CARNEGIE MELLON UNIVERSITY

DOCTORAL THESIS

Improving performance on unsupervised biological tasks with hybrid models

Author:
Haotian TENG

Thesis Comittee:
Dr. Ziv BAR-JOSEPH, Chair
Dr. Carl KINGSFORD
Dr. David Ryan KOES
Dr. Ye YUAN

Submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy

in the

Systems Biology Group
Computational Biology Department
School of Computer Science

November 29, 2022

CARNEGIE MELLON UNIVERSITY

Abstract

Faculty Name
Computational Biology Department
School of Computer Science

Doctor of Philosophy

Improving performance on unsupervised biological tasks with hybrid models

by Haotian TENG

Many fundamental biological tasks require unsupervised learning where ground truth labels are unavailable, but shallow unsupervised machine learning methods have poor performance on these tasks due to the complexity of the problem. Deep learning models, with their strong representation power, have been widely applied to solve challenging tasks; however, they usually require large amounts of labeled data. To take advantage of the strong representation power of deep learning while applying them to unsupervised tasks, we developed several hybrid models that combine deep neural networks and unsupervised machine learning models. We used these models to improve performance on unsupervised biological tasks, including cell type clustering, basecalling, and lead optimization. First, we present an unsupervised cell type clustering model for recently developed single-molecule spatially resolved transcriptomics data, where a deep Neural Network (NN) encoder is used to generate low-dimensional Gaussian distributed gene embedding, so it can be combined with the spatial relationship using a Gaussian-Multinomial Mixture Model developed by us to predict the cell-type clustering. The second problem we try to tackle is to call m6A methylated bases in RNA generated from long-read sequencing. m6A modification plays essential roles in regulating gene expression while lacking an efficient way to detect it systemically. The long-read sequencing from Oxford Nanopore Technologies has been shown to be sensitive to post-transcriptional modification, but an m6A sensitive basecaller for directly detecting this subtle sequencing signal has not yet been developed. We used a CNN-RNN (Convolutional-Recurrent Neural network) model previously developed by us for canonical basecalling to train a Non-homogeneous HMM (NHMM) where its transition matrix is conditioned on the deep NN output. Using the hybrid synthetically m6A methylation data sampled from the NHMM, we were able to train a NN basecaller to call m6A base. We applied our method to call the methylome on Yeast RNA without requiring knock-out comparison data. For the third application, I propose a deep generative model with a deep Graph Neural Network and diffusion model to lead optimization problems in drug discovery, where the binding affinity is unreachable, and the deep learning model is suitable to deal with the complexity of the problem.

Contents

List of Figures

List of Tables

List of Abbreviations

NHMM	**N**on-homogeneous **H**idden **M**arkov **M**odel
ONT	**O**xford **N**anopore **T**enologies

Chapter 1

Introduction

Deep learning methods have been widely used to tackle challenges in computational biology, including the recently succeed Alphafold2[3] which predict accurate protein structure given the protein sequence, alphafold achieved the accuracy by sophisticatedly network structure design and trained on large amount of training dataset, however, many questions in computational biology are presented in an unsupervised manner[4], where true labels are usually beyond the reach, thus supervised deep learning methods that require large amounts of labeling data are not directly applicable. Hybrid model that combine classical machine learning methods and deep learning are actively explored, one type of model applied classical clustering on a dimensional-reduced representation, the representation can be generated by a deep neural network through self-training including Denoise-Autoencoder[5] transformer-based encoder-decoder model[6], a process where a hidden representation is projected from the original or corrupted input sample, and the model is trained to reconstruct the input samples totally or partially from this projected hidden representation which is also called embedding. The model is trained by minimizing the reconstruction error so the underlying distribution of input samples are captured by the embedding of input samples, and the embedding can be used as the input with few-shot learning when there are very few labels or with clustering method when there is no label. Another way is through generative models such as probabilistic graphical models, probabilistic graph models are hard to apply as they suffer from intractable exact posterior inference, variational inference is used with a posterior distribution usually approximated by a neural network[7]. Hybrid model have achieved promising result in unsupervised and semi-supervised biological task, for example, Autoencoder-style models are used to generate low-dimensional gene embedding for single-cell RNA expression data[8-14], the gene embedding generated are then be used in afterward tasks such as cell type clustering[15]. In this thesis we designed and applied several new hybrid models to solve several different biological tasks including cell clustering on spatial transcriptomics data, post transcriptional modification detection on long read sequencing platform and drug discovery.

1.1 Cell clustering

Cell clustering is a process where the cell is assigned to several groups based on their gene expression profile, it's a fundamental biological task and is requested by many downstream analyses in single-cell RNA sequencing[16]. Recent advances in Fluorescence in-situ Hybridization (FISH) technique enable recording a single cell level spatial transcriptomics for large numbers of cells[1,2,17], however, scRNA pipeline is usually used to analyze this data where spatial information is not taken into account when conduct cell clustering, so we developed a mixture model with denoise

autoencoder embedding called FICT which combines both expression and neighborhood information when assigning cell type.

1.2 RNA modification detection using Oxford Nanopore sequencing

RNA modification played an important role in various biological processes including stem cell differentiation and renewal, brain function, immunity and cancer progression [18], among the several RNA modification, N6-Methyladenosine (m6A) is one of the most abundant modification, involved in mRNA expression, splicing, nuclear export, translation efficiency, RNA stability and miRNA processing[18]. Among several m6A detection methods, long read sequencing using Oxford Nanopore sequencer is a way that can give qualitative information about the whole m6A methylome, and with the potential to detect single-molecule read level modification, however it requires optimized detection methods to call the m6A information from the subtle signal change. There are several m6A datasets available, In-vitro transcription dataset is made from synthetic sequence [19,20], by introducing only canonical adenine or modified adenine when doing in-vitro transcription, all modified or non-modified datasets are created. However, training a basecaller directly on the non-or-whole modified read would fail on basecall modified state on reads sequencing from real biological samples, as the modification state is usually mixed in one read. So several methods address this problem by training a local kmer model, where a classifier is trained to call modification on segmented signal segments. The performance of the trained classifier relies on a good post-segmentation of the sequencing signal which is usually done by a HMM [21], whose performance is limited by its parameters learning from canonical RNA sequencing. Data generated from antibody capturing techniques [22,23] can only provide site-level modifications whereas read-level modifications are unknown. We developed a new kind of hybrid model where a NHMM is trained semi-supervised by conditioning on the output from a deep learning network, the modified and canonical reads are then segmented by the trained NHMM and added into a graph, reads with mixing modification state then sampled from the graph to produce a training dataset, this step can be seen as a data augmentation that eliminate the inductive bias when trained using homogeneous modified read, we then trained a new deep learning model on this dataset which give accurate methylation basecalling.

1.3 Drug discovery with deep learning

Drug discovery is one of the most important Related Biological technology

Chapter 2

Clustering Spatial Transcriptomics

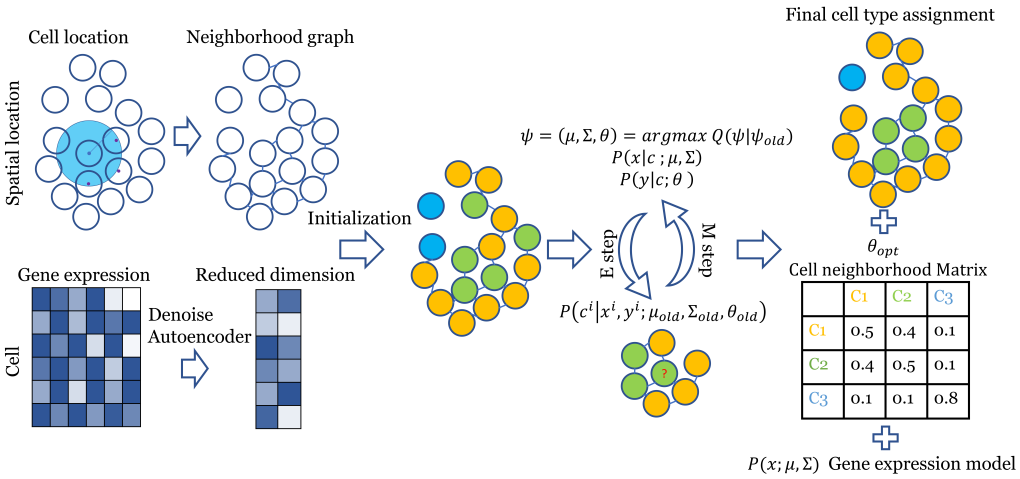

FIGURE 2.1: FICT pipeline. A reduced dimension expression profile is generated using a Denoising Autoencoder **[vincent2008extracting]**, and an undirected graph is constructed according to the spatial locations information. Cells are initially clustered using an expression only GMM. Next, the the model is iteratively optimized using an EM algorithm to improve the joint likelihood of the expression and neighborhood models given both the gene expression representation and the spatial graph. The final output is an assignment of cells to clusters, a Gaussian gene expression model and a Multinomial neighborhood model for each class.

We developed a joint expression and location clustering method called FICT to infer cell types in spatial transcriptomics studies. A generative mixture model is defined firstly: each cell is assigned a cell type given its neighborhood, and then the dimension reduced representation of gene expression levels are drawn from cell-type specific distribution. We next learn the parameters of this generative model by maximizing the joint likelihood of gene expression and cell location (Figure **??**). The cell type is then inferred by the posterior distribution of this generative model given the gene expression level and cell location. To test the method we used both simulated and real single cell spatial transcriptomics data.

2.0.1 A generative model for spatial transcriptomics data

We assume an undirected, weighted graph G representing cell neighborhoods. Each node in G is a cell. We assume a total of M cell types in the dataset. We denote by Z the cell type assignments for nodes in G where z^i is the cell type of cell i and denote

by $X = x^i$ the gene expression matrix. Here, x^i is the gene expression levels vector for cell i and X is the gene expression matrix for the expression of all cells. Finally, we define the neighbors of cell i in G using $N_{G(i)}$. Neighborhood is either defined using the k nearest neighbors (we used k=10 in this paper) or a cutoff on the distance between i and other nodes in G (a cutoff on the edge weight). Using these definitions we assume the following generative model for a single cell transcriptomics dataset.: (1) First, a cell type is selected according to $P_\theta(Z) \propto \prod_i P(z^i|N_G(z^i))$, in which $P(z^i|N_G(z^i))$ is the conditional distribution for the assignment of cell i given its neighborhood capturing the relationship between neighboring cells $N_G(z)$ in G. (2) Next, expression levels **X** are generated according to a cell type specific probability distribution $P(x^i|z^i)$.

Given this model the likelihood of a dataset with a set of gene expression levels X and cell locations (G) is:

$$P(X) = \sum_Z (P(X|Z) \cdot P(Z)) \propto \sum_{z \in Z} (\prod_i P(x^i|z^i) P_\theta(z^i, N_G(z^i))) \tag{2.1}$$

We use a multinomial distribution to model the relationship with neighborhood cells and so the product of the conditional probability can be written as:

$$P_\theta(z^i, N_G(z^i)) = P(y^i|z^i) \tag{2.2}$$

Where y^i is a vector summarizing the cell type assignments for neighbors of i. Specifically, y^i is of dimensions M (number of cell types) and each entry j demotes the number of neighbors of cell i assigned to cell type j. Combined, the overall likelihood function is:

$$P(X, Y) = \prod_{i=1}^{D} \sum_k P(z^i = k) P(x^i|z^i = m) P(y^i|z^i = m) \tag{2.3}$$

Where X is the dimension reduced gene expression matrix and Y is the neighborhood cell type count matrix for each cell, m denotes the m_{th} cell type, we also change the order of product and sum as y is now treated as a property of the cells. We assume that $P(x^i|z^i = k)$ follows a Gaussian distribution and $P(y^i|z^i = k)$ follows a multinomial distribution.

2.0.2 Inferring cell types (E-step)

We use an Expectation Maximization (EM) approach to learn the parameters of the model. EM iterates between the expectation (E) and maximization (M) steps. Given the generative model, to infer cell types we need to calculate the posterior probability $P(z|x, y)$. However, computing these assignments is challenging since changing the assignment of a specific cell type (i.e. changes to Z') also change the neighborhood count Y for other cells. Thus, we perform an iterative procedure as follows: In the first phase Y is treated as a fixed vector for each cell, and is used to calculate the posterior distribution of cell i given the gene expression matrix x_i and current neighborhood count y_i by setting:

$$P(z^i = m|x^i, y^i) \propto \mathcal{N}(x_i; \mu_m, \Sigma_m) \mathcal{M}(y_i; \theta_m) \tag{2.4}$$

In which $\mathcal{N}(\mu_m, \Sigma_m)$ is a multi-variate Gaussian distribution with mean μ_m and covariance matrix Σ_m, and $\mathcal{M}(\theta_m)$ is a Multinomial distribution with θ_m as the

frequency parameter, and we use $\psi = (\mu, \Sigma, \theta)$ to denote all the model parameters. We next use the posterior distribution calculations to update cell type assignments for a subset of the cells. Specifically, we randomly select a set of non-adjacent cells in the adjacency graph G and update their types by the posterior probability. Next, the neighborhood count matrix for all cells, $\mathbf{Y}$, is updated, and is used in the next iteration. We continue with this iterative process until convergence. This method extends the well known Iterative Condition Modes (ICM) update method [**besag1986statistical**] by updating multiple cells in each iteration instead of a single one. However, since we only update non adjacent cells, those updated cells still have the same neighborhood after each round of updates guaranteeing convergence due to the monotonical increase in overall likelihood.

2.0.3 Learning model parameters (M-step)

For M-step, we have:

$$Q(\psi|\psi_{old}) = \sum_{i=1}^{D} \sum_{m} log[P_{\psi}(x^i, y^i, z^i = m)] \cdot P_{\psi_{old}}(z^i = m|x^i, y^i) \tag{2.5}$$

When conditioning on the cell type, the values observed for the gene expression x^i and neighborhood for a cell become independent. Thus, we can write:

$$Q(\psi|\psi_{old}) = \sum_{i=1}^{D} \sum_{m} log[P_{\psi}(x^i|z^i = m) \cdot P_{\psi}(y^i|z^i = m) \cdot P_{\psi}(z^i = m)] \cdot P_{\psi_{old}}(z^i = m|x^i, y^i) \tag{2.6}$$

$$P_{\psi_{old}}(z^i = m|x^i, y^i) = \frac{P_{\psi_{old}}(x^i|z^i = m) \cdot P_{\psi_{old}}(y^i|z^i = m) \cdot P_{\psi_{old}}(z^i = m)}{\sum_{z^i} P_{\psi_{old}}(x^i|z^i) \cdot P_{\psi_{old}}(y^i|z^i) \cdot P_{\psi_{old}}(z^i)} \tag{2.7}$$

So as mentioned above (Section **??**), the posterior distribution is calculated using an alternated ICM algorithm, in which $P(x^i|z = m)$ follows a multivariate Gaussian distribution $\mathcal{N}(\mu_m, \Sigma_m)$, and the neighborhood vector for each cell $P(y^i|z = m)$ follows a Multi-Nominal distribution $\mathcal{M}(\theta_m)$. We set $P(y^i|z = m) = \frac{k!}{y_1^i!...y_M^i!}\theta_{m,1}^{y_1^i} \cdots \theta_{m,M}^{y_M^i}$, where M is the number of cell types, k is the number of neighbourhood cells, $(\theta_{ij}) \in \mathbb{R}_{M \times M}$ is the neighborhood frequency of cell type j given the current cell type i, and is row-wise normalized so that $||\theta_m||_1 = 1$, where θ_m is the m_{th} row of θ. $\pi_m = P_{\theta}(z^i = m)$ is the prior distribution for cell types.

With $P_{\phi_{old}}(z^i = m|x^i, y^i) = \gamma_{im}$, then by maximizing the given Q function, we can obtain the parameters:

$$\mu_m = \frac{\sum_i \gamma_{im} \cdot x^i}{\sum_i \gamma_{im}},\ \Sigma_m = \frac{\sum_i \gamma_{im} \cdot (x^i - \mu_m)(x^i - \mu_m)^T}{\sum_i \gamma_{im}}, \pi_m = \frac{\sum_i \gamma_{im}}{\sum_{i,m} \gamma_{im}},\ \theta_{m,j} = \frac{\sum_i \gamma_{im} \cdot y_j^i}{\sum_{i,j} \gamma_{im} \cdot y_j^i} \tag{2.8}$$

The above likelihood function assumes equal weight for each term in the two types of data (expression and neighborhood). However, there are often much more genes than cell types which can lead to over reliance on the expression data. We use two ways to address this problem, first our model is using the dimensional-reduced gene expression as input, instead of the raw expression profile. But the dimension of this input can still be high, e.g. 20, compared to the typical cell type number to be clustered, for example 7, thus then we include a weight term that balances the contribution of the gene and spatial components, named power factor (see section

??). And also during EM training, the neighborhood count is calculated in term of the assigned probability (a soft update), while usual multinomial distribution is defined in $\mathbb{N}$, so we expand the scope of the multinomial distribution to $\mathbb{R}$ to address this. See Appendix **??** for details.

2.0.4 Dimensionality reduction using denoising autoencoder

A dimension reduced representation of the original gene expression data is used as the input to our model. While the original gene expression data usually does not follow a Gaussian distribution, by using a denoising autoencoder we can transform the data to better fit such model [**vincent2008extracting**]. We use a single layer linear neural network for the auto-encoder though it is possible to adapt the method to use multi-layered networks if the outcome does not fit the required Guassian distribution. We note that when comparing FICT to the expression only GMM method we use the same reduced dimension data as input to both. Thus, the only difference between the GMM model and FICT is the use of the spatial information.

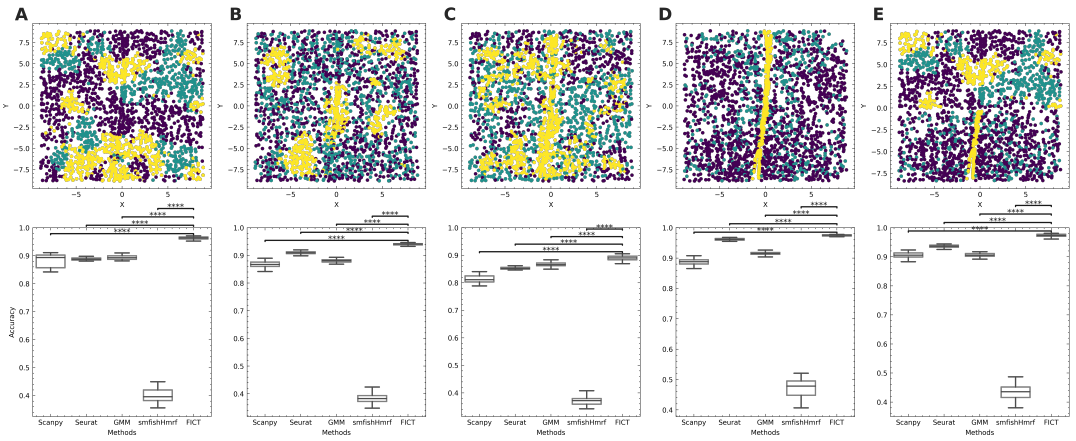

FIGURE 2.2: Evaluation using simulated data. **Top** Simulated ground truth cell type assignments. Cells locations are from the MERFISH dataset (see Figure **??** for selected cells). 4 neighbourhood frequency configurations were simulated: (A) Addictive configuration where cells prefer to aggregate with cells from same type. (B) Exclusive configuration where type 1 and type 2 cells are mixed (green and purple cells) while type 3 cells (yellow cells) cluster together. (C) Consecutive configuration where, type 1 cells surround type 2 cells but not type 3 cells. (D) Cell types assignments from the MERFISH paper (yellow - Ependymal cells, green - Excitatory cells and purple - inhibitory cells). (E) A mixture model where neighborhood distribution for each cell type is a mixture of the distributions in A and D. **Bottom** performance of the 5 methods we tested on simulated datasets. Accuracy for each method is averaged from 50 random expression assignment (Methods). p value is calculated using paired samples t-test. **** P<0.0001

2.0.5 Evaluation using simulated data

While a number of spatial transcriptomics datasets exist, we do not have ground truth information about cell types in these studies. Thus, we first tested our method using simulated data where we can assign both expression and cell type and test

if the method can correctly recover the cell types. As noted in Methods, generating simulated data for such analysis is not trivial since the data needs to satisfy both expression and location constraints. To enable a realistic setting for simulation analysis we used the spatial information from a real dataset (subset of the MERFISH dataset**??**). See Methods for details about the simulation setup. We used the simulated data to test FICT and to compare it to four prior generative and discriminative methods that have been previously used to assign cell types in spatial transcriptomics data. Three of these (GMM [**tian2019clustering**, **xie2016unsupervised**], scanpy [**wolf2018scanpy**, **traag2019louvain**] and Seurat [**stuart2019comprehensive**, **butler2018integrating**, **blondel2008fast**]) only use expression data for clustering while the fourth, smfishHmrf combines gene expression data with cell location and neighborhood information. However, unlike FICT smfishHmrf only considers neighboring cells of the same type (similar to only manually setting the diagonal values in the FICT cell neighborhood matrix and ignoring the off diagonal elements).

In addition to using the cell type assignments from the original paper we also simulated four other cell type assignment settings. Results are presented in Fig.**??**. As can be seen, for all simulation settings FICT is the best performing method followed by Seurat. FICT obtains almost perfect accuracy on all settings, significantly improving upon Seurat and all other methods we compared to ($P < 0.0001$ using paired samples t-test)**??**. Cluster assignment examples for all methods can be found in supplementary Figure **??**

We also compared FICT and the other methods using simulated location and expression data (Methods). Again, FICT significantly outperformed all other methods (Figure **??** and Table **??**). we also tested the robustness of FICT and determined that it was robust to random initialization and to a wide range of values for determining the set of neighbors for each cell (Figures **??** and **??**).

2.0.6 Performance on the MERFISH dataset

We next tested FICT using real single cell spatial transcriptomics data. We first focused on mouse hypothalamus data generated by the multiplexed error-robust fluorescence *in situ* hybridization (MERFISH) method [**moffitt2018molecular**]. The MERFISH data profiles the expression of 258 genes in 480,000 cells from 11 animals (4 females and 7 males). Since there is no ground truth for this data, we used a different approach to compare the different clustering methods. For all gender pairs (i.e. 21 male pairs and 6 female pairs) we performed the following analysis. Let A and B be a pair of animals from the same gender. We first train FICT on A and use the parameters learned for the model trained on A to assign cells in B. We next learn a FICT model for B. We then compare the Adjusted Rand Index (ARI) of the clustering results for the two animals. Higher ARIs mean that the results are more consistent between animals indicating better fit to the underlying biology. Note that this process is not symmetric and so results for training on A and testing on B would be different from those trained on B and tested on A.

Results for this comparison are presented in Figure **??** for both female and male animals. Note that since both Seurat and scanpy are not generative methods the models they learn on one dataset cannot be directly applied to another. Thus, for the real data we compared FICT to smfishHmrf and GMM. Results show that for 32 of the 54 pairs (59%) FICT is more consistent than GMM. The result for the larger dataset of male pairs is (29/42, 69%). The improvement upon smfishHmrf is even larger than that and FICT is more consistent in 52 of the 54 pairs (96.3%). We also tried compare to Seurat and scanpy by learning a classifier using the clustering of

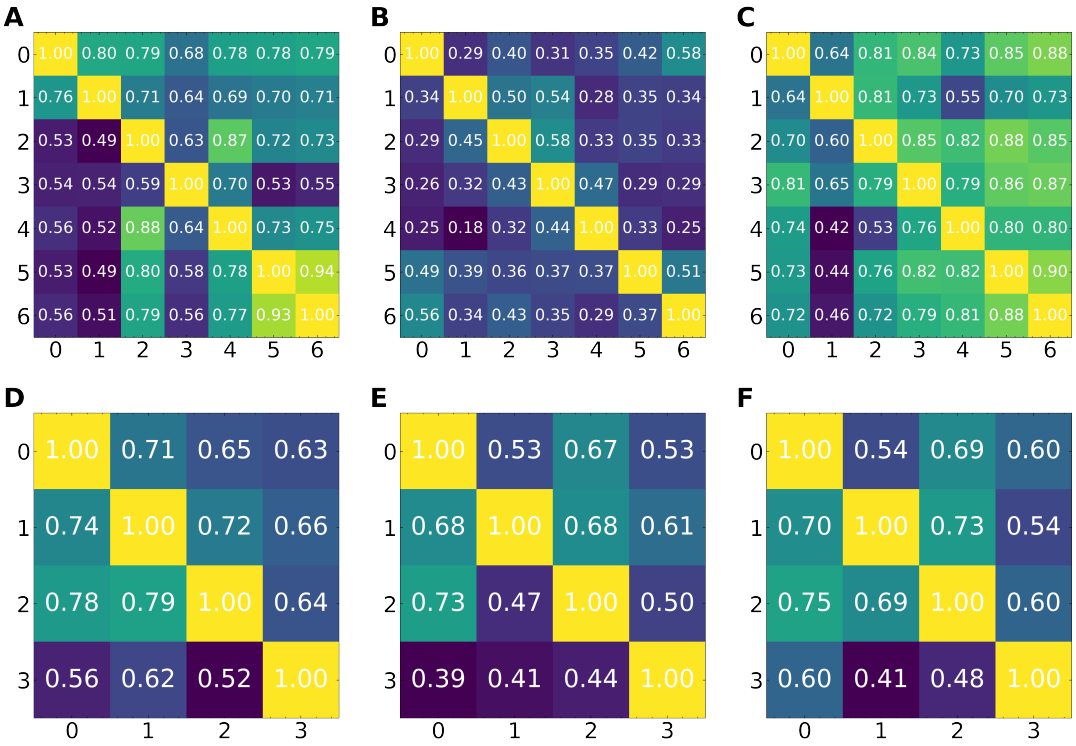

FIGURE 2.3: Mean Adjusted Rand index (ARI) based on cross validation analysis of the MERFISH dataset. Results presented for expression only GMM, smfishHmrf and FICT. Each entry (i,j) in the matrix represents the ARI of the two cluster assignments (one learned on animal A and applied to animal B and the other learned directly on B). (A-C) Results for the 7 Male animals (A) GMM, (B) smfishHmrf and (C) FICT. (D-F) Results for the 4 Females (D) GMM, (E) smfishHmrf and (F) FICT. The x and y axis is the index of the dataset being cross validated on.

one animal and comparing the assignments of the learned classifier to the unsupervised clustering using Seurat and scanpy on another animal. As expected, results indicate that performance of such supervised / unsupervised comparisons is inferior to the results of the generative models as we show in Figure **??**. We note that based on prior studies that indicated that gene expression and cell distribution differ based on gender [**dewing2003sexually**], [**mccarthy2011reframing**], the above analysis was performed by only testing models learned from male animals on male animals and from female animals on female animals. An example of the difference in assignments between expression only GMM clustering and FICT is presented in Figure **??**. As can be seen, the yellow cells (Ependymal cells) are spatially clustered in the center of the hypothalamus tile profiled. However, due to small variations in gene expression, GMM assigns some cells in that cluster as OD Immature cells. In contrast FICT is able to correctly assign these cells as shown in the inset.

Sub-type clustering

An important question in the analysis of brain single cell data is the identification of new sub-types of various neuronal cells [**lake2016neuronal**]. We thus examined the assignments to see if FICT can identify new subtypes of neurons. For this, we focused on the subset of excitatory neurons identified in the MERFISH dataset. FICT

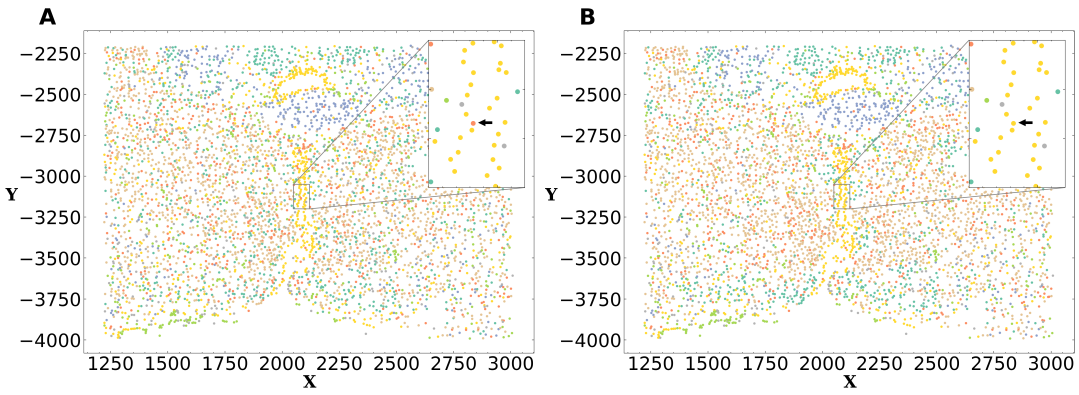

FIGURE 2.4: FICT can correct expression noise. Cell type assignments using expression only GMM (left) and FICT (right). Using the spatial information FICT correctly assigns Ependymal cells along the periventricular hypothalamic nucleus. In contast, the GMM method mistakenly classified the cell as OD Immature Cell.

identified three sub-types of cells that were all determined to be excitatory in the original analysis but displayed different spatial patterns (Figure **??**). To determine if the three sub-clusters are indeed different we performed differential expression (DE) analysis for each of the sub-clusters. While, as expected, their overall expression profiles are similar (leading to their similar assignment by the expression only method) we were able to identify a number of distinct genes for each of these sub-types using MAST [**finak2015mast**]. We next performed GO enrichment analysis [**ashburner2000gene**, **gene2019gene**, **mi2017panther**] on the significant DE genes in each sub-clusters. Results are presented in Figure **??**. As can be seen, some unique functional terms are associated with each of the three sub-clusters. For example, the first sub-cluster (e0) seems to be mainly related to response to chemicals. The second (e1) seems to be related to signaling and regulation of calcium homeostasis while the third (e2) is linked to responses to activity changes and behavior. Thus, while all share similar expression profiles and act as excitatory neurons, each of the sub-clusters may have a further specific function as predicted by the spatial clustering. We performed similar sub clustering analysis using the other methods we compared to. Results are presented in Figures **??** - **??** and indicate that FICT finds both, relevant GO terms such as 'behavior' that are not identified by other methods for this data and more significant enrichment for GO categories related to cell and synapse signaling. We performed similar sub-clustering analysis for inhibitory neurons and obtained similar results both in terms of the more coherent placing of cells from different sub-types and in terms of the unique genes and functions assigned to each of the sub-types identified by FICT (Figure **??** C and D).

2.0.7 Performance on osmFISH and seqFISH

To demonstrate the generality of our method we further tested it on two other datasets from two additional spatial transcriptomics platforms: osmFISH [**codeluppi2018spatial**] and seqFISH [**zhu2018identification**]. The osmFISH dataset profiled 6,470 cells in the mouse somatosensory cortex. The seqFISH dataset profiled 1,597 cells in the mouse visual cortex. Since both datasets only profiled a single animal we performed the cross validation by manually splitting each dataset into 4 smaller regions with approximately the same number of cells. Results for these analyses are presented in

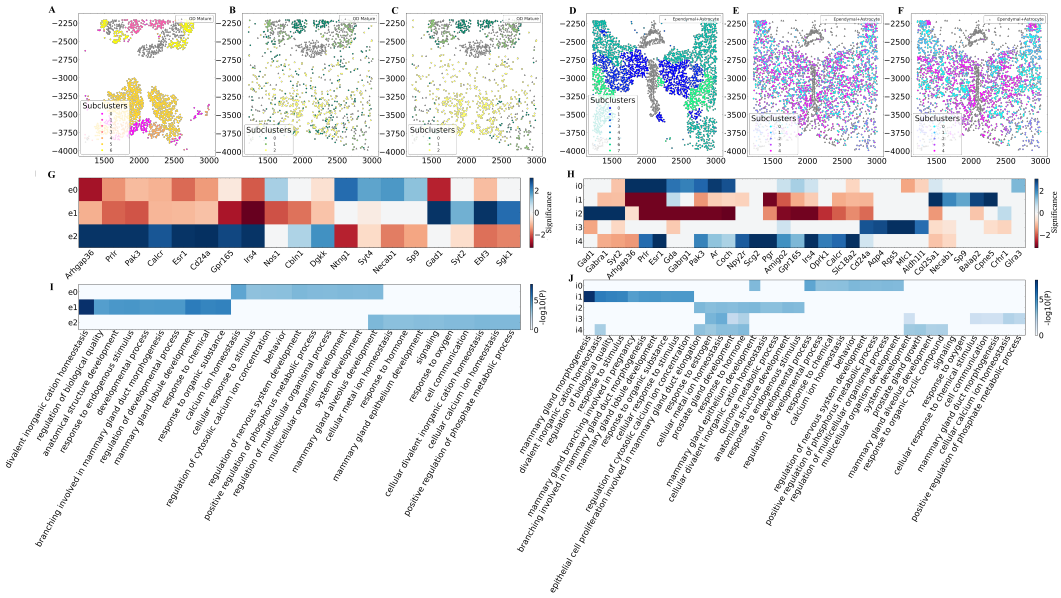

FIGURE 2.5: Cell subtype clustering on MERFISH data from animal 1. We used smfishHmrf (A and D), expression only GMM (B and E) and FICT (C and F) to sub-cluster excitatory neurons cells (A, B and C) and inhibitory neuron cells (D, E and F). As can be seen, for both types of neurons FICT assignments are better spatially conserved creating a central core for sub-cluster 2 surrounded by cells assigned to sub-cluster 0. In contrast, the expression only assignment mixes cells from different sub-types much more. smfishHmrf with Potts model only assigns affinity score between the same cell types making it harder to infer more complex structures of synergistic activity. (E) DE genes for the three FICT sub-clusters from the excitatory neurons and (F) inhibitory neurons. As can be seen, even though the sub-clusters are overall similar in terms of their expression profiles, some genes can be identified for each of the sub-clusters. (G) **GO** enrichment analysis identifies unique functions for each of the sub-clusters on excitatory neurons and (H) inhibitory neurons. Significance of the differential expressed genes is mesured by the log of gene enrichment fold change.

Figure **??**. As can be seen, FICT was able to successfully cluster cells not just based on type but also based on their layer where as clustering using only the expression data, as was performed in the original study, cannot separate layers as well. We also performed cross validation analysis, as we did for the MERFISH data. Given the small number of cells for each dataset we see a drop in performance for all generative model methods. As the figure shows, smfishHmrf was unable to identify more than a single cell type for many of the cross validation runs resulting in errors. As for GMM and FICT while both were able to successfully assign cells in the cross validation runs for the osmFISH and seqFISH datasets, results were not as good as the MERFISH results presented above. Still, even though FICT fits more parameters than the expression only model we observe comparable performance on these smaller datasets suggesting that there is no downside to using the joint expression-spatial assignment **??**.

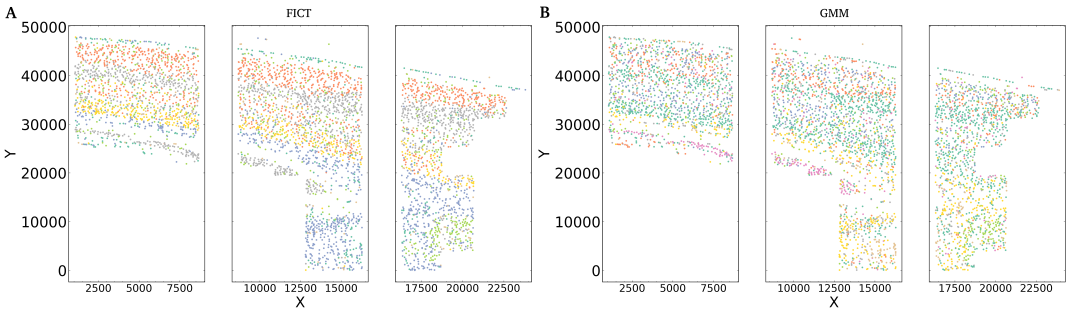

FIGURE 2.6: Cluster assignment scatter plot for osmFISH dataset. (A) clusters generated by FICT and (B) clusters based on using expression data only as was done in the original paper. As can be seen, FICT correctly distinguishes between neurons in different layers of the brain whereas expression only clustering mixes cells from different brain layers.

Appendix A

Frequently Asked Questions

A.1 How do I change the colors of links?

The color of links can be changed to your liking using:

`\hypersetup{urlcolor=red}`, or
`\hypersetup{citecolor=green}`, or
`\hypersetup{allcolor=blue}`.

If you want to completely hide the links, you can use:

`\hypersetup{allcolors=.}`, or even better:
`\hypersetup{hidelinks}`.

If you want to have obvious links in the PDF but not the printed text, use:

`\hypersetup{colorlinks=false}`.