

Notice utilisateur

Création d'un outil informatique pour contribuer à l'évaluation de néo-fongicides bio-sourcés

Semestre 2

Etudiants :

Matteo BATTUT

Marie LEMAITRE

Victor LORIDAN

Oussama NABILI

Malo TOURNEUR

Nina T'SERSTEVENS

Commanditaires : Caroline DEWEER, Jérôme MUCHEMBLED, Karin SAHMER

Année universitaire : 2022 – 2023

Introduction

Il est intéressant aujourd'hui de s'intéresser à de nouvelles solutions de biocontrôle, notamment de fongicides bio-sourcés. L'équipe de recherche BioGAP de Junia ISA Lille est chargée de trouver un fongicide efficace, solution contre les champignons parasites, tout en étant d'origine biologique. Ce projet fait suite à un stage effectué par Axel Belotti en 2019 au sein de l'équipe de recherche. C'est lui qui a créé le programme R qui permet aujourd'hui aux biologistes d'effectuer leurs tests statistiques. Le projet a été continué par Sidy D DOUCOURE et Benoit GOZE à l'ISE en 2020. Ils ont travaillé sur la méthode du Bootstrap pour offrir une alternative et résoudre certains problèmes engendrés par le test de permutation. Ce sont des tests qui servent à estimer la fiabilité ainsi que les intervalles de confiance des résultats.

Le but de ce projet était dans un premier temps, de faire un état de l'art du sujet et d'expliquer le contexte pour amorcer au mieux la suite du projet au deuxième semestre. Cette seconde partie a pour but d'améliorer les performances du programme, ainsi que de proposer une interface graphique facilitant la prise en main de celui-ci pour les biologistes l'utilisant.

Cette notice contient dans un premier temps une explication du code R initial ainsi que les nouveautés apportées sur du code R Shiny.

Cette deuxième phase du projet, l'équipe s'est donc concentrée sur l'aspect technique. Le premier objectif consistait à choisir le langage de programmation approprié, avec Python et R comme options initiales. Après plusieurs jours de réflexion, R a été sélectionné, principalement parce que le projet visait à créer une interface graphique basée sur des scripts R existants. Il semblait plus simple d'utiliser R Shiny pour l'interface plutôt que de tout traduire en Python.

Tout au long du projet, l'équipe a poursuivi divers objectifs intermédiaires, définis lors des réunions hebdomadaires auxquelles les commanditaires et encadrants ont participé. Tout d'abord, il a été nécessaire de s'approprier et de comprendre le code existant, son architecture et ses nombreuses fonctions et variables grâce notamment avec la création d'une carte mentale. Ensuite, il a fallu clarifier et organiser le code en séparant les scripts à conserver, modifier, ou supprimer. Pour continuer, il a été indispensable de s'assurer que les fonctions du programme pouvaient fonctionner avec R Shiny. Parallèlement au troisième objectif, un premier prototype d'interface a été présenté. Enfin, une interface intuitive pour les commanditaires a été proposée, leur permettant de saisir simplement leurs paramètres et fichiers de données afin d'obtenir les graphiques et les calculs souhaités.

Fonctionnement de l'interface

Le but de la création de cette interface et de rendre l'utilisation du programme plus simple et intuitive pour les biologistes. Avant d'accéder à cette première page de l'interface il faut télécharger l'ensemble du dossier « ProgPrincip », puis ouvrir le fichier App.R. Avant de lancer l'interface, il faut s'assurer que l'environnement R soit vidé pour que les anciennes données n'interfèrent pas avec les prochaines données. Ensuite pour ouvrir l'interface, il faut appuyer sur le bouton "Run App" situé en haut à gauche de la fenêtre pour lancer le programme.

Ensuite, la première page de l'interface s'ouvre dans laquelle les chercheurs peuvent choisir les paramètres tel que le choix des données à importer, la valeur de l'écart type et bien d'autres. Il peut aussi renseigner le nom des colonnes de son jeu de données. Une fois que tous les paramètres sont indiqués, un bouton est présent permettant de lancer tous les calculs nécessaires pour produire les graphiques à étudier. Le programme met en certain temps à exécuter l'ensemble des actions. Mais une fois que les calculs sont faits, un tableau du jeu de données apparaît. En haut de la page principale, des onglets ont été créés. Ils permettent d'accéder aux feuilles de calcul présentées sous la forme d'un tableau ou bien affichent les graphiques créés. Ces derniers sont plus utiles et représentatifs pour les biologistes. En effet, cela leur permet d'avoir un premier aperçu sur les résultats de leurs recherches.

Lors de l'utilisation de l'application, il était impératif de pouvoir télécharger tous ce qui est produit par le programme (tableaux et graphiques). C'est une option qui a été rajoutée au code. Au lancement des calculs, un dossier est créé. Il est possible de choisir son nom dans l'interface en bas de l'encadré gauche. Puis le programme ajoute toutes les figures produites dans différents dossiers comme pour le programme initial. Cela permet aux biologistes de conserver une trace de leurs résultats.

L'onglet « Data » contient un tableau avec le jeu de donnée après leur traitement pour garder seulement les informations utiles. Ensuite, l'onglet « Stas » contient le tableau de données après la statistique descriptive. Les graphiques créés lors de la statistique descriptive sont présents dans l'onglets « Plot ». En ce qui concerne l'étape du calcul des CI50, le fichier csv et les plots sont respectivement affichés dans les onglets « Calcul_CI50 » et « Plots_CI50 ». Pour continuer, le fichier csv créé lors de diagnostic est présenté dans l'onglet de même nom. L'onglet « Comparaison_CI50 » montre les deux fichiers csv lors de la comparaison globale et la comparaison par paire. Pour la comparaison globale, une valeur approchée de la pvalue est affichée dans le tableau mais sa valeur réelle est écrite juste en dessous. Pour finir, les documents créés lors de l'étape résumé du programme sont présents dans l'onglet « Resume_CI50 ».

La dernière option qui a été rajoutée permet d'ouvrir l'interface sans devoir ouvrir un script sur R studio. Pour ce faire, il faut suivre les instructions suivantes :

1- Ouvrir le fichier RunApp.R dans le dossier ProgPrincip :

Changer le contenu de la variable "folder_address" par le chemin vers le fichier "AppPrincip.R".

Par exemple : "C:/ProjetM1/ProgrammePrincipal/AppPrincip.R"

```
require(shiny)
----> folder_address = '*****'
runApp(folder_address, launch.browser=TRUE)
```

2- Renommer le fichier RunApp.bat en un fichier RunApp.txt :

Faire un clic droit sur le fichier RunApp.bat.

Appuyer sur "Renommer".

Changer ".txt" en ".bat".

(Si le format du fichier n'est pas affiché, il suffit d'aller dans l'onglet « Afficher » de l'explorateur de fichier puis aller sans afficher/masquer pour enfin cocher la case « Extension des noms de fichier ».)

3- Ouvrir RunApp.txt :

Changer le chemin vers le fichier "R.exe".

Par exemple : "C:\R-4.2.3\bin\R.exe" CMD BATCH "RunApp.R"

4- Renommer le fichier RunApp.txt en un fichier RunApp.bat.

5- Il est désormais possible de lancer l'interface graphique sans passer par Rstudio :

Ouvrir le fichier RunApp.bat.

Perspectives d'amélioration

Il existe plusieurs pistes d'amélioration pour le programme. Tout d'abord, il est possible d'optimiser davantage le code afin d'améliorer les performances et la rapidité d'exécution. De plus, il pourrait être intéressant de proposer le choix des IC allant de 0 à 100, et pas uniquement IC50, pour offrir une plus grande flexibilité aux utilisateurs. Il serait également judicieux de rendre fonctionnelle la méthode de calcul d'intervalle de confiance Wald en plus de la méthode Bootstrap.

Enfin, dans un avenir plus lointain, il pourrait être envisagé de traduire le code en anglais dans le cas où ce programme serait utilisé à l'international ou si la continuité du projet se fait avec des chercheurs ou étudiants anglophones. En résumé, ces différentes améliorations permettraient d'offrir un programme encore plus performant et convivial pour les utilisateurs.

Conclusion

En conclusion, le projet a abouti à la création d'une interface graphique sympathique et pratique pour les biologistes qui effectuent des tests statistiques pour évaluer l'efficacité de bio fongicides. Cette interface offre aux utilisateurs une méthode plus maniable pour analyser leurs données, gagner du temps. Dans l'ensemble, ce projet est un exemple réussi de la conception d'interfaces utilisateur pour des applications scientifiques. L'interface peut être utilisée comme modèle pour d'autres projets similaires, montrant comment une interface bien conçue peut améliorer l'efficacité et la qualité du travail des scientifiques.