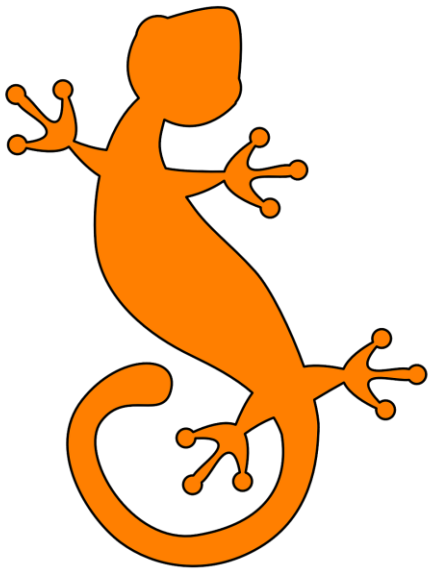
Cahier des charges : Iguana

(PGROU n°1)



Encadrants : Carito Guziolowski & Bertrand Miannay
Justin Voïnéa
Jules Paris
Khalil Boulkenafet
Jinhui Liu
Pierre Le Jeune

1. Présentation du sujet :

Ce projet fait suite au projet d'application de cette année qui a conduit à la création d'une IHM s'appelant Iguana. Lors de ce projet, les élèves ont utilisés les travaux de M Miannay et Mme Guziolowski sur le myélome multiple afin de réutiliser une partie de leurs scripts et les intégrer dans une interface graphique permettant à la fois de simplifier le processus de traitement des graphes mais aussi d'afficher ces graphes au fur et à mesure. Ce nouveau projet à pour but d'ajouter de nouvelle fonctionnalités à Iguana permettant de catégoriser le patient selon les données qui ont été récupérées. En effet, il existe plusieurs catégories de malade et il est important de bien les catégoriser afin d'adapter le traitement au mieux. Cette fonctionnalité supplémentaire utilise les graphes créés avec la première version d'Iguana et des méthodes de deep learning afin d'établir une classification la plus juste possible.

2. Tâches à effectuer :

Importation et structure des données.

Objectif : Permettre à Iguana de charger une matrice transcriptomique (matrice de dimension n*m à valeur dans [0,1] représentant l'expression de n gènes pour m patients) à partir d'un fichier source. Ce fichier, qui aura un format bien défini, contiendra des données cliniques sur ces patients.

Description : Le fichier de données est chargé par l'utilisateur via une interface de chargement (explorateur de fichiers). Les données sont ensuite structurées sous la forme d'une matrice.

Contraintes : Vérifier que le fichier respecte la nomenclature établie.

Tâches à réaliser :

- Interface de chargement des fichiers patient
- Mise en forme des données pour les utiliser dans le script Python

Calcul de similarité.

Objectif : Ajouter la fonctionnalité de calcul de similarité à Iguana.

Description : Les données génétiques de M individus sont contenus dans une matrice de taille n*M. Cela signifie que, pour chacun de ces individus, seuls n gènes sont concernés. Ces données représentent l'état génétique du patient. De plus, un diagnostic pour ces patients à déjà été établi et est connu.
Nous disposons également d'un graphe représentant les interactions entre N gènes du génome humain (N >= n). Celui-ci a été réduit, sans perte d'information, en k sous-graphes de taille inférieure plus facilement exploitables.
Il va donc s'agir d'appliquer, pour chaque individu, l'algorithme de similarité entre les n gènes (qui correspondent à n noeuds du graphe dans un état particulier) et les k sous-graphes.

Nous obtenons alors une matrice k*M dont chaque colonne représente la similarité entre l'état du patient et l'état qu'il doit avoir au vue de son information génétique.

Contraintes : L'utilisateur doit avoir réduit un graphe et importé une matrice transcriptomique avant de pouvoir lancer le calcul de similarité.
Comment les résultats du calcul sont transmis à l'utilisateur ? Ils apparaissent à l'écran ou sont écrits dans un fichier ?

Tâches à réaliser :

- Prise en main des scripts Iggy-POC pour le calcul et de l'environnement
- Ajout de l'interface graphique dans Iguana
- Implémentation du calcul en lui même
    - faire le lien entre les données et le graphe (en fonction du nombre de gènes)
- Affichage du résultat de la similarité (vertue pédagogique)
    - Création du graphe des composants avec des couleurs/chiffres pour représenter la similarité
    - Si possible générer des images (à la manière de heatmap)
- Deux modes de fonctionnement:
    - Calcul de similarité sur l'ensemble de la base de donnée patient
    - Calcul de similarité sur un seul patient avec affichage dans Cytoscape du graphe des composants correspondant.

Classificateur :

Objectif : Récupérer les données fournies par le calcul de similarité et les utiliser afin d'établir une classification en utilisant les travaux de data mining déjà créés par M Miannay.

Description : Pour cela, il faudra utiliser les données de recherche déjà utilisées lors du concours de classification de 2017. Ces données seront récupérables grâce à la machine virtuelle Docker qu'il faudra prendre en main puis implémenter ces fonctionnalités dans Iguana ainsi que tester plusieurs méthodes de Data Mining afin d'affiner le modèle. On pourra entre faire varier le nombre de données disponibles pour l'apprentissage.

Tâches à réaliser :

- Insertion d'un nouveau module dans Iguana
    - Fonctionnalité de chargement de données d'apprentissage
    - Création du classificateur à partir de Random Forest
        - Prise en main des scripts en R dans Iggy-POC
        - Intégration de ces de ces scripts dans Iguana (Soit R soit Python)
    - Module de test
        - Chargement des données de test
        - Affichage des résultats de test
- Amélioration du modèle de départ

Portage vers plateforme Unix (MacOS ou Linux).

Objectif : Rendre l'application Iguana entièrement opérationnelle sur ces OS tout en conservant la compatibilité Windows

Tâches à réaliser :

- Choix de la plateforme au démarrage
- Possibilité de modifier la plateforme si besoin
- Conserver le fonctionnement des différents modules (à valider avec les graphes d'essai)

3. Répartition des tâches et organisation du projet