AI slipping on tiles: data leakage in digital pathology

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

Alessia Marcolini[*]
HK3 Lab, Milano, Italy
alessia.marcolini@hk3lab.ai

Valerio Maggio
University of Bristol, Bristol, United Kingdom
valerio.maggio@bristol.ac.uk

Giuseppe Jurman[†]
Fondazione Bruno Kessler, Trento, Italy
jurman@fbk.eu

Cesare Furlanello[†]
HK3 Lab, Milano, Italy
cesare.furlanello@hk3lab.ai
[*] joint first author, [†] joint last author, [◇] Corresponding author

Abstract

Reproducibility of AI models on biomedical data still stays as a major concern for their acceptance into the clinical practice. Initiatives for reproducibility in the development of predictive biomarkers as the MAQC Consortium already underlined the importance of appropriate Data Analysis Plans (DAPs) to control for different types of bias, including data leakage from the training to the test set. In the context of digital pathology, the leakage typically lurks in weakly designed experiments not accounting for the subjects in their data partitioning schemes. This issue is then exacerbated when fractions or subregions of slides (i.e. "tiles") are considered. Despite this aspect is largely recognized by the community, we argue that it is often overlooked. In this study, we assess the impact of data leakage on the performance of machine learning models trained and validated on multiple histology data collection. We prove that, even with a properly designed DAP (10×5 repeated cross-validation), predictive scores can be inflated up to 41% when tiles from the same subject are used both in training and validation sets by deep learning models. We replicate the experiments for 4 classification tasks on 3 histopathological datasets, for a total of 374 subjects, 556 slides and more than $27,000$ tiles. Also, we discuss the effects of data leakage on transfer learning strategies with models pre-trained on general-purpose datasets or off-task digital pathology collections. Finally, we propose a solution that automates the creation of leakage-free deep learning pipelines for digital pathology based on `histolab`, a novel Python package for histology data preprocessing. We validate the solution on two public datasets (TCGA and GTEx).

keywords: reproducibility, deep learning, digital pathology.

1 Introduction

Bioinformatics on high-throughput omics data has been plagued by uncountable issues with reproducibility since its early days; Ioannidis and colleagues [1] found that almost 90% of papers in a

To appear in the ICPR 2020 Workshop Proceedings Springer volume

leading journal in genetics were not repeatable due to methodological or clerical errors. Although the landscape seems to have improved [2], and broad efforts have been spent across different biomedical fields [3], computational reproducibility and replicability still fall short of the ideal. Lack of reproducibility has been linked to inaccuracies in managing batch effects [4, 5], small sample sizes [6], or flaws in the experimental design such as data normalization simultaneously performed on development and validation data [7, 8]. The MAQC-II project for reproducible biomarker development from microarray data demonstrated, through a community-wide research effort, that a well-designed Data Analysis Plan (DAP) is mandatory to avoid selection bias flaws in the development of models for high-dimensional datasets [9].

Among the various types of selection bias that threaten the reproducibility of machine learning algorithms, *data leakage* is possibly the most subtle one [10]. Data leakage refers to the use of information from outside the training dataset during model training or selection [11]. A typical leakage occurs when data in the training, validation and/or test sets share indirect information, leading to overly optimistic results. For example, one of the preclinical sub-dataset in the MAQC-II study consisted of microarray data from mice triplets. These triplets were expected to have an almost identical response for each experimental condition, and therefore they had to be kept together in DAP partitioning to circumvent any possible leakage from training to internal validation data [9].

The goal of this study is to provide evidence that similar issues are still lurking in the grey areas of preprocessing, ready to emerge in the everyday practice of machine learning for digital pathology. The BreaKHis [12] dataset, one of the most popular histology collection of breast cancer samples, has been used in more than 40 scientific papers to date [13], with reported results spanning a broad range of performance. In a non-negligible number of these studies, overfitting effects due to data leakage are suspected to impact their outcomes.

Deep learning pipelines for histopathological data typically require Whole Slide Images (WSIs) to be partitioned into multiple patches (also referred to as "tiles" [14]) to augment the original training data, and to comply with memory constraints imposed by GPU hardware architectures. For example, a single WSI of size $67,727 \times 47,543$ pixels can be partitioned in multiple 512×512 tiles, which are randomly extracted, and verified such that selected subregions preserve enough tissue information. These tiles are then processed by data augmentation operators (e.g. random rotation, flipping, or affine transformation) to reduce the risk of overfitting. As a result, the number of multiple subimages originating from the very same histological specimen is significantly amplified [15, 16], consequently increasing the the risk for data leakage. Protocols for data partitioning (e.g. a repeated cross-validation DAP) are not naturally immune against replicates, and so the source originating each tile should be considered to avoid any risk of bias [17].

In this work, we quantify the importance of adopting *Patient-Wise* split procedures with a set of experiments on digital pathology datasets. All experiments are based on DAPPER [18], a reproducible framework for predictive digital pathology composed of a deep learning core ("backbone network") as feature encoder, and multiple task-related classification models, i.e. Random Forest or Multi-Layer Perceptron Network (see Fig. 1). We test the impact of various data partitioning strategies on the training of multiple backbone architectures, i.e. DenseNet [19], and ResNet models [20], fine-tuned to the histology domain.

Our experiments confirm that train-test contamination (in terms of modeling) is a serious concern that hinders the development of a dataset-agnostic methodology, with impact similar to the lack of standard protocols in the acquisition and storage of WSIs in digital pathology [21]. Thus, we present a protocol to prevent data leakage during data preprocessing. The solution is based on `histolab`, an open-source Python library designed as a reproducible and robust environment for WSI preprocessing, available at https://github.com/histolab/histolab. The novel approach is demonstrated on two public large scale datasets: GTEx [22] (i.e. non-pathological tissues), and TCGA [23] (i.e. cancer tissues).

2 Data description

We tested our experimental pipeline on three public datasets for image classification in digital pathology, namely GTEx [22], Heart Failure (HF) [24], and BreaKHis [12]. Descriptive statistics of the datasets are reported in Table 1, and Fig.1.

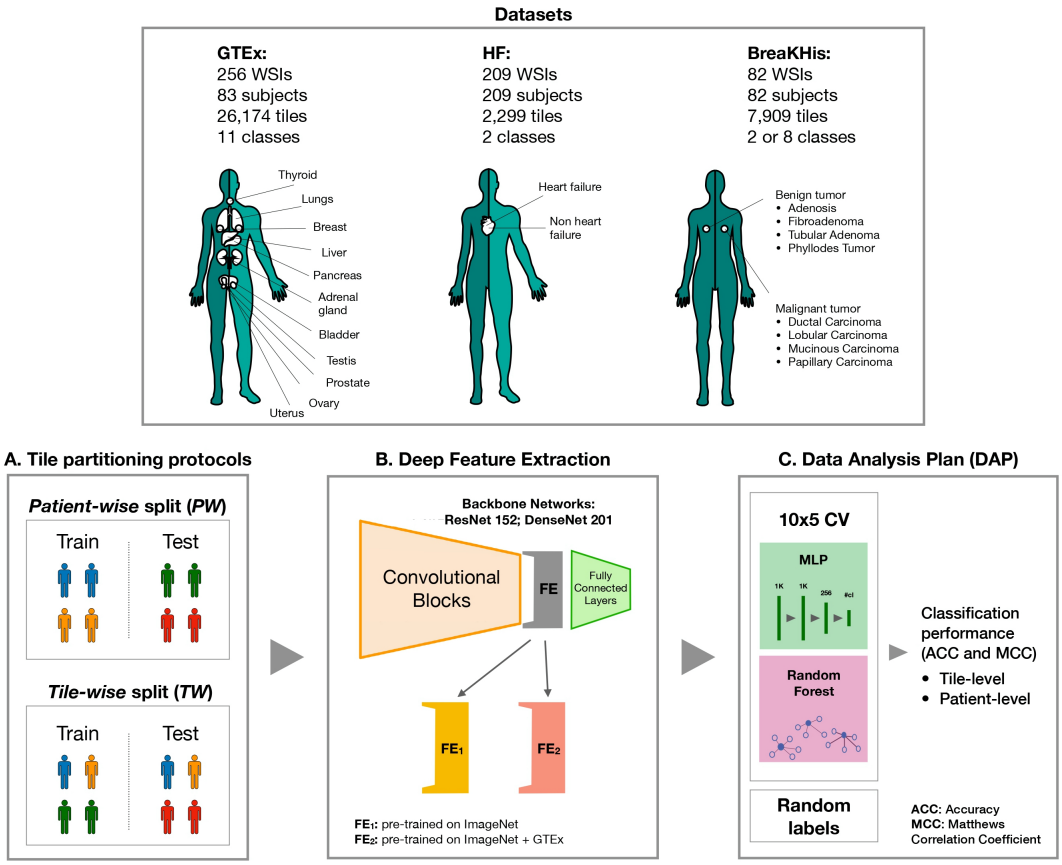

Figure 1: Experimental environment for evaluation of data leakage impact on machine learning models in digital pathology. (A) Tile datasets are split into train/test set following either the *Tile-Wise* or the *Patient-Wise* protocol; (B) the train set is used to train a backbone network for feature extraction, using different transfer learning strategies; (C) machine learning classifiers on the deep features are evaluated within the Data Analysis Plan.

Dataset	Subjects	WSIs	WSIs per Subject			Tiles	Tiles per Subject		
			Min	Max	Median		Min	Max	Median
GTEx	83	265	1	7	3	26,174	1	700	300
HF	209	209	1			2,299	11		
BreaKHis	82	82	1			2,013	9	62	21

Table 1: Statistics of the datasets considered in this study.

The GTEx dataset The current release of GTEx (v8) includes a total of $15,201$ H&E-stained WSIs, retrieved with an Aperio scanner ($20\times$ native magnification) and gathered from a cohort of 838 nondiseased donors[1]. In this work, we consider a subset of 265 WSIs randomly selected from 11 histological classes, for a total of 83 subjects. From this subset, we randomly selected a balanced number of WSIs per tissue: adrenal gland ($n = 24$); bladder ($n = 19$); breast ($n = 26$); liver ($n = 26$); lung ($n = 21$); ovary ($n = 26$); pancreas ($n = 26$); prostate ($n = 24$); testis ($n = 26$); thyroid ($n = 26$); uterus ($n = 21$).

We implemented a data preprocessing pipeline to prepare the tile dataset from the GTEx collection. First, the tissue region is automatically detected in each WSI; this process combines the *Otsu-threshold* binarization method [25] with the dilation and hole-filling morphological operations. A maximum of 100 tiles of size 512×512 is then randomly extracted from each slide. To ensure that

[1] https://gtexportal.org/home/releaseInfoPage

only high-informative images are used, tiles with tissue area that accounts for less than the 85% of the whole patch are automatically rejected. At the end of this step, a total of $26,174$ random tiles is extracted from the WSIs, each available at different magnification levels (i.e., $20\times, 10\times, 5\times$). In this paper we limit experiments and discussions to tiles at $5\times$ magnification, with no loss of generality.

The HF dataset The Heart Failure collection [24] originates from 209 H&E-stained WSIs of the left ventricular tissue, each corresponding to a single subject. The learning task is to distinguish images of *heart failure* ($n = 94$) from those of *non-heart failure* ($n = 115$). Slides in the former class are categorized according to the disease subtype: ischemic cardiomyopathy ($n = 51$); idiopathic dilated cardiomyopathy ($n = 41$); undocumented ($n = 2$). Subjects with no heart failure are further grouped in: normal cardiovascular function ($n = 41$); non-HF and no other pathology ($n = 72$); non-HF and other tissue pathology ($n = 2$). WSIs in this dataset have been acquired with an Aperio ScanScope at $20\times$ native magnification, and then downsampled at $5\times$ magnification by authors. From each WSI, 11 non-overlapping patches of size 250×250 were randomly extracted. The entire collection of $2,299$ tiles is publicly available on the Image Data Resource Repository [2] (IDR number: idr0042).

The BreaKHis dataset The BreakHis histopathological dataset [12] collects $7,909$ H&E-stained tiles (size 700×460) of malignant or benign breast tumour biopsies. Tiles correspond to regions of interest manually selected by expert pathologists from a cohort of 82 patients, and made available at different magnification factors, i.e., $40\times$, $100\times$, $200\times$, $400\times$) [12]. To allow for a more extensive comparison with the state of the art, only the $200\times$ magnification factor will be considered in this paper. The BreakHis dataset currently contains 4 histological distinct subtypes of benign, and malignant tumours, respectively: 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 ($n = 560$). This dataset is used for two classification tasks: (BreaKHis-2) binary classification of benign and malignant tumour samples; (BreaKHis-8) classification of the 8 distinct tumour subtypes.

3 Methods

The pipeline used in this work is based on the DAPPER framework for digital pathology [18], extended by (i) integrating specialised train-test splitting protocols, i.e. *Tile-Wise* and *Patient-Wise*; (ii) extending the feature extractor component with new backbone networks; (iii) applying two transfer learning strategies for feature embedding. Fig. 1 shows the three main blocks of the experimental environment defined in this paper: (A) dataset partition in train and test set; (B) feature extraction procedure with different transfer learning strategies; (C) the DAP employed for machine learning models.

A. Dataset partitioning protocols The tile dataset is partitioned in the *training* set and *test* set, considering 80% and 20% split ratio for the two sets, respectively. We compare two data partitioning protocols to investigate the impact of a train-test contamination (Fig. 1A): in the *Tile-Wise* (TW) protocol, tiles are randomly split between the training and the test sets, regardless of the original WSI. The *Patient-Wise* (PW) protocol splits the tile dataset strictly ensuring that all tiles extracted from the same subject are found either in the training or the test set. To avoid other sources of leakage due to class imbalance [26], the two protocols are both combined with stratification of samples over the corresponding classes, and any class imbalance is accounted for by weighting the error on generated predictions.

B. Deep Learning models and feature extraction The training set is then used to train a deep neural network for feature extraction (Fig. 1B), i.e. a "backbone" network whose aim is to learn a vector representation of the data (*features embedding*). In this study, we consider two backbone architectures in the residual network (ResNet) family, namely ResNet-152 [20] and DenseNet-201 [19]. Given that the DenseNet model has almost the double of parameters[3], and so a higher footprint in computational resources, diagnostic experiments and transfer learning are performed only with the ResNet-152 model. Similarly to [16], and [18], we started from off-the-shelf

[2] idr.openmicroscopy.org/

[3] DenseNet-201: $\sim$ 12M parameters; ResNet-152: $\sim$ 6M parameters.

version of the models, pre-trained on ImageNet, and then fine-tuned to the digital pathology domain using transfer learning. Specifically, we trained the whole network for 50 epochs with a learning rate $\eta = 1e-5$, and Adam optimizer [27], 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 regularization. To reduce the risk of overfitting, we use train-time data augmentation, namely random rotation and random flipping of the input tiles.

The impact of adopting a single or double-step transfer learning strategy in combination with the *Patient-Wise* partitioning protocol is also investigated in this study. Two sets of features embeddings (FE) are generated: FE_1, backbone model fine-tuned from ImageNet; FE_2, backbone model sequentially fine-tuned from ImageNet and GTEx.

C. Classification and Data Analysis Plan (DAP) The classification is finally performed on the feature embedding within a DAP for machine learning models (Fig. 1C). In this work, we compare the performance of two models: Random Forest (RF) and Multi-Layer Perceptron Network (MLP). In particular, we apply the 10×5-fold CV schema proposed by the MAQC-II Consortium [9]. In the DAP setting, the input datasets are the two separate training and test sets, as resulted from the 80-20 train-test split protocol. The test set is kept completely unseen to the model, and only used for the final evaluation. The training set further undergoes a 5-fold CV iterated 10 times with a different random seed, resulting in 50 separated internal *validation* sets. These validation sets are generated adopting the same protocols used in the previous train-test generation, namely *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) [28] and Accuracy (ACC), both with 95% Studentized bootstrap confidence intervals (CI). Moreover, results have been reported both at tile-level and at patient-level, in order to assess the ability of machine learning models to generalise on unseen subjects (see section 4).

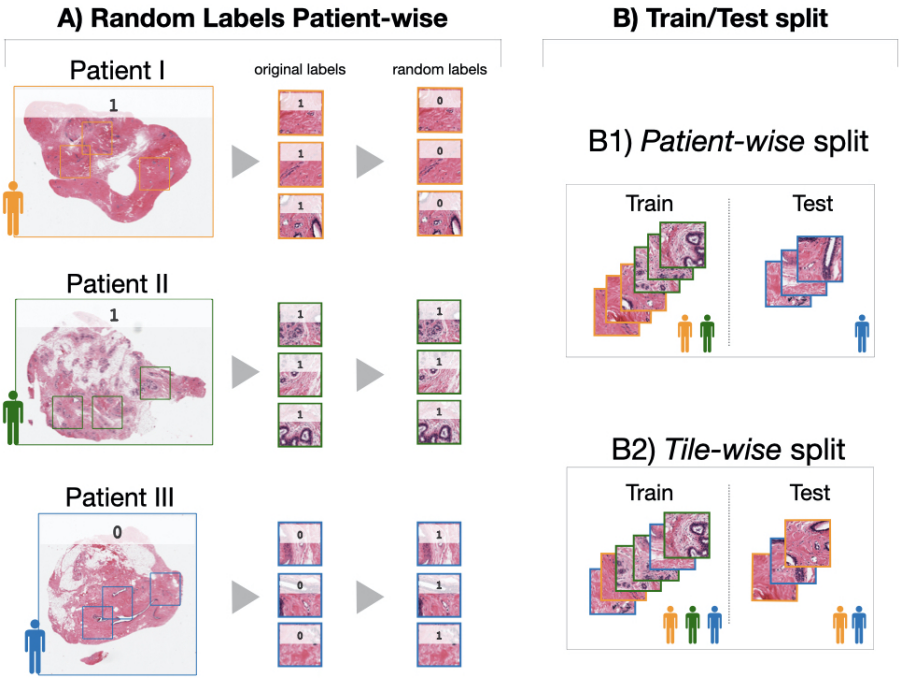

Figure 2: Random Labels experimental settings. A) The labels of the extracted tiles are randomly shuffled consistently with the original patient. B) The train/test split is then performed either *Patient-Wise* or *Tile-Wise*.

As an additional caution to check for selection bias, the DAP integrates a *random labels* schema (RLab) (Fig. 2). In this setting, the training labels are randomly shuffled and presented as reference ground truth to the machine learning models. In particular, we consistently randomize the labels for all the tiles of a single subject, thus they would all share the same random label (Fig. 2A); then

we alternatively use the *Patient-Wise* (Fig. 2B1) or the *Tile-Wise* (Fig. 2B2) splits within the DAP environment. Notice that an average MCC score close to zero ($MCC \approx 0$) indicates a protocol immune from sources of bias, including data leakage; we focus on the RLab validation to emphasise evidence of data leakage derived from the *TW* and the *PW* protocols.

Performance metrics Several patient-wise performance metrics have been defined in the literature [12, 29, 24]. Two metrics are considered in this study: (1) *Winner-takes-all* (WA), and (2) *Patient Score* (PS).

In the WA metric, the label associated to each patient corresponds to the majority of the labels predicted for their tiles. With this strategy, standard metrics based on the classification confusion matrix can be used as overall performance indicators. In this paper, ACC is used for comparability with the PS metric. The PS metric is defined for each patient [12] as the ratio of the N_c correctly classified tiles over the N_P total number of tiles per patient, namely $PS = \frac{N_c}{N_P}$. The overall performance is then calculated using the *global recognition rate* (RR), defined as the average of all the PS scores for all patients:

$$RR = \frac{\sum PS}{|P|}$$

In this paper, the WA metric and the PS metric are used for comparison of patient-level results on the HF dataset and the BreaKHis dataset, respectively.

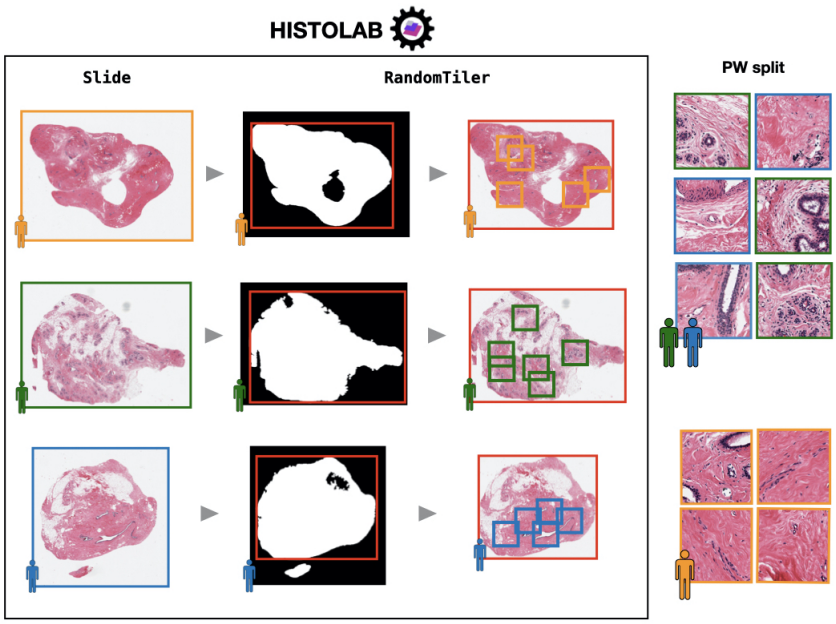

Figure 3: Workflow of the proposed protocol against data leakage in digital pathology, using the histolab software. The documentation of histolab is available at http://histolab.readthedocs.io.

Preventing Data Leakage: the `histolab` library As a solution to the data leakage pitfall, we have developed a protocol for image and tile splitting based on `histolab`, an open source software recently developed for reproducible WSI preprocessing in digital pathology. This library implements a tile extraction procedure, whose reliability and quality result from robust design, and extensive software testing. A high level interface for image transformation is also provided, making `histolab` an easy-to-adopt tool for complex histopathological pipelines.

In order to intercept data leakage conditions, the protocol is designed to create a data-leakage free collection (tile extraction with the *Patient-Wise* split) that can be easily integrated in a deep learning workflow (Fig. 3). The protocol is already customized for standardizing WSI preprocessing on

Dataset	MLP		RF		MLP		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 [30]
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 [31]

Table 2: DAP results for each classifier head, using the *Tile-Wise* partitioning protocol, and the FE_1 feature embedding with the ResNet-152 as backbone model. The average cross validation metrics ($\mathrm{MCC_v}$ and $\mathrm{ACC_v}$) with 95% CI are reported for each classification task, along with metrics on the test set ($\mathrm{MCC_t}$ and $\mathrm{ACC_t}$). The *Others* column reports the highest accuracy achieved among the compared papers.

GTEx and TCGA, two large scale public repositories that are widely used in computational pathology. The code can be also adapted to rebuild the training and test datasets from GTEx used in this study, thus extending the HINT collection presented in [18].

4 Results

Data Leakage effects on classification outcome The results of the four classification tasks using the ResNet-152 pre-trained on ImageNet as backbone model (i.e. feature vectors FE_1) are reported in Table 2 and Table 3, with 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 presented, along with results on the test set (i.e. $\mathrm{MCC_t}$, and $\mathrm{ACC_t}$). State of the art results (i.e. *Others*) are also reported for comparison, whenever available.

Dataset	MLP		RF		MLP		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 [24]
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 [29]
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 [29]

Table 3: DAP results for each classifier head, using the *Patient-Wise* partitioning protocol, and the FE_1 feature embedding with the ResNet-152 as backbone model. The average cross validation metrics ($\mathrm{MCC_v}$ and $\mathrm{ACC_v}$) with 95% CI are reported for each classification task, along with metrics on the test set ($\mathrm{MCC_t}$ and $\mathrm{ACC_t}$). The *Others* column reports the highest accuracy achieved among the compared papers.

As expected, estimates are more favourable for the *TW* protocol (Tab. 2) with respect to the *PW* one (Tab. 3), both in validation and in test and consistently for all the datasets. Moreover, the inflation of the *Tile-Wise* estimates is amplified in the multi-class setting (see BreaKHis-2 vs BreaKHis-8). Notably, these results are comparable with those in the literature, suggesting the evidence of a data leakage for studies adopting the *Tile-Wise* splitting strategy. Results on the GTEx dataset do not suggest significant differences using the two protocols; however both MCC and ACC metrics lie in a very high range. Analogous results (not reported here) were obtained using the DenseNet-201 backbone model, further confirming the generality of the derived conclusions.

Random Labels detects signal in the *Tile-Wise* split A data leakage effect is signalled for the *Tile-Wise* partitioning with a MCC consistently positive in the RLab validation schema (Sect. 3). For instance, as for BreaKHis-2 coupled with MLP, $\mathrm{MCC_{RL}}= 0.354\ (0.319, 0.392)$ 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, with corresponding $\mathrm{ACC_{RL}}$ values also included for completeness. Notably, all the tests using the *Patient-Wise* split perform as expected, i.e. with median values near 0, whereas results of the *Tile-Wise* case exhibit a high variability, especially for the BreaKHis-2 dataset (Fig.4).

Dataset	$\mathbf{MCC_{RL}}$		$\mathbf{ACC_{RL}}$	
	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)

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

Dataset	MLP		RF		MLP		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 [24]
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 [29]
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 [29]

Table 5: DAP results for each classifier head, using the *Patient-Wise* partitioning protocol, and the FE_2 feature embedding with ResNet-152 as 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 $\mathrm{ACC_t}$). The *Others* column reports the highest accuracy achieved among the compared papers.

Benefits of domain-specific transfer learning The adoption of the GTEx domain-specific dataset for transfer learning (Table 5) proves to be beneficial over the use of ImageNet only (Table 3). Notably, the *Patient-Wise* partitioning protocol with the FE_2 embedding have comparable performance with FE_1 and the inflated TW splitting (Tab. 2). However, minor improvements are achieved on the BreaKHis-8 task, with results not reaching state of the art. It must be observed that the BreaKHis dataset is highly imbalanced in the multi-class task. As a countermeasure, authors in [34, 35] adopted a balancing strategy during data augmentation, which we did not introduce here for comparability with the other experiments.

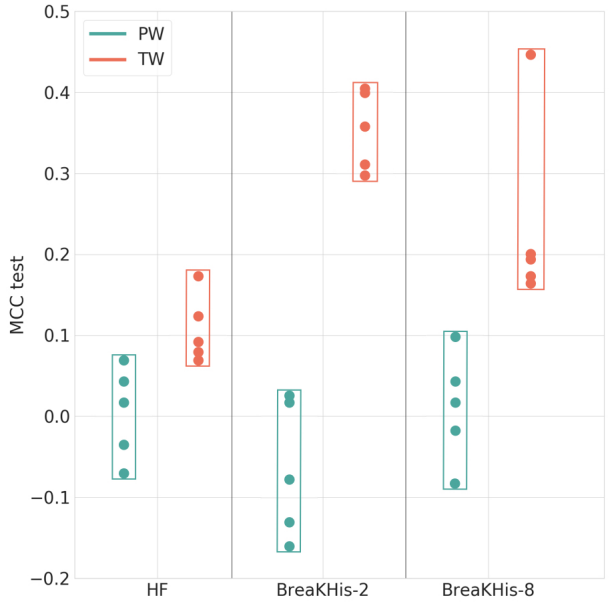

Figure 4: $\mathrm{MCC_{RL}}$ results on the test set. TW: *Tile-Wise*, PW: *Patient-Wise*.

Dataset	Metric	Partitioning Protocol	MLP		RF		Others
			$\mathrm{ACC_v}$	$\mathrm{ACC_t}$	$\mathrm{ACC_v}$	$\mathrm{ACC_t}$	$\mathrm{ACC_t}$
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 [24]
BreaKHis-2	PS	TW	0.995 (0.994, 0.996)	0.997	0.997 (0.996, 0.998)	0.998	0.872 [32]
		PW	0.864 (0.851, 0.877)	0.885	0.883 (0.869, 0.898)	0.893	0.976 [29]
BreaKHis-8	PS	TW	0.963 (0.960, 0.967)	0.950	0.957 (0.955, 0.959)	0.962	0.964 [33]
		PW	0.687 (0.667, 0.709)	0.752	0.705 (0.685, 0.728)	0.725	0.967 [29]

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 ($\mathrm{ACC_v}$), and corresponding scores on the test set ($\mathrm{ACC_t}$), are reported. The *Others* column reports the highest accuracy achieved among the compared papers.

To verify how much of previous domain-knowledge can be still re-used for the original task, we devised an additional experiment on the GTEx dataset: on the *Feature Extractor* component (i.e. Convolutional Layers) of the model trained on GTEx and fine-tuned on BreakHis-2, we add back the MLP classifier of the model trained on GTEx. Notably, this configuration recover high predictive performance (i.e. $\mathrm{MCC_t}$=0.983) on the classification task after only a single epoch of full training on GTEx.

Patient-level Performance Analysis We report patient-wise performance using the ResNet-152 backbone model with either the FE_1 feature embedding and both *Tile-Wise* and *Patient-Wise* protocols (Table 6), or with the FE_2 strategy and the *Patient-Wise* split (Table 7).

Dataset	Patient-level Metric	MLP		RF		Others
		$\mathrm{ACC_v}$	$\mathrm{ACC_t}$	$\mathrm{ACC_v}$	$\mathrm{ACC_t}$	$\mathrm{ACC_v}$
HF	WA	0.992 (0.989, 0.995)	0.976	0.989 (0.984, 0.992)	0.976	0.940 [24]
BreaKHis-2	PS	0.941 (0.930, 0.951)	0.971	0.958 (0.948, 0.968)	0.991	0.976 [29]
BreaKHis-8	PS	0.691 (0.669, 0.716)	0.721	0.699 (0.676, 0.723)	0.724	0.967 [29]

Table 7: Patient-level results for each classifier head, with 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 ($\mathrm{ACC_v}$) and corresponding scores on the test set ($\mathrm{ACC_t}$) are reported. The *Others* column reports the highest accuracy achieved among the compared papers.

5 Discussion

We report here a short description of the approach employed by comparable studies on the same datasets considered in this work; we refer to a *Patient-Wise* partitioning protocol when the authors clearly state the adoption of a train-test split consistent with the patient, or when the code is provided as reference. Notice that the different accuracy scores obtained for deep learning models applied on the same data can be explained by the adoption of diverse experimental protocols (e.g. preprocessing, data augmentation, transfer learning methods).

Nirschl et al. [24] train a CNN on the HF dataset to distinguish patients with or without heart failure. They systematically apply the *Patient-Wise* rule for the initial train-test split (50-50) and for the training partition into three-folds for cross-validation. Data augmentation strategies are also applied, including random cropping, rotation, mirroring, and staining augmentation. As for the BreaKHis dataset, Alom et al. [29] use a 70-30 *Patient-Wise* partitioning protocol to train a CNN with several (not specified) hidden layers, reporting average results from 5-fold cross-validation. Further, the authors apply augmentation strategies (i.e., rotation, shifting, flipping) to increase the

dataset by a factor of $21\times$ for each magnification level. The work of Han et al. [34] propose a novel CNN adopting a *Tile-Wise* partition with the training set accounting for the 50% of the dataset. Data augmentation (i.e. intensity variation, rotation, translation, and flipping) is used to adjust for imbalanced classes. Jiang et al. [30] train two different variants of the ResNet model to address the binary and the multi-class task, for each magnification factor. They adopt a *Tile-Wise* partitioning protocol for the train-test split, using 60% and 70% of the data in the training set for BreaKHis-2 and BreaKHis-8, respectively. Data augmentation is also exploited in the training process, and experiments are repeated 3 times.

Other authors employed a similar protocol to address the BreaKHis-8 task by training a CNN pretrained on ImageNet: Nawaz et al. [33] implemented a DenseNet-inspired model, while Nguyen et al. [36] choose a custom CNN model, instead. Both studies use a *Tile-Wise* partition on the BreaKHis dataset (70-30 and 90-10, respectively), and do not apply any data augmentation. Xie et al. [32] adapt a pre-trained ResNet-V2 to the binary and multiclass tasks of BreaKHis, at different magnification factors, using a 70-30 *Tile-Wise* partition. Data augmentation has been applied to balance the least represented class in BreaKHis-8. Jannesary et al. [31] used a 90-10 *Tile-Wise* train-test split with data augmentation (i.e. resizing, rotations, cropping and flipping) to fine-tune a ResNet-V1 for binary and multi-class prediction. Moreover, experiments in [31] were performed combining images at different magnification factors in a unified dataset. Finally, both [37] and [38] used a *Tile-Wise* train-test split for prediction of malignant vs benign samples using a pre-trained CNN and [38] also employed data augmentation (rotation and flipping).

6 Conclusions

Possibly even more than other areas of computational biology, digital pathology faces the risk of data leakage. The first part of this study clearly demonstrates the impact of weakly designed experiments with deep learning for digital pathology. In particular, we found that the predictive performance estimates are inflated if the DAP does not flawlessly concentrate the subject and/or the tissue specimen from which tiles are extracted either in the training or test datasets. Fortunately, many studies already adopt the correct procedure [16, 17, 12, 24, 35, 34]. However, we argue that this subtle form of selection bias still constitutes a threat to reproducibility of AI models that may have affected a considerable number of works. Indeed, a significant number of studies considered in this work do not explicitly mention the patient-wise strategy [39, 33, 32, 30, 31, 40]. We encourage the community to adopt our code (https://github.com/histolab/histolab/tree/master/examples) as a launchpad for reproducibility of AI pipelines in digital pathology.

References

[1] J. P. A. Ioannidis et al. Repeatability of published microarray gene expression analyses. *Nature Genetics*, 41(2):149, 2009.

[2] S. A. Iqbal et al. Reproducible research practices and transparency across the biomedical literature. *PLoS Biology*, 14(1):e1002333, 2016.

[3] National Academies of Sciences, Engineering, and Medicine, Policy and Global Affairs. *Reproducibility and Replicability in Science*. National Academies Press, 2019.

[4] J. T. Leek et al. Tackling the widespread and critical impact of batch effects in high-throughput data. *Nature Reviews Genetics*, 11(10):733, 2010.

[5] S. Moossavi et al. Repeatability and reproducibility assessment in a large-scale population-based microbiota study: case study on human milk microbiota. bioRxiv:2020.04.20.052035, 2020.

[6] B. O. Turner et al. Small sample sizes reduce the replicability of task-based fMRI studies. *Communications Biology*, 1(1):1–10, 2018.

[7] A. Barla et al. Machine learning methods for predictive proteomics. *Briefings in Bioinformatics*, 9(2):119–128, 2008.

[8] L. Peixoto et al. How data analysis affects power, reproducibility and biological insight of RNA-seq studies in complex datasets. *Nucleic Acids Research*, 43(16):7664–7674, 2015.

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

[10] T. Ching et al. Opportunities and obstacles for deep learning in biology and medicine. *Journal of The Royal Society Interface*, 15(141):20170387, 2018.

[11] N. Saravanan et al. Data wrangling and data leakage in machine learning for healthcare. *International Journal of Emerging Technologies and Innovative Research*, 5(8):553–557, 2018.

[12] F. A. Spanhol et al. A Dataset for Breast Cancer Histopathological Image Classification. *IEEE Transaction in Biomedical Engineering*, 63(7):1455–1462, 2016.

[13] F. Shahidi et al. Breast Cancer Classification Using Deep Learning Approaches and Histopathology Image: A Comparison Study. *IEEE Access*, 8:187531–187552, 2020.

[14] S. Cohen. *Artificial Intelligence and Deep Learning in Pathology*. Elsevier, 2020.

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

[16] R. Mormont et al. 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.

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

[18] A. Bizzego et al. Evaluating reproducibility of AI algorithms in digital pathology with DAPPER. *PLOS Computational Biology*, 15(3):1–24, 2019.

[19] G. Huang et al. Densely Connected Convolutional Networks. In *Proceedings of the 2018 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2261–2269. IEEE, 2018.

[20] K. He et al. 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.

[21] L. Barisoni et al. Digital pathology and computational image analysis in nephropathology. *Nature Reviews Nephrology*, 16:669–685, 2020.

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

[23] K. Tomczak et al. The Cancer Genome Atlas (TCGA): an immeasurable source of knowledge. *Contemporary Oncology*, 19(1A):A68, 2015.

[24] J. J. Nirschl et al. A deep-learning classifier identifies patients with clinical heart failure using whole-slide images of H&E tissue. *PLOS ONE*, 13(4):e0192726, 2018.

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

[26] S. Raschka. Model evaluation, model selection, and algorithm selection in machine learning. arXiv:1811.12808v3, 2020.

[27] D. P. Kingma et al. Adam: A Method for Stochastic Optimization. arXiv:1412.6980; Published as a conference paper at ICLR 2015, 2014.

[28] G. Jurman et al. A Comparison of MCC and CEN Error Measures in Multi-Class Prediction. *PLOS ONE*, 7(8):1–8, 08 2012.

[29] M. Z. Alom et al. Breast Cancer Classification from Histopathological Images with Inception Recurrent Residual Convolutional Neural Network. *Journal of Digital Imaging*, 32(4):605–617, 2019.

[30] Y. Jiang et al. Breast cancer histopathological image classification using convolutional neural networks with small SE-ResNet module. *PLOS ONE*, 14(3):e0214587, 2019.

[31] M. Jannesari et al. Breast Cancer Histopathological Image Classification: A Deep Learning Approach. In *Proceedings of the 2018 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, pages 2405–2412, 2018.

[32] J. Xie et al. Deep Learning Based Analysis of Histopathological Images of Breast Cancer. *Frontiers in Genetics*, 10:80, 2019.

[33] M. Nawaz et al. Multi-class breast cancer classification using deep learning convolutional neural network. *International Journal of Advanced Computer Science and Applications*, 9(6):316–332, 2018.

[34] Z. Han et al. Breast Cancer Multi-classification from Histopathological Images with Structured Deep Learning Model. *Scientific Reports*, 7(1):4172, 2017.

[35] M. J. Alom et al. Advanced Deep Convolutional Neural Network Approaches for Digital Pathology Image Analysis: a comprehensive evaluation with different use cases. arXiv:1904.09075, 2019.

[36] P. T. Nguyen et al. Multiclass Breast Cancer Classification Using Convolutional Neural Network. In *Proceedings of the 2019 International Symposium on Electrical and Electronics Engineering (ISEE)*, pages 130–134. IEEE, 2019.

[37] E. Deniz et al. Transfer learning based histopathologic image classification for breast cancer detection. *Health Information Science and Systems*, 6(1):18, 2018.

[38] J. L. Myung et al. Deep Convolution Neural Networks for Medical Image Analysis. *International Journal of Engineering & Technology*, 7(3):115–119, 2018.

[39] X. Pan et al. Multi-task deep learning for fine-grained classification and grading in breast cancer histopathological images. In *Proceedings of the 2018 Cognitive Internet of Things: Frameworks, Tools and Applications (ISAIR)*, volume 810 of *Studies in Computational Intelligence*, pages 85–95. Springer, 2020.

[40] R. M. Shallu. Breast cancer histology images classification: Training from scratch or transfer learning? *ICT Express*, 4(4):247–254, 2018.