DSB205, Winter 2025
Problem Set 3: Mixture models and dimensionality reduction
Due February 28, 2025 at 11:59pm PST

Submission instructions

- Submit your solutions electronically on the course Gradescope site as PDF files.
- If you plan to typeset your solutions, please use the LaTeX solution template. If you must submit scanned handwritten solutions, please use a black pen on blank white paper and a high-quality scanner app.

1 Mixture model and EM [30 pts]

We will define a mixture model for genetic data and propose an EM algorithm for inference in this model.

We observe data at m SNPs across n individuals. There are K ancestral populations. Each individual i is associated with a single population denoted by the random variable $\boldsymbol{Z}_i$ where $\boldsymbol{Z}_i$ is a binary vector with a single one in the entry corresponding to the population that individual i belongs to. The population membership $\boldsymbol{Z}_i$ is chosen according to a vector of mixture proportions $\boldsymbol{\pi} = (\pi_1, \dots, \pi_K), \pi_k \geq 0, \sum_k \pi_k = 1$. For each SNP j in individual i, $X_{i,j}$ denotes the genotype (0 or 1) at SNP j in individual i (we simplify this model so that each individual has a genotype that takes two values instead of three values). If individal i belongs to population k, $X_{i,j}$ are drawn independently according to a Bernoulli distribution with parameter $f_{j,k}$.

We can write our model as:

$$\begin{aligned} \boldsymbol{Z}_i|\boldsymbol{\pi} &\overset{iid}{\sim} \text{Mult}\left(\boldsymbol{\pi}\right) \\ X_{i,j}|Z_{i,k} = 1, \{Z_{i,l} = 0\}_{l \neq k}, f_{j,k} &\sim \text{Ber}\left(f_{j,k}\right) \end{aligned}$$

Our goal is to estimate the parameters $\boldsymbol{\theta} = (\boldsymbol{\pi}, \boldsymbol{F} = (\boldsymbol{f}_1, \dots, \boldsymbol{f}_K))$ and to infer the population labels of each individual $\boldsymbol{Z}_i$. Here $\boldsymbol{f}_k$ denotes a vector of allele frequencies at each of m SNPs in population k.

(a) The complete data log likelihood for this model can be written as

$$\sum_{i=1}^{n} \log P(\boldsymbol{x}_{i,1:m}, z_i|\theta) = \sum_{i=1}^{n} \log \left(\prod_{k=1}^{K} (\pi_k \prod_{j=1}^{m} P(x_{i,j}|f_{j,k}))^{z_{i,k}} \right)$$

Write the expected complete data log-likelihood: $Q(\boldsymbol{\theta}; \boldsymbol{\theta}^{(t)})$. Express your answer in terms of $r_{i,k} = P(Z_{i,k} = 1|\boldsymbol{x}_{i,1:m}, \boldsymbol{\theta}^{(t)})$. We will use the notation $\boldsymbol{x}_{i,1:m}$ to denote the genotype vector for individual i at SNPs $1, \dots, m$.

Let's first simplify our Log-Likelihood further to make the computation easier:

$$\sum_{i=1}^{n} \log \left(\prod_{k=1}^{K} (\pi_k \prod_{j=1}^{m} P(x_{i,j}|f_{j,k}))^{z_{i,k}} \right) = \sum_{i=1}^{n} \sum_{k=1}^{K} z_{i,k} \log \left(\pi_k \prod_{j=1}^{m} P(x_{i,j}|f_{j,k}) \right)$$

$$= \sum_{i=1}^{n} \sum_{k=1}^{K} \sum_{j=1}^{m} z_{i,k} \log\left(\pi_k P(x_{i,j}|f_{j,k})\right) = \sum_{i=1}^{n} \sum_{k=1}^{K} \sum_{j=1}^{m} z_{i,k} (\log \pi_k + \log P(x_{i,j}|f_{j,k}))$$

$$Q(\theta; \theta^{(t)}) = \mathbb{E}_{\theta^t} LL(\theta) = \mathbb{E}_{\theta^t} \left[\sum_{i=1}^{n} \sum_{k=1}^{K} \sum_{j=1}^{m} z_{i,k} (\log \pi_k + \log P(x_{i,j}|f_{j,k})) \right]$$

$$= \sum_{i=1}^{n} \sum_{k=1}^{K} \sum_{j=1}^{m} \mathbb{E}_{\theta^t} \left[z_{i,k} (\log \pi_k + \log P(x_{i,j}|f_{j,k})) \right]$$

Now, the only random variable is $z_{i,k}$, and everything else is a constant. So, we get:

$$= \sum_{i=1}^{n} \sum_{k=1}^{K} \sum_{j=1}^{m} \mathbb{E}_{\theta^t} \left[z_{i,k} \right] \left(\log \pi_k + \log P(x_{i,j} | f_{j,k}) \right)$$

$$= \sum_{i=1}^{n} \sum_{k=1}^{K} \sum_{j=1}^{m} r_{i,k} (\log \pi_k + \log P(x_{i,j} | f_{j,k}))$$

(b) Use Bayes theorem to write the expression for $r_{i,k}^{(t)}$ in terms of $\boldsymbol{x}_{i,1:m}$, $\pi_k^{(t)}$ and $\boldsymbol{f}_k^{(t)}$.

$$r_{i,k} = P(Z_{i,k} = 1|x_{i,1:m}, \theta^{(t)}) = \frac{P(x_{i,1:m}|Z_{i,k} = 1, \theta^{(t)})P(Z_{i,k} = 1|\theta^{(t)})}{P(x_{i,1:m}|\theta^{(t)})}$$

$$P(x_{i,1:m}|Z_{i,k} = 1, \theta^{(t)}) = \prod_{j=1}^{m} f_{j,k}^{x_{i,j}}(1 - f_{j,k})^{1-x_{i,j}}$$

$$P(Z_{i,k} = 1|\theta^{(t)}) = \pi_k$$

$$P(x_{i,1:m}|\theta^{(t)}) = \sum_{k'=1}^{K} P(x_{i,1:m}|Z_{i,k'} = 1, \theta^{(t)})P(Z_{i,k} = 1|\theta^{(t)})$$

$$= \sum_{k'=1}^{K} \pi_{k'} \prod_{j=1}^{m} f_{j,k'}^{x_{i,j}}(1 - f_{j,k'})^{1-x_{i,j}}$$

$$r_{i,k} = \frac{\pi_k \prod_{j=1}^{m} f_{j,k}^{x_{i,j}}(1 - f_{j,k})^{1-x_{i,j}}}{\sum_{k'=1}^{K} \pi_{k'} \prod_{j=1}^{m} f_{j,k'}^{x_{i,j}}(1 - f_{j,k'})^{1-x_{i,j}}}$$

(c) We would like to obtain the MLE of $\boldsymbol{\theta}$. Derive the M-step updates for π_k and $f_{j,k}$ in terms of $r_{i,k}$.

$$Q(\theta;\theta^t) = \sum_{i=1}^{n}\sum_{k=1}^{K}\sum_{j=1}^{m} r_{i,k}(\log \pi_k + \log P(x_{i,j}|f_{j,k}))$$

Let's take the derivative with respect to pi first

$$\frac{\partial Q}{\partial \pi_k}\sum_{i=1}^{n}\sum_{k=1}^{K}\sum_{j=1}^{m} r_{i,k}\log \pi_k + \text{const} = \sum_{i=1}^{n}\sum_{k=1}^{K} r_{i,k}\log \pi_k = \sum_{k=1}^{K}\log \pi_k \sum_{i=1}^{n} r_{i,k}$$

$$= \sum_{i=1}^{n} r_{i,k}\frac{\partial Q}{\partial \pi_k}\sum_{k=1}^{K}\log \pi_k \implies \frac{\partial Q}{\partial \pi_k} = \sum_{i=1}^{n} r_{i,k}\frac{1}{\pi_k} = 0$$

Now, using the normalization constraint of the mixture weights, and the fact that we expect the mixture weights to be proportional to the expected cluster assignment:

$$\sum_{k=1}^{K}\pi_k = 1 \quad \pi_k \propto \sum_{i=1}^{n} r_{i,k} \implies$$

$$\sum_{k=1}^{K}\pi_k = \sum_{k=1}^{K}\frac{\sum_{i=1}^{n} r_{i,k}}{n} = 1 \implies \pi_k = \frac{1}{n}\sum_{i=1}^{n} r_{i,k}$$

Taking the derivative with respect to f

$$\frac{\partial Q}{\partial f_{j,k}}\sum_{i=1}^{n}\sum_{k=1}^{K}\sum_{j=1}^{m} r_{i,k}(\text{const} + \log P(x_{i,j}|f_{j,k}))$$

$$= \frac{\partial Q}{\partial f_{j,k}}\sum_{i=1}^{n}\sum_{k=1}^{K}\sum_{j=1}^{m} r_{i,k}(\log(f_{j,k}^{x_{i,j}}(1-f_{j,k})^{1-x_{i,j}}))$$

$$= \frac{\partial Q}{\partial f_{j,k}}\sum_{i=1}^{n}\sum_{k=1}^{K}\sum_{j=1}^{m} r_{i,k}[x_{i,j}\log f_{j,k} + (1-x_{i,j})\log(1-f_{j,k})] = 0$$

Since we're taking the derivative with respect to j and k, we can remove those summations:

$$\frac{\partial Q}{\partial f_{j,k}} \sum_{i=1}^{n} r_{i,k}[x_{i,j} \log f_{j,k} + (1 - x_{i,j}) \log(1 - f_{j,k})] = 0$$

$$\sum_{i=1}^{n} r_{i,k}[\frac{x_{i,j}}{f_{j,k}} - \frac{(1 - x_{i,j})}{(1 - f_{j,k})}] = 0$$

$$\sum_{i=1}^{n} r_{i,k}(\frac{x_{i,j}}{f_{j,k}}) - \sum_{i=1}^{n} r_{i,k}\frac{(1 - x_{i,j})}{(1 - f_{j,k})} = 0$$

$$\sum_{i=1}^{n} r_{i,k}(\frac{x_{i,j}}{f_{j,k}}) = \sum_{i=1}^{n} r_{i,k}\frac{(1 - x_{i,j})}{(1 - f_{j,k})}$$

$$(1 - f_{j,k}) \sum_{i=1}^{n} r_{i,k}(\frac{x_{i,j}}{f_{j,k}})f_{j,k} = f_{j,k} \sum_{i=1}^{n} r_{i,k}\frac{(1 - x_{i,j})}{(1 - f_{j,k})}(1 - f_{j,k})$$

$$(1 - f_{j,k}) \sum_{i=1}^{n} r_{i,k}(x_{i,j}) = f_{j,k} \sum_{i=1}^{n} r_{i,k}(1 - x_{i,j})$$

$$\sum_{i=1}^{n} r_{i,k}(x_{i,j}) - f_{j,k} \sum_{i=1}^{n} r_{i,k}(x_{i,j}) = f_{j,k} \sum_{i=1}^{n}(r_{i,k} - r_{i,k}x_{i,j})$$

$$\sum_{i=1}^{n} r_{i,k}(x_{i,j}) - f_{j,k} \sum_{i=1}^{n} r_{i,k}(x_{i,j}) = f_{j,k} \sum_{i=1}^{n}(r_{i,k} - r_{i,k}x_{i,j})$$

$$\sum_{i=1}^{n} r_{i,k}(x_{i,j}) - f_{j,k} \sum_{i=1}^{n} r_{i,k}(x_{i,j}) = f_{j,k} \sum_{i=1}^{n} r_{i,k} - f_{j,k} \sum_{i=1}^{n} r_{i,k}x_{i,j}$$

$$\sum_{i=1}^{n} r_{i,k}(x_{i,j}) = f_{j,k} \sum_{i=1}^{n} r_{i,k}$$

$$f_{j,k} = \frac{\sum_{i=1}^{n} r_{i,k}(x_{i,j})}{\sum_{i=1}^{n} r_{i,k}}$$

2 EM Implementation and data analysis [35 pts]

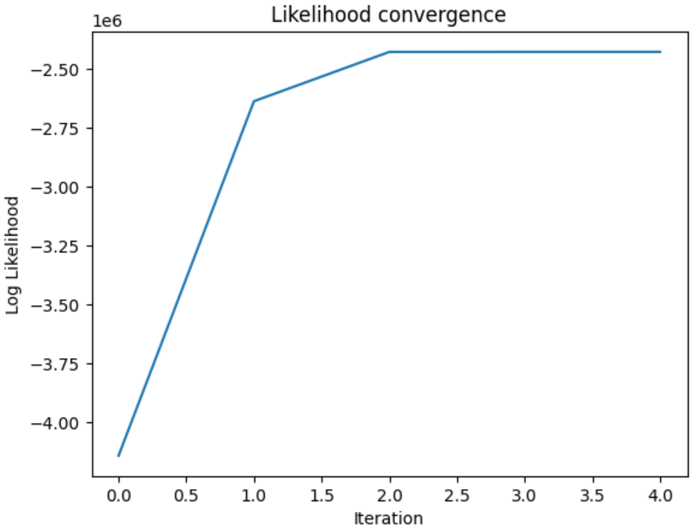
You are given two datasets each of which contains genotype data from 1000 individuals at 5000 SNPs. Dataset 1 contains the values of the hidden values and parameters that you can use to debug.

For this question, you will implement an EM algorithm and obtain MLE of the parameters of the mixture model for each dataset as well as infer the hidden variables. The first dataset (dataset 1) includes the true values of the θ and Z and can be used to test your implementation. The second dataset (dataset 2) does not include the true parameter values.

Use the following specifications for the EM.

- *Initialization*: To initialize, draw each $f_{j,k}$ from a uniform random variable. Draw each π_k from an exponential random variable and normalize (remember that π_k needs to sum to 1). Alternately, draw π from a Dirichlet distribution.
- *Stopping criterion*: Run for at most 100 iterations or till the change in log likelihood $< 10^{-8}$.
- *Number of random restarts*: Since EM only reaches a local optimum, we often run the algorithm with multiple random initializations and pick the solution with the highest log likelihood. We use 3 restarts.
- *The number of mixture components or populations K*: Set $K = 2$.

(a) For dataset 1, plot the log likelihood as a function of iteration for a single run of EM (*i.e.* for a single initialization of EM).



(b) For dataset 1, what is the MLE of π?
The MLE of pi appears to be 0.394 and 0.606.

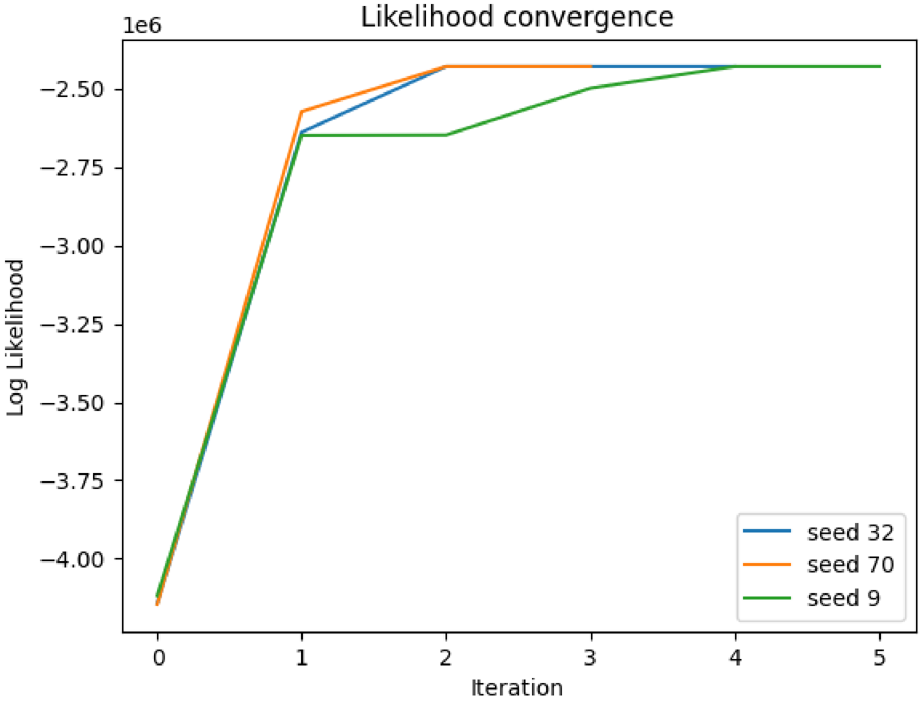
(c) For dataset 1, fixing π and f at its MLE from the previous question, we can compute the posterior probabilty of Z_i for each indvidual. Note that these posterior probabilities are computed in the last E-step of the EM algorithm. We would like to compute the accuracy of the inferred Z_i by comparing the posterior probability of Z_i to the true Z_i available in mixture1.ganc file (each row of this file indicates which population the corresponding individual belongs to). The inferred population label for individual i is set to the MAP (maximum a posteriori) estimate, *i.e.*, choose the population with the maximum value of the posterior probability. Our measure of accuracy is simply the fraction of individuals for which the MAP estimate matches the true population label.

On dataset 1, What is the accuracy of the inferred population label?

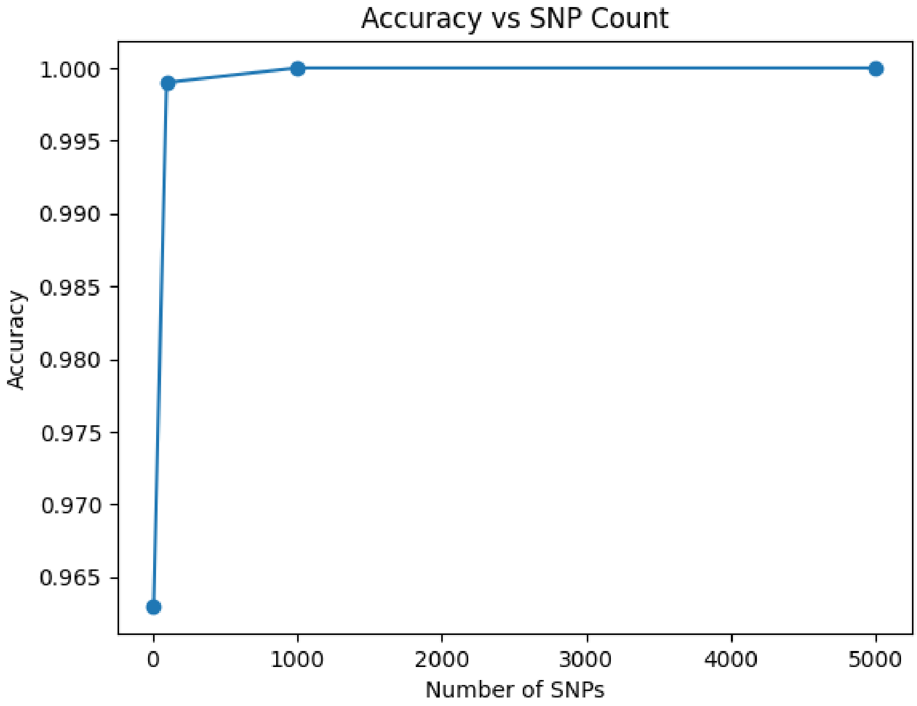
The accuracy of the inferred population label seems to be 100%.

(d) How similar are the solutions on dataset 1 across the different initializations. Report the log likelihoods of the optimum found as well as the accuracy of the population label.

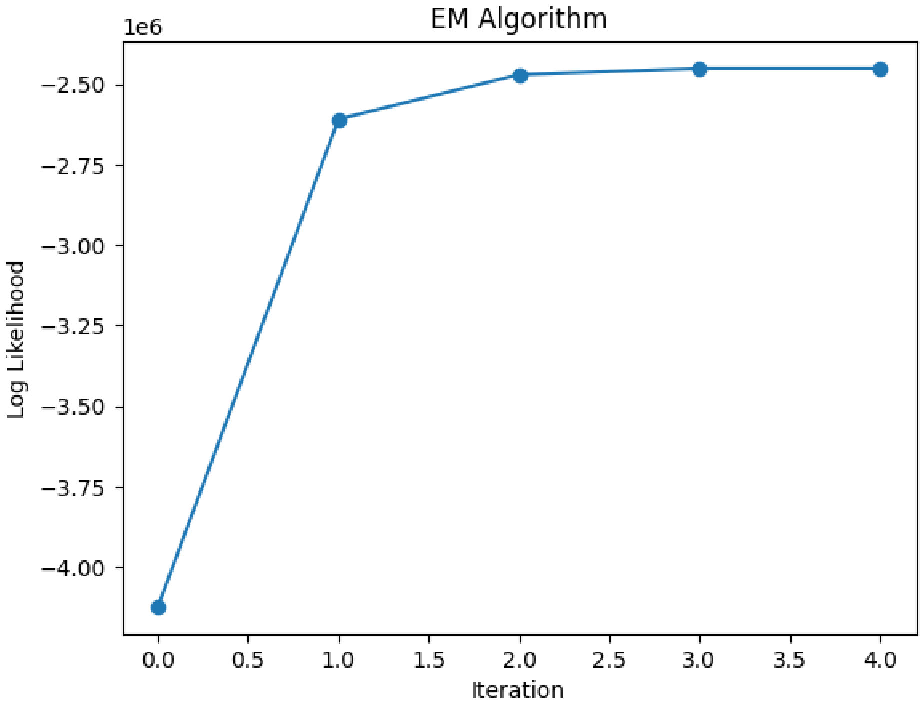
Despite different initialization values, the solutions on data set 1 converge on the same likelihood value, and also attain 100% accuracy. Here is a plot to show how all the likelihoods converge to the same value across 3 different random initializations.

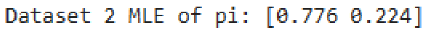


(e) To see how accuracy of predicting population label 1 changes as we get more SNPs, redo the EM using only the first 10, 100, 1000 SNPs and 5000 SNPs. Plot accuracy as a function of number of SNPs.



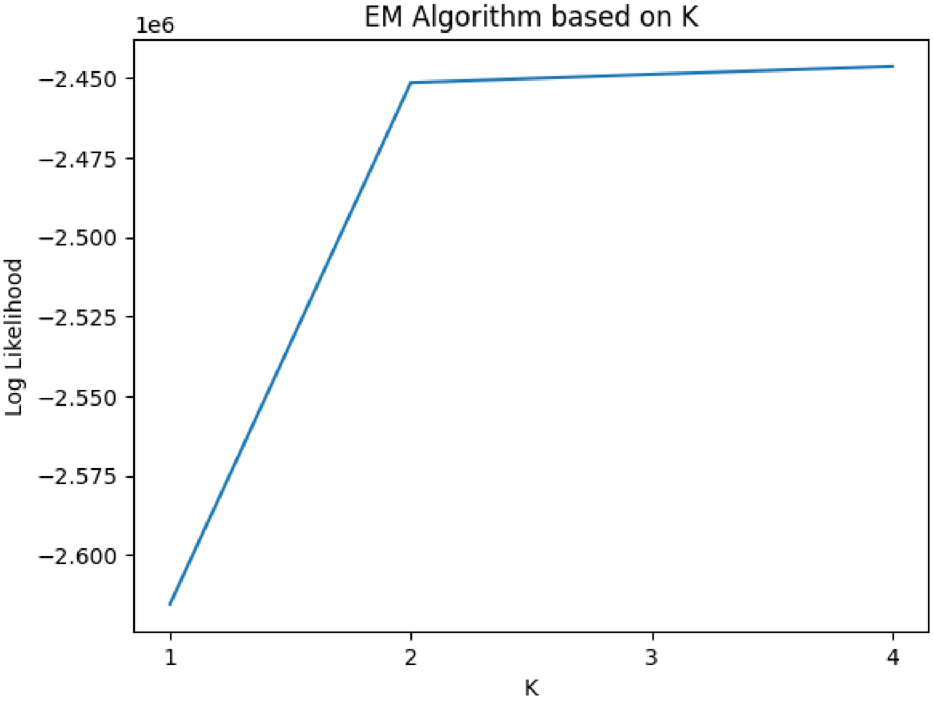
(f) For dataset 2, run EM and report the MLE of π.





```
Dataset 2 MLE of pi: [0.776 0.224]
```

(g) For dataset 2, plot how the log likelihood varies with K (try $K = 1, \ldots, 4$). Which value of K would you choose based on this plot ?



Based on the plot, I would choose K=2. We can see that at K=2 greatly increases the log likelihood from K=1, which suggests that it does a better job of explaining our data. However, subsequent values of K flatten the log likelihood out, showing that they don't significantly improve our model.

Implementation details: The fixed point to which EM algorithm converges is not guaranteed to be the global optimum. To obtain a better solution, it is recommended to start the algorithm at multiple random points in the parameter space and pick the theta with the maximum likelihood across the random restarts.

The log likelihood is guaranteed to be non-decreasing in each iteration and is a useful check of your implementation.

You will need to compute the posterior probabilities $r_{i,k}$ by applying Bayes theorem. One difficulty is that the likelihood, $P(\boldsymbol{x}_{i,1:m}|z_i = k, \theta)$ can become very small when the number of SNPs is large leading to underflow errors. One solution is to work with the log likelihood and to notice that the posterior probability is invariant if we re-scale each likelihood by the same amount. Let $l_i^* = max_{k'} l_{i,k'}$. We can then write

$$\begin{aligned} r_{i,k} &= \frac{\exp(l_{i,k})}{\sum_{k'} \exp(l_{i,k'})} \\ &= \frac{\exp(l_{i,k} - l_i^*)}{\sum_{k'} \exp(l_{i,k'} - l_i^*)} \end{aligned}$$

The second line is better behaved in practice.

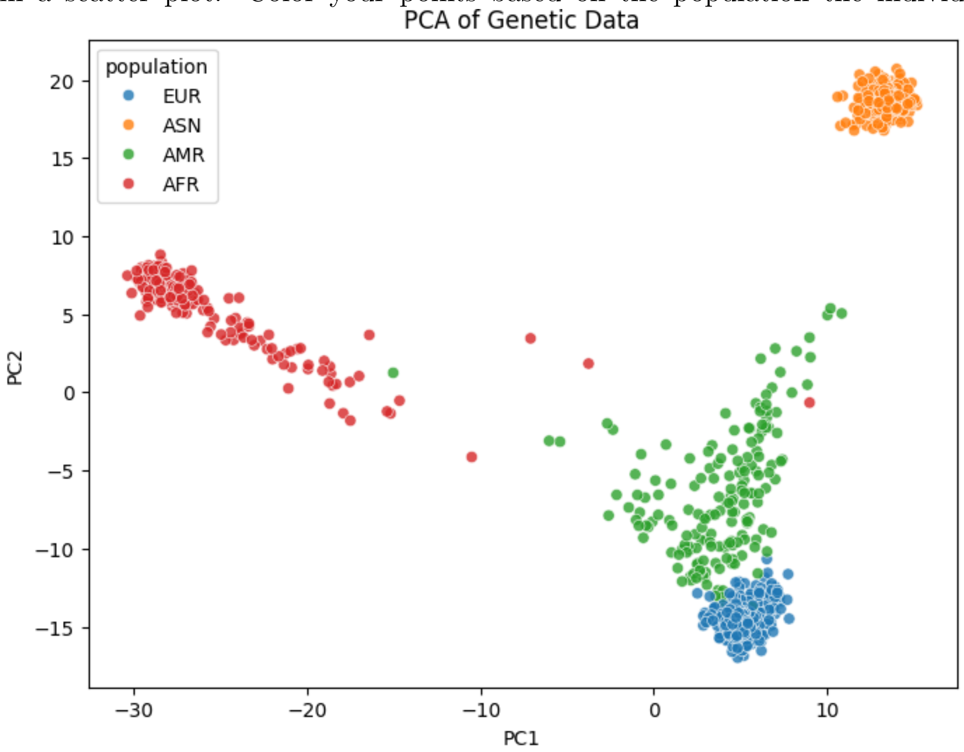
3 Principal Component Analysis [15 pts]

We will apply Principal Component Analysis on the genetic data of 1,092 (real) individuals from the 1000 Genomes Project.

You are given the genotype data containing $M = 13,237$ SNPs for $N = 1,092$ individuals of African, East Asian, (Admixed) American and European descent. Both the SNP data and the true population labels are given.

If using R, please set the seed for the random number generator to 0 in the beginning of your script (set.seed(0)). If using sklearn, use np.random.seed(0).

(a) Use the prcomp function in R to run PCA on this dataset and plot PC1 against PC2 in a scatter plot. Color your points based on the population the individual comes from.

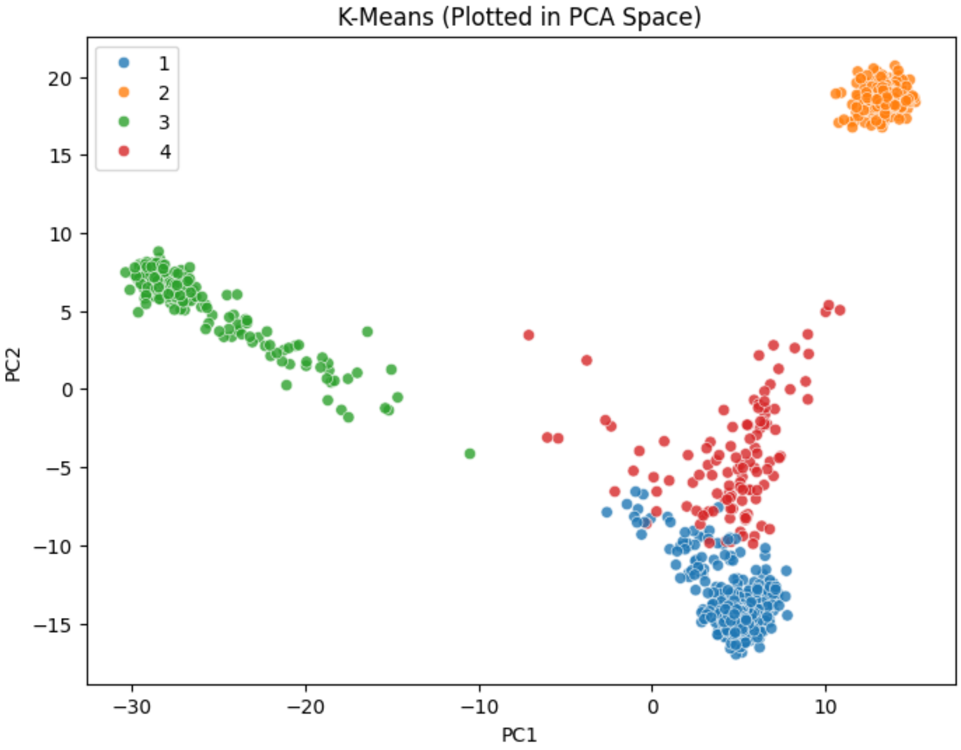


(b) Briefly comment on why you think the first two components of PCA exhibit clustering by population. Why can PCA successfully capture population membership?

PCA can capture population membership because the first few components maximize the variance within the data. Since the allele variances between the populations are already high due to demographics, and PCA captures the variances, PCA will successfully show these as different clusters in the plot.

(c) Use the kmeans function in R to run the k-means algorithm on the SNP data (*not* the PCs) with K=4 and nstart=5 (*i.e.* use 5 different initializations since kmeans is not guaranteed to converge to a global optimum). Now rename all the clusters you have obtained by size: the largest cluster by number of individuals should be named Cluster1 and the smallest cluster should be named Cluster4. Generate the same plot as in part (a) (PC1 vs. PC2), but this time color your points based on cluster assignments (*i.e.* was an individual assigned to cluster 1, 2, 3 or 4) instead of the population labels.

(The location of the points on the PC1 vs. PC2 plot should not change, but their colors should now indicate cluster membership instead of true population labels.)



(d) Match clusters to population labels by inspection (*e.g.* "Cluster 1 most closely resembles the ASN population.") What fraction of the cluster assignments agree with the true population labels?

Through inspection, it seems cluster 1 resembles EUR population, cluster 2 resembles ASN, cluster 3 resembles AFR, and cluster 4 is AMR. Roughly 93.4% of cluster assignments agree with the true population labels.