, Analysis and Methods

Ching-Wei Wang [1,2*], Yu-Ching Lee [2,3], Evelyne Calista [1,2], Fan Zhou [5], Hongtu Zhu [5,6], Ryohei Suzuki [7,8], Daisuke Komura [7], Shumpei Ishikawa [7], Shih-Ping Cheng [4]

[1]Graduate Institute of Biomedical Engineering, National Taiwan University of Science and Technology, Taiwan .
[2]NTUST Center of Computer Vision and Medical Imaging, Taiwan.
[3]Graduate Institute of Applied Science and Technology, National Taiwan University of Science and Technology, Taiwan.
[4]Department of Surgery, Mackay Memorial Hospital, Taipei, Taiwan.
[5]Department of Biostatistics, University of North Carolina at Chapel Hill, USA.
[6]Department of Biostatistics, University of Texas, MD Anderson Cancer Center, USA.
[7]Department of Genomic Pathology, Medical Research Institute, Tokyo Medical and Dental University, Tokyo, Japan.
[8]Department of Physics, The University of Tokyo, Tokyo, Japan

Received on XXXXX; revised on XXXXX; accepted on XXXXX

Associate Editor: XXXXXXX

Table 1. Overall data distribution

Images	#TMA	#patients	#cores
Training	14 (H&E) 13 (IHC)	102 (H&E) 93 (IHC)	368 (H&E) 318 (IHC)
Testing	14 (H&E) 13 (IHC)	51 (H&E) 47 (IHC)	182 (H&E) 162 (IHC)

Due to limited length of paper, this supplementary document describes the data distributions in section 1, the information about the computation time and hardware/software specifications of each method in section 2, analysis and discussion of the automated methods in section 3 and the details of the automated methods in section 4.

1 DATA DISTRIBUTIONS

Table 1 show the data distributions in training and testing with respect to the number of TMAS, patients and tissue cores, and the data distributions w.r.t. the cancer type, sex and hashimoto status are presented in Table 2, 3 and 4.

2 COMPUTER SPECIFICATION AND EFFICIENCY

1. Zhou and Zhu's method: The method was implemented in MATLAB for Step (I) of TMA-D^{2}LM, whereas Steps (II)

*to whom correspondence should be addressed

Table 2. Data distribution w.r.t. the cancer subtype

Cancer subtype	Training				Testing			
	IHC		H&E		IHC		H&E	
	#patients	#cores	#patients	#cores	#patients	#cores	#patients	#cores
11	65	222	73	263	36	120	38	136
12	10	36	10	36	4	16	5	16
13	1	3	1	3	0	0	0	0
25	6	18	7	26	2	8	3	12
26	5	16	5	17	2	7	2	7
30	5	19	5	19	2	8	2	8
40	1	4	1	4	1	3	1	3

Table 3. Data distribution w.r.t. sex

Sex	Training				Testing			
	IHC		H&E		IHC		H&E	
	#patients	#cores	#patients	#cores	#patients	#cores	#patients	#cores
Male	12	42	13	47	7	19	7	24
Female	75	253	83	298	37	132	41	147
Unknown	6	23	6	23	3	11	3	11

Unknown: the information is missing in the database.

Table 4. Data distribution w.r.t. the hashimoto status

Hashimoto	Training				Testing			
	IHC		H&E		IHC		H&E	
	#patients	#cores	#patients	#cores	#patients	#cores	#patients	#cores
0	84	287	89	321	41	139	44	157
1	3	8	7	24	3	12	4	14
Unknown*	6	23	6	23	3	11	3	11

*the information is missing in the database.

Table 5. Suzuki *et al.*'s runtime (Wall time) for hyperparameter optimization, training and testing of gradient boosting trees.

	Extension	N	stage	size
hyperparameter opti. (s)	974	1437	549	1007
training (ms)	0.068	0.023	0.02	0.023
test (ms)	18.3	21.3	18.9	35.3

and (III) are written in Python. Specifically, when training the dictionary model, Zhou and Zhu use the Python module 'Deep-Semi-NMF' (Trigeorgis *et al.*, 2017) built on Theano. Therefore, running the algorithm on GPU highly increases the computation efficiency. The computational complexity for the pre-training stage of Deep Semi-NMF is of order $O(mt(pnk+nk^2+kp^2+kn^2))$, where $m=5$ is the number of layers, t is number of iterations, $p=512^2$, and k corresponds to the maximum number of components out of all layers. The execution time of Algorithm 1 with 1,000 epochs takes less than 1 hour on GPU. Training the XGBoost model and doing the predictions take less than 10 minutes for all the five outcomes. Zhou and Zhu use the computer cluster Longleaf at University of North Carolina at Chapel Hill to store the data and run all the scripts. Longleaf is a brand new cluster explicitly designed to address the computational, data-intensive, memory-intensive, and big data needs of researchers and research programs that require scalable information-processing capabilities that are not of the MPI and/or OpenMP+MPI hybrid variety. Longleaf includes 117 'General-Purpose' nodes (24-cores each; 256-GB RAM; 2x10Gbps NIC) and 24 'Big-Data' nodes (12-cores each; 256-GB RAM; 2x10Gbps; 2x40Gbps), 5 large memory nodes (3-TB RAM each), 5 'GPU' nodes each with GeForce GTX1080 cards (102,400 CUDA cores in total) of 8-GB memory. All running jobs are done on either 'General-Purpose' nodes or 'GPU' nodes.

2. Suzuki *et al.*'s method: For BRAF mutation predictor, A quad-processor system with Intel Xeon E5-4617@2.9 GHz (6 cores), 512 GB RAM with a Tesla K20c GPU and 64-bit operating system (CentOS release 6.7) was used for the data preparation and network training of the BRAF predictor. Suzuki *et al.* trained the network with the training data (84,800 patch sets, 254,400 images) for 15 epochs using a Tesla GPU, and the runtime for the network training session was 25 hours 47 mins. It took 13 min 1 sec to calculate the prediction for all the testing data (25,600 patch sets, 76,800 images) by forward propagation. For predictors of size, extension, N and stage, the quad-processor system described above was used for the nuclei segmentation and calculation of nuclear features. Runtime using all the 24 cores for training data (2770 images) and test data (1340 images) was 41 min 28 sec and 14 min 50 sec, respectively. Next, MacBook Pro with Intel Core i7@2.2 GHz (4 cores), 16 GB RAM and 64-bit operating system (OS X El Capitan) was used for building classifier and the prediction. The implementation of building classifier and the prediction was in Python 2.7.12 (Anaconda 4.0.0). Runtime (wall time, single threaded) for each model is shown in Table 5.
3. Wang *et al.*'s method: Wang *et al.* use ImageJ 1.50i, Java(TM) SE runtime environment 1.8.0_77 (64 bit) as the compiler. All training and testing were executed on a computer with Intel Core i5 processor at 3.20 GHz and 16.0 GB RAM under the OS of Windows 10 Pro. The average runtime for one image to extract quantification value was 55 seconds. Furthermore, the computing time for training all 318 tissue core images and testing all 162 tissue core images of each parameter are presented at Table 6.

Table 6. Total computing time by Wang *et al.*'s method for training 318 images and testing 162 images

Computing time	BRAF	Stage	Extension	N	Size
Total Training (s)	0.08	0.59	0.13	0.23	0.01
Total Testing (s)	0.01	0.03	0.01	0.03	0.01

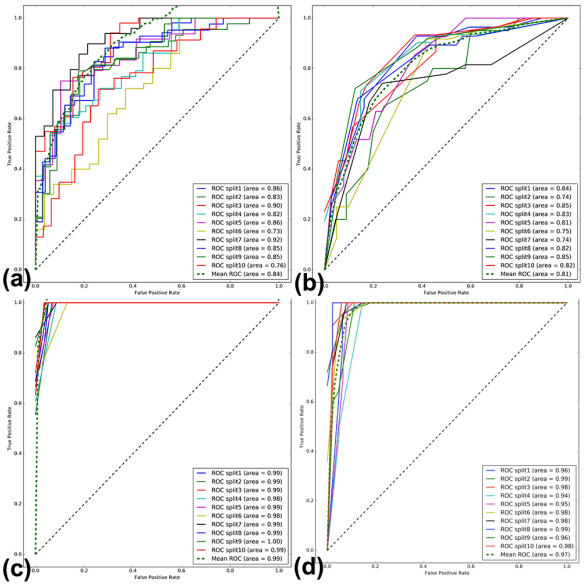


Fig. 1: ROC curves based on 10-times cross-validation of (a)BRAF prediction, (b)Extension, (c,d)2-class stage prediction: (stage 1,2) and (stage 3,4).

3 SUPPLEMENTARY METHOD ANALYSIS AND DISCUSSION

3.1 Zhou and Zhu: TMA-D^{2}LM

TMA-D^{2}LM uses image features and demographic covariates to predict 'BRAF' and 'Extension' first, and then predicts 'Size' and 'N' with the prediction of first two outcomes. Finally, it predicts 'Stage' by using the predictions of all the other four responses. Next, the patient-level prediction will be voted by the image-level predictions (images from same patient have same demographic covariates and 'BRAF', 'Extension' status).

The training data were randomly split into train and test set for calculate the prediction accuracy. The preliminary test for BRAF results show that the highest prediction accuracy can reach 85%. The accuracy of patient-level predictions, which is voted by the image-level predictions, can be slightly better. Figure 1(a) shows the ROC curves based on 10-fold cross-validation analysis.

The prediction of 'Extension' is similar to that of 'BRAF' except that the most discriminative patches are selected by projecting $H\&E$ tissue microarrays onto the 'extension=0' space. XGBoost is still used to predict 'Extension' and 'N'. To visualize the model performance, we combine 'Extension=1' and 'Extension=2' and plot the corresponding ROC curves (see Figure 1(b)). In this case, the highest prediction accuracy can reach 0.82.

In 'Size' prediction, we assign scores to the continuous 'Tumor size' according to five groups. The first group includes those with tumor size smaller than 1.2, the second one between 1.2 and 2, the third one between 2 and 3, the fourth one between 3 and 4 and the fifth one with size bigger than 4. We use 1, 1.5, 2.5, 3.5 and 5 as the 'size' value for observations fall into the corresponding group. Then, we also use XGBoost to do projection. Due to the transformation and large number of missing values, the highest accuracy for 'Size' is 0.72.

Cancer stage is the last response for prediction. For comparison, we will use the true values of 'N', 'Extension', and 'Size' to test the finite sample performance of our predictive model by splitting the training data set. Based on the true values, the best accuracy can be higher than 95%. However, based on the predicted values of the other four responses, the highest accuracy can also reach 94%. Moreover, we divide the four stages into two classes: (stage 1,2) and (stage 3,4), and perform the 10-fold cross-validation and present the ROC curves corresponding to both true and predicted values in Figure 1(c) and (d).

For future works, there are many potential improvements in the future. First, we will add convolutional layers before running the deep dictionary learning model in order to avoid losing some importance shape features. Second, we will explore other methods to find the center of the 'normal' space, which may give more accurate results. Third, for the continuous responses like 'size', we will develop better regression methods for predicting 'tumor size'.

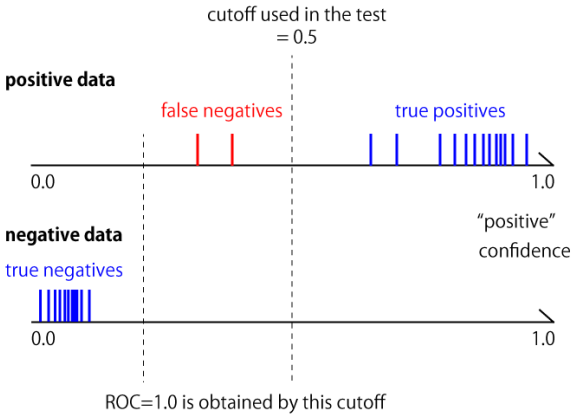


Fig. 2: ROC cutoff in the preliminary test of Suzuki *et al*'s BRAF predictor.

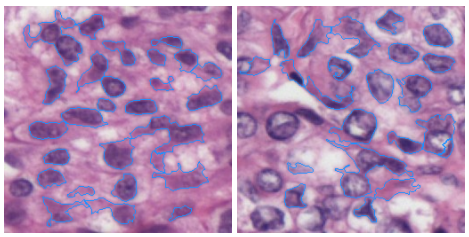


Fig. 3: Segmentation results of the given dataset (left: HE02_016_T1, right: HE03_025_T1). Each nucleus is indicated by a closed contour with light blue color.

3.2 Suzuki *et al*: Hybrid Prediction

3.2.1 BRAF predictor-Preliminary Test We trained the prediction model using 148 tumor slides and validated it with 64 tumor slides. The validation result is shown in Table 7 compared with unmodified GoogLeNets that take an image of single magnification level (224px, 448px, and 896px in the original slide images) as the input. The model demonstrated 91.0% patch-wise prediction accuracy, 95.3% slide-wise prediction accuracy and 95.7% patient-wise accuracy, respectively. It surpasses the unmodified GoogLeNets of all magnification levels in both the accuracy and AUC values. In Table 7, accuracies of e.g. 95.7% imply AUC $<$ 1.0 and yet AUC$=$ 1.0, which is because that all of the negative data were correctly judged with high confidence (e.g. 99% negative) and some positive data were judged as "negative" with weak confidence (e.g. 70% negative) by the proposed CNN model. An illustration is given in Figure 2.

3.2.2 Other clinical diagnoses – training data preparation Nuclei segmentation and feature extraction from HE stained slides of tumor were performed using CellProfiler version 2.2.0 (Carpenter *et al*., 2006). Since we could not apply these analysis to whole slide images directly due to memory and computational time problem, we performed the analysis to randomly sampled mini patches and summarized the features for each nucleus into representative features for each case. First, all the whole slide images labelled as tumor were split into non-overlapping 256 $\times$ 256 px mini patches. After removing patches containing large background regions, we randomly picked 10 patches for each tumor slide.

Then hematoxylin stain image for each HE stain patch was obtained using the 'UnmixColors' module, and 'IdentifyPrimaryObjects' module with adaptive Otsu thresholds was applied to identify the cell nuclei. Next, 60 element features were calculated using 'Measure Object Intensity' and 'Measure Object Size Shape'. The quantitative features covered the size, shapes including Zernike shape features, pixel intensity distributions. These features were aggregated across the case by calculating median and median absolute deviations (MAD) of the values. We did not use mean and standard deviations, which are less robust to outliers compared to median and MAD, for aggregation because we observed intersecting nuclei was often segmented as a single nucleus as shown in Figure 11.

3.2.3 Other clinical diagnoses – prediction model We applied extreme gradient boosting trees (Chen and Guestrin, 2016) using xgboost package version 0.6 [1] called from Python for the classification or regression tasks. Linear regression model was used for the prediction of size, while classification models were used for

[1] `https://pypi.python.org/pypi/xgboost/0.6`

Table 7. Preliminary test of Suzuki et al's BRAF-mutation prediction model compared with unmodified GoogLeNets with various patch magnification levels.

	patch-wise		slide-wise		patient-wise	
model	accuracy (%)	AUC	accuracy (%)	AUC	accuracy (%)	AUC
GoogLeNet (224 px)	84.5	0.91	85.9	0.97	87.0	**1.0**
GoogLeNet (448 px)	86.9	0.94	89.1	0.98	**95.7**	**1.0**
GoogLeNet (896 px)	87.9	0.95	85.9	0.98	**95.7**	0.99
our model	**91.0**	**0.97**	**95.3**	**0.99**	**95.7**	**1.0**

the other discrete ordinal variables such as extrathyroidal extension and lymph node metastasis instead of regression because noise distribution does not follow Gaussian distribution (Van den Oord *et al*., 2016). Multiclass classification using the softmax objective was used for the classification tasks. Estimated BRAF status, sex, age, hashimoto, BMI, BW, BH, the cancer type and 120 nuclei features from tumor HE stained slides described above were used as features.

We optimized hyperparameters of XGBoost using Bayesian optimization (over a validation set different from the final test set). BayesianOptimization package[2], a python implementation of global optimization with gaussian processes, was used for the purpose. These parameters included the number of trees to train, the maximum depth of each decision tree, and the minimum weight allowed on each decision leaf, the data subsampling ratio, and the minimum gain required to create a new decision branch.

3.2.4 5 fold Cross Validation Test We performed 5-fold cross validation for the training data. The predictive accuracy for tumor size measured by the mean absolute error and those for extrathyroidal extension, lymph node metastasis and TNM stage measured by prediction accuracy were listed in Table 8. We also calculated predictive accuracy simply using mean for size or majority class for the others among the training samples as baseline. Surprisingly, only accuracy of stage was significantly better than that of the baseline, which indicated that our features were useless for the prediction of extrathyroidal extension, lymph node metastasis and TNM stage.

Gradient boosting trees measure feature importance by F-score, which is the number of times a feature appears in a tree. For TNM stage, which was the only parameter successfully predicted, only age was used for the prediction. The results indicate that the model could learn this rule automatically, but it failed to find nuclear features useful for the prediction of TNM stage.

Table 8. 5-fold cross validation for the training data on Suzuki et al's model.

Predicted variable	Suzuki et al	mean or majority class
size (MAE)	1.37 ± 0.84 (cm)	0.89 (cm)
extension	43.7 ± 15.2 (%)	56.3 (%)
metastasis	56.3 ± 6.2 (%)	53.1 (%)
stage	$\mathbf{78.2 \pm 9.0}$ **(%)**	59.4 (%)

Advantages of Suzuki *et al.*'s approach using gradient boosting trees with Bayesian Optimization are three folds: 1) handle heterogeneous features including categorical, discrete and continuous variable in a unified manner, 2) calculate feature importance, 3) fully automatic tuning of hyperparameters. However, Suzuki *et al.*'s approach failed to predict the values using image features. There are two possibilities for the failure. One is that HE images of thyroid cancer is intrinsically useless for the prediction. It is possible because even human pathologists may not be able to predict these parameters using only HE images of tissue microarray in thyroid cancer. The other is nuclear image features that Suzuki *et al.*'s method to extract was incomplete or image features other than nucleus such as cytoplasm are useful. Nuclear segmentation is a challenging task since such as color variations in tissue appearance, occulusions, inclusion of nuclei of non-tumor cells would affect the performance of the segmentation. More sophisticated algorithms for stain normalization (Khan *et al*., 2014), nuclear segmentation (Irshad *et al*., 2014), or feature representations using deep learning could lead to more accurate results.

[2] `https://github.com/fmfn/BayesianOptimization`

3.3 Wang *et al*: Ensemble

Wang *et al.*'s method is demonstrated to be promising in prediction of BRAF mutation and provide acceptable prediction accuracy in stage and relative high correlation score in estimating tumor size. However, as the method does not utilize the morphological patterns in H&E, the method has limitations in prediction of the clinical outcomes, which relate to tissue morphology, such as the Extension, N and size. For future improvements, it is expected that the model may produce better and more reliable predictions outcomes for all five parameters by integration of morphological features extracted from H&E images.

4 SUPPLEMENTARY METHODS

This section describes the three automated methods in thyroid cancer diagnosis using tissue microarrays and patient background information.

- Zhou and Zhu, TMA-D^{2}LM: Tissue Microarray Analysis via A Deep Dictionary Learning Method (USA).
- Suzuki *et al*., Hybrid Prediction Model for Thyroid Cancer Diagnosis (Japan).
- Wang *et al.* Ensemble Machine Learning Based Approaches for Thyroid Cancer Diagnosis (Taiwan).

4.1 TMA-D^{2}LM: Tissue Microarray Analysis via A Deep Dictionary Learning Method

Zhou and Zhu developed an TMA analysis model using deep dictionary learning method (TMA-D^{2}LM). Zhou and Zhu's algorithm consists of three steps: 1) pre-processing and segmentation, 2)

feature extraction and 3) predictive model building. The path diagram of TMA-D^{2}LM is shown at Figure 4. The pre-processing step is used for remove outlying tissues with staining ingredient, dust or cracked glass. In this step, Otsu's thresholding, downsampling and segmenting are performed. Zhou and Zhu downsample the tissue parts of all images from the original dimension $10^4 \times 10^4$ to 2500×2500 and then segment them into many small 512×512 squared patches (Xu *et al.*, 2016). Patches with more than 40% non-tissue pixels are dropped.

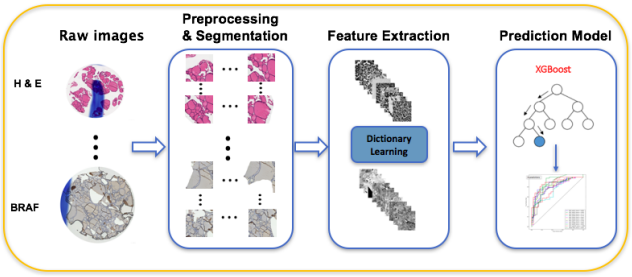


Fig. 4: The path diagram of TMA-D^{2}LM consists of : 1) pre-processing and segmentation, 2) feature extraction and 3) predictive model building

For feature extraction, a deep dictionary learning method (Trigeorgis *et al.*, 2017) is used to find the most discriminative patch from each image and build a low dimensional representation of the selected patch to denote the features of the 'mother' images. For example, for the BRAF tissue microarray images, all the patches extracted from the images with BRAF$=0$ are chosen, producing a low dimensional representation for each patch by applying a five-layer dictionary learning model. It allows to build a 'BRAF $= 0$' ('normal') space. Specifically, the 'normal' space is then constructed by all the patches which are classified into the bigger class by running a 2-class K-Means on all subjects in the low-dimension space. Subsequently, the method projects all the patches corresponding to all BRAF $= 0$ and $= 1$ images onto the 'normal' space and finds the most discriminative patch of each image, which has the longest distance from the center of the 'normal' space. Since all patches are already mapped onto a low-dimensional space, the method will use the low-dimensional representation of the 'discriminative' patches, which is a $k \times 1$ vector where k is much smaller than the original patch dimension, in order to represent the key features of their 'mother' images.

To run the deep dictionary learning algorithm (Trigeorgis *et al.*, 2017), the method firstly transforms each individual patch from the RGB color space into the grayscale space and then reorders the original 512×512 pixels into a single vector $\mathbf{x}$ of length $p = 512^2$. The RGB-to-grayscale conversion is performed by computing a weighted sum of the R, G and B components of the color image according to $0.2989 \times R + 0.5870 \times G + 0.1140 \times B$ (Linder *et al.*, 2012). These are applied to all n_i patches extracted from the $i-$th image labelled with BRAF $= 0$ in the training data set. Let $\mathbf{x}_{ij}$ denote the vector corresponding to the $j-$th patch of the $i-$th image for $j = 1, \ldots, n_i$ and $i = 1, \ldots, N_0$. There are in total $\sum_{i=1}^{N_0} n_i = N$ patches. Finally, we can obtain a $p \times N$ input matrix $X = [\mathbf{x}_{11}, \ldots, \mathbf{x}_{1n_1}, \ldots, \mathbf{x}_{N_0 1}, \ldots, \mathbf{x}_{N_0 n_{N_0}}]$.

Secondly, it constructs a non-negative low dimensional representation H^+ of the input matrix X with a projection matrix Z between X and H^+. We use $Z = (Z_1, Z_2, Z_3, Z_4, Z_5)$ consisting of five projection layers, which will deeply train the dictionary learning model and learn internal factorization compared to the single-layer model. Specifically, it factorizes the input X into five factors as follows:

$$X \approx Z_1 Z_2 Z_3 Z_4 Z_5 H_5^+, \tag{1}$$

where $H_j^+ = Z_{j+1} \cdots Z_m H_5^+$ for $j \in \{1, 2, 3, 4\}$. The Z is estimated by minimizing the objective function given by

$$C_{\text{deep}} = \frac{1}{2} \left\| X - Z_1 Z_2 Z_3 Z_4 Z_5 H_5^+ \right\|_F^2. \tag{2}$$

Estimating Z is by solving the gradient equation $\partial C_{\text{deep}} / \partial z_i = 0$ for $i = \{1, 2, 3, 4, 5\}$. It leads to updating Z_i according to

$$Z_i = \Phi_i^\dagger X \tilde{H}_i^\dagger, \tag{3}$$

where $\dagger$ denotes the Moore-Penrose pseudo-inverse and $\tilde{H}_i$ is the update of H_i during each layer based on the weight matrix learned from the last iteration. Afterwards, the following operation will be applied on H_i to make it to be non-negative:

$$H_i = H_i \odot \sqrt{\frac{[\Phi_i^T X]^{pos} + [\Phi_i^T \Phi_i]^{neg} H_i}{[\Phi_i^T X]^{neg} + [\Phi_i^T \Phi_i]^{pos} H_i}}, \tag{4}$$

where $[\cdot]^{pos}$ and $[\cdot]^{neg}$ represents operations that replace negative or positive elements in the target matrix by 0. In our real data analysis, we set the dimension of Z_i to be $Z_1 \in R^{p \times 400}, Z_2 \in R^{400 \times 300}, Z_3 \in R^{300 \times 200}, Z_4 \in R^{200 \times 100}$, and $Z_5 \in R^{100 \times 50}$, respectively. This dimension setting comes from the experiment and the final low-dimensional representation $H_5^+ \in R^{50 \times N}$ is in an acceptable feature dimension. After convergence, we get the final output $H_5^+ = [h_{11}, \ldots, h_{1n_1}, \ldots, h_{N_0 1}, \ldots, h_{N_0 n_{N_0}}] \in R^{50 \times N}$, where h_{ij} denotes the low-dimensional representation of patch j from image i.

Thirdly, it run the 2-class K-Means clustering to cluster all N patches into two sets $S = (S_1, S_2)$ by using the following objective function:

$$\arg\min_S \sum_{l=1}^{2} \sum_{h \in S_l} \|h - \mu_l\|^2 \tag{5}$$

where μ_l is the center of class l for $l = 1, 2$. The idea of using the 2-class K-Means is that it may be reasonably assumed that most patches obtained from the images labeled with BRAF $= 0$ cannot detect BRAF mutation and share the similar features. Ideally, most of these patches will fall into a larger class, which is confirmed by our experiment, since only $10 - 20\%$ of patches go to the small class. So it uses all the 'normal' patches in the large class to build the BRAF $= 0$ space and then calculate its center, denoted as μ^*.

Fourthly, it projects $X^* \in R^{p \times N^*}$ with all the patches extracted from the training data set including those extracted from the images labelled with BRAF $= 1$ to the 'normal' space by using the same weight matrix Z^*, which gives us the low-dimensional representation of the $j-$th patch for the $i-$th image, denoted as h_{ij}^*, for $j = 1, \cdots, n_i$ and $i = 1, \ldots, N_1$, where N_1 is the total number of patches corresponding to both BRAF $= 1$ and BRAF $= 0$, and $\sum_{i=1}^{N_1} n_i = N^*$. For the $i-$th subject, it then chooses the

patch with the longest distance from the 'normal' center as the most discriminative patch h_i^* by

$$h_i^* = \{h_{ij}^* : \min_{j\in\{1,\ldots,n_i\}} \left\|h_{ij}^* - \mu^*\right\|^2\}. \quad (6)$$

Finally, it obtains $\mathbf{h}^* = (h_1^*, \ldots, h_{N_1}^*) \in R^{50\times N_1}$ as the feature matrix corresponding to all images. Figure 5 presents the path-diagram of these four steps.

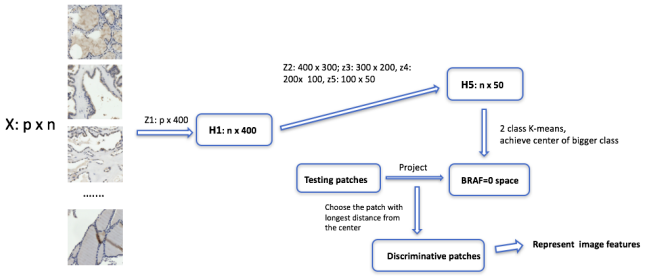


Fig. 5: The Deep Dictionary Learning framework for the patches corresponding to BRAF$=0$.

The reason that we only select the most discriminative patch for each image is that we do not have any prior knowledge of these patches that are related to the BRAF mutation. A possible extreme case is that fewer than 10% of patches from a BRAF $=1$ image have this kind of mutation, whereas all the others are 'normal'. If we use the whole image to extract features or randomly select a patch to represent the image, the prediction results may be biased. Thus, we assume that the most discriminative patch for each BRAF $=1$ image should be far from the center μ^* compared with those from images corresponding to BRAF$=0$. For the testing data set, although we do not have the patch-level BRAF annotation, we project all the patches onto the BRAF $=0$ space and pick out the discriminative patch to represent each image. The complete process is summarized in Algorithm 1.

Algorithm 1 train a deep dictionary model and extract image features. Let $D(\cdots)$ be the deep dictionary learning function and $K(\cdots)$ be the K-Means clustering algorithm, respectively.

Algorithm 1.1
Input: $X \in R^{p\times N}$; set S including all layer dimensions
Output: Z^* and H_i for $i \in \{1,2,3,4,5\}$
Initialization:
for all layers **do**
  $Z_i, H_i \leftarrow D(H_{i-1}, S)$
repeat
  for all layers **do**
    $\tilde{H}_i = Z_{i+1}\tilde{H}_{i+1}$ for $i \in \{1,2,3,4\}$
    $Z_i = \Phi_i^\dagger X \tilde{H}_i^\dagger$
    $H_i = H_i \odot \sqrt{\frac{[\Phi_i^T X]^{pos} + [\Phi_i^T \Phi_i]^{neg} H_i}{[\Phi_i^T X]^{neg} + [\Phi_i^T \Phi_i]^{pos} H_i}}$.
until Convergence

Algorithm 1.2
Input: $X^* \in R^{p\times N^*}$; trained weight matrix Z^*
Output: h^* and μ^*
$(S_1, S_2, \mu_1, \mu_2) = K(H_5)$
$\mu^* = \mu_i : |S_i| > |S_j|$
$h_{ij}^* \leftarrow D(X^*, Z^*)$ for $j = 1, \cdots, n_i$ and $i = 1, \ldots, N_1$
for each i **do**
  $h_i^* = \{h_{ij}^* : \min_{j\in\{1,2,\ldots,n_i\}} \left\|h_{ij}^* - \mu^*\right\|^2\}$.

For the last step, TMA-D^{2}LM uses XGBoost (Chen and Guestrin, 2016) as the classification algorithm. XGBoost is short for 'Extreme Gradient Boosting', in which the Gradient Boosting stands for the algorithm to produce a prediction model in the form of an ensemble of weak prediction models, typically decision trees. These extracted image features along with the seven demographic covariates are used as predictors to predict the five clinical parameters of interest. The prediction framework is built with four layers shown in Figure 6. Specifically, the bottom layer contains the image features extracted from BRAF and H&E TMA, and then they are combined with the demographic covariates to predict BRAF and Extension, respectively. Then, they use the demographic covariates, BRAF and extension to predict size and N, respectively. Finally, size, N, BRAF and extension are combined to predict the cancer stage. Before training the XGBoost prediction model, TMA-D^{2}LM sorts all the features according to their marginal correlation with the outcome and sequentially add predictors until the overall AUC value does not increase within a certain number of steps to prevent over-fitting when learning the XGBoost structure. Then it iteratively drops out the features considered with 'importance $=0$' by XGBoost until convergence and uses the predictive model with remaining covariates to do the final prediction.

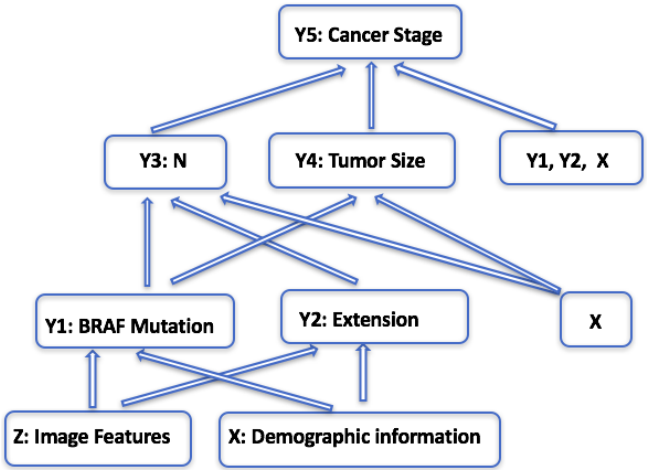


Fig. 6: the prediction structure of TMA-D^{2}LM on how to predict the five clinical parameters

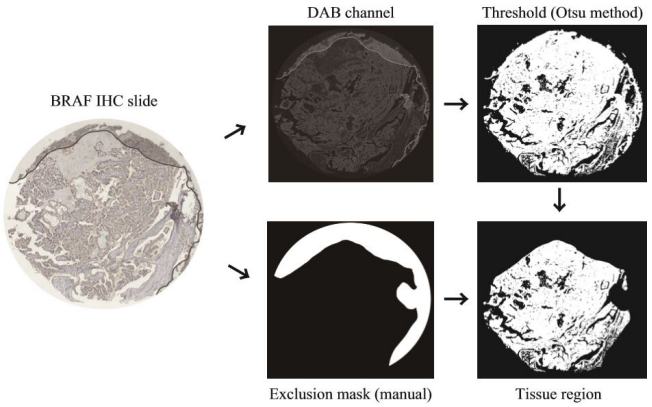


Fig. 7: The workflow for extracting positively stained tissue region from a whole slide.

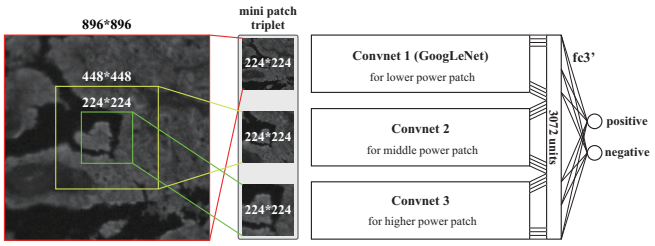


Fig. 8: The multi-resolution convnet architecture.

4.2 Hybrid Prediction Model for Thyroid Cancer Diagnosis

Suzuki *et al*. built a hybrid prediction model for thyroid cancer diagnosis. They separate the prediction model into two sub-modules. The first module is dedicated to predict BRAF mutation status using only IHC slide as input, and the second module predicts all the other clinical diagnoses using the rest of input data, namely H&E slide image, clinical features such as age and sex, as well as the BRAF mutation prediction from the first module. The two modules employ different machine learning approaches for building the individual prediction models reflecting the nature of tasks. Suzuki *et al*. use a deep convolutional network (convnet)-based approach for building the mini patch-level discriminative model, respecting the promising performance of convnets in the recent literature in image recognition and competitions in the field of medical imaging (Szegedy *et al*., 2015; Ronneberger *et al*., 2015; Wang *et al*., 2016). A novel network architecture is built to take a set of overlapping image patches with different magnification levels as the input for capturing the image features of cancer slides in diverse biological scales from individual cells to tissue structures. In addition, they employ additional ad-hoc techniques reflecting pathologists' observations for preparing training datasets and deriving the final decision.

Moreover, Suzuki *et al.* hypothesized that the nuclear features including size, shape and texture of HE stained slides of thyroid cancer could be useful for the prediction, as some studies demonstrated that aneuploidy correlates to aggressiveness in papillary thyroid carcinoma (Sturgis *et al*., 1999) and aneuploidy could affect size and texture of nuclei where abnormal quantities of DNA are contained. In order to build predictive models for other clinical diagnoses, Suzuki *et al.* use image features of nucleus in HE tissue microarray image as well as clinical features and BRAF mutation. Since these features include categorical, discrete and continuous variables, Suzuki *et al.* use gradient boosting trees (Chen and Guestrin, 2016) for the prediction, which are known to be effective and powerful in such situation. Hyperparameter optimization of the prediction model was efficiently performed using Bayesian optimization technique.

4.2.1 BRAF predictor - data preparation First step in this method is convert IHC slide images from RGB to HED (Hematoxylin-Eosin-DAB) color space using `rgb2hed` routine of scikit-image (Van der Walt *et al*., 2014; Ruifrok and Johnston, 2001), followed by extracting only the DAB channel as a grayscale image. Then the method of (Otsu, 1979) is applied to the grayscale image to find the threshold of DAB intensity to extract only the positively stained tissue region from the whole slide (Figure 7 upper).

Since the provided IHC slides for training and testing contained considerable amount of unusable regions, mask images is manually created to exclude such regions for each slide (Figure 7 lower). Suzuki *et al.*'s method excluded regions where the tissue is obviously out of focus, the tissue is folded, or the stain is spilled. Suzuki *et al.* also excluded 22 slides whose appearance and the label obviously seem not to match from the viewpoint of a trained pathologist, which are possibly because of uncommon biological processes or mere labeling errors. For each slides, multiple positions within the tissue regions where the exclusion mask do not cover are randomly selected, and finally a set of mini patches is extracted from the DAB image around each selected position. Furthermore, 400 mini patch sets are prepared for each IHC slide, and use them after applying random rotation in the training session. Suzuki *et al.*'s method only use the patches from tumor slides in both training and validation session.

A mini patch set consists of three mini patches with different magnification levels sharing the center position in a slide. Their original resolutions are 896 × 896 px, 448 × 448 px, and 224 × 224 px in the original IHC slide, respectively, and they are resized to 224 × 224 px. 224 px is about 10 times as large as the diameter of a individual cell in the given IHC slide, therefore this magnification level is suitable for capturing the staining appearance of individual cells. On the other hand, lower-power (448 px and 896 px) mini patches contain several hundreds of cells, so they convey the essential information for identifying the morphology of tissues composed of many V600E-positive cells.

4.2.2 Network architecture and training method of BRAF predictor

Suzuki *et al*. designed a novel network that takes a triplet of mini patches as the input and outputs a binary classification result (Figure 8). The network is simply composed of three parallel convolutional networks, each of them are nearly equivalent to the GoogLeNet (Szegedy *et al*., 2015). Then, the final fully-connected layers of the sub-networks is replaced by a single fully-connected layer (fc3') that integrates the convolution results from all the three sub-networks. Suzuki *et al.* also modify the number of output units to two to support binary classification. Furthermore, the following loss function is empirically chosen. $\mathrm{loss1}_n$ and $\mathrm{loss2}_n$ are equivalent to the loss1 and loss2 functions defined in the original network, respectively, and loss3 is the softmax cross entropy of the output layer (fc3').

$$0.3 \times \sum_{n=1}^{3} (\mathrm{loss1}_n + \mathrm{loss2}_n) + \mathrm{loss3} \tag{7}$$

The training of the network is conducted by an ordinary manner of supervised training. Suzuki *et al*.'s method put "positive" (1) label for all the mini patch sets from a slide whose patient has V600E-positive label, and put "negative" (0) label for the other mini patch sets. For conducting self-testing, provided training slides are divided into two groups, then the network is trained with the patches from the first group of the slides, and validated with the second group. Suzuki *et al*.'s method combine the two groups to train the final network for predicting the testing dataset. Network training is carried out with standard back-propagation and Adam (Kingma and Ba, 2014) optimization algorithm with three Tesla K20 GPUs for about a day. Suzuki *et al.* employed Chainer (Tokui *et al*., 2015) as the framework for neural network implementation and training. BVLC GoogLeNet caffemodel [3] was used as the pretrained model for sub-networks.

[3] `http://dl.caffe.berkeleyvision.org/bvlc_googlenet.caffemodel`

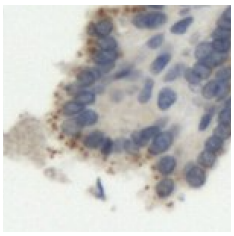


Fig. 10: Dot-like potentially V600E non-specific staining pattern in the given dataset (BRAF03_027_T3).

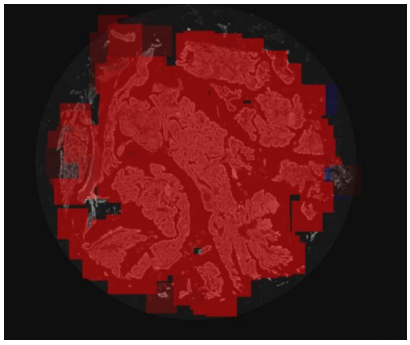


Fig. 9: An example output of minipatch-wise prediction results for a validation slide (BRAF12_124_T1). Red/blue regions were predicted as V600E-positive/negative by the trained model. Opacity means the confidence for each patch.

4.2.3 Patient - wise prediction of BRAF mutation

To decide the BRAF mutation status for each patient, the first step is calculate the probability of mutation for each tumor slide of the patient (Figure 9), then average the probabilities for all the slides from the patient. Mutation probability of a slide is basically calculated as the average of the positive probability (i.e. softmax output of the positive unit of fc3') for all the mini patch sets from the slide.

According to the previous reports (Jones *et al*., 2015), anti-V600E antibodies sometimes generate non-specific staining. They are usually granular, dot-like, or nuclear pattern, which are different from true-positive homogeneous cytoplasmic staining pattern. Indeed, Suzuki *et al.* found some slides with such dot-like stain patterns in the given dataset (Figure 10). Hence, Suzuki *et al.* added a handcrafted additional routine to the prediction model to exclude such false-negative cases. This routine counts dots with high DAB intensity employing histogram-based thresholding and a standard blob detection algorithm of OpenCV, then judges a slide to be dot-like false positive if it has more than 1×10^{-5}dots per px^2. Current implementation calculates the threshold of stain intensity for discriminating dots as the 4-SD value of the rightmost peak of the histogram, approximating the peak as a gaussian distribution. The V600E mutation probability of a slide calculated by the network above is reduced by half if it is judged as dot-like positive.

4.2.4 Other clinical diagnoses - training data preparation

Nuclei segmentation and feature extraction from HE stained slides

of tumor were performed using CellProfiler version 2.2.0 (Carpenter *et al*., 2006). Since these analysis could not apply to whole slide images directly due to memory and computational time problem, Suzuki *et al.* performed the analysis to randomly sampled mini patches and summarized the features for each nucleus into representative features for each case. First, all the whole slide images labeled as tumor were split into non-overlapping 256 × 256 px mini patches. After removing patches containing large background regions, 10 patches for each tumor slide are randomly picked .

Then hematoxylin stain image for each HE stain patch was obtained using the 'UnmixColors' module, and 'IdentifyPrimaryObjects' module with adaptive Otsu thresholds was applied to identify the cell nuclei. Next, 60 element features were calculated using 'Measure Object Intensity' and 'Measure Object Size Shape'. The quantitative features covered the size, shapes including Zernike shape features, pixel intensity distributions. These features were aggregated across the case by calculating median and median absolute deviations (MAD) of the values. Suzuki *et al.* did not use mean and standard deviations, which are less robust to outliers compared to median and MAD, for aggregation because they observed intersecting nuclei was often segmented as a single nucleus as shown in Figure 11. Finally, 120 quantitative nuclear features are obtained for each case.

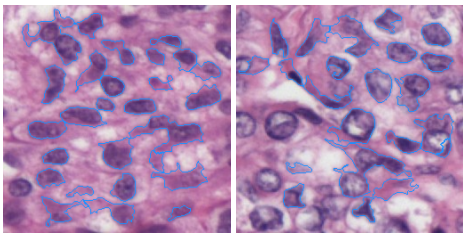


Fig. 11: Segmentation results of the given dataset. Each nucleus is indicated by a closed contour with light blue color.

4.2.5 Other clinical diagnoses - prediction model

Suzuki *et al.* applied extreme gradient boosting trees (Chen and Guestrin, 2016) using xgboost package version 0.6 [4] called from Python for the classification or regression tasks. Linear regression model was used for the prediction of size, while classification models were used for the other discrete ordinal variables such as extrathyroidal extension and lymph node metastasis instead of regression because noise distribution does not follow Gaussian distriution (Van den Oord *et al*., 2016). Multiclass classification using the softmax objective was used for the classification tasks. Estimated BRAF status, sex, age, hashimoto, BMI, BW, BH, the cancer type and 120 nuclei features from tumor HE stained slides described above were used as features.

Suzuki *et al.* optimized hyperparameters of XGBoost using Bayesian optimization (over a validation set different from the final test set). Bayesian Optimization package[5], a python implementation of global optimization with gaussian processes, was used for the purpose. These parameters included the number of trees to train, the maximum depth of each decision tree, and the minimum weight allowed on each decision leaf, the data subsampling ratio, and the minimum gain required to create a new decision branch.

[4] `https://pypi.python.org/pypi/xgboost/0.6`

[5] `https://github.com/fmfn/BayesianOptimization`

4.3 Ensemble Machine Learning Based Approaches for Thyroid Cancer Diagnosis

The ability of immunohistochemistry to quantify a potential biomarker provides the opportunity to study the relationship between the biomarker and chemosensitivity in tumour sub-groups and thereby enables hypothesis generation for additional translational research (Wang, 2013). Using IHC, proteins can be directly visualized by antibodies in their natural cellular localization. In Wang et al's method, an automated quantification method is firstly applied to the IHC images not only for measuring the BRAF expression levels but also for localization of tissues of interests. Next, machine learning models are trained based on the IHC quantification scores and patient's background information. For quantification of BRAF expression and segmentation of tissue of interests, color deconvolution (Ruifrok and Johnston, 2001) is applied to extract independent haematoxylin and DAB/BRAF stain contributions from individual IHC images using orthonormal transformation of RGB. Color deconvolution has been demonstrated to be effective in tissue image analysis in various studies (Wang and Chen, 2013; Wang, 2013; Wang *et al.*, 2014a). In this study, the normalised OD matrix, E, to describe the color system for orthonormal transformation:

$$E = \begin{bmatrix} R & G & B & \\ 0.65 & 0.704 & 0.286 & Haematoxylin \\ 0.268 & 0.570 & 0.776 & DAB/BRAF \\ 0.0 & 0.0 & 0.0 & \end{bmatrix} \tag{8}$$

Given C is the 3 × 1 vector for amounts of the stains at a particular pixel, the vector of OD levels detected at that pixel is equal to $L = CE$. Therefore, multiplication of the OD image with the inverse of OD matrix results in orthogonal representation of the stains forming the image ($C = E^{-1}L$). The color de-convolution matrix is defined as

$$K = E^{-1} = \begin{bmatrix} R & G & B & \\ k_{11} & k_{12} & k_{13} & Haematoxylin \\ k_{21} & k_{22} & k_{23} & DAB/BRAF \\ k_{31} & k_{32} & k_{33} & \end{bmatrix} \tag{9}$$

Given a particular pixel with intensity level (r,g,b) the BRAF OD (I_{BRAF}) is formalized as follows.

$$I_{BRAF} = \exp\left(\frac{(-(r_s + g_s + b_s) - (2^c - 1)) \times \log(2^c - 1)}{(2^c - 1)}\right) \tag{10}$$

$$\begin{aligned} r_s &= r_{log} \times k_{21} \\ g_s &= g_{log} \times k_{22} \\ b_s &= b_{log} \times k_{23} \end{aligned} \tag{11}$$

$$r_{log} = -\left(\frac{(2^c-1)\times\log(\frac{r+1}{2^c-1})}{(2^c-1)\times\log(2^c-1)}\right)$$
$$g_{log} = -\left(\frac{(2^c-1)\times\log(\frac{g+1}{2^c-1})}{(2^c-1)\times\log(2^c-1)}\right) \quad (12)$$
$$b_{log} = -\left(\frac{(2^c-1)\times\log(\frac{b+1}{2^c-1})}{(2^c-1)\times\log(2^c-1)}\right)$$

where c represents the number of bits used to represent each pixel in each channel.

For segmentation of region/tissue of interests (ROI), a clustering process is performed using Otsu's thresholding method (Otsu, 1979), and the background cluster and foreground stain cluster are automatically separated by selecting an optimal local threshold t with the overlap of the background distribution and foreground stain distribution minimized. The histogram distribution of image intensities is regarded as a probability distribution, $p(g) = n_g/n$, where n_g is the number of the pixels having greyscale intensity g and n is the number of pixels, the within-class variance is defined as the weighted sum of the variances.

$$\sigma^2_{within}(t) = n_B(t)\sigma^2_B(t) + n_F(t)\sigma^2_F(t) \quad (13)$$

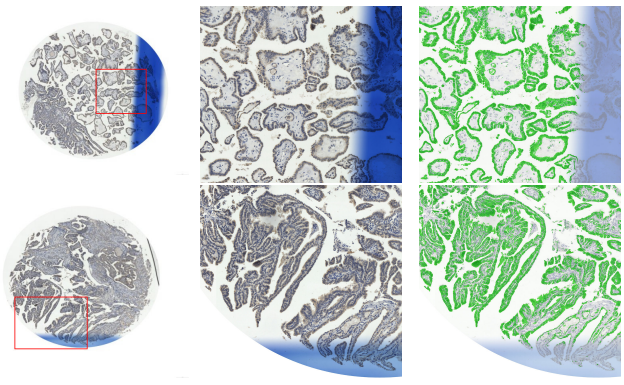
where [0,N-1] is the range of intensity level, $n_B(t) = \sum_{i=0}^{t-1} p(i)$, $n_F(t) = \sum_{N-1}^{i=t} p(i)$, $\sigma^2_B(t)$ is the variance of the pixels in the background cluster (below t) and $\sigma^2_F(t)$ is the variance of the pixels in the foreground cluster (above t).

Then, the between-class variance is the total variance of the combined distribution minus within-class variance.

$$\sigma^2_{between}(t) = \sigma^2 - \sigma^2_{within}(t) = n_B(t)n_F(t)[\mu_B(t) - \mu_F(t)]^2 \quad (14)$$

where $\mu_B(t)$ and $\mu_F(t)$ are the cluster means.

The optimal separation threshold t is the one that maximizes the between-class variance, and a foreground map J for segmented cells is produced.



(a) IHC image (b) Zoom-in View (c) ROI

Fig. 12: Segmentation of tissue of interests (ROI) marked in green.

$$t = \arg\max(\sigma^2_{between}(t)), J_{x,y} = \begin{cases} 0 & ,I^{BRAF}_{x,y} < t \\ 1 & ,I^{BRAF}_{x,y} \geq t \end{cases} \quad (15)$$

A map of ROI Ω can be obtained by applying automated clustering to BRAF image. Figure 12 shows the segmentation results of region of interests for further quantification and diagnosis purposes. Next, five quantification scores of the BRAF expression are computed from the segmented ROI using the five equations below.

$Q_{1_{BRAF}}$ is the quantification result of the mean BRAF expression in tissues sampling from the segmented ROI, which can be defined as

$$Q_{1_{BRAF}} = \frac{\Sigma I_{BRAF}}{\#\Omega} \quad (16)$$

where I_{BRAF} is the intensity level of BRAF image; Ω is the total pixels of ROI.

$$Q_{2_{BRAF}}(\alpha) = \frac{\sum_{(x,y)} I_{BRAF}(x,y)}{\#\Omega}\Big|_{(x,y)\in\Omega, I_{BRAF}(x,y)>\alpha\times(2^c-1)} \quad (17)$$

where c represents the number of bits used to represent each pixel in each channel and $\alpha = 1/3$.

$$Q_{3_{BRAF}}(\alpha) = \frac{\sum_{(x,y)} S_{(x,y)}}{\#\Omega}\Big|_{(x,y)\in\Omega} \quad (18)$$

where

$$S_{(x,y)} = \begin{cases} 5 & ,I_{BRAF} > 80\% \times (2^c-1) \\ 4 & ,I_{BRAF} > 60\% \times (2^c-1) \\ 3 & ,I_{BRAF} > 40\% \times (2^c-1) \\ 2 & ,I_{BRAF} > 20\% \times (2^c-1) \end{cases} \quad (19)$$

$Q_{4_{BRAF}}$ is the mode expression of ROI. This quantification is to find the intensity level of ROI that appears most often.

$$Q_{4_{BRAF}} = \arg\max_i(\#I^i_{BRAF}) \quad (20)$$

$Q_{5_{BRAF}}$ is the total of mean and standard deviation. The standard deviation can be defined as follows.

$$Q_{5_{BRAF}} = \mu(I_{BRAF}) + \sigma(I_{BRAF}) \quad (21)$$

$$\sigma = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(x_i - \mu)^2} \quad (22)$$

where x_i is the intensity level of BRAF image and μ is the mean value.

Next, various machine learning models are utilized to generate the prediction models, which is based on the quantification scores and the background information (see Table 9).

As multiple cores are collected for each patient, for patient-based prediction, a voting model is developed to generate the final prediction result from the core-based prediction outcomes. Given a testing set $S : (x^n, x^t_1, ..., x^t_M)$ with one normal tissue core and M tumor tissue cores, a pre-trained machine learning model U and a set of possible prediction outcomes $O : \{o_1, ..., o_N\}$, a core-based prediction can be generated as follows.

Table 9. Machine Learning Approaches for building Core-based Prediction Models

Parameter	Method	
BRAF	AdaBoostM1 (50) Decision Stump	F-measure = 0.93
Stage	AdaBoostM1 (80) J48	F-Measure = 0.75
Extension	Bagging with Decision Stump	F-Measure = 0.444
N	Random Forest	F-Measure = 0.55
Size	Simple Linear Regression	Correlation = 0.58 using Type

$$o_j = U(x_i^t)|_{i=1..M} \tag{23}$$

The patient based prediction is formulated as the most frequently predicted class using the tumor tissue cores. If there is no tumour samples, then the prediction will be based on the normal tissue core x^n.

$$o^* = \begin{cases} U(x^n) & ,x^t \in \emptyset \\ \arg\max_{o_j}(U(x_i^t)) & ,x \notin \emptyset \end{cases} \tag{24}$$

REFERENCES

Carpenter, AE. *et al.* (2006) CellProfiler: image analysis software for identifying and quantifying cell phenotypes, *Genome Biol.*, **7:10**, R100.

Chen,T., and Guestrin, C. (2016) XGBoost: A Scalable Tree Boosting System, *arXiv preprint*, arXiv:1603.02754.

Irshad, H. *et al.* (2014) Methods for nuclei detection, segmentation, and classification in digital histopathology: a review-current status and future potential, *IEEE Rev. Biomed. Eng.*, **7**, 97-114.

Jones, RT., *et al* (2015) Cross-reactivity of the BRAF VE1 antibody with epitopes in axonemal dyneins leads to staining of cilia, *Modern Pathology*, **28:4**, 596-606.

Khan, AM. *et al.* (2014) A nonlinear mapping approach to stain normalization in digital histopathology images using image-specific color deconvolution, *IEEE Trans. Biomed. Eng.*, **61:6**, 1729-1738.

Kingma,D. and Ba,J. (2014) Adam: A method for stochastic optimization, *arXiv preprint*, arXiv:1412.6980.

Linder, N. *et al.* (2012) Indentification of Tumor Epithelium and Stroma in Tissue Microarrays using Texture Analysis, *Diagnostic pathology*, **7(1)**:22.

Otsu, N. (1979) A threshold selection method from gray level histograms, *IEEE Trans Systems Man Cybern*, **9**, 62-66.

Ruifrok, AC. and Johnston DA. (2001) Quantification of histochemical staining by color deconvolution, *Anal Quant Cytol Histol*, **23(4)**, 291-299.

Ronneberger,O., Fischer,P. and Brox, T. (2015) U-Net: Convolutional Networks for Biomedical Image Segmentation, *Proceedings of MICCAI 2015*, 234-241.

Sturgis, CD. *et al.* (1999) Image analysis of papillary thyroid carcinoma fine-needle aspirates: significant association between aneuploidy and death from disease, *Cancer*, **87:3**, 155-160.

Szegedy,C. *et al.*. (2015) Going Deeper With Convolutions, *The IEEE Conference on Computer Vision and Pattern Recognition*, 1-9.

Tokui,S. *et al.* (2015) Chainer: a Next-Generation Open Source Framework for Deep Learning, *Proceedings of Workshop on Machine Learning Systems (LearningSys) in The Twenty-ninth Annual Conference on Neural Information Processing Systems (NIPS)*.

Trigeorgis, G. *et al.* (2017) A Deep Matrix Factorization Method for Learning Attribute Representations. *IEEE transactions on pattern analysis and machine intelligence*, **39(3)** :417-429.

Van den Oord, A. *et al.* (2016) WaveNet: A Generative Model for Raw Audio, *arXiv preprint* arXiv:1609.03499.

Van der Walt,S. *et al.* (2014) scikit-image: image processing in Python, *PeerJ*, **2**, e453.

Wang, Ching-Wei. (2013) Fast quantification of immunohistochemistry tissue microarrays in lung carcinoma, *Taylor & Francis*, **16(7)**, 707-716.

Wang, Ching-Wei. and Chen, Hsiang-Chou (2013) Improved image alignment method in application to X-ray images and biological images, *Bioinformatics*, doi:10.1093/bioinformatics/btt309.

Wang, Ching-Wei, Ka, Shuk-Man and Chen, Ann (2014a) Robust image registration of biological microscopic images, *Sci. Rep.*, **4(6050)**.

Wang,D. *et al.* (2016) Deep Learning for Identifying Metastatic Breast Cancer, *arXiv preprint*, arXiv:1606.05718.

Xu, J. *et al.* (2016) A Deep Convolutional Neural Network for Segmenting and Classifying Epithelial and Stromal Regions in Histopathological Images, *Neurocomputing*, **191**, 214-223.