Not again! Data Leakage in Digital Pathology

Nicole Bussola*
Fondazione Bruno Kessler & University of Trento
Trento, Italy
bussola@fbk.eu

Alessia Marcolini*
Fondazione Bruno Kessler
Trento, Italy
amarcolini@fbk.eu

Valerio Maggio, Giuseppe Jurman, Cesare Furlanello
Fondazione Bruno Kessler
Trento, Italy
vmaggio|jurman|furlan@fbk.eu
* joint first author

Abstract

Bioinformatics of high throughput omics data (e.g. microarrays and proteomics) has been plagued by uncountable issues with reproducibility at the start of the century. Concerns have motivated international initiatives such as the FDA's led MAQC Consortium, addressing reproducibility of predictive biomarkers by means of appropriate Data Analysis Plans (DAPs). For instance, repreated cross-validation is a standard procedure meant at mitigating the risk that information from held-out validation data may be used during model selection. We prove here that, many years later, Data Leakage can still be a non-negligible overfitting source in deep learning models for digital pathology. In particular, we evaluate the impact of (i) the presence of multiple images for each subject in histology collections; (ii) the systematic adoption of training over collection of subregions (i.e. "tiles" or "patches") extracted for the same subject. We verify that accuracy scores may be inflated up to 41%, even if a well-designed 10×5 iterated cross-validation DAP is applied, unless all images from the same subject are kept together either in the internal training or validation splits. Results are replicated for 4 classification tasks in digital pathology on 3 datasets, for a total of 373 subjects, and 543 total slides (around $27,000$ tiles). Impact of applying transfer learning strategies with models pre-trained on general-purpose or digital pathology datasets is also discussed.

1 Introduction

The community-wide research effort of the MAQC-II project demonstrated that a well-designed Data Analysis Plan (DAP) is mandatory to avoid selection bias flaws in the development of models in $f \gg n$ conditions, where the f features can be highly correlated [1]. Ioannidis and coll. [2] found that almost 90% of papers in a leading journal in genetics were not repeatable due to methodological or clerical errors. High impact on reproducibility has been linked with inaccuracies in managing batch effects [3], or data normalization derived on development and validation data together, as used in proteomics [4]. In general, *data leakage* is a form of selection bias that happens when information from outside the training dataset is used during model development or selection. For instance, one of the preclinical sub-datasets of the MAQC-II study had microarray data from mice triplets for each experimental condition. It was found that lab mice can be expected to be almost identical in their response; in these cases, the repeated cross-validation DAP must keep replicates together either in the training or in internal validation batches [1]. The goal of this study is to provide evidence that similar issues are still lurking, ready to emerge in the everyday practice of machine learning for

Preprint. Under review.

Dataset	Tot. Subjects	Tot. WSIs	WSIs per Subject			Tot. Tiles	Tiles per Subject		
			Min	Max	Med		Min	Max	Med
GTEx [18]	82	252	1	7	4	23,266	16	782	246.5
HF [19]	209	209	1			2,299	11		
BreaKHis [5]	82	82	1			2,013	9	62	21

Table 1: Dataset statistics

digital pathology. BreaKHis [5] - one of the most popular histology dataset for breast cancer - has been used in more than 30 scientific papers to date as a benchmark for classification algorithms, and data analysis strategies targeting binary or multi-class problems, with results spanning a broad range of performances. Notably, in a non negligible number of studies, overfitting effects due to data leakage are affecting the reported outcomes [6, 7, 8, 9, 10, 11].

Typical deep learning pipelines work on subsamples of the Whole Slide Images (WSIs) to operate on smaller memory chunks on GPUs (e.g 512×512 patches extracted from a $83,663 \times 64,804$ WSI). Further, a significant upscale of the amount of training data available for the deep learning pipeline is obtained by adding a data augmentation step, usually by applying random rotations or flipping. Besides the presence of replicates due to the augmentation process, the situation is anyway complex at the origin because often WSI datasets include images of multiple slices or subregions of the same tissue portion. In summary, a population of hundreds of subimages from the same pathology sample may enter in the WSI analysis [12] [13], thus opening the door to data leakage. Protocols for data partitioning (e.g. a repeated cross-validation DAP) are indeed not immunized against replicates and they should take into account the provenance of samples to avoid any bias induced by overfitting slides or patches related to the same subjects [14]. Such bias will inflate the accuracy estimates on the development data, leading to disappointment on novel held-out data.

In this study, we demonstrate the importance of subject-wise split procedures with a group of experiments on digital pathology images. All experiments are based on DAPPER [15], an environment for predictive digital pathology composed by a core deep learning network ("backbone") as feature encoder and alternative (task-related) classification models, e.g. Random Forest or Fully-Connected Networks (see Fig. 1). We test the impact of different train-test splitting strategies considering multiple deep learning backbone architectures, i.e. DenseNet [16] and ResNet models [17], fine-tuned to the digital pathology domain using transfer learning (see Sec. 3 for more details).

2 Dataset

Three publicly available image classification datasets for digital pathology are considered, namely GTEx [18], Heart Failure (HF) [19], and BreakHis [5]. Statistics of the datasets are reported in Table 1.

GTEx: The dataset comprises a total of $7,051$ H&E stained WSIs ($20\times$ native magnification), gathered from a cohort of 449 donors. In this paper, we consider a subset of 252 WSI randomly selected from 82 subjects, further organised according to their corresponding histological types. The selected 11 tissue types (*classes*) are 1) *adrenal gland*, 2) *bladder*, 3) *breast*, 4) *liver*, 5) *lung*, 6) *ovary*, 7) *pancreas*, 8) *prostate*, 9) *testis*, 10) *thyroid*, 11) *uterus*. These types have been chosen as they all share a comparable number of slides in the original dataset [18]. A total of $23,266$ random tiles of size 512×512 have been extracted from the WSIs, each available at different magnification levels (i.e. $20\times, 10\times, 5\times$). With no loss of generality, in this study we use tiles at $5\times$ magnification.

HF: A collection of 209 WSIs of the left ventricular tissue, each corresponding to a single subject. The learning setting is to classify slides of *heart failure* (N=94) from those labelled as *non-heart failure* (N=115). In particular, the first class includes slides categorised as *ischemic cardiomyopathy* (N=51), *idiopathic dilated cardiomyopathy* (N=41), and *undocumented* (N=2). On the other hand, subjects with no heart failure are also grouped in *normal cardiovascular function* (N=41), *non-HF, no other pathology* (N=72), *non-HF, other tissue pathology* (N=2). The WSIs, originally acquired at $20\times$ magnification, have been downsampled at $5\times$ magnification, and 11 non-overlapping images of regions of interest randomly extracted [19]. The entire collection of $2,299$ tiles is publicly available

on the Image Data Resource repository [1].

BreaKHis: An histopathological dataset composed by $7,909$ tiles of malignant or benign breast tumours, collected from a cohort of 82 patients at different magnification factors ($40\times$, $100\times$, $200\times$, $400\times$) [5]. For comparability with state-of-the-art, only tiles at $200\times$ magnification are used in our experiments. The dataset currently contains four histological distinct types of benign breast tumours, and four malignant tumours. In details: *Adenosis* (N=444); *Fibroadenoma* (N=$1,014$); *Tubular Adenoma* (N=453); *Phyllodes Tumor* (N=569); *Ductal Carcinoma* (N=$3,451$), *Lobular Carcinoma* (N=626); *Mucinous Carcinoma* (N=792); *Papillary Carcinoma* (N=560). In this study, we use this dataset for two classification tasks: (a) binary classification of benign and malignant tumour tissues (BreaKHis-2); (b) classification of the 8 distinct tumour subtypes (BreaKHis-8).

3 Method

The experimental environment defined in this paper is organised into three main steps: (A) WSI processing and tiles generation; (B) the feature extraction protocol; (C) the DAP for machine learning models. Fig. 1 represents the three steps in details. This pipeline leverages on the DAPPER environment [15], which has been further extended by (i) integrating specialised train-test splitting protocols, i.e. *Tile-Wise* and *Patient-Wise*; (ii) updating the feature extractor component with new backbone network architectures; (iii) considering two different transfer learning strategies as for feature embeddings.

(A) Tiles generation A data preprocessing pipeline is used to prepare the WSIs. First the tissue region of interest is automatically identified in each WSI (i.e. the box "Detect Tissue Region" in Fig. 1); then at most 100 tiles of size 512×512 pixel are extracted from each slide. Tiles in which the area of extracted tissue accounts for less than the 85% of the whole patch are rejected. This process combines binarization method, also referred to as *Otsu-threshold* [20], with dilation and hole-filling. At the end of this step, the dataset of tiles is generated.

(B) Feature extraction The dataset resulting from the previous step is then used as input to train a backbone deep neural network for feature extraction. Therefore, *training* set and *test* set are then generated, considering 80% and 20% split ratio for the two sets, respectively. In this study, two data partitioning strategies are considered: in the *Tile-Wise* (TW) protocol, tiles are randomly split between training and test datasets, with no consideration of the original WSI. On the other hand, the *Patient-Wise* (PW) protocol takes into consideration the patient from which the tiles are extracted from, and splits tiles in training and test sets, accordingly. Fig. 2 (B) depicts an example of training/test sets as resulted by the two splitting protocols. Both the two protocols are combined with stratification of samples over corresponding classes. Any class imbalance is accounted for by weighting the error on generated predictions.

The goal of the backbone network is to learn a representation of features for tiles (i.e. *feature embedding*) that will be used as input for machine learning models within a DAP. In this study, two different backbone architectures are considered, namely DenseNet-201 [16] and ResNet-152 [17]. Similar to [13] and [15], we start from off-the-shelf versions of these models pre-trained on ImageNet, and then we fine-tune them to the digital pathology domain using transfer learning. We train the whole network for 50 epochs with a learning rate $\eta = 1e-5$. The Adam algorithm [21] is used as optimizer, in combination with the categorical cross-entropy loss. The β_1 and β_2 parameters of the optimizer are respectively set to 0.9 and 0.999, with no weight decay as suggested in the original paper. The fine-tuning is done using the training set exclusively. We use data augmentation, i.e. random rotation and flipping of input tiles, to reduce the risk of overfitting. Furthermore, we investigate the impact of applying transfer learning using backbone models pre-trained on the combination of ImageNet (general purpose) and the GTEx (domain specific) datasets.

(C) Data Analysis Plan (DAP) The last step of the pipeline comprises the application of the DAP for machine learning models. We adopted a 10×5-fold CV schema [1]. The input datasets are the two separate training and test sets, as resulted from the $80-20$ train-test splitting protocol (see *Feature Extraction*). The test set is kept completely unseen to the model, and only used for the final model

[1] `idr.openmicroscopy.org/`

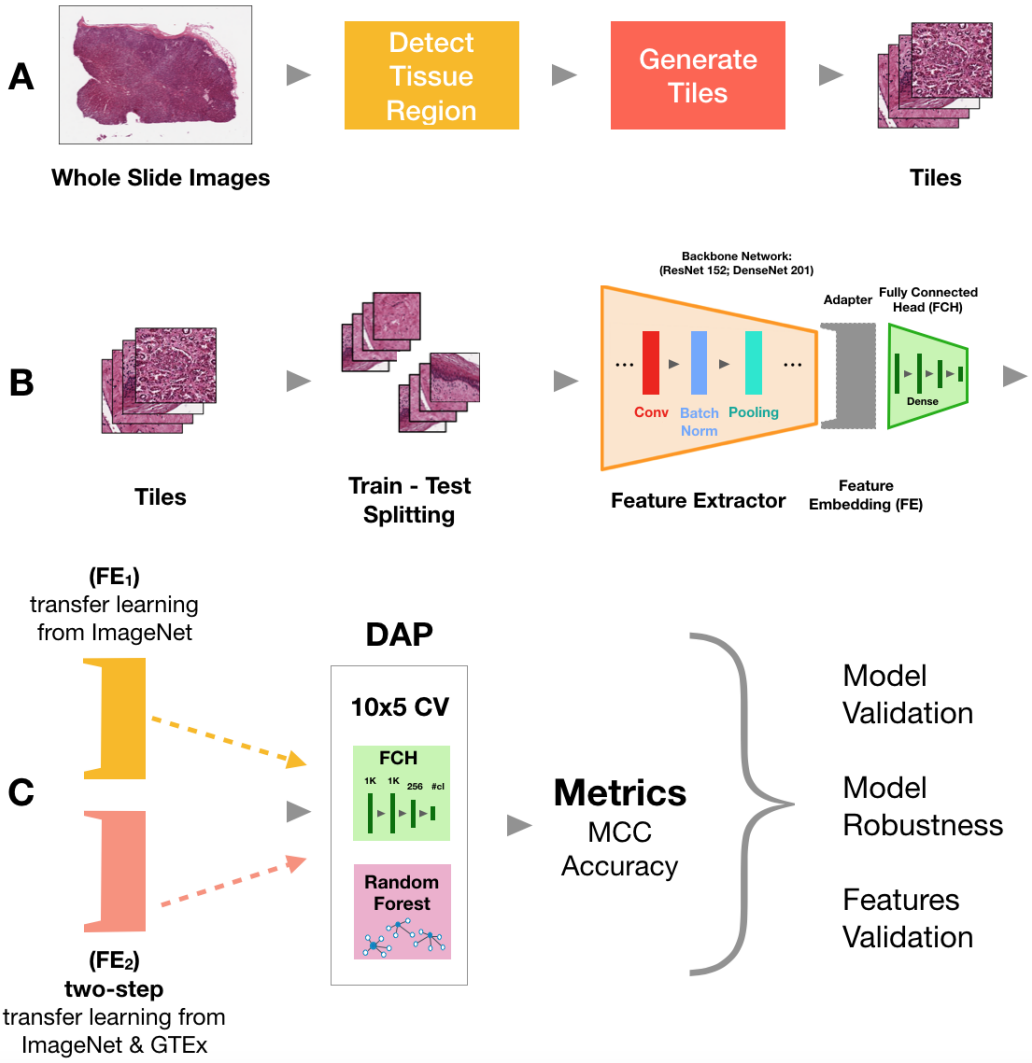

Figure 1: Experimental Environment: (A) WSI preprocessing and tiles generation. (B) Feature extraction protocol. (C) The Data Analysis Plan for machine learning models.

evaluation. The training set further undergoes a 5-fold CV iterated 10 times, resulting in 50 separated internal *validation* sets. These validation sets are generated adopting the same protocols used in the previous train-test splitting, i.e. *Tile-Wise* or *Patient-Wise*. The overall performance of the model is evaluated across all the iterations, in terms of average Matthews Correlation Coefficient (MCC) and Accuracy (ACC) with 95% Studentized bootstrap confidence intervals (CI); and then on the test set. As for multi-class problems, we consider the extension of the MCC metric as defined in [22].

We compared the performance of two machine learning models, i.e. Random Forest (RF) and Fully-Connected Head (FCH), considering two sets of input features: (FE_1) Feature Embedding generated by a backbone model fine-tuned from ImageNet (**transfer learning**); (FE_2) Feature Embedding generated by a backbone model fine-tuned from ImageNet and GTEx (**two-step transfer learning**).

As an additional strategy to corroborate the validity of predictions, the DAP adopts a *random labels* schema (RLab). In this setting, a number of artificially generated labels are provided as reference ground truth for machine learning models. If the adopted data partitioning protocol is immune by any source of data leakage, no signal should be learnt by models, resulting in an average MCC score near zero ($MCC \approx 0$). To emphasise evidence of data leakage derived by the two splitting protocols, the RLab validation is applied: the labels for all the tiles of a single patient are changed consistently, thus

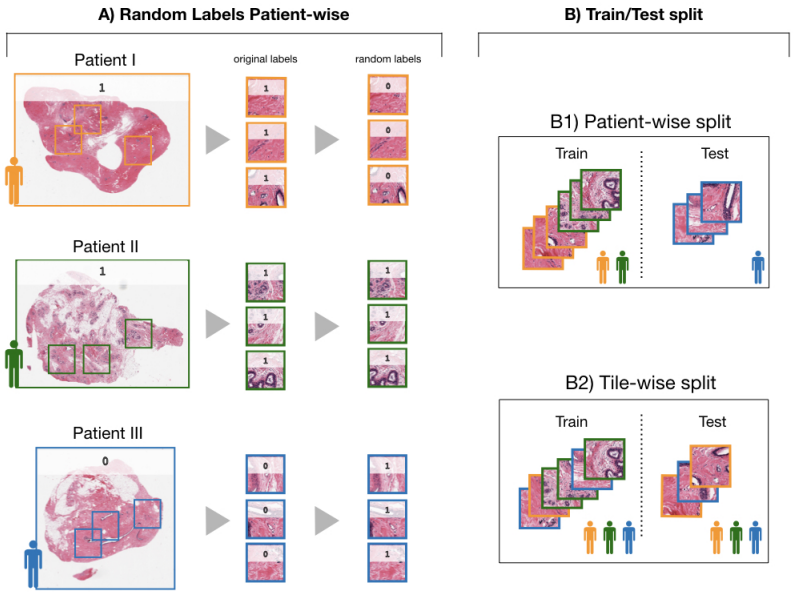

Figure 2: Random Labels experimental settings. A) tiles labels are randomly shuffled patient-wise. B) The train/test split is then performed either *Patient-Wise* or *Tile-Wise*.

all these tiles share the same random labels (Fig. 2-A); the *Patient-Wise* (Fig. 2-B1) and the *Tile-Wise* (Fig. 2-B2) protocols are alternatively used within the DAP.

4 Results and Discussion

Results obtained using a ResNet-152 backbone model pre-trained on ImageNet (i.e. FE_1) are reported in Table 2 and Table 3, considering the *Tile-Wise* and the *Patient-Wise* partitioning protocols, respectively. The average cross validation $\mathrm{MCC_v}$ and $\mathrm{ACC_v}$ with 95% CI are reported, along with results on the test set (i.e. $\mathrm{MCC_t}$, and $\mathrm{ACC_t}$). Corresponding state of the art results (i.e. *Others*) are also reported for comparison, whenever available. in this regard, it is worth mentioning that experimental protocols used in related work are all different, and they are not consistent even in the case of the same dataset (i.e. BreakHis). Therefore, we report here for completeness a short description of these settings for state of the art papers we consider in the comparisons.

Nirschl et al. [19] apply a patient-wise 50-50 train-test split on the HR dataset. Data augmentation is applied in training. Augmentation operations include random cropping, rotation, mirroring, stain color augmentation. Experiments are repeated 3 times.

As for the BreaKHis dataset, Jiang et al. [9] adopt a *Tile-Wise* partitioning protocol. In particular, 60-40 train-test split is used for BreaKHis-2, whilst 70-30 split is used for BreaKHis-8. Experiments are repeated 3 times, with data augmentation in the training process (i.e. flipping, shifting). Authors in [7], and [8] adopt a similar experimental protocol, but no data augmentation is used, and the average of 5 trials are reported. Finally, Motlagh et al. [10] use a *Tile-Wise* 90-10 train-test split with data augmentation (i.e. resizing, rotations, cropping and flipping); whereas Alom et al. [23] use a 70-30 *Patient-Wise* partitioning protocol, and data augmentation (i.e. rotation, shifting, flipping).

As expected, estimates are more favourable for the *TW* protocol with respect to the *PW* one, consistently for all the datasets (both in validation and in test). Moreover, the inflation of the *Tile-Wise* estimates is amplified in the multi-class setting (e.g. see BreaKHis-2 vs BreaKHis-8). Notably, these results are comparable with those in the literature, suggesting the evidence of a data leakage for those adopting the *Tile-Wise* splitting strategy. Results on GTEx do not suggest significant differences using the two protocols; however they are in a very high range for both MCC and ACC metrics.

Analogous results (not reported here) were obtained using the DenseNet-201 backbone model, further confirming the generality of the derived conclusions. However, this model has almost the double

number of parameters[2], and so a higher demand of computation. Therefore diagnostic experiments and transfer learning were performed only using the ResNet-152.

Table 2: DAP results for each classifier head, using the *Tile-Wise* partitioning protocol, and the FE_1 feature embedding using the ResNet-152 backbone model. The average cross validation $\mathrm{MCC_v}$ and $\mathrm{ACC_v}$ with 95% CI are reported, along with results on the test set (i.e. $\mathrm{MCC_t}$, and $\mathrm{ACC_t}$)

Dataset	**FCH**		**RF**		**FCH**		**RF**		**Others**
	$\mathrm{MCC_v}$	$\mathrm{MCC_t}$	$\mathrm{MCC_v}$	$\mathrm{MCC_t}$	$\mathrm{ACC_v}$	$\mathrm{ACC_t}$	$\mathrm{ACC_v}$	$\mathrm{ACC_t}$	$\mathrm{ACC_t}$
GTEx	0.999 (0.999, 0.999)	0.998	0.999 (0.999, 0.999)	0.997	0.999 (0.999, 0.999)	0.999	0.999 (0.999, 0.999)	0.998	-
HF	0.959 (0.956, 0.963)	0.956	0.956 (0.953, 0.959)	0.960	0.980 (0.978, 0.982)	0.978	0.978 (0.977, 0.980)	0.980	-
BreaKHis-2	0.989 (0.987, 0.991)	0.988	0.990 (0.988, 0.992)	0.994	0.995 (0.994, 0.996)	0.994	0.996 (0.995, 0.997)	0.997	0.993 [9]
BreaKHis-8	0.945 (0.942, 0.949)	0.922	0.929 (0.925, 0.932)	0.921	0.959 (0.956, 0.962)	0.940	0.946 (0.943, 0.949)	0.940	0.985 [10]

Table 3: DAP results for each classifier head, using the *Patient-Wise* partitioning protocol, and the FE_1 feature embedding using the ResNet-152 backbone model. The average cross validation $\mathrm{MCC_v}$ and $\mathrm{ACC_v}$ with 95% CI are reported, along with results on the test set (i.e. $\mathrm{MCC_t}$, and $\mathrm{ACC_t}$)

Dataset	**FCH**		**RF**		**FCH**		**RF**		**Others**
	$\mathrm{MCC_v}$	$\mathrm{MCC_t}$	$\mathrm{MCC_v}$	$\mathrm{MCC_t}$	$\mathrm{ACC_v}$	$\mathrm{ACC_t}$	$\mathrm{ACC_v}$	$\mathrm{ACC_t}$	$\mathrm{ACC_t}$
GTEx	0.998 (0.998, 0.998)	0.998	0.997 (0.997, 0.997)	0.997	0.998 (0.998, 0.998)	0.998	0.997 (0.997, 0.998)	0.997	-
HF	0.852 (0.847, 0.858)	0.856	0.848 (0.836, 0.860)	0.833	0.927 (0.924, 0.929)	0.915	0.924 (0.918, 0.930)	0.915	0.932 [19]
BreaKHis-2	0.695 (0.665, 0.724)	0.801	0.709 (0.671, 0.746)	0.863	0.870 (0.856, 0.882)	0.924	0.876 (0.859, 0.892)	0.946	0.973 [23]
BreaKHis-8	0.561 (0.529, 0.594)	0.541	0.594 (0.562, 0.631)	0.471	0.679 (0.655, 0.703)	0.644	0.701 (0.681, 0.732)	0.600	0.973 [23]

4.1 Random Labels

Table 4: Random Labels (RLab) results using ResNet-152 backbone model, and *Tile-Wise* (TW) and *Patient-Wise* (PW) train-test split protocols. The average $\mathrm{MCC_{RL}}$ and $\mathrm{ACC_{RL}}$ with 95% CI are reported.

Dataset	$\mathrm{MCC_{RL}}$		$\mathrm{ACC_{RL}}$	
	TW	*PW*	*TW*	*PW*
HF	0.107 (0.078, 0.143)	0.004 (-0.042, 0.048)	0.553 (0.534, 0.570)	0.502 (0.474, 0.530)
BreaKHis-2	0.354 (0.319, 0.392)	-0.065 (-0.131, 0.001)	0.637 (0.613, 0.662)	0.560 (0.506, 0.626)
BreaKHis-8	0.234 (0.173, 0.341)	0.013 (-0.042, 0.065)	0.318 (0.215, 0.506)	0.097 (0.056, 0.143)

A *caveat emptor* concern comes from the experiments with the RLab validation schema, in which results are consistently over the expected $MCC \approx 0$ using the *Tile-Wise* partitioning. For instance, as for BreaKHis-2 and FCH, $\mathrm{MCC_{RL}}= 0.354\ (0.319, 0.392)$ in the *Tile-Wise* setting, to be compared with $\mathrm{MCC_{RL}}= -0.065\ (-0.131, 0.001)$ using the *Patient-Wise* protocol. Full $\mathrm{MCC_{RL}}$ results considering 5 trials of the RLab test are reported in Table 4. Corresponding $\mathrm{ACC_{RL}}$ values are also included for completeness. Fig.3 shows the boxplot of the distributions of $\mathrm{MCC_{RL}}$ scores for all the datasets, and the two compared partitioning protocols. All the tests using the *Patient-Wise* split perform as expected, with median values near 0, whereas results of the *Tile-Wise* case exhibit a high variability. The worst case is for the BreaKHis-2 dataset in which evidence of some signal learnt by the model are reported, and so consequently of a data leakage.

4.2 Transfer Learning

We then investigate the impact of using the **two-step transfer learning** setting, in combination with the *Patient-Wise* leakage-free partitioning protocol. In particular, two feature extractor backbone models are considered for the task: ResNet-152 pre-trained on ImageNet (FE_1), and ResNet-152 pre-trained on ImageNet and then GTEx (FE_2). Experimental results for this two-step setting is reported in Table 5, to be compared with Table 3.

[2] *DenseNet-201*: $\approxeq$ 12 M parameters; *ResNet-152*: $\approxeq$ 6 M parameters.

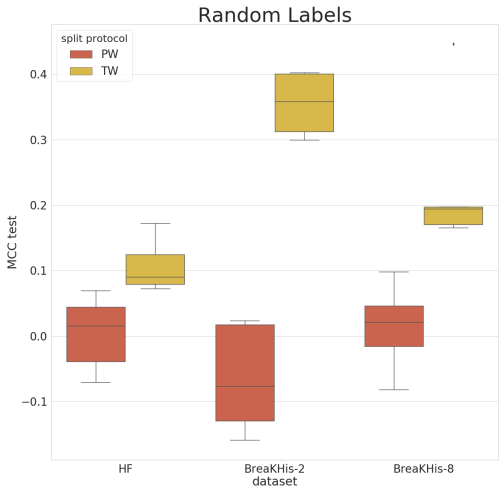

Figure 3: Boxplots of the $\mathrm{MCC}_{\mathrm{RL}}$ results using the *Tile-Wise* and the *Patient-Wise* protocols.

Table 5: DAP results for each classifier head, using the *Patient-Wise* partitioning protocol, and the FE_2 feature embedding using the ResNet-152 backbone model. The average cross validation $\mathrm{MCC_v}$ and $\mathrm{ACC_v}$ with 95% CI are reported, along with results on the test set (i.e. $\mathrm{MCC_t}$, and $\mathrm{ACC_t}$)

Dataset	**FCH**		**RF**		**FCH**		**RF**		**Others**
	$\mathrm{MCC_v}$	$\mathrm{MCC_t}$	$\mathrm{MCC_v}$	$\mathrm{MCC_t}$	$\mathrm{ACC_v}$	$\mathrm{ACC_t}$	$\mathrm{ACC_v}$	$\mathrm{ACC_t}$	$\mathrm{ACC_t}$
HF	0.956 (0.952, 0.960)	0.964	0.955 (0.943, 0.958)	0.950	0.978 (0.976, 0.980)	0.982	0.977 (0.975, 0.979)	0.978	0.932 [19]
BreaKHis-2	0.864 (0.839, 0.888)	0.948	0.912 (0.892, 0.932)	0.961	0.941 (0.930, 0.952)	0.980	0.963 (0.955, 0.971)	0.984	0.973 [23]
BreaKHis-8	0.573 (0.539, 0.602)	0.478	0.586 (0.552, 0.621)	0.482	0.685 (0.661, 0.712)	0.603	0.699 (0.675, 0.724)	0.606	0.973 [23]

The adoption of a domain-specific dataset (i.e. GTEx) in transfer learning is beneficial over the use of ImageNet only. In fact, predictive performance of machine learning models with a *Patient-Wise* partitioning protocol and the FE_2 embedding are higher, and comparable to those obtained using FE_1, but with the inflated TW splitting (see also Table 2 for comparisons). However, very slight improvements are achieved on the BreaKHis-8 task, and results are still below the current state of the art. Notably, the BreaKHis dataset is highly imbalanced in the multi-class task. As a countermeasure, Han and coll. [24] adopted a balancing strategy in the data augmentation pre-processing, that we did not introduce here for comparability with the other experiments.

Finally, to verify how much of previous domain-knowledge can be still re-used for the original digital pathology task (i.e. GTEx classification) we devised the following experiment: we retained the *Feature Extractor* component (i.e. Convolutional Layers) of the model pre-trained on GTEx and fine-tuned on BreakHis-2, and re-attached the Fully-Connected Head of the model trained on GTEx. Notably, we could reach back full predictive performance (i.e. $\mathrm{MCC_t}$=0.983) on the GTEx task after only one single epoch of full training using the GTEx samples.

4.3 Patient-level Performance Analysis

In order to assess the ability of machine learning model to generalise on unseen patients, patient-wise performance metrics have been defined in the literature [5, 23, 19]. Two metrics will be considered in this study: (1) **Winner-takes-all** (WA), and (2) **Patient Score** (PS).

Let $\hat{Y}_p = \{\hat{y}_t\}$ be the set of labels predicted by a machine learning model for all the tiles t of a single patient $p \in P$; $N_p = |\hat{Y}_p|$, whilst N_{rec} is the total number of tiles in $\hat{Y}_p$ correctly classified.

In the Winner-takes-all metric, the label associated to each patient corresponds to the majority of the labels predicted for their tiles. More formally, for each patient p:

$$\hat{y}_p = \hat{y}_t : \{\hat{y}_t\}^n \in \hat{Y}_p, n = 2k+1 \geq j, \forall \{\bar{y}_t\}^j \in \hat{Y}_p$$

Using this strategy, the overall performance indicator can be either standard ACC or MCC using $\hat{y}_p$ as reference predictions for all the patients. In this study we used ACC, for comparability with the PS metric, and results reported in the literature.

On the other hand, the PS metric is defined as an accuracy restricted to tiles in $\hat{Y}_p$ [5], for each patient. The overall performance is then calculated using the *global recognition rate* (RR), defined as the average of all the PS scores for all the patients. Therefore:

$$PS = \frac{N_{\text{rec}}}{N_p}; \;\; RR = \frac{\sum PS}{|P|}$$

In this paper, the WA metric is used to compare our patient-level results with those reported in [19], for the HR dataset. The PS metric is used for comparison on the BreaKHis dataset.

Patient-level results for both *Tile-Wise* and *Patient-Wise* partitioning protocols are reported in Table 6, using the ResNet-152 backbone model, and the FE_1 feature embeddings. Results of patient-level metrics for FE_2 and *Patient-Wise* protocol are reported in Table 7.

Table 6: Patient-level results for each classifier head, using the *Patient-Wise* and *Tile-Wise* partitioning protocols, and the FE_1 feature embedding with the ResNet-152 backbone model. The average cross-validation Patient-level accuracy with 95% CI (ACC_v), and corresponding scores on the test set (ACC_t), are reported.

Dataset	Patient-level Metric	Partitioning Protocol	FCH		RF		Others
			ACC_v	ACC_t	ACC_v	ACC_t	ACC_t
HF	WA	TW	0.984 (0.982, 0.987)	0.995	0.984 (0.981, 0.986)	0.995	-
		PW	0.981 (0.975, 0.986)	0.951	0.977 (0.971, 0.983)	0.927	0.940 [19]
BreaKHis-2	PS	TW	0.995 (0.994, 0.996)	0.997	0.997 (0.996, 0.998)	0.998	0.872 [8]
		PW	0.864 (0.851, 0.877)	0.885	0.883 (0.869, 0.898)	0.893	0.976 [23]
BreaKHis-8	PS	TW	0.963 (0.960, 0.967)	0.950	0.957 (0.955, 0.959)	0.962	0.964 [7]
		PW	0.687 (0.667, 0.709)	0.752	0.705 (0.685, 0.728)	0.725	0.967 [23]

Table 7: Patient-level results for each classifier head, using the *Patient-Wise* partitioning protocol, and the FE_2 feature embedding with the ResNet-152 model. The average cross-validation Patient-level accuracy with 95% CI (ACC_v), and corresponding scores on the test set (ACC_t), are reported.

Dataset	Patient-level Metric	FCH		RF		Others
		ACC_v	ACC_t	ACC_v	ACC_t	ACC_v
HF	WA	0.992 (0.989, 0.995)	0.976	0.989 (0.984, 0.992)	0.976	0.940 [19]
BreaKHis-2	PS	0.941 (0.930, 0.951)	0.971	0.958 (0.948, 0.968)	0.991	0.976 [23]
BreaKHis-8	PS	0.691 (0.669, 0.716)	0.721	0.699 (0.676, 0.723)	0.724	0.967 [23]

References

[1] The MAQC Consortium. The MAQC-II Project: A comprehensive study of common practices for the development and validation of microarray-based predictive models. *Nature Biotechnologyogy*, 28(8):827–838, 2010.

[2] J. P. A. Ioannidis, D. B. Allison, C. A. Ball, I. Coulibaly, X. Cui, A. C. Culhane, M. Falchi, C. Furlanello, L. Game, G. Jurman, J. Mangion, T. Mehta, M. Nitzberg, G. P. Page, E. Petretto, and V. van Noort. Repeatability of published microarray gene expression analyses. *Nature Genetics*, 41(2):149, 2009.

[3] J T Leek, R B Scharpf, H Bravo, D Simcha, B L, W E J, D Geman, K Baggerly, and R A Irizarry. Tackling the widespread and critical impact of batch effects in high-throughput data. *Nature Reviews Genetics*, 11(10):733, 2010.

[4] A. Barla, G. Jurman, S. Riccadonna, S. Merler, M. Chierici, and C. Furlanello. Machine learning methods for predictive proteomics. *Briefings in Bioinformatics*, 9(2):119–128, 2008.

[5] F. A. Spanhol, L. S. Oliveira, C. Petitjean, and L. Heutte. A Dataset for Breast Cancer Histopathological Image Classification. *IEEE Transaction in Biomedical Engineering*, 63(7):1455–1462, 2016.

[6] L. Li, X. Pan, H. Yang, Z. Liu, Y. He, Z. Li, Y. Fan, Z. Cao, and L. Zhang. Multi-task deep learning for fine-grained classification and grading in breast cancer histopathological images. *Multimedia Tools and Applications*, Epub ahead of print:7 Dec, 2018.

[7] Majid Nawaz, Adel A Sewissy, and Taysir Hassan A Soliman. Multi-class breast cancer classification using deep learning convolutional neural network. *Int. J. Adv. Comput. Sci. Appl*, 9(6):316–332, 2018.

[8] Juanying Xie, Ran Liu, IV Luttrell, Chaoyang Zhang, et al. Deep Learning Based Analysis of Histopathological Images of Breast Cancer. *Frontiers in Genetics*, 10:80, 2019.

[9] Yun Jiang, Li Chen, Hai Zhang, and Xiao Xiao. Breast cancer histopathological image classification using convolutional neural networks with small SE-ResNet module. *PLOS ONE*, 14(3):e0214587, 2019.

[10] Nima Habibzadeh Motlagh, Mahboobeh Jannesary, HamidReza Aboulkheyr, Pegah Khosravi, Olivier Elemento, Mehdi Totonchi, and Iman Hajirasouliha. Breast cancer histopathological image classification: A deep learning approach. *bioRxiv*, page 242818, 2018.

[11] Rajesh Mehra et al. Breast cancer histology images classification: Training from scratch or transfer learning? *ICT Express*, 4(4):247–254, 2018.

[12] D. Komura and S. Ishikawa. Machine Learning Methods for Histopathological Image Analysis. *Computational and Structural Biotechnology Journal*, 16:34–42, 2018.

[13] R. Mormont, P. Geurts, and R. Marée. Comparison of Deep Transfer Learning Strategies for Digital Pathology. In *Proceedings of the 2018 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 2343–234309. IEEE, 2018.

[14] R. Marée. The Need for Careful Data Collection for Pattern Recognition in Digital Pathology. *Journal of Pathology Informatics*, 8(1):19, 2017.

[15] A. Bizzego, N. Bussola, M. Chierici, V. Maggio, M. Francescatto, L. Cima, M. Cristoforetti, G. Jurman, and C. Furlanello. Evaluating reproducibility of AI algorithms in digital pathology with DAPPER. *PLOS Comp Biol*, 15(3):1–24, 2019.

[16] G. Huang, L. Zhuang, L. van der Maaten, and K. Q. Weinberger. Densely Connected Convolutional Networks. In *Proceedings of the 2018 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2261–2269. IEEE, 2018.

[17] K. He, X. Zhang, S. Ren, and J. Sun. Deep Residual Learning for Image Recognition. In *Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778. IEEE, 2016.

[18] The GTEx Consortium. The genotype-tissue expression (GTEx) project. *Nature Genetics*, 45(6):580–585, 2013.

[19] J. J. Nirschl, A. Janowczyk, E. G. Peyster, R. Frank, K. B. Margulies, M. D. Feldman, and A. Madabhushi. A deep-learning classifier identifies patients with clinical heart failure using whole-slide images of H&E tissue. *PLOS ONE*, 13(4):e0192726, 2018.

[20] N. Otsu. A threshold selection method from gray-level histograms. *IEEE Transactions on Systems, Man, and Cybernetics*, 9(1):62–66, 1979.

[21] D. P. Kingma and J. Ba. Adam: A Method for Stochastic Optimization, 2014.

[22] Giuseppe Jurman, Samantha Riccadonna, and Cesare Furlanello. A comparison of mcc and cen error measures in multi-class prediction. *PLOS ONE*, 7(8):1–8, 08 2012.

[23] M. Z. Alom, C. Yakopcic, S. Nasrin, T. M. Taha, and V. K. Asari. Breast Cancer Classification from Histopathological Images with Inception Recurrent Residual Convolutional Neural Network. *Journal of Digital Imaging*, 32(4):605–617, 2019.

[24] Z. Han, B. Wei, Y. Zheng, Y. Yin, K. Li, and S. Li. Breast Cancer Multi-classification from Histopathological Images with Structured Deep Learning Model. *Scientific Reports*, 7(1):4172, 2017.