Universidade do Minho
Escola de Engenharia

Ana Marta Fernandes Tavares Sequeira

A deep learning framework towards the multidimensional topography of carcinogenesis to guide prevention and therapy

April, 2025

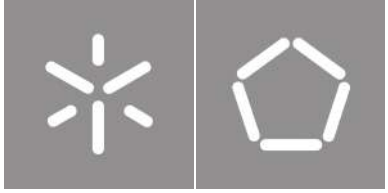
A deep learning framework towards the multidimensional topography of carcinogenesis to guide prevention and therapy

Ana Marta Sequeira

UMinho | 2025

REPÚBLICA PORTUGUESA | CIÊNCIA, TECNOLOGIA E ENSINO SUPERIOR

fct Fundação para a Ciência e a Tecnologia



Universidade do Minho
Escola de Engenharia

Ana Marta Fernandes Tavares Sequeira

A deep learning framework towards the multidimensional topography of carcinogenesis to guide prevention and therapy

Doctorate Thesis
Doctorate in Informatics

Work developed under the supervision of:
Miguel Francisco de Almeida Pereira da Rocha
Noel Filipe da Cunha Carvalho de Miranda
Dina Ruano Neto

April, 2025

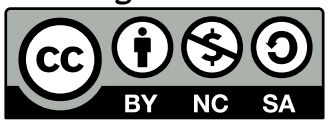
COPYRIGHT AND TERMS OF USE OF THIS WORK BY A THIRD PARTY

This is academic work that can be used by third parties as long as internationally accepted rules and good practices regarding copyright and related rights are respected.

Accordingly, this work may be used under the license provided below.

If the user needs permission to make use of the work under conditions not provided for in the indicated licensing, they should contact the author through the RepositoriUM of Universidade do Minho.

License granted to the users of this work



Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International
CC BY-NC-SA 4.0

`https://creativecommons.org/licenses/by-nc-sa/4.0/deed.en`

This document was created with the (pdf/Xe/Lua)LaTeX processor and the NOVAthesis template (v7.1.24).

Acknowledgements

This thesis marks the culmination of a truly transformative journey — shaped by perseverance, curiosity, and a deep commitment over the past four years. Reaching this point would not have been possible without the support of many people and institutions who, directly or indirectly, contributed to this achievement.

First and foremost, I want to express my gratitude to Fundação para a Ciência e Tecnologia for awarding me the PhD scholarship (ref. 2020.07867.BD), which provided essential financial support throughout this time. I'm also deeply thankful to the Centre of Biological Engineering at the University of Minho and the Leiden University Medical Center (LUMC) for hosting and supporting my research.

To my supervisors — Dr. Miguel Rocha, Dr. Noel de Miranda, and Dr. Dina Ruano — thank you for your guidance, encouragement, and mentorship.

I'm also grateful to all the members of the BISBII research group at the University of Minho. I'm especially grateful to the Tumour Immunogenomics group and the LUMC community for their warm welcome and the sense of belonging they provided during my two years abroad. A special thank you to Dr. Marieke Ijsselsteijn and Dr. Jessica Roelands, whose direct scientific contributions, and insightful discussions greatly shaped this thesis. I'm also deeply thankful for their genuine friendship throughout this journey.

To my friends from Leiria, Aveiro, Braga, and Leiden — thank you for your continuous support throughout every stage, despite the distance and my tendency to go silent. A heartfelt thank you to Tine Turso, my "second mom" in the Netherlands for her kindness, and *gezelligheid* that brought a sense of home when I was far from mine.

To everyone who crossed my path and added joy, learning, or laughter to this chapter — thank you.

Finally, to my family — for your constant love and support, no matter the distance.

STATEMENT OF INTEGRITY

I hereby declare having conducted this academic work with integrity. I confirm that I have not used plagiarism or any form of undue use of information or falsification of results along the process leading to its elaboration.

I further declare that I have fully acknowledged the Code of Ethical Conduct of the Universidade do Minho.

_____, _____

(Place) (Date)

_____

(Ana Marta Fernandes Tavares Sequeira)

Propositions

In the Netherlands, PhD theses are traditionally accompanied by a list of "propositions" (*Stellingens*) — statements inspired by the research and reflections of the author. As my journey was deeply marked by my time in the Netherlands, I share here my own propositions:

1. Completing a PhD is like a cycling tour — while it may appear to be solely an individual effort, crossing the finish line is only possible with the support of an entire team.
2. Artificial Intelligence is a powerful tool, but not every problem requires it. Sometimes, simpler solutions are more effective.
3. The availability of biological datasets has a major impact on scientific progress, enabling the research community to explore new questions and validate findings collaboratively.
4. Interdisciplinary collaboration in medical research — particularly between clinicians and scientists — is essential for driving meaningful, patient-centered discoveries.
5. Scientific achievement depends not only on knowledge but on collaboration, communication, and a genuine curiosity about others' work.
6. Science builds upon science. — each breakthrough rests on a foundation laid by others.
7. In research, we constantly navigate a trade-off between risk, novelty, usefulness, feasibility, and time. Knowing when to take bold steps and when to focus on what's doable is key to making meaningful and timely contributions.
8. As biology enters the era of big data, integrating computational expertise within experimental research groups is essential. Bioinformaticians and biologists must work closely together, as neither can thrive in isolation.
9. Artificial Intelligence is transforming science, forcing us to rethink how we conduct, evaluate, and share research. The role of the scientist is changing — and so must the structure of science itself.
10. A PhD is a significant milestone, and universities should celebrate and recognize its achievement more visibly.
11. Creativity is essential in science — and especially in engineering — where novel solutions often emerge from unconventional thinking.

Resumo

Uma ferramenta de aprendizagem profunda para a topografia multidimensional da carcinogénese para guiar prevenção e terapia

A ómica espacial é um campo em rápida evolução que visa mapear características moleculares, como proteínas e transcritos de RNA, directamente nos tecidos, preservando o seu contexto espacial. Esta abordagem oferece percepções inéditas sobre a biologia e patologia dos tecidos, revelando heterogeneidade, organização espacial e interações intercelulares, especialmente importantes no contexto da investigação do cancro. No entanto, a análise destes dados apresenta desafios computacionais significativos, incluindo a alta dimensionalidade, ruído, esparsidade e a necessidade de segmentação celular. A integração de técnicas modernas de aprendizagem de máquina tem grande potencial para enfrentar estes desafios e possibilitar o uso destes dados na medicina personalizada, particularmente no cancro.

Esta tese apresenta novas ferramentas e métodos computacionais para a análise de dados ómicos espaciais, com um foco particular no cancro colorrectal como caso de estudo. O trabalho abrange várias áreas-chave: o desenvolvimento de PENGUIN, uma ferramenta de pré-processamento para dados de proteómica espacial; uma comparação e avaliação das estratégias de segmentação celular para proteómica espacial; e a introdução de GRIDGEN, um método não baseado em segmentação celular direcionado para dados de transcriptómica espacial com resolução subcelular. Estas contribuições visam simplificar e melhorar a análise destes dados, que são complexos e de alta dimensionalidade, tornando-os assim mais acessíveis aos investigadores e facilitando a identificação de características biológicas relevantes.

Os métodos e ferramentas desenvolvidos nesta tese são de código aberto, fornecendo uma base para futuras pesquisas com base em dados ómicos espaciais. Este trabalho também destaca a importância de combinar técnicas computacionais clássicas com abordagens modernas de aprendizagem máquina para permitir análises escaláveis, reprodutíveis e capazes de extrair conhecimento de dados de ómicas espaciais. Ao abordar desafios-chave no pré-processamento de dados, segmentação e extração de características, esta tese contribui para o crescente corpo de trabalho em dados ómicos com dimensão espacial e oferece novas perspectivas para investigações futuras, particularmente na biologia do cancro e medicina personalizada.

Palavras-chave: Análise de dados espaciais, Cancro Colorectal, Microambiente tumoral, Ómicas espaciais, Segmentação celular

Abstract

A deep learning framework towards the multidimensional topography of carcinogenesis to guide prevention and therapy

Spatial omics is a rapidly evolving field that aims to map molecular features, such as proteins and RNA transcripts, directly within tissues, preserving their spatial context. This approach provides unprecedented insights into tissue biology and pathology, revealing cellular heterogeneity, spatial organization, and intercellular interactions, particularly in cancer research. However, the analysis of spatial omics data presents significant computational challenges, including high dimensionality, noise, sparsity, and the need for accurate cell segmentation. The integration of modern machine learning techniques holds great promise for addressing these challenges and enabling the use of spatial omics in personalized medicine and cancer treatment.

This thesis presents novel computational frameworks and methods for the analysis of spatial omics data, with a particular focus on colorectal cancer (CRC) as a case study. The work spans several key areas: the development of PENGUIN, a robust pre-processing pipeline for imaging mass cytometry data; a comprehensive benchmarking of cell segmentation strategies for spatial proteomics; and the introduction of GRIDGEN, a cell-free method for analyzing subcellular spatial transcriptomics data. These contributions aim to streamline and enhance the analysis of complex, high-dimensional spatial omics datasets, making them more accessible to researchers and facilitating the identification of meaningful biological insights.

The methods and tools developed in this thesis are open-source, providing a foundation for future research in spatial omics. Through extensive validation, including CRC case studies, this work demonstrates the potential of these tools for uncovering spatial biomarkers, mapping cellular interactions, and advancing cancer research. The research also underscores the importance of combining classical computational techniques with modern deep learning approaches to enable scalable, reproducible, and insightful analyses of spatial omics data.

By addressing key challenges in data pre-processing, segmentation, and feature extraction, this thesis contributes to the growing body of work in spatial omics and provides new avenues for future research, particularly in cancer biology and personalized medicine.

Keywords: Cell segmentation, Colorectal Cancer, Spatial data processing, Spatial Omics, Tumour microenvironment

Contents

List of Figures **xi**

List of Tables **xiii**

Acronyms **xiv**

1 Introduction **1**

1.1 Context and motivation 1
1.2 Research Objective 2
1.3 Thesis outline 3

2 Background **4**

2.1 Machine and Deep Learning 4
2.1.1 Unsupervised learning 4
2.1.2 Supervised learning 5
2.1.3 Traditional Supervised Machine Learning Algorithms 8
2.1.4 Deep learning 12
2.1.5 Architectures of deep neural networks 15
2.1.6 Self-supervised learning 21
2.2 Cancer: definitions, microenvironment and heterogeneity 22
2.2.1 Tumour microenvironment 23
2.2.2 Approaches to Studying Cancer 24
2.3 Spatially Resolved Omics Techniques 25
2.3.1 Spatial proteomics 25
2.3.2 Spatial transcriptomics 27
2.3.3 Spatial metabolomics 28
2.3.4 Spatial multi-omics 28
2.4 Analytical Frameworks for Spatial Omics Data 30
2.4.1 Data preprocessing 30

2.4.2 Cell segmentation . . . 31
2.4.3 Cell phenotyping . . . 32
2.4.4 Cellular neighbourhood analysis . . . 34
2.4.5 Spatial domains . . . 35
2.4.6 Spatially Variable genes . . . 36
2.4.7 Functional analysis . . . 36
2.4.8 How to distinguish between sample conditions . . . 37
2.4.9 Frameworks and tools for spatial omics analysis . . . 39
2.4.10 Closing remarks . . . 39

3 A rapid and efficient image preprocessing tool for multiplexed spatial proteomics 42
3.1 Introduction . . . 42
3.2 Results . . . 44
3.2.1 Comparison of available methods for IMC denoising . . . 45
3.2.2 PENGUIN enhances downstream analysis . . . 49
3.2.3 PENGUIN can be applied to other image modalities . . . 50
3.3 Discussion . . . 50
3.4 Methods . . . 52
3.4.1 Dataset . . . 52
3.4.2 Denoising Methods comparison . . . 52
3.4.3 PENGUIN implementation . . . 53
3.4.4 Comparison of methods for downstream analysis . . . 53
3.5 Data and code availability . . . 55
3.6 Supplementary information . . . 55

4 Cell segmentation in multiplexed proteomics: a comparative analysis 63
4.1 Introduction . . . 64
4.2 Results . . . 65
4.2.1 Derivation and evaluation of segmentation algorithms . . . 65
4.2.2 Phenotyping, morphological and intensity assessment . . . 69
4.2.3 Downstream analysis comparison . . . 70
4.3 Discussion and conclusion . . . 73
4.4 Methods . . . 76
4.4.1 Dataset . . . 76
4.4.2 Preprocessing . . . 76
4.4.3 Annotation dataset . . . 76
4.4.4 Fine-tuning of Cellpose models . . . 76
4.4.5 Comparison of segmentation models on test set . . . 77
4.4.6 Segmentation metrics . . . 77

4.4.7 Derivation of OneGo and Sequential final segmentations . . . 78
4.4.8 Preprocessing of cell data . . . 79
4.4.9 Phenotyping . . . 79
4.4.10 Downstream analysis . . . 79
4.5 Data and code availability . . . 79
4.6 Supplementary information . . . 80

5 GRIDGEN: Guided Region Identification based on Density of GENes 97
5.1 Introduction . . . 97
5.2 Results . . . 99
5.2.1 Description of GRIDGEN . . . 99
5.2.2 Tumour Stroma analysis with GRIDGEN . . . 102
5.2.3 Single and Multiclass Object analysis . . . 103
5.2.4 Mapping with cell segmentation . . . 105
5.2.5 Alternative ways of masking . . . 105
5.3 Discussion and Conclusions . . . 108
5.4 Methods . . . 110
5.4.1 GRIDGEN implementation . . . 110
5.4.2 Data . . . 112
5.4.3 Tumour stroma definition in CosMx and Xenium . . . 112
5.4.4 Tumour stroma Interface analysis . . . 112
5.4.5 Population specific objects definition . . . 113
5.4.6 Cell segmentation . . . 113
5.4.7 Cell phenotyping . . . 113
5.4.8 Cell segmentation and mask overlay . . . 114
5.4.9 SOM-based masks . . . 114
5.4.10 Neighbour counting based masks . . . 114
5.5 Data and code availability . . . 114
5.6 Supplementary information . . . 115

6 Conclusion and Future Work 118
6.1 Main contributions . . . 118
6.2 Publications . . . 120
6.3 Perspectives for future work . . . 121

Bibliography 124

List of Figures

2.1 Schematic of a simple Artificial Neural Network (ANN) and a single neuron. 11
2.2 Fully connected feedforward DNN schematic. 16
2.3 Convolutional Neural Network schematic. 16
2.4 Recurrent Neural Network schematic. 18
2.5 Autoencoder schematic. 20
2.6 Graph overview of spatial omics methods analysed in this review. 41

3.1 Overview of the PENGUIN tool. 46
3.2 Application of various percentiles and thresholds with PENGUIN. 47
3.3 SABR vs PENGUIN approach. 49
3.4 Overview of downstream analysis. 51
3.5 PENGUIN applied to IF images. 52
S3.1 Noise and real signal across four channels of interest. 56
S3.2 Processing with diverse strategies - CD20 channel. 57
S3.3 Processing with diverse strategies - β-catenin channel. 58
S3.4 Zoomed-in view from Figure 3.3. 59
S3.5 Boxplot comparing the PSNR and SSIM for RAW and PENGUIN-denoised images. 59
S3.6 Boxplot of maximum and 99th percentile values per channel. 59
S3.7 Histograms of the signal distribution across all channels before and after normalization. . 60
S3.8 Histograms of signal distribution for positive values across all channels. 61

4.1 Evaluation of Segmentation Methods on the test Set at ROI level. 68
4.2 Evaluation of Segmentation Methods on the test Set at pixel level. 69
4.3 Comparison of phenotyping, morphology, and marker intensity across cell segmentation methods. 71
4.4 Comparison of T cell quantification across segmentation methods and FOVs. 72
4.5 Visual comparison of segmentation and phenotyping results across methods. 74
S4.1 Comparison of ground truth masks and masks generated by Cellpose models. 80
S4.2 Pixel Evaluation metrics of segmentation generated by Cellpose models. 81

S4.3 Segmentation mask filtering steps 83
S4.4 Evaluation of the segmentation methods at ROI level on lymphoid cells. 84
S4.5 Evaluation of the segmentation methods at ROI level on myeloid cells. 85
S4.6 Evaluation of the segmentation methods at ROI level on Vimentin cells. 86
S4.7 Evaluation of the segmentation methods at ROI level on tumour cells. 87
S4.8 Schematics of the proposed decision tree method used for phenotyping. 88
S4.9 Cell count distribution by segmentation method and cell type. 89
S4.10Cell area distribution by segmentation method and cell type. 89
S4.11Cell area distribution across segmentation methods. 90
S4.12Distribution of sum of intensity across segmentation methods. 91
S4.13Number of total T cells per intraepithelial percentage. 92

5.1 Overview of the GRIDGEN tool. 99
5.2 GRIDGEN Contour identification strategies. 101
5.3 Tumour and Stroma analysis with GRIDGEN. 104
5.4 Population specific analysis with GRIDGEN. 106
5.5 Overlay of Cell Segmentation and Tumour-Stroma-interface Masks. 107
S5.1 Kernel size influence on GRIDGEN masks. 115
S5.2 Comparison of GRIDGEN population-specific masks and cell segmentation. 116
S5.3 Alternative ways of masking with GRIDGEN. 117

List of Tables

2.1 Commonly used terms in deep learning workflows. 14
2.2 Comparison table of key features of spatial proteomics profiling methods. 26
2.3 Comparison table of key features of spatial transcriptomics profiling methods. 29
2.4 Comparison table of key features of spatial multi-omics methods. 30

S3.1 Marker Percentiles and Thresholds. 62
S3.2 Thresholds and percentile values for the IF markers. 62

4.1 Number of FOVs and ROIs in training, test, and total annotation sets across the different mask types. 66
4.2 Segmentation evaluation metrics with descriptions and formulas. 82
S4.1 Pixel evaluation metrics on test set for Cellpose models. 93
S4.2 Comparison of ROI evaluation metrics on test set. 95
S4.3 Comparison on Pixel evaluation metrics, for segmentations on the test set. 96

Acronyms

AE	Autoencoder
ANN	Artificial neural network
AUC	Area Under the Curve
CAFs	cancer-associated fibroblasts
CCC	cell–cell communication
CITE	co-indexing of transcriptomes and epitopes
CNN	Convolutional Neural Network
CODEX	co-detection by indexing
CRC	Colorectal cancer
DGE	Differentially Gene Expression
DIMR	Differential Intensity Map-based Restoration
DL	Deep Learning
DNN	Deep Neural Network
DSP	Digital Spatial Profiler
DT	Decision Tree
ECM	extracellular matrix
FN	False Negative
FOV	Field-of-Views
FP	False Positive
GCN	Graph Convolutional Network
GNB	Gaussian Naive Bayes
GNN	Graph Neural Networks

GPU	Graphics Processing Units
GRIDGEN	Guided Region Identification based on Density of GENes
GRU	Gated Recurrent Unit
HSNE	Hierarchical Stochastic Neighbour Embedding
IBEX	iterative bleaching extends multiplexity
IF	immunofluorescence
IHC	immunohistochemistry
IMC	Imaging Mass Cytometry
Immuno-SABER	immunostaining with signal amplification by exchange reaction
ISH	*In Situ* Hybridization
ISS	*In Situ* Sequencing
K-D Tree	K-Dimensional-Tree
KNN	K-nearest neighbor
L-R	ligand–receptor
LgR	Logistic regression
LLM	Large Language Models
LR	Linear regression
LSTM	Long Short-Term Memory
MALDI-MSI	Matrix-Assisted Laser Desorption-Ionization Mass Spectrometry Imaging
MCC	Matthews Correlation Coefficient
MERFISH	Multiplexed error-robust FISH
MIBI-TOF	multiplexed ion beam imaging by time of flight
MIL	multiple instance learning
ML	Machine Learning
MLP	Multilayer Perceptron
MOFA	Multi-Omics Factor Analysis
MRF	Markov Random Field
NK	Natural Killer
NMF	Nonnegative Matrix Factorization
NSF	Nonnegative Spatial Factorization
P	Percentile

PCA principal component analysis
PENGUIN Percentile Normalization GUI Image denoising
PR precision-recall
PSNR Peak Signal-to-Noise Ratio

RF Random forest
RMSE Root Mean Squared Error
RNN Recurrent Neural Network
ROC Receiver Operating Characteristic
ROI Regions-of-interest

SABR semi-automated background removal
scRNA single-cell RNA
SD Standard Deviation
seqFISH Sequential fluorescence in situ hybridization
SGD Stochastic Gradient Descent
smFISH single molecule FISH
SOM Self-Organizing Maps
SSE Sum of Squared Errors
SSIM Structural Similarity Index
SVG Spatially variable genes
SVM Support vector machine

T Threshold
t-CyCIF tissue-based cyclic immunofluorescence
t-SNE T-distributed Stochastic Neighbor Embedding
TF transcription factor
TME tumour microenvironment
TN True Negative
TP True Positive

UMAP Uniform Manifold Approximation and Projection

XGBoost Extreme Gradient Boosting

1 Introduction

This thesis addresses the development of computational methods for the analysis of spatial omics data. It was carried out as part of a collaborative effort between the Bioinformatics and Systems Biology (BISBI) research group from the Centre of Biological Engineering (CEB) at the University of Minho and the Leiden University Medical Center (LUMC) in the Netherlands. This collaboration strongly shaped the direction of the work, as it was influenced by the hospital environment, where access to patient samples, clinical insight, and interdisciplinary expertise helped define both the scientific questions and technical approaches. This setting emphasized the clinical relevance of understanding cancer at the tissue level and motivated the creation of computational tools that can handle the complexity of spatial omics data in a way that aligns with real-world needs.

1.1 Context and motivation

Precision medicine is a defining endeavour of our generation, aiming to provide tailored therapies for specific groups of patients. Among its many challenges, cancer remains as one of the most complex. Tumours are not homogeneous masses of malignant cells, they are biologically intricate systems marked by multiple layers of heterogeneity, including genetic, epigenetic, cellular, and metabolic variation, even within the same tissue. Furthermore, tumours do not evolve in isolation; the tumour microenvironment (TME), composed of immune cells, fibroblasts, endothelial cells, and other stromal components, plays a crucial role in shaping tumour development, immune evasion, and therapeutic response [1, 2, 3].

Biology, by nature, is spatial. In living tissues, a cell's identity and function are determined not only by its molecular profile, but also by its physical location and its interactions with neighbouring cells. This spatial dimension is central to understanding how tissues function (or malfunction) in disease. Recent advances in spatial profiling technologies now allow researchers to measure gene and protein expression directly in situ, preserving spatial context. These methods offer multiscale insights, from subcellular features to tissue-level architecture, and have fundamentally changed how we study tissue biology and pathology [4, 5, 6, 7]. Spatial proteomics[8, 9, 10, 11] and spatial transcriptomics [12, 13, 14] technologies now enable the quantification of hundreds of proteins or RNA transcripts directly within intact tissues. These tools are reshaping our ability to investigate cellular heterogeneity, spatial organization,

and intercellular interactions, offering powerful opportunities for biomarker discovery, drug development, and patient stratification [6, 7, 15, 16].

While spatial omics holds tremendous promise, it also introduces significant computational challenges. The data produced is extremely large, high-dimensional, and complex — often noisy, and sparse. Computational methods for the analysis of such data are still under development, and significant challenges remain, including robust pre-processing, accurate cell segmentation, and flexible frameworks to extract meaningful spatial features. Addressing these challenges is essential to unlock the full potential of spatial omics and enable its integration into downstream predictive models [5, 15, 16, 17].

In parallel, machine learning (ML) and deep learning (DL) have rapidly advanced cancer research by enabling models that can learn meaningful patterns from raw and multimodal data [18]. DL approaches have been widely adopted in genomics, transcriptomics, and pathology image analysis, driving progress in areas such as diagnostics, prognostics, biomarker discovery, and therapy response prediction [18, 19, 20, 21, 22, 23, 24]. Yet, their application to spatial omics remains limited, mainly due to a lack of large, well-annotated datasets and the absence of robust computational pipelines capable of preparing spatial data for modelling.

Integrating spatial omics with modern computational frameworks is not just a need, it is a major opportunity. The ability to incorporate spatially resolved molecular information alongside advanced analytics can provide a more holistic view of cancer as a tissue-level disease. This integration is particularly critical for understanding key transitional phases of cancer, such as the shift from pre-malignant to invasive states, a process often overlooked in genomics-focused studies [6, 7, 15, 25].

1.2 Research Objective

Given the landscape outlined above, the main goal of this work is to develop robust computational methods for pre-processing and analysing high-dimensional spatial omics data, with a particular focus on enabling downstream integration with deep learning frameworks. We apply these methods in the context of colorectal cancer (CRC), aiming to support the development of personalized strategies for cancer prevention and treatment. By ensuring that input data are clean, well-segmented, and spatially meaningful, we lay the groundwork for future work in predictive modelling, including the identification of spatial biomarkers and inference of cell–cell communication networks.

This thesis presents novel contributions across the following scientific and technological objectives:

- To review the current landscape of spatial omics technologies applied to cancer research, as well as existing computational methods and tools, including recent advances in machine and deep learning relevant to this domain.

- To develop robust pre-processing pipelines for spatial proteomics data, addressing the challenges of noise, sparsity, and batch effects in highly multiplexed imaging data.

- To benchmark and evaluate existing cell segmentation strategies in spatial proteomics, including the creation of a manually annotated dataset to support segmentation validation and tool development.
- To develop novel methods for the analysis of spatial transcriptomics data, alternatively to cell segmentation strategies.
- To apply the developed tools to clinically relevant CRC case studies, demonstrating their utility in extracting biologically meaningful insights from real-world spatial omics datasets.
- To release the developed software as open-source, promoting reproducibility, and future extensibility by the scientific community.
- To publish the results of this work through peer-reviewed journal articles and international conferences, contributing to the advancement of knowledge at the intersection of computational biology, spatial omics, and cancer research.

1.3 Thesis outline

This thesis is organized into six chapters, a brief description of each one of them is given as follows:

- Chapter 1 introduces the topic, presents the motivation for the study, outlines the main objectives, and provides a summary of the thesis structure.
- Chapter 2 presents the background and theoretical foundations of the work, including key concepts in machine and deep learning, cancer biology, as well as an overview of spatial omics technologies and relevant computational methods.
- Chapter 3 introduces PENGUIN, a pre-processing framework for spatial proteomics data, and illustrates its application in a colorectal cancer case study.
- Chapter 4 provides a comparative analysis of cell segmentation strategies for spatial proteomics and describes the construction of a manually annotated CRC dataset for segmentation validation.
- Chapter 5 presents GRIDGEN, a method for extracting spatial region-level information from ultra-high dimensional spatial transcriptomics data, with applications in cancer tissue analysis.
- Chapter 6 concludes the thesis by summarizing the main findings, discussing scientific contributions, and outlining directions for future research.

2

Background

2.1 Machine and Deep Learning

In 1959, Arthur Samuel described Machine Learning (ML) as the "field of study that gives computers the ability to learn without being explicitly programmed" [26]. Since then, ML has become an essential component of information technology, and with the growing availability of data, it is reasonable to expect its influence will continue to expand [27]. ML is a branch of artificial intelligence that systematically applies algorithms to uncover underlying patterns within data [28].

Learning methods in ML are typically classified into two major categories: supervised and unsupervised learning. Other categories include reinforcement learning and semi-supervised learning. Each of these approaches is suited to different types of problems, depending on the nature of the input data and the desired outputs. Supervised learning focuses on inferring the relationship between input data and target variables to make predictions, whereas unsupervised learning seeks to extract patterns or structures directly from input data without requiring annotated labels [18, 29].

Another distinction exists between conventional ML, which relies heavily on manual feature engineering to design data representations, and more recent Deep Learning (DL) approaches, which aim to learn feature representations directly from raw data [18, 30].

In the field of computational biology, ML and DL algorithms have been widely applied, driven by the exponential growth of biological and clinical data, particularly following the advent of next-generation sequencing technologies [31, 32]. The application of these algorithms holds great promise for advancing our understanding of living systems, and numerous research groups are actively leveraging these methods to uncover novel biological insights [23].

2.1.1 Unsupervised learning

In unsupervised learning, the input data is unlabeled. Algorithms aim to discover similarities, structures, and hidden patterns within the data without external guidance [28]. These methods are often employed in exploratory data analysis and pre-processing to better understand the organizational structure of datasets [33]. Two of the most prominent unsupervised learning techniques are clustering and dimensionality

reduction.

2.1.1.1 Clustering

Clustering techniques group unlabeled data points based on similarity. Ideally, points within the same cluster are more similar to each other, while points from different clusters are more dissimilar within the relevant feature space. This approach enables segmentation of datasets, reduction of complexity, and identification of hidden structures or relationships. Common clustering algorithms include K-means clustering and hierarchical clustering [33].

Self-Organizing Maps (SOM) are a type of unsupervised neural network that organizes data into a lower-dimensional grid while preserving its topological properties. SOMs map high-dimensional input data into a 2D lattice of neurons, where similar data points are grouped together, making SOMs useful for visualization, pattern recognition, and clustering tasks [34]. By training the SOM on a dataset, it can cluster the data into regions, providing insights into the structure and distribution of the data.

2.1.1.2 Dimensionality reduction

Dimensionality reduction techniques aim to represent high-dimensional data in fewer dimensions, summarizing the essential characteristics of the original data into a smaller set of features. These techniques are useful for visualization, data compression, and improving subsequent analysis by focusing on the most informative aspects of the data [33].

Common dimensionality reduction algorithms include principal component analysis (PCA), Nonnegative Matrix Factorization (NMF), T-distributed Stochastic Neighbor Embedding (t-SNE), and Uniform Manifold Approximation and Projection (UMAP).

PCA generates new orthogonal latent features, called principal components, which capture the maximum variance in the data [32, 33]. In contrast, NMF is often used for feature extraction and interpretable parts-based representations, particularly when data is non-negative [35]. t-SNE [36] and UMAP [37] are widely employed for visualizing high-dimensional data in two or three dimensions, with UMAP providing better preservation of the global data structure and scalability for larger datasets compared to t-SNE.

2.1.2 Supervised learning

Supervised learning aims to infer the relationship between input data and a target variable, enabling prediction on unseen data [28, 33]. Depending on whether the target variable is discrete or continuous, supervised algorithms are categorized into classification and regression models, respectively.

Developing a supervised ML workflow involves several key steps: data collection and preparation, sampling, training and evaluating the model, and finally, model deployment.

2.1.2.1 Data collection and preparation

The first step of an ML pipeline is to obtain and prepare the data for model building and training. This process includes dataset construction, pre-processing, feature extraction, and feature selection.

A dataset is typically organized as a matrix, where each row represents an instance and each column represents an attribute (feature). One of the columns must represent the output (target variable), while the remaining columns represent the input attributes. Features can be binary, categorical, or continuous [38].

Feature extraction transforms raw data (such as text or images) into structured features suitable for modeling [33]. In some cases, dimensionality reduction techniques like PCA may also be used to compress high-dimensional data into a lower-dimensional, informative representation [39].

Pre-processing steps typically include formatting data consistently [28], handling missing or corrupted values, and standardizing feature scales when necessary [28, 40].

Feature selection, distinct from feature extraction, involves identifying and selecting a subset of relevant features that best represent the target concept [41]. Reducing the number of features can improve training speed, decrease model complexity, enhance generalization, and mitigate overfitting. Common approaches to feature selection include filter methods, wrapper methods, and embedded methods [42].

2.1.2.2 Sampling

Following data preparation, the dataset is typically divided into a training set and a test set. The training data are used to train the model, while the test data are reserved for evaluating the final model's performance.

When hyperparameter tuning or model selection is performed, an additional validation set is necessary. The validation set allows for model optimization without contaminating the test set, ensuring an unbiased final performance evaluation [28].

2.1.2.3 Training and evaluation

Once the data are prepared and sampled, the next step is to select an algorithm and train the model. During training, the algorithm uses the training set to learn the relationship between input features and the target variable. Performance is evaluated by comparing the model's predictions to the true target values using appropriate error metrics [40].

Evaluation in classification tasks In classification problems, a confusion matrix provides a summary of prediction outcomes, with rows corresponding to actual classes and columns to predicted classes [33, 43, 44]. It enables the calculation of key quantities:

- True Positive (TP): correctly predicted positive instances;
- False Positive (FP): incorrectly predicted positive instances;

- True Negative (TN): correctly predicted negative instances;
- False Negative (FN): incorrectly predicted negative instances.

From these quantities, various performance metrics can be derived:

- Accuracy measures the proportion of correctly predicted instances among all instances:

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

- Precision measures the proportion of correctly predicted positive observations among all predicted positives:

$$\text{Precision} = \frac{TP}{TP + FP}$$

- Recall (or sensitivity) measures the proportion of correctly predicted positive observations among all actual positives:

$$\text{Recall} = \frac{TP}{TP + FN}$$

- F1 score is the harmonic mean of precision and recall, providing a balance between the two, particularly useful when dealing with imbalanced datasets.
- Matthews Correlation Coefficient (MCC) is a balanced measure that considers all elements of the confusion matrix. It returns a value between -1 (total disagreement) and 1 (perfect prediction) and is suitable even for imbalanced datasets [45]:

$$MCC = \frac{TP \times TN - FP \times FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}}$$

Additionally, Receiver Operating Characteristic (ROC) curves and precision-recall (PR) curves are useful tools for evaluating model performance across different classification thresholds. ROC curves plot the True Positive Rate against the False Positive Rate, while PR curves plot Precision against Recall. The Area Under the Curve (AUC)quantifies the overall ability of the model to discriminate between classes, ranging from 0 (no discrimination) to 1 (perfect discrimination) [33, 43, 44].

ROC curves are generally more appropriate when classes are balanced, whereas PR curves are more informative when dealing with imbalanced datasets [46].

Evaluation in regression tasks In regression problems, evaluation is based on the difference between predicted and actual target values. Common metrics include:

- Sum of Squared Errors (SSE): the sum of the squared differences between predicted and actual values.
- Root Mean Squared Error (RMSE): the square root of the mean squared differences, with lower values indicating better performance.
- R-squared (R^2): the proportion of variance in the target variable explained by the model. It ranges from 0 (no explanatory power) to 1 (perfect fit) [33, 43, 44].

Cross-validation Cross-validation provides a more reliable estimate of model performance. In k-fold cross-validation, the dataset is partitioned into k subsets (folds). In each iteration, $k-1$ folds are used for training and the remaining fold for testing. This process is repeated k times, with the final performance estimate obtained by averaging the test results across all folds [33]. While highly effective, cross-validation can be computationally intensive for large datasets.

2.1.2.4 Hyperparameter optimization

Many ML algorithms have hyperparameters — settings defined before training that cannot be learned from the data. Hyperparameter optimization seeks the set of values that yields the best performance for a specific task [40, 44].

Common strategies include:

- Grid search: exhaustively trying all combinations of a predefined set of hyperparameter values;
- Randomized search: sampling combinations randomly from distributions over hyperparameter values.

Hyperparameter tuning typically requires an independent validation set or the application of cross-validation techniques within the training data [33].

2.1.2.5 Overfitting and underfitting

Two critical issues in model development are overfitting and underfitting. Overfitting occurs when a model is too complex and fits noise in the training data, leading to poor generalization to unseen data [40]. Underfitting happens when a model is too simple to capture the underlying data patterns. The goal is to develop models that balance bias and variance, achieving good generalization performance [33].

2.1.3 Traditional Supervised Machine Learning Algorithms

Supervised machine learning algorithms learn patterns from labeled data to make predictions on new, unseen data. Various algorithms exist, each employing different strategies to identify these patterns and build predictive models. This section provides an overview of several well-known and widely used traditional supervised learning algorithms, highlighting their core principles and characteristics.

Linear regression (LR) is one of the simplest supervised ML algorithms for solving regression problems. It finds the best-fitting linear combination of the input variables that can accurately predict the output for a continuous dependent variable. Regularization techniques such as L1 (Lasso) or L2 (Ridge) can be used to prevent overfitting and manage multicollinearity between features.

Logistic regression (LgR) is a linear model for classification where the probabilities describing the possible outcomes of a single trial are modeled using a logistic function that maps values between 0 and 1 [47]. It is based on the concept of maximum likelihood estimation. Logistic models are fast to train and predict, work well with large datasets and sparse data, and are intuitive to interpret. However, when highly

correlated features are present, the model coefficients can be difficult to interpret, and logistic models do not capture nonlinear relations unless interaction terms or transformations are added [33].

Support vector machine (SVM) generates optimal hyperplanes that can maximize the distance between the samples of different classes. The distance to the hyperplane in a higher-dimensional space can be computed using several different kernel functions (e.g., linear, RBF). These models are powerful, work well with sparse data, and are less prone to overfitting. However, they require careful pre-processing of the data to avoid issues with scaling and feature normalization [33].

K-nearest neighbor (KNN) is an instance-based algorithm that classifies a query sample based on the majority class of its nearest neighbors. It computes distances (e.g., Euclidean) between the samples and assigns the label according to the closest ones [47]. Important parameters include the number of neighbors considered and the choice of distance metric. The method is sensitive to the feature scale and tends not to perform well on datasets with many features (curse of dimensionality) [33]. Furthermore, it requires storing the entire dataset, which makes it computationally expensive, especially for large datasets [47].

Gaussian Naive Bayes (GNB) is a probabilistic classification algorithm based on Bayes' theorem, assuming feature independence. Despite its simple and often unrealistic assumptions, it has shown competitive performance in many classification tasks [33]. Gaussian Naive Bayes models the likelihood of the features assuming a normal distribution. They are similar to linear models but tend to be even faster to train and predict, handle large and high-dimensional data well, and are relatively robust to parameter tuning.

Decision Tree (DT) are flowchart-like structures that learn a hierarchy of if/else questions leading to a decision. These models are particularly effective at capturing non-linear relationships. Although they are intuitive and interpretable, they do not tend to perform well on high-dimensional sparse data but work well on large datasets. In this algorithm, the maximum depth of the tree is a critical parameter to tune to avoid underfitting and overfitting [33].

2.1.3.1 Ensemble models

Using multiple models may achieve better performance than using a single model. Ensemble methods combine multiple independent machine learning models to create a more powerful and robust algorithm. These methods typically aim to reduce variance, bias, or both, improving predictive performance compared to single models. Ensemble methods can be broadly categorized into two main types: Bagging and Boosting [40, 43, 48, 49].

Bagging involves training multiple models independently and then combining their predictions (either by averaging for regression tasks or by voting for classification tasks). The main goal is to reduce variance by leveraging multiple independent models. Unlike bagging, boosting is a sequential method where each new model is trained to correct the errors made by the previous ones. In boosting, models are trained sequentially, and more weight is given to instances that were misclassified by earlier models. This iterative correction leads to reduced bias and improved performance.

Random forest (RF) is a popular bagging method. It builds a collection of decision trees, where each tree is trained on a random subset of the data, and the features for each split are randomly selected. This helps improve model robustness and reduces overfitting compared to a single decision tree. Key hyperparameters in RF include the number of trees and the maximum depth of each tree [33, 50].

Extreme Gradient Boosting (XGBoost) is one of the most popular boosting algorithms due to its efficiency and high predictive power, especially with large datasets [51]. In XGBoost, models are trained in a series, where each new decision tree focuses on the errors made by the previous trees. Key hyperparameters in XGBoost include the learning rate, number of boosting rounds, and maximum depth of each tree.

While the examples above utilize decision trees as base learners, the principles of bagging and boosting can be applied to other types of models as well. Ensemble methods, in general, are powerful tools for improving model generalization and reducing the risk of overfitting, particularly when dealing with complex datasets.

2.1.3.2 Artificial Neural Networks

Artificial neural network (ANN)s are a class of machine learning algorithms inspired by the structure and functioning of the human brain, and they can be used for a variety of tasks, including both regression and classification.

They consist of interconnected nodes or "neurons," which process information and learn patterns from data. The architecture of an ANN typically includes an input layer, one or more hidden layers, and an output layer [18, 40]:

- Input Layer: The leftmost layer, which receives the input features.
- Hidden Layer(s): These layers perform computations and transformations on the input data, enabling the model to learn complex patterns.
- Output Layer: The final layer that produces the model's predictions, with one neuron for regression or binary classification tasks, or a number of neurons equal to the number of classes for multiclass classification tasks.

A Multilayer Perceptron (MLP) is the simplest type of ANN and consists of multiple layers, including one or more hidden layers, that enable it to learn complex relationships between input and output. MLPs are feedforward networks, meaning information moves in one direction from input to output. They are particularly effective for supervised learning tasks like regression and classification [18, 52].

The basic building block of an ANN is the neuron, which receives inputs either from the dataset or from other neurons, computes a weighted sum, adds a bias, and applies a non-linear activation function to produce an output. The weights and biases are learned during the training process and are key to the model's ability to make accurate predictions. The bias allows the model to shift the activation function and better capture patterns that don't naturally pass through the origin.

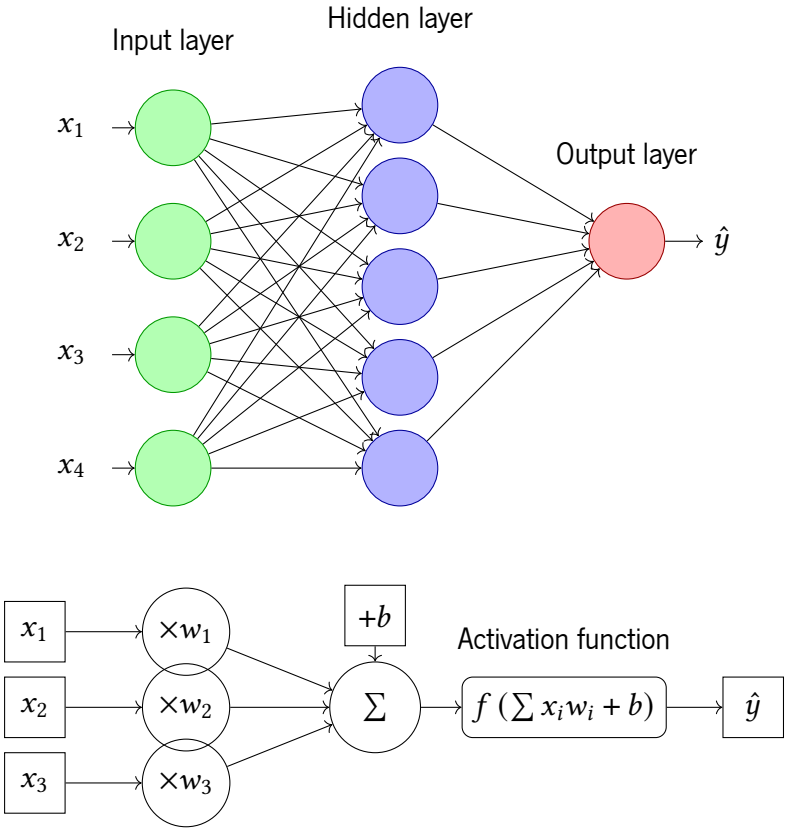


Figure 2.1: Schematic of a simple ANN and a single neuron. (Top) Schematic of a simple ANN with input, 1 hidden, and output layers. The structure illustrates the flow of information from the input with 4 neurons to the output (a single neuron) through the hidden layer (with 5 neurons). (Bottom) Diagram of a single neuron with n inputs x, weights w and bias b, activation function f and output y. The output of the neuron is obtained by applying a nonlinear transformation to the weighted sum of inputs: $y = f\left(\sum x_i w_i + b\right)$ where f is the activation function. Diagram created using TikZ.

Figure 2.1A illustrates the architecture of an ANN with one hidden layer. In the figure, each neuron in the hidden layer computes a weighted sum of its inputs and passes this sum through an activation function. This non-linearity is crucial because it allows the model to learn from a variety of complex data patterns.

Activation functions are mathematical functions that introduce non-linearity into the model. Without these functions, the network would simply behave as a linear model, unable to learn complex data relationships. Common activation functions include:

- Sigmoid: Produces outputs between 0 and 1, useful for binary classification problems.
- Tanh (Hyperbolic Tangent): Outputs values between -1 and 1, making it better for handling both positive and negative inputs.
- ReLU (Rectified Linear Unit): Outputs the input value if it's positive, and zero otherwise. This function is the most widely used in deep networks due to its ability to speed up training and reduce the likelihood of vanishing gradients [18, 47].

Figure 2.1B presents a scheme of one neuron with respective inputs x and weights w, bias b, weighted sum function activation function f and output y.

The training process for an ANN involves two key steps: forward propagation and backpropagation.

- Forward Propagation: The input data passes through the network, and each layer performs computations. The outputs of the network are compared with the true labels using a loss function (e.g., Mean Squared Error for regression or Cross-Entropy for classification).

- Backpropagation: After calculating the error (or loss), backpropagation helps update the model's parameters (weights and biases). It uses the gradient of the loss function with respect to each weight, which is calculated using the chain rule. This gradient is then used to adjust the weights in the direction that minimizes the loss [52]. Backpropagation, while essential for updating weights, also faces challenges such as local minima or saddle points in the loss function, which can affect the efficiency of the optimization process [30].The most common optimization algorithms used for this process include Stochastic Gradient Descent (SGD) and the Adam optimizer [18, 53].

In the training process, the model's weights are initialized randomly, and the forward and backward propagation steps are repeated over multiple iterations (or epochs) to refine the model's weights and minimize the loss. The optimization process continues until the model reaches a satisfactory level of accuracy, or a stopping criterion is met.

When ANNs contain more than one hidden layer, they are referred to as Deep Neural Network (DNN)s, that will be covered in the next section.

2.1.4 Deep learning

DL is a subfield of ML that focuses on the use of multi-layered ANNs. Compared to traditional shallow ML models, DL architectures are characterized by greater depth, with multiple layers and millions of trainable parameters, allowing for higher model capacity and flexibility. One of the main advantages of DL is its ability to automatically learn features and complex patterns from data, reducing the need for extensive feature engineering by domain experts. However, DL models require large amounts of labeled data, are more computationally intensive, and are often harder to interpret [18, 30].

A deep neural network can be viewed as a mathematical function built by composing multiple simpler transformations, organized in layers. Each layer receives the outputs of the previous layer as its inputs, allowing the model to build increasingly abstract representations of the data. Stacking these transformations enables the modeling of highly complex relationships, while still allowing the system to be trained using efficient algorithms such as backpropagation [30].

Developing DL models requires managing and tuning a large number of hyperparameters, including the number and type of layers, the number of units per layer, the learning rate, and the optimization algorithm, among others. Modern Python libraries such as PyTorch [54] and TensorFlow [55] have greatly simplified the implementation and training of DL models. Although training deep models can be computationally expensive, Graphics Processing Units (GPU) are commonly employed to parallelize computations

and accelerate training times. The main concepts and workflow is described below and summarized in Table 2.1.

As discussed previously, DL builds upon the foundations of ANNs. While a typical ANN may have a single hidden layer, a DNN is distinguished by having multiple hidden layers stacked sequentially. The basic operation of neurons and layers in DNNs remains the same as described earlier: neurons compute weighted sums of their inputs, apply an activation function, and pass the result to the next layer.

2.1.4.1 Training process

Training a deep learning model begins by initializing the model's weights randomly. A loss function is used to measure the discrepancy between the model's predictions and the true target values. To minimize this loss, the model's weights are updated iteratively based on the computed error.

Usually, a SGD optimization algorithm is used. This method consists of computing the outputs, errors and the average gradient for just a few examples and adjusting the weights accordingly. Small sets of examples from training set are used until the average of loss stops decreasing. This way, each small set of examples give a noisy estimate of the average gradient for all training set. The batch size is the hyperparameter that defines the number of samples that are used before updating the internal model parameters [56].

To compute gradients across multiple layers, the backpropagation algorithm is employed. Backpropagation systematically applies the chain rule of calculus to propagate error gradients from the output layer back through the network, enabling efficient computation of parameter updates [18].

An important hyperparameter in this process is the learning rate, which controls the magnitude of the weight updates. A high learning rate may cause the model to converge too quickly to a suboptimal solution, while a very low learning rate may slow down training.

In addition to classical SGD, more advanced optimization algorithms such as Adam are widely used. Unlike SGD, which maintains a single learning rate throughout training, Adam adapts the learning rate for each parameter by maintaining an exponentially decaying average of past gradients and squared gradients [53]. The decay rates of these averages are controlled by the β_1 and β_2 parameters.

2.1.4.2 Regularization techniques

Due to their high capacity, deep learning models are particularly prone to overfitting. Several regularization strategies are commonly employed to promote generalization:

- **L1 and L2 Regularization:** These methods add a penalty term to the loss function based on the magnitude of model weights. L1 regularization (Lasso) encourages sparsity by penalizing the absolute values of the weights, while L2 regularization (Ridge) penalizes the squared values of the weights, discouraging excessively large weights [29, 30].

- **Dropout:** In each iteration of training, a random subset of neurons is deactivated, forcing the network to develop redundant representations and reducing overfitting [57].

Table 2.1: Commonly used terms in deep learning workflows, adapted from [23].

Concept	**Definition**
Activation Function	Mathematical function f applied to the neuron's input $w^T x + b$ to produce its output.
Backpropagation	Algorithm for computing gradients and updating weights by propagating the loss backward through the network.
Batch	Subset of the training data used to compute a single update of the model weights.
Batch Size	Number of samples included in each batch.
Dropout Rate	Percentage of neurons randomly deactivated at each training iteration.
Early Stopping	Technique to halt training when performance on a validation set stops improving.
Epoch	One complete pass through the entire training dataset.
Kernel	Multidimensional array of weights (especially used in convolutional layers).
Learning Rate	Step size at each iteration during optimization.
Loss	Function that measures the discrepancy between predicted and actual target values.
Neuron	Basic computational unit of a deep learning model.
Optimizer	Algorithm used to update the weights and biases to minimize the loss function.
SGD (Stochastic Gradient Descent)	Optimization algorithm that updates model parameters using mini-batches of data.
Training	Process of learning model parameters from data.

- **Early Stopping:** Training is halted when performance on a validation set stops improving, preventing the model from overfitting the training data [58].
- **Data Augmentation:** When data is limited, artificial samples are created by applying transformations such as flipping, rotation, or translation to existing examples. This technique is widely used in domains like computer vision but is not universally applicable across all data types [59].

2.1.4.3 Transfer learning

Another important concept is the one of transfer learning. Transfer learning is an approach that allows to transfer the knowledge of a pre-built and pre-trained classifiers successfully trained on a large dataset to a newly created one, typically a smaller dataset. This allows to create robust models with limited data, reduces training time and reduces the computational power required. The process that allows this transfer of knowledge is the fine-tuning. Fine-tuning freezes part of a network (typically the first layers) to extract

generic low-level features and modifies the last few layers to focus on specific features that are relevant to the new dataset [60].

2.1.5 Architectures of deep neural networks

DL is characterized by its use of multiple layers of neurons, each contributing to the model's ability to learn complex patterns and representations from data. While all DL methods share the fundamental principle of stacking multiple layers, various architectures have been developed to address different types of problems. The most prominent architectures include DNNs, Convolutional Neural Network (CNN)s, Recurrent Neural Network (RNN)s, Autoencoder (AE)s and Graph Neural Networks (GNN)s. Each of these architectures leverages distinct mechanisms to process data and make predictions, tailored to specific use cases.

The following sections provide an overview of each of these popular architectures. Other less commonly used but emerging architectures, such as generative models and transformers are outside the scope of this discussion.

2.1.5.1 Fully Connected Deep Neural Networks

Fully connected DNNs (Figure 2.2) represent one of the most fundamental and widely used DL architectures. These networks extend the basic structure of ANNs by incorporating multiple hidden layers, thus enabling the model to capture more intricate relationships in data. In a fully connected DNN, each neuron in one layer is connected to every neuron in the subsequent layer, forming a dense network of interconnections. This structure facilitates the learning of hierarchical features at multiple levels of abstraction [18, 30, 52].

Despite their simplicity, fully connected DNNs are often prone to overfitting due to their large number of parameters. When trained on small datasets, these networks can learn highly specific patterns that do not generalize well to unseen data. To mitigate this, regularization techniques such as dropout, L1/L2 regularization, and early stopping are commonly employed to prevent the model from memorizing the training data and to improve its generalization ability [61].

2.1.5.2 Convolutional neural networks

CNNs (Figure 2.3) are specialized deep neural networks primarily used for analyzing images or other types of multidimensional data. They have proven to be particularly effective in tasks such as image recognition, object detection, and classification [29, 62]. CNNs can process input variables that are distributed along a spatial pattern, with one, two, or three dimensions, depending on the complexity of the input data. Unlike fully connected DNNs, CNNs incorporate several distinct types of layers, including convolutional and pooling layers. These layers allow CNNs to effectively learn hierarchical features and represent spatial relationships.

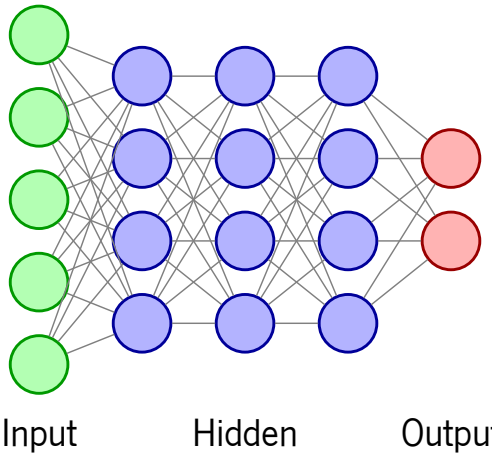


Figure 2.2: Schematic representation of a fully connected feedforward deep neural network created using TikZ. The network consists of an input layer with five neurons, three hidden layers with four neurons each, and an output layer with two neurons. Each neuron in a layer is connected to all neurons in the subsequent layer.

Convolutional layers act as feature extractors, identifying essential patterns in the input data. Pooling layers, on the other hand, help reduce the dimensionality of the feature maps, which improves computational efficiency and helps prevent overfitting. The final layers of CNNs are typically dense layers, which are responsible for the classification of the learned patterns [18, 62].

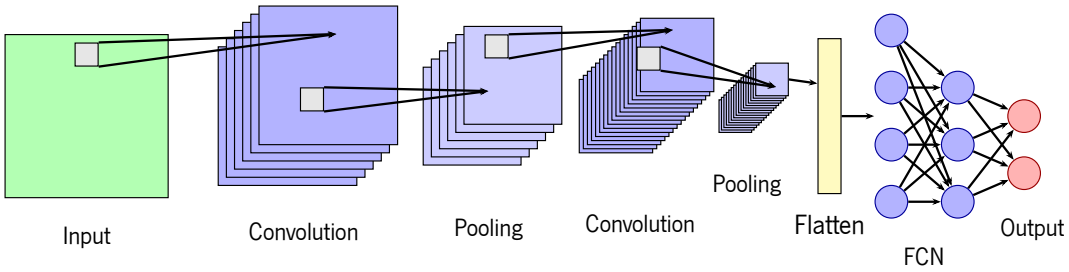


Figure 2.3: Schematic of a CNN architecture, illustrating key layers from input to output. The network begins with an input image, followed by a convolutional layer, a pooling layer, a second convolutional layer, and another pooling layer. These are followed by a flattening operation, two fully connected layers, and a final output layer with two neurons. Convolution and pooling operations are represented with stacked feature maps and filters. All components were created using TikZ.

The convolutional operation In contrast to fully connected layers, which learn global patterns in the input space, convolutional layers are specifically designed to learn local patterns by applying convolutional operations. A convolution operation involves a filter (also referred to as a kernel) that is applied across the input feature map, which is typically a 3D tensor consisting of spatial dimensions (height and width) and depth (channels, such as RGB color channels in images) [18, 29, 62].

The convolutional filter is slid across the input feature map in a process known as striding, which allows the network to capture spatially localized patterns. Each filter is responsible for extracting certain features, such as edges, textures, or more complex patterns, depending on the depth of the network.

As the filter slides over the input, it produces a new 3D tensor, known as the output feature map. After each convolutional operation, an activation function, typically ReLU, is applied element-wise to introduce non-linearity [62].

This process of convolving and applying activation functions enables the network to learn increasingly abstract and complex features as it moves through multiple layers. The resulting feature maps encode the spatial "presence" of various learned patterns across the input data.

The pooling operation After convolutional layers, pooling layers are typically applied to downsample the feature maps. Pooling reduces the spatial dimensions of the input feature maps, retaining the most important information while discarding less relevant details. The most commonly used pooling operation is max pooling, where the maximum value from a specified window (or region) of the input feature map is selected and passed onto the next layer. This operation helps to reduce the dimensionality and computational load of the network.

In addition to max pooling, another variant is average pooling, which computes the average of all values in the window. While average pooling can be used, max pooling has been shown to yield better performance, particularly in tasks like image classification [29]. Pooling operations allow the network to induce spatial hierarchies, where successive layers focus on progressively larger receptive fields, making it easier for the model to capture more abstract features as it deepens.

2.1.5.3 Recurrent neural networks

RNNs (Figure 2.4) are specialized neural networks designed to model sequential data, making them particularly suited for tasks involving temporal dependencies, such as time series analysis, speech recognition, and natural language processing [29, 30, 63]. Unlike traditional feedforward networks, RNNs contain recurrent feedback loops within their layers, which allow them to maintain an internal state. This "memory" enables RNNs to process sequences by utilizing past information while integrating current inputs, making them effective for tasks that require context from previous elements in the data [29].

However, a major limitation of standard RNNs is their difficulty in handling long-term dependencies. Specifically, RNNs suffer from the vanishing gradient problem, where the gradients (used for updating weights during training) shrink exponentially as they propagate through time, causing the network to forget information from earlier time steps. This results in RNNs having short-term memory and failing to capture long-term dependencies in sequences [64]. To address this issue, more advanced architectures, such as Long Short-Term Memory (LSTM) [63, 65] networks and Gated Recurrent Unit (GRU) [66], have been developed to better handle long-range dependencies.

Long Short Term Memory Networks: LSTM networks, introduced by Hochreiter and Schmidhuber [65], are a type of RNN specifically designed to overcome the vanishing gradient problem. The key feature of LSTMs is their ability to retain information over long time periods, thanks to their cell state, which serves as a memory of the network. The cell state is updated by three gates—forget, input, and output gates—that regulate the flow of information into and out of the network. These gates learn during training what

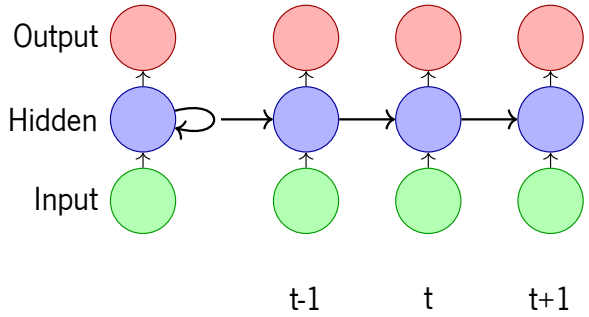


Figure 2.4: Schematic representation of a RNN, showing both the compact form (left) and the unrolled computational graph over time steps $t-1$, t, and $t+1$. The diagram illustrates how the hidden state is passed across time steps, enabling the network to retain temporal information. Created using TikZ.

information is essential to retain or discard [18, 63, 65]. The operation of an LSTM network can be summarized as follows:

- **Forget Gate:** The forget gate determines which information should be discarded from the cell state. It takes the previous hidden state and the current input, and applies a sigmoid function to generate a value between 0 and 1. A value close to 0 indicates that the information should be forgotten, while a value close to 1 indicates that the information should be retained. This output is then used in a pointwise multiplication operation with the cell state to eliminate unnecessary information.

- **Input Gate:** The input gate controls which new information is relevant and should be added to the cell state. This gate first applies a sigmoid function to the previous hidden state and current input, transforming the values between 0 (important) and 1 (not important). Then, the tanh function is applied to generate a weighted vector of potential new values. These new values are combined with the outputs from the sigmoid function to decide what should be added to the cell state.

- **Cell State Operation:** The previous cell state is multiplied pointwise by the output of the forget gate, allowing the network to decide which information should be discarded. The result is added to the output of the input gate, updating the cell state with new information. This operation allows the network to adjust its memory over time based on the incoming sequence data.

- **Output Gate:** The output gate determines what the next hidden state should be. It first applies a sigmoid function to the previous hidden state and the current cell state. The result is passed through the tanh function, which squashes the values between -1 and 1. Finally, the outputs of the sigmoid and tanh functions are combined through a pointwise multiplication to decide what information will be carried forward as the hidden state for the next time step.

For some types of sequences, such as biological sequences, a bidirectional approach can improve model performance. In bidirectional LSTM networks, two LSTMs are trained simultaneously: one processes the sequence from start to end, while the other processes the sequence from end to start. This

dual processing allows the model to learn from both past and future context, improving its ability to capture relevant information [29].

Gated Recurrent Unit Networks: GRUs networks are a simplified version of LSTMs. They have two gates: the reset gate and the update gate [29, 66].

- **Update gate:** This gate controls how much of the previous information should be retained and how much of the new information should be incorporated. Essentially, it combines the functions of the forget and input gates in an LSTM, deciding what to keep from the past and what new information is important to add. It also merges the cell state and the hidden state into a single hidden state variable, simplifying the memory mechanism.
- **Reset gate:** The reset gate determines how much of the past information should be discarded. It helps the network forget irrelevant parts of the sequence, allowing the GRU to focus on the most relevant inputs at each time step [29, 66, 67].

Because GRUs use fewer gates and simpler calculations, they generally require fewer tensor operations, making them faster to train compared to LSTMs. However, LSTMs are still preferred when dealing with longer sequences because they have a better ability to capture long-range dependencies and remember distant relationships in the data.

GRUs strike a balance between efficiency and performance, making them ideal for tasks where computational resources are limited or when training speed is critical[29].

2.1.5.4 Autoencoders

AEs (Figure 2.5) are a type of unsupervised neural network designed to learn efficient representations of input data, typically for the purpose of dimensionality reduction. The key idea behind an AE is that the network learns to reconstruct its inputs through a compressed, low-dimensional latent representation. This process creates a ”bottleneck” in the network, where the data must pass through a restricted hidden layer. The result is a compressed latent-space representation that captures only the most essential features of the data, enabling it to perform tasks such as dimensionality reduction and feature extraction for subsequent ML or DL models [18, 30, 68].

The architecture of a basic autoencoder consists of several key components [68]:

- **Encoder:** The encoder maps the input data into a lower-dimensional latent representation. This process compresses the data, removing redundancy and focusing on the most important features.
- **Bottleneck layer:** This layer is the core of the autoencoder, where the data has its lowest dimension. The bottleneck holds the latent-space representation, which contains the essential information required to reconstruct the original input.
- **Decoder:** The decoder attempts to reconstruct the input data from the compressed latent-space representation, working to approximate the original input as closely as possible.

- **Reconstruction loss:** A loss function is used to evaluate the model's performance by comparing the output of the decoder to the original input. The goal is to minimize this reconstruction error, ensuring that the autoencoder accurately reconstructs the input data.

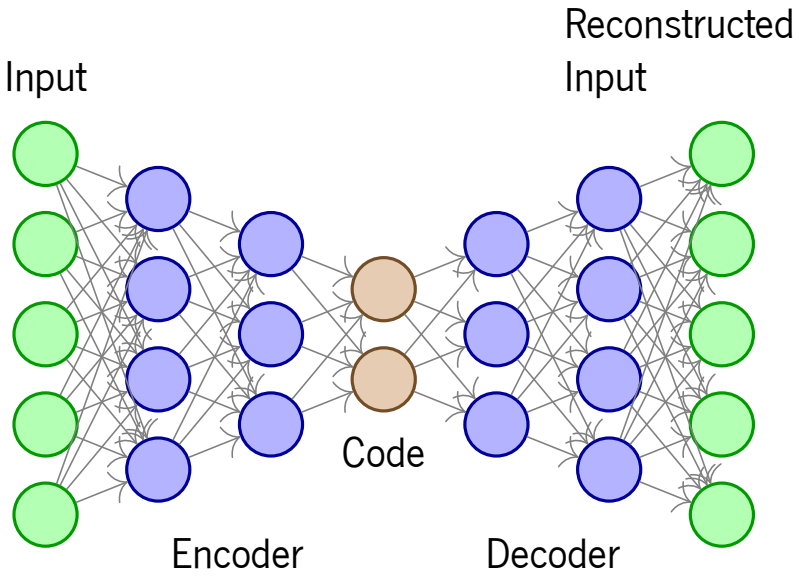


Figure 2.5: Schematic representation of an autoencoder architecture created using TikZ. The model consists of an encoder that compresses the input into a lower-dimensional latent space (code), followed by a decoder that reconstructs the original input from this compressed representation.

In addition to the basic autoencoder, several advanced variants exist, such as variational autoencoders and adversarial autoencoders, which incorporate probabilistic and adversarial techniques to enhance their capability for tasks like generative modeling and robust feature learning. In addition to the basic autoencoder, other advanced variants exist. For example, variational autoencoders introduce probabilistic reasoning into the encoding process [69], while adversarial autoencoders leverage adversarial training to improve the quality of learned representations [70]. However, these variants are beyond the scope of this review.

2.1.5.5 Graph Neural Networks

GNNs are a class of DL models specifically designed for processing graph-structured data. Unlike traditional neural networks that operate on grid-like data, GNNs excel at capturing the complex relationships and dependencies between entities in a graph. In a graph, entities are represented as nodes, and their relationships are represented as edges [71, 72]. This structure makes GNNs particularly valuable in a range of domains, including social network analysis, recommendation systems, and natural language processing. However, GNNs are also emerging as transformative tools in computational biology. They have demonstrated strong potential in applications such as protein interaction networks, image segmentation, and digital pathology, where they help analyze complex biological data at various scales [73, 74, 75].

Furthermore, GNNs are increasingly being applied to knowledge graphs in biological research to represent relationships between biological entities and enable deeper insights into complex systems [76].

At the core of GNNs lies the idea of message passing, where information is passed between nodes in the graph. Each node aggregates information from its neighbors to update its own state. The aggregation process can be done in several ways, such as summing, averaging, or applying more complex functions. This process is typically repeated for several iterations (or layers), allowing nodes to collect information from a wider neighborhood.

A standard GNN model can be described in the following components:

- **Node Features:** Each node in the graph has associated features (such as text, numerical values, etc.). These features are used as the starting point for learning.
- **Edge Features:** Sometimes, edges may also have features that define the type or weight of the relationship between nodes.
- **Message Passing:** During each iteration, each node aggregates information from its neighbors. This aggregation could be based on the type of edge, the node's own features, or both.
- **Graph Convolution:** The aggregation step is typically followed by a non-linear activation function, such as ReLU, and potentially another transformation through a neural network layer. This process is referred to as graph convolution.
- **Graph Pooling:** After several iterations of message passing, a pooling operation can be applied to extract a fixed-size graph representation, which can be useful for tasks such as classification or regression.

One of the most prominent GNN architectures is the Graph Convolutional Network (GCN), introduced by Kipf and Welling [72], which defines graph convolutions by a localized first-order approximation of spectral graph convolutions. Other influential models include GraphSAGE [77], which generalizes the message-passing framework by allowing nodes to aggregate information from a fixed-size neighborhood, and Graph Attention Networks [78], which use attention mechanisms to weigh the importance of neighboring nodes.

Despite their growing success, GNNs are still a relatively recent and developing area of research. As such, several challenges remain, particularly with respect to scaling GNNs to large graphs, handling dynamic or temporal graph data [71], and improving their interpretability [79, 80].

2.1.6 Self-supervised learning

Self-supervised learning (SSL) has gained significant attention as a powerful approach for utilizing unlabeled data. Unlike traditional supervised learning, where models learn from labeled datasets, SSL generates pseudo-labels directly from the data itself, allowing the model to learn useful representations without

relying on manual annotations. This approach has shown notable success in fields such as natural language processing and computer vision, and is becoming increasingly important for tasks where labeled data is scarce or difficult to obtain [81].

A prominent example of SSL in image processing is Noise2Void, a method designed for image denoising that predicts missing or corrupted pixels using the surrounding context of the image, without needing ground-truth noise-free data [82].

Another well-known self-supervised approach is contrastive learning, which has become a popular method for learning representations by comparing data points in an unsupervised manner [81]. Contrastive learning encourages the model to bring similar data points closer in representation space while separating dissimilar points.

While self-supervised learning has gained significant attention in recent years and offers promising approaches, especially for tasks with limited labeled data, it is beyond the scope of this review.

2.2 Cancer: definitions, microenvironment and heterogeneity

Cancer is a heterogeneous disease, characterized by the uncontrolled proliferation of abnormal cells that can invade nearby tissues and metastasize to distant organs. It arises from accumulated mutations due to genetic, epigenetic, and environmental factors that disrupt cellular homeostasis [1, 2]. According to the World Health Organization, cancer accounted for nearly 10 million deaths in 2020, with Colorectal cancer (CRC) comprising approximately 1.93 million new cases, making it the third most commonly diagnosed cancer and second leading cause of cancer-related death globally [83, 84, 85].

Biologically, tumorigenesis involves a multistep process of progressive transformation, whereby normal cells acquire capabilities to sustain growth, evade death, and resist immune attack. These capabilities are widely conceptualized as the "Hallmarks of Cancer", a set of core traits and enabling mechanisms that make the cancer cells [1, 2]. The core traits are:

1. Sustaining proliferative signalling
2. Evading growth suppressors
3. Resisting cell death
4. Enabling replicative immortality
5. Inducing or accessing vasculature (angiogenesis)
6. Activating invasion and metastasis
7. Reprogramming cellular metabolism
8. Avoiding immune destruction

These traits are supported by enabling characteristics, which further drive malignancy:

1. Genome instability and mutation
2. Tumour-promoting inflammation
3. Unlocking phenotypic plasticity
4. Non-mutational epigenetic reprogramming
5. Polymorphic microbiomes
6. Senescent cells

Cancer is also defined by its heterogeneity, which manifests at multiple levels: inter-patient, intra-tumoral, and inter-lesional [86, 87]. This structural and functional complexity contributes to varied therapeutic responses among patients and cancer types. As such, comprehensive characterization of tumour heterogeneity is critical for the development of effective therapies and the advancement of personalized medicine.

Besides their functional characteristics, cancers are largely influenced by the environment around them.

2.2.1 Tumour microenvironment

Once considered a passive backdrop to cancer cells, the tumour microenvironment (TME) is now recognized as an active participant in tumour progression, therapy resistance, and metastasis [88]. The TME is composed of a diverse cellular and non-cellular milieu including immune cells, cancer-associated fibroblasts (CAFs), endothelial cells, pericytes, extracellular matrix (ECM), and various soluble factors [2, 88, 89]. These components engage in dynamic, bidirectional communication with cancer cells, modulating proliferation, angiogenesis, immune evasion, and metastatic dissemination. Importantly, the spatial arrangement of these cells within tumours shapes clinical outcomes and therapeutic responses [88]. Besides the cancer cells, typically the TME includes:

2.2.1.1 Immune cells

Immune cells fall into two categories: adaptive immune cells and innate immune cells. Adaptive immunity is activated by exposure to specific antigens and uses an immunological memory to 'evaluate' the threat and enhance immune responses. T cells, B cells and Natural Killer (NK) cells belong to the adaptive immune response. Innate immunity is a non-specific defense mechanism that comes into play within hours of a foreign antigen entering the body. Cells that carry out an innate immune response include macrophages, neutrophils and dendritic cells. In the TME, immune cells play a dual role in the TME, either suppressing or promoting tumour growth [88].

Immune cells within the TME operate along a functional spectrum—ranging from tumour-suppressive (e.g., CD8+ T cells, dendritic cells, NK cells) to tumour-promoting (e.g., M2 macrophages, regulatory

T cells, myeloid-derived suppressor cells). Their distribution and activation states define the tumour's immunological landscape, which can be classified into:

- Immune-infiltrated: active immune response, high intratumoral immune presence;
- Immune-excluded: immune cells restricted to the tumour periphery;
- Immune-desert: paucity of immune infiltration and limited immune activation [88, 90].

In CRC, a high infiltration of Th1 cells, M1 macrophages, and T follicular helper cells, CD8+T, DCs and NK cells correlates with better prognosis, while M2 macrophages, Th17 cells, and B cells, in certain contexts, are linked to poor outcomes [91]. To translate the prognostic value of immune cell localization into clinical practice, the Immunoscore, which quantifies CD3 and CD8 T cells at the tumour core and invasive margin, has been validated as a robust prognostic tool in CRC and is under consideration for clinical adoption [92, 93].

2.2.1.2 Stromal cells

Stromal cells, including vascular endothelial cells and fibroblasts, play a key role in tumour development. Upon recruitment to the TME, they secrete factors that support angiogenesis, proliferation, invasion, and metastasis [88].

CAFs are a major stromal component, originating from resident fibroblasts, mesenchymal stem cells, or via trans-differentiation of epithelial/endothelial cells. In contrast to normal fibroblasts, CAFs in the TME acquire pro-tumorigenic phenotypes—marked by the secretion of ECM components, growth factors, cytokines, and matrix-remodelling enzymes such as MMPs. CAFs contribute to ECM remodelling, tissue stiffness, angiogenesis, and immunosuppression, often establishing niches that facilitate tumour immune evasion [94, 95].

Endothelial cells form the vascular architecture within tumours, enabling not only nutrient delivery but also serving as conduits for immune cells and metastatic spread. In tumour endothelial cells often exhibit abnormal morphology and permeability [2, 88].

2.2.1.3 Non-cellular components of the TME

The non-cellular matrix of the TME includes soluble factors (e.g., cytokines, chemokines, and growth factors) and structural elements such as the ECM. The ECM—composed of collagen, fibronectin, elastin, and laminin—serves not only as a structural scaffold but also as a key regulator of tumour cell behaviour, influencing migration, invasion, and therapy resistance [88, 94].

2.2.2 Approaches to Studying Cancer

Understanding the complexity of cancer, particularly its cellular heterogeneity and dynamic microenvironment, relies heavily on advanced analytical technologies.

Traditionally, cancer research has been dominated by genomic and transcriptomic approaches, which have illuminated key mutations, epigenetic alterations, and dysregulated pathways across tumour types [96, 97, 98, 99]. More recently, multi-omics strategies—including epigenomics [100, 101, 102], proteomics [103, 104, 105], and metabolomics [106] — have expanded our understanding of tumour biology by capturing additional layers of molecular regulation.

Single-cell technologies have further revolutionized the field by enabling the dissection of intra-tumoral heterogeneity at cellular resolution, uncovering rare cell populations and lineage trajectories that bulk analyses obscure [107, 108]. These innovations have also extended to single-cell proteomics [109, 110, 111] and metabolomics [112, 113], offering increasingly comprehensive insights into tumour ecosystems. However, despite their power, these methods often lack spatial context—an essential component for understanding how cellular localization and cell–cell interactions within the tumour microenvironment influence disease progression and treatment response.

On the other hand, spatial information is routinely obtained in the clinic through histopathological staining (e.g., H&E), immunohistochemistry (IHC), and radiological imaging [114]. These approaches offer spatial resolution at the tissue or organ level and remain vital for diagnosis and clinical decision-making, but are inherently limited in molecular detail and throughput, lacking the capacity to resolve cellular heterogeneity or dynamic molecular interactions at scale.

To bridge this gap, spatial omics technologies have emerged as powerful tools that integrate high-resolution molecular profiling with spatial localization. These approaches allow the mapping of transcripts, proteins, and metabolites directly within their tissue context—preserving the native architecture of tumours and their microenvironments.

2.3 Spatially Resolved Omics Techniques

Spatial omics technologies have the potential to advance our understanding of tumour ecosystems and improve clinical outcomes. The essence of spatial omics lies in its aptitude for the simultaneous detection of molecular constituents at exact spatial coordinates [115]. However different techniques vary greatly in resolution, scale and molecular complexity [15].

2.3.1 Spatial proteomics

Spatial proteomics encompasses technologies that enable the in-situ profiling of proteins within tissues, preserving their spatial localization. Most spatial proteomics techniques detect proteins using antibodies tagged with fluorophores, metals, or DNA barcodes. These tags are then read using technologies such as fluorescence microscopy, mass spectrometry, or DNA-based imaging to map the spatial distribution of proteins within tissues [25, 115, 116, 117]. A summary of key features across these technologies is provided in Table 2.2.

Immunohistochemistry (IHC) is one of the most established clinical tools for protein detection [118]. It uses enzyme-linked antibodies to generate a chromogenic signal visible under brightfield microscopy.

While widely available and routinely used in diagnostics, traditional IHC is generally limited to detecting one or a few markers per tissue section, making it unsuitable for high-dimensional spatial profiling.

Among the fluorescence-based approaches, immunofluorescence (IF)[119] is widely used but limited in multiplexing. More advanced cyclic immunofluorescence methods—such as tissue-based cyclic immunofluorescence (t-CyCIF)[120] and iterative bleaching extends multiplexity (IBEX)[121] - use iterative staining and imaging cycles to overcome spectral limitations, enabling the detection of 40–60+ markers with spatial resolution down to 200–300 nm.

DNA-barcoded approaches such as co-detection by indexing (CODEX) [8] and immunostaining with signal amplification by exchange reaction (Immuno-SABER)[122] further enhance multiplexing. CODEX utilizes DNA-barcoded antibodies and sequential hybridization of fluorescent probes, achieving high multiplexing (up to 60 proteins) with single-cell resolution (∼500 nm) in a single imaging plane [123]. Immuno-SABER employs orthogonal DNA concatemers for signal amplification achieving 32-plex with same resolution as COSMX [122].

Mass spectrometry-based approaches—notably Imaging Mass Cytometry (IMC) [10] and multiplexed ion beam imaging by time of flight (MIBI-TOF)[124]—use antibodies conjugated to isotopically pure lanthanide metals, which are detected using laser ablation (IMC) or ion beams (MIBI-TOF). These methods avoid fluorescence background and allow simultaneous quantification of 40–50 proteins per tissue section. IMC offers spatial resolution of approximately 1 m, while MIBI achieves higher resolution (∼300 nm), albeit with more complex instrumentation.

Table 2.2: Comparison table of key features of spatial proteomics profiling methods.

Method	**Tag Type**	**Detection**	**Multiplexing Capacity**	**Resolution**	**Resolution Scale**
IHC	Enzyme (chromogen)	Brightfield Microscopy	1–2	∼1–5 µm	Cellular
IF	Fluorophore	Fluorescence Microscopy	4–7	∼200–500 nm	Subcellular
cyCIF / 4i / IBEX	Fluorophore (iterative)	Cyclic Fluorescence Imaging	>60	∼250 nm	Subcellular
CODEX	DNA Barcode	Fluorescence Microscopy	>40–60+	∼250 nm	Subcellular
Immuno-SABER	DNA Barcode (concatemers)	Fluorescence Microscopy	>50+	∼200–300 nm	Subcellular
IMC	Metal Isotopes	Laser Ablation + MS	∼40	∼1 µm	Cellular
MIBI	Metal Isotopes	Ion Beam + MS	∼40	∼300 nm	Subcellular

Together, these spatial proteomics platforms offer complementary advantages in terms of marker throughput, resolution, and sensitivity, enabling comprehensive characterization of the tumour microenvironment, immune cell infiltration, and cellular architecture in situ.

2.3.2 Spatial transcriptomics

Spatial transcriptomics enables the study of gene expression within the tissue architecture, preserving spatial context at cellular or subcellular resolution. These platforms can be broadly divided into two main categories based on detection strategy: imaging-based methods—including *In Situ* Hybridization (ISH) and *In Situ* Sequencing (ISS) —and spatial barcoding methods, which rely on capture-based approaches followed by sequencing. Each approach presents trade-offs in terms of resolution, transcriptome coverage, throughput, and tissue compatibility [115, 125, 126, 127] (Table 2.3).

2.3.2.1 In Situ Hybridization Imaging-Based Approaches

ISH-based techniques use fluorescently labelled probes that hybridize directly to target RNA molecules in fixed tissue sections. These methods are highly accurate and can offer single-molecule and single-cell resolution, but typically have limitations in multiplexing capacity unless cyclic imaging or barcode strategies are used.

Basic single molecule FISH (smFISH) can detect individual transcripts with high spatial precision but is limited in the number of genes detectable due to the finite number of distinguishable fluorophores.

Advanced multiplexed methods, such as Multiplexed error-robust FISH (MERFISH), and seqFISH+, overcome these limits by iterative cycles of hybridization and imaging, enabling detection of hundreds to thousands of genes. MERFISH [128] uses combinatorial labelling and error-correcting barcodes to detect thousands of RNA species in single cells. Sequential fluorescence in situ hybridization (seqFISH)+ [129] leverages sequential rounds of hybridization with an expanded pseudo-color palette, enabling detection of over 10,000 genes at subcellular resolution.

Commercial platforms such as CosMx (Nanostring/Bruker Spatial) [128], MERscope (Vizgen), and Xenium (10x Genomics) [12] implement these strategies to achieve high-resolution imaging of hundreds to thousands of RNA species, often with optional co-detection of proteins.

2.3.2.2 In situ sequencing Imaging-Based Approaches

ISS techniques sequence RNA molecules directly within tissues, preserving both spatial context and nucleotide identity. Unlike ISH, ISS provides sequence information, enabling mutation and splice isoform detection.

STARmap [130] improves detection efficiency by using DNA nanostructures and hydrogel-tissue chemistry, while FISSEQ [131], enables untargeted, whole-transcriptome analysis through in situ reverse transcription and random-hexamer priming. These methods retain spatial localization while offering a more detailed molecular readout than hybridization alone.

Both ISH and ISS require fluorescence microscopy for imaging readouts and are collectively referred to as imaging-based spatial transcriptomics.

2.3.2.3 Spatial Barcoding and Sequencing-Based Methods

Unlike imaging methods, spatial barcoding techniques rely on sequencing-based detection. They use spatially encoded oligonucleotides (barcodes) fixed to a surface (e.g., slide, bead, or grid) to capture RNA from overlying tissue. After RNA capture, reverse transcription and sequencing are performed, and spatial information is reconstructed based on barcode identity.

Prominent examples include Visium (10X Genomics)[132], which captures RNA on slide-mounted barcoded spots (∼55 µm resolution), and it's new version, Visium HD, that offers spatial resolution to ∼2–5 µm. Other examples include Slide-seq [133] and Slide-seqV2 [134] that utilise barcoded beads with known spatial locations (∼10 µm resolution).

These methods offer broader transcriptomic coverage—often approaching the whole transcriptome—and are scalable to larger tissue sections. However, they typically offer lower spatial resolution than imaging-based platforms and may not reveal single-cell or subcellular detail.

2.3.3 Spatial metabolomics

Spatial metabolomics explores the spatial distribution of metabolites directly in tissue sections, providing insight into biochemical activity within the anatomical context [135]. The field is largely driven by Matrix-Assisted Laser Desorption-Ionization Mass Spectrometry Imaging (MALDI-MSI)[136], a label-free, untargeted technique capable of detecting a wide range of small molecules including lipids, neurotransmitters, and drugs.

In MALDI-MSI, tissue sections are coated with a matrix that facilitates ionization when hit by a laser. The resulting ions are analysed by mass spectrometry to reconstruct spatial metabolite maps. MALDI-MSI offers spatial resolution in the range of 10–50 μm, with coverage of hundreds to thousands of metabolites, depending on the tissue and matrix.

2.3.4 Spatial multi-omics

Spatial multi-omics technologies (Table 2.4) enable the simultaneous or integrative profiling of multiple molecular layers—such as RNA, proteins, and metabolites—within their spatial tissue context, offering a more comprehensive understanding of cellular states and interactions.

One of the most established platforms, GeoMx Digital Spatial Profiler (DSP)(NanoString) [137], allows for high-plex profiling of both RNA and proteins within defined regions of interest using oligonucleotide-tagged probes and UV-directed barcode collection, though at limited spatial resolution ($\sim$ 10100 μm). Other advanced methods, such as deterministic barcoding in tissue for spatial omics sequencing (DBiT-seq) [138], achieve co-detection of RNA and proteins through microfluidic-based spatial barcoding on the same section, offering high spatial resolution ($\sim$ 10 μm) and true multimodal readouts. Similarly, Spatial-CITE-seq [139] adapts co-indexing of transcriptomes and epitopes (CITE) to the spatial dimension, enabling the capture of transcriptomes alongside surface protein markers.

Table 2.3: Comparison table of key features of spatial transcriptomics profiling methods.

Method	**Category**	**Transcript Coverage**	**Resolution**	**Resolution Scale**
smFISH	ISH	1–10 genes	~200–300 nm	Subcellular
MERFISH	ISH	1 ,000+ genes	~100–300 nm	Subcellular
seqFISH seqFISH+	ISH-based	>10,000 genes	~200 nm	Subcellular
CosMx	ISH / smFISH	~1,000 genes	~250 nm	Subcellular
MERscope	MERFISH	~500–1,000 genes	~300 nm	Subcellular
Xenium	smFISH	~300–400 genes	~280 nm	Subcellular
STARmap	ISS	~1,000 genes	~2–3μm	Single-cell
FISSEQ	ISS	Whole transcriptome	~300 nm–1μm	Single-cell / Subcellular
Visium	Spatial Barcoding (Slide)	Whole transcriptome	~55μm	Multicellular
Visium HD	Spatial Barcoding (Slide)	Whole transcriptome	~2–5μm	Close to single-cell
Slide-seq	Spatial Barcoding (Beads)	Whole transcriptome	~10μm	Cellular
Slide-seqV2	Spatial Barcoding (Beads)	Whole transcriptome	~10μm	Cellular

Furthermore, MALDI-MSI have been successfully combined on the same slide with both IMC [140] and spatial transcriptomics platforms like 10x Visium [141].

Altogether, spatial omics are accelerating insights into tissue biology and transforming multiple areas of medicine. In particular, they have been extensively applied to study complex microenvironments, such as tumours, where spatial context and cellular heterogeneity are critical and often obscured in bulk or single-modality data [55, 127, 142, 143]. They have also been used to study clonality differences in space [10, 144]. As the applications are vast and rapidly evolving, they are reviewed in detail elsewhere [127, 142, 143].

With this foundation in the underlying technologies and their biological potential, the next section shifts focus to the analytical challenges and computational frameworks required to interpret and integrate spatial omics data.

Table 2.4: Comparison table of key features of spatial multi-omics methods.

Method	RNA Targets	Protein Targets	Metabolite Coverage	Spatial Resolution
GeoMx DSP	∼18,000	100+	–	∼10–100μm (ROI-based)
DBiT-seq	∼6,000-10,000	∼30-100	–	∼10μm
Spatial-CITE-seq	∼5,000-10,000	∼100-200	–	∼20–50μm
MALDI-MSI + IMC	–	∼30-50	100s to 1,000s	∼1–10μm
MALDI-MSI + Visium	18 000	–	100-1000	∼10–50μm (MALDI); 55μm (ST)

2.4 Analytical Frameworks for Spatial Omics Data

The increasing availability of high-resolution spatial omics data—particularly at subcellular resolution—brings new computational challenges, including those related to the spatial dimension, increased data complexity, and multiplexing. This chapter focuses on the analysis of single-molecule spatial transcriptomics and proteomics, where each detected molecule is mapped with precise spatial coordinates. While the field is rapidly evolving, with a growing number of specialized tools and packages emerging at a fast pace, this work will centre on the core analytical pipeline common to most imaging-based platforms. Methods specific to certain technologies—such as deconvolution approaches designed for lower-resolution platforms like 10x Visium—or complex multimodal integration strategies with single-cell data will not be covered here.

2.4.1 Data preprocessing

Both spatial proteomics and spatial transcriptomics require dedicated preprocessing steps to mitigate technical noise and artifacts that may obscure true biological signals. These preprocessing strategies differ significantly, reflecting the distinct nature of the underlying technologies.

2.4.1.1 Spatial proteomics

For spatial proteomics noise sources can vary based on the antibodies, detection channels, and tissue types used, and may manifest as artifacts such as hot pixels, shot noise, background noise and channel crosstalk [145, 146, 147] and may also exhibit low signal-to-noise ratios [148].

Channel crosstalk, where signals from one channel interfere with adjacent channels, can be mitigated through well-designed antibody panels [148] or correction methods like CATALYST [147], which uses pre-acquisition compensation matrices, as well as post-processing techniques [146, 149].

Beyond crosstalk, other sources of noise—such as hot pixels, background signal, and shot noise—require distinct computational strategies. Some studies have employed adjacent approaches, including those implemented in Steinbock [150] and MAUI (MBI analysis interface) [146], while others have developed "homebrew" methods based on traditional image filtering techniques [149, 151, 152]. However, these methods are often insufficient for fully addressing complex noise patterns.

Tools like Ilastik [153] provide supervised pixel classification to distinguish background noise from true signal on a per-marker basis [150, 153, 154]. Although this approach requires extensive manual annotation, it is currently considered state-of-the-art, as it effectively removes background noise and improves signal normalization and batch effect reduction across samples [151]. More recently, IMC-Denoise [155] has been introduced as a two-step pipeline combining traditional algorithms with a self-supervised deep learning model based on Noise2Void [82].

Notably, this thesis contributes to this specific challenge, as discussed in Chapter 3.

2.4.1.2 Spatial transcriptomics

For spatial transcriptomics, preprocessing aims to reduce technical noise introduced during tissue handling, permeabilization, probe hybridization, and sequencing. One of the first steps involves removing low-quality spatial barcodes or transcript reads, which can arise from ambient RNA contamination, incomplete permeabilization, or sequencing artifacts. In some commercial platforms (such as Xenium), quality scores or confidence values are assigned to individual transcripts; Transcripts below a defined threshold can be excluded to avoid false positives. To further refine the dataset, transcripts detected at extremely low frequency across the tissue may be removed.

2.4.2 Cell segmentation

A central challenge in spatial omics is the accurate assignment of molecular measurements—transcripts or proteins—to individual cells. This process, known as cell segmentation, is essential for translating spatial data into single-cell level insights and directly influences downstream biological interpretation.

2.4.2.1 Spatial proteomics

In spatial proteomics, cell segmentation is typically performed using the imaging data generated during acquisition. These multiplexed images often include nuclear stains to identify nuclei and membrane markers to delineate cell boundaries. These signals are combined into RGB or multi-channel images and processed using image analysis pipelines.

Traditional image segmentation approaches, such as thresholding, edge detection, and the watershed algorithm, are commonly employed. These methods are implemented in established tools like CellProfiler [156, 157] and ImageJ/Fiji [158], which allow customization of segmentation pipelines using scripting interfaces. In recent years, DL models have gained prominence due to their improved accuracy, generalizability, and community availability [159]. Notably, Mesmer, part of the DeepCell framework, DL-based model specifically trained on spatial proteomics datasets, leveraging TissueNet, a large annotated dataset

[160]. Another widely adopted method is Cellpose, a generalist DL segmentation tool that predicts vector flows to delineate cell boundaries instead of relying on direct pixel-wise classification. Though originally trained on a broad array of microscopy images, Cellpose also offers specialist models, including those fine-tuned on TissueNet subsets to enhance performance on spatial proteomics images [161, 162].

Segmentation strategies in spatial proteomics range from nucleus-based expansion methods, which approximate cell outlines from DAPI staining, to full-cell segmentation techniques that incorporate both nuclear and membrane markers. A growing number of domain-specific models continue to emerge, each tailored to address the distinct noise profiles, spatial resolution, and multiplexing characteristics of spatial proteomics data [163, 164, 165, 166, 167].

This thesis also contributes to this domain, as discussed in Chapter 4.

2.4.2.2 Spatial transcriptomics

The landscape of spatial transcriptomics segmentation is more varied than in proteomics. In subcellular-resolution platforms such as Xenium and CosMx, segmentation is often performed using accompanying IF images, which typically include nuclear (e.g., DAPI) and membrane markers. These images enable the identification of cell boundaries, which are then used to assign transcripts to individual cells by overlaying segmentation masks with transcript positions.

Alternatively, several methods bypass image-based segmentation altogether. Since transcripts are detected directly as spatial coordinates, these approaches aim to infer cell boundaries based solely on the spatial distribution and density of transcripts, reducing dependency on auxiliary images [168, 169, 170, 171, 172, 173, 174, 175, 176].

A prominent example is Baysor [168], a general probabilistic framework implemented in Julia. Based on Markov Random Field (MRF), Baysor models cells by jointly considering spatial proximity and transcriptional similarity. It can incorporate prior segmentations, assigning confidence scores to them, or operate de novo using only transcript data. Baysor functions in both 2D and 3D, optimizing the likelihood that neighbouring transcripts originate from the same cell. Notably, it has shown strong performance across datasets and has been reported to outperform the default segmentation provided in Xenium workflows [177], highlighting its practical utility and growing adoption.

2.4.3 Cell phenotyping

Cell phenotyping is the process of assigning biological identities to individual cells based on their molecular profiles, such as gene or protein expression, and serves as a critical step in interpreting spatial omics data.

2.4.3.1 Spatial proteomics

Unlike transcriptomic approaches, spatial proteomics typically profiles a pre-selected panel of ~30–50 protein markers, chosen based on the biological question. This reduced and curated feature space simplifies downstream computational analysis but increases reliance on prior biological knowledge for interpreting clusters [150].

Commonly used clustering algorithms include Cytosplore, a GUI which leverages Hierarchical Stochastic Neighbour Embedding (HSNE)[178], Phenograph that builds a shared nearest neighbor graph and uses Louvain algorithm to partition of the graph into communities [178], and FlowSOM that applies SOM [179, 180].

In addition to unsupervised clustering, other strategies for phenotyping include: Manual gating, where marker thresholds are used to define known populations based on expert knowledge; Supervised machine learning models, such as Random Forests, which classify cells using labelled training data and reference mapping, where new datasets are aligned to previously annotated references to infer phenotypes.

2.4.3.2 Spatial transcriptomics

Spatial transcriptomics datasets that approach single-cell resolution often adopt analysis pipelines developed for single-cell RNA (scRNA). This involves several key steps:

The process begins with data preprocessing, where cells of low quality are removed through quality control steps. Following this, normalization is applied—most commonly scaling gene counts to 10,000 per cell and applying a log-transformation—to standardize data across cells. Since these datasets are high-dimensional, dimensionality reduction is used to make the data more manageable. PCA is commonly applied first to identify the most informative gene expression patterns. For visualization and further analysis, nonlinear techniques like t-SNE and UMAP are used to project the data into two or three dimensions while preserving its underlying structure [181].

Next, clustering algorithms group similar cells together. This is usually done by building a KNN graph in the PCA-reduced space. Each cell is connected to its K most similar neighbours, and clustering algorithms such as Louvain [182] and Leiden [183], often with better performance [184] are applied to identify densely connected communities within the graph. These clusters are assumed to represent groups of cells with similar transcriptional profiles, often corresponding to distinct cell types or states.

Once clusters are identified, cell type annotation is performed to assign biological identities to the groups of cells. This can be done either manually or automatically.

Manual annotation relies on known marker genes—genes that are characteristically expressed in specific cell types. In practice, this can be done by checking whether known markers are expressed in each cluster or, conversely, by identifying differentially expressed genes within clusters and matching them to known biological signatures. While manual annotation is transparent and interpretable, it can be subjective, labour-intensive, and limited by the availability and specificity of known markers [181].

Alternatively, automated annotation methods are also available, which leverage reference gene atlases and machine/deep learning models (e.g., CellTypist [185]), or label transfer approaches (e.g., Symphony [186]). These methods provide faster and more scalable annotation but tend to be less interpretable and dependent on the similarity between the reference and the query data [181].

Currently, automated approaches are not yet standard for spatial transcriptomics, particularly in cases where the data is limited by lower gene counts, targeted panels, or lower sequencing depth compared to single-cell RNA. Additionally, many spatial datasets do not align well with existing reference atlases due

to differences in tissue context, resolution, or platform. As a result, manual annotation remains more reliable and commonly used in spatial transcriptomics, particularly when high-resolution spatial context and known marker expression are available.

2.4.4 Cellular neighbourhood analysis

Cellular neighbourhood analysis begins with the construction of a spatial connectivity graph that captures the proximity relationships between cells within the tissue. This spatial graph serves as the backbone for downstream spatial statistics and interaction analyses. Different methods can be used to construct such graphs, including k-nearest neighbours, where each cell is connected to a fixed number of closest neighbours; radius-based expansion, which connects all cells within a given physical distance; and Delaunay triangulation, which uses geometric criteria to define adjacency without imposing arbitrary thresholds. Importantly, the same principles and methodologies for neighbourhood analysis apply across these two modalities, despite differences in the underlying molecular readouts [150, 181].

Once a graph is established, it enables the exploration of spatial interactions between annotated cell types. A common approach involves the computation of neighbourhood enrichment scores, which statistically test whether specific cell types are found adjacent to each other more or less frequently than expected by chance. These analyses rely on permutation-based null models, which shuffle cell-type labels while preserving tissue structure, providing a robust statistical framework to assess enrichment or depletion of interactions [187]. Another related method is the computation of co-occurrence scores, which estimate how likely it is to observe specific cell-type pairs within increasing radii around each cell. These scores reflect the conditional probability of observing a certain cell type given the presence of another nearby, offering an interpretable measure of spatial association across scales [181].

Alternatively, simple interaction matrices can be computed, summarizing the raw counts of neighbouring cell-type pairs [150]. Though not statistical tests, these matrices are useful for exploratory data analysis.

Beyond pairwise interaction analyses, more integrative approaches aim to cluster cells based on the composition of their local neighbourhoods. One strategy involves computing, for each cell, the fraction of surrounding cell types within its local environment (e.g., 20 nearest neighbours). These local composition profiles can then be clustered using unsupervised methods such as k-means or Leiden algorithms, grouping cells into recurring spatial neighbourhoods. This approach was introduced by studies such as Goltsev et al. [8] and Schürch et al. [188], which revealed that specific combinations of neighbouring discrete cell types are often spatially organized in conserved patterns.

A related method involves aggregating gene or protein expression features across the neighbourhood, effectively capturing a summary of the local microenvironment. These aggregated features can then serve as the basis for clustering or downstream modelling, this is approach is implemented for example in covariance environment (COVET) [189], NicheCompass, a graph deep-learning approach to identify and quantitatively characterize niches by learning cell embeddings encoding signalling events as spatial gene program activities [190] and Banksy [191].

Through these computational strategies, spatial neighbourhoods can be identified and interpreted biologically as spatial niches—localized, recurring configurations of cells and their molecular environment that reflect functionally relevant tissue microenvironments. Spatial niches are particularly informative in contexts such as immuno-oncology, where immune-tumour interactions shape the progression or suppression of disease.

2.4.5 Spatial domains

Closely related to cellular neighbourhoods, spatial domains usually refer to broader, tissue-scale regions characterized by coherent gene or protein expression patterns and underlying structural organization [181]. While cellular neighbourhoods capture the immediate microenvironment around a cell—defined by local interactions and spatial proximity—spatial domains extend this concept to larger anatomical or functional areas of tissue. These regions often consist of diverse cell types and multiple neighbourhood configurations, working together to support complex biological functions.

Because of the overlap between microenvironmental patterns and larger spatial structures, some computational approaches can identify both neighbourhoods and domains using similar principles. These methods generally integrate molecular profiles with spatial information, allowing for the detection of both fine-grained and broader-scale tissue organization. A common strategy involves combining a cell's individual gene or protein expression with the aggregated expression from its surrounding neighbours. This allows the model to capture both intrinsic cell identity and the influence of the local microenvironment, which is particularly useful in tissues where cells of the same type are dispersed rather than spatially clustered.

A wide range of computational strategies has been developed for domain segmentation. Some early methods relied on spatial smoothing techniques, such as Markov Random Fields, which encourage physically proximate cells to be assigned the same label [192, 193]. This assumes that a cell's transcriptome resembles the average of its domain, although this may not always hold true in tissues where multiple cell types are intermixed. Other approaches employ deep learning, particularly graph-based neural networks that can incorporate either histological information [194, 195] or spatial graphs derived purely from molecular and positional data [196, 197, 198]. These methods offer flexibility and can capture complex spatial dependencies, but their performance can vary with dataset size and structure.

More recently, methods introducing hierarchical and multiscale representations of spatial structure have emerged. For instance, NeST [199] identifies nested co-expression hotspots—spatially contiguous regions that co-express subsets of genes—by simultaneously searching across gene and spatial dimensions. This approach captures biological structure at multiple scales and can reveal overlapping or nested domains, without requiring prior assumptions about gene sets or spatial resolution. Similarly, the concept of tissue schematics [200] has been proposed as a way to abstract spatial organization into higher-order motifs, by identifying modular assemblies of cellular neighbourhoods. These schematics can be used to represent tissue architecture in both healthy and diseased states, offering insights into how local interactions scale up to tissue-level functionality.

2.4.6 Spatially Variable genes

Spatially variable genes (SVG)s are genes whose expression displays significant spatial patterning across a tissue. Identifying SVGs reveals insights into tissue architecture, cell–cell communication, and local microenvironmental influences, extending beyond what is captured by cell types or spatial domains alone. These spatial patterns may arise from gradients in signalling molecules, regional functional differences, or heterogeneous cell compositions [181]. Notably, SVG detection does not rely on cell segmentation, making it applicable across different spatial omics technologies.

Several computational methods have been developed to detect SVGs by decomposing gene expression variability into spatial and non-spatial components. A widely used statistic is Moran's I, which quantifies spatial autocorrelation by measuring how similar gene expression is between neighbouring spots [201]. Model-based approaches like SpatialDE [202], trendseek [203], SPARK [204], and SPARK-X [205] use Gaussian processes or spatial statistical models to test for spatially structured expression while accounting for noise and overdispersion. While these tools test each gene independently and return p-values, they often overlook spatial domain context, which can limit biological interpretability. Other strategies take different modelling perspectives: Sepal [206] applies a Gaussian diffusion process, scGCO [207] uses graph cuts to detect spatial expression boundaries, and SpaGCN [195], defines both domains and SVGs integrating histology and spatial proximity through graph convolutional networks.

Importantly, the identification of SVGs is modality-agnostic and applicable to both spatial transcriptomics and spatial proteomics. Regardless of whether gene expression or protein abundance is measured, spatially variable features represent key molecular signatures of tissue structure and function, and form an essential layer of spatial omics analysis.

2.4.7 Functional analysis

Understanding how spatial context shapes cellular function is an increasingly important focus in spatial omics. Beyond identifying cell types, neighbourhoods, or spatially variable genes, functional analysis aims to reveal how cells operate and interact in situ. This includes investigating intracellular signalling pathways, transcription factor (TF) activity, and particularly ccell–cell communication (CCC) events—often inferred from curated interaction networks derived from transcriptomics data [208, 209].

A broad landscape of computational tools has emerged for CCC analysis [209, 210, 211, 212, 213, 214, 215, 216, 217, 218], reflecting the growing complexity and richness of spatial omics data. Broadly, these methods fall into two sections: identifying pairs of genes that interact, such that expression of the gene in one cell influences that of the other gene in others; and identifying pairs of cells in which that gene pair interacts [219].

Most methods for CCC leverage ligand–receptor (L-R) interactions, relying heavily on prior biological knowledge, often curated into databases such as CellPhoneDB [211], OmniPath [220], and CellChat [212]. These databases catalogue known signalling pathways and interaction pairs from the literature. Using such resources, CCC inference can be extended to the spatial context by identifying L–R pairs co-expressed in nearby cells [209] for example using spatial co-expression [209, 215], or incorporating

different approaches such as optimal transport frameworks [214] or Bayesian multiple instance learning (MIL)[218] to more precisely link cells by both location and expression.

Complementing these prior knowledge-driven approaches are data-driven methods that model spatial gene expression to discover novel interactions. These models, like NCEM[210], GCNG[217] or SVCA[221] and Misty[222] (while not specific for CCC), aim to capture interactions that explain spatial expression variance across multiple genes. They enable inference of new communication patterns not captured in curated L–R databases, though their accuracy depends on model design and training data.

Furthermore, some CCC frameworks estimate global co-localization across entire tissue sections [210, 213, 222], and others focus on local cell-cell proximity [214, 215].

In parallel, there is increasing interest in inferring intracellular functional activity—such as pathway activation, TF activity, or gene set enrichment—within individual cells or regions. Tools like decoupIR[223] apply prior knowledge from pathway perturbation signatures[224] or regulatory networks[225] to estimate these functional states at single-cell or spot level. As with CCC, these functional scores can then be spatially mapped. These approaches are currently more mature for spatial transcriptomics, where broader gene coverage enables more reliable functional inference. In contrast, spatial proteomics remains limited by smaller marker panels and less comprehensive prior knowledge, constraining the resolution of functional state estimation.

While spatial omics provides a key advantage over single-cell RNA through direct measurement of cell proximity—eliminating the need for probabilistic neighbourhood modelling and enabling exclusion of implausible, long-range interactions—this often comes with trade-offs. Platforms with true cellular resolution (e.g., CosMx, Xenium) may capture fewer genes, complicating pathway-level analyses and downstream functional interpretation. Moreover, the heavy reliance on curated databases across most CCC methods introduces variability and potential inconsistencies, as predictions can differ significantly depending on the resource used.

Ultimately, understanding how a cell's functional state is shaped by its environment—and how it, in turn, influences surrounding cells—is a central and still largely untapped frontier in spatial biology. As methods advance, this bidirectional view of cellular function in context promises to transform our understanding of tissue organization, disease progression, and therapeutic targeting.

2.4.8 How to distinguish between sample conditions

While the tasks described in the previous sections—such as preprocessing, cell segmentation, and feature extraction—lay the foundation for spatial omics analysis, deriving meaningful biological insights often requires comparing groups, such as different disease states, treatment responses, or tissue types.

2.4.8.1 Summary statistics

One of the earliest approaches involves aggregating spatial features across samples and comparing them using statistical summaries. For example, scores for cell–cell communication, number of neighbours, or other spatial statistics can be averaged per sample and used to detect differences between conditions

[209, 222, 226]. However, this strategy is often too simplistic, as it can blur important heterogeneity by averaging out spatially localized effects.

2.4.8.2 Matrix factorization methods

Matrix factorization methods provide a more nuanced alternative. These approaches reduce high-dimensional data into a smaller set of latent factors that capture major patterns of variation. Classical approaches like PCA achieve this through linear decompositions, but newer tools like Multi-Omics Factor Analysis (MOFA)[227, 228, 229], MEFISTO[230], and DIALOGUE[231] extend the concept to handle multiple data types, structured variation, or intercellular coordination. These methods can incorporate spatial information embedded in the matrix.

MOFA and its variants allow joint analysis of multi-omic datasets, identifying latent factors that capture both shared and group-specific variation. Other tools like DIALOGUE[231] and scITD[232] focus on identifying coordinated gene programs across multiple cell types, enabling the characterization of multicellular processes in an unsupervised fashion.

NMF provides a more interpretable, parts-based representation, especially useful for identifying additive biological signals like distinct cell states or spatial domains. Spatially-aware versions of NMF, such as Nonnegative Spatial Factorization (NSF), extend this idea by directly incorporating spatial coordinates, better capturing localized gene expression patterns [233].

These matrix-based methods not only enhance the ability to uncover biologically relevant spatial and intercellular patterns, but also enable unsupervised identification of group-level differences—whether driven by disease, environment, or treatment—without requiring explicit labels.

2.4.8.3 Machine Learning and Deep Learning approaches

Beyond matrix factorization, supervised ML and DL methods are increasingly being explored to classify samples or predict disease conditions using spatial and single-cell omics data. These models are particularly well-suited to capturing nonlinear relationships and subtle patterns in high-dimensional datasets. However, their application in this domain remains relatively limited—mainly due to the scarcity of large, well-annotated datasets needed to train robust deep models. While DL approaches typically require thousands of samples to achieve generalizability, spatial omics datasets have only recently begun to reach that scale.

It's worth noting that DL has been widely adopted in other stages of the spatial omics pipeline—such as image preprocessing, segmentation, domain identification, phenotyping, and modelling of cell–cell communication—but its use for direct classification or outcome prediction is still sparse.

Among early efforts in this direction, NaroNet [234] introduced a patch contrastive learning strategy combined with graph representations to predict patient outcomes from IMC data. In the same cohort, Fu et al.[235] proposed a deep multimodal graph-based network that integrates IMC data with clinical variables to predict cancer survival. In another example, Risom et al.[236] applied a random forest classifier to a MIBI dataset of ductal carcinoma in situ (DCIS), using over 400 spatial features—including

tissue compartment enrichment and TME morphometrics—to distinguish between progressors and non-progressors.

A more recent and notable advance is S3-CIMA [237], a weakly supervised, single-layer convolutional neural network that learns disease-specific spatial compositions of the tumour microenvironment. By modelling local cell-type organizations from high-dimensional proteomic imaging data (IMC and CODEX), it enables the discovery of outcome-associated microenvironments in colorectal cancer. Similarly, graph deep learning has been used to predict prognosis in gastric cancer, where Cell-Graphs built from multiplexed immunohistochemistry (mIHC) data enable prognosis prediction from spatial arrangements of cell types [238].

Important to note is that DL methods are already well-established in computational pathology and digital histopathology, where large annotated datasets and well-defined visual features have allowed CNNs to thrive in image classification, segmentation, and prognosis prediction tasks [239, 240].

2.4.9 Frameworks and tools for spatial omics analysis

A growing ecosystem of software tools supports the analysis of single-cell and spatial omics data. Seurat (R) [241] and Scanpy (Python) [242] are widely used for single-cell analysis. Scanpy is built around the efficient AnnData structure [243], while Seurat uses its own SeuratObject. Dedicated spatial omics tools such as Giotto (R) [244] and Squidpy (Python) [226] offer integrated workflows for spatial statistics, neighbourhood analysis, and visualization with also new data frameworks like SpatialData (Python) [245] emerging to better support spatial modalities. Napari [246], a general-purpose image viewer, complements these tools with interactive, high-dimensional image visualization—useful for working with spatial coordinates and tissue images.

Python and R remain the dominant programming languages in this space. Python is increasingly favoured for spatial omics due to its speed, memory efficiency, and compatibility with deep learning and image processing (e.g., PyTorch), while R remains popular for exploratory analysis and visualization[127]. Meanwhile, Julia is gaining attention for its performance, with tools like Baysor [168] highlighting its potential in spatial omics analysis [247].

Given the rapid growth of tools—and the risk of incompatibilities in data formats, APIs, and user interfaces—standardization efforts like scverse [248] have emerged. These initiatives promote well-maintained, interoperable core functionality, supporting a more cohesive and collaborative spatial omics software ecosystem.

2.4.10 Closing remarks

This review has focused on spatial omics methods that operate on the basis of cell-defined units. However, it is important to note that many of these approaches can also be applied at the transcript level, bypassing the need for explicit cell segmentation—this is particularly true for tasks such as domain identification. While cell-based analysis remains the prevailing standard in spatial omics, alternative strategies that complement or replace cell segmentation are gaining ground [198, 249]. This shift is partly driven

by the technical difficulty of accurately segmenting individual cells, especially in complex tissues where cells may be overlapping, densely packed, or poorly defined. Acknowledging these challenges, this thesis introduces novel approaches for cell-free analysis of spatial transcriptomics data, as elaborated in Chapter 5.

Furthermore, this review has specifically addressed computational methods applicable to single-cell resolution spatial data, and does not cover broader-scale approaches such as spatial deconvolution or data imputation, which are more relevant to spot-based or lower-resolution platforms.

Even within the focused scope of single-cell approaches, many valuable tools and developments could not be fully covered, reflecting the sheer volume and velocity of innovation in the field. Figure 2.6 provides a visual overview of the spatial omics methods discussed in this review, highlighting the diversity and complexity of the computational landscape. Nodes are organized by their functional role in the analysis pipeline and scaled by citation count, illustrating both methodological clustering and community impact. Furthermore, an interactive visualization is available at `https://marta-seq.github.io/SOME/`. The increasing availability of spatially resolved data across multiple modalities has fuelled an avalanche of computational methods, making it increasingly difficult for researchers to keep pace and make informed decisions about which tools best suit their needs. In this evolving landscape, there is a growing need for "living reviews"[127], curated repositories, and community-driven benchmarking [240] efforts that can adapt to the field's rapid progress.

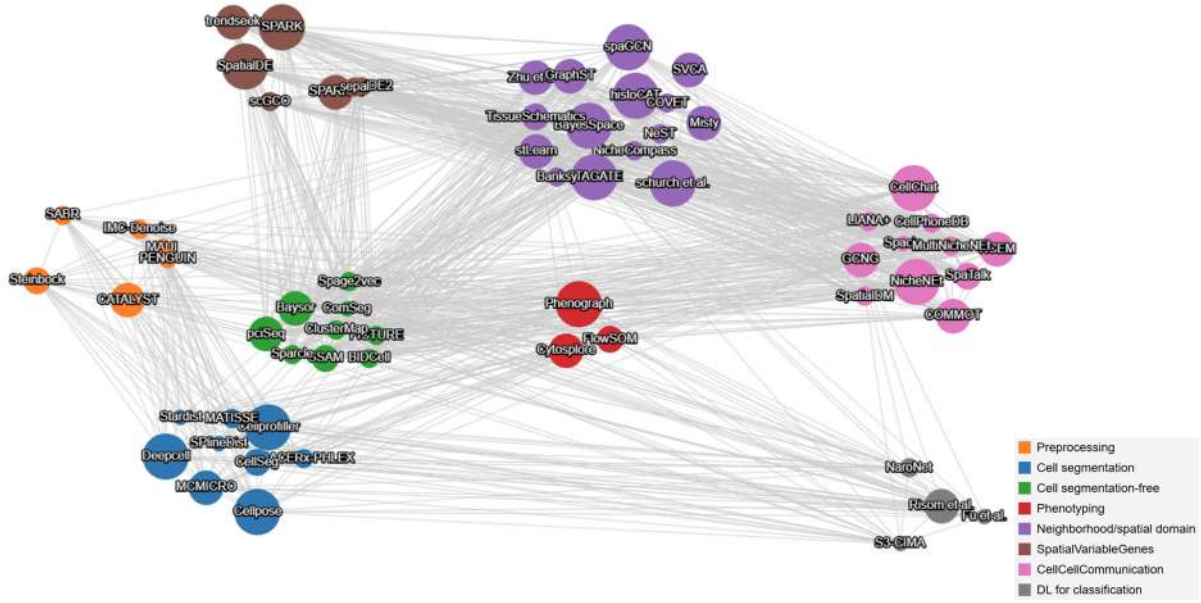


Figure 2.6: Graph overview of spatial omics methods analysed in this review. Each node represents a method included in this review, with the colour indicating its role in the spatial omics analysis pipeline (e.g., orange – Preprocessing, blue – Cell Segmentation, etc.) and the size of the bubble proportional to its citation count, reflecting community adoption. Nodes are grouped spatially by conceptual similarity and method type. Only full spatial omics methods are shown, excluding foundational algorithms or reused modules from unrelated pipelines. This layout reveals the landscape and complexity of spatial omics development. The interactive version of this graph is available at: `https://marta-seq.github.io/SOME/`. This approach can be extended to incorporate newly published methods or customized for other use cases.

3

A rapid and efficient image preprocessing tool for multiplexed spatial proteomics

Multiplex spatial proteomic methodologies can provide a unique perspective on the molecular and cellular composition of complex biological systems. Several challenges are associated to the analysis of imaging data, specifically in regard to the normalization of signal-to-noise ratios across images and subtracting background noise. However, there is a lack of user-friendly solutions for denoising multiplex imaging data that can be applied to large datasets. We have developed PENGUIN –Percentile Normalization GUI Image deNoising: a straightforward image preprocessing tool for multiplexed spatial proteomics data. Compared to existing approaches, PENGUIN distinguishes itself by eliminating the need for manual annotation or machine learning models. It effectively preserves signal intensity differences while reducing noise, improving downstream tasks such as cell segmentation and phenotyping. PENGUIN's simplicity, speed, and intuitive interface, available as both a script and a Jupyter notebook, make it easy to adjust image processing parameters, providing a user-friendly experience. We further demonstrate the effectiveness of PENGUIN by comparing it to conventional image processing techniques and solutions tailored for multiplex imaging data.

Keywords: Multiplex imaging, Immunophenotyping, Denoising, Normalization, Background subtraction, Spatial omics

---

This chapter was co-authored by Dr. Marieke Ijsselsteijn and was published in the following peer-reviewed article: *Sequeira, A. M., Ijsselsteijn, M. E., Rocha, M., & de Miranda, N. F. (2024). PENGUIN: a rapid and efficient image preprocessing tool for multiplexed spatial proteomics. Computational and Structural Biotechnology Journal, 23, 3920-3928*.

3.1 Introduction

In recent years, multiplexed imaging technologies have advanced significantly, enabling spatially resolved profiling of biological samples [5, 250]. IMC [10] and MIBI-TOF [124] make use of metal-conjugated antibodies for the detection of proteins by means of mass spectrometry. This process involves quantifying

isotopic reporter masses released from tissue after the ablation of small regions using a laser beam or ion beams. Other antibody-based multiplex approaches, such as CODEX [8], rely on fluorescent-labeled antibodies.

Despite the growing use of multidimensional proteomics spatial technologies, the resulting data presents challenges that conventional image analysis methods often struggle to address. A common challenge is the presence of noise in data, which must be removed before analysis [145]. Noise sources can vary based on the antibodies, detection channels, and tissue types used, and may manifest as artifacts such as hot pixels or background noise [145, 146, 147]. Additionally, some immunodetections may exhibit weak signals and low signal-to-noise ratios [148]. Together, these noise factors degrade image quality and complicate downstream analyses of multiplex imaging data. Consequently, robust and reliable denoising methods have become increasingly important as imaging technologies are more widely applied [145]. Specifically, IMC noise includes hot pixels, shot noise and channel crosstalk [155]. Channel crosstalk, where signals from one channel interfere with adjacent channels, can lead to signal misidentification [145, 148]. To address this, methods like CATALYST [147] use pre-acquisition signal compensation matrices, while post-acquisition solutions are also available [146, 149]. However, well-designed antibody panels can often reduce the need for correction by minimizing overlap between channels [148, 155]. Hot pixels are characterized by individual pixels with significantly higher signal intensities than their surroundings. Additionally, small clusters of consecutive hot pixels, which do not represent biological structures, may form due to nonspecific antibody binding, antibody aggregates, or contamination by dust particles [146, 148, 155]. Shot noise, resulting from ion counting imaging processes, is pixel-independent but signal-dependent manifesting as random signal variations that typically follow a Poisson distribution [155]. This means noise is more pronounced at lower signal levels.

Hot pixels are commonly addressed using traditional image filtering methods and thresholds. Researchers have also developed numerous “homebrew” computational strategies, customized for specific projects [149, 151, 152]. One notable approach, implemented in Steinbock, a Python toolkit for processing multidimensional images, mitigates hot pixels by comparing each pixel’s value to those of its eight surrounding neighbors. If the difference between the pixel and any of its neighbors exceeds a predetermined threshold, the pixel’s value is adjusted to match the highest neighboring pixel value [150].

In addition to conventional denoising techniques, tools like Ilastik provide supervised pixel classification, allowing for the distinction between background noise and true signal pixels for each marker [150, 153, 154]. In this approach, an experienced user trains the Ilastik Random Forest pixel classifier by manually labeling pixels as signal or background. Once trained, the model is applied across all images, generating a binary expression map where non-noise pixels are assigned a value of 1.This method has proven effective not only in removing background noise but also in standardizing and normalizing signals across samples, thereby reducing batch effects [154]. However, the requirement for manual annotation across images can be labor-intensive. This manual curation process is particularly challenging when dealing with large datasets, making this approach less practical for large datasets. Recently, Lu et al. [155] introduced IMC-Denoise, a two-step pipeline that combines a Differential Intensity Map-based Restoration (DIMR) algorithm to remove hot pixels and a self-supervised DL algorithm called DeepSNiF for filtering

shot noise. DeepSNiF, inspired by the widely used Noise2Void algorithm, does not rely on pretrained data. Instead, it trains a neural network to eliminate noise by leveraging information from surrounding pixels within the same image, eliminating the need for a separate, clean reference image. The process begins by identifying hot pixels using Anscombe transformation differential maps. Then, DeepSNiF filters shot noise by randomly masking pixels using a stratified sampling approach. However, DL approaches like this are often time-consuming, resource-intensive and require careful parameter tuning.

Despite the availability of various IMC denoising methods, there is still no consensus on the optimal approach, and the lack of a standardized strategy hinders comparisons between datasets. Current methods either require the manual identification of noisy pixels across all markers – a time-consuming and inconsistent process – or fail to adequately address all types of noise and nonspecific antibody signals.

In this work, we introduce Percentile Normalization GUI Image denoising (PENGUIN), an enhanced pipeline for denoising multiplexed proteomics data. PENGUIN employs scaling, thresholding, and percentile-based filters to address various noise sources in multiplex images, including hot pixels and shot noise. The method is fast, scalable, and reproducible, without requiring manual pixel annotation or extensive hardware resources. We demonstrate its effectiveness using an existing IMC dataset, benchmarking it against current preprocessing methods and those specifically designed for multiplex imaging data analysis. Additionally, we show its versatility by applying PENGUIN to multiplex IF images, illustrating its applicability across different imaging modalities. The tool is available on `https://github.com/deMirandaLab/PENGUIN`.

3.2 Results

To mitigate noise in IMC images, the following considerations were identified:

1. Hot pixels, characterized by random pixels exhibiting significantly higher signal intensities than their surroundings, can be efficiently addressed by filtering out sparse signals, specifically pixels that lack neighboring signals (Supplementary Figure S3.1).

2. Shot noise, arising from inherent variability in photon detection, can result in low-intensity signals in regions where no specific signal is expected. This noise is particularly prominent in areas with weak or no biological signal and can produce non-zero pixel values due to random detection fluctuations (Supplementary Figure S3.1).

3. Noise typically occurs independently across different channels within the same image, meaning one channel can be affected without others being impacted. However, within a specific cohort, noise generally follows consistent patterns across all images from the same channel.

Based on these considerations, PENGUIN was developed to integrate scaling, thresholding, and percentile-based filters for denoising IMC data (Figure 3.1A). PENGUIN follows a multistep approach: 1) Images are saturated at the 99th percentile to reduce the impact of extremely bright pixels; 2) Signal

intensities across each channel and image are normalized to a scale from 0 to 1; 3) An adaptable threshold is applied to remove low-intensity signals; and 4) Hot pixels are identified using adaptable percentile filters and removed. A key feature of this strategy is that the percentile filter is used solely for hot pixel identification, ensuring their removal without affecting image clarity or edge definition. As a result, post-processed images maintain sharpness and edge integrity. Additionally, the normalization of values from 0 to 1 simplifies their direct use in subsequent analyses, such as cell segmentation.

To streamline the visualization and deployment of the preprocessing pipeline, PENGUIN is provided as a user-friendly Jupyter notebook (Figure 3.1B). This tool allows users to directly observe how thresholds and percentiles impact different markers, aiding in the selection of the most suitable thresholds for each marker and image. The code for the preprocessing pipeline is freely available, allowing users to also apply the functions directly via scripting.

Key adjustable parameters in this pipeline include the Threshold (T) value and the Percentile (P) value, both of which are critical to its performance (Figure 3.2). A higher T value removes more signal during the initial phase of the pipeline. If a channel primarily contains background signal, with only high-intensity values representing true signals, T should be set high. Conversely, in channels with minimal background noise, T can be lower or even omitted. The P parameter plays a crucial role, with lower values (e.g., 25) being stricter, removing nearly all sparse signals, while higher values (e.g., 75) are more lenient, allowing more sparse signals to remain. Tailoring these parameters for each channel is essential due to the significant variability in marker behavior. For example, β-catenin and FOXP3 showed optimal noise reduction with a T of 0.1 and a P of 50, while CD20 required a higher threshold of T 0.3 to effectively eliminate noise. In contrast, clear images for vimentin and CD45 were achieved with P set at 25, without needing a threshold setting (Figure 3.2).

3.2.1 Comparison of available methods for IMC denoising

PENGUIN was evaluated against both traditional image denoising techniques and IMC-specific methods using a public dataset of 39 cellular markers analyzed by IMC across 61 samples [251].

To enhance the performance of standard image denoising methods, we applied saturation at the 99th percentile to remove bright outliers and normalized each channel's signal intensity between 0 and 1 [154]. We then tested a variety of classical filtering methods, including Gaussian, mean, percentile, non-local means, bilateral, total variation, wavelet, anisotropic diffusion, and BM3D filters (see Supplementary Figures S3.2& S3.3)). As expected, linear filters like Gaussian and mean filters tended to blur the images, causing a loss of detail and definition in tissue boundaries. While these filters did not directly eliminate hot pixels, they made the pixels less noticeable by blurring them. The other classical filters either failed to effectively reduce noise (e.g., anisotropic and bilateral filters) or distorted the images, as seen with the non-local mean filter, making them unsuitable for this type of data.

We also evaluated DL methods designed for denoising images without requiring paired ground truth data, specifically Noise2Void [82] and denoising autoencoders [252], trained for 20 or 50 epochs. These methods did not effectively denoise the IMC images and led to image distortion, suggesting they may not



Figure 3.1: Overview of the PENGUIN tool. (A) Overview of the PENGUIN pipeline: 1 - the original image contains noise and extreme outliers, which obscure the rest of the signal. 2 - outliers are managed by capping values above the 99th percentile, enhancing visualization. Each channel and image are then normalized to a scale of 0 to 1. 3 - pixels with values below a selected threshold are removed. 4 - percentile filtering is applied to detect and remove noise pixels. For each central pixel, a 3x3 window is considered and the pixel values are sorted in ascending order. If the pixel is at the 0th percentile, it is classified as noise and set to 0; if it is above 0, it is considered part of the signal. For example, a 50th percentile (P50) requires that 5 out of 9 pixels be positive, the 75th percentile requires 2 out of 9, and the 25th percentile (P25) requires 7 out of 9. Depending on the characteristics of the channel, low percentile values may result in the removal of true signals, while high percentile values may retain noise. (B) Sneak peek of the PENGUIN Jupyter notebook.

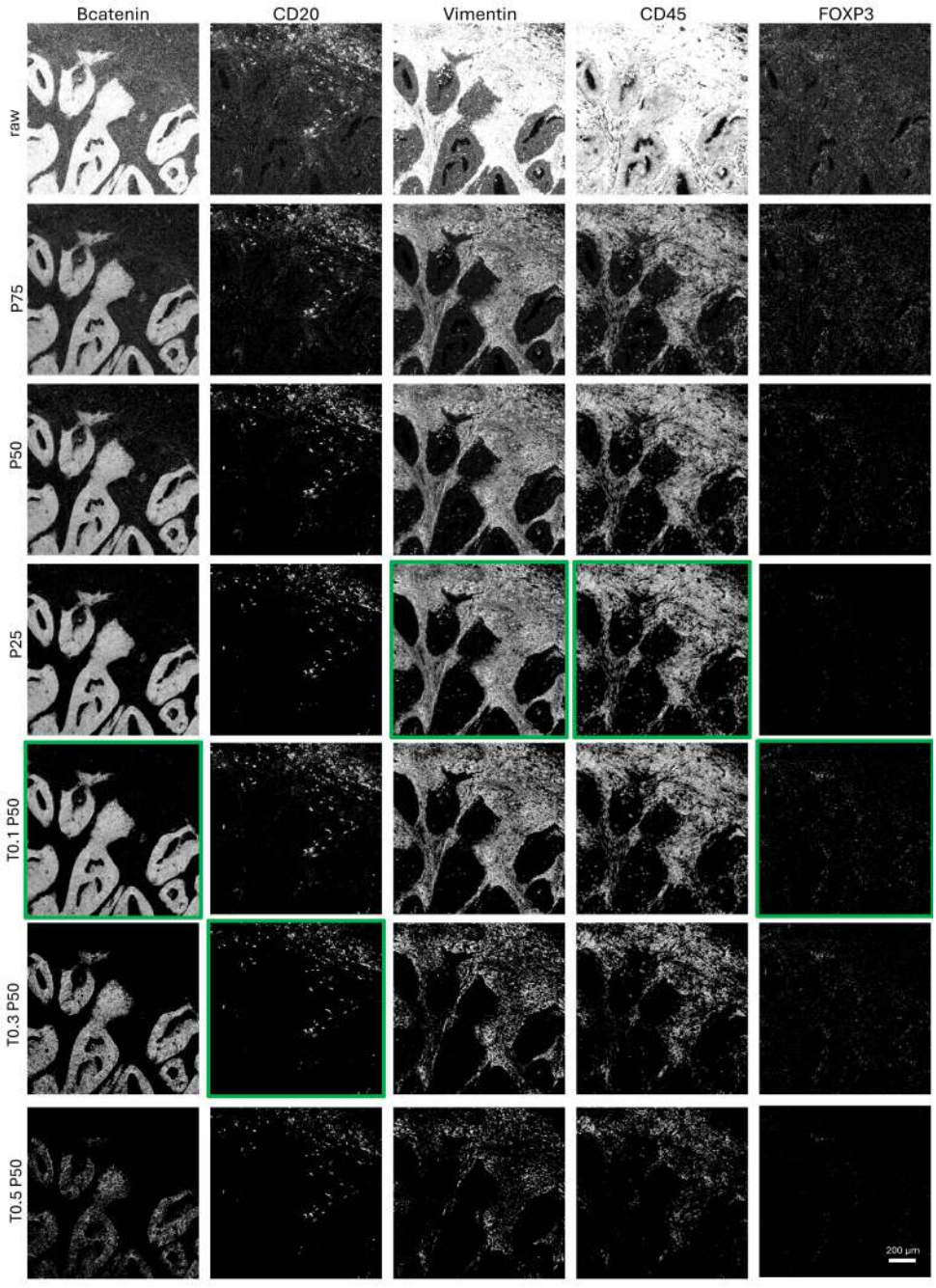


Figure 3.2: Application of various percentiles (P90, P75, P50, P25, P10) and thresholds (T0.1, T0.3, T0.5 with P50) with PENGUIN were used to assess immunodetection of β-catenin, CD20, vimentin, CD45, and FOXP3. The optimal parameters for each marker are highlighted in green.

be well-suited for this type of data. However, it is important to note that DL models tend to perform better with larger datasets, and as more IMC image datasets become available, the performance of these models may improve. Additionally, our experiments only employed a basic convolutional denoising autoencoder, leaving room for significant optimization through different combinations of layers, epochs, and parameters. However, DL methods are time-intensive, requiring considerate effort for development, tuning, and training, and often necessitate specialized hardware, which that may not be readily accessible to all users.

Next, we applied two IMC-specific methodologies. First, we tested a hot pixel filter [150] that uses a 3x3 kernel to determine if the difference between a pixel and its maximum neighboring value exceeds a predefined threshold; if so, the pixel is classified as noise and replaced with the maximum neighbor's value. This method was applied to both normalized images, using thresholds ranging from 0.05 to 0.2, and raw images with a default threshold of 50. While this approach successfully removed some hot pixels, the processed images still contained noise, and in some cases, real signals were lost (see Supplementary Figures S3.2& S3.3). Second, we applied IMC Denoise [155], using both the DIMR algorithm alone and in combination with the DeepSNIF DL algorithm, following the authors' guidelines and default settings. The DIMR algorithm alone proved insufficient for fully removing hot pixels. When combined with DeepSNIF, similar issues arose as with Noise2Void, including image distortions. As with other DL methods, the possible configurations for IMC Denoise are virtually limitless; however, fine-tuning the optimal parameters is a labor-intensive process. Additionally, separate models must be trained for each channel, further complicating the implementation of this approach.

To evaluate IMC denoising performance with PENGUIN, we employed our lab's semi-automated background removal (SABR) pipeline [154], which utilizes Ilastik (Figure 3.3 & Supplementary Figure S3.4). Given the lack of a true ground truth, we used the manually annotated SABR-denoised dataset as a reference standard. Using this dataset, we calculated the Peak Signal-to-Noise Ratio (PSNR), where higher values denote reduced error, and the Structural Similarity Index (SSIM), which ranges from -1 to 1, with a score of 1 indicating perfect similarity (Supplementary Figure S3.5). These metrics enabled a comparative analysis of the denoising performance of raw (rescaled) and PENGUIN-denoised images relative to the SABR dataset, our designated reference. Results demonstrated a marked improvement in image quality over raw data, with PENGUIN's denoising closely aligning with the SABR-annotated dataset. Furthermore, the signal distributions across raw, 99th percentile normalized values, SABR, and PENGUIN-denoised images are presented in Supplementary Figures S3.6,S3.7 and S3.8. It is worth noting that SABR denoising may occasionally result in signal loss for certain markers, as observed with β-catenin immunodetection (Figure 3.3 and Supplementary Figure S3.4). By applying PENGUIN to the same image set, we observed effective noise reduction akin to that achieved by the SABR pipeline, with the added advantage of retaining more signal than SABR.

In summary, PENGUIN effectively eliminates noise from IMC images. While it requires manual adjustment of thresholds and percentiles for each channel, it does not necessitate developing specific classifiers to differentiate between signal and noise for each marker. Additionally, the settings are marker-dependent and often translatable between images, making it easy to apply to new images obtained with the same antibody panel. Unlike the Ilastik-based method, which produces binary images indicating only the presence

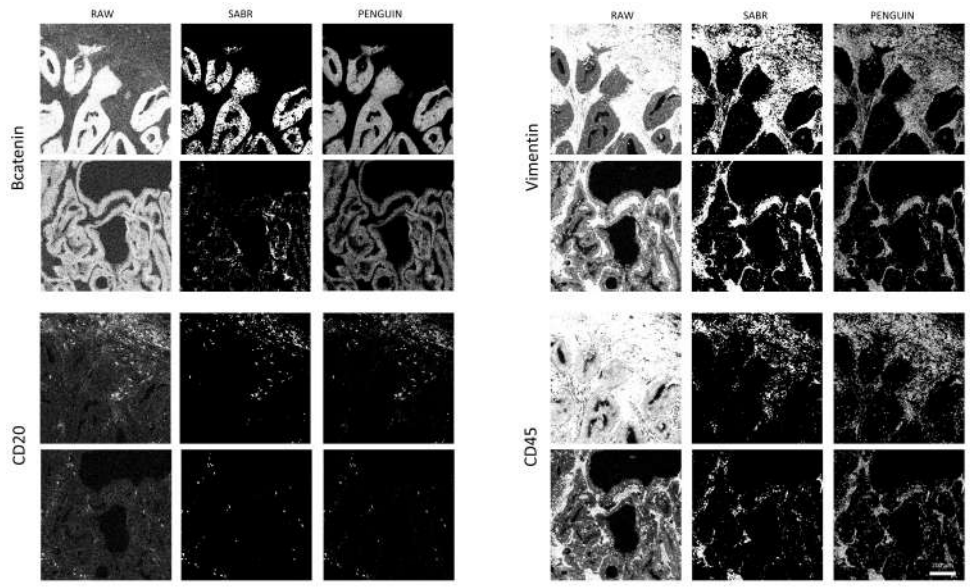


Figure 3.3: β-catenin, Vimentin, CD20, and CD45 immunodetection in two regions of interest, showing raw images, processed by the Ilastik-based SABR approach, and PENGUIN.

or absence of signals, PENGUIN generates images with normalized values ranging from 0 to 1, capturing a range of signal intensities. This allows for more nuanced analysis, as PENGUIN not only retains more signal but also preserves critical details of cellular structures.

Setting the thresholds and percentiles is straightforward, further simplified by the provided notebook. Processing all 61 images per channel took under 4 seconds on an Intel(R) Core(TM) i7-9700 CPU – a major improvement over Ilastik, which requires 5 to 15 minutes per marker. PENGUIN processing itself takes about 4 seconds per channel, with an additional 5 minutes for expert parameter selection. In contrast, SABR requires approximately 10 minutes of algorithm training plus 5 minutes per channel for processing. This highlights PENGUIN's efficiency over both Ilastik and SABR. Additionally, without the need for model training, PENGUIN's results are fully reproducible and readily accessible, offering a significant improvement over existing methods.

3.2.2 PENGUIN enhances downstream analysis

After confirming the effectiveness of PENGUIN for image processing, we evaluated its impact on downstream analysis. This included comparing raw, unprocessed images with those corrected using the Ilastik-based SABR method and the PENGUIN pipeline, focusing on cell segmentation and cell phenotyping/clustering.

A key advantage of PENGUIN over previous approaches is that its output can be directly used in cell segmentation pipelines, eliminating the need to train an additional Ilastik model to distinguish between membrane, nucleus, and background markers, thereby saving significant time. Furthermore, as highlighted earlier and shown in Figure 4, PENGUIN's normalization process preserves more signal and retains important cellular features.

To compare these methods, we combined the normalized images with their corresponding cell segmentation masks to calculate relative marker expression per cell. Cells were then clustered using t-SNE to identify cell subsets (Figure 3.4A). Consistent with findings by Ijsselsteijn et al. [154], normalization and background removal are crucial for accurately identifying signals and cells, as well as normalizing inter-sample variation in IMC data for automated downstream analysis. Like current state-of-the-art techniques, PENGUIN adjusts for interpatient variability, resulting in well-defined cells and clusters. Next, both the SABR and PENGUIN-denoised datasets were phenotyped (Figure 3.4B). PENGUIN-normalized cellular phenotypes were easily identifiable and displayed clear marker patterns, comparable to those in SABR-normalized data. Additionally, mapping these phenotypes back onto the images demonstrated that cellular distribution was consistent with the original data (Figure 3.4C).

3.2.3 PENGUIN can be applied to other image modalities

Various imaging modalities in multiplex spatial proteomics, such as MIBI, CODEX, and multiplex IF, produce different types of images. Encouraged by the successful outcomes of the PENGUIN pipeline, and given that these modalities share similar limitations, we extended the application of PENGUIN to a set of in-house generated IF images (Figure 3.5). The results demonstrated high-quality denoised images, achieved with significantly reduced time, resources, and operator-related variability.

3.3 Discussion

Spatial proteomics data, such as that generated by IMC, offers unique insights into the spatial organization of tissues, with broad applications. However, managing signal-to-noise ratios, especially with large antibody panels, presents significant challenges. Therefore, effective computational preprocessing methods are crucial to ensuring the reliability of subsequent analyses [145]. While various custom algorithms have been successful in denoising IMC data, the growing volume of data calls for a standardized and efficient preprocessing pipeline. In this context, we introduce PENGUIN, a denoising pipeline that effectively processes IMC images using scaling, thresholding, and percentile filters, yielding excellent results. We compared PENGUIN to traditional filtering methods, DL-based approaches, and IMC-specific solutions, demonstrating that it delivers clear, high-quality images quickly, easily, and reproducibly. PENGUIN is available as a user-friendly Jupyter notebook, enabling the simple adjustment of thresholds and percentiles for each marker and streamlining the transformation and saving of images. The core code is also available for direct scripting. Compared to the recently developed Ilastik-based SABR method, PENGUIN eliminates the need for manual annotation or machine learning model training, preserves signal intensity variations across tissues, reduces noise, and retains more structural details. Additionally, PENGUIN improves critical downstream tasks like cell segmentation and phenotyping. Due to its simplicity, speed, intuitive interface, and reproducibility, PENGUIN is a valuable tool for the analysis of IMC data and, potentially, other multiplex imaging platforms.

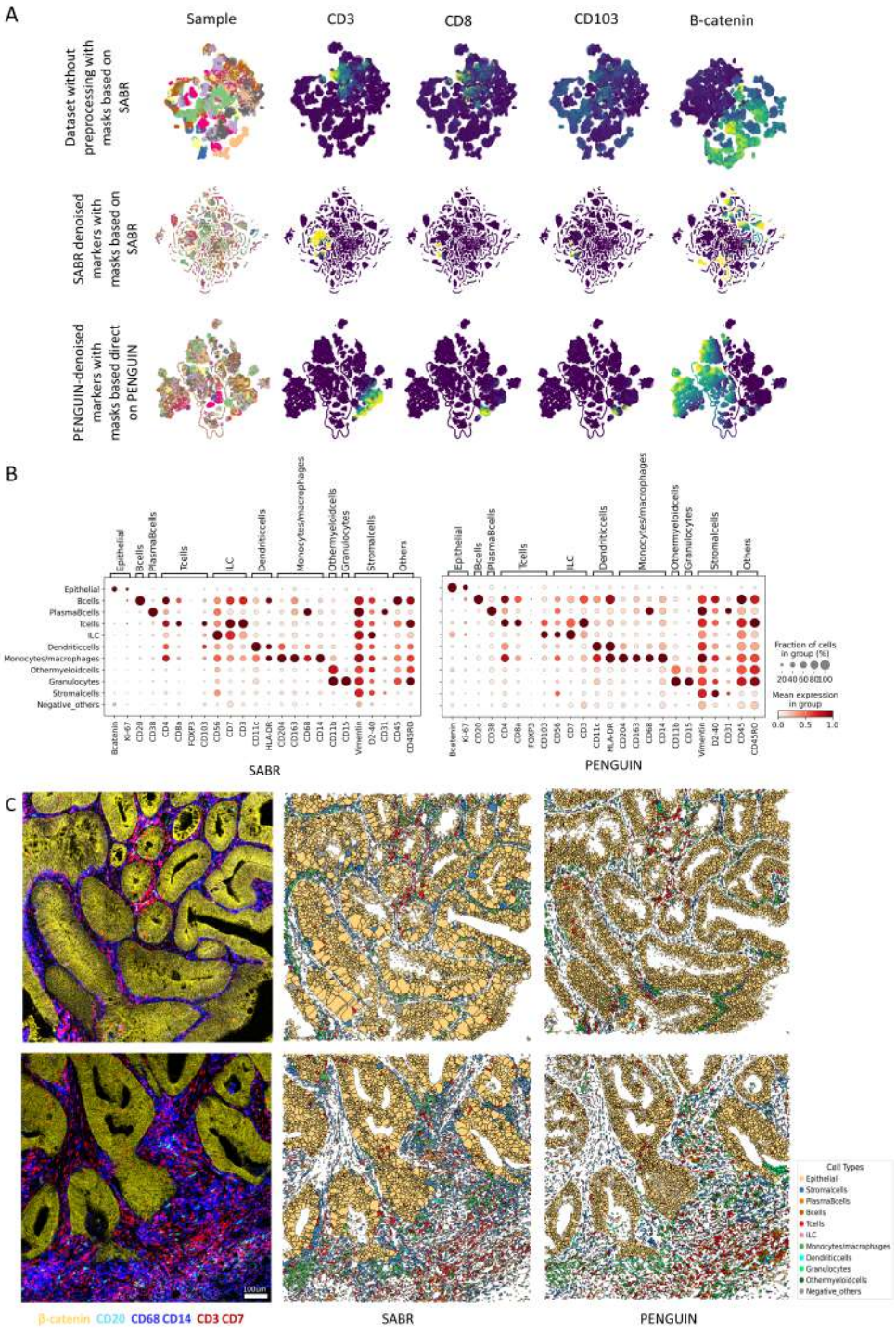


Figure 3.4: Overview of downstream analysis. A) t-SNE plots generated from raw marker expression with SABR Ilastik-based cell masks(top), SABR Ilastik-denoised marker expression with SABR Ilastik-based cell masks (middle), and PENGUIN-denoised marker expression and PENGUIN-based cell masks (bottom). The t-SNE plots are colored by sample, CD3, CD8, CD103 and b-catenin expression levels. B) Dot plot showing the scaled expression of marker genes across epithelial, stromal, immune lymphoid lineages (B cells, plasma B cells, T cells, innate lymphoid cells (ILC)) and immune myeloid lineages (dendritic cells, monocytes, macrophages, and granulocytes) for each cell type identified using cell segmentation and marker expression from SABR and PENGUIN denoising strategies. C) Comparison of cell segmentation and phenotype obtained using SABR and PENGUIN denoising strategies within a sample.

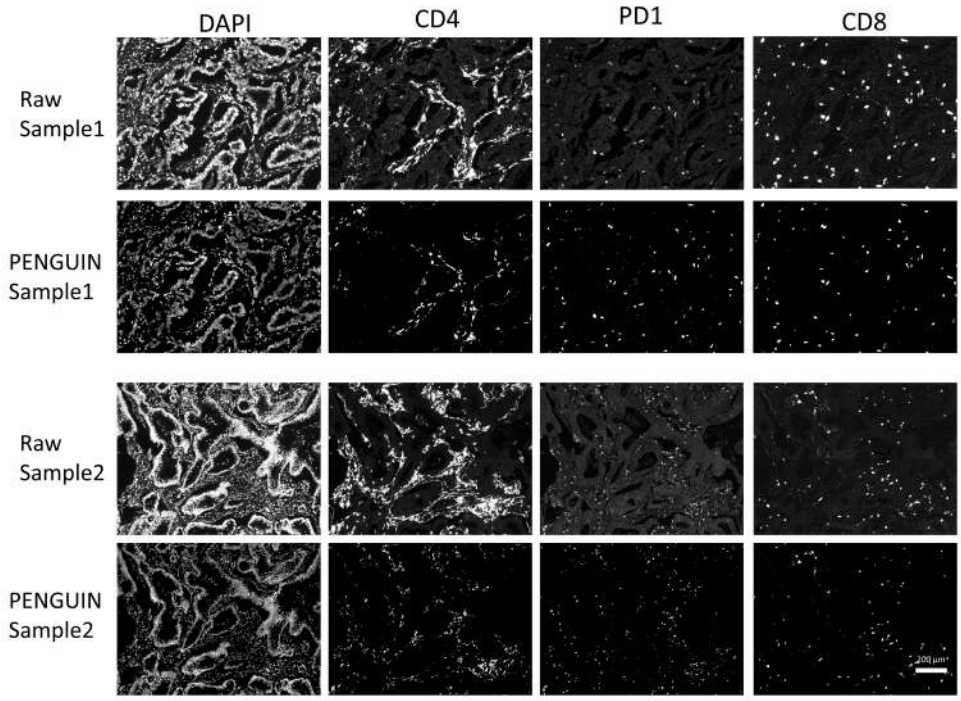


Figure 3.5: PENGUIN applied to IF images, specifically for DAPI, CD4, PD-1 and CD8 immunofluorescent detection.

3.4 Methods

All the tests were run in a Intel(R) Core(TM) i7-9700 CPU. The DL models were run using a GPU Nvidia TU102 GeForce RTX 2080 Ti.

3.4.1 Dataset

All the results for IMC were obtained using the publicly available cohort of 61 IMC raw images with 39 markers of colorectal cancer [251].

3.4.2 Denoising Methods comparison

Before application of the denoising filters, images were saturated at the 99th percentile and normalized between 0 and 1. Raw images were also tested, but normalization led to significant improvements in image quality. The following filters were applied to each channel: Gaussian filter with sigma values of 0.7 and 1; mean filter; percentile filter with values of 25, 50, and 75; non-local means filter using a patch size of 5, patch distance of 11, and sigma of 0.2); bilateral filter; total variation filter with a weight of 0.3; Wavelet filter; anisotropic diffusion filter and BM3D filter with sigma values of 0.1, 0.2 and estimated sigma values between these two. For parameters not explicitly specified, default settings were used. Gaussian, mean, and percentile filters were implemented using SciPy [253], while non-local means, bilateral, total variation and wavelet filters were based on the scikit-image package [254]. BM3D filtering was retrieved from its

source package [255]. The hot pixel removal filter for IMC images was implemented following Windager et al. [150], using a Scipy maximum filter with a 3x3 window. If the difference between the central pixel and its maximum neighbor exceeded a defined threshold, the central pixel was set to the value of the maximum neighbor. Thresholds ranging from 0.05 to 0.2 were applied to normalized images, while the default threshold of 50 was used for raw images. Noise2Void was trained per channel as described by Krull et al. [82], with 0.2 of the data used for validation. Loss was defined as MSE, with a neighborhood radius 5 and UNET kernel size 3 for a patch size of 64. Models were trained for 20, 50, and 100 epochs. The convolutional autoencoder for image denoising were built as described by Chollet et al [252]. The encoder consisted of two consecutive 2D convolutional layers (32 filters, kernel size 3x3, ReLU activation) followed by max pooling (pool size 2x2). The decoder included two transpose convolutional layers (32 filters, kernel size 3x3, ReLU activation) and a final convolutional layer with 1 filter and sigmoid activation. The Adam optimizer and binary cross-entropy loss function were used. Training was performed for 20 or 50 epochs per channel. The DL implementation was done using Tensorflow [55], with any unspecified parameters set to default. IMC Denoise [155] was implemented according to the instructions in the tutorial notebook, with default parameters. IMC Denoise consists of two steps: the DIMR algorithm and DeepSNIF. We tested both the DIMR algorithm applied per channel (using 4 neighbors, a window size of 3, and 3 iterations) and the combination of the DIMR algorithm followed by the DeepSNIF, which was trained for 50 epochs.

3.4.3 PENGUIN implementation

PENGUIN is implemented in Python 3.10 and available as both a code package and Jupyter notebook. For each image and channel, pixels above the 99th percentile are capped at that value and normalized to a scale of 0 to 1. Following this, pixels with values below a defined threshold are removed, and a percentile filter (using a 3x3 window) is applied to determine which pixels are either considered noise or signal. The same thresholds and percentiles are applied consistently across all images in the cohort for each channel. To demonstrate the approach, we tested percentile values of 90, 75, 50, 25, and 10, as well as the 50th percentile after removing pixels below thresholds of 0.1, 0.3, and 0.5. This comparison was performed for β-catenin, CD20, vimentin, CD45, and FOXP3, given their importance as markers and varying signal distributions. Two Jupyter notebooks are available: one for cases where each region of interest has a separate TIFF file per channel, and another for cases where all channels are contained in a single TIFF stack.

3.4.4 Comparison of methods for downstream analysis

3.4.4.1 PENGUIN parameters definition

The raw images were analyzed by an IMC expert using PENGUIN notebook for better visualization. Parameters such as thresholds and percentiles were adjusted for each channel using the Notebook tool (Supplementary Table S3.1) and then applied uniformly across all images in the cohort.

3.4.4.2 Ilastik-based background removal

For each marker, a random forest classifier was trained in Ilastik [153] to distinguish between background and signal pixels, as described by Ijsselsteijn et al.[154]. After thorough training on approximately 10% of the dataset, the classifier was applied to all images for a given marker. The data was then exported as simple segmentation masks, with background pixels set to zero and signal pixels set to one.

3.4.4.3 Denoising evaluation metrics

To evaluate the quality of denoising in the absence of an actual ground truth, we used SABR [154] as a reference, given its manual annotation and reliability as a close approximation. We calculated the PSNR and the SSIM to compare the performance of rescaled raw and PENGUIN-denoised images against the SABR output, treated as the reference standard. In order to make this comparison, positive pixels in SABR (i.e., pixels considered signal) were replaced with their original values from the rescaled raw image. PSNR is defined as:

$$\text{PSNR} = 10 \cdot \log_{10}\left(\frac{(\text{MAX}_I)^2}{\text{MSE}}\right)$$

where MAX_I is the maximum possible pixel value of the image, and MSE is the mean squared error. SSIM is defined as:

$$\text{SSIM}(x, y) = (l(x, y))^{\alpha} \cdot (c(x, y))^{\beta} \cdot (s(x, y))^{\gamma}$$

where luminance (l) is measured by averaging the pixel values, contrast (c) is calculated as the standard deviation of the pixel values, and structure (s) is derived from the covariance between the signals, normalized by the standard deviation. The parameters $\alpha > 0$, $\beta > 0$, $\gamma > 0$ represent the relative importance of each component. We used the scikit-image package for both PSNR and SSIM calculations.

3.4.4.4 Cell segmentation & phenotyping

Cell masks were created using CellProfiler V3.0 [156]. For comparison, masks were generated from both raw data and PENGUIN-normalized images. For raw data, a combination of Ilastik/CellProfiler was applied: first, probability masks for the nucleus, membrane, and background were generated in Ilastik using the DNA, vimentin and keratin images. These masks were then loaded into a CellProfiler pipeline to identify primary objects (nuclei), followed by membrane assignment using the "Identify Secondary Objects" module. Visual inspection was conducted to compare the masks with the original IMC Images. After PENGUIN normalization, the normalized images could be directly used to create cell segmentation masks in Ilastik using the same workflow.

The two different cell segmentation masks (Ilastik-based and PENGUIN-based) were loaded into ImaCytE [256], along with the raw, Ilastik-denoised, and PENGUIN-denoised marker profiles, to define relative marker expression per cell. FCS files were generated, and clustering of cells was performed using t-SNE in Cytosplore [257] to identify cell subsets.

3.4.4.5 Immunofluorescence labelling and imaging

Immunofluorescence labeling was performed using the OPAL methodology. Briefly, four μm thick sections from formalin-fixed paraffin-embedded tissues underwent deparaffinisation, rehydration, endogenous peroxidase blocking, and antigen retrieval with 10 mM citrate buffer (pH 6.0). Sections were then blocked with Superblock solution (Thermo Fisher Scientific) and incubated for 1 hour with anti-CD4. Opal amplification was performed by consecutive 10-minute incubations with a polymeric HRP-linker antibody conjugate (Immunologic, the Netherlands) and OPAL650 reagent (Akoya Biosciences). The sections were boiled for 15 minutes in 10 mM citrate buffer (pH 6.0). This process of blocking, primary antibody incubation, followed by HRP and OPAL amplification, was repeated for each target (Supplementary Table S3.2). Between each step, sections were washed with PBS-Tween. Finally, the sections were incubated with 1 µM DAPI, washed with PBS (without Tween) and mounted using Prolong® Gold Antifade Reagent (Cell Signaling Technologies). The slides were imaged and spectrally unmixed using the Vectra 3 imaging system and the InForm software (Akoya Biosciences), after which the component images were normalised with PENGUIN. Parameters, including thresholds and percentiles, were set for each channel (Supplementary Table S3.2) and applied across all images in the cohort.

3.5 Data and code availability

The IMC data used in this study is publicly available from Roelands et al. [251] and can be accessed through the BioImage Archive (Accession number S-BIAD587, `https://www.ebi.ac.uk/biostudies/bioimages/studies/S-BIAD587`). The PENGUIN pipeline is fully available and ready to use at `https://github.com/deMirandaLab/PENGUIN`. The repository also contains all the necessary code to replicate the methods comparison, along with detailed Jupyter notebooks for easy understanding and implementation.

3.6 Supplementary information

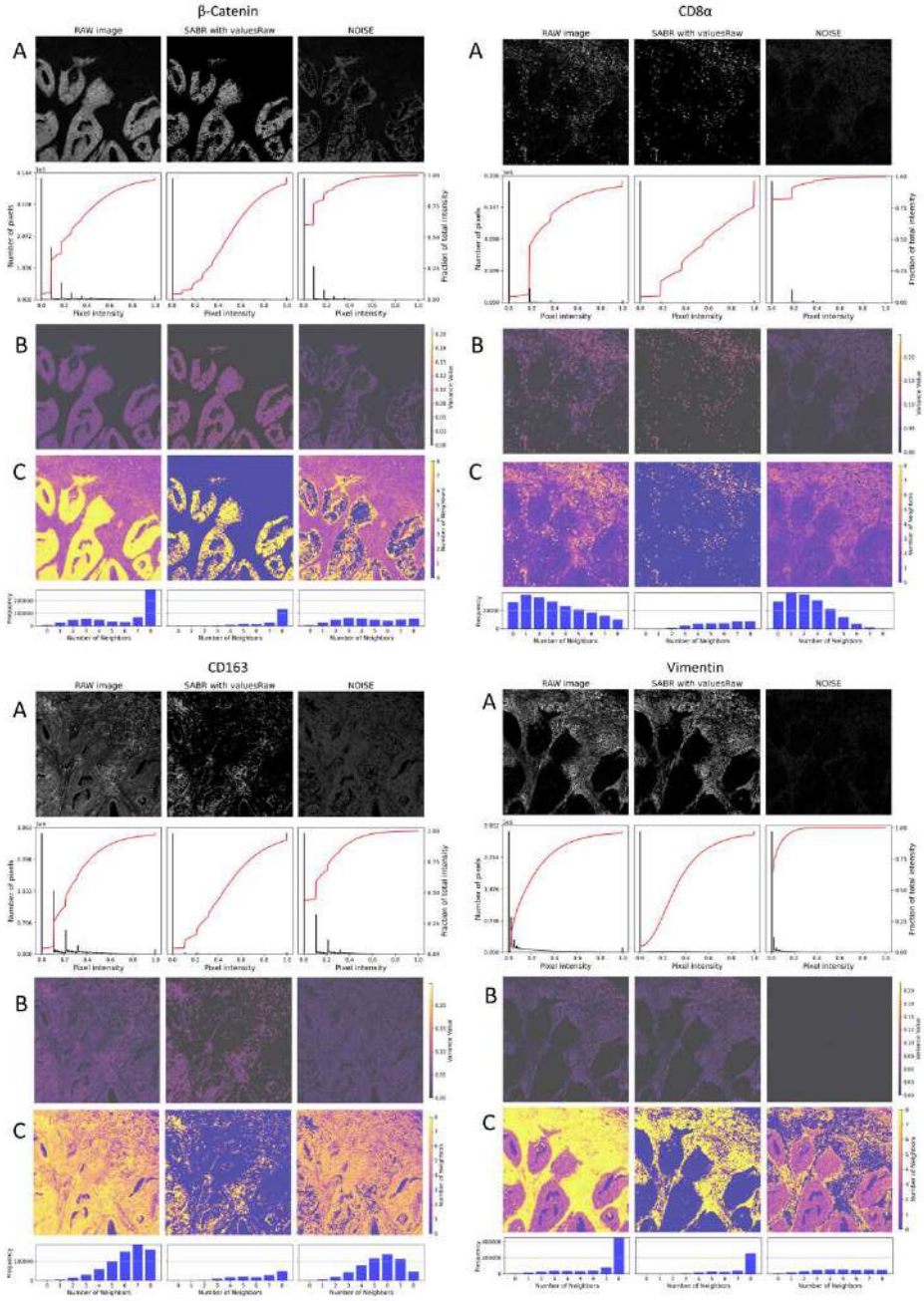


Figure S3.1: Noise and real signal across four channels of interest: β-catenin, CD8a, CD163, and Vimentin. To identify noise, images normalized with semi-automated background removal (SABR) [154] were used. For each channel, the following data were considered: raw image signal values (rescaled raw values); SABR images with rescaled values as an approximation of the ground truth, and the noise image, computed as the difference between the raw image and the SABR image (Noise = Raw – True). A) Panel showing the three considered images side by side. B) Histograms of signal distribution revealing the distribution of signal intensities for the three images. A notable proportion of low-intensity values in the noise image indicates the presence of noise. D) Local variance maps, calculated over a 3x3 window, capturing the variability of pixel intensities in the surrounding area. These maps help visualize the variability and inconsistency in noise-prone regions. D) Panel displaying the count of positive neighboring pixels for each positive pixel within a 3x3 window. It demonstrates that isolated pixels and those with fewer neighbors are more frequent in the noise images. Additionally, a histogram shows the frequency of pixels with different neighborhood counts, confirming that isolated pixels and those with fewer neighbors are more abundant in the noise images.

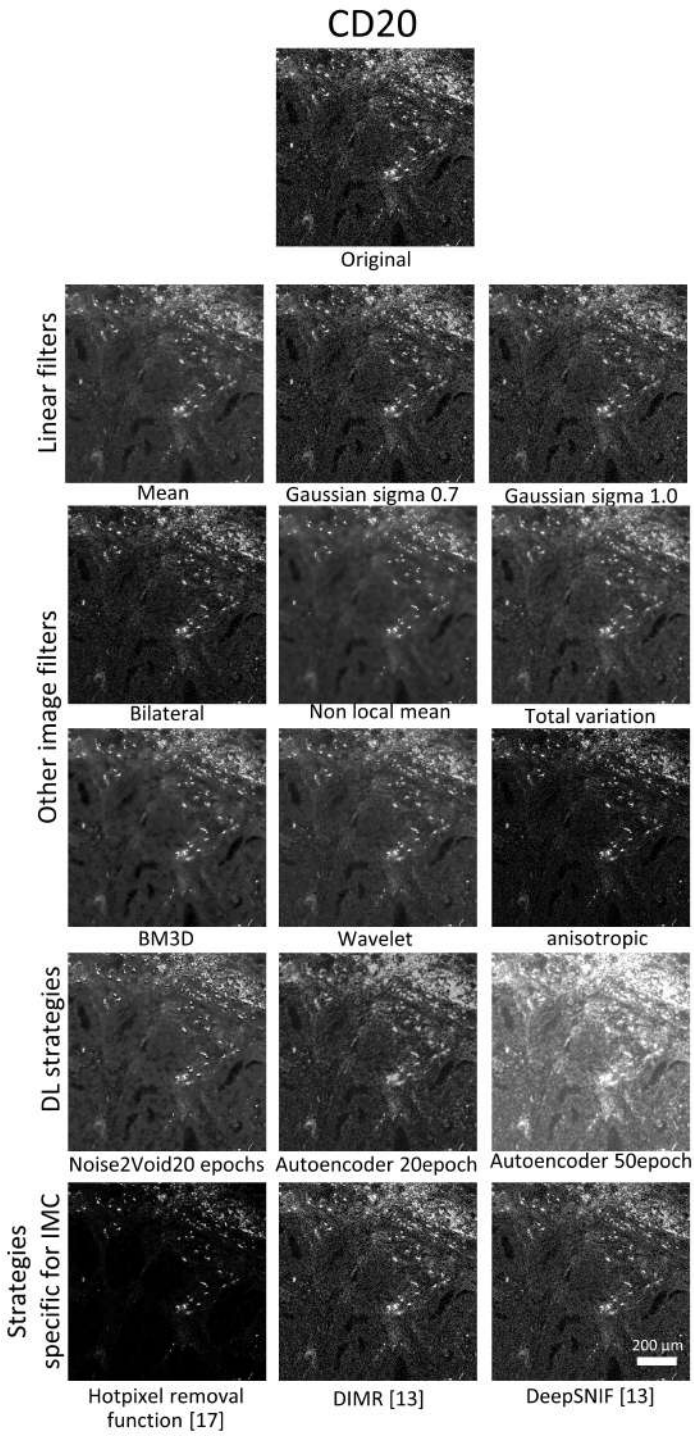


Figure S3.2: CD20 channel of a sample processed with various linear filters (mean, Gaussian with sigma 0.7 and 1); traditional image filters (bilateral, non-local means, total variation, BM3D, wavelet, and anisotropic); DL-based denoising strategies, including Noise2Void and a denoising autoencoder (trained for 20 and 50 epochs); and IMC-specific strategies, such as hot pixel filter from Windhager et al. [150], and DIMR with DeepSNIF from Lu et al. [155].

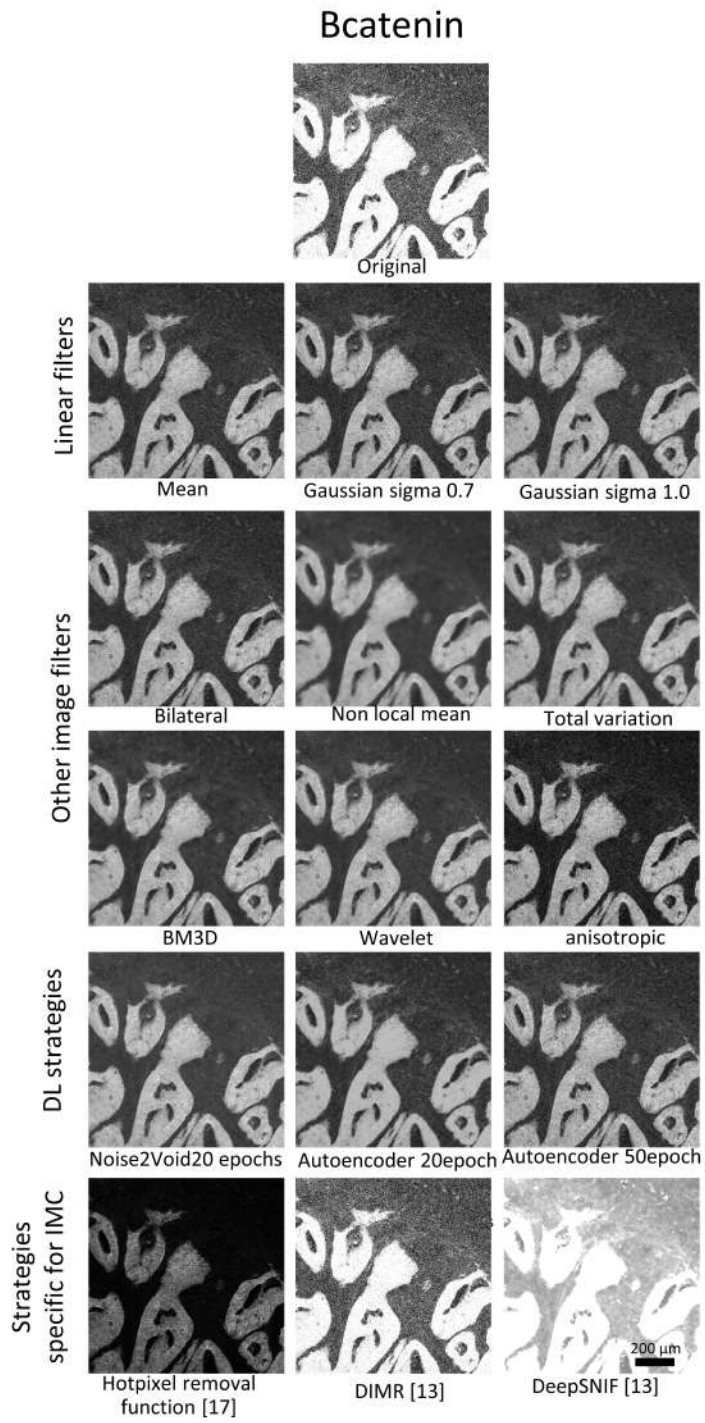


Figure S3.3: The β-catenin immunodetection channel of a single sample processed using a variety of filter approaches, including: linear filters (mean filter, Gaussian filters with = 0.7 and 1); traditional image filters (bilateral filter, non-local means, total variation, BM3D, wavelet transform, and anisotropic diffusion); and deep learning-based denoising strategies (Noise2Void and a denoising autoencoder trained for 20 and 50 epochs). Additionally, IMC-specific denoising methods are applied, including the hot pixel filter defined by Windhager et al. [150], as well as DIMR and DIMR combined with DeepSNIF from Lu et al. [155].

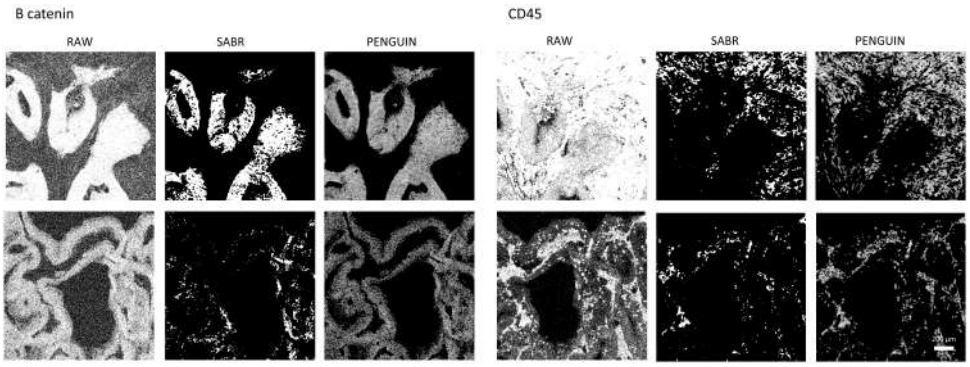


Figure S3.4: Zoomed-in view from Figure 3.3 focusing on β-catenin and CD45 immunodetection. In comparison to the SABR approach [154], PENGUIN preserves more real signal and retains key structural features of the cells.

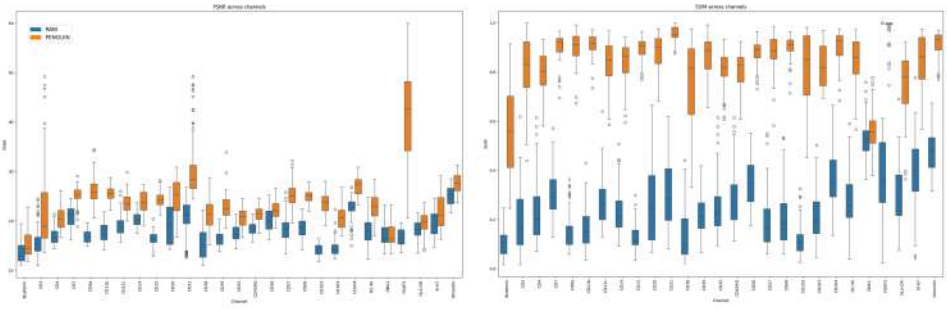


Figure S3.5: Boxplot comparing the Peak Signal-to-Noise Ratio (PSNR) and the Structural Similarity Index (SSIM) for RAW (rescaled) and PENGUIN-denoised images against the SABR dataset, which is used as the reference standard. The PSNR (left), shows significantly higher values for PENGUIN compared to raw images across most channels, indicating lower error and closer approximation to the SABR method. The SSIM (right), which ranges from -1 to 1, shows that PENGUIN achieves higher values than raw images, with medians above 0.8 and approaching 1 in most channels, indicating strong structural similarity.

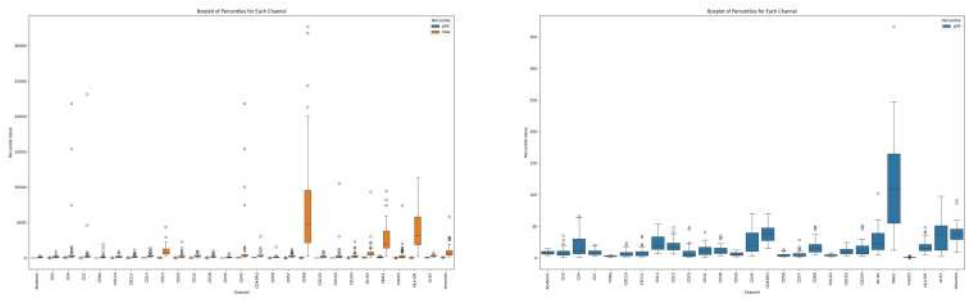


Figure S3.6: Boxplot displaying the maximum and 99th percentile values (left) and the 99th percentile values (right) across fields of view (FOVs) per channel for the raw data.

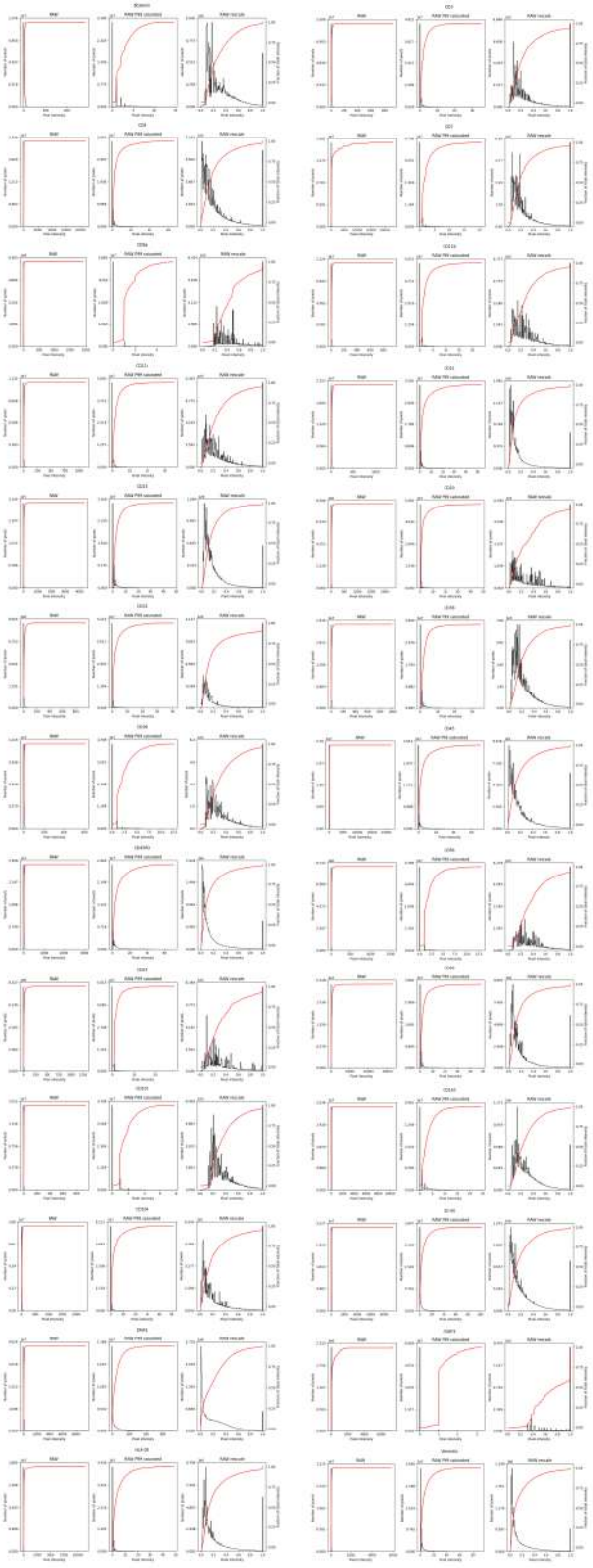


Figure S3.7: Histograms showing the signal distribution across all channels before and after normalization. For each channel, the signal distribution of all images in the cohort is combined. The plots display the raw signal, the raw signal saturated at the 99th percentile, and the normalized signal. To improve the clarity of the data, zero-value pixels were excluded from the histograms.

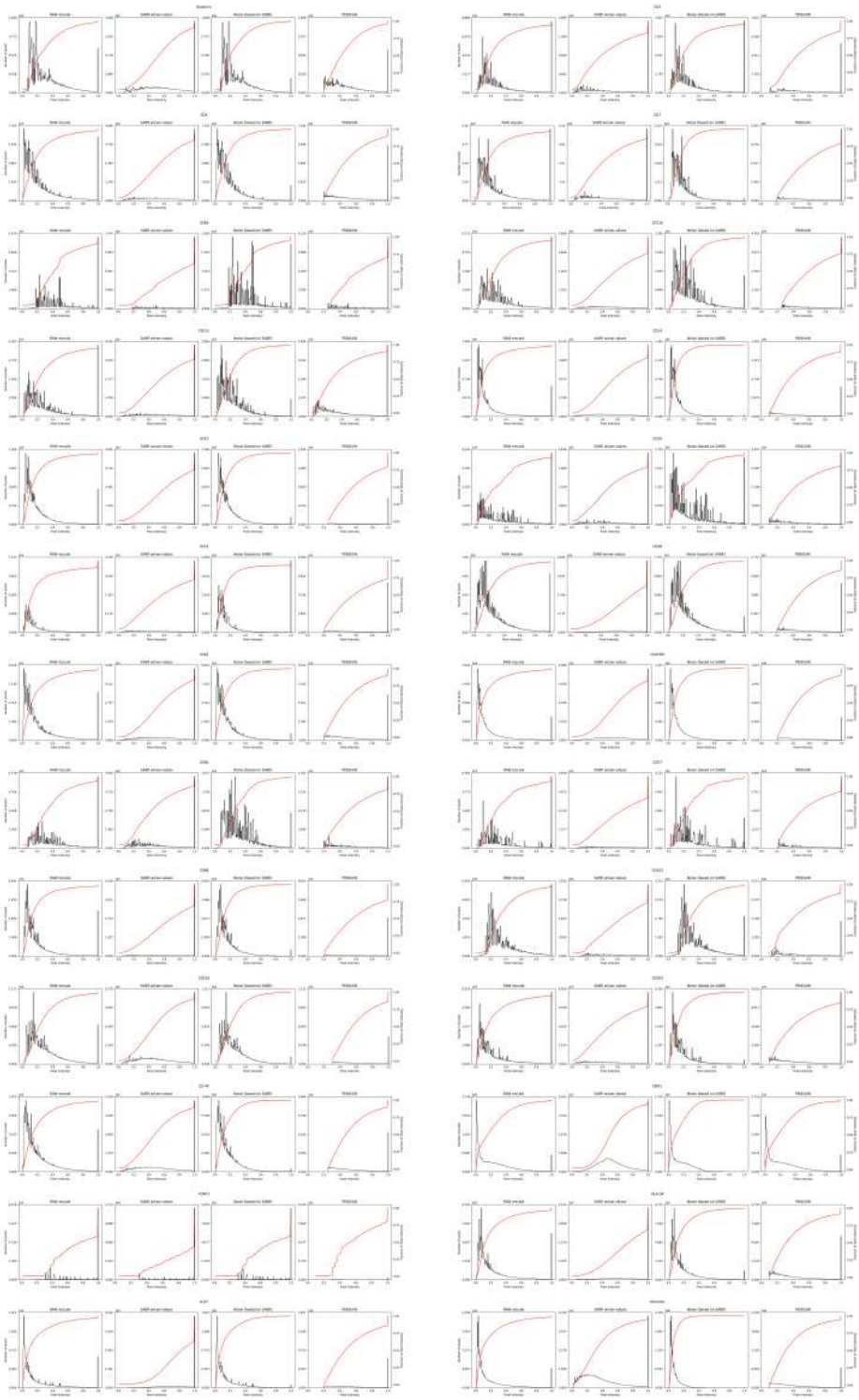


Figure S3.8: Histograms illustrating the signal distribution for positive values across all channels. For each channel, the signal distribution of all images in the cohort is combined. The histograms show the raw signal distribution (after saturation and normalization), the signal after applying the SABR and PENGUIN denoising methods, and the noise distribution, with SABR considered as the ground truth. Since SABR produces binary outputs (0/1), the histograms represent the raw normalized signal values for pixels classified as signal. Zero-value pixels were excluded to improve the visibility of the remaining data.

Table S3.1: Marker Percentiles and Thresholds.

Marker	Percentile	Threshold	Marker	Percentile	Threshold
B catenin	60	0.2	CD8a	60	0.25
CD103	30	0.1	cleavedcaspase	40	0.3
CD11b	50	0.25	D2-40	50	0.25
CD11c	40	0	FOXP3	50	0.1
CD14	40	0.1	Granzyme B	40	0.2
CD15	40	0.25	HLA-DR	50	0.1
CD163	40	0.3	ICOS	50	0.1
CD20	50	0.1	IDO	30	0.3
CD204	50	0.1	Keratin	50	0.5
CD3	50	0.1	Ki-67	50	0.2
CD31	50	0.2	P16ink4a	50	0.2
CD38	40	0.2	PD-1	40	0.2
CD39	40	0.2	PD-L1	50	0.3
CD4	50	0.2	Tbet	60	0.1
CD45	50	0.2	TCRgd	40	0.1
CD45RO	50	0.2	TGFbeta	40	0.2
CD56	50	0.2	Vimentin	40	0.1
CD57	60	0.2	VISTA	50	0.2
CD68	40	0.2	CD7	50	0.2

Table S3.2: Thresholds and percentile values for the IF markers.

Marker	Channel	Percentile	Threshold
DAPI	DAPI	60	0.2
CD4	Opal650	50	0.5
PD1	Opal570	40	0.7
CD8	Opal690	40	0.7

4

Cell segmentation in multiplexed proteomics: a comparative analysis

Cell segmentation is a critical step in the analysis of multiplexed imaging data, directly influencing downstream analyses and biological insights. In this study, we provide a comprehensive comparison of three widely used cell segmentation algorithms—CellProfiler, Mesmer, and Cellpose—on imaging mass cytometry (IMC) data from colorectal cancer (CRC) tissues, covering a spectrum from classical segmentation techniques to deep learning-based approaches. We evaluate these methods not only in their standard one-go segmentation approach but also in a sequential segmentation strategy, where different cell types are segmented with specific, optimized parameters. By assessing segmentation outputs in terms of cell numbers, morphology, phenotyping, and the distribution of lineage markers, we explore the impact of these approaches on subsequent analyses, such as immune infiltration classification. Our results highlight the unique strengths and limitations of each method: CellProfiler often produces irregularly shaped cells with angular borders, Mesmer tends to over-segment cells into smaller, rounded shapes, while Cellpose achieves a balance of natural cell shapes and varying sizes, particularly after fine-tuning. We demonstrate that sequential segmentation mitigates the risk of losing small, infiltrative cells in dense tissue environments, though at a higher computational cost. Moreover, to support further model development, we introduce a manually annotated dataset of over 1,900 regions of interest (ROIs) from CRC tissues, including lymphoid, myeloid, vimentin, and epithelial cells. This dataset offers a valuable resource for training and fine-tuning cell segmentation models. Our study underscores the importance of selecting appropriate segmentation strategies tailored to specific research needs and highlights the potential of deep learning models in advancing spatial proteomics.

Keywords: Cell segmentation, Multiplex imaging, Spatial proteomics, deep learning, sequential segmentation, comparison

---

This chapter was written in collaboration with Dr. Marieke Ijsselsteijn and is currently being prepared for publication.

4.1 Introduction

Spatial omics has revolutionized research by enabling the study and understanding of the organization of complex tissues [11]. Specifically, spatial proteomics techniques like CODEX [8], MIBI-TOF [9] or IMC [10], generate multiplexed images of the protein composition in tissues. A critical step in these analyses is cell segmentation, where each pixel is assigned to a specific cell. Despite its importance, cell segmentation, remains one of the most challenging tasks, with inaccuracies potentially leading to misinterpretation of data [157, 160, 161]. Challenges arise due to variability in cell shapes and sizes, tissue density (cells tightly packed together) and limitations of 2D sections in capturing 3D cellular structures [160, 161].

Approaches to cell segmentation vary widely. While some approaches aim to segment the nucleus and generate expansions to mimic the full cell, others use whole cell segmentation using both nucleus and membrane markers. In terms of algorithms, some classical imaging analysis techniques make use of the gradient field of the image to detect boundaries, as it is the case of the watershed algorithm that treats pixels as local elevations (topography) [157]. On the other hand, more recent techniques are based on deep learning (DL) and trained on large, labelled datasets. Here, a distinction can be made between models that specifically use data from spatial proteomics [160] or ones that leverage the power of foundation models trained with different types of data [161, 258, 259].

Parameter tuning is critical to the performance of cell segmentation, as for instance the extension of segmentation and cell splitting can lead to larger, potentially under-segmented, or smaller, potentially over-segmented, cells. This is an important consideration, as cells in a tissue vary in size and shape, e.g., a cancer cell compared to a T-cell, and are thus difficult to segment with a single set of parameters. Because of this, and to prevent loss of small cells that overlap with tumour cells, a solution may be sequential segmentation, where different cell types are segmented with distinct settings and are processed in a prioritized order [260]. Maldegem et al [260] introduced a pipeline leveraging sequential segmentation and using propagation-based thresholding strategies. Later, it was improved into a multiplexed consensus cell segmentation pipeline [261], which uses nuclear predictions and progressive masking of specific marker channels. Recognizing the same problem, Allam et al., in an effort to derive an immune infiltration score from cancer tissues, demonstrated that pixel-level classification could be superior to cell segmentation to detect highly heterogeneous cell populations, including CD8, M1, and M2 cells, spatially neighbouring cancer cells [262]. It is important to note that, despite its difficulty to balance the trade-offs of under- and over-segmentation in tissues with heterogeneous cell types, one-go segmentation is still the most common approach.

To our knowledge, no systematic comparison of one-go versus sequential segmentation methods has been conducted. Here, we describe the use of classical and deep learning methods to compare these approaches. Additionally, we introduce a manually annotated dataset of over 1,900 Regions-of-interest (ROI) from colorectal cancer tissues, containing lymphoid, myeloid, vimentin, and epithelial cells.

4.2 Results

The segmentation approaches benchmarked in this comparative study were applied to a public dataset of 39 cellular markers analysed by IMC across 61 samples of CRC [263]. The raw IMC images were denoised and pre-processed as described in [264] prior to cell segmentation.

4.2.1 Derivation and evaluation of segmentation algorithms

In this work, we compared three segmentation algorithms that are commonly used in IMC. These methods utilize different segmentation strategies, including traditional image processing, DL or both. The first, CellProfiler is an open-source image analysis program that enables the generation of modular pipelines for processing microscopy images. For segmentation, CellProfiler employs traditional algorithms like thresholding, edge detection and Watershed [157, 265]. The second approach, Mesmer, part of DeepCell, is a deep learning-based method that integrates a ResNet50 backbone with a feature pyramid network to predict cell centroids and boundaries. These predictions are combined with a Watershed algorithm to generate final masks. Mesmer is trained on TissueNet, a dataset with over 1 million annotated cells from methods like CODEX, MIBI, IMC, and others, constructed with a human-in-the-loop approach [160]. The third approach, Cellpose, is a generalist DL method that uses a U-Net backbone to predict vector flow representations of cells, which are then converted into masks [161, 162]. While being a generalist model trained on a variety of images, including fluorescence, brightfield microscopy and non-microscopy images [161], Cellpose also provides specialist models, such as those trained on TissueNet and its subsets TN1 and TN2, separated based on different segmentation styles [162].

As Cellpose offers a variety of models, we tested the Cellpose generalist model "cyto3" and specialist models, trained on TissueNet and its subsets "TN1" and "TN2". We found segmentations from TissueNet and the TN1 to perform more accurately compared to the generalist model.

To fully utilise the functionality of Cellpose for fine-tuning, we manually annotated 25 Field-of-Views (FOV)s of 256×256 pixels, capturing nuclei in combination with various membrane markers: lymphoid (7 FOVs with 461 ROIs), myeloid (6 FOVs with 321 ROIs), Vimentin (7 FOVs with 532 ROIs) and with tumour markers (5 FOVs with 628 ROIs). Overall, this represents a dataset with 1942 ROIs annotated (Table 4.1), where a cell is only considered with both membrane and nuclei. This dataset was further split into training (roughly 60% of the ROIs) and test (40%) partitions, representing a valuable resource for optimizing segmentation models. To our knowledge, it is the first IMC dataset annotated in this fashion.

Using this dataset (training examples), we fine-tuned three models: the generalist cyto3, the TissueNet Cellpose and the TN1 Cellpose, training for 500 epochs (see methods section for other default parameters). The segmentations were assessed on pixel level metrics, visually and on a ROI level on a separate test set (Figures S S4.1, S S4.2, Table S4.1). The previously tested generalist model failed to predict cells for all cell types, while after training the model on the annotated dataset, the model performs comparable to the other models. Retraining TissueNet and TN1 resulted in a lower number of false positives (specially in non-epithelial areas) compared to their untrained counterparts. Furthermore, predicted cells were

Table 4.1: Number of FOVs and ROIs in training, test, and total annotation sets across the different mask types.

Mask type	**Training**		**Test**		**Total**	
	FOVs	ROIs	FOVs	ROIs	FOVs	ROIs
Lymphoid	5	315	2	146	7	461
Myeloid	4	211	2	110	6	321
Vimentin	5	297	2	235	7	532
Tumour	3	350	2	278	5	628
Total	17	1173	8	769	25	1942

bigger, possibly associated to the training of epithelial masks in the dataset that contains bigger cells. Training from scratch just with the training dataset also achieved good results. As the optimal results were obtained for the TN1-retrained, all further comparisons will be performed using this model, utilising it for both one-go and sequential segmentation.

Using the same test set, we evaluated the performance of CellProfiler and Mesmer. To allow comparison between the three methods, Mesmer and CellProfiler masks were filtered to include only cells with high expression of the membrane markers (Figure S S4.3). The performance of each method was evaluated at a ROI level by visual inspection (Figure 4.1, details on Figures S S4.4- S S4.7, Table S S4.2) and at pixel level (Figure 4.2, Table S S4.3) by calculating standard segmentation metrics, such as the IoU threshold and Dice coefficient to summarize how well the predicted mask overlaps with the true mask.

It was observed that the segmentation performance is highly dependent on the cell type, with tumour cells consistently yielding the highest scores (e.g., Dice/IoU often > 0.8) across all methods, while Vimentin and lymphoid cells appear most challenging (lowest scores). For the lymphocyte mask, Cellpose-based models generally demonstrated higher IoU and Dice coefficients. Mesmer generally demonstrates the strongest performance across most metrics and cell types. Retraining Cellpose yielded mixed results, improving scores for specific types like lymphocytes, but without significantly or consistently outperforming the original Cellpose model or Mesmer; the differences between the original TN1 model and the retrained version did not show clear differences. Methods like Mesmer often achieve very high Precision (e.g., > 0.9 for tumour), suggesting few false positive pixels, while Recall varies more significantly, indicating challenges in capturing all positive pixels. CellProfiler generally yields lower scores compared to the deep learning methods (Cellpose, Mesmer) across most metrics and cell types, highlighting the performance gains achievable with deep learning approaches for this task.

At the ROI level, different tendencies for over/under-counting objects were observed. Retrained Cellpose under-counts, while original Cellpose, Mesmer and CellProfiler over-count certain cells. This highlights that high pixel-level scores do not always guarantee accurate object enumeration. Overall, Mesmer consistently achieves high F1 scores, indicating a good balance between finding correct objects (Recall) and avoiding false detections (Precision), even though it may slightly over-segment some cell types such as vimentin (non-lymphoid and myeloid) and tumour.

Several factors should be considered when interpreting these quantitative scores. Firstly, the inherent

behaviour of CellProfiler and Mesmer, which can predict nuclei even with weak membrane signal, might lead to inflated area measurements, despite filtering is based on marker expression. Secondly, the small size of the test set limits the generalizability of these findings, and caution is warranted, particularly as these segmentations are intended for integration in downstream analyses. Furthermore, a key qualitative observation across all methods was that precise boundary alignment with manual annotations was often imperfect. However, this was frequently mitigated as the methods generally succeeded in identifying the correct positive areas, ensuring relevant regions were not missed, even if boundary definitions varied sometimes leading to over- or under-segmentation relative to annotations. Finally, morphological differences were noted: Mesmer and Cellpose-based models tended to produce smoother, more rounded shapes, whereas CellProfiler often generated more irregular outlines.

To generate comprehensive cellular masks for subsequent analysis, the algorithms were implemented using both one-go and sequential strategies. In one-go segmentation, cells were assigned based on nuclei and a composite of the membranous markers β-Catenin and Vimentin, to account for both epithelial and stromal/immune cells. In the sequential approach, cells were segmented in the following order: lymphoid cells, using nuclei and membrane markers CD7, CD20 and CD38; myeloid cells, using nuclei and membrane markers CD68, CD14 and CD11c; remaining stromal cells, using nuclei and Vimentin; epithelial cells, using nuclei, β-Catenin, and Keratin; and the remaining cells, segmenting unassigned nuclei. After each round, segmented cells were removed from the mask to avoid double identification. It Is important to note that Mesmer, CellProfiler and Cellpose have some key differences. While Mesmer and CellProfiler can segment cells with only a nucleus but missing membrane markers, Cellpose only segments cells that consist of both a nuclear and membrane signal. For this reason, the pipelines required additional filtering to ensure segmented cells had at least 40% membrane marker coverage. Parameters like the interior threshold (Mesmer), typical diameter of objects (CellProfiler) and flow threshold (Cellpose) were tuned to accommodate different cell morphologies and differences in size between, for example, tumour cells and lymphocytes. The final masks were obtained by merging all the generated masks, with the same sequential order priority.

In terms of processing time, both Mesmer and Cellpose models were run programmatically and utilized GPUs, while CellProfiller was executed using CellProfiller Desktop app. CellProfiler took 10 minutes. For the Cellpose model (using the TN1 Cellpose model as an example), the One GO pipeline segmented all 61 IMC images in the cohort in 249 seconds (4.15 minutes), averaging 4 seconds per image. In contrast, the sequential pipeline took 982 seconds (16.4 minutes), with an average of 16 seconds per image. Retraining the Cellpose model required 76 seconds. The Mesmer pipeline completed segmentation of all 61 images in 484 seconds (8 minutes), averaging 7.76 seconds per image. When run sequentially, it took 54 seconds per image. The extended processing time for the sequential pipeline is expected, as each image requires five cycles of nucleus-membrane composite creation, model prediction, and filtering, followed by the generation of a final composite. Additionally, more parameters must be fine-tuned in this approach. Although there is a difference in processing times between the Cellpose and Mesmer models, this difference is not considered significant and does not present a bottleneck in the workflow.

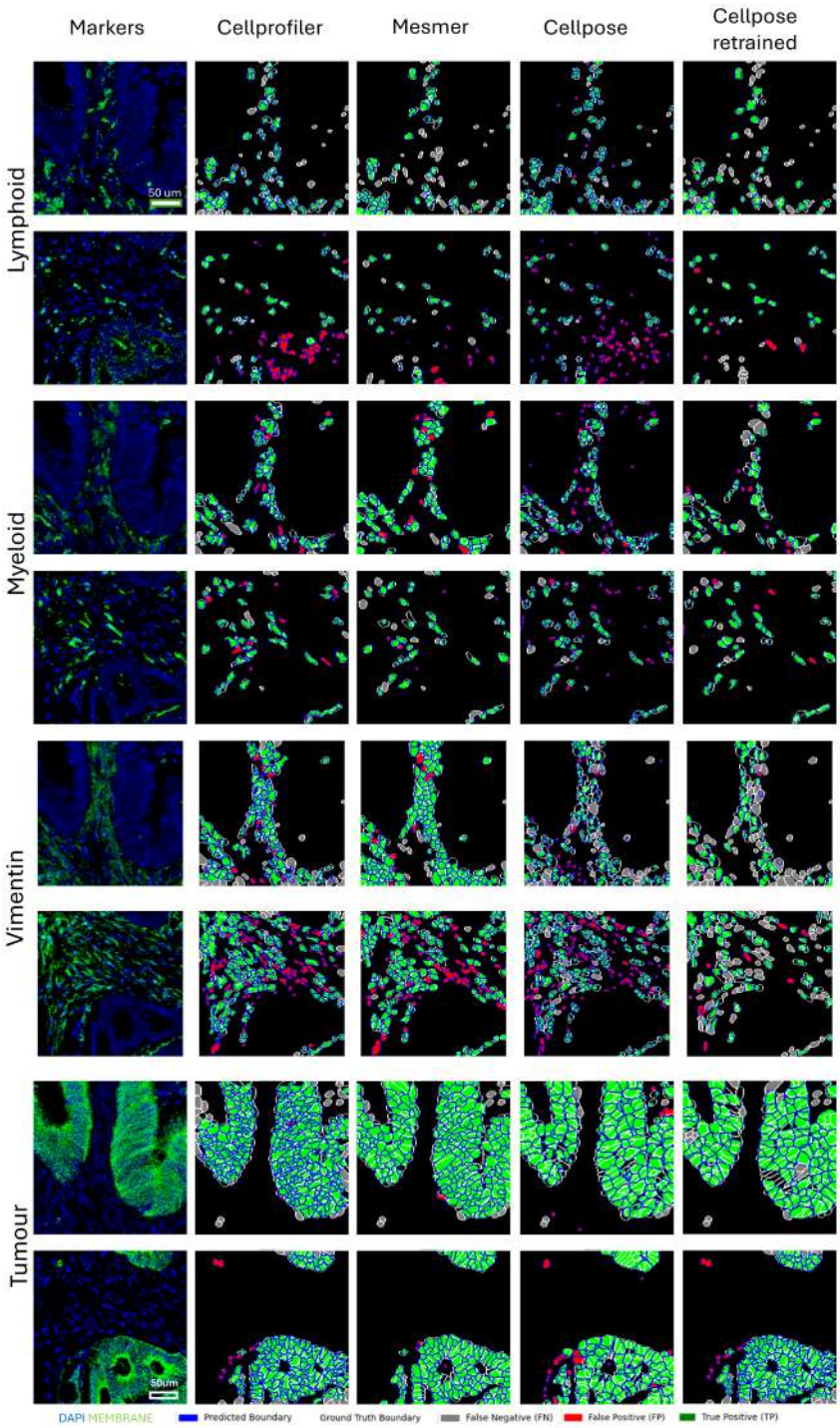


Figure 4.1: Evaluation of Segmentation Methods (CellProfiler, Mesmer, Cellpose, and Cellpose retrained) on the test set for Lymphoid, Myeloid, Vimentin, and Tumour Masks. Evaluation of the segmentation results at ROI level. Green represents true positive objects, red indicates false positives, and gray shows false negative areas. Due to the low overlap between annotated and predicted objects, a 0.2 threshold of overlap between masks was applied instead of using the IoU score for ROI evaluation. In this approach, if more than 0.2 of the predicted object's area overlaps with positive annotated pixels, it is considered a true positive; otherwise, it is classified as a false positive. Objects in the annotations that do not overlap with predicted pixels are coloured as false negatives. White lines represent the annotation segmentation boundaries, while blue lines indicate the predicted segmentation boundaries.

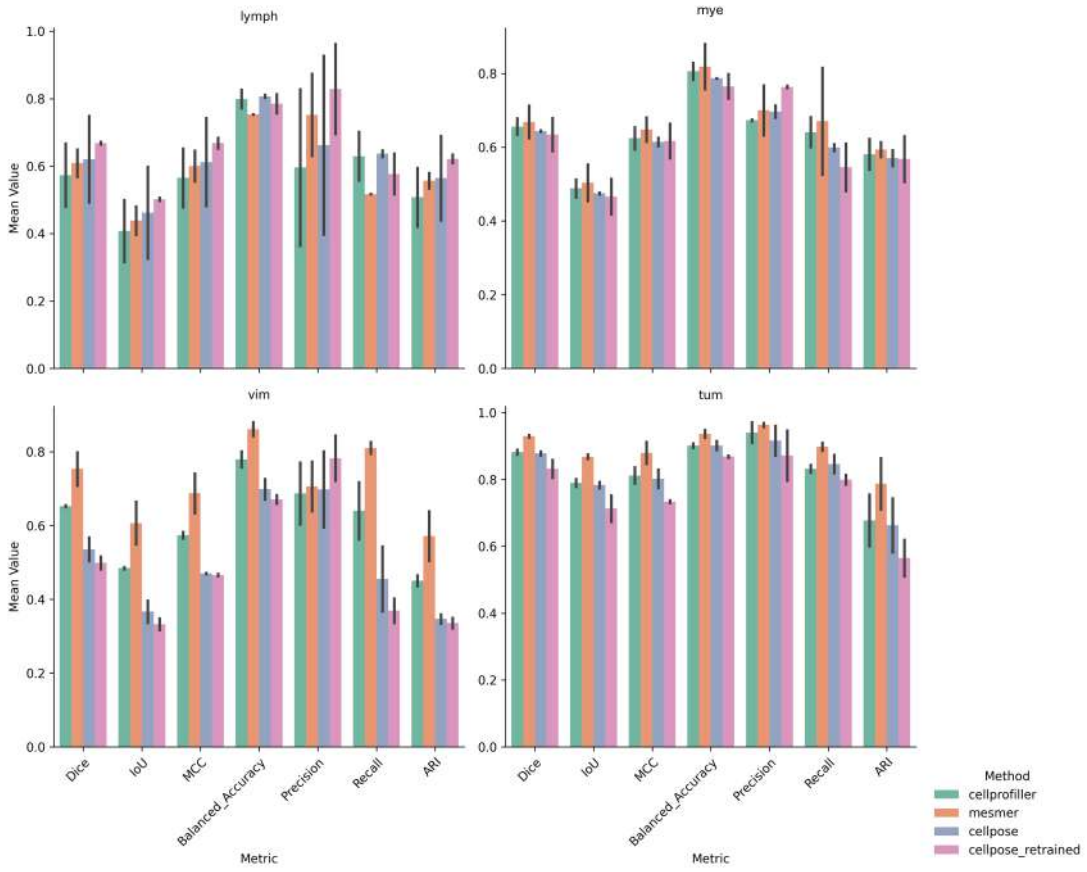


Figure 4.2: Evaluation of Segmentation Methods on the test Set at pixel level. Bar plots displaying pixel-based evaluation metrics for CellProfiler, Mesmer, Cellpose, and Cellpose retrained on the test Set for Lymphoid, Myeloid, Vimentin, and Tumour Masks. Dice, IoU, MCC, Balanced accuracy, precision, recall and ARI are displayed.

4.2.2 Phenotyping, morphological and intensity assessment

To further investigate how the cell segmentation approaches influence downstream processing and data interpretation, we checked the differences in cell type identification. Cell phenotyping is typically performed by dimensionality reduction and clustering algorithms, to group cells based on their marker expression [178, 179, 266]. Due to the number of sets to be compared, and to eliminate any potential bias from manual gating clusters in different cohorts, we deployed a simple method to phenotype all cells based on a decision tree (Figure S S4.8). Simply put, if either β-Catenin or keratin are higher than any of non-tumour markers, the cell is considered epithelial. Non epithelial cells are classified further, in a sequential manner and taking into account different markers and respective thresholds, as a CD3+ T cell, CD20+ B cell, CD68+ macrophage, CD14+ monocyte, CD15+ neutrophil, CD31+ vessel, another CD45+ cell, or Vimentin+; the remaining are labelled as Other. Important to note is that this approach is restricted to major cell populations and does not identify rare subsets.

We applied the segmentation approaches optimised in the previous sections to the full cohort of IMC images, both in one-go and sequential manner. We then investigated morphological characteristics,

intensity distribution and cell type abundance.

In terms of cell type abundance (Figure 4.3A-B, S S4.9-S4.10), some discrepancies are noted between counts of cells and area covered, due to over and under segmentation. Sequential methods cover more total area than their sequential non counterparts, with Cellpose retrained having the highest total area covered even with a smaller number of cells, followed by Mesmer, CellProfiler-sequential and OneGo, which had the least covered area. A comparison of the different cell types (Figure 4.3A) showed that Cellpose retrained and Mesmer-sequential cover more area for T cells, monocytes, B cells and macrophages. Mesmer (both ways) had the most area covered by undefined cells. Sequential models also tended to have more area covered for epithelial cells. Mesmer is the method that presents more cells overall, however, for T cell, monocytes, B cells and macrophages Cellpose sequential, followed by Mesmer models, presents overall more cells.

Compared to one-go segmentation, the sequential approach produced more and smaller cells (Figure S S4.11). Large differences were observed between the distinct approaches, as Cellpose and Mesmer presented the lowest number of small cells (3666 and 6955 respectively) and CellProfiler-sequential presented the largest (75688). This may be due to the method in which CellProfiler merges the masks, which may lead to fragmented cells. The distribution of sum of intensity (Figure S S4.12) was comparable between the methods and peaked around 1.25-1.5. For quality assurance, cells with an area encompassing less than 10 pixels were excluded.

Importantly, despite differences between approaches, across all methods and segmentation strategies, cell expression patterns for marker genes remained consistent, with clear expression profiles corresponding to their phenotyped categories (Figure 4.3C). This similarity is expected, as cell phenotyping directly relies on these markers for classification.

4.2.3 Downstream analysis comparison

While most cells remained phenotypically identical across methods, differences might arise from cell borders, over-segmentation, or under-segmentation. Changes in detection of tumour infiltrates or misclassified cells, if happening, affect only a small subset of cells relative to the total population, making overall differences in marker expression imperceptible. However, these subtle variations could be critical for more detailed analyses of the tumour microenvironment, particularly in detecting rare infiltrating cell populations or resolving edge cases.

To evaluate whether segmentation impacts challenging cases of cell identification, we focused on intraepithelial T cells. Using Voronoi diagrams, we extracted the 1^{st}-hop neighbourhoods of cells, calculating the number of cells with more than 50% epithelial neighbourhoods (Figure 4.4, S4.13). Our results revealed discrepancies in the values across segmentation methods for certain FOVs, not only on number of T cells but also in the percentage that are surrounded by epithelial neighbours. It is relevant to note that in FOVs with more cells, the differences get less noticeable.

Finally, to produce an overview of the final segmentation results, Figure 4.5 illustrates two $400 \times 400\ \mu$m crops of distinct tissue sections, showcasing marker overlays with segmentation boundaries and

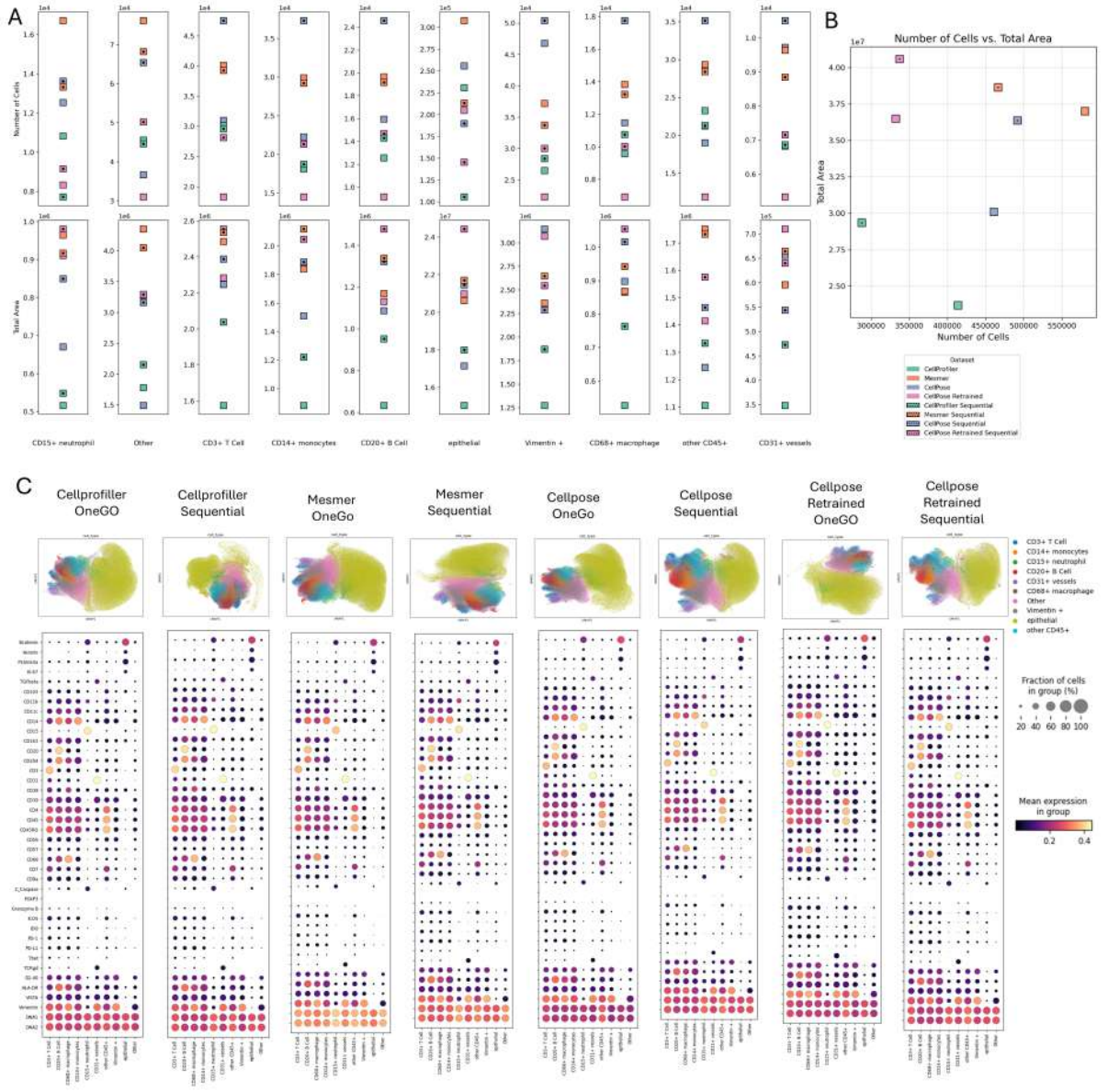


Figure 4.3: Comparison of phenotyping, morphology, and marker intensity across cell segmentation methods. (A) Description of total number of cells and area covered across each cell type across methods. Detailed boxplots of this information are provided in Figures S S4.9 and S S4.10.(B) Relation between area covered and overall count of cells across the 8 segmentation methods(right).(C) UMAP of all cells in each segmentation style coloured by the phenotype signed and dotplot of the expression of all markers across each phenotype.

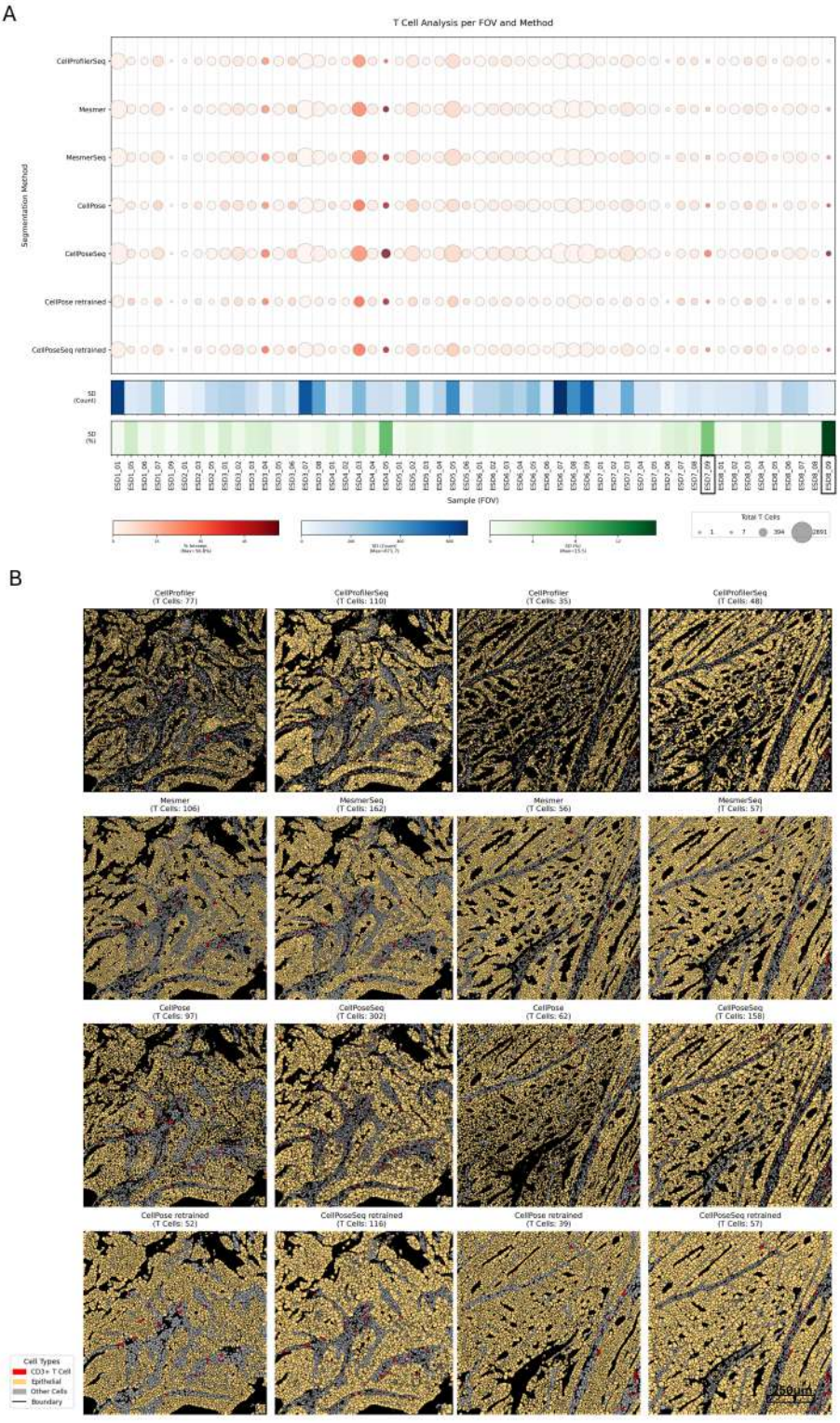


Figure 4.4: Comparison of T cell quantification across segmentation methods and FOVs. (A) Dotplot that represents the number of total T cells and percentage of intraepithelial T cells. Bubble color indicates the percentage of T cells classified as intraepithelial, the bubble size is proportional to the total number of total T cells. The two horizontal strips illustrate the variability between the different methods: in blue the Standard Deviation (SD) of the total T cell count (darker blue: more variability); in green the SD of the percentage of intraepithelial T cells across (darker green: more variability). (B) The 2 FOVS with higher variability for intraepithelial percentage are plotted for all methods (in red are T cells, in yellow Epithelial cells and in grey all other cells).

the corresponding phenotyped images. The most apparent differences lie in cell shape and size: CellProfiler tends to produce irregular, angular cell boundaries, whereas Mesmer results in rounder, smaller cells. Sequential models are often better distinguishing the size variations between tumour cells and other cell types. Additionally, segmentation methods differ in the spatial coverage of cells. CellProfiler tends to leave larger empty spaces, while other methods fill more the available space. While differences in immune cell infiltrates are visually noticeable, they remain challenging to quantify precisely. Note that the finetuned Cellpose model did not segment membranes without nuclei, but rather increased the overall size of cells, probably due to the bias in the training set that contains annotated epithelial cells with bigger diameters.

4.3 Discussion and conclusion

In this study, we presented a comparison of three widely used cell segmentation algorithms — CellProfiler, Mesmer, and Cellpose—applied to an IMC cohort of CRC for whole cell segmentation. These methods, ranging from traditional image-processing approaches to advanced deep-learning models, were assessed both in a single-pass segmentation mode and in a sequential approach designed to address specific challenges in cell segmentation, particularly in highly dense tissue environments. The segmentations generated were assessed on their cell numbers, morphology, phenotyping, expression of lineage markers and distribution of these phenotypes and, finally, on the eventual consequences on downstream analysis, such as classifying a tumour according to its immune infiltration. The comparison performed in this work highlighted the strengths and limitations of each method. CellProfiler tended to produce irregularly shaped cells with angular borders, and sometimes over-segmented cells into larger fragments. In contrast, Mesmer created smaller, rounded cells, often leading to over-segmentation. Cellpose, especially after retraining the model on a manually annotated IMC dataset, achieved a more balanced segmentation, producing more natural cell shapes and accounting for varying sizes. Sequential segmentation, accounting for diverse cell sizes and shapes, may mitigate the challenge of annotating crumbled or overlapping cells. Although the runtime of sequential segmentation is higher, the trade-off might be justifiable as the risk of losing infiltrate cells is lower.

While the observed differences between segmentation methods may not drastically affect large-scale studies with hundreds of thousands of cells, they can have significant implications for smaller-scale analyses, such as studies focusing on the tumour microenvironment and immune infiltration. Misclassification or under-segmentation of infiltrating immune cells, particularly T cells, could lead to erroneous conclusions in these contexts.

Deep learning-based methods, particularly Mesmer and Cellpose, demonstrated superior adaptability and performance compared to traditional algorithms like CellProfiler. This trend aligns with the increasing adoption of deep learning models in the field, as evidenced by platforms such as Steinbock [150] and CellProfiler[156] itself, which now support the integration of pre-trained neural networks. These advancements are critical for addressing the unique challenges posed by IMC datasets, which often involve dense and overlapping cells with low-intensity contrast.

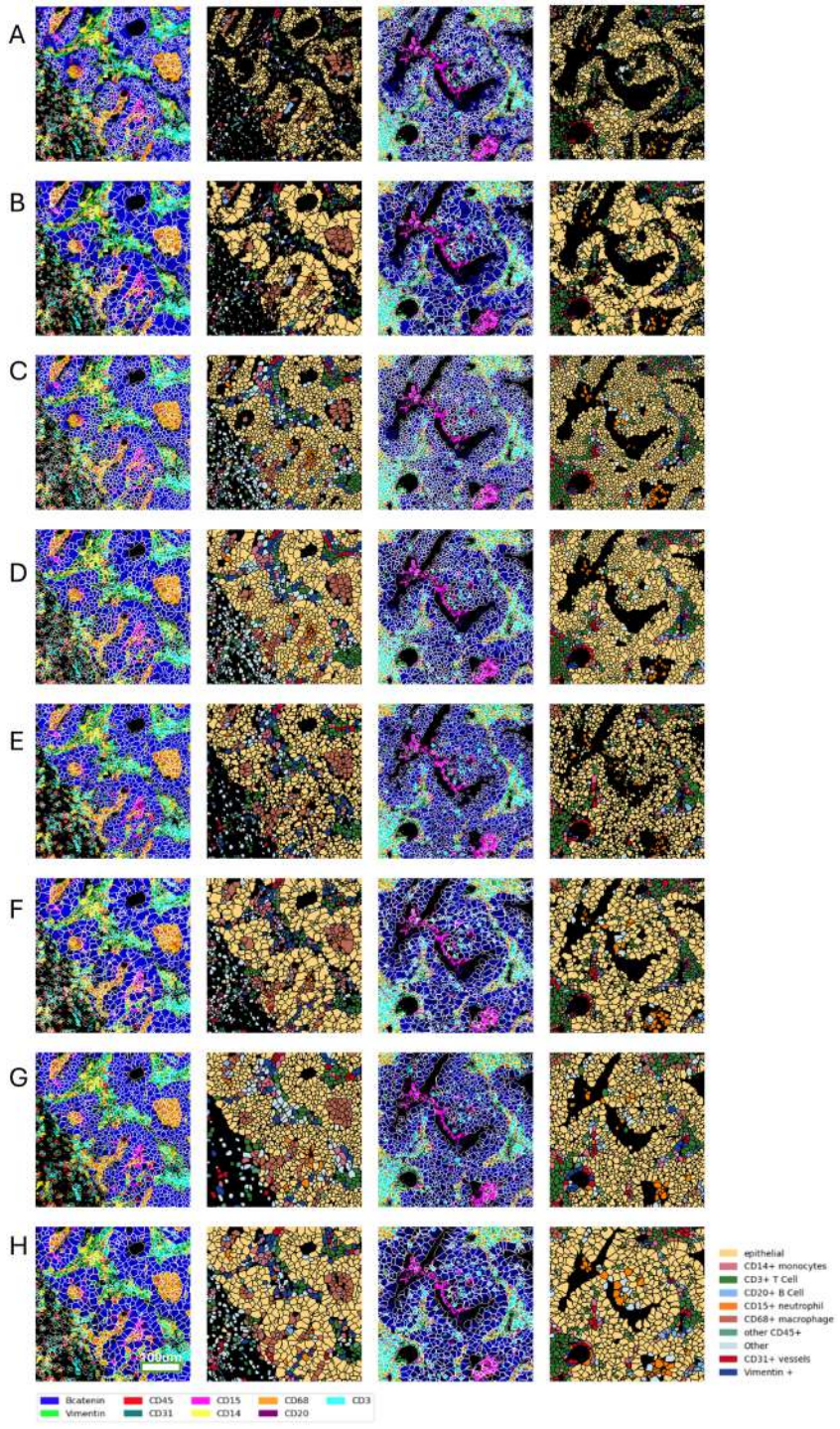


Figure 4.5: Visual comparison of segmentation and phenotyping results across methods. Comparison of $400 \times 400\,\mu$m tissue crops stained with membrane markers (CD3, CD20, CD68,CD14,CD15,CD31,CD45,Vimentin and β-Catenin), showing segmentation borders overlaid on corresponding images coloured by assigned cell phenotype. Results are shown for CellProfiler OneGO (A), CellProfiler Sequential (B), Mesmer (C), Mesmer Sequential (D), Cellpose (E), Cellpose Sequential (F), Cellpose Retrained (G), and Cellpose Retrained Sequential (H).

The importance of high-quality training data for deep learning models cannot be overstated. Both Mesmer and Cellpose [160, 161] have demonstrated that models trained on the same datasets can achieve comparable accuracy, despite their architectural differences. Fine-tuning models with user-annotated ROIs — even with a relatively small number of annotations (500–1000 ROIs) — can significantly improve performance for specific research needs [162] and adaptability of segmentation styles. This highlights the value of curated datasets and the growing role of human-in-the-loop pipelines for annotation and model optimization. In light of this, we manually segmented more than 1900 ROIs spanning several cell types. This annotation defines a cell as a structure containing both a nucleus and an associated membrane. Importantly, cells with irregular shapes were included in our annotations as long as these two features were present. It is worth noting that annotating IMC images is inherently challenging, particularly when dealing with epithelial cells. These cells often appear crowded, with overlapping or irregular shapes, making it difficult to accurately delineate individual cell boundaries. Given these complexities, it is unsurprising that segmentation algorithms may also encounter difficulties in accurately segmenting such dense and intricate cellular structures.

This study does not aim to declare a definitive "best" segmentation algorithm but rather seeks to provide a comprehensive comparison and raise awareness of the potential impacts of segmentation strategies on downstream analyses. The key contributions of this work include: a comparison on cell segmentation algorithms and a strategy to deploy DL models in a sequential fashion and a fully annotated IMC cohort available for public use, this dataset can be leveraged to fine-tune generalist cell segmentation models or to train specialist models, paving the way for more accurate and tailored segmentation approaches.

Our findings underscore the growing importance of deep learning-based segmentation methods in spatial proteomics. As more pre-trained models become available and platforms evolve to facilitate easy integration and fine-tuning, we anticipate these methods will become the standard in the field [267]. Moving forward, the development of new strategies and the availability of diverse, high-quality training datasets will be critical for improving segmentation accuracy. Pre-trained models offer a promising starting point [162], but their effectiveness depends on their ability to adapt to datasets with differing characteristics. Additionally, the integration of tools and the refinement of non-cell-specific models, such as Segment Anything Model (SAM) [258] for cell segmentation [259] highlight exciting directions for future research.

Segmentation in IMC remains an evolving field, with no one-size-fits-all solution. Researchers must consider the specific needs of their studies when choosing segmentation strategies, balancing accuracy, runtime, and the ability to handle diverse tissue contexts. Our study serves to broaden perspectives on segmentation methodologies and provides actionable insights for researchers aiming to optimize their workflows. We hope this work stimulates further exploration and refinement of segmentation techniques in spatial proteomics.

4.4 Methods

All the tests were run in a Intel(R) Core(TM) i7-9700 CPU. The DL models were run using a GPU Nvidia TU102 GeForce RTX 2080 Ti.

4.4.1 Dataset

All the results for IMC were obtained using the publicly available cohort of 61 IMC raw images with 39 markers of colorectal cancer [263].

4.4.2 Preprocessing

The cohort was denoised and normalized using the in-house PENGUIN framework and with the percentiles and thresholds defined in the paper [264].

4.4.3 Annotation dataset

From the full cohort, random patches of 256×256 μm were generated. Composite images of the nucleus marker and different membrane markers for the different mask types were generated, namely: maximum value of DNA markers and the maximum value of Lymphoid markers - CD7, CD20 and CD38; maximum value of DNA markers and the maximum value of myeloid markers – CD68, CD14, CD11c; maximum value of DNA markers and the value of vimentin; maximum value of DNA markers and the maximum value of tumour markers – β-Catenin and Keratin. A total of 25 FOVs, containing 1942 ROIs across the 4 mask types (detailed in Table 4.1) were annotated using the CellposeGUI [162]. This set was further divided into a training set to fine-tune and train models from scratch and a test set to assess the performance of the models. Our annotation approach defines a cell as a structure containing both a nucleus and an associated membrane. This dataset is fully manually annotated, without recurring to the human-in-the-loop strategy offered by Cellpose, so to keep the fully annotated set unbiased.

4.4.4 Fine-tuning of Cellpose models

Models from the Cellpose family were fine-tuned using the training annotation set. The generalist model "cyto3" was retrained for 500 epochs, batch size 8, learning rate 0.1, weight decay 0.0001 and SGD optimizer. In the same way, specialist models. "TissueNet_cp3", trained using TissueNet dataset, and "TN1", trained using the TN1 partition of tissueNet, were fine-tuned using the same parameters. Finally, a model from scratch, i.e., using the Cellpose architecture and not using any pretrained weights, was derived using the same parameters. FOVs from the test set were predicted using the generalist "cyto3", "finetuned cyto3", "TissueNet_cp3", "finetuned TissueNet_cp3", "TN1", "finetuned TN1" and the model trained from scratch, with a flow threshold of 1 and a cell probability threshold of -2. The quality of the annotations predicted on the test set was assessed both visually and through segmentation evaluation metrics, defined below, namely IoU, Dice coefficient, accuracy, precision, recall, and F1 score. Due to the

very high number of true negative pixels compared to true positive pixels, the MCC and balanced accuracy were also included.

4.4.5 Comparison of segmentation models on test set

Segmentations of individual cell types were also evaluated against the test set. Lymphoid, Myeloid, Vimentin and tumour composites were segmented using CellProfiler, Mesmer and Cellpose.

Mesmer allows to tweak parameters that control the outcome of the segmentation, namely:

- the interior threshold, controls how conservative the model is in estimating what is a cell vs what is background. Lower values will result in larger cells, whereas higher values will result in smaller cells;
- maxima threshold: controls what the model considers a unique cell. Lower values will result in more separate cells being predicted, whereas higher values will result in fewer cells.

All the segmentations used the whole cell segmentation model and the resolution changed for 1 (resolution of IMC). For lymphoid masks, maxima, interior and radius were set as the default, 0.075, 0.2 and 2 respectively and a radius of 2. For myeloid and vimentin masks, to allow for bigger cells, interior of 0.1, maxima of 0.2 and radius of 4 were used. Finally, for tumour masks interior was 0.1, maxima 0.2 and radius 6. These parameters were chosen after manual tweaking parameters in training set to achieve the best concordant segmentation. The Mesmer model segments cells even without membrane, so to derive the segmentation cells from only that mask, objects were filtered based on the presence of membrane marker, to exclude cells only with nucleus signal (Figure SS4.3). Lymphoid, myeloid and vimentin objects were considered if they had at least 40% of area covered with pixels with intensity higher than 0.6. Tumour objects were considered if at least 30% of area was covered with pixels with intensities higher than 0.5. For the Cellpose TN1 model and the TN1 retrained model, the parameters are as follows: a flow threshold of 0.6, cell probability of 0 for lymphoid, myeloid and vimentin masks; a flow threshold of 1.1 and cell probability of -2 was used for tumour masks, the expected diameter was also changed from 10 to 15. Cellpose models do not segment nucleus without membranes, but may sometimes segment membranes without nucleus. To account for that, only objects with at least a sum of intensity of 4, for the DNA channel, were considered.

The quality of the annotations predicted on the test set was assessed with pixel evaluation and ROI metrics (check details in the segmentation metrics section).

Cellpose was versioned at 3.1.0, deepcell 0.12.6 and cellProfiler 4.2.6.

4.4.6 Segmentation metrics

To assess the quality of the predicted segmentations compared to the annotation segmentation, several metrics were used. Considering TP as True Positives, FP as False Positives, FN as False negatives, and TN as True negatives, for predicted pixels, the following metrics are defined below in Table 4.2:

To assess segmentation performance at the ROI level, we compared annotated objects to the predicted objects. Initially, the IoU was calculated for each predicted object against every annotated object. However, predictions and annotations showed little overlap as objects appear often under or over segmented or with irregular shapes. To address this issue and visualize the ROIs more effectively, we replaced the IoU metric with a 0.2 threshold for mask overlap. Specifically, if a predicted object had more than 20% of its area overlapping with positive annotated pixels, it was counted as a TP. If the overlap was less than 20%, the predicted object was considered a FP. Additionally, annotated objects that were not overlapped by predicted pixels were marked as FN. Images reflecting this information were generated for better clarity.

4.4.7 Derivation of OneGo and Sequential final segmentations

For each FOV, several composite images were created by combining nucleus and membrane markers to serve as input for the segmentation models. The nucleus component was always constructed in the same way, using the DNA markers. The membrane component, however, varied depending on the composite type: a lymphoid composite used the maximum intensity value of the CD7, CD20, and CD38 markers; a myeloid composite used the maximum value of the CD68, CD14, and CD11c markers; the vimentin membrane was represented solely by the Vimentin marker; and the tumor composite used the maximum value of the β-Catenin and Keratin markers for the membrane. The lymphoid, myeloid, vimentin, and tumor masks were used for sequential segmentations. In contrast, for the One GO segmentation, the DNA marker was combined with a membrane composed of the average value of the Vimentin, β-Catenin, and Keratin markers.

CellProfiler OneGO was run by identifying primary objects (nuclei) using minimum-cross-entropy thresholding, correction factor of 1.2, typical diameter of 10-20 pixels and cell declumping by shape. Nuclei with a size < 5 pixels were removed and secondary objects (cell membranes) were identified by propagation. For CellProfiler-sequential primary objects were identified using the same settings as OneGo, but secondary objects were identified in the following order: lymphocytes, myeloid cells and stromal cells. Between each step, the identified cells were removed from the images. For cancer cells, the remaining nuclei were re-segmented using a typical diameter of 10-50 pixels after which membranes were added.

Mesmer OneGO was run with default parameters and a resolution of 1, using the appropriate composite image. For sequential segmentation with Mesmer, the lymphoid composite was processed first, and the results were filtered as described earlier. Next, the myeloid composite was applied, with the nuclei and membranes already assigned in the lymphoid mask being filtered out. The updated composite was then segmented and filtered using the same parameters. This process was repeated sequentially for the vimentin and tumor composites. Finally, any remaining unassigned pixels from the nucleus markers were segmented using the nucleus model without pixel expansion. To generate a non-overlapping composite, the lymphoid, myeloid, vimentin, tumor, and remaining segments were merged in order of priority.

For Cellpose, both the TN1 and TN1-retrained models were tested. In the One GO approach, cells were considered with a flow threshold of 1.1 and a cell probability threshold of -2. Cellpose sequential

segmentation followed a similar pipeline to Mesmer: after segmenting each composite, cells not belonging to the current mask were filtered out, and the nucleus and membrane pixels already assigned were removed from the next composite. This process was repeated for each composite. The nuclei model in Cellpose was used to segment the remaining unassigned pixels. The composite images for Cellpose were generated in the same way as for Mesmer and CellProfiler.

4.4.8 Preprocessing of cell data

From the 8 final segmentations masks, morphology parameters – area, equivalent diameter, diameter and centroids – and intensity information were retrieved per cell. Using Scanpy [242] a PCA with 20 components and an UMAP (using 10 neighbors) were derived. Cells with area inferior of 10 pixels were annotated has low quality.

4.4.9 Phenotyping

Due to the number of sets to be compared and to eliminate any potential bias from manual gating, a phenotyping approach based on a decision tree was used (Supp. Figure 8). For each cell, if either β-Catenin or keratin are higher than any of non-tumour markers (Non epithelial markers: CD45, CD45RO, CD3, CD20, CD68, CD14, CD15, Vimentin, CD11b, CD11c), the cell is considered epithelial. Non epithelial cells are classified further, and in a sequential manner if markers passed threshold, cells are classified in CD3+ T cell, CD20+ B cell, CD68+ macrophage, CD14+ monocytes, CD15+ neutrophil, CD31+ vessels, other CD45+ cells, Vimentin+ and the remaining labelled as Other. The threshold applied to all was the value of 0.2 expression of that marker in that cell. The distribution of cell types and area occupied per each cell type per segmentation was assessed. Furthermore, were also derived dotplots of the expression of markers per cell type.

4.4.10 Downstream analysis

To check eventual differences in immune infiltration, Voronoi diagrams were derived and the 1^{st}-hop neighbourhoods of cells analyzed. For each CD3 T cell, we calculated the number of cells with more than 50% epithelial neighborhoods, classified as intraepithelial T cell.

Finally, 400×400 μm crops of distinct tissue sections, showcasing marker overlays with segmentation boundaries and the corresponding phenotyped images were produced.

4.5 Data and code availability

The IMC data used in this study is publicly available from Roelands et al. [263] and can be accessed through the BioImage Archive (Accession number S-BIAD587, `https://www.ebi.ac.uk/biostudies/bioimages/studies/S-BIAD587`). The segmentation code is fully available at `https://github.com/deMirandaLab/CellCytoLego`

4.6 Supplementary information

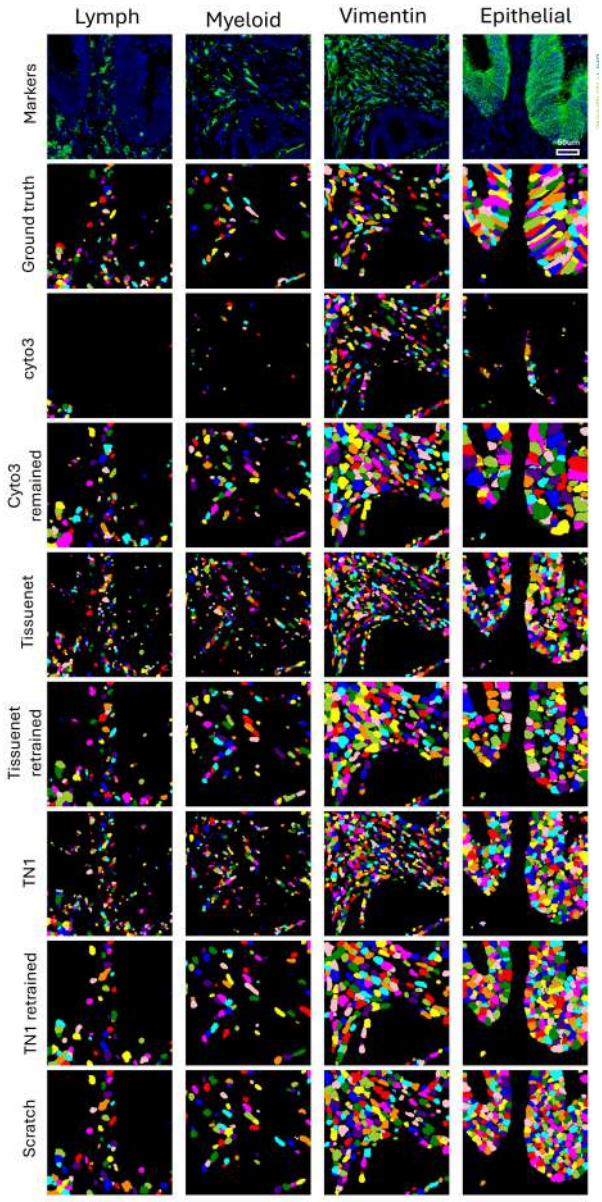


Figure S4.1: Comparison of ground truth masks and masks generated by Cellpose models. Composite test image with nucleus and membrane markers for lymphoid, Myeloid, vimentin and epithelial cells with ground truth mask and segmentation masks obtained using Cellpose generalist model, Cellpose trained on TissueNet, Cellpose trained on TN1 section of TissueNet and the masks obtained by retraining each of these masks by 500 epochs. The final row is the masks obtained by running Cellpose trained from scratch using just the dataset provided.

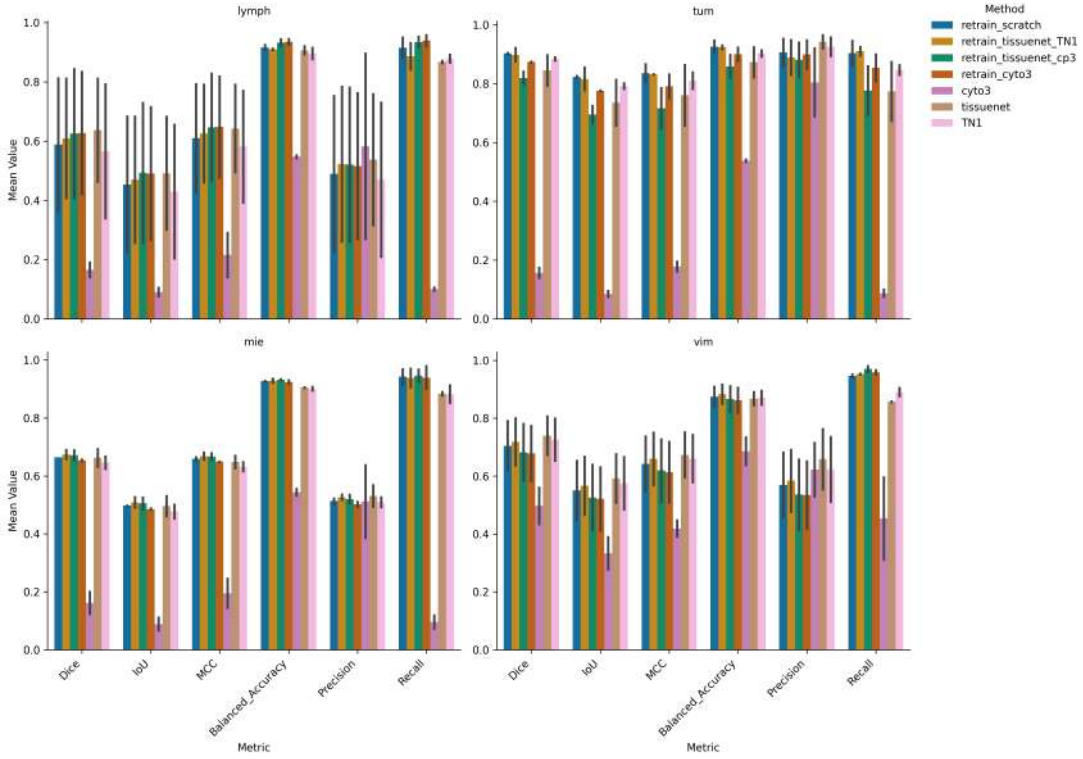


Figure S4.2: Pixel Evaluation metrics of segmentation generated by Cellpose models. Bar plot of Pixel evaluation metrics, Dice coefficient, IoU, MCC, balanced accuracy, precision and recall, for test set of lymphoid, myeloid, tumour and vimentin masks using Cellpose models and their correspondent retrained versions, namely, Cellpose generalist cyto3 model, Cellpose model specialist trained on TissueNet, Cellpose specialist model trained on the subset TN1 from TissueNet and a Cellpose trained from scratch.

Table 4.2: Segmentation evaluation metrics with descriptions and formulas.

Description	**Formula**
Intersection over Union (IoU): Measures overlap between prediction and ground truth regions. Ranges from 0 (no overlap) to 1 (perfect overlap).	$\text{IoU} = \frac{\lvert A \cap B \rvert}{\lvert A \cup B \rvert} = \frac{TP}{TP + FP + FN}$
Dice Coefficient: Emphasizes overlap, useful for imbalanced datasets. Harmonic mean of precision and recall.	$\text{Dice} = \frac{2\lvert A \cap B \rvert}{\lvert A \rvert + \lvert B \rvert} = \frac{2TP}{2TP + FP + FN}$
Accuracy: Overall proportion of correctly classified pixels. Can be misleading for imbalanced data.	$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$
Precision: Proportion of predicted positives that are true. Indicates model reliability in positive predictions.	$\text{Precision} = \frac{TP}{TP + FP}$
Recall (Sensitivity): Proportion of actual positives that were correctly predicted.	$\text{Recall} = \frac{TP}{TP + FN}$
F1 Score: Harmonic mean of precision and recall. Good for datasets with uneven class distribution.	$F_1 = \frac{2 \cdot \text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}$
Matthews Correlation Coefficient (MCC): Balanced metric that considers all four confusion matrix categories. Ranges from -1 (total disagreement) to 1 (perfect prediction).	$\text{MCC} = \frac{TP \cdot TN - FP \cdot FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}}$
Balanced Accuracy (BA): Average of sensitivity and specificity. More appropriate than plain accuracy for class-imbalanced problems.	$\text{BA} = \frac{1}{2}\left(\frac{TP}{TP + FN} + \frac{TN}{TN + FP}\right)$
Adjusted Rand Index (ARI): Measures similarity between segmentations, adjusting for chance grouping. Ranges from -1 to 1.	*Implemented as [47]*

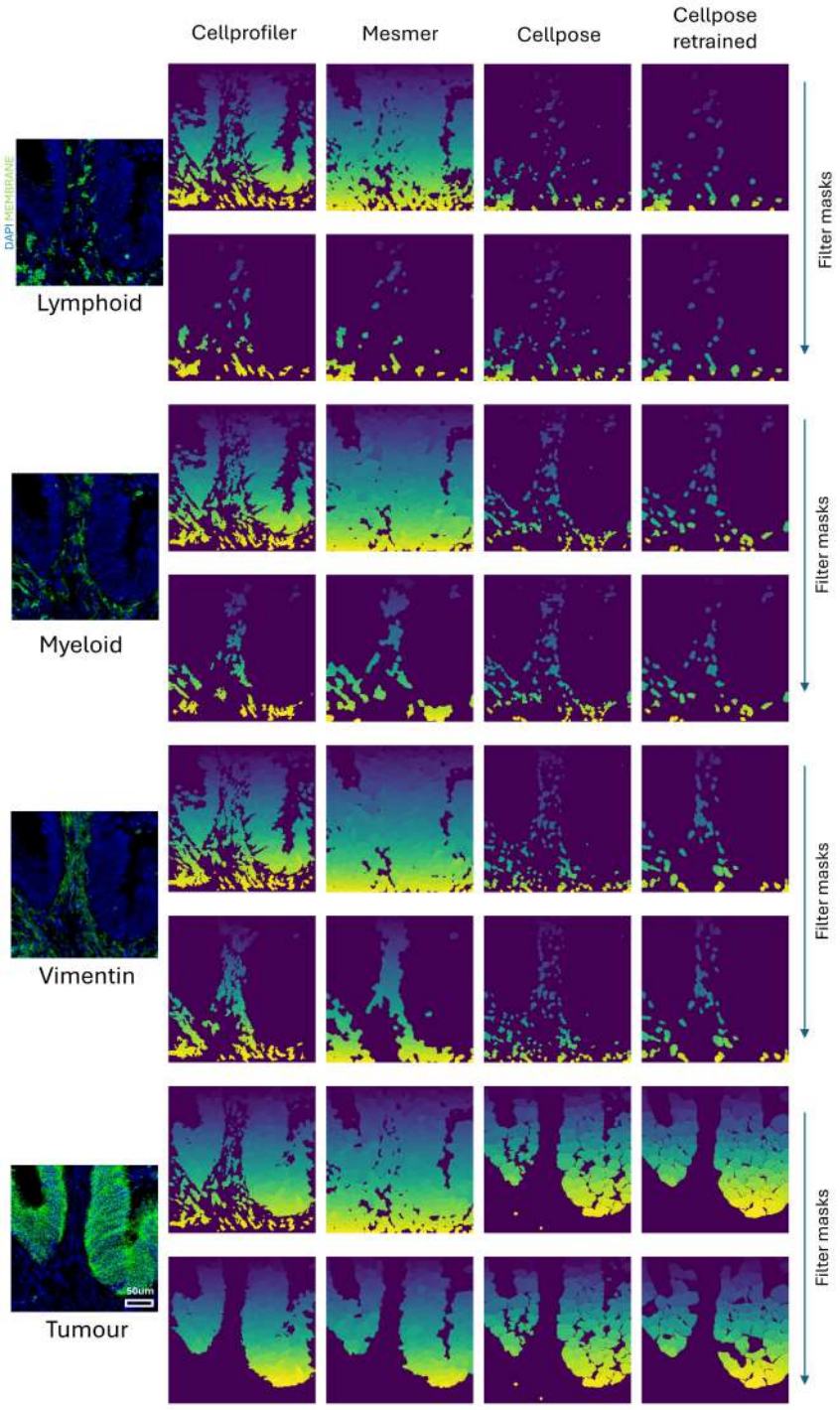


Figure S4.3: Segmentation mask filtering steps. Example of filtering lymphoid, myeloid, vimentin and tumour masks. For each of the segmentation methods, produced segmentations are filtered in order to keep only the segmented cells with high percentage of membrane markers specific for that cell type (methods). This step is important as models such as CellProfiler and Mesmer segment an object as a cell even without a membrane marker, Cellpose, on the contrary, follows the same segmentation style and only objects with nucleus and membrane are segmented.

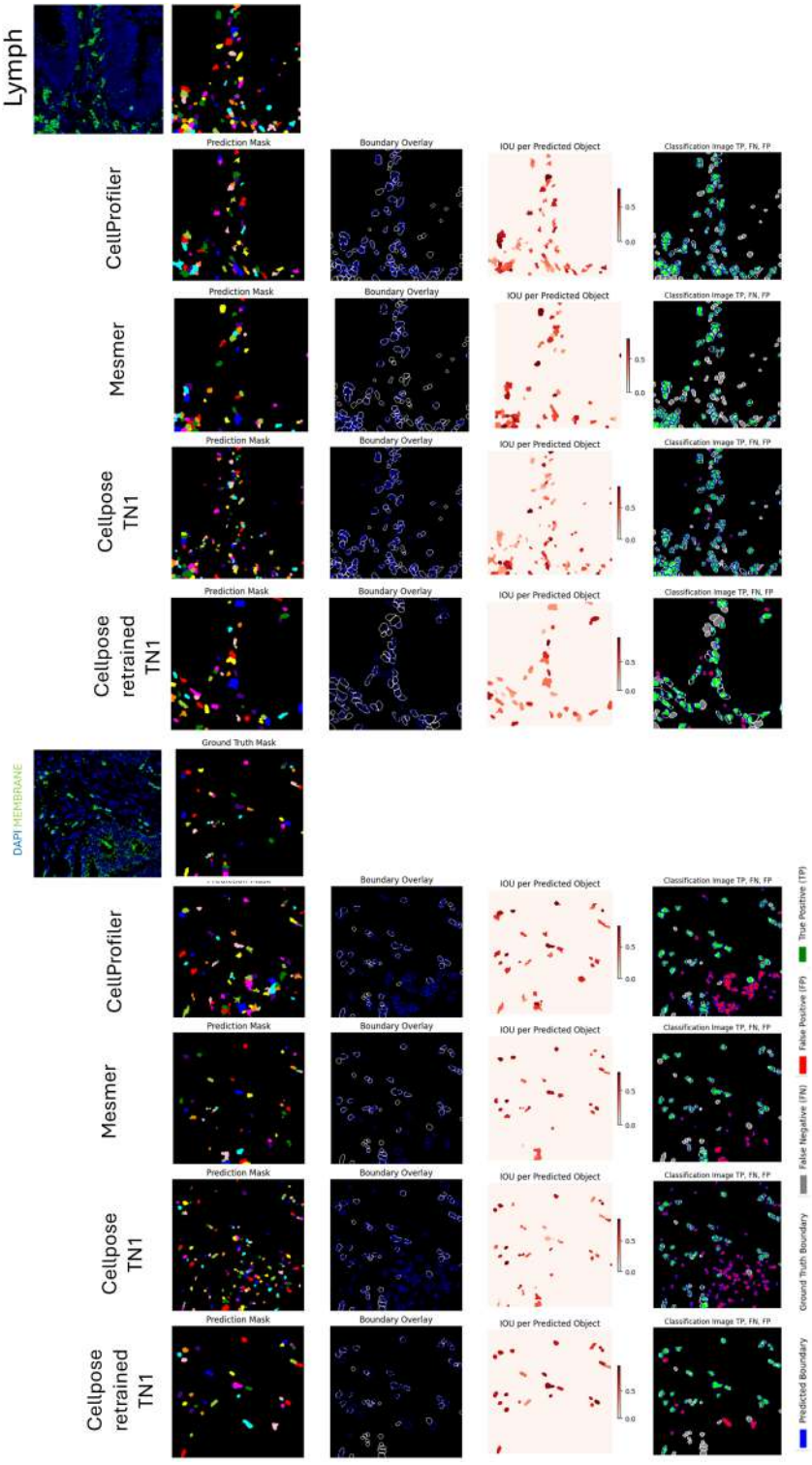


Figure S4.4: Evaluation of the segmentation methods at ROI level on **lymphoid**. Evaluation of the segmentation methods, CellProfiler, Mesmer, Cellpose and Celpose retrained, in the test set for lymphoid masks. From left to right: Ground and prediction masks; overlap of the segmentation boundaries both annotated (white) and predicted (blue); IoU per each predicted ROI; classification of each ROIS as True Positive (green), False positives (red) and False negatives (grey). Due to the low overlap of annotated objects with the predicted ones, instead of the IoU score between ROIs in the mask, an 0.2 threshold of overlapping between masks was used.

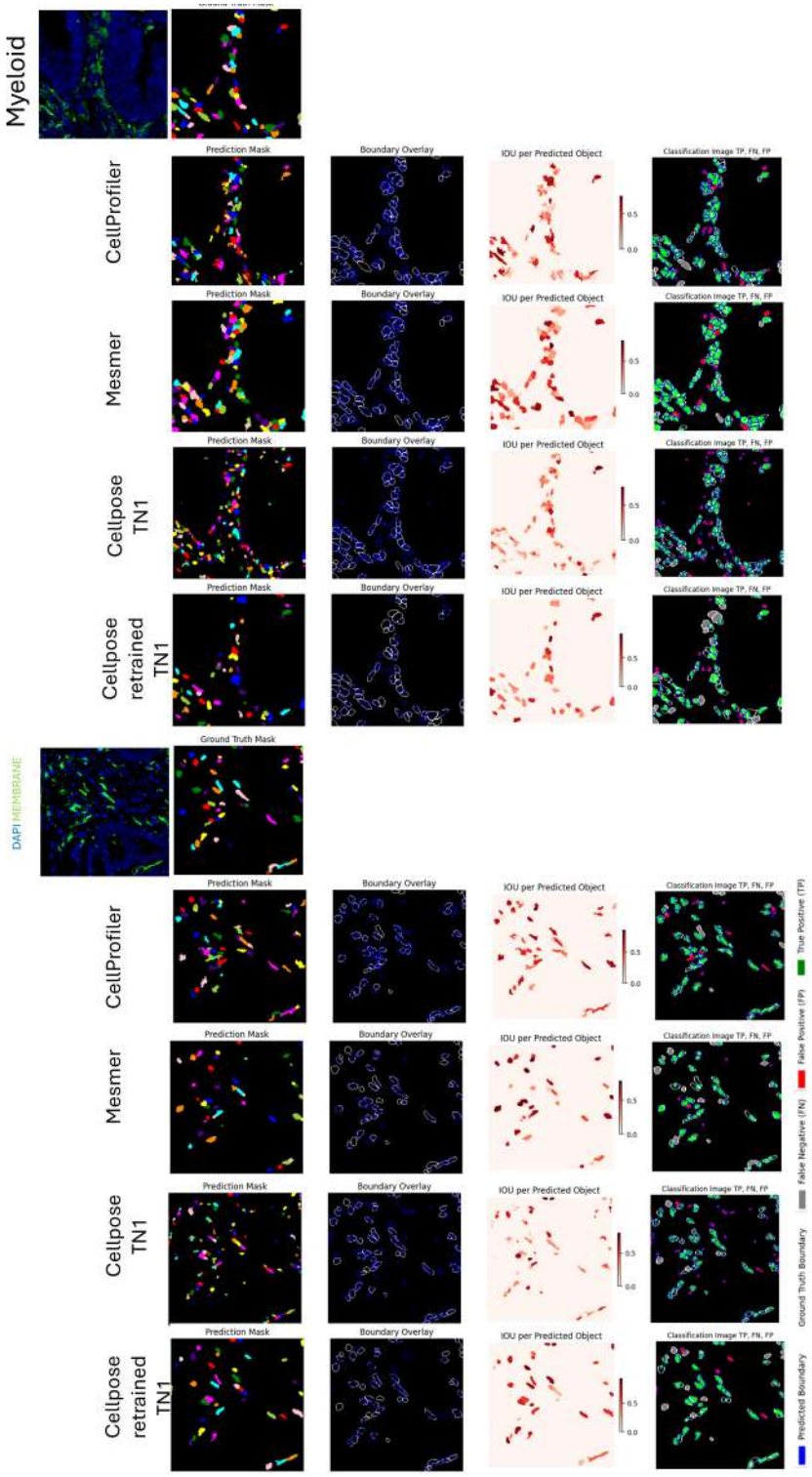


Figure S4.5: Evaluation of the segmentation methods at ROI level on **myeloid**. Evaluation of the segmentation methods, CellProfiler, Mesmer, Cellpose and Celpose retrained, in the test set for myeloid masks. From left to right: Ground and prediction masks; overlap of the segmentation boundaries both annotated (white) and predicted (blue); IoU per each predicted ROI; classification of each ROIS as True Positive (green), False positives (red) and False negatives (grey). Due to the low overlap of annotated objects with the predicted ones, instead of the IoU score between ROIs in the mask, an 0.2 threshold of overlapping between masks was used.

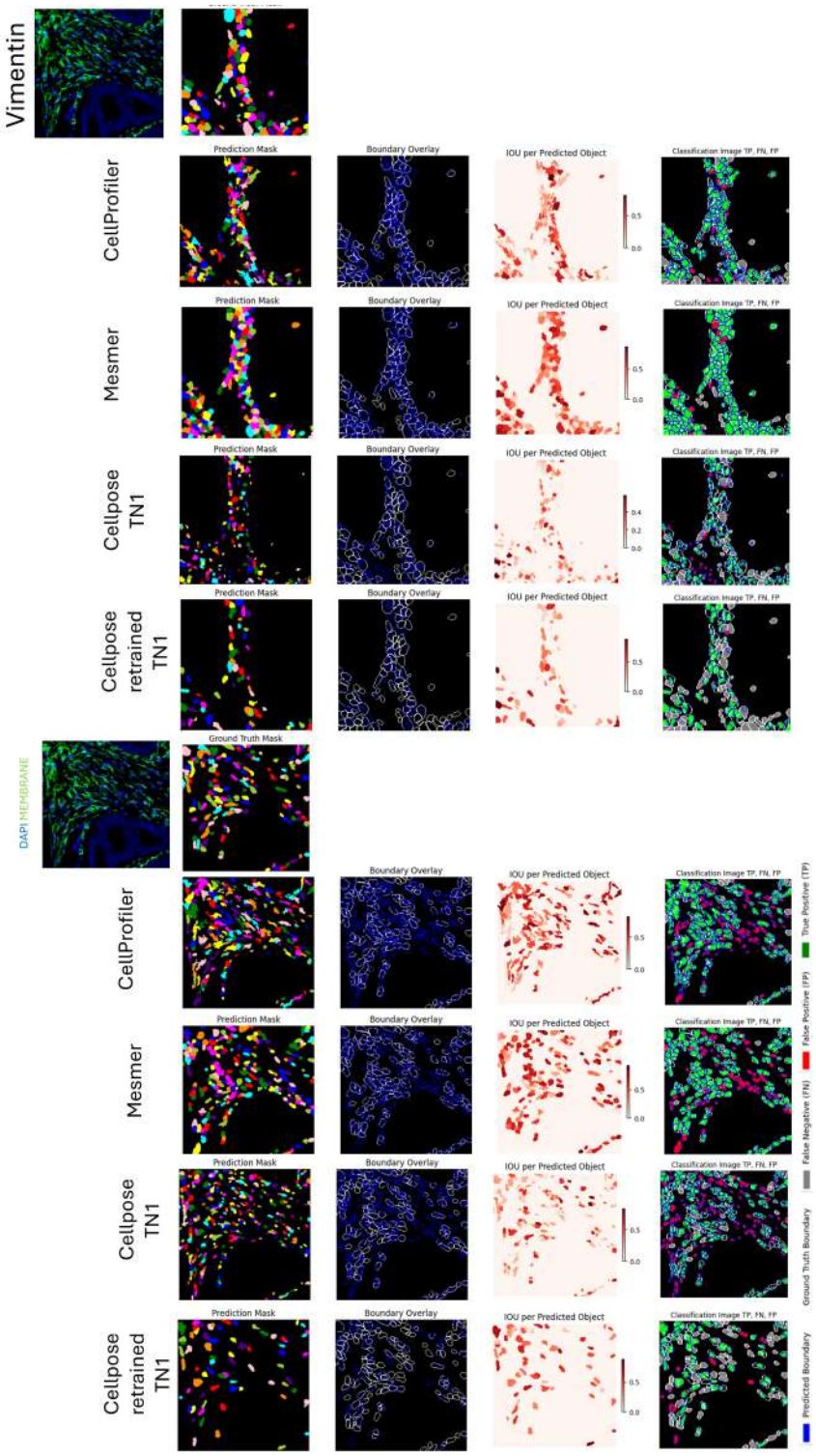


Figure S4.6: Evaluation of the segmentation methods at ROI level on **Vimentin+**. Evaluation of the segmentation methods, CellProfiler, Mesmer, Cellpose and Celpose retrained, in the test set for Vimentin+ masks. From left to right: Ground and prediction masks; overlap of the segmentation boundaries both annotated (white) and predicted (blue); IoU per each predicted ROI; classification of each ROIS as True Positive (green), False positives (red) and False negatives (grey). Due to the low overlap of annotated objects with the predicted ones, instead of the IoU score between ROIs in the mask, an 0.2 threshold of overlapping between masks was used.

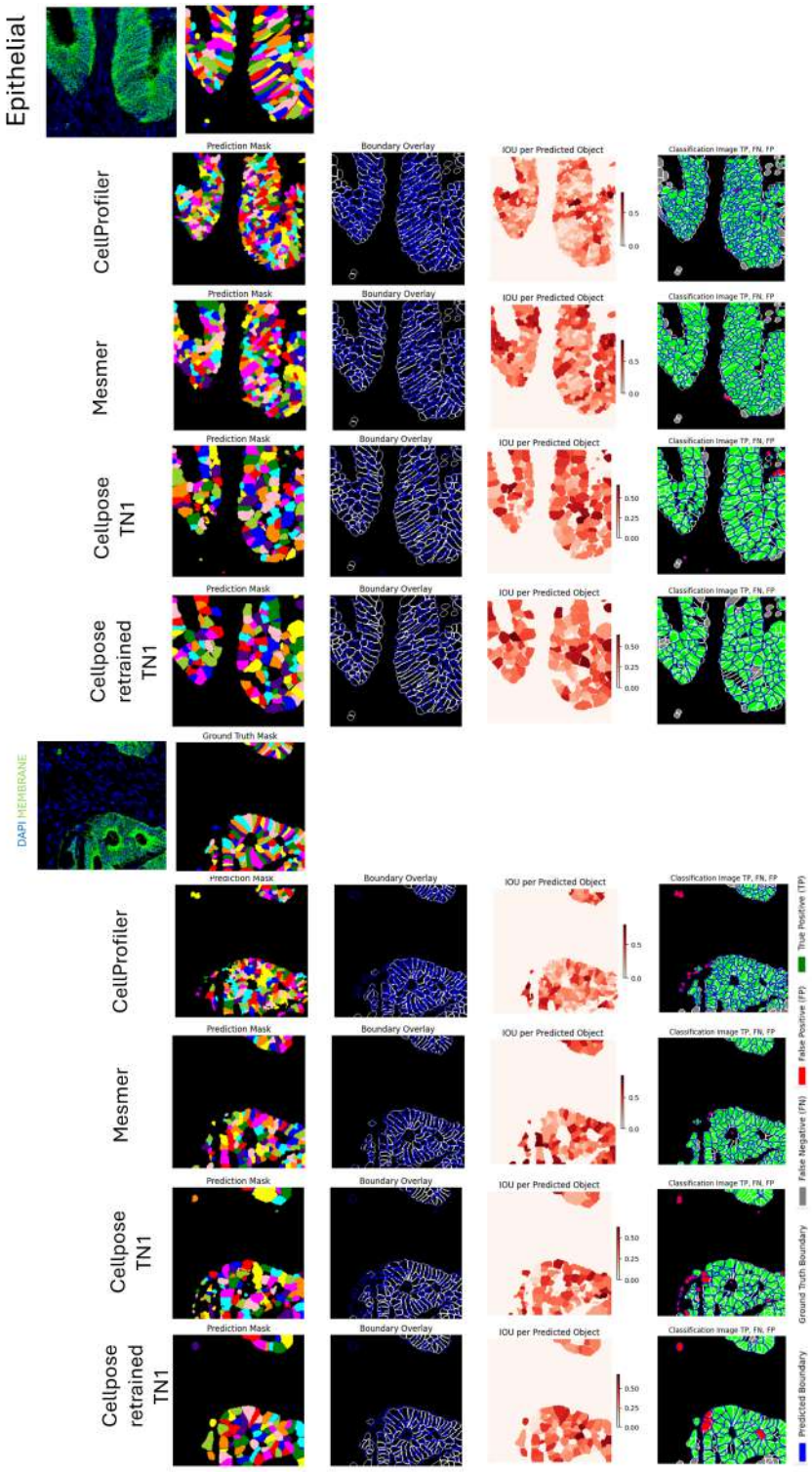


Figure S4.7: Evaluation of the segmentation methods at ROI level on **tumour**. Evaluation of the segmentation methods, CellProfiler, Mesmer, Cellpose and Celpose retrained, in the test set for tumour masks. From left to right: Ground and prediction masks; overlap of the segmentation boundaries both annotated (white) and predicted (blue); IoU per each predicted ROI; classification of each ROIS as True Positive (green), False positives (red) and False negatives (grey). Due to the low overlap of annotated objects with the predicted ones, instead of the IoU score between ROIs in the mask, an 0.2 threshold of overlapping between masks was used.

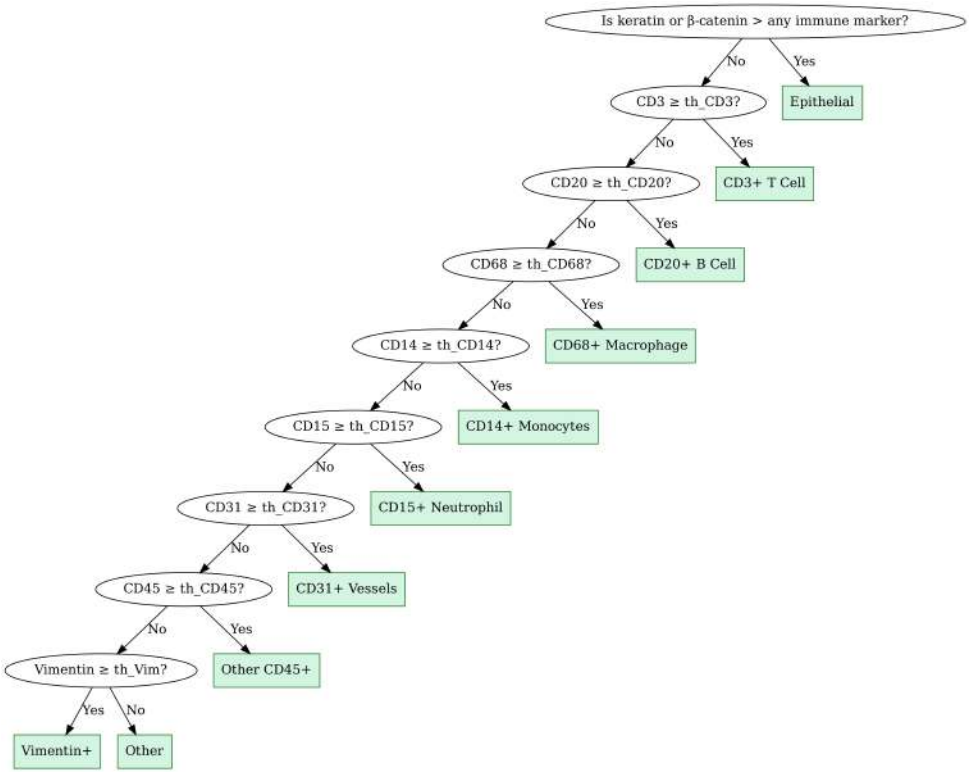


Figure S4.8: Schematics of the proposed decision tree method used for phenotyping. If either β-Catenin or keratin are higher than any of non-tumour markers, the cell is considered epithelial. Non epithelial cells are classified further, and in a sequential manner if markers passed threshold, into CD3+ T cell, CD20+ B cell, CD68+ macrophage, CD14+ monocytes, CD15+ neutrophil, CD31+ vessels, other CD45+ cells, Vimentin+ and the remaining labelled as Other. Schematic produced using graphviz software [268].

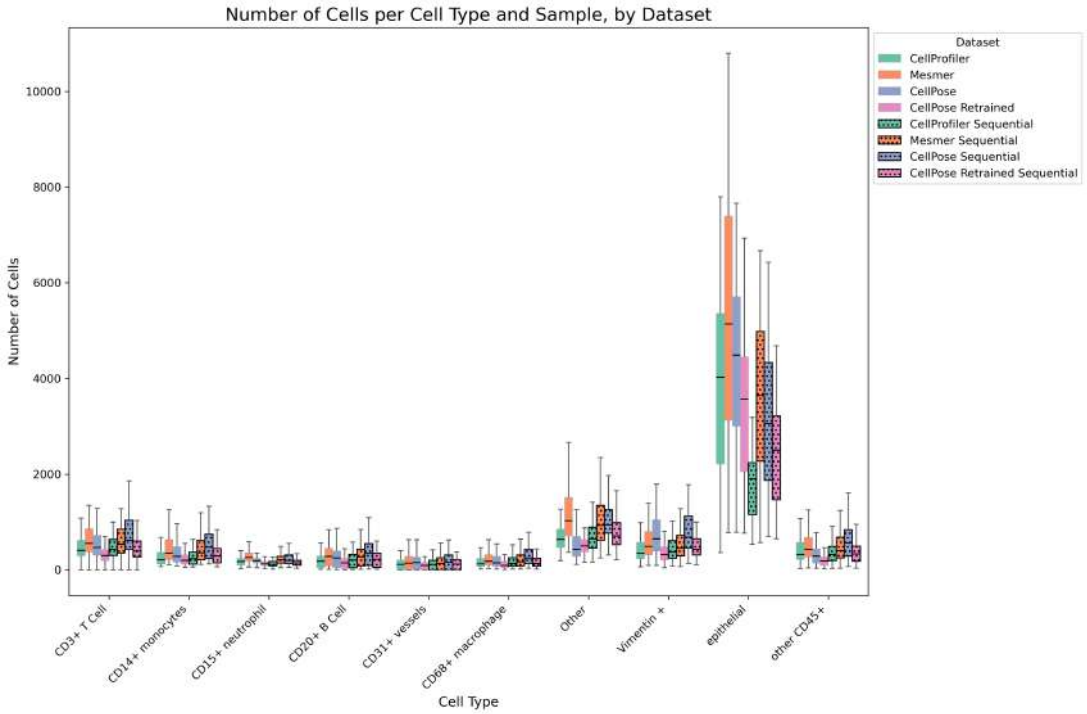


Figure S4.9: Distribution of cell counts per sample across different segmentation methods and cell types. Each boxplot shows the median, interquartile range, and whiskers (1.5x IQR) for cell counts obtained using the indicated segmentation method (color/hatch) for a specific cell type (x-axis).

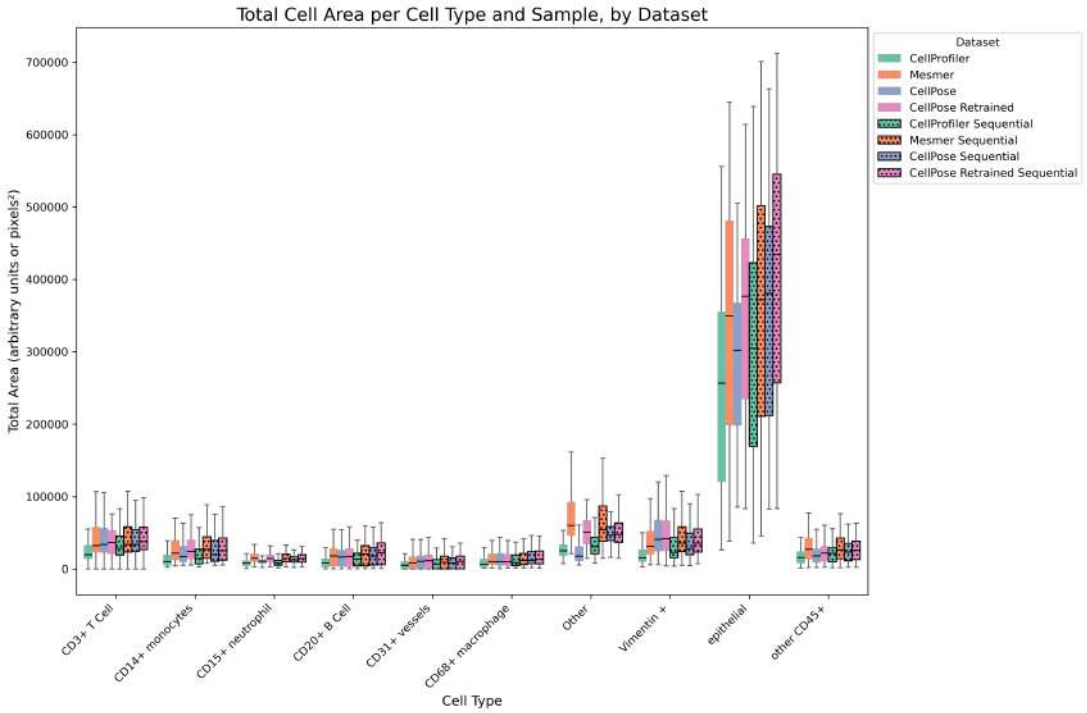


Figure S4.10: Distribution of cell area covered per sample across different segmentation methods and cell types. Each boxplot shows the median, interquartile range, and whiskers (1.5x IQR) for cell counts obtained using the indicated segmentation method (color/hatch) for a specific cell type (x-axis).

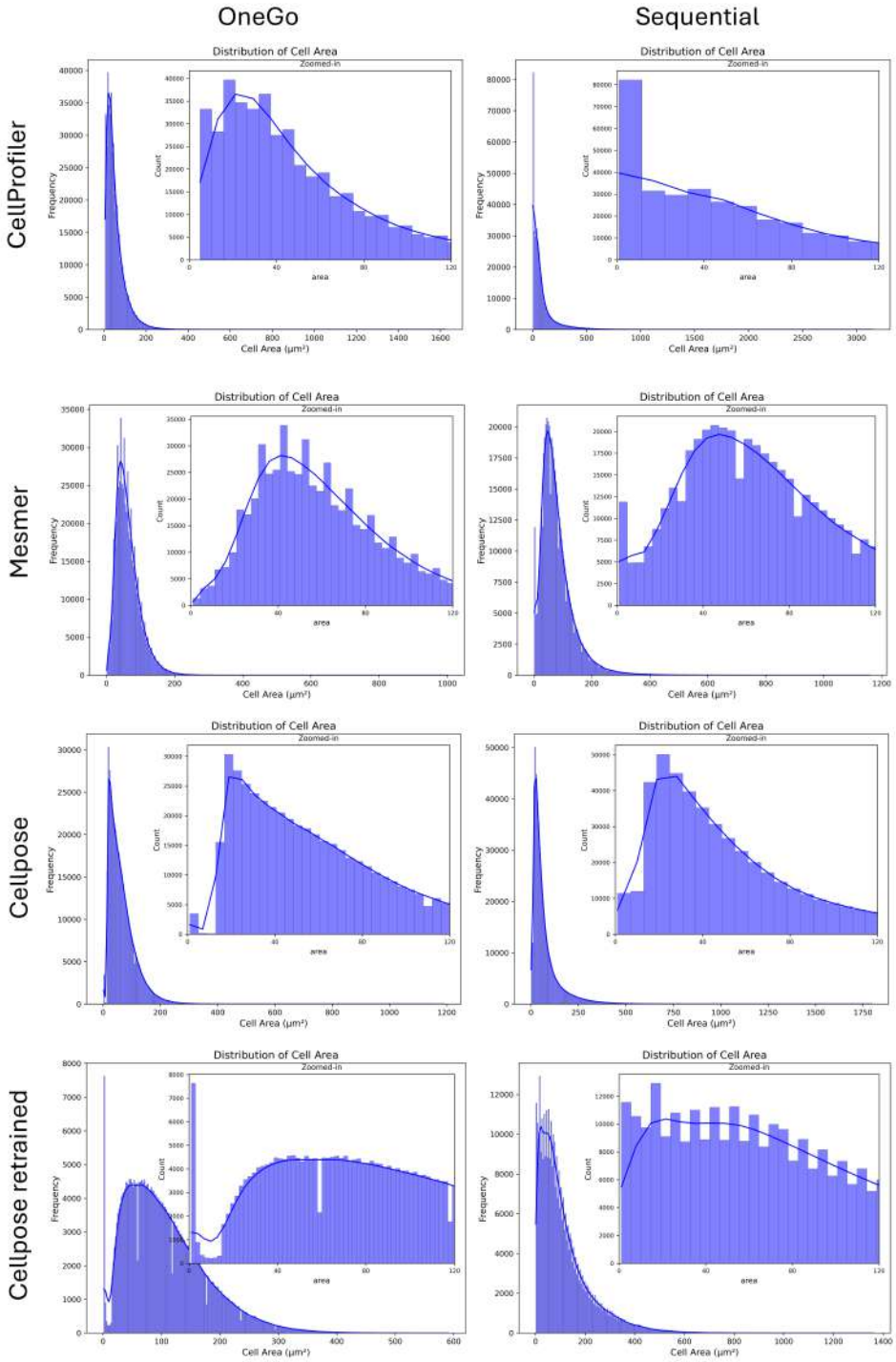


Figure S4.11: Distribution of area per cell across all cells in CellProfiler, Mesmer, Cellpose, and Cellpose retrained, both OneGO and run in a sequential fashion. Each graphic depicts the full distribution with a zoom-in graphic on top for better visualization, showing cell areas from 0 to 120 μm.

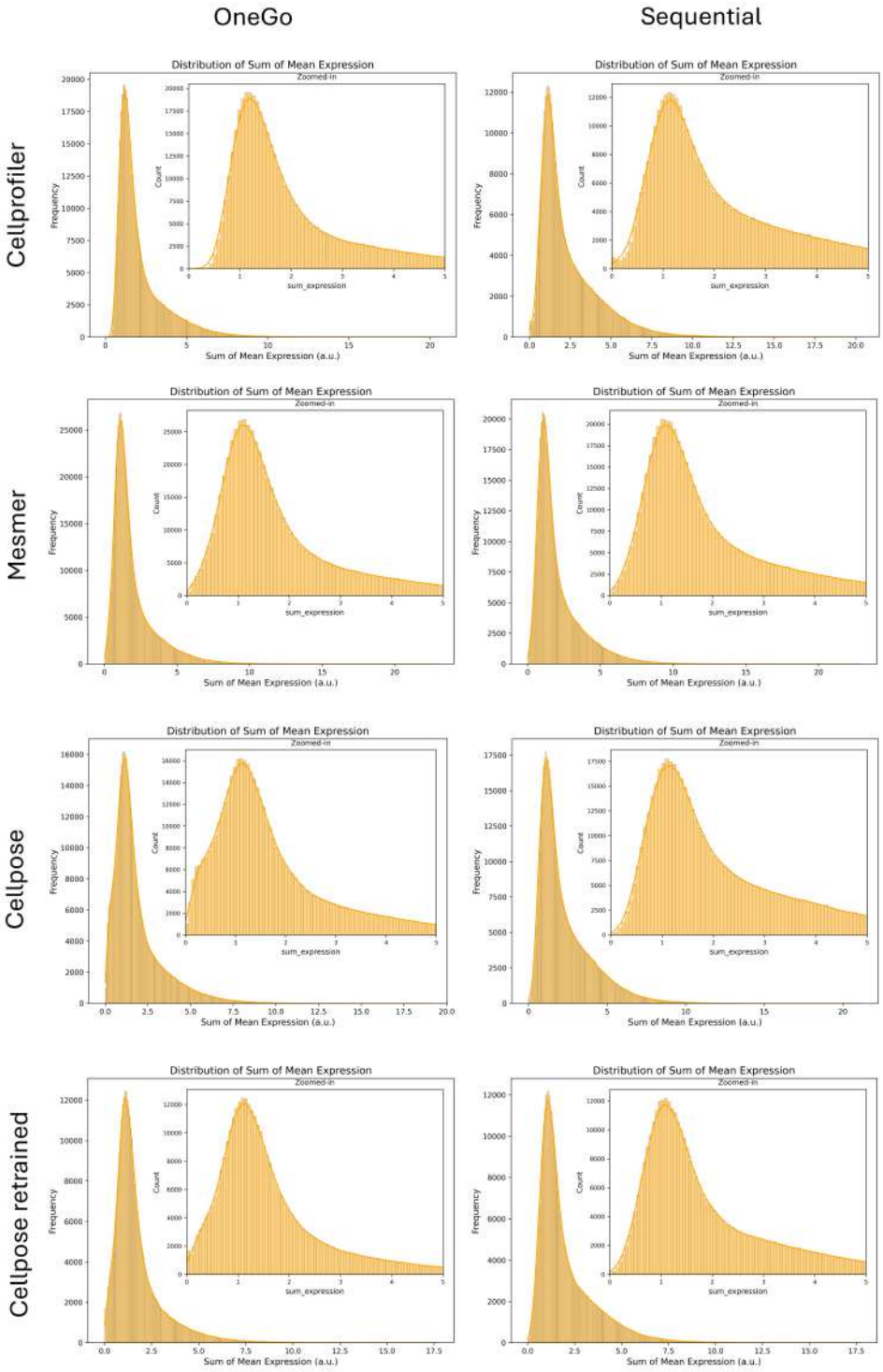


Figure S4.12: Distribution of sum of intensity across all cells in different segmentation methods, showing a zoom-in on the sum of the mean expression of all markers from 0 to 5.

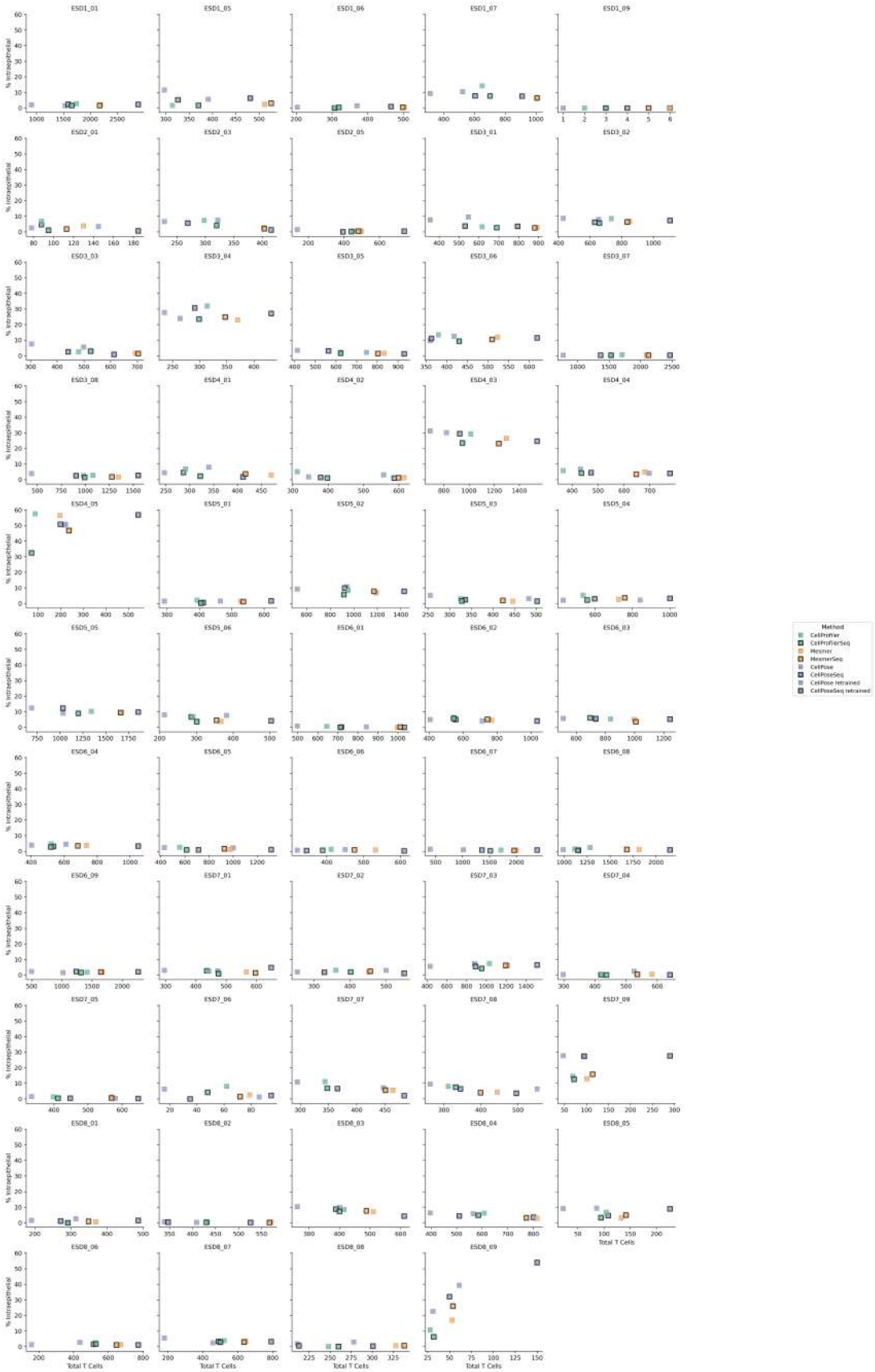


Figure S4.13: Number of total T cells per percentage of intraepithelial T cells across all segmentation strategies and FOVs.

Table S4.1: Pixel evaluation metrics on test set for Cellpose models. Pixel evaluation metrics, IoU, Dice coefficient, precision, recall, F1 score, MCC and balanced accuracy, for test set of lymphoid, myeloid, tumour and vimentin masks using Cellpose models and their correspondent retrained versions, namely, Cellpose generalist cyto3 model, Cellpose model specialist trained on TissueNet, Cellpose specialist model trained on the subset TN1 from TissueNet and a Cellpose model trained from scratch.

Model	Dice	IoU	MCC	Precision	Recall	Balanced Acc.	ARI	Accuracy
Lymphoid								
cyto3	0.166 ± 0.034	0.09 ± 0.02	0.216 ± 0.105	0.583 ± 0.442	0.101 ± 0.007	0.547 ± 0.007	0.136 ± 0.026	0.931 ± 0.032
cyto3 R.	0.627 ± 0.292	0.491 ± 0.317	0.648 ± 0.241	0.516 ± 0.348	0.939 ± 0.025	0.935 ± 0.013	0.556 ± 0.311	0.93 ± 0.047
TissueNet	0.637 ± 0.246	0.492 ± 0.269	0.642 ± 0.209	0.537 ± 0.312	0.867 ± 0.005	0.907 ± 0.019	0.569 ± 0.252	0.94 ± 0.027
TissueNet R.	0.626 ± 0.307	0.493 ± 0.334	0.646 ± 0.256	0.521 ± 0.369	0.934 ± 0.026	0.932 ± 0.016	0.555 ± 0.331	0.929 ± 0.053
TN1	0.566 ± 0.32	0.43 ± 0.319	0.581 ± 0.267	0.47 ± 0.368	0.878 ± 0.019	0.896 ± 0.027	0.484 ± 0.338	0.909 ± 0.064
TN1 R.	0.61 ± 0.285	0.469 ± 0.302	0.625 ± 0.233	0.522 ± 0.369	0.886 ± 0.063	0.91 ± 0.003	0.536 ± 0.304	0.927 ± 0.047
scratch	0.589 ± 0.315	0.453 ± 0.325	0.61 ± 0.257	0.489 ± 0.372	0.915 ± 0.048	0.916 ± 0.011	0.509 ± 0.338	0.915 ± 0.063
Myeloid								
cyto3	0.162 ± 0.054	0.088 ± 0.032	0.195 ± 0.071	0.512 ± 0.177	0.096 ± 0.032	0.544 ± 0.016	0.124 ± 0.04	0.915 ± 0.022
cyto3 R.	0.654 ± 0.004	0.485 ± 0.004	0.649 ± 0.001	0.502 ± 0.012	0.939 ± 0.056	0.924 ± 0.008	0.54 ± 0.042	0.913 ± 0.031
TissueNet	0.662 ± 0.043	0.496 ± 0.048	0.648 ± 0.03	0.531 ± 0.052	0.884 ± 0.009	0.905 ± 0.001	0.562 ± 0.02	0.924 ± 0.011
TissueNet R.	0.671 ± 0.025	0.505 ± 0.028	0.667 ± 0.016	0.52 ± 0.021	0.946 ± 0.03	0.932 ± 0.003	0.565 ± 0.009	0.921 ± 0.019
TN1	0.646 ± 0.031	0.477 ± 0.033	0.632 ± 0.022	0.509 ± 0.024	0.882 ± 0.043	0.901 ± 0.009	0.538 ± 0.003	0.917 ± 0.02
TN1 R.	0.674 ± 0.022	0.509 ± 0.025	0.668 ± 0.017	0.526 ± 0.013	0.938 ± 0.045	0.929 ± 0.009	0.57 ± 0.013	0.922 ± 0.021
scratch	0.664 ± 0.001	0.497 ± 0.001	0.659 ± 0.007	0.513 ± 0.012	0.942 ± 0.037	0.928 ± 0	0.554 ± 0.045	0.917 ± 0.03
Tumour								
cyto3	0.157 ± 0.024	0.085 ± 0.014	0.178 ± 0.024	0.805 ± 0.164	0.087 ± 0.017	0.538 ± 0.006	0.035 ± 0.061	0.623 ± 0.194
cyto3 R.	0.874 ± 0.001	0.776 ± 0.002	0.791 ± 0.056	0.9 ± 0.068	0.854 ± 0.064	0.9 ± 0.033	0.645 ± 0.138	0.902 ± 0.046
TissueNet	0.845 ± 0.073	0.736 ± 0.11	0.761 ± 0.145	0.943 ± 0.031	0.774 ± 0.142	0.873 ± 0.072	0.584 ± 0.295	0.878 ± 0.101

Model	Dice	IoU	MCC	Precision	Recall	Balanced Acc.	ARI	Accuracy
TissueNetR.	0.82 ± 0.03	0.695 ± 0.043	0.716 ± 0.098	0.881 ± 0.082	0.777 ± 0.117	0.859 ± 0.056	0.528 ± 0.215	0.862 ± 0.078
TN1	0.884 ± 0.008	0.792 ± 0.013	0.81 ± 0.039	0.926 ± 0.045	0.847 ± 0.022	0.904 ± 0.015	0.677 ± 0.112	0.913 ± 0.037
TN1 R.	0.898 ± 0.033	0.816 ± 0.055	0.832 ± 0.001	0.889 ± 0.084	0.911 ± 0.02	0.924 ± 0.009	0.715 ± 0.032	0.924 ± 0.012
scratch	0.903 ± 0.003	0.823 ± 0.005	0.836 ± 0.044	0.906 ± 0.066	0.904 ± 0.059	0.926 ± 0.03	0.715 ± 0.105	0.924 ± 0.033
Vimentin								
cyto3	0.497 ± 0.089	0.333 ± 0.079	0.419 ± 0.039	0.622 ± 0.131	0.454 ± 0.201	0.686 ± 0.067	0.294 ± 0.041	0.821 ± 0.005
cyto3 R.	0.679 ± 0.134	0.521 ± 0.154	0.614 ± 0.148	0.535 ± 0.165	0.959 ± 0.01	0.862 ± 0.061	0.382 ± 0.246	0.805 ± 0.106
TissueNet	0.74 ± 0.094	0.592 ± 0.119	0.673 ± 0.11	0.659 ± 0.148	0.857 ± 0.002	0.868 ± 0.032	0.534 ± 0.159	0.874 ± 0.053
TissueNet R.	0.682 ± 0.139	0.526 ± 0.161	0.62 ± 0.152	0.536 ± 0.171	0.97 ± 0.015	0.866 ± 0.063	0.385 ± 0.259	0.804 ± 0.112
TN1	0.727 ± 0.103	0.576 ± 0.128	0.661 ± 0.116	0.623 ± 0.158	0.89 ± 0.02	0.871 ± 0.034	0.498 ± 0.184	0.858 ± 0.066
TN1 R.	0.718 ± 0.116	0.567 ± 0.142	0.66 ± 0.128	0.584 ± 0.151	0.953 ± 0.003	0.883 ± 0.047	0.463 ± 0.209	0.842 ± 0.079
scratch	0.705 ± 0.121	0.551 ± 0.145	0.642 ± 0.134	0.57 ± 0.157	0.946 ± 0.008	0.874 ± 0.05	0.438 ± 0.219	0.831 ± 0.086

Table S4.2: ROI level evaluation of the test set of lymphoid, myeloid, tumour and vimentin masks using CellProfiler, Mesmer, Cellpose model TN1 and retrained version. Metrics reported are Precision, Recall, F1 score, total number of annotated objects, and number of ROIs predicted.

Model	Precision	Recall	F1 Score	N Objects	N Predictions
Lymphoid					
CellProfiler	0.73 ± 0.39	0.72 ± 0.08	0.68 ± 0.16	72.00 ± 36.77	68.50 ± 9.19
Mesmer	0.85 ± 0.21	0.63 ± 0.07	0.71 ± 0.03	72.00 ± 36.77	50.50 ± 10.61
Cellpose	0.64 ± 0.40	0.89 ± 0.02	0.71 ± 0.27	72.00 ± 36.77	130.00 ± 12.73
Cellpose R.	0.90 ± 0.14	0.51 ± 0.12	0.64 ± 0.06	72.00 ± 36.77	33.00 ± 2.83
Myeloid					
CellProfiler	0.80 ± 0.01	0.86 ± 0.03	0.83 ± 0.02	54.00 ± 7.07	68.00 ± 15.56
Mesmer	0.86 ± 0.00	0.81 ± 0.17	0.83 ± 0.09	54.00 ± 7.07	64.00 ± 31.11
Cellpose	0.68 ± 0.02	0.95 ± 0.00	0.79 ± 0.02	54.00 ± 7.07	132.00 ± 33.94
Cellpose Retrained	0.83 ± 0.06	0.73 ± 0.01	0.78 ± 0.03	54.00 ± 7.07	43.00 ± 5.66
Tumour					
CellProfiler	0.97 ± 0.02	0.92 ± 0.01	0.94 ± 0.01	141.00 ± 39.60	244.00 ± 113.14
Mesmer	0.99 ± 0.01	0.97 ± 0.02	0.98 ± 0.01	141.00 ± 39.60	175.00 ± 72.13
Cellpose	0.89 ± 0.09	0.96 ± 0.02	0.92 ± 0.04	141.00 ± 39.60	136.00 ± 35.36
Cellpose Retrained	0.96 ± 0.05	0.88 ± 0.01	0.91 ± 0.03	141.00 ± 39.60	101.00 ± 36.77
Vimentin					
CellProfiler	0.75 ± 0.17	0.86 ± 0.07	0.80 ± 0.07	116.50 ± 26.16	172.00 ± 66.47
Mesmer	0.82 ± 0.14	0.95 ± 0.04	0.88 ± 0.07	116.50 ± 26.16	189.00 ± 42.43
Cellpose	0.73 ± 0.14	0.85 ± 0.08	0.77 ± 0.05	116.50 ± 26.16	229.00 ± 103.24
Cellpose Retrained	0.90 ± 0.08	0.49 ± 0.02	0.63 ± 0.04	116.50 ± 26.16	61.50 ± 12.02

Table S4.3: Pixel evaluation metrics—IoU, Dice coefficient, precision, recall, F1 score, MCC, and balanced accuracy—for the test set of lymphoid, myeloid, tumour, and vimentin masks using CellProfiler, Mesmer, Cellpose model TN1 and its retrained version.

Model	Dice	IoU	MCC	Precision	Recall	Balanced Acc.	ARI	Accuracy
Lymph								
CellProfiler	0.573 ± 0.13	0.408 ± 0.13	0.565 ± 0.12	0.596 ± 0.33	0.629 ± 0.10	0.799 ± 0.04	0.508 ± 0.12	0.942 ± 0.00
Mesmer	0.609 ± 0.06	0.439 ± 0.06	0.601 ± 0.06	0.752 ± 0.17	0.517 ± 0.00	0.753 ± 0.00	0.557 ± 0.03	0.956 ± 0.02
Cellpose	0.620 ± 0.18	0.462 ± 0.19	0.612 ± 0.18	0.662 ± 0.37	0.637 ± 0.01	0.807 ± 0.01	0.564 ± 0.18	0.952 ± 0.01
Cellpose R.	0.668 ± 0.01	0.502 ± 0.01	0.668 ± 0.02	0.828 ± 0.19	0.577 ± 0.09	0.785 ± 0.04	0.621 ± 0.02	0.962 ± 0.02
Myeloid								
CellProfiler	0.656 ± 0.03	0.488 ± 0.03	0.624 ± 0.04	0.673 ± 0.00	0.641 ± 0.06	0.806 ± 0.03	0.581 ± 0.06	0.941 ± 0.02
Mesmer	0.668 ± 0.06	0.504 ± 0.07	0.647 ± 0.05	0.700 ± 0.10	0.671 ± 0.20	0.818 ± 0.09	0.594 ± 0.03	0.944 ± 0.02
Cellpose	0.644 ± 0.00	0.475 ± 0.00	0.614 ± 0.02	0.696 ± 0.02	0.599 ± 0.01	0.787 ± 0.00	0.570 ± 0.03	0.942 ± 0.02
Cellpose R.	0.634 ± 0.06	0.466 ± 0.07	0.617 ± 0.07	0.763 ± 0.00	0.545 ± 0.09	0.765 ± 0.05	0.568 ± 0.09	0.945 ± 0.02
Tumour								
CellProfiler	0.883 ± 0.01	0.790 ± 0.02	0.812 ± 0.03	0.940 ± 0.04	0.832 ± 0.02	0.901 ± 0.01	0.678 ± 0.11	0.913 ± 0.04
Mesmer	0.929 ± 0.01	0.868 ± 0.01	0.879 ± 0.05	0.963 ± 0.01	0.898 ± 0.02	0.937 ± 0.02	0.787 ± 0.11	0.944 ± 0.03
Cellpose	0.879 ± 0.01	0.783 ± 0.01	0.802 ± 0.04	0.916 ± 0.06	0.846 ± 0.04	0.902 ± 0.02	0.663 ± 0.11	0.909 ± 0.04
Cellpose R.	0.832 ± 0.04	0.713 ± 0.06	0.734 ± 0.01	0.872 ± 0.11	0.800 ± 0.02	0.869 ± 0.00	0.565 ± 0.08	0.879 ± 0.03
Vimentin								
CellProfiler	0.653 ± 0.00	0.484 ± 0.00	0.574 ± 0.01	0.687 ± 0.12	0.640 ± 0.11	0.780 ± 0.03	0.450 ± 0.02	0.861 ± 0.01
Mesmer	0.754 ± 0.06	0.607 ± 0.08	0.688 ± 0.08	0.706 ± 0.10	0.811 ± 0.02	0.861 ± 0.03	0.572 ± 0.10	0.891 ± 0.03
Cellpose	0.536 ± 0.05	0.367 ± 0.04	0.470 ± 0.00	0.698 ± 0.15	0.455 ± 0.12	0.699 ± 0.04	0.347 ± 0.02	0.841 ± 0.00
Cellpose R.	0.499 ± 0.03	0.332 ± 0.02	0.466 ± 0.00	0.783 ± 0.09	0.369 ± 0.05	0.671 ± 0.02	0.335 ± 0.02	0.848 ± 0.01

5

GRIDGEN: Guided Region Identification based on Density of GENes – Use of transcript densities to unveil the tissue microenvironment

The advent of spatial omics brought unprecedented power to study biological processes within tissues while preserving spatial context and morphology. Most current analysis methods are cell-centric, relying on cell segmentation in 2D images to identify and characterize individual cells before performing downstream analyses, such as neighbourhood and niche identification. However, certain biological questions may be better addressed using cell-free approaches, which also eliminate unnecessary computations when cell segmentation is not essential. To address this need, we developed GRIDGEN (Guided Region Identification based on Density of GENes), an innovative approach for defining regions of interest based on transcript density. GRIDGEN enables the identification of biologically relevant tissue compartments – such as cancer-stroma interfaces or cell-type-enriched areas – without requiring prior knowledge. By bypassing cell segmentation, our approach provides a flexible analysis of spatial omics data, allowing researchers to investigate tissue organization and microenvironmental features.

Keywords: Subcellular spatial transcriptomics, segmentation-free approach, regions, cancer, tissue mask

---

This chapter was co-authored by Dr. Marieke Ijsselsteijn and Dr. Jessica Roelands and is being prepared for publication.

5.1 Introduction

The advent of spatial omics has revolutionized the study of tissue microenvironments, providing unprecedented insight into spatial organization, cellular heterogeneity, and cell-to-cell interactions [6, 7, 25]. In particular, for spatial transcriptomics, several commercially available technologies enable the detection of hundreds of RNA transcripts at subcellular resolution [12, 13], significantly expanding our ability to

explore complex biological systems. However, these advancements also expose limitations in current analytic approaches.

Standard analysis pipelines for single-molecule spatial transcriptomics data are primarily cell-centric, relying on image segmentation to identify and classify cells before downstream analysis [226, 242]. Tissues comprise heterogeneous cell populations with diverse shapes, sizes, and packing densities, often lacking a consistent orientation, which makes accurate segmentation inherently challenging and increases the risk of information loss. These inherent complexities of spatial data can limit the effectiveness of cell-centric approaches, potentially hindering the detection of specific cell populations. Furthermore, cell segmentation and identification may not always be necessary for understanding biological processes in spatial contexts.

An alternative to cell-centric methods is the use of segmentation-free approaches, which analyse spatial omics data without relying on cell segmentation. Instead, these methods define spatial neighbourhoods or compartments based on the underlying distribution of molecular points. Studies have shown that segmentation-free approaches can accurately identify cells [168, 169]. Additionally, there has been a recent surge in methods designed for unsupervised identification of spatial domains [195, 269] and transcript expression patterns [202, 270]. For example, Points2Regions is a Python tool that identifies regions based on mRNA composition within spatial bins of varying sizes. By integrating hierarchical and k-means clustering, it enables fast and scalable region discovery [198].

While unsupervised methods are valuable, many studies may benefit from a more targeted approach. Researchers often need to define specific regions based on prior knowledge to address focused questions, such as identifying tissue compartments with distinct cellular compositions (e.g., cancer vs. stroma), analysing biological processes occurring at specific sites within a tissue, or characterizing regions dominated by particular cell types or gene expression profiles. Moreover, rare cell types or those defined by only a few transcripts may be overlooked during segmentation or clustering. Directly interrogating tissue for regions of interest based on transcript density provides a straightforward and effective solution to these challenges.

This paper introduces Guided Region Identification based on Density of GENes (GRIDGEN), a Python package for supervised exploration of spatial omics data. GRIDGEN allows researchers to identify regions of interest by incorporating prior biological knowledge about the tissue and its cellular composition, enabling a hybrid approach that combines transcript density with cell segmentation data. Using high-resolution single-molecule spatial transcriptomics data from CosMx [13] and Xenium [12] on CRC samples, this study demonstrates how GRIDGEN can extract insights without relying on traditional cell segmentation pipelines. GRIDGEN's flexibility and precision make it a powerful tool for both exploratory and hypothesis-driven spatial- omics research.

5.2 Results

5.2.1 Description of GRIDGEN

GRIDGEN (Guided Region Identification based on Density of GENes) is a Python package designed for identifying and analysing regions of interest based on transcript density, enabling rapid and customizable exploration of tissues in spatial omics data. It allows users to create tissue masks based on prior knowledge while being a flexible tool. The package is freely available at `https://github.com/deMirandaLab/GRIDGEN`.

GRIDGEN facilitates the analysis of spatial transcriptomics data by providing a step-by-step workflow for identifying and characterizing regions of interest (Figure 5.1).

First, users define areas of interest within the tissue (Contour identification), which GRIDGEN converts into digital masks (Mask generation). These masks can be expanded to analyse the surrounding tissue microenvironment. GRIDGEN can retrieve relevant data within these masks, including transcript counts, morphological features, and hierarchical tissue organization. Finally, these generated masks can be overlaid onto existing cell segmentation data, enabling rapid annotation of cells based on the defined regions. This integration facilitates a complementary approach to cell segmentation, allowing researchers to combine transcript density information with cell-based data. Each submodule is explained in detail below.

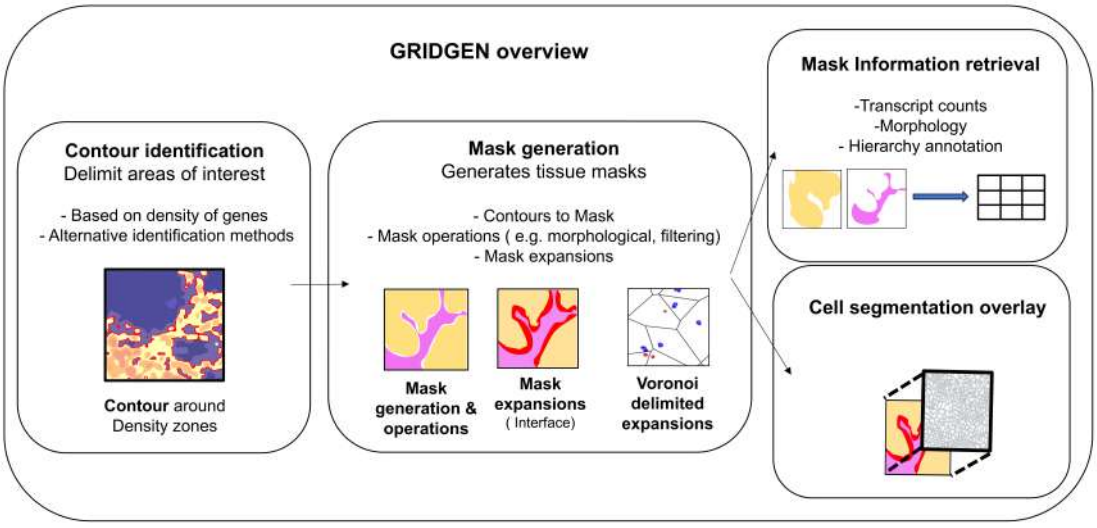


Figure 5.1: GRIDGEN overview of the different modules namely Contour Identification, Mask generation, Mask information retrieval and Cell segmentation overlay.

5.2.1.1 Contour identification

GRIDGEN's primary objective is to identify and outline areas of interest - contouring. It accomplishes this by detecting regions with a high density of user-defined transcripts and generating contours around them (Figure 5.2). GRIDGEN primarily relies on the convolutional sum method to locate these regions.

In the convolutional sum method, each point is assigned a value based on the aggregated counts within a defined window size. As this sliding window moves across the tissue, it generates a new image representing the convolutional sum. Contours are then defined using specific rules, such as highlighting regions where values exceed or fall below a given threshold.

In addition to convolutional-based region detection, GRIDGEN provides two alternative methods for mask generation: neighbourhood counting using K-Dimensional-Tree (K-D Tree) and unsupervised clustering via self-organizing maps (SOM).

The neighbourhood counting method, based on K-D Tree algorithm, operates similarly to the convolutional sum approach but differs in implementation and output. Instead of applying a sliding window to create a continuous density map, this method uses a K-D Tree to identify all neighbouring transcripts within a user-defined radius for each point. The result is a new set of point-wise neighbour counts, rather than a pixel-based image. This approach is particularly suited for analyses focusing on sparse or localized transcript populations. However, because the method assigns values only to transcript locations (not the entire spatial grid), interpolation may be necessary to generate contiguous regions. Once interpolated, contours are extracted using the same density-based thresholding rules as with the convolutional approach. While this method may be computationally intensive for large-scale datasets, where the convolutional sum might be more efficient, it offers finer granularity for smaller, focused regions.

GRIDGEN also supports a SOM-based region detection method. In this approach, transcriptomics data is firstly aggregated into spatial bins, which are then classified using the SOM algorithm [34, 249] in an unsupervised fashion. The bin size determines the spatial resolution, while the number of SOM units defines how many distinct region types are identified. This method enables the discovery of spatially coherent expression domains without relying on predefined gene sets or thresholds, making it especially useful for exploratory analyses.

GRIDGEN provides key adjustable parameters — such as kernel size and minimum transcript counts — allowing users to fine-tune the granularity of detected regions. It also supports contour filtering, enabling the removal of regions based on customizable criteria (e.g., excluding contours below a certain size or with insufficient gene representation). For added flexibility, GRIDGEN allows users to manually define, and input contours generated through other techniques.

5.2.1.2 Tissue Mask Mapping and Expansion with GRIDGEN

GRIDGEN enables the conversion of contour data into tissue mask maps, providing options for refining and manipulating these masks to facilitate tissue segmentation into biologically relevant compartments.

1. From contours to masks:

   Upon receiving a set of contours, GRIDGEN generates corresponding masks that can be enhanced using various techniques:

   - Object Filtering: Removes small objects below a defined area threshold, ensuring only significant features remain.

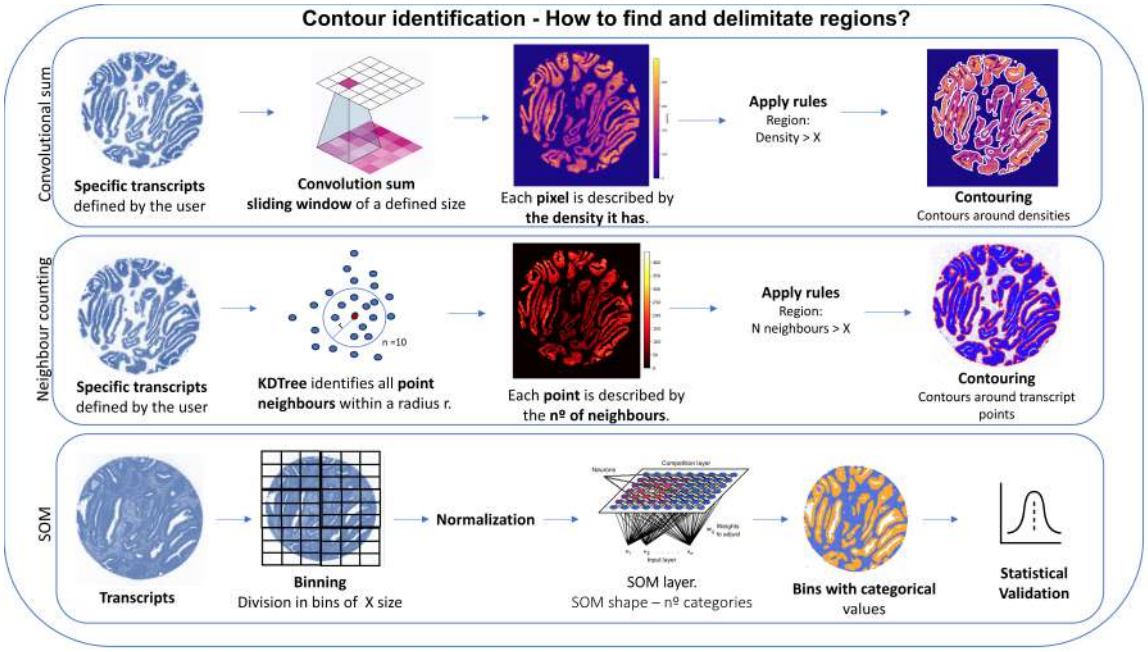


Figure 5.2: Contour identification strategies namely Convolutional sum, neighbourhood counting, and SOM based methods.

- Morphological Operations: Applies operations such as hole filling and smoothing to improve mask quality and consistency.
- Mask-Level Operations: Combines, intersects, or differentiates masks to create meaningful tissue representations.

2. Expanding Masks:

   GRIDGEN supports mask expansion to explore the surroundings of defined regions. The expansions can be:

   - Simple Distance-Based Expansion: Expands masks uniformly by specified distances (e.g. 5 µm) from the mask's edge.
   - Multiclass Object Analysis Mode: Ensures spatially distinct object expansions without overlap. Using Voronoi diagrams,

GRIDGEN defines precise boundaries, making it ideal for comparing neighbourhoods of distinct objects that can be closely located.

5.2.1.3 Mask Information Retrieval

GRIDGEN provides comprehensive tools to extract data from masks:

- Transcript Analysis: Extract transcript counts from masks.
- Morphological Analysis: Measures attributes such as area, perimeter, and shape descriptors.

- Hierarchy Annotation: Identifies hierarchical relationships between original objects and their expansions, linking derived regions to their parent contours.

This information can be retrieved over:

- Entire Mask: Measurements across the full area covered by a type of mask (e.g., the full tumour or stroma region).
- Individual Objects: Analysis of each connected region within a mask.
- Grids Within Objects: Data extracted from regularly spaced grids placed inside each object.

5.2.1.4 Cell Segmentation Overlay

GRIDGEN seamlessly integrates with cell segmentation data by overlaying segmentation results onto tissue masks. It annotates cells based on their spatial relationships with defined regions, providing insights into cell distribution within specific regions.

5.2.2 Tumour Stroma analysis with GRIDGEN

For solid cancer types, the main biological compartments within a tumour tissue include epithelial regions, primarily including cancer cells, and stromal regions, which primarily include stromal and immune cells. Pathologists typically identify these regions routinely based on tissue morphology, in ST experiments, these can be delineated based on areas with high expression of defined gene sets, bypassing the need for cell segmentation. Deriving masks for the major biological compartments (e.g., epithelial and stroma regions) can facilitate downstream analyses such as understanding spatial heterogeneity, tumour microenvironment interactions, or identifying tumour-stroma border zones.

5.2.2.1 Definition of epithelium-stroma analysis in CRC on CosMx and Xenium

As an illustrative example, we applied GRIDGEN on CosMx and Xenium, platforms with inherent technologic variations in resolution and transcriptome coverage, to generate tumour and stroma masks in CRC samples. To achieve this, convolutional kernels were used to delineate empty regions - with low overall density of genes – and tumour regions based on tumour-specific genes. After identifying contours for tumour and non-empty regions, mask maps of tumour-stroma were derived (Figure 5.3A-B). A tumour mask is defined by the tumour contours and a stroma mask is derived by negative selection, encompassing all regions that are neither tumour nor empty (Figure 5.3C). The stroma is defined through negative selection because its heterogeneous nature makes it challenging to identify a universal set of defining genes. By excluding tumour and empty areas, the remaining regions effectively represent the stroma.

Parameters of kernel size and density thresholds were adapted accordingly to each technology to reflect each platform's unique characteristics. Furthermore, these parameters can also be adjusted to define regions with varying levels of granularity—ranging from highly detailed, cell-level distinctions to broader,

smoother masks. For instance, this flexibility allows immune cells within the tumour to be included within the tumour mask, thereby defining a comprehensive tumour region (Figure S5.1).

5.2.2.2 Tumour Stroma border analysis in Xenium CRC data

In addition to delineating tumour and stromal regions, it is biologically relevant to characterise the border between these compartments. To address this, the tumour mask can be expanded into the surrounding stroma areas for further analysis. We used a cohort of CRC tumour cores, profiled by Xenium, and defined the tumour and stroma areas as previously described. The tumour mask was subsequently expanded by 10 and 20 μm into the stroma, and the transcript counts within these expanded regions were analysed. Differential gene expression across these masks was assessed (Figure 5.3D), and the identified transcripts were plotted back at the cores (Figure 5.3E), highlighting the distinct transcriptomic patterns at the tumour-stroma interface. Of note, the same analysis can be performed on individual tumour objects separately, enabling the assessment of their respective borders.

5.2.3 Single and Multiclass Object analysis

Beyond delineating tumour and stroma areas, defining regions based on pre-defined transcripts can be particularly valuable, for instance in scenarios where cell segmentation encounters challenges.

5.2.3.1 Definition of population-specific objects in CRC in CosMx

Cell segmentation, while a cornerstone of many spatial transcriptomics workflows, has limitations. For instance, overlapping cells often pose significant challenges, particularly in cancer tissues where small immune cells infiltrate tumours. These infiltrating cells are frequently under-segmented, leading to their erroneous assignment to larger tumour cells. Additionally, cell phenotyping can be a bottleneck when using spatial transcriptomics platforms with a limited transcript repertoire, as cell types defined by only 2–3 transcripts can easily be misidentified (Figure S5.2).

Using the same CRC CosMx cohorts analysed above, we sought to identify $\gamma\delta$ T cells, which are otherwise difficult to detect through cell clustering and phenotyping (Figure 5.4A). To achieve this, we applied GRIDGEN's convolutional approach to identify putative regions enriched for $\gamma\delta$ T cell transcripts. Specifically, we identified areas where *TRGC* and *TRDC* transcripts were within 5.5 μm of each other, with contours highlighting these regions. Beyond the presence of TRGC and TRDC transcripts in a confined area, specific criteria were applied, including thresholds on transcript counts, contour size, and transcript combinations, specifically requiring the sum of *TRGC* and *TRDC* counts to be higher than the sum of *TRAC* and *TRBC*, with at least one *TRGC* and one *TRDC*. The resulting regions represent putative $\gamma\delta$ T cell areas (Figure 5.4B) which can be further analysed to explore neighbouring regions or to assess their spatial relationship with tumour and stromal compartments. They could also be used, paired with cell segmentation masks, to identify cells that, not only have the same rules of transcripts but, where these transcripts are close together.

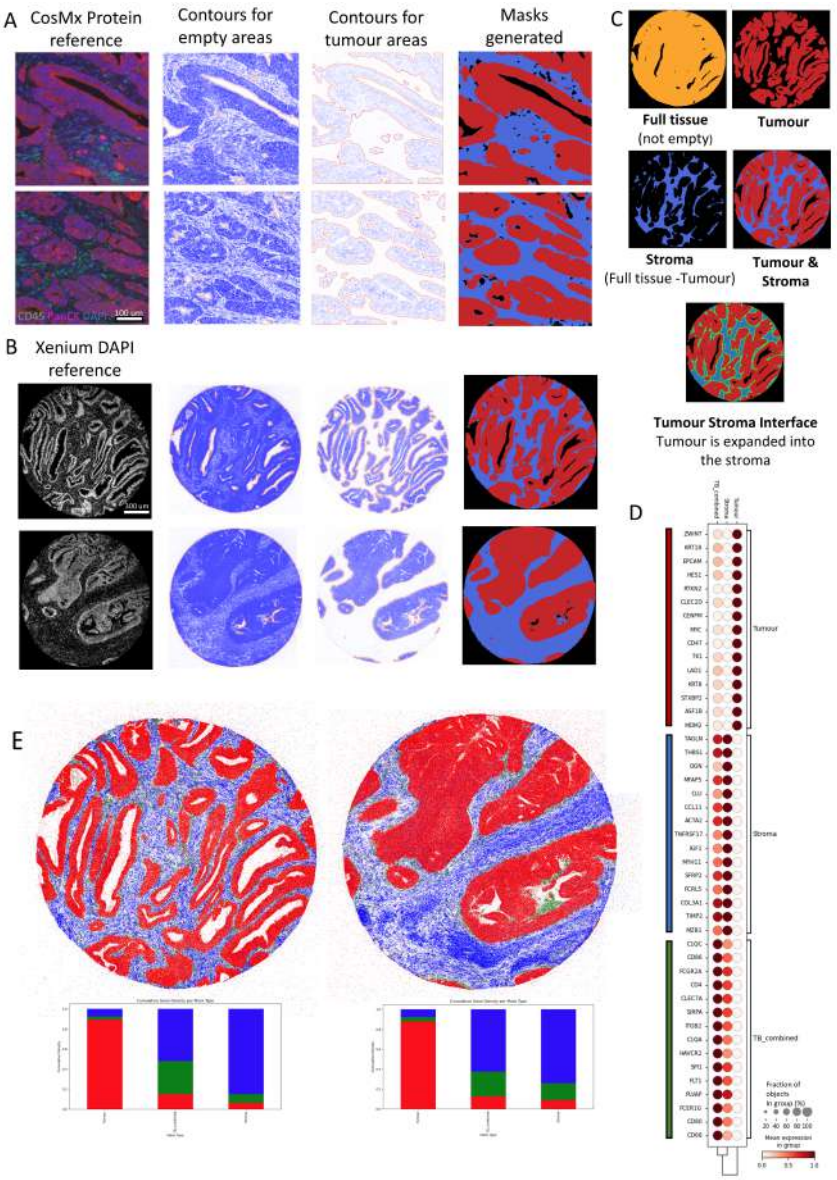


Figure 5.3: Tumour and Stroma analysis with GRIDGEN. A) Example with CosMx data. From left to right, Protein reference with Pan Cyto keratin (pink) and general membrane markers (green), Contours for empty and tumour areas; and the generated masks for tumour and stroma areas. B) Example with Xenium data. From left to right, DAPI reference image that represents the nuclei, Contours for empty and tumour areas; and the generated masks for tumour and stroma areas. C) Schematic of the pipeline to generate tumour stroma masks: first, all the points are transformed into a tissue mask to identify empty regions; a tumour mask is defined based on the tumour transcript points; stroma regions are defined subtracting the tumour from the tissue region; the full tissue is now divided in tumour and stroma masks; to study the tumour-stroma interface, the tumour mask is expanded inside the stroma. D) For the cohort of 24 Xenium cores, tumour and stroma masks were defined and further expansions of 20um of the tumour were assessed, transcript counts were retrieved from the masks and differential gene expression between the 3 different regions was assessed and plotted. E) The transcripts differentially expressed were plotted back in two example cores with points for tumour in red, stroma in blue and points differentially expressed in the tumour border in green. Graphics that show the cumulative transcript density per mask for each specific example core further demonstrate the higher presence in each of the mask of the genes detected in the differential analysis.

5.2.3.2 Multi class Object analysis

Following the same principles outlined in the single-object analysis, it is also possible to extend the approach to explore multiple object types simultaneously. While running the single-object analysis multiple times could achieve this, it does not account for overlapping areas between different object types. The multiclass object analysis resolves this by generating Voronoi diagrams, which constrain expansions to areas closer to one object type than to any other. This ensures that regions around object type A are strictly associated with A and not influenced by proximity to object type B. We exemplify this approach using the previously defined $\gamma\delta$ contours along with contours identifying CD8+ T-cell regions (Figure 5.4C). CD8+ T-cell regions were defined using a kernel size of 5.5 μm to identify areas containing more than two transcripts of *CD8*, *TRAC*, and *TRBC*. These regions were further filtered to ensure they contained more *TRGC+TRDC* transcripts than *TRAC+TRBC* transcripts, with at least one count of each transcript present. The analysis includes differential gene expression profiling between areas surrounding $\gamma\delta$ T and CD8+ T-cell regions, highlighting distinct transcriptomic patterns in their respective vicinities (Figure 5.4D). The vicinity of $\gamma\delta$ T cells is characterised by the expression of epithelial markers such as *EPCAM* and *KRT18*, suggesting close proximity to epithelial cells. In contrast, areas surrounding CD8 T cells exhibit elevated expression of HLA-related genes, including *HLA-DRB*, *CD74*, *HLA-DRA*, and *HLA-DQA1*, indicating active antigen presentation in their local environment.

5.2.4 Mapping with cell segmentation

Defining regions of interest does not preclude the use of cell segmentation; rather, the two approaches can be seamlessly integrated to facilitate specific tasks. GRIDGEN-generated masks can be overlaid onto cell segmentation data to assign spatial labels to cells—for instance, identifying them as intratumoral, stromal, or within the tumour border. While similar results can be achieved through cell segmentation alone, this approach often requires defining specific parameters, such as the number of neighbouring cells or distances in micrometres, which can be complex and less intuitive. By leveraging masks, these tasks become more straightforward and less time-consuming. In Figure 5.5, we illustrate an example of a Xenium core, overlaying the tumour, stroma, and tumour-border masks (generated above) with cell segmentation. Immune cells are colour-coded by the region they belong to, revealing distinct spatial patterns: T cells are more concentrated in tumour regions, B cells are prevalent in the stroma and tumour-border areas, and myeloid cells are more evenly distributed across all regions.

5.2.5 Alternative ways of masking

While the primary strategy for deriving contours in this work relies on convolutional approaches, several other methods for generating effective masks are also possible. In addition to convolutional techniques, GRIDGEN supports masks created using K-D Tree-based neighbour counting and Self-Organizing Maps (Figure S5.3).

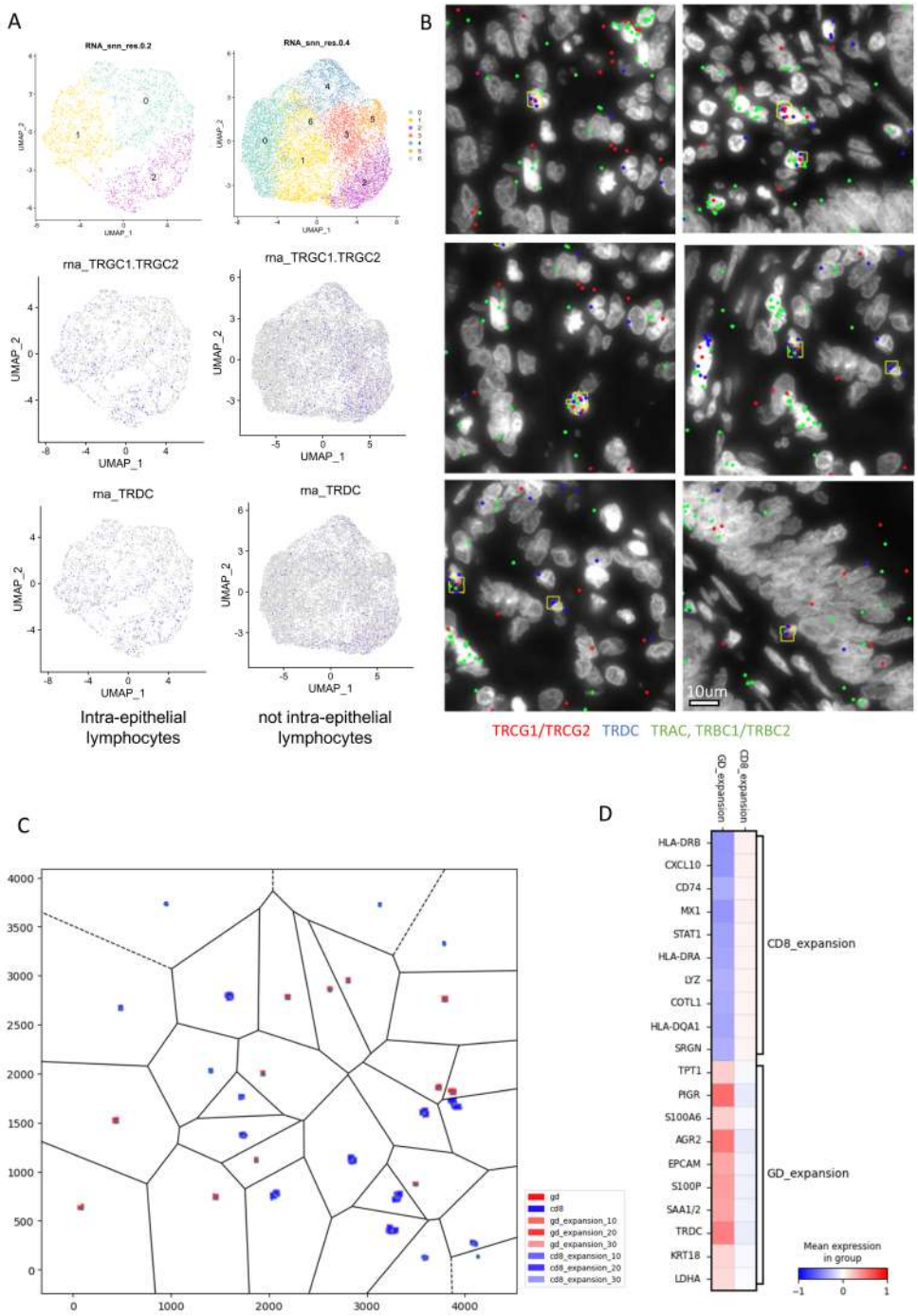


Figure 5.4: Population specific analysis with GRIDGEN. A) T-SNE clustering of subclusters of lymphocytes obtained by cell segmentation, intraepithelial and not intraepithelial and the presence of *TRGC* and *TRDC* in the cells, showing that $\gamma\delta$ T cells are not possible to subset using cell segmentation and clustering. B) Example visualization of GRIDGEN-defined contours for $\gamma\delta$ T objects overlaid on DAPI-stained tissue with transcript points (*TRGC*, *TRDC*, *TRBC*, *TRAC*) to find areas with *TRGC* and *TRDC* close together (points distant maximum 5.5 μm apart) and that pass the defined thresholds. C) Multiclass Object Analysis of $\gamma\delta$ and CD8+ T-cell Regions in CRC Samples. Example of a Voronoi diagram used to partition regions, ensuring non-overlapping expansions for the two object types. D) Differential gene expression analysis comparing transcriptomic patterns in the vicinity of $\gamma\delta$ regions and CD8+ T-cell regions, with expansions of 20 μm. The heatmap illustrates significant gene expression differences across these areas, underscoring distinct microenvironmental characteristic.

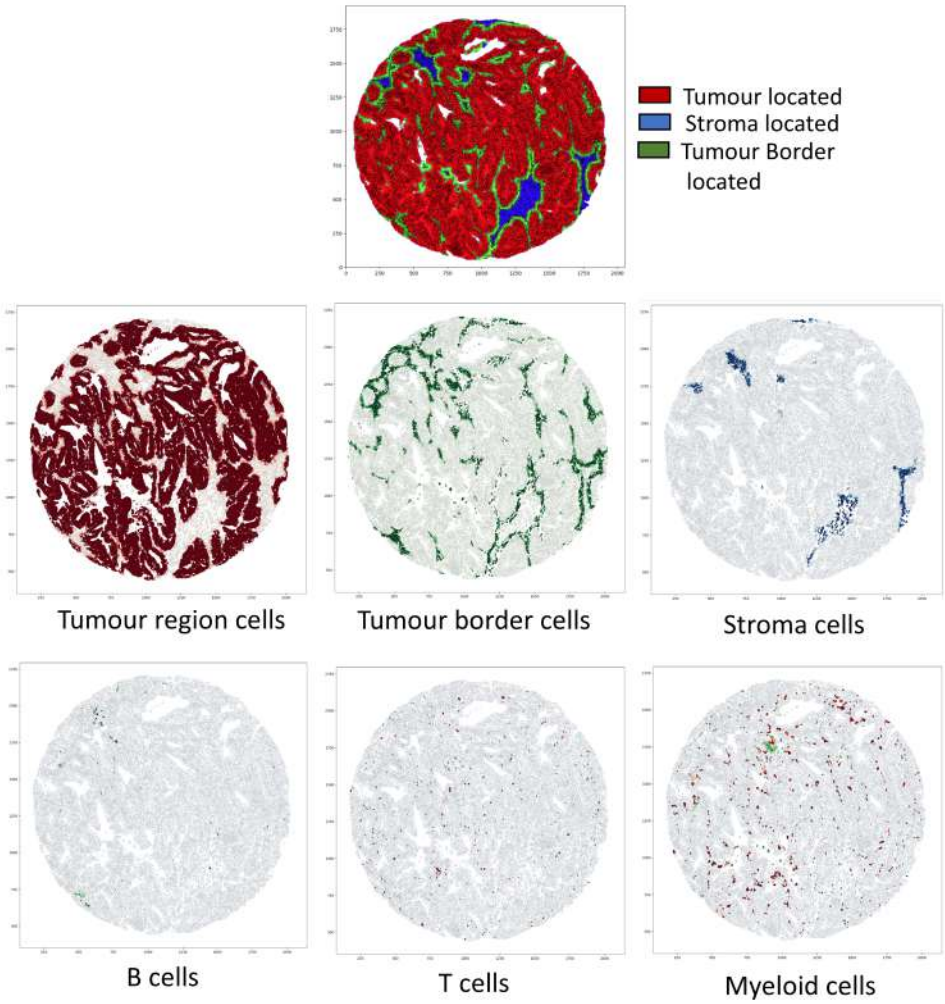


Figure 5.5: Overlay of Cell Segmentation and Tumour-Stroma-interface Masks. A) Xenium core (1.5 μm) with cell segmentation overlaid tumour (red), stroma (blue), and tumour-border (green) masks. B) Cells are coloured by the percentage of overlap with masks, red for tumour, green for tumour interface and blue for overlap with stroma. C) Immune cells identified through cell segmentation are colour-coded based on the region they overlap with: T cells are enriched in tumour regions, B cells are more prominent in stroma and tumour-border areas, and myeloid cells are distributed across all regions.

Deriving from previous work in spatial proteomics [249], SOM-based mask generation applies the SOM algorithm [34] to bins of image data, classifying each bin in an unsupervised manner. The bin size can be adjusted to control the granularity of the masks, while the number of SOM units determines the number of resulting groups. In this study, we used SOM with two units, leading to an unsupervised separation of stromal and epithelial compartments (Figure S5.3B), distinction supported by differential gene expression observed between the two classes (Figure S5.3C). This offers a possibility of deriving masks in an unsupervised fashion.

Neighbour counting based mask generation, in contrast to convolutional methods, relies on local neighbourhood point counts. The K-D Tree algorithm, a space-partitioning structure, efficiently organizes points in multidimensional space and is commonly used to accelerate nearest-neighbour searches in

local neighbourhood analysis. To exemplify this approach, we derived tumour and stromal regions (Figure S5.3D) by calculating, for each point, the number of neighbouring transcripts within a $10\,\mu$m radius — both across the full set of genes and a tumour-specific gene set. Points with a high number of tumour gene neighbours were classified as tumour regions, while areas lacking nearby transcripts were designated as empty or stromal.

Because this computation is performed on a per-point basis, the K-D Tree-based method is generally slower than convolutional approaches, particularly for large datasets. However, it can be effective for smaller regions or targeted analyses, where its higher resolution and point-level granularity may be advantageous.

We also applied the K-D Tree method to identify spatial associations between *TRGC* and *TRDC* transcripts (Figure S5.3E). Specifically, we searched for transcript pairs located within a $7\,\mu$m radius. Due to the sparsity of these transcripts and the lack of sufficient density for contour generation, we defined regions by drawing circular masks centered on the identified transcript pairs. These proximity-based interactions could also be projected onto segmented cell boundaries, enabling the identification of individual cells where *TRGC* and *TRDC* co-localize.

5.3 Discussion and Conclusions

Defining regions of interest in spatial transcriptomics offers an efficient approach for addressing specific biological questions. While cell segmentation-based methods are powerful, they may not always align with the needs of studies focused on tissue-level organization or areas defined by molecular density. For example, delineating tumour-stroma borders, identifying immune-enriched niches, or exploring spatial relationships between regions of interest can be more straightforward when approached through region-level analysis. Regions defined by transcript density provide an intuitive framework for such tasks, reducing dependence on segmentation pipelines that can struggle with small or overlapping cells. GRIDGEN introduces a straightforward method to define and analyse regions based on gene density. By leveraging user-defined gene sets and flexible kernel-based convolutional operations, the tool facilitates the identification of biologically meaningful areas without relying on cell segmentation. Its ability to generate tumour, stroma, and hybrid regions (e.g., tumour-stroma borders) was demonstrated effectively, and additional applications, such as identifying rare or ambiguous cell types, were also explored. Moreover, its capacity to integrate with cell segmentation pipelines ensures it can complement traditional workflows.

We would like to note that the reliability of this approach depends on several methodological parameters that require careful consideration. First, the accuracy of GRIDGEN relies heavily on the quality and representativeness of the selected genes. The identification of regions is constrained by the availability and specificity of transcript markers, which may introduce biases or limit applicability in poorly annotated systems. Second, the choice of kernel size and density thresholds, while offering flexibility, requires careful parameter tuning to balance precision and robustness. Misparameterization could result in either over-segmentation or under-segmentation of regions. Another consideration is that while GRIDGEN

bypasses cell segmentation challenges, it does not replace the need for cell-level analysis in certain contexts. Questions involving detailed cellular phenotyping or fine-scale interactions remain better addressed by traditional cell segmentation approaches or a hybrid strategy.

Taking into account these important considerations, GRIDGEN offers a complementary perspective in spatial omics analysis, particularly for questions that benefit from region-level insights. GRIDGEN provides an efficient and adaptable framework to define and analyse these regions, enabling researchers to address specific biological questions while mitigating some of the challenges posed by traditional cell segmentation.

While GRIDGEN presents a robust and flexible framework for region-level analysis, several avenues for future development remain. Currently, GRIDGEN requires users to fine-tune parameters and inspect region masks manually using scripts or notebooks. One major future improvement would be to integrate GRIDGEN into interactive tools such as a Napari plugin [246], rendering the notebook to a dashboard using Voilá, or create data apps using tools such as Dash Plotly or Streamlit. These interfaces would allow users to adjust parameters (like kernel size or gene set) in real time, visualize the resulting regions, and make the pipeline more accessible and intuitive, especially for non-technical users.

Another point of improvement would be the possibility to refine region boundaries around specific populations. While GRIDGEN is not designed to perform exact cell segmentation, it can sometimes partially capture structures like rare or population-specific cells. However, these delineations can be incomplete or imprecise when based on sparse transcript coverage. A future improvement would be to refine these regions by extending and completing partial contours. A coherent approach similar to others developed with good results [168], might involve:

(a) *Anchor Point Identification*: Start with transcript-dense "anchor" points identified by GRIDGEN.

(b) *Neighbourhood Feature Extraction*: For each point, derive a neighbourhood composition vector - a feature set summarizing nearby gene expression.

(c) *Probabilistic Assignment*: Use probabilistic models (e.g., Bayesian Gaussian Mixture Models, non-parametric density estimation) to learn the distribution of features from anchor points.

(d) *Point Classification*: Classify surrounding points by computing the likelihood that they belong to the same population, based on how well their feature vector matches the learned distribution. Proximity to anchor points can also be incorporated to guide this process.

(e) *Iterative Region Expansion*: Refine region boundaries using spatial models such as graph-based diffusion or MRFs, which propagate region membership probabilities across the spatial graph. Alternatively, the initial likelihood estimates from (d) can serve as input to a unified probabilistic framework, where an MRF jointly infers point assignments by minimizing an energy function that balances local evidence and spatial smoothness.

(f) *Boundary Definition*: Define final contours using the outermost high-confidence points, forming a more complete and biologically realistic boundary.

This process would allow GRIDGEN to better capture the shape and extent of sparse or ambiguous populations without requiring a full segmentation pipeline. Finally, GRIDGEN could be extended to include spatial statistics approaches to define regions based on spatial gene patterns. This could be achieved using spatial autocorrelation, such as Moran's I or Getis-Ord Gi* [271] to identify regions where genes of interest are spatially clustered or enriched and detect areas where sets of genes are spatially co-expressed.

5.4 Methods

5.4.1 GRIDGEN implementation

GRIDGEN is implemented in Python 3.10 and leverages several scientific libraries, including NumPy [272], Pandas [273], Matplotlib [274], SciPy [275], Scikit-image [254], and OpenCV [276]. It follows a modular design, enabling each module to function independently. For instance, externally derived masks or contours can be used as inputs without requiring GRIDGEN-generated data. To ensure transparency and traceability throughout the pipeline, structured logging is implemented. Additionally, the system supports extensive plotting at each pipeline stage for thorough visual validation.

5.4.1.1 Convolution-Based Contour Generation

To identify regions of enriched transcript density, we implemented a convolution-based contour detection approach. The method begins by summing expression values across all genes at each pixel, producing a 2D transcript abundance map. A square or circular kernel is then applied to compute local transcript densities. Contours are extracted from this smoothed image by applying user-defined thresholds and detecting connected regions. Depending on the selected direction ("higher" or "lower"), either high- or low-density regions are detected. Spurious contours are removed based on a minimum area criterion and the presence of non-zero transcript counts in the original data. The method also includes automated validation to discard noise and regions with no biological signal, enhancing both accuracy and interpretability. The contour operations are based on OpenCV library.

5.4.1.2 Neighbourhood counting-based Contour Generation

As an alternative, enriched transcript areas can be calculated directly with transcript points neighbouring counts, a method based on K-D Tree and query ball tree algorithm (Scipy implementation). This method proceeds in several steps. First, all foreground pixels (i.e., values equal to 1) are extracted from a binary array and interpreted as spatial coordinates. These are indexed using a K-D Tree structure to enable efficient local neighbour queries. Points are then filtered to retain only those that have at least N neighbouring points within a given radius, thus removing isolated detections and reducing noise.

From here, contours can be generated as described above. As transcripts are often sparse, to achieve better contours, OpenCV inpaint function can be used to interpolate values at missing positions. Alternatively, the filtered points can be grouped into spatial clusters using the DBSCAN [277] algorithm

(scikit-learn [47] implementation) and contours can be generated from these labelled groups of points. If the number of annotated points is very small, simpler methods of deriving contours are available: the simplest method computes a minimum enclosing circle for each cluster, ensuring a compact representation of local signal accumulation. Alternatively, convex or concave hulls (alpha shapes) are computed around each point cloud, depending on the desired boundary fidelity.

5.4.1.3 SOM-based Contour Generation

This method uses grid binning based on spatial coordinates, with customizable grid shapes, and leverages the AnnData [243] structure for preprocessing using Scanpy [242]. Spatial bins are classified using a SOM [34] implemented via MiniSOM package [278]. The classification results are mapped back to the original coordinates to define regions of interest. Statistical validation of SOM-based separation is available through t-tests, providing an assessment of the significance of spatial clustering.

5.4.1.4 Mask generation

From a set of contours binary masks are generated using OpenCV. Several post-processing operations are implemented, including area filtering to remove small, disconnected components; Hole filling; and morphological operations (erosion, dilation, opening, closing) to refine boundaries and reduce noise. Mask subtraction is also supported, allowing specific regions to be isolated by subtracting one or more masks from a base mask.

Masks can be expanded via OpenCV dilation and can be constrained by other masks (e.g., expanding a tumour mask only within the stroma). To avoid overlaps in multi-class mask expansions, a Voronoi-based constraint system is implemented. Using centroids of all objects, a Voronoi diagram is generated (via SciPy), and each object's expansion is limited to its respective Voronoi cell—ensuring that expansion remains closest to its source object.

Dedicated classes support the integration and expansion of multiple masks into cohesive, non-overlapping tissue masks.

5.4.1.5 Mask information retrieving

All mask objects are labelled using scikit-image's labelling tools. Transcript counts can be retrieved for the entire mask, individual labelled objects and gridded subregions within objects. This process supports multiprocessing for improved performance on large datasets. Morphological characteristics (e.g., area, perimeter, shape descriptors) are computed using scikit-image regionprops function. A hierarchical analysis is also supported: expanded objects can be traced back to their originating ("mother") object by storing and mapping object labels throughout the pipeline. The information is stored in a pandas' DataFrame.

5.4.1.6 Cell segmentation overlay

Cell segmentations (in mask or polygon format) can be overlaid onto the generated masks. For each cell, the number of overlapping pixels with each mask is computed, and the percentage of coverage is reported.

5.4.2 Data

Two cohorts of CRC cancer were used: 20 cores of $500\,\mu$m $\times$ $500\,\mu$m profiled with CosMx [13] and 20 cores with 1.5 mm profiled with Xenium [12]. There are key differences between the two cohorts: In CosMx tissue cores are square-shaped, measuring $500\,\mu$m$500\,\mu$m corresponding to images of 4245×4245 pixels (1pixel $\approx 0.12\,\mu$m,$1\,\mu$m ≈ 85pixels). CosMx provides data for 999 genes from a standard panel. In the Xenium cohort, tissue cores are circular, with a diameter of $1500\,\mu$m, corresponding to 1500 pixels (1pixel $\approx 1\,\mu$m). Xenium data includes 480 genes from a custom panel optimized for cell identification. These differences in resolution, pixel size, and gene coverage influence the analytical strategies and parameters used for each platform, ensuring accurate region delineation while accounting for their respective strengths.

Data received from the CosMx and Xenium were filtered for negative probes. Furthermore, and for Xenium, transcripts with quality metrics inferior to 20 were removed.

5.4.3 Tumour stroma definition in CosMx and Xenium

First, regions that have tissue were delimited and distinguished from empty regions. For CosMx, this was achieved by a kernel of 80×80 pixels (approximately 10×$10\,\mu$m) and by identifying areas with fewer than 140 total transcripts. For Xenium, a kernel of 10x $10\,\mu$m and identifying regions with at least 40 genes. Parameters for empty area identification were similarly adapted to the platform's resolution and gene count. Then, tumour regions were identified. For CosMx, tumour regions were identified using a set of tumour-specific genes and objects were defined using a convolutional kernel size of 80×80 pixels (approximately 10×$10\,\mu$m) with at least 40 tumour genes. For Xenium, a kernel of 10×$10\,\mu$m, and objects were retained if they contained at least 10 genes of the tumour genes genes. Tumour genes were chosen based on prior knowledge: CosMx tumour genes were *EPCAM*, *KRT19*, *KRT8*, *KRT18*, *KRT17*, *CEACAM6*, *SPINK1*, *CD24*, *S100A6*, *RPL37*, *S100P*. For Xenium the genes used were *EPCAM*, *SMIM22*, *CLDN3*, *KRT18*, *LGALS4*, *KRT8*, *ELF3*, *TSPAN8*, *STMN1*, *CD47*, *MYC*, *LGALS3*. The differences in these parameters reflect the inherent variations in resolution and transcript coverage between the two technologies, highlighting the importance of tailoring analyses to each platform's characteristics.

After identifying contours for tumour and non-empty regions, mask maps of tumour-stroma were derived. A tumour mask is defined by the tumour contours and a stroma mask is derived by negative selection, encompassing all regions that are neither tumour nor empty.

5.4.4 Tumour stroma Interface analysis

For the Xenium cohort, the tumour mask was subsequently expanded by 10 and $20\,\mu$m into the stroma, and the transcript counts within these expanded regions were analysed. The transcripts of the 4 masks (Tumour border to $10\,\mu$m, tumour border to $20\,\mu$m, the remaining stroma and the tumour mask), were extracted collectively for each mask and sample. The values followed the usual pipeline for single cell: the values were normalized to target sum 1×10^{-4} and log transformed. PCA, neighbours and the UMAP were

calculated using Scanpy[242] functions. For simplicity purposes, the tumour border was merged into one and genes for the different masks were ranked using Wilcoxon method and only considered with min fold change of 1. The values detected for stroma, tumour and border were plotted back in two example cores.

5.4.5 Population specific objects definition

Using the same CosMx cohort analysed above, we identified putative $\gamma\delta$ T areas using the convolutional sum approach, using a kernel of $45x45$ pixels ($5.3\ \mu$m) and a minimum of 2 transcripts of *TRGC* or/and *TRDC*. The identified regions were filtered and only considered valid if they had at least 1 *TRDC* and 1 *TRGC* transcript; the sum of *TRDC* and *TRGC* transcripts should be higher than the sum of *TRAC* or *TRBC*. CD8 T regions were identified using a kernel of $40x40$ pixels with a minimum of 2 transcripts of the set *TRAC*, *TRBC1/TRBC2*, *CD8A*, *CD8B*. Similarly, identified regions were only considered value if had at least one transcript of CD8 (A or B); at least one transcript of *TRAC*, at least one transcript of *TRBC*; the sum of *TRAC* or *TRBC* transcripts should be higher than the sum of *TRDC* and *TRGC*; and CD8 transcripts should be higher than CD4 transcript. Both the $\gamma\delta$ T areas and the CD8 T areas were transformed into masks. To derive expansions, we used the multiclass object analysis, that generates Voronoi diagrams, which constrain expansions to areas closer to one object type than to any other. Expansions of $20\ \mu$m of $\gamma\delta$ and CD8 T regions were derived in this fashioned and analysed in a similar way as the tumour interface masks, with normalization, PCA and Differentially Gene Expression (DGE). The DGE was calculated between the two expansions to interrogate directly the differences around these regions.

5.4.6 Cell segmentation

Cell segmentation for the CosMx cohort was the provided by the company and described in detail in [13]. For Xenium cohort, in an effort to improve the segmentation, we refine the segmentation using Baysor [168], a method that segments based on joint likelihood of transcriptional composition and cell morphology and can incorporate previous membrane stains. The segmentation provided by Xenium company was used as a prior with confidence of 0.7, min molecules per cell was set to 30, scale 10 and clusters 3. The remaining parameters were set to default.

5.4.7 Cell phenotyping

Similar approaches for cell phenotyping were taken for Xenium and CosMx cohorts. The cell counts were extracted, and cells with low transcripts were excluded. The cell counts were log normalized and the neighbours, PCA and UMAP were calculated. Leiden algorithm was used to identify separate clusters. First, stroma was separated from the epithelial cluster; the stroma cluster was further refined for general stroma cells such as fibroblasts and immune cells, namely, B cells, T cells and myeloid cells

5.4.8 Cell segmentation and mask overlay

Masks for the Tumour Stroma Interface were overlayed with the correspondent cell segmentation and phenotyping. For each cell segmented, the percentage of area inside each of the masks was calculated. Considering the highest mask in each cell, the frequency of T cells, B cells and myeloid cells were calculated per mask.

5.4.9 SOM-based masks

For the Tumour Stroma interface analysis using the Xenium cohort, an alternative approach based on SOM was applied. All the images were divided in bins of 5X5 μm, bins with less than 10 transcript counts were removed and the counts per bin was log normalized. A SOM is run with shape (2,1) with 5000 iterations, sigma 0.5 and learning rate 0.5. With a shape of (2,1), the bins are classified in 2 classes. To check if the categories divided are correspondent to Stroma and epithelial compartments, a statistical t-test to rank the top genes per category can be run. The classified bins were plotted back into the samples and the masks were derived.

5.4.10 Neighbour counting based masks

For exemplification purposes, we use K-D-tree-based mask generation to also derive Tumour stroma masks in Xenium. A KD-tree (implemented from Scipy [275]) was constructed using the spatial coordinates of the transcriptomic points to enable efficient nearest-neighbour searches in Euclidean space. For each point in the dataset, the query_ball_point method was employed to retrieve the indices of all neighbouring points within a 10-pixel radius, enabling the computation of local neighbourhood densities with optimized performance for large-scale spatial data. Given the inherent sparsity of transcriptomic points, we applied inpainting using the OpenCV[276] inpaint function to interpolate missing values and optimize contour detection. From this smoothed neighbourhood map, contours were generated following the same procedure described in Section 5.4.1.1.

Additionally, we used local point search (based on K-D Tree to identify the points of *TRGC* and *TRDC* close to each other using a radius of 60 pixels (≈ 7 μm). We label all the sets of close points as belonging to the same object using DBSCAN [277] a Density-Based Spatial Clustering implemented using scikit-learn [47]. Because only these transcripts were classified and the density of points was insufficient to generate contours, we instead drew circular regions that enclosed the identified points. These regions were then filtered using the same exclusion criteria described in Section 5.4.5.

5.5 Data and code availability

The GRIDGEN pipeline is fully available and ready to use at `https://github.com/deMirandaLab/GRIDGEN`. The repository also contains all the necessary code to replicate the methods comparison, along with detailed Jupyter notebooks for easy understanding and implementation.

5.6 Supplementary information

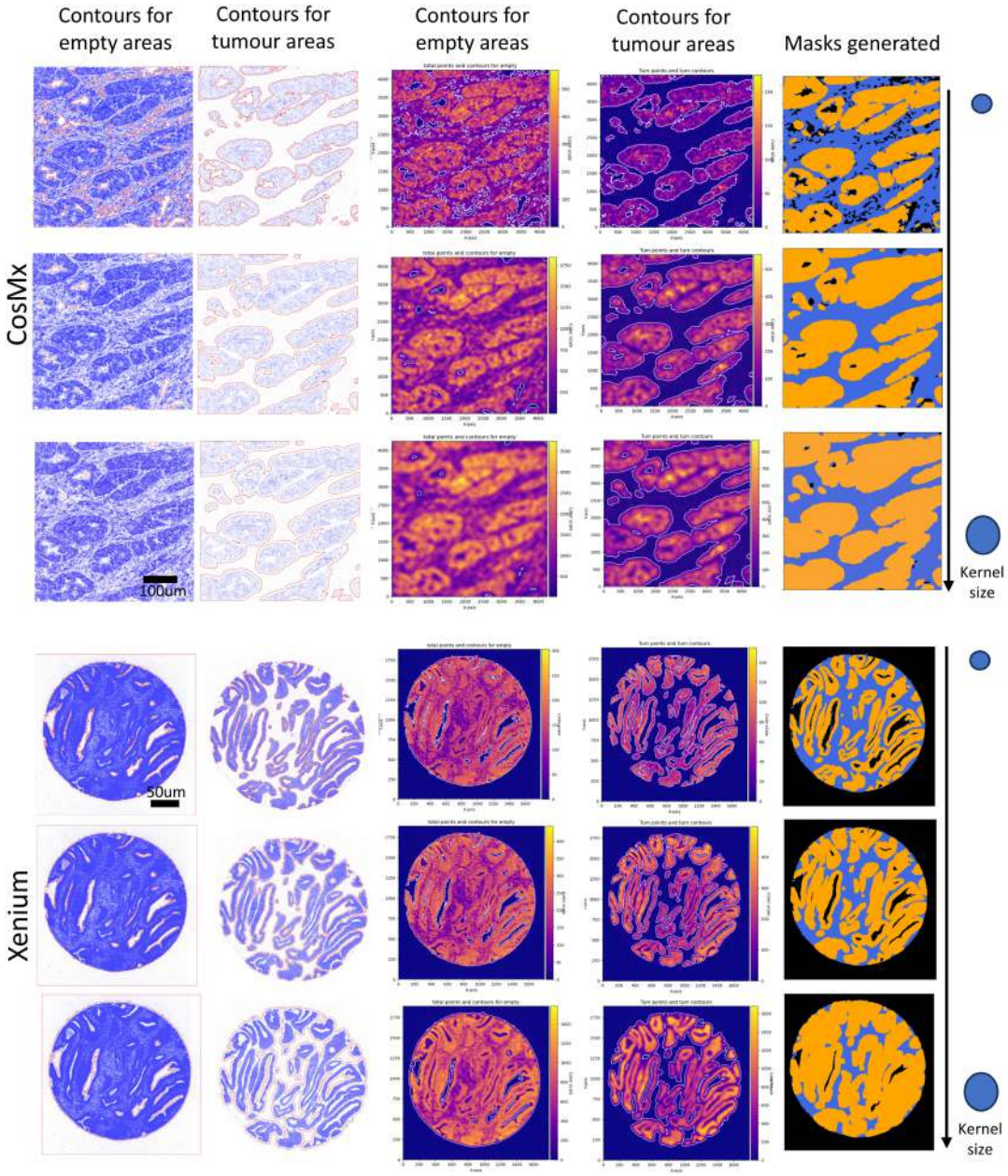


Figure S5.1: How the kernel size parameter can influence the generation of masks While bigger kernel sizes leads to more smooth masks, lower kernel size can be used to more fine grained masks.

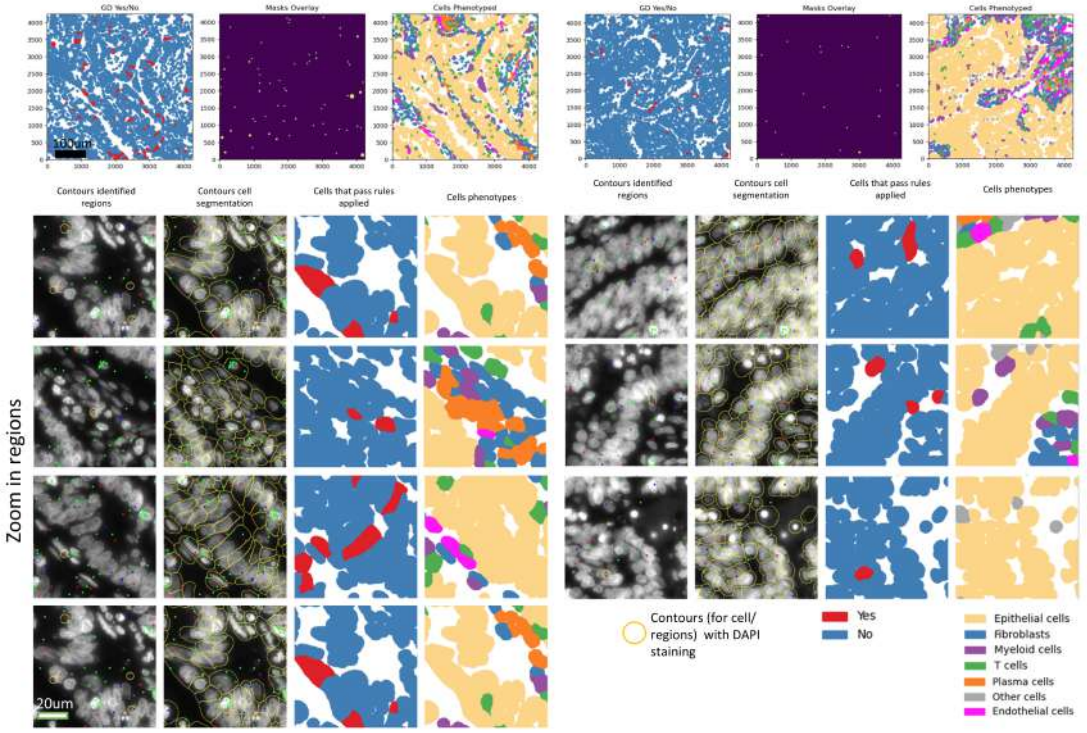


Figure S5.2: Comparison of GRIDGEN population-specific masks and cell segmentation in two example CosMx samples (left and right). Top row: From left to right — (1) Cell segmentation with classification of cells as $\gamma\delta$ T cells based on GRIDGEN-defined rules: presence of at least one transcript of both *TRGC* and *TRDC*, and the sum of *TRGC* and *TRDC* greater than the sum of *TRAC* and *TRBC*; (2) GRIDGEN-defined regions putatively enriched for $\gamma\delta$ T cells; (3) Cells colored by phenotype as inferred from clustering. Bottom row: Zoom-in on $80\ \mu\text{m}$ regions. From left to right — (1) DAPI signal overlaid with *TRGC*, *TRDC*, *TRAC*, and *TRBC* transcript locations, with contours identified via a K-D Tree approach (surrounded by simple circles); (2) Same image with cell segmentation borders; (3) Cells classified as $\gamma\delta$ T cells using the same rules; (4) Cells colored by phenotype. Applying the same rules to segmented cells is challenging due to the spatial distribution of transcripts. In many cases, transcripts such as *TRGC* and *TRDC* are spread across large cells, producing false positives. Conversely, closely located *TRGC* and *TRDC* transcripts may be incorrectly assigned to separate cells, fragmenting a true $\gamma\delta$ T-cell region. Additionally, GRIDGEN-defined contours often correspond to cells misclassified as non-immune despite nearby high-density T-cell-related transcripts.

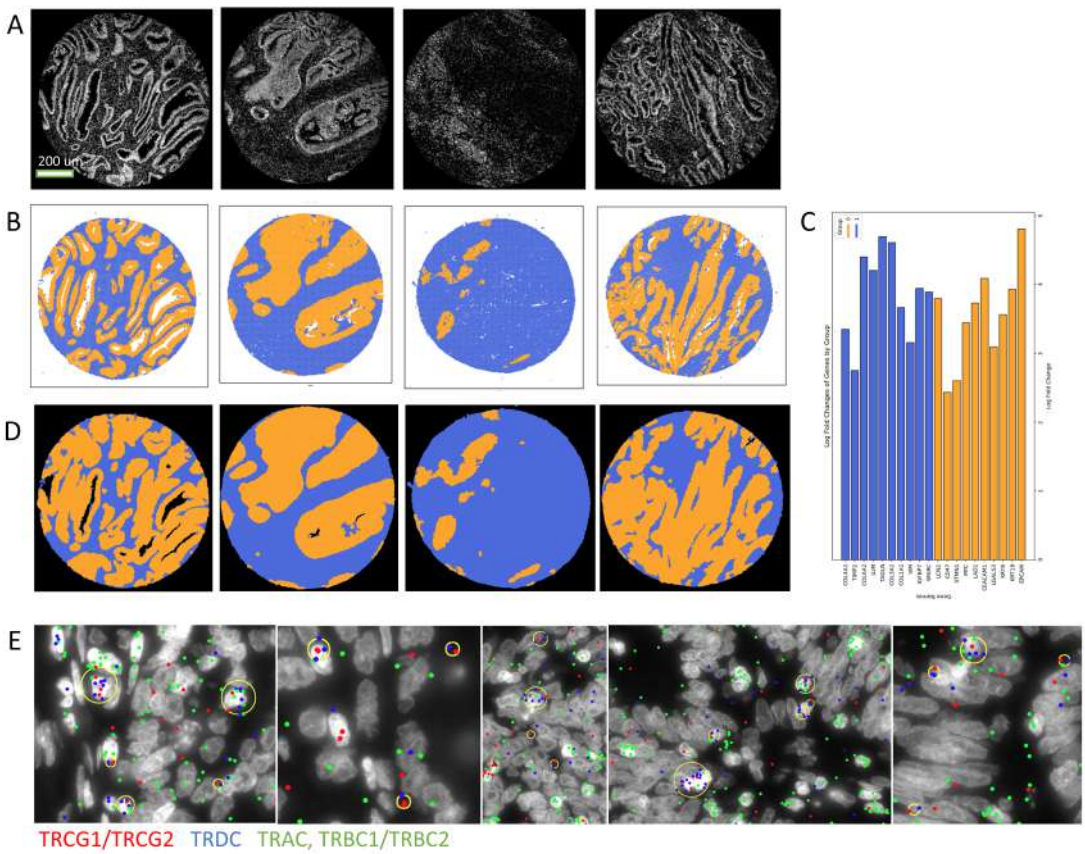


Figure S5.3: Alternative ways of masking with GRIDGEN.(A) Xenium DAPI references of 4 CRC samples.(B) Tumour and Stroma masks constructed with a SOM approach, where grids of 10*10 pixels are classified in a SOM with 2 units, that seem to represent Stroma and Tumour areas.(C) Differential gene expression of the grids generated in B showing a clear distinction between the 2 classes separated by SOM supporting that they are Stroma and Tumour. (D) Tumour and Stroma masks deriving contours based on number of neighbours of each point, calculated with a K-D Tree and a radius of $10\ \mu$m. (E) identify points of *TRGC* and *TRDC* that dist $7\ \mu$m or less apart surrounding by a circle. The circles are discarded if they have more *TRAC* or *TRBC* present. These contours may be used to overlap with cell segmentation to identify $\gamma\delta$ T cells, or further refined and used as regions.

6

Conclusion and Future Work

This final chapter presents the overall conclusions drawn from the research conducted in this thesis, along with perspectives on future directions that may address new challenges and questions emerging from this work.

The overarching aim of the thesis was to develop robust methods and frameworks for the analysis of spatial omics data, with a focus on enabling scalable, reproducible, and insightful biological interpretation. While the core of the work does not rely solely on deep learning, it integrates DL concepts and tools where appropriate—particularly in enhancing data pre-processing and segmentation—demonstrating how classical and modern approaches can be combined effectively.

Although this research marks a significant step forward in the spatial omics field, it also highlights numerous avenues for further exploration, especially as new technologies and data types continue to emerge. The following sections summarize the main contributions of the thesis and propose directions for future work that could extend the impact and applicability of this research.

6.1 Main contributions

One of the key contributions of this thesis lies in the development of innovative computational tools and frameworks tailored for the analysis of high-dimensional spatial omics data.

Chapter 3 introduced PENGUIN, a denoising pipeline specifically designed for Imaging Mass Cytometry data. Through a combination of scaling, thresholding, and percentile filtering, PENGUIN delivers clear, high-quality images with minimal user intervention. It was benchmarked it against classical image filters, deep learning-based solutions, and IMC-specific approaches, demonstrating its superior performance in preserving signal integrity while minimizing noise. The implementation is complemented by an intuitive GUI within Jupyter notebooks, enabling researchers to fine-tune thresholds and percentiles for each marker and streamline batch image processing. Unlike many existing solutions, PENGUIN requires no manual annotations or model training and is both reproducible and computationally efficient. The framework was tested on CRC IMC data, as well as other tissue types, and immunofluorescence data, representing a robust, user-friendly tool for multiplexed imaging data analysis, —addressing a significant bottleneck in the field: managing signal-to-noise ratios in large antibody panels.

Chapter 4 focused on the crucial task of cell segmentation in spatial proteomics, a step that strongly influences downstream analyses and biological interpretation. Using CRC IMC data, we performed an extensive comparison of three commonly used segmentation tools—CellProfiler, Mesmer, and Cellpose—representing a spectrum from traditional image processing to state-of-the-art deep learning methods. Beyond default settings, we evaluated both standard and sequential segmentation strategies, the latter allowing for cell-type-specific parameter optimization. Our analysis highlighted key differences in segmentation morphology, cell count, and marker distribution, showing that each method has distinct advantages depending on tissue structure and cell type. Importantly, we demonstrated that sequential segmentation improves the identification of small or rare cell types, such as infiltrating immune cells, albeit at increased computational cost. To support further work in this area, we also released a curated dataset containing over 1,900 manually annotated ROIs spanning multiple cell lineages, which can serve as a valuable benchmark for future model development.

Together, the contributions of Chapters 3 and 4 lay a solid foundation for unbiased pre-processing of spatial proteomics data. By reducing the need for expert intervention in denoising and segmentation, these tools make it more feasible to start analyses from raw data, crucial for minimizing biases introduced by pre-processed or inconsistently annotated datasets. These foundations enable more robust applications of deep learning models on large spatial proteomics cohorts, opening new opportunities for disease modelling and biomarker discovery.

Chapter 5 shifted focus toward subcellular spatial transcriptomics, an area still in early stages of exploration. While many existing methods are centered on cell segmentation, GRIDGEN was developed as a flexible, cell-free framework to define tissue compartments based on transcript density. Current spatial transcriptomics analysis often relies heavily on segmented cells to derive features, but in some cases, particularly where segmentation is unreliable or unnecessary, an alternative approach can provide clearer insight. GRIDGEN identifies meaningful biological regions, such as tumour-stroma interfaces or transcript-enriched microenvironments, without requiring cell boundaries. It offers multiple contouring strategies and mask generation tools and supports downstream transcript quantification and morphological analysis. GRIDGEN's modular, reproducible structure makes it applicable across diverse spatial omics datasets, including high-resolution methods such as CosMx and Xenium.

All developed tools in this thesis are open-source and freely available to the scientific community. PENGUIN and GRIDGEN are released as installable Python packages, with detailed documentation. The segmentation benchmarking framework from Chapter 4, along with the curated IMC annotation dataset, is also provided to support reproducibility and further research. It is the author's hope that these contributions will not only facilitate the analysis of current datasets but also serve as a springboard for new methods, collaborative projects, and deeper insights into spatial biology.

Finally, this thesis includes a review (with an interactive visualization) of computational methods applied for spatial omics data, focusing on spatial transcriptomics and proteomics.

6.2 Publications

- Chapter 3 was published in a peer-reviewed scientific journal: Sequeira, A. M., Ijsselsteijn, M. E., Rocha, M., & de Miranda, N. F. (2024). PENGUIN: a rapid and efficient image preprocessing tool for multiplexed spatial proteomics. Computational and Structural Biotechnology Journal, 23, 3920-3928.

- At the time of writing, Chapter 4 is being prepared for submission to a peer-reviewed scientific journal.

- Chapter 5 was presented as an oral communication at the EMBO workshop "The Spatial Biology of Cancer", 1-3 September 2024, London. This chapter is being prepared for submission to a peer-reviewed scientific journal.

Additionally, the author contributed to the following peer-reviewed scientific publications during the PhD. Related to machine and deep learning applied to protein sequences:

- Sequeira, A. M., Rocha, M., & Lousa, D. (2025). Machine and deep learning to predict viral fusion peptides. Computational and Structural Biotechnology Journal, 27, 692-704.

- Sequeira, A. M., Lousa, D., & Rocha, M. (2022). ProPythia: A Python package for protein classification based on machine and deep learning. Neurocomputing, 484, 172-182.

- Sequeira, A. M., Gomes, I., & Rocha, M. (2023, August). Word embeddings for protein sequence analysis. In 2023 IEEE Conference on Computational Intelligence in Bioinformatics and Computational Biology (CIBCB) (pp. 1-8). IEEE.

- Sequeira, A. M., & Rocha, M. (2021, August). Recurrent deep neural networks for enzyme functional annotation. In International Conference on Practical Applications of Computational Biology & Bioinformatics (pp. 62-73). Cham: Springer International Publishing.

- Moreira, P., Sequeira, A. M., Pereira, S., Rodrigues, R., Rocha, M., & Lousa, D. (2021). ViralFP: a web application of viral fusion proteins. Frontiers in Medical Technology, 3, 722392

And related to machine and deep learning applied to histopathological images:

- Doeleman, T., Brussee, S., Hondelink, L. M., Westerbeek, D. W., Sequeira, A. M., Valkema, P. A., ... & Schrader, A. M. (2024). Deep Learning–Based Classification of Early-Stage Mycosis Fungoides and Benign Inflammatory Dermatoses on H&E-Stained Whole-Slide Images: A Retrospective, Proof-of-Concept Study. Journal of Investigative Dermatology.

6.3 Perspectives for future work

While the work presented in this thesis already represents a meaningful contribution to the field by introducing robust tools and methodologies for spatial omics analysis, it also opens up new avenues for exploration. Future efforts, both to refine and expand the current framework and to address broader challenges within the field, hold great potential to further advance research and practical applications. Besides the improvements in the frameworks already discussed in the previous chapters, future work would include:

Integration of Spatial Omics with Deep Learning Models: A natural progression of this work involves exploring graph-based deep learning models to extract meaningful insights from spatial omics data, particularly for disease understanding and characterization. Although this was one of the primary objectives outlined in this thesis, the volume of publicly available spatial omics data has only recently begun to reach the scale necessary to support data-hungry deep learning approaches. The contributions made in Chapters 3 and 4, specifically around denoising and cell segmentation on spatial proteomics, lay an important foundation. They allow for unbiased pre-processing of raw spatial data, enabling cell segmentation workflows that do not rely on labour-intensive expert annotations. This makes it increasingly feasible to start from raw data — a crucial step to avoid introducing bias from pre-processed or over-annotated datasets.

With these tools in place and the emergence of large-scale spatial cohorts, graph neural networks emerge as a particularly suitable modelling approach. They can naturally encode cell-cell relationships within tissue contexts by using spatial proximity and expression profiles as part of the graph structure. These models could not only extract functional insights, but also help identify disease-specific spatial patterns and cellular interaction signatures, particularly relevant in heterogeneous environments, like the tumour microenvironment. Furthermore, if these models reveal interesting or clinically relevant features, they could be extended to learn from routine histopathology images, which are inexpensive, scalable, and widely available in clinical settings. Such models could bridge high-resolution molecular insights and accessible diagnostic tools, paving the way for multimodal pathology. This direction becomes even more compelling with the emergence of multi-dimensional datasets that integrate spatial proteomics, transcriptomics, and more recently, spatial metabolomics. The ability to merge and model these rich modalities has the potential to be transformative, enabling a systems-level view of tissue biology. Deep models trained on such integrative data could not only generate predictive biomarkers, but also provide mechanistic insights, for example into carcinogenesis, cellular differentiation, or immune evasion.

Living Reviews and AI-Assisted Curation in Spatial Omics: Spatial omics is a rapidly evolving field, with a constant influx of new computational methods and analytical tools. As a relatively young discipline, it has not yet converged on a universally accepted analysis pipeline, which presents both opportunities and challenges. Numerous methodological reviews have been published in recent years, yet in the author's view, many of these become quickly outdated and of limited practical use, as they are static, purely theoretical summaries that cannot keep pace with the field's momentum. A promising direction would be the creation of a living methods review—a dynamic, regularly updated resource that automatically

compiles and categorizes new spatial omics analysis methods. Such a system could be powered by a combination of manually annotated seed papers and AI-driven literature mining, incorporating data from paper repositories and method sections. Each method entry would ideally include references to the tools it compares to, the datasets it was tested on, and the specific steps of spatial analysis it addresses.

With the emergence of Large Language Models (LLM) capable of summarizing and synthesizing scientific literature (e.g., BioGPT [279], SciBERT [280], biochatter [281]), we are at the cusp of a paradigm shift in how reviews are written and consumed. Particularly for methodological work, where the pace of publication is high, automated or semi-automated reviews could ensure that researchers stay informed about the most current tools and best practices. Beyond being a useful catalogue, such a living database would also allow researchers to track which methods are being most widely adopted, and what types of data are commonly used in benchmarking.

This concept could be further expanded to include a spatial omics dataset index—not a storage repository, but a searchable metadata database that catalogues existing datasets. It would include critical, but often poorly indexed information, such as transcript or protein panels used. This would greatly enhance the ability to identify and cross-reference relevant cohorts, facilitating data reuse and integration for specific scientific questions. Crucially, such a database could help determine which cohorts are suitable for merging, an essential step toward enabling AI-driven method development and benchmarking. As more datasets become publicly available, the ability to combine them meaningfully depends on understanding their design and compatibility at a granular level. Importantly, this underscores the need for integration at the raw data level, rather than using derived data (e.g., cell-by-feature matrices), which are often processed differently across pipelines and can introduce significant bias or incompatibility. A shared emphasis on raw-level harmonization would be a pivotal step in advancing reproducible and generalizable AI applications in spatial omics.

The Study of Cell-Cell Interactions and Functional Understanding of Signaling Networks: The study of cell-cell interactions and functional pathways has been a longstanding focus in molecular and cellular biology. The emergence of spatial omics technologies now provides an unprecedented opportunity to incorporate spatial proximity as a key parameter in these analyses. This shift enables a more nuanced understanding of whether potentially interacting cells are not only expressing relevant ligands and receptors, but also physically close enough for such communication to be biologically plausible. While some efforts have already been made to explore spatially informed interactions, many current approaches still rely on simplified models that overlook the complexity of cellular microenvironments. With the growing availability of curated databases for signalling pathways, transcription factor activity, and ligand-receptor interactions, there is immense potential to move beyond basic co-expression models. Future work could focus on how the expression profile of one cell may actively influence or induce transcriptional changes in neighbouring cells, especially within complex environments such as the tumour microenvironment, where spatial context is often crucial for understanding cellular behaviour and immune evasion mechanisms.

The Untapped Potential of Subcellular Resolution in Spatial Transcriptomics: Several cutting-edge spatial transcriptomics technologies, such as CosMx and Xenium, offer subcellular resolution, enabling the potential to distinguish between transcripts located in the nucleus versus the cytoplasm, or

even observe spatial patterns of transcript polarization within individual cells. Despite this unprecedented level of detail, this subcellular information remains largely unexplored in most analyses. Only a few recent studies have begun to scratch the surface of what this resolution could reveal. In most current applications, the subcellular data is simply aggregated at the cell level, often discarding the spatial localization within the cell that could provide critical insights into transcript dynamics. While the resolution is not yet at the scale of dedicated organelle-level imaging studies, it is sufficient to begin exploring spatial features, such as nuclear retention, asymmetric RNA localization and local translation zones, all of which are known to play significant roles in processes like cell polarity, development, and disease progression. The author regards this as an exciting and underutilized dimension of spatial data analysis. As datasets with subcellular resolution become more available, future work could focus on developing tools and frameworks that truly leverage this level of detail, potentially opening a new chapter in our understanding of intracellular transcriptomics.

Others: Beyond the directions discussed above, additional future work could focus on non-cellular components, such as the extracellular matrix or vasculature, which play critical roles in tissue organization and function but are often overlooked. Another exciting avenue is the exploration of 3D spatial omics data, as relying solely on 2D sections may distort spatial relationships and limit biological interpretation. Finally, and particular relevant for CRC, is the exploration of spatial microbiome. Understanding how microbial communities are distributed in tissue, and how they spatially interact with host cells, could open up entirely new avenues for studying tumorigenesis, immune responses, and therapeutic strategies.

While this thesis does not answer all the questions it raises, it contributes valuable tools and perspectives to an emerging area of science.There is still much to explore, but the progress made so far gives reason for optimism about what lies ahead. As technologies evolve, integrating depth and structure into spatial analysis will become increasingly important for achieving a more accurate and comprehensive understanding of tissue biology. With continued development, these tools and insights can transition from academic exploration to clinical relevance, guiding diagnostics, therapeutic strategies, and personalized medicine.

This thesis represents not only a scientific contribution, but also a reflection of a deep curiosity about how cells organize, interact, and misbehave in disease. Through analytical frameworks, it aims to advance efforts to decode these complex biological processes. The open questions left unanswered leave an exciting invitation to explore, challenge, and expand the field in directions we are only beginning to imagine.

Bibliography

[1] D. Hanahan. “Hallmarks of Cancer: New Dimensions”. In: *Cancer Discovery* 12.1 (2022), pp. 31–46. doi: 10.1158/2159-8290.CD-21-1059.

[2] D. Hanahan and R. A. Weinberg. “Hallmarks of Cancer: The Next Generation”. In: *Cell* 144.5 (2011), pp. 646–674. doi: 10.1016/j.cell.2011.02.013.

[3] R. D. Schreiber, L. J. Old, and M. J. Smyth. “Cancer Immunoediting: Integrating Immunity’s Roles in Cancer Suppression and Promotion”. In: *Science* 331.6024 (2011), pp. 1565–1570. doi: 10.1126/science.1203486.

[4] N. L. De Vries et al. “Unraveling the Complexity of the Cancer Microenvironment With Multidimensional Genomic and Cytometric Technologies”. In: *Frontiers in Oncology* 10 (2020), p. 1254. doi: 10.3389/fonc.2020.01254.

[5] D. Bressan, G. Battistoni, and G. J. Hannon. “The dawn of spatial omics”. In: *Science* 381.6657 (2023), eabq4964. doi: 10.1126/science.abq4964.

[6] A. Bollhagen and B. Bodenmiller. “Highly Multiplexed Tissue Imaging in Precision Oncology and Translational Cancer Research”. In: *Cancer Discovery* 14.11 (2024), pp. 2071–2088. doi: 10.1158/2159-8290.CD-23-1165.

[7] J. Chen et al. “Spatial landscapes of cancers: insights and opportunities”. In: *Nature Reviews Clinical Oncology* 21.9 (2024), pp. 660–674. doi: 10.1038/s41571-024-00926-7.

[8] Y. Goltsev et al. “Deep Profiling of Mouse Splenic Architecture with CODEX Multiplexed Imaging”. In: *Cell* 174.4 (2018), 968–981.e15. doi: 10.1016/j.cell.2018.07.010.

[9] M. Angelo et al. “Multiplexed ion beam imaging of human breast tumors”. In: *Nature Medicine* 20.4 (2014), pp. 436–442. doi: 10.1038/nm.3488.

[10] C. Giesen et al. “Highly multiplexed imaging of tumor tissues with subcellular resolution by mass cytometry”. In: *Nature Methods* 11.4 (2014), pp. 417–422. doi: 10.1038/nmeth.2869.

[11] “Method of the Year 2024: spatial proteomics”. In: *Nature Methods* 21.12 (2024), pp. 2195–2196. doi: 10.1038/s41592-024-02565-3.

[12] A. Janesick et al. “High resolution mapping of the tumor microenvironment using integrated single-cell, spatial and in situ analysis”. In: *Nature Communications* 14.1 (2023), p. 8353. doi: 10.1038/s41467-023-43458-x.

[13] S. He et al. “High-plex imaging of RNA and proteins at subcellular resolution in fixed tissue by spatial molecular imaging”. In: *Nature Biotechnology* 40.12 (2022), pp. 1794–1806. doi: 10.1038/s41587-022-01483-z.

[14] V. Marx. “Method of the Year: spatially resolved transcriptomics”. In: *Nature Methods* 18.1 (2021), pp. 9–14. doi: 10.1038/s41592-020-01033-y.

[15] G. Palla et al. “Spatial components of molecular tissue biology”. In: *Nature Biotechnology* 40.3 (2022), pp. 308–318. doi: 10.1038/s41587-021-01182-1.

[16] M. Allam, S. Cai, and A. F. Coskun. “Multiplex bioimaging of single-cell spatial profiles for precision cancer diagnostics and therapeutics”. In: *npj Precision Oncology* 4.1 (2020), p. 11. doi: 10.1038/s41698-020-0114-1.

[17] B. Velten and O. Stegle. “Principles and challenges of modeling temporal and spatial omics data”. In: *Nature Methods* 20.10 (2023), pp. 1462–1474. doi: 10.1038/s41592-023-01992-y.

[18] Y. LeCun, Y. Bengio, and G. Hinton. “Deep learning”. In: *Nature* 521.7553 (2015), pp. 436–444. doi: 10.1038/nature14539.

[19] G. F. Gao et al. “Before and After: Comparison of Legacy and Harmonized TCGA Genomic Data Commons’ Data”. In: *Cell Systems* 9.1 (2019). doi: 10.1016/j.cels.2019.06.006.

[20] S.-K. Low, H. Zembutsu, and Y. Nakamura. “Breast cancer: The translation of big genomic data to cancer precision medicine”. In: *Cancer Science* 109.3 (2018), pp. 497–506. doi: 10.1111/cas.13463.

[21] Y. Sun et al. “Identification of 12 cancer types through genome deep learning”. In: *Scientific Reports* 9.1 (2019), p. 17256. doi: 10.1038/s41598-019-53989-3.

[22] S. Yousefi et al. “Predicting clinical outcomes from large scale cancer genomic profiles with deep survival models”. In: *Scientific Reports* 7.1 (2017). doi: 10.1038/s41598-017-11817-6.

[23] D. Baptista, P. G. Ferreira, and M. Rocha. “Deep learning for drug response prediction in cancer”. In: *Briefings in Bioinformatics* 22.1 (2021), pp. 360–379. doi: 10.1093/bib/bbz171.

[24] J. Saltz et al. “Spatial Organization and Molecular Correlation of Tumor-Infiltrating Lymphocytes Using Deep Learning on Pathology Images”. In: *Cell Reports* 23.1 (2018), 181–193.e7. doi: 10.1016/j.celrep.2018.03.086.

[25] S.-y. Jing et al. “Quantifying and interpreting biologically meaningful spatial signatures within tumor microenvironments”. In: *npj Precision Oncology* 9.1 (2025), p. 68. doi: 10.1038/s41698-025-00857-1.

[26] A. L. Samuel. “Some studies in machine learning using the game of checkers”. In: *IBM Journal of Research and Development* (2000). doi: 10.1147/rd.441.0206.

[27] A. Smola and S. Vishwanathan. *Introduction to machine learning*. Cambridge University Press 2008, 2008. doi: 10.1007/978-1-62703-748-8_7.

[28] M. Awad and R. Khanna. *Efficient Learning Machines*. Apress Media, 2015.

[29] F. Chollet and F. Chollet. *Deep learning with Python*. Simon and Schuster, 2021.

[30] I. Goodfellow et al. *Deep learning*. Vol. 1. 2. MIT press Cambridge, 2016.

[31] D. Chicco. "Ten quick tips for machine learning in computational biology". In: *BioData Mining* 10.1 (2017), pp. 1–17. doi: 10.1186/s13040-017-0155-3. arXiv: arXiv:1011.1669v3.

[32] C. S. Greene et al. "Big data bioinformatics". In: *Journal of Cellular Physiology* 229.12 (2014), pp. 1896–1900. doi: 10.1002/jcp.24662. arXiv: 15334406.

[33] A. C. Muller and S. Guido. *Introduction to Machine Learning with Python: A guide for data scientists*. O'Reilly Media, 2017. doi: 10.1017/CBO9781107415324.004.

[34] T. Kohonen. "The self-organizing map". In: *Proceedings of the IEEE* 78.9 (1990), pp. 1464–1480. doi: 10.1109/5.58325.

[35] D. D. Lee and H. S. Seung. "Learning the parts of objects by non-negative matrix factorization". In: *nature* 401.6755 (1999), pp. 788–791.

[36] L. Van der Maaten and G. Hinton. "Visualizing data using t-SNE." In: *Journal of machine learning research* 9.11 (2008).

[37] L. McInnes, J. Healy, and J. Melville. "Umap: Uniform manifold approximation and projection for dimension reduction". In: *arXiv preprint arXiv:1802.03426* (2018).

[38] I. Guyon and A. Elisseef. "An Introduction to Feature Extraction". In: 2006, pp. 1–24. doi: 10.1007/978-3-540-35488-8_1. arXiv: arXiv:1011.1669v3.

[39] "Feature selection in machine learning: A new perspective". In: *Neurocomputing* 300 (2018), pp. 70–79. doi: 10.1016/j.neucom.2017.11.077.

[40] K. T. Butler et al. "Machine learning for molecular and materials science". In: *Nature* 559.7715 (2018), pp. 547–555. doi: 10.1038/s41586-018-0337-2.

[41] S. Piramuthu. "Evaluating Feature Selection Methods for Learning in Data Mining Applications". In: 00.c (1998).

[42] Y. Saeys, I. Inza, and P. Larrañaga. "A review of feature selection techniques in bioinformatics". In: *Bioinformatics* 23.19 (2007), pp. 2507–2517. doi: 10.1093/bioinformatics/btm344.

[43] M. Bowles. *Machine Learning in python: essential techniques for predictive analysis*. 2015.

[44] S. Raschka. *Python Machine learning*. Packt Publishing, 2015.

[45] S. Boughorbel, F. Jarray, and M. El-Anbari. "Optimal classifier for imbalanced data using Matthews Correlation Coefficient metric". In: *PLoS ONE* 12.6 (2017), pp. 1–17. doi: 10.1371/journal.pone.0177678.

[46] J. Davis and M. Goadrich. “The relationship between Precision-Recall and ROC curves”. In: (2016), pp. 233–240. arXiv: 1609.07195.

[47] F. Pedregosa et al. “Scikit-learn: Machine Learning in Python”. In: *Journal of Machine Learning Research* 12 (2011), pp. 2825–2830.

[48] L. Breiman. “Bagging predictors”. In: *Machine learning* 24 (1996), pp. 123–140.

[49] J. Dean. *Big data, data mining, and machine learning: Value creation for business leaders and practitioners*. Wiley, 2014.

[50] L. Breiman. “Random forests”. In: *Machine learning* 45 (2001), pp. 5–32.

[51] T. Chen and C. Guestrin. “Xgboost: A scalable tree boosting system”. In: *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*. 2016, pp. 785–794.

[52] D. E. Rumelhart, G. E. Hinton, and R. J. Williams. “Learning representations by back-propagating errors”. In: *nature* 323.6088 (1986), pp. 533–536.

[53] D. P. Kingma and J. Ba. “Adam: A method for stochastic optimization”. In: *arXiv preprint arXiv:1412.6980* (2014).

[54] A. Paszke et al. “PyTorch: An Imperative Style, High-Performance Deep Learning Library”. In: *arXiv preprint arXiv:1912.01703* (2019). url: https://arxiv.org/abs/1912.01703.

[55] M. Abadi et al. *TensorFlow: Large-Scale Machine Learning on Heterogeneous Distributed Systems*. Version Number: 2. 2016. doi: 10.48550/ARXIV.1603.04467.

[56] S. Ruder. “An overview of gradient descent optimization algorithms”. In: *arXiv preprint arXiv:1609.04747* (2016).

[57] N. Srivastava et al. “Dropout: a simple way to prevent neural networks from overfitting”. In: *The journal of machine learning research* 15.1 (2014), pp. 1929–1958.

[58] L. Prechelt. “Early stopping-but when?” In: *Neural Networks: Tricks of the trade*. Springer, 2002, pp. 55–69.

[59] C. Shorten and T. M. Khoshgoftaar. “A survey on image data augmentation for deep learning”. In: *Journal of big data* 6.1 (2019), pp. 1–48.

[60] F. Zhuang et al. “A Comprehensive Survey on Transfer Learning”. In: *arXiv preprint arXiv:1911.02685* (2019). doi: 10.48550/arXiv.1911.02685.

[61] A. Y. Ng. “Feature selection, L 1 vs. L 2 regularization, and rotational invariance”. In: *Proceedings of the twenty-first international conference on Machine learning*. 2004, p. 78.

[62] Y. LeCun et al. “Gradient-based learning applied to document recognition”. In: *Proceedings of the IEEE* 86.11 (1998), pp. 2278–2324.

[63] I. Sutskever, O. Vinyals, and Q. V. Le. “Sequence to sequence learning with neural networks”. In: *Advances in neural information processing systems* 27 (2014).

[64] R. Pascanu, T. Mikolov, and Y. Bengio. “On the difficulty of training recurrent neural networks”. In: *International conference on machine learning*. Pmlr. 2013, pp. 1310–1318.

[65] S. Hochreiter and J. Schmidhuber. “Long Short-Term Memory”. In: *Neural Computation* 9.8 (1997), pp. 1735–1780. doi: 10.1162/neco.1997.9.8.1735.

[66] K. Cho et al. “Learning phrase representations using RNN encoder-decoder for statistical machine translation”. In: *arXiv preprint arXiv:1406.1078* (2014).

[67] D. Bahdanau, K. Cho, and Y. Bengio. “Neural machine translation by jointly learning to align and translate”. In: *arXiv preprint arXiv:1409.0473* (2014).

[68] G. E. Hinton and R. R. Salakhutdinov. “Reducing the dimensionality of data with neural networks”. In: *science* 313.5786 (2006), pp. 504–507.

[69] D. P. Kingma, M. Welling, et al. *Auto-encoding variational bayes*. 2013.

[70] A. Makhzani et al. “Adversarial autoencoders”. In: *arXiv preprint arXiv:1511.05644* (2015).

[71] Z. Wu et al. “A comprehensive survey on graph neural networks”. In: *IEEE transactions on neural networks and learning systems* 32.1 (2020), pp. 4–24.

[72] T. N. Kipf and M. Welling. “Semi-supervised classification with graph convolutional networks”. In: *arXiv preprint arXiv:1609.02907* (2016).

[73] X.-M. Zhang et al. “Graph neural networks and their current applications in bioinformatics”. In: *Frontiers in genetics* 12 (2021), p. 690049.

[74] P. Pati et al. “Hierarchical graph representations in digital pathology”. In: *Medical image analysis* 75 (2022), p. 102264.

[75] G. Jaume et al. “Histocartography: A toolkit for graph analytics in digital pathology”. In: *MICCAI Workshop on Computational Pathology*. PMLR. 2021, pp. 117–128.

[76] P. Chandak, K. Huang, and M. Zitnik. “Building a knowledge graph to enable precision medicine”. In: *Scientific Data* 10.1 (2023), p. 67. doi: 10.1038/s41597-023-01960-3.

[77] W. Hamilton, Z. Ying, and J. Leskovec. “Inductive representation learning on large graphs”. In: *Advances in neural information processing systems* 30 (2017).

[78] P. Velickovic et al. “Graph attention networks”. In: *stat* 1050.20 (2017), pp. 10–48550.

[79] Z. Ying et al. “Gnnexplainer: Generating explanations for graph neural networks”. In: *Advances in neural information processing systems* 32 (2019).

[80] A. Longa et al. “Explaining the explainers in graph neural networks: a comparative study”. In: *ACM Computing Surveys* 57.5 (2025), pp. 1–37.

[81] X. Chen, S. Xie, and K. He. “A Simple Framework for Contrastive Learning of Visual Representations”. In: *Proceedings of the International Conference on Machine Learning (ICML)*. 2020.

[82] A. Krull, T.-O. Buchholz, and F. Jug. “Noise2Void - Learning Denoising From Single Noisy Images”. In: *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. Long Beach, CA, USA: IEEE, 2019, pp. 2124–2132. doi: 10.1109/CVPR.2019.00223.

[83] H. Sung et al. “Global Cancer Statistics 2020: GLOBOCAN Estimates of Incidence and Mortality Worldwide for 36 Cancers in 185 Countries”. In: *CA: A Cancer Journal for Clinicians* 71.3 (2021), pp. 209–249. doi: 10.3322/caac.21660.

[84] W. H. Organization. *Cancer Fact Sheet*. 2021. url: https://gco.iarc.fr/today/en.

[85] J. Ferlay et al. “Cancer statistics for the year 2020: An overview”. In: *International Journal of Cancer* 149.4 (2021), pp. 778–789. doi: 10.1002/ijc.33588.

[86] I. Dagogo-Jack and A. T. Shaw. “Tumour heterogeneity and resistance to cancer therapies”. In: *Nature Reviews Clinical Oncology* 15.2 (2018). doi: 10.1038/nrclinonc.2017.166.

[87] P. L. Bedard et al. “Tumour heterogeneity in the clinic”. In: *Nature* 501.7467 (2013), pp. 355–364. doi: 10.1038/nature12627.

[88] N. M. Anderson and M. C. Simon. “The tumor microenvironment”. In: *Current Biology* 30.16 (2020), R921–R925. doi: 10.1016/j.cub.2020.06.081.

[89] O. Farc and V. Cristea. “An overview of the tumor microenvironment, from cells to complex networks (Review)”. In: *Experimental and Therapeutic Medicine* 21.1 (2020), p. 96. doi: 10.3892/etm.2020.9528.

[90] W. H. Fridman et al. “The immune contexture in cancer prognosis and treatment”. In: *Nature Reviews Clinical Oncology* 14.12 (2017), pp. 717–734. doi: 10.1038/nrclinonc.2017.101.

[91] J. Roelands et al. “Immunogenomic Classification of Colorectal Cancer and Therapeutic Implications”. In: *International Journal of Molecular Sciences* 18.10 (2017), p. 2229. doi: 10.3390/ijms18102229.

[92] F. Pagès et al. “International validation of the consensus Immunoscore for the classification of colon cancer: a prognostic and accuracy study”. In: *The Lancet* 391.10135 (2018), pp. 2128–2139. doi: 10.1016/S0140-6736(18)30789-X.

[93] J. Galon et al. “Type, Density, and Location of Immune Cells Within Human Colorectal Tumors Predict Clinical Outcome”. In: *Science* 313.5795 (2006), pp. 1960–1964. doi: 10.1126/science.1129139.

[94] D. Yang et al. “Cancer-associated fibroblasts: from basic science to anticancer therapy”. In: *Experimental & Molecular Medicine* 55.7 (2023), pp. 1322–1332. doi: 10.1038/s12276-023-01013-0.

[95] E. Sahai et al. “A framework for advancing our understanding of cancer-associated fibroblasts”. In: *Nature Reviews Cancer* 20.3 (2020), pp. 174–186. doi: 10.1038/s41568-019-0238-1.

[96] The Cancer Genome Atlas Research Network et al. “The Cancer Genome Atlas Pan-Cancer analysis project”. In: *Nature Genetics* 45.10 (2013), pp. 1113–1120. doi: 10.1038/ng.2764.

[97] K. A. Hoadley et al. “Cell-of-Origin Patterns Dominate the Molecular Classification of 10,000 Tumors from 33 Types of Cancer”. In: *Cell* 173.2 (2018), 291–304.e6. doi: 10.1016/j.cell.2018.03.022.

[98] The Cancer Genome Atlas Network. “Comprehensive molecular characterization of human colon and rectal cancer”. In: *Nature* 487.7407 (2012), pp. 330–337. doi: 10.1038/nature11252.

[99] L. Nunes et al. “Prognostic genome and transcriptome signatures in colorectal cancers”. In: *Nature* 633.8028 (2024), pp. 137–146. doi: 10.1038/s41586-024-07769-3.

[100] Y. Li. “Modern epigenetics methods in biological research”. In: *Methods (San Diego, Calif.)* 187 (2021), pp. 104–113. doi: 10.1016/j.ymeth.2020.06.022.

[101] X. Chen et al. “Mapping epigenetic modifications by sequencing technologies”. In: *Cell Death & Differentiation* 32.1 (2025), pp. 56–65. doi: 10.1038/s41418-023-01213-1.

[102] S. B. Baylin and P. A. Jones. “A decade of exploring the cancer epigenome — biological and translational implications”. In: *Nature Reviews Cancer* 11.10 (2011), pp. 726–734. doi: 10.1038/nrc3130.

[103] M. Uhlén et al. “Tissue-based map of the human proteome”. In: *Science* 347.6220 (2015), p. 1260419. doi: 10.1126/science.1260419.

[104] K. Suhre, M. I. McCarthy, and J. M. Schwenk. “Genetics meets proteomics: perspectives for large population-based studies”. In: *Nature Reviews Genetics* 22.1 (2021), pp. 19–37. doi: 10.1038/s41576-020-0268-2.

[105] R. Aebersold and M. Mann. “Mass-spectrometric exploration of proteome structure and function”. In: *Nature* 537.7620 (2016), pp. 347–355. doi: 10.1038/nature19949.

[106] D. R. Schmidt et al. “Metabolomics in cancer research and emerging applications in clinical oncology”. In: *CA:A Cancer Journal for Clinicians* 71.4 (2021), pp. 333–358. doi: 10.3322/caac.21670.

[107] S. Aldridge and S. A. Teichmann. “Single cell transcriptomics comes of age”. In: *Nature Communications* 11.1 (2020), p. 4307. doi: 10.1038/s41467-020-18158-5.

[108] G. S. Kinker et al. “Pan-cancer single-cell RNA-seq identifies recurring programs of cellular heterogeneity”. In: *Nature Genetics* 52.11 (2020), pp. 1208–1218. doi: 10.1038/s41588-020-00726-6.

[109] M. S. Mansuri, K. Williams, and A. C. Nairn. “Uncovering biology by single-cell proteomics”. In: *Communications Biology* 6.1 (2023), p. 381. doi: 10.1038/s42003-023-04635-2.

[110] “Single-cell proteomics: challenges and prospects”. In: *Nature Methods* 20.3 (2023), pp. 317–318. doi: 10.1038/s41592-023-01828-9.

[111] S. M. Setayesh et al. “Targeted single-cell proteomic analysis identifies new liquid biopsy biomarkers associated with multiple myeloma”. In: *npj Precision Oncology* 7.1 (2023), p. 95. doi: 10.1038/s41698-023-00446-0.

[112] C. Seydel. “Single-cell metabolomics hits its stride”. In: *Nature Methods* 18.12 (2021), pp. 1452–1456. doi: 10.1038/s41592-021-01333-x.

[113] T. Hu et al. “Single-cell spatial metabolomics with cell-type specific protein profiling for tissue systems biology”. In: *Nature Communications* 14.1 (2023), p. 8260. doi: 10.1038/s41467-023-43917-5.

[114] K. Clark et al. “The Cancer Imaging Archive (TCIA): Maintaining and Operating a Public Information Repository”. In: *Journal of Digital Imaging* 26.6 (2013), pp. 1045–1057. doi: 10.1007/s10278-013-9622-7.

[115] L. Liu et al. “Spatiotemporal omics for biology and medicine”. In: *Cell* 187.17 (2024), pp. 4488–4519. doi: 10.1016/j.cell.2024.07.040.

[116] E. Lundberg and G. H. H. Borner. “Spatial proteomics: a powerful discovery tool for cell biology”. In: *Nature Reviews Molecular Cell Biology* 20.5 (2019), pp. 285–302. doi: 10.1038/s41580-018-0094-y.

[117] J. W. Hickey et al. “Spatial mapping of protein composition and tissue organization: a primer for multiplexed antibody-based imaging”. In: *Nature Methods* 19.3 (2022), pp. 284–295. doi: 10.1038/s41592-021-01316-y.

[118] S. Magaki et al. “An Introduction to the Performance of Immunohistochemistry”. In: *Biobanking*. Ed. by W. H. Yong. Vol. 1897. Series Title: Methods in Molecular Biology. New York, NY: Springer New York, 2019, pp. 289–298. doi: 10.1007/978-1-4939-8935-5_25.

[119] M. E. Ijsselsteijn et al. “Cancer immunophenotyping by seven□colour multispectral imaging without tyramide signal amplification”. In: *The Journal of Pathology: Clinical Research* 5.1 (2019), pp. 3–11. doi: 10.1002/cjp2.113.

[120] J.-R. Lin et al. “Highly multiplexed immunofluorescence imaging of human tissues and tumors using t-CyCIF and conventional optical microscopes”. In: *eLife* 7 (2018), e31657. doi: 10.7554/eLife.31657.

[121] A. J. Radtke et al. “IBEX: an iterative immunolabeling and chemical bleaching method for high-content imaging of diverse tissues”. In: *Nature Protocols* 17.2 (2022), pp. 378–401. doi: 10.1038/s41596-021-00644-9.

[122] S. K. Saka et al. “Immuno-SABER enables highly multiplexed and amplified protein imaging in tissues”. In: *Nature Biotechnology* 37.9 (2019), pp. 1080–1090. doi: 10.1038/s41587-019-0207-y.

[123] S. Black et al. “CODEX multiplexed tissue imaging with DNA-conjugated antibodies”. In: *Nature protocols* 16.8 (2021), pp. 3802–3835.

[124] L. Keren et al. “MIBI-TOF: A multiplexed imaging platform relates cellular phenotypes and tissue structure”. In: *Science Advances* 5.10 (2019), eaax5851. doi: 10.1126/sciadv.aax5851.

[125] M. Asp, J. Bergenstråhle, and J. Lundeberg. “Spatially Resolved Transcriptomes—Next Generation Tools for Tissue Exploration”. In: *BioEssays* 42.10 (2020). doi: 10.1002/bies.201900221.

[126] J. Du et al. “Advances in spatial transcriptomics and related data analysis strategies”. In: *Journal of Translational Medicine* 21.1 (2023), p. 330. doi: 10.1186/s12967-023-04150-2.

[127] L. Moses and L. Pachter. “Museum of spatial transcriptomics”. In: *Nature Methods* 19.5 (2022), pp. 534–546. doi: 10.1038/s41592-022-01409-2.

[128] K. H. Chen et al. “Spatially resolved, highly multiplexed RNA profiling in single cells”. In: *Science* 348.6233 (2015), aaa6090. doi: 10.1126/science.aaa6090.

[129] C.-H. L. Eng et al. “Transcriptome-scale super-resolved imaging in tissues by RNA seqFISH+”. In: *Nature* 568.7751 (2019), pp. 235–239. doi: 10.1038/s41586-019-1049-y.

[130] X. Wang et al. “Three-dimensional intact-tissue sequencing of single-cell transcriptional states”. In: *Science (New York, N.Y.)* 361.6400 (2018), eaat5691. doi: 10.1126/science.aat5691.

[131] J. H. Lee et al. “Fluorescent in situ sequencing (FISSEQ) of RNA for gene expression profiling in intact cells and tissues”. In: *Nature Protocols* 10.3 (2015), pp. 442–458. doi: 10.1038/nprot.2014.191.

[132] P. L. Ståhl et al. “Visualization and analysis of gene expression in tissue sections by spatial transcriptomics”. In: *Science* 353.6294 (2016), pp. 78–82. doi: 10.1126/science.aaf2403.

[133] S. G. Rodriques et al. “Slide-seq: A scalable technology for measuring genome-wide expression at high spatial resolution”. In: *Science* 363.6434 (2019), pp. 1463–1467. doi: 10.1126/science.aaw1219.

[134] R. R. Stickels et al. “Highly sensitive spatial transcriptomics at near-cellular resolution with Slide-seqV2”. In: *Nature Biotechnology* 39.3 (2021), pp. 313–319. doi: 10.1038/s41587-020-0739-1.

[135] M. Aichler and A. Walch. “MALDI Imaging mass spectrometry: current frontiers and perspectives in pathology research and practice”. In: *Laboratory Investigation* 95.4 (2015), pp. 422–431. doi: 10.1038/labinvest.2014.156.

[136] M. Stoeckli et al. “Imaging mass spectrometry: A new technology for the analysis of protein expression in mammalian tissues”. In: *Nature Medicine* 7.4 (2001), pp. 493–496. doi: 10.1038/86573.

[137] C. R. Merritt et al. “Multiplex digital spatial profiling of proteins and RNA in fixed tissue”. In: *Nature Biotechnology* 38.5 (2020), pp. 586–599. doi: 10.1038/s41587-020-0472-9.

[138] Y. Liu et al. “High-Spatial-Resolution Multi-Omics Sequencing via Deterministic Barcoding in Tissue”. In: *Cell* 183.6 (2020), 1665–1681.e18. doi: 10.1016/j.cell.2020.10.026.

[139] Y. Liu et al. “High-plex protein and whole transcriptome co-mapping at cellular resolution with spatial CITE-seq”. In: *Nature Biotechnology* 41.10 (2023), pp. 1405–1409. doi: 10.1038/s41587-023-01676-0.

[140] J. B. Nunes et al. “Integration of mass cytometry and mass spectrometry imaging for spatially resolved single-cell metabolic profiling”. In: *Nature Methods* 21.10 (2024), pp. 1796–1800. doi: 10.1038/s41592-024-02392-6.

[141] M. Vicari et al. “Spatial multimodal analysis of transcriptomes and metabolomes in tissues”. In: *Nature Biotechnology* 42.7 (2024), pp. 1046–1050. doi: 10.1038/s41587-023-01937-y.

[142] C. Engblom and J. Lundeberg. “Putting cancer immunotherapy into spatial context in the clinic”. In: *Nature Biotechnology* 43.4 (2025), pp. 471–476. doi: 10.1038/s41587-025-02596-x.

[143] W.-C. Hsieh et al. “Spatial multi-omics analyses of the tumor immune microenvironment”. In: *Journal of Biomedical Science* 29.1 (2022), p. 96. doi: 10.1186/s12929-022-00879-y.

[144] C. Engblom et al. “Spatial transcriptomics of B cell and T cell receptors reveals lymphocyte clonal dynamics”. In: *Science* 382.6675 (2023), eadf8486. doi: 10.1126/science.adf8486.

[145] H. Baharlou et al. “Mass Cytometry Imaging for the Study of Human Diseases—Applications and Data Analysis Strategies”. In: *Frontiers in Immunology* 10 (2019), p. 2657. doi: 10.3389/fimmu.2019.02657.

[146] A. Baranski et al. “MAUI (MBI Analysis User Interface)—An image processing pipeline for Multiplexed Mass Based Imaging”. In: *PLOS Computational Biology* 17.4 (2021). Ed. by D. Schneidman-Duhovny, e1008887. doi: 10.1371/journal.pcbi.1008887.

[147] S. Chevrier et al. “Compensation of Signal Spillover in Suspension and Imaging Mass Cytometry”. In: *Cell Systems* 6.5 (2018), 612–620.e5. doi: 10.1016/j.cels.2018.02.010.

[148] V. Milosevic. “Different approaches to Imaging Mass Cytometry data analysis”. In: *Bioinformatics Advances* 3.1 (2023). Ed. by G. Yu, vbad046. doi: 10.1093/bioadv/vbad046.

[149] Y. J. Wang et al. “Multiplexed In Situ Imaging Mass Cytometry Analysis of the Human Endocrine Pancreas and Immune System in Type 1 Diabetes”. In: *Cell Metabolism* 29.3 (2019), 769–783.e4. doi: 10.1016/j.cmet.2019.01.003.

[150] J. Windhager et al. “An end-to-end workflow for multiplexed image processing and analysis”. In: *Nature Protocols* 18.11 (2023), pp. 3565–3613. doi: 10.1038/s41596-023-00881-0.

[151] L. Keren et al. “A Structured Tumor-Immune Microenvironment in Triple Negative Breast Cancer Revealed by Multiplexed Ion Beam Imaging”. In: *Cell* 174.6 (2018), 1373–1387.e19. doi: 10.1016/j.cell.2018.08.039.

[152] A. F. Rendeiro et al. “The spatial landscape of lung pathology during COVID-19 progression”. In: *Nature* 593.7860 (2021), pp. 564–569. doi: 10.1038/s41586-021-03475-6.

[153] S. Berg et al. “ilastik: interactive machine learning for (bio)image analysis”. In: *Nature Methods* 16.12 (2019), pp. 1226–1232. doi: 10.1038/s41592-019-0582-9.

[154] M. E. Ijsselsteijn et al. “Semi‐automated background removal limits data loss and normalizes imaging mass cytometry data”. In: *Cytometry Part A* 99.12 (2021), pp. 1187–1197. doi: 10.1002/cyto.a.24480.

[155] P. Lu et al. “IMC-Denoise: a content aware denoising pipeline to enhance Imaging Mass Cytometry”. In: *Nature Communications* 14.1 (2023), p. 1601. doi: 10.1038/s41467-023-37123-6.

[156] C. McQuin et al. “CellProfiler 3.0: Next-generation image processing for biology”. In: *PLOS Biology* 16.7 (2018). Ed. by T. Misteli, e2005970. doi: 10.1371/journal.pbio.2005970.

[157] A. E. Carpenter et al. “CellProfiler: image analysis software for identifying and quantifying cell phenotypes”. In: *Genome Biology* 7.10 (2006), R100. doi: 10.1186/gb-2006-7-10-r100.

[158] J. Schindelin et al. “Fiji: an open-source platform for biological-image analysis”. In: *Nature Methods* 9.7 (2012), pp. 676–682. doi: 10.1038/nmeth.2019.

[159] L. Von Chamier et al. “Democratising deep learning for microscopy with ZeroCostDL4Mic”. In: *Nature Communications* 12.1 (2021), p. 2276. doi: 10.1038/s41467-021-22518-0.

[160] N. F. Greenwald et al. “Whole-cell segmentation of tissue images with human-level performance using large-scale data annotation and deep learning”. In: *Nature Biotechnology* 40.4 (2022), pp. 555–565. doi: 10.1038/s41587-021-01094-0.

[161] C. Stringer et al. “Cellpose: a generalist algorithm for cellular segmentation”. In: *Nature Methods* 18.1 (2021), pp. 100–106. doi: 10.1038/s41592-020-01018-x.

[162] M. Pachitariu and C. Stringer. “Cellpose 2.0: how to train your own model”. In: *Nature Methods* 19.12 (2022), pp. 1634–1641. doi: 10.1038/s41592-022-01663-4.

[163] M. Y. Lee et al. “CellSeg: a robust, pre-trained nucleus segmentation and pixel quantification software for highly multiplexed fluorescence images”. In: *BMC Bioinformatics* 23.1 (2022), p. 46. doi: 10.1186/s12859-022-04570-9.

[164] D. Schapiro et al. “MCMICRO: a scalable, modular image-processing pipeline for multiplexed tissue imaging”. In: *Nature Methods* 19.3 (2022), pp. 311–315. doi: 10.1038/s41592-021-01308-y.

[165] M. J. D. Baars et al. “MATISSE: a method for improved single cell segmentation in imaging mass cytometry”. In: *BMC Biology* 19.1 (2021), p. 99. doi: 10.1186/s12915-021-01043-y.

[166] S. Mandal and V. Uhlmann. *SplineDist: Automated Cell Segmentation With Spline Curves*. 2020. doi: 10.1101/2020.10.27.357640.

[167] U. Schmidt et al. “Cell Detection with Star-convex Polygons”. In: (2018). Publisher: arXiv Version Number: 2. doi: 10.48550/ARXIV.1806.03535.

[168] V. Petukhov et al. “Cell segmentation in imaging-based spatial transcriptomics”. In: *Nature Biotechnology* 40.3 (2022), pp. 345–354. doi: 10.1038/s41587-021-01044-w.

[169] J. Park et al. “Cell segmentation-free inference of cell types from in situ transcriptomics data”. In: *Nature Communications* 12.1 (2021), p. 3545. doi: 10.1038/s41467-021-23807-4.

[170] S. Prabhakaran. “Sparcle: assigning transcripts to cells in multiplexed images”. In: *Bioinformatics Advances* 2.1 (2022). Ed. by A. Bateman, vbac048. doi: 10.1093/bioadv/vbac048.

[171] Y. He et al. “ClusterMap for multi-scale clustering analysis of spatial gene expression”. In: *Nature Communications* 12.1 (2021), p. 5909. doi: 10.1038/s41467-021-26044-x.

[172] Y. Si et al. “FICTURE: scalable segmentation-free analysis of submicron-resolution spatial transcriptomics”. In: *Nature Methods* 21.10 (2024), pp. 1843–1854. doi: 10.1038/s41592-024-02415-2.

[173] X. Fu et al. “BIDCell: Biologically-informed self-supervised learning for segmentation of subcellular spatial transcriptomics data”. In: *Nature Communications* 15.1 (2024), p. 509. doi: 10.1038/s41467-023-44560-w.

[174] T. Defard et al. “A point cloud segmentation framework for image-based spatial transcriptomics”. In: *Communications Biology* 7.1 (2024), p. 823. doi: 10.1038/s42003-024-06480-3.

[175] G. Partel and C. Wählby. “Spage2vec: Unsupervised representation of localized spatial gene expression signatures”. In: *The FEBS Journal* 288.6 (2021), pp. 1859–1870. doi: 10.1111/febs.15572.

[176] X. Qian et al. “Probabilistic cell typing enables fine mapping of closely related cell types in situ”. In: *Nature Methods* 17.1 (2020), pp. 101–106. doi: 10.1038/s41592-019-0631-4.

[177] S. Marco Salas et al. “Optimizing Xenium In Situ data utility by quality assessment and best-practice analysis workflows”. In: *Nature Methods* 22.4 (2025), pp. 813–823. doi: 10.1038/s41592-025-02617-2.

[178] V. Van Unen et al. “Visual analysis of mass cytometry data by hierarchical stochastic neighbour embedding reveals rare cell types”. In: *Nature Communications* 8.1 (2017), p. 1740. doi: 10.1038/s41467-017-01689-9.

[179] S. Van Gassen et al. “FlowSOM: Using self‐organizing maps for visualization and interpretation of cytometry data”. In: *CytometryPartA* 87.7 (2015), pp. 636–645. doi: 10.1002/cyto.a.22625.

[180] K. Quintelier et al. “Analyzing high-dimensional cytometry data using FlowSOM”. In: *Nature Protocols* 16.8 (2021), pp. 3775–3801. doi: 10.1038/s41596-021-00550-0.

[181] L. Heumos et al. “Best practices for single-cell analysis across modalities”. In: *Nature Reviews Genetics* 24.8 (2023), pp. 550–572. doi: 10.1038/s41576-023-00586-w.

[182] V. D. Blondel et al. “Fast unfolding of communities in large networks”. In: *Journal of Statistical Mechanics: Theory and Experiment* 2008.10 (2008), P10008. doi: 10.1088/1742-5468/2008/10/P10008.

[183] L. Waltman and N. J. Van Eck. “A smart local moving algorithm for large-scale modularity-based community detection”. In: *The European Physical Journal B* 86.11 (2013), p. 471. doi: 10.1140/epjb/e2013-40829-0.

[184] V. A. Traag, L. Waltman, and N. J. Van Eck. “From Louvain to Leiden: guaranteeing well-connected communities”. In: *Scientific Reports* 9.1 (2019). doi: 10.1038/s41598-019-41695-z.

[185] C. Domínguez Conde et al. “Cross-tissue immune cell analysis reveals tissue-specific features in humans”. In: *Science* 376.6594 (2022), eabl5197. doi: 10.1126/science.abl5197.

[186] J. B. Kang et al. “Efficient and precise single-cell reference atlas mapping with Symphony”. In: *Nature Communications* 12.1 (2021), p. 5890. doi: 10.1038/s41467-021-25957-x.

[187] D. Schapiro et al. “histoCAT: analysis of cell phenotypes and interactions in multiplex image cytometry data”. In: *Nature Methods* 14.9 (2017), pp. 873–876. doi: 10.1038/nmeth.4391.

[188] C. M. Schürch et al. “Coordinated Cellular Neighborhoods Orchestrate Antitumoral Immunity at the Colorectal Cancer Invasive Front”. In: *Cell* 182.5 (2020), 1341–1359.e19. doi: 10.1016/j.cell.2020.07.005.

[189] D. Haviv et al. “The covariance environment defines cellular niches for spatial inference”. In: *Nature Biotechnology* 43.2 (2025), pp. 269–280. doi: 10.1038/s41587-024-02193-4.

[190] S. Birk et al. “Quantitative characterization of cell niches in spatially resolved omics data”. In: *Nature Genetics* 57.4 (2025), pp. 897–909. doi: 10.1038/s41588-025-02120-6.

[191] V. Singhal et al. “BANKSY unifies cell typing and tissue domain segmentation for scalable spatial omics data analysis”. In: *Nature Genetics* 56.3 (2024), pp. 431–441. doi: 10.1038/s41588-024-01664-3.

[192] E. Zhao et al. “Spatial transcriptomics at subspot resolution with BayesSpace”. In: *Nature Biotechnology* 39.11 (2021), pp. 1375–1384. doi: 10.1038/s41587-021-00935-2.

[193] Q. Zhu et al. “Identification of spatially associated subpopulations by combining scRNAseq and sequential fluorescence in situ hybridization data”. In: *Nature Biotechnology* 36.12 (2018), pp. 1183–1190. doi: 10.1038/nbt.4260.

[194] D. Pham et al. “Robust mapping of spatiotemporal trajectories and cell–cell interactions in healthy and diseased tissues”. In: *Nature Communications* 14.1 (2023), p. 7739. doi: 10.1038/s41467-023-43120-6.

[195] J. Hu et al. “SpaGCN: Integrating gene expression, spatial location and histology to identify spatial domains and spatially variable genes by graph convolutional network”. In: *Nature Methods* 18.11 (2021), pp. 1342–1351. doi: 10.1038/s41592-021-01255-8.

[196] K. Dong and S. Zhang. “Deciphering spatial domains from spatially resolved transcriptomics with an adaptive graph attention auto-encoder”. In: *Nature Communications* 13.1 (2022), p. 1739. doi: 10.1038/s41467-022-29439-6.

[197] Y. Long et al. “Spatially informed clustering, integration, and deconvolution of spatial transcriptomics with GraphST”. In: *Nature Communications* 14.1 (2023), p. 1155. doi: 10.1038/s41467-023-36796-3.

[198] A. Andersson et al. “Points2Regions: Fast, interactive clustering of imaging‐based spatial transcriptomics data”. In: *Cytometry Part A* 105.9 (2024), pp. 677–687. doi: 10.1002/cyto.a.24884.

[199] B. L. Walker and Q. Nie. “NeST: nested hierarchical structure identification in spatial transcriptomic data”. In: *Nature Communications* 14.1 (2023). doi: 10.1038/s41467-023-42343-x.

[200] S. S. Bhate et al. “Tissue schematics map the specialization of immune tissue motifs and their appropriation by tumors”. In: *Cell Systems* 13.2 (2022), 109–130.e6. doi: 10.1016/j.cels.2021.09.012.

[201] A. Getis. “Spatial Autocorrelation”. In: *Handbook of Applied Spatial Analysis*. Ed. by M. M. Fischer and A. Getis. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 255–278. doi: 10.1007/978-3-642-03647-7_14.

[202] V. Svensson, S. A. Teichmann, and O. Stegle. “SpatialDE: identification of spatially variable genes”. In: *Nature Methods* 15.5 (2018), pp. 343–346. doi: 10.1038/nmeth.4636.

[203] D. Edsgärd, P. Johnsson, and R. Sandberg. “Identification of spatial expression trends in single-cell gene expression data”. In: *Nature Methods* 15.5 (2018), pp. 339–342. doi: 10.1038/nmeth.4634.

[204] S. Sun, J. Zhu, and X. Zhou. “Statistical analysis of spatial expression patterns for spatially resolved transcriptomic studies”. In: *Nature Methods* 17.2 (2020), pp. 193–200. doi: 10.1038/s41592-019-0701-7.

[205] J. Zhu, S. Sun, and X. Zhou. “SPARK-X: non-parametric modeling enables scalable and robust detection of spatial expression patterns for large spatial transcriptomic studies”. In: *Genome Biology* 22.1 (2021), p. 184. doi: 10.1186/s13059-021-02404-0.

[206] A. Andersson and J. Lundeberg. “*sepal* : identifying transcript profiles with spatial patterns by diffusion-based modeling”. In: *Bioinformatics* 37.17 (2021). Ed. by A. Mathelier, pp. 2644–2650. doi: 10.1093/bioinformatics/btab164.

[207] K. Zhang, W. Feng, and P. Wang. “Identification of spatially variable genes with graph cuts”. In: *Nature Communications* 13.1 (2022), p. 5488. doi: 10.1038/s41467-022-33182-3.

[208] E. Armingol, H. M. Baghdassarian, and N. E. Lewis. “The diversification of methods for studying cell–cell interactions and communication”. In: *Nature Reviews Genetics* 25.6 (2024), pp. 381–400. doi: 10.1038/s41576-023-00685-8.

[209] D. Dimitrov et al. “LIANA+ provides an all-in-one framework for cell–cell communication inference”. In: *Nature Cell Biology* 26.9 (2024). doi: 10.1038/s41556-024-01469-w.

[210] D. S. Fischer, A. C. Schaar, and F. J. Theis. “Modeling intercellular communication in tissues using spatial graphs of cells”. In: *Nature Biotechnology* 41.3 (2023), pp. 332–336. doi: 10.1038/s41587-022-01467-z.

[211] K. Troulé et al. “CellPhoneDB v5: inferring cell–cell communication from single-cell multiomics data”. In: *Nature Protocols* (2025). doi: 10.1038/s41596-024-01137-1.

[212] S. Jin, M. V. Plikus, and Q. Nie. “CellChat for systematic analysis of cell–cell communication from single-cell transcriptomics”. In: *Nature Protocols* 20.1 (2025), pp. 180–219. doi: 10.1038/s41596-024-01045-4.

[213] X. Shao et al. “Knowledge-graph-based cell-cell communication inference for spatially resolved transcriptomic data with SpaTalk”. In: *Nature Communications* 13.1 (2022), p. 4429. doi: 10.1038/s41467-022-32111-8.

[214] Z. Cang et al. “Screening cell–cell communication in spatial transcriptomics via collective optimal transport”. In: *Nature Methods* 20.2 (2023). doi: 10.1038/s41592-022-01728-4.

[215] Z. Li et al. “SpatialDM for rapid identification of spatially co-expressed ligand–receptor and revealing cell–cell communication patterns”. In: *Nature Communications* 14.1 (2023), p. 3995. doi: 10.1038/s41467-023-39608-w.

[216] R. Browaeys, W. Saelens, and Y. Saeys. “NicheNet: modeling intercellular communication by linking ligands to target genes”. In: *Nature Methods* 17.2 (2020), pp. 159–162. doi: 10.1038/s41592-019-0667-5.

[217] Y. Yuan and Z. Bar-Joseph. “GCNG: graph convolutional networks for inferring gene interaction from spatial transcriptomics data”. In: *Genome Biology* 21.1 (2020). doi: 10.1186/s13059-020-02214-w.

[218] J. Zhu et al. “Mapping cellular interactions from spatially resolved transcriptomics data”. In: *Nature Methods* 21.10 (2024), pp. 1830–1842. doi: 10.1038/s41592-024-02408-1.

[219] B. L. Walker et al. “Deciphering tissue structure and function using spatial transcriptomics”. In: *Communications Biology* 5.1 (2022), p. 220. doi: 10.1038/s42003-022-03175-5.

[220] D. Türei, T. Korcsmáros, and J. Saez-Rodriguez. “OmniPath: guidelines and gateway for literature-curated signaling pathway resources”. In: *Nature Methods* 13.12 (2016), pp. 966–967. doi: 10.1038/nmeth.4077.

[221] D. Arnol et al. “Modeling Cell-Cell Interactions from Spatial Molecular Data with Spatial Variance Component Analysis”. In: *Cell Reports* 29.1 (2019). doi: 10.1016/j.celrep.2019.08.077.

[222] J. Tanevski et al. “Explainable multiview framework for dissecting spatial relationships from highly multiplexed data”. In: *Genome Biology* 23.1 (2022). doi: 10.1186/s13059-022-02663-5.

[223] P. Badia-i-Mompel et al. “decoupleR: ensemble of computational methods to infer biological activities from omics data”. In: *Bioinformatics Advances* 2.1 (2022). Ed. by M. L. Kuijjer, vbac016. doi: 10.1093/bioadv/vbac016.

[224] M. Schubert et al. “Perturbation-response genes reveal signaling footprints in cancer gene expression”. In: *Nature Communications* 9.1 (2018), p. 20. doi: 10.1038/s41467-017-02391-6.

[225] S. Müller-Dott et al. “Expanding the coverage of regulons from high-confidence prior knowledge for accurate estimation of transcription factor activities”. In: *Nucleic Acids Research* 51.20 (2023), pp. 10934–10949. doi: 10.1093/nar/gkad841.

[226] G. Palla et al. “Squidpy: a scalable framework for spatial omics analysis”. In: *Nature Methods* 19.2 (2022), pp. 171–178. doi: 10.1038/s41592-021-01358-2.

[227] R. Argelaguet et al. “Multi□Omics Factor Analysis—a framework for unsupervised integration of multi□omics data sets”. In: *Molecular Systems Biology* 14.6 (2018), e8124. doi: 10.15252/msb.20178124.

[228] R. Argelaguet et al. “MOFA+: a statistical framework for comprehensive integration of multi-modal single-cell data”. In: *Genome Biology* 21.1 (2020). doi: 10.1186/s13059-020-02015-1.

[229] R. O. Ramirez Flores et al. “Multicellular factor analysis of single-cell data for a tissue-centric understanding of disease”. In: *eLife* 12 (2023), e93161. doi: 10.7554/eLife.93161.

[230] B. Velten et al. “Identifying temporal and spatial patterns of variation from multimodal data using MEFISTO”. In: *Nature Methods* 19.2 (2022), pp. 179–186. doi: 10.1038/s41592-021-01343-9.

[231] L. Jerby-Arnon and A. Regev. “DIALOGUE maps multicellular programs in tissue from single-cell or spatial transcriptomics data”. In: *Nature Biotechnology* 40.10 (2022), pp. 1467–1477. doi: 10.1038/s41587-022-01288-0.

[232] J. Mitchel et al. “Coordinated, multicellular patterns of transcriptional variation that stratify patient cohorts are revealed by tensor decomposition”. In: *Nature Biotechnology* (2024). doi: 10.1038/s41587-024-02411-z.

[233] F. W. Townes and B. E. Engelhardt. “Nonnegative spatial factorization applied to spatial genomics”. In: *Nature Methods* 20.2 (2023), pp. 229–238. doi: 10.1038/s41592-022-01687-w.

[234] D. Jiménez-Sánchez et al. “NaroNet: Discovery of tumor microenvironment elements from highly multiplexed images”. In: *Medical Image Analysis* 78 (2022), p. 102384. doi: 10.1016/j.media.2022.102384.

[235] X. Fu et al. “Deep multimodal graph-based network for survival prediction from highly multiplexed images and patient variables”. In: *Computers in Biology and Medicine* 154 (2023), p. 106576. doi: 10.1016/j.compbiomed.2023.106576.

[236] T. Risom et al. “Transition to invasive breast cancer is associated with progressive changes in the structure and composition of tumor stroma”. In: *Cell* 185.2 (2022), 299–310.e18. doi: 10.1016/j.cell.2021.12.023.

[237] S. Babaei et al. “S3-CIMA: Supervised spatial single-cell image analysis for identifying disease-associated cell-type compositions in tissue”. In: *Patterns* 4.9 (2023), p. 100829. doi: 10.1016/j.patter.2023.100829.

[238] Y. Wang et al. “Cell graph neural networks enable the precise prediction of patient survival in gastric cancer”. In: *npj Precision Oncology* 6.1 (2022), p. 45. doi: 10.1038/s41698-022-00285-5.

[239] R. Perez-Lopez et al. “A guide to artificial intelligence for cancer researchers”. In: *Nature Reviews Cancer* 24.6 (2024), pp. 427–441. doi: 10.1038/s41568-024-00694-7.

[240] M. Unger and J. N. Kather. “A systematic analysis of deep learning in genomics and histopathology for precision oncology”. In: *BMC Medical Genomics* 17.1 (2024), p. 48. doi: 10.1186/s12920-024-01796-9.

[241] Y. Hao et al. “Integrated analysis of multimodal single-cell data”. In: *Cell* 184.13 (2021), 3573–3587.e29. doi: 10.1016/j.cell.2021.04.048.

[242] F. A. Wolf, P. Angerer, and F. J. Theis. “SCANPY: large-scale single-cell gene expression data analysis”. In: *Genome Biology* 19.1 (2018), p. 15. doi: 10.1186/s13059-017-1382-0.

[243] I. Virshup et al. “anndata: Access and store annotated datamatrices”. In: *Journal of Open Source Software* 9.101 (2024), p. 4371. doi: 10.21105/joss.04371.

[244] R. Dries et al. “Giotto: a toolbox for integrative analysis and visualization of spatial expression data”. In: *Genome Biology* 22.1 (2021), p. 78. doi: 10.1186/s13059-021-02286-2.

[245] L. Marconato et al. “SpatialData: an open and universal data framework for spatial omics”. In: *Nature Methods* 22.1 (2025), pp. 58–62. doi: 10.1038/s41592-024-02212-x.

[246] N. Sofroniew et al. *napari: a multi-dimensional image viewer for Python*. 2025. doi: 10.5281/ZENODO.3555620.

[247] E. Roesch et al. “Julia for biologists”. In: *Nature Methods* 20.5 (2023), pp. 655–664. doi: 10.1038/s41592-023-01832-z.

[248] I. Virshup et al. “The scverse project provides a computational ecosystem for single-cell omics data analysis”. In: *Nature Biotechnology* 41.5 (2023). doi: 10.1038/s41587-023-01733-8.

[249] C. C. Liu et al. “Robust phenotyping of highly multiplexed tissue imaging data using pixel-level clustering”. In: *Nature Communications* 14.1 (2023). doi: 10.1038/s41467-023-40068-5.

[250] N. L. De Vries et al. “Unraveling the Complexity of the Cancer Microenvironment With Multidimensional Genomic and Cytometric Technologies”. In: *Frontiers in Oncology* 10 (2020), p. 1254. doi: 10.3389/fonc.2020.01254.

[251] J. Roelands et al. “Transcriptomic and immunophenotypic profiling reveals molecular and immunological hallmarks of colorectal cancer tumourigenesis”. In: *Gut* 72.7 (2023), pp. 1326–1339. doi: 10.1136/gutjnl-2022-327608.

[252] F. Chollet. *Building autoencoders in keras*. 2016.

[253] P. Virtanen et al. “SciPy 1.0: fundamental algorithms for scientific computing in Python”. In: *Nature Methods* 17.3 (2020), pp. 261–272. doi: 10.1038/s41592-019-0686-2.

[254] S. Van Der Walt et al. “scikit-image: image processing in Python”. In: *PeerJ* 2 (2014), e453. doi: 10.7717/peerj.453.

[255] K. Dabov et al. “Image Denoising by Sparse 3-D Transform-Domain Collaborative Filtering”. In: *IEEE Transactions on Image Processing* 16.8 (2007). doi: 10.1109/TIP.2007.901238.

[256] A. Somarakis et al. “ImaCytE: Visual Exploration of Cellular Micro-Environments for Imaging Mass Cytometry Data”. In: *IEEE Transactions on Visualization and Computer Graphics* 27.1 (2021), pp. 98–110. doi: 10.1109/TVCG.2019.2931299.

[257] T. Höllt et al. “Cytosplore: Interactive Immune Cell Phenotyping for Large Single□Cell Datasets”. In: *Computer Graphics Forum* 35.3 (2016), pp. 171–180. doi: 10.1111/cgf.12893.

[258] A. Kirillov et al. *Segment Anything*. 2023. doi: 10.48550/arXiv.2304.02643.

[259] U. Israel et al. *A Foundation Model for Cell Segmentation*. 2023. doi: 10.1101/2023.11.17.567630.

[260] F. Van Maldegem et al. “Characterisation of tumour microenvironment remodelling following oncogene inhibition in preclinical studies with imaging mass cytometry”. In: *Nature Communications* 12.1 (2021), p. 5906. doi: 10.1038/s41467-021-26214-x.

[261] A. Magness et al. “Deep cell phenotyping and spatial analysis of multiplexed imaging with TRACERx-PHLEX”. In: *Nature Communications* 15.1 (2024). doi: 10.1038/s41467-024-48870-5.

[262] M. Allam et al. “Spatially variant immune infiltration scoring in human cancer tissues”. In: *npj Precision Oncology* 6.1 (2022), p. 60. doi: 10.1038/s41698-022-00305-4.

[263] J. Roelands et al. “Transcriptomic and immunophenotypic profiling reveals molecular and immunological hallmarks of colorectal cancer tumourigenesis”. In: *Gut* 72.7 (2023), pp. 1326–1339. doi: 10.1136/gutjnl-2022-327608.

[264] A. Sequeira et al. “PENGUIN: A rapid and efficient image preprocessing tool for multiplexed spatial proteomics”. In: *Computational and Structural Biotechnology Journal* 23 (2024), pp. 3920–3928. doi: 10.1016/j.csbj.2024.10.048.

[265] D. R. Stirling et al. “CellProfiler 4: improvements in speed, utility and usability”. In: *BMC Bioinformatics* 22.1 (2021), p. 433. doi: 10.1186/s12859-021-04344-9.

[266] J. H. Levine et al. “Data-Driven Phenotypic Dissection of AML Reveals Progenitor-like Cells that Correlate with Prognosis”. In: *Cell* 162.1 (2015), pp. 184–197. doi: 10.1016/j.cell.2015.05.047.

[267] J. Ma et al. “The multimodality cell segmentation challenge: toward universal solutions”. In: *Nature Methods* 21.6 (2024), pp. 1103–1113. doi: 10.1038/s41592-024-02233-6.

[268] J. Ellson et al. “Graphviz— Open Source Graph Drawing Tools”. In: *Graph Drawing*. Ed. by G. Goos et al. Vol. 2265. Series Title: Lecture Notes in Computer Science. Berlin, Heidelberg: Springer Berlin Heidelberg, 2002, pp. 483–484. doi: 10.1007/3-540-45848-4_57.

[269] J. Kim et al. “Unsupervised discovery of tissue architecture in multiplexed imaging”. In: *Nature Methods* 19.12 (2022), pp. 1653–1661. doi: 10.1038/s41592-022-01657-2.

[270] S. Sun, J. Zhu, and X. Zhou. “Statistical analysis of spatial expression patterns for spatially resolved transcriptomic studies”. In: *Nature Methods* 17.2 (2020), pp. 193–200. doi: 10.1038/s41592-019-0701-7.

[271] A. Getis and J. K. Ord. “The Analysis of Spatial Association by Use of Distance Statistics”. In: *Geographical Analysis* 24.3 (1992). doi: 10.1111/j.1538-4632.1992.tb00261.x.

[272] C. R. Harris et al. “Array programming with NumPy”. In: *Nature* 585.7825 (2020), pp. 357–362. doi: 10.1038/s41586-020-2649-2. url: https://doi.org/10.1038/s41586-020-2649-2.

[273] T. pandas development team. *Pandas*. 2024. doi: 10.5281/ZENODO.3509134.

[274] J. D. Hunter. “Matplotlib: A 2D Graphics Environment”. In: *Computing in Science & Engineering* 9.3 (2007), pp. 90–95. doi: 10.1109/MCSE.2007.55.

[275] P. Virtanen et al. “SciPy 1.0: fundamental algorithms for scientific computing in Python”. In: *Nature Methods* 17.3 (2020), pp. 261–272. doi: 10.1038/s41592-019-0686-2.

[276] G. Bradski. “The OpenCV Library”. In: *Dr. Dobb’s Journal of Software Tools* (2000).

[277] M. Ester et al. “A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise”. In: *Proceedings of the 2nd International Conference on Knowledge Discovery and Data Mining*. Portland: AAAI Press, 1996, pp. 226–231.

[278] G. Vettigli. *MiniSom: minimalistic and NumPy-based implementation of the Self Organizing Map*. 2018.

[279] R. Luo et al. “BioGPT: generative pre-trained transformer for biomedical text generation and mining”. In: *Briefings in Bioinformatics* 23.6 (2022). doi: 10.1093/bib/bbac409.

[280] I. Beltagy, K. Lo, and A. Cohan. “SciBERT: A Pretrained Language Model for Scientific Text”. In: *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. Ed. by K. Inui et al. Hong Kong, China: Association for Computational Linguistics, 2019, pp. 3615–3620. doi: 10.18653/v1/D19-1371.

[281] S. Lobentanzer et al. “A platform for the biomedical application of large language models”. In: *Nature Biotechnology* (2025), pp. 1–4.