**Réunion du 29 décembre 2018 avec Thomas**

La réunion a permis d'affiner les spécifications pour le pipeline à déployer.

En entrée on fournit un dossier de données patients avec l'architecture suivante : (que l'on retrouve dans le jeu de données fourni par Thomas)

* Un dossier de données contenant des dossiers patients sous forme xxx-xxx (x étant un digit entre 0 et 9)
* Dans chaque dossier patient on retrouve toujours la même architecture : un dossier dcm contenant la pile de dicom et un ou plusieurs dossiers contenant les masques notés l1, l2, etc, associés à chaque lésion du patient.
* Dans chaque dossier de lésion, on retrouve trois masques : le kmean sous forme de .tif, un dossier pour le masque 2.5 (pile de masque) et un dossier pour le masque 40 (pile de masque) *NB: ces deux derniers (2.5 et 40) peuvent être sous forme de .tiff et non en pile si ils ont été corrigés (axe z inversé et ajout de couches dans la pile).*

Le vendredi 29 janvier après-midi, nous présentons à Thomas un pipe qui permet de traiter un jeu de plusieurs données patients avec pour output une structure qui associe à chaque lésions d'un patient les caractéristiques extraites de pyradiomics.

Informations supplémentaires pour la mise en place du pipeline :

-> Pour chaque lésion d'un patient, on forme un masque par la méthode du vote majoritaire à partir des 3 masques : 2.5, 40 et kmean. (*Méthode du vote majoritaire seulement dans un premier temps, pas de STAPLE). Nous nous sommes inspirés du code .py de Thomas pour le vote majoritaire.*

-> Le masque résultant du vote majoritaire est fourni, avec la pile de dicom du patient correspondant, en entrée à pyradiomics pour extraire les features.

-> On oublie dans un premier temps la technique de crop (réduction de dimension) qui était réalisé dans les scripts fiji : on va ici fournir les piles entières.

-> L'output devra être une structure de données permettant d'extraire facilement les caractéristiques obtenues par pyradiomics pour chaque lésion de chaque patient.

Une fois ce pipeline fonctionnel et validé par Thomas, nous organiserons une réunion avec Diana pour attaquer la partie machine learning.

---------------------------

**Réunion du mercredi 24 janvier 2018 avec Thomas**

Objectif : Validation du pipe de pré-traitements des données patients pour l'obtention de caractéristiques des lésions.

Problème pour faire tourner le script sur notre PC portable.

Information supplémentaire : les piles de DCM doivent être pré-traitées pour pouvoir être en unité SUV/Bequerel standardisée.

---------------------------

**Réunion du lundi 29 janvier 2018 avec Diana et Thomas**

*Objectifs*

#1 Présentation du pipe fonctionnel de pré-traitement

#2 Sélection des features à extraire pour comparaison de résultats

#3 Réfléchir au choix du modèle de machine learning

--- Résumé

**Points d'intérêts**

**-**Pipe de pré-traitement

Effectuer la conversion, par patient et pour chacune des slices dicom, dans l'unité standardisée en récupérant la valeur liée à la bonne balise dans les slices (références des balises dans le code Fiji de Thomas).

Pré-traitement scalable mais penser à exporter le code en .py pour contourner des limitations de ressources des notebooks lorsque jeu de données en situation réelle (>100Go)

**-**Extraction de paramètres

Extraire le fichier de configuration au format yaml utilisé pour initialiser l'extracteur de pyradiomics et l'envoyer à Thomas pour valider le choix des paramètres d'extractions. (*e.g.* le bin width doit être à 0.3)

**-**Choix du modèle d'apprentissage

Les labels qui seront utilisés dans l'apprentissage sont stockés dans un tableur contenant le temps au bout duquel un patient a rechuté.

D'un point de vue clinique, une classification binaire "patient a rechuté" / "patient n'a pas rechuté" n'a pas de sens car les patients atteints de cette maladie rechutent dans la majeure partie des cas. Un modèle de SVM binaire n'est pas pertinent.

Envisager l'approche random forest et lire le papier de 2008 RANDOM SURVIVAL FORESTS 1 By Hemant Ishwaran, Udaya B. Kogalur, Eugene H. Blackstone and Michael S. Lauer et se renseigner sur le code python existant.

--- Prochaines étapes

* Récupérer le jeu de données au CHU auprès de Thomas (RDV jeudi 01/02 après-midi, horaire à définir)
* Implémenter la standardisation des données dans le pipe de pré-traitement
* Envoyer le fichier de configuration d'extraction de pyradiomics à Thomas
* Valider les valeurs des features sur le *sample*des 3 patients
* Lire le papier random forest, rechercher un code Python équivalent et designer un code python pouvant traiter des batchs de patients labelisés (le label n’est pas pris en compte actuellement dans le pipe --> pourrait simplement être un attribut de la *class* Patient)

Les features à extraire pour comparer avec les données de Thomas sont les suivantes :

- EntropyGLCM

- HomogeneityGLCM

- SUVmax

- SUVpeak

- Volume

- TLG

- NbVoxel

- DissimilarityGLCM

- HGRE

- ZLNU

- SZHGE

- ZP

---------------------------

**5 février 2018**

- instancier un extracteur depuis un fichier params.yaml

- regarder les features demandées par Thomas et comparer les définitions mathématiques avec celles de pyradiomics.

- décomposer le code en une partie extraction de features et un partie machine learning

- décomposer le pipe en plusieurs scripts .py pour un code plus modulaire

- code pour convertir la pile de dcm en suv.

---------------------------

**Réunion du 7 février 2018 avec Thomas**

Récupération des données

- Faire attention aux masques déjà construit 25.tif et 40.tif --> ils sont de la forme numérique 0 et 255 (seulement les deux valeurs extrêmes) - en gros ils ne sont pas au format binaire - il faut les remettre en binaire pour éviter les problèmes lors du calcul du masque par méthode vote majoritaire.

---- Lexique du fichier csv dont on extrait les features par lésion et par patient :

- endpointPFS ->  1 si patient a rechuté, 0 s’il ne rechute pas

- time to PFS : temps en jours au cours duquel le patient a rechuté. Pour les patients qui n'ont pas rechuté, cette date correspond aux dernières nouvelles

Attention, pour faire la comparaison entre les valeurs de référence des features fournies par Thomas et celles calculées par pyradiomics, se baser sur le masque 40 uniquement.

Ne pas prendre en compte la feature TLG dans un premier temps. GLCM validé.

Prendre ZP avec précaution (pas la même définition dans pyradiomics que celle adoptée par Thomas dans son calcul de features).

--

**Note concernant le la conversion des images Dicom et simple ITK**

La fonction combine\_slices de dicom\_numpy applique le rescale slope / rescale intercept s’ils sont présents dans les tags de l'image. Cf doc <http://dicom-numpy.readthedocs.io/en/latest/index.html?highlight=combine%20slice>

The image array dtype will be preserved, unless any of the DICOM images contain either the [Rescale Slope](https://dicom.innolitics.com/ciods/ct-image/ct-image/00281053) or the [Rescale Intercept](https://dicom.innolitics.com/ciods/ct-image/ct-image/00281052) attributes. If either of these attributes are present, they will be applied to each slice individually.

---

Faire un tuto sur le git pour la création d'un virtual environnement avec les modules nécessaires et versions.