

# Rapport de stage technicien

## 23 Avril – 02 Septembre 2019

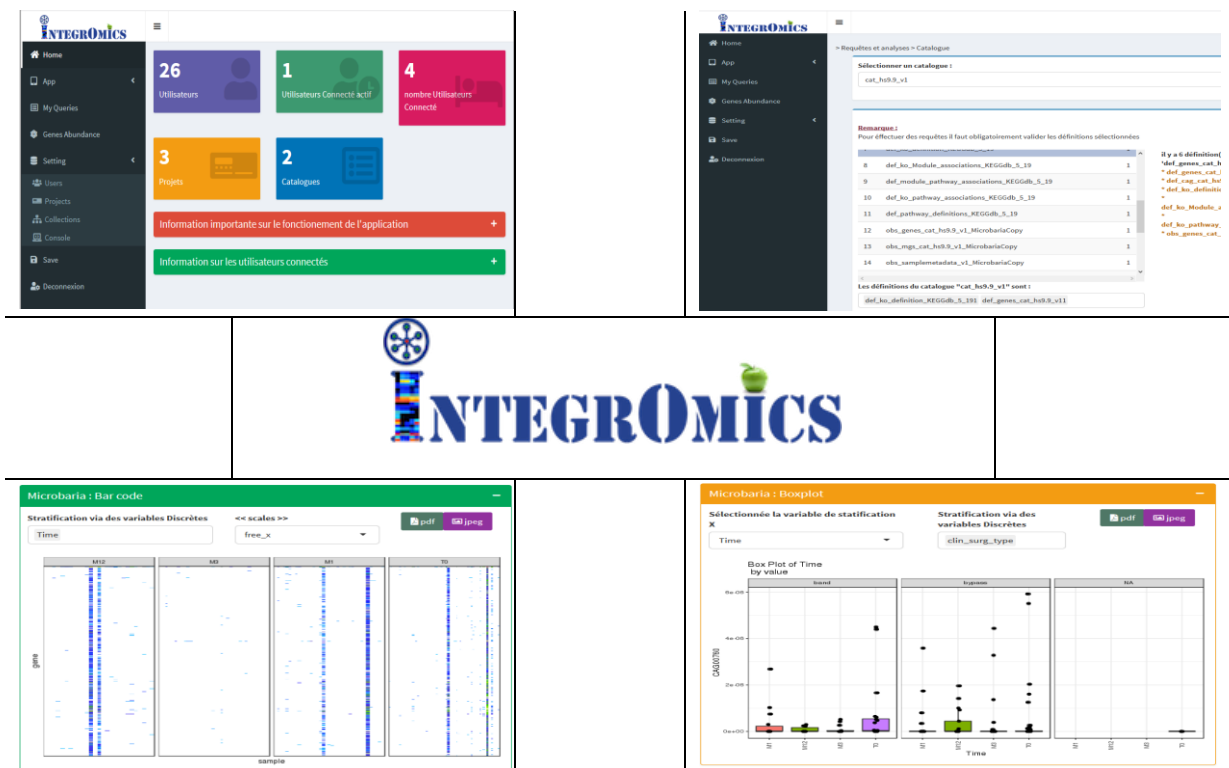


ICAN, 47-83 boulevard de l'hôpital,  
75013 Paris



Université Paris 13, 99 Avenue Jean Baptiste Clément,  
93430 Villetaneuse

Développement d'une application Web permettant de faire des requêtes et des analyses sur des données méta génomiques



Ludovik TEKAM ( [tekamludovik23@gmail.com](mailto:tekamludovik23@gmail.com) )  
Elève Ingénieur en 2<sup>ème</sup> Année Informatique

Maîtres de stage :  
Eugeni BELDA ( [e.belda@ican-institute.org](mailto:e.belda@ican-institute.org) )  
Edi PRIFTI ( [e.prifti@ican-institute.org](mailto:e.prifti@ican-institute.org) )

Tuteur de stage :  
Fatma CHAMEKH ( [fatma.chamekh@univ-paris13.fr](mailto:fatma.chamekh@univ-paris13.fr) )  
Responsable de formation :  
Thierry Hamon ( [thierry.hamon@univ-paris13.fr](mailto:thierry.hamon@univ-paris13.fr) )

Année Universitaire 2018-2019



## Remerciement

Je remercie mes maîtres de stage, Dr. Eugeni Belda et Dr. Edi Prifti pour m'avoir guidé tout au long de mon stage. Merci Edi pour m'avoir appris une nouvelle méthodologie et de nouveaux moyens d'organisation et de conduite de projet. Merci Eugeni de m'avoir guidé et donné des exemples concrets des tâches que je devais réaliser. Je vous remercie également de votre patience et d'avoir pris le temps de m'expliquer des notions qui parfois n'étaient pas liées à ma formation.

Je remercie également Minh Dao Quang, l'ingénieur système de l'ICAN sans qui je n'aurais pas pu avoir certain droit d'accès sur le serveur.

Je tiens également à remercier Florian Specque, stagiaire en bio-informatique avec qui j'ai notamment travaillé sur un module. Ainsi que Daniel Mengue stagiaire juriste avec qui j'ai pu avoir une petite idée de comment fonctionner l'ICAN sur le plan juridique.

Je remercie enfin ma tutrice de stage, pour son aide et ses conseils.

# Résumer

## 1. En Français

Dans le cadre de ma formation d'ingénieur, j'ai réalisé un stage technicien qui s'est déroulé d'avril à septembre au sein de l'Institut de Cardiométabolisme et Nutrition (ICAN). Le choix de ce stage a été fortement motivé par la nature des activités de l'ICAN et des nombreuses compétences que j'aurais pu acquérir en côtoyant leur environnement, en manipulant leur base de données et en travaillant avec leur outil de développement et d'analyse.

Mon stage a consisté au développement d'une application web (<http://iogold.integromics.fr/>) qui permettrait aux chercheurs de l'ICAN d'analyser des données génomiques. On pourra retrouver dans l'application, des interfaces destinées à effectuer des requêtes sur des catalogues méta génomiques et les projets de recherche qui y sont liées. On y retrouvera également des trames de données et des plots qui permettent de représenter visuellement les données résultantes des requêtes effectuées, ce qui facilite leurs interprétations. L'application sera également capable de sauvegarder et de restaurer les données individuelles de chaque utilisateur.

Ce rapport a pour but de présenter d'une part mon travail et les différents moyens que j'ai mis en place afin d'atteindre les objectifs initiaux et d'autre part les leçons que j'ai tirées de cette expérience de cinq mois. Pour cela, je vais tout d'abord vous présenter l'ICAN et les missions qui m'ont été confiées. Puis, je vous parlerai du déroulement du stage et de comment j'ai réalisé les différentes tâches qui m'étaient affectées. Enfin, je vous exposerai les perspectives d'évolution de l'application avant de terminer avec le bilan de mon stage.

## 2. En Anglais

As part of my engineering training, I have done a technician internship that have takes place from april to September within the Institute of Cardiometabolism And Nutrition (ICAN). The choice of this internship was strongly motivated by the nature of the activities of the ICAN and the many skills that I could have acquired by rubbing shoulders with their environment, by manipulating their database and by working with their tool of development and analysis.

My internship involved the development of a web application (<http://iogold.integromics.fr/>) that will allow ICAN researchers to explore and analyze metagenomics data. The application will include query interfaces on meta genomic gene catalogs and related research projects. It will also include data frames and plots that can visually represent the resulting data of queries made, which facilitates their interpretations. The application will also be able to save and restore the individual data of each user.

The purpose of this report is to present, firstly, my work and the various means I have put in place to achieve the initial objectives and, secondly, the lessons I have learned from this five-month experience. For that, I will first introduce you to the ICAN and the missions that were entrusted to me. Then I will tell you about the course of the internship and how I developed the various tasks that were assigned to me. Finally I will expose you the perspectives of evolution of the application before finishing with the report of my internship.

## Sommaire

Remerciement.....	3
Résumer .....	4
1. En Français .....	4
2. En Anglais .....	4
Sommaire .....	5
Introduction .....	7
I- L'Institut de Cardiométabolisme et Nutrition (ICAN) .....	8
1. Présentation.....	8
2. Mission : .....	8
II- Présentation du projet et des missions .....	9
1. Le projet .....	9
a. Le context.....	9
b. Objectif principale.....	9
c. Les données .....	9
d. Les compétences requises .....	12
2. Les missions.....	12
a. Missions initial.....	12
b. Evolution des missions .....	12
III- Déroulement du stage .....	13
1. Les outils à disposition .....	13
a. Les logiciels.....	13
b. Le matériel .....	13
2. État du projet au 23 avril 2019.....	13
3. Réalisations des missions .....	16
a. La base de données.....	16
b. Structure de l'application.....	17
c. Gestion des sessions et sauvegarde des données .....	18
d. Les principaux modules.....	19
e. Les modules fonctionnels .....	23
IV- Perspective d'évolution et remarque .....	29
1. Niveau applicatif .....	29
2. Niveau structuration du code .....	29

3. Niveau interface .....	29
4. Niveau base de données .....	29
5. Niveau sauvegarde des données .....	29
6. Niveau Conception.....	30
7. Niveau fonctionnalité.....	30
V- Bilan du stage .....	31
1. Les résultats obtenus sur la productivité et la gestion du temps .....	31
2. Les problèmes ou difficultés rencontrés.....	31
3. Les solutions apportées .....	32
4. Les connaissances et compétences acquises.....	32
5. Mes attentes vs la réalité .....	32
Conclusion.....	33
Table des matières .....	34
Annexe .....	36
Webographie .....	36
Documents .....	36

## Introduction

Du 19 Avril au 2 Septembre 2019, j'ai effectué mon stage technicien au sein de l'IHU ICAN (Institut de Cardiométabolisme et Nutrition) à Paris. Ce stage m'a non seulement permis de manipuler de grands volumes de données, mais aussi de développer ma première application web ( <http://iogold.integromics.fr/> ) destiné aux analyses biomédical.

Mon stage s'est déroulé dans le département INTEGROMICS qui est constitué d'une équipe de bio-informaticiens qui luttent contre l'obésité, le diabète, les maladies cardiovasculaires, la NASH et les dyslipidémies en se servant d'outil informatique. La mission qui m'avait été confiés était celle du développement d'une application qui faciliterait certaines tâches quotidiennes, donc la recherche d'information sur des patients, sur des gènes ou des Espèces Méta génomique (MGS), mais aussi l'analyse des données résultantes et leur visualisation.

Comme la plupart des étudiants, avant d'obtenir ce stage j'ai effectué de nombreuses demande. Toutefois, j'ai orienté mes recherches vers les des entreprises qui sont axées sur analyse de données pour la résolution de problèmes quotidiens. Le secteur d'activité de l'entreprise ou encore la nature de ces activités n'avait pas beaucoup d'importance dans le choix de mon stage, je souhaite uniquement apprendre une méthodologie de travail sur de grands volumes de données qui me serait utile pour la suite de mon parcours professionnel.

Ce rapport de stage contiendra mon retour d'expérience, mais aussi un descriptif des missions effectuées sans toutefois entrées dans des détails pointus du développement. Cependant, afin de vous permettre d'avoir accès aux détaillées techniques des processus de résolution des problèmes rencontrés et de facilité la reprise de mon projet par d'autres développeurs, j'ai rédigé une documentation complète du projet, qui sera fournir en pièce jointe de ce rapport.

# I- L'Institut de Cardiométabolisme et Nutrition (ICAN)

## 1. Présentation

L'institut de Cardiométabolisme et Nutrition (ICAN) développe la médecine du futur dans le domaine du cardiométabolisme et de la nutrition.



Situé à l'hôpital de la Pitié-Salpêtrière à Paris, ICAN a été officiellement inauguré en Novembre 2011 parmi les six centres d'excellence scientifique en France. Le financement de cet Institut a été accordé dans le cadre du programme d'investissements d'avenir initié par le gouvernement français pour faire face aux enjeux de santé en Europe et dans le monde.

ICAN s'appuie sur les forces et l'expertise des unités de recherche médicale et scientifique de l'INSERM et de l'Université Pierre et Marie Curie et des équipes médicales de l'AP-HP.

La communauté ICAN rassemble des chercheurs et des cliniciens du cœur et du métabolisme ainsi que des paramédicaux (infirmiers, diététiciens, psychologues...).

## 2. Mission :

ICAN vise à lutter contre l'obésité, le diabète, les maladies cardiovasculaires, la NASH (stéatose hépatique non alcoolique) et les dyslipidémies en traduisant les découvertes engendrées par la Recherche centrée sur le patient en innovations thérapeutiques et diagnostiques. Pour cela, ICAN met en œuvre une approche pluridisciplinaire

Les grands objectifs d'ICAN sont de :

- Développer une recherche translationnelle dans le domaine des maladies cardiométaboliques à l'échelle internationale
- Développer une médecine personnalisée dans laquelle les innovations sont traduites en soins
- Former les futurs professionnels de santé
- Valoriser la recherche via des partenariats public/privé
- Disséminer la connaissance scientifique au sein des professionnels et du grand public

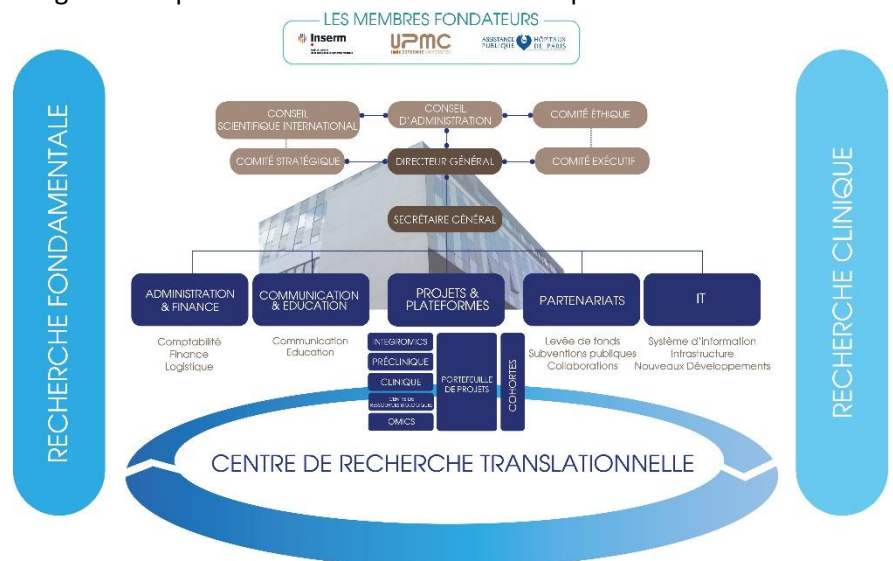


Figure I-1 : organisation de l'ICAN



## II- Présentation du projet et des missions

### 1. Le projet

#### a. Le contexte

Le microbiome intestinal humain (i.e. ensemble des microorganismes habitant notre intestin) est fortement associé à notre santé. Sa dérégulation ou dysbiose peut être à l'origine ou refléter de multiples maladies chroniques humaines, comme l'obésité, le diabète, différents cancers, les maladies inflammatoires de l'intestin etc. Ces espèces bactériennes sont difficiles à étudier car il est compliqué de les isoler. La métagénomique est un nouveau domaine en forte explosion qui permet d'analyser très précisément la composition du microbiote ainsi que ses fonctions.

Différents projets internationaux ont généré de grands volumes de données de séquences génomiques (fragments d'ADN) à partir des milliers d'échantillons de l'intestin humain. Ces données ont été utilisées pour reconstruire des catalogues de gènes de référence du microbiome humain représentant le contenu génétique des espèces microbiennes le constituant. Par des approches d'apprentissage non supervisés nous avons pu établir un catalogue d'espèces métagénomique (MGS), qui sont définis comme des ensembles de gènes provenant du même génome bactérien basé sur la corrélation des abondances des gènes sur plusieurs échantillons. Certaines de ces MGS sont des cibles d'intérêt et peuvent aider à traiter différentes maladies chroniques.

Le problème est qu'on connaît peu de choses sur ces entités et qu'il y en a beaucoup qui ont des interactions entre elles. Nous souhaiterions rassembler de la connaissance pertinente et construire des outils qui nous permettent de mieux voir et comprendre ces espèces et leur rôle dans la physiopathologie de l'hôte.

#### b. Objectif principale

L'objectif principale est de développement d'une application web permettant d'exploiter les résultats de diverses requêtes sur une base de données déjà existante pour ensuite analyser les résultats via des graphes et de tableaux explicites.

#### c. Les données

Les données initialement stockées dans MongoDB étaient :

- Un catalogue de gènes (**hs9.9**) du microbiome intestinal issue du projet metahit et publiée dans <https://www.ncbi.nlm.nih.gov/pubmed/24997786>

qui contenait les collections suivantes :

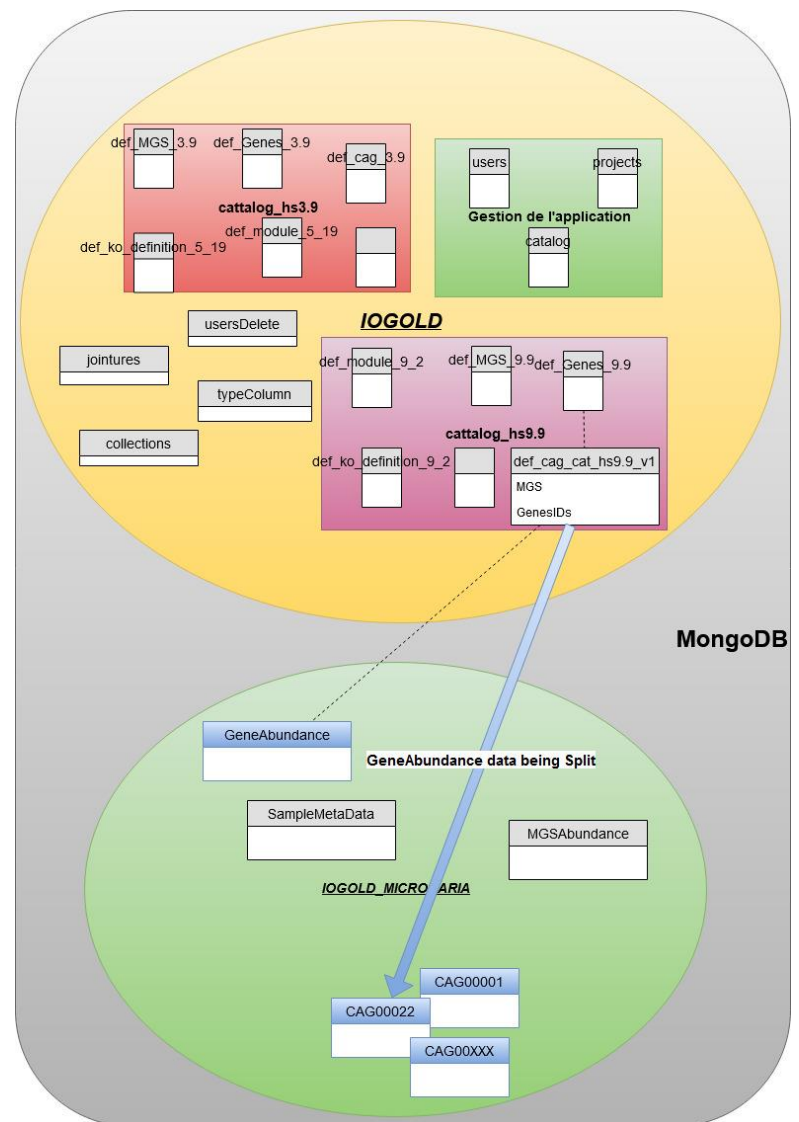


Figure II-1: vue partiel des bases de données et collections dans MongoDB

- **def\_genes\_cat\_hs9.9\_v1** : collection décrivant les 9,9 millions de gènes du catalogue.  
Description repartir sur 36 variables :

```
[1] "GeneID"           "GeneName"
[3] "GeneLength"       "GeneCompletenessStatus"
[5] "CohortOrigin"     "TaxonomicPhylumAnnotation"
[7] "TaxonomicGenusAnnotation" "KOs"
[9] "eggNOGs"          "SampleOccurrenceFrequency"
[11] "IndividualOccurrenceFrequency" "KEGGFunctionalCategories"
[13] "eggNOGFunctionalCategories" "CohortAssembled"
[15] "interpro"         "GO"
[17] "KEGG_MetaCyc_Reactome" "Pfam"
[19] "CDD"              "Coils"
[21] "Gene3D"           "Hamap"
[23] "Pirsf"            "PRINTS"
[25] "Phobius"          "ProDom"
[27] "PrositesPatterns" "PrositesProfiles"
[29] "SMART"            "SUPERFAMILY"
[31] "SignalP_GRAM_NEGATIVE" "SignalP_GRAM_POSITIVE"
[33] "TIGRFAM"          "TMHMM"
[35] "KOs_embl_metacardis" "MGS"
```

Exemple : les dix premiers gènes de la collection

	GeneID	GeneName	GeneLength	GeneCompletenessStatus	CohortOrigin
MH0206_GL0172211	23968	MH0206_GL0172211	4782	Lack 5-end	EUR
MH0206_GL0056796	35463	MH0206_GL0056796	4275	Complete	EUR
MH0206_GL0107433	36279	MH0206_GL0107433	4251	Complete	EUR
MH0206_GL0072030	49145	MH0206_GL0072030	3906	Lack 3-end	EUR
MH0206_GL0183386	113025	MH0206_GL0183386	3132	Lack both ends	EUR
MH0206_GL0155300	179244	MH0206_GL0155300	2697	Lack both ends	EUR
MH0206_GL0093999	181536	MH0206_GL0093999	2685	Complete	EUR
MH0439_GL0161824	218469	MH0439_GL0161824	2538	Lack both ends	EUR
MH0206_GL0136728	236739	MH0206_GL0136728	2472	Complete	EUR
MH0206_GL0191060	255298	MH0206_GL0191060	2415	Lack both ends	EUR

Figure II-2: les dix première lignes de la collection des gènes

- **def\_mgstaxonomy\_cat\_hs9.9\_v1** : collection décrivant les 3463 MGS ( espèces méta génomiques)  
Variable descriptive :

```
[1] "size"           "NA_pc"           "BHit_pc"         "BHit"           "BH_ali"
[6] "BH_id"          "NA_pc_"          "Taxo_pc"         "Taxo"           "Taxo_ali"
[11] "Taxo_id"        "Taxo_level"      "annot"           "species"         "genus"
[16] "family"         "order"           "class"           "phylum"        "superkingdom"
```

Exemple : les dix premiers MGS de la collection

	size	NA_pc	BHit_pc	BHit	BH_ali	BH_id	NA_p
CAG00001	12638	78.1	14.5	Blastocystis hominis	65.2	80.5	
CAG00002	11313	90.8	4.7	Blastocystis hominis	71.4	82.7	
CAG00003	10882	75.8	16.0	Blastocystis hominis	65.7	80.8	
CAG00004	10466	21.6	77.9	Blastocystis sp. ST4 (obsolete)	98.0	99.7	
CAG00008	6646	0.0	98.9	[Clostridium] bolteae 90A5	99.2	99.1	
CAG00009	6333	14.1	62.0	Lachnospiraceae bacterium 3_1_57FAA_CT1	91.9	83.5	
CAG00011	5523	0.0	20.9	Escherichia coli D9	98.7	99.5	
CAG00012	5363	0.0	66.0	Klebsiella pneumoniae subsp. pneumoniae HS11286	99.7	99.5	
CAG00013	5283	5.4	76.7	Clostridiales bacterium 1_7_47FAA	95.4	91.5	
CAG00014	5115	1.4	96.1	Lachnospiraceae bacterium 3_1_57FAA_CT1	99.0	98.1	

Figure II-3: les dix première lignes de la collection des MGS

- Un ensemble de collections provenant de la base de données KEGG<sup>1</sup> (<https://www.genome.jp/kegg/>) qui permettent de mettre les annotations fonctionnelles des gènes du catalogue basées sur cette espace fonctionnel dans un contexte biologique plus large (modules fonctionnels, voies métaboliques)

Liste des collections :

```
[1] "def_module_definitions_KEGGdb_5_19" : Annotation des modules fonctionnels
[2] "def_ko_definition_KEGGdb_5_19" : Annotation fonctionnel des groupe KO
[3] "def_ko_Module_associations_KEGGdb_5_19" : Associations des KO groups lies aux
different modules
[4] "def_ko_pathway_associations_KEGGdb_5_19"
[5] "def_pathway_definitions_KEGGdb_5_19"
[6] "def module pathway associations KEGGdb 5 19"
```

- Un ensemble de collections qui représente un projet de recherche (**Microbaria** <https://www.ncbi.nlm.nih.gov/pubmed/29899081>) où les abondances des gènes et des MGS ont été quantifiées sur un ensemble de patients sévèrement obèses avant et après une chirurgie bariatrique.

Ce projet est constitué de trois collections :

- ❖ Collection d'abondances de gène : cette collection est une quantification des gènes du catalogue **hs9.9**

	MB16_3	MB50_1	MB08_1	MB37_3	MB29_1	MB12_1	MB54_3	MB33_1
T2D-6A_GL0083352	0.0000e+00	0.0000e+00	8.7478e-10	0.0000e+00	8.3510e-10	0.0000e+00	8.6934e-10	5.6150
V1.UC4-5_GL0154511	0.0000e+00	8.5815e-10	2.8915e-08	2.8453e-09	1.6729e-09	0.0000e+00	0.0000e+00	3.9932
MH0198_GL0136826	0.0000e+00	0.0000e+00	0.0000e+00	0.0000e+00	0.0000e+00	0.0000e+00	0.0000e+00	6.0845
V1.FI28_GL0106390	0.0000e+00	0.0000e+00	8.1841e-08	0.0000e+00	0.0000e+00	0.0000e+00	0.0000e+00	0.0000
SZEY-41A_GL0088289	1.9682e-09	3.3513e-07	6.1331e-07	1.1537e-07	4.2841e-07	2.8747e-07	1.5423e-07	1.3031
O2.UC30-1_GL0131651	0.0000e+00	0.0000e+00	0.0000e+00	0.0000e+00	3.8614e-09	0.0000e+00	0.0000e+00	5.5635
MH0169_GL0104706	0.0000e+00	0.0000e+00	0.0000e+00	0.0000e+00	0.0000e+00	0.0000e+00	0.0000e+00	1.8923
MH0422_GL0083623	0.0000e+00	0.0000e+00	0.0000e+00	0.0000e+00	5.9181e-09	0.0000e+00	0.0000e+00	9.4744
MH0369_GL0168249	0.0000e+00	6.0813e-09	0.0000e+00	0.0000e+00	7.5083e-08	0.0000e+00	0.0000e+00	1.7651
MH0385_GL0138259	0.0000e+00	0.0000e+00	0.0000e+00	0.0000e+00	0.0000e+00	0.0000e+00	0.0000e+00	0.0000

Figure II-4: dix première lignes de la collection d'abondance des gènes

- ❖ Collection d'abondance de MGS : cette collection est une quantification des espèces méta génomiques du catalogue **hs9.9**

	MB16_3	MB50_1	MB08_1	MB37_3	MB29_1	MB12_1	MB54_3	MB33_1	MB59_1
CAG00001	0.0000e+00	0.0000e+00	0.0000e+00	0.0000e+00	0.0000e+00	0.0000e+00	0.0000e+00	6.5715e-07	0.0000
CAG00002	0.0000e+00	0.0000e+00	0.0000e+00	0.0000e+00	0.0000e+00	0.0000e+00	0.0000e+00	0.0000e+00	0.0000
CAG00003	0.0000e+00	0.0000e+00	0.0000e+00	0.0000e+00	0.0000e+00	0.0000e+00	0.0000e+00	0.0000e+00	0.0000
CAG00004	0.0000e+00	0.0000e+00	0.0000e+00	0.0000e+00	0.0000e+00	0.0000e+00	0.0000e+00	0.0000e+00	0.0000
CAG00008	2.5506e-07	1.4644e-08	0.0000e+00	1.4792e-06	9.7458e-08	2.0929e-08	9.6302e-08	0.0000e+00	2.110
CAG00009	0.0000e+00	0.0000e+00	0.0000e+00	0.0000e+00	0.0000e+00	0.0000e+00	0.0000e+00	0.0000e+00	0.0000
CAG00011	5.8112e-06	0.0000e+00	4.1096e-08	2.1250e-07	1.9818e-08	3.9405e-07	1.9171e-06	1.7950e-07	5.793
CAG00012	0.0000e+00	0.0000e+00	0.0000e+00	7.3125e-07	0.0000e+00	0.0000e+00	5.4762e-07	0.0000e+00	0.0000
CAG00013	2.2589e-07	0.0000e+00	0.0000e+00	2.9229e-07	3.6622e-08	9.6339e-09	8.3791e-08	0.0000e+00	1.297
CAG00014	1.1821e-07	9.5425e-09	8.9206e-09	1.7688e-07	1.1901e-08	2.8656e-08	2.7458e-08	0.0000e+00	3.040

Figure II-5 : dix première lignes de la collection d'abondance des mgs

<sup>1</sup> KEGG désigne un ensemble de bases de données relatives aux génomes, aux voies métaboliques et aux composés biochimiques

❖ Collection de métadonnées : contient les informations sur les patients avant et après chirurgie.

	down_3M_una	down_5M_una	down_10M_una	down_11M_una	down_15M_una	down_20M_una
MB01_3	291234	362828	481312	499527	563891	6
MB01_1	353382	433508	558414	577088	640050	7
MB01_2	284942	350373	453615	469188	523034	5
MB02_2	359720	449838	593066	614992	688938	7
MB02_1	385463	478627	624960	646922	721694	7
MB04_2	259214	324241	431320	447852	505755	5
MB04_3	411864	516604	686256	712419	801051	8
MB04_1	388247	480542	625153	646766	720169	7
MB04_4	510041	632485	822508	850405	945541	10
MB05_2	465121	567326	721633	744441	820670	8

Figure II-6: dix première lignes de la collection des métadonnées

#### d. Les compétences requises

Pour mener à bien ce projet il faut idéalement savoir coder en **langage R**, avoir des connaissances en **NoSQL** et connaître les concepts de la **programmation web**. Cependant, des connaissances basic en programmation web, moyenne en SQL et avancer en algorithmique et programmation orienté objet sont suffisantes.

## 2. Les missions

### a. Missions initial

- Optimisation des requêtes sur MongoDB avec notamment parallélisation et optimisation des performances.
- Développement des interfaces pour l'analyse des données méta génomiques et l'exploration de biomarqueurs.
- Intégration d'outils R développées par l'équipe pour la quantification des fonctions métaboliques et pour le placement phylogénétique des espèces métagénomiques
- LA mise en production de la plateforme dans les serveurs de l'équipe et réalisation de tests unitaires et d'intégration.
- Fournir une documentation du projet

### b. Evolution des missions

Au cours de mon stage, mes missions ont évolué. La partie développement a été privilégier à la partir analyse. Donc il a fallu bien structurer l'application et de s'assurer de la bonne sauvegarde et récupération des données. En plus de cela, se sont rajoutés de nouvelles tâches :

- Optimiser l'application afin de sauvegarder les données sur le serveur avec un espace de sauvegarde spécifique à chaque utilisateur
- Développer un module permettant d'exécuter du script R dans l'application
- Ajouter un onglet **log** qui permettra d'avoir un retour sur le fonctionnement de l'application notamment les éventuelles erreurs ainsi que des notifications d'exécution de certaines fonctions
- Optimisation de l'interface de requêtes sur les catalogues et projets

### III- Déroulement du stage

Pour ce stage, j'ai dû partir de zéro ou presque, dans cette partie je vais vous présenter :

- Les outils que j'ai utilisés dans la réalisation de ce projet,
- Une vue globale du projet avant que je commence mon stage,
- Le processus de réalisation des différentes tâches qui m'ont été confiées

#### 1. Les outils à disposition

Tout projet nécessite l'utilisation d'un certain nombre d'outils qu'ils soient matériels ou logiciels.

##### a. Les logiciels

L'ICAN à mi à ma disposition

- Un compte **RStudio**<sup>2</sup> en ligne, ce qui est assez pratique vu que je pouvais me connecter sur n'importe quel ordinateur et développer sans soucis. (<http://icr2.integromics.fr/>)
- Un accès à **MongoDB**<sup>3</sup>, avec des droits limité mais suffisant pour les tâche qui m'était confier. Cependant, pour certaines tâches il a fallu m'ajouter des droits notamment en ce qui concerne l'accès à certains fichiers qui contenait des données à jours ou encore à des fichiers de configuration. (<https://www.mongodb.com/fr>)
- Un compte sur leur **GitLab**<sup>4</sup> pour gérer les différentes versions du projet. ([https://git.integromics.fr/users/sign\\_in](https://git.integromics.fr/users/sign_in))
- Un compte sur leur serveur qui était lier à la clés **SSH**<sup>5</sup> de mon ordinateur personnel

##### b. Le matériel

J'ai eu droit à un ordinateur de bureaux directement relie au réseaux de l'ICAN et à un espace de travail assez confortable



Figure III-1: poste de travail ICAN

#### 2. État du projet au 23 avril 2019

Ce projet avait déjà été commencé par d'autres équipes et stagiaires avant le début de mon stage.

- Le premier fut un stagiaire qui a travaillé dessus au cour de l'été 2018, cependant, son travailler c'était limité à la structuration des données et au peuplement de la base de données.

Voici le diagramme de classe des données du catalogue qu'il a mis en place

---

<sup>2</sup> RStudio est un environnement de développement gratuit, libre et multiplateforme pour R, un langage de programmation utilisé pour le traitement de données et l'analyse statistique.

<sup>3</sup> MongoDB est un système de gestion de base de données orienté documents, répartitionnable sur un nombre quelconque d'ordinateurs et ne nécessitant pas de schéma prédéfini des données

<sup>4</sup> GitLab est un logiciel libre de forge basé sur git proposant les fonctionnalités de wiki, un système de suivi des bugs, l'intégration continue et la livraison continue.

<sup>5</sup> Secure SHell (**SSH**) est à la fois un programme informatique et un protocole de communication sécurisé. Le protocole de connexion impose un échange de **clés** de chiffrement en début de connexion

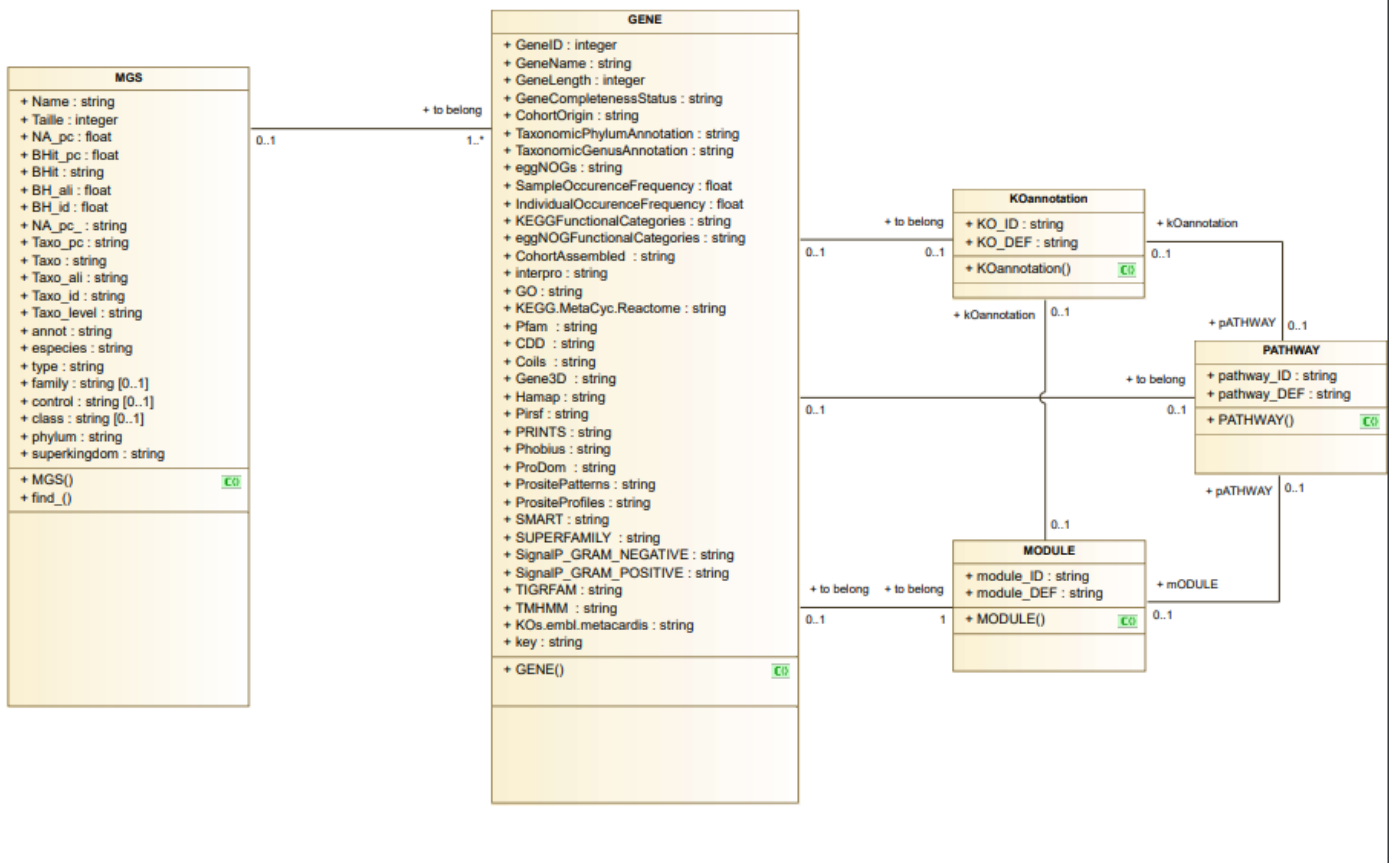


Figure9 : diagramme de classe des données du catalogue

- Les seconds furent un groupe d'étudiants auquel je faisais partie, nous avons
  - Régénérer et rajouter de nouvelle collection à la base de données car elle avait été supprimer suite à une panne des serveurs de l'ICAN.
  - Développer une application web pour récupérer les données de la base de données et effectué quelques visualisations avec les résultats obtenus

### Interface des requêtes

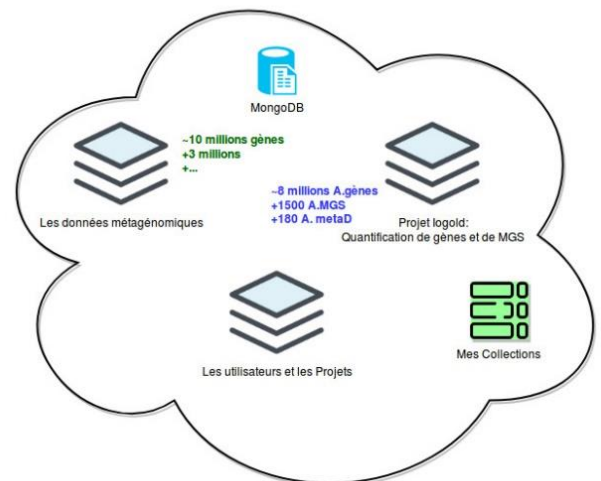


Figure 10: Vu sur les collections MongoDB



**Choisir une Collection pour la Requête indirecte**

ko\_definition

**Choisir une Variable pour la Requête indirecte**

V2

**Saisir la valeur recherchée**

alcoh

☒ Uniquement les Genes avec MGS

Valider la Recherche

*Figure III-2: interface de requêtage sur les collections*

A. Res. Indirect. A. Res. Simple R  storer Requet. Sauvegarder Requet.

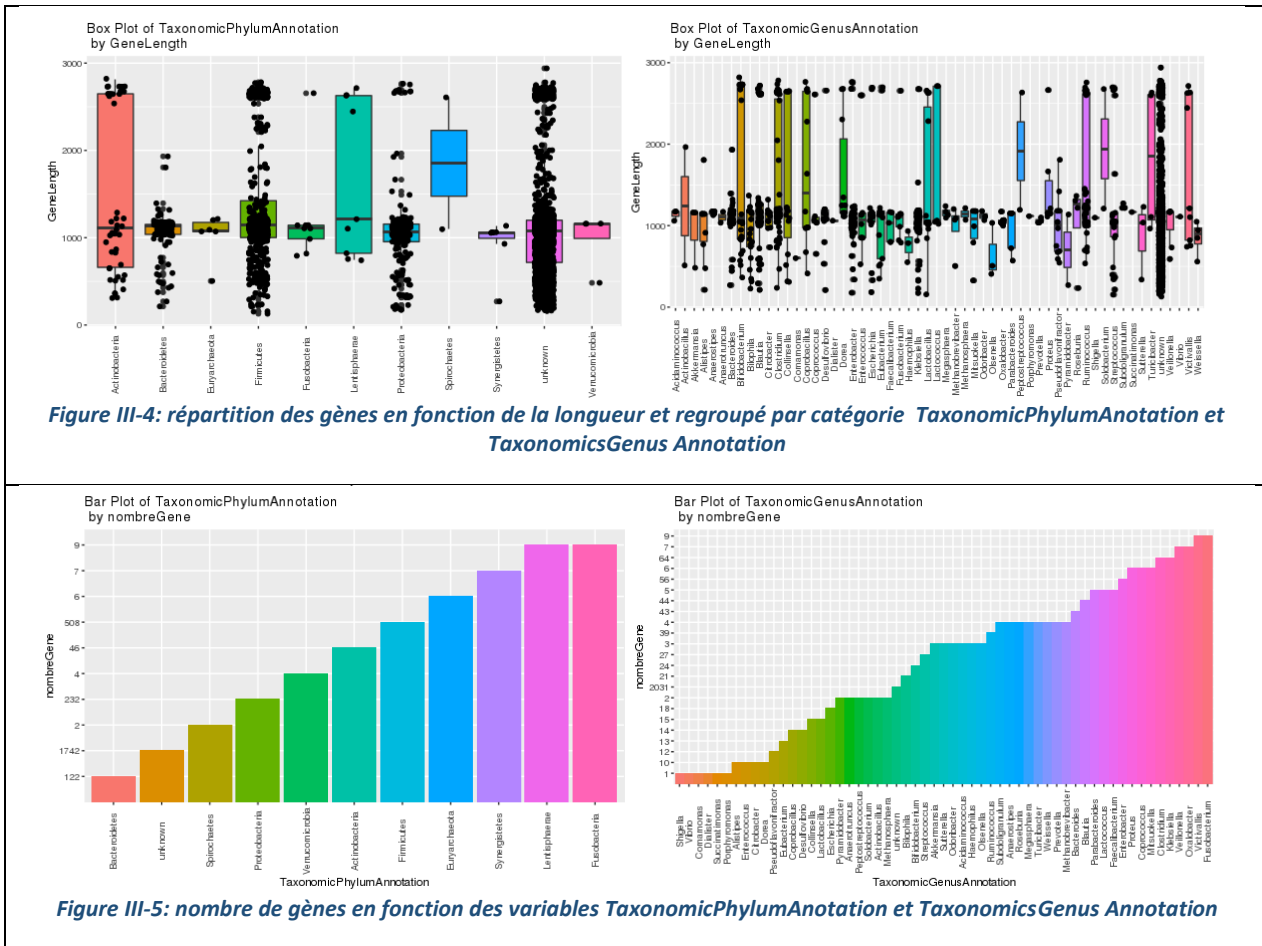
[ Requet. Indirect. ] Les Donn  es [ Requet. Simple ] Les Donn  es Historique Des Requetes

Show 10 entries Search:

GeneID	KOs_embi_metacardis	datagene
171949	K04072	5c44ce45e03df3dc0b76abe0_K04072.adhE; acetaldehyde dehydrogenase / alcohol dehydrogenase [EC:1.2.1.10.1.1.1]
2712010	K04072	5c44ce45e03df3dc0b76abe0_K04072.adhE; acetaldehyde dehydrogenase / alcohol dehydrogenase [EC:1.2.1.10.1.1.1]
1655485	K00001	5c44ce45e03df3dc0b769d0a_K00001.E1.1.1.1; adh; alcohol dehydrogenase [EC:1.1.1.1]
1539469	K00001	5c44ce45e03df3dc0b769d0a_K00001.E1.1.1.1; adh; alcohol dehydrogenase [EC:1.1.1.1]
2213912	K18369	5c44ce45e03df3dc0b76e24d_K18369.adh2; alcohol dehydrogenase [EC:1.1.1.-]
199596	K04072	5c44ce45e03df3dc0b76abe0_K04072.adhE; acetaldehyde dehydrogenase / alcohol dehydrogenase [EC:1.2.1.10.1.1.1]
739805	K04072	5c44ce45e03df3dc0b76abe0_K04072.adhE; acetaldehyde dehydrogenase / alcohol dehydrogenase [EC:1.2.1.10.1.1.1]

*Figure III-3: visualisation des r  sultats d'une requ  te sur la collection ko\_definition*

### Quelques plots de repr  sentation visuel des r  sultats obtenu



Le fait d'avoir d  j travaill   sur ce projet m'a aid      bien commencer mon stage. Malgr   tout, j'ai d   repartir de z  ro, car la conception de l'application n'avait pas   t   bien r  fl  ch  es. Et il   tait plus simple de repartir de z  ro que d'essayer de

corriger les défauts de l'application existante. Toutefois, j'ai conservé les fonctions d'accès à la base de données et les fonctions de visualisation qui pouvaient être utilisées à l'extérieur de l'application donc facile à améliorer et à réutiliser.

### 3. Réalisations des missions

#### a. La base de données

##### i. Optimisation des requêtes

L'une de mes premières tâches avait été d'optimiser la vitesse d'exécution des requêtes effectuées sur la base de données. En effet, certaines collections sont assez volumineuses et rechercher une information peut parfois être couteuse en temps, j'ai donc rajouté des index spécifiques aux requêtes fréquemment effectuées.

L'ajout d'un index permet simplement de spécifier sur quelles colonnes les recherches sont le fréquemment émises, cela m'a permis dans certains cas d'augmenter jusqu'à 10 fois la vitesse d'obtention des résultats.

Exemple : supposons que je souhaite obtenir la liste des gènes qui ont pour **GeneID** = [1,2,7,9,5000,4000,458...]

Si la collection n'a pas d'index sur la colonne **GeneID**, la requête prendra probablement 10 secondes. Mais si je rajoute un index sur cette colonne elle prendra moins de temps, car on ne regardera que les **GeneID** et non l'ensemble de la ligne.

Bien sûr il y a d'autres méthodes d'optimisation des requêtes, que je détaille dans la documentation, mais l'ajout d'index est la plus performante.

##### ii. Fonctions d'administration de la base de données

Après avoir optimisé les requêtes j'ai développé et optimisé de nombreuses fonctions d'ajout de collections, de projets et de catalogues. De cette manière, les mises à jour de données ou la reconstruction de la base de données, en cas de panne des serveurs, seraient plus simple.

Ces fonctions vont être très utile pour l'administrateur de l'application. Pour cette raison, j'ai également ajouté un module<sup>6</sup> sur l'application (module de gestion des projets). Ce module est accessible uniquement par les administrateurs et permet de rajouter des données pas très volumineuse sur l'application à l'aide d'une interface assez simple.

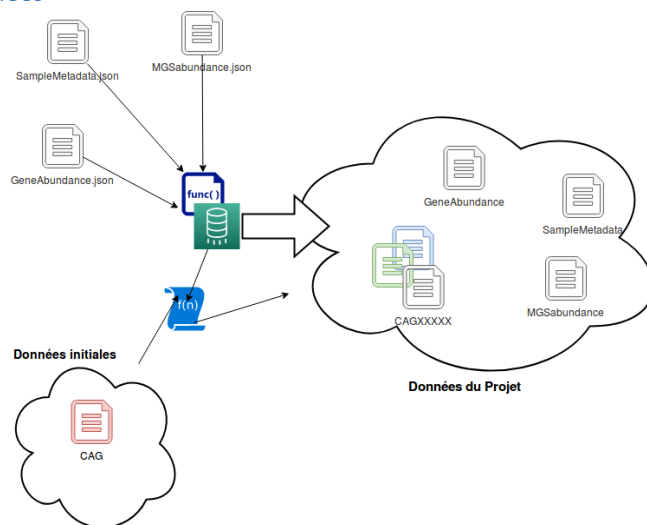


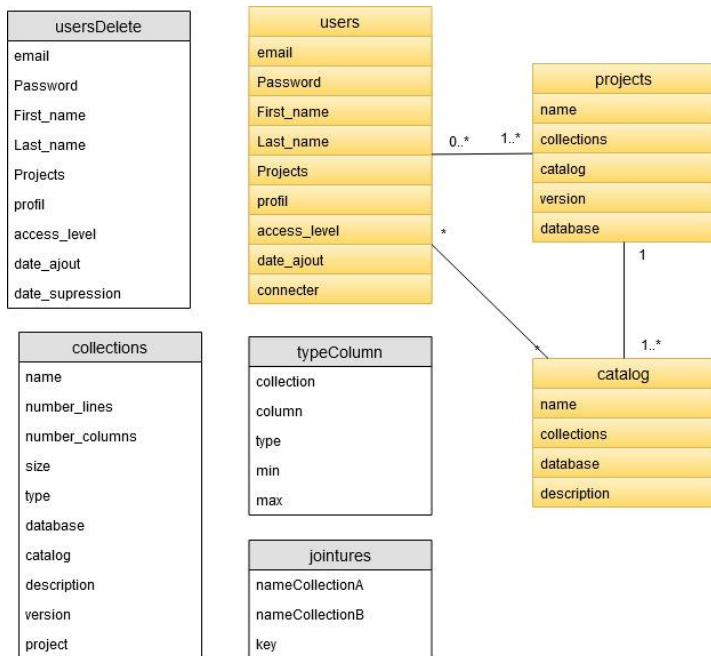
Figure III-6: illustration de l'ajout des données d'un projet dans mongoDB

<sup>6</sup> Fonctionnalité de l'application qui traite d'une tâche particulière et qui peut être dépendante ou non d'autres modules de l'application et ou de l'application elle-même



### iii. La nouvelle architecture de la base de données

Afin de mieux gérer l'application, il a fallu rajouter de nombreuses collections donc une collection de gestion des utilisateurs, de gestion des projets, des catalogues, des jointures et bien d'autre.



Il a également fallu redéfinir l'architecture des bases de données ce qui nous amène à l'architecture final ci-dessous :

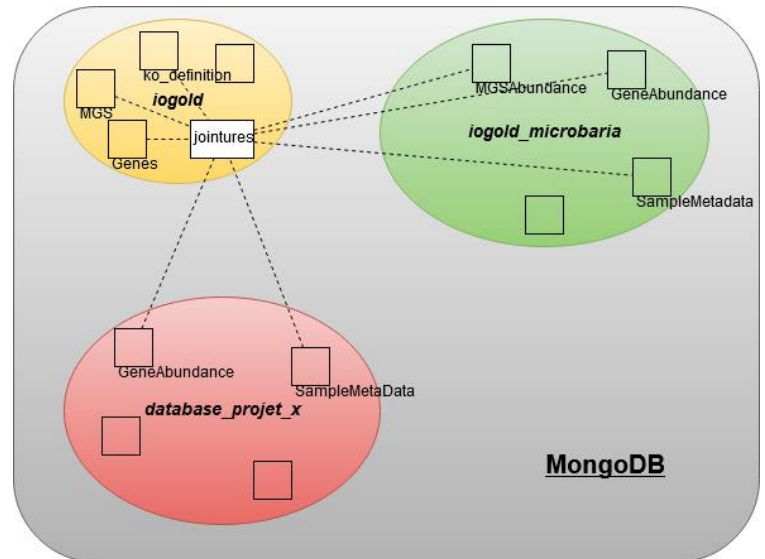


Figure III-7: Diagramme de classe de la gestion des utilisateur , projet et catalogue de l'application.

Figure IV – 7 : visualisation du rôle de la collection jointure dans MongoDB

### b. Structure de l'application

Chaque application à une structuration unique, une arborescence de fichiers et répertoire propre aux fonctionnalités à développer et aux évolutions possible.

Pour cette application, après quelques réunions on s'est fixé la structuration ci-contre.

Pour avoir des informations détailler sur le contenu de chaque fichier, vous pouvez consulter la documentation du projet (partie sur la structuration de la description de l'arborescence de l'application) Ici je ne présenterais que partiellement le contenu des fichiers et répertoires.

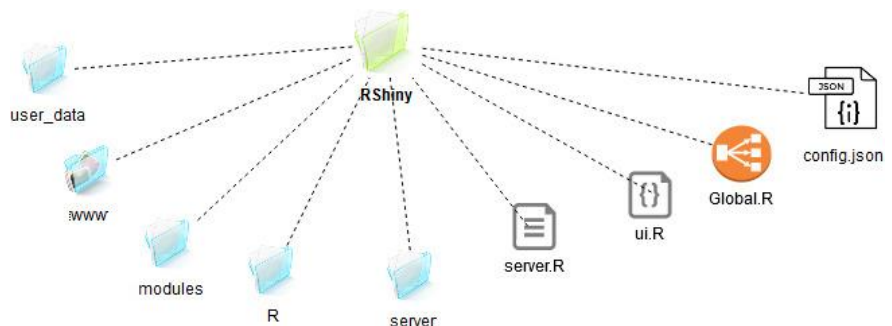


Figure III-8: structure des fichier principaux du développement de l'application

- **Config.json** : c'est le fichier de configuration de l'application. Il contient toutes les informations qui permettent de lancer et de faire tourner l'application donc notamment
- **Global.R** : c'est le plus important des trois fichiers principaux de l'application. Il contient :

- La liste des librairies utilisées lors du développement
- Les instructions d'appel des fonctions que j'ai développées
- Une variable `app`, qui contient les informations sur le chargement des modules, la liste des utilisateurs connectés et des données qu'il utilise
- **Ui.R** : permet de définir la structure visuelle de l'application en appelant notamment la structure de chaque module.
- **Server et server.R** : c'est qu'est définie les structures de variables de chaque module (Server /serverVariable.R), les appels au serveur de chaque module (au code qui permet d'exécuter de tâche en fonction de l'ui défini )
- **Modules** : contient les différents modules de l'application
- **R** : contient les différentes fonctions que j'ai définies afin de faciliter certaines opérations (la génération de plot, l'ajout de collection, la consultation de la base de données, la création de la base de données)
- **www** : contient toutes les images utilisées par l'appli
- **user\_data** : correspond au répertoire de gestion des sessions. Pour le moment, un répertoire plus sécurisé n'a pas encore été créé. Une fois qu'il aura été défini, il suffira de modifier le chemin dans le fichier de configuration principale.

### c. Gestion des sessions et sauvegarde des données

La réalisation de ces tâches est indispensable au bon fonctionnement de toute application. L'objectif ici étant de gérer les sessions de chaque utilisateur, sauvegarder et restaurer les données des utilisateurs, concevoir et implémenter un système de gestion de fichiers avec une arborescence intuitive, amovible et facile à modifier pour les futurs super administrateurs.

Pour réaliser ces tâches j'ai dû apporter des modifications à la fois sur le code de l'application (voir documentation du projet : module sauvegarde) mais j'ai également créé un répertoire sur le serveur qui contiendra toutes les données produites par l'application. La difficulté était d'éviter de répéter des liens dans tout le code et de pouvoir les changer facilement. Pour cette raison, j'ai défini un fichier de configuration dans lequel on retrouve les informations importantes donc notamment le chemin vers le répertoire des sessions utilisateurs

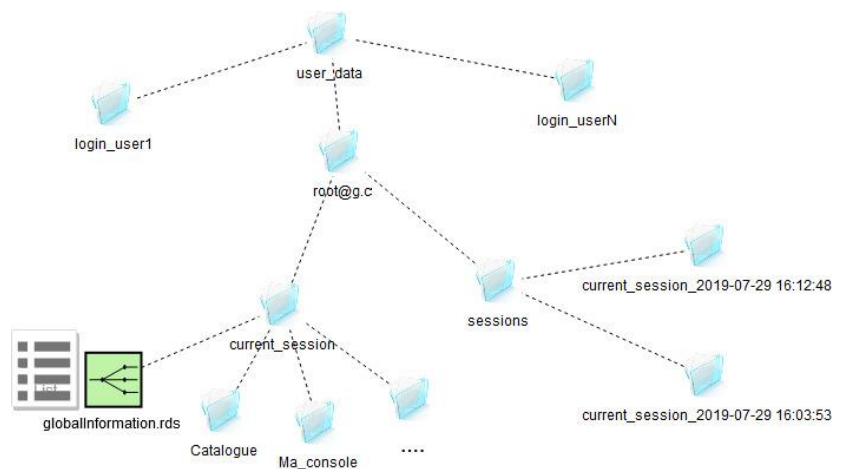


Figure III-9 : arborescence de sauvegarde données utilisateurs

```
{
  "server_id" : "portal",
  "activated":0,
  "default_user_email":"root@g.c",
  ...
  ...

  "workDirectory":"/share/apps/iogold/iogoldapp/user_data",
  "appDirectory":"/share/apps/iogold/iogoldapp",
  "pathToModule":"/share/apps/iogold/iogoldapp/modules/",
  ...
  ...
},
```

#### d. Les principaux modules

Dans cette partie je présenterai les modules de gestion de l'application. Ce sont les modules que l'on retrouve généralement dans toutes les applications

##### i. Connexion

Ce module permet à l'utilisateur de se connecter à l'application.

En fonction de son niveau d'accès (0 à 10), il aura accès, après connexion, à certaine fonctionnalité et pas à d'autre.

Les niveaux d'accès permettent de mieux spécifier quel utilisateur a droit d'accès à une fonctionnalité x et pas à une fonctionnalité y. De cette manière deux chercheurs n'auront pas forcément le même menu. Il en est de même pour les administrateurs

#### Exemple

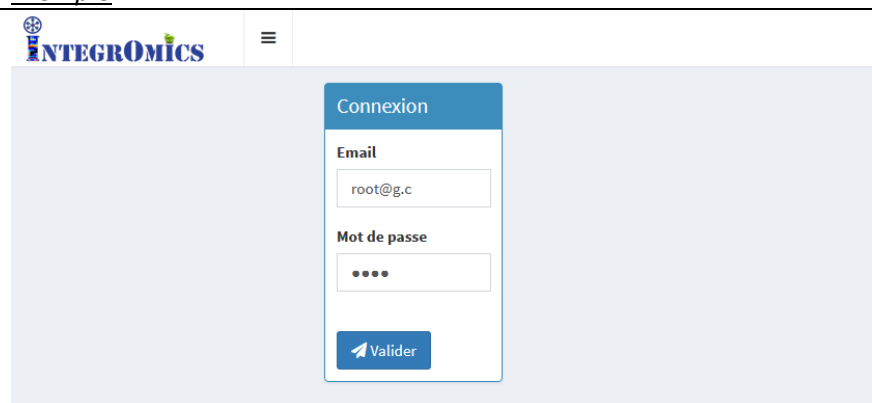


Figure III-10 : Interface de connexion

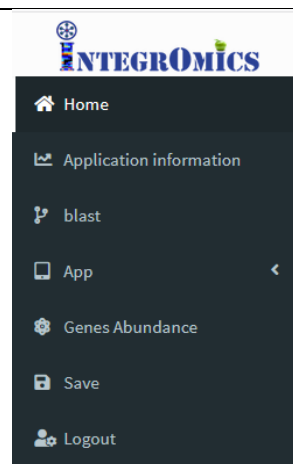


Figure III-11 : Menu d'un utilisateur de niveau 5 (compte chercheur)

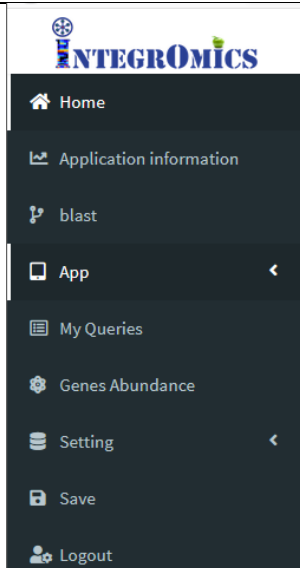


Figure III-12 : Menu d'un utilisateur de niveau 10 (compte administrateur)

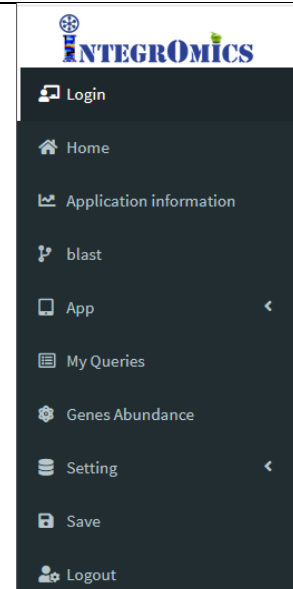


Figure III-13 : Menu d'un utilisateur de niveau 11 (compte super administrateur)

## ii. Accueil

Présente un résumé de ce que fait l'application, à qui elle est destinée et comment l'utiliser

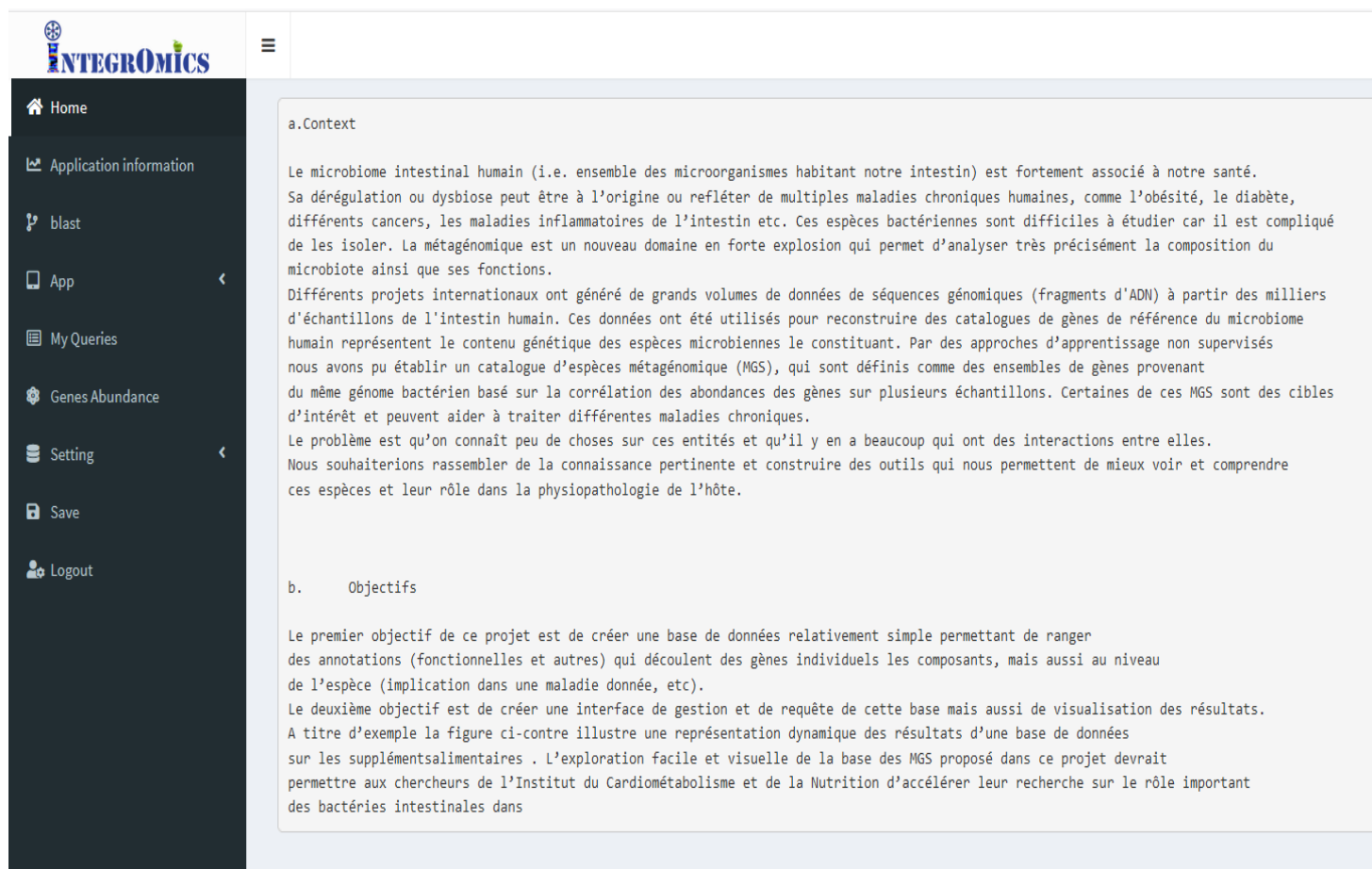


Figure III-14 : capture d'écran de l'onglet "home" de l'application INTEGROMICS

## iii. Information application (module administrateur)

Présente quelques statistiques sur les utilisateurs connectés, le nombre de projets, de catalogues, les paramètres de configuration.

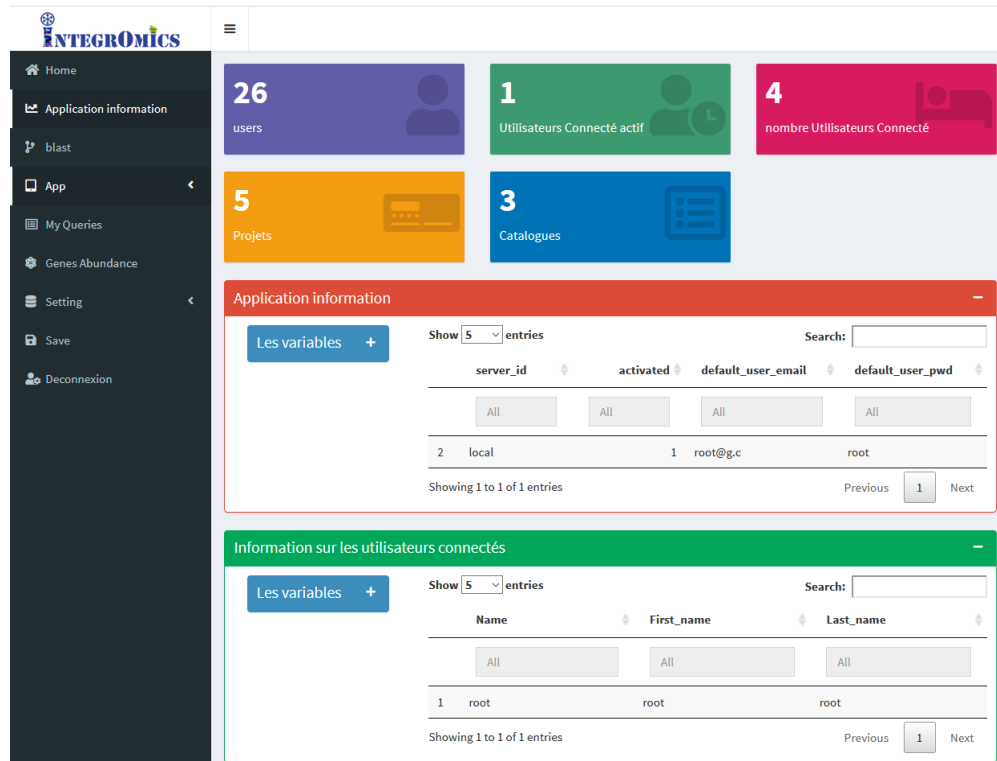


Figure III-15 : capture d'écran de l'onglet "Applicationinformation" de l'application INTEGROMICS

#### iv. Gestion des utilisateurs (module administrateur)

Comme son nom l'indice, ce module permet de gérer les utilisateurs, on pourra non seulement ajouter, modifier et supprimer un utilisateur mais aussi affecter un projet à un utilisateur

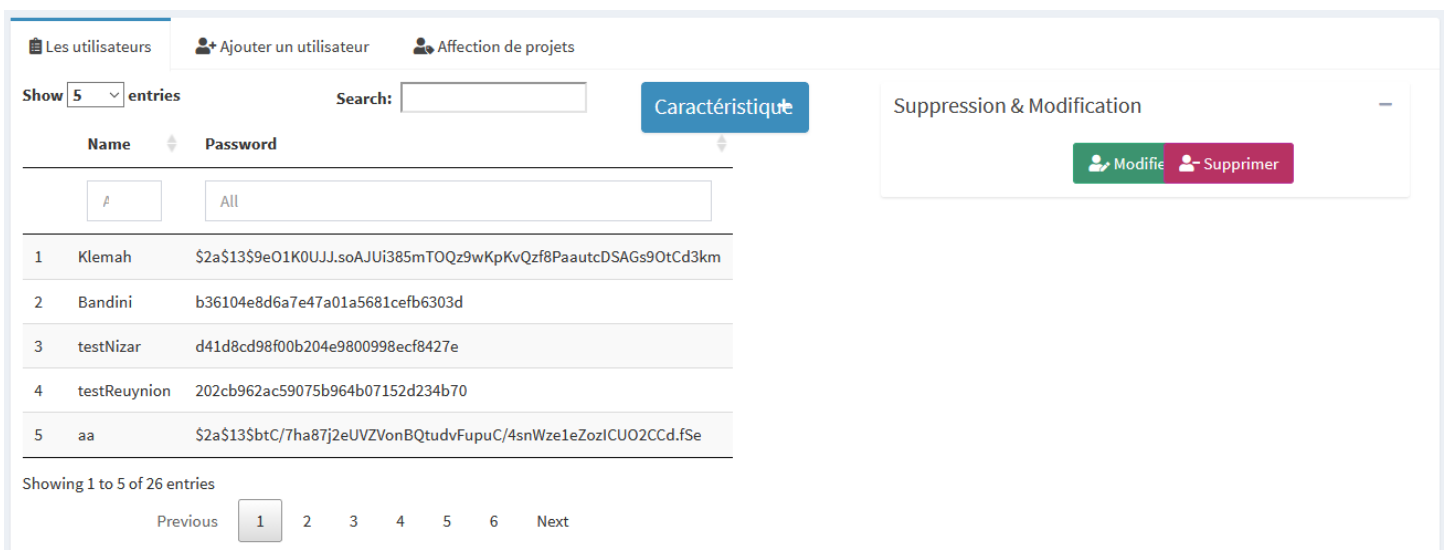


Figure III-16 : capture d'écran de l'onglet "users" de l'application INTEGROMICS

#### v. Gestion des projets (module administrateur)

Ce module est destiné à l'ajout de nouvelles collections, cependant, il n'est efficace que pour des collections de capacités inférieures à 5MG. Cette capacité max peut être modifiée, mais il faudra faire beaucoup de tests pour s'assurer que la nouvelle capacité est bien prise en compte, car dans certain cas même en modifiant la capacité, le chargement des fichiers peut être erronée.

#### vi. Collections

Permet de visualiser toutes les collections utiliser par l'application, afin d'avoir une meilleure visibilité des données que l'on manipule

On pourra également à partir de cette interface supprimer des collections si besoin

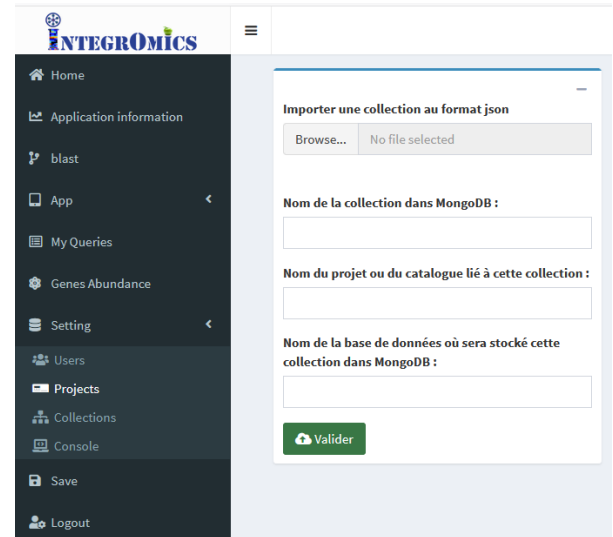


Figure III-17 : capture d'écran de l'onglet "projects" de l'application INTEGROMICS

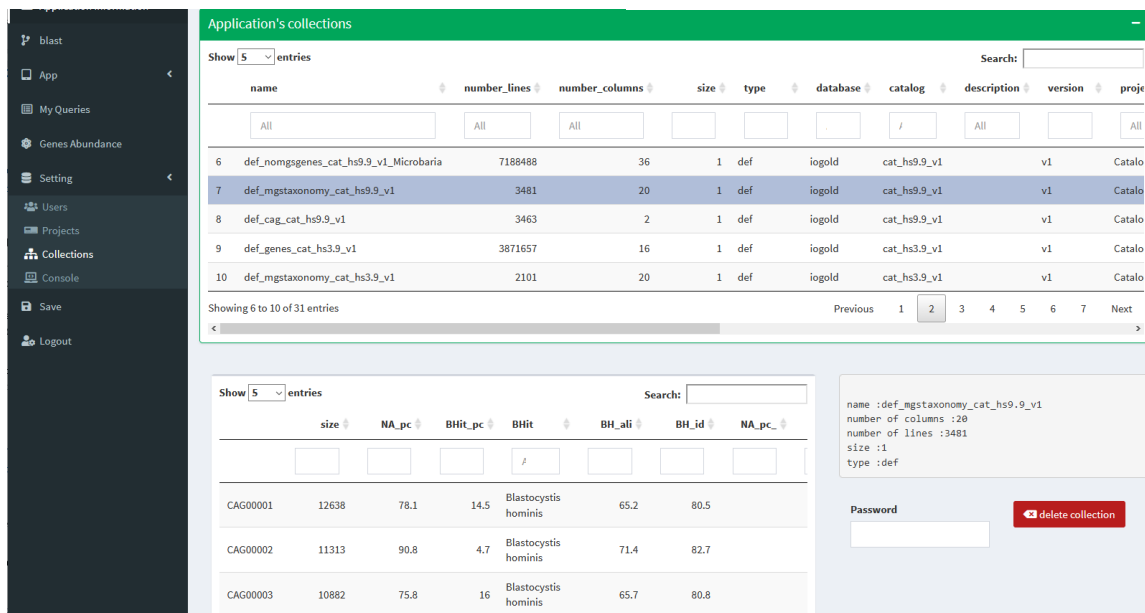


Figure III-18 : capture d'écran de l'onglet "collections" de l'application INTEGROMICS

#### vii. Déconnexion

Ce module permet à l'utilisateur de se déconnecter et de modifier la langue. Quand j'ai conceptualisé ce module, il devrait également contenir un descriptif de l'utilisateur connecter, quelques statiques sur ces anciennes connexions et la possibilité de vérifier que les sauvegardes sont bien effectuées. N'ayant pas eu le temps de tout implémenter, je les ai rajoutés à la documentation comme perspective d'évolution du module

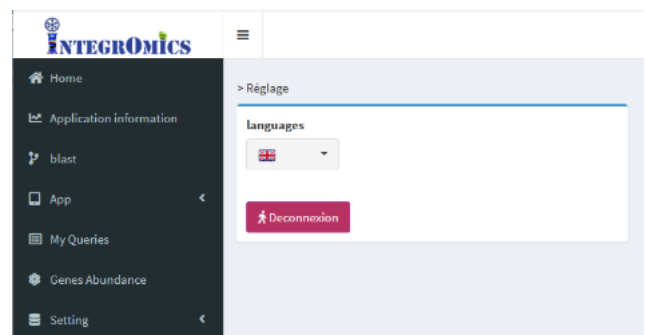
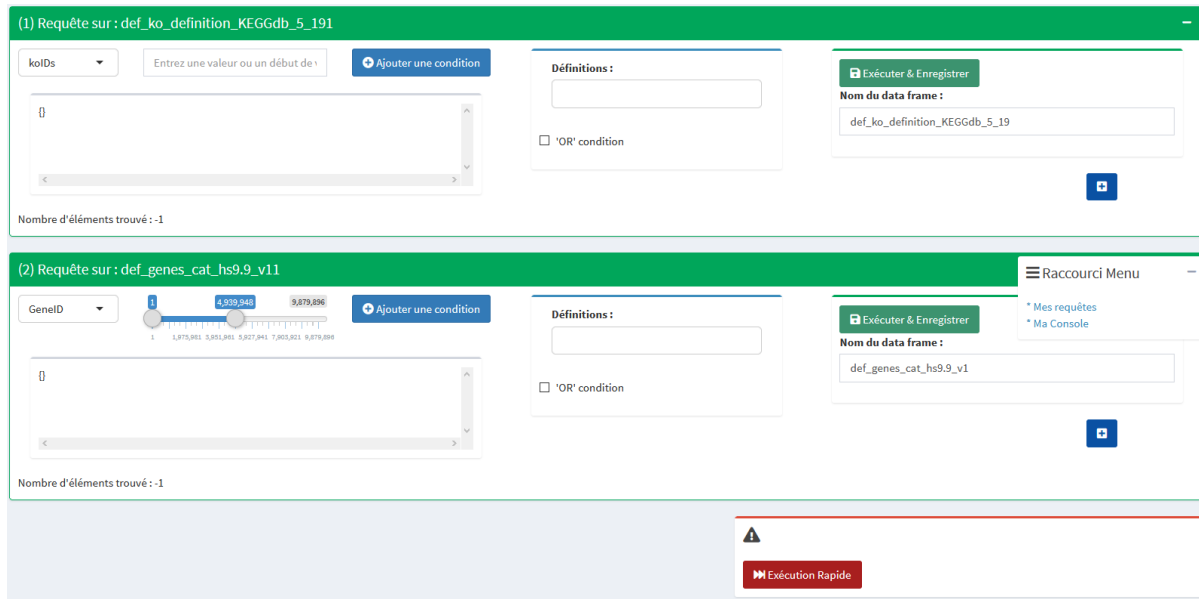


Figure III-19 : capture d'écran de l'onglet "logOut" de l'application INTEGROMICS

## e. Les modules fonctionnels

### i. Générateur de requêtes

C'est le module sur lequel j'ai le plus travaillé. Il permet de personnaliser des requêtes et d'en sauvegarder les résultats. Ces résultats seront ensuite utilisés par d'autres modules pour par exemple générer des graphiques, filtrer d'autres données ou tout simplement pour répertorier les requêtes importantes et réutilisables par d'autres chercheurs.

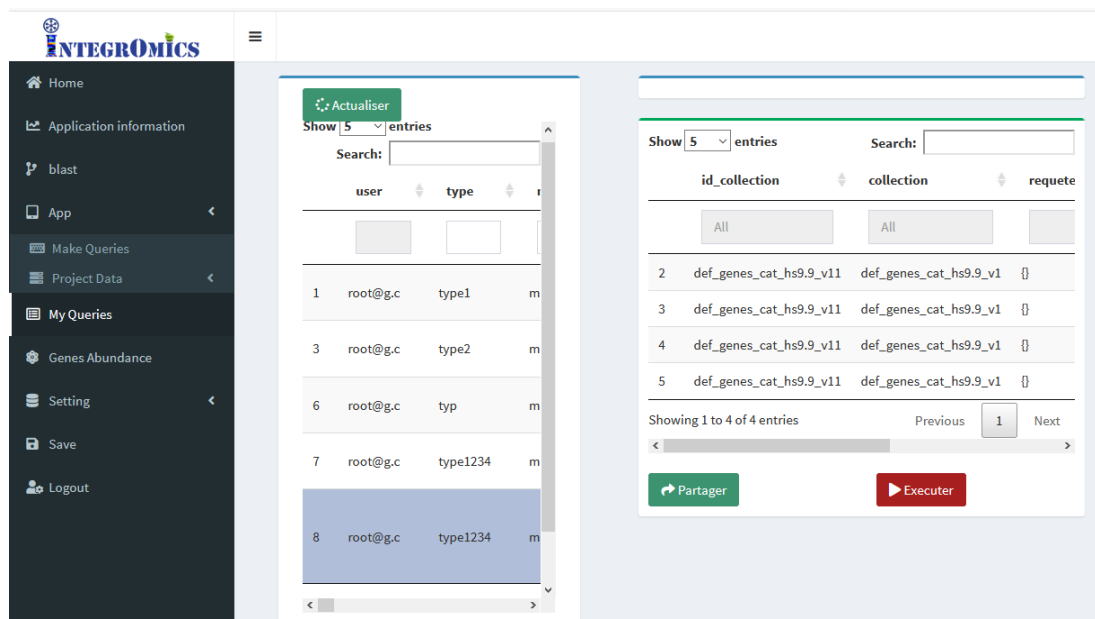


The screenshot shows two panels of the query generator interface. Panel (1) is titled 'Requête sur : def\_ko\_definition\_KEGGdb\_5\_191' and features a 'koIds' dropdown, a search bar, and an 'Ajouter une condition' button. Panel (2) is titled 'Requête sur : def\_genes\_cat\_hs9.9\_v11' and features a 'GeneID' dropdown, a search bar, and an 'Ajouter une condition' button. Both panels include a 'Définitions' section with an 'OR' condition checkbox and an 'Exécuter & Enregistrer' button. A 'Raccourci Menu' sidebar is visible on the right of panel (2).

Figure III-20 : interface permettant de faire des requêtes sur les collections de projets et de catalogues

### ii. Consultation des requêtes

Ce module permet de visualiser les requêtes qui ont été sauvegardées sur le disque. Ce qui permet de consulter les résultats des requêtes sans les ré-exécuter, on pourra également les modifier et les exécuter si besoin.



The screenshot shows the query consultation interface. On the left is a sidebar with navigation links: Home, Application information, blast, App, Make Queries, Project Data, My Queries, Genes Abundance, Setting, Save, and Logout. The main area displays a table of saved queries with columns for 'id\_collection', 'collection', and 'requete'. The table shows 5 entries. Below the table, there are buttons for 'Partager' and 'Exécuter'.

	id_collection	collection	requete
2	def_genes_cat_hs9.9_v11	def_genes_cat_hs9.9_v1	{}
3	def_genes_cat_hs9.9_v11	def_genes_cat_hs9.9_v1	{}
4	def_genes_cat_hs9.9_v11	def_genes_cat_hs9.9_v1	{}
5	def_genes_cat_hs9.9_v11	def_genes_cat_hs9.9_v1	{}

Figure III-21 : interface permettant de visualiser les requêtes sauvegardées

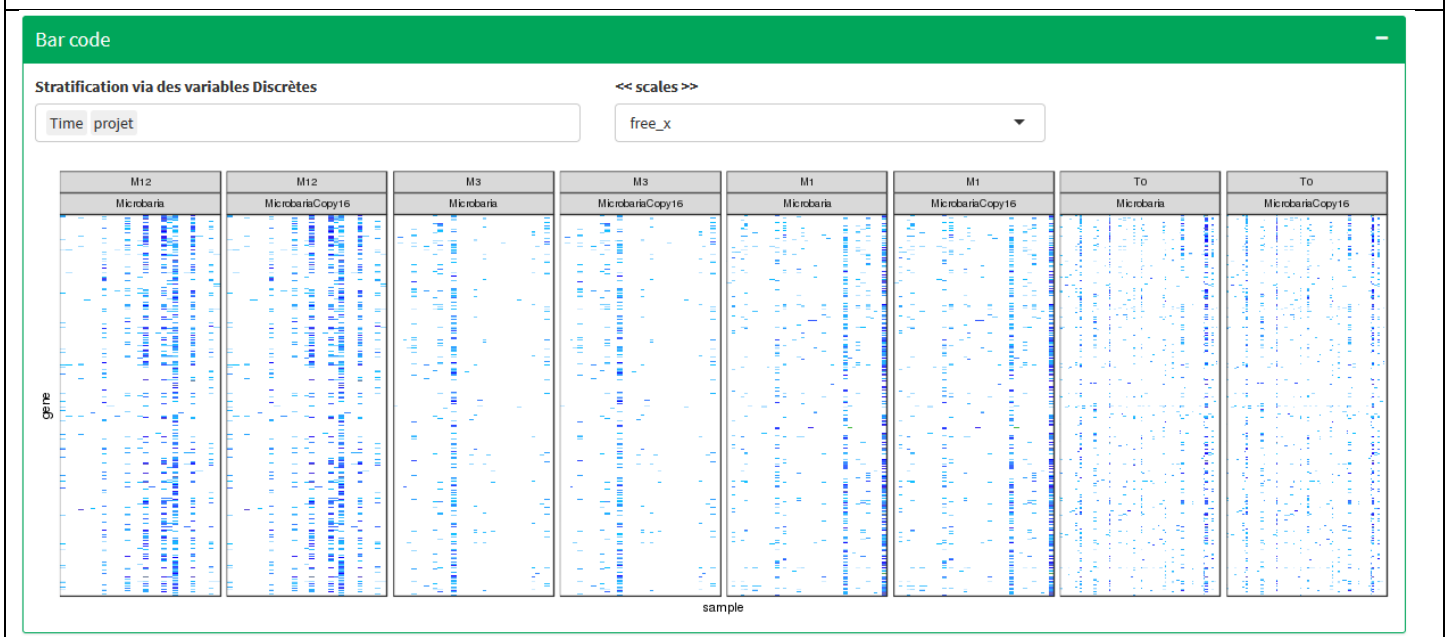
### iii. Analyse et visualisation des données projets

Ce module traite uniquement des données de projet, en particulier celle des abondances de MGS<sup>7</sup>, il permet notamment de :

- Comparer des projets

Après avoir sélectionné un projet je peux les comparer à l'aide de graphique explicite. Dans l'exemple ci-dessous je compare le MGS **CAG00004** de deux projets (actuellement il n'y a qu'un seul projet donc j'ai dupliqué les données, c'est pourquoi sur l'image les des MGS sont identique).

Contrairement, je compare les abondances de gènes d'un même MGS (axe des y) sur les différents échantillons (axe des x). Le résultat de cette comparaison peut être stratifié en fonction des différentes métadonnées du projet Microbaria.



- Visualise des plots spécifiques aux analyses médical

Notamment des barres plot, box plot et autre

Exemple : visualisation de l'évolution de la quantification du MGS **CAG001386** au cours de temps

<sup>7</sup> MGS ( metagenomics species)



## Microbaria : Boxplot

Sélectionnée la variable de stratification  
X

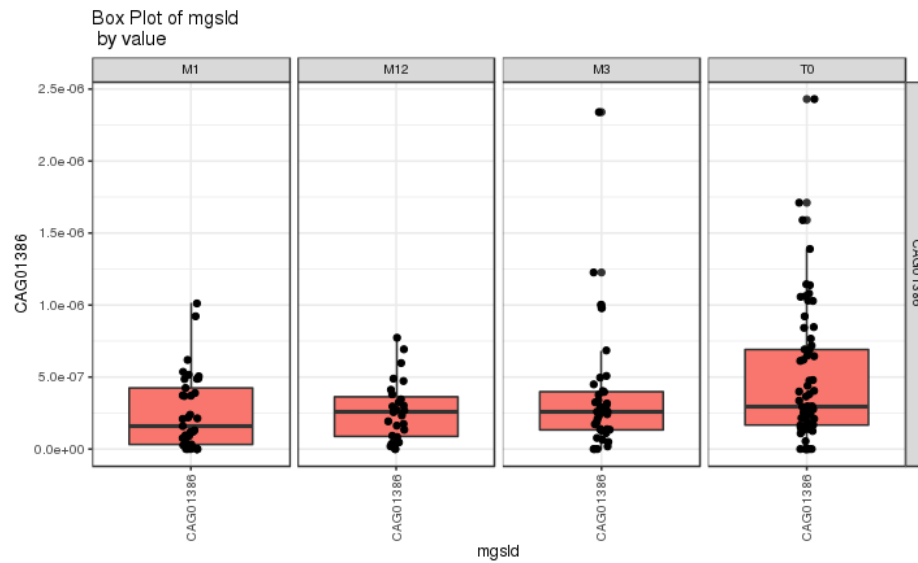
mgslid

Stratification via des  
variables Discrètes

mgslid Time

pdf

jpeg



Exemple : visualisation de l'évolution de la quantification du MGS **CAG001386** au cour de temps pour chaque type de chirurgie

## Microbaria : Bar code

Stratification via des variables Discrètes

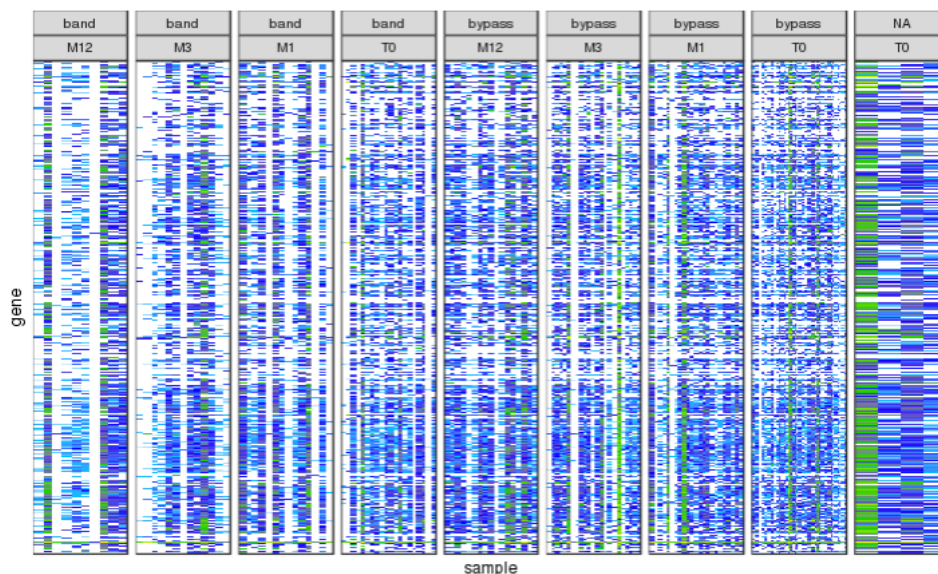
clin\_surg\_type Time

<< scales >>

free\_x

pdf

jpeg



#### iv. Gestion des sauvegardes

C'est l'un des modules qui a pris le plus de temps à mettre en place. Il a d'abord fallu revoir toute la structuration du code, les variables normales, les variables réactives (celles qui peuvent réagir au différent évènement). Puis définir une structure de données adaptées pour la sauvegarde des données au niveau applicatif (au niveau du code) et en enfin sauvegarder les données, dans des répertoires adaptés à chaque utilisateur, sur le serveur.

Au niveau de l'application, ce module permet simplement de gérer la périodicité de sauvegarde des données. Pour chaque module

Remarque : Il faut noter ici que lorsqu'on définit un module, on doit définir également la manière avec laquelle les données seront sauvegardées et comment elles seront restaurées. Donc contrairement aux autres modules, ce module est destiné à évoluer avec l'application.

Sur l'application actuel, tous les modules ne respectent pas ce format de sauvegarde, vu qu'il est assez long a implémenté, je n'ai pas eu le temps de définir cette méthode que pour deux modules, le module **ma console** et le **module de génération de requêtes** qui sont les plus modules plus important de l'application. Donc, tous utilisateurs qui se connectera et fera des opérations sur l'un de ses deux modules aura ces données sauvegardées sur le serveur et restaurer en cas re-connexion

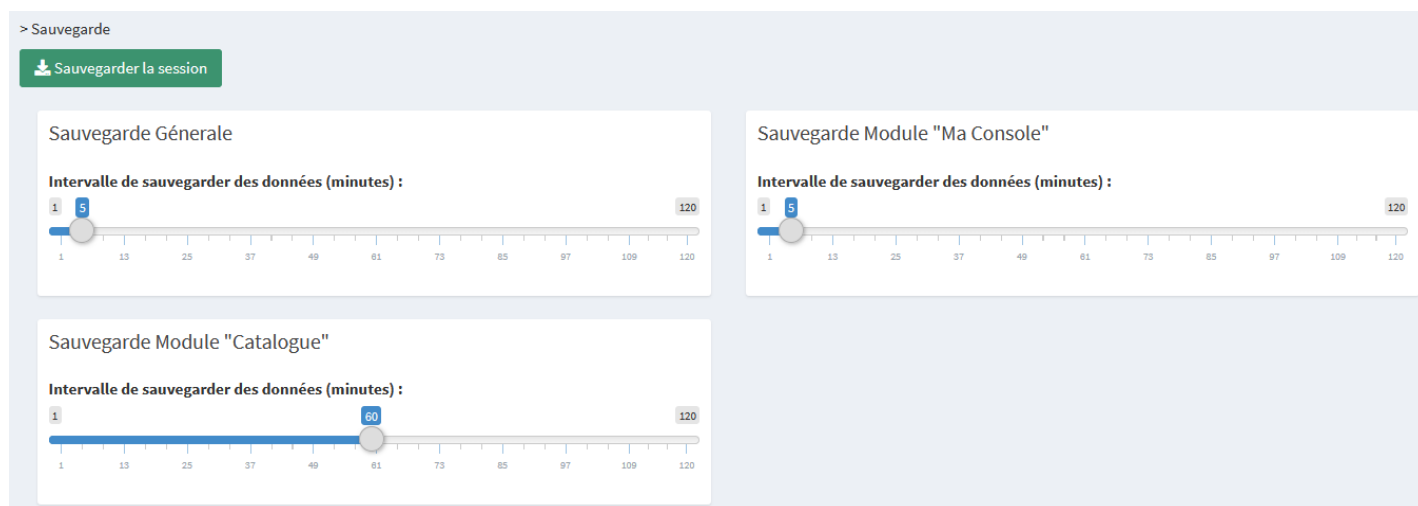


Figure III-22 : capture d'écran de l'onglet "save" de l'application INTEGROMICS

#### v. Console (module administrateur)

C'est le module le plus intéressant de toutes l'application, dans ce module, j'ai pu intégrer un environnement de développement qui permet d'exécuter du code R. Sa particularité est qu'on peut théoriquement accéder à toutes les variables qui permettent de faire fonctionner l'application. Ce qui représente également une très grosse faille de sécurité. Toutefois, il faudrait avoir des connaissances très avancées de l'application pour arriver à exploiter cette faille.

## Les variable à utiliser :

En fonction du résultat de retour de la console, il faudra sauvegarder le résultat dans la variable qui correspond le mieux.

globalInformation\$modules\$Ma\_console\$text : Pour afficher du texte

globalInformation\$modules\$Ma\_console\$dataFrame : Pour afficher un data frame

globalInformation\$modules\$Ma\_console\$plot : Pour afficher un plot

### Ma console : Permet Exécuter du code R

```
1 gi<-list()
2 gi$modules <- list()
3 name_module <- names(globalInformation$modules)
4 for( i in 1:length(name_module))
5 {
6   nmod <- name_module[i]
7   printy(nmod)
8   x<-reactiveValuesToList(globalInformation$modules)
9   gi$modules[[paste0("e",i)]] x
10  printy(names(x))
11 }
12 printy("fin")
13 printy(length(gi$modules))
14 #printy(names(gi$modules$e3))
15
16
17
18
19
```

### Plot : print\_plot(x)

### Text : print\_text(x)

```
length = tracking
length = root@g.c

length = tracking      length = root@g.c
length = tracking      length = root@g.c
azert
access_level  admin  catalogSelected connected
dataFrameLesUtilisateurs
dataFrameRequete_Mes_requetes  email  id
```

### DataFrame : print\_dataframe(x)

Show  entries

Search:

	MB16_3	MB50_1	MB08_1	MB37_1
1000570.HMPREF9966_0917	0	0		
1006551.KOX_14105	0	0		
1006551.KOX_14915	0	0	0	
1006551.KOX_14930	0	0	0	
1028307.EAE_06030	0	0	0	

Showing 1 to 5 of 6,681 entries

Previous  2 3 4 5 ... 1337 Next

Figure III-23 : capture d'écran de l'onglet "console" de l'application INTEGROMICS

## vi. Blast (Basic Local Alignment Search Tool)

Dans ce module, je facilite simplement une opération déjà existante, et ceux en définissant une interface plus simple d'utilisation que les commandes linux. En effet, j'utilise les fonctionnalités du module **blast-2.6.0+**<sup>8</sup> afin de :

- Comparer et recherche des séquences de gène similaire à celle passer en entrant au format texte ou via un fichier fasta
- Visualiser les gènes résultants de la requête au format souhaiter
- Jouer avec les options afin d'obtenir des informations personnalisées

Contrairement, ce module compare des séquences de gènes et retourne celle qui ont le plus de similarité avec celle passe en paramètre

**Exemple :** Résultats obtenue suite à la comparaison d'une séquence de nucléotide.

on veut retrouver des gènes similaires a niveaux de leur séquence avec le gène codifier pour l'enzyme **biotin synthase** Le résultat obtenu montre les gènes du catalogue les plus similaires aux séquence de la requête

<sup>8</sup> **BLAST** (acronyme de **basic local alignment search tool**) est une méthode de recherche [heuristique](#) utilisée en [bio-informatique](#). Il permet de trouver les régions similaires entre deux ou plusieurs [séquences](#) de [nucléotides](#) ou d'[acides aminés](#), et de réaliser un [alignement](#) de ces régions homologues.

Séquence envoi :

>eco:b0775 K01012 biotin synthase [EC:2.8.1.6] | (RefSeq) bioB;  
biotin synthase (N)

atggctcaccgccacgctggacattgtcgcaagtcacagaattatttgaacacggtg  
ctggatctgctgtttgaagcgcagcaggtgcatgccagcatttcgatcctcgtcaggtg  
caggtcagcagcttgctgtcgattaagaccggagctgtccggaagattgcaatactgc  
ccgcaaagctcgcgctacaaaacgggctggaagccgagcgggtgatggaagttgaacag  
gtgctggagtcggcgcgcaaagcgaaagcggcaggtatcgacgcgcttctgtatggcgcg  
gcgtggaagaatccccacgaacgcgatatgccgtacctggaacaaatggtgcagggggtg  
aaagcgtatggggctggaggcgtgtatgacgctgggcagcttgagtgaatctcagggcgag  
cgctcgcgaacgccgggctggattactacaaccacaacctggacacctcgccggagttt  
tacggcaatatcatcaccacacgcacttatcaggaacgcctcgatacgtggaacaaagt  
cgcatgccgggatcaaagtctgttctggcggtcattgtgggcttaggcgaacggtgtaaa  
gatcgcgccggattattgctgcaactggcaacctgccgacgcgcgggaaagcgtgcc  
atcaacatgctggtgaaggtgaaaggcagccgcttgcgataacgatgatgtcgatgcc  
tttgattttattcgaccattcggtcgcgcggatcatgatgcaacctcttacgtgcgc  
ctttctgcggagcgcgagcagatgaacgaacagactcaggcgatgtgctttatggcaggc  
gcaaactcgattttctacggttgcaaactgtgaccacgccgaatccggaagaagataaa  
gacctgcaactgttccgcaaactggggtctaatccgcagcaaactccgtgtgagggg  
gataacgaacaacgaacgtcttgaacaggcgctgatgacccggacaccgacgaatat  
tacaacgcggcagcattatga

## Résultat au format texte, avec visualisation du croisement des séquences

Reference: Zheng Zhang, Scott Schwartz, Lukas Wagner, and Webb Miller (2000), "A greedy algorithm for aligning DNA sequences", J Comput Biol 2000; 7(1-2):203-14.

Database: IGC.fa  
9,879,896 sequences; 7,436,156,055 total letters

Query= eco:b0775 K01012 biotin synthase [EC:2.8.1.6] | (RefSeq) bioB;  
biotin synthase (N)

Length=1041

Sequences producing significant alignments:	Score (Bits)	E Value
MH0014_GL0111028 [gene] locus=scaffold903_8:91042:92082:-[Complete]	1757	0.0
469595.CSAG_00561 protein_id="ZP_04561231.1" GI="237730750" prod...	1070	0.0
MH0277_GL0004481 [gene] locus=scaffold3692_2:2:1012:-[Lack 3'-end]	1027	0.0
T2D-108A_GL0004668 [gene] locus=scaffold107191_1:27473:28513:+[C...	968	0.0
507522.KPK_3773 protein_id="YP_002239592.1" gene="bioB" GI="2065...	959	0.0
MH0260_GL0024994 [gene] locus=scaffold41307_13:6154:7170:+[Compl...	957	0.0
ED13A_GL00080247 [gene] locus=scaffold45954_1:43912:44928:+[Compl...	905	0.0
MH0415_GL0232806 [gene] locus=scaffold99662_2:2:550:-[Lack both ...]	798	0.0
V1.CD3-0-PT_GL0031393 [gene] locus=scaffold1092_11:875:1912:-[Co...	486	2e-134

>MH0014\_GL0111028 [gene] locus=scaffold903\_8:91042:92082:-[Complete]  
Length=1041

Score = 1757 bits (951), Expect = 0.0  
Identities = 1011/1041 (97%), Gaps = 0/1041 (0%)  
Strand=Plus/Plus

Query	1	ATGGCTCACC GCCACGCTGGACATTGTGCAAGTCACAGAATTATTTGAAAAACCGTTG	60
Sbjct	1	ATGGCTCACC GCCACGCTGGACATTGTGCAAGTCACAGAATTATTTGAAAAACCGTTG	60
Query	61	CTGGATCTGCTGTTGAAGCGCAGCAGGTGCATCGCCAGCATTTTCGATCTCGTCAGGTG	120
Sbjct	61	CTGGATCTGCTGTTGAAGCGCAGCAGGTGCATCGTCAGCATTTTCGATCTCGTCAGGTG	120
Query	121	CAGGTCAGCACGTTGCTGTCGATTAAGACCGGAGCTTGTCCGGAAGATTGCAAAATACTGC	180
Sbjct	121	CAGGTCAGCACGTTGCTGTCGATTAAGACCGGAGCTTGTCCGGAAGATTGCAAAATACTGC	180
Query	181	CCGCAAGCTCGCGCTACAAAACGGGCTGGAAGCCGAGCGGTTGATGGAAGTTGAACAG	240
Sbjct	181	CCGCAAGCTCGCGCTACAAAACGGGCTGGAAGCCGAGCGGTTGATGGAAGTTGAACAG	240

## Visualisation sous forme de tableau

outfmt

6

blast!

eco:b0775	MH0014_GL0111028	97.118	1041	30	0	1	1041	1	1041	0.0	1757
eco:b0775	469595_CSAG_00561	85.672	1019	142	4	16	1032	16	1032	0.0	1070
eco:b0775	MH0277_GL0004481	85.119	1008	146	4	26	1031	2	1007	0.0	1027
eco:b0775	T2D-108A_GL0004668	84.545	977	151	0	1	977	1	977	0.0	968
eco:b0775	507522.KPK_3773_83.317	1043	170	4	1	1041	1	1041	0.0	959	
eco:b0775	MH0260_GL0024994	83.710	1019	160	6	26	1041	2	1017	0.0	957
eco:b0775	ED13A_GL00080247_82.953	1009	166	6	26	1031	2	1007	0.0	905	
eco:b0775	MH0415_GL0232806	92.896	549	39	0	271	819	1	549	0.0	798
eco:b0775	V1.CD3-0-PT_GL0031393	76.923	858	196	2	42	898	42	898	2.09e-134	486

## IV- Perspective d'évolution et remarque

Malgré les dix-neuf semaines que j'ai passé sur ce projet, il y a encore beaucoup d'amélioration à faire pour obtenir une application complète et sécurisée.

Cependant, au cours de ces 19 semaines de stage, j'ai pu développer les bases de l'application et établir une documentation qui facilitera les futures développeurs / analystes dans l'amélioration de celle-ci.

### 1. Niveau applicatif

Ils pourront par exemple :

- Implémenter de nouveaux modules de visualisation de données en utilisant les résultats des requêtes déjà effectuées
- Ajouter des modules complètement indépendants du reste du projet afin de centraliser les opérations des chercheurs

Il y a beaucoup de possibilités cela dépendra des besoins des chercheurs.

### 2. Niveau structuration du code

A mon avis il faudra probablement revoir la structuration des données que j'ai utilisées pour effectuer certains traitements, en particulier celle du module de création de requêtes, qui malgré le fait qu'elle tourne comme il faut, pourrais être moins complexe.

### 3. Niveau interface

Au niveau de l'interface, elle pourrait être modifier en utilisant des librairies plus pointues. Pour ma part j'ai commencé par utiliser les interfaces par défauts de **shiny**, puis j'ai ajouté les fonctionnalités de **shinydashboard** et ensuite j'ai découvert de nouvelles librairies que je n'ai malheureusement pas eu le temps d'exploiter, il s'agit de :

- **shinydashboardplus**
- **semantic.dashboard**

L'idéal serait donc d'utiliser les fonctions de l'une de ses librairies et pourquoi pas faire un mixte.

### 4. Niveau base de données

Les possibilités d'optimisation de la vitesse d'obtention des résultats après une requête sur la base de données sont assez nombreuses, j'ai optimisé les requêtes de base (celle où l'on effectue les recherches avec des conditions sur une colonne exemple, la liste des gènes qui ont pour **GeneID** inclus dans un certain intervalle), mais si l'on connaît sur quelle(s) attribut(s) les requêtes sont généralement lancées, il serait possible améliorer la vitesse de retour des données résultantes.

### 5. Niveau sauvegarde des données

Il faudrait trouver un répertoire sécurisé pour la sauvegarde des données et automatiser la sauvegarde de ce répertoire dans un autre afin de se protéger d'une éventuelle panne ou destruction de données.

## 6. Niveau Conception

Avant de continuer ce projet, ou au moins avant de rajouter de nouveau module, il faudrait bien conceptualiser la ou les tâches à effectuer, designer les interfaces, étudier les cas les plus basic et bien déterminer le point d'arriver que ce soit au niveau du développement ou au niveau du résultat à obtenir.

## 7. Niveau fonctionnalité

On pourrait implémenter de nouvelles fonctionnalités qui ne sont pas forcément liées au besoin des chercheurs, mais qui permettrait tout de même d'avoir une meilleure vue sur l'évolution de l'application. Ceci permettra aux administrateurs de mieux gérer les utilisateurs. On pourrait par exemple ajouter une interface de visualisation de statistique d'utilisation de l'application par les utilisateurs (connexion, temps de travail, nouveau utilisateur inscrit ...etc) j'ai commencé à développer une partie de cette fonctionnalité. Celle-ci qui est disponible sur l'application (module Information application) mais qui reste à améliorer.

## V- Bilan du stage

Au final, j'ai passé quatre mois et trois semaines à développer mes compétences, plus important j'en ai acquis de nouvelles. J'ai eu l'occasion de travailler dans un environnement différent de celui de mon école d'ingénieur. Cependant, comme dans tous apprentissages, il y a eu de bon et de mauvais moments. Je parle notamment de ces jours où je passais des heures sur un problème qui au finale n'était pas si compliqué. De toutes ces heures passées à lire des documentations et à consulter des forums, qui en passant ont été d'une aide plus que précieuse.

### 1. Les résultats obtenus sur la productivité et la gestion du temps

Je pense avoir été très productif et je pense avoir su gérer mon temps afin d'atteindre les différents objectifs fixés. Parmi tous les modules que j'ai développés, beaucoup n'ont pas été implémentés dans la version finale de l'application, car ils étaient incomplets ou ne respectaient pas ou plus l'architecture de la nouvelle application ou les besoins initiaux

### 2. Les problèmes ou difficultés rencontrés

- Connaissance insuffisante en Design et ergonomie d'application  
Lors du développement de l'application j'ai été confronté pour la première fois à un problème de design, après chaque réunion, il fallait modifier l'interface, repositionner des boutons, ajouter des zones de texte, supprimer des zones de texte...etc.

En soit-ce n'étaient pas les tâches les plus compliquées, mais cela m'a tout de même fait perdre énormément de temps.

- Développement assez long et toujours évolutif  
Lorsque je développais de nouvelles fonctionnalités et les interfaces qui vont avec, cela inspirait mes maîtres de stage qui me demandaient de rajouter tel ou tel autre fonctionnalité, au final celle initialement développée ne correspondait plus et il fallait plus ou moins tout recommencer.
- Évolution des objectifs initiaux  
Au cours de mon stage j'ai beaucoup plus travaillé sur le côté codage et la structuration de l'application que sur l'analyse des données et malheureusement on n'a pas pu implémenté beaucoup de module d'analyse.
- Module inachevé  
J'ai développé beaucoup de module qui au finale ne serviront pas, car ils sont inachevés.
- Vue d'ensemble du projet du point de vue du développement inexistante  
Vue que c'est un projet ressenti, il n'avait pas forcément une ligne de conduite à suivre au niveau du développement, d'où les nombreuses modifications effectuées.
- Vue d'ensemble du projet du point de vue pratique pas assez clair et inexistante sur papier  
Malgré le fait qu'ils connaissent plus ou moins les fonctionnalités qu'ils auraient aimé avoir dans la version finale il n'y avait pas de document regroupant toutes ces fonctionnalités. Ce qui à mon avis aurait permis d'économiser du temps et pas la même occasion d'être plus productif.

### 3. Les solutions apportées

- Faire de mini conception avant de commencer à développer  
Vu qu'il n'avait pas de conception réel des différentes fonctionnalités à développer, à chaque nouvelle fonctionnalité, je prenais entre une demie journée et une journée pour la conception et le design de l'interface associer.
- Augmentation du nombre de rencontre hebdomadaire  
Au début du stage, on faisait des réunions une fois par semaines et par la suite nous sommes passé à deux réunions par semaine ce qui a grandement amélioré la productivité et l'évolution de l'application.

### 4. Les connaissances et compétences acquises

Ce stage m'a permis d'apprendre :

- Le **langage R**, qui est un langage majoritairement pour faire des études statistiques.  
J'ai principalement utilisé la librairie **shiny** de R qui permet de développer des applications web
- Le langage **NoSQL**, qui est un langage permettant d'exploiter des bases de données qui ne sont pas nécessairement relationnelles. Cependant je n'ai travaillé que sur **MongoDB** qui est un système de gestion de base de données orienté documents. Ce qui m'a permis de manipuler de grands volumes de données
- Gérer un projet avec **GitLab**
- Développé une application configurable et évolutive

### 5. Mes attentes vs la réalité

Je m'attendais à faire beaucoup d'analyse de données, mais au finale je n'ai fait que du développement et quelques analyses. Toutefois, faire beaucoup de développement était indispensable et m'a permis de créer une application donc je suis plutôt fière. Et vue que ce projet sera probablement repris par d'autre stagiaire, ils auront beaucoup plus de facilité pour implémenter les modules.



## Conclusion

À travers mon expérience de stage à l'ICAN, j'ai découvert le fonctionnement d'un institut de recherche, j'ai également pu avoir une idée du travail des chercheurs et de leurs quotidiens. Connue pour lutter contre l'obésité, le diabète, les maladies cardiovasculaires, la NASH et les dyslipidémies, l'ICAN est un des principaux acteurs de la médecine de demain.

L'application que j'ai développée a pour but d'aider les chercheurs dans leurs analyses et interprétation des données. Le développement de cette application m'a non seulement permis de mettre en pratique les concepts et connaissances étudiés au cours de ma formation en particulier ceux acquis lors des cours d'administration système, de base de données, de conception et de programmation orienté objet, mais aussi de développer mon autonomie, ma faculté d'organisation, de structuration et de recherche d'information.

L'aspect que j'ai le plus apprécié au cours de ce stage a été mon autonomie, même si travailler en équipe aurait amélioré l'efficacité de l'application je pense que l'avoir fait seul m'a permis de mieux comprendre et apprécier les petits avantages et inconvénients d'un travail de groupe.

Ce stage jouera forcément un rôle très important dans la suite de mon parcours d'ingénieur. Et même si je ne compte pas travailler plus tard dans la recherche j'aurais au moins rencontré des personnes exceptionnelles.

## Table des matières

Remerciement.....	3
Résumer .....	4
1. En Français .....	4
2. En Anglais .....	4
Sommaire .....	5
Introduction .....	7
I- L'Institut de Cardiométabolisme et Nutrition (ICAN) .....	8
1. Présentation.....	8
2. Mission : .....	8
II- Présentation du projet et des missions .....	9
1. Le projet .....	9
a. Le context.....	9
b. Objectif principale.....	9
c. Les données .....	9
d. Les compétences requises .....	12
2. Les missions.....	12
a. Missions initial.....	12
b. Evolution des missions .....	12
III- Déroulement du stage .....	13
1. Les outils à disposition .....	13
a. Les logiciels.....	13
b. Le matériel .....	13
2. État du projet au 23 avril 2019.....	13
3. Réalisations des missions .....	16
a. La base de données.....	16
i. Optimisation des requêtes.....	16
ii. Fonctions d'administration de la base de données .....	16
iii. La nouvelle architecture de la base de données.....	17
b. Structure de l'application.....	17
c. Gestion des sessions et sauvegarde des données .....	18
d. Les principaux modules.....	19
i. Connexion .....	19

ii.	Accueil.....	20
iii.	Information application (module administrateur).....	20
iv.	Gestion des utilisateurs (module administrateur).....	21
v.	Gestion des projets (module administrateur) .....	22
vi.	Collections.....	22
vii.	Déconnexion .....	22
e.	Les modules fonctionnels .....	23
i.	Générateur de requêtes.....	23
ii.	Consultation des requêtes .....	23
iii.	Analyse et visualisation des données projets .....	24
	• Comparer des projets.....	24
	• Visualise des plots spécifiques aux analyses médical .....	24
iv.	Gestion des sauvegardes .....	26
v.	Console (module administrateur) .....	26
vi.	Blast (Basic Local Alignment Search Tool).....	27
IV-	Perspective d'évolution et remarque .....	29
1.	Niveau applicatif .....	29
2.	Niveau structuration du code .....	29
3.	Niveau interface.....	29
4.	Niveau base de données .....	29
5.	Niveau sauvegarde des données .....	29
6.	Niveau Conception.....	30
7.	Niveau fonctionnalité.....	30
V-	Bilan du stage .....	31
1.	Les résultats obtenus sur la productivité et la gestion du temps .....	31
2.	Les problèmes ou difficultés rencontrés.....	31
3.	Les solutions apportées .....	32
4.	Les connaissances et compétences acquises.....	32
5.	Mes attentes vs la réalité.....	32
	Conclusion.....	33
	Table des matières.....	34
	Annexe .....	36
	Webographie .....	36
	Documents .....	36

## Annexe

### Webographie

- Documentation mongolite

<https://github.com/jeroen/mongolite>

<https://blog.exploratory.io/an-introduction-to-mongodb-query-for-beginners-bd463319aa4c>

- Débuter avec shiny

<https://shiny.rstudio.com/tutorial/>

- Débuter avec shinydashboard

<https://rstudio.github.io/shinydashboard/structure.html#boxes>

<https://www.r-pkg.org/pkg/shinyjs>

- ShinyAce

<https://github.com/trestletech/shinyAce>

<https://cran.r-project.org/web/packages/shinyAce/readme/README.html>

[https://rdr.io/cran/dqshiny/man/autocomplete\\_input.html](https://rdr.io/cran/dqshiny/man/autocomplete_input.html)

- Documentation datatable

<https://github.com/rstudio/DT/pull/480>

<https://github.com/rstudio/DT/releases>

- Démo blast

<https://2-bitbio.com/2017/06/running-blast-in-shiny-web-application.html>

<https://github.com/larsgr/RLinuxModules>

### Documents

Consulter la documentation du projet (elle est fournie avec le rapport)