Developing an AI-based approach to predict response to anti-EGFR treatment for metastatic colorectal cancer patients using super-resolution imaging of EREG

Oliver Umney

Submitted in accordance with the requirements for the degree of

Doctor of Philosophy

The University of Leeds

School of Computer Science

September 2025

I confirm that the work submitted is my own, except where work which has formed part of jointly authored publications has been included. My contribution and the other authors to this work has been explicitly indicated below. I confirm that appropriate credit has been given within the thesis where reference has been made to the work of others.

- Chapter 2 is based on Umney, O., Leng, J., Canettieri, G., Riobo-Del Galdo, N. A., Slaney, H., Quirke, P., Peckham, M., & Curd, A. (2024). Annotation and automated segmentation of single-molecule localisation microscopy data. Journal of Microscopy, 296, 214–226. https://doi.org/10.1111/jmi.13349. I was the lead author, conducted the research, wrote the draft manuscript, and generated all tables/figures. The analysis used an image dataset previously generated by Hayley Slaney from cells prepared by Gianluca Canettieri and Natalia A. Riobo-Del Galdo. Michelle Peckham, Alistair Curd, Philip Quirke, and Joanna Leng aided in project design, provided feedback on the draft and final manuscripts.

- Chapters 3 and 4 are based on Umney, O., Slaney, H., Williams, C. J. M., Quirke, P., Peckham, M., & Curd, A. (2025). ClusterNet: Classifying Single-Molecule Localization Microscopy Datasets with Graph-Based Deep Learning of Supra-Cluster Structure. [Accepted and in production]. Small Science. https://doi.org/10.1002/smsc.202500255. I was the lead author, conducted the research, wrote the draft manuscript, and generated all tables/figures. Hayley Slaney assisted in preparing samples provided by Phil Quirke and Christopher Williams. Alistair Curd, Michelle Peckham, Philip Quirke and Christopher

Williams aided in project design, provided feedback on the draft and final manuscripts.

Acknowledgments

I would like to thank my supervisors Alistair Curd, Michelle Peckham, Phil Quirke and Joanna Leng. I appreciate all the feedback, insights and wisdom you have given me over the last four years. Particularly to Alistair and Michelle for making me a better researcher (I hope) and reminding me to keep it simple.

I would also like to thank Hayley Slaney, Begona Caballero Ruiz, Gianluca Canettieri, Natalia A. Riobo-Del Galdo and Christopher Williams for help with preparing the biological samples and acquiring the SMLM data.

This study would not have been possible without the investment by the UK taxpayer, the patients who allow their data to be used for research and the many educators and researchers who willingly share their knowledge. It takes a village to do research, something easily forgotten amidst the currently booming AI industry of socialised costs and privatised profits.

Thanks to Richard Stephens for your mentorship and the emails from sunny places on a rainy day in Leeds.

Thanks to UK EPSRC for the funding and to the Leeds Medical AI CDT for creating a fun and welcoming environment. Thanks also to the CDT and the many others who supported the public engagement activities, and most importantly to the public and patients who challenged and inspired me.

Thank you to the friends I have met in Leeds: Aron, Joe, Taz, Morgan, Rachael and those that followed… (Mills); and to my family and friends outside, particularly to my sisters, dad and mum for putting up with me for the last 27 years.

Finally, a special thank you to Lucy for your unwavering support. I would not have made it without your love and encouragement. Also, for fostering an unhealthy addiction to peanut butter. Crunch crunch!

Abstract

Every year, there are approximately two million new cases of colorectal cancer worldwide. Globally, there is a growing number of cases of early-onset (< 50 years old) colorectal cancer, which is often diagnosed at an advanced stage. The outcomes for metastatic patients are grim, with a 5-year net survival rate of only 1 in 10. For these patients, treatment that targets epidermal growth factor receptor (EGFR), a cell surface protein that is involved in cell signalling, division and growth, can help shrink metastases for resection or slow cancer progression for palliative care. However, approximately 40% of patients receiving this treatment do not respond. The objective of this study was to investigate whether the nanoscale spatial organisation of epiregulin (EREG), one of the ligands for EGFR, could help predict response to anti-EGFR treatment for metastatic colorectal cancer patients.

To achieve this objective, we imaged tissue samples from metastatic colorectal cancer patients using single-molecule localisation microscopy (SMLM), which could resolve the high-precision positions of EREG proteins. We then developed and tested artificial intelligence (AI) based pipelines, *locpix* and *ClusterNet*, to segment and classify large-scale structures in SMLM data, such as cells. These pipelines were then applied to the SMLM data from the patients to manually segment the cells and classify them by response to treatment.

This approach may improve over an existing method for predicting response, which uses the protein expression level of EREG, but was inconclusive due to the small sample size in this study. More broadly, this study showed that the organisation of EREG may help predict response to anti-EGFR treatment. Further, we anticipate that the two novel AI-based pipelines may be generally useful for the analysis of SMLM data. This includes the first example of a graph-neural network designed for whole-graph classification of SMLM data. These pipelines could help to realise the use of SMLM data to characterise phenotypes and predict response to treatment across a wide variety of disorders.

Table of Contents

List of Tables

List of Figures

List of Abbreviations

ADCC	antibody-dependent cellular cytotoxicity
AI	artificial intelligence
AKT	protein kinase B
APC	adenomatous polyposis coli
APES	3-aminopropyltriethoxysilane
AREG	amphiregulin
ASAP	automated structures analysis program
AUC	area under the curve
AUCNPR	normalised area under the precision-recall curve
AUROC	area under the receiver operator curve
BRAF	rapidly accelerated fibrosarcoma isoform b
BSA	bovine serum albumin
CODI	collaborative discovery platform
CNN	convolutional neural network
CT	computed tomography
DBSCAN	density-based spatial clustering of applications with noise
DL	deep learning
dMMR	mismatch repair deficiency
DNA	deoxyribonucleic acid
dSTORM	direct stochastic optical reconstruction microscopy
ECLIPSE	enhanced classification of localized point clouds by shape extraction
EGF	epidermal growth factor
EGFR	epidermal growth factor receptor
ERBB	erythroblastic leukaemia viral oncogene
EREG	epiregulin
ERK	extracellular signal-regulated kinase
FFPE	formalin-fixed paraffin-embedded
FN	false negative

FOV	field of view
FP	false positive
FPR	false positive rate
GFP	green fluorescent protein
GNN	graph neural network
GT	ground truth
GUI	graphical user interface
H&E	haematoxylin and eosin
HCF	handcrafted feature
HER2	human epidermal growth factor receptor 2
IgG1	immunoglobulin G1
IHC	immunohistochemistry
KRAS	Kirsten rat sarcoma virus
LCF	learnt cluster feature
LIDAR	light detection and ranging
MEK	mitogen-activated protein kinase
ML	machine learning
MLP	multilayer perceptron
MRI	magnetic resonance imaging
mRNA	messenger RNA
MSD	mean-squared displacement
MSI-H	high microsatellite instability
MTOR	mammalian target of rapamycin
NA	numerical aperture
NN	neural network
NRAS	neuroblastoma rat sarcoma virus
ONI	Oxford nanoimaging
PAINT	point accumulation in nanoscale topography
PALM	photoactivated localisation microscopy
PBS	phosphate-buffered saline

PC	principal component
PCA	principal component analysis
PI3K	phosphoinositide 3-kinase
PICCOLO	panitumumab, irinotecan & ciclosporin in colorectal cancer therapy
PLCγ	phospholipase c-γ
PR	precision-recall
PSF	point spread function
PTB	phosphotyrosine-binding
PTCH1	patch1
RAF	rapidly accelerated fibrosarcoma
RAS	rat sarcoma virus
RDF	radial distribution function
RECIST	response evaluation criteria in solid tumours
RF	random forest
RNA	ribonucleic acid
ROC	receiver operator curve
ROI	region of interest
RTS	reserved test set
SEMORE	segmentation and morphological fingerprinting
SH2	Src homology 2
SHAP	Shapley additive explanations
SMLM	single-molecule localisation microscopy
STORM	stochastic optical reconstruction microscopy
TIRFM	total internal reflection fluorescence microscopy
TMA	tissue microarray
TN	true negative
TP	true positive
TP53	tumour protein p53
t-SNE	t-distributed stochastic neighbour embedding

UMAP	uniform manifold approximation and projection
VEGF	vascular endothelial growth factor
WT	wild type

1 Introduction

1.1 Colorectal cancer

1.1.1 What is colorectal cancer?

Cancer is a disease where abnormal cells grow and spread uncontrollably, often characterised by its hallmarks, such as sustained proliferative signalling, the ability of cancer to maintain constant growth independent of normal cell signalling [1-3]. Colorectal cancer is a cancer that develops in the colon and/or rectum [4]. Every year, there are roughly 2 million new cases of colorectal cancer worldwide, with over 40,000 in the UK [5,6]. Over 16,000 people will die from colorectal cancer each year in the UK, and rising incidence rates mean this is projected to increase to approximately 19,000 by 2038-2040, despite a fall in mortality rates [6]. This is alongside a worrying global increase in early-onset (< 50 years old) colorectal cancer, which is often diagnosed at an advanced stage, potentially due to a lack of cancer screening for this age group [7-9].

Colorectal cancer can develop through several different pathways, but the majority (70-90%) develop through the conventional adenoma-carcinoma pathway (Figure 1.1) [10,11]. Normally, epithelial cells are arranged into crypts inside the colon. Cancer development starts with a growth in tissue from an abnormal crypt starting in its base, where the stem cells are located. This occurs alongside losing the function of adenomatous polyposis coli (APC), a tumour suppressor gene, or other aberrations to the Wnt signalling pathway [12]. The benign tumour (adenoma) can develop into a cancerous tumour (carcinoma) and then invade locally and spread to other sites (metastasis), driven by deletions and mutations in tumour suppressor genes and oncogenes, such as tumour protein P53 (TP53) and Kirsten rat sarcoma virus

(KRAS) genes [10,11]. It takes approximately 10-15 years for it to develop through this pathway, before it can metastasise, ultimately leading to death if not stopped [10,11].

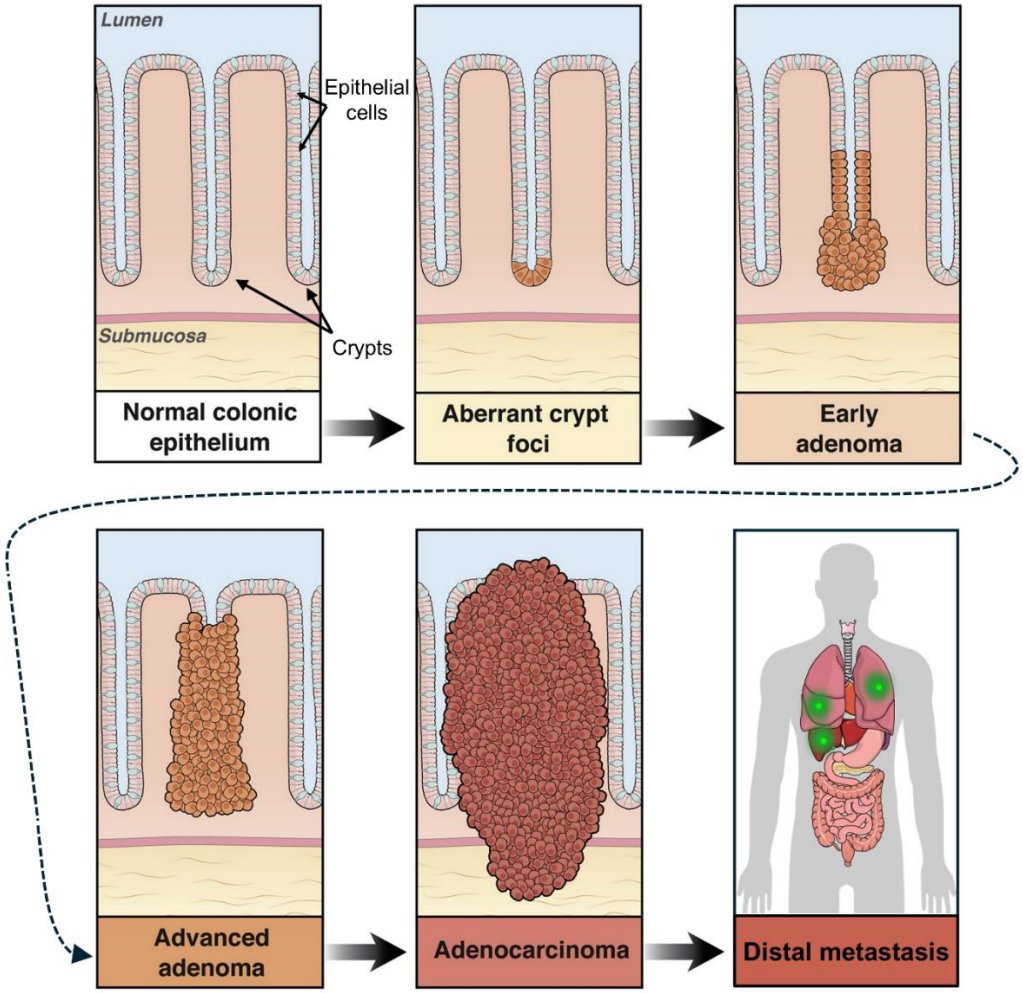

Figure 1.1. Colorectal cancer development via the conventional adenoma-carcinoma pathway. Schematic showing development from normal colonic crypts to metastatic cancer (left to right). Adapted from [13] with permission from Elsevier.

1.1.2 Diagnosis and management

Colorectal cancer patients are often asymptomatic until they have reached an advanced stage, which makes diagnosis more challenging [10,14,15]. However, there are a wide range of indicators and symptoms that may present, such as abdominal pain, bleeding from the rectum and changes to bowel habits. For accurate diagnosis, a faecal immunochemical test is usually performed. If this is positive, a colonoscopy and biopsy are carried out, often assisted by further imaging (e.g. MRI/CT scan of

the bowel) or lab tests (e.g. complete blood count test). Pathological examination of tumour samples removed during endoscopy or surgery is used for confirmation of the diagnosis, staging and further management of the disease. This includes visualising the microscopic structure of the tumour and testing for mutations to genes that serve as prognostic or predictive markers, such as KRAS.

Management of colorectal cancer is complex and includes using combinations of local (e.g. endoscopic resection, major surgery, radiotherapy) and systemic treatments (e.g. chemotherapy) [10,14]. The choice of treatment depends on the cancer stage. For patients with early-stage cancer, cure is possible and normally includes endoscopic resection or major surgery to remove the tumour. Surgery may be combined with neoadjuvant (before surgery) or adjuvant (after surgery) chemotherapy, particularly for cancers at a later stage.

Approximately 1 in 5 colorectal cancer cases in England are diagnosed when the cancer is metastatic (stage IV), at which point the 5-year net survival rate is only about 1 in 10 [16,17]. As only a small number of metastatic patients will be cured, it is important to decide if treatment will be curative or palliative [10,14,18]. Some patients may have only a few metastases, which may be resectable depending on their size and location. For example, metastases in the peritoneum are usually seen as untreatable, whereas ablative and surgical treatments may be considered for metastases in the liver or lung. For patients with unresectable metastases, systemic treatment can be used to shrink metastases for resection or to slow progression for palliative care. The choice of systemic treatment depends on numerous factors. For example, chemotherapy type is determined by the presence of genetic mutations (e.g. mutations to RAS or RAF, rapidly accelerated fibrosarcoma) and whether the

tumour is in the rectum or the left side of the colon (left-sided) or in the right side of the colon (right-sided). Further, immunotherapy can also be considered for patients with mismatch repair deficiency (dMMR) or high microsatellite instability (MSI-H) [10,14]. Patients may undergo several lines of treatment depending on changes to their mutation status, comorbidities, and response to treatment.

Systemic treatment for unresectable metastatic colorectal cancer, which is not MSI-H or dMMR, typically uses different combinations of chemotherapies (fluorouracil, oxaliplatin, irinotecan) and antibodies [10,14]. These antibodies target vascular endothelial growth factor (VEGF) or epidermal growth factor receptor (EGFR) [10,14]. Anti-VEGF antibodies prevent angiogenesis, the formation of new blood vessels, and anti-EGFR antibodies prevent downstream signalling and cell proliferation [9,18-21]. Anti-EGFR antibodies may be preferred over anti-VEGF antibodies when aiming to shrink metastases to make them resectable [10,14,22]. Currently, anti-EGFR treatment is only recommended for tumours that are left-sided and do not have mutations, or in other words are wild-type (WT), to the RAS family of genes [10,14,18,22,23]. Anti-EGFR treatment can be less effective for patients with mutations to BRAF (RAF isoform b), but may be considered for these patients when combined with a BRAF inhibitor [10,14,18,22-24].

Around 40% of patients who receive anti-EGFR treatment do not respond, as determined by a radiologist [25]. These patients may also suffer negative side effects and have poorer outcomes than if they had not received anti-EGFR treatment [25-27]. Identifying new predictors is important to avoid unnecessary treatment for these patients and to avoid the significant financial cost of this treatment to the healthcare system [25,27]. Better prediction methods could also lead to a greater understanding of

the underlying mechanisms driving the response to anti-EGFR treatment. To explore this further, it is necessary to understand more about EGFR and the mechanisms affecting its activation and the resulting downstream signalling.

1.1.3 EGFR activation and downstream signalling

EGFR (ErbB1) is a transmembrane protein, comprised of extracellular, transmembrane and intracellular domains, where the latter includes a kinase domain and a C-terminal tail (Figure 1.2a) [28]. It is part of the ErbB family of proteins, a group of receptor tyrosine kinases, which also includes ErbB2 (Her2), ErbB3 (Her3) and ErbB4 (Her4), where each receptor shares a similar structure [28]. Without its ligands, EGFR exists in an equilibrium between monomer and dimer, with minimal kinase activity (Figure 1.2a,b) [28-32].

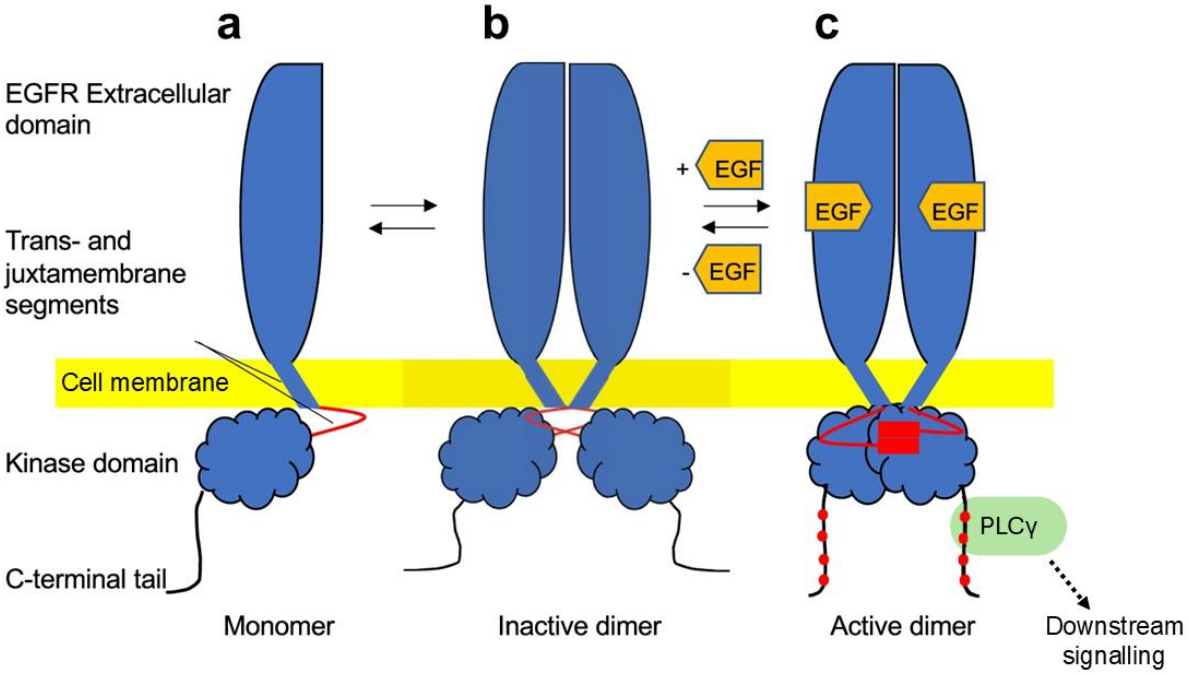

Figure 1.2. EGFR structure and activation. a,b EGFR exists in a monomer-dimer equilibrium without its ligands. **c** Ligand binding stabilises the dimer, which promotes dimerisation. Tyrosine residues are autophosphorylated (red circles), allowing them to recruit signalling molecules, such as PLCγ. This initiates downstream signalling pathways. PLCγ: phospholipase C-γ, EGF: epidermal growth factor. Adapted from [32] under a Creative Commons License (Attribution 4.0 International).

Ligands bind to the extracellular domain of EGFR, causing conformational changes in the receptor, promoting and stabilising dimerisation and higher-order oligomerisation (Figure 1.2c) [28-34]. These ligands include epiregulin (EREG), amphiregulin (AREG), transforming growth factor alpha and epidermal growth factor (EGF) [33]. Tyrosine residues in the intracellular C-terminal tail are autophosphorylated, allowing these sites to recruit proteins containing Src homology 2 (SH2) and phosphotyrosine-binding (PTB) domains (Figure 1.2c) [28-30,32-35]. The recruited proteins may be directly involved in signalling (e.g. phospholipase C-γ, PLCγ) or may be docking proteins (e.g. fibroblast growth factor receptor substrate-2), which recruit further downstream signalling molecules [35]. This initiates signalling pathways that regulate multiple downstream effects, including cell division, migration and survival [20,28-31,33,34,36,37].

Clustering of EGFR into dimers and higher-order oligomers is critical to downstream signalling [30,31,34,38,39]. These higher-order oligomers (larger clusters) may facilitate and amplify signalling, by allowing one ligand to stimulate multiple receptors, as the non-ligand bound EGFR can be activated by clustering with ligand-activated EGFR [40,41]. Alternatively, these larger clusters may amplify signalling by making it easier for receptors to be recognised by new ligands, or by promoting dimerisation due to the reduction in the inter-EGFR distances [40-42].

More broadly, the spatial organisation of EGFR and its ligands is associated with the activation and downstream signalling of EGFR. For example, ligand-binding and clustering of EGFR may be asymmetrically distributed within a cell, leading to regions with greater activation and downstream signalling of EGFR, such as the cell periphery or membrane regions with high curvature [42-46]. The organisation of EGFR

and its ligands in the cell interior could also be worth investigating, but this was beyond the scope of this work, which primarily focuses on the organisation at the membrane. Further, *in vitro* experiments that arrange the ligands for EGFR into artificial patterns have shown that the spatial organisation (e.g. clustering) of the ligands may affect downstream signalling [47,48]. Similar experiments have also shown that the amount, spacing and clustering of ligands can impact receptor signalling in other ligand-receptor systems [49,50].

1.1.4 Targeting the EGFR pathway in cancer

Cancer cells can use the EGFR signalling network to proliferate, survive, and metastasise. In multiple cancers, overexpression of EGFR and its ligands and constitutive activation of EGFR (independent of ligand binding), due to mutations in EGFR for example, can lead to hyperactivation of this network [20,33,37,51,52]. Tumour cells and tumour stroma can also produce the ligands for EGFR, which leads to overproduction of the ligands and hyperactivation of EGFR [33,52]. This has given rise to treatments that target the EGFR pathway in cancer.

There are two types of molecules used in cancer treatment that target EGFR: tyrosine kinase inhibitors and monoclonal antibodies [19,33,51,53]. Tyrosine kinase inhibitors bind to the intracellular tyrosine kinase domain, preventing phosphorylation and ultimately downstream signalling [19,51]. Monoclonal antibodies bind to the extracellular domain of EGFR instead of its endogenous ligands, preventing dimerisation, phosphorylation and ultimately downstream signalling (Figure 1.3) [19,21,51,52,54]. Cetuximab and panitumumab are two monoclonal antibodies used for anti-EGFR treatment of metastatic colorectal cancer [10,14,18,22,23]. Cetuximab is a chimeric immunoglobulin G1 (IgG1) monoclonal antibody (mouse/human), whilst

panitumumab is a fully human monoclonal antibody of IgG2 subtype [19-21,33,51,52]. Cetuximab can also inhibit cancer progression by promoting EGFR internalisation and subsequent degradation, and through antibody-dependent cellular cytotoxicity (ADCC) via immune effector cells [21,52,53,55-57].

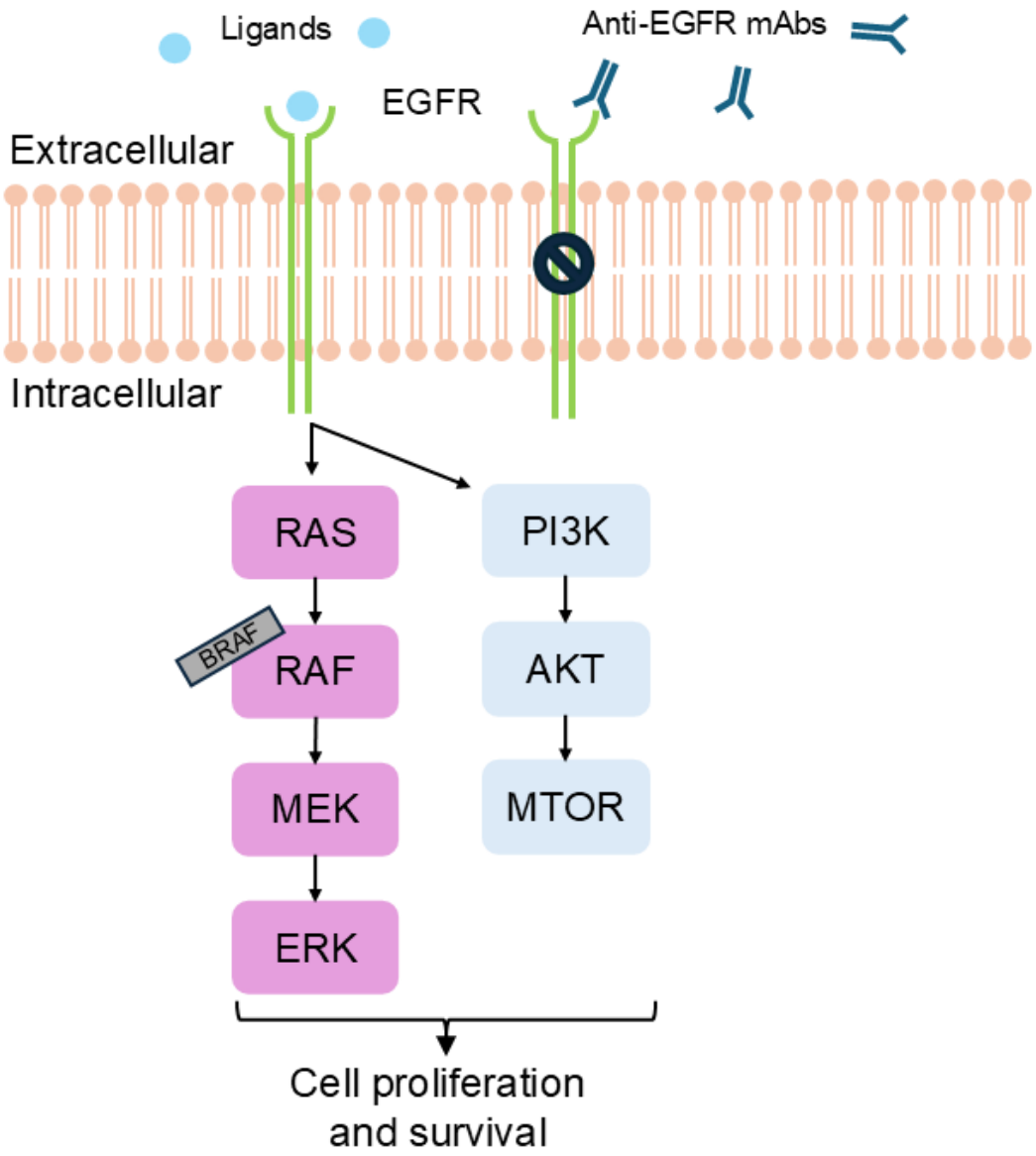

Figure 1.3. Cell signalling via EGFR. Ligands of EGFR, such as EREG and AREG, bind to the extracellular domain of EGFR, which leads to activation of downstream signalling pathways such as RAS-RAF-MEK-ERK and PI3K-AKT-MTOR. Anti-EGFR monoclonal antibodies (mAbs), such as cetuximab and panitumumab, bind to EGFR, preventing downstream signalling. RAS: rat sarcoma virus. RAF: rapidly accelerated fibrosarcoma. MEK: mitogen-activated protein kinase. ERK: extracellular signal-regulated kinase. PI3K: phosphoinositide 3-kinase. AKT: protein kinase B. MTOR: mammalian target of rapamycin. Adapted from [58] with permission from AACR.

Cancers can evade anti-EGFR treatment through multiple mechanisms. This can be through mutations to genes in the downstream signalling pathways for EGFR, such as the RAS/RAF/MEK/ERK and PI3K/AKT/MTOR pathways (Figure 1.3) [20,21,59]. This can allow for EGFR-independent activation of these pathways, for example, by putting downstream signalling proteins in an active state independent of EGFR signalling, thereby providing intrinsic resistance to anti-EGFR treatment [20,21,59]. As mentioned previously, this has been incorporated into patient selection for metastatic colorectal cancer patients receiving anti-EGFR treatment, with testing for mutations to the RAS and RAF family of genes, or more specifically to KRAS, Neuroblastoma RAS (NRAS) and B-RAF [20,21,59]. Testing for overexpression of HER2 is also being incorporated into patient selection, as cancers can evade anti-EGFR treatment by activating downstream signalling pathways through HER2 instead [10,14,18,22,23,59].

Other emerging predictors and mechanisms for resistance to anti-EGFR treatment include mutations and/or changes to the expression of EGFR and its ligands, and increased activation of alternative pathways such as insulin-like growth factor and mesenchymal–epithelial transition factor [20,21,59]. EGFR mutations and decreased EGFR gene copy number have also been linked with a poorer response to anti-EGFR treatment, where gene copy number is measured by counting the number of copies of the EGFR gene in tumour cells, which often have more than the usual 2 copies of each gene per cell [20,21,23,59-61]. However, the association between EGFR gene copy number and protein expression is still not clear [62-65]. While the level of EGFR protein expression is not associated with response to treatment, the expression levels of the ligands for EGFR have emerged as promising predictors [20,21,23,59,60].

Increased mRNA and protein expression levels of EREG and AREG have been associated with an increased response to anti-EGFR treatment [20,21,58,66,67]. This could be because tumours sensitive to anti-EGFR treatment overproduce the ligands of EGFR [52,68,69]. Recently, immunohistochemistry (IHC) imaging and analysis of fixed-formalin paraffin-embedded (FFPE) tissue generated by the PICCOLO trial identified that high AREG/EREG protein expression predicted a positive response to anti-EGFR treatment (panitumumab) for RAS-WT patients [25,58,70]. The protein expression of EREG and AREG were combined into one biomarker, which was negative if the expression of both was low and positive otherwise [25]. This has been further verified on a dataset of patients that received anti-EGFR treatment (cetuximab or panitumumab) at one of eight different UK cancer centres [70]. This is being further tested in the ARIEL trial, which is focusing on right-sided RAS-WT patients, as right-sided tumours are usually associated with poorer response to anti-EGFR treatment [22,23,58,71,72].

Despite these advances, new predictors are still required to better identify patients who will benefit from anti-EGFR treatment. The spatial organisation of EGFR and its ligands is associated with downstream signalling and activation of EGFR, as discussed earlier. Therefore, it could also be associated with or used to predict response to anti-EGFR treatment. There is already evidence that localisation of EGFR in the nucleus is associated with resistance to anti-EGFR treatment in cell lines and poorer outcomes for cancer patients [73,74]. Further, in breast and lung cancer, the spatial organisation of signalling molecules, including the clustering and density of HER2, has been associated with response to treatment [75,76].

Crucially for this study, the higher-order clusters of EGFR (tetramers, octamers, decamers, and above), which appear to be critical to downstream signalling (Section 1.1.3), are approximately ~20-50 nm in size [31,38,39,41,46]. This is further corroborated by observations that ligand-ligand spacing on the order of 30-40 nm is also crucial to EGFR activation [48]. The clustering of HER2 that has been associated with patient response to treatment is also of approximately the same size [75]. Further, it has been suggested that these higher order oligomers of EGFR depend on larger features of the cell for their formation, such as regions of high membrane curvature or lipid rafts, where the latter are approximately 10-200 nm in size [42,46]. Therefore, this complex pattern of higher-order clustering of EGFR and its spatial organisation within each cell warrants further investigation as a potential predictor of response to treatment.

To investigate the spatial organisation of EGFR or its ligands in tumour on this scale, high-resolution imaging of the proteins in patient tissue is required.

1.2 Imaging

Conventionally, tissue from cancer patients is fixed, dehydrated, embedded in resin and sectioned to generate thin slices for staining and imaging using light microscopy [77]. Sections are usually stained with haematoxylin and eosin (H&E), which stains the DNA and RNA (acidic structures) a purple colour and proteins in the cytoplasm/extracellular matrix (generally basic) a pink colour. Alternatively, IHC can be used to label specific proteins with antibodies [78]. However, the resolution of a conventional light microscope is limited by the diffraction of light to at best 200 – 300 nm, meaning that the detailed spatial organisation of proteins cannot be resolved [79]. Electron microscopy can increase the resolution to about 0.2-0.5 nm, but requires complex sample preparation, has low throughput and is very expensive [78,80]. Instead,

fluorescence microscopy can be used to characterise the nanoscale organisation of EGFR and its ligands using advanced techniques such as single-molecule localisation microscopy [31,38,39,41,42,46,48,81].

1.2.1 Single-molecule localisation microscopy (SMLM)

Single-molecule localisation microscopy (SMLM) is a type of fluorescence-microscopy that can provide high spatial resolution for fluorescent objects and could be integrated into routine clinical practice [82-84]. This technique can reveal the organisation of subcellular structures, which has led to breakthroughs in cell biology [83,85]. For example, this revealed for the first time that actin forms rings within the axons of neurons [86].

Crucially for this study, SMLM has been widely used to characterise the organisation of EGFR and its ligands over the required length-scale (20-50 nm) [31,38,39,42,48,81]. Further, we had available a commercial SMLM microscope (ONI, UK), which had already been successfully integrated into a workflow for analysing nanoscale protein organisation in tissue samples from patients [84]. This microscope used a type of SMLM, direct stochastic optical reconstruction microscopy (dSTORM), which has been used to investigate EGFR and its ligands [31,38,42]. Further, importantly for this study, SMLM studies of EGFR organisation have used secondary antibodies to resolve the organisation over our required length scale, as later used in this study (Chapters 2 and 4) [81]. dSTORM has also been used to resolve the clustering of HER2 in tissue from breast cancer patients and reveal associations between this organisation and response to treatment [75].

SMLM methods repeatedly image subsets of the fluorophores in a sample that are spatially well separated, allowing their high-precision positions to be calculated

(Figure 1.4). In a light microscope, the image generated by a point object is spread out by diffraction through the optics. As the images that are formed from multiple points overlap, the points and their locations cannot be resolved below a resolution of $\frac{\lambda}{2a}$, where λ is the wavelength of light and a is the numerical aperture (Figure 1.4b) [87]. SMLM methods ensure that only a small subset of the fluorophores are 'ON' (fluorescent) in each frame (so-called blinking behaviour). The 'ON' molecules are spatially separated enough for their positions to be identified to a high precision (Figure 1.4c). By imaging over multiple frames (typically 10,000 or more), different subsets of molecules are 'ON' and imaged in each frame. This generates a list of coordinates (or localisations) for the fluorophores in a sample to a high precision (20-50 nm or better) (Figures 1.4d,e). A final image can be reconstructed from these *xy* (or *xyz*) positions, commonly termed a histogram.

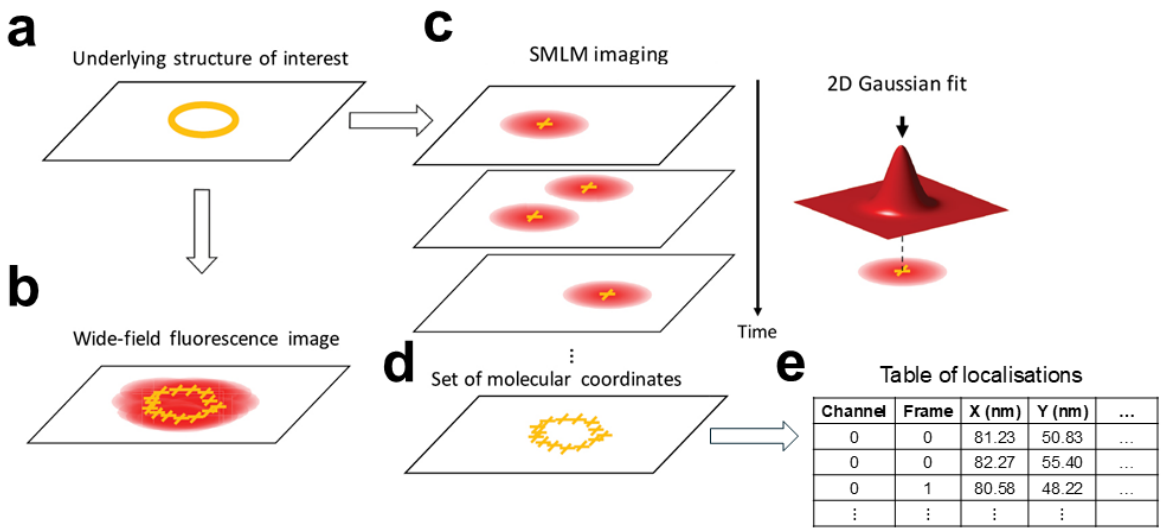

Figure 1.4. Principle of single-molecule localisation microscopy (SMLM). a Underlying structure that is being imaged. **b** Conventional imaging of the sample via wide-field fluorescence. Each fluorophore (yellow cross) appears as a spot of light (red spot). **c** SMLM builds up an image of the sample over time, from multiple frames. In each frame, fluorophore coordinates are calculated by fitting a 2D Gaussian to each non-overlapping spot. **d** Coordinates for the fluorophores (localisations). **e** Localisations are represented in a table, ready for downstream analysis. Adapted from [88] under a Creative Commons License (Attribution 3.0).

There are many different SMLM methods, including direct stochastic optical reconstruction microscopy (dSTORM), photoactivated localisation microscopy (PALM) and point accumulation in nanoscale topography (PAINT) [82,83,89-93]. They all differ in how they restrict the number of 'ON' fluorophores in each frame (all require laser irradiation of the fluorophores). dSTORM uses photoswitchable dyes that reversibly switch between emitting and not-emitting states [90]. This usually requires the use of a buffer that contains a thiol reagent (such as β-mercaptoethanol) and an oxygen scavenger (glucose oxidase) to increase the number of fluorophores in the dark state [82,83,90]. Newer blinking dyes that blink at rates suitable for dSTORM in phosphate-buffered saline, such as JF635b, have also been developed [94]. PALM uses photoactivatable (e.g. photoactivatable GFP and mCherry) or photoconvertible fluorescent proteins (e.g. mEos3), which are stochastically switched from 'OFF' to 'ON', or from emitting in the 'green' to emitting in the 'red' part of the spectrum, imaged and then bleached [82,83,91,92]. PAINT uses dyes or dye-labelled ligands that freely diffuse through the sample, appearing as blurred background until they temporarily or permanently bind to the target for long enough to be localised [82,83,93]. In DNA-PAINT, a popular version of PAINT, the fluorophores are attached to oligonucleotides that can transiently hybridise with complementary DNA strands, which are attached to the object of interest, normally using antibodies [82,83,95].

Detection and localisation of each molecule in the sample can also proceed via a variety of methods implemented in different software packages [83,96-98]. For example, after removing background, emissions can be detected by identifying the local maxima in the diffraction limited image or by calculating the correlation between the image and the expected emission pattern for a molecule. Single-molecule

localisation, where the *xy(z)* position of each molecule from each emission pattern is calculated, normally uses maximum likelihood estimation. This calculates the coordinates for each molecule that maximises the probability of generating the observed emission. However, different algorithms vary in how they estimate the expected emission pattern for a localisation or how they process the background. For high signal-to-noise ratio data, maximum likelihood estimation can approach the theoretical limit for the precision of each localisation, $\sigma_{loc,}$ which can be estimated using the Cramér-Rao lower bound, $\sigma_{loc} \geq \sigma_0/\sqrt{N}$, where σ_0 and N are the width and number of photons respectively for the emission [83,99]. The data is normally further processed to remove sub-optimal localisations, for example, by grouping localisations in consecutive frames that are nearby to each other and therefore likely emerging from the same molecule, or to correct for systematic errors such as sample drift during acquisition.

1.2.2 Quantitative analysis of protein organisation via SMLM

Unlike conventional imaging data, SMLM generates a set of fluorophore positions (localisations), often described as a point cloud. This data can be reconstructed into an image, such as a 2D (or 3D) histogram, for analysis via the multitude of image-based algorithms [100-102]. The choice of data representation should reflect the required precision for the analysis. For example, rendering the data as an image, and therefore binning the localisations into pixels, requires large images (e.g. 1 pixel per nm) to incorporate SMLM data of high-precision, as it otherwise risks introducing artifacts and sacrificing the potential gain in spatial precision SMLM provides over conventional imaging [100,103]. This rendering does not scale well for larger regions of interest or when moving from 2D to 3D, as the size of the image increases exponentially [103]. Instead, quantitative analysis of SMLM data usually makes direct

use of the point localisations and includes analysis of the spatial distribution of molecules (e.g. clustering and colocalisation), segmentation, measuring the number of protein molecules in target structures, single particle tracking and classification, for example (Figure 1.5) [104-107].

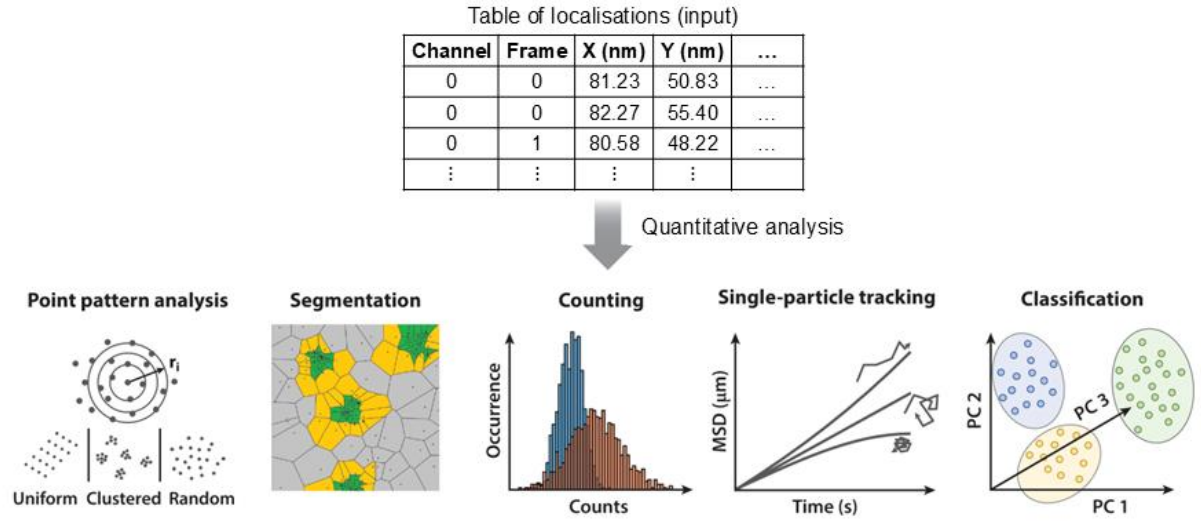

Figure 1.5. Quantitative analysis of SMLM data. Examples of techniques used to analyse SMLM data (table of localisations). MSD: mean squared displacement, PC: principal component. Adapted from [107] under a Creative Commons License (Attribution 4.0 International).

Analysis often focuses on quantifying clusters of localisations, which can be identified using many different approaches, including statistical-, density- and Voronoi Tessellation-based methods [104-107]. Cluster analysis has been widely used to study proteins, such as transferrin receptor or tyrosine-protein kinase Lck, at the plasma membrane [108-110]. This has been driven in part by the association between clusters of membrane proteins and activation of downstream signalling pathways; for example, it has been shown that clustering of EGFR changes upon ligand stimulation [31,38,39,42,111,112]. Cluster analysis has also revealed differences in density and clustering of HER2 between trastuzumab-sensitive and -resistant cancer cell lines and the effect of ligand stimulation on both [75,113]. In other cancer cell lines, this form of analysis has also revealed the changes to clustering of membrane proteins,

including EGFR and HER2, after addition of anti-EGFR (cetuximab) or anti-HER2 antibodies (trastuzumab and pertuzumab) [39,114,115]. The distribution of nearest neighbour distances for EGFR imaged using SMLM can also provide a more detailed (1-20 nm range) understanding of EGFR organisation within clusters [112]. For example, this can indicate the frequency and make-up of larger molecules (oligomers) formed from multiple EGFR monomers (Figure 1.2). While cluster analysis is focused on identifying and analysing multiple molecules in close spatial proximity, artifacts from SMLM imaging, such as repeated blinking and localisation of a dye in multiple frames, can lead to artificial clusters (pseudoclusters) [105]. Distinguishing genuine clusters from pseudoclusters is not straightforward, but can be achieved in some cases through careful pre-processing of the data, for example by modelling the photophysics of the blinking fluorophore [116].

1.2.3 Characterising and classifying cancer from SMLM data

The spatial organisation of proteins at the nanometre scale, derived from SMLM data, has been used to characterise and classify cancer in cell lines. This can reveal differences in the colocalisation and clustering of membrane proteins between cells from normal vs. cancer cell lines [117-120], or between cancer cell lines with different expression of the membrane protein [121,122]. These differences can be quantified through features such as the cluster size, cluster circularity and the localisations per cluster (per area) [117,119,121,122]. For EGFR, this approach has been used to show that epithelial cells from a pancreatic cancer cell line have more EGFR-enriched extracellular vesicles than epithelial cells from a normal pancreas cell line [118]. In colorectal cancer cell lines, this approach has revealed differences in the spatial organisation of clathrin (a protein involved in internalisation of receptors at the cell membrane) between cells with different metastatic potential [123].

Analysis of SMLM data has also been used to characterise cancer phenotype using patient samples, from primary cells [119,120,124], exosomes/extracellular vesicles [118,125,126] and tissue (Figures 1.6a-c) [75,117,119,120,122,127-130]. This has mainly focused on comparing protein organisation between normal and cancer patient samples [117-120,124-126,128,130,131]. For example, analysis of dSTORM imaging data for EGFR has shown that it forms more and larger clusters in primary epithelial cells isolated from lung cancer tissue compared to normal lung tissue [124], and that there are more EGFR-enriched extracellular vesicles in patient plasma from pancreatic cancer patients compared to healthy patients [118]. Beyond this, analysis of STORM imaging data of FFPE tissue samples from colorectal cancer patients has shown how the higher-order structure (clustering) of chromatin in nuclei is disrupted during cancer progression (Figure 1.6c) [127,129]. Changes to this clustering, not possible to identify by conventional fluorescence microscopy, could be used as a biomarker to identify patients at greater risk of more aggressive disease progression [127,129]. High HER2 density and clustering have been associated with response to anti-HER2 treatment (trastuzumab) for breast cancer patients, by analysing dSTORM data of patient tissue, which is of particular relevance to this thesis [75].

Other quantitative analysis methods can be applied to SMLM data to characterise cancer and other diseases that do not use protein organisation at the nanometre scale. Protein expression levels measured from SMLM data has been used to classify different types of breast cancer (cell lines), to identify myeloma cells that may benefit from immunotherapy (cell lines and primary cells) and to classify patients into different cancer types (exosomes from patient blood) [132-134]. However, measuring absolute protein expression from SMLM data can be unreliable without

careful calibration [135]. Alternatively, features of protein organisation, such as cluster shape, can be more reliably compared between samples imaged under the same conditions.

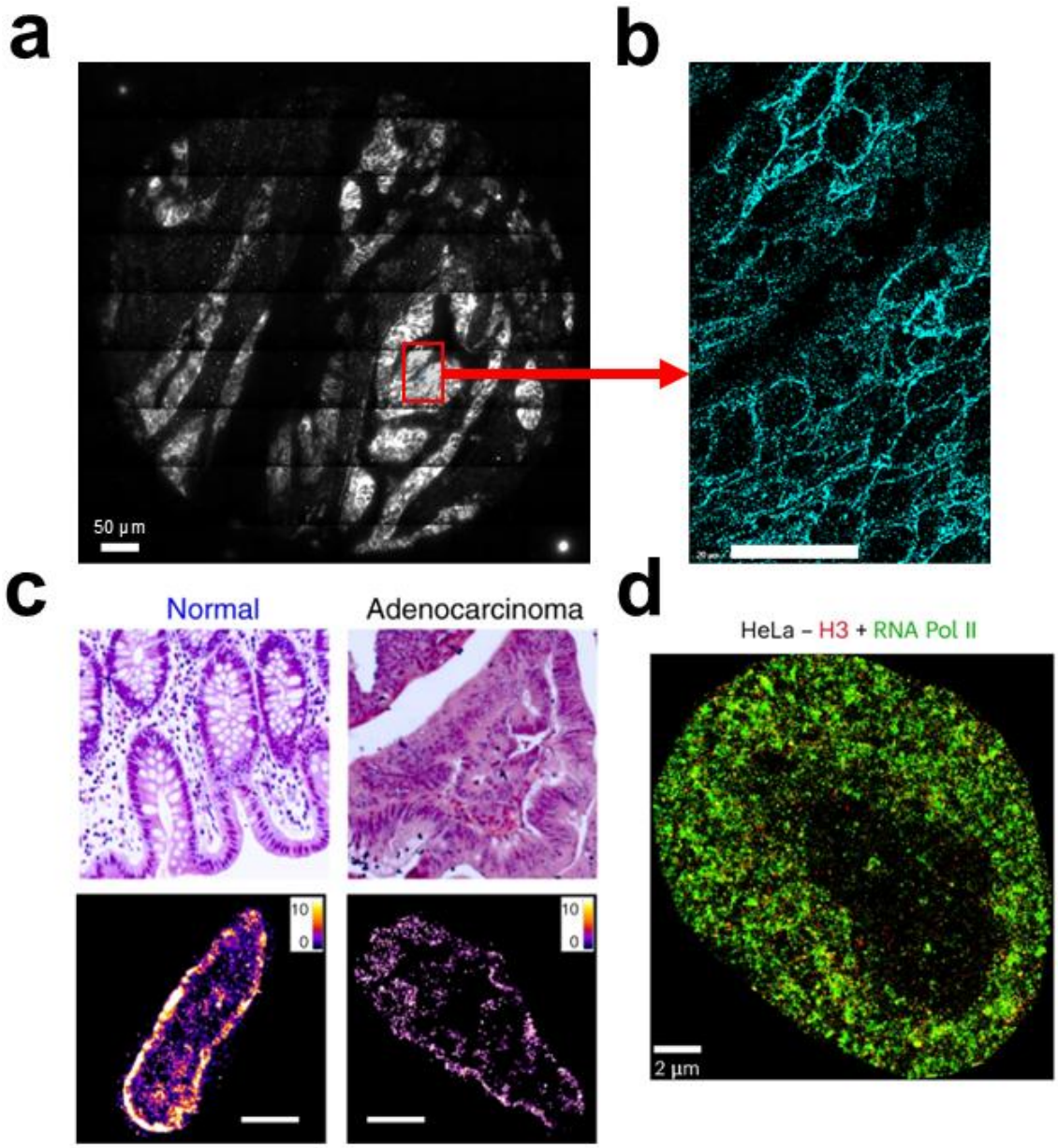

Figure 1.6. Examples of SMLM imaging of cancer. a-b Widefield and paired dSTORM image of colorectal cancer patient tissue stained for epiregulin. Core in **a** from a tissue microarray, derived from a formalin-fixed paraffin-embedded (FFPE) block. **c** Haematoxylin and eosin and paired STORM imaging of heterochromatin structure in colorectal cancer patient tissue. Normal tissue is over 10 cm away from the tumour. Reproduced from [127]. **d** HeLa (cervical cancer cell line) cell, stained for nuclear proteins histone H3 (red) and RNA polymerase II (green) and imaged via STORM. Reproduced from [136]. **a**,**b** were generated in this study. **c**,**d** are reproduced under a Creative Commons License (Attribution 4.0 International). Scale bars: **a** 50 µm **b** 20 µm **c** 2 µm **d** 2 µm.

Recently, methods that render the SMLM data into an image have been used to identify cervical cancer cells (cell lines, Figure 1.6d) and to predict diagnosis for patients with different renal diseases (FFPE tissue samples) [136,137]. While these methods may allow consideration of protein organisation over a whole cell or field of view, by rendering the data into an image, they may not exploit the full spatial resolution of the data and consider the nanometre-scale organisation.

In summary, SMLM imaging can identify features of nanometre-scale protein organisation, which could help in the diagnosis, prognosis and prediction of response to treatment in multiple cancers [131]. By providing a high level of detail, SMLM may be able to identify new predictors that cannot be identified by existing techniques [127,129,131,133]. Furthermore, this technique is not limited to characterising and classifying cancer. For example, dSTORM has revealed differences in the organisation (clustering) of membrane receptor proteins: in human tissue from healthy patients and those with major depressive disorder; and in mouse tissue when subjected to inflammatory pain [138,139].

1.3 Artificial Intelligence

Despite the promise of SMLM data and analysis, it is not straightforward to manually identify patterns that differentiate samples, beyond simple cluster metrics. As an alternative, Artificial Intelligence (AI) could be used to automatically learn which patterns (or features) of the protein organisation are associated with disease. AI has been used to classify SMLM data of cancer and kidney disease from patient samples, using the protein expression data in a machine-learning model or using a deep-learning approach that renders the SMLM data into an image [134,137]. But it also

has the potential to classify and characterise cancer, while still considering the nanometre-scale organisation.

1.3.1 What is AI, and what tasks can it help to address?

Defining AI in its broadest sense is difficult, as there is no single definition of intelligence accepted by philosophers, scientists, engineers, etc. and consequently no single accepted definition of AI, nor indeed what artificial means in this context [140,141]. AI could be defined by the characteristics that it might have, such as thinking and acting humanly and rationally [142]. For computer scientists, the current definition of AI is more constrained, defining a group of techniques that use computers to perform tasks that would normally require human-like (not human-level [143]) intelligence [144,145]. To perform these tasks, AI may require several capabilities, such as the ability to answer questions and infer new findings from stored data (automated reasoning) or to learn from experience and adapt to new situations (machine learning) [142].

Machine learning approaches in particular have been used to address a wide variety of problems (Figure 1.7a). Machine learning algorithms learn from data (model training), extracting information that has not been explicitly encoded by humans [142,146]. At the forefront of this is deep learning, a type of machine learning that requires little human involvement, driven by its ability to exploit the increase in large datasets suitable for training (Figure 1.7a) [142,144-147]. Deep learning algorithms represent complex inputs using combinations of simpler representations automatically learnt from the data [146]. For example, for an image of a cell, the representation could combine representations for the nucleus and membrane, which

could in turn be represented by combinations of simpler features in the image data, such as contours, edges, or corners.

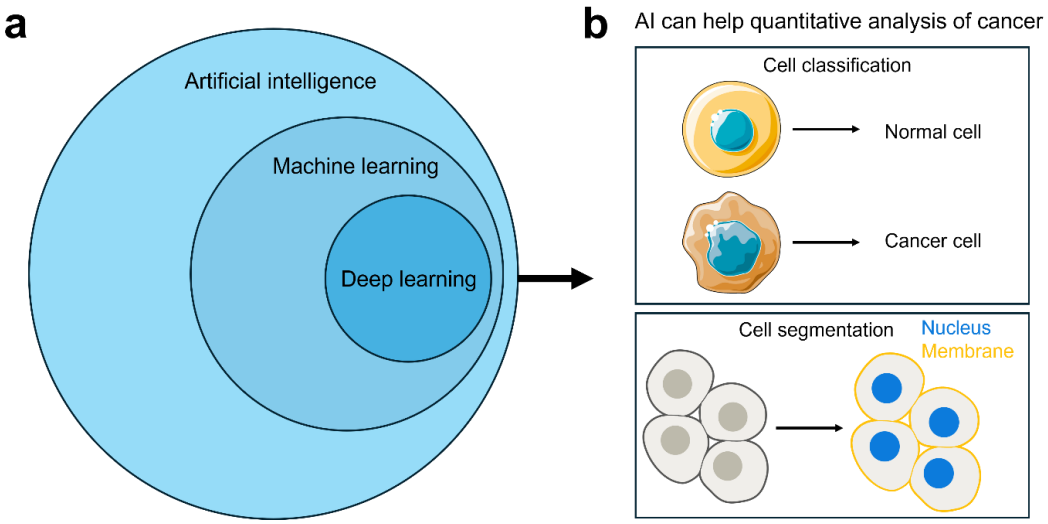

Figure 1.7. Using AI to help quantitative analysis of cell biology and cancer. a AI field and its subdisciplines. **b** Examples of quantitative tasks in cell biology that AI could assist with. Cell drawings for the cell classification panel in **b** provided by Servier Medical Art (https://smart.servier.com/) under a Creative Commons License (Attribution 4.0 International).

AI can be used to address a variety of problems, such as classic computational and bioimage analysis tasks, segmentation, and classification (Figure 1.7b). Segmentation is defined as breaking down a larger input into smaller regions representing the same thing, for example, identifying the pixels that belong to the nucleus in the image of a cell and those that do not [142,145]. Classification is defined as assigning an item to a class or category, for example, identifying if a cell is cancerous or healthy from an image [142,145]. These are common approaches used in the quantitative analysis of cancer [145]. For example, segmentation (of cells) and classification of immunohistochemistry data are key steps in pipelines used to predict the risk of recurrence or response to treatment for colorectal cancer patients [148,149].

The algorithm used to address a task depends on the type of input data. For SMLM, the raw data after protein localisation and preprocessing is a point cloud of

localisations, which can be acted on by both simpler machine learning models and deep-learning models (Figure 1.8).

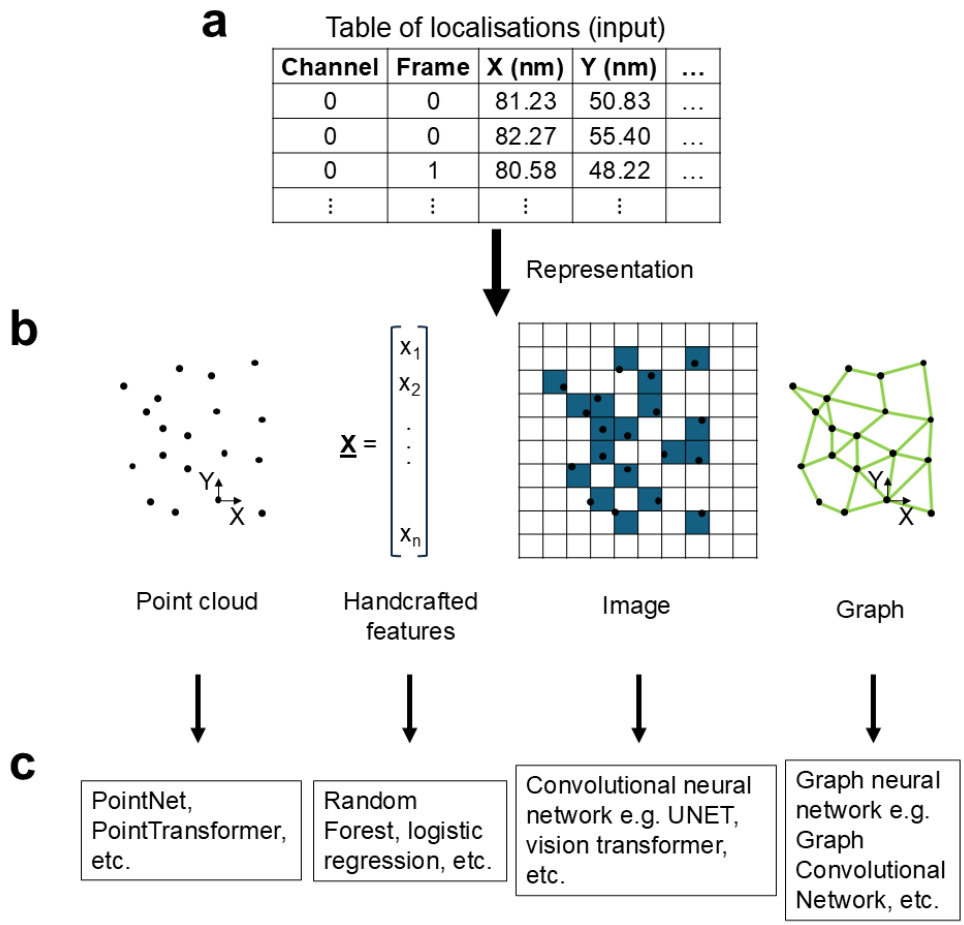

Figure 1.8. Applying AI algorithms to SMLM data. a Raw SMLM data for analysis. **b** Possible representations of the raw SMLM data. **c** Algorithms used for downstream analysis, depending on the choice of representation.

The simpler machine learning models, such as logistic regression or random forest models, act on features manually calculated from the point cloud (handcrafted features), such as the size of protein clusters [150]. Alternatively, deep learning models can act directly on the point cloud, image or graph representation of the data [103,151]. These deep learning models do not require handcrafted features, as they can automatically extract features from the data. In theory, this could make them more accurate than simpler machine learning models, as the features are not pre-determined and can adapt to the task at hand. However, in general, this hypothesis

was not supported in this study, and the increased complexity of these features also made them harder to interpret (Chapters 3 and 4).

The archetypal deep-learning model is the feedforward neural network (Figure 1.9a) [142,146]. These networks contain units (or neurons) arranged into a series of layers. Each unit combines the values of units in the previous layer, according to the weights (parameters) of the model. A non-linear activation function is then applied to each unit to give its updated value. The optimal weights in the model are learnt by comparing the output with the target (e.g. predicted vs ground-truth class) using a loss function. This gives an error which is backpropagated through the network to update the weights.

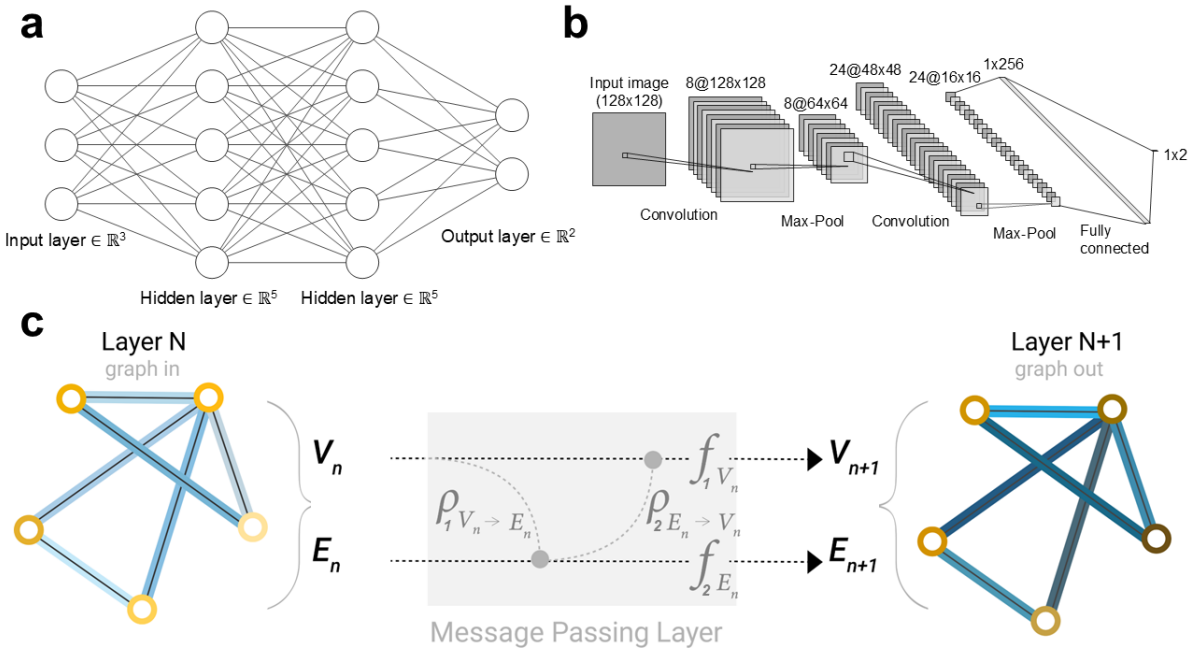

Figure 1.9. Deep learning models. a Example feedforward neural network (or multi-layer perceptron) acting on an input vector. Each circle is a unit (neuron), and each line is a weight (parameter learnt by the model). **b** Example convolutional neural network (CNN) acting on an image [152]. **c** Typical message passing layer in a graph neural network. ***Vn*** and ***En*** are the node and edge features in layer N, which are updated by the layer to give ***Vn+1*** and ***En+1***. ρ_1 and ρ_2 are functions (e.g. maximum pooling) that pool the node features to the edges, $V_n \to E_n$, or edge features to the nodes, $E_n \to V_n$. f_1 and f_2 are functions, such as multi-layer perceptrons, that update each node and edge feature in isolation. **a,b** Produced using [153]. **c** Adapted from [154] under a Creative Commons License (Attribution 4.0).

Neural networks have been extended to act on images (structured 2D or 3D input) through the widely successful convolutional neural networks (CNNs) (Figure 1.9b) [155]. They have also been used for point clouds; a set of points that have no inherent ordering (i.e. {a,b,c} ≡ {b,c,a}), are invariant to transformations in space (e.g. 3D rotation does not change its class) and have coordinates in Euclidean space (i.e. the distance between points is meaningful) [156]. Neural networks have also been used for graphs, using message passing schemes to aggregate information over neighbouring nodes and edges [154] (Figure 1.9c). More recently, the attention mechanism was created to guide models into paying different amounts of attention to different parts of the input during the forward pass [157]. This has led to the widespread use of transformer architectures, which are based on the attention mechanism, for text [158,159], images [158,159], point clouds [160,161] and graphs [162].

1.3.2 Using AI for quantitative analysis of SMLM data

Handcrafted features of protein organisation have been combined with machine learning algorithms to characterise and classify SMLM data. Existing workflows such as ASAP, SEMORE, ECLIPSE, and SuperResNET all share a similar workflow [150,163-165]. First, they segment or cluster the data into smaller structures from which handcrafted features are extracted. Each method calculated between approximately 5-70 features per cluster, which could be grouped by what they represent, such as: symmetry, geometry, network properties, circularity, boundary properties, skeleton properties, texture and topology. Then, these features are used as inputs into machine-learning algorithms (e.g. random forest) for classification. For SMLM data from cell lines, these workflows have been used to classify different tau protein clusters that may be implicated in neurodegenerative diseases [164], to classify and characterise caveolae and noncaveolae scaffolds formed by caveolin-1 protein in

prostate and breast cancer cells [103,166-169] and to show changes to clathrin organisation in breast cancer cells after drug treatment [170]. Despite the success of these workflows, they have not been extended to the classification of structures larger than single clusters of localisations and therefore, are not suitable for classification of entire cells (Chapter 4).

The expression of subcellular proteins as measured by SMLM has also been used for classification by machine learning algorithms. For example, the expression and density of EGFR and other membrane proteins, as determined by SMLM, have been used by machine learning algorithms (random forests and linear discriminant analysis) to classify whether clusters come from cultured breast cancer cells that were EGF-stimulated or -starved [111], to classify two different breast cancer cell lines [132], and to classify patients by cancer type (pancreatic, breast or healthy) by imaging exosomes from blood samples [134]. For larger or more complicated structures, it may be challenging to design handcrafted features that separate the classes. In these cases, deep learning models that extract high-dimensional features automatically from the data may perform better. However, for the classification of larger structures such as whole cells in this thesis (Chapters 3 and 4), we found that handcrafting generally outperformed deep learning when extracting features from smaller structures (clusters of localisations) in the data.

Deep learning models that act on images (e.g. convolutional neural networks) have also been used to classify SMLM data [103,171]. These models have been used to classify cells as healthy or cancerous from cell lines and to diagnose renal diseases from FFPE patient tissue [136,137]. However, this requires the localisations to be rendered into an image, which introduces redundancy and may sacrifice the spatial

precision gained by using SMLM (Section 1.2.2) [103]. This could, in turn, reduce the classification accuracy, supported by evidence that rendering the SMLM data into a smaller image decreased the classification accuracy (although they also showed that beyond a certain size, there were no gains in performance either) [136].

Deep learning models that act directly on the point cloud or on graph representations that do not sacrifice the precision of the SMLM localisations are potentially a better alternative. For example, a point-based model (PointNet) has been used to classify clusters of caveolin-1 localisations as caveolae or noncaveolae scaffolds and to help align localisations in 3D [103,156,172]. Alternatively, representing the localisations as a graph (e.g. localisations are nodes and edges within a user-defined distance threshold) can allow graph neural networks to be used. For example, graph neural networks have been used on SMLM data to identify clusters of localisations [151], to identify noisy localisations in microtubule data [173] and to characterise single-molecule trajectories [174,175]. So far, however, point- and graph-based networks have not generated many biological insights, nor been used for classification of anything larger than clusters of localisations.

1.3.3 Can AI help us to understand more about biology?

By analysing successful classification models, AI may also be able to provide further insights into the underlying differences between different biological systems or states [176]. For example, post-hoc analysis of simple machine learning models may reveal the most important handcrafted features for classification, revealing the key difference between the classes. For SMLM data, evaluating the importance of features (co-expression of proteins) to a random forest algorithm, which predicted

whether breast cancer cells from a cell line were EGF-treated or not, indicated the proteins that EGFR may interact with after EGF stimulation [111].

Deep-learning algorithms also provide unique opportunities to reveal key biological differences between classes. This could be through identifying the key part of the input for the classification, revealing what distinguishes the sample. For example, applying occlusion and class-activation maps to an image-based model (convolutional neural network) that classified SMLM data of human-induced pluripotent stem cells and their somatic cells, identified the parts of the cells most relevant to their prediction [136]. While there are many algorithms for explaining image-based AI models, equivalent algorithms for point cloud neural networks are not as well developed and have not been used on SMLM data yet [177,178]. Similarly, algorithms designed to explain graph-based neural networks have not been tested on SMLM data [179-181]. Graph explainers have been successfully applied to diffraction-limited histopathology images. For example, GNNExplainer has been used to identity the most relevant nuclei and their features (e.g. size) for subtyping of breast cancer, using haematoxylin and eosin-stained images of tissue [182,183].

Alternatively, analysing the high-dimensional features generated by deep learning models (deep features) during classification can also provide further insight. This normally assumes that data from the same class or representing the same concept should be closer in feature space, and vice versa [175,184]. To proceed with this analysis, the deep features can be visualised in two- and three-dimensions using dimension-reduction methods such as PCA, t-SNE and UMAP [184]. Applying this to handcrafted features of SMLM data has revealed how well the features separate the classes, uncovered subpopulations of the classes (within the feature space) and

revealed which input features are most discriminative (by observing changes to the separation between classes in the feature space) [103,150,164]. These techniques have also been applied to deep features from SMLM data, for example, to reveal data points with unique characteristics by identifying isolated points in the feature space [103,175].

In summary, machine- and deep-learning models can help to characterise and classify SMLM data. The nanoscale organisation of membrane proteins from SMLM data has not yet been used by point-based machine-learning models to classify patients or fields-of-views from their tissue. Point- or graph-based networks acting on SMLM data could allow for cell and patient classification, predicting response to treatment for example, while still retaining the spatial precision of the localisations. For successful classification algorithms, algorithms for explaining the predictions may then be able to provide further biological insight.

1.4 Objectives of this study

The overall objective of this study is to investigate whether the nanoscale spatial organisation of EREG can predict response to anti-EGFR treatment for metastatic colorectal cancer patients, as current biomarkers do not accurately predict response for all patients. EREG was analysed due to the existing evidence that high protein expression levels of EREG (and AREG) are associated with response to anti-EGFR treatment [25,70]. AREG was not analysed due to difficulties in optimising the antibody for SMLM (unpublished).

The objective of this study leads to several key questions:

- Can SMLM, which can provide the necessary spatial resolution to resolve the nanoscale organisation, be used to image patient tissue as part of an approach for predicting response?
- Can AI automatically segment the membrane and cells from SMLM data of membrane proteins in sections from FFPE samples?
- Can AI automatically classify structures larger than clusters of localisations in the SMLM data, such as cells, while still retaining the precision of the localisations?
- Can existing explainability techniques for AI algorithms identify features of the organisation associated with response?

An AI-based approach was developed to address this using SMLM imaging data. This approach combined a novel segmentation pipeline, *locpix*, with a novel classification pipeline, *ClusterNet*.

- *locpix* was developed to extract cells from SMLM data of membrane proteins in sections from FFPE samples, either through manual annotation or image-based deep-learning models (Chapter 2). The dataset mimicked SMLM data obtained from patient tissue and was obtained using the same microscope that was later used to image the patient samples. The samples were stained for the membrane proteins EGFR and EREG.
- *ClusterNet* was developed to classify SMLM point cloud data using graph-based deep learning and to identify the structures that led to the classification (Chapter 3). This was tested on a model SMLM dataset that shared characteristics with SMLM data from biological samples.

- *locpix* and *ClusterNet* were combined and used to predict response to anti-EGFR treatment for cells and patients in a dataset of EREG localisations, which was collected by imaging tissue samples from metastatic colorectal cancer patients with dSTORM (Chapter 4). *ClusterNet* was compared against simpler machine learning models that used handcrafted features of the clusters and cells. Features of the organisation for EREG that were associated with a positive response to treatment were also identified.

2 *locpix*: Annotation and Automated Segmentation of Single-Molecule Localisation Microscopy Data

2.1 Introduction

To analyse the organisation of protein complexes in SMLM data, a useful first step is to segment the dataset. Segmentation of an image extracts the boundaries of an object, such as the nucleus or plasma membrane, and allows the analysis of that specific region [185]. Segmentation of an SMLM dataset labels each molecular localisation (a data point with *xyz* position, channel identifier, other properties) as belonging to a particular target structure (nucleus, etc.) or not. This enables the nanoscale analysis of protein organisation in those structures, vital for understanding sub-cellular structures and function.

Several methods have been developed for image segmentation, such as U-Net [186,187], Cellpose [188], reviewed in [185], and Ilastik [189]. U-Net is a widely used neural network developed for biomedical segmentation, with many pre-trained models available to non-experts through plug-and-play style interfaces. Cellpose provides U-Net style models for cell segmentation and is pre-trained on a very large and diverse set of cell images that have been manually annotated, avoiding the need for retraining [188]. Ilastik is a semi-automated machine learning software that provides a GUI (graphic user interface) with access to multiple workflows, including segmentation [189]. However, as image processing methods, these approaches cannot currently be used directly on an SMLM dataset (point cloud).

Methods for segmenting cells and subcellular regions directly from SMLM data are not as well established [190-192]. A range of software has been developed to segment clusters from *xy* or *xyz* localisations in SMLM data [193]. However, these approaches

are generally not designed for high throughput processing of many images, require careful parameter tuning for each FOV and require a subsequent processing step to combine the segmented clusters into whole cells. Alternatively, a widefield image of the FOV, automatically thresholded, can be overlaid onto an SMLM point cloud to segment and extract the localisations in this region [194]. However, thresholding can struggle to deal with images that are noisy or that have large variations in the intensity of the background or object, which in turn leads to poor segmentation. Nanowrap is a relatively new approach that extracts subcellular membrane surfaces, approximating the SMLM data using a coarse, density-based isosurface or density-thresholded mesh[195]. However, this method cannot be used to automatically annotate an entire dataset, as it requires manual oversight and adjustment to the hyperparameters to segment each FOV. Further, this method is not designed for data such as ours, which includes localisations that are not noise and are not located at the membrane, which should not be segmented.

To allow the nanoscale analysis of protein distributions in many new subcellular features and in cells, we have developed *locpix*, a pipeline for automatic segmentation of localisations in structures of interest. *locpix* automatically makes downstream analysis possible on specific structures in SMLM datasets over many fields of view. It has also been made available as a Python package (https://github.com/oubino/locpix). *locpix* also provides a means of manually annotating SMLM data rendered as an image. This process labels individual localisations in the SMLM point cloud, giving ground-truth data for both training (in machine learning methods) and testing of segmentation.

In this chapter, we apply *locpix* to generate a per-localisation cell segmentation from a SMLM dataset of EREG and EGFR localisations in cell sections (Figure 2.1). To do this, we first segment the plasma membrane for the cells at the pixel-level, by rendering the SMLM data as an image and testing several segmentation algorithms. This membrane segmentation is used to generate the cell segmentation at the pixel-level, separating each instance of a cell (instance segmentation). Pixels located at the boundaries between touching cells are not considered differently to other membrane pixels (see Section 2.2.7 for more details). This contrasts with previous approaches, which consider the boundary between neighbouring cells as background or as a separate class to avoid ambiguity, and which can also perform the instance segmentation in a single-step [196,197]. The data was represented as an image, as the membranes and cells were large enough that they could be resolved and segmented without requiring the full precision of the SMLM data (Section 1.2.2).

We then use the pixel-level membrane and cell segmentations to generate a per-localisation segmentation. This separates the localisations into groups belonging to different cells with an additional label that details whether it is situated at the cell membrane (or not). The performance of the membrane segmentation was quantitatively evaluated on the localisations and their labels, as the localisations were the subject of downstream analysis, so it was important to know their accuracy (Section 1.2.2). The performance of the cell segmentation was only qualitatively compared by visualising and comparing the cell segmentation at the image level, as the cells were not manually annotated. Finally, we analyse the protein localisations from the segmented datasets, comparing the organisation between membrane and non-membrane protein localisations over different cells.

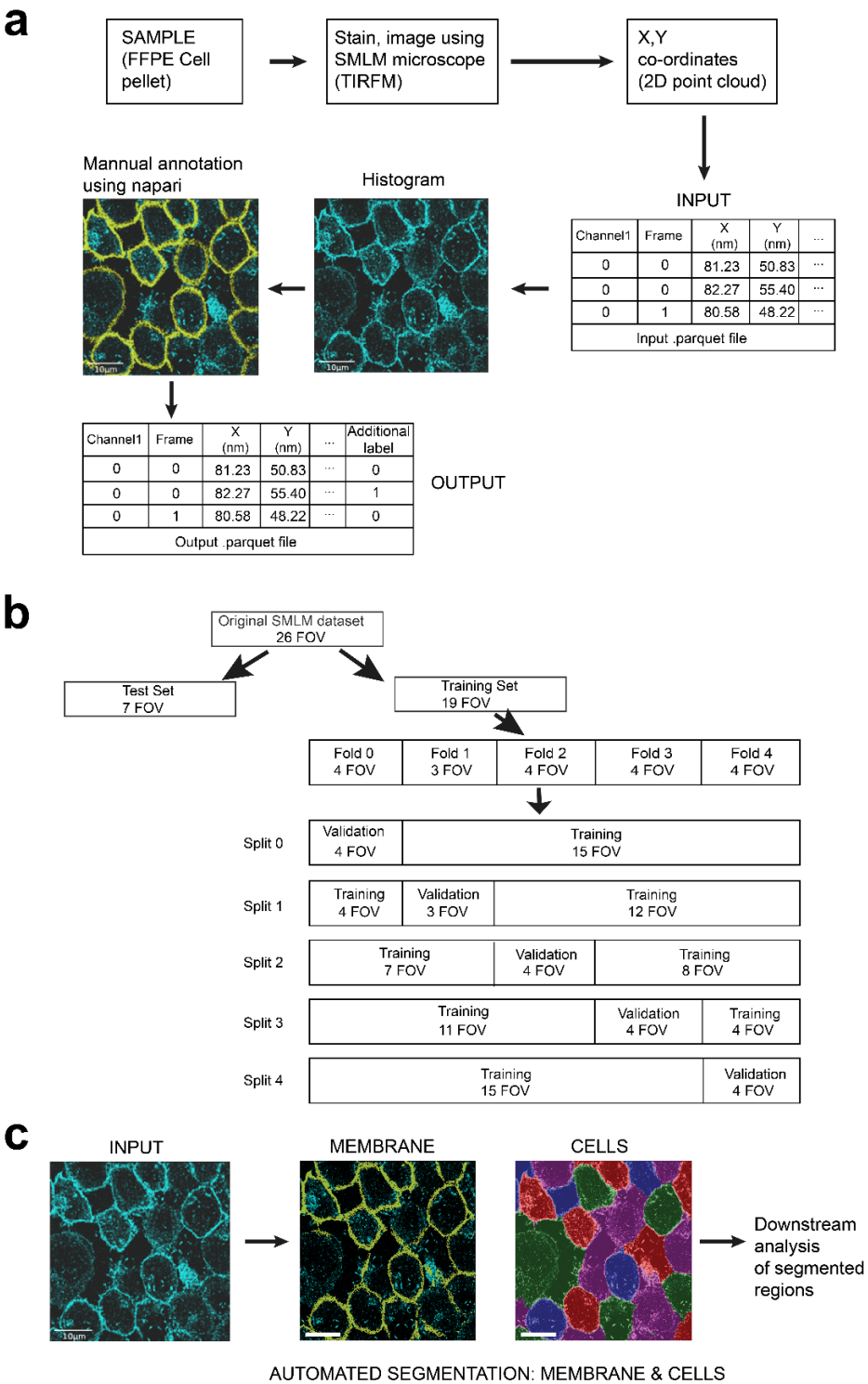

Figure 2.1. *locpix* analysis pipeline for segmentation of SMLM data. a The point cloud data is converted from tabular style data to an image (2D histogram). Images are manually annotated in *napari* (yellow lines are manual annotations of plasma membrane). The output tabular data has an additional ground-truth label for each localisation (zero for non-membrane and one for membrane in this case). **b** Partitions of the dataset for training and testing of a segmentation algorithm (26 FOVs in this case, using 5-fold cross-validation in training). **c** Automated segmentation obtained on data from FOVs previously unseen by the segmentation algorithm (membrane segmentation followed by cell segmentation here).

2.2 Materials and methods

2.2.1 Sample preparation and SMLM data acquisition

To test our pipeline, we used SMLM data obtained from imaging sections of cell pellets, taken from formalin-fixed and paraffin-embedded (FFPE) samples. The cell pellets were generated from an engineered single-cell-derived clone (C15) of a metastatic colorectal cancer cell line, SW620, which contained a targeted mutation in exon 22 of PTCH1 (Patch1) [198]. This mutation upregulated expression levels of EGFR and EREG in these cells (personal communication, 2022). The sections were labelled using antibodies to the plasma membrane protein EGFR (epidermal growth factor receptor) using an anti-EGFR antibody (5B7: rabbit monoclonal, Roche) and to its ligand EREG (epiregulin) using an anti-EREG antibody (SP326: rabbit monoclonal, Roche), followed by donkey anti-rabbit Alexa 647 and goat anti-rabbit CF568 secondary antibodies. A goat anti-rabbit fragment antigen-binding antibody was used for blocking before adding the second primary anti-EGFR antibody. Staining for these two proteins generated a high-density membrane localisation, often visible as cell outlines in sections through the cell pellet.

To image the samples, we performed TIRF (total internal reflection fluorescence) dSTORM (direct stochastic optical reconstruction microscopy) imaging using a commercial system (NanoImager, ONI) and a 100 × 1.4 NA oil-immersion objective lens with a 50 × 80 μm field of view. Samples were bathed in STORM buffer (B-cubed buffer, ONI, BCA0017). Using an exposure time of 30 ms, 5000 frames per channel were acquired sequentially. The 640 nm laser was set to 60% power, and the 561 nm laser to 20% power of the maximum excitation output of the NanoImager. 2D localisation of fluorescence emission events was performed while imaging using NimOS (ONI, UK), the inbuilt software for the NanoImager.

We obtained 26 FOVs from four samples with approximately 1.5×10^7 localisations per FOV. In 12 of the 26 FOVs, EREG/EGFR were imaged in the 568/647 nm channels, respectively. In the remaining 14 FOVs, EREG/EGFR were imaged in the 647/568 nm channels, respectively. This was to qualitatively confirm that there was no difference in staining and imaging results when changing the secondary antibody used to target each primary. The cells were prepared by Gianluca Canettieri and Natalia A. Riobo-Del Galdo. Hayley Slaney prepared the samples and acquired the SMLM data. I performed all subsequent preprocessing and analysis.

2.2.2 Data preprocessing

Drift correction, filtering, and temporal grouping for each FOV were performed using CODI (COllaborative DIscovery platform, ONI, UK; https://oni.bio/nanoimager/software/codi-software/). The filtering step removed localisations in the 647 nm channel in frames 5000-10,000 (while imaging at 568 nm) and in the 568 nm channel in frames 0-4999 (while imaging at 647 nm). In addition, localisations with >30,000 photons, with a standard deviation of the fitted point spread function (PSF) <75 nm or >200 nm, with a p-value for the fitted PSF above 0.01 or with a localisation precision >25 nm were removed as recommended by ONI and often performed to remove spurious localisations in SMLM data [199-203]. Starting from the default values from CODI, the values were adjusted after visualisation of the rendered data and the histogram for each parameter via the CODI interface. Localisations within 60 nm and no more than two frames apart were grouped, removing those that existed for longer than five frames. This resulted in ~ 250,000 localisations per FOV.

Each FOV was then reconstructed into one 2D histogram (image) per channel. The data in its proprietary format was first converted into an Apache Parquet file, a column-orientated data format which can be more efficient for querying and storing than .csv files (https://parquet.apache.org/). For each localisation, the channel, frame number and *xy* coordinates were stored. For each FOV, the point cloud data for the EGFR and EREG channels were binned into separate 2D histograms and rendered as images, with pixel grey levels equal to the bin values. Each histogram consisted of 500 × 500 pixels over the x and y range of the FOV. Since the range varied between FOVs, each pixel was between 99-100 nm wide and 157-160 nm tall. For the analysis presented here, we summed the pixel values from the two channels to obtain one overall 'membrane' dataset.

2.2.3 Manual annotation

The EGFR and EREG preprocessed localisation distributions (point clouds) were binned into 2D histograms, rendered as images, and loaded into separate channels in *napari,* an existing open-source image viewer implemented in Python [204]. In *napari*, the cell membranes were manually traced using the freehand drawing tool to generate a ground-truth labelled image, in which each pixel had an integer value of either zero for non-membrane or one for membrane. The ground-truth label for each pixel was then assigned to all localisations within the corresponding 2D histogram bin. The localisations were then exported into a new Apache Parquet file, with an additional column for the ground-truth label (Figure 2.1a). This manual annotation step is available as an open-source napari plugin at https://www.napari-hub.org/plugins/napari-locpix.

2.2.4 Dataset partitions

Separate datasets for training and evaluation were created (Figure 2.1b). First, the entire dataset was divided into a training set (70% of FOVs) and a test set (30% of FOVs). The test set was generated from the FOVs with the highest percentage of membrane localisations according to the manual annotations, to ensure the models were tested on FOVs with multiple clearly visible cells. The test set was not used until performance analysis. The training set was then divided into five subsets (hereafter referred to as folds). Five different splits of the training dataset were then generated, each with a different fold for validation and the remaining folds for training (Figure 2.1b). For each of the five methods used for membrane and cell segmentation detailed below, we developed a model for each split of the training dataset, using the training folds for training the model where relevant (standard U-Net, Cellpose (retrained) and Ilastik) and the validation folds for comparing performance. Each model was then evaluated on the test set for the final comparison.

2.2.5 Segmentation algorithms: probability map generation

We developed several methods to predict the probability for each pixel in a FOV that it was located within the plasma membrane (probability map). These probabilities were then assigned to the localisations belonging to each pixel. First, for each method except Ilastik, the EGFR and EREG images (2D localisation histograms) were summed into a single channel for processing. For Ilastik, the input was a two-channel EGFR and EREG image. The pixel values in the resulting images (and each channel in the Ilastik method) were scaled by $\log_2$ to reduce skew, thresholded above zero and scaled to between 0 and 255. We obtained membrane probability maps or pseudo probability maps from these images using the following approaches.

- Otsu thresholding: A binary segmentation mask is obtained after Otsu thresholding of the transformed and scaled image. This assigned a value of 0 or 1 to each pixel and its underlying localisations, which was treated as the probability (or pseudo probability) for simplicity and consistency between methods.

- U-Net: Standard U-Net architecture, with four encoder blocks and decoder blocks with skip connections between them, and a final sigmoid function to convert the raw output values to normalised probabilities for image pixels (Figure 2.2). Images were normalised by subtracting the mean and dividing by the standard deviation of all pixel values in the images in the training folds. Further, random augmentation including rotations, horizontal and vertical flips, erasing, and perspective shifting, was applied to the training folds. The model was trained for 1000 epochs using a dice loss function and Adam optimiser with a learning rate of 0.01 and weight decay of 0.0001 [205]. The model was saved when the loss on the validation fold was lowest. This model was implemented in PyTorch using code adapted from https://github.com/milesial/Pytorch-UNet.

- Cellpose (pre-trained): The Cellpose architecture is a modification of the standard U-Net (Figure 2.2), see Appendix (Section 7.1.2) for further details. We used the 'LC1' model in Cellpose, pre-trained on phase-contrast images of cells with only a single channel for cytoplasm. The LC1 model was considered the most appropriate, given that we expect the edge of the cytoplasm and EGFR/EREG proteins to define a similar boundary for the cell. Brief experimentation on a training image also indicated this was the

best-performing Cellpose model. Cell diameter in LC1 was manually set to 100 pixels as determined from training FOVs. The raw prediction generated by Cellpose (pre-trained) for each pixel indicated the likelihood that it belonged to a cell. This was scaled to between 0 and 1, as usually performed by Cellpose, and reassigned as the probability the pixel belonged to a membrane. Cellpose version 2.0 was used [188,206].

- Cellpose (retrained): We modified the Cellpose training script to change the loss function to calculate binary cross-entropy logits loss between the ground-truth label image from manual annotation and the output membrane probability map, manually set the cell diameter and mean cell diameter for all images to 100 pixels and allow for *k*-fold cross-validation. The pre-trained Cellpose LC1 model was retrained for 1000 epochs, with a weight decay of 0.0001 and a learning rate of 0.01. We performed limited tuning for these hyperparameters by training with a small partition of split zero's training folds (Figure 2.1b), without evaluation on the test dataset. The output probabilities were scaled as in Cellpose (pre-trained). The model was saved when the loss on the validation fold was lowest. Cellpose version 2.0 was used [188,206].

- Ilastik: The two-channel images and ground-truth membrane annotation images were used to perform the Ilastik pixel classification workflow available through the Ilastik GUI. We trained the model using the same training data with all possible image features available in Ilastik. No further annotations were made to the ground-truth images once they were loaded into the GUI. We chose a label of 2, rather than 0, for non-membrane pixels (in Ilastik, a label of zero means no label is present) and randomly removed ~80% of

these non-membrane ground-truth pixel labels to reduce the computational overhead.

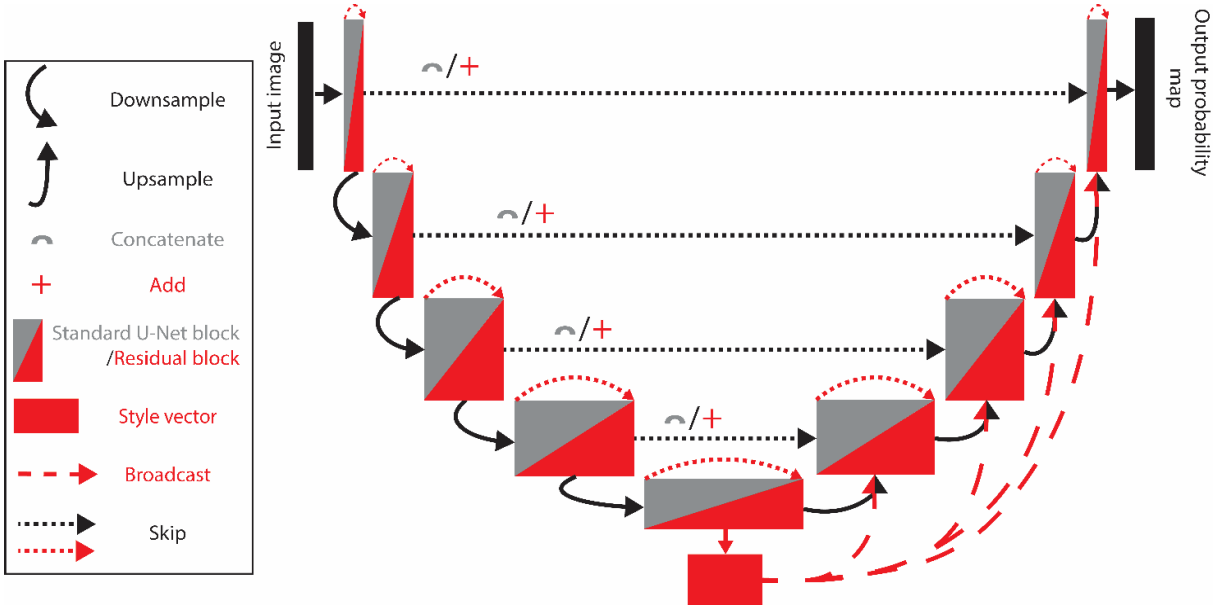

Figure 2.2. Standard U-Net (grey) and Cellpose (red) architectures [196,206]. Operations in black are common to both architectures.

2.2.6 Membrane segmentation

The point cloud datasets were then segmented by classifying each localisation as membrane (positive class) or non-membrane (negative class). Performance was evaluated at this per-localisation level, as this is most relevant to downstream analysis of nanoscale protein organisation. More information on the metrics used to evaluate performance can be found in the Appendix (Section 7.1.1).

Following the membrane probability map generation with each method, the probabilities assigned to the localisations underlying each pixel were used together with the ground-truth annotation of those pixels (membrane or not membrane) to plot precision-recall (PR) curves. We prefer this approach to the commonly used ROC (receiver operating characteristic) curve based on its sensitivity to changes in false positives (FP), despite a large number of true negatives (TN) [207,208]. First, for each

split of the training data, the localisations from all FOVs in the training folds were aggregated into one table. In this table, every localisation now had both its ground-truth label from manual annotation (0: non-membrane or 1: membrane) and its model-generated probability of belonging to a membrane. PR curves were generated by calculating the precision and recall for different values of a classification threshold, τ, that increased from zero to one, where localisations with a probability above each τ were assigned to the membrane. PR curves for the validation folds and the test set were generated using the same method. The normalised area under the PR curve (AUCNPR: area under curve normalised PR) was then calculated for the test set [209]. For the Cellpose (pre-trained) and Otsu models, which were not trained on our data, the probability maps on the test set did not depend on our training dataset splits, resulting in only one PR curve and AUCNPR in each case.

The F_1 score was used to evaluate the performance for the final segmentation as it accounts for class imbalance, as is present in our dataset (more non-membrane than membrane localisations) [208,210]. For each segmentation method, the probability threshold τ that maximised the F_1 score for localisation classification was determined for the training fold for each split. These values of τ were then applied to the probability maps for the test dataset, used to calculate the accuracy, precision, recall and the F_1 score for each model on previously unseen data. For the Otsu method, the membrane probability map was already binary, and setting different probability classification thresholds, τ, between 0 and 1 did not affect results. Therefore, there was a single result on the test set for each metric, independent of the training dataset splits.

2.2.7 Cell segmentation

For each membrane segmentation method apart from Ilastik, the cells were segmented using the watershed algorithm, using manually identified seed locations for the cell centres. Seeds were also placed across the background to prevent cell segmentations from extending into the background and to avoid assigning the same label to well-separated membrane localisations on different cells. For standard U-Net and both Cellpose methods, the watershed algorithm was applied to the probability maps from membrane segmentation. For the Otsu method, the EGFR and EREG 2D localisation histograms were summed, transformed by $\log_2$, thresholded above zero and scaled to between zero and 255 as performed for membrane segmentation. The result was used as input to the watershed algorithm. The decision to use this thresholded and scaled histogram rather than the raw histogram was based on a visual analysis of the performance on one training histogram. For these methods the watershed algorithm was implemented via scikit-image, an open-source Python package for image processing [211].

For Ilastik, the cells were segmented using the Ilastik multicut workflow available through the Ilastik GUI [189]. This uses the watershed algorithm to generate superpixels, using automatically calculated seed locations from the local maxima of the distance transform applied to the probability map for the membrane segmentation. A random forest is then trained to predict whether neighbouring superpixels should be merged, by training an edge classifier applied to a graph representation of the superpixel map. We trained the model using only five histograms from the training folds to save time. Ilastik then batch-processed the remaining histograms from the entire dataset using these trained parameters.

As each method uses the watershed algorithm, pixels located at the shared boundary between touching cells will be roughly divided around the middle of their shared boundary.

2.2.8 Downstream analysis

We performed an example of downstream analysis of segmented membranes as follows [212]. First, we manually picked well-segmented cells from the test set based on a visual inspection of the cell segmentation and by picking cells that were not situated at the edge of the FOV (only partially visible). Next, comparing localisations at the plasma membrane and interior of each cell, we calculated the 2D radial distribution function and clustered the localisations using density-based spatial clustering of applications with noise (DBSCAN). The 2D radial distribution function,

$$g(r) = \lim_{dr \to 0} \frac{p(r)}{2\pi \left(\frac{N^2}{A}\right) r dr}$$

measures the increased chance compared to a random distribution of finding a localisation at distance *r* from another localisation, where *p(r)* is the number of localisations found between distance *r* and *r* + *dr* apart, *N* is the number of localisations per cell and *A* is the cell area [110,213]. DBSCAN identifies clusters of localisations by identifying the regions of higher density and is commonly used for clustering SMLM data [190,193]. DBSCAN has two parameters, epsilon and minimum points, which define the radius of each localisation's neighbourhood and the minimum number of neighbouring localisations to assign a localisation to a cluster. Here, epsilon and minimum points were set to 75 nm and 5, respectively. Then, aggregating over the cells for the test dataset, we calculated the overall 2D radial

distribution function, localisations per cluster, and cluster length (using the convex hull) for the cell interiors and plasma membranes.

2.3 Results

2.3.1 Manual annotation

All 26 FOVs from the original SMLM dataset were manually annotated. Each localisation that belonged to a membrane was first manually annotated using the custom image annotation script (Figures 2.1a and 2.3a-d). It was not always possible to clearly differentiate between the membrane and the cell interior or general background (defined here as non-membrane). The manual membrane annotations thus did not always delineate the entirety of a cell (Figure 2.3a-d). This contributed to a small imbalance in the dataset, with ~1.5 times more non-membrane than membrane localisations for the dataset.

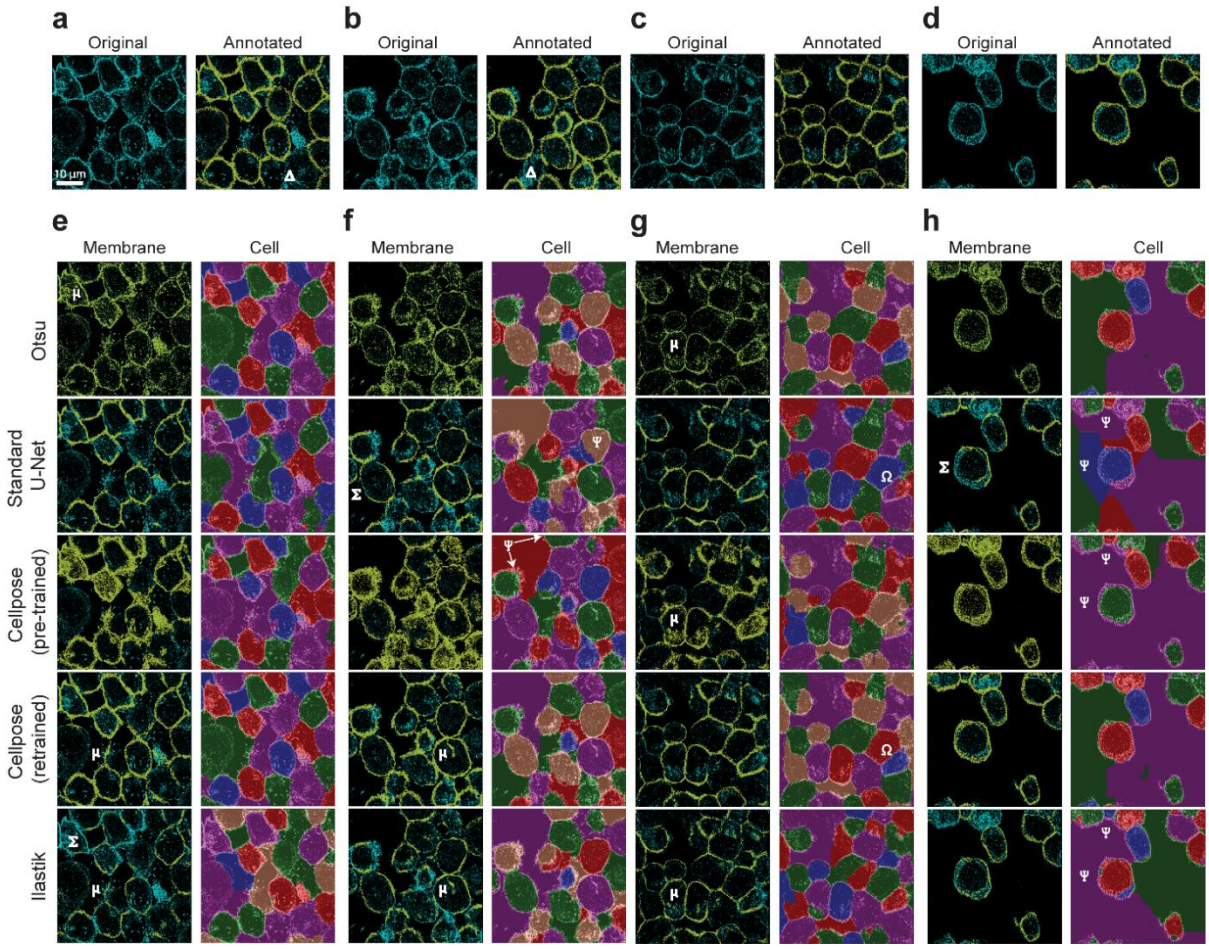

Figure 2.3. Manual annotation, membrane segmentation, and cell segmentation results. a-d The original 2D histogram (sum of EGFR and EREG) together with the annotated 2D dataset (yellow: membrane annotation) for four FOVs from the test set (unseen to all trained methods), where all results are from the same split (zero). **e-h** The membrane and cell segmentations for each of the five methods. For cell segmentation, each colour represents a different cell label, and non-bordering segments of the same colour represent different cell labels. KEY: **Δ**: non-annotated region that could be membrane **μ**: non-membrane localisations predicted as membrane **Σ**: membrane localisations predicted as non-membrane **Ψ**: error in cell segmentation **Ω**: instability of the watershed algorithm.

2.3.2 Membrane segmentation

Multiple methods for membrane segmentation were developed, trained, and validated on five splits of the training dataset. These included methods that learnt from our training dataset (Standard U-Net, Cellpose (retrained), Ilastik): one that had been pre-trained on a different dataset (Cellpose (pre-trained)) and one without any machine learning (Otsu method). Results on the training and validation folds are

given (Figure 2.4a,b), but the performance of each method was compared by considering the quantitative and qualitative performance on the test set (Figure 2.4c, Table 2.1 and Figure 2.3). We used AUCNPR (area under the curve, normalised precision-recall) as the key measure of performance, as it balances precision and recall and accounts for class imbalance, while the F_1 score was most useful when evaluating the performance on the final segmentation [150,208,210]. We were mostly confident that the annotated membranes were true membrane regions, but we cannot rule out that some may have been missed; therefore, true positive and false negative counts were the most reliable. This made recall the most reliable metric (Table 2.1), despite the pitfall of predicting all localisations as membrane, which gives the maximum recall (1.0).

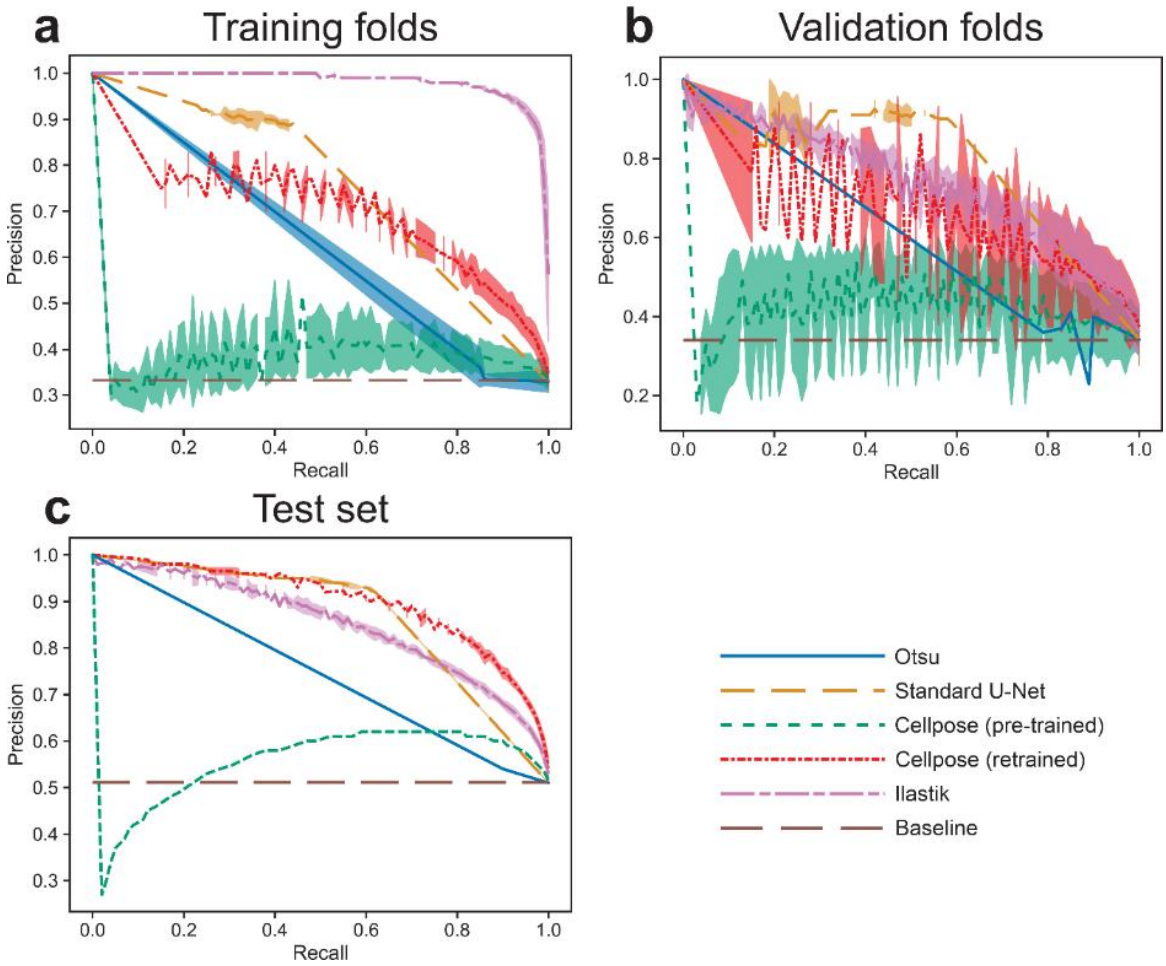

Figure 2.4. Precision-recall curves for the training folds, validation folds, and test set. For each method, the curve is the average over the five models (one model per dataset split) with ± one standard deviation shaded. **a,b,c** The baseline model predicts all localisations as belonging to a membrane. **c** Otsu and Cellpose (pre-trained) have zero variance, as the algorithm applied to the test data does not depend upon the training data (Section 2.2).

Table 2.1. Performance metric scores for each method. Scores are presented as the mean ± standard deviation over the five splits evaluated on the test set. Recall is given by $\frac{TP}{TP+FN}$, precision (prec.) by $\frac{TP}{TP+FP}$, F_1 score by $\frac{2\times precision\times recall}{precision+recall}$ and accuracy (acc.) by $\frac{TP+TN}{TP+TN+FP+FN}$, , where TP, TN, FP and FN are the number of true positive, true negative, false positive and false negative predictions respectively, and either non-membrane (non-memb.) or membrane (memb.) is the positive class. AUCNPR is the normalised area under a curve that plots precision against recall for different thresholds applied to the probability map. For Otsu and Cellpose (pre-trained) there is no variance for AUCNPR as the split has no impact on the probability map for the test set (Section 2.2). For Otsu, the remaining metrics have no variance as changing the threshold for each split has no impact on the probability maps (Section 2.2). The best scores for each metric are highlighted in **bold**.

	Recall non-memb.	**Recall memb.**	**Precision non-memb.**	**Precision memb.**	**F_1 score memb.**	**Accuracy memb.**	**AUCNPR memb.**
Otsu	0.205	0.902	0.666	0.543	0.678	0.561	0.63
Standard U-Net	**0.948 ± 0.001**	0.608 ± 0.006	0.698 ± 0.003	**0.924 ± 0.001**	0.734 ± 0.004	0.774 ± 0.003	0.810 ± 0.003
Cellpose (pre-trained)	0.273 ± 0.036	**0.943 ± 0.013**	0.824 ± 0.019	0.576 ± 0.008	0.715 ± 0.003	0.616 ± 0.011	0.36
Cellpose (retrained)	0.783 ± 0.023	0.842 ± 0.009	**0.826 ± 0.005**	0.803 ± 0.015	**0.822 ± 0.004**	**0.813 ± 0.007**	**0.853 ± 0.008**
Ilastik	0.911 ± 0.012	0.539 ± 0.033	0.654 ± 0.014	0.864 ± 0.010	0.663 ± 0.023	0.721 ± 0.012	0.784 ± 0.011

We found that Otsu was the second worst-performing model for membrane segmentation (AUCNPR: 0.63, Table 2.1). It overestimated the membrane localisations, as shown by the high membrane recall (0.902, Table 2.1) and low membrane precision (0.543, Table 2.1). Furthermore, the false positives were in areas unlikely to be regions of membrane that were mislabelled during annotation, such as cell interiors (Figure 2.3e and g, μ).

Standard U-Net significantly outperformed Otsu and was the second-best of all approaches (AUCNPR: 0.810 ± 0.003, Table 2.1). It had higher non-membrane recall and membrane precision but lower membrane recall than Otsu (Table 2.1), demonstrating that it predicted more membrane localisations as non-membrane but

made fewer false-positive membrane predictions. In particular, it predicted fewer localisations in cell interiors than membranes compared to Otsu (Figure 2.3e and g, μ).

Cellpose (pre-trained) performed the worst of all (AUCNPR: 0.36, Table 2.1). Despite outperforming Otsu in all metrics apart from AUCNPR (Table 2.1), the predictions appeared visually similar (Figure 2.3e-h). Further, like Otsu and unlike standard U-Net, it mistook cell interiors as membranes (Figure 2.3g, μ). It did have higher membrane recall than standard U-Net (Table 2.1), but it also made more false-positive mistakes (lower membrane precision, Table 2.1), predicting more of the non-membrane localisations as belonging to the membrane.

Cellpose (retrained) was the best-performing model (AUCNPR: 0.853± 0.008, Table 2.1). Retraining Cellpose demonstrated a clear performance improvement (rise in the PR curve vs. pre-trained, Figure 2.4c). This model predicted fewer localisations as belonging to a membrane (decrease in the membrane recall, Table 2.1), noticeably predicting fewer cell interiors as membranes compared to Otsu, Ilastik and Cellpose (pre-trained) (Figure 2.3g, μ). Compared to standard U-Net, Cellpose (retrained) gave more extensive segmentations (higher membrane recall, Table 2.1) but at the cost of more false positives (lower membrane precision, Table 2.1). At points standard U-Net seemed to better reflect our annotations, omitting an edge that Cellpose (retrained) predicted (Figure 2.3e, μ), which, despite looking membranous, was not manually annotated. Further, standard U-Net correctly segmented localisations in cell interiors that were mislabelled as membrane by Cellpose (retrained) (Figure 2.3f, μ). However, standard U-Net also omitted regions that were clearly membrane, which Cellpose (retrained) correctly segmented (Figure 2.3f and

h, Σ). Cellpose (retrained) also identified regions that may have been membrane, which we were not confident enough to annotate (Figure 2.3a and b, Δ).

Finally, we found that Ilastik performed worse than retrained Cellpose and standard U-Net (AUCNPR: 0.784 ± 0.011, Table 2.1). Like standard U-Net, it predicted more membrane as non-membrane than Cellpose (retrained) (higher non-membrane recall and lower membrane recall, Table 2.1). This included missing a significant proportion of the membranes (Figure 2.3e, Σ), which Cellpose (retrained) correctly segmented. Further, it made similar mistakes to Cellpose (retrained) with interiors (Figure 2.3f and g, μ) and trailing edges (Figure 2.3e, μ), the latter of which were not manually annotated despite looking membranous. Ilastik overfitted the training data, as evidenced by the poorer performance for the validation folds and test set compared to the training folds (Figure 2.4). One likely reason for this is the small size of the dataset. A second is that we did not monitor for overfitting during training by evaluating the performance on the validation folds. Therefore, the probability threshold determined using the F_1 score on the training set was also likely to be suboptimal when applied to the test set.

2.3.3 Cell segmentation

Multiple methods for cell segmentation were developed, trained, and validated on five splits of the training dataset and the results were compared using a qualitative analysis on the test set (Figure 2.3). Quantitative performance metrics on the localisations could not be evaluated as the cells were not manually annotated.

All methods segmented some cells correctly, but generally performed poorly. This was most evident in specific examples (Figure 2.3h), where we were most confident in the ground truth, reflected in the extensive manual annotations (Figure 2.3d: ~7

identified cells). Cellpose (retrained) did not make the same mistakes that Cellpose (pre-trained), standard U-Net and Ilastik made for cell segmentation (Figure 2.3h, Ψ). This was expected as these methods relied on the quality of the membrane segmentation, which was best for Cellpose (retrained). Further, as Cellpose (retrained) provided more extensive annotations than the other high-performing model, standard U-Net, it was less likely to divide cells in two because there were gaps in the membrane annotation (Figure 2.3f, standard U-Net, Ψ). Despite this, the performance of Cellpose (retrained) was almost identical to the much simpler Otsu method (Figure 2.3e-h).

There were problems using the watershed algorithm for cell segmentation across all methods. Firstly, localisations from the exterior of different cell membranes were incorrectly assigned the same label, despite being far apart (Figure 2.3f, Cellpose (pre-trained), Ψ). Even though the same markers were used for all methods (apart from Ilastik), small differences in the membrane segmentation caused large differences in the cell segmentation (Figure 2.3g, Ω).

2.3.4 Downstream analysis

Once the data is segmented, it can then be used in downstream analysis, as we show here for membrane and cell segmentation results, by exploring EGFR distribution and clustering in the cell membrane and interior (Figure 2.5). From the automated segmentation from Cellpose (retrained), we manually selected well-segmented cells from all FOVs in the test dataset (60 cells) and separately calculated the 2D radial distribution function and clustering of their localisations (DBSCAN: epsilon = 75 nm, minimum points = 5) predicted as ‘membrane’ or ‘non-membrane’. Localisations within the interior of the cells were characteristically

found in close proximity (≤ approx. 100 nm), while typical distances between those at the membrane extended over a longer distribution (Figure 2.5b). Localisations at the cell membrane formed clusters with significantly higher localisations per cluster and cluster length than the cell interior (Figure 2.5d). These clusters included repeated localisations of the same fluorescent dye molecule, multiple dye molecules per secondary labelling antibody and any clustered instances of EGFR. Although many membrane localisations form large clusters in space, there is a smaller number of these large clusters, and the difference between the distributions appears to reveal a more subtle difference between the arrangement of EGFR in the interior and at the membrane. The cluster parameter distributions in the interior likely include a major contribution from monomeric EGFR, with multiple dye molecules per labelling antibody. The increase in the median length and number of localisations per cluster at the membrane may be a result of the known dimerisation of EGFR at the membrane, or a larger number close together, although results are confounded by the dense packing at the membrane and the multiple fluorescent molecules per labelling antibody.

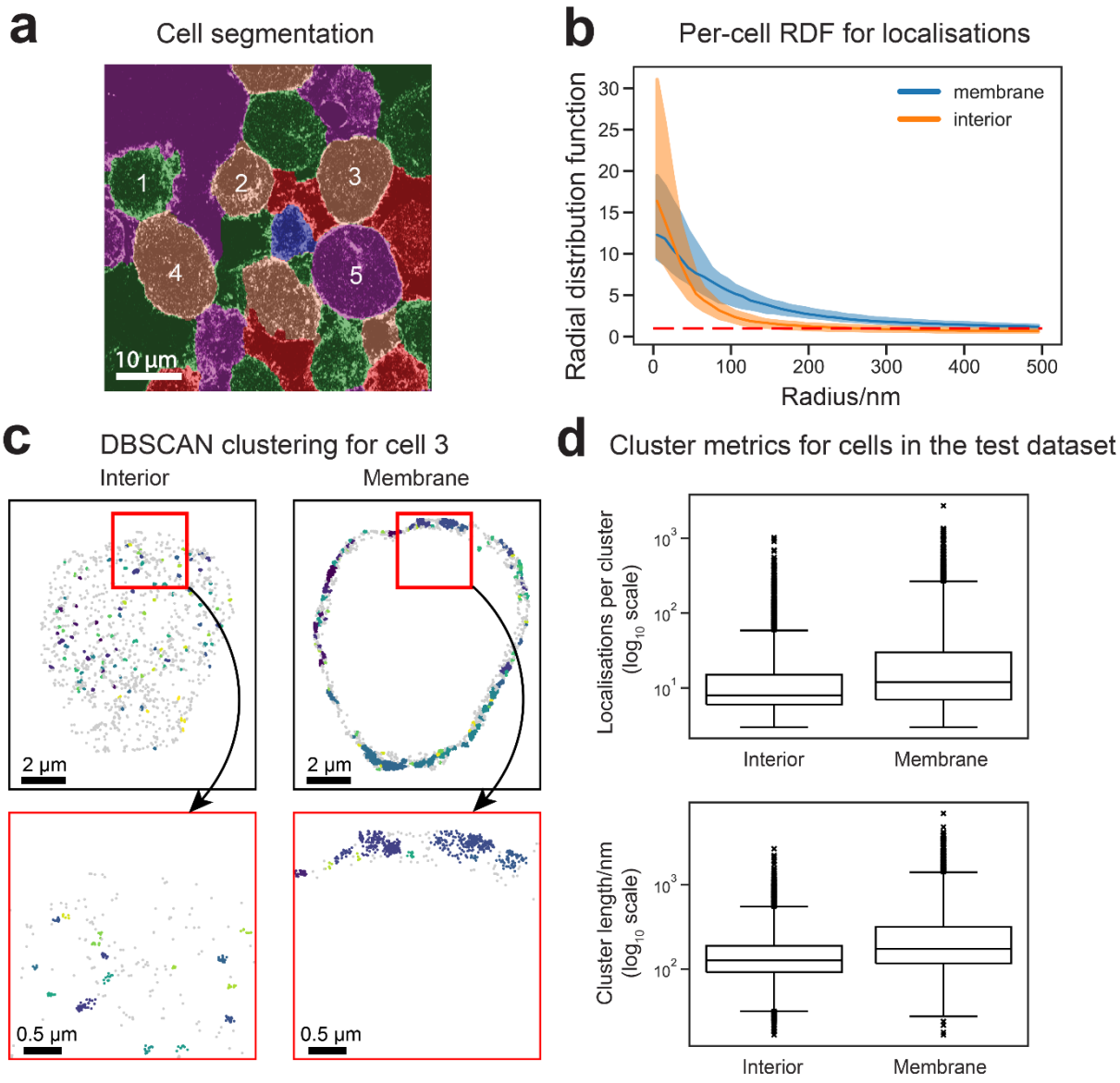

Figure 2.5. Analysis of segmented data. a Part of a FOV from the test dataset that shows five of the eight cells that were manually selected following cell segmentation. **b** Radial distribution function (RDF) for the membrane and cluster localisations calculated for each cell and aggregated over the whole test set. The median value for each radius is plotted with the interquartile range shaded. **c** Interior localisations and plasma membrane localisations for cell 3 from panel A. Each colour represents a different cluster from DBSCAN (epsilon = 75 nm, minimum points = 5). Grey localisations do not belong to a cluster. Localisations within the red box are shown at higher magnification below as indicated. **d** Cluster metrics for the cells in the test dataset. Box plots for the number of localisations per cluster (top) and cluster length (bottom) for each cell aggregated over the whole test dataset. The three central horizontal lines are the first (Q1), second (Q2) and third quartile (Q3) from bottom to top, the whiskers are Q1 - 1.5 × (Q3 - Q1) and Q3 - 1.5 × (Q3 - Q1), and the outliers are plotted as crosses. Median localisations per cluster: 8 (interior), 12 (membrane), Two-tailed Mann-Whitney $U = 1.9 \times 10^7$, $n_{membrane}$=6183, $n_{interior}$ = 8141, $p \leq 0.05$. Median cluster lengths: 127 nm (interior), 174 nm (membrane), Two-tailed Mann-Whitney $U = 1.8 \times 10^7$, $n_{membrane}$= 6183, $n_{interior}$ = 8141, $p \leq 0.05$.

2.4 Discussion

This chapter has demonstrated a pipeline for annotating and automatically segmenting cells and membranes from SMLM point-cloud data. This pipeline converts the localisations to images, to allow segmentation via image-based tools, and then converts back to the point-cloud representation, to allow performance evaluation and downstream subcellular analysis at the localisation level. Using the normalised area under the precision-recall curve (AUCNPR) to compare methods, we found that Cellpose (retrained) performed the best for membrane segmentation. This type of approach is useful for identifying molecular localisations as belonging to specific cells and regions of cells, to enable directed analysis of the high-precision localisation data specific to those regions, for instance, using one of the many pre-existing algorithms [105,191,214].

The trained models outperformed traditional methods for membrane segmentation (Otsu method) [215]. This was expected, as the heterogeneity in membrane staining between cells makes it challenging to set a threshold for the entire FOV. Further, standard thresholding techniques fail to consider both local (cells) and global (entire FOV) context and can struggle to deal with small objects and images that are noisy or show significant variation in the background or object intensity [216,217].

The inability of these trained models to outperform the Otsu method in cell segmentation points to issues with applying the watershed algorithm to this task. Small changes in the membrane segmentation led to large differences in cell segmentation, and localisations from different cell membranes that were large distances apart could be assigned the same label. This is expected based on the

known disadvantages of watershed, that it is sensitive to noise and inhomogeneity of background and object intensity, and it does not consider the global context [218,219].

When comparing membrane segmentation models, one should consider whether it is more important for downstream analysis to minimise the number of missing membranes (false negatives) or the number of non-membrane localisations predicted as belonging to a membrane (false positives). Cellpose (retrained) had higher membrane recall but lower membrane precision than standard U-Net. If it is more important to avoid false positives while allowing more false negatives, then standard U-Net would be better, and vice versa for Cellpose (retrained). Given the small size of our dataset, Cellpose (retrained) was preferred in our work to retain as much information as possible, while allowing for some false positives. However, future users of this pipeline with larger datasets may prefer that only the highest quality data is retained. The accuracy of these metrics should also be considered; for example, some of the false positives from Cellpose (retrained) were in regions likely to contain membrane that we were not confident enough to annotate.

More generally, the downstream task should inform the minimum acceptable score for each metric [220]. For example, while this was hard to estimate for the membrane segmentation in this study besides comparing with the performance of a random classifier (e.g. 0.5 for AUROC), the performance was good enough to allow the organisation of membrane and cell proteins to be distinguished. In Chapter 4, when using *locpix* to segment cells from patients, this issue was avoided by using manual annotation. This ensured that the cell segmentation was not the limiting factor for downstream classification. However, observing how segmentation performance affects downstream classification would be an interesting subject of future study.

Future work should also explore different methods to automatically perform quality control of the cell segmentation, to identify high-quality data for downstream analysis, while maintaining high throughput. This could be achieved by calculating features of the cells such as area or intensity and removing cells at the extremes or above/below user-determined cutoff values [221]. Alternatively, we could use prior knowledge to remove cells that deviate from our expectations, for example, removing cells that deviate from the expected or known pattern or level of protein expression [222]. Care should be taken when developing these approaches, to mitigate the risk of biasing the analysis and removing rare or unexpected results [221].

Using an ensemble of metrics can give a fairer indication of model performance across both positive and negative classes, rather than focusing on a single metric. For example, membrane recall could be misleading in isolation, because it can be maximised by classifying all localisations as membrane and giving no true negatives. Combining PR curves and AUCNPR with *k*-fold cross-validation measures how robust the methods are to changes in threshold and changes to training and evaluation data, respectively. The performance of Cellpose (retrained) was less variable across a range of thresholds, evidenced by the highest AUCNPR (Table 2.1), which is important if setting the threshold is challenging.

The information gained by using SMLM instead of conventional imaging may be lost when rendering the SMLM data as an image, which bins the high-precision localisations into pixels (Section 1.2.2) [100,103]. For many segmentation tasks, rendering the data into a sufficiently large image will prevent this issue. This was demonstrated in this study, where the membranes could still be accurately identified. Nonetheless, deep-learning algorithms designed to work directly on point-clouds

may avoid this issue entirely. For example, since publishing the work in this chapter, a graph-based neural network has been proposed for segmentation of SMLM data [173]. However, this was limited to identifying noisy localisations in simple subcellular structures: microtubules and vesicles.

In this work, the seed locations were manually selected for the watershed algorithm to ensure a high-quality dataset for downstream analysis. However, future work could explore different ways to automate this, to increase throughput and improve usability for non-experts. For example, Ilastik generates seed locations for watershed by identifying the local maxima of the distance transform of the membrane segmentation [189]. However, this still requires further manual processing using the graph-cut algorithm to merge neighbouring regions (groups of pixels) that belong to the same cell, which often results from the watershed algorithm. Alternatively, future work could trial methods that do not require the watershed algorithm. For example, the full Cellpose pipeline uses a U-Net to generate a gradient map, which tracks each pixel back to its cell centre [206]. Pixels that map to the same location are then grouped into cells. This could not be used as-is on our dataset, as the cells had a low intensity and consequently a low probability of belonging to a cell within the cell interior. Therefore, further modifying the Cellpose architecture for this dataset could be the subject of future work.

Future work could also explore methods for cell segmentation that do not require generation of an intermediate membrane segmentation. This includes methods specifically designed for cell segmentation, such as the full Cellpose pipeline, and general purpose methods fine-tuned for cell segmentation, such as Mask R-CNN [206] [223,224]. Mask R-CNN uses a CNN to generate bounding boxes for each instance in

the image (e.g. each separate cell), classify each instance (e.g. by cell type) and then segment each instance within each separate bounding box [224]. The backbone of this network can also be replaced by a transformer-based network to improve performance, as successfully demonstrated for cell segmentation [223,225]. The transformer architecture has also been incorporated into CellSAM, a foundation model specifically designed for cell segmentation [226].

Looking forward, *locpix* could be applied to SMLM data from a broad variety of samples. In this study, sections from FFPE samples were imaged using a commercial microscope, mimicking a recently established protocol for acquiring SMLM data from patient tissue in the clinic [84]. This suggests that with future improvements to fully automate the pipeline and improve its accuracy, *locpix* could help segment SMLM data from clinical samples, allowing for the analysis of high-precision molecular distributions in specific subcellular regions. This, in turn, could help bring about the use of SMLM as a new tool for assisting in the analysis of patient samples in the clinic.

3 *ClusterNet*: Classifying Single-Molecule Localisation Microscopy Datasets with Graph-Based Deep Learning of Supra-Cluster Structure

3.1 Introduction

Sample classification is an important step in the analysis of SMLM data, allowing for automated recognition of sample type and downstream aggregation and analysis of data from many samples of the same type. Using deep learning (DL) algorithms for this task may also facilitate biological discovery, despite only having sample-level labels (weakly supervised) [176]. While DL algorithms have classified SMLM data of complex structures such as whole cells, they have first rendered the data as a pixelated image [136,171]. While this may be appropriate for the task, using the right choice of image and pixel size, it risks sacrificing the full potential of the precision gain of SMLM over conventional imaging, and hence, the information available (Section 1.2.2). Further, this doesn't scale well for classification tasks requiring high-precision data, or when moving from 2D to 3D (Section 1.2.2).

Existing pipelines for classifying SMLM point cloud datasets do not extend to the classification of structures larger than single clusters of localisations. Current methods, such as ASAP, SEMORE, ECLIPSE and SuperResNet (Section 1.3.2), focus on classifying individual localisations and clusters, using either pre-determined (handcrafted) features of clusters, such as cluster area and length [150,163-167,169,170], or more abstract cluster features learnt automatically by a point- or graph-based DL network [103,151,172]. However, the arrangement of multiple clusters in a sample (the supra-cluster structure), as well as the combination of the features of the different clusters, is likely to hold important biological information.

Extending algorithms using handcrafted cluster features to complex localisation patterns, such as multiple clusters, is not straightforward, requiring new calculations for features that would discriminate between them. Particle averaging has been used to classify more complex patterns than handcrafted features have typically described, but is still restricted to the classification of single particles with highly consistent structure [227,228]. Further, cells have been classified using handcrafted features calculated from graph representations of SMLM point cloud data [166]. However, this only considered the distances between localisations and not the spatial location of clusters of localisations or the shape of the clusters and the cell (except for the graph diameter and radius). Moreover, differences in protein organisation in different regions of the cell may be lost by this method, as only features of the whole cell or features averaged over each localisation were calculated. Point- and graph-based DL pipelines could better capture this information, but so far, have also been limited to classifying localisations and clusters, and ignore the sample-level supra-cluster structure. New methods are therefore required for the classification of variable and complex structures in SMLM data, including whole FOVs.

This chapter presents a deep learning classification pipeline for SMLM point cloud datasets that incorporates supra-cluster structure (Figure 3.1). We have developed *ClusterNet*, a graph-based DL network, to classify graphs constructed from clusters in the localisation data. The clusters in each graph are represented by discriminative features extracted from their constituent localisations, thereby retaining the original precision of the localisations. Each cluster also has coordinates, thereby encoding the spatial arrangement of the clusters relative to each other.

We demonstrate this new pipeline on a model, open-source SMLM dataset of DNA-PAINT localisations from DNA origami nanostructures designed to resemble a selection of digits and letters [229]. We present implementations of *ClusterNet* using both handcrafted features of individual clusters and features learnt with a neural network, achieving balanced classification accuracy of 99% across the dataset, improving on previous workflows [171,228,229]. Further, we include and demonstrate the use of DL explainability algorithms to interpret the output of *ClusterNet*, for a data-driven exploration of the results [136]. Localisations in SMLM FOVs of any spatial extent and complexity may be clustered and used as input to *ClusterNet*, allowing sample-level classification from the entire point cloud.

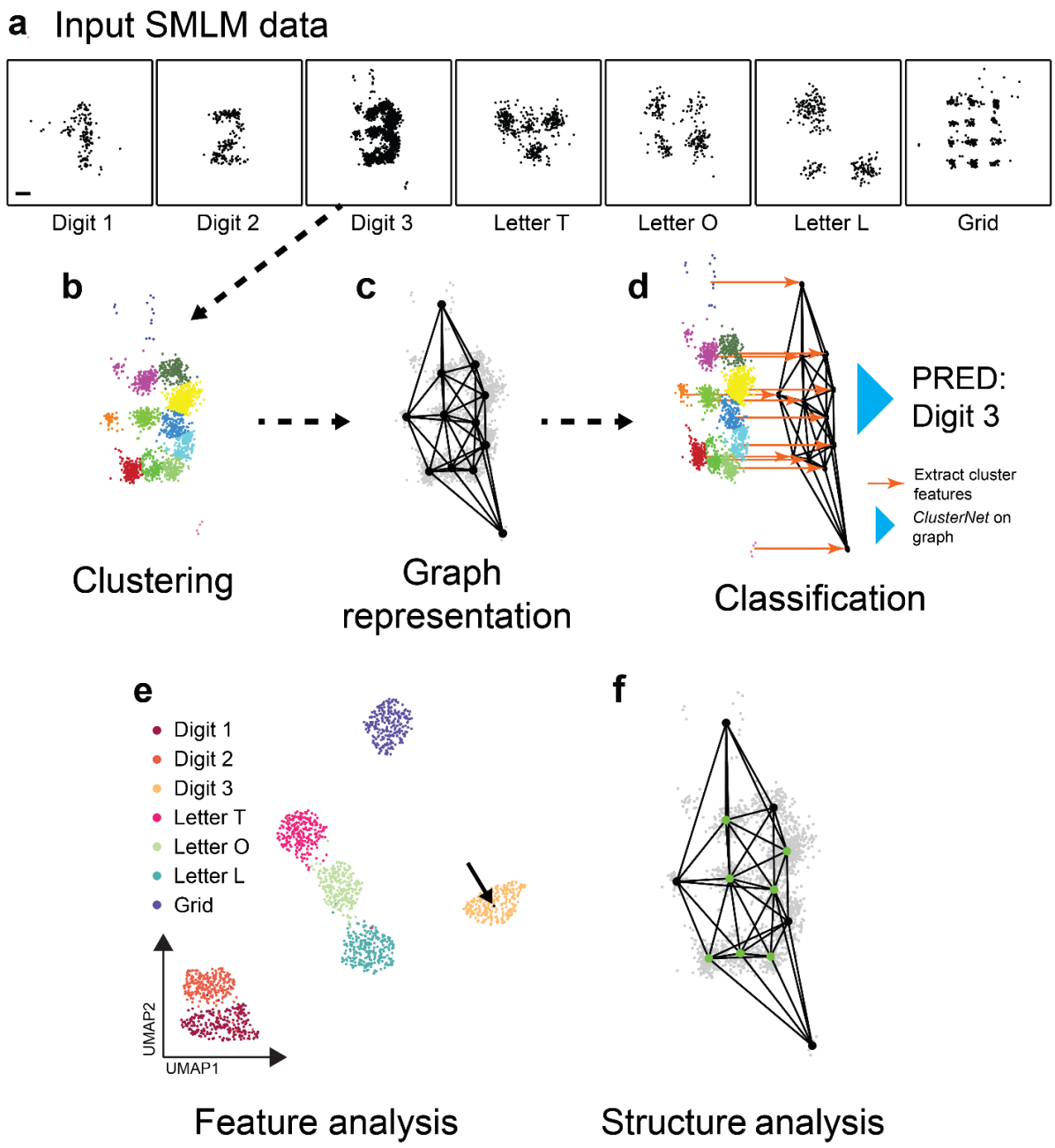

Figure 3.1. SMLM data classification pipeline. a Example SMLM ROIs for each DNA origami structure, with ground truth labels below [229]. Scale bar: 13 nm. **b** Clusters of localisations from an ROI with ground truth (GT) Digit 3, coloured by cluster identity. c Graph representation of the ROI in b with localisation nodes (grey dots), cluster nodes (black dots), and edges. **d** Handcrafted features (in *ClusterNet-HCF*) or automatically learnt features from deep learning (in *ClusterNet-LCF*) are extracted for each cluster, and the graph composed from these clusters is passed to *ClusterNet* to give a final prediction (PRED). **e** 2D representation of cluster or graph embeddings from UMAP, coloured by GT (reserved test dataset, graph-level features in this case, black arrow: ROI in **b**–**d**. **f** Important subgraph (green nodes) for classification, extracted using SubgraphX.

3.2 Materials and methods

3.2.1 Digits and Letters dataset

We developed and tested our pipeline on regions of interest (ROIs) from an existing DNA-PAINT dataset acquired from DNA origami Digits and Letters (Figure 3.1a) [229]. We chose this dataset for the following reasons. It is large and contains point-cloud data with the spatial coordinates and an estimate of their precision for each single-molecule localisation. It also had readily available ground truth labels, making it well-suited to training and validating a deep learning model. Finally, our pipeline could be compared against previous methods used to classify this dataset. The dataset additionally has features characteristic of most SMLM datasets, such as clustered localisations arranged in a non-random pattern and a wide range in the number of localisations and the localisation precision between ROIs and classes (Table 3.1, Figure 3.2). As found for more complex biological datasets, not all the localisations in each ROI contribute to the structure being analysed or are as relevant to the ground-truth label (e.g. background localisations or localisations far from any binding site).

This dataset combines two originally separate datasets, one containing the digits and grid and the other containing the letters. Further details on how these datasets were acquired can be found at [171] for the digits and grid, and at [228] for the letters. The combined Digits and Letters dataset comprises 22,047 SMLM regions of interest (ROIs) from DNA-PAINT data acquired from DNA origami structures, with one structure per ROI. This included Digits (4155 x Digit 1, 4943 x Digit 2, 2541 x Digit 3), Letters (1161 x Letter L, 991 x Letter T, 560 x Letter O) and a 3 x 4 grid (7696 x Grid), imaged separately per class [171,228].

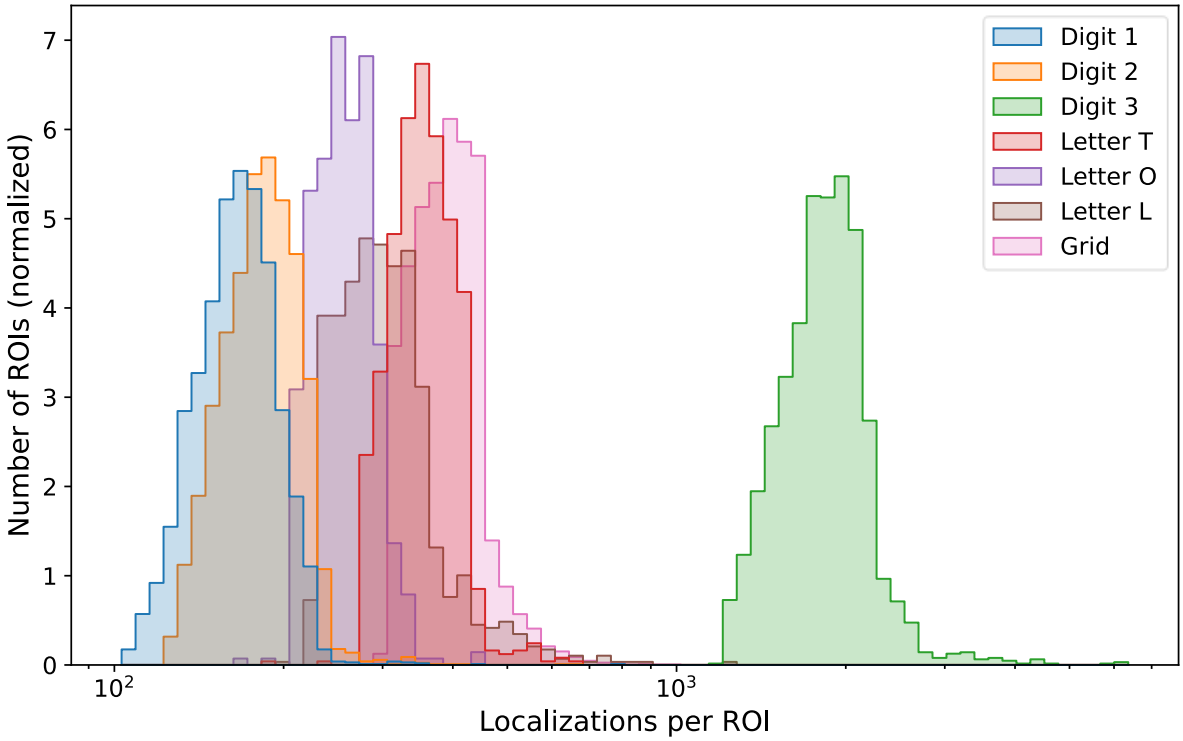

Figure 3.2. Localisations per ROI for the digits and letters dataset. Y-axis is normalised such that each histogram has the same area. X-axis is a log scale.

Table 3.1. Characterising the digits and letters dataset. σ: localisation uncertainty in *x* and *y*.

Class	Min. localisations per ROI	Max. localisations per ROI	Mean localisations per ROI	Min. σ [nm]	Max. σ [nm]	Mean σ [nm]
Digit 1 (n=4155)	103	788	167	9	30	16
Digit 2 (n = 4943)	127	645	184	9	94	15
Digit 3 (n = 2541)	1186	6360	1855	9	37	15
Letter T (n = 991)	185	650	357	11	38	21
Letter O (n = 560)	167	454	258	13	38	24
Letter L (n = 1161)	197	1228	313	6	31	13
Grid (n = 7696)	305	1244	395	8	73	13

3.2.2 Preprocessing and clustering

Each ROI was pre-processed and clustered in preparation for feature extraction and graph representation as follows. The point cloud was initially partitioned into clusters following a similar approach to PointTransformer v2 [230]. First, the *xy* localisation

coordinates for each ROI were converted from a MATLAB file to an Apache Parquet file and labelled in the metadata according to the character they represent. This gave seven classes (Digit 1, Digit 2, Digit 3, Letter T, Letter O, Letter L or Grid). Next, *k*-means clustering was applied to each ROI with *k* set to 12 to ensure every well-separated group of DNA-PAINT binding sites was recovered (Grid had 12 well-separated binding sites; Digits had more binding sites, but not well-separated). Clusters with two or fewer localisations were discarded to allow future calculation of the convex hull and principal components. We chose *k*-means clustering over DBSCAN, as DBSCAN requires careful tuning of two hyperparameters rather than one and risks dropping many important localisations considered as noise [190].

3.2.3 Handcrafted feature extraction

Eight handcrafted features were calculated for each cluster. First, the number of localisations per cluster (count), the mean squared distances of localisations within the cluster from the cluster centroid (radius of gyration squared) and the perimeter from its convex hull were calculated. Next, principal component analysis (PCA) was used to calculate the variance of the clusters along the two principal components, λ_0 and λ_1, where $\lambda_0 > \lambda_1$. These were used to calculate linearity ($\frac{\sqrt{\lambda_0}-\sqrt{\lambda_1}}{\sqrt{\lambda_0}}$), planarity ($\sqrt{\frac{\lambda_1}{\lambda_0}}$), length ($2.35 \times \lambda_0$) and area ($2.35^2 \times \lambda_0\lambda_1$) of each cluster (the full width at half maximum of a Gaussian is given by 2.35 times the standard deviation) [231,232]. In 3D, planarity is given by $\frac{\sqrt{\lambda_1}-\sqrt{\lambda_2}}{\sqrt{\lambda_0}}$, where λ_2 is the third principal component [231]. The data was 2D in this study, so $\lambda_2 = 0$ and linearity = 1 – planarity, which made one of linearity or planarity redundant. However, nothing is lost by including the redundant

features in 2D, except for a small increase in memory usage and training time. Finally, the density for each cluster was calculated by dividing the count by the area.

3.2.4 Graph construction

Each SMLM ROI was represented as a graph using PyTorch Geometric [233]. Each graph contained localisation and cluster nodes (from clustering as described), where each localisation node belonged to its cluster node and undirected edges connected each cluster node to itself and its nearest $\mathcal{N}_{cluster}$ neighbours, where $\mathcal{N}_{cluster} = 5$ for the digits and letters dataset. The position of each localisation node was given by its coordinates, and the position of each cluster node was given by the centre of mass of its constituent localisations. *xy* node positions were normalised to between -1 and 1, $x \rightarrow \frac{2(x-\min(x))}{\max(x_{range}, y_{range})} - 1$ and $y \rightarrow \frac{2(y-\min(y))}{\max(x_{range}, y_{range})} - 1$, where the minimum and range were measured over the parent graph. Cluster nodes initially had either no features or the eight handcrafted features, depending on the downstream model (learnt or handcrafted cluster features). When present, these features, *h*, were normalised to between zero and one, $h \rightarrow \frac{h-\min(h)}{\max(h)-\min(h)}$, where the minimum and maximum values were measured over the whole training set. The localisation nodes had no input features.

3.2.5 Dataset partitions

20,367 graphs (Digit 1: 3915, Digit 2: 4703, Digit 3: 2301, Letter L: 921, Letter T: 751, Letter O: 320, Grid: 7456) were used for *k*-fold cross-validation, where *k* = 5. A further 240 graphs from each class formed a reserved test set. The five different splits in cross-validation each contained a training (64%), validation (16%), and test set (20%, non-overlapping between splits). For each split, the training, validation,

and test set each had the same proportion of classes as the overall cross-validation dataset.

3.2.6 Model architectures

Two different neural network models were developed to classify each graph: *ClusterNet-HCF* (Handcrafted Cluster Features) and *ClusterNet-LCF (Learned Cluster Features)*. *ClusterNet-HCF* passed handcrafted cluster features, as described, and the positions of the cluster nodes, through a novel graph neural network, *ClusterNet*. *ClusterNet-LCF* instead generated cluster features via deep learning using an additional point-based module, *LocNet*, and passed them, with cluster position, through *ClusterNet* in a single trainable network. Descriptions of *LocNet* and *ClusterNet* are given below (Figure 3.3).

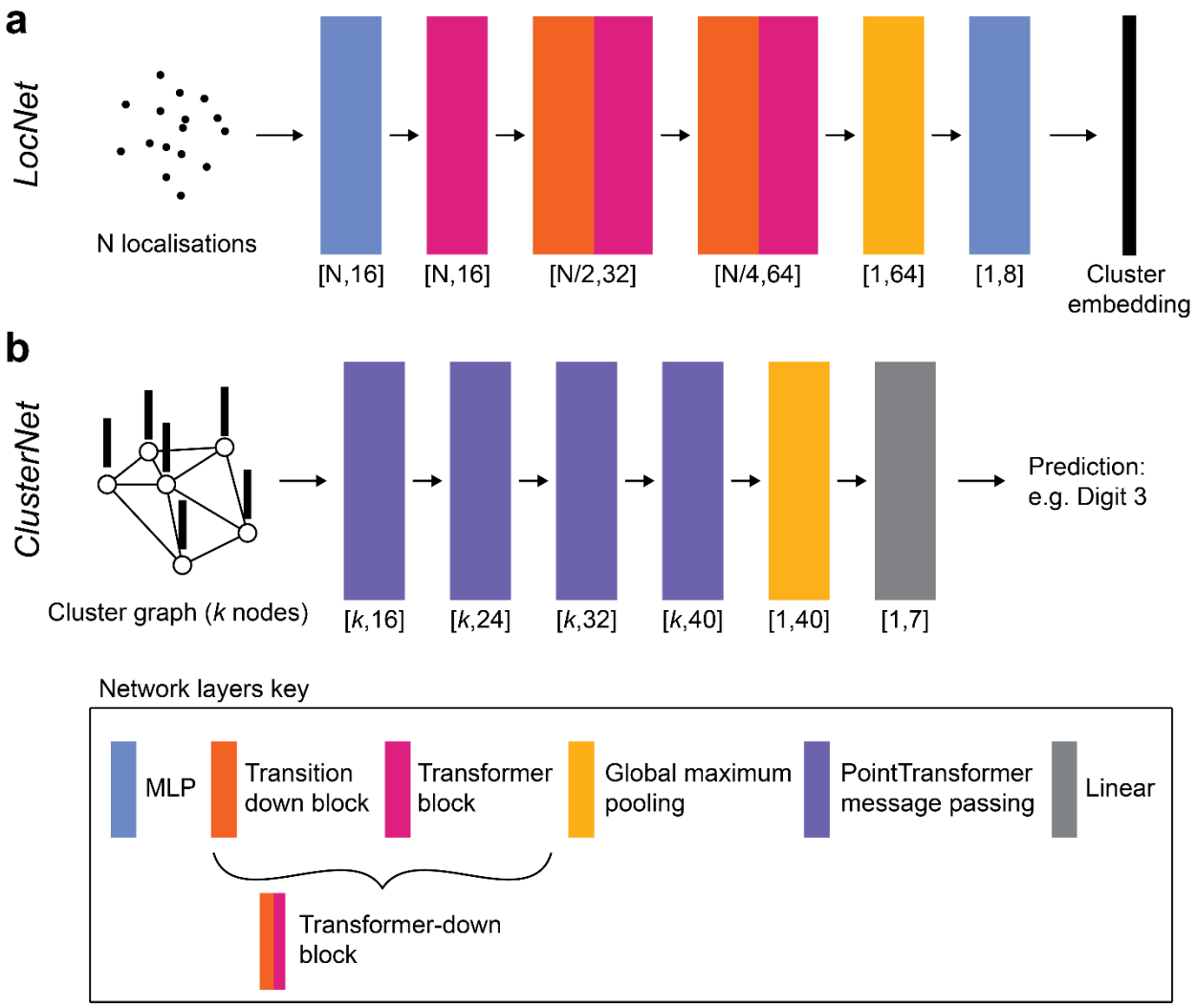

Figure 3.3. *LocNet* and *ClusterNet* architectures. a *LocNet* transforms input SMLM localisations for a cluster into a cluster embedding, via a PointTransformer v1-based network [161]. **b** *ClusterNet* classifies cluster graph, *k* nodes with handcrafted or *LocNet* embedded features, using PointTransformer-based convolutions as message passing layers. Output dimensions of each layer are given in square brackets [X, Y], where X: number of localisations (**a**) or nodes (**b**), Y: per localisation/node feature size.

LocNet architecture

LocNet acted on each cluster independently, taking the constituent localisation node positions, $\mathbf{p} \in \mathbb{R}^2$, as input and embedding the localisations into a feature vector (length 8) for each cluster using PointTransformer v1 [156,161] (Figure 3.3a). The feature vector was chosen to be length 8 to allow a fair comparison between *ClusterNet-HCF*, which used an 8-dimensional input feature, and *ClusterNet-LCF.*

Increasing the dimension of this feature vector may allow *ClusterNet-LCF* to represent and classify more complex structures.

PointTransformer v1 was adapted from the example in PyTorch Geometric [161,233]. The PointTransformer was composed of an initial multilayer perceptron (MLP), a transformer block, two transformer-down blocks, global maximum pooling from localisations into the cluster and a final output MLP.

First, the features for each localisation node, i, were inputted to an MLP with ReLU activation function and batch normalisation, but as there were no input features for the localisations, each feature was set to a dummy value (a vector of ones). Next, the output localisation node features from the MLP were inputted to a transformer block. Each transformer block was composed of an initial linear layer with ReLU activation, a PointTransformer convolution and a final linear layer with ReLU activation. For each localisation node, i, the PointTransformer convolution gave an output feature vector

$$\mathbf{x}_i' = \max_{j \in \mathcal{N}_1(i) \cup \{i\}} \alpha_{i,j} \left(\mathbf{W}_3 \mathbf{x}_j + \delta_{ij} \right) \tag{3.1}$$

, where $\mathcal{N}_1(i)$ denoted the $\mathcal{N}_1$ nearest nodes, the vector attention coefficients were given by

$$\alpha_{i,j} = \text{softmax}\left(\gamma_{\mathbf{\Theta}} \left(\mathbf{W}_1 \mathbf{x}_i - \mathbf{W}_2 \mathbf{x}_j + \delta_{i,j} \right) \right) \tag{3.2}$$

and the position embedding by

$$\delta_{i,j} = h_{\boldsymbol{\omega}} \left(\mathbf{p}_i - \mathbf{p}_j \right), \tag{3.3}$$

where $\mathbf{x}_i$ and $\mathbf{x}_j$ were the input features of nodes i and j, $\mathbf{p}_i$ and $\mathbf{p}_j$ were the 2D coordinates of nodes i and j, $\gamma_{\mathbf{\Theta}}$ and $h_{\boldsymbol{\omega}}$ were MLPs parameterised by $\mathbf{\Theta}$ and $\boldsymbol{\omega}$ respectively with batch normalisation and ReLU activation function, and $\mathbf{W}_1, \mathbf{W}_2$ and $\mathbf{W}_3$ were learned weights of a linear layer. The PointTransformer convolution calculated the maximum of neighbouring features, rather than the sum as was used in the original architecture [161]. This was because SMLM data is prone to artefacts, and maximum pooling can be robust to outliers and missing points and encourage the model to learn the overall structure [135,156,234]. However, maximum pooling doesn't capture the exact structure or distribution of a graph and its features [234]. While this was only theoretically motivated at this point, this was empirically tested on a more challenging classification task in the following chapter, where we also trial attention-based pooling (Sections 4.2.9 and 4.3.2).

The output localisation node features were then inputted to the transformer-down blocks. Each transformer-down block consisted of a transition down block, followed by a transformer block. In the transition down block, the localisation node features were inputted to a MLP with ReLU activation function and instance normalisation, then a fraction, $r_{LocNet,}$ of the nodes in the cluster were chosen using farthest point sampling and assigned the maximum of the features of their $\mathcal{N}_2$ nearest neighbours and themselves. $\mathcal{N}_1$ and $\mathcal{N}_2$ were varied via the same parameter, $\mathcal{N}_{LocNet}$, or in other words, $\mathcal{N}_1 = \mathcal{N}_2 = \mathcal{N}_{LocNet}$. For the digits and letters dataset $r_{LocNet} = 0.5$ and $\mathcal{N}_{LocNet} = 5$.

Finally, global maximum pooling aggregated the localisation node features into the feature vector for the cluster, which was inputted to a MLP with a ReLU activation function, a plain last layer, and no normalisation, giving the final embedding for the

cluster. Global maximum pooling was used rather than global mean pooling as in the original architecture, to be more robust to noise and outliers, as outlined above [161]. Output cluster node features from *LocNet* were then scaled to between 0 and 1 using the sigmoid function.

ClusterNet architecture

ClusterNet acted on the graph of cluster nodes with their feature vectors and their *xy* position (Figure 3.3b). It was composed of four message passing layers with PointTransformer convolutions, a global maximum pooling layer over the cluster features, resulting in a feature vector for the graph, and a final linear layer, generating a class prediction.

First, cluster node features were inputted to the four message passing layers. At each layer, the feature for each node was updated according to the PointTransformer convolution defined above (Equations 3.1-3.3), where the neighbours, $\mathcal{N}_1(i)$, of each cluster node, i, were defined by the edges from graph construction. After the fourth message passing layer, the cluster features were aggregated into a feature vector for the graph using maximum pooling. This was inputted to a linear layer followed by the log softmax function to give the log probabilities of the graph belonging to each class.

3.2.7 Training procedure

For each split in *k*-fold cross-validation, the model was trained on the training dataset for 100 epochs using the ADAM optimiser with a learning rate of 0.001, a weight decay of 0.0001 and a batch size of 128 [205]. During training, a weighted random sampler that over-sampled from the minority class and under-sampled from the majority class was used to encourage equal performance across the classes.

Further, random rotations in the *xy* plane were applied to the graphs for data augmentation. For each graph, the model outputted the log probability for each class, and the negative log-likelihood loss was calculated. After each epoch, the model was evaluated on the validation set. The model that gave the lowest loss on the validation set over all the epochs was chosen as the best model. Training for 100 epochs on a NVIDIA GeForce RTX 2060 with 6GB RAM took ~1.5 hours. *ClusterNet-LCF* was trained in an end-to-end manner, meaning that *LocNet* and *ClusterNet* were trained together as a single network.

3.2.8 Evaluation procedure

The best model for each split was evaluated on the test set for each split. For evaluation, each graph (without random rotation) was classified according to the highest probability class from the model. The predictions for each class were evaluated using recall and combined into a single metric for all the classes using the arithmetic mean (balanced accuracy) [235]. The probabilities for each class were evaluated using the area under the receiver operating characteristic curve (AUROC). More information on the metrics used to evaluate performance can be found in the Appendix (Section 7.1.1).

3.2.9 Evaluation on the reserved test set

The performances of *ClusterNet-HCF* and *ClusterNet-LCF* were ultimately compared on the reserved test set. First, the entire dataset (excluding the reserved test set) was split into a training (80%) and validation (20%) set, by randomly taking ~20% of each class into the validation set. Next, each model was trained on the training set and saved when the loss was lowest on the validation set. These models were then evaluated on the reserved test set following the evaluation procedure above.

3.2.10 Feature analysis via UMAP

The relative discriminative power of the per-cluster (handcrafted vs deep) and whole-graph (deep) features was measured by comparing the separation between classes in a 2D representation of the feature space generated using UMAP (Uniform Manifold Approximation and Projection) [236]. Other methods for dimension reduction were considered, such as principal component analysis (PCA) and t-SNE (t-distributed stochastic neighbour embedding) [184]. UMAP was preferred as like t-SNE it is non-linear, so can capture complex relationships in the data [237]. Further, it is faster than t-SNE, and has been successfully used to visualise handcrafted features and trajectories of particles in SMLM data [150,175,236]. Four sets of features were visualised: handcrafted cluster features, cluster features embedded by *LocNet*, cluster features after the fourth message passing layer of *ClusterNet*, and whole-graph features after the final maximum pooling layer of *ClusterNet*. The handcrafted and *LocNet* features only consider the structure at cluster-level and smaller, and are later referred to as isolated per-cluster features. The cluster features from *ClusterNet* have a larger receptive field, incorporating information from neighbouring clusters. Finally, the whole-graph features from *ClusterNet* pool information over all constituent clusters. In all cases, features were normalised by subtracting the mean and dividing by the variance of each feature independently, measured over the entire dataset excluding the reserved test set. UMAP generated a lower-dimensional (2D) representation of the features, with 20 neighbours for each feature vector and 0.5 minimum distance in the lower-dimensional space. UMAP plots can be visualised interactively, displaying the parent ROI, GT, and prediction.

3.2.11 Structure analysis via SubgraphX

Existing graph-explainability algorithm SubgraphX was used to identify structures in the graphs constructed from the cluster nodes (cluster graphs) and their features (handcrafted or embedded by *LocNet*) that were important for the classification [238]. SubgraphX follows a game-theory inspired approach to identify the most important connected subgraph for the classification of a graph. The importance of each subgraph is given by its Shapley value, which estimates the contribution of each subgraph by measuring the difference in the model prediction when including or excluding the subgraph [239]. To make this tractable, SubgraphX uses Monte Carlo tree search to search for the optimal subgraph and Monte Carlo sampling to estimate the Shapley value.

SubgraphX was preferred over similar methods due to its high performance in non-SMLM graph data benchmarks [181]. SubgraphX searches for the most important connected subgraph in the cluster graph by feeding subgraphs induced by different sets of nodes into *ClusterNet* and measuring their relative contribution to the model's prediction. We chose the 'split' method of assessing subgraphs, where nodes outside of a subgraph being assessed are removed from the graph [238,240]. This avoids the positions of the cluster nodes being set to (0, 0) coordinates in the 'zero_filling' method, which would have a greater effect on the supra-cluster structure and message passing in *ClusterNet*.

The optimal subgraph was required to have no more than 8 nodes, and the number of rollouts was increased from 20 (default value) to 100, to minimise instability of the prediction (values of other parameters in Table 3.2). In our visualisations, self-loops and edge direction are not shown.

Table 3.2. SubgraphX parameters.

Parameter	Value	Default value
rollout	100	no
min_atoms	5	yes
c_puct	10.0	yes
expand_atoms	14	yes
high2low	False	yes
local_radius	4	yes
sample_num	100	yes
reward_method	mc_l_shapley	yes
subgraph_building_method	split	no

Positive and negative fidelity scores measured the necessity and sufficiency, respectively, of the optimal subgraph (subset of cluster nodes) for the prediction [179]. They calculate the difference between the probability of the predicted class when the whole graph is fed into the model, and when the graph minus the subgraph (positive fidelity) or only the subgraph (negative fidelity) is fed into the model [180]. Best performance is given by a positive fidelity of one and a negative fidelity of zero, and worst performance by a positive fidelity of zero and a negative fidelity of one.

3.3 Results

3.3.1 Classification pipeline and performance

Both *ClusterNet-HCF* and *ClusterNet-LCF* successfully classified the data from the seven DNA origami structures. *k*-means clustering was used to pre-process the localisation data from the Digits and Letters, although any clustering algorithm could be applied. Both models achieved the maximum value of 1.00 for the area under the receiver operator curve (AUROC) for every class in the reserved test set. *ClusterNet-HCF* outperformed *ClusterNet-LCF* on the reserved test set (Table 3.3) and on the training, validation, and test folds (Tables S3–5, Appendix, Section 7.2).

Table 3.3. Classification performance on the digits and letters dataset. Recall values for *ClusterNet-HCF* and *ClusterNet-LCF* on the reserved test set.

	Digit 1	Digit 2	Digit 3	Letter T	Letter O	Letter L	Grid	Mean ± S.D.
ClusterNet-HCF	0.99	0.98	1.00	0.96	0.99	0.99	1.00	**0.99 ± 0.01**
ClusterNet-LCF	0.97	0.94	1.00	0.93	0.95	0.91	0.99	0.96 ±0.03

ClusterNet-HCF and *ClusterNet-LCF* both outperformed previous results on the same dataset (Table 3.4). In addition, the accuracy for previous methods (*nanoTRON* and point cloud registration) was calculated for classification within subsets (Digits/Grid or Letters) of the seven classes of DNA origami structure, whereas the accuracy of *ClusterNet* was from classification over all seven classes, which is a harder task [171,228]. The previous methods would be expected to give lower accuracies than reported (Table 3.4), if tested on the same task as *ClusterNet*.

Table 3.4. Classification performance compared with previous methods on the digits and letters dataset. Accuracy values for *ClusterNet-HCF* and *ClusterNet-LCF* on the reserved test set and closest comparisons with previous methods, *nanoTRON* and point cloud registration [171,228]. **a** The test dataset was reserved from a larger dataset, formed from an 11-fold expansion (via data augmentation) of 21k unique ROIs [171]. **b** Accuracy for *ClusterNet-HCF* and *ClusterNet-LCF* included misclassifications between all Digits/Grid and Letters structures. Accuracy for point cloud registration included misclassifications only within either Digits/Grid or Letters subsets (not between them). *nanoTRON* was only tested on Digits/Grid structures. **c** The dataset included only Digits and Grid. **d** 5,000 ROIs were randomly sampled from all ROIs over all Digits and Grid classes. **e** Accuracy when misfolded DNA origami Letter structures were classified as one of the three Letters (with no extra 'misfolds' class), as in our method. 200 ROIs sampled per GT class. **f** Calculated from Table S6 and S7, Appendix, Section 7.2. 240 ROIs per GT class in the reserved test dataset.

Model	Digits and Grid	Letters
nanoTRON	~98% (n = 74k) [a,b]	n/a [c]
Point cloud registration	96.4% (n = 5,000) [b,d]	89.0% (n = 600) [b,e]
ClusterNet-HCF	**99.1%** (n = 960) [b,f]	**98.2%** (n = 720) [b,f]
ClusterNet-LCF	97.4% (n = 960) [b,f]	93.1% (n = 720) [b,f]

Classification performance (recall) was greater for the Digits and Grid than for the Letters (Table 3.3). This was expected as there were misfolded DNA origami structures in the Letters dataset, which did not resemble the intended Letter [228]. This had less impact on the Digits and Grid, as the authors of the published dataset excluded some of the misfolded Digits and Grids in particle picking [228]. There may be an additional contribution to this from the imbalance in the validation dataset used during training (fewer Letters than Digits and Grid), despite attempts to mitigate this with weighted random sampling of the training set based on the prevalence of each class.

3.3.2 Feature analysis via UMAP

We then compared the separation between classes in the 2D representation of the feature space generated by UMAP. This showed that isolated per-cluster features, with no incorporation of supra-cluster structure, could not separate the classes, except for the Digit 3 (Figure 3.4a and Figure 3.4d). However, the handcrafted features did slightly separate the Letters T, O and L from Digits 1 and 2 and the Grid (Figure 3.4a). This reflected the differences in the DNA origami structures, where the Digits have continuous structures with no spaces between their binding sites, while the Letters have discrete structures, with groups of binding sites well separated from each other [228]. The deep cluster features generated by *LocNet* did not separate the Digits and Letters but better separated the Grid, suggesting that they captured different aspects of the input data from the handcrafted features (Figure 3.4d). Digit 3 may have been separable at the per-cluster level because it had a significantly different localisation density from the other classes (Figure 3.2 and Table 3.1). However, the remaining classes could not be separated based on differences in

localisation density alone, as they had a similar average number of localisations per ROI (Figure 3.2 and Table 3.1).

The cluster features generated by *ClusterNet* significantly improved separation into each class, showing the importance of considering the supra-cluster structure via information from neighbouring clusters (Figure 3.4b and 3.4e). Similar to the handcrafted and *LocNet* cluster features, the Grid and Digit 3 classes were the most clearly separated for both models. The remaining Digits, 1 and 2, were separated from the Letters, although there was significant overlap within these two groups.

The whole-graph features further improved the separation of the classes (Figure 3.4c and 3.4f). This showed the importance of moving from per-cluster to whole-graph features, highlighted by Digits 1 and 2, which changed from overlapping to well-separated for *ClusterNet-HCF* (Figure 3.4b and 3.4c). *ClusterNet-HCF* had the most compact and well-separated representations, reflecting its classification performance (Figure 3.4c). The Letters and Digits 1 and 2 were still assigned into two distinct groups. We also observed that the whole-graph features for misclassified ROIs were normally located at the edges of each class group (Figure 3.5).

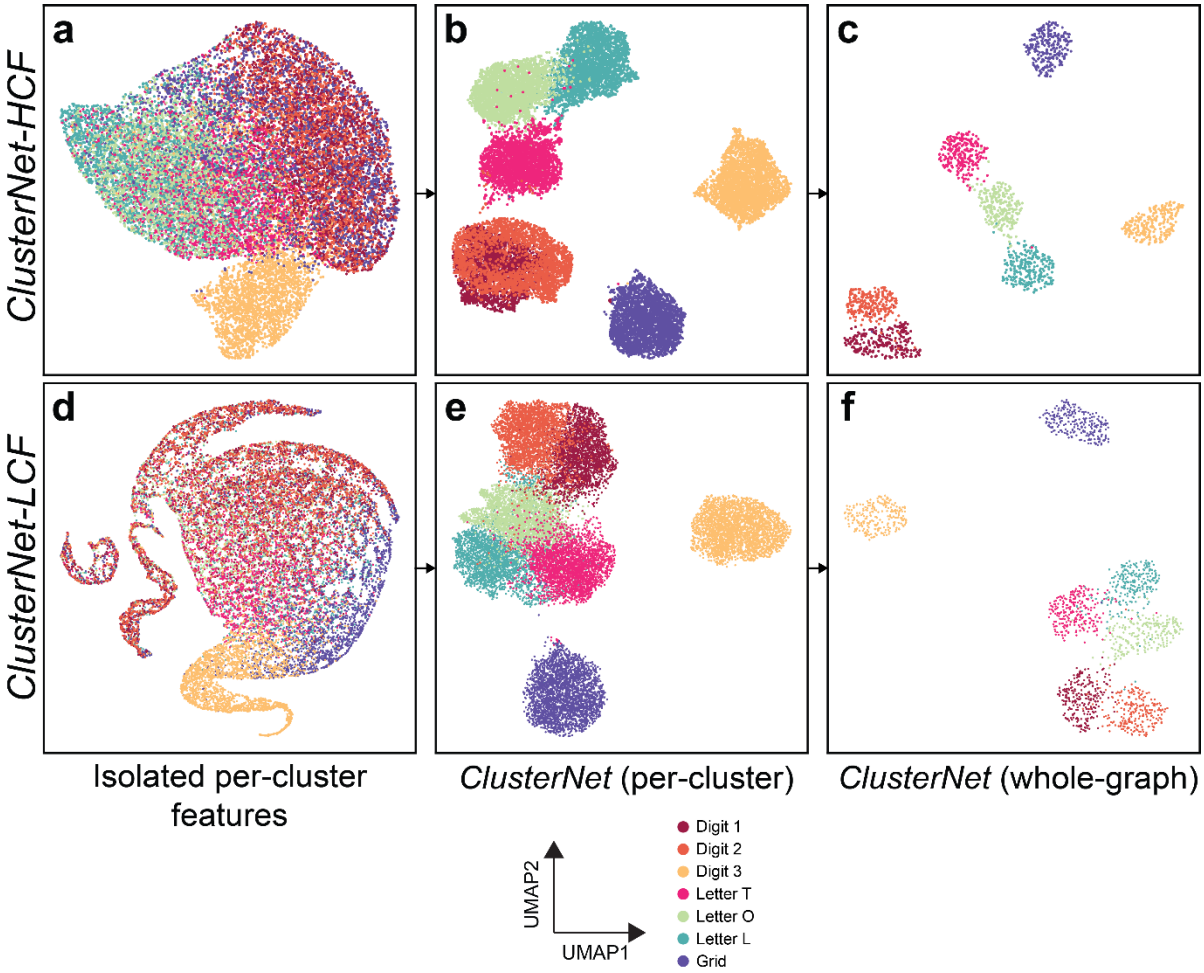

Figure 3.4. Feature analysis of SMLM ROI classification results. 2D feature representations (from UMAP) for the ROIs in the reserved test set, for *ClusterNet-HCF* (**a**-**c**) and *ClusterNet-LCF* (**d**-**f**). The features incorporate larger structure from left to right: isolated per-cluster handcrafted or *LocNet* embedded features (**a**, **d**); per-cluster features after message passing but before global pooling in *ClusterNet* (**b**, **e**); and whole-graph features aggregated from *ClusterNet* (**c**, **f**).

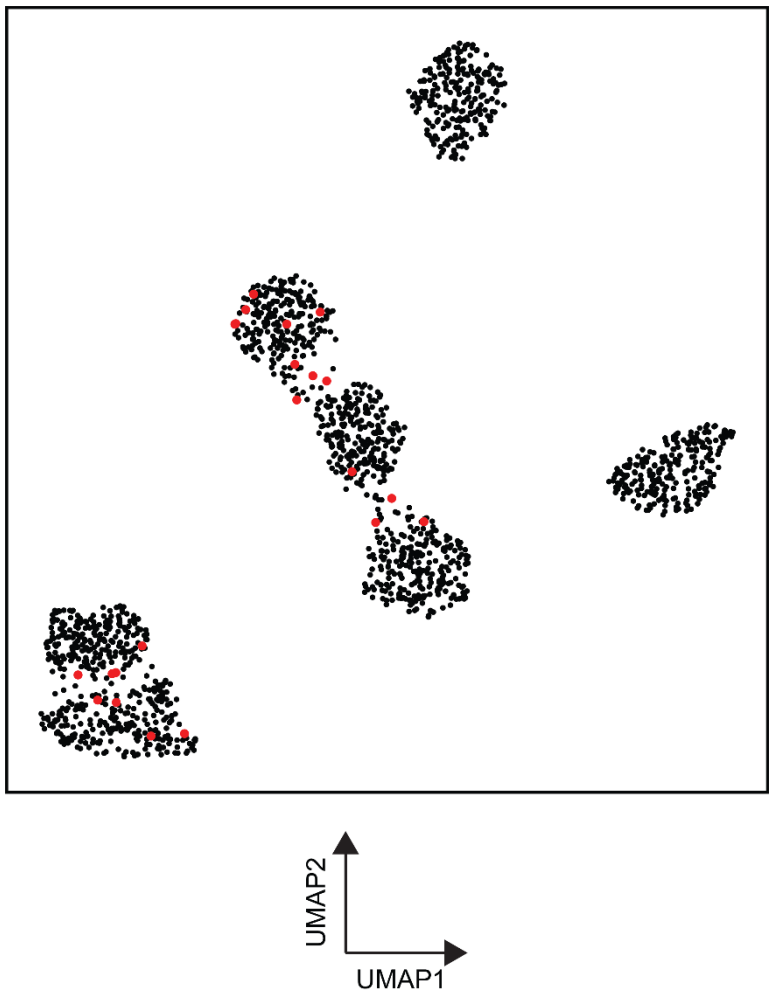

Figure 3.5. Feature analysis for incorrectly classified ROIs. 2D whole-graph feature representation (from UMAP) for *ClusterNet-HCF* for the reserved test dataset ROIs, coloured according to whether they were correctly (black) or incorrectly (red, larger for clarity) classified.

3.3.3 Structure analysis via SubgraphX

Different classes may be distinguished by the arrangement of some or all of their clusters (supra-cluster structure). We tested a method for identifying these structures by finding the important parts of a cluster graph for the classification of an SMLM ROI. SubgraphX searches for the most important subgraph for the graph classification [238], in this case, a subset of clusters and their supra-cluster structure. We analysed the classification results from *ClusterNet-HCF* (the best performing model).

The 2D representation of the whole-graph feature space (Figure 3.4c) allowed us to choose ROIs closest to and furthest from the rest of their class members, measured by the distance to the centroid of the class features. After graph representation and classification, SubgraphX returned the most important subgraph for the classification that it found (Figure 3.6; parameters in Section 3.2.11). Positive and negative fidelity metrics (Fid+, Fid-) measured the necessity and sufficiency, respectively, of the subgraph for the classification (Section 3.2.11; best performance: Fid+ = 1, Fid- = 0; worst performance: Fid+ = 0 and Fid- = 1) [179].

An exemplary graph closest to its class members was from a clear DNA-PAINT example of a Digit 3, correctly predicted (Figure 3.6c). The subgraph identified by SubgraphX was both necessary and sufficient for the classification (Fid+: 1.0, Fid-: 2.6×10^{-5}) and reflected the Digit 3 shape that would be expected. An example furthest from its class members may have been a misfolded item, as there only appear to be three well-separated groups of localisations, whereas the Letter O (the GT label) should have four (Figure 3.6l and 3.3a) [228]. It was incorrectly predicted as the Letter T, which was reflected in its location near the other Letter Ts in the UMAP representation. The subgraph extracted by SubgraphX was important for its classification (Fid+: 0.96, Fid-: 2.5×10^{-2}) and resembled the Letter T structure.

In general, the important subgraph for incorrectly classified ROIs did not appear to reflect the structure expected of the correct class (Figure 3.6), and a closer resemblance to the incorrectly predicted class could sometimes be discerned (Figure 3.6h,l,m,n). This suggests that this approach could allow us to begin to identify specific patterns in the supra-cluster structure that lead to classification results in SMLM datasets, including misclassifications.

We also note that some clusters appeared outside of the designed DNA origami patterns, arising from imperfections in the original sample and raw data acquisition, influencing classification with *ClusterNet* (Figure 3.6a,e,h). SubgraphX showed that these spurious clusters were sometimes considered important in classification (Figure 3.6a,h), also explaining some incorrect results (Figure 3.6h).

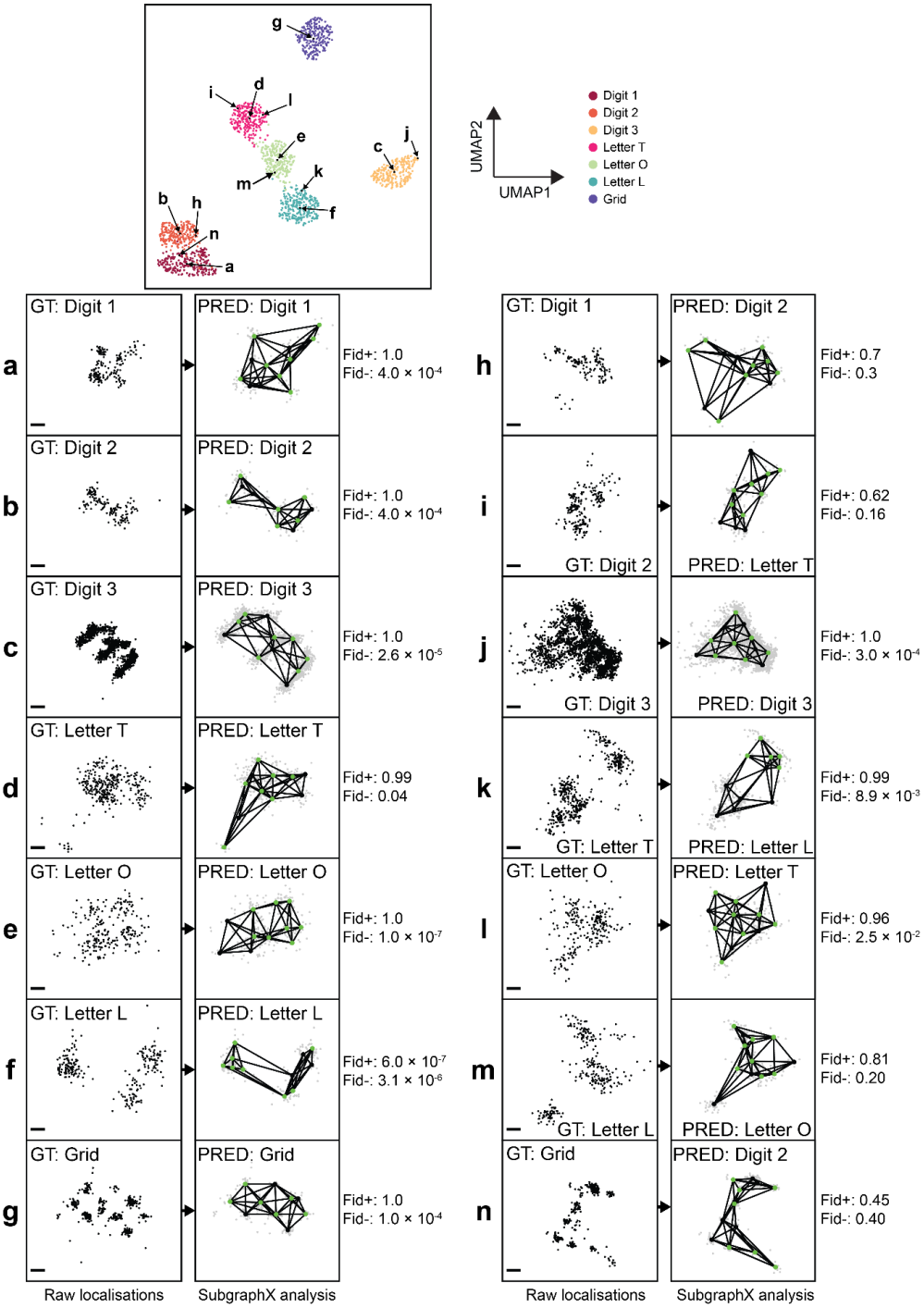

Figure 3.6. Structure analysis of SMLM ROI classification by *ClusterNet-HCF*. 2D whole-graph feature representation (from UMAP) for *ClusterNet-HCF* for the reserved test dataset ROIs (Top), highlighting ROIs with whole-graph features (from their classified cluster graph) closest to (**a**-**g**) or furthest from (**h**-**n**) their fellow class members. (**a**-**n**) DNA-PAINT localisations (scale: 13 nm) represented as a graph, classified by *ClusterNet-HCF,* and analysed with SubgraphX (GT: ground truth, PRED: prediction). SubgraphX results show the important subgraph (supra-cluster structure) for the class prediction (green nodes). Positive fidelity (Fid+) and negative fidelity (Fid-) measure the necessity and sufficiency, respectively, of the important subgraph (best performance: Fid+ = 1, Fid- = 0; worst performance: Fid+ = 0 and Fid- = 1).

3.4 Discussion

We have demonstrated *ClusterNet*, a pipeline for classifying SMLM fields of view, which combines per-cluster features extracted from the localisation pattern with supra-cluster structure extracted via graph-based deep learning. Per-cluster features may either be handcrafted and fully interpretable (e.g. cluster area, perimeter) or also obtained from deep learning. We have also incorporated analysis of the features and structures learnt during classification using UMAP and SubgraphX [236,238].

Combining handcrafted cluster features with *ClusterNet (ClusterNet-HCF)* outperformed the model that extracted deep features at both the per-cluster and whole-graph scale (*ClusterNet-LCF*). This shows that incorporating known handcrafted features can boost classification performance, besides being more interpretable. This reflects the success achieved by previous methods that have combined handcrafted cluster features with machine learning to classify SMLM data, such as ASAP, SEMORE, ECLIPSE and SuperResNet (Section 1.3.2) [150,163-165]. *ClusterNet-HCF* improves upon these methods, which are limited to classification of clusters of localisations, to classify structures formed from multiple clusters. In other SMLM datasets, deriving discriminative handcrafted features may be more difficult, as the differences may not be easily discernible by eye or have a straightforward mathematical representation. We anticipate that generating cluster features via deep learning, as with *LocNet*, may be more useful in these cases, although this was not supported by our work classifying whole-cells later in this thesis (Chapter 4). Further, extending *ClusterNet-LCF* to include photophysical parameters as features of the localisations, or weighting localisations by these parameters in the calculation of handcrafted features, could allow the model to account for localisation uncertainty.

Our classification pipeline outperformed previous methods applied to the Digits and Letters dataset (Table 3.4). One was designed for particle fusion and requires no training and minimal supervision [228], and the other was an image-based model [171]. An alternative image-based method could likely match our classification performance, but it would require large images (e.g. 1 pixel per nm) to incorporate the high precision of the SMLM data, as the localisations are binned into pixels (Section 1.2.2) [100,103]. Such rendering does not scale well for larger ROIs or when moving from 2D to 3D SMLM data, as the size of the image increases exponentially [103]. Instead, point-based methods such as the method developed here can be simply extended without a large increase in the size of the data representation. For *ClusterNet*, this would include changing the handcrafted features to characterise 3D rather than 2D shapes (e.g. area to volume), changing the dimension of the position vector for each localisation and node from 2D to 3D and including 3D rotations during training to learn invariance to rotations not in the optical plane.

Analysing high-dimensional features of SMLM data can help to identify and investigate the underlying substructures that distinguish different SMLM data structures. This has mainly been restricted to visualisation of the features in a lower-dimensional space, via dimension reduction techniques such as UMAP [103,150,164,175]. For the Digits and Letters dataset, this revealed intra-class variability and identified misfolded structures by visualising graphs closest to and furthest from the centre of the class, without requiring an additional class to capture the misfolds [228]. This also demonstrated that the supra-cluster structure was required to separate the classes and revealed subpopulations such as Digit vs Letter structures without user specification.

Deep learning explainability algorithms can more directly identify the underlying substructures by measuring their impact on the model's prediction. So far, this has been used on SMLM data represented as images, but not on graph representations [136]. For the Digits and Letters dataset, SubgraphX was used to identify the discriminative substructure (subset of clusters in their supra-cluster arrangement) in each FOV represented as a graph. In some cases, this was able to identify the substructure that aligned with our knowledge of the ground truth. However, future work is required to make it more reliable if this is to be applied to other SMLM data.

In this study, the pipeline was demonstrated on DNA-PAINT data, but the pipeline could be applied to a broad range of SMLM data from other techniques that generate point cloud data (e.g. dSTORM and PALM). Our dataset had characteristics common to most SMLM data (e.g. being clustered or containing localisations that are less relevant to the classification task), making it a useful proof of concept. Moving between these different techniques or experimental conditions should not significantly affect handcrafted features that capture the overall shape and size of clusters. Learnt cluster features can adapt to any differences, as seen by the range of data types (e.g. 3D shapes, indoor and outdoor scenes, etc.) that point-based deep learning networks have been applied to [160]. The model can be retrained or adjusted to adapt to any remaining differences.

A simulated dataset, which contained protein organisation that differentiated the classes at different length scales or that mimicked real experimental data from cells (in preparation for Chapter 4), could have been generated and used instead [193]. However, besides the time it would have taken time to build and validate the simulator, the digits and letters dataset allowed us to compare with previous

methods to classify it, that have already been well documented and optimised. Further, the digits and letters dataset contained significant heterogeneity within each class (e.g. misfolds), characteristic of the heterogeneity observed in real experimental data, which would be harder to simulate beyond simply adding in background noise.

In the next chapter, this pipeline is further refined and used to classify SMLM data from human tissue by response to treatment. *ClusterNet* could therefore contribute to realising the use of SMLM data as a new modality in characterisation of phenotype, classification of disease progression and prediction of response to treatment in more complex biological samples [75,124,127,129].

4 Predicting Response to Anti-EGFR Treatment in Metastatic Colorectal Cancer

4.1 Introduction

Approximately 40% of metastatic colorectal cancer patients who receive anti-EGFR treatment do not respond, while suffering the negative side effects and potentially worse outcomes than if they had not received treatment [25-27]. However, recent work suggests that higher protein expression of epiregulin (EREG) and amphiregulin (AREG) is associated with better response to anti-EGFR treatment [20,21,25,70]. Further, the spatial organisation (location, clustering, etc.) of EGFR and its ligands is associated with downstream signalling [38,40-42,47,48,241,242] and response to anti-EGFR and other cancer therapies (Section 1.1.3-1.1.4) [73,74]. This suggests that the spatial organisation of the ligands for EGFR could also help predict response to anti-EGFR treatment.

SMLM can reveal features of the nanoscale protein organisation that discriminates between response to treatment in tissue samples from cancer patients (Section 1.2.3) [75]. Machine learning algorithms can help to classify the SMLM data from different sample types, using simpler models that require handcrafted input features (e.g. protein expression levels) or deep learning models that automatically learn features from the data (Section 1.3.2) [134,136,137]. For example, image-based neural networks (deep learning) have been successfully used to classify cells and tissue imaged via SMLM [136,137]. Post-hoc analysis of these models can reveal the important parts of the input data for the classification, which can help identify new features that discriminate between the classes [136]. Alternatively, point- and graph-based neural network architectures have been applied to SMLM data, realising the precision gain

of SMLM over conventional imaging, but have not yet been used to classify whole cells [103,151,172].

In this chapter, an AI-approach was developed to predict the response to anti-EGFR treatment in metastatic colorectal cancer patients using the spatial arrangement of EREG extracted via SMLM (Figure 4.1). We analysed EREG rather than AREG, due to difficulties in optimising the AREG antibody for SMLM (unpublished) and in imaging AREG and EREG simultaneously, as both primary antibodies were anti-rabbit. Further, previous analysis of protein expression levels in tumour cells indicated that both AREG and EREG give similar results when treated as separate markers or combined into one marker of response to anti-EGFR treatment [70].

This approach was developed and tested on patient tissue samples from the PICCOLO clinical trial, in which the association between EREG expression and response to anti-EGFR treatment has been shown [25,26,70,243]. Tissue samples from patients with no-response or any-response to anti-EGFR treatment were imaged using dSTORM. Cells were manually annotated in the dSTORM data using *locpix* (Chapter 2) and classified using AI-based models. We tested simpler traditional machine learning models, logistic regression and random forest, that used handcrafted features for the expression and organisation of EREG. We also tested a deep-learning method, *ClusterNet* (Chapter 3), which combined either handcrafted or learnt cluster features with the spatial arrangement of those clusters in a graph neural network. The approach was first tested and optimised for per-cell and per-patient prediction using *k*-fold cross-validation (Sections 4.3.1-4.3.5) and then evaluated on a reserved test set of cells, also from PICCOLO (Section 4.3.6). Post-hoc analysis of the models was then performed to identify features that may

predict response to treatment (Sections 4.3.7 and 4.3.8). This showed that the spatial organisation of EREG may predict response to anti-EGFR treatment. Ultimately, this approach could improve upon or be used in combination with existing methods for predicting response that only consider ligand expression.

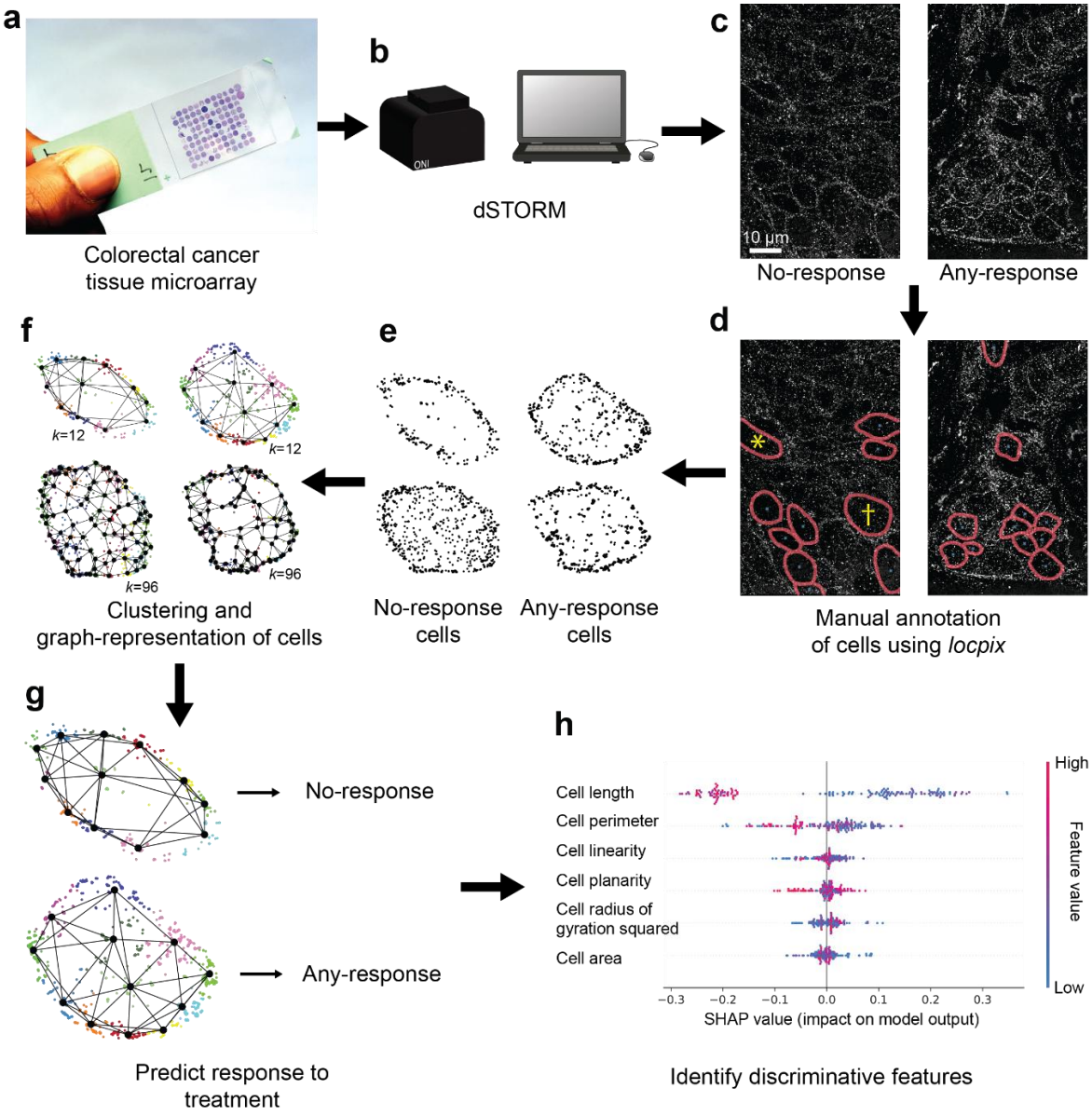

Figure 4.1. Overview of the approach. a Section from a pre-constructed formalin-fixed paraffin-embedded tissue microarray (TMA) block from metastatic colorectal patients who later received anti-EGFR treatment. **b** EREG, a ligand of EGFR, is imaged in the TMAs using dSTORM. **c** Example fields of view from patients with different responses to anti-EGFR treatment. **d** Cells are manually annotated using *locpix* (developed in Chapter 2, red lines: membrane, blue crosses: cell centres). Membrane annotations form closed loops with (yellow asterisk) or without (yellow dagger) FOV boundary. **e** EREG localisations for two example cells per response class. **f** Localisations in each cell are clustered using *k*-means, handcrafted features for each cluster are extracted and the cell is represented as a graph with cluster nodes (black dots) and edges. Localisations are coloured by cluster identity. Clustering and graph representation are shown for different numbers of clusters (*k*) as examples. **g** Response to treatment is predicted using logistic regression, a random forest or *ClusterNet* with handcrafted or learned cluster features (developed in Chapter 3). **h** The features that best predict response to treatment are identified, for example, by SHAP analysis. Computer icon in **b** is from DBCLS (https://togotv.dbcls.jp/en/pics.html) and is licensed under CC-BY 4.0.

4.2 Materials and methods

4.2.1 Tumour dataset from the PICCOLO trial

PICCOLO was a randomised phase III trial of second-line irinotecan with or without panitumumab (anti-EGFR therapy) in advanced colorectal cancer [26,243]. As part of the trial, pre-treatment tumour samples were collected from each patient (mainly from the primary tumours but with ~5% from metastases). The samples from metastases were included in our study, but primarily the samples we used came from the primary tumours. Best response by Response Evaluation Criteria In Solid Tumours (RECIST) criteria within 1-year follow-up from randomisation was recorded [26,244].

In our study, we imaged cells in tumour cores from the pre-constructed formalin-fixed paraffin-embedded (FFPE) tissue microarray (TMA) blocks from PICCOLO, preferred over whole slides to minimise time identifying tumour regions and increase throughput. As mentioned above, these tissue samples primarily came from the patients' primary tumours but also included samples from the patients' metastases. Only patients who received anti-EGFR therapy (panitumumab arm) and tested wild-type (WT) for confounding genetic mutations were included (mutations in *KRAS* codons 12, 13, 61 and 146, *NRAS* codons 12, 13 and 61 and *BRAF* codon 1799T>A). Patients with *BRAF*-mutant tumours were excluded because anti-EGFR therapy is less effective for this subgroup [24]. The best response variables (from RECIST) were grouped into any-response (complete response, partial response) and no-response (radiological progression, clinical progression, death), with patients that only achieved stable disease excluded to simplify the problem.

4.2.2 Sample preparation and SMLM imaging

3µm sections were cut from each pre-constructed FFPE TMA block and placed on APES-coated 1.5H coverslips. Coverslips were placed on a hotplate at 70°C for 30 minutes and then inside a pressure cooker with Reveal Decloaker (pH 6.0) for antigen retrieval. At room temperature, samples were then quenched with ammonium chloride for 15 minutes, permeabilised with 0.5% Triton-X in PBS for 1 hour and blocked with 3% BSA plus 20% donkey serum for 1 hour. Samples were then incubated with Roche ready-to-use anti-EREG antibody (SP326: rabbit monoclonal, Roche) overnight at 4°C and then with donkey anti-rabbit Alexa 647 for 1 hour at room temperature. Sample preparation was performed up to this point by Hayley Slaney. I performed all subsequent imaging, data processing, and analysis.

Each section contained samples from up to ~30 patients, with three tumour cores (pre-selected from the region of highest tumour) per patient. For each patient included in this study, the core with the highest tumour content (greatest area of anti-EREG staining in widefield scans) was selected for imaging. One FOV was imaged at the region of greatest staining within the core. Alongside this, one FOV was imaged at the centre of each core (no selection bias) and reserved for later testing (Section 4.2.13).

The samples were imaged using dSTORM in TIRF as described in Section 2.2.1, using the same microscope, buffer, and software. Drift correction, filtering and temporal grouping for each FOV were then performed using CODI as described in Section 2.2.2. The imaging procedure was slightly amended from Section 2.2.1; here, 10,000 frames were acquired using the 640 nm laser excitation at 80% power with an exposure time of 30 ms. Further, as imaging was performed in only one

channel, the filtering step described in Section 2.2.2 that removed localisations in the wrong channels during certain frames was no longer required.

The data in its proprietary format was then converted into an Apache Parquet file. For each localisation, the channel, frame number and *xy* coordinates were stored, alongside ten optional photophysical features extracted by NimOS (ONI, UK), the inbuilt software for the NanoImager, from the fluorescence emission events [245]. This included: the mean and variance of the photon density from the background; the mean and variance of the standard deviation of the fitted point spread function in x and y that was fit to the emission event; the mean and variance of the number of photons from the emission event; the variance in the x and y localisation coordinates from the emission event, where the mean and variance for these features were measured from the temporal grouping of the localisations.

4.2.3 Cell annotation, preprocessing and clustering

Cells within FOVs were annotated and their localisations extracted using *locpix* (Chapter 2) [246]. First, the unfiltered localisations were rendered into a histogram to facilitate annotation. Next, I manually annotated the membrane for the visible cells in the FOV, to generate a membrane mask. These cells were identified as those that had low intensity at their centre and high intensity at their membrane cells and were approximately 5-15 μm in diameter. The cells could also be at the edge of the FOV and therefore only partially visible, provided that 50% or more of the cell was visible in the FOV. Partial cells were included, as the size of the dataset would have been considerably reduced otherwise, however, handcrafted features of these partial cells may have been inaccurate, and this may have confused downstream interpretation of the classifiers. The membrane annotations formed closed loops, where the edge

of the FOV is treated as part of the loop for partially visible cells (see Figure 4.1d in Section 4.3 for examples). It was hard to identify cells in some FOVs, which may have introduced bias into the cells selected for this analysis. Further, the TMAs contain multiple cell types that could not be identified from the dSTORM data (e.g. lymphocytes, neutrophils, eosinophils, endothelial cells, fibroblasts, etc.). Therefore, the cells could be of different types, which differs from previous analyses of PICCOLO samples that analyse tumour cells only [25,70], The cell containing region in the FOV was generated by flood-filling the membrane mask at seed locations manually placed at the cell centres. The watershed algorithm was then applied to the membrane mask, over the cell-containing region, to generate the individual cell masks. In other words, this separated out each separate instance of a cell in the FOV. Having first checked these annotations still correctly overlaid the higher quality localisations (with drift correction, filtering, and temporal grouping), the cell masks and membrane annotations were used to extract the higher quality localisations for each cell and label each localisation according to its location (membrane or interior). All localisations not part of a cell were removed. Cells with fewer than 500 localisations, or with fewer than 5 interior or membrane localisations, were removed to ensure sufficient data for analysis of each cell and for calculation of interior and membrane features. The *xy* localisation coordinates for each cell were saved as an Apache Parquet file with a label in the metadata for the treatment response (any-response or no-response).

Each cell was clustered in preparation for feature extraction and graph representation. We trialled *k*-means clustering, with k = 12, 24, 48, 72, 96, 120 or 144, and DBSCAN clustering, with epsilon, ε = 50 nm, 75 nm or 100 nm and

minimum samples, minPts = 3, 5, or 7. These DBSCAN parameters varied around the values used in Chapter 2 (ε = 75 nm, minPts = 5), where DBSCAN revealed differences in the organisation of EREG and EGFR between interior and membrane localisations. This range also covered the values used in a previous study to identify clusters of EGFR in SMLM data [81]. Clusters with two or fewer localisations were discarded to allow calculation of the convex hull and principal components.

4.2.4 Handcrafted feature extraction

Eight handcrafted features: count, radius of gyration squared, perimeter, linearity, planarity, length, area, and density, were calculated for each cluster and cell as described in Section 3.2.3. For the cells, the features could also be calculated separately for the interior and membrane localisations.

4.2.5 *k*-fold dataset generation

The cell dataset was partitioned for 5-fold cross-validation of model performance. The five different splits each contained a training (64%), validation (16%), and test set (20%, non-overlapping between splits). For each split, the training, validation, and test set each had the same proportion of classes as the overall cell dataset, or as close as possible. The cells from a single patient could not be in both the training and test set for a split. The validation set monitors the performance of the model during training.

An alternative method to generate the validation set was also tested (Section 4.3.2). Here, one patient from each class, with approximately equal number of cells and together constituting ~15% of the entire dataset was held out as a validation set. Then 4-fold cross-validation was performed on the remaining ~85% of the data, as there were only four any-response patients after removing the validation set. This

monitors performance during training on patients not used to train the model; however, it biases the model based on this initial patient selection and may not show the model's ability to generalise.

4.2.6 Logistic regression and random forest classification

The cells were classified using logistic regression and random forest (RF) models, using different sets of manual features extracted from the cells and their clusters (Table 4.1). In binary logistic regression, the probability of the positive class is given by

$$\frac{e^{\omega_0 + \omega_1 X_1 + \omega_2 X_2 + \ldots}}{1 + e^{\omega_0 + \omega_1 X_1 + \omega_2 X_2 + \ldots}} \quad (4.1)$$

, where X_i gives the value of each feature and i, and ω_i are the learned parameters of the model [247]. During optimisation of the parameters, a regularisation term can be added, which shrinks the values of ω_i to prevent overfitting [142]. A random forest is a collection of independent decision trees, where each is trained to predict the probability of the positive class [248,249]. The final probability is given by the average probability from the trees. For both logistic regression and random forest models, the final prediction is given by the class with the highest probability.

Clusters were extracted using DBSCAN with ε = 75 nm and minPts = 5, as in Chapter 2, which identified clusters of EGFR. Clusters with two or fewer localisations were dropped. Handcrafted features of the clusters and cells were extracted as detailed in Section 4.2.4. These features may have been unreliable for partial cells at the edge of the FOV; however, these partial cells were not removed, as this would have reduced the size of the dataset considerably. The features were scaled using the standard scaler (using the mean and standard deviation of the training data, see

below) before input to each model. Models using per-cluster features generated a probability of being from the any-response class for each cluster in the cell. These probabilities were averaged, and the cell was classified as any-response if this probability was over 0.5. The ElasticNet regularisation (includes a penalty on the sum of the absolute values of the coefficients, L1 penalty, and a penalty on the sum of the squared values of the coefficients, L2 penalty) was used for logistic regression, as it tends to group highly correlated variables, which may be included here [250,251].

Table 4.1. Handcrafted feature sets used for logistic regression and random forest classification. x indicates the feature was included as part of the set.

	Feature set				
Features	Per-cell count features	Per-cell count features separated into interior and membrane	Per-cell size and shape features	Per-cluster count features	Per-cluster size and shape features
Localisations per cell per area	x				
Clusters per cell per area	x				
Interior localisations per cell per area		x			
Membrane localisations per cell per membrane perimeter		x			
Interior clusters per cell per area		x			
Membrane clusters per cell per membrane perimeter		x			
Cell radius of gyration squared			x		
Cell linearity			x		
Cell planarity			x		
Cell length			x		
Cell area			x		
Cell perimeter			x		
Localisation per cluster				x	
Cluster length					x
Cluster area					x
Cluster radius of gyration squared					x
Cluster linearity					x
Cluster planarity					x
Cluster perimeter					x

For each approach, the model hyperparameters were optimised by calculating the average performance over the validation sets for the different splits. For logistic regression, there were three hyperparameters: the regularisation strength (*C*), where smaller values give higher regularisation and vice versa; maximum iterations to converge on a solution (max_iter); and L1 ratio, which gives the ratio of L1 to L2 penalty. For random forest, there were four hyperparameters: the number of trees for the forest (n_estimators); the maximum depth of each tree (max_depth); the maximum number of features to consider at each split (max_features); and the maximum number of leaf nodes for each tree (max_leaf_nodes). For logistic regression, the values swept over were *C* = [0.1, 1, 10, 100], L1 ratio = [0.25, 0.5,

0.75], and max_iter = 5000. For random forest, the values swept over were n_estimators = [50,100,200,300], max_depth = [3,5,10], max_leaf_nodes = [3,5,10] and max_features = [2, 4, …, up to the total number of features in the feature set]. For each combination of hyperparameters, the model was fit to the training sets and evaluated on the validation sets. The area under the receiver operator curve (AUROC) was measured and averaged over the validation sets for the different splits to determine the optimal hyperparameters. The balanced accuracy was also calculated and presented but was not used to optimise the hyperparameters.

The performance of each approach was compared by evaluating on the test sets. Using the best hyperparameters from model tuning, each model was fit to the training and validation set and evaluated on the test set for each split. Model performance was then compared using the AUROC and balanced accuracy averaged across the test sets [235].

The most important features for classification were determined by inspecting the models. These were given by the features with the largest coefficients (absolute value) for logistic regression and the highest Gini importance for RF, despite the limitations of these approaches, particularly for highly correlated variables [251-253]. Gini importance calculates the importance of each feature, by measuring how much each feature changes the impurity (how mixed up the data is by class) at each node in the tree [254]. More important features have a greater decrease in impurity (better separate the classes), summing over each node in the tree they are involved in and averaging over all of the trees in the forest [254]. Further exploration of feature importance was undertaken using SHapley Additive exPlanation (SHAP) values implemented via the SHAP Python package [255]. For a given model and data point (here, a cell), the SHAP

value for each feature indicates the magnitude and direction of its impact on the model's prediction [255,256]. The contribution of each feature is given by the difference between the model's prediction with the feature present and when it is absent (random forest) or when it is replaced by the average value of that feature (logistic regression) [255,257]. Here, the SHAP values for each cell were calculated when it was in the test set. The SHAP values from each sample in the dataset were then aggregated and visualised to indicate the global impact of each feature on the classification task [258]. More information on Shapley values can be found in Appendix (Section 7.1.3).

4.2.7 MLP classification

Per-cell classification with a multi-layer perceptron (MLP) proceeded similarly to logistic regression and random forest classification. For each split, an MLP was trained on the training set for 200 epochs, saved when the loss was lowest on the validation set and then evaluated on the test set. The model size was optimised by calculating the average performance over the validation sets for the different splits, sweeping over different sizes for the hidden layers = [100, (100,100), (100,100,100) or (100,100,100,100)], where 100 represents one hidden layer with 100 neurons, (100,100) represents two hidden layers both with 100 neurons, and so on. All other hyperparameters were set to their default value from *Scikit-learn* [249]. Results were given for the MLP with optimal model size.

4.2.8 Graph construction

Each cell was represented as a graph using the same method as for the ROIs in Section 3.2.4, but with some modifications. We trialled $\mathcal{N}_{cluster}$ = 3, 5 and 7. We also trialled normalising the cell size over the dataset (per-dataset), as well as the

per-graph (per-cell) basis described in Section 3.2.4. For per-dataset normalisation: $x \rightarrow \frac{2(x-\min(x))}{\max(x_{range}, y_{range})} - \frac{\max(x)-\min(x)}{\max(x_{range}, y_{range})}$ and similarly for *y*, where the ranges (x_{range}, y_{range}) were now measured over the training set (training set for a split), while the minima and maxima ($\min(x)$, $\max(x)$, $\min(y)$, $\max(y)$) were still measured over the given cell. For per-dataset cell size normalisation of the validation and test sets, both *x* and *y* were clamped to between -1 and +1, in case they had cells with a larger x_{range} or y_{range} than the training set. Further, we also tested including the ten photophysical features as features of the localisation nodes.

4.2.9 ClusterNet model architectures

The cells were classified using two different neural network models: *ClusterNet-HCF* (Handcrafted Cluster Features) and *ClusterNet-LCF* (Learned Cluster Features), introduced in Chapter 3. In this chapter, several amendments were made to these models: the number of output channels was changed from 7 to 2; we trialled r_{LocNet} = 0.25, 0.5 and 0.75, and $\mathcal{N}_{LocNet}$ = 3, 5 and 7; and the number of input channels for the *LocNet* was changed to ten when the localisation nodes had features.

We also trialled including dropout for both *LocNet* and *ClusterNet.* For *LocNet*, it was added into the first layer of the final MLP ($dropout_{LocNet}$) and into the position, h_ω, and attention, γ_Θ, MLPs in the PointTransformer convolutions (position and attention dropout controlled by one parameter: $dropout_{LocNetPositionAttention}$). For the *ClusterNet,* it was added into the position, $h_{\boldsymbol{\omega}}$, and attention, $\gamma_{\boldsymbol{\Theta}}$, MLPs in the PointTransformer convolutions ($dropout_{ClusterNet}$).

In *ClusterNet*, we trialled attention-based aggregation of the cluster features into a graph feature, instead of global maximum pooling [259,260]. This layer either focused on

particular nodes (node-level gating) or features (feature-level gating) that it identified as most relevant. For a graph with cluster nodes, *i*, with features, $\boldsymbol{x_i}$, the output feature vector was given by

$$\sum_i \mathrm{softmax}(h_{gate}(\boldsymbol{x_i})) \cdot h_\theta(\boldsymbol{x_i}), \tag{4.2}$$

where h_{gate} and h_θ were linear layers. h_θ had *N* input and output channels, where *N* is the number of output channels from the four message passing layers. For node-level gating, h_{gate} had *N* input and 1 output channels. For feature-level gating, the same linear layer is used for both h_{gate} and h_θ (i.e. $h_{gate} = h_\theta$), with *N* input and output channels [260].

4.2.10 Superclusters

Experiments were also performed with two additional layers of clusters, by identifying clusters of clusters (super-clusters) and clusters of super-clusters (super-super-clusters). Clusters were extracted using *k*-means, with *k* varied. Each cell graph incorporated these as additional nodes, where each cluster node belonged to a super-cluster node and each super-cluster node belonged to a super-super-cluster node. The new nodes had positions given by the centroid of their constituent clusters, calculated from the normalised and scaled positions. Undirected edges connected the nodes to themselves and their nearest five neighbours. They had no input features.

ClusterNet was modified to include these additional clusters. After the fourth message passing layer, the cluster features were mean-pooled into their super-clusters, giving the super-cluster features. These were then inputted to an MLP (with no normalisation, ReLU activation and 40 output channels) and then the

sigmoid activation function. This was followed by three message passing layers as defined in *ClusterNet* (Figure 3.3), with output dimensions of [*k*,48], [*k*,56] and [*k*,64]. This was then similarly repeated for the super-super-cluster features, with output dimensions of [*k*,72], [*k*,80] and [*k*,88] for the three message passing layers. A global maximum pooling layer then aggregated the super-super-cluster features into a feature vector for the graph. This was inputted to a linear layer followed by the log softmax function to give the log probabilities of the graph belonging to each class.

4.2.11 Training procedure

The models were trained using the same training procedure as described in Section 3.2.7 but with a batch size of 8. Additional data augmentation was also trialled via random scaling, shearing, or flipping (in x and in y) of the node positions. Training for 100 epochs on a NVIDIA GeForce RTX 2060 with 6GB RAM took ~5 minutes.

4.2.12 Evaluation procedure

The best model for each split was evaluated on the test set for each split. For evaluation, each graph was run through the model 25 times without random augmentation and the average output was calculated, to account for random sampling of the point cloud in *LocNet*. The graphs were then classified according to the highest probability class from the model. The predictions and probabilities were evaluated using balanced accuracy and AUROC for binary classification [235]. More information on the metrics used to evaluate performance can be found in the Appendix (Section 7.1.1).

The per-cell predictions could then be aggregated into per-patient predictions. Without further retraining, the per-patient probability was calculated from the predictions on the test sets by averaging (arithmetic mean) the per-cell probabilities

for each patient. The patient was predicted as any-response if this probability was over 0.5. Per-patient metrics could not be separately calculated for each test set due to a lack of patients in each test set. Therefore, the patients from different test sets were aggregated, and the per-patient AUROC and balanced accuracy were calculated. 95% confidence intervals were calculated for these metrics using bootstrapping as implemented in Python package MLstatkit [261]. In brief, the model predictions were resampled with replacement, and the metric was calculated. This was repeated 1,000 times. This indicated the variability in the metric and was used to calculate the 95% confidence interval. If the confidence interval crossed 0.5, this may indicate that the model was not better than random guessing. The per-patient AUROC may have been unreliable, as calculating this metric involved setting a threshold over probabilities from different models. This assumed that the probabilities from different models could be directly compared; for example, equal probabilities should represent an equal likelihood of the underlying event. This was unlikely, as the models were not calibrated, which meant that the probability from a model may not correspond to the true likelihood of the event [235].

When combining the predictions from the logistic regression or random forest models with those from the graph neural network models, the probabilities were averaged, and the cells were classified according to the highest probability class, without further retraining. As mentioned above, these models might be calibrated differently. Therefore, while probabilities from the same model for different cells can be compared (relatively), averaging probabilities from different types of model for the same cell may have been unreliable [262]. However, we can compare (relatively) between these combined models, when they average the same types of model.

4.2.13 Reserved test set generation and evaluation

An additional reserved test set of cells was generated from the FOVs that were reserved during the collection of the main dataset (Section 4.2.2). These FOVs came from the same patient tumour samples used to generate the main dataset. Cells were extracted and processed from these FOVs in the same way as for the main dataset. Models were evaluated on the reserved test set using the same procedures as for the main dataset, but with some amendments. For per-cell classification, the best model had one trained instance from each split. Each of these was evaluated on the reserved test set without further retraining (e.g. if there were five splits, this would give five trained instances of the model, which would in turn give five probabilities for each cell in the reserved test set). For per-patient classification, the per-cell probabilities were averaged (arithmetic mean) over the different instances to give a single probability for each cell in the reserved test set. The per-cell probabilities for each patient were then averaged (arithmetic mean) to give a single probability for each patient. Calculation of per-patient AUROC and balanced accuracy then proceeded as for the main dataset for consistency. As mentioned previously, averaging probabilities from different models may have been unreliable, as the models may have been calibrated differently.

4.2.14 Feature analysis

We used Uniform Manifold Approximation and Projection (UMAP) to visualise the per-cluster and whole-graph features for the reserved test set to explore if they separated the classes, similar to Section 3.2.10. The handcrafted per-cluster features, the per-cluster features after the fourth message passing layer of *ClusterNet*, and the whole-graph features aggregated from *ClusterNet* were visualised. The trained instance of the model from the split from *k*-fold

cross-validation, which had the best performance on the reserved test set, was used to generate the features. Features were generated by running each graph through the model 25 times and taking the average. The features were normalised by subtracting the mean and dividing by the variance of each feature independently, measured over the training and validation sets for the split. We used 20 neighbours for each feature vector and 0.5 minimum distance in the lower-dimensional space, as described in Section 3.2.10.

4.2.15 Immunohistochemistry (IHC) dataset

Previous studies on the PICCOLO dataset have shown that high EREG expression predicts response to anti-EGFR treatment with panitumumab [25,70]. However, as not all of these patients responded, there is an opportunity to improve this by successfully identifying patients with high EREG expression that do not respond to treatment. Therefore, previously acquired immunohistochemistry (IHC) data of EREG expression for patients in the PICCOLO trial was used to refine patient selection, to focus on the high EREG expression patients [25]. The IHC dataset was described in detail by the original study [25], but an overview is provided here. Briefly, 4 μm sections were cut from FFPE blocks of whole tissue samples (not tissue microarrays) and placed on Superfrost Plus slides. The sections were stained with anti-EREG (SP326; Roche Tissue Diagnostics) rabbit monoclonal antibody and imaged using a VENTANA DP 200 scanner at 200 × magnification, generating whole slide images. Within the tumour areas (as identified by pathologists) of each slide, image-analysis algorithms identified the cells, classified them as tumour or non-tumour, and classified them as positive or negative for EREG. From this, the percentage of cells within the tumour area that were positively stained for EREG was calculated (%pos). In this study, as per the original study [25], %pos was used to

define high EREG expression (%pos > 50%) and low EREG expression (%pos ≤ 50%) for each patient. Several patients did not have data from IHC. For these patients, the average number of localisations per cell, as measured by dSTORM, was compared to the other patients in the dataset to decide whether they should be included or excluded from the high EREG expression group.

4.3 Results

4.3.1 Per-cell classification of the *k*-fold dataset using traditional ML models

A dataset of EREG localisations in cells in tissue was collected from advanced colorectal cancer patients (PICCOLO trial), who received anti-EGFR treatment and showed either no-response (best RECIST response rate: radiological progression, clinical progression or death) or any-response (best RECIST response rate: complete response or partial response) [26,243]. Tissue microarray samples, derived from formalin-fixed paraffin-embedded (FFPE) blocks, were stained for EREG and the regions of greatest staining were imaged using dSTORM. This gave 173 cells (no-response: 121 cells, any-response: 52 cells) from 23 patients. Filtering the cells by the number of localisations (localisations: per cell ≥ 500, at the interior ≥ 5, at the membrane ≥ 5) removed ten cells (Figure 4.2). This gave the filtered dataset, which contained 163 cells from 23 patients in which the number of cells per patient was unequal, ranging from one to thirty-one cells per patient (Table 4.2, Table 4.3, Figure 4.3).

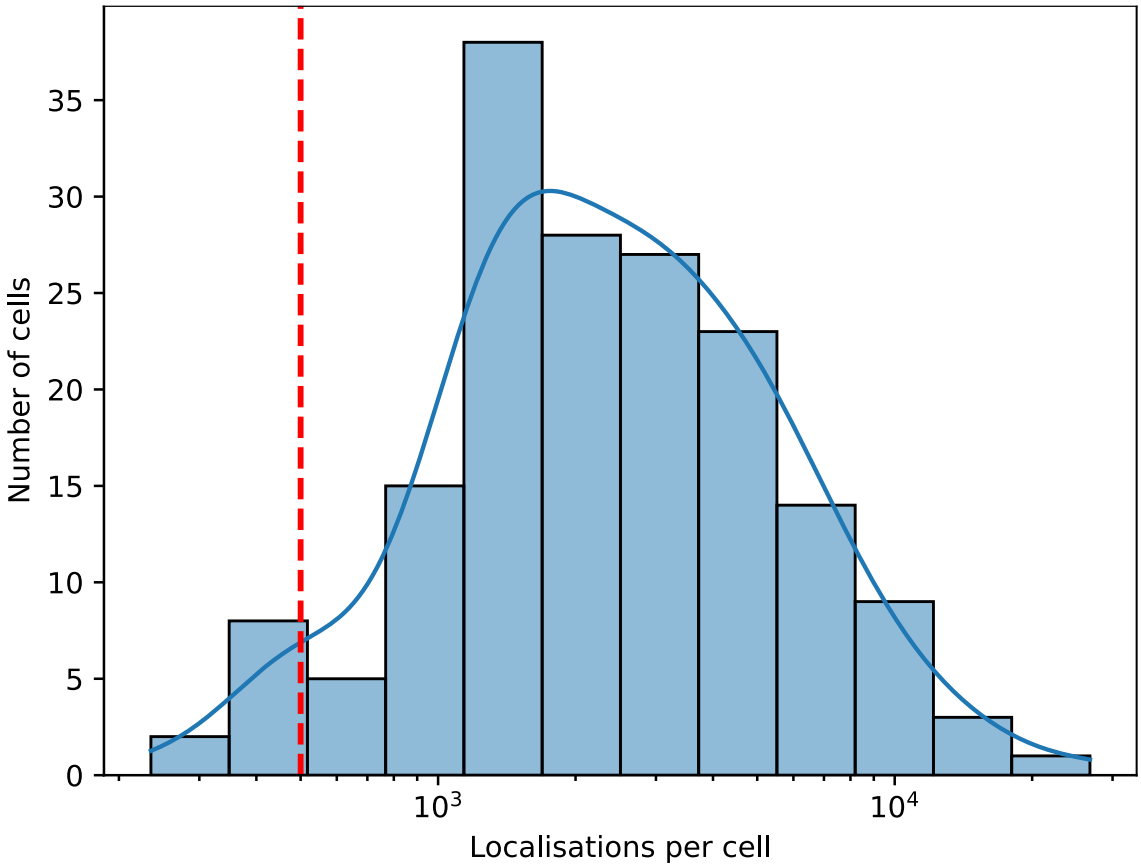

Figure 4.2. Localisations per cell for the *k*-fold dataset before filtering. Dashed red line is the 500 localisations per cell threshold. Solid blue line is the kernel density estimate. X-axis is a log scale.

Table 4.2. Number of cells and patients in the filtered and unfiltered *k*-fold datasets. Number of no-response vs any-response cells and patients given in brackets (no- vs. any- response).

Dataset	Cells (no- vs. any-response)	Patients (no- vs. any-response)
Unfiltered	173 (121 vs. 52)	23 (18 vs. 5)
Filtered	163 (117 vs. 46)	23 (18 vs. 5)

Table 4.3. Characterising the cell dataset with or without filtering by localisation parameters and counts per cell. σ: localisation uncertainty (mean of uncertainty in x and y).

Class	Min localisations per cell	Max. localisations per cell	Mean localisations per cell	Min. σ [nm]	Max. σ [nm]	Mean σ [nm]
No-response unfiltered (n = 121)	2,211	80,753	13,197	0.1	275.7	25.7
Any-response unfiltered (n = 52)	1,933	40,439	8,451	0.2	304.5	24.6
No-response filtered (n = 117)	508	26,767	3,820	0.6	25.0	9.8
Any-response filtered (n = 46)	516	15,304	2,670	0.7	24.9	10.3

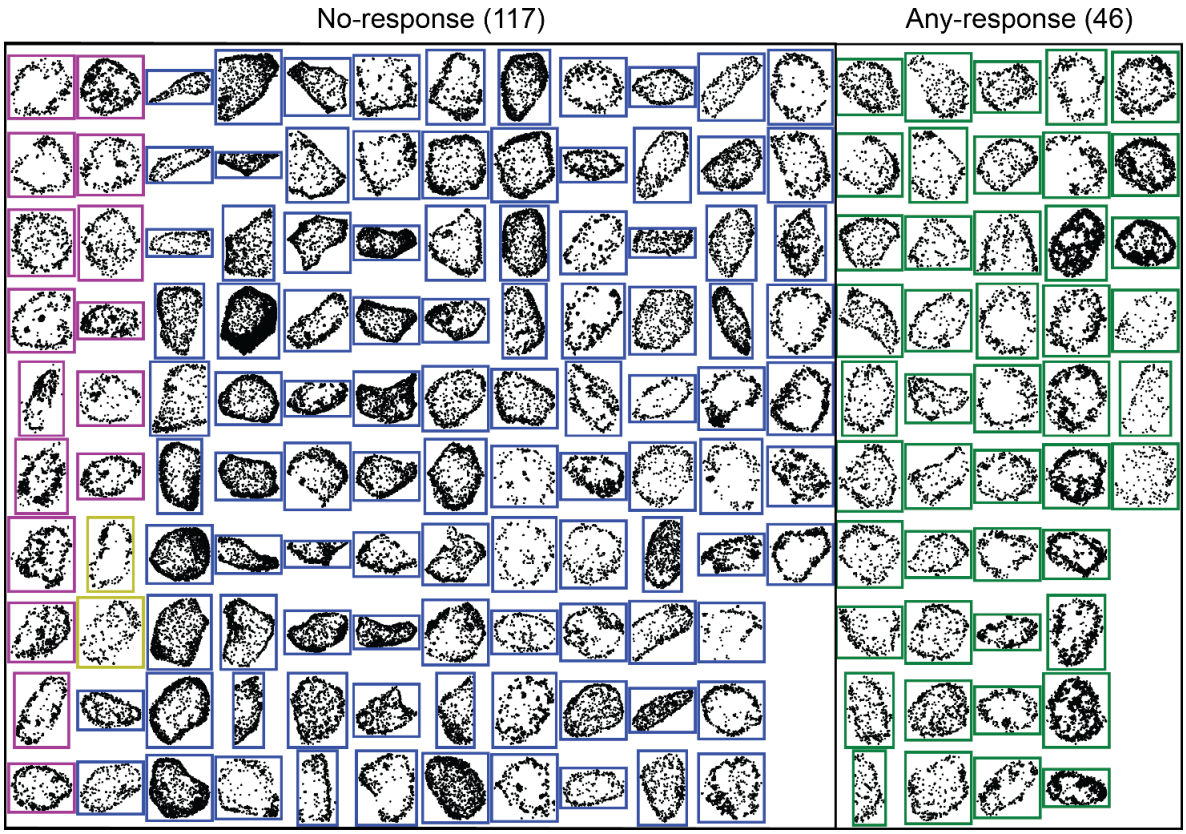

Figure 4.3. EREG localisations for each cell in the filtered *k*-fold dataset. Borders coloured by response to anti-EGFR treatment: death (pink), clinical progression (yellow), radiological progression (blue), and partial response (green). Each cell is rescaled independently to approximately the same size for the plot, therefore, cell sizes should not be compared. Number in brackets indicates the number of cells for that class.

The cells from the filtered *k*-fold dataset were classified into no-response and any-response using traditional ML classification models, logistic regression, and random forest (RF). These models classified the cells using different sets of handcrafted features for the cells and their clusters (Table 4.1). This included per-cell count features, per-cell count features separated into interior and membrane, per-cell size and shape features, per-cluster count features or per-cluster size and shape features. Models using per-cell size and shape features achieved the best results, as measured by area under the receiver operator curve (AUROC) (Table 4.4). RF outperformed logistic regression for all feature sets, apart from for per-cell count, size, and shape features, as measured by AUROC (Table 4.4). The balanced

accuracy score was lower than the AUROC for both models, potentially indicating that the models were poorly calibrated [235].

Table 4.4. Per-cell classification of the filtered *k*-fold dataset using traditional ML models. Performance metric scores are presented as the mean ± standard deviation over the test sets for the logistic regression (Log) and random forest (RF) models trained with different features of the cells and their clusters (Table 4.1). For both models, the best score for each metric is highlighted in bold.

Model	Feature set	AUROC	Balanced accuracy
Log	Per-cell count features	0.28 ± 0.26	0.49 ± 0.03
	Per-cell count features separated into interior and membrane	0.19 ± 0.15	0.47 ± 0.03
	Per-cell size and shape features	**0.66 ± 0.19**	0.46 ± 0.06
	Per-cell count, size, and shape features	0.59 ± 0.21	0.48 ± 0.04
	Per-cluster count features	0.45 ± 0.07	**0.5 ± 0.0**
	Per-cluster size and shape features	0.41 ± 0.07	**0.5 ± 0.0**
RF	Per-cell count features	0.28 ± 0.20	0.43 ± 0.04
	Per-cell count features separated into interior and membrane	0.28 ± 0.23	0.46 ± 0.06
	Per-cell size and shape features	**0.77 ± 0.12**	**0.60 ± 0.18**
	Per-cell count, size, and shape features	0.70 ± 0.16	0.59 ± 0.20
	Per-cluster count features	0.42 ± 0.04	0.5 ± 0.0
	Per-cluster size and shape features	0.47 ± 0.08	0.5 ± 0.0

Models using the other feature sets did not discriminate between the classes (Table 4.4). This included per-cell count features, even after separation into interior and membrane (Table 4.4). Combining the successful per-cell size and shape features with per-cell count features reduced the AUROC compared to size and shape features alone (Table 4.4). Aggregating per-cluster classifications into a cell classification also had poor accuracy and AUROC (Table 4.4). As an extension, a multi-layer perceptron using per-cell size and shape features (the best performing feature set) was tested but failed to improve performance over the RF (test sets results, AUROC: 0.69 ± 0.12, balanced accuracy: 0.54 ± 0.05).

In summary, traditional ML models using per-cell size and shape features were the best at classifying cells by response to anti-EGFR treatment, as measured by AUROC. This indicated that the size and shape of these cells may be associated with the response to treatment. This could be because of increased ploidy (number

of chromosomes), which has been associated with tumour progression, more aggressive disease and poorer prognosis for cancer, and can cause the cells to become larger [263-265]. Or, the observed differences in shape and size may be reflecting different populations of cell types (e.g. lymphocytes, neutrophils, eosinophils, endothelial cells, fibroblasts, etc.), which could not be differentiated from the dSTORM data, and has been associated with response to treatment [266]. On the other hand, using per-cell count features gave worse classification performance, despite previous analysis of tumour samples from the PICCOLO trial showing that responders had elevated EREG expression (as measured by IHC) [25,70]. This may be because it is unreliable to measure absolute protein expression from SMLM data without careful calibration, which includes determining the expected number of protein molecules per localisation [135]. Further, using per-cluster features was no better than randomly guessing, despite increased clustering of EGFR being associated with increased downstream signalling [38,40-42,241,242]. This motivated investigation into whether the larger-scale structure (supra-cluster structure) can separate the classes instead.

4.3.2 Per-cell classification of the filtered *k*-fold dataset using graph neural networks

The cells from the filtered *k*-fold dataset were classified into no-response and any-response using the graph neural-networks *ClusterNet-HCF* and *ClusterNet-LCF* introduced in Chapter 3. These models were first tested using the same graph construction (*k*-means, $k = 12$, per-cell normalisation), training procedure (including the same data augmentations), model architecture, and evaluation procedure as for the digits and letters dataset. The only initial amendments included changing the output channels from 7 to 2, the batch size from 128 to 8 and the evaluation

procedure to take the average of 25 runs (Materials and Methods). *ClusterNet-HCF* had higher accuracy and AUROC than *ClusterNet-LCF*, but both models performed poorly (Table 4.5A).

Table 4.5. Per-cell graph neural network classification of the filtered *k*-fold dataset. Performance metric scores are presented as the mean ± standard deviation over the test sets for *ClusterNet-HCF* and *-LCF* models. (Iteration A) Models have their original architectures from Chapter 3. (Iterations B-K) Changes to the pooling operations (B), etc. relative to iteration A, with multiple variants tested. For both models, the best score for each metric is highlighted in bold. Details for each variant are noted in brackets when first introduced. *CN-HCF*: *ClusterNet-HCF*. *CN-LCF*: *ClusterNet-LCF*. BA: Balanced accuracy.

Iteration: Change	Model	Variants tested	AUROC	BA
A: n/a	*CN-HCF*	n/a	0.60 ± 0.13	0.55 ± 0.12
	CN-LCF	n/a	0.45 ± 0.27	0.44 ± 0.14
B: Pooling operations in PointTransformer convolution and to aggregate localisations	*CN-LCF*	B.1 (sum + global mean)	0.36 ± 0.27	0.49 ± 0.13
	CN-LCF	B.2 (sum + global max)	0.41 ± 0.22	0.46 ± 0.11
	CN-LCF	B.3 (max + global mean)	0.53 ± 0.20	0.55 ± 0.09
C: Per-dataset normalisation	*CN-HCF*	C	0.54 ± 0.19	0.53 ± 0.16
	CN-LCF	C + B.3	0.50 ± 0.25	0.49 ± 0.21
D: Validation set generation using the alternative method	*CN-HCF*	D	0.55 ± 0.17	0.48 ± 0.14
	CN-LCF	D + B.3	0.58 ± 0.22	0.56 ± 0.13
E: Additional data augmentation	*CN-HCF*	E	0.52 ± 0.20	0.52 ± 0.20
	CN-LCF	E + B.3	0.44 ± 0.25	0.47 ± 0.16
F: Model architecture	*CN-HCF*	F.1 (feature-level attention-based aggregation of cluster features in *ClusterNet*)	0.47 ± 0.25	0.48 ± 0.17
	CN-HCF	F.2 (node-level attention-based aggregation of cluster features in *ClusterNet*)	0.50 ± 0.20	0.49 ± 0.16
	CN-HCF	F.3 (include dropout)	0.56 ± 0.23	0.49 ± 0.15
	CN-HCF	F.4 (increase size of hidden layers)	0.53 ± 0.27	0.54 ± 0.15
	CN-LCF	F.1 + B.3	0.51 ± 0.20	0.48 ± 0.08
	CN-LCF	F.2 + B.3	0.49 ± 0.19	0.48 ± 0.16
	CN-LCF	F.3 + B.3	0.58 ± 0.26	0.50 ± 0.09
	CN-LCF	F.4 + B.3	0.48 ± 0.23	0.47 ± 0.16
	CN-LCF	F.5 ($\mathcal{N}_{LocNet}$ = 3) + B.3	0.48 ± 0.18	0.47 ± 0.12
	CN-LCF	F.6 ($\mathcal{N}_{LocNet}$ = 7) + B.3	0.35 ± 0.28	0.45 ± 0.16
	CN-LCF	F.7 (r_{LocNet} = 0.25) + B.3	0.44 ± 0.18	0.45 ± 0.09
	CN-LCF	F.8 (r_{LocNet} = 0.75) + B.3	0.30 ± 0.21	0.42 ± 0.15
G: Number of clusters in *k*-means	*CN-HCF*	G.1 (*k* = 24)	0.51 ± 0.08	0.40 ± 0.08
	CN-HCF	G.2 (*k* = 48)	0.65 ± 0.20	0.57 ± 0.12
	CN-HCF	G.3 (*k* = 72)	0.66 ± 0.17	0.61 ± 0.12
	CN-HCF	G.4 (*k* = 96)	**0.70 ± 0.15**	**0.64 ± 0.14**
	CN-HCF	G.5 (*k* = 120)	0.66 ± 0.18	0.58 ± 0.14
	CN-HCF	G.6 (*k* = 144)	0.68 ± 0.21	0.63 ± 0.16
	CN-LCF	G.1 + B.3	0.48 ± 0.27	0.49 ± 0.16
	CN-LCF	G.2 + B.3	0.48 ± 0.29	0.54 ± 0.23
	CN-LCF	G.3 + B.3	0.59 ± 0.34	0.61 ± 0.25
	CN-LCF	G.4 + B.3	0.50 ± 0.24	0.55 ± 0.14
	CN-LCF	G.5 + B.3	0.58 ± 0.30	0.60 ± 0.17
	CN-LCF	G.6 + B.3	0.57 ± 0.30	0.59 ± 0.24
H: DBSCAN	*CN-HCF*	H.1 (ε = 75 nm, minPts = 5)	0.45 ± 0.27	0.49 ± 0.13
	CN-HCF	H.1 + C	0.35 ± 0.27	0.42 ± 0.12
	CN-LCF	H.1 + B.3	0.47 ± 0.27	0.45 ± 0.18
	CN-LCF	H.1 + C + B.3	0.36 ± 0.27	0.40 ± 0.24
	CN-HCF	H.2 (ε = 50 nm, minPts = 3)	0.70 ± 0.20	0.63 ± 0.19
	CN-HCF	H.3 (ε = 50 nm, minPts = 5)	0.44 ± 0.25	0.51 ± 0.14
	CN-HCF	H.4 (ε = 50 nm, minPts = 7)	0.35 ± 0.25	0.42 ± 0.19
	CN-HCF	H.5 (ε = 75 nm, minPts = 3)	0.65 ± 0.17	0.65 ± 0.18
	CN-HCF	H.6 (ε = 75 nm, minPts = 7)	0.41 ± 0.32	0.52 ± 0.18
	CN-HCF	H.7 (ε = 100 nm, minPts = 3)	0.60 ± 0.21	0.59 ± 0.14
	CN-HCF	H.8 (ε = 100 nm, minPts = 5)	0.63 ± 0.13	0.60 ± 0.11
	CN-HCF	H.9 (ε = 100 nm, minPts = 7)	0.48 ± 0.32	0.52 ± 0.17
I: Number of nearest neighbours	*CN-HCF*	I.1 ($\mathcal{N}_{cluster}$ = 3) + G.4	0.66 ± 0.13	0.62 ± 0.11
	CN-HCF	I.2 ($\mathcal{N}_{cluster}$ = 7) + G.4	0.64 ± 0.16	0.53 ± 0.12
	CN-LCF	I.1 + G.4 + B.3	0.50 ± 0.22	0.53 ± 0.15
	CN-LCF	I.2 + G.4 + B.3	0.55 ± 0.30	0.56 ± 0.20
J: Superclusters	*CN-HCF*	J + G.4	0.68 ± 0.18	**0.64 ± 0.14**
	CN-LCF	J + G.4 + B.3	0.55 ± 0.30	0.57 ± 0.21
K: Best *ClusterNet-LCF*	*CN-LCF*	J + I.2 + G.3 + F.4 + D + B.3	0.58 ± 0.35	0.52 ± 0.19
	CN-LCF	J + I.2 + G.3 + D + B.3	**0.62 ± 0.34**	**0.65 ± 0.14**

Incremental changes were made to both models to improve performance, as measured by AUROC (Table 4.5B-K). Changes that affected both models were only kept if they improved the AUROC of both, rather than being implemented for only the improved model, unless these changes gave the highest overall AUROC. This kept the models similar to each other, allowing for a more direct comparison between them. Graphs may differ slightly between runs as *k*-means clustering is non-deterministic.

In *LocNet* in Chapter 3, maximum pooling was used in the PointTransformer convolution and global maximum pooling was used to aggregate localisations into a per-cluster feature, instead of sum and global mean as in the original architecture, to be more robust to outliers [161]. To test whether this improved the AUROC, the original architecture was trialled (Table 4.5B.1). Two further amendments were also trialled: sum pooling in the PointTransformer convolution and global maximum pooling to aggregate localisations into a per-cluster feature (Table 4.5B.2) and max pooling in the PointTransformer convolution and global mean pooling to aggregate localisations into a per-cluster feature (Table 4.5B.3). Keeping maximum pooling in the PointTransformer convolution but changing back to global mean pooling gave the highest AUROC for *ClusterNet-LCF* and was incorporated into future iterations (Table 4.5B.3).

Further modifications to normalisation, validation set generation and model architecture were unsuccessful. Normalising per-dataset (as opposed to per-cell), allowing the model to account for the size of each cell, did not improve the AUROC (Table 4.5C). Generating the validation set using the alternative method and therefore, validating the model performance on patients not used to train the model,

only improved the AUROC for *ClusterNet-LCF* (Table 4.5D). Including additional random augmentation during training (scaling up to 5%, shearing up to 5%, x and y flips), to reduce the chances of overfitting to the training dataset, did not improve the AUROC (Table 4.5E). None of the following changes to the model architecture improved the AUROC either (except for including dropout for *ClusterNet-LCF*): including feature-level attention-based aggregation of cluster features in *ClusterNet,* under the hypothesis that some of the cluster features may be more relevant than others (Table 4.5F.1); including node-level attention-based aggregation of cluster features in *ClusterNet,* under the hypothesis that some of the clusters may be more relevant than others (Table 4.5F.2); including dropout ($dropout_{LocNet}$ = $dropout_{LocNetPositionAttention}$ = $dropout_{ClusterNet}$ = 0.2) to reduce the chances of overfitting to the training dataset (Table 4.5F.3); increasing the size of the hidden layers, in case the model was not large enough to capture the complexity of the dataset (Table 4.5F.4); setting $\mathcal{N}_{LocNet}$ = 3 (Table 4.5F.5) or $\mathcal{N}_{LocNet}$ = 7 (Table 4.5F.6) to reduce or increase respectively the size of the neighbourhood around each localisation that *LocNet* considered; and setting r_{LocNet} = 0.25 (Table 4.5F.7) or r_{LocNet} = 0.75 (Table 4.5F.8) to reduce or increase respectively the number of localisations that *LocNet* sampled from each cluster.

Increasing the number of clusters used to construct the graph, *k* (*k*-means clustering), improved the AUROC. To determine which values of *k* to trial, each cell in the dataset was clustered using DBSCAN (ε = 75 nm, minPts = 5, same as for logistic regression & random forest classification). This gave a minimum of 9 clusters per cell, a maximum of 544, an arithmetic mean of 109 and a median of 85. Therefore, *k* = 24 (Table 4.5G.1), *k* = 48 (Table 4.5G.2), *k* = 72 (Table 4.5G.3), *k* =

96 (Table 4.5G.4), *k* = 120 (Table 4.5G.5) and *k* = 144 (Table 4.5G.6) were trialled, extending to and just beyond the mean and median number of clusters, with *k* = 96 giving the highest overall AUROC. For further iterations, *k* = 96 was used for both models, to allow for direct comparison between them, despite *k* = 72 giving the highest AUROC for *ClusterNet-LCF*.

Clustering algorithms that account for the variation in the number of clusters per cell may improve over *k*-means clustering, which sets a fixed maximum number of clusters per cell. Therefore, DBSCAN was trialled to see if extracting clusters based on the density of the localisations could improve the AUROC. Using DBSCAN with the same parameters as for the traditional ML classification (ε = 75 nm and minPts = 5) did not improve the AUROC (Table 4.5H.1). This included trialling with per-dataset normalisation to match the input used by the logistic regression and random forest models. A wider range of parameters for DBSCAN were trialled for the better performing network, *ClusterNet-HCF*: ε = 50 nm, minPts = 3 (4H.2), ε = 50 nm, minPts = 5 (4H.3), ε = 50 nm, minPts = 7 (4H.4), ε = 75 nm, minPts = 3 (4H.5), ε = 75 nm, minPts = 7 (4H.6), ε = 100 nm, minPts = 3 (4H.7), ε = 100 nm, minPts = 5 (4H.8), ε = 100 nm, minPts = 7 (4H.9). DBSCAN, with ε = 50 nm and minPts = 3, achieved the same AUROC as *k*-means with optimal number of clusters (*k* = 96), but with greater variance and a decreased balanced accuracy, and therefore was not chosen for future iterations. Furthermore, small changes in the parameters caused large changes to the results, which made it hard to identify the optimal parameters. For example, for ε = 50 nm, changing from minPts = 3 to minPts = 5 led to a drop in AUROC of 0.26. Therefore, using a fixed maximum number of clusters per cell

(*k*-means) made it easier to learn across cells that showed a large range in the number of clusters per cell and was more robust to changes in the parameters.

Additional iterations were unable to improve the overall AUROC. Changing the number of nearest neighbours each cluster was connected to, $\mathcal{N}_{cluster}$, to 3 (Table 4.5I.1) and 7 (Table 4.5I.2) to reduce or increase, respectively, the size of the neighbourhood around each cluster that *ClusterNet* considered, did not improve the AUROC, except for $\mathcal{N}_{cluster}$ = 7 for *ClusterNet-LCF*. Including superclusters (*k* = 44 super-clusters and *k* = 12 super-super-clusters) to try to better capture the larger scale structures in the cell, only improved the AUROC for *ClusterNet-LCF* (Table 4.5J). For the best-performing model (Table 4.5G.4), increasing the number of epochs to 250 and using the alternative method for validation set generation did not improve the AUROC (results not shown). Simultaneously implementing all the changes (except for dropout) that improved *ClusterNet-LCF* performance gave the highest AUROC for *ClusterNet-LCF,* but not the highest overall (Table 4.5K). Here, *k* = 42 rather than *k* = 44 super-clusters were used as one cell had only 42 clusters.

In summary, *ClusterNet-HCF* with *k* = 96 clusters (*k*-means clustering) was the best graph-neural network at classifying cells by response to anti-EGFR treatment, as measured by AUROC. The graph neural network models did not improve on the traditional ML models (lower AUROC but higher balanced accuracy), despite the theoretical ability of deep learning models to capture more complex patterns in the data. Further, as seen for the Digits and Letters dataset in Chapter 3, *ClusterNet-HCF,* which had handcrafted cluster features, outperformed *ClusterNet-LCF*, which had learnt cluster features. This could indicate that there was no more to learn beyond the handcrafted features, or that the *LocNet* architecture

was sub-optimal. Changes to the model architecture and graph representation were tested, but most did not improve performance. This included allowing the model to account for the size of each cell, contradicting the results from the traditional ML models, which performed best using per-cell size and shape features. Only changing the number of clusters improved the performance of *ClusterNet-HCF*, showing the importance of the graph representation for classification. Further, despite the differences in clustering between cells, models that had a fixed number of clusters per cell (*k*-means) outperformed the models which accounted for these differences (DBSCAN). This may indicate that there were other downsides to our approach that used DBSCAN. For example, using a fixed number of message passing layers for every cell may sub-optimally aggregate information over cells with different numbers of clusters.

4.3.3 Per-cell classification of the unfiltered *k*-fold dataset

To determine the effect of the filtering applied during preprocessing, the dataset was generated again without filtering of localisations or removing cells with too few localisations. This gave 173 cells (no-response: 121 cells, any-response: 52 cells) from the same 23 patients, ten more cells than before (no-response: +4, any-response: +6). The training, validation and test splits were generated using the same method but were different due to the inclusion of the additional cells.

The cells were classified using the same traditional ML classification models: logistic regression and random forest. Models using per-cell size and shape features achieved the highest AUROC (Table 4.6), but did not improve upon the models that used filtered data (Table 4.4). Models using per-cell count features (total and separated into interior and membrane) improved the AUROC compared to the

equivalent results on the filtered data, but not enough to improve over random guessing (Table 4.6).

Table 4.6. Per-cell classification of the unfiltered *k*-fold dataset using traditional ML models. Performance metric scores are presented as the mean ± standard deviation over the test sets for the logistic regression (Log) and random forest (RF) models trained with different features of the cells and their clusters (Table 4.1). Arrows indicate an increase (↑), decrease (↓), or no change (−) in the result compared to per-cell classification on the filtered dataset using traditional ML models (Table 4.4). For both models, the best score for each metric is highlighted in bold.

Model	Feature set	AUROC	Balanced accuracy
Log	Per-cell count features	0.24 ± 0.16 (↓)	0.5 ± 0.0 (↑)
	Per-cell count features separated into interior and membrane	0.56 ± 0.30 (↑)	0.46 ± 0.05 (↓)
	Per-cell size and shape features	**0.64 ± 0.15 (↓)**	**0.51 ± 0.02 (↑)**
	Per-cell count, size, and shape features	0.47 ± 0.26 (↓)	0.48 ± 0.05 (−)
	Per-cluster count features	0.49 ± 0.07 (↑)	0.5 ± 0.0 (−)
	Per-cluster size and shape features	0.48 ± 0.05 (↑)	0.5 ± 0.0 (−)
RF	Per-cell count features	0.34 ± 0.30 (↑)	0.48 ± 0.08 (↑)
	Per-cell count features separated into interior and membrane	0.41 ± 0.20 (↑)	0.44 ± 0.06 (↓)
	Per-cell size and shape features	**0.61 ± 0.16 (↓)**	0.54 ± 0.06 (↓)
	Per-cell count, size, and shape features	0.59 ± 0.29 (↓)	**0.63 ± 0.16 (↑)**
	Per-cluster count features	0.46 ± 0.03 (↑)	0.5 ± 0.0 (−)
	Per-cluster size and shape features	0.51 ± 0.04 (↑)	0.5 ± 0.0 (−)

The best *ClusterNet-HCF* and *-LCF* models from the results using the filtered data (*ClusterNet-HCF*: Table 4.5G.4, *ClusterNet-LCF*: Table 4.5K, J+I.2+G.3+D+B.3) were re-trained and evaluated on the unfiltered dataset (Table 4.7). This slightly improved the AUROC for *ClusterNet-HCF* but not for *ClusterNet-LCF*. Further changes to *ClusterNet-LCF*: including the ten photophysical features for each localisation so that the model could learn how to account for these features and therefore the quality of the localisation, rather than filtering out the localisations based on user-defined thresholds on these features; and replacing the first multi-layer perceptron in *LocNet* with node-level gating (Equation 4.2 without the summation, Section 4.2.9), under the hypothesis that some localisations may be more important than others, improved the AUROC. This was the highest AUROC of

any *ClusterNet-LCF* model, with similar performance to the *ClusterNet-HCF* model (Table 4.7). An additional experiment that changed all the aggregations in *LocNet* and *ClusterNet* to mean did not improve the AUROC (results not shown).

Table 4.7. Per-cell graph neural network classification of the unfiltered *k*-fold dataset. Performance metric scores are presented as the mean ± standard deviation over the test sets for the best *ClusterNet-HCF* and *-LCF* models from the filtered data, with some further changes made to *ClusterNet-LCF*. For both models, the best score for each metric is highlighted in bold.

Model	Changes	AUROC	Balanced accuracy
ClusterNet-HCF	None	**0.72 ± 0.14**	**0.63 ± 0.09**
ClusterNet-LCF	None	0.58 ± 0.27	0.53 ± 0.14
	Photophysical features	0.65 ± 0.19	0.52 ± 0.08
	Photophysical features + Node-level gating	**0.70 ± 0.18**	**0.61 ± 0.20**

In summary, removing the filtering steps during preprocessing did not give the highest AUROC overall. This was still given by the random forest with per-cell size and shape features on the filtered dataset (AUROC: 0.77 ± 0.12, Balanced accuracy: 0.60 ± 0.18, Table 4.4). The traditional ML models were not improved, probably because the handcrafted features were now calculated on lower-quality localisations. On the other hand, the AUROC improved for the best graph neural network, *ClusterNet-HCF*, which was surprising as it used handcrafted features of the clusters. *ClusterNet-LCF* had a higher AUROC on the unfiltered rather than filtered data, after including the photophysical features for the localisations and node-level gating. This allowed the model to account for the quality of the localisations and focus on particular localisations in each cell, which could indicate that certain localisations may be more important.

4.3.4 Per-cell classification of the *k*-fold dataset: patients with high EREG expression and combining traditional ML and graph neural network models

Previous studies on the PICCOLO dataset have shown that high EREG expression predicts response to anti-EGFR treatment with panitumumab [25,70]. Using previously acquired immunohistochemistry (IHC) data of whole sections for patients in the PICCOLO trial, the EREG expression for each patient in our dataset was obtained [25]. This showed that any-response patients had significantly higher EREG expression than no-response patients in our dataset, despite being a much smaller subset of the original dataset (Figure 4.4). In the original study, patients with over 50% of tumour cells within the tumour area for a section that was stained positive for EREG (high EREG expression) were predicted to respond to treatment. However, as not all of these patients responded, there is an opportunity to improve this by successfully identifying patients with high EREG expression that do not respond to treatment.

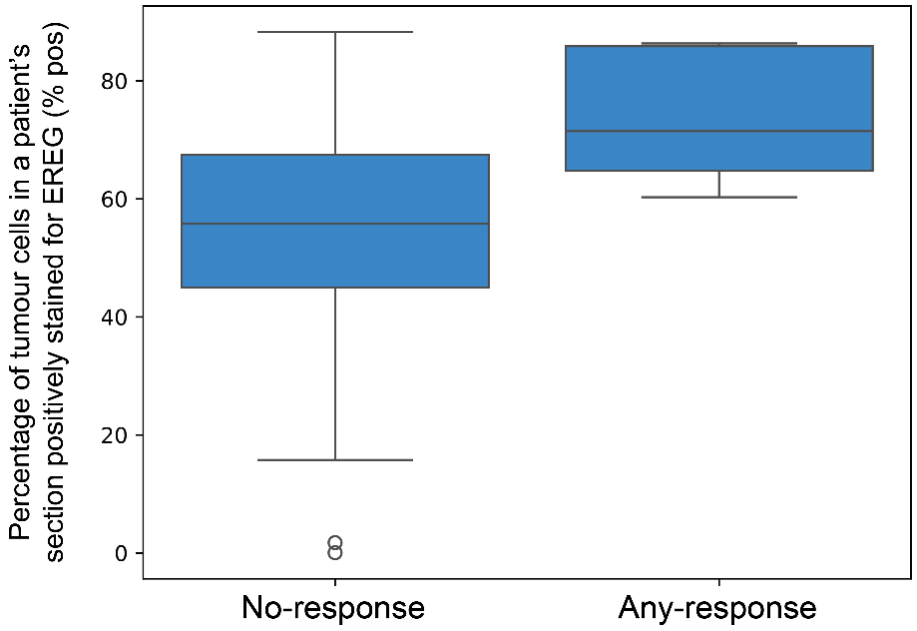

Figure 4.4. Comparing EREG expression for patients in the *k*-fold dataset by response to treatment. Measurements are from previously acquired immunohistochemistry data. Mean %pos: 51.4 (no-response), 73.8 (any-response), $t(16) = -2.6$, $p \leq 0.05$ (two-tailed t-test with unequal variances). Both distributions are normally distributed (Shapiro-Wilk test).

Using the EREG expression measured from IHC, cells from patients with low EREG expression (below 50%) were excluded to focus on patients that were expected to respond. For our dataset, patients with low EREG expression also had low AREG expression (data not shown) and therefore would still be predicted as non-responders by IHC if using AREG expression instead. Two patients with no data from IHC were included in the high EREG expression group, as they had a high average number of EREG localisations per cell (1st and 4th highest EREG localisations per cell of the *k*-fold patients). 17 (filtered dataset) or 18 (unfiltered dataset) cells from four no-response patients were excluded (Table 4.8). The best traditional ML and graph neural network models were retrained and evaluated on the filtered and unfiltered datasets with these cells excluded, regenerating the *k*-fold splits using the same methods. This did not improve the AUROC for the filtered dataset (Table 4.9, Filtered (only high EREG expression)). For the unfiltered dataset, this improved the AUROC for both traditional ML and graph neural network models, giving the highest AUROC for a graph neural network on any dataset (Table 4.9, Unfiltered (only high EREG expression)).

Table 4.8. Number of cells and patients in the filtered and unfiltered *k*-fold datasets, including or excluding patients with low EREG expression. Number of no-response vs any-response cells and patients given in brackets (no- vs. any- response).

Dataset	Cells (no- vs. any-response)	Patients (no- vs. any-response)
Filtered	163 (117 vs. 46)	23 (18 vs. 5)
Unfiltered	173 (121 vs. 52)	23 (18 vs. 5)
Filtered (only high EREG expression)	146 (100 vs. 46)	19 (14 vs. 5)
Unfiltered (only high EREG expression)	155 (103 vs. 52)	19 (14 vs. 5)

Table 4.9. Best per-cell classification of the *k*-fold dataset. Performance metric scores are presented as the mean ± standard deviation over the test sets for the filtered and unfiltered dataset, including or excluding cells from patients with low EREG expression. [a] Random Forest, per-cell size, and shape features. [b] Logistic regression, per-cell size, and shape features. [c] *ClusterNet-HCF*, *k* = 96. The best score for each metric is highlighted in bold.

Dataset	Model	AUROC	Balanced accuracy
Filtered	Traditional ML [a]	0.77 ± 0.12	0.60 ± 0.18
	Graph neural network [c]	0.70 ± 0.15	**0.64 ± 0.14**
	Combined [a+c]	0.77 ± 0.11	**0.64 ± 0.13**
Unfiltered	Traditional ML [b]	0.64 ± 0.15	0.51 ± 0.02
	Graph neural network [c]	0.72 ± 0.14	0.63 ± 0.09
	Combined [b+c]	0.72 ± 0.14	0.61 ± 0.12
Filtered (only high EREG expression)	Traditional ML [a]	0.73 ± 0.24	0.47 ± 0.04
	Graph neural network [c]	0.67 ± 0.31	0.57 ± 0.24
	Combined [a+c]	0.66 ± 0.24	0.61 ± 0.19
Unfiltered (only high EREG expression)	Traditional ML [b]	0.71 ± 0.24	0.58 ± 0.13
	Graph neural network [c]	0.76 ± 0.20	0.63 ± 0.12
	Combined [b+c]	**0.78 ± 0.16**	**0.64 ± 0.11**

Combining the predictions from the best-performing traditional ML and graph neural-network classification models, without further re-training, improved the AUROC in some cases. For both filtered and unfiltered datasets that included cells from patients with low EREG expression, combining the predictions did not improve over the individual models, as measured by either AUROC or balanced accuracy (Table 4.9). For the datasets that excluded cells from patients with low EREG expression, combining the predictions improved the balanced accuracy for the filtered dataset and improved both the AUROC and balanced accuracy for the unfiltered dataset, over the individual models (Table 4.9).

In summary, restricting classification to the patients with high EREG expression improved the performance of the traditional ML and graph neural network models for the unfiltered dataset, but not for the filtered dataset. Further, combining the traditional ML and graph neural network model predictions gave the highest AUROC overall, which was on the unfiltered dataset for the high EREG expression patients.

This showed that these models can learn complementary features of the cells, which can be combined to give better predictions. These results may also indicate a way to improve over previous methods that would incorrectly predict all high EREG expression patients to respond [25,70].

4.3.5 Per-patient classification of the *k*-fold dataset

Next, the response to treatment was predicted at the patient level. The per-cell predictions from the best-performing individual and combined models for each dataset were aggregated into per-patient predictions without further retraining (described in Section 4.2.12). The highest point estimates of AUROC and balanced accuracy were given by the graph neural network and combined model for the unfiltered cell data from patients with high EREG expression (Table 4.10). Further, the combined model on the unfiltered data, with or without high EREG expression patients, was the only model that was better than random guessing, according to the 95% confidence interval for the AUROC. This reflected the per-cell classification results, where the combined model on the unfiltered data from the high EREG expression patients gave the highest AUROC and balanced accuracy (Table 4.9). A similar trend was also seen here as for the per-cell classification: traditional ML model performance was decreased on the unfiltered datasets compared to the filtered datasets, whilst graph neural network performance was increased. Overall, the per-patient results were approximately equal to the per-cell results but with larger variance. In some cases, the per-patient improved over the per-cell results, which could indicate that cells from each patient were heterogeneous and that some cells may be more informative. Therefore, simply averaging the predictions from multiple cells may not have been the optimal way to aggregate the per-cell predictions.

Table 4.10. Per-patient classification of the *k*-fold dataset. Performance metric scores for the patients from the filtered and unfiltered datasets, including or excluding cells from patients with low EREG expression, using the best models from per-cell classification (Table 4.9). [a] Random Forest, per-cell size, and shape features. [b] Logistic regression, per-cell size, and shape features. [c] *ClusterNet-HCF*, *k* = 96. 95% confidence intervals are shown in brackets. Asterisk indicates lower bound of confidence interval is greater than 0.5. The highest point estimate for each metric is highlighted in bold.

Dataset	Model	AUROC	Balanced accuracy
Filtered	Traditional ML [a]	0.72 [0.47 - 0.93]	0.54 [0.38 - 0.77]
	Graph neural network [c]	0.67 [0.36 - 0.92]	0.62 [0.38 - 0.89]
	Combined [a+c]	0.69 [0.41 - 0.95]	0.67 [0.45 - 0.95]
Unfiltered	Traditional ML [b]	0.56 [0.29 - 0.80]	0.50 [0.50 - 0.50]
	Graph neural network [c]	0.70 [0.37 - 0.95]	0.63 [0.34 - 0.88]
	Combined [b+c]	0.74 [0.52 - 0.94]*	0.66 [0.38 - 0.90]
Filtered (only high EREG expression)	Traditional ML [a]	0.74 [0.50 - 0.93]	0.53 [0.37 - 0.75]
	Graph neural network [c]	0.63 [0.21 - 0.96]	0.58 [0.34 - 0.77]
	Combined [a+c]	0.70 [0.39 - 0.94]	0.62 [0.31 - 0.88]
Unfiltered (only high EREG expression)	Traditional ML [b]	0.67 [0.41 - 0.90]	0.43 [0.33 - 0.50]
	Graph neural network [c]	**0.77** [0.50 - 0.98]	**0.69** [0.43 - 0.93]
	Combined [b+c]	**0.77** [0.57 - 0.96]*	**0.69** [0.43 - 0.93]

These results were compared against a previous method that used immunohistochemistry to predict a positive response to treatment for all patients in the PICCOLO trial with high EREG expression [25,70]. Using this approach for the datasets in this chapter gave a balanced accuracy of 0.61 for the datasets that included low and high EREG expression patients, whereas the combined model gave a higher point estimate of the balanced accuracy for the same dataset: 0.66 (unfiltered dataset, Table 4.10). Alternatively, restricting the combined model to patients with high EREG expression gave a per-patient balanced accuracy point estimate of 0.69 (Table 4.10) and a standard accuracy of 0.74 [0.53-0.95]. These are higher than the balanced accuracy given by predicting all these patients as responders: 0.5 (by definition) and the corresponding standard accuracy: 0.26. However, each of our models had a large confidence interval that crossed 0.5 for balanced accuracy, suggesting they may be no better than random guessing (Table 4.10). These results also came from a small number of patients. Therefore, further

work is needed to ensure that these results are repeatable and applicable to other datasets.

In summary, the per-cell predictions were aggregated to give predictions for each patient, the more clinically relevant task. The results were similar to those for the per-cell classification, with the best-performing model and dataset on the per-cell classification achieving the highest results for per-patient classification. Further, the results suggested that this approach could improve over a previous approach that classified patients in the PICCOLO dataset using the EREG expression. However, our results had large confidence intervals and were from a small number of patients.

4.3.6 Per-cell and per-patient classification of the reserved test set

The models were evaluated on the reserved test set to check their performance on unseen cells, mostly from the same patients, but with some cells from patients that were not in the *k*-fold dataset. This dataset was collected alongside the *k*-fold dataset using identical methods, with the only difference that one FOV was imaged at the centre of each core rather than at the region of greatest staining. From this reserved test set, four different datasets were prepared as above, by filtering (or not) for localisations per cell and excluding (or including) cells from patients with low EREG expression (Table 4.11). Similar to the *k*-fold dataset, the number of cells per patient was unequal, ranging from one to seventeen cells per patient. One patient with no data from IHC was included in the high EREG expression group, as they had a high average number of EREG localisations per cell (4th highest EREG localisations per cell for the RTS).

Table 4.11. Number of cells and patients in the filtered and unfiltered reserved test set datasets, including or excluding patients with low EREG expression. Number of no-response vs any-response cells and patients given in brackets (no- vs. any- response). Number of patients not in the equivalent *k*-fold dataset (unseen) is also given.

Dataset	Cells (no- vs. any-response)	Patients (no- vs. any-response)	Unseen patients
Filtered	54 (39 vs. 15)	15 (10 vs. 5)	5
Unfiltered	66 (48 vs. 18)	17 (12 vs. 5)	5
Filtered (only high EREG expression)	42 (30 vs. 12)	10 (7 vs. 3)	1
Unfiltered (only high EREG expression)	50 (37 vs. 13)	11 (8 vs. 3)	1

The best performing individual and combined model for each dataset from *k*-fold training was evaluated on the equivalent dataset from the reserved test set without further retraining (Table 4.12). The highest AUROC and balanced accuracy for per-cell classification was achieved by the combined model for the unfiltered cell data from patients with high EREG expression (Table 4.12). This reflected its high performance at per-cell and per-patient classification during *k*-fold training (Tables 4.9 and 4.10). The highest point estimates of AUROC and balanced accuracy for per-patient classification were achieved by the traditional ML model on the same dataset (unfiltered, only high EREG expression), despite its poorer performance at per-cell classification compared to the graph neural network and combined models. This was the only approach for per-patient classification that was better than random guessing, according to the 95% confidence interval for the AUROC.

Trends observed in the results for the *k*-fold dataset were reinvestigated for the reserved test set. Combining the models only improved the AUROC and balanced accuracy in some cases. In general, the AUROC and balanced accuracy were higher for the unfiltered datasets than the filtered ones. Further, removing patients with low EREG expression improved the AUROC and balanced accuracy for the unfiltered

dataset and the filtered dataset to a lesser extent. This may be in part because there were fewer unseen patients in the high EREG expression datasets (Table 4.11).

Table 4.12. Per-cell and per-patient classification of the reserved test set. Performance metric scores for the patients from the filtered and unfiltered datasets, including or excluding cells from patients with low EREG expression, using the best models from *k*-fold training. Per-cell results are presented as the mean ± standard deviation over the different trained instances of the model from *k*-fold training. Arrows indicate an increase (↑), decrease (↓), or no change (−) in the result compared to per-cell and per-patient classification of the *k*-fold dataset (Tables 4.9 and 4.10). For per-patient results, 95% confidence intervals are shown in brackets. Asterisk indicates lower bound of the confidence interval is greater than 0.5. For both per-cell and per-patient classification, the highest score for each metric is highlighted in bold.

		Per cell		Per patient	
Training dataset	**Model**	**AUROC**	**Balanced accuracy**	**AUROC**	**Balanced accuracy**
Filtered	Traditional ML	0.51 ± 0.06 (↓)	0.50 ± 0.04 (↓)	0.46 [0.00 - 0.86] (↓)	0.50 [0.29 - 0.75] (↓)
	Graph neural network	0.63 ± 0.05 (↓)	0.54 ± 0.07 (↓)	0.38 [0.06 - 0.75] (↓)	0.45 [0.33 - 0.50] (↓)
	Combined	0.60 ± 0.07 (↓)	0.52 ± 0.04 (↓)	0.36 [0.07 - 0.69] (↓)	0.40 [0.27 - 0.50] (↓)
Unfiltered	Traditional ML	0.52 ± 0.05 (↓)	0.53 ± 0.04 (↑)	0.43 [0.13 - 0.79] (↓)	0.50 [0.50 - 0.50] (−)
	Graph neural network	0.69 ± 0.10 (↓)	0.64 ± 0.07 (↑)	0.47 [0.17 - 0.74] (↓)	0.57 [0.31 - 0.77] (↓)
	Combined	0.67 ± 0.09 (↓)	0.65 ± 0.08 (↑)	0.48 [0.17 - 0.77] (↓)	0.35 [0.14 - 0.58] (↓)
Filtered (only high EREG expression)	Traditional ML	0.62 ± 0.06 (↓)	0.55 ± 0.09 (↑)	0.57 [0.16 - 1.00] (↓)	0.43 [0.25 - 0.50] (↓)
	Graph neural network	0.58 ± 0.12 (↓)	0.58 ± 0.10 (↑)	0.48 [0.00 - 0.95] (↓)	0.40 [0.06 - 0.69] (↓)
	Combined	0.62 ± 0.11 (↓)	0.57 ± 0.09 (↓)	0.52 [0.11 - 0.92] (↓)	0.48 [0.11 - 0.79] (↓)
Unfiltered (only high EREG expression)	Traditional ML	0.74 ± 0.04 (↑)	0.63 ± 0.06 (↑)	**0.83** [0.54 - 1.00]* (↑)	**0.77** [0.39 - 1.00] (↑)
	Graph neural network	0.79 ± 0.06 (↑)	0.69 ± 0.05 (↑)	0.63 [0.22 - 0.94] (↓)	0.58 [0.20 - 0.88] (↓)
	Combined	**0.81 ± 0.05** (↑)	**0.74 ± 0.05** (↑)	0.75 [0.25 - 1.00] (↓)	0.58 [0.20 - 0.88] (↓)

The per-patient classification results were compared against a previous method that predicted all patients with high EREG expression in the PICCOLO trial as responding to treatment [25,70]. Using this approach for the datasets in this chapter gave a balanced accuracy of 0.45 for the filtered dataset and 0.47 for the unfiltered dataset, when including low and high EREG expression patients. In comparison, the highest point estimate of balanced accuracy using our approach on the same dataset was 0.57 (unfiltered dataset, graph neural network, Table 4.12). Alternatively, looking only at patients with high EREG expression, the point estimate of balanced accuracy for the traditional ML model on the unfiltered dataset was 0.77 (Table 4.12) and the standard accuracy was 0.82 [0.55 – 1.0]. These are higher than the balanced accuracy given by predicting all these patients as responders: 0.5 (by definition) and

the corresponding standard accuracy: 0.27. However, each of our models had a large confidence interval that crossed 0.5 for balanced accuracy, suggesting they may be no better than random guessing (Table 4.12). These results also came from a small number of patients. Therefore, further work is needed to ensure that these results are repeatable and applicable to other datasets.

In summary, the models were evaluated without further training on a dataset of unseen cells from a mixture of unseen and seen patients from the PICCOLO trial. For per-cell classification, the best model for the *k*-fold dataset (combined model, unfiltered, only high EREG expression) gave the highest AUROC and balanced accuracy for the reserved test set. However, the graph neural network model (and the combined model by extension) was poorer at per-patient classification of the reserved test set, suggesting that it failed to capture features of the cells that generalised across all of the patients. Instead, for per-patient classification, the best-performing model was the traditional ML model acting on the unfiltered cell data from patients with high EREG expression. This suggested that traditional ML per-cell features, such as size and shape, may best discriminate between non- and any-responders to treatment. Further, the results on this admittedly small dataset suggested that this approach could improve over a previous approach that classified patients in the PICCOLO dataset using the EREG expression. The highest per-cell and per-patient point estimates of AUROC and balanced accuracy for the reserved test set were greater than the best results for the *k*-fold dataset, which may suggest that the models could generalise to new datasets. However, the per-patient results had large confidence intervals, and any conclusions were limited by the small

number of unseen patients in the reserved test set (only one for the unfiltered, high EREG expression dataset).

4.3.7 Handcrafted feature analysis

The most important handcrafted features for traditional ML classification of the *k*-fold dataset were determined by measuring their impact on the model prediction. This was done using Gini impurity and SHapley Additive exPlanation (SHAP) analysis on the best-performing per-cell traditional ML classification model & dataset combination (Table 4.9, Filtered, traditional ML) [251,255]. Gini impurity identified cell length, perimeter, and planarity as the three most important features (most to least important). SHAP analysis also gave length and perimeter as the two most important features, associating lower values of both with any-response (Figure 4.5). However, it identified linearity instead of planarity as the third most important feature (Figure 4.5). This indicated that any-responders and non-responders may be differentiated by the size of their cells (i.e. length, perimeter) and less so by their shape (linearity, planarity).

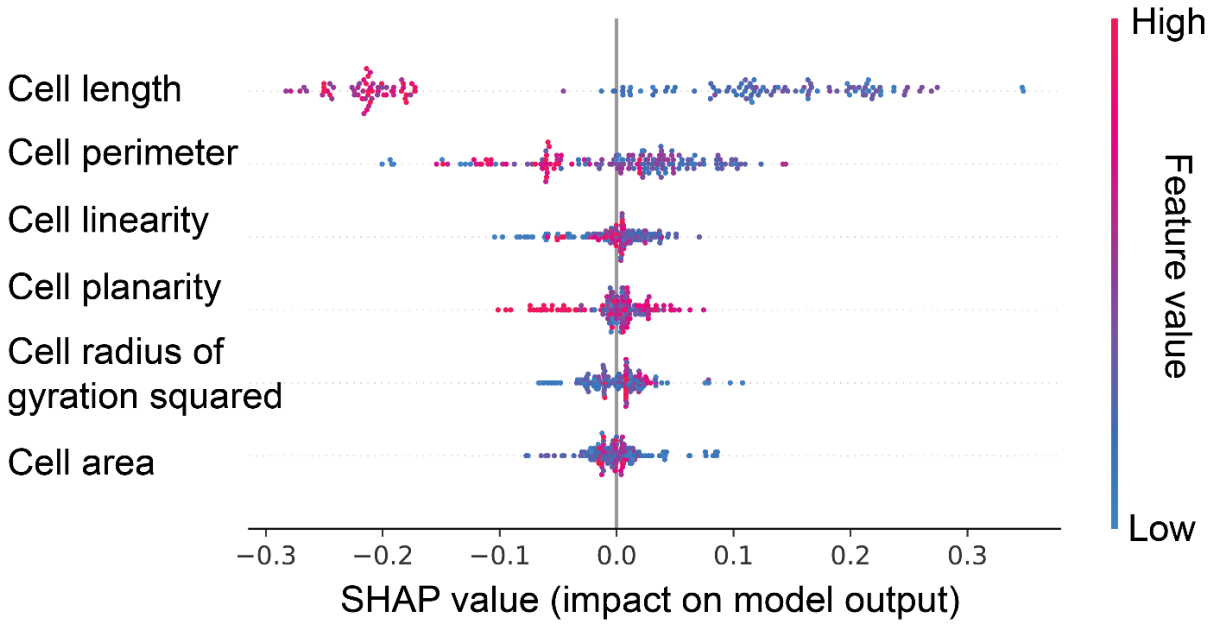

Figure 4.5. SHAP analysis of per-cell traditional ML classification of the *k*-fold dataset. Features are ordered from most (top) to least (bottom) important based on the mean absolute SHAP value. Positive SHAP values indicate the prediction is more likely to be any-response, and negative values indicate the prediction is more likely to be no-response. SHAP values are from the test sets, using the best-performing traditional ML classification model (random forest with per-cell size and shape features).

Investigating the effect on the predicted class was limited as the models did not achieve perfect accuracy. Further, SHAP analysis has numerous limitations, such as assuming that the input features are independent and providing different explanations depending on the input model [258,267]. Therefore, the distribution for each feature was compared between the classes (ground-truth) for the filtered *k*-fold dataset. This corroborated the SHAP analysis with any-response patients having a significantly lower median value for the two most important features, cell length, and perimeter (Figure 4.6a-b). For the remaining per-cell size and shape features, cell area and radius of gyration were also significantly lower for any-responders (Figure 4.6c-d), but linearity and planarity were not significantly different (Figure 4.6e-f). None of the per-cell count features, including those separated into interior and membrane (Table 4.1), were significantly different between non-response and

any-response (Mann–Whitney U: $p > 0.05$, plots not shown). This further suggested that any-responders and non-responders may be differentiated by the size of their cells, but not by their shape.

These handcrafted features could then be compared between any- and non-responders in the reserved test set. For the equivalent reserved test set (filtered, including low EREG expressing patients), the median cell length, perimeter, radius of gyration, and area were also lower for any-responders compared to non-responders, but not significantly (Mann–Whitney U: $p > 0.05$, plots not shown). However, the highest AUROC and balanced accuracy for per-patient classification of the reserved test set was given by the traditional ML model on the unfiltered cell data from patients with high EREG expression (Section 4.3.6). For this dataset, the median cell length, perimeter, radius of gyration and area were significantly lower for any-responders compared to non-responders (Figure 4.7a-d), and there was no significant difference in linearity or planarity (Figure 4.7e-f).

In summary, post-hoc analysis of the traditional ML classification models suggested that any-responders and non-responders can be differentiated by the size of their cells. Further, comparing the distributions of the per-cell features in the k-fold dataset showed that cells from any-responders were significantly smaller (length, perimeter, radius of gyration and area) than those from non-responders, but with no significant difference in shape (linearity and planarity) (Figure 4.6). These trends were also seen in the reserved test set, but were only statistically significant after removing the patients with low EREG expression. There was no difference in the per-cell count features, reflecting the poor performance of the models that used these features, despite previous analysis of tumour samples from the PICCOLO trial showing that

responders had elevated EREG expression (as measured by IHC) [25,70]. This may be because it is unreliable to measure absolute protein expression from SMLM data without careful calibration [135].

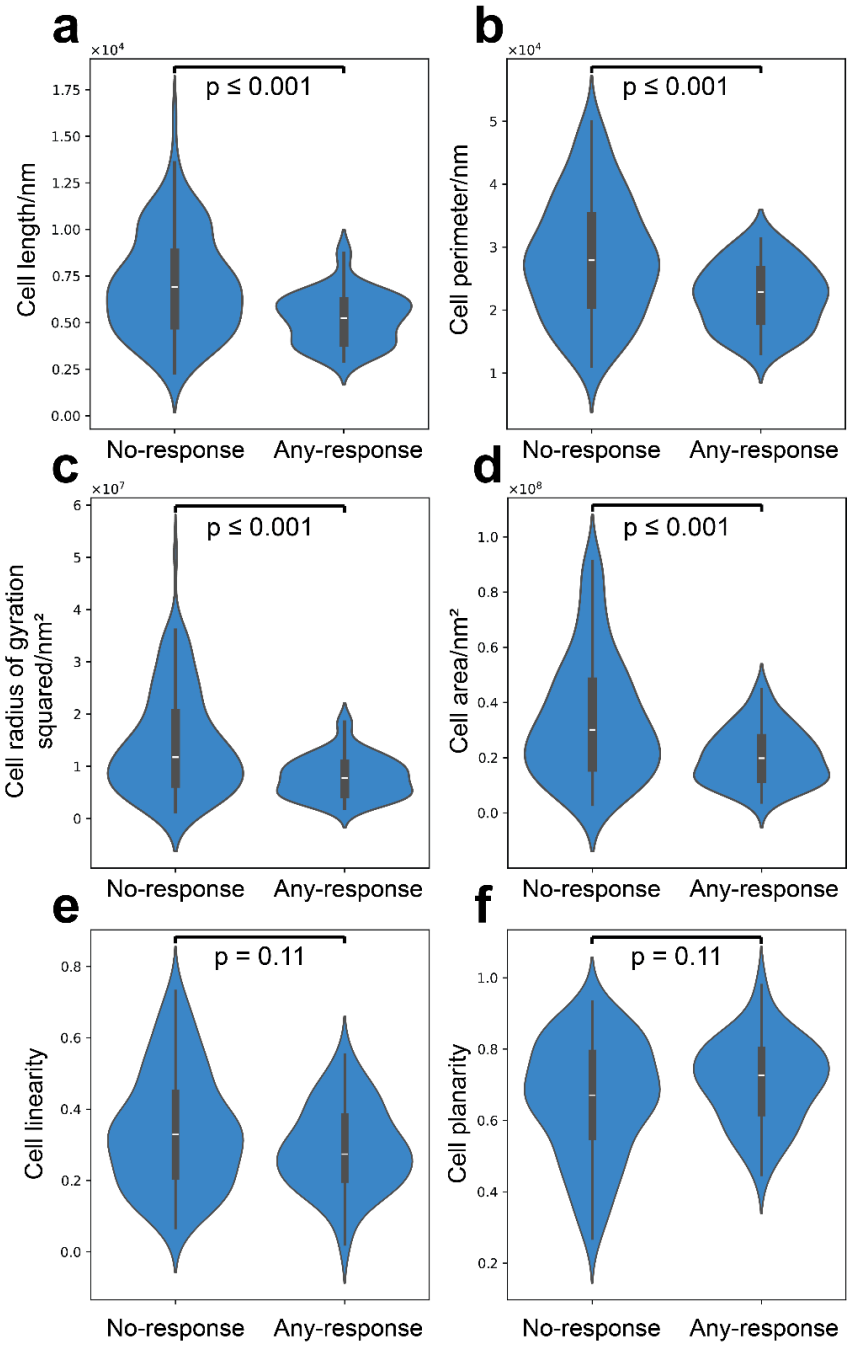

Figure 4.6. Comparing per-cell size and shape features by ground-truth response for the filtered *k*-fold dataset. Violin plots are shown, and the median of the distributions are compared using a two-tailed Mann–Whitney *U* rank test, with the p-value indicated. $n_{\text{no-response}} = 117$, $n_{\textit{any-response}} = 46$. The Shapiro-Wilk test of the null hypothesis that data drawn from a normal distribution was rejected for all distributions ($p \leq 0.05$), apart from any-response cell perimeter, area, linearity, and planarity.

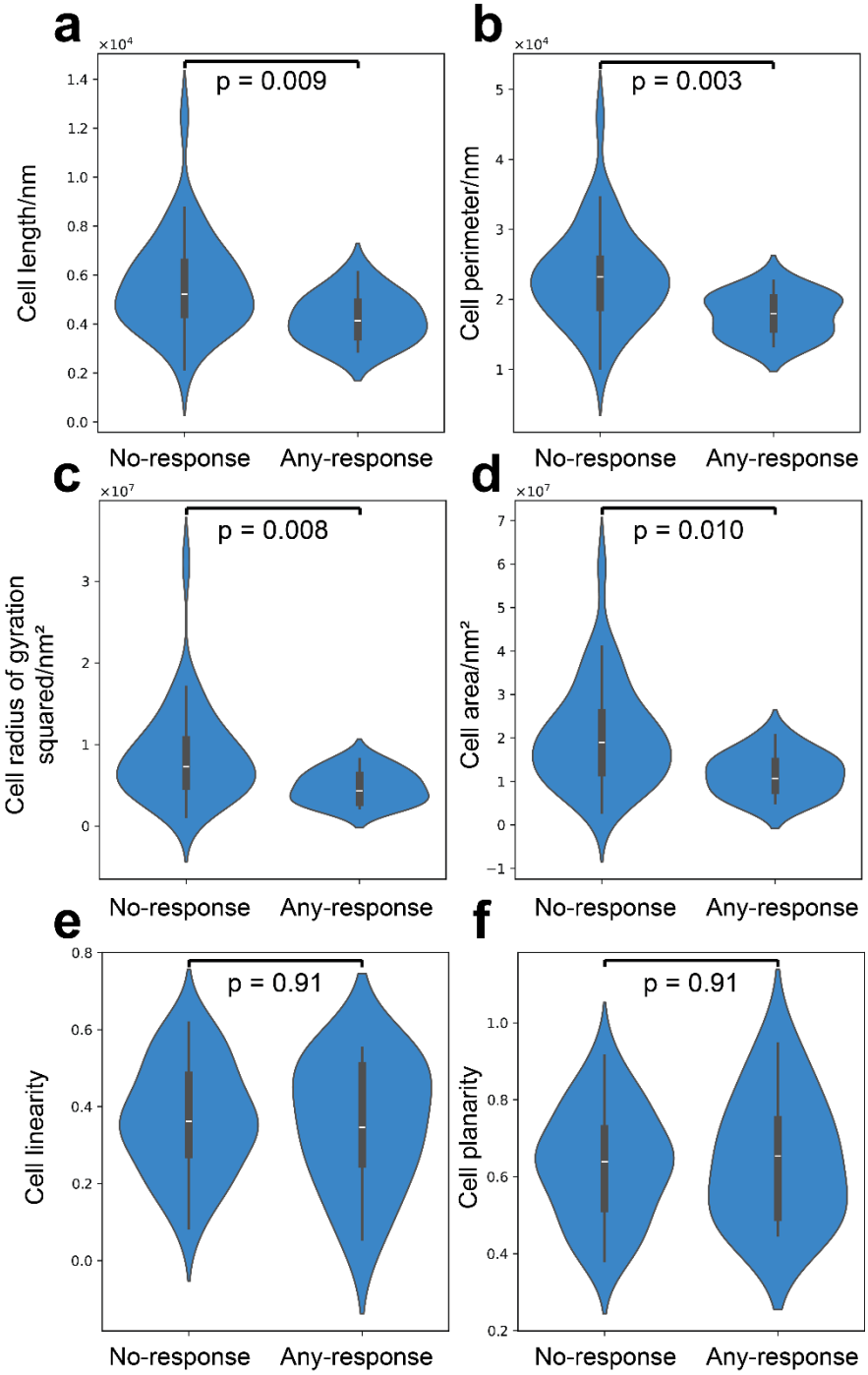

Figure 4.7. Comparing per-cell size and shape features by ground-truth response for the unfiltered high EREG expression reserved test set. Violin plots are shown, and the median of the distributions are compared using a two-tailed Mann–Whitney *U* rank test, with the p-value indicated. $n_{\text{no-response}} = 37$, $n_{\textit{any-response}} = 13$. The Shapiro-Wilk test of the null hypothesis that data drawn from a normal distribution was only rejected ($p \leq 0.05$) for no-response cell length, area and radius of gyration squared.

4.3.8 Deep feature analysis via UMAP

Deep features generated by the graph neural network were represented in 2D (using UMAP) to identify sub-populations within the dataset and provide further insight into model performance. Features from cells in the reserved test set were visualised to allow comparison between multiple patients simultaneously using the best-performing graph neural network & dataset combination: *ClusterNet-HCF* on the unfiltered dataset with only high EREG expression patients (Figure 4.8). The *k*-fold dataset was not analysed, as each test set was embedded by a different model that learnt a unique embedding, which meant that features could not be compared between patients in different test sets. Instead, each test set would have to be analysed separately, restricting comparisons to a limited number of patients at a time.

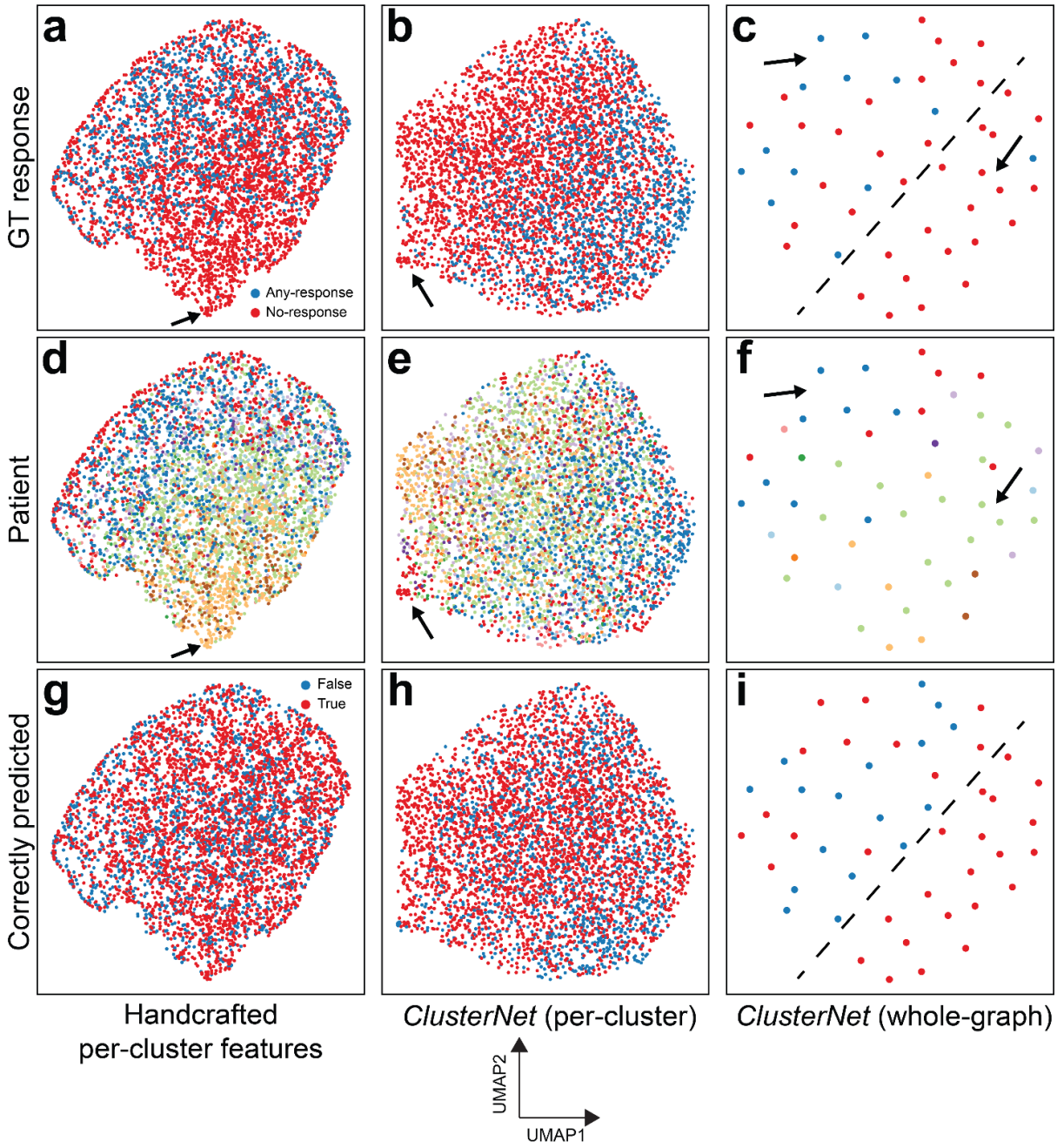

Figure 4.8. Handcrafted and graph neural network-generated features for the reserved test set cells. Features coloured by ground-truth (GT) response (**a**-**c**), patient (**d**-**f**), or prediction accuracy (**g**-**i**). Features incorporate larger structure from left to right: Handcrafted per-cluster features (**a**, **d**, **g**); per-cluster features after message passing but before global pooling in *ClusterNet* (**b**, **e**, **h**); and whole-graph features aggregated from *ClusterNet* (**c**, **f**, **i**). Dashed black line shows the separation between correctly predicted and mostly no-response (below) and a mixture of no-response, any-response, correctly and incorrectly predicted (above). Black arrows indicate clusters of the same patient. *ClusterNet* features generated by *ClusterNet-HCF* for the unfiltered cells from patients with high EREG expression, using the trained instance of the model, from *k*-fold training, which had the highest AUROC on the reserved test set. 2D representations generated by UMAP.

Analysis of the deep features did not provide any significant insights. Per-cluster features (handcrafted and *ClusterNet*) did not separate the classes, but did not appear to be completely random either, with some clusters of features from the same patient (Figure 4.8a,b,d and e, black arrows). Similarly, there was no clear pattern or regions where predictions were more likely to be (in)correct at this scale (Figure 4.8g-h). The whole-graph features seemed to be split between a group that were correctly predicted and mostly no-response (Figure 4.8c and i, below the dashed black line) and the remaining features that formed a mixture of no-response, any-response, correctly and incorrectly predicted (Figure 4.8c and i, above the dashed black line). At this scale, there were also clusters of features from the same patient, similarly to the per-cluster features (Figure 4.8c and f, black arrows. Therefore, incorporating the supra-cluster structure over the whole cell (graph) helped to identify a large group of non-responders and helped to better separate the classes (Figure 4.8c). However, it could not completely separate the classes, reflecting the inaccuracy of the model (Figure 4.8c, Table 4.12).

In summary, post-hoc analysis of the deep features from the graph neural networks via 2D projections provided further insight (Figure 4.8). It showed that per-cluster features did not separate the classes, as seen previously by the poor performance of traditional ML models using per-cluster features, and how the supra-cluster structure in the cell helped to separate the classes. In some cases, it showed that cells from the same patient formed clusters in the feature space, which may indicate that cells from each patient were similar. This highlighted that per-cell classification results should be treated with caution, as the models may learn to recognise cells from the same patient, rather than by their response to treatment. This issue could be avoided

with a larger dataset from more patients, allowing the models to be tested on more unseen patients. On the other hand, it could be useful if different cells from the same patient are similar, as fewer cells from each patient need to be analysed. Visualising the whole-graph features from the patients with high EREG expression revealed a group of non-responders that were correctly predicted by the model. This could help to prevent unnecessary and potentially detrimental treatment for these patients, who would be incorrectly classified based on their EREG expression from IHC. Post-hoc analysis of the traditional ML models provided more insight because of their simplicity, which should be considered against any improvements in classification performance for more complex deep learning models.

4.4 Discussion

In summary, this study has shown promising signs that features derived from individual cells from advanced colorectal cancer patients, stained for EREG and imaged using SMLM, may predict response to anti-EGFR treatment (panitumumab). This included both per-cell and per-patient prediction of response to treatment, where the latter was the most clinically relevant task. The best results were achieved using traditional ML classification models acting on handcrafted features of the cells or a graph-neural network model, *ClusterNet-HCF*, that combined handcrafted features of the clusters with their spatial arrangement. Comparing cell size between any- and non-responders, prompted by post-hoc analysis of the traditional ML models, indicated that cell size may be associated with response to treatment. This study did not conclusively show that SMLM imaging was necessary to predict response, as the best-performing models for per-cell and per-patient classification varied in whether they required the high-precision localisation coordinates.

Nonetheless, this study showed the potential of approaches that extract features from individual cells to improve upon or combine with existing methods that use the expression of EREG to predict response to treatment.

Previous analysis of the tumour samples from the PICCOLO trial showed that responders have elevated EREG expression, which is thought to indicate that the tumour is dependent on the EGFR pathway and therefore more likely to respond to anti-EGFR treatment [25,68,70]. However, in this study, features that represented the amount of EREG localisations (count) were not associated with response to treatment. Traditional ML models that used per-cell counts of EREG localisations divided by whole cell area did not differentiate the cells, and there was no significant difference between any-responders and non-responders for these features. This disagreement could be because previous analyses of PICCOLO samples were restricted to tumour cells only, whereas the cell type for each cell could not be identified in this study [25,70].

On the contrary, classifying the cells with traditional ML models using handcrafted features indicated that the size and, to a lesser extent, the shape of the cells may differentiate responders and non-responders to anti-EGFR treatment. Comparing the distributions of these features showed that cells from any-responders were significantly smaller (length, perimeter, radius of gyration and area) than from non-responders, but with no significant difference in shape (linearity and planarity). One possible explanation is that cells from non-responders may be larger due to increased ploidy (number of chromosomes), which has been associated with tumour progression, more aggressive disease and poorer prognosis for cancer [263-265]. However, as the TMAs contain multiple cell types that could not be identified from

the dSTORM data (e.g. lymphocytes, neutrophils, eosinophils, endothelial cells, fibroblasts, etc.), it may be that the difference in cell size was due to differences in cell types, rather than being a feature of the tumour cells. Further analysis of larger datasets of cells from more patients is required to corroborate (or otherwise) these findings.

More broadly, the study showed that the spatial organisation of EREG may help predict response to anti-EGFR treatment, but it was inconclusive as to whether the nanoscale organisation was required. On the one hand, the nanoscale spatial organisation of EREG appeared to help prediction. *ClusterNet-HCF*, which used the cluster coordinates and handcrafted features calculated from the high-precision coordinates of the localisations, showed promise for predicting response to treatment. For example, the best-performing model for per-cell classification of the reserved test set combined the traditional ML model with *ClusterNet-HCF.* This also suggested that *ClusterNet-HCF* learnt complementary and distinct features to the traditional ML model, which used cell size and shape. Furthermore, increasing the number of clusters to represent the cell from its value in Chapter 3, which meant that each cluster represented a smaller spatial area and captured finer detail, improved the performance of *ClusterNet-HCF*. Also, cell size did not appear as significant for predictions by *ClusterNet-HCF* as it was for the traditional ML models, as normalising on a per-cell rather than per-dataset basis performed better. However, *ClusterNet-HCF* may have accounted for cell size and the amount of ligand per cell through the input cluster features (length, area, number of localisations, etc.), which were normalised on a per-dataset level.

On the other hand, the nanoscale spatial organisation of EREG did not always appear necessary for predicting response. The traditional ML model that used per-cell size and shape features was the best-performing model for per-patient classification of the reserved test set, which was the most clinically relevant task. Further, post-hoc analysis of *ClusterNet-HCF* showed that per-cluster features alone did not separate the classes, as also seen from the poor performance of traditional ML models using per-cluster features, and that incorporating the supra-cluster structure helped to separate the classes. This is despite evidence that clustering of the receptor EGFR can lead to greater downstream signalling [38,40-42,241,242], and that the increased clustering of a closely related receptor, HER2, has been associated with response to anti-HER2 targeted therapy in breast cancer [75]. However, expecting the organisation of EGFR to be reflected in the organisation of EREG may be misguided, as ligand binding is not required for clustering of EGFR [39,124,242].

Nonetheless, this study was able to demonstrate the benefit of moving beyond features calculated from groups of cells from patients (e.g. percentage of cells with high EREG expression) to per-cell features (e.g. cell size). This study suggested that this approach may improve the prediction of response to anti-EGFR treatment compared to an approach used in previous studies that predicts all patients with high EREG expression as responders to treatment [25,70]. In particular, the approach developed here may help to identify patients who will not respond to treatment, amongst those with high EREG expression from IHC. However, this was limited by the small number of patients in our dataset and large confidence intervals for per-patient classification metrics. If, as discussed above, the nanoscale detail provided by SMLM was not required, conventional diffraction-limited imaging of

tissue through H&E and IHC may have been sufficient for the per-cell analysis. For example, cell size and shape features do not require super-resolution imaging to be resolved. In this case, image-based methods may have been appropriate and sufficient, but would not have been as future-proof as our point-based methods for future analysis containing and requiring data of higher precision (Sections 1.2.2 and 3.4).

In future work, some of the limitations of this analysis could be addressed. This includes: investigating both AREG and EREG as the expression of both ligands has been linked with response to treatment in the PICCOLO trial [25,70,268]; improving cell extraction to distinguish between malignant tumour cells and non-malignant cells, which may not change their organisation of EREG in cancer, and to exclude cells at the edge of a field of view that may have had unreliable features; investigating alternative methods to aggregate the per-cell predictions for each patient, particularly if some cells (e.g. of different types or at the edge of the field-of-view) are less informative; increasing the size of the dataset to prevent overfitting to the *k*-fold dataset, to determine the true extent to which cells from the same patient are similar, to allow testing of the models on greater numbers of unseen patients and to confirm whether the association between cell size and response to treatment were artefacts of this dataset or represent a genuine association; improving the model architecture, for example increasing the expressiveness of the traditional ML and *ClusterNet-HCF* models, by including more handcrafted features of different types, such as those from SEMORE and ECLIPSE, which use up to ~70 features per cluster (Section 1.3.2), or by increasing the size of the cluster feature generated by *LocNet* [150,164]; and extending the prediction to the multiple sub-types of any-response and

no-response (e.g. partial response, radiological progression etc.) or to predicting continuous outcomes such as progression-free survival or overall survival.

While this study faced numerous limitations, it provides a basis for future work, which could help to prevent unnecessary treatment for patients that would not respond to treatment and suffer avoidable side effects or to identify patients that would respond, but who may not be identified under current testing procedures.

5 Conclusions and Future Work

This study set out to investigate whether the nanoscale spatial organisation of EREG can predict response to anti-EGFR treatment for metastatic colorectal cancer patients. To address this, we developed two novel AI-based pipelines, *locpix* and *ClusterNet*, for segmenting and classifying SMLM data represented as a point cloud (Chapters 2 and 3). The manual annotation tool from *locpix* was then combined with *ClusterNet* to predict response to treatment from dSTORM images of patient tumour samples (Chapter 4). Comparing this approach to an existing method that uses ligand expression showed promising signs that this type of approach could help predict response to treatment. Ultimately, however, this study could not confirm the overall hypothesis that the nanoscale organisation helped predict response to treatment for these patients.

The approach demonstrated in this study used dSTORM to image sections from formalin-fixed paraffin-embedded (FFPE) tissue microarray (TMA) blocks, which are routinely prepared for pathological analysis of tumour. This provided more evidence that SMLM could be introduced into the routine analysis of clinical samples [84]. This imaging was not always straightforward; for example, high background fluorescence in the images made it hard to identify and annotate cells in many cases. Developments to SMLM imaging of tissue could improve the quality of the data, for example, allowing the sample to be imaged at a greater depth or correcting for the inhomogeneous refractive index in the sample [269-276]. More broadly, we also acknowledge the unresolved challenge of analysing receptor organisation when fixing samples, which can affect the protein organisation at the single-molecule resolution [138,277-279]. Ultimately, we believe this shows that with future optimisation to both the acquisition and analysis, SMLM can feasibly be introduced into healthcare

systems for predicting response to treatment, addressing the first key question in this study (Section 1.4).

The pipelines developed in this study showed how AI algorithms can help to analyse SMLM data, using different data representations (e.g. image, point-cloud or graph) depending on the task and required precision. In developing *locpix*, we showed that existing AI methods for segmenting images can be adapted to successfully extract structures of interest within SMLM data and return annotations at the localisation level. However, the sub-optimal performance, the lack of automatic seed selection for watershed and the lack of automatic quality control for the segmentations meant this was at best a semi-automated pipeline. Further, *locpix* was only used manually to segment the cells from the FFPE samples. Therefore, we could not confirm the second key question in this study (Section 1.4). On the other hand, *ClusterNet*, a novel graph-neural network (GNN), showed how AI can classify structures larger than individual clusters of localisations, while still retaining the precision of the localisations. This directly addressed the third key question in this study (Section 1.4). To the best of our knowledge, we are the first to do whole-graph classification of SMLM data using a GNN. These pipelines may be broadly useful in the analysis of SMLM data from different imaging techniques and samples. This includes clinical samples, where the classes (e.g. disease phenotypes) may not be visually identifiable from reconstructed images.

Existing explainability techniques for AI algorithms provided further insight into the classification of the SMLM data. For example, for both the digits and letters and tumour datasets, visualising a lower-dimensional representation of the deep features identified the spatial scale (e.g. per-cluster vs supra-cluster structure) required to

differentiate the classes. This also revealed outliers and sub-populations in each class. Further, for the digits and letters dataset, SubgraphX was able to reveal sub-structures in SMLM data associated with the classification. To the best of our knowledge, this is the first study to use an explainability algorithm designed for graph neural networks on SMLM data. Despite this, further work is needed to improve the reliability of these algorithms so that they can be confidently used on experimental data, such as the tumour dataset, to identify the relevant substructure for the class. This could include developing explainability algorithms tailored to SMLM data. Ultimately, therefore, existing explainability techniques for AI algorithms could not identify features of organisation associated with response to treatment, the final key question in this study (Section 1.4).

By applying our AI algorithms to analyse dSTORM data from metastatic colorectal cancer patients, we showed how the organisation of EREG could help predict response to anti-EGFR treatment. This reflects the observed association between the organisation of EGFR and its ligands and downstream signalling [30,31,34,38,39,42-48]. However, the study was inconclusive as to whether the nanoscale organisation was required. Nonetheless, for the dataset analysed in this study, our approach improved over an existing method that uses the expression of EREG as measured from immunohistochemistry (IHC). However, this was limited by the small number of patients in our dataset and large confidence intervals for per-patient classification metrics. To the best of our knowledge, this is the first study to use point-based machine learning models to classify patient samples using the nanoscale organisation of membrane proteins from SMLM data. With careful fine-tuning, this kind of analysis could be applied to other disease areas or biological domains.

While the results of this study were promising, the accuracy and reliability of the predictions will need to improve significantly before this approach can be used for routine prediction of response to treatment. This approach would also need to address other issues besides those that were discussed in Chapter 4. These include improving the throughput of the imaging and analysis, analysing datasets from more patients imaged at multiple different sites and centres and conducting a more thorough analysis of per-patient prediction. For example, future work could investigate whether the automatic segmentation methods developed in *locpix* can successfully annotate the cells in the tumour dataset, avoiding manual annotation and thereby increasing throughput.

Future analysis of this dataset could also compare SMLM with other lower-resolution imaging techniques. This could better determine whether the nanoscale detail of protein organisation provided by SMLM is necessary. Alternatively, SMLM could be combined with these techniques to simultaneously consider the protein organisation across multiple different length scales. For example, our approach could be combined with similar models that have been used to predict patient outcomes from diffraction-limited fluorescence imaging of TMAs from colorectal cancer [148,149]. It could also be complemented by methods that apply graph neural networks to graphs of protein-protein interactions to predict response to anti-EGFR treatment in lung cancer [76,280]. Further, our approach could be extended to include two colour imaging of both EREG and AREG, as higher protein expression of AREG is also associated with response to anti-EGFR treatment [58].

More generally, developing and testing *locpix* and *ClusterNet* generated insights and ideas for future AI algorithms that analyse SMLM data. Foremost, it showed the

importance of the data representation and preprocessing steps, such as how the data was filtered or the number of clusters used to represent each cell. As there are many ways to preprocess and represent the data, future work could explore if incorporating these steps into the learnt stages of the algorithm improves classification performance. For example, letting *ClusterNet-LCF* decide which localisations were most important in each cell for the cancer dataset (Chapter 4), by including photophysical features for each localisation and node-level gating of the input, improved its performance on the unfiltered dataset. Similarly, future work could test models that learn how to cluster the localisations and pool information from them, rather than using a fixed clustering algorithm [281-284]. However, *k*-means outperformed DBSCAN for the cancer dataset in this study, showing the benefits of a more static representation in this case. Aggregating the per-cell predictions for each patient could also be incorporated into the model's learning. For example, attention-based pooling could allow the model to learn which cells are more informative, which could in turn reveal if different types of cell (e.g. tumour cells vs. lymphocytes) are more important [285]. This study also showed that while there are many existing AI algorithms, which can be readily adapted to SMLM data, further development is currently held back by a lack of large, labelled datasets of experimental SMLM data from different sample types. Further, *ClusterNet-HCF* consistently outperforming *ClusterNet-LCF* indicated the value of including domain expertise in future AI algorithms, in this case through handcrafted features. However, this may have instead been because the *LocNet* architecture was suboptimal or because it was easier to train *ClusterNet* on its own.

This study demonstrates the possibility of developing clinical prediction tools that combine AI analysis with high-precision data from SMLM. If successful, this could help clinicians to decide the best treatment for a patient, ultimately leading to better patient outcomes. However, achieving this will require extensive development and testing on larger datasets, which is likely to take a long time. Therefore, rather than aiming for clinical prediction tools, the more immediate benefits from this kind of analysis may be in hypothesis generation. By applying post-hoc analysis tools to accurate classification models, features of protein organisation that are associated with response to treatment could be identified. This could then form a feedback loop, where possible features or important subcellular regions identified by AI can be more thoroughly investigated in further lab experiments. This shifts the focus to using AI to help generate biological discoveries from SMLM data [176]. Unlike a prediction tool, these insights are not tied to a particular algorithm, and therefore could ultimately benefit more patients.

6 References

1 Brown, J. S. *et al.* Updating the Definition of Cancer. *Mol Cancer Res* **21**, 1142-1147 (2023). https://doi.org/10.1158/1541-7786.MCR-23-0411

2 Hanahan, D. & Weinberg, R. A. The hallmarks of cancer. *Cell* **100**, 57-70 (2000). https://doi.org/10.1016/s0092-8674(00)81683-9

3 Hanahan, D. & Weinberg, R. A. Hallmarks of cancer: the next generation. *Cell* **144**, 646-674 (2011). https://doi.org/10.1016/j.cell.2011.02.013

4 Hossain, M. S. *et al.* Colorectal Cancer: A Review of Carcinogenesis, Global Epidemiology, Current Challenges, Risk Factors, Preventive and Treatment Strategies. *Cancers (Basel)* **14**, 1732 (2022). https://doi.org/10.3390/cancers14071732

5 Morgan, E. *et al.* Global burden of colorectal cancer in 2020 and 2040: incidence and mortality estimates from GLOBOCAN. *Gut* **72**, 338-344 (2023). https://doi.org/10.1136/gutjnl-2022-327736

6 Cancer Research UK. *Bowel cancer statistics*, <https://www.cancerresearchuk.org/health-professional/cancer-statistics/statistics-by-cancer-type/bowel-cancer> (2025).

7 Sung, H. *et al.* Colorectal cancer incidence trends in younger versus older adults: an analysis of population-based cancer registry data. *Lancet Oncol* **26**, 51-63 (2025). https://doi.org/10.1016/S1470-2045(24)00600-4

8 Ugai, T. *et al.* Is early-onset cancer an emerging global epidemic? Current evidence and future implications. *Nat Rev Clin Oncol* **19**, 656-673 (2022). https://doi.org/10.1038/s41571-022-00672-8

9 Done, J. Z. & Fang, S. H. Young-onset colorectal cancer: A review. *World J Gastrointest Oncol* **13**, 856-866 (2021). https://doi.org/10.4251/wjgo.v13.i8.856

10 Dekker, E., Tanis, P. J., Vleugels, J. L. A., Kasi, P. M. & Wallace, M. B. Colorectal cancer. *Lancet* **394**, 1467-1480 (2019). https://doi.org/10.1016/S0140-6736(19)32319-0

11 Li, J., Ma, X., Chakravarti, D., Shalapour, S. & DePinho, R. A. Genetic and biological hallmarks of colorectal cancer. *Genes Dev* **35**, 787-820 (2021). https://doi.org/10.1101/gad.348226.120

12 Popow, O. *et al.* Identification of Endogenous Adenomatous Polyposis Coli Interaction Partners and beta-Catenin-Independent Targets by Proteomics. *Mol Cancer Res* **17**, 1828-1841 (2019). https://doi.org/10.1158/1541-7786.MCR-18-1154

13 Grady, W. M., Yu, M. & Markowitz, S. D. Epigenetic Alterations in the Gastrointestinal Tract: Current and Emerging Use for Biomarkers of Cancer. *Gastroenterology* **160**, 690-709 (2021). https://doi.org/10.1053/j.gastro.2020.09.058

14 Eng, C. *et al.* Colorectal cancer. *Lancet* **404**, 294-310 (2024). https://doi.org/10.1016/S0140-6736(24)00360-X

15 Fleming, M., Ravula, S., Tatishchev, S. F. & Wang, H. L. Colorectal carcinoma: Pathologic aspects. *J Gastrointest Oncol* **3**, 153-173 (2012). https://doi.org/10.3978/j.issn.2078-6891.2012.030

16 Cancer Research UK. *Early Diagnosis Hub: Incidence by Stage (Gold Standard)*, <https://crukcancerintelligence.shinyapps.io/EarlyDiagnosis/> (2024).

17 NHS England. *Cancer Survival in England, cancers diagnosed 2016 to 2020, followed up to 2021*, <https://digital.nhs.uk/data-and-information/publications/statistical/cancer-survival-in-england/cancers-diagnosed-2016-to-2020-followed-up-to-2021> (2025).

18 National Institute for Health and Care Excellence. *Colorectal cancer: Recommendations*, <https://www.nice.org.uk/guidance/ng151/chapter/recommendations> (2025).

19 Ciardiello, F. & Tortora, G. EGFR antagonists in cancer treatment. *N Engl J Med* **358**, 1160-1174 (2008). https://doi.org/10.1056/NEJMra0707704

20 Xie, Y. H., Chen, Y. X. & Fang, J. Y. Comprehensive review of targeted therapy for colorectal cancer. *Signal Transduct Target Ther* **5**, 22 (2020). https://doi.org/10.1038/s41392-020-0116-z

21 Li, Q. H. *et al.* Anti-EGFR therapy in metastatic colorectal cancer: mechanisms and potential regimens of drug resistance. *Gastroenterol Rep (Oxf)* **8**, 179-191 (2020). https://doi.org/10.1093/gastro/goaa026

22 Cervantes, A. *et al.* Metastatic colorectal cancer: ESMO Clinical Practice Guideline for diagnosis, treatment and follow-up. *Ann Oncol* **34**, 10-32 (2023). https://doi.org/10.1016/j.annonc.2022.10.003

23 Bando, H., Ohtsu, A. & Yoshino, T. Therapeutic landscape and future direction of metastatic colorectal cancer. *Nat Rev Gastroenterol Hepatol* **20**, 306-322 (2023). https://doi.org/10.1038/s41575-022-00736-1

24 Stintzing, S. *et al.* FOLFOXIRI Plus Cetuximab or Bevacizumab as First-Line Treatment of BRAF(V600E)-Mutant Metastatic Colorectal Cancer: The Randomized Phase II FIRE-4.5 (AIO KRK0116) Study. *J Clin Oncol* **41**, 4143-4153 (2023). https://doi.org/10.1200/JCO.22.01420

25 Williams, C. J. M. *et al.* Artificial Intelligence-Assisted Amphiregulin and Epiregulin IHC Predicts Panitumumab Benefit in RAS Wild-Type Metastatic Colorectal Cancer. *Clin Cancer Res* **27**, 3422-3431 (2021). https://doi.org/10.1158/1078-0432.CCR-21-0120

26 Seymour, M. T. *et al.* Panitumumab and irinotecan versus irinotecan alone for patients with KRAS wild-type, fluorouracil-resistant advanced colorectal cancer (PICCOLO): a prospectively stratified randomised trial. *Lancet Oncol* **14**, 749-759 (2013). https://doi.org/10.1016/S1470-2045(13)70163-3

27 Saltz, L. B. The Cost and Value of Anti-Epidermal Growth Factor Receptor Therapies: Let's Not Be Rash. *JAMA Oncol* **1**, 141-142 (2015). https://doi.org/10.1001/jamaoncol.2015.0287

28 Citri, A. & Yarden, Y. EGF-ERBB signalling: towards the systems level. *Nat Rev Mol Cell Biol* **7**, 505-516 (2006). https://doi.org/10.1038/nrm1962

29 Bakker, J., Spits, M., Neefjes, J. & Berlin, I. The EGFR odyssey - from activation to destruction in space and time. *J Cell Sci* **130**, 4087-4096 (2017). https://doi.org/10.1242/jcs.209197

30 Burgess, A. W. Regulation of Signaling from the Epidermal Growth Factor Family. *J Phys Chem B* **126**, 7475-7485 (2022). https://doi.org/10.1021/acs.jpcb.2c04156

31 Mudumbi, K. C. *et al.* Distinct interactions stabilize EGFR dimers and higher-order oligomers in cell membranes. *Cell Rep* **43**, 113603 (2024). https://doi.org/10.1016/j.celrep.2023.113603

32 Tito, C., Masciarelli, S., Colotti, G. & Fazi, F. EGF receptor in organ development, tissue homeostasis and regeneration. *J Biomed Sci* **32**, 24 (2025). https://doi.org/10.1186/s12929-025-01119-9

33 Hynes, N. E. & Lane, H. A. ERBB receptors and cancer: the complexity of targeted inhibitors. *Nat Rev Cancer* **5**, 341-354 (2005). https://doi.org/10.1038/nrc1609

34 Kovacs, E., Zorn, J. A., Huang, Y., Barros, T. & Kuriyan, J. A structural perspective on the regulation of the epidermal growth factor receptor. *Annu Rev Biochem* **84**, 739-764 (2015). https://doi.org/10.1146/annurev-biochem-060614-034402

35 Lemmon, M. A. & Schlessinger, J. Cell signaling by receptor tyrosine kinases. *Cell* **141**, 1117-1134 (2010). https://doi.org/10.1016/j.cell.2010.06.011

36 Levantini, E., Maroni, G., Del Re, M. & Tenen, D. G. EGFR signaling pathway as therapeutic target in human cancers. *Semin Cancer Biol* **85**, 253-275 (2022). https://doi.org/10.1016/j.semcancer.2022.04.002

37 Yarden, Y. & Sliwkowski, M. X. Untangling the ErbB signalling network. *Nat Rev Mol Cell Biol* **2**, 127-137 (2001). https://doi.org/10.1038/35052073

38 van Lengerich, B., Agnew, C., Puchner, E. M., Huang, B. & Jura, N. EGF and NRG induce phosphorylation of HER3/ERBB3 by EGFR using distinct oligomeric mechanisms. *Proc Natl Acad Sci U S A* **114**, E2836-E2845 (2017). https://doi.org/10.1073/pnas.1617994114

39 Wollman, A. J. M. *et al.* Critical roles for EGFR and EGFR-HER2 clusters in EGF binding of SW620 human carcinoma cells. *J R Soc Interface* **19**, 20220088 (2022). https://doi.org/10.1098/rsif.2022.0088

40 Ichinose, J., Murata, M., Yanagida, T. & Sako, Y. EGF signalling amplification induced by dynamic clustering of EGFR. *Biochem Biophys Res Commun* **324**, 1143-1149 (2004). https://doi.org/10.1016/j.bbrc.2004.09.173

41 Needham, S. R. *et al.* Measuring EGFR separations on cells with ~10 nm resolution via fluorophore localization imaging with photobleaching. *PLoS One* **8**, e62331 (2013). https://doi.org/10.1371/journal.pone.0062331

42 Gao, J. *et al.* Mechanistic insights into EGFR membrane clustering revealed by super-resolution imaging. *Nanoscale* **7**, 2511-2519 (2015). https://doi.org/10.1039/c4nr04962d

43 Chung, I. *et al.* Spatial control of EGF receptor activation by reversible dimerization on living cells. *Nature* **464**, 783-787 (2010). https://doi.org/10.1038/nature08827

44 Klarlund, J. K. Dual modes of motility at the leading edge of migrating epithelial cell sheets. *Proc Natl Acad Sci U S A* **109**, 15799-15804 (2012). https://doi.org/10.1073/pnas.1210992109

45 Winckler, P. *et al.* Identification and super-resolution imaging of ligand-activated receptor dimers in live cells. *Sci Rep* **3**, 2387 (2013). https://doi.org/10.1038/srep02387

46 Needham, S. R. *et al.* EGFR oligomerization organizes kinase-active dimers into competent signalling platforms. *Nat Commun* **7**, 13307 (2016). https://doi.org/10.1038/ncomms13307

47 Zhang, Q. & Reinhard, B. M. Ligand Density and Nanoparticle Clustering Cooperate in the Multivalent Amplification of Epidermal Growth Factor Receptor Activation. *ACS Nano* **12**, 10473-10485 (2018). https://doi.org/10.1021/acsnano.8b06141

48 Mayer, I. *et al.* Surface-Patterned DNA Origami Rulers Reveal Nanoscale Distance Dependency of the Epidermal Growth Factor Receptor Activation. *Nano Lett* **24**, 1611-1619 (2024). https://doi.org/10.1021/acs.nanolett.3c04272

49 Fang, T. *et al.* Spatial Regulation of T-Cell Signaling by Programmed Death-Ligand 1 on Wireframe DNA Origami Flat Sheets. *ACS Nano* **15**, 3441-3452 (2021). https://doi.org/10.1021/acsnano.0c10632

50 Dong, R. *et al.* DNA origami patterning of synthetic T cell receptors reveals spatial control of the sensitivity and kinetics of signal activation. *Proc Natl Acad Sci U S A* **118**, e2109057118 (2021). https://doi.org/10.1073/pnas.2109057118

51 Vecchione, L., Jacobs, B., Normanno, N., Ciardiello, F. & Tejpar, S. EGFR-targeted therapy. *Exp Cell Res* **317**, 2765-2771 (2011). https://doi.org/10.1016/j.yexcr.2011.08.021

52 Arteaga, C. L. & Engelman, J. A. ERBB receptors: from oncogene discovery to basic science to mechanism-based cancer therapeutics. *Cancer Cell* **25**, 282-303 (2014). https://doi.org/10.1016/j.ccr.2014.02.025

53 Tomas, A., Futter, C. E. & Eden, E. R. EGF receptor trafficking: consequences for signaling and cancer. *Trends Cell Biol* **24**, 26-34 (2014). https://doi.org/10.1016/j.tcb.2013.11.002

54 Garcia-Foncillas, J. *et al.* Distinguishing Features of Cetuximab and Panitumumab in Colorectal Cancer and Other Solid Tumors. *Front Oncol* **9**, 849 (2019). https://doi.org/10.3389/fonc.2019.00849

55 Kurai, J. *et al.* Antibody-dependent cellular cytotoxicity mediated by cetuximab against lung cancer cell lines. *Clin Cancer Res* **13**, 1552-1561 (2007). https://doi.org/10.1158/1078-0432.CCR-06-1726

56 Kimura, H. *et al.* Antibody-dependent cellular cytotoxicity of cetuximab against tumor cells with wild-type or mutant epidermal growth factor receptor. *Cancer Sci* **98**, 1275-1280 (2007). https://doi.org/10.1111/j.1349-7006.2007.00510.x

57 Okada, Y. *et al.* EGFR Downregulation after Anti-EGFR Therapy Predicts the Antitumor Effect in Colorectal Cancer. *Mol Cancer Res* **15**, 1445-1454 (2017). https://doi.org/10.1158/1541-7786.MCR-16-0383

58 Appleyard, J. W., Williams, C. J. M., Manca, P., Pietrantonio, F. & Seligmann, J. F. Targeting the MAP kinase pathway in colorectal cancer: A journey in personalized medicine. *Clin Cancer Res* **31**, 2565–2572 (2025). https://doi.org/10.1158/1078-0432.CCR-25-0107

59 Zhou, J., Ji, Q. & Li, Q. Resistance to anti-EGFR therapies in metastatic colorectal cancer: underlying mechanisms and reversal strategies. *J Exp Clin Cancer Res* **40**, 328 (2021). https://doi.org/10.1186/s13046-021-02130-2

60 Custodio, A. & Feliu, J. Prognostic and predictive biomarkers for epidermal growth factor receptor-targeted therapy in colorectal cancer: beyond KRAS mutations. *Crit Rev Oncol Hematol* **85**, 45-81 (2013). https://doi.org/10.1016/j.critrevonc.2012.05.001

61 Cappuzzo, F. *et al.* Epidermal growth factor receptor gene and protein and gefitinib sensitivity in non-small-cell lung cancer. *J Natl Cancer Inst* **97**, 643-655 (2005). https://doi.org/10.1093/jnci/dji112

62 Personeni, N. *et al.* EGFR and HER2 in colorectal cancer: Analysis of gene copy number and protein expression over disease progression. Does intratumoral heterogeneity matter? *Cancer Research* **67**, 5030-5030 (2007).

63 Algars, A., Lintunen, M., Carpen, O., Ristamaki, R. & Sundstrom, J. EGFR gene copy number assessment from areas with highest EGFR expression predicts response to anti-EGFR therapy in colorectal cancer. *Br J Cancer* **105**, 255-262 (2011). https://doi.org/10.1038/bjc.2011.223

64 Algars, A. *et al.* Heterogeneous EGFR gene copy number increase is common in colorectal cancer and defines response to anti-EGFR therapy. *PLoS One* **9**, e99590 (2014). https://doi.org/10.1371/journal.pone.0099590

65 Del Carmen, S. *et al.* Prognostic implications of EGFR protein expression in sporadic colorectal tumors: Correlation with copy number status, mRNA levels and miRNA regulation. *Sci Rep* **10**, 4662 (2020). https://doi.org/10.1038/s41598-020-61688-7

66 Yoshida, M. *et al.* A novel predictive strategy by immunohistochemical analysis of four EGFR ligands in metastatic colorectal cancer treated with anti-EGFR antibodies. *J Cancer Res Clin Oncol* **139**, 367-378 (2013). https://doi.org/10.1007/s00432-012-1340-x

67 Lu, X. *et al.* Prognostic and predictive biomarkers for anti-EGFR monoclonal antibody therapy in RAS wild-type metastatic colorectal cancer: a systematic review and meta-analysis. *BMC Cancer* **23**, 1117 (2023). https://doi.org/10.1186/s12885-023-11600-z

68 Oliveras-Ferraros, C. *et al.* Cross-suppression of EGFR ligands amphiregulin and epiregulin and de-repression of FGFR3 signalling contribute to cetuximab resistance in wild-type KRAS tumour cells. *Br J Cancer* **106**, 1406-1414 (2012). https://doi.org/10.1038/bjc.2012.103

69 Misale, S., Di Nicolantonio, F., Sartore-Bianchi, A., Siena, S. & Bardelli, A. Resistance to anti-EGFR therapy in colorectal cancer: from heterogeneity to convergent evolution. *Cancer Discov* **4**, 1269-1280 (2014). https://doi.org/10.1158/2159-8290.CD-14-0462

70 Williams, C. J. M. *et al.* Associations between AI-Assisted Tumor Amphiregulin and Epiregulin IHC and Outcomes from Anti-EGFR Therapy in the Routine Management of Metastatic Colorectal Cancer. *Clin Cancer Res* **29**, 4153-4165 (2023). https://doi.org/10.1158/1078-0432.CCR-23-0859

71 Williams, C. *et al.* A biomarker enrichment trial of anti-EGFR agents in right primary tumor location (rPTL), RAS wild-type (RAS-wt) advanced colorectal cancer (aCRC): ARIEL (ISRCTN11061442). *Journal of Clinical Oncology* **40**, TPS3633-TPS3633 (2022). https://doi.org/10.1200/JCO.2022.40.16_suppl.TPS3633

72 Airoldi, M. *et al.* First-Line Therapy in Metastatic, RAS Wild-Type, Left-Sided Colorectal Cancer: Should Everyone Receive Anti-EGFR Therapy? *Curr Oncol Rep* **26**, 1489-1501 (2024). https://doi.org/10.1007/s11912-024-01601-x

73 Wang, Y. N. & Hung, M. C. Nuclear functions and subcellular trafficking mechanisms of the epidermal growth factor receptor family. *Cell Biosci* **2**, 13 (2012). https://doi.org/10.1186/2045-3701-2-13

74 Lee, H. H., Wang, Y. N. & Hung, M. C. Non-canonical signaling mode of the epidermal growth factor receptor family. *Am J Cancer Res* **5**, 2944-2958 (2015).

75 Maddox, A. L. *et al.* Molecular Assessment of HER2 to Identify Signatures Associated with Therapy Response in HER2-Positive Breast Cancer. *Cancers (Basel)* **14**, 2795 (2022). https://doi.org/10.3390/cancers14112795

76 Cai, S. *et al.* Spatially resolved subcellular protein-protein interactomics in drug-perturbed lung-cancer cultures and tissues. *Nat Biomed Eng* (2024). https://doi.org/10.1038/s41551-024-01271-x

77 Peckham, M. *Histology at a Glance*. 1st edn, (Wiley-Blackwell, 2011).

78 Ovalle, W. K., Chovan, J., Nahirney, P. C. & Netter, F. H. *Netter's essential histology : with correlated histopathology*. 3rd edn, (Elsevier, Incorporated, 2021).

79 Abbe, E. Beiträge zur Theorie des Mikroskops und der mikroskopischen Wahrnehmung. *Archiv für Mikroskopische Anatomie* **9**, 413-468 (1873). https://doi.org/10.1007/BF02956173

80 Liu, S., Hoess, P. & Ries, J. Super-Resolution Microscopy for Structural Cell Biology. *Annu Rev Biophys* **51**, 301-326 (2022). https://doi.org/10.1146/annurev-biophys-102521-112912

81 Jahnke, K. *et al.* Formation of EGFRwt/EGFRvIII homo- and hetero-dimers in glioblastoma cells as detected by single molecule localization microscopy. *Nanoscale* **16**, 15240-15255 (2024). https://doi.org/10.1039/d4nr01570c

82 Sauer, M. & Heilemann, M. Single-Molecule Localization Microscopy in Eukaryotes. *Chem Rev* **117**, 7478-7509 (2017). https://doi.org/10.1021/acs.chemrev.6b00667

83 Lelek, M. *et al.* Single-molecule localization microscopy. *Nat Rev Methods Primers* **1**, 39 (2021). https://doi.org/10.1038/s43586-021-00038-x

84 Brockmoeller, S. F. *et al.* Single-molecule localisation microscopy (SMLM) is feasible in human and animal formalin fixed paraffin embedded (FFPE) tissues in medical renal disease. *J Clin Pathol* (2025). https://doi.org/10.1136/jcp-2024-209853

85 Baddeley, D. & Bewersdorf, J. Biological Insight from Super-Resolution Microscopy: What We Can Learn from Localization-Based Images. *Annu Rev Biochem* **87**, 965-989 (2018). https://doi.org/10.1146/annurev-biochem-060815-014801

86 Xu, K., Zhong, G. & Zhuang, X. Actin, spectrin, and associated proteins form a periodic cytoskeletal structure in axons. *Science* **339**, 452-456 (2013). https://doi.org/10.1126/science.1232251

87 Abbe, E. XV.—The Relation of Aperture and Power in the Microscope (continued). *Journal of the Royal Microscopical Society* **3**, 790-812 (1883). https://doi.org/10.1111/j.1365-2818.1883.tb05956.x

88 Laine, R. F., Kaminski Schierle, G. S., van de Linde, S. & Kaminski, C. F. From single-molecule spectroscopy to super-resolution imaging of the neuron: a review. *Methods Appl Fluoresc* **4**, 022004 (2016). https://doi.org/10.1088/2050-6120/4/2/022004

89 Rust, M. J., Bates, M. & Zhuang, X. Sub-diffraction-limit imaging by stochastic optical reconstruction microscopy (STORM). *Nat Methods* **3**, 793-795 (2006). https://doi.org/10.1038/nmeth929

90 Heilemann, M. *et al.* Subdiffraction-resolution fluorescence imaging with conventional fluorescent probes. *Angew Chem Int Ed Engl* **47**, 6172-6176 (2008). https://doi.org/10.1002/anie.200802376

91 Betzig, E. *et al.* Imaging intracellular fluorescent proteins at nanometer resolution. *Science* **313**, 1642-1645 (2006). https://doi.org/10.1126/science.1127344

92 Hess, S. T., Girirajan, T. P. & Mason, M. D. Ultra-high resolution imaging by fluorescence photoactivation localization microscopy. *Biophys J* **91**, 4258-4272 (2006). https://doi.org/10.1529/biophysj.106.091116

93 Sharonov, A. & Hochstrasser, R. M. Wide-field subdiffraction imaging by accumulated binding of diffusing probes. *Proc Natl Acad Sci U S A* **103**, 18911-18916 (2006). https://doi.org/10.1073/pnas.0609643104

94 Holland, K. L. *et al.* A series of spontaneously blinking dyes for super-resolution microscopy. *bioRxiv* (2024). https://doi.org/10.1101/2024.02.23.581625

95 Jungmann, R. *et al.* Single-molecule kinetics and super-resolution microscopy by fluorescence imaging of transient binding on DNA origami. *Nano Lett* **10**, 4756-4761 (2010). https://doi.org/10.1021/nl103427w

96 Strauss, M. T. Picasso-server: a community-based, open-source processing framework for super-resolution data. *Commun Biol* **5**, 930 (2022). https://doi.org/10.1038/s42003-022-03909-5

97 Ries, J. SMAP: a modular super-resolution microscopy analysis platform for SMLM data. *Nat Methods* **17**, 870-872 (2020). https://doi.org/10.1038/s41592-020-0938-1

98 Ovesny, M., Krizek, P., Borkovec, J., Svindrych, Z. & Hagen, G. M. ThunderSTORM: a comprehensive ImageJ plug-in for PALM and STORM data analysis and super-resolution imaging. *Bioinformatics* **30**, 2389-2390 (2014). https://doi.org/10.1093/bioinformatics/btu202

99 Kay, S. M. *Fundamentals of statistical signal processing: estimation theory*. (Prentice-Hall, Inc., 1993).

100 Baddeley, D., Cannell, M. B. & Soeller, C. Visualization of localization microscopy data. *Microsc Microanal* **16**, 64-72 (2010). https://doi.org/10.1017/S143192760999122X

101 Caicedo, J. C. *et al.* Data-analysis strategies for image-based cell profiling. *Nat Methods* **14**, 849-863 (2017). https://doi.org/10.1038/nmeth.4397

102 Moen, E. *et al.* Deep learning for cellular image analysis. *Nat Methods* **16**, 1233-1246 (2019). https://doi.org/10.1038/s41592-019-0403-1

103 Khater, I. M., Aroca-Ouellette, S. T., Meng, F., Nabi, I. R. & Hamarneh, G. Caveolae and scaffold detection from single molecule localization microscopy data using deep learning. *PLoS One* **14**, e0211659 (2019). https://doi.org/10.1371/journal.pone.0211659

104 Nicovich, P. R., Owen, D. M. & Gaus, K. Turning single-molecule localization microscopy into a quantitative bioanalytical tool. *Nat Protoc* **12**, 453-460 (2017). https://doi.org/10.1038/nprot.2016.166

105 Khater, I. M., Nabi, I. R. & Hamarneh, G. A Review of Super-Resolution Single-Molecule Localization Microscopy Cluster Analysis and Quantification Methods. *Patterns (N Y)* **1**, 100038 (2020). https://doi.org/10.1016/j.patter.2020.100038

106 Hyun, Y. & Kim, D. Recent development of computational cluster analysis methods for single-molecule localization microscopy images. *Comput Struct Biotechnol J* **21**, 879-888 (2023). https://doi.org/10.1016/j.csbj.2023.01.006

107 Hugelier, S., Colosi, P. L. & Lakadamyali, M. Quantitative Single-Molecule Localization Microscopy. *Annu Rev Biophys* **52**, 139-160 (2023). https://doi.org/10.1146/annurev-biophys-111622-091212

108 Subach, F. V. *et al.* Photoactivatable mCherry for high-resolution two-color fluorescence microscopy. *Nat Methods* **6**, 153-159 (2009). https://doi.org/10.1038/nmeth.1298

109 Owen, D. M. *et al.* PALM imaging and cluster analysis of protein heterogeneity at the cell surface. *J Biophotonics* **3**, 446-454 (2010). https://doi.org/10.1002/jbio.200900089

110 Sengupta, P. *et al.* Probing protein heterogeneity in the plasma membrane using PALM and pair correlation analysis. *Nat Methods* **8**, 969-975 (2011). https://doi.org/10.1038/nmeth.1704

111 Werbin, J. L. *et al.* Multiplexed Exchange-PAINT imaging reveals ligand-dependent EGFR and Met interactions in the plasma membrane. *Sci Rep* **7**, 12150 (2017). https://doi.org/10.1038/s41598-017-12257-y

112 Masullo, L. A. *et al.* Spatial and stoichiometric in situ analysis of biomolecular oligomerization at single-protein resolution. *Nat Commun* **16**, 4202 (2025). https://doi.org/10.1038/s41467-025-59500-z

113 Mukund, K. *et al.* Molecular Atlas of HER2+ Breast Cancer Cells Treated with Endogenous Ligands: Temporal Insights into Mechanisms of Trastuzumab Resistance. *Cancers (Basel)* **16**, 553 (2024). https://doi.org/10.3390/cancers16030553

114 Pilarczyk, G., Papenfuss, F., Bestvater, F. & Hausmann, M. Spatial Arrangements of Connexin43 in Cancer Related Cells and Re-Arrangements under Treatment Conditions: Investigations on the Nano-Scale by Super-Resolution Localization Light Microscopy. *Cancers (Basel)* **11** (2019). https://doi.org/10.3390/cancers11030301

115 Wakefield, D. L. *et al.* Using quantitative single molecule localization microscopy to optimize multivalent HER2-targeting ligands. *Front Med (Lausanne)* **10**, 1064242 (2023). https://doi.org/10.3389/fmed.2023.1064242

116 Annibale, P., Vanni, S., Scarselli, M., Rothlisberger, U. & Radenovic, A. Quantitative photo activated localization microscopy: unraveling the effects of photoblinking. *PLoS One* **6**, e22678 (2011). https://doi.org/10.1371/journal.pone.0022678

117 Jorand, R. *et al.* Molecular signatures of mu opioid receptor and somatostatin receptor 2 in pancreatic cancer. *Mol Biol Cell* **27**, 3659-3672 (2016). https://doi.org/10.1091/mbc.E16-06-0427

118 Lennon, K. M. *et al.* Single molecule characterization of individual extracellular vesicles from pancreatic cancer. *J Extracell Vesicles* **8**, 1685634 (2019). https://doi.org/10.1080/20013078.2019.1685634

119 Qi, X. *et al.* Mechanistic insights into CDCP1 clustering on non-small-cell lung cancer membranes revealed by super-resolution fluorescent imaging. *iScience* **26**, 106103 (2023). https://doi.org/10.1016/j.isci.2023.106103

120 Li, B. *et al.* Super-resolution imaging reveals the role of DDR1 cluster in NSCLC proliferation. *Talanta* **282**, 127024 (2025). https://doi.org/10.1016/j.talanta.2024.127024

121 Boyd, P. S. *et al.* Clustered localization of EGFRvIII in glioblastoma cells as detected by high precision localization microscopy. *Nanoscale* **8**, 20037-20047 (2016). https://doi.org/10.1039/c6nr05880a

122 Tobin, S. J. *et al.* Single molecule localization microscopy coupled with touch preparation for the quantification of trastuzumab-bound HER2. *Sci Rep* **8**, 15154 (2018). https://doi.org/10.1038/s41598-018-33225-0

123 Cresens, C. *et al.* Flat clathrin lattices are linked to metastatic potential in colorectal cancer. *iScience* **26**, 107327 (2023). https://doi.org/10.1016/j.isci.2023.107327

124 Wang, Y. *et al.* Regulation of EGFR nanocluster formation by ionic protein-lipid interaction. *Cell Res* **24**, 959-976 (2014). https://doi.org/10.1038/cr.2014.89

125 Wei, J. *et al.* Highly Accurate Profiling of Exosome Phenotypes Using Super-resolution Tricolor Fluorescence Co-localization. *ACS Nano* **18**, 10206-10215 (2024). https://doi.org/10.1021/acsnano.4c00534

126 Jiang, N. *et al.* Multiparametric profiling of HER2-enriched extracellular vesicles in breast cancer using Single Extracellular VEsicle Nanoscopy. *J Nanobiotechnology* **22**, 589 (2024). https://doi.org/10.1186/s12951-024-02858-x

127 Xu, J. *et al.* Super-resolution imaging reveals the evolution of higher-order chromatin folding in early carcinogenesis. *Nat Commun* **11**, 1899 (2020). https://doi.org/10.1038/s41467-020-15718-7

128 Lang, F. *et al.* Tackling Tumour Cell Heterogeneity at the Super-Resolution Level in Human Colorectal Cancer Tissue. *Cancers (Basel)* **13** (2021). https://doi.org/10.3390/cancers13153692

129 Xu, J. *et al.* Ultrastructural visualization of chromatin in cancer pathogenesis using a simple small-molecule fluorescent probe. *Sci Adv* **8**, eabm8293 (2022). https://doi.org/10.1126/sciadv.abm8293

130 Wang, Y. *et al.* Affinity fine-tuning anti-CAIX CAR-T cells mitigate on-target off-tumor side effects. *Mol Cancer* **23**, 56 (2024). https://doi.org/10.1186/s12943-024-01952-w

131 Toms, L., FitzPatrick, L. & Auckland, P. Super-resolution microscopy as drug discovery tool. *SLAS Discov*, 100209 (2025). https://doi.org/10.1016/j.slasd.2025.100209

132 Chen, Z. *et al.* Quantitative analysis of multiple breast cancer biomarkers using DNA-PAINT. *Anal Methods* **14**, 3671-3679 (2022). https://doi.org/10.1039/d2ay00670g

133 Nerreter, T. *et al.* Super-resolution microscopy reveals ultra-low CD19 expression on myeloma cells that triggers elimination by CD19 CAR-T. *Nat Commun* **10**, 3137 (2019). https://doi.org/10.1038/s41467-019-10948-w

134 Chen, C. *et al.* Profiling of Exosomal Biomarkers for Accurate Cancer Identification: Combining DNA-PAINT with Machine- Learning-Based Classification. *Small* **15**, e1901014 (2019). https://doi.org/10.1002/smll.201901014

135 Panconi, L., Owen, D. M. & Griffie, J. Cluster analysis for localisation-based data sets: dos and don'ts when quantifying protein aggregates. *Front Bioinform* **3**, 1237551 (2023). https://doi.org/10.3389/fbinf.2023.1237551

136 Carnevali, D. *et al.* A deep learning method that identifies cellular heterogeneity using nanoscale nuclear features. *Nat Mach Intell* **6**, 1021-1033 (2024). https://doi.org/10.1038/s42256-024-00883-x

137 Curd, A. *et al.* Diagnosis with Nanoscale Protein Distributions: Single-Molecule Fluorescence Localization Microscopy and Attention-Based Learning in 2025 IEEE 22nd International Symposium on Biomedical Imaging (ISBI).Edition edn 1-5 (2025).

138 Sams, M. *et al.* Spatial cluster analysis of nanoscopically mapped serotonin receptors for classification of fixed brain tissue. *J Biomed Opt* **19**, 011021 (2014). https://doi.org/10.1117/1.JBO.19.1.011021

139 Woodhams, S. G., Markus, R., Gowler, P. R. W., Self, T. J. & Chapman, V. Cell type-specific super-resolution imaging reveals an increase in calcium-permeable AMPA receptors at spinal peptidergic terminals as an anatomical correlate of inflammatory pain. *Pain* **160**, 2641-2650 (2019). https://doi.org/10.1097/j.pain.0000000000001672

140 Bringsjord, S. & Govindarajulu, N. S. Given the web, what is intelligence, really? *Metaphilosophy* **43**, 464-479 (2012).

141 Bringsjord, S. & Govindarajulu, N. S. in *The Stanford Encyclopedia of Philosophy* (eds Edward N. Zalta & Uri Nodelman) (Metaphysics Research Lab, Stanford University, 2024).

142 Russell, S. & Norvig, P. *Artificial Intelligence: A Modern Approach*. 4th edn, (Pearson, 2021).

143 Brynjolfsson, E. The Turing Trap: The Promise & Peril of Human-Like Artificial Intelligence. *Daedalus* **151**, 272-287 (2022). https://doi.org/10.1162/daed_a_01915

144 Alowais, S. A. *et al.* Revolutionizing healthcare: the role of artificial intelligence in clinical practice. *BMC Med Educ* **23**, 689 (2023). https://doi.org/10.1186/s12909-023-04698-z

145 Perez-Lopez, R., Ghaffari Laleh, N., Mahmood, F. & Kather, J. N. A guide to artificial intelligence for cancer researchers. *Nat Rev Cancer* **24**, 427-441 (2024). https://doi.org/10.1038/s41568-024-00694-7

146 Goodfellow, I., Bengio, Y. & Courville, A. *Deep Learning*. (MIT Press, 2016).

147 Rajpurkar, P., Chen, E., Banerjee, O. & Topol, E. J. AI in health and medicine. *Nat Med* **28**, 31-38 (2022). https://doi.org/10.1038/s41591-021-01614-0

148 Uttam, S. *et al.* Spatial domain analysis predicts risk of colorectal cancer recurrence and infers associated tumor microenvironment networks. *Nat Commun* **11**, 3515 (2020). https://doi.org/10.1038/s41467-020-17083-x

149 Foersch, S. *et al.* Multistain deep learning for prediction of prognosis and therapy response in colorectal cancer. *Nat Med* **29**, 430-439 (2023). https://doi.org/10.1038/s41591-022-02134-1

150 Bender, S. W. B., Dreisler, M. W., Zhang, M., Kaestel-Hansen, J. & Hatzakis, N. S. SEMORE: SEgmentation and MORphological fingErprinting by machine learning automates super-resolution data analysis. *Nat Commun* **15**, 1763 (2024). https://doi.org/10.1038/s41467-024-46106-0

151 Saavedra, L. A., Mosqueira, A. & Barrantes, F. J. A supervised graph-based deep learning algorithm to detect and quantify clustered particles. *Nanoscale* **16**, 15308-15318 (2024). https://doi.org/10.1039/d4nr01944j

152 Lecun, Y., Bottou, L., Bengio, Y. & Haffner, P. Gradient-based learning applied to document recognition. *Proceedings of the IEEE* **86**, 2278-2324 (1998). https://doi.org/10.1109/5.726791

153 LeNail, A. NN-SVG: Publication-Ready Neural Network Architecture Schematics. *Journal of Open Source Software* **4**, 747 (2019). https://doi.org/10.21105/joss.00747

154 Sanchez-Lengeling, B., Reif, E., Pearce, A. & Wiltschko, A. B. A Gentle Introduction to Graph Neural Networks. *Distill* (2021). https://doi.org/10.23915/distill.00033

155 Alzubaidi, L. *et al.* Review of deep learning: concepts, CNN architectures, challenges, applications, future directions. *J Big Data* **8**, 53 (2021). https://doi.org/10.1186/s40537-021-00444-8

156 Qi, C. R., Su, H., Mo, K. & Guibas, L. J. PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation. arXiv:1612.00593 (2016). https://doi.org/10.48550/arXiv.1612.00593

157 Bahdanau, D., Cho, K. & Bengio, Y. Neural Machine Translation by Jointly Learning to Align and Translate. arXiv: 1409.0473 (2015). https://doi.org/10.48550/arXiv.1409.0473

158 Vaswani, A. *et al.* Attention is All you Need in Advances in Neural Information Processing Systems.Edition edn (eds I. Guyon *et al.*) 6000-6010 (Curran Associates, Inc., 2017).

159 Lin, T., Wang, Y., Liu, X. & Qiu, X. A survey of transformers. *AI Open* **3**, 111-132 (2022). https://doi.org/10.1016/j.aiopen.2022.10.001

160 Bello, S. A., Yu, S., Wang, C., Adam, J. M. & Li, J. Review: Deep Learning on 3D Point Clouds. *Remote Sensing* **12**, 1729 (2020).

161 Zhao, H., Jiang, L., Jia, J., Torr, P. & Koltun, V. Point Transformer in 2021 IEEE/CVF International Conference on Computer Vision (ICCV).Edition edn 16239-16248 (2021).

162 Müller, L., Galkin, M., Morris, C. & Rampášek, L. Attending to Graph Transformers. *Transactions on Machine Learning Research* (2024).

163 Danial, J. S. H. & Garcia-Saez, A. J. Quantitative analysis of super-resolved structures using ASAP. *Nat Methods* **16**, 711-714 (2019). https://doi.org/10.1038/s41592-019-0472-1

164 Hugelier, S. *et al.* ECLiPSE: a versatile classification technique for structural and morphological analysis of 2D and 3D single-molecule localization microscopy data. *Nat Methods* **21**, 1909-1915 (2024). https://doi.org/10.1038/s41592-024-02414-3

165 Li, Y. L. *et al.* SuperResNET: Model-Free Single-Molecule Network Analysis Software Achieves Molecular Resolution of Nup96. *Advanced Intelligent Systems*, 2400521 (2024). https://doi.org/10.1002/aisy.202400521

166 Khater, I. M., Meng, F., Wong, T. H., Nabi, I. R. & Hamarneh, G. Super Resolution Network Analysis Defines the Molecular Architecture of Caveolae and Caveolin-1 Scaffolds. *Sci Rep* **8**, 9009 (2018). https://doi.org/10.1038/s41598-018-27216-4

167 Khater, I. M., Meng, F., Nabi, I. R. & Hamarneh, G. Identification of caveolin-1 domain signatures via machine learning and graphlet analysis of single-molecule super-resolution data. *Bioinformatics* **35**, 3468-3475 (2019). https://doi.org/10.1093/bioinformatics/btz113

168 Khater, I. M., Liu, Q., Chou, K. C., Hamarneh, G. & Nabi, I. R. Super-resolution modularity analysis shows polyhedral caveolin-1 oligomers combine to form scaffolds and caveolae. *Sci Rep* **9**, 9888 (2019). https://doi.org/10.1038/s41598-019-46174-z

169 Wong, T. H. *et al.* Single molecule network analysis identifies structural changes to caveolae and scaffolds due to mutation of the caveolin-1 scaffolding domain. *Sci Rep* **11**, 7810 (2021). https://doi.org/10.1038/s41598-021-86770-6

170 Wong, T. H. *et al.* SuperResNET - single-molecule network analysis detects changes to clathrin structure induced by small-molecule inhibitors. *J Cell Sci* **138**, JCS263570 (2025). https://doi.org/10.1242/jcs.263570

171 Auer, A., Strauss, M. T., Strauss, S. & Jungmann, R. nanoTRON: a Picasso module for MLP-based classification of super-resolution data. *Bioinformatics* **36**, 3620-3622 (2020). https://doi.org/10.1093/bioinformatics/btaa154

172 Fan, Y. *et al.* Stacked Pointnets For Alignment Of Particles With Cylindrical Symmetry In Single Molecule Localization Microscopy in 2021 IEEE 18th International Symposium on Biomedical Imaging (ISBI).Edition edn 858-862 (2021).

173 Chowdhury, A. S., Majhi, S., Jadhav, S., Agarwal, K. & Prasad, D. K. Label Modulated Dynamic Graph Convolution for Subcellular Structure Segmentation from Nanoscopy Images in Graph-Based Representations in Pattern Recognition.Edition edn (eds Luc Brun, Vincenzo Carletti, Sébastien Bougleux, & Benoît Gaüzère) 68-78 (Springer Nature Switzerland, 2025).

174 Blanc, T. *et al.* Towards Human in the Loop Analysis of Complex Point Clouds: Advanced Visualizations, Quantifications, and Communication Features in Virtual Reality. *Front Bioinform* **1**, 775379 (2021). https://doi.org/10.3389/fbinf.2021.775379

175 Verdier, H. *et al.* Simulation-based inference for non-parametric statistical comparison of biomolecule dynamics. *PLoS Comput Biol* **19**, e1010088 (2023). https://doi.org/10.1371/journal.pcbi.1010088

176 Nabi, I. R. *et al.* AI analysis of super-resolution microscopy: Biological discovery in the absence of ground truth. *J Cell Biol* **223** (2024). https://doi.org/10.1083/jcb.202311073

177 Patrício, C., Neves, J. C. & Teixeira, L. F. Explainable Deep Learning Methods in Medical Image Classification: A Survey. *ACM Comput. Surv.* **56**, Article 85 (2023). https://doi.org/10.1145/3625287

178 Mulawade, R. N., Garth, C. & Wiebel, A. Explainable Artificial Intelligence (XAI) for Methods Working on Point Cloud Data: A Survey. *IEEE Access* **12**, 146830-146851 (2024). https://doi.org/10.1109/ACCESS.2024.3472872

179 Amara, K. *et al.* GraphFramEx: Towards Systematic Evaluation of Explainability Methods f or Graph Neural Networks Edition edn (eds Bastian Rieck & Razvan Pascanu) (PMLR, 2022).

180 Yuan, H., Yu, H., Gui, S. & Ji, S. Explainability in Graph Neural Networks: A Taxonomic Survey. *IEEE Transactions on Pattern Analysis & Machine Intelligence* **45**, 5782-5799 (2023). https://doi.org/10.1109/tpami.2022.3204236

181 Agarwal, C., Queen, O., Lakkaraju, H. & Zitnik, M. Evaluating explainability for graph neural networks. *Sci Data* **10**, 144 (2023). https://doi.org/10.1038/s41597-023-01974-x

182 Jaume, G. *et al.* Quantifying Explainers of Graph Neural Networks in Computational Pathology in *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR).* 8102-8112.

183 Ying, R., Bourgeois, D., You, J., Zitnik, M. & Leskovec, J. GNNExplainer: Generating Explanations for Graph Neural Networks. arXiv:1903.03894 (2019). https://doi.org/10.48550/arXiv.1903.03894

184 Huang, H., Wang, Y., Rudin, C. & Browne, E. P. Towards a comprehensive evaluation of dimension reduction methods for transcriptomic data visualization. *Commun Biol* **5**, 719 (2022). https://doi.org/10.1038/s42003-022-03628-x

185 Gogoberidze, N. & Cimini, B. A. Defining the boundaries: challenges and advances in identifying cells in microscopy images. *Curr Opin Biotechnol* **85**, 103055 (2024). https://doi.org/10.1016/j.copbio.2023.103055

186 Falk, T. *et al.* U-Net: deep learning for cell counting, detection, and morphometry. *Nat Methods* **16**, 67-70 (2019). https://doi.org/10.1038/s41592-018-0261-2

187 Siddique, N., Paheding, S., Elkin, C. & Devabhaktu, V. U-Net and its variants for medical image segmentation: theory and applications. arXiv.2011.01118 (2021). https://doi.org/10.48550/arXiv.2011.01118

188 Pachitariu, M. & Stringer, C. Cellpose 2.0: how to train your own model. *Nat Methods* **19**, 1634-1641 (2022). https://doi.org/10.1038/s41592-022-01663-4

189 Berg, S. *et al.* ilastik: interactive machine learning for (bio)image analysis. *Nat Methods* **16**, 1226-1232 (2019). https://doi.org/10.1038/s41592-019-0582-9

190 Ester, M., Kriegel, H.-P., Sander, J. & Xu, X. A density-based algorithm for discovering clusters in large spatial databases with noise in Proceedings of the Second International Conference on Knowledge Discovery and Data Mining.Edition edn 226–231 (AAAI Press, 1996).

191 Nieves, D. J. & Owen, D. M. Analysis methods for interrogating spatial organisation of single molecule localisation microscopy data. *Int J Biochem Cell Biol* **123**, 105749 (2020). https://doi.org/10.1016/j.biocel.2020.105749

192 Pike, J. A. *et al.* Topological data analysis quantifies biological nano-structure from single molecule localization microscopy. *Bioinformatics* **36**, 1614-1621 (2020). https://doi.org/10.1093/bioinformatics/btz788

193 Nieves, D. J. *et al.* A framework for evaluating the performance of SMLM cluster analysis algorithms. *Nat Methods* **20**, 259-267 (2023). https://doi.org/10.1038/s41592-022-01750-6

194 Sieben, C., Banterle, N., Douglass, K. M., Gonczy, P. & Manley, S. Multicolor single-particle reconstruction of protein complexes. *Nat Methods* **15**, 777-780 (2018). https://doi.org/10.1038/s41592-018-0140-x

195 Marin, Z., Fuentes, L. A., Bewersdorf, J. & Baddeley, D. Extracting nanoscale membrane morphology from single-molecule localizations. *Biophys J* **122**, 3022-3030 (2023). https://doi.org/10.1016/j.bpj.2023.06.010

196 Ronneberger, O., Fischer, P. & Brox, T. in *Medical Image Computing and Computer-Assisted Intervention - MICCAI 2015* (eds N. Navab, J. Hornegger, W.M. Wells, & A.F. Frangi) 234–241 (Cham: Springer International Publishing, 2015).

197 Guerrero-Peña, F. A. *et al.* Multiclass Weighted Loss for Instance Segmentation of Cluttered Cells in *2018 25th IEEE International Conference on Image Processing (ICIP).* 2451-2455.

198 Caballero-Ruiz, B. *et al.* Partial Truncation of the C-Terminal Domain of PTCH1 in Cancer Enhances Autophagy and Metabolic Adaptability. *Cancers (Basel)* **15** (2023). https://doi.org/10.3390/cancers15020369

199 ONI. *Analysis: Filter*, <https://onidesk.zohodesk.eu/portal/en/kb/articles/analysis-filter> (2025).

200 Willems, J., Westra, M. & MacGillavry, H. D. in *Fluorescent Microscopy* (ed Bryan Heit) 271-288 (Springer US, 2022).

201 Schnitzbauer, J., Strauss, M. T., Schlichthaerle, T., Schueder, F. & Jungmann, R. Super-resolution microscopy with DNA-PAINT. *Nat Protoc* **12**, 1198-1228 (2017). https://doi.org/10.1038/nprot.2017.024

202 Andronov, L., Lutz, Y., Vonesch, J. L. & Klaholz, B. P. SharpViSu: integrated analysis and segmentation of super-resolution microscopy data. *Bioinformatics* **32**, 2239-2241 (2016). https://doi.org/10.1093/bioinformatics/btw123

203 Duim, W. C., Jiang, Y., Shen, K., Frydman, J. & Moerner, W. E. Super-resolution fluorescence of huntingtin reveals growth of globular species into short fibers and coexistence of distinct aggregates. *ACS Chem Biol* **9**, 2767-2778 (2014). https://doi.org/10.1021/cb500335w

204 Ahlers, J. *et al. Napari: a multi-dimensional image viewer for Python (v0.4.18). doi: 10.5281/zenodo.8115575*, <https://napari.org> (2022).

205 Kingma, D. P. & Ba, J. Adam: A Method for Stochastic Optimization in 3rd International Conference on Learning Representations.Edition edn (eds Yoshua Bengio & Yann LeCun) (2015).

206 Stringer, C., Wang, T., Michaelos, M. & Pachitariu, M. Cellpose: a generalist algorithm for cellular segmentation. *Nat Methods* **18**, 100-106 (2021). https://doi.org/10.1038/s41592-020-01018-x

207 He, H. & Garcia, E. A. Learning from Imbalanced Data. *IEEE Transactions on Knowledge and Data Engineering* **21**, 1263-1284 (2009). https://doi.org/10.1109/TKDE.2008.239

208 Tharwat, A. Classification assessment methods. *Applied Computing and Informatics* **17**, 168–192 (2021). https://doi.org/10.1016/j.aci.2018.08.003

209 Boyd, K., Santos Costa, V., Davis, J. & Page, C. D. Unachievable Region in Precision-Recall Space and Its Effect on Empirical Evaluation. *Proc Int Conf Mach Learn* **2012**, 349 (2012).

210 Maxwell, A. E., Warner, T. A. & Guillén, L. A. Accuracy Assessment in Convolutional Neural Network-Based Deep Learning Remote Sensing Studies—Part 1: Literature Review. *Remote Sensing.* **13** (2021). https://doi.org/10.3390/rs13132450

211 van der Walt, S. J. *et al.* scikit-image: Image processing in Python. *PeerJ* **2**, e453 (2014). https://doi.org/10.7717/peerj.453

212 Umney, O. *locpix: Analysis notebook*, <https://github.com/oubino/locpix/blob/main/examples/c15_data_ds_analysis/analysis.ipynb> (2025).

213 Levine, B. G., Stone, J. E. & Kohlmeyer, A. Fast Analysis of Molecular Dynamics Trajectories with Graphics Processing Units-Radial Distribution Function Histogramming. *J Comput Phys* **230**, 3556-3569 (2011). https://doi.org/10.1016/j.jcp.2011.01.048

214 Curd, A. P. *et al.* Nanoscale Pattern Extraction from Relative Positions of Sparse 3D Localizations. *Nano Lett* **21**, 1213-1220 (2021). https://doi.org/10.1021/acs.nanolett.0c03332

215 Otsu, N. Threshold Selection Method from Gray-Level Histograms. *IEEE Transactions on Systems, Man, and Cybernetics.* **9**, 62-66 (1979). https://doi.org/10.1109/TSMC.1979.4310076

216 Lee, S. U., Yoon, C. S. & Park, R. H. A comparative performance study of several global thresholding techniques for segmentation. *Computer Vision, Graphics, and Image Processing.* **52**, 171–190 (1990). https://doi.org/10.1016/0734-189X(90)90053-X

217 Poletti, E., Zappelli, F., Ruggeri, A. & Grisan, E. A review of thresholding strategies applied to human chromosome segmentation. *Computer Methods and Programs in Biomedicine.* **108** 679–688 (2012). https://doi.org/10.1016/j.cmpb.2011.12.003

218 Zhang, M., Zhang, L. & Cheng, H. A neutrosophic approach to image segmentation based on watershed method. *Signal Processing.* **90**, 1510-1517 (2010). https://doi.org/10.1016/j.sigpro.2009.10.021

219 Beucher, S. Use of watersheds in contour detection. *Proceedings of the International Workshop on Image Processing. CCETT;* (1979).

220 Jalalifar, S. A., Soliman, H., Sahgal, A. & Sadeghi-Naini, A. Impact of Tumour Segmentation Accuracy on Efficacy of Quantitative MRI Biomarkers of Radiotherapy Outcome in Brain Metastasis. *Cancers (Basel)* **14** (2022). https://doi.org/10.3390/cancers14205133

221 Baker, G. J. *et al.* Quality control for single-cell analysis of high-plex tissue profiles using CyLinter. *Nat Methods* **21**, 2248-2259 (2024). https://doi.org/10.1038/s41592-024-02328-0

222 Lee, E. *et al.* ESQmodel: biologically informed evaluation of 2-D cell segmentation quality in multiplexed tissue images. *Bioinformatics* **40** (2024). https://doi.org/10.1093/bioinformatics/btad783

223 Wang, Y. *et al.* A systematic evaluation of computational methods for cell segmentation. *Brief Bioinform* **25** (2024). https://doi.org/10.1093/bib/bbae407

224 He, K., Gkioxari, G., Dollár, P. & Girshick, R. Mask R-CNN. *arXiv:1703.06870* (2017). https://doi.org/10.48550/arXiv.1703.06870

225 Liu, Z. *et al.* Swin Transformer: Hierarchical Vision Transformer using Shifted Windows. *arXiv:2103.14030* (2021). https://doi.org/10.48550/arXiv.2103.14030

226 Israel, U. *et al.* CellSAM: A Foundation Model for Cell Segmentation. *bioRxiv* (2025). https://doi.org/10.1101/2023.11.17.567630

227 Sabinina, V. J. *et al.* Three-dimensional superresolution fluorescence microscopy maps the variable molecular architecture of the nuclear pore complex. *Mol Biol Cell* **32**, 1523-1533 (2021). https://doi.org/10.1091/mbc.E20-11-0728

228 Huijben, T. *et al.* Detecting structural heterogeneity in single-molecule localization microscopy data. *Nat Commun* **12**, 3791 (2021). https://doi.org/10.1038/s41467-021-24106-8

229 Huijben, T. A. P. M. *et al.* (4TU.ResearchData doi:10.4121/14074091.v1, 2021).

230 Wu, X., Lao, Y., Jiang, L., Liu, X. & Zhao, H. Point Transformer V2: Grouped Vector Attention and Partition-based Pooling. arXiv:2210.05666 (2022). https://doi.org/10.48550/arXiv.2210.05666

231 Demantké, J., Mallet, C., David, N. & Vallet, B. Dimensionality Based Scale Selection in 3d LIDAR Point Clouds. *ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences* **3812**, 97-102 (2011). https://doi.org/10.5194/isprsarchives-XXXVIII-5-W12-97-2011

232 Baddeley, D. *et al.* Optical single-channel resolution imaging of the ryanodine receptor distribution in rat cardiac myocytes. *Proc Natl Acad Sci U S A* **106**, 22275-22280 (2009). https://doi.org/10.1073/pnas.0908971106

233 Fey, M. & Lenssen, J. E. Fast Graph Representation Learning with PyTorch Geometric. *arXiv e-prints*, arXiv:1903.02428 (2019). https://doi.org/10.48550/arXiv.1903.02428

234 Xu, K., Hu, W., Leskovec, J. & Jegelka, S. How Powerful are Graph Neural Networks? , arXiv:1810.00826 (2018). https://doi.org/10.48550/arXiv.1810.00826

235 Maier-Hein, L. *et al.* Metrics reloaded: recommendations for image analysis validation. *Nat Methods* **21**, 195-212 (2024). https://doi.org/10.1038/s41592-023-02151-z

236 McInnes, L., Healy, J. & Melville, J. UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction. arXiv:1802.03426 (2018). https://doi.org/10.48550/arXiv.1802.03426

237 Suarez-Barrera, D. *et al.* Efficient and reliable spike sorting from neural recordings with UMAP-based unsupervised nonlinear dimensionality reduction. *PLoS Biol* **23**, e3003527 (2025). https://doi.org/10.1371/journal.pbio.3003527

238 Yuan, H., Yu, H., Wang, J., Li, K. & Ji, S. On Explainability of Graph Neural Networks via Subgraph Explorations. arXiv:2102.05152 (2021). https://doi.org/10.48550/arXiv.2102.05152

239 Shapley, L. S. in *Contributions to the Theory of Games, Volume II* (eds Harold William Kuhn & Albert William Tucker) 307-318 (Princeton University Press, 1953).

240 Liu, M. *et al.* DIG: A Turnkey Library for Diving into Graph Deep Learning Research. *Journal of Machine Learning Research* **22**, 10873-10881 (2021).

241 Clayton, A. H. *et al.* Ligand-induced dimer-tetramer transition during the activation of the cell surface epidermal growth factor receptor-A multidimensional microscopy analysis. *J Biol Chem* **280**, 30392-30399 (2005). https://doi.org/10.1074/jbc.M504770200

242 Ibach, J. *et al.* Single Particle Tracking Reveals that EGFR Signaling Activity Is Amplified in Clathrin-Coated Pits. *PLoS One* **10**, e0143162 (2015). https://doi.org/10.1371/journal.pone.0143162

243 Middleton, G. *et al.* A randomised phase III trial of the pharmacokinetic biomodulation of irinotecan using oral ciclosporin in advanced colorectal cancer: results of the Panitumumab, Irinotecan & Ciclosporin in COLOrectal cancer therapy trial (PICCOLO). *Eur J Cancer* **49**, 3507-3516 (2013). https://doi.org/10.1016/j.ejca.2013.06.017

244 Eisenhauer, E. A. *et al.* New response evaluation criteria in solid tumours: revised RECIST guideline (version 1.1). *Eur J Cancer* **45**, 228-247 (2009). https://doi.org/10.1016/j.ejca.2008.10.026

245 ONI. *How to guide: Filtering tool*, <https://help.codi.bio/portal/en/kb/articles/how-to-guide-filtering-tool> (2025).

246 Umney, O. *et al.* Annotation and automated segmentation of single-molecule localisation microscopy data. *J Microsc* **296**, 214-226 (2024). https://doi.org/10.1111/jmi.13349

247 Stoltzfus, J. C. Logistic regression: a brief primer. *Acad Emerg Med* **18**, 1099-1104 (2011). https://doi.org/10.1111/j.1553-2712.2011.01185.x

248 Breiman, L. Random Forests. *Machine Learning* **45**, 5-32 (2001). https://doi.org/10.1023/A:1010933404324

249 Pedregosa, F. *et al.* Scikit-learn: Machine Learning in Python. *J. Mach. Learn. Res.* **12**, 2825–2830 (2011).

250 Zou, H. & Hastie, T. Regularization and Variable Selection Via the Elastic Net. *Journal of the Royal Statistical Society Series B: Statistical Methodology* **67**, 301-320 (2005). https://doi.org/10.1111/j.1467-9868.2005.00503.x

251 Murphy, K. P. *Probabilistic Machine Learning: An introduction*. (MIT Press, 2022).

252 Strobl, C., Boulesteix, A.-L., Kneib, T., Augustin, T. & Zeileis, A. Conditional variable importance for random forests. *BMC Bioinformatics* **9**, 307 (2008). https://doi.org/10.1186/1471-2105-9-307

253 Grömping, U. Variable importance in regression models. *WIREs Computational Statistics* **7**, 137-152 (2015). https://doi.org/10.1002/wics.1346

254 Louppe, G., Wehenkel, L., Sutera, A. & Geurts, P. Understanding variable importances in forests of randomized trees in Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 1.Edition edn 431–439 (Curran Associates Inc., 2013).

255 Lundberg, S. M. & Lee, S.-I. A Unified Approach to Interpreting Model Predictions in Advances in Neural Information Processing Systems.Edition edn (eds I. Guyon *et al.*) 4768 - 4777 (2017).

256 Rodriguez-Perez, R. & Bajorath, J. Interpretation of machine learning models using shapley values: application to compound potency and multi-target activity predictions. *J Comput Aided Mol Des* **34**, 1013-1026 (2020). https://doi.org/10.1007/s10822-020-00314-0

257 Lundberg, S. M. *et al.* From local explanations to global understanding with explainable AI for trees. *Nature Machine Intelligence* **2**, 56-67 (2020). https://doi.org/10.1038/s42256-019-0138-9

258 Salih, A. M. *et al.* A Perspective on Explainable Artificial Intelligence Methods: SHAP and LIME. *Advanced Intelligent Systems* **7**, 2400304 (2025). https://doi.org/10.1002/aisy.202400304

259 Li, Y., Tarlow, D., Brockschmidt, M. & Zemel, R. S. Gated Graph Sequence Neural Networks in 4th International Conference on Learning Representations.Edition edn (eds Yoshua Bengio & Yann LeCun) (2016).

260 Li, Y., Gu, C., Dullien, T., Vinyals, O. & Kohli, P. Graph Matching Networks for Learning the Similarity of Graph Structured Objects in Proceedings of the 36th International Conference on Machine Learning.Edition edn (eds Kamalika Chaudhuri & Ruslan Salakhutdinov) 3835-3845 (2019).

261 Huang, Y.-Z. *MLstatkit*, <https://pypi.org/project/MLstatkit/> (2025).

262 Dormann, C. F. Calibration of probability predictions from machine-learning and statistical models. *Global Ecology and Biogeography* **29**, 760-765 (2020). https://doi.org/https://doi.org/10.1111/geb.13070

263 Lu, X., Lu, X. & Kang, Y. Organ-specific enhancement of metastasis by spontaneous ploidy duplication and cell size enlargement. *Cell Res* **20**, 1012-1022 (2010). https://doi.org/10.1038/cr.2010.93

264 Krajcovic, M. *et al.* A non-genetic route to aneuploidy in human cancers. *Nat Cell Biol* **13**, 324-330 (2011). https://doi.org/10.1038/ncb2174

265 Krajcovic, M. & Overholtzer, M. Mechanisms of ploidy increase in human cancers: a new role for cell cannibalism. *Cancer Res* **72**, 1596-1601 (2012). https://doi.org/10.1158/0008-5472.CAN-11-3127

266 Ferkel, S. A. M., Holman, E. A., Sojwal, R. S., Rubin, S. J. S. & Rogalla, S. Tumor-Infiltrating Immune Cells in Colorectal Cancer. *Neoplasia* **59**, 101091 (2025). https://doi.org/10.1016/j.neo.2024.101091

267 Sturmfels, P., Lundberg, S. & Lee, S.-I. Visualizing the Impact of Feature Attribution Baselines. *Distill* (2020). https://doi.org/10.23915/distill.00022

268 Seligmann, J. F. *et al.* Combined Epiregulin and Amphiregulin Expression Levels as a Predictive Biomarker for Panitumumab Therapy Benefit or Lack of Benefit in Patients With RAS Wild-Type Advanced Colorectal Cancer. *JAMA Oncol* **2**, 633-642 (2016). https://doi.org/10.1001/jamaoncol.2015.6065

269 Kim, J. *et al.* Oblique-plane single-molecule localization microscopy for tissues and small intact animals. *Nat Methods* **16**, 853-857 (2019). https://doi.org/10.1038/s41592-019-0510-z

270 Klevanski, M. *et al.* Automated highly multiplexed super-resolution imaging of protein nano-architecture in cells and tissues. *Nat Commun* **11**, 1552 (2020). https://doi.org/10.1038/s41467-020-15362-1

271 Cheng, J. *et al.* A single-molecule localization microscopy method for tissues reveals nonrandom nuclear pore distribution in Drosophila. *J Cell Sci* **134** (2021). https://doi.org/10.1242/jcs.259570

272 Siemons, M. E., Hanemaaijer, N. A. K., Kole, M. H. P. & Kapitein, L. C. Robust adaptive optics for localization microscopy deep in complex tissue. *Nat Commun* **12**, 3407 (2021). https://doi.org/10.1038/s41467-021-23647-2

273 Narayanasamy, K. K., Rahm, J. V., Tourani, S. & Heilemann, M. Fast DNA-PAINT imaging using a deep neural network. *Nat Commun* **13**, 5047 (2022). https://doi.org/10.1038/s41467-022-32626-0

274 Villegas-Hernandez, L. E. *et al.* Chip-based multimodal super-resolution microscopy for histological investigations of cryopreserved tissue sections. *Light Sci Appl* **11**, 43 (2022). https://doi.org/10.1038/s41377-022-00731-w

275 Zhang, P. *et al.* Deep learning-driven adaptive optics for single-molecule localization microscopy. *Nat Methods* **20**, 1748-1758 (2023). https://doi.org/10.1038/s41592-023-02029-0

276 Park, S. *et al.* Label-free adaptive optics single-molecule localization microscopy for whole zebrafish. *Nat Commun* **14**, 4185 (2023). https://doi.org/10.1038/s41467-023-39896-2

277 Kusumi, A. & Suzuki, K. Toward understanding the dynamics of membrane-raft-based molecular interactions. *Biochim Biophys Acta* **1746**, 234-251 (2005). https://doi.org/10.1016/j.bbamcr.2005.10.001

278 Whelan, D. R. & Bell, T. D. Image artifacts in single molecule localization microscopy: why optimization of sample preparation protocols matters. *Sci Rep* **5**, 7924 (2015). https://doi.org/10.1038/srep07924

279 Pereira, P. M. *et al.* Fix Your Membrane Receptor Imaging: Actin Cytoskeleton and CD4 Membrane Organization Disruption by Chemical Fixation. *Front Immunol* **10**, 675 (2019). https://doi.org/10.3389/fimmu.2019.00675

280 Zhang, N. *et al.* Graph-Based Spatial Proximity of Super-Resolved Protein-Protein Interactions Predicts Cancer Drug Responses in Single Cells. *Cell Mol Bioeng* **17**, 467-490 (2024). https://doi.org/10.1007/s12195-024-00822-1

281 Lee, J., Lee, I. & Kang, J. Self-Attention Graph Pooling in Proceedings of the 36th International Conference on Machine Learning.Edition edn (eds Kamalika Chaudhuri & Ruslan Salakhutdinov) 3734-3743 (2019).

282 Knyazev, B., Taylor, G. W. & Amer, M. R. Understanding Attention and Generalization in Graph Neural Networks in Proceedings of the 33rd International Conference on Neural Information Processing Systems.Edition edn 378 (2019).

283 Itoh, T. D., Kubo, T. & Ikeda, K. Multi-Level Attention Pooling for Graph Neural Networks: Unifying Graph Representations with Multiple Localities. *Neural Networks* **145**, 356-373 (2022). https://doi.org/10.1016/j.neunet.2021.11.001

284 Zhao, Z. *et al.* ENADPool: The Edge-Node Attention-based Differentiable Pooling for Graph Neural Networks. *arXiv e-prints*, arXiv:2405.10218 (2024). https://doi.org/10.48550/arXiv.2405.10218

285 Baranwal, M., Krishnan, S., Oneka, M., Frankel, T. & Rao, A. CGAT: Cell Graph ATtention Network for Grading of Pancreatic Disease Histology Images. *Front Immunol* **12**, 727610 (2021). https://doi.org/10.3389/fimmu.2021.727610

286 He, K., Zhang, X., Ren, S. & Sun, J. Deep Residual Learning for Image Recognition. *arXiv:1512.03385* (2015). https://doi.org/10.48550/arXiv.1512.03385

287 Gatys, L. A., Ecker, A. S. & Bethge, M. Image Style Transfer Using Convolutional Neural Networks. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2414–2423 (2016). https://doi.org/10.1109/CVPR.2016.265

288 Hollandi, R. *et al.* nucleAIzer: A Parameter-free Deep Learning Framework for Nucleus Segmentation Using Image Style Transfer. *Cell Syst* **10**, 453-458 e456 (2020). https://doi.org/10.1016/j.cels.2020.04.003

289 Linardatos, P., Papastefanopoulos, V. & Kotsiantis, S. Explainable AI: A Review of Machine Learning Interpretability Methods. *Entropy (Basel)* **23** (2020). https://doi.org/10.3390/e23010018

290 Angelov, P. P., Soares, E. A., Jiang, R., Arnold, N. I. & Atkinson, P. M. Explainable artificial intelligence: an analytical review. *WIREs Data Mining and Knowledge Discovery* **11**, e1424 (2021). https://doi.org/10.1002/widm.1424

291 Saranya, A. & Subhashini, R. A systematic review of Explainable Artificial Intelligence models and applications: Recent developments and future trends. *Decision Analytics Journal* **7**, 100230 (2023). https://doi.org/10.1016/j.dajour.2023.100230

292 Hassija, V. *et al.* Interpreting Black-Box Models: A Review on Explainable Artificial Intelligence. *Cognitive Computation* **16**, 45-74 (2024). https://doi.org/10.1007/s12559-023-10179-8

293 Kalasampath, K. *et al.* A Literature Review on Applications of Explainable Artificial Intelligence (XAI). *IEEE Access* **13**, 41111-41140 (2025). https://doi.org/10.1109/ACCESS.2025.3546681

294 Chen, H., Lundberg, S. & Lee, S.-I. *Understanding Shapley value explanation algorithms for trees*, <https://hughchen.github.io/its_blog/> (2022).

295 Meena, J. & Hasija, Y. Application of explainable artificial intelligence in the identification of Squamous Cell Carcinoma biomarkers. *Comput Biol Med* **146**, 105505 (2022). https://doi.org/10.1016/j.compbiomed.2022.105505

7 Appendix

7.1 Extended methods

7.1.1 Performance metrics

We provide a brief introduction to the performance metrics used in this study. For more detail, please see [208,210,235].

Consider a binary classification task (e.g. Chapters 2 and 4), where the model calculates the probability that each data point belongs to the positive class (Chapter 2: positive class = membrane, data point = *xy* localisation; Chapter 4: positive class = positive response to treatment, data point = cell). If this probability is above a threshold, τ, the model predicts the data point is positive. In Chapter 4, the cells are classified according to the highest probability class, therefore, by definition, $\tau = 0.5$. For each data point, the actual label (ground truth) and the predicted label can be plotted in a confusion matrix (Table S1). This gives the number of true/false positives and negatives. These can then be used to calculate further metrics (Table S2), which range from zero (worst performance) to one (best performance), except for the false positive rate, where zero is best and one is worst. F_1 score and balanced accuracy can be used where there is a class imbalance [208].

Table S1. Confusion matrix for binary classification.

		Actual values	
		Positive	Negative
Predicted values	Positive	True positive (TP)	False positive (FP)
	Negative	False negative (FN)	True negative (TN)

Table S2. Performance metrics for binary classification models.

Metric	**Formula**
Accuracy	$\frac{TP+TN}{TP+TN+FP+FN}$
Precision	$\frac{TP}{TP+FP}$
Recall/True positive rate (TPR)	$\frac{TP}{TP+FN}$
True negative rate (TNR)	$\frac{TN}{TN+FP}$
False positive rate (FPR)	$\frac{FP}{FP+TN}$
F_1 score	$\frac{2\times precision\times recall}{precision+recall}$
Balanced accuracy	$\frac{1}{2}\times$ (TPR + TNR)

Consider a multi-class classification task (Chapter 3), where the model calculates the probability that the data point belongs to the class, for each of the classes. Per-class metrics are calculated by treating each class in turn as the positive class and the remaining classes as the negative class. Calculation of the metrics then proceeds as for the binary classification task. When calculating a single metric for all the classes, the metrics differ from the binary case. For example, balanced accuracy is now

defined as the arithmetic mean of the recall for each class. Further, while accuracy still gives the ratio between the number of correctly classified samples and the total number of samples, it is now calculated by taking the sum of the diagonal entries in the confusion matrix and dividing by the total number of samples.

The probabilities generated by the model can also be evaluated, often via a receiver operating characteristic (ROC) or precision-recall (PR) curve. To plot a ROC curve for a binary classification task, τ is varied between zero and one, and the TPR/FPR is evaluated for each value of τ. The FPR values are then plotted on the x-axis and the TPR values on the y-axis. The closer the area under this curve (AUC) is to one, the better the performance. A PR curve can also be used (x-axis: recall, y-axis: precision), as it is more sensitive to changes in FP despite large TN than ROC curves [207,208]. For a multi-class classification task, these metrics are calculated on a per-class basis, where each class is taken as the positive class in turn and evaluated as in the binary case.

There are points in a PR curve that are unattainable for a given dataset. This gives rise to a minimum AUC for a PR curve (AUC_{min}) for any model [209]. Therefore, the normalised area under the PR curve is defined as,

$$AUCNPR = \frac{AUC - AUC_{min}}{AUC_{max} - AUC_{min}} \tag{7.1}$$

, where AUC_{max} is the maximum AUC (1 in our case) and

$$AUC_{min} = 1 + \frac{(1-\pi)\,ln(1-\pi)}{\pi} \tag{7.2}$$

, where $\pi = \frac{P}{N}$, N is the total number of negative samples, and P is the total number of positive samples [209]. This varies from zero (worst) to one (best). Performance is compared to the baseline, a classifier which predicts all of the localisations as the positive class [210].

7.1.2 Cellpose architecture

Cellpose has the same architecture as the standard UNET, with some key differences [206]. Standard U-Net concatenates features across skip connections, whereas Cellpose adds the features in the corresponding channels, reducing the number of parameters. Cellpose also replaces the standard U-Net blocks with residual blocks [286]. Residual blocks, like standard U-Net blocks, consist of convolutional layers, batch normalisation, and an activation function. However, while the standard U-Net block attempts to learn the underlying mapping H(x), where x is the input to the block, the residual block adds g(x) to its output, where g(·) is either the identity function or a 1x1 convolution. Therefore, this layer attempts to learn f(x) = H(x) – g(x), which is easier to train and improves accuracy for deeper networks [286]. Finally, Cellpose learns a style vector for each image, hypothesising that images fall into different style categories, which should be approached differently [287,288]. In their study, these changes improved the average precision compared to standard U-Net by an average of 0.025 [206].

7.1.3 Shapley values

Inspired by the field of game theory, many explainability methods, such as SHAP (Shapley Additive explanation) use Shapley values to explain traditional ML and deep learning models [239,289-293]. For traditional ML or deep learning classification models that act on handcrafted features, SHAP can calculate the importance of each

feature by estimating its contribution to the prediction [255]. For example, for a linear model, SHAP estimates the contribution of a feature to the prediction for a sample by calculating the difference between the model's prediction with the feature present and when it is replaced by the average value of that feature [255]. For ensembles of trees (e.g. random forest), Interventional Tree SHAP calculates the contribution of each feature by calculating the difference between when the feature is present and when it is absent [294]. The SHAP values from each sample in the dataset can then be aggregated and visualised to indicate the global impact of each feature on the classification task [258]. This analysis has been applied to medical tasks, for example, to identify genes associated with the progression of skin cancer [295]. Shapley values have also been used to explain graph neural networks, by identifying the most important connected subgraph for classification of a graph (SubgraphX) [238].

7.2 Extended results of *ClusterNet* on the digits and letters dataset

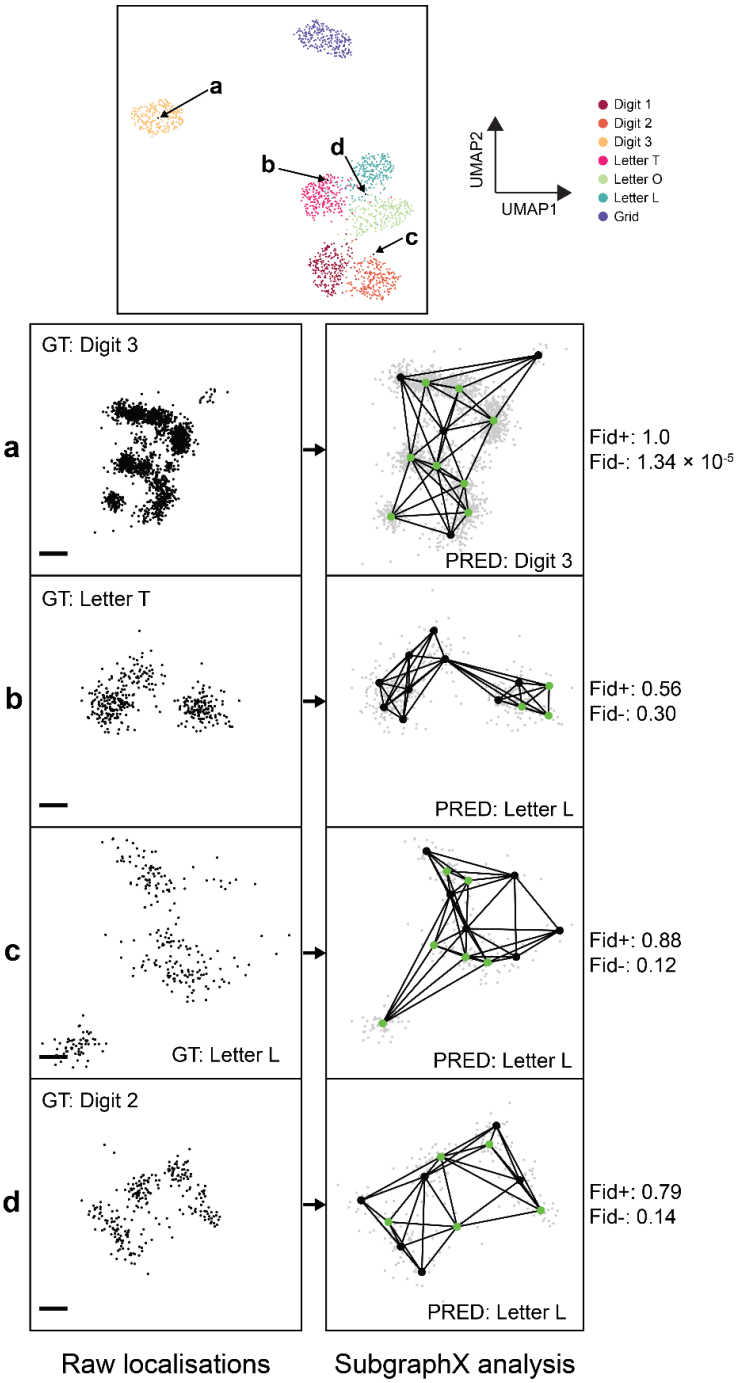

Figure S1. Structure analysis of SMLM ROI classification by *ClusterNet-LCF*. 2D whole-graph feature representation (from UMAP) for *ClusterNet-LCF* for the reserved test dataset ROIs (Top), highlighting whole-graph features (from their classified cluster graph) from four selected ROIs (**a-d**). (**a-d**) DNA-PAINT localisations (scale: 13 nm) represented as a graph, classified by *ClusterNet-LCF,* and analysed with SubgraphX (GT: ground truth, PRED: prediction). SubgraphX results show the important subgraph (supra-cluster structure) for class prediction (green nodes). Positive fidelity (Fid+) and negative fidelity (Fid-) measure the necessity and sufficiency, respectively, of the important subgraph (best performance: Fid+ = 1, Fid- = 0; worst performance: Fid+ = 0 and Fid- = 1).

Table S3. Performance of *ClusterNet-HCF* and *ClusterNet-LCF* on the training sets from *k*-fold training. Results for each class are averaged over the training sets. AUROC = area under the receiver operator curve.

Model	Metric	Digit 1	Digit 2	Digit 3	Letter T	Letter O	Letter L	Grid	Mean ± S.D.
ClusterNet-HCF	Recall	0.98	0.96	1.00	0.99	1.00	1.00	0.99	0.99 ± 0.01
ClusterNet-LCF		0.97	0.97	1.00	0.97	0.98	0.95	1.00	0.98 ± 0.02
ClusterNet-HCF	AUROC	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00 ± 0.00
ClusterNet-LCF		1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00 ± 0.00

Table S4. Performance of *ClusterNet-HCF* and *ClusterNet-LCF* on the validation sets from *k*-fold training. Results for each class are averaged over the validation sets. AUROC = area under the receiver operator curve.

Model	Metric	Digit 1	Digit 2	Digit 3	Letter T	Letter O	Letter L	Grid	Mean ± S.D.
ClusterNet-HCF	Recall	0.98	0.97	1.00	0.97	0.95	0.99	0.99	0.98 ± 0.02
ClusterNet-LCF		0.96	0.97	1.00	0.93	0.88	0.92	1.00	0.95 ± 0.04
ClusterNet-HCF	AUROC	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00 ± 0.00
ClusterNet-LCF		1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00 ± 0.00

Table S5. Performance of *ClusterNet-HCF* and *ClusterNet-LCF* on the test sets from *k*-fold training. Results for each class are averaged over the test sets. AUROC = area under the receiver operator curve.

Model	Metric	Digit 1	Digit 2	Digit 3	Letter T	Letter O	Letter L	Grid	Mean ± S.D.
ClusterNet-HCF	Recall	0.98	0.96	1.00	0.97	0.94	0.99	0.99	0.98 ± 0.02
ClusterNet-LCF		0.96	0.96	1.00	0.92	0.90	0.91	0.99	0.95 ± 0.04
ClusterNet-HCF	AUROC	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00 ± 0.00
ClusterNet-LCF		1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00 ± 0.00

Table S6. Confusion matrix for *ClusterNet-LCF* on the reserved test set.

		Predicted						
		Digit 1	Digit 2	Digit 3	Letter T	Letter O	Letter L	Grid
Actual	Digit 1	233	4	0	1	2	0	0
	Digit 2	11	226	0	0	2	1	0
	Digit 3	0	0	239	0	0	0	1
	Letter T	1	0	0	224	5	10	0
	Letter O	2	0	0	5	227	6	0
	Letter L	0	0	0	14	7	219	0
	Grid	0	1	0	0	0	2	237

Table S7. Confusion matrix for *ClusterNet-HCF* on the reserved test set.

		Predicted						
		Digit 1	Digit 2	Digit 3	Letter T	Letter O	Letter L	Grid
Actual	Digit 1	238	2	0	0	0	0	0
	Digit 2	5	234	0	1	0	0	0
	Digit 3	0	0	240	0	0	0	0
	Letter T	3	1	0	231	3	1	1
	Letter O	0	0	0	1	238	1	0
	Letter L	0	0	0	0	2	238	0
	Grid	0	1	0	0	0	0	239

7.3 Ethics approval

Ethical approval for the work involving human tissue was granted by the North East York Research Ethics Committee (08/H0903/62). The study includes patients who, at the point of consent for the PICCOLO trial, had agreed to the use of their archival tissue in future research and for whom adequate stored formalin-fixed, paraffin-embedded (FFPE) tumour tissue remained. The ethics for PICCOLO trial consent and use in research is COREC 06/Q0906/38.

7.4 Data availability

For the development of *locpix* (Chapter 2), the raw Apache parquet files and the processed data required for downstream analysis are both available in the *locpix* GitHub repository at https://github.com/oubino/locpix/tree/main/examples/c15_data/raw and https://github.com/oubino/locpix/tree/main/examples/c15_data_ds_analysis, respectively.

For the development of *ClusterNet* (Chapter 3), the original Digits and Letters dataset is available at https://doi.org/10.4121/14074091 [229] and the files for reproducing the results on this dataset including the processed data and models, can be found at https://doi.org/10.5281/zenodo.14246303.

The filtered *k*-fold dataset of EREG localisations in cells from advanced colorectal cancer patients with the response to treatment ground-truth labels (Chapter 4), can be found at https://doi.org/10.5281/zenodo.17019520.

7.5 Code availability

For Chapter 2, *locpix* can be found at https://github.com/oubino/locpix and is installable via the Python Package Index (https://pypi.org/project/locpix/). The manual annotation tool is available as a napari plugin from https://www.napari-hub.org/plugins/napari-locpix. The modified Cellpose training script is also available at https://github.com/oubino/cellpose. The downstream analysis is available as a Jupyter notebook in the *locpix* GitHub repository at https://github.com/oubino/locpix/blob/main/examples/c15_data_ds_analysis/analysis.ipynb. For more details on the commands used to produce the analysis, please see the README in the *locpix* repository https://github.com/oubino/locpix.

For Chapter 3, all code used for analysis was written in Python and is available at https://github.com/oubino/locpix_points/tree/v0.0.1, with latest developments available at https://github.com/oubino/locpix_points.