Admixture 2

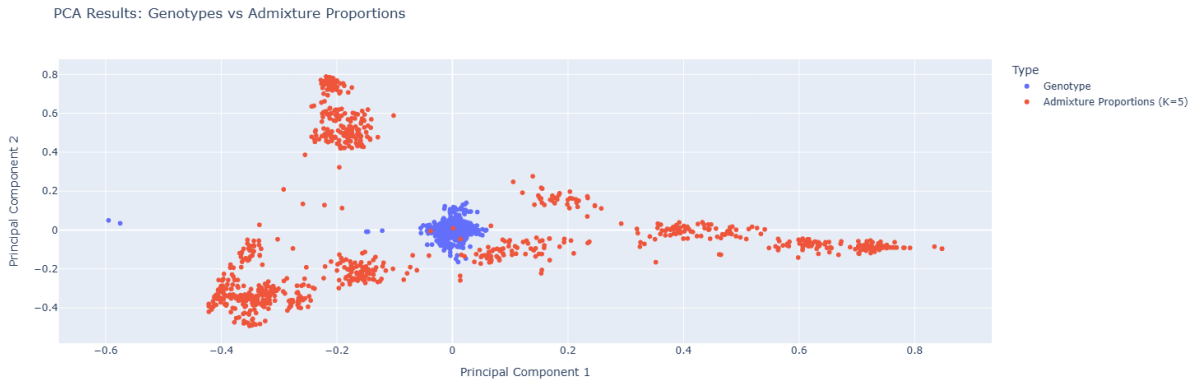
Используя результаты прошлого домашнего задания Admixture, .ipynb прошлого домашнего задания

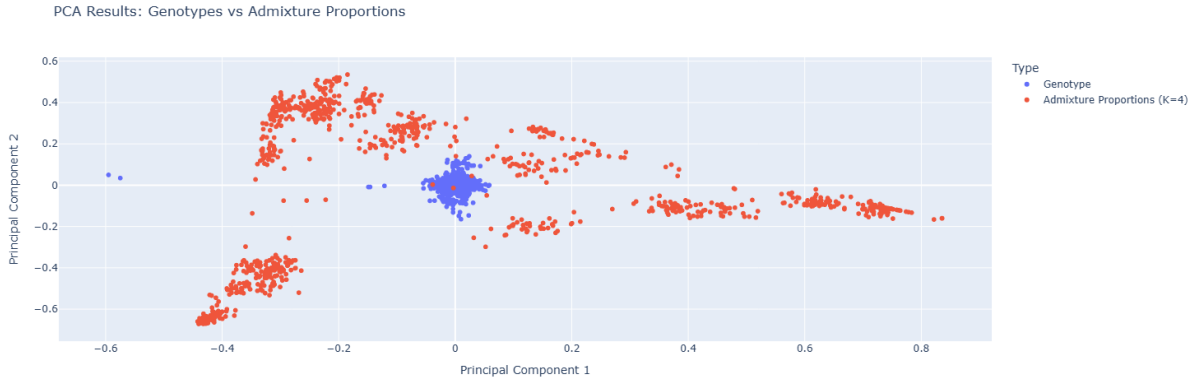
PCA по генотипам vs PCA по пропорциям Admixture

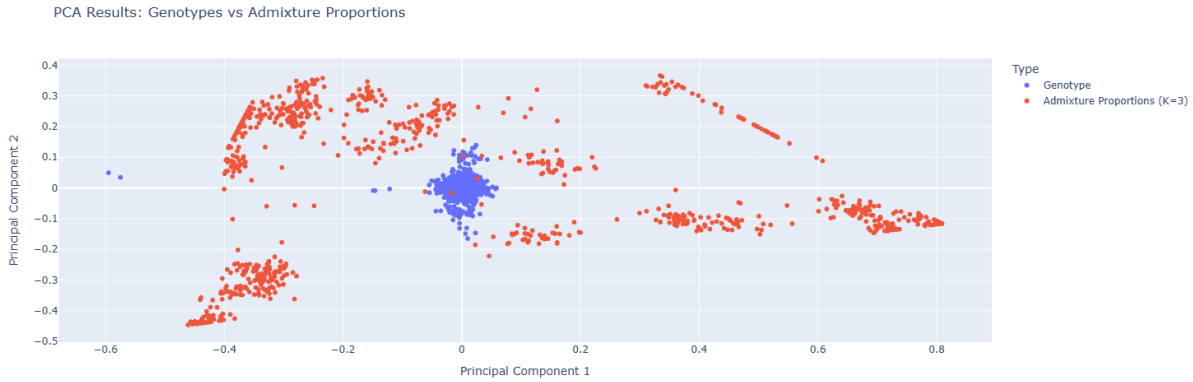
Code

```
import pandas as pd
from sklearn.decomposition import PCA
import plotly.express as px

pca_data = pd.read_csv('biengi_pca.eigenvec', sep='\s+', header=None)
pca_data.columns = ['FID', 'IID'] + [f'PC{i}' for i in range(1, 21)]
genotype_pca = PCA(n_components=2)
genotype_pca_result = genotype_pca.fit_transform(pca_data.iloc[:, 2:22])

K_values = [3, 4, 5]
for K in K_values:
    q_file = f"biengi.{K}.Q"
    admixture_data = pd.read_csv(q_file, sep='\s+', header=None)
    admixture_pca = PCA(n_components=2)
    admixture_pca_result = admixture_pca.fit_transform(admixture_data)

    # DataFrame for genotype
    genotype_df = pd.DataFrame(genotype_pca_result, columns=['PC1', 'PC2'])
    genotype_df['Type'] = 'Genotype'

    # DataFrame for admixture
    admixture_df = pd.DataFrame(admixture_pca_result, columns=['PC1', 'PC2'])
    admixture_df['Type'] = f'Admixture Proportions (K={K})'

    combined_df = pd.concat([genotype_df, admixture_df], ignore_index=True)

    fig = px.scatter(combined_df, x='PC1', y='PC2', color='Type',
                     title='PCA Results: Genotypes vs Admixture Proportions',
                    labels={'PC1': 'Principal Component 1', 'PC2': 'Principal Component 2'},
                     hover_data=['Type'])

    fig.show()
```

Получим (для разных K):







Теперь сравним UMAP и tSNE
Для этого установим umap и перезапустим сессию

Code

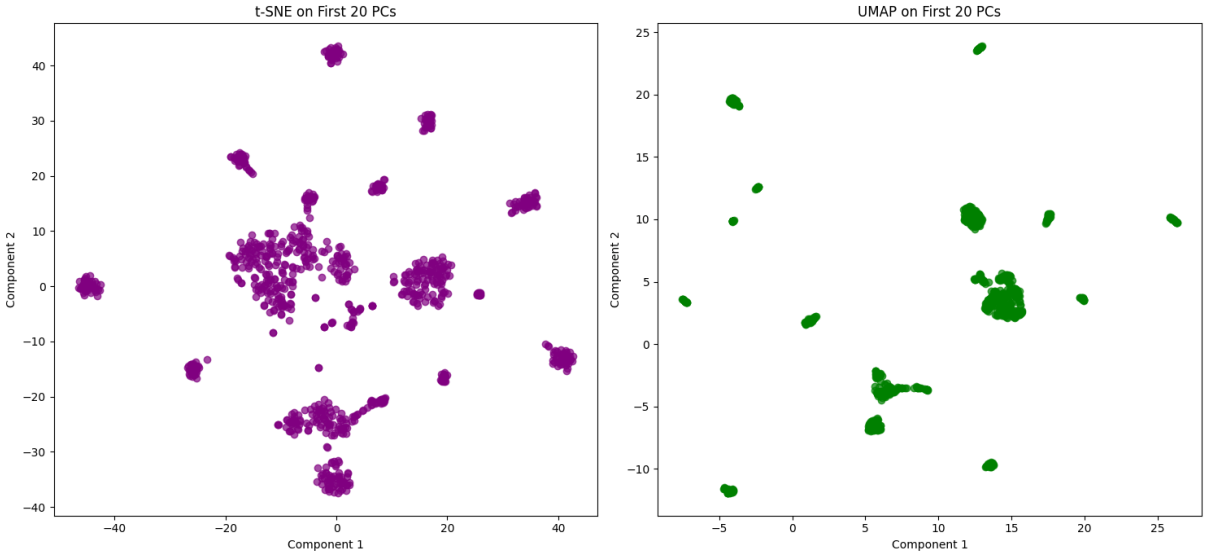
```
!pip install umap-learn
```

Теперь, установив umap, выполним следующее:

Code

```
import pandas as pd
from sklearn.decomposition import PCA
from sklearn.manifold import TSNE
from umap import UMAP
import matplotlib.pyplot as plt

pca_data = pd.read_csv('biengi_pca.eigenvec', sep='\s+', header=None)
pca_data.columns = ['FID', 'IID'] + [f'PC{i}' for i in range(1, 21)]

# first 20 PCs
pca_20_data = pca_data.iloc[:, 2:22]

# t-SNE
tsne = TSNE(n_components=2, random_state=42)
tsne_result = tsne.fit_transform(pca_20_data)

# UMAP
umap = UMAP(n_components=2, random_state=42)
umap_result = umap.fit_transform(pca_20_data)

fig, axes = plt.subplots(1, 2, figsize=(15, 7))

# t-SNE
axes[0].scatter(tsne_result[:, 0], tsne_result[:, 1], alpha=0.7, c='purple')
axes[0].set_title('t-SNE on First 20 PCs')
axes[0].set_xlabel('Component 1')
axes[0].set_ylabel('Component 2')

# UMAP
axes[1].scatter(umap_result[:, 0], umap_result[:, 1], alpha=0.7, c='green')
axes[1].set_title('UMAP on First 20 PCs')
axes[1].set_xlabel('Component 1')
axes[1].set_ylabel('Component 2')

plt.tight_layout()
plt.show()
```

Получим:



Также весь код можно посмотреть тут