Index

普段どのような環境で分析されているのかわからないので、terminal と bcftools を使う場合のコードを記載しています。jupyter notebook でも実行可能です。

※ terminal で実行した場合、グラフ画像が生成された段階で実行が一時停止する可能性があります。その場合は画像（保存可能）を閉じると実行が再開されます。jupyter notebook 上で実行した場合はそのようなことは起きません。

※ 各実行には少々時間がかかります。30 分以内におさまることを目的に作られています。

1. ファイルについて

chrY_package.py … レポート等を出力する関数が書かれているファイル。

snps_hg38.csv.zip … 圧縮された ISOGG Ybrowse のマスターcsv ファイル。

2. プログラム実行手順（terminal 内でも実行可）

① Python 実行環境の構築、必要ライブラリのインストール（python, pip はインストール済と仮定）

（ア）pandas, requests, matplotlib.pyplot, matplotlib.image, openai の 5 つのパッケージのインストール

（例）terminal 内において pip install pandas

（イ）chrY_package.py、snps_hg38.csv（解凍後）を含むファイルで入れクトリに移動

（例）terminal 内において cd ディレクトリパス名

（ウ）python をスタート

（例）terminal 内において python

② レポートを出力する関数のインポート

（例）terminal 内において from chrY_package import get_result, get_gpt_output

③ レポートの取得

（例 1）result = get_result(“インプットファイルのディレクトリ名”)

（例 2）messages = get_gpt_output(result, “共有 APIKEY の名前”) #こちらは例 1 の実行後にしか実行できません。

3. インプット vcf の形式について

CHROM POS ID REF ALT QUAL FILTER INFO FORMAT ~

の順にデータが並んでいてヘッダー（コラム名を含め、＃から始まる行）のない vcf ファイル（Y 染色体以外のデータを含んでいても実行可能)です。
（例）bcftools の view -o -H　メソッドで取得

4. プログラムの出力について（出力例）

result = get_result(“インプットファイルのディレクトリ名”)の実行結果：

実行して間も無く分析ファイルをアップデートするかどうかを聞かれますが、時間がかかるのであまりお勧めしません**（n とタイプしていただければスキップされます)**。ただ、Y ハプログループのデータベースは高頻度でアップデートされているようです。

十数分後、画像が四つ生成されます。

画像１：**ISOGG** による分類に基づいた各ハプログループに対する**変異の一致数の棒グラフ**（Y 軸：一致する変異の数、X 軸：ハプログループの名称）unknown, not listed は分類に使われていない変異を示しています。

画像２： **Y Chromosome Consortium (YCC)** による分類に基づいた各ハプログループに対する**変異の一致数の棒グラフ**（Y 軸：一致する変異の数、X 軸：ハプログループの名称）unknown, not listed は分類に使われていない変異を示しています。

画像３：**ISOGG** による分類に基づいた各ハプログループに対する**変異の一致数の棒グラフ**（Y 軸：一致する変異の数、X 軸：ハプログループの名称）unknown, not listed は分類に使われていない変異を示しています。画像１との違いは、**複数の名称を持つ変異の中から１つしか利用していない**点です。同一の変異を示していながら別の名称のハプログループに分類されている場合がありましたので、画像１と画像３をどちらも表示することにしました。

画像４： **Y Chromosome Consortium (YCC)** による分類に基づいた各ハプログループに対する**変異の一致数の棒グラフ**（Y 軸：一致する変異の数、X 軸：ハプログループの名称）unknown, not listed は分類に使われていない変異を示しています。画像２との違いは、**複数の名称を持つ変異の中から１つしか利用していない**点です。同一の変異を示していながら別の名称のハプログループに分類されている場合がありましたので、画像２と画像４をどちらも表示することにしました。

messages = get_gpt_output(result, “共有 APIKEY の名前”)の実行結果

こちらは GPT モデルを利用して地図画像とハプログループの説明を出力します。

実行して間も無く推奨される**情報源の URL リスト**が出力されるので、GPT の出力の正確性や各情報源の違いなども確認できます。

リンクの次にテキストの塊（GPT から出力される生データ）が出力されますが、こちらはエラーが出た時のためのデバッグ用の出力なので**無視していただいて構いません**。

最後に画像と説明文が出力されます。
画像は地図のヒートマップとなっており、変異の一致数が多いハプログループのエリアがより濃い色で示されています。
説明文は画像の内容の説明となっております（カスタマイズ可)。