JUNIA ISEN

Ingénieurs pour le numérique.

Notice technique

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 a 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.

Familiarisation avec le code initial et les nouveautés à y apporter

Explication du code R

Pour comprendre le programme initial, une carte mentale a été créée avec le site internet lucid.app qui permet de créer et partager des cartes mentales et autres graphiques. Afin d'organiser cette carte mentale, un jeu de couleurs a été choisi. Les appels de fonctions et de fichiers sont écrits en rouge, des captures d'écran des résultats du programme ont été ajoutées avec des étiquettes pour chaque image en fonction des conditions d'entrées. Ainsi, les étiquettes bleues sont lorsque le jeu de données ne contient qu'une manipulation d'un seul composé, celles rouges pour une manipulation et plusieurs composés et enfin les vertes pour plusieurs composés et manipulations. Aussi, les différentes étapes : Statistiques descriptives, calcul CI50 et comparaisons globales des CI50 sont écrites en violet. Pour finir, des modifications possibles à apporter sont écrites en vert.

Afin, d'expliquer le programme des parties de la carte mentale seront présentées ici, la carte mentale dans son intégralité sera présente à l'annexe n°1. Pour certaines figures de cette partie un « 1 » est placé pour indiquer le début et le sens de lecture de la carte mentale.

Pour commencer le programme, il faut le fichier utilisateur.R, il est le seul fichier utilisé par les biologistes afin de personnaliser le programme en fonction de leurs données et leurs besoins (Figure 1). Voici ci-dessous la partie de la carte mentale décrivant le fichier utilisateur.R qui est directement relié au fichier programme.R avec la flèche rouge. Il faut commencer la lecture des caractéristiques de ce fichier par « on vide l'environnement … », puis lire dans le sens antihoraire.

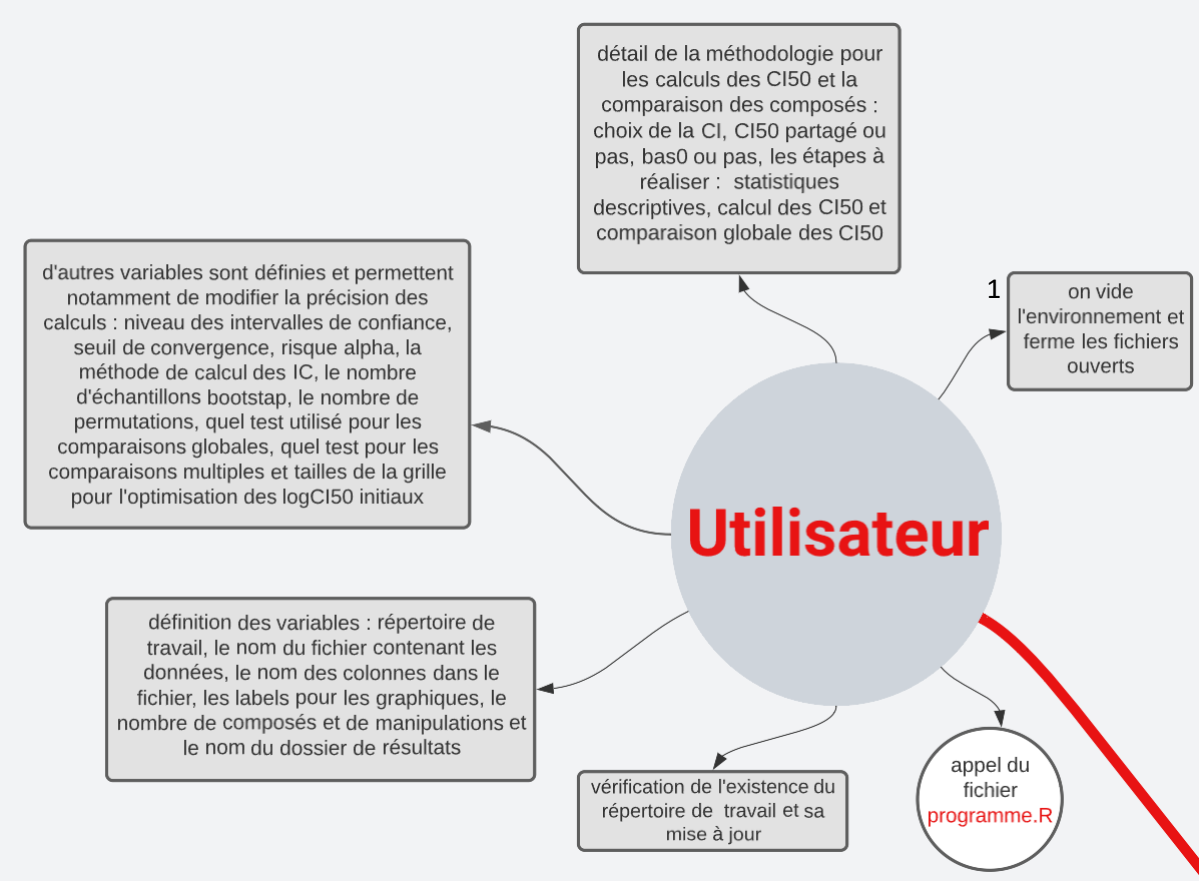


Figure 1- utilisateur.R

Ensuite, le fichier central de programme et le fichier programme.R ci-dessous. Ce fichier contient les appels de fonctions pour calculer et afficher les résultats. Ci-dessous, seules les parties principales sont inscrites, elles seront vues plus en détails par la suite (Figure 2). Ici, c'est dans le sens horaire qu'il faut lire la figure.

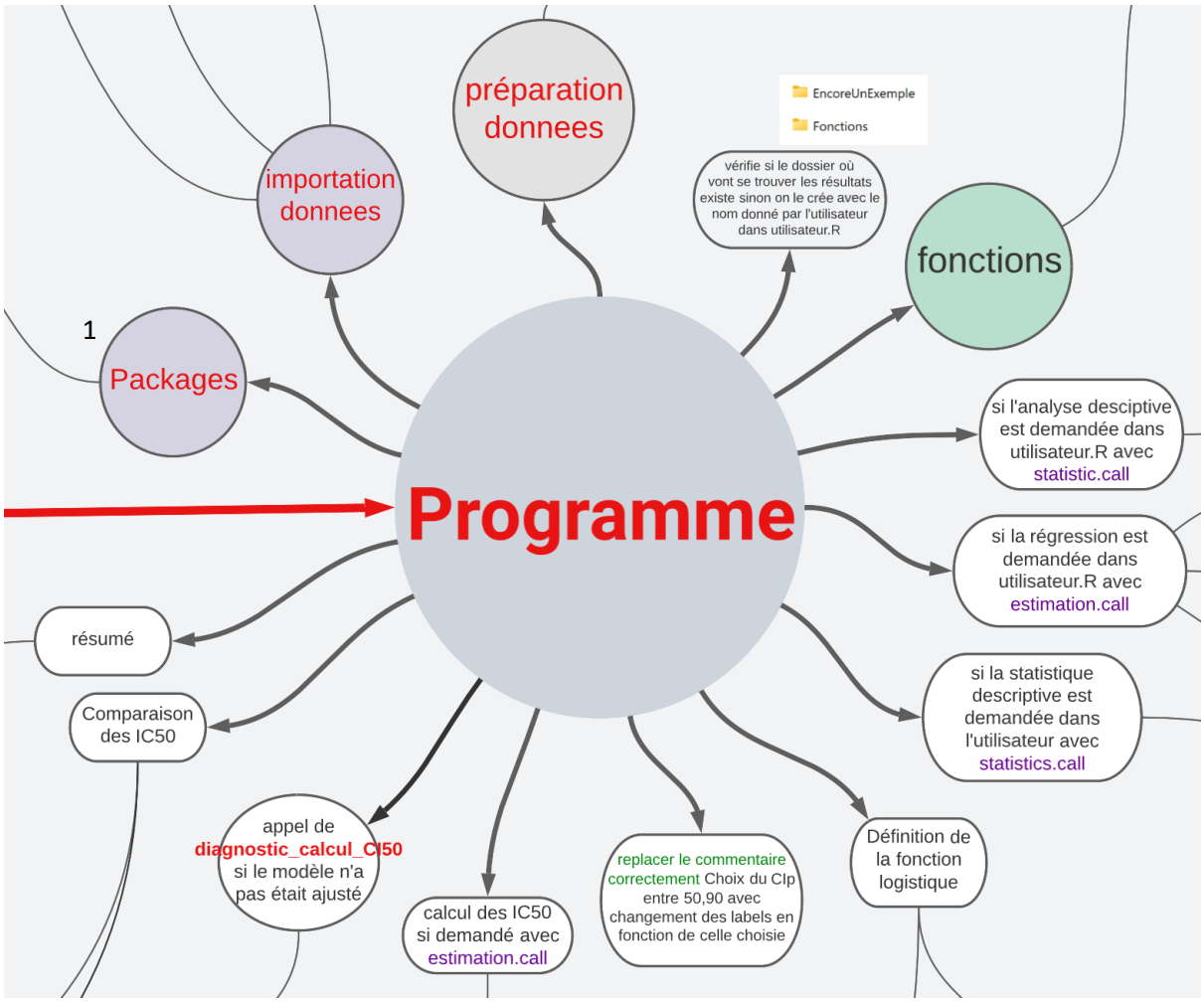


Figure 2 - programme.R

Le premier fichier appelé par programme est : packages.R, il permet de charger les packages utiles à l'exécution du programme. Voir ci-après Figure 3.

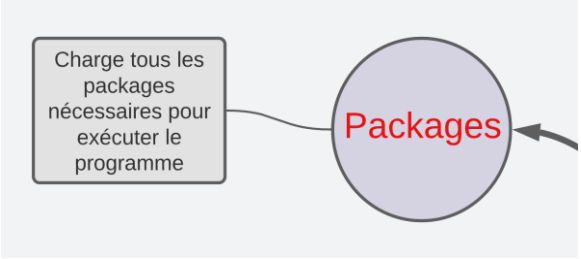


Figure 3 – packages.R

Ensuite, le fichier importation_donnees.R est appelé, il permet comme son nom l'indique d'importer les données du fichier csv renseigné par l'utilisateur (Figure 4).

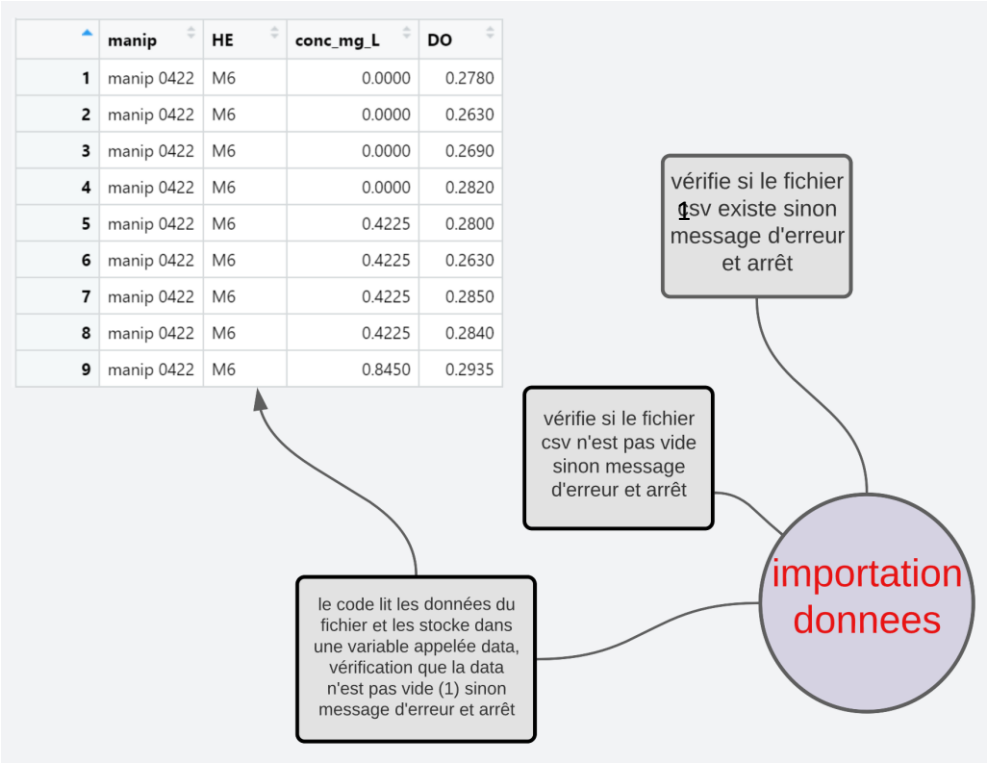


Figure 4 - importation_donnees.R

Pour continuer, le fichier preparation_donnees.R est appelé (Figure 5). Il sert à sélectionner les données souhaitées et les mettre en page.

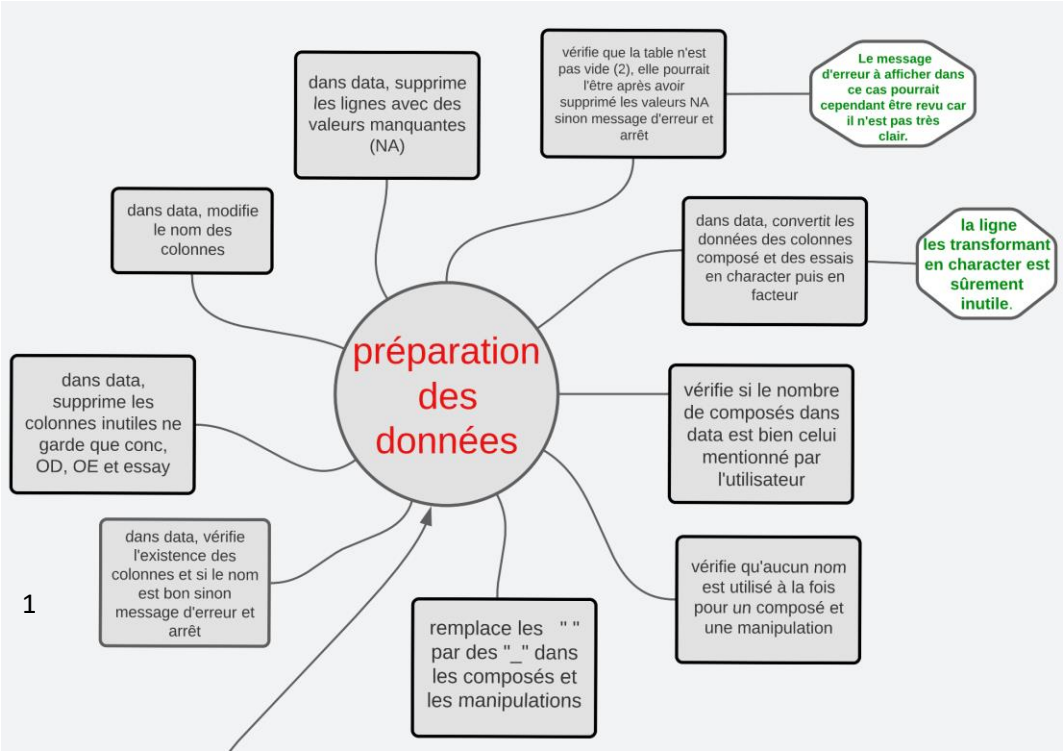


Figure 5 - preparation_donnees.R

Comme on peut le voir sur la figure 1, la prochaine action du fichier programme est de vérifier si le dossier de résultats a été créé (Figure 6). Ensuite le fichier appelle le dossier Fonctions qui regroupe toutes les fonctions mathématiques utiles au calcul et comparaison des CI (Figure 7).

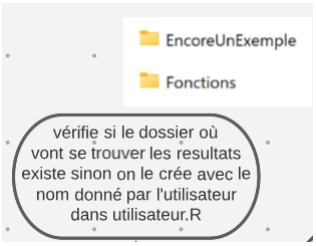


Figure 6 - Vérification de l'existence du dossier

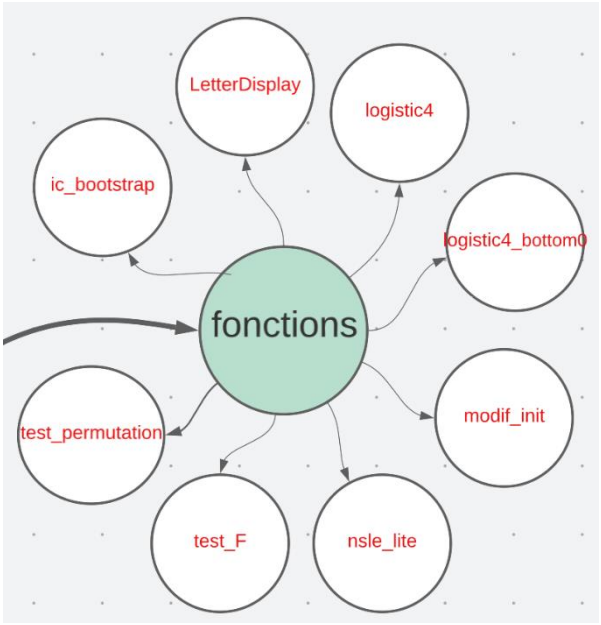


Figure 7 - Fonctions

Pour poursuivre, en fonction des étapes du programme demandées par l'utilisateur, de nouveaux dossiers sont créés, ils vont contenir les résultats (Figure 8).

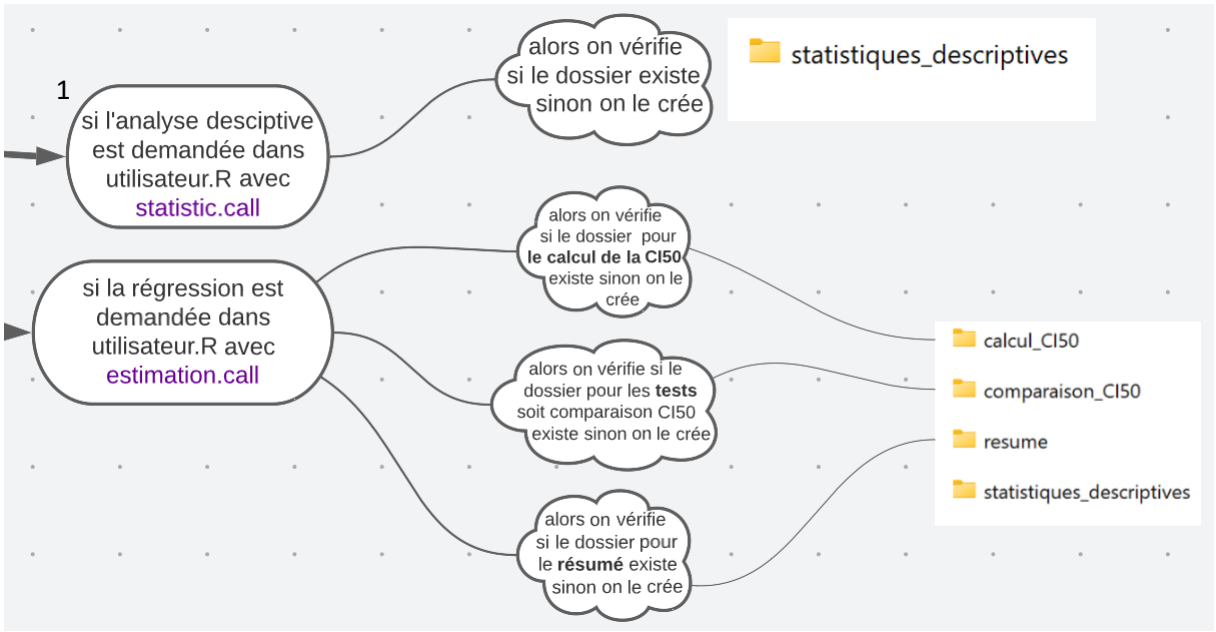


Figure 8 - création de nouveaux dossiers

Ensuite, l'étape analyse descriptive est effectuée si elle a été demandée par l'utilisateur (Figure 9). Elle permet de créer des représentations graphiques pour représenter la relation entre les densités optiques et les concentrations des échantillons pour chaque manipulation et chaque composé. Les images et document créés par cette étape sont enregistrés dans le dossier « statistiques_descriptives » (Figures 10).

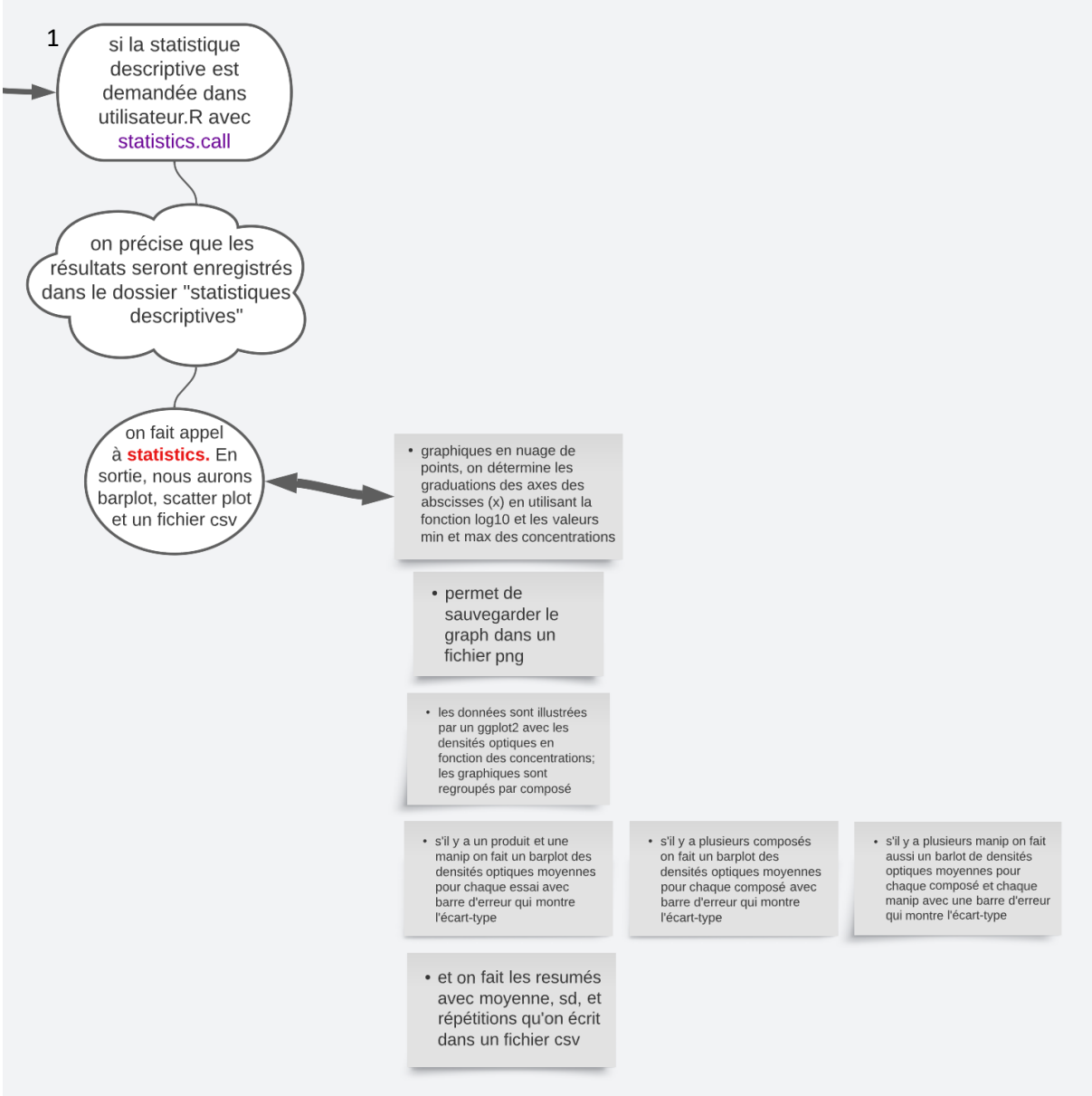


Figure 9 - Statistiques descriptives

	A	B	C	D	E	F
1	EO	essay	conc	mean	sd	repetitions
2	Cu	manip_0422	0	0,267	0,03441899	4
3	Cu	manip_0422	1	0,24675	0,01822773	4
4	Cu	manip_0422	2	0,21275	0,02632331	4
5	Cu	manip_0422	3,90625	0,234	0,01534058	4
6	Cu	manip_0422	7,8125	0,20225	0,01878608	4
7	Cu	manip_0422	15,625	0,17675	0,00644851	4
8	Cu	manip_0422	31,25	0,15975	0,02844146	4
9	Cu	manip_0422	62,5	0,119	0,01538397	4
10	Cu	manip_0422	125	0,05675	0,00780491	4
11	Cu	manip_0422	250	0,01525	0,00298608	4
12	Cu	manip_0422	500	0,0155	0,001	4
13	Cu	manip_0422	1000	0,0095	0,00351188	4
14	M10	manip_0422	0	0,242	0,02483277	4
15	M10	manip_0422	0,4225	0,23675	0,02317146	4
16	M10	manip_0422	0,845	0,197	0,01948504	4
17	M10	manip_0422	1,69	0,21775	0,01674067	4
18	M10	manip_0422	3,38	0,218	0,00374166	4

Figure 10 – Exemples de documents de sortie après l'analyse descriptive

A la suite de la statistique descriptive, le calcul de la fonction logistique est fait en fonction des paramètres d'entrée (Figure 11). Puis le choix entre la CI50 et la CI90 est faite (Figure 12). Actuellement seule la CI50 est fonctionnelle.

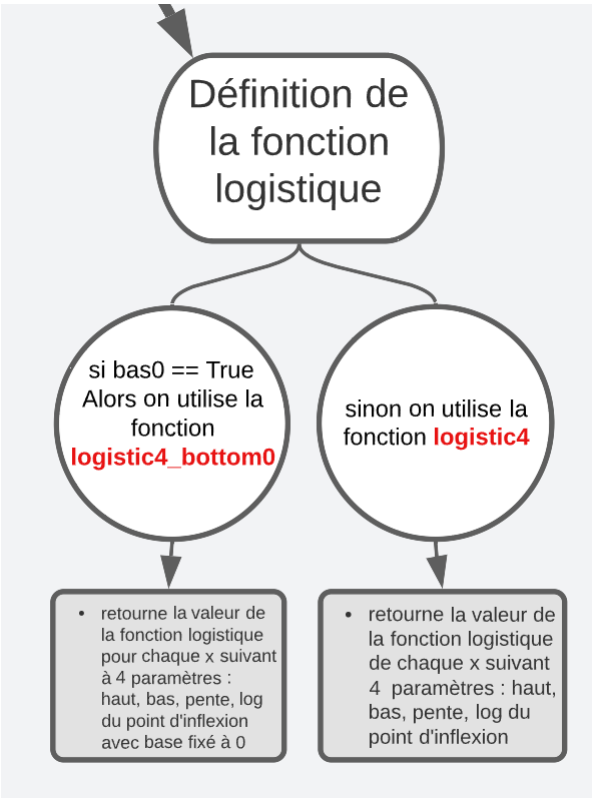


Figure 11 - Calcul de la fonction logistique

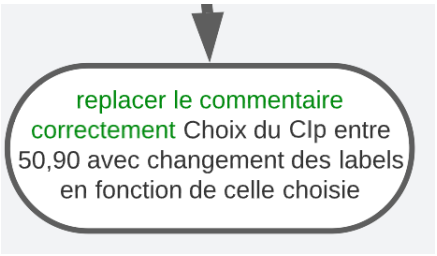


Figure 12 - Choix de l'IC

La prochaine étape est celle du calcul de la CI grâce à différentes fonctions et fichier R (Figure 13). Elle a comme sortie différents graphiques (Figure 14).

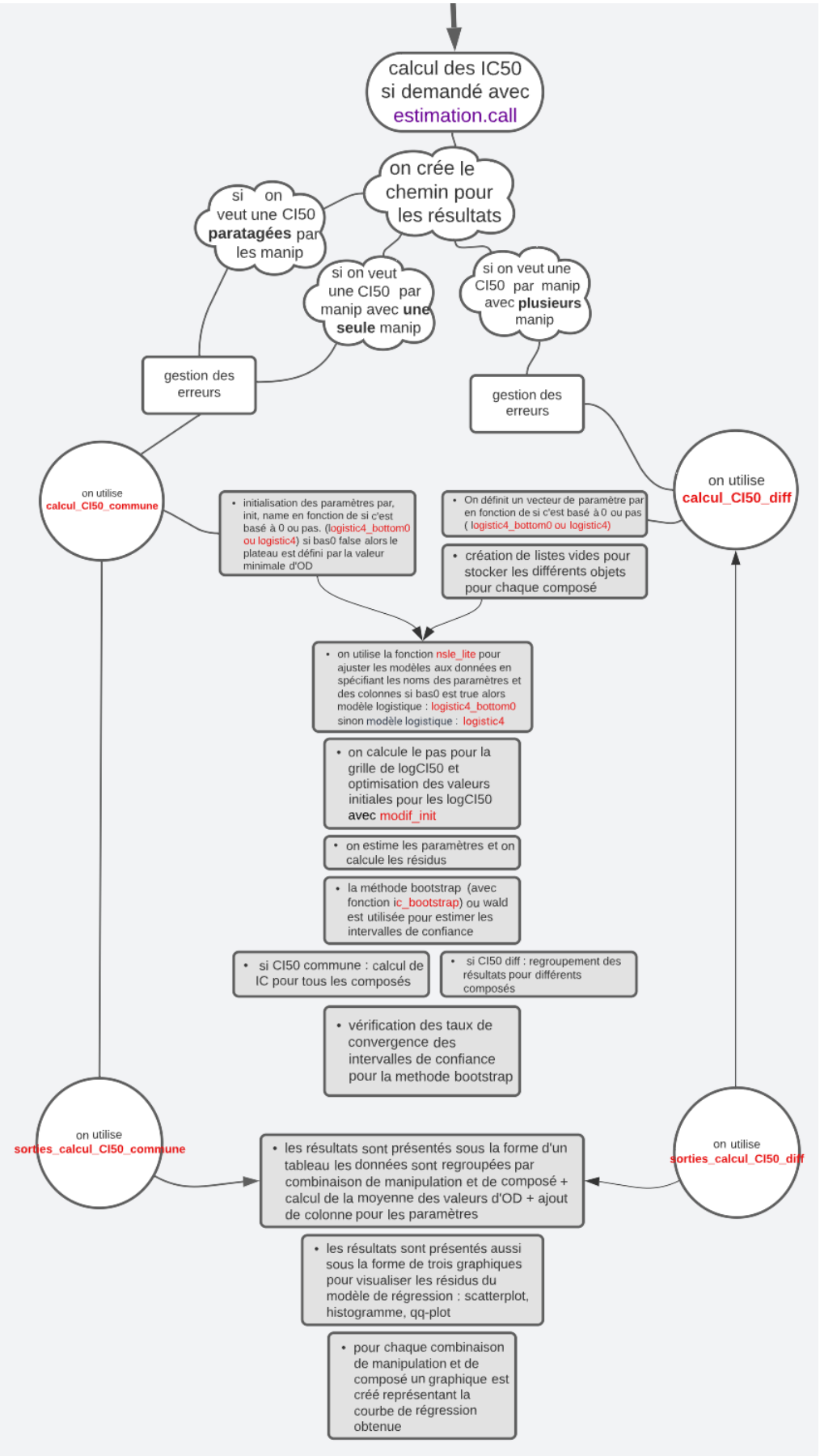


Figure 13 - Calcul des CI

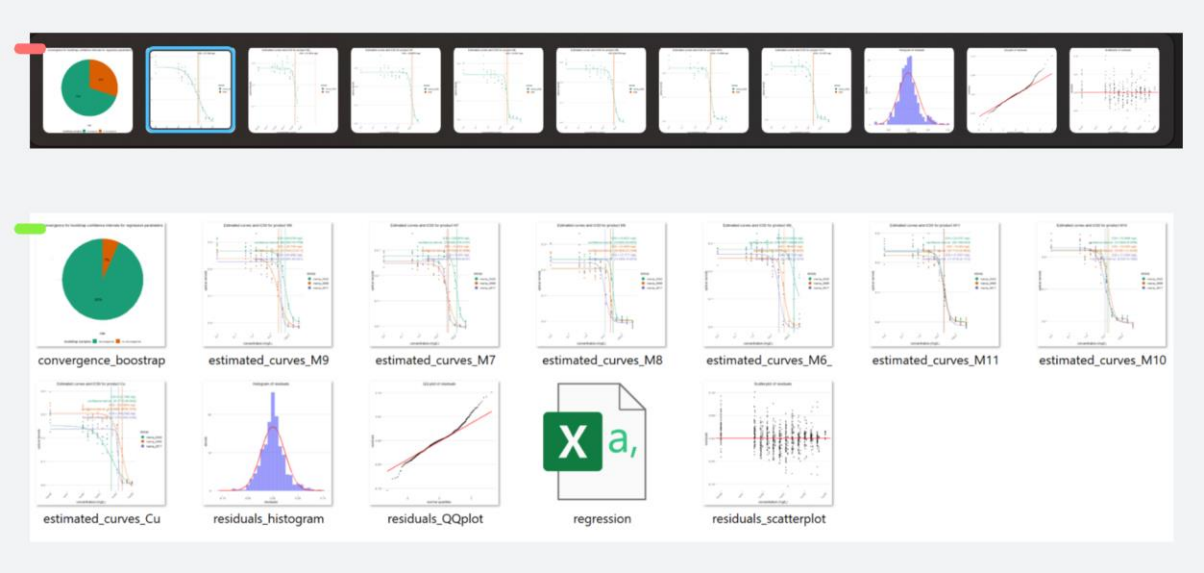
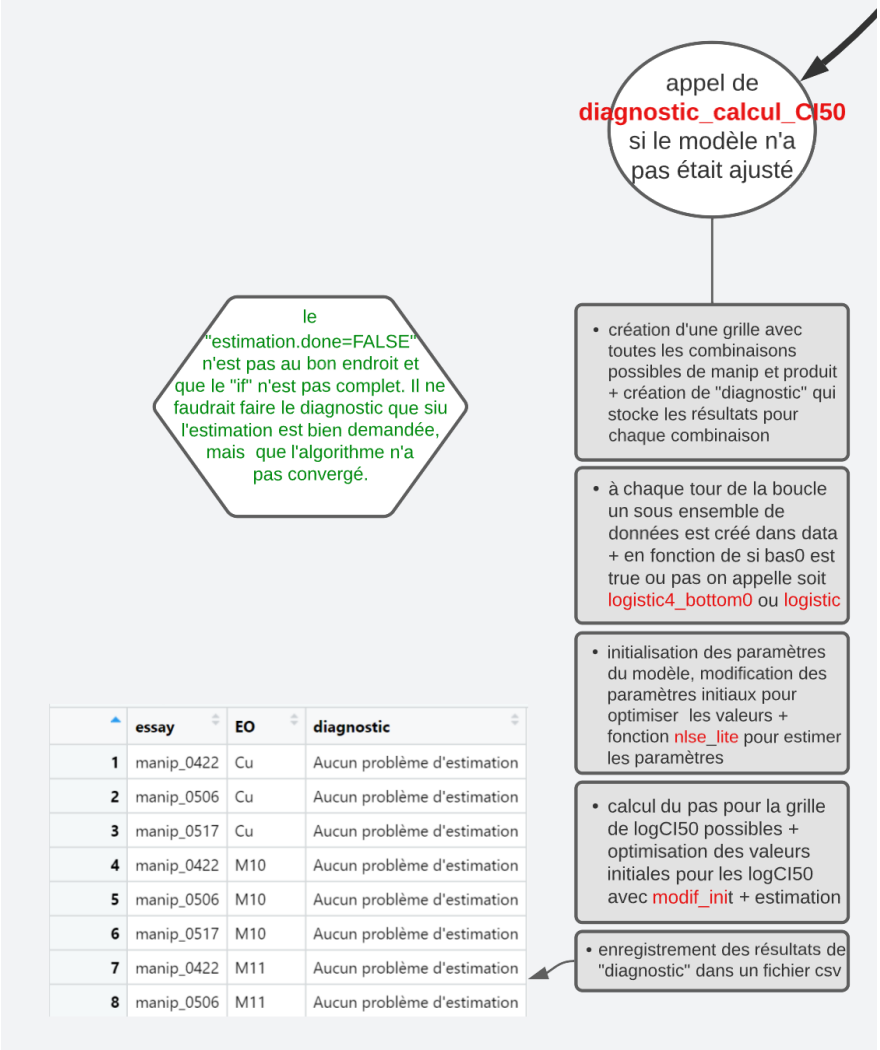


Figure 14 - Exemples de documents après calcul des CI

Pour continuer, il faut effectuer le diagnostic de la CI50, cela permet de savoir si le modèle est adapté pour la suite des calculs (Figure 15).



	essay	EO	diagnostic
1	manip_0422	Cu	Aucun problème d'estimation
2	manip_0506	Cu	Aucun problème d'estimation
3	manip_0517	Cu	Aucun problème d'estimation
4	manip_0422	M10	Aucun problème d'estimation
5	manip_0506	M10	Aucun problème d'estimation
6	manip_0517	M10	Aucun problème d'estimation
7	manip_0422	M11	Aucun problème d'estimation
8	manip_0506	M11	Aucun problème d'estimation

Figure 15 - diagnostic_calcul_CI50.R

Une fois ceci fait, les comparaisons des CI50 peuvent être faites (Figure 16). Avec cette partie de code deux fichiers csv sont créés en fonction de si la comparaison est globale ou par paires (Figure 17).

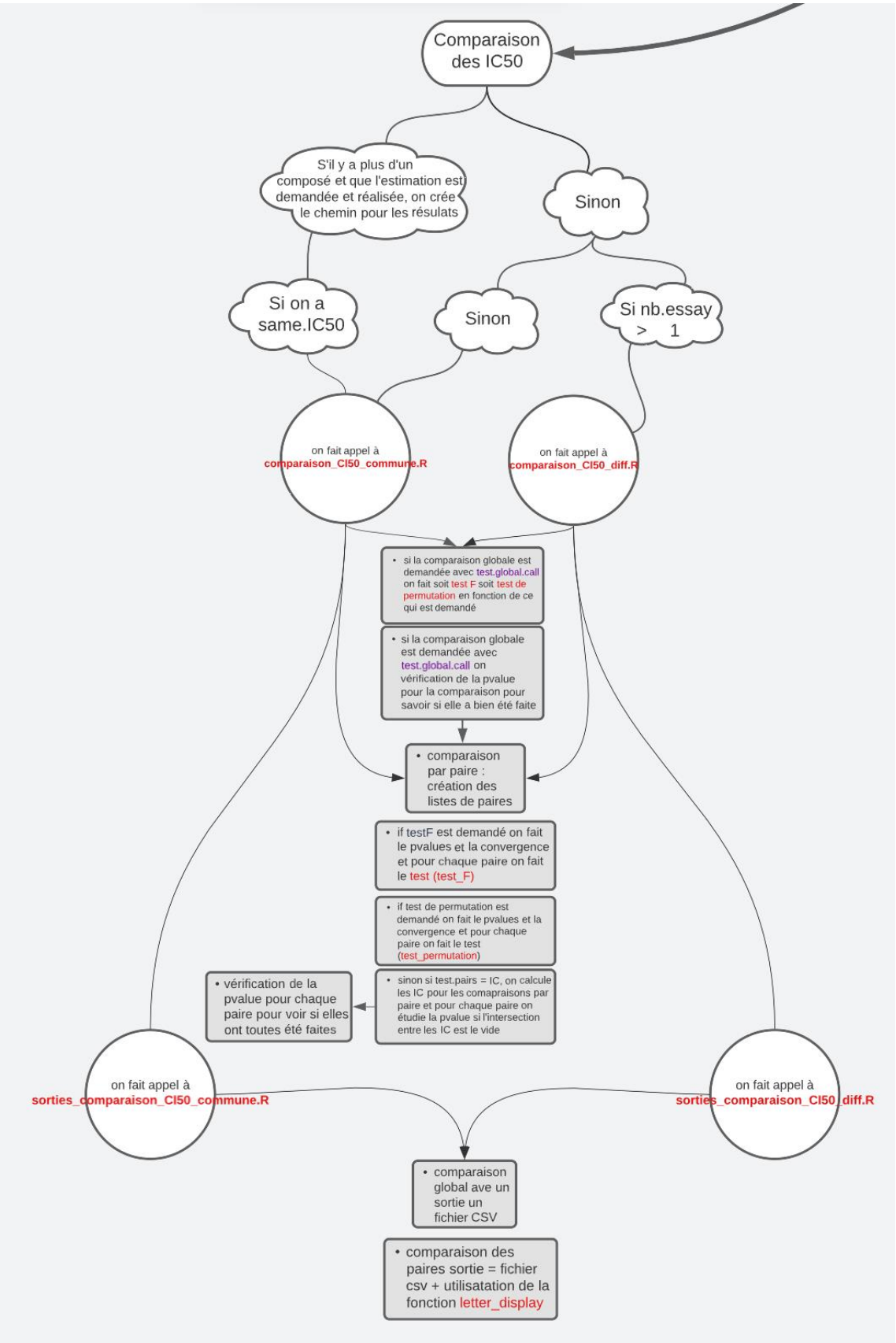
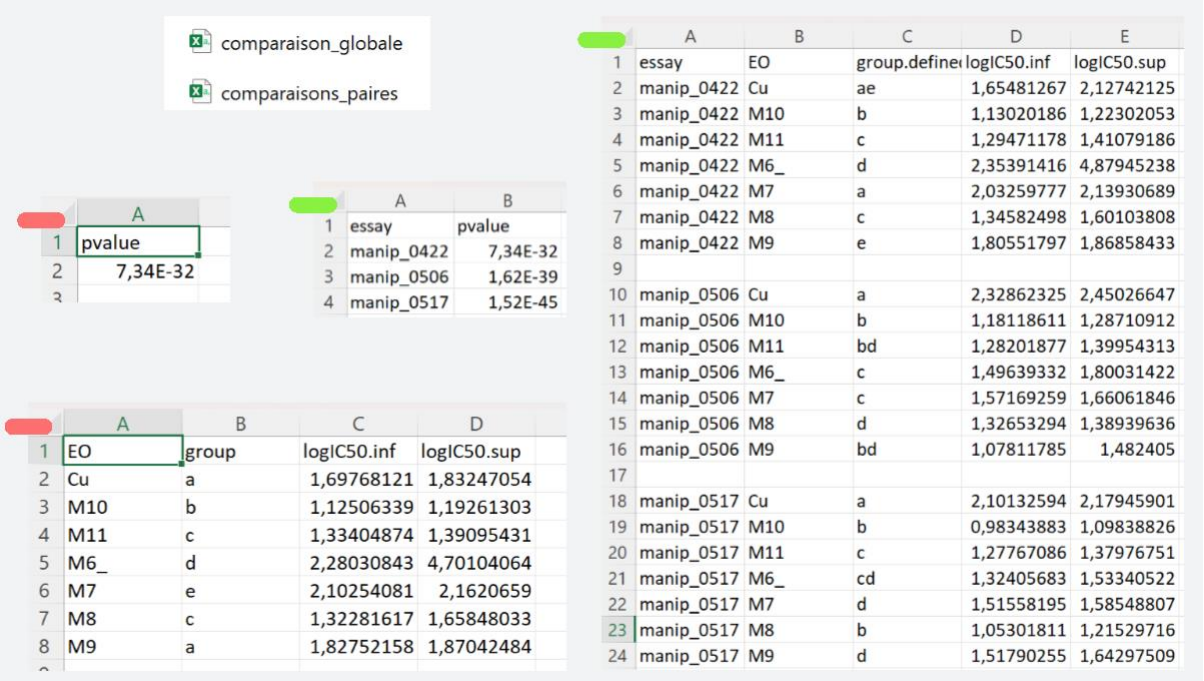


Figure 16 - Comparaison des CI50



	A
1	pvalue
2	7,34E-32

	A	B
1	essay	pvalue
2	manip_0422	7,34E-32
3	manip_0506	1,62E-39
4	manip_0517	1,52E-45

	A	B	C	D
1	EO	group	logIC50.inf	logIC50.sup
2	Cu	a	1,69768121	1,83247054
3	M10	b	1,12506339	1,19261303
4	M11	c	1,33404874	1,39095431
5	M6_	d	2,28030843	4,70104064
6	M7	e	2,10254081	2,1620659
7	M8	c	1,32281617	1,65848033
8	M9	a	1,82752158	1,87042484

	A	B	C	D	E
1	essay	EO	group.define	logIC50.inf	logIC50.sup
2	manip_0422	Cu	ae	1,65481267	2,12742125
3	manip_0422	M10	b	1,13020186	1,22302053
4	manip_0422	M11	c	1,29471178	1,41079186
5	manip_0422	M6_	d	2,35391416	4,87945238
6	manip_0422	M7	a	2,03259777	2,13930689
7	manip_0422	M8	c	1,34582498	1,60103808
8	manip_0422	M9	e	1,80551797	1,86858433
9					
10	manip_0506	Cu	a	2,32862325	2,45026647
11	manip_0506	M10	b	1,18118611	1,28710912
12	manip_0506	M11	bd	1,28201877	1,39954313
13	manip_0506	M6_	c	1,49639332	1,80031422
14	manip_0506	M7	c	1,57169259	1,66061846
15	manip_0506	M8	d	1,32653294	1,38939636
16	manip_0506	M9	bd	1,07811785	1,482405
17					
18	manip_0517	Cu	a	2,10132594	2,17945901
19	manip_0517	M10	b	0,98343883	1,09838826
20	manip_0517	M11	c	1,27767086	1,37976751
21	manip_0517	M6_	cd	1,32405683	1,53340522
22	manip_0517	M7	d	1,51558195	1,58548807
23	manip_0517	M8	b	1,05301811	1,21529716
24	manip_0517	M9	d	1,51790255	1,64297509

Figure 17 - Résultats après comparaison des CI50

Pour finir, le résumé est effectué (Figure 18), ce morceau de code permet d'avoir un fichier tableur et un graphique (Figure 19) permettant de regrouper l'ensemble des informations les plus importantes pour les biologistes.

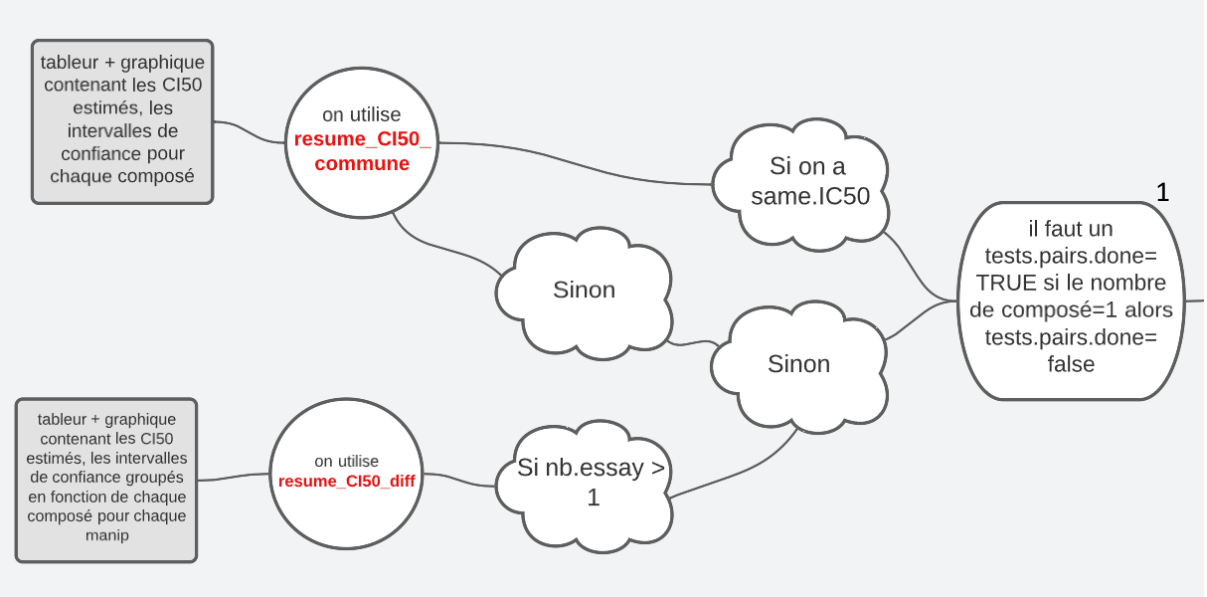


Figure 18 - Résumé

	A	B	C	D	E
1	EO	IC50.inf	IC50	IC50.sup	group
2	Cu	49,8518415	52,1388197	67,993992	a
3	M10	13,337161	15,0096053	15,5816352	b
4	M11	21,5798658	23,476728	24,601088	c
5	M6_	190,681441	213,603399	50238,9604	d
6	M7	126,631226	129,367648	145,233199	e
7	M8	21,0288814	33,6220517	45,5491552	c
8	M9	67,2235704	69,0758873	74,2035764	a

	A	B	C	D	E	F
1	essay	EO	IC50.inf	IC50	IC50.sup	group.def
2	manip_0422	Cu	45,1661079	52,1388197	134,097676	ae
3	manip_0422	M10	13,4959003	15,0096053	16,7116962	b
4	manip_0422	M11	19,7111416	23,476728	25,7508673	c
5	manip_0422	M6_	225,898923	213,603399	75762,1662	d
6	manip_0422	M7	107,794791	129,367648	137,818299	a
7	manip_0422	M8	22,1730267	33,6220517	39,9059889	c
8	manip_0422	M9	63,9025182	69,0758873	73,8897722	e
9	manip_0506	Cu	213,119531	242,991198	282,011272	a
10	manip_0506	M10	15,177006	18,2305328	19,3690855	b
11	manip_0506	M11	19,1433865	23,0639542	25,0924538	bd
12	manip_0506	M6_	31,361247	59,0671253	63,1414013	c
13	manip_0506	M7	37,2986055	41,6523938	45,7739577	c
14	manip_0506	M8	21,2096226	24,0926275	24,512994	d
15	manip_0506	M9	11,9706532	28,7705277	30,3672174	bd
16	manip_0517	Cu	126,277491	144,23817	151,167703	a
17	manip_0517	M10	9,62584414	11,5363805	12,5426199	b
18	manip_0517	M11	18,9526898	21,6550532	23,9754911	c
19	manip_0517	M6_	21,089041	29,752618	34,1511412	cd
20	manip_0517	M7	32,7779622	35,4910266	38,5024234	d
21	manip_0517	M8	11,2984303	12,1171402	16,4171271	b
22	manip_0517	M9	32,9535762	40,4892055	43,9516408	d

Figure 19 - Exemples de documents après le résumé

Explication de R Shiny

R Shiny est un package pour le langage de programmation R qui permet de créer des applications web interactives pour visualiser et analyser des données. Il se compose de deux éléments clés :

1. Interface utilisateur (UI) : L'UI est définie dans le fichier **ui.R**. Il s'agit de la mise en page et de l'apparence de l'application, y compris les éléments interactifs tels que les boutons, les curseurs et les sélections. L'UI est généralement construite en utilisant des fonctions Shiny telles que **fluidPage**, **titlePanel**, **sidebarLayout**, **sidebarPanel**, **mainPanel** et divers éléments d'entrée et de sortie.

2. Serveur : Le serveur est défini dans le fichier **server.R**. Il gère la logique de l'application et décrit comment les données sont traitées, analysées et présentées en fonction des interactions de l'utilisateur avec l'interface. Le serveur utilise des fonctions réactives pour définir des relations entre les entrées de l'utilisateur et les sorties de l'application.

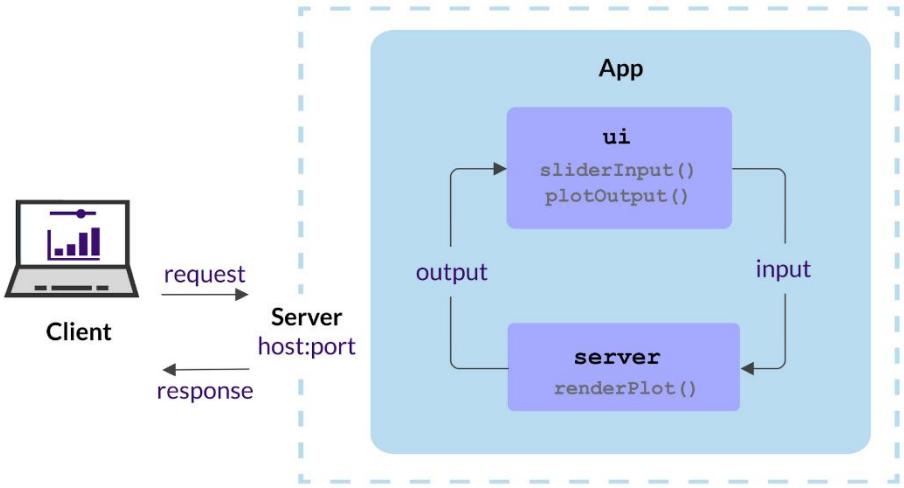


Figure 20 - Schéma du fonctionnement R Shiny (UI et Server)

https://hosting.analythium.io/content/images/2021/04/3-blank.png

Un schéma explicatif simplifié est le suivant : la partie UI gère les entrées de l'utilisateur, tandis que la partie SERVER traite ces entrées pour produire une ou plusieurs sorties, qui sont ensuite affichées à l'aide de la partie UI.

Les applications Shiny reposent sur le concept de réactivité. Cela signifie que lorsque l'utilisateur interagit avec l'application, les entrées sont mises à jour automatiquement, et les résultats sont recalculés et affichés sans avoir à rafraîchir manuellement la page.

Pour créer une application Shiny, vous devez définir à la fois l'interface utilisateur et la logique du serveur en utilisant les fonctions et les structures fournies par le package Shiny. Ensuite, vous pouvez exécuter l'application localement dans RStudio en cliquant sur "Run App" ou déployer l'application sur un serveur Shiny pour la partager avec d'autres utilisateurs.

En résumé, R Shiny permet de créer des applications web interactives pour analyser et visualiser des données en combinant une interface utilisateur personnalisable avec une logique serveur basée sur la réactivité.

Résultats

La conversion du code R en R Shiny a été réalisée en suivant plusieurs étapes clés. Tout d'abord, il fallait identifier les parties interactives du programme, c'est-à-dire les éléments qui nécessitaient une entrée de l'utilisateur ou produisaient une sortie dynamique. Ensuite, une étude de la documentation et les exemples de R Shiny était nécessaire pour se familiariser avec la manière d'intégrer ces éléments interactifs dans le nouveau code.

Au cours de la conversion, plusieurs défis se sont révélés. L'un d'eux concernait la réorganisation du code pour séparer l'interface utilisateur (UI) de la logique du serveur. Cela a nécessité de revoir certaines fonctions et de les réécrire pour qu'elles soient compatibles avec la structure de R Shiny. De plus, gérer les réactivités au sein de R Shiny a dû être appris. En effet, celle-ci permet d'assurer une interaction fluide entre les entrées de l'utilisateur et les sorties du programme.

Pour surmonter ces défis, les ressources en ligne étaient très utiles, notamment des tutoriels, des forums et la documentation officielle de R Shiny. Grâce à ces efforts, il a été possible de convertir le code R initial en une application R Shiny fonctionnelle et interactive. Dans la figure n°21 ci-dessous est présenté le programme utilisé par les chercheurs avant le projet. Ils devaient inscrire leurs paramètres dans le code directement. Dans la figure suivante n°22, c'est une présentation de la nouvelle interface créée au cours de ce projet.

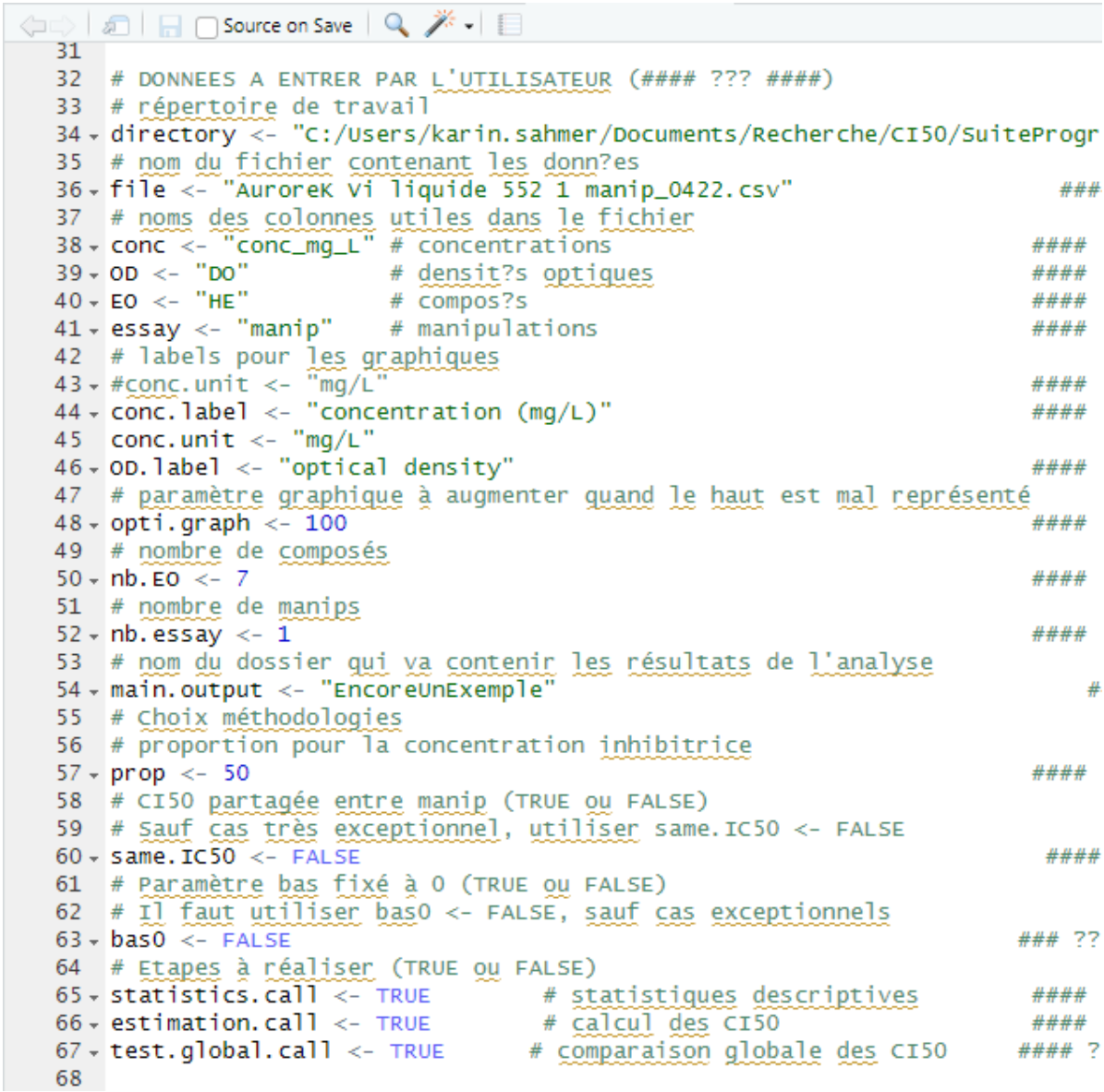


Figure 21 - Fichier Utilisateurs où écrire les entrées

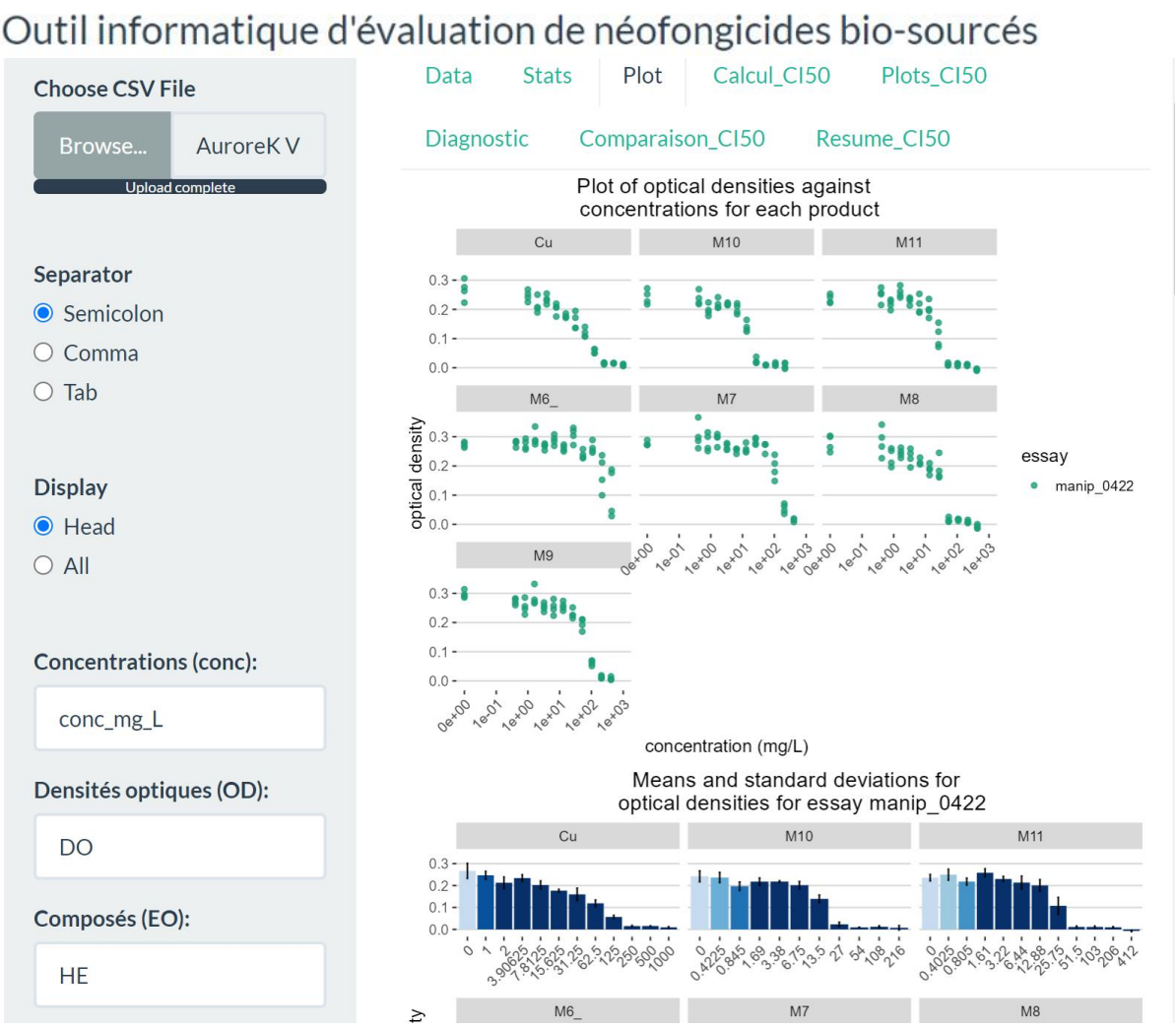


Figure 22 - Interface R Shiny avec les entrées à gauche

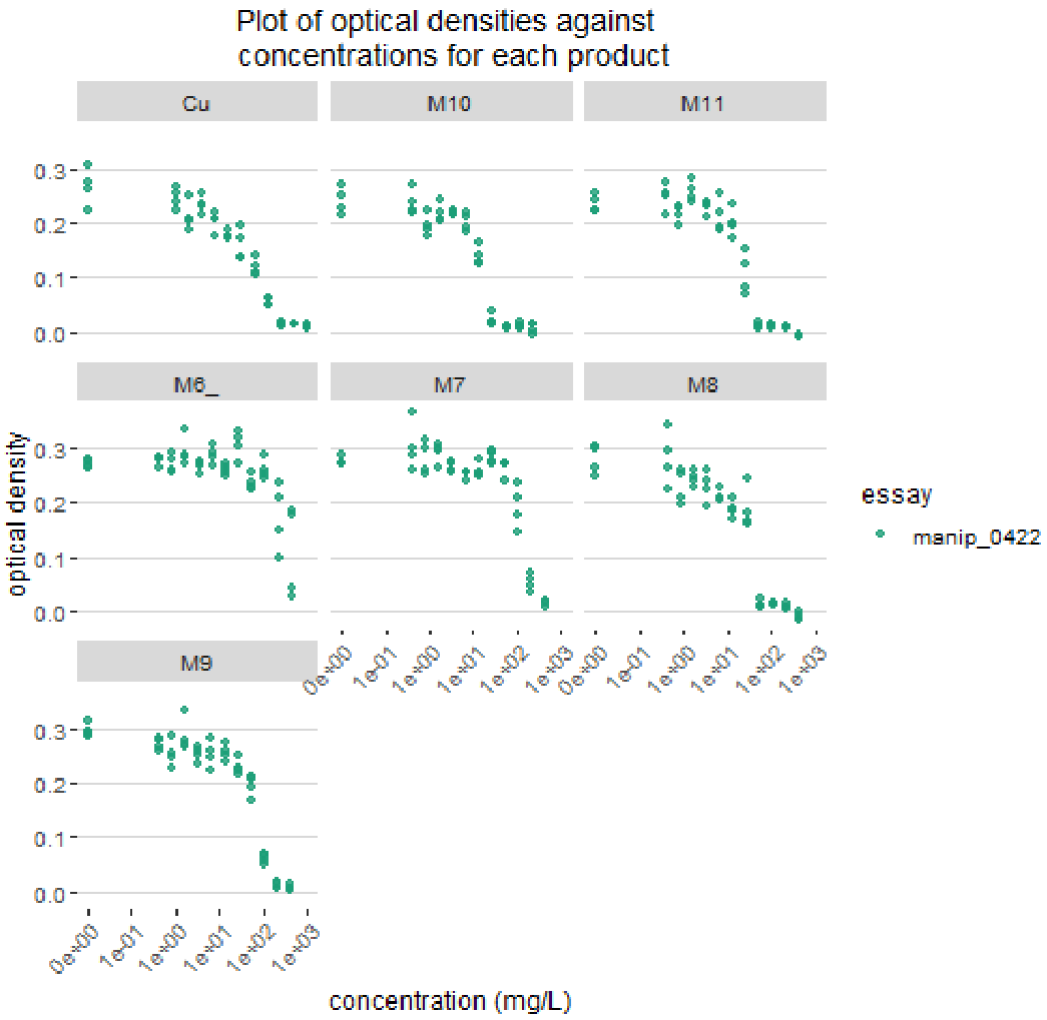


Figure 23 - Données obtenues à l'aide du programme initial

Des comparaisons ont été faites pour toutes les données. Cela prouve que les données obtenues par le programme initial et le programme R Shiny sont bien les mêmes, comme sur les figures 22 et 23.

Perspectives d'amélioration

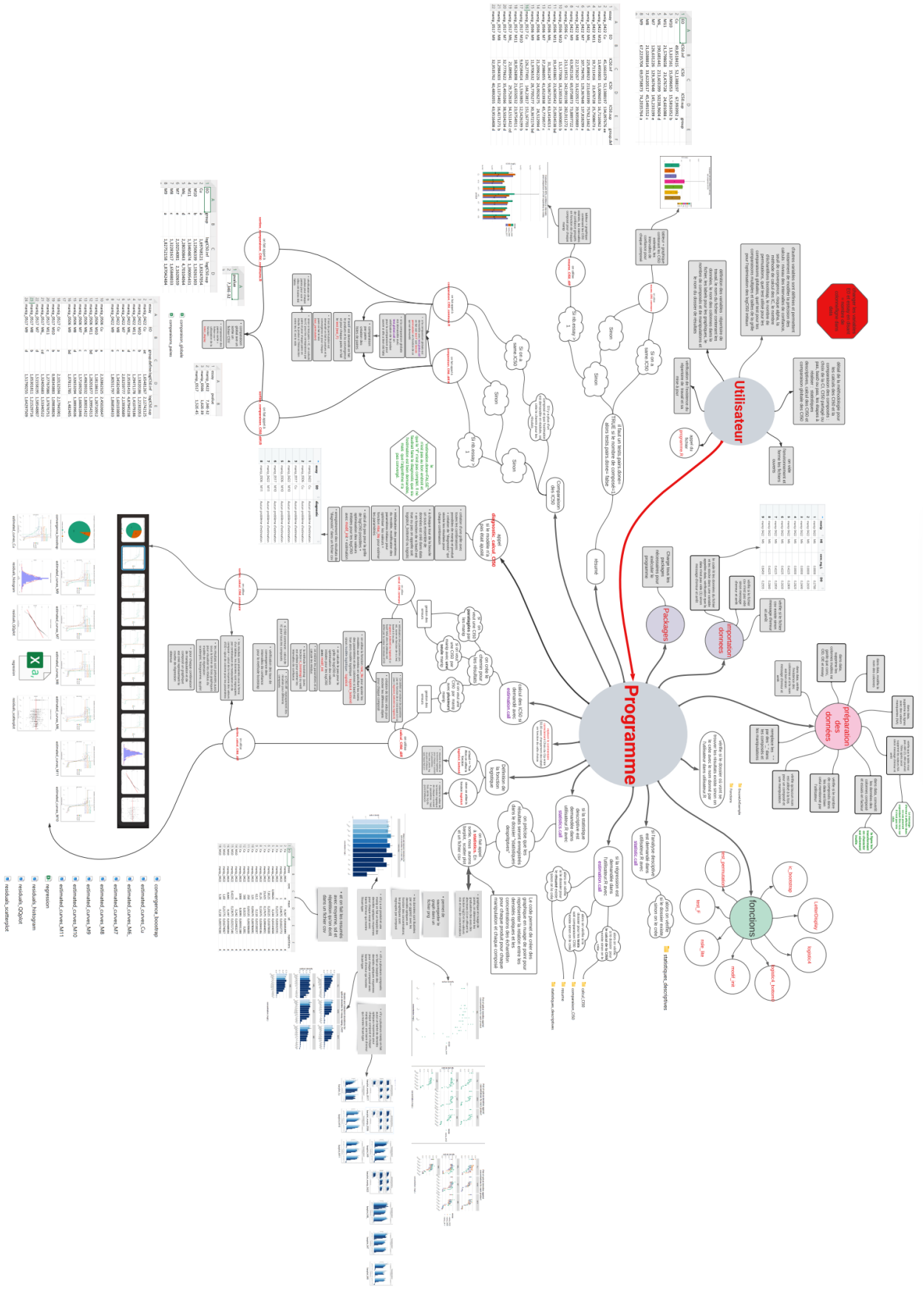
Il existe plusieurs pistes d'amélioration pour le programme. Tout d'abord, il est possible d'optimiser davantage le code afin d'améliorer les performances et la rapidité d'exécution. De plus, il pourrait être intéressant de proposer le choix des IC allant de 0 à 100, et pas uniquement IC50, pour offrir 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 intuitive 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 et permet de gagner du temps. Dans l'ensemble, ce projet est un exemple réussi de la conception d' une interface 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.

Annexe N°1


Programme
Utilisateur
Packages
importation données
préparation des données
fonctions
statistiques_descriptives

Carte mentale réalisée sur : https://lucid.app/lucidspark/59cc6d8a-0f24-445c-b4ff-05f0be2baef9/edit?viewport_loc=-16%2C4%2C2560%2C1298%2C0_0&invitationId=inv_94998b2d-890d-4fd0-bd52-c24377938974