Cell–ECM Graphs: A Graph-Based Method for Joint Analysis of Cells and the Extracellular Matrix

M. Ghafoor[a,c], JE. Parkinson[b,c], T. Pham[e], TE. Sutherland[d], M. Rattray[a]

a. Division of Informatics, Imaging and Data Sciences, Faculty of Biology, Medicine and Health, University of Manchester, Manchester M13 9PL, UK.

b. Lydia Becker Institute of Immunology and Inflammation, Faculty of Biology, Medicine and Health, University of Manchester, Manchester, M13 9PL, UK

c. Wellcome Centre for Cell-Matrix Research, Faculty of Biology, Medicine and Health, University of Manchester, Manchester, M13 9PL, UK

d. Institute of Medical Sciences, School of Medicine, Medical Sciences and Nutrition, University of Aberdeen, Aberdeen, AB25 2ZD, UK

e. Department of Pharmacology and Therapeutics, University of Liverpool, Liverpool, UK, L69 7ZX

Corresponding authors: magnus.rattray@manchester.ac.uk, mohamed.ghafoor@postgrad.manchester.ac.uk

Keywords: Cell graphs, Cell-ECM graphs, Cell-ECM Interactions, Extracellular Matrix, Matrisome

Abstract

Spatial proteomics technologies enable in situ characterization of both cells and the extracellular matrix (ECM), yet methods to jointly analyse their interactions remain limited. Here, we present a computational framework for constructing Cell–ECM graphs from spatial proteomics data, representing cells and ECM clusters as vertices and encoding cell–cell, ECM–ECM, and cell–ECM interactions as edges within a

unified graph. This framework enables the application of established graph analytical methods to matrix biology, including node classification, unsupervised niche discovery, interaction analysis and whole-graph classification with explainable graph neural networks. Using both synthetic and real data, we show that Cell–ECM graphs capture alterations in ECM and cell–ECM interactions that are not resolved by traditional cell-only graphs. To promote accessibility and reproducibility, we provide an open-source Python package implementing the method, enabling its broad application to spatial proteomics studies of the ECM.

Background

The extracellular matrix (ECM) is a three-dimensional, non-cellular scaffold present in all tissues. It is constructed from proteins, glycoproteins, glycosaminoglycans, and proteoglycans. At the cellular level, the ECM participates in cell adhesion, migration, proliferation and differentiation, while at the organ level it influences biomechanical properties such as tensile strength and elasticity [1]. The importance of the ECM is highlighted by the spectrum of severe genetic disorders caused by mutations in ECM related genes [2].

Graphs offer a mathematical framework for representing complex systems as a set of vertices and edges between them. In the context of biology, vertices can represent entities such as cells, proteins, or genes, and edges can encode relationships such as spatial proximity, molecular interactions, or functional similarity [3]. This representation is highly flexible: different types of vertices and edges, each with their own features, can be incorporated to model multi-scale biological systems, from molecular networks to whole-tissue architectures. Once an appropriate graph representation has been constructed, a wide range of analytical approaches from

graph theory and graph machine learning can be applied, including network statistics, community detection, link prediction, and graph neural networks (GNNs).

Advances in imaging-based spatial proteomics, such as Imaging Mass Cytometry (IMC), now enable the simultaneous measurement of ~40 protein markers in intact tissue sections, capturing both cellular and ECM components. Several graph-based computational methods have been developed for the analysis of cells in such data, such as spatial community analysis [4], cellular neighborhood analysis [5] and interaction analysis [6]. In contrast, there is a lack of graph based computational tools for ECM analysis, despite its inherent network-like organisation making it well suited to graph-based representation.

Our work builds on the concepts of cell graphs [7,8] and ECM-aware cell graphs [9]. Cell graphs model the structural organization of tissues using only cellular vertices, while ECM-aware cell graphs extend this by annotating cells according to the composition of their surrounding ECM in haematoxylin and eosin images. Our method advances these ideas by representing ECM clusters as independent vertices with their own connectivity. This design enables direct analysis of ECM structure and its relationships to cells in higher-dimensional spatial proteomics data.

Here, we present Cell–ECM Graphs, a unified graph representation that integrates the spatial distribution of both cells and ECM in spatial proteomics. This design allows downstream analysis to be performed on the complete graph or separately on its cellular or ECM components. We demonstrate the method on synthetic and real datasets, applying GNNs with explainability techniques to identify key interactions, exploring cell-ECM immune interaction niches and showing improved node classification and spatial clustering compared to conventional cell graphs.

Results and Discussion

Overview of Cell-ECM Graphs

The Cell–ECM Graph framework integrates cellular and ECM components into a unified graph, enabling explicit modelling of cell–cell, ECM–ECM, and cell–ECM interactions (**Fig. 1**). A graph is defined as is $G = (V, E)$ where V is the set of vertices and E the set of edges (also referred to as nodes and links respectively).

In the cell graph, each segmented cell is represented as a vertex, labelled with its cell type. Edges are formed between the $N_{edge}(G)$ nearest neighbouring cells based on distances, with any connections exceeding the maximum cell–cell distance $D_{max}(CC)$ removed. This parameter defines the spatial scale of cell–cell interactions.

The ECM graph is generated by selecting only ECM marker channels from the image and dividing the tissue into P_{size} x P_{size} patches. Each patch is represented by an N -dimensional vector, where N is the number of ECM markers, containing the mean intensity of each marker within the patch. Patches are clustered into K_{ECM} clusters using K-means clustering. A background cluster (cluster 0), containing patches with little or no ECM signal, is removed. The remaining patches become ECM vertices, labelled by their cluster ID. Edges are drawn between neighbouring ECM vertices using $N_{edge}(G)$ nearest neighbours, representing ECM–ECM connectivity.

The Cell-ECM graph is formed by merging the cell and ECM graphs. Cell-ECM edges are added by connecting each cell vertex to $N_{edge}(G)$ nearest ECM vertices within the maximum cell–ECM distance $D_{max}(CE)$.

Several experiments were performed to test the effect of the Cell-ECM graph hyperparameters. The effects of modifying P_{size} and K_{ECM} are visualized in (**Fig. S1**). The silhouette score was used to select the optimal K_{ECM} as 4 and P_{size} as 5. The optimal $D_{max}(CC)$ and $D_{max}(CE)$ were selected based on the results of Bayesian optimization during node classification.

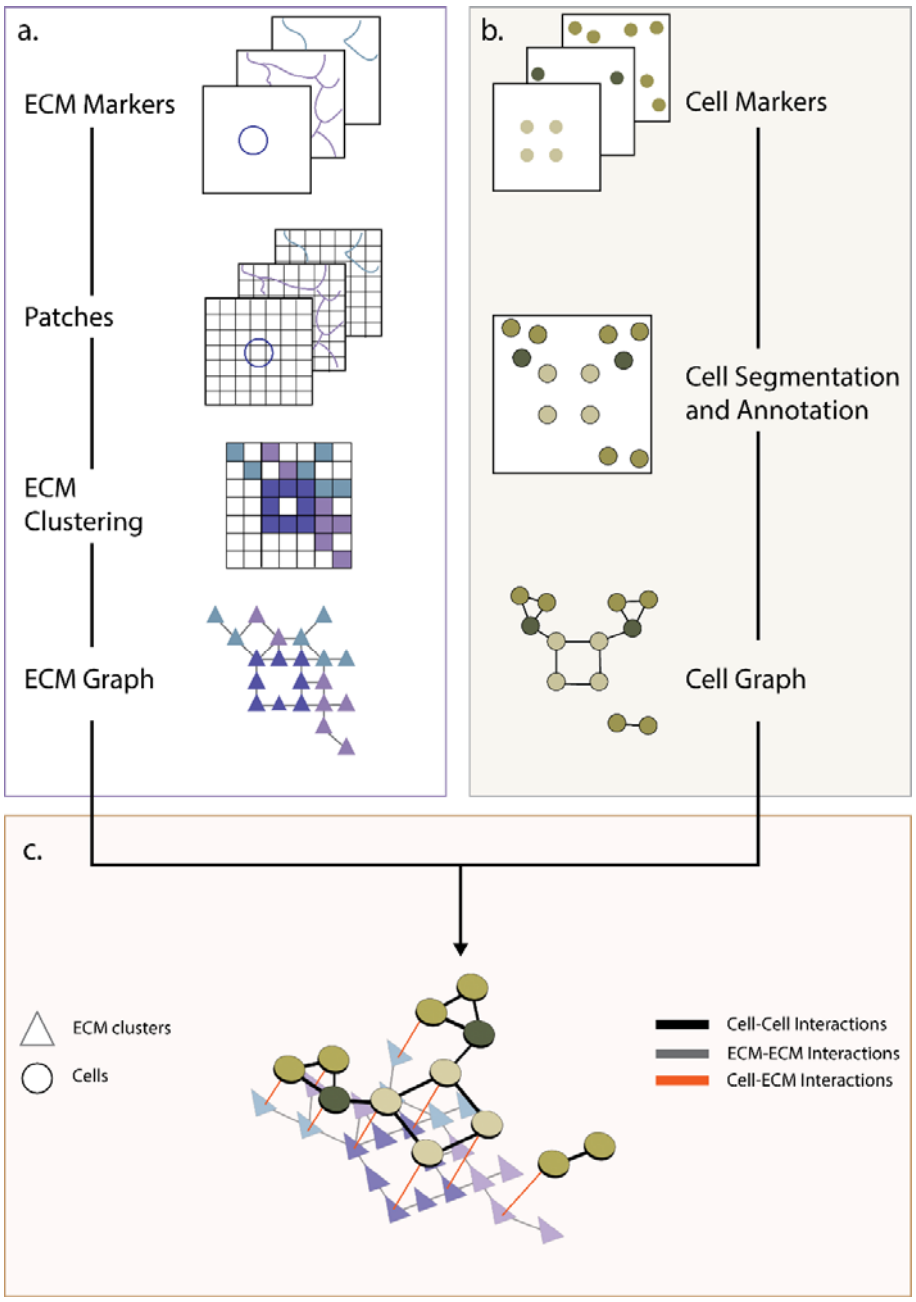

Figure 1: Cell-ECM Graph Overview. (a) ECM channels are split into patches, the mean intensity of each channel is computed, and patches are clustered. Background patches are removed, and the remaining patches become ECM vertices. ECM vertices are connected by edges if they are spatially adjacent creating an ECM graph. **(b)** Cell markers are used for segmentation and annotation, with each segmented cell represented as a vertex. Edges are added between cell vertices if they are spatially interacting creating a cell graph. **(c)** The Cell–ECM Graph is generated by integrating the ECM and cell graphs, adding edges between cell and ECM vertices when they are spatially interacting.

Synthetic Studies

To evaluate the utility of Cell–ECM Graphs under controlled conditions, we generated synthetic spatial proteomic datasets comprising three or four ECM channels and four annotated cell types. Specific cell types were localized to defined ECM clusters to establish ground truth cell–ECM relationships. The simulation retained 40 regions of interest (ROIs) to match the scale of the real dataset and incorporated cell–cell, ECM–ECM, and cell–ECM interactions within the Cell–ECM Graph framework **(Fig. S2a)**. Three dataset configurations were created to test changes in ECM–ECM connectivity **(Fig. S2b)**, cell–ECM interactions **(Fig. S2c)**, and spatial clustering.

Graph deep learning methods with explainability were adapted to focus on the ECM and cell-ECM interactions. GNN explainers have been applied to a range of biological datasets, such as the mutagenicity of molecules [10], drug-disease therapeutic mechanisms [11], EEG data of schizophrenia [12] and histology-based cell graphs of breast cancer [13]. Machine learning has recently been used to identify patterns of remodeling in the ECM [14]; however, we are the first to use GNN

explainers in a cell-ECM graph which can identify both ECM and cell-ECM alterations associated with disease. GNNs were trained for whole-graph classification (e.g., control vs. disease). To identify key interactions driving classification, we applied a graph explainer to highlight the most influential edges (**Fig. 2a**).

In the first configuration, ECM–ECM interactions were modified by disconnecting ECM clusters 1 and 2 in one group, and linking them in the other (**Fig. 2b**). A noise parameter (0–1) was introduced, with 0 representing the original structure with no noise and 1 representing complete random allocation of ECM clusters and complete structural loss (**Fig. 2c**). The GNN achieved an F1 score of 100% at noise = 0, with predictive ability declining to random performance by noise = 0.6 (**Fig. 2d**). The explainer correctly highlighted the altered region and ranked ECM 1–2 edges as most important in the linked group and ECM 3 – 3 and ECM 1 - 3 edges as most important in the disconnected group (**Fig. 2e**).

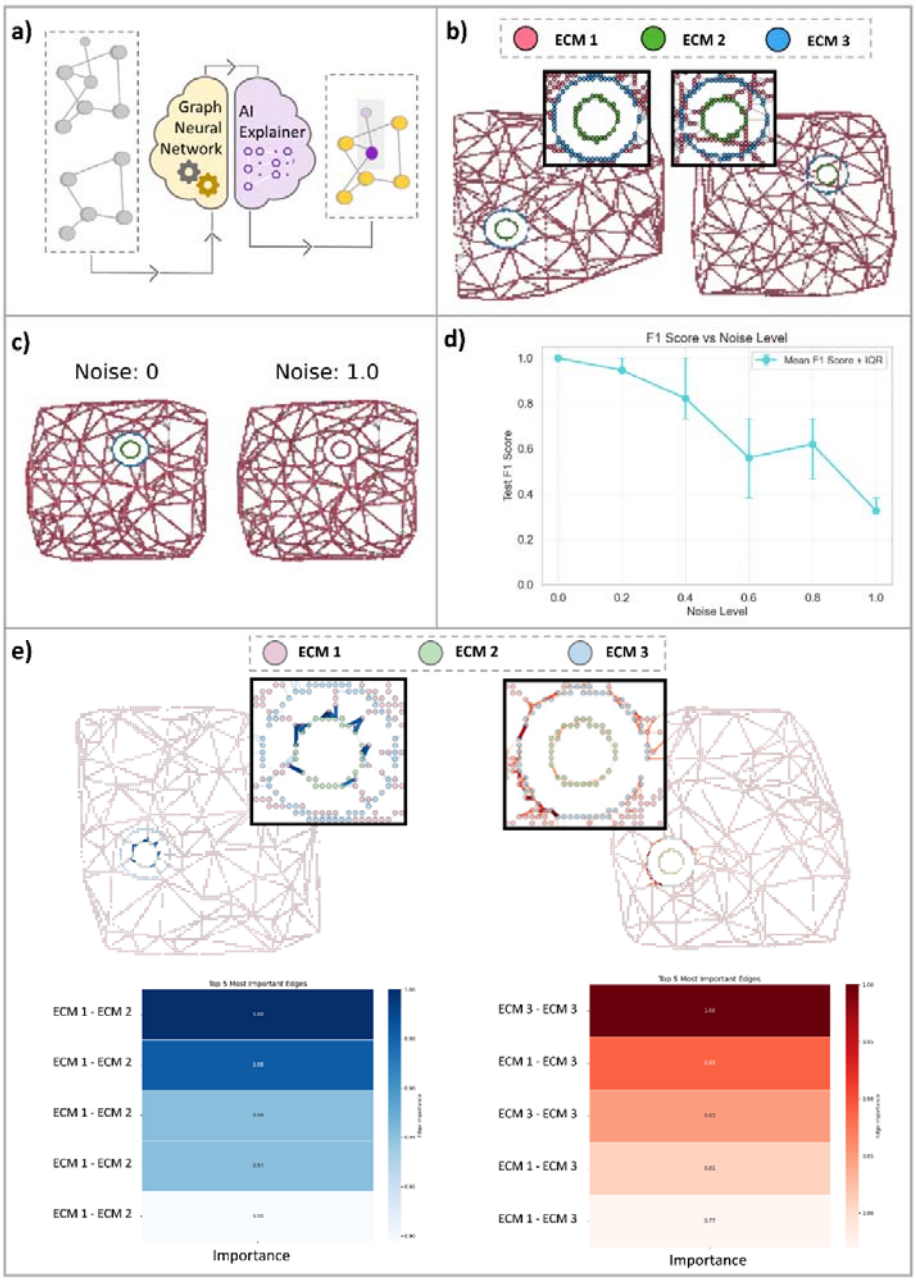

Figure 2: Cell-ECM graphs on synthetic dataset. (a) Overview of whole graph classification using explainable graph neural networks to identify important edges. **(b)** Ground truth ECM changes in synthetic dataset. **(c)** Visualization of the effect of noise parameter 0 (no noise) and 1 (complete noise). **(d)** F1-score of whole graph classification between ECM changes across different noise parameters. **(e)** GNN explainer identifies ground truth ECM changes and top 5 features.

In the second configuration, we altered cell–ECM interactions by introducing a fifth cell type (Cell E) interacting with ECM cluster 1 alongside the dominant Cell A, simulating a rare but specific interaction. The GNN and explainer successfully identified this altered cell–ECM relationship (**Fig. S3**).

The final configuration comprised four spatial regions differing in cell–ECM composition (**Fig. 3a**). Region 1 contained two cell types (Cells C and D) on ECM 4. Region 2 featured a unique pairing of Cell B with ECM 2, whereas Regions 3 and 4 shared the same cell type (Cell A) but were associated with different ECM clusters (ECM 1 and ECM 3, respectively) (**Fig. 3b**).

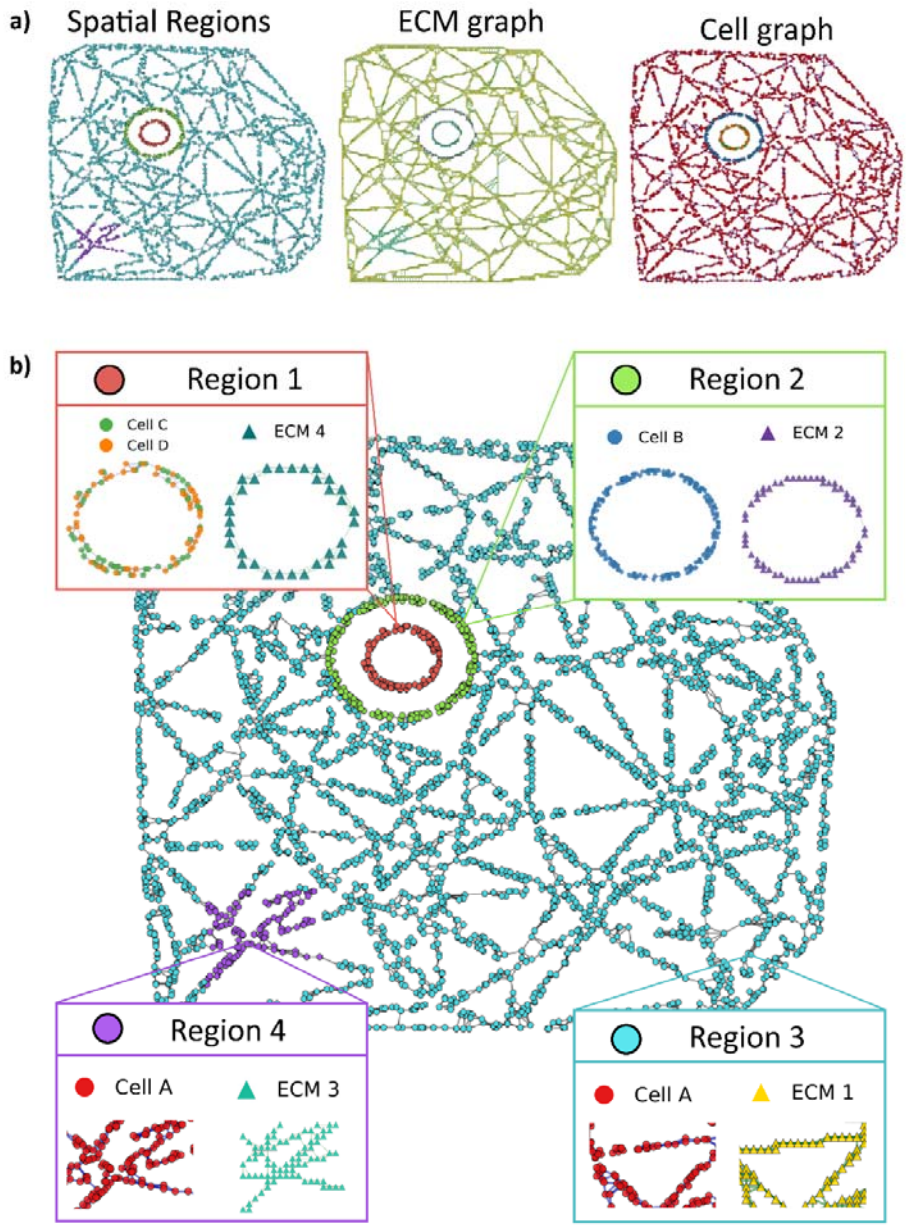

Figure 3: Spatial regions in synthetic dataset. **(a)** Graph representations of spatial regions, cells and the ECM. **(b)** Synthetic dataset with 4 spatial regions that differ in cell and ECM combinations. Region 1 contains two cell types (C and D) on the same ECM (ECM 4), Region 2 has a specific cell type (B) on a unique ECM (ECM 2), while Regions 3 and 4 contain the same cell type (A) on different ECM clusters (ECM 1 and ECM 3, respectively).

We compared the ability of traditional cell graphs and cell–ECM graphs to identify these spatial regions using four clustering approaches: neighbourhood analysis (NH) [5], LisaClust [15], spectral clustering [16], and Leiden clustering [17] (**Fig. 4a**). Performance was evaluated using adjusted Rand index (ARI) and normalized mutual information (NMI) (**Fig. 4b**).

Among all methods evaluated, cell–ECM neighbourhood analysis performed best. Both cell–ECM neighbourhood analysis and LisaClust successfully identified all four spatial regions. Although LisaClust is not graph-based, it effectively captures local spatial associations and improved with incorporating the ECM. In contrast, Leiden and spectral clustering are graph-based but do not incorporate node labels, likely contributing to their lower performance. Even with these methods cell-ECM graphs outperformed the cell only representations. As the ECM is not incorporated within the cell graphs, they were unable to distinguish between region 4 from region 3 (the same cell types on differing ECM).

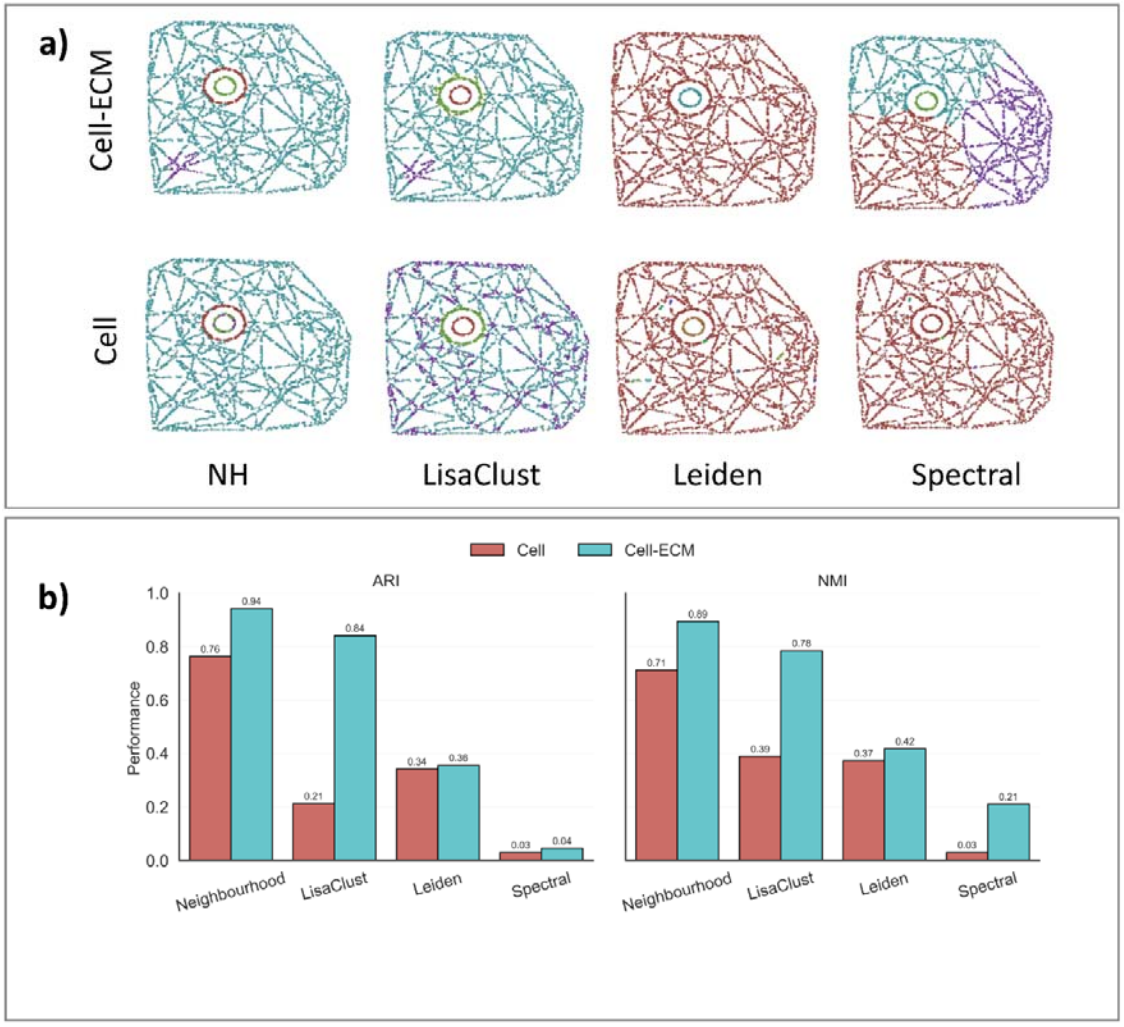

Figure 4: Spatial Clustering Performance Comparison. **(a)** Clustering results from four methods applied to synthetic data with four defined spatial regions, comparing Cell-ECM graphs (top) versus cell-only graphs (bottom). **(b)** Performance quantification using Adjusted Rand Index (ARI) and Normalized Mutual Information (NMI).

Cell-ECM Graphs of Allergic Airway Inflammation

To evaluate our framework on a real spatial proteomic dataset, we applied Cell-ECM graphs to a murine model of allergic airway inflammation analysed by IMC [17]. The dataset comprised 36 regions of interest, including both control and allergen-exposed samples. Cellular markers were used for segmentation and subsequent

annotation of cell types, while 10 ECM markers were incorporated to capture ECM composition. These inputs were integrated into Cell-ECM graphs, where cellular vertices are represented as circles and ECM vertices as triangles, with edges denoting cell–cell, ECM–ECM, and cell–ECM interactions.

To illustrate the components of the Cell–ECM graph, we first visualized the conventional cell graph (**Fig. 5a**). We then constructed the ECM graph (**Fig. 5b**), which revealed three distinct ECM clusters, each aligned with a specific anatomical structure and characterized by unique protein compositions. ECM-1 localized to the airway region, dominated by Collagen VI and Fibronectin. ECM-2 was associated with the alveolar compartment, where Heparan Sulfate, Collagen IV, and Fibrinogen predominated. ECM-3 surrounded vascular structures and was enriched for Collagen I and Collagen III. Cell–ECM interactions linking cells to their local ECM environment are shown in **Fig. 5c**, and the final integrated Cell–ECM graph, combining all interaction types, is presented in **Fig. 5d**, including a magnified view highlighting the multi-layered network structure. The entire dataset ECM patch clusters (**Fig. S4**), cell graphs (**Fig. S5)**, ECM graphs (**Fig. S6**), cell-ECM interactions (**Fig. S7)**, and cell-ECM graphs (**Fig. S8**) are provided within the supplementary. We next explore the utility of Cell–ECM graphs on the real dataset through node classification, spatial clustering, graph neural networks (GNNs) with explainability, and interaction analysis.

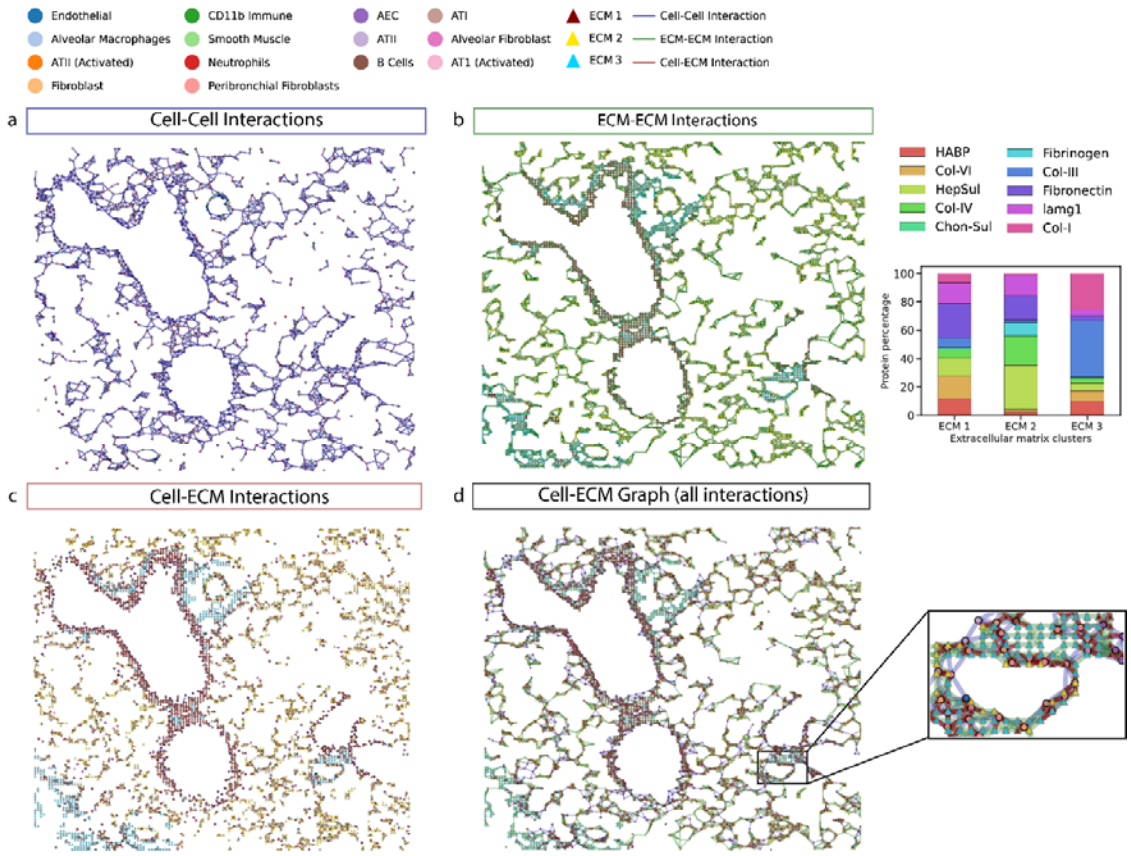

Figure 5: Cell-ECM Graphs of Murine Lung in Imaging Mass Cytometry. a) Cell graph representing cell-cell interactions as edges and cells depicted as vertices. **b)** ECM graph representing ECM-ECM interactions as edges, with ECM clusters as vertices and ECM molecule expression per ECM cluster. **c)** Both cells and ECM are vertices with cell-ECM interactions as edges**. d)** The full Cell-ECM graph with cells and ECM clusters as vertices and cell-cell, ECM-ECM

We performed cellular node classification and compared two NH feature vectors for predicting cell types: (i) using cell–cell interactions, corresponding to a cell-graph, and (ii) cell–cell and cell–ECM interactions, corresponding to a Cell-ECM graph. Hyperparameters were independently optimized for each graph type. Cell-ECM graphs achieved higher macro F1-score compared to cell-graphs (52.5% vs. 46.5%), indicating that the inclusion of the ECM provides additional informative structure, thereby improving the representational capacity of the graph (**Table 1**).

Table 1. Cell vs Cell-ECM Node Classification HP after optimization in real dataset.

Optimized	$D_{max}CC$	$D_{max}CE$	$N_{edge}(G)$	$K_{classifier}$	F1-Score
Cell graph	50	-	3	29	0.465
Cell-ECM graph	17	7	24	15	**0.525**

- HP = Hyperparameters

To explore the use of spatial clustering in the real dataset, we clustered immune cells in an interaction space defined by their cell and ECM interactions into 4 clusters which we refer to as Interaction-Defined Immune Niches (IDINs) (**Fig. 6a**).

The relative distribution of IDINs across conditions (**Fig. 6b**), mean interactions (**Fig. 6c**), and cell-type composition (**Fig. 6d**) allowed biological interpretation of each IDIN. IDIN-1 corresponded to B-cell aggregates present in allergic samples, consistent with previous reports that B cells are required for the formation of peribronchiolar lymphoid aggregates during lung inflammation [18]. IDIN-2 represented an alveolar macrophage–epithelial interface with ECM-2/AT1/ATII contacts; alveolar macrophages and epithelial cells are known to exchange immunosuppressive signals [19], and this niche was the largest within controls. IDIN-3 reflected an airway niche represented by a mix of immune cells with monocytes a large constituent interacting with AEC, peribronchial fibroblasts and ECM 1 within allergic samples. Finally, IDIN-4 was defined as a lymphoid–myeloid stromal niche containing a mixed population of B cells and $CD11b^{+}$ immune cells interacting with ECM-3, B-cells and fibroblasts. Representative examples of IDINs in control and allergen-exposed tissue are visualized within cell-ECM graphs in **Fig. 6e(i)**.

To illustrate the use of GNNs with explainability, we focused on cell-ECM interactions. We trained a GNN for graph classification with nested cross-validation, achieving F1-Score of 94.7% ± 1.7% for distinguishing allergic vs control samples. We then applied a GNN explainer to held-out individual ROIs to obtain instance-level edge attributions. In the control ROI shown (**Fig. 6e(ii)**), Fibroblast–ECM-3 appeared among the top-ranked explanatory edges in controls; in the allergen-exposed ROI, Monocyte–ECM-1 and CD11b-immune–ECM-3 were top-ranked.

Interaction tests are commonly used to identify significant interactions in cell graphs. We applied the interaction test on cell-ECM graphs which included the ECM within the analysis (**Fig. 6e(iii)**). No significant changes in ECM–ECM connectivity were observed. However, altered cell–ECM interactions were detected. Allergen interactions increased interactions between ATI and ATII with ECM cluster 2. Immune–ECM interactions were also affected: B cells showed reduced interactions with ECM cluster 3, while CD11b^{+} immune cells exhibited a significant increase in interactions with ECM cluster 3.

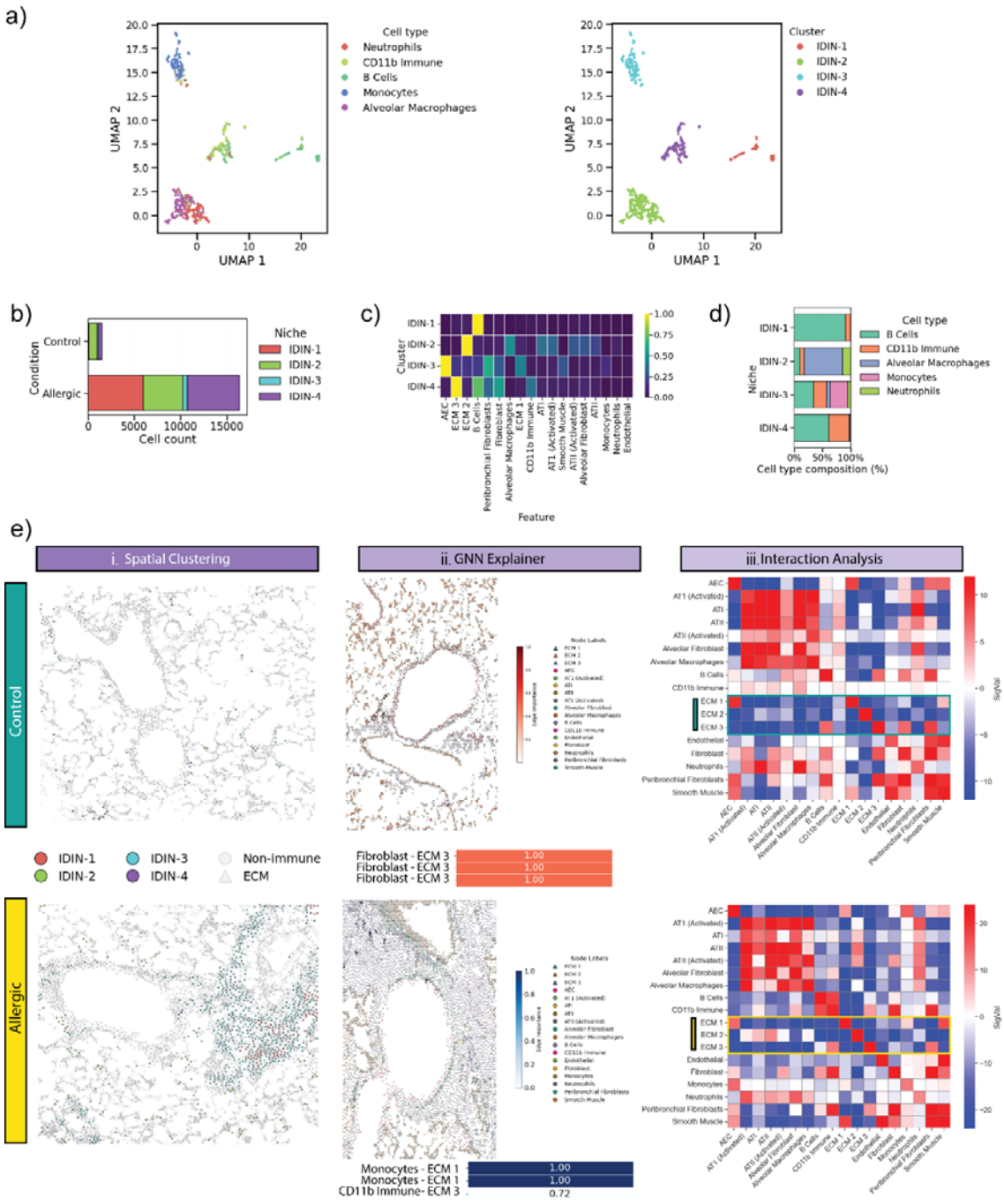

Figure 6. Cell-ECM graphs on real IMC dataset. **(a)** Spatial clustering of immune-cells based on cell and ECM interactions into Interaction Defined Immune Niches (IDIN). **(b)** Proportion of IDIN clusters in control vs allergic. **(c)** Mean cell and ECM interactions per IDIN cluster. **(d)** Proportions of cell types per IDIN cluster. **(e)** Application of i) immune spatial clustering, ii) GNN explainer and ii) interaction analysis on control and allergic ROIs.

Conclusion

We introduce a general method to analyse ECM architecture and cell–ECM interactions by representing cells and ECM within a unified graph. Leveraging graph analytics—spatial clustering, interaction analysis, and GNNs with explainers—we show that ECM- and cell–ECM–centric alterations associated with disease can be detected and spatially localized. To enable broad adoption and further development, we provide a user-friendly Python package with tutorials that implements the full workflow from graph construction to visualization and downstream graph analytics and machine learning.

Methods

Cell-ECM Graphs Tutorial

A step-by-step tutorial with an included synthetic dataset is available at: https://github.com/moeghaf/Cell-ECM-Graphs. We show how to load cell-ECM graphs with some downstream analysis. The tutorials can be fully run on a web browser via Google Colabs.

Cell-ECM graph generation

The equation for a graph is $G = (V, E)$, where V represents a set of vertices and E a set of edges between vertices. The cell graph was constructed by treating each cell as a vertex and the cell type label assigned to each vertex using each cell's centroids. To define the edges, representing cell-cell interactions, we find the

$N_{edge}(G)$ nearest neighbouring cells based on spatial proximity. Subsequently, we pruned any edges with a distance greater than $D_{max}(CC)$ to refine the cell graph. This "cell-cell distance" hyperparameter controls the length-scale of a cell interaction.

To represent the ECM as a graph, only the ECM markers were selected. The image was divided into P_{size} x P_{size} patches with N channels, where N is the number of ECM markers. K-means clustering, with K_{ECM} clusters (determined using silhouette coefficient), was then applied to these patches where patches were represented by a N-dimensional vector representing the mean of each marker within each patch. Patches assigned to a background cluster (little or no ECM expression and relabelled as ECM cluster 0) were removed and the remaining patches became vertices with a label determined by the cluster assignment. To create edges representing ECM-ECM interactions, an edge was added between $N_{edge}(G)$ nearest neighbouring ECM nodes.

The merging of the cell graph and ECM-graph through cell-ECM edges produced the final Cell-ECM graph. Cell-ECM interactions were added based on finding the $N_{edge}(G)$ nearest ECM vertices to each cell and adding edges for cells within distance $D_{max}(CE)$ of the ECM node (this "cell-ECM distance" is the final hyperparameter which controls the length-scale of the cell-ECM interactions). This integration created a comprehensive graph that captured cell-cell, ECM-ECM and cell-ECM interactions.

Datasets

Synthetic Spatial Proteomic Dataset with ECM Markers

To systematically evaluate cell-ECM Graphs, we generated a synthetic IMC dataset with three ECM markers and controlled spatial organization of ECM and cell types.

The ECM was simulated on a 500×500 grid by randomly generating 100 points and connecting them via Delaunay triangulation (ECM marker 1). Additionally, a hollow circular structure (ECM marker 2) containing a smaller circle (ECM marker 3) were added at random locations. Each ECM channel was assigned a distinct intensity range (500, 1000, and 1500, respectively) to simulate marker variability.

Cell types were then randomly overlaid onto the ECM: 4000 cells of type A on ECM marker 1, 200 cells of type B on ECM marker 2, and 50 cells each of types C and D on ECM marker 3. To test the ability of Cell-ECM Graphs to capture changes in ECM-ECM and cell-ECM interaction, two datasets were generated. The first altered ECM-ECM interactions by either removing or retaining connections between ECM markers 1 and 2. The second altered cell-ECM interactions by adding 50 type E cells to ECM marker 1. To mimic the scale of murine datasets, each dataset contained 40 regions of interest, with 20 per condition. A noise parameter was introduced by randomly shuffling node labels, where 0 corresponds to no shuffling (0% of vertices) and 1 corresponds to complete shuffling (100% of vertices).

Murine Imaging Mass Cytometry Dataset

The real dataset [20] is derived from the murine lung and the markers are outlined in **Table S1**. A total of 36 760 x 790µm regions of interest were captured from 5 mice administered Phosphate-Buffered Saline (PBS), as controls, and 6 mice administered a multiallergen cocktail (Dust mite, Ragweed and Aspergillus fumigatus; DRA), both via the intranasal route [21]. Cell segmentation was performed using the Steinbock pipeline [22]. The DeepCell option with min-max normalization was selected. This approach uses a pre-trained model that predicts cell segmentations from cytoplasmic and nuclear markers from the raw image data. Cells were manually annotated using biological knowledge from known cell markers.

Spatial Clustering Methods

Cell neighborhood feature vectors were generated for each cell by counting the number of neighboring interactions with each cell type. Similarly, cell-ECM neighborhood feature vectors were generated by counting the number of interactions with both cell and ECM types. These feature vectors were clustered using K-means with K = 4.

Local Indicators of Spatial Association Clustering (LisaClust) does not use a graph representation. It operates on spatial coordinates and cell type labels by performing local enrichment analysis using Ripley's K-function and clustering based on spatial autocorrelation. A key hyperparameter is the interaction radius, which was optimized (**Table S2**). Although originally designed for cell data, we adapted the method by treating ECM vertices analogously to cells.

Leiden clustering is a graph-based community detection algorithm that partitions the graph to optimize a quality function such as modularity. It does not use node features (i.e., cell and ECM labels). A key hyperparameter is the resolution, which controls cluster granularity—higher values yield more clusters and lower values yield fewer (**Table S3** for optimization).

Spectral clustering operates by constructing a similarity matrix from the graph, computing its Laplacian, and performing dimensionality reduction via its eigenvectors. K-means is then applied to the resulting low-dimensional representation. Like Leiden, it does not incorporate node features. The primary hyperparameter is the number of clusters K, set to 4 in this benchmark.

Graph Neural Network and Explainability

We implemented a GNN for whole graph classification on Cell–ECM graphs with node features set as one-hot encoding of the node label (cell type or ECM cluster). The model was trained using nested cross validation, with 5 inner and 5 outer folds [23]. The model hyperparameters were optimized using Bayesian optimization. The model consisted of graph convolution layers followed by fully connected layers. Message passing was followed by global pooling to generate graph-level embeddings. The final output was passed through a log-softmax activation to predict graph class probabilities. The optimized hyperparameters per fold are available in **Table S4.** Training and validation losses for the whole graph classification on the synthetic ECM graph (**Fig. S9**), synthetic cell-ECM graph (**Fig. S10**) and real cell-ECM graphs (**Fig. S11**). Adam optimizer and negative log-likelihood loss function were used. Early stopping was employed based on validation loss to prevent overfitting.

To interpret model predictions, we applied Integrated Gradients [24], which attributes prediction outcomes to edge importance by integrating gradients from a baseline graph (edge weights set to 0) to the actual input graph (edge weights set to 1). For each edge, the attribution is approximated as:

$$Attribution_{ei} = \int_{\alpha=0}^{1} \frac{\partial F(x_a)}{\partial w_{ei}} d\alpha$$

where is $x_a$the graph with edge weights scaled by α and (x_a) is the model's output. The integral was estimated using a discrete Riemann sum implemented with the Captum library [25]. The GNN was built using PyTorch Geometric [26].

Node Classification and Optimization

Once the graphs were built, we performed cellular node classification, using either the neighbouring cells (predicting cell types from direct cell-cell interactions) or neighbouring cell and ECM (predicting cell types from interacting cells and ECM). This was performed using all the ROIs of the real murine dataset. A feature vector was created by counting the number of cellular, or both ECM and cellular interactions a specific node, with a K-nearest neighbour (KNN) classifier then used to assign a label to each node. For Bayesian optimization, the data was split into 80% training, 10% validation and 10% testing and optimized for 100 iterations using Bayesian optimization by maximizing the F1-score for cell graphs and cell-ECM graphs independently. The hyper parameter search space $D_{max}(CC)$ and $D_{max}(CE)$ was from 0 to 50, while the search space for $N_{edge}(G)$ and $K_{classifier}$ ranged from 3 to 30.

Interaction test

The interaction test followed that of imcRtools testInteractions function [27]. The ‘classic’ method was used to count and summarize interactions, whereby each count was divided by the total number of cell types or ECM clusters. The labels of these counts were randomly shuffled 1000 times to create a distribution of the interaction count under spatial randomness. The observed interaction count was compared against this Null distribution to derive empirical p-values (p=0.05). A significant value of +1 indicated significant interactions, -1 avoidance and 0 neither. These values were also summed to compare the interactions of multiple ROIs under a specific condition.

Code

The graphs were generated using Python, and the code for the Cell-ECM Graphs tool is publicly available on GitHub: https://github.com/moeghaf/Cell-ECM-Graphs.

Data

The synthetic datasets have been made publicly through the GitHub tutorial. The murine IMC dataset will be provided upon reasonable request (it will be published when the preprint is accepted).

References

[1] C. Frantz, K.M. Stewart, V.M. Weaver, The extracellular matrix at a glance, J. Cell Sci. 123 (2010) 4195–4200. https://doi.org/10.1242/jcs.023820 .

[2] S.R. Lamandé, J.F. Bateman, Genetic disorders of the extracellular matrix, Anat. Rec. (Hoboken) 303 (2020) 1527–1542. https://doi.org/10.1002/ar.24086 .

[3] W. Huber, *et al.*, Graphs in molecular biology, BMC Bioinformatics 8 Suppl 6 (2007) S8. https://doi.org/10.1186/1471-2105-8-S6-S8 .

[4] H.W. Jackson, *et al.*, The single-cell pathology landscape of breast cancer, Nature 578 (2020) 615–620. https://doi.org/10.1038/s41586-019-1876-x .

[5] Y. Goltsev, *et al.*, Deep profiling of mouse splenic architecture with CODEX multiplexed imaging, Cell 174 (2018) 968–981. https://doi.org/10.1016/j.cell.2018.07.010 .

[6] D. Schapiro, *et al.*, histoCAT: analysis of cell phenotypes and interactions in multiplex image cytometry data, Nat. Methods 14 (2017) 873–876. https://doi.org/10.1038/nmeth.4391 .

[7] C. Gunduz, B. Yener, S.H. Gultekin, The cell graphs of cancer, Bioinformatics 20 Suppl 1 (2004) 145–151. https://doi.org/10.1093/bioinformatics/bth933 .

[8] C. Demir, S.H. Gultekin, B. Yener, Augmented cell-graphs for automated cancer diagnosis, Bioinformatics 21 Suppl 2 (2005) ii7–ii12. https://doi.org/10.1093/bioinformatics/bti1100 .

[9] C.C. Bilgin, P. Bullough, G.E. Plopper, B. Yener, ECM-aware cell-graph mining for bone tissue modeling and classification, Data Min. Knowl. Discov. 20 (2010) 416–438. https://doi.org/10.1007/s10618-009-0153-2 .

[10] Z. Ying, D. Bourgeois, J. You, M. Zitnik, J. Leskovec, GNNExplainer: Generating Explanations for Graph Neural Networks, Adv. Neural Inf. Process. Syst. 32 (2019).

[11] Y. Zhang, *et al*., DDTExplainer: Mining Drug-Disease Therapeutic Mechanisms based on GNN Explainability, 2024 IEEE Int. Conf. Bioinformatics Biomedicine (BIBM) (2024) 1350–1355. https://doi.org/10.1109/BIBM62325.2024.10822060 .

[12] M. Zhdanov, S. Steinmann, N. Hoffmann, Investigating Brain Connectivity with Graph Neural Networks and GNNExplainer, 2022 26th Int. Conf. Pattern Recognit. (ICPR) (2022) 5155–5161. https://doi.org/10.1109/ICPR56361.2022.9956201 .

[13] G. Jaume, *et al*., Quantifying Explainers of Graph Neural Networks in Computational Pathology, 2021 IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR) (2021) 8102–8112. https://doi.org/10.1109/CVPR46437.2021.00801 .

[14] Emerson MJ, Willacy O, Madsen CD, Reuten R, Brøchner CB, Lund TK, Dahl AB, Jensen THL, Erler JT, Mayorca-Guiliani AE. Machine learning identifies remodeling patterns in human lung extracellular matrix. Acta Biomater. 195 (2025) 94–103. https://doi.org/10.1016/j.actbio.2024.12.062 .

[15] E. Patrick, *et al*., Spatial analysis for highly multiplexed imaging data to identify tissue microenvironments, Cytometry A 103 (2023) 593–599. https://doi.org/10.1002/cyto.a.24729 .

[16] A. Ng, M. Jordan, Y. Weiss, On Spectral Clustering: Analysis and an Algorithm, Adv. Neural Inf. Process. Syst. 14 (2001).

[17] V.A. Traag, L. Waltman, N.J. van Eck, From Louvain to Leiden: guaranteeing well-connected communities, Sci. Rep. 9 (2019) 5233. https://doi.org/10.1038/s41598-019-41695-z .

[18] J.A. Poole, *et al*., A role for B cells in organic dust induced lung inflammation, Respir. Res. 18 (2017) 214. https://doi.org/10.1186/s12931-017-0703-x .

[19] K. Westphalen, *et al*., Sessile alveolar macrophages communicate with alveolar epithelium to modulate immunity, Nature 506 (2014) 503–506. https://doi.org/10.1038/nature12902 .

[20] J.E. Parkinson, *et al*., Extracellular matrix phenotyping by imaging mass cytometry defines distinct cellular matrix environments associated with allergic airway inflammation, bioRxiv (2024). https://doi.org/10.1101/2024.11.15.623782 .

[21] N. Goplen, *et al*., Combined sensitization of mice to extracts of dust mite, ragweed, and Aspergillus species breaks through tolerance and establishes chronic features of asthma, J. Allergy Clin. Immunol. 123 (2009) 925–932. https://doi.org/10.1016/j.jaci.2009.02.009 .

[22] J. Windhager, V.R.T. Zanotelli, D. Schulz, *et al.*, An end-to-end workflow for multiplexed image processing and analysis, Nat. Protoc. (2023). https://doi.org/10.1038/s41596-023-00881-0 .

[23] M.J. Lewis, *et al.*, nestedcv: an R package for fast implementation of nested cross-validation with embedded feature selection designed for transcriptomics and high-dimensional data, Bioinformatics Adv. 3 (2023) vbad048. https://doi.org/10.1093/bioadv/vbad048 .

[24] M. Sundararajan, A. Taly, Q. Yan, Axiomatic Attribution for Deep Networks, arXiv Preprint (2017). https://doi.org/10.48550/arXiv.1703.01365 .

[25] N. Kokhlikyan, *et al.*, Captum: A unified and generic model interpretability library for PyTorch, arXiv Preprint (2020). https://doi.org/10.48550/arXiv.2009.07896 .

[25] M. Fey, J.E. Lenssen, Fast Graph Representation Learning with PyTorch Geometric, arXiv Preprint (2019). https://doi.org/10.48550/arXiv.1903.02428 .