v1.1 Projet CY-Antibio-Tech

FILIERE préING2 • 2024-2025
AUTEURS R.GRIGNON – C. RODRIGUES NASCIMENTO
E-MAILS romuald.grignon@cyu.fr – clarissa.rodrigues-nascimento@cyu.fr

DESCRIPTION GENERALE

- Le but de ce projet est de pouvoir générer des graphiques automatiquement à partir de données qui proviennent de laboratoires. Ces données sont issues d'une expérience pour laquelle on a requis votre aide.
- Vous devez, en plus de fournir un outil capable de générer une synthèse des données, être capable d'interpréter les résultats obtenus.
- Techniquement vous devez réaliser un Programme Python qui va devoir traiter un fichier CSV en entrée, puis créer plusieurs fichiers de données en sortie, ainsi que plusieurs graphiques.
- Le fichier de données d'entrée devra être copié dans un dossier '*input*'
- Les graphiques générés seront stockés dans des images sur le disque dur dans un dossier '*images*'
- Les fichiers de données de sortie générés par votre programme seront placés dans un dossier '*output*'.
- Le fichier de script Python sera placé à la racine de votre projet.

EXPERIENCE

- L'expérience consiste à traiter des groupes de souris avec un cocktail d'antibiotiques au début de leur vie, et de voir l'impact que cela a sur le taux de bactéries au niveau du microbiote intestinal.
- Un groupe de souris sera traité avec l'antibiotique, et un autre avec un placebo (contrôle négatif) afin de faire une comparaison.
- Toutes les souris ne démarrent la phase d'expérimentation qu'à partir de leur 14ème jour de vie.
- Le traitement est donné à chaque souris pendant 7 jours d'affilée et au 21ème jour, démarre la phase de rinçage, c'est à dire que le cocktail de traitement n'est plus fourni.
- Les données recueillies à différents moment de l'expérimentation sont des estimations de la quantité de bactéries vivantes dans les matières fécales des souris (fèces).
- Pour des besoins expérimentaux, il est possible que certaines souris soient sacrifiées à un moment du processus, afin de récolter la quantité de bactéries présentes dans les différents étages du système digestif (iléon, caecum).

- L'expérience doit pouvoir fournir des graphiques comme suit :
  - un graphique sous forme de lignes multi-segments (1 par souris) pour la quantité de bactéries au niveau fécal. Il y aura un groupe de courbes pour les souris avec antibiotique, et un groupe de courbes pour les souris avec placebo.
  - un graphique de type violon, représentant la dispersion de la quantité de bactéries pour l'ensemble des souris au niveau cécal et iléal (2 graphiques). Sur ce même graphique, on pourra distinguer la dispersion pour le groupe de souris avec antibiotiques, du groupe sans.
- Tous les graphiques doivent avoir des couleurs cohérentes entre eux pour identifier rapidement les résultats du groupe avec antibiotique, de celui avec placebo.
- De plus, un titre, une légende, et des axes avec unités doivent être présents pour qu'il n'y ait pas d'ambiguité sur les données visualisées par les biologistes.
- Enfin, pour pouvoir créer les précédents graphiques, il faudra filtrer et traiter les données brutes provenant du fichier CSV. Il est demandé que les données filtrées, nécessaires à la construction de chaque graphique, soient stockées au format CSV dans des fichiers.
- Il y aura donc 3 fichiers de données générés par le programme, chacun correspondant aux données des 3 graphiques.
- Pour les besoins pédagogiques, les véritables données de l'expérience ne seront fournies que vers la fin du projet : d'autres valeurs vous seront fournies pour que vous puissiez préparer votre programme correctement, et une fois prêt, les véritables données vous parviendront et vous pourrez commencer à interpréter les résultats.

FORMAT DES DONNEES

- Les fichiers CSV provenant des laboratoires ont tous la même structure. Les noms des colonnes et leurs fonctions sont décrits comme suit :
  - mouse_strain : souche de souris
  - experiment_id : identifiant de l'expérience
  - **sample_type** : type d'échantillon recueilli (fécal, iléal, cécal)
  - timepoint : nom de l'étape de l'expérience
  - **mouse_id** : identifiant de la souris
  - **treatment** : antibiotique ou placebo
  - frequency_live_bacteria : pourcentages de bactéries vivantes
  - **experimental_day** : jour relatif au 21ème jour des souris
  - **counts_live_bacteria** : quantité absolue de bactéries vivantes
  - mouse_age_days : âge absolu de la souris en jours
  - mouse_sex : souris mâle ou femelle
- Il conviendra donc de traiter ces informations pour créer des courbes de chaque souris, en fonction des graphiques attendus.
- Le nombre de souris n'est pas fixe, et votre programme devra pouvoir en gérer un nombre indéfini.
- De même, pouvoir gérer un nombre indéfini de 'jours' d'expérimentation serait un plus au niveau de votre programme.

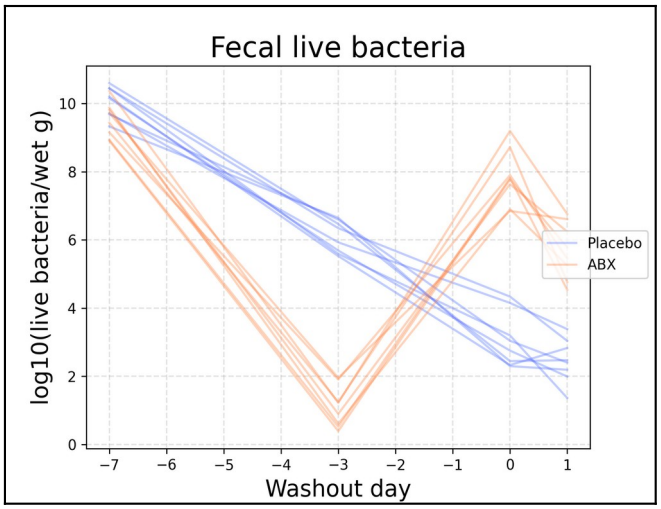
DONNEES & GRAPHIQUES

➢ **Données pour les graphiques en lignes**

Les données doivent être filtrées en ligne pour conserver le type voulu (mesures dans les matières fécales)

Ensuite il faut filtrer en colonne pour ne conserver que l'identifiant de la souris (pour pouvoir créer sa courbe dédiée) et son traitement (groupe de courbes), le numéro du jour de l'expérience (abscisse), et la quantité de bactéries vivantes (ordonnée).

Grâce à ces informations, vous serez en mesure de créer le fichier de données filtrées, ainsi que le graphique associé.



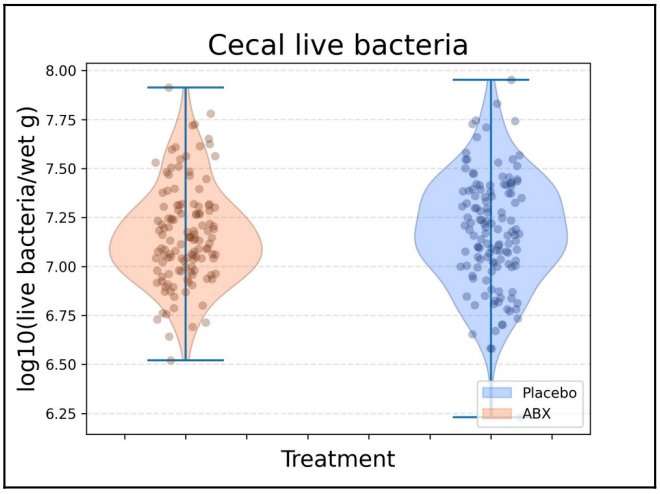
➢ **Données pour les graphiques de type violon**

Ici les données doivent être toujours filtrées en ligne pour avoir uniquement le type d'échantillon voulu (cécal ou iléal).

Ensuite cela ne sert plus à rien de conserver l'identifiant de la souris, car le but est de récupérer les valeurs de tout le groupe.

On ne va conserver que le 21ème jour de vie des souris (7ème jour de l'expérience) car c'est à cette date que certaines souris sont sacrifiées et que les mesures sont faites.

Bien entendu on conserve la valeur de bactéries présentes.



- **Les fichiers de données fournis :**
  Sur la page de cours vous trouverez des dossiers contenant les fichiers de données que vous pourrez utiliser : il y a 4 dossiers nommés small, medium, large and huge. Tous ces répertoires contiennent des données similaires, seules le nombre de données contenues changent.

  Le but est de vous fournir des fichiers CSV de différentes tailles pour que vous puissiez tester que votre programme s'adapte automatiquement en fonction de la quantité de données fournies.
  Tous les fichiers CSV ont la même structure (nombre et ordre des colonnes) : seules le nombre de lignes et les valeurs vont changer. Votre programme doit être suffisamment générique pour traiter n'importe lequel de ces fichiers CSV.
  Commencez par traiter le plus petit, et une fois le programme opérationnel, vérifiez/corrigez votre programme avec les fichiers CSV plus gros.

  Dans ces dossiers se trouvent également des images de référence pour que vous puissiez vérifier que vos graphiques générés par le programme sont corrects.

CRITERES DE NOTATION

- Le rendu du travail sera un **lien github** menant au projet. Inutile d'envoyer les fichiers par email/Teams : le seul livrable attendu sera le lien du dépôt git dans lequel doivent se trouver tous les fichiers de votre projet. **Avant la date** de rendu vous pouvez configurer ce dépôt en « **privé** » pour ne pas laisser d'autres personnes vous plagier.
  A la date de rendu, ce dépôt devra être visible **publiquement** pour que vos chargés de projet puissent y accéder librement. Tout retard d'accès à ce dépôt public aura un impact négatif sur votre note finale.
- Ce dépôt de code contiendra en plus des fichiers de code, un fichier ReadMe contenant les instructions pour lancer et utiliser votre application. Il contiendra aussi les limitations fonctionnelles de votre application (la liste de ce qui n'est pas implémenté, et/ou de ce qui est implémenté mais qui ne fonctionne pas correctement/totalement) afin de montrer que vous connaissez le périmètre fonctionnel de votre projet.
- Le rendu est un travail individuel : si des similitudes étranges entre projets sont trouvées, et/ou si des exemples disponibles sur Internet sont découverts sans être sourcés, une procédure de fraude à un examen pourra être envisagée. Le but pédagogique de ce projet est que vous réalisiez par vous-même ce programme, et que vous maitrisiez l'ensemble du code fourni.
- Votre code sera **commenté** (modules, fonctions, structures, constantes, …) en langue **anglaise** et correctement **indenté**.
- Les **symboles** du code (variables, fonctions, types, fichiers, …) seront dans la **même langue** que les **commentaires** (en anglais).

- Vous disposez de fichiers de données CSV figés. Il est donc possible pour un étudiant de « coder en dur » les résultats attendus. Pour éviter ce cas de triche, il est possible que l'évaluation de votre programme se fasse avec des fichiers de données CSV **différents** de ceux que vous aurez eu (mais similaires en terme de structure, de taille, …). Pensez donc à faire un programme véritablement générique pour éviter une mauvaise surprise lors de l'évaluation.
- Chaque graphique sera évaluée sur les données affichées, la séparation visuelle des groupes antibiotique et placebo, les titres et unités sur les axes, le titre de la figure, la légende.
- Chaque fichier de donnée généré sera évalué sur le contenu de la première ligne, l'utilisation d'un séparateur de colonnes, le nombre de colonnes et de lignes, et les valeurs des données.

RESSOURCES UTILES

GitHub

- site Web : https://github.com/

Format CSV

- site Web : https://fr.wikipedia.org/wiki/Comma-separated_values

Matplotlib

- site Web : http://matplotlib.org/