# RNA-Expression Prediction from Whole-Slide Images Using Attention-Based Feature Extraction and Fundamental Models

Shay Reardon[1†]

[1]Department of Computer Information and Science, University of Florida, 903 West University Avenue, Gainesville, 32601-5117, Florida, USA.

Contributing authors: reardons@ufl.edu;
[†]These authors contributed equally to this work.

**Abstract**

Cancer is a dynamic group of diseases that are characterized by significant heterogeneity acquired throughout the course of its evolution [1–4]. To meet this problem of heterogeneity, the field of precision medicine has developed RNA-sequencing (RNA-seq) technology to advance our understanding of gene expression in cancer and tailor prognosis to the patient. However, high costs and scalability challenges hinder their widespread adoption. Studies have explored the use of Whole-Slide Images (WSI) as a cost-effective alternative for predicting transcriptional heterogeneity. We introduce a novel preprocessing method, integrating the cutting-edge CLAM and UNI pipelines, to help assist model performance when predicting transcriptional data from WSIs. Here we show that the introduced pipeline effectively enhance model performance on the feature space even in models constrained by computational power. We found that ridge regression preformed the best across all tested models on test Pearson's Correlation Coefficient ($r = 0.19 \pm 0.01$) with lasso regression and RBF-SVM preforming close behind ($r = 0.18 \pm 0.00$; $r = 0.18 \pm 0.00$). Our implemented transformer MLP achieved the highest testing RMSE of the implemented models at $1.51 \pm 0.03$. Our results demonstrate that by introducing the CLAM pipeline into preprocessing approaches for transcriptional data prediction, models can successfully predict gene expression data for bladder cancer. We anticipate that our work will contribute to more effectively integrating precision medicine into oncological workflows, ultimately improving their ability to account for heterogeneity.

**Keywords:** keyword1, Keyword2, Keyword3, Keyword4

# 1 Author Summary

This paper aims to replace the commonly used CNN in whole slide image (WSI) patch segmentation workflows for RNA-sequence expression prediction. We introduce the CLAM attention-based patch extraction model to help improve generalizability of the models. We test ridge regression, lasso regression, RBF-SVM, XGBoost, and two multilayer perceptron (MLP) methods supported by the UNI feature extractor for gene expression prediction in bladder cancer tissue samples.

# 2 Introduction

Cancer is a dynamic and intricate group of diseases that are characterized by abnormal cell growth and a significant heterogeneity acquired over-time. This heterogeneity arises from various patient- and cell-specific factors, including genetic variations, epigenetic modifications, transcriptomic differences, and differential gene expression [2–4]. Broadly, heterogeneity is classified into inter-patient heterogeneity and intra-tumoral heterogeneity [1]. Inter-patient heterogeneity is the genetic difference in cancers between patients, even if a histopathological diagnosis is shared, and is strongly influenced by patient-specific factors [1, 5]. Intra-tumoral heterogeneity refers to the genomic and transcriptomic differences found within distinct cell sub-populations in a single-tissue sample [1, 6]. Understanding cancer heterogeneity and developing systems that account for its diverse attributes are critical for improving patient outcomes and providing effective diagnosis.

Precision medicine offers a promising approach to these challenges. By utilizing molecular profiling and gene expression, precision medicine hopes to tailor prognosis to each patient and tumor. RNA-sequencing (RNA-seq) technology have been instrumental in advancing our understanding of gene expression in cancer. Bulk RNA-seq has enabled differential gene expression analysis and biomarker discovery at the cancer tissue-level, discovering novel kinase gene fusions to help guide targeted therapies [7, 8]. Spatial transcriptomics tools, such as 10x Genomics Visium or STARmap, provides a mapping of gene expression levels across cancer tissue while preserving spatial context[9, 10]. Single-cell RNA sequencing (scRNA-seq) has enabled researchers to analyze transcriptional heterogeneity at a single-cell resolution, revealing critical insights into tumor microenvironments and drug resistance mechanisms [11]. These advancements have expanded our understanding of heterogeneity in cancers and its role in the disease's progression. However, despite their potential in improving patient outcomes, high costs and scalability challenges hinder their widespread clinical adoption. Addressing these barriers is essential for realizing their full impact on improving patient outcomes and assisting in diagnosis.

To address the expensive nature and scalability issues present in current RNA-seq technologies, studies have explored the use of Whole Slide Images (WSI) – high resolution digitization of tissue samples on slides – as an alternative approach. WSI are collected routinely in cancer diagnosis and are in-expensive to add into existing workflows. In the recent decade, machine learning and deep learning approaches have shown great success in adapting morphological features described in WSIs to approximate RNA-seq expression data. Attention-based workflows, such as CLAM and UNI,

have been developed to extract meaningful features from these slides and have performed well across tumor classification tasks [12, 13]. Convolutional neural networks (CNNs) approaches with deep learning have also shown great success in this field, HE2RNA and SEQUOIA perform extremely well in predicting gene expression profiles across various cancer types [14, 15]. Methods such as these helps bridge the gap between the morphological and transcriptomic, helping move away from the intensive workflows that prevent the wide-scale adoption of RNA-seq workflows.

Current workflows fail to take complete advantage of emerging pipelines and pre-processing models, hindering their full ability from understanding the morphological features on the WSIs. Segmentation of the slides is handled via CNNs by top models, such as HE2RNA and SEQUOIA, using thresholding functions to detect cancerous tissue from the background. In this study, we propose the integration of CLAM over commonly used CNNs in the segmentation of WSIs. CLAM is an attention-based model that segments a slide into predefined patches. With instance-based clustering, the model is able to refine the feature space to improve the accuracy of downstream models [12]. For testing, we had attempted to implement the SEQUOIA linearized transformer and MLP with no success. The current publicly available SEQUOIA model does not function correctly with the data. Instead, we opted for the implementation of base-line machine learning algorithms such as ridge regression, lasso regression, RBF-SVM, XGBoost, and two multilayer perceptron (MLP) due to time constraints. By incorporating these models with CLAM, we aim to generally improve upon their ability to understand morphological features and generalization capabilities for transcriptomic profiling. This approach represents a step forward in bridging histopathological analysis with transcriptomics data for precision medicine applications.

# 3 Results

We introduce a novel pre-processing application for the CLAM and UNI pipeline to increase the performance of downstream models in predicting gene expression from whole slide images (WSI). CLAM is a patching method used to segment each WSI into patches while masking out the background and holes in the tissue. Patch feature extraction is then accomplished through the UNI model. To test this proposed pre-processing pipeline, we implement three classical machine learning models (ridge regression, lasso regression, and RBF-SVM), an ensemble model (XGBoost), and two neural network models (Fig. 1, Materials & Methods). The first neural network is a classical multilayer perceptron (MLP) learner. The second is an MLP implemented with a transformer head, similar to that found in SEQUOIA [15]. A set of WSI with matched transcriptomics data was curated from The Cancer Genome Atlas (TCGA) for training and testing [16]. We confined our study to the bladder carcinomas (TCGA-BLCA) cohort. The performance of each model was determined via a combination of root mean squared error (RMSE) and Pearson's Correlation Coefficient per fold (Materials & Methods).

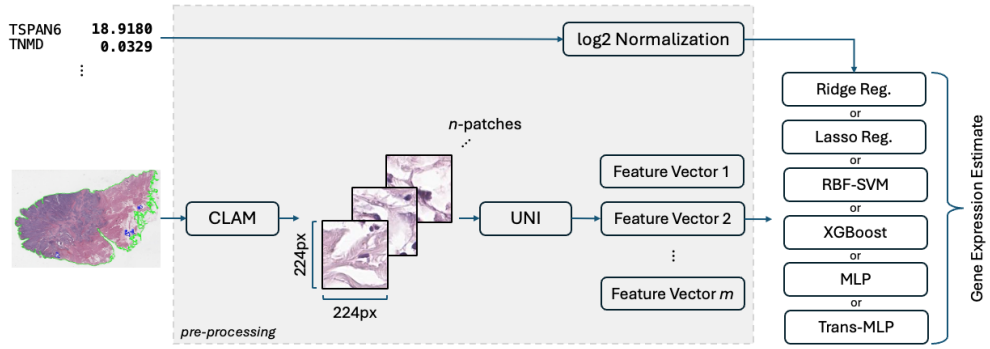



**Fig. 1**: **Proposed Pre-processing Pipeline** — During training, gene expression quantification values and WSI data are passed through the pre-processing pipeline. Gene expression values are inputted as a table to the log2 transform unit where they undergo a log 2 transformation. WSI images are first passed to the CLAM pipeline for segmentation into n 224px-by-224px patches, each patch stored as a coordinate for their respective files. UNI then extracts m=100 feature vectors from the patches. Features from the UNI and log2 transforms are then passed to the chosen machine learning model: Ridge Regression, Lasso Regression, RBF-SVM, XGBoost, or a MLP. This module then outputs the expected gene expression estimation. For testing, the log2 normalized module is not connected to the machine learning models and is only used to calculate performance metrics.

## 3.1 Classical Machine Learning with Pre-processing Pipeline

To validate the proposed preprocessing pipeline, we first evaluated its performance using classical machine learning models: ridge regression, lasso regression, and RBF-SVM (Materials & Methods). Ridge regression exhibited stable performance across folds, with test RMSE values ranging from 1.31 (fold 1) to 1.25 (fold 5) (Fig. 2a). The model showed strong bias toward the training set, with an average difference of $0.25 \pm 0.00$ between training and testing Pearson's Correlation values. Test Pearson's Correlation peaked at fold 2 ($r = 0.21$), but was lowest at fold 5 ($r = 0.18$) (Fig. 2b). The average RMSE and Pearson's Correlation values across all folds are summarized in Table 1. The consistent performance of ridge regression across folds may be attributed to its regularization mechanism, which mitigates overfitting by penalizing large coefficients (Materials & Methods).

Lasso regression displayed similar patterns to ridge regression but showed slight improvements on certain folds. Training Pearson's Correlation remained consistent at an average of $0.33 \pm 0.00$, while testing Pearson's Correlation fluctuated around $0.18 \pm 0.00$ (Fig. 2b, Table 1). Test RMSE ranged from 1.31 (fold 1) to 1.25 (fold 5), with fold 4 showing the greatest variance (RMSE = 1.29) (Fig. 2a). Fold 4 shows the greatest variance across all models, potentially indicating a shift in the input data distribution that challenges the stability of the models. Comparative analysis reveals that ridge regression outperformed lasso regression in test RMSE by an average of

| Average | Train Pearson's (r) | Train RMSE | Test Pearson's (r) | Test RMSE |
|---|---|---|---|---|
| **Ridge** | 0.44 ± 0.01 | 1.16 ± 0.01 | 0.19 ± 0.01 | 1.27 ± 0.02 |
| **Lasso** | 0.33 ± 0.01 | 1.21 ± 0.01 | 0.18 ± 0.00 | 1.28 ± 0.02 |
| **RBF-SVM** | 0.60 ± 0.01 | 1.04 ± 0.01 | 0.18 ± 0.00 | 1.29 ± 0.02 |
| **XGBoost** | 0.95 ± 0.00 | 0.61 ± 0.00 | 0.13 ± 0.00 | 1.41 ± 0.03 |
| **Class. MLP** | 0.48 ± 0.02 | 1.15 ± 0.02 | 0.14 ± 0.02 | 1.41 ± 0.03 |
| **Trans. MLP** | 0.55 ± 0.05 | 1.09 ± 0.06 | 0.12 ± 0.02 | 1.51 ± 0.03 |

**Table 1**: **Average Pearson's Correlation and RMSE per Model** — For each fold, the average RMSE and Pearson's Correlation Coefficient are calculated per model (Materials & Methods). These metrics are used to evaluate the predictive performance of each model.

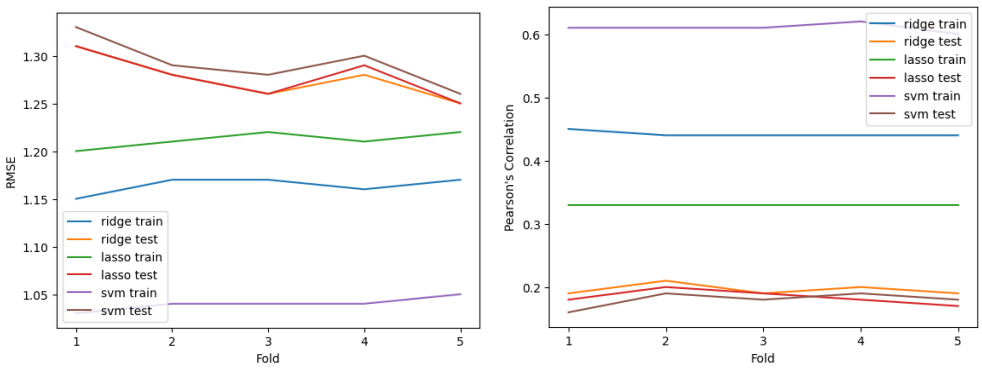
0.01, while lasso regression achieved slightly higher Pearson's Correlation values by an average of 0.01 across folds (Fig. 2a, Table 1).

RBF-SVM performed unexpectedly across all folds, showing no significant improvements in test Pearson's Correlation compared to ridge regression (r = -0.01) or lasso regression (r = 0.00) (Fig. 2b). Additionally, it did not demonstrate greater stability in RMSE. The average difference from the test RMSE mean was 0.09, which is comparable to both ridge and lasso regression (Fig. 2a). RBF-SVM achieved the lowest training RMSE (1.04 ± 0.00), as shown in Table 1, but its testing RMSE (1.29 ± 0.02) was higher than both that of ridge regression and lasso regression. This potentially indicates the model is overfitting. While RBF-SVM performed better on the training set for Pearson's Correlation, it exhibited significantly worse than both other models in test RMSE (Table 1). A grid search was done to optimize the normalization factor and epsilon-tube width; however, this led to a degradation in test Pearson's Correlation by approximately 7.68%, potentially due to overfitting on specific training folds. Further testing with this kernel may be required to achieve better correlation performance.

These results indicate that ridge regression and lasso regression provide comparable performance in terms of RMSE and Pearson's Correlation on the TCGA-BLCA cohort data, with minor differences between models across folds. RBF-SVM demonstrated superior RMSE performance but struggled to capture meaningful correlations within the testing data, highlighting potential limitations in its ability to generalize under this preprocessing pipeline (Fig. 2a). Pearson's Correlation Coefficient for all models was significantly lower during testing compared to training, with ridge regression achieving the highest average correlation (r = 0.19) (Fig. 2b). The consistent performance of ridge and lasso regression highlights their suitability for datasets with moderate feature complexity, as demonstrated by their stable RMSE and correlation metrics.

## 3.2 XGBoost Results with Pre-processing Pipeline

The proposed preprocessing pipeline was then evaluated using the ensemble method XGBoost, which we hypothesized would perform well due to the tabular nature of the data. The model demonstrated consistent performance on the testing data, achieving an average test RMSE of 1.41 ± 0.03, with significant deterioration observed only in folds 3 and 5 (Fig. 3a, Table 1). This deterioration may indicate variability in the

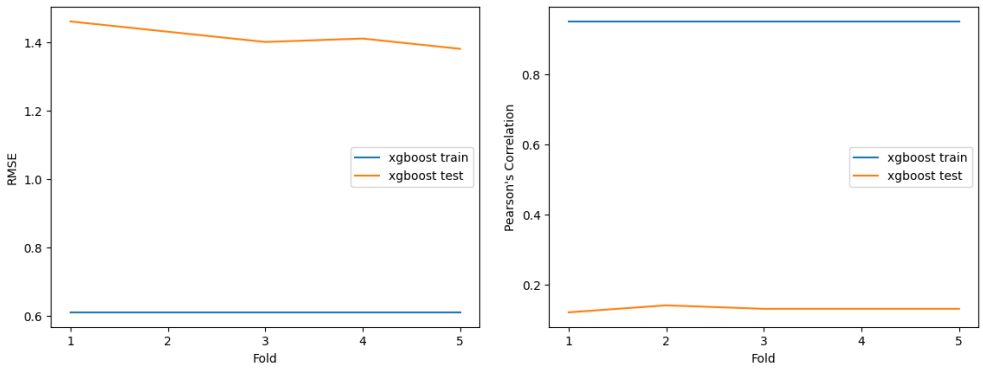



(a) (b)

**Fig. 2**: **Evaluating Classical Machine Learning Models** — Panel (a) shows RMSE values for ridge regression, lasso regression, and RBF-SVM across five cross-validation folds, while panel (b) displays Pearson's Correlation Coefficients for the same models. Each fold holds out 20% of the dataset for testing, this held-out data is unique for each fold. Ridge regression and lasso regression demonstrated consistent RMSE performance across folds, while RBF-SVM showed higher variability between training and testing sets. Pearson's Correlation was comparable across all models on the test set, but varied significantly on the training set.

underlying data distribution or outliers within these specific folds, which could affect model performance. There are no signs of overfitting as evidence by an average difference between training and testing sets of $0.80 \pm 0.03$ (Fig. 3a). In terms of Pearson's Correlation Coefficient, XGBoost exhibited a trend worse than the classical machine learning methods tested. Although the correlation remained consistent throughout the training and testing sets, the average testing correlation was low ($r = 0.13 \pm 0.00$) across all folds (Fig. 3b, Table 1). This indicates that the model struggled to capture meaningful relationships from the feature space despite the strong training performance ($r = 0.95 \pm 0.00$) (Fig. 3b, Table 1). Additionally, the model performs worse against folds 3 and 5, indicating there may be differences in the data distribution caused by the 5-fold cross validation method (Fig. 3b). These results suggest the model struggled to generalize correlations to unseen data, meaning some parameter tuning is needed. A grid search was performed to optimize hyperparameters. While this resulted in a slight increase in RMSE, it also resulted in the overall degradation in the Pearson's Correlation Coefficient. Consequently, we retained the original model configuration to preserve the higher correlation performance. Additional optimization of this model may improve its correlation performance.

When compared to the classical machine learning models, XGBoost demonstrated higher overall test RMSE ($1.41 \pm 0.03$) than ridge regression ($1.28 \pm 0.02$), lasso regression ($1.29 \pm 0.03$) and RBF-SVM ($1.29 \pm 0.02$) (Table 1). XGBoost also demonstrated greater consistency across folds, with significant deterioration only being observed in



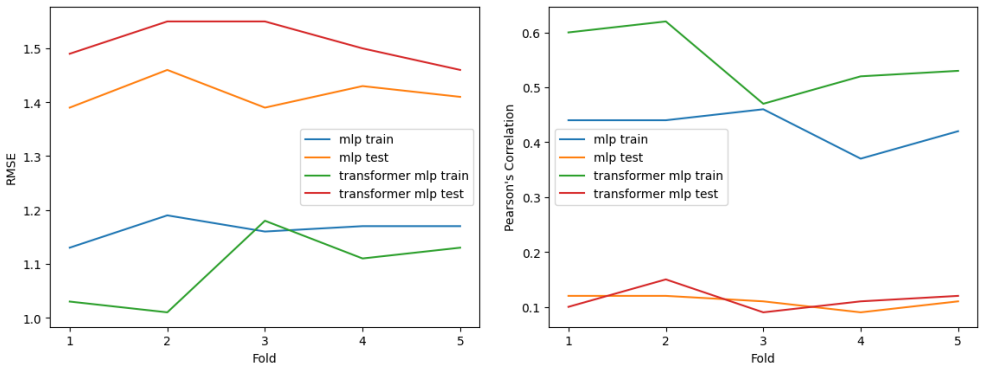

(a) (b)

**Fig. 3**: **Evaluating XGBoost Model** — Panel (a) illustrates RMSE values for the XGBoost algorithm across five cross-validation folds, with training RMSE remaining constant at 0.61 ± 0.00 and testing RMSE averaging 1.41 ± 0.03. Training RMSE also displays slight variability on folds 3 and 5. Panel (b) displays Pearson's Correlation Coefficients, with training correlations consistently high (r = 0.95 ± 0.00) and testing correlations averaging r = 0.13 ± across all folds. Each fold held out a unique 20% of the dataset for testing, while the remaining data was used for training.

folds 3 and 5 (Fig. 3a). In terms of Pearson's Correlation Coefficient, XGBoost performed significantly worse than the classical methods, achieving an average testing correlation r = 0.06 less than ridge regression, r = 0.05 less than lasso regression, and r = 0.05 less than RBF-SVM (Table 1). Unlike RBF-SVM, however, XGBoost maintained a consistent difference between training and testing RMSE (0.80 ± 0.03) indicating better overall generalization (Fig. 3a).

### 3.3 Multilayer Perceptron Results with Pre-processing Pipeline

The final two models evaluated against the proposed preprocessing pipeline were two different MLP learner architectures: a classical MLP and an MLP with an attention head (Materials & Methods). We refer to the MLP with an attention head as a transformer MLP. The classical MLP varied around its 1.41 ± 0.03 test RMSE mean throughout each fold, the average distance from the mean being 0.02 ± 0.02 (Fig. 4a, Table 1). RMSE shows no signs of overfitting with the classical MLP having an average distance of 0.25 ± 0.01 between the training and testing sets (Fig. 4a). The Pearson's Correlation Coefficient paints a different story. The training Pearson's Correlation Coefficient varies greatly across its mean (r = 0.48 ± 0.02), showing a large dip in performance on fold 4 (r = 0.37). This deteriorated performance is mirrored in its testing set on fold 4, having a Pearson's Correlation Coefficient r = 0.08 (Fig. 4b). This behavior may be occurring due to fold 4 holding a large amount of the input data's variance. It is worth noting that this performance deterioration is not seen in the




(a) (b)

**Fig. 4**: **Evaluating Multilayer Perceptron Models** — Panel (a) displays the RMSE values across each fold for the classical MLP and the transformer MLP architectures. The RMSE of both models displays a high degree of variability, with the classical MLP dipping in test RMSE performance on fold 3 and the transformer MLP spiking in performance on the same fold. Panel (b) summarizes the Pearson's Correlation Coefficients for the same two models, which remains comparable across all models for the testing set. The models varied widely in Pearson's Correlation Coefficient across the training sets.

classical MLP's RMSE, rather these values are greater than the mean RMSE across folds (training RMSE = 1.17; testing RMSE = 1.43) (Fig. 4b). This relationship could occur due to Pearson's Correlation Coefficient measuring the strength and direction between predicted and actual values while RMSE only measures the magnitude of the prediction errors. Similar to the other model's this may indicate a difference in data distribution across fold 4.

To confirm any potential differences in the data distributions and variance across the folds, we conduct a PCA and Kolmogorov-Smirnov (K-S) statistical examination on each of the 5 folds (Materials & Methods). Analysis of the training data revealed no distinct clusters or significant differences in spread between the folds (Fig. 5a). Each fold formed a single cohesive cluster, with no statistically significant difference in data distribution ($p > 0.05$) (Fig. 5a, Fig. 6b). These results indicate robust training folds across the k-fold cross-validation processes. In contrast, PCA analysis of the testing sets revealed some variability across folds, with distinct outliers observed in folds 3 and 5 (Fig. 6b), While each fold conformed to a single cluster, statistically significant differences in distribution were observed between all testing folds ($p < 0.05$). The largest magnitude of difference was observed between fold 1 and fold 5 (K-S statistic = 0.39) (Fig. 6a). Fold 4 had an average magnitude difference of $0.31 \pm 0.00$ across all other folds, with each comparison being statistically significant ($p < 0.05$) (Fig. 6b, Fig. 6a). The consistent magnitude difference may indicate why the RMSE of the classic MLP would stay consistent as the Pearson's Correlation Coefficient varies greatly on this fold.


(a) (b)

**Fig. 5**: **Principal Component Analysis of Input Data** — PCA was conducted across each fold of the five-fold cross-validation. Panel (a) illustrates the PCA analysis done on the training data set across each fold. The x-axis is the value of the first principal component and the y-axis is the second principal component. Training data clusters around (0, 0), with visible outliers extending toward (-20, 0) and (0, -20). Panel (b) shows PCA results for the testing data across folds, which also cluster near (0, 0). Outliers are sparser in the testing dataset compared to the training data and are more pronounced in folds 3 and 5.

The Transformer MLP exhibited unexpected behavior across k-fold cross-validation. The model outperformed the classical MLP on training RMSE across all folds, achieving an average RMSE of 1.51 $\pm$ 0.03 (Table 1). However, its training performance varied greatly with an average distance from the mean RMSE (RMSE = 1.09) of 0.06 – the greatest variance of all tested models (Fig. 4a). Notably, the model showed a spike in train RMSE performance on fold 3, despite PCA analysis identifying fold 3 as having the largest outliers. This behavior suggests that the transformer MLP may have overfit those points during training as folds 2 and 4 preformed closer to the mean. Interestingly, this overfitting did not negatively impact testing RMSE. On Pearson's Correlation Coefficient, the transformer MLP (r = 0.55 $\pm$ 0.05) outperforms the classical MLP (r = 0.48 $\pm$ 0.02) on the training (Table 1). Unexpectedly, however, the classical MLP achieved a higher average test Pearson's Correlation Coefficient (r = 0.14 $\pm$ 0.02) compared to the transformer MLP (r = 0.12 $\pm$ 0.02), outperforming it on folds 1 and 3 (Fig. 4b). The transformer MLP exhibited a dip in Pearson's Correlation Coefficient on fold 3 for both training and testing. Despite the training overfitting not affecting the testing RMSE, it seems to have negatively impacted the model's Pearson's Correlation Coefficient performance. These findings suggest further investigation into normalization techniques may be required to improve this model's ability to generalize.

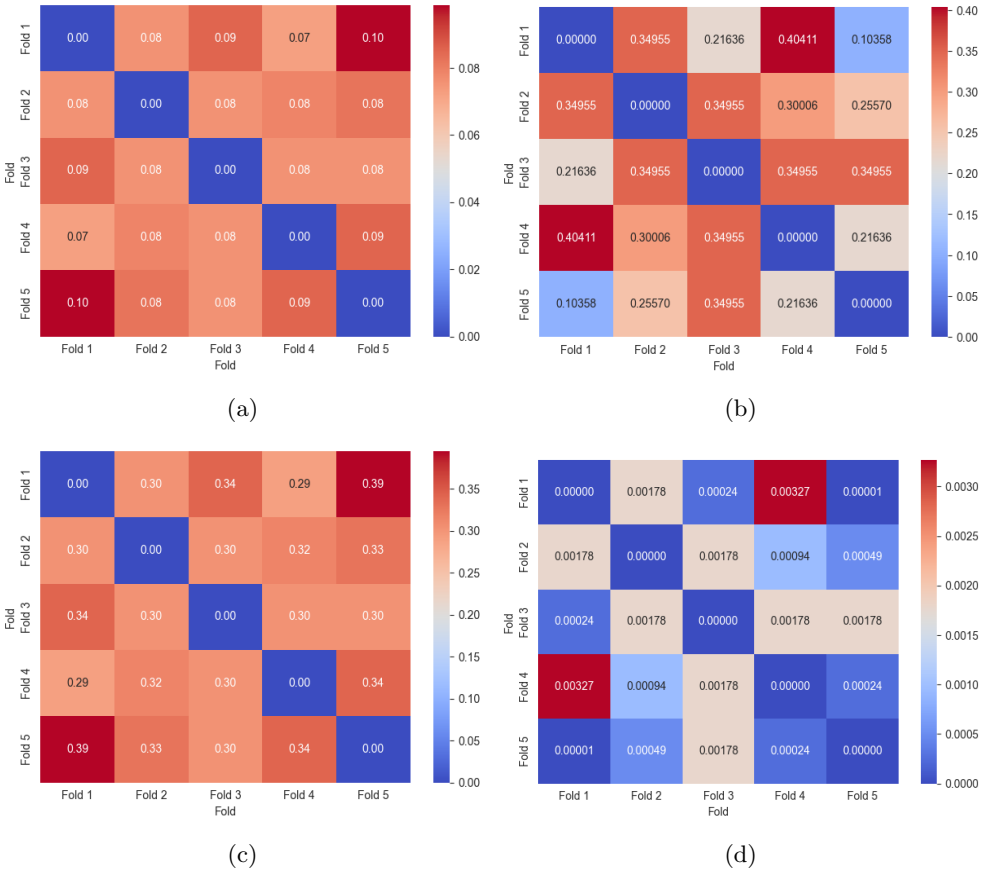



**Fig. 6**: **Kolmogorov-Smirnov Statistical Analysis Across Folds** — Heatmaps were created to summarize the results of K-S statistical analysis conducted across folds during k-fold cross-validation. Panel (a) displays the maximum K-S statistical values for features in the training set, indicating moderate differences between the distributions. Panel (b) shows the corresponding p-values, with all p-values exceeding p = 0.05, indicating that none of the differences are statistically significant (p ¿ 0.05). Panel (c) displays the maximum K-S statistic values for features in the test set, revealing differences of much larger magnitude. Panel (d) shows the corresponding p-values for these testing set K-S statistics which indicate statistically significant differences in feature distributions ($p < 0.05$).

# 4 Materials & Methods

## 4.1 The Cancer Genome Atlas Program (TCGA) Datasets

TCGA is a cancer genomics program focused on the molecular characterization of 33 cancer types. Samples in the program are composed of at least 60% tumor

nuclei with matching physical (normal tissue or blood samples) and genomic data. All TCGA RNA-seq data is processed via the Genomic Data Commons (GDC) hg38 harmonized pipeline (v28.0) with STAR alignment and HTSeq-count quantification. We validated the 60% tumor nuclei threshold via the 'TCGAbiolinks' (https://github.com/BioinformaticsFMRP/TCGAbiolinks) estimate purity function (v2.34.0). We retrived publicly available formalin paraffin-embedded (FFPE) WSI and matched transcriptomic profile data using 'TCGAbiolinks' across several cancer types. Training and testing of our model was conducted across the bladder carcinoma TCGA-BLCA cohort.

The TCGA-BLCA cohort details four total disease types: adenomas and adenocarcinomas, epithelial neoplasms (NOS), squamous cell neoplasms, transitional cell papillomas, and carcinomas [16]. All disease types were retrieved for use in this study. All WSI images with matched transcriptomic profiles (451) in this cohort were used for training and testing our models.

## 4.2 TCGAbiolinks

'TCGAbiolinks' (v2.34.0) is a R and Bioconductor package for downloading, preprocessing, and analyzing samples available through TCGA. We leveraged this toolkit to download and conduct preliminary analysis on the TCGA-BLCA cohort. The Genomic Data Commons (GDC) was initially queried for the transcriptome profiling and WSI individually. To locate matching data, the cases (or barcodes) of the WSI were compared with the submitter ID of the transcriptomic profiling. These intersecting cases were queried once more and downloaded for training and testing the models.

## 4.3 Pre-Processing of RNA-seq Data

Transcriptomics data from TCGA are outputs from the STAR aligner with TPM, FPKM, and FPKM-UQ normalized gene expression values. FPKM-UQ (Equation 1) normalized values were chosen due to their superior performance on datasets with large differences in read counts (such as that in TCGA-BLCA) and their overall robustness to technical noise.

$$FPKM - UQ = \frac{(\text{reads mapped to gene} * 10^9)}{(\text{Upper quartile reads} * \text{Gene length in bp})} \tag{1}$$

A log2 transform was further applied to normalize the magnitude of the FPKM-UQ values. We trained and tested our model across protein-encoding genes, miRNA, and lncRNA values.

## 4.4 CLAM Pre-Processing of WSI Slides

WSI from TCGA are given as SVS files of a range from 500 MB to 2.00 GB in size. TCGA has a default down sampling value of 20x. We implemented the CLAM (https://github.com/mahmoodlab/CLAM) pipeline to handle masking out the background (including holes in the tissue) and segmenting the tissue into 224px-by-224px

patches. The segmentation level ('seg-level') was set to use the default WSI down sampling value of 20x. CLAM outputs an array for each WSI slide containing coordinates for each 224px-by-224px patch, which can then be used for feature extraction by UNI (https://github.com/mahmoodlab/UNI).

## 4.5 UNI Feature Extraction

UNI is a pretrained vision encoder developed for computational pathology tasks, particularly feature extraction and classification of histopathology images. UNI was trained across 34 cancer and clinical tasks (such as bladder carcinomas), showing great generalizability for rare and underrepresented cases. We leverage UNI's predefined pipeline with the CLAM preprocessing model for our feature extraction. Feature vectors outputted by the UNI model are then outputted into H5 files and can be used as input vectors for downstream models.

## 4.6 Ridge Regression Architecture

Ridge Regression is a type of linear regression that adds a L2-regularization term to the ordinary least squares (OLS) loss function (Equation 2). This term ($j$) shrinks the coefficients ($\lambda$) toward zero to prevent overfitting when predictors are highly correlated.

$$O = \sum_{i=0}^{n} (y_i - \hat{y}_i)^2 + \lambda \sum j = 1^p \beta_j^2 \tag{2}$$

We hypothesis that despite the inherent complexity of the input data, the simplistic approach will stave off overfitting and help deal with potential multicollinearity. Additionally, we decided on this model due to its simplicity and ease of implementation. Due to the time constraints imposed by the attempted implementation of the SEQUOIA model, this would allow us to prove that our introduced feature extraction method could function on the feature data for prediction.

The model was trained using 5-fold cross-validation, each fold holding out 20% of the data for testing and utilizing the rest for training. The model leveraged principal component analysis (PCA) with 100 components to lower the dimensionality of the feature space on every fold. This was done to help the model generalize better to the input feature space.

The model was completely implemented using the 'sklearn' (v1.6.1) Python (v3.10.0) library. A single A100 GPU was utilized for training and testing.

## 4.7 LASSO Regression Architecture

Lasso Regression is a type of linear regression that implements a L1-regularization term to the ordinary least squares (OLS) loss function (Equation 3). Rather than only shrinking the coefficients towards zero, the penalty will remove insignificant predictors from the model. This enables us to use the tool as a powerful feature selector.

$$O = \sum_{i=0}^{n} (y_i - \hat{y}_i)^2 + \lambda \sum j = 1^p |\beta_j| \tag{3}$$

Once more, we hypothesis that the simplistic approach will stave off overfitting and provide a powerful enough tool to prove our pre-processing method on this task. Time constraints imposed by the attempted implementation of SEQUOIA influenced our decision making as well.

The model was trained using 5-fold cross-validation with 80% of the data used for training and 20% used for testing. PCA was also leveraged with 100 components to help reduce the dimensionality of the feature space per fold.

The model was completely implemented using the 'sklearn' (v1.6.1) Python (v3.10.0) library. A single A100 GPU was utilized for training and testing.

## 4.8 RBF-SVM Architecture

Support Vector Machines (SVMs) are a supervised machine learning algorithm effective in classification and regression tasks on high-dimensionality datasets. Radial Basis Function SVM (RBF-SVM) is a type of SVM that utilizes the RBF kernel to map the data into a higher-dimension where it may be linearly separable (Equation 4).

$$K(x, x) = e^{-\gamma |x - x|^2} \tag{4}$$

The model was trained using 5-fold cross-validation with 80% of the data used for training and 20% used for testing. A grid search was performed per fold to optimize the model's hyperparameters. Each grid search utilized 20% of the training data for validation to find the model with the best R2 value on the validation set were chosen.

With the added complexity, we hypothesis that this algorithm may perform slightly better than that of Ridge and Lasso Regression with better stability. However, due to the inherent complexity of the data, this effect will most likely be minimal.

The model was completely implemented using the 'sklearn' (v1.6.1) Python (v3.10.0) library. A single A100 GPU was utilized for training and testing.

## 4.9 XGBoost Architecture

Gradient Boosting is a foundational ensemble learning method implementing sequential weak learners with their predictions being aggregated for a final prediction. Each subsequent weak learner is trained to predict the errors of its predecessor, updating its weights by computing the negative gradient of the loss function. Mean squared error (MSE) is commonly used as a loss function criterion (Equation 5).

$$L = \frac{1}{2n} \sum_{i=0}^{n} (y_i - \hat{y}_i)^2 \tag{5}$$

We chose to implement XGBoost, an extension of the Gradient Boosting algorithm that implements an L1 and L2 regularization term to the loss function to penalize model complexity (Equation 6). We implemented the MSE function (Equation 5) for the training process of the algorithm. A grid-search utilizing a MSE function was

performed to optimize the model's hyperparameters per fold. Each grid search would train a model on 80% of the training set and leave 20% of the training-set for validation. The model with the best MSE were chosen for continued use.

$$\hat{y}_t^n = \sum_{i=1}^{t} f_i(\overrightarrow{f} v_t^n) \tag{6}$$

Implementation of the XGBoost algorithm was accomplished via the gpu-enabled 'xgboost' (v3.0.0) Python (v3.10.0) library. The grid search was implemented with the 'sklearn' (V1.6.1) library for Python (v3.10.0). Two A100 GPUs were utilized for training and testing the model.

## 4.10 Multilayer Perceptron Architecture

Multilayer Perceptron (MLP) learners are a type of feedforward neural network consisting of multiple interconnected layers made up of neurons. We chose to implement two different MLP learners: a classical version and a transformer version. The classical MLP implements 12 hidden layers with each layer separated by a batch normalization layer. The hidden layers of the model implemented a total of 27,648 neurons with each utilizing the ReLU activation function. The final output layer implemented a linear activation function. A learning rate of 0.01 was employed with a mean squared error (MSE) loss function.

The transformer version of the MLP implemented a multi-headed attention layer with a total of 4 heads. The input layer was a dense layer followed by a layer normalization and a 20% drop-out layer. There was a single multi-headed attention layer followed by a standard MLP learner with 9 hidden layers. Each layer was separated by a batch normalization layer and used the ReLU activation function. The final output layer, once more, implemented a linear activation function. A learning rate of 0.001 was used across this model with a MSE loss function.

Both models were implemented using the 'tensorflow' (v2.16.1) Python (v3.10.0) library. Cross-validation and standardization was done via the 'sklearn' (v1.6.1) library for Python (v3.10.0) and the 'scipy' (v1.15.2) library was used for statistics calculation. Two A100 GPUs were utilized for training and testing both MLP learners.

## 4.11 Kolmogorov-Smirnov Analysis

Kolmogorov-Smirnov (K-S) analysis is a non-parametric statistical test used to compare the probability distribution between two different distributions. In this study, we implement it to analyze the difference between two dataset distributions. The null hypothesis ($H_0$) of the test is that the distributions being compared are identical. The K-S test evaluates the maximum difference between two cumulative distributions, calculating a statistic representing the magnitude of the difference (Equation 7).

$$D = \sup_x |F_1(x) - F_2(x)| \tag{7}$$

Where $F_1(x)$ and $F_2(x)$ are cumulative distribution functions. $D$ represents the magnitude of the distance between the two distributions over all possible values of $x$.

The p-value is measured alongside the K-S statistic, determining if the magnitude is significant or not.

### 4.12 Training and Testing on TCGA Dataset

Training and testing on the TCGA-BLCA dataset was conducted using five-fold cross-validation. Each fold's data was divided into 80% training and 20% testing, with 20% of the training data being used as validation during parameter grid searchs. Cross-validation was implemented via the 'sklearn' (V1.6.1) library for Python (v3.10.0). Each fold calculated the RMSE and the average Pearson's Correlation for each gene prediction within the fold. We used RMSE to help grade the prediction quality of the fold. The Pearson's Correlations were extracted, with the mean helping us identify well-predicted folds within the model. RMSE was implemented via the 'sklearn' (v1.6.1) library while Pearson's Correlation was implemented using the 'scipy' (v1.15.2) for Python (v3.10.0).

## 5 Discussion

We introduce a novel pre-processing approach to predicting transcriptomics data from WSI, leveraging the state-of-the-art CLAM and UNI models to extract meaningful features from WSIs. We tested across three classical machine learning models - ridge regression, lasso regression, and RBF-SVM – and observed results consistent with expectations for high-dimensional data. Linear models often struggle to identify meaningful relationships in such a setting; however, ridge regression and lasso regression benefited from PCA to reduce feature space dimensionality. Ridge regression showed the highest test Pearson's Correlation Coefficient ($r = 0.19 \pm 0.01$) across all folds among the classical models (Fig 2b). In contrast RBF-SVM implemented a kernel-based approach to project features into a higher-dimensional space where the features may be linearly separable. RBF-SVM preformed the best on the testing RMSE across all folds, reaching an average RMSE of $1.29 \pm 0.02$ (Fig. 2a, Table 1). All three of these methods out preformed XGBoost and the base MLP implementation in testing Pearson's Correlation Coefficient. Ridge regression achieved an average testing correlation $r = 0.06$ greater than XGBoost, $r = 0.05$ greater than the classical MLP, and $r = 0.07$ greater than the transformer MLP (Table 1). Lasso regression and RBF-SVM achieved $r = 0.01$ less than ridge regression in testing Pearson's Correlation Coefficient (Table 1). The stable performance of ridge regression and lasso regression across folds suggests their suitability for scalable clinical workflows where computational simplicity is prioritized. Furthermore, K-S statistical analysis revealed significant differences in testing set distribution across folds, reflecting the inherent heterogeneity within the dataset (Fig. 6). Despite this variability, ridge regression and lasso regression maintained stable RMSE performance, highlighting their robustness in handling moderate feature complexity.

We also tested our pre-processing pipeline with XGBoost, hypothesizing that it would excel on the tabular nature of the data extracted from WSIs despite their inherent complexity. While XGBoost did demonstrate consistent testing RMSE ($1.41 \pm 0.03$) performance, it underperformed compared to the classical models on testing

Pearson's Correlation Coefficient (r = 0.13) (Table 1). The model struggled to generalize correlations to unseen data despite hyperparameter optimization efforts (Fig. 3b). This limitation suggests the need for an objective function that implements correlation directly into the model's training processes. Such an objective function would encourage the model to optimize for the magnitude and direction of errors, potentially improving its ability to capture meaningful relationships in testing data. Nevertheless, XGBoost outperformed the transformer MLP on testing Pearson's Correlation Coefficient (r = 0.01 higher). It is worth noting that the model exhibited minimal variability in performance metrics across folds, despite the high variability found per fold (Fig. 3a, Fig. 3b). This suggests XGBoost has the potential for stabilizing downstream predictions when working with WSI pre-processing workflows.

The final two models we evaluated were two different implementations of a MLP learner: a classical MLP and a transformer MLP. The performance from both implementations were unexpected, with both being outperformed by the other classical and ensemble methods implemented. The classical MLP had a variable test RMSE, with each fold fluctuating around the mean test RMSE ($1.41 \pm 0.03$) at an average distance of $0.02 \pm 0.02$ (Fig. 4a, Table 1). Despite the input data being scaled to approximate a normal distribution, fluctuations in test RMSE may be attributed to differences in variability and data distributions across testing folds (Fig. 5, Fig. 6). This suggests that the feature space from the pre-processing pipeline may benefit from a different scaling approach. The classical MLP preformed similarly to the XGBoost algorithm in average test RMSE ($1.41 \pm 0.03$), under performing compared to the transformer MLP architecture by $0.10 \pm 0.00$ (Table 1). The model outperformed the transformer MLP in test Pearson's Correlation Coefficient by 0.02, suggesting a greater ability to generalize to the correlation of the testing set. The transformer MLP outperformed all other models in test RMSE, achieving an average RMSE of $1.51 \pm 0.03$ (Table 1). The model fluctuated in Pearson's Correlation Coefficient on both the test and training sets across folds, indicating a high sensitivity to the data distribution differences and variability in the sets (Fig. 5, Fig. 6). The observed decrease in training Pearson's Correlation Coefficient may reflect its inability to consistently capture both magnitude and directionality of the residuals (Fig. 4b). While RMSE measures only error magnitude, Pearson's Correlation considers both magnitude and directionality; therefore, similar magnitudes but varying directions of errors could result in stable RMSE, but lower correlation values. These findings support our recommendation for implementing a combined object function that optimizes for both error magnitude and directionality during training.

The novelty of our approach lies in the integration of the CLAM segmentation method during pre-processing and feature extraction of morphological features from WSIs. Slides from the TCGA-BLCA dataset, paired with matched transcriptomic features, were segmented using CLAM's transformer. Each patch's coordinates were stored for subsequent processing by the UNI pipeline. By incorporating CLAM into the pre-processing phase, we aimed to enhance a model's generalization capabilities for transcriptomic profiling. Initially, we planned to implement the SEQUOIA model to evaluate our pre-processing pipeline and establish a baseline for comparison. However, the publicly available SEQUOIA model had technical issues preventing it from

functioning correctly on the data types listed in their manuscripts. Due to time limitations, to address this we developed and implemented alternative approaches to assess our pre-processing pipeline. Future work will aim to validate the findings from the SEQUOIA manuscript while comparing our pre-processing pipeline to alternative methodologies. Across the XGBoost and MLP methods, we found that while test RMSE was maximized, the test Pearson's Correlation Coefficient was lower compared to the classical machine learning methods. Studies in other fields have shown success in implementing a hybrid RMSE and Pearson's Correlation Coefficient object functions in training models ([17–19]). This approach warrants further investigation in the context of transcriptomic data prediction from WSIs.

# 6 Conclusion

We developed a novel use-case for the CLAM and UNI pipelines to assist in predicting transcriptomic data from WSIs. This approach aimed to improve model performance and generalizability to morphological features extracted from WSIs. The approach was tested across ridge regression, lasso regression, RBF-SVM, XGBoost, and two multilayer perceptron (MLP) methods. Ridge regression preforming the best on average test Pearson's Correlation Coefficient (r = 0.19 +- 0.01) while the transformer MLP preforming the best on average test RMSE (RMSE = 1.51 +- 0.03) (Table 1). These results reveal that the introduced approach increases model performance on the feature space even in models constrained by computational power. PCA and K-S statistical analysis revealed significant variability in testing set distributions across folds, despite training set distributions remaining consistent (Fig. 5, Fig. 6). These findings highlight the inherent challenges related to data heterogeneity in transcriptomic prediction tasks using WSIs. We recommend future work investigate alternative normalization and scaling schemes to address these distributional differences. Additionally, we recommend exploring hybrid objective functions that combine RMSE with Pearson's Correlation Coefficient to optimize both residual magnitude and relational accuracy. Such an approach may enable models to better capture meaningful relationships within transcriptomic data while maintaining robust predictive performance across folds.

# 7 Code & Data Availability

Custom code was developed as part of the analysis reported here, and has been deposited on GitHub: https://github.com/magmalamp/ShayReardon-CIS6930Project. Data used in the analysis reported is publicly available through the Genomic Data Commons (GDC) data portal: https://portal.gdc.cancer.gov/projects/TCGA-BLCA.

# References

[1] Hausser, J., Alon, U.: Tumour heterogeneity and the evolutionary trade-offs of cancer **20**(4), 247–257 https://doi.org/10.1038/s41568-020-0241-6

[2] Collisson, E.A., Campbell, J.D., Brooks, A.N., Berger, A.H., Lee, W., Chmielecki, J., Beer, D.G., Cope, L., Creighton, C.J., Danilova, L., Ding, L., Getz, G., Hammerman, P.S., Neil Hayes, D., Hernandez, B., Herman, J.G., Heymach, J.V., Jurisica, I., Kucherlapati, R., Kwiatkowski, D., Ladanyi, M., Robertson, G., Schultz, N., Shen, R., Sinha, R., Sougnez, C., Tsao, M.-S., Travis, W.D., Weinstein, J.N., Wigle, D.A., Wilkerson, M.D., Chu, A., Cherniack, A.D., Hadjipanayis, A., Rosenberg, M., Weisenberger, D.J., Laird, P.W., Radenbaugh, A., Ma, S., Stuart, J.M., Averett Byers, L., Baylin, S.B., Govindan, R., Meyerson, M., Rosenberg, M., Gabriel, S.B., Cibulskis, K., Sougnez, C., Kim, J., Stewart, C., Lichtenstein, L., Lander, E.S., Lawrence, M.S., Getz, G., Kandoth, C., Fulton, R., Fulton, L.L., McLellan, M.D., Wilson, R.K., Ye, K., Fronick, C.C., Maher, C.A., Miller, C.A., Wendl, M.C., Cabanski, C., Ding, L., Mardis, E., Govindan, R., Creighton, C.J., Wheeler, D., Balasundaram, M., Butterfield, Y.S.N., Carlsen, R., Chu, A., Chuah, E., Dhalla, N., Guin, R., Hirst, C., Lee, D., Li, H.I., Mayo, M., Moore, R.A., Mungall, A.J., Schein, J.E., Sipahimalani, P., Tam, A., Varhol, R., Gordon Robertson, A., Wye, N., Thiessen, N., Holt, R.A., Jones, S.J.M., Marra, M.A., Campbell, J.D., Brooks, A.N., Chmielecki, J., Imielinski, M., Onofrio, R.C., Hodis, E., Zack, T., Sougnez, C., Helman, E., Sekhar Pedamallu, C., Mesirov, J., Cherniack, A.D., Saksena, G., Schumacher, S.E., Carter, S.L., Hernandez, B., Garraway, L., Beroukhim, R., Gabriel, S.B., Getz, G., Meyerson, M., Hadjipanayis, A., Lee, S., Mahadeshwar, H.S., Pantazi, A., Protopopov, A., Ren, X., Seth, S., Song, X., Tang, J., Yang, L., Zhang, J., Chen, P.-C., Parfenov, M., Wei Xu, A., Santoso, N., Chin, L., Park, P.J., Kucherlapati, R., Hoadley, K.A., Todd Auman, J., Meng, S., Shi, Y., Buda, E., Waring, S., Veluvolu, U., Tan, D., Mieczkowski, P.A., Jones, C.D., Simons, J.V., Soloway, M.G., Bodenheimer, T., Jefferys, S.R., Roach, J., Hoyle, A.P., Wu, J., Balu, S., Singh, D., Prins, J.F., Marron, J.S., Parker, J.S., Neil Hayes, D., Perou, C.M., Liu, J., Cope, L., Danilova, L., Weisenberger, D.J., Maglinte, D.T., Lai, P.H., Bootwalla, M.S., Van Den Berg, D.J., Triche Jr, T., Baylin, S.B., Laird, P.W., Rosenberg, M., Chin, L., Zhang, J., Cho, J., DiCara, D., Heiman, D., Lin, P., Mallard, W., Voet, D., Zhang, H., Zou, L., Noble, M.S., Lawrence, M.S., Saksena, G., Gehlenborg, N., Thorvaldsdottir, H., Mesirov, J., Nazaire, M.-D., Robinson, J., Getz, G., Lee, W., Arman Aksoy, B., Ciriello, G., Taylor, B.S., Dresdner, G., Gao, J., Gross, B., Seshan, V.E., Ladanyi, M., Reva, B., Sinha, R., Onur Sumer, S., Weinhold, N., Schultz, N., Shen, R., Sander, C., Ng, S., Ma, S., Zhu, J., Radenbaugh, A., Stuart, J.M., Benz, C.C., Yau, C., Haussler, D., Spellman, P.T., Wilkerson, M.D., Parker, J.S., Hoadley, K.A., Kimes, P.K., Neil Hayes, D., Perou, C.M., Broom, B.M., Wang, J., Lu, Y., Kwok Shing Ng, P., Diao, L., Averett Byers, L., Liu, W., Heymach, J.V., Amos, C.I., Weinstein, J.N., Akbani, R., Mills, G.B., Curley, E., Paulauskis, J., Lau, K., Morris, S., Shelton, T., Mallery, D., Gardner, J., Penny, R., Saller, C., Tarvin, K., Richards, W.G., Cerfolio, R., Bryant, A., Raymond, D.P., Pennell, N.A., Farver, C., Czerwinski, C., Huelsenbeck-Dill, L., Iacocca, M., Petrelli, N., Rabeno, B., Brown, J., Bauer, T., Dolzhanskiy, O., Potapova, O., Rotin, D., Voronina, O., Nemirovich-Danchenko, E., Fedosenko, K.V., Gal, A., Behera, M., Ramalingam, S.S., Sica, G., Flieder, D., Boyd, J., Weaver, J., Kohl, B., Huy

Quoc Thinh, D., Sandusky, G., Juhl, H., The Cancer Genome Atlas Research Network, Disease analysis working group, Genome sequencing centres: The Eli & Edythe L. Broad Institute, Washington University in St. Louis, Baylor College of Medicine, Canada's Michael Smith Genome Sciences Centre, B.C.C.A., The Eli & Edythe L. Broad Institute, Harvard Medical School/Brigham & Women's Hospital/MD Anderson Cancer Center, North Carolina, C.H., University of Kentucky, The USC/JHU Epigenome Characterization Center, Genome data analysis centres: The Eli & Edythe L. Broad Institute, Memorial Sloan-Kettering Cancer Center, California, S.C.I., Oregon Health & Sciences University, The University of Texas MD Anderson Cancer Center, Biospecimen core resource: International Genomics Consortium, Analytical Biological Service, I., Brigham & Women's Hospital, University of Alabama at Birmingham, Cleveland Clinic, Christiana Care, Cureline, Emory University, Fox Chase Cancer Center, ILSbio, Indiana University, Indivumed, John Flynn Hospital: Comprehensive molecular profiling of lung adenocarcinoma **511**(7511), 543–550 https://doi.org/10.1038/nature13385 . Publisher: Nature Publishing Group. Accessed 2025-03-30

[3] Cajal, S., Sesé, M., Capdevila, C., Aasen, T., De Mattos-Arruda, L., Diaz-Cano, S.J., Hernández-Losa, J., Castellví, J.: Clinical implications of intratumor heterogeneity: challenges and opportunities **98**(2), 161–177 https://doi.org/10.1007/s00109-020-01874-2 . Accessed 2025-03-30

[4] Buikhuisen, J.Y., Torang, A., Medema, J.P.: Exploring and modelling colon cancer inter-tumour heterogeneity: opportunities and challenges **9**(7), 1–15 https://doi.org/10.1038/s41389-020-00250-6 . Publisher: Nature Publishing Group. Accessed 2025-03-30

[5] Bedard, P.L., Hansen, A.R., Ratain, M.J., Siu, L.L.: Tumour heterogeneity in the clinic **501**(7467), 355–364 https://doi.org/10.1038/nature12627 . Publisher: Nature Publishing Group. Accessed 2025-02-19

[6] Safri, F., Nguyen, R., Zerehpooshnesfchi, S., George, J., Qiao, L.: Heterogeneity of hepatocellular carcinoma: from mechanisms to clinical implications **31**(8), 1105–1112 https://doi.org/10.1038/s41417-024-00764-w . Publisher: Nature Publishing Group. Accessed 2025-02-19

[7] Peng, L., Bian, X.W., Li, D.K., Xu, C., Wang, G.M., Xia, Q.Y., Xiong, Q.: Large-scale RNA-seq transcriptome analysis of 4043 cancers and 548 normal tissue controls across 12 TCGA cancer types **5**(1), 13413 https://doi.org/10.1038/srep13413 . Publisher: Nature Publishing Group. Accessed 2025-03-30

[8] Wang, Y., Mashock, M., Tong, Z., Mu, X., Chen, H., Zhou, X., Zhang, H., Zhao, G., Liu, B., Li, X.: Changing technologies of RNA sequencing and their applications in clinical oncology **10** https://doi.org/10.3389/fonc.2020.00447 . Publisher: Frontiers. Accessed 2025-03-30

[9] Genomics: Visium Spatial Gene Expression

[10] Wang, X., Allen, W.E., Wright, M.A., Sylwestrak, E.L., Samusik, N., Vesuna, S., Evans, K., Liu, C., Ramakrishnan, C., Liu, J., Nolan, G.P., Bava, F.-A., Deisseroth, K.: Three-dimensional intact-tissue sequencing of single-cell transcriptional states **361**(6400), 5691 https://doi.org/10.1126/science.aat5691 . Publisher: American Association for the Advancement of Science. Accessed 2025-03-30

[11] Chen, Y.-C.D., Chen, Y.-C., Rajesh, R., Shoji, N., Jacy, M., Lacin, H., Erclik, T., Desplan, C.: Using single-cell RNA sequencing to generate predictive cell-type-specific split-GAL4 reagents throughout development **120**(32), 2307451120 https://doi.org/10.1073/pnas.2307451120 . Publisher: Proceedings of the National Academy of Sciences. Accessed 2025-02-19

[12] Lu, M.Y., Williamson, D.F.K., Chen, T.Y., Chen, R.J., Barbieri, M., Mahmood, F.: Data-efficient and weakly supervised computational pathology on whole-slide images **5**(6), 555–570 https://doi.org/10.1038/s41551-020-00682-w . Publisher: Nature Publishing Group. Accessed 2025-02-13

[13] Chen, R.J., Ding, T., Lu, M.Y., Williamson, D.F.K., Jaume, G., Song, A.H., Chen, B., Zhang, A., Shao, D., Shaban, M., Williams, M., Oldenburg, L., Weishaupt, L.L., Wang, J.J., Vaidya, A., Le, L.P., Gerber, G., Sahai, S., Williams, W., Mahmood, F.: Towards a general-purpose foundation model for computational pathology **30**(3), 850–862 https://doi.org/10.1038/s41591-024-02857-3 . Publisher: Nature Publishing Group. Accessed 2025-03-30

[14] Schmauch, B., Romagnoni, A., Pronier, E., Saillard, C., Maillé, P., Calderaro, J., Kamoun, A., Sefta, M., Toldo, S., Zaslavskiy, M., Clozel, T., Moarii, M., Courtiol, P., Wainrib, G.: A deep learning model to predict RNA-seq expression of tumours from whole slide images **11**(1), 3877 https://doi.org/10.1038/s41467-020-17678-4 . Publisher: Nature Publishing Group. Accessed 2025-03-30

[15] Pizurica, M., Zheng, Y., Carrillo-Perez, F., Noor, H., Yao, W., Wohlfart, C., Vladimirova, A., Marchal, K., Gevaert, O.: Digital profiling of gene expression from histology images with linearized attention **15**(1), 9886 https://doi.org/10.1038/s41467-024-54182-5 . Publisher: Nature Publishing Group. Accessed 2025-02-13

[16] TCGA-BLCA. https://www.cancerimagingarchive.net/collection/tcga-blca/ Accessed 2025-03-12

[17] Haut, N., Banzhaf, W., Punch, B.: Correlation versus RMSE Loss Functions in Symbolic Regression Tasks. arXiv. https://doi.org/10.48550/arXiv.2205.15990 . http://arxiv.org/abs/2205.15990 Accessed 2025-04-02

[18] Waldmann, P.: On the use of the pearson correlation coefficient for model evaluation in genome-wide prediction **10**, 899 https://doi.org/10.3389/fgene.2019.00899 . Accessed 2025-04-02

[19] Shahani, N.M., Zheng, X., Wei, X., Hongwei, J.: Hybrid machine learning approach for accurate prediction of the drilling rate index **14**(1), 24080 https://doi.org/10.1038/s41598-024-75639-z . Publisher: Nature Publishing Group. Accessed 2025-04-02