G3 Genes | Genomes | Genetics

INVESTIGATIONS

A deep learning framework for characterization of genotype data

Kristiina Ausmees* **and Carl Nettelblad***,[1]

*Department of Information Technology, Uppsala University, 752 37, Uppsala, Sweden

ABSTRACT

Dimensionality reduction is a data transformation technique widely used in various fields of genomics research, with principal component analysis one of the most frequently employed methods. Application of principal component analysis to genotype data is known to capture genetic similarity between individuals, and is used for visualization of genetic variation, identification of population structure as well as ancestry mapping. However, the method is based on a linear model that is sensitive to characteristics of data such as correlation of single-nucleotide polymorphisms due to linkage disequilibrium, resulting in limitations in its ability to capture complex population structure.

Deep learning models are a type of nonlinear machine learning method in which the features used in data transformation are decided by the model in a data-driven manner, rather than by the researcher, and have been shown to present a promising alternative to traditional statistical methods for various applications in omics research. In this paper, we propose a deep learning model based on a convolutional autoencoder architecture for dimensionality reduction of genotype data.

Using a highly diverse cohort of human samples, we demonstrate that the model can identify population clusters and provide richer visual information in comparison to principal component analysis, and also yield a more accurate population classification model. We also discuss the use of the methodology for more general characterization of genotype data, showing that models of a similar architecture can be used as a genetic clustering method, comparing results to the ADMIXTURE software frequently used in population genetic studies.

KEYWORDS

deep learning, convolutional autoencoder, dimensionality reduction, genetic clustering, population genetics

INTRODUCTION

The increasing availability of large amounts of data has led to a rise in the use of machine learning (ML) methods in several fields of omics research. For many applications dealing with complex and heterogeneous information, the data-driven approach has become a promising alternative or complement to more traditional model-based methods (Xu and Jackson 2019; Libbrecht and Noble 2015; Schrider and Kern 2018).

Deep learning (DL) is an active subdiscipline of ML that has had a large impact in several fields, including image analysis and speech recognition (LeCun *et al.* 2015). DL methods comprise models that compute a nonlinear function of their input data using a layered structure that learns abstract feature representations in a hierarchical manner, and can be used for supervised learning tasks such as prediction and also in unsupervised settings for pattern recognition and data characterization problems (Goodfellow *et al.* 2016). A key aspect of DL is that the features used in data transformation are learned by the model as opposed to being defined by the researcher, resulting in a higher level of flexibility than alternative ML algorithms such as support vector machines (Zou *et al.* 2019).

Advances have been made in developing DL techniques for various types of omics data (Eraslan *et al.* 2019a). The current state-of-the-art for predicting effects of genetic variants on splicing is a DL model (Cheng *et al.* 2019). The DeepBind model (Alipanahi

Manuscript compiled: Tuesday 29th September, 2020

[1]Box 337 751 05 UPPSALA. carl.nettelblad@it.uu.se

et al. 2015) outperformed several previous non-DL approaches for predicting sequence specificities of DNA-binding proteins. For the task of variant calling of single-nucleotide polymorphisms (SNPs) and small indels, the DeepVariant model of Poplin *et al.* (2018) was shown to give improved results over existing tools (Nawy 2018).

DL has also been applied to unsupervised problems, including imputation of metabolite and SNP data (Scholz *et al.* 2005; Chen and Shi 2019; Sun and Kardia 2008), de-noising of ChIP–sequencing data (Koh *et al.* 2017) and outlier detection of RNA sequencing gene expression data (Brechtmann *et al.* 2018). In the field of single-cell RNA-sequencing, DL methods have been used for imputation, de-noising as well as dimensionality reduction (Talwar *et al.* 2018; Ding *et al.* 2018; Eraslan *et al.* 2019b).

Dimensionality reduction is a data transformation technique that is commonly applied to SNP data in the fields of population and quantitative genetics. Applications include visualization of genetic variation, detection of population structure and correcting for stratification in genome-wide association studies (GWAS) (Patterson *et al.* 2006; Price *et al.* 2006). One of the most widely used methods for performing dimensionality reduction is Principal Component Analysis (PCA), in which a linear transformation is made onto uncorrelated dimensions that maximize the variance of the projected data (Pearson 1901). It has been shown that the lower-dimensional representation resulting from PCA can capture patterns in genetic variation, e.g. by reconstructing geographical relationships from genotype data (Novembre *et al.* 2008).

Although PCA is an efficient and reliable method, there are limitations associated with it. Firstly, it can be sensitive to attributes of sequence data such as the presence of rare alleles and SNPs that are correlated due to linkage disequilibrium (LD), which can cause groupings of samples that reflect such phenomena rather than genome-wide population structure (Mathieson and McVean 2012; Tian *et al.* 2008). To avoid such spurious effects, a stringent filtering procedure to remove low frequency variation and SNPs in high LD is usually required prior to performing PCA, although methods to handle LD by e.g. shrinkage methods have been proposed (Zou *et al.* 2010).

Further limitations of PCA are related to the inability to capture nonlinear patterns in the data, as discussed in e.g. Mathieson and McVean (2012) and Alanis-Lobato *et al.* (2015). In the latter, the nonlinear method of non-centred Minimum Curvilinear Embedding (ncMCE) is proposed and shown to detect population structure in cases where PCA fails. Other nonlinear methods for dimensionality reduction that have been successfully applied to SNP data are t-distributed stochastic neighbor embedding (t-SNE), Uniform Manifold Approximation and Projection (UMAP) and generative topographic mapping (GTM) (Li *et al.* 2017; McInnes *et al.* 2018; Gaspar and Breen 2019). A DL approach consisting of a variational autoencoder is presented in Battey *et al.* (2020), where they show that their model can capture subtle features of population structure, while preserving global geometry to a higher degree than both t-SNE and UMAP.

In this paper, we present a DL framework denoted Genotype Convolutional Autoencoder (GCAE) for nonlinear dimensionality reduction of SNP data based on convolutional autoencoders. We describe adaptations to network architecture implemented to capture local as well as global patterns in sequence data, and compare dimensionality reduction performance to that of PCA on a highly diverse cohort of human samples. We also demonstrate the broader applicability of the framework for general characterization of SNP data by showing that minor modification of network structure can produce a model for solving the genetic clustering problem, and compare results to a model-based method commonly used in population genetic studies.

MATERIALS AND METHODS

Model architecture and training strategy

The proposed model for dimensionality reduction of SNP data is a convolutional autoencoder. Autoencoders are a class of DL models that transform data to a lower-dimensional latent representation from which it is subsequently reconstructed (Kramer 1991; Hinton and Salakhutdinov 2006). The idea is to learn features, or variables derived from the original data, that capture the important characteristics of the data. The structure of the model is shown in Figure 2. It comprises four types of layers that transform the input in a sequential manner: convolutional, pooling, fully-connected and upsampling layers.

Convolutional layers consist of a number of weight matrices, or kernels, of a specified size that are used to compute a sliding dot product of the input. Each kernel is convolved over the input layer along its spatial dimension with a given stride, or step size. In our models we only use a stride of size 1. The depth of the layer output is thus determined by the number of kernels, and the spatial dimension of the data is unchanged. The pooling layers perform downsampling by applying a max filter over sliding windows of the data, separately for each depth dimension, reducing the size of the spatial dimension and leaving the depth unchanged.

The encoder alternates convolutional and pooling layers to increase the depth and reduce the spatial dimension of the data. The center of the model consists of a series of fully-connected layers in which the latent representation, or encoding, is defined. In contrast to convolutional layers, fully-connected layers contain weights between all pairs of variables in the input and output. The decoder roughly mirrors the structure of the encoder, with upsampling performed by means of nearest-neighbor interpolation to increase the spatial dimension of the data.

Residual connections, shown as black dashed lines in Figure 2, are used to stabilize the training process. These add the output from one layer to later parts of the network, skipping over layers in between, and have been shown to facilitate the optimization of deep networks for different applications (He *et al.* 2015).

Convolutional layers allow the model to capture local patterns and make use of the sequential nature of genetic data, allowing it to incorporate essential features such as LD at various length sclaes.

In order to facilitate the learning of global patterns in the input data, the model has two additional sets of variables. Each of these sets contains one variable per marker that is updated during the optimization process, allowing the model to capture marker-specific behavior. The two sets of marker-specific variables, illustrated in Figure 2 in red and green, are both inserted into the model by concatenation to layers in the decoder. One set of variables is also concatenated to the model input at every stage of the training process.

The activation function exponential linear unit (elu) is applied to convolutional layers, after which batch normalization is performed. The fully-connected layers also have elu applied to them, except the outermost ones which have linear activation. The final convolution is performed with a kernel size of 1, with linear activation and no batch-normalization. In order to regularize the network and avoid overfitting, dropout is used on the weights of the fully-connected layers, except those surrounding the latent representation, and Gaussian noise is added to the latent representation during training.

Input data is represented as an $(n_{samples} \times n_{markers})$ matrix of diploid genotypes, normalized to the range $[0, 1]$ by mapping $0 \rightarrow 0.0$, $1 \rightarrow 0.5$, $2 \rightarrow 1.0$. When calculating the loss function, target genotypes are represented using one-hot encoding with 3 classes $[p(0), p(1), p(2)]$ with $p(g) = 1.0$ for the true genotype, and $p(g) = 0.0$ for the others. Model output o_{ij} of sample i at site j has the softmax function applied to it, after which it is interpreted as the allele frequency at the site. This scalar is transformed to the format of the targets using the formula for genotype frequencies $f(g)$ obtained from the principle of Hardy-Weinberg equilibrium: $[f(0) = (1 - o_{ij})^2, f(1) = 2(1 - o_{ij})o_{ij}, f(2) = o_{ij}^2]$.

The network is trained to reduce the categorical cross-entropy error $E(y, \hat{y}) = \sum_i^3 y_i log(\hat{y}_i)$ between target (y) and reconstructed $(\hat{y})$ genotypes, with an added L2 penalty on the values of the latent representation for regularization. Network optimization is performed by means of the ADAM algorithm (Kingma and Ba 2014), with a further exponential decay of the learning rate applied.

Additional regularization of the training process is performed by means of data augmentation. For every sample in the training process, a fraction of genotypes is randomly set to missing in the input data, represented by the value -1.0. A dimension representing missing and non-missing genotypes with the values 0 and 1, respectively, is added to the input data, depicted in light blue in Figure 2. The fraction of missingness for each training batch is randomly selected from a pre-defined range.

The data was randomly split into training and validation sets consisting of 80% and 20% of samples, stratified by population. The training set was used for network optimization, and the epoch at which the model's error on the validation set was the lowest was used for determining when to terminate training.

Different model architecture settings and hyperparameters were evaluated, and the best-performing setups were chosen by means of a hierarchical search procedure. We refer to Supplemental File S1 for details about the evaluated options, including the final model architecture settings and hyperparameter values used for obtaining the presented results.

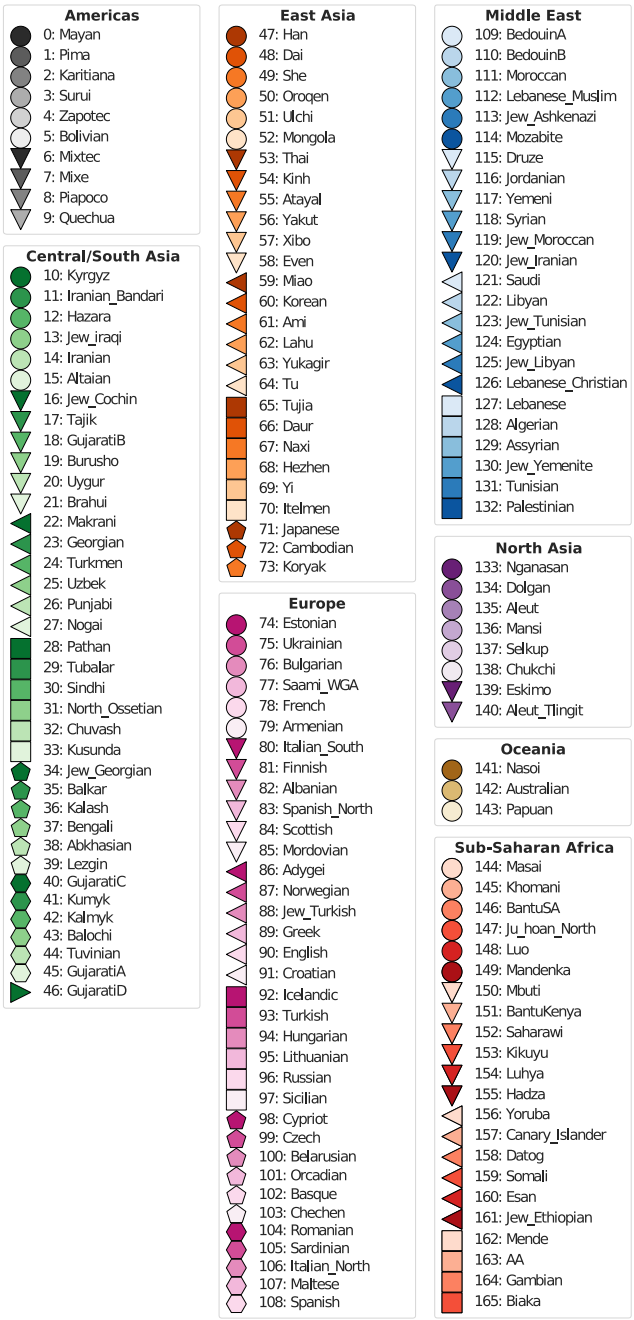

Figure 1 Populations and superpopulations of the Human Origins panel of genotype data. The coloring and numbering serve as a legend for Figures 3 - 5.

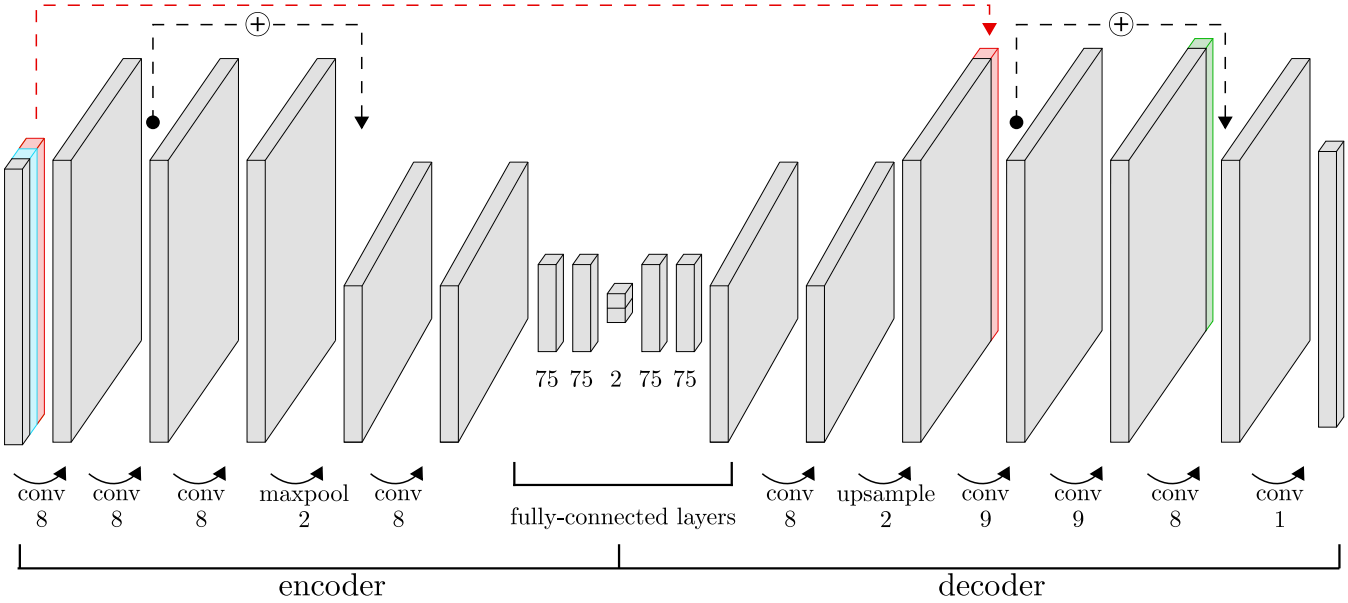

Figure 2 Architecture of the GCAE model used for dimensionality reduction. The encoder transforms data to a lower-dimensional latent space through a series of convolutional, pooling and fully-connected layers. The decoder reconstructs the input genotypes. The input consists of three layers: genotype data (gray), a binary mask representing missing data (blue), and a marker-specific trainable variable per SNP (red). The red dashed line indicates where this marker-specific variable is concatenated to a layer in the decoder. Another marker-specific trainable variable, shown in green, is also concatenated to the second-last layer in the decoder. Black dsahed lines indicate residual connections, where the output of a layer is added to that of another layer later in the network. The numbers below the layers indicate the number of kernels for convolutional layers, down- or upsampling factor for pooling and upsampling layers, and number of units for fully-connected layers. The displayed numbers are those of the final model used to obtain the presented results for dimensionality reduction. For the genetic clustering application, the number of units in the middle layer was $k = 5$.

Data

The data used in this study was derived from the fully public present-day individuals of the Affymetrix Human Origins SNP array analyzed in Lazaridis *et al.* (2016). This data set is designed for population genetic studies and represents worldwide genetic variation, containing 2,068 samples from 166 populations. These were categorized into 8 superpopulations, displayed in Figure 1 which also serves as a legend for the plots in the Results section.

The data was filtered to exclude sex chromosomes and non-informative sites, and one sample (NA13619) was removed due to relation to another (HGDP01382). In order to obtain a single data set for fair comparison between methods, the genotypes were further filtered according to the procedure that is common to perform prior to applying PCA on SNP data. A minor allele frequency (MAF) threshold of 1% was enforced, and LD pruning was performed by removing one of each pair of SNPs in windows of 1 centimorgan that had an allelic r^2 value greater than 0.2.

As the comparison of robustness of different methods to missing data was beyond the scope of this study, missing genotypes were set to the most frequent value per SNP so as to avoid their influence over dimensionality reduction results. The final data set consisted of 2,067 individuals typed at 160,858 biallelic sites.

Evaluation of dimensionality reduction performance

Comparison of performance between the autoencoder and PCA was performed by means of evaluating the ability of the dimensionality reduction to capture population structure. A k-Nearest Neighbors (k-NN) classification model was defined based on the projected data by assigning a population label to each sample based on the most frequent label among its k nearest neighbors. Classification performance was measured by the F1 score, defined per population label c as the harmonic mean of the precision and recall: $F1_c = \frac{2*precision*recall}{precision*recall}$. The total F1 score for a model was defined as the average population F1 score, weighted by the number of samples per population.

PCA was performed using the same normalization method as in the software SMARTPCA Patterson *et al.* (2006), i.e. by subtraction of the mean and division with an estimate of the standard deviation of the population allele frequency per SNP.

The entire data set was used for performance evaluation for PCA and the autoencoder. As the performance metrics are based on the population labels of the data which are not used by either model, a separate test set of samples unseen to the models during training was not used for performance evaluation.

Extension to genetic clustering

The problem of genetic clustering refers to the characterization of individual genomes by proportional assignment to a set of clusters, or genetic components. These may be used in the analysis of population structure, in identifying patterns of genetic variation between populations and for assigning individuals to populations.

The DL model for genetic clustering was developed by making minor changes to the autoencoder architecture used for dimensionality reduction described above. The number of units in the encoding layer was changed from 2 to k, the number of clusters. In order to obtain a proportional assignment, softmax normalization was further applied to the encoding to obtain a vector of k values that sum to 1. Supplemental File S1 contains more information about the network architecture settings and training options used for the genetic clustering model.

We consider the widely used software ADMIXTURE Alexander *et al.* (2009) as a comparison method for the genetic clustering application, and present results in a similar manner using bar graphs displaying the proportional assignment of clusters for each sample.

Implementation

GCAE is implemented in Python 3 using Tensorflow 2.0 Abadi *et al.* (2015) and is available at https://github.com/kausmees/GenoCAE as a command-line program. The software plink 1.9 (Shaun Purcell, Christopher Chang 2020; Chang *et al.* 2015) was used for filtering of genotype data. All other preprocessing of data, as well as performance evaluation and visualization, were implemented in Python. The library scikit-learn Pedregosa *et al.* (2011) was used for calculating the F1 score metric.

PCA was performed using the implementation in scikit-learn, and the reference results for genetic clustering were obtained using the software ADMIXTURE 1.3.0 using the em method.

All computations were performed on the resources of Uppsala Multidisciplinary Center for Advanced Computational Science (UPPMAX) on a cluster of compute servers equipped with 128 GB memory, each comprising two 8-core Xeon E5-2660 processors running at 2.2 GHz. Computations were run under the projects SNIC 2019/8-38 and SNIC 2020/5-91, provided by the Swedish National Infrastructure for Computing (SNIC).

Data Availability

The fully public Affymetrix Human Origins present-day individuals from Lazaridis *et al.* (2016) are available for download from https://reich.hms.harvard.edu/datasets.

RESULTS

Figure 3 shows dimensionality reduction results using PCA and GCAE. Both methods are able to capture global geometry to a high degree, with a visible clustering according to superpopulation. Both methods also result in a consistent global pattern with the African superpopulation separating distinctly, a gradient from the Middle East to Europe and Asia that roughly reflects the geography of Eurasia, and a separate Oceanian cluster.

On a local scale, however, PCA shows a significant overlap within and between superpopulations, visible for example at the top of the PC2 axis where North Asian populations overlap with the East Asian and American clusters without clear differentiation.

The latent space of GCAE displays a higher level of resolution, distinguishing individual populations and reflecting global geographical patterns to a larger extent. Within the African superpopulation, samples from the north east of the continent are in closer proximity to the Middle Eastern cluster, and differentiated from the large cluster of west-African populations. The San populations Khomani and Ju Hoan form a distinct cluster, which is not evident in the PCA plot. A similar observation holds for the the Mbuti and Biaka groups of the Congo Basin area. An east-west gradient is also visible within Asian populations, with e.g. the Turkic Nogai closer to Europe, and the Yukagir, Koryak, Chukchi and Eskimo populations of Far East Siberia closer to the Americas, which in turn appears as a clearly distinguished cluster.

Table 1 shows the classification performance of k-NN models based on the dimensionality reduction results of PCA and GCAE using 2 dimensions. For comparison, we also include results for PCA using 4 dimensions. Scores are calculated per population, and shown averaged over superpopulations as well as for the total data set, for $k = 3, 5, 8$. Overall, the 3-NN model gave the highest F1 scores for both methods. For $k = 3$, PCA using 2 dimensions resulted in an F1 score of 0.657 on the entire data set,

which increased to 0.758 when considering 4 dimensions. For GCAE, the score was 0.750 using 2 dimensions. F1 scores for all populations and models are available in Supplemental Table S1.

Figures 4 and 5 show the genetic clustering results with 5 clusters using ADMIXTURE and GCAE, with the order of the clusters adjusted to be analogous for comparison. Both models result in distinct American, Oceanian and African clusters, with the latter also showing a very similar pattern of the blue component. The ADMIXTURE results further show the East Asian and European superpopulations as largely distinct clusters, whereas GCAE reveals these as composite.

For European populations, the most prominent components in the GCAE clustering are blue, which is mainly present in the Middle East for both methods, and red. A possible interpretation is that the red cluster signifies the genetic component of herders that migrated to Europe from the Pontic–Caspian steppe around 4.5 kyr ago (Nielsen *et al.* 2017; Haak *et al.* 2015). This would be consistent with a presence in most European populations, with the exception of Sardinians (Lazaridis *et al.* 2014), as well as in South Asia, with particular prominence in e.g. the Kalash (Lamnidis *et al.* 2018; Pathak *et al.* 2018). This component also appears in East Asian populations, which also include American and Oceanian ancestry. Compared to ADMIXTURE, the GCAE results display higher resolution of European as well as East and South/Central Asian populations overall, resulting in a more nuanced depiction of Eurasian genetic variation.

DISCUSSION

One approach to evaluation of dimensionality reduction results involves assessing the correspondence between transformed data and its geographical sampling location. Quantitative studies have shown that geographical effects in the form of migration and the impact of physical distance on gene flow play a role in creating population structure (Wang *et al.* 2012). Striking similarities between PCA and geography have mainly been reported from limited geographical areas such as Europe (Novembre *et al.* 2008; Lao *et al.* 2008), with world-wide cohorts generally resulting in a less resolved V-shape similar to that shown in Figure 3 A (Jakobsson *et al.* 2008; Biswas *et al.* 2009). Our results demonstrate that GCAE provides a projection that is visually more informative than that of PCA on this highly diverse cohort of samples, with higher resolution of population structure and global patterns of genetic variation.

The interpretation of dimensionality reduction results, particularly the inference of the underlying processes behind observed structure, is however not always straightforward. In Novembre and Stephens (2008) and François *et al.* (2010), the effects of past migration and expansion events on PCA is discussed, and how the assumptions of linearity and orthogonality of the model can result in counter-intuitive patterns in PC-space. The mathematical properties of PCA and its resulting sensitivity to attributes of the data such as LD and so-called "informative missingness", due to e.g. different sequencing panels or the higher uncertainty associated with heterozygote calls, are also extensively discussed in Patterson *et al.* (2006).

As GCAE is a nonlinear model, it has the potential to capture patterns in the input that PCA cannot, allowing for more complex characteristics of the data to inform the inference population structure. For the population classification model, this resulted in an F1 score for GCAE with 2 dimensions that was closer to that of PCA using twice as many dimensions.

We also note that in order to obtain a fair comparison, we have performed filtering of the SNP set in terms of MAF and LD according to standard protocols for PCA even though these steps are not necessarily required for GCAE, in which such patterns can be learned during the training process. The GCAE architecture also includes a representation of missing genotypes, unlike PCA. In practice, missing data for PCA, is often handled by either imputation with the empirical mean and/or filtering to remove sites with high missingness. In this sense, GCAE can present a more robust alternative to PCA that is more suitable for low-coverage samples such as ancient DNA, requiring less filtering and allowing more of the data to be retained for analysis.

Other nonlinear methods such as GTM, t-SNE, UMAP and the variational autoencoder popvae presented in Battey *et al.* (2020) have been shown to alleviate many of the above discussed issues with PCA, and give improved dimensionality reduction results on genotype data. The main difference between these and GCAE is that they, like PCA, do not take the sequential nature of genotype data into account, but rather treat every SNP as an independent variable. Using the convolution operation allows GCAE to learn spatial hierarchies, capturing different scales of patterns in the data such as those resulting from varying local ancestry and LD structures.

Our results demonstrate that the use of convolution to capture sequential information is feasible for genotype data, in spite of its fundamental differences to images to which such networks are typically applied. Genotype data is position-dependent, with a unique meaning to every dimension. Pixels in images, in contrast, typically represent information that is translation invariant. We found that the use of marker-specific variables improved performance, and suggest this as a means of allowing the model to represent some of the positional information that is lost with convolution.

Many commonly used methods for genetic clustering such as TreeMix Pickrell and Pritchard (2012), ADMIXTUREGRAPH Leppälä *et al.* (2017) and GLOBETROTTER Hellenthal *et al.* (2014) are, unlike the dimensionality reduction methods discussed, model-based. STRUCTURE (Pritchard *et al.* 2000), for example, represents LD and includes explicit modeling of admixture blocks and the transitions between them. The model assumes the existence of a set of differentiated ancestral populations, and that the sample is a result of relatively recent mixing of these. ADMIXTURE, which we use as a reference method in this work, is based on the same underlying statistical model.

GCAE, in contrast, constitutes a more flexible and data-driven approach to the genetic clustering problem, without any explicit modeling assumptions. Our results demonstrate that GCAE is able to capture very similar population structure as that found by ADMIXTURE, but that it also identifies additional characteristics for some populations that are consistent with existing findings in the literature.

As previously discussed regarding dimensionality reduction, interpretation and evaluation of genetic clustering is not straightforward. The correctness of an assignment is not well-defined, and different underlying processes can give rise to similar observed patterns (Lawson *et al.* 2018). Evaluation of results requires additional information, such as putting them into the context of other methods that are based on different models, and the analysis of metrics like F and D statistics.

A purely data-driven black-box approach such as DL can be more difficult to interpret. The features used in the data transformation are unknown and therefore cannot be used for validation of whether certain modeling assumptions hold for the data in question. On the other hand, the alternative methodology of GCAE can

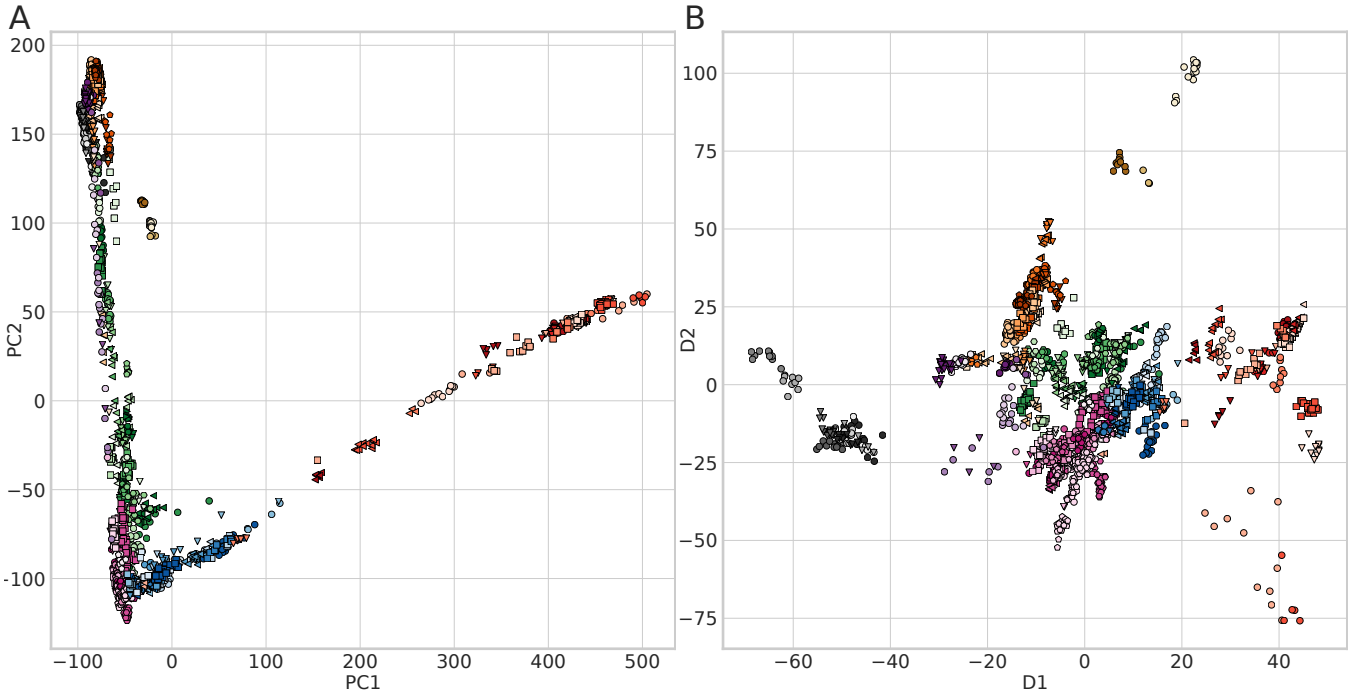

Figure 3 Dimensionality reduction results for PCA (A) and GCAE (B) on the Human Origins panel. Legend in Figure 1.

Table 1 F1 scores of k-NN population classification models based on the dimensionality reduction results of PCA and GCAE for $k = 3, 5, 8$ using different number of dimensions. Scores are calculated per population class, and averaged over superpopulations by weighting with the number of samples per population. Total scores show the scores averaged over all populations.

Model	k	Americas	Central/ South Asia	East Asia	Europe	Middle East	North Asia	Oceania	Sub-Saharan Africa	Total
PCA 2 dim	3	0.558	0.748	0.594	0.621	0.605	0.632	1.000	0.738	0.657
	5	0.528	0.675	0.509	0.553	0.536	0.543	1.000	0.703	0.591
	8	0.479	0.607	0.433	0.486	0.499	0.459	0.839	0.653	0.528
PCA 4 dim	3	0.734	0.785	0.761	0.705	0.671	0.848	1.000	0.884	0.758
	5	0.615	0.747	0.712	0.660	0.618	0.824	1.000	0.825	0.709
	8	0.560	0.704	0.656	0.652	0.568	0.818	0.839	0.811	0.675
GCAE 2 dim	3	0.743	0.739	0.721	0.760	0.668	0.864	0.960	0.829	0.750
	5	0.677	0.701	0.674	0.690	0.575	0.829	1.000	0.815	0.697
	8	0.653	0.647	0.628	0.656	0.511	0.765	0.841	0.744	0.646

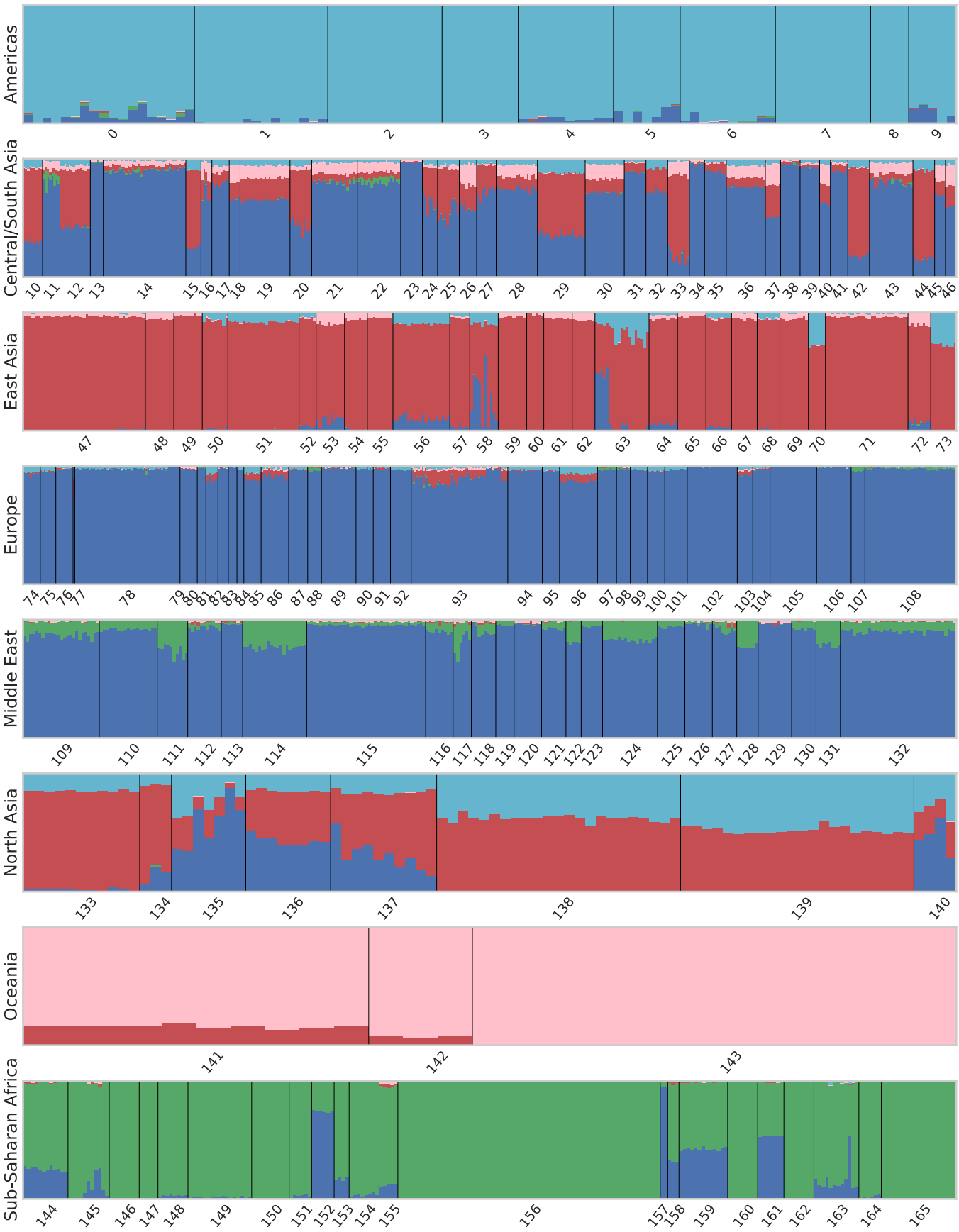

Figure 4 Genetic clustering results with $k = 5$ clusters using ADMIXTURE. Each bar represents a sample from the Human Origins panel, with colors indicating the proportional assignment into k clusters for that sample. Samples are ordered by population and superpopulation, with numbering according to the legend in Figure 1.

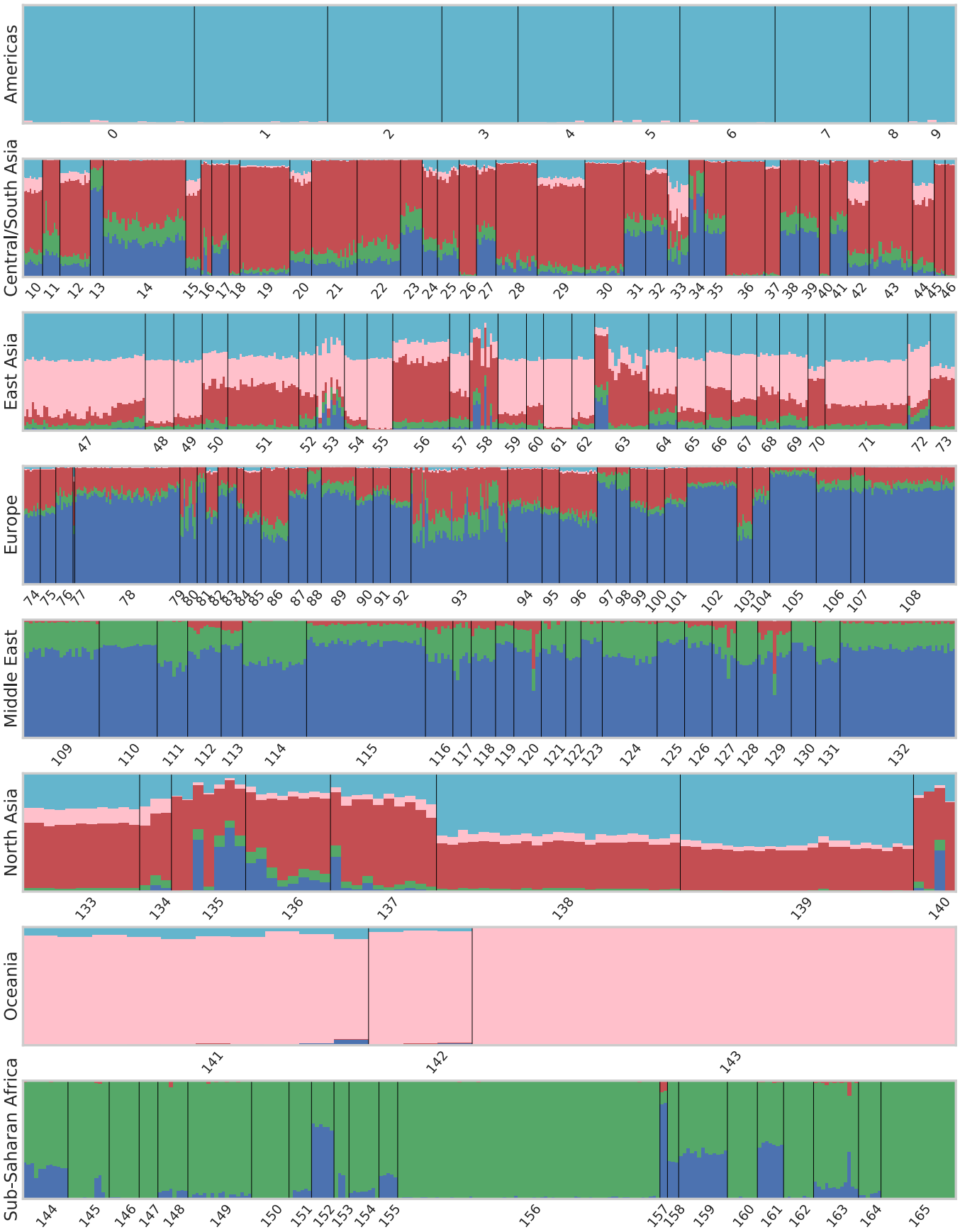

Figure 5 Genetic clustering results with $k = 5$ clusters using GCAE. Each bar represents a sample from the Human Origins panel, with colors indicating the proportional assignment into k clusters for that sample. Samples are ordered by population and superpopulation, with numbering according to the legend in Figure 1.

allow it to capture additional aspects of the data, and provide a useful complement to the toolset used for exploratory data analysis in population genetics.

Another characteristic of convolutional layers is that they require less trainable variables than a corresponding fully-connected layer, leading to reduced computational requirements for training. Depending on overall network architecture and training strategy, this may allow for the design of models that are more feasible to train on large data sets.

The dimensionality reduction and genetic clustering models took 26.9 and 46.6 hours, respectively, to train using 11 cores on UPPMAX. The computational requirements of GCAE are thus greater than that of PCA and ADMIXTURE. As the purpose of this study is to evaluate the applicability of convolutional autoencoders to the chosen problems, optimization of computational efficiency is considered out of scope and left for future work. We also note that, due to practical concerns, we opted to only run on CPUs, and that GPUs could be used to improve computational performance significantly in the future.

Our results demonstrate that GCAE can learn features that characterize genotype data in a meaningful way. The minor model changes required to change the application from dimensionality reduction to genetic clustering further demonstrate the flexibility of the method, and future efforts will involve investigating the application of GCAE to other problems. A simple alternative application would be imputation of missing genotypes. As the training procedure is based on reconstructing the input, and since we already include a representation of missing data, this would mainly involve finding a suitable number of units to use in the latent layer. We are also currently exploring the use of GCAE in the context of quantitative genetics by incorporating phenotypic information into the model.

LITERATURE CITED

Abadi, M., A. Agarwal, P. Barham, E. Brevdo, Z. Chen, *et al.*, 2015 TensorFlow: Large-scale machine learning on heterogeneous systems. Software available from tensorflow.org.

Alanis-Lobato, G., C. V. Cannistraci, A. Eriksson, A. Manica, and T. Ravasi, 2015 Highlighting nonlinear patterns in population genetics datasets. Scientific Reports **5**: 8140.

Alexander, D., J. Novembre, and K. Lange, 2009 Fast model-based estimation of ancestry in unrelated individuals. Genome Research **19**: 1655–1664.

Alipanahi, B., A. Delong, M. T. Weirauch, and B. J. Frey, 2015 Predicting the sequence specificities of dna- and rna-binding proteins by deep learning. Nature Biotechnology **33**: 831–838.

Battey, C. J., G. C. Coffing, and A. D. Kern, 2020 Visualizing population structure with variational autoencoders. bioRxiv .

Biswas, S., L. B. Scheinfeldt, and J. M. Akey, 2009 Genome-wide insights into the patterns and determinants of fine-scale population structure in humans. The American Journal of Human Genetics **84**: 641 – 650.

Brechtmann, F., C. Mertes, A. Matusevičiūtė, V. A. Yépez, Žiga Avsec, *et al.*, 2018 Outrider: A statistical method for detecting aberrantly expressed genes in rna sequencing data. The American Journal of Human Genetics **103**: 907 – 917.

Chang, C. C., C. C. Chow, L. C. Tellier, S. Vattikuti, S. M. Purcell, *et al.*, 2015 Second-generation PLINK: rising to the challenge of larger and richer datasets. GigaScience **4**, s13742-015-0047-8.

Chen, J. and X. Shi, 2019 Sparse convolutional denoising autoencoders for genotype imputation. Genes **10**: 652, 31466333[pmid].

Cheng, J., T. Y. D. Nguyen, K. J. Cygan, M. H. Çelik, W. G. Fairbrother, *et al.*, 2019 Mmsplice: modular modeling improves the predictions of genetic variant effects on splicing. Genome Biology **20**: 48.

Ding, J., A. Condon, and S. P. Shah, 2018 Interpretable dimensionality reduction of single cell transcriptome data with deep generative models. Nature communications **9**: 2002–2002, 29784946[pmid].

Eraslan, G., Z. Avsec, J. Gagneur, and F. J. Theis, 2019a Deep learning: new computational modelling techniques for genomics. Nature Reviews Genetics **20**: 389–403.

Eraslan, G., L. M. Simon, M. Mircea, N. S. Mueller, and F. J. Theis, 2019b Single-cell rna-seq denoising using a deep count autoencoder. Nature Communications **10**: 390.

François, O., M. Currat, N. Ray, E. Han, L. Excoffier, *et al.*, 2010 Principal Component Analysis under Population Genetic Models of Range Expansion and Admixture. Molecular Biology and Evolution **27**: 1257–1268.

Gaspar, H. A. and G. Breen, 2019 Probabilistic ancestry maps: a method to assess and visualize population substructures in genetics. BMC Bioinformatics **20**: 116.

Goodfellow, I., Y. Bengio, and A. Courville, 2016 *Deep Learning*. MIT Press, http://www.deeplearningbook.org.

Haak, W., I. Lazaridis, N. Patterson, N. Rohland, S. Mallick, *et al.*, 2015 Massive migration from the steppe was a source for indo-european languages in europe. Nature **522**: 207–211.

He, K., X. Zhang, S. Ren, and J. Sun, 2015 Deep residual learning for image recognition. CoRR **abs/1512.03385**.

Hellenthal, G., G. B. J. Busby, G. Band, J. F. Wilson, C. Capelli, *et al.*, 2014 A genetic atlas of human admixture history. Science **343**: 747–751.

Hinton, G. E. and R. R. Salakhutdinov, 2006 Reducing the dimensionality of data with neural networks. Science **313**: 504–507.

Jakobsson, M., S. W. Scholz, P. Scheet, J. R. Gibbs, J. M. VanLiere, *et al.*, 2008 Genotype, haplotype and copy-number variation in worldwide human populations. Nature **451**: 998–1003.

Kingma, D. P. and J. Ba, 2014 Adam: A method for stochastic optimization.

Koh, P. W., E. Pierson, and A. Kundaje, 2017 Denoising genome-wide histone ChIP-seq with convolutional neural networks. Bioinformatics **33**: i225–i233.

Kramer, M. A., 1991 Nonlinear principal component analysis using autoassociative neural networks. AIChE Journal **37**: 233–243.

Lamnidis, T. C., K. Majander, C. Jeong, E. Salmela, A. Wessman, *et al.*, 2018 Ancient fennoscandian genomes reveal origin and spread of siberian ancestry in europe. Nature Communications **9**: 5018.

Lao, O., T. T. Lu, M. Nothnagel, O. Junge, S. Freitag-Wolf, *et al.*, 2008 Correlation between genetic and geographic structure in europe. Current Biology **18**: 1241 – 1248.

Lawson, D. J., L. van Dorp, and D. Falush, 2018 A tutorial on how not to over-interpret structure and admixture bar plots. Nature Communications **9**: 3258.

Lazaridis, I., D. Nadel, G. Rollefson, D. C. Merrett, N. Rohland, *et al.*, 2016 Genomic insights into the origin of farming in the ancient near east. Nature **536**: 419–424.

Lazaridis, I., N. Patterson, A. Mittnik, G. Renaud, S. Mallick, *et al.*, 2014 Ancient human genomes suggest three ancestral populations for present-day europeans. Nature **513**: 409–413.

LeCun, Y., Y. Bengio, and G. Hinton, 2015 Deep learning. Nature **521**: 436–444.

Leppälä, K., S. V. Nielsen, and T. Mailund, 2017 admixturegraph:

an R package for admixture graph manipulation and fitting. Bioinformatics **33**: 1738–1740.

Li, W., J. E. Cerise, Y. Yang, and H. Han, 2017 Application of t-sne to human genetic data. Journal of Bioinformatics and Computational Biology **15**: 1750017, PMID: 28718343.

Libbrecht, M. W. and W. S. Noble, 2015 Machine learning applications in genetics and genomics. Nature Reviews Genetics **16**: 321–332.

Mathieson, I. and G. McVean, 2012 Differential confounding of rare and common variants in spatially structured populations. Nature genetics **44**: 243–246.

McInnes, L., J. Healy, and J. Melville, 2018 Umap: Uniform manifold approximation and projection for dimension reduction.

Nawy, T., 2018 Variants from the deep. Nature Methods **15**: 861–861.

Nielsen, R., J. M. Akey, M. Jakobsson, J. K. Pritchard, S. Tishkoff, *et al.*, 2017 Tracing the peopling of the world through genomics. Nature **541**: 302–310.

Novembre, J., T. Johnson, K. Bryc, Z. Kutalik, A. R. Boyko, *et al.*, 2008 Genes mirror geography within europe. Nature **456**: 98–101.

Novembre, J. and M. Stephens, 2008 Interpreting principal component analyses of spatial population genetic variation. Nature Genetics **40**: 646–649.

Pathak, A. K., A. Kadian, A. Kushniarevich, F. Montinaro, M. Mondal, *et al.*, 2018 The genetic ancestry of modern indus valley populations from northwest india. The American Journal of Human Genetics **103**: 918 – 929.

Patterson, N., A. L. Price, and D. Reich, 2006 Population structure and eigenanalysis. PLOS Genetics **2**: 1–20.

Pearson, K., 1901 On lines and planes of closest fit to systems of points in space. The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science **2**: 559–572.

Pedregosa, F., G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, *et al.*, 2011 Scikit-learn: Machine learning in Python. Journal of Machine Learning Research **12**: 2825–2830.

Pickrell, J. K. and J. K. Pritchard, 2012 Inference of population splits and mixtures from genome-wide allele frequency data. PLOS Genetics **8**: 1–17.

Poplin, R., P.-C. Chang, D. Alexander, S. Schwartz, T. Colthurst, *et al.*, 2018 A universal snp and small-indel variant caller using deep neural networks. Nature Biotechnology **36**: 983–987.

Price, A. L., N. J. Patterson, R. M. Plenge, M. E. Weinblatt, N. A. Shadick, *et al.*, 2006 Principal components analysis corrects for stratification in genome-wide association studies. Nature Genetics **38**: 904–909.

Pritchard, J. K., M. Stephens, and P. Donnelly, 2000 Inference of population structure using multilocus genotype data. Genetics **155**: 945–959, 10835412[pmid].

Scholz, M., F. Kaplan, C. L. Guy, J. Kopka, and J. Selbig, 2005 Non-linear PCA: a missing data approach. Bioinformatics **21**: 3887–3895.

Schrider, D. R. and A. D. Kern, 2018 Supervised machine learning for population genetics: A new paradigm. Trends in genetics : TIG **34**: 301–312, 29331490[pmid].

Shaun Purcell, Christopher Chang, 2020 Plink 1.9. www.cog-genomics.org/plink/1.9/.

Sun, Y. V. and S. L. R. Kardia, 2008 Imputing missing genotypic data of single-nucleotide polymorphisms using neural networks. European Journal of Human Genetics **16**: 487–495.

Talwar, D., A. Mongia, D. Sengupta, and A. Majumdar, 2018 Autoimpute: Autoencoder based imputation of single-cell rna-seq data. Scientific Reports **8**: 16329.

Tian, C., P. K. Gregersen, and M. F. Seldin, 2008 Accounting for ancestry: population substructure and genome-wide association studies. Human Molecular Genetics **17**: R143–R150.

Wang, C., S. Zöllner, and N. A. Rosenberg, 2012 A quantitative comparison of the similarity between genes and geography in worldwide human populations. PLOS Genetics **8**: 1–16.

Xu, C. and S. A. Jackson, 2019 Machine learning and complex biological data. Genome Biology **20**: 76.

Zou, F., S. Lee, M. R. Knowles, and F. A. Wright, 2010 Quantification of population structure using correlated snps by shrinkage principal components. Human heredity **70**: 9–22, 20413978[pmid].

Zou, J., M. Huss, A. Abid, P. Mohammadi, A. Torkamani, *et al.*, 2019 A primer on deep learning in genomics. Nature Genetics **51**: 12–18.