

Фармакогенетика

- Установите [фармкат](#) локально или на сервере сами (не используя докер). Подсказка: тул `pharmcat` имеет расширение `.jar`
- Запустите `pharmcat` на данных [pharmcat.example.vcf](#) по tutorialу приведенному сверху
- Получите отчет Reporter в формате `.html` Задания:
- Напишите полученные генотипы для генов `VKORC1` и `CYP2C9`
- Напишите фенотип метаболизатора варфарина
- Какая доза варфарина рекомендована гайдлайном CPIC для этого пациента?

Загрузим файлы

Code

```
# Install necessary packages
!pip install gdown plotly pandas

# Download the files using gdown
import gdown

files = {
    '12DyJHUh2RjduOolcW6L-GRTC3BQYRpZ': 'biengi.bed',
    '12NS-FTtRFM32j1POjFEe0AH7i6Ze_RRI': 'biengi.bim',
    '12HmHDS0AslpQZ-REJnqLS4odML_-rfEu': 'biengi.pedind',
    '12DbtxNZCEvdOzyxHVyrVKRLZMFKNh97f': 'biengi.fam'
}

for file_id, file_name in files.items():
    gdown.download(f'https://drive.google.com/uc?id={file_id}', file_name, quiet=False)
```

Установим plink

Code

```
!wget https://s3.amazonaws.com/plink1-assets/plink_linux_x86_64_20231211.zip
!unzip plink_linux_x86_64_20231211.zip
!chmod +x plink
!mv plink /usr/local/bin/
!plink --version
```

запустим Plink

Code

```
# Run PLINK PCA analysis
!plink --bfile biengi --pca --out biengi_pca
```

Получим PCA

Code

```
import pandas as pd

# Load PCA eigenvectors
pca_data = pd.read_csv('biengi_pca.eigenvec', delim_whitespace=True, header=None)
pca_data.columns = ['FID', 'IID'] + [f'PC{i}'] for i in range(1, pca_data.shape[1] - 1)

# Display the first few rows
pca_data.head()
```

получим:

```
<ipython-input-11-ed508b78036e>4: FutureWarning: The 'delim_whitespace' keyword in pd.read_csv is deprecated and will be removed in a future version. Use 'sep='\s+' instead
pca_data = pd.read_csv('biengi_pca.eigenvec', delim_whitespace=True, header=None)

   FID      IID      PC1      PC2      PC3      PC4      PC5      PC6      PC7      PC8      ...      PC11      PC12      PC13      PC14      PC15      PC16      PC17      PC18      PC19      PC20
0  4461875510_R01C01  4461875510_R01C01  0.022805  0.046644  0.025682 -0.009200 -0.003159 -0.009250  0.004704 -0.003999  ...  0.031829 -0.024709 -0.043020  0.049520  0.081407  0.054900  0.031768 -0.037673 -0.043701  0.016445
1  4461875510_R01C02  4461875510_R01C02  0.029046  0.031433  0.032579 -0.015769 -0.002586 -0.024625  0.003845 -0.007098  ... -0.021524  0.001733 -0.025462 -0.005282  0.023335  0.009862 -0.035812  0.000900  0.003050  0.012582
2  4461875510_R02C01  4461875510_R02C01  0.025890  0.043415  0.024259 -0.010237 -0.005302 -0.018654  0.014151 -0.010939  ...  0.042643 -0.042631 -0.014694  0.050725  0.074761  0.093167  0.024564  0.008311 -0.047672  0.038053
3  4461875513_R02C02  4461875513_R02C02  0.022355  0.045329  0.024989 -0.014831 -0.008035  0.001440  0.027770 -0.023837  ...  0.038537 -0.021936  0.001573  0.052039  0.107735  0.090628  0.001787 -0.048874 -0.061143 -0.003882
4  4461875513_R01C02  4461875513_R01C02  0.024226  0.043572  0.015883 -0.009853 -0.004136 -0.011679  0.010524 -0.010580  ...  0.044584 -0.037172 -0.037817  0.047730  0.075309  0.111941 -0.006808 -0.028436 -0.023186  0.053598
5 rows x 22 columns
```

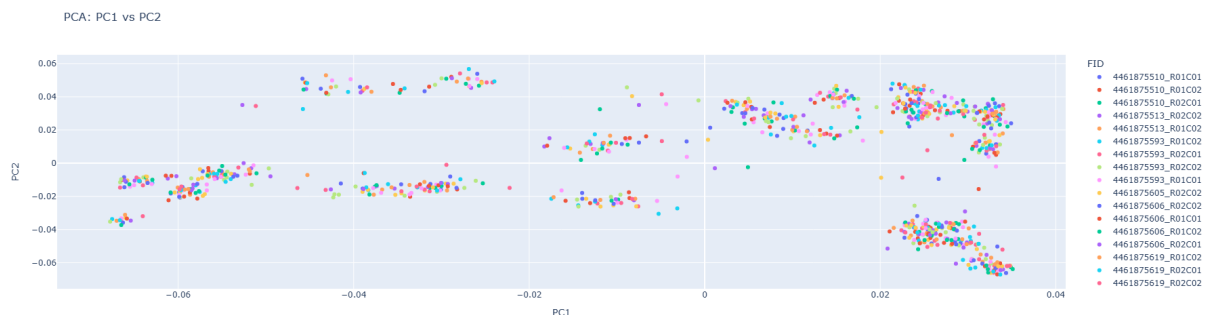
Code

```
import plotly.express as px

# Create an interactive scatter plot
fig = px.scatter(
    pca_data,
    x='PC1',
    y='PC2',
    color='FID',
    hover_data=['IID'],
    title='PCA: PC1 vs PC2'
)

# Show the interactive plot
fig.show()
```

получим:



Также весь код можно посмотреть [тут](#)