

# BioDataScience Administration

Philippe Grosjean & Guyliann Engels

2020-05-23



# Contents

<b>Préambule</b>	<b>5</b>
<b>1 Introduction</b>	<b>7</b>
<b>2 Organisation générale</b>	<b>9</b>
2.1 Infos utiles concernant le DSI UMONS . . . . .	11
<b>3 Git &amp; Github</b>	<b>13</b>
3.1 Installation Git et cie . . . . .	13
3.2 BioDataScience-Course . . . . .	13
3.3 Github Classroom . . . . .	14
3.4 Astuces Git et Github . . . . .	14
<b>4 Bookdown</b>	<b>15</b>
<b>5 Wordpress</b>	<b>17</b>
5.1 Installation Wordpress . . . . .	20
<b>6 Shiny</b>	<b>25</b>
<b>7 Moodle</b>	<b>27</b>
7.1 Badges . . . . .	27
7.2 Notes . . . . .	27
<b>8 H5p</b>	<b>29</b>
<b>9 Vidéos</b>	<b>31</b>
<b>A Installation</b>	<b>33</b>
<b>Références</b>	<b>35</b>



# Préambule

Notre organisation **BioDataScience-Course** sur Github devient de plus en plus complexe avec le temps car nous combinons plusieurs outils différents pour fournir à nos étudiants un environnement de travail et du matériel pédagogique riches et variés. Ce bookdown centralise l'information nécessaire pour les enseignants et les administrateurs de ces outils.



---

*Le matériel dans cet ouvrage est distribué sous licence CC BY-NC-SA 4.0.*

---

Détails concernant le système utilisé pour compiler ce bookdown...

## 0.0.0.0.1 Information système

```
sessioninfo::session_info()
```

```
# Session info
```

```

# setting value
# version R version 3.6.3 (2020-02-29)
# os      macOS Catalina 10.15.4
# system  x86_64, darwin15.6.0
# ui      X11
# language (EN)
# collate en_US.UTF-8
# ctype   en_US.UTF-8
# tz      Europe/Brussels
# date    2020-05-23
#
# Packages
# package * version      date      lib source
# assertthat 0.2.1      2019-03-21 [1] CRAN (R 3.6.0)
# bookdown   0.18      2020-03-05 [1] CRAN (R 3.6.0)
# cli        2.0.2      2020-02-28 [1] CRAN (R 3.6.0)
# crayon     1.3.4.9000 2020-04-29 [1] Github (r-lib/crayon@dcf6d44)
# digest     0.6.25     2020-02-23 [1] CRAN (R 3.6.0)
# evaluate   0.14      2019-05-28 [1] CRAN (R 3.6.0)
# fansi      0.4.1      2020-01-08 [1] CRAN (R 3.6.0)
# glue       1.4.0      2020-04-03 [1] CRAN (R 3.6.2)
# htmltools  0.4.0.9003 2020-04-29 [1] Github (rstudio/htmltools@984b39c)
# knitr      1.28      2020-02-06 [1] CRAN (R 3.6.0)
# magrittr   1.5       2014-11-22 [1] CRAN (R 3.6.0)
# rlang      0.4.5      2020-03-01 [1] CRAN (R 3.6.0)
# rmarkdown  2.1       2020-01-20 [1] CRAN (R 3.6.0)
# sessioninfo 1.1.1     2018-11-05 [1] CRAN (R 3.6.0)
# stringi    1.4.6      2020-02-17 [1] CRAN (R 3.6.0)
# stringr    1.4.0      2019-02-10 [1] CRAN (R 3.6.0)
# withr      2.2.0      2020-04-20 [1] CRAN (R 3.6.2)
# xfun       0.13      2020-04-13 [1] CRAN (R 3.6.2)
# yaml       2.2.1      2020-02-01 [1] CRAN (R 3.6.0)
#
# [1] /Library/Frameworks/R.framework/Versions/3.6/Resources/library

```

# Chapter 1

## Introduction

Chaque section couvre un outil en particuliers, avec la section 2 qui présente la façon dont les différents outils sont assemblés pour réaliser notre environnement de travail pour les cours de **Science des Données Biologiques** à l'UMONS.





## Chapter 2

# Organisation générale

Les divers outils logiciels que nous utilisons pour créer un environnement d'apprentissage riche et varié pour nos étudiants se basent pour la plupart sur des solutions Open Source et sur différents serveurs.

- R est le logiciel principal de statistique, visualisation et manipulation des données que nous enseignons à nos étudiants en même temps que les concepts de statistique et science des données eux-mêmes. C'est un logiciel puissant et gratuit que l'étudiant pourra s'approprier plus tard pour faire ses analyses tout au long de sa carrière s'il le souhaite. R est préféré à Python ici car il est plus orienté statistiques et moins programmation, ce qui convient mieux à nos étudiants biologistes.
- Rstudio en version serveur pour nos étudiants, ou en version serveur ou desktop pour les enseignants, couplé à R Markdown permet d'éditer du code R et des rapports ou autres documents au format R Markdown.
- Git est un système de gestion de version largement utilisé en programmation et de plus en plus utilisé en science des données. Combiné à R, RStudio et R Markdown, il offre la possibilité de travailler de manière *reproductible*. Nous enseignons git en même temps que les outils plus orientés science des données à proprement parler avec d'amener nos étudiants à prendre des bonnes habitudes en matière de traçabilité et reproductibilité de leurs analyses. Nous avons choisi Github comme système d'hébergement des dépôts git sur le Net, parce qu'il est gratuit, puissant, universellement utilisé et qu'il est couplé à RStudio. L'interface git et Github de RStudio rend son utilisation simple, et Github Desktop complète l'ensemble pour effectuer des tâches non reprises dans l'interface git de RStudio. Enfin, Github Classroom est une extension permettant de gérer les nombreux dépôts créés par nos étudiants dans le cadre de leur formation.
- La SciViews Box est une machine virtuelle sous VirtualBox totalement préconfigurée. Nous créons une SciViews Box chaque année avec tous les logiciels nécessaires aux étudiants mais pas seulement: il s'agit aussi d'une machine virtuelle permettant de réaliser de nombreuses analyses de façon parfaitement reproductible. Faire installer tous les logiciels nécessaires par les étudiants n'est pas possible. Plusieurs solutions s'offrent à nous :
- Utiliser RStudio cloud serait la solution la plus simple au départ car l'étudiant n'a qu'à se logger et c'est parti. Même avec un chromebook, il peut y travailler. Cependant, la session n'est disponible *que* pour le cours et l'étudiant doit tout installer

sur son ordinateur ensuite s'il veut continuer à les utiliser. cela nous paraît être une barrière importante à l'adoption de ces outils, même si beaucoup de (grandes) universités ont choisi cette voie.

- Utiliser Docker et Rocker qui est une solution de containerisation pour R et RStudio. Si ces outils fonctionnent bien en production, nous les avons trouvés moins pratiques que la machine virtuelle pour nos étudiants, surtout dans un contexte où la quasi-totalité des ordinateurs utilisés sont sous Windows ou MacOS.
- Enfin, la solution de la machine virtuelle avec la SciViews Box nous permet d'offrir une série d'autres logiciels en version interface graphique (sous Xubuntu). A l'usage, cela fonctionne bien, mais l'installation était encore un peu laborieuse lorsqu'elle devait se faire à la main dans VirtualBox. Nous avons donc conçu des logiciels annexes d'installation et de lancement de la machine virtuelle de sorte que celle-ci ne soit qu'à un clic souris de nos étudiants débutants ! A l'usage, cela fonctionne très bien. Nous avons un lanceur rapide par édition de SciViews Box, et ce, à la fois sous Windows et sous MacOS.

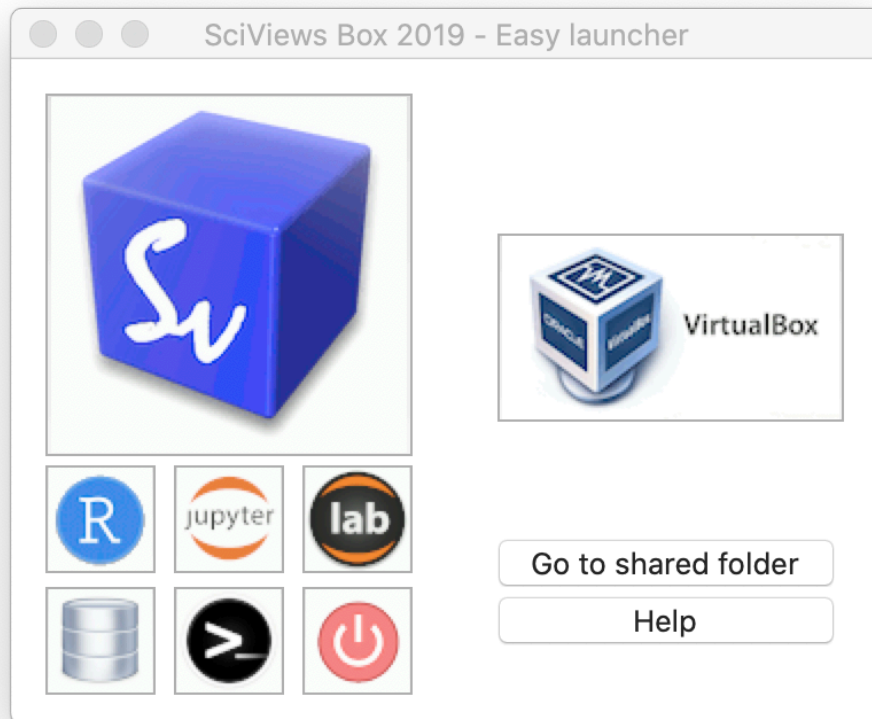


Figure 2.1: Lanceur rapide de la SciViews Box 2019

- Moodle est la plateforme d'apprentissage en ligne utilisée par notre université, l'UMONS. Si elle offre de nombreuses possibilités, il n'est pas possible d'y intégrer les outils d'apprentissage liés à R comme les learnrs (voir plus loin) et les ouvrages au format bookdown, voire encore des documents R Markdown. Nous essayons d'intégrer autant que possible nos outils dans cette plateforme, mais nous

sommes obligés de recourir à d'autres solutions pour pouvoir couvrir le panel des outils qui nous sont nécessaires (voir ci-après).

- Bookdown est un type de sortie R MArkdown multi-documents qui permet d'intégrer également des calculs et graphiques de R. Nous l'utilisons comme moyen de centraliser tout notre matériel pédagogique et l'organiser en modules qui sont autant de chapitres différents du bookdown. Nous avons également développé différentes extensions pour bookdown afin d'y intégrer tout ce que nous voulons (voir 4), et notamment des applications Shiny (voir 6) et des widgets h5p (voir 8). Un chapitre entier y est donc consacré.
- Shiny permet de créer des applications sous R avec interface graphique sur un browser web. Les possibilités sont pratiquement sans limites pour créer des applications interactives et démontrer les concepts "en live" à nos étudiants. Les applications Shiny sont intégrables dans bookdown, mais elles doivent être exécutées sur un serveur dédié. TODO: une phrase sur ce serveur. Un type particulier d'application Shiny est constitué par les learnrs. Ces documents interactifs présentent des exercices dans R et des choix multiples aux étudiants. Ces learnrs forment la base des outils d'apprentissage à domicile et aussi des examens théoriques de nos cours.
- H5p propose une large variété de widgets javascript intégrables dans une page web sous forme d'iframes pour varier et rendre plus efficace l'apprentissage. Il s'agit d'un logiciel gratuit open source, mais il faut installer son propre serveur sous Moodle, Wordpress ou Drupal. Des essais ont été fait sous Moodle de l'UMONS. Cela fonctionne, mais il apparait que les widgets h5p de Moodle ne sont *pas* intégrables dans un bookdown, en tous cas pas la version de test telle qu'elle était configurée. Nous avons donc décidé d'installer notre propre serveur Wordpress, essentiellement pour pouvoir faire tourner nos widgets h5p. Ce serveur est hébergé chez Gandi.net qui gère aussi le nom de domaine **sciviews.org** sous lequel nos créations sont servies sur le net, ainsi que les boites mails correspondantes, comme **sdd@sciviews.org**.
- Les vidéos relatives au cours sont hébergées sur Youtube dans un canal dédié **BioDataScience-Course**. La section relative aux vidéos (voir 9) explique aussi en détails comment nous créons ces vidéos et avec quels outils.
- Enfin, les données relatives à la progression des étudiants dans les learnrs sont collectées dans une base de données MongoDB hébergée sur le cloud MongoDB Atlas.



Ces nombreux outils et services devant collaborer entre eux pour fournir un environnement d'apprentissage complet et cohérent, il est important de les configurer et de les faire collaborer correctement. C'est le but principal du présent document que d'expliquer comment nous y arrivons.

Les sections suivantes détaillent l'installation et la configuration de différents outils logiciels tandis que les paragraphes suivants dans la présente section reprennent quelques informations générales utiles.

## 2.1 Infos utiles concernant le DSI UMONS

- Les documents **Comment ?** expliquent comment réaliser pas mal de choses (connexion, gestion des mails, partage de fichiers, etc.).



## Chapter 3

# Git & Github

Nous utilisons abondamment et enseignons l’usage de git et Github dans le cadre de nos cours de **Science des Données Biologiques** à l’UMONS. De plus, nous utilisons Github Classroom pour gérer la distribution des travaux aux étudiants sous forme de dépôts Github. Cette section couvre ces logiciels.

### 3.1 Installation Git et cie

Dans la SciViews Box, git, Github Desktop et RStudio server (incluant une interface vers git) sont préinstallés. Hors machine virtuelle, il faut installer git, RStudio et Github Desktop. Le reste se fait en interagissant avec Github via le navigateur web.

### 3.2 BioDataScience-Course

L’organisation BioDataScience-Course regroupe tous les dépôts Github relatifs à nos cours, y compris les travaux des étudiants.

#### 3.2.1 Convention de noms

Comme l’organisation contient plusieurs centaines de dépôts, il est utile de les nommer correctement pour s’y retrouver. Les dépôts créés via Github Classroom seront traités plus loin. Voici quelques règles :

- Les dépôts important (matériel pédagogique, gestion des étudiants et de leur progression, ...) commencent tous par **sdd**. Pour les dépôts moins importants ou de test, utiliser un autre préfixe.
- Lorsqu’un dépôt est relatif à une cohorte d’étudiants en particulier, nous l’indiquons par un suffixe reprenant les deux derniers chiffres de l’année initiale, suivi de **m** pour les étudiants montois et **c** pour les étudiants de Charleroi. Par exemple, **sdd\_students19m** concerne la cohorte d’étudiants montois inscrits en premier cours (Bab2 de biologie) en 2019-2020. De même **sdd\_students20c** concerne la cohorte d’étudiants de Charleroi inscrits en Bab2 en 2020-2021.

### 3.3 Github Classroom

Nous utilisons Github Classroom pour générer la distribution des travaux aux étudiants sous forme de dépôts Github.

TODO: Guyliann: les éléments importants de la gestion de Github Classroom sont à présenter ici...

TODO: intégration de Github Classroom avec Moodle, voir <https://classroom.github.com/help/setup-moodle> & <https://classroom.github.com/help/import-roster-from-lms>.



Une autre ressource intéressante dans le contexte de l'utilisation de Github/Github Classroom pour l'enseignement est Github Education.

### 3.4 Astuces Git et Github

Quelques astuces qui nous sont utiles...

- Comparaison de branches sur Github : ajouter `/compare` à la fin de l'URL du dépôt et une page de comparaison de branche apparaît. sinon, cette comparaison est aussi possible dans Github Desktop (menu **Branch** -> **Compare to Branch**).
- Lorsqu'une branche n'est plus utile, elle peut être facilement éliminée depuis Github Desktop (entrée de menu **Branch** -> **Delete...**). Cette branche doit être activée d'abord, et tous les commits doivent avoir été pushés. Il est conseillé d'éliminer la branche sur le serveur en même temps que la branche locale.

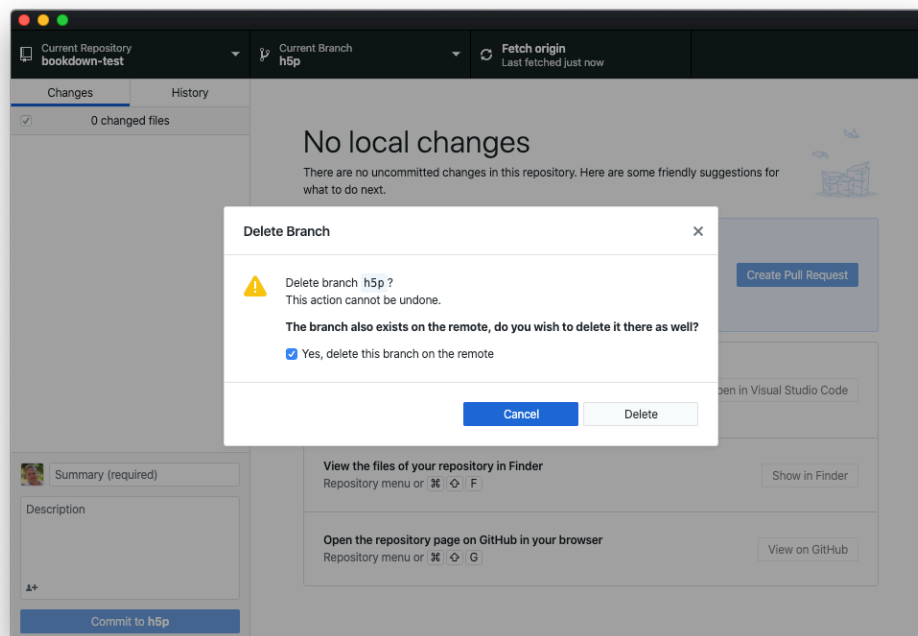


Figure 3.1: La boîte de dialogue d'élimination d'une branche dans Github Desktop.

## Chapter 4

# Bookdown

...





## Chapter 5

# Wordpress

Nous utilisons Wordpress principalement pour gérer nos widgets **h5p**. Cependant, étant basé sur un serveur LAMP (Linux-Apache-MySQL-PHP), il permet également de réaliser des fonctions basées sur des scripts côté serveur qui seraient difficiles, voire impossibles avec un site statique tel que l'offre bookdown, par exemple. Ainsi, nous pouvons aussi y intégrer un enregistrement de l'utilisateur via Github (protocole OAuth2). Ceci est bien pratique pour faire le lien entre le compte UMONS/Moodle et le compte Github de l'étudiant.

En effet, en 2018-2019 et 2019-2020, nous avons travaillé uniquement avec un serveur statique bookdown hébergé sur Github. Si le résultat est correct, nous avons noté plusieurs problèmes :

- Pas d'enregistrement des étudiants,
- Pas de h5p. Nous pourrions utiliser <https://h5p.org>, mais cette plateforme n'est prévue que pour des tests, et ne permet pas de récupérer les activités des étudiants,
- Pas de service d'applis Shiny et de learnrs. Nous exécutons ces applis en local dans la SciViews Box via les packages R BioDataScience et BioDataScience2. Cela permet d'utiliser les learnrs, mais sans être vraiment très pratique pour les étudiants. Les learnrs qui y sont intégrés se connectent à la base de données MongoDB via une petite appli Shiny intégrée qui demande le login Github de l'étudiant et son email pour l'identifier. Toutefois, comme aucun contrôle n'est fait sur ce qu'il entre dans ces champs, nous avons eu beaucoup d'enregistrements fantaisistes qu'il a fallu régler grâce à une table de corrections appliquée aux données de la base.

Au final, même si l'approche bookdown statique sur Github + learnr/Shiny en local dans la SciViews Box permet d'obtenir le minimum de fonctions requises pour le suivi de la progression d'apprentissage des étudiants, ce n'est pas pratique à l'usage et nous avons dû dépanner souvent des étudiants... sans compter leur stress de savoir si leur activité est bien enregistrée dans la base de données ou non (deux points sur vingt dans la note totale sont réservés à l'exécution de ces learnrs, d'où l'importance de bien enregistrer qui fait quoi).

Donc, en 2020-2021, nous avons décidé de mettre en œuvre un serveur Wordpress pour fournir des widgets h5p qui nous manquent et dont l'utilisation est enregistrée (protocole xAPI), et aussi pour nous permettre de mettre en œuvre une authentification des étudiants plus fiable. Notre objectif est double au niveau de l'authentification de nos étudiants. D'une part, nous voulons obtenir de manière fiable à la fois leurs données UMONS (via

Moodle) et leurs données Github, ce qui implique que ces données doivent être soit fournies automatiquement, soit vérifiées durant l'authentification. D'autre part, nous voulons pouvoir exécuter des applications Shiny en mode sécurisé pour, par exemple, leur offrir un rapport de progression calculé en direct. La nouvelle authentification se fait comme suit :

1. Nous intégrons nos outils dans Moodle qui nous fournit alors l'authentification UMONS de l'étudiant. On obtient, entre autres, son numéro de matricule, son adresse email UMONS, son nom et son prénom, le cours qu'il suit, ... Les détails sont dans la section correspondante (voir 7) et ne seront donc pas discutés ici.
2. Par ailleurs, nous utilisons abondamment Github pour les exercices des étudiants. Leur login y est différent, et il est aussi possible qu'ils y aient indiqué une adresse email différente (même si nous leurs demandons de mettre la même). Dans Wordpress, nous pouvons intégrer une authentification sur base de leur compte Github via le protocole OAuth2. C'est doublement intéressant car nous avons leur réel login Github à tous les coups, et aussi, nous sommes certains que leur compte Github est créé et actif dès le moment où ils sont enregistrés dans Wordpress. Que des avantages, donc. De plus, dans l'implémentation choisie, la connexion via Github montre clairement à l'étudiant s'il est connecté ou non.

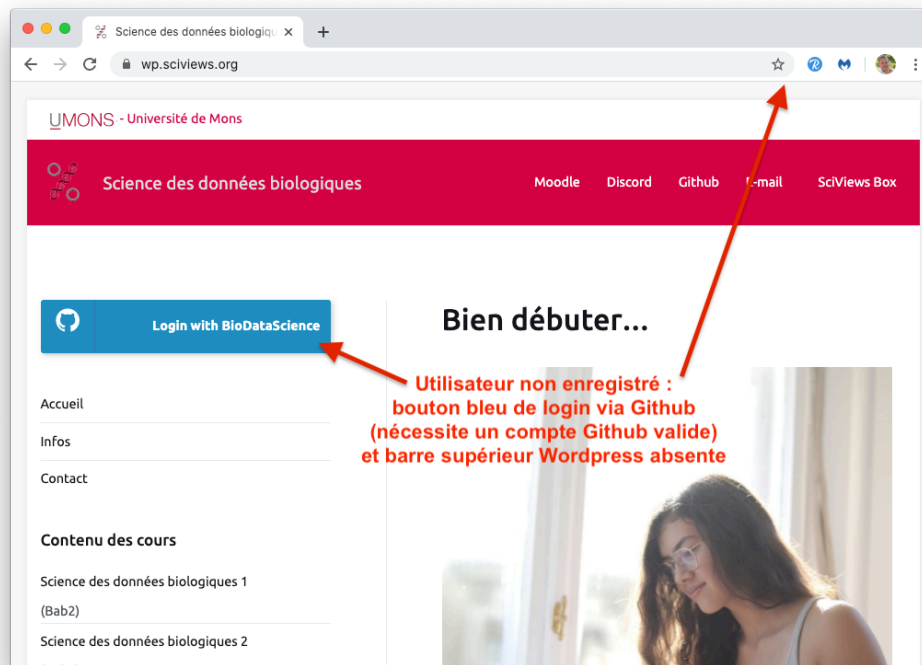


Figure 5.1: Page principale du site lorsque l'utilisateur n'est pas connecté (au travers d'un login BioDataScience dans Github).

Dès que l'étudiant s'est enregistré, les données utiles venant de Github sont rentrées dans la table des utilisateurs Wordpress. Nous pouvons donc combiner cette information avec les données provenant de Moodle pour établir un profil complet de l'étudiant dans notre site Wordpress automatiquement.

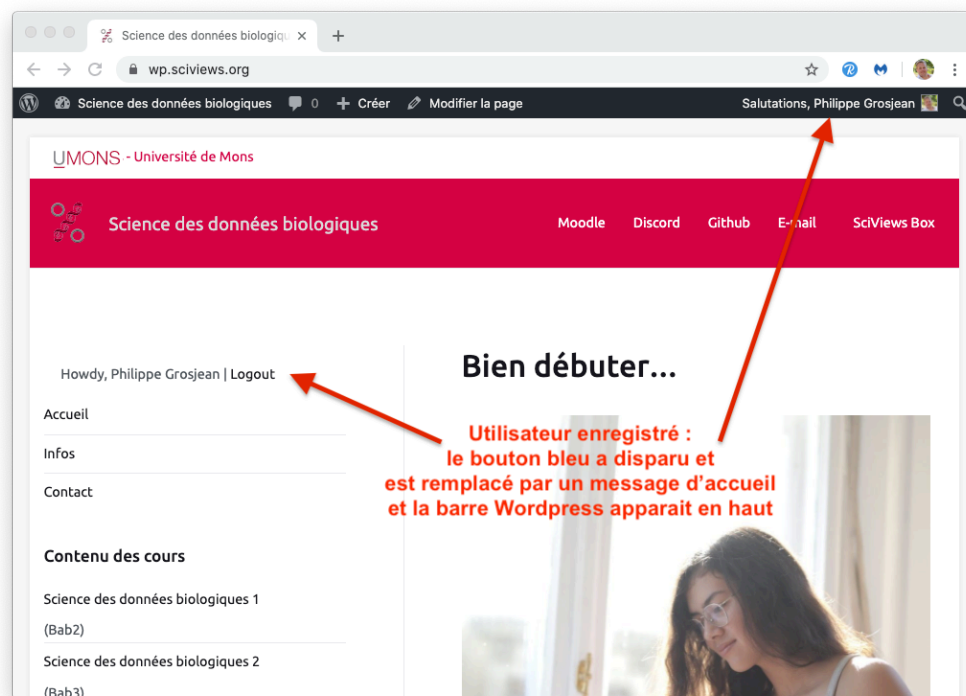


Figure 5.2: La même page principale du site lorsque l'utilisateur est enregistré et connecté via Github.

## 5.1 Installation Wordpress

Notre nom de domaine **sciviews.org** est géré par <https://gandi.net>. Ce provider propose aussi un hébergement web Wordpress dans un serveur LAMP préconfiguré. Nous utilisons cette option pour l'instant en test. Le serveur utilise 1 cpu et 256Mo de RAM avec 20Go d'espace de stockage, et un quota de 300.000 pages vues par mois et 60Go par mois. Il est upgradable à tous niveau. Nous verrons comment il se comporte en charge avec nos étudiants et pourrons toujours passer vers un plan supérieur si des lenteurs sont constatées. Le certificat SSL pour avoir une adresse sécurisée en **https://** est inclu dans le package.

Le serveur est installé initialement avec **PHP 7.3** et une base de données **MySQL 5.7** ne permettant que des requêtes locales. **Wordpress** est à la version **5.4.1**. Nous avons installé les extensions suivantes :

- **H5P 1.15.0**. Il faut installer les libraires séparément et les principales ont été rajoutées dès le départ. Les 5 ou 6 widgets H5P que nous avons déjà créés sont facilement importés en les exportant de leur site original au format **.h5p** en cliquant sur le bouton **reuse** en bas à gauche du widget. Ensuite dans Wordpress, on va dans le tableau de bord -> H5P Content, on clique **Add New** puis **Téléverser** et on sélectionne le fichier **.h5p**. Un backup de ces fichiers **.h5p** est également effectué sur p-cloud dans le dossier **sdd\_h5p\_backup**. Le plugin est ensuite configuré pour ne pas motrer la bannière h5p en bas des widgets et pour générer tous les événements xAPI possibles.
- **SNORDIAN's H5PxAPIkatchu 0.4.2** capture les événements xAPI générés par H5P et les stockent directement dans la base de données MySQL liée à Wordpress. Il y a une interface simpliste pour visualiser les tables, les filtrer et explorer au format CSV. **Dans un premier cela nous servira, mais nous devons développer une autre façon de récupérer les données, voire, dériver un autre plugin pour envoyer les événements directement dans la même base de données MongoDB que celle utilisée par nos applis Shiny (voir 6).**
- **OAuth Single Sign On - SSO (OAuth Client) 6.15.2** configuré pour enregistrer et connecter un utilisateur via son compte Github. Pour permettre ceci une application Github nommée **BioDataScience** a été créée. La configuration de ce plugin est disponible depuis le tableau de bord de Wordpress -> miniOrange -> OAuth.
- **Import Users from CSV 1.0.1** est un plugin permettant de créer des utilisateurs en batch depuis un fichier CSV. Nous pourrons ainsi exporter en bloc les données venant de Moodle et créer en avance les comptes Wordpress avec plein d'informations supplémentaires.
- **Advanced iFrame 2020.2.1** permet d'insérer un iframe avec plein d'options : sécurisation, adaptation de la taille, passage de paramètres, etc. Nous utilisons ce plugin pour intégrer les bookdown dans le site Wordpress, et ainsi conserver la bannière supérieure et le pied de page de notre site tout en affichant le bookdown.
- **UpdraftPlus 1.16.24** comme système de backup de l'ensemble (fichiers Wordpress + base MySQL). Le backup se fait en local pour l'instant dans `lamp0/web/vhosts/wp.sciviews.org/htdocs/wp-content/updraft`, mais il est recopié sur notre disque p-cloud dans le dossier **sdd\_site\_backup**.
- Quelques autres plugins aidant à gérer le site Wordpress : **Broken Link Checker 1.11.12**, **Health Check & Troubleshooting 1.4.4** & **WP-Optimize 3.0.19**.

Le site est servi à l'adresse <https://wp.sciviews.org>. Les enseignants y sont enregistrés comme éditeurs. En outre, un compte administrateur est aussi créé. Le serveur est accessible par SSH (après activation provisoire dans gandi) et sFTP. Toutes les informations nécessaires seront transmises aux intéressés par voie sécurisée.

### 5.1.1 Intégration des bookdowns dans le site

Pour l'instant, il y a deux versions :

- Science des données biologiques 1 renvoie vers le bookdown de test directement. A partir de ce moment, nous perdons le contrôle et c'est le bookdown qui doit tout gérer... or comme c'est une page statique, les possibilités sont limitées à du JavaScript qu'on aurait été y mettre,
- Science des données biologiques 3 revoie à la même page, **mais intégrées sous forme d'iframe** dans une page de Wordpress dédiée. Nous utilisons le plugin *Advanced iFrame* qui permet de contrôler la taille de l'iframe, de lui passer des paramètres et encore d'autres choses. Cela donne pas mal, et l'utilisateur conserve la barre d'outils supérieure (accès à Moodle, Discord, etc.) S'il veut lire le contenu du bookdown sans voir cette barre, il lui suffit de faire coulisser la page jusque tout en bas et la barre de menu du site sera cachée.

### 5.1.2 Intégration des h5p dans bookdown

H5p s'affiche dans un iframe dans le document principal. Il lui transmet les événements xAPI, et c'est donc ce document principal qui doit les traiter. Nous avons ici une impossibilité à le faire si nous sommes dans un cas où h5p est servi par un serveur différent que notre bookdown. En effet, les accès en "Cross-Origin Resource Sharing" ou CORS sont restreintes pour éviter des attaques de site évidentes. Ainsi un iframe issu d'un serveur différent de la page qui l'héberge ne peut communiquer librement avec la page qui le contient si elle provient d'un site différent. Or, nous avons besoin de cette communication pour transmettre les événements xAPI de h5p.

La solution que nous avons trouvée est d'ajouter le dossier qui contient nos pages statiques du bookdown à Wordpress directement. Ainsi les deux sont maintenant servis à partir de la même adresse <https://wp.sciviews.org>. Pour l'instant, nous copions simplement ces fichiers dans un sous dossier **sdd-umons** dans **lamp0/web/vhosts/wp.sciviews.org/htdocs**. Ainsi, le bookdown en question est servi par Wordpress à l'adresse <https://wp.sciviews.org/sdd-umons>. Et à présent, la communication des xAPI h5p est possible... à condition d'ajouter le code nécessaire dans bookdown.

Nous créons un fichier **header.html** dans bookdown qui contient ce code, extrait du serveur Wordpress, puisque nous sommes dedans maintenant. Il suffit alors de le renseigner dans l'entête YAML du fichier principal **index.Rmd** comme :

```
includes:
  in_header: header.html
```

et le tour est joué. Il nous faut maintenant inclure facilement les widgets h5p depuis Wordpress dans nos pages bookdown. Ces widgets sont identifiés par un numéro (ID) dans Wordpress.

Nous devons créer un iframe dans la page bookdown, mais nous pouvons déléguer ce travail à une fonction R :

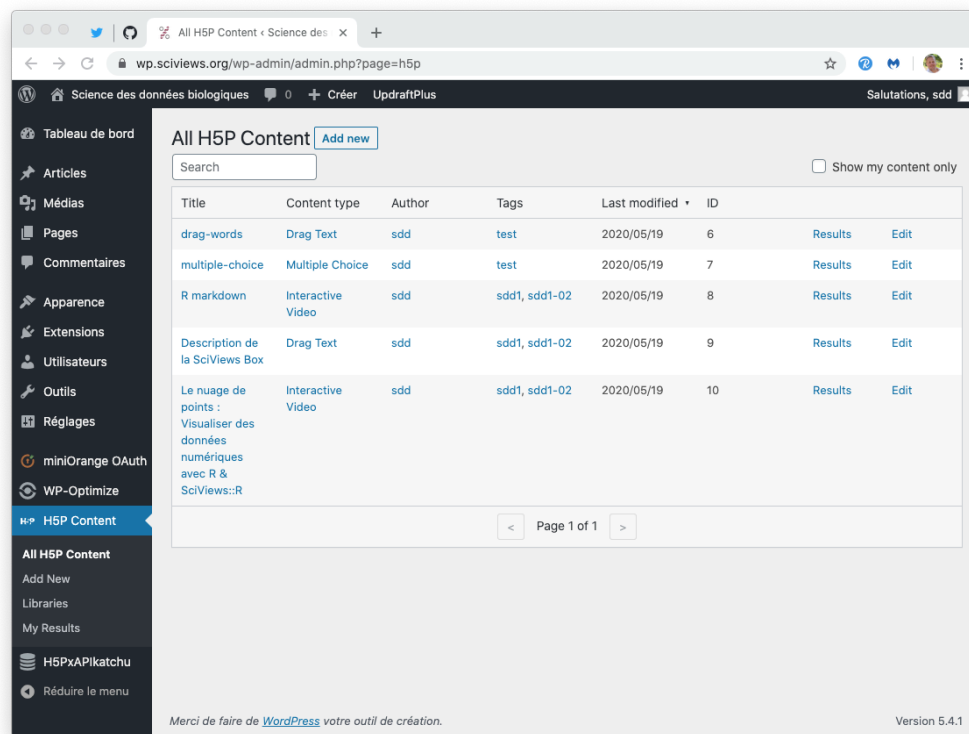


Figure 5.3: H5P content dans Wordpress. Les widgets sont identifiés par leur ID (ici, 6, 7, 8, 9 & 10).

```
h5p <- function(id, width = 780, height = 500) {
  glue::glue("\n<iframe src=\"https://wp.sciviews.org/wp-admin/admin-ajax.php?action=h5p_embed&")
}
```

Ensuite, ajouter un widget h5p est aussi simple que d'ajouter `h5p(id)` dans du code R en ligne où "id" est l'ID tel que renseigné dans le tableau sous Wordpress. On peut aussi jouer sur les arguments `width=` et `height=` pour ajuster l'emplacement du widget h5p dans la page.

Ainsi programmé notre widget h5p se comporte comme suit :

- pour un utilisateur enregistré dans Wordpress, les événements xAPI sont enregistrés dans notre base de données MySQL. Nous pouvons donc suivre la progression de nos étudiants via les widgets h5p également, en plus des applis widgets et des learnrs. De plus, l'état d'achèvement de l'exercice est aussi enregistré pour un bon nombre de widgets et rétabli lorsque la page est rechargée par le même utilisateur.
- pour un utilisateur non enregistré, les widgets h5p fonctionnent normalement, mais aucun événement xAPI n'est enregistré, et l'état de complétude de l'exercice n'est pas conservé non plus. Si la page est rechargée, les widgets h5p retournent à leur état initial à chaque fois.

Ceci est un inconfort mineur et permet l'utilisation des widgets h5p de manière optimale pour nos étudiants dûment enregistrés, tout en permettant une libre utilisation par d'autres visiteurs sans polluer notre base de données.

Nous avons également tenté de passer des arguments supplémentaires via l'URL qui lance la page bookdown, et qui les transfèrent vers l'iframe h5p, mais sans succès car apparemment Wordpress réécrit l'URL et laisse tomber ces arguments supplémentaires. C'est pas important si nous pouvons tracer nos utilisateurs via leur login (= login Github pour rappel).

### 5.1.3 Création de contenu h5p

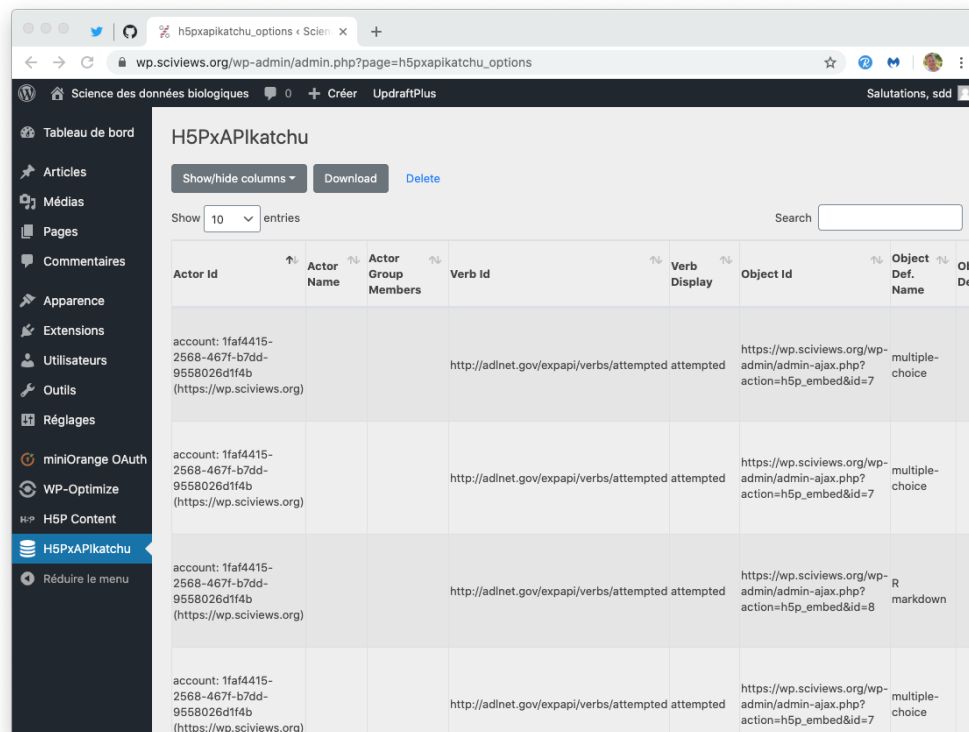
Tout éditeur Wordpress (les enseignants) peuvent créer du contenu nouveau h5p à partir du tableau de bord de Wordpress, et éditer les widgets existants. L'édition de manière quasi-wysiwyg avec les mêmes éditeurs que ceux disponibles sur <https://h5p.org>. C'est donc très facile. Il faudrait juste se discipliner à faire des backups via les fichiers .h5p sur p-Cloud dans `sdd_h5p_backup` par sécurité.

TODO: il serait utile de pouvoir sortir une table de correspondance entre l'ID du widget h5p et un nom plus explicite. En effet, seul l'ID apparaît dans l'événement xAPI.

### 5.1.4 Gestion des événements xAPI

Ces événements sont accessibles à partir du tableau de bord de Wordpress dans la section H5PxAPIkatchu.

Cette interface est basique, mais elle n'est pas prévue pour se substituer à des outils plus évolués. TODO: nous devons voir comment récupérer ces événements facilement et les injecter dans notre base de données MongoDB, à côté des événements learnr et Shiny. Pour rappel, la base de données MySQL n'est **pas** accessible depuis l'extérieur pour des raisons de sécurité et l'accès SSH doit être réactivé manuellement pour une courte durée.



The screenshot shows the WordPress admin dashboard for a site named 'Science des données biologiques'. The left sidebar contains various menu items, with 'H5PxAPIkatchu' highlighted. The main content area displays the 'H5PxAPIkatchu' settings page, which includes a table of xAPI events. The table has columns for Actor Id, Actor Name, Actor Group Members, Verb Id, Verb Display, Object Id, Object Def. Name, and Object Description. There are four rows of data, each representing an xAPI event. Above the table, there are buttons for 'Show/hide columns', 'Download', and 'Delete', and a search bar. The 'Download' button is highlighted in blue.

Actor Id	Actor Name	Actor Group Members	Verb Id	Verb Display	Object Id	Object Def. Name	Object Description
account:1fa4415-2568-467f-b7dd-9558026d1f4b (https://wp.sciviews.org)			http://adlnet.gov/expapi/verbs/attempted	attempted	https://wp.sciviews.org/wp-admin/admin-ajax.php?action=h5p_embed&id=7	multiple-choice	
account:1fa4415-2568-467f-b7dd-9558026d1f4b (https://wp.sciviews.org)			http://adlnet.gov/expapi/verbs/attempted	attempted	https://wp.sciviews.org/wp-admin/admin-ajax.php?action=h5p_embed&id=7	multiple-choice	
account:1fa4415-2568-467f-b7dd-9558026d1f4b (https://wp.sciviews.org)			http://adlnet.gov/expapi/verbs/attempted	attempted	https://wp.sciviews.org/wp-admin/admin-ajax.php?action=h5p_embed&id=8	R markdown	
account:1fa4415-2568-467f-b7dd-9558026d1f4b (https://wp.sciviews.org)			http://adlnet.gov/expapi/verbs/attempted	attempted	https://wp.sciviews.org/wp-admin/admin-ajax.php?action=h5p_embed&id=7	multiple-choice	

Figure 5.4: Événements xAPI dans H5PxAPIkatchu. En bas du tableau, on peut filtrer les données, et le bouton **Download** permet de récupérer le tableau filtré au format CSV. Le lien **Delete** permet de tout effacer et n'est disponible que pour un administrateur du site.



## Chapter 6

# Shiny

...



# Chapter 7

## Moodle

...

### 7.1 Badges

Les badges sont très utiles car ils matérialisent la progression des étudiants. C'est à la fois une façon ludique d'apprendre, et un moyen efficace de pousser un étudiant à travailler. Les badges sont générés dans Moodle, mais bien sûr, ils sont liés à des .... activités Moodle. Avec nos applications externes, nous ne pouvons fournir des badges que manuellement, ce qui devient très vite laborieux. TODO: trouver un moyen d'automatiser l'octroi de badges. Moodle peut se connecter à <https://badgr.io>, et nous pouvons peut-être travailler via son interface ? A creuser !

Pour créer des badges, nous pouvons utiliser une application en ligne. Il suffit d'indiquer son adresse email, et de concevoir son badge. La même interface peut d'ailleurs être utilisée pour concevoir visuellement des hex-stickers (à faible résolution toutefois puisque c'est des pichiers .png de 200x200 pixels). Voici un exemple de badge que nous pourrions utiliser dans nos cours :

Les images correspondant à nos différents badges sont sur p-Cloud, dans le sous-dossier `sdd_badges`.

### 7.2 Notes

TODO: il est possible d'envoyer les notes à Moodle via une URL qui renvoie un fichier XML... Explorer cette piste pour créer et mettre à jour les notes Moodle depuis nos applis...



Figure 7.1: Exemple de badge

## Chapter 8

### H5p

...



## Chapter 9

# Vidéos

...





# Appendix A

## Installation

TODO: installation des outils logiciels nécessaires pour nos cours...



# Références