IC-VAE: A Novel Deep Learning Framework for Interpreting Multiplexed Tissue Imaging Data

Huy Nguyen
BioTuring
San Diego, CA, USA
huy@bioturing.com

Hy Vuong
BioTuring
San Diego, CA, USA
hy@bioturing.com

Thao Truong
BioTuring
San Diego, CA, USA
thao@bioturing.com

Son Pham
BioTuring
San Diego, CA, USA
sonpham@bioturing.com

Abstract

Interpreting protein expression in multiplexed tissue imaging data presents a significant challenge due to the high dimensionality of the resulting images, the variety of intracellular structures, cell shapes resulting from 2-D tissue sectioning, and the presence of technological noise and imaging artifacts. Here, we introduce the Information-Controlled Variational Autoencoder (IC-VAE), a deep generative model designed to tackle this challenge. The contribution of IC-VAE to the VAE framework is the ability to control the shared information among latent subspaces. We use IC-VAE to factorize each cell's image into its true protein expression, various cellular components, and background noise, while controlling the shared information among some of these components. Compared with other normalization methods, this approach leads to superior results in downstream analysis, such as analyzing the expression of biomarkers, classification for cell types, or visualizing cell clusters using t-SNE/UMAP techniques.

Keywords Autoencoder · VAE · Multiplexed tissue imaging data · Spatial biology · Cell-type annotation

1 Introduction

Recent advances in highly multiplexed tissue imaging technologies, such as multiplexed immunofluorescence and imaging mass spectrometry [1] have enabled the simultaneous measurement of various proteins at single-cell resolution. Central to the analysis of highly multiplexed imaging data is the accurate identification of individual cells and the quantification of protein expression levels within them. Most current workflows [2] follow the following steps:

- Perform cell segmentation on the nucleus staining image and dilate the segmentation to obtain cell membrane. (As membrane staining is less reliable than nucleus staining, and current cell segmentation algorithms work better with nucleus segmentation).
- Quantify protein expression by summing the intensity of each protein for pixels within the cell boundaries.
- Normalize the protein expression of each cell by the corresponding cell area.

This workflow faces several limitations. First, the dilating procedure assumes the symmetry property of cells in the tissue section, which is often violated. Second, the tissue section cuts cells at different angles, resulting in different cell compartments (e.g., membrane, nucleus, cytoplasm, etc) in the image. While the protein signal is captured differently in cell compartments (e.g., proteins on the membrane will more likely be captured in the membrane compartment), normalizing the total expression by the cell area implicitly admits proteins can appear in any compartment in a cell at the same rate. Additionally, in areas where cells are densely located, signals from neighboring cells create a noisy background and should be eliminated.

We here propose a model to accurately quantify protein expression by using variational autoencoder (VAE) to reconstruct the original images of each cell using 3 components: the protein expression, the compartments of each cell, and background noise. We build a neural network to calculate and minimize the mutual information among the components. Through multiple benchmarks, we show that our new approach is more accurate than MCMICRO [2], the current state-of-the-art approach for quantifying protein expression on highly multiplexed tissue images.

2 Methods

Given an image A of dimensions $h \times w$, which captures the signal corresponding to protein distribution within a cell, we observe that the signal intensity varies across the image. This variation arises because different cellular compartments possess varying quantities of protein molecules, a heterogeneity intrinsic to the biological nature of the cell. Let r represent the number of distinct cellular compartments (e.g., nucleus, cytoplasm, membrane, etc.) within the cell, and let s denote the aggregate number of protein molecules present in the cell. Then, we can model the distribution of protein signals in the image as follows:

$$s = s_1 + s_2 + s_3 + ... + s_r \tag{1}$$

where s_i is the number of protein molecules in the i-th cell compartment, $s_i \geq 0$.

Furthermore, the image A captures a two-dimensional cross-section of a three-dimensional cell, which leads to the variability of protein signal within the same cell compartment. Let us define a $h \times w$ matrix

$$W_k = (w_{ij}), w_{ij} \geq 0, k \in \{1, 2, 3, ..., r\} \tag{2}$$

representing the extent to which the position i, j on the sectioning plane of the k-th cell compartment influences the protein signal magnitude.

From (1) and (2), we can interpret the image A as:

$$A = W_1 s_1 + W_2 s_2 + ... + W_r s_r + bJ = [W_1 \quad W_2 \quad \cdots \quad W_r] \begin{bmatrix} s_1 \\ s_2 \\ \vdots \\ s_r \end{bmatrix} + bJ \tag{3}$$

where b represents the background noise and imaging artifacts, J is the matrix of ones with the same dimension as A (and each W_k).

In a highly multiplexed tissue imaging dataset, scientists measure the signal of multiple proteins. Let c be the number of proteins. Assuming the effect of cell compartments on measuring protein signals is identical among proteins. Based on (3), we have c equations:

$$\begin{aligned} A_1 &= [W_1 \quad W_2 \quad \cdots \quad W_r] \begin{bmatrix} s_{11} \\ s_{12} \\ \vdots \\ s_{1r} \end{bmatrix} + b_1 J, \\ &\vdots \\ A_c &= [W_1 \quad W_2 \quad \cdots \quad W_r] \begin{bmatrix} s_{c1} \\ s_{c2} \\ \vdots \\ s_{cr} \end{bmatrix} + b_c J \end{aligned} \tag{4}$$

Let $t = h \cdot w$. The images $A_1, A_2, \ldots, A_c$ can be represented as a matrix V of size $t \times c$ where each column contains t pixel values of one of the c images in the set, flattened into a vector. Therefore, equation (4) is equivalent to:

$$
\begin{aligned}
V &= [\text{vec}(A_1) \quad \text{vec}(A_2) \quad \cdots \quad \text{vec}(A_c)] \\
&= [W_1 \quad W_2 \quad \cdots \quad W_r] \begin{bmatrix} s_{11} & s_{21} & \cdots & s_{c1} \\ s_{12} & s_{22} & \cdots & s_{c2} \\ \vdots & \vdots & \ddots & \vdots \\ s_{1r} & s_{2r} & \cdots & s_{cr} \end{bmatrix} + [b_1 J \quad b_2 J \quad \cdots \quad b_c J] \\
&= WS + [b_1 J \quad b_2 J \quad \cdots \quad b_c J] \stackrel{\text{def}}{=} WS + B
\end{aligned} \tag{5}
$$

where S is a $r \times c$ matrix, s_{ij} is the number of molecules of the j-th protein at the i-th cell compartment, W is a $t \times r$ matrix where each column contains the flattened values of W_i, with J being a vector of length t with all elements equal to 1. B is a $t \times c$ matrix, each i-th columns contains t flattened values of $b_i J$.

We are interested in S. By summing elements within the columns of S, we obtain a c-dimensional vector $\vec{p}$ containing the total number of molecules of each protein:

$$
\vec{p} = \mathbf{1}_r^T S = [p_1 \quad p_2 \quad \cdots \quad p_c] \tag{6}
$$

$\vec{p}$ is the protein expression level that we want to quantify given V. As a multiplexed tissue imaging dataset contains multiple cells, by solving (5) iteratively over all the cells, we obtain the protein expression matrix:

$$
P = \begin{bmatrix} \vec{p_1} \\ \vec{p_2} \\ \vdots \\ \vec{p_n} \end{bmatrix} = \begin{bmatrix} p_{11} & p_{12} & \cdots & p_{1c} \\ p_{21} & p_{22} & \cdots & p_{2c} \\ \vdots & \vdots & \ddots & \vdots \\ p_{n1} & p_{n2} & \cdots & p_{nc} \end{bmatrix}
$$

where n is the number of cells, c is the number of protein, $p_{ij} \geq 0, \ \forall \ 0 \leq i \leq n, 0 \leq j \leq c$.

To construct P, we introduce a deep learning framework called Information-Controlled Variational Autoencoder (IC-VAE). The principal steps are as follows:

First, we design a VAE-based neural network that encodes the protein signal images of each cell to a latent distribution. From this latent distribution, we sample three latent subspaces:

- The protein expression latent subspace $Z_{expression}$.
- The cell compartments latent subspace $Z_{compartments}$.
- The background noise latent subspace $Z_{background}$.

We then decode the latent representations to obtain $W_{decoded}$, $S_{decoded}$, $B_{decoded}$ and reconstruct the original image as $W_{decoded}S_{decoded} + B_{decoded}$. The decoding process includes normalization steps: each compartment within $W_{decoded}$ is normalized to control for compartment-specific effects, and normalization across cells is applied to ensure comparability of the protein expression profiles between different cells. Moreover, to maintain the distinctiveness of each latent subspace, we incorporate a specialized neural network to calculate and minimize the Mutual Information (MI) among them. This step ensures that the subspaces contain non-redundant, independent features relevant to their respective roles in reconstructing the protein signal images.

2.1 IC-VAE Algorithm

Inputs

We describe how to construct the input for our method - IC-VAE, given a multiplexed tissue imaging dataset $\mathcal{D}$ containing c protein signal images of size $h \times w$. Since we aim to quantify the protein expression level for each cell in the dataset, we first need to identify individual cells on the image. Our method does not rely on membrane segmentation of the cell. Therefore, readers can apply state-of-the-art cell segmentation methods on the nucleus staining channel of the dataset. In this work, we use cellpose v2.2 [3], model $cyto2$, with default parameters to obtain the nucleus segmentation mask.

The nucleus segmentation mask $A = (a_{i,j})$ is a $h \times w$ matrix, where $a_{i,j} = 0$ indicates no cell detected at the pixel (i, j), $a_{i,j} = k$ indicates the pixel (i, j) belongs to cell k^{th}.

Using the nucleus segmentation mask A, for each cell, we extract an equal size $m \times m$ image for every protein, such that the center of the cell is at the position $(m//2, m//2)$ of the image. Then, we normalize the images to the range $[0, 1]$ and construct a tensor T of size $c \times m \times m$ representing one cell. The set $\mathcal{X} = \{T_1, T_2, ..., T_n\}$, with n is the number of cells in $\mathcal{D}$, is the input for our model.

For the training convention, we split $\mathcal{X}$ into two sets with the ratio $9 : 1$: the training set $\mathcal{X}_{training}$ and the validation set $\mathcal{X}_{validation}$. During training, we feed $\mathcal{X}_{training}$ to IC-VAE in batches of size N. Following the best practices, we shuffle $\mathcal{X}_{training}$ for each epoch and use the $\mathcal{X}_{validation}$ to validate the model after every epoch to avoid overfitting.

Notice that the choice of m varies between datasets. We want to select m large enough so that the biggest cell in $\mathcal{D}$ completely lies within the sub-image and is small enough for efficient training performance. In practice, we usually select $m = 128$.

Hyperparameters

Our method requires one hyperparameter r - which represents the number of cell compartments. We choose r such that $r << nc$. In practice, we choose $r = 4$ - under the intuition that $r = 4$ represents the cell nucleus, the cell cytoplasm, the cell membrane, and the rest.

Model architecture

Here, we outline the general architecture of IC-VAE. (Figure 1).

2.1.1 The encoder

The encoder receives a tensor of size $N \times c \times m \times m$ as input. The input is passed through a series of convolutional layers, followed by a linear layer to simultaneously obtain the μ and σ of the latent distribution. Using the reparameterization trick [4], we obtain the d-dimensional latent space $Z = \mu + \epsilon \cdot \sigma, \epsilon \in \mathcal{N}(0, 1)$. Z is a $N \times d$ matrix. Each row is a latent vector representation of the corresponding cell. Next, we split Z into three sub-matrices:

- A $N \times d_1$ matrix $Z_{expression}$ represents the d_1-dimensional protein expression latent subspace.
- A $N \times d_2$ matrix $Z_{compartment}$ represents the d_2-dimensional cell compartments latent subspace.
- A $N \times d_3$ matrix $Z_{background}$ represents the d_3-dimensional background noise latent subspace.

where $d = d_1 + d_2 + d_3$. Here, the choice of d_1, d_2, and d_3 will influence the resulting protein expression matrix. In practice, we usually choose $d_1 = d_2 = 2^{\lceil (log_2(c)) \rceil}$ and $d_3 = \frac{d_1}{2}$. Also, the number of convolutional layers and the number of in/out channels per convolutional layer varies between datasets. Empirical evidence from our experiments across several datasets suggests that a shallower encoder architecture tends to yield more desirable results in this specific application than a deeper one.

2.1.2 The decoders

Instead of having a single decoder, we design three decoders with different architectures to obtain the desired outputs:

Expression decoder
The expression decoder transforms $Z_{expression}$ to a $N \times rc$ matrix through a series of linear layers ending with the softplus activation function to ensure the non-negativity constraint. We reshape this matrix to a tensor $S_{decoded}$ of size $N \times r \times c$, each $r \times c$ matrix from this tensor is S from equation (5) of the corresponding cell.

Cell compartments decoder The cell compartments decoder transforms $Z_{compartments}$ to a tensor of size $N \times r \times m \times m$ through a series of convolutional layers, each sub-tensor $r \times m \times m$ is the matrix W from equation (5). We further pass W through a normalization layer to constrain the scale of W, which also constrains the scale of S. The cell compartments decoder also ends with a softplus activation function to ensure the non-negativity constraint and produce the final output $W_{decoded}$.

Background decoder The background decoder transforms $Z_{background}$ to a tensor $B_{decoded}$ of size $N \times c$ through a series of linear layers with no activation function layer at the end to allow the output to be at any scale. Besides using the output tensor to obtain B from equation (5), we also want to use this tensor as a bias for later reconstruction. One could represent the output of this layer as $B_{decoded} = B + Bias$.

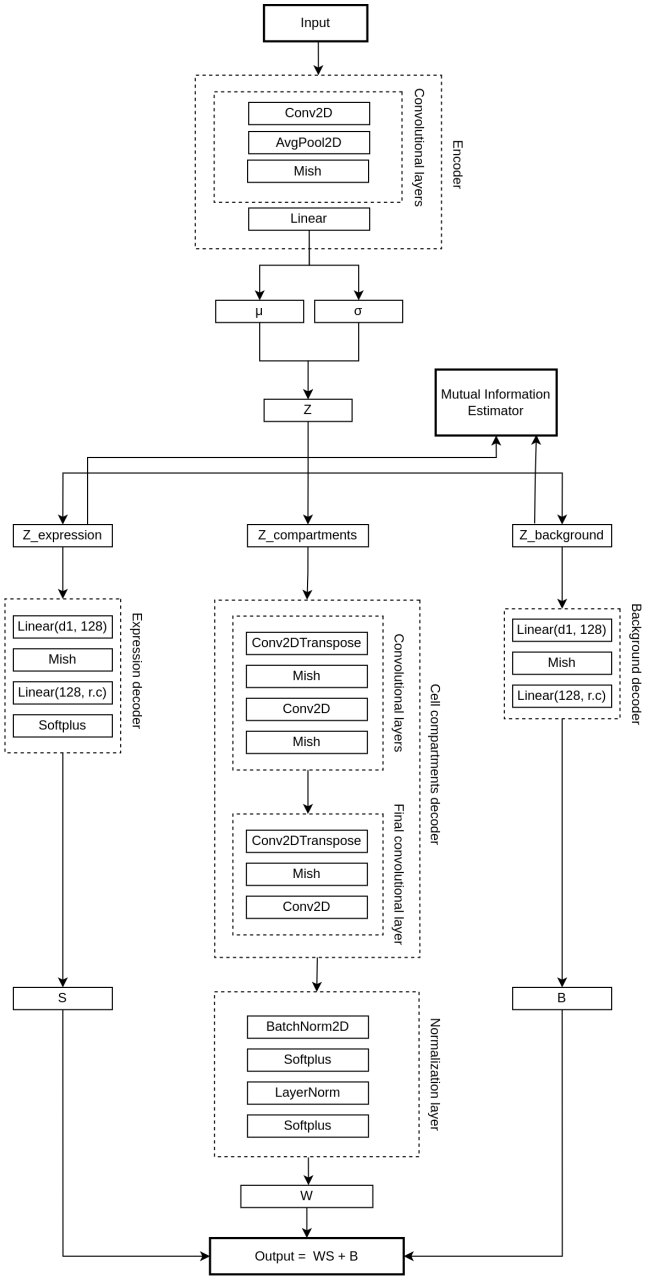


Figure 1: Information-Controlled Variational Auto-Encoder (VAE) based architecture for image analysis. The encoder consists of convolutional layers that process the input image and map it to a latent space representation, Z. The latent representation is then used by three separate decoders: the background decoder, the cell compartments decoder, and the expression decoder. Each decoder consists of convolutional layers and reconstructs different aspects of the image. The final output is normalized before being outputted. The Mutual Information Estimator component is used to control the shared information between $Z_{expression}$ and $Z_{background}$

.

Mutual Information We use MINE algorithm [5] to calculate the mutual information between $Z_{expression}$ and $Z_{background}$ with the goal of minimizing the shared information between them. Details about the MINE algorithm are described in the Supplementary Materials.

Reconstruction After having $S_{decoded}$, $W_{decoded}$, and $B_{decoded}$, we apply the Einstein summation with the convention $Ncm, Nmhw \rightarrow Nchw$ on $S_{decoded}$ and $W_{decoded}$ to obtain the output tensor O of size $N \times c \times m \times m$. Then, we

broadcast the tensor $B_{decoded}$ to a tensor of size $N \times c \times m \times m$ and add this tensor to O to obtain the reconstructed images.

$$Output = einsum('Ncm, Nmhw \rightarrow Nchw', S_{decoded}, W_{decoded}) + broadcast(B_{decoded}, (N, c, m, m))$$

Loss function

Our IC-VAE loss comprises the ELBO loss in variational inference and mutual information loss. The ELBO loss comprises the expectation of log-likelihood and the weighted KL divergence. The expectation of log-likelihood corresponds to the difference between the input and the reconstructed images from the latent representation. Weighted KL-divergence loss corresponds to the distance between the posterior and the chosen prior. This helps regularize the latent space.

We denote the prior of the latent space as $p(z)$, the likelihood of the sample given the latent representation as $p(x|z)$ computed from a decoder $D(z)$, the posterior of the latent space given the sample as $p(z|x)$

To make it feasible, we approximate the posterior $p(z|x)$ as $q(z|x)$ using an encoder $E(x) = (E_\mu(x), E_\sigma(x))$

We chose our prior $p(z)$ to be a normal distribution $z \sim \mathcal{N}(0, 1)$, our likelihood to be $x|z \sim \mathcal{N}(D(z), I)$ and our posterior $z|x \sim \mathcal{N}(E_\mu(x), \mathrm{diag}(E_\sigma(x)^2))$

$$\begin{aligned}\mathcal{L}_{\mathrm{IC-VAE}} &= \mathbb{E}_{q(z|x)}[\log p(x|z)] - \beta \mathrm{KL}(q(z|x)||p(z)) + \gamma \mathcal{L}_{\mathrm{MI}} \\ &= \frac{1}{2}\mathbb{E}_{z\sim q(.|x)}\left[\|x - D(z)\|_2^2\right] - \frac{1}{2}\beta\left(\|E_\sigma(x)\|_2^2 + \|E_\mu(x)\|_2^2 - 2\sum_i \ln(E_\sigma(x)_i)\right) + \gamma\mathcal{L}_{\mathrm{MI}}\end{aligned} \tag{7}$$

We denote the parameterized mutual information estimator as $MI(z_{\mathrm{exp}}, z_{\mathrm{bg}})$.

We define the mutual information loss, adopted from [6], the Jensen-Shannon Divergence as,

$$\mathcal{L}_{\mathrm{MI}} = \mathbb{E}_{z\sim q(.|x), z'\sim q(.)}\left[-\mathrm{softplus}(-MI(z_{\mathrm{exp}}, z_{\mathrm{bg}})) - \mathrm{softplus}(-MI(z_{\mathrm{exp}}, z'_{\mathrm{bg}})) - MI(z_{\mathrm{exp}}, z'_{\mathrm{bg}})\right] + \mathrm{const} \tag{8}$$

For each mini-batch, we first optimize $\mathcal{L}_{\mathrm{IC-VAE}}$ while fixing the parameters of MI, then we optimize $\mathcal{L}_{\mathrm{MI}}$ while fixing the parameters of the decoder D and the encoder E.

3 Benchmarks

For our experiment, we compare IC-VAE against MCMICRO. To assess the efficiency of each method, we use the protein expression matrix that both methods generated as input for evaluating some common downstream analysis tasks, including analyzing the expression level of a protein, clustering cells for cell typing, and presenting the cell clusters on UMAP [7] plots.

3.1 Dataset description

We use the TONSIL-1 40X image of the first batch from a human tonsil cancer t-CyCIF dataset [8]. The image is a whole-slide 2-D image of size $16,843 \times 16,125$ pixels. It includes 44 different staining channels for cell nuclei, background control, and protein biomarkers. Imaging parameters and preprocessing methodology (e.g., background and shading correction, stitching, and registration) are described in [8].

3.2 Data pre-processing

3.2.1 Cell segmentation

We select the first nucleus staining channel to perform cell segmentation. We use cellpose v2.2 [3] with $model = cyto2$ and $diameter = None$. Other parameters are kept as default. The resulting segmentation masks are later used as input for MIMICRO and IC-VAE.

3.2.2 Cell-type labeling

To evaluate the clustering and dimensional reduction results, BioTuring's pathologist manually annotates the cell type labels for three different sub-regions. The detailed description is presented in the supplementary materials.

3.2.3 MCMICRO settings

We run the MCMICRO pipeline - revision 01c11e5615 (https://github.com/labsyspharm/mcmicro) using Nextflow version 23.04.4. For each dataset, we use the OME-TIFF whole-slide 2-D image and the cell segmentation mask from 3.2.1 as the input. Since we already have the processed image and cell segmentation, we configure the pipeline to start at the $quantification$ step and stop at the $downstream$ step. Other parameters are kept as default. The pipeline produces a CSV file containing the protein expression matrix.

3.2.4 IC-VAE settings

We set the number of cell compartments $r = 4$. The mutual information weight γ is initialized with the value $1e{-}5$ and increased by $1e{-}5$ for every epoch until it reaches $1e{-}3$. The KL weight β is initialized with $1e{-}5$ and increased by $1e{-}5$ for every epoch until it reaches $1e{-}4$. We select the dimension for the latent subspaces as $d_1 = 32$, $d_2 = 32$, $d_3 = 16$. We train the model with $batch\_size = 128$ and $n\_epochs = 100$. If the validation loss is not increased after 10 epochs, we stop the training and use the best checkpoint to evaluate the entire input $\mathcal{X}$.

3.2.5 Downstream analysis

We use standard scanpy [9] pipeline for the downstream analysis benchmarks, including:

- PCA: For MCMICRO, we use PCA with default parameters. For IC-VAE, as described in section 2, the encoder produces an expression latent space. We utilize this latent space as a compressed representation of the data (similar to PCA).
- k-NN with $n\_neighbors = 30$. Other parameters are kept as default.
- UMAP with $min\_dist = 0.1$. Other parameters are kept as default.
- Leiden clustering with $resolution = 1.0$. Other parameters are kept as default.

Then, we perform the clustering and UMAP on the labeled cells from selected regions only. We combine three different regions to evaluate the ability to overcome the technical variants among cells from different field-of-views between MCMICRO and ICVAE.

3.3 Experimental results

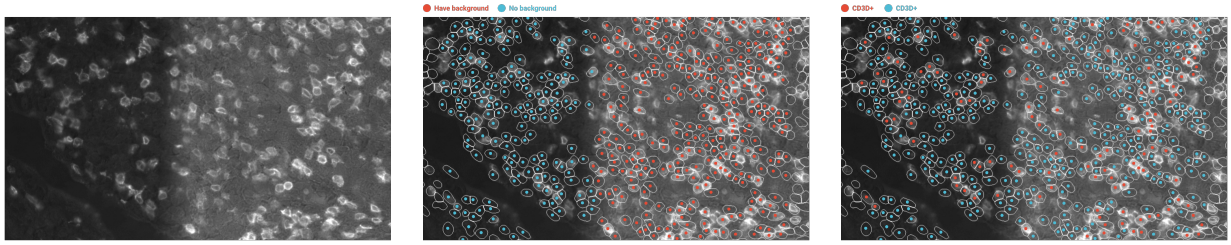
3.3.1 Analyzing the expression level of a protein

To evaluate the ability to overcome the background noise and imaging artifacts of the MCMICRO and IC-VAE, we select a region from the evaluated dataset where the signal of $CD3D$ is highly affected by the aforementioned factors. As shown in Figure 2a, the right-hand side part of this region contains the background signal of $CD3D$, while the left side does not. We create a label called "Background region" and annotate the cells on the left side as "no background" and the cells on the right side as "have background" (Figure 2b). We also create a label called "CD3D expression" and annotate the cells in this region as $CD3D^+$ - if the cell has strong $CD3D$ signals surrounding its nucleus - otherwise, the cell is annotated as $CD3D^-$ (Figure 2c).

Figure 3a indicates the background signal heavily influences the expression of $CD3D$ quantified by MCMICRO. From the MCMICRO result, the expression of $CD3D$ in $CD3D^-$ cell population inside of the background noise region is even higher than the expression of $CD3D$ in $CD3D^+$ cell population outside of the background noise region (Figure 4a). On the other hand, in the result of IC-VAE model, the influence of the background signal on the $CD3D$ expression is reduced (Figure 3b), the expression of $CD3D$ in $CD3D^-$ inside of the background noise region is lower than the expression of $CD3D$ in $CD3D^+$ cell population outside of the background noise region (Figure 4b).

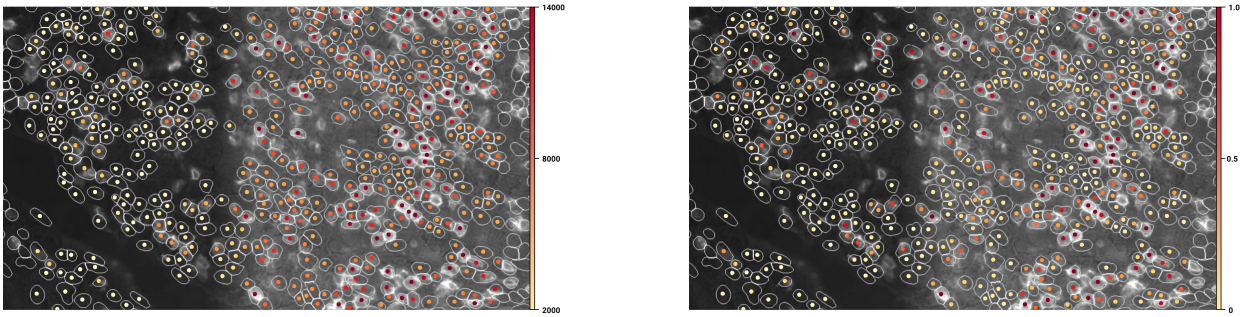
3.3.2 Clustering cells for cell typing

In this experiment, we extract cells from three labeled regions and perform the clustering analysis as described in 3.2.5. Ideally, cells within a cluster should be the same cell type. We first evaluate clustering results using RandIndex [10] - given the cell-type annotation as the ground truth. As shown in Table 1, the RandIndex between the cell type label



(a) (b) (c)

Figure 2: The region is highly affected by imaging artifacts. (a) The fluorescent image of $CD3D$. (b) Cells labeled by artifacts region. The blue dots stand for cells in the region without the background signal. The red dots stand for cells in the region with the background signal. (c) Cells labeled by protein expression. The blue dots stand for $CD3D^-$ cells and the red dots stand for $CD3D^+$ cells. The label is manually annotated with the same cell typing procedure described in the Supplementary Material.



(a) MCMICRO (b) IC-VAE

Figure 3: The $CD3D$ expression quantification in the regions with and without background signal. (a) From the result of the MCMICRO, cells in the region with the high background signal always have a high expression value, regardless of whether they belong to the CD3+ or CD3- groups (b) The expression values from IC-VAE model are not affected by the background signal.

and IC-VAE clustering results are higher than the RandIndex between the cell type label and MCMICRO clustering results on all regions. This indicates that IC-VAE clustering results are more similar to the cell type annotations than MCMICRO clustering results.

Regions	MCMICRO	IC-VAE
Region 1	0.2185	**0.2811**
Region 2	0.1324	**0.3054**
Region 3	0.1951	**0.3685**
Region 1 & 2 & 3	0.1331	**0.2273**

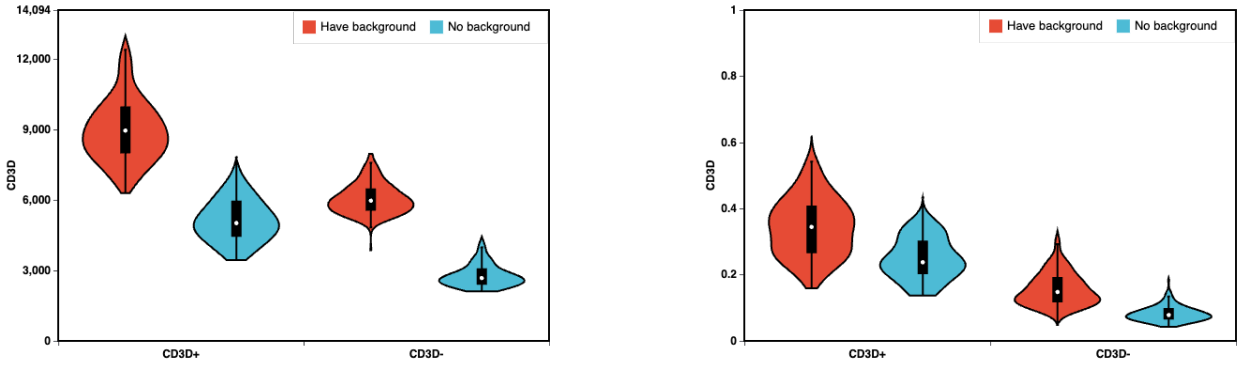
Table 1: RandIndex between cell type labels and clustering result. IC-VAE outperforms MCMICRO in individual regions and in the combined region.

	MCMICRO	IC-VAE
Score	1.0779	**1.600**

Table 2: The ratio between $out\_group\_distance$ and $in\_group\_distance$ on the UMAP. If cells of the same cell type stay closer to each other, while cells of different cell types are more distinctly separated on the UMAP, the ratio will be higher

We then further evaluate the clustering specificity to different cell type labels by:

- Assign all cells in a cluster with the label of the largest cell type population.



(a) MCMICRO (b) IC-VAE

Figure 4: The protein expression distribution of $CD3D$ between two groups CD3D+/CD3D-, divided by artifact region label. According to the result of MCMICRO, cells in the CD3- group located in regions with background signal even exhibit a higher level of protein expression compared to cells in the CD3+ group situated in regions without background signal

- For each cell type A, we create two binary vectors: the $truth\_vector$, derived from the actual truth labels, and the $predict\_vector$ formed using the newly assigned labels. In these vectors, the $TRUE$ value is assigned to cells labeled as A, and the rest are $FALSE$.
- Calculate the F1-score between each pair of $truth\_vector$ and $predict\_vector$

As shown in Figure S3 and Table S2, the IC-VAE model has a higher specificity for cell type clustering. MCMICRO fails to separate the $epithelialcell$ population from other cell types. Both methods fail to cluster the $fibroblast$ and $regulatory\ T\ cell$ populations out of the remaining cells.

3.3.3 Visualize cell clusters on the UMAP

Similar to clustering, for the UMAP results, we expect cells from the same cell type label to be close together on the 2-D coordinates, while cells from different labels should be far apart. We evaluate the result for each cell type by:

- Normalize the UMAP coordinates to range from 0 to 1 for both x and y.
- Compute the centroid coordinates for each cell type based on the scaled UMAP coordinates of the cells within that cell type
- For each cell, determine its distance to the centroid of its respective cell type group, which is referred to as the $in\_group\_distance$. For $out\_group\_distance$, calculate the distance from the cell to the centroids of all other cell type groups
- The final score is defined as $median(out\_group\_distance)$ / $median(in\_group\_distance)$. The larger the score, the better the performance.

The results from Figure 5 and Table 2 indicate the better performance of the IC-VAE model. In the UMAP representation generated by IC-VAE, different cell-type clusters exhibit a greater degree of separation, with particular ease in distinguishing $myeloid\ leukocyte$ and $lymphoma$ clusters. In contrast, cells of the same cell type on the UMAP generated by MCMICRO form multiple clusters ($B\ cells$ are split into 3 clusters, $myeloid\ leukocytes$ are split into 2 clusters and $lymphomacells$ are also split into 2 clusters). This observed phenomenon might be explained by the technical variance discussed in section 3.3.1. Furthermore, in the UMAP representation produced by MCMICRO, the $epithelial\ cell$ population appears to be intermixed with a substantial portion of $myeloid\ leukocytes$ and $B\ cells$. This outcome aligns with the clustering results from MCMICRO, where the epithelial cell population is also inseparable.

On both clustering and UMAP results, the IC-VAE model (and MCMICRO) fails to cluster the $regulatory\ T\ cell$ and $fibroblast$ population. To investigate, we visualize the expression matrix generated by IC-VAE (Figure S4). We found that the model had difficulty identifying the expression signal of $FOXP3$. Upon comparing this expression result with the original image, we observed that this was due to the fact that, unlike other marker expressions that exhibit distinct cellular patterns, $FOXP3$ expression was uniformly distributed within the cells (Figure 6. This uniform distribution closely resembled the background pattern, causing confusion for the model. These findings introduce new challenges that should be addressed in future works.

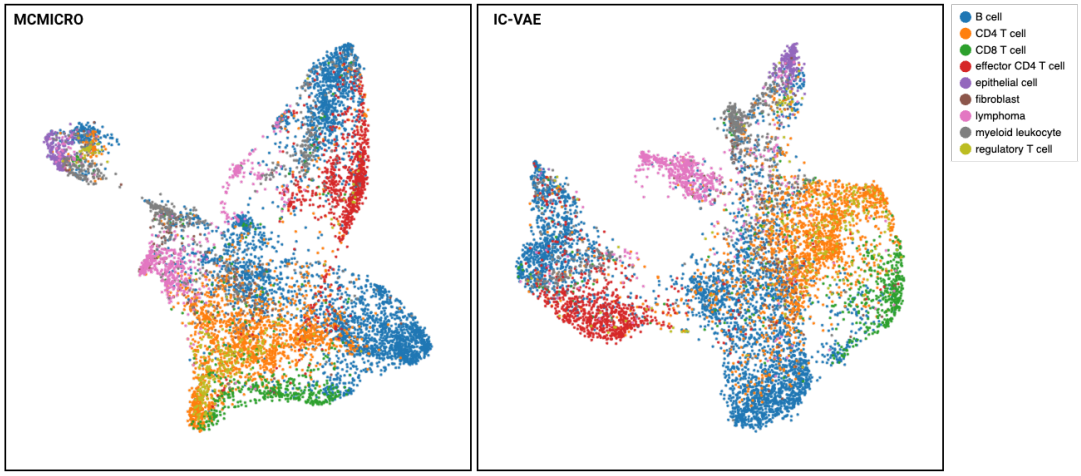


Figure 5: Comparative visualization of cell type distributions using UMAP. The UMAP plot from MCMICRO and IC-VAE with each color representing a different cell type. UMAP plot using IC-VAE is much more consistent with cell type annotations.

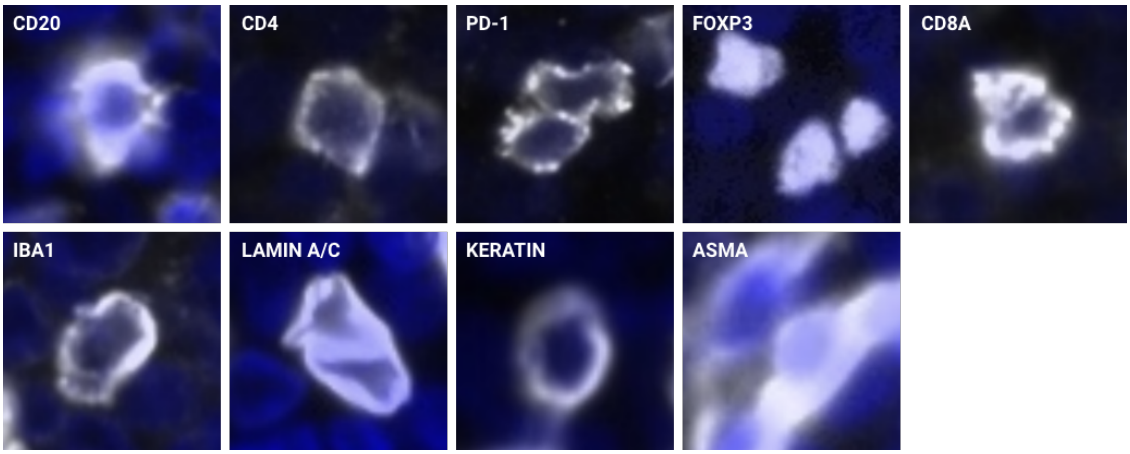


Figure 6: $FOXP3$ signal has a different pattern compared to other proteins, which leads to an incorrect estimation of $FOXP3$ expression in both methods.

4 Discussion

In this work, we introduce Information-Controlled Variational Autoencoder (IC-VAE). This leads to a significant improvement in interpreting protein expression from multiplexed tissue imaging data. The IC-VAE surpasses traditional approaches by allowing for controlled shared information among latent subspaces, which effectively separates true protein expression from cellular components and background noise. Unlike methods that rely on assumptions about cell symmetry and uniform protein distribution, IC-VAE accommodates the diverse cell shapes and intracellular structures resulting from 2-D tissue sectioning. Additionally, as our method just needs nucleus segmentation, it bypasses the need for adopting cell membrane segmentation, which usually provides inaccurate results. [1]

The model provides more accurate protein quantification compared to the state-of-the-art tool MCMICRO, especially in densely packed cell regions where signals from neighboring cells often confound traditional analysis. We further showed that downstream applications, such as biomarker analysis, cell typing, and cell cluster visualization with t-SNE/UMAP, benefit from the precision of IC-VAE.

[1] Currently, many cell segmentation algorithms yield robust results for nucleus segmentation, but not for membrane segmentation.

References

[1] J Kennedy-Darling, G Dakshinamoorthy, J Singh, S Mistry, N Nikulina, and C Streck. Po-281 automated multiparametric tissue imaging platform using existing microscope hardware for the detection of spatially resolved single-cell resolution data. *ESMO Open*, 3:A337–A338, 2018.

[2] Denis Schapiro, Artem Sokolov, Clarence Yapp, Yu-An Chen, Jeremy L Muhlich, Joshua Hess, Allison L Creason, Ajit J Nirmal, Gregory J Baker, Maulik K Nariya, et al. Mcmicro: a scalable, modular image-processing pipeline for multiplexed tissue imaging. *Nature methods*, 19(3):311–315, 2022.

[3] Marius Pachitariu and Carsen Stringer. Cellpose 2.0: how to train your own model. *Nature methods*, 19(12):1634–1641, 2022.

[4] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[5] Mohamed Ishmael Belghazi, Aristide Baratin, Sai Rajeshwar, Sherjil Ozair, Yoshua Bengio, Aaron Courville, and Devon Hjelm. Mutual information neural estimation. In *International conference on machine learning*, pages 531–540. PMLR, 2018.

[6] R Devon Hjelm, Alex Fedorov, Samuel Lavoie-Marchildon, Karan Grewal, Phil Bachman, Adam Trischler, and Yoshua Bengio. Learning deep representations by mutual information estimation and maximization. *arXiv preprint arXiv:1808.06670*, 2018.

[7] Leland McInnes, John Healy, and James Melville. Umap: Uniform manifold approximation and projection for dimension reduction. *arXiv preprint arXiv:1802.03426*, 2018.

[8] Rumana Rashid, Giorgio Gaglia, Yu-An Chen, Jia-Ren Lin, Ziming Du, Zoltan Maliga, Denis Schapiro, Clarence Yapp, Jeremy Muhlich, Artem Sokolov, et al. Highly multiplexed immunofluorescence images and single-cell data of immune markers in tonsil and lung cancer. *Scientific data*, 6(1):323, 2019.

[9] F Alexander Wolf, Philipp Angerer, and Fabian J Theis. Scanpy: large-scale single-cell gene expression data analysis. *Genome biology*, 19:1–5, 2018.

[10] William M Rand. Objective criteria for the evaluation of clustering methods. *Journal of the American Statistical association*, 66(336):846–850, 1971.

SUPPLEMENTARY

1 MINE algorithm

Algorithm 1 MINE

$\theta \leftarrow$ initialize network parameters
repeat
  Draw b minibatch samples from the joint distribution:
  $(x^{(1)}, z^{(1)}), \ldots, (x^{(b)}, z^{(b)}) \sim P_{XZ}$
  Draw b samples from the Z marginal distribution:
  $z^{(1)}, \ldots, z^{(b)} \sim P_Z$
  Evaluate the lower-bound:
  $V(\theta) \leftarrow \frac{1}{b}\sum_{i=1}^{b} T_\theta(x^{(i)}, z^{(i)}) - \log(\frac{1}{b}\sum_{i=1}^{b} e^{T_\theta(x^{(i)}, z^{(i)})})$
  Evaluate bias corrected gradients (e.g., moving average):
  $G(\theta) \leftarrow \nabla_\theta V(\theta)$
  Update the statistics network parameters:
  $\theta \leftarrow \theta + G(\theta)$
until convergence

2 Cell type labeling

We annotate cell types based on the segmentation result and each individual image channel. From the list of input proteins, we define cell type using the markers listed in Table S1. The procedure for labeling each cell type is as below:

- Render the cell center onto the image, which is a combination of 1 DAPI channel and corresponding cell type biomarker channels (Figure S1)
- Select the cells that have the protein signal forming a complete cell structure (Figure S1a) and skip cells with no detectable signal (Figure S1b) or cells with noise/un-structured signal (Figure S1c)

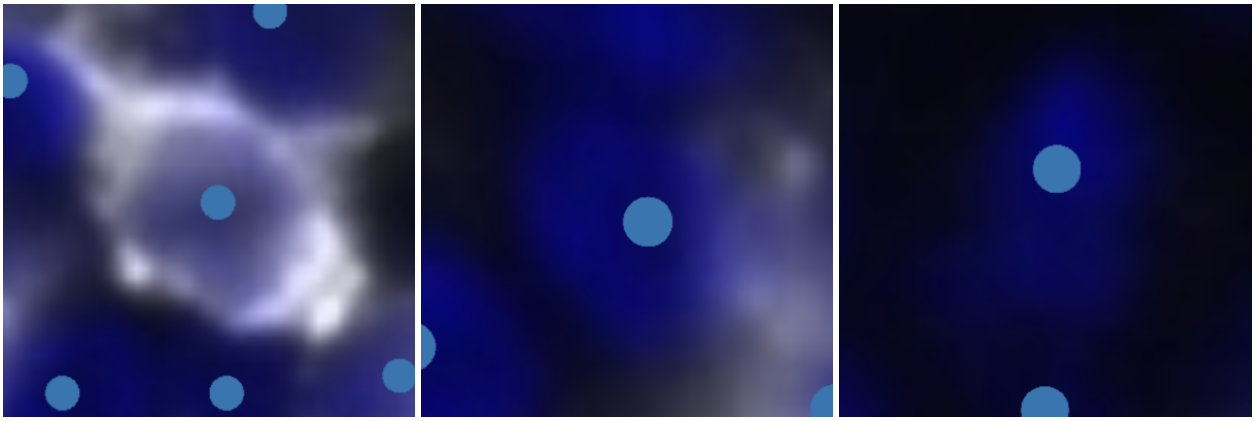
The image channel for each biomarker across three selected region are shown in Figure S1.

3 Clustering evaluation result

4 Marker expression heatmap

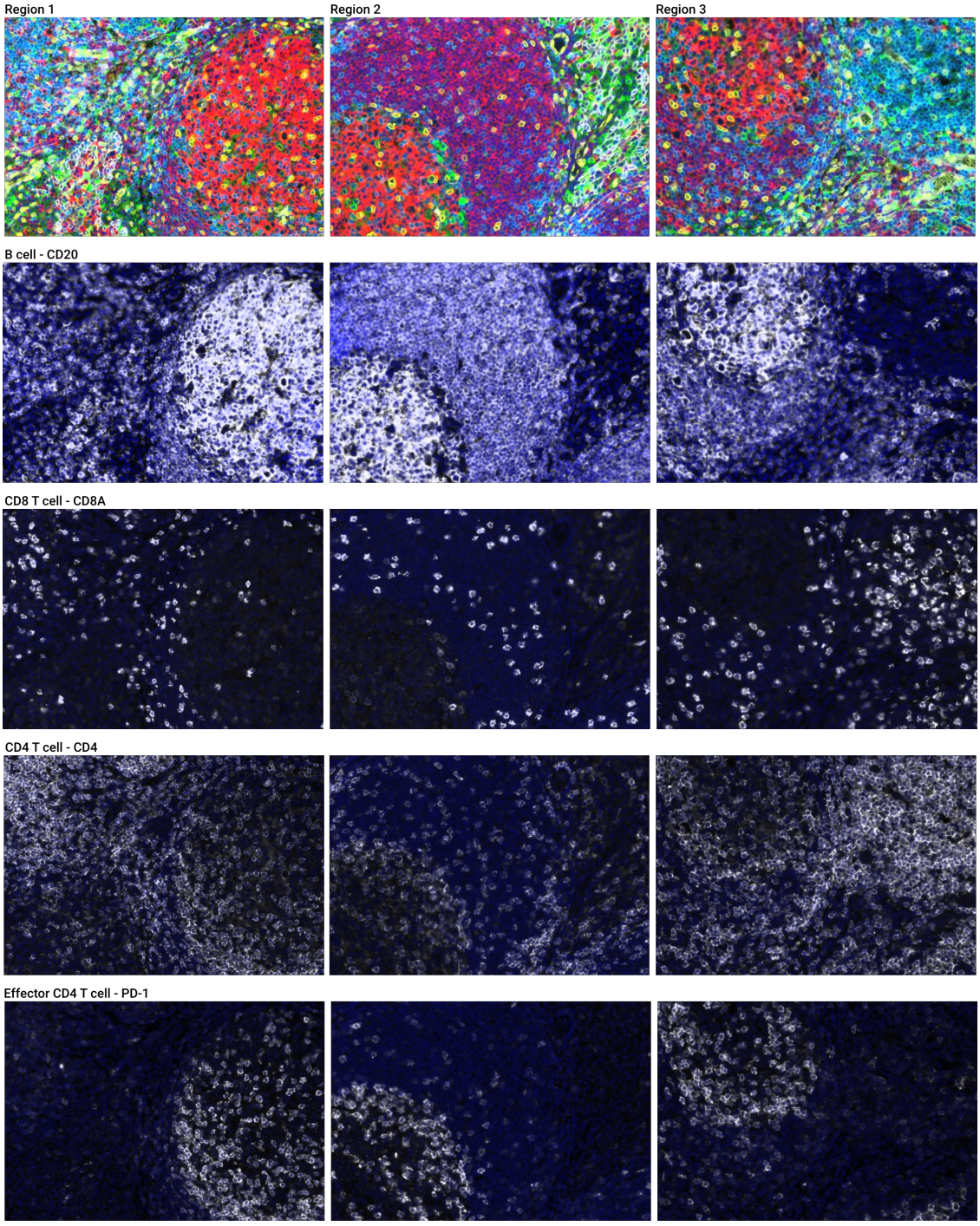
Cell type	Biomarkers
B cell	CD20
CD4 T cell	CD3D, CD4
effector CD4 T cell	CD3D, CD4, PD-1
regulatory T cell	CD3D, FOXP3
CD8 T cell	CD8A
myeloid leukocyte	IBA1
lymphoma	LAMIN A/C
fibroblast	ASMA
epithelial cell	KERATIN

Table S1: Biomarkers used for label each cell type



(a) True signal (b) Noise (c) No signal

Figure S1: Distinguish between a cell's true expression signal, background noise, and no signal. True cell type signal exhibits a consistent and complete cellular structure, which remains similar across cells of the same type. In contrast, noise signal displays a random pattern that varies from cell to cell and lacks the coherence seen in true cell type signal.



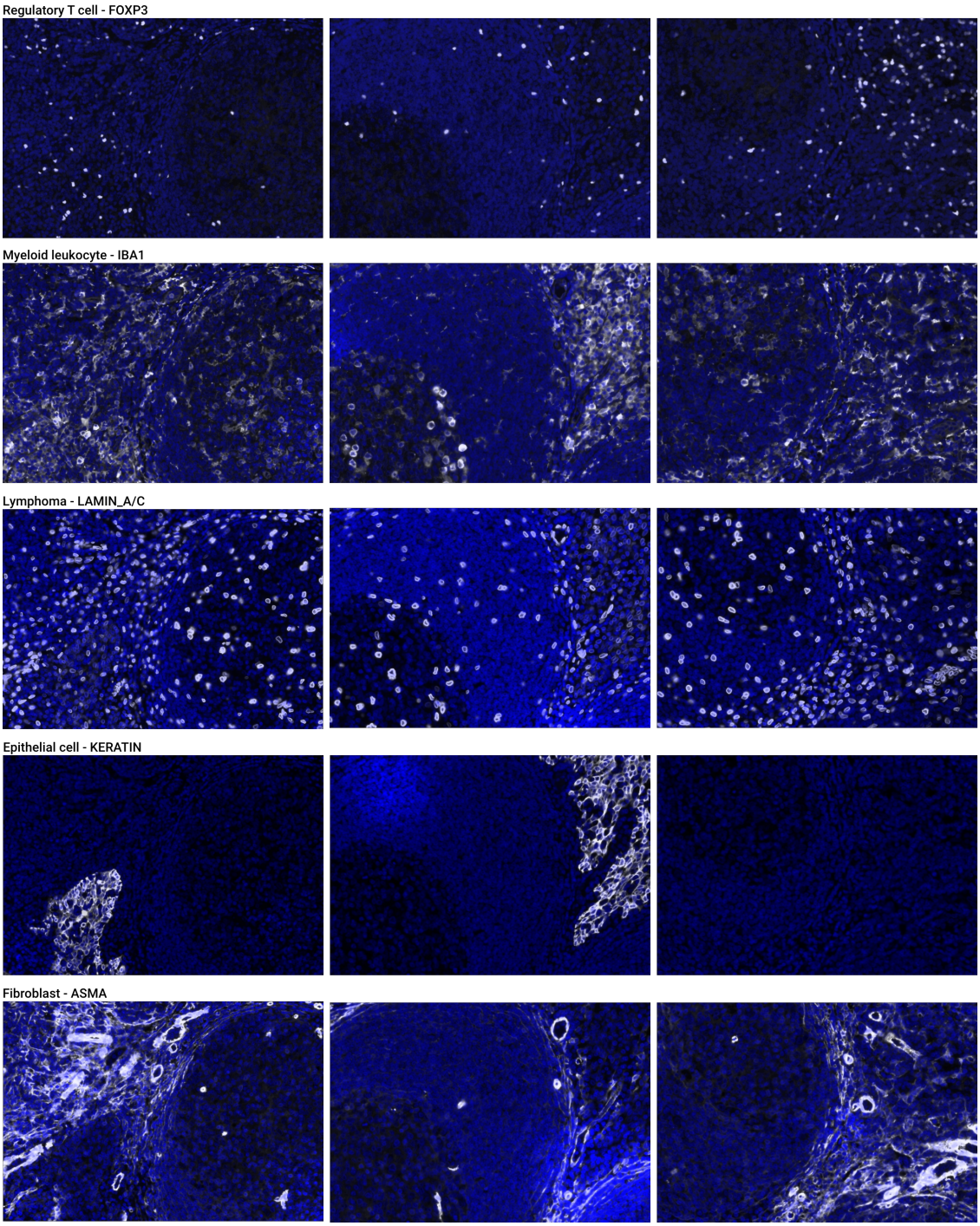


Figure S2: Protein channels used for manually labeling each cell type

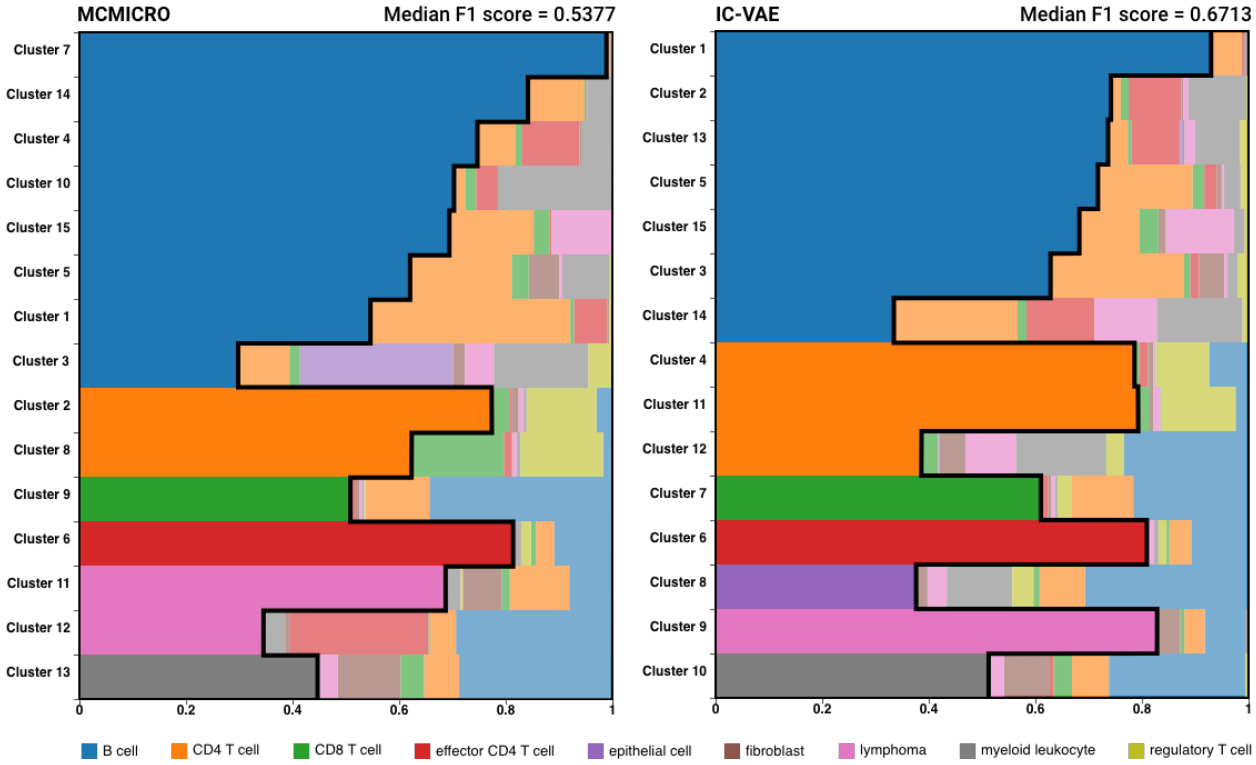


Figure S3: Composition plot between cell type label and clustering result of MCMICRO and IC-VAE. We then assign all cells from the cluster with the label of the largest cell type proportion. The epithelial cell population is missing from the MCMICRO result.

Cell type	MCMICRO	IC-VAE
B cell	0.7405	**0.7700**
CD4 T cell	0.6044	**0.6542**
effector CD4 T cell	0.7356	**0.7785**
regulatory T cell	0.0000	0.0000
CD8 T cell	0.5377	**0.6883**
myeloid leukocyte	0.3521	**0.4467**
lymphoma	0.6590	**0.7853**
fibroblast	0.0000	0.0000
epithelial cell	0.0000	**0.5452**
Median F1-score	0.5377	**0.6713**

Table S2: F1-score for each cell type between the truth label and the clustering assignment result

Figure S4: The heatmap of IC-VAE protein expression across cell type. FOXP3 shows no significant expression.