Variational Inference compared to Markov Chain Monte Carlo for modelling gene expression

Theodoros Bitsakis, Tushar Goel, Siddhartha Jain, Wouter Saelens
École Polytechnique Fédérale de Lausanne, Switzerland

I. Introduction

All multicellular organisms are composed of cells. These cells can further be described using their *gene expressions*. Depending on the function of the cell, gene activity can differ significantly. Genes that are more active have more "copies" present in the cell, called mRNAs.

Computational biologists often study how the gene expression changes depending on the type of cell. The best approach for this is to use Bayesian modelling, but Markov Chain Monte Carlo (MCMC) can be prohibitively slow. In this project we study what the quality of the approximation provided by Variational Inference (VI) compared to the exact posterior for MCMC.

II. Design

A. Model Creation

Before writing any code, the first step was to the read the relevant biology. It started with understanding the central dogma of molecular biology, stated popularly as "DNA makes RNA, and RNA makes protein." In the context of the project, we then read about single-cell transcriptomics - a technology which examines the gene expression level of individual cells. The dataset which we work with in this study represents the number of mRNAs inside a cell for each gene, which reflects how "active" those genes are for that given cell.

The objective of the project is to conclude the best way to model the effect of *perturbations* to the cells. For example, a perturbation can correspond to adding some chemicals or adding an extra gene to the cell. We implement parameterised models for the effect of the perturbation, and trained them using either MCMC or VI. Let D be the deviation in the gene expression, observed on application of perturbation p (normalised to be in $[0, 1]$). We used three models:

- *Nothing*: As the name suggests, this model assumes perturbations do not affect the count of the corresponding gene in the cell. This model then fits the best constant to the data.
- *Linear*: This model assumes there is a linear relation between perturbation and mRNA count. The slope β is modelled as a random variable, with prior $P(\beta)$ a normal distribution with $\mu = 0$ and σ empirically estimated.

$$D = 1 + \beta p$$

- *Switch*: This model would ideally assume there is a threshold after which the mRNA count shoots up, and the value does not change at all on either side of this threshold. As the non-differentiability around the switch point can cause unstable training for both MCMC and VI, we use a relaxed approximation based on a sigmoid function. We use γ to denote the switch threshold, and δ is a skew value to achieve a steep slope.

$$D = 1 + \frac{\beta}{1 + exp(-\delta(p - \gamma))}$$

We set β to have the same prior as the Linear model, δ to be a hyperparameter with value 50, and we give γ the prior of the Uniform distribution on $[0, 1]$.

To implement the models, we used a combination of `jax` and `numpyro`. These libraries are the state-of-the-art in terms of fast, accurate probabilistic programming. `jax` also has the advantage of implementing Autograd, which meant fast gradients without needing to implement them for each individual model.

B. Agenda

After we created the models, we had to formalise exactly what we wanted to test. We essentially want to know if MCMC and VI return similar posteriors,

1) What does the "exact" posterior look like?
2) Does VI produce a posterior similar to that of MCMC?

Further, the following questions have to be answered for each of the aforementioned questions.

1) Is this variable dependent?
2) Is this model dependent?
3) Is this gene dependent?

III. Methodology

A. Data Generation

For conducting this study, we generated a synthetic dataset of 400 cells with 30 genes per cell. Real world data would be too large and the time taken by MCMC would inhibit a careful study of the posterior distributions.

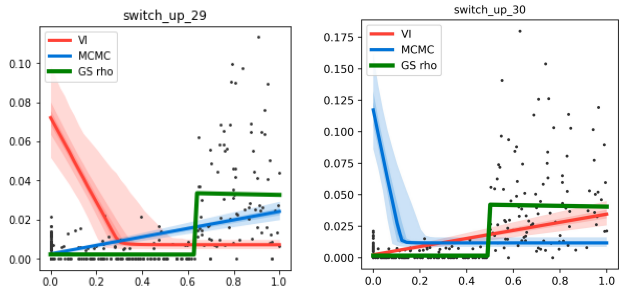
The 30 genes we generate have the following composition:

- 10 genes follow the Nothing model, in which genes which stay the same in all cells.
- 10 genes follow the Linear model (out of which 5 have positive slope and 5 have negative slope).
- 10 genes follow the Switch model (out of which 5 switch up and 5 switch down).

We use the same dataset throughout the report for the posterior distributions, by saving a randomly generated sample. This ensured consistency in our inferences. For the parameter estimates since they are not affected too much by taking different samples, we use multiple samples to get a more holistic picture.

B. Training

We observed multiple issues while training MCMC and VI, which significantly affect the quality of the posterior distributions.



(a) Bad initialisation for VI (b) Bad initialisation for MCMC

Figure 1: VI and MCMC converge to local minima on bad initialisations

Initialization of the VI parameters was crucial in assessing the performance of the model. We noticed that bad initialization of the VI parameters could lead the parameters to local optima, as seen in Fig 1a. We were initially using the `init_to_feasible` method, which initialises to an arbitrary feasible point, ignoring the distribution parameters. We observed that this resulted in VI getting stuck in a local minima very often, as illustrated in Fig 1a. This outcome was avoided by instead using the `init_to_median` method which samples 15 points from the distribution of the prior and initialises on their median. A natural question was whether initialising to the uniform sample from the prior also leads to bad behaviour. In Fig 2 we show the parameter convergence of the β parameter in the Switch model, comparing the two initialisations. As it can be seen, the choice between these two methods have a very little impact for the switch type genes. A similar trend was also observed for the other parameters.

MCMC also converged to local minima, for eg for the `switch_up_30` gene as shown visually in Fig 1b. One possible way to overcome this may be using more warmup steps.

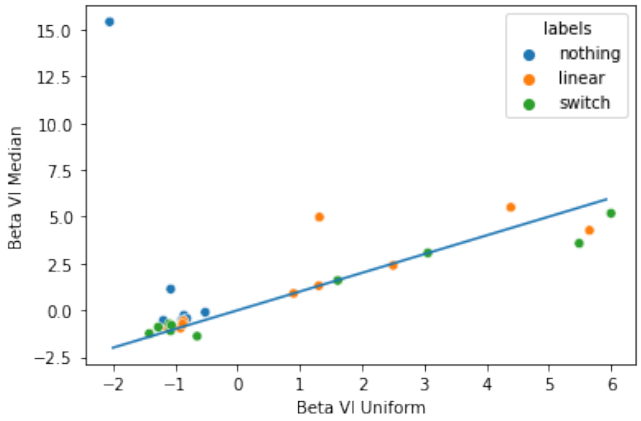


Figure 2

C. Visualisation

We consider three visualisations of the results of the training to make inferences. One option is to directly visualise the posterior distributions returned by both algorithms.

To go beyond simply visualising the posterior distributions returned by MCMC and VI, we plot some statistics of the parameters returned by both MCMC (on the X axis) and VI (on the Y axis) and draw the identity line to see if VI is clearly underestimating or overestimating any parameter.

Finally, we can also visualise the marginals of each parameter in the distributions.

IV. Results

A. Posterior distributions

We see cases where VI underestimates the variance of the posterior distribution significantly. An exmaple of this is shown in Fig 3.

B. Parameter statistics

Plotting the mean of the parameters learned by MCMC and VI (see Fig 4), we make the following observations:

- For the Switch model it underestimates in one gene, but is quite comparable otherwise. It can also be seen that for the switch genes, the parameter estimates are the same. We see a similar trend for the switch (γ) parameters in the Switch model.
- We can see from Fig 6 and Fig 9 that while MCMC and VI are able to agree on the mean values of the parameters corresponding to their respective genes, they do not have the same distribution. Moreover, VI always underestimates the variance of these parameters.

Given such a performance, we also try to see how the models of these parameter estimates look compared to the actual data. For this, we select a few interesting genes to

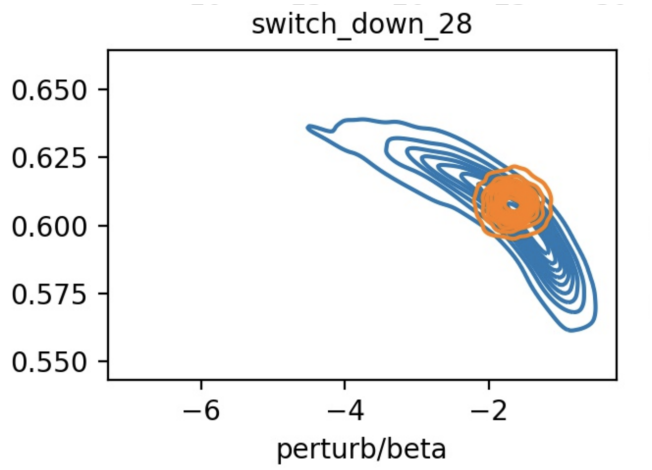


Figure 3: The "exact" posteriors returned by MCMC and VI for a particular gene using the Switch model

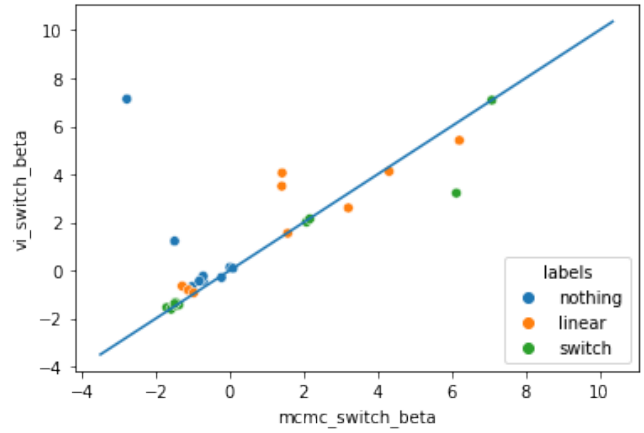


Figure 4: Mean of the β parameter for the Switch Model returned by MCMC and VI

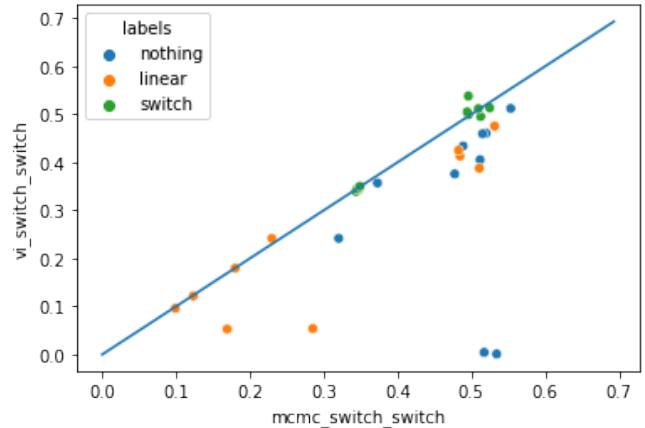


Figure 5: Mean of the Switch parameter for the Switch Model returned by MCMC and VI

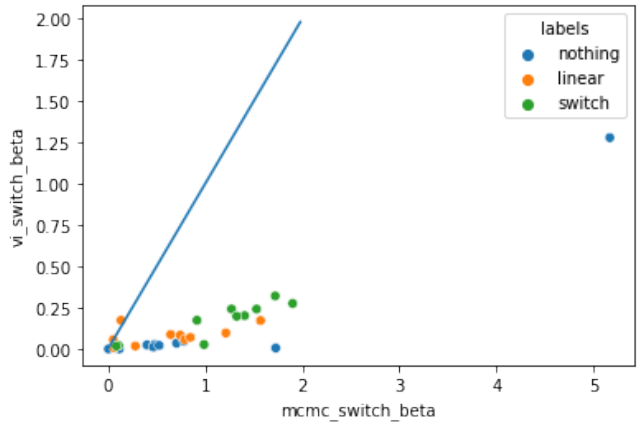


Figure 6: Variance of the β parameter returned by MCMC and VI

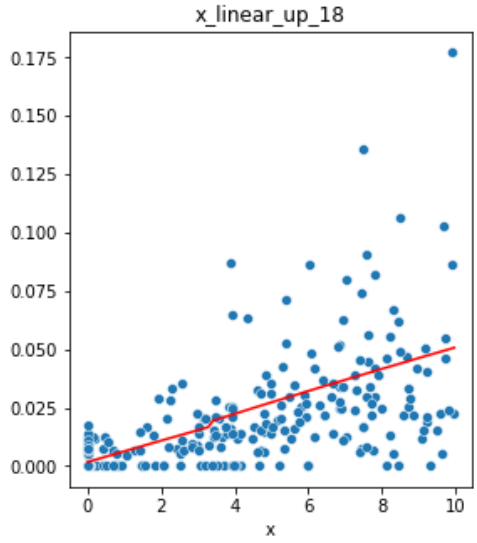


Figure 7: Linear Gene Model

illustrate the performance of our estimates. It is seen that the linear models perform well for the linear genes as can be seen in Fig 7. Please note that these models perform poorly for the other gene types as it can be seen in Fig 4 and Fig 5

The same however can not be said for the Switch Models. We observe that both MCMC and VI are able to make fairly accurate predictions for the switch_up genes. Unfortunately, these models are not able to do the same for switch_down genes. This might be attributed to the choice of our priors on the parameters. MCMC corresponds to green ● and VI corresponds to red ●.

C. Marginal distributions

We see that VI (orange ●) estimates the parameters of the models very well, as the results are very close to those of

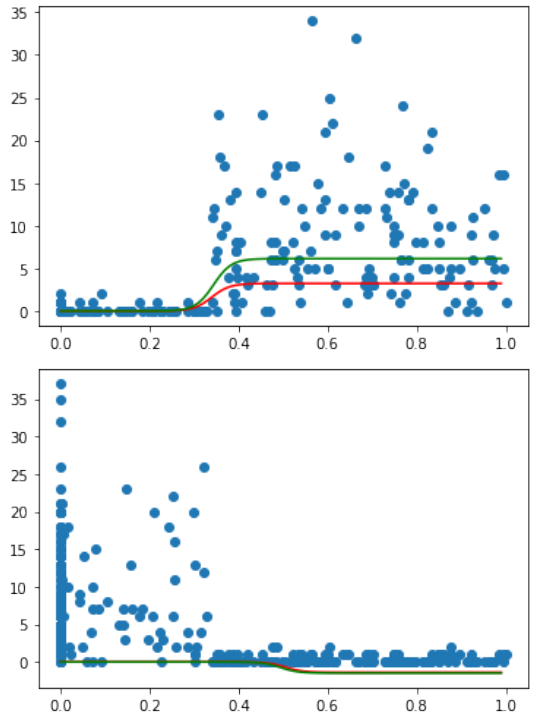


Figure 8: Switch Gene Model

MCMC (blue •). For eg, in Fig 9 we see a comparison of the β distributions for the Linear model.

D. Pearson correlation test

We ran the Pearson correlation test on the distributions returned by MCMC and VI. There were 43 significant results, which can be found in the the `correlations/` directory. We include the most significant results here, for the Switch model.

Pearson coefficients for the Switch model			
Parameter 1	Parameter 2	Coefficient	Gene
freq	β	-0.7404	x_linear_up_11
freq	β	-0.7337	x_linear_up_16
freq	β	-0.5076	switch_up_26
freq	β	-0.6121	switch_up_29
freq	γ	0.6073	nothing_1
freq	γ	0.5156	x_linear_up_11
freq	γ	0.5557	x_linear_up_16

We see that there are dependencies in our parameters which were not part of our initial prior.

V. Conclusion

We observe that most models work well for their respective gene types. We also observe that while MCMC and VI make the same parameter estimates for most models, their posterior distributions are very different from each other. There is some dependency of the model performances on

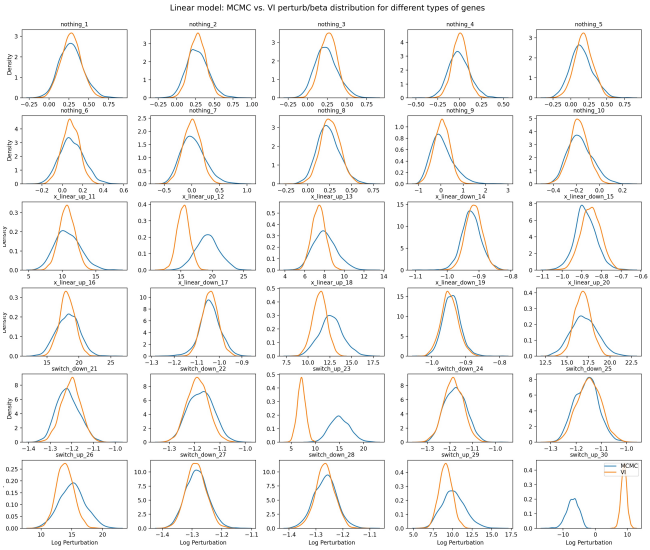


Figure 9

the gene types as well. It is also important to notice the non-convexity of the Variational Inference loss functions as they very often lead to local minima. Our basic assumptions about the model parameters being independent of each other are also tested when we notice significant correlations for each model type. From our preliminary analysis, we would conclude that it is safe to use VI approximations for most gene types, but it should be done with precaution as noted above.

A. Future work

A comprehensive scalability analysis is a possibility for further study. In particular, we think that VI may be the best choice for an initial global analysis of a dataset, followed with detailed analysis using MCMC on a subset of genes for exact statistical inference. A natural extension of this study would be to answer the questions we consider for other models as well, for eg an Exponential or a Spline model.

Further, we assume that the gene expressions are independent of each other given the pertubations. This might not be true due to the complex dynamics of gene expressions. Another area of study which this project could extend to is modeling the gene expressions with dependencies in between them.

References

[1] J. Bradbury, R. Frostig, P. Hawkins, M. J. Johnson, C. Leary, D. Maclaurin, G. Necula, A. Paszke, J. VanderPlas, S. Wanderman-Milne, and Q. Zhang, "JAX: composable transformations of Python+NumPy programs," 2018. [Online]. Available: http://github.com/google/jax

[2] D. Phan, N. Pradhan, and M. Jankowiak, "Composable effects for flexible and accelerated probabilistic programming in numpyro," *arXiv preprint arXiv:1912.11554*, 2019.

[3] T. Salimans, D. P. Kingma, and M. Welling, "Markov chain monte carlo and variational inference: Bridging the gap," 2015.

[4] C. K. Yau, C., "Bayesian statistical learning for big data biology," 2019.