Prompts - Atelier : L'IA, un outil pour le code

Prompt 1 : Le prompt maladroit et flou

"Fais-moi un graphique avec mes données d'expression génique du fichier `breast_cancer.tsv`. J'ai des gènes et des groupes de samples. **Donne-moi un script Python pour le faire.**"

Prompt 2 : Le prompt plus précis

"Je veux visualiser l'expression génique différentielle. Mon fichier `breast_cancer.tsv` a une colonne **`ID_REF`** pour les noms des gènes, et les autres colonnes sont des niveaux d'expression pour différents échantillons. Les échantillons sont nommés `Ctrl_X`, `BRCA1_X`, et `BRCA2_X`. **Génère un script Python qui me donne un graphique pour voir les différences entre ces trois groupes.**"

Prompt 3 : Le prompt très précis et complet avec Chain of Thought (CoT)

"Agis comme un bioinformaticien expert. Je souhaite créer une visualisation claire et informative de l'expression génique différentielle. **Génère un script Python complet pour réaliser ceci.**

Contexte : J'ai un fichier TSV nommé `breast_cancer.tsv`. La première colonne est **`ID_REF`** et contient les identifiants de gènes. Les colonnes suivantes sont les valeurs d'expression pour différents échantillons. Les noms de colonnes pour les échantillons suivent le format `[Condition]_[Numéro]`, où `[Condition]` peut être 'Ctrl', 'BRCA1', ou 'BRCA2'.

Objectif : Générer un **graphique en violon (violin plot)** pour un ensemble de gènes spécifiques (par exemple, `1007_s_at`, `1053_at`, `121_at`) afin de comparer la distribution de leur expression entre les trois conditions : 'Ctrl', 'BRCA1', et 'BRCA2'.

Instructions détaillées (Chain of Thought) :

1. **Chargement des données :** Commence par charger les données depuis le fichier `breast_cancer.tsv` dans un DataFrame pandas. Assure-toi de gérer le séparateur de tabulations.
2. **Transformation des données :** Le DataFrame est en format "wide". Convertis-le en format "long" pour faciliter la visualisation avec seaborn. Cela signifie avoir des

colonnes comme `Gene_ID`, `Condition`, et `Expression_Value`. Les conditions ('Ctrl', 'BRCA1', 'BRCA2') doivent être extraites des noms de colonnes d'échantillons.

3. **Filtrage des gènes :** Sélectionne les gènes d'intérêt spécifiques : `1007_s_at`, `1053_at`, et `121_at`.
4. **Génération du plot :** Utilise `matplotlib.pyplot` et `seaborn` pour créer le graphique en violon.
    - L'axe des X doit représenter les conditions ('Ctrl', 'BRCA1', 'BRCA2').
    - L'axe des Y doit représenter le niveau d'expression.
    - Chaque violon doit correspondre à une condition.
    - Utilise `col='Gene_ID'` pour créer une facette pour chaque gène sélectionné (un graphique séparé par gène).
    - Personnalise les couleurs pour les conditions si possible (par exemple, 'Ctrl' en bleu, 'BRCA1' en orange, 'BRCA2' en vert).
    - Ajoute un titre clair : 'Distribution de l'Expression Génique pour les Gènes Sélectionnés par Condition'.
    - Les légendes des axes doivent être 'Condition' et 'Niveau d'Expression'.
    - Assure-toi que le graphique est lisible et que les étiquettes ne se chevauchent pas.
    - Enregistre le graphique sous 'gene_expression_violin_plot.png' avec une haute résolution."