On Bayesian sparse canonical correlation analysis via Rayleigh quotient framework

Qiuyun Zhu*1 and Yves Atchade†1

1Department of Mathematics and Statistics, Boston University

Abstract

We propose a semi-parametric Bayesian method for the principal canonical pair that employs the scaled Rayleigh quotient as a quasi-log-likelihood with the spike-and-slab prior as the sparse constraints. Our approach does not require a complete joint distribution of the data, and as such, is more robust to non-normality than current Bayesian methods. Moreover, simulated tempering is used for solving the multi-modality problem in the resulting posterior distribution. We study the numerical behavior of the proposed method on both continuous and truncated data, and show that it compares favorably with other methods. As an application, we use the methodology to maximally correlate clinical variables and proteomic data for a better understanding of covid-19 disease. Our analysis identifies the protein Alpha-1-acid glycoprotein 1 (AGP 1) as playing an important role in the progression of Covid-19 into a severe illness.

1 Introduction

Canonical correlation analysis is a statistical technique –dating back at least to [1] – that is used to maximally correlate multiple datasets for joint analysis. The technique has become a fundamental tool in biomedical research where technological advances have led to a huge number of multi-omic datasets ([2]; [3]; [4]). Over the past two decades, limited sample sizes, growing dimensionality, and the search for meaningful biological interpretations, have led to the development of sparse canonical correlation analysis ([2]), where a sparsity assumption is imposed on the canonical correlation vectors.

This work falls under the topic of the Bayesian estimation of sparse canonical correlation vectors. Model-based approaches to canonical correlation analysis were developed in the mid 2000’s (see e.g., [5]), and paved the way for a Bayesian treatment of canonical correlation analysis ([6];[7]) and sparse canonical correlation analysis ([8]). However an serious shortcoming of such a Bayesian treatment is that this approach naturally requires a complete specification of the joint distribution of the data, so as to specify the likelihood function. This requirement is a serious limitation in many applications, where the data generating process is poorly understood, for example, image data.

*rachyun@bu.edu
†atchade@bu.edu
Even when the generating process is well understood, we may still encounter highly intractable likelihood, which is very common due to unobserved variables.

On the other hand, there exist several frequentist methods (see e.g., [8], [9] and [10]) that first modify the canonical correlation analysis problem by imposing certain sparse constraints or regularizations, and then directly solve the modified problem. For these approaches, the aforementioned model misspecification problem seems much less serious. Among these methods, [10] formulated the sparse canonical correlation analysis problem based on Rayleigh quotient, which typically possesses many local maxima, and proposed a two-stage approach to solve this nonconvex problem. Compared to the previous works, their method not only enjoys favorable statistical guarantees, but also has minimal assumptions on the covariance matrix. In addition, via Rayleigh quotient, their method overcomes problems caused by the singularity of the sample covariance matrix estimators in the high-dimensional setting. Their approach first solves a convex relaxation of the nonconvex problem to obtain a good initialization point, and then uses this point to run gradient descent. As stated in [10], the quality of the initialization point is crucial for the success of their method. However, the computational cost of obtaining it (via solving the convex relaxation) can be extremely high in practice, especially in the high-dimensional setting. Even if such an initialization can be computed, it is very hard to verify the quality of the initialization in practice, although the criterion is well established in theory.

In this work, we propose a flexible semi-parametric Bayesian method that overcomes the above challenges faced by Bayesian and frequentist methods. We summarize our contributions as follows.

First, inspired by [10], we develop a (quasi-)Bayesian approach that employs the scaled Rayleigh quotient as a quasi-log-likelihood of the principal canonical pair, with the spike-and-slab prior ([11]) as the sparse constraints. Our approach does not require a complete joint distribution of the data, and as such, is more robust to non-normality than current Bayesian methods (e.g., [6]). Moreover, our method works well with a wide range of the covariance estimators, which are used to form the Rayleigh quotient. For example, we observe that in our experiments, both the sample covariance matrix estimator and Kendall’s-tau-based covariance matrix estimator ([12]) work well, even when they are singular. Indeed, we can straightforwardly extend our method to other generalized eigenvalue problems via Rayleigh quotient in a similar spirit to [10], but we do not include it in the paper.

Second, as the targeted posterior distribution is complicated, we apply simulated tempering to solve multi-modality. It is well-known that empirically sampling deals with multi-modality better than optimization. Moreover, multi-modality can be further mitigated in sampling by tempering ideas such as parallel tempering, simulated tempering and related ([13]; [14]). Typically, simulated tempering can sample from a complicated posterior distribution without stringent requirements on initialization. This address the challenge faced by [10] regarding the initialization. Indeed, in our numerical studies, our algorithm performs better than the method in [10], in terms of both accuracy and efficiency.

The remaining of the paper is organized as follows. In Section 2 we introduce our Bayesian approach and the sampling algorithm. In Section 3 we study the numerical behavior of the proposed method on both continuous and truncated data, and compare it with other methods. In Section 4 we also explore the behavior of the method on a real data analysis that aims to correlate clinical and proteomic data from covid-19 patients, for a better understanding of the
disease. Our analysis identifies Alpha-1-acid glycoprotein 1 (AGP 1) as playing an important role in the progression of Covid-19 into a severe illness.

2 Bayesian Sparse canonical correlation analysis via Rayleigh quotient

Let \((X, Y) \in \mathbb{R}^{p_x} \times \mathbb{R}^{p_y}\) be a pair of high-dimensional zero-mean random vectors with joint distribution \(f\), with covariance matrices \(\Sigma_x \overset{\text{def}}{=} \mathbb{E}(XX^T)\), \(\Sigma_y \overset{\text{def}}{=} \mathbb{E}(YY^T)\) and \(\Sigma_{xy} \overset{\text{def}}{=} \mathbb{E}(XY^T)\).

The sparse canonical correlation analysis problem involves finding a pair of principal canonical correlation vectors of \(f\) that are sparse, i.e., with a small number of nonzero entries. A pair of vectors \((v_{x*}, v_{y*}) \in \mathbb{R}^{p_x} \times \mathbb{R}^{p_y}\) is called principal canonical correlation vectors of \(f\), if they solve the following optimization problem:

\[
\max_{v_x \in \mathbb{R}^{p_x}, v_y \in \mathbb{R}^{p_y}} v_x^T \Sigma_{xy} v_y \quad \text{s.t.} \quad v_x^T \Sigma_x v_x = v_y^T \Sigma_y v_y = 1. \tag{1}
\]

If we define \(p \overset{\text{def}}{=} p_x + p_y\), and matrices

\[
A = \begin{bmatrix} 0 & \Sigma_{xy} \\ \Sigma_{xy}^T & 0 \end{bmatrix} \quad \text{and} \quad B = \begin{bmatrix} \Sigma_x & 0 \\ 0 & \Sigma_y \end{bmatrix}, \tag{2}
\]

then it is easily seen that (1) share the same set of optimal solutions as the following GEP:

\[
\max_{\theta = (v_{x*}^T, v_{y*}^T)^T \in \mathbb{R}^p} \theta^T A \theta \quad \text{s.t.} \quad \theta^T B \theta = 2. \tag{3}
\]

Clearly, to find an optimal solution of (3), it is equivalent to find an optimal solution of

\[
\max_{\theta = (v_{x*}^T, v_{y*}^T)^T \in \mathbb{R}^p} \mathcal{R}(\theta) \overset{\text{def}}{=} \frac{\theta^T A \theta}{\theta^T B \theta}, \quad \text{s.t.} \quad \theta^T B \theta = 2. \tag{4}
\]

which we denote by \(\theta_* = (v_{x*}^T, v_{y*}^T)^T\). The objective function \(\mathcal{R}\) in (4) is called the (generalized) Rayleigh quotient of \(A\) and \(B\).

Note that (4) requires specifying matrices \(A\) and \(B\), which are typically unknown. In practice, one often constructs estimators of \(A\) and \(B\), denoted by \(\hat{A}\) and \(\hat{B}\), respectively, based on \(n\) i.i.d. samples \(Z \overset{\text{def}}{=} \{(X_1, Y_1), \ldots, (X_n, Y_n)\}\) drawn from \(f\). Specifically, given \(Z\), one first constructs estimators of \(\Sigma_x\), \(\Sigma_y\), and \(\Sigma_{xy}\), denoted by \(\hat{\Sigma}_x\), \(\hat{\Sigma}_y\), and \(\hat{\Sigma}_{xy}\), respectively, and then constructs \(\hat{A}\) and \(\hat{B}\) from \(\hat{\Sigma}_x\), \(\hat{\Sigma}_y\), and \(\hat{\Sigma}_{xy}\), in the same fashion as in (2). (In Section 3, we will provide some examples of how to construct the estimators \(\hat{\Sigma}_x\), \(\hat{\Sigma}_y\), and \(\hat{\Sigma}_{xy}\).) Now we focus on maximizing the sample version of the Rayleigh quotient defined as

\[
\mathcal{R}_n(\theta; Z) \overset{\text{def}}{=} \frac{\theta^T \hat{A} \theta}{\theta^T \hat{B} \theta}, \quad \theta \in \mathbb{R}^{p_x + p_y}.
\]

It is worth mentioning that in the high-dimensional regime, the constructed estimators \(\hat{\Sigma}_x\) and \(\hat{\Sigma}_y\), e.g. sample covariance matrices, are usually singular, which can result in non-uniqueness of the solutions. To avoid that, sparsity constraint is typically needed. Some frequentist approaches for solving this problem is to impose \(l_1\) regularization (see e.g., [8]) or \(l_0\) constraint (see e.g., [10]). Our Bayesian approaches use it as prior information.
2.1 A Bayesian approach

We develop a Bayesian inference procedure of $\theta_*$, by using the function

$$\theta \mapsto (2n/\sigma^2)R_n(\theta; Z)$$

as quasi-log-likelihood, where $\sigma > 0$ is a scale parameter. For the prior distribution on $\theta$, we choose the spike-and-slab distribution, which has been frequently used in Bayesian variable selection (see e.g., [15]). Given a variable selection (hyper-)parameter $\delta \in \Delta \equiv \{0,1\}^p$, let the conditional distribution of $\theta$ be

$$\pi(\theta|\delta) = \prod_{j=1}^p \pi(\theta_j|\delta_j), \quad \text{where} \quad \theta_j|\delta_j = \theta_j|\delta_j \sim \begin{cases} N(0, \rho_1^{-1}), & \text{if } \delta_j = 1 \\ N(0, \rho_0^{-1}), & \text{if } \delta_j = 0 \end{cases},$$

and $\rho_0 > \rho_1 > 0$ are precision parameters. In addition, the prior distribution of $\delta$ is an independent product of Bernoulli distributions with mean $0 < q < 1$. If we combine the spike-and-slab prior with the quasi-log-likelihood in (5), we then obtain the posterior distribution

$$\Pi(\delta, d\theta|Z) \propto \exp \left( a \|\delta\|_0 - \frac{\rho_1}{2} \|\theta_\delta\|_2^2 - \frac{\rho_0}{2} \|\theta - \theta_\delta\|_2^2 \right) \exp \left( (2n/\sigma^2)R_n(\theta_\delta; Z) \right) d\theta,$$

where $a \equiv \log(q/(1-q))$, $\theta_\delta$ the entry-wise product of $\theta$ and $\delta$, and $\| \cdot \|_2$ is the Euclidean norm. Note that the posterior distribution $\Pi(\cdot|Z)$ depends on three user-defined parameters, namely $q$, $\rho_0$ and $\rho_1$. Throughout this work, we use $\rho_0 = 10$, $\rho_1 = 0.5$, and $q = p^{-1.5}$.

Clearly $\Pi(\cdot|Z)$ is not a posterior distribution in the standard sense, since $(2n/\sigma^2)R_n$ is not a proper log-likelihood. Bayesian inference using loss functions other than the negative log-likelihood has been extensively studied in recent years. This extension of the classical Bayesian framework remains a coherent belief update mechanism ([16]), and yields posterior distributions with similar large sample theoretical properties ([17]). However some care must be taken when comparing inference drawn from such generalized posterior distributions and their frequentist counterparts. For instance coverage from resulting credible sets do not typically match those of their corresponding frequentist confidence sets without further calibration ([18]). In our setting, if so desired, such calibration can be achieved by adjusting $\sigma$ according. However in work we set $\sigma = 1$ throughout.

2.2 Markov Chain Monte Carlo

To sample from (6), we face the challenge that the distribution has multiple modes, with typically complicated landscape. Indeed, all the generalized eigenvectors of the pair $(\hat{A}, \hat{B})$ are stationary points of $R_n(\cdot; Z)$, with some of them being local maxima. In addition, if $(u, v)$ maximizes $R_n(\cdot; Z)$, so does $(-u, -v)$. To deal with this challenge, we leverage simulated tempering, a MCMC sampling technique that has shown promising performance in sampling from multi-modal distributions (see e.g., [14]).

Given $K$ temperatures $1 = t_1 < t_2 < \ldots < t_K$, and $K$ positive weights $c_1, \ldots, c_K$, we introduce an extended distribution on $X \overset{\text{def}}{=} \Delta \times \mathbb{R}^p \times \{1, \ldots, K\}$, which is

$$\tilde{\Pi}(\delta, d\theta, k|Z) \propto \frac{1}{c_k} \exp \left( \frac{a}{t_k} \|\delta\|_0 \right) \exp \left( -\frac{\rho_1}{2t_k} \|\theta_\delta\|_2^2 - \frac{\rho_0}{2t_k} \|\theta - \theta_\delta\|_2^2 \right) \exp \left( \frac{2n}{\sigma^2 t_k}R_n(\theta_\delta; Z) \right) d\theta.$$

4
Algorithm 1: Simulated tempering for sparse canonical correlation analysis

**Model Input:** Matrices $\hat{A}, \hat{B}$, $\sigma^2 = 1$, prior parameters $\rho_0, \rho_1, q$.

**MCMC Input:** Number of iterations $N$, batch size $J$, temperatures $1 = t_1 < \ldots < t_K$, weights $(c_1, \ldots, c_K)$, and step-sizes $(\eta_1, \ldots, \eta_K)$.

**Initialization:** Set $k^{(0)} = 1$. Draw $\delta^{(0)} \overset{i.i.d.}{\sim} \text{Ber}(0.5), \forall j = 1, \ldots, p$, and independently $\theta^{(0)} \sim N(0, I_p)$.

for $t = 0$ to $N - 1$, given $(k^{(t)}, \delta^{(t)}, \theta^{(t)}) = (k, \delta, \theta)$ do

1. **Update $\delta$:** Uniformly randomly select a subset $J$ from $\{1, \ldots, p\}$ of size $J$ without replacement, and draw $\tilde{\delta} \sim Q^{(J)}_{k, \delta}(\delta, \cdot)$, where the transition kernel described in (10).

2. **Update $\theta$:** Draw the components of $[\theta]_\delta$ independently from $N(0, \rho_0^{-1} t_k)$. Draw $[\theta]_{\tilde{\delta}} \sim P_{\eta, k, \delta}([\theta]_\delta, \cdot)$, where $P_{\eta, k, \delta}$ denotes the transition kernel of the MALA with step-size $\eta$ and invariant distribution given by $W_{k, \delta}$, whose density is proportional to $[\Pi]$.

3. **Update $k$:** Draw $\tilde{k} \sim T_{\tilde{\delta}, \tilde{\theta}}(k, \cdot)$, where $T_{\tilde{\delta}, \tilde{\theta}}$ is the transition kernel of the Metropolis-Hastings on $\{1, \ldots, K\}$ with invariant distribution given by (12) and random walk proposal that has reflection at the boundaries.

4. **New MCMC state:** Set $(\delta^{(t+1)}, \theta^{(t+1)}, k^{(t+1)}) = (\tilde{\delta}, \tilde{\theta}, \tilde{k})$.

end for

**Output:** $\{(\delta^{(t)}, \theta^{(t)}, k^{(t)}): 0 \leq t \leq N \text{ s.t. } k^{(t)} = 1\}$

To sample from $q_j$, we design a MCMC sampler, which is shown in Algorithm 1.

First, we update $\delta$ by applying a Gibbs sampler to the conditional distribution of $\delta$ given $k$ and $\theta$. Note that the conditional distribution of $\delta_j$ given $k, \theta$ and $\delta_{-j}$, where $\delta_{-j} \overset{\text{def}}{=} (\delta_1, \ldots, \delta_{j-1}, \delta_{j+1}, \ldots, \delta_p)$, is the Bernoulli distribution $\text{Ber}(q_j)$, with probability of success given by

$$q_j \overset{\text{def}}{=} \left\{1 + \exp\left(-\frac{a}{t_k} + \frac{1}{2 t_k} (\rho_1 - \rho_0) \theta^2 \right) \right\} \exp\left(\frac{2n}{\sigma^2 t_k} R_n(\theta_{\delta_{(j,0)}}; Z) - \frac{2n}{\sigma^2 t_k} R_n(\theta_{\delta_{(j,1)}}; Z)\right) \right\}^{-1},$$

where

$$\delta_{(j,0)} \overset{\text{def}}{=} \begin{cases} 0 & i = j \\ \delta_i & i \neq j \end{cases}, \quad \delta_{(j,1)} \overset{\text{def}}{=} \begin{cases} 1 & i = j \\ \delta_i & i \neq j \end{cases}.$$  

If $\delta_{(j,0)} = 0$, we use the convention that $R_n(\theta_{\delta_{(j,0)}}) = -\infty$, so that $||\delta||_0 > 0$. Given $k, \theta$ and $j$, denote the transition kernel on $\Delta$ as $Q^{(J)}_{k, \theta}$ which, given $\delta$, leaves $\delta_i$ unchanged for all $i \neq j$, and draws $\delta_j \sim \text{Ber}(q_j)$. We put several such moves together to update $\delta$. The resulting transition kernel on $\Delta$ can be written as follows: randomly draw a subset $J = \{j_1, \ldots, j_J\}$ of size $J$ from $\{1, \ldots, p\}$, we update $\delta$ using the transition kernel on $\Delta$ given by

$$Q^{(J)}_{k, \theta} \overset{\text{def}}{=} Q_{k, \theta}^{(j_1)} Q_{k, \theta}^{(j_2)} \cdots Q_{k, \theta}^{(j_J)}.$$  

Second, given $k$ and $\delta$, we update $\theta$. We let $[\theta]_\delta$ denote the $\delta$-selected component of $\theta$ listed in their original order: $[\theta]_\delta \overset{\text{def}}{=} \{\theta_j: j \in \{1 \leq k < p : \delta_k = 1\}\}$, and $[\theta]_{\delta^c} \overset{\text{def}}{=} \{\theta_j: j \in \{1 \leq k < p : \delta_k = 0\}\}$. We utilize the fact that the selected components $[\theta]_\delta$ and the unselected components $[\theta]_{\delta^c}$ of $\theta$ are independent conditional on $k$ and $\delta$ to update $\theta$. In addition, given $k$ and $\delta$, the
components of $[\theta]_\delta$ are i.i.d. $N(0, t_k \rho_0^{-1})$ and the distribution of $[\theta]_\delta$, denoted by $W_{k, \delta}$, has density on $\mathbb{R}^{\|\delta\|_0}$ proportional to

$$u \mapsto \exp \left( -\frac{\rho_1}{2t_k} \|u\|^2 + \frac{2n}{\sigma^2 t_k} R_n((u, 0)_\delta; \mathbf{Z}) \right),$$

(11)

where the notation $(u, 0)_\delta$ denotes the vector in $\mathbb{R}^p$ such that $[(u, 0)_\delta] = u$ and $[(u, 0)_\delta] = 0$. Hence we update $[\theta]_\delta$ using a standard Metropolis adjusted Langevin algorithm (MALA) with target distribution $W_{k, \delta}$, and step-size $\eta_k$. For details on MALA, see e.g., [19]. For convenience, let us denote the transition kernel of this Markov chain on $\mathbb{R}^{\|\delta\|_0}$ as $P_{u, k, \delta}$.

Third, given $\delta$ and $\theta$, we update $k$ using a standard Metropolis-Hastings algorithm with a random walk proposal that has reflection at the boundaries. Specifically, at $k$ we propose with equal probability either $k-1$ or $k+1$, except at 1, where we only propose 2, and at $K$, where we only propose $K-1$. Let $T_{\delta, \theta}$ denote the transition kernel on $\{1, \ldots, K\}$ of this Metropolis-Hastings algorithm with invariant distribution

$$i \mapsto \frac{1}{c_i} \exp \left\{ a \|\bar{\delta}\|_0 - \frac{\rho_1}{2t_i} \|\bar{\theta}\|_2^2 - \frac{\rho_0}{2t_i} \|\bar{\theta} - \bar{\delta}\|_2^2 + \frac{2n}{\sigma^2 t_i} R_n(\bar{\delta}; \mathbf{Z}) \right\}.$$  

(12)

Last, we collect samples. Note that these samples have invariant distribution $[7]$, to obtain samples from $[6]$ we can collect samples from the iterations for which $k = 1$.

2.3 Parameter choices and adaptive tuning

Algorithm [1] as outlined above depends on the user-defined parameters $J$, $(t_1, \ldots, t_K)$, $(c_1, \ldots, c_K)$, and $(\eta_1, \ldots, \eta_K)$. The parameter $J$ (the Gibbs sampling batch size) does not greatly impact performance, and setting $J = 100$ works well in most settings. Efficient tuning of temperatures in simulated tempering has received some attention ([13, 20]), however despite some progress ([21]), to the best of our knowledge, there is no practical and scalable algorithm to do so. In our implementation we use variations of the geometric scaling. We refer the reader to Section 3 for specific choices.

We tune the step-sizes $\eta = (\eta_1, \ldots, \eta_K)$ and the weights $(c_1, \ldots, c_K)$ using adaptive MCMC methods, see e.g., [22]. To tune $\eta_k$, we follow the algorithm proposed in [23], with a targeted acceptance probability of 30%. For simulated tempering to visit all temperature levels frequently, the weights $(c_1, \ldots, c_K)$ need to be adequately tuned. We refer the reader to [13] for an extensive discussion of the issue. This problem can be efficiently solved using the Wang-Landau algorithm for simulated tempering as developed in [24]. We follow this approach here. The full adaptive MCMC sampler is presented in Algorithm [??] in the appendix.

2.4 MCMC output processing

Given the output $\{(\delta^{(t)}, \theta^{(t)}, k^{(t)}): 0 \leq t \leq N \text{ s.t. } k^{(t)} = 1\}$ from Algorithms [1] or [??] we use three quarters of the total iterations as burn-in. Then we can treat the remaining output at iteration $t \in T \defeq \{t : \frac{3}{4} N \leq t \leq N \text{ s.t. } k^{(t)} = 1\}$ as samples from the distribution of interest in $[6]$. At iteration $t \in T$, we estimated the principal canonical correlation pair by $\nu^{(t)}$ where $\nu^{(t)} \defeq [\theta^{(t)}]_{\delta^{(t)}}$ is the entry-wise product of $\delta^{(t)}$ and $\theta^{(t)}$. We denote $v_i^{(t)} \defeq (v_i^{(t)} : 1 \leq i \leq p_x)$ and
\(v_y(t) \overset{\text{def}}{=} (v_i(t) : p_x + 1 \leq i \leq p)\). Then we normalize \(v_x(t)\) and \(v_y(t)\) to have unit Euclidean norm as the estimators of \(v_{x*}\) and \(v_{y*}\) at iteration \(t\).

We can obtain a final estimator of the principal canonical pair by processing the output in the following way. To estimate the support of the canonical correlation vector pair \((v_{x*}^T, v_{y*}^T)^T\), we use the support of the mode of \(\{\delta(t) : t \in T\}\), denoted by \(\delta\). We denote the estimated support of \(v_{x*}\) and \(v_{y*}\) by \(\delta_x \overset{\text{def}}{=} (\delta_i : 1 \leq i \leq p_x)\) and \(\delta_y \overset{\text{def}}{=} (\delta_i : p_x + 1 \leq i \leq p)\). Then the canonical correlation pair itself is estimated by the sample average of \(\{|v(t)|_\delta : t \in T\}\) denoted by \(\tilde{v}\). We denote \(\tilde{v}_x \overset{\text{def}}{=} (\tilde{v}_i : 1 \leq i \leq p_x)\) and \(\tilde{v}_y \overset{\text{def}}{=} (\tilde{v}_i : p_x + 1 \leq i \leq p)\). We typically normalize the vectors \(\tilde{v}_x\) and \(\tilde{v}_y\) to have unit Euclidean norm as estimators of \(v_{x*}, v_{y*}\).

### 3 Numerical studies

We extensively test the proposed approach. Since the success of the method is predicated on a good behavior of the MCMC sampler, we first investigate empirically the mixing time of Algorithm 1 as the dimension \(p\) increases. Then we perform a simulation study that compares our approach to the frequentist method Rifle in [10]. The results clearly show the advantage of using the Raleigh quotient in a Bayesian setting. We also investigate the behavior of the proposed method in settings where one of part of samples are subject to truncation, and we compare the results to the method proposed in [12] to handle such mixed datasets. It is shown that our algorithm performs better in variable selection.

#### 3.1 Simulated data generation

To generate the simulated datasets, we follow the approach from [10]. For simplicity, we let \(p_x = p_y\). Then we consider two \((p/2)\)-dimensional random vectors \(X\) and \(Y\) with joint distribution \((X, Y) \sim \mathcal{N}(0, \Sigma)\). Here we assume that

\[
\Sigma = \begin{pmatrix}
\Sigma_x & \Sigma_{xy} \\
\Sigma_{yx} & \Sigma_y
\end{pmatrix}, \quad \Sigma_{xy} = \frac{\lambda_1 \Sigma_x v_{x*} v_{y*}^T \Sigma_y}{\sqrt{v_{x*}^T \Sigma_x v_{x*} \sqrt{v_{y*}^T \Sigma_y v_{y*}}}},
\]

where \(0 < \lambda_1 < 1\) is the largest generalized eigenvalue, and \(v_{x*}\) and \(v_{y*}\) are the principal canonical correlation vectors. Clearly, \(v_{x*}\) and \(v_{y*}\) are maximizer of the Raleigh quotient and \(\lambda_1\) is the maximum value.

For this simulation study, we consider the case when \(\Sigma_x\) and \(\Sigma_y\) are block diagonal matrix with five blocks, each of dimension \(p/10 \times p/10\), where the \((j, j')\)-th element of each block takes value \(0.8^{|j-j'|}\). We let \(\lambda_1 = 0.9\), \((v_{x*})_j = (v_{y*})_j = 1/\sqrt{3}\) for \(j \in \{1, 6, 11\}\), and \((v_{x*})_j = (v_{y*})_j = 0\) otherwise. Then We are able to generate \(n\) samples \((x_i, y_i), i = 1, \ldots, n\) from \(\mathcal{N}(0, \Sigma)\).

#### 3.2 Empirical mixing time of Algorithm 1

We first investigate empirically the mixing time of Algorithm 1 on the simulated data described in Section 3.1 as the dimension \(p\) of the problem increases. Let \(\{X(t), t \geq 0\}\) be the Markov chain generated by Algorithm 1 where \(X(t) = (\delta(t), \theta(t), k(t)) \in X\). As \(t \to \infty\), the distribution of \(X(t)\), denoted by \(\Pi(t)\), converges to \(\Pi\) as given in (7) in total variation by aperiodic ergodic theorem. The rate of convergence is of primary interest as it indicates the efficiency of the proposed
method. We use the approach in \cite{25} (that builds on \cite{26}) to estimate empirically the mixing time of Algorithm \cite{1}. The method is briefly described as follows. Let $P$ denote the transition kernel of the Markov chain $(X^{(t)}, \ t \geq 0)$. Let $\tilde{P}$ be a coupling of $P$ with itself: that is, a transition kernel on $X \times X$ such that $\tilde{P}(x, y), A \times X) = P(x, A)$, $\tilde{P}(x, y), X \times B) = P(y, B)$, for all $x, y \in X$, and all measurable sets $A, B$. The coupling $\tilde{P}$ is constructed in such a way that $\tilde{P}(x, y), \mathcal{D}) > 0$, where $\mathcal{D} \overset{\text{def}}{=} \{(x, x) : x \in X\}$ and $\tilde{P}(x, x), \mathcal{D}) = 1$. We refer the reader to \cite{25} and \cite{26} for more details on the construction of such coupled kernels. The striking point of their method is that even in problems as involved in the one considered in this work, these coupled kernels are easy to construct and simulate.

The method then proceeds as follows. Fix a lag $L \geq 1$. Draw $X^{(0)} \sim \tilde{\Pi}^{(0)}, Y^{(0)} \sim \tilde{\Pi}^{(0)}$ (where $\tilde{\Pi}^{(0)}$ is the initial distribution as given in the initialization step in Algorithm \cite{1}). Draw $X^{(k)}|(X^{(0)}, Y^{(0)}) \sim P^{k}(X^{(0)}, \cdot)$, for all $1 \leq k \leq L$. Then for any $k \geq 1$, draw,

$$(X_{L+k}, Y_{k}) \sim \tilde{P}(X_{L+k-1}, Y_{k-1}), \ldots, (X_{L}, Y_{0}) \sim \tilde{P}(X_{L+k-1}, Y_{k-1}, \cdot), \ k \geq 1.$$ 

Setting the meeting time,

$$\tau^{(L)} \overset{\text{def}}{=} \inf \{k > L : X_{k} = Y_{k-L}\},$$

it then holds under some ergodicity assumptions on $P$ (see \cite{25}) that

$$||\tilde{\Pi}^{(t)} - \Pi||_{TV} \leq E\left[\max\left(0, \frac{\tau^{(L)} - L - t}{L}\right)\right],$$

where $\lfloor x \rfloor$ denote the smallest integer above $x$. The implication of \cite{13} is that we can empirically upper bound the left hand side of \cite{13} by simulating multiple copies of the joint chain as described above and then approximating the expectation on the right hand side of \cite{13} by Monte Carlo.

Specializing this method to our case, we modify Algorithm \cite{1} to construct the coupled kernel $\tilde{P}$. Let $(\delta^{(1,t)}, \theta^{(1,t)}, k^{(1,t)})$ and $(\delta^{(2,t)}, \theta^{(2,t)}, k^{(2,t)})$ denote the states of the two chains at time $t$. For $t \geq 1$, given $(\delta^{(1,L+t)}, \theta^{(1,L+t)}, k^{(1,L+t)}) = (\delta^{(1)}, \theta^{(1)}, k^{(1)})$ and $(\delta^{(2,t)}, \theta^{(2,t)}, k^{(2,t)}) = (\delta^{(2)}, \theta^{(2)}, k^{(2)})$, we detail the modifications of updating steps as follows.

In step 1, to update $\delta^{(1)}$ and $\delta^{(2)}$, we first make use of the same randomly drawn subset $J$. For $i = 1, 2$, drawing $\tilde{\delta}^{(i)} \sim Q_{k,\delta^{(i)}}^{(J)}(\delta^{(i)}, \cdot)$ is equivalent to let $\tilde{\delta}_{i}^{(i)} = \delta_{i}^{(i)}$, and for any $j \in J$, draw $\tilde{\delta}_{j}^{(i)} \sim \text{Ber}(q_{j}^{(i)})$ which we implement in the following way. We first draw a common uniform number $u_{j} \sim \text{Uniform}(0, 1)$, then we obtain $\tilde{\delta}_{j}^{(i)} = 1\{q_{j}^{(i)} \leq u_{j}\}$ for $i = 1, 2$.

In step 2, to update $\theta^{(1)}$ and $\theta^{(2)}$, we partition the indices $\{1, \ldots, p\}$ into four groups: $G_{ab} = \{j : \tilde{\delta}_{j}^{(1)} = a, \tilde{\delta}_{j}^{(2)} = b\}$ for $a, b = 0, 1$. To update the components of $\theta_{G_{00}}^{(1)}$ and $\theta_{G_{00}}^{(2)}$, for any $j \in G_{00}$ we first draw a common standard normal random variables $Z_{j}$, and then obtain $\theta_{j}^{(i)} = t_{k^{(i)}}\rho_{0}^{-1}Z_{j}$ for $i = 1, 2$. To update the components of $\theta_{G_{01}}^{(1)}$ and $\theta_{G_{01}}^{(2)}$, for any $j \in G_{01}$ we again first draw a common standard normal random variables $Z_{j}$, and then obtain $\theta_{j}^{(i)} = t_{k^{(i)}}\rho_{0}^{-1}Z_{j}$ and simultaneously draw $\tilde{\theta}_{j}^{(2)}$ using MALA with proposal $\theta_{j}^{(2)} + \eta_{k(2)}\nabla \log \pi(\theta_{j}^{(2)}) + \sqrt{2}\eta_{k(2)}Z_{j}$, where $\pi(\theta_{j}^{(2)})$ is the marginal posterior distribution of $\theta_{j}^{(2)}$. Notice that the joint distribution of $[\theta_{j}^{(2)}]_{G(2)}$ is given by $W_{k(2),i}^{(2)}$, whose density is proportional to \cite{11}. A similar update procedure is used for updating the components of $\theta_{G_{10}}^{(1)}$ and $\theta_{G_{10}}^{(2)}$. To update the components of $\theta_{G_{11}}^{(1)}$ and $\theta_{G_{11}}^{(2)}$, we draw reflection-coupled MALA proposals in \cite{24}, and then for the acceptance step, $\theta_{G_{11}}^{(1)}$ and $\theta_{G_{11}}^{(2)}$ share the same uniform random variables.
In step 3, to update $k^{(1)}$ and $k^{(2)}$, we first draw two common uniform numbers $w, u \sim \text{Uniform}(0, 1)$ to couple two chains in the Metropolis-Hastings algorithm approach. In the random walk proposal step, if $w \leq 0.5$, for both two chains we propose the left neighbor except at the boundaries. In the acceptance step, for both chains, we accept the proposal if the acceptance probability is greater than $u$, otherwise we reject the proposal.

Although the empirical mixing time estimation method described above only applies to Markov chains of fixed parameters, in practice we have applied it here to the adaptively tuned version that is Algorithm ?? Intuitively, the adaptation won’t introduce any significant bias in the method. However we stress that the theory of the extension of the unbiased MCMC methodology to adaptive chains remains to be developed.

Now we describe the implementation details. We use datasets generated from the data model described in Section 3.1 for each $p \in \{100, 200, \ldots, 5000\}$ with sample size $n = \frac{p}{2}$. We use sample covariance matrices as estimators of $\Sigma_x$, $\Sigma_y$, and $\Sigma_{xy}$ to construct the extended posterior distribution $\bar{\Pi}$ in ?? For simulated tempering we use the set of temperatures $\{1, 1/0.9, 1/0.8, 1/0.7, 1/0.6\}$. We set the lag $L = p$ and the maximum iterations as $10 \times p + 1000$. For each value of $p$, we repeat the simulations 50 times to estimate the distribution of the meeting time $\tau^{(L)}$ of the chain. More precisely, using $\varepsilon = 0.1$, we estimate the mixing time of the chain as the first iteration $t$ for which the Monte Carlo estimate of the right hand side of (13) is less than $\varepsilon$. Fig. 1 below shows the plot of the meeting times and the estimated mixing times (both divided by $p$), as functions of $p$. The results suggest that Algorithm 1 has a mixing time that scales roughly linearly in dimension $p$.

![Figure 1: Boxplots of the distributions of meeting times (divided by $p$) for each value of $p$. The median meeting time (solid line) as well as the estimated meeting time (dashed line) are also shown.](image-url)
We now compare Algorithm 1 with adaptive tuned parameters to Rifle\((k)\) in \([10]\). Rifle\((k)\) is a two-stage algorithm. It first solves a convex relaxation of \((1)\). Then using the solution as the initial value, Rifle\((k)\) iteratively perform a gradient ascent step on \(R\) in \((4)\), and a truncation step by only preserving the top \(k\) entries with setting the remaining entries to 0.

We generate the data using the model in Section 3.1, with sample size \(n = 200\) and dimension \(p = 500\). We use sample covariance matrices as estimators of \(\Sigma_x\), \(\Sigma_y\), and \(\Sigma_{xy}\) as in Section 3.2. For the MCMC sampling we consider two implementations of Algorithm 1: one without simulated tempering (by only using one temperature, i.e. \(k = 1\)), and one with simulated tempering by setting the set of temperatures to be \(\{1, 1/0.9, 1/0.8, 1/0.7, 1/0.6\}\). We run the MCMC sampler without tempering for number of \(N = 2000\) iterations, and \(N = 10000\) iterations for simulated tempering.

In both case, we process the output as in Section 2.4, then obtaining unit vectors \(v^{(t)}_x\) and \(v^{(t)}_y\) as estimators of \(v^*_x\), \(v^*_y\) at iteration \(t \in T\). In the simulated data experiments where the true values of \(v^*_x\), \(v^*_y\) are known, we assess the quality of the inference by computing the posterior mean square errors \(\widehat{\text{mse}}(v_x)\) and \(\widehat{\text{mse}}(v_y)\), where

\[
\widehat{\text{mse}}(v_x) \overset{\text{def}}{=} \frac{1}{|T|} \sum_{t \in T} \text{mse}(v^{(t)}_x), \quad \text{and} \quad \text{mse}(v_x) \overset{\text{def}}{=} \min \left( \|v_x - v^*_x\|_2^2, \|v_x + v^*_x\|_2^2 \right),
\]

(14)

and \(\widehat{\text{mse}}(v_y)\) is computed similarly.

For the comparison we use Rifle\((6)\) (that is the method is provided with the exact number of nonzero components). Because the first stage of the algorithm to obtain an initial estimate is too slow, here we explored two possibilities. In the first experiment, the initial estimate in Rifle\((6)\) is generated from mean zero standard Gaussian noise, the same as our simulated tempering sampler. In the second experiment, we initialize Rifle\((6)\) from the ground truth \(v^*_x\), \(v^*_y\) perturbed with mean zero Gaussian noise with standard deviation 0.2. In such initialization, the elements of the initial value at indices when \(\theta^* \neq 0\) will be greater than \(1/\sqrt{3} - 0.2 \times 1.5 \approx 0.3\) with probability over 93%, and the elements of the initial value at indices when \(\theta^* = 0\) will be less than \(0.2 \times 1.5 = 0.3\) with probability over 93%. In that case, after truncation step, the support of the initial value will be around the true support of \(\theta^*\) with high probability, which assures the goodness of the initial estimate in the second experiment. As with the Bayesian inference we report the mean square errors of Rifle (see Equation (14)). These mean square errors estimates are then averaged over 100 data simulations. The results are shown in Table 1.

We can see from the results that Rifle\((6)\) typically requires a good initialization and performs very poorly otherwise. However, even with good initialization the method still significantly under-perform our Bayesian method. The results also illustrate the advantage of using simulated tempering. Across all the datasets, indeed observed in our experiment, in roughly one out of four initializations, the plain MCMC sampler – without tempering – gets trapped in a local mode. This explains its relatively poor performance.
|          | Rifle(6) with random Init. | Rifle(6) with good Init. | MCMC         | Simulated Temp. |
|----------|----------------------------|--------------------------|--------------|-----------------|
| mse($v_x$) | 1.38 (0.31)                | 0.32 (0.25)              | 0.50 (0.80) | 0.06 (0.28)     |
| mse($v_y$) | 1.40 (0.31)                | 0.31 (0.27)              | 0.48 (0.81) | 0.06 (0.28)     |

Table 1: Mean square errors of Rifle(6) and proposed Bayesian method. The algorithm Rifle(6) is initialized either from random noise (first column), or from the truth plus a small noise (2nd column). The numbers in parenthesis are standard errors obtained from 100 data replications.

### 3.4 Application to mixed data type and comparison with [12]

In this section, we investigate the performance of our method when the data generating process deviates from the multivariate Gaussian generator described in Section 3.1. We consider the truncated latent Gaussian copula model in [12]. The truncated latent Gaussian copula model is useful in modeling truncated and non-Gaussian data, which are commonly encountered in for instance biological sciences. Our proposed framework yields a straightforward Bayesian implementation of the model.

**Definition 1** (Gaussian copula model). A random vector $Z = (Z_1, \ldots, Z_p)^T$ is a realization of the Gaussian copula model, if there exists a transformation $h: \mathbb{R}^p \rightarrow \mathbb{R}^p$ such that $h(Z) = (h_1(Z_1), \ldots, h_p(Z_p))^T \sim \mathcal{N}(0, \Sigma)$ and for each $j = 1, \ldots, p$, transformation $h_j: \mathbb{R} \rightarrow \mathbb{R}$ is monotonically increasing. We write this as $Z \sim \mathcal{NPN}(0, \Sigma, h)$.

**Definition 2** (Truncated Gaussian copula model). A random vector $Z = ((Z^{(1)})^T, (Z^{(2)})^T)^T$, where $Z^{(1)} \in \mathbb{R}^{p_x}$ and $Z^{(2)} \in \mathbb{R}^{p_y}$, is a realization of a latent Gaussian copula model with truncation if there exists a random vector $U \in \mathbb{R}^{p_y}$ such that $(Z^{(1)}, U) \sim \mathcal{NPN}(0, \Sigma, h)$ and $Z^{(2)}_j = I(U_j > C_j)U_j$ for all $j = 1, \ldots, p_y$, where $C = (C_1, \ldots, C_{p_y})$ is a truncation parameter. We write $Z \sim \mathcal{TNPN}(0, \Sigma, h, C)$.

In truncated Gaussian copula model, the observed variable is $Z$ due to data transformation and truncation, but the random vector of interest is the latent variable $h(Z^{(1)}, U) \sim \mathcal{N}(0, \Sigma)$. We denote $h(Z^{(1)}, U)$ by $(X, Y)$, where $X \in \mathbb{R}^{p_x}$, and $Y \in \mathbb{R}^{p_y}$. And therefore the parameter of interest is the principal canonical correlation vectors of $\Sigma$ as defined above. It is shown by [12] that a Kendall’s-tau-based estimator of the covariance matrix $\Sigma$ can be derived from $n$ realizations of $\mathcal{TNPN}(0, \Sigma, h, C)$. We refer the reader to [12] for details. Let $\hat{\Sigma} = \begin{pmatrix} \hat{\Sigma}_x & \hat{\Sigma}_{xy} \\ \hat{\Sigma}_{yx} & \hat{\Sigma}_y \end{pmatrix}$ be the resulting estimator, where $\hat{\Sigma}_x$ and $\hat{\Sigma}_{xy}$ are estimators for covariance matrix of $X$ and $Y$ and $\hat{\Sigma}_y$ is the estimator between $X$ and $Y$. Based on $\hat{\Sigma}_x$, $\hat{\Sigma}_{xy}$ and $\hat{\Sigma}_y$, the target posterior distribution in [6] can then be constructed. We compare our approach to the frequentist estimator proposed by [12] that estimates the principal canonical correlation pair by solving

$$\max_{v_x, v_y} v_x^T \hat{\Sigma}_{xy} v_y - \lambda_1 \|v_x\|_1 - \lambda_2 \|v_y\|_1, \quad \text{s.t.} \quad v_x^T \hat{\Sigma}_x v_x \leq 1, \quad v_y^T \hat{\Sigma}_y v_y \leq 1.$$  \hspace{1cm} (15)

We now introduce the implementation. We set sample size $n = 200$ and dimensions $p_x, p_y = 100$, and generate the dataset from $\mathcal{TNPN}(0, \Sigma, h, C)$, where $\Sigma$ is constructed the same way in Section 3.1. Since as shown in [12], the Kendall’s-tau-based estimator $\hat{\Sigma}$ is invariant to $h$, we set $h$ to be identity function without loss of generality. For simplicity, we only test for $C = c1_{p_y}$, where
\( c = -2, -1, 0 \). The truncation level (percentage of zero elements across the variables) is in the range of 2 – 50\%. We implement Algorithm 1 with the number of iterations \( N = 10000 \) and the set of temperatures \( \{1, 1/0.9, 1/0.8, 1/0.7\} \). The weights \((c_1, \ldots, c_K)\), and step-sizes \((\eta_1, \ldots, \eta_K)\) are adaptively tuned by Algorithm 2. We process the output as in Section 2.4, then obtaining unit vectors \( v^{(t)}_x \) and \( v^{(t)}_y \) as estimators of \( v_x^* \), \( v_y^* \) at iteration \( t \in T \). We evaluate the quality of the inference by computing the posterior mean square errors \( \text{mse}(v_x) \) and \( \text{mse}(v_y) \) as in (14). We also assess the variable selection performance at iteration \( t \in T \) by computing true-positive rate (TPR) and true-negative rate (TNR) defined as

\[
\text{TPR}(v^{(t)}_x) \overset{\text{def}}{=} \frac{\#\{ j : (v^{(t)}_x)_j \neq 0, (v_x^*)_j \neq 0 \}}{\#\{ j : (v_x^*)_j \neq 0 \}}, \quad \text{TNR}(v^{(t)}_x) \overset{\text{def}}{=} \frac{\#\{ j : (v^{(t)}_x)_j = 0, (v_x^*)_j = 0 \}}{\#\{ j : (v_x^*)_j = 0 \}}, \quad (16)
\]

and \( \text{TPR}(v^{(t)}_y) \), \( \text{TNR}(v^{(t)}_y) \) is computed similarly. We assess the algorithm performance by averaging TPR and TNR over \( t \in T \), denoted \( \overline{\text{TPR}} \) and \( \overline{\text{TNR}} \).

For the comparison we compute the estimator (15) using the algorithm mixeddcca in the the R package mixedCCA of [12], using BIC1 and BIC2 criteria respectively for selecting the regularization parameters \( \lambda_1 \) and \( \lambda_2 \). Mean (and standard deviation) of \( \text{mse} \), \( \overline{\text{TPR}} \) and \( \overline{\text{TNR}} \) of proposed Bayesian method and \( \text{mse} \), \( \overline{\text{TPR}} \) and \( \overline{\text{TNR}} \) of mixedCCA over 50 datasets is presented in Table 2. We can see from Table 2a that overall our approach performs better than mixeddcca in terms of accuracy, in particular with smaller standard deviation. We also note that the mean square error is consistently better for estimating \( v_x \), which is reasonable, since only \( Z_2 \) is truncated. In the meantime, the mean square error increases with \( c \), which is also consistent with the data generating process.

4 Principal canonical correlation of clinical and proteomic data in covid-19 patients

COVID-19 is an infectious disease that is rapidly sweeping through the world, with more than 18.9 million cases and 712,000 deaths as of this writing. The disease is caused by a severe acute respiratory syndrome coronavirus (SARS-CoV-2). There is currently an intense global effort to better understand the virus and find cures and vaccines. We use our methodology to re-analysis a data set produced by [27] to identify biomarkers for early detection of severely ill Covid-19 patients. To correlate proteins and clinics data, we choose 66 samples from training data and 20 samples from testing data. To that end, the study enrolled 86 patients (some non-Covid-19 patients, and among the Covid-19 patients, some that developed mild symptoms, and some that became severely ill) (To correlate proteins and clinics data, we choose 66 samples from training data and 20 samples from testing data). The exact protocol for recruiting those patients is not clearly explained. For each patient they measured three (3) physical characteristics (sex, age, and body mass index), twelve (12) clinical variables as routinely measured from blood samples (white blood cells count, lymphocytes count, C-reactive protein, etc...). Furthermore, the serum of each patient is analyzed by liquid mass spectrometry-based proteomics to quantify their proteome and

\[ \text{mse} \]
Table 2: Mean (and standard deviation) of $\hat{\text{mse}}$, $\hat{\text{TPR}}$ and $\hat{\text{TNR}}$ of proposed Bayesian method and $\text{mse}$, $\text{TPR}$ and $\text{TNR}$ of mixedCCA for different truncation values $c$ when $h$ is identity function.
metabolome. In [27], the data is used to build a statistical model to predict whether or not a Covid-19 patient will progress to a severe state of illness. The dataset of [27] is freely available from the journal website.

We use canonical correlation analysis to re-analyze the data. A common working assumption is that SARS-CoV-2 induces patterns of molecular changes that can be detected in the sera of patients. Canonical correlation analysis can help identify these patterns. To do this we focus on the proteomic data, and we estimate the principal sparse canonical correlation between the physical and clinical variables on one hand and the proteomic variables on the other. See for instance [28] for a similar analysis on tuberculosis and malaria.

We pre-process the data by removing all the proteins for which 50% or more values are missing, leading to a total of $p_y = 513$ proteins, and $p_x = 15$ clinical and physical variables. The sample size is $n = 86$ patients. Liquid mass spectrometry-based proteomics typically produces a large quantity of missing values ([29, 30]). We make the assumption here that the missing values are driven mainly by detection limit truncation ([29]), which makes the truncated latent Gaussian copula model described above appropriate for this example. We apply simulated tempering with 100000 iterations and the set of temperatures $\{1, 1/0.9, 1/0.8, 1/0.7\}$.

We found that the posterior distribution of $\nu_{x^*}$ has only one component (corresponding to C-reactive protein) with estimated inclusion probability of $\Pi(\delta_j = 1|Z) = 0.99$. All other physical and clinical variables have probabilities smaller than 0.1. We found also that the principal canonical vector of the proteomic data is also driven by a single protein (P02763, also known as Alpha-1-acid glycoprotein 1 or AGP 1), with estimated inclusion probability of $\Pi(\delta_j = 1|Z) = 0.89$. All other proteins have inclusion probability smaller than 0.1. Fig. 2 shows the boxplot of the posterior distribution of the estimated correlation $\hat{\rho}$ between the two data set, as well as the box-plot and autocorrelation function of the MCMC output of the coefficients of CRP and AGP 1 in the posterior distribution. The fast decay of the autocorrelation functions show a good mixing of the MCMC sampler.

The highly sparse nature of the estimated canonical correlation vectors is striking and very interesting. Several studies have observed the predictive power of C-reactive protein (CRP) in the progression of Covid-19 into a severe illness (see for instance [31] for a meta-analysis). This suggests that the correlation detected in our analysis between the two datasets is indeed driven by the progression of Covid-19 into a severe illness. Therefore, our analysis suggests that protein AGP 1 may also be playing an important role in the progression of Covid-19 into a severe illness.

We learn from Uniprot\textsuperscript{2} that AGP 1 functions as transport protein in the blood stream, and appears to function in modulating the activity of the immune system during the acute-phase reaction. Furthermore, AGP 1 appears on the list of differentially expressed proteins in the sera of severely ill Covid-19 patients designed by [27], and also appeared in the literature as playing a role in the immune system’s response to malaria ([32]). All together we are fairly confidence that our statistical analysis has picked up a genuine signal in how Covid-19 interacts with our immune system. However, more biomedical research is obviously needed to fully understanding its meaning.

\textsuperscript{2}https://www.uniprot.org
Figure 2: Boxplot and autocorrelation plots from MCMC output.

Figure 3: Distribution of CRP and AGP by group of patients.

# 5 Conclusion

We have developed in this work a flexible semiparametric Bayesian inference for sparse canonical correlation analysis. Although not addressed here the method can be further extended to capture more than one canonical correlation vector, either by deflation, or by reformulating the problem as a higher dimensional canonical correlation analysis estimation problem as in [10]. Another aspect of the methodology where more research would be beneficial is the theoretical analysis of the statistical properties of the proposed posterior distribution. Understanding the conditions under which posterior contraction, and the Bernstein-von Mises phenomenon hold will provide further guarantees to practitioners.

## References

[1] H Hotelling. Relations between two sets of variates. *Biometrika*, 1936.
[2] Daniela M Witten and Robert J Tibshirani. Extensions of sparse canonical correlation analysis with applications to genomic data. *Statistical applications in genetics and molecular biology*, 8(1):1–27, 2009.

[3] Qianxing Mo, Ronglai Shen, Cui Guo, Marina Vanucci, Keith S Chan, and Susan G Hilsenbeck. A fully bayesian latent variable model for integrative clustering analysis of multi-type omics data. *Biostatistics*, 19(1):71–86, 2017.

[4] Nimrod Rappoport and Ron Shamir. Multi-omic and multi-view clustering algorithms: review and cancer benchmark. *Nucleic acids research*, 46(20):10546–10562, 2018.

[5] Francis Bach and Michael Jordan. A probabilistic interpretation of canonical correlation analysis. 2005.

[6] Arto Klami, Seppo Virtanen, and Samuel Kaski. Bayesian canonical correlation analysis. *Journal of Machine Learning Research*, 14(Apr):965–1003, 2013.

[7] Eun Jeong Min, Changgee Chang, and Qi Long. Generalized bayesian factor analysis for integrative clustering with applications to multi-omics data. In *2018 IEEE 5th International Conference on Data Science and Advanced Analytics (DSAA)*, pages 109–119. IEEE, 2018.

[8] Daniela Witten, Robert Tibshirani, and Trevor Hastie. A penalized matrix decomposition, with applications to sparse principal components and canonical correlation analysis. *Biostatistics (Oxford, England)*, 10:515–34, 05 2009.

[9] Chao Gao, Zongming Ma, Zhao Ren, and Harrison H. Zhou. Minimax estimation in sparse canonical correlation analysis. *Ann. Statist.*, 43(5):2168–2197, 10 2015.

[10] Kean Ming Tan, Zhaoran Wang, Han Liu, and Tong Zhang. Sparse generalized eigenvalue problem: Optimal statistical rates via truncated rayleigh flow. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 80(5):1057–1086, 2018.

[11] Hemant Ishwaran and J. Sunil Rao. Spike and slab variable selection: Frequentist and bayesian strategies. *Ann. Statist.*, 33(2):730–773, 04 2005.

[12] Grace Yoon, Raymond J. Carroll, and Irina Gaynanova. Sparse semiparametric canonical correlation analysis for data of mixed types. *arXiv: Methodology*, 2018.

[13] Charles J. Geyer and Elizabeth A. Thompson. Annealing markov chain monte carlo with applications to ancestral inference. *Journal of the American Statistical Association*, 90(431):909–920, 1995.

[14] Jun S Liu. *Monte Carlo strategies in scientific computing*. Springer Science & Business Media, 2008.

[15] Edward I. George and Robert E. McCulloch. Approaches for bayesian variable selection. *Statistica Sinica*, 7(2):339–373, 1997.
[16] P. G. Bissiri, C. C. Holmes, and S. G. Walker. A general framework for updating belief distributions. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 78(5):1103–1130, 2016.

[17] Yves A. Atchade. On the contraction properties of some high-dimensional quasi-posterior distributions. *Ann. Statist.*, 45(5):2248–2273, 10 2017.

[18] Benjamin A. Shaby. The open-faced sandwich adjustment for mcmc using estimating functions. *Journal of Computational and Graphical Statistics*, 23(3):853–876, 10 2014.

[19] Gareth O. Roberts and Richard L. Tweedie. Exponential convergence of langevin distributions and their discrete approximations. *Bernoulli*, 2(4):341–363, 1996.

[20] Yves Atchade, Gareth Roberts, and Jeffrey Rosenthal. Towards optimal scaling of metropolis-coupled markov chain monte carlo. *Statistics and Computing*, 21:555–568, 10 2011.

[21] Btażej Miasojedow, Eric Moulines, and Matti Vihola. An adaptive parallel tempering algorithm. *Journal of Computational and Graphical Statistics*, 22(3):649–664, 2013.

[22] Christophe Andrieu and Johannes Thoms. A tutorial on adaptive mcmc. *Statistics and Computing*, 18(4):343–373, 2008.

[23] Yves Atchadé and Jeffrey Rosenthal. On adaptive markov chain monte carlo algorithms. *Bernoulli*, 11, 10 2005.

[24] Yves F. Atchadé and Jun S. Liu. The wang-landau algorithm in general state spaces: Applications and convergence analysis. *Statistica Sinica*, 20(1):209–233, 2010.

[25] Niloy Biswas, Pierre E. Jacob, and Paul Vanetti. Estimating convergence of markov chains with l-lag couplings, 2019.

[26] Pierre Jacob, John O’Leary, and Yves Atchadé. Unbiased markov chain monte carlo with couplings. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 08 2017.

[27] Bo Shen, Xiao Yi, Yaoting Sun, Xiaojie Bi, Juping Du, Chao Zhang, Sheng Quan, Fangfei Zhang, Rui Sun, Lijia Qian, Weigang Ge, Wei Liu, Shuang Liang, Hao Chen, Ying Zhang, Jun Li, Jiaqin Xu, Zebao He, Baofu Chen, Jing Wang, Haixi Yan, Yufen Zheng, Donglian Wang, Jiasheng Zhu, Ziqing Kong, Zhuyang Kang, Xiao Liang, Xuan Ding, Guan Ruan, Nan Xiang, Xue Cai, Huanhuan Gao, Lu Li, Sainan Li, Qi Xiao, Tian Lu, Yi Zhu, Huafen Liu, Haixiao Chen, and Tiannan Guo. Proteomic and metabolomic characterization of covid-19 patient sera. *Cell*, 182(1):59 – 72.e15, 2020.

[28] Juho Rousu, Daniel D. Agranoff, Olugbemiro Sodeinde, John Shawe-Taylor, and Delmiro Fernandez-Reyes. Biomarker discovery by sparse canonical correlation analysis of complex clinical phenotypes of tuberculosis and malaria. *PLOS Computational Biology*, 9(4):1–10, 04 2013.
[29] Yuliya V. Karpievitch, Ashoka D. Polpitiya, Gordon A. Anderson, Richard D. Smith, and Alan R. Dabney. Liquid chromatography mass spectrometry-based proteomics: Biological and technological aspects. *Ann. Appl. Stat.*, 4(4):1797–1823, 12 2010.

[30] Jonathon J. O’Brien, Harsha P. Gunawardena, Joao A. Paulo, Xian Chen, Joseph G. Ibrahim, Steven P. Gygi, and Bahjat F. Qaqish. The effects of nonignorable missing data on label-free mass spectrometry proteomics experiments. *Ann. Appl. Stat.*, 12(4):2075–2095, 12 2018.

[31] Bikash R. Sahu, Raj Kishor Kampa, Archana Padhi, and Aditya K. Panda. C-reactive protein: A promising biomarker for poor prognosis in covid-19 infection. *Clinica Chimica Acta*, 509:91 – 94, 2020.

[32] M J Friedman. Control of malaria virulence by alpha 1-acid glycoprotein (orosomucoid), an acute-phase (inflammatory) reactant. *Proceedings of the National Academy of Sciences*, 80(17):5421–5424, 1983.
Algorithm 2: Adaptive version of simulated tempering for Canonical correlation analysis

**Model Input:** Matrices $\hat{A}, \hat{B}$, $\sigma^2 = 1$, prior parameters $\rho_0, \rho_1, q$.

**MCMC Input:** Number of iterations $N$, Batch size $J$, temperatures $t_1 < \ldots < t_K$.

**Adaptive MCMC Input:** $a = 10$ and $w \in (0, 1)$.

**MCMC Initialization:** Set $k^{(0)} = 1$. Draw $\delta_j^{(0)} \sim \text{Ber}(0.5), \forall j = 1, \ldots, p$, and independently $\theta^{(0)} \sim N(0, I_p)$.

**Adaptation Parameters Initialization:** Set $\tau^{(0)} = 0 \in \mathbb{R}^K$, $v^{(0)} = (0, \ldots, 0) \in \mathbb{R}^K$, and choose $c^{(0)} \in (0, \infty)^K$.

**for** $t = 1$ to $N - 1$, given $(k^{(t)}, \delta^{(t)}, \theta^{(t)}) = (k, \delta, \theta)$, $\tau^{(t)} = \tau$, $c^{(t)} = c$, and $v^{(t)} = v$ **do**

1. **Update $\delta$:** Uniformly randomly select a subset $J$ from $\{1, \ldots, p\}$ of size $J$ without replacement, and draw $\bar{\delta} \sim Q^{(J)}_{\bar{\delta}}(\delta, \cdot)$, where the transition kernel described in (10).

2. **Update $\theta$ and $\tau$:** Draw the components of $[\bar{\delta}]_\theta$ independently from $N(0, \rho_0^{-1} t_k)$. Draw $[\bar{\delta}]_\theta \sim P_{\eta, k, \delta}(\theta, \cdot)$, where $\eta = e^{c_\theta}$ and $P_{\eta, k, \delta}$ denotes the transition kernel of the MALA with step-size $\eta$ and invariant distribution given by $W_{k, \delta}$, whose density is proportional to (11). Denote the acceptance probability of the MALA update. Set

$$\bar{\tau}_k = \tau_k + v_k^{-0.6}(\alpha - 0.3),$$

and for $i \neq k$, set $\bar{\tau}_i = \tau_i$.

3. **Update $k$, $c$ and $v$:** Draw $\bar{k} \sim T_{\bar{\delta}, \bar{\theta}}(k, \cdot)$, where $T_{\bar{\delta}, \bar{\theta}}$ is the transition kernel of the Metropolis-Hastings on $\{1, \ldots, K\}$ with invariant distribution given by (12) and random walk proposal that has reflection at the boundaries. We then set

$$\bar{c}_k = c_k e^a, \quad \bar{v}_k = v_k + 1,$$

and for $i \neq \bar{k}, \bar{c}_i = c_i$, and $\bar{v}_i = v_i$.

4. **Update $a$ and $v$:** If $\|\bar{v}/(\sum_{k=1}^K \bar{v}_k) - 1/K\|_\infty \leq w/K$, then set $a = a/2, \bar{v} = 0 \in \mathbb{R}^K$.

5. **New MCMC state:** Set $(\delta^{(t+1)}, \theta^{(t+1)}, k^{(t+1)}) = (\bar{\delta}, \bar{\theta}, \bar{k}), \tau^{(t+1)} = \bar{\tau}, c^{(t+1)} = \bar{c}$, and $v^{(t+1)} = \bar{v}$.

**end for**

**Output:** $\{(\delta^{(t)}, \theta^{(t)}, k^{(t)}): 0 \leq t \leq N \text{ s.t. } k^{(t)} = 1\}$