



Collège Sciences et technologies

Cahier des charges

Développement d'un outil client pour la
visualisation de l'annotation
fonctionnelle de données protéomiques

Auteurs :

Julie Blasquiz
Romain Luck
Marc Mongy
Michèle Raffaelli

Encadrants : Patricia Thebault, Karine Dementhon

Bordeaux

26-02-2018

L^AT_EX

Table des matières

1	Introduction	3
2	Contexte et état de l'art	4
3	Besoins fonctionnels	7
4	Besoins non fonctionnels	9
5	Les choix	10

1 Introduction

La recherche à l'université de Bordeaux repose sur trois grands départements dont celui des Sciences du vivant et de la santé.

Ce département regroupe différents types de structures dont l'unité de recherche Microbiologie Fondamentale et Pathogénicité (UMR 5234). Cette unité travaille sur les champignons d'intérêt médical qui constituent la première cause de maladies parasitaires opportunistes dans les pays industrialisés en raison du faible nombre de molécules antifongiques disponibles, de la diversité croissante des pathogènes émergents et des problèmes de résistance antifongique.

C'est au sein de cette unité de recherche qu'une étudiante de M2 Recherche Biologie Santé Parcours Microbiologie Immunologie à été accueillie dans le cadre d'un stage dans l'équipe Candida et Pathogénicité qui effectue des recherches sur les levures Candida d'intérêt médical.

Pour compléter le travail effectué par cette étudiante portant sur l'analyse fonctionnelle de gènes impliqués dans la résistance des levures Candida à la phagocytose macrophagique[1], un sujet de programmation visant à développer un outil de visualisation d'annotation fonctionnelle de données protéomiques nous a été demandé.

2 Contexte et état de l'art

Contexte biologique

Dans le cadre d'une étude sur les *Candida*[1], 3 espèces vont être étudiées. Elles ont été sélectionnées en fonction de leur différence en terme de stratégies de pathogénicité chez l'Homme. L'objectif de cette étude est de comparer, par une analyse protéomique, les mécanismes de survie de ces espèces phagocytées dans les macrophages.

Les expériences reposent sur une infection *in vitro* de macrophages avec les pathogènes d'intérêt. En contrôle, les levures seules en absence des macrophages sont utilisées. Les extraits protéiques de levures internalisées ou seules sont préparés. Les protéines sont identifiées et quantifiées par chromatographie liquide couplée à de la spectrométrie de masse en tandem (LC-MS/MS), afin de sélectionner les protéines statistiquement dérégulées en condition de phagocytose macrophagique. Les résultats préliminaires sont obtenus par spectrométrie de masse en tandem afin de sélectionner des protéines statistiquement dérégulées. Entre 400 et 1000 protéines, selon l'espèce *Candida*, sont sélectionnées.

L'analyse de ces protéines se fait manuellement par BLAST afin d'attribuer une fonction putative à chaque protéine dérégulée.

Le projet de stage de l'étudiante en Master 2 Microbiologie Immunologie est centré autour du lien entre les interactions moléculaires et les comportements communs ou atypiques chez les levures étudiées. Devant de telles quantités de données à analyser il semble évident qu'un traitement manuel est trop fastidieux. C'est pourquoi une solution faisant appel à des outils bioinformatiques va être proposée.

Ces outils existent déjà mais il faut les chercher séparément sur les différentes plateformes correspondantes.

L'intérêt de ce projet, résumé en figure 1, est donc de proposer, sous forme d'une page Internet, un accès aux différentes informations nécessaires à l'étude menée (sur la base d'une Gene Ontology-SLIM rentrée par le biologiste sur la page Internet).

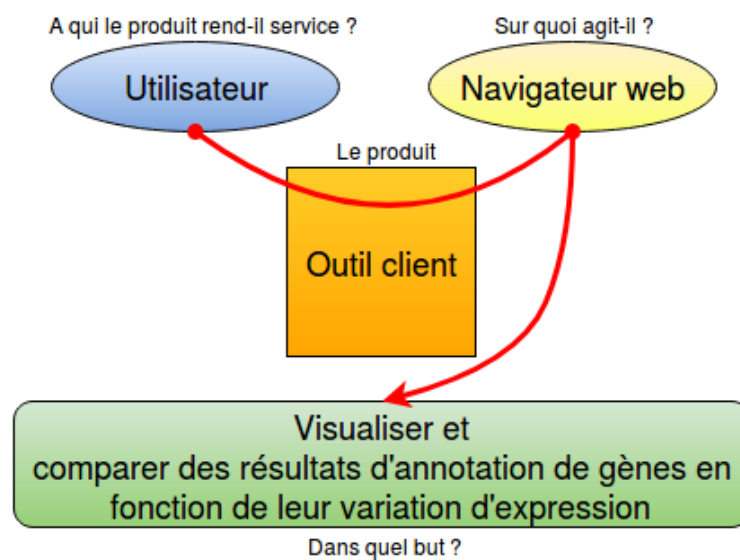


FIGURE 1 – Bête à cornes représentant la demande du client
1

Ainsi l'utilisateur peut avoir un accès simple, clair et rapide à toutes les informations qu'il désire en un clic.

1. Le terme "bête à cornes" est issu de la méthode APTE : APplication aux Techniques d'Entreprise.

En alliant ainsi biologie et informatique il est possible de faire gagner un temps précieux au biologiste et de fournir un outil de recherche qui aura pour vocation d'être réutilisable dans un autre contexte également (nécessité de créer un outil qui ne soit pas ultra spécifique mais qui permette toutefois de répondre précisément à une question posée). De ce fait il est important de travailler en étroite collaboration avec les biologistes.

Technologies à utiliser

Afin de mener à bien ce projet plusieurs outils vont être exploités.

Les langages de programmations utilisés vont être HTML/CSS et JavaScript.

Le serveur Node.js et le framework AngularJS nous ont été imposés. En ce qui concerne le framework AngularJS, cela constitue une bonne base de Framework JavaScript de par sa large communauté d'utilisateurs, une documentation riche, un environnement complet (Modèle Vue Contrôleur) qui en permet un usage facilité côté client et une interaction côté serveur. De plus, afin de pouvoir intégrer du JavaScript côté serveur la plateforme Node.js nous a été prescrite. Elle permet donc d'obtenir une exécution de code JavaScript en s'affranchissant d'un navigateur internet.

Enfin, en ce qui concerne l'affichage des graphes, et du fait que ceux-ci doivent être "cliquables" il nous est imposé d'utiliser la librairie de référence actuelle, à savoir D3.js.

3 Besoins fonctionnels

L'outil client développé ici dans le cadre du projet de programmation doit répondre à certains besoins. L'ensemble de ces besoins est résumé en figure 2. On distingue les besoins fonctionnels (en vert sur la figure 2) des besoins non fonctionnels (en violet sur la figure 2).

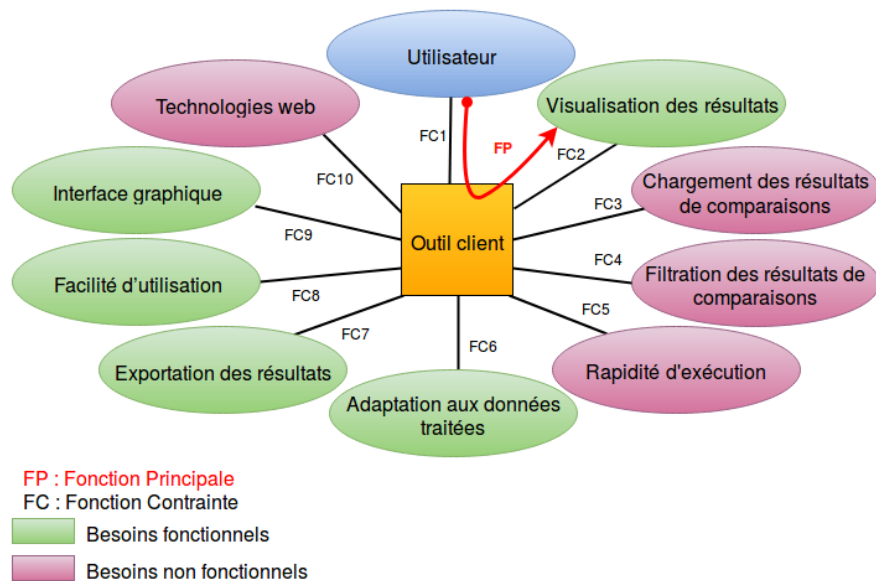


FIGURE 2 – Graphe des interactions représentant les besoins de l'outil client

Les besoins fonctionnels sont définis comme étant des fonctionnalités demandées spécifiquement par le client et dont l'action peut être visualisée par l'utilisateur du produit.

Ici, le besoin fonctionnel majeur de l'outil est la visualisation des résultats (FC2 sur la figure 2) : des graphes de résultats de comparaisons interactifs doivent s'afficher après analyse des données de l'utilisateur.

Un autre besoin fonctionnel est celui de l'adaptation de l'outil aux données traitées (FC6 sur la figure 2) : l'utilisateur doit avoir la possibilité de rentrer, en tant

que critère, sur quel organisme portent les données et la fonction biologique du sujet d'étude.

Il devra également avoir la possibilité de choisir le type de graphe qu'il souhaite obtenir en résultat (histogramme ou diagramme de Venn).

L'exportation des résultats de visualisation (FC7 sur la figure 2) répond au besoin fonctionnel suivant : l'utilisateur doit avoir la possibilité d'exporter les résultats de visualisation qu'il a obtenu.

La facilité d'utilisation de l'outil (FC8 sur la figure 2) va être également considérée comme un besoin fonctionnel : l'outil doit être intuitif et simple d'utilisation.

Enfin, le dernier besoin fonctionnel n'est autre que celui de l'interface graphique (FC9) : l'interface de l'outil web développée doit être agréable au visuel.

4 Besoins non fonctionnels

Les besoins non fonctionnels sont les éléments qui participent au bon fonctionnement du programme mais qui ne sont pas directement visibles par l'utilisateur.

Technologies web (FC10 figure 2) : le script doit pouvoir être accessible depuis un site internet. Pour ce faire les technologies HTML et CSS qui permettent de créer des pages web seront utilisées, ainsi que le langage de programmation JavaScript associé à JQuery et Node.js qui permettent entre autres de réaliser des scripts, rendre les pages web dynamiques et permettre des interactions entre la page et le serveur.

Chargement des résultats de comparaison (FC3 figure 2) : une fois les résultats calculés ils doivent être suffisamment légers pour pouvoir être chargés sans problèmes, sans nécessiter trop de processeur ou de bande passante.

Filtration des résultats de comparaison (FC4 figure 2) : une fois les résultats calculés et chargés, l'utilisateur doit être capable de choisir ceux qui l'intéressent pour les afficher.

Rapidité d'exécution (FC5 figure 2) : dans l'éventualité où le client doit utiliser le script de manière intensive il est conseillé que le script soit optimisé pour réduire les temps de calcul et utiliser moins de processeur.

5 Les choix

Les annotations de gènes devront être issues de l'outil d'analyse d'annotation DAVID. Le web service de DAVID permet d'interagir avec l'outil afin d'automatiser les tâches souhaitées par l'utilisateur. Le web service proposé sur le site de DAVID est disponible en différents langages de programmation : Java, Perl, PHP (phpDAVIDws), Python ou encore R. Créer un web service par nous même, sans passer par les packages existants sur le site, en récupérant les informations nécessaires en utilisant simplement l'URL (afin de créer des scripts dans d'autres langages) est également envisagé.

Quel que soit le procédé utilisé, notre choix de langage de programmation s'est porté sur Python par le fait que nous traitons ici des informations biologiques sous forme de texte et que ce langage est connu pour sa puissance de traitement de texte. La flexibilité du langage Python aux données biologiques via des modules tel que BioPython, est un second critère de choix. Python offre également des bibliothèques destinées au traitement de nombreux formats de fichiers utilisés en biologie tels que les fichiers csv, txt, FASTA...

De ce fait, et de manière plus générale, tous les webs services utilisés dans ce projet se feront avec le langage de programmation Python.

De même, pour le cas des "Gene Ontologies", Categorizer, codé en Python semble être une bonne alternative pour l'enrichissement.

Références

- [1] Lesbats J. (2018). Analyse fonctionnelle de gènes impliqués dans la résistance des levures *Candida* à la phagocytose macrophagique. Université de Bordeaux, UFR Sciences de la Santé.