Supplementary material: Non-stationary Gaussian process discriminant analysis with variable selection for high-dimensional functional data

Weichang Yu

Melbourne Centre for Data Science, University of Melbourne
and
Sara Wade
School of Mathematics, University of Edinburgh
and
Howard D. Bondell
School of Mathematics and Statistics, University of Melbourne
and
Lamiae Azizi
School of Mathematics and Statistics, University of Sydney

September 29, 2021

S1 Model details and illustration (Section 2)

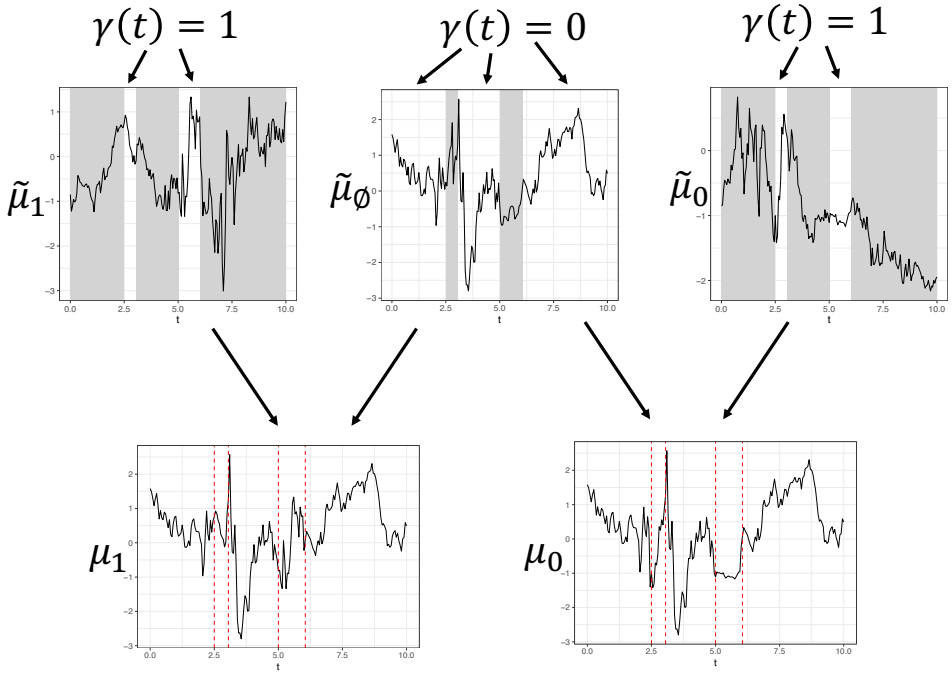


Fig. S1: Illustration of the binary variable selection process.

S1.1 Tridiagonal precision matrices following discretization

Assuming the locations on the grid are equally-spaced, i.e., $t_j - t_{j-1} = \delta$ (for notational simplicity), the precision matrix of $\boldsymbol{z}_i$ induced by the Euler-Mayurama discretization is

$$Q_{NS;\tau,\boldsymbol{\nu}_i} = \frac{1}{\tau}\begin{bmatrix} 1+b_{i11}^2 & b_{i11}b_{i01} & 0 & \dots & 0 \\ b_{i11}b_{i01} & b_{i01}^2+b_{i12}^2 & b_{i12}b_{i02} & \dots & 0 \\ 0 & b_{i12}b_{i02} & b_{i02}^2+b_{i13}^2 & \dots & 0 \\ \vdots & & \ddots & \ddots & \vdots \\ 0 & \dots & 0 & b_{i1,T-1}b_{i0,T-1} & b_{i0,T-1}^2 \end{bmatrix},$$

with $b_{i0j} = \sqrt{\exp(\nu_{ij})/2\delta}$, $b_{i1j} = -\sqrt{\exp(\nu_{ij})/2\delta}\,\{1-\delta/\exp(\nu_{ij})\}$, $\boldsymbol{\nu}_i = (\nu_{i1},\dots,\nu_{iT})$, and $\nu_{ij} = \nu_i(t_j)$. Similarly, the precision matrix of $\boldsymbol{R}$ induced by the Euler-Mayurama discretization is

$$Q_{S;\tau_2,\lambda} = \frac{1}{\tau_2}\begin{bmatrix} 1+c_1^2 & c_0c_1 & 0 & \dots & 0 \\ c_0c_1 & c_0^2+c_1^2 & c_0c_1 & \dots & 0 \\ 0 & c_0c_1 & c_0^2+c_1^2 & \dots & 0 \\ \vdots & & \ddots & \ddots & \vdots \\ 0 & \dots & 0 & c_0c_1 & c_0^2 \end{bmatrix},$$

where $c_0 = \sqrt{\lambda/2\delta}$; $c_1 = -\sqrt{\lambda/2\delta}(1-\delta/\lambda)$.

Algorithm S1 Posterior inference

Initialize variational parameters for $\widetilde{\boldsymbol{\sigma}}_1^2$, $\widetilde{\boldsymbol{\sigma}}_0^2$, $\widetilde{\boldsymbol{\sigma}}_\emptyset^2$, $\{\boldsymbol{z}_i\}_{i=1}^n$, $\boldsymbol{\gamma}$, τ, and $\boldsymbol{R}$.
Initialize MAP estimates of $\{\zeta_i\}_{i=1}^n$, α, and β.
while $\|\boldsymbol{\psi}_1^{(t)} - \boldsymbol{\psi}_1^{(t-1)}\| < \kappa_1$, **do**
1: CAVI update for $\widetilde{\boldsymbol{\mu}}_1$, $\widetilde{\boldsymbol{\mu}}_0$, and $\widetilde{\boldsymbol{\mu}}_\emptyset$.
2: CAVI update for $\widetilde{\tau}_1$, $\widetilde{\tau}_0$, and $\widetilde{\tau}_\emptyset$.
3: SVB update for $\widetilde{\boldsymbol{\nu}}_1$, $\widetilde{\boldsymbol{\nu}}_0$, and $\widetilde{\boldsymbol{\nu}}_\emptyset$.
4: Compute MAP estimate for $\widetilde{\lambda}$ and $\widetilde{\eta}$.
5: CAVI update for $\{\boldsymbol{z}_i\}_{i=1}^n$.
6: CAVI update for τ.
7: SVB update for $\boldsymbol{R}$.
8: Compute MAP estimate for λ and τ_2.
9: Compute MAP estimate for $\{\zeta_i\}_{i=1}^n$.
10: CAVI update for $\widetilde{\boldsymbol{\sigma}}_1^2$, $\widetilde{\boldsymbol{\sigma}}_0^2$, and $\widetilde{\boldsymbol{\sigma}}_\emptyset^2$.
11: CAVI update for $\boldsymbol{\gamma}$.
12: Compute MAP estimate for α and β.
13: Evaluate stop criterion.
Store converged values from estimation phase.

Algorithm S2 Classification

Initialize variational parameters for $\{\boldsymbol{z}_{n+i}\}_{i=1}^m$ and $\{y_{n+i}\}_{i=1}^m$.
Initialize MAP estimates of $\{\zeta_{n+i}\}_{i=1}^m$.
while $\|\boldsymbol{\psi}_2^{(t)} - \boldsymbol{\psi}_2^{(t-1)}\| < \kappa_2$, **do**
1: CAVI update for $\{\boldsymbol{z}_{n+i}\}_{i=1}^m$.
2: Compute MAP estimates for $\{\zeta_{n+i}\}_{i=1}^m$.
3: CAVI update for $\{y_{n+i}\}_{i=1}^m$.
4: Evaluate stop criterion.

S1.2 Weakly informative priors

For the non-location-varying length scale parameters, e.g. λ, we assign the hyperprior $\lambda \sim \text{LogN}(\mu_\lambda, \sigma_\lambda^2)$. Here, the hyperparameters $\mu_\lambda, \sigma_\lambda^2$ are specified such that the resultant prior is weakly informative (Betancourt, 2017). This mitigates potential non-identifiability issues that arise in the estimation of GP length scales. More specifically, GP length scales are non-identifiable beyond the feasible range (the range of pairwise distances between observed locations). The weakly informative prior circumvents this problem by ensuring that most of the prior mass concentrates on this range. In our context, we specify μ_λ and σ_λ^2 such that 99% of the prior mass is on the observed range of pairwise location distances.

S2 Details on posterior inference (Section 3)

A pseudocode description of the posterior inference and classification phases are provided in Algorithm S1 and Algorithm S2, respectively. Further details supporting specific steps are provided in the following subsections.

S2.1 Step 2 of Algorithm S1: CAVI update for $\widetilde{\tau}_k$

The product component for $\widetilde{\tau}_k$ is:

$$q(\widetilde{\tau}_k) = \text{InvGa}(a_{\widetilde{\tau}_k}, b_{\widetilde{\tau}_k}),$$

where $a_{\widetilde{\tau}_k} = A_{\widetilde{\tau}} + T/2$ and $b_{\widetilde{\tau}_k} = B_{\widetilde{\tau}} + \text{tr}[\mathbb{E}(\widetilde{\boldsymbol{\mu}})k\widetilde{\boldsymbol{\mu}}_k^\top)\mathbb{E}C_{NS;\widetilde{\boldsymbol{\nu}}_k}]/2$, where $Q_{NS;\widetilde{\tau}_k,\widetilde{\boldsymbol{\nu}}_k} = 1/\widetilde{\tau}_k C_{NS;\widetilde{\boldsymbol{\nu}}_k}$. While a naive calculation of $b_{\widetilde{\tau}_k}$ would be of order $\mathcal{O}(T^2)$, this can be reduced to $\mathcal{O}(T)$:

$$\begin{aligned}
&\text{tr}[\mathbb{E}(\widetilde{\boldsymbol{\mu}}_k\widetilde{\boldsymbol{\mu}}_k^\top)\mathbb{E}C_{NS;\widetilde{\boldsymbol{\nu}}_k}] \\
&\quad = \mathbb{E}(\widetilde{\mu}_{k1}^2) + \sum_{j=1}^{T-1}\left\{\mathbb{E}(\widetilde{b}_{k1j}^2)\mathbb{E}(\widetilde{\mu}_{kj}^2) + 2\mathbb{E}(\widetilde{b}_{k0j}\widetilde{b}_{k1j})\mathbb{E}(\widetilde{\mu}_{kj}\widetilde{\mu}_{kj+1}) + \mathbb{E}(\widetilde{b}_{k0j}^2)\mathbb{E}(\widetilde{\mu}_{kj+1}^2)\right\},
\end{aligned}$$

with $\widetilde{b}_{k0j} = \sqrt{\exp(\widetilde{\nu}_{kj})/2\delta}$ and $\widetilde{b}_{k1j} = -\sqrt{\exp(\widetilde{\nu}_{kj})/2\delta}\,\{1 - \delta/\exp(\widetilde{\nu}_{kj})\}$.

S2.2 Step 3 of Algorithm S1: SVB for location-varying log length scales

In this section, we illustrate the use of stochastic variational Bayes for updating the variational parameters of $\widetilde{\boldsymbol{\nu}}_k$. Recall that the VB posterior for $\widetilde{\boldsymbol{\nu}}_k$ is of the form $q(\widetilde{\boldsymbol{\nu}}_k) = \text{N}(m_{\widetilde{\boldsymbol{\nu}}_k}, \Omega_{\widetilde{\boldsymbol{\nu}}_k}^{-\top}\Omega_{\widetilde{\boldsymbol{\nu}}_k})$, where a sparse structure is specified for $\Omega_{\widetilde{\boldsymbol{\nu}}_k}$ to enhance computational efficiency (refer to Section S3 for further details). By writing $\widetilde{\boldsymbol{\nu}}_k = m_{\widetilde{\boldsymbol{\nu}}_k} + \Omega_{\widetilde{\boldsymbol{\nu}}_k}^{-\top}\boldsymbol{v}$ where $\boldsymbol{v} \sim \text{N}_T(\boldsymbol{0}, \boldsymbol{I})$, we may express the ELBO as

$$\begin{aligned}
&\text{ELBO}(m_{\widetilde{\boldsymbol{\nu}}_k}, \Omega_{\widetilde{\boldsymbol{\nu}}_k}), \\
&= \mathbb{E}\log p(\widetilde{\boldsymbol{\mu}}_k, \widetilde{\boldsymbol{\nu}}_k) - \mathbb{E}\log\phi(\widetilde{\boldsymbol{\nu}}_k; m_{\widetilde{\boldsymbol{\nu}}_k}, \Omega_{\widetilde{\boldsymbol{\nu}}_k}^{-\top}\Omega_{\widetilde{\boldsymbol{\nu}}_k}^{-1}), \\
&= \mathbb{E}\log p(\widetilde{\boldsymbol{\mu}}_k, m_{\widetilde{\boldsymbol{\nu}}_k} + \Omega_{\widetilde{\boldsymbol{\nu}}_k}^{-\top}\boldsymbol{v}) - \log|\Omega_{\widetilde{\boldsymbol{\nu}}_k}| + \text{Constant},
\end{aligned}$$

where

$$\begin{aligned}
&\mathbb{E}\left[\log p(\widetilde{\boldsymbol{\mu}}_k \,|\, \widetilde{\boldsymbol{\nu}}_k)\right] = \tfrac{1}{2}\mathbf{1}^\top\mathbb{E}_q(\widetilde{\boldsymbol{\nu}}_k) - \tfrac{1}{2}m_{1/\tau}\text{tr}[\mathbb{E}(\widetilde{\boldsymbol{\mu}}_k\widetilde{\boldsymbol{\mu}}_k^\top)\mathbb{E}C_{NS;\widetilde{\boldsymbol{\nu}}_k}] \\
&= \tfrac{1}{2}\mathbf{1}^\top\mathbb{E}_q(\widetilde{\boldsymbol{\nu}}_k) - \tfrac{1}{2}m_{1/\tau}\sum_{j=1}^{T-1}\left\{\mathbb{E}(\widetilde{b}_{k1j}^2)\mathbb{E}(\widetilde{\mu}_{kj}^2) + 2\mathbb{E}(\widetilde{b}_{k0j}\widetilde{b}_{k1j})\mathbb{E}(\widetilde{\mu}_{kj}\widetilde{\mu}_{kj+1}) + \mathbb{E}(\widetilde{b}_{k0j}^2)\mathbb{E}(\widetilde{\mu}_{kj+1}^2)\right\}.
\end{aligned}$$

The gradient of ELBO with respect to $m_{\widetilde{\boldsymbol{\nu}}_k}$ and $\Omega_{\widetilde{\boldsymbol{\nu}}_k}$ are

$$\nabla_{m_{\widetilde{\boldsymbol{\nu}}_k}}\text{ELBO} = \mathbb{E}_{\boldsymbol{v}}\left[\nabla_{\widetilde{\boldsymbol{\nu}}_k}\mathbb{E}_{q(\widetilde{\boldsymbol{\mu}}_k)}\log p(\widetilde{\boldsymbol{\mu}}_k, m_{\widetilde{\boldsymbol{\nu}}_k} + \Omega_{\widetilde{\boldsymbol{\nu}}_k}^{-\top}\boldsymbol{v})\right]$$

and

$$\nabla_{\Omega_{\widetilde{\boldsymbol{\nu}}_k}}\text{ELBO} = -\mathbb{E}_{\boldsymbol{v}}\left[\Omega_{\widetilde{\boldsymbol{\nu}}_k}^{-\top}\boldsymbol{v}\nabla_{\widetilde{\boldsymbol{\nu}}_k}\mathbb{E}_{q(\widetilde{\boldsymbol{\mu}}_k)}\log p(\widetilde{\boldsymbol{\mu}}_k, m_{\widetilde{\boldsymbol{\nu}}_k} + \Omega_{\widetilde{\boldsymbol{\nu}}_k}^{-\top}\boldsymbol{v})^\top\Omega_{\widetilde{\boldsymbol{\nu}}_k}^{-\top}\right] - \text{dg}\left(\{1/\Omega_{\widetilde{\boldsymbol{\nu}}_k;jj}\}_{j=1}^T\right).$$

Hence, each update of the variational scheme involves running Algorithm S3 until convergence of $\text{ELBO}(m_{\widetilde{\boldsymbol{\nu}}_k}, \Omega_{\widetilde{\boldsymbol{\nu}}_k})$. Note that the learning rate ρ_l may be specified with any suitable scheme such as ADAM, ADAGRAD or ADADELTA.

Algorithm S3 SVB update algorithm for $\widetilde{\boldsymbol{\nu}}_k$.

Initialize $m^{(0)}_{\widetilde{\boldsymbol{\nu}}_k} = \mathbf{0}$ and $\Omega^{(0)}_{\widetilde{\boldsymbol{\nu}}_k} = \boldsymbol{I}$. Specify learning rate $\{\rho_l\}$.
At iteration l,

1: Generate $\boldsymbol{v}^{(l)} \sim \mathrm{N}(\mathbf{0}, \boldsymbol{I})$.
2: Compute $\widetilde{\boldsymbol{\nu}}_k^{(l)} = m^{(l-1)}_{\widetilde{\boldsymbol{\nu}}_k} + \Omega^{(l-1)\,-\top}_{\widetilde{\boldsymbol{\nu}}_k}\boldsymbol{v}^{(l)}$.
3: Compute $g_m^{(l)} = \nabla_{\widetilde{\boldsymbol{\nu}}_k}\mathbb{E}\log p(\widetilde{\boldsymbol{\mu}}_k, m^{(l-1)}_{\widetilde{\boldsymbol{\nu}}_k} + \Omega^{(l-1)\,-\top}_{\widetilde{\boldsymbol{\nu}}_k}\boldsymbol{v}) + \Omega^{(l-1)\,\top}_{\widetilde{\boldsymbol{\nu}}_k}\boldsymbol{v}^{(l)}$.
4: Update $m^{(l)}_{\widetilde{\boldsymbol{\nu}}_k} = m^{(l-1)}_{\widetilde{\boldsymbol{\nu}}_k} + \rho_l g_m^{(l)}$.
5: Compute $g^{(l)}_{\Omega'} = -\Omega^{(l-1)\,-\top}_{\widetilde{\boldsymbol{\nu}}_k}\boldsymbol{v}^{(l)}(\Omega^{(l-1)\,-1}_{\widetilde{\boldsymbol{\nu}}_k} g_m^{(l)})^\top$.
6: Replace diagonal entries of $g^{(l)}_{\Omega'}$ with $g^{(l)}_{\Omega'} \odot \mathrm{diag}(\Omega^{(l-1)}_{\widetilde{\boldsymbol{\nu}}_k})$.
7: Update $\Omega^{(l)}_{\widetilde{\boldsymbol{\nu}}_k} = \Omega^{(l-1)}_{\widetilde{\boldsymbol{\nu}}_k} + \rho_l g^{(l)}_{\Omega'}$.
8: Replace diagonal entries of $\Omega^{(l)}_{\widetilde{\boldsymbol{\nu}}_k}$ with $\exp\{\mathrm{diag}(\Omega^{(l)}_{\widetilde{\boldsymbol{\nu}}_k})\}$.

S2.3 Step 6 of Algorithm S1: CAVI update for τ

Similar to update of $\widetilde{\tau}_k$, for the magnitude τ of the latent processes, $q(\tau) = \mathrm{InvGa}(r_\tau, s_\tau)$ where, $r_\tau = A_\tau + nT/2$ and

$$s_\tau = b_\tau + \tfrac{1}{2}\left[\sum_{i=1}^{n} \mathrm{tr}\left(\mathbb{E}[\boldsymbol{z}_i\boldsymbol{z}_i^\top]\mathbb{E}[C_{NS;\boldsymbol{\nu}_i}]\right)\right],$$

with $Q_{NS;\tau,\boldsymbol{\nu}_i} = 1/\tau C_{NS;\boldsymbol{\nu}_i}$. Again, this last term can be computed efficiently thanks to the tridiagonal structure of the precision matrix:

$$\mathrm{tr}\left(\mathbb{E}[\boldsymbol{z}_i\boldsymbol{z}_i^\top]\mathbb{E}[C_{NS;\boldsymbol{\nu}_i}]\right) = \mathbb{E}(z_{i1}^2) + \sum_{t=1}^{T-1}\left\{\mathbb{E}(b_{i1t}^2)\mathbb{E}(z_{it}^2) + 2\mathbb{E}(b_{i0t}b_{i1t})\mathbb{E}(z_{it}z_{it+1}) + \mathbb{E}(b_{i0t}^2)\mathbb{E}(z_{it+1}^2)\right\}.$$

S2.4 Step 12 of Algorithm S1: MAP for Ising parameters

The hyperparameters α and β controlling sparsity and correlation of the Ising prior are updated as the maximizer of the objective:

$$\begin{aligned}\mathrm{OBJ}(\alpha,\beta) &= \mathbb{E}\left[\log p(\boldsymbol{\gamma}\,|\,\alpha,\beta) + \log p(\alpha) + \log p(\beta)\right],\\ &= \mathrm{Constant} - \alpha\sum_{t=1}^{T} w_t + \beta\sum_{t=1}^{T-1} w_t w_{t+1} + (K-L)T - \log(\aleph_1^p + \aleph_2^p) + \log p(\alpha) + \log p(\beta),\end{aligned}$$

where $K = \beta/4$, $L = (\beta-\alpha)/2$, $\aleph_{1,2} = e^K\cosh(L) \pm \sqrt{e^{2K}\cosh^2(L) - 2\sinh(2K)}$, $p(\alpha) = \mathrm{N}(\mu_\alpha, \sigma_\alpha^2)$ and $p(\beta) = \mathrm{LogN}(\mu_\beta, \sigma_\beta^2)$. Note we have used results from Salinas (2001) to obtain a closed form expression for the partition function in computations above.

S2.5 Convergence criterion

Step 13 of Algorithm S1. For the posterior inference phase, let $\boldsymbol{\psi}_1^{(t)}$ denote a vector containing all variational parameters and MAP estimates in vectorized format at cycle t, i.e.,

$$\boldsymbol{\psi}_1^{(t)} = (m^{(t)}_{\widetilde{\mu}_1}, \mathrm{vech}(\Sigma^{(t)}_{\widetilde{\mu}_1}), m^{(t)}_{\widetilde{\mu}_0}, \mathrm{vech}(\Sigma^{(t)}_{\widetilde{\mu}_0})\ldots\alpha^{(t)}, \beta^{(t)}).$$

We stop the algorithm if $\|\boldsymbol{\psi}_1^{(t)} - \boldsymbol{\psi}_1^{(t-1)}\| < \kappa_1$, where κ_1 is a pre-specified tolerance.

Step 4 of Algorithm S2. For the classification phase, let $\boldsymbol{\psi}_2^{(t)}$ denote a vector containing all variational parameters and MAP estimates in vectorized format at cycle t, i.e.,

$$\boldsymbol{\psi}_2^{(t)} = (m_{z_{n+1}}^{(t)}, \text{vech}(\Sigma_{z_{n+1}}^{(t)})).$$

We stop the algorithm if $\|\boldsymbol{\psi}_2^{(t)} - \boldsymbol{\psi}_2^{(t-1)}\| < \kappa_2$, where κ_2 is a pre-specified tolerance.

S2.6 Computational cost of the log length-scale process

As mentioned in Section 2 of the main mauscript, our parameterization for the log length-scale process $\nu_i(t) = S(t) + \zeta_i$ allows us some flexibility to account for between-observation variation in the location-varying length scale while reducing the computional complexity. Here, we compute the computational complexity of Algorithm S3 (with ADADELTA learning rate) under two scenarios - the full parameterization where $\nu_i(t) \sim \text{GP}(\mu_\nu, Q^{-1}_{S;\tau,\lambda})$, and our proposed parameterization where $\nu_i(t) = S(t) + \zeta_i$. Figure S2 provides details on the breakdown of the computational complexity for the SVB update under each of the two scenarios.

Step in Algorithm 3	Vector/Matrix Operation	Full Parameterization	Proposed Parameterization
		Cost per loop	*Cost per loop*
1	General normal random variates	T	T
2	Triangular solve, then addition	2T	2T
3	Compute gradient, then multiply sparse matrix with vector	9T	(6n+3)T
–	Compute learning rate for mean	12T	12T
4	Vector addition	T	T
5	Solve linear equations, then compute product of two vectors	4T	4T
6	Multiply diagonals	T	T
–	Compute learning rate for precision	24T	24T
7	Sparse matrix addition	2T	2T
8	Multiply diagonals	T	T
	Total cost per loop	57T	6nT + 51T
	Total number of loops	nB	B
	Total cost for all loops	57nBT	6nBT + 51BT
	Additional cost for Newton's algorithm	0	6nB'T + 4nB'
	Final cost	57nBT	6nBT + 51BT + 6nB'T + 4nB'

Fig. S2: Comparison of computational cost between full and proposed parameterization for observation-specific log length scale. Notations: n = sample size; T = number of observed locations; B = number of stochastic VB iterations; B' = number of newton's algorithm iterations for computing the MAP estimates of $\boldsymbol{\zeta}$.

S3 Details on scalability and efficiency (Section 4)

Thomas' algorithm The computation of the variational mean for the mean functions and the latent process involves solving a large system of linear equations of the form

$$\mathbf{Q}\mathbf{a} = \mathbf{b},$$

where $\mathbf{Q}$ is a $T \times T$ tridiagonal matrix and $\mathbf{a}, \mathbf{b} \in \mathbb{R}^T$. Note we are solving for $\mathbf{a}$ in the above problem. The traditional approach to obtain a solution for $\mathbf{a}$ is the Gaussian elimination which has complexity $O(T^3)$. Here, we describe the Thomas' algorithm (Higham, 2002) for solving a system of linear equations with complexity $O(T)$. The low computation cost is achieved by utilising the tridiagonal structure of $\mathbf{Q}$. In particular, let a_j and b_j denote the j-th entry of $\mathbf{a}$ and $\mathbf{b}$ respectively and let Q_{ij} denote the (i, j)-th entry of $\mathbf{Q}$. Then, the solution to $\mathbf{a}$ may be obtained via the following algorithm:
Set $f_{1,1} = Q_{1,2}/Q_{1,1}$ and $f_{2,1} = b_1/Q_{1,1}$. For $j = 2, \ldots, T-1$,

1. $f_{1,j} = Q_{j,j+1}/(Q_{j,j} - Q_{j,j-1}f_{1,j-1})$.
2. $f_{2,j} = (b_j - Q_{j,j-1}f_{2,j-1})/(Q_{j,j} - Q_{j,j-1}f_{1,j-1})$.

Set $a_T = (b_T - Q_{T,T-1}f_{2,T-1})/(Q_{T,T} - Q_{T,T-1}f_{1,T-1})$. For $j = T-1, \ldots, 1$, set

$$a_j = f_{2j} - f_{1j}a_{j+1}.$$

Sparse inverse subset The computation of the variational covariance matrix for the mean functions, latent processes, and log length scales involves inverting a large tridiagonal precision matrix $\mathbf{Q}$. However, only the main diagonal and first off-diagonal entries of the inverse precision matrix are required. Therefore, we adopt the sparse inverse subset algorithm (Takahashi, 1973) for efficient computation of the main diagonal and first off-diagonal entries. The algorithm begins by computing the Cholesky decomposition of the precision matrix, resulting in the factorization $\mathbf{Q} = \mathbf{L}\mathbf{D}\mathbf{L}^\top$, where $\mathbf{L}$ also turns out to be a unit lower triangular 1-banded matrix and $\mathbf{D}$ is a diagonal matrix. The complexity of this factorization is $O(T)$ and its implementation is described in Algorithm 1 of Durrande et al. (2019). Then, the required entries of the inverse precision matrix are computed by first recognizing that the inverse precision matrix may be expressed as

$$\mathbf{Q}^{-1} = \mathbf{D}^{-1}\mathbf{L}^{-1} + (\boldsymbol{I} - \mathbf{L}^\top)\mathbf{Q}^{-1},$$

where $\mathbf{D}^{-1}\mathbf{L}^{-1}$ is a lower triangular matrix with diagonal entries given by $1/D_{j,j}$ and $\mathbf{L}^\top$ is a unit upper triangular 1-banded matrix. Following the above expression, a recursive algorithm for computing the required entries is

$$P_{j,j} = 1/D_{j,j} - \sum_{j'>j} L_{j',j}P_{j',j},$$
$$P_{j-1,j} = -\sum_{j'\geq j} L_{j',j-1}P_{j',j},$$

where $\mathbf{P} = \mathbf{Q}^{-1}$. Details of the algorithm are provided in Algorithm 3 of Durrande et al. (2019).

Specification of variational precision matrix for log length scale Each SVB update involves repeated solving of systems of T linear equations. To reduce the computational complexity of the SVB updates for the log length scale parameters, we specify a sparse Cholesky decomposition for the variational precision matrix. In particular, for the approximate posterior of $\widetilde{\boldsymbol{\nu}}_k$, we specify the following 1-banded Cholesky decomposition

$$\Omega_{\widetilde{\boldsymbol{\nu}}_k} = \begin{bmatrix} \Omega_{\widetilde{\boldsymbol{\nu}}_k;11} & 0 & 0 & \dots & 0 \\ \Omega_{\widetilde{\boldsymbol{\nu}}_k;21} & \Omega_{\widetilde{\boldsymbol{\nu}}_k;22} & 0 & \dots & 0 \\ 0 & \Omega_{\widetilde{\boldsymbol{\nu}}_k;32} & \Omega_{\widetilde{\boldsymbol{\nu}}_k;33} & \dots & 0 \\ \vdots & & \ddots & \ddots & \vdots \\ 0 & \dots & 0 & \Omega_{\widetilde{\boldsymbol{\nu}}_k;TT-1} & \Omega_{\widetilde{\boldsymbol{\nu}}_k;TT} \end{bmatrix}.$$

Following the above specification, the precision matrix is tridiagonal which allows us to utilise Thomas' algorithm for solving the system of linear equations with cost $O(T)$.

S4 Details on numerical results (Section 5)

S4.1 Details on competing methods

Both `VNPDA` and `VLDA` are Bayesian discriminant analysis models that assume independence between the locations. `VLDA` assumes the distribution of the variables are Gaussian, whereas `VNPDA` uses a nonparametric approach to fitting the distribution of the variables. `penLDA-FL` is a penalized Fisher's discriminant analysis model that assumes independence between locations and utilizes a fused-lasso penalty to obtain smooth estimate for the group-specific mean processes. A set of discriminative locations is elicited from random forest by computing the variable importance measure of each location and identify the top 20th-percentile. `SparseLDA` is a Fisher's discriminant analysis model where the discriminant vectors are penalized through an elastic net minimization. Both `SVM-L1` and `SVM-L2` employ their respective penalty functions on the linear weights for the predictors. The cost parameter for `SVM-L2` is specified using the heuristic approach provided by the `Liblinear` package, whereas the cost parameter is tuned using five-fold CV by partitioning the training dataset. The competing classifiers that account for dependence between time points are `GPDA`, random forest, `SparseLDA`, and both versions of `SVM`.

S4.2 Details on simulation datasets

Noise variances The values of the true noise variances are as follows.
Simulation 1: $\sigma^{\star 2}_{\epsilon,1t} = \kappa_\epsilon\{\gamma^\star_t e^{\sin(0.1t)} + \frac{1}{2}(1-\gamma^\star_t)e^{\cos(0.1t)}\}$; $\sigma^{\star 2}_{\epsilon,0t} = \kappa_\epsilon\{\gamma^\star_t e^{\cos(0.1t)} + \frac{1}{2}(1-\gamma^\star_t)e^{\cos(0.1t)}\}$; $\kappa_\epsilon = 0.005$.

Simulation 2: $\sigma^{\star 2}_{\epsilon,1t} = \kappa_\epsilon\{\gamma^\star_t e^{\cos(0.1t)} + \frac{1}{2}(1-\gamma^\star_t)e^{\cos(0.1t)}\}$; $\sigma^{\star 2}_{\epsilon,0t} = \kappa_\epsilon\{\gamma^\star_t e^{\sin(0.1t)} + \frac{1}{2}(1-\gamma^\star_t)e^{\cos(0.1t)}\}$; $\kappa_\epsilon = 0.5$.

Simulation 3: $\sigma^{\star 2}_{\epsilon,1t} = \sigma^{\star 2}_{\epsilon,0t} = 2 + \cos(0.01t)$.

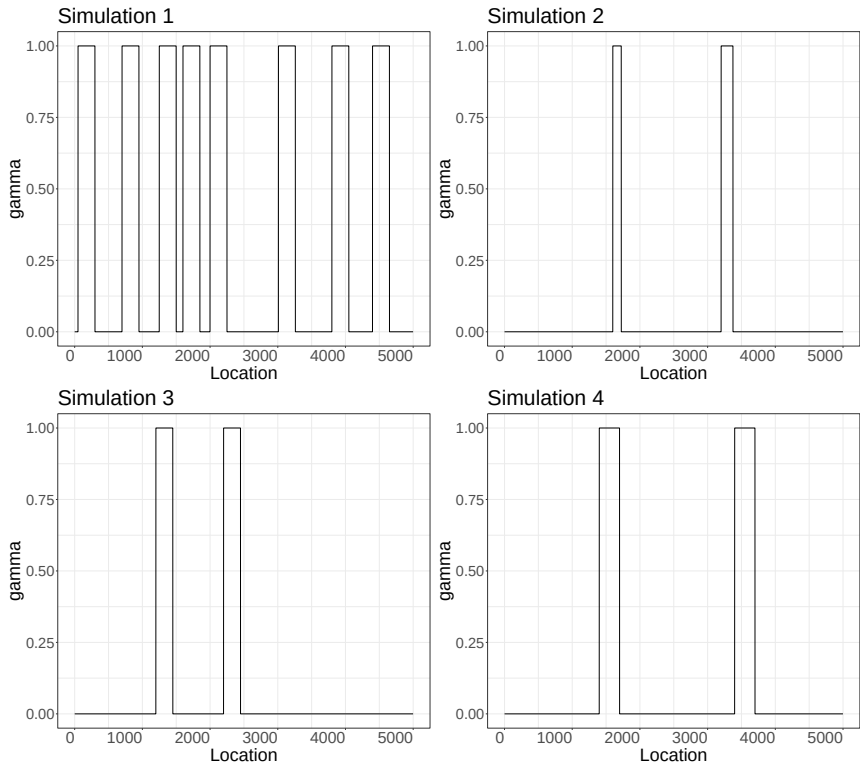


Fig. S3: Simulation settings for $\gamma^\star$.

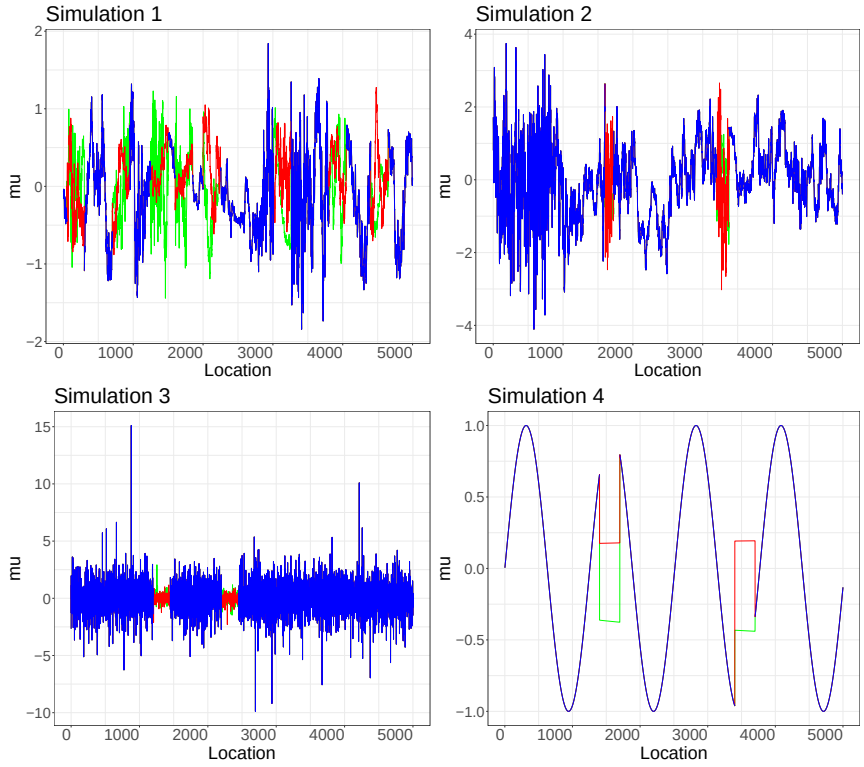


Fig. S4: Simulation settings for $\boldsymbol{\mu}_1^\star$ and $\boldsymbol{\mu}_0^\star$. At discriminative locations, red denotes group 1 and green denotes group 0. Blue denotes non-discriminative locations.

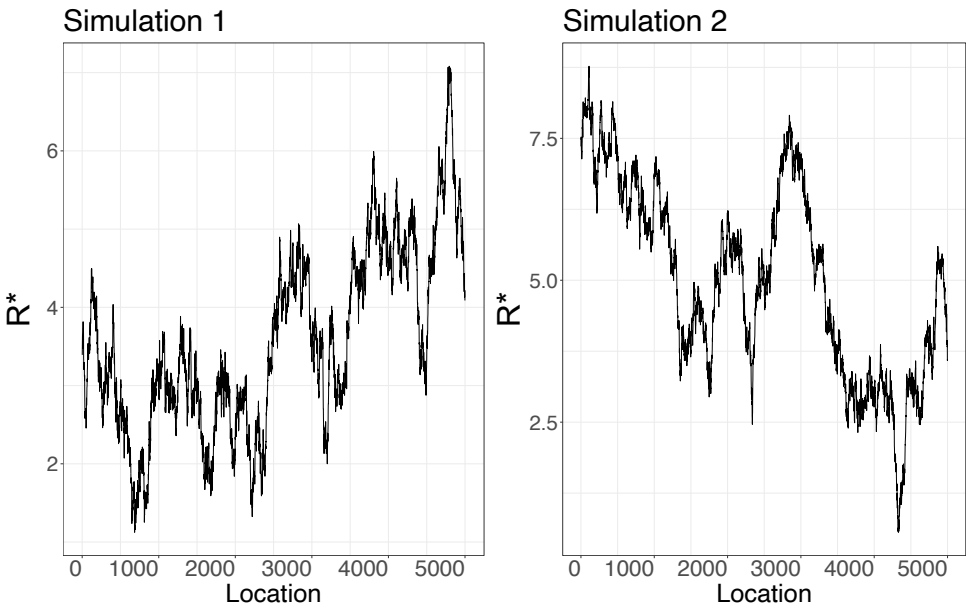
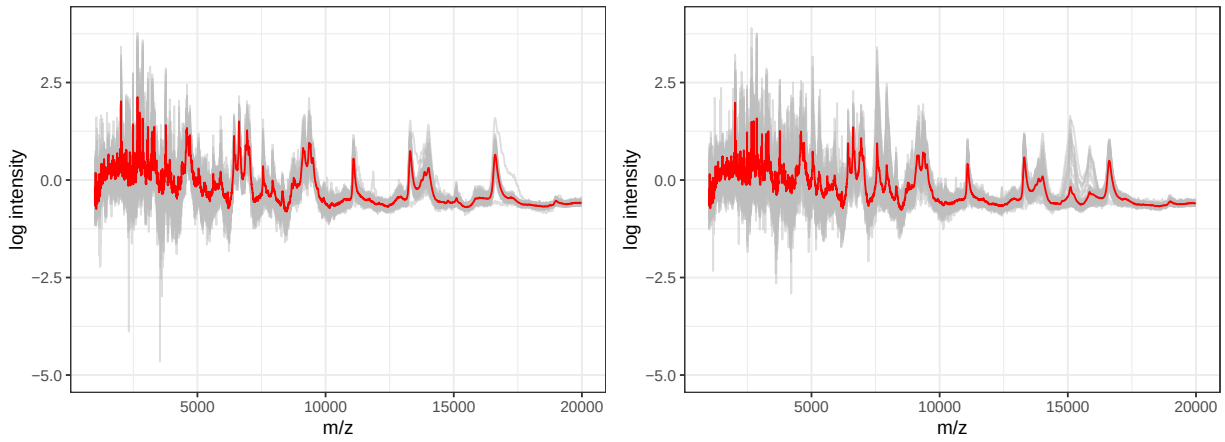


Fig. S5: Simulation settings for $R^\star$.

S4.3 Pre-processing steps for proteomics datasets

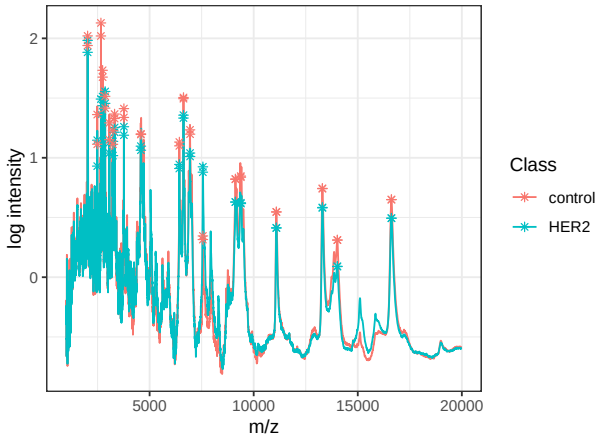
SARS-CoV-2 dataset The samples were collected during the March-April 2020 outbreak from laboratories in three different countries: Argentina (281 samples), Chile (30 samples), and Peru (51 samples). The processed data are contained in the Supplementary Information (Nachtigall et al., 2020) and the raw spectra are available from the PRIDE repository under accession code PXD021388. Following the steps of (Nachtigall et al., 2020), we preprocess the data using the `MALDIquant` and `MALDIquantForeign R` packages (Gibb and Strimmer, 2012; Gibb, 2019). Specifically, we first trim the spectra to the range 3 to 15.5 kDA. A square root transformation was applied for variance stabilization, followed by smoothing, baseline correction, and normalization using total ion current calibration. To account for differences across laboratories and instruments, alignment of the spectra was performed using the `warpMassSpectra` function in the `MALDIquant` package. Lastly, the `intensityMatrix` function is used to interpolate intensity values across a common m/z range.

Breast cancer dataset The dataset contains 167 plasma samples from 155 women, among which 64 were normal health controls, 55 were HER2 positive, 35 were ER/PR positive, and the remaining 13 samples are multiple samples from a single individual. The data are stored in the `ProData R` package (Li, 2020b), and we preprocess the spectra with the `PROcess R` package (Li, 2020a), as described in (Li et al., 2005). More specifically, we apply a baseline correction to remove baseline drifts due to chemical noise. Next, we trim the initial 1 kDA of the spectra and then normalize to correct for systematic variation.



(a) Controls

(b) HER2



(c) Class-averaged spectra

Fig. S6: Illustration of the breast cancer data, with the log intensities of the preprocessed spectra for (a) healthy controls and (b) HER2 positive patients (class-average given in red). The class averages are compared in (c) with the detected peaks identified through the traditional two-stage pipeline (available in the `ProData` package) marked with stars.

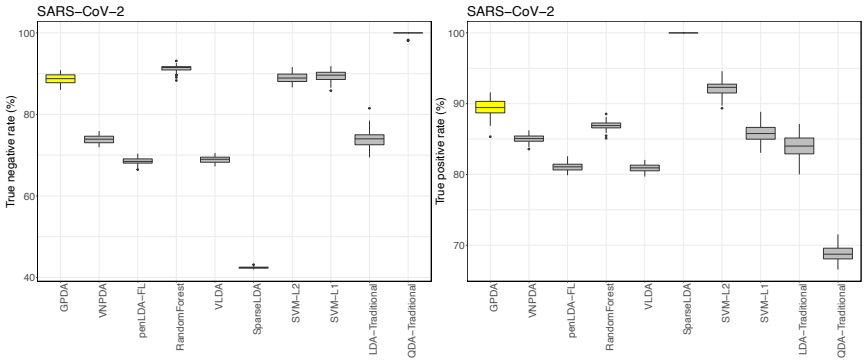


Fig. S7: True positive rates (left) and true negative rates (right) of all methods for SARS-CoV-2 dataset.

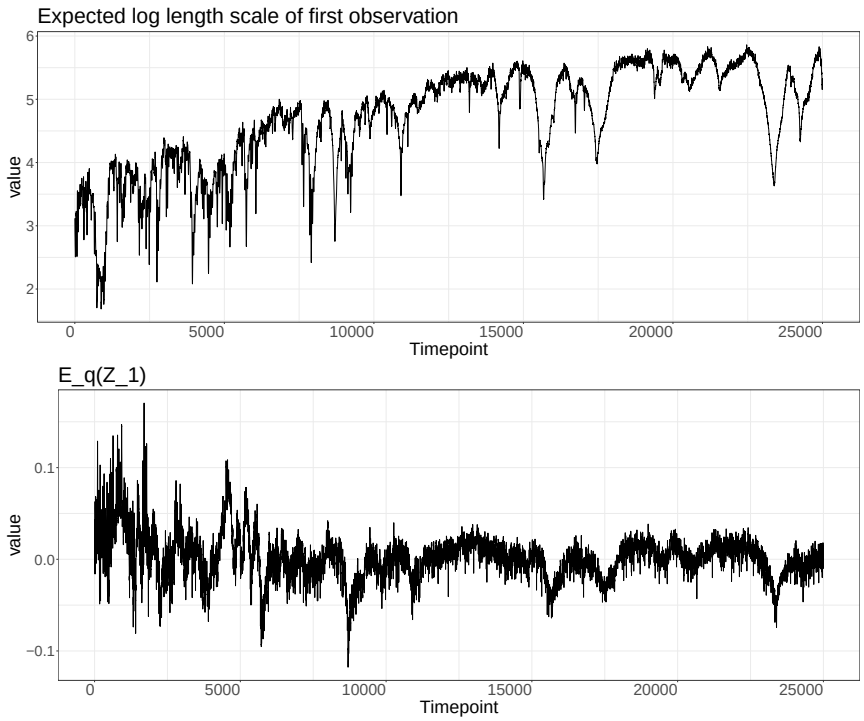


Fig. S8: Posterior expectation of $\boldsymbol{\nu}_1$ (top) and $\boldsymbol{z}_1$ (bottom) for SARS-CoV-2 dataset.

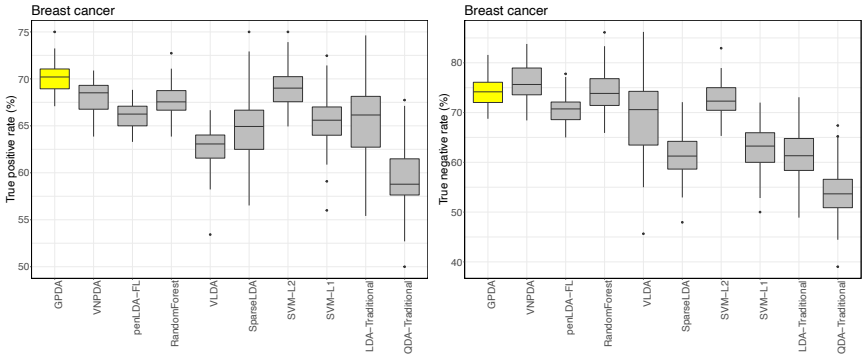


Fig. S9: True positive rates (left) and true negative rates (right) of all methods for breast cancer dataset.

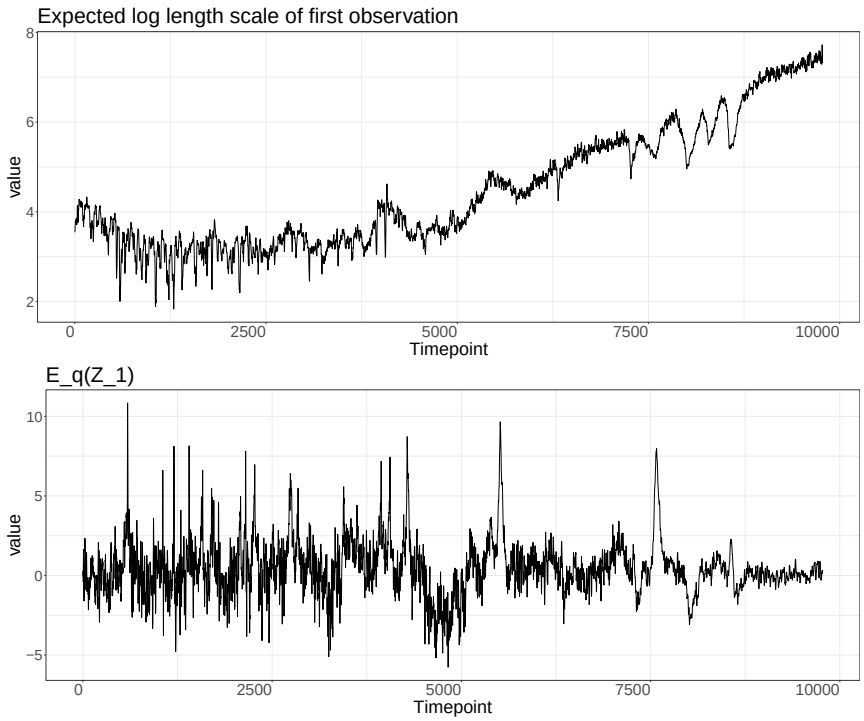


Fig. S10: Posterior expectation of $\boldsymbol{\nu}_1$ (top) and $\boldsymbol{z}_1$ (bottom) for breast cancer dataset.

References

Betancourt, M. (2017). Robust Gaussian Processes in Stan III. https://betanalpha.github.io/assets/case_studies/gp_part3/part3.html. 3

Durrande, N., Adam, V., Bordeaux, L., Eleftheriadis, S., and Hensman, J. (2019). Banded matrix operators for Gaussian Markov models in the automatic differentiation era. In Proceedings of the 22nd International Conference on Artificial Intelligence and Statistics, volume 89. 7

Gibb, S. (2019). MALDIquantForeign: Import/Export Routines for 'MALDIquant'. R package version 0.12. 10

Gibb, S. and Strimmer, K. (2012). MALDIquant: a versatile R package for the analysis of mass spectrometry data. Bioinformatics, 28(17):2270–2271. 10

Higham, N. J. (2002). Accuracy and stability of numerical algorithms: second editions. Society for Industrial and Applied Mathematics. 7

Li, X. (2020a). PROcess: Ciphergen SELDI-TOF Processing. R package version 1.64.0. 10

Li, X. (2020b). ProData: SELDI-TOF data of Breast cancer samples. R package version 1.26.0. 10

Li, X., Gentleman, R., Lu, X., Shi, Q., Iglehart, J., Harris, L., and Miron, A. (2005). Seldi-tof mass spectrometry protein data. In Bioinformatics and Computational Biology solutions using R and Bioconductor, pages 91–109. Springer. 10

Nachtigall, F. M., Pereira, A., Trofymchuk, O. S., and Santos, L. S. (2020). Detection of SARS-CoV-2 in nasal swabs using MALDI-MS. Nature Biotechnology, 38(10):1168–1173. 10

Salinas, S. (2001). Introduction to Statistical Physics. Springer Science & Business Media. 5

Takahashi, K. (1973). Formation of sparse bus impedance matrix and its application to short circuit study. In Proc. PICA Conference. 7