Bioinformatics 593 Special Topics Projects, 2023, pp. 1–7

doi:
Advance Access Publication Date: December 2023
Paper

PAPER

Breast Tissue Metastaticity Prediction using Spatial Transcriptomics and Gene Expression

Monica Bonilla,[1,*] Douglas Maldonado-Torres,[1,*] Michael Akpabey,[1,*] Sarah Kang,[1,*] Jenni Liu[1,*] and Jan Rosa[1,*]

[1]Department of Computational Medicine and Bioinformatics, University of Michigan, 100 Washtenaw Ave, Ann Arbor, 48109, Michigan, United States of America
[*]To whom correspondence should be addressed.

Abstract

This research project aims to address the challenges associated with predicting breast cancer metastatic potential by proposing an innovative approach that utilizes the spatial location and gene expression components of spatial transcriptomic data through a Graph Convolutional Network (GCN) model. The study utilizes datasets from the 10x Visium Spatial Transcriptomics dataset, including breast cancer and normal tissue samples, which were preprocessed and annotated. The GCN model is designed to classify metastatic potential based on an extracellular matrix (ECM) signature derived from previous literature comprising ESR1, TP53, NF1, AKT1, KMT2C, and PTEN genes. The methodology involves creating a graph from the spatial data, generating an adjacency matrix, and converting the graph into an interpretable model input. The GCN architecture consists of three convolutional layers, and the model undergoes training and testing with an Adam optimizer. Initial results show promising accuracy after adjusting for data imbalance with an area under receiver operating characteristic curve (AUC) of 0.91. The study concludes by acknowledging limitations, such as the sparse availability of breast cancer transcriptomic datasets, and suggests future steps, including validation on external datasets and exploring pooling modalities for improved generalizability.

Key words: Graph Convolution Networks (GCN), Spatial Transcriptomics, Metastasis, Breast Cancer

Introduction

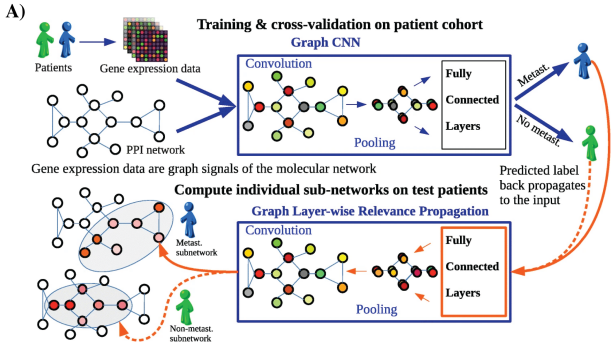
Breast Cancer (BC) continues to be a traumatizing experience for many women across the United States and the world over. In the US, the average percent risk of a woman developing breast cancer in her lifetime is as high as 13%. Of greater concern is that 30% of the total yearly diagnoses of new cancer cases will be of breast cancer.[1] Suffice to say, the overall impact of breast cancer on women in particular remains devastatingly large. As with all cancers, early detection is key to treatment of BC. Although breast cancer has a relatively high rate of long-term survival, 90.8% 5-year relative survival, this drops steeply once the cancer metastasizes (31%)[2]. Previous work has identified some genetic trends that dictate key development patterns of breast cancer cells, including genetic markers that are predictive of BC metastasis. These markers include ESR1, TP53, NF1, AKT1, KMT2C and PTEN[3]. In this project, we set out to build a model that could predict metastatic potential based on the expression of this metastatic gene signature, alongside their spatial localization, using a machine learning (ML) model with Graph Convolutional Network architecture (GCN).

Previous ML models have been proposed for a similar task, such as Graph Neural Networks (GNN) in order to predict breast cancer metastatic potential based on gene expression. For example, Chereda et. al.[4] took a more macroscopic approach to BC metastatic classification utilizing a GNN framework that classified patients based on metastatic potential. Their goal was to predict the occurrence of distant metastasis using the expression data of various patients and explain said prediction. They made a graph input of an undirected weighted graph $G = (V, EA)$, where V, E, and A are the vertices, edges, and adjacency matrix respectively. The vertices and edges of the graph were adapted from the Human Protein Reference Database (HPRD) protein-protein interaction (PPI) network. They utilized a breast cancer patient dataset available from the Gene Expression Omnibus (GEO) repository and performed some exploratory analysis and data pre-processing. Post pre-processing, the dataset contained 12,1769 genes in 969 patients. Of which 393 classified into having distant metastasis and 576 without metastasis concluding a follow-up appointment in the last 5-10 years of data collection. The genes were mapped to the PPI vertices and resulted in a main connected component of 6888 vertices (mapped genes) ; all other connected components were discarded for this model. The adjacency matrix would be of $m \times m$ dimensionality and for the unweighted HPRD PPI network the matrix had only 0s and 1s.

At its core, the task set out was a binary classification of expression data $X \in R^{nxm}$ a target variable $Y \in \{0,1\}^n$. With n being the number of patients and m being the number of genes. A row of X contains data from a single point (the patient) and was mapped to the vertices of graph G which acts as the graph signal. Thus, the end goal for the researchers would be to explain the prediction of metastasis of a BC patient by providing molecular sub-networks for each patient. A diagram showing the workflow of this project is provided by the researchers, and now us, for the sake of aiding the explanation of what they set out to achieve (Fig. 1a).

Validation was performed on expression data from human umbilical vein endothelial cells (HUVECs) that had/had not undergone tumor necrosis factor (TNF) alpha treatment. This connected component resulted in 7798 genes in the main connected component. Regardless, the results of the 10 fold cross validation of the prediction task done by the GCNN prior and post normalization are shown in Figure 1b.

A)



B)

Method	Std	100*AUC	Accuracy, %	F1-weighted, %
Graph-CNN	-	82.57 ±1.25	76.07 ±1.30	75.82 ±1.33
Random Forest	-	81.27 ±1.66	74.23 ±1.73	73.47 ±1.84
Graph-CNN	+	82.16 ±1.25	76.18 ±1.36	75.86 ±1.35
Random Forest	+	81.40 ±1.76	74.74 ±1.67	74.00 ±1.82
glmgraph	+	80.88 ±1.37	75.14 ±1.30	74.73 ±1.39

Fig. 1. A) Each PPI network would be populated with each individual patient's expression data. The ML architecture would have each graph exist as a connected layer and would use the information of all layers to make a determination of metastatic potential. The researchers went further beyond and would create subgraphs that would explain the results of said prediction. B) Results of Graph CNN prediction on metastatic events. (Std is the standardization of features)

From the author's own admittance the changes of prior knowledge on the results of GCNN for this specific PPI network based framework still need to be investigated as studies have shown that MNIST images of a random connecting pixel affects GCNN performance significantly[5]. And although their own model didn't see performance effects on the classifier when permuting the PPI network, the network itself is small and implies high degrees of connectivity between vertices (¡ 6 vertices of distance between two nodes). Most other results were related to subtype generation and were of little relevance to our current paper, but it is worth noting that their created subgraphs would contain common potential oncogenic genes for patients that would be classified as metastatic. This is a single example; but this framework appears to be a basic standard in many other GNNs utilized for cancer classification (not just necessarily metastatic potential classification). In many others the same advantages and shortcomings are echoed throughout.[6]

Spatial transcriptomics has contributed leaps and bounds to advanced studies related to developmental biology and the study of the tumor microenvironment, adding the additional context of the spatial relationships and interactions of cells to one another. Thus our choice to take advantage of the new information provided by spatial transcriptomic technologies to approach the BC metastatic potential classification task was a logical next step in trying to improve model performance. This integration of expression-based and spatially-based features could serve as a means of overcoming the pitfalls of utilizing a PPI network as the base of the graph input. It may also offer some other means of reducing the effects of noise (a flawed PPI network) while also providing the robust and accurate learning offered by a GCN, as it learns from trends of its multiple layers.

Thus for our project, we propose the following innovation. A Graph Convolutional Network (GCN) classifier of metastatic potential that is able to utilize both the gene expression and spatial location information present in spatial transcriptomic data collected from breast cancer tissue to predict cell metastatic potential.

Approach

Our general initial approach is as follows. We utilized publicly available human breast cancer data collected from the 10x Visium Spatial Transcriptomics platform. This dataset should be split into roughly 80 % training and 20% testing to establish model performance. After tuning our hyperparameters until a sufficient level of accuracy is achieved (70% minimum) we aim to test our model on two new datasets. One of which should a healthy human breast tissue spatial transcriptomics dataset, pre-processed in R and converted to a .h5ad file so it can be used for model testing in python. The other validation dataset should be a new breast cancer dataset. This is done to further prove our model can accurately predict enrichment of metastatic gene signature in diseased tissue and is able to discern healthy tissue. The proposed model is a GCN that predicts a binary classification of metastatic potential from the metastatic gene signature expression and spatial x, y coordinates of the spatial transcriptomics data.

Methods

Datasets

The initial breast cancer dataset was acquired from 10x Genomics[7]. The downloaded dataset contained information from 1 slide of a FFPE BC tissue block containing 2 Ductal Carcinoma In Situ (DCIS) regions. Additional Visium datasets, both breast cancer and healthy were acquired from a recently published Human Breast Cell Atlas, downloaded at cellxgene.cziscience.com[8]. The data was loaded as an AnnData object and we confirmed the annotation of cell types present. The listed cell types were the following ST1-Adipo, ST2-B, ST3-Fibro,ST4-LumHR, ST5-LumSex, ST6-LumSec/Basal, ST7-Lymphatic, ST8-Vas/Peri, ST9-Vascular. Simple mitochondrial pre-processing was performed on the tissue samples before continuing downstream analysis. Counts were checked and outliers were filtered accordingly. We also checked to make sure that there only existed a single desired disease type "Breast Cancer" and a non-disease state "Normal" for the according datasets.

Preprocessing

We loaded in the filtered feature BC spatial data from the h5 file, then assessed count distribution throughout the tissue slice since variance in molecular counts across the tissue is common in spatial data. This can be due to cell density differences in spots. To address this heterogeneity, we performed normalization, with SCTransform "which builds regularized negative binomial models of gene expression in order to account for technical artifacts while preserving biological variance"[9,10] (Fig 2). Then we ran the standard

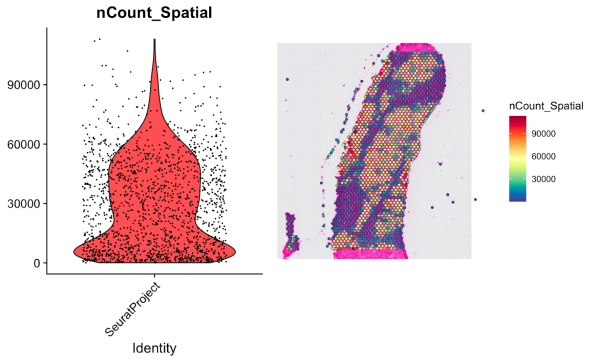


Fig. 2. Normalization of spatial count data. The left hand side depicts a violin plot of total transcript counts of each spot across the entire sample tissue. The right hand side overlays these counts with a spatial representation of the tissue.

Seurat pipeline for PCA and UMAP dimensionality reduction and clustering that mirrors scRNA-Seq[9] (Fig 3). Next, to

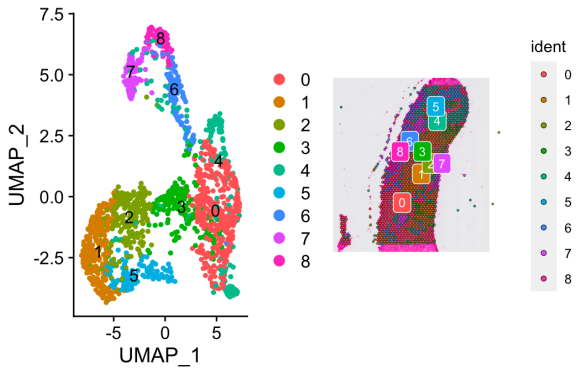


Fig. 3. Initial UMAP clustering of data and spatial representation of corresponding clusters

annotate cluster markers from previous literature, cell2.0, and pangloDB were utilized. Feature maps assessing key gene expression on a per cluster basis were also used to aid in assessment (Fig 4). This allowed us to parse immune cells, cancer cells, and non-malignant cells with high confidence. Markers for T-cells, fibroblasts, basal cells, and basal cells with a high ECM potential were assessed as well.

Further, we examined the most variable spatial features in each cluster with the FindSpatiallyVariable function with the *moransi* statistics parameter (Fig 5), followed by the FindMarkers function to "compare each cluster against all

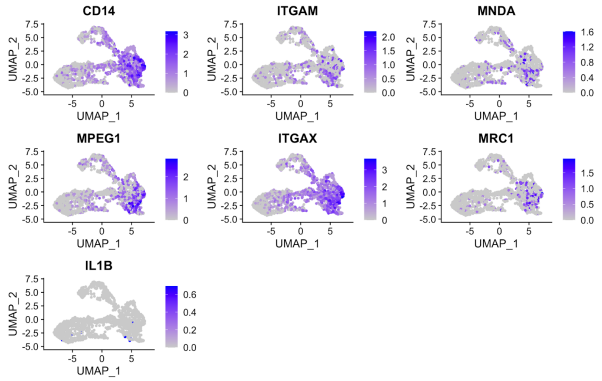


Fig. 4. Feature plots showing expression of known myeloid cell markers across all UMAP clusters.

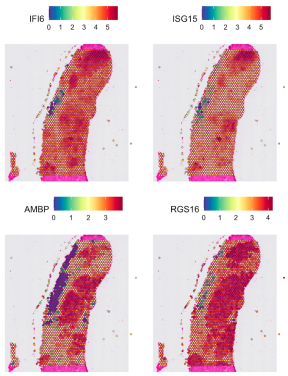


Fig. 5. Expression levels of top 4 features exhibiting spatial patterning, identified through the FindSpatiallyVariable function.

other clusters to identify potential marker genes unique to each cluster." Finally the following annotations were added to the Seurat object and it was saved in a python compatible .h5ad format (Fig 6).

0 = T Cell
1 = Cancer Cell
2 = Basal Cell
3 = Fibroblast
4 = T Cell
5 = Basal Cell
6 = Myeloid
7 = Myeloid
8 = Myeloid

After pre-processing, our dataset consisted of 1647 total spots, 268 of which were annotated as cancer cells and 1389 of which were not.

Creation of the Graph and Adjacency Matrix

With pre-processing done the graph input could be created. This is a valuable step as we know that a graph neural network requires an adjacency matrix in order to properly give us our prediction. The .h5ad file was loaded into python as an AnnData object. The coordinates for the spot positions were loaded and then added to the obs of the AnnData object. Finally the annotations and spatial data were loaded. With all these elements loaded

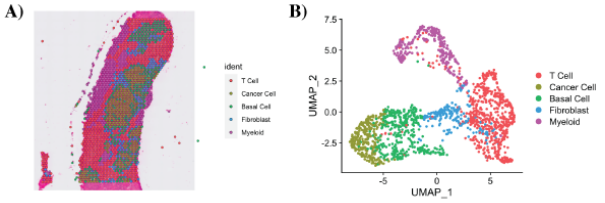


Fig. 6. A) Final cell type annotations depicted in the spatial slide representation of the data. B) Final cell type annotations depicted in the UMAP representation of the data.

a new AnnData object was created with the following obs: '*orig.ident*', '*nCount_Spatial*', '*nFeature_Spatial*', '*nCount_SCT*', '*nFeature_SCT*', '*SCT_snn_res.0.5*', '*seurat_clusters*', '*manual_clusters*', '*array_row*', '*array_col*', '*clusters*'. Now when plotting the UMAP we were able to visualize the clusters with proper annotations (Figure 7).

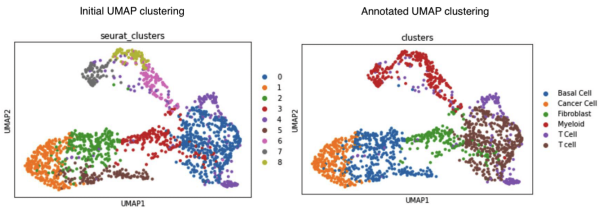


Fig. 7. UMAP clustering before and after annotation of the dataset.

The last addition to our object would be to create binary classifications wherein cancer cells are 1 and all other cells are 0. We started by creating an expression_matrix and a boolean mask indicating which genes were in our ECM gene signature list (ESR1, TP53, NF1, AKT1, KMT2C, PTEN). The AnnData object was subset based on the boolean mask. Finally from the subset object, we created a $n \times 6$ matrix, where n is the total number of spots in the data (Figure 8).

	0	1	2	3	4	5
0	3.663562	1.609438	0.000000	2.639057	2.079442	0.693147
1	3.737670	1.098612	0.693147	2.833213	1.945910	0.693147
2	3.737670	2.079442	0.000000	2.772589	2.079442	0.000000
3	3.663562	1.386294	0.000000	2.708050	1.098612	0.693147
4	3.218876	2.079442	0.000000	2.397895	2.302585	0.000000
...	...	...	...	...	...	...
1652	3.332205	0.693147	0.693147	2.944439	1.609438	0.000000
1653	3.258097	1.386294	1.386294	2.302585	1.098612	0.000000
1654	3.526361	1.609438	0.000000	2.708050	1.098612	1.098612
1655	3.367296	1.098612	0.693147	2.833213	1.945910	0.693147
1656	3.688879	1.609438	0.693147	2.995732	1.386294	0.693147

Fig. 8. The $n \times 6$ ECM gene signature matrix

Using scipy.spatial KDTree we created our graph based on euclidean distance and gene expression correlation. Each node in the graph represents one spot in the dataset. Neighbors were determined by the distances between nodes, where a distribution was created by calculating the euclidean distances of each node to every other node in the graph. The 2.5^{th} percentile of this distribution was set as the distance threshold, with a value of 5.6. Using the generated ECM signature gene matrix, we calculated the Pearson's correlation according to the following equation using the .corr() function, which computes pairwise correlations of columns. We set our correlation threshold to 0.985.

$$r = \frac{\sum(x_i - \hat{x})(y_i - \hat{y})}{\sqrt{\sum(x_i - \hat{x})^2(y_i - \hat{y})^2}} \tag{1}$$

r = correlation coefficient
x_i = values of the x-variable in a sample
$\hat{x}$ = mean of the values of the x-variable
y_i = values of the y-variable in a sample
$\hat{y}$ = mean of the values of the y-variable

To create the graph, we began by considering neighbors within the distance threshold (5.6) and then only adding edges with neighboring nodes if they had a high similarity in gene expression, surpassing the correlation threshold (0.985). Thus, if two nodes managed to cross both thresholds, an undirected edge was added to the graph. The resulting graph representation of the data is shown in Figure 9.

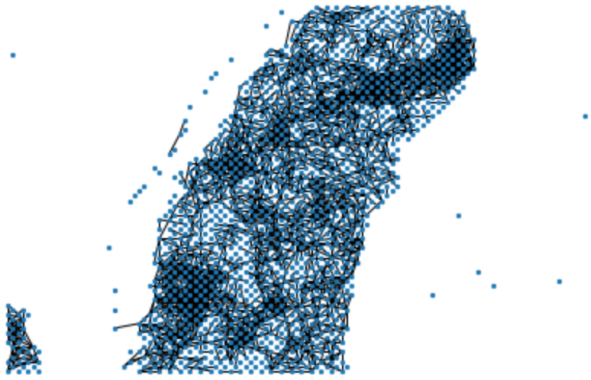


Fig. 9. The created graph representation of the dataset, which looks similar to the spatial layout of the data on the slide.

Converting the Graph into Interpretable Model Input

The graph edges were converted to a list and then a python tensor. Node features were created by the adata object's array_row and array_col obs. The ECM matrix was also converted into a tensor and concatenated with node features as a single tensor. The binary labels of the adata obj were made into a tensor as well.

GCN Model, Architecture, and Evaluation

The GCN model consists of 3 convolution layers. It receives 8 features - consisting of spatial x, y coordinates and expression of our 6 metastatic genes - and our 2 binary classes [0, 1] as its two primary inputs. It increases the 8 initial features to 32 features and then 64 features through these 3 layers. These 64 features are then reduced back down to the prediction of the 2 binary classes. On the output of the final convolutional layer

we apply the following log_softmax operation. A scheme of said model can be seen in Figure 10. In the softmax function x is the output data and the operation is applied across the columns of the output.

$$log(\frac{exp(x)}{\sum exp(x)} = x - log(\sum exp(x))) \tag{2}$$

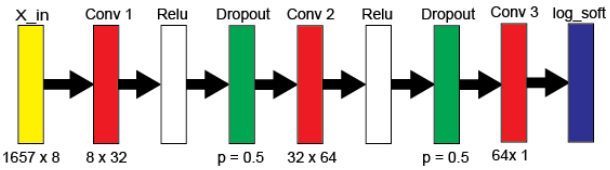


Fig. 10. GCN Network Schemata. Conv = Convolutional Layer. Log_soft = Log_softmax operation. In total, there are 3 convolutional layers with dropout and ReLU activation in between each one. The log_softmax operation is applied at the very end.

Training and Testing

Data was split into 80% training and 20% testing sets, stratified across classes. The model was trained using an Adam optimizer algorithm[1]1 (Figure 11). The choice of utilization of this specific algorithm was its utility when working with sparse data such as ours given we only had two BC slides to train our model with. The utilized learning rate and weight decay were 0.0001 and 1e-4 respectively. The model underwent 25000 epochs of

Algorithm 1: *Adam*, our proposed algorithm for stochastic optimization. See section 2 for details, and for a slightly more efficient (but less clear) order of computation. g_t^2 indicates the elementwise square $g_t \odot g_t$. Good default settings for the tested machine learning problems are $\alpha = 0.001$, $\beta_1 = 0.9$, $\beta_2 = 0.999$ and $\epsilon = 10^{-8}$. All operations on vectors are element-wise. With β_1^t and β_2^t we denote β_1 and β_2 to the power t.

Require: α: Stepsize
Require: $\beta_1, \beta_2 \in [0, 1)$: Exponential decay rates for the moment estimates
Require: $f(\theta)$: Stochastic objective function with parameters θ
Require: θ_0: Initial parameter vector
  $m_0 \leftarrow 0$ (Initialize 1st moment vector)
  $v_0 \leftarrow 0$ (Initialize 2nd moment vector)
  $t \leftarrow 0$ (Initialize timestep)
  while θ_t not converged **do**
    $t \leftarrow t + 1$
    $g_t \leftarrow \nabla_\theta f_t(\theta_{t-1})$ (Get gradients w.r.t. stochastic objective at timestep t)
    $m_t \leftarrow \beta_1 \cdot m_{t-1} + (1 - \beta_1) \cdot g_t$ (Update biased first moment estimate)
    $v_t \leftarrow \beta_2 \cdot v_{t-1} + (1 - \beta_2) \cdot g_t^2$ (Update biased second raw moment estimate)
    $\widehat{m}_t \leftarrow m_t / (1 - \beta_1^t)$ (Compute bias-corrected first moment estimate)
    $\widehat{v}_t \leftarrow v_t / (1 - \beta_2^t)$ (Compute bias-corrected second raw moment estimate)
    $\theta_t \leftarrow \theta_{t-1} - \alpha \cdot \widehat{m}_t / (\sqrt{\widehat{v}_t} + \epsilon)$ (Update parameters)
  end while
  return θ_t (Resulting parameters)

Fig. 11. ADAM algorithm pseudocode implementation[6]

training. Loss, accuracy, and epoch number were stored for each iteration.

Loss Calculation and Accuracy Validation

Given that we would be assessing a binary classification it was thought ideal to simplify the process of validation with the negative log likelihood of loss to assess performance. Consequently, a sigmoid activation serves as the ideal to map values from $(-\infty, \infty)$ to [0,1].

$$\sigma = \frac{1}{1 + exp(-z)} \tag{3}$$

Likelihood in our case is defined as

$$P(D|\theta) = \prod_{i=1}^{n} \hat{y}_{\theta,i}^{y_i} (1 - \hat{y}_{\theta,i})^{1-y_i} \tag{4}$$

With log-likelihood[7]

$$logP(D|\theta) = \sum_{i=1}^{n} (y_i log(\hat{y}_{\theta,i}) + (1 - y_i) log(1 - \hat{y}_{\theta,i})) \tag{5}$$

y_i = the predicted probability of the data point i being positive
$(1 - y_i)$ = the predicted probability of the i^th data point being negative[7]

Since the log function is monotonic we minimize the loss by taking values from the negative log likelihood.

$$l(\theta) = -\sum_{i=1}^{n} (y_i log(\hat{y}_{\theta,i}) + (1 - y_i) log(1 - \hat{y}_{\theta,i})) \tag{6}$$

We calculate accuracy using the following formula

$$accuracy = \frac{correct}{total} \tag{7}$$

Where *correct* is obtained by summing the number of predicted items that match the masked train data.

Results

We begin by presenting the initial results of our training loss and accuracy, both of which were deemed acceptable at 0.256 and 0.892 respectively (Figure 12). Testing accuracy was 0.871.

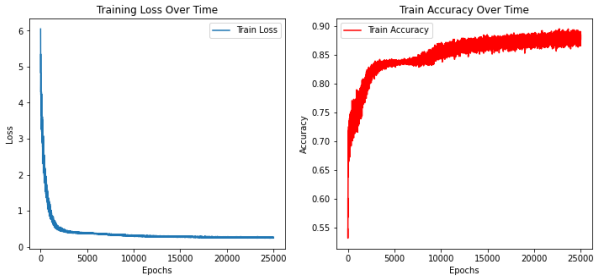


Fig. 12. Training loss (left) was 0.256 and training accuracy (right) was 0.892.

When examining the individual class accuracies we found that our model was much better at correctly classifying benign cells relative to cancer cells (Table 1). Class 0, as mentioned prior, is the determination that a cell is benign, and class 1 is the determination that the cell has metastatic potential. Based on the resulting accuracy measures, we believed that the model was not performing as was idealized. To which we consulted with Dr. Jie Liu to consider the best course of action. Based on the fact that our data was sparse and imbalanced, containing more benign spots than cancer spots and few spots overall, he recommended we integrate an ROC curve as an additional means of evaluation (Figure 13). The resulting AUC (0.91) of our model, gives us great confidence in its performance.

Table 1. Percent Prediction Accuracy per Class and Total

Class	Accuracy
Class 0	96.47
Class 1	50.75
Total	89.08

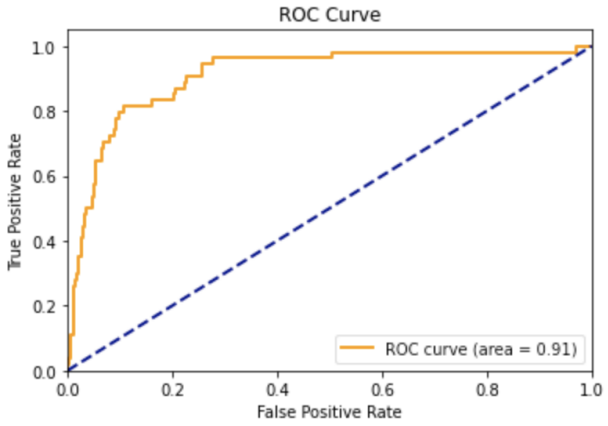


Fig. 13. ROC with AUC of .91

Discussion

Based on the results, we can state with confidence that the implementation of our model was of great success. This is given that our model achieves high levels of true positive rates relative to false positive rates at every time point. It's worth noting that our model achieves a true positive rate greater than 80% at the 0.2 threshold which is quite promising and further incentivizes testing on other BC transcriptomic datasets. Compared to the initial model proposed by Chereda et. al. we achieve a higher level AUC accuracy in our binary classification task of metastatic potential. However, we are unable to definitively conclude that ours has better performance given the lack of testing on additional independent BC datasets.

This was a very cutting edge project that took significantly more time and resources than what was initially anticipated. There were several limitations and problems that arose because of the innovativeness of such an implementation that effectively limited how well we can conclude our model to be generalizable across other datasets and/or if it can hold practical value in a clinical setting. Most importantly, this comes in regards to the availability of bc transcriptomic slides. We wanted to validate our model on an external transcriptomics dataset to make sure our model is representative across generalized tissue slides. Unfortunately there aren't many more publicly available BC datasets right now. The one we found and tried to use for additional testing, as mentioned in our initial approach[8], has a very different format from our inital BC dataset (working with transcripts and EMSEMBL IDs instead of gene symbols). Alhough we tried mapping these IDs to their respective human gene symbol, using the biomaRt package in R, the lack of one to one mapping made it difficult to achieve the same data preprocessing and annotation standards used for our initial training dataset.

Our choice of using an ROC curve to measure the accuracy of our classifications was motivated because of the data imbalance. Traditional accuracy measurements didn't work presumably because levels of expression are a gradient rather than a uniform expression that crosses well defined thresholds that could be evaluated easily. If we had more tissue samples we could potentially compare expression levels across tissues and through pooling do some proper validation taking into account relative expression levels of multiple tissue samples and getting some better idea of what "average" level will be most associated with metastasis.

Based on all these factors, and the sparsity of our data, we believe that future steps would be to find more BC tissue transcriptomics datasets that could be tested against to prove that our model is generalizable with similar performance across multiple BC slides. Furthermore, we believe that with more BC tissue slides we can implement a pooling modality (akin to Chereda, et al.) that uses information from multiple layers (ie. slides) of breast cancer tissue to make the final determination of metastatic potential.

Acknowledgements

M.B., D.M., M.A., S.K., J.L., and J.R. for planning the project proposal. M.B. for the project idea, finding the ECM signature, and pre-processing and annotation of the data. D.M., M.A., for implementation, optimization, and training of the GCN model and creation of the graph. J.R. and J.L. for researching and investigating other potential models to approach our task. J.L. for finding additional datasets. J.L. and S.K. for contributions to data pre-processing and attempts to work with additional datasets through gene ID mapping. J.R. and S.K. for writing and preparing the manuscript. M.B., D.M., M.A., J.L. for editing the manuscript.

Special thanks to Dr. Josh Welch and Dr. Jie Liu for their support in the successful completion of this project and the consultation they provided. Without their help we would not have overcome various pitfalls and setbacks.

Funding

This work has been supported by no one in particular.

Bibliography

1. Breast Cancer Facts and Statistics 2023. https://www.breastcancer.org/facts-statistics.
2. National Cancer Institute. (n.d.). Cancer Stat Facts: Female Breast Cancer. SEER. https://seer.cancer.gov/statfacts/html/breast.html
3. Angus, L., Smid, M., Wilting, S.M. et al. The genomic landscape of metastatic breast cancer highlights changes in mutation and signature frequencies. Nat Genet 51, 1450–1458 (2019). https://doi.org/10.1038/s41588-019-0507-7
4. Chereda, H. et al. Explaining decisions of graph convolutional neural networks: patient-specific molecular subnetworks responsible for metastasis prediction in breast cancer. Genome Med. 13, 42 (2021).
5. Defferrard, M., Bresson, X., and Vandergheynst, P. Convolutional Neural Networks on Graphs with Fast Localized Spectral Filtering. arXiv.org https://arxiv.org/abs/1606.09375v3 (2016).

6. Alharbi, F. and Vakanski, A. Machine Learning Methods for Cancer Classification Using Gene Expression Data: A Review. Bioengineering 10, 173 (2023).
7. Janesick, A., Shelansky, R., Gottscho, A. D., Wagner, F., Rouault, M., Beliakoff, G., de Oliveira, M. F., Kohlway, A., Abousoud, J., Morrison, C. A., Drennon, T. Y., Mohabbat, S. H., Williams, S. R., 10x Development Teams, and Taylor, S. E. B. (2022). High resolution mapping of the breast cancer tumor microenvironment using integrated single-cell, spatial, and in situ analysis of FFPE tissue. bioRxiv, 2022.10.06.510405. https://doi.org/10.1101/2022.10.06.510405
8. Kumar, T., Nee, K., Wei, R. et al. A spatially resolved single-cell genomic atlas of the adult human breast. Nature 620, 181–191 (2023). https://doi.org/10.1038/s41586-023-06252-9
9. Kingma, D. P. and Ba, J. Adam: A Method for Stochastic Optimization. Preprint at http://arxiv.org /abs/1412.6980 (2017).
10. Hafemeister, C., Satija, R. Normalization and variance stabilization of single-cell RNA-seq data using regularized negative binomial regression. Genome Biol 20, 296 (2019). https://doi.org/10.1186/s13059-019-1874-1
11. Satija Lab. (n.d.). Analysis, visualization, and integration of spatial datasets with Seurat. Seurat. https://satijalab.org/seurat/articles/spatial_vignette.html#data-preprocessing
12. Lau, R. Cross-Entropy, Negative Log-Likelihood, and All That Jazz. Medium https://towardsdatascience.com/cross-entropy-negative-log-likelihood-and-all-that-jazz-47a95bd2e81 (2022).