.

**Exercício de Mapeamento de Admistura (ADM) seguindo o artigo de Mandla et al. (2025)**

**1. rfmix.v2**

https://github.com/slowkoni/rfmix/blob/master/README.md

**- Instalando**

cd /mnt/disk2t/bioinfo\_labs/tools/rfmix-master

sudo apt-get update

sudo apt-get install automake autoconf libtool build-essential

sudo apt-get install libgsl-dev libhts-dev

autoreconf –install

./configure

make

cd ..

mv /mnt/disk2t/bioinfo\_labs/tools/rfmix-master /mnt/disk2t/bioinfo\_labs/tools/rfmix\_v2

**- Acrescentar caminho de busca**

vim ~/.basrc

i

PATH="/mnt/disk2t/bioinfo\_labs/tools/rfmix\_v2:$PATH"

esc

:wq

source ~/.bashrc

#end

**2. Mini-projeto ADM com dados simulados**

Como não temos acesso aos dados reais do All of Us (que são restritos), este projeto irá:

- Simular o conjunto de dados: fenótipos, covariáveis e, dados de ancestralidade local.

- Utilizar admix-kit para executar a análise de ADM.

- Visualizar os resultados em um gráfico de Manhattan.

**Passo 1: Pré-requisitos.** Instalar sowftare e bibliotecas

$ pip install admix-kit

$ pip install pandas numpy matplotlib

**Passo 2: Simulação dos Dados de Entrada**

- Criar um cenário onde a ancestralidade africana (AFR) em uma região específica do cromossomo 1 aumenta o valor de um traço quantitativo (QT). Arquivo “simular\_dados.py”:

import numpy as np

import pandas as pd

# --- 1. Parâmetros da Simulação ---

N\_INDIVIDUOS = 500

N\_JANELAS = 2000

CHR = 1

POSICAO\_CAUSAL\_INICIO = 1000

POSICAO\_CAUSAL\_FIM = 1010

EFEITO\_BETA = 0.8

# --- 2. Gerar Arquivo de Ancestralidade Local (.msp) ---

# Formato do RFMix: cada linha é uma janela, colunas são haplótipos

# 0 = EUR (ancestralidade de referência), 1 = AFR

# Vamos simular que a ancestralidade AFR é, em média, 25%

msp\_dados = np.random.choice([0, 1], size=(N\_JANELAS, N\_INDIVIDUOS \* 2), p=[0.75, 0.25])

# Inserir o sinal causal: indivíduos com ancestralidade AFR nesta região terão o efeito

# Contamos o número de haplótipos AFR (valor 1) na região causal para cada indivíduo

ancestralidade\_causal\_por\_individuo = []

for i in range(N\_INDIVIDUOS):

haplo1 = msp\_dados[POSICAO\_CAUSAL\_INICIO:POSICAO\_CAUSAL\_FIM, i\*2].mean()

haplo2 = msp\_dados[POSICAO\_CAUSAL\_INICIO:POSICAO\_CAUSAL\_FIM, i\*2+1].mean()

# Se a média de ancestralidade AFR for > 0.5 na região, consideramos que o indivíduo a possui

n\_haplos\_afr = (haplo1 > 0.5) + (haplo2 > 0.5)

ancestralidade\_causal\_por\_individuo.append(n\_haplos\_afr)

with open("admixed.msp.tsv", "w") as f:

f.write(f"chr{CHR}\t" + "\t".join([f"I{i}" for i in range(N\_INDIVIDUOS \* 2)]) + "\n")

for linha in msp\_dados:

f.write("\t".join(map(str, linha)) + "\n")

print("Arquivo .msp gerado: admixed.msp.tsv")

# --- 3. Gerar Arquivo de Mapa Genético (.gmap) ---

# Posições físicas e genéticas das janelas

posicoes\_fisicas = np.linspace(1\_000\_000, 2\_000\_000, N\_JANELAS, dtype=int)

posicoes\_geneticas = np.linspace(0, 100, N\_JANELAS)

gmap\_df = pd.DataFrame({

'CHR': CHR,

'VAR\_ID': [f'snp\_{i}' for i in range(N\_JANELAS)],

'CM': posicoes\_geneticas,

'POS': posicoes\_fisicas

})

gmap\_df.to\_csv("chr1.gmap", sep="\t", index=False, header=False)

print("Arquivo .gmap gerado: chr1.gmap")

# --- 4. Gerar Arquivo de Fenótipo e Covariáveis ---

# IDs dos indivíduos

individuos\_ids = [f"I{i}" for i in range(N\_INDIVIDUOS)]

# Covariáveis (simulando idade, sexo e 2 PCs, como no estudo)

covar\_df = pd.DataFrame({

'IID': individuos\_ids,

'AGE': np.random.randint(20, 70, N\_INDIVIDUOS),

'SEX': np.random.choice([0, 1], N\_INDIVIDUOS),

'PC1': np.random.normal(0, 1, N\_INDIVIDUOS),

'PC2': np.random.normal(0, 1, N\_INDIVIDUOS)

})

covar\_df.to\_csv("covariates.txt", sep="\t", index=False)

print("Arquivo de covariáveis gerado: covariates.txt")

# Fenótipo (QT = ruído + efeito da ancestralidade local causal)

ruido = np.random.normal(0, 1, N\_INDIVIDUOS)

fenotipo\_qt = ruido + (np.array(ancestralidade\_causal\_por\_individuo) \* EFEITO\_BETA)

pheno\_df = pd.DataFrame({

'IID': individuos\_ids,

'QT': fenotipo\_qt

})

pheno\_df.to\_csv("phenotype.txt", sep="\t", index=False)

print("Arquivo de fenótipo gerado: phenotype.txt")

print("\nSimulação concluída!")

**Execute** este script para gerar seus dados de teste:

$ python simular\_dados.py

**Passo 2: Execução da Análise de ADM com admix-kit**

Agora, com os arquivos gerados, vamos executar o mapeamento de admistura. O comando abaixo informa ao admix-kit para realizar um teste de associação (adm) entre a ancestralidade local e o fenótipo QT, ajustando pelas covariáveis.

Execute o seguinte comando no seu terminal:

$ admix-kit adm \

--msp admixed.msp.tsv \

--gmap chr1.gmap \

--pheno phenotype.txt \

--covar covariates.txt \

--trait QT \

--out adm\_results

--msp: Arquivo com a inferência de ancestralidade local.

--gmap: Mapa genético do cromossomo.

--pheno: Arquivo de fenótipo.

--covar: Arquivo de covariáveis para ajuste do modelo.

--trait: Nome da coluna do fenótipo a ser testada.

--out: Prefixo para os arquivos de saída.

O programa irá gerar um arquivo chamado adm\_results.adm, que contém os resultados da associação.

**Passo 3: Visualização dos Resultados**

Para interpretar os resultados, vamos criar um gráfico de Manhattan que mostra a força da associação (-log10 do P-valor) ao longo do cromossomo.

Crie um arquivo chamado plotar\_manhattan.py e cole o código:

import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

# Carregar os resultados do admix-kit

try:

results\_df = pd.read\_csv("adm\_results.adm", sep="\t")

except FileNotFoundError:

print("Erro: Arquivo 'adm\_results.adm' não encontrado. Execute o passo 2 primeiro.")

exit()

# Calcular -log10(P)

results\_df['-log10P'] = -np.log10(results\_df['P'])

# Posição da nossa associação simulada para destaque

pos\_causal\_fisica = int(np.linspace(1\_000\_000, 2\_000\_000, 2000)[1000])

# Criar o gráfico

plt.figure(figsize=(12, 6))

plt.scatter(results\_df['POS'], results\_df['-log10P'], s=10, alpha=0.7)

# Adicionar linha de significância (ex: Bonferroni)

plt.axhline(y=-np.log10(0.05 / len(results\_df)), color='r', linestyle='--', label='Significância de Bonferroni')

# Destacar a região causal

plt.axvline(x=pos\_causal\_fisica, color='g', linestyle=':', label='Região Causal Simulada')

plt.title('Gráfico de Manhattan - Mapeamento de Admistura Simulado')

plt.xlabel('Posição no Cromossomo 1')

plt.ylabel('-log10(P-valor)')

plt.legend()

plt.tight\_layout()

plt.savefig("manhattan\_plot.png")

plt.show()

print("\nGráfico de Manhattan salvo como 'manhattan\_plot.png'")

**Execute** o script de plotagem:

$ python plotar\_manhattan.py

**Resultado Esperado**

Você verá um gráfico de Manhattan com um pico claro de associação na região que simulamos como "causal". O P-valor nessa região será extremamente baixo (e, portanto, o valor de -log10(P) será alto), indicando que o método de ADM detectou com sucesso a associação entre a ancestralidade local AFR e o aumento do fenótipo.

Este mini-projeto demonstra de forma prática e simplificada o fluxo de trabalho central do estudo de Mandla et al., permitindo uma compreensão tangível de como o mapeamento de admistura é aplicado para encontrar loci associados a traços.