



Rapport de stage

**Mise en place d'une application web
surveillant la croissance des coraux
en mésocosmes.**

Jordan Benrezkallah

Maître de stage : Philippe Grosjean

Encadrant de stage : Guyliann Engels

Promoteur : Aline Leonet et David Coornaert

HEH - Campus technique - Bloc 3 du cursus Bachelor en Biotechnique

Année académique 2018-2019





Rapport de stage

**Mise en place d'une application web
surveillant la croissance des coraux
en mésocosmes.**

Jordan Benrezkallah

Maître de stage : Philippe Grosjean

Encadrant de stage : Guyliann Engels

Promoteur : Aline Leonet et David Coornaert

HEH - Campus technique - Bloc 3 du cursus Bachelor en Biotechnique

Année académique 2018-2019



Remerciements

Je souhaite tout d'abord remercier mon professeur Aline Leonet qui m'a aidé à trouver mon stage, ensuite le professeur Philippe Grosjean qui m'a accueilli dans son laboratoire afin d'effectuer ce stage et qui a permis la mise en oeuvre de celui-ci.

Je voudrai également remercier les autres personnes du laboratoire d'Écologie Numérique des Milieux Aquatiques. Tout d'abord, je remercie Guyliann Engels qui m'a suivi de près depuis le début, qui a répondu aux nombreuses questions que je pouvais avoir et qui m'a accueilli dans son bureau. Je remercie Antoine, le technicien capable de maintenir les mésocosmes, il a de nombreuses fois pris le temps de m'aider. Je remercie également les mémorants, Madeleine et Rémy pour tous les échanges instructifs que j'ai pu avoir avec eux lors de ce stage. Je remercie aussi Nicolas pour les conseils qu'il m'a prodigués.

Un dernier remerciement revient à Raphaël qui a créé l'application sur laquelle je me suis inspiré pour commencer la mienne.

Résumé

Depuis plusieurs années, les scientifiques et le grand public s'intéressent fortement aux effets du changement climatique sur les écosystèmes. Les coraux forment des écosystèmes marins complexes parmi les plus riches en biodiversité. Le service d'EcoNum étudie l'éco-physiologie des scléractiniaires.

Les objectifs de ce stage sont multiples. Dans un premier temps il faudra acquérir des connaissances suffisantes du langage de programmation R. Ensuite, développer des outils permettant le monitoring des coraux. Des boutures de coraux devront être réalisées ainsi que des relevés réguliers de leurs masses tout au long de ce stage.

Un tableur en ligne sera conçu et contiendra l'ensemble des données relevées. À partir de cela, une application web mise en place générera plusieurs onglets. Le premier est une visualisation dynamique de la croissance des boutures, le second est un tableau interactif qui permet de trier par colonnes dans une plage donnée. Enfin, le dernier onglet une aide est disponible.

Summary

For several years, scientists and the general public have been strongly interested in the effects of climate change on ecosystems. Corals form complex marine ecosystems that are among the richest in biodiversity. The EcoNum department studies the ecophysiology of sclerotia.

The objectives of this internship are multiple. First of all, it will be necessary to acquire sufficient knowledge of the R programming language. Then, develop tools to monitor corals. Coral cuttings should be carried out as well as regular surveys of their masses throughout this course.

A spreadsheet will be designed and will contain all the data collected. From this, a web application set up will generate several tabs. The first is a dynamic visualization of the growth of cuttings, the second is an interactive table that allows you to sort by columns in a given range. Finally, the last tab a help is available.

Table des matières

1 Présentation	9
1.1 Présentation de l'entreprise	9
1.2 Présentation de l'équipe	11
1.2.1 Philippe Grosjean	11
1.2.2 Guyliann Engels	11
1.2.3 Antoine Batigny	12
1.2.4 Rémy Dugauquier	12
1.2.5 Madeleine Gille	12
2 Introduction	13
3 Objectifs du travail	17
3.1 Stage	17
3.2 Outils utilisés	18
4 Résultats et interprétations	19
4.1 Acquisition de données	19
4.1.1 Multiplication par bouturage	19
4.2 Outils monitorings	19
4.2.1 Masse immergée et masse squelettique	19
4.2.2 Tableur en ligne	21
4.3 Présentation de l'application Shiny	22
4.4 Difficultés rencontrées	27
4.4.1 R vs python	27
4.4.2 Shiny communication entre ui.R et server.R	27
4.5 Objectifs réalisés	28
5 Conclusion	30
6 Note	32
7 Annexe	33
7.1 ui.R	33
7.2 server.R	35
Bibliographie	42

Lexique

Aragonite : minéral composé de carbonate de calcium.

Carbonate de calcium (CaCO_3) : composant majeur du calcaire et constituant principal des coquilles d'animaux marins et du corail.

Cnidaire : groupe (embranchemet) d'espèces animales spécifiques du milieu aquatique.

Corail : animal de l'embranchemet des Cnidaires.

EcoNum : Service d'Écologie Numérique des Milieux Aquatiques.

Symbiose : association biologique, durable et réciprocement profitable, entre deux organismes vivants.

Monitoring : surveillance, contrôle.

Scléractiniaire : ordre principal des coraux durs.

Script : suite de commandes permettant d'automatiser une tâche.

Seriatopora hystrix Dana 1846 : espèce de scléractiniaires.

UMONS : université de Mons.

Zooxanthelle : algue unicellulaire pouvant vivre en symbiose avec le corail.

Chapitre 1

Présentation

1.1 Présentation de l'entreprise

Mon stage de fin d'études, se déroule à l'université de Mons dans le service d'Écologie Numérique des Milieux Aquatiques (abrégé en EcoNum) du département de Biologie.

L'Université de Mons (UMONS), est une université francophone implantée en Belgique, dans la province du Hainaut. Elle est constituée de 2 écoles et de 7 facultés, dont la faculté des Sciences.

Le Département de biologie de la faculté des Sciences est impliqué dans la formation des étudiants et dans la recherche.

Ce département comprend 5 services, dont le service d'Écologie Numérique des Milieux Aquatiques. Ce dernier étudie le comportement des écosystèmes aquatiques complexes, tels les communautés planctoniques et les récifs coralliens, face aux changements de leur environnement.

Le Service développe également des outils en science des données, y compris dans le domaine du data mining, des big data, et de la recherche reproductible. Il participe à des études sur les logiciels Open Source.

Le laboratoire d'EcoNum est situé sur le campus de la plaine de Nimy (A) (Fig. 1.1) dans le pentagone (1) (Fig. 1.2).

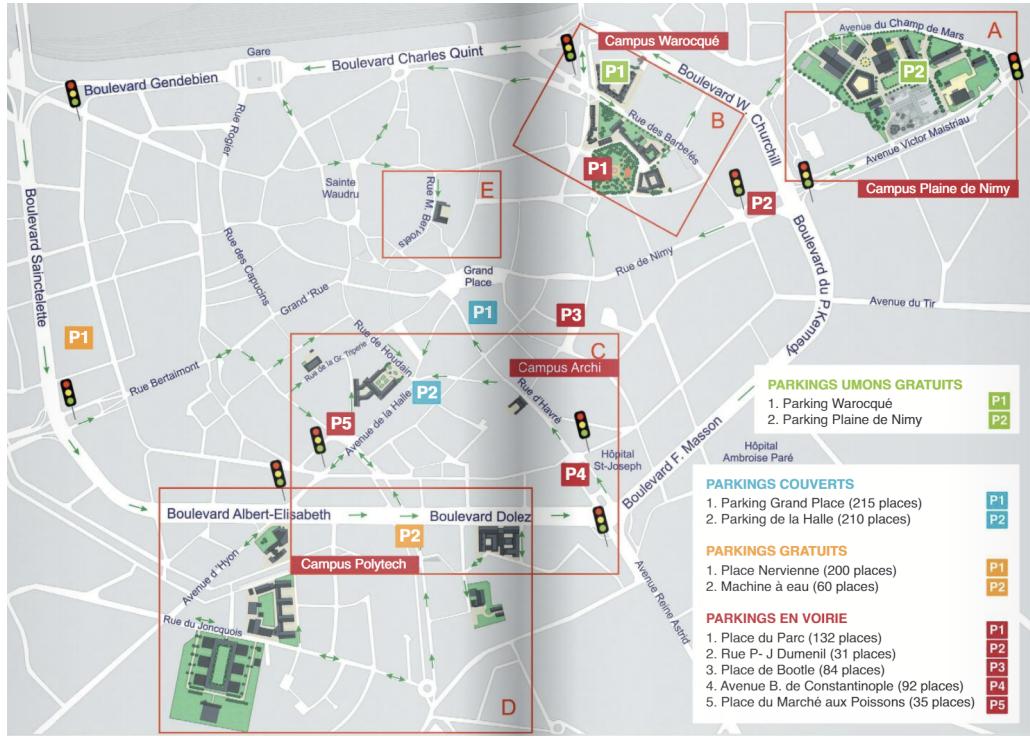


FIGURE 1.1 – Carte de la ville de Mons

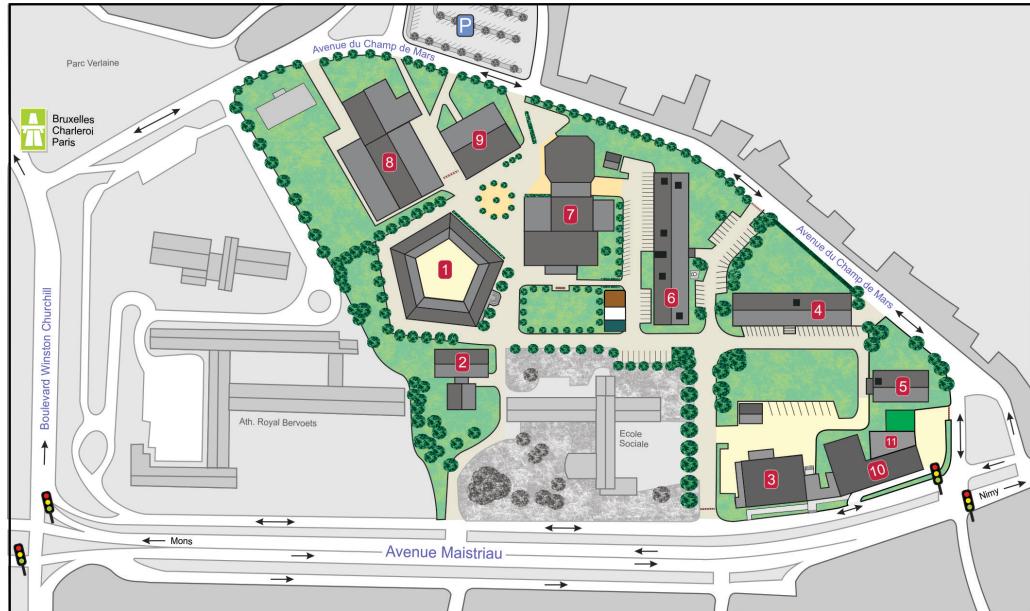


FIGURE 1.2 – Carte du campus de la plaine de Nimy

1.2 Présentation de l'équipe

1.2.1 Philippe Grosjean



Mon maître de stage est Monsieur Philippe Grosjean. Il mène plusieurs projets de recherches sur l'identification automatique du plancton par des algorithmes de *machine learning*, sur l'écophysiologie des scléractiniaires et sur le développement de logiciel pour l'écologie.

Il enseigne également la science des données biologiques, l'écologie marine, l'écophysiologie et l'océanographie générale aux étudiants en biologie.

Il développe des outils Open Source comme la *SciViews Box*, qui est une machine virtuelle contenant une suite de logiciels préconfiguré pour l'utilisation de ses étudiants et des chercheurs.

Il encadre 1 doctorant et 2 étudiants en masters.

1.2.2 Guyliann Engels



Guyliann Engels est chercheur et assistant au sein du service. Il effectue sa thèse sur l'écophysiologie du corail, où il utilise un mésocosme pour étudier les stress des coraux engendrés par la modification de leurs nutriments essentiels (composés azotés et phosphorés). Il utilise fréquemment les outils de statistiques R et RStudio (avec R Markdown, R Notebook).

Il encadre mon travail.

1.2.3 Antoine Batigny



Antoine Batigny est le technicien du service. Il gère principalement les mésocosmes.

1.2.4 Rémy Dugauquier



Rémy Dugauquier est en dernière année du master en biologie des organismes et écologie. Il réalise son T.F.E. sur l'écologie des organismes planctoniques en baie de Calvi, France.

1.2.5 Madeleine Gille



Madeleine Gille est étudiante en dernière année du master en biologie des organismes et écologie. Elle réalise son T.F.E. sur les effets d'un stress salin (hyper et hyposalin) sur *Seriatopora hystrix* (Dana, 1815).

Chapitre 2

Introduction

Les coraux sont des animaux de l'embranchement des cnidaires, tout comme les méduses. Les individus sont nommés « polypes » (Fig. 2.1). Au sein des cnidaires, 1609 espèces de coraux durs (scléractinaire hermatypique) forment les récifs coralliens (Fig. 2.2). La majorité des coraux scléractiniens vivent en symbiose avec des microalgues unicellulaires les zooxanthelles qui fournissent l'énergie nécessaire à la formation de leur squelette carbonaté de calcium (Fig. 2.3)(Veron 2000a).

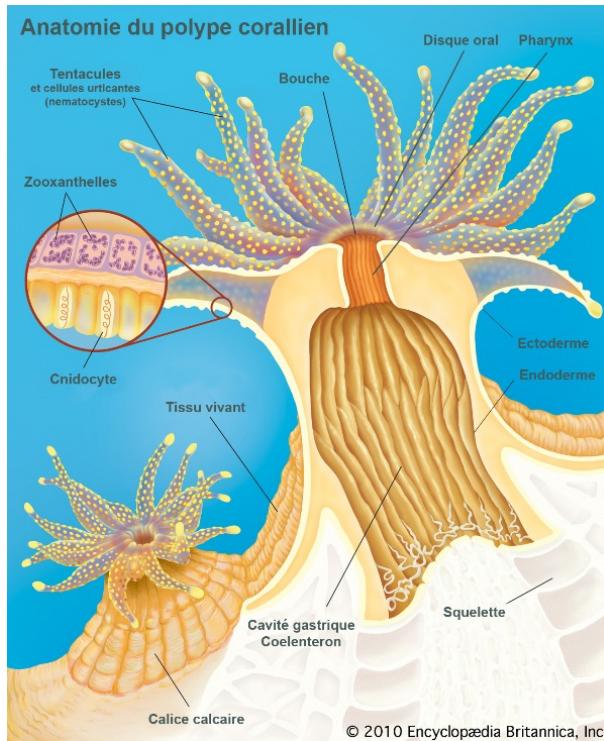


FIGURE 2.1 – Schéma d'un polype



FIGURE 2.2 – Récif corallien

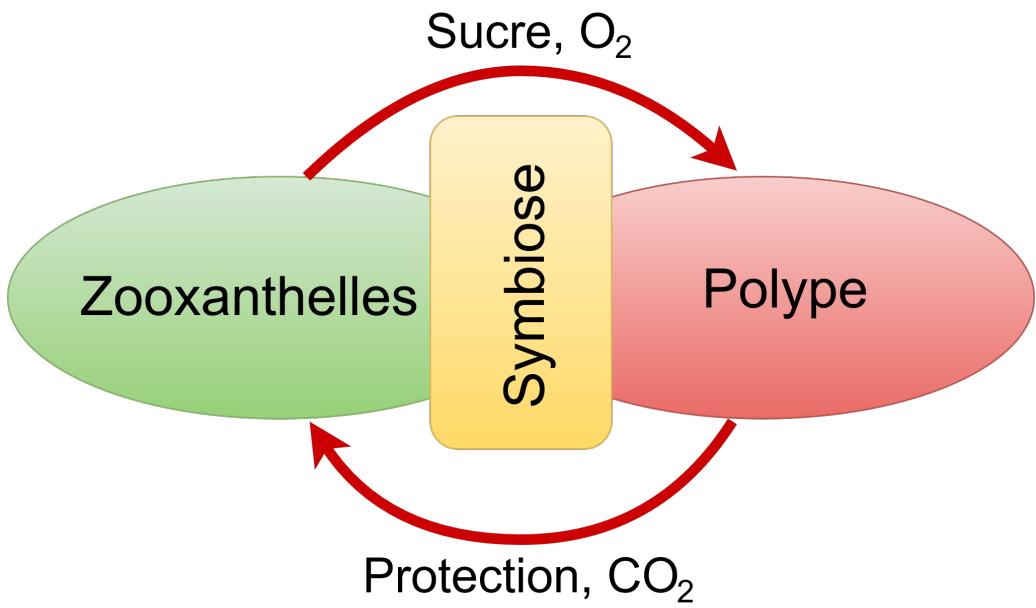


FIGURE 2.3 – Relation symbiotique entre les zooxanthelles et les polypes

Les récifs coralliens fournissent d'importantes niches écologiques à de nombreux animaux qui en sont dépendants. Il est donc crucial de les protéger.

En situation de stress, le corail peut expulser ses zooxanthelles, ce qui ne laisse paraître seulement la coloration blanche de son squelette. Ce blanchissement affaiblit considérablement le corail (Fig. 2.4). Divers facteurs peuvent stresser le corail : l'acidité, la salinité, la température, la pollution, etc (Veron 2000b).

Le processus du blanchissement

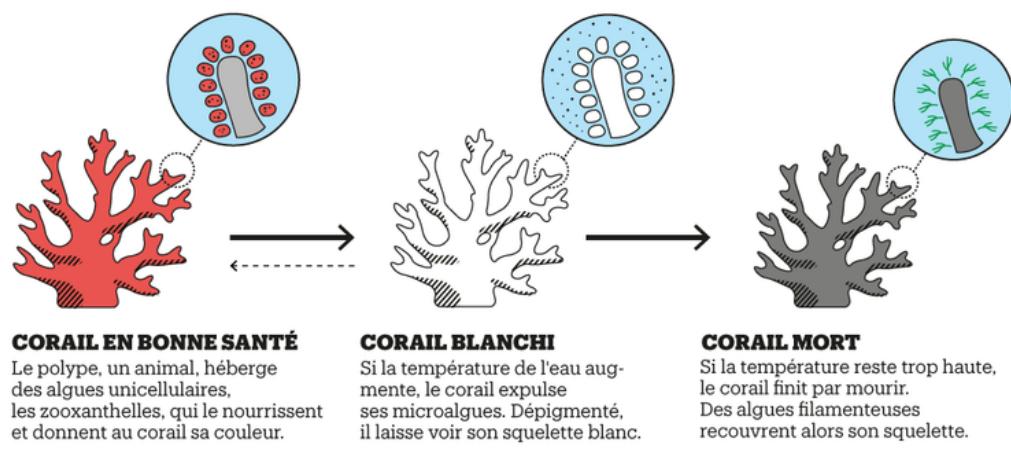


FIGURE 2.4 – Etape du blanchissement du corail

Le service d'écologie numérique des milieux aquatiques étudie en mésocosme les réponses écophysiologiques des coraux à divers stress sur *Seriatopora hystrix* Dana 1846 principalement.

Le précédent stagiaire, Raphaël Conotte a développé une application en local à partir du langage R permettant le monitoring des coraux (Fig. 2.5, Fig. 2.6).

Les objectifs du stage sont multiples. Premièrement, il va falloir apprendre le langage R et ses multiples packages. Deuxièmement, se baser sur le travail de Raphaël pour créer une version améliorée de celui-ci. Dernièrement, une mesure régulière des coraux sera effectuée tout au long du stage. Cela consiste à reproduire des échantillons de coraux qui seront ensuite pesé deux à trois fois par semaine, ce travail se fait en parallèle avec la partie informatique.

L'application créée pendant ce stage aura été complètement réécrite et simplifié. Le code de Raphaël étant beaucoup trop complexe pour un débutant dans R.

Coral Growth Dataset Plot

Options des tableaux

Choix des variables pour Data : ▾

Choix des variables pour Growth : ▾

Choix de l'ID à afficher :

head ▾

Date range :

2018-10-23 to 2019-02-04

Data Growth

Show 10 entries Search: ▾

	localisation	species	id	date	skel_weight
1	B0	s.hystrix	1	2018-10-23T14:15:00Z	2.53028859678207
2	B0	s.hystrix	2	2018-10-23T14:15:00Z	1.15152697062291
3	B0	s.hystrix	3	2018-10-23T14:15:00Z	0.658673427196304
4	B0	s.hystrix	4	2018-10-23T14:15:00Z	0.902797144968361
5	B0	s.hystrix	5	2018-10-23T14:15:00Z	1.4324995514549

Showing 1 to 5 of 5 entries

Previous 1 Next

FIGURE 2.5 – Application de Raphaël : onglet Dataset

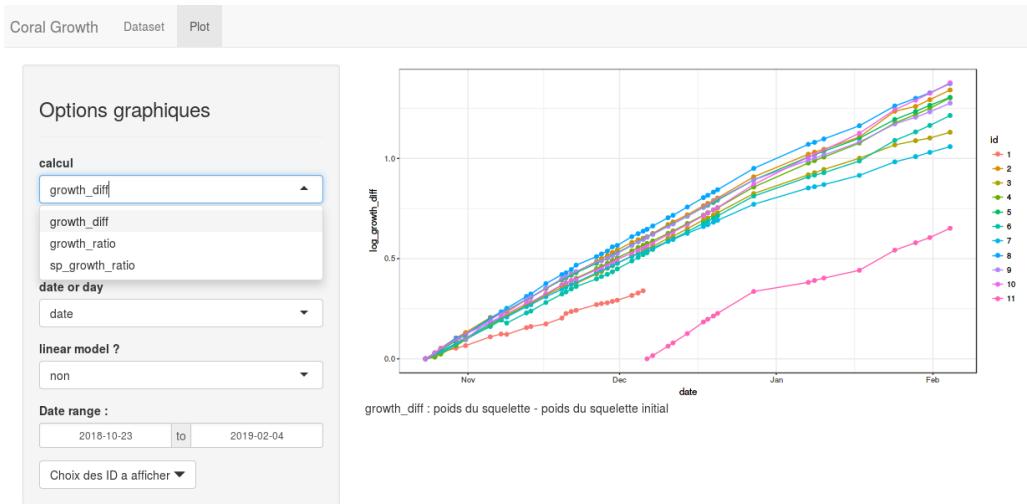


FIGURE 2.6 – Application de Raphaël : onglet Plot

Chapitre 3

Objectifs du travail

Le but du stage est de créer une application web via le package Shiny développé par RStudio sur R, qui suit l'évolution des coraux dans les mésocosmes. Les coraux seront utilisés dans des expériences par le laboratoire, il est donc nécessaire de visualiser leur croissance. L'application doit pouvoir être utilisée facilement par d'autres personnes à *posteriori*, il faut donc l'automatiser et anticiper les problèmes à venir.

Le stage se déroule en 2 parties, la première est une phase d'apprentissage, la deuxième est la création de l'application et l'implémentation d'outils pour le monitoring de la croissance des coraux.

3.1 Stage

La phase d'apprentissage comprend :

- Apprentissage du langage de programmation R, de ses packages et de l'environnement RStudio.

La phase de création d'outils comprend :

- L'acquisition des données de croissance régulière des coraux.
- La réalisation d'une application web Shiny, surveillant la croissance (monitoring) des coraux de l'espèce *S. hystrix*.

3.2 Outils utilisés

Les outils utilisés sont :

- La machine virtuelle *SciViews Box*, contenant un Linux (Xubuntu), R, RStudio et les paquets nécessaires pré-installés.
- Les langages de programmation : R.
- Les paquets : Shiny, tidyverse, ggplot2, dyplr, plotly, googlesheets, ect.
- Le service web GitHub.

Chapitre 4

Résultats et interprétations

4.1 Acquisition de données

4.1.1 Multiplication par bouturage

Dans le but d'acquérir de nouvelles données de croissance, on a utilisé une technique de multiplication asexuée : le bouturage. Cela consiste à séparer à l'aide d'une pince des branches de coraux. Le nombre de boutures s'élève à 84, toutes suspendu dans l'eau à l'aide de fil de pêche sur une règle qui porte un numéro d'identification propre à chacune (Fig. 4.1, Fig. 4.2).

4.2 Outils monitorings

4.2.1 Masse immergée et masse squelettique

Pour évaluer la croissance des boutures de coraux, on utilise la masse squelettique. Pour l'obtenir sans détruire le corail, on mesure la masse immergée du corail dans l'eau de mer avec une balance munie d'un crochet (Fig. 4.3). Cette méthode de mesure est rapide et peu stressante pour les organismes. Après avoir mesuré la température et la salinité, on peut convertir la masse immergée en masse squelettique à l'aide de la formule ci-dessous mise au point par Jokiel *et al* (1978) :

$$m_{squelettique} = \frac{m_{immerge}}{1 - \frac{\rho_{eau}}{\rho_{squelettique}}} \quad (4.1)$$

ρ_{eau} est déterminé par l'équation d'état de l'eau de mer grâce à la mesure de la salinité

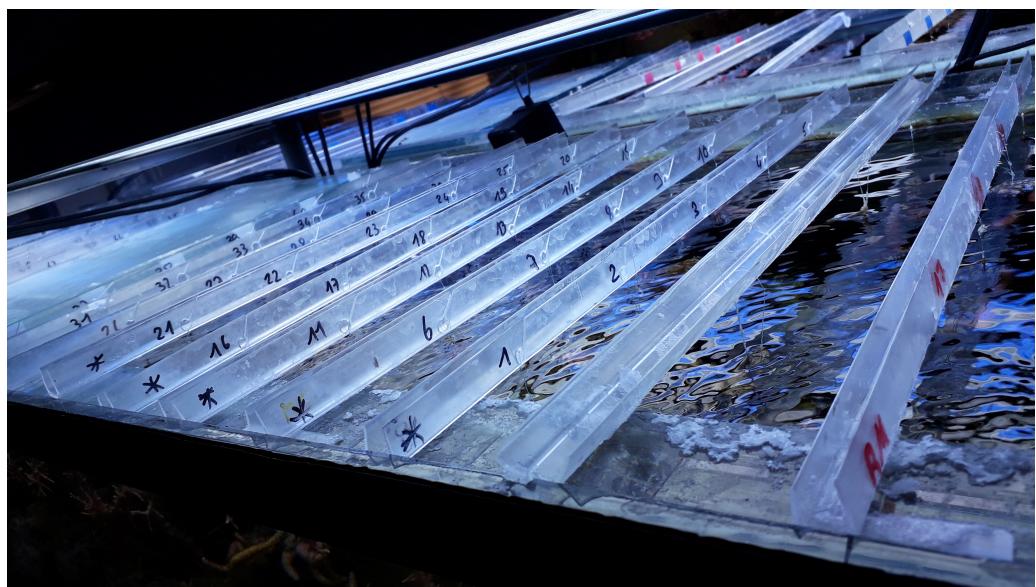


FIGURE 4.1 – Règle numérotée



FIGURE 4.2 – Boutures suspendues de *Seriatopora hystrix*

et de la température. Le $\rho_{squelettique}$ est la densité de l'aragonite (CaCO_3) du squelette du corail.

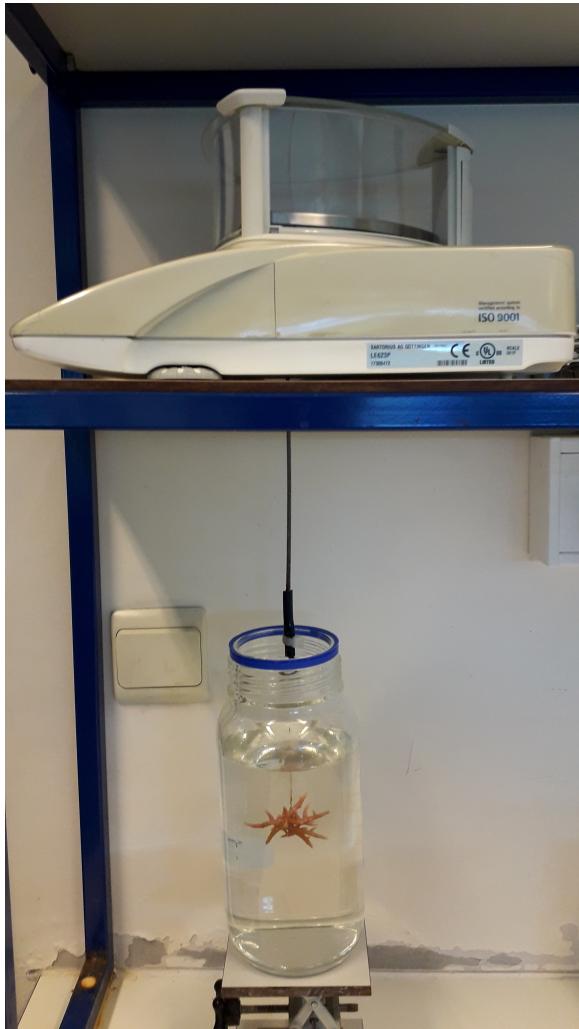


FIGURE 4.3 – Mesure du poids immergé d'une bouture

4.2.2 Tableur en ligne

Les mesures effectuées sur les coraux et les paramètres de l'eau des mésocosmes sont dans un premier temps notés dans un cahier de laboratoire. Il sera nécessaire de créer un nouveau tableau de donnée afin d'utiliser les données.

Au début, le tableur choisi était Excel, car c'est le logiciel le plus connu et que la HEH me permet d'utiliser une licence. Cela fonctionnait bien avec les fichiers en local. Malheureusement, aucun package ne permet d'utiliser Excel en ligne.

C'est avec Googlesheets qu'une solution fut trouvée.

Le tableur est en ligne cela permet à n'importe quelle personne de manipuler le tableau de données depuis n'importe quelle machine connectée à internet.

Afin d'éviter au maximum des erreurs d'encodages, des règles de mise en forme conditionnelles ont été créées pour mettre en évidence les cases non remplies, formater le type des cellules et mettre un dégradé de couleur suivant l'avancement des données (Fig. 4.4).

	A	B	C	D	E	F	G	H	I	J	K	
1	project	date	author	aqua	condition	species	id	weight	temperature	salinity	status	comment
2	mesocosm_monitoring	2019-02-11 14:20:00	jordan	B0	normal	s.hystrix	1	0.415	25.1	35.1	good	
3	mesocosm_monitoring	2019-02-11 14:20:00	jordan	B0	normal	s.hystrix	2	0.286	25.1	35.1	good	
4	mesocosm_monitoring	2019-02-11 14:20:00	jordan	B0	normal	s.hystrix	3		25.1	35.1	dead	
5	mesocosm_monitoring	2019-02-11 14:20:00	jordan	B0	normal	s.hystrix	4	1.059	25.1	35.1	good	
6	mesocosm_monitoring	2019-02-11 14:20:00	jordan	B0	normal	s.hystrix	5	0.677	25.1	35.1	good	
7	mesocosm_monitoring	2019-02-11 14:20:00	jordan	B0	normal	s.hystrix	6	0.394	25.1	35.1	good	
8	mesocosm_monitoring	2019-02-11 14:20:00	jordan	B0	normal	s.hystrix	7	0.795	25.1	35.1	good	
9	mesocosm_monitoring	2019-02-11 14:20:00	jordan	B0	normal	s.hystrix	8	0.228	25.1	35.1	good	
10	mesocosm_monitoring	2019-02-11 14:20:00	jordan	B0	normal	s.hystrix	9	0.508	25.1	35.1	good	
11	mesocosm_monitoring	2019-02-11 14:20:00	jordan	B0	normal	s.hystrix	10	0.929	25.1	35.1	good	
12	mesocosm_monitoring	2019-02-11 14:20:00	jordan	B0	normal	s.hystrix	11	0.519	25.1	35.1	good	
13	mesocosm_monitoring	2019-02-11 14:20:00	jordan	B0	normal	s.hystrix	12	1.088	25.1	35.1	good	
14	mesocosm_monitoring	2019-02-11 14:20:00	jordan	B0	normal	s.hystrix	13	0.603	25.1	35.1	good	
15	mesocosm_monitoring	2019-02-11 14:20:00	jordan	B0	normal	s.hystrix	14	0.224	25.1	35.1	good	
16	mesocosm_monitoring	2019-02-11 14:20:00	jordan	B0	normal	s.hystrix	15		25.1	35.1	dead	

FIGURE 4.4 – Tableur en ligne Google Sheets

Le tableur est divisé en 12 colonnes :

- project : différencie chaque expérience réalisée, généralement on préférera recréer un nouveau tableur pour chacune des expériences
- date : date et heure à laquelle les relevés de mesures ont été pris
- author : nom de la personne ayant encodé dans le tableur
- aqua : nom du mésocosme où la bouture a été prélevé
- condition : condition spécifique appliquée à la bouture (exemple : stress hypersalin)
- species : nom de l'espèce mesurée
- id : numéro de la bouture mesurée
- weight : masse immergée mesurée
- temperature : température de l'eau de mer
- salinity : salinité de l'eau de mer
- status : état de santé de la bouture
- comment : commentaire

4.3 Présentation de l'application Shiny

L'application est divisée en deux fichiers, une partie “ui” (User Interface), c'est la partie qui affiche les éléments graphiques de l'interface Shiny à l'utilisateur et une partie “server”, qui contient toutes les commandes R qui s'opère côté serveur.

Il est possible mettre l'intégralité du code dans un seul fichier app.R. Cependant, j'ai divisé mon script en deux fichiers ui.R et server.R pour plus de clarté (voir partie annexe).

Mon application présente 3 onglets, le premier créer un graphique interactif (Fig. 4.5).

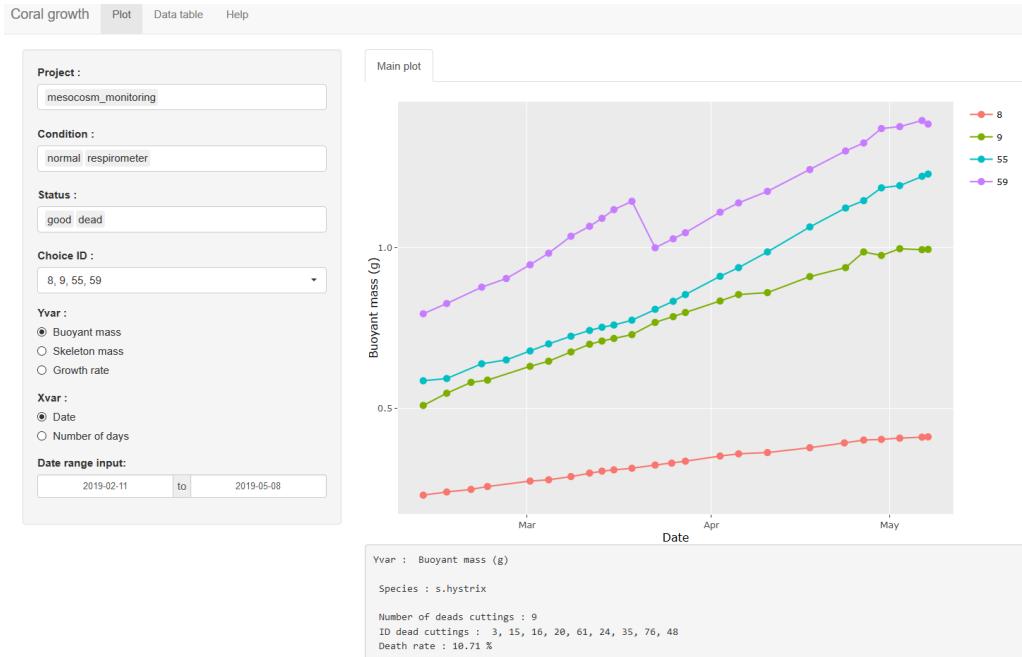


FIGURE 4.5 – Application Shiny : onglet "Plot"

Par défaut, le graphique utilise en ordonnée la masse immergée des boutures et en abscisse la date de la mesure. Les boutures sélectionnées sont peu nombreuses pour l'exemple, mais il est possible de toutes les sélectionner.

Différents paramètres peuvent modifier le graphique (Fig. 4.6).

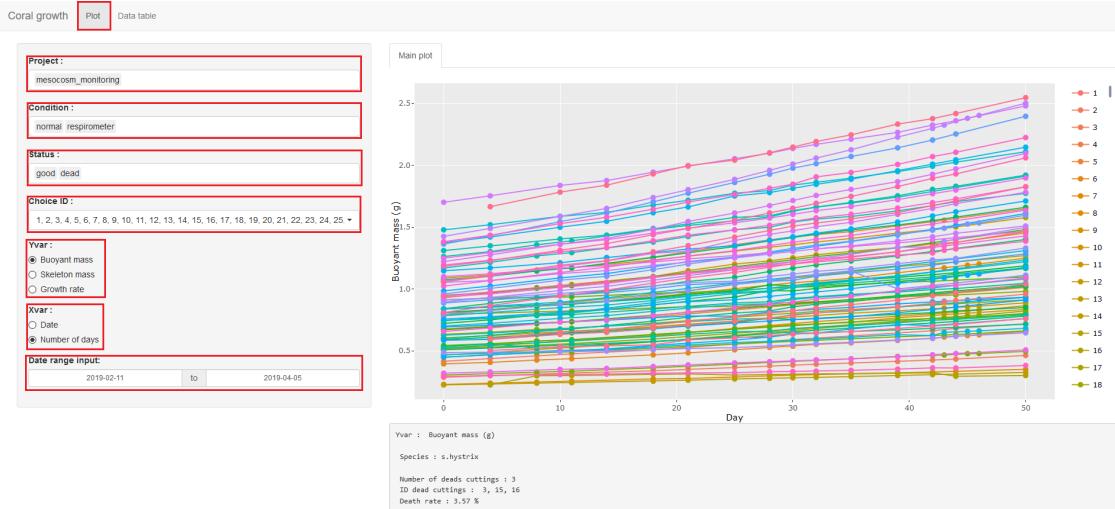


FIGURE 4.6 – Application Shiny : paramètres

En ordonnée, on peut choisir :

- la masse immergée
- la masse squelettique
- le taux de croissance (Fig. 4.7)

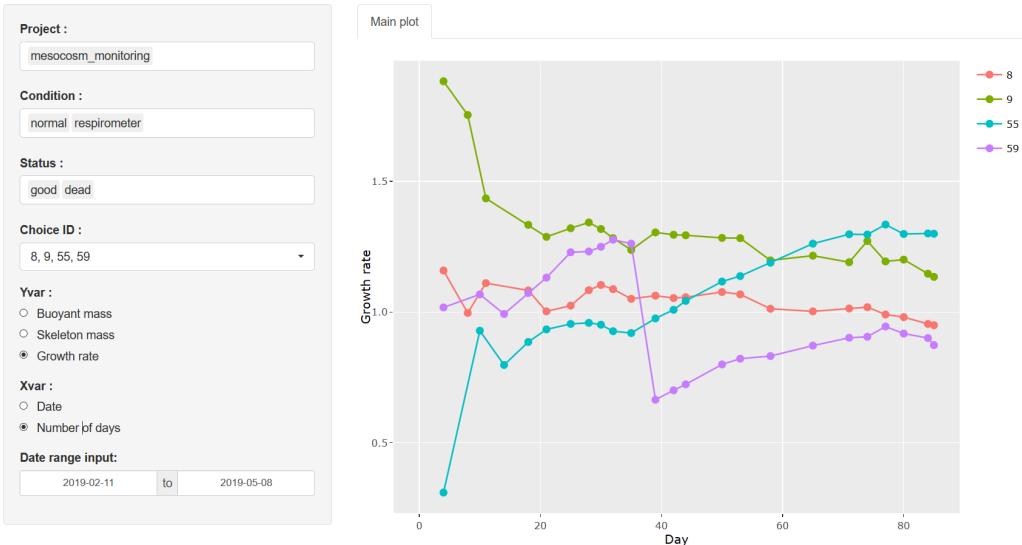


FIGURE 4.7 – Application Shiny : taux de croissance

En abscisse, on peut choisir :

- la date de la mesure
- le nombre de jours écoulés depuis la première mesure

Il est également possible de restreindre la période de temps (option *Date range input*).

Il est aussi possible de sélectionner les ID dans un menu déroulant, on peut également désélectionner les lignes en cliquant sur le numéro associé à la couleur de l'ID à droite de l'écran (Fig. 4.8, Fig. 4.9).

La fenêtre du menu déroulant a maintenant une taille adaptée et ne dépasse pas du cadre. Il permet de sélectionner ou de désélectionner toutes les boutures en deux cliques ou de sélectionner/désélectionner les boutures unes à unes.



FIGURE 4.8 – Application Shiny : menu déroulant

En passant le curseur sur les points du graphique, on peut obtenir quelques informations supplémentaires (Fig. 4.10).

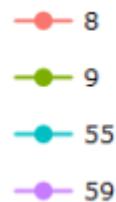


FIGURE 4.9 – Application Shiny : affichage interactif

En bas du graphique, des informations supplémentaires sont données (Fig. 4.11) :

- Yvar : l'ordonnée du graphique



FIGURE 4.10 – Application Shiny : information via le curseur

- Species : l'espèce des boutures
- Number of deads cuttings : le nombre de boutures mortes
- ID dead cuttings : l'ID des boutures mortes
- Death rate : le taux de mortalité

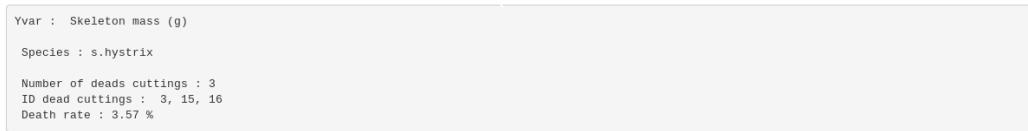


FIGURE 4.11 – Application Shiny : informations supplémentaires

Le deuxième onglet contient le tableau de donnée où de nouvelles colonnes ont été calculées, il y a l'ajout de la masse squelettique et du “ratio” qui correspond au taux de croissance (Fig. 4.12).

Coral growth														
Plot														
Data table														
Help														
Table														
Show 10 entries														
Search:														
project	date	author	aqua	condition	species	id	weight	temperature	salinity	status	comment	skw	delta_date	ratio
All	All	All	All	All	All	All	All	All	All	All	All	All	All	
2011	mesocosm_monitoring	2019-05-07T10:05:00Z	madeleine	B0	normal	s.hystrix	51	0.224	31.79	42.3	good	3.276	85	1.012
2012	mesocosm_monitoring	2019-05-07T10:05:00Z	madeleine	B0	normal	s.hystrix	52	28	good	2.289	85	1.143		
2013	mesocosm_monitoring	2019-05-07T10:05:00Z	madeleine	B0	normal	s.hystrix	53	2.052	24	34.4	good	3.153	85	1.28
2014	mesocosm_monitoring	2019-05-07T10:05:00Z	madeleine	B0	normal	s.hystrix	54	1.092	25.8	28	good	1.673	85	0.673
2015	mesocosm_monitoring	2019-05-07T10:05:00Z	madeleine	B0	normal	s.hystrix	55	1.23	24	34.4	good	1.89	85	1.3
2016	mesocosm_monitoring	2019-05-07T10:05:00Z	madeleine	B0	normal	s.hystrix	56	1.191	24	34.4	good	1.83	85	0.404
2017	mesocosm_monitoring	2019-05-07T10:05:00Z	madeleine	B0	normal	s.hystrix	57	2.822	25.5	42.3	good	4.349	85	1.268
2018	mesocosm_monitoring	2019-05-07T10:05:00Z	madeleine	B0	normal	s.hystrix	58	2.027	24	34.4	good	3.115	85	1.312
2019	mesocosm_monitoring	2019-05-07T10:05:00Z	madeleine	B0	normal	s.hystrix	59	1.386	25.8	28	good	2.124	85	0.874
2020	mesocosm_monitoring	2019-05-07T10:05:00Z	madeleine	B0	normal	s.hystrix	60	1.355	25.8	28	good	2.076	85	0.572

FIGURE 4.12 – Application Shiny : tableau de donnée

Le dernier onglet contient une page d'aide (Fig. 4.13). Cette documentation permettra aux utilisateurs de comprendre comment utiliser l'application et permettra aussi de comprendre comment fonctionne le code. Il est important de documenter son travail si l'on veut qu'il puisse être réutilisé par la suite.

FIGURE 4.13 – Application Shiny : onglet aide

4.4 Difficultés rencontrées

4.4.1 R vs python

La principale difficulté rencontrée au début est de passer de l'apprentissage du langage de programmation *python* à *R*. Ce sont tous les deux des langages de programmation interprétés qui peuvent être utilisés dans le domaine du traitement de donnée et de création d'application web. *python* a été créé pour faire de la programmation informatique généraliste, il est utilisé dans de larges domaines par des informaticiens. A l'inverse, *R* est dédié aux analyses statistiques, plutôt utilisées par des spécialistes ou des scientifiques.

Dans le domaine du *data scientist*, *R* et *python* sont couramment employé.

4.4.2 Shiny communication entre ui.R et server.R

Les applications web gérées par shiny utilisent deux fonctions communiquant entre elles **ui** et le **server**.

Le schéma de communication basique entre les deux scripts commence par la déclaration d'une variable *inputId = ma_variable* dans ui.R. Celui-ci est appelé dans server.R sous la forme *input\$ma_variable*, cette variable sera ensuite traitée dans un bloc de code délimiter par des crochets.

Shiny utilise du Javascript pour dynamiser l'interface de l'utilisateur sous une couche de code masqué, cette couche simplifie grandement le travail avec R. Si on sort du cadre de l'utilisation prévu par Shiny, on se heurte à de grands soucis de codage. Shiny restreint

donc, la communication entre les différents blocs de code. Dans certaines situations cela complique le travail, ce fut notamment le cas lors de la création du menu déroulant qui a besoin de connaître dans ui.R le nombre d'ID qui est nécessaire, sauf que la variable donnant cette information est dans server.R et Shiny permet difficilement de faire cela.

4.5 Objectifs réalisés

Les objectifs réalisés sont :

- Apprendre les bases du langage R et différents packages R.
- Maîtriser l'outil RStudio.
- Bouturer les coraux et relever leurs masses immergées.
- Créer un tableau de données en ligne contenant les données nécessaires.
- Créer une application web exploitant en continue les données du tableau en ligne.
- Réaliser un code simple et documenté.
- Utiliser le service web Github afin de réaliser un travail professionnel.

L'outil insight de Github permet de visualiser le travail des différents contributeurs sur un même projet. On peut constater le travail réalisé pendant le stage (Fig. 4.14).

Nov 18, 2018 – May 8, 2019

Contributions: Commits ▾

Contributions to master, excluding merge commits

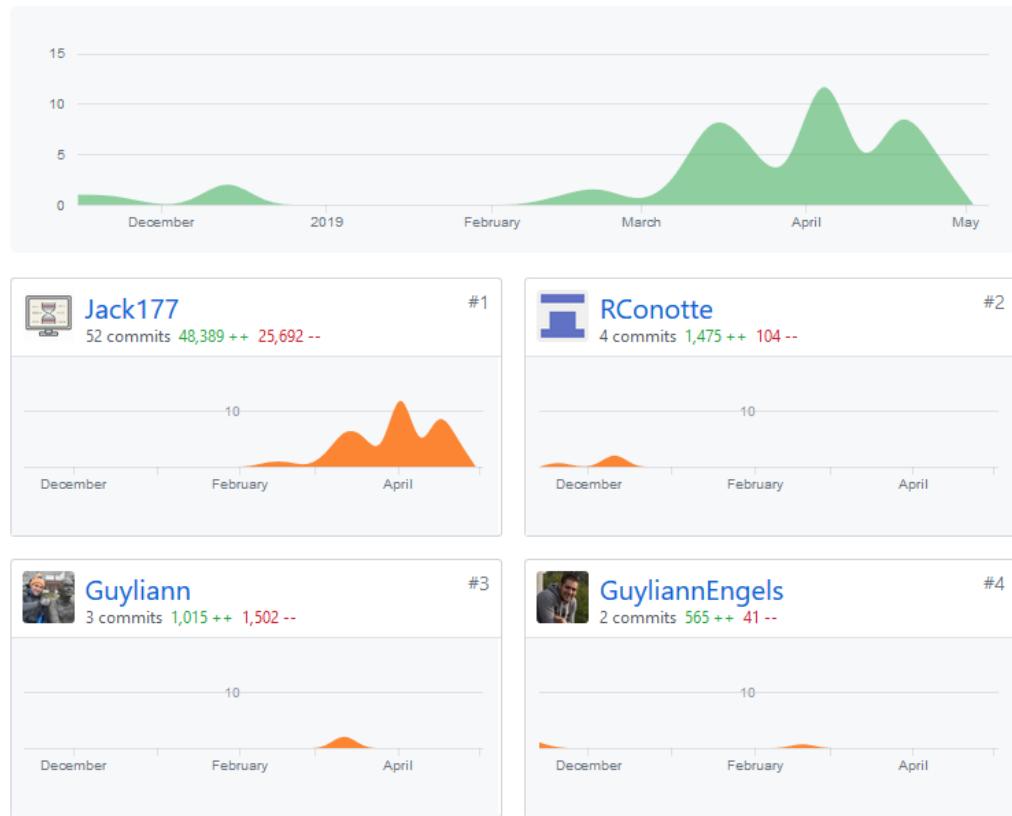


FIGURE 4.14 – Contributeur de l'application

Chapitre 5

Conclusion

Le point de départ de ce stage est l’application précédente effectuée par Raphaël Conotte. Pour comprendre son travail, il a fallu dans un premier temps apprendre le langage de programmation R et ses composants. Deuxièrement, il a fallu lire et comprendre son code ainsi que le jeu de données qu’il a utilisé. Dans le même temps, j’ai appris à multiplier du corail, les peser et créer mon propre jeu de données.

On m’a donné la possibilité de soit repartir du code précédent, soit de tout réécrire. C’est le second choix que j’ai pris, car le code de Raphaël étant encore aujourd’hui toujours trop complexe à mes yeux, je pense qu’il a voulu tester différentes manières de faire.

Le début de mon code était donc un fichier vide que j’ai rempli au fur et à mesure, en m’inspirant parfois du code de Raphaël. Je pense avoir rendu un code beaucoup plus simple à lire et mieux documenté. L’application que j’ai réalisée se démarque de celle de Raphaël. En plus de trier la base de données par projet, elle permet également de combiner les tris par condition et par l’état des boutures.

Contrairement à l’application précédente qui ne contenait que 6 boutures, mon travail en compte 84. Dans le but de faciliter la sélection de celles-ci, le menu déroulant a été perfectionné.

Le tableau de données présenté dans l’onglet “Data table” a été simplifié, mais il permet de faire un tri sur chacune des colonnes à l’aide d’un curseur de déplacement “slider”.

L’application est actuellement en ligne et est disponible à l’adresse suivante : <https://jack177.shinyapps.io/coralgrowth/>

Une documentation au format *bookdown* a été écrite pour les utilisateurs du laboratoire, mais aussi pour l’informaticien qui souhaiterait améliorer le code. En effet, le travail peut-être amélioré. On pourrait ajouter plusieurs jeux de données à partir d’une url, ajouter des visualisations différentes afin de permettre un autre point vue sur le jeu de données ou

encore de placer mon code dans un package afin d'être utilisé par n'importe qui utilisant R.

Le dernier onglet “Help” contient la documentation bookdown et le dépôt Github où se trouve la dernière version du code utilisée.

Il est également possible de scanner le *QR code* (Fig. 4.15) qui redirige vers l'application.



FIGURE 5.1 – QR code

D'un point de vue personnel, ce stage m'a permis de m'insérer dans une petite équipe où je me suis rapidement intégré. J'ai pu fréquenter un laboratoire de recherche et en apprendre beaucoup sur la vie marine et le milieu universitaire.

Le travail que j'ai effectué était passionnant. Je me suis rendu compte des différentes possibilités que peut offrir l'informatique à la biologie ainsi que la valeur ajoutée que j'ai pu proposer.

J'ai travaillé comme un professionnel sur un projet que j'ai vu se réaliser petit à petit, en utilisant des outils propres aux informaticiens comme Github. J'ai acquis des connaissances sur le traitement de données, découvert le langage de programmation *R* et appris à rédiger en *LaTeX/R Markdown* ce rapport.

Je souhaiterai poursuivre cette voie en apprenant à traiter des données avec *python* à l'aide des paquets *Numpy* et *Pandas* afin de le comparer à R et également apprendre le langage *javascript* utilisé par le package *shiny* qui rend les pages web dynamiques.

Chapitre 6

Note

Ce rapport en pdf interactif a été créé en R Markdown, il permet d'utiliser à la fois le langage LaTeX et R. LaTeX utilise des algorithmes pour améliorer la lecture, il automatise la plupart des paramètres comme la position des images. Il empêche parfois d'afficher l'agencement souhaité ou il renverse des images verticales pour utiliser au maximum l'espace de la page.

Le code en annexe a dû être retravaillé pour être correctement affiché.

La dernière version du code est disponible à l'adresse suivante :

https://github.com/EcoNum/coral_growth001

Chapitre 7

Annexe

7.1 ui.R

```
library(shiny)
library(shinyWidgets)
library(DT)
library(plotly)
library(shinythemes)
library(shinyWidgets)

shinyUI(
  navbarPage(
    #theme = shinytheme("slate"),
    title = "Coral growth", # Titre onglet 1
    ##### Onglet principal : Graphique
    tabPanel(title = "Plot",
      ## Sidebar : volet de gauche - Input
      sidebarPanel(
        uiOutput(outputId = "u_choice_project"),
        uiOutput(outputId = "u_choice_condition"),
        uiOutput(outputId = "u_choice_status"),
        uiOutput(outputId = "u_choice_id"), # Sélection des ID
        uiOutput(outputId = "u_choice_plot")#Sélection du graphique
        uiOutput(outputId = "u_choice_nbr_day"),# Sélection de Xvar
        uiOutput(outputId = "u_choice_date")      #Sélection date
      ),
      ## MainPanel : Volet de droite - Output
```

```

mainPanel(
  tabsetPanel(
    # Sous-onglet
    tabPanel(title = "Main plot",
      plotlyOutput(outputId = "u_plot",
                  height = "600px"),
      #sortie console
      verbatimTextOutput(outputId = "u_info"))
    #tabPanel(title = "Test plot")
  )
)
),
#### Onglet principal : Tableau de donnée
tabPanel("Data table",
  # Sidebar : Volet de gauche - Input
  # sidebarPanel(
  # ),
  # MainPanel : Volet de droite - Output
  mainPanel(
    tabsetPanel(
      tabPanel(title = "Table", DTOutput(outputId = "u_table"))
    )
  )
),
#### Onglet principal : Aide
tabPanel(title = "Help",
  fluidRow(
<<<<< HEAD
  column(12, includeMarkdown(
    "../analysis/Notebook/Notebook-Manuel.Rmd")))
=====

  uiOutput(outputId = "u_lien")
))
>>>>> 9c027f3a6d120b3a6a1d70f266e15f6a12e4e577
  )
)

```

7.2 server.R

```
library(shiny)
library(ggplot2)
library(lubridate)
library(tidyverse)
library(dplyr)
library(plotly)
library(shinyWidgets)
SciViews::R

### ----- Partie logique du serveur -----
shinyServer(function(input, output, session) {

  # Tableur de Madeleine :
  # coral_url <- "https://docs.google.com/spreadsheets/d/e/2PACX-1vTJLtfjj
  #UM4VK6aM177ly9GCKyMHFrFqQdsqjhJCtpe4DUGuZW0e2fZWB5xTZEz3WAcW08BVEBFfn2C
  #/pub?gid=0&single=true&output=csv"

  # Tableur de Jordan :
  coral_url <- "https://docs.google.com/spreadsheets/d/e/2PACX-1vSoBfvhztF
  gALk1fc1jBbYP03D-fRIEy7mu1DrHKZ--BXYZWHFxUujac_-gFStzM99p7CFQILT_eXcC/pu
  b?gid=0&single=true&output=csv"

  #Importation et format des colonnes
  read_csv(coral_url,
    col_types = cols( .default = col_character(),
                      date = col_datetime(),
                      weight = col_double(),
                      temperature = col_double(),
                      salinity = col_double() )) %>%
    mutate(.,
      project = factor(project), author = factor(author),
      aqua = factor(aqua),
      condition = factor(condition),
      species = factor(species),
      id = factor(id, levels = 1:length(unique(id))),
      status = factor(status)
    ) -> df
```

```

### Calcul du poids squelettique :
#a corriger : rho_aragonite
#P = Pression hydrostatique, elle vaut 0 a la surface
skeleton_weight <- function(S, T, P = 0,
                                buoyant_weight,
                                rho_aragonite = 2930){
  rho_water <- seacarb::rho(S = S, T = T , P = P)
  skl_wgt <- buoyant_weight / (1 - (rho_water / rho_aragonite))
  skl_wgt <- round(skl_wgt, digits = 3)
  return(skl_wgt)
}

# Ajout de la colonne du poids squelettique
df <- mutate(df,
             skw = skeleton_weight(S = salinity,
                                   T = temperature,
                                   buoyant_weight = weight))

# Nombre de ID different
nbr_id <- unique(df$id)

# Conditions
nbr_condition <- unique(df$condition)

# Projet
nbr_projet <- unique(df$project)

# Statut
nbr_status <- unique(df$status)

# Taux de croissance
df %>%
  group_by(., id) %>%
  arrange(., date) %>%
  mutate(.,
        delta_date = (as.numeric(difftime(date,
                                            date[1], units = "days"))),
        ratio = round(((skw-skw[1])/skw[1]/delta_date)*100,digits = 3),
        delta_date = round(delta_date, digits = 0)) %>%
  ungroup(.) -> df

### -----_Fin traitement du tableau de données__ ----- ###

```

```

#=====#
# ----- Selection des dates -----
output$u_choice_date <- renderUI({


  dateRangeInput(inputId = "s_choice_date",
                 label = 'Date range input: ',
                 start = min(df$date), end = max(df$date),
                 min = min(df$date), max = Sys.Date()
  )
})

# ----- Selection Xvar -----
output$u_choice_nbr_day <- renderUI({


  radioButtons(inputId = "s_choice_nbr_day",
              label = 'Xvar : ',
              choices = c("Date", "Number of days"),
              selected = "Number of days"
  )
})

#-----Selection id-----
output$u_choice_id <- renderUI({


  pickerInput(inputId = "s_choice_id",
              label = "Choice ID :",
              choices = nbr_id,
              options = list(`actions-box` = TRUE),
              multiple = T,
              selected = c(8, 9, 55, 9)
  )
})

# ----- Choix des ID -----
observe({
  print(input$s_choice_id)
})

#-----Choix graphique (variable y)-----
output$u_choice_plot <- renderUI({


  radioButtons(inputId = "s_choice_plot", label = "Yvar :",
              choices = c("Buoyant mass", "Skeleton mass",

```

```

        "Growth rate"),
selected = "Buoyant mass")
})

#-----Choix projet-----
output$u_choice_project <- renderUI({


  selectInput(inputId = "s_choice_project",
              label = "Project :",
              choices = nbr_projet,
              multiple = TRUE,
              selected = nbr_projet)
})

#-----Choix condition-----
output$u_choice_condition <- renderUI({


  selectInput(inputId = "s_choice_condition",
              label = "Condition :",
              choices = nbr_condition,
              multiple = TRUE,
              selected = nbr_condition)
})

#-----Choix statut-----
output$u_choice_status <- renderUI({


  selectInput(inputId = "s_choice_status",
              label = "Status :",
              choices = nbr_status,
              multiple = TRUE,
              selected = nbr_status)
})

####-----Output de mon graphique-----####
output$u_plot <- renderPlotly({


  # Filtre en fonction des choix
  df %>%
    filter(., 
          project %in% input$s_choice_project,

```

```

    condition %in% input$s_choice_condition,
    status %in% input$s_choice_status,
    date >= input$s_choice_date[1] & date <= input$s_choice_date[2],
    id %in% input$s_choice_id
) -> df

# Choix de la masse squelettique
if ("Skeleton mass" %in% input$s_choice_plot) {
  yvar = df$skw
  y_axis_name <- "Skeleton mass (g)"
}

# Choix de la masse immergée
if ("Buoyant mass" %in% input$s_choice_plot) {
  yvar = df$weight
  y_axis_name <- "Buoyant mass (g)"
}

# Choix du taux de croissance
if ("Growth rate" %in% input$s_choice_plot) {
  yvar = df$ratio
  y_axis_name <- "Growth rate"
}

# Choix par nombre de jour
if ("Number of days" %in% input$s_choice_nbr_day) {
  xvar = df$delta_date
  xlabel = "Day"
}

# Choix par date du jour
if ("Date" %in% input$s_choice_nbr_day) {
  xvar = df$date
  xlabel = "Date"
}

ggplot(df, aes(x = xvar, y = yvar, colour = id)) +
  geom_point(size = 2, show.legend = FALSE, na.rm = TRUE) +
  geom_line(show.legend = FALSE, na.rm = TRUE) +
  xlab(xlabel) + ylab(y_axis_name) -> p

p <- ggplotly(p, show.legend = FALSE)
}

```

```

#####-----Sortie console-----#####
output$u_info <- renderPrint({

  #Affichage de la formule utilisé
  formule <- ""

  if ("Buoyant mass" %in% input$s_choice_plot) {
    formule <- "Buoyant mass (g)"
  }
  if ("Skeleton mass" %in% input$s_choice_plot) {
    formule <- "Skeleton mass (g)"
  }
  if ("Growth rate" %in% input$s_choice_plot) {
    formule <- "Growth rate = ( (skeleton_mass_n - skeleton_mass_n-1) /
      skeleton_mass_n-1 ) / (time_n - time_n-1) * 100"
  }

  # Calculs boutures mortes
  nbr_dead <- as.numeric(count(unique(subset(df, status == "dead", id))))
  death_rate <- as.numeric(round
    ((nbr_dead / length(levels(nbr_id))) * 100,
     digits = 2))
  id_dead <- unique(subset(df, status == "dead", id))
  id_dead <- id_dead$id

  cat("Yvar : ", formule, "\n",
      "Species : ", as.character(unique(df$species)), "\n", "\n",
      "Number of deads cuttings : ", nbr_dead, "\n",
      "ID dead cuttings : ", paste(id_dead, collapse = ", "), "\n",
      "Death rate : ", death_rate, "%")
}

# -----Onglet tableau-----
output$u_table <- renderDT({
  datatable(df, filter = "top")
})

# Recuperation de l'ID du fichier ui.R
output$u_choice_table <- renderUI({

```

```

radioButtons(inputId = "s_choice_table", label = "Filtrer",
             choices = c("Yes", "No"),
             selected = "No")
})

output$u_subchoice_table <- renderUI({

  dropdown(
    radioButtons(inputId = "s_subchoice_table",
                 label = "by",
                 choices = c("skeleton weight", "growth rates"),
                 selected = c("skeleton weight")),
    width = "200px",
    size = "default",
    label = "Variable type",
    tooltip = tooltipOptions(placement = "right",
                            title = "Choice variable type")
  )
}

output$u_choice_var <- renderUI({

  numericInput(inputId = "s_choice_var",
               label = if (input$s_subchoice_table == "growth rates")
                 {"Growth rates higher than :"}
               else {"Skeleton weight higher than :"},
               value = 1)
}

output$u_lien <- renderUI({
  url1 <- a("Bookdown CoralGrowth", href = "https://econum.github.io/coral_growth001_book/")
  url2 <- a("Github repository", href = "https://github.com/EcoNum/coral_growth001")
  tagList(tags$h2( "Help page at the following adress : " ),
          tags$h3(url1),
          tags$h3(url2))
}

})

```

Bibliographie

- Veron, John Edward Norwood. 2000a. *Corals of the world, Volume 1*. Edited by Mary Stafford-Smith. doi :10.1007/s00044-012-0037-7.
- . 2000b. *Corals of the world, Volume 3*. Edited by Mary Stafford-Smith. doi :10.1007/s00044-012-0037-7.