ÉCOLE DE TECHNOLOGIE SUPÉRIEURE
UNIVERSITÉ DU QUÉBEC

RAPPORT DE PROJET PRÉSENTÉ À
L'ÉCOLE DE TECHNOLOGIE SUPÉRIEURE

COMME EXIGENCE PARTIELLE
À L'OBTENTION DE LA
MAITRISE EN GÉNIE, CONCENTRATION
TECHNOLOGIES DE L'INFORMATION

PAR
CÉDRIC URVOY

CONCEPTION, RÉINGÉNIERIE ET DÉVELOPPEMENT D'UNE INTERFACE UTILISATEUR POUR LE PROJET PROTOTYPE DU LOGICIEL GOAT

MONTRÉAL, LE 18 DÉCEMBRE 2015

PRÉSENTATION DU JURY

CE RAPPORT DE PROJET A ÉTÉ ÉVALUÉ

PAR UN JURY COMPOSÉ DE :

Professeur Alain April, directeur de projet
Département de génie logiciel et des TI à l'École de technologie supérieure

Dr. Pavel Hamet, codirecteur du projet
Département de médecine, Université de Montréal

Professeur Alain Abran, jury
Département de génie logiciel et TI à l'École de technologie supérieure

REMERCIEMENTS

Je tiens à remercier dans un premier temps, toute l'équipe pédagogique de l'École de Technologie Supérieur (ÉTS) et les intervenants professionnels responsables de la formation. Ainsi que l'équipe pédagogique des Hautes Études d'ingénieur (HÉI) qui m'a permis d'effectuer le double diplôme que je réalise actuellement.

Je tiens à remercier tout particulièrement et à témoigner toute ma reconnaissance aux personnes suivantes, pour l'expérience enrichissante qu'elles m'ont fait vivre :

Professeur Alain APRIL (directeur de recherche, département de génie logiciel et des TI à l'École de Technologie Supérieure) pour m'avoir permis d'effectuer mon projet, ainsi que de m'avoir soutenu et accordé sa confiance sur ce projet.

Docteur Pavel HAMET (codirecteur de recherche, chercheur et professeur de médecine à l'Université de Montréal) pour m'avoir accepté dans son service.

Docteur Michael PHILLIPS (spécialiste en médecine de précision) pour m'avoir accompagné dans ce projet au sein du CRCHUM.

Béatriz KANZKI (bio-informaticienne stagiaire) pour sa collaboration au cours du projet et sa bonne humeur permanente.

Ainsi qu'au reste de l'équipe du laboratoire de recherche du Dr. Pavel Hamet au CRCHUM.

CONCEPTION, RÉINGÉNIERIE ET DÉVELOPPEMENT D'UNE INTERFACE UTILISATEUR POUR LE PROJET PROTOTYPE DU LOGICIEL GOAT

CÉDRIC URVOY

RÉSUMÉ

Ce rapport de projet présente les activités de conception, réingénierie et développement réalisés sur le prototype d'application, en génomique « Genetic output Analysis Tool » (GOAT) en collaboration avec le CRCHUM (Centre de Recherche du Centre Hospitalier Universitaire de Montréal). Ce logiciel libre, quand il sera terminé, sera destiné aux professionnels de la recherche en médecine qui souhaitent effectuer des visualisations particulières du génome humain afin d'investiguer les GWAS. Le premier prototype a été réalisé à l'aide du langage Python, du cadriciel Django et de HTML5/CSS3/JS pour l'interface utilisateur. Il a pour objectif d'afficher des informations en tableaux et graphiquement et vise à être plus efficace que les logiciels libres disponibles actuellement, c'est-à-dire LocusZoom ou GWAS pipeline. Plus précisément, GOAT vise à offrir des requêtes interactives et graphiques sur de très grandes quantités de données d'une manière très efficace.

L'objectif de ce projet de maîtrise appliquée, de 15 crédits, vise à améliorer la conception de la première version du prototype et de concevoir des interfaces utilisateurs modernes. Au niveau de la partie « front-end » du logiciel GOAT, l'utilisation de technologies émergentes telles que : React (Facebook Inc., 2015), une librairie JavaScript développée par Facebook, a permis de concevoir une interface claire et rapide. L'utilisation du composant Griddle (Ryan Lanciaux, s.d.), créé par R. Lanciaux, permet aussi la génération de tableaux à l'aide de la librairie React. Finalement, pour le reste du « front-end », l'utilisation de SASS (Hampton Catlin, s.d.) et de la méthodologie SMACSS (Snook.ca Web Development, Inc., s.d.) a permis de mettre en œuvre une conception modulaire au niveau de l'intégration afin que les futures maintenances soit simplifiées. Finalement, afin de réaliser ce projet, il a fallu effectuer la réingénierie du prototype existant et le réadapter selon les normes des représentants de l'équipe de génie logiciel de l'ÉTS.

Mots-Clés : GWAS, Big Data, interface utilisateur, génomique, front-end

USER INTERFACE DEVELOPEMENT FOR PROJECT GOAT

CÉDRIC URVOY

ABSTRACT

This master degree applied research report present the design, reengineering and development done on the genomic application prototype « Genetic output Analysis Tool » (GOAT). The development of the application has been realized in collaboration with the CRCHUM. This open-source software, when completed, will allow medicine research professionals to interactively visualize GWAS information about the human genome of a cohort under investigation. A first prototype of GOAT was developed using Python, the Django framework and HTML5/CSS3/JS for the user interface. Goat's goal is to provide information using interactive Manhattan graphs and associated tables. It aims to be faster and more efficient than the existing open-source softwares available: LocusZoom or GWAS pipeline. GOAT, when finished, will also ensure scalability for interactive and graphical request on a very large quantity of genomic data.

The objective of this 15 credits applied research project is to improve the design of the first version of the prototype and to add a modern user interface to it. In order to offer a simple and fast user interface on the front-end of the application, emerging technologies such as React: a JavaScript library developed by Facebook, was used. Then, the Griddle component (Ryan Lanciaux, s.d.) designed by R. Lanciaux is used to allow quick table generation with the use of the React library. Finally, the use of SASS (Hampton Catlin, s.d.) and the SMACSS methodology (Snook.ca Web Development, Inc., s.d.) allowed for designing a modular approach for the interfaces in order to simplify future maintenances. Finally, reengineering of the prototype was necessary in order to meet the standards imposed by the software engineering team of ETS.

Keywords: GWAS, Big Data, user interface, genomic, front-end

TABLE DES MATIÈRES

Page

LISTE DES TABLEAUX

Page

LISTE DES FIGURES

Page

LISTE DES ALGORITHMES

Page

LISTE DES ABRÉVIATIONS, SIGLES ET ACRONYMES

AJAX : *Asynchronous Javascript and XML*

API : *Application Programming Interface*

CRCHUM : Centre de Recherche du Centre Hospitalier de Montréal

CSS : *Cascading Style Sheets*

ÉTS : École de Technologie Supérieure

GOAT : Genomic Output Analysis Tool

HEI : Hautes Études d'Ingénieur

HTML : *Hypertext Markup Language*

NPM : *Node Package Management*

IDE : *Integrated Development Environment*

IEEE : Institute of Electrical and Electronics Engineers

JSON : *JavaScript Object Notation*

SASS : ***S****yntactically* ***A****wesome* ***S****tyle****s****heets*

SMACSS : *Scalable and Modular Architecture for CSS*

SQL : *Structured Query Language*

URL : *Uniform Resource Location*

INTRODUCTION

Dans le domaine de la recherche médicale, la visualisation des données génomiques est une problématique grandissante. Il est donc nécessaire d'étudier les cas d'utilisation des médecins chercheurs afin d'améliorer les logiciels existants de manière à accélérer la recherche médicale. Dans ce contexte, les nouvelles technologies vont permettre une avancée non négligeable afin de réduire les efforts des bios informaticiens qui sont sollicités, à la pièce, pour effectuer des requêtes sur les bases de données. Ces nouvelles technologies visent aussi à créer une avancée concernant les délais d'obtention des résultats. En effet, avec l'essor du BigData, il devient possible aujourd'hui de traiter de grandes quantités de données efficacement et interactivement plutôt qu'en lot.

Le professeur Alain April, de l'ÉTS, supervise plusieurs étudiants gradués sur des projets concernant le domaine de la recherche en santé. Le projet présenté, dans ce rapport, est réalisé en collaboration avec le CRCHUM. Le CRCHUM est le centre de recherche, en santé, de l'Université de Montréal. Une grande quantité de chercheurs du domaine de la santé travaillent au CRCHUM. Conséquemment, il est intéressant pour le CRCHUM d'utiliser des technologies de pointe pour leurs outils informatiques afin d'accélérer la recherche médicale et les découvertes.

Ce projet de recherche appliquée de 15 crédits, au niveau maîtrise, vise à concevoir, effectuer de la réingénierie et développer des fonctionnalités requises par l'équipe du Docteur Pavel Hamet à partir d'un premier prototype expérimental qui a démontré et validé une preuve de concept. Le tableau 1, qui suit, présente le cadre expérimental (Basili, 1986) de ce projet de recherche :

Tableau 1-1 - Définition du projet

Motivation	Objet	Objectif	Utilisateurs
Aider dans la conception d'une application web sur la génomique. Initialement sur le front-end.	Application permettant l'affichage de graphiques et tableaux suite à une recherche précise sur une zone, un rs_id, un gène.	Utilisation des technologies web afin de combler les besoins d'une visualisation de données complexes	Les chercheurs en génomique.

Tableau 1-2 - Planification du projet

Étapes du projet	Intrants	Bien livrables
-Compréhension des attentes utilisateurs et études sur l'existant; -Prototypage de l'interface et choix des technologies; -Conception et développement de l'interface	-Entrevues avec les différents acteurs du projet; -Réalisation existante sur le projet.	-Prototype visuel de l'application; -Installation sur le serveur d'un prototype fonctionnel Code source du projet.

Tableau 1-3 - Exécution du projet

Développement	Validation	Amélioration des livrables
Conception de l'architecture et de l'organisation front-end	Création de l'interface utilisateur de l'application GOAT	Tableau interactif, Menu latéral rétractable, page de login, réorganisation du « front-end ».

Tableau 1-4 - Interprétation du projet

Contexte	Extrapolation	Travaux futurs
Présentation du l'outil à l'équipe.	Discussion sur la réalisation et rétro-ingénierie du code.	Refonte du « back-end » et mise en place d'un traitement de données multithread

Ce rapport présente le contexte ainsi que les différentes étapes du projet : la conception, le développement et la réingénierie du « back-end ». Une partie sera aussi consacrée aux discussions qui peuvent se poser suite à ce projet.

CHAPITRE 1

CONTEXTE

Afin de comprendre la problématique du projet, il est nécessaire d'établir le contexte autour de ce projet.

1.1 Le CRCHUM & le service :

Le CRCHUM est une institution de recherche sur la médecine qui constitue la partie dédiée à la recherche et l'apprentissage du CHUM. Ces activités concernent principalement : l'éducation, la recherche et l'évaluation des technologies. En recherche, le CRCHUM couvre la recherche fondamentale, clinique et en santé des populations.

L'axe de recherche du service est principalement concentré sur l'étude du diabète, il est donc important de pouvoir étudier des gènes précis et leurs phénotypes.

1.2 Problématique du service et projet GOAT

Le laboratoire de recherche du Dr. Pavel Hamet souhaite créer une nouvelle application pour remplacer un ancien logiciel qui permet d'exécuter des recherches sur des gènes et des phénotypes. Cette application doit être capable d'afficher des graphiques dynamiques pour accélérer l'investigation qui se fait en lot actuellement. Ces graphiques sont des GWAS qui demandent du post traitement important au niveau du « back-end ». Chaque donnée récupérée doit en effet être formater au niveau du « back-end » pour correspondre aux normes requises pour les publications

Un premier projet de développement logiciel a été lancé ayant pour objectif de prototyper ces besoins : le projet GOAT (Genetic Output Analysis Tool). Le projet GOAT est un projet de développement d'un premier prototype d'application web permettant aux chercheurs de valider leurs exigences. À terme il permettra d'effectuer des requêtes rapides sur leur base de

données et de produire des graphiques visuels tels que des GWAS. Le logiciel permet alors, interactivement, d'obtenir des informations sur les gènes, les phénotypes et les biomarqueurs.

Le produit final sera donc destiné aux professionnels du domaine de la recherche médicale qui souhaiteront obtenir des informations ou effectuer des recherches en temps réel sur certains aspects de la génomique de grandes cohortes de patients.

1.3 Réalisation de l'équipe

En septembre 2015, le projet avait déjà un prototype fonctionnel. La génération des graphiques était déjà présente et la partie « back-end » permettait de récupérer les données à visualiser.

1.3.1 Architecture de l'application

Ce premier prototype utilise Python comme langage « back-end ». Il s'appuie sur le cadriciel Django qui est de type MTV (c.-à-d. Model, Template, View). Ce cadriciel permet de gérer efficacement des projets en python via son architecture et facilite les requêtes avec la base de données.

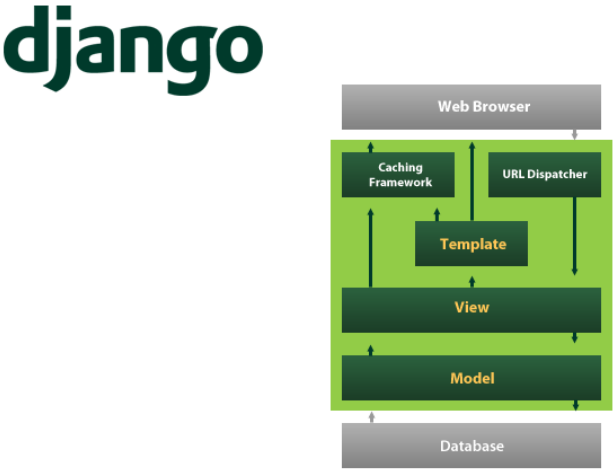


Figure 1-1 - Django Framework (Croft, s.d.)

L’architecture MTV est une architecture très proche du MVC (c.-à-d. Model, View, Controlleur), un patron d’architecture logiciel très utilisé. La majeure différence se situe sur les appellations des catégories. En effet, dans Django les contrôleurs habituels sont remplacés par les vues, alors que les vues sont elles-mêmes remplacées par les « templates ». Les vues sont donc ici directement des pages HTML rendus grâce au moteur de « templates » de Django.

Suite à une étude approfondie du « back-end », et suit à l’a réalisation du « front-end », un problème important est révélé : la conception et l’utilisation de Django pour l’architecture n’avaient malheureusement pas était correctement suivie, car trop complexe sans formation logiciel. Il sera important dans la suite du projet d’effectuer une réingénierie en vue de futurs changements.

1.3.2 Front-end

Au niveau du « front-end », la première version du prototype utilisait Twitter Bootstrap pour un prototypage rapide. Une page index.html permettait alors d’afficher les différentes pages en fonction des données envoyées par le « back-end ». L’un des problèmes de cette mise en place était le manque de modularité concernant la génération des « templates ». Un des objectifs de la première activité de réingénierie consiste alors à retravailler sur cette modularité du « front-end » pour rendre les différentes pages indépendantes les unes des autres.

La figure suivante montre l’interface utilisateur en début du projet :

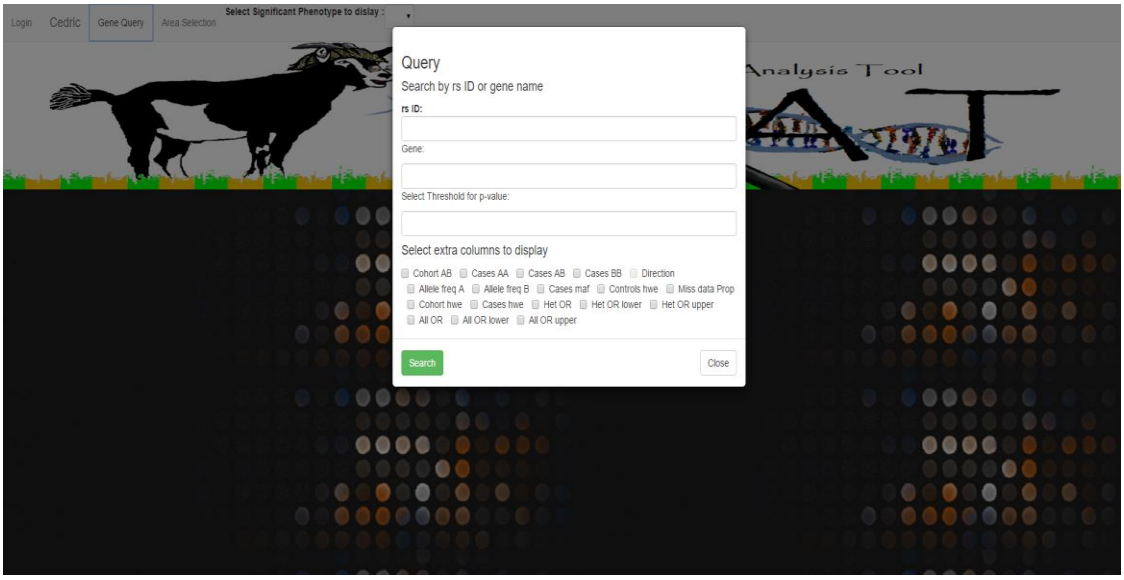


Figure 1-2 - Interface utilisateur précédemment réalisée

1.4 L'objectif de la mission

L'objectif de la mission est de travailler au sein de l'équipe pour comprendre la terminologie et le domaine d'application du projet de logiciel. Dans un premier temps, il sera nécessaire d'étudier le logiciel existant et de reconcevoir le « front-end » afin d'obtenir un code efficace, maintenable et compréhensible pour le logiciel final. De plus GOAT devra être en mesure de répondre aux diverses exigences de qualité et de performance d'un produit de recherche en santé.

1.4.1 Exigences

1.4.1.1 Exigences fonctionnelles

Les exigences fonctionnelles sont présentées dans le document de vision de l'annexe A. Ce document était présent au moment du démarrage de ce projet de recherche appliquée.

1.4.1.2 Exigence non fonctionnelle

Les exigences non fonctionnelles concernant le front-end sont les suivantes :

Accessibilité : le logiciel ne doit pas demander un grand effort d'apprentissage pour pouvoir l'utiliser;

Lisibilité : l'interface se doit d'être claire et concise afin d'éviter que l'utilisateur s'y perdre;

Maintenabilité : le logiciel libre résultant doit être facilement maintenable pour qu'un développeur, extérieur au projet, puisse facilement évoluer l'application pour ses besoins;

Disponibilité : le logiciel doit être disponible en tout temps sur le serveur du CRCHUM. En cas de problème il faut être capable de le relancer rapidement;

Temps de réponse : Au niveau du front-end, l'application doit être en mesure de répondre rapidement aux interactions avec l'utilisateur.

1.5 Conclusion

Ce contexte permet donc de comprendre le projet à réaliser ainsi que les attentes sur ce projet. Il sera donc important de concevoir une architecture « front-end » simple et efficace, mais il faudra également se concentrer sur l'aspect « back-end » du projet qui devra être revu dans une version future.

CHAPITRE 2

CONCEPTION DU « FRONT-END »

Ce chapitre présente la conception de la partie « front-end » de GOAT ainsi que les choix effectués pour le projet (que ce soit au niveau des technologies ou du prototype).

2.1 Livrables

Dans le cadre du projet, plusieurs livrables sont attendus :

- Document de vision : ce document précise la problématique, les différents acteurs du projet ainsi que les objectifs de base. Il présente également certains aspects techniques du projet comme son envergure. Ce document se trouve à l'annexe A;
- Prototype de l'application sur le serveur : l'application GOAT doit être accessible, sur le serveur du laboratoire, afin de pouvoir la tester à l'aide de données réelles;
- Code source : le code source devra être accessible sous License libre.

2.2 Planification du Projet

Dans un premier temps, il a été nécessaire de planifier les étapes du projet. Ces étapes se sont vues modifiées en fonction des problématiques rencontrées et des changements encourus.

La figure 2-1 montre les différentes étapes planifiées ainsi que leurs jalons.

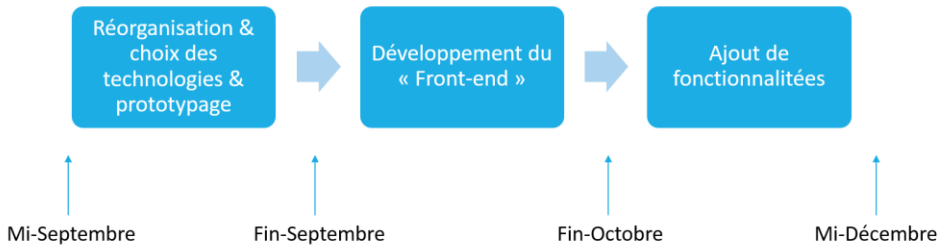


Figure 2-1 - Planification du projet

La première étape vise principalement à réorganiser et améliorer le code du prototype « front-end ». Il a aussi été important de choisir des technologies modernes à utiliser, qui seront en mesure de répondre aux exigences des ingénieurs logiciels (c.-à-d. en termes de modularité, patrons et maintenabilité) et également d'être peu complexe de compréhension pour des maintenances futures.

La seconde étape consiste au développement du « front-end » ainsi que sa mise en place sur le serveur du laboratoire pour les essais. Cette étape vise l'apprentissage, la mise en œuvre et les essais des technologies choisies et la mise en place d'une bonne architecture logicielle.

Finalement, l'ajout de fonctionnalités prévues initialement n'a pas été réalisé et a laissé place à une phase de réingénierie du « back-end » devenue incontournable, car il y avait un impact sur les travaux du « front-end ».

2.3 Réalisation d'un Prototype

Pour confirmer le travail à effectuer et pouvoir présenter une maquette, l'utilisation d'un prototype logiciel peut permettre des rendus réalistes et très proches du résultat final. Plusieurs outils ont été étudiés en vue du prototypage du logiciel. Le tableau 5 présente une analyse comparative de différents logiciels. Cette analyse a été possible grâce au cours MGL 835, Interaction Humain Machine du professeur Michael McGuffin qui avait couvert ce sujet en détail:

Tableau 2-1 - Analyse des logiciels de prototypage

	Ergonomie de l'interface	**Choix de résolution**	**Fidélité visuelle**	**Navigateur de widgets**	**Navigateur d'objet sur le prototype**
Axure	= (trop d'outils)	+	=	=	-
Balsamiq	-	-	- (Faux dessin)	=	-
JustInMind	++	+	++	=	+
Proto.io	+	+	+ (Uniquement pour les mobiles)	- (Pas de recherche)	+

Ici, le logiciel JustInMind Prototyper semble le plus approprié, car il permet d'obtenir rapidement un prototype réaliste, de plus il devient aussi une aide pour le développement du style de l'application. Un prototype général de l'application a donc été réalisé pour rapidement permettre une vue d'ensemble et ainsi faciliter sa validation auprès des clients.

Les rendus obtenus sont les suivants :

La page de login montrée avec la figure 2-2, doit d'être clair et de pouvoir rapidement afficher les erreurs. Le formulaire présenté en figure 2-3

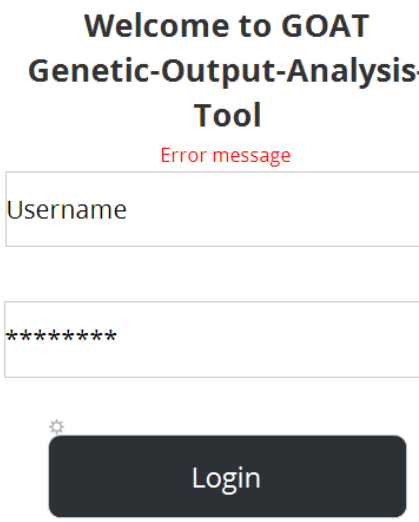


Figure 2-2 - Prototypage de la page de login

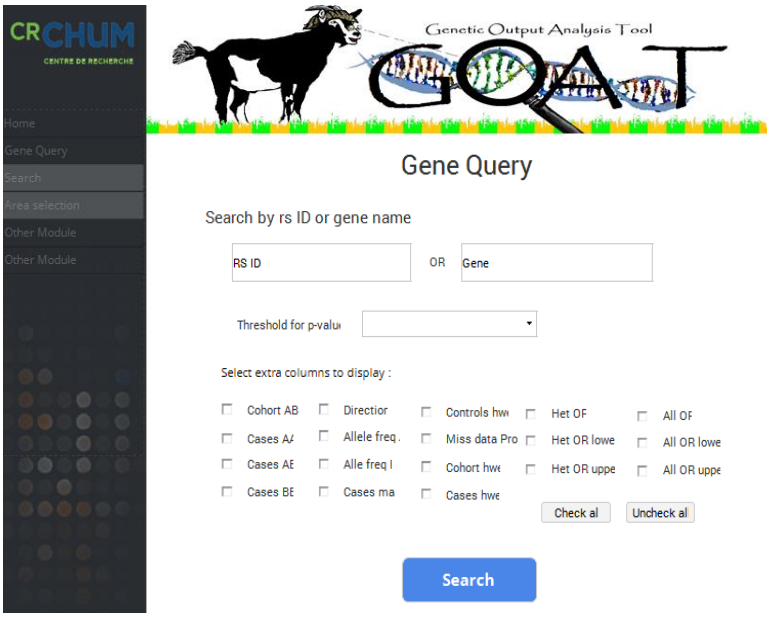


Figure 2-3 - Prototypage de la page de formulaire

La page de résultat, figure 2-4, est l'une des plus importantes, il faut en effet être capable de pouvoir y lire facilement les informations recherchées.

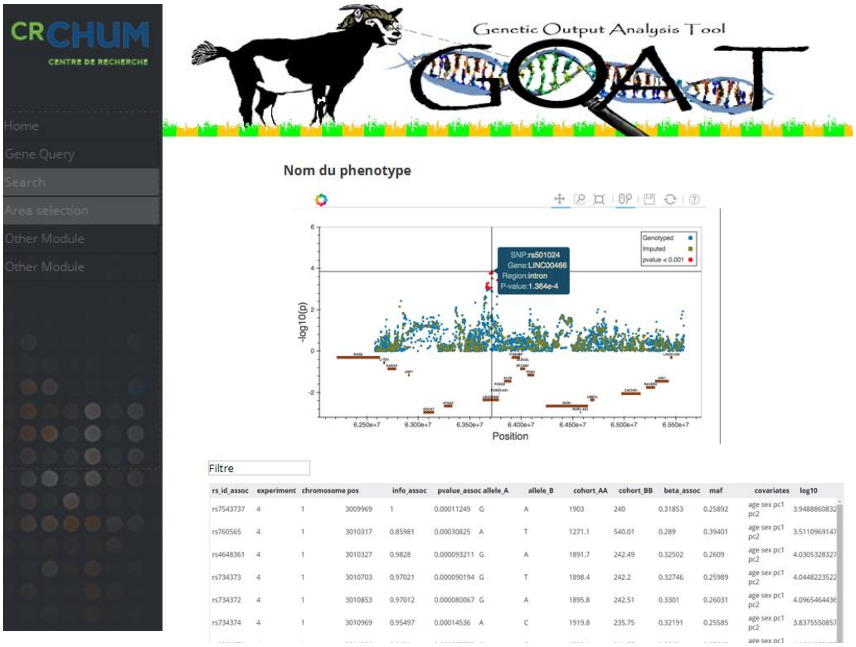


Figure 2-4 - Prototypage de la page de résultat

Ce prototype permet donc une vue d'ensemble de l'application et montre les points importants à travailler. En effet, il est possible de constater par exemple que le tableau pour être affiché clairement doit être d'une grandeur adéquate. Pour cela, il serait intéressant de concevoir un menu latéral qui se rétracte lorsqu'il n'est pas utilisé.

2.4 Choix des technologies

Une des principales activités de la première étape du projet a été le choix des technologies pour le « front-end ». En effet, il existait, au moment de débuter ce projet, sur la partie « front-end » du projet deux problématiques :

1. La première problématique consiste à l'affichage d'un nombre très important de données en temps réel. Pour ce faire il est nécessaire d'utiliser des technologies

« front end » qui ne vont pas ralentir le processus d'affichage et qui vont donc être également capable d'arriver rapidement à traiter ces données envoyées par le « back-end »";

2. La seconde problématique concerne davantage les travaux futurs sur l'application. En effet, si l'outil vient à évoluer et à changer, l'application doit être facilement compréhensible par de nouveaux développeurs et contributeurs. Il faut donc un code maintenable et des technologies qui évoluent bien avec l'architecture de l'application.

2.4.1 Technologie pour le CSS

Le CSS permet la mise en forme des documents HTML. C'est une technologie de base utilisée sur tous les sites web modernes. Néanmoins, il existe aujourd'hui de nombreux langages permettant d'apporter un appui à l'écriture de programme CSS. Ces langages sont des préprocesseurs et il est nécessaire de les transformer en CSS valide pour que les navigateurs puissent les interpréter correctement. Les deux préprocesseurs CSS les plus utilisés sont LESS (Sellier, s.d.) et SASS (Hampton Catlin, s.d.). Ils sont assez similaires dans l'ensemble et n'ont que quelques différences. Le choix des développeurs et donc souvent basé sur leurs habitudes.

Les principaux avantages d'utiliser un préprocesseur :

- Utilisation de variables ;
- Imbrication des propriétés ;
- « Mixins » (Classes qui peuvent être héritées par d'autres classes) ;
- Opérations mathématiques (comme des boucles).

Figure 2-5 - Logo Sass

Pour ce projet, les connaissances existantes en SASS en ont décidé ce choix. De plus SASS une communauté plus importante comme le démontre la figure suivante :

Figure 2-6 - Google trends (Sass et Less)

2.4.2 Librairie JavaScript

Afin d'effectuer des interactions faciles avec l'utilisateur, il est important d'utiliser une technologie efficace qui permet des interactions plus intuitives. Pour cela, il existe actuellement de nombreux cadriciels « front end » disponibles afin de développer cette partie de l'architecture du logiciel GOAT. Cette section les présente.

2.4.2.1 État de l'art sur les cadriciels « front-end »

De nombreux développeurs utilisent pour le front-end des cadriciels JavaScript. Voici un tableau qui présente la popularité des cadriciels ou librairies étudiés au sein de la communauté. Ce tableau est construit avec des chiffres tirés de GitHub (GitHub, Inc, 2015) le 17 décembre 2015.

Tableau 2-2 - Chiffres clés sur GitHub

	Stars	Contributeurs	Tickets traités
AngularJS	45 220	1 370	6 245
BackboneJS	23 720	279	2 164
React	33 200	583	2 222

Angular

AngularJS (Google, 2015), est un cadriciel en logiciel libre développé par Google. Il est actuellement l'un des plus utilisés dans le monde comme le montre le tableau 2-2. C'est en effet le 4^{e} répertoire possédant le plus de start sur GitHub (GitHub, 2015). Ce cadriciel comble le manque de structure de JavaScript sur les applications HTML. Suivant le patron de programmation MVC, il permet de bien structurer l'architecture des applications complexes côté client. Malheureusement, il va passer en version 2.0 dans quelque temps et ne sera pas rétrocompatible. Le risque de son utilisation est donc de ne plus pouvoir le maintenir dans les versions futures de GOAT. De plus, Angular JS nécessite de bonnes connaissances techniques en plus d'avoir une forte architecture « front-end » de type MVC disponible.

Backbone JS

BackboneJS (Jeremy Ashkenas, 2015), créée en 2010, est également un cadriciel MVC très utilisé. Basé sur la librairie UnderscoreJS (Ashkenas, 2015), il structure une application en un

assemblage de vues autonomes. Il a l'avantage d'être très léger et permet aussi d'obtenir de bonne performance. Néanmoins, il est plus difficile à utiliser, car moins structuré et il sert principalement au développement de site « one-page ».

React

React (Facebook Inc., 2015) est, contrairement aux cadriciels présentés précédemment, une librairie JavaScript créée par Facebook sous licence libre BSD (*Berkeley Software Distribution License*) qui permet de l'utiliser sans restriction, qu'elle soit intégré dans un logiciel libre ou propriétaire. À la différence d'Angular, il est possible de l'utiliser simplement en tant que librairie et non cadriciel entier. Il est aussi possible de l'utiliser comme un MVC également en mettant en place une architecture de type Flux (Facebook Inc., 2015) inventée également par Facebook. Cette architecture est un complément à React et est utilisée pour créer des applications web côté client tout comme Angular.

Figure 2-7 - Logo React

React fonctionne comme le V du MVC. Il utilise un concept qui émerge actuellement, celui des composants. Cette librairie permet notamment d'avoir de très bonnes performances de rendu (Harrington, 2015). Cette approche de composant permet de créer des petits éléments indépendants et pouvant être imbriquées les uns dans les autres.

Choix du cadriciel

Les cadriciels tels que AngularJS ou BackboneJS ont un bon potentiel, mais dans l'état actuel du prototype, cela impliquerait d'ajouter de la complexité au code existant. React, en tant que librairie, semble donc un choix intéressant pour ce projet. Il apporte en effet une technologie récente pour obtenir un « front-end » efficace. De plus étant uniquement une librairie et non un cadriciel en son entier, il est possible dans le cadre du projet de ne l'appliquer que sur certains éléments et donc éviter d'encombrer le « front-end » d'une architecture complexe.

Conclusion

Dans le cadre du projet, sera alors utilisé React afin de répondre à des besoins de performance sur des tableaux avec une grande quantité de données. De plus **g**râce à l'utilisation de React il est alors possible de se servir de sous-librairies de composant qui permettent de résoudre des problèmes communs comme la gestion interactive d'un tableau composé de milliers de lignes. Dans le projet, l'utilisation de Griddle (Ryan Lanciaux, s.d.), un composant React sous licence MIT créée par R.Lanciaux et al a contribué à obtenir ce tableau interactif et fonctionnel.

2.5 Réorganisation du front-end

Pour améliorer la lisibilité du code source et également afin de permettre une plus grande modularité, la réorganisation (c.-à-d. la réingénierie) de la partie « front-end »du prototype a été essentielle.

2.5.1 Réorganisation des vues

Dans un premier temps, une organisation plus claire des « templates » rendus par le « back-end » afin de pouvoir rapidement distinguer les différents modules. De plus grâce au moteur

de « templates » de Django, et ses extensions ou ses inclusions de bloc, il est possible de diviser les pages en plusieurs parties réutilisables.

La figure 2-8 présente l'arbre réalisé à la suite de la réorganisation. Il y est montré la séparation du dossier « template » en plusieurs sous dossier, pour permettre une meilleure compréhension de la part d'une personne extérieure.

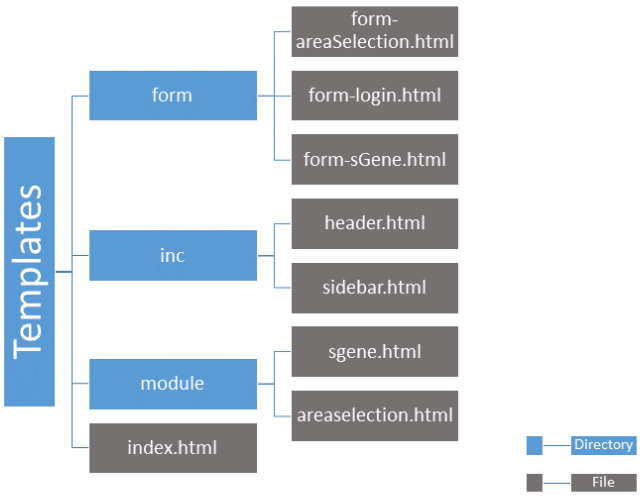


Figure 2-8 - Organisation des vues

2.5.2 Réorganisation du dossier public

Le dossier « Static » est un dossier qui contient la majeure partie des fichiers accessibles publiquement. S'y retrouvent les fichiers de style qui doivent être chargés par le navigateur, ainsi que les fichiers JavaScript et enfin les images.

Tel que présenté dans la section des choix technologiques, les fichiers de styles, en format SCSS, doivent être rendus en format CSS. Ce format permet notamment l'inclusion de fichier et aussi une syntaxe qui permet de ne pas compiler certains fichiers dans leur

homonyme CSS. Pour l'organisation de ce dossier, la méthodologie SMACSS (Snook.ca Web Development, Inc., s.d.) a été utilisée. Ces ensembles de règles favorisent une bonne organisation du code SCSS. Cette architecture se compose en cinq catégories distinctes.

- Base : concerne toutes les règles css touchant les éléments directement (ex: div, h1 …);
- Layout : contiens les règles comme la génération de la grille;
- Module : ce sont les règles concernant des modules particuliers pour des vues précises;
- State : des règles d'état (ex: .isActive);
- Tool : à la différence de SMACSS, l'application n'a pas besoin de thème distinct. Cette partie est donc remplacée par les différentes « mixins » amènes d'être utilisées par le SCSS.

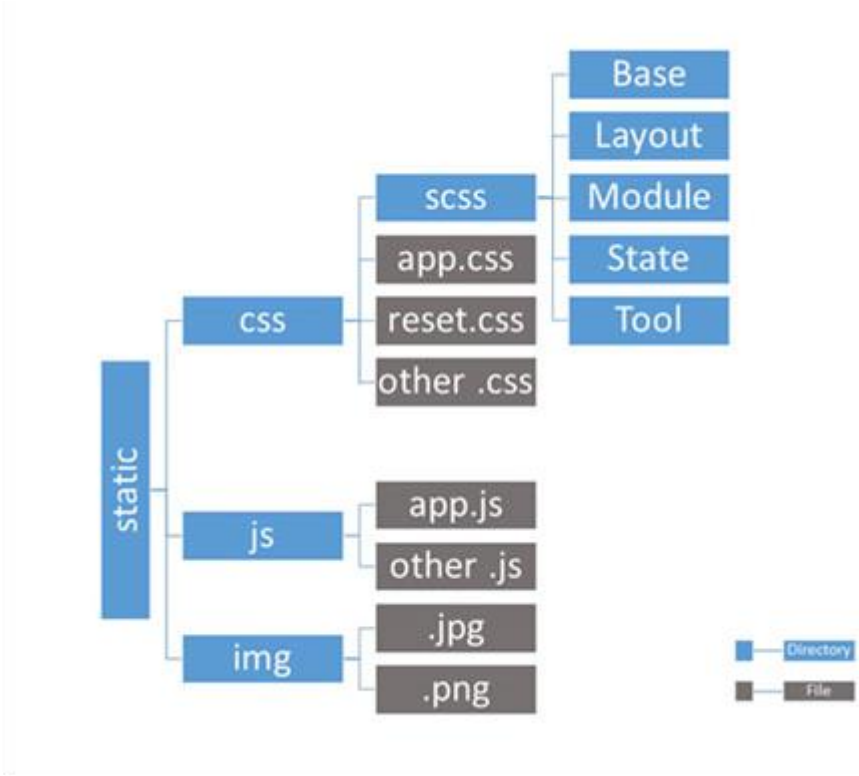


Figure 2-9 - Organisation du dossier public

Cette méthodologie implique donc de respecter l'organisation établie dans la réorganisation des fichiers « Statics ».

2.6 Conclusion

La conception est un des points importants de ce projet logiciel. Une bonne conception architecturale permettra d'encadrer solidement la suite du développement de GOAT et aidera à établir une bonne architecture logicielle du projet. Au niveau du « front-end », la conception proposée amène donc ici à une réorganisation du code.

CHAPITRE 3

DÉVELOPPEMENT DU « FRONT-END »

3.1 Méthodologie

Pour développer l'application, il est nécessaire d'utiliser les bons outils ainsi que le bon fonctionnement. Pour cela, il est important d'établir un bon environnement de travail et un bon processus de travail. Pour aider dans l'organisation du travail un outil comme Trello (Trello Inc, 2015) a permis de définir des tâches et de voir l'avancée sur celle-ci. Trello est un outil de gestion de projet en ligne

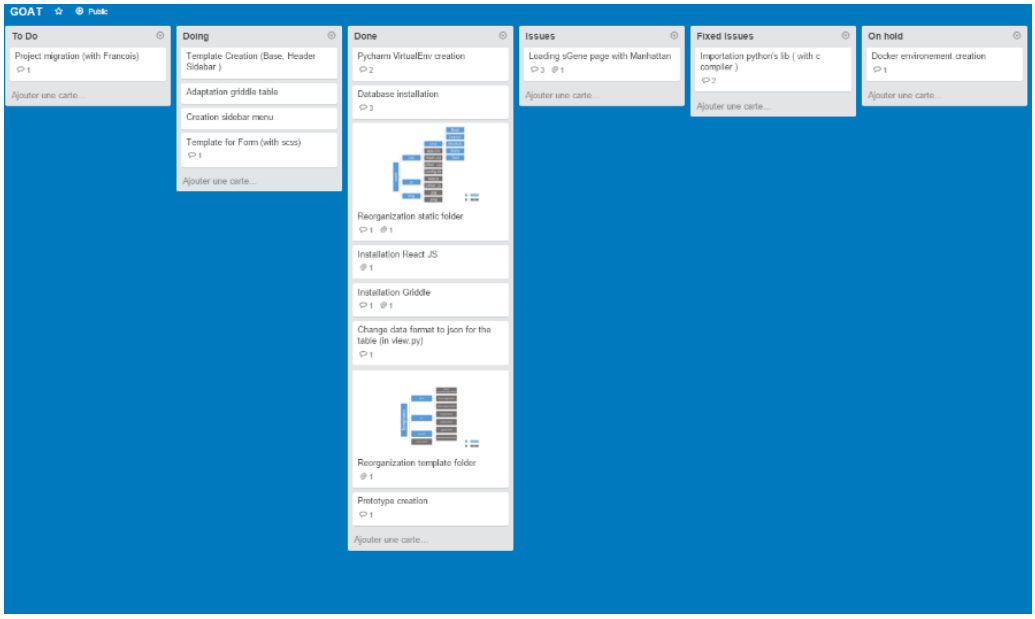


Figure 3-1 - Interface du logiciel Trello

3.2 Outils de développement

3.2.1 IDE et Éditeur de texte

3.2.1.1 PyCharm

Un IDE est très utile quand les connaissances des développeurs ne sont pas très approfondies dans un type de langages. En effet, celui-ci assure une gestion plus simple des nombreux paramètres d'un projet et permet également une grande aide lors du développement. Le projet ayant déjà été développé sur PyCharm (IntelliJ IDEA, s.d.), un IDE développé par JetBrains et se concentrant principalement sur le langage Python. Avoir un IDE orienté Python permet à celui-ci d'être plus complet et particulièrement sur la gestion de projet Django.

Figure 3-2 – Logo de PyCharm

3.2.1.2 Atom

Pour développer sur la partie « front end » un simple éditeur de texte tel qu'Atom ou SublimeText qui sont grandement personnalisables selon les besoins permettent de travailler sur des environnements connus.
Atom (GitHub, Inc., 2015) est un outil développé par GitHub, qui est fortement modulable et moins chargé qu'un IDE.

Figure 3-3 - Logo d'Atom

3.3 Environnement de travail

La première étape du projet a été la mise en place d'un environnement de travail compatible avec le projet. Le projet développé en python et nécessitant de nombreuses librairies, il est important de mettre en place la totalité de ces librairies afin de pouvoir développer dans de bonnes conditions et afin que le projet fonctionne.

Dans un premier temps, était envisagée l'installation de Docker (Docker, 2015), un logiciel libre permettant le déploiement d'applications dans des conteneurs. Cela aurait permis de simplifier l'installation des librairies sur les serveurs ou ordinateurs par la simple installation d'un conteneur Docker contenant déjà toutes les librairies. Des problématiques se sont alors imposées et une autre solution a dû être trouvée.

Finalement PyCharm offre une solution simple, car il permet directement de créer un environnement virtuel pour chaque projet et ainsi donc éviter les incompatibilités. La mise en place de cet environnement est fondamentale pour correctement effectuer la suite du projet. Sa création a été néanmoins assez difficile, car de nombreux problèmes dus à Windows et aux interpréteurs de C causaient des soucis lors de l'installation de certaines librairies.

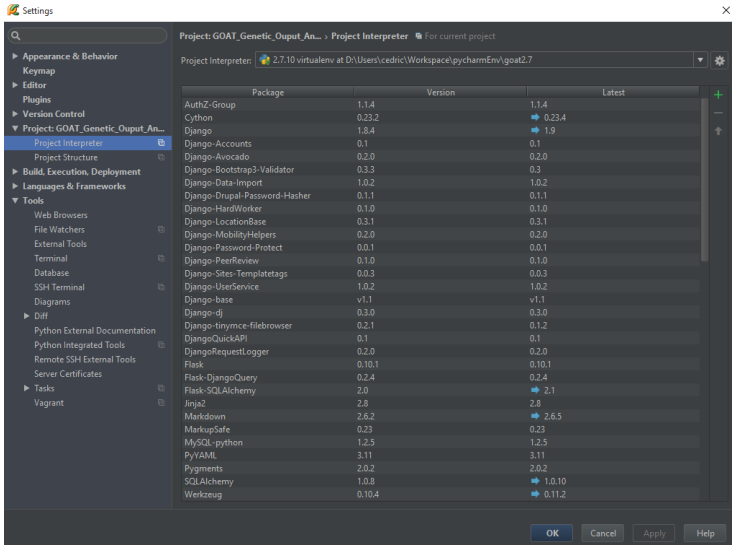


Figure 3-4 - Environnement virtuel avec PyCharm

3.4 Processus de travail

3.4.1 Développement du « front-end »

Dans un premier temps et pour travailler efficacement sur le front-end, l'installation d'une base de données en local et une limite sur les requêtes a été effectuée. Les problèmes de performances de l'outil ont rendu obligatoire cette limite sur les requêtes pour éviter une simple attente de cinq minutes pour chaque modification d'un fichier SASS.

Pour améliorer la rapidité lors des modifications SASS, un outil tel que GULP est important. Cet outil permet de grâce à un fichier de configuration de lancer des automatisations de tâches. Pour utiliser cet outil, il a fallu installer NPM sur le projet afin d'obtenir GULP facilement. GULP permet donc de lancer via la commande « gulp watch » définie dans le fichier, de vérifier toute modification dans le dossier « static » et d'effectuer alors une compilation des fichiers SASS et une récupération des fichiers « statics » souhaités par Django.

```
var gulp = require('gulp');
var shell = require('gulp-shell');
var changed = require('gulp-changed');
var plumber = require('gulp-plumber');
var sass = require ('gulp-sass');

var OPTIONS = {
    COLLECSTATIC:{
        watch:'static/**/*.*'
    },
    SASS:{
        src:'static/css/scss/app.scss',
        dest:'static/css'
    }
}

gulp.task('sass',function(){
  return gulp.src('static/css/scss/app.scss')
    .pipe(sass().on('error',sass.logError))
    .pipe(gulp.dest('static/css'))
});

gulp.task('collectstatic',['sass'], shell.task([
  'python manage.py collectstatic --noinput --ignore "*.scss"'
]));

gulp.task('watch', function () {
    gulp.watch('static/**/*.*');
    gulp.watch('biomarqueurs/templates/**/*.*');

});

gulp.task('default',['collectstatic'],function(){

})
```

Algorithme 3-1 - Fichier gulpfile.js

3.4.2 Développement sur le serveur

Afin de pouvoir tester l'application sur le serveur, et ainsi pouvoir passer les problèmes de performances sur des ordinateurs personnels, une installation a été requise. Les diverses librairies utilisées sur le projet ont dû être importées et installées sur le serveur. Une fois cette étape passée, il est alors possible de tester l'application. Un serveur est en permanence lancé à l'adresse suivante : http://10.54.203.119

Un processus a dû être mis en place, pour être capable d'installer les modifications apportées entre chaque phase de développement.

Dans un premier temps, il est requis d'installer sur son ordinateur PUTTY (seulement nécessaire pour la première connexion) ainsi que PSCP.exe (pour le transfert de données entre le serveur et l'ordinateur)

Pour se connecter, il est faut utiliser la commande suivante :

```
ssh 10.54.203.119

username@10.54.203.119's password:
```

Après avoir accédé au dossier du projet, il est possible à présent lancer un second serveur afin d'accéder aux logs plus facilement.

```
python manage.py runserver 0.0.0.0:8000
```

Pour une meilleure productivité, l'utilisation de deux autres consoles connectées en ssh au serveur est recommandée. L'une avec la commande «top » de lancée pour suivre l'activité du serveur en fonction des requêtes, ainsi qu'une seconde pour effectuer de rapides modifications sur les fichiers du serveur.

Pour le transfert des fichiers sur le serveur, il faut utiliser la commande suivante (l'option «-r » est pour copier le dossier en entier, sinon il faut préciser le fichier à transférer) :

```
pscp.exe -r C:\your\path\project\folder 10.54.203.119:/server/path/project/folder/
```

Cette méthodologie de travail a permis de pouvoir tester les changements effectués sur le projet.

3.5 Modules

3.5.1 Login

Le premier module vu par l'utilisateur est celui du « login ». Il était donc important d'avoir une page claire. Le « front-end » a été assez rapide à développer, car assez simple, mais permet de montrer la base des autres modules. Pour chaque module est utilisée la fonctionnalité de « template » de Django qui implique d'étendre d'autres modules. L'algorithme 3-2 montre la page index.html, qui est la base de tous les autres modules. Ce code permet d'importer toutes les librairies pour le « front-end ». Ensuite la page « login » étend donc directement cette page-ci.

```
<!DOCTYPE html>

<html xmlns="http://www.w3.org/1999/xhtml"
xmlns="http://www.w3.org/1999/html">

    {% block head %}
    <head>
      <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
        <title>GOAT</title>

        <!-- StyleSheets -->
        <link href="../static/css/bokeh-0.9.0.min.css" rel="stylesheet">
        <link href="../static/css/app.css" rel="stylesheet" />

        <!-- Scripts -->
        <script src="../static/js/jquery-2.1.4.min.js"></script>
        <script src="../static/js/jquery.cookie.js"></script>
        <script src="../static/js/bootstrap.js"></script>
        <script src="../static/js/bokeh-0.9.0.min.js"></script>
        <script src="../static/js/underscore-min.js"></script>
        <script src="../static/js/JSXTransformer.js"></script>
        <script src="../static/js/react.js"></script>
        <script src="../static/js/griddle.js"></script>
        <script src="../static/js/app.js"></script>
    </head>
    {% endblock head %}

    {% block body %}
        <body>
        </body>
    {% endblock body %}

</html>
```

Algorithme 3-2 - index.html

```
{% extends "index.html" %}

{% block body %}
<body>

    {% block container %}

    <div class="" id="login">
        <img src="../static/img/GOAT.png" alt="goat" class="logo-con-
tainer"/>
        <form method="post">
            <input type='hidden' name='csrfmiddlewaretoken'
value='MSBWTwBs4iNmhZY2I07GgqJxWVTCAPNN' />
            <div class="login-form">
                {% csrf_token %}
                <h3 class="login-title">Welcome to GOAT</h3>
                <h4 class="login-subtitle">Genetic-Output-Analysis-
Tool</h4>
                <p class="login-error">{{ error|safe }}</p>
                <div class="login-block">
                    <h1>Login</h1>
                    <input type="text" value="" placeholder="Username"
id="username" name="username"/>
                    <input type="password" value="" placeholder="Password"
id="password" name="password" />
                    <input type="submit" class="login-btn" value="Login"/>
                </div>
            </div>
        </form>
    </div>

    {% endblock container %}

</body>
{% endblock body %}
```

Algorithme 3-3 - login.html

Cette organisation des « templates » permet d'avoir de nombreux blocs indépendants et donc une maintenabilité beaucoup plus facile. De plus cela oblige un code clair et organisé au niveau du HTML.

Au niveau du « back-end », la partie « login » a nécessité des modifications afin d'obtenir les bonnes pages et suivre les bonnes URLs. L'application utilise ici les fonctions du module django.contrib.auth qui supporte la gestion des sessions utilisateurs et de la relation avec la base de données.

```
def index ( request ) :

    if request.user.is_authenticated():
        return render(request, 'home.html', {'content':'modules/home',
'username': request.user})
    else:
        if request.method == 'POST':                    #Si methodes est post
            username = request.POST['username']         #demander user name
et mot de passe
            pwd = request.POST['password']
            user = authenticate(username=username, password=pwd)  #authen-
tifie l'utilisateur
            if user is not None:
                if user.is_active:
                    login( request, user)                       #retourner le
nom de l'utilisateur
                    request.session.set_expiry(300)
                    return render(request, 'home.html', {'content':'mo-
dules/home', 'username': request.user})
                else:
                    return render(request, "modules/login.html")
            else:
                print ("erreur de login")
                return render(request, "modules/login.html",{'error':'Er-
reur de login'})
        else:
            return render(request, "modules/login.html")
```

Algorithme 3-4 - Gestion des requêtes pour les connexions

De plus une fonction de déconnexion a été ajoutée.

```
url(r'^logout/','biomarqueurs.views.logoutUser',name="logoutUser"),
```

3.5.2 Page d'accueil

Une simple page d'accueil a été ajoutée au projet. Cette page présente les différents modules (présent ou future) de l'application. Il est possible d'accéder aux formulaires de requête. Le développement de cette page a permis de créer le « layout » du projet ainsi que les bases. Le menu latéral est présent sur cette page. Il est géré directement avec les fonctionnalités CSS de « hover ». Cela permet d'éviter de surcharger la page de JavaScript. Par contre cela peut être mal interprété sur d'anciens navigateurs.

La figure présente un exemple de l'utilisation de « hover » ainsi que l'indentation permise dans SASS

```
.side-nav:hover {
width:250px;
overflow:visible;
  >ul{
    visibility: visible;
    opacity: 100;
    display: block;
  }
  >.side-icon{
    visibility: hidden;
    opacity: 0;
    display: none;
    z-index: 5;
  }
}
```

Algorithme 3-5 - exemple de code avec SASS pour le menu latéral

Ce menu latéral, visible dans les résultats permet d'accéder rapidement aux formulaires des différents modules et est rétractable pour laisser un plus grand espace aux résultats affichés

3.5.3 Gene Query

Pour le module principal, des changements ont dû dans un premier temps être effectué sur le « back-end ». En effet, l'application renvoyait suite au formulaire, la même page, avec directement des retours HTML générés dans le « back-end ». Pour améliorer les performances et dispenser uniquement des variables à la nouvelle page, une modification du code donné lieu à un envoie de données au format JSON. C'est grâce à ce format que le tableau pouvait être géré facilement.

```
sorted_data_seuil_json = sorted_data_seuil.to_json(orient='records')
```

Une fois les données envoyées au « template » avec le format JSON, il est possible de générer le tableau avec Griddle. Au niveau du « back-end », les données sont gérées par une librairie python : Pandas (PyData Development Team, s.d.). La documentation de Pandas indique les options pour le format de sortie du JSON. Celle qui convient le mieux est alors

l'orientation « records », qui possède un format du type : [{column -> value}, ... , {column -> value}]

L'un des points importants était également la possibilité de changer le phénotype présenté. Une liste présentant sur les phénotypes importants sur le « gene query » effectué permet alors de changer de phénotype et de d'effectuer de nouveau une requête.

L'utilisation de React a permis de générer des composants pour les liens dans les tableaux. En effet, il est possible de gérer facilement avec React et Griddle, une récupération d'informations sur les autres données du tableau. Le code suivant montre l'utilisation d'un composant React gérer un lien au niveau de la position des rs_id pour directement remplir le formulaire d' « area selection ».

```
var LinkComponent = React.createClass({
    handleClick:function(e){
        $('.areaSelection').children("input[name$='posi-
tion']").val(this.props.rowData.pos);
        $('.areaSelection').children("input[name$='chromo-
some']").val(this.props.rowData.chromosome);
    },
    render: function(){
        url ="javascript:openForm('form-areaSelection')";
        var boundClick = this.handleClick.bind(this);
        return <a href={url} onClick={boundClick}>{this.props.data}</a>
    }
});
```

Algorithme 3-6 - composant React pour les liens

Ce système de composant fonctionne un peu comme une classe avec de multiples instanciations. Il est donc possible d'avoir des fonctions différentes selon chaque instanciation.

3.5.4 Area selection

Ce module est très semblable à « gene query », à part qu'il possède lui un graphique généré avec Bokeh. Il est appelé à la suite d'une recherche « gene query » en indiquant la position et

le chromosome du rs_id voulu pour effectuer un zoom sur cette zone. Ici les composants pour les liens ont été également très importants afin de pouvoir afficher dans le tableau, des liens vers les sites d'information de génomique telle que dbSNP.

3.6 Conclusion

Suite au travail sur le « front-end », une des problématiques révélées est la mauvaise utilisation de Django comme architecture du « back-end ». En effet, les modules crées sont dépendent les un des autres et ne compose à eux seuls qu'une seule définition dans le « back-end ». Pour rendre le projet maintenable et modulable, il va donc être important de retravailler sur ce point en effectuant une réingénierie du projet.

CHAPITRE 4

RÉINGENERIE DU « BACK-END »

Dans ce projet, le prototype actuel n'est pas en mesure de répondre correctement à toutes les demandent de changement sur son fonctionnement. Il est donc essentiel d'effectuer ce processus de réingénierie pour aider dans le développement futur du projet.

4.1 Méthodologie de réingénierie

Pour effectuer une réingénierie du logiciel, comme vu dans la figure 4-1, il faut dans un premier temps effectuer une rétro-ingénierie afin d'étudier l'existant dans un projet pour en déterminer le fonctionnement et ainsi être capable de représenter ce système pour l'améliorer et le reconstruire.

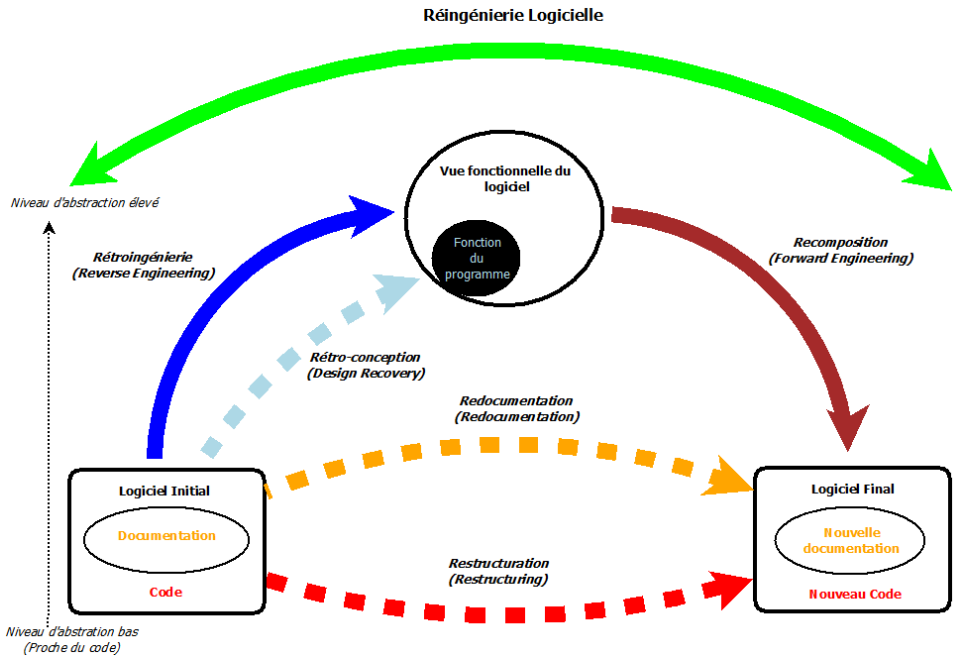


Figure 4-1 - Réingénierie

Il est donc important de comprendre le fonctionnement du « back-end » de l'application avant d'effectuer sa recomposition.

Plusieurs outils peuvent être utilisés pour aider dans le processus de rétro-ingénierie afin d'évaluer la complexité du système existant :

- L'évaluation du code source de l'application
- L'étude de logiciels existants
- Les discussions avec les parties prenantes

Ces techniques permettent de comprendre correctement le code pour pouvoir passer à l'étape suivante : la recomposition. Cette recomposition de l'application a pour objectif de suivre l'analyse de rétro-ingénierie afin de corriger les problématiques du logiciel existant.

4.2 Couplage et cohésion

Afin d'effectuer une rétro-ingénierie efficace sur ce projet, il est important de connaître les concepts de couplage et de cohésion dans un logiciel.

Le couplage informatique est une notion qui définit le degré d'interaction entre les différents modules ou composant d'un logiciel. Un couplage fort implique donc une faible modularité au sein du logiciel et donc des risques de ruptures si des modifications sont effectuées. Un couplage faible implique lui des modules dépendants qui communiquent via des interfaces.

La cohésion informatique définit le degré de responsabilité de chaque module. La cohésion est forte lorsque la responsabilité de chaque module est faible alors qu'elle est très faible si un seul module effectue de nombreuses tâches et possède donc de fortes responsabilités.

4.3 Application au projet

Ici à la différence de la plupart des rétro-ingénieries en informatique, celle-ci n'est pas concentrée sur la structure de la base de données, mais sur l'architecture en elle-même de l'application.

En effet suite à des discussions avec les diverses parties prenantes du projet, le fonctionnement du projet a changé, mais les changements sur le processus d'obtention des résultats se sont vus impossibles à réaliser suite à un trop fort couplage de l'application.

En effet, une analyse du « back-end » a démontré qu'une seule définition au niveau de « view.py » gérait toutes les requêtes différentes au sein de l'application. Dans cette définition s'y trouve la requête permettant d'effectuer une « gene query » ainsi que la requête permettant d'effectuer une « area selection ». Il est donc important de séparer ces modules pour permettre une modularité plus grande de l'application et ainsi diminuer ainsi son couplage.

De plus, un autre problème majeur et une mauvaise utilisation d'une architecture REST. En effet, l'application actuelle n'utilise que des requêtes POST pour accéder à ses données, or celle-ci devrait principalement utiliser des requêtes GET. Ces différentes requêtes POST sont alors traitées suivant une unique URL et non pas des URL différentes selon les types de résultats voulus.

Après discussion avec l'équipe, il est en effet possible de rendre l' « area selection » indépendante. Pour cela, elle doit prendre en paramètres d'entrée, le phénotype, le « threshold », la position, le chromosome et l'intervalle voulus sur la sélection. En faire une requête différente à part entière permettrait d'éviter les nombreuses erreurs qui peuvent avoir lieu entre les deux modules actuellement. Cela permettrait aussi aux utilisateurs d'effectuer directement des requêtes qui les intéressent.

Au niveau de la cohésion, il est également possible d'observer des fonctions comme « area sélection » qui effectue un très grand nombre de tâches, que ce soit le traitement des données, son formatage puis son écriture sur le « template ». Il serait important de séparer ses modules afin d'obtenir une cohésion plus forte ainsi qu'un code plus clair et ordonné.

Enfin, un des soucis majeur de l'application est une faible performance lors de la récupération des données et leur formatage en « dataframe ». Il en résulte un temps important lors d'une requête « gene query ». Deux solutions sont alors envisageables :

- Dans un premier temps, utiliser les modèles Django pourrait possiblement permettre un gain de temps lors de cette requête. En effet, dans le cas actuel, c'est directement Panda qui fait l'interface avec la base de données et non Django.
- Mise en place de Spark pour améliorer le traitement des données.

4.4 Conclusion

La phase de rétro-ingénierie, non prévu au début du projet a néanmoins était indispensable, car le code était bloquant pour le développement « front-end ». Par manque de temps sur ce projet, la recomposition n'a pas pu avoir lieu, mais un partage de connaissance peut permettre d'arriver plus rapidement à corriger ces problèmes.

CHAPITRE 5

RÉSULTATS

5.1 Présentation des résultats au niveau « front-end »

Les figures suivantes présentent les différentes vues de l'application suite au développement de l'application.

5.1.1 Page de login

Cette page permet la connexion à l'application. C'est également la page sur laquelle il est possible d'être redirigé si la connexion au serveur est arrivée à terme. Pour l'utilisateur il est important d'avoir une page claire à cet endroit. Un module de récupération de mot de passe pourra être ajouté dans une prochaine version de l'application.

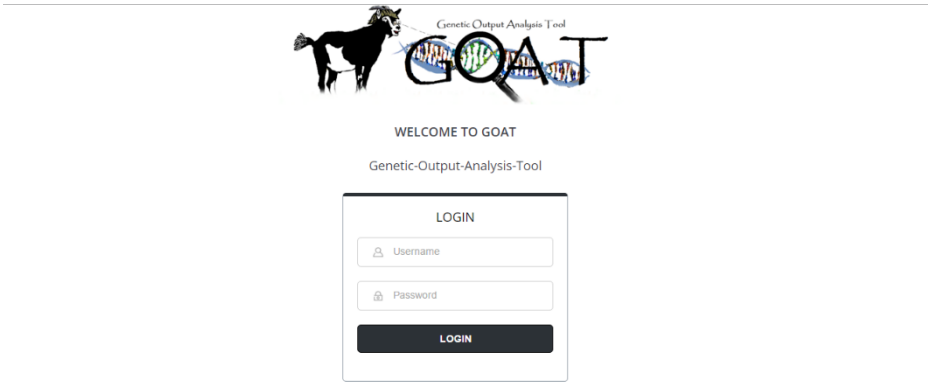


Figure 5-1 - Page de login

5.1.2 Page d'accueil

Sur cette vue, les différents modules de l'application pourront être rajoutés avec leur description et leur lien. Il est également possible de voir le menu latéral non rétracté dans la figure 5-2.

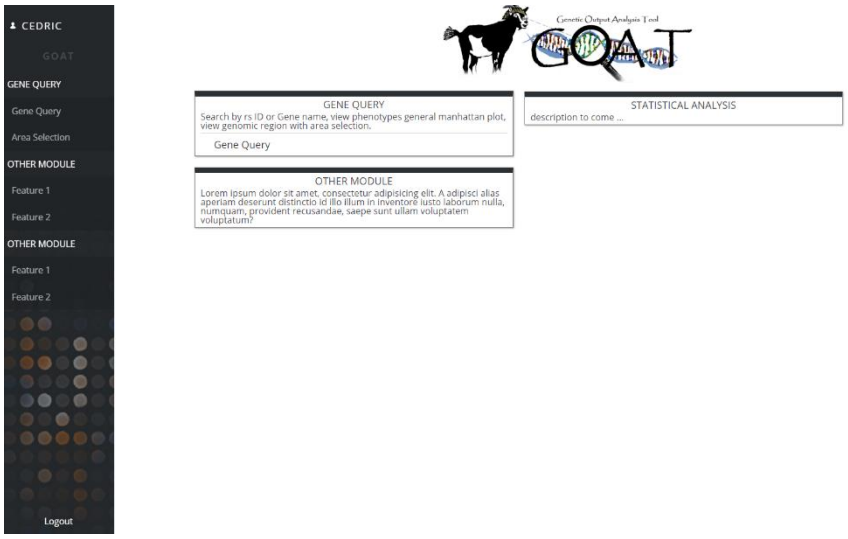


Figure 5-2- Page d'accueil avec descriptifs des modules

5.1.3 Gene Query

Ce module est composé de deux vues qui dans la version actuelle sont dépendantes l'une de l'autre.

Dans un premier temps, il faut effectuer une requête sur un « rs_id » ou sur un gène. Cela procure alors la génération d'un fichier de type « Manhattan » ainsi qu'un tableau d'informations. Ceci est montré avec les figures 5-3 et 5-4.

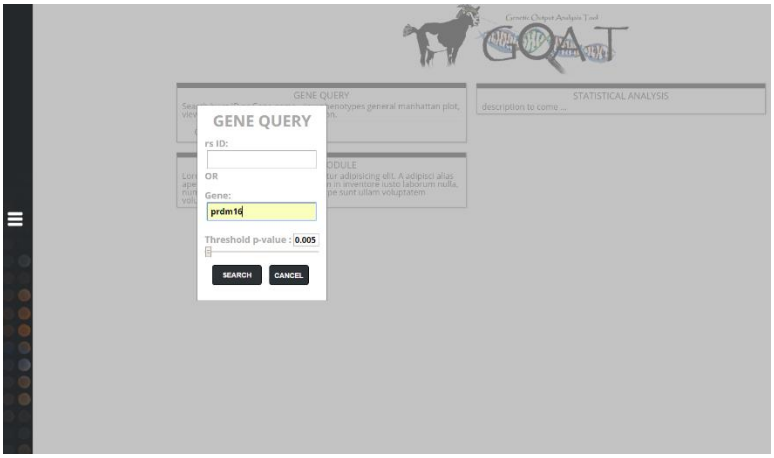


Figure 5-3 - Formulaire GeneQuery

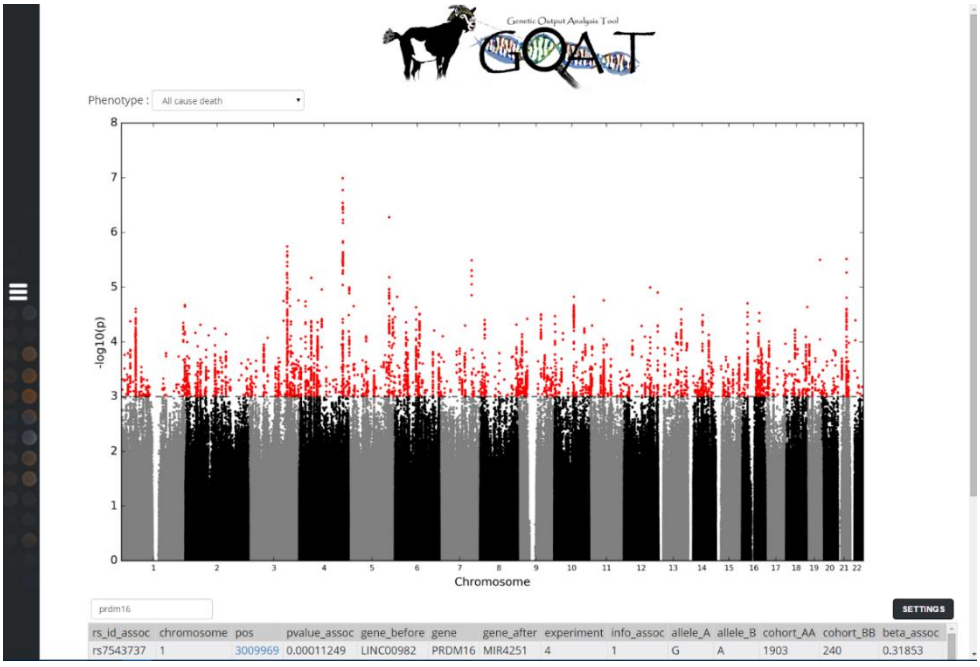


Figure 5-4 - Résultat GeneQuery

Dans un second temps, l'utilisateur peut cliquer sur une des positions dans le tableau et ainsi accéder au formulaire d' « area selection » pré rempli, figure 5-5. Cette requête permet ensuite d'accéder à un une seconde vue possédant un graphique généré avec « Bokeh server » et un nouveau tableau possédant des liens vers les sites d'informations sur les gênes ou les « rs_id ».

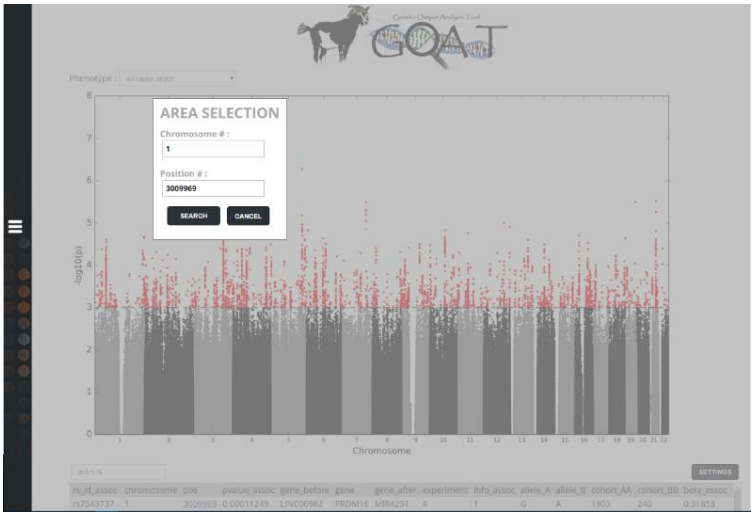


Figure 5-5 - Formulaire d'Area Selection

La page de résultat de l'« *Area Selection* », figure 5-6, permet de voir le type d'interaction que pourrait avoir le graphe de Manhattan dans un futur proche si celui-ci n'est plus généré sur le serveur, mais du côté client. Le tableau présenté ici bas, permet rapidement d'accéder aux différents sites regroupant des informations sur des « rs_id » ou des gènes particuliers.

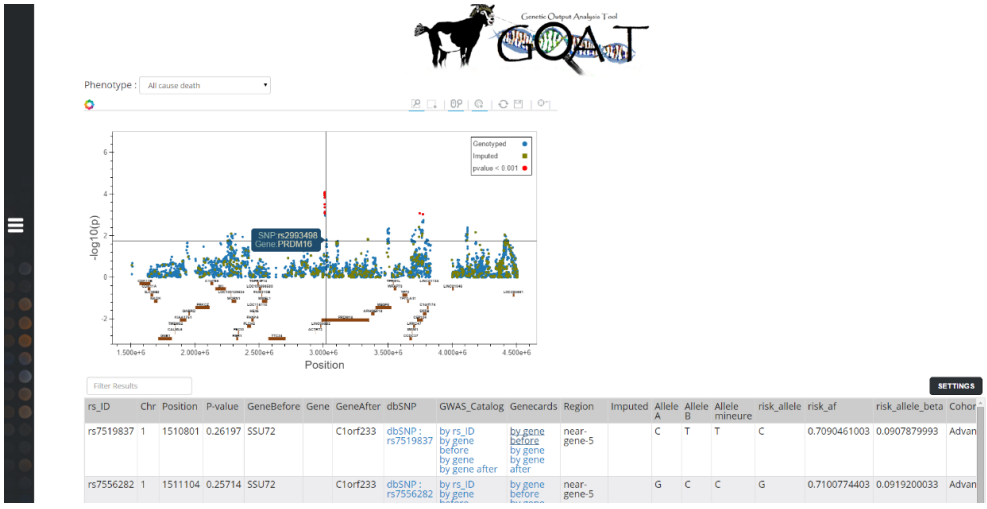


Figure 5-6 - Résultat Area Selection

CHAPITRE 6

DISCUSSION

6.1 Difficultés techniques rencontrées

6.1.1 Utilisation de Docker

L'utilisation du langage de programmation Python demandant l'installation de nombreuses librairies variées pour chaque projet il aurait été intéressant de déployer une infrastructure Docker afin de pouvoir par la suite rapidement le partager sur serveur ou avec d'autres collaborateurs. Malheureusement sans les connaissances suffisantes et à la suite de nombreuses incompatibilités en septembre 2015 entre Docker, Virtual Box et Windows 10, l'objectif a été annulé. La création d'un simple environnement virtuel créée avec PyCharm a été utilisée pendant le reste du projet.

6.1.2 Mise en place de la base de données en local

La base de données de test étant très volumineuse (4,5 go), il a été difficile de l'installer sur un ordinateur Windows avec WAMP (plateforme de développement web permettant la création facile d'une base de données locale). En effet, la taille du fichier n'était pas acceptée par PHPMyAdmin qui s'occupe de la gestion de la base de données. Le seul moyen trouvé a été d'importer la base de données via une commande :

```
mysql -u root -p hits-of-opti-tera< BackupSQL.sql
```

6.1.3 Architecture du « back-end »

L'un des problèmes importants sur le projet est son architecture « back-end ». En effet dû à un manque de connaissance de l'équipe sur le cadriciel Django, celui-ci n'a pas était utilisé à son plein potentiel et en a résulté une architecture très peu modulable et provoquant des

soucis de performances ainsi que de nombreuses difficultés d'intégration avec le « front-end ».

L'un des objectifs majeurs sur la suite du projet est donc une refonte complète du le « back-end » de l'application.

Les principaux problèmes de l'architecture sont :

- L'utilisation d'une URL unique;
- L'utilisation de POST uniquement au lieu de GET;
- La non-utilisation des modèles définis.

6.1.4 Utilisation d'Ajax

Pour les formulaires, des tests le choix a été dans un premier temps d'utiliser AJAX pour permettre des requêtes asynchrones et avoir donc un retour efficace sur les erreurs qui étaient traitées au niveau du « back end ». Néanmoins après de multiples essais, le choix s'est révélé infructueux. En effet, il alors qu'il est possible de gérer des erreurs sur la vue, il est en ce moment non recommandé d'utiliser AJAX, car il n'est pas possible d'y appliquer des « renders » classiques. Le « back-end » pose ici un souci dû à son architecture, car il n'est composé que de requêtes POST et ne permet pas la redirection sur un GET. Dans des travaux futurs, il serait intéressant de réutiliser AJAX pour permettre une validation des données au niveau du « back-end ».

```
// Setup AJAX pour la gestion des token
var csrftoken = $.cookie('csrftoken');

function csrfSafeMethod(method) {
    return (/^(GET|HEAD|OPTIONS|TRACE)$/.test(method));
}
$.ajaxSetup({
    beforeSend: function(xhr, settings) {
        if (!csrfSafeMethod(settings.type) && !this.crossDomain) {
            xhr.setRequestHeader("X-CSRFToken", csrftoken);
        }
    }
});

//  Ajax sur Submit
$(document).ready(function() {
   $("#SGene").submit(function(event){
        $.ajax({
             type:"POST",
             url:"/SGene/",
             data: {
                    'rs_ID': $('#rs_ID').val(),
                    'gene': $('#gene').val(),
                    'threshold': $('#threshold').val()
                    },
             success: function(data){
                    console.log(data);
             },
            error: function(error){
                console.log(error);
                $("#error").text(error.responseText);
            }
        });
        return false;
   });
});
```

Algorithme 6-1 - utilisation d'Ajax pour une version future

Pour l'utilisation d'Ajax pour la prochaine version, il est nécessaire de faire un Setup comme suit pour son utilisation avec les « csrftoken ». Dans le fichier view.py, la réponse doit être retournée selon une réponse HTTP contenant au minimum un message et un statut :

```
return HttpResponse("YOUR THRESHOLD IS INVALID, PLEASE SELECT A THRESHOLD UNDER 1", status=400)
```

6.2 Constats

6.2.1 Travail chez le client

Travailler à l'hôpital n'a pas toujours était simple. En effet, le fait de travailler chez le client impose une pression de résultat et cela peut entraîner le bâclage de certaines tâches pour avoir un prototype fonctionnel. De plus, faire partie intégrante de l'équipe oblige à être présent à la réunion du service tous les mercredis matin. Bien qu'intéressante, d'un point de vue théorique, les sujets abordés étaient très spécifiques et peu pertinentes pour le projet en cours. Travaillé en extérieur aurait peut-être pu permettre d'imposer des réunions avec le client plus régulières et ainsi travailler de façon agile sur le projet sans pour autant faire partie du service.

6.2.2 Prototypage

Le prototypage est une étape importante lors de la création d'une interface utilisateur. Bien que réalisé, ce prototype présentait principalement les aspects visuels de l'application. Il aurait été plus judicieux de présenter principalement un« story-board » aux utilisateurs afin de revoir dès le départ le cheminement qu'ils souhaitent pour obtenir les informations. En effet, des demandes de changement ont été effectuées dans le projet, mais ceci une fois l'application quasiment fonctionnelle.

6.3 Recommandations futures

6.3.1 Travailler avec GIT

Pour la suite du projet, il serait intéressant de travailler avec GIT. En effet, cela permettrait de pouvoir créer des branches distinctes et ainsi travailler sur différentes tâches de façon indépendante.

6.3.2 Refonte entière du back-end :

Comme présenté dans le présent document, il est important qu'une refonte du « back-end » soit apportée sur le projet. En effet, il ne permet pas actuellement une modularité suffisante pour s'adapter aux nouvelles attentes du CRCHUM, ni pour rajouter de nouvelles fonctionnalités.

6.3.3 Réécriture des « Users Requirements » avec le client :

Suite à une présentation du prototype lors d'une réunion, en est sortie des changements sur le flux d'étapes de l'application. En effet, certaines requêtes comme l'« *area sélection* » doivent pouvoir se faire de façon totalement indépendante. Il serait en effet intéressant de créer une indépendance des différents modules afin de pouvoir s'adapter à des changements plus facilement.

CONCLUSION

Ce projet consistait au développement de l'interface utilisateur sur le projet GOAT. L'interface doit être réactive et claire afin de permettre aux utilisateurs qui sont des chercheurs en génomique d'obtenir des résultats interactifs et pertinents.

L'outil réalisé n'est pas encore complet, il doit en effet être revu dans son ensemble et particulièrement le « back-end » afin d'améliorer ses performances et reconstruire son architecture. Pour cela une rétro-ingénierie du « back-end » a été effectuée afin de préparer la recomposition. Un prototype est néanmoins fonctionnel sur le serveur de CRCHUM, mais doit par la suite évoluer selon les besoins.

Ce projet m'a permis de rencontrer des problématiques au quelle je n'avais jamais était confronté. En effet, je ne suis jamais arrivé sur un projet déjà en grande partie réalisé, et ceci implique de devoir fortement s'adapter au code et de le comprendre. De plus, travailler pour un domaine qui est très spécifique implique également de devoir s'accommoder à un langage technique complexe qui peut être une barrière lors de réunion.

.

ANNEXE I

Document de vision GOAT

GOAT

Genetic Output Analysis Tool

Version : 1.0
Release date : May 3rd 2015
Revision date : September 30th 2015

VISION DOCUMENT

Authors (software developers):

Dr. Alain APRIL – ÉTS Montréal, Canada

CRCHUM

Beatriz Kanzki – ÉTS

David Lauzon – ÉTS

Cédric Urvoy – ÉTS

Supervisor :

Dr. Alain APRIL – ÉTS

Collaborators (users expressing their need)

Presented to : Pavel Hamet – CRCHUM

Michael Phillips -

approved on ___/___/___ approved on ___/___/___

Revision History

Date	Version	Description	Author
2015-05-02	0.1	First version	Beatriz Kanzki
2015-05-02	0.2	Review of version 0.1	Dr. Alain April
2015-05-02	0.3	Integration of David's comments	Beatriz Kanzki
2015-05-03	0.4	Integration of Alain's comments	Beatriz Kanzki
2015-05-03	0.5	Review of applicable sections and general document structure	David Lauzon
2015-05-05	0.6	Add responsibilities of users and put products available at competition	Beatriz Kanzki
2015-05-06	0.7	Added product overview diagram. Modified constraints and quality criteria (moved from B08). Moved B07 to FEA05. Moved MetaboAnalyst screenshots to Appendix B. Changed license.	David Lauzon
2015-05-06	0.8	Added Dre Tremblay's expectations and	Beatriz Kanzki

		responsibilities	
2015-05-07	0.9	Added description and example of product expected behavior, workflow and user requirement for tool Appendix B,C, D	Beatriz Kanzki
2015-05-10	0.10	Updated product perspective, expanded user needs to 28 features.	David Lauzon
2015-05-14	0.11	Clarified sections 3.3, 4.1, 4.2, 5 + removed appendix B - D	David Lauzon
2015-05-15	0.12	Added feature definitions and URL's, and figure 1 of for scope of the project	Beatriz kanzki
2015-05-15	0.13	Reviewed format, licensing text	Alain April
2015-05-15	0.14	Added process diagrams	Beatriz Kanzki
2015-05-16	0.15	Reviewed english, scope text, requirements formulation, table number sequences and open comments	Alain April
2015-05-20	0.16	Reviewed Scope Figure, inserted potential UI examples, reviewed Gene Query process figure and simplify product features, updated feature attributes with state and effort values, assigned an open source creative commons license	Alain April Beatriz Kanzki David Lauzon
2015-05-25	0.16.1	Changed the scope figure and other minor	David Lauzon

		display issues (table 7 and added appendix B)	
2015-05-27	0.16.2	Added SigmaPlot in competition	Beatriz Kanzki
2015-05-28	0.16.3	Workflow process image changed	Beatriz Kanzki
2015-09-20	1.0	final review	Alain April

Summary

3.4.7 USC Genome viewer

3.4.8 IGV

4. Product overview

4.1 Product perspective

4.2 Summary of major benefits and features

4.3 Assumptions and dependencies

4.4 Cost and Price

4.5 Licensing and installation

5. Product features

FEA01 – Import data from SNPTest

FEA02 – Filter SNPs by access key

FEA03 – Contextual gene information from: GeneCards

FEA04 – Contextual gene information from: NHGRI-EBI GWAS Catalog

FEA05 – Contextual gene information from: ENcode

FEA06 – Contextual gene information from: Epigenomic catalog

FEA07 – Contextual gene information from: FDR

FEA08 – Log Book (provenance/traceability)

FEA09 – Export charts and tables

FEA10 – Visualization: GWAS Manhattan Plot

FEA11 – Visualization: BoxPlot

FEA12 – Interactive visualization: Genome Viewer

FEA13 – Interactive visualization: Region Selection

FEA14 – Interactive visualization: Modify threshold

Table list

Figures list

1. Introduction

1.1 Objective

The object of this document is to determine the project for the visualization and analysis tool for genomic datasets (coined GOAT) in order to have a common vision between the development team and the various project stakeholders. The figure 1 describes the whole research workflow. Notice that the scope of this document, GOAT, is restricted to the green section of the workflow (in the middle of the figure.

1.2 Scope

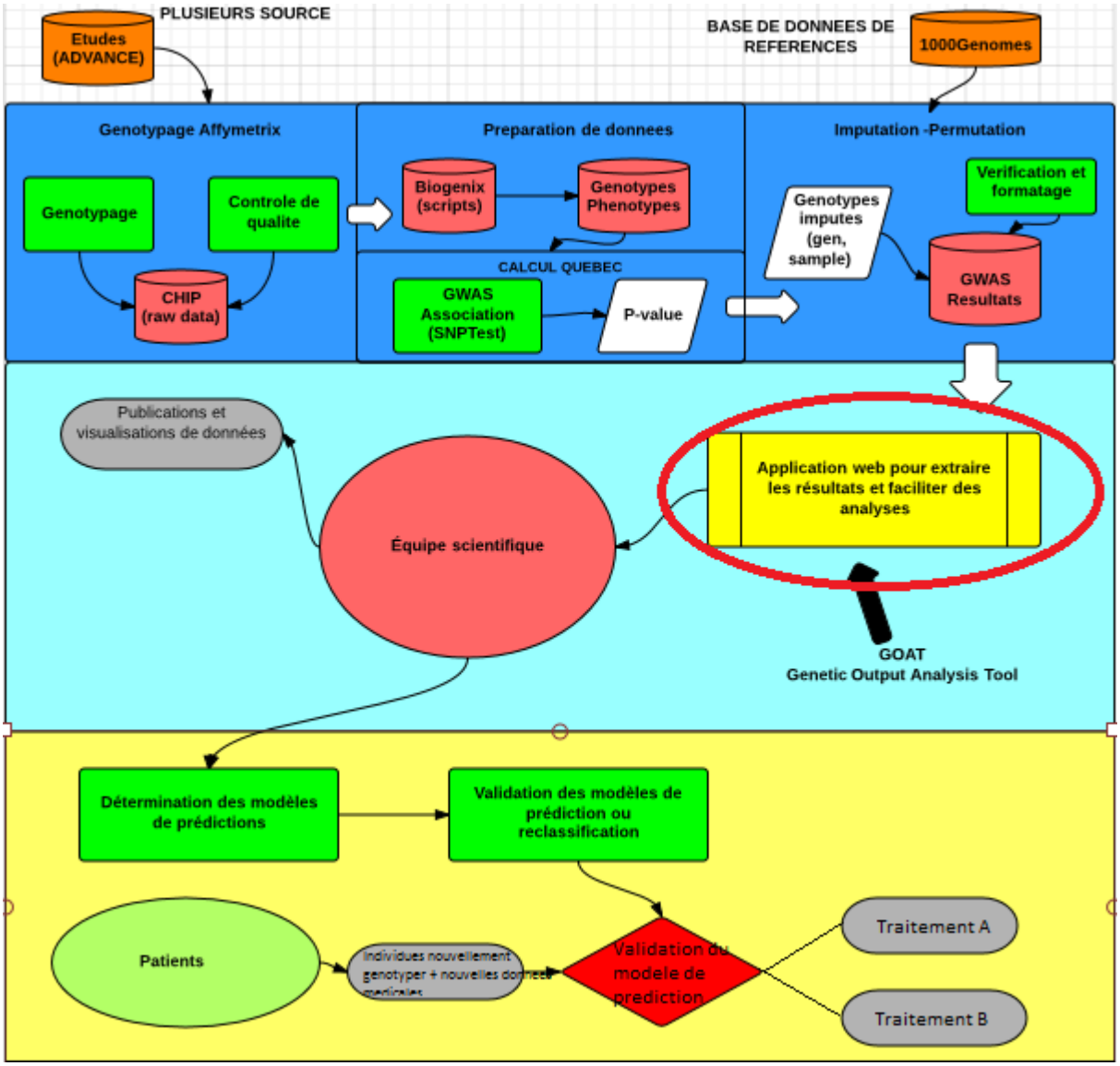


Figure 1 shows the whole process of the discovery workflow. **As indicated by the big black arrow and red circle, the scope of this Vision document is limited by the green zone**.

3 phases of software development are planned for GOAT:

1. Access and retrieval of data from the database
   - GWAS results
   - Intersection between GWAS and other resources.
   - Downloadable tables.
2. Visualization, analysis, phenotype queries.
   - Graph (interactive Manhattan)
   - BoxPlot (ratio case/control)
   - Gene Annotation
3. Post-processing analysis
   - Reanalyze processed data live
   - Create new preformatted
   - Boxplot of quantitative data compared to qualitative data
   - Statistics on the phenotype (comparison table)

Note that GOAT will be an independent open source software and can run on any genotyped dataset. The project does not require and is not dependent on the CRCHUM database content or its internal structure

1.3 Definitions, acronyms and abbreviations

Terms	Definition
ETL	Extract Transform Load. Process used Extract data from a source, Transform it, and Load into a destination database.
GWAS	Genome wide association study is an examination of many common genetic variants in different individuals to see if any variant is associated with a trait.
PI	Principal investigator
CAU	Caucasian population
AS	Asian population
AA	African American population
rs ID	Accession number for a SNP
SNP	Single nucleotide polymorphism is a DNA variation sequence
PubMed	Free full-text archive of biomedical and life sciences journal literature at the NIH

NIH	National Institutes of Health
1000 Genome	Catalog of human genetic variations
Genecards	Search, integrated database of human genes that gives concise genomic information, on all known predicted human genes.
ClinVar	Clinical variations is a freely accessible public archive of reports of the relationships among human variations and phenotypes hosted by NCBI
NCBI	National center for biotechnology information gives access to biomedical and genomic information.
SNP-TEST	Program for analysis of SNP association in genome-wide studies.
CFR21 part 11	Code of Federal Regulations Title 21: electronic records - electronic signatures

Table 1: Definitions, acronyms and abbreviations

6.4 1.4 Summary description of product

The final product is a web site based tool allowing the researcher to query its own database in order to extract info on phenotypes, genetic datasets, or biomarkers, visualize them in order to get a file that could be used for further statistical analysis, without having to go through the bioinformaticians office. This application would allow users to locate the information, interactively and visualize it, and get a table of significant results where each gene would be a hyperlink towards external sites for further information.

6.5 1.5 References

http://www.metaboanalyst.ca/faces/ModuleView.xhtml

Glossary : to come in a separate document

2. Positioning

2.1 Business opportunity

This software could be in demand by health researchers.

2.2 Problem definition

Problem	- Health researchers rely on bioinformatics specialists for data extraction from the database and for their visualization.

	- The current softwares are not efficient, or user friendly for their needs.
Affects	● Bioinformatics specialists job who cannot cope with all demands coming from stakeholders; ● And stakeholders turnaround time to analyze data
Impact	Generates a waiting list and a dependency on the bioinformatics specialist availability.
A good solution	A good solution would allow a researcher to locate the information that needs to be analyzed, from a database, conduct statistical analysis and visualize the resulting graphs, interactively (i.e. meaning a drill down facility) and allowing to obtain identification of gene using external best practice information interactively for further study. The researcher could save an ongoing study to continue the research another time.

Table 2: Problem Definition

2.3 Product positioning

For	Health research Labs that want to analyze genetic data
Who wants to	Improve information access from their internal databases, conduct statistical analysis and visualization seamlessly.
GOAT	is a web-based software as a service (SaaS) for bioinformatics genomic and genetic research.
Which allows	To display an interactive graph, varying depending on the information required, referring to significant genes that can be selected. To save the work session in order to access it later.

Unlike	-Having to call on a bioinformatics specialist to obtain the information. -Using existing open source tools that are currently available and do not offer the functionality required.
The product	Needs a professional and user-friendly interface that improves information retrieval efficiency and offers a modern visualization of genomic data. The product must also allow to save the user's session and work for furthering analysis.

Table 3: Product positioning

3. Description of stakeholders and users

3.1 Summary of stakeholders and users.

Name	Occupation
Dr Pavel Hamet	Presents his requirements : Principal investigator
Dre Tremblay	Presents her requirements
Michael Philips	User representative : Director
Francois Harvey	Bioinformatician Specialist
Francois Marois	Bioinformatician Specialist
Gilles Godefroid	Bioinformatician Specialist
Carole Long	Biologist
John Raelson	Geneticist
Mounsif Haloui	Biologist
Ramzan Tahir	Biostatistician
Paul Simon	Doctoral student, Observer
Alain April	Developers leader: Sftwr Project Supervisor (ÉTS)

Table 4: Stakeholders and users of GOAT

3.2 User and stakeholder profiles

3.2.1 Health Researcher Profile (Dr. Hamet)

Responsabilities	*Presents high level requirements and use tool to prepare conference and show data by doing analysis and extracting data. Add gene annotation on DNA region.*
Success Criteria	*Data extraction is successful and can download graphs* *Gene annotation recorded in database*
Deliverables	*Downloadable tables and graphs*
Comments / Issues	*Has to access database from inside the CHUM, Query in tool available now are slow, not user friendly and takes a lot of steps on different screens before getting the actual result.*

3.2.2 Health Researcher Profile (Dre. Tremblay)

Responsabilities	*Use tool to prepare conference and do search queries for genes, phenotype.* *Compare two phenotypes for a SNP, and then compare to GWAS catalog.*
Success Criteria	*Name of dataset where interesting results are is identified* *Whole SNPTest output is present* *Complete GWAS output with possibility to zoom* *Link to other interesting SNPs can be clicked easily after review of results in table.*
Deliverables	*Downloadable tables and graphs to use in powerpoint or reanalysis.*
Comments / Issues	*Has to access database from inside the CHUM, Has to access database from inside the CHUM, Query in tool available now are slow, not user friendly and takes a lot of*

	steps on different screens before getting the actual result.

3.2.3 Bioinformatician Profile (F. Harvey, F. Marois, G. Godefroid)

Responsabilities	*Enrich database with information that will be available for tool*
Success Criteria	*Data has been recorded*
Deliverables	*N/A*
Comments / Issues	*N/A*

3.2.4 ÉTS Stakeholder Profile

Responsabilities	● Development of the open source software ● Maintenance and long-term evolution of the software
Success Criteria	● Health researchers are more productive with the new product than with current solutions. ● New feature request can be implemented with minimal effort
Deliverables	● An open source software that can be implemented in the lab
Comments / Issues	● Frequent feedback from the CRCHUM's team, and the senior management board, is required to ensure success of the project.

3.2.5 Research Student

Responsabilities	*Extract data from database and analyse them* *Generate results, and make quality control* *Generate graphs*

Success Criteria	*Data extraction successful* *Analysis can be done by tool* *Graphs and tables are downloadable*
Deliverables	*Downloadable table and graphs*
Comments / Issues	*N/A*

3.2.6 Biostatician

Responsabilities	*Extract data from database and analyse them* *Quality control (verify distribution of data)*
Success Criteria	*Data extraction successful* *Quality control of data successful with external tool* *Graphs and tables are downloadable*
Deliverables	*Downloadable table and graphs*
Comments / Issues	*N/A*

3.2.7 Geneticist

Responsabilities	*Extract data from database and analyse them* *Generate results, and make quality control* *Publish results and graph*
Success Criteria	*Data extraction successful* *External quality control tests passed by data* *Graphs and tables are downloadable*

Deliverables	*Downloadable table and graphs*
Comments / Issues	*N/A*

3.2.8 Biologists

Responsabilities	*Extract data from database and analyse them* *Generate results, and make quality control* *Generate graphs* *Locate publications according to genes*
Success Criteria	*Data extraction successful* *Analysis can be done by tool* *Graphs and tables are downloadable* *All available resources on internet appear on the tool.*
Deliverables	*Downloadable table and graphs*
Comments / Issues	*N/A*

3.3 Major needs/user requirements of all stakeholders and users

Needs	Concerns	Current Solution / Tool	Proposed Solution
B01 – Query by Gene	Researcher claim that some information is unavailable on the existing tool. When investigating with the bioinformatics	Information is difficult to find, non-existent, or lost with current software or it has display/query problems.	Modernize/review the query function UI for ease of use. The researcher should be able to filter the list of markers, from the database, using the most popular access keys just like popular open source catalogs.

	specialists they show that it is available.		
B02 – Data visualization	- Today, resulting graphs have to be generated manually; - Users would like to delete or select significant data on visualization before saving session;	Current software only displays a table with results with links to external web sites.	Develop a new visualization function that can display interactive graphs: 1) of different types based on the type of data extracted; 2) that have data table(s) with integrated hyperlinks pointing to external websites and other sources (see B05).
B03 – Graph Export	- Users would like to use the graphs readily for publication.	Good tools exist to produce interactive graphs, but these are not readily acceptable for publications	The researcher will be able to export (i.e. download or copy/paste) a graph being visualized, as an image, in a publication ready format
B04 – Query recorder	Today, queries using the existing software cannot be saved and reused	This is not available today.	Develop a functionality to save current session work, using the user ID and description of the current analysis at any point. This session can be reopened and continued.
B05 – Contextual Gene Information	Obtain the up-to-date information for a gene from a reputed external source	Current tools available do not display all the updated publications available online or returns that they are unavailable	GOAT will provide information, from reputed online databases, in regards to the current visualization context. For example, if visualizing a gene, the researcher will be able to easily find more information on the engine.

Table 5: Major needs/user requirements for GOAT

3.4 Alternatives and competition

3.4.1 MetaboAnalyst, (*open source*)

Serves as a visualization and analysis tool but dedicated exclusively to metabolites. The client would like a user interface similar to MetaboAnalyst. Refer to **Appendix B** for MetaboAnalyst screenshots.

http://www.metaboanalyst.ca/

6.5.1 3.4.2 LocusZoom (*open source*)

Used to plot regional association results from genome-wide association scans or candidate gene studies but have to know locus region from data of the researcher prior to accessing it on this site in order to visualize it.

http://locuszoom.sph.umich.edu/locuszoom/

6.5.2 3.4.3 SNPTest (*open source*)

Program for the analysis of single SNP association in GWAS studies, but does not provide visualization tools.

https://mathgen.stats.ox.ac.uk/genetics_software/snptest/old/snptest.html

6.5.3 3.4.4 GWAS Diagram Browser (*open source*)

Pipeline is dedicated to GWAS studies but doesn't treat SNPTest file outputs.

http://www.ebi.ac.uk/fgpt/gwas/

6.5.4 3.4.5 HAPGEN (*open source*)

Program to simulate case control datasets at linked SNP markers conditional upon a set of known haplotypes.

https://mathgen.stats.ox.ac.uk/genetics_software/hapgen/hapgen2.html

6.5.5 3.4.6 Biopython packages (*open source*)

Demands programming knowledge in that specific language and for that package.

http://biopython.org/

6.5.6 3.4.7 UCSC genome browser (*proprietary closed source*)

Contains the reference sequence and working draft assemblies for a large collection of genomes. It is a very large software suite with many parameters; and only 1% of it is needed by the geneticist.

https://genome.ucsc.edu/cgi-bin/hgGateway

6.5.7 3.4.8 IGV

The **Integrative Genomics Viewer (IGV)** is a high-performance visualization tool for interactive exploration of large, integrated genomic datasets. It supports a wide variety of data types, including array-based and next-generation sequence data, and genomic annotations. Great tool to visualize region surrounding SNPs, but has to be access from external links with region of interest known prior to query.

https://www.broadinstitute.org/igv/

6.5.8 3.4.8 Sigmaplot (*proprietary closed source*)

SigmaPlot is **a software** product that helps researchers and engineers analyze their data, create precise plots and charts, develop publication-quality graphs and customize all analysis needs.

http://www.sigmaplot.com/products/sigmaplot/sigmaplot-details.php

4. Product overview

4.1 Product perspective

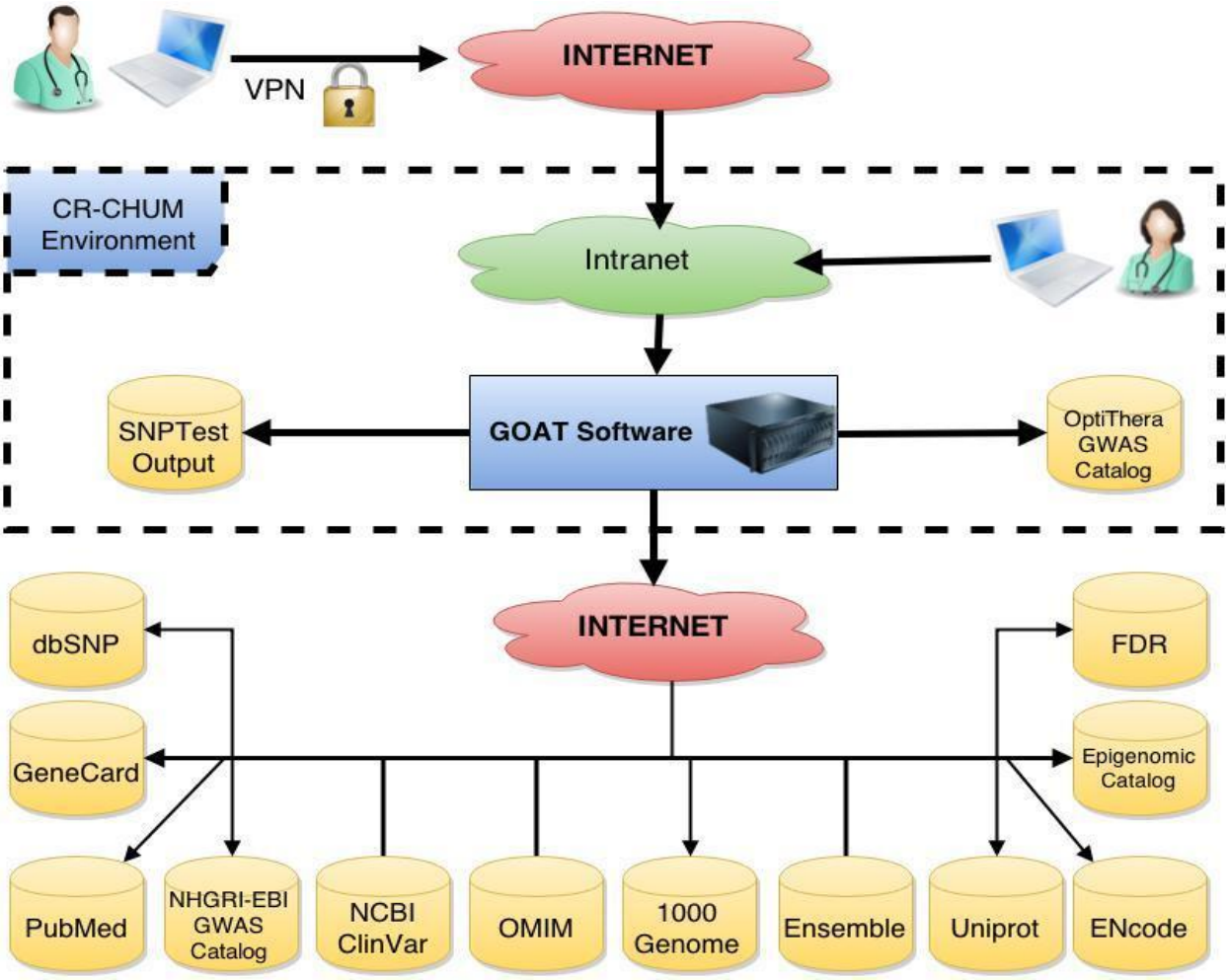
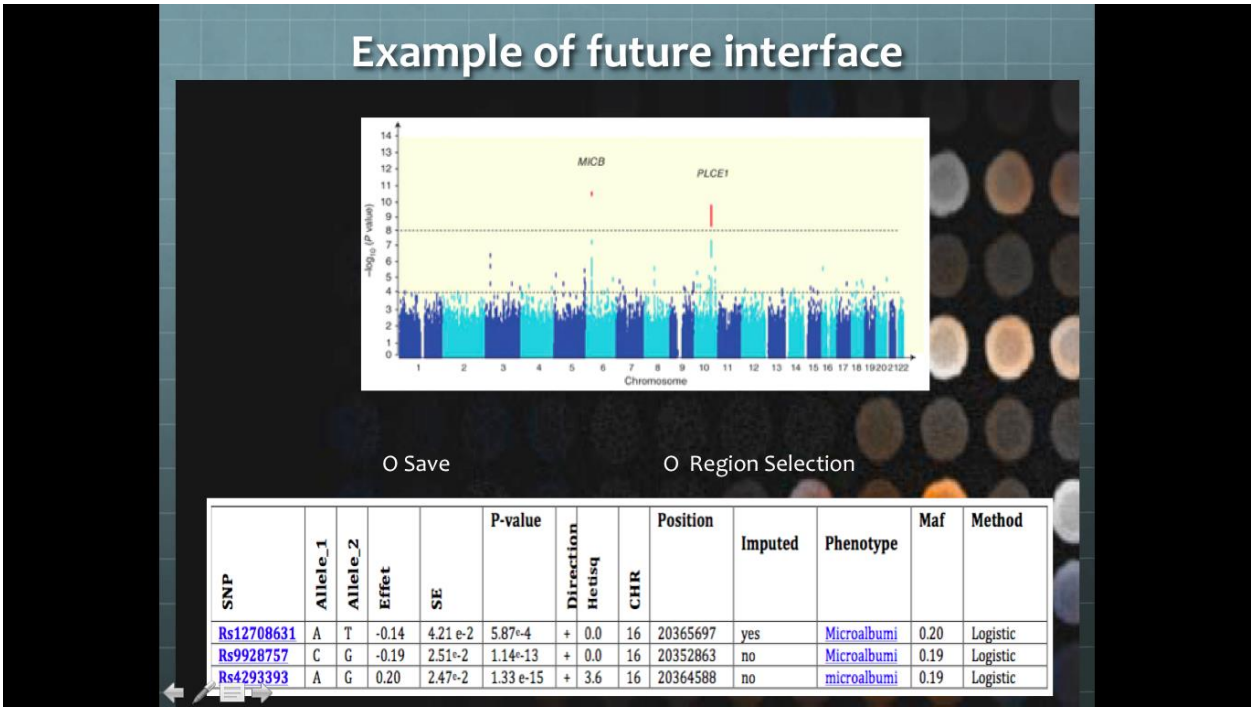


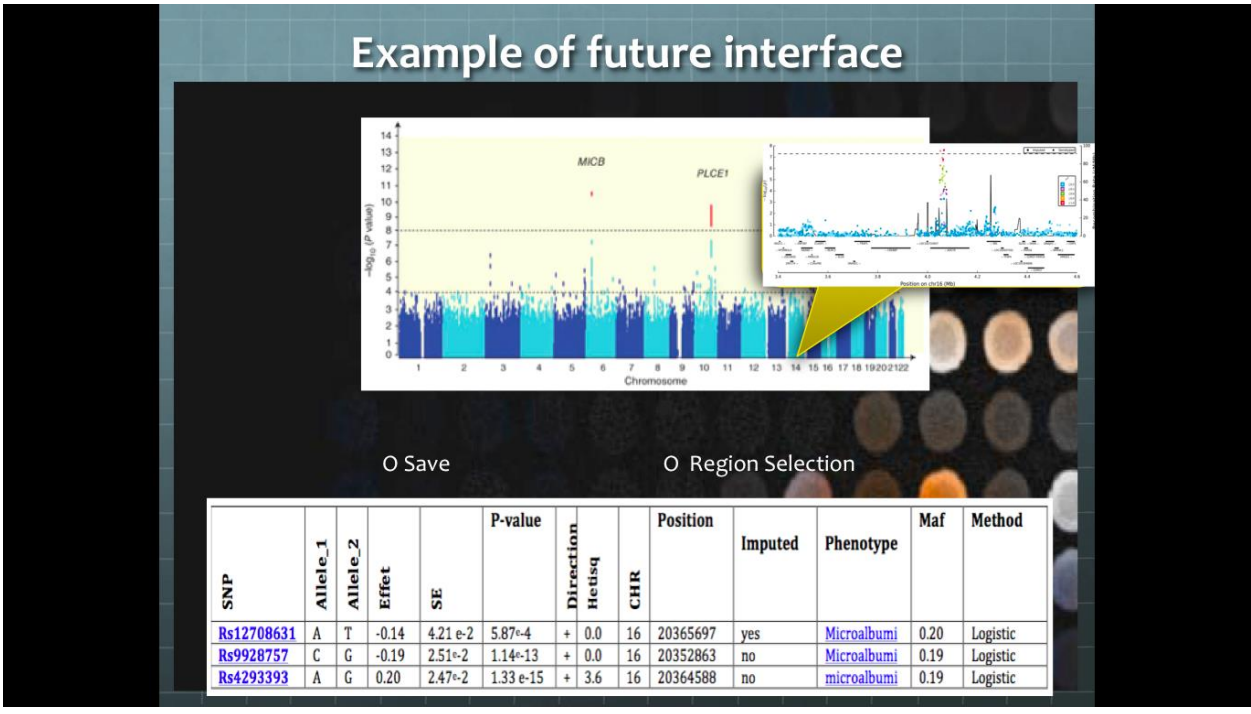
Figure 2. GOAT user environment and related open source projects involved

Ideas of User Interfaces to be developed



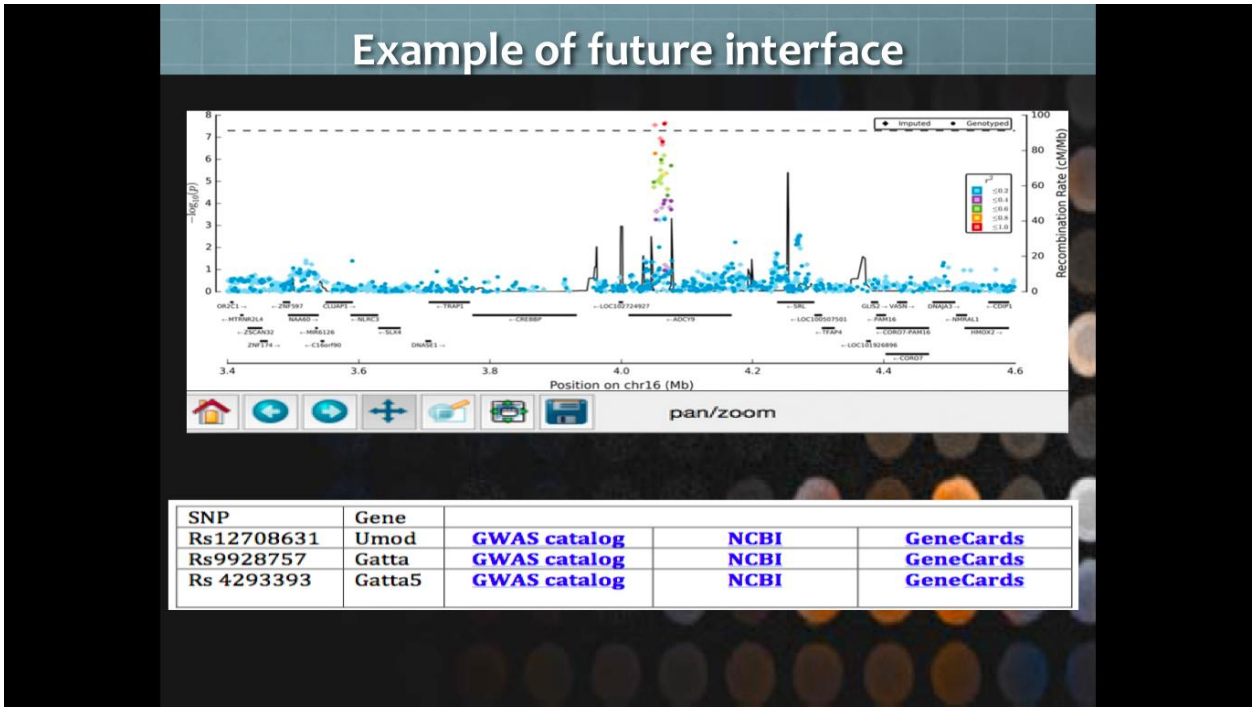
SNP	Allele_1	Allele_2	Effet	SE	P-value	Direction	Hetisq	CHR	Position	Imputed	Phenotype	Maf	Method
Rs12708631	A	T	-0.14	4.21 e-2	5.87e-4	+	0.0	16	20365697	yes	Microalbumi	0.20	Logistic
Rs9928757	C	G	-0.19	2.51e-2	1.14e-13	+	0.0	16	20352863	no	Microalbumi	0.19	Logistic
Rs4293393	A	G	0.20	2.47e-2	1.33 e-15	+	3.6	16	20364588	no	microalbumi	0.19	Logistic

Figure 2. First page to appear after gene Query



SNP	Allele_1	Allele_2	Effet	SE	P-value	Direction	Hetisq	CHR	Position	Imputed	Phenotype	Maf	Method
Rs12708631	A	T	-0.14	4.21 e-2	5.87e-4	+	0.0	16	20365697	yes	Microalbumi	0.20	Logistic
Rs9928757	C	G	-0.19	2.51e-2	1.14e-13	+	0.0	16	20352863	no	Microalbumi	0.19	Logistic
Rs4293393	A	G	0.20	2.47e-2	1.33 e-15	+	3.6	16	20364588	no	microalbumi	0.19	Logistic

Figure 3. Region selection generates a popup.



6.6 **Figure 4. Information of region selection, and gene information from external links**

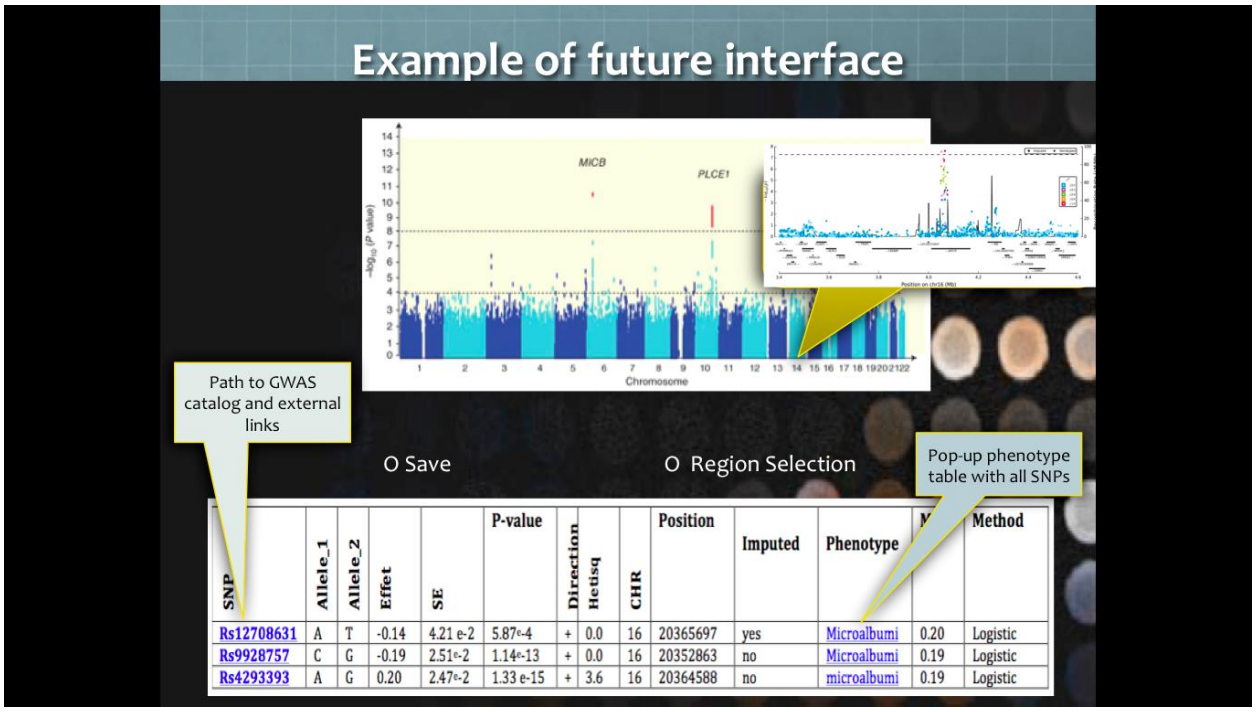


Figure 5. Overall view of GOAT gene query interface.

4.2 Assumptions and dependencies

- GOAT will collect external database information for which the rights are free or can be acquired.
- CRCHUM will provide test data to the development team in order to facilitate development.

4.3 Cost and Price

This software will be developed free of charge by the members of the project team: Dr A. April (professor at ÉTS), D. Lauzon (a PhD student in software Engineering at ÉTS). B.Kanzki is the principal developer and C.Urvoy will improve the UI of the prototype developed by B.Kanzki.

4.4 Licensing

In this document the ÉTS/UdeM team have translated/enriched user requirements (expressed by users) into more detail in order to capture the overall long term product requirement of GOAT. The document refers to public information about the following open-source projects: MetaboAnalyst, LocusZoom, SNPTest, GWAS Diagram Browser, HapGen, BioPython, IGV and many other open source projects. Before GOAT uses any of the information of these sources, their individual licences will have to be investigated. Only open-source and publicly available material will be integrated to GOAT.

GOAT Software License: The resulting GOAT software will be licensed: **General Public License: Gpl v3 or any later versions**

Installation: The open source software will be available publicly. Installation of the software can easily be made by CRCHUM bioinformatics specialists at Dr. Hamet CRCHUM Lab.

GOAT Vision Document License: The resulting GOAT vision document is licensed: Creative commons - **attribution-shareAlike 4.0 International license**

5. Product features

This section highlights the key features of this new software product. Figure 3 presents the GOAT overall process.

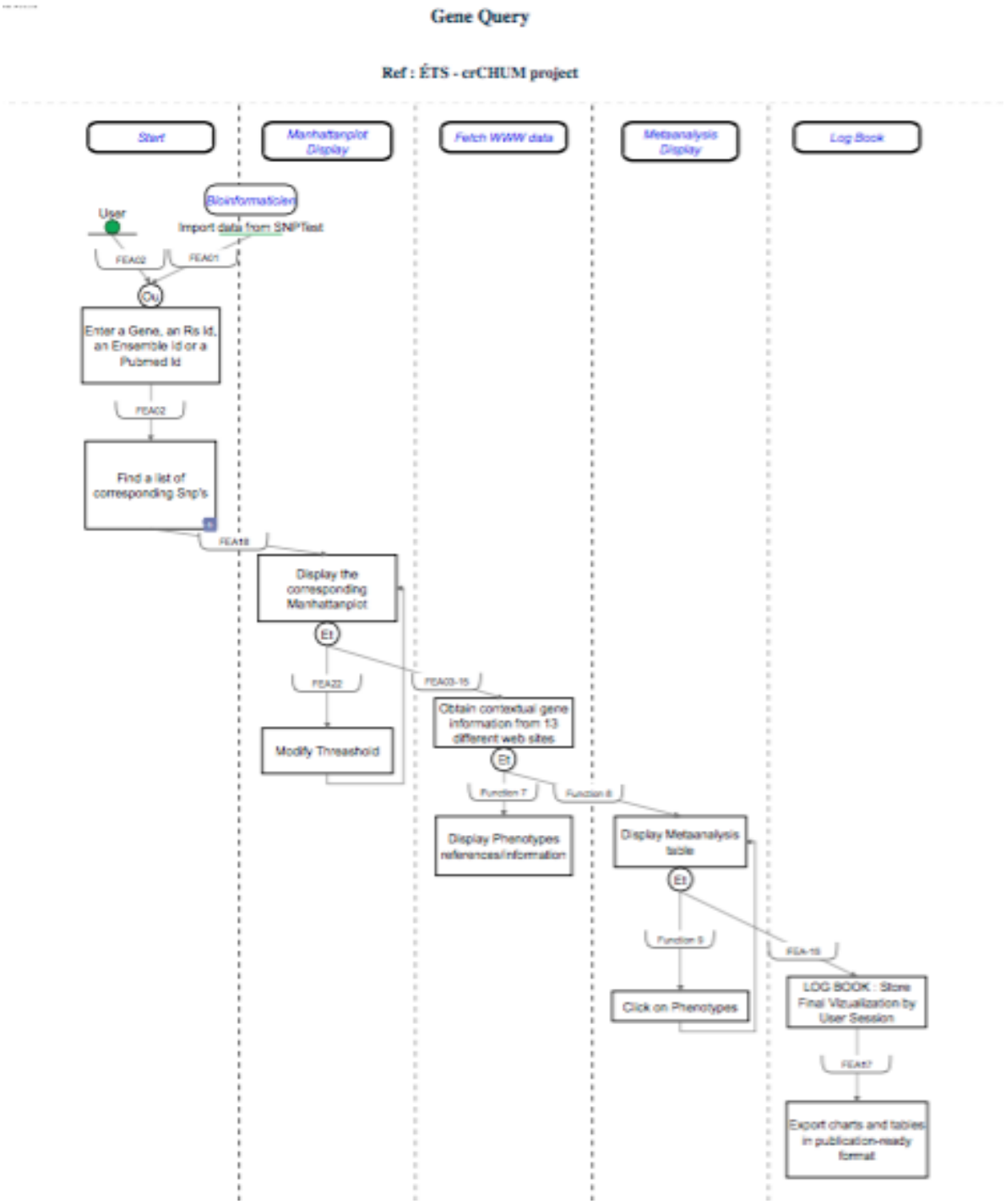


Figure 6: GOAT process diagram

6.7 FEA01 – Import data from SNPTest

The bio-informaticians would like a way to import data into the GOAT's database. The file format is an output from SNPTest that is currently stored in a MySQL database. Therefore a functionality is required to load this data from the current sql tables of the existing database. GOAT also needs to be able to support custom columns from SNPtest.

6.8 FEA02 – Filter SNPs by access key

Users would like to be able to find the list of SNPs associated with a gene using one of the following four access keys:

- Gene name
- RS ID
- Ensemble ID
- Pubmed ID

Each of these access keys can select one or more SNPs. A minimum of 2 SNPs are required to be able to visualize the data.

6.9 FEA03 – Contextual gene information from: GeneCards

When visualizing data, the users would like to have an hyperlink to the *GeneCards* website directly to gene page information. This means that gene information would be a parameter automatically passed to the corresponding website (when this functionality is allowed) to contextualize the link precisely on the information required.

GeneCards is a searchable, integrated database of human genes that provides comprehensive, updated, and user-friendly information on all known and predicted human genes. It includes Ensemble, Uniprot, OMIM, NCBI, ClinVar, Pubmed, dbSNP. GeneCards extracts and integrates gene-related data, including genomic, transcriptomic, proteomic, genetic, clinical, and functional information. This is automatically mined from >100 carefully selected web sources, thereby allowing one-stop access to a very broad information base.

GeneCards overcomes barriers of data format and heterogeneity, and uses standard nomenclature and approved gene symbols. It presents a rich subset of data for each gene, and provides deep links to the original sources for further scrutiny. GeneCards is widely used, and assists in the understanding of gene-related aspects of biology and medicine. http://www.genecards.org/

dbSNP is a free public archive for genetic variation within and across different species developed and hosted by NCBI. http://www.ncbi.nlm.nih.gov/projects/SNP/

PubMed comprises more than 24 million citations for biomedical literature from MEDLINE, life science journals, and online books. Citations may include links to full-text content from PubMed Central and publisher web sites. http://www.pubmed.org/

ClinVaraggregates information about genomic variation and its relationship to human health. http://www.ncbi.nlm.nih.gov/clinvar/

OMIM (*Online Mendelian Inheritance in Man*) is a comprehensive, authoritative compendium of human genes and genetic phenotypes that is freely available and updated daily. OMIM is authored and edited at the McKusick-Nathans Institute of Genetic Medicine, Johns Hopkins University School of Medicine, under the direction of Dr. Ada Hamosh. http://www.ncbi.nlm.nih.gov/omim or http://www.omim.org/

1000 Genomes Project finds most genetic variants that have frequencies of at least 1% in the populations studied. The team wants to see allele frequencies per population. http://www.ncbi.nlm.nih.gov/variation/tools/1000genomes/

The Ensembl project produces genome databases for vertebrates and other eukaryotic species, and makes this information freely available online. http://useast.ensembl.org/index.html?redirect=no

The mission of **UniProt** is to provide the scientific community with a comprehensive, high-quality and freely accessible resource of protein sequence and functional information.
http://www.uniprot.org/

6.10 FEA04 – Contextual gene information from: NHGRI-EBI GWAS Catalog

When visualizing data, the users would like to have an hyperlink to the *NHGRI GWAS Catalog* website directly to gene page information.

GWAS Catalog is a quality controlled, manually curated, literature-driven collection of all published genome-wide association studies, produced by a collaboration between EMBL-EBI and NHGRI. These GWAS studies assays at least 100,000 SNPs and all SNP-trait associations have p-values < 1.0×10^{-5}. https://www.ebi.ac.uk/gwas/

6.11 FEA05 – Contextual gene information from: ENcode

When visualizing data, the users would like to have an hyperlink to the *ENcode* website directly to gene page information.

The Encyclopedia of DNA Elements (**ENCODE**) Consortium is an international collaboration of research groups funded by the National Human Genome Research Institute (NHGRI). The goal of ENCODE is to build a comprehensive parts list of functional elements in the human genome, including elements that act at the protein and RNA levels, and regulatory elements that control cells and circumstances in which a gene is active.
http://genome.ucsc.edu/cgi-bin/hgTracks?db=hg18&position=chrX%3A151073054-151383976&hgsid=426788787_i7aYCjc4DgZSO4MS6dmiTzZlAmma

6.12 FEA06 – Contextual gene information from: Epigenomic catalog

When this project is operation, the users would like to have an hyperlink to the *Epigenomic* website directly to gene page information (*this project is not yet operational at this time*).

Human epigenomic catalog (HEP) aims to identify, catalogue and interpret genome wide DNA methylation patterns of all human genes in all major tissues.
http://www.epigenome.org/?page=project

6.13 FEA07 – Contextual calculations of gene information from: FDR

Users would like to have to see, in the presented table of information, the FDR calculations done for the information visualized.

False discovery rate (FDR) is is the expected proportion of Type I errors among the rejected hypotheses FDR = E(V/R | R>0)P(R>0). In many cases (particularly in genomics) we can live with a certain number of false positives. In these cases, the more relevant quantity to control is the false discovery rate (FDR). False discovery rate (FDR) is designed to control the proportion of false positives among the set of rejected hypotheses.

6.14 FEA08– Log Book (provenance/traceability)

Users would like to have provenance/traceability of the discovery actions.

Every user actions will be recorded in a “Log Book” database (refer to CFR21 part 11 requirements). This will allow the user to easily find, select and replay any previously saved exploration sessions including visualizations. This functionality would be accessible by user sessions.

6.15 FEA09 – Export charts and tables.

Users would like to allow visualized Charts to be exported in pdf format and visualized tables to be exportable to .csv format. This action should be simple, resulting figures should be of high quality.

6.16 FEA10 – Visualization: GWAS Manhattan Plot

User would like to see this kind of plot which will give them an idea of which SNPs are the most interesting, by chromosome.

A **Manhattan plot** is a type of scatter plot, usually used to display data with a large number of data-points - many of non-zero amplitude, and with a distribution of higher-magnitude values. In GWAS Manhattan plots, genomic coordinates are displayed along the X-axis, with the negative logarithm of the association *P*-value for each single nucleotide polymorphism (SNP) displayed on the Y-axis, meaning that each dot on the Manhattan plot signifies a SNP. Because the strongest associations have the smallest *P*-values (e.g., 10^{-15}), their negative logarithms will be the greatest (e.g., 15).

6.17 FEA11 – Visualization: Box Plot

In some cases, user would like to see a Box-plot representation of some data in order to see the distribution of the data.

Box-Plot is n descriptive statistics, a **box plot** or **boxplot** is a convenient way of graphically depicting groups of numerical data through their quartiles. Box plots may also have lines extending vertically from the boxes (*whiskers*) indicating variability outside the upper and lower quartiles, hence the terms **box-and-whisker plot** and **box-and-whisker diagram**. Outliers may be plotted as individual points. This is also called a "box and whisker plot".

6.18 FEA12 – Interactive visualization: Genome Viewer

Users would like to be able to interact with a graph currently visualised. The interaction would be similar to a biopython based graph with selected region feature, gene name, and other information to be assessed in the SRS stage.

6.19 FEA13 – Interactive visualization: Region Selection

User would like to be able to determine gene region visualization, by selecting an interval underlying before and after the SNP of interest. Could be 1000 bases before and 1000 after.

6.20 FEA14 – Interactive visualization: Modify threshold

Users would like to have a system set threshold default default that can be changed by the user interactively. The user would be able to input the desired threshold that he wishes to apply on the dataset and changes would appear on the manhattan plot interactively.

6.21 FEA15 – LD Regression Score

Users would like GOAT to Computes LD (Linkage Disequilibrium) regression score for a region of interest in the graph.

LD Score regression, that quantifies the contribution of each by examining the relationship between test statistics and linkage disequilibrium (LD). The LD Score regression intercept can be used to estimate a more powerful and accurate correction factor than genomic control.

6.22 FEA16 – MAF per population (from 1000 Genome Project)

Users would like GOAT to display the MAF per population for region of interest.

Allele Frequency. Population: AA, AS, CAU.
This information would be obtained directly from the 1000 Genome Project.

6. Constraints

The first version/prototype of GOAT needs should be delivered within 40 days of the acceptance of this vision document (a usable proof of concept).

7. Quality criteria (non-functional requirements)

CN01 – Data must be secured: During SRS security requirements must be explicitly defined

CN02 – Maintainability: GOAT should use Software Engineering best practices to allow a design pattern approach for easier maintainability.

CN03 – Usability. The user interface must be easy to use

8. Features attributes

This section summarizes the features of the system according to the benefits they bring to the customer, the effort required to implement the risk associated with their implementation and their stability (probability of change). Each value of these attributes are detailed in Appendix A.

Features	State	Benefits	Effort	Risk
FEA01 – Generic interface from external data sources (import data from SnpTest)	Partial		Low	just import a .csv file for now
FEA02 – Filter SNPs by access key	Approved		Low	
FEA03 – Contextual gene information from GeneCards	Approved		high	licences, availability of ap
FEA04 – Contextual gene information from: NHGRI-EBI GWAS Catalog	Approved		low	licences, availability of APii
FEA05 – Contextual gene information from: ENcode	Approved		low	licences, availability of APi
FEA13 – Contextual gene information	Delayed		low	project not available

from: Epigenomic catalog				
FEA07 – Contextual calculations of gene information from FDR	Approved		low	formula details
FEA08 – Log Book (provenance/traceability)	Delayed		high	other project and vision document will be done
FEA09 – Export charts and tables	Approved		low	
FEA10 – Visualization: GWAS Manhattan Plot	Approved		medium	
FEA11 – Interactive visualization: Box Plot	Delayed		medium	
FEA12 – Interactive visualization: GWAS Manhattan Plot	Approved		medium	
FEA13 – Interactive visualization: Region Selection	Approved		low	
FEA14 – Interactive visualization: Modify threshold	Approved		low	
FEA15 – LD Regression Score calculations	Approved		low	
FEA16 – MAF per population (from 1000 Genome Project)	Approved		low	

Table 7: GOAT features state, benefit, effort, risk and stability

9. Other Product Requirements

9.1 Applicable standards

6.22.1 9.1.1 Internal

crCHUM representatives have not presented any specific internal standards requirements to be imposed on this software.

6.22.2 9.1.2 External

The CFR21 Part 11 requirements may be imposed on GOAT. This will be assessed in another vision document addressing provenance and another separate project leaded by D.Lauzon.

9.2 System requirements

9.2.1 Security

9.2.2 Accessibility: to be handled by Dr.Hamet informatics staff:

- GOAT should be available internally, at the crCHUM, by Dr Pavel Hamet's staff, but Dr Hamet will ask the CRCHUM to give him external accesses in order to access GOAT from outside the crCHUM.

9.2.3 Portability

- The client software will be accessible from any desktop workstation with the with the following web browser installed: Firefox version 35
- The server will be compatible with Ubuntu Linux 14.04 or later.

9.3 Performance requirements

GOAT should be faster the the existing tool used. SRS should assess design approaches that allow the following functions to be responsive. The following items requires good response time:

- FEA01-Extraction of data from the internal database.
- FEA10-Statistical analysis execution
- FEA11-Exploring a graph
- FEA3,4,5,6 and 16 - obtaining information from external sources
- FEA7, 16-computing FDR and LD
- FEA09-Exporting a graph or a table
- FEA8-LogBook capture

9.4 Environmental requirements.

A web server, in the CRCHUM, accessible to Dr Hamet's staff and having a VPN access for secure external access is planned.

10. Documentation requirements.

10.1 User's Manual

No user manual is planned for GOAT

10.2 Online help

No online help is planned for GOAT

6.23 10.3 Installation Guides, configuration, and README file

At the end of the project, the software will be installed, by the bioinformatics specialist, and will be operated a crCHUM web server for internal use only. Configuration and readme file will be developed for GOAT and made available publicly.

CHAPITRE 7APPENDIX

CHAPITRE 8Appendix A - Feature attributes.

This section defines the attributes we associate with different characteristics of the system.

State

Proposed	This status indicates that the feature is available and must be the subject of discussion within the project team for its acceptance or rejection.
Partial	Will be partially implemented in the current system
Approved	This status indicates that the feature has been selected for an upcoming development.
Incorporated	This status indicates that this feature has been incorporated into the system during development .
Delayed	Feature will be implemented in a future version

Benefits

Critical	This level indicates that the feature is essential for the software. This means that the customer will not want to have a system without this feature.
Important	This level indicates that this feature is important. However, if it is not implemented, the software can be used. It is up to the customer to decide whether to have the software without this feature or if the

	project stops.
Useful	The system will integrate all the important features to achieve the anticipated profit. However if time permits and if the project team is available, some features can be added to increase the customer's benefit.

Effort

High	This level indicates that the amount of effort required is at least two weeks of work.
Average	This level indicates that the amount of effort required is between one and two weeks of work.
Low	This level indicates that the amount of effort required is below a workweek.

Risk

High	This level indicates that there is a high level of uncertainty about the duration or cost of implementation, or even a risk of cancellation .
Average	This level indicates that there is some uncertainty about the length or cost of the implementation.
Low	This level indicates that the duration and costs are well defined and are unlikely to change.

Stability

High	This level indicates that this feature will not undergo change. This also shows that it was well understood.
Average	This level indicates that the feature may be subject to change.
Low	This level of stability shows the probability that this feature can be changed over time.

Appendix B - License

Creative Commons Corporation (“Creative Commons”) is not a law firm and does not provide legal services or legal advice. Distribution of Creative Commons public licenses does not create a lawyer-client or other relationship. Creative Commons makes its licenses and related information available on an “as-is” basis. Creative Commons gives no warranties regarding its licenses, any material licensed under their terms and conditions, or any related information. Creative Commons disclaims all liability for damages resulting from their use to the fullest extent possible.

Using Creative Commons Public Licenses

Creative Commons public licenses provide a standard set of terms and conditions that creators and other rights holders may use to share original works of authorship and other material subject to copyright and certain other rights specified in the public license below. The following considerations are for informational purposes only, are not exhaustive, and do not form part of our licenses.

Considerations for licensors: Our public licenses are intended for use by those authorized to give the public permission to use material in ways otherwise restricted by copyright and certain other rights. Our licenses are irrevocable. Licensors should read and understand the terms and conditions of the license they choose before applying it. Licensors should also secure all rights necessary before applying our licenses so that the public can reuse the material as expected. Licensors should clearly mark any material not subject to the license. This includes other CC-licensed material, or material used under an exception or limitation to copyright. More considerations for licensors.

Considerations for the public: By using one of our public licenses, a licensor grants the public permission to use the licensed material under specified terms and conditions. If the licensor’s permission is not necessary for any reason–for example, because of any applicable exception or limitation to copyright–then that use is not regulated by the license. Our licenses grant only permissions under copyright and certain other rights that a licensor has authority to grant. Use of the licensed material may still be restricted for other reasons, including

because others have copyright or other rights in the material. A licensor may make special requests, such as asking that all changes be marked or described. Although not required by our licenses, you are encouraged to respect those requests where reasonable. More considerations for the public.

Creative Commons Attribution-ShareAlike 4.0 International Public License

By exercising the Licensed Rights (defined below), You accept and agree to be bound by the terms and conditions of this Creative Commons Attribution-ShareAlike 4.0 International Public License ("Public License"). To the extent this Public License may be interpreted as a contract, You are granted the Licensed Rights in consideration of Your acceptance of these terms and conditions, and the Licensor grants You such rights in consideration of benefits the Licensor receives from making the Licensed Material available under these terms and conditions.

Section 1 – Definitions.

Adapted Material means material subject to Copyright and Similar Rights that is derived from or based upon the Licensed Material and in which the Licensed Material is translated, altered, arranged, transformed, or otherwise modified in a manner requiring permission under the Copyright and Similar Rights held by the Licensor. For purposes of this Public License, where the Licensed Material is a musical work, performance, or sound recording, Adapted Material is always produced where the Licensed Material is synched in timed relation with a moving image.

Adapter's License means the license You apply to Your Copyright and Similar Rights in Your contributions to Adapted Material in accordance with the terms and conditions of this Public License.

BY-SA Compatible License means a license listed at creativecommons.org/compatiblelicenses, approved by Creative Commons as essentially the equivalent of this Public License.

Copyright and Similar Rights means copyright and/or similar rights closely related to copyright including, without limitation, performance, broadcast, sound recording, and Sui Generis Database Rights, without regard to how the rights are labeled or categorized. For purposes of this Public License, the rights specified in Section 2(b)(1)-(2) are not Copyright and Similar Rights.

Effective Technological Measures means those measures that, in the absence of proper authority, may not be circumvented under laws fulfilling obligations under Article 11 of the WIPO Copyright Treaty adopted on December 20, 1996, and/or similar international agreements.

Exceptions and Limitations means fair use, fair dealing, and/or any other exception or limitation to Copyright and Similar Rights that applies to Your use of the Licensed Material.

License Elements means the license attributes listed in the name of a Creative Commons Public License. The License Elements of this Public License are Attribution and ShareAlike.

Licensed Material means the artistic or literary work, database, or other material to which the Licensor applied this Public License.

Licensed Rights means the rights granted to You subject to the terms and conditions of this Public License, which are limited to all Copyright and Similar Rights that apply to Your use of the Licensed Material and that the Licensor has authority to license.

Licensor means the individual(s) or entity(ies) granting rights under this Public License.

Share means to provide material to the public by any means or process that requires permission under the Licensed Rights, such as reproduction, public display, public performance, distribution, dissemination, communication, or importation, and to make material available to the public including in ways that members of the public may access the material from a place and at a time individually chosen by them.

Sui Generis Database Rights means rights other than copyright resulting from Directive 96/9/EC of the European Parliament and of the Council of 11 March 1996 on the legal protection of databases, as amended and/or succeeded, as well as other essentially equivalent rights anywhere in the world.

You means the individual or entity exercising the Licensed Rights under this Public License. Your has a corresponding meaning.

Section 2 – Scope.

License grant.

Subject to the terms and conditions of this Public License, the Licensor hereby grants You a worldwide, royalty-free, non-sublicensable, non-exclusive, irrevocable license to exercise the Licensed Rights in the Licensed Material to:

reproduce and Share the Licensed Material, in whole or in part; and

produce, reproduce, and Share Adapted Material.

Exceptions and Limitations. For the avoidance of doubt, where Exceptions and Limitations apply to Your use, this Public License does not apply, and You do not need to comply with its terms and conditions.

Term. The term of this Public License is specified in Section 6(a).

Media and formats; technical modifications allowed. The Licensor authorizes You to exercise the Licensed Rights in all media and formats whether now known or hereafter created, and to make technical modifications necessary to do so. The Licensor waives and/or agrees not to assert any right or authority to forbid You from making technical modifications necessary to exercise the Licensed Rights, including technical modifications necessary to circumvent Effective Technological Measures. For purposes of this Public License, simply making modifications authorized by this Section 2(a)(4) never produces Adapted Material.

Downstream recipients.

Offer from the Licensor – Licensed Material. Every recipient of the Licensed Material automatically receives an offer from the Licensor to exercise the Licensed Rights under the terms and conditions of this Public License.

Additional offer from the Licensor – Adapted Material. Every recipient of Adapted Material from You automatically receives an offer from the Licensor to exercise the Licensed Rights in the Adapted Material under the conditions of the Adapter's License You apply.

No downstream restrictions. You may not offer or impose any additional or different terms or conditions on, or apply any Effective Technological Measures to, the Licensed Material if doing so restricts exercise of the Licensed Rights by any recipient of the Licensed Material.

No endorsement. Nothing in this Public License constitutes or may be construed as permission to assert or imply that You are, or that Your use of the Licensed Material is, connected with, or sponsored, endorsed, or granted official status by, the Licensor or others designated to receive attribution as provided in Section 3(a)(1)(A)(i).
Other rights.

Moral rights, such as the right of integrity, are not licensed under this Public License, nor are publicity, privacy, and/or other similar personality rights; however, to the extent possible, the Licensor waives and/or agrees not to assert any such rights held by the Licensor to the limited extent necessary to allow You to exercise the Licensed Rights, but not otherwise.
Patent and trademark rights are not licensed under this Public License.
To the extent possible, the Licensor waives any right to collect royalties from You for the exercise of the Licensed Rights, whether directly or through a collecting society under any voluntary or waivable statutory or compulsory licensing scheme. In all other cases the Licensor expressly reserves any right to collect such royalties.
Section 3 – License Conditions.

Your exercise of the Licensed Rights is expressly made subject to the following conditions.

Attribution.

If You Share the Licensed Material (including in modified form), You must:

retain the following if it is supplied by the Licensor with the Licensed Material:
identification of the creator(s) of the Licensed Material and any others designated to receive attribution, in any reasonable manner requested by the Licensor (including by pseudonym if designated);
a copyright notice;
a notice that refers to this Public License;
a notice that refers to the disclaimer of warranties;

a URI or hyperlink to the Licensed Material to the extent reasonably practicable;
indicate if You modified the Licensed Material and retain an indication of any previous modifications; and
indicate the Licensed Material is licensed under this Public License, and include the text of, or the URI or hyperlink to, this Public License.
You may satisfy the conditions in Section 3(a)(1) in any reasonable manner based on the medium, means, and context in which You Share the Licensed Material. For example, it may be reasonable to satisfy the conditions by providing a URI or hyperlink to a resource that includes the required information.
If requested by the Licensor, You must remove any of the information required by Section 3(a)(1)(A) to the extent reasonably practicable.
ShareAlike.
In addition to the conditions in Section 3(a), if You Share Adapted Material You produce, the following conditions also apply.

The Adapter’s License You apply must be a Creative Commons license with the same License Elements, this version or later, or a BY-SA Compatible License.
You must include the text of, or the URI or hyperlink to, the Adapter's License You apply. You may satisfy this condition in any reasonable manner based on the medium, means, and context in which You Share Adapted Material.
You may not offer or impose any additional or different terms or conditions on, or apply any Effective Technological Measures to, Adapted Material that restrict exercise of the rights granted under the Adapter's License You apply.
Section 4 – Sui Generis Database Rights.

Where the Licensed Rights include Sui Generis Database Rights that apply to Your use of the Licensed Material:

for the avoidance of doubt, Section 2(a)(1) grants You the right to extract, reuse, reproduce, and Share all or a substantial portion of the contents of the database;

if You include all or a substantial portion of the database contents in a database in which You have Sui Generis Database Rights, then the database in which You have Sui Generis Database Rights (but not its individual contents) is Adapted Material, including for purposes of Section 3(b); and

You must comply with the conditions in Section 3(a) if You Share all or a substantial portion of the contents of the database.

For the avoidance of doubt, this Section 4 supplements and does not replace Your obligations under this Public License where the Licensed Rights include other Copyright and Similar Rights.

Section 5 – Disclaimer of Warranties and Limitation of Liability.

Unless otherwise separately undertaken by the Licensor, to the extent possible, the Licensor offers the Licensed Material as-is and as-available, and makes no representations or warranties of any kind concerning the Licensed Material, whether express, implied, statutory, or other. This includes, without limitation, warranties of title, merchantability, fitness for a particular purpose, non-infringement, absence of latent or other defects, accuracy, or the presence or absence of errors, whether or not known or discoverable. Where disclaimers of warranties are not allowed in full or in part, this disclaimer may not apply to You.

To the extent possible, in no event will the Licensor be liable to You on any legal theory (including, without limitation, negligence) or otherwise for any direct, special, indirect, incidental, consequential, punitive, exemplary, or other losses, costs, expenses, or damages arising out of this Public License or use of the Licensed Material, even if the Licensor has been advised of the possibility of such losses, costs, expenses, or damages. Where a limitation of liability is not allowed in full or in part, this limitation may not apply to You.

The disclaimer of warranties and limitation of liability provided above shall be interpreted in a manner that, to the extent possible, most closely approximates an absolute disclaimer and waiver of all liability.

Section 6 – Term and Termination.

This Public License applies for the term of the Copyright and Similar Rights licensed here. However, if You fail to comply with this Public License, then Your rights under this Public License terminate automatically.

Where Your right to use the Licensed Material has terminated under Section 6(a), it reinstates:

automatically as of the date the violation is cured, provided it is cured within 30 days of Your discovery of the violation; or

upon express reinstatement by the Licensor.

For the avoidance of doubt, this Section 6(b) does not affect any right the Licensor may have to seek remedies for Your violations of this Public License.

For the avoidance of doubt, the Licensor may also offer the Licensed Material under separate terms or conditions or stop distributing the Licensed Material at any time; however, doing so will not terminate this Public License.

Sections 1, 5, 6, 7, and 8 survive termination of this Public License.

Section 7 – Other Terms and Conditions.

The Licensor shall not be bound by any additional or different terms or conditions communicated by You unless expressly agreed.

Any arrangements, understandings, or agreements regarding the Licensed Material not stated herein are separate from and independent of the terms and conditions of this Public License.

Section 8 – Interpretation.

For the avoidance of doubt, this Public License does not, and shall not be interpreted to, reduce, limit, restrict, or impose conditions on any use of the Licensed Material that could lawfully be made without permission under this Public License.

To the extent possible, if any provision of this Public License is deemed unenforceable, it shall be automatically reformed to the minimum extent necessary to make it enforceable. If the provision cannot be reformed, it shall be severed from this Public License without affecting the enforceability of the remaining terms and conditions.

No term or condition of this Public License will be waived and no failure to comply consented to unless expressly agreed to by the Licensor.

Nothing in this Public License constitutes or may be interpreted as a limitation upon, or waiver of, any privileges and immunities that apply to the Licensor or You, including from the legal processes of any jurisdiction or authority.

Creative Commons is not a party to its public licenses. Notwithstanding, Creative Commons may elect to apply one of its public licenses to material it publishes and in those instances will be considered the "Licensor." The text of the Creative Commons public licenses is dedicated to the public domain under the CC0 Public Domain Dedication. Except for the limited purpose of indicating that material is shared under a Creative Commons public license or as otherwise permitted by the Creative Commons policies published at creativecommons.org/policies, Creative Commons does not authorize the use of the trademark "Creative Commons" or any other trademark or logo of Creative Commons without its prior written consent including, without limitation, in connection with any unauthorized modifications to any of its public licenses or any other arrangements, understandings, or agreements concerning use of licensed material. For the avoidance of doubt, this paragraph does not form part of the public licenses.

Creative Commons may be contacted at creativecommons.org.

LISTE DE RÉFÉRENCES BIBLIOGRAPHIQUES

Ashkenas, J. (2015). *UnderscoreJS*. Récupéré sur UnderscoreJS: http://underscorejs.org/

Croft, J. (s.d.). *Django Presentation*. Consulté le 12 6, 2015, sur http://www.thomaswhitton.com/django-presentation/#/4

Facebook Inc. (2015). *Flux*. Consulté le 12 9, 2015, sur Flux: https://facebook.github.io/flux/

Facebook Inc. (2015). *React*. Consulté le 12 9, 2015, sur React: https://facebook.github.io/react/index.html

GitHub, I. (2015). *Most Stars Repository*. Récupéré sur GitHub: https://github.com/search?q=stars:%3E1&s=stars&type=Repositories

GitHub, Inc. (2015). *GitHub*. Récupéré sur GitHub: https://github.com

GitHub, Inc. (2015). *Atom*. Consulté le 12 2015, 4, sur Atom: https://atom.io/

Google. (2015). *Angular Js*. Consulté le 12 7, 2015, sur AngualarJS: https://angularjs.org/

Hampton Catlin, N. W. (s.d.). *SASS*. Consulté le 11 29, 2015, sur SASS: http://sass-lang.com/

Harrington, C. (2015, 01 12). *reactjs-vs-angular-js-performance-comparison-knockout*. Récupéré sur www.codementor.io: https://www.codementor.io/reactjs/tutorial/reactjs-vs-angular-js-performance-comparison-knockout

IntelliJ IDEA. (s.d.). *PyCharm*. Récupéré sur https://www.jetbrains.com/pycharm/

Jeremy Ashkenas, D. (2015). *BackboneJS*. Récupéré sur BackboneJS: http://backbonejs.org/

PyData Development Team. (s.d.). *Pandas*. Récupéré sur http://pandas.pydata.org/

Ryan Lanciaux, J. L. (s.d.). *Griddle*. Consulté le 12 7, 2015, sur http://griddlegriddle.github.io/Griddle/

Sellier, A. (s.d.). *LESS*. Consulté le 11 29, 2015, sur http://lesscss.org/

Snook.ca Web Development, Inc. (s.d.). *SMACSS*. Consulté le 11 29, 2015, sur https://smacss.com/

Trello Inc. (2015). *Trello*. Récupéré sur Trello: https://trello.com/