
Measuring Cluster Stability for Bayesian non-Parametrics Using the Linear Bootstrap

Ryan Giordano*
rgiordano@berkeley.edu

Runjing Liu*
runjing_liu@berkeley.edu

Nelle Varoquaux*
nelle@berkeley.edu

Michael Jordan
jordan@cs.berkeley.edu

Tamara Broderick
tbroderick@csail.mit.edu

§ 1. Introduction. Clustering is the canonical unsupervised learning problem, in which we aim to find an assignment of data points to groups, or clusters, that represent meaningful latent structure in a data set. Bayesian non-parametric (BNP) models form a particularly popular set of Bayesian models for clustering due to their flexibility and coherent assessment of uncertainty. As with any Bayesian model of moderate complexity, typically the Bayesian posterior cannot be computed exactly for BNP clustering problems, and an approximation must be employed. Mean-field variational Bayes (MFVB) forms a posterior approximation by solving an optimization problem and is widely used due to its speed [Blei and Jordan, 2006]. An exact BNP posterior might, at least in theory, vary dramatically when presented with different data. Certainly we expect small, rare clusters—which are ubiquitous in BNP—to vary substantially based on the observed data. When reporting the summaries of the clustering for the purposes of scientific inquiry, it behooves us to understand how stable, or alternatively how sensitive, this report is relative to the data [Yu et al., 2013].

If one were to use the bootstrap to assess stability in this analysis pipeline, it would require a new run of MFVB for each simulated data set. This time cost is often prohibitively expensive, especially for exploratory data analyses. We instead propose to provide a fast, automatic approximation to a full bootstrap analysis based on the infinitesimal jackknife [Jaekel, 1972, Efron, 1982], which can be seen as a linear approximation to the global stability measure provided by the full bootstrap. This locality can buy drastic time savings, with the infinitesimal jackknife sometimes running orders of magnitude faster than the bootstrap. We here demonstrate how to apply this idea to a data analysis pipeline consisting of an MFVB approximation to a BNP clustering posterior. We show that the necessary calculations can be done nearly automatically, without tedious derivations by a practitioner, using modern automatic differentiation software [Maclaurin et al., 2015]. This automation suggests a generality to our methods beyond BNP clustering.

In the remainder, we describe the BNP model and MFVB inference in more detail in Section 2. We review summaries for assessing the output of our clustering, across which we can in turn assess stability, in Section 3. We describe our new stability assessment procedure in Section 4. And we demonstrate our ability to quickly and accurately quantify stability in Section 5 on an application to clustering time-course gene expression data [Shoemaker et al., 2015, Luan and Li, 2003].

§ 2. Data, Model, and Inference. Clustering procedures typically estimate which data points are clustered together, a quantity of primary importance in many analyses. It can be used to reduce the dimensionality, or to facilitate the interpretation of complex data sets. For example, genomics experiment often assess cell activity genome-wide, but many genes behave the same way. Clustering them thus allows dimensionality reduction that can facilitate interpretation. Finding robust and stable clusters is thus crucial for appropriate downstream analysis.

*These authors contributed equally.

Because the differences and evolution over time of gene expression yields important insight on gene regulation of the cell-cycle, or on how cells react to toxins, drugs or viruses, we focus on the specific task of clustering time course gene-expression data. We use a publicly available data set of mice gene expression [Shoemaker et al., 2015], composed 14 time points after mice are infected with the influenza virus. See Appendix A for more details.

The observed data consists of expression levels y_{gt} for genes $g = 1, \dots, n_g$ and time points $t = 1, \dots, n_t$ (see Fig. (3) for a single-gene time course data). As described by Luan and Li [2003], we model the time series as a gene-specific constant additive offset plus a B-spline basis of degree 3 and 7 degrees of freedom. We denote the basis matrix by X (see Fig. (3) in Appendix A).

Let b_g denote the additive offset for gene g , and y_g the vector of observations $(y_{g1}, \dots, y_{gt})^\top$. Denote the variance of the errors as σ^2 and let I_T be the $n_t \times n_t$ identity matrix. We model each gene's B-spline coefficients, β_g , using a stick-breaking representation of a Bayesian non-parametric (BNP) Dirichlet process mixture model [Ferguson, 1973, Sethuraman, 1994]. Excluding priors, the generative model is then:

$$\begin{aligned} \nu_k | \alpha &\stackrel{iid}{\sim} \text{Beta}(1, \alpha) & \pi_k(\nu) &:= \nu_k \prod_{j=1}^{k-1} (1 - \nu_j) & \beta_k &\stackrel{iid}{\sim} G_0 \\ z_g | \nu &\stackrel{iid}{\sim} \text{Mult}(\pi(\nu)) & \beta_g | z_g &= \sum_{k=1}^{\infty} \beta_k z_{gk} & y_g | X, \beta_g, b_g, \sigma^2 &\stackrel{iid}{\sim} \mathcal{N}(X\beta_g + b_g, I_T\sigma^2) \end{aligned} \quad (1)$$

See Appendix B for details of the priors.

For brevity, use the single vector θ to represent all the unknown parameters $\nu, \beta_k, z_{gk}, \sigma^2$, and b_g , for all k and $g = 1, \dots, n_g$. We are interested in the posterior $p(\theta|y)$, which is intractable. To approximate $p(\theta|y)$, we form a variational approximation to $p(\theta|y)$, denoted $q^*(\theta)$ and parameterized by a real-valued parameter η , using a truncated representation of the BNP prior with $K = 30$ components, which was large enough that more than half of the clusters were essentially unoccupied [Blei and Jordan, 2006]. The variational distribution is chosen as a local minimum of the KL divergence from the true posterior:

$$q^*(\theta) := q(\theta|\eta^*) \text{ where } \eta^* := \underset{\eta}{\operatorname{argmin}} KL(q(\theta|\eta) || p(\theta|Y)). \quad (2)$$

See Appendix B for details of the variational approximation. Ideally, we would like a global minimum of Eq. (2), but due to the non-convexity of the problem, we can only guarantee finding a local minimum. Importantly for the assessment of co-clustering, knowledge of η^* allows us to approximate the posterior probability $\zeta_{gk}(\eta^*) := E_{q^*}[z_{gk}]$, the posterior probability of gene g belonging to cluster k . We write ζ without subscripts to refer to the $n_g \times K$ matrix with entries ζ_{gk} .

Finally, we introduce some additional notation related to data sensitivity that will be useful to describe the bootstrap and the infinitesimal jackknife in Section 4. To assess data sensitivity, we augment our model with scalar per-gene weights, $w_g \geq 0$, where $W = (w_1, \dots, w_{n_g})^\top$, where we define the weighted likelihood and corresponding optimal variational parameter:

$$\log p(Y|\theta, W) = \sum_{g=1}^{n_g} w_g \log p(y_g|\theta) \Rightarrow \eta^*(W) := \underset{\eta}{\operatorname{argmin}} KL(q(\theta;\eta) || p(\theta|Y, W)). \quad (3)$$

Defining $W_1 := (1, \dots, 1)^\top$ we recover the original variational posterior $\eta^* = \eta^*(W_1)$. By setting W to other integer-valued vectors, we can produce the effect of removing or repeating datapoints, since $p(Y|\theta)$ is exchangeable in y_g . In particular, by drawing n_b bootstrap weights $W_b \sim \text{Multinomial}(n_b, n_b^{-1})$, for $b = 1, \dots, n_b$, the bootstrap distribution of a function $\phi(\zeta(\eta^*))$ can be approximated with the draws $\phi(\zeta(\eta^*(W)))$. In the remainder of the paper, in a slight abuse of notation, we will write $\phi(W)$ in place of $\phi(\zeta(\eta^*(W)))$ below when the meaning is clear from the context.

§ 3. Clustering stability measures. To quantify the stability of a clustering procedure, we must first define measures of similarity between different clustering outputs. In particular, we will consider the similarity between the clustering $\zeta = \zeta(W_1)$, which is clustering at the optimum η^* ,

and $\tilde{\zeta} := \zeta(W_b)$ at bootstrap weights W_b . We will use three clustering similarity measures: the Fowlkes-Mallows index² [Fowlkes and Mallows, 1983], the normalized mutual information [Vinh et al., 2010], and the bootstrap standard deviation of the elements of the matrix $\zeta(W_b)$. Let $P(k_1, k_2) = \zeta_{k_1}^\top \zeta_{k_2}$, and denote the entropy of a distribution $H(P) := \sum_k P(k) \log P(k)$. Treating ζ as fixed, the two similarity measures can be written as:

$$\phi_{FM}(\tilde{\zeta}) = \frac{\sum_{g_1 g_2} (\zeta_{g_1}^\top \zeta_{g_2}) (\tilde{\zeta}_{g_1}^\top \tilde{\zeta}_{g_2})}{\sqrt{\sum_{g_1 g_2} (\zeta_{g_1}^\top \zeta_{g_2})^2 \cdot (\tilde{\zeta}_{g_1}^\top \tilde{\zeta}_{g_2})^2}} \quad \phi_{MI}(\tilde{\zeta}) = \frac{\sum_{k_1 k_2} P(k_1, k_2) \log \left(\frac{P(k_1, k_2)}{P(k_1)P(k_2)} \right)}{\sqrt{H(P) H(\tilde{P})}} \quad (4)$$

See Appendix D for details. Both yield scores ranging between 0 and 1, where the higher the scores, the more similar the clusterings are. Note that while we focus on these measures, the procedure described below can be applied to any similarity measures $\phi(\tilde{\zeta})$.

§ 4. Data sensitivity. We now derive a local approximation to the bootstrap using the weight notation from Section 2. Noting that Eq. (3) is well-defined even for non-integer values of W , and observing that $KL(q(\theta; \eta) || p(\theta|Y, W))$ is smooth in both η and W , it follows that $\eta^*(W)$ is smooth in W in a neighborhood of W_1 . Using the results from Giordano et al. [2017, Appendix D], and adopting the shorthand notation $KL(\eta, W) := KL(q(\theta; \eta) || p(\theta|Y, W))$, we can then calculate a “weight sensitivity matrix” S as

$$S := \left. \frac{d\eta^*(W)}{dW} \right|_{W=W_1} = - \left(\frac{\partial^2 KL(\eta, W)}{\partial \eta \partial \eta^\top} \right)^{-1} \left. \frac{\partial^2 KL(\eta, W)}{\partial \eta \partial W} \right|_{W=W_1}. \quad (5)$$

Although Eq. (5) would be tedious to calculate by hand, it can be calculated exactly using automatic differentiation in just a few lines of code (see Appendix C for more details).

Using S , and a single-term Taylor expansion, we can approximate $\eta^*(W)$ and, in turn, a clustering metric $\phi(W)$:

$$\eta^*(W) \approx \eta_{Lin}^*(W) := \eta^* + S(W - W_1) \quad (6)$$

$$\phi(W) = \phi(\zeta(\eta^*(W))) \approx \phi_{Lin}(W) := \phi(\zeta(\eta_{Lin}^*(W))). \quad (7)$$

Note that the quantities ζ , which are probabilities and must lie between 0 and 1, can be expected to be extremely non-linear functions of η^* , but they can be calculated quickly for any given η^* . We take advantage of this fact to make a linear approximation only on η^* rather than calculating $\frac{d\phi_{gk}}{dW}$ directly.

This is nearly equivalent to the “infinitesimal jackknife” of Jaeckel [1972] (see also Efron [1982, Chapter 6]), where η^* is thought of as a statistic depending on the data y_g . The only difference is that we linearize η^* rather than the full statistic ϕ . In order to avoid confusion with the jackknife estimator of variance, we will refer to $\phi_{Lin}(W_b)$ as the “linear bootstrap” in Section 5 below. Note that the right-hand side of Eq. (6) involves only a matrix multiplication once S has been calculated, but evaluating $\eta^*(W)$ exactly for $W \neq W_1$ typically involves re-solving the optimization problem Eq. (3). So although $\phi_{Lin}(W_b)$ is only an approximation, it can generally be calculated much more quickly than $\phi(W)$ (as is shown in Table (1) below).

§ 5. Results. We optimized Eq. (3) in Python using the `trust-ncg` method of `scipy-optimize` [Jones et al., 2001] using an initialization based on K-means. We calculated the necessary derivatives for the optimization and for Eq. (5) using the automatic differentiation library `autograd` [Maclaurin et al., 2015]. See Appendix C for details.

We first found a high-quality optimum for the original dataset (that is, at $W = W_1$) by choosing lowest KL divergence achieved amongst 200 random restarts³. We take this optimum to be η^* , the value at which we calculate the sensitivity S in Eq. (5). Then, for $n_b = 200$ different bootstrap

²Note that we extend the traditional Fowlkes-Mallows index to be a function of the posterior probabilities ζ_{gk} . See Appendix D for more details.

³Our results are not quite as good if we take η^* to be an optimum chosen after only 10 rather than 200 initializations – see Appendix E for more details and discussion.

	η^* (200 inits)	η_{Cold}^* (10 inits)	η_{Warm}^* (1 init)	η_{Lin}^* (given S)	S
Time (s):	16088	931	53	0.0003	12

Table 1: Median times to compute each bootstrap sample (or related quantities)

weights W_b , we calculate three different estimates of $\eta^*(W_b)$: “warm starts”, $\eta_{Warm}^*(W_b)$, which optimize $\eta^*(W_b)$ starting at η^* ; “cold starts”, $\eta_{Cold}^*(W_b)$ which optimize $\eta^*(W_b)$ taking the best of ten new random K-means initializations, and the linear bootstrap estimates, which are $\eta_{Lin}^*(W_b)$ of Eq. (6). For each of these three optima, we compare the bootstrap distribution of the three stability measures of Section 3. The median times to calculate each of these measures are given in Table (1).

Fig. (1) shows the distribution of $\phi_{MI}(W_b)$ and $\phi_{FM}(W_b)$ for the three measures. Although the bootstrap based on is $\eta_{Lin}^*(W_b)$ biased slightly upwards relative to both of the actual bootstraps, it is a good approximation to the warm start bootstrap.

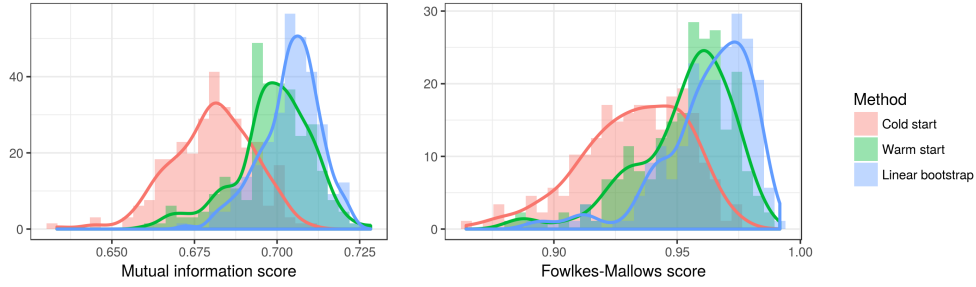


Figure 1: Cluster quality

Finally, we look at the bootstrap standard deviation of the elements of the matrix $\zeta(W_b)$. Fig. (2) shows the relationship between the co-clustering standard deviation as measured by $\eta_{Warm}^*(W_b)$ on the x-axis and $\eta_{Lin}^*(W_b)$ or $\eta_{Cold}^*(W_b)$ on the y-axes. Each point in the graph corresponds to a single value of W_b , so each graph contains $B = 200$ points. Because the vast majority pairs have very small standard deviation in both measures of the graph, we condition on at least one standard deviation being larger than 0.03. For both the cold start and the linear bootstrap, most of the mass lies on the diagonal, indicating a good qualitative correspondence with the warm start, though there is more frequent extreme deviation in the linear bootstrap.

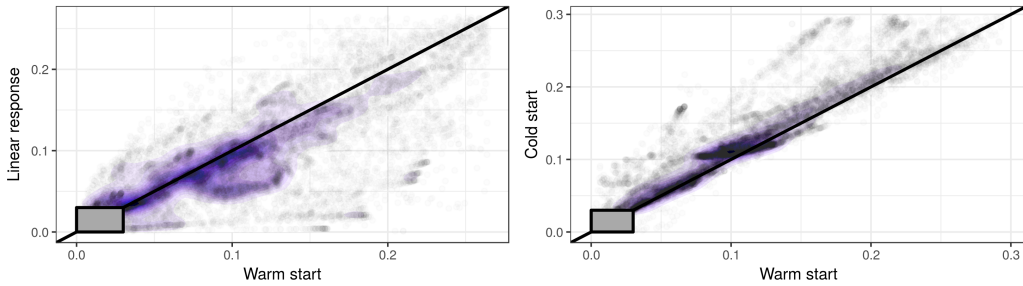


Figure 2: Standard deviations of elements of the co-clustering matrix for a randomly selected subset of genes. Pairs with standard deviations < 0.03 on both axes are not shown.

§ 6. Discussion. In this work, we studied the stability of time-course gene expression clustering, using a BNP model and MFVB inference. We compared the bootstrap, a traditional but computationally intensive approach to assess stability with a fast, approximate stability assessment procedure, the linear bootstrap. Instead of re-sampling the data and refitting the model a large number of times, the linear bootstrap leverages auto-differentiation tools to obtain a first order approximation of the re-sampling scheme. We show that the linear bootstrap is a fast and reasonably accurate alternative to the full bootstrap.

References.

- D. M. Blei and M. I. Jordan. Variational inference for Dirichlet process mixtures. *Bayesian Analysis*, 1(1):121–143, 2006.
- Bradley Efron. *The jackknife, the bootstrap and other resampling plans*. SIAM, 1982.
- Thomas S Ferguson. A bayesian analysis of some nonparametric problems. *The annals of statistics*, pages 209–230, 1973.
- E. B. Fowlkes and C. L. Mallows. A method for comparing two hierarchical clusterings. *Journal of the American Statistical Association*, 78(383):553–569, 1983. ISSN 01621459. URL <http://www.jstor.org/stable/2288117>.
- Ryan Giordano, Tamara Broderick, and Michael I Jordan. Covariances, robustness, and variational bayes. *arXiv preprint arXiv:1709.02536*, 2017.
- Louis A Jaeckel. The infinitesimal jackknife. Technical report, Bell Laboratories, Murray Hill, NJ, 1972. Memorandum MM 72-1215-11.
- E. Jones, T. Oliphant, P. Peterson, et al. SciPy: Open source scientific tools for Python, 2001. URL <http://www.scipy.org/>.
- Yihui Luan and Hongzhe Li. Clustering of time-course gene expression data using a mixed-effects model with B-splines. 19(4):474–482, 2003. doi: 10.1093/bioinformatics/btg014.
- D. Maclaurin, D. Duvenaud, and R. P. Adams. Autograd: Effortless gradients in numpy. In *International Conference on Machine Learning 2015 AutoML Workshop*, 2015.
- R. M. Neal and G. E. Hinton. A view of the EM algorithm that justifies incremental, sparse, and other variants. In *Learning in Graphical Models*, pages 355–368. Springer, 1998.
- Jayaram Sethuraman. A constructive definition of dirichlet priors. *Statistica sinica*, pages 639–650, 1994.
- Jason E. Shoemaker, Satoshi Fukuyama, Amie J. Einfeld, Dongming Zhao, Eiryo Kawakami, Saori Sakabe, Tadashi Maemura, Takeo Gorai, Hiroaki Katsura, Yukiko Muramoto, Shinji Watanabe, Tokiko Watanabe, Ken Fuji, Yukiko Matsuoka, Hiroaki Kitano, and Yoshihiro Kawaoka. An Ultrasensitive Mechanism Regulates Influenza Virus-Induced Inflammation. *PLoS Pathogens*, 11(6):1–25, 2015. ISSN 15537374. doi: 10.1371/journal.ppat.1004856.
- John D Storey, Wenzhong Xiao, Jeffrey T Leek, Ronald G Tompkins, and Ronald W Davis. Significance analysis of time course microarray experiments. *Proceedings of the National Academy of Sciences of the United States of America*, 102(36):12837–42, 2005. ISSN 0027-8424. doi: 10.1073/pnas.0504609102. URL <http://www.pnas.org/content/102/36/12837.abstract>.
- Nguyen Xuan Vinh, Julien Epps, and James Bailey. Information theoretic measures for clusterings comparison: Variants, properties, normalization and correction for chance. *Journal of Machine Learning Research*, 11(Oct):2837–2854, 2010.
- Bin Yu et al. Stability. *Bernoulli*, 19(4):1484–1500, 2013.

§ A. Data description and processing. We use the publicly available mice micro array data set [Shoemaker et al., 2015]. Mice were infected with different influenza viruses, and gene expression was assessed at 14 time points after infection. We focus on the influenza virus “A/California/04/2009”, a mildly pathogenic virus from the 2009 pandemic season. We normalize the data as described in Shoemaker et al. [2015]. We then apply the differential analysis tool EDGE between the influenza infected mice and control mice [Storey et al., 2005]. EDGE yields for each gene a p-value assessing how differently the genes behave between the two conditions. We then rank the genes from most significantly differentially expressed, to least significantly expressed and perform all downstream analysis on the top 1000 genes.

The observations are unevenly spaced, with more frequent observations at the beginning. As shown Fig. (3), each gene also has multiple measurements at each time point (called biological replicates). By modeling gene expression as a smooth function, via a B-spline basis, we naturally model the time aspect of the data, as well as provide an easy framework for including biological replicates in the clustering. The reader may observe that the sparse observations at later times leads to apparent non-smoothness in the fitted time series at late times, though the B-splines enforce smoothness in actual calendar time as desired.

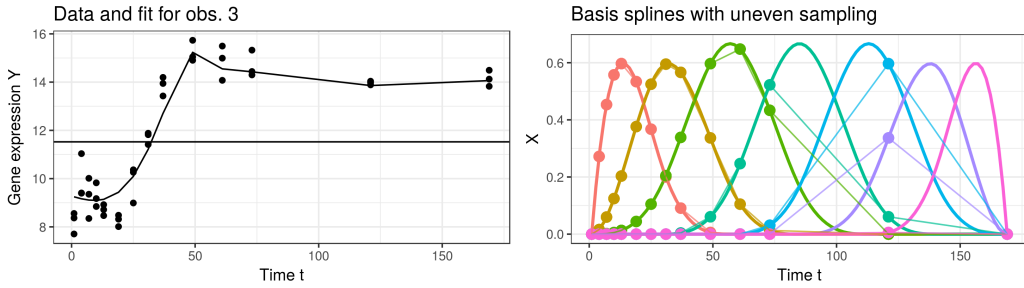


Figure 3: Data and splines

§ B. Variational Inference. We used the following priors:

$$\begin{aligned}\alpha &= 2 \\ \beta_{ki} &\stackrel{iid}{\sim} \mathcal{N}(0.38, 0.10^{-1}) \\ b_g &\stackrel{iid}{\sim} \mathcal{N}(0, 0.10^{-1}) \\ \tau &:= \sigma^{-2} \sim \text{Gamma}(0.10, 10.00) \text{ (shape / scale parameterization)}\end{aligned}$$

The variational approximation was

$$q(\theta|\eta) = \delta(\beta) \delta(\tau) \prod_{k=1}^K \left\{ q(\nu_k) \prod_g q(z_{gk}) q(b_g | z_{gk} = 1) \right\},$$

where $\delta(\cdot)$ denotes a point mass at a parameterized location [Neal and Hinton, 1998]⁴, $q(\nu_k)$ is a beta distribution, $q(z_{gk})$ is a multinomial distribution, $q(b_g | z_{gk} = 1)$ is a normal distribution, and η denotes the vector of parameters for all these distributions. With this approximation, we seek $\eta^* := \underset{\eta}{\operatorname{argmin}} KL(q(\theta|\eta) || p(\theta|Y))$. See Appendix C for details of the optimization.

§ C. Optimization . Note that by parameterizing $q(b_g, z_g) = \prod_{k=1}^K q(b_g | z_{gk} = 1) q(z_g)$, the updates for $q(b_g, z_g)$ have a closed form given $q(\beta, \tau, \nu)$. Denote the parameters for $q(b_g, z_g)$ as η_{local} and the parameters for $q(\beta, \tau, \nu)$ as η_{global} , and write

$$\hat{\eta}_{local}(\eta_{global}) := \underset{\eta_{local}}{\operatorname{argmin}} KL(\eta_{global}, \eta_{local}),$$

⁴Technically, a true point mass does not have a well-defined KL divergence with respect to the Lebesgue measure on β and τ . But $\delta(\beta; \eta_\beta)$ can be thought of as a density with constant entropy, and where $\mathbb{E}_\delta[\beta] \approx \eta_\beta$. Such a distribution can be approximated arbitrarily closely with a multivariate normal distribution with vanishing variance, for example.

we can write the optimization problem Eq. (2) as a function of η_{global} only:

$$\eta_{global} = \underset{\eta_{global}}{\operatorname{argmin}} KL(\eta_{global}, \hat{\eta}_{local}(\eta_{global})) . \quad (8)$$

This is valuable because the size of η_{local} grows with the number of genes, but the size of η_{global} does not. In addition to speeding up optimization, Eq. (8) can be easily differentiated using autograd to calculate the sensitivity matrix S in Eq. (5).

To solve Eq. (8), we use a combination of Newton and quasi-Newton methods. We first choose an initialization by fitting individual B-splines to each gene expression, and use K-means to cluster the coefficients; the centroids were used to initialize the variational means for β_K . From this initialization, we ran BFGS for 300 iterations; at the point where BFGS terminated, we computed the Hessian of the KL objective, Eq. (8). This Hessian was used as a preconditioner for the final Newton trust region steps, which was iterated to convergence. Hessians were computed using autograd [Maclaurin et al., 2015], while BFGS and the newton trust-region routines were done with the BFGS and trust-ncg methods of scipy-optimize [Jones et al., 2001], respectively.

§ D. Clustering stability measures . In this work, we focus on two standard clustering stability measures: the Fowlkes-Mallows index and the Normalized mutual information. As mentioned in 3, we adapt the stability measures to be a function of ζ and $\tilde{\zeta}$. We here describe in more details those similarity measures and how we adapted them for our use case.

First, we will take a closer look at the Fowlkes-Mallows index [Fowlkes and Mallows, 1983]. Ignoring for the moment the variational distribution, consider a general clustering algorithm that outputs binary indicators z_{gk} for gene g belonging to cluster k . Suppose two different runs of the algorithm (e.g. runs with two different initializations) give two different outputs z_{gk} and \tilde{z}_{gk} . Then the Fowlkes-Mallows similarity index is defined as

$$FM = \frac{\sum_{g_1 g_2} C_{g_1 g_2} \tilde{C}_{g_1 g_2}}{\sqrt{(\sum_{g_1 g_2} C_{g_1 g_2}^2) \cdot (\sum_{g_1 g_2} \tilde{C}_{g_1 g_2}^2)}} \quad (9)$$

where $C_{g_1 g_2} := \sum_{k=1}^K z_{g_1 k} z_{g_2 k}$ is the indicator that genes g_1 and g_2 are clustered together under the first clustering; and $\tilde{C}_{g_1 g_2}$ denotes the same quantity under the second clustering. The numerator in Eq. (9) then counts the number of gene pairs that were co-clustered by both two clustering results, and the denominator normalizes the index to be between 0 and 1; hence, values closer to 1 suggest a more similar clustering.

We modify this definition slightly for our case since we have more than just binary indicators: we have posterior probabilities for z_{gk} approximated by the variational distribution. This then gives the probability of co-clustering under the variational distribution, $\mathbb{E}_{q^*}[C_{g_1 g_2}]$. Having two different clustering results now corresponds to having two different variational distributions for z . To measure clustering similarity here, we simply replace $C_{g_1 g_2}$ and $\tilde{C}_{g_1 g_2}$ in Eq. (9) with $\mathbb{E}_{q^*}[C_{g_1 g_2}]$ and $\mathbb{E}_{\tilde{q}^*}[C_{g_1 g_2}]$, their expectations under two different variational distributions.

Now, let's turn to the normalized mutual information score. Let q and \tilde{q} be two different variational distributions, with $\mathbb{E}_{q^*}[z_{gk}] := \zeta_{gk}$ and $\mathbb{E}_{\tilde{q}^*}[z_{gk}] := \tilde{\zeta}_{gk}$. Suppose we consider the distribution on labels induced by drawing a random gene g , and then drawing the labels $k_1|g \sim q(z_g)$ and $k_2|z_g \sim \tilde{q}(z_g)$. Then define $P(k_1) = \frac{1}{n_g} \sum_g \zeta_{gk_1}$, the probability of cluster k_1 under the first variational distribution, and $\tilde{P}(k_2) = \frac{1}{n_g} \sum_g \tilde{\zeta}_{gk_2}$, the probability of cluster k_2 under the second variational distribution; also let $P(k_1, k_2) = \frac{1}{n_g} \sum_g \zeta_{gk_1} \tilde{\zeta}_{gk_2}$, the joint cluster probabilities. Then the normalized mutual information score for clustering similarity is given by

$$NMI = \frac{\sum_{k_1 k_2} P(k_1, k_2) \log\left(\frac{P(k_1, k_2)}{P(k_1) \tilde{P}(k_2)}\right)}{\sqrt{(\sum_k P(k) \log P(k)) \cdot (\sum_k \tilde{P}(k) \log \tilde{P}(k))}} \quad (10)$$

The numerator is the mutual information between the two clustering outputs defined by the variational distributions q and \tilde{q} , with a larger mutual information representing more similar clusterings; the denominator then normalizes such that the score is between 0 and 1.

§ E. Local Optima. Many unsupervised clustering problems exhibit multiple local optima in the objective function, even for permutation-invariant quantities like co-clustering measures, and the problem described in the present work is no exception. Measures of uncertainty which are based on local information (like the infinitesimal jackknife) cannot be expected to capture the frequentist variability due to different initializations leading to substantively different local optima. The fact that the cold starts have lower-quality co-clustering than the warm starts in Fig. (1) indicates that there exist different local optima relatively far from η^* . In this section, we briefly discuss two additional observations concerning local optima.

One might first ask whether the local optima found by the cold start are much worse than those found by the warm start. The distribution of KL divergences across the bootstrap samples is shown in Fig. (4). Each point in Fig. (4) corresponds to two different estimates at the same weights W_b , so there are $B = 200$ points in each graph. The linear response KL divergence, which is not evaluated at an actual optimum, is larger than the corresponding optimal value, as expected. Note that the cold start KL divergence is not actually noticeably worse than the warm start KL divergence, suggesting that there may be meaningful frequentist variability due to local optima that is not captured by either $\eta_{Warm}^*(W_b)$ nor $\eta_{Lin}^*(W_b)$.

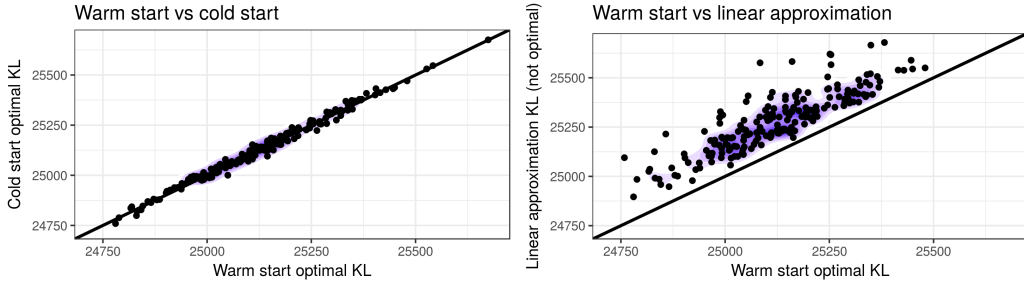


Figure 4: Distribution of KL divergence relative to the warm start

Finally, we note that the results in Section 5 depend in part on the fact that we are re-starting the optimization in our bootstrap samples at a high-quality optimum, η^* , chosen as the best out of 200 random restarts. If, instead, we set η^* to be the best optimum found after only 10 random restarts, the results are not quite as good, as seen in Fig. (5). This is probably due both to the base set of cluster assignments, ζ in Eq. (4), is not as high-quality an optimum, and to the fact that optima near η^* , being of lower quality, is chosen less often during the bootstrap procedure.

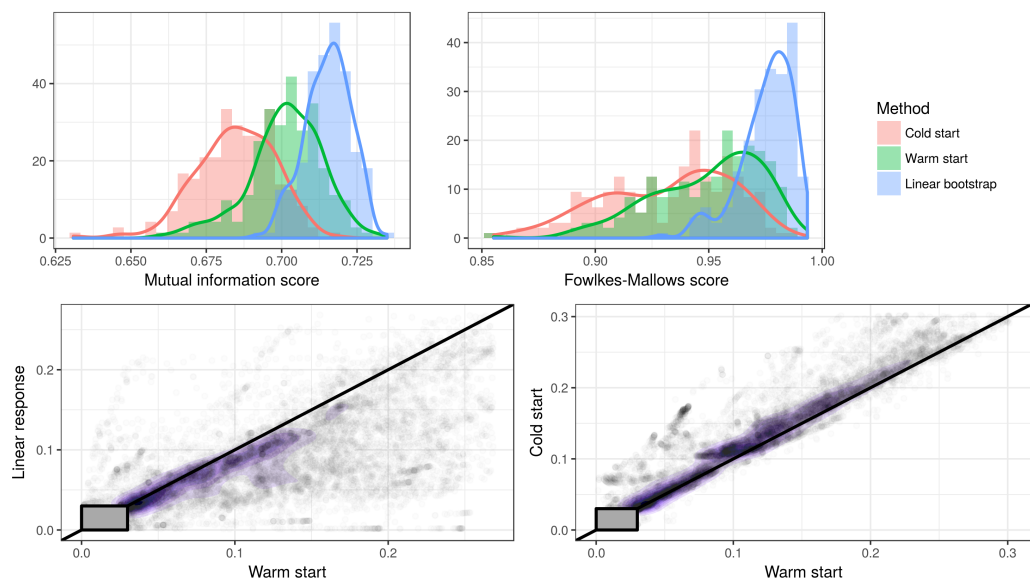


Figure 5: Results with an initial optimum based on only 10 random restarts rather than 200