

Résultats et interprétations

Acquisition de données

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 suspendues 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).

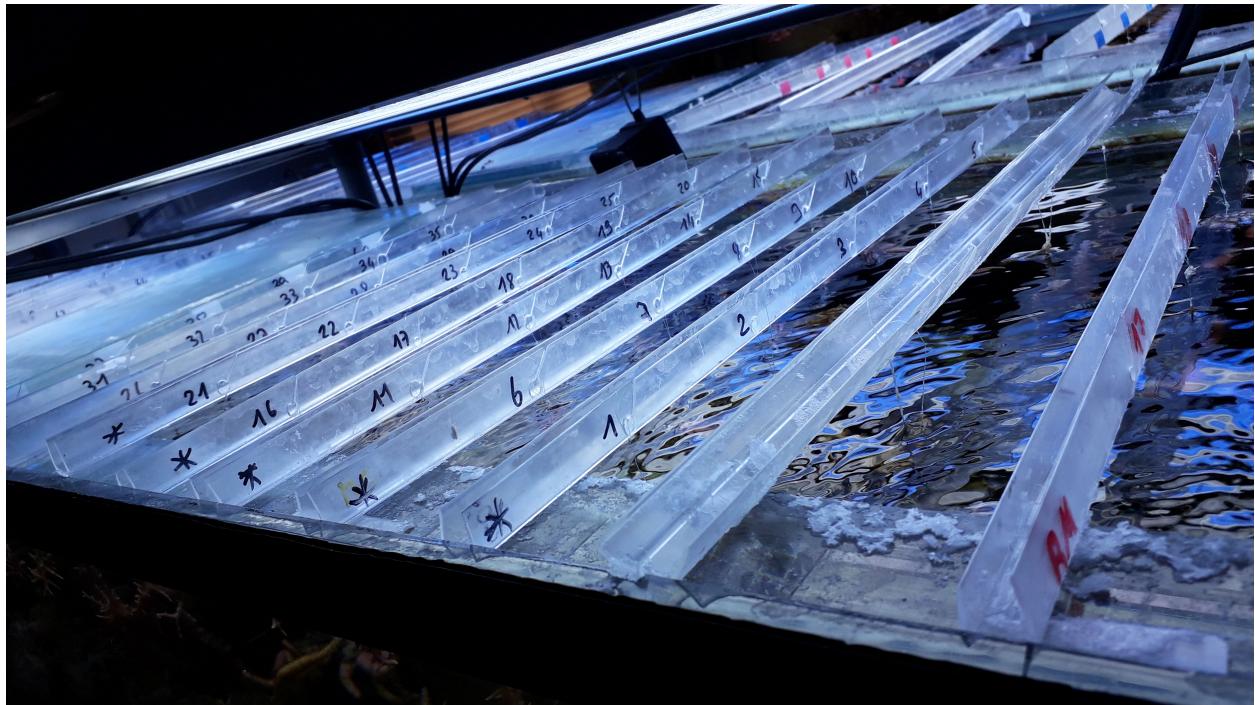


Figure 1: Règle numérotée

Outils monitorings

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.##). 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) :

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

ρ_{eau} est déterminé par l'équation d'état de l'eau de mer grâce à la mesure de la salinité et de la température. Le $\rho_{\text{squelettique}}$ est la densité de l'aragonite (CaCO_3) du squelette du corail.



Figure 2: Boutures suspendues de *Seriatopora hystrix*

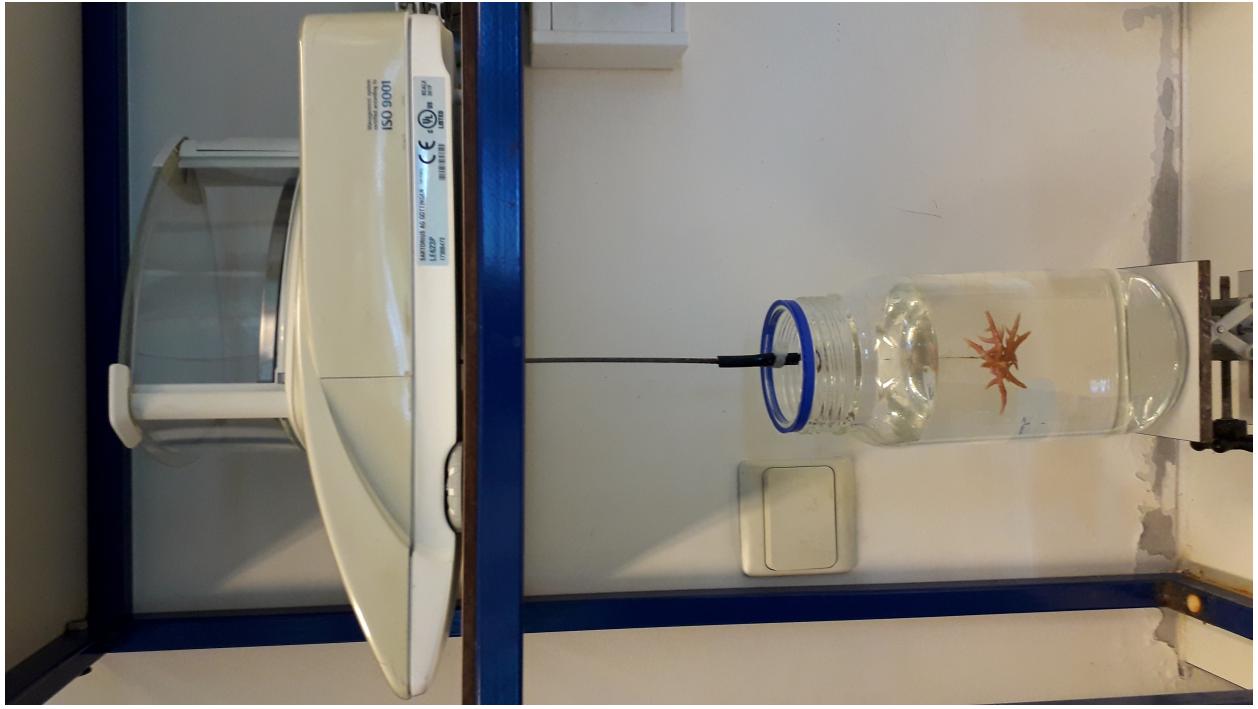


Figure 3: Tableur en ligne Google Sheets

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 choisit 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 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. 4.3).

	A	B	C	D	E	F	G	H	I	J	K	L
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	
17	mesocosm_monitoring	2019-02-11 14:20:00	jordan	B0	normal	s. hystrix	16		25.1	35.1	good	Numéro 16 à rejeter, mesure erronée

Figure 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 prises
- 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

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. 4.4).

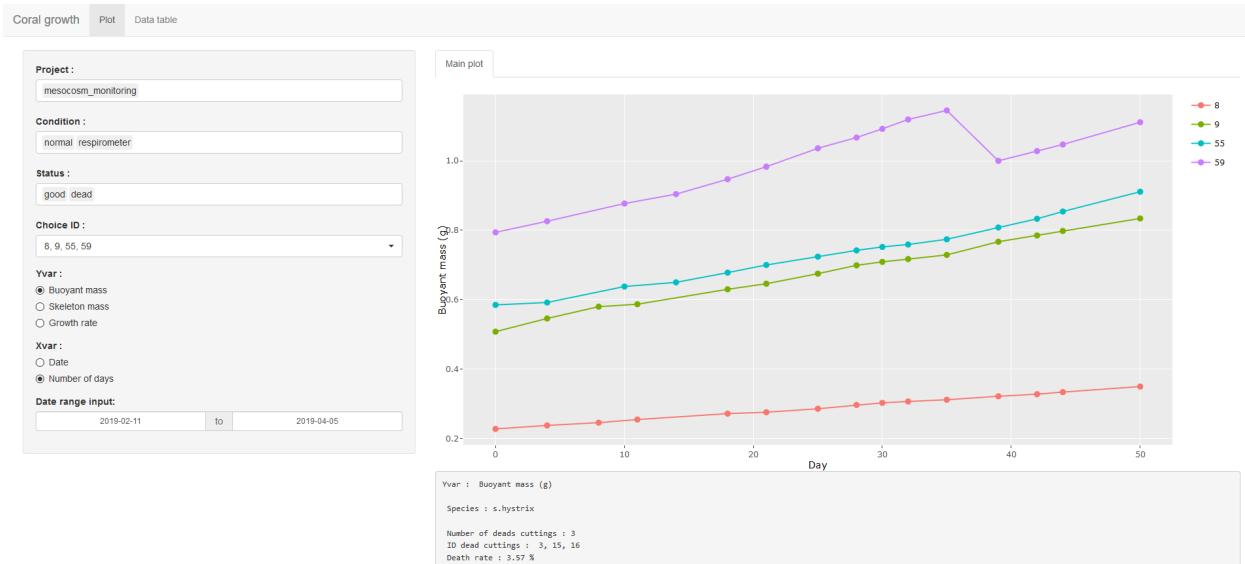


Figure 5: Application Shiny : onglet "Plot"

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. 4.5).

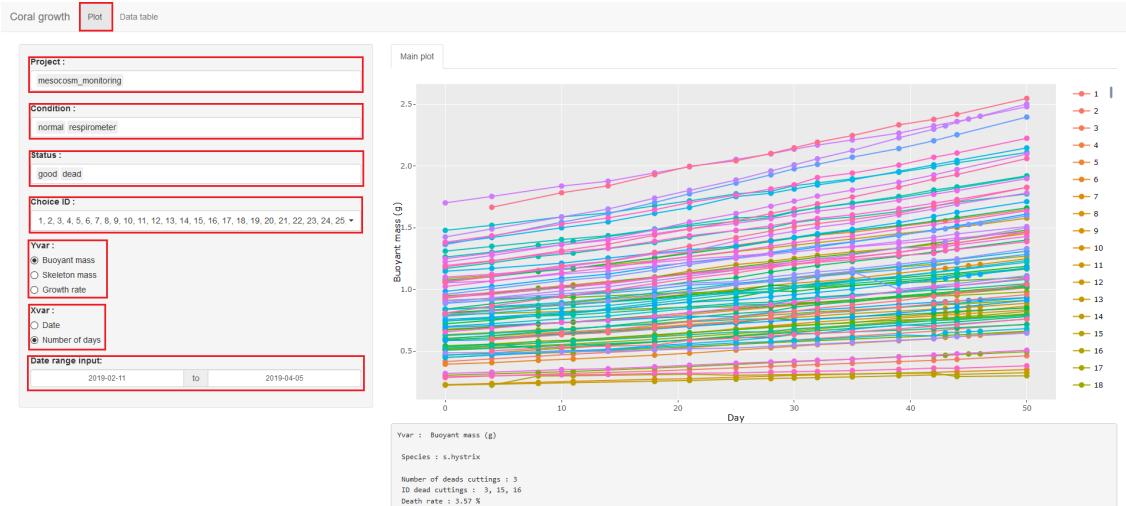


Figure 6: Application Shiny : paramètres

En ordonné, on peut choisir :

- la masse immergée
- la masse squelettique
- le taux de croissance (Fig. 4.6)

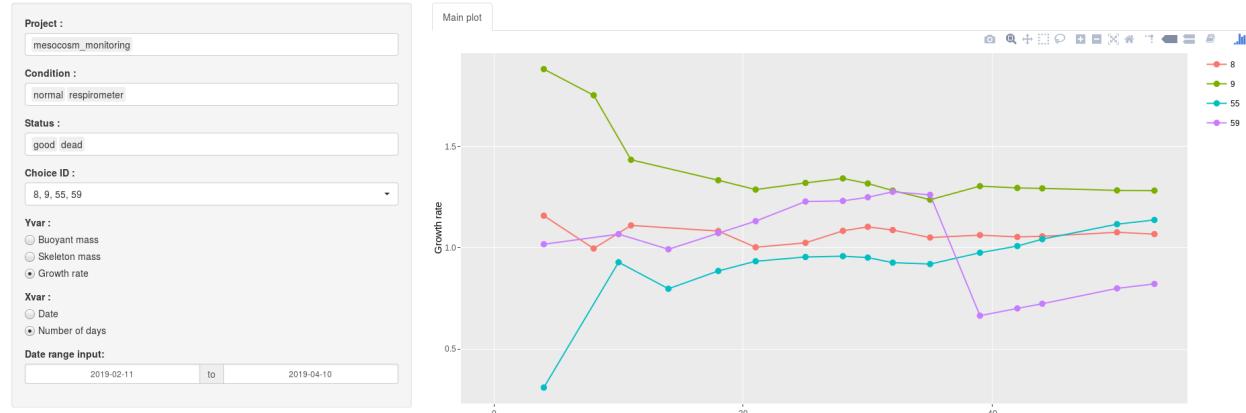


Figure 7: Application Shiny : taux de croissance

En abscisse, on peut choisir :

- la date de la mesure
- le nombre de jour écoulé 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 ou de directement cliquer à droite du graphique sur les ID triés par couleur (Fig. 4.7, Fig. 4.8).

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

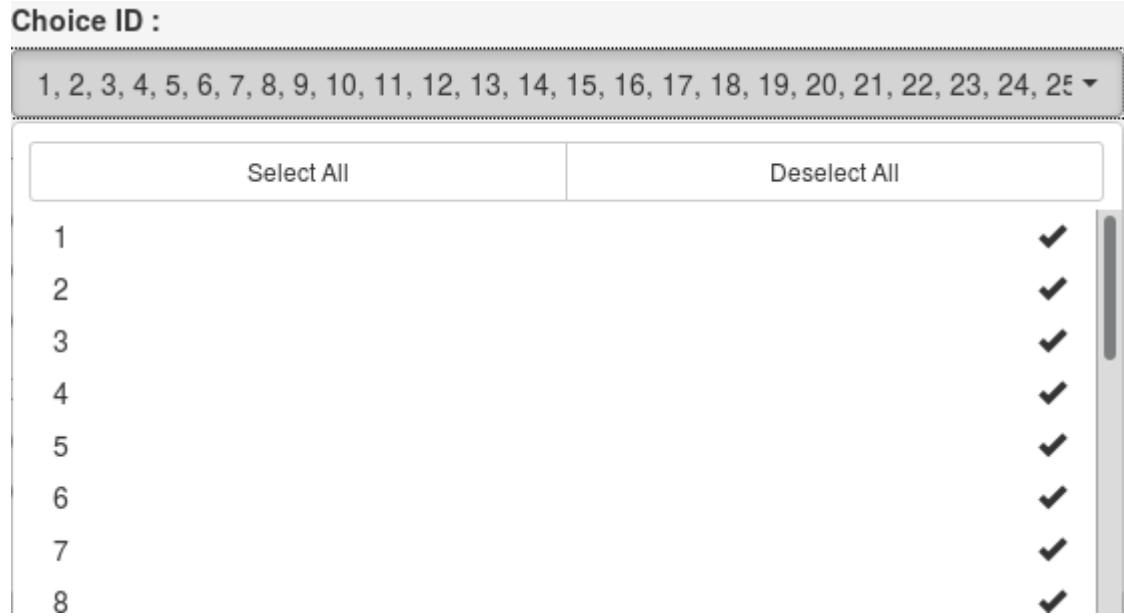


Figure 8: Application Shiny : menu déroulant

En passant le curseur sur les points du graphique, on peut obtenir quelques informations supplémentaires (Fig. 4.8). On peut également désélectionner les lignes en cliquant sur le numéro associé à la couleur de l'ID à droite de l'écran (Fig. 4.9).



Figure 9: Application Shiny : information via le curseur

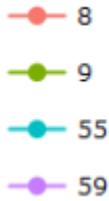


Figure 10: Application Shiny : affichage interactif

En bas du graphique, des informations supplémentaires sont données (Fig. 4.10) :

- Yvar : l'ordonnée du graphique
- 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

```

Yvar : Skeleton mass (g)

Species : s.hystrix

Number of deads cuttings : 3
ID dead cuttings : 3, 15, 16
Death rate : 3.57 %

```

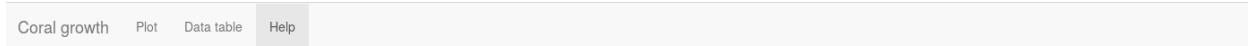
Figure 11: Application Shiny : informations supplémentaires

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.

project	date	author	aqua	condition	species	id	weight	temperature	salinity	status	comment	skw	delta_date	ratio
1	mesocosm_monitoring	2019-02-11T14:20:00Z	jordan	B0	normal	s.hystrix	1	0.224	2.580	35.1	good	0.638	0	
2	mesocosm_monitoring	2019-02-11T14:20:00Z	jordan	B0	normal	s.hystrix	2			35.1	good	0.44	0	
3	mesocosm_monitoring	2019-02-11T14:20:00Z	jordan	B0	normal	s.hystrix	3			25.1	dead		0	
4	mesocosm_monitoring	2019-02-11T14:20:00Z	jordan	B0	normal	s.hystrix	4	1.059	25.1	35.1	good	1.627	0	
5	mesocosm_monitoring	2019-02-11T14:20:00Z	jordan	B0	normal	s.hystrix	5	0.677	25.1	35.1	good	1.04	0	
6	mesocosm_monitoring	2019-02-11T14:20:00Z	jordan	B0	normal	s.hystrix	6	0.394	25.1	35.1	good	0.605	0	
7	mesocosm_monitoring	2019-02-11T14:20:00Z	jordan	B0	normal	s.hystrix	7	0.795	25.1	35.1	good	1.222	0	
8	mesocosm_monitoring	2019-02-11T14:20:00Z	jordan	B0	normal	s.hystrix	8	0.228	25.1	35.1	good	0.35	0	
9	mesocosm_monitoring	2019-02-11T14:20:00Z	jordan	B0	normal	s.hystrix	9	0.508	25.1	35.1	good	0.781	0	
10	mesocosm_monitoring	2019-02-11T14:20:00Z	jordan	B0	normal	s.hystrix	10	0.929	25.1	35.1	good	1.428	0	

Figure 12: Application Shiny : tableau de donnée

Le dernier onglet contient une page d'aide (Fig. 4.11). 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.



Utilisateur

L'application *Coral Growth* permet de suivre l'évolution de la croissance de boutures de corail à partir de l'URL d'un Google Sheets.

Tableur Google Sheets

Afin de mieux comprendre comment fonctionne l'application, il est important de connaître le jeu de données (dataframe).

Le tableau est divisé en 12 colonnes :

- project : différence chaque expérience réalisée, généralement on créera un nouveau tableau 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 tableau
- 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

	A	B	C	D	E	F	G	H	I	J	K
1	project	date	author	aqua	condition	species	id	weight	temperature	salinity	status
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

Figure 13: Application Shiny : onglet aide

Difficultés rencontrées

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 données 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. À 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 employés.

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 délimité 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.

Objectifs réalisés

Les objectifs réalisés sont :

- Bouturer les coraux et relever leurs masses immergées.
- Créer un tableau de donnée en ligne contenant les données nécessaires.
- Créer une application web répondant aux besoins du service à l'aide du paquet Shiny.

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

Conclusion

L'application web répond aux attentes.

Une documentation (bookdown) est intégrée à l'application.

Elle est disponible en ligne à l'adresse : <https://jack177.shinyapps.io/coralgrowth/>

Il est également possible de scanner le QR code.

Nov 18, 2018 – May 8, 2019

Contributions: Commits ▾

Contributions to master, excluding merge commits

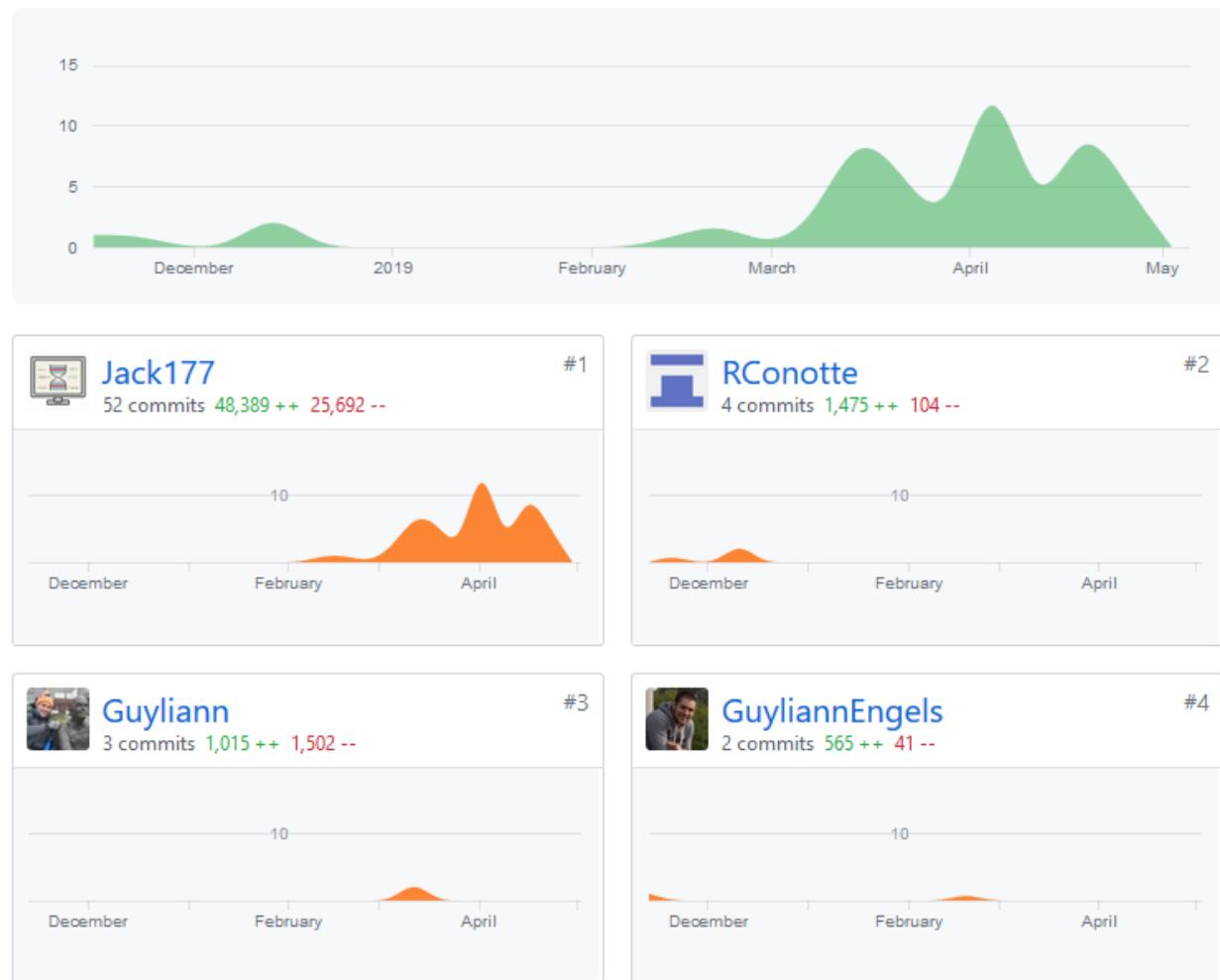


Figure 14: Contributeur de l'application



Figure 15: QR code