Integrating Histology Imaging and Spatial Transcriptomics through Low-Dimensional Embeddings to Predict Gene Expression Patterns

Gaurav Anand
gauvand9@stanford.edu

Lauren Cooper
cooperlc@stanford.edu

Susie Avagyan
savagyan@stanford.edu

***Abstract*—Spatial transcriptomics (ST) is emerging as a powerful tool for understanding biological information at the tissue, cellular, or sub-cellular scale. On the flip side, existing technologies to obtain and analyze this data can be complex and costly. Here, we present multiple approaches for predicting gene expression from histology images, which are cost effective to obtain, along with spatial adjacency information. We calculated embeddings of our input data with ResNet50 and k-nearest neighbors. We then utilized various regression methods (linear, negative binomial) and deep learning models (a simple neural network, graph neural network). Upon further improvements, these results could be used to reduce the cost of spatial sequencing by using histology images to predict gene expression.**

I. Introduction

The rapid progress in spatial sequencing technologies has unlocked exciting directions for studying tissues in an increased cellular resolution, consequently making the development of computational methods for analyzing spatial transcriptomics (ST) data increasingly urgent. Analyzing this data can offer insights into the organization of cells within tissues that drive biological functions and processes, especially studying tumors. And while spatial transcriptomics technologies may address these needs, they are practically expensive for large-scale clinical oncology studies.

A cost-effective substitute to study spatial characteristics of tumor specimens are hematoxylin and eosin-stained histology images, which are a viable option for large-scale studies. Prior research has indicated a connection between patterns of gene expression and the features observed in histological images, hinting at the potential to predict gene expression based on histology. [1]

To motivate the problem of predicting gene expression from images, it's essential to note the limitations of ST data. ST data often lacks resolution or is sparse. Therefore, images can be used to fill in information or resolution which may be lacking in the genetic expression data. Additionally, images may be used to provide insights into genetic information when researchers lack the resources required for expensive spatial sequencing experiments. Histology images contain distinct information relevant to cellular communities that genetic data alone may sometimes miss.

In our study, we aim to advance ST analysis by combining image processing and adjacency information while incorporating statistical models of gene expression levels. Ultimately the developed model will be able to predict gene expression given just a histology image. [6]. This model could potentially be used for identification of spatially localized cellular communities and gene expression patterns as well [7].

II. Related Work

Previous authors have explored the potential of combining images with gene expression for enhanced understanding of ST data, or in some cases, previous authors have utilized histology images for predicting gene expression levels. Imaging-based ST techniques are becoming increasingly prevalent, delivering structural information through histology images or even molecular insights through immunofluorescence [13]. The importance of image features is often overlooked in ST analysis, but can reveal highly detailed cellular or tissue-level information [3]. In their recent study, Zhang et al. [10] leveraged high-resolution imaging to extract spatially-informed, super-resolution genetic data, utilizing a recognition neural network to include both global and microscopic anatomical features in predictions. Genetic expression was modeled with a conditional negative binomial distribution and passed through a convolutional generator network. In another study, STnet was developed to integrate ST data with histopathology images for understanding gene expression heterogeneity [6]. The authors utilized image patches as inputs into a CNN, DenseNet 121, which contains 120 convolutional layers and a fully connected layer. STnet demonstrated ability to detect subtle biological signals, as well as generalizability, successfully predicting gene expression on an external dataset. ResNet50, known for effective image classification, was used to distinguish between fibrotic and necrotic tissues, enhancing clustering capabilities in ST analysis [11]. These previous studies motivate the research direction of our projects and provide a baseline of approaches, some of which that we utilize and validate in our study.

III. Dataset and Features

We utilize the ”Human breast cancer in situ capturing transcriptomics” dataset from spatialresearch.org [8], encompassing data from 23 breast cancer patients across luminal A, luminal B, triple-negative, and HER2-positive subtypes. Our data modalities comprise spatial coordinates of spots, gene expression profiles at the cellular level, and histology

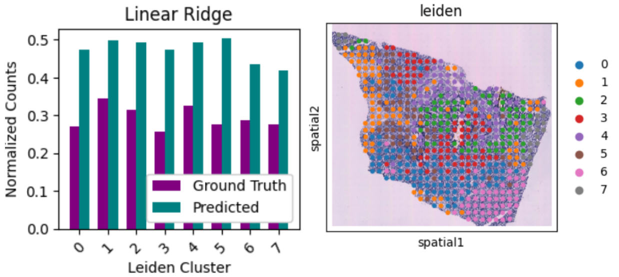

Fig. 1. (Left) Average counts per cluster, calculated with Leiden clustering, for gene TSPAN6. The ground truth and predictions are shown. (Right) Leiden clustering calculation for patient BC23506, replicate C2.

images of the sequenced tissue. The dataset includes 30,612 spots, each 100 µm in diameter, from 68 breast tissue sections, ranging from 256 to 712 spots per sample. Histology images accompanying these samples measure approximately 10,000 by 10,000 pixels. Our analysis is limited to the 250 genes exhibiting the highest variance and mean expression levels.

The dataset was split into three subsets for modeling:

- A train dataset containing 17,682 spots from across 38 tissue samples, 13 patients, and 5 cancer types.
- A validation dataset containing 6,875 spots from across 15 tissue samples, 5 patients, and 5 cancer types, and
- A test dataset containing 6,098 spots from across 15 tissue samples and 5 patients, and 5 cancer types.

A. Gene Counts Preprocessing

Many of our models utilize log-normalized counts for prediction. Log-normalization is standard practice in ST analysis, as gene counts often exhibit high variability, with highly expressed outliers skewing the distributions. Here, log-normalization is calculated with the Scanpy package, applying the transformation $X = \log(X + 1)$. For discrete models, including Poisson and Negative Binomial regression, we continue to use raw counts for optimal model compatibility.

B. Image Preprocessing

We extract features from 224x224 pixel windows centered on each spot's XY coordinates using a pretrained ResNet50 model, sans its top layer. To achieve a compact feature set, we perform PCA on the training dataset features, selecting the top 32 principal components. Additionally, we employ K-Nearest Neighbor pooling by averaging the features of each spot's 15 nearest neighbors, further refining our feature set with 32-component PCA to encapsulate both individual and regional spatial information.

C. Graph Preprocessing

We construct a graph structure relating each spot to every other spot in a sample by calculating the distance between them using their XY spatial coordinates. The adjacency matrix is based on computed 15-nearest neighborhood graph of observations using estimated connectivities of the points. To obtain a low-dimensional feature representation of the adjacency matrix for each spot, we apply truncated SVD with 32 components. [14]

IV. Methods

A. Linear Regression Baseline

Our baseline consists of a simple linear regression model for multivariate regression, with each gene as a separate task. It uses low-dimensional latent representations from two data types—embedded image data ($z_1 \in \mathbb{R}^{n\times 1}$) and dimensionality-reduced adjacency matrix data ($z_1 \in \mathbb{R}^{n\times 1}$)—to predict gene expression $\hat{y}_{i,j}$ at each spatial location. Both modalities are transformed into 32-dimensional representations, utilizing weights $w_1, w_2 \in \mathbb{R}^{n\times 1}$ for prediction.

The predicted gene expression level $\hat{y}_{i,j}$ for a given spatial location (i) and gene (j) is computed as follows:

$$\hat{y}_{i,j} = w_1^T z_{1,i} + w_2^T z_{2,i} \quad (1) \tag{1}$$

B. Negative Binomial Regression

Our raw data represents gene counts, therefore, we hypothesized that a model based on negative binomial (NB) regression would yield an improved baseline. The probability distributions for NB regression is given by Eq. 2 and is implemented with the 'statsmodels' package in Python.

$$\begin{aligned} p(y|x;\mu,\alpha) = & \frac{\Gamma(y_i + \alpha^{-1})}{\Gamma(y_i + 1)\Gamma(\alpha^{-1})} \\ & \times \left(\frac{\alpha^{-1}}{\alpha^{-1} + \mu_i}\right)^{-1/\alpha} \left(\frac{\mu}{\alpha^{-1} + \mu}\right)^{y} \end{aligned} \tag{2}$$

We use the image embedding $z_1 \in \mathbb{R}^{n\times 1}$ and adjacency embedding $z_1 \in \mathbb{R}^{n\times 1}$ as inputs for the model, similar to Eq. 1. However, in these models we omit the learned gene expression embedding $z_3 \in \mathbb{R}^{n\times 1}$ for computational simplicity. We estimate α from the data as well as test various initial values of α to enhance performance.

C. Simple neural network

We then designed a simple neural network that takes in z_1 and z_2, each contributing 32 features for a total input size of 64 features. The TensorFlow-based model for regression employs a compact sequential architecture:

- an input layer of 64 neurons (ReLU activated)
- two identical hidden layers, each with 64 neurons and ReLU activation, separated by 0.5 dropout layers for overfitting prevention.
- linearly activated output layer, dimensioned for the target gene count (250).

The model uses the Adam optimizer, mean squared error (MSE) for loss function, mean absolute error (MAE) as an additional metric for performance evaluation. Aditionally it involves an early stopping mechanism to prevent overfitting, monitoring validation loss with a patience of 30 epochs and a minimal delta of 10^{-8} for improvement. The model is trained with a batch size of 32 over a maximum of 1000 epoch.

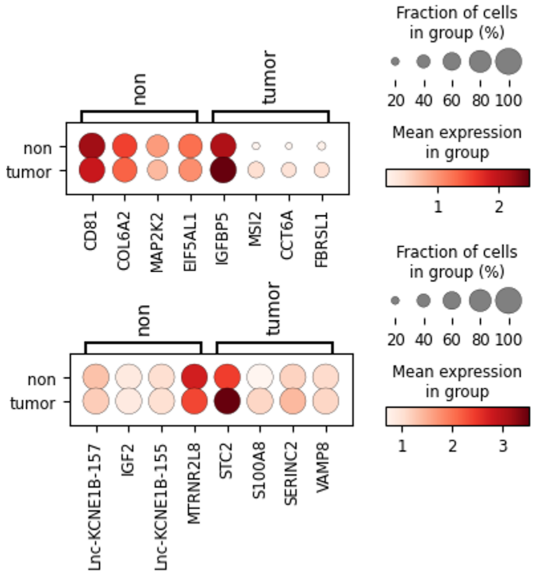

Fig. 2. Differentially expressed genes (DEG) between pathologist annotated tumor and non-tumor regions based on ground truth (first dotplot) and predicted gene expresssion values (second dotplot). Although there is no overlap in DEG identities, the prediction based DEG's still show tumor specific overexpression for genes that share biological function with ground truth based DEGs.

D. GNN

Considering the applicability, of graph neural networks (GNNs), we applied a Graph Neural Network (GNN) Regressor to predict expression levels for 250 genes using a graph $G = (V, E)$, where nodes V represent histological spots, and edges E connect spots within the 15 nearest neighbors based on spatial proximity. Each node $v_i \in V$ is tied to a feature vector $\mathbf{x}_i$ from the corresponding histology image patch, forming the node feature matrix $X \in \mathbb{R}^{n \times d}$, with $n = |V|$ and feature dimensionality $d = 32$.

The graph convolution process updates the node representations by aggregating and combining features from the neighborhood. For a node v_i, its updated representation $\mathbf{h}_i$ is computed as follows:

$$\mathbf{h}'_i = \text{FFN}\left(\text{CONCAT}\left(\mathbf{h}_i, \text{AGGREGATE}_{j \in \mathcal{N}(i)}(\mathbf{h}_j)\right)\right)$$

where $\mathcal{N}(i)$ denotes the set of neighbors of v_i in G, and AGGREGATE denotes the mean pooling operation over the features of $\mathcal{N}(i)$. CONCAT represents the concatenation of v_i's representation with its aggregated neighborhood features, and FFN denotes a feed-forward neural network that processes the concatenated vector to produce the new state of the node representation $\mathbf{h}'_i$.

E. Evaluation metrics

1) Quantitative metrics: We use multiple quantitative metrics to evaluate the performance of the models, including mean squared error (MSE) and mean absolute error (MAE). For a prediction $\hat{y}_{ij}$ and ground truth y_{ij}, the MSE is given by Eq. 3.

$$\text{MSE} = (\hat{y}_{ij} - y_{ij})^2 \tag{3}$$

$$\text{MAE} = |\hat{y}_{ij} - y_{ij}| \tag{4}$$

Model performance was assessed using R^2 (Eq. 7), Pearson (Eq. 5), and Spearman's correlation coefficients (Eq. 6), with metrics calculated and averaged separately for each of the 250 gene tasks in the multitask regression model. The performance metrics are calculated and averaged over each gene separately, since our model is defined as a multitask regression, where each of the 250 genes is one task. This enhances interpretability and robustness as gene count distributions may vary significantly.

$$r = \frac{\sum (x_i - \overline{x})(y_i - \overline{y})}{\sqrt{\sum (x_i - \overline{x})^2 \sum (y_i - \overline{y})^2}} \tag{5}$$

$$\rho = 1 - \frac{6 \sum d_i^2}{n(n^2 - 1)} \tag{6}$$

$$R^2 = 1 - \frac{\sum (y_i - \hat{y}_i)^2}{\sum (y_i - \overline{y})^2} \tag{7}$$

2) Qualitative metrics: We utilized qualitative metrics to evaluate the models performance based on the biological consistency of predictions. Such metrics include:

- *Average gene expression per spatial cluster*
  Given the spatial transcriptomics data, prior to subsetting our features to 250 target genes, we have performed Leiden clustering to detect communities of cells that have similar transcriptomic landscape and spatially colocalize. Assuming that average expression within each cluster provides an unbiased measure of gene 'signal', we averaged this signal over all genes and measured the error.
- *Differentially expressed genes between pathology regions*
  The spatial samples we used included annotations of tumor and non-tumor regions done manually by a pathologist. Certain genes become differentially expressed in tumor regions compared to non-tumor regions. Therefore, we wanted to see if differential expression analysis would show similar biological function genes using ground truth and predicted values of expression.

V. Experiment/Results/Discussion

A. Generalized Linear Models

We attempted a Poisson Regression to model the counts of the gene expression over all 250 top genes, treating every gene as an individual task. The Poisson negative binomial regression models were fit using the *statsmodels* package in Python. [15]. However, Poisson regression performed poorly, resulting in data that was heavily skewed towards 0. Considering that Poisson distribution expects equal mean and variance, which

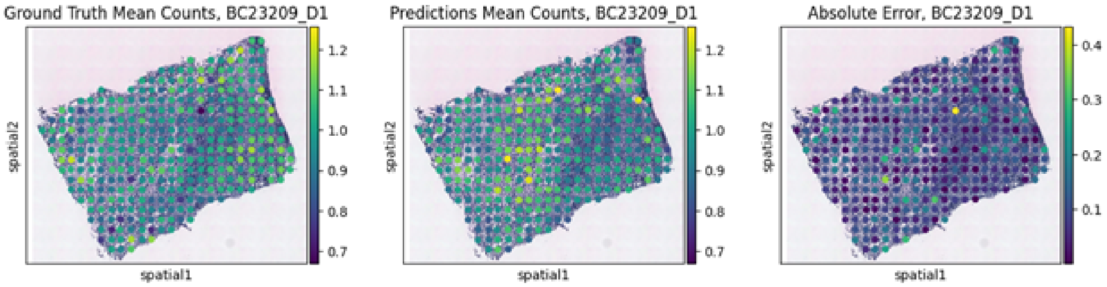

Fig. 3. Spatial gene expression is plotted over the provided histology image for a representative sample. The ground truth (left), predictions (middle), and error (right) are shown.

TABLE I
Modeling Metric Comparison across final model suite (truncated for space)

Name	Split	PCC	SRCC	MSE	MAE	R2
Simple Neural Network	Train	0.3023	0.2827	0.1884	0.3070	0.1016
	Valid	0.0391	0.0430	0.3229	0.3916	-0.1321
	Test	0.0639	0.0738	0.3028	0.3775	-0.0787
Linear Regression Lasso	Train	0.1607	0.1670	0.2344	0.3444	0.0018
	Valid	0.1759	0.1684	0.2751	0.3799	-0.0235
	Test	0.1243	0.1287	0.2881	0.3819	-0.0530
NB Alpha Apriori Estimation	Train	0.2130	0.2596	128.6232	1.0054	-46.1515
	Valid	0.0568	0.0859	8.0991	0.8830	-13.4501
	Test	0.0420	0.0868	26.1523	1.1578	-20.7280
KNN NB Alpha Apriori Estimation	Train	0.2838	0.3363	588.8546	1.0869	-191.3547
	Valid	0.0687	0.0902	6.1521	0.8503	-9.7980
	Test	0.0273	0.0708	49.3884	1.3207	-68.1276
KNN Pooled GNN	Train	0.14297	0.17471	0.2363	0.3433	0.00137
	Valid	-0.0260	-0.0170	0.3036	0.3898	-0.0876
	Test	0.02887	0.00324	0.2960	0.3793	-0.0697
KNN Linear Regression Ridge	Train	0.32554	0.31120	0.1766	0.2994	0.11517
	Valid	0.25974	0.25554	0.2243	0.3503	0.06605
	Test	0.06064	0.06856	0.3600	0.4164	-0.2345

does not hold true for our data, NB distribution, being a two-parameter distribution that accounts for overdispersion in the data, was found to be a better fit.

We then attempted two types of Negative Binomial Regression models: one model that used single-spot image patch embedding, and one that used the KNN pooled image patch embedding as described in Section III. Additionally, we attempted two ways of modeling the negative binomial regression task: by setting the dispersion parameter α constant at 1.0, and by calculating α based on the sample mean and variance of the train set and then fitting the negative binomial regressor using the estimated a-priori α estimate. We see in Table I, that the model a-priori α estimation models had the best performance on the validation set, with the KNN-pooled image embedding models having a higher Pearson correlation coefficient and lower MSE on the validation set.

Despite the NB regression models' ability to handle overdispersion and their increased generalizability compared to Poisson regression, the performance of the models on the validation set was still unsatisfactory. The complexities associated with estimating the α parameter and the inherent assumptions of the NB regression model may be limiting the models' ability to accurately capture the underlying patterns in the gene expression data. Furthermore, it appears that the models are appears that the models are overfitting on the validation set. We hypothesized that a simpler approach using linear regression with log-normalization of the target distribution and regularization could yield better results. Moreover, neural networks also tend to work better with normalized data, so we utilized log-normalization for those as well.

B. Linear Regression on Log-Normalized Target

We compare several linear regression models for log-normalized target gene expression counts, employing both image embedding techniques described in the previous section. The models under consideration include:

- Unregularized regression
- Lasso regularized regression
- Ridge regularized regression

For the regularized regression models (Lasso and Ridge), we employ a grid search to determine the optimal regularization tuning parameter λ for each task, based on a specified range of search parameters. The search range for Ridge regression extends from 0 (unregularized) to 100, while for Lasso regression, it ranges from 0.0000001 (due to solver constraints preventing the use of 0) to 10.

The best models from this suite were the linear regression model with ridge regularization and KNN-pooled image em-

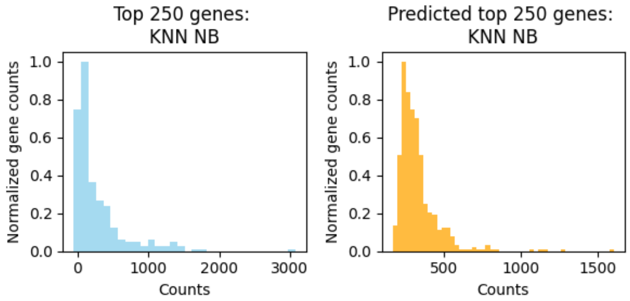

Fig. 4. KNN NB model: Histogram of counts for predicted and ground truth

beddings, and the linear regression model with lasso regularization and single-spot image embeddings.

C. Simple Neural Network

The target for the prediction task for GNN was the 250 gene counts in log-normalized space, and the input involved 64 features for each spot: 32 features from single-spot image patch embedding and the other 32 from single-spot adjacency embedding. For the simple NN using Adam optimizer, we performed grid search over different values of learning rate, namely $[10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}]$, from which 10^{-3} performed comparatively better given the training and validation losses.

D. Graph Neural Network

The target for the prediction task for GNN was the 250 gene counts in log-normalized space, the input involved just 32 features from knn-pooled image patch embedding, and instead of the adjacency embedding we passed an edge list based on the spot adjacencies. For the architecture with two Graph convolutional layers (GCN) and FFN including batch normalization and dropout, we performed grid search over different values of hidden layer size, learning rate, dropout rate and the optimizer. Bigger hidden layer sizes, as well as small dropout showed poor performance and overfit quickly. Higher learning rates resulted in bigger maximum loss. From all tested combinations, we chose Nadam optimizer with 0.001 learning rate, 0.8 dropout and 32 hidden layer size. Adam optimizer also showed similar performance with the same hyperparameters, but Nadam brings the benefit of the accelerated convergence and has less fluctuation in the validation performance. Additionally training involves an early stopping mechanism to prevent overfitting, monitoring validation loss with a patience of 15 epochs, with batch size of 128 over maximum of 100 epochs. The patience, batch size, and epoch parameters were manually curated towards smaller values to outweigh overfitting risk.

E. Quantitative Model Performance

Table I presents a comprehensive comparison of the modeling metrics across different data splits (train, validation, and test) and model suites. The linear regression models with ridge regularization and KNN-pooled image embeddings, as well as the lasso regularization with single-spot image embeddings, emerged as the best-performing models within the linear regression suite. These models achieved the highest PCC and lowest MSE on the validation set, indicating their ability to capture linear relationships between the predicted and actual gene expression values.

The simple neural network (NN) model demonstrated slightly better performance on the validation and test sets compared to the Graph Neural Network (GNN) model. This difference in performance can be attributed to the increased feature space in the GNN model, resulting from the use of an edge list instead of adjacency embeddings.

F. Qualitative Model Performance

For a qualitative perspective on our results, we have plotted some representative measurements for various models from the experiment. Fig. 1 demonstrates the performance of the model measured over different clusters, along with the Leiden clustering results. Fig. 2, shows the differential expression of gene expression for tumor versus non-tumor regions. Although there is no overlap in DEG identities, the prediction based DEG's still show tumor specific overexpression for genes that share biological function with ground truth based DEGs. Finally, the spatial predictions of our model, normalized by the average counts, are shown in Fig. IV-D.

VI. Conclusions and Future Work

In conclusion, our quantitative analysis highlights the potential of integrating histology imaging and spatial transcriptomics data through low-dimensional embeddings for predicting target gene expression. The linear regression models with regularization and the simple NN model showed promising results, outperforming the GNN and NB regression models. However, the moderate performance metrics obtained on the test set underscore the need for further research and development of more sophisticated modeling approaches to fully leverage the information contained in these complex datasets. Additionally, we could standardize our model selection approach across both the linear regression and negative binomial regression model suites by adding regularization and hyperparameter tuning.

Given that all of our models are utilizing only image and adjacency information, we see a big area of improvement in the potential of including gene expression embeddings into the training routine. By employing more advanced network architectures we can combine three data modalities including gene expression, adjacency information and images, and learn a joint latent representation for all the modalities involved. This can be done using simple methods, such as building a gene-by-gene similarity matrix based on shared biological function and co-expression of genes, and use this matrix to build gene embeddings. Or as a more complex option, a graph-autoencoder architecture [5] can be used for this task, and the latent space would allow to not only base our predictions on the assumption that image features are correlated with gene expression patterns, but actually model and learn from that relationship by using the gene expression data.

VII. Contributions

All team members contributed equally to the project, both in terms of amount of work done, and in terms of diversity of work. Susie helped develop the initial project plan, and contributed heavily to the ideation process – leveraging her expertise in the field to help with modeling, and preprocessing. She developed the highly complex GNN model, and provided insights into negative binomial modeling, downstream analysis, and clustering. Lauren drew most of the visualizations used in the paper, and used for the analysis. She also spearheaded the inital data gathering process, and clustering of the data for use in modeling. Additionally, Lauren also helped develop the simple linear regression models. Gaurav implemented the preprocessing steps that were used to feed the modeling process. He also fit the models with KNN-pooled image modeling, and prepare the test results. Additionally, he took charge of creating and maintaing the google cloud resources required to conduct the modeling used in this project.

References

[1] Badea, L., and Stănescu, E. (2020). Identifying transcriptomic correlates of histology using deep learning. PloS one, 15(11)

[2] Zeng, Z. Li, Y. Li, Y. Luo, Y. (2022), Statistical and machine learning methods for spatially resolved transcriptomics data analysis. Genome Biology. 23, 83.

[3] Rao, A. Barkley, G. França, I. Yanai. (2021). Exploring tissue architecture using spatial transcriptomics. Nature. 596, 211-220

[4] Moses, L. Pachter, L. (2022), Museum of spatial transcriptomics. Nature Methods. 19:534-546

[5] Zhang, X., Wang, X., Shivashankar, G.V. et al. Graph-based autoencoder integrates spatial transcriptomics with chromatin images and identifies joint biomarkers for Alzheimer's disease. Nat Commun 13, 7480 (2022). https://doi.org/10.1038/s41467-022-35233-1

[6] B. He, L. Bergenstråhle, L. Stenbeck, A. Abid, A. Andersson, Å. Borg, J. Maaskola, J. (2020), Lundeberg, J. Zou. Integrating spatial gene expression and breast tumour morphology via deep learning. Nature Biomedical Engineering. 4, 827-834

[7] S. Longo, M. Guo, A. Ji, P. Khavari. (2021), Integrating single-cell and spatial transcriptomics to elucidate intercellular tissue dynamics. Nature Reviews Genetics. 22, 627-644.

[8] Stenbeck, Linnea; Bergenstråhle, Ludvig; Lundeberg, Joakim; Borg, Åke (2021), "Human breast cancer in situ capturing transcriptomics", Mendeley Data, V5, 10.17632/29ntw7sh4r.5

[9] Satija, R., Farrell, J., Gennert, D. et al. Spatial reconstruction of single-cell gene expression data. Nat Biotechnol 33, 495–502 (2015). https://doi.org/10.1038/nbt.3192

[10] Bergenstråhle, L.; He, B.; Bergenstråhle, J.; Abalo, X.; Mirzazadeh, R.; Thrane, K.; Ji, A.L.; Andersson, A.; Larsson, L.; Stakenborg, N.; Boeckxstaens, G.; Khavari, P.; Zou, J.; Lundeberg, J.; Maaskola, J. (2021). Nature Biotechnology. 40, 476-479.

[11] Song, J.; Lamstein, J.; Ramaswamy, V.; Webb, M.; Zada, G.; Finkbeiner, S.; Craig, D. (2024). Enhancing Spatial Transcriptomics Analysis by Integrating Image-Aware Deep Learning Methods. Pacific Symposium on Biocomputing.

[12] Petukhov, V.; Xu, R.; Soldatov, R.; Cadinu, P.; Khodosevich, K.; Moffitt, J.; Kharchenko, P. (2022). Nature Biotechnology. 40, 345-354.

[13] Tian, L.; Chen, F.; Macosko, E. (2023). The expanding vistas of spatial transcriptomics. Nature Biotechnology. 41, 773-782.

[14] Halko, et al. Finding structure with randomness: Stochastic algorithms for constructing approximate matrix decompositions. SIAM Rev., Survey and Review section, Vol. 53, num. 2, pp. 217-288, June (2011)

[15] Josef Perktold, "statsmodels/statsmodels: Release 0.14.1". Zenodo, Dec. 14, 2023. doi: 10.5281/zenodo.10378921.

[16] Traag, V.A., Waltman, L. and van Eck, N.J. From Louvain to Leiden: guaranteeing well-connected communities. Sci Rep 9, 5233 (2019)

[17] McInnes and Healy, UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction, arXiv (2018)