Présentation de l'organisme

présentation générale

Fondée en 1976 lors de la cission de l'université de Clermont, l'Université d'Auvergne (UdA) actuellement présidée par Alain Eschalier compte plus de 16 000 étudiants dont 3 000 étrangers rassemblés au sein de 7 composantes :

- École de Droit
- École d'Économie
- École Universitaire de Management
- Faculté de Médecine
- Faculté de Pharmacie
- Faculté de Chirurgie Dentaire
- Institut Universitaire Technologique

L'université dispose aussi de 22 laboratoirs de recherche

conditions du stage

J'ai effectué mon stage au sein du laboratoir EA CIDAM (Conception, Ingenierie et Developpement de l'Aliment et du Medicament).

Dirigé par le professeur M. ALRIC, l'équipe de recherche travaille nottament à comprendre, évaluer et analyser, dans l'environnement digestif, différentes situations physio-pathologiques liées au vieillissement, à la présence de bactéries pathogènes (en particulier d'Escherichia coli entérohémorragiques), ou encore à celle de produits toxiques, de xénobiotiques, en particulier de polluants.

C'est dans ce cadre là que Bérénice BATUT, ma tutrice de stage, dévellope un environnement bioinformatique permettant de faciliter l'analyse de données massives issues du microbiote AsaiM.

AsaiM :*an environment to analyze intestinal microbiota data*

ASaiM (*Auvergne Sequence analysis of intestinal Microbiota*) est un environnement créé pour analyxer les microbiotes, et plus particulièrement le microbiote intestinal. Afin d'avoir une analyse correcte de ce microbiote, il faut étudier l'ensemble des génomes issus du milieu et donc faire de la métagénomique.

Il existe de nombreux outils bioinformatiques permettants d'analyser des données métagénomiques et présentant un intérêt certain à êtres intégrés au sein d'Asaim nottament HumanN2, MetaphLan2 ou encore GraphLan.

Cependant ces outils, comme la grande majorité des outils bioinformatqies, doivent êtres lancés en ligne de commande. Ce qui pose généralement problème lorsque l'utilisateur est un biologiste sans compétence en programmation.

Afin de venir à bout de ce problème, une insance basée sur Galaxy à étée intégrée à AsaiM afin de permettre une utilisation facilité de ces outils.

Galaxy :

Galaxie est une plate-forme ouverte et open source, ses données étant accessibles librement en ligne, permettant d'utiliser simplement de nobreux outils d'analyse via son interface.

On peut accéder à Galaxy de différentes façons. Tout d'abord, il existe plusieurs serveurs publics de galaxy. L'avantage de ces serveurs étant qu'ils disposent souvent d'une importante puissance de clacul. Cependant énormément de personnes utilisent ces serveurs ce qui cause d'important temps d'attentes. De plus un nombre limité d'outils sont installés sur ces serveurset il y a très peu de chances qu'un administrateur installe des outils si la demande lui est faite par une petite communauté.

Il existe cependant de très nombreux serveurs liés à des laboratoirs ou des entreprises ces serveurs disposent généralement d'une bonne puissance de calcul et les administrateurs sont sencés être disponibles losqu'il s'agit d'ajouter des outils.

On peu aussi télécharger une instance de Galaxy en local sur n'importe quel ordinateur. Cependant dans ce cas là, il faut gérer soit même l'instalation des outils et la puissance de calcul est limité à celle de l'ordinateur.

Un des avantages de Galaxy est que si vous êtes connectés sur un compte, tous les fichiers d'entrée ou de sortie produit grâce à galaxy sont stockés dans l'historique de l'instance et ne peuvent être définitivement supprimés que par un administrateur. Ainsi même plusieurs mois après, on peut récupérer a nouveau nos données.

Mais l'intérêt majeur de cette instance est de pouvoir lancer des outils de bioinformatque, qui fonctionnent normalement en ligne de commande afin d'effectuer des analyse complètes mais surtout reproductibles. Cela grâce aux *wrappers*, des interfaces faisant le lien entre l'utilisateur et la ligne de commande de l'outil.

Wrappers :

Q'est ce qu'un wrappers ?

Les wrappers sont la base de galaxy. Ils servent d'interface entre l'utilisateur et la ligne de commande qui lance l'outil.
Un wrapper utilise une version de l'outil fixée préalablement. Ainsi même si

l'outil est mis à jour, le wrapper continue d'utiliser la version prédéfinie. Celà rend les analyses effectuées reproductibles. En effet, après une mise a jour, l'outil peut analyser les données différement nécessiter un type de fichiers d'entrée différent ou encore fournir des fichiers de sortie n'ayants rien a voir avec les fichiers fornis avant la mise a jour. Dans ce cas là il y a un fort risque que les résultats issus de la conparaison de fichiers issus de différentes ne soit biaisée et fausse les résultats finaux.

De plus les wrappers gèrent les fichiers d'entrée et de sortie, les paramètres nécessaires ainsi que les dépendances de l'outil, tous les programmes nécessaires au bon fonctionnement de l'outil.

Composition d'un wrapper

Les wappers utilisés dans Galaxy sont de logiciels codés en langage .xml et sont composés de la manière suivante :

On retrouve d'abord les informations générales sur l'outil qui regroupent sont son nom, son id, sa version et sa description. Viennent ensuite les packages requis par l'outil pours lesquels on précise la version.

Ensuite vient la partie "command". C'est elle qui fait le lien entre la ligne de command de l'outil et les paramètres donnés dans galaxy. Elle reprend la totalité de la ligne de commande : le nom de l'outil, le ou les fichier d'entrée et de sortie ainsi que toutes les options de l'outil. Chaque fichier ou valeur d'option est founie par la section input ou la section output.

Puis la section inputs, c'est dans cette partie que l'on définit ce qui va être affiché à l'écran lors de l'utilisation de l'outil dans galaxy notamment les noms que l'on souhaite donner aux différentes entrées afin de faciliter la compréhention ou encore un commentaide d'aide. On y définit les propriétées des fichiers d'entée, nottament leur type (txt, Fasta, FastQ,etc.), mais aussi celles de chaque paramètre que l'on souhaite afficher. Tout d'abord le type du paramètre, il peut s'agir d'une valeur à rentrer (entier ou décimal) ou d'un text, ce paramètre peut aussi être une barre de sélection ou encore une check box.
Chaque fichier d'entrée ou paramètre que l'on souhaite utiliser doit être présent dans cette section.

La section outputs permet de gérer les sorties de l'outil que l'on veux voir apparaître dans la barre de l'historique de galaxy. On y définit notamment le nom que l'on veux donner à la sortie. On peut aussi modifier son format. Toutes les sorties de l'outil doivent être référencées dans cette section.

La section tests est, comme son nom l'indique, liée a la phase de test du

wrapper. On fournit dans cette section les fichier d'entrée et de sortie nécessaires pour le test du wrapper ainsi que les valeures des différents paramètres. On peut fournir autant de jeux de données différents afin avoir un batterie de tests la plus varié possible. Cependant la totalité des fichiers et des paramètres doivent être testés au minimum une foi chacuns.

On retrouve à la fin la section help qui permet d'afficher des information sur l'outil. Et la section citations dans laquelle on liste les différentes citations avec leur doi.

Test logiciel :

Tout logiciel ou programme informatique, tels que les wrappers par exemple, a besoin d'être testé afin de garantir une bonne fiabilité au niveau du fonctionnement du programme ainsi que son comportement en présence des différentes entrées.
L'objectif d'un test est d'exécuter un programme dans l'intention d'y trouver des défauts et non pas pour démontrer que le programme ne contient plus d'erreur. Il faut donc que la personne chargée des tests ai pour but de trouver des erreurs, autrement elle n'en trouvera que peu ou pas.
Cependant il est impossible d'obtenir un programme sans défauts en effet les tests ne peuvent vérifier qu'une partie des possibilitées. Cependant un objectif réalisable est de corriger les erreurs sévères et récurentes à l'aide de données de test représentatives.

Un autre problème est que le test logiciel est un processus destructif, à l'opposé de la programmation qui est un processus constructif. En effet le but du programmateur est de créer un logiciel qui fonctionne et rechigne souvent a effectuer les tests.
On peut comparer la programmation à l'orthographe, on a toujours plus de mal à corriger ses propores fautes et cela demande plus d'efforts. Ainsi, il arrive souvent que des logiciels créés au sein d'équipes restreintes, comme les wrappers de Galaxy, soient publiés sans tests et donc sans garantie de fonctionnement. Ce qui peu poser problème lors de l'utilisation.
L'une des meilleures solutions afin de produire un test convenable est de faire appel à des personnes extérieures à l'équipe de développement (ce qui n'est pas forcément possible pour des travaux de faible envergure) pour développer ces tests avec un regard neuf sur le logiciel.

Il existe de nombreux types de tests différents (unitaires, intégration, performance, etc.) cependant deux types de tests sont principalement utilisés. Tout d'abord les tests unitaires dont le principe est de tester de façon indépendante un seul élément du logiciel.
Le second type de tests est le test fonctionnel qui consiste dans le fait de tester

la totalité d'un programme en reproduisant les « conditions réelles ». Celà veux dire que lors du test on effectu ce que l'utilisateur va potentiellement faire. Celà permet de voir à quelles erreures ce dernier peut être confronté. C'est ce type de test que nous avons pratiqué pour nos wrappers.