Autoencoders for genomic variation analysis

Margarita Geleta, Daniel Mas Montserrat, Xavier Giro-i-Nieto, et al.

Genome Res. 2026 36: 348-360 originally published online January 20, 2026
Access the most recent version at doi:10.1101/gr.280086.124

Method

Autoencoders for genomic variation analysis

Margarita Geleta,[1,2,3] Daniel Mas Montserrat,[1] Xavier Giro-i-Nieto,[2] and Alexander G. Ioannidis[1,4,5]

[1]*Department of Biomedical Data Science, Stanford University School of Medicine, Stanford, California 94305, USA;* [2]*Department of Signal and Theory Communications, Universitat Politècnica de Catalunya, Barcelona 08034, Spain;* [3]*Department of Electrical Engineering and Computer Science, University of California, Berkeley, Berkeley, California 94720, USA;* [4]*Genomics Institute, University of California, Santa Cruz, Santa Cruz, California 95060, USA;* [5]*Institute for Computational and Mathematical Engineering, Stanford University School of Engineering, Stanford, California 94305, USA*

Modern biobanks are providing numerous high-resolution genomic sequences of diverse populations. In order to account for diverse and admixed populations, new algorithmic tools are needed in order to properly capture the genetic composition of populations. Here, we explore deep learning techniques, namely, variational autoencoders (VAEs), to process genomic data from a population perspective. We show the power of VAEs for a variety of tasks relating to the interpretation, compression, classification, and simulation of genomic data with several worldwide whole genome data sets from both humans and canids, and evaluate the performance of the proposed applications with and without ancestry conditioning. The unsupervised setting of autoencoders allows for the detection and learning of granular population structure and inferring of informative latent factors. The learned latent spaces of VAEs are able to capture and represent differentiated Gaussian-like clusters of samples with similar genetic composition on a fine scale from single nucleotide polymorphisms (SNPs), enabling applications in dimensionality reduction and data simulation. These individual genotype sequences can then be decomposed into latent representations and reconstruction errors (residuals), which provide a sparse representation useful for lossless compression. We show that different populations have differentiated compression ratios and classification accuracies. Additionally, we analyze the entropy of the SNP data, its effect on compression across populations, and its relation to historical migrations, and we show how to introduce autoencoders into existing compression pipelines.

[Supplemental material is available for this article.]

Introduction

Deep learning is becoming ubiquitous across all areas of science and engineering, with artificial neural networks (ANNs) being used to model highly nonlinear and complex data and proving successful in a wide range of applications. Recent works have begun to introduce such techniques within the fields of population genetics and precision medicine (Romero et al. 2016; Montserrat et al. 2020; Dominguez Mantes et al. 2023). Here, we explore the use of variational autoencoders (VAEs), a type of neural network used to learn a low-dimensional representation of the data, to analyze sequences of single nucleotide polymorphisms (SNPs) and showcase many applications including dimensionality reduction, data compression, classification, and simulation.

The number of human genomes being sequenced every year is growing rapidly, fueled by improvements in sequencing technology (Hernaez et al. 2019). Biobanks, paramount in areas like precision medicine, are powering genome-wide association studies (GWAS), where many genetic variants (e.g., SNPs) are analyzed across different subjects to find the relationships between genetic and phenotypic traits (Mardis 2011; Wojcik et al. 2019), are used to develop new treatments and drugs (Shah and Gaedigk 2018), and to address disparities and promote equity in precision medicine (Rhead et al. 2023). This creates a need for new, efficient, and accurate data-driven algorithmic tools to store, visualize, and characterize high-dimensional genomic data. Whereas many traditional statistical techniques for genomic data like hidden Markov models (HMMs) (Tang et al. 2006; Corbett-Detig and Nielsen 2017; Browning et al. 2018, 2023) become computationally expensive and slow when faced with biobank-sized data, neural networks can provide a powerful alternative.

Genomic sequence compression

The storage and transmission of high-dimensional biobank data require substantial space and channel capacity. This need has motivated the development of high-performance compression tools tailored to the unique particularities of this type of data (Brandon et al. 2009; Giancarlo et al. 2009; Nalbantoglu et al. 2010; Hernaez et al. 2019), which include the inherently high dimensionality of the data and its sparse yet complex structure. The demand for more powerful compression approaches becomes increasingly vital in modern large biobanks that contain genomic sequences produced by high-throughput sequencing. ANN-based approaches had already shown superior compression ratios decades ago (Schmidhuber and Heil 1996; Mahoney 2000). The first neural compressors mimicked the adaptive prediction by partial matching (PPM) model, where a neural network is used to estimate (and/or adjust) the probabilities for each symbol and then a usual coding method such as arithmetic coding is used to convert the data into a compressed bitstream (Mahoney 2000). Following this line, "DeepZip" (Goyal et al. 2018) estimated the probabilities by processing genomic sequences with gated recurrent units

Corresponding author: geleta@berkeley.edu
Article published online before print. Article, supplemental material, and publication date are at https://www.genome.org/cgi/doi/10.1101/gr.280086.124. Freely available online through the *Genome Research* Open Access option.

36:348–360 Published by Cold Spring Harbor Laboratory Press; ISSN 1088-9051/26; www.genome.org

(GRUs) and long short-term memory units (LSTMs). Mahoney (2005) introduced "logistic mixing," a neural network without hidden layers that uses a simple update rule to adjust the output probabilities for better bitstream coding. This idea has been brought to genomic compression of sequencing reads with GeCo3 (Silva et al. 2020). Another recent use of neural networks for DNA compression has been shown in DeepDNA (Wang et al. 2018), trained specifically on mitochondrial DNA data, using a convolutional layer to capture local features which are then combined and fed into a recurrent layer to output the probabilities for each symbol. On the other hand, autoencoders present an alternative to PPM. The first attempt to compress genomic data with simple autoencoders has been performed on sequencing reads data (Absardi and Javidan 2019), followed by a more sophisticated autoencoder-based method, GenCoder (Sheena and Nair 2024). To the authors' knowledge, there is no documented research leveraging autoencoders to losslessly compress SNP data sets.

Ancestry prediction

The relationship between ancestry, ethnicity, and genetic variation is complex, involving genetics, history, and society. Genetic variation does not follow identities established culturally and historically in a simple way (Roberts 2011). The term *ancestry* refers to a class of genetic similarity that can be associated with a shared origin (Xie et al. 2001; Fujimura and Rajagopalan 2011). The framework for classifying or regressing genetic ancestry uses the genome sequence for its features (Nelson et al. 2018). Indeed, an individual's ancestral geographic origin can be inferred with remarkable accuracy from their DNA (Novembre et al. 2008). This task, known as *global ancestry inference*, can be either approached from a discrete perspective (classifying the ancestry label) or from a continuous one (regressing the geographical coordinates of origin). Some widely adopted techniques include ADMIXTURE (Pritchard et al. 2000; Alexander et al. 2009), a clustering technique based on probabilistic non-negative matrix factorization, and its more recent neural counterpart—Neural ADMIXTURE (Dominguez Mantes et al. 2023); Locator (Battey et al. 2020), a multilayer perceptron (MLP) that addresses the *geographical coordinate regression* problem by estimating a nonlinear function mapping genotypes to locations; and Diet Networks (Romero et al. 2016), which represent another deep-learning-based ancestry classifier within this paradigm of methods. Nevertheless, characterizing ancestry as a set of discrete, predefined labels can be limiting, as individuals are at some level all admixed, stemming from ancestors belonging to multiple ancestral population groups (Supplemental Methods S1). For instance, the genomes of numerous African-Americans have variable proportions of segments that could be classified as European and West African ancestry (Ali-Khan and Daar 2010). To characterize these different genomic segments, one can use local ancestry inference (LAI) methods like RFMix (Maples et al. 2013). These can rely on neural networks as well, for instance LAI-Net (Montserrat et al. 2020) and SALAI-Net (Sabat Oriol et al. 2022).

Genomic sequence simulation

Although the number of sequenced genomes has grown substantially over the years, there is a clear disparity among the ancestries represented. The proportion of participants of non-European descent has remained constant (Wojcik et al. 2019), potentially introducing bias toward European genomes and giving rise to the "missing diversity" problem (Popejoy and Fullerton 2016). To illustrate, as of 2018, the majority of GWAS encompassed approximately 78% individuals of European ancestry. Additionally, certain communities, predominantly composed of individuals with non-European ancestry, are reluctant to participate in genetic studies due to privacy concerns or apprehensions about potential misuse, as seen in prior cases (Maher 2015; Guglielmi 2019). To circumvent these challenges, data simulation tools can be used to augment genomic databases and offer mechanisms for sharing synthetic data possessing equivalent statistical properties, all while safeguarding the privacy of individuals. Several recent studies have explored the effectiveness of generative deep neural networks in generating simulated genotypes. Montserrat et al. (2019) use a class-conditional VAE-GAN to generate artificial yet realistic genotypes, whereas Yelmen et al. (2021) generate high-quality synthetic genomes with GAN and RBM. Moment matching networks have provided competitive results for data simulation (Perera et al. 2022). Furthermore, another work by Battey et al. (2021) has attempted to use VAEs for genotype simulation.

To summarize the contributions: In this work, we demonstrate how a single model can address several critical tasks in genomics research. By leveraging the nonlinearities and the modular architecture of our VAE, we implicitly model linkage disequilibrium (LD) and facilitate interpretable latent structures. First, we illustrate how a VAE can be employed for lossless compression by storing the latent representation of the SNP sequences and their corresponding compressed residuals for error correction at decoding time. Our method represents the first instance of integrating autoencoders in existing compression pipelines, increasing the compression factors of large SNP data sets. Second, we introduce an ancestry-conditioned formulation of the VAEs and provide both qualitative and quantitative evaluations of the clustering quality in the latent space, comparing it to the commonly used principal component analysis (PCA), which still remains a strong baseline in population genetics (Tan and Atkinson 2023). Finally, we present an elegant method for sampling new SNP sequences from the modeled distribution, conditioned on ancestry, and contrast several metrics (including the LD structure and the SNP entropy) of the simulated sequences with that of real sequences as a genotype simulation quality assessment.

Methods

Variational autoencoders

Representation learning, also known as *feature learning*, attempts to recover a compact set of so-called latent q-dimensional variables $\boldsymbol{z}$ that describe a distribution over the d-dimensional observed data $\boldsymbol{x}$, with $q < d$. PCA is a well-established statistical procedure for dimensionality reduction and widely used in the population genetics community (Tan and Atkinson 2023). For a set of observed data $\boldsymbol{x}$, the latent variables $\boldsymbol{z}$ are the orthonormal axes onto which the retained variance under projection of the data points is maximal. PCA can be given a natural probabilistic interpretation as the dimensionality reduction process can be considered in terms of the distribution of the latent variables, conditioned on the observation (Tipping and Bishop 1999), where, from factor analysis, the relationship between $\boldsymbol{x}$ and $\boldsymbol{z}$ is linear and, conventionally, Gaussianity assumptions are taken. However, there are cases where the relationship between $\boldsymbol{x}$ and $\boldsymbol{z}$ is not linear and the common simplifying assumptions about Gaussianity of the marginal or posterior probabilities do not reflect real data. In those cases, autoencoders are a perfect fit because they learn a direct encoding—a parametric map from inputs $\boldsymbol{x}$ to their latent representation $\boldsymbol{z}$,

becoming a nonlinear generalization of PCA (Hinton and Salakhutdinov 2006). In this setup, two closed-form parametrized functions are defined: (a) the *encoder* $\mathcal{V}_e : \mathcal{X} \to \mathcal{Z}$ and (b) the *decoder* $\mathcal{V}_d : \mathcal{Z} \to \mathcal{X}$. Both $\mathcal{V}_e(\cdot)$ and $\mathcal{V}_d(\cdot)$ can be as expressive as desired: from a single linear layer to a MLP, or any other ANN architecture. In VAEs for SNP modeling, the input $\boldsymbol{x} \in \mathbb{B}^d$ is encoded with $\mathcal{V}_e(\cdot)$ into the mean $\boldsymbol{\mu}$ and a function of the variance $\boldsymbol{\sigma}$ vectors. Reparametrizing, the latent representation $\boldsymbol{z} = \boldsymbol{\mu} + \boldsymbol{\sigma} \odot \boldsymbol{\epsilon}$ is obtained, with $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$, $\odot$ being the elementwise product, and $\boldsymbol{z}, \boldsymbol{\mu}, \boldsymbol{\sigma}, \boldsymbol{\epsilon} \in \mathbb{R}^q$. The latent representation can then be decoded with the decoder $\mathcal{V}_d(\cdot)$ to obtain the reconstruction $\hat{\boldsymbol{x}} = \mathbb{1}_{1/2}(\boldsymbol{o})$, $\hat{\boldsymbol{x}} \in \mathbb{B}^d$, where $\boldsymbol{o} = \mathcal{V}_d(\mathcal{V}_e(\boldsymbol{x})) \in [0, 1]^d$ is the output of the network and $\mathbb{1}_{1/2}(\cdot)$ is a unit step function applied elementwise on the output, binarizing each element of the output with a threshold of 1/2. We define the composition of the encoder–decoder pair and the binarization layer as $f_{\boldsymbol{\theta}} = \mathbb{1}_{1/2} \circ \mathcal{V}_d \circ \mathcal{V}_e$, with parameters $\boldsymbol{\theta}$, such that $\hat{\boldsymbol{x}} = f_{\boldsymbol{\theta}}(\boldsymbol{x})$, and because our input is always a sequence of binary values, we have $f_{\boldsymbol{\theta}} : \mathbb{B}^d \to \mathbb{B}^d$. We adopt an under-complete architecture, in which the latent representation between the encoder and the decoder (traditionally referred to as the *bottleneck*; Vincent et al. 2010) has a smaller dimensionality than the input ($q < d$). In this setup, the primary learning objective is to compel the encoder to preserve as much of the relevant information as possible within this limited dimensionality.

Autoencoders learn the mapping function from input to feature space (the space spanned by $\boldsymbol{z}$) and the reverse mapping without learning an explicit probability distribution of the data. In contrast, VAEs learn a probability distribution of the data (Kingma and Welling 2014) by enforcing an isotropic Gaussian prior over the latent variables. Because they learn to model the data, new samples can be generated by sampling, meaning that VAE are *generative* autoencoders (Supplemental Methods S2, S5, and S6).

Ancestry-conditional VAEs

Note that the above approach does not condition VAE on ancestry labels during training—it trains on all populations together. However, an ancestry-conditioned VAE includes the additional information of the population label y_n for each input sample $\boldsymbol{x}_n$. This means we need to account for the labeled data set $\mathcal{D} = (\mathcal{D}_x, \mathcal{D}_y)$ in the loss function, where data set $\mathcal{D}_x = \{\boldsymbol{x}_n \mid 1 \le n \le N\}$ is the set of d-dimensional samples and $\mathcal{D}_y = \{y_n \mid 1 \le n \le N, y_n \in \mathcal{Y}\}$ is the set of corresponding ancestry labels, where $\mathcal{Y}$ is the set of populations present in the data. After incorporating ancestry conditioning in Equation 17, and following a similar derivation as in Equations 18 and 19 (Supplemental Methods S6), we arrive at the same generative loss objective for the ancestry-conditioned VAE:

$$\begin{aligned} p(\mathcal{D}_x \mid \mathcal{D}_y, \boldsymbol{\theta}) &= \prod_{n=1}^{N} p(\boldsymbol{x}_n \mid y_n, \boldsymbol{\theta}) \\ &= \prod_{k \in \mathcal{Y}} \prod_{n : y_n = k} p(\boldsymbol{x}_n \mid Y = k, \boldsymbol{\theta}). \end{aligned} \tag{1}$$

This indicates that the objective for the generative loss remains consistent in the ancestry-conditioned VAE:

$$\log p(\mathcal{D}_x \mid \mathcal{D}_y, \boldsymbol{\theta}) = -\sum_{k \in \mathcal{Y}} \sum_{n : y_n = k} \ell_{\mathrm{BCE}}(\boldsymbol{x}_n, \boldsymbol{o}_n^{(k)}), \tag{2}$$

denoting the VAE output conditioned on ancestry label k as $\boldsymbol{o}_n^{(k)} = f_\theta^{(k)}(\boldsymbol{x}_n)$. In practical terms, there are different ways to incorporate conditioning on the kth ancestry into both the encoder and the decoder, which we denote as $\mathcal{V}_e^{(k)}(\cdot)$ and $\mathcal{V}_d^{(k)}(\cdot)$, respectively. One approach, which we refer to as *regular* C-VAE (conditioned VAE), appends a one-hot encoded ancestry label to both the encoder and the decoder. An alternative approach for conditioning fits a separate VAE for each population group individually, resulting in ancestry-specific overfitted VAEs. We refer to this approach as Y-VAE (Y-overfitted VAE), a method which has $|\mathcal{Y}|$ number of times more parameters than a regular VAE.

Bayesian-motivated classification objectives

When employing ancestry conditioning on a VAE, an ancestry label can be inferred through maximum a posteriori (MAP) estimation. The output of the VAE conditioned on ancestry k is a vector containing Bernoulli probabilities $\boldsymbol{o}_n^{(k)} \in \mathbb{R}^d$. To infer an ancestry label, this vector is thresholded with a value greater than 1/2, resulting in $\hat{\boldsymbol{x}}_n$. Because each SNP position is independently modeled as a Bernoulli distribution, the joint distribution can be expressed as the product of individual SNP distributions. Applying Bayes' rule, we can derive $p(Y = k \mid \boldsymbol{x}_n) \propto p(\boldsymbol{x}_n \mid Y = k) p(Y = k)$, where $p(\boldsymbol{x}_n \mid Y = k)$ represents the Bernoulli likelihood and the prior $p(Y = k)$, for simplicity, is defined as the categorical distribution over K ancestry labels because the data used for the classification task have uniformly distributed ancestry labels. Therefore,

$$p(Y = k \mid \boldsymbol{x}_n) \propto \prod_{i=1}^{d} \left(o_{ni}^{(k)}\right)^{x_{ni}} \left(1 - o_{ni}^{(k)}\right)^{(1 - x_{ni})}. \tag{3}$$

Given that d represents a relatively large number of dimensions, numerical computation can lead to the entire expression collapsing to zero. This effect is due to two factors. First, if at least one SNP position is reconstructed incorrectly, that is, the true value of the SNP position is 0 but the VAE returns 1 or vice versa, the expression becomes zero. Second, when many positions have extremely small values due to numerical precision limitations, the product of these values becomes effectively zero. To address these issues and improve numerical stability, it is common practice to operate in the space of log-probabilities. We therefore apply logarithms to Equation 3:

$$\begin{aligned} \log p(Y = k \mid \boldsymbol{x}_n) &\propto \sum_{i=1}^{d} x_{ni} \log\left(o_{ni}^{(k)}\right) + (1 - x_{ni}) \log\left(1 - o_{ni}^{(k)}\right) \\ &= -\ell_{\mathrm{BCE}}(\boldsymbol{x}_n, \boldsymbol{o}_n^{(k)}). \end{aligned} \tag{4}$$

Selecting the ancestry label that maximizes the posterior $p(Y = k \mid \boldsymbol{x}_n)$ is equivalent to minimizing the BCE loss:

$$y_n = \arg\max_{k \in \mathcal{Y}} p(Y = k \mid \boldsymbol{x}_n) = \arg\min_{k \in \mathcal{Y}} \ell_{\mathrm{BCE}}(\boldsymbol{x}_n, \boldsymbol{o}_n^{(k)}). \tag{5}$$

To address the issues associated with the logarithm of zero and the potential misclassification of SNP positions, some form of smoothing or clamping is necessary. One way to implement a form of *Laplace smoothing* is by adjusting the sigmoid temperature, which makes the Bernoulli probabilities $o_{ni}^{(k)}$ less sharp and less likely to be clamped to zero quickly. However, even with this approach, there can still be issues with zeroing. To mitigate this, we employ the trick used in the PyTorch implementation of the BCE loss (Paszke et al. 2019). This involves clamping the probabilities to a certain value if they go beyond a specified threshold. Specifically, we clamp $p(Y = k \mid \boldsymbol{x}_n)$ to e^{-100}. In terms of logarithms, this is equivalent to clamping $\log p(Y = k \mid \boldsymbol{x}_n)$ to -100. This approach ensures that the loss remains finite and avoids the issues associated with infinite values due to the logarithm of zero.

Another important conclusion is that the Bernoulli likelihood maximization problem can be reduced to the minimization of the L_1 discrepancy between the input and the output of the VAE: Let $\ell_1(\boldsymbol{x}, \boldsymbol{o}^{(k)}) = ||\boldsymbol{x} - \boldsymbol{o}^{(k)}||_1$ be the L_1 discrepancy between the SNP array $\boldsymbol{x}$ and its Bernoulli probabilities by VAE conditioned on kth

ancestry. And let $b(\boldsymbol{x}, \boldsymbol{o}^{(k)})$ be the multivariate i.i.d. Bernoulli likelihood function. Then, for each ith element in $\boldsymbol{x}$:

i. $\ell_1(x_i, o_i^{(k)}) = o_i^{(k)}$ and $b_i(x_i, o_i^{(k)}) = 1 - o_i^{(k)}$ if $x_i = 0$.
ii. $\ell_1(x_i, o_i^{(k)}) = 1 - o_i^{(k)}$ and $b_i(x_i, o_i^{(k)}) = o_i^{(k)}$ if $x_i = 1$.

Thus, $b_i(x_i, o_i^{(k)}) = 1 - \ell_1(x_i, o_i^{(k)})$, from which follows the expression

$$b(\boldsymbol{x}, \boldsymbol{o}^{(k)}) = \prod_i \left(1 - \ell_1(x_i, o_i^{(k)})\right). \tag{6}$$

Taking logarithm on Equation 6 yields $\log b(\boldsymbol{x}, \boldsymbol{o}^{(k)}) = \sum_i \log(1 - \ell_1(x_i, o_i^{(k)}))$. The Taylor series of $\log(1 - z)$ about zero is

$$\begin{aligned} \log(1 - z) &\approx \log 1 + \frac{\partial}{\partial z} \log(1 - z) z + \mathcal{O}(z^2) \\ &= -z + \mathcal{O}(z^2). \end{aligned} \tag{7}$$

For small z, all terms of order z^2 are negligible and we can employ the approximation $\log(1 - z) \approx -z$. Therefore, $\arg\min \ell_{\text{BCE}}(\boldsymbol{x}, \boldsymbol{o}^{(k)})$ can be approximated by $\arg\min \ell_1(\boldsymbol{x}, \boldsymbol{o}^{(k)})$. That is, the multivariate i.i.d. Bernoulli likelihood maximization problem can be approximately reduced to the minimization of the L_1 discrepancy between the binary input and output of the VAE.

Results

Compression factor with VAE improves over PCA-based approach

Like many natural signals, SNP sequences can be viewed as realizations of a stochastic process. Such data exhibit a particularly high level of redundancy and correlation, owing in part to LD. For that reason, we have conducted a comprehensive entropy analysis to understand the statistical nature of SNP sequences and to motivate how population-specific genetic diversity influences compression performance, thereby connecting these observations to the main theme of efficient genomic data modeling with VAE. A detailed description of the employed data sets can be found in Supplemental Methods S4.

To estimate the entropies, we leverage the fact that for a Bernoulli random variable, its distribution parameter is given by the mean of the random variable. To avoid bias from the unbalancedness of the data set, we compute the entropy estimates per population by bootstrapping 32 samples from the founders' pool 50 times and average them. The choice of 32 is significant as it is half of the size of the smallest human superpopulation in the data set, and the choice of 50 corresponds approximately to the fraction between the size of the largest human continental population divided by 32. The resulting density plots of the entropy rates for human Chromosome 22 for each superpopulation are depicted in Figure 1A. Among these populations, the African population (AFR) presents the highest variability in SNP values and, thus, highest entropy values per SNP. In contrast, the Oceanian (OCE) and Native American-like (AMR) populations exhibit the lowest values of entropy. These results align with the out of Africa (OOA) theory (Nielsen et al. 2017), suggesting that populations that migrated out of Africa experienced a reduction in genetic diversity and an increase in LD. Averaging the entropy vectors for each population yields the average SNP entropy per population, as shown in Figure 1B. These values signal an interesting connection to historical human migrations, as depicted in Figure 1C. Starting with African individuals, migrations led humans to expand to West Asia (WAS), South Asia (SAS), Europe (EUR), and East Asia (EAS). In these out-of-Africa regions, populations experienced a noteworthy reduction in average SNP entropy, reflecting a decrease in genetic diversity over time likely due to founder effects. As time progressed, subsequent migrations brought individuals to Oceania (OCE) and the Americas (AMR), culminating in populations with minimal genetic variability, as evidenced by their lower average SNP entropy. This observation aligns with the notion that genetic diversity tends to decrease as populations migrate further away from their ancestral origins and undergo genetic bottlenecks.

Based on the VAE architecture, the decoder obtains the reconstruction $\hat{\boldsymbol{x}}$, which is a lossy representation of $\boldsymbol{x}$. To achieve lossless compression, it is necessary to store the residual, which is the elementwise difference between the input and the reconstruction, denoted as $\boldsymbol{r} = |\boldsymbol{x} - \hat{\boldsymbol{x}}|$, with $\boldsymbol{r}, \boldsymbol{x}, \hat{\boldsymbol{x}} \in \mathbb{B}^d$. This residual is used for error correction of the reconstruction, and with a high-generalization level, a well-trained VAE can produce a sufficiently sparse $\boldsymbol{r}$. This sparsity can be compressed, in its turn, with another lossless algorithm $\mathcal{A}$. For the sake of simplicity, in our PCA-VAE comparison experiments, we use run-length encoding (RLE) for $\mathcal{A}$. We

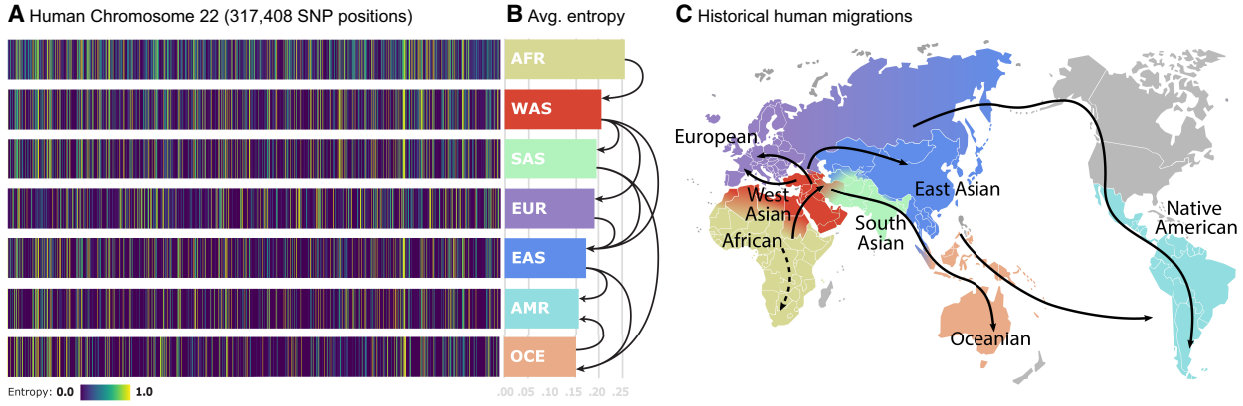

Figure 1. SNP entropy variation between populations. (*A*) We find 317,408 SNP entropies computed for human Chromosome 22 for each continental population. Observe that the uncertainty levels for particular SNP positions are different for each population. This is directly related to genetic variability. (*B*) Average of entropy vectors from *A*. Those values would correspond to estimated lower bounds of compression for each human population. Arrows represent the migration paths. (*C*) World map showing the main directions of human population migrations.

consider a compression execution successful when the inequality from Equation 8 holds:

$$\ell(\boldsymbol{z}) + \ell(\mathcal{A}(\boldsymbol{r})) < \ell(\boldsymbol{x}), \tag{8}$$

where $\ell(\cdot)$ is the *size function*, which computes the number of digits of the binary representation according to the data type. The latent vector $\boldsymbol{z}$ is composed of floating-point latent factors, whereas the reconstructed bits can be stored as Booleans, which, in the best case scenario, should be sparse enough to enable compression with RLE into a smaller integer array.

Table 1 presents the results of human SNP compression for sequences of length of 10,000 SNPs, comparing PCA versus VAE models trained genome-wide. These models were fit to single-ancestry simulated data with 400 generations from founders with 100 individuals in each generation. As observed, VAEs with a bottleneck dimensionality of 32, 64, and 128 latent factors are capable of compressing all populations, including the African population (AFR) which exhibits the highest degree of variability. Notably, European (EUR) and Native American-like (AMR) ancestries can be compressed to half their original size. In contrast, PCA, being a linear method, struggles to reconstruct effectively from $\boldsymbol{z}$, resulting in a residual vector $\boldsymbol{r}$ that is not sufficiently sparse. Consequently, PCA leads to an expansion in size by factors of 2×, 3×, and 4×, rather than achieving compression. The VAE takes advantage of the nonlinearities in the decoder for improved reconstruction. We also explored compression using a C-VAE (Supplemental Results S1 and Supplemental Table S3).

Leveraging autoencoders for lossless SNP compression in practice

The presented results inspire us to push even further the limits of genotype compression. Whereas previously we have used RLE for residuals, in practical production settings, we should transition to more efficient coding methods. Additionally, recognizing that a discrete representation often consumes less memory than a continuous one, we explore the concept of vector quantization within the autoencoder framework, known as VQ-VAE (van den Oord et al. 2018). We show that introducing the autoencoder into existing compression pipelines, such as Genozip (Lan et al. 2021) or

Table 1. Compression factors of PCA versus VAE

Models			Populations						
Type	$\lvert z\rvert$	α	European (EUR)	East Asian (EAS)	Native American (AMR)	South Asian (SAS)	African (AFR)	Oceanian (OCE)	West Asian (WAS)
PCA	2	–	×0.41	×0.38	×0.42	×0.39	×0.33	×0.36	×0.38
VAE		10^{-4}	×0.68	×0.64	×0.76	×0.62	×0.53	×0.50	×0.67
		10^{-5}	×0.65	×0.61	×0.73	×0.60	×0.52	×0.49	×0.63
PCA	4	–	×0.41	×0.37	×0.41	×0.38	×0.33	×0.36	×0.38
VAE		10^{-4}	×0.77	×0.69	×0.87	×0.68	×0.56	×0.58	×0.71
		10^{-5}	×0.73	×0.69	×0.81	×0.66	×0.54	×0.63	×0.68
PCA	8	–	×0.41	×0.37	×0.41	×0.38	×0.33	×0.36	×0.37
VAE		10^{-4}	×1.00	×0.93	**×1.17**	×0.88	×0.63	×0.68	×0.89
		10^{-5}	×0.96	×0.90	**×1.08**	×0.88	×0.63	×0.74	×0.88
PCA	16	–	×0.40	×0.36	×0.41	×0.38	×0.32	×0.35	×0.37
VAE		10^{-4}	**×1.59**	**×1.39**	**×1.73**	**×1.32**	×0.84	**×1.01**	**×1.37**
		10^{-5}	**×1.25**	**×1.12**	**×1.35**	**×1.04**	×0.70	×0.90	**×1.08**
PCA	**32**	–	×0.39	×0.36	×0.40	×0.37	×0.32	×0.34	×0.36
VAE		**10^{-4}**	**×2.00**	**×1.75**	**×2.33**	**×1.69**	**×1.03**	**×1.27**	**×1.72**
		10^{-5}	**×1.67**	**×1.45**	**×1.85**	**×1.39**	**×1.85**	**×1.23**	**×1.28**
PCA	**64**	–	×0.37	×0.34	×0.38	×0.35	×0.31	×0.33	×0.34
VAE		**10^{-4}**	**×2.04**	**×1.82**	**×2.27**	**×1.75**	**×1.16**	**×1.47**	**×1.82**
		10^{-5}	**×1.69**	**×1.54**	**×1.96**	**×1.47**	×0.96	**×1.30**	**×1.49**
PCA	**128**	–	×0.34	×0.32	×0.35	×0.32	×0.29	×0.30	×0.32
VAE		**10^{-4}**	**×1.54**	**×1.45**	**×1.61**	**×1.41**	**×1.06**	**×1.25**	**×1.43**
		10^{-5}	**×1.47**	**×1.37**	**×1.56**	**×1.35**	×0.97	**×1.20**	**×1.37**
PCA	256	–	×0.30	×0.28	×0.30	×0.28	×0.25	×0.27	×0.28
VAE		10^{-4}	×0.97	×0.93	×1.00	×0.93	×0.79	×0.86	×0.93
		10^{-5}	×0.94	×0.90	×0.97	×0.89	×0.73	×0.84	×0.90
PCA	512	–	×0.24	×0.23	×0.24	×0.23	×0.21	×0.22	×0.23
VAE		10^{-4}	×0.54	×0.53	×0.55	×0.53	×0.48	×0.50	×0.53
		10^{-5}	×0.53	×0.52	×0.54	×0.51	×0.45	×0.49	×0.52

The compression factors are computed as $\ell(\boldsymbol{x})/(\ell(\boldsymbol{z}) + \ell(\mathcal{A}(\boldsymbol{r})))$ using test data. A compression ratio of 1 corresponds to the identity, and values <1 and >1 correspond to compression and expansion, respectively. VAEs with a bottleneck of 32, 64, and 128 latent factors are capable of lossless compression of all human populations. Successful compression is marked in bold. $\lvert\boldsymbol{z}\rvert$ is the number of latent factors and α stands for *learning rate*.

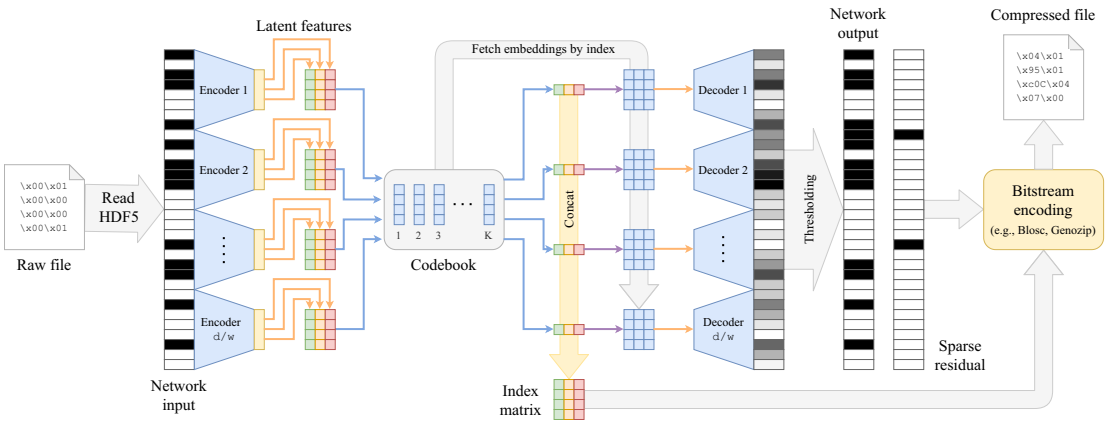

Figure 2. Proposed VQ-VAE architecture for genotype compression. The window-based VQ-VAE autoencoder processes an input SNP sequence $\boldsymbol{x}$ and encodes with $\mathcal{V}_e(\cdot)$ into H bottleneck representations (H is the number of heads in the encoder). The quantizer $\mathcal{Q}$ substitutes the bottleneck representations by the closest codebook embeddings. Finally, the latent representation can be encoded as an integer index matrix. For the decoding step, codebook embeddings are fetched according to the indices of the index matrix and decoded as usual with the window-based autoencoder. The output is thresholded to obtain the reconstruction. The difference of the input with the reconstruction yields the residual $\boldsymbol{r}$ which, together with the index matrix, can be integrated in any bitstream-coding-based compression pipeline, such as Genozip (Lan et al. 2021), Zstandard (Collet and Kucherawy 2018), or Blosc (https://www.blosc.org).

Lempel–Ziv codecs (Collet and Kucherawy 2018; http://www.blosc.org), can lead to significant improvements in compression factors for large SNP data sets (Fig. 2).

To discretize the latent representation of $\boldsymbol{x}$, we employ a quantizer denoted as $\mathcal{Q}$. This quantizer represents $\boldsymbol{x}$ as a matrix of positive integer indices, which point to a specific set of embeddings, as described in van den Oord et al. (2018). Formally, $(\mathcal{Q} \circ \mathcal{V}_e) : \mathbb{B}^d \longrightarrow \mathbb{Z}_+^{H \times \lceil d/w \rceil}$, where d is the input dimensionality, w is the window size, and H corresponds to the number of heads in each window-encoder. With a uniform prior, the latent representation is defined by indices pointing to a fixed set of embeddings. We define the codebook size as $K \times q$, where K is the number of embeddings, and q signifies the bottleneck dimensionality. We opt for utilizing multihead encoders with H heads. The rationale behind incorporating multiple heads is that the number of potential representations is determined by K^H. By choosing $H = 1$, we would limit the possible representations to K embeddings, which might lead to different inputs mapping to the same quantized latent vector, resulting in nonsparse residuals.

In the context of discrete-space autoencoders, it is important to note that the $\mathcal{Q}$ operator is not differentiable. To address this limitation, we employ the straight-through gradient estimator, which allows for gradient flow during backpropagation. Additionally, within the VQ-objective, we introduce two additional loss terms: (a) the *embedding loss*, which encourages the embeddings to align with the encoder outputs, and (b) the *commitment loss*, which incentivizes the encoder to output $\boldsymbol{z}$ closer to the codebook embeddings $\boldsymbol{e}_k$, $1 \le k \le K$, where K is the size of the codebook and ξ denotes the stop-gradient operator:

$$\mathcal{L}_{\mathrm{VQ}}(\boldsymbol{x}, \boldsymbol{o}) = \mathcal{L}_{\mathrm{BCE}}(\boldsymbol{x}, \boldsymbol{o}) + \underbrace{\|\xi[\boldsymbol{z}] - \boldsymbol{e}_k\|_2^2}_{\text{Embedding loss}} + \underbrace{\|\boldsymbol{z} - \xi[\boldsymbol{e}_k]\|_2^2}_{\text{Commitment loss}}. \quad (9)$$

The stop-gradient operator ξ acts as the identity during forward computation and has zero partial derivatives during backpropagation, treating its operand as a nonupdated constant (van den Oord et al. 2018), to avoid collapsing the optimization process because $\boldsymbol{z}$ and $\boldsymbol{e}_k$ are mutually related, and both need to be optimized.

For production purposes, we consider window-based autoencoders with nonoverlapping windows. These autoencoders have two main hyperparameters: the window size w and the bottleneck size b for each window. Therefore, the index matrix maintains a fixed size of $H \times \lceil d/w \rceil$. In our window-based architecture, an important decision revolved around the selection of values for w and b. A larger value for b results in more information being compressed into the latent representation, leading to improved reconstruction of $\boldsymbol{x}$ and consequently a sparser residual. A sparser residual can be better encoded with a coding algorithm $\mathcal{A}$ because of the longer homogeneous regions of zeros. Concerning the choice of w, we performed a heuristic analysis to identify an appropriate window size, settling on $w = 2500$. We first calculate the *average intrawindow entropy* (which depends on the number of unique sequences in a window) for each candidate window size in the set $w \in \{50, 100, 500, 1000, 2500, 5000, 7500, 10{,}000\}$. This yields a list of window sizes $\boldsymbol{w} = [w_1, w_2, \ldots, w_n]$ and a corresponding list of average intrawindow entropies across human Chromosome 22 for each choice (see Fig. 3), $\boldsymbol{E} = [E_1, E_2, \ldots, E_n]$. Our rationale is that smaller windows would require a larger bottleneck to represent at least the same amount of information. For instance, if a binary sequence has length 18, using $w = 3$, the smallest possible bottleneck is going to be of size 6, whereas using $w = 6$, the smallest possible bottleneck is going to be of size 3. Consequently, there is an inherent trade-off between window size and the complexity needed in the model. To formalize our selection, we computed the difference in consecutive window sizes: $\Delta w_i = w_{i+1} - w_i$ and the difference in consecutive intrawindow entropies: $\Delta E_i = E_{i+1} - E_i$. For each interval i, we combine Δw_i and ΔE_i (e.g., via their product, $\Delta w_i \times \Delta E_i$) to gauge how *much* extra entropy is gained when the window size changes from w_i to w_{i+1}, and we look for the index i that maximizes this combined metric. Practically, this represents an "elbow," beyond which increasing the window size further does not substantially alter the average intrawindow entropy. Thus, we aim to strike a balance between *model simplicity* (not using an overly large window size, which might lose fine-grained structure) and *encoding efficiency* (not using an excessively

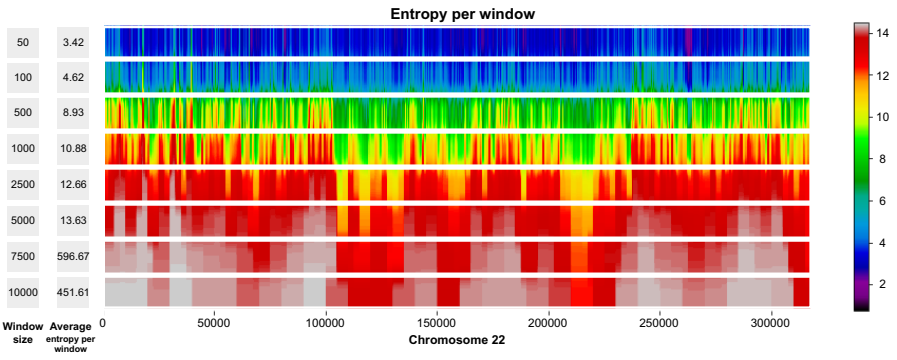

Figure 3. Average entropy per window across different window sizes on Chromosome 22. For each window size $w \in \{50, 100, 500, 1000, 2500, 5000, 7500, 10{,}000\}$, we compute the average entropy per window over the entire chromosome.

small window size, which would require a large bottleneck to capture all local information). In our analysis, $w = 2500$ results in a good compromise between capturing local genomic structure and maintaining a reasonable model bottleneck size.

The window-based encoder $\mathcal{V}_e$ takes as input a SNP sequence $\boldsymbol{x}$ of a specific length d and compresses it to the dimensionality of the bottleneck $q = d/w \cdot b$. In this manner, we obtain the vector of latent factors $\boldsymbol{z}$, which in its turn is quantized with $\mathcal{Q}$. Next, the decoder $\mathcal{V}_d$ reconstructs the input with some errors, $\hat{\boldsymbol{x}} = \mathbb{1}_{1/2}(\boldsymbol{o})$, where $\boldsymbol{o}$ represents the network's output. As before, storing the residual $\boldsymbol{r} = |\boldsymbol{x} - \hat{\boldsymbol{x}}|$ allows for a lossless compression of $\boldsymbol{x}$, as the reconstruction errors can be corrected using the residual. With a sufficiently large q, the residual $\boldsymbol{r}$ becomes sparse, making it amenable to compression. This sparsity can be compressed further with a bitstream coding lossless algorithm $\mathcal{A}$, along with the latent representation $\boldsymbol{z}$. The celebrated Lempel–Ziv algorithm belongs to the class of universal compression schemes, and in our experiments, we use its variations for the role of $\mathcal{A}$.

In this case, we consider a compression execution successful when Equation 10 holds (note the difference with Eq. 8):

$$\ell(\mathcal{A}(\boldsymbol{z})) + \ell(\mathcal{A}(|\boldsymbol{x} - \hat{\boldsymbol{x}}|)) = \ell(\mathcal{A}(\boldsymbol{z})) + \ell(\mathcal{A}(\boldsymbol{r})) < \ell(\boldsymbol{x}). \tag{10}$$

In the proposed method (Fig. 2), the window-based VQ-VAE autoencoder is composed of three hidden layers. Each fully connected layer is followed by a batch normalization layer (Ioffe and Szegedy 2015). The activation preceding the bottleneck is transformed by means of the hyperbolic tangent function, to a range which is useful for quantization in the context of discrete latent spaces. The activation at the output of the network is a sigmoid, which converts the activations into Bernoulli probabilities. All the other activation units in intermediate layers are ReLUs. At the beginning of the training process, all the weights and biases are initialized with Xavier initialization (Glorot and Bengio 2010). We employ the Adam optimizer (Kingma and Ba 2017) with the best-performing learning rate of $\alpha = 0.025$ and a weight decay of $\gamma = 0.01$. A scheduler has been set to reduce the learning rate by a factor of $\gamma = 0.1$ in the event of learning stagnation. Furthermore, a dropout (Srivastava et al. 2014) of 50% has been introduced in all layers because it offers a better generalization providing larger compression factors (Supplemental Methods S8).

We benchmark the performance of our autoencoder + bitstream coding compression strategy against several compression methods, namely: Gzip (general-purpose), ZPAQ (Mahoney 2005) (for text), Zstandard (Collet and Kucherawy 2018; https://www.blosc.org) (general-purpose), bref3 (Browning et al. 2018) (for VCF), and Genozip (Lan et al. 2021) (general-purpose optimized for genomic data). We evaluate the compression on four different test sets, each containing 11,772 simulated individuals generated using Wright–Fisher simulation. These test sets consisted of HDF5 files with different numbers of SNPs from human Chromosome 22: 10,000 SNPs, 50,000 SNPs, 80,000 SNPs, and the entirety of human Chromosome 22 (317,400 SNPs). For compression with bref3 (Browning et al. 2018), we had to run an additional preprocessing step which would convert the HDF5 into a VCF file (the conversion runtime is not included in the benchmark). The results of this benchmark are summarized in Table 2, highlighting the advantages of incorporating autoencoders within compression pipelines.

Dimensionality reduction with VAE

Genotypes can unravel population structure. The identification of genetic clusters can be important when performing GWAS and provides an alternative to self-reported ethnic labels, which are culturally constructed and vary according to the location and individual. A variety of unsupervised dimensionality reduction methods have been explored in the past for such applications, including PCA, MDS, t-SNE (Van der Maaten and Hinton 2008), and UMAP (McInnes et al. 2018). Recently, VAEs have been introduced into population structure visualization (Battey et al. 2021; Meisner and Albrechtsen 2022). Battey et al. 2021). The singular feature of VAEs is that they can represent the population structure as a Gaussian-distributed continuous multidimensional representation and as classification probabilities providing flexible and interpretable population descriptors. Besides, latent maps allow for meaningful interpretation of distances between ancestry groups. Although it is true that proximity in the latent space cannot be directly interpreted as proportional to similarity—a recurrent issue highlighted in nonlinear dimensionality reduction techniques, such as t-SNE and UMAP (Battey et al. 2021; Chari and Pachter 2023) but also present in PCA (Elhaik 2022; Montserrat and Ioannidis 2023)—the implicit regularization of the optimization process of VAEs and the capacity of the encoder/decoder can limit the distortion on the local and global distances and, as observed experimentally, such projections can still provide insights in the structure of the data which are discussed next.

We quantitatively assess the quality of the VAE clusters by comparing the clustering performance of PCA and VAE clusters. We use the pseudo F statistic (Caliński and Harabasz 1974), the Davies–Bouldin index (DBI) (Davies and Bouldin 1979), and the silhouette coefficient (SC) (Kaufman and Rousseeuw 2009) as clustering metrics (Supplemental Methods S7). We use two principal components to cluster populations, and although two might be a small number of components for data explainability, the quantitative (Table 3) and qualitative (Fig. 4) results show that two VAE components retain more information than two PCA components.

Visually, the VAE clusters still preserve the geographic vicinity of adjacent human populations and, additionally, discriminate more than PCA some subpopulations within each cluster, as it can

Table 2. Compression benchmark for subsets of SNPs of human Chromosome 22

Data	10,000 SNPs	50,000 SNPs	80,000 SNPs	317,400 SNPs
Original size	112.27	561.33	898.14	3536.40
Gzip (clevel 9)	6.48 (×17.3) [1 m 0.691 s]	40.68 (×13.8) [6 m 9.370 s]	65.20 (×13.8) [10 m 45.854 s]	263.30 (×13.4) [44 m 5.341 s]
ZPAQ (clevel 3) (Mahoney 2005)	5.92 (×18.9) [1 m 59.611s]	28.83 (×19.5) [9 m 52.687 s]	45.18 (×19.9) [24 m 39.143 s]	183.38 (×19.3) [98 m 46.042 s]
Zstandard (Collet and Kucherawy 2018)	11.29 (×9.9) [0 m 0.209 s]	57.08 (×9.8) [0 m 1.017 s]	92.75 (×9.7) [0 m 2.143 s]	372.74 (×9.5) [0 m 6.535 s]
Genozip (Lan et al. 2021)	**0.94 (×119.4)** [0 m 12.899 s]	29.89 (×18.8) [0 m 2.681 s]	48.67 (×18.5) [0 m 3.249 s]	200.13 (×17.7) [0 m 11.741 s]
bref3 (Browning et al. 2018)	4.35 (×25.8) [0 m 1.383 s]	**19.91 (×28.2)** [0 m 4.322 s]	**27.31 (×32.9)** [0 m 10.709 s]	**115.52 (×30.6)** [0 m 22.916 s]
VQ-VAE + Zstandard (ours)	**3.42 (×32.83)** [0 m 12.905 s]	25.37 (×22.12) [1 m 0.564 s]	40.17 (×22.4) [1 m 42.669 s]	160.68 (×22.0) [6 m 37.681 s]
VQ-VAE + Genozip (ours)	3.59 (×31.3) [0 m 6.984 s]	**19.44 (×28.9)** [0 m 14.447 s]	**27.77 (×32.3)** [0 m 26.471 s]	**115.24 (×30.7)** [1 m 23.828 s]

The file size in MB is compared between methods, along with its compression factor and running time. We mark in bold the top two choices based on compression factors.

be clearly observed in the case of African (AFR) and Native American-like (AMR) populations in Figure 4B. Therefore, VAE projections to the two-dimensional space allow for a more insightful and fine-grained exploratory analysis. As an example, having a closer look to the aforementioned ancestry groups, shown in Figure 4B, observe that PCA is not capable of differentiating their subpopulations. In contrast, VAE clusters in the African population clearly distinguish Mbuti and Biaka subpopulations, which both are hunter gatherer populations from the central African cluster (Supplemental Figs. S3–S5). Another interesting visualization is the projection of canine genotypes to the VAE latent space (Fig. 4A). For instance, the Asian Spitz clade is found closer to the wolves, which suggests their genetic similarity as they were one of the first domesticated canids (Yang et al. 2017). The list of human and canine populations can be found in Supplemental Tables S1 and S2.

Regarding other nonlinear manifold learning techniques, such as t-SNE (Van der Maaten and Hinton 2008) and UMAP (McInnes et al. 2018), these are not directly comparable with our approach as they do not allow mapping new samples to the spanned space, which makes them not usable for the tasks of compression, classification, and simulation. The reason is that both methods learn a nonparametric mapping; that is, they do not learn an explicit function that maps data from the input space to the map (Supplemental Fig. S6). Therefore, it is not possible to embed test points in an existing map.

Table 3. Comparison of clustering performance of PCA versus VAE

	Data type	Human data	Canine data
Pseudo *F* statistic	PCA	50,315.56	151.89
	VAE	**56,409.29**	**276.03**
Silhouette coefficient	PCA	0.69	**0.12**
	VAE	**0.77**	0.07
Davies–Bouldin index	PCA	0.48	3.87
	VAE	**0.29**	**3.39**

PCA and VAE parameters have been fitted to human and canine SNP data sets of 839,629 and 198,473 SNP positions, respectively. Clustering metrics have been computed on seven self-reported human ancestry groups and 16 canine clades composed of 144 distinct canine breeds. The 2D latent coordinates of the samples have been standardized. Bold values indicate the better-performing method for each metric and data type (higher is better for Pseudo *F* and Silhouette; lower is better for Davies–Bouldin).

Different objectives for ancestry classification

Individuals that are more closely related in ancestry have spatial autocorrelations in their genomic sequences. This phenomenon is translated into population clusters using dimensionality reduction techniques (Fig. 4). The most common approaches to address ancestry classification and regression are based on dimensionality reduction and clustering techniques (Tan and Atkinson 2023). Genomic sequences from known and unknown origins are jointly analyzed—unknown samples are assigned to the nearest labeled cluster of the feature space (e.g., PCA space). Yet there are some caveats (Baran et al. 2013; Battey et al. 2020): (a) The results can be nonsensical if the individuals to classify are admixed, that is, are descendants of individuals from different ancestries, or do not originate from any of the sampled reference populations (*out-of-sample*); (b) commonly used dimensionality reduction techniques such as PCA do not model LD. Correlations induced by LD violate the inherent assumptions of independency between SNP positions. The cumulative effect of those correlations not only decreases accuracy but can also bias the results, a fact that can be observed in PCA projections of sequential SNP positions compared to SNP positions selected at random (Supplemental Figs. S7 and S8). In contrast, VAE, the nonlinear counterpart of PCA, is able to model up to a certain degree these induced correlations (see Fig. 5B) allowing for less LD bias with a relatively small number of SNP positions as input and consequently, yielding better visualizations.

For the classification task, we have trained VAEs on 10,000 sequential SNP positions from human Chromosome 22. The learned representations with VAEs have been assessed based on the population labels we have for each individual. We provide a quantitative evaluation of these learned representations using different classification approaches which are described in detail next.

The simplest idea for classification in the latent space is computing the population centroids and assigning each individual to the nearest one. We refer to this method as the *nearest latent*

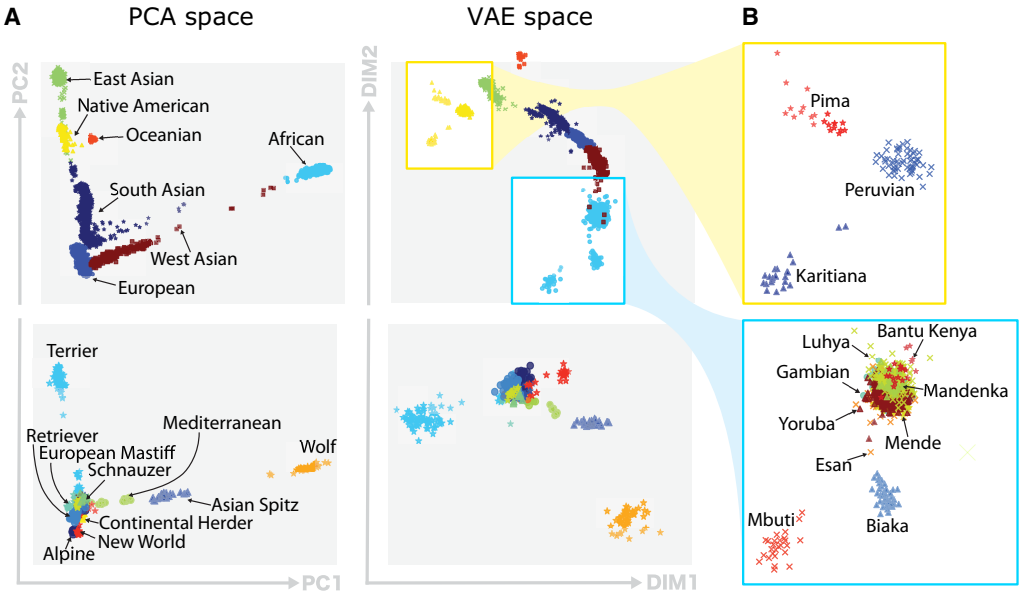

Figure 4. Qualitative comparison of PCA and VAE projections. (*A*) The *top* row illustrates the projections generated by both PCA and VAE for 4894 human samples using 839,629 SNPs. The *second* row displays projections of 489 canine samples using 198,473 SNP positions. (*B*) Focus of VAE projections of Native American-like subpopulations (in yellow) and African subpopulations (in blue).

centroid heuristic. It is a form of nearest-neighbor classification based on the latent features extracted by the VAE. Each centroid $\boldsymbol{c}_k$, $1 \le k \le |\mathcal{Y}|$, is computed as

$$\boldsymbol{c}_k = \frac{\sum_{n=1}^{N} \mathbb{1}_k(\boldsymbol{x}_n)\boldsymbol{z}_n}{\sum_{n=1}^{N} \mathbb{1}_k(\boldsymbol{x}_n)}, \tag{11}$$

where $\boldsymbol{z}_n = \mathcal{V}_e(\boldsymbol{x}_n)$ and $\mathbb{1}_k(\cdot)$ is an indicator function that denotes membership to ancestry label k. For each $\boldsymbol{x}_n$, we assign the label that minimizes the distance between the latent representation and the corresponding centroid:

$$y_n = \arg\min_{k \in \mathcal{Y}} \|\boldsymbol{z}_n - \boldsymbol{c}_k\|^2. \tag{12}$$

In the classification results presented in this study, we set $|\mathcal{Y}| = 7$ as we have seven human superpopulations (continental) groups in the provided data set. We hypothesized that each ancestry-conditioned VAE would better reconstruct the population to which it belongs. In the case of Y-VAE, each independent VAE learns to reconstruct better the population on which it has been trained, which is translated into the minimization of the L_1 norm between the input SNP array and the reconstruction. Let us denote the composition of encoder $\mathcal{V}_e^{(k)}(\cdot)$, decoder $\mathcal{V}_d^{(k)}(\cdot)$, and binarization $\mathbb{1}_{1/2}(\cdot)$ functions, as $f_{\boldsymbol{\theta}}^{(k)}(\cdot)$ to which we refer to as the VAE model conditioned on kth ancestry with parameters $\boldsymbol{\theta}$. Then, the ancestry of the nth sample:

$$\begin{aligned} y_n &= \arg\min_{k \in \mathcal{Y}} \|\boldsymbol{x}_n - \hat{\boldsymbol{x}}_n^{(k)}\|_1 \\ &= \arg\min_{k \in \mathcal{Y}} \|\boldsymbol{x}_n - f_{\boldsymbol{\theta}}^{(k)}(\boldsymbol{x}_n)\|_1 \end{aligned}. \tag{13}$$

With such conditioning, a Bayesian parameter estimation approach can be adopted for ancestry label inference via MAP

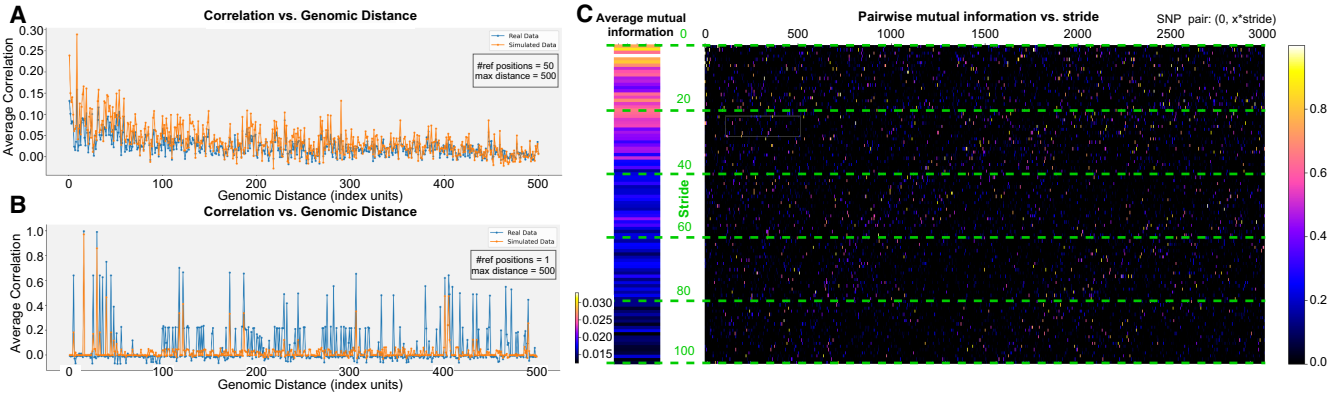

Figure 5. Metrics for assessing LD structure and inter-SNP correlation. (*A*) Average correlation versus genomic distance for real (blue) and simulated (orange) genotypes. We select multiple reference SNPs and plot the correlation with neighboring positions up to specified maximum distances (in the example, we use a distance of 500 with 50 reference positions). (*B*) Correlation of a single SNP position, illustrating how well the synthetic data reproduce the LD structure. (*C*) Pairwise mutual information between a reference SNP and other SNPs (*x*-axis) separated by varying stride lengths (*y*-axis), so that the resulting value corresponds to the SNP pair $(0, x \cdot y)$. The color scale indicates the magnitude of mutual information, from negligible (dark) to higher (bright) values.

estimation. Refer to the "Methods" section for the complete mathematical derivation.

The encoder–decoder architecture, when conditioned on kth ancestry, produces a vector of Bernoulli probabilities $\boldsymbol{o}_n^{(k)}$, which is subsequently thresholded to generate $\hat{\boldsymbol{x}}_n^{(k)} = \mathbb{1}_{1/2}(\boldsymbol{o}_n^{(k)})$. By training the network with the BCE loss, we assume that each individual SNP position x_{ni}, $1 \le i \le d$, follows a Bernoulli distribution. This assumption allows us, in theory, to calculate the Bernoulli likelihood for each SNP position. A MAP estimate of Y is the one that maximizes the posterior probability $p(Y=k|\boldsymbol{x}_n)$ for a given sample $\boldsymbol{x}_n$, which is given by

$$\begin{aligned} y_n &= \arg\min_{k\in\mathcal{Y}} p(Y=k\,|\,\boldsymbol{x_n}) \\ &\propto \arg\min_{k\in\mathcal{Y}} \sum_{i=1}^{d} x_{ni}\log o_{ni}^{(k)} + (1-x_{ni})\log\left(1-o_{ni}^{(k)}\right) \\ &= \arg\min_{k\in\mathcal{Y}} \ell_{\mathrm{BCE}}(\boldsymbol{x}_n, \boldsymbol{o}_n^{(k)}). \end{aligned} \tag{14}$$

Note that the BCE minimization problem can be approximated by the minimization of the L_1 discrepancy between the input and the output of the VAE. Refer to the "Methods" section for details.

To conclude the classification methods section, we present the results for each method in Table 4. The first two rows compare the *nearest latent centroid* approach in PCA and VAE spaces. Notably, there is a substantial improvement when using the VAE-generated space instead of PCA. The overall test accuracy increases from 74.1% to 85.7%, representing a >15% increase in accuracy. Both C-VAE and Y-VAE are evaluated based on two criteria: the minimization of the discrepancy between the input and the reconstruction, and the maximization of the Bernoulli likelihood. Among these, C-VAE, which maximizes the BCE loss, demonstrates the best performance across all populations, achieving an accuracy of 87.1% on the test data. It is worth noting that Oceanian (OCE) and West Asian (WAS) populations exhibit the lowest classification accuracy. Coincidentally, those two populations had the smallest number of founders for simulation. One possible explanation for this phenomenon is that the variability within the simulated samples is insufficient to provide robust generalization for the VAE, necessitating a larger number of founders for improved performance.

Synthetic data generated by VAE

Generating synthetic data with VAE methods is reasonably straightforward: one *could* simply sample $z \sim \mathcal{N}(0, 1)$ and decode to generate synthetic SNP sequences. However, our goal is to provide a mechanism for producing samples *specific to a given ancestry*. In the initial approach, a regular VAE (without conditioning) has been used to compute the population-specific centroids and variances in learned latent space, $\boldsymbol{\mu}$ and $\boldsymbol{\sigma}^2$, respectively. With these central points for each cluster, we sample from the isotropic multivariate Gaussian distribution, $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\sigma}^2\boldsymbol{I})$, and decode the resulting latent vector with the VAE decoder $\mathcal{V}_d(\cdot)$. However, there is no inherent reason to assume each population forms a Gaussian cluster and this approach does not yield distinct clusters because the distances between the centroids are not sufficiently large to differentiate between populations (Fig. 6A). Based on the insights gained from our simulation experiments, we have recognized that explicit conditioning is essential. The refined simulation algorithm involves sampling a multivariate Gaussian vector, $\boldsymbol{z} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{1})$, and then conditioning the decoder $\mathcal{V}_d^{(k)}(\cdot)$ to map this q-dimensional latent vector into a d-dimensional simulated SNP array $\hat{\boldsymbol{x}}$ of kth ancestry. By passing ancestry labels directly to the decoder, we gain an explicit "handle" on which population's genetic templates we want to express (Supplemental Methods S3 and Supplemental Fig. S10).

In order to quantitatively assess the quality of the simulated individuals, we employ population genetics metrics such as LD patterns among SNPs (correlation structures, see Fig. 5) and folded allele frequency spectra (histogram of SNPs at 1%, 2%,…up to 50%; Supplemental Fig. S9), and we also leverage the entropy measure, as previously described in Geleta et al. (2025). In what follows, we differentiate between random variables, for example, $\boldsymbol{X}$, and samples, for example, $\boldsymbol{x}$, using uppercase and lowercase letters, respectively. In a genotype array denoted as $\boldsymbol{X} = [X_1, \ldots, X_d]$, each SNP X_i, where $1 \le i \le d$, is considered a random variable taking values in the Boolean domain $\mathbb{B} = \{0, 1\}$ with $P_{X_i}(x)$ representing the probability mass function for X_i. SNPs are typically modeled using a Bernoulli distribution. The entropy of X_i is defined as

$$H(X_i)|_{X_i\sim P_{X_i}} = -\sum_{x\in\mathbb{B}} P_{X_i}(x)\log_2 P_{X_i}(x). \tag{15}$$

We use the entropy to compute the mutual information for a pair of SNPs (X_i, X_j), which measures their mutual dependence and quantifies how much information one position provides about

Table 4. Accuracy of classification methods

		All		EUR		EAS		AMR		SAS		AFR		OCE		WAS	
Model	Criterion	TR	TS	TR	TS	TR	TS	TR	TS	TR	TS	TR	TS	TR	TS	TR	TS
PCA	$\arg\min_k \lVert\boldsymbol{z}_n - \boldsymbol{c}_k\rVert_2^2$	78.6	74.1	64.9	66.3	71.1	74.3	87.2	77.8	58.5	57.2	97.5	93.4	95.4	76.6	75.6	73.4
VAE		93.2	85.7	81.7	78.6	96.9	96.3	99.5	92.1	81.6	78.5	99.3	96.8	99.4	71.7	94.0	**86.1**
C-VAE	$\arg\min_k \lVert\boldsymbol{x}_n - \hat{\boldsymbol{x}}_n^{(k)}\rVert_1$	93.4	78.0	84.4	70.6	96.4	92.1	99.9	92.4	84.4	76.4	99.9	96.8	100	62.8	88.5	54.7
	$\arg\max_k p(Y=k\|\boldsymbol{x_n}, \boldsymbol{\theta})$	97.5	**87.1**	96.4	**87.1**	98.5	95.5	100	**97.4**	90.0	81.2	99.4	94.9	100	**79.8**	98.1	73.5
Y-VAE	$\arg\min_k \lVert\boldsymbol{x}_n - \hat{\boldsymbol{x}}_n^{(k)}\rVert_1$	98.9	83.2	96.9	81.5	99.6	96.2	99.9	87.2	98.5	**90.1**	100	**98.4**	100	68.6	97.5	59.9
	$\arg\max_k p(Y=k\|\boldsymbol{x_n}, \boldsymbol{\theta})$	99.1	85.2	97.6	84.3	99.7	**96.6**	100	90.9	98.2	88.5	100	98.2	100	72.7	98.3	65.0

TR refers to accuracy computed on training data and TS on test data, accordingly. The values represent the accuracy in %. Note that regular VAE, C-VAE, and Y-VAE have 10,371,760, 10,378,928, and 72,602,320 parameters, respectively. Bold values indicate the highest accuracy (best performance) on test samples across the compared models and criteria.

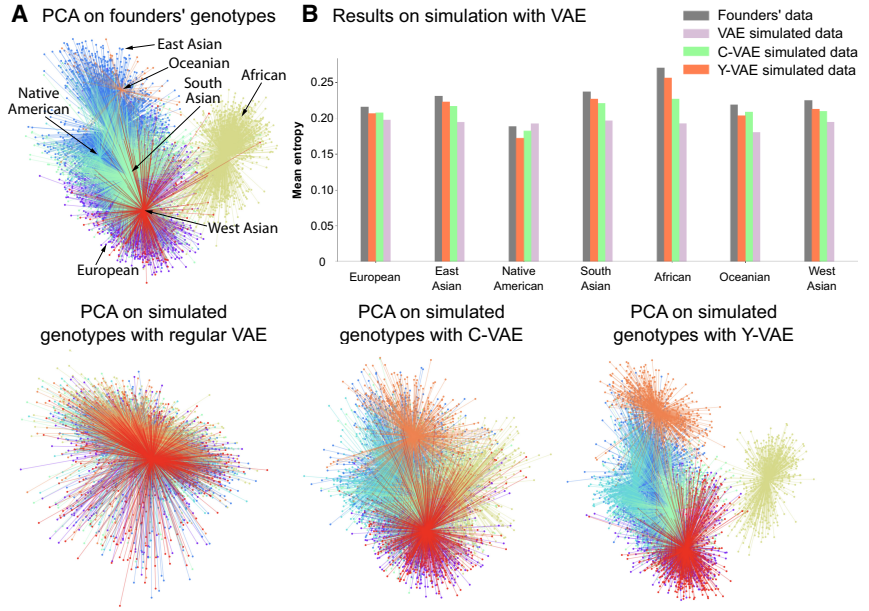

Figure 6. PCA of simulated genotypes and entropy comparison. The number of simulated samples is equal to the number of founders. (*A*) The *top-left* plots display the PCA projection of founders' genotypes. The plots at the *bottom* line display PCA projections of synthetic genotypes simulated with regular VAE, C-VAE, and Y-VAE, in that order. (*B*) The method that best approximates the entropy distribution is Y-VAE. The least effective method is without conditioning, because the entropy is approximately the same across all populations.

the other. Formally, it is defined as

$$I(X_i; X_j) = \sum_{x_i \in \mathbb{B}} \sum_{x_j \in \mathbb{B}} P_{(X_i,X_j)}(x_i, x_j) \log_2 \left(\frac{P_{(X_i,X_j)}(x_i, x_j)}{P_{X_i}(x_i) P_{X_j}(x_j)} \right). \quad (16)$$

Our analysis of pairwise mutual information (Fig. 5C) indicates that most pairs of SNPs exhibit negligible mutual information, which is consistent with the expectation that long-range LD is weak or absent due to recombination (Gravel 2012).

In Figure 5A and B, we contrast LD patterns between real and simulated samples, observing that the correlation between variants decays with distance. Notably, our VAE-generated SNP sequences capture locus-specific LD structure—evident in correlation peaks that align with those in real data. Because simulated samples of a specific ancestry should closely follow the distribution of their respective founders (the real genotype samples from our simulation pool), comparing the entropy of real and simulated SNPs provides a robust measure of divergence. Indeed, Figure 6B demonstrates that the best-performing simulation method (Y-VAE) produces an entropy distribution closely matching that of the founders, whereas the least effective method (unconditional VAE) results in an entropy distribution that is nearly uniform across populations. Additionally, Supplemental Results S2 and Supplemental Fig. S1 show that our synthetic genotypes support accurate downstream ancestry classification.

Finally, to contextualize VAE performance with another deep generative approach, we include a comparison to a generative moment matching network (GMMN), trained on the same subset (5000 SNPs from 10,000 samples) following Perera et al. (2022). Detailed ablations, computational constraints, and additional figures are provided in Supplemental Results S3 and Supplemental Fig. S2.

Discussion

We have demonstrated the power of VAEs applied to genomic variation analysis, providing promising performance in a variety of applications. We have conducted both qualitative and quantitative assessments of the quality of VAE clusters on human and canine SNP data sets, reinforcing the benefits of VAE for dimensionality reduction in a population genetics context. Most notably, we have developed a novel VAE-based lossless compression system tailored for SNP data and demonstrated how it can be integrated into existing SNP data set compression pipelines. We have also benchmarked VAE-based global ancestry classification against PCA-based classification and proved that the nonlinear approach performs better. Because of the introduced nonlinearities, the method is less sensitive to correlations of SNPs due to LD, resulting in an increased ability to capture complex population structure and represent relatively good ancestry differentiation and, unlike previous fully connected approaches (Battey et al. 2021), our window-based approach has shown to capture LD. In contrast to previous work for genotype simulation with VAEs (Battey et al. 2021), we have used VAE conditioning, which we found essential for generating high-quality simulated SNP sequences. VAE simulation provides an efficient method for SNP data simulation—we have conducted an entropy study on SNP data and reaffirmed the hypothesized migration paths (Nielsen et al. 2017) and phylogenetic relationships of the population groups, along with the theoretical compression bounds based on the statistical nature of SNP data.

In the context of our discussion, it is imperative to acknowledge several limitations associated with VAEs. First, when VAEs are employed for applications in data interpretability or visualization, the nonlinearity of the model may give rise to potentially misleading insights, akin to the caveats encountered in various dimensionality reduction techniques (Battey et al. 2021; Chari and Pachter 2023; Montserrat and Ioannidis 2023). Second, models trained specifically for a single task in a supervised fashion (e.g., classification) can outperform VAEs in terms of accuracy, a pattern that has been observed in other multimodal generalist models (Tu et al. 2024), and thus, the adoption of those task-specific specialist models may be a better option in scenarios where precise task execution is the primary objective. Finally, an additional limitation arises when considering the widespread adoption of VAEs for genomic compression. VAEs trained on a specific set of SNPs may exhibit challenges when applied to genomic data encompassing different genetic positions, and an adaption to new genetic positions could require retraining the network.

Code availability

The source code is publicly available at GitHub (https://github.com/AI-sandbox/aegen) and as Supplemental Code.

Competing interest statement

A.G.I. and D.M.M. own shares in Galatea Bio, Inc. The remaining authors declare no competing interests.

Acknowledgments

We thank A. Aw, J. Lin, and A. Bajwa for valuable feedback. In particular, we wish to acknowledge A. Bajwa for her assistance in proofreading the manuscript. Additionally, we are grateful to A.M. Martinez for the insightful conversations throughout the writing process.

Author contributions: M.G., A.G.I., and D.M.M. designed the research. M.G. performed the research and wrote the code base. M.G., X.G., A.G.I., and D.M.M. interpreted the results. M.G. and D.M.M. wrote the manuscript.

References

Absardi ZN, Javidan R. 2019. A fast reference-free genome compression using deep neural networks. In *2019 Big Data, Knowledge and Control Systems Engineering (BdKCSE)*, Sofia, Bulgaria, pp. 1–7. IEEE, Piscataway, NJ. doi:10.1109/BdKCSE48644.2019.9010661

Alexander D, Novembre J, Lange K. 2009. Fast model-based estimation of ancestry in unrelated individuals. *Genome Res* **19:** 1655–1664. doi:10.1101/gr.094052.109

Ali-Khan S, Daar A. 2010. Admixture mapping: From paradigms of race and ethnicity to population history. *Hugo J* **4:** 23–34. doi:10.1007/s11568-010-9145-y

Baran Y, Quintela I, Carracedo A, Pasaniuc B, Halperin E. 2013. Enhanced localization of genetic samples through linkage-disequilibrium correction. *Am J Hum Genet* **92:** 882–894. doi:10.1016/j.ajhg.2013.04.023

Battey C, Ralph P, Kern A. 2020. Predicting geographic location from genetic variation with deep neural networks. *eLife* **9:** e54507. doi:10.7554/eLife.54507

Battey CJ, Coffing GC, Kern AD. 2021. Visualizing population structure with variational autoencoders. *G3 (Bethesda)* **11:** jkaa036. doi:10.1093/g3journal/jkaa036

Brandon M, Wallace D, Baldi P. 2009. Data structures and compression algorithms for genomic sequence data. *Bioinformatics* **25:** 1731–1738. doi:10.1093/bioinformatics/btp319

Browning S, Waples R, Browning B. 2023. Fast, accurate local ancestry inference with flare. *Am J Hum Genet* **110:** 326–335. doi:10.1016/j.ajhg.2022.12.010

Browning B, Zhou Y, Browning S. 2018. A one-penny imputed genome from next-generation reference panels. *Am J Hum Genet* **103:** 338–348. doi:10.1016/j.ajhg.2018.07.015

Caliński T, Harabasz J. 1974. A dendrite method for cluster analysis. *Commun Stat* **3:** 1–27. doi:10.1080/03610927408827101

Chari T, Pachter L. 2023. The specious art of single-cell genomics. *PLoS Comput Biol* **19:** e1011288. doi:10.1371/journal.pcbi.1011288

Collet Y, Kucherawy M. 2018. Zstandard compression and the application/zstd media type. *RFC 8478*. doi:10.17487/RFC8878

Corbett-Detig R, Nielsen R. 2017. A hidden Markov model approach for simultaneously estimating local ancestry and admixture time using next generation sequence data in samples of arbitrary ploidy. *PLoS Genet* **13:** e1006529. doi:10.1371/journal.pgen.1006529

Davies D, Bouldin D. 1979. A cluster separation measure. *IEEE Trans Pattern Anal Mach Intell* **PAMI-1:** 224–227. doi:10.1109/TPAMI.1979.4766909

Dominguez Mantes A, Montserrat D, Bustamante C, Giró-i Nieto X, Ioannidis A. 2023. Neural admixture for rapid genomic clustering. *Nat Comput Sci* **3:** 621–629. doi:10.1038/s43588-023-00482-7

Elhaik E. 2022. Principal component analyses (PCA)-based findings in population genetic studies are highly biased and must be reevaluated. *Sci Rep* **12:** 14683. doi:10.1038/s41598-022-14395-4

Fujimura J, Rajagopalan R. 2011. Different differences: The use of "genetic ancestry" versus race in biomedical human genetic research. *Soc Stud Sci* **41:** 5–30. doi:10.1177/0306312710379170

Geleta M, Montserrat D, Ioannidis A. 2025. A Tsallis-entropy lens on genetic variation. arXiv:2511.03063 [cs.IT]. doi:10.48550/arXiv.2511.03063

Giancarlo R, Scaturro D, Utro F. 2009. Textual data compression in computational biology: A synopsis. *Bioinformatics* **25:** 1575–1586. doi:10.1093/bioinformatics/btp117

Glorot X, Bengio Y. 2010. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, Vol. 9, pp. 249–256. Proceedings of Machine Learning Research, Chia Laguna Resort, Sardinia.

Goyal M, Tatwawadi K, Chandak S, Ochoa I. 2018. DeepZip: Lossless data compression using recurrent neural networks. In *2019 Data Compression Conference (DCC)*, Snowbird, UT, pp. 575–575. IEEE, Piscataway, NJ. doi:10.1109/DCC.2019.00087

Gravel S. 2012. Population genetics models of local ancestry. *Genetics* **191:** 607–619. doi:10.1534/genetics.112.139808

Guglielmi G. 2019. Facing up to injustice in genome science. *Nature* **568:** 290–293. doi:10.1038/d41586-019-01166-x

Hernaez M, Pavlichin D, Weissman T, Ochoa I. 2019. Genomic data compression. *Annu Rev Biomed Data Sci* **2:** 19–37. doi:10.1146/biodatasci.2019.2.issue-1

Hinton G, Salakhutdinov R. 2006. Reducing the dimensionality of data with neural networks. *Science* **313:** 504–507. doi:10.1126/science.1127647

Ioffe S, Szegedy C. 2015. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *Proceedings of the 32nd International Conference on Machine Learning. PMLR* **37:** 448–456. https://proceedings.mlr.press/v37/ioffe15.html.

Kaufman L, Rousseeuw P. 2009. *Finding groups in data: An introduction to cluster analysis*. Wiley-Interscience, New York. doi:10.1002/9780470316801

Kingma D, Ba J. 2017. Adam: A method for stochastic optimization. arXiv:1412.6980 [cs.LG]. doi:10.48550/arXiv.1412.6980

Kingma DP, Welling M. 2014. Auto-encoding variational Bayes. In *Conference proceedings: papers accepted to the International Conference on Learning Representations (ICLR) 2014*. arXiv:1312.6114 [stat.ML]. doi:10.48550/arXiv.1312.6114

Lan D, Tobler R, Souilmi Y, Llamas B. 2021. Genozip: A universal extensible genomic data compressor. *Bioinformatics* **37:** 2225–2230. doi:10.1093/bioinformatics/btab102

Maher B. 2015. Genomics: Bioethics on stage. *Nature* **524:** 289–289. doi:10.1038/524289a

Mahoney MV. 2000. Fast text compression with neural networks. In *Proceedings of the Thirteenth International Florida Artificial Intelligence Research Society Conference (FLAIRS 2000)*, Orlando, FL (ed. Etheredge JN, Manaris BZ), pp. 230–234. The AAAI Press, Menlo Park, CA.

Mahoney M. 2005. *Adaptive weighing of context models for lossless data compression*. Technical Report CS-2005-16. Florida Institute of Technology, Melbourne, FL.

Maples B, Gravel S, Kenny E, Bustamante C. 2013. RFMix: A discriminative modeling approach for rapid and robust local-ancestry inference. *Am J Hum Genet* **93:** 278–288. doi:10.1016/j.ajhg.2013.06.020

Mardis E. 2011. A decade's perspective on DNA sequencing technology. *Nature* **470:** 198–203. doi:10.1038/nature09796

McInnes L, Healy J, Saul N, Großberger L. 2018. UMAP: Uniform Manifold Approximation and Projection. *J Open Source Softw* **3:** 861. doi:10.21105/joss.00861

Meisner J, Albrechtsen A. 2022. Haplotype and population structure inference using neural networks in whole-genome sequencing data. *Genome Res* **32:** 1542–1552. doi:10.1101/gr.276813.122

Montserrat DM, Bustamante C, Ioannidis A. 2019. Class-conditional VAE-GAN for local-ancestry simulation. In *14th Machine Learning in Computational Biology (MLCB 2019) meeting*, Vancouver, Canada. arXiv:1911.13220 [q-bio.GN]. doi:10.48550/arXiv.1911.13220

Montserrat D, Bustamante C, Ioannidis A. 2020. Lai-Net: Local-ancestry inference with neural networks. In *ICASSP 2020—2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 1314–1318. IEEE, Barcelona. doi:10.1109/ICASSP40776.2020.9053662

Montserrat D, Ioannidis A. 2023. Adversarial attacks on genotype sequences. In *ICASSP 2023—2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Rhodes Island, Greece, pp. 1–5. doi:10.1109/ICASSP49357.2023.10096857

Nalbantoglu A, Russell D, Sayood K. 2010. Data compression concepts and algorithms and their applications to bioinformatics. *Entropy* **12:** 34–52. doi:10.3390/e12010034

Nelson S, Yu J, Wagner J, Harrell T, Royal C, Bamshad M. 2018. A content analysis of the views of genetics professionals on race, ancestry, and genetics. *AJOB Empir Bioeth* **9:** 222–234. doi:10.1080/23294515.2018.1544177

Nielsen R, Akey J, Jakobsson M, Pritchard J, Tishkoff S, Willerslev E. 2017. Tracing the peopling of the world through genomics. *Nature* **541:** 302–310. doi:10.1038/nature21347

Novembre J, Johnson T, Bryc K, Kutalik Z, Boyko A, Auton A, Indap A, King K, Bergmann S, Nelson M, et al. 2008. Genes mirror geography within Europe. *Nature* **456:** 98–101. doi:10.1038/nature07331

Paszke A, Gross S, Massa F, Lerer A, Bradbury J, Chanan G, Killeen T, Lin Z, Gimelshein N, Antiga L, et al. 2019. PyTorch: An imperative style, high-performance deep learning library. In *33rd Conference on Neural Information Processing Systems (NeurIPS 2019)*, Vancouver, Canada. Neural Information Processing Systems Foundation, Inc. (NeurIPS). Curran Associates, Inc., Red Hook, NY.

Perera M, Montserrat D, Barrabes M, Geleta M, Giró-i Nieto X, Ioannidis A. 2022. Generative moment matching networks for genotype simulation. In *2022 44th Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC)*, Glasgow, UK, pp. 1379–1383. IEEE, Piscataway, NJ.

Popejoy A, Fullerton S. 2016. Genomics is failing on diversity. *Nature* **538:** 161–164. doi:10.1038/538161a

Pritchard J, Stephens M, Donnelly P. 2000. Inference of population structure using multilocus genotype data. *Genetics* **155:** 945–959. doi:10.1093/genetics/155.2.945

Rhead B, Haffener PE, Pouliot Y, De La Vega FM. 2023. Imputation of race and ethnicity categories using genetic ancestry from real-world genomic testing data. *Biocomputing 2024:* pp. 433–445. World Scientific. doi:10.1142/9789811286421_0033

Roberts D. 2011. *Fatal invention: How science, politics, and big business re-create race in the twenty-first century*. Faculty Scholarship at Penn Law, p. 433.

Romero A, Carrier P, Erraqabi A, Sylvain T, Auvolat A, Dejoie E, Legault M, Dubé M, Hussin J, Bengio Y. 2016. Diet networks: Thin parameters for fat genomics. In *International Conference on Learning Representations*. arXiv:1611.09340 [cs.LG]. doi:10.48550/arXiv.1611.09340

Sabat Oriol B, Montserrat D, Giro-i Nieto X, Ioannidis A. 2022. SALAI-Net: Species-agnostic local ancestry inference network. *Bioinformatics* **38** (Supplement_2)**:** ii27–ii33. doi:10.1093/bioinformatics/btac464

Schmidhuber J, Heil S. 1996. Sequential neural text compression. *IEEE Trans Neural Netw* **7:** 142–146. doi:10.1109/72.478398

Shah R, Gaedigk A. 2018. Precision medicine: Does ethnicity information complement genotype-based prescribing decisions? *Ther Adv Drug Saf* **9:** 45–62. doi:10.1177/2042098617743393

Sheena K, Nair M. 2024. GenCoder: a novel convolutional neural network based autoencoder for genomic sequence data compression. *IEEE/ACM Trans Comput Biol Bioinform* **21:** 405–415. doi:10.1109/TCBB.2024.3366240

Silva M, Pratas D, Pinho AJ. 2020. Efficient DNA sequence compression with neural networks. *GigaScience* **9:** giaa119. doi:10.1093/gigascience/giaa119

Srivastava N, Hinton G, Krizhevsky A, Sutskever I, Salakhutdinov R. 2014. Dropout: A simple way to prevent neural networks from overfitting. *J Mach Learn Res* **15:** 1929–1958.

Tan T, Atkinson E. 2023. Strategies for the genomic analysis of admixed populations. *Annu Rev Biomed Data Sci* **6:** 105–127. doi:10.1146/biodatasci.2023.6.issue-1

Tang H, Coram M, Wang P, Zhu X, Risch N. 2006. Reconstructing genetic ancestry blocks in admixed individuals. *Am J Hum Genet* **79:** 1–12. doi:10.1086/504302

Tipping M, Bishop C. 1999. Probabilistic principal component analysis. *J R Stat Soc Ser B* **61:** 611–622. doi:10.1111/1467-9868.00196

Tu T, Azizi S, Driess D, Schaekermann M, Amin M, Chang P, Carroll A, Natarajan V. 2024. Towards generalist biomedical AI. *New Engl J Med AI* **1**. doi:10.1056/AIoa2300138

van den Oord A, Vinyals O, Kavukcuoglu K. 2018. Neural Discrete Representation Learning. In *Advances in Neural Information Processing Systems 30 (NIPS 2017)*, Long Beach, CA (ed. Guyon I, et al.). Neural Information Processing Systems Foundation, Inc. (NeurIPS). Curran Associates, Inc., Red Hook, NY.

Van der Maaten L, Hinton G. 2008. Visualizing data using t-SNE. *J Mach Learn Res* **9:** 2579–2605.

Vincent P, Larochelle H, Lajoie I, Bengio Y, Manzagol P. 2010. Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. *J Mach Learn Res* **11:** 3371–3408.

Wang R, Bai Y, Chu Y, Wang Z, Wang Y, Sun M, Li J, Zang T, Wang Y. 2018. DeepDNA: a hybrid convolutional and recurrent neural network for compressing human mitochondrial genomes. In *2018 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, Madrid, pp. 270–274. IEEE, Piscataway, NJ. doi:10.1109/BIBM.2018.8621140

Wojcik G, Graff M, Nishimura K, Tao R, Haessler J, Gignoux C, Highland H, Patel Y, Sorokin E, Avery C, et al. 2019. Genetic analyses of diverse populations improves discovery for complex traits. *Nature* **570:** 514–518. doi:10.1038/s41586-019-1310-4

Xie H-G, Kim RB, Wood AJ, Stein CM. 2001. Molecular basis of ethnic differences in drug disposition and response. *Annu Rev Pharmacol Toxicol* **41:** 815–850. doi:10.1146/pharmtox.2001.41.issue-1

Yang H, Wang G, Wang M, Ma Y, Yin T, Fan R, Wu H, Zhong L, Irwin D, Zhai W, et al. 2017. The origin of chow chows in the light of the East Asian breeds. *BMC Genomics* **18:** 174. doi:10.1186/s12864-017-3525-9

Yelmen B, Decelle A, Ongaro L, Marnetto D, Tallec C, Montinaro F, Furtlehner C, Pagani L, Jay F. 2021. Creating artificial human genomes using generative neural networks. *PLoS Genet* **17:** e1009303. doi:10.1371/journal.pgen.1009303

Received October 1, 2024; accepted in revised form October 28, 2025.