*Bioinformatics*, 2026, X, 1–7
**ECCB 2026**

# Neretva: 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 a probabilistic latent variable model and employs auto-encoding variational Bayes (AEVB) for inference. Leveraging neural inference networks, our approach scales efficiently to complex gene families such as CYP (cytochrome P450) and KIR (killer-cell immunoglobulin-like receptors), and achieves competitive or improved accuracy over the current state of the art on both families. Neretva is open source and is freely available at** `https://github.com/0xTCG/neretva`**.**

## Introduction

High-throughput sequencing (HTS) technologies are a promising solution for enabling precision medicine through comprehensive and rapid analysis of genotypes of clinically important gene families [48]. Key families include the cytochrome P450 (CYP) pharmacogenes and killer-cell immunoglobulin-like receptor (KIR) genes. The CYP family encodes enzymes responsible for the metabolism of more than 70% of clinically prescribed drugs [49], while KIR genes regulate natural killer cell and T-cell responses, and are linked, among other things, to HIV progression, autoimmune disorders, and organ transplant success rates [30, 1, 32].

Despite its importance, accurate genotyping of these genes in HTS data remains challenging due to several factors. The first challenge stems from the fact that their alleles are defined as haplotypes, which requires accurate phasing to establish a clinically actionable phenotype. Phasing is often hindered by short read length, as well as extreme inter-gene sequence similarity between pharmacogenes and their paralogs (often present within pharmacogene clusters [36]) and the intra-gene allelic identity (often exceeding >98%), which introduce significant ambiguity during assigning individual HTS reads to their true origin [4, 35]. Many pharmacogene families also exhibit substantial population-specific variation [47], thus rendering the strict reliance on the previously characterized alleles insufficient. Furthermore, some pharmacogenes exhibit extensive copy number variation and are subject to complex structural variations such as gene deletions, duplications, and paralog-related gene fusions [12]. Clusters such as KIR exhibit even higher levels of complexity; around half of the 17 KIR genes exhibit extensive diversity and are subject to copy number variation and cross-gene structural rearrangements [17, 18]. Finally, we note that other clinically important gene families not covered here present similar challenges; examples include *HLA* genes, immunoglobulin loci, T-cell receptor genes, and other immune receptor families such as *FCGR* and *LILR* [10, 2, 28, 43].

Many specialized frameworks were developed to accurately infer such genotypes from HTS data. Tools such as Stargazer [25], Cyrius [7], StellarPGx [45], PyPGx [26], Aldy [35, 15] and Astrolabe [46] combine the existing allele databases such as PharmVar [13], as well rule-based strategies and various statistical and combinatorial techniques, to accurately infer genotypes for wide variety of pharmacogenes. For genotyping KIR genes, such as T1K [41], Locityper [38], Geny [50], kir-mapper [5] and PING [31] often rely on combinatorial and likelihood-based assignment strategies to maximize the likelihood of observed read data and estimate candidate allele abundances, together with various pre-processing schemes such as $k$-mer-based fingerprinting. Other gene clusters are also handled by specialized tools; some examples include OptiType [43] and HLAreporter [16] (for HLA), Immunotyper-SR (for IGH) [10, 11], and the recent pangenome-based approaches such as PGR-TK [8].

As demonstrated by the performance of the tools such as Aldy (pharmacogenes)[1], Geny (KIR), ImmunoTyper-SR (IGH), OptiType (HLA), as well as the recent pangenome-based PHI [6], one of the most effective ways to genotype complex genes is through combinatorial optimization [34, 44]. In this approach, the candidate alleles are first filtered by eliminating alleles with low abundance (through a rule or statistical-based approach), and the optimal solution is then found by solving one

[1] For example, Aldy was shown to consistently produce good results across many different HTS technologies, often achieving up to 99% accuracy

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

or more constrained integer linear programming (ILP) models, often composed into a multi-stage process where each stage handles a different aspect of the genotyping problem (e.g., Aldy employs separate ILPs for structural variant detection and for allele phasing). Although such approaches guarantee a theoretical optimum and constraint enforcement, and avoid converging to an incorrect local optimum (unlike with likelihood-based approaches), they are often NP-hard [35] and scale poorly to new data types and complex gene families. While such models are feasible and practical for pharmacogenes (where the ILP instances contain a manageable number of variables), handling of complex gene families such as KIR leads to a combinatorial explosion that results in either impractical runtimes or in sacrificing optimality guarantees. ILP-based approaches also rely on black-box (and often proprietary) solvers such as Gurobi or Google OR-Tools, which make their use impractical in specialized clinical applications that require use of privacy-preserving technologies (e.g., homomorphic encryption) for handling patient data. Lastly, ILP solutions are rigid, and it is not easy to establish the confidence of an ILP solution in a noisy dataset.

These 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) 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 [24, 3], while neural variational models such as ProdLDA [42], NVDM [33] and ETM [9] have scaled topic modeling to millions of documents and thousands of topics. In genomics, deep learning approaches such as CAECseq [20] (which uses convolutional autoencoders for haplotype assembly and viral quasispecies reconstruction) have demonstrated that neural networks can effectively cluster sequencing reads originating from highly similar genomic regions. All these methods leverage GPU-accelerated gradient optimization and inference networks to achieve orders-of-magnitude speedup over traditional approaches without impacting the model accuracy. Beyond scalability, AEVB models also offer flexibility for incorporating domain constraints as differentiable regularizers, and avoid the need to re-derive update equations (as in likelihood-based strategies) or redesign samplers (as in MCMC), which makes them desirable for large-scale inference problems like KIR genotyping.

In this paper, we present a new unified AEVB-based framework, called Neretva, for genotyping both pharmacogene (CYP) and KIR gene families. Our approach combines the variant-centric modelling philosophy of Aldy with the scalability of modern variational inference. 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 complex gene family and a collection of sequencing reads in BAM/CRAM/FASTQ, Neretva infers the overall genotype of the given family by (1) determining the number of copies of each gene within that family, and by (2) constructing the exact sequence content (haplotype) of each such copy (in other words, assign an allele designation to it).

## Database Preparation

Consider a multi-gene model where $\mathcal{G} = \{G_1, \ldots, G_L\}$ represents a family of $L \geq 1$ genes (e.g., KIR family contains $L = 17$ genes). Let us define a set $\mathcal{M}_G$ of all possible variants occurring within $G$ 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 phenotype or a protein function of $G$), and $\mathcal{M}_G^\times$ that contain *minor variants* (silent variants without known functionality).

Each gene $G \in \mathcal{G}$ defines a *reference allele* with no variants (often referred to as *1 allele). Each other allele $a_i^G$ of that gene is defined by a set of variants (denoted as $\mathcal{A}^G$), itself split into a set of core ($\mathcal{M}_{a_i^G}^+$) and minor ($\mathcal{M}_{a_i^G}^\times$) variants, that differentiates such allele from the reference allele. 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 characterized alleles that includes the known alleles for each gene of interest, along with the variant and structural variation information. For KIR genes, we rely on IPD-KIR database v2.12.0 [40], while for other genes, we utilize the PharmVar database 6.2 [13]. In both databases, allele names roughly follow the star-allele nomenclature [39], where the major allele designation (typically represented as star-number; e.g., *CYP2D6*2*) describes the allele's functionality (and is defined by its core variants), while the minor allele designation (e.g., *2A) distinguishes alleles with differing minor variations.

## Alignment and Filtering

Consider an HTS sample with reads $\mathcal{R} = \{r_1, \ldots, r_{|\mathcal{R}|}\}$. For pharmacogenes, we assume each read alignment to be correct and use it as-is. As for KIR, where many reads ambiguously map to different genes, we align reads against all known allele sequences with minimap2 [27] in all-to-all mode[2]. 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 search space, we filter down the allele set $\mathcal{A}$ into $\mathcal{A}'$ so that each $A \in \mathcal{A}'$ has each $m \in \mathcal{M}_A^+$ covered by at least 3 reads.

In clusters where incorrect cross-gene alignments are common (e.g., KIR), 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 improper medical treatment. Hence, for each read $r$ that induces a variant $m_k^G \in \mathcal{M}_r$, we check whether there is an allele $a_i^{G'} \in \mathcal{A}_r$ such that $G' \neq G$. If so, 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"

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

has no biological implications; it is only used 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) alleles in a given sample [51]. Thus, to identify the set of alleles that best describes $\mathbf{Y}$, we must determine the respective base for each $a_k^G \in \mathcal{A}$ at each $p \in \mathcal{P}$ (if $p$ describes any variant in $a_k^G$'s *extended variant set*). Both the allele proportions and the base emissions are modelled as latent variables; more precisely, $\boldsymbol{\theta} \in \Delta^K$ represents the relative proportion of each of $K = |\mathcal{A}'|$ candidate alleles, where $\Delta^K$ denotes a $K-1$ simplex. $\boldsymbol{\Psi} = \{\psi_{a,p} \mid a \in \mathcal{A}', p \in \mathcal{P}\}$ represents the base emission probabilities, 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 pharmacogenes, 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 $H_\delta$ is the Huber loss with threshold $\delta$ that provides robustness to outliers from coverage anomalies. The final copy number is obtained by minimizing $\mathcal{L}(\mathbf{n})$ via Adam optimizer [21].

## Bias Factor Matrix

While many HTS experiments in theory promise uniform coverage, coverage of some bases might not match the expected coverage due to sequencing biases or mapping artifacts, especially in complex families such as KIR. 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 gene profile that accounts for known coverage biases. For shadow variants, we assign bias factors based on empirical misalignment rates obtained from the simulated alignment data. Such matrix is sparse because it has no entries for positions where no variant (true or shadow) is observed. This bias adjustment allows the model to appropriately amplify the signal from the true variants.

## Allele Identification Model

Alleles are inferred by a generative process for the count matrix $\mathbf{Y}$ collected from a data sample. Let $\mathcal{LN}(\mu, \Sigma)$ be a logistic normal distribution. For clarity, 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. The generation process is 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 [22] to approximate the posterior $p(\boldsymbol{\theta}, \boldsymbol{\Psi}|\mathbf{Y})$ for each sample independently. For

computational efficiency, we adopt mean-field variational approximation [19]:

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

Under this factorization, the evidence lower bound (ELBO) becomes (see Appendix for the complete derivation):

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

However, while maximizing (3) produces a valid data fit across all variants, it does not prioritize core variants that capture the true biological phenotype of each allele. To address this, consider a 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 $\mathcal{M}_+$: (1) the empirical functional variant count distribution $\hat{p} \in \Delta^{|\mathcal{M}_+|}$ derived from $\mathbf{Y}$, and (2) 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 [29], whose symmetry allows us to penalize both the lack and the surplus of core variants.

Denote $\mathrm{D}_{\mathrm{JS}}(\hat{p}\|\tilde{\pi})$ 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:

$$\tilde{\mathcal{L}} = -\mathcal{L} + \beta \cdot D_{\mathsf{JS}}(\hat{p}\|\tilde{\pi}) + \gamma \cdot H(\boldsymbol{\Psi}). \tag{4}$$

The hyperparameters $\beta$, $\gamma$ are balancing terms for the final loss term. We minimize this loss with Adam optimizer [21], where gradients are computed via the reparameterization trick [22] (see Appendix for details).

# Results

We chose three pharmacogene families in the increasing order of complexity to comprehensively benchmark Neretva [3]. 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*, and exhibits both copy number changes and structural variants. Finally, the most complex KIR family contains 17 genes and, in addition to copy number and structural variants, exhibits high levels of ambiguous mappings.

The pharmacogene (CYP) families are evaluated on 70 GeT-RM Illumina WGS samples with an average sequencing depth of 30× [15][4]. All ground truths were previously validated by the GeT-RM consortium and augmented by the recently published updates [14, 37, 15]. This dataset contains, in addition to diploid genotypes, a number of *CYP2D6* copy number variants (both deletions and duplications) and paralog-related fusions. Besides Neretva, we also benchmarked the current state-of-the-art pharmacogene genotypers: StellarPGX, Aldy, Cyrius, Stargazer, Astrolabe and PyPGx. Note that, despite the fact that CYP pharmacogene genotyping is a well-studied problem (most tools routinely obtain a near-perfect $F_1$ score), we still benchmark it to show that our model meets the expected baseline performance.

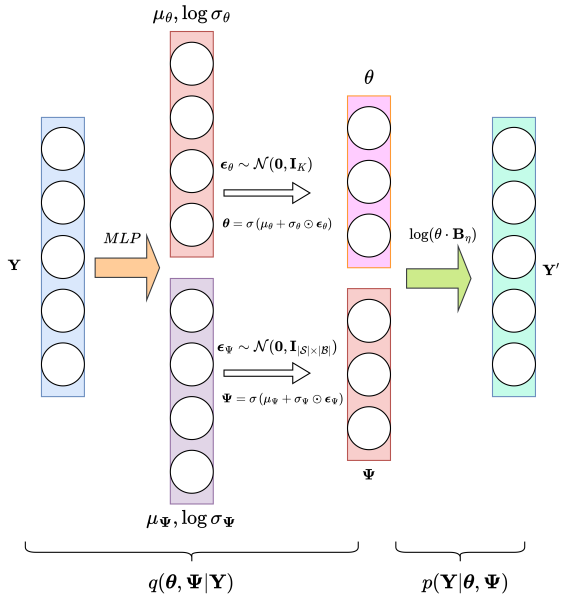


**Figure 1** Network structure of inference model $q(\boldsymbol{\theta}, \boldsymbol{\Psi}|\mathbf{Y})$ and generative model $p(\mathbf{Y}|\boldsymbol{\theta}, \boldsymbol{\Psi})$.

Neretva was also benchmarked against the following KIR genotypers: Geny, Locityper, T1K, kir-mapper and PING on 40 Human Pangenome Reference Consortium (HPRC) samples [28]. The KIR ground truth calls for the HPRC samples were obtained from the HPRC assemblies [28] via the BAKIR annotation tool [11].

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[5], 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 is defined as the number of true positives divided by the number of samples, while the precision is defined as the

[3] Other pharmacogenes were not considered to avoid clutter, and also because their complexity does not surpass the complexity of the families evaluated here [15, 14]

[4] While GeT-RM project describes 137 samples, only 70 of them are publicly available in the evaluated WGS cohort.

[5] Concordance means that the allele call matches the ground truth at functional level; in case of pharmacogenes, 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 pharmacogenes and KIR genes. In addition to Neretva, only top three tools are shown here (other tools are shown in Supplementary Table 3). The best results are highlighted in bold.

| **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 |
| 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** |
| Neretva | 70 | 69 | 0.986 | 0.986 | **1.000** | 0.993 | 162 | 161 | 0.994 | 0.994 | 0.994 | 0.994 |
| **KIR genes** | **Per Call (Sample-level)** | | | | | | **Per Allele** | | | | | |
| **Tool** | $n$ | **Correct** | **Accuracy** | **Precision** | **Recall** | $F_1$ | $n$ | **Correct** | **Accuracy** | **Precision** | **Recall** | $F_1$ |
| 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 |
| Neretva | 442 | **372** | **0.842** | **0.845** | 0.982 | **0.908** | 749 | **682** | **0.911** | **0.914** | **0.911** | **0.912** |

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.

*Pharmacogene performance*

As shown in Table 1 (and Supplementary Table 3), Neretva was able to consistently perfectly genotype *CYP2C19*, and matched the state of the art, such as Aldy, Astrolabe, StellarPGX and PyPGx. Same trends were observed for *CYP2C8*, and *CYP2C9* as well (see Supplementary Table 3). 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 1 (and Supplementary Table 3), 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 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, T1K achieved lower accuracy on this dataset, with a sample accuracy of 0.520.

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). Per-gene results are available in Supplementary Table 4. 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 tools.

Note that Locityper's performance is significantly lower than originally expected (Supplementary Table 3). We have reached out to the authors for further clarification and guidance. We also note that our evaluation of all tools differs slightly from the results published in Geny [50]. The reason for that is improved handling of the false negatives in Table 1 (in some instances, false negatives were systematically ignored for all tools in Geny's evaluation script).

*Ablations and computational efficiency.*

To show the importance of each modelling feature, we performed a set of ablation experiments to demonstrate the impact of the Jensen-Shannon divergence term, entropy penalty, and bias modelling on KIR genes, where all terms are required for optimal performance. The Supplementary Table 1 shows that the $F_1$ score is highest when all modelling features are enabled.

We also show that Neretva's approach is more practical than ILP modelling on the most complex dataset (KIR). For this, we devised an ILP model analogous to Neretva and used the state-of-the-art Gurobi solver to solve it (Supplementary Table 2). While Neretva spent less than 8 minutes per model, no ILP was able to converge to the optimal solution in 30 minutes (time limit set to 4× of that of Neretva). The solutions that were obtained were all strictly worse than Neretva, as shown in Supplementary Table 2.

In general, Neretva inference takes no more than a few minutes to complete on pharmacogenes. The more substantial improvements occur on KIR genes; while other tools might take an hour to complete, Neretva typically takes less than 20 minutes. The inference itself takes no more than a few minutes (8 minutes in the worst case); the rest is spent on alignment and data parsing. All experiments were done on an Intel Xeon 8260 machine with 1 TB of RAM and an 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. A key innovation of Neretva is its reformulation of the genotyping problem from combinatorial optimization to variational inference, enabling scalable gradient-based optimization that naturally accommodates uncertainty while maintaining biological interpretability. Unlike ILP approaches that guarantee optimality only under rigid model assumptions and lack scalability, we have shown through extensive experiments across major complex gene families that Neretva's probabilistic framework naturally handles uncertainty and can scale to large gene families with complex features.

Several limitations of the current version of Neretva warrant discussion and further exploration. First, the current model treats variant observations as independent counts at each position and does not consider read-level linkage information. We hope to extend the model to incorporate read-level information. Secondly, support for various KIR fusions involving one or more genes, in addition to certain complex structural variations within the *CYP2D6* locus, still requires improvements in copy number modelling and a more extensive evaluation dataset (while Neretva can call gene copy numbers and some fusions, the accuracy is currently sensitive to sequencing noise).

A few promising directions merit future investigation. The first is the support for long-read sequencing platforms such as Oxford Nanopore. Moreover, the generality of our framework suggests a potential inclusion of other clinically important gene families, such as HLA and T-cell receptor genes. Another promising direction is using pangenome graph references to obtain alignments and model base counts. Recent advances in graph neural networks, including variational graph auto-encoders [23], also 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, our reliance on open-source libraries enables us to explore the potential integration with privacy-preserving technologies that provide support for secure tensor algebra in the near future.

Our work demonstrates that Neretva achieves competitive or improved accuracy compared to existing tools, especially on more complex gene families such as KIR, and that its variational inference framework provides a principled and extensible foundation for genotyping complex genomic loci. We hope that our work, however modest, contributes to improved decisions in precision medicine.

## Acknowledgments

**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.

**Conflicts of interest:** None declared.

**Data availability:** Neretva is open source and is freely available at `https://github.com/0xTCG/neretva`. This repository contains all tool outputs, as well as a detailed description of the results and data sources. Data sources are listed in Appendix.

**Author contributions:** 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, et al. Copy number polymorphism in FCGR3 predisposes to glomerulonephritis in rats and humans. *Nature*, 439(7078):851–855, 2006.
3. D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent dirichlet allocation. *Journal of machine Learning research*, 3(Jan): 993–1022, 2003.
4. J. Bruijnesteijn, M. Van der Wiel, et al. Rapid characterization of complex killer cell immunoglobulin-like receptor (KIR) regions using Cas9 enrichment and nanopore sequencing. *Frontiers in Immunology*, 12:722181, 2021.
5. E. C. Castelli, R. N. Pereira, G. S. Paes, et al. kir-mapper: A toolkit for killer-cell immunoglobulin-like receptor (KIR) genotyping from short-read second-generation sequencing data. *HLA*, 105(3):e70092, 2025.
6. G. Chandra et al. Integer programming framework for pangenome-based genome inference. In *International Conference on Research in Computational Molecular Biology*, pages 405–408, 2025.
7. X. Chen, F. Shen, N. Gonzaludo, et al. Cyrius: accurate CYP2D6 genotyping using whole-genome sequencing data. *The pharmacogenomics journal*, 21(2):251–261, 2021.
8. C.-S. Chin, S. Behera, A. Khalak, et al. Multiscale analysis of pangenomes enables improved representation of genomic diversity for repetitive and clinically relevant genes. *Nature Methods*, 20(8):1213–1221, 2023.
9. 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.
10. M. K. Ford, A. Hari, O. Rodriguez, 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.
11. M. K. Ford, A. Hari, Q. Zhou, et al. Biologically-informed killer cell immunoglobulin-like receptor gene annotation tool. *Bioinformatics*, 40(11):btae622, 2024.
12. A. Gaedigk, L. K. M. Jaime, et al. Identification of novel CYP2D7-2D6 hybrids: non-functional and functional variants. *Frontiers in pharmacology*, 1:121, 2010.
13. A. Gaedigk, K. Sangkuhl, M. Whirl-Carrillo, et al. The evolution of PharmVar. *Clinical pharmacology and therapeutics*, 105(1):29, 2018.
14. A. Gaedigk, E. C. Boone, 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.
15. A. Hari, Q. Zhou, N. Gonzaludo, et al. An efficient genotyper and star-allele caller for pharmacogenomics. *Genome Research*, 33(1):61–70, 2023.
16. Y. Huang, J. Yang, D. Ying, et al. HLAreporter: a tool for HLA typing from next generation sequencing data. *Genome medicine*, 7(1):25, 2015.

17. W. Jiang, C. Johnson, J. Jayaraman, et al. 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.
18. 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.
19. M. I. Jordan et al. An introduction to variational methods for graphical models. *Machine learning*, 37(2):183–233, 1999.
20. 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.
21. D. P. Kingma. Adam: A method for stochastic optimization. *arXiv:1412.6980*, 2014.
22. D. P. Kingma and M. Welling. Auto-encoding variational bayes. *arXiv:1312.6114*, 2013.
23. T. N. Kipf and M. Welling. Variational graph auto-encoders. *arXiv:1611.07308*, 2016.
24. D. D. Lee and H. S. Seung. Learning the parts of objects by non-negative matrix factorization. *Nature*, 401(6755): 788–791, 1999.
25. S.-b. Lee, M. M. Wheeler, K. Patterson, et al. 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.
26. S.-b. Lee, J.-Y. Shin, et al. ClinPharmSeq: A targeted sequencing panel for clinical pharmacogenetics implementation. *PLoS One*, 17(7):e0272129, 2022.
27. H. Li. Minimap2: pairwise alignment for nucleotide sequences. *Bioinformatics*, 34(18):3094–3100, 2018.
28. W.-W. Liao, M. Asri, J. Ebler, et al. A draft human pangenome reference. *Nature*, 617(7960):312–324, 2023.
29. J. Lin. Divergence measures based on the Shannon entropy. *IEEE Trans. on Information theory*, 37(1):145–151, 2002.
30. L.-I. I. Loga, R. Suharoschi, et al. Orchestrating the impact of KIR/HLA interactions on kidney transplant. *International Journal of Molecular Sciences*, 25(15):8228, 2024.
31. W. M. Marin, R. Dandekar, D. G. Augusto, et al. High-throughput interpretation of killer-cell immunoglobulin-like receptor short-read sequencing data with ping. *PLoS computational biology*, 17(8):e1008904, 2021.
32. M. P. Martin, X. Gao, J.-H. Lee, et al. Epistatic interaction between KIR3DS1 and HLA-B delays the progression to AIDS. *Nature genetics*, 31(4):429–434, 2002.
33. Y. Miao, L. Yu, and P. Blunsom. Neural variational inference for text processing. In *International conference on machine learning*, pages 1727–1736. PMLR, 2016.
34. 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.
35. I. Numanagić, S. Malikić, M. Ford, et al. Allelic decomposition and exact genotyping of highly polymorphic and structurally variant genes. *Nature Communications*, 9 (1):828, Feb. 2018. ISSN 2041-1723.
36. S.-T. Pan, D. Xue, Z.-L. Li, et al. 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.
37. V. M. Pratt, R. E. Everts, P. Aggarwal, et al. 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.
38. 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.
39. 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.
40. 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.
41. 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.
42. A. Srivastava and C. Sutton. Autoencoding variational inference for topic models. *arXiv:1703.01488*, 2017.
43. A. Szolek, B. Schubert, C. Mohr, et al. OptiType: precision hla typing from next-generation sequencing data. *Bioinformatics*, 30(23):3310–3316, 2014.
44. D. Twesigomwe, G. E. Wright, B. I. Drögemöller, et al. A systematic comparison of pharmacogene star allele calling bioinformatics algorithms: a focus on CYP2D6 genotyping. *NPJ genomic medicine*, 5(1):30, 2020.
45. D. Twesigomwe, B. I. Drögemöller, G. E. Wright, et al. StellarPGx: a nextflow pipeline for calling star alleles in cytochrome P450 genes. *Clinical Pharmacology & Therapeutics*, 110(3):741–749, 2021.
46. G. P. Twist, A. Gaedigk, N. A. Miller, 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.
47. W. Y. Wang, D. Twesigomwe, C. Nofziger, et al. Characterization of novel CYP2D6 alleles across sub-saharan african populations. *Journal of Personalized Medicine*, 12 (10):1575, 2022.
48. R. M. Weinshilboum and L. Wang. Pharmacogenomics: precision medicine and drug response. In *Mayo Clinic Proceedings*, volume 92, pages 1711–1722. Elsevier, 2017.
49. 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.
50. Q. Zhou, M. Ghezelji, A. Hari, et al. Geny: a genotyping tool for allelic decomposition of killer cell immunoglobulin-like receptor genes. *Frontiers in Immunology*, 15:1494995, 2024.
51. 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.