Genetics & Genomics – Python Exercise 1

We have received genotyping data from ~50 human samples. We plan to perform a GWAS analysis (even if low powered). To do so we will need to remove potential confounding factors from the data such as sex, batch or ethnicity since they can hide the biological signal of interest. However, the clinical data from these samples only contains information about the sex, batch effect, etc. (data not provided for this exercise) but no information about the ethnicity. **Your goal is to use the 1000 Genome Project dataset for inferring the ethnicity of these samples.**

You are given a VCF file "genotypes.vcf" containing a merged dataset of ~2500 samples from the 1000G project, and the ~50 unknown samples from our study. To facilitate your task the VCF file is already filtered to contain a restricted list of ~1500 SNPs that discriminate the different ethnic populations.

An additional file, "ethnicities.txt", downloaded from 1000 Genome project website, contains the list of 1000 Genome samples and their known ethnicity.

For inferring ethnicities, we will simply perform a PCA of the genotyping data (merged 1000G and our samples), hoping that all population are sufficiently separated, so that our samples will be clearly distinguishable.

1. 1000G ethnicities

a. Start by loading the ethnicity file in Python. You can use the *pd.read_csv()* function from the **Pandas** package.

b. Plot the 3rd column as a bar plot to summarize the number of samples from all the different subpopulations. Note that the bar plot needs a summary indicating the number of observation (human samples) in each category (subpopulation). Sort the bars by their values. Make sure that category names are fully visible (hint: look into the ways of rotating of *xticklabels* parameter).

c. Also generate a PDF file containing the plot using the *plt.savefig()* functions. Specify the resolution (300 should be good enough) and make the figure background transparent.

d. Plot similarly the 4th and 5th columns as bar plot and compare their content. You can try playing with the *plt.subplots()* function parameters to visualize two plots side by side. Sort the bars by their values.

Note: For this exercise, use the **matplotlib** library for visualization purposes.

2. Reading the VCF file

Read the VCF file (be careful, first two lines are comments/header (use the *skiprow* parameter), third line should be used to generate the column names). Check that the output dataframe is properly generated. You can use the *pd.read_csv()* function from the **Pandas** library.

Note: The rows correspond to the different VCFs. The 9 first columns are VCF annotations. The other columns are the samples genotype. LA0# are the ~50 unknown human samples and HG# and

NA# are the ~2500 1000G samples. The values of the two alleles on each chromosome of the individuals are separated by "|". A value of 0 correspond to the reference allele (REF) and 1 to the alternative allele (ALT).

3. Merge both ethnic and VCF datasets

a. Check which samples from the VCF are also present in the ethnicity file, and which ones are not (the “Unknown” samples that need to be predicted).

b. For the samples with missing annotation, set their ethnicities to "Unknown". Merge the ethnicities dataframe with the one with samples having missing annotations.

c. Generate a pie chart (*pie()* function) of the final ethnicities (including the "Unknown" samples). Similar to the bar plot input, the *pie()* function needs a table indicating the number of observation per category.

4. Prepare the data for the PCA

a. Create a matrix called *df_for_pca*, from the VCF file, that contains only the genotyping data (without the first 9 columns).

b. Then, transform this matrix in a numerical matrix containing 0 for homozygous ref values, 1 for heterozygous, and 2 for homozygous alternative. Remember that for the PCA to be performed on the samples (and not the genes), the rows of the matrix should be the samples (and thus the columns are the SNPs). Be careful with the class of the values, it needs to be numerical so that the PCA can run without error.

5. Running PCA

a. Perform the PCA using the *PCA()* function from the **scikit-learn** library.

b. Visualize the first two components of the PCA and color samples by ethnicity. Are the different groups clearly visible/separated?

c. Visualize the first three components of the PCA in 3D using the matplotlib library, color samples by ethnicity and save the final figure into PDF.

d. Find a way to highlight the “Unknown” samples more clearly and save the final figure into PDF.

e. Highlighting the “Unknown” samples with different markers and different sizes.

f. Generate an interactive PCA plot in 3D using the **Plotly** package and the *Scatter3d*() function.

6. Conclusion

Can you conclude on the ethnicities of the “Unknown” samples?