Generative Models for scRNA

Jackson Loper

March 23, 2018

Status Succesfully trained the new "iterative model," evaluated it to some extent

Todo Compare clusters against Allen, consider linear model, do more layers

1 Overview

The Allen Institute has developed a method for clustering, one that has apparently allowed them to distinguish 116 unique cell types using single cell RNA data. The method follows the following iterative procedure:

Algorithm 1: itclust

Data: RNA expression for a collection of cells

```
Run PCA on the data (e.g. take first 16 components);
Run T-SNE on the resulting low-d representation;
Circle things that look like clusters;
if there is more than one cluster then
    for each cluster c do
        run itclust on the cells inside c;
    end
end
```

This method is nice because the important low-dimensional representations *within* a cluster may be quite different from the ones that help you distinguish *between* two subclusters within that cluster. This is what allows them to achieve the level of granularity that they have.

Unfortunately, it is a bit difficult to grasp the uncertainty involved in the process. There is certainly noise in the data, and different practitioners of this basic procedure may have different understandings of what this noise is. The result is that when Bosilijka Tasic (one of the lead experimentalists at Allen) gives two data scientists the same data, they each come back with different answers. And they have no way to compare or evaluate these two different answers. One essential question is "what level of granularity does the signal-to-noise ratio support?" The algorithm above, as stated, runs iteratively until subclusters cannot be distinguished via T-SNE. But this is obviously a very subjective matter. What to do?

We propose to help clarify some of these issues by building a fully generative model of gene expression, one that rigorously measures some kinds of uncertainty in every step of the process. To be most useful to the Allen Institute, however, we also want this generative model to fit into the scientific paradigm under which they are operating. In particular, the treatment of clusters should be hierarchical in nature, and it is desirable that the model

have latent representations that roughly correspond to the low-dimensional representations found in the Allen algorithm.

2 Model

Towards this end, we propose the following generative model for gene expression of a single cell. We will denote the expression of gene g in cell c by X_g^c. Let H denote a tree of cell types.

1. $T^c \sim \text{Categorical}(\pi)$, one of the leaves of H. Let $t_0, t_1(T^c), t_2(T^c), \cdots, t_\ell^c(T^c)$, indicate the nodes in the unique path in H which leads from the root to the leaf T. Here t_0 is the root of the tree, ℓ^c is the depth of the leaf T^c, and $t_\ell^c(T^c) = T^c$. To consolidate notation, we take $T_i^c \triangleq t_i(T^c)$.
2. For each $i < \ell$ let $Z^c_{T_i^c} \sim \mathcal{N}(\mu_{T^c_{i+1}}, \Sigma_{T^c_{i+1}})$, and take $Z^c_{T_\ell^c} \sim \mathcal{N}(0, I)$
3. For each gene g, let $\lambda_g^c \in \mathbb{R}^p$ denote a set of parameters. These will be a deterministic function of the Zs:
$$\lambda_g^c = f_g\left(\sum_{i=0}^{\ell} h_{T_i^c}(Z^c_{T_i^c})\right)$$
4. Conditioned on Z^c we will have $X_g^c \sim p(x_g; \lambda_g^c)$.

Let us unpack some of this:

- The latent representation Z_i^c corresponds to features of that cell which are specific to all the cells which are descendents of T_i^c.
- The the cell's subcluster T_{i+1}^c effects the distribution of its latent features at the ith level, Z_i^c. That is, the latent distribution on the ith latent representation is governed by what subcluster it is part of within T_i^c.
- The distribution of the final latent representation, Z_T^c, is simply a standard normal.
- The functions $h_{T_i^c}$ indicates how these cell-type-specific features alter the distribution on the gene expressions for cells in that branch of the tree.

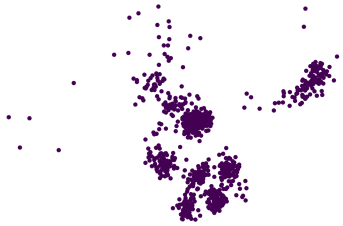
For inference, we use amortized variational inference. In particular, we take the variational family that T and all the Z_{T_i} are independent, T is some categorical, and Zs are isotropic gaussians. We build a neural network to estimate the variational parameters from gene expressions. Note that this network must estimate the parameters for a gaussian at *each node* in the tree H.

3 Empirical results

At the current moment, we have only actually implemented this model in the case the H is a tree with depth 2 (i.e. one root node and some children). We take the data model to be a zero-inflated negative binomial. It looks fairly promising. For a simple qualitative assessment, consider the following.

Scatterplot of the aggregate posterior distribution of Z_{T_0}, colored by posterior mean of the cluster assigment. That is, each dot is a cell, its position is given by a posterior sample of Z_{T_0}, and its color has something to do with what cluster we expect it to be in.

Notice that there are only so many clusters that are really clearly distinct in this latent space. However, let's focus on that cluster on the right-hand side. What happens if we look at the Z_{T_1} variables for those cells? We get substantial subclustering apparent in the Z_{T_1} space. In the following picture, each dot is a cell from the cluster on the right-hand-side of the previous figure, and the position is given by a posterior sample of Z_{T_1}.



3.1 Held out log likelihood

To quantitatively analyse our method, we used a held out log likelihood:

- Train the method on 75% of the cells
- For each of the remaining 25% of the cells, select 75% of the genes ("observed"):
  - Compute the posterior mode of Z given the expression of those genes
  - Compute the log probability of the observed genes given the posterior mode
  - Compute the log probability of the unobserved genes given the posterior mode
- Average values over those 25% of the cells

Here are the results for this "Posterior mode predictive (PMP)" approach:

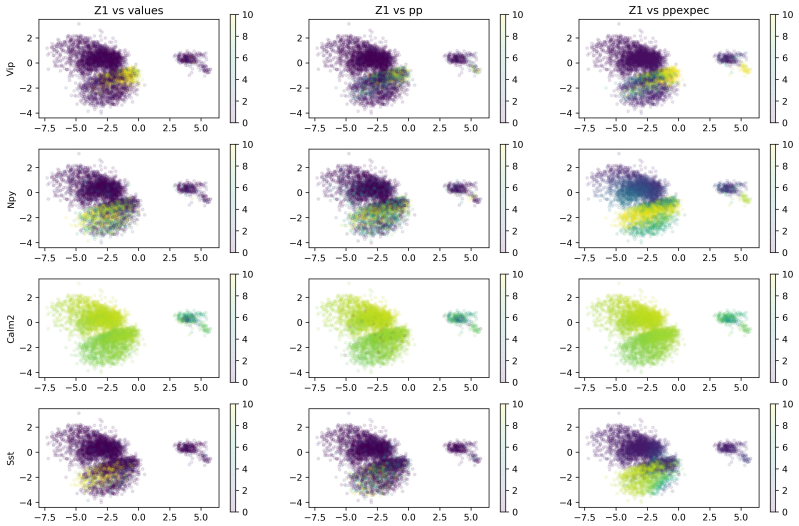
Method	PMP, observed	PMP, unobserved
Factor analysis $k = 16$	-4.2	-4.2
Factor analysis $k = 64$	-4.1	-4.3
HNBPF $k = 2$	-5.7	-5.7
HNBPF $k = 16$	-3.8	-3.9
HNBPF $k = 64$	-3.8	-5.7
ZINB-Vae $k = 2$	-3.6	-3.6
Smooth ZINB-Vae $k = 2$	-3.6	-3.6
Smooth Hierarchical ZINB $k = 2 + 2$	-3.5	-3.5

Here HNBPF is a negative-binomial-poisson-factorization approach, ZINB-Vae is based on a Michael Jordan paper, k is the number of latent dimensions. The results are certainly not mind-blowing, but its nice to see that we're at least not doing worse.

To do posterior inference on held out likelihood, our approach is to first set all of the "unobserved" entries to the average value for the corresponding genes, and apply our inference network unchanged to get estimates for the variational parameters. We then refine these estimates with gradient descent. We note that if we use completely random guesses to initialize the gradient descent, we often converge to inferior local minima; this shows the benefit of the amortized inference.

3.2 Posterior predictive samples

One limitation of the method as we have implemented it is that the posterior predictive distribution seems to be substantially underfitting. To see what we mean, consider the following figure:



Here we four rows and three columns of plots. Each row corresponds to a different gene. Each dot in each plot corresponds to a cell. The position of the dot indicates the latent representation of the cell, as inferred by posterior inference. The color indicates the amount of expression for the particular gene for that cell, measured three ways:

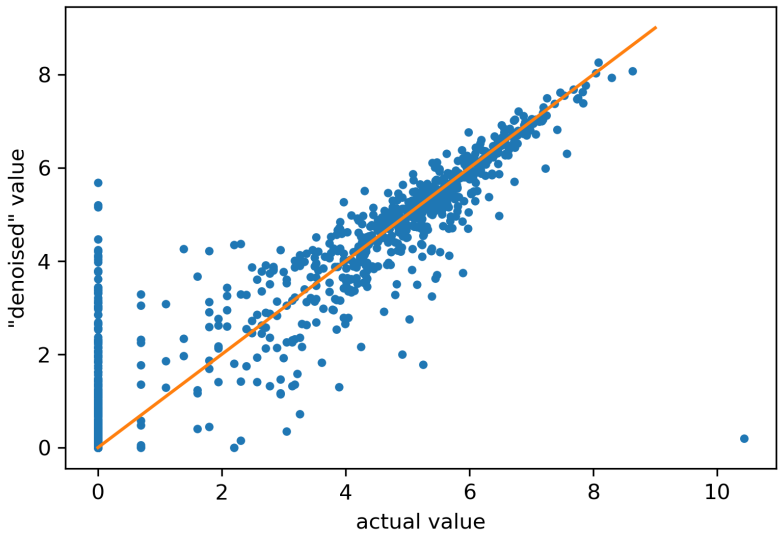
1. The first column shows the true expression of the appropriate gene for each cell (i.e the color of each dot is the gene expression for that cell)
2. The second column shows a posterior predictive sample of the gene expression for each cell
3. The third column shows the mean of such samples for each cell

We see that the true expression doesn't match the posterior predictive expression for the Vip and Sst genes as well as we might like. Something better is needed. Possibly more layers. Possibly more training. Worth thinking about.

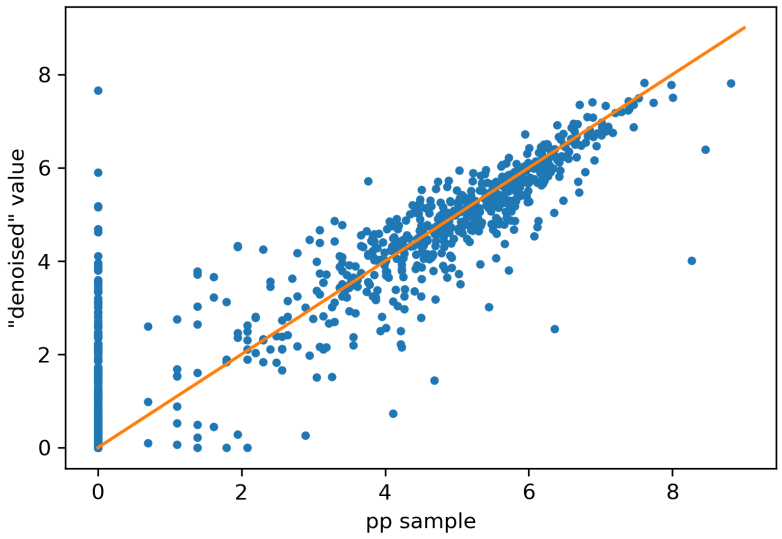
3.3 Posterior predictive means

The posterior predictive mean can be interpreted as a "denoised" version of the original. In this section we examine the Allen data from that point view.

For example, here we have selected 1000 random entries from the gene by cell matrix. For each entry, we plot a point in which the horizontal location indicates the true value of that entry, and the vertical position indicates the posterior predictive expectation of that entry.

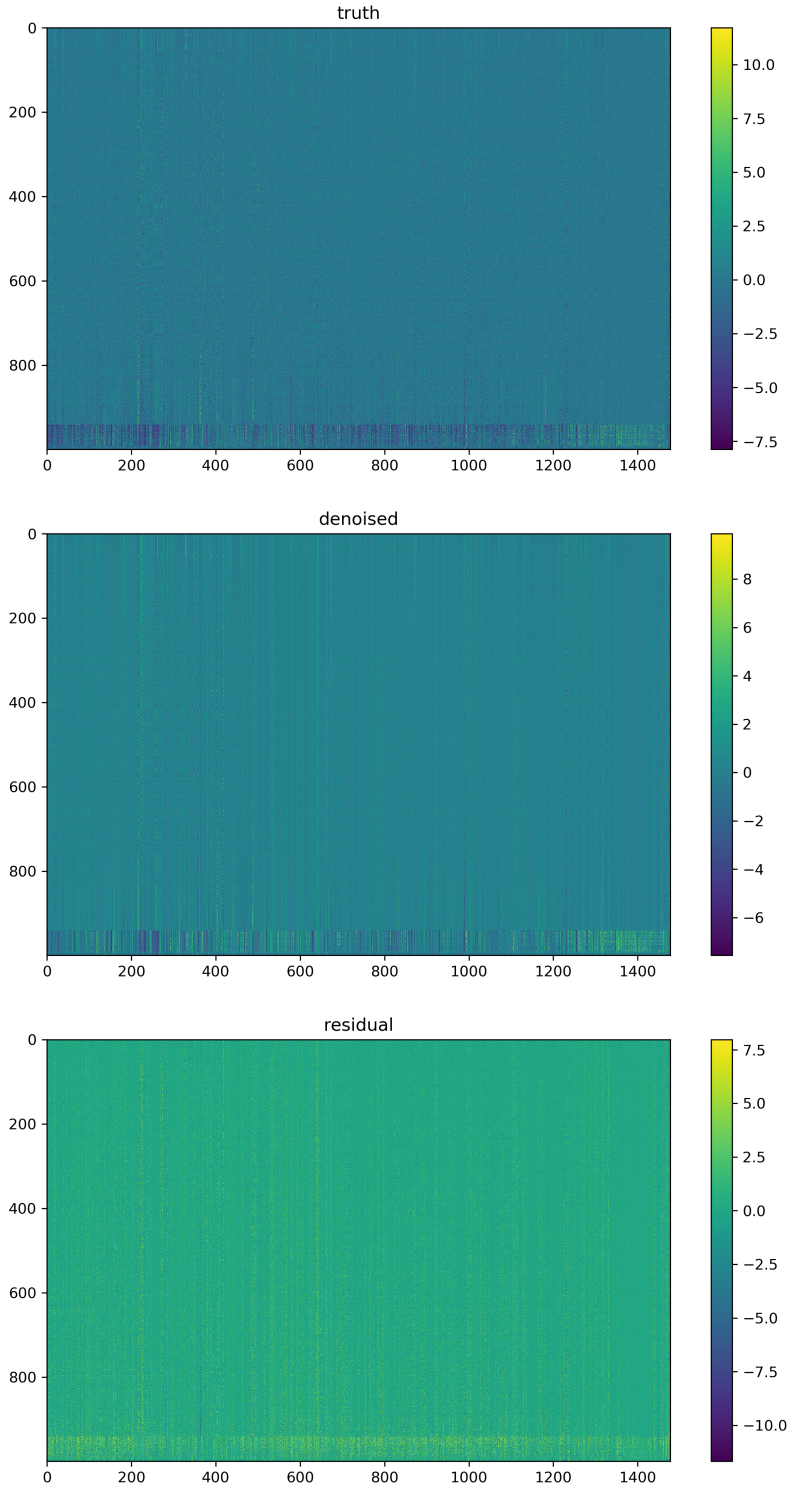


We can compare this with a similar plot using posterior predictive samples instead of the true values. This then forms a different kind of posterior predictive check.



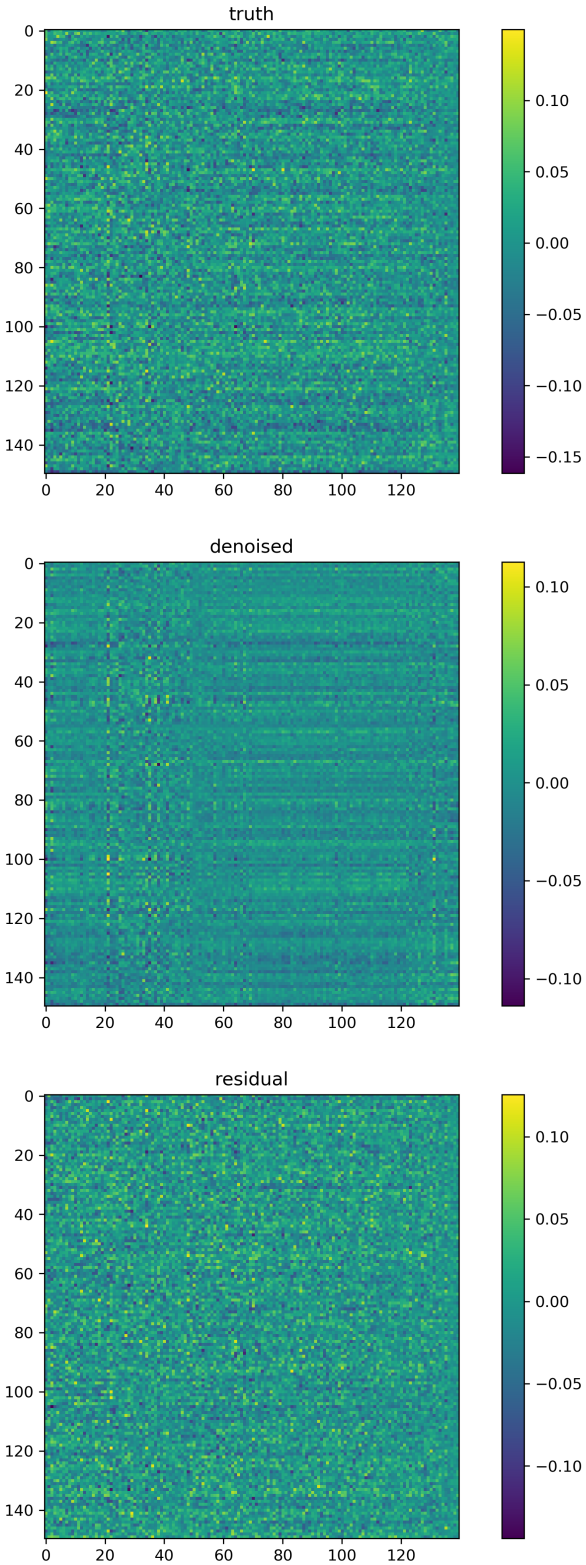
The fact that these two plots look the same is encouraging.

We can also use heatmaps to try to give more structure to our understanding of this "denoising" process. Here we have three plots: one giving the original gene expression for the first 1000 cells, one giving the posterior expected gene expression for the same cells, and one giving the residual. In all cases we have transformed the data by taking $\log(1+x)$ and subtracting gene-level means. Each column corresponds to a gene, and each row corresponds to a cell. We have sorted the cells by one of the latent components, and sorted the genes by using a property of the gene decoder.



Note that there is a group of cells which are much more difficult to understand, which are found at the bottom of the plot. These are mostly astrocytes.

Since we have 20000 cells, it might be nice to make a similar plot for all the cells. However, such a plot would be very tall and perhaps difficult to look at. We can instead take the full 20000 x 1400 plot and average neighboring rows and columns together. This gives a plot like this:



If there was a lot of structure in the residual, we would feel that perhaps there was something important and obvious we were missing. Since there isn't, we are reasonably encouraged.