

ÉCOLE POLYTECHNIQUE DE L'UNIVERSITÉ FRANÇOIS RABELAIS DE TOURS

Spécialité Informatique

64 av. Jean Portalis

37200 TOURS, FRANCE

Tél +33 (0)2 47 36 14 31

[www.polytech.univ-tours.fr](http://www.polytech.univ-tours.fr)

## CAHIER DE SPECIFICATIONS

Projet : RFAI14		Développement de techniques de segmentation automatique d'images IRM 3D de cerveaux animaux.		
Emetteur :		Léo Boulanger	Coordonnées : leo.boulanger-2@etu.univ-tours.fr	
Encadrants :		Jean-Yves Ramel	Coordonnées : jean-yves.ramel@univ-tours.fr	
		Antoine Bourlier	Coordonnées : antoine.bourlier@univ-tours.fr	
Date d'émission :		Mardi 12 décembre 2023		
Validation				
Nom		Date	Valide (O/N)	Commentaires
Historique des modifications				
Version	Date	Description de la modification		
00	12/12/2023	Version initiale		

## TABLE OF CONTENTS

<b>CAHIER DE SPECIFICATIONS</b>	<b>3</b>
1. INTRODUCTION	3
2. CONTEXTE DE LA REALISATION	3
2.1. <i>Contexte</i>	3
2.2. <i>Objectifs</i>	3
3. DESCRIPTION GENERALE	4
3.1. <i>Environnement du projet</i>	4
3.2. <i>Caractéristiques des utilisateurs</i>	4
3.3. <i>Fonctionnalités du système</i>	4
3.4. <i>Structure générale du système</i>	5
4. DESCRIPTION DES INTERFACES EXTERNES DU LOGICIEL	5
4.1. <i>Interfaces matériel/logiciel</i>	5
4.2. <i>Interfaces homme/machine</i>	5
4.3. <i>Interfaces logiciel/logiciel</i>	5
5. SPECIFICATIONS FONCTIONNELLES	6
5.1. <i>Fonctions de traitement de données</i>	6
5.1.1. Définition de la fonction 1.1 : import_mri_data	6
5.1.2. Définition de la fonction 1.2 : load_settings	6
5.1.3. Définition de la fonction 1.3 : setup	7
5.2. <i>Fonctions de clustering</i>	8
5.2.1. Définition de la fonction 2.1 : start_process	8
5.2.2. Définition de la fonction 2.2 : local_membership	10
5.2.3. Définition de la fonction 2.3 : global_membership	11
5.2.4. Définition de la fonction 2.4 : combined_membership	12
5.2.5. Définition de la fonction 2.5 : objective_function	13
5.2.6. Définition de la fonction 2.6 : compute_new_cluster	14
5.2.7. Définition de la fonction 2.7 : histogram_peak_analysis	15
5.2.8. Définition de la fonction 2.8 : subsegment	15
5.3. <i>Fonctions utilitaires</i>	16
5.3.1. Définition de la fonction 3.1 : save_progress	16
5.3.2. Définition de la fonction 3.2 : load_progress	16
5.3.3. Définition de la fonction 3.3 : append_log_file	17
5.3.4. Définition de la fonction 3.4 : update_log_key	17
5.3.5. Définition de la fonction 3.5 : remove_log_key	17
6. SPECIFICATIONS NON FONCTIONNELLES	18
6.1. <i>Contraintes de développement et conception</i>	18
6.2. <i>Contraintes de fonctionnement et d'exploitation</i>	18
6.2.1. Performances	18
6.2.2. Contrôlabilité	18
6.2.3. Sécurité	18
6.2.4. Intégrité	18
<b>GLOSSAIRE</b>	<b>19</b>
<b>BIBLIOGRAPHIE</b>	<b>20</b>

## CAHIER DE SPECIFICATIONS

### 1. Introduction

Ce document définit les spécifications concernant le projet de fin d'étude « Développement de techniques de segmentation automatique d'images IRM 3D de cerveaux animaux ». Ce projet s'intègre dans une collaboration entre le Laboratoire d'informatique fondamentale et appliquée de Tours (LIFAT) et l'Institut national de recherche pour l'agriculture, l'alimentation et l'environnement (INRAE).

Ce cahier des spécifications est rédigé par Léo Boulanger, étudiant à Polytech Tours, représentant le MOE. Le projet sera encadré par Antoine Bourlier et Jean-Yves Ramel, représentant le MOA.

### 2. Contexte de la réalisation

#### 2.1. Contexte

L'avancement des technologies d'imagerie médicale, en particulier l'Imagerie par Résonance Magnétique (IRM), nous permet de comprendre de façon approfondie les structures cérébrales des animaux. La segmentation manuelle des structures cérébrales sur des images de cerveaux animaux représente actuellement une tâche laborieuse et chronophage pour les chercheurs. La combinaison de la complexité morphologique des cerveaux avec la variabilité inter-espèce et interindividuelle rend ce processus particulièrement exigeant, et peut également introduire des biais, causés par les experts effectuant les analyses. Afin de palier ces défis, les chercheurs aspirent à mettre au point des méthodes de segmentation automatiques, permettant, non seulement de réduire significativement le temps de l'analyse des données, mais aussi d'améliorer la cohérence et la précision des résultats obtenus.

#### 2.2. Objectifs

L'objectif de ce projet est de développer un programme informatique permettant à un utilisateur de segmenter automatiquement une image IRM 3D. Ce programme sera basé principalement sur la méthode « Unsupervised modified spatial fuzzy c-means » publiée par Kamarujjaman, et Mausumi Maitra<sup>[1]</sup>, pour segmenter les données en un nombre de clusters donné par l'utilisateur, ou défini automatiquement. Le programme devra être utilisable sur tous les systèmes d'exploitation pouvant exécuter des scripts Python, mais la validation des fonctionnalités de ce projet sera faite uniquement sur les systèmes Windows et Linux. Il sera donc préférable d'utiliser Windows ou Linux afin de réduire les risques de dysfonctionnement.

### 3. Description générale

#### 3.1. Environnement du projet

Ce projet est lié à la thèse d'Antoine Bourlier<sup>[2]</sup>, dont l'objectif est de proposer de nouveaux algorithmes d'analyse et de comparaison *anatomo-fonctionnel*. Certains de ces algorithmes utilisent la théorie des graphes pour modéliser les structures du cerveau, où les nœuds correspondent aux structures, et les arêtes aux connexions anatomiques ou fonctionnelles. Cependant, il est nécessaire d'avoir une segmentation pour modéliser les structures en nœuds.

#### 3.2. Caractéristiques des utilisateurs

Les principaux utilisateurs sont les chercheurs spécialisés en *neurologie*, mais l'utilisation du programme reste accessible pour les personnes travaillant dans le domaine de l'informatique et de la *neuroscience*.

Le programme est donc développé pour des utilisateurs qui :

- ont peu de connaissances en informatique
- n'ont pas d'expérience avec ce genre d'outils
- se serviront occasionnellement de cet outil
- ne nécessitent pas de privilèges spécifiques dans leur système

#### 3.3. Fonctionnalités du système

Le programme peut être chargé de 3 façons :

- L'utilisateur exécute le programme en incluant des arguments, puis chaque configuration manquante lui est demandé via l'invite de commandes.  
Exemple : « /python segmentation\_script.py path\to\mri\_data.nii.gz -n 20 » appliquera la segmentation sur l'image IRM située à « path\to\mri\_data.nii.gz », pour 20 structures.
- L'utilisateur peut aussi inclure un fichier de configuration en argument, qui appliquera les configurations y étant définies. Les paramètres passés en argument écraseront ceux importés depuis le fichier de configuration.
- L'utilisateur peut aussi lancer la reprise de l'exécution en passant uniquement un fichier de sauvegarde au programme.

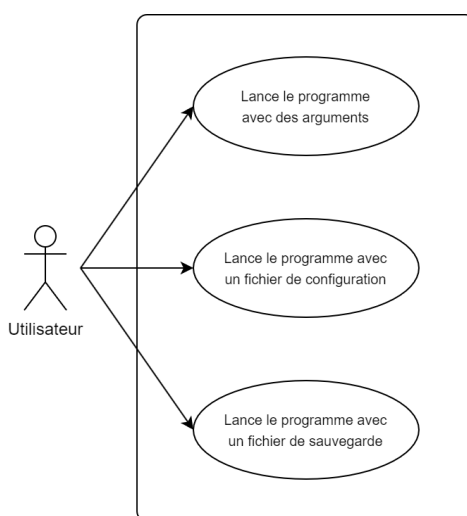


Figure 1 - Diagramme de cas d'utilisation général du programme

### 3.4. Structure générale du système

Le code du projet sera séparé en 3 fichiers Python :

- configuration.py
- 3D\_umsfcm.py
- logger.py

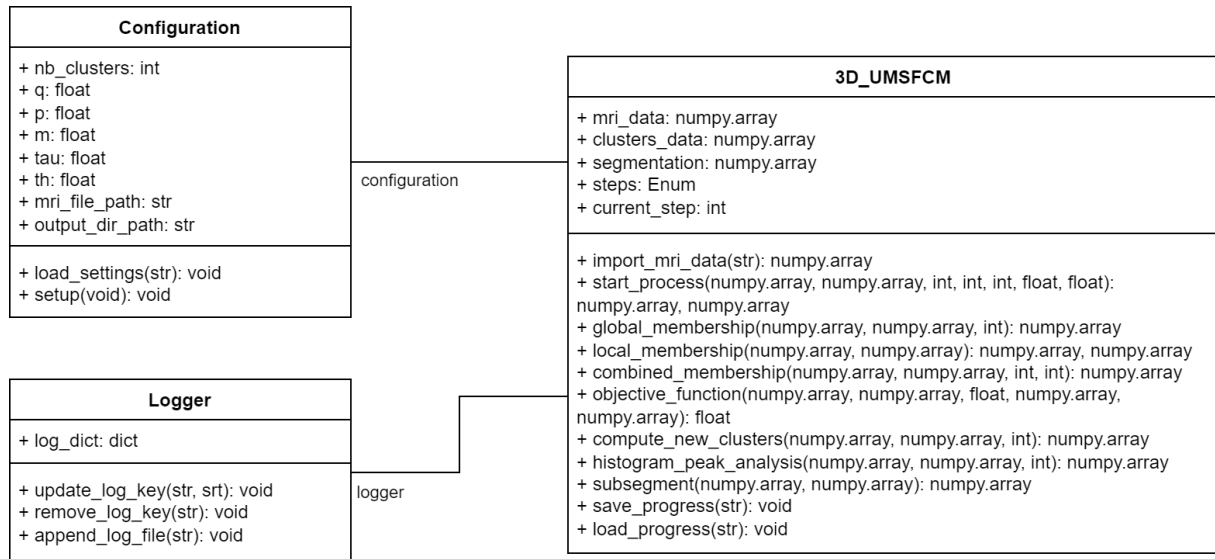


Figure 2 - Diagramme de classes



## 4. Description des interfaces externes du logiciel

### 4.1. Interfaces matériel/logiciel

Il n'y a pas d'interface matériel/logiciel spécifiquement défini pour ce projet.

### 4.2. Interfaces homme/machine

Le programme sera utilisé via console, aucune interface homme/machine ne sera fournie.

### 4.3. Interfaces logiciel/logiciel


Il n'y a pas d'interface logiciel/logiciel défini pour ce projet.

## 5. Spécifications fonctionnelles


### 5.1. Fonctions de traitement de données

#### 5.1.1. Définition de la fonction 1.1 : import\_mri\_data

Identification :


- Nom : import\_mri\_data
- Cette fonction importe les données d'un fichier NIfTI pour les rendre exploitables par le reste du programme.
- Priorité : primordiale 

Description :

- Entrées 
  - file\_path : Chemin vers le fichier NIfTI à importer.
- Sorties :
  - mri\_data : Données exploitables de l'image IRM importée
- Gestion des erreurs :
  - Le fichier est introuvable : Lever une erreur indiquant le chemin du fichier pour que l'utilisateur puisse vérifier et corriger l'emplacement du fichier.
  - Une erreur lors de la lecture du fichier : Lever une erreur indiquant à l'utilisateur que le fichier est illisible ou corrompu.

#### 5.1.2. Définition de la fonction 1.2 : load\_settings

Identification :

- Nom : load\_settings
- Cette fonction permettra de charger une configuration du programme à partir d'un fichier 
- Priorité : optionnelle

Description :

- Entrées :
  - file\_path : Chemin vers le fichier de configuration.
- Sorties :
  - Les paramètres de la configuration seront appliqués à l'objet Configuration appelant cette fonction, il n'y a donc pas de données en sortie.
- Gestion des erreurs :
  - Le fichier est introuvable : Lever une erreur indiquant le chemin du fichier pour que l'utilisateur puisse vérifier et corriger l'emplacement du fichier.
  - Une erreur lors de la lecture du fichier : Lever une erreur indiquant à l'utilisateur que le fichier est illisible ou corrompu.

### 5.1.3. Définition de la fonction 1.3 : setup

#### Identification :

- Nom : setup
- Cette fonction ajoute les configurations passées en paramètres de l'application, puis demande à l'utilisateur via la console d'ajouter les paramètres manquants.
- Priorité : secondaire

#### Description :

- Entrées :
  - Aucune donnée n'est nécessaire en entrée
- Sorties :
  - Les paramètres de la configuration seront appliqués à l'objet Configuration appelant cette fonction, il n'y a donc pas de données en sortie.
- Gestion des erreurs :
  - L'ensemble des paramètres nécessaires ne sont pas affectés à la fin de la fonction : Lever une erreur indiquant à l'utilisateur les paramètres nécessaires indéfinis.


## 5.2. Fonctions de clustering

### 5.2.1. Définition de la fonction 2.1 : start\_process

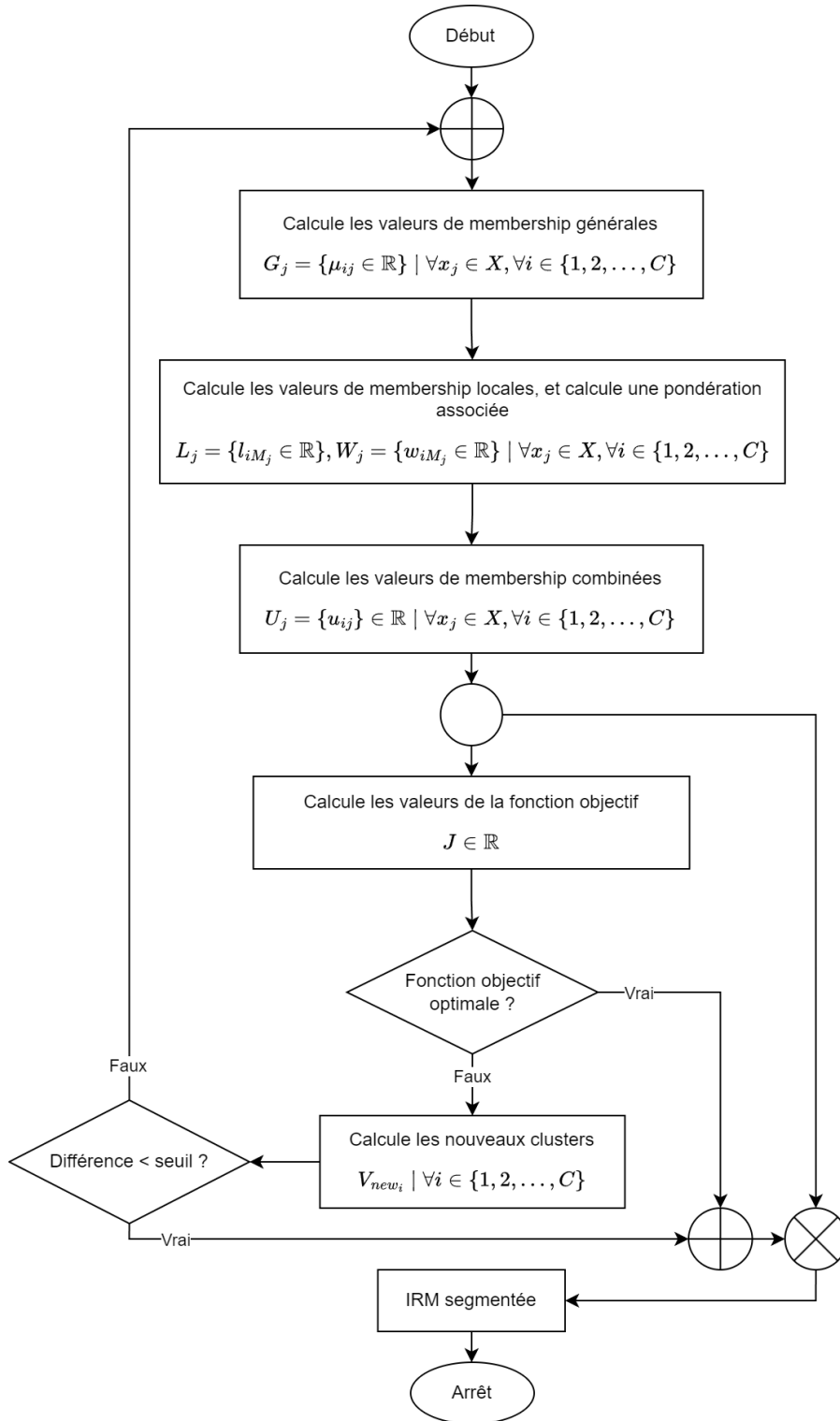
Identification :

- Nom : start\_process
- Cette fonction initialise les éléments, puis utilise les fonctions nécessaires pour appliquer la méthode *3D unsupervised modified spatial fuzzy c-means* sur les données.
- Priorité : primordiale

Description :


- Entrées :
  - mri\_data : matrice 3D contenant les données à segmenter
  - cluster\_values : liste des valeurs des centres de cluster
  - p : paramètre de contrôle d'importance du membership global
  - q : paramètre de contrôle d'importance du membership local
  - m : le « fuzzifier », valeur utilisée pour définir un degré d'incertitude appliqué dans la fonction de calcul du membership local
  - threshold : seuil à partir duquel la différence entre les valeurs des clusters, avant et après calcul des nouveaux clusters, est considéré comme trop faible, impliquant l'arrêt de la boucle d'entraînement
  -  le taux d'inclusion des informations spatiales dans la fonction objectif
- Sorties :
  - La liste des valeurs de chaque centre de cluster
  - La matrice des voxels segmentés
- Gestion des erreurs :
  - Une erreur est survenue lors de l'exécution : Lever une erreur indiquant à quelle étape elle a eu lieu, puis indiquer le chemin vers la dernière sauvegarde.
- Flowchart :





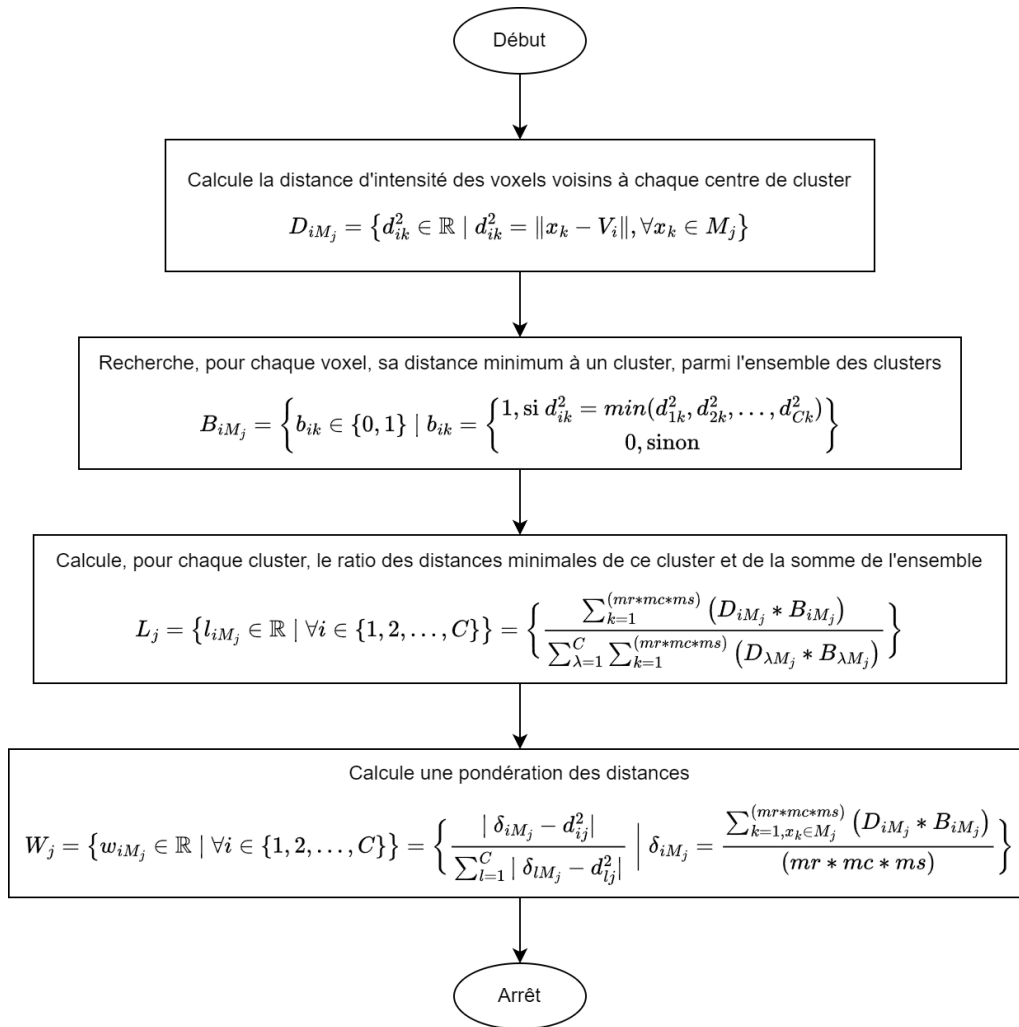
### 5.2.2. Définition de la fonction 2.2 : local\_membership

Identification :

- Nom : local\_membership
- Cette fonction calcule les valeurs de membership locales d'un voxel pour chaque centre de cluster, et calcule une pondération pour optimiser la fonction objectif.
- Priorité : primordial 

Description :

- Entrées :
  - mask\_data : matrice des données locales du voxel analysé
  - cluster\_values : valeurs des centres de clusters
- Sorties :
  - Une liste des valeurs de membership locales
  - Une liste des valeurs de la pondération
- Gestion des erreurs :
  - Erreur lors de l'exécution de la fonction : Lever une erreur indiquant qu'une erreur est survenue dans la fonction local\_membership
- Flowchart :



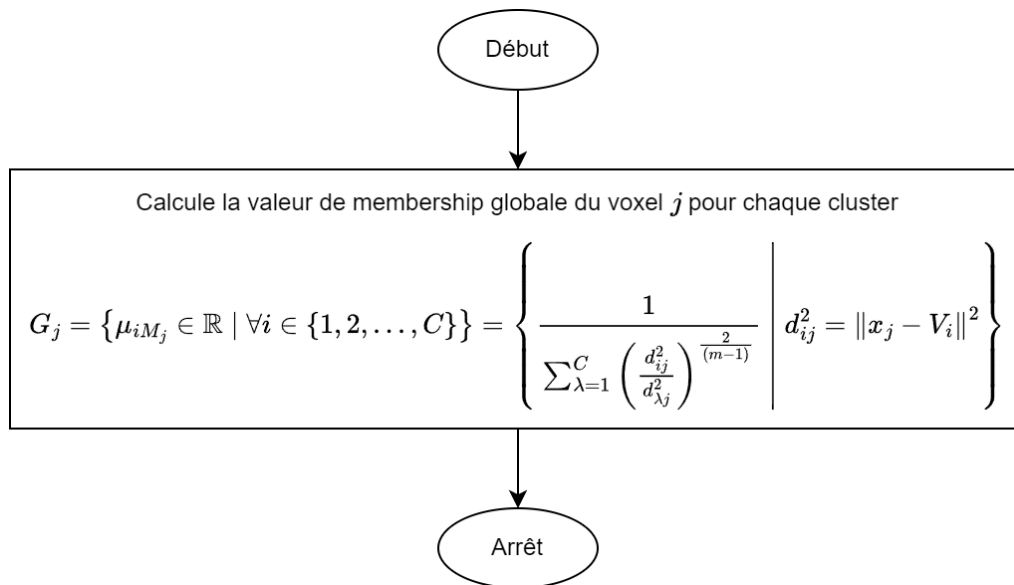
### 5.2.3. Définition de la fonction 2.3 : global\_membership

Identification :

- Nom : global\_membership
- Cette fonction calcule les valeurs de membership générale d'un voxel pour un centre de cluster.
- Priorité : prioritaire

Description :

- Entrées :
  - mri\_data : matrice des données
  - cluster\_values : valeurs des centres de clusters
  - m : valeur du fuzzifier
- Sorties :
  - Une liste des valeurs de membership générales
- Gestion des erreurs :
  - Erreur lors de l'exécution de la fonction : Lever une erreur indiquant qu'une erreur est survenue dans la fonction global\_membership.
- Flowchart :



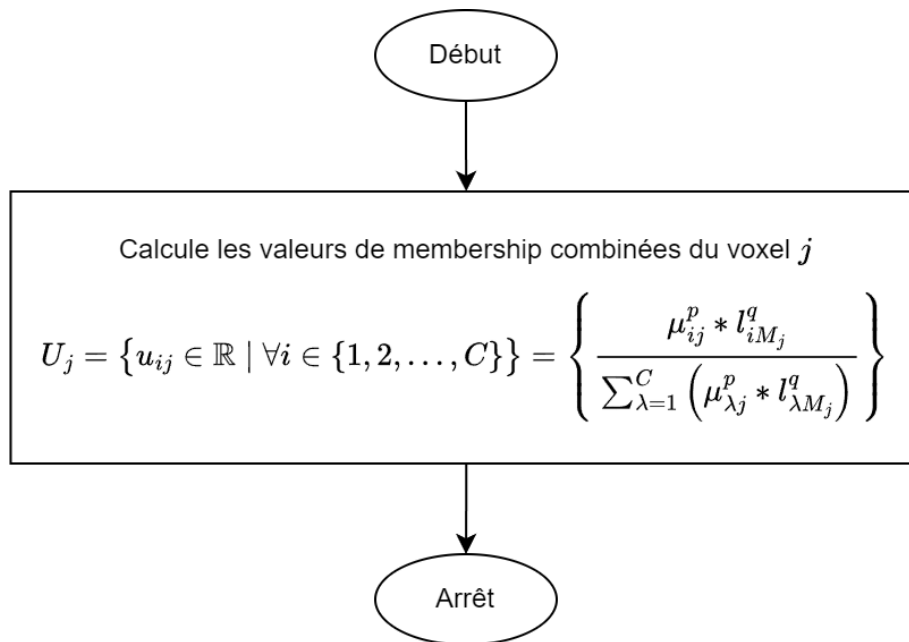
#### 5.2.4. Définition de la fonction 2.4 : combined\_membership

Identification :

- Nom : combined\_membership
- Cette fonction calcule les valeurs de membership d'un voxel, obtenues en combinant les valeurs de membership locales et générales.
- Priorité : primordiale

Description :

- Entrées :
  - local\_membership : liste des valeurs de membership locales
  - global\_membership : liste des valeurs de membership globales
  - p : paramètre de contrôle d'importance du membership global
  - q : paramètre de contrôle d'importance du membership local
- Sorties :
  - Une liste des valeurs de membership combinées
- Gestion des erreurs :
  - Erreur lors de l'exécution de la fonction : Lever une erreur indiquant qu'une erreur est survenue dans la fonction combined\_membership.
- Flowchart :




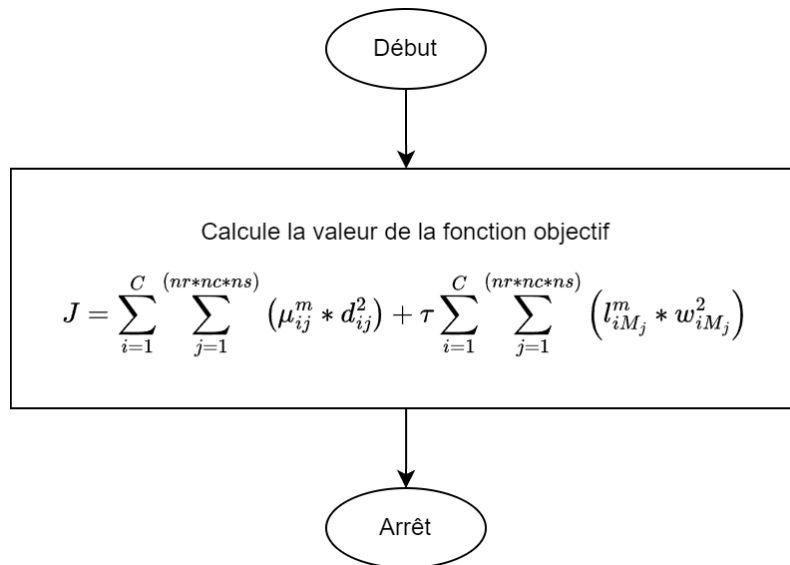
### 5.2.5. Définition de la fonction 2.5 : objective\_function

Identification :

- Nom : objective\_function
- Fonction objectif utilisée pour rechercher l'optimalité de la solution.
- Priorité : primordiale

Description :

- Entrées :
  - global\_membership : matrice des valeurs de membership générales
  - cluster\_values : liste des valeurs des clusters
  -  le taux d'inclusion des informations spatiales dans la fonction objectif
  - local\_membership : matrice des valeurs de membership locales
  - weighted\_data : matrice des valeurs pondérées
- Sorties :
  - La valeur de la fonction objectif
- Gestion des erreurs :
  - Erreur lors de l'exécution de la fonction : Lever une erreur indiquant qu'une erreur est survenue dans la fonction objective\_function.
- Flowchart :



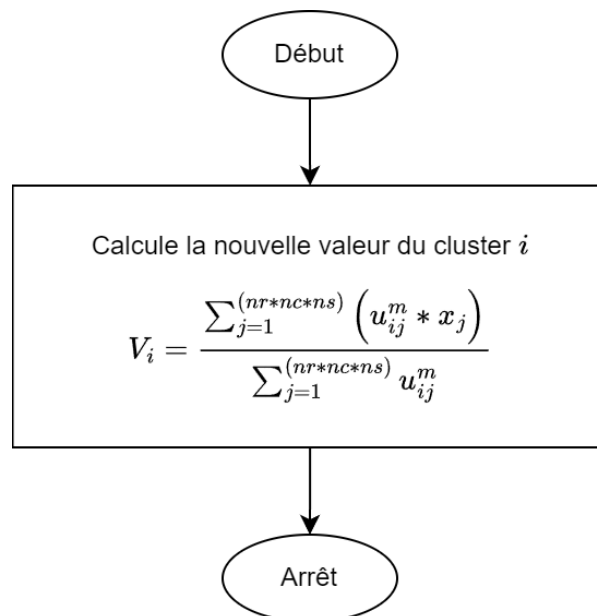
### 5.2.6. Définition de la fonction 2.6 : compute\_new\_cluster

Identification :

- Nom : compute\_new\_cluster
- Cette fonction utilisera une méthode d'analyse de pics d'histogramme<sup>[3]</sup> pour obtenir automatiquement **clusters**.
- Priorité : **priorité** prioritaire

Description :

- Entrées :
  - nb\_clusters : le nombre de clusters recherchés
- Sorties :
  - Une liste de valeurs de clusters
- Gestion des erreurs :
  - Erreur lors de l'exécution de la fonction : Lever une erreur indiquant qu'une erreur est survenue dans la fonction compute\_new\_cluster.
- Flowchart :



### 5.2.7. Définition de la fonction 2.7 : histogram\_peak\_analysis

Identification :

- Nom : histogram\_peak\_analysis
- Cette fonction utilisera une méthode d'analyse de pics d'histogramme pour obtenir automatiquement un nombre prédéfini, ou non de clusters.
- Priorité : optionnelle

Description :

- Entrées :
  - mri\_data : la matrice des données de l'image IRM
  - nb\_clusters : le nombre de clusters recherchés. Si « 0 », le nombre de clusters sera recherché automatiquement.
- Sorties :
  - Une liste contenant les valeurs des clusters trouvés
- Gestion des erreurs :
  - Erreur lors de l'exécution de la fonction : Lever une erreur indiquant qu'une erreur est survenue dans la fonction histogram\_peak\_analysis.

### 5.2.8. Définition de la fonction 2.8 : subsegment

Identification :

- Nom : subsegment
- Cette fonction applique un masque binaire sur une image IRM, pour isoler une partie des données.
- Priorité : optionnelle

Description :

- Entrées :
  - mri\_data : matrice des données de l'image IRM
  - mask : matrice binaire utilisée pour isoler une partie des données, doit être de la même taille que mri\_data
- Sorties :
  - La matrice des données de l'image IRM associée au masque binaire fourni.
- Gestion des erreurs :
  - Le masque n'a pas la même taille que l'image : Lever une erreur indiquant à l'utilisateur les tailles des 2 matrices.

### 5.3. Fonctions utilitaires

#### 5.3.1. Définition de la fonction 3.1 : save\_progress

Identification :

- Nom : save\_progress
- Sauvegarde l'état et l'avancée de la segmentation.
- Priorité : secondaire

Description :

- Entrées :
  - save\_path : chemin vers le fichier où sauvegarder la progression
- Sorties :
  - Aucune sortie n'est nécessaire
- Gestion des erreurs :
  - Le fichier ne peut pas être créé ou modifié : Lever une erreur informant l'utilisateur qu'une erreur est survenue lors de l'interaction avec le fichier.

#### 5.3.2. Définition de la fonction 3.2 : load\_progress

Identification :

- Nom : load\_progress
- Reprend l'exécution du programme à partir d'un fichier de sauvegarde.
- Priorité : secondaire

Description :

- Entrées :
  - load\_path : chemin vers le fichier de sauvegarde à lire
- Sorties :
  - Aucune sortie n'est nécessaire
- Gestion des erreurs :
  - Le fichier est introuvable : Lever une erreur informant l'utilisateur que le fichier est introuvable, et afficher le chemin vers le fichier attendu.
  - Le fichier est illisible : Lever une erreur indiquant à l'utilisateur que le fichier ne peut pas être lu.



### **5.3.3. Définition de la fonction 3.3 : append\_log\_file**

Identification :

- Nom : append\_log\_file
- Ajoute une information dans le fichier de logs
- Priorité : secondaire

Description :

- Entrées :
  - log\_info : chaîne de caractères à ajouter dans le fichier de logs
- Sorties :
  - Aucune sortie n'est nécessaire
- Gestion des erreurs :
  - Le fichier de logs ne peut pas être créé ou modifié : Lever une erreur indiquant à l'utilisateur qu'il est impossible d'interagir avec le fichier de logs

### **5.3.4. Définition de la fonction 3.4 : update\_log\_key**

Identification :

- Nom : update\_log\_key
- Ajoute ou modifie un couple clef / valeur dans le dictionnaire de logs.
- Priorité : secondaire

Description :

- Entrées :
  - key : nom de la clef
  - value : description de la clef
- Sorties :
  - Aucune sortie n'est nécessaire.
- Gestion des erreurs :
  - Erreur lors de l'exécution de la fonction : Lever une erreur indiquant qu'une erreur est survenue dans la fonction update\_log\_key.

### **5.3.5. Définition de la fonction 3.5 : remove\_log\_key**

Identification :

- Nom : remove\_log\_key
- Supprime une clef du dictionnaire de logs
- Priorité : secondaire

Description :

- Entrées :
  - key : nom de la clef à supprimer
- Sorties :
  - Aucune sortie n'est nécessaire
- Gestion des erreurs :
  - Erreur lors de l'exécution de la fonction : Lever une erreur indiquant qu'une erreur est survenue dans la fonction remove\_log\_key.

## 6. Spécifications non fonctionnelles

### 6.1. Contraintes de développement et conception

- Python est imposé comme langage de programmation.
- Le code et la documentation devront être rédigés en anglais.
- Les données des images IRM seront au format NIfTI (.nii / .nii.gz).
- Le programme doit être utilisable sur les systèmes d'exploitation Windows et Linux.

### 6.2. Contraintes de fonctionnement et d'exploitation

Le programme devant être exécutable sous Windows et Linux, la manipulation de fichiers devra être compatible pour les différentes syntaxes : séparateurs ( « / » pour Linux, « \ » pour Windows), et prise en compte des jokers ( « \* », « ? », « [] », etc...).

#### 6.2.1. Performances

L'exécution du programme doit être complétée de préférence en moins de 10 minutes. Cependant, une durée d'exécution de quelques heures est acceptable pour l'analyse d'images IRM de très grandes dimensions.

#### 6.2.2. Contrôlabilité

Un système de fichiers de log permettra de suivre l'état de l'algorithme le long de son exécution. De plus, à chaque itération de l'entraînement du clustering, l'état du programme est sauvegardé pour reprendre la progression en cas d'erreur.

#### 6.2.3. Sécurité

Le système ne nécessitera pas d'élévation des privilèges pour fonctionner, ni de quelconque information d'identification.

#### 6.2.4. Intégrité

Dans le cas de coupure lors de l'exécution, le système récupérera les données sauvegardées lors de la dernière étape du processus. Les données de l'étape où l'erreur a eu lieu seront perdues.

## GLOSSAIRE

Anatomo-fonctionnel : Relatif à l'anatomie et aux fonctions physiologiques.

Cluster : Groupe d'éléments. Dans le contexte de ce projet, un cluster regroupe un ensemble de voxels étant plus proche (en intensité) du centre de ce cluster que des autres.

Membership : Mot anglais signifiant « Adhésion ». Dans le contexte de ce projet, la valeur de membership d'un voxel à un centre de cluster correspond à l'intensité à laquelle le voxel « adhère » au cluster.

Neurologie : Spécialité de la médecine qui s'intéresse au système nerveux et à ses pathologies.

Neuroscience : Discipline scientifique étudiant le système nerveux et la cognition.

Voxel : Mot-valise contractant « pixel » et « volume », correspond à un élément d'un volume. Dans le contexte de ce projet, un voxel correspond à un point de l'image IRM 3D analysée.

**BIBLIOGRAPHIE**

- [1] Kamarujjaman, Maitra, M. 3D unsupervised modified spatial fuzzy c-means method for segmentation of 3D brain MR image. *Pattern Anal Applic* **22**, 1561–1571 (2019). <https://doi.org/10.1007/s10044-019-00806-2>
- [2] Antoine Bourlier, Apprentissage profond sur graphes pour l'analyse et la comparaison morphofonctionnelle d'encéphales. <https://www.theses.fr/s350453>
- [3] Namburu, Anupama & Kumar, S & Edara, Sreenivasa. (2017). Generalized rough intuitionistic fuzzy c-means for MR brain image segmentation. IET Image Processing. 11. 10.1049/iet-ipr.2016.0891.