CAAP 2019 Genetics 🧬 Lab 6

What can modern genetics teach us?

Author: Chris Porras

In [1]:

```
################### Run this cell to load essential packages ###############
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import matplotlib as mpl
from mpl_toolkits import mplot3d
import plotly.express as px
import seaborn as sns
np.random.seed(42)

%matplotlib inline
import matplotlib.font_manager as fm
fm.findSystemFonts(fontpaths=['/Library/Fonts/'], fontext='ttf')

mpl.rcParams['font.family'] = "Arial"
mpl.rcParams['font.sans-serif'] = "Arial"
mpl.rcParams['font.size'] = 8

rc = {'lines.linewidth': 3,
      'axes.labelsize': 18,
      'axes.titlesize': 18,
      'axes.facecolor': 'DFDFE5'}
sns.set_context('notebook', rc=rc)
sns.set_style('darkgrid', rc=rc)
sns.set_palette('colorblind')

plt.rc('font', family="Arial")
mpl.font_manager._rebuild()
```

In [2]:

```
### Load in data files
df = pd.read_csv('data/1K_genomes_PCA.csv')
PC_load = np.load('data/1kGenomesPCloading.npy')
superpop_key = pd.read_csv("data/20131219.superpopulations.tsv", sep='\t')
pop_key = pd.read_csv('data/20131219.populations.tsv',sep='\t').iloc[:,0:3]
```

Peek at the data set we'll be studying

In [3]:
```
df.iloc[[i for i in np.arange(5)*200]]
```

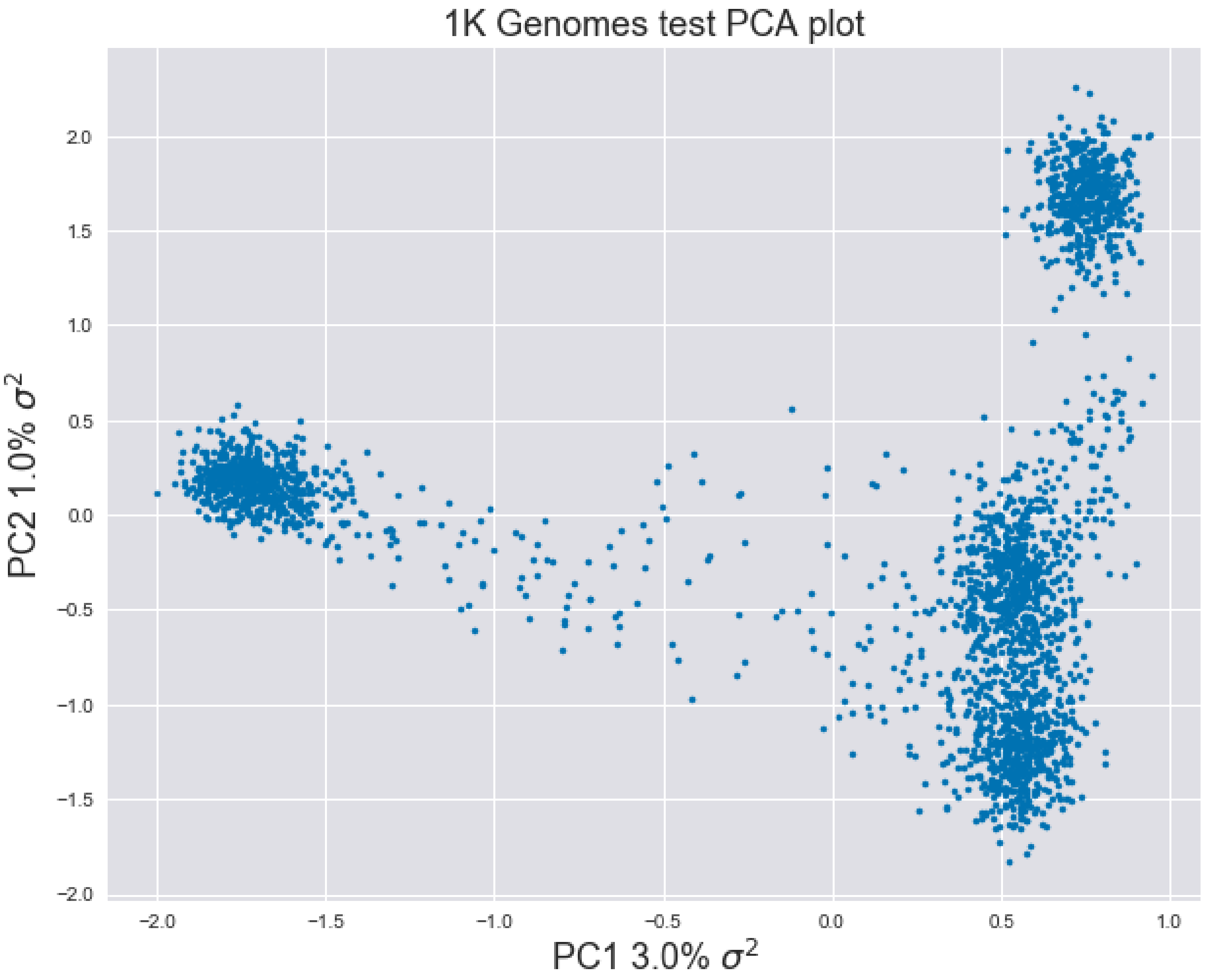
Out[3]:

	Sample_ID	PC1_0.03008	PC2_0.01143	PC3_0.00354	Super_Population	Population	Population_
0	HG00096	0.475450	-1.353762	0.507889	UNKNOWN	UNKNOWN	
200	HG00446	0.797393	1.689049	-0.395801	EAS	CHS	Southern H
400	HG01204	0.322578	-1.194499	-0.103794	AMR	PUR	Puerto Ric
600	HG01767	0.529042	-1.086297	-0.288918	EUR	IBS	Iberian p
800	HG02128	0.675850	1.839144	0.212096	EAS	KHV	Kinh in Ho C

> We want to infer the population that our first sample was chosen from by comparing genetic data across a global sample from the 1000 Genomes Project (https://en.wikipedia.org/wiki/1000_Genomes_Project).

Unlabeled PCA plot

```
In [4]: plt.figure(figsize=(10,8))
        plt.plot(df['PC1_0.03008'],df['PC2_0.01143'],'.')
        plt.title('1K Genomes test PCA plot')
        plt.xlabel(f'PC1 {np.round(PC_load[0],2)*100}% '+r'$\sigma^2$')
        plt.ylabel(f'PC2 {np.round(PC_load[1],2)*100}% '+r'$\sigma^2$')
        plt.show()
```



We can plot the first two principal components for each sample, but this doesn't tell us much without labels!

```
In [5]: #### Key for Population codes ####
        superpop_key
```

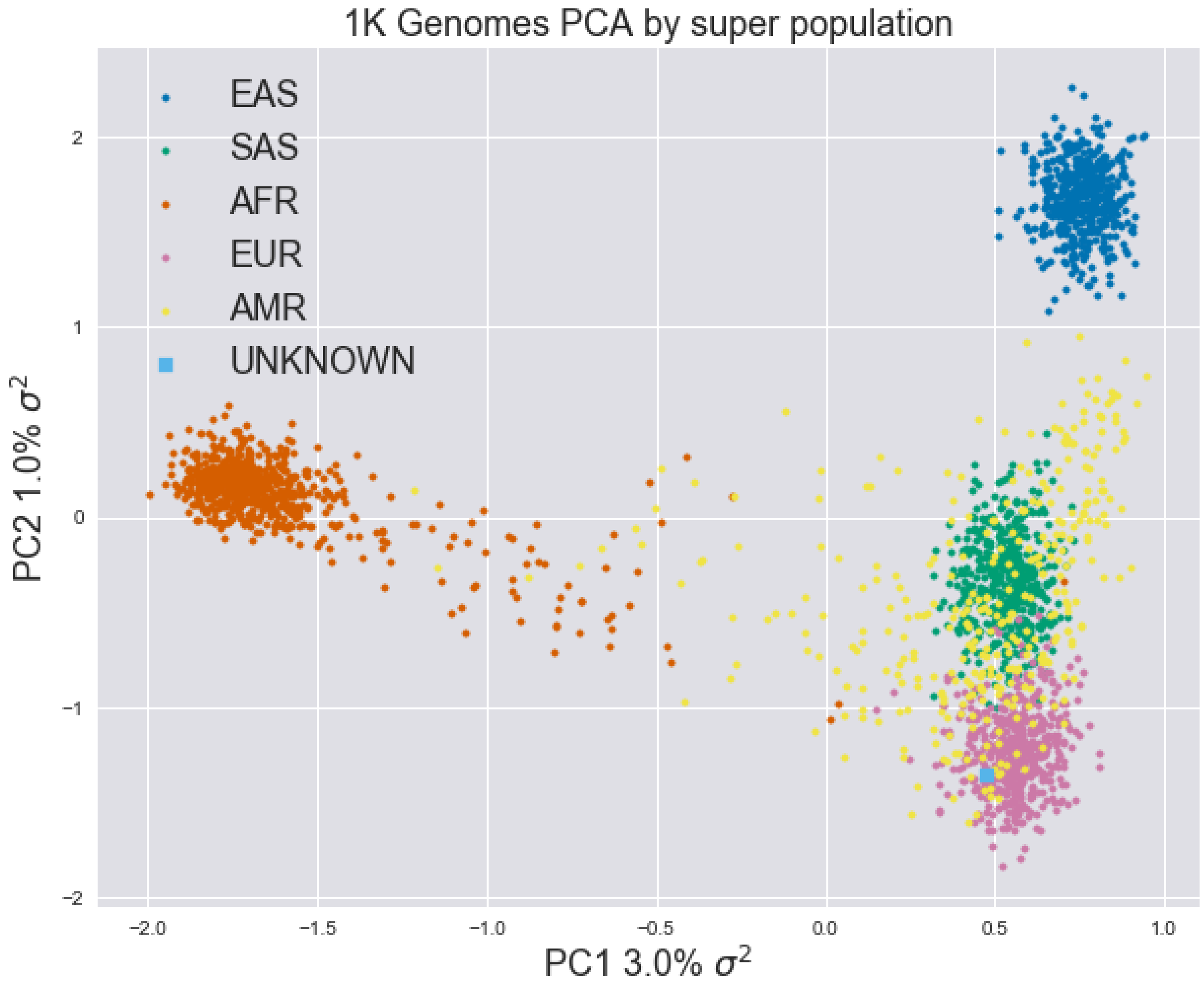
Out[5]:

	Description	Population Code
0	East Asian	EAS
1	South Asian	SAS
2	African	AFR
3	European	EUR
4	American	AMR
5	UNKNOWN	UNKNOWN

Label PCA plot by super population

In [6]:

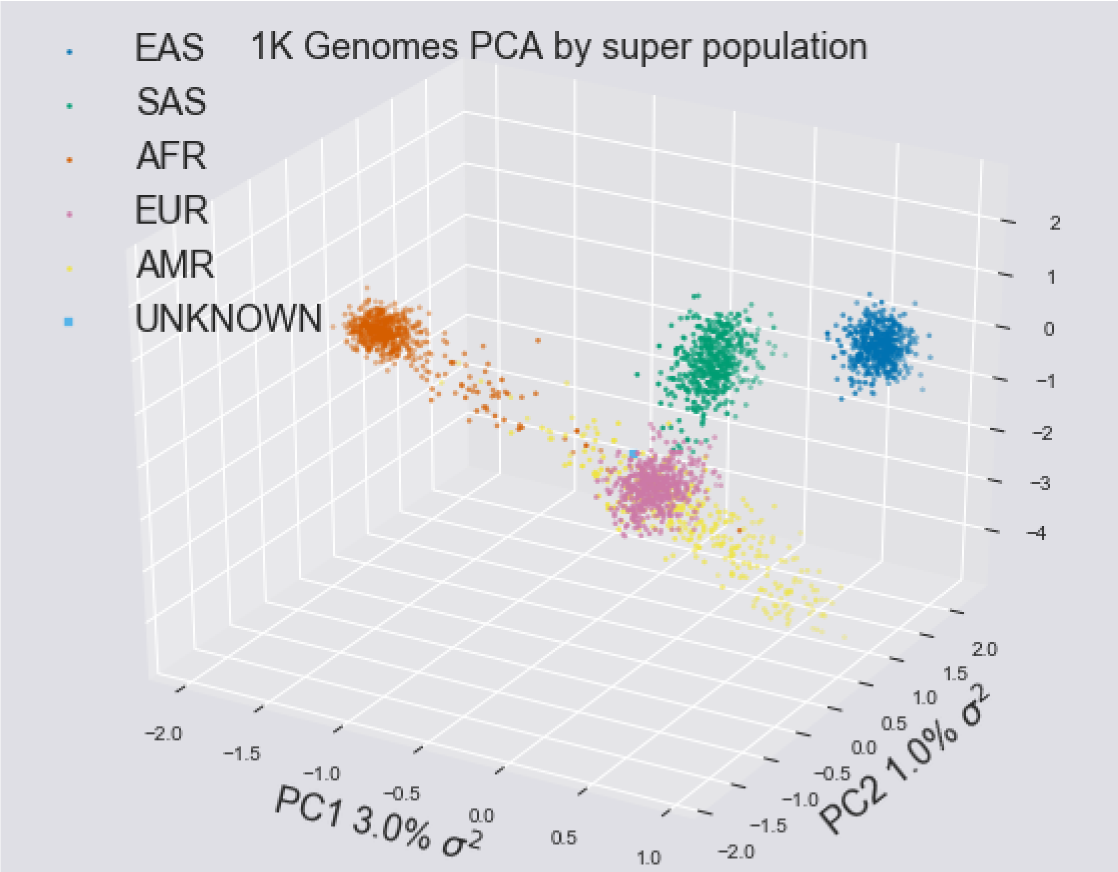
```
plt.figure(figsize=(10,8))
plt.title('1K Genomes PCA by super population')
plt.xlabel(f'PC1 {np.round(PC_load[0],2)*100}% '+r'$\sigma^2$')
plt.ylabel(f'PC2 {np.round(PC_load[1],2)*100}% '+r'$\sigma^2$')
for pop in superpop_key['Population Code']:
    PCs = df.loc[df['Super_Population']==pop].iloc[:,1:3]
    if pop == 'UNKNOWN':
        marker = 's'
    else:
        marker = '.'
    plt.scatter(PCs['PC1_0.03008'],PCs['PC2_0.01143'],label=pop,marker=marke
plt.legend(fontsize=18)
plt.show()
```



Now we're getting somewhere! Still, we can use another dimension to better view our unknown.

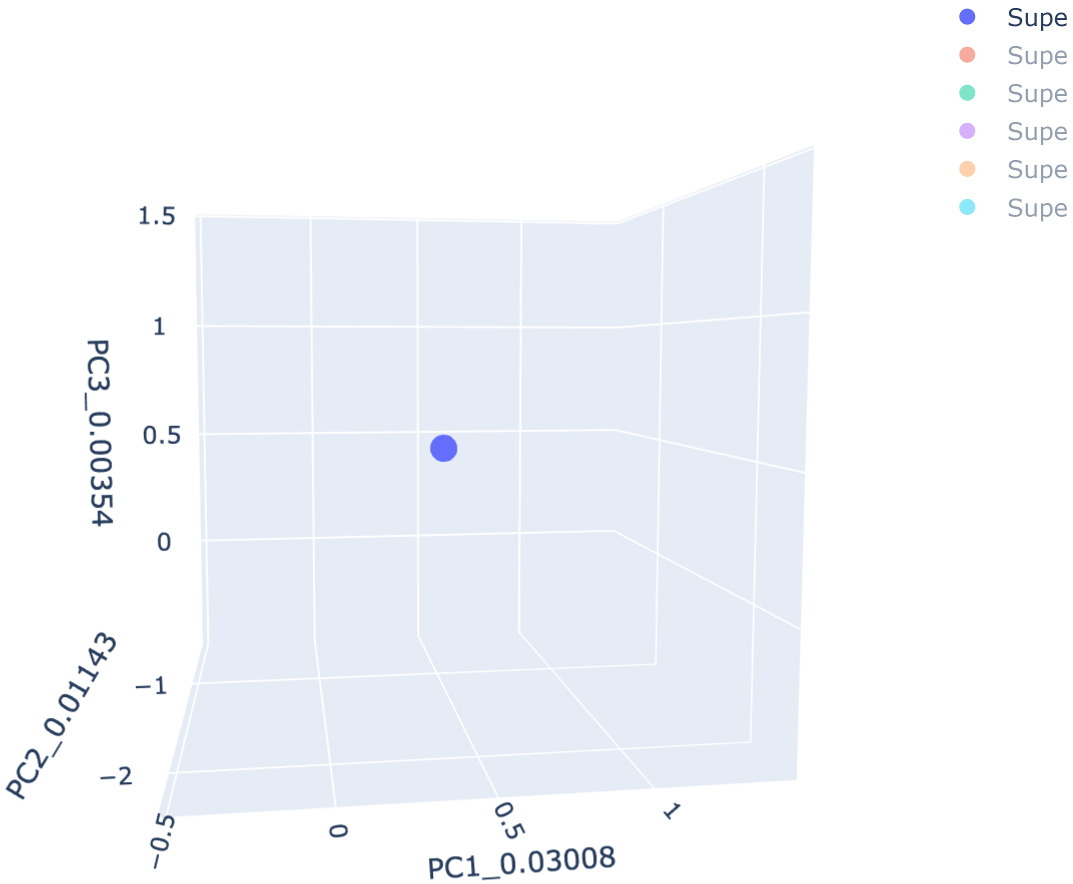
In [7]:
```
fig = plt.figure(figsize=(10,8))
ax = plt.axes(projection='3d')
ax.set_title('1K Genomes PCA by super population')
ax.set_xlabel(f'PC1 {np.round(PC_load[0],2)*100}% '+r'$\sigma^2$',labelpad=1
ax.set_ylabel(f'PC2 {np.round(PC_load[1],2)*100}% '+r'$\sigma^2$',labelpad=1
for pop in superpop_key['Population Code']:
    PCs = df.loc[df['Super_Population']==pop].iloc[:,1:4]
    if pop == 'UNKNOWN':
        marker = 's'
    else:
        marker = '.'
    ax.scatter(PCs['PC1_0.03008'],PCs['PC2_0.01143'],PCs['PC3_0.00354'],labe
ax.legend(fontsize=18,loc='upper left')
```

Out[7]:
```
<matplotlib.legend.Legend at 0x11271ff60>
```



If only we could rotate our plot around to see different angles...

In [8]:
```
fig = px.scatter_3d(df, x='PC1_0.03008',
                        y='PC2_0.01143',
                        z='PC3_0.00354',
                     color='Super_Population')
fig.show()
```



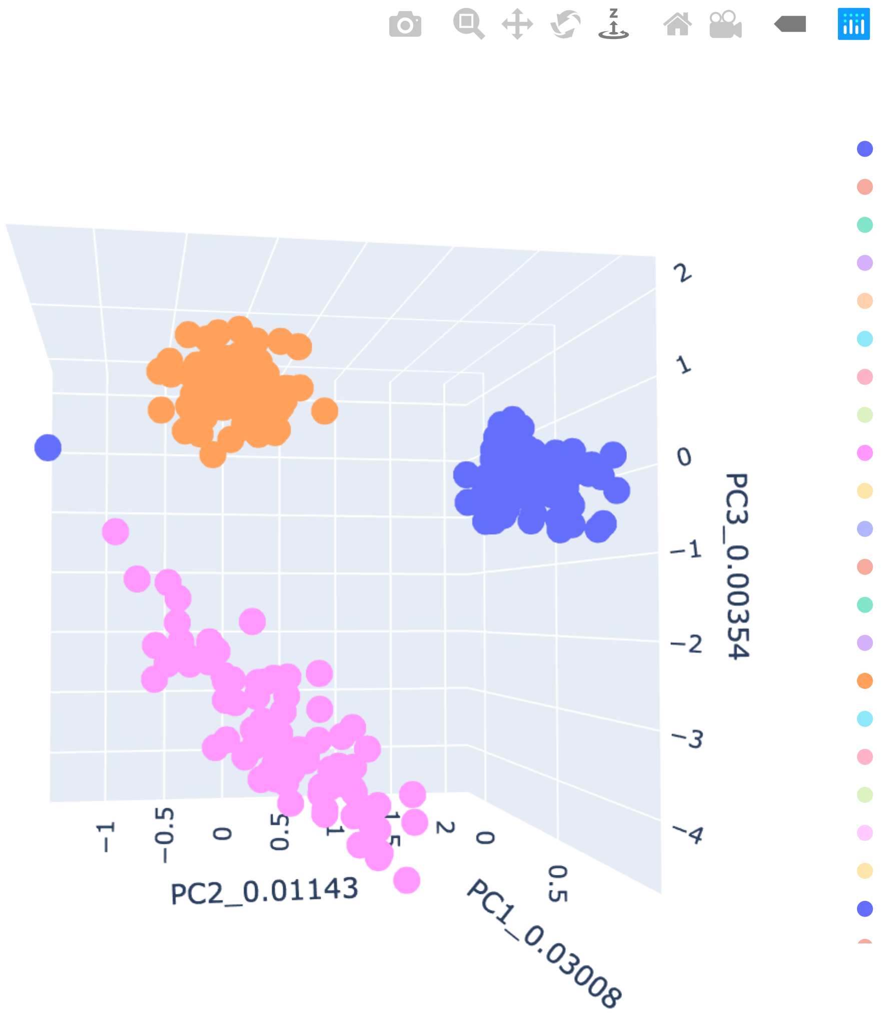
Try clicking other populations on the legend to hide them from view!

Label PCA plot by sub population

In [9]:
```
#### Key describing sub population codes ####
pop_key
```

17	British in England and Scotland	GBR	EUR
18	Finnish in Finland	FIN	EUR
19	Iberian populations in Spain	IBS	EUR
20	Toscani in Italy	TSI	EUR
21	Utah residents with Northern and Western Europ...	CEU	EUR
22	Colombian in Medellin, Colombia	CLM	AMR
23	Mexican Ancestry in Los Angeles, California	MXL	AMR
24	Peruvian in Lima, Peru	PEL	AMR
25	Puerto Rican in Puerto Rico	PUR	AMR
26	Total	NaN	NaN
27	NaN	NaN	NaN
28	NaN	NaN	NaN

In [10]:
```
fig = px.scatter_3d(df, x='PC1_0.03008',
                        y='PC2_0.01143',
                        z='PC3_0.00354',
                       color='Population')
fig.show()
```



Try hiding various combinations of subpopulations to identify which of these cluster most closely to our unknown.

Where is our unknown individual from?

Your answer here