

Rapport de stage partiel

**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 Bachelier en Biotechnique

Année académique 2018-2019

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

Remerciements

Je remercie le professeur Philippe Grosjean qui m'a accueillie 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'Ecologie Numérique des Milieux Aquatiques. Tout d'abord, je remercie Guyliann Engels qui m'a suivie depuis le début, qui a répondu aux nombreuses questions que je pouvais avoir et qui m'a soutenu tout au long de ce stage dans son bureau. Je remercie Antoine, le technicien qui arrive à maintenir tout les mésocosmes en équilibre, qui m'a de nombreuses fois pris le temps de m'aider. Je remercie également les mémorants, Madeleine et Rémy pour tout 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 prodigué.

Un dernier remerciement revient à Raphael qui a créé l'application sur laquelle je me suis basé 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. Dans le service d'EcoNum, on étudie les effets du stress salin sur les coraux.

Le but de ce stage est multiple, dans un premier temps il faudra acquérir des connaissances suffisantes du langage de programmation R et ensuite de développer des outils permettant le monitoring des coraux. Tout en faisant cela, des boutures de coraux devront être réalisées ainsi que des relevés réguliers de leurs masses.

Un tableau contiendra l'ensemble des données des relevés. À partir de cela, l'application web génère plusieurs onglets dont une visualisation dynamique de la croissance des boutures, un tableau interactif qui permet de trier par colonnes dans une plage donnée et dans le dernier onglet une aide est disponible.

Summary

For several years, scientists and mainstream have been strongly interested in the effects of climate change on ecosystems. Corals form complex marine ecosystems that are among the richest in biodiversity. In the EcoNum department, we study the effects of salt stress on corals.

The purpose of this internship is multiple, first it will be necessary to acquire sufficient knowledge of the R programming language and then to develop tools for coral monitoring. While doing this, coral cuttings should be carried out as well as regular surveys of their masses.

A spreadsheet will contain all the data from the surveys. From this, the web application generates several tabs including a dynamic visualization of the growth of cuttings, an interactive table that allows you to sort by columns in a given range and in the last tab a help is available.

Table des matières

1 Présentation	8
1.1 Présentation de l'entreprise	8
1.2 Présentation de l'équipe	10
1.2.1 Philippe Grosjean	10
1.2.2 Guyliann Engels	10
1.2.3 Antoine Batigny	11
1.2.4 Rémy Dugauquier	11
1.2.5 Madeleine Gille	12
2 Introduction	13
3 But	14
3.1 Stage	14
4 Analyse	15
4.1 Acquisition de données réelles	15
4.1.1 Multiplication par bouturage	15
4.2 Outils monitorings	15
4.2.1 Masse immergée et masse squelettique	15
4.2.2 Tableur	15
4.2.3 Application Shiny	17
4.3 Outils utilisés	19
4.4 Objectifs réalisés	19
4.5 Planning de travail	20
5 Communications interpersonnelles	21
5.1 Difficultés rencontrées	21
5.1.1 R vs python	21
5.1.2 Shiny communication entre ui.R et server.R	21
5.1.3 Apport au sein de l'entreprise	21
6 Note	23
7 Annexe	24
7.1 ui.R	24
7.2 server.R	25

Lexique

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

Corail : animal de l'embranchement des Cnidaires.

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

EcoNum : Service d'Écologie Numérique des Milieux Aquatiques.

UMons : Université de Mons.

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

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

Seriatopora hystrix Dana 1846 : espèce de scléractiniaires.

Scléractiniaire : ordre principal des coraux durs.

Monitoring : surveillance, contrôle.

Chapitre 1

Présentation

1.1 Présentation de l'entreprise

Mon stage de fin d'études, se déroule dans à l'université de l'UMons 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.

Le département de biologie de la faculté des Sciences comprend 5 services dont le service d'Écologie Numérique des Milieux Aquatiques. Ce dernier étudie les 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) (voir figure 1.1) dans le pentagone (1) (voir figure 1.2).

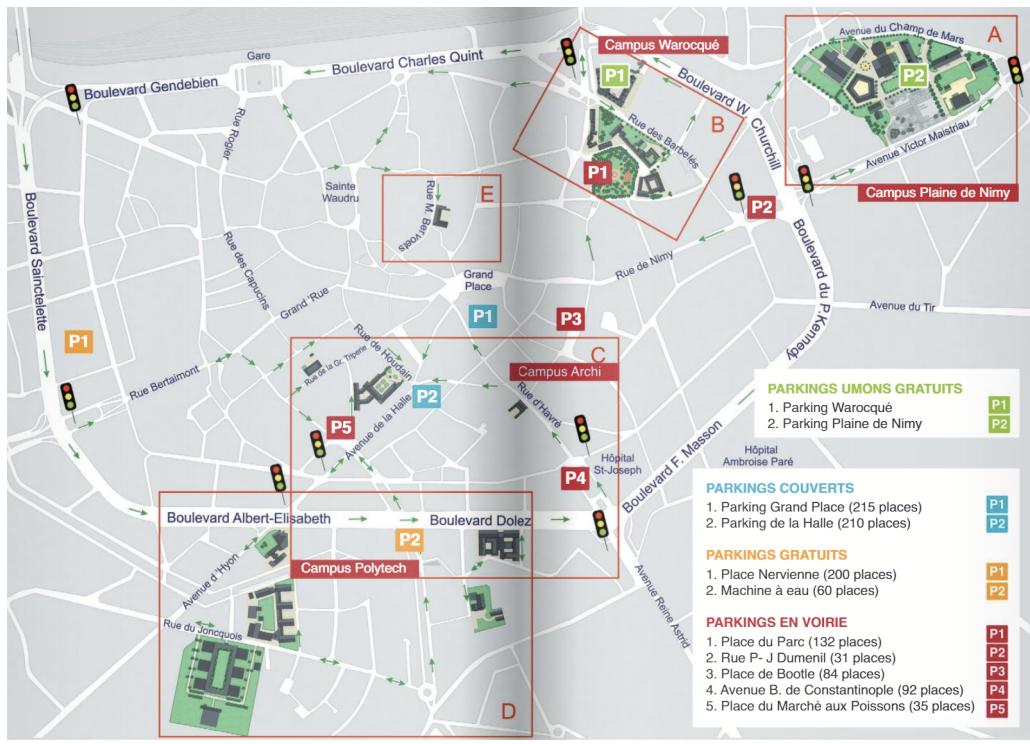


FIGURE 1.1 – Carte de la ville de Mons



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 logiciel pré-configuré 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 s'occupe principalement de gérer 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 salins (hyper et hyposalin) sur *Seriatopora hystrix* (Dana, 1815).

Chapitre 2

Introduction

Les coraux sont des animaux de l'embranchement des cnidaires. Les individus sont nommés « polypes ». Au sein des cnidaires, 1609 espèces de coraux durs (scléactiniaire hermatypique) forment les récifs coralliens. Les coraux durs vivent en symbiose avec une microalgue unicellulaire les zooxanthelles qui fournit l'énergie nécessaire à la formation de leur squelette carbonate de calcium.

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, peu expulser ses zooxanthelles, ce qui ne laisse paraître seulement la coloration blanche de son squelette. Ce blanchissement affaiblit considérablement le corail. Divers facteurs peuvent stresser le corail : l'acidité, la salinité, la température, la pollution, etc..

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 but du stage est de mettre à disposition des outils à partir du langage R permettant le monitoring des coraux tout en relevant régulièrement la croissance de ceux-ci.

Chapitre 3

But

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*.

Chapitre 4

Analyse

4.1 Acquisition de données réelles

4.1.1 Multiplication par bouturage

Dans le but d'acquérir des nouvelles données de croissance, on a utiliser une technique de multiplication asexuée : le bouturage. Cela consiste à séparer à l'aide d'une pince des branches de coraux. Elles sont suspendues dans l'eau à l'aide de fil de pêche sur une règle qui porte un numéro d'identification propre à chaque bouture.

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. Cette méthodde 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) :

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

ρ_{eau} est déterminé via l'équation d'état de l'eau de mer grâce à la mesure de la salinité et de la température. Le $\rho_{squelettique}$ est la densité de l'aragonite(CaCO₃) du squelette du corail.

4.2.2 Tableur

Les mesures effectuées sur les coraux et les paramètres de l'eau des mésocosmes sont encodé dans un tableau de données.

Pour l'instant, j'utilise ma licence d'Excel d'office 365 fourni par la HEH. Par la suite, j'aimerai utiliser un tableur en ligne afin que n'importe qui, qui a besoin de remplir un tableau de donnée puisse le faire depuis n'importe quelle machine connectée à internet.

Afin d'éviter au maximum des erreurs d'encodages, j'ai utilisé des règles pour mettre en évidence les cases non remplies, formater le type des cellules et mettre un dégradé de couleur suivant l'avancement des données.

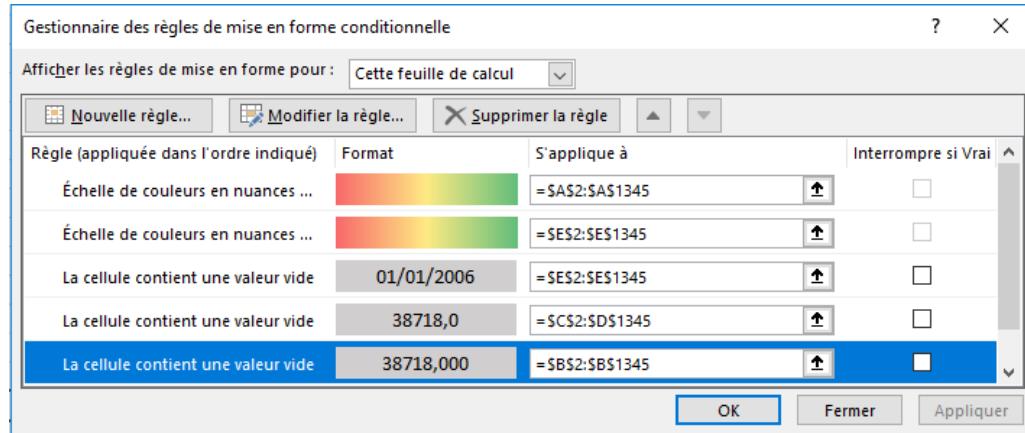


FIGURE 4.1 – Mise en forme conditionnelle d'Excel

Le tableau de donnée contient 7 colonnes :

- ID : corresponds à l'identifiant de la bouture.
- weight : corresponds à la masse immergée de la bouture.
- temp : corresponds à la température de l'eau de mer.
- salinity : corresponds à la salinité de l'eau de mer
- date : corresponds à la date et heure du relevé.
- commentaire : donne quelques annotations.

ID	weight	temp	salinity	date	Commentaire
1	0,415	25,1	35,1	11/2/19 14:20	
2	0,286	25,1	35,1	11/2/19 14:20	
3		25,1	35,1	11/2/19 14:20	
4	1,059	25,1	35,1	11/2/19 14:20	
5	0,677	25,1	35,1	11/2/19 14:20	
6	0,394	25,1	35,1	11/2/19 14:20	
7	0,795	25,1	35,1	11/2/19 14:20	
8	0,228	25,1	35,1	11/2/19 14:20	
9	0,508	25,1	35,1	11/2/19 14:20	
10	0,929	25,1	35,1	11/2/19 14:20	
11	0,519	25,1	35,1	11/2/19 14:20	
12	1,088	25,1	35,1	11/2/19 14:20	
13	0,603	25,1	35,1	11/2/19 14:20	
14	0,224	25,1	35,1	11/2/19 14:20	
15		25,1	35,1	11/2/19 14:20	
16		25,1	35,1	11/2/19 14:20	numéro 16 à rejeter, mal mesuré
17	0,465	25,1	35,1	11/2/19 14:20	

FIGURE 4.2 – Tableau de donnée

4.2.3 Application Shiny

L'application est divisée en deux éléments, 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, mais pour plus de clarté j'ai divisé mon script en deux fichiers ui.R et server.R (voir page annexe).

Mon application présente 2 onglets, le premier créer un graphique interactif.

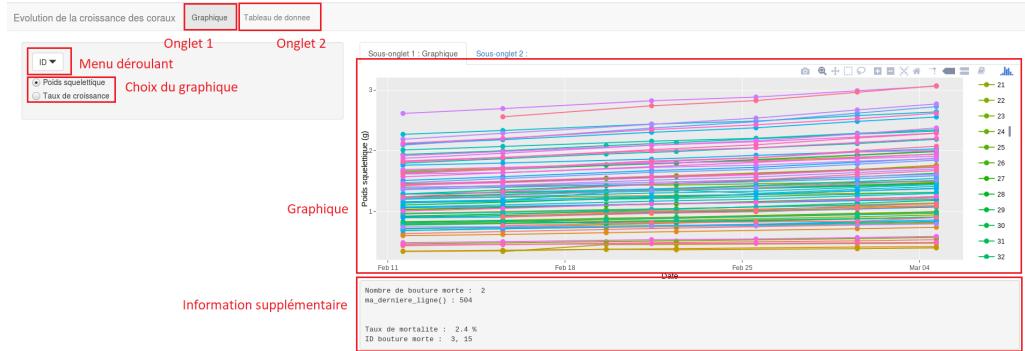


FIGURE 4.3 – Application Shiny : légende

Par défaut, le graphique montre l'évolution de la masse squelettique en fonction du temps.

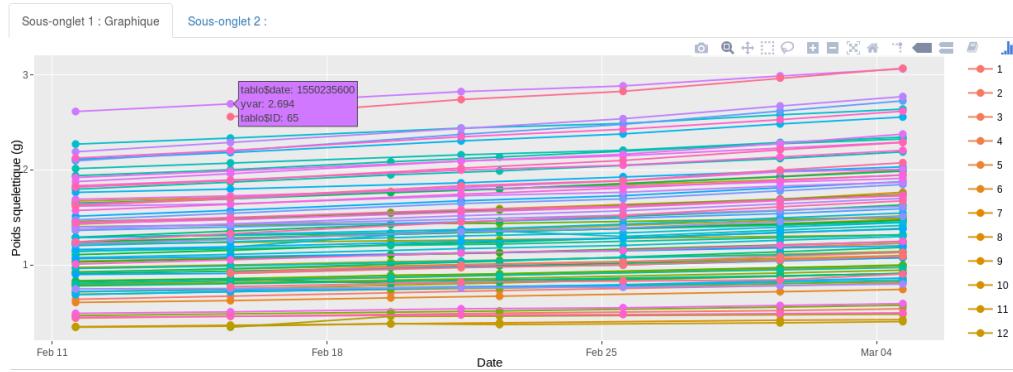


FIGURE 4.4 – Application Shiny : masse squelettique

On peut sélectionner le taux de croissance en fonction du temps.

Il est possible de sélectionner les ID dans un menu déroulant ou de directement cliquer à droite du graphique sur les ID triés par couleur.

Le menu déroulant permet de tout sélectionner ou de tout désélectionner.

En passant le curseur sur les points du graphique, on peut obtenir quelques informations.

Sous le graphique, des informations supplémentaires : le nombre de boutures mortes, leur ID et le taux de mortalité sont calculés.

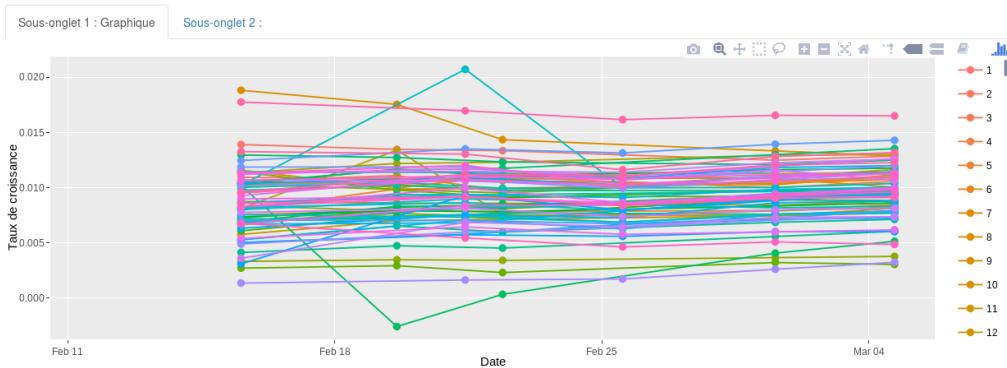


FIGURE 4.5 – Application Shiny : taux de croissance

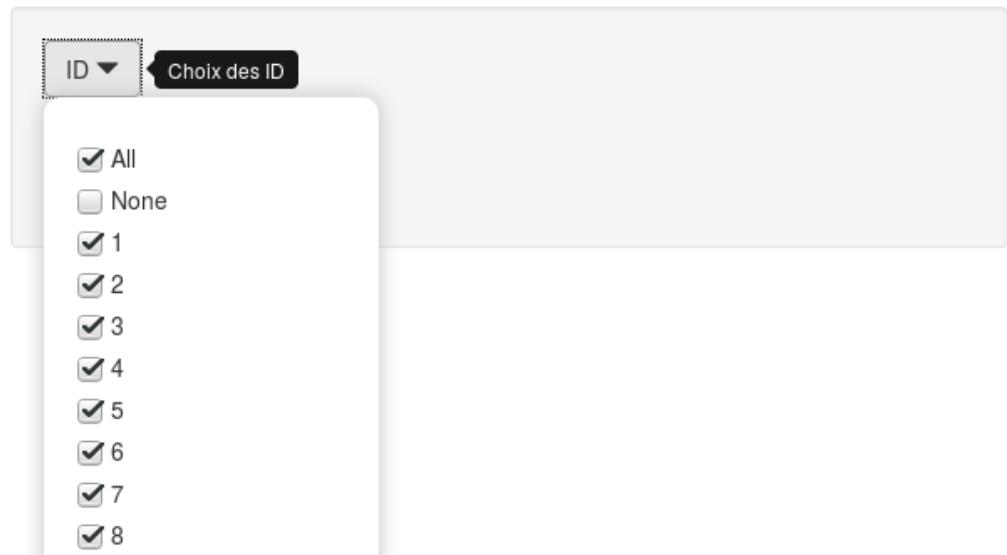


FIGURE 4.6 – Application Shiny : menu déroulant

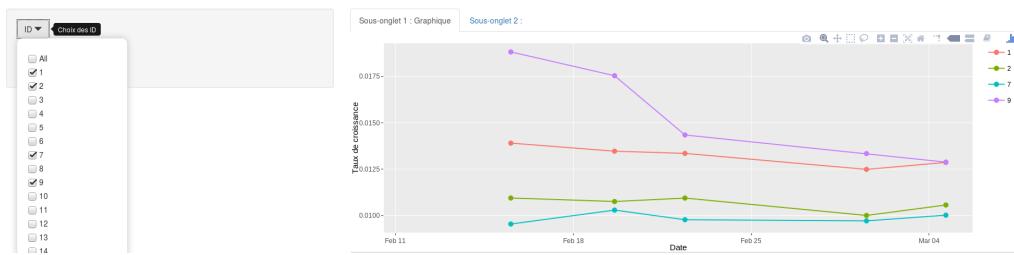


FIGURE 4.7 – Application Shiny : affichage interactif

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

```

Nombre de bouture morte : 2
ma_derniere_ligne() : 504

Taux de mortalite : 2.4 %
ID bouture morte : 3, 15

```

FIGURE 4.8 – Application Shiny : informations supplémentaires

croissance.

Show 10 entries							Search:
	ID	weight	temp	salinity	date	skw	ratio
1	1	0.415	25.1	35.1	2019-02-11T14:20:00Z	0.638	
2	2	0.286	25.1	35.1	2019-02-11T14:20:00Z	0.44	
3	3		25.1	35.1	2019-02-11T14:20:00Z		
4	4	1.059	25.1	35.1	2019-02-11T14:20:00Z	1.627	
5	5	0.677	25.1	35.1	2019-02-11T14:20:00Z	1.04	
6	6	0.394	25.1	35.1	2019-02-11T14:20:00Z	0.605	
7	7	0.795	25.1	35.1	2019-02-11T14:20:00Z	1.222	
8	8	0.228	25.1	35.1	2019-02-11T14:20:00Z	0.35	
9	9	0.508	25.1	35.1	2019-02-11T14:20:00Z	0.781	
10	10	0.929	25.1	35.1	2019-02-11T14:20:00Z	1.428	

Showing 1 to 10 of 504 entries

Previous 1 2 3 4 5 ... 51 Next

FIGURE 4.9 – Application Shiny : tableau de donnée

4.3 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.

4.4 Objectifs réalisés

Les objectifs réalisés sont :

- Bouturer les coraux et relever leurs masses immergées.
- Créer un tableau Excel contenant les données nécessaires.
- Créer un prototype d'application web à l'aide du paquet Shiny.

4.5 Planning de travail

Les horaires de stages sont flexibles, on peut arriver entre 7 et 9 heure et il faut prester au moins 8 heures par jour.

Chapitre 5

Communications interpersonnelles

5.1 Difficultés rencontrées

5.1.1 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. *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.

5.1.2 Shiny communication entre ui.R et server.R

Les application web gérée par shiny utilise 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.

5.1.3 Apport au sein de l'entreprise

Pour l'instant, la contribution à l'entreprise revient principalement aux relevés des bouteurs (monitoring) et l'encodage, ceci permet de libérer du temps au technicien. L'ap-

plication qui est en développement est utilisé dans le travail de mémoire de Madeleine Gilles.

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

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. Toutefois, dû à la récente prise en main de celui-ci, il y a un problème de positionnement des images qui n'a pas pu être résolu. LaTeX positionne automatiquement les images selon son bon vouloir.

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 à afficher
        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"))
      )
    )
  )
)
```

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) {

  # Madeleine :
  #coral_url <- "https://docs.google.com/spreadsheets/d/e/2PACX-1vTJLtfjjUM4VK6aM177ly9GCKyMHFr

  # Jordan :
  coral_url <- "https://docs.google.com/spreadsheets/d/e/2PACX-1vSoBfvhzFgALk1fc1jBbYP03D-fRIE

  #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 ) /"

}

# 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", "\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)
})
})

```