Bioinformatics, 2026, X, 1–9
ISMB 2026

Neretva: Unsupervised Neural Variational Inference for Allele-level Genotyping of Highly Polymorphic Genes

Qinghui Zhou,[1] Seyed Pouria Ahmadi[1] and Ibrahim Numanagić[1,*]

[1]Department of Computer Science, University of Victoria, Victoria, V8P 5C2, B.C., Canada
*Corresponding author. inumanag@uvic.ca

Abstract

Accurate genotyping and phasing of highly polymorphic gene families are essential for precision medicine. Yet, the genotyping problem remains computationally challenging due to extreme sequence similarity between related genes, copy number variation, and structural complexity. Current methods typically rely on integer linear programming or maximum likelihood-based approaches that often suffer from scalability and flexibility issues. Here, we present Neretva, a new framework that models the genotyping problem as an instance of a hierarchical probabilistic model based on auto-encoding variational Bayes (AEVB). Leveraging the reparameterization strategy and neural inference networks, our approach scales efficiently to complex pharmacogene families such as CYP (cytochrome P450) and KIR (killer-cell immunoglobulin-like receptors), and achieves improved accuracy over the current state of the art on both CYP and KIR gene families. Neretva is open source and is freely available at `https://github.com/0xTCG/neretva`**.**

Introduction

The advance of high-throughput sequencing (HTS) technologies has transformed the landscape of genomic medicine [46]. A significant clinical advantage of HTS technologies lies in their promise to comprehensively and rapidly analyze genotypes of clinically important gene families (also known as *pharmacogenes*) that are critical to precision medicine decision making [58]. Key pharmacogenes include the cytochrome P450 (CYP) and killer-cell immunoglobulin-like receptor (KIR) gene families. The CYP gene family encodes enzymes responsible for the metabolism of more than 70% of clinically prescribed drugs, as well as their therapeutic efficacy [59, 22]. Meanwhile, KIR genes regulate natural killer cell and T-cell responses, and are linked, among other things, to HIV progression and autoimmune disorders [1, 28, 40]. More importantly, KIR genes are the key markers in determining the success of organ transplants [38], and in the context of hematopoietic stem cell transplantation, accurate KIR genotyping is needed because KIR-HLA interaction mismatches can precipitate severe complications with potentially fatal outcomes [16].

Despite their clinical importance, accurate genotyping of pharmacogene families from HTS data remains computationally challenging due to several factors. The first major challenge is the fact that pharmacogene alleles are defined on a haplotype level, which requires accurate gene phasing to establish a clinically actionable phenotype. The phasing process is often hindered by both extreme inter-gene sequence similarity between pharmacogenes and their paralogs (that often exist within these clusters [43]), as well as the intra-gene allelic identity that often exceeds >98%, which cause large levels of ambiguity during assigning individual HTS reads to their true allelic origin [5, 42]. Most pharmacogene families also exhibit substantial population-specific variation, and novel alleles are frequently discovered in understudied populations [49, 57], rendering the strict reliance on the previously published genotypes and phases insufficient. Furthermore, some pharmacogenes exhibit extensive copy number variation and are subject to complex structural variations that include gene deletions, duplications, and hybrid formations (such as paralog-related gene fusions) [13]. Other clusters exhibit even higher levels of complexity; for example, 17 genes in the KIR cluster are organized into two groups consisting of relatively conserved A haplotypes and the highly diverse B haplotypes enriched for activating receptors that exhibit extensive copy number variation and cross-gene structural rearrangements, with the number of genes ranging from 4 to 14 per haplotype [23, 24]. Finally, we note that other clinically important gene families present similar, if not even greater challenges; examples include *HLA* genes, immunoglobulin loci, T-cell receptor genes (*TRA*, *TRB*), and other immune receptor families such as *FCGR* and *LILR* [11, 2, 17, 19, 36, 52].

Many specialized tools were developed with the aim of addressing the mentioned challenges and making HTS data useful for accurate pharmacogene genotyping. Tools such as Stargazer [33], Cyrius [8], StellarPGx [54], PyPGx [34], Aldy [42, 18] and Astrolabe [55], combine the existing databases such as PharmVar [14], as well rule-based strategies and various statistical and combinatorial techniques to accurately infer the genotype from HTS data for wide variety of pharmacogenes. As for the more complex KIR gene cluster, tools such as T1K [50], Locityper [45], Geny [60], kir-mapper [6] and PING [39] typically rely on combinatorial and likelihood-based assignment strategies to maximize the log-likelihood of observed read data and estimate candidate allele abundances, together with various pre-processing schemes such as k-mer or minimizer-based

© The Author(s) 2026. Copyright and licence statements to be updated by the publisher during production.

fingerprinting approaches. Other gene clusters are also handled by specialized tools; some examples include OptiType [52] and HLAreporter [20] for HLA genes, Immunotyper-SR for the IGH cluster [11, 12], and the recent pangenome-based approaches such as PGR-TK [9].

Currently, one of the most successful and robust genotyping strategies follows the combinatorial optimization approach that models the genotyping problem as an instance of one or more integer linear programs (ILPs). This is supported by the performance of tools such as Aldy (CYP), Geny (T1K), ImmunoTyper-SR (IGH), OptiType (HLA), as well as the recent pangenome-based approach PHI [7]. For example, Aldy was shown to consistently produce good results across many different HTS technologies (and sequencing depths) in various contexts [41, 53, 56], often achieving up to 99% overall genotyping accuracy. In these tools, the candidate allele set is first filtered by eliminating alleles with low abundance (through a rule-based or statistical-based approach), and the optimal allele set is chosen by solving one or more constrained integer linear programs, often composed into a multi-stage process where each stage handles a different aspect of the genotyping problem (e.g., Aldy uses separate ILPs for structural variant detection and for determining the functional genotypes). Although such approaches guarantee a theoretical global optimum and constraint enforcement, and are unlikely to converge to a local optimum (a common issue with likelihood-based approaches), they are often NP-hard [26, 42]. Consequently, the implementations often do not scale well to new instances and complex gene families. While such models are in practice solvable for CYP genes (where the model instances have a limited set of variables), the larger pharmacogene families, such as KIR, lead to a combinatorial explosion that precludes directly inheriting ILP's optimality guarantees at practical runtime. Furthermore, ILP-based approaches rely on black-box (and often proprietary) solvers such as Gurobi or Google OR-Tools, which make their use impractical in specialized clinical applications where the patient data is handled through privacy-preserving technologies (e.g., homomorphic encryption). Lastly, ILP solutions are often rigid, and establishing the confidence of an ILP solution, or an acceptable neighbourhood, is far from trivial.

The above limitations motivate a probabilistic ILP relaxation that can enable scalable gradient-based optimization while naturally accommodating uncertainty. Recent advances in variational inference, such as *auto-encoding variational Bayes* (AEVB) and the reparameterization strategy [30], have shown remarkable success in high-dimensional problems with a similar latent structure to the genotyping problem. In natural language processing, one such example is NMF-fashioned LDA [32, 4], while neural variational models such as ProdLDA [51] and ETM [10] have scaled topic modelling to millions of documents and thousands of topics. In genomics, deep learning approaches have also shown promise: CAECseq [27] applied convolutional autoencoders to haplotype assembly and viral quasispecies reconstruction, demonstrating that neural networks can effectively cluster sequencing reads originating from highly similar genomic components. All these methods leverage GPU-accelerated gradient-based optimization and inference networks to achieve orders-of-magnitude speedup over traditional approaches while maintaining or improving accuracy. Beyond scalability, AEVB methods also offer flexibility for incorporating domain constraints as differentiable regularizers, thus avoiding the need to re-derive update equations (as in likelihood-based strategies) or redesign samplers (as in MCMC), which is desirable for large-scale variational inference problems like KIR genotyping.

In this paper, we present a new unified AEVB-based framework, called Neretva, for genotyping both CYP and KIR gene families. Our approach combines the variant-centric modelling philosophy of Aldy with the scalability of modern variational inference, and achieves several key innovations: (1) learnable variant emission probabilities that can learn novel alleles, (2) explicit bias modelling for handling cross-gene mapping ambiguities in highly homologous regions, and (3) GPU-accelerated neural network-based inference that enables analysis of complex samples in minutes rather than hours. We demonstrate that Neretva achieves high accuracy, precision and recall across both gene families, with F_1 scores reaching up to 1.0 in CYP and 0.912 in KIR experiments, thus matching or surpassing the current state of the art while maintaining the interpretability of the results essential for precision medicine applications. Thus, we hope that Neretva will serve as a sound and practical solution for employing HTS data to genotype complex pharmacogenes in both research and clinical settings.

Methods

Given a pharmacogene family and a collection of sequencing reads in BAM/CRAM/FASTQ, Neretva aims to determine the number of copies of each gene within that family, and to reconstruct the exact sequence content (haplotype) of each such copy (in other words, assign an allele to it), thus determining the overall genotype of the given family.

Database Preparation

Consider a multi-gene model where the gene set $\mathcal{G} = \{G_1, \ldots, G_L\}$ represents a family of $L \geq 1$ genes (e.g., KIR cluster consists of $L = 17$ genes). Each gene $G \in \mathcal{G}$ defines a *reference allele* with no variants. All other alleles are defined by a set of variants that differentiates them from the reference allele.

Let us define a set $\mathcal{M}_G$ of all possible variants occurring within gene G as $\mathcal{M}_G = \{m_1^G, \ldots, m_{|\mathcal{M}_G|}^G\}$, and partition it into $\mathcal{M}_G^+$ consisting of *core variants* (that impact the drug metabolism or a protein function of G), and $\mathcal{M}_G^\times$ that contain *minor variants* (silent variants that have no known functional impact).

Let $\mathcal{A}^G$ denote the set of known alleles for gene G. Each allele $a_i^G \in \mathcal{A}^G$ is characterized by sets of core and minor variants that differentiate its sequence from the reference allele (often referred to as *1 allele). In other words, the set of variants $\mathcal{M}_{a_i^G}$ of allele $a_i^G \in \mathcal{A}^G$ consists of a core variant set $\mathcal{M}_{a_i^G}^+$ and a minor variant set $\mathcal{M}_{a_i^G}^\times$. We will later refer to $\mathcal{M}_{a_i^G}$ as allele a_i^G's *original variant set*, and define $\mathcal{M} = \bigcup_{G \in \mathcal{G}} \mathcal{M}_G$ as a set of all variants being considered.

For successful genotyping, it is essential to construct a database of previously known alleles that includes the known set of alleles for each gene of interest, along with the variant and structural variation information. For KIR genes, we

acquire the allele sequence contents from the IPD-KIR database v2.12.0 [48], while for other genes, we use the PharmVar database v6.2 [14]. In both databases, allele names roughly follow the star-allele nomenclature [47], where the major allele designation (typically represented as number; e.g., *CYP2D6*2) is defined by core variants and describes the allele's functionality, while minor allele designation (often denoted as a letter suffix; e.g., *2A) describes variants with no known functional impact.

Alignment and Filtering

Let us take an HTS sample that contains reads $\mathcal{R} = \{r_1, \ldots, r_{|\mathcal{R}|}\}$. For CYP genes, we assume the read mapping is correct and extract the reads from a given sample (in BAM/CRAM format) as-is. In the case of KIR, where many reads can map to different genes equally well, we align reads directly against all known allele sequences with minimap2 [35] in all-to-all mode[1]. Then, for each $r \in \mathcal{R}$, we gather a set of alleles $\mathcal{A}_r$ to which r maps, and a set of variants $\mathcal{M}_r \subseteq \mathcal{M}$ that r supports. To reduce the solution space, we filter down the allele set $\mathcal{A}$ into a subset $\mathcal{A}'$ so that each $A \in \mathcal{A}'$ has each $m \in \mathcal{M}_A^+$ covered by at least 3 reads.

In gene clusters such as KIR, incorrect cross-gene alignments are common, and it is not clear if an observed variant m_k^G is indeed a true variant in G or a *shadow variant* caused by a mapping artifact. An incorrect variant assignment can result in an incorrect allele call, which, in turn, can lead to a wrong medical intervention. Hence, for each read r that induces a variant $m_k^G \in \mathcal{M}_r$, we check whether there exists an allele $a_i^{G'} \in \mathcal{A}_r$ such that $G' \neq G$; if that happens, we extend $\mathcal{M}_{a_i^{G'}}$ to include m_k^G as well. This allows us to construct an *extended variant set* $\mathcal{V}_{a_i^{G'}}$ for each allele $a_i^{G'}$, which contains both true G' variants and the shadow variants that stem from incorrect alignments. This extension has no biological implications; it is used solely to guide subsequent optimization steps towards the correct solution.

Allele Identification Problem

Let us define, for each gene G, a list of positions $\mathcal{P}^G = \{p_1^G, \ldots, p_m^G\}$, that describe genomic coordinates which can harbor a mutation $m_k^G \in \mathcal{M}_G$, and the set of all positions as $\mathcal{P} = \bigcup_{G \in \mathcal{G}} \mathcal{P}^G$. We also define a set of *bases* $\mathcal{B} = \{A, C, T, G, N, P\}$, where A, C, T, G stand for the four possible nucleotides, while N and P denote deletions and insertions, respectively. (Note that base is not the same as variant; the reference allele still contains bases despite having no variants.)

After the alignment, we observe a base count matrix $\mathbf{Y} \in \mathbb{N}^{|\mathcal{P}| \times |\mathcal{B}|}$ collected from the read set $\mathcal{R}$, where each entry $\mathbf{Y}_{p,b}$ denotes the observed count for base $b \in \mathcal{B}$ at position $p \in \mathcal{P}$:

$$\mathbf{Y} = \begin{pmatrix} Y_{p_1,A} & Y_{p_1,C} & Y_{p_1,T} & Y_{p_1,G} & Y_{p_1,N} & Y_{p_1,P} \\ Y_{p_2,A} & Y_{p_2,C} & Y_{p_2,T} & Y_{p_2,G} & Y_{p_2,N} & Y_{p_2,P} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ Y_{p_m,A} & Y_{p_m,C} & Y_{p_m,T} & Y_{p_m,G} & Y_{p_m,N} & Y_{p_m,P} \end{pmatrix}.$$

Note that we cannot assume that the allele database can solely explain $\mathbf{Y}$ due to the possibility of novel (i.e., uncataloged) mutations (and alleles) in a given sample [3, 61]. Thus, to identify the set of alleles that best describes $\mathbf{Y}$, we need to determine the respective base for each allele $a_k^G \in \mathcal{A}$ at each $p \in \mathcal{P}$ (if p describes any variant in a_k^G's *extended variant set*). Each such inference is modelled as a latent variable; more precisely, $\boldsymbol{\theta} \in \Delta^K$ represents the relative proportion of each of $K = |\mathcal{A}'|$ candidate alleles, where Δ^K denotes a $K-1$ simplex. $\boldsymbol{\Psi} = \{\psi_{a,p} \mid a \in \mathcal{A}', p \in \mathcal{P}\}$ represents the base emission probability vectors, where each $\psi_{a,p} \in \Delta^{|\mathcal{B}|}$ is a distribution over bases $\mathcal{B}$ at position p for an allele a.

Copy Number Estimation

The allele identification determines relative allele proportions $\boldsymbol{\theta}$ on the simplex, thus revealing which alleles are present, as well as their relative abundances. However, before determining an exact count of each allele of interest, we first need to estimate the underlying gene copy numbers, as alleles are constrained by the copy number of their parent genes.

Let us formulate the gene copy number estimation as a coverage reconstruction problem. We observe aggregated coverage values across C genomic regions (typically representing introns and exons), where each region's coverage c_r represents the mean read depth normalized by the expected coverage depth. The observed coverage is modelled as a linear combination of unknown gene copy numbers. Let $\mathbf{c} \in \mathbb{R}^C$ denote the observed region coverage vector, $\mathbf{S} \in \mathbb{R}^{L \times C}$ denote the gene-to-region contribution matrix where element $S_{g,r}$ represents the expected coverage contribution rate from gene G_g to region r. Let $\mathbf{n} \in \mathbb{R}^L$ denote the unknown copy numbers for L genes such that $0 \leq n_g \leq n_{\max}$. For KIR genes, we set $n_{\max} = 5$, while for CYP, we use $n_{\max} = 4$. Then, the expected coverage for each region is given by:

$$\hat{c}_r = \sum_{g=1}^{L} n_g \cdot S_{g,r}.$$

We estimate copy numbers via linear regression by minimizing the following error:

$$\mathcal{L}(\mathbf{n}) = \frac{1}{C} \sum_{r=1}^{C} H_\delta(c_r - \hat{c}_r),$$

where H_δ is the Huber loss [21] with threshold δ that provides robustness to outliers from coverage anomalies. For CYP genes, we used $\delta = 0.5$, and for KIR, $\delta = 0.3$. To obtain the final copy number, we optimize the above loss using Adam [29] optimizer with 1000 iterations.

Bias Factor Matrix

While many HTS technologies in theory promise uniform coverage, coverage of some bases might not match the expected coverage due to sequencing biases or shadow variants and other mapping artifacts. To model such biases, we define a *bias matrix* $\mathbf{H} \in \mathbb{R}^{K \times M}$ for each $a \in \mathcal{A}'$ and $v \in \mathcal{V}_a$ (where $M = |\mathcal{P}| \times |\mathcal{B}|, K = |\mathcal{A}'|$). Its each element $h_{a,m}$ represents the expected coverage for a variant $m \in \mathcal{M}_a$.

For variants that are part of an allele's original variant set, the bias factor is initialized to single copy coverage, or, when available, derived from a predetermined profile that accounts for known sequencing coverage biases. For shadow variants, we assign bias factors based on empirical misalignment rates obtained from the simulated alignment data. The bias matrix

[1] By using `-x sr --secondary=yes -c --dual=no` parameters.

is sparse because it has no entries for positions where an allele neither contains a true or a shadow variant. This bias adjustment allows the model to appropriately amplify the signal from the true variants.

Allele Identification Model

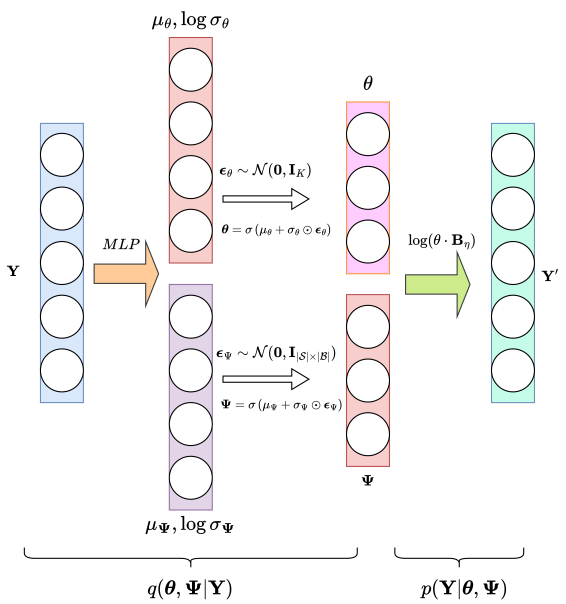

Figure 1 Network structure of inference model $q(\boldsymbol{\theta}, \boldsymbol{\Psi}|\mathbf{Y})$ and generative model $p(\mathbf{Y}|\boldsymbol{\theta}, \boldsymbol{\Psi})$.

Let us describe the generative process for the count matrix $\mathbf{Y}$ collected from a sequencing sample as follows. Let $\mathcal{LN}(\mu, \Sigma)$ be a logistic normal distribution. For clarity of notation, we define $\mathcal{S}$ as the set of all allele-position pairs such that $\mathcal{S} = \{(a, p) \mid a \in \mathcal{A}', p \in \mathcal{P}\}$, where each element $s \in \mathcal{S}$ represents a specific allele-position pair, thus specifying the generation process as follows:

1. draw allele proportions $\boldsymbol{\theta} \sim \mathcal{LN}(\mathbf{0}, \mathbf{I})$;
2. for each allele position $s \in \mathcal{S}$, draw base emissions $\boldsymbol{\psi}_s \sim \mathcal{LN}(\boldsymbol{\mu}_s, \boldsymbol{\Sigma}_s)$;
3. for the observed mutation counts matrix, (a) compute mutation probabilities $\boldsymbol{\pi} \propto \boldsymbol{\theta}^T \boldsymbol{B}_\eta$, and (b) draw counts $\mathbf{Y} \sim \text{Multinomial}(N, \boldsymbol{\pi})$.

Here, $\boldsymbol{\Sigma}_s$ is assumed to be a diagonal covariance matrix, and $\mathbf{B} \in \mathbb{R}^{K\times M}$ denotes the matrix of all the allele base emission probability vectors, $\mathbf{H} \in \mathbb{R}^{K\times M}$ is the bias matrix containing the bias factors, and $\mathbf{B}_\eta = \mathbf{B} \odot \mathbf{H}$ is the strength-adjusted emission matrix. N denotes the total observed variant counts across all positions $\mathcal{P}$ and $N = \sum_{p,b\in\mathbf{Y}} Y_{p,b}$. The database-informed priors $\boldsymbol{\mu}_s$ encode confidence in known variants from the reference database. Under this generative procedure, the marginal likelihood of the observed data is:

$$p(\mathbf{Y}) = \int_\theta \int_\Psi p(\mathbf{Y}|\boldsymbol{\theta}, \boldsymbol{\Psi})\, p(\boldsymbol{\theta})\, p(\boldsymbol{\Psi})\, d\boldsymbol{\theta}\, d\boldsymbol{\Psi}. \tag{1}$$

Variational Inference

The marginal likelihood in Equation (1) is intractable due to the high-dimensional latent space. Thus, we employ variational inference with the SGVB estimator [30] to approximate the posterior $p(\boldsymbol{\theta}, \boldsymbol{\Psi}|\mathbf{Y})$ for each sample independently. For computational efficiency, we adopt a structured mean-field variational approximation [25]:

$$q(\boldsymbol{\theta}, \boldsymbol{\Psi}|\mathbf{Y}) = q(\boldsymbol{\theta}|\mathbf{Y}) \prod_{s\in\mathcal{S}} q(\boldsymbol{\psi}_s|\mathbf{Y}). \tag{2}$$

Deriving the Evidence Lower Bound.
Let us derive the evidence lower bound (ELBO) for our variational inference objective. From the log-marginal likelihood:

$$\log p(\mathbf{Y}) = \log \int_\theta \int_\Psi p(\mathbf{Y}|\boldsymbol{\theta}, \boldsymbol{\Psi}) p(\boldsymbol{\theta}) p(\boldsymbol{\Psi})\, d\boldsymbol{\theta}\, d\boldsymbol{\Psi},$$

we introduce the variational distribution $q(\boldsymbol{\theta}, \boldsymbol{\Psi}|\mathbf{Y})$ and apply Jensen's inequality:

$$\begin{aligned}\log p(\mathbf{Y}) &= \log \int_\theta \int_\Psi p(\mathbf{Y}|\boldsymbol{\theta}, \boldsymbol{\Psi}) p(\boldsymbol{\theta}) p(\boldsymbol{\Psi}) \frac{q(\boldsymbol{\theta}, \boldsymbol{\Psi}|\mathbf{Y})}{q(\boldsymbol{\theta}, \boldsymbol{\Psi}|\mathbf{Y})} d\boldsymbol{\theta}\, d\boldsymbol{\Psi} \\ &= \log \mathbb{E}_{q(\boldsymbol{\theta}, \boldsymbol{\Psi}|\mathbf{Y})} \left[\frac{p(\mathbf{Y}|\boldsymbol{\theta}, \boldsymbol{\Psi}) p(\boldsymbol{\theta}) p(\boldsymbol{\Psi})}{q(\boldsymbol{\theta}, \boldsymbol{\Psi}|\mathbf{Y})} \right] \\ &\geq \mathbb{E}_{q(\boldsymbol{\theta}, \boldsymbol{\Psi}|\mathbf{Y})} \left[\log \frac{p(\mathbf{Y}|\boldsymbol{\theta}, \boldsymbol{\Psi}) p(\boldsymbol{\theta}) p(\boldsymbol{\Psi})}{q(\boldsymbol{\theta}, \boldsymbol{\Psi}|\mathbf{Y})} \right].\end{aligned}$$

After expanding the logarithm, we obtain:

$$\begin{aligned}\log p(\mathbf{Y}) \geq\ & \mathbb{E}_q[\log p(\mathbf{Y}|\boldsymbol{\theta}, \boldsymbol{\Psi})] + \mathbb{E}_q[\log p(\boldsymbol{\theta})] \\ & + \mathbb{E}_q[\log p(\boldsymbol{\Psi})] - \mathbb{E}_q[\log q(\boldsymbol{\theta}, \boldsymbol{\Psi}|\mathbf{Y})].\end{aligned}$$

Under the mean-field approximation as specified in (2), we get:

$$\mathbb{E}_q[\log q(\boldsymbol{\theta}, \boldsymbol{\Psi}|\mathbf{Y})] = \mathbb{E}_q[\log q(\boldsymbol{\theta}|\mathbf{Y})] + \sum_{s\in\mathcal{S}} \mathbb{E}_q[\log q(\boldsymbol{\psi}_s|\mathbf{Y})]$$

In a similar vein, for the prior $p(\boldsymbol{\Psi}) = \prod_{s\in\mathcal{S}} p(\boldsymbol{\psi}_s)$, we get:

$$\mathbb{E}_q[\log p(\boldsymbol{\Psi})] = \sum_{s\in\mathcal{S}} \mathbb{E}_q[\log p(\boldsymbol{\psi}_s)].$$

Finally, by rearranging the terms, we obtain the final ELBO:

$$\begin{aligned}\mathcal{L} = \mathbb{E}_q[\log p(\mathbf{Y}|\boldsymbol{\theta}, \boldsymbol{\Psi})] - D_{\mathsf{KL}}[q(\boldsymbol{\theta}|\mathbf{Y}) \| p(\boldsymbol{\theta})] \\ - \sum_{s\in\mathcal{S}} D_{\mathsf{KL}}[q(\boldsymbol{\psi}_s|\mathbf{Y}) \| p(\boldsymbol{\psi}_s)].\end{aligned} \tag{3}$$

To optimize this objective, we parameterize the variational posterior using an encoder network with parameters ϕ:

$$(\boldsymbol{\mu}_\theta, \log \boldsymbol{\sigma}_\theta^2, \boldsymbol{\mu}_\Psi, \log \boldsymbol{\sigma}_\Psi^2) = \text{EncoderNeuralNet}_\phi(\mathbf{Y}).$$

To enable gradient-based optimization through the stochastic layers, apply the reparameterization strategy [30]. First, sample

noise from standard Gaussians:

$$\boldsymbol{\epsilon}_\theta \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_K);$$
$$\boldsymbol{\epsilon}_\Psi \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_{|\mathcal{S}|\times|\mathcal{B}|}).$$

Then, transform these to obtain samples from the variational posterior:

$$\boldsymbol{\theta} = \sigma\left(\mu_\theta + \boldsymbol{\sigma}_{\boldsymbol{\theta}} \odot \boldsymbol{\epsilon}_\theta\right);$$
$$\boldsymbol{\Psi} = \sigma\left(\mu_\Psi + \boldsymbol{\sigma}_{\boldsymbol{\Psi}} \odot \boldsymbol{\epsilon}_\Psi\right).$$

This yields the variational posteriors:

$$q(\boldsymbol{\theta}|\mathbf{Y}) = \mathcal{LN}(\mu_\theta, \mathrm{diag}(\sigma_\theta^2))$$
$$q(\boldsymbol{\Psi}|\mathbf{Y}) = \mathcal{LN}(\mu_\Psi, \mathrm{diag}(\sigma_\Psi^2)),$$

where $\sigma(\cdot)$ denotes the softmax function. This reparametrization allows us to optimize the ELBO using standard backpropagation with the SGVB estimator [30] via a single Monte Carlo sample.

Modified Objective with Biological Regularization

Although the above ELBO optimization produces a valid data fit across all variants, it does not prioritize core variants that capture the biological phenotype of each selected allele. To address this issue, let us introduce the set of all core variants $\mathcal{M}_+ = \bigcup_{G\in\mathcal{G}} \mathcal{M}_+^G$ over all genes $G \in \mathcal{G}$. Now, consider two probability vectors over $\mathcal{M}_+$: the empirical functional variant count distribution $\hat{p} \in \Delta^{|\mathcal{M}_+|}$ derived from $\mathbf{Y}$ and the reconstructed functional variant count distribution $\tilde{\pi} \in \Delta^{|\mathcal{M}_+|}$ derived from our inference model. We can measure the discrepancy between $\hat{p}$ and $\tilde{\pi}$ via the Jensen–Shannon divergence [37]. This measure is symmetrical and, hence, allows us to penalize both the lack and the surplus of core variants.

Denote $\mathrm{D_{JS}}(\hat{p}\|\tilde{\pi})$ as the Jensen–Shannon divergence between empirical $\hat{p}$ and the estimated core variant count distribution $\tilde{\pi}$. We also regularize the base emission distributions $\boldsymbol{\Psi}$ with an entropy penalty. Since each allele possesses a single true base at each genomic position and sequencing error rates are low (typically less than 1% for Illumina platforms), we expect $\boldsymbol{\Psi}$ to be sharply peaked rather than diffused. We use an entropy penalty $H(\boldsymbol{\Psi}) = \mathbb{E}_{s\in\mathcal{S}}[H(\boldsymbol{\psi_s})]$ to encourage the model to commit to a dominant base at each position, and acquire the augmented loss term as:

$$\tilde{\mathcal{L}} = -\mathcal{L} + \beta \cdot D_{\mathrm{JS}}(\hat{p}\|\tilde{\pi}) + \gamma \cdot H(\boldsymbol{\Psi}).$$

The hyperparameters β, γ are balancing terms for the final loss term. We minimize the augmented loss with Adam [29] for 3000 iterations.

Results

We chose three gene families in the increasing order of complexity to comprehensively benchmark Neretva. The first family, CYP2C, consists of three genes (*CYP2C8*, *CYP2C9*, and *CYP2C19*) whose alleles are defined solely in terms of variants, and which do not exhibit any copy number changes, structural variants or mapping ambiguities. The other family, CYP2D, consists of *CYP2D6* and its paralog *CYP2D7*. Here, both copy number changes and structural variants are possible. Finally, the KIR family contains 17 genes and, in addition to copy number and structural variants, exhibits high levels of ambiguous mappings.

The CYP families are evaluated on 70 GeT-RM Illumina WGS samples with an average sequencing depth of 30× [18]. All ground truths were previously validated by the GeT-RM consortium and augmented by the recently published updates [15, 44, 18]. This dataset contains, in addition to standard haploid genotypes, a sizeable number of *CYP2D6* copy number variants (both deletions and duplications), as well as paralog-related fusions. Besides Neretva, we also benchmarked the current state-of-the-art CYP genotype callers: StellarPGX (v1.2.7), Aldy (v4.8.2), Cyrius (v1.1.1), Stargazer (v1.0.9), Astrolabe (v0.8.7.2) and PyPGx (v0.25.0).

As for the KIR gene family, we benchmarked Neretva against the following KIR genotypers: Geny (v1.0), Locityper (v1.2.3), T1K (v1.0.9), kir-mapper (v1.01) and PING (commit `b4b6957`) on 40 Human Pangenome Reference Consortium (HPRC) samples [36]. The KIR ground truth calls for the HPRC samples were obtained from the HPRC assemblies [36] via the BAKIR annotation tool [12].

We evaluated the performance of each tool and gathered both sample-level and allele-level metrics consisting of the number of correct calls, accuracy, precision, recall and F_1 score. On sample level, a call is considered as correct if both (1) allele calls are concordant with ground truth annotation[2], and (2) copy number is correct. Neretva always produces a unique solution; for tools that produce multiple solutions for a sample, we mark a call as correct if any of the produced solutions is concordant. The accuracy of each tool is defined as the number of true positives divided by the number of samples, while the precision is defined as the number of true positives divided by the number of true and false positives. The recall is defined as number of true positives divided by the number of true positives and false negatives.

CYP gene family performance.

As shown in Table 1, Neretva was able to consistently perfectly genotype simple pharmacogenes such as *CYP2C8*, *CYP2C9*, and *CYP2C19*, and matched the state of the art, such as Aldy, Astrolabe, StellarPGX and PyPGx.

As for *CYP2D6*, Neretva achieved both sample-level and allele-level F_1 of 0.99, putting it on par with Aldy (the second-best tool) and just one call below Cyrius, which achieved perfect accuracy. After further investigation, we observed that our model miscalled *2/*36+*10 as *2/*36+*2 due to the misidentification of a single variant distinguishing *2 and *10. While Aldy was able to solve this case, it failed on another instance with an increased copy number that Neretva was able to handle successfully.

KIR gene family performance.

As shown in Table 2, Neretva offers improved performance across the evaluated KIR samples in nearly all categories. At the sample level, Neretva achieved the highest accuracy (0.842) and F_1 (0.908), substantially outperforming Geny, the second-best

[2] Concordance means that the allele call matches the ground truth at functional level; in case of CYP genes, that implies the major star-allele match (e.g., *2A is regarded as *2), while for KIR genes, the match of the first 3 digits within the allele name (e.g., *0030201 is regarded as *003).

Table 1. Performance comparison between Neretva and other tools on CYP genes. The best results are highlighted in bold.

CYP2C8	**Per Call (Sample-Level)**						**Per Allele**					
Tool	n	**Correct**	**Accuracy**	**Precision**	**Recall**	F_1	n	**Correct**	**Accuracy**	**Precision**	**Recall**	F_1
Aldy	70	**70**	**1.000**	**1.000**	**1.000**	**1.000**	140	**140**	**1.000**	**1.000**	**1.000**	**1.000**
Stargazer	**70**	68	0.971	0.971	**1.000**	0.986	140	138	0.986	0.986	0.986	0.986
PyPGx	70	**70**	**1.000**	**1.000**	**1.000**	**1.000**	140	**140**	**1.000**	**1.000**	**1.000**	**1.000**
Astrolabe	70	**70**	**1.000**	**1.000**	**1.000**	**1.000**	140	**140**	**1.000**	**1.000**	**1.000**	**1.000**
Neretva	70	**70**	**1.000**	**1.000**	**1.000**	**1.000**	140	**140**	**1.000**	**1.000**	**1.000**	**1.000**
CYP2C9	**Per Call (Sample-Level)**						**Per Allele**					
Tool	n	**Correct**	**Accuracy**	**Precision**	**Recall**	F_1	n	**Correct**	**Accuracy**	**Precision**	**Recall**	F_1
Aldy	70	**70**	**1.000**	**1.000**	**1.000**	**1.000**	140	**140**	**1.000**	**1.000**	**1.000**	**1.000**
StellarPGx	70	**70**	**1.000**	**1.000**	**1.000**	**1.000**	140	**140**	**1.000**	**1.000**	**1.000**	**1.000**
Stargazer	70	68	0.971	0.971	**1.000**	0.986	140	138	0.986	0.986	0.986	0.986
PyPGx	70	68	0.971	0.971	**1.000**	0.986	140	138	0.986	0.986	0.986	0.986
Astrolabe	70	67	0.957	0.957	**1.000**	0.978	140	137	0.979	0.979	0.979	0.979
Neretva	70	**70**	**1.000**	**1.000**	**1.000**	**1.000**	140	**140**	**1.000**	**1.000**	**1.000**	**1.000**
CYP2C19	**Per Call (Sample-Level)**						**Per Allele**					
Tool	n	**Correct**	**Accuracy**	**Precision**	**Recall**	F_1	n	**Correct**	**Accuracy**	**Precision**	**Recall**	F_1
Aldy	70	69	0.986	0.986	**1.000**	0.993	140	139	0.993	0.993	0.993	0.993
StellarPGx	70	69	0.986	0.986	**1.000**	0.993	140	139	0.993	0.993	0.993	0.993
Stargazer	70	69	0.986	0.986	**1.000**	0.993	140	139	0.993	0.993	0.993	0.993
PyPGx	70	**70**	**1.000**	**1.000**	**1.000**	**1.000**	140	**140**	**1.000**	**1.000**	**1.000**	**1.000**
Astrolabe	70	**70**	**1.000**	**1.000**	**1.000**	**1.000**	140	**140**	**1.000**	**1.000**	**1.000**	**1.000**
Neretva	70	**70**	**1.000**	**1.000**	**1.000**	**1.000**	140	**140**	**1.000**	**1.000**	**1.000**	**1.000**
CYP2D6	**Per Call (Sample-Level)**						**Per Allele**					
Tool	n	**Correct**	**Accuracy**	**Precision**	**Recall**	F_1	n	**Correct**	**Accuracy**	**Precision**	**Recall**	F_1
ALDY	70	69	0.986	0.986	**1.000**	0.993	162	159	0.981	0.988	0.981	0.985
StellarPGX	70	66	0.943	0.943	**1.000**	0.971	162	157	0.969	0.981	0.969	0.975
Cyrius	70	**70**	**1.000**	**1.000**	**1.000**	**1.000**	162	**162**	**1.000**	**1.000**	**1.000**	**1.000**
Stargazer	70	40	0.571	0.571	**1.000**	0.727	162	123	0.759	0.879	0.759	0.815
PYPGX	70	37	0.529	0.529	**1.000**	0.692	162	122	0.753	0.871	0.753	0.808
Astrolabe	70	36	0.514	0.514	**1.000**	0.679	162	120	0.741	0.857	0.741	0.795
Neretva	70	69	0.986	0.986	**1.000**	0.993	162	161	0.994	0.994	0.994	0.994

Table 2. Performance comparison between Neretva and other tools on KIR genes. The best results are highlighted in bold.

KIR	**Per Call (Sample-level)**						**Per Allele**					
Tool	n	**Correct**	**Accuracy**	**Precision**	**Recall**	F_1	n	**Correct**	**Accuracy**	**Precision**	**Recall**	F_1
PING	442	229	0.518	0.577	0.640	0.607	749	464	0.619	0.699	0.619	0.657
T1K	442	230	0.520	0.482	**1.000**	0.651	749	539	0.720	0.744	0.720	0.732
Geny	442	352	0.796	0.800	**0.986**	0.883	749	647	0.864	0.906	0.864	0.884
kir-mapper	442	277	0.627	0.778	0.747	0.762	749	580	0.774	0.866	0.774	0.817
Locityper	442	59	0.133	0.107	0.431	0.172	749	343	0.458	0.311	0.458	0.371
Neretva	442	**372**	**0.842**	**0.845**	0.982	**0.908**	749	**682**	**0.911**	**0.914**	**0.911**	**0.912**

method, which reached a sample accuracy of 0.796. kir-mapper demonstrated competitive precision (0.778) but a noticeably lower recall (0.747), indicating reduced sensitivity. In contrast, Locityper and T1K performed less optimal on this dataset, with sample accuracies of 0.133 and 0.482, respectively [3].

At the allele level, Neretva correctly identified 682 out of 749 alleles, achieving an F_1 score of 0.912, thus improving on Geny (allele-level F_1 score: 0.884) and T1K (allele-level F_1 score: 0.732). These results suggest that Neretva more effectively handles the high sequence similarity and read-mapping ambiguity characteristic of the KIR region, thereby reducing the false positive rates observed in other genotyping tools.

Computational efficiency.

Similar to other tools, Neretva inference takes no more than a few minutes to complete on CYP genes. The more substantial improvements happen on KIR genes; while other tools might require an hour to complete, Neretva typically takes less than 20 minutes. The inference itself takes no more than a few minutes; the majority of the time is spent on alignment and the data

[3] Locityper's performance is significantly lower than originally expected. This might be due to an error in result interpretation on our end; we have reached out to the authors for further clarification and guidance.

collection. All experiments were done on an Intel Xeon 8260 machine with 1 TB of RAM and a NVidia Tesla V100 GPU with 32 GB of VRAM.

Discussion

In this paper, we have presented Neretva, a unified variational inference framework for genotyping highly polymorphic gene families. Unlike ILP approaches that guarantee optimality only under rigid model assumptions and lack scalability, Neretva's probabilistic framework naturally handles uncertainty and is able to scale to large gene families with complex features. This was clearly shown by the extensive experiments, which demonstrated that Neretva matches or exceeds state-of-the-art performance on both CYP and KIR gene families.

A key innovation of Neretva is learning sample-specific base emission probabilities instead of treating the allele database as the ground truth, which allows the flexibility of adapting to novel variants. In addition, Neretva is completely implemented on open source libraries, offers practical advantages for integration into clinical pipelines and enables future extensions for privacy-preserving inference.

Several limitations warrant discussion and further exploration. First, the current model treats variant observations as independent counts at each position, discarding read-level linkage information. We hope to extend the model in the future to incorporate read-level information. Secondly, certain complex structural variations within the *CYP2D6* locus, as well as various KIR fusions involving one or more genes, require further testing and may require additional modelling components (while Neretva handles gene copy number estimation and fusions, the fusion calling accuracy remains sensitive to sequencing noise as of the current version).

A few promising directions merit future investigation. First, we plan to implement support for long-read sequencing platforms such as PacBio HiFi and Oxford Nanopore. In addition, extending to other clinically important gene families, including HLA, immunoglobulin loci, and T-cell receptor genes, would be a natural step forward, since the framework is general. Another promising direction is integration with pangenome graph references. Recent advances in graph neural networks, including variational graph auto-encoders [31], provide principled approaches for learning representations over such structures, and we aim to explore their integration to better reason about overlapping haplotypes and complex structural variation. Finally, we plan to explore integrating our framework with privacy-preserving technologies that provide support for secure tensor algebra.

Conclusion

The reformulation of allele identification from combinatorial optimization to variational inference enables Neretva to jointly learn allele proportions and base emission probabilities, providing robustness to novel variants while maintaining biological interpretability. Our evaluation confirms that this approach matches or surpasses the existing methods across gene families of varying complexity. We hope Neretva becomes a practical tool for clinical pharmacogenomic pipelines, contributing to more accurate and equitable precision medicine.

Conflicts of interest

None declared.

Funding

This work was supported by NSERC Discovery RGPIN-2019-04973 and RGPIN-2025-04073, Canada Research Chairs, Canada Foundation for Innovation and BC Knowledge Development Fund grants.

Data availability

Neretva is open source and is freely available at `https://github.com/0xTCG/neretva`. This repository contains all logs produced by other tools, as well as a detailed description of the results. The sequencing data is available at `https://github.com/Illumina/Polaris/wiki/HiSeqX-PGx-Cohort` and `https://humanpangenome.org/data/`.

Author contributions statement

Q.Z. conceived the study and designed and implemented the framework. I.N. oversaw the study. S.P.A. conducted the experiments. All authors jointly wrote and reviewed the manuscript.

References

1. S. Agrawal and S. Prakash. Significance of kir like natural killer cell receptors in autoimmune disorders. *Clinical immunology*, 216:108449, 2020.
2. T. J. Aitman, R. Dong, T. J. Vyse, P. J. Norsworthy, M. D. Johnson, J. Smith, J. Mangion, C. Roberton-Lowe, A. J. Marshall, E. Petretto, et al. Copy number polymorphism in fcgr3 predisposes to glomerulonephritis in rats and humans. *Nature*, 439(7078):851–855, 2006.
3. L. M. Amorim, D. G. Augusto, N. Nemat-Gorgani, G. Montero-Martin, W. M. Marin, H. Shams, R. Dandekar, S. Caillier, P. Parham, M. A. Fernández-Viña, et al. High-resolution characterization of kir genes in a large north american cohort reveals novel details of structural and sequence diversity. *Frontiers in immunology*, 12:674778, 2021.
4. D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent dirichlet allocation. *Journal of machine Learning research*, 3(Jan): 993–1022, 2003.
5. J. Bruijnesteijn, M. Van der Wiel, N. G. De Groot, and R. E. Bontrop. Rapid characterization of complex killer cell immunoglobulin-like receptor (kir) regions using cas9 enrichment and nanopore sequencing. *Frontiers in Immunology*, 12:722181, 2021.
6. E. C. Castelli, R. N. Pereira, G. S. Paes, H. S. Andrade, M. R. Ferreira, Í. S. de Freitas Santos, N. Vince, N. R. Pollock, P. J. Norman, and D. Meyer. kir-mapper: A toolkit for killer-cell immunoglobulin-like receptor (kir) genotyping from short-read second-generation sequencing data. *HLA*, 105(3):e70092, 2025.
7. G. Chandra, M. Helal Hossen, S. Scholz, A. T. Dilthey, D. Gibney, and C. Jain. Integer programming

framework for pangenome-based genome inference. In *International Conference on Research in Computational Molecular Biology*, pages 405–408. Springer, 2025.

8. X. Chen, F. Shen, N. Gonzaludo, A. Malhotra, C. Rogert, R. J. Taft, D. R. Bentley, and M. A. Eberle. Cyrius: accurate cyp2d6 genotyping using whole-genome sequencing data. *The pharmacogenomics journal*, 21(2):251–261, 2021.
9. C.-S. Chin, S. Behera, A. Khalak, F. J. Sedlazeck, P. H. Sudmant, J. Wagner, and J. M. Zook. Multiscale analysis of pangenomes enables improved representation of genomic diversity for repetitive and clinically relevant genes. *Nature Methods*, 20(8):1213–1221, 2023.
10. A. B. Dieng, F. J. Ruiz, and D. M. Blei. Topic modeling in embedding spaces. *Transactions of the Association for Computational Linguistics*, 8:439–453, 2020.
11. M. K. Ford, A. Hari, O. Rodriguez, J. Xu, J. Lack, C. Oguz, Y. Zhang, A. J. Oler, O. M. Delmonte, S. E. Weber, et al. Immunotyper-sr: A computational approach for genotyping immunoglobulin heavy chain variable genes using short-read data. *Cell systems*, 13(10):808–816, 2022.
12. M. K. Ford, A. Hari, Q. Zhou, I. Numanagić, and S. C. Sahinalp. Biologically-informed killer cell immunoglobulin-like receptor gene annotation tool. *Bioinformatics*, 40(11): btae622, 2024.
13. A. Gaedigk, L. K. M. Jaime, J. S. Bertino Jr, A. Bérard, V. M. Pratt, L. D. Bradfordand, and J. S. Leeder. Identification of novel cyp2d7-2d6 hybrids: non-functional and functional variants. *Frontiers in pharmacology*, 1:121, 2010.
14. A. Gaedigk, K. Sangkuhl, M. Whirl-Carrillo, G. P. Twist, T. E. Klein, N. A. Miller, P. S. Committee, et al. The evolution of pharmvar. *Clinical pharmacology and therapeutics*, 105(1):29, 2018.
15. A. Gaedigk, E. C. Boone, S. E. Scherer, S.-b. Lee, I. Numanagić, C. Sahinalp, J. D. Smith, S. McGee, A. Radhakrishnan, X. Qin, et al. Cyp2c8, cyp2c9, and cyp2c19 characterization using next-generation sequencing and haplotype analysis: a get-rm collaborative project. *The Journal of Molecular Diagnostics*, 24(4):337–350, 2022.
16. K. Gagne, G. Brizard, B. Gueglio, N. Milpied, P. Herry, F. Bonneville, M.-L. Chéneau, N. Schleinitz, A. Cesbron, G. Folléa, et al. Relevance of kir gene polymorphisms in bone marrow transplantation outcome. *Human immunology*, 63 (4):271–280, 2002.
17. C. E. Hargreaves, M. J. Rose-Zerilli, L. R. Machado, C. Iriyama, E. J. Hollox, M. S. Cragg, and J. C. Strefford. Fcγ receptors: genetic variation, function, and disease. *Immunological reviews*, 268(1):6–24, 2015.
18. A. Hari, Q. Zhou, N. Gonzaludo, J. Harting, S. A. Scott, X. Qin, S. Scherer, S. C. Sahinalp, and I. Numanagić. An efficient genotyper and star-allele caller for pharmacogenomics. *Genome Research*, 33(1):61–70, 2023.
19. K. Hirayasu and H. Arase. Functional and genetic diversity of leukocyte immunoglobulin-like receptor and implication for disease associations. *Journal of human genetics*, 60(11): 703–708, 2015.
20. Y. Huang, J. Yang, D. Ying, Y. Zhang, V. Shotelersuk, N. Hirankarn, P. C. Sham, Y. L. Lau, and W. Yang. Hlareporter: a tool for hla typing from next generation sequencing data. *Genome medicine*, 7(1):25, 2015.
21. P. J. Huber. Robust estimation of a location parameter. In *Breakthroughs in statistics: Methodology and distribution*, pages 492–518. Springer, 1992.
22. M. Ingelman-Sundberg. Genetic polymorphisms of cytochrome p450 2d6 (cyp2d6): clinical consequences, evolutionary aspects and functional diversity. *The pharmacogenomics journal*, 5(1):6–13, 2005.
23. W. Jiang, C. Johnson, J. Jayaraman, N. Simecek, J. Noble, M. F. Moffatt, W. O. Cookson, J. Trowsdale, and J. A. Traherne. Copy number variation leads to considerable diversity for b but not a haplotypes of the human kir genes encoding nk cell receptors. *Genome research*, 22(10): 1845–1854, 2012.
24. W. Jiang, C. Johnson, N. Simecek, M. López-Álvarez, D. Di, J. Trowsdale, and J. Traherne. qkat: a high-throughput qpcr method for kir gene copy number and haplotype determination. *Genome medicine*, 8(1):99, 2016.
25. M. I. Jordan, Z. Ghahramani, T. S. Jaakkola, and L. K. Saul. An introduction to variational methods for graphical models. *Machine learning*, 37(2):183–233, 1999.
26. R. M. Karp. Reducibility among combinatorial problems. In *50 Years of Integer Programming 1958-2008: from the Early Years to the State-of-the-Art*, pages 219–241. Springer, 2009.
27. Z. Ke and H. Vikalo. A convolutional auto-encoder for haplotype assembly and viral quasispecies reconstruction. *Advances in Neural Information Processing Systems*, 33: 13493–13503, 2020.
28. S. I. Khakoo, C. L. Thio, M. P. Martin, C. R. Brooks, X. Gao, J. Astemborski, J. Cheng, J. J. Goedert, D. Vlahov, M. Hilgartner, et al. Hla and nk cell inhibitory receptor genes in resolving hepatitis c virus infection. *Science*, 305(5685): 872–874, 2004.
29. D. P. Kingma. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.
30. D. P. Kingma and M. Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.
31. T. N. Kipf and M. Welling. Variational graph auto-encoders. *arXiv preprint arXiv:1611.07308*, 2016.
32. D. D. Lee and H. S. Seung. Learning the parts of objects by non-negative matrix factorization. *nature*, 401(6755): 788–791, 1999.
33. S.-b. Lee, M. M. Wheeler, K. Patterson, S. McGee, R. Dalton, E. L. Woodahl, A. Gaedigk, K. E. Thummel, and D. A. Nickerson. Stargazer: a software tool for calling star alleles from next-generation sequencing data using cyp2d6 as a model. *Genetics in medicine*, 21(2):361–372, 2019.
34. S.-b. Lee, J.-Y. Shin, N.-J. Kwon, C. Kim, and J.-S. Seo. Clinpharmseq: A targeted sequencing panel for clinical pharmacogenetics implementation. *PLoS One*, 17 (7):e0272129, 2022.
35. H. Li. Minimap2: pairwise alignment for nucleotide sequences. *Bioinformatics*, 34(18):3094–3100, 2018.
36. W.-W. Liao, M. Asri, J. Ebler, D. Doerr, M. Haukness, G. Hickey, S. Lu, J. K. Lucas, J. Monlong, H. J. Abel, et al. A draft human pangenome reference. *Nature*, 617(7960): 312–324, 2023.
37. J. Lin. Divergence measures based on the shannon entropy. *IEEE Transactions on Information theory*, 37(1):145–151, 2002.

38. L.-I. I. Loga, R. Suharoschi, F. I. Elec, A. D. Chiorean, A. D. Elec, A. M. Muntean, M. D. Suciu, O. Antal, A. T. Toth, R. L. Lucaciu, et al. Orchestrating the impact of kir/hla interactions on kidney transplant. *International Journal of Molecular Sciences*, 25(15):8228, 2024.
39. W. M. Marin, R. Dandekar, D. G. Augusto, T. Yusufali, B. Heyn, J. Hofmann, V. Lange, J. Sauter, P. J. Norman, and J. A. Hollenbach. High-throughput interpretation of killer-cell immunoglobulin-like receptor short-read sequencing data with ping. *PLoS computational biology*, 17(8):e1008904, 2021.
40. M. P. Martin, X. Gao, J.-H. Lee, G. W. Nelson, R. Detels, J. J. Goedert, S. Buchbinder, K. Hoots, D. Vlahov, J. Trowsdale, et al. Epistatic interaction between kir3ds1 and hla-b delays the progression to aids. *Nature genetics*, 31(4):429–434, 2002.
41. T. K. Nguyen, C. T. Pham, T. Hoang, N. S. Vo, and T. H. Phung. A comprehensive assessment of pharmacogenomic annotation tools for next-generation sequencing data: an emphasis on cyp2d6 and vietnamese genomic data. *Journal of Human Genetics*, pages 1–7, 2025.
42. I. Numanagić, S. Malikić, M. Ford, X. Qin, L. Toji, M. Radovich, T. C. Skaar, V. M. Pratt, B. Berger, S. Scherer, and S. C. Sahinalp. Allelic decomposition and exact genotyping of highly polymorphic and structurally variant genes. *Nature Communications*, 9(1):828, Feb. 2018. ISSN 2041-1723. doi: 10.1038/s41467-018-03273-1. URL `https://www.nature.com/articles/s41467-018-03273-1`. Publisher: Nature Publishing Group.
43. S.-T. Pan, D. Xue, Z.-L. Li, Z.-W. Zhou, Z.-X. He, Y. Yang, T. Yang, J.-X. Qiu, and S.-F. Zhou. Computational identification of the paralogs and orthologs of human cytochrome p450 superfamily and the implication in drug discovery. *International journal of molecular sciences*, 17 (7):1020, 2016.
44. V. M. Pratt, R. E. Everts, P. Aggarwal, B. N. Beyer, U. Broeckel, R. Epstein-Baak, P. Hujsak, R. Kornreich, J. Liao, R. Lorier, S. A. Scott, C. H. Smith, L. H. Toji, A. Turner, and L. V. Kalman. Characterization of 137 Genomic DNA Reference Materials for 28 Pharmacogenetic Genes: A GeT-RM Collaborative Project. *The Journal of molecular diagnostics: JMD*, 18(1):109–123, Jan. 2016. ISSN 1943-7811. doi: 10.1016/j.jmoldx.2015.08.005.
45. T. Prodanov, E. G. Plender, G. Seebohm, S. G. Meuth, E. E. Eichler, and T. Marschall. Locityper enables targeted genotyping of complex polymorphic genes. *Nature Genetics*, pages 1–8, 2025.
46. J. A. Reuter, D. V. Spacek, and M. P. Snyder. High-throughput sequencing technologies. *Molecular cell*, 58(4): 586–597, 2015.
47. J. D. Robarge, L. Li, Z. Desta, A. Nguyen, and D. A. Flockhart. The star-allele nomenclature: retooling for translational genomics. *Clinical Pharmacology & Therapeutics*, 82(3):244–248, 2007.
48. J. Robinson, J. A. Halliwell, H. McWilliam, R. Lopez, and S. G. Marsh. Ipd—the immuno polymorphism database. *Nucleic acids research*, 41(D1):D1234–D1240, 2012.
49. D. Roe, C. Vierra-Green, C.-W. Pyo, K. Eng, R. Hall, R. Kuang, S. Spellman, S. Ranade, D. Geraghty, and M. Maiers. Revealing complete complex kir haplotypes phased by long-read sequencing technology. *Genes & Immunity*, 18(3):127–134, 2017.
50. L. Song, G. Bai, X. S. Liu, B. Li, and H. Li. Efficient and accurate kir and hla genotyping with massively parallel sequencing data. *Genome Research*, 33(6):923–931, 2023. doi: 10.1101/gr.277585.122.
51. A. Srivastava and C. Sutton. Autoencoding variational inference for topic models. *arXiv preprint arXiv:1703.01488*, 2017.
52. A. Szolek, B. Schubert, C. Mohr, M. Sturm, M. Feldhahn, and O. Kohlbacher. Optitype: precision hla typing from next-generation sequencing data. *Bioinformatics*, 30(23): 3310–3316, 2014.
53. D. Twesigomwe, G. E. Wright, B. I. Drögemöller, J. Da Rocha, Z. Lombard, and S. Hazelhurst. A systematic comparison of pharmacogene star allele calling bioinformatics algorithms: a focus on cyp2d6 genotyping. *NPJ genomic medicine*, 5(1):30, 2020.
54. D. Twesigomwe, B. I. Drögemöller, G. E. Wright, A. Siddiqui, J. da Rocha, Z. Lombard, and S. Hazelhurst. Stellarpgx: a nextflow pipeline for calling star alleles in cytochrome p450 genes. *Clinical Pharmacology & Therapeutics*, 110(3):741–749, 2021.
55. G. P. Twist, A. Gaedigk, N. A. Miller, E. G. Farrow, L. K. Willig, D. L. Dinwiddie, J. E. Petrikin, S. E. Soden, S. Herd, M. Gibson, et al. Constellation: a tool for rapid, automated phenotype assignment of a highly polymorphic pharmacogene, cyp2d6, from whole-genome sequences. *NPJ genomic medicine*, 1(1):1–10, 2016.
56. M. van der Lee, M. Kriek, H.-J. Guchelaar, and J. J. Swen. Technologies for pharmacogenomics: a review. *Genes*, 11 (12):1456, 2020.
57. W. Y. Wang, D. Twesigomwe, C. Nofziger, A. J. Turner, L.-S. Helmecke, U. Broeckel, A. D. Derezinski, S. Hazelhurst, and A. Gaedigk. Characterization of novel cyp2d6 alleles across sub-saharan african populations. *Journal of Personalized Medicine*, 12(10):1575, 2022.
58. R. M. Weinshilboum and L. Wang. Pharmacogenomics: precision medicine and drug response. In *Mayo Clinic Proceedings*, volume 92, pages 1711–1722. Elsevier, 2017.
59. U. M. Zanger and M. Schwab. Cytochrome p450 enzymes in drug metabolism: regulation of gene expression, enzyme activities, and impact of genetic variation. *Pharmacology & therapeutics*, 138(1):103–141, 2013.
60. Q. Zhou, M. Ghezelji, A. Hari, M. K. Ford, C. Holley, S. C. Sahinalp, and I. Numanagić. Geny: a genotyping tool for allelic decomposition of killer cell immunoglobulin-like receptor genes. *Frontiers in Immunology*, 15:1494995, 2024.
61. Y. Zhou and V. M. Lauschke. The genetic landscape of major drug metabolizing cytochrome p450 genes—an updated analysis of population-scale sequencing data. *The Pharmacogenomics Journal*, 22(5):284–293, 2022.