خطة تنفيذ المشروع لكورس مقدمة في الجينوميات الحاسوبية

بناءً على الملفات المرفقة، سأقدم خطة شاملة مقسمة إلى مراحل واضحة مع خطوات تنفيذ **Challenge_2025.pdf** لتنفيذ المشروع المطلوب في ملف كل مرحلة. كما سأشرح وظيفة كل ملف وما يحتويه وكيفية استخدامه، وأحدد المخرجات المطلوب تسليمها بدقة. الهدف هو تحقيق متطلبات المشروع باستخدام البرمجة وعلم الأحياء مع الاستفادة من جميع الملفات المقدمة.

وظيفة كل ملف ومحتواه وكيفية استخدامه

1. **extract_sequences.py**
   - **الوظيفة**: يحتوي على دالة Python تُسمى `extract_flanked_region` لاستخراج تسلسلات جينية من تسلسل كروموسومي بناءً على مواقع البداية والنهاية مع إضافة مناطق مجاورة (upstream وdownstream) مع مراعاة اتجاه السلسلة (strand).
   - **المحتوى**: كود Python 1) يأخذ معطيات مثل التسلسل الكروموسومي، الاتجاه (+ أو -)، مواقع البداية والنهاية (1-based)، وعدد القواعد المجاورة، ويُرجع التسلسل المستخرج.
   - **كيفية الاستخدام**: سأستخدم هذه الدالة لاستخراج التسلسلات الجينية من ملف **100_genes.csv** بناءً على بيانات الطفرات في ملفات **train_muts_data.csv**، و**test_muts_data.csv** مع إضافة مناطق مجاورة (مثل 10 قواعد) لتحليل الطفرات.

2. **test_feats.csv**
   - **الوظيفة**: يحتوي على بيانات ميزات (features) لعينات الاختبار، وهي عدد الطفرات لكل جين لكل عينة.
   - **المحتوى**: جدول CSV يحتوي على أعمدة `case_id` (مثل 0، 1، 2 وأعمدة لـ 100 جين مع قيم عددية تمثل عدد الطفرات...).
   - **كيفية الاستخدام**: سأستخدمه كمدخل للتنبؤ بالتسميات (labels) باستخدام نموذج تعلم آلي مدرب على **train_feats.csv**.

3. **Challenge_2025.pdf**
   - **الوظيفة**: يحتوي على وصف المشروع والمتطلبات الأساسية.
   - **المحتوى**: OCR (النص بالعبرية والعربية مختلط وغير مفهوم)، لكن يمكن استنتاج أن تعليمات غير واضحة تمامًا بسبب مشاكل الـ المطلوب هو تحليل بيانات الطفرات وبناء نموذج للتنبؤ بالتسميات مع تقديم تقرير. يظهر معادلة لحساب الخطأ: `Error = sum(df_known_test['Label'] != df_known_test['predict_label']) / len(df_known_test)`.
   - **كيفية الاستخدام**: سأعتمد على هذا الملف لفهم الهدف العام (تصنيف العينات بناءً على الطفرات) وأستخدم المعادلة لتقييم النموذج.

4. **train_feats.csv**
   - **الوظيفة**: بيانات تدريب تحتوي على ميزات العينات مع تسمياتها (labels).
   - **المحتوى**: جدول CSV يحتوي على `case_id`، `Label` (قيم مثل 1.0 أو 2.0)، وأعمدة لـ 100 جين مع عدد الطفرات.
   - **كيفية الاستخدام**: سأستخدمه لتدريب نموذج تعلم آلي للتنبؤ بالتسميات بناءً على عدد الطفرات.

5. **100_genes.csv**
   - **الوظيفة**: يحتوي على معلومات الجينات المرجعية (تسلسلات ومواقع التسلسل النووي)، `Sequence`، `strand` (الاتجاه + أو -)، `gene` (اسم الجين)، و`Info` (معلومات موقع الجين على الكروموسوم).
   - **المحتوى**: جدول CSV يحتوي على أعمدة
   - **كيفية الاستخدام**: سأستخدم التسلسلات لاستخراج مناطق الطفرات باستخدام **extract_sequences.py** لتحليل التأثير الجيني.

6. **test_muts_data.csv**
   - **الوظيفة**: يحتوي على بيانات الطفرات التفصيلية لعينات الاختبار.

- **المحتوى**: جدول CSV يحتوي على `case_id`، `Gene_name`، `Chromosome`، `Start_Position`، `End_Position`، `Mut_Strand`، `Variant_Classification` (مثل Missense_Mutation)، ومعلومات الأليلات.
- **كيفية الاستخدام**: سأستخدمه لتحليل الطفرات واستخراج التسلسلات لمقارنتها مع **100_genes.csv**، وللتحقق من دقة **test_feats.csv**.

7. **train_muts_data.csv**
   - **الوظيفة**: يحتوي على بيانات الطفرات التفصيلية لعينات التدريب مع التسميات.
   - **المحتوى**: جدول CSV مشابه لـ **test_muts_data.csv** مع إضافة عمود `Label`.
   - **كيفية الاستخدام**: سأستخدمه لتحليل الطفرات وتدريب النموذج وربط التسميات مع **train_feats.csv**.

الخطة المقترحة لتنفيذ المشروع

المرحلة 1: تحليل البيانات وفهمها

- **المدة الزمنية**: 2-3 أيام
- **الخطوات**:
  1. **قراءة الملفات**: استيراد جميع الملفات باستخدام Python (مكتبة pandas لـ CSV).
  2. **فحص البيانات**:
     - التحقق من توافق عدد الطفرات في **test_feats.csv** و**test_muts_data.csv**.
     - التحقق من توافق **train_feats.csv** و**train_muts_data.csv**.
     - فحص أسماء الجينات في **100_genes.csv** مع الجينات في ملفات الطفرات.
  3. **معالجة البيانات الأولية**:
     - تنظيف البيانات (إزالة القيم المفقودة إن وجدت).
     - تحويل التسميات في **train_feats.csv** (مثل 1 و2 إلى قيم عددية).
  4. **فهم العلاقات**: تحليل العلاقة بين الطفرات التفصيلية (مثل Missense) وعدد الطفرات الكلي في ملفات الميزات.

المرحلة 2: استخراج التسلسلات الجينية

- **المدة الزمنية**: 3-4 أيام
- **الخطوات**:
  1. **إعداد البيانات**:
     - استخراج مواقع الطفرات (`Start_Position`, `End_Position`, `Mut_Strand`) من **train_muts_data.csv** و**test_muts_data.csv**.
     - ربطها مع التسلسلات في **100_genes.csv** باستخدام أسماء الجينات.
  2. **تعديل الدالة**:
     - استخدام `extract_flanked_region` من **extract_sequences.py** مع إضافة معلمة لتحديد حجم المناطق المجاورة (مثل upstream=10، downstream=10).
  3. **استخراج التسلسلات**:
     - لكل طفرة، استخراج التسلسل المرجعي مع المناطق المجاورة.
     - كما في الدالة (reverse complement) إذا كان الاتجاه "-"، حساب التكامل العكسي.
  4. **تخزين النتائج**: إنشاء ملفات جديدة (مثل `train_sequences.csv` و`test_sequences.csv`) تحتوي على التسلسلات المستخرجة مع تفاصيل الطفرات.

المرحلة 3: بناء وتدريب نموذج التعلم الآلي

- **المدة الزمنية**: 4-5 أيام
- **الخطوات**:
  1. **تحضير البيانات**:
     - استخدام **train_feats.csv** كمدخل (الميزات: عدد الطفرات لكل جين، الهدف: `Label`).
     - تقسيم البيانات إلى تدريب (80%) واختبار داخلي (20%) باستخدام `train_test_split` من scikit-learn.
  2. **اختيار النموذج**:
     - تجربة نماذج بسيطة مثل Logistic Regression أو Random Forest لأنها مناسبة لتصنيف متعدد الفئات.
  3. **تدريب النموذج**:
     - تدريب النموذج على بيانات التدريب باستخدام مكتبة scikit-learn.
     - ضبط المعلمات (مثل عدد الأشجار في Random Forest) باستخدام GridSearchCV.
  4. **تقييم النموذج**:
     - استخدام معادلة الخطأ من **Challenge_2025.pdf**: `Error = sum(predictions != true_labels) / len(true_labels)` على بيانات الاختبار الداخلية.
     - حساب مقاييس أخرى مثل Accuracy وF1-score.
  5. **التحسين**: إذا كان الأداء ضعيفًا، تجربة ميزات إضافية (مثل نوع الطفرة من **train_muts_data.csv**).

المرحلة 4: التنبؤ وتحليل النتائج

- **المدة الزمنية**: 2-3 أيام
- **الخطوات**:
  1. **التنبؤ على بيانات الاختبار**:
     - استخدام النموذج المدرب للتنبؤ بتسميات **test_feats.csv**.
  2. **مقارنة التسلسلات**:
     - مقارنة التسلسلات المستخرجة من **test_muts_data.csv** (مثل تغيير مع التسلسلات المرجعية لتحديد تأثير الطفرات الأحماض الأمينية).
  3. **إنشاء ملف النتائج**:
     - ملف CSV يحتوي على `case_id` و`predict_label` لكل عينة في **test_feats.csv**.

المرحلة 5: كتابة التقرير وإعداد التسليم

- **المدة الزمنية**: 2-3 أيام
- **الخطوات**:
  1. **كتابة التقرير**:
     - مقدمة عن المشروع وأهدافه.
     - وصف المنهجية (استخراج التسلسلات، بناء النموذج، التنبؤ).
     - النتائج (دقة النموذج، أمثلة لتسلسلات الطفرات).
     - الاستنتاجات والتحديات (مثل مشاكل OCR في **Challenge_2025.pdf**).
  2. **إعداد الملفات**:
     - ملف التنبؤات (`predictions.csv`).
     - ملفات التسلسلات المستخرجة.
     - كود Python المستخدم مع تعليقات.
  3.