Inferring the genetic ancestry of individuals using a labeled training set and principal components analysis

Principal components analysis (PCA) is a dimensional-reduction technique that maps high-dimensional data onto lower-dimensional space (principal components) in such a way that each lower-dimensional component is orthogonal to (i.e., independent of) the others. PCA has been extensively used in the medical and population genetics community as a tool to visualize and summarize the genetic ancestry of individuals and is regularly used in our computational pipelines to infer and classify samples of unknown ancestry. In this use case, we map the high-dimensional space of genotypes in a call set onto a lower-dimensional space of principal components. For a given individual with millions of genotypes, we can reduce the amount of data that goes into the ancestry classification problem into a handful of principal components; this data is then fed into a classification algorithm of our choosing that runs on labeled training data to generate a model that can be used to predict the classification of unlabeled samples.

Your assignment recapitulates this common analysis problem: directly inferring the ancestry of samples with missing ancestry labels. You are provided with a file containing the genotypes of each individual in a cohort of samples with mixed labeling status: some samples have known (labeled) ancestries, and others do not. Your task is to generate the following:

1. A set of principal component values for the call set (i.e., a table containing the PC values for each sample for each principal component)
2. A final classification or ancestry label assigned to each sample that is missing a label
3. A visualization of the distribution of PC values for each sample in the call set, along with the labeled and predicted ancestry classifications

A few general remarks:
You may use whatever pre-existing computational tools/packages you wish to handle the genotype data, perform PCA, and cluster/classify data. However, to demonstrate competence in the group's core languages, please submit your scripts and any work done outside pre-existing packages in Python, UNIX/bash, R, and/or Matlab. Please document and comment all code that went into the generation of the final deliverables. When you are finished with the project, please upload the results and the code you wrote to generate the results into a private GitHub repo. You will need to send me your GitHub username so that I can add you to the repo.

A few hints:

As stated above you can use any tools/packages or language that you want, but here are some packages you may find useful in completing the assignment: vcftools, plink, eigensoft (smartpca). Also, to give you a sense of how our lab writes and documents code, here is our GitHub repo for gnomAD (https://github.com/broadinstitute/gnomad_methods). Note that this repo relies heavily on Hail (https://hail.is/docs/0.2/tutorials-landing.html).

Successful, clean PCA on human genetic data will require filtering data to high-quality variants that are linkage disequilibrium (LD)-pruned. In general, we like to run PCA on high-callrate, bi-allelic, common (allele frequency >0.01) variants that are pruned to $r^2<0.1$; but you are welcome to run PCA on whichever set of variants you find work best for you. You will also need to normalize your genotypes after filtering to high-quality variants.

Make sure you explain your choice of classification algorithm and any parameters chosen (if applicable).