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 туториалу приведенному сверху
- Получите отчет Reporter в фомате .html Задания:
- Напишите полученные генотипы для генов VKORC1 и CYP2C9
- Напишите фенотип метаболизатора варфарина
- Какая доза варфарина рекомендована гайдлайном CPIC для этого пациента?

Установим все:

Code

```
# Colab code for running PharmCAT v2.15.4 on PharmCAT example data

# Step 1: Install Java 17 (required for PharmCAT v2.15.4)
!apt-get update -y
!apt-get install -y openjdk-17-jdk

# Step 2: Download the PharmCAT example VCF file from Google Drive
# Required: Install gdown to download files from Google Drive
!pip install gdown
!gdown "https://drive.google.com/uc?id=12A_R_R1393ckeo78fwq_JTiiIiZKh6MI"
    -O pharmcat_example.vcf

# Step 3: Download the PharmCAT v2.15.4 release files
# Download the main PharmCAT JAR file
!wget https://github.com/PharmGKB/PharmCAT/releases/download/
    v2.15.4/pharmcat-2.15.4-all.jar

# Step 4: Download and extract the VCF Preprocessor
!wget https://github.com/PharmGKB/PharmCAT/releases/download/v2.15.4/
    /pharmcat-preprocessor-2.15.4.tar.gz
!tar -xvf pharmcat-preprocessor-2.15.4.tar.gz

# Move files from ’preprocessor’ folder to current directory
!cp -r preprocessor/* ./
```

Запустим на наших данных pharmcat.example.vcf:

Code

```
# Step 5: Run the matcher stage with the example VCF file
!java -jar pharmcat-2.15.4-all.jar
    --matcher-vcf pharmcat_example.vcf
    -o ./matcher_output

# Step 6: Run the phenotyper stage using the matcher output as input
# Specify the correct input file from the matcher output folder
!java -jar pharmcat-2.15.4-all.jar
    --phenotyper
    --phenotyper-input ./matcher_output/pharmcat_example.match.json
    -o ./phenotyper_output

# Step 7: Run the reporter stage using the phenotyper output as input
!java -jar pharmcat-2.15.4-all.jar
    --reporter
    --reporter-input ./phenotyper_output/pharmcat_example.phenotype.json
    -o ./report_output

# Step 8: Check the contents of the report output folder
!ls ./report_output

# Step 9: Display the path of the generated report
from IPython.display import display, HTML
display(HTML("PharmCAT report has been generated as
’report_output/pharmcat_example.report.html’."))
```

Репорт можно посмотреть следующим образом или по ссылке:

Code

```
with open("/content/report_output/pharmcat_example.report.html", "r") as file:
    display(HTML(file.read()))
```

Чтобы ответить на последние 3 вопроса

- Напишите полученные генотипы для генов VKORC1 и CYP2C9
- Напишите фенотип метаболизатора варфарина
- Какая доза варфарина рекомендована гайдлайном CPIC для этого пациента?

воспользуемся следующим кодом:

Code

```
import json

# Load the JSON data
with open(’/content/matcher_output/pharmcat_example.match.json’, ’r’) as file:
    data = json.load(file)

# Function to extract genotypes for VKORC1 and CYP2C9
def extract_genotypes(data):
    genotypes = {}
    for result in data[’results’]:
        gene = result[’gene’]
        if gene in [’VKORC1’, ’CYP2C9’]:
            diplotype = result[’diplotypes’][0][’name’]
            genotypes[gene] = diplotype
    return genotypes

# Function to determine phenotype based on genotypes
def determine_phenotype(genotypes):
    vkorc1_genotype = genotypes.get(’VKORC1’, ’’)
    cyp2c9_genotype = genotypes.get(’CYP2C9’, ’’)

    # VKORC1 sensitivity
    is_sensitive = ’A’ in vkorc1_genotype
    # Reduced activity alleles
   is_cyp2c9_reduced_activity = any(allele in cyp2c9_genotype for allele in [’*2’, ’*3’])

    if is_sensitive or is_cyp2c9_reduced_activity:
        return "Warfarin Sensitive Metabolizer"
    else:
        return "Warfarin Normal Metabolizer"

# Function to recommend warfarin dose based on phenotype
def recommend_warfarin_dose(phenotype):
    if phenotype == "Warfarin Sensitive Metabolizer":
        return "Lower dose recommended (e.g., 1-3 mg)"
    elif phenotype == "Warfarin Normal Metabolizer":
        return "Standard dose recommended (e.g., 5-7 mg)"
    return "Consult CPIC guidelines for specific dosing"
```

Code

```
# Execute functions
genotypes = extract_genotypes(data)
phenotype = determine_phenotype(genotypes)
warfarin_dose = recommend_warfarin_dose(phenotype)

# Output results
print("Genotypes:")
print(f"VKORC1: {genotypes.get('VKORC1', 'Not Found')}")
print(f"CYP2C9: {genotypes.get('CYP2C9', 'Not Found')}")

print("\nPhenotype of Warfarin Metabolizer:")
print(phenotype)

print("\nRecommended Warfarin Dose:")
print(warfarin_dose)
```

получим:

Code

```
Genotypes:
VKORC1: rs9923231 reference (C)/rs9923231 reference (C)
CYP2C9: *1/*1

Phenotype of Warfarin Metabolizer:
Warfarin Normal Metabolizer

Recommended Warfarin Dose:
Standard dose recommended (e.g., 5-7 mg)
```

весь output также можно посмотреть тут