

# IMN530 - Reconstruction et analyse d'image médicale

## TP 1 - Modalités d'imagerie médicale, bruit et débruitage

Vous devez me remettre un rapport PDF. Le style et la forme de vos réponses sont libres. Vous pouvez travailler en équipe de 2-3 ou seul. Soyez originaux. Gérer votre temps adéquatement. Il y a beaucoup de questions et c'est facile de s'enfoncer et perdre du temps sur des problèmes mineurs.

Vous devez remettre un rapport PDF:

- Plus vous décrivez ce que vous avez fait dans votre rapport, plus je peux vous donner des points! Le rapport est aussi important, sinon plus, que le code. Prenez le temps de décrire ce que vous avez fait et pourquoi.
- Pour chaque étape, décrivez votre démarche et décrivez l'organisation de votre code.
- Décrivez comment lancer votre code. Ex : Écrivez en langage bash  
`>> python mon_code.py argument1 argument`
- S'il y a beaucoup de lignes de code, vous pouvez aussi fournir un fichier .sh

Codez en python:

- Votre script (main) devrait contenir très peu de code et appeler les sous-fonctions que vous aurez créées.
- Utilisez un *argparser*. Le nom des fichiers (ex, path des images) devrait être envoyé en paramètre, et non être écrits directement dans le code.
- Python est très lent pour les boucles «for» imbriquées. Tentez de remplacer les boucles par des calculs sur les matrices.

**La date de remise du TP sera déterminée en classe.**

## 1. [30 pts]

a) Créez un script ***view\_image.py*** qui devra lancer un “viewer”, un visualisateur interactif qui permet d’afficher une image.

Option 1 : Votre viewer peut prendre en argument la vue choisie (axiale, coronale, sagittale) et afficher cet axe.

Option 2 : Votre viewer peut présenter 3 images à la fois; une par axe.

Autres détails :

- Vous devez ajouter un « slider » sous l’axe choisi (option 1) ou sous chaque axe (option 2).
- Votre fonction doit savoir gérer des images 2D/3D/4D.  
Indice: Vous devez choisir quoi faire avec la 4e dimension!
- Vous n’avez qu’à supporter le format d’image NIFTI.  
Indice : vous pouvez utiliser *nibabel* pour la gestion d’images (ex, *nib.load*).
- Vous pouvez utiliser *matplotlib.pyplot* pour l’affichage, mais ce n’est qu’une suggestion.
- Les images doivent être affichées “à l’endroit”. Indices: vous pourriez avoir besoin de transposer les images (.T) avant de les afficher, ou jouer avec le paramètre “origin” de *matplotlib*.
- Les images doivent être réalistement visibles. Indice: jouer avec le paramètre “aspect” de *matplotlib*.

**\*\* Je ne m’attends pas à un viewer de professionnel, juste une petite fonction qui vous permet de balayer vos différentes coupes et vos différents jeux de données. Ne prenez pas trop de temps sur cette question!**

\*\* Je m’attends à une section écrite dans le rapport. Expliquez vos choix.

b) Créez un script ***compute\_mIP\_MIP.py*** qui calcule le maximum et minimum intensity projection le long d’un axe. L’axe choisi, et l’option (min ou max) doivent être des arguments du script.

Indice: Utilisez votre viewer pour déterminer les paramètres intéressants! Justifiez dans votre rapport.

c) Indiquez dans votre rapport l’orientation de vos images (LPI / RAS / etc) sur disque (ex, *mrinfo*). Vérifiez l’orientation de l’image une fois « loadée » avec *nibabel*?

2. [40 pts] Pour les images TDM (i.e. CT-scans), IRM (flair, T1, coeur), et ultrason fournis, répondez aux questions i - viii ci-dessous.

- Créez un script ***show\_image\_stats.py*** qui affiche les informations nécessaires (question ii obligatoire. Autres questions facultatives). Elle doit prendre un nom d'image en entrée.
- Décrivez votre démarche pour chaque section dans votre rapport.
- Compléter les tableaux ci-dessous avec vos réponses.

	i) Voxels= Image=	ii) M = RMS =	iii)	iv)
Epaule				
rat111				
Flair				
Heart				
T1				
us				

	v)	vi)	vii)	viii)
Epaule				
rat111				
Flair				
Heart				
T1				
us				

- Quelle est la taille des voxels (résolution) et la taille de chaque image? (Donnez les unités!) (Utilisez *mrinfo*, *3dinfo*, le header de *nibabel*, etc.)
- Calculez le contraste de Michelson et RMS de chaque image? (Codez vos fonctions vous-mêmes).
- Quelle est la taille de la plus petite structure détectable dans l'image? Vous n'êtes pas des médecins mais creusez un peu, tentez de comprendre ce que vous voyez. (Aucun code nécessaire; vous pouvez utiliser Fiji ou Mi-Brain)
- Voyez-vous des effets de volume partiel? Si oui, où?
- Quelle est la nature du bruit dans chaque image? (Justifiez)
- Le bruit semble-t-il uniforme (i.e pareil partout) dans l'image? (Justifiez)
- Y-a-t-il des artefacts dans les images? (Vous n'êtes pas spécialistes mais creusez un peu.)
- Quel est le ou les SNR dans ces images? (Quel est votre calcul?)

3. [30 pts] Créez un script ***denoise\_image.py***. Pour l'image IRM Flair et l'image T1, roulez NLMEANS et choisissez 2 autres méthodes de débruitage vues en classe pour comparer.

- a. Pourquoi ce choix de méthode? Votre choix devrait dépendre de certaines caractéristiques dans les images.
- b. Pour chaque méthode, décrivez les paramètres utilisés et justifiez vos choix.
- c. Discutez des avantages et limites de chaque approche.
- d. Quelle méthode est la meilleure pour chaque image? Justifiez.

Vous avez le choix de prendre...

- i. Filtrage isotrope (diffusion linéaire)
- ii. Filtrage anisotrope (diffusion non linéaire)
- iii. Filtrage médian
- iv. Filtrage bilatéral
- v. Filtrage moyen non-local (non-local means)
- vi. Block matching
- vii. Autres techniques avancées si vous voulez...

*\*\*Ces techniques sont très connues et il existe des filtres python, des plugins ImageJ et des fonctions Matlab pour tous ces filtres. Je vous laisse la liberté de choisir et de prendre ce que vous voulez. Justifiez vos choix et montrez-moi que vous faites plus que rouler des boîtes noires.*