

Faculté des Sciences Aix Marseille
Université Montpellier
6, Avenue du Pignonnet
13090 Aix en Provence

Université Aix-Marseille - Site de Luminy
Inserm TAGC UMR_S1090
Parc Scientifique de Luminy, Case 928
13288 Marseille cedex 9, France

Achèvement de l'application mimicINTWeb Rapport de stage 08/06/2023

Remerciements

Tout d'abord, je tiens à remercier mes maîtres de stage, Andreas Zanzoni et Lionel Spinelli de m'avoir guidé durant mon stage.

J'aimerais également remercier Mégane Boujeant de m'avoir partagé son savoir et de m'avoir beaucoup aidé à la réalisation de mes missions et à résoudre certains problèmes rencontrés.

De plus, je souhaiterais remercier tous les membres de l'équipe biologie des réseaux qui ont été derrière moi pour mes missions et la rédaction de mon rapport.

Enfin, je voudrais remercier l'ensemble des professeurs de la faculté d'informatique d'Aix Montperrin pour m'avoir guidé durant cette année de licence.

Fiche Technique

Étudiant : Kevin Maldonado

Année : 2023

Raison sociale de l'entreprise : Institut national de la Santé et de la Recherche Médicale

Maîtres de stage : Andreas Zanzoni, enseignant chercheur; Lionel Spinelli, ingénieur de recherche.

Tuteur : Severine Fratani

Mission : Finalisation de l'interface d'une application web pour la prédiction et l'analyse des interactions moléculaires entre microbes et cellules humaines. Les missions consistent à faire communiquer un outil bioinformatique avec une interface web.

Plateforme informatique et système :

Ubuntu, PostGreSQL, Docker

Outils et langages :

Python, Shell, framework Django, Bootstrap, HTML, CSS, JavaScript, Ajax,

Web, Python, Bio-informatique

Sommaire

Remerciements	2
Fiche Technique	3
Sommaire	4
Introduction	5
Présentation de l'entreprise	6
l'INSERM : son histoire, son rôle et ses caractéristiques	7
L'unité de recherche TAGC	8
L'équipe Biologie des réseaux	8
Travail réalisé	10
Paragraphe introductif / mission	11
Matériels et outils logiciels	11
Django : de Python à un site web	11
Le workflow mimicINT	12
SLURM & Snakemake : Deux outils essentiels à l'exécution du workflow	13
Docker : La conteneurisation de logiciels	13
Sourcesup & Smartgit : Deux outils de versionnage de projet collaboratif	14
Grandes étapes du travail	14
Réadaptation et reprise du projet	14
Barre de progression & déroulement du workflow	16
Debugging & testing de la dernière version de mimicINT	18
Déploiement du site & hébergement	19
Conclusion et perspective	21
Bilan du stage	23
Difficulté rencontré	24
Conclusion	25
Liste des abréviations, acronymes et sigles	26
Bibliographie	27
Table des illustrations	29

Introduction

De nombreux micro-organismes sont vecteurs de maladie pour l'Homme, causant des millions de morts à travers le monde. Afin de développer des traitements plus efficaces, il est nécessaire de comprendre et d'identifier les mécanismes moléculaires des interactions entre les microbes, tels que les bactéries et les virus, et leurs hôtes.

Les approches expérimentales peuvent être très coûteuses en ressources humaines et matérielles ainsi qu'en temps.

C'est pour cela que l'utilisation d'outils informatiques, notamment bioinformatiques, peut être un très bon moyen de prédire et modéliser ces interactions.

C'est dans ce contexte et dans une unité de recherche rattachée à l'*Institut National de la Santé et de la Recherche Médicale* (INSERM), qu'une approche bioinformatique a été développée pour inférer les interactions des protéines des microbes avec les protéines humaines.

La mission de mon stage, réalisé dans le laboratoire *Theories and Approaches of Genomic Complexity* (TAGC) de l'INSERM, consistait à finaliser une interface web permettant de faciliter l'utilisation d'un outil bioinformatique d'inférence d'interactions développé par l'équipe "*Biologie des réseaux*".

Many microorganisms are vectors of human diseases, causing millions of deaths worldwide. In order to develop more effective treatments, it's necessary to understand and identify the molecular mechanisms of the interactions between microbes, such as bacteria and viruses, and their hosts.

Experimental approaches can be very costly in terms of human and material resources, and time.

This is why the use of computational tools, especially bioinformatic tools, can be a very good way to predict and model these interactions.

It's in this context and in a research unit attached to the *Institut National de la Santé et de la Recherche Médicale* (INSERM), that a bioinformatic approach has been developed to infer the interactions between microbial and human proteins.

The mission of my internship, carried out in the laboratory *Theories and Approaches of Genomic Complexity* (TAGC) of the INSERM, was to finalize a web interface to facilitate the use of a bioinformatic tool for inferring protein interactions developed by the "*Biologie des réseaux*" team.

Présentation de l'entreprise

l'INSERM : son histoire, son rôle et ses caractéristiques

L'Institut national de la santé et de la recherche médicale (INSERM), fondé par Raymond Marcellin en 1964, est un établissement public consacré à la recherche biologique, médicale et à la santé humaine. L'INSERM a pour objectif d'améliorer la santé de tous en accumulant des connaissances sur le vivant ainsi que les maladies. Il a été l'acteur d'avancées médicales majeures depuis sa création et continue de jouer un rôle important, notamment sur la recherche des maladies infectieuses comme le COVID-19 par exemple.

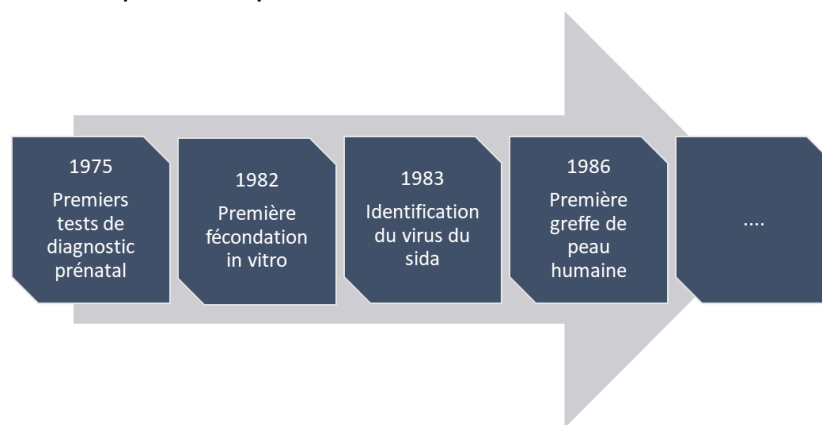


Figure 1 : Frise chronologique de quelques dates clés où l'INSERM a été un acteur majeur

Cet institut est constitué de nombreuses unités de recherche, avec 281 unités en 2017 réparties dans toute la France et compte près de 15 000 personnes travaillant au sein de ces structures cette même année. Chaque unité, souvent implantée dans des hôpitaux ou des campus universitaires, se consacre à la recherche d'une thématique liée à la santé ou à la biologie, permettant alors un plus grand champ d'exploration dans ces domaines vastes et complexes. À noter que l'INSERM travaille souvent sur des projets en partenariat avec d'autres infrastructures (hôpitaux, autres instituts de recherche étrangers).

Étant donné son grand nombre d'unités, mais aussi sa vision globale qui est animée par l'apprentissage et l'accumulation des connaissances, cette organisation met en place plusieurs moyens pour permettre aux scientifiques de toujours s'informer sur les nouvelles avancées. Outre les collaborations possibles entre différentes unités et les publications d'autres chercheurs, l'INSERM met en place un site web ainsi qu'un magazine mensuel pour toujours informer son effectif sur les travaux des autres employés.

L'unité de recherche TAGC

J'ai pu travailler dans l'une de ces unités, l'unité TAGC (*Theories and Approaches of Genomic Complexity*) dont l'activité suit toujours le principe de la recherche en biologie. C'est un laboratoire, localisé dans le campus universitaire de Luminy, dont la thématique de recherche est centrée sur la biologie des systèmes et les maladies multifactorielles.

Les projets sont organisés autour de deux axes dans ce laboratoire :

- Le premier axe, nommé "Bioinformatique et génomique des réseaux moléculaires", vise à comprendre le réseau d'interactions moléculaires au sein d'une même cellule ou entre cellules via des outils bioinformatiques.
- Le deuxième axe, qui porte le nom "génétique et génomique des maladies multifactorielles", cherche à identifier le rôle et les mécanismes des gènes exprimés dans les cellules, conduisant à l'apparition de certaines maladies multifactorielles (malaria, sepsis ...).

Pour tenir informés les employés de l'avancement ou des découvertes des autres personnes travaillant au sein de l'unité, tous les vendredis se tient un séminaire scientifique au cours duquel un chercheur, un post-doctorant ou un doctorant présente son sujet d'étude.

Le laboratoire est composé de plusieurs équipes travaillant sur plusieurs projets (souvent en collaboration avec d'autres laboratoires) sur les deux axes de recherche évoqués précédemment.

L'équipe Biologie des réseaux

J'ai intégré l'équipe Biologie des réseaux, qui est rattachée au premier axe de recherche du TAGC. L'objectif de cette équipe est de mieux comprendre les réseaux d'interactions moléculaires cellulaires, notamment les réseaux d'interactions protéine - protéine.

En effet, on ne connaît pas encore toutes les interactions possibles entre chaque protéine présente dans les cellules, car elles peuvent ne survenir que dans certaines conditions (type cellulaire, conditions de stress etc.), qui peuvent être totalement inconnues.

Tous les membres de l'équipe ne travaillent pas forcément sur le même projet, c'est pour cela qu'il y a tous les vendredis, une réunion avec l'ensemble des membres de l'équipe.

En premier lieu un tour de table est organisé, où chacun résume ce qu'il a fait dans la semaine.

Ensuite il suit la présentation d'un projet par une personne, afin qu'elle puisse recueillir les avis et suggestions sur l'avancée scientifique du projet, mais aussi s'entraîner et s'améliorer dans sa présentation en vue d'un prochain séminaire ou d'une conférence scientifique.

Par ailleurs, l'un des projets sur lequel travaille l'équipe et auquel j'ai participé, est un projet nommé *mimicINT*. Ce projet consiste à prédire le réseau d'interactions entre les protéines d'un microbe (bactérie ou virus) et les protéines d'une cellule hôte grâce à un outil bioinformatique.

Travail réalisé

Paragraphe introductif / mission

Au sein de l'équipe Biologie des réseaux, un outil bioinformatique permettant de prédire le réseau d'interactions entre les protéines d'un microbe pathogène et les protéines de l'hôte a été développé.

Cet outil, nommé *mimicINT*, consiste en un enchaînement de tâches d'analyse, d'interprétation et de modélisation d'un réseau d'interactions entre protéines (appelé *workflow*).

Néanmoins, l'utilisation de *mimicINT* est assez difficile à prendre en main. Au sein de la communauté scientifique, notamment chez les biologistes, peu de personnes sont à l'aise en informatique. Ces personnes ne possèdent pas les aptitudes nécessaires pour pouvoir manipuler *mimicINT*, étant donné que son utilisation se faisait exclusivement en ligne de commande. Il fallait donc trouver un moyen de rendre plus accessible cet outil.

La solution adoptée pour répondre à cette problématique a été la réalisation d'un projet de site web nommé *mimicINTWeb*.

Au début, j'ai eu la chance aussi de reprendre le projet *mimicINTWeb* dans l'état où je l'avais laissé lors de mon dernier stage, ce qui m'a permis de pouvoir facilement le reprendre en main.

Néanmoins, une chose a changé lors de ma reprise : le workflow *mimicINT* a continué d'être développé entre temps. Par conséquent, cela a eu un impact sur le côté web.

Mes tâches ont donc consisté à :

- Résoudre tous les problèmes de compatibilité liés à la dernière version de *mimicINT*.
- Finaliser l'aspect de certaines pages
- Permettre à l'utilisateur de savoir en temps réel où en est le workflow
- Déployer le site web pour le tester en condition d'hébergement

Il est important de noter que les missions présentées ci-dessus sont organisées dans un ordre de priorité et de logique, mais pour des raisons de clarté, elles seront présentées dans le rapport selon leur ordre chronologique d'exécution.

Matériels et outils logiciels

Django : de Python à un site web

Django est un framework Python, c'est-à-dire une bibliothèque d'outils informatiques permettant de faciliter le développement de sites web. Il permet entre autres de faciliter la séparation de l'interface web et des données pour un affichage dynamique.

Django fonctionne selon le modèle MVT (Modèle, Vue, Template, une variante MVC (Modèle, Vue, Contrôleur). Ces modèles sont des patrons de conception utilisés pour faciliter la lisibilité et la correction des différents fichiers. Cela nous simplifie donc la programmation en permettant d'éviter l'utilisation de certaines balises HTML ou requête SQL via l'utilisation d'un langage orienté objet en Python.

Une base de données a aussi été utilisée pour le site web. *PostgreSQL* a été choisie comme système de gestion de base de données. Une interface graphique, nommée *Adminer*, a été déployée pour faciliter la manipulation de cette base de données.

Le workflow *mimicINT*

Un workflow est un enchaînement de processus d'analyse de telle sorte que le paramètre de sortie d'un processus sert de paramètre d'entrée au prochain.

Il permet d'effectuer un ensemble de tâches d'analyse, aussi appelé règles, permettant de traiter un paramètre d'entrée (ici une liste de protéines) et quelques autres variables soumises par l'utilisateur.



Figure 2 : Schéma des différentes tâches d'analyse du workflow *mimicINT*

SLURM & Snakemake : Deux outils essentiels à l'exécution du workflow

Le site web se base sur un workflow pour analyser en arrière-plan les séquences de protéines soumises par l'utilisateur.

Pour la gestion des tâches du workflow, Snakemake a été utilisé. Il permet de gérer et formaliser un enchaînement de processus. Il assure donc le bon déroulement du workflow, en assurant l'ordre d'exécution des processus et la traçabilité des erreurs survenues.

SLURM, quant à lui, est un gestionnaire de ressources. Il décide, pour chacun des processus qui lui est soumis, quand l'exécuter en fonction des ressources physiques disponibles (CPU, RAM, ...) et de celles requises par l'exécution du processus.

SLURM s'allie très bien avec Snakemake car de nombreuses instances du workflow peuvent être en cours ou soumises simultanément, ce qui rend la gestion de ces ressources indispensable dans le cadre de l'application web.

Docker : La conteneurisation de logiciels

Docker est un logiciel permettant d'emballer une application avec toutes ses dépendances dans un conteneur. Il est utilisé dans le but d'isoler chacune des applications vu précédemment dans des conteneurs afin d'éviter des conflits entre elles, mais aussi pour faciliter le déploiement de ces applications. On peut voir Docker comme une alternative aux machines virtuelles bien que l'utilisation des ressources du serveur soit très différente.

Dans le cas du projet *mimicINTweb*, il existe 4 conteneurs :

- Le conteneur serveur web, qui possède les dépendances nécessaires pour utiliser Django.
- Le conteneur PostgreSQL, qui s'occupe de la base de données
- Le conteneur SLURM & Snakemake, qui permet d'utiliser le gestionnaire de jobs SLURM et d'exécuter le workflow avec Snakemake.
- Un conteneur Adminer, qui facilite la gestion de la base de données avec une interface graphique (non essentiel à l'interface web de *mimicINT*)

Grâce à un fichier `docker-compose.yaml`, on peut définir des interdépendances entre chaque conteneurs. En effet, il est possible de faire communiquer les conteneurs entre eux , permettant ainsi l'échange de données.

Néanmoins le conteneur SLURM & Snakemake ne pouvaient, à l'origine, pas communiquer avec les autres conteneurs en raison de soucis de sécurité. En effet, ce conteneur avait un accès privilégié aux ressources de la machine.

Pour permettre tout de même l'échange de données, on inscrit toutes les données qui doivent transiter dans des fichiers stockés sur l'ordinateur pour que les autres conteneurs puissent les interpréter.

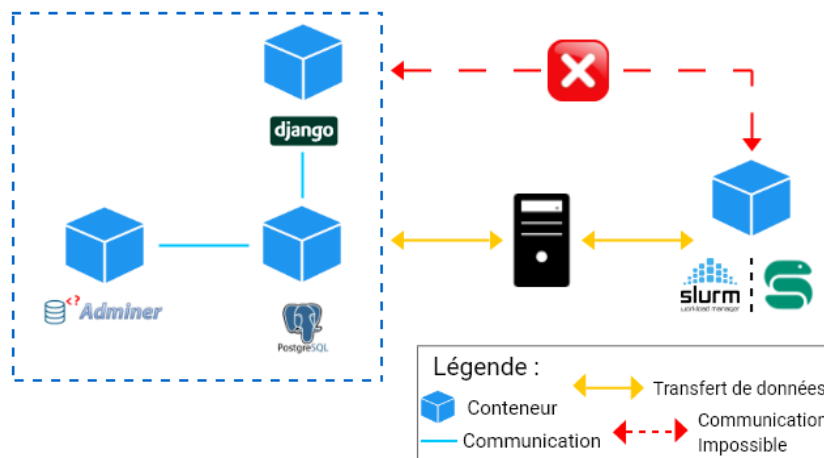


Figure 3 : Schéma des communications entre les conteneurs Docker

Sourcesup & Smartgit : Deux outils de versionnage de projet collaboratif

Sourcesup est une forge logicielle, c'est-à-dire un outil informatique permettant à plusieurs développeurs de participer ensemble au développement d'un projet commun. Cette forge est très similaire à GitHub dans son fonctionnement. Néanmoins on l'utilise car elle est destinée aux établissements d'enseignement supérieur et de la recherche française.

Smartgit est une interface logicielle permettant de faciliter la gestion du versionnage de code via Git. Il permet d'éviter l'utilisation de lignes de commande Git, parfois complexes, mais aussi d'éviter de potentielles erreurs de manipulation.

Grandes étapes du travail

Réadaptation et reprise du projet

Les premières semaines ont été des semaines dédiées à la réadaptation où j'ai dû reprendre mes marques sur les différents outils utilisés.

Pour cela, mes premières missions étaient surtout liées à l'aspect de certaines pages qui n'étaient pas responsives ou dans lesquelles il manquait tout simplement des éléments.

Un exemple est la page de contact. Un des problèmes de cette page était qu'il manquait des champs d'écriture nécessaires pour le formulaire d'envoi d'email.

En temps normal, le framework Django peut s'en occuper sans l'utilisation de balise HTML, mais ici, pour afficher cette page contact, il a été décidé d'afficher une fenêtre *contact us* disponible depuis la barre de tâche, ce qui posait un problème pour Django. Étant donnée que cette fenêtre devrait être disponible depuis toutes les pages comportant la barre de tâche, la solution adoptée a tout simplement été de mettre de côté les méthodes de Django et donc d'utiliser des balises HTML directement (`<form></form>`).

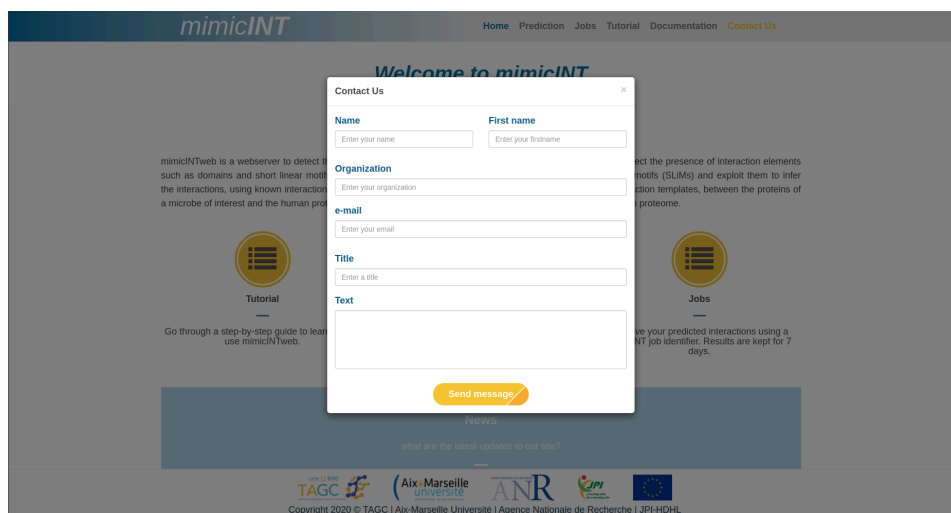


Figure 4 : fenêtre de contact de *mimiciNTWeb*

Un autre exemple est celui de la page permettant de faire une requête pour savoir à quelle étape se trouve le processus d'analyse des données biologiques (Figure 5).

Juste après avoir validé le formulaire de demande d'analyse, un identifiant appelé *run_id* est délivré à l'utilisateur. Ce *run_id* est nécessaire pour retrouver les résultats liés aux données fournies grâce à la page jobs.

Néanmoins, lorsqu'un id inexistant était entré, aucune action n'était effectuée ce qui pouvait conduire l'utilisateur dans l'incompréhension. C'est pourquoi qu'une de mes tâches a été d'informer l'utilisateur lorsqu'il rentre un mauvais *run_id*.

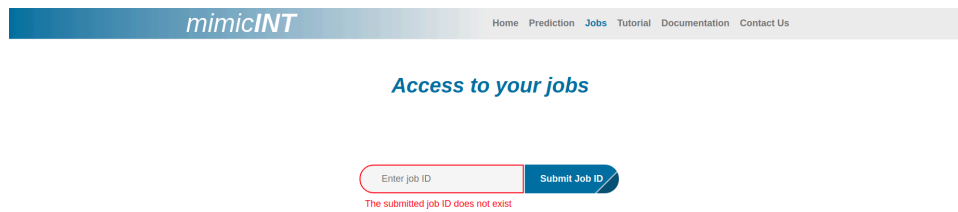


Figure 5 : Page jobs de *mimicINTWeb*

Barre de progression & déroulement du workflow

Dans un second temps, une de mes tâches a été de créer une page permettant de suivre en temps réel la progression du workflow. Lorsqu'un utilisateur entre un identifiant existant depuis la page décrite précédemment, il est censé pouvoir retrouver des informations sur les données qu'il a fournies depuis la page *prédiction*. La page de *résultats des données* et la page d'*erreur* (qui survient si l'exécution du workflow a subi une erreur) ont déjà été créées auparavant, il fallait s'occuper du cas où *mimicINT* était en cours.

Pour cela, il fallait que je respecte 2 conditions pour cette page :

- 1) Avoir un aperçu sur les étapes finalisées du workflow
- 2) Faire en sorte que la barre de progression évolue selon une durée définie en temps réel

Pour la 1er condition, j'ai dû reprendre un script que j'avais précédemment créé. Celui-ci permettait entre autres de pouvoir déterminer dans quel état était le dossier run (ensemble des fichiers liés à l'exécution du workflow). Si il était en état en cours, ce script cherchait dans le dossier run le statut de chaque étape (finie ou non) pour ensuite me retourner dans un fichier `rule_index`, à quel étape l'analyse en était.

Il fallait donc récupérer cette valeur et l'utiliser comme indicateur pour le site web. Cependant, il était nécessaire au préalable de récupérer le nombre d'étapes totales du workflow. C'est pourquoi, plutôt que d'indiquer manuellement ce nombre, j'ai préféré utilisé la base de données du site , plus précisément la table `mimicINTapp_pipeline_rule`.

<input type="checkbox"/> Modification	rule_id	rule_order	rule_description	rule_duration	log_path
<input type="checkbox"/> modifier	parse_3did	1	Parse the flat file from 3did to gather domain-domain interaction templates	60	
<input type="checkbox"/> modifier	ddl_template_pfam_to_interpro	2	Convert the Pfam accessions in the file parsed from 3did into InterPro accessions	45	
<input type="checkbox"/> modifier	parse_elm	3	Filter ELM classes based on their probability of occurrence	30	
<input type="checkbox"/> modifier	elm_domain_interactions_to_interpro	4	Convert the Pfam accessions in the ELM - domain interaction file into InterPro accessions	30	
<input type="checkbox"/> modifier	detect_domain_query	5	Use InterProScan to detect the domains in the query sequences	330	
<input type="checkbox"/> modifier	parse_domain_query	6	Parse the InterProScan output run on the query sequences to get useful information	30	
<input type="checkbox"/> modifier	split_query_dataset	7	Split large query fasta file into several fasta files	15	
<input type="checkbox"/> modifier	detect_slim_query	8	Use SLIMProb (SLIM suite) to identify the SLIM in the query sequences, and optionally perform a conse...	150	
<input type="checkbox"/> modifier	aggregate_detect_slim_query_output	9	Aggregate the outputs of the detect_slim_query rule into one single file	45	
<input type="checkbox"/> modifier	match_query_sqce_names	10	Compute the correspondences between the sequence names used by SLIMProb and the name of the sequence...	15	
<input type="checkbox"/> modifier	parse_slim_query	11	Parse the output of SLIM Prob in order to get useful information	30	
<input type="checkbox"/> modifier	compute_query_disorder_propensity	12	Compute the disordered propensity for query sequences	45	
<input type="checkbox"/> modifier	interaction_inference	13	Inference the domain(target) - domain(query) and domain(target) - SLIM(query) interactions by combin...	90	
<input type="checkbox"/> modifier	generate_json_interaction_inference	14	Generate a json file that may be used by Cytoscape	210	
<input type="checkbox"/> modifier	generate_json_query_features	15	Generate a json file that may be used to display the sequences on the web interface	60	
<input type="checkbox"/> modifier	simplify_sequence_names	16	Rename the sequences in all the output (optional rule)	300	
<input type="checkbox"/> modifier	target_enrichment_gprofiler	17	Perform a gProfiler analysis on the target interactors	600	

Figure 6 : Table des étapes du workflow

Ma méthode pour obtenir ce nombre total a été d'additionner le nombre de règles (étapes) existantes dans cette table. Grâce à ça, en cas de mise à jour du workflow, il suffira simplement de changer cette table dans la base de données.

A ce stade, il fallait que je trouve un moyen de communiquer à l'utilisateur combien de temps il restait avant de pouvoir obtenir ces résultats. Pour cela j'ai opté pour une barre de progression, qui permet de transmettre visuellement une estimation du temps restant.

Afin d'obtenir ce type de page, il était nécessaires de recueillir plusieurs données :

- La date de commencement du workflow
- La date supposés finales du workflow
- La date actuelle

Pour la 1ere date, il m'a fallu stocker dans une autre table de la base de données (mimicINTapp_job_infos), la date de soumission du workflow. Cette date est obtenue depuis une autre page du site déjà finalisée, la page formulaire (prediction) où l'utilisateur rentre un certain nombre de données pour le workflow *mimicINT*.

Pour la 2eme date, j'ai dû additionner l'ensemble des durées de chaque règles pour ensuite l'ajouter à la date de soumission, me donnant donc une date supposée finale. A noter que cette date ne signifie pas forcément que le workflow aura fini à ce moment. Cette date correspond à une date limite qui, si elle est dépassée, fera que le processus sera considéré comme trop long et donc en échec.

Enfin, grâce à la date actuelle, on peut faire évoluer la barre de progression.

Workflow in progress

01/7

[go to home page](#)

Copyright 2020 © TAGC | Aix-Marseille Université | Agence Nationale de Recherche | JPI-HDHL

Figure 7 : Page de progression de *mimicINTWeb*

Debugging & testing de la dernière version de *mimicINT*

Comme expliqué auparavant, le site *mimicINTWeb* se base sur un workflow nommé *mimicINT*. Lors de mon précédent stage, je travaillais sur une version de ce workflow maintenant obsolète et inutilisable pour le site.

Il fallait donc que je déploie la nouvelle version disponible (<https://github.com/TAGC-NetworkBiology/mimicINT>). Or à cause de certains besoins que nécessite le site, mais aussi à cause de certaines dépendances qu'utilise le workflow, de nombreux problèmes ont eu lieu lors de son installation et de son exécution.

Ma mission a donc été de répertorier les différentes erreurs rencontrées, et apporter des corrections afin de pouvoir utiliser l'outil pour *mimicINTWeb*.

Tout comme la version web, le workflow *mimicINT* utilise aussi Docker et son système de conteneurisation. Cela permet, entre autres, de stocker dans plusieurs conteneurs les diverses dépendances nécessaires à l'exécution de l'outil.

Or un des problèmes de la version délivrée par le github était qu'elle ne prenait pas en compte les dernières versions de certaines dépendances, notamment celle concernant une dépendance nommée Singularity. Tout comme Docker, Singularity est un outil de conteneurisation d'application, à la différence qu'il est surtout utilisé pour les environnements nécessitant des calculs de hautes performances. A noter que cet outil comme Docker fonctionne avec des fichiers *images*, c'est-à-dire des

fichiers contenant toutes les dépendances qu'on veut installer dans notre conteneur. Il est tout à fait possible de convertir des *images* Docker en *images* Singularity, néanmoins les solutions permettant cette conversion sont très différentes entre chaque version de Singularity.

De ce fait, ma tâche était de faire en sorte que le workflow puisse supporter cette dernière version.

De plus, il fallait que je fasse en sorte que son exécution soit adaptée au besoin du site web. Comme précisé auparavant, le site web lui aussi utilise des conteneurs pour son fonctionnement et ses dépendances.

Ici je vais m'attarder sur le conteneur Snakemake & SLURM.

Ces deux dépendances font partie d'un seul et même conteneurs, Snakemake permet d'exécuter le workflow alors que SLURM permet de gérer les ressources allouées pour le workflow.

La version initiale de *mimicINT* essayait de proposer une version incluant un fichier de configuration (appelé `configuration cluster`) permettant d'interconnecter plusieurs nœuds de calculs, créant alors une infrastructure de calcul distribué. Ce fichier de configuration cluster est nécessaire pour SLURM, car il s'appuie sur celui-ci gérer les ressources disponibles dans cette infrastructure.

Néanmoins, il fallait adapter ce fichier aux ressources disponibles sur le serveur/machine sur lequel était hébergé le projet *mimicINTWeb*.

Et donc, en parallèle avec toutes ses modifications, j'ai effectué une phase de testing pour vérifier que mes modifications permettent de rendre le projet fonctionnel, mais aussi de faire en sorte qu'elles ne modifient pas le déroulement et les fichiers de sorties supposés de *mimicINT*.

Déploiement du site & hébergement

Enfin, il fallait faire en sorte d'héberger le site web sur un serveur robuste et sécurisé. Pour cela, la décision a été de déployer le projet sur un serveur dédié de l'IFB (Institut Français de Bioinformatique).

Pour héberger et déployer des projets sur les serveurs de l'IFB, l'institut propose l'utilisation de machine virtuelle (VM). Une VM est un environnement informatique

émulant un système d'exploitation et ses applications comme s'il s'agissait d'un ordinateur physique.

L'institut propose un choix vaste de machines virtuelles avec diverses dépendances installées. Il est également possible de choisir les ressources allouées pour la session.

Dans notre cas, nous avons décidé de tester le projet sur une machine virtuelle avec le système d'exploitation Ubuntu 18.04, Docker et possédant 8 coeurs, 16 Go de ram ainsi que 200 Go de stockage. Ces caractéristiques répondent très bien aux ressources nécessaires pour faire fonctionner le projet *mimicINTWeb*.

Pour accéder à l'hébergeur, il était nécessaire d'utiliser une connexion SSH pour garantir la confidentialité des échanges. Cette connexion est établie grâce à une paire de clé publique et privée utilisant le chiffrement RSA. Ce chiffrement est un algorithme de cryptographie asymétrique générant une clé publique permettant de chiffrer les données, et une clé privée pouvant déchiffrer les données.

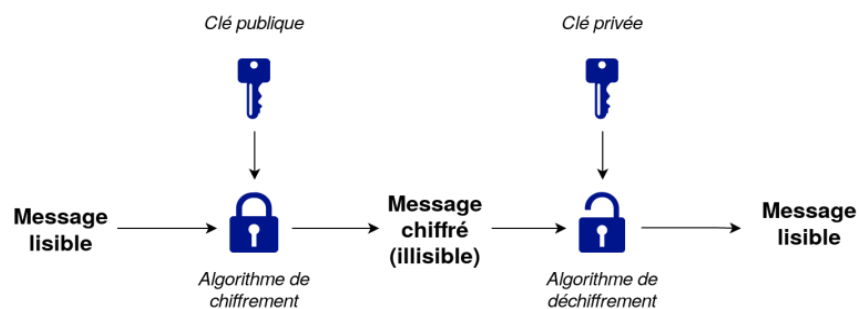


Figure 8 : Schéma du chiffement asymétrique

Dans notre cas, afin d'accéder à l'hébergeur, il était nécessaire de fournir une clé publique associée à un compte de l'IFB. Cette clé publique a été générée localement sur l'ordinateur en même temps que la clé privée. Et donc en utilisant toujours ce principe de chiffement asymétrique, il est possible d'établir une connexion SSH avec le serveur et notre machine local de manière sécurisée.

Néanmoins, il était aussi nécessaire de prendre en compte la confidentialité des données transmises entre le site web et ses utilisateurs, par conséquent il était nécessaire de s'informer sur une manière de pouvoir sécuriser les communications en utilisant le protocole HTTPS.

Une manière possible permettant d'implémenter ce protocole est le système d'autocertification. En effet, il permet de générer un certificat électronique SSL

(Secure Sockets Layers) qui authentifie l'identité d'un site Web et permet alors une connexion chiffrée (et ainsi permet l'utilisation du protocole HTTPS).

A noter que cette solution est utilisée de manière temporaire pour la phase beta du projet, car elle témoigne d'une connexion sécurisée entre l'utilisateur et le site web mais ne garantit en aucun cas la fiabilité du site.

Conclusion et perspective

Grâce aux missions effectuées j'ai pu mettre en place une version fonctionnelle de l'application *mimicINTWeb*.

Maintenant, il est désormais possible d'utiliser *mimicINTWeb* en local avec la dernière version de *mimicINT*, grâce à la page de formulaire qui envoie bien les informations nécessaires au workflow et grâce à la page de requête, qui affiche soit le résultat du workflow, soit l'erreur survenue ou alors en temps réel à quel stade le workflow en est.

En perspective, il faudrait mettre en place toute la partie hébergement et déploiement en œuvre. Pour l'instant, j'ai pu faire qu'un travail de recherche concernant le déploiement du site, mais je compte d'ici la fin de mon stage pouvoir déployer l'application web sur les serveurs de l'IFB.

Bilan et perspectives

Bilan du stage

Ce stage m'a permis d'approfondir mon expérience professionnelle dans le domaine de l'informatique. Ce fut pour moi ma seconde expérience dans ce domaine.

J'ai eu la chance de pouvoir travailler dans un bon cadre que je connaissais déjà, ce qui m'a évité un certain temps d'adaptation comme la première fois. Pouvoir effectuer ce stage dans la même équipe et sur le même sujet que l'année dernière fut d'une très grande aide pour moi.

Concernant les missions que j'ai pu effectuer, elles étaient à la hauteur de mes compétences et de ce que j'étais capable de réaliser. De plus, j'estime qu'elles étaient assez variées, j'ai pu développer une certaine expertise sur le logiciel Docker, mais aussi faire de la programmation, du développement web ainsi que du debugging & testing.

Grâce aux membres de l'équipe biologie des réseaux, cette période m'a permis de pouvoir élargir mon point de vue et mes connaissances en informatique ainsi que sur le monde professionnel. Leurs accompagnements et leurs conseils m'ont permis dans un premier temps de mener à bien mes missions, mais aussi de pouvoir en apprendre plus à m'aider en cas de problèmes.

Difficulté rencontré

J'ai rencontré plusieurs problèmes lors de cette période, beaucoup notamment sur la partie debugging & testing.

Cette partie m'a posé des difficultés, car même si je connaissais déjà les outils du workflow (Singularity, SLURM & Snakemake), je n'étais pas forcément très à l'aise avec eux vu qu'il n'était pas utile pour mes missions.

Néanmoins cette année, je devais approfondir mes connaissances que je possédais sur Docker, mais aussi sur tous ces outils afin de mener à bien cette mission.

J'ai été énormément aidé par une membre de l'équipe, Boujeant Mégane, qui m'a beaucoup accompagné dans la résolution des problèmes d'installation et d'exécution qu'on a pu avoir avec la dernière version du workflow *mimicINT*.

Au-delà de cette mission, les autres moments où j'avais des difficultés que j'estimais ne pas pouvoir résoudre tout seul, je n'ai jamais hésité à solliciter l'aide de mes maîtres de stages et des membres de l'équipe, qui ont su être là pour moi.

Conclusion

Dans le cadre du deuxième semestre de ma licence 3 en informatique, j'ai réalisé un stage de neuf semaines qui s'est étendu jusqu'au 17 juillet 2023.

L'unité de recherche TAGC, rattachée à l'INSERM, m'a accueilli pour faire ce stage afin de finaliser le développement de l'application web *mimicINTWeb*.

Cette application web a pour but de délivrer à la communauté scientifique un outil permettant de prédire les interactions entre les protéines d'un microbe et les protéines d'une cellule hôte et qui soit simple d'utilisation.

Mes missions étaient axées sur la livraison d'une version bêta du site web hébergé, fonctionnel et prêt au phase de test et à l'utilisation étant à ce jour une réussite.

Ce stage finalise ma troisième année en licence informatique. Je souhaite à l'avenir effectuer un master en cybersécurité (FSI) ou en intelligence artificielle (IAAA) afin de finaliser mon cursus scolaire.

Liste des abréviations, acronymes et sigles

TAGC : Theories and Approaches of Genomic Complexity

INSERM : Institut National de la Santé de la Recherche médicale

SQL : Structured Query Language

HTML : HyperText Markup Language

MVC : Modèle Vue Contrôleur

MVT : Modèle Vue Template

SLURM : Simple Linux Utility for Resource Management

CPU : Central Processing Unit

RAM : Random Access Memory

UML : Unified Modeling Language

IFB : Institut Français de Bioinformatique

VM : Virtual Machine

SSH : Secure Shell

RSA : Nom des créateurs de cet algorithme (Ron Rivest, Adi Shamir, Leonard Adleman)

SSL : Secure Sockets Layer

HTTPS : HyperText Transfer Protocol Secure

FSI : Fiabilité et Sécurité Informatique

IAAA : Intelligence Artificielle et Apprentissage Automatique

Bibliographie

- Documentation de Snakemake
<https://snakemake.readthedocs.io/en/stable/>
- Github de mimicINT
<https://github.com/TAGC-NetworkBiology/mimicINT>
- Documentation de Docker
<https://docs.docker.com/>
- Documentation de Bootstrap
<https://getbootstrap.com/docs/5.2/getting-started/introduction/>
- Documentation de SLURM
<https://slurm.schedmd.com/documentation.html>
- Documentation de Python
<https://docs.python.org/fr/3/>
- Documentation de Django
<https://docs.djangoproject.com/fr/4.0/>
- Site de la forge sourcesup
<https://sourcesup.renater.fr/>
- Site OpenClassroom
<https://openclassrooms.com/fr/>
- Site de l'INSERM
<https://www.inserm.fr/>
- MALDONADO Kevin, *finalisation de l'application mimicINTweb*, 2022 (rapport de stage, IUT - Département informatique - Aix-Marseille Université)
- DRETS Lilian, *finalisation de l'application mimicINTweb*, 2021 (rapport de stage, IUT - Département informatique - Aix-Marseille Université)

- CRISTIANINI Marceau, *développement d'une application web dédiée à l'inférence d'interactions hôte-microbiote*, 2020 (rapport de stage, Master en Bioinformatique - Développement de Logiciels et Analyse de Données, Aix-Marseille Université)

Table des illustrations

Figure 1 : Frise chronologique de quelques dates clés où l'INSERM a été un acteur majeur

Figure 2 : Schéma des différentes tâches d'analyse du workflow *mimicINT*

Figure 3 : Schéma des communications entre les conteneurs Docker

Figure 4 : fenêtre de contact de *mimicINTWeb*

Figure 5 : Page jobs de *mimicINTWeb*

Figure 6 : Table des étapes du workflow

Figure 7 : Page de progression de *mimicINTWeb*

Figure 8 : Schéma du chiffrement asymétrique