Cell-Type Annotation Priors for scRNA-seq

Oscar Clivio$^{1,2}$, Drausin Wulsin$^1$, Evgeny Kiner$^1$, Noam Solomon$^1$, Luis Voloch$^1$, Jeffrey Regier$^{1,3}$

$^1$ Immunai, $^2$ Department of Statistics, University of Oxford, $^3$ Department of Statistics, University of Michigan

Abstract

Variational autoencoders (VAEs) are popular for interpreting scRNA-seq data. However, the unimodal prior distribution they typically assume is unrealistic: cells in most scRNA-seq datasets cluster by cell type in the posterior distribution. This model misspecification harms performance on downstream tasks. To address this problem, we propose capVI, a VAE that uses a hierarchical prior with one mode per cell type. To ensure that each mode corresponds to exactly one cell type, capVI leverages known cell-type annotations to pre-train a classifier. This classifier is then incorporated into a VAE-style encoder-decoder network which is trained end-to-end with unannotated scRNA-seq data. We validated capVI using two datasets: a public dataset containing scRNA-seq measurements for 11k peripheral blood mononuclear cells (PBMCs) and a second dataset containing scRNA-seq measurements for 41k PBMC cells which were sequenced and annotated by Immunai. For both datasets, capVI substantially outperformed both a popular VAE-based method (scVI) and Scanorama in terms of cLISI, a metric describing the extent to which cells are separated by cell type in latent space. UMAP-based visualizations confirmed that capVI tightly clusters cells of the same type while separating cells of different types.

1 Introduction

Variational autoencoders (VAEs) are increasingly popular for interpreting scRNA-seq data [1,2,3,4,5]. VAEs combine the benefits of probabilistic modeling, such as interpretability and composability, with the benefits of deep neural networks, such as flexibility and computational efficiency. VAEs achieve state-of-the-art performance on tasks like visualization, harmonization, annotation, batch effect removal, and detection of differential expression.

Although VAE-based approaches differ in their precise probabilistic formulations, almost all assume cells can be represented in latent space as samples from a low-dimensional standard normal distribution. This is an unreasonable assumption for essentially all scRNA-seq datasets. By plotting the posterior distribution under these models, we see clusters of cells of the same cell type (as can be seen in Figure 3). These multimodal data are clearly not drawn from the assumed unimodal prior distribution. In fact, the point of many analyses is to discover the clustering structure that these models assume cells do not have.

This model misspecification has several consequences. The posterior distribution may not separate cell types well. Some of the most interesting variation between cell types may be “washed out,” and distinct cell types may even blend together. The log likelihood of held-out data is lower, as are performance metrics for downstream tasks. There is also less basis for correcting batch effects when the latent variables are less clearly resolved.

Hierarchical priors are an appealing solution because they can encode multimodal latent spaces. Unfortunately, models with multiple levels of latent variables are difficult to train, particularly if discrete random variables are present.

Researchers have attempted to perform unsupervised clustering based on mixture priors [6,7] and application of a Gaussian mixture model prior to scRNA-seq [8]. Unfortunately, without supervision, the mixture components found tend to merge rarer cell types while splitting common cell types.

Our approach, called capVI (cell-type annotation prior variational inference), circumvents the challenges of training hierarchical models by using side information, namely, cell-type annotations. Our procedure performs inference under a semi-supervised hierarchical model with a discrete latent variable representing the cell type. Once the cell type is integrated out, the prior over the interpretable latent space has a multimodal structure (Section 2).

To train the model efficiently, we propose a multi-stage procedure that leverages known annotations during pre-training to overcome the challenges of nonconvex optimization. The first step is to train an accurate classifier to annotate cells. Then the weights from the classifier are used to initialize a subset of the weights of an encoder network, followed by unsupervised end-to-end training of both the encoder and decoder (Section 3).

To test our method, we used a dataset of 41k high-quality PBMCs from 24 batches processed by Immunai. For validation, full batches of data are held out, not just cells selected at random from batches also used in training. The proposed method, capVI, when compared to a VAE equipped with a standard normal prior, learned latent representations of the data that better preserve cell-type structure. These performance improvements held up on a smaller public dataset and this method also outperformed Scanorama, a well-regarded non-probabilistic approach (Section 4).
2 The capVI probabilistic model

Let \( N \) be the number of cells in the dataset. Let \( C \) be the number of cell types. For cell \( n = 1, \ldots, N \), let \( c_n \in \{1, \ldots, C\} \) denote its cell type. In capVI,

\[
c_n \sim \text{Categorical}(\phi),
\]

where \( \phi \) is a learned constant vector restricted to the \( C \)-dimensional simplex. Cell type \( c_n \) can be either observed or latent, depending on whether annotations are available.

Let \( D \) denote the dimension of a low-dimensional latent space. In capVI, \( z_n \in \mathbb{R}^D \) is a latent variable representing the biology of cell \( n \). In capVI, the latent representation of each cell depends on its cell type:

\[
z_n \mid c_n \sim \mathcal{N}(\mu_{c_n}, \sigma^2 I).
\]

Here, for \( c \in \{1, \ldots, C\} \), \( \mu_c \in \mathbb{R}^D \) denotes a cell-type-specific centroid, and \( \sigma^2 \) denotes a variance shared across cell types. (We may give each cell type a unique variance in future work.) Note that with this formulation, the marginal distribution over latent variable \( z_n \) (i.e., if \( c_n \) were integrated out) is multimodal.

The model presented thus far differs from scANVI [9] in that the conditional distribution of \( z_n \) given \( c_n \) is unimodal and isotropic. The training procedures (Section 3) and the downstream applications for each method also differ.

The remainder of the capVI model follows scVI [1], a particularly popular VAE model for scRNA-seq. Let \( S \) denote the observed number of scRNA-seq batches, and let \( s_n \in \{1, \ldots, S\} \) denote the observed batch id for cell \( n \). The library size for each cell is a latent variable distributed as

\[
\ell_n \mid s_n \sim \text{LogNormal}(\nu_{s_n}, \tau_{s_n}),
\]

where \( \nu_s \) and \( \tau_s \) are the empirical mean and standard deviation of the log-library size for batch \( s \).

Now, to define the likelihood function, let \( \theta \in \mathbb{R}^{D \times S} \) represent dispersion parameters shared among all cells. Let \( f_w \) and \( f_h \) denote neural networks. For cell \( n \) and gene \( g \), let the normalized gene expression

\[
\rho_{ng} = f_w^g(z_n, s_n)
\]

and the zero-inflation rate

\[
\pi_{ng} = f_h^g(z_n, s_n).
\]

Then, the observed transcript count for gene \( g \) in cell \( n \) is

\[
x_{ng} \mid z_n, \ell_n, s_n \sim \text{ZINB}(\ell_n \rho_{ng}, \theta_{gs_n}, \pi_{ng}).
\]

ZINB here refers to a zero-inflated negative binomial distribution. This distribution was also used as the likelihood function in ZINB-WaVE [10] and later scVI [11]. A negative binomial (without zero inflation) may also be appropriate for some genes and could be substituted for the ZINB distribution in capVI [11].

3 Variational inference

Exact posterior inference in the capVI model is intractable, so we approximate the posterior distribution using variational inference [12]. Our model can be fitted both with and without cell-type annotations. Hence, we derive two variational bounds, one for use with annotated cells and the other for unlabeled cells. For both cases, \( q \) denotes our posterior approximation. We map the data for each cell to the parameters of \( q \) using a neural network—an amortized inference approach to defining a variational distribution [13].

**Observed cell type** When \( c_n \) is observed, the variational distribution has the form

\[
q(z_n, \ell_n \mid x_n, c_n, s_n) = q(z_n \mid x_n, c_n, s_n)q(\ell_n \mid x_n, s_n),
\]

where \( q(z_n \mid x_n, c_n) \) is normal with a diagonal covariance matrix and \( q(\ell_n \mid x_n) \) is log normal. The log likelihood \( p(x_n, c_n) \) is lower bounded by the evidence lower-bound (ELBO):

\[
\mathcal{L}(x_n, c_n) = \mathbb{E}_{q(z_n, \ell_n \mid x_n, c_n, s_n)}[\log p(x_n \mid z_n, \ell_n, s_n)]
- KL[q(z_n \mid x_n, c_n, s_n) \mid \mid p(z_n \mid c_n)p(\ell_n)] + p(c_n).
\]

**Unobserved cell type** When \( c_n \) is unobserved, the variational distribution has the form

\[
q(z_n, \ell_n, c_n \mid x_n, s_n) = q(z_n, \ell_n \mid x_n, c_n, s_n)q(c_n \mid x_n),
\]

where \( q(z_n, \ell_n \mid x_n, c_n, s_n) \) is given above and \( q(c_n \mid x_n) \) is categorical with parameters outputted by a neural network ending with a softmax layer. Let \( H \) denote entropy. Then, the ELBO for unannotated data is

\[
U(x_n) = \mathbb{E}_{q(c_n \mid x_n)}[\mathcal{L}(x_n, c_n)] + H(q(c_n \mid x_n)).
\]

**Batch mixing penalty** To reduce batch effects, we add an additional penalty term to the ELBO which is intended to improve the batch mixing. This penalty term acts as a (soft) constraint on the variational distribution, restricting it to distributions that integrate batches well. Let \( \bar{z}_n \) be the mean of \( q(z_n \mid x_n, c_n) \). Intuitively, we want the mean of the \( \bar{z}_n \)'s for cells in batch \( s \) with cell type \( c \) to be close to the cell-type-specific centroid \( \mu_c \). For every cell \( n \) and cell type \( c \), let \( \bar{q}(c \mid x_n) \) be such that, if cell \( n \) has an annotation \( c_n \), then \( \bar{q}(c \mid x_n) = 1 \) if and only if \( c_n = c \); if cell \( n \) is not annotated, then \( \bar{q}(c \mid x_n) = q(c \mid x_n) \). Assuming a penalty weight \( \lambda \), we define the penalty as

\[
P = \frac{\lambda}{CS} \sum_{c=1}^C \sum_{s=1}^S \left( \left\| \sum_{n: c_n = c} \bar{q}(c \mid x_n)(\bar{z}_n - \mu_c) \right\|_2^2 \right).
\]

**Training procedure** We train capVI in three stages. In stage 1, we train the classifier representing the posterior distribution \( q(c_n \mid x_n) \) using weighted cross-entropy. We perform hyperparameter optimization and pick the best performing classifier with regard to validation set cross-entropy. In stage 2, we initialize capVI’s cell-type posterior distribution \( q(c_n \mid x_n) \) with the architecture and parameters of this classifier and train the rest of the capVI
We assessed capVI, scVI, and Scanorama’s integration function \[ \epsilon \] which is estimated using importance sampling with 5000 samples. Throughout the training procedure, we used Adam \[ 14 \] with \( \epsilon = 0.01 \).

4 Experiments

4.1 Datasets and baseline methods

We assessed capVI, scVI, and Scanorama’s integration function \[ \epsilon \] on two PBMC datasets. The larger dataset, which we call PBMC-41K, was processed by our lab. Cells were pooled and loaded into the 10X Chromium Next GEM Single Cell 5’ Library and Cell Bead Kit v1.1. Libraries were sequenced on the NovaSeq 6000 system using an S3 2x150 kit from Illumina and contain 24 batches from healthy and sick patients, totalling 41,074 cells and 14,243 genes. We subsampled 2000 variable genes and removed 85 specific genes known to be related only to internal cell cycles. Cells were annotated by an immunology expert. To form training, validation, and testing sets, we randomly partitioned batches using a 50/25/25 split. Critically, scVI and PBMC datasets, every annotation appears in each of the training, validation, and testing sets.

The smaller dataset \[ 16 \], which we call PBMC-11K, contains 2 batches, totaling 11,527 cells and 3,346 genes. We followed the same processing used with the PBMC-41K dataset. Training, validation, and testing sets were formed by randomly partitioning cells using a 50/25/25 split.

4.2 Training details

The classifier was first trained using hyperparameter optimization to set the number of hidden layers (1, 3, 5), the number of hidden units (128, 256), the learning rate (0.004, 0.001, 0.00025), and the dropout rate (0.25, 0.4, 0.55). We used early stopping based on held-out standard cross entropy.

During the next stages of training capVI, we tried \( \sigma = 1, 1/8, 1/32, \) and \( \lambda = 0, 0.01, 0.1 \) as hyperparameters; in these stages, we did not use the ground-truth annotations. For both capVI (excluding the classifier network) and scVI, we used 3 hidden layers, 256 hidden units per layer, a dropout rate of 0.25, and a learning rate of \( 10^{-3} \). We used early stopping based on the validation reconstruction error.

Scanorama was run on the entire dataset, including validation and testing sets for scVI and capVI. All three models used a 20-dimensional latent space.

4.3 Classification results

Classification results for capVI’s chosen classifier on the PBMC-41K test set are shown visually in Figure 1. Classification results for PBMC-11, not displayed here, show accuracies greater than 90% for all but one cell type; however, the accuracy for the remaining cell type was still greater than 88%.

![Confusion matrix on PBMC-41K’s testing set.](Image)

4.4 Held-out log likelihood

On PBMC-11K, the best performing capVI model (in terms of validation log likelihood), indexed by \( \sigma = 1/32 \) and \( \lambda = 0.1 \), achieved a test set log likelihood of -655.85, outperforming scVI, which achieved -662.79. All other capVI models also achieved higher test log likelihood than scVI. On PBMC-41K, the best performing capVI model, indexed by \( \sigma = 1 \) and \( \lambda = 0 \), achieved -683.73, compared to -687.26 for scVI.

4.5 Batch and cell-type mixing

We evaluate the quality of data integration using the cLISI and iLISI metrics \[ 17 \]. cLISI describes the effective number of cell types in a cell’s neighborhood. cLISI is therefore lower bounded by 1 and upper bounded by the total number of cell types in the dataset. Because we hope to preserve cell type purity, a lower cLISI score is generally better. iLISI describes the effective number of batches in a cell’s neighborhood. We consider the pairwise iLISI for every cell, and we compute the iLISIs for each pair of the cell’s batch and a neighboring batch, allowing each value to fall between 1 and 2. We then take the median of these pairwise iLISI scores as the iLISI value for each cell.

Figure 2 compares LISI distributions of Scanorama, scVI, and the best capVI model on the PBMC-41K testing set. We can see that the deep generative models outperform Scanorama on both cLISI and iLISI, whereas capVI attains greater label purity relative to scVI, without significantly degrading batch mixing. We observed analogous results on PBMC-11K. In contrast, a fully end-to-end version of capVI, like the one in scVAE, led to a cLISI
distribution similar to that observed using scVI on PBMC-
41K.

Figure 2: Distributions of cLISIs (left) and iLISIs (right)
on the PBMC-41K testing set.

4.6 Latent representation visualization
UMAP plots of the latent spaces from scVI and capVI
on the PBMC-41K testing set are shown in Figures 3
and 4 respectively. capVI better separates CD4 T subtypes
(naive, memory, and Treg) and also pulls non-T cells
(RBC, pDCs, and megakaryocytes) away from the T cells.
CD8 T memory cells are also better separated from NKs.

However, many CD4 T naive cells in the capVI UMAP are
actually in the cluster of CD4 T EM cells. This illustrates
classification results similar to those in Figure 1 which
show that capVI’s classifier predicts a high proportion
of CD4 T naive cells as CD4 T EM.

Finally, we observed that a fully end-to-end version of
capVI, as in scVAE, produced a UMAP similar to scVI.
Together with the LISI results, this shows that pre-training
the classifier may be critical to better identifying cell
types.

5 Discussion
We developed one of the first approaches to address the
limitations of a standard Gaussian prior distribution in
VAEs for scRNA-seq data. By conditioning on cell types,
our prior distribution better models multimodal data. In-
deed, for the two PBMC datasets we examined, capVI
better separated cells into tighter and more distinguish-
able clusters without compromising performance at batch
mixing.

The approach used in capVI has the potential to advance
the field of single-cell data harmonization by, for example,
facilitating the clustering of cell types in an integrated
dataset whose inputs stem from different technologies
or laboratories. This could pave the way to fully au-
tomatic cell-type annotation, which is potentially more
reproducible, less subjective, and less costly than manual
annotation by experts.

Nevertheless, our model is not without limitations. First,
our method may not improve on the alternatives if few
annotated cells are available or if the annotations are of
poor quality. The negative influence of incorrect annota-
tions is only partially mitigated by our training scheme:
the annotations are used to pre-train encoder weights, but
the weights are ultimately learned by unsupervised end-
to-end training. Second, it is less clear how to handle
a priori unknown cell types with our method. For our
PBMC datasets, this was not a significant concern, as few
cells were thought to be of an unknown type; however,
for other datasets, this may be a concern. In future work,
we may consider the setting in which some of the cell
types that appear in the testing set are excluded from the
training set.
References


