Title: UniFORM: Towards Universal Immunofluorescence Normalization for Multiplex Tissue Imaging

Authors: Kunlun Wang[1], Kaoutar Ait-Ahmad[1], Samuel D Kupp[1], Zachary Sims[2], Eric Cramer[2], Zeynep Sayar[1], Jessica Yu[1], Melissa H. Wong[3,4], Gordon B. Mills[4], S. Ece Eksi[1,4,#], Young Hwan Chang[2,4,#]

Affiliation:

1. Cancer Early Detection Advanced Research (CEDAR), Knight Cancer Institute, Oregon Health & Science University (OHSU), Portland, OR, USA
2. Department of Biomedical Engineering and Computational Biology Program, OHSU, Portland, OR, USA
3. Department of Cell, Developmental and Cancer Biology, OHSU, Portland, OR, USA
4. Knight Cancer Institute, OHSU, Portland, OR, USA

Co-corresponding author:

S. Ece Eksi, PhD
Assistant Professor
Cancer Early Detection Advanced Research (CEDAR)
Knight Cancer Institute
Oregon Health and Science University
Portland, Oregon
Email: eksi@ohsu.edu

Young Hwan Chang, PhD
Associate Professor
Department of Biomedical Engineering and Computational Biology Program
Knight Cancer Institute
Oregon Health and Science University
Portland, Oregon
Email: chanyo@ohsu.edu

Abstract

Multiplexed tissue imaging (MTI) technologies enable high-dimensional spatial analysis of tumor microenvironments but face challenges with technical variability in staining intensities. Existing normalization methods, including z-score, ComBat, and MxNorm, often fail to account for the heterogeneous, right-skewed expression patterns of MTI data, compromising signal alignment and downstream analyses. We present UniFORM, a non-parametric, Python-based pipeline for normalizing both feature- and pixel-level MTI data. UniFORM preserves marker intensity distribution shapes while maintaining positive population proportions without prior assumptions and uses automated normalization. Benchmarking across two distinct MTI platforms and datasets demonstrates that UniFORM outperforms existing methods in mitigating batch effects while maintaining biological signal fidelity. This is evidenced by improved marker distribution alignment, enhanced kBET scores, and improved downstream analyses such as UMAP visualizations and Leiden clustering. UniFORM also introduces a novel guided fine-tuning option for complex and heterogeneous datasets. Although optimized for fluorescence-based platforms, UniFORM provides a scalable and robust solution for MTI data normalization, enabling accurate and biologically meaningful interpretations.

Introduction

Multiplexed tissue imaging (MTI) technologies, including cyclic immunofluorescence (CyCIF)[1], multiplexed immunohistochemistry (IHC)[2–4], and CODEX[5], have significantly advanced our ability to visualize and analyze the complexities of the tumor microenvironment at both single-cell and subcellular levels[6]. These techniques enable the simultaneous detection of multiple markers within a single tissue section, offering a detailed view of cellular interactions and functional states[6]. However, a major challenge remains in integrating data from various samples due to technical variations in staining intensities[7], which can arise from differences in tissue fixation, antibody concentrations, and imaging conditions[7–9]. These variations necessitate robust normalization procedures to ensure accurate biological interpretations.

MTI datasets typically exist in two formats: 1) feature-level data, which usually contains mean intensity values across markers for individual cells, and 2) pixel-level data, which comprises the raw registered multi-channel images[10]. Most existing intensity normalization methods are tailored to feature-level data, where intensity measurements of individual markers are adjusted across different samples to mitigate technical variability. Commonly used methods can be categorized into parametric and non-parametric approaches. Parametric methods assume the data follows a specific parametric distribution (e.g., Gaussian or normal distribution). Typical examples of this category include z-score[11,12] and ComBat[12,13], where z-score normalization standardizes data based on mean and standard deviation and ComBat adjusts for batch effects using empirical Bayes methods. Non-parametric methods make no explicit assumptions about the data's distribution. Some commonly used methods include MxNorm[8], which aligns intensity distributions across samples using B-spline functional data registration.

Despite their utility, these methods often fall short when applied to MTI datasets. z-score and ComBat normalization, both feature-level normalization methods, rely on the assumptions of normally or symmetrically distributed data and consistent cell composition across samples[13–15]. However, the right-skewed and heterogeneous expression patterns, driven by variations in cellular composition across tissues – typical characteristics of MTI data – violate these fundamental assumptions. Although MxNorm, a feature-level normalization method, does not require consistent cell composition, its B-spline functional data registration algorithm assumes a normally or symmetrically distributed, continuous, and smooth data profile[16]. Consequently, these methods often face challenges in accurately aligning signals, preserving the original data distribution, and maintaining the integrity of the marker positive population percentage. This

makes them poorly suited for MTI feature normalization and can negatively impact downstream analyses, such as cell phenotyping and clustering.

Meanwhile, there is growing interest in pixel-level MTI applications such as image-to-image translation or virtual staining[17–19], driven by recent advancements in deep learning. Notable examples include SHIFT[20] and MIM-CyCIF[21], both of which highlight the critical need for robust pixel-level normalization techniques to enhance performance. One of the few options is FLINO[9], an R-based script that uses a grid-based quantile normalization workflow. While it may be effective in certain cases, FLINO fails to handle the right-skewness and heterogeneity of MTI data, resulting in overcorrection or undercorrection of marker intensities, misalignment of biologically meaningful signals, and potential amplification of batch effects or local intensity differences. Moreover, FLINO is computationally inefficient and lacks the scalability to handle large-scale MTI datasets.

A significant challenge in MTI applications is the lack of a universal normalization method that effectively handles both feature-level and pixel-level data. Existing approaches tend to prioritize one level over the other, limiting their ability to provide a comprehensive solution for the complex structure of MTI datasets. This underscores the need for a novel, unified approach that normalizes data at both levels, ensuring computational efficiency while accommodating the unique characteristics of MTI datasets to enable accurate and meaningful biological interpretations.

To address this need, we present UniFORM, a non-parametric and automated MTI intensity normalization pipeline developed in Python, designed for both feature-level and pixel-level datasets (**Figure 1a**). UniFORM is not limited by any of the assumptions of existing methods, and is tailored specifically to the complex nature of MTI data. A key innovation of UniFORM lies in its novel automatic normalization approach, which aligns marker intensity distributions while preserving both distribution shapes and the proportion of positive populations across samples—without requiring prior knowledge. Although this automated workflow performs effectively in most cases, guided fine-tuning options are also available for scenarios involving markers with high noise, and heterogeneous expression pattern samples, allowing for further refinement when necessary.

We demonstrate the feasibility of UniFORM as a universal normalization method for both feature-level and pixel-level data by benchmarking it against commonly used approaches,

including MxNorm, z-score, and ComBat, across two cancer MTI datasets: CRC-ORION (colorectal cancer, 6 samples, 1 batch, ORION platform by Rarecyte, **Table S1a**) and PRAD-CyCIF (prostate cancer, 20 samples, 7 batches, CyCIF platform, **Table S1b**). We evaluated each method's performance by assessing their signal alignment, kBET score, positive population change, UMAP, and Leiden clustering[22] before and after normalization (**Figure 1b**). Our results highlight the fundamental limitations of existing methods and show that UniFORM consistently outperforms them across all evaluation metrics, underscoring its effectiveness and suitability for MTI data.

Results

Technical variations can significantly affect overall signal intensity in images, leading to inconsistencies across samples or batches. To address this, UniFORM utilizes an approach that aligns negative population peaks – those ideally representing background or baseline signals – to correct for these variations[7]. This alignment ensures that baseline signals remain consistent across samples, creating a stable foundation for comparing intensities across samples.

UniFORM's automatic pipeline is based on two assumptions:

1. **Minimal Biological Variability in Negative Population:** For any given marker, the negative population across samples should exhibit minimal biological variability, serving primarily as an indicator of technical variation rather than biological differences.
2. **Consistent Distribution Patterns in Positive/Negative Population:** Marker intensity distributions are assumed to maintain a consistent pattern in positive and negative populations across samples.

The first assumption posits that the negative peak corresponds to cells lacking marker expression, making it a reliable baseline for background signals that account for technical variations. This assumption allows the negative peak to function as a stable baseline signal, typically reflecting background fluorescence and autofluorescence rather than specific marker expression. Because this negative peak reflects technical rather than biological variability, it serves as a stable baseline. Aligning these negative peaks across samples standardizes background signals, thereby correcting for technical variations and enabling comparable intensity measurements. The second assumption reinforces this approach by expecting consistency in the distribution of intensity patterns across samples, expecting a consistent pattern in marker intensity distributions across samples, where either the negative or positive

peak consistently dominates. This consistency across samples allows us to align distributions automatically by maximizing the correlation between them, ensuring that intensity patterns remain comparable across samples. However, due to factors like tissue heterogeneity, marker specificity, or technical challenges, variations may occasionally arise. To address these situations, UniFORM includes a quick and guided fine-tuning option, allowing users to make adjustments as needed to maintain accuracy and consistency in their data.

Feature-Level Normalization: Intensity Distribution Alignment and Batch Effect Removal

The feature-level MTI dataset, derived from cell or nuclei segmentation masks applied to pixel-level MTI data (see **Methods**) comprises mean cell intensity values along with metadata and spatial information. Our feature-level normalization approach (see **Methods**) is designed to be efficient and effective. Initially, we calculate the histogram distribution of the log-transformed cell intensities for each marker, allowing us to access insights into the central tendency, variability, skewness, modality, and inter-sample variations within the dataset. These distributions are then converted into functional data curves, which are aligned to an appropriate reference distribution using a maximum cross-correlation algorithm (**Figure 1c**). This approach eliminates the need for subjective, time-consuming manual gating. Finally, the raw feature data is normalized by applying the normalization factors estimated from this alignment process (see **Methods**).

Building on this approach, we performed the automatic normalization on the CRC-ORION and PRAD-CyCIF datasets (see **Data availability**). To evaluate the effectiveness of UniFORM in comparison to other methods, we first compared the alignment of the intensity distributions before and after normalization using MxNorm, z-score, and ComBat against UniFORM. To further assess the effectiveness of batch correction, we used the kBET algorithm[23] (**see Methods**), which constructs a k-nearest neighbor graph to evaluate the batch-label distribution within subsets of neighboring cells relative to the overall batch-label distribution. This approach allows us to quantify batch effect mitigation by measuring the degree of batch mixing within the data, providing a detailed comparison of UniFORM, MxNorm, ComBat, and z-score normalization methods specifically within the PRAD-CyCIF dataset.

Figure 2a shows examples of marker intensity distributions before and after normalization for the unimodal distribution of marker PD-L1 (CRC-ORION) and the bimodal distribution of marker ECAD (PRAD-CyCIF). For PD-L1, all methods seem to achieve a good alignment, however, only UniFORM preserves the distribution shape without alterations. For ECAD, UniFORM alone

successfully aligns the peaks and retains the original distribution shape while other methods fail in both alignment and shape preservation. Additional marker alignments are provided in **Supplementary Figures S1 and S2**, illustrating UniFORM's performance across a range of markers and further validating its ability to maintain alignment and distribution shape consistency. The results demonstrate that UniFORM achieved near-perfect alignment of negative peaks across all markers (**Figure S1b, Figure S2b**) while preserving their original distribution shapes, unlike other methods, which were unable to align the signals accurately and maintain their distribution shapes (**Figure S1c-e, Figure S2c-e**), which is critical for accurate cell phenotyping of MTI data across multiple batches.

To further evaluate the performance of UniFORM in batch normalization, we used kBET analysis on MTI data generated from multiple batches. kBET implicitly assumes that samples across batches share similar cellular compositions to accurately assess batch effects without being confounded by biological variability. To ensure a fair kBET analysis, we selected samples with comparable cellular compositions from different batches. Specifically, we focused on five groups with distinct cellular populations to maintain heterogeneity across the selected samples (**Table S2**). We evaluated batch effect removal using three metrics: kBET acceptance score, average Chi-square statistics, and average p-value (see **Methods**). A higher kBET acceptance score indicates better mixing of similar cell types, reflecting more effective batch correction. The Chi-square statistic measures the magnitude of batch imbalance, with lower values indicating better batch mixing. The average p-value reflects the statistical significance of batch mixing, with higher values indicating closer alignment to an ideal batch-free distribution.

As demonstrated in **Figure 2b**, group 1 (PRAD02 & PRAD14) exhibits significant population differences, with one sample much larger than the other. This discrepancy artificially inflated the kBET score. However, UniFORM still achieved the best performance in all metrics, highlighting its robustness even with imbalanced datasets. For the remaining groups, where samples exhibit similar population size and cell composition but also significant batch effect, UniFORM outperformed other methods by a significant margin (**Table 1** and **Table S2**), with the highest average kBET score of 0.60843 (raw=0.00098, MxNorm=0.38788, z-score=0.28570, ComBat=0.27266), lowest average Chi-square statistics of 6.4473 (raw=20.093, MxNorm=9.9575, z-score=14.823, ComBat=15.298), and highest average p-value of 0.25835 (raw=0.00145, MxNorm=0.15307, z-score=0.11459, ComBat=0.10981). We have further demonstrated UniFORM's effective batch correction through UMAP visualization shown in **Figure S3**.

Feature-Level Normalization: Positive Population Percentage Preservation

Effective normalization should adjust for technical variations and biases without altering the inherent biological distribution of marker expression. To evaluate this, we compared the percentage of positive populations in samples before and after normalization. The primary goal of normalization is to establish a reliable global threshold across samples that accurately differentiate positive and negative populations while maintaining the integrity of the positive population percentage. This consistency is essential for downstream analyses, particularly in cell phenotyping, where a global threshold ensures that any subsequent classification of cells into phenotypic categories remains consistent, reducing variability and minimizing bias from sample-specific thresholds.

To achieve this, we applied a two-component Gaussian Mixture Model[24] (GMM, see **Methods**) to determine local thresholds for each raw sample and a global threshold for the normalized samples in each marker of the CRC-ORION and PRAD-CyCIF dataset. The mean percentage change (see **Methods**) was then calculated by comparing the local thresholds and the global threshold. While perfect preservation of the positive population percentage is unlikely, significant changes should be avoided to ensure data integrity.

UniFORM effectively preserves population levels from local thresholds (pre-normalization) to global thresholds (post-normalization), with minimal percentage changes and low standard deviation across all markers in both datasets. In contrast, other methods exhibited substantial shifts in thresholds, high mean percentage changes, and significant standard deviations (**Figure 2c**, **Figure 2d**). These high standard deviations indicate inconsistent performance across markers and samples, reflecting poor alignment of distributions and signal distortion that adversely affected the GMM thresholding (**Table S3a, Table S3b**, **Figure S4a, Figure S4b**). EpCAM and CD45 in the PRAD-CyCIF dataset exhibited higher mean percentage changes, which can be attributed to the heterogeneity of these markers. Certain samples have a pronounced positive population, while others exhibit negligible positive populations. For samples with low positive populations, applying a local GMM threshold can lead to biases, as the minimal positive population may skew the threshold estimation. In contrast, a global GMM threshold, calculated across all samples, provides a more reliable and consistent threshold. However, the global threshold may differ from the local thresholds, particularly in heterogeneous datasets, contributing to the observed percentage change.

Feature-Level Normalization: Impact on downstream analysis

To evaluate the impact of UniFORM normalization on downstream analysis, we performed UMAP visualizations and Leiden clustering (see **Methods**). As shown in **Figure 3a**, the pre-normalization UMAP revealed distinct sample separation, indicating the presence of batch effects across samples. In contrast, post-normalization UMAP embeddings showed a well-mixed distribution of samples, demonstrating that UniFORM effectively minimized batch effects. To further assess whether similar cell types from different batches clustered together after UniFORM, we examined the UMAP plots color-coded by the intensity values of key protein markers (CD45, αSMA, and ECAD). In the raw data, high-intensity regions for these markers were dispersed across multiple clusters, indicating batch-induced variability in marker expression. After normalization, these high-intensity regions clustered together, highlighting that UniFORM successfully aligned positive cell populations across batches, supporting both improved normalization across samples.

For the Leiden clustering analysis, we calculated the distribution of cluster proportions across samples before and after normalization. As shown in **Figure 3b**, before normalization, cluster proportions were highly skewed, with many samples dominated by a single cluster or containing only a limited subset of clusters. This pattern indicates sample-specific biases, likely driven by batch effects and technical variability, which masked the true biological heterogeneity. After normalization, cluster representation across samples became more balanced, with each sample contributing more uniformly to the overall clustering structure. This improved cluster consistency suggests that UniFORM successfully reduced technical artifacts, enabling more meaningful and biologically relevant clustering. This ensures clustering was driven by intrinsic biological variation rather than batch effect, which enhances downstream analyses, such as cell type annotation, and allows for more accurate comparisons between experimental conditions.

Evaluation of Pixel-Level Normalization Methods

A major novel contribution of UniFORM is the introduction of a robust pixel-level normalization workflow (see **Methods**), which effectively aligns marker signals while preserving distribution shape. Similar to feature-level normalization, this approach applies the normalization factor at the pixel level. UniFORM uses only pixels within the cell/nuclei segmentation masks, excluding non-cell regions to yield a cleaner and more accurate representation of the true signal, thereby enhancing the accuracy of signal alignment. UniFORM processes unsigned 16-bit images—the

standard format for MTI images—and implements a mechanism to prevent range compression or overflow issues that can occur when the image is multiplied by a normalization factor and scaled back to 16-bit again. Consequently, the normalized images are also saved as 16-bit files without any unwanted artifacts.

Pixel-Level Normalization: Intensity Distribution Alignment

Following our feature-level evaluation, we performed the automatic normalization on the CRC-ORION and PRAD-CyCIF pixel-level datasets and evaluated the efficacy of signal alignment. UniFORM demonstrated consistent effectiveness in aligning the pixel intensity distribution to the cell mean intensity distribution. **Figure 4a** shows that both PD-L1 and ECAD achieved good alignment while preserving the original shape of the distribution, and the rest of the markers also achieved good alignment (**Figure S5a-b, Figure S6a-b**).

Pixel-level normalization differs from feature-level normalization in how the normalization factor affects data distribution. When a normalization factor < 1 is applied at the pixel level, the marker intensity distribution shifts left, resulting in a narrower range. The conversion back to uint16 introduces compression artifacts ('spikes') due to limited unique values in the range. In contrast, feature-level normalization, which is maintained in float64, avoids this issue and preserves data precision. To address this compression issue, we incorporated a linear stretching mechanism (see **Methods**), applying a stretching factor to each sample to restore its range to pre-normalization levels. This approach ensures even the most compressed samples avoid artifacts associated with data compression while maintaining the original distribution shape.

Figure 4b illustrates the effect of applying and not applying a stretching factor. For PD-L1, the raw CRC05 distribution (blue) is expected to align with the reference (light green). However, due to leftward shifts in some samples, a common stretching factor is calculated and applied uniformly to all samples, including the reference, after the normalization step. This process aligns the distributions while linearly stretching them preserving the original distribution shape. For ECAD, where the reference is already the right-most sample, no stretching is required as all distributions are shifted to the right. The reference sample remains unchanged (i.e., no stretching factor), and the raw PRAD-CyCIF sample aligns effectively with the reference distribution. In both cases, the signals align accurately with their references, addressing compression and overflow issues and ensuring robust pixel-level normalization.

To further illustrate that no image artifacts are introduced after normalization, we examined the PD-L1 channel of CRC05 before and after normalization. The results clearly show that no artifacts were introduced during the process, and the single-cell granularity was perfectly preserved (**Figure 4c**).

Pixel-Level Normalization: correlation with feature-level normalization

Finally, to assess the relationship between features extracted from pixel-level normalized images and those from the feature-level normalized dataset, we computed Spearman correlations and plotted the line of best fit (see **Methods**). This analysis was performed using the PD-L1 marker from the CRC-ORION dataset and ECAD marker from PRAD-CyCIF dataset. The features extracted from pixel-level normalized images in both datasets showed great alignment comparable to those from the feature-level normalized dataset, displaying a linear relationship with near-perfect Spearman correlations (**Figure 4d**). These results indicate that normalizing the pixel-level image dataset also effectively normalizes the feature-level mean intensity dataset. Furthermore, the strong correlation (**Figure 4d, 4e**) suggests that features derived from normalized images consistently maintain the positive population percentage.

Discussion

MTI datasets, both at pixel and feature levels, often exhibit right-skewed distributions, largely influenced by commonly observed trends. This skewness primarily arises from the biological nature of marker expression, where a subset of cells exhibits high-intensity signals while the majority show lower or background levels. These high-intensity signals contribute to the long tail of the distribution that varies across samples due to heterogeneous expression patterns inherent in tissue samples. The three benchmarked methods—MxNorm, z-score, and ComBat struggle to effectively account for the interplay of biological variability, technical noise, and distribution skewness. Consequently, these methods frequently misalign signals, distort original distributions, and change the proportions of positive populations, highlighting the need for approaches tailored to the unique characteristics of MTI data.

MxNorm utilizes functional data registration with B-spline functions to align histogram distributions. B-spline function data registration creates many smooth cubic curves to fit the distribution, assuming a somewhat normal or symmetric distribution and relative smoothness in data. These assumptions are not well-suited for the right-skewed distributions data in MTI data, as they can distort the distributions, particularly in the long tail, by imposing excessive

smoothness that results in a loss of key biological information. While MxNorm may handle simpler distributions, such as PD-L1, it struggles with more complex distributions like ECAD. This is evident from the major distortion of distribution in **Figure 2a** and the significant positive percentage change in **Figure 2c**.

z-score normalization, while widely used for its simplicity, relies on assumptions of normally or symmetrically distributed data and consistent cell composition and signal distribution across samples. However, MTI data often show inherent right skewness and heterogeneity, making z-score normalization less suitable. In right-skewed data, the mean is shifted toward higher values due to the tail. When samples have consistent cell composition (as with PD-L1 in previous examples), z-score normalization may perform adequately. However, when cell composition varies significantly (as seen with ECAD), each sample's reference point (mean) and scale (standard deviation) can be very different, resulting in misalignment. Furthermore, in right-skewed data, z-score normalization causes lower-intensity data points to become overly compressed around zero, while high-intensity points may become exaggerated. This distorts the natural spread and distribution of data. Similar limitations are also observed in other methods like mean division and arcsinh normalization.

Similar to z-score normalization, ComBat assumes a common distribution and consistent cell composition across samples, with additive batch effects that can be linearly modeled. However, the right skewness, heterogeneous positive signals, and non-uniform technical variation in MTI data violate these assumptions. As a result, when the samples have significantly different cell compositions, ComBat may struggle to differentiate between technical batch effects and biological variation, potentially resulting in over-correction or under-correction. Additionally, ComBat can alter the overall distribution of data, particularly when data deviate from normality, making it inherently unsuitable for MTI normalization.

FLINO uses a grid-based quantile normalization (Q75) approach, dividing each image into multiple grids and adjusting raw intensities using a multiplicative scaling factor. For each grid, raw intensity is normalized by multiplying it with the ratio of a global quantile (calculated across all images) to a local quantile (specific to the grids in each sample). This aligns intensities to a shared global quantile reference, assuming all samples share a common intensity distribution at the chosen quantile. However, this assumption falters with right-skewed and heterogeneous MTI marker distributions. For right-skewed markers, the Q75 value, influenced by a long tail of high-intensity values, poorly represents the central trend, leading to overcorrection in low-

intensity grids or undercorrection in high-intensity grids. Additionally, FLINO does not distinguish between negative peaks (background) and positive peaks (biological signals), risking distortion of key biological features. Normalizing each grid independently can also amplify batch effects or local intensity differences, introducing artificial variability or failing to correct pre-existing inconsistencies. From a computational standpoint, FLINO's R-based workflow and reliance on dividing images into grids make it resource-intensive, inefficient, and impractical for modern MTI datasets, which often contain millions of pixels and thousands of grids per slide.

Our universal normalization pipeline, UniFORM, is free from restrictive assumptions, making it well-suited for any MTI dataset. **Table 2** summarizes and highlights that UniFORM is the most comprehensive method, outperforming others in all evaluating metrics and effectively correctly both inter-batch and intra-batch variations. However, the effectiveness of UniFORM depends on the quality of the cell/nuclei segmentation masks used to extract signals. We employed the MESMER cell segmentation model[25] (see **Methods**), a state-of-the-art tool, to generate these masks. For datasets with minimal cell loss, like CRC-ORION, we used the first-round DAPI channel and corresponding morphology channels to extract the masks. For datasets with significant cell loss in later staining rounds like the PRAD-CyCIF dataset, we adjusted the masks to account for tissue loss by comparing the first-round DAPI masks with the last-round masks, ensuring cleaner and meaningful biological signals.

Normalization methods in MTI and single-cell analysis often face a trade-off between achieving a good mixture of cells (as indicated by kBET acceptance rates) and preserving biological signal[26–29]. In our study, UniFORM demonstrated the highest kBET metrics across all groups, outperforming other methods that rely on artificial homogenization to improve the mixture at the cost of destroying biological integrity. This highlights UniFORM's superior ability to achieve both robust mixing and signal preservation, setting it apart as a more balanced and effective normalization approach.

In certain cases such as the CD45 marker in the PRAD-CyCIF dataset, marker intensity distributions exhibit irregular patterns but biologic patterns, such as multimodal distributions or inconsistent positive-to-negative population ratios (**Figure S7a**). These irregularities violate the assumptions of the automatic normalization workflow. To address this challenge, we implemented a guided manual fine-tuning option (see **Methods, Figure S7b**), which allows users to specify "landmarks" on the distribution curves to guide alignment. This feature

demonstrates the flexibility of the pipeline, enabling accurate signal alignment even in the most complex cases where automated methods alone may fall short.

A limitation of our normalization pipeline is its reliance on the linear application of normalization factors. While this approach is well-suited to IF-based techniques such as CyCIF and CODEX, which exhibit a linear relationship between signal intensity and antigen concentration, it is unsuitable for multiplexed immunohistochemistry (mIHC). In mIHC, chromogenic detection produces non-linear signal intensities because of enzyme-catalyzed colorimetric amplification, which disrupts the proportionality between signal intensity and antigen concentration[2–4]. Applying a linear normalization factor to mIHC data can lead to misrepresentation, as the scaling may unevenly affect intensity levels, exaggerating or diminishing the actual signal distribution. To address this, a warping function might be needed to perform a non-linear normalization of the mIHC samples.

Ultimately, this work demonstrates the strength of UniFORM as a versatile normalization pipeline, balancing automatic approaches with manual flexibility to achieve precise signal alignment, across diverse MTI data sets and cancer types. By addressing key challenges in batch correction and normalization, UniFORM sets the foundation for accurate, reproducible analysis of multiplex imaging data, with the potential to significantly improve downstream biological insights and clinical applications.

Methods

Dataset

Detailed information on CRC-ORION and PRAD-CyCIF datasets is outlined in **Table S1** and the **Data Availability** section.

Preprocessing

The CRC-ORION has already been processed and made available online[30]. Preprocessing steps of the PRAD-CyCIF dataset are detailed in our previous publication[31]. Both datasets are prepared in the ome.tif format. Mesmer cell segmentation is used to segment cells and nuclei for both datasets. Nuclear marker DAPI, and membrane markers (a composite projection of Pan-cytokeratin and CD45 marker) were used for the CRC-ORION dataset. Nuclear marker DAPI, and membrane markers (a composite projection of EpCAM, CD45, and ECAD marker) were used for the PRAD-CyCIF dataset. The masks were registered and refined to ensure each

nucleus had a corresponding cell membrane. The cell ID, centroids, and cell mean intensities were extracted with scikit-image[32].

Quality control

Quality control procedures were implemented to account for tissue loss and ensure the removal of imaging defects and artifacts. Once the masks were refined to ensure each cell had a corresponding nucleus, a binary mask from the last-round nuclear mask was used to rapidly filter cells based on the overlap with the first-round nuclear mask. Cells were considered lost if their mean intensity overlapped with the last-round nuclear mask and fell below a specified threshold. We used 0.10 in our pipeline to remove the lost cells and maintain the cells that might have drifted mildly due to local tissue deformation. The filtered masks were processed to ensure that nuclear and cell masks maintained consistent IDs. Code to perform tissue loss filtering is available along with our normalization pipeline. Besides tissue loss, other common imaging defects and artifacts include tissue folding and wrinkling, tissue disintegration, significant autofluorescence, bright spots due to dye aggregation, and air bubbles and debris. Samples with these artifacts were either dropped or cropped depending on the severity.

UniFORM: Feature-level normalization

UniFORM's feature-level normalization pipeline can be divided into three steps: 1. Calculate histogram, 2. Automatic functional data registration, 3. Normalization. Feature-level normalization takes in the raw cell mean intensity data, undergoes the automatic max-correlation based functional data alignment, and then outputs the normalized cell mean intensity data.

1. **Calculate histogram**

   For each marker sample, we apply a logarithmic transformation step to enhance dynamic range and skewness. Then for each marker, it iterates through samples to determine a common global range, this ensures a uniform and consistent comparison across samples and avoids confounding effects due to varying axis scales. After a common global range is established, it then iterates through samples to produce histograms along with plots that provide an essential visualization of raw data distribution.

2. **Automatic functional data registration**

In the second step of the automatic normalization pipeline, we align the distributions of cell mean intensities across samples using a Fast Fourier Transform (FFT)-based max-correlation algorithm. This approach ensures that for each marker, all sample histograms are aligned to a selected reference histogram. The reference sample is chosen to be representative of the marker's characteristics and serves as the baseline for alignment.

The algorithm works by computing the cross-correlation between each sample's histogram and the reference histogram to determine the optimal shift that maximizes their alignment. To facilitate this process, histograms are first transformed into functional data objects (FDataGrid), converting discrete histograms into continuous functional representations. This transformation allows for smooth mathematical operations and precise functional alignment.

The FFT-based cross-correlation efficiently identifies the shift that maximizes similarity between the sample and reference histograms. The calculated shifts are then applied to realign the sample histograms to the reference, standardizing the distribution of cell mean intensities across all samples. The algorithm is summarized in the pseudocode below:

Algorithm 1 Correlation-Based Normalization Using FFT

procedure AUTOMATICALIGNMENT(`histogram_dict`, `markers_list`, `reference`)
  Initialize shifts_fft_dict $\leftarrow \{\}$
  for each marker $m \in$ `markers_list` **do**
    for each sample histogram $h_i \in$ `histogram_dict`$[m]$ **do**
      Convert h_i into functional data object:

$$f_i(t) = \mathrm{FDataGrid}\big(h_i, t\big), \quad t = \{0, 1, \ldots, N-1\}$$

      Compute cross-correlation with reference h_{ref}:

$$C_i(\tau) = \mathcal{F}^{-1}\big(\mathcal{F}(h_{\mathrm{ref}}) \cdot \overline{\mathcal{F}(h_i)}\big), \quad \forall \tau \in [-N+1, N-1]$$

      Determine optimal shift τ_i^*:

$$\tau_i^* = \arg\max_{\tau} C_i(\tau)$$

      Store τ_i^* in shifts_fft_dict$[m]$
    end for
    Shift functional data $f_i(t)$ by τ_i^*:

$$f_i'(t) = f_i(t - \tau_i^*)$$

  end for
  return shifts_fft_dict
end procedure

3. Normalization

In step 3 of the normalization pipeline, we apply the normalization factors derived from FFT-based shifts to the raw data to achieve normalization across all samples. This is achieved by first translating the normalization factors in the FData scale to the logarithmic scale to ensure these shifts correspond accurately to the scale of the log-transformed raw data. Then for each sample and marker, the normalization factor is applied by scaling the original marker intensity data e^{shift} to make sure the shift is applied at the scale of the raw data. The normalized data is then saved as float64 data type. The algorithm can be described by the pseudocode below:

Algorithm 2 Performing Data Normalization

```
procedure NORMALIZATION(range_dict, shifts_fft_dict, marker_list, feature_data)
    Initialize shift_in_log_dict ← {}
                                        ▷ Step 1: Translate shifts to log scale
    for marker ∈ marker_list do
        (global_min, global_max) ← range_dict[marker]
        Δ ← (global_max − global_min) / (num_bins − 1)
        shifts ← shifts_fft_dict[marker]
        shift_in_log ← {Δ · s | s ∈ shifts}
        shift_in_log_dict[marker] ← shift_in_log
    end for
                                        ▷ Step 2: Normalize raw data
    Initialize normalized_data ← {}
    for sample ∈ feature_data do
        for marker ∈ marker_list do
            shift ← shift_in_log_dict[marker]
            normalized[marker] ← feature_data[sample, marker] · exp(shift)
            Store normalized[marker] in normalized_data[sample, marker]
        end for
    end for
    return normalized_data
end procedure
```

UniFORM: Pixel-level normalization

Pixel-level normalization follows a very similar workflow to feature-level, differing only in how the input is processed and how the normalization factor is applied to the raw image data. The pixel-level pipeline handles raw pixel data in the format of Dask arrays for computational efficiency and uses cell/nuclei segmentation masks to filter out non-relevant areas and ensure that only pixels corresponding to cells or nuclei are used for the histogram calculation. Then it undergoes a similar workflow of histogram calculation and automatic normalization factor estimation.

However, the two workflows differ in how the normalization factor is applied to the raw data. As described in the results section, compression and overflow issues are specific to pixel-level data as they need to be maintained in uint16 after normalization.

The stretching mechanism in the pixel-level normalization pipeline is designed to address compression artifacts that arise when applying normalization factors < 1 to raw pixel data. When pixel intensity values are scaled-down and converted back to uint16, the same number of unique intensity values needs to be rounded to a narrow range, this restriction can introduce artifacts causing spikes in the histogram where multiple original values round to the same output value. To determine the stretching factor, for each marker, the most compressed normalization factor (corresponds to the sample with the most left-shift) is determined, and we will be able to calculate a stretching factor that ensures the range of compressed data can be expanded to its pre-normalization level. All normalized samples will be linearly stretched by this stretching factor for consistency. The algorithm can be described by the pseudocode below:

Algorithm 3 Pixel-level Normalization and Stretching Mechanism

```
procedure PixelNormalize(raw_data, marker_list, shift_in_log_dict)
                                        ▷ Step1: Determine stretching factor
    for each marker in marker_list do
        Find min_shift as the minimum value in shift_in_log_dict
        The most compressed value (in original scale) is e^(min_shift)
        Compute stretching_factor as 65535 / e^(min_shift)
        Store stretching_factor in stretch_factor_dict
    end for
                                        ▷ Step2: Apply normalization and stretching
    for each sample in raw_data do
        for each marker in marker_list do
            Retrieve shift from shift_in_log_dict for the sample and marker
            if shift ≤ 0 then                ▷ Range compression or no shift
                Compute norm_factor as e^(shift)
                Normalize normalized_data = raw_data × norm_factor
            else                                          ▷ Range expansion
                Convert raw data to uint32 to prevent overflow
                Normalize normalized_data = raw_data_uint32 × e^(shift)
            end if
                                        ▷ Step3: Apply stretching to normalized data
            Retrieve stretching_magnitude from stretch_factor_dict
            Apply stretching:
```

$$\text{stretched_normalized} = \left(\frac{\text{normalized_data} - \min(\text{normalized_data})}{\text{stretching_factor} - \min(\text{normalized_data})}\right) \times 65535$$

```
        end for
    end for
    Save the normalized data as uint16
end procedure
```

The normalized pixel-level data is saved as an uint16 image stack in the ome.tiff format, the same as the input image format.

UniFORM: Manual fine-tuning option (landmark normalization)

The manual landmark normalization step in the pipeline allows for normalization adjustment based on selected landmarks that represent user-defined negative peaks. These landmarks are critical points chosen by the user, either manually or with guided assistance using Gaussian Mixture Model (GMM) curves as a visual aid. The process starts by plotting functional data curves of the histogram distribution for each marker on an interactive *plotly* plot (https://github.com/plotly/plotly.py), allowing the user to quickly and easily identify landmarks.

The landmark function calculates the shift needed to align the functional data curves of each sample to the reference negative peak. If landmarks are provided for a marker, the landmark function adjusts the curves so that the user-defined locations become aligned with the user-defined reference. If landmarks are not provided, the pipeline uses the automatic max-correlation based method.

Implementation of normalization benchmarking techniques

* **MxNorm Normalization:** We followed the guidelines on the MxNorm Github repository: https://github.com/ColemanRHarris/mxnorm to conduct normalization. We used the best-reported normalization methods for analysis, specifically transform = “log10_mean_divide” and method = “registration” (meaning that performing a log10 and mean division transformation of the data before performing B-spline function data registration).

* **ComBat Normalization:** Our ComBat normalization is implemented using pyComBat, a new Python implementation of ComBat algorithm to adjust for batch effects within our data. Prior to applying ComBat, a log10 transformation with a small offset was applied to the data. We followed the guidelines on pyComBat website: https://epigenelabs.github.io/pyComBat/ to conduct normalization.

* **z-score Normalization:** We perform z-score normalization on the sample level using z-score implementation from SciPy (scipy.stats). Prior to applying z-score, a log10 transformation with a small offset was applied to the data.

Gaussian Mixture Model (GMM) Analysis

2-component GMMs, statistical methods that fit two Gaussian distributions to data to identify underlying subpopulations, were used in our study. We adopted the SCIMAP[24] implementation, which uses a simple GMM model, for visualization purposes in Figure 2a. For more accurate positive population thresholding (in our positive population change analysis), we implemented a more complex GMM which uses more complex model initiation and uses the Gaussian probability density functions (PDFs) to determine the best threshold for positive population. The algorithm can be described by the pseudocode below:

a SCIMAP GMM for Visualization

1: **procedure** SCIMAPGMM($\mathbf{X}$)

2: Transform input distribution $\mathbf{X}$:

$$\mathbf{X}_{\log} = \log_{10}(\mathbf{X} + 1)$$

3: Fit a Gaussian Mixture Model (GMM) with 2 components:

$$\mathcal{L}(\mathbf{X}_{\log}; \boldsymbol{\mu}, \boldsymbol{\sigma}) = \sum_{k=1}^{2} w_k \cdot \mathcal{N}(\mathbf{X}_{\log}; \mu_k, \sigma_k^2)$$

where w_k are component weights, and $\mathcal{N}$ represents the Gaussian distribution.

4: Compute the cut-off value as the mean of the component means:

$$\text{cut-off} = \frac{\mu_1 + \mu_2}{2}$$

5: **return** cut-off

6: **end procedure**

b Custom GMM for Thresholding

1: **procedure** GMM($\mathbf{X}$)

2: Transform input distribution $\mathbf{X}$:

$$\mathbf{X}_{\log} = \log_{10}(\mathbf{X} + 1)$$

3: Fit a Gaussian Mixture Model (GMM) with 2 components using custom initialization and maximum iterations:

$$\mathcal{L}(\mathbf{X}_{\log}; \boldsymbol{\mu}, \boldsymbol{\sigma}, \boldsymbol{\pi}) = \sum_{k=1}^{2} \pi_k \cdot \mathcal{N}(\mathbf{X}_{\log}; \mu_k, \sigma_k^2)$$

where π_k are initialized weights.

4: Predict class labels $\mathbf{y}$ for $\mathbf{X}_{\log}$:

$$\mathbf{y} = \arg\max_k \left(\pi_k \cdot \mathcal{N}(\mathbf{X}_{\log}; \mu_k, \sigma_k^2)\right)$$

5: Separate data into negative ($\mathbf{X}_{\text{neg}}$) and positive ($\mathbf{X}_{\text{pos}}$) classes based on predictions.

6: Compute the cut-off as the maximum of the negative class:

$$\text{cut-off} = \max(\mathbf{X}_{\text{neg}})$$

7: **return** cut-off

8: **end procedure**

Positive population change analysis

For each normalization method, we conducted a positive population change analysis using the GMM approach described above to quantify its efficacy. For each marker, a local positive population threshold is calculated for each raw sample, while a global threshold is determined for the combined normalized samples. Then we evaluated the proportions of cells above the local threshold and global threshold for each sample and the mean and standard deviation of changes in positive population proportions can be determined.

kBET analysis

Due to the computational inefficiency of the R implementation of kBET, we performed kBET analysis using pegasuspy, a python package with kBET implementation, to measure batch effects before and after normalization. We followed the guidelines on the package's website: https://pegasus.readthedocs.io/en/stable/ for implementation. Preprocessing included data standardization (z-score) and dimensionality reduction using principal component analysis (PCA), followed by the construction of a neighborhood graph and a UMAP embedding to enhance the representation of local data structures. The kBET algorithm was applied to the UMAP representation to assess batch effects more effectively.

kBET leveraged the neighborhood graph generated from the UMAP representation to evaluate how well samples from different batches mixed within their local neighborhoods. This analysis involved determining the k-nearest neighbors (kNN) for each cell and assessing whether the observed distribution of batch labels within these neighborhoods deviated from an ideal, uniform distribution. The deviation was quantified using a chi-squared test, resulting in a kBET score that reflects the extent of batch mixing. The kBET acceptance rate, calculated from the proportion of tests where the null hypothesis (no batch effect) was accepted, provided an overall measure of batch effect mitigation. A higher acceptance rate indicated more successful batch effect removal and a more homogeneous sample distribution across batches.

Metric	Definition and Equation	Interpretation
kBET acceptance score	Proportion of cells for which the null hypothesis (random batch mixing) is not rejected at significance level α. $$\text{Acceptance score} = \frac{\text{Number of cells with } p \geq \alpha}{\text{Total number of cells}}$$	Higher values indicate better mixing of cells from different batches.
Chi-square statistic (χ^2)	Measures the deviation between observed (O_i) and expected (E_i) batch distributions within a cell's k-nearest neighbors: $$\chi^2 = \sum_i \frac{(O_i - E_i)^2}{E_i}$$ where i represents batch categories.	Lower values indicate reduced batch imbalance and better correction.
Average p-value	Mean p-value for the null hypothesis test based on the χ^2 statistic: $$\text{p-value} = 1 - F_{\chi^2}(\chi^2, \text{dof})$$ where F_{χ^2} is the cumulative distribution function of the chi-squared distribution with degrees of freedom (dof).	Higher values indicate closer alignment with ideal random batch mixing.

UMAP visualization

For UMAP visualization, we utilized the pegasuspy package (https://pegasus.readthedocs.io/en/stable/index.html) to construct UMAP embeddings on top of the neighborhood graph. The UMAP was derived from the dataset after neighborhood graph construction to effectively capture the local and global structure of the data. To visualize the UMAP, we colored the plots by sample names and by CD45, αSMA, and ECAD marker intensity values.

Leiden clustering analysis

We performed Leiden clustering analysis using Scanpy[33] to identify distinct groups within the dataset before and after normalization. First data is transformed by z-score scaling, A neighborhood graph was constructed using 50 nearest neighbors, followed by Leiden clustering at a resolution of 0.3. To evaluate sample representation in each cluster, we calculated proportions by dividing the count of cells per sample by the total cell count in each cluster.

Data visualization

Matplotlib and seaborn packages in Python were used to generate all plots in this manuscript.

Statistics and reproducibility

All analyses were conducted using publicly available MTI dataset CRC-ORION and our in-house dataset PRAD-CyCIF, encompassing pixel-level and feature-level data across multiple batches. Data preprocessing involved log transformation with an offset to stabilize the variance. Benchmarking of UniFORM against z-score, ComBat, and MxNorm was performed using kBET to evaluate batch effect mitigation and Gaussian Mixture Models (GMMs) to assess the preservation of positive population percentages. Pixel-level normalization was evaluated by aligning pixel intensity distributions with feature-level data, with Spearman correlations quantifying the consistency between levels.

For reproducibility, we provide the necessary dataset and code used in our analysis.

Data availability

The CRC-ORION dataset is publicly available through the Human Tumor Atlas Network (HTAN) at https://humantumoratlas.org/, under participant IDs: HTA7_926, HTA7_927, HTA7_932, HTA7_934, HTA7_966, and HTA7_947. Our in-house PRAD-CyCIF dataset is accessible on Zenodo at https://zenodo.org/records/14257384.

Code availability

All software used in the study is detailed in the Methods section and supplementary material. All scripts are available via GitHub: https://github.com/kunlunW/UniFORM

Acknowledgments

This work was carried out with major support from the National Cancer Institute (NCI) Human Tumor Atlas Network (HTAN) Research Centers at OHSU (U2CCA233280), and funding (CEDAR3410918) from the Cancer Early Detection Advanced Research Centre at Oregon Health & Science University, Knight Cancer Institute (S.E.E.). Y.H.C. is supported by R01 CA253860, R01CA276224, U01 294548, Kuni Foundation Imagination Grants, and the Cancer Early Detection Advanced Research Center (CEDAR) pilot project grant. M.H.W is supported by R01 CA253860. The resources of the Exacloud high-performance computing environment developed jointly by OHSU and Intel and the technical support of the OHSU Advanced Computing Center are gratefully acknowledged.

Author Contributions

Conceptualization: K.W., Y.H.C. Software implementation: K.W., Y.H.C. Data analysis: K.W., S.D.K., K.A.A. Quality check and code review: K.W., Y.H.C., S.E.E., Z.S., E.C. Data generation: Z.S., J.Y. Manuscript writing: K.W., Y.H.C., S.E.E. Funding: M.H.W., G.B.M., S. E.E., Y.H.C. All authors contributed to the paper either through discussions or directly.

Competing interests

The authors declare the following competing interests: G.B.M. is a SAB member or Consultant: for Amphista, Astex, AstraZeneca, BlueDot, Chrysallis Biotechnology, Ellipses Pharma, GSK, ImmunoMET, Infinity, Ionis, Leapfrog Bio, Lilly, Medacorp, Nanostring, Nuvectis, PDX Pharmaceuticals, Qureator, Roche, Signalchem Lifesciences, Tarveda, Turbine, Zentalis Pharmaceuticals. G.B.M. has Stock/Options/Financial relationships with: Bluedot, Catena Pharmaceuticals, ImmunoMet, Nuvectis, SignalChem, Tarveda, and Turbine. G.B.M. has Licensed Technology: HRD assay to Myriad Genetics, DSP patents with Nanostring. G.B.M. has Sponsored research with AstraZeneca. The other authors declare no competing interests.

Figures

Figure 1: UniFORM offers universal automatic intensity normalization for both feature-level and pixel-level MTI data. **(a)** UniFORM accepts both pixel-level and feature-level data as input. Nucleus/cell segmentation masks are required for pixel-level normalization to exclude non-cell regions. **(b)** The study workflow includes datasets from two multiplex imaging platforms: CRC-ORION (17 channels, 6 patients, 1 batch) and PRAD-CyCIF (20 channels, 20 patients, 7 batches). UniFORM was benchmarked against commonly used normalization methods (MxNorm, z-score, and ComBat) and a baseline. The performance of each method was evaluated through signal alignment, kBET analysis, positive population change, and UMAP-Leiden clustering. **(c)** Overview of the UniFORM automatic normalization workflow. The process begins with histogram calculation, followed by signal alignment using functional data registration, and concludes with normalization. A major novelty and highlight of UniFORM is its pixel-level normalization capability. It undergoes the same pipeline but uses cell/nuclei segmentation masks and a slightly different normalization step.

Figure 2: Feature-level normalization evaluation and comparison. (a) Evaluation of cell mean intensity signal alignment across different normalization methods of unimodal marker PD-L1 and bimodal marker ECAD, with local positive population thresholds (colorful lines) and global threshold (black line) annotated. **(b)** Evaluation of batch effect correction with kBET with five patient groups, each composed of patient samples with similar cell composition but of different batches, measured by kBET acceptance rate (higher rate indicates better batch correction). **(c)** Evaluation of positive population change post-normalization across different normalization methods. The size of the bubbles indicates the magnitude of the absolute positive population mean percentage change, and the color of the heatmap indicates the standard deviation of the mean percentage change. **(d)** Violin plots show the spread and central tendency of the positive percentage change for PD-L1 and ECAD.

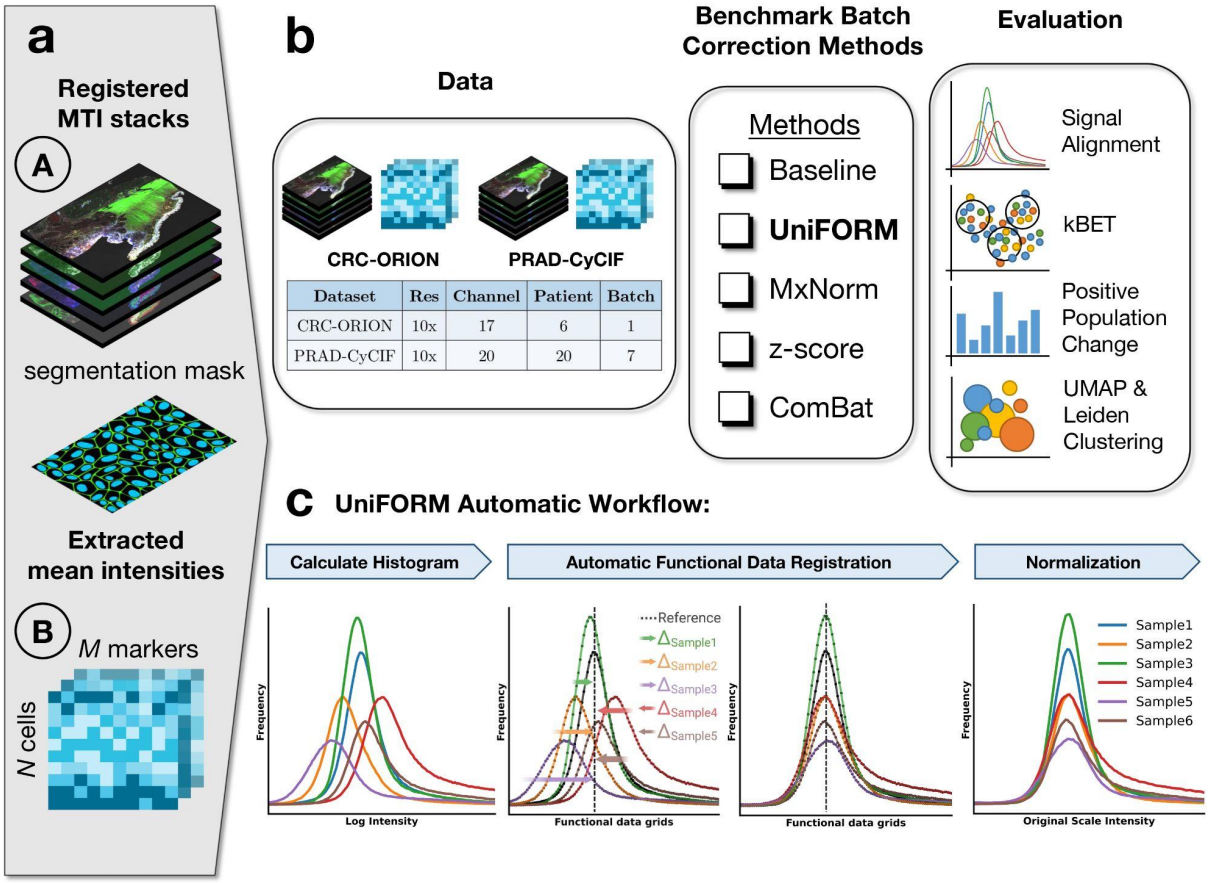
Figure 3: Impact of UniFORM on downstream analysis. (a) UMAP of raw and normalized PRAD-CyCIF dataset colored by sample IDs showing mixture after normalization and UMAP of raw and normalized PRAD-CyCIF dataset colored by the intensity value of marker CD45, α-SMA, and ECAD showing clustering of higher intensity regimen after normalization. **(b)** Leiden clustering of raw and normalized PRAD-CyCIF dataset showing enriched cluster proportions across samples after normalization.

Figure 4: Pixel-level normalization evaluation. (a) Evaluation of UniFORM's pixel intensity signal alignment of unimodal marker PD-L1 and bimodal marker ECAD **(b)** Demonstration of the sample signal alignment with respect to its reference and the effect of stretching mechanism. **(c)** Visual inspection of the PD-L1 channel before and after normalization confirms no artifacts are introduced. **(d)** Features extracted from normalized images achieve good signal alignment and almost perfect Spearman correlation with the feature-level normalized data. **(e)** Spearman correlation heatmap showing extracted features having almost perfect correlation with the feature-level normalized in both CRC-ORION and PRAD-CyCIF data. Refer to **Figure 2a** for legend information.

Table 1: Summary of the average statistics of the kBET analysis. UniFORM outperformed all other methods by a significant margin in all three kBET evaluation statistics.

Table 2: Comparison of normalization methods analyzed in this study. UniFORM outperformed existing methods across all evaluated criteria, demonstrating robustness and effectiveness. An asterisk (*) denotes features claimed by the authors but lacking implementation or comprehensive evaluation.

Figure 1:



Dataset	Res	Channel	Patient	Batch
CRC-ORION	10x	17	6	1
PRAD-CyCIF	10x	20	20	7

Figure 2

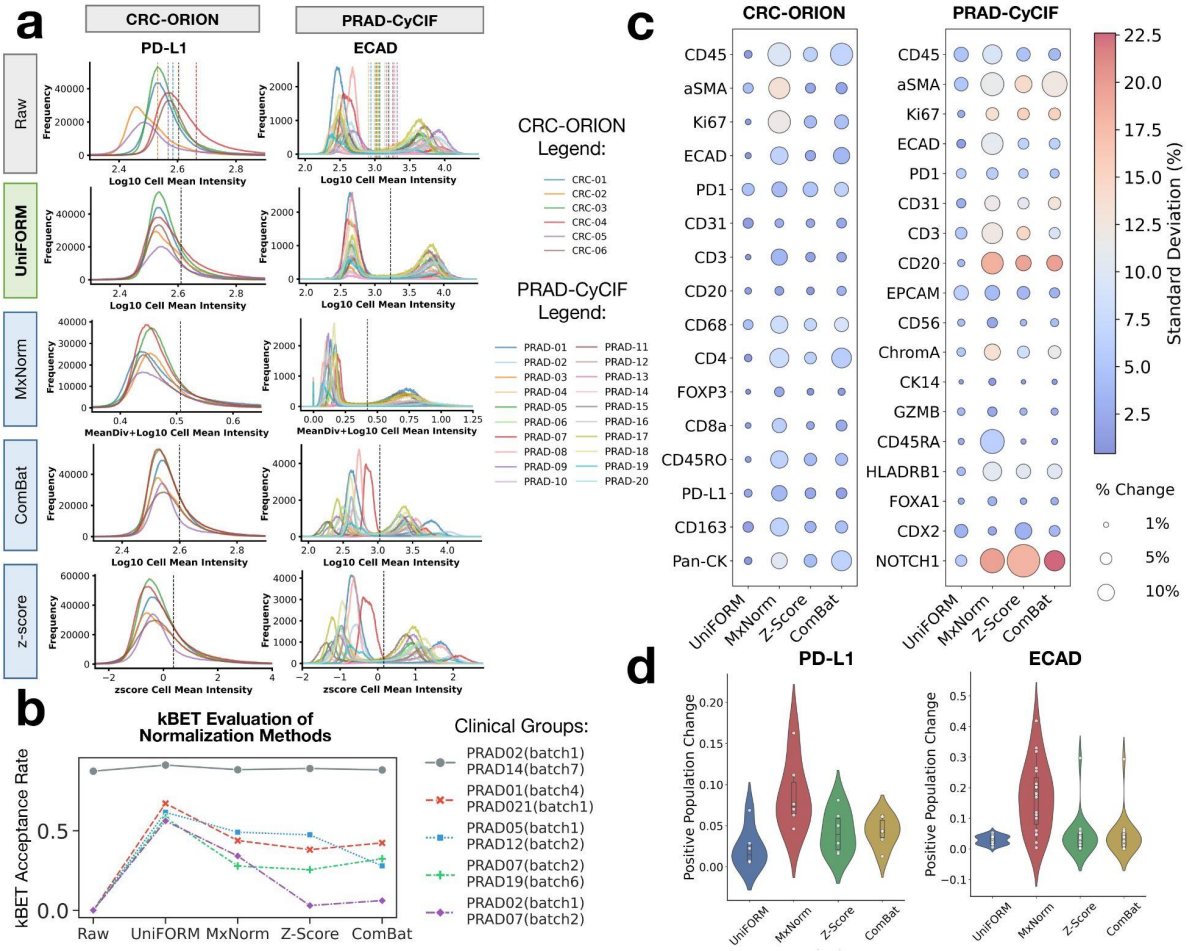

a
CRC-ORION
PRAD-CyCIF
PD-L1
ECAD
Raw
UniFORM
MxNorm
ComBat
z-score
Frequency
Log10 Cell Mean Intensity
MeanDiv+Log10 Cell Mean Intensity
zscore Cell Mean Intensity
CRC-ORION Legend:
CRC-01
CRC-02
CRC-03
CRC-04
CRC-05
CRC-06
PRAD-CyCIF Legend:
PRAD-01
PRAD-02
PRAD-03
PRAD-04
PRAD-05
PRAD-06
PRAD-07
PRAD-08
PRAD-09
PRAD-10
PRAD-11
PRAD-12
PRAD-13
PRAD-14
PRAD-15
PRAD-16
PRAD-17
PRAD-18
PRAD-19
PRAD-20
b
kBET Evaluation of Normalization Methods
kBET Acceptance Rate
Raw
UniFORM
MxNorm
Z-Score
ComBat
Clinical Groups:
PRAD02(batch1)
PRAD14(batch7)
PRAD01(batch4)
PRAD021(batch1)
PRAD05(batch1)
PRAD12(batch2)
PRAD07(batch2)
PRAD19(batch6)
PRAD02(batch1)
PRAD07(batch2)
c
CRC-ORION
PRAD-CyCIF
CD45
aSMA
Ki67
ECAD
PD1
CD31
CD3
CD20
CD68
CD4
FOXP3
CD8a
CD45RO
PD-L1
CD163
Pan-CK
EPCAM
CD56
ChromA
CK14
GZMB
CD45RA
HLADRB1
FOXA1
CDX2
NOTCH1
UniFORM
MxNorm
Z-Score
ComBat
Standard Deviation (%)
% Change
1%
5%
10%
d
PD-L1
ECAD
Positive Population Change
UniFORM
MxNorm
Z-Score
ComBat

Figure 3:

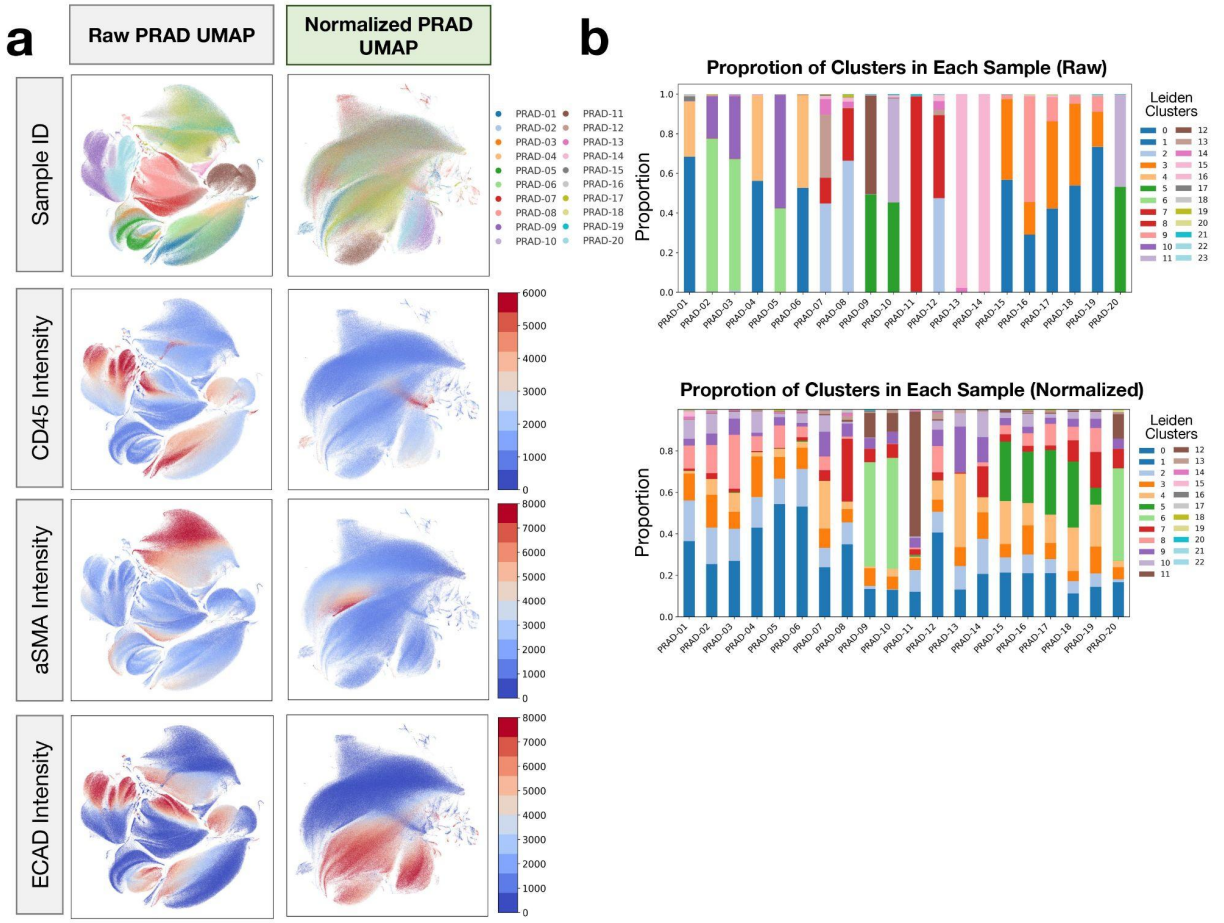

a
Raw PRAD UMAP
Normalized PRAD UMAP
Sample ID
CD45 Intensity
aSMA Intensity
ECAD Intensity
b
Proprotion of Clusters in Each Sample (Raw)
Proprotion of Clusters in Each Sample (Normalized)
Proportion
Leiden Clusters

Figure 4:

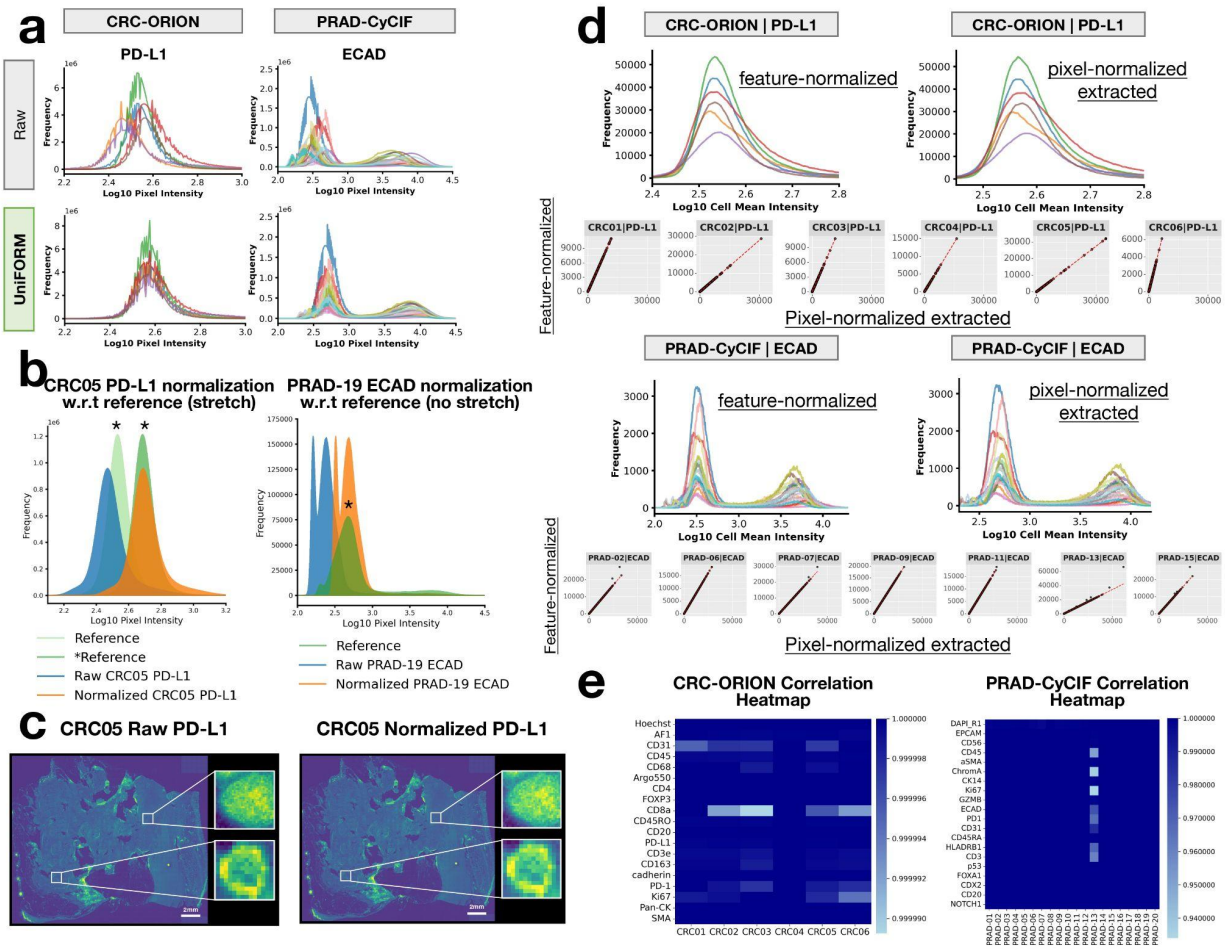

a
CRC-ORION
PRAD-CyCIF
PD-L1
ECAD
Raw
UniFORM
Frequency
Log10 Pixel Intensity
b
CRC05 PD-L1 normalization w.r.t reference (stretch)
PRAD-19 ECAD normalization w.r.t reference (no stretch)
Reference
*Reference
Raw CRC05 PD-L1
Normalized CRC05 PD-L1
Reference
Raw PRAD-19 ECAD
Normalized PRAD-19 ECAD
c
CRC05 Raw PD-L1
CRC05 Normalized PD-L1
d
CRC-ORION | PD-L1
feature-normalized
pixel-normalized extracted
Log10 Cell Mean Intensity
Feature-normalized
Pixel-normalized extracted
PRAD-CyCIF | ECAD
e
CRC-ORION Correlation Heatmap
PRAD-CyCIF Correlation Heatmap

Table 1:

Method	Average kBET Score	Average Chi-square Statistics	Average p-value
UniFORM	**0.60843**	**6.4473**	**0.25835**
Raw	0.00098	20.093	0.00145
MxNorm	0.38788	9.9575	0.15307
z-score	0.28570	14.823	0.11459
ComBat	0.27266	15.298	0.10981

Table 2:

Methods	Feature-level normalization	Align correct signals	Preserve distribution	Preserve population counts	Pixel-level normalization
UniFORM	✓	✓	✓	✓	✓
MxNorm	✓	✓	X	X	X*
Z-Score	✓	X	X	X	X
ComBat	✓	X	X	X	X
FLINO	X	X	✓	X	✓

Reference

1. Lin, J.-R., Fallahi-Sichani, M., Chen, J.-Y. & Sorger, P. K. Cyclic Immunofluorescence (CycIF), A Highly Multiplexed Method for Single-cell Imaging. *Curr. Protoc. Chem. Biol.* **8**, 251–264 (2016).
2. Tan, W. C. C. *et al.* Overview of multiplex immunohistochemistry/immunofluorescence techniques in the era of cancer immunotherapy. *Cancer Commun. Lond. Engl.* **40**, 135–153 (2020).
3. Hofman, P. *et al.* Multiplexed Immunohistochemistry for Molecular and Immune Profiling in Lung Cancer-Just About Ready for Prime-Time? *Cancers* **11**, 283 (2019).
4. Harms, P. W. *et al.* Multiplex Immunohistochemistry and Immunofluorescence: A Practical Update for Pathologists. *Mod. Pathol. Off. J. U. S. Can. Acad. Pathol. Inc* **36**, 100197 (2023).
5. Black, S. *et al.* CODEX multiplexed tissue imaging with DNA-conjugated antibodies. *Nat. Protoc.* **16**, 3802–3835 (2021).
6. Burlingame, E. A. *et al.* Toward reproducible, scalable, and robust data analysis across multiplex tissue imaging platforms. *Cell Rep. Methods* **1**, 100053 (2021).
7. Chang, Y. H. *et al.* RESTORE: Robust intEnSiTy nORmalization mEthod for multiplexed imaging. *Commun. Biol.* **3**, 111 (2020).
8. Harris, C. R. *et al.* Quantifying and correcting slide-to-slide variation in multiplexed immunofluorescence images. *Bioinforma. Oxf. Engl.* **38**, 1700–1707 (2022).
9. Graf, J. *et al.* FLINO: a new method for immunofluorescence bioimage normalization. *Bioinforma. Oxf. Engl.* **38**, 520–526 (2022).
10. Schapiro, D. *et al.* MITI minimum information guidelines for highly multiplexed tissue images. *Nat. Methods* **19**, 262–267 (2022).
11. Hickey, J. W., Tan, Y., Nolan, G. P. & Goltsev, Y. Strategies for Accurate Cell Type Identification in CODEX Multiplexed Imaging Data. *Front. Immunol.* **12**, 727626 (2021).

12. Eng, J. *et al.* A framework for multiplex imaging optimization and reproducible analysis. *Commun. Biol.* **5**, 438 (2022).

13. Behdenna, A. *et al.* pyComBat, a Python tool for batch effects correction in high-throughput molecular data using empirical Bayes methods. *BMC Bioinformatics* **24**, 459 (2023).

14. Argelaguet, R., Cuomo, A. S. E., Stegle, O. & Marioni, J. C. Computational principles and challenges in single-cell data integration. *Nat. Biotechnol.* **39**, 1202–1215 (2021).

15. Haghverdi, L., Lun, A. T. L., Morgan, M. D. & Marioni, J. C. Batch effects in single-cell RNA-sequencing data are corrected by matching mutual nearest neighbors. *Nat. Biotechnol.* **36**, 421–427 (2018).

16. He, S., Ye, H. & He, K. Spline estimation of functional principal components via manifold conjugate gradient algorithm. *Stat. Comput.* **32**, 106 (2022).

17. Yoon, C. *et al.* Deep learning-based virtual staining, segmentation, and classification in label-free photoacoustic histology of human specimens. *Light Sci. Appl.* **13**, 226 (2024).

18. Li, Y. *et al.* Virtual histological staining of unlabeled autopsy tissue. *Nat. Commun.* **15**, 1684 (2024).

19. Zhou, Z. *et al.* Virtual multiplexed immunofluorescence staining from non-antibody-stained fluorescence imaging for gastric cancer prognosis. *EBioMedicine* **107**, 105287 (2024).

20. Burlingame, E. A. *et al.* SHIFT: speedy histological-to-immunofluorescent translation of a tumor signature enabled by deep learning. *Sci. Rep.* **10**, 17507 (2020).

21. Sims, Z., Mills, G. B. & Chang, Y. H. MIM-CyCIF: Masked Imaging Modeling for Enhancing Cyclic Immunofluorescence (CyCIF) with Panel Reduction and Imputation. *BioRxiv Prepr. Serv. Biol.* 2023.05.10.540265 (2023) doi:10.1101/2023.05.10.540265.

22. Traag, V. A., Waltman, L. & Van Eck, N. J. From Louvain to Leiden: guaranteeing well-connected communities. *Sci. Rep.* **9**, 5233 (2019).

23. Büttner, M., Miao, Z., Wolf, F. A., Teichmann, S. A. & Theis, F. J. A test metric for assessing single-cell RNA-seq batch correction. *Nat. Methods* **16**, 43–49 (2019).

24. Nirmal, A. J. & Sorger, P. K. SCIMAP: A Python Toolkit for Integrated Spatial Analysis of Multiplexed Imaging Data. *ArXiv* arXiv:2405.02076v1 (2024).

25. Greenwald, N. F. *et al.* Whole-cell segmentation of tissue images with human-level performance using large-scale data annotation and deep learning. *Nat. Biotechnol.* **40**, 555–565 (2022).

26. Tran, H. T. N. *et al.* A benchmark of batch-effect correction methods for single-cell RNA sequencing data. *Genome Biol.* **21**, 12 (2020).

27. Arevalo, J., Su, E., van Dijk, R., Carpenter, A. E. & Singh, S. Evaluating batch correction methods for image-based cell profiling. *BioRxiv Prepr. Serv. Biol.* 2023.09.15.558001 (2024) doi:10.1101/2023.09.15.558001.

28. Luecken, M. D. *et al.* Benchmarking atlas-level data integration in single-cell genomics. *Nat. Methods* **19**, 41–50 (2022).

29. Yan, X., Zheng, R., Wu, F. & Li, M. CLAIRE: contrastive learning-based batch correction framework for better balance between batch mixing and preservation of cellular heterogeneity. *Bioinforma. Oxf. Engl.* **39**, btad099 (2023).

30. Lin, J.-R. *et al.* High-plex immunofluorescence imaging and traditional histology of the same tissue section for discovering image-based biomarkers. *Nat. Cancer* **4**, 1036–1052 (2023).

31. Ak, Ç. *et al.* Multiplex imaging of localized prostate tumors reveals altered spatial organization of AR-positive cells in the microenvironment. *iScience* **27**, 110668 (2024).

32. van der Walt, S. *et al.* scikit-image: image processing in Python. *PeerJ* **2**, e453 (2014).

33. Wolf, F. A., Angerer, P. & Theis, F. J. SCANPY: large-scale single-cell gene expression data analysis. *Genome Biol.* **19**, 15 (2018).

Supplementary material

See separate supplementary material PDF