GENERATION OF ARTIFICIAL SEQUENCE DATA BASED ON CATEGORICAL MIXTURE MODELS

SEQUENCE BIOINFORMATICS WINTER SEMESTER 2019-20

Manuel Glöckler
Finn Mier
Patrick Schirm

January 12, 2020

ABSTRACT

Individual variations of genes can have major impacts on phenotype and have to be considered e.g. in medical development. Therefore we present a method for modeling individual variations for defined genomic regions based on Categorical Mixture Models. We showed that such a model can be used to analyze variational patterns across populations. Furthermore, we showed that the generative power of our model is several times better than simply using variant frequencies.

1 Introduction

While the human genome was decoded 18 years ago, it can still be considered incomplete, especially from a population or individual-specific perspective. Single nucleotide polymorphisms (SNPs), small insertions and deletions (INDELs), and structural variations (SVs) are frequently detected and can result in major phenotypic complications [1, 2]. One example is the well-known sickle cell disease which is caused by SNPs on chromosome 11, leading to a replacement of the amino acid glutamic acid with valine in hemoglobin, which leads to atypic red blood cells that can lead to venous thrombosis, among other things[3]. Due to the rapid decrease of sequencing cost over the past decade, it has become popular to use pan-genome analysis to reveal gene variations within a species or a population. The first human pan-genome study was carried out in 2010; however, only two representative genomes from Africa and Asia were analyzed. The explosion of human whole-genome sequencing since then brings challenges, both because of the huge amount of data, but also because it gives rise to tremendous opportunities to study the pan-genome [2].

One such opportunity is provided by the 1000 genomes project [4], where the genomes of 2,504 individuals from 26 populations in Africa (AFR), East Asia (EAS), Europe (EUR), South Asia (SAS), and the Americas (AMR) were reconstructed. The corresponding data was integrated into the Ensembl [5] database and made publicly accessible through various APIs. In this project, we will use the data provided via Ensembl to model local genomic variations across populations probabilistically. Thereby we implemented efficient algorithms to extract variational data from Ensembl and infer parameters of a generative model based on infinite mixture models.

A reason why such a generative model could be useful was given by Kim and others [6]: The accurate assessment of health disparities requires unbiased knowledge of genetic risks in different populations. However, most genome-wide association studies use genotyping arrays and European samples. Therefore it was suggested, that caution must be taken when extrapolating GWAS results from one population to predict disease risks in another population. One potential solution can be sampling from our modeled distribution to balance knowledge of genetic risks through such artificial samples. Therefore we built and analyzed the generative capability of our suggested learned models.

For a specific genomic region of arbitrary size,e.g. a gene, we extract all SNPs, INDELS, and SV detected in phase three of the 1000 genomes project. Such a set of variations clearly defines an allele X. We, therefore, have 2504 samples from the distribution of alleles $p(X)$ that we can use for inference.

There are two main mechanisms of how alleles can change: Germline mutation and cross-over. Mutation may be the origin of many variations on an evolutionary time scale; however, for small time frames, it is negligible. We will, therefore, focus on cross-over. Cross-over is the exchange of genetic material between two homologous chromosomes that results in recombinant chromosomes during sexual reproduction. We can distinguish single and double cross-overs:

Say X_1, X_2 are homologous chromosomes, X_1X_2, X_2X_1 can be seen as single cross-over and $X_1X_2X_1, X_2X_1X_2$ as double cross-overs. These events are much more frequent and can be considered as the main source of variation for small time frames. Systematic analysis of the patterns in which genetic variants are shared among individuals and populations provides detailed accounts of population history. Due to the shared ancestry of all humans, only a modest number of variants show large frequency differences among populations. Nevertheless about 762,000 variants that are rare within the global sample but much more common in at least one population [4].

These are several facts that a good generative model should take into account. However, considering all these dependencies in a probabilistic fashion would lead to highly complex models and result in computationally expensive inference. Therefore we made general generative assumptions and decided to use a categorical mixture model, which satisfies some but not all of the above-described facts. Additionally, by virtue of this type of model, we also receive a clustering of all samples, which we used to analyze hidden population allele variation.

2 Material

We used sequence data from phase three of the 1000 Genomes project [4], during which a total of 2504 individuals from different populations were sequenced and assembled. Due to the relation of all humans, much of a genomic sequence will be identical with some variants in between. With our generative model, we are only interested in positions that vary, thus we disregard the rest of the sequence during model construction. The 1000 Genomes Project was integrated into the Ensembl [5] database. We, therefore, wrote a program to easily fetch and store variational information of an arbitrary region in the human genome. Using sqlite [7] as a local database, data in ga4gh format is fetched via the Ensembl Rest API [5] as is described in figure 1. For a defined region, all variants of the reference genome assembly (we used GRCh38) are obtained. A variant is defined for a position or region in the reference genome and annotates alternate bases as well as the genotypes and additional metainformation for all individuals. As humans are diploid the genotype is represented as bit tuple, which is 1 if the variation is present on one allele and 0 otherwise. Some alternate notations were used in the 1000 Genomes project, which we normalized into bit tuple format.

Let N be the number of individuals and M the number of variants in a restricted region on a human chromosome, then we can represent all variational information for all individuals as $N \times M$ matrix $X = \{x_{ij}\}$ with $i \in \{1, \dots, N\}, j \in \{1, \cdots, M\}$. Where $x_{ij} \in \{0, 1\}$ is an indicator for sample i if variant j is present in the allele or not. Each individual having two alleles is thereby represented twice in the matrix. This format may be less human readable, but allows for effective inference algorithms described in section 3. In section 4 we tested our probabilistic model for the human leukocyte antigen A (HLA-A) region (Chromosome 6, GRCh38 coordinates: 29,941,260-29,945,884). For this data the inference matrix was of shape 5008×438.

3 Methods

One of the most important choices for a generative model is the type of model. Considering the characteristic vector X_i of variations from sample i, we have to note that in reality each variant x_{ik} is dependent on all other variants $x_{ij} \forall j \neq k$. For simplicity's sake, most projects assume independence when computing allele frequencies. This assumption seems to hold for some regions on a big genomic scale, as was validated with "Linkage Disequilibrium" in the 1000 Genomes project. However, as we want to model small regions this assumption does not seem to be directly applicable. Here we give one example that illustrates this:
Assume we observe the following alleles:

$$X_1 = [1, 1, 0, 0, 0, 0, 0], X_2 = [0, 0, 1, 1, 1, 1, 1]$$

From a simple generative model, where $p(X_i) = \prod_{j=1}^{M} Cat(x_{ij}|\theta)$, inferring the parameters θ for X_1, X_2 would lead to $\theta = (\frac{1}{7}, \frac{1}{7}, \frac{1}{7}, \frac{1}{7}, \frac{1}{7}, \frac{1}{7}, \frac{1}{7})$.For this model two equally likely results could be $X_{1new} = [1, 1, 1, 1, 1, 1, 1]$ or $X_{2new} = [1, 1, 0, 1, 0, 0, 1]$. Recalling that the main mechanism of variation for small time scales is cross over rather than mutation, X_{1new} can be explained by one cross-over event of X_1 and X_2 and therefore would be fairly likely. However, the observed data X_{2new} would assume multiple cross-over events in a small genomic region which would be highly unlikely.

Nevertheless, dropping the assumption of independence, we find ourselves now challenged to consider chromosome-wide dependencies or at least for the defined genomic regions, which would lead to a highly complex model that is outside of the time frame for this project. We, therefore, try to find a model that is complex enough to describe the more complex reality while being computationally inexpensive.

We can see that we can describe our example much better with two mixed categorical distributions $p(X_i) = \sum_k^K \pi_k p(X_i|\theta_k)$ where in this case $\pi_1 = \pi_2 = \frac{1}{2}$ and $\theta_1 = (\frac{1}{2}, \frac{1}{2}, 0, 0, 0, 0, 0)$, $\theta_2 = (0, 0, \frac{1}{5}, \frac{1}{5}, \frac{1}{5}, \frac{1}{5}, \frac{1}{5})$ which

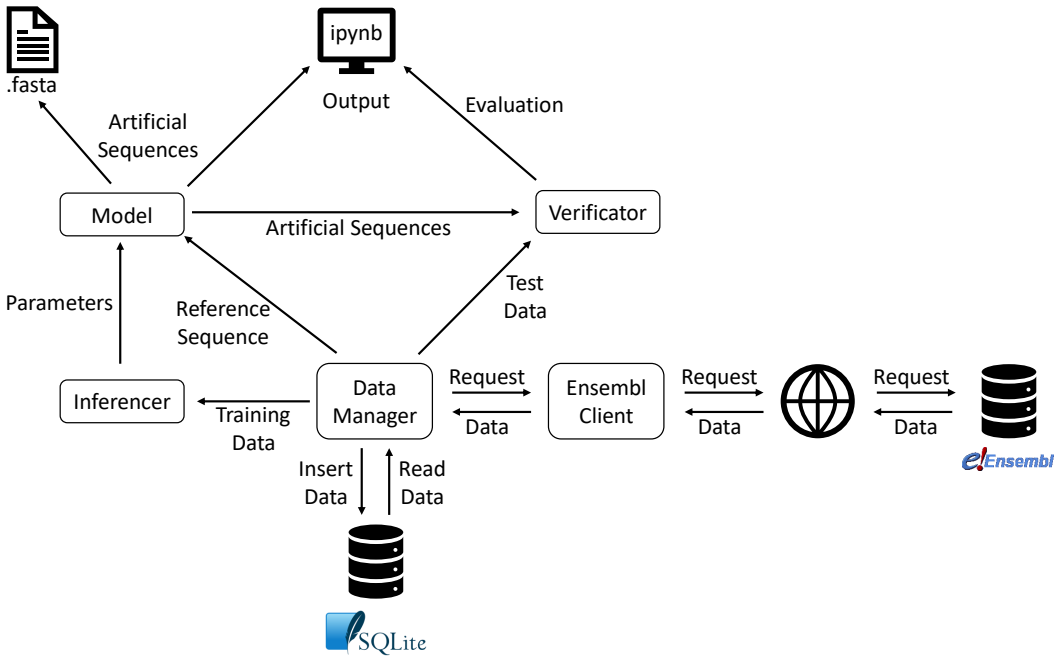

Figure 1: Graph showing the project structure. Data is managed via the class $Data\,Manager$. It is fetched via the Ensembl Rest API [5] and then written to a local SQLite database [7]. The $Inferencer$ class is used to derive the parameters for the $Model$ which then can create sets of variants which in conjunction with the reference sequence are used to generate the artificial sequences which can be written to FASTA files. The evaluation described in 4.2 is performed by the $Verificator$. Finally, the application and outputs are presented in a jupyter notebook.

describes the observed data better. This example illustrates a worst case situation as the datapoints are complements; however, it illustrates that in a more complex scenario the assumption of independence is more valid if we cluster similar alleles together and then infer the parameters of a categorical distribution. Therefore we will use Categorical Mixture Models to describe the searched distribution.

3.1 Categorical Mixture Models

We will use the following notation:

- X: Variational allele data for N samples, where X_n represents the allele of sample n. Each $X_n = (v_1, \ldots, v_{V_n})$ contains V_n variants.
- K: number of mixture components i.e. clusters of similar alleles. $Z = (\{z_n\}_{n=1}^N)$ with $z_n \in \{1, \ldots, K\}$ be the cluster assignment of each datapoint.
- I: Dictionary of all variants M across all samples.
- $\Theta = (\pi, \{\theta_k\})$ set of all parameters of the model. With $\pi = (\pi_1, \ldots, \pi_K)$ and $\theta_k = (\theta_{k1}, \ldots, \theta_{kM})$.

We can denote the assumed generative model then by:

$$p(\{x_n\}_{n=1}^N|\Theta) = \prod_{n=1}^{N} \sum_{k=1}^{K} \pi_k p(x_n|\theta_k) \text{ where } p(x_n|\theta_k) = \prod_{j=1}^{V_n} Cat(x_{nj}|\theta_k)$$

As we were interested in the clustering of our data, we introduce a hidden random variable Z, which maps each allele x_n to a cluster. The inference of all these parameters is nontrivial, as e.g. the maximum likelihood estimation (MLE) solution has no closed-form solution (since there is a sum in the log). The likelihood also is non-convex we have at least $K!$ equivalent MLE solutions, as there are $K!$ ways to assigning K sets of parameters to K components.

Therefore determine an optimal solution is unfavorable; nevertheless, there are multiple ways to approximately infer the parameters. One way would be expectation maximization (EM); however, we choose the Bayesian way and will infer Bayesian Mixture Models with Gibbs sampling.

Thereby we will treed the likelihood parameters as random variables, such that the mixture model can be written as:

$$p(X, Z, \pi, \Theta) = p(\pi|\alpha)p(\Theta|\gamma) \prod_{n=1}^{N} p(z_n|\pi)p(x_n|z_n, \Theta)$$

Where $p(\pi|\alpha) = Dir(\pi|\alpha)$ and $p(\Theta|\gamma) = \prod_{k=1}^{K} Dir(\theta_k|\gamma)$ as the Dirichlet distribution is the conjugate prior of the categorical distribution. Through definition of Z we have $p(z_n|\pi) = Cat(z_n|\pi)$, and similar to above $p(x_n|z_n, \Theta) = \prod_{j=1}^{V_n} Cat(x_{nj}|\theta_{z_n})$

We use a Gibbs sample algorithm as shown in 1 to approximately infer the parameters from the posterior distribution $p(\pi, Z, \Theta|X)$. As Gibbs sampling is a Markov Chain Monte Carlo (MCMC) method the algorithm returns after a "burn-in" period samples from the posterior distribution, which are solutions for our inference problem. We want samples from the unknown joint distribution $p(\pi, Z, \Theta|X)$. Through Gibbs sampling we can sample from the disjoint distributions $p(\pi|X, Z, \Theta), p(\theta_k|X, Z, \pi), p(z_n|X, Z_{-n}, \Theta, \pi)$ and will obtain samples from the joint distribution at convergence. We can analytically determine closed-form solutions for these, as shown in the appendix (Equitation 1,2,3) [8].

Algorithm 1 Gibbs sampling algorithm

Initialize cluster assignments Z and model parameters π, Θ;
while not converged **do**
  Sample $\pi \sim p(\pi|X, Z, \Theta) = p(\pi|Z) = Dir(\pi|\alpha')$
  for k = $1, \ldots, K$ **do**
    Sample $\theta_k \sim p(\theta_k|X, Z, \pi) = p(\theta_k|X, Z) = Dir(\theta_k|\gamma'_k)$
  end for
  for $n = 1, \ldots, N$ **do**
    Sample $z_n \sim p(z_n|X, Z_{-n}, \pi, \Theta) = p(z_n, x_n, \pi, \Theta) = \frac{p(x_n|\theta_k)\pi_k}{\sum_{k'} \pi_{k'} p(x_n|\theta_{k'})}$
  end for
end while

In our case, we do not know the number of clusters K apriori. Globally we would expect that alleles from the same populations are more similar, however, for small snippets of the genome, this assumption does not hold. In this case for conserved regions, we expect it to be smaller, while for less conserved ones it can be higher. Therefore, we still have a model selection problem based on the parameter K. To avoid this problem we will use infinite mixture models IMM, as explained in the next section.

3.2 Infinite Categorical Mixture Model

An ICMM is a Bayesian nonparametric model and defines a probability distribution over infinite-dimensional parameter space. In practice, it only uses a finite subset, as e.g. for clustering it is obsolete to have more clusters than data points. The 'right' number of clusters is adaptively chosen to match the complexity of the data.

We use a Dirichlet process to define the infinite mixture model and can, with some modifications, use the Gibbs sampling algorithm defined above. For a similar generative model:

$$p(X, Z, \pi, \Theta) = p(\pi|\alpha)p(\Theta|\gamma) \prod_{n=1}^{N} p(z_n|\pi)p(x_n|z_n, \Theta)$$

with $p(\Theta|\gamma) = \prod_{k=1}^{K+}$ where $K+$ is the number of observed clusters, $p(z_n|\pi) = Cat(z_n|\pi)$, $p(x_n|z_n, \Theta) = \prod_{j=1}^{V_n} Cat(x_{nj}|\theta_{z_n})$. For convenience we define a prior distribution π as a symmetric Dirichlet with $p(\pi|\alpha) = Dir(\pi|\{\alpha/K\}_{k=1}^{\infty})$.

Unfortunately we can not sample from a infinite dimensional distribution given for π and therefore have to collapse out the mixing components. Sampling Θ is conditional independent from π and therefore remains unchanged. However sampling Z is dependent on the mixture components, thus we obtain:

$$p(z_n = k|Z_{-n}, X, \pi, \Theta) \propto p(z_n = k|Z_{-n})p(x_n|z_n = k, Z_{-n}, \Theta) = p(z_n = k|Z_{-n})p(x_n|\theta_k)$$

Where we can write $p(z_n = k|Z_{-n})$ as:

$$p(z_n = k|Z_{-n}) = \int p(z_n = k|\pi)p(\pi|Z_{-n})d\pi = \frac{\sum_{i \neq n}[z_i = k] + \alpha/K}{N - 1 + \alpha}$$

For $K \to \infty$ and $m_k = \sum_{i \neq n} [z_i = k]$ we obtain:

$$p(z_n = k | Z_{-n}) = \frac{m_k}{N - 1 - \alpha}$$

In order to be a valid probability mass function we need to ensure that $\lim_{K \to \infty} \sum_{i=1}^{K} p(z_n | Z_{-n}) = 1$. We then can write the probability of assigning a sample to any of the unseen clusters as:

$$p(z_n = k_{new} | Z_{-n}) = 1 - \sum_{k=1}^{K+} \frac{m_k}{N - 1 + \alpha} = \frac{\alpha}{N - 1 + \alpha}$$

We now can rewrite our Gibbs sampler as shown in 2. As we collapsed out π the sampler will not infer these; however, with MLE or MAP estimation, we can reconstruct π, given the cluster assignments Z [9].

Algorithm 2 Dirichlet Process Gibbs sampling algorithm

```
Initialize cluster assignments Z and model parameters π, Θ;
while not converged do
    for n = 1, ..., N do
        Sample z_n ~ p(z_n | X, Z_{-n}, Θ)
        if z_n = K_new then
            K^+ += 1
            θ_{K^+} = p(θ | x_n)
        end if
    end for
    If necessary remove empty clusters.
    for k = 1, ..., K^+ do
        Sample θ_k ~ p(θ_k | X, Z)
    end for
end while
```

4 Results and Analysis

4.1 Modeling HLA-A

As explained in section 2 we generated the inference matrix for the HLA-A genomic region. We trained several infinite Categorical Mixture Models (iCMM) as following: First, we used several different Dirichlet Prior Parameters α, γ. These represent our prior knowledge of the inference problem. Small Dirichlet parameters correspond to a draw unbalanced categorical distributions with most probability mass centered on one position, while high parameters correspond to a more uniform distribution. As α is the Dirichlet Prior for π and we do not have any prior knowledge for the clustering, we trained several models for $\alpha \in \{0.1, 1, 10\}$. For the distribution of variation in one cluster we expect a nonuniform distribution and therefore used rather low values $\gamma \in \{0.05, 0.1, 0.5, 1\}$. We trained for all combinations of parameters two models independently, as MCMC methods can be influenced by initialization values.

We computed the Akaike information criterion (AIC) with $AIC = 2|\Theta| - 2 \log p(X)$ for all trained models and sorted all trained models by its value. The AIC combines the goodness of fit (likelihood) by the number of parameters $|\Theta|$. Therefore, it prevents overfitting during model selection. In our case the best model had 23 Clusters, while the best model of lower complexity was ranked at position 13 with 14 Clusters. In Figure 2 we show a PCA plot for the used data, labeled first by population then by cluster assignment of the model. We can observe that the clustering worked; however, from the PCA perspective, we may intuitively cluster the data a bit different. This is most likely due to the dimension reduction only showing about half of the variation present and clusters that appear to be similar can differ strongly from another dimensional perspective.

In Figure 4 we can see the distribution of variations for all clusters. We observe several similar but also some different patterns in these clusters which now represent allelic super-families.

In Figure 3 we can see the distribution of populations across these clusters. We note that we cannot see that populations split up in different clusters; however, we can see a different distribution across clusters. Especially the African population varies across most clusters and there are several African dominated small clusters. This is in line with the results of the 1000 Genomes project, where the African population showed the highest inner population variation [4]. Europeans seem to be similar to most other populations, as they are majorly present in Cluster 5 with all other populations, especially Americans. East Asians are majorly present in Cluster 1,2 and 5. South Asians are also present in this cluster; however, they are also strongly present in Cluster 8, 10 and 12.

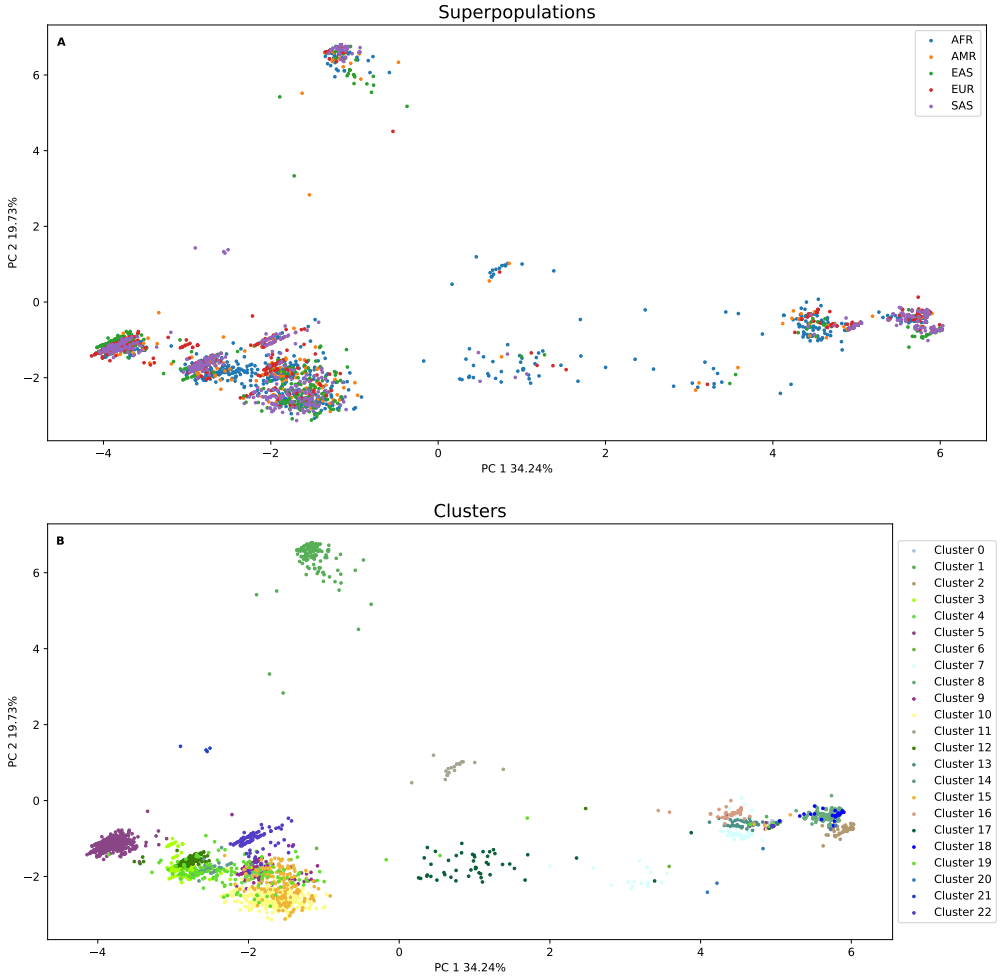

Figure 2: PCA plot that covers in total 54% of the variance in the data in bit vector format. **(A)** The alleles are labeled by their origin superpopulation. **(B)** The alleles are labeled by their assigned cluster

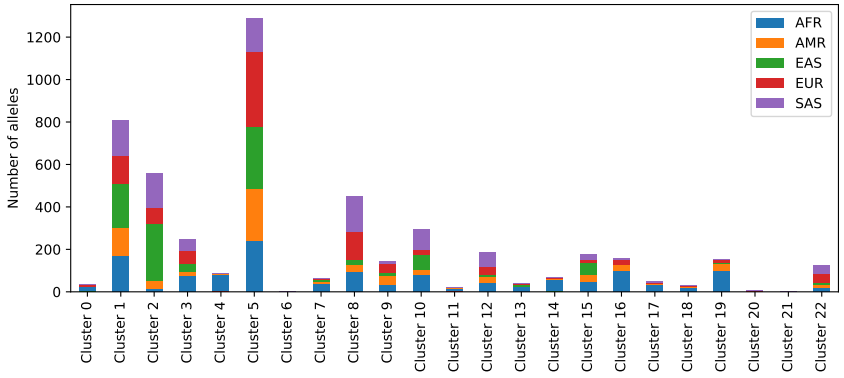

Figure 3: Number of alleles assigned to each cluster, where the fractions assigned to each population are marked by color.

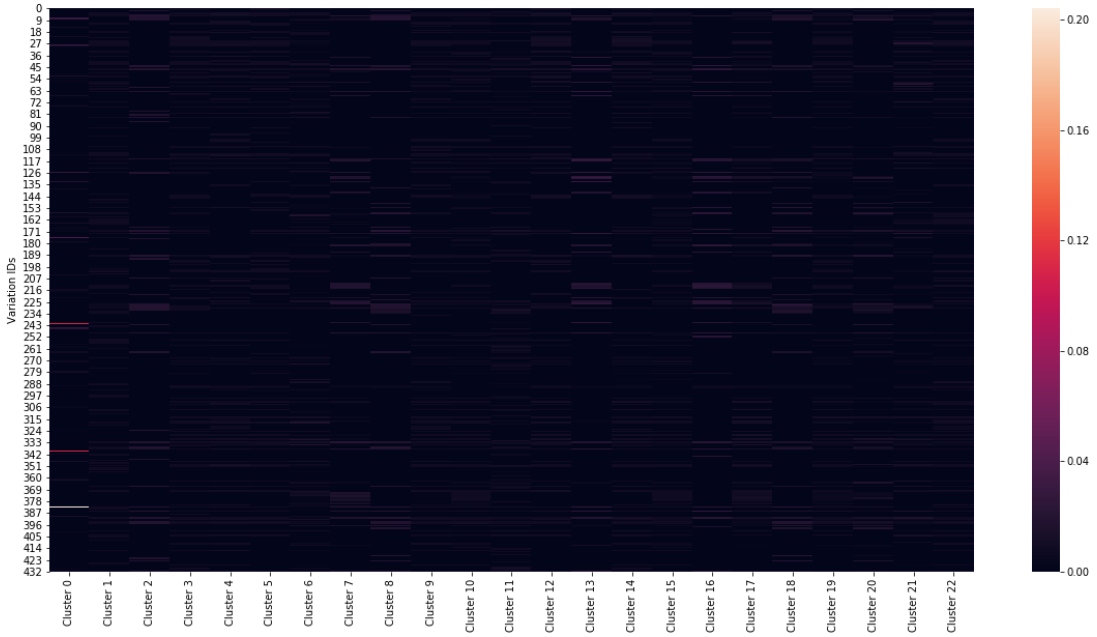

Figure 4: Probability mass distribution across clusters.

4.2 Generative validation

In order to measure our models success at generating realistic sequences we split our multiset of known sequences $\mathbb{S}$ into two disjoint multisets $\mathbb{S}_{train}$ and $\mathbb{S}_{test}$ with $\mathbb{S}_{train} \dot{\bigcup} \mathbb{S}_{test} = \mathbb{S}$. The set $\mathbb{S}_{train}$ is then used to construct the model whereas $\mathbb{S}_{test}$ is used as a reference (ideal) solution. In the following $\hat{X}$ (i.e. the hat) will represent the set constructed from the multiset X. We will assume the natural extension of all set operations to multisets.

We use the set $\mathbb{S}_{train}$ to construct our model and then generate a multiset of sequences $\mathbb{S}_{gen}$ with that model which represents the approximation of realistic sequences generated by our model. The model's success can then be measured by how similar $\mathbb{S}_{gen}$ and $\mathbb{S}_{train}$ are. For this, we want to both measure the similarity of the sequences themselves as well as the distribution of the sequences within the sets.

4.2.1 Similarity of sequences

In order to measure the similarity of the sequences we assign (with the function $\mathcal{A}$) every generated sequence $S_g \in \mathbb{S}_{gen}$ to a sequence in the test set $S_t \in \hat{\mathbb{S}}_{test}$ which is closest to it according to a distance function $dist$:

$$\mathcal{A} : \mathbb{S}_{gen} \to \hat{\mathbb{S}}_{test}$$

$$\mathcal{A}(S_g) = S_t,\ \forall S'_t \in \hat{\mathbb{S}}_{test} \mid dist(S_t, S_g) \leq dist(S'_t, S_g)$$

We thus measure our solutions success in creating similar sequences $m_S(\mathbb{S}_{gen}, \mathbb{S}_{test})$ as the average distance from every generated sequence to its assigned sequence:

$$m_S(\mathbb{S}_{gen}, \mathbb{S}_{test}) = \frac{\sum_{S_g \in \mathbb{S}_{gen}} dist(\mathcal{A}(S_g), S_g)}{|\mathbb{S}_{gen}|}$$

In our tests we choose $dist$ to be the Hamming distance of the characteristic vectors of expressed variants in the respective sequences (as they are used during inference).

4.2.2 Similarity of distributions

For our model to be called successful it not only must produce sequences which are similar to realistic sequences but also to generate them according to a realistic distribution. We can generally observe in our test sets that certain sequences appear a lot more often than others. Thus we also expect generated sequences similar to such sequences to

appear more often than those which are similar to more rare sequences. To measure this we use the well established Kullback–Leibler divergence:

$$D_{KL}(P \,||\, Q) = \sum_{x \in X} P(x) \log \left(\frac{P(x)}{Q(x)} \right)$$

We convert the multisets $\mathbb{S}_{test}$ and $\mathbb{S}_{gen}$, which we want to compare, into distributions the following way:

$$d_{test} : \hat{\mathbb{S}}_{test} \to \mathbb{R},\ d_{test}(S_t) = \frac{|\mathbb{S}_{test} \cap \{S_t\}|}{|\mathbb{S}_{test}|}$$

$$d_{gen} : \hat{\mathbb{S}}_{test} \to \mathbb{R},\ d_{gen}(S_t) = \frac{|\mathcal{A}^{-1}(S_t)|}{|\mathbb{S}_{gen}|}$$

Note that this is dependant on the function $\mathcal{A}$ defined in section 4.2.1. Thus we measure our solutions success in creating sequences with a realistic distribution $m_D(\mathbb{S}_{gen}, \mathbb{S}_{test})$ as:

$$m_D(\mathbb{S}_{gen}, \mathbb{S}_{test}) = D_{KL}(d_{gen} \,||\, d_{test}) = \sum_{S_t \in \hat{\mathbb{S}}_{test}} d_{gen}(S_t) \log \left(\frac{d_{gen}(S_t)}{d_{test}(S_t)} \right)$$

4.2.3 Evaluation

In this section we compare our generative models to themselves and other sets of sequences. We always used two thirds of the available data as training set and the remaining data as test set.

Table 1: Average hamming distance of 200 generated sequences to closest sequences in test set for different inference parameters (as described in 4.2.1) on the HLA-A gene.

m_S	$\alpha = 0.1$	$\alpha = 1.0$	$\alpha = 10.0$
$\gamma = 0.05$	9.605	9.765	9.140
$\gamma = 0.10$	10.290	10.805	12.100
$\gamma = 0.50$	19.645	14.795	18.405
$\gamma = 1.00$	23.260	22.715	18.290

Table 2: Kullback–Leibler divergence between 200 generated sequences and test set for different inference parameters (as described in 4.2.2) on the HLA-A gene.

m_D	$\alpha = 0.1$	$\alpha = 1.0$	$\alpha = 10.0$
$\gamma = 0.05$	1.638986	1.738764	1.768897
$\gamma = 0.10$	1.623178	1.830960	1.627762
$\gamma = 0.50$	1.913128	1.628685	1.767024
$\gamma = 1.00$	1.675794	1.751062	1.645912

At first, in order to analyze the effect of the inference parameters, we ran the inference for the parameters of our generative model with different inference parameters on the HLA-A gene. We built models for each combination of $\alpha \in \{0.1, 1.0, 10.0\}$ and $\gamma \in \{0.05, 0.10, 0.50, 1.00\}$. Then 200 sequences for every model were generated and evaluated against the test set. The results are shown in table 1 and table 2. First, we can see, that the parameter choice does not seem to influence the Kullback–Leibler divergence between the generated sequences and the test sequences. This seems to be more tied to the number of sequences generated, as 200 seems too low to give a good approximation (when using 2000 samples in table 3 the divergence is improved to 1 for the best model). We observed that the choice of alpha does not play a major role in the quality of our result, however, a higher alpha seemed to perform generally better. The choice of gamma seemed to be much more important and lower values for gamma, close to 0.1 seem to perform favorably. This is sensible since a higher gamma moves the probabilities for the variants within a cluster to the uniform distribution thus defeating its purpose.

Table 3: Comparison of our generative model (with the best and the worst model from table 4) with a null model and the training set. We generated 2000 sequences and compared for HLA-A and lactase (LCT) genes.

	m_S for HLA	m_S for LCT	m_D for HLA	m_D for LCT
best model	11.6245	9.052	1.0536	0.9742
worst model	26.102	15.838	1.2005	1.0251
null model	63.2045	46.0085	1.9956	2.4579
training set	2.4505	0.87	0.4128	0.4828

To evaluate our generative model we compared it to a null model that chooses its variants based on their observed relative frequencies in the training set, as well as the sequences from the training set. We did this both for the previously

mentioned HLA-A gene as well as the gene for lactase (LCT, Chromosome 2, GRCh38 coordinates: 135,787,850-135,837,184). As representatives for our model, we choose the best and worst model from the prior analysis. We then generated 2000 sequences with each of the models and compared the generated sets with the test set. The results are presented in table 3. We can see that both of our representative models throughout outperform the null model significantly. The differences between the worst and best models are significant for the average distance of the sequences to their assigned sequences but are lower for the KL-divergence. This further consolidates our findings that the distributions aren't as dependant on the inference parameters. Finally, we can see, that the differences between our model and the training set are still quite significant. As this estimates what we would expect in reality our model can still be improved.

Table 4: AIC for the models for each combination of $\alpha \in \{0.1, 1.0, 10.0\}$ and $\gamma \in \{0.05, 0.10, 0.50, 1.00\}$ on the HLA-A gene. The minimal score was subtracted from all the scores to increase readability.

AIC	$\alpha = 0.1$	$\alpha = 1.0$	$\alpha = 10.0$
$\gamma = 0.05$	0.0	4174.0	7473.9
$\gamma = 0.10$	137.5	3703.8	3205.1
$\gamma = 0.50$	26806.2	6920.1	27655.1
$\gamma = 1.00$	52911.2	44334.3	16627.4

An interesting observation we made is that a low AIC score seems to correlate fairly well with our measured performance. This is important since computing m_S or m_D is much more expensive than just the AIC due to the need to assign every sequence to its closest counterpart. Thus, in practice, it may be more sensible to use the AIC to select a model.

5 Discussion

The observed population distribution across clusters may be explained by population ancestry, as e.g. a huge portion of Americans have ancestors from European populations and similar for East Asians. As each human originates from an ancestral African population [10], samples from African individuals are assigned to most clusters.

From the allele super-families (i.e. inner cluster variational distribution) we can observe mutation or cross-over events that appeared throughout time. Groups of variations that have a high probability, which is interspersed with groups of lower probability variations, can be explained by a past cross-over event. Similarly, single low probability variations can be explained by mutations that appeared recently on an evolutionary time scale.

The ability to generate new gene sequences comes with the possibility to use those sequences as a standard for other tools. Tools in bioinformatics such as MSA tools, variance caller, mapping or phylogenetic tools are commonly used. All of those are using some kind of genetic sequence as input and can output different calculated data for the given sequences. Known sequences are necessary to test and analyze existing tools. The comparison of output and known input makes the benchmarks feasible for those tools. As stated in the introduction, it is important to note that our generated sequences are very similar. Dependent on the desired benchmark used this can be very useful and a good test case or it can completely disqualify the generated sequences as a test set. Further research could include enabling more user settings. It could be made possible to give the user the chance to filter for biological meaningful sequences. Tools such as SnpEff [11] can be used for effect prediction. Using those predictions, generated sequences that are likely to be nonfunctional can be deleted. Filtering that way would negatively affect randomness but would give more biological meaning to the generated sequences. A downside would be that mutations with unknown positive effects could be erased, too.

At this state all variations are used for inference; however, a lot of them do not have a phenotypic effect due to e.g. the degenerated translation to proteins. One could introduce filters to only consider relevant variations. Furthermore one could associate phenotype with different clusters and so one.

More development in verifying our sequences could involve using other existing tools. One option could be using tools such as BWA-MEM [12] to map our sequences against the human reference genome. Doing this we would assume that our sequences are reads such as those from a long read sequencer. A file generated by BWA-MEM [12] can be used in a variant calling tool such as bcftools [13] and the output should enable a comparison to the sequences itself and the underlying mutations in our model. The benefit here would be using our data as a real-world example to evaluate our used methods in real-world conditions.

References

[1] Mark D Adams, Jenny M Kelley, Jeannine D Gocayne, Mark Dubnick, Mihael H Polymeropoulos, Hong Xiao, Carl R Merril, Andrew Wu, Bjorn Olde, Ruben F Moreno, et al. Complementary dna sequencing: expressed sequence tags and human genome project. *Science*, 252(5013):1651–1656, 1991.

[2] Zhongqu Duan, Yuyang Qiao, Jinyuan Lu, Huimin Lu, Wenmin Zhang, Fazhe Yan, Chen Sun, Zhiqiang Hu, Zhen Zhang, Guichao Li, Hongzhuan Chen, Zhen Xiang, Zhenggang Zhu, Hongyu Zhao, Yingyan Yu, and Chaochun Wei. Hupan: a pan-genome analysis pipeline for human genomes. *Genome Biology*, 20(1):149, 2019.

[3] Abdullah Kutlar. Sickle cell disease: A multigenic perspective of a single gene disorder. *Hemoglobin*, 31(2):209–224, 2007.

[4] Adam Auton, Gonçalo R. Abecasis, and David M. Altshuler. A global reference for human genetic variation. *Nature*, 526(7571):68–74, 2015.

[5] Daniel R Zerbino, Premanand Achuthan, and Akanni. Ensembl 2018. *Nucleic Acids Research*, 46(D1):D754–D761, 11 2017.

[6] Michelle S. Kim, Kane P. Patel, Andrew K. Teng, Ali J. Berens, and Joseph Lachance. Genetic disease risks can be misestimated across global populations. *Genome Biology*, 19(1):179, 2018.

[7] Mistachkin J, Hipp D R, and Kennedy D. Sqlite (version 3.22) [computer software]. `https://www.sqlite.org/download.html`.

[8] Christopher M Bishop. *Pattern recognition and machine learning*. springer, 2006.

[9] Kevin P Murphy. *Machine learning: a probabilistic perspective*. MIT press, 2012.

[10] Jun Z. Li, Devin M. Absher, Hua Tang, Audrey M. Southwick, Amanda M. Casto, Sohini Ramachandran, Howard M. Cann, Gregory S. Barsh, Marcus Feldman, Luigi L. Cavalli-Sforza, and Richard M. Myers. Worldwide human relationships inferred from genome-wide patterns of variation. *Science*, 319(5866):1100–1104, 2008.

[11] P. Cingolani, A. Platts, M. Coon, T. Nguyen, L. Wang, S.J. Land, X. Lu, and D.M. Ruden. A program for annotating and predicting the effects of single nucleotide polymorphisms, snpeff: Snps in the genome of drosophila melanogaster strain w1118; iso-2; iso-3. *Fly*, 6(2):80–92, 2012.

[12] Heng Li and Richard Durbin. Fast and accurate long-read alignment with burrows–wheeler transform. *Bioinformatics*, 26(5):589–595, 2010.

[13] Heng Li. A statistical framework for snp calling, mutation discovery, association mapping and population genetical parameter estimation from sequencing data. *Bioinformatics*, 27(21):2987–2993, 2011.