Forensic Science International: Genetics 65 (2023) 102890

Contents lists available at ScienceDirect

Forensic Science International: Genetics

journal homepage: www.elsevier.com/locate/fsigen

Research paper

Variational inference accelerates accurate DNA mixture deconvolution

Mateusz Susik [a,b,*], Ivo F. Sbalzarini [b,c,d]

[a] Biotype GmbH, Dresden, 01109, Germany
[b] Technische Universität Dresden, Faculty of Computer Science, Dresden, 01187, Germany
[c] Max Planck Institute of Molecular Cell Biology and Genetics, Dresden, 01307, Germany
[d] Center for Systems Biology Dresden, Dresden, 01307, Germany

ARTICLE INFO

Keywords:
Probabilistic genotyping
Stein variational gradient descent
Variational inference
Bayesian inference
Runtime
Precision

ABSTRACT

We investigate a class of DNA mixture deconvolution algorithms based on variational inference, and we show that this can significantly reduce computational runtimes with little or no effect on the accuracy and precision of the result. In particular, we consider Stein Variational Gradient Descent (SVGD) and Variational Inference (VI) with an evidence lower-bound objective. Both provide alternatives to the commonly used Markov-Chain Monte-Carlo methods for estimating the model posterior in Bayesian probabilistic genotyping. We demonstrate that both SVGD and VI significantly reduce computational costs over the current state of the art. Importantly, VI does so without sacrificing precision or accuracy, presenting an overall improvement over previously published methods.

1. Introduction

DNA mixture analysis using probabilistic genotyping (PG) software is at the core of forensic science methodologies. While the greatest attention has to be placed on the accuracy of the results, there are other factors that play a role. Recently, it has been shown how the precision of a PG system can be improved [1]. Run-to-run variability was reduced by using Hamiltonian Monte Carlo inference and enforcing strict convergence criteria. Another factor that can be considered is the computational runtime. In many cases, estimating a likelihood ratio (LR) can take more than an hour. Given the workload forensic laboratories face, faster PG algorithms are desirable as long as they produce results of equal accuracy and precision.

The accuracy of a PG result directly depends on how well the PG algorithm is able to approximate or estimate the probabilities of the parameters of a model given an observed electropherogram. Often, this is formulated as a Bayesian estimation problem where the task is to estimate the posterior distribution. In state-of-the-art PG models [1,2], it is not possible to calculate the posterior directly by integrating over the parameters in a reasonable timeframe. Instead, sampling algorithms are used, most prominently Markov-Chain Monte-Carlo (MCMC) methods with random walk [2] or Hamiltonian proposal distributions [1]. These algorithms are used to provide samples for approximating the posterior distribution.

The posterior distribution lives in a space whose dimensionality depends on the number of unknown parameters to be estimated. Since the computational cost of a PG algorithm scales with the number of model likelihood evaluations it requires, it is desirable to have PG methods that estimate the posterior distribution as accurately as possible using as few samples as possible.

The cost-performance trade-off is a classic research topic in Bayesian inference [3]. There, besides MCMC methods, also other types of algorithms have been proposed. In particular variational methods have been successful at improving the computational performance of Bayesian inference [4]. It therefore seems natural to also adopt these approaches in DNA mixture deconvolution and benchmark their performance against the state of the art in PG.

Here, we present implementations of two variational inference techniques adapted to PG applications: variational inference with an evidence lower bound objective (VI) [4] and Stein Variational Gradient Descent (SVGD) [5]. We show that both SVGD and VI achieve shorter runtimes than the MCMC-based methods. Importantly, VI does so without sacrificing precision or accuracy, presenting an overall improvement over the state of the art in PG.

2. Materials and methods

In the context of PG, we seek to calculate a LR that compares two competing hypotheses: the hypothesis of the prosecutor (H_p) and the hypothesis of the defendant (H_d):

$$\mathrm{LR} = \frac{P(V|H_p)}{P(V|H_d)} = \frac{\sum_j P(V|S_j)P(S_j|H_p)}{\sum_j P(V|S_j)P(S_j|H_d)}. \quad (1)$$

* Corresponding author at: Biotype GmbH, Dresden, 01109, Germany.
E-mail address: m.susik@biotype.de (M. Susik).

https://doi.org/10.1016/j.fsigen.2023.102890
Received 1 December 2022; Received in revised form 2 May 2023; Accepted 15 May 2023
Available online 20 May 2023

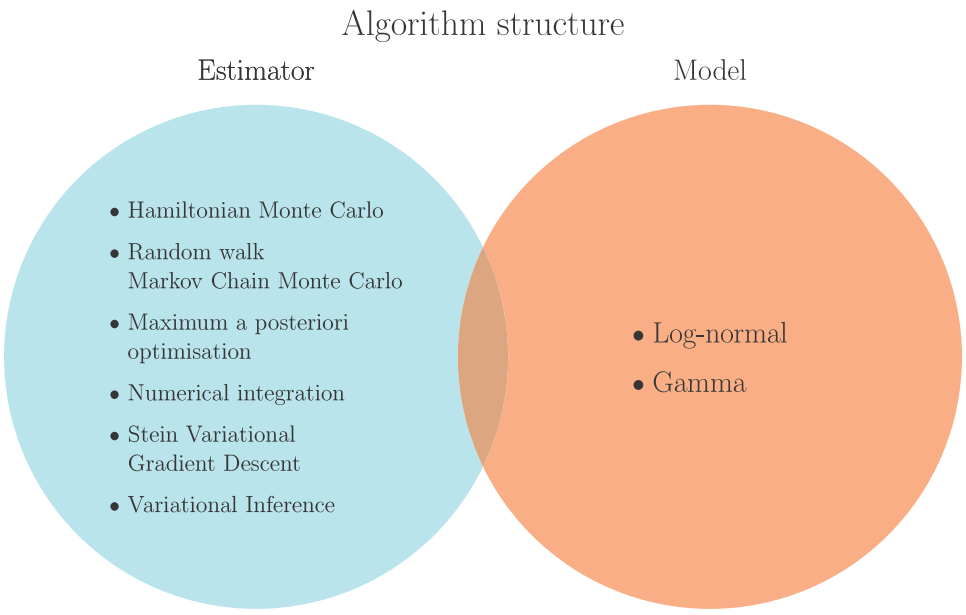

Fig. 1. The basic structure of a probabilistic genotyping algorithm: The algorithm consists of an *estimator*, a tool for approximating a distribution, and a *model* that defines the unnormalised posterior, usually by defining the likelihood and the prior. Different popular choices of estimator methods and PG models are given in the circles. Their choices are largely mutually independent.

To calculate this, one introduces a model M for the probability of observing the evidence, given a genotype S_j ("genotype weight"). The parameters of the model M are estimated by Bayesian inference. In Bayesian inference, the main task is to estimate the posterior probability of the model parameters given data/evidence V:

$$P(M|V) = \frac{P(V|M)P(M)}{P(V)}\,. \tag{2}$$

The probability of the evidence $P(V)$ is a constant that can be neglected for the purpose of optimisation. This leads to an unnormalised value proportional to the posterior.

The probability of observing the evidence assuming a hypothesis is calculated as the sum over all the possible genotype sets S_j. The probability of observing the evidence assuming a genotype set can be then calculated from the unnormalised posterior:

$$P(M|V) \propto P(M)\sum_j P(V|M,S_j)\,. \tag{3}$$

The proportionality comes from the fact that $P(S_j|M)$ is assumed to be a uniform distribution and is therefore dropped. In practice, the genotype weights are estimated from the posterior distribution (Eq. 5 in Supplementary Material 3) using Bayesian inference with H_d the assumed hypothesis. The likelihood ratio is independent of the probability of the evidence, as it crosses out.

A Bayesian inference algorithm can in practice be seen as consisting of two parts: an estimator and a model (Fig. 1). The model defines the unnormalised posterior, and the estimator defines the way how an approximation of this distribution is obtained. These two parts are largely independent of each other, meaning that, for example, an estimator can be replaced with another one.

In practice, this ideally means that data scientists can create a model based on observed data and/or theoretical knowledge, while different estimators can be used interchangeably in order to optimise for computational cost and/or accuracy and precision of the PG result. Yet, the model might constrain the choice of the estimator, as different estimators have different structural limitations.

2.1. Model definition

In this work, we follow the model and hyper-parameterisation used in Hamiltonian Monte Carlo (HMC) [1] with a few exceptions as described next. These minor adaptations are required for variational estimators to work correctly. We recollect the complete description of the original model in Supplementary Material 3. The first adaptation to the model concerns the likelihood probability distribution suggested by the authors of STRmix™ [2]:

$$\ln\frac{O}{x} \sim \mathcal{N}\left(0, \frac{c^2}{x}\right)\,. \tag{4}$$

This postulates that the ratio of the observed peak height O to the expected peak height x follows a log-normal distribution with mean 0 and a variance proportional to the square of a parameter c simulated by the model and inversely proportional to the expected peak height x. During inference, the estimator might thus try values of c close to 0, as long as the Gamma prior [6] does not forbid this. The probability density $f_{\mathrm{Log}\mathcal{N}}$ of this parameterised log-normal distribution can then reach arbitrarily large values:

$$\{O = x \wedge x > 0\} \Rightarrow \lim_{c\to 0} f_{\mathrm{Log}\mathcal{N}}(x) \to \infty\,, \tag{5}$$

as the distribution collapses to a Dirac delta distribution. This is the first issue hampering the use of variational inference methods, as any variational estimator will exploit the resulting high likelihoods and thus ignore reasonable values of parameters.

The second issue is similar: Consider the prior probability distributions for the locus-specific amplification efficiencies (LSAE) α with a hyper-parameter σ_α set by the laboratory:

$$\log_{10}\alpha \sim \mathcal{N}(0, i^2)\,, \tag{6}$$

$$i^2 \sim \mathrm{Exp}(\sigma_\alpha)\,. \tag{7}$$

The prior density for the LSAEs f_{LSAE} is then also unbounded:

$$\{\alpha = 1\} \Rightarrow \lim_{i\to 0} f_{\mathcal{N}(0,i^2)}(\log_{10}\alpha) \to \infty\,. \tag{8}$$

These singularities are not a problem for HMC estimators, who will avoid them because of the high curvature of the posterior in the vicinity of the singularities. When the sampler tries to explore these parts of the posterior, the trajectory of the simulated Hamiltonian differs too much from the expected Hamiltonian. The sample is then rejected and marked as a "divergence". These samples then negatively impact the

Table 1
Estimated priors based on 37 2- and 3-contributor Globalfiler™ ProvedIT mixtures not used for the test benchmark.

Prior	Distribution
Stutter peak height standard deviation	$c_s - 6.086 \sim \text{LogN}(2.485, 1.031)$
Allelic peak height standard deviation	$c_p - 1.63 \sim \text{LogN}(1.975, 0.325)$
LSAE standard deviation hyper-prior	$i \sim \text{LogL}(-1.894, 0.236)$

runtime of the algorithm. The convergence of the chains is slower and, in extreme cases, can lead to a bias in the estimate.

Variational inference estimators, however, are not able to work with posteriors that contain singularities. Any evaluation of the gradient around a singular point would result in a drastic, possibly infinite change of learnable parameters that renders inference unstable. We therefore modify the model as follows:

- use shifted log-normal (LogN) priors for allele and stutter peak-height standard deviations, instead of the Gamma priors used by STRmix™.
- use a log–logistic (LogL) hyper-prior for the LSAE standard deviation, instead of an exponential one on LSAE variance (Eq. (7)).

The exact choice of the prior distribution families, in our case log-normal and log–logistic, is not crucial as long as the estimates densities protect the algorithms from the singularities.

To estimate the parameters of these prior distributions, we start with posterior estimation of a training set of DNA mixtures[1] using flat priors. We then extract, for each mixture, the samples (peak height variance, stutter height variance, LSAE variance) that are larger than the 90th percentile of the estimated parameter distribution. Finally, we choose the parameters of the prior distributions that maximise the likelihood of this subsample, following the "empirical Bayes" method.

For the following results, we used 37 2- and 3-contributor filtered Globalfiler™ mixtures that were not a part of the test benchmark [8,9]. For the list of mixtures, please see Supplementary Material 1. The estimated priors are presented in Table 1 and compared with the priors from Riman et al. [8] in Supplementary Material 4, Figure 1.

The shifted peak-height standard deviation priors prohibit parameter values smaller than the shift. The procedure behind the variational algorithms will therefore never explore parameter values that cause the undesirable behaviour. One could argue that this introduces a bias into the analysis, as the "true" value of the parameter might not be accessible to the estimator. While this is true, it is not specific to our work. Indeed, also in other models these distributions are shifted to the right of the estimates from the model, see, e.g., Figure 4 from Taylor et al. [7]. Moreover, a natural meaning of the peak height standard deviation parameter is the confidence of the model when estimating peak heights. The lower the value of the parameter, the more confident the model. An overestimation of these parameters might be then desirable, since this increases the level of uncertainty of the estimator.

2.2. Variational inference

In Bayesian inference, one typically distinguishes sampling methods, such as MCMC and HMC, from variational methods. Sampling methods iteratively draw samples from the posterior. They construct a Markov chain of the samples. The chains, as long as they are ergodic, converge to the desired stationary distribution, which is the (unnormalised) posterior of the model in the limit of infinitely many samples. In practice, however, the number of samples is finite. Therefore, convergence criteria (e.g. the Gelman–Rubin test) are used to determine when to stop the sampler. This trades off computational cost, which is proportional to the number of samples drawn, with accuracy and precision (i.e., reproducibility) of the result. Variational methods avoid this trade-off by directly building an approximation to the posterior. This approximation comes from a family of distributions, called the variational family, which is selected by the design of the algorithm. For simpler models, it is sometimes possible to construct an exact variational family, but this is not the case for the model considered here. Once an LR needs to be calculated, a variational method provides a sample from its variational distribution. We note, however, that sampling in VI is significantly cheaper than inference.

The principles behind the two implemented methods are illustrated in Fig. 2 and explained in more detail in the next two sub-sections.

2.2.1. Variational inference with an evidence lower-bound objective

The first variational method we consider uses multivariate normal distributions as the variational family $\mathbb{Q}$ parameterised with M. It then aims to minimise the Kullback–Leibler (KL) divergence of a variational distribution $Q \in \mathbb{Q}$:

$$\text{KL}\left(Q(M) \parallel P(M|V)\right) = \int_M q(m) \log\left(\frac{q(m)}{p(m|V)}\right) \mathrm{d}m\,, \tag{9}$$

where $p(m|V)$ and $q(m)$ are the densities of the posterior and the variational distributions, respectively. The final result of VI is the distribution from the variational family that resembles the posterior the most, where resemblance is measured by the KL divergence between the two distributions. While KL divergence cannot be calculated directly in our case, it can be shown that in general:

$$\text{KL}\left(Q(M) \parallel P(M|V)\right) = \log P(V) - \text{ELBO}\,, \tag{10}$$

$$\text{ELBO} = \mathbb{E}_{m' \sim Q}\left[\log \frac{p(m', V)}{q(m')}\right]\,, \tag{11}$$

where ELBO is the so-called "evidence lower-bound objective". Since the log-evidence is a constant independent of the model, the KL divergence is minimised by maximising ELBO. This formulation of the optimisation problem is flexible w.r.t. the choice of the variational family. In order to consider a wide choice of possible solutions, we here use a multivariate Gaussian family with full covariance matrix.

This choice of variational family could harm the quality of the results if the true posterior distribution cannot be approximated by multivariate Gaussians. Empirically, however, we observe that the posterior estimated by HMC, which should be a close approximation of the true posterior, is similar to a transformed multivariate Gaussian.[2] We therefore expect VI to work well. We compare the posterior estimated by the different methods in Supplementary Material 4, Figure 2. We run all three methods identically with the priors presented in Section 2.1 on the F04_RD14—0003—42_43—1;9—M3a—0.15GF—Q0.5_06.15sec (referred to as "F04") mixture from the ProvedIT dataset [10]. Since the probability space is transformed [1], the marginal density plots do not always look Gaussian. A transformed normal distribution output by VI, however, approximates well the answer provided by HMC. Still, we note some differences exist, such as decreased variance of the estimated marginal degradation distributions.

We maximise the ELBO using the Adamax algorithm [11]. This algorithm internally uses a Monte-Carlo estimator for the ELBO according to Eq. (10). We use 10 samples per estimation and a variable learning

[1] In our experiments, the use of mixture profiles instead of single-source profiles for calibrating the model provided wider posterior estimates for the peak-height standard deviation, as also previously reported [7], which avoided model restriction issues

[2] The transformation is required, as the algorithms we use cannot work on constrained subspaces.

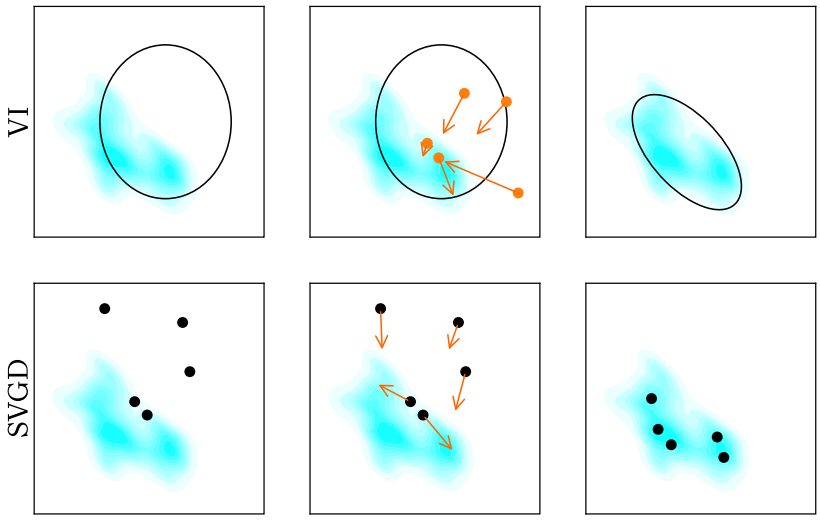

Fig. 2. A visual illustration of the concepts behind the algorithms used here to estimate the posterior distribution. The true posterior distribution in this two-dimensional example is shown by the cyan cloud. **Top:** VI starts with an initial distribution from the variational family (left), here a multi-variate Gaussian shown by the outline of its covariance ellipse (black). A Monte Carlo sample is drawn from the current distribution (middle), and the model gradient is computed at each sample (arrows). The parameters of the distribution are adapted accordingly, and the procedure is repeated until the estimated distribution fits the posterior (right). **Bottom:** In SVGD, particles are initialised randomly (left). An update rule attracts the particles to the posterior's probability mass and repels them from one another (middle). These updates are repeated until the particle density fits the posterior (right). (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

rate[3] (LRate) schedule: The first iterations are performed with LRate = 0.01; then, after every 100 iterations, the LRate is multiplied by 1.5 until it reaches 0.05. This adaptation prevents diverging gradients that could otherwise be caused by bad initialisation of the variational family. Indeed, during the first few iterations, gradient computation could become numerically unstable, as an outlier sampled from the variational distribution could cause it to significantly depart from reasonable parameter values. Convergence of the optimiser is monitored by comparing the mean value of the ELBO every 100 iterations. If this mean is smaller than the mean from the previous hundred iterations, the optimiser is stopped, since no further improvement was achieved.

2.2.2. Stein variational gradient descent

Stein variational gradient descent (SVGD) [5] intends to find a composition of transformations for an ensemble of n particles x that maps an initial distribution to the best approximation of the true posterior as quantified by the KL divergence. The optimal trajectory for particles is approximated by taking a discretisation schema. A single iteration is defined as:

$$x_i \leftarrow x_i + \frac{\epsilon}{n}\sum_{j=1}^{n}\left[k(x_i,x_j)\nabla_{x_j}\log p(x_j) + \nabla_{x_j}k(x_i,x_j)\right] . \tag{12}$$

The scalar ϵ is the learning rate, and ∇ denotes a gradient operator w.r.t. the subscripted variable. Intuitively, Eq. (12) means that:

- The particles prefer to be in regions of high probability density, as indicated by the gradient of the log-posterior. The log-posterior of the neighbours will dominate this term for any particle given the multiplication with the kernel.
- At the same time they are repelled from one another by the gradient of the kernel in order to not collapse into a single mode and cover the whole posterior.

Following Liu & Wang [5], we use a radial basis function kernel $k(x_i,x_j) = \exp\left(-\|x_i - x_j\|_2^2/h\right)$, and we set bandwidth h to $\text{med}^2/\ln n$ where med is the median of the pairwise distances between the n particles. In our experiments, we use 100 particles and run 500 updates.

[3] The learning rate is hyper-parameter that scales the magnitude of the gradient-descent updates.

The number of particles is manually chosen to balance the quality of posterior estimation and the speed of inference. Our implementation uses the Adamax algorithm [11] with learning rate 0.25 to efficiently approximate the gradient descent in Eq. (12). Adjusting the learning rate, as done for VI, is not necessary, since instabilities in the gradient computation cannot occur.

When only one particle is used, $n = 1$, SVGD reduces to maximum a posteriori estimation, which is the method used in Euroformix [12].

We reduce memory consumption by iteratively evaluating the unnormalised posterior p in 10 equal batches of 10 particles each. This hyper-parameter can be adjusted depending on the available computational resources. Memory consumption scales linearly with batch size.

2.3. Implementation details

From the estimated posteriors, we compute LRs as described [1]. We first draw a sample from the estimated posterior distribution. For SVGD, the 100 particles directly represent the sample [5]. For VI, we sample 1000 points from the estimated Gaussian. We note that increasing the number of particles in SVGD would linearly increase the computational time, while in case of VI we sample from an explicit Gaussian distribution only after the inference is finished, and thus we can afford a larger sample. Subsequently, the same approach as in HMC is used for both algorithms: we calculate the deconvolution by averaging the values across all samples.

We implement both the SVGD and VI estimators using the Tensorflow Probability library [13]. Gradients are computed using Tensorflow's automatic differentiation. All benchmarks are performed on NC4as T4 v3 Azure cloud GPU instances unless specified otherwise.

3. Results

We validate the use of variational inference for forensic PG and provide a comparative study between the two presented methods, SVGD and VI, and HMC. We benchmark three characteristics important for any PG system: the accuracy of the method, its precision, and the computational runtime. We use the term *scenario* to indicate the combination of a DNA mixture with a certain prosecutor/defendant hypothesis.

Table 2
Performance metrics for the three tested algorithms (HMC: Hamiltonian Monte Carlo [1]; SVGD: Stein variational gradient descent [this work]; VI: variational inference with evidence lower-bound objective [this work]) on the ProvedIT benchmark [10]; OotNT = Opposite of the Neutral Threshold [9].

		HMC	SVGD	VI
ROC AUC	2 contributors	0.99999	0.99999	0.99997
	3 contributors	0.99894	0.99892	0.99886
	4 contributors	0.99812	0.99643	0.99819
	Combined	0.99896	0.99843	0.99887
OotNT with false contributors	2 contributors	0	1	1
	3 contributors	2	2	2
	4 contributors	11	13	13
OotNT with true contributors	2 contributors	1	1	0
	3 contributors	10	10	14
	4 contributors	10	11	9
OotNT rate	2 contributors	0.002	0.002	0.003
	3 contributors	0.014	0.014	0.018
	4 contributors	0.021	0.024	0.022

We compare the results on the ProvedIT mixtures from the NIST comparative study [8]. The benchmark consists of 154 two-person mixtures, 147 three-person mixtures, and 127 four person mixtures. For each mixture, we test all possible scenarios with one true contributor and the same number of scenarios with random false contributors, following the work of Riman et al. [8]. Therefore, each scenario contains one assumed contributor (the "person of interest", POI) in H_p, and only unknown contributors in H_d. With the exception of the precision benchmark, each scenario is run once. Many of the mixtures in the dataset were degraded before analysis, and there are mixtures with different amounts of template and different fractions of contributor DNA material.

3.1. Accuracy

We observe almost identical performance in terms of the ROC area-under-the-curve (AUC) between HMC (0.99896) and VI (0.99887), see Table 2. SVGD performs slightly worse (0.99843), which is caused by one scenario with true contributor 33 resulting in a large negative $\log_{10}$ LR of −16.3753, see Supplementary Material 4, Figure 5. This is mixture E05, which was already described in Supplementary Material 2 of Ref. [9]. The LR for the locus D12S391 and the sub-sub-source hypothesis that resulted in the largest LR overall is $2.71 \cdot 10^{-27}$. When we consider the same alternative scenario as in Ref. [9], with the peak 18.3 (1220 RFU) added in locus D12S391, SVGD estimates a $\log_{10}$ LR of 11.646, which is close to the HMC estimate of 12.8367 in the same case. The sub-sub-source LR for the D12S391 locus then becomes 89.2053.

An interesting question is why SVGD is more sensitive to such missed peaks than both HMC and VI ($\log_{10}$ LR of 2.4072). It is known that SVGD is particularly prone to the curse of dimensionality [14]. In Supplementary Material 4, Figure 2, we present data showing that SVGD underestimates (compared to the other estimators) the allele peak-height standard deviation. This increases the confidence of the model. Therefore, this estimator is less robust against extreme observations, such as an uncalled peak with an RFU larger than 1000. It is likely that this improves when a larger number of particles is used, at the expense of increased computational cost. However, we do not further consider this here, since VI outperforms SVGD in both accuracy and performance then.

Next, we consider the numbers of OotNT (Opposite of the Neutral Threshold [9]) scenarios with true and false contributors, as well as the OotNT rates [9]. The results are again given in Table 2. In these metrics, both SVGD and VI perform slightly worse than HMC. For example, VI provided 4 OotNT scenarios with true contributors more than HMC for the 3-contributor mixtures, and both SVGD and VI provided 2 OotNT scenarios with false contributors more than HMC for the 4-contributor mixtures. All of these scenarios are characterised by low certainty. For example, the additional 3-contributor OotNT scenarios with true contributors had $\log_{10}$ LRs of −0.7756, −0.2065, −0.2970, and −0.3269 in case of VI, and 0.1248, 0.3516, 0.2888, and 0.462 in case of HMC. The detailed results are available in Supplementary Material 1.

Visualisations of the full results are given in Supplementary Material 4, Figures 3–8. All comparisons confirm the strong agreement between the compared methods, with the difference between the $\log_{10}$ LRs from the different methods rarely exceeding 2 bans (see Fig. 3). In the few scenarios where differences are larger, both methods provide strong evidence. The results for HMC are taken from the supplementary materials of Ref. [9], where the priors of Riman et al. [8] were used, whereas here we use the modified priors described in Section 2.1. This indicates that the exact choice of priors is not crucial to the reported results.

3.2. Precision

A desirable property of a PG software is low run-to-run variability. It has been shown that HMC greatly reduces this variability compared to random-walk MCMC [1]. Low run-to-run variability implies high reproducibility of the results, which is known as *precision*. We compare the precision of HMC as previously benchmarked [9] with the precision of SVGD and VI. For this, we perform 10 independent repetitions of the analyses for each scenario and compare two statistics: the standard deviation of the resulting $\log_{10}$ LR and the difference between the largest and smallest $\log_{10}$ LR. The measurements are presented in Fig. 4. SVGD is overall less precise than the other two algorithms, with 8 scenarios having a run-to-run standard deviation of the $\log_{10}$ LR larger than 0.2. The other two algorithms result in 3 scenarios each where the standard deviations are larger than 0.1. In 14 (HMC) and 13 (VI) scenarios, the standard deviations across runs are larger than 0.05. The precision of SVGD can be improved by either increasing the number of particles or using more iterations with learning-rate annealing. As we will see below in Section 3.3, however, this would not compare well against VI in terms of the computational cost and is therefore not explored.

Visualisations of the full results are available in Supplementary Material 2, with the raw data provided in Supplementary Material 1.

3.3. Runtime

The third performance metric of interest is the computational runtime. Variational techniques are usually faster, which is the main reason why they became popular with the deep-learning community. We confirm this here for PG, observing significant speedups as shown in Fig. 5. We compare the runtimes on identical computer hardware for all 4-contributor scenarios of the benchmark [8]. The computationally most demanding mixture takes 1 h and 54 min to be solved with HMC. For SVGD, the longest observed inference time is 36 min 28 s. VI provides the largest speedups, with the slowest inference completing in 18 min and 27 s. On average, VI is 4.33 times faster than HMC and 1.72 times faster than SVGD on 4-contributor scenarios. In 81.1% of the 4-contributor scenarios (103/127) VI completes the inference in under 10 min and in 96.1% (122/127) in under 12 min. SVGD completes 91.4% (116/127) of these scenarios in under 20 min.

These runtime benchmarks used GPU hardware. Most forensic laboratories, however, still use CPUs to compute. We therefore confirm the runtimes of the best-performing algorithm, VI, on four vCPU cores (AMD EPYC 7V12 Rome) for the F03_RD14—0003—48_49_50_29—1;4;4;4—M2U15—0.403GF—Q1.3_06.15sec mixture. For this scenario, VI runs for 10 min and 13 s on the GPU, whereas the same inference on the CPU takes 67 min and 14 s. We note, however, that the code contains no CPU-specific optimisations so that this figure could probably be improved if needed.

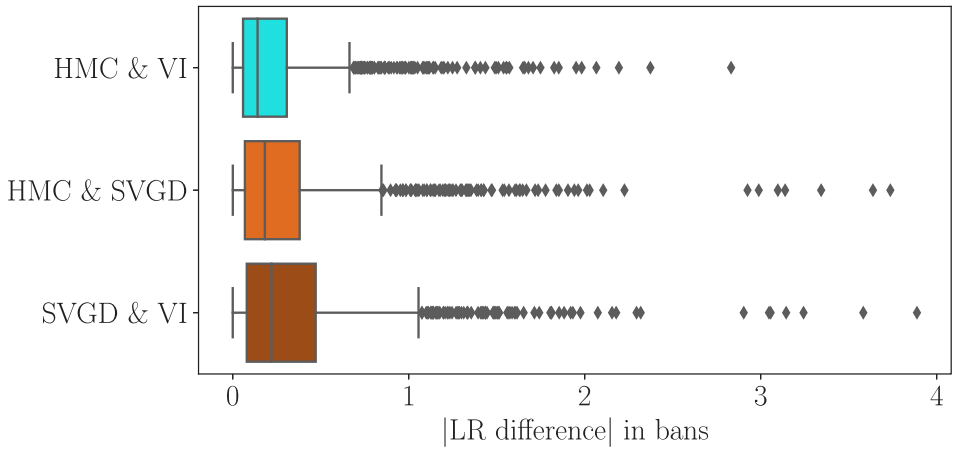

Fig. 3. Box plots of the absolute values of LR differences between estimators across all scenarios with true contributors, except for mixture E05 with Contributor 33.

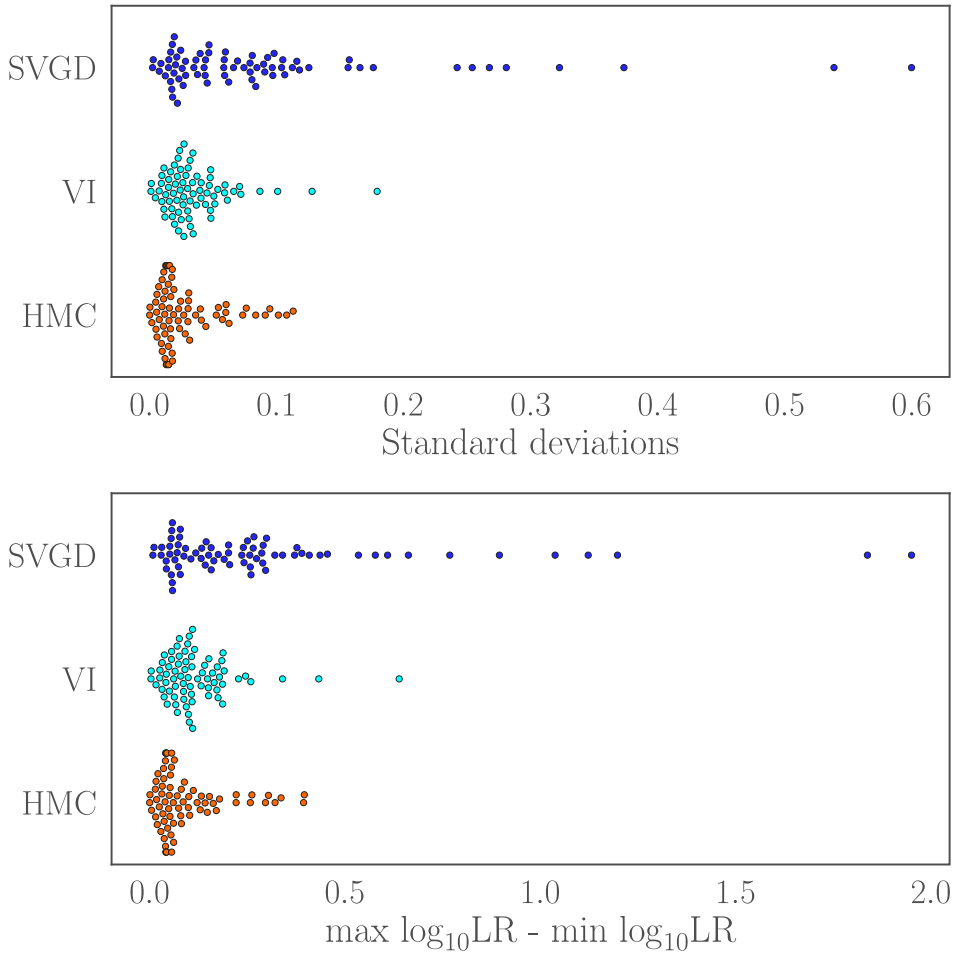

Fig. 4. Precision comparison of the tested algorithms both in terms of the $\log_{10}$ LR standard deviation (top) and the difference between the largest and smallest $\log_{10}$ LR (bottom) across 10 independent repetitions for each scenario (1 scenario = 1 dot).

4. Conclusions

We presented two variational Bayesian inference algorithms for probabilistic genotyping (PG): Stein variational gradient descent (SVGD) and variational inference with an evidence lower-bound objective (VI). These are applicable to PG models that are free of singularities. We have therefore also shown how to adapt the PG models of STRmix™ [2] to allow for variational inference. We then described the algorithms and explained their working principles and underlying assumptions.

We then validated the algorithms and checked the validity of the assumptions on the ProvedIT mixtures from the NIST comparative study [8] and compared with the Hamiltonian Monte Carlo (HMC) method, which was recently benchmarked on the same data set [9]. All three methods, HMC, SVGD, and VI were comparable in terms of accuracy with HMC slightly more accurate than the other two. This could, however, also be due to the different priors used in the

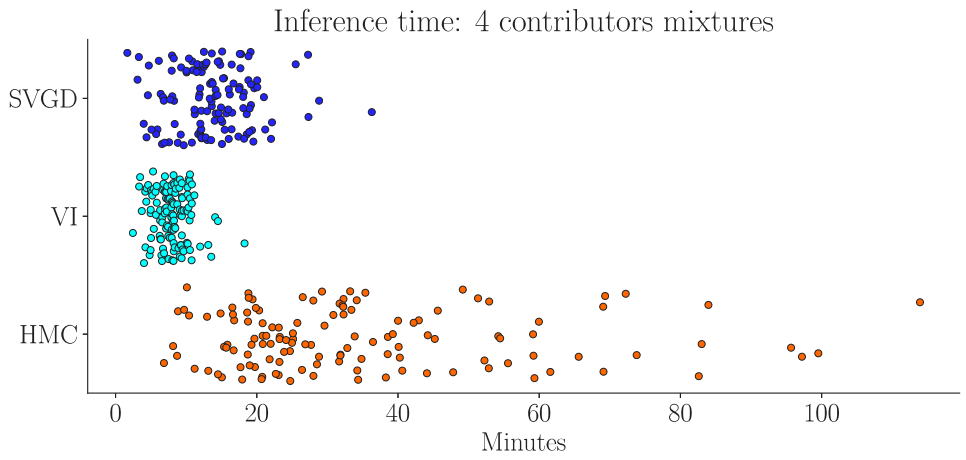

Fig. 5. Comparison of the computational runtime of the different algorithms on the same 4-contributor mixtures. Each dot represents a different scenario.

HMC model. Importantly, VI achieved significantly lower computational runtimes than both SVGD and HMC, while maintaining the high precision of HMC. It therefore could be a candidate for replacing MCMC-based algorithms in practice and provide better user experience due to faster runtimes. Faster runtimes would also enable laboratories to run independent repetitions of the inference in order to quantify reproducibility.

The run-to-run variability of all tested methods can be reduced at the expense of additional computational cost. For HMC, this can be achieved by performing more iterations. For SVGD, it can be done by using more particles and/or by lowering the learning rate. For VI, one could increase the size of the sample in the Monte Carlo scheme and/or reduce the learning rate over iterations. In all cases, however, this would increase the computational time of the inference, representing the typical precision/runtime trade-off. The key question therefore is which method offers the best trade-off while still maintaining high accuracy. The present results suggest that VI offers the best trade-off between accuracy, precision, and runtime.

While not outperforming in the benchmarks, SVGD is an algorithmically interesting method that offers ample opportunity for optimisation and links to established mathematical frameworks such as particle filters [15] for which efficient parallel software exists [16]. While our implementation of SVGD was significantly faster than HMC, it was overly sensitive to missing peaks as seen in case of the E05 mixture.

An important limitation of the present work is that our implementation of VI was limited to a multivariate Gaussian approximation of the posterior. Recently, it has been shown that it is possible to construct more general approximations using VI, theoretically even a universal one [17]. This greater approximation power is achieved by learning invertible distribution transformations, called *normalising flows*. We tested possible flow architectures and obtained promising results with inverse autoregressive flows [18]. Then, we often observed improved quality of the posterior estimation, but the method had two significant drawbacks: First, the computation time increased to become comparable with that of HMC. Second, we could not find a robust way to prevent gradient divergence, which happened occasionally when normalising flows were used.

Taken together, our results not only provide a way of accelerating inference in Bayesian PG without sacrificing much accuracy or precision, but they also open the field of research toward a more diverse range of inference algorithms beyond sampling-based MCMC methods. We hope this might trigger a discussion in the field and reinvigorate the search for better, more scalable, and mathematically founded algorithms for DNA mixture deconvolution in forensic genetics.

Declaration of competing interest

Authors have no conflict of interest to declare.

Appendix A. Supplementary data

Supplementary material related to this article can be found online at https://doi.org/10.1016/j.fsigen.2023.102890.

References

[1] M. Susik, H. Schönborn, I.F. Sbalzarini, Hamiltonian Monte Carlo with strict convergence criteria reduces run-to-run variability in forensic DNA mixture deconvolution, Forensic Sci. Int.: Genet. 60 (2022) 102744.
[2] D. Taylor, J.-A. Bright, J. Buckleton, The Interpretation of single source and mixed DNA profiles, Forensic Sci. Int.: Genet. 7 (5) (2013) 516–528.
[3] R. van de Schoot, S. Depaoli, R. King, B. Kramer, K. Märtens, M.G. Tadesse, M. Vannucci, A. Gelman, D. Veen, J. Willemsen, C. Yau, Bayesian statistics and modelling, Nat. Rev. Methods Primers 1 (1) (2021).
[4] D.M. Blei, A. Kucukelbir, J.D. McAuliffe, Variational inference: A review for statisticians, J. Amer. Statist. Assoc. 112 (518) (2017) 859–877.
[5] Q. Liu, D. Wang, Stein variational gradient descent: A general purpose Bayesian inference algorithm, in: D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, R. Garnett (Eds.), Advances in Neural Information Processing Systems, Vol. 29, Curran Associates, Inc., 2016.
[6] H. Kelly, J.-A. Bright, M. Kruijver, S. Cooper, D. Taylor, K. Duke, M. Strong, V. Beamer, C. Buettner, J. Buckleton, A sensitivity analysis to determine the robustness of STRmix™ with respect to laboratory calibration, Forensic Sci. Int.: Genet. 35 (2018) 113–122.
[7] D. Taylor, J. Buckleton, J.-A. Bright, Factors affecting peak height variability for short tandem repeat data, Forensic Sci. Int. Genet. 21 (2016) 126–133.
[8] S. Riman, H. Iyer, P.M. Vallone, Examining performance and likelihood ratios for two likelihood ratio systems using the PROVEDIt dataset, in: U. Qamar (Ed.), PLOS ONE 16 (9) (2021) e0256714.
[9] M. Susik, I.F. Sbalzarini, Analysis of the Hamiltonian Monte Carlo genotyping algorithm on PROVEDIt mixtures including a novel precision benchmark, Forensic Sci. Int.: Genet. 64 (2023) 102840.
[10] L.E. Alfonse, A.D. Garrett, D.S. Lun, K.R. Duffy, C.M. Grgicak, A large-scale dataset of single and mixed-source short tandem repeat profiles to inform human identification strategies: PROVEDIt, Forensic Sci. Int.: Genet. 32 (2018) 62–70.
[11] D.P. Kingma, J. Ba, Adam: A method for stochastic optimization, in: Y. Bengio, Y. LeCun (Eds.), 3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings, 2015.
[12] Ø. Bleka, G. Storvik, P. Gill, EuroForMix: An open source software based on a continuous model to evaluate STR DNA profiles from a mixture of contributors with artefacts, Forensic Sci. Int.: Genet. 21 (2016) 35–44.
[13] J.V. Dillon, I. Langmore, D. Tran, E. Brevdo, S. Vasudevan, D. Moore, B. Patton, A. Alemi, M. Hoffman, R.A. Saurous, TensorFlow distributions, 2017.
[14] J. Ba, M.A. Erdogdu, M. Ghassemi, S. Sun, T. Suzuki, D. Wu, T. Zhang, Understanding the variance collapse of SVGD in high dimensions, in: International Conference on Learning Representations, 2022.

[15] Ö. Demirel, I. Smal, W.J. Niessen, E. Meijering, I.F. Sbalzarini, Piecewise constant sequential importance sampling for fast particle filtering, in: Proc. 10th IET Conf. Data Fusion & Target Tracking, IET, Liverpool, UK, 2014.
[16] Ö. Demirel, I. Smal, W.J. Niessen, E. Meijering, I.F. Sbalzarini, PPF – A parallel particle filtering library, in: Proc. 10th IET Conf. Data Fusion & Target Tracking, IET, Liverpool, UK, 2014.
[17] C.-W. Huang, L. Dinh, A.C. Courville, Solving ODE with universal flows: Approximation theory for flow-based models, in: ICLR 2020, 2020.
[18] D.P. Kingma, T. Salimans, R. Jozefowicz, X. Chen, I. Sutskever, M. Welling, Improved variational inference with inverse autoregressive flow, in: D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, R. Garnett (Eds.), Advances in Neural Information Processing Systems, Vol. 29, Curran Associates, Inc., 2016.