SEPAL: Spatial Gene Expression Prediction from Local Graphs

Gabriel Mejia, Paula Cárdenas*, Daniela Ruiz*, Angela Castillo, Pablo Arbeláez
Center for Research and Formation in Artificial Intelligence
Universidad de los Andes, Bogotá, Colombia
{gm.mejia,p.cardenasg,da.ruizl1,a.castillo13,pa.arbelaez}@uniandes.edu.co

Abstract

Spatial transcriptomics is an emerging technology that aligns histopathology images with spatially resolved gene expression profiling. It holds the potential for understanding many diseases but faces significant bottlenecks such as specialized equipment and domain expertise. In this work, we present SEPAL, a new model for predicting genetic profiles from visual tissue appearance. Our method exploits the biological biases of the problem by directly supervising relative differences with respect to mean expression, and leverages local visual context at every coordinate to make predictions using a graph neural network. This approach closes the gap between complete locality and complete globality in current methods. In addition, we propose a novel benchmark that aims to better define the task by following current best practices in transcriptomics and restricting the prediction variables to only those with clear spatial patterns. Our extensive evaluation in two different human breast cancer datasets indicates that SEPAL outperforms previous state-of-the-art methods and other mechanisms of including spatial context.

1. Introduction

Histopathology is the study of diseases in tissues through microscopic sample examination. Among the different staining methods, Hematoxylin and Eosin (H&E) is the most common one and is currently considered the gold standard for diagnosing a wide range of diseases [31, 28, 24]. More recently, this approach has been complemented with molecular biomarkers, such as mRNA expression profiling, offering high specificity and the ability to directly predict prognosis and determine treatments [29, 5]. Interestingly, these two data types prove complementary: while H&E imaging lacks the specificity of transcriptomics, gene profiling lacks the physiological insights derived from morphology. By aligning dense spatial mRNA profiling with H&E histopathological images, Spatial Transcriptomics technologies (ST) provide comprehensive insights into the spatial organization of gene expression within tissues [3].

The advent of direct gene expression assessment on tissue harbors the potential for an unprecedented understanding of the mechanistic causes behind many diseases. However, obtaining these datasets in real clinical practice encounters major bottlenecks, primarily stemming from the need for specialized equipment, domain expertise, and considerable time requirements [36]. To overcome these burdens and leverage the fact that H&E images are ubiquitous in medical settings, the computer vision community has recently delved into predicting gene expression from tissue images. Although various works demonstrate promising results [23, 21, 2, 12, 34, 35], existing methods are still far from clinical deployment.

Upon closer examination of the problem, it becomes evident that changes in gene expression are typically associated with alterations in tissue appearance. However, it is important to note that this correlation does not necessarily apply to all genes. For example, constitutive genes that exhibit constant expression within the spatial context [7] are unsuitable for prediction based solely on visual information. Hence, methods should focus on genes with a verifiable dependence on tissue appearance for the task to be well defined. Another challenge lies in the scarcity of data. The current publicly available datasets encompass $2-70$ Whole Slide Images (WSI) with $5,000-15,000$ genes for a set of $300-3500$ coordinates, depending on the technology [3]. Consequently, generating such high-dimensional predictions with such limited samples is intrinsically difficult. Finally, as the technology is still in development, ground-truth data is sparse and noisy; specifically, pepper noise is observed in the expression maps [36].

Current approaches present a dichotomy between complete globality, which uses the WSI to jointly predict an expression map for all the coordinates at once (WSI-based methods [23, 21, 2], Fig.1.A), and complete locality, which only uses visual information available at each coordinate to predict gene expression (patch-based methods [12, 34, 35],

*These authors contributed equally to this work

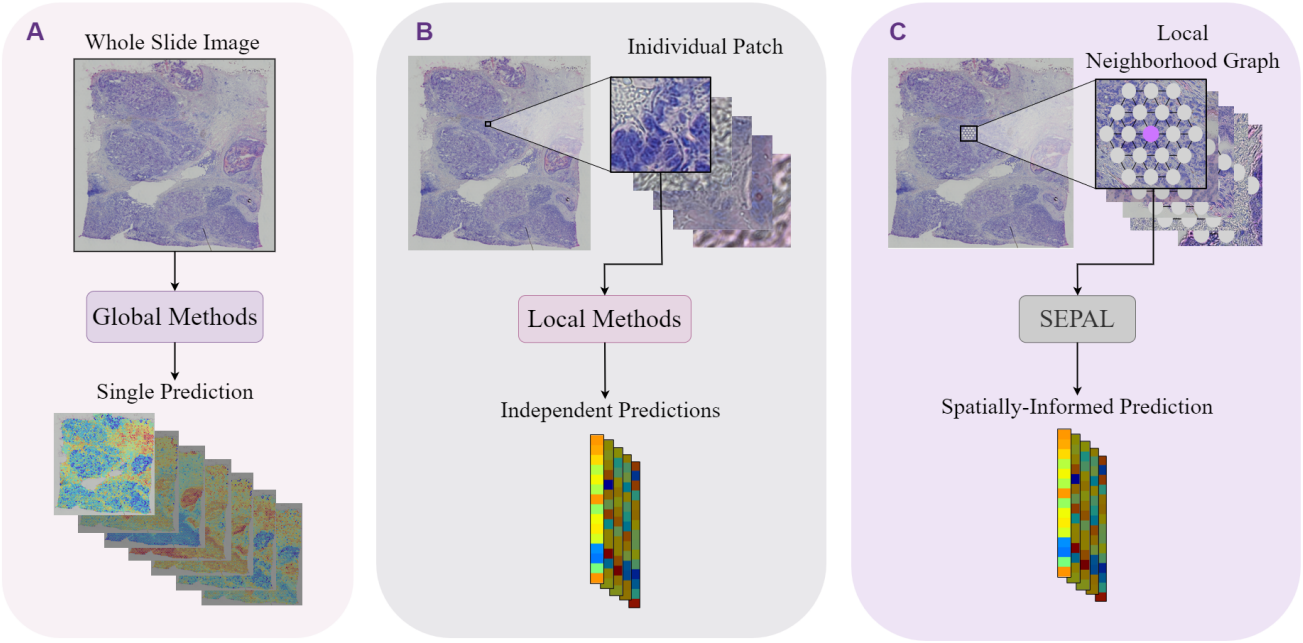

Figure 1. Different approaches for predicting gene expression from tissue images. The inputs of each model type are enclosed by a black frame. (A) *Global methods* analyze a whole slide image and make a prediction about the tissue expression in every spot at once. (B) *Local methods* process the image by patches and predict the expression of each individual patch, one at a time. (C) SEPAL uses graphs that contain information from multiple patches to represent spatial information and predict gene expression for the central node of each graph.

Fig.1.B). While complete globality leverages spatial information and long-range interactions, it suffers from severe data scarcity, making models prone to overfitting. In contrast, complete locality benefits from abundant data for deep learning training but disregards spatial relations, resulting in suboptimal performance.

To overcome these challenges, we propose a new problem formulation, benchmark, and state-of-the-art method for **S**patial **E**xpression **P**rediction by **A**nalysing **L**ocal graphs (**SEPAL**). Our problem formulation strategically exploits the biological nature of the problem; our benchmark uses a robust bioinformatic pipeline to overcome acquisition issues; and our model bridges the gap between locality and globality by performing local spatial analysis.

In terms of problem formulation, we leverage a domain-specific advantage: the expression of a gene is expected to be within a specific range of values, and the variations inside that range are the ones with physiological significance. Rather than solely focusing on the absolute value of gene expression, we exploit this knowledge by bounding the prediction space within a defined box and using its center as an inductive bias. By estimating this bias from the training data, we can focus on learning relative differences instead of absolute values. Specifically, we supervise expression changes with respect to the mean expression of each gene in the training dataset. This novel approach differs from previous works since they directly predict the absolute gene expression.

We build our benchmark by first incorporating standard bioinformatic processing normalizations (TPM [1]), which were previously lacking. Then, we apply a modified version of an adaptive median filter [6] to manage the pepper noise. And finally, to ensure the selection of relevant prediction genes, we filter by Moran's I [20] value, a statistic designed to identify significant spatial patterns over a graph. By leveraging Moran's I, we ensure our focus remains on genes that depend on tissue appearance.

Lastly, we introduce a novel approach that harnesses the power of local spatial analysis. Our strategy starts with a completely local learning stage and then integrates information from local neighborhoods surrounding each patch with the help of a graph neural network (GNN, see Fig.1.C). Our key hypothesis is that gene expression is predominantly influenced by nearby visual characteristics rather than long-range interactions. SEPAL benefits from the advantages of local-based and global-based training (spatial relations and enough data) without succumbing to their respective limitations. We conduct extensive experimentation on two different human breast cancer datasets obtained with different technologies and report favorable results relative to existing techniques.

Our contributions can be summarized as follows:

- We propose a paradigm shift to supervise gene expression changes relative to the mean rather than absolute

values.

- We propose a benchmark that follows current best practices in transcriptomics, deals with pepper noise, and restricts prediction genes to only those with clear spatial patterns.
- We develop a new state-of-the-art method that applies local spatial analysis via graph neural networks.

To promote further research on ST, our project's benchmark and source code is publicly available at https://github.com/BCV-Uniandes/SEPAL.

2. Related Work

Multiple approaches have been proposed to tackle the gene expression prediction task, with works focusing on different aspects of the visual data. State-of-the-art methods can be divided into two paradigms: global (WSI-based) and local (patch-based) focused.

2.1. Global Methods

Global methods predict the gene expression of all the spots of a WSI at once, meaning that their input corresponds to the complete data from a high-resolution histopathology image. The most notable work of this family of methods is HisToGene [21], which receives a WSI and divides it into patches that are represented through image and positional embeddings fed to a Vision Transformer (ViT) architecture [8].

The mechanism in HisToGene enables the model to consider spatial associations between spots [21]. Nonetheless, this method demands many WSIs, posing a challenge as WSIs are often scarce in most datasets. Additionally, processing the entire WSI incurs in a high computational cost. Therefore, we propose a more efficient spatial analysis at a smaller scale. Instead of using an entire sample as a single data element, we adopt a patch-based strategy, enabling us to execute predictions one patch at a time. This granular approach not only conserves computational resources but also mitigates the overfitting risks associated with using large WSIs.

2.2. Local Methods

Unlike global methods, local methods estimate the gene expression one spot at a time by dividing the WSI into individual patches. Some examples of this approach include STNet [12], EGN [34], and EGGN [35]. The focal point of local methods is the visual information in the patch of interest, and they do not take into consideration characteristics such as the vicinity of the patch or long-range interactions.

For instance, STNet [12], which is one of the most popular methods, formulates the task as a multivariate regression problem, and its architecture consists of a finetuned CNN (DenseNet-121 [13]) whose final layer is replaced with a linear layer that predicts the expression of 250 genes. A characteristic strategy of STNet is that during inference, it predicts the gene expression for 8 different symmetries of that image (4 rotation angles and their respective reflections) and returns the mean result as the final estimation. This model generalizes well across datasets and has high performance when predicting the spatial variation in the expression of well-known cancer biomarkers [12].

Other examples of this approach include EGN [34] and its upgraded version, EGGN [35]. The core of these methods is exemplar guidance learning [34], a tool that they apply to base their predictions on the expressions of the patches that are most visually similar to the patch of interest. These reference patches are known as the exemplars and correspond to the nearest neighbors of a given patch in the latent space of an image encoder. The difference between these two models lies in the main processing of the input, where EGN uses the exemplars to guide a ViT [8], while EGGN uses the exemplars to build visual similarity graphs that are fed to a GraphSAGE-based backbone [10].

The key hypothesis of EGN and EGGN is that similar images have similar gene expression patterns, no matter their location within a tissue. Nevertheless, depending on the scale of the patches, this assumption could neglect their local context. For instance, if each patch contains a single cell, several similar patches with different physiological contexts might differ in their transcriptomic profile. Thus, for our approach we choose to guide our model with spatially close patches rather than with visually similar patches. When we consider the surroundings of a specific patch, we take into account its location within the tissue and the possible differences in its biological profile. As a result, we tackle the potential limitation that the scale of the input could impose.

3. SEPAL

3.1. Problem Formulation

Given an input image patch $X \in \mathbb{R}^{[H,W,3]}$, and k spatial neighbors $Z \in \mathbb{R}^{[k,H,W,3]}$, we want to train an estimator $F_\theta(\cdot)$ that predicts the difference between the gene expression y of patch X and the mean expressions in the training set $\bar{y}_{\text{train}}$. Consequently, we aim to optimize a set of parameters θ^* such that:

$$F_{\theta^*}(X, Z) \approx \Delta y = y - \bar{y}_{\text{train}} \tag{1}$$

Where, $\Delta y \in \mathbb{R}^{[n_g,1]}$ is the difference between $\bar{y}_{\text{train}} \in \mathbb{R}^{[n_g,1]}$ and the real gene expression $y \in \mathbb{R}^{[n_g,1]}$ of the patch. This paradigm shift of predicting Δy instead of y, has the purpose of allowing our method to focus directly on the nuances in the data since we are centering the dynamic range of the prediction space around zero.

3.2. Architecture Overview

SEPAL is comprised of two stages: local learning and spatial learning, which are shown in Fig.2. While local learning follows the classic approach of finetuning and image encoder, the spatial learning of SEPAL relies on representing the input patch and its neighbors as a graph, where the central node corresponds to the image for which we want to predict the gene expression. With this representation, our model has access to the visual features in the current location and in its surroundings.

Prior to the construction of the graphs, in the first stage of our proposal (Fig.2.A), we train a feature extractor $I(\cdot)$ to process an input image patch X and return a low-dimensional embedding $I_{\text{emb}} \in \mathbb{R}^{[d_{\text{emb}},1]}$. Besides, this module also outputs a local prediction $\Delta\hat{y}_i \in \mathbb{R}^{[n_g,1]}$ obtained by applying a linear layer $L(\cdot)$ to I_{emb} as follows:

$$I(X) = I_{\text{emb}} \tag{2}$$

$$L(I_{\text{emb}}) = \Delta\hat{y}_i \approx y - \bar{y}_{\text{train}} \tag{3}$$

Consequently, the preliminary prediction $\Delta\hat{y}_i$ is completely based on X and is later refined in the spatial learning stage. After training $I(\cdot)$, we fix it and use it to obtain the visual features of all the patches in the dataset. We integrate these embeddings, together with a transformer-like positional encoding, to construct a local neighborhood graph $\mathcal{G}(X)$ for each patch (Fig.2.B).

Lastly, in the spatial learning stage (Fig.2.C), input graphs are processed by a GNN Module to obtain a spatial correction vector $\hat{s} \in \mathbb{R}^{[n_g,1]}$ which is then added to $\Delta\hat{y}_i$ to obtain $\Delta\hat{y}$. This spatially aware prediction is summed with the bias $\bar{y}_{\text{train}}$ to present the final gene expression estimation $\hat{y}$ for the input patch:

$$\Delta\hat{y} = \hat{s} + \Delta\hat{y}_i \tag{4}$$

$$\hat{y} = \Delta\hat{y} + \bar{y}_{\text{train}} \tag{5}$$

3.3. Graph construction

The process of building the graphs is shown in Fig.2.B and aims to follow the spatial connectivity of the WSI. Therefore, for a patch of interest X, we first select the k neighbors within an $m-$hop vicinity of X. For example, in Fig.2B $m = 1$ and $k = 6$ because of the hexagonal coordinate geometry. We join the patch and its neighbors in a single set $P = \{X, Z\} \in \mathbb{R}^{[k+1,H,W,3]}$ and compute the visual embedding matrix $M_i \in \mathbb{R}^{[d_{\text{emb}},k+1]}$ using our frozen image encoder $I(\cdot)$. Additionally, to enrich the spatial information beyond the topology of our graphs, we calculate a positional embedding $E_{\text{pos}} \in \mathbb{R}^{[d_{\text{emb}},1]}$ for each patch in P using the 2D transformer-like positional encoder from [33]. The inputs of that encoder are the relative coordinates of each neighbor w.r.t. the center patch. This computation gives us a positional matrix $M_p \in \mathbb{R}^{[d_{\text{emb}},k+1]}$ that is added with M_i to give the final graph features M. Summarizing, we define graphs as:

$$G(X) = \mathcal{G}(P, E, M) \tag{6}$$

$$M = M_i + M_p \tag{7}$$

Where E is a binary and undirected set of edges defined by dataset geometry.

3.4. Spatial Learning Module

Once a graph $\mathcal{G}(X)$ is fed to the spatial learning module, it is passed through a series of h Graph Convolutional Operators ($\text{GNN}_i(\cdot)$) with a sequence $C = \{d_{\text{emb}}, c_1, c_2, \ldots, c_{h-1}, n_g\}$ of hidden channels following the recursive expression:

$$g_0 = \mathcal{G}(X) \tag{8}$$

$$g_{i+1} = \sigma\left(\text{GNN}_i(g_i)\right) \tag{9}$$

$$\hat{s} = \text{Pooling}(g_h) \tag{10}$$

Where g_i is the representation of $\mathcal{G}(X)$ at layer $i \in \{0, 1, 2, \ldots, h\}$, $\sigma(\cdot)$ is an activation function, and the $\text{Pooling}(\cdot)$ operator represents a global graph pooling operator. The correction vector $\hat{s}$ represents the contribution of local spatial information to the final prediction.

4. Experiments

4.1. Datasets

We evaluate our performance in two breast cancer datasets produced with different technologies: (1) the 10x Genomics breast cancer spatial transcriptomic dataset [Section 1, Section 2] (referred to as *Visium* because of the experimental protocol), and (2) the human breast cancer in situ capturing transcriptomics dataset [27, 26] (referred to as *STNet dataset* because of the first deep learning method that used this data). The Visium dataset contains two slide images from a breast tissue sample with invasive ductal carcinoma from one patient, each with 3798 and 3987 spots of $\approx 55\mu$m detected under the tissue. On the other hand, STNet dataset consists of 68 slide images of H&E-stained tissue from 23 patients with breast cancer and their corresponding spatial transcriptomics data. Specifically, the number of spots of size $\approx 150\mu$m varies between 256 and 712 in each replication, so the complete dataset contains $30,612$ gene expression data points with their respective spatially associated image patch. For both datasets, we take reshaped patches to a $[224, 224, 3]$ dimension as input for SEPAL.

4.2. Benchmark

To design a robust benchmark, we focus on three main characteristics: (1) a bioinformatic pipeline on par with current best practices in transcriptomic analysis, (2) a pepper

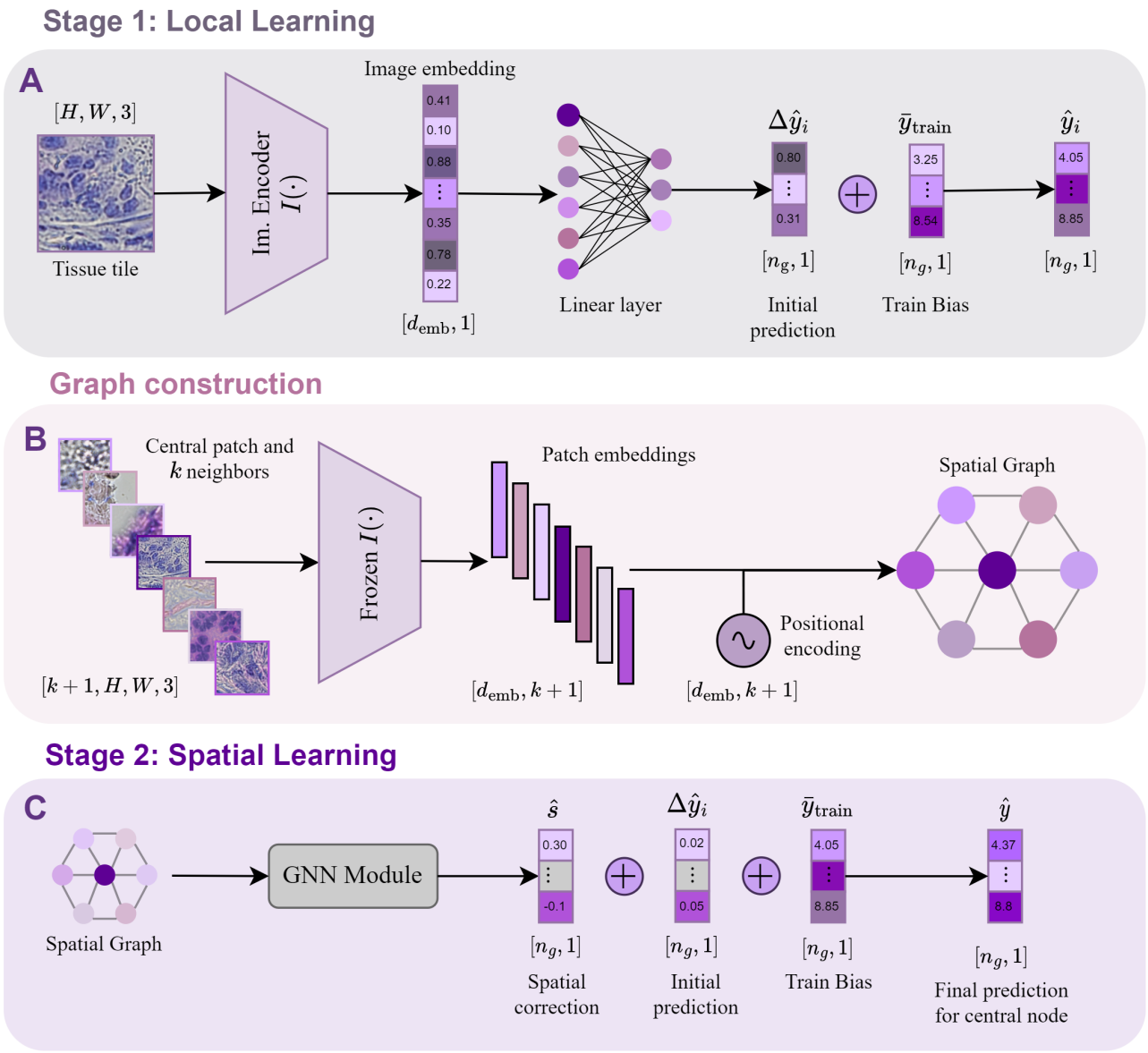

Figure 2. (A)First stage of our proposal. Pretraining of the Image Encoder $I(\cdot)$ and a linear layer $L(\cdot)$ to output the Image Embedding (I_{emb}) of a patch X, along with a preliminar prediction $\Delta\hat{y}_i$ of the difference between the expression in the patch and the mean expression in the train dataset. (B) The Graph Construction process begins with an image patch of interest and its spatial neighbors to build the graph representation based on the patch embeddings returned by the frozen $I(\cdot)$ and the positional encoding of each neighbor. (C) Architecture of the spatial learning module, which receives as input a Spatial Graph of the patch neighborhood and applies a GNN to predict the spatial correction $\hat{s}$ that further improves the $\Delta\hat{y}_i$ to get the $\Delta\hat{y}$ associated to the center patch of the graph and obtain the final gene expression prediction $\hat{y}$.

noise filter to improve data quality and allow better model training, and (3) a selection strategy to ensure that all genes have spatial patterns.

In terms of the processing pipeline, we first filter out both genes and samples with total counts outside a defined range (See Supplementary Table 1 for detailed values in each dataset). Then, we discard genes based on their sparsity. Here, we ensure that the remaining variables are expressed in at least ε_T percent of the total dataset and ε_{WSI} percent of each WSI. Following the filtering, we perform TPM [32] gene normalization and a $\log_2(x+1)$ transformation.

To address pepper noise, we applied a modified version of the adaptive median filter [14]. Shortly, for each zero value in a gene map, we replace it with the median of a growing circular region around the interest patch up to the

7^{th} unique radial distance. If no value is obtained at the end of this process, we assign the median of nonzero entries of the WSI. The results of this procedure can be appreciated for a particularly noisy gene map in Figure 3. It is worth noting that the percentage of imputed values is 5.3% and 26.0% for the Visium and STNet datasets, respectively, as we have already filtered genes based on their sparsity ($\varepsilon_T, \varepsilon_{WSI}$).

Once the bioinformatic pipeline and the denoising procedure are complete, we select the final prediction variables with the help of Moran's I [20]. This statistic is a spatial autocorrelation measure and can detect if a given gene has a pattern over spatial graphs. The closer its value to one, the more autocorrelated the variable is. For our benchmark, we compute Moran's I for every gene and WSI and average across the slide dimension. We select the top $n_g = 256$ genes with the highest general Moran's I value as our final prediction variables (See supplementary Figures 4-7). Finally, if batch effects are observed in UMAP [19] embeddings (Supplementary Figures 1-3) of the data (only seen in the STNet dataset), they are corrected with ComBat [15].

Summarizing, the processed Visium and STNet datasets have a total of 7,777 and 29,820 samples, respectively, along with a set of 256 prediction genes. As the Visium dataset only contains two WSIs, we use one for training (3795 samples) and the other one as the validation/test set (3982 samples). For the STNet dataset, from the 23 patients, we randomly choose 15 for training (20,734 samples), 4 for validation (3,397 samples), and 4 for testing (5,689 samples).

4.3. Evaluation Metrics

We use three standard metrics in multivariate regression problems: global standard errors (MSE, MAE), Pearson Correlation Coefficients (PCC-Gene, PCC-Patch), and linear regression determination coefficients (R2-Gene, R2-Patch). Both PCC and R2 have gene and patch variants since they address two aspects of the problem. The gene type metrics aim to quantify how good expression maps are in general, while the patch type metrics evaluate how good multiple gene predictions are for a specific patch. For instance, to compute PCC-Gene, we obtain PCC values for each one of the n_g gene maps and then average over the gene dimension. Conversely, computing PCC-Patch involves calculating PCC values for each patch and the average over that dimension. Importantly, the imputed values are ignored for metric computation, and consequently, performance measurements are based exclusively on real data.

4.4. State-of-the-art Methods

We compare SEPAL to four of the most popular methods in this task, including three local options (STNet, EGN, EGGN), as well as one global method (HisToGene). For a fair comparison, we choose the best performance between 50 different training protocols. If the method allows batch size as a hyperparameter (STNet, EGN), we test combinations with an empirical Bayes approach by selecting learning rates in the logarithmic range $[10^{-2}, 10^{-6}]$ and batch sizes from the list $[32, 64, 128, 256, 320]$. If the method only accepts the learning rate, we perform a logarithmic grid search within the range $[10^{-2}, 10^{-6}]$. Both the best epoch during training and the best model of the sweep are selected based on the validation MSE. The only exception to this protocol (due to computational cost) is the STNet method in the STNet dataset, for which we report the best between the original hyperparameters and the best Visium hyperparameters.

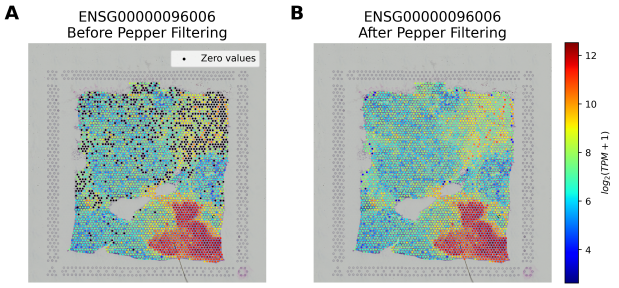

Figure 3. Example of the pepper denoising for a specific gene map in the Visium dataset.

4.5. Architecture Optimization

We extensively experiment with our spatial module, aiming to select the most effective architecture to integrate local information. For this purpose, we: (1) optionally introduce pre-processing and post-processing stages via multilayer perceptrons of varying sizes, (2) allow the positional encoding to be added or concatenated during the graph construction, (3) change the number of hops m from one to three, (4) try six different convolutional operators, and (5) vary the hidden dimensions h of our graph convolutional network going from one to four layers. Furthermore, we train all architecture variations with 12 different settings of learning rate and batch size. For a detailed explanation of every tunned hyperparameter, we refer the reader to the Supplementary Material (Sec. 2).

4.6. Implementation Details

After comprehensive experimentation (Supplementary Material Sec. 3), we choose ViT-B-16 [8] as our image encoder. We use ELU as our activation function, and SAGPooling [18] as our pooling function. We train in the denoised version of the dataset but only use real data for metric computation during inference. We implement SEPAL using Pytorch [22] and Pytorch geometric [9] for graph operators. All experiments run on a single NVIDIA

Quadro RTX 8000 GPU.

5. Results

Table 1 presents the final hyperparameter configurations of SEPAL for the Visium and the STNet datasets.

Hyperparameters	STNet Dataset	Visium
Number of hops	1	3
Embeddings aggregation	Sum	Concat
Graph operator	GraphConv[16]	GCNConv[17]
Preprocessing Stage	-	d_{emb}, 512
Graph hidden channels	d_{emb}, 256	512, 256, 128
Postprocessing Stage	-	128, 256
Learning rate	10^{-4}	10^{-5}
Batch size	256	256

Table 1. Hyperparameters with the best performance for both datasets.

5.1. Main Results

		Local			Global	Hybrid	
	Method	**STNet** [12]	**EGN** [34]	**EGGN** [35]	**HisToGene** [21]	**SEPAL**	**SEPAL***
Visium	MAE	0.654	0.659	0.645	0.665	**0.630**	0.636
	MSE	0.762	0.772	0.736	0.784	**0.708**	0.717
	PCC-Gene	0.300	0.314	0.313	0.199	**0.383**	0.353
	R2-Gene	0.053	0.038	0.070	0.024	**0.106**	0.091
	PCC-Patch	0.924	0.922	0.926	0.921	**0.928**	0.927
	R2-Patch	0.843	0.841	0.846	0.839	**0.853**	0.851
STNet dataset	MAE	0.560	0.520	0.550	0.529	**0.519**	0.527
	MSE	0.537	0.480	0.549	0.493	**0.478**	0.489
	PCC-Gene	0.030	**0.064**	0.011	-0.007	-0.004	0.002
	R2-Gene	-0.165	-0.037	-0.228	-0.066	**-0.028**	-0.052
	PCC-Patch	0.910	**0.911**	0.908	**0.911**	**0.911**	**0.911**
	R2-Patch	0.779	0.806	0.780	0.799	**0.809**	0.802

Table 2. Quantitative comparison with state-of-the-art methods on Visium and STNet datasets. The best performance is written in **bold**, and the second best result is underlined for each metric. *: SEPAL architecture with optimal parameters from the other dataset

Table 2 depicts the performance of local and global state-of-the-art methods against SEPAL on the Visium and STNet datasets. Our method consistently outperforms these methods on all but one evaluation metric. In particular, we attend primarily to the standard error metrics and find that SEPAL presents significant improvements on the Visium dataset and performance on par with state-of-the-art for the STNet dataset. Likewise, the R2 metric calculated on the genes increases for both datasets when using SEPAL.

Furthermore, we find that calculating the PCC and the R2 metrics in a gene-wise fashion results in a remarkably poorer performance compared to the patch-wise evaluation. This means that predicting the distribution of the expression of a single gene in a WSI is a significantly more difficult task than aiming to obtain the expression of all the genes in one single spot. Nevertheless, despite this different trend for gene or patch-focused evaluations, our method consistently achieves the best results.

Method	**ViT**	**ViT+Δ**	**ViT+Δ+S7**	**SEPAL**
MAE	0.655	0.638	0.648	**0.630**
MSE	0.760	0.725	0.737	**0.708**
PCC-Gene	0.282	0.347	0.339	**0.383**
R2-Gene	0.053	0.086	0.065	**0.106**
PCC-Patch	0.924	0.927	0.925	**0.928**
R2-Patch	0.843	0.849	0.847	**0.853**

Table 3. Control experiments on the Visium validation/test set. Δ: predicting differences with respect to the mean expression $\bar{y}_{\text{train}}$. S7: input patch is 7 times bigger than the original one.

HisToGene has poorer performance on Visium than on STNet, and overall it shows the worst results on the Visium dataset. These differences within the results of HisToGene support the observation that data scarcity of small datasets like Visium leads to deficient results in global methods. Conversely, we demonstrate that our method is able to retrieve important information from the input despite the difference in the data acquisition technologies and number of samples since it achieves high performance on both datasets.

Finally, Fig.5 shows a histogram of the PCC between the ground-truth and the predictions of each gene on Visium. None of the genes has a negative correlation, and the lowest PCC is 0.052 and goes as far as 0.639. Overall, our model has a satisfactory performance for the evaluation of the genes selected. We observe that the PCC has an approximately normal distribution, with no evident outliers.

As a sidenote observation, we also validated our data imputation protocol by obtaining the main results for the Visium dataset when training in noisy data. The metrics show a consistent drop in performance disregarding the method (Sec. 4 Supplementary Material), which supports the need for our denoising approach and sets a best practice for future works.

5.2. Control Experiments

Table 3 shows the results for the ablation experiments. Comparing the results between predicting the absolute expression (ViT) and predicting the expression variations of the genes (ViT+Δ), we notice that the latter option has a better performance in every metric. For instance, when predicting delta variations, the MSE is 0.035 points below that of the absolute expression prediction. The PCC-Gene also increased 0.065 points with our problem formulation. These results reflect the suitability of the paradigm shift that we propose by learning the difference between y and $\bar{y}_{\text{train}}$ instead of directly predicting y.

We evaluate the benefit of using a larger neighborhood to determine how raw spatial information affects gene pre-

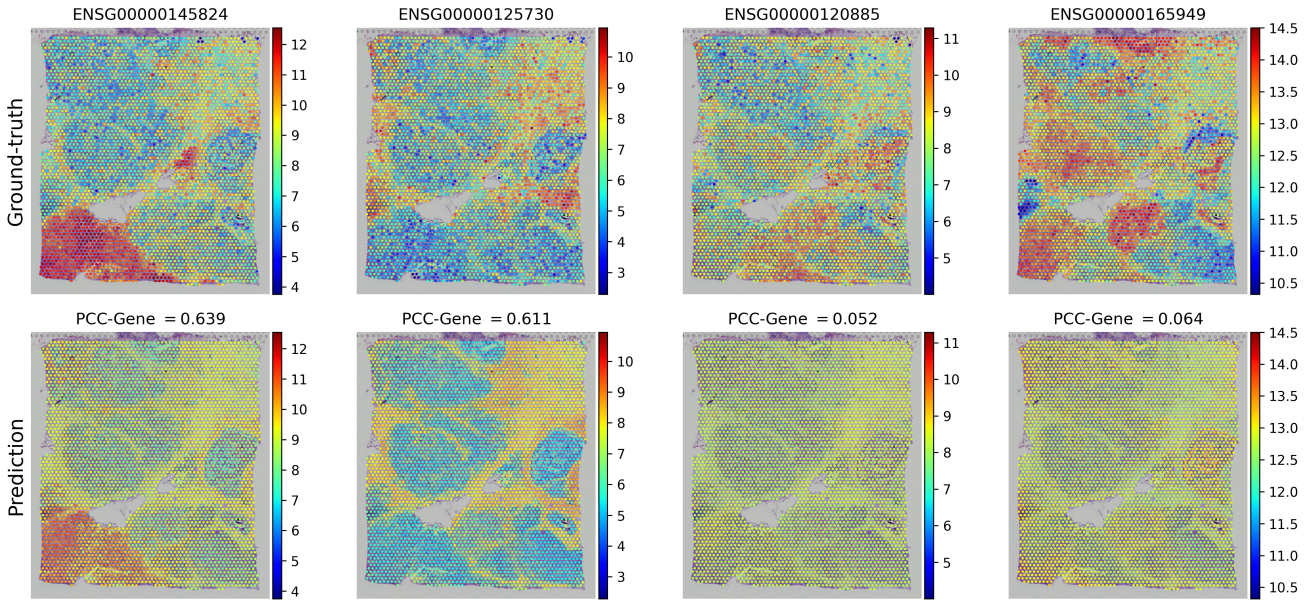

Figure 4. Visualization of the two genes with the highest (left) and lowest (right) Pearson Correlation Coefficient. At the top is the Ground-Truth of the expression and at the bottom is the qualitative prediction of our method with its respective PCC.

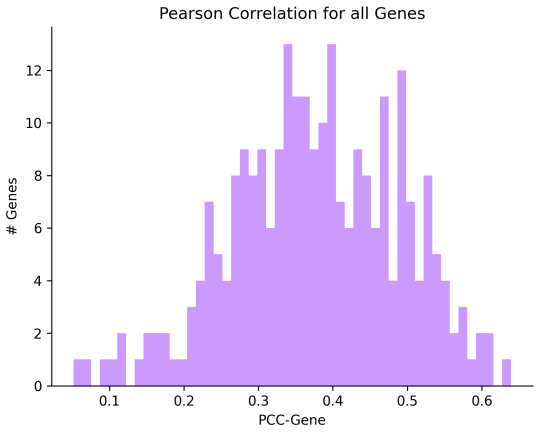

Figure 5. Histogram of the Pearson correlation between the ground-truth and the prediction of each gene. The X-axis displays the values of the Pearson correlation coefficient, while the Y-axis shows the number of genes that have that particular correlation.

diction (ViT+Δ against ViT+Δ+S7). Table 3 compares the behavior of the exact same image encoder while solely altering the scale of the patches (with seven times more visual context). For all metrics, keeping a scale of 1.0 remains the best option among the ViT architectures tested. Our findings suggest that increasing the visual coverage of an image encoder does not yield significant improvements in gene prediction.

In addition, the results from SEPAL show an improvement over all metrics, with respect to ViT+Δ+S7. Notably, both SEPAL and ViT+Δ+S7 have access to the same visual context in the WSI and are differentiated only by how spatial information is represented. This compelling outcome underscores the importance of incorporating spatial features in the description of each patch and constructing graphs to glean highly relevant information for accurate expression prediction. The performance of SEPAL shows that predictions from the spatial module do further improve the preliminary predictions obtained during the local learning stage. Our results validate the efficacy of our novel approach, emphasizing the value of spatial interactions in gene expression prediction.

5.3. Qualitative Results

Figure.4 shows the heatmaps for the real and the predicted expression distribution of the genes with the best and worst performances. Focusing on the genes with the highest PCC, we see that for the second best gene, the expressions both on the ground-truth and on the prediction are highly associated with the tissue color. Note that the regions with darker tissue obtain lower expression predictions, and regions with lighter tissue obtain higher expression predictions. These results suggest that our model might be basing estimates solely on the color of the patches rather than looking for specific morphology patterns. Nevertheless, for the best gene, the predicted expressions are not uniformly the same for all dark or light tissue sections, conveying that our model does not rely only on the tone of images and is actually learning from the spatial context of the patches and tissue morphology.

The predicted expressions show a lower intensity than the ground-truth for both genes, indicating that the dynamic range of SEPAL predictions may not match that of the real expression levels. Notably, for the two genes with the highest PCC, the output of our method appears over-smoothed compared to the ground-truth. An evident distinction arises when comparing the real expression, which exhibits adjacent spots with drastically different expression levels, to the predictions, where no regions display sudden changes in expression tendencies. While our model's consistent predictions showcase its strength, this attribute may also be considered a drawback when seeking to detect gene expression deviations with high spatial resolution.

Regarding the hardest genes, Fig.4 shows that the predictions tend to correspond to the mean expression value of each gene and are practically constant throughout the entire WSI. For these cases, SEPAL fails to capture the spatial dependencies, even though clear patterns are present in the ground-truths. We hypothesize that these shortcomings are due to the joint prediction of 256 genes.

6. Conclusions

In this work, we develop a novel framework to approach the spatial gene expression prediction task by integrating local context and exploiting inductive biases inherent to the biological nature of the problem. Our proposed SEPAL consistently outperforms state-of-the-art models and closes the gap between completely global and completely local analysis. Furthermore, aligning with biological expectations, it is capable of recognizing patterns in histological data that go beyond simple color intensities. Consequently, our approach represents a significant step forward in spatial expression prediction, enhancing the applicability of deep learning methods in the context of disease analysis and precision medicine.

7. Acknowledgements

Gabriel Mejia acknowledges the support of a UniAndes-DeepMind Scholarship 2022. This work was supported by Azure sponsorship credits granted by Microsoft's AI for Good Research Lab. We thank Andres Hernandez for his valuable help during the conception of the project.

References

[1] Zachary B Abrams, Travis S Johnson, Kun Huang, Philip RO Payne, and Kevin Coombes. A protocol to evaluate rna sequencing normalization methods. *BMC bioinformatics*, 20(24):1–7, 2019. 2

[2] Areej Alsaafin, Amir Safarpoor, Milad Sikaroudi, Jason D. Hipp, and H. R. Tizhoosh. Learning to predict rna sequence expressions from whole slide images with applications for search and classification. *Communications Biology 2023 6:1*, 6:1–9, 3 2023. 1

[3] Michaela Asp, Joseph Bergenstråhle, and Joakim Lundeberg. Spatially resolved transcriptomes—next generation tools for tissue exploration. *BioEssays*, 42(10):1900221, 2020. 1

[4] Shaked Brody, Uri Alon, and Eran Yahav. How attentive are graph attention networks? *arXiv preprint arXiv:2105.14491*, 2021. 11

[5] Anton Buzdin, Maxim Sorokin, Andrew Garazha, Alexander Glusker, Alex Aleshin, Elena Poddubskaya, Marina Sekacheva, Ella Kim, Nurshat Gaifullin, Alf Giese, Alexander Seryakov, Pavel Rumiantsev, Sergey Moshkovskii, and Alexey Moiseev. Rna sequencing for research and diagnostics in clinical oncology. *Seminars in Cancer Biology*, 60:311–323, 2 2020. 1

[6] Chin-Chen Chang, Ju-Yuan Hsiao, and Chih-Ping Hsieh. An adaptive median filter for image denoising. In *2008 Second international symposium on intelligent information technology application*, volume 2, pages 346–350. IEEE, 2008. 2

[7] Joanne R Chapman and Jonas Waldenström. With reference to reference genes: A systematic review of endogenous controls in gene expression studies. *PloS one*, 10:e0141853, 11 2015. 1

[8] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020. 3, 6

[9] Matthias Fey and Jan Eric Lenssen. Fast graph representation learning with pytorch geometric. *arXiv preprint arXiv:1903.02428*, 2019. 6, 11

[10] Will Hamilton, Zhitao Ying, and Jure Leskovec. Inductive representation learning on large graphs. *Advances in neural information processing systems*, 30, 2017. 3

[11] Will Hamilton, Zhitao Ying, and Jure Leskovec. Inductive representation learning on large graphs. *Advances in neural information processing systems*, 30, 2017. 11

[12] Bryan He, Ludvig Bergenstråhle, Linnea Stenbeck, Abubakar Abid, Alma Andersson, Åke Borg, Jonas Maaskola, Joakim Lundeberg, and James Zou. Integrating spatial gene expression and breast tumour morphology via deep learning. *Nature Biomedical Engineering*, 4:827–834, 6 2020. 1, 3, 7

[13] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4700–4708, 2017. 3

[14] H. Hwang and R.A. Haddad. Adaptive median filters: new algorithms and results. *IEEE Transactions on Image Processing*, 4:499–502, 4 1995. 5

[15] W Evan Johnson, Cheng Li, and Ariel Rabinovic. Adjusting batch effects in microarray expression data using empirical bayes methods. *Biostatistics*, 8(1):118–127, 2007. 6, 13

[16] Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016. 7, 11

[17] Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016. 7, 11

[18] Junhyun Lee, Inyeop Lee, and Jaewoo Kang. Self-attention graph pooling. In *International conference on machine learning*, pages 3734–3743. PMLR, 2019. 6

[19] Leland McInnes, John Healy, and James Melville. Umap: Uniform manifold approximation and projection for dimension reduction. *arXiv preprint arXiv:1802.03426*, 2018. 6, 13

[20] Patrick AP Moran. Notes on continuous stochastic phenomena. *Biometrika*, 37(1/2):17–23, 1950. 2, 6

[21] Minxing Pang, Kenong Su, and Mingyao Li. Leveraging information in spatial transcriptomics to predict super-resolution gene expression from histology images in tumors. *bioRxiv*, page 2021.11.28.470212, 11 2021. 1, 3, 7

[22] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019. 6

[23] Benoît Schmauch, Alberto Romagnoni, Elodie Pronier, Charlie Saillard, Pascale Maillé, Julien Calderaro, Aurélie Kamoun, Meriem Sefta, Sylvain Toldo, Mikhail Zaslavskiy, Thomas Clozel, Matahi Moarii, Pierre Courtiol, and Gilles Wainrib. A deep learning model to predict rna-seq expression of tumours from whole slide images. *Nature Communications 2020 11:1*, 11:1–15, 8 2020. 1

[24] Neil J. Sebire. Oncology: histopathology and imaging in the future. *Pediatric Radiology*, 41:170–171, 5 2011. 1

[25] Yunsheng Shi, Zhengjie Huang, Shikun Feng, Hui Zhong, Wenjin Wang, and Yu Sun. Masked label prediction: Unified message passing model for semi-supervised classification. *arXiv preprint arXiv:2009.03509*, 2020. 11

[26] Linnea Stenbeck, Ludvig Bergenstråhle, Joakim Lundeberg, and Åke Borg. Human breast cancer in situ capturing transcriptomics. 5, 2021. 4

[27] Patrik L. Ståhl, Fredrik Salmén, Sanja Vickovic, Anna Lundmark, José Fernández Navarro, Jens Magnusson, Stefania Giacomello, Michaela Asp, Jakub O. Westholm, Mikael Huss, Annelie Mollbrink, Sten Linnarsson, Simone Codeluppi, Åke Borg, Fredrik Pontén, Paul Igor Costea, Pelin Sahlén, Jan Mulder, Olaf Bergmann, Joakim Lundeberg, and Jonas Frisén. Visualization and analysis of gene expression in tissue sections by spatial transcriptomics. *Science*, 353:78–82, 7 2016. 4

[28] Yoshihisa Takahashi. Histopathology of nonalcoholic fatty liver disease/nonalcoholic steatohepatitis. *World Journal of Gastroenterology*, 20:15539, 11 2014. 1

[29] Laura J van 't Veer, Hongyue Dai, Marc J van de Vijver, Yudong D He, Augustinus A M Hart, Mao Mao, Hans L Peterse, Karin van der Kooy, Matthew J Marton, Anke T Witteveen, George J Schreiber, Ron M Kerkhoven, Chris Roberts, Peter S Linsley, René Bernards, and Stephen H Friend. Gene expression profiling predicts clinical outcome of breast cancer. *Nature*, 415:530–6, 1 2002. 1

[30] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. Graph attention networks. *arXiv preprint arXiv:1710.10903*, 2017. 11

[31] Vincenzo Villanacci, Alessandro Vanoli, Giuseppe Leoncini, Giovanni Arpa, Tiziana Salviato, Luca Reggiani Bonetti, Carla Baronchelli, Luca Saragoni, and Paola Parente. Celiac disease: histology-differential diagnosis-complications. a practical approach. *Pathologica*, 112:186–196, 9 2020. 1

[32] Günter P Wagner, Koryu Kin, and Vincent J Lynch. Measurement of mrna abundance using rna-seq data: Rpkm measure is inconsistent among samples. *Theory in biosciences*, 131:281–285, 2012. 5

[33] Zelun Wang and Jyh-Charn Liu. Translating math formula images to latex sequences using deep neural networks with sequence-level training, 2019. 4

[34] Yan Yang, Md Zakir Hossain, Eric A Stone, and Shafin Rahman. Exemplar guided deep neural network for spatial transcriptomics analysis of gene expression prediction. *arXiv*, 10 2022. 1, 3, 7

[35] Yan Yang, Zakir Hossain, Eric Stone, and Shafin Rahman Projector. Spatial transcriptomics analysis of gene expression prediction using exemplar guided graph neural network. *bioRxiv*, page 2023.03.30.534914, 3 2023. 1, 3, 7

[36] Qichao Yu, Miaomiao Jiang, and Liang Wu. Spatial transcriptomics technology in cancer research. *Frontiers in Oncology*, 12:1019111, 10 2022. 1

Supplementary Material for SEPAL:

A. Benchmark

	Dataset	
Parameter	**Visium**	**STnet**
`cell_min_counts`	10^3	7×10^2
`cell_max_counts`	10^5	10^5
`gene_min_counts`	10^3	10^3
`gene_max_counts`	10^6	10^6
`min_exp_frac` (ε_{WSI})	0.8	0.01
`min_glob_exp_ frac` (ε_T)	0.8	0.6
`combat_key`	-	Patient

Table 4. Filter hyperparameters for Visium and the STNet dataset

In order to filter our datasets, we define the hyperparameters shown in Table 4. The terms `cell_min_counts` and `cell_max_counts` refer to the minimum and maximum total counts required for a cell to remain in the dataset. Similarly, `gene_min_counts` and `gen_max_counts` specify the minimum and maximum total counts necessary for a gene to be included in the dataset. The parameter `min_exp_frac` determines the fraction of observations in a slide that must express a gene to consider it. Likewise, `min_global_exp_frac` is the fraction of observations across the entire dataset needed to include a gene. Finally, `combat_key` sets the key in dataset to use as batch for ComBat. However, in the case of Visium, this parameter is set to "None" indicating that batch correction is unnecessary.

B. Architecture Optimization

We varied different stages of our model that are enumerated as follows:

1. **Positional encodings**: We decided to sum or concatenate the positional embedding to the patch embedding.

2. **Number of hops** (m): We tuned the number of steps required to reach from one node to another within the graph between 1, 2, and 3 in the Visium dataset, and only 1 in the STNet dataset due to computational cost.

3. **Convolutional graph operator** ($\text{GNN}_i(\cdot)$: We tried different convolutional graph operators from pytorch geometric [9] such as GCNConv[17], SAGEConv[11], GraphConv[16], GATConv[30], GATv2Conv[4], and TransformerConv[25].

4. **Learning rate**: We used 10^{-4}, 10^{-5} and 10^{-6} as learning rate options.

5. **Batch size**: We experimented with batches of size 512, 256, 128 and 64.

6. **Pre-processing stage**: We tried to progressively reduce the dimensionality of our graph input by adding an MLP with the following options of hidden channels h_i:

   - No MLP
   - $h_i = \{d_{\text{emb}}, 512\}$
   - $h_i = \{d_{\text{emb}}, 512, 256\}$
   - $h_i = \{d_{\text{emb}}, 512, 256, 128\}$

   Where d_{emb} refers to the embedding dimension (768 if the positional encoding is sumed or 1536 if it is concatenated).

7. **Post-processing stage**: We tried to progressively decompress the graph network output adding an MLP with the following options of hidden channels h_o:

   - No MLP
   - $h_o = \{128, n_g\}$
   - $h_o = \{64, 128, n_g\}$

   Where n_g is the number of genes to predict, set at 256.

8. **Graph hidden channels**: To fit the dimensions of pre and post-processing stages, we follow an autoencoder-like architecture by powers of 2. The options of hidden channel dimensions go from 1 to 4 graph convolutional layers as shown in Table 6.

With this systematic procedure, we generate 3888 hyperparameter combinations from 324 module variations in the Visium dataset, and 1296 hyperparameter combinations from 108 module variations in the STNet dataset.

C. Image Encoder Selection

To establish the architecture of our image encoder we experiment with nine of the most recent and popular image classification models, replacing their respective last layer to predict the expression of the n_g genes of interest. With each potential image encoder we do a grid search modifying the learning rate inside $[10^{-2}, 10^{-3}, 10^{-4}, 10^{-5}, 10^{-6}]$ and the batch size inside $[320, 256, 128, 64]$ to achieve the best possible performance in the Visium dataset. Table 5 shows the best results for each one of the architectures. The results show that ShuffleNet and ViT have similar performance in most metrics, achieving the best results among almost all the architectures examined. However, ViT has a slightly better MSE (0.7245 Vs 0.7249), which is the selection criteria. Hence, we build SEPAL applying ViT as the image encoder $I(\cdot)$.

Method	ResNet18	DenseNet-121	ConvNeXt-T	WideResNet-50	MobileNetV3-S	ResNeXt-50	ShuffleNetV2_X0_5	Swin-T	ViT-B-16
MAE	0.653	0.641	0.653	0.649	0.645	0.642	**0.635**	0.650	<u>0.638</u>
MSE	0.761	0.745	0.754	0.756	<u>0.736</u>	0.739	**0.725**	0.751	**0.725**
PCC-Gene	0.296	**0.379**	0.294	0.323	0.338	0.341	<u>0.354</u>	0.303	0.347
R2-Gene	0.051	0.071	0.050	0.060	0.084	0.080	**0.094**	0.055	<u>0.086</u>
PCC-Patch	0.923	<u>0.925</u>	0.924	0.924	<u>0.925</u>	<u>0.925</u>	**0.927**	0.924	**0.927**
R2-Patch	0.843	0.846	0.844	0.844	0.846	0.846	**0.850**	0.844	<u>0.849</u>
Params (M)	11.7	8.0	28.6	68.9	2.5	25.0	1.4	28.3	86.6
Batch Size	64	320	128	320	128	64	256	256	320
lr	1×10^{-2}	1×10^{-3}	1×10^{-4}	1×10^{-2}	1×10^{-2}	1×10^{-3}	1×10^{-2}	1×10^{-5}	1×10^{-4}

Table 5. Results of the extensive experimentation performed on image encoder selection over the Visium dataset. The best performance is written in **bold**, and the second best result is <u>underlined</u> for each metric.

MLP pre	GNN	MLP pos
-	d_{emb}, n_g	-
-	d_{emb}, 512, n_g	-
-	d_{emb}, 512, 256, 128	128, n_g
-	d_{emb}, 512, 256, 128, 64	64, 128, n_g
d_{emb}, 512	512, 256, n_g	-
d_{emb}, 512, 256	256, 128, n_g	-
d_{emb}, 512, 256	256, 128, 64	64, 128, n_g
d_{emb}, 512	512, 256, 128	128, n_g
d_{emb}, 512, 256, 128	128, 128, 128, 128	128, n_g

Table 6. Hidden channels dimensions for preprocessing MLP, Graph Neural Network and postprocessing MLP.

Method	STNet	EGN	EGGN	HisToGene	SEPAL
MAE	0.731	0.785	**0.651**	0.742	<u>0.663</u>
MSE	0.955	1.077	**0.745**	0.967	<u>0.789</u>
PCC-Gene	0.106	-0.028	**0.297**	-0.048	<u>0.292</u>
R2-Gene	-0.189	-0.304	**0.059**	-0.191	<u>0.012</u>
PCC-Patch	0.911	0.901	**0.926**	0.901	<u>0.922</u>
R2-Patch	0.796	0.770	**0.845**	0.800	<u>0.836</u>

Table 7. Results of SEPAL along with state-of-the-art models when trained in the noisy version of the Visium dataset. The best performance is written in **bold**, and the second best result is <u>underlined</u> for each metric.

D. Training with Noisy Data

To validate our denoising approach, we replicate the experimentation pipeline (training protocol optimization) for the main results in the Visium dataset. However, this time the training data does not pass through the modified adaptive median filter resulting in a noisy training dataset. The results can be observed in Table 7 and, outstandingly, even with just 5.3% of noisy data, the performance of all methods degrades significantly. This observation validates our missing data imputation protocol disregarding the computational method and sets a best practice for future works. Importantly SEPAL falls to the second position of the ranking because the noisy data greatly affects the computation of $\bar{y}_{\text{train}}$ producing a systematic bias in the predictions.

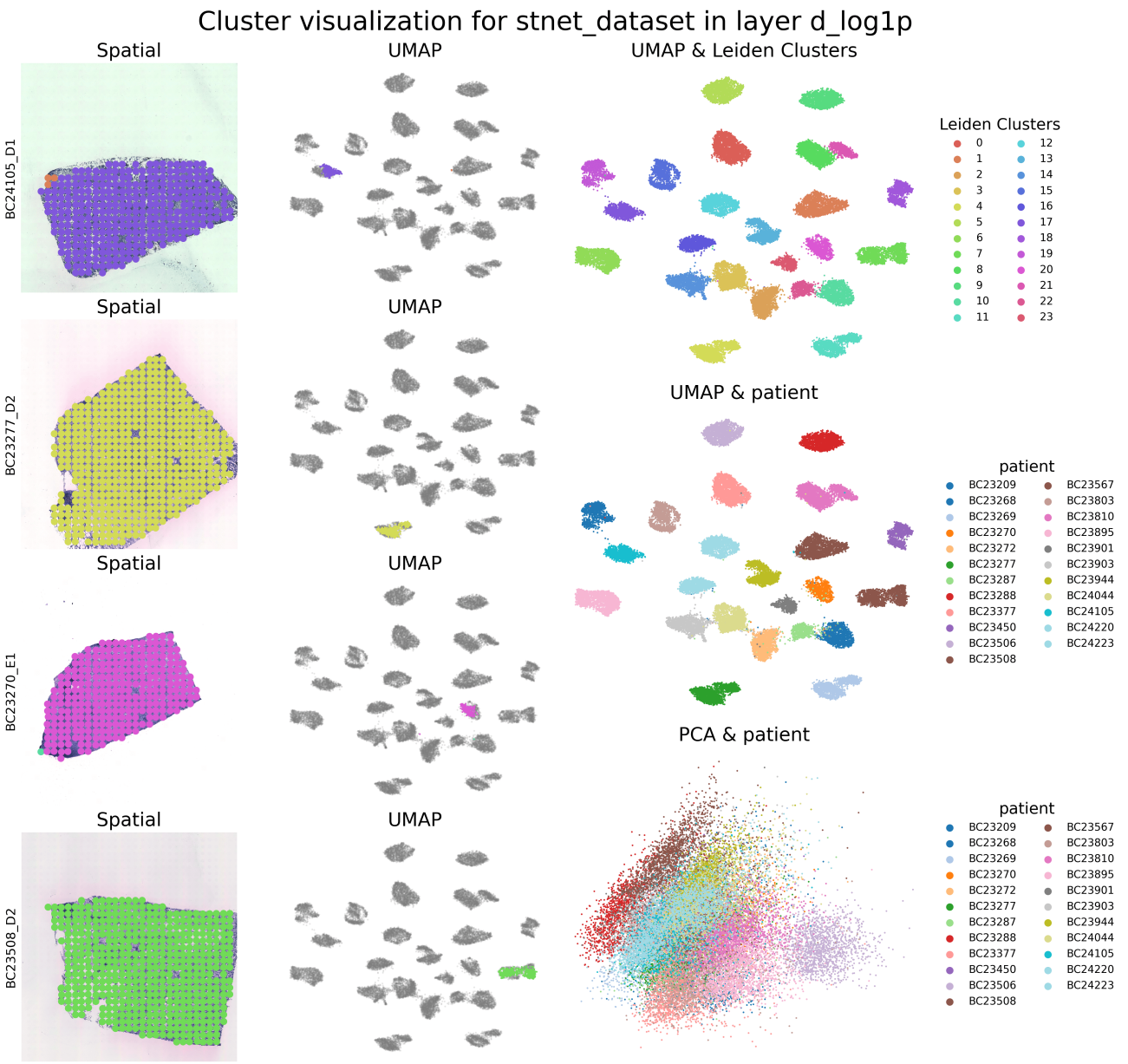

Figure 6. UMAP dimensionality [19] reduction and clusters visualization for STNet dataset without batch correction. As strong batch effects are seen depending on the patient, ComBat [15] was applied with the patient as the batch key in the processing pipeline of this dataset.

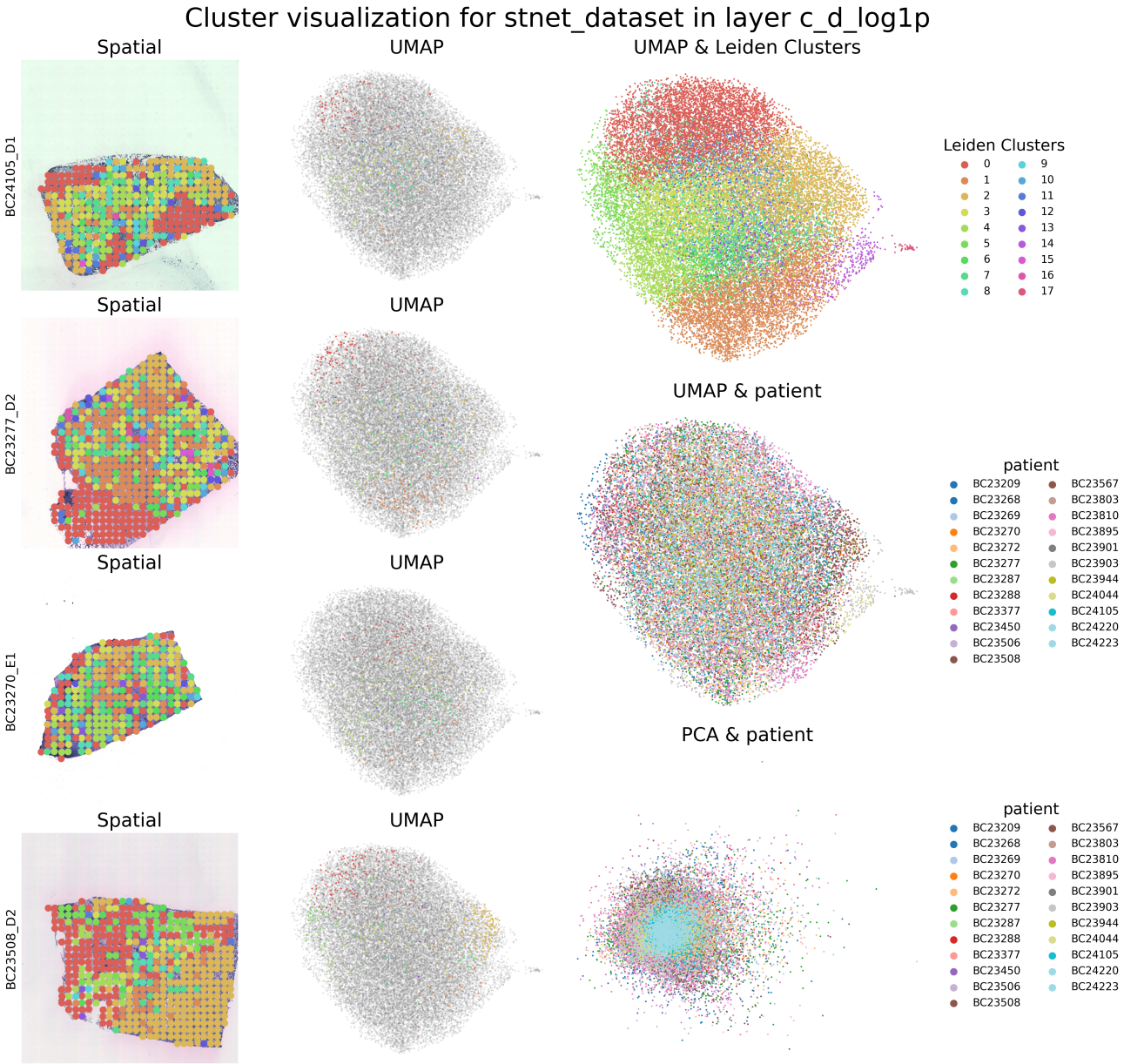

Figure 7. UMAP dimensionality reduction and clusters visualization for STNet dataset after ComBat batch correction.

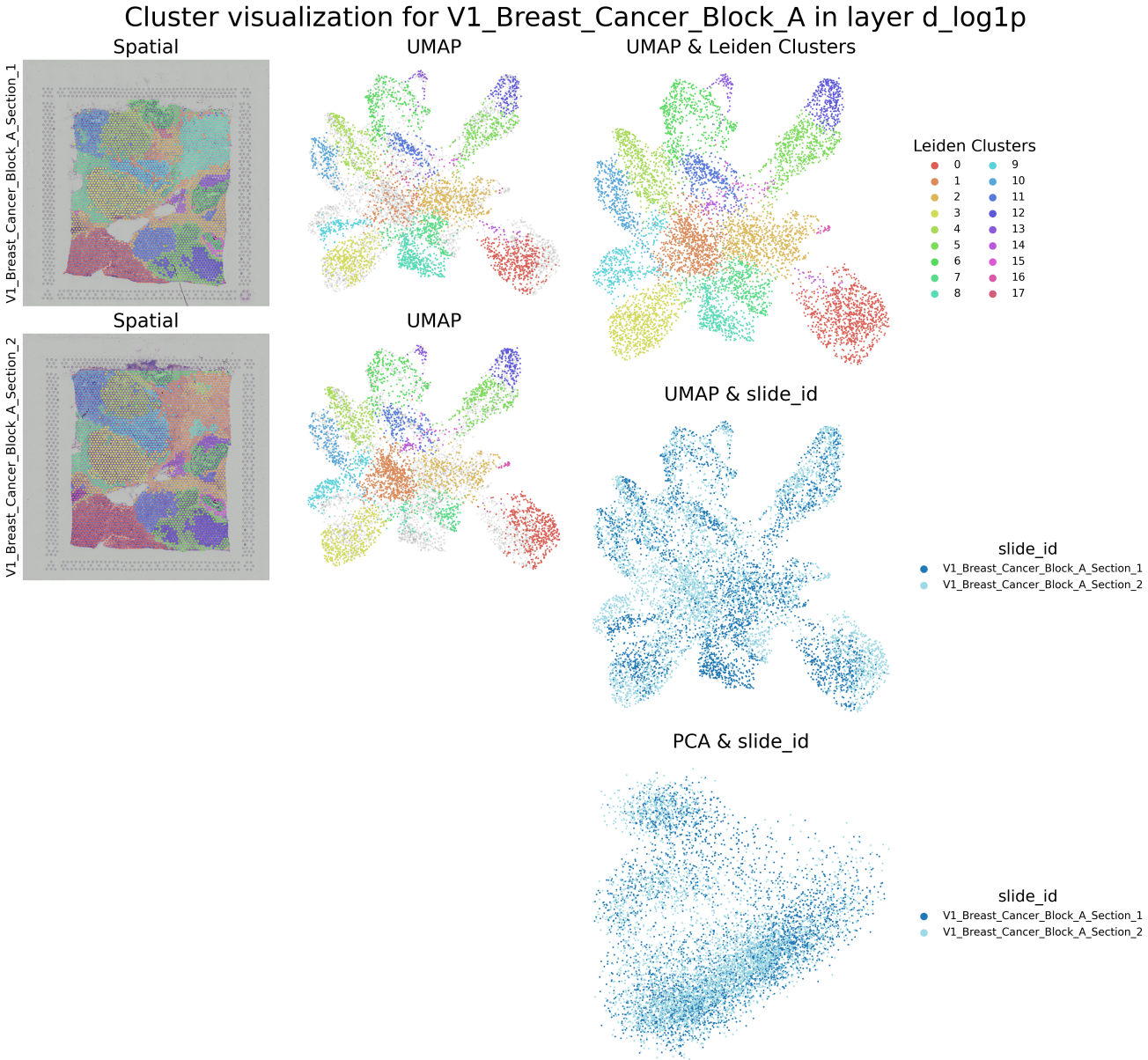

Figure 8. UMAP dimensionality reduction and clusters visualization for Visium dataset without batch correction. As no evident batch effects are present, ComBat was not applied in the processing pipeline of this dataset.

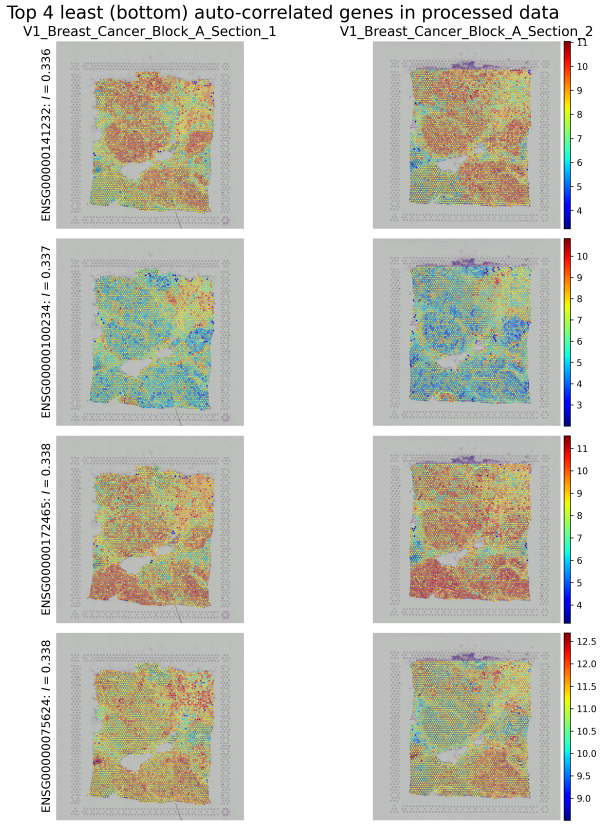

Figure 9. Visualization of the four least auto-correlated genes in processed data for Visium in both whole slide images.

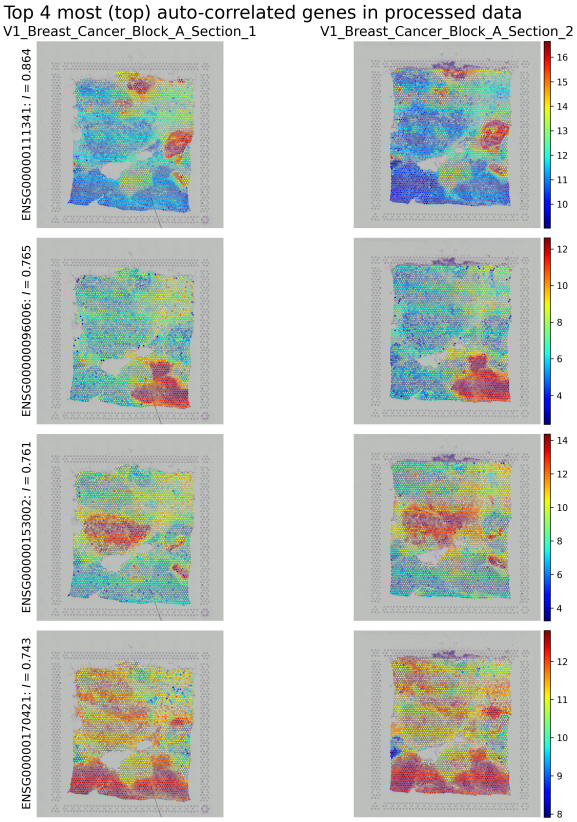

Figure 11. Visualization of the four most auto-correlated genes in processed data for Visium in both whole slide images.

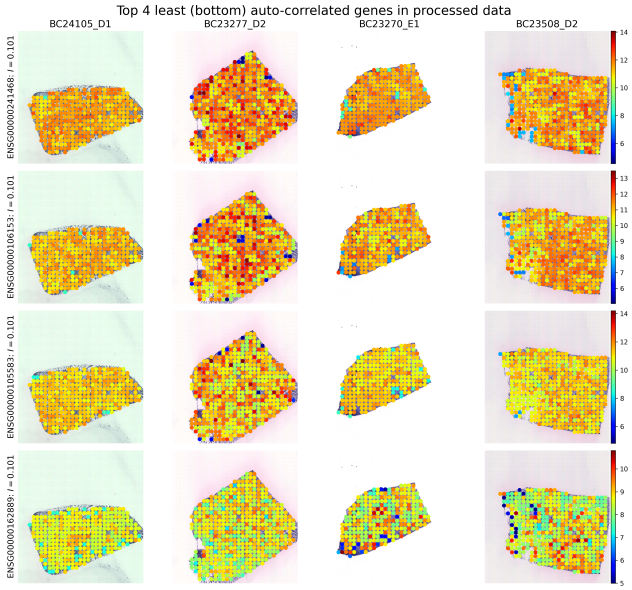

Figure 10. Visualization of the four least auto-correlated genes in processed data for STNet in four whole slide images.

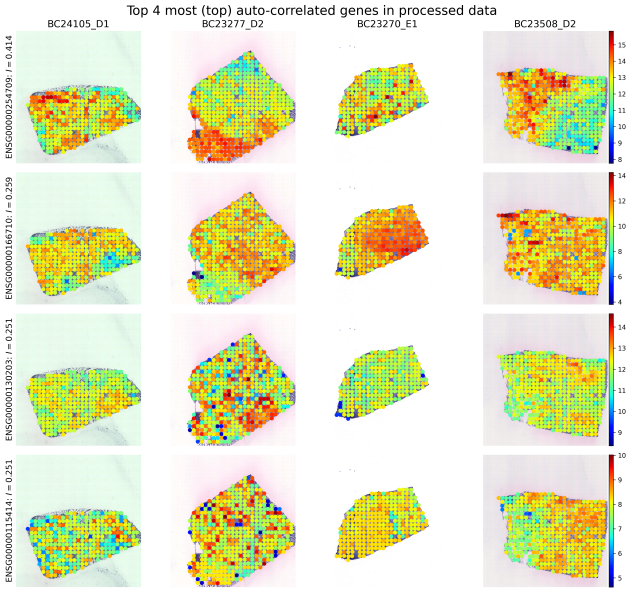

Figure 12. Visualization of the four most auto-correlated genes in processed data for STNet in four whole slide images.