Performance Analysis of Clustering Algorithms in Brain Tumor Detection from PET Images

Master’s thesis

Faculty of Medicine, University of Turku

Department of Biosciences/Cell Biology,

Åbo Akademi University

30.05.2023

Master’s degree of Biomedical Imaging

Credits: 20 ECTS

Supervisors:

1. Maria Jaakkola (PhD)

2. Riku Klén (Assistant Professor)

3. Xiang-Guo Li (Assistant Professor)

Examiners:

1. ……………...

2. …..…………

UNIVERSITY OF TURKU
Institute of Biomedicine
Faculty of Medicine

Anting Li: Performance Analysis of Clustering Algorithms in Brain Tumor Detection from PET Images

Master´s thesis, 48 pp., 10 Appendices
Specialization level
30.05.2023

Brain metastases remain fatal and challenging, and their early detection is imperative. With the advancement in non-invasive imaging techniques, positron emission tomography, as a functional imaging, has been widely employed in oncological studies, including pathophysiological mechanisms of the tumors. While manual analysis and integration of dynamic 4D PET images are challenging and inefficient. Therefore, automated segmentation is adopted to improve the efficiency and accuracy. In recent years, clustering-based image segmentation has been gaining popularity in detecting tumors. This thesis applies three clustering-based algorithms to automatically identify and segment metastatic brain tumors from dynamic 4D PET images of mice. The clustering algorithms used include K-means and Gaussian mixture model clustering in combination with pre-processing principal component analysis, independent component analysis and post-processing connected component analysis. The performances of three clustering algorithms in execution time and accuracy were evaluated by the Jaccard index and validated by time activity curve. The results indicate that K-means clustering is the best-performing among the three clustering methods when combined with independent component analysis, and the post-processing method connected component analysis has significantly improved the performance of K-means clustering.

KEYWORDS**:** brain metastases, metastatic brain tumor detection, automated segmentation, clustering algorithms, PET imaging

LIST OF USED ABBREVIATIONS

[^{11}C]methionine	L-[*methyl*-^{11}C]methionine
[^{18}F]FDG	2-deoxy-2-[^{18}F]fluoro-D-glucose
[^{68}Ga]Ga-DOTA	^{68}Ga-labeled1,4,7,10-tetraazacyclododecane-1,4,7,10-tetraacetic acid
BBB	Blood-brain barrier
BM	Brain metastases
CCA	Connected Component Analysis
CNS	Central nervous system
CT	Computed tomography
DICOM	Digital Imaging and Communications in Medicine standard
FCM	Fuzzy C-Means Clustering
GMM	Gaussian Mixture Model
ICA	Independent Component Analysis
MRI	Magnetic resonance imaging
NIfTI	Neuroimaging Informatics Technology Initiative
NMI	Normalized Mutual Information
PCA	Principal Component Analysis
PET	Positron emission tomography
ROI	Region of interest
TAC	Time-activity curve

Table of Contents

List of used abbreviations III

1 Introduction 1

1.1 Brain Metastases 1

1.1.1 Sources of brain metastases 1

1.1.2 Classification of brain metastases 2

1.1.3 Symptoms of brain metastases 3

1.1.4 Treatment of brain metastases 3

1.2 Diagnosis of Brain Metastases 4

1.2.1 PET imaging 5

1.2.2 Static and dynamic PET imaging 5

1.2.3 Radiotracers of PET imaging 6

1.3 Image Processing and Analysis 8

1.3.1 Pre-processing 8

1.3.2 Processing 9

1.3.3 Post-processing 11

1.4 Evaluation and Validation of Clustering 11

2 The most important literature 12

3 Aims and hypotheses 14

4 Materials and methods 15

4.1 Material 15

4.1.1 PET images 15

4.1.2 Data format 16

4.2 Manual Segmentation 17

4.3 Automated Segmentation 20

4.3.1 Pre-processing 22

4.3.2 Clustering 23

4.3.3 Post-processing 24

4.4 Evaluation 25

4.4.1 Execution Time 25

4.4.2 Jaccard Index 26

4.4.3 TACs of manual segmentation and automated segmentation 26

4.4.4 Statistical analysis 27

5 Results 28

5.1 Visualization Inspection of Clustering Results 28

5.2 Execution Time 29

5.2.1 Average execution time prior to CCA ... 29
5.2.2 Average execution time after CCA ... 29
5.2.3 Statistical analysis of execution time of different clustering methods ... 30

5.3 Accuracy ... 31
5.3.1 Jaccard Index ... 31
5.3.2 Visualization of manual segmentation and cluster with max Jaccard index ... 33

5.4 Time Activity Curve ... 34
5.4.1 TACs of manual segmentation and automated segmentation ... 34

6 Discussion ... 39

7 Conclusion ... 41

8 Acknowledgments ... 42

10 References ... 43

11 Appendix ... 49

11.1 Automated segmentation pipeline ... 49

11.2 Calculation and visualization of Jaccard index ... 54

11.3 Acquisition of TACs ... 58

1 INTRODUCTION

More than 120 types of intracranial tumors that range from grade 1 to grade 4 were identified by the World Health Organization in 2021 (Louis et al., 2021), and they pose a significant threat to human life. Major brain tumors include gliomas, glioneuronal tumors, neuronal tumors, meningiomas and pineal tumors, and these types were then further divided into more than 120 subtypes or families (Louis et al., 2021). Among these brain tumors, metastatic brain tumors exhibit a higher incidence rate and mortality rate than benign gliomas. The incidence of metastatic brain tumors is four times more common than primary brain tumors based on American Association of Neurological Surgeons published in 2013. The survival time of patients diagnosed with metastatic brain tumors is less than 24 months (Achrol et al., 2019).

1.1 Brain Metastases

Unlike primary brain tumors originating from the brain, metastatic brain tumors, also known as brain metastases (BM), refer to intracranial tumors that metastasize to the central nervous system (CNS) from other body parts (Achrol et al., 2019). Metastatic brain tumors are usually malignant, and patients diagnosed with metastatic brain tumors generally present with widespread systemic malignancy (Achrol et al., 2019). This means these metastatic brain tumors proliferate rapidly and invade surrounding tissues, resulting in altering the brain's structure and can significantly impair normal brain function and even the whole human body in worse cases.

1.1.1 Sources of brain metastases

BM can be caused by any cancer, but the incidence rate varies by the condition of the patient and the cancer type (Gavrilovic & Posner, 2005). Among these resources, lung cancer, breast cancer, renal cell carcinoma, and melanoma are leading causes of brain metastases (Achrol et al., 2019). The study conducted by Berghoff et al. compared the incidence rate, diagnosis time, and survival times of different sources of brain metastases. The study demonstrates that lung cancer constitutes the largest proportion of BM with an incidence rate of 43.2% and followed by breast cancer, melanoma, and renal cell carcinoma with an incidence rate of 15.7%, 16.4%, and 9.1%, respectively (Barnholtz-Sloan et al., 2004). For patients diagnosed with stage IV breast cancer, more than 10% of them would develop brain metastases, whereas the actual incidence rate might exceed 10% as autopsy results indicate that 30% of these patients are detected BM (Lin et al., 2004). The incidence rate of BM in aggressive breast cancer subtypes is higher for than normal breast cancer, such as receptor 2-positive and triple-negative breast cancer (Witzel et al., 2016).

1.1.2 Classification of brain metastases

BM could occur in the brain as a single metastasis or multiple metastases, and their causes and treatment also differ. Berghoff et al.'s study demonstrates that a singular metastatic brain tumor is the most frequent, with an incidence rate of 48.7% out of 2149 patients, and more than 3 metastatic brain tumors are least frequent, with an incidence rate 23.5% (Barnholtz-Sloan et al., 2004). Breast, colorectal, and renal cancer are major sources of a single metastasis, whereas lung cancer and melanoma are major contributors to multiple metastases (Achrol et al., 2019). A single metastasis is treated by operating surgeries or utilizing low-dose radiation, whereas these treatments have limited effectiveness in multiple metastases, thus more complicated systemic treatment, such as chemotherapy and immunotherapy are adopted in multiple metastases treatment (Linnert et al., 2012).

BM can also be categorized based on their location within the brain. The location brain tumors are influence by factors such as the blood flow and the size of blood vessels. As a result, 80% of the BM is found in cerebral hemisphere, and 15% is found in the cerebellum, and the rest 5% in the brain stem (Eichler & Loeffler, 2007).

1.1.3 Symptoms of brain metastases

BM typically manifest with various symptoms, and it is reported that more than two-thirds of patients with BM exhibit symptoms during the course of their illness (Patcheil, 1995). According to Johns Hopkins Medicine, metastatic brain tumors commonly present with a wide range of symptoms such as headaches, seizures, and sensory deficits. As the disease progresses, patients may also experience a deterioration of balance, memory, behavior, and personality. Additionally, sensory impairments, including visual and auditory disturbances, may arise in worse cases (Metastatic Brain Tumors, 2021).

1.1.4 Treatment of brain metastases

The treatment of BM is influenced by many factors, such as the size and location of tumors, primary tumor types, and general health condition of the patients. Generally, BM treatment involves multiple modalities, including surgical resection, radiation therapy, chemotherapy, targeted drug treatments, and immunotherapy. Surgical resection is an option for patients whose primary cancer is under control and metastatic brain tumors can be safely removed (Achrol et al., 2019). If metastatic brain tumors are in life-threatening sites, surgical resection is not an ideal choice. Radiation therapy involves using high-energy radiation, such as X-rays or gamma rays, to destroy cancer cells and prevent their growth. Radiation therapy is a good substitute for surgery when tumors are smaller than 3 cm in diameter and is advantageous in treating multi-metastatic tumors (Achrol et al., 2019). Conventional chemotherapy has limited effectiveness in treating BM because chemotherapeutic agents are incapable of penetrating the blood brain barrier (BBB) in the CNS (Watase et al., 2021). Targeted therapy is a novel form of chemotherapy that selectively use chemotherapeutic agents that have ability to cross BBB, and now is widely used in the treatment of BM (Di Lorenzo & Ahluwalia, 2017). In recent years, immunotherapy has also gained significant attention in treating BM. Immunotherapy is a cancer treatment that utilizes substances to stimulate the patient's immune system to destroy cancer cells (Di Giacomo et al., 2023). Several studies have demonstrated immunotherapy can effectively stimulate immune response of patients, particularly BM patients. BM patients benefit more from immunotherapy compared with patients with primary brain tumors (Di Giacomo et al., 2023).

Undoubtedly, the introduction of novel therapies has greatly extended survival rate and enhanced the quality of treatment. However, it is essential to emphasize the importance of the early identification and diagnosis of BM in treatment by allowing healthcare professionals to implement treatment as early as possible and then improve the treatment outcomes.

1.2 Diagnosis of Brain Metastases

The identification and diagnosis of brain tumors depend heavily on noninvasive imaging techniques, such as magnetic resonance imaging (MRI), and computed tomography (CT), and their introduction has greatly improved the detection of BM, especially these small asymptomatic BM. MRI has been primarily used to locate the lesion and measure the tumor’s size. However, MRI, as an anatomic and functional imaging modality, is limited in differentiating malignant tumor tissue from the surrounding edema (Langen et al., 2017). CT provides high-quality 3D images of structural information about the body by utilizing the difference in tissues' ability to absorb X-rays (Prince & Links, 2006). CT enables researcher to have better visualization of internal organs, soft tissues, and bones. However, the absorption of X-rays (high-energy particles) in tissues also exposes patients to high-dose ionizing radiation, which might increase the risk of cancer to patients because ionizing radiation that might damage DNA. Based on the Imaging Procedures and Their Approximate Effective Radiation Doses published by Harvard Medical School in 2021, the average dose of a head CT scan is 2 mSv (3 times higher than Mammogram), and the range reported in the literature review is 0.9-4 mSv (Harvard Health, 2021). CT is primarily employed to detect hemorrhage, herniation, and hydrocephalus, but it can also potentially detect oligodendrogliomas or meningiomas (Mabray et al., 2015).

1.2.1 PET imaging

In addition to high-resolution structural imaging mentioned above, functional imaging, such as Positron Emission Tomography (PET), has been actively incorporated with MRI/CT to detect tumors and grade their malignancies. PET is a nuclear medical imaging technique that involves the injection of radiotracers containing positron-emitting radionuclides into the body. These radiotracers produce gamma rays that can be detected by a PET scanner, which generates 4D images of the location and dynamics of physiological processes in the body. In the context of brain tumors, PET imaging can provide valuable insights into the pathophysiological mechanisms of the tumors over time. Another benefit of the injected radiotracers in PET imaging is that they do not disrupt BBB as contrast-enhanced MRI sequences do (Verger et al., 2022). PET images enables the differentiation of normal tissues and tumor tissues because of their distinct radioactivity accumulations that results from their different metabolic speed or targeting specificity. Tumor tissues have higher uptake of radiotracer than normal tissues because of tumor cells uptake, for example, a larger amount of amino acid to increase protein synthesis (K. Chen & Chen, 2011), and other biological mechanisms.

1.2.2 Static and dynamic PET imaging

Based on the acquisition, PET imaging can be categorized as dynamic or static imaging. Static PET imaging contains one image or a series of images at specific time points some time after the radiotracer injection (Turku PET Centre, 2021). In contrast, dynamic PET imaging involves the continuous acquisition of images over a period, typically ranging from a few minutes to several hours, immediately after the radiotracer injection (Turku PET Centre, 2021). Dynamic PET imaging provides a quantitative concentration of radioactivity from each voxel in the image as a function of time, and it is widely applied in the cardiovascular disease (Yoshinaga et al., 2018), neurological disorders, and oncology (Rahmim et al., 2019). The application of dynamic PET imaging in oncology enables the quantification of radiotracer uptake, thus improving the performance of tumor characterization and treatment response by applying modeling to the PET images (Dunnwald et al., 2011).

One advantage of dynamic PET imaging is that it provides the concentration of radioactivity as a function of the time for each voxel, also called the time-activity curve (TAC). Besides the TAC of a single voxel, it is also possible to obtain an average TAC of a particular region or tissue which is of interest by drawing the region of interest (ROI) on the PET image (Turku PET Centre, 2021). In the case of detecting brain tumors, obtaining regional average TACs could not only mitigate noise but also provide precise and detailed radioactivity and metabolic information on the manually segmented tumors (Turku PET Centre, 2021).

1.2.3 Radiotracers of PET imaging

Radionuclides are essential for PET imaging as they contain positron-emitting isotopes which influence radioactivity uptake, including fluorine-18, carbon-11, and gallium-68. Various radionuclides offer insight into different aspects of brain tumor physiology.

1.2.3.1 [^{18}F]FDG

2-Deoxy-2-[^{18}F]fluoro-D-glucose ([^{18}F]FDG) is the dominant PET radiotracer in oncological studies, and it is widely utilized in cancer staging, treatment evaluation and prognosis assessment (K. Chen & Chen, 2011). Fluorine-18 has a physical half-life of approximately 110 minutes (Ashraf & Goyal, 2023). [^{18}F]FDG is an analog of glucose, which makes it an ideal indicator of glucose metabolism by characterizing tumors with an increased glucose uptake (Ashraf & Goyal, 2023). This is because tissues that metabolize more glucose will accumulate more [^{18}F]FDG, and the accumulation of [^{18}F]FDG in tissues is directly proportional to their glucose metabolism rate. As a result, PET imaging with [^{18}F]FDG enables differentiation between tumor and non-tumor tissues with different glucose consumption. Furthermore, this radiotracer also can distinguish malignant from benign tumor tissues based on the extent of glucose metabolism in many cases.

However, [^{18}F]FDG also has a limitation in differentiating whether increased metabolism is caused by cancer or infection/inflammation. [^{18}F]FDG is not a target-specific tracer, so it cannot differentiate between cells with a high metabolic rate associated with neoplasia (K. Chen & Chen, 2011). In addition, its detection of specific brain tumors is limited because some malignant tumors may not present high metabolic rates (Almuhaideb et al., 2011). For example, low-grade gliomas cannot be detected by [^{18}F]FDG (Verger et al., 2018).

1.2.3.2 [^{11}C]Methionine

In contrast, amino acid radiotracer L-[*methyl*-^{11}C]methionine ([^{11}C]methionine) shows increased uptake in gliomas compared with non-tumor tissues (Singhal et al., 2012). The accumulation of [^{11}C]methionine reflects the intracellular metabolism of amino acids, and it is useful in the cerebral tumor imaging (Cosentino et al., 2020). This is because brain tumors can also be easily identified by [^{11}C]methionine due to an increased uptake of [^{11}C]methionine in brain tumor tissues (Cosentino et al., 2020). Although the increased uptake of [^{11}C]methionine in brain tumors has not been well defined yet, it is believed that passive diffusion through the damaged BBB and active tumor uptake in brain tumors might contribute to this (Cosentino et al., 2020).

The application of [^{11}C]Methionine is also extended to differentiate recurrent or progressive tumors from radiation necrosis (Terakawa et al., 2008). However, its use is limited by the short physical half-life (20 minutes) of carbon-11, and a cyclotron is needed to produce carbon-11 (Mabray et al., 2015).

1.2.3.3 Gallium-68 labeled peptides

^{68}Ga-labeled 1,4,7,10-tetraazacyclododecane-1,4,7,10-tetraacetic acid ([^{68}Ga]Ga-DOTA)-conjugated peptides are another widely employed tracers with high sensitivity and specificity in neuroendocrine tumors, such as prostate tumors (Raj & Reidy-Lagunes, 2018) and meningiomas (Verger et al., 2022). It is commonly used in primary tumor detection but can also provide additional information of tumors that cannot be detected by [^{18}F]FDG and [^{11}C]methionine. The half-life of ^{68}Ga is approximately 68 minutes (Martiniova et al., 2016.).

The use of ^{68}Ga-DOTA-conjugated peptides is also advantageous in shortening the imaging time and is more cost-effective than radiotracers a cyclotron generates. ^{68}Ga-Radionuclide can be conveniently obtained from a generator instead of a cyclotron, which facilitates its use for radiolabeling of tracers for PET application (Khan et al., 2009).

1.3 Image Processing and Analysis

Conventional image processing and analysis is skill-demanding and time-consuming as it relies heavily on experts' manual operation of the image process software. Recently, computerized image analysis has been receiving increasing attention as its ability to improve the efficiency and accuracy of image analysis qualitatively and quantitively through image registration and image segmentation (W. Lu et al., 2015). Automated image processing, also known as computerized image analysis, typically consists of three stages: pre-processing, processing, and post-processing.

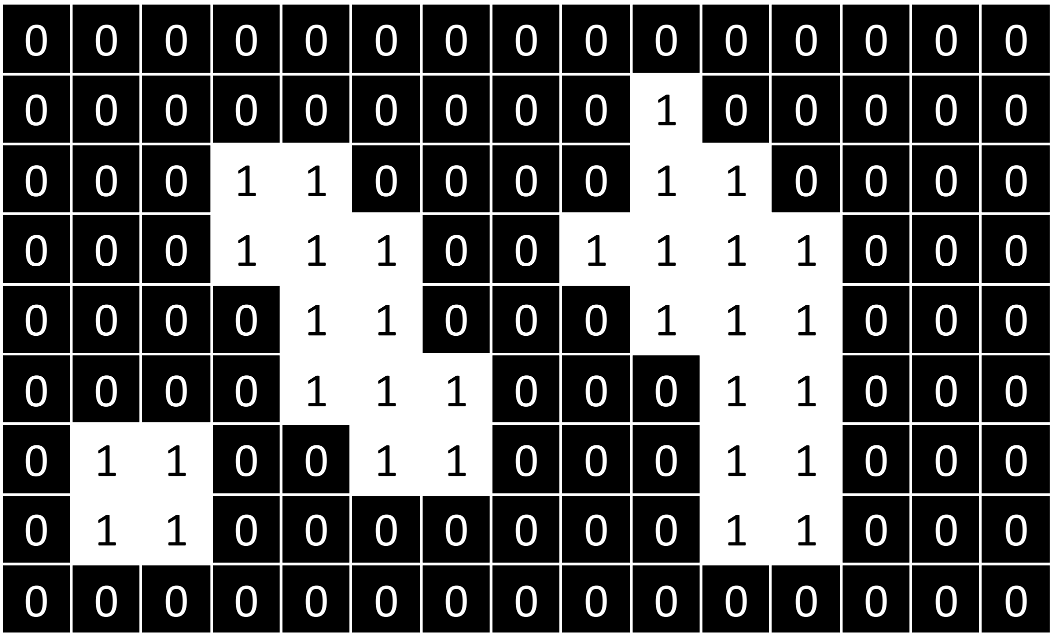
1.3.1 Pre-processing

Pre-processing is a key step for image processing because it prepares input images for the best output. Pre-processing is the process where adjustments will be made to the size and orientation of images. The purpose of pre-processing is to improve image quality and eliminate unnecessary information so that precedent processing can function effectively and generates desired output.

1.3.2 Processing

Image processing depends on the desired output. This thesis aims to segment brain metastases from PET images, which means image segmentation will be the fundamental processing. Image segmentation is a typical image processing step to extract useful information from images by dividing an image into several homogeneous groups (Szeliski, 2022). In this thesis, image segmentation is to separate tumors from non-tumor tissues.

Segmentation can be achieved by various methods, including thresholding, region-based methods, deformable model methods, and clustering. Thresholding uses a fixed thresholding value of intensity to separate pixels and pixels above the thresholding value are the foreground, and below are the background (Kang et al., 2009). Region-based methods form regions by merging connected pixels with similar properties (Fasihi & Mikhael, 2016). Deformable model methods are used to handle challenging 3D image segmentation. Deformable model is a technique that iteratively defines curves based on the internal and external forces to best contour the boundaries of objects in the image (Fasihi & Mikhael, 2016). Grouping data into subsets with high similarity within the subset but the low similarity in inter groups is called clustering (Madhulatha, 2012). Clustering is viewed as unsupervised learning from the perspective of machine learning (Malathi & Kamal, 2015).



0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
0	0	0	1	1	0	0	0	0	1	1	0	0	0	0
0	0	0	1	1	1	0	0	1	1	1	1	0	0	0
0	0	0	0	1	1	0	0	0	1	1	1	0	0	0
0	0	0	0	1	1	1	0	0	0	1	1	0	0	0
0	1	1	0	0	1	1	0	0	0	1	1	0	0	0
0	1	1	0	0	0	0	0	0	0	1	1	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 1. Clustering items into different groups based on their similarities and differences.

Clustering has a wide application in image segmentation, such as drawing regions of interest (ROI), due to its distinct advantages. One distinct advantage of clustering is that predefined classes and training examples are not required when grouping the data because clustering is an unsupervised machine learning (Malathi & Kamal, 2015). Clustering-based algorithms are also advantageous in handling multidimensional data (Pham et al., 2000), such as 4D PET images in our study. The application of clustering in PET images is to partition voxels with similar TACs into the same subsets so that tumor tissues can be differentiated from non-tumor tissues based on their TACs.

Clustering algorithms are categorized as hierarchical and partition algorithms based on their structure. Hierarchical algorithms separate the given data into subsets hierarchically, whereas partition algorithms divide the given data into multiple subsets in one step. Clustering can also be categorized into various algorithms based on characteristics, such as K-Means Clustering and Gaussian Mixture Model (GMM). This essay focuses on K-Means Clustering and GMM, which will be discussed in detail in the Material and Methods section.

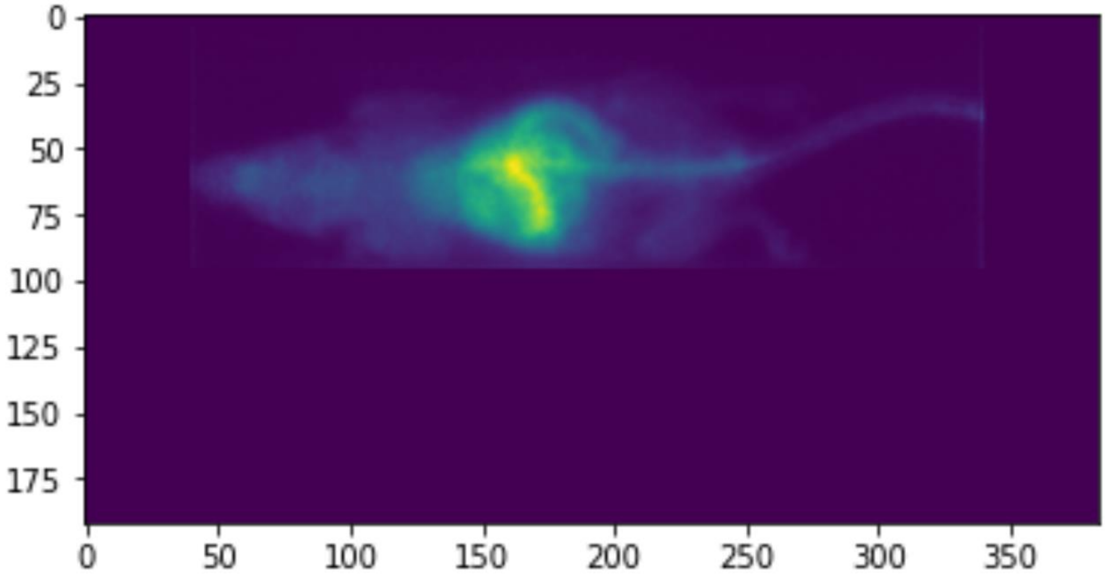


Figure 2. 2D visualization of an example 4D PET image.

1.3.3 Post-processing

Post-processing is used to reconstruct the image; such a procedure is important to enhance the raw result generated from clustering. Commonly used post-processing of medical images include smoothing and resizing.

1.4 Evaluation and Validation of Clustering

Certain criteria and metrics are adopted to evaluate the performance of automated image segmentation. These metrics will evaluate the segmentation from three aspects: accuracy, precision, and processing time. Commonly used metrics include Confusion Matrix, Log Loss, and Jaccard index. The confusion Matrix compares the actual value with the predicted value and divides the predicted value into four categories based on the comparison. Log loss indicates the probability of how close the predicted result is to the actual result. In contrast, the Jaccard index measures the similarities between the predicted and actual results. The higher is Jaccard index is, the more accurate the algorithm is. Another metric to validate the accuracy of automated segmentation is by comparing manually segmented and automatically segmented tumors. The thesis uses the Jaccard index and comparison of TACs, and they will be discussed in a later section.

2 THE MOST IMPORTANT LITERATURE

Clustering has been widely applied in cancer diagnoses, such as brain cancer (M & P, 2018) and breast cancer (C.-H. Chen, 2014) and neurological diseases like Alzheimer's disease (Alashwal et al., 2019). An integrated clustering of K means and spatial Fuzzy C means (FCM) clustering was implemented on MRI images to segment human brain tumors and demonstrated a good segmentation performance with an 87% classification efficiency and an increased classification efficiency of 93.28% after post-processing with discrete wavelet transform (M & P, 2018).

Various clustering-based algorithms are also applied in segmenting brain tumors in PET images, such as the Iterative FCM (Zhu & Jiang, 2003), Spatial FCM (Arumugam & Raja, 2013), and K-Means Clustering (Abualhaj et al., 2017). Both normal FCM and iterative FCM can detect tumors and generate volume mesh from PET images; however, iterative FCM outperforms normal FCM when tumors differ from non-tumor tissue significantly (Zhu & Jiang, 2003).

K-means clustering indicates promising results in segmenting different functional structures in the brain from dynamic PET images (Abualhaj et al., 2017). Spatial FCM clustering is effective in identifying affected portions of the brain from dynamic PET images of Alzheimer's disease patients (Arumugam & Raja, 2013).

The evaluation and comparison of multiple clustering-based algorithms in the MRI brain tumor images (Selvy et al., 2018.), and CT brain tumor images (Sharma & Aggarwal, 2010) have been investigated. Although applications of single clustering algorithms K-Means in PET brain images were also investigated, the evaluation and comparison of multiple clustering-based algorithms to detect metastatic brain tumors in mice PET images have not yet been studied. This thesis tests the K-means clustering’ and Gaussian mixture model’s ability to detect metastatic brain tumors from PET images and evaluate their performances and fills this gap.

3 AIMS AND HYPOTHESES

This thesis focuses on one hypothesis and two aims.

This thesis hypothesizes that clustering can be used to identify the tumor tissue from PET images.

The first aim (Aim 1) is to test the performance of the different clustering-based algorithms in detecting brain tissues from PET images in terms of processing time and accuracy. Furthermore, the other aim (Aim 2) is to identify the best-performing algorithm based on the result obtained from Aim 1.

4 MATERIALS AND METHODS

4.1 Material

4.1.1 PET images

19 PET images of 19 mice intracranially injected (Injection site, Figure 3) with human breast cancer cells were analyzed. Among them, three radiotracers were respectively injected into each mouse [^{18}F]FHC (n=11), [^{11}C]methionine (n=4), and [^{68}Ga]DOTA (n=4) to compare the efficacy of tracers.

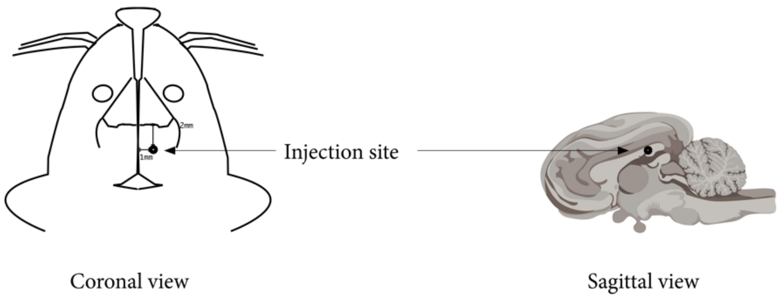
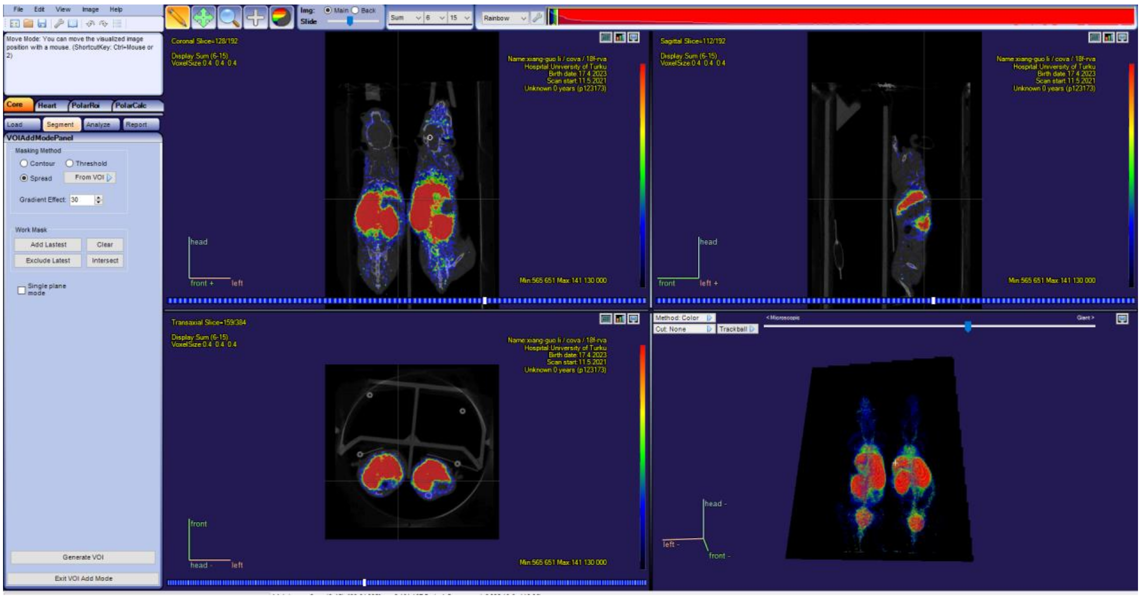


Figure 3. The injection site in the mouse brain from coronal and sagittal views.

A PET image (Figure 4) has 4 dimensions, coronal, sagittal, and transaxial, and time. For images injected with [^{18}F]FHC [^{11}C]methionine, and [^{68}Ga]DOTA, their total number of timeframes are respectively 21 (6 x 10s, 4 x 60s, 5 x 300s, 6 x 600s, 90 min), 21 (6 x 10s, 8 x 30s, 5 x 60s, and 2 x 300s, 20 min), and 16 (6x10s, 4x60s, 5x300s, 1x600s, 40 min).


Move Mode: You can move the visualized image position with a mouse. (ShortcutKey: Ctrl+Mouse or 2)

Masking Method: ● Spread

Gradient Effect: 30

Coronal Slice=128/192

Sagittal Slice=112/192

Transaxial Slice=159/384

Display:Sum (6-15)
VoxelSize:0.4 0.4 0.4

Name:xiang-guo li / cova / 18f-rva
Hospital:University of Turku
Birth date:17.4.2023
Scan start:11.5.2021
Unknown 0 years (p123173)

Figure 4. Screenshot of the 4D visualization of a PET image in Carimas from transaxial, coronal, and sagittal views over 21 time points as well as 3D rendering.

PET images of mice were both segmented manually and automatically by the clustering pipeline. Manual segmentation was operated using Carimas 2.10 (Turku PET center, Turku, Finland), and automated segmentation was done with a clustering pipeline built in Python (version 3.11.1). Manual segmentation is considered as ground truth in this case, and the automated segmentation was compared with the ground truth to evaluate the accuracy and efficiency of each clustering algorithm.

4.1.2 Data format

All PET images used in this thesis are stored in two data formats: Digital Imaging and Communications in Medicine standard (DICOM) or in Neuroimaging Informatics Technology Initiative (NIfTI).

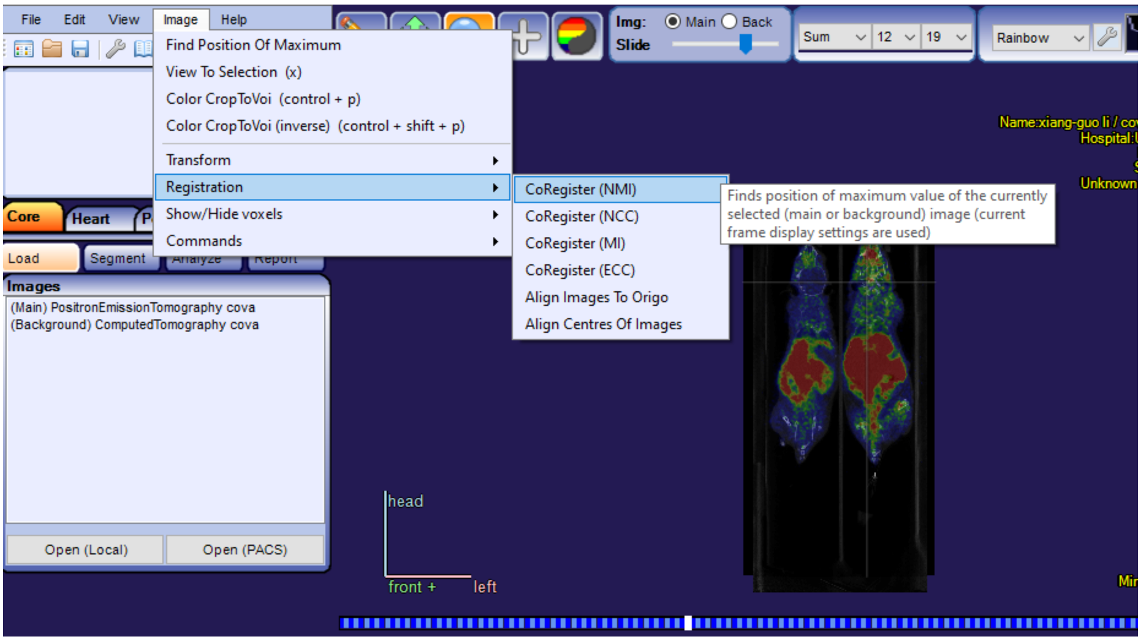
DICOM is used to integrate imaging equipment that might be installed in multiple manufacturers, such as PET and CT images. DICOM is different from other image formats because it groups information into data sets (Varma, 2012). PET images in DICOM format are read in Carimas.

NIfTI is another data format that is used to store medical images, which is developed by Biomedical Imaging Resource (BIR) at Mayo Clinic. NIfTI contains metadata of voxels up to 7 dimensions and supports a variety of data types. And it also provides additional information about the coordinate system and how to interpret the data of the image (NIfTI (Neuroimaging Informatics Technology Initiative) Reader/Writer, n.d.). PET images in NIfTI are read in Python.

4.2 Manual Segmentation

Manual segmentation was performed in the PET image processing software Carimas with the anatomical reference from CT images. Manual segmentation involves four major steps.

Step 1: Registration of PET and CT images. The PET and corresponding CT images were loaded to Carimas, where PET was the main image and CT was the background (Figure 5). Next, the PET image and CT image were registered using Normalized Mutual Information (NMI). After registration, PET/CT image gives a good visualization of radioactivity with an anatomical reference (Figure 6).



Finds position of maximum value of the currently selected (main or background) image (current frame display settings are used)

(Main) PositronEmissionTomography cova
(Background) ComputedTomography cova

Figure 5. Screenshots of the automatic registration of PET and CT image in Carimas based on NMI.

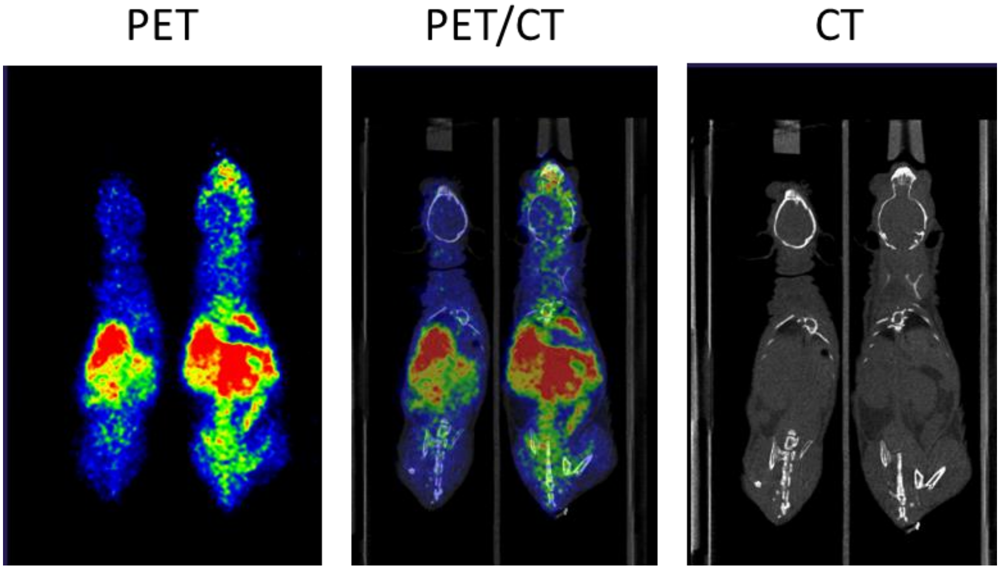


Figure 6. Screenshots of the visualization of PET/CT image with both functional and anatomical information after registration.

Step 2: Adjustment of the scaling bar. Maintaining objectivity in the adjustment of scaling parameters is challenging. To maintain consistency, most images were adjusted to the range of 10^5-10^6 Bq/ml to achieve better visualization where voxels with higher radioactivity in the skull can be identified and differentiated.

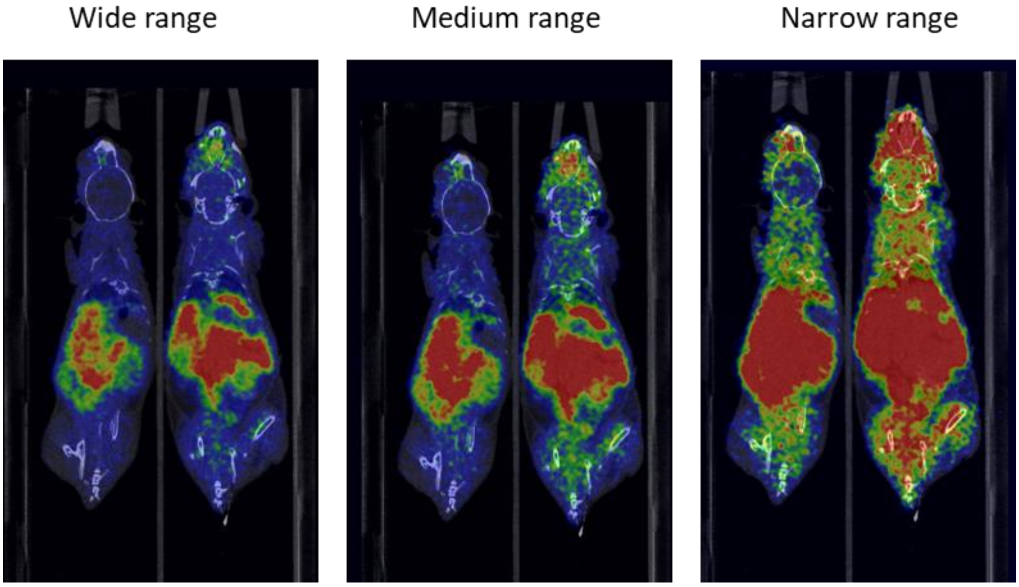
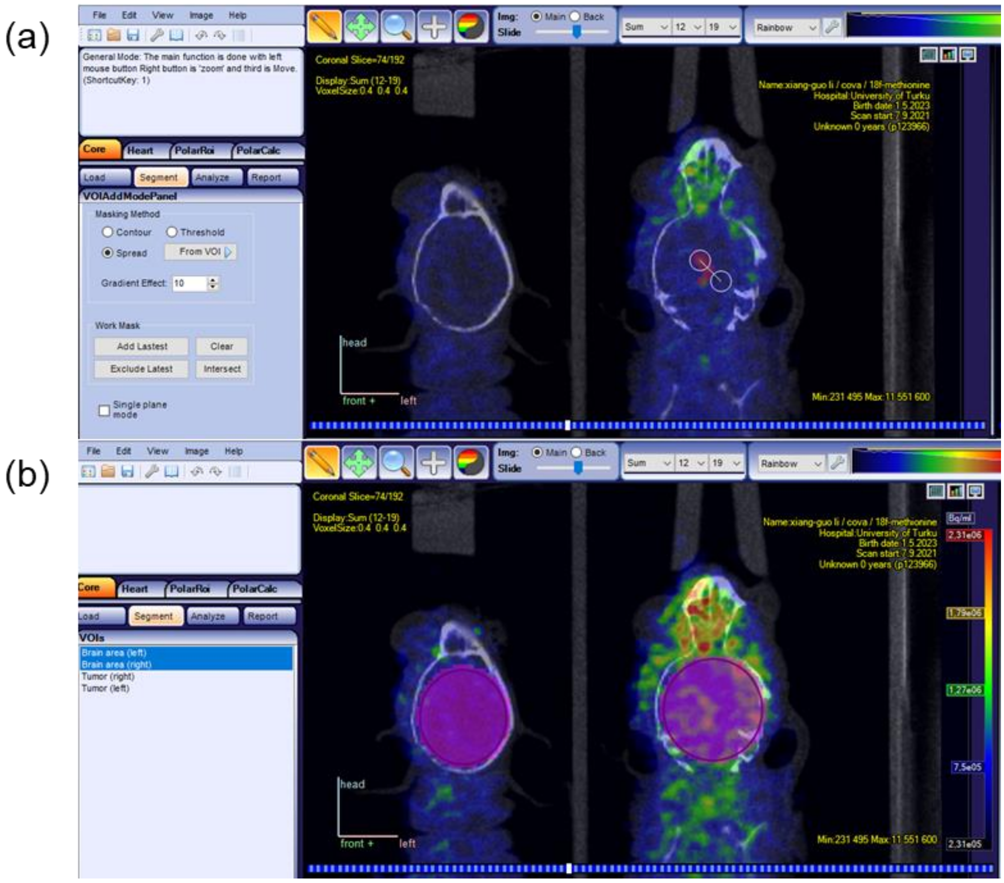


Figure 7. Screenshots of the visualization of PET/CT images with different scaling ranges. The scaling range should be carefully chosen in order to differentiate tumors from surrounding non-tumor tissues.

Step 3: Selection of timeframes (Figure 7). The selection of timeframes depends on the half-life time of the injected radiotracer. In this thesis, the initial and final time points were excluded, and only the selected time intervals were retained. Images injected with [^{18}F]FHC, timeframe 6-15 or 10-15 were retained, and images injected with [^{11}C]methionine timeframe 12-19 were retained. Selecting proper timeframes was done along with adjusting the scaling bar.

Step 4: Draw a 3D region of interest (ROI) (Figure 8). ROIs were drawn with the Carimas plug-in tool called Found Region based on the anatomical reference provided by CT images and the injection site (Figure 4). The injection gave an insight into the possible location of the metastatic tumor. The brain area of each mouse was also drawn with a 3D Sphere tool.



General Mode: The main function is done with left mouse button Right button is 'zoom' and third is Move. (ShortcutKey: 1)

Coronal Slice=74/192

Display:Sum (12-19)
VoxelSize:0.4 0.4 0.4

Name:xiang-guo li / cova / 18f-methionine
Hospital:University of Turku
Birth date 1.5.2023
Scan start 7.9.2021
Unknown 0 years (p123966)

Min:231 495 Max:11 551 600

Brain area (left)
Brain area (right)
Tumor (right)
Tumor (left)

Figure 8. Screenshots of drawing ROIs in Carimas. 7(a),the drawing of brain metastases; 7(b), drawing of brain area.

Step 5: Exportation of ROIs. After drawing ROIs, and brain area was exported as separate NIfTI files for later use in automated segmentation. Quantitative analyses were performed on the defined ROIs to obtain time-activity curves in Results.

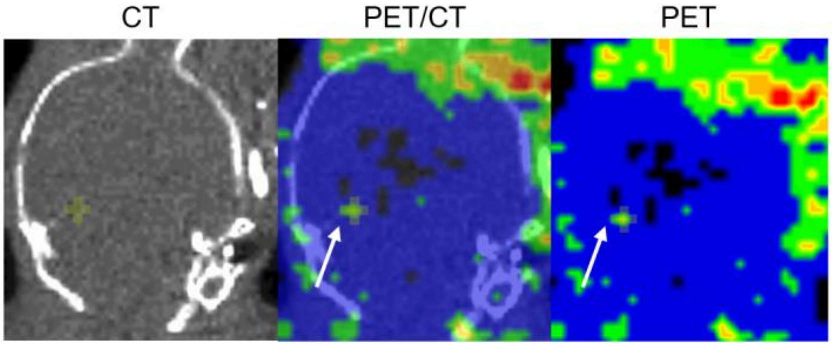


Figure 9. Visualization of manual segmentation in PET/CT

4.3 Automated Segmentation

Automated segmentation was performed on each PET in addition to manual segmentation. This approach employed a pipeline that involved reading DICOM file, converting them into binary mask, performing clustering algorithms, and reconstruction of 3D image.

Python is used as the main programming language. Python packages pydicom (2.3.1) (Mason, 2011), numpy (1.24.2), matplotlib (3.7.1) (McKinney et al., 2010), pandas (2.0.0), sklearn (0.0.post1) (Pedregosa et al., 2011), nibble (5.1.0), seaborn (0.12.3) (Mwaskom et al., 2017) are used.

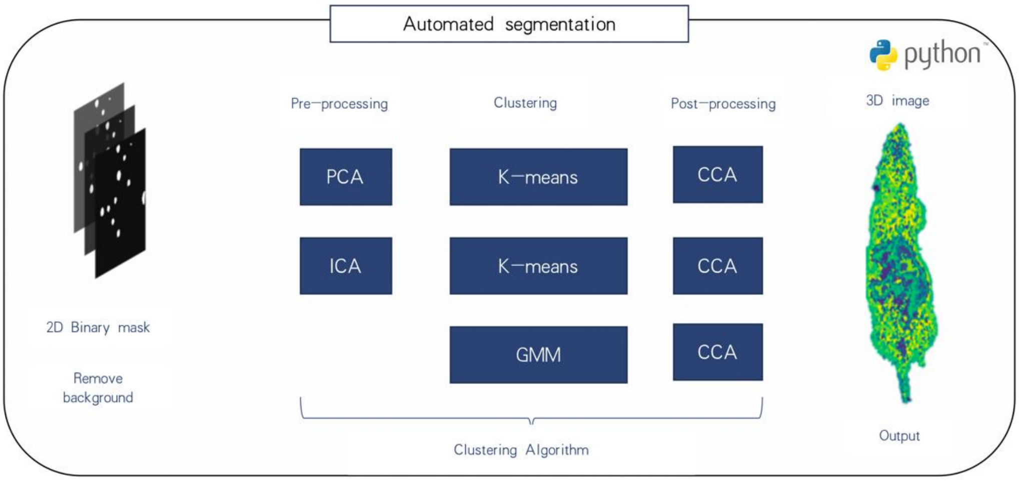


Figure 10. Schematic drawing of automated segmentation pipeline

An automated segmentation pipeline consists of 4 steps in the, as suggested in Figure 10.

Step 1: Reading DICOM format as a 4D array. DICOM file is read as an array (192, 384,384,21) by the Python function pydicom. dcmread, where the metadata stored in each array voxel was accessed, including the intensity of radioactivity, location, and orientation.

Step 2: Converting into a binary mask. This step converts the 4D array into a 2D binary mask. Firstly, the quantile radioactivity of all voxels was calculated. Subsequently, the image was transformed into a binary mask based on the radioactivity of each voxel. Voxels with intensity greater than 95% quantile intensity were considered foreground, and those with intensity lower than 95% quantile intensity were considered background. Next, the background was eliminated from the image. Lastly, the coordination of voxels from the foreground was extracted and stored separately for later use.

Step 3: Performing clustering algorithms. Each clustering algorithm consists of pre-processing, clustering, and post-processing. In the pre-processing step, the images were denoised, and the important features were extracted by either PCA or ICA. Subsequently, images were clustered by K-means clustering or GMM. Finally, post-processing CCA was performed to refine the segmentation results.

Three clustering algorithms are:

I. *Clustering 1*: pre-processing PCA, K-means clustering (with pre-defined clustering number 21), and post-processing CCA.
II. *Clustering 2*: pre-processing ICA, K-means clustering (with pre-defined clustering number 22), and post-processing CCA.
III. *Clustering 3*: no pre-processing, GMM clustering (with pre-defined clustering number 18), and post-processing.

Clustering results from each clustering method were exported as NIfTI file in .nii format to extract TCAs.

Step 4: Reconstruction of a 3D image from clustering results. After performing clustering, the obtained NIfTI file was read in combination with the coordinate extracted from step 2 to reconstruct a 3D image.

4.3.1 Pre-processing

Prior to clustering, PET images are pre-processed to remove the noise from the images that might affect the capacity of the segmentation algorithms. Pre-processing used in this thesis is principal component cnalysis (PCA) and independent component analysis (ICA). PCA is applied to extract meaningful features from confounding data and use these extracted second-order statistics to present the original data (Bugli & Lambert, 2007). The application of PCA maximizes data interpretability while minimizing information loss. Unlike classical pre-processing methods, ICA uses linearly combined independent and non-normal variables to present a multidimensional random dataset (Hyvärinen & Oja, 2000). Its assumption of non-Gaussianity data allows it to identify its original, underlying components, which cannot be done by classical methods (Hyvärinen, 2013). ICA is widely used to derive input functions from dynamic PET image (Lee et al., 2001).

After being pre-processed by either PCA or ICA, the 4D array was converted into a 2D array. In this step, only important features of the original data were extracted and retained. This 2D array assigns a unique label to each extracted voxel and presents the time activity function of each voxel as a function of time.

4.3.2 Clustering

Clustering methods used in this thesis are K-means clustering and Gaussian mixture model (GMM). K-means clustering is the fundamental algorithm that assigns a centroid for each group and finds the minimum sum of distances between the data point and their corresponding cluster centroid by iterating the value of the centroids (Madhulatha, 2012). K-means clustering can handle massive data quickly and efficiently (Malathi & Kamal, 2015). For PET images, K-means clustering categorizes voxels into regions with similar kinetic behavior time-activity curves (Abualhaj, 2017).

In contract, Gaussian mixture model clustering groups voxels into a set of clusters based on their probability of being a mixture of several normal distributions with different mean and standard deviation, this is achieved by iterating expectation-maximization (EM) algorithm (Baselli et al., 2016).

This thesis employed three different clustering methods, with 21, 22, and 18 predefined clusters, respectively. When clustering, voxels in ROIs were grouped into these predefined groups based on their similarities and differences in TACs.

Once clustering is completed, the 2D array was reread in combination with the extracted coordinate of each voxel to reconstruct 3D image.

4.3.3 Post-processing

While separating different objects in this binary image, connected components analysis (CCA) operation is advantageous in refining the clusters. Connected components analysis, also known as connected components labeling, clusters pixels or voxels into subgroups based on whether pixels are physically connected (He et al., 2017). Whether two pixels/voxels are connected is determined by if one pixel is located in the other pixel's predefined neighborhood (4, 8, and 26 neighbors/ connectivity), as indicated in Figure 11.

After CCA, clusters obtained from clustering further divided into subclusters, and each subcluster was assign to a unique label. Furthermore, features of these the labeled subcluster can be calculated and accessed, such as shape, size, and location (He et al., 2017). In this thesis, the operation of CCA enables us to find the cluster (or automatically segmented tumor) that has the best overlap with manual segmented tumors.

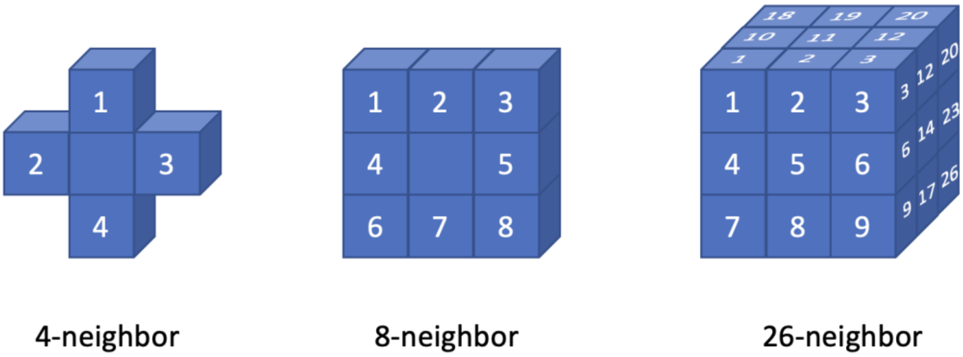
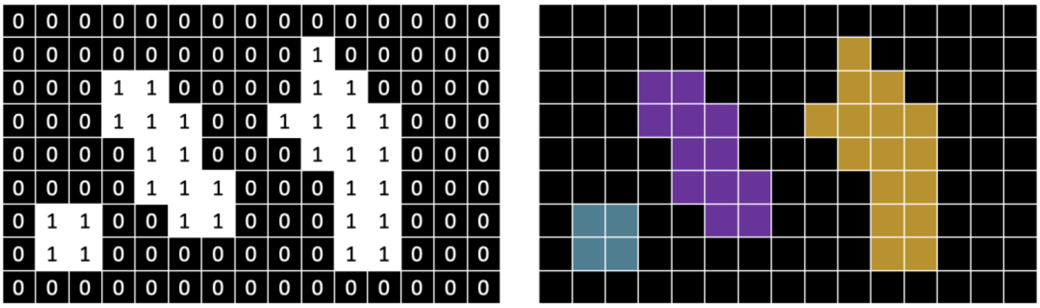


Figure 11. Presentation of connected component analysis with 4-neighbor, 8-neighbor, and 26-neighbor.

Figure 12 gives us a good visualization of the effect of CCA. Prior to post-processing, voxels in clusters 1, 2, and 3 were considered as a single cluster because they exhibited similar TAC, even though they were not physically connected in the image. CCA separated these three clusters based on their spatial connectivity and reassigned them with new labels. As a result, three distinct clusters were obtained after performing CCA. In this thesis, I tested CCA on K-means-based segmentations Clustering 1 and Clustering 2, as they worked on all images.



0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
0	0	0	1	1	0	0	0	0	1	1	0	0	0	0
0	0	0	1	1	1	0	0	1	1	1	1	0	0	0
0	0	0	0	1	1	0	0	0	1	1	1	0	0	0
0	0	0	0	1	1	1	0	0	0	1	1	0	0	0
0	1	1	0	0	1	1	0	0	0	1	1	0	0	0
0	1	1	0	0	0	0	0	0	0	1	1	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Prior to CCA

After CCA

Figure 12. Comparison of cluster labeling prior to CCA and after CCA.

4.4 Evaluation

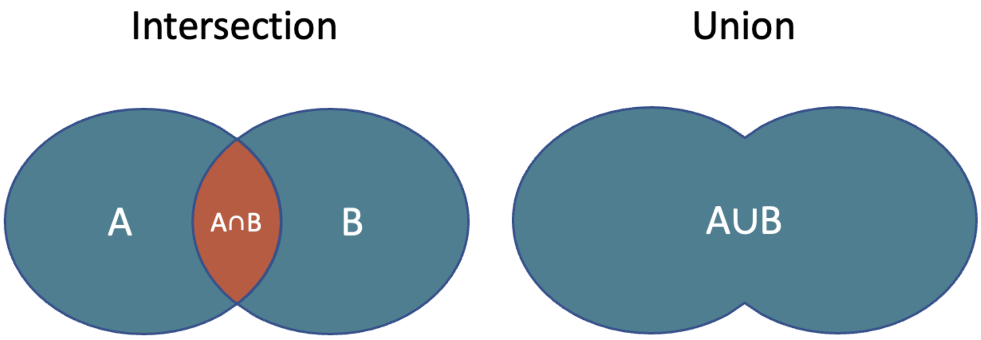
To evaluate the performances of the three different clustering methods, the automated segmentation was compared to the ground truth (manual segmentation). Two metrics were used to assess their performance: processing time and accuracy. Aim 1 was evaluated by calculating the execution time of each clustering method and the Jaccard index between manual segmentation and automated segmentation from each clustering method. Furthermore, Aim 2 was evaluated based on the overall performance of the execution time and the Jaccard index, in conjunction with the validation of TACs.

4.4.1 Execution Time

The approximate time to manually segment 1 PET image in Carimas was documented manually. The time to execut each clustering method was documented automatically with the Python perf_counter function in seconds. The execution time of manual segmentation and different clustering methods were compared based on the p-value, which is obtained from the Mann-Whitney test after testing their normality.

4.4.2 Jaccard Index

The Jaccard index evaluates the accuracy of each clustering result. Jaccard index compares the similarities between automatically segmented tumors and manually segmented tumors. The jaccard index, also known as the Intersection over Union (I/U), quantifies the overlapping percentage between manual segmentation (considered the ground truth in our case) and automated segmentation (Figure 13). The intersection (A ∩ B) consists of the voxels that exist in both the manual segmentation and clustering output. In contrast, the union (A ∪ B) consists of all voxels in the manual segmentation or clustering output. The higher the Jaccard index is, the higher overlapping there is, and the better the clustering method is.



$$J(A,B) = \frac{|A \cap B|}{|A \cup B|} = \frac{|A \cap B|}{|A| + |B| - |A \cap B|}$$

Figure 13. Visualization of Jaccard Index

4.4.3 TACs of manual segmentation and automated segmentation

Another metric to validate the accuracy of clustering is through the comparison of manually segmented tumors and automatically segmented tumors. To verify whether two TACs share similar kinetics, statistical tools, such as runs test, or maximum run length are usually applied to test their similarity (Turku PET Centre, 2021). However, due to the limitation of time and technical issues, this thesis only compared the visual analysis of plotted TACs of manually segmented tumors and automatically segmented tumors to validate if tumors segmented from two methods share certain similarity.

4.4.4 Statistical analysis

Statistical methods to evaluate the differences between Jaccard indices and the running times of different clustering approaches included the Mann-Whitney and Wilcoxon rank-sum tests. Clustering 1 and clustering 2 were tested with Wilcoxon rank-sum tests because they are paired. Whereas clustering 3 was tested against clustering 1/2 with the Mann-Whitney test because GMM used in clustering 3 did not work on all images. To define the statistically significant difference between the approaches, a conventional p-value cutoff of 0.05 was used.

5 RESULTS

5.1 *Visualization Inspection of Clustering Results*

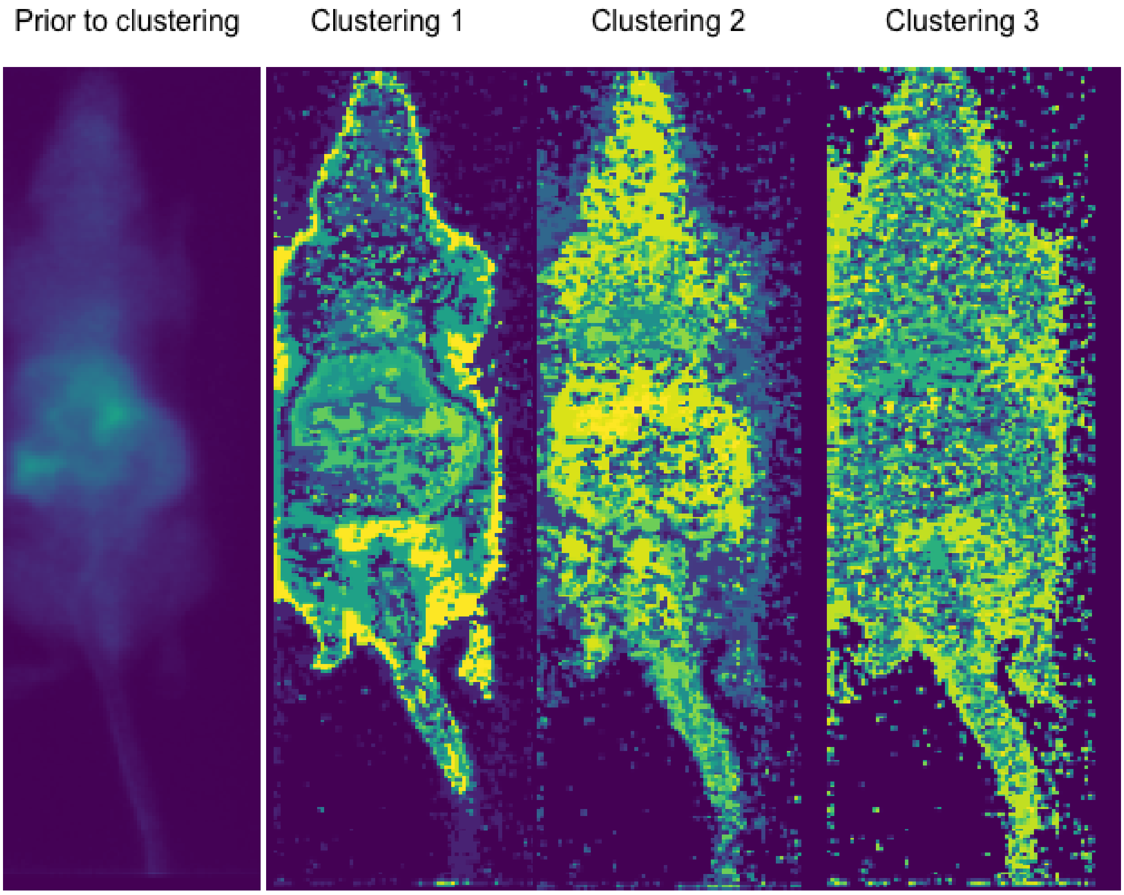


Figure 14. Visualization of PET images after automated segmentation.

The performance of three different clustering methods has been visualized prior to CCA, as visualized in Figure 14. Based on the visualization, Clustering 1 yielded the best results. A clear outline and easily discerned internal organs suggest that Clustering 1 has produced a highly accurate segmentation of the image. In contrast, the results of Clustering 3 appear to be the least satisfactory, as evidenced by the mouse's blurry outline and the difficulty distinguishing organs from surrounding tissues. Quantitative analysis will be presented later for a more comprehensive evaluation of the performance of each clustering method.

5.2 Execution Time

5.2.1 Average execution time prior to CCA

Prior to CCA, Clustering 1 is the most efficient among the three clustering methods. The average execution times of the three clustering methods are 39.1s, 53.8s, and 445.8s, respectively. The execution times of Clustering 1 and Clustering 2 are relatively similar, whereas the execution time of Clustering 3 is more than ten times higher than that of Clustering 1 (Figure 15).

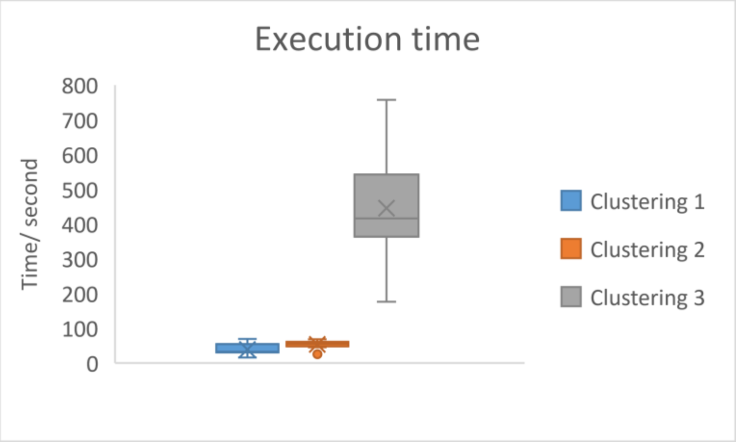


Figure 15. Execution time of different clustering methods.

5.2.2 Average execution time after CCA

After CCA, the execution times of all three clustering methods remained the same as those of prior to CCA. It appears that CCA has little impact on the execution times of Clustering 1 and Clustering 2 because the CCA of these two clustering methods took less than 1 second.

However, it seems that CCA is not effective for Clustering 3 because CCA did not work on images segmented by Clustering 3 and no data is available for Clustering 3 after post-processing. A potential reason might be that CCA is not compatible with GMM, and further studies are necessary to address the incapability issue between CCA and GMM.

5.2.3 Statistical analysis of execution time of different clustering methods

Table 1. Statistics Analysis of Execution Time of Different Clustering Methods.

Statistics Analysis of Execution Time of Different Clustering Methods			
Method 1	**Method 2**	**Test**	**P-value**
Clustering 1	Clustering 2	Mann Whitney test	0.03
Clustering 1	Clustering 3	Wilcoxon signed-rank	0.01
Clustering 2	Clustering 3	Wilcoxon signed-rank	0.01

It has been observed that prior to performing CCA, Clustering 1 is significantly faster than Clustering 2 and Clustering 3, with a p-value of 0.03 and 0.01, respectively (Table 1). Moreover, Clustering 2 is also significantly faster than Clustering 3, with a p-value of 0.01, as determined by the Wilcoxon signed-rank test (Table 1).

5.3 Accuracy

5.3.1 Jaccard Index

Prior to CCA, the Jaccard indices of three clustering methods were fairly small, with the maximum Jaccard index of Clustering 1, Clustering 2, and Clustering 3 being 1.24E-05, 7.11E-05, and 1.36E-05 respectively (Table 2). This suggests that the accuracies of the three clustering methods are relatively low.

Table 2. Comparison of the Jaccard Indices of Different Clustering Methods Prior to and After CCA.

The Jaccard Indices of Different Clustering Methods Prior to and After CCA					
Clustering Method	**Clustering 1**		**Clustering 2**		**Clustering 3**
Mous number	**Prior to CCA**	**After CCA**	**Prior to CCA**	**After CCA**	**Prior to CCA**
1	9.13E-05	6.61E-05	1.41E-04	3.64E-01	3.43E-05
2	2.72E-04	1.05E-01	3.62E-04	1.61E-01	1.18E-04
3	3.96E-04	9.09E-02	2.45E-04	2.50E-01	1.12E-04
4	1.31E-03	1.67E-01	1.16E-03	1.96E-01	7.90E-04
5	1.24E-05	1.15E-05	7.11E-05	5.00E-01	1.36E-05
6	1.14E-04	3.85E-01	2.47E-04	1.74E-01	
7	2.32E-04	1.67E-01	1.21E-03	7.14E-01	1.08E-04
8	4.86E-05	2.35E-01	3.60E-04	2.86E-01	3.24E-04
9	1.14E-04	1.13E-04	1.56E-04	6.98E-02	4.01E-04
10	6.52E-05	7.72E-05	1.95E-04	2.63E-02	
11	2.13E-04	5.17E-02	6.25E-04	9.55E-02	6.26E-04
12		5.25E-06		8.04E-05	
13		4.38E-04		2.29E-02	
Average	2.60E-04	9.24E-02	4.33E-04	2.20E-01	2.81E-04

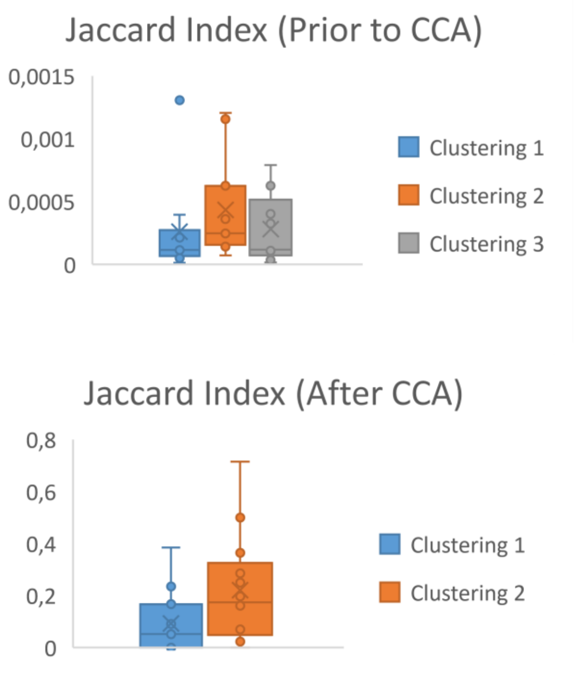


Figure 16. Boxplots of Jaccard indices of different clustering methods prior to and after CCA.

Among the three clustering methods, Clustering 2 had the highest average Jaccard index prior to CCA (Table 2). However, the Jaccard Indices of Clustering 2 were not statistically significantly higher than those of Clustering 1 and Clustering 3, with a p-value of 0.12 and a p-value of 0.29, respectively. This suggests that Clustering 2 is not significantly more accurate in detecting brain tumors than Clustering 1 and Clustering 3 prior to CCA (Table 3).

Table 3. Statistical Analysis of the Jaccard Index of Different Clustering Methods (Prior to CCA)

Statistical Analysis of Jaccard Index of Different Clustering Methods (Prior to CCA)

Method 1	Method 2	Test	P-value
Clustering 1	Clustering 2	Mann Whitney test	0.12
Clustering 1	Clustering 3	Wilcoxon signed-rank	0.71
Clustering 2	Clustering 3	Wilcoxon signed-rank	0.29

The accuracy of Clustering 1 and Clustering 2 has improved after post-processing, with maximum Jaccard Indices of 0.39 and 0.71, respectively (Table 2). In contrast, the performance of Clustering 3 does not improve after CCA. The Jaccard Indices of Clustering 2 have significantly improved, with a p-value of 0.0001, whereas Clustering 1 has not significantly improved, with a p-value of 0.15 (Table 4).

Table 4. Statistical Analysis of the Jaccard Index of Different Clustering Methods

Statistical Analysis of Jaccard Index of Different Clustering Methods after CCA

Method 1	Method 2	Test	P-value
Clustering 1 (prior to CCA)	Clustering 1 (after CCA)	Mann Whitney test	0.15
Clustering 2 (prior to CCA)	Clustering 2 (after CCA)	Mann Whitney test	0.0001

Furthermore, after post-processing, Clustering 2 is statistically significantly more accurate than Clustering 1 (Table 5), with a p-value of 0.045 determined by the Mann-Whitney test (Table 5). This suggests that the post-processing CCA has a more significant positive impact on the accuracy of Clustering 2 than on Clustering 1.

Table 5. Statistical Analysis of the Jaccard Index of Different Clustering Methods (After CCA)

Statistical Analysis of Jaccard Index of Different Clustering Methods (After CCA)			
Method 1	**Method 2**	**Test**	**P-value**
Clustering 1	Clustering 2	Mann Whitney test	0.045

5.3.2 Visualization of manual segmentation and cluster with max Jaccard index

Prior to CCA, the identification of brain tumors from automatically segmented images was challenging because of the presence of “noises” (presented as blue dots in (Figure 17(a)(b)(c)). When calculating the Jaccard Index, numerous clusters were selected from the substantial clusters obtained after automated segmentation based on their similarity to manually segmented tumors. However, not all these clusters correspond to actual brain tumors because some of them locate far away from the injection site and even outside the skull. These clusters were regarded as noises, and the elimination of noises was crucial to identify brain tumors. A high level of noise also corresponds to previously observed low Jaccard indices prior to post-processing (Table 2).

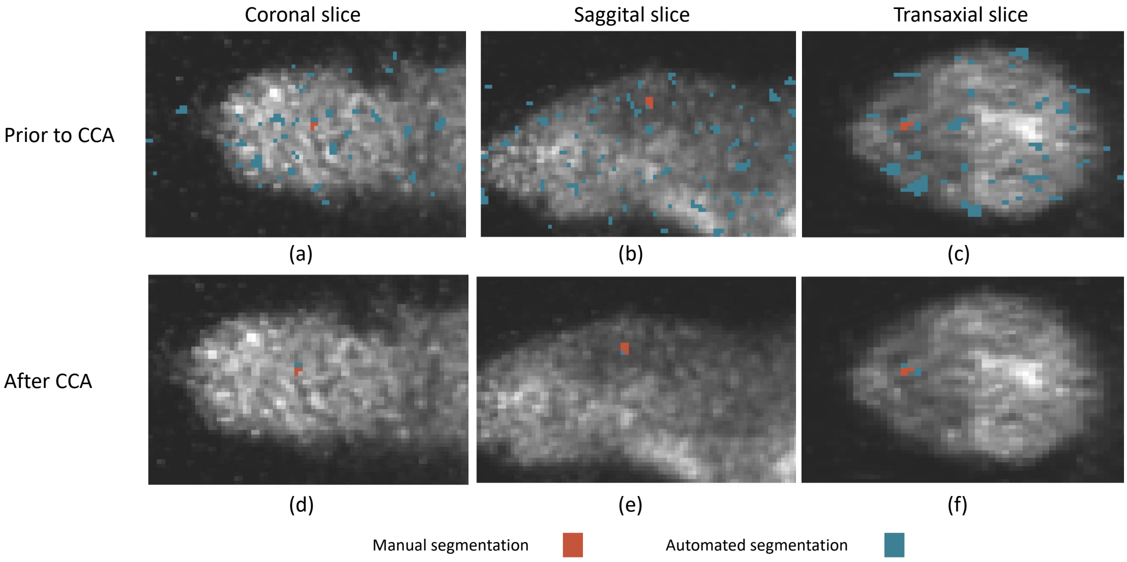


Figure 17. Visualization of manual segmentation and the automatically segmented cluster with the highest Jaccard Index prior to and after CCA in coronal, sagittal, and transaxial views.

Prior to CCA, low Jaccard indices (Table 2) and poor visualization results (Figure 17(a)(b)(c)) indicate that there was a significant difference between manual segmentation and automated segmentation. However, after the application of CCA, an improvement in the accuracy of the automated segmentation results is witnessed in Figure 17(d)(e)(d). This is primarily achieved by effectively eliminating noises (those clusters share similar TAC but are physically unconnected to the cluster with the highest Jaccard index). Therefore, the performance of automated segmentation has significantly improved after the removal of these noises.

5.4 Time Activity Curve

5.4.1 TACs of manual segmentation and automated segmentation

In this section, TACs were used to validate that the cluster exhibiting the highest Jaccard index shares identical TAC with the manually segmented tumors. The clusters with both the highest and lowest Jaccard Index of Clustering 1 and Clustering 2 were compared, and their TACs prior to and after CCA were compared. In addition, the cluster with the highest Jaccard index obtained from Clustering 3 was also plotted separately. For Clustering 1 and Clustering 2, each plot features four TACs: manual segmentation, cluster with the highest/lowest Jaccard index (cluster), and their corresponding clusters after post-processing (cluster_CCA), and brain area. TACs of Clustering consist of three: manual segmentation, cluster with the highest/lowest Jaccard index (cluster), and brain area. And each TAC demonstrated the concentration of radioactivity in the unit of Bq/mL as a function of time.

5.4.1.1 TACs of the cluster with the highest Jaccard index prior to CCA

Before applying CCA, the mouse injected with radiotracer [^{11}C]methionine and processed with Clustering 1 presented the highest Jaccard index (Figure 18). For this mouse, the TAC of the cluster shares a similar tendency with its brain’s TAC in general, but with increased radioactivity since timepoint 7 and remaining at a higher radioactivity level. In contrast, the TAC of manual segmentation exhibits a different trend characterized by fluctuating levels of radioactivity that rise and drop irregularly from the beginning to the end of the scanned time.

After post-processing, the tendency of the cluster's TAC changed totally, it resembled more like the TAC of manual segmentation instead of that of the brain area (Figure 18).

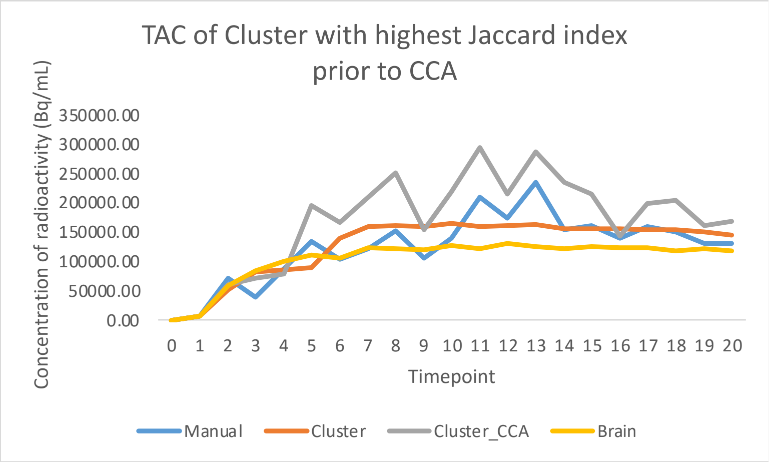


Figure 18. TACs of manual segmentation, automated segmentation prior to CCA with the highest Jaccard Index, and brain area.

5.4.1.2 TACs of the cluster with the lowest Jaccard index prior to CCA

Before CCA, the mouse injected with [^{18}F]FHC and processed PCA presented the lowest Jaccard index. The cluster of this mouse shared a similar trend with the brain but with higher radioactivity from the beginning, showing the most significant difference at time point 5. On the contrary, the manual segmentation showed an almost constant low level of radioactivity until timepoint 5, followed by a sharp increase in radioactivity at timepoint 6 and a subsequent drop to almost 0 at time point 7. After post-processing, the tendency of TAC resembles more like manual segmentation (Figure 19).

After CCA, little change was observed in the TAC of cluster_CCA, especially after timepoint 7. Cluster_CCA remains to share a similar trend with that of the brain area but with a higher concentration of radioactivity (Figure 19).

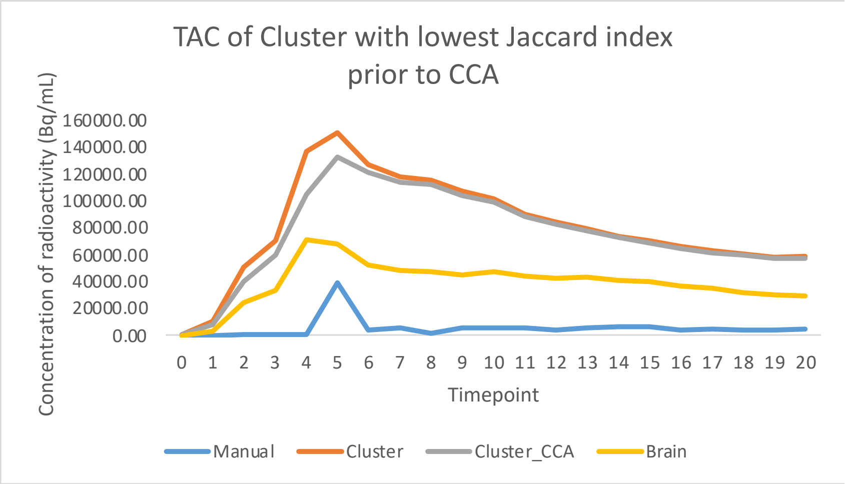


Figure 19. TACs of manual segmentation, automated segmentation prior to CCA with the lowest Jaccard Index, and brain area.

5.4.1.3 TACs of the cluster with the highest Jaccard index after CCA

Figure 20 presents the TACs of the mouse with the highest Jaccard index after post-processing. This mouse was injected with [^{11}C]methionine and was segmented by Clustering 2. Even though with a relatively high Jaccard Index of 0.5 (Table 2), the TAC of automated segmentation is not necessarily similar to the TAC of manual segmentation. Undoubtedly, the TAC of the cluster share certain similarity to manual segmentation, such as a gradual increase until timepoint 4, and then followed by a more drastic increase until timepoint 5 then drastically dropped to timepoint 7. However, the highest concentration of radioactivity of the cluster is much greater than that of manual segmentation.

When we look back at the cluster_CCA prior to CCA, the TAC of the cluster share a similar tendency with cluster_CCA, but with even hugh difference with manual segmentation (Figure 20).

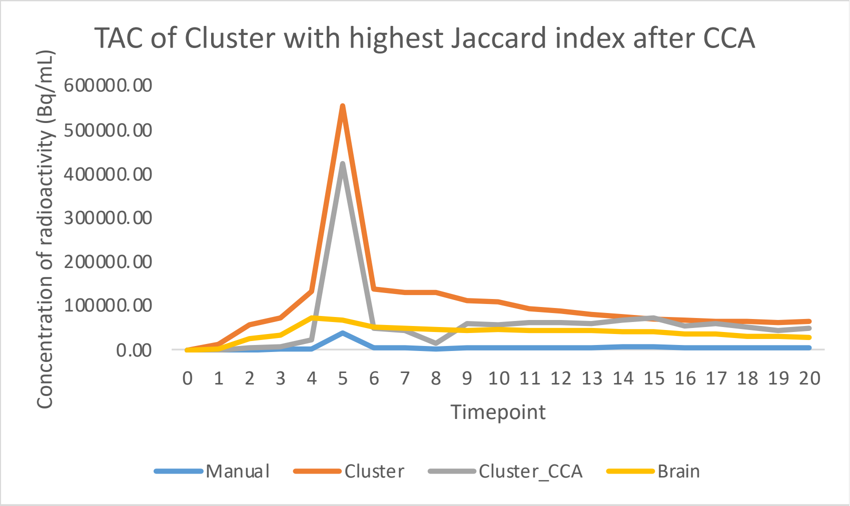


Figure 20. TACs of manual segmentation, automated segmentation with the highest Jaccard Index, and brain area after CCA.

5.4.1.4 TACs of the cluster with the lowest Jaccard index after CCA

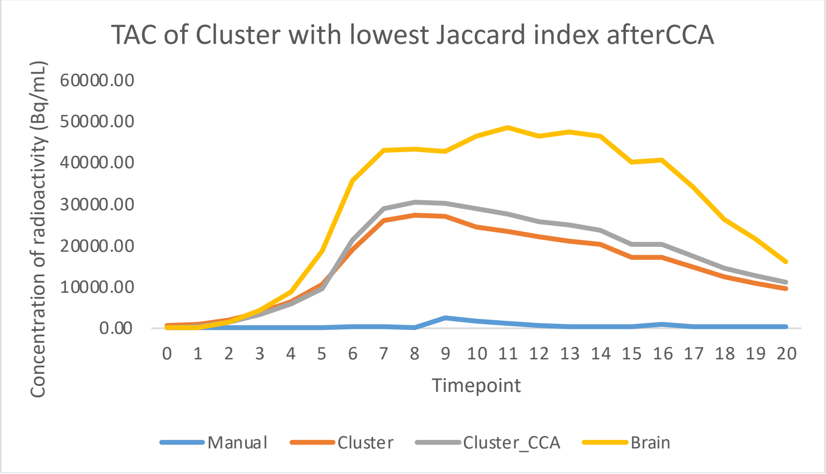


Figure 21. TACs of manual segmentation, automated segmentation with the lowest Jaccard Index, and brain area after CCA.

The mouse with the lowest Jaccard index was injected with [^{18}F]FHC and segmented by Clustering 1. As shown in Figure 21, neither the cluster nor cluster_CCA exhibit a similar tendency as the manual segmentation or brain.

5.4.1.5 TACs of the cluster with the highest Jaccard index processed by Clustering 3

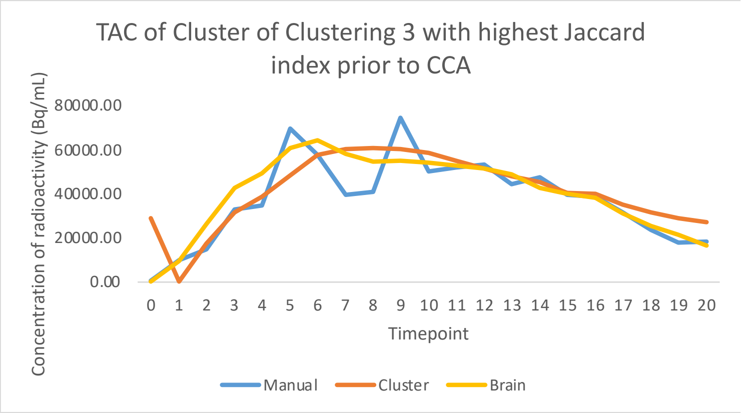


Figure 22. TACs of manual segmentation, automated segmentation with the lowest Jaccard Index, and brain area prior to CCA

Figure 22 presents the TACs of the mouse with the highest Jaccard index processed with GMM. Generally, manual segmentation shares a similar tendency with the cluster and brain, even though some unstable fluctuations are witnessed in timepoint 5 and timepoint 9. Noticeably, there’s an unusual radioactivity decrease in the cluster from timepoint 0 to timepoint 1.

6 DISCUSSION

The aims of this thesis were to:

Aim 1: evaluate the performance of different clustering methods in identifying metastatic brain tumors from dynamic brain PET images.

Aim 2: Identify the best-performing clustering method.

The statistical analysis demonstrates that Clustering 2 performs better than Clustering 1, despite the visualized clustering results (Figure 14) showing that Clustering 1 segments organs and the body structure more clearly than Clustering 2. The distinction between the two clustering methods is the pre-processing approaches, where Clustering 1 utilizes Principal Component Analysis while Clustering 2 employs Independent Component Analysis to extract features and decompose signals. The superior performance of ICA suggests that the original data, in the context of mouse PET images with brain tumors, can be effectively represented by non-Gaussian distributions. This implies that normality-based differentiation might be an effective means to distinguish normal and non-normal tissues. Moreover, the time activity curve of automatically segmented tumors in PET images extracted by ICA closely resembles that of actual tumors.

Pre-processing methods ICA and PCA have been widely used in extracting features and removing noises. The satisfactory performance of ICA is consistent with previous studies that ICA can well represent PET data (Naganawa et al., 2005), and an accuracy of 86.78% in classifying Alzheimer’s Disease signals from PET images after pre-processed by ICA (Wenlu et al., 2011). Promising results of PCA in combination with K-means clustering can also be evidenced by Hagos et al.’s study in 2018 where PCA combined with K-means clustering as an efficient and accurate method for brain tumor localization and segmentation from PET images can also (Hagos et al., 2018) However, the observation that ICA outperforms PCA in pre-processing PET images does not align with the findings of the study conducted by Razifar et al. in 2009 where PCA outperforms ICA in extracting features and reducing noises from stimulated dynamic PET images both qualitatively and quantitatively (Razifar et al., 2009). This might be explained by Pedersen et al.'s study that suggests PCA might perform poorly in separating PET data with high noise (Pedersen et al., 1994).

The time activity curve of the selected cluster closely resembles that of the manually segmented tumors. However, the TAC of the manually segmented tumors exhibits "spike-like" fluctuations, suggesting potential imprecision in the manual segmentation process where background noise might have been included. It is important to note that manual segmentation is subjective, operator-dependent, and its reliability is difficult to ascertain (Foster et al., 2014). However, the identification of automated tumors depends heavily on manual segmentation. Automatically segmented tumors would not be reliable if manual segmentation is not reliable. Further studies are needed to improve the precision of manual segmentation and enhance its reliability.

However, the study had several limitations, including a small dataset, the small size of metastatic brain tumors, and the subjective nature of manual segmentation. The first limitation of this thesis is a small dataset (19 mice PET images). Further studies with expanded datasets are needed. Another limitation is the tiny size of metastatic brain tumors and low animal PET image resolution, which makes it difficult to identify tumors from both manual segmentation and automated segmentation qualitatively. Maybe with rats' images or human images, it is easier to identify metastatic brain tumors. The last limitation is the lack of objective criteria for manual segmentation and inevitable subjectivity in manual segmentation. In this thesis, the selection of automated segmentation is highly dependent on manual segmentation, it is essential to operate manual segmentation carefully and minimize potential errors. Lastly, the orientation of PET images is a technical challenge when aligning manual and automated segmentation with different formats.

7 CONCLUSION

Clustering can be used to identify metastatic brain tumors from dynamic PET images of mice.

Among the three clustering methods, Clustering 2, which is the combination of pre-processing approach independent component analysis and K-means clustering, is the best performing with the highest accuracy and second shortest processing time.

K-means clustering outperforms the Gaussian mixture model in segmenting metastatic brain tumors with higher accuracy and shorter processing time, especially when used in combination with the pre-processing independent component analysis approach.

In addition to pre-processing, the study also found that proper post-processing can significantly improve the performance of K-means clustering methods. Specifically, connected component analysis significantly improved the accuracy of K-means clustering. Most importantly, the connected component analysis did not significantly increase the computational time.

In conclusion, this study highlights the importance of pre-processing and post-processing in improving the performance of clustering methods in identifying metastatic brain tumors from dynamic brain PET images. Future studies with larger datasets and comparative analysis with more clustering methods could further enhance the accuracy and efficiency of segmentation or tumor detection from PET images.

8 ACKNOWLEDGMENTS

I would like to express my sincere gratitude to my supervisors, Maria Jaakkola, Riku Klén, and Xiang-Guo Li, for their unwavering support and guidance during the whole process. It would not be possible to complete my master's thesis without their guidance. They have not only provided invaluable guidance with my thesis but also ignited my passion for image analysis. Most importantly, their enthusiasm for the field has shown me how incredibly rewarding and fulfilling research work can be.

I am deeply grateful to my family for their unconditional support financially and emotionally in my life.

I would also like to thank the BIMA program, also teachers, and all individuals who have contributed to making this program so interesting and enjoyable, I have been enjoying every moment of it. Especially, I would like to thank Junel Solis and Joanna Pylvänäinen for their invaluable assistance in coding and providing guidance for my career choices.

I would also like to thank my friends Marcela Rivera, Rafael Diaz, and Monika Vaitkevičiūtė for their companionship and encouragement. Pursuing a master's degree has been challenging, but sharing experiences with them make this journey so much more fun.

10 REFERENCES

Abualhaj, B. (2017). *[18F]FET-PET brain image segmentation using k-means: Evaluation of five cluster validity indices* [Dissertation]. https://doi.org/10.11588/heidok.00023560

Abualhaj, B., Weng, G., Ong, M., Attarwala, A. A., Molina, F., Büsing, K., & Glatting, G. (2017). Comparison of five cluster validity indices performance in brain [18F]FET-PET image segmentation using k-means. *Medical Physics*, *44*(1), 209–220. https://doi.org/10.1002/mp.12025

Achrol, A. S., Rennert, R. C., Anders, C., Soffietti, R., Ahluwalia, M. S., Nayak, L., Peters, S., Arvold, N. D., Harsh, G. R., Steeg, P. S., & Chang, S. D. (2019). Brain metastases. *Nature Reviews Disease Primers*, *5*(1), Article 1. https://doi.org/10.1038/s41572-018-0055-y

Alashwal, H., El Halaby, M., Crouse, J. J., Abdalla, A., & Moustafa, A. A. (2019). The Application of Unsupervised Clustering Methods to Alzheimer's Disease. *Frontiers in Computational Neuroscience*, *13*, 31. https://doi.org/10.3389/fncom.2019.00031

Alauddin, M. M. (2011). Positron emission tomography (PET) imaging with 18F-based radiotracers. *American Journal of Nuclear Medicine and Molecular Imaging*, *2*(1), 55–76.

Almuhaideb, A., Papathanasiou, N., & Bomanji, J. (2011). 18F-FDG PET/CT Imaging In Oncology. *Annals of Saudi Medicine*, *31*(1), 3–13. https://doi.org/10.4103/0256-4947.75771

Arumugam, M., & Raja, R. (2013). Spatial Fuzzy C Means PET Image Segmentation of Neurodegenerative Disorder. *Indian J. Comput. Sci. Eng.*, *4*.

Ashraf, M. A., & Goyal, A. (2023). Fludeoxyglucose (18F). In *StatPearls*. StatPearls Publishing. http://www.ncbi.nlm.nih.gov/books/NBK557653/

Barnholtz-Sloan, J. S., Sloan, A. E., Davis, F. G., Vigneau, F. D., Lai, P., & Sawaya, R. E. (2004). Incidence proportions of brain metastases in patients diagnosed (1973 to 2001) in the Metropolitan Detroit Cancer Surveillance System. *Journal of Clinical Oncology: Official Journal of the American Society of Clinical Oncology*, *22*(14), 2865–2872. https://doi.org/10.1200/JCO.2004.12.149

Baselli, G., De Bernardi, E., Soffientini, C., Gianoli, C., Faggiano, E., & Zito, F. (2016). Model based lesion segmentation in FDG-PET from raw data and clinical images. *2016 IEEE 2nd International Forum on Research and Technologies for Society and Industry Leveraging a Better Tomorrow (RTSI)*, 1–5. https://doi.org/10.1109/RTSI.2016.7740585

Bugli, C., & Lambert, P. (2007). Comparison between Principal Component Analysis and Independent Component Analysis in Electroencephalograms Modelling. *Biometrical Journal*, *49*(2), 312–327. https://doi.org/10.1002/bimj.200510285

Chen, C.-H. (2014). A hybrid intelligent model of analyzing clinical breast cancer data using clustering techniques with feature selection. *Applied Soft Computing*, *20*, 4–14. https://doi.org/10.1016/j.asoc.2013.10.024

Chen, K., & Chen, X. (2011). Positron Emission Tomography Imaging of Cancer Biology: Current Status and Future Prospects. *Seminars in Oncology*, *38*(1), 70–86. https://doi.org/10.1053/j.seminoncol.2010.11.005

Cosentino, S., Scopelliti, F., Murè, G., Baldari, S., & Ippolito, M. (2020). *11C-Methionine* (pp. 193–209). https://doi.org/10.1007/978-3-030-27779-6_11

Di Giacomo, A. M., Mair, M. J., Ceccarelli, M., Anichini, A., Ibrahim, R., Weller, M., Lahn, M., Eggermont, A. M. M., Fox, B., & Maio, M. (2023). Immunotherapy for brain metastases and primary brain tumors. *European Journal of Cancer*, *179*, 113–120. https://doi.org/10.1016/j.ejca.2022.11.012

Di Lorenzo, R., & Ahluwalia, M. S. (2017). Targeted therapy of brain metastases: Latest evidence and clinical implications. *Therapeutic Advances in Medical Oncology*, *9*(12), 781–796. https://doi.org/10.1177/1758834017736252

Dunnwald, L. K., Doot, R. K., Specht, J. M., Gralow, J. R., Ellis, G. K., Livingston, R. B., Linden, H. M., Gadi, V. K., Kurland, B. F., Schubert, E. K., Muzi, M., & Mankoff, D. A. (2011). PET Tumor Metabolism in Locally Advanced Breast Cancer Patients Undergoing Neoadjuvant Chemotherapy: Value of Static versus Kinetic Measures of Fluorodeoxyglucose Uptake. *Clinical Cancer Research*, *17*(8), 2400–2409. https://doi.org/10.1158/1078-0432.CCR-10-2649

Eichler, A. F., & Loeffler, J. S. (2007). Multidisciplinary Management of Brain Metastases. *The Oncologist*, *12*(7), 884–898. https://doi.org/10.1634/theoncologist.12-7-884

Fasihi, M. S., & Mikhael, W. B. (2016). Overview of Current Biomedical Image Segmentation Methods. *2016 International Conference on Computational Science and Computational Intelligence (CSCI)*, 803–808. https://doi.org/10.1109/CSCI.2016.0156

Gavrilovic, I. T., & Posner, J. B. (2005). Brain metastases: Epidemiology and pathophysiology. *Journal of Neuro-Oncology*, *75*(1), 5–14. https://doi.org/10.1007/s11060-004-8093-6

Hagos, Y. B., Minh, V. H., Khawaldeh, S., Pervaiz, U., & Aleef, T. A. (2018). Fast PET Scan Tumor Segmentation Using Superpixels, Principal Component Analysis and K-Means Clustering. *Methods and Protocols*, *1*(1), Article 1. https://doi.org/10.3390/mps1010007

He, L., Ren, X., Gao, Q., Zhao, X., Yao, B., & Chao, Y. (2017). The connected-component labeling problem: A review of state-of-the-art algorithms. *Pattern Recognition*, *70*, 25–43. https://doi.org/10.1016/j.patcog.2017.04.018

Hyvärinen, A. (2013). Independent component analysis: Recent advances. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, *371*(1984), 20110534. https://doi.org/10.1098/rsta.2011.0534

Kang, W.-X., Yang, Q.-Q., & Liang, R.-P. (2009). The Comparative Research on Image Segmentation Algorithms. *2009 First International Workshop on Education Technology and Computer Science*, *2*, 703–707. https://doi.org/10.1109/ETCS.2009.417

Khan, M. U., Khan, S., El-Refaie, S., Win, Z., Rubello, D., & Al-Nahhas, A. (2009). Clinical indications for Gallium-68 positron emission tomography imaging. *European Journal of Surgical Oncology (EJSO)*, *35*(6), 561–567. https://doi.org/10.1016/j.ejso.2009.01.007

Langen, K.-J., Galldiks, N., Hattingen, E., & Shah, N. J. (2017). Advances in neuro-oncology imaging. *Nature Reviews Neurology*, *13*(5), 279–289. https://doi.org/10.1038/nrneurol.2017.44

Lee, J. S., Lee, D. S., Ahn, J. Y., Cheon, G. J., Kim, S. K., Yeo, J. S., Seo, K., Park, K. S., Chung, J. K., & Lee, M. C. (2001). Blind separation of cardiac components and extraction of input function from H(2)(15)O dynamic myocardial PET using independent component analysis. *Journal of Nuclear Medicine: Official Publication, Society of Nuclear Medicine*, *42*(6), 938–943.

Lin, N. U., Bellon, J. R., & Winer, E. P. (2004). CNS Metastases in Breast Cancer. *Journal of Clinical Oncology*, *22*(17), 3608–3617. https://doi.org/10.1200/JCO.2004.01.175

Linnert, M., Iversen, H. K., & Gehl, J. (2012). Multiple brain metastases—Current management and perspectives for treatment with electrochemotherapy. *Radiology and Oncology*, *46*(4), 271–278. https://doi.org/10.2478/v10019-012-0042-y

Louis, D. N., Perry, A., Wesseling, P., Brat, D. J., Cree, I. A., Figarella-Branger, D., Hawkins, C., Ng, H. K., Pfister, S. M., Reifenberger, G., Soffietti, R., von Deimling, A., & Ellison, D. W. (2021). The 2021 WHO Classification of Tumors of the Central Nervous System: A summary. *Neuro-Oncology*, *23*(8), 1231–1251. https://doi.org/10.1093/neuonc/noab106

Lu, W., Wang, J., & Zhang, H. H. (2015). Computerized PET/CT image analysis in the evaluation of tumour response to therapy. *The British Journal of Radiology*, *88*(1048), 20140625. https://doi.org/10.1259/bjr.20140625

M, M., & P, S. (2018). MRI Brain Tumour Segmentation Using Hybrid Clustering and Classification by Back Propagation Algorithm. *Asian Pacific Journal of Cancer Prevention : APJCP*, *19*(11), 3257–3263. https://doi.org/10.31557/APJCP.2018.19.11.3257

Mabray, M. C., Barajas, R. F., & Cha, S. (2015). Modern Brain Tumor Imaging. *Brain Tumor Research and Treatment*, *3*(1), 8–23. https://doi.org/10.14791/btrt.2015.3.1.8

Madhulatha, T. S. (2012). *An Overview on Clustering Methods* (arXiv:1205.1117). arXiv. http://arxiv.org/abs/1205.1117

Martiniova, L., Palatis, L. D., Etchebehere, E., & Ravizzini, G. (n.d.). Gallium-68 in Medical Imaging. *Current Radiopharmaceuticals*, *9*(3), 187–207.

Patcheil, R. A. (1995). Metastatic Brain Tumors. *Neurologic Clinics*, *13*(4), 915–925. https://doi.org/10.1016/S0733-8619(18)30025-2

Pham, D. L., Xu, C., & Prince, J. L. (2000). Current Methods in Medical Image Segmentation. *Annual Review of Biomedical Engineering*, *2*(1), 315–337. https://doi.org/10.1146/annurev.bioeng.2.1.315

Prince, J. L., & Links, J. (2006). *Medical imaging signals and systems*. Prentice Hall.

Rahmim, A.,Lodge, M. A., Karakatsanis, N. A., Panin, V. Y., Zhou, Y., McMillan, A., Cho, S., Zaidi, H., Casey, M. E., & Wahl, R. L. (2019). Dynamic whole-body PET imaging: Principles, potentials and applications. *European Journal of Nuclear Medicine and Molecular Imaging*, *46*(2), 501–518. https://doi.org/10.1007/s00259-018-4153-6

Razifar, P., Muhammed, H. H., Engbrant, F., Svensson, P.-E., Olsson, J., Bengtsson, E., Långström, B., & Bergström, M. (2009). Performance of principal component analysis and independent component analysis with respect to signal extraction from noisy positron emission tomography data—A study on computer simulated images. *The Open Neuroimaging Journal*, *3*, 1–16. https://doi.org/10.2174/1874440000903010001

Selvy, P. T., Palanisamy, V., & Purusothaman, T. (n.d.). *Performance Analysis of Clustering Algorithms in Brain Tumor Detection of MR Images*. 10.

Sharma, N., & Aggarwal, L. M. (2010). Automated medical image segmentation techniques. *Journal of Medical Physics / Association of Medical Physicists of India*, *35*(1), 3–14. https://doi.org/10.4103/0971-6203.58777

Singhal, T., Narayanan, T. K., Jacobs, M. P., Bal, C., & Mantil, J. C. (2012). 11C-methionine PET for grading and prognostication in gliomas: A comparison study with 18F-FDG PET and contrast enhancement on MRI. *Journal of Nuclear Medicine: Official Publication, Society of Nuclear Medicine*, *53*(11), 1709–1715. https://doi.org/10.2967/jnumed.111.102533

Terakawa, Y., Tsuyuguchi, N., Iwai, Y., Yamanaka, K., Higashiyama, S., Takami, T., & Ohata, K. (2008). Diagnostic accuracy of 11C-methionine PET for differentiation of recurrent brain tumors from radiation necrosis after radiotherapy. *Journal of Nuclear Medicine: Official Publication, Society of Nuclear Medicine*, *49*(5), 694–699. https://doi.org/10.2967/jnumed.107.048082

Varma, D. R. (2012). Managing DICOM images: Tips and tricks for the radiologist. *The Indian Journal of Radiology & Imaging*, *22*(1), 4–13. https://doi.org/10.4103/0971-3026.95396

Verger, A., Arbizu, J., & Law, I. (2018). Role of amino-acid PET in high-grade gliomas: Limitations and perspectives. *The Quarterly Journal of Nuclear Medicine and Molecular Imaging: Official Publication of the Italian Association of Nuclear Medicine (AIMN) [and] the International Association of Radiopharmacology (IAR), [and] Section of the Society Of...*, *62*(3), 254–266. https://doi.org/10.23736/S1824-4785.18.03092-3

Verger, A., Kas, A., Darcourt, J., & Guedj, E. (2022). PET Imaging in Neuro-Oncology: An Update and Overview of a Rapidly Growing Area. *Cancers*, *14*(5), 1103. https://doi.org/10.3390/cancers14051103

Watase, C., Shiino, S., Shimoi, T., Noguchi, E., Kaneda, T., Yamamoto, Y., Yonemori, K., Takayama, S., & Suto, A. (2021). Breast Cancer Brain Metastasis—Overview of Disease State, Treatment Options and Future Perspectives. *Cancers*, *13*(5), 1078. https://doi.org/10.3390/cancers13051078

Wenlu, Y., Fangyu, H., Xinyun, C., & Xudong, H. (2011). ICA-Based Automatic Classification of PET Images from ADNI Database. In B.-L. Lu, L. Zhang, & J. Kwok (Eds.), *Neural Information Processing* (pp. 265–272). Springer. https://doi.org/10.1007/978-3-642-24955-6_32

Witzel, I., Oliveira-Ferrer, L., Pantel, K., Müller, V., & Wikman, H. (2016). Breast cancer brain metastases: Biology and new clinical perspectives. *Breast Cancer Research*, *18*(1), 8. https://doi.org/10.1186/s13058-015-0665-1

Yoshinaga, K., Manabe, O., & Tamaki, N. (2018). Absolute quantification of myocardial blood flow. *Journal of Nuclear Cardiology*, *25*(2), 635–651. https://doi.org/10.1007/s12350-016-0591-3

Zhu, W., & Jiang, T. (2003). *Automation segmentation of PET image for brain tumors* (p. 2629 Vol.4). https://doi.org/10.1109/NSSMIC.2003.1352428

Mason, D. (2011). SU-E-T-33: pydicom: an open source DICOM library. Medical Physics, 38(6Part10), 3493–3493.

McKinney, W., & others. (2010). Data structures for statistical computing in python. In Proceedings of the 9th Python in Science Conference (Vol. 445, pp. 51–56).

Pedregosa, F., Varoquaux, Ga"el, Gramfort, A., Michel, V., Thirion, B., Grisel, O., … others. (2011). Scikit-learn: Machine learning in Python. Journal of Machine Learning Research, 12(Oct), 2825–2830.

Waskom, M., Botvinnik, Olga, (2017). mwaskom/seaborn: v0.8.1 (September 2017). Zenodo. https://doi.org/10.5281/zenodo.883859

Brain Tumors - Classifications, Symptoms, Diagnosis and Treatments. (n.d.). Brain Tumors - Classifications, Symptoms, Diagnosis and Treatments. https://www.aans.org/

11 APPENDIX

I categorized coding into 3 parts: automated segmentation pipeline, calculation and visualization of Jaccard index, and acquisition of TACs.

11.1 Automated segmentation pipeline

```
import os
import nibabel as nib
from time import perf_counter
import numpy as np
from sklearn.cluster import KMeans
from sklearn.mixture import GaussianMixture
from sklearn.decomposition import PCA
from sklearn.decomposition import FastICA
import matplotlib.pyplot as plt
import csv
import cc3d
import sklearn

def prepare_TACs(directory, image_id, mouse_index):

    start_time = perf_counter()
    tac_array=[]
    tac_array_inds=[]

    pet_nifti = os.path.join(directory, image_id + "_nifti.img")
    print(pet_nifti)
    image = nib.load(pet_nifti)
    image_data = image.get_fdata()

    x_dim = image_data.shape[0]
    y_dim = image_data.shape[1]
    z_dim = image_data.shape[2]
    frame_dim = image_data.shape[3]

    # Split by masking one mouse with 0's
    if mouse_index == 1:
        image_data[0:round(x_dim/2),:,:,:] = 0
    else:
        image_data[round(x_dim/2):x_dim,:,:,:] = 0
    mean_per_voxel=np.mean(image_data,axis=3)
    binary_mask_for_mean=mean_per_voxel > np.quantile(mean_per_voxel,
0.95)

    for x_ind in range(x_dim):
        for y_ind in range(y_dim):
            for z_ind in range(z_dim):
                    if binary_mask_for_mean[x_ind,y_ind,z_ind]:
                        tac_array.append(image_data[x_ind,y_ind,z_ind,:])
                        tac_array_inds.append((x_ind,y_ind,z_ind))

    end_time = perf_counter()
    print("Preparing the image took "+str(round(end_time-start_time,0))+ "
seconds"+"\n")

    return image_data, tac_array, tac_array_inds


def clustering_TACs(image_id,
image,perform_pca,perform_ica,perform_gmm,cluster_number):

    label_list=[] # stores the label arrays from each clustering method
    processing_time=[] # stores the processing times for each clustering

    #PCA+KMeans clustering
    if perform_pca:
        print("PCA+KM")
        PCA_start = perf_counter() # PCA_start and PCA_end record time spent
on PCA and KMeans
        PCA_clustering =
PCA(n_components=cluster_number[0]).fit_transform(image) # Principal
Component Analysis
        PCA_labels =
KMeans(n_clusters=cluster_number[0]).fit_predict(PCA_clustering) # KMeans
clustering
        PCA_end = perf_counter()
        print(str(round(PCA_end-PCA_start,0))+" seconds"+"\n")
        processing_time.append(round(PCA_end-PCA_start,0))
        label_list.append(PCA_labels) # save labels

    #ICA+KMeans
    if perform_ica:
        print("ICA+KM")
        ICA_start = perf_counter() # ICA_start and ICA_end record time spent on
ICA and KMeans
```

```
ICA_clustering=FastICA(n_components=cluster_number[1]).fit_transform(image
) # Independent Component Analysis

ICA_labels=KMeans(n_clusters=cluster_number[1]).fit_predict(ICA_clustering)
# KMeans clustering
        ICA_end = perf_counter()
        print(str(round(ICA_end-ICA_start,0))+" seconds"+"\n")
        processing_time.append(round(ICA_end-ICA_start,0))
        label_list.append(ICA_labels) # save labels

    #Gaussian mixture model
    if perform_gmm:
        print("GMM")
        GMM_start = perf_counter() # GMM_start and GMM_end record time
spent on PCA and KMeans

        # GMM might not converge if the cluster number is high. The clustering is
done again until it succeeds
        gmm_counter = 0
        while gmm_counter < 10:
            try:

gmm_model=GaussianMixture(n_components=cluster_number[2]).fit(np.array(i
mage)) # Gaussian mixture model clustering
                GMM_labels=gmm_model.predict(image) # extract labels from the
model
                GMM_end = perf_counter()
                print(str(round(GMM_end-GMM_start,0))+" seconds"+"\n")
                processing_time.append(round(GMM_end-GMM_start,0))
                label_list.append(GMM_labels) # save labels
                gmm_counter = 10
            except:
                # if error occurs, try again
                print('GMM was not successful')
                gmm_counter = gmm_counter + 1

    return label_list,processing_time

def run_pipeline(mouse_index):
    CCA_time = []
    image_data, tac_data, tac_data_inds = prepare_TACs(input_dir, image_id,
mouse_index)

    np.save("tac_data_inds_" + str(image_id) + "_" + str(mouse_index) +
".npy",tac_data_inds)
```

```
    number_of_clusters=[21,22,18]
    label_list,process_time =
clustering_TACs(image_id,tac_data,True,True,False,number_of_clusters)
    print(np.unique(label_list))

    # Make images
    dimx=dimy=192
    dimz=384

    # Initialise images with zeros
    # The zeros() function is used to get a new array of given shape and type,
filled with zeros.
    PCA_image=np.zeros((dimx,dimy,dimz))
    ICA_image=np.zeros((dimx,dimy,dimz))
    GMM_image=np.zeros((dimx,dimy,dimz))

    # Indices and dimensions
    if len(label_list) > 2:
        for k in range(len(tac_data_inds)):

PCA_image[tac_data_inds[k][0],tac_data_inds[k][1],tac_data_inds[k][2]]=label_li
st[0][k] + 1

ICA_image[tac_data_inds[k][0],tac_data_inds[k][1],tac_data_inds[k][2]]=label_li
st[1][k] + 1

GMM_image[tac_data_inds[k][0],tac_data_inds[k][1],tac_data_inds[k][2]]=label_
list[2][k] + 1
    else:
        for k in range(len(tac_data_inds)):

PCA_image[tac_data_inds[k][0],tac_data_inds[k][1],tac_data_inds[k][2]]=label_li
st[0][k] + 1

ICA_image[tac_data_inds[k][0],tac_data_inds[k][1],tac_data_inds[k][2]]=label_li
st[1][k] + 1

    # Save images, the eye() function is used to create a 2-D array with ones on
the diagonal and zeros elsewhere.
    # eye, identity matrix
    # define the orientation of nifti format
    PCA_CCA_start = perf_counter()
    labels_in_PCA = np.asarray(PCA_image)
    cca_PCA = cc3d.connected_components(labels_in_PCA, connectivity = 26)
    PCA_image_nifti=nib.Nifti1Image(cca_PCA,np.eye(4))
    PCA_CCA_end = perf_counter()
    print(str(round(PCA_CCA_end-PCA_CCA_start,0))+" seconds"+"\n")
    CCA_time.append(round(PCA_CCA_end-PCA_CCA_start,0))


    nib.save(PCA_image_nifti,os.path.join(output_dir, str(image_id) + "_mouse_"
+ str(mouse_index) +'_pca.nii'))

    ICA_CCA_start = perf_counter()
    labels_in_ICA = np.asarray(ICA_image)
    cca_ICA = cc3d.connected_components(labels_in_ICA, connectivity = 26)
    ICA_image_nifti=nib.Nifti1Image(cca_ICA,np.eye(4))
    ICA_CCA_end = perf_counter()
    print(str(round(ICA_CCA_end-ICA_CCA_start,0))+" seconds"+"\n")
    CCA_time.append(round(ICA_CCA_end-ICA_CCA_start,0))

    nib.save(ICA_image_nifti,os.path.join(output_dir, str(image_id) + "_mouse_"
+ str(mouse_index) +'_ica.nii'))

    plt.imshow(PCA_image[130,:,:])
    plt.imshow(ICA_image[130,:,:])


    if np.max(GMM_image) > 1:
        GMM_CCA_start = perf_counter()
        labels_in_GMM = np.asarray(GMM_image)
        cca_GMM = cc3d.connected_components(labels_in_GMM, connectivity =
26)
        GMM_image_nifti = nib.Nifti1Image(cca_GMM,np.eye(4))
        GMM_CCA_end = perf_counter()
        print(str(round(GMM_CCA_end-GMM_CCA_start,0))+" seconds"+"\n")
        CCA_time.append(round(GMM_CCA_end-GMM_CCA_start,0))

        nib.save(GMM_image_nifti,os.path.join(output_dir, str(image_id) +
"_mouse_" + str(mouse_index) +'_gmm.nii'))

        plt.imshow(GMM_image[130,:,:])

    return label_list,process_time,CCA_time

input_dir = 'Pipeline/PET nifti/'
output_dir = 'Pipeline3\Clustering1'
output_dir_tac = 'Pipeline3\tac_data_inds1'
image_ids = ["P123173", "P123177", "P123179", "P123964", "P123966",
"P124258", "P124260"]
# image_ids = ["P123173"]
# save as cvs
# fields = image_ids
total_time = {}
for i in range(len(image_ids)):
    image_id = image_ids[i]
    time = []
    for mouse_index in [0,1]:
        print(f"We are now clustering {image_id}, mouse {mouse_index}.")
        label_list, processing_time,CCA_time = run_pipeline(mouse_index)
        print(CCA_time)
        time.append(processing_time)
    total_time.setdefault(image_id,[]).append(time)
```

11.2 Calculation and visualization of Jaccard index

```
import os
import fnmatch
import numpy as np
import matplotlib.pyplot as plt
import nibabel as nib
import seaborn as sb
import json

def calculate_jaccard(clusters, manual_segmentation):

    # Check that the manual segmentation and clustering results have the same
dimensions
    if clusters.shape != manual_segmentation.shape:
        print("Dimensions of clustering results and manual segmentation do not
match, please check!")

    # This is just to make sure that the manual segmentation really includes only
0 and 1
    # np.where returns the indices of elements in an input array where the given
condition is satisfied
    manual_segmentation_index = np.where(manual_segmentation > 0.5)
    mask_manual = np.zeros(clusters.shape, dtype=np.int32)
    mask_manual[manual_segmentation_index] = np.int32(1)

    # Check if manual segmentation and clustered voxels overlap
    tumor_labels = clusters[np.where(manual_segmentation > 0.2)]
    if np.any(tumor_labels > 0.5):
        # Detect cluster labels that overlap with manual segmentation
        present_labels = clusters[manual_segmentation_index]
        potential_labels = np.unique(present_labels) # Find the unique elements of
an array

        # Calculate Jaccards for all labels overlapping with the organ
        max_jaccard = 0
        max_label = 0
        for j in potential_labels:

            # Create a 3D array filled with 0, but 1 in voxels having cluster label j
            label_index = np.where(clusters == j)
            mask_label = np.zeros(clusters.shape, dtype=np.int32)
            mask_label[label_index] = np.int32(1)

            # Calculate Jaccard
            mask_sum = mask_manual + mask_label
            jaccard = len(np.where(mask_sum == np.int32(2))[0]) /
len(np.where(mask_sum > 0.1)[0])
            print(len(np.where(mask_sum == np.int32(2))[0]))

            # If this cluster has better Jaccard index, make it into the main candidate
            if jaccard > max_jaccard:
                max_jaccard = jaccard
                max_label = j
    else:
        max_jaccard, max_label = np.nan, np.nan


    return max_jaccard, max_label

def plot_results(pet, clusters, manual, use_axis, use_cluster, path_output):

    fig = plt.figure(figsize=(8, 5))  # Modify if needed

    # Select the slice to plot
    manual_sums = np.sum(manual, axis=tuple(np.setdiff1d((0, 1, 2), use_axis)))
    use_slice = np.argmax(manual_sums)

    # Plot the underlying PET image in gray scale
    pet = np.sum(pet, axis=3)
    if use_axis == 0:
        pet = pet[use_slice, :, :]
    elif use_axis == 1:
        pet = pet[:, use_slice, :]
    else:
        pet = pet[:, :, use_slice]
    sb.heatmap(pet, cbar=False, xticklabels=False, yticklabels=False,
cmap=sb.dark_palette("white", as_cmap=True))

    # Keep only the selected cluster and flatten to 2D
    cluster = np.zeros(clusters.shape)
    cluster[np.where(clusters == use_cluster)] = 1
    if use_axis == 0:
        cluster = cluster[use_slice, :, :]
    elif use_axis == 1:
        cluster = cluster[:, use_slice, :]
    else:
        cluster = cluster[:, :, use_slice]
    cluster[np.where(cluster == 0)] = np.nan

    # Add the selected cluster to the plot
    sb.heatmap(cluster, cbar=False, xticklabels=False, yticklabels=False,
cmap=sb.dark_palette('cyan', as_cmap=True), alpha=0.5)

    # Flatten manual segmentation to 2D
    manual_3d = np.zeros(manual.shape)
    manual_3d[np.where(manual > 0.2)] = 1
    if use_axis == 0:
        manual_2d = manual_3d[use_slice, :, :]
    elif use_axis == 1:
        manual_2d = manual_3d[:, use_slice, :]
    else:
        manual_2d = manual_3d[:, :, use_slice]
    manual_2d[np.where(manual_2d == 0)] = np.nan

    # Add the manual segmentation to the plot
    sb.heatmap(manual_2d, cbar=False, xticklabels=False, yticklabels=False,
cmap=sb.dark_palette('red', as_cmap=True), alpha=0.5)

    # Save image
    plt.savefig(path_output)



def jaccard_pipeline(work_dir, image_ids, clustering_list):

    data = {}

    for i in range(len(image_ids)):
        image_id = image_ids[i]
        mouse_data = []
        for mouse_index in [0,1]:
            clustering_methods= []
            jaccard_index = []
            labels = []

            print(f'Now, {image_id}, mouse {mouse_index} is being processed.')

            nifti_pet = os.path.join(work_dir, 'PET nifti', image_id + '_nifti.img' )
            nifti_manual =  os.path.join(work_dir, 'Manual segmentation', image_id +
' manual segmentation tumor.img' )

            img_pet = nib.load(nifti_pet)
            pet = img_pet.get_fdata()
            img_manual = nib.load(nifti_manual)
            manual = img_manual.get_fdata()

            x_dim = pet.shape[0]

            if mouse_index == 1:
                pet[0:round(x_dim/2),:,:,:] = 0
                manual[0:round(x_dim/2),:,:] = 0
            else:
                pet[round(x_dim/2):x_dim,:,:,:] = 0
                manual[round(x_dim/2):x_dim,:,:] = 0

            manual[np.where(manual < 0.2)] = 0
            manual[np.where(manual > 0.2)] = 1

            # reading clustering files
            for file in clustering_list:
                if image_id in file:
                    if 'mouse_' + str(mouse_index) in file:
                        clustering_methods.append(file)
            print(f'For {image_id}, mouse {mouse_index}, clustering methods are
{clustering_methods}.')

            for val in range(len(clustering_methods)):
                clustering_method = clustering_methods[val]
                nifti_clustering = os.path.join(work_dir, 'Clustering',
clustering_method)

                img_clustering = nib.load(nifti_clustering)
                clustering = img_clustering.get_fdata()
                max_jaccard, max_label = calculate_jaccard(clustering, manual)
            jaccard_index.append(max_jaccard)
            labels.append(max_label)

            plot_results(pet, clustering, manual, 0, max_label,
os.path.join(work_dir, 'test', image_id + "_mouse_" + str(mouse_index) + '_' +
clustering_methods[val][16:20] + "_axis_0" + ".png"))
            plot_results(pet, clustering, manual, 1, max_label,
os.path.join(work_dir, 'test', image_id + "_mouse_" + str(mouse_index) + '_' +
clustering_methods[val][16:20] + "_axis_1" + ".png"))
            plot_results(pet, clustering, manual, 2, max_label,
os.path.join(work_dir, 'test', image_id + "_mouse_" + str(mouse_index) + '_' +
clustering_methods[val][16:20] + "_axis_2" + ".png"))

        print(f'And their corresponding jaccard indexes are {jaccard_index}, and
max labels are {labels}.')

        data_dict = {
            "jaccard" : jaccard_index,
            "labels" : labels
            }

        mouse_data.append(data_dict)


        data.setdefault(image_id, []).append(data_dict)

    return data

work_dir2 = 'Pipeline2'
image_ids = ["P123173", "P123177", "P123179", "P123964", "P123966",
"P124258", "P124260"]
clustering_list = fnmatch.filter(os.listdir(os.path.join(work_dir2, 'Clustering')),
'*.nii')
data = jaccard_pipeline(work_dir2, image_ids, clustering_list)
```

11.3 Acquisition of TACs

```
import numpy as np
import nibabel as nib
import matplotlib.pyplot as plt
import pandas as pd
import os
import csv
```

```
# prior to CCA: highest: P123966 mouse 1, PCA; lowest: P123173 mouse0,
PCA
# After CCA: highest P123173, mouse 0, ICA; lowest: P1234260 mouse0, pca


def TAC_linechart(pet, cluster, manual, brain, mouse_index, max_label):

    # PET
    image_pet = nib.load(pet)
    image_pet_data = image_pet.get_fdata()

    x_dim = 192

    if mouse_index == 1:
        image_pet_data[0:round(x_dim/2),:,:] = 0
    else:
        image_pet_data[round(x_dim/2):x_dim,:,:] = 0

    # Manual
    image_manual = nib.load(manual)
    image_manual_data = image_manual.get_fdata()

    if mouse_index == 1:
        image_pet_data[0:round(x_dim/2),:] = 0
    else:
        image_pet_data[round(x_dim/2):x_dim,:] = 0

    # Cluster
    image_cluster = nib.load(cluster)
    image_cluster_data = image_cluster.get_fdata()

    # find the cluster with highest jaccard index
    # label_index = np.where(image_cluster_data == max_label)
    label_index = np.where((max_label - 0.5 < image_cluster_data) &
(image_cluster_data < max_label + 0.5))
    mask_label = np.zeros(image_cluster_data.shape, dtype=np.int32)
    mask_label[label_index] = 1#np.int32(1)

    # find brain
    image_brain = nib.load(brain)
    image_brain_data = image_brain.get_fdata()
    x_dim = int(image_brain_data.shape[1])

    if mouse_index == 1:
        image_brain_data[0:round(x_dim/2),:,:] = 0
    else:
        image_brain_data[round(x_dim/2):x_dim,:,:] = 0


    tac_manual = []
    tac_cluster = []
    tac_brain = []
    for val in range(image_pet.shape[3]):
        pet_time = image_pet_data[:, :, :, val]
        tac_manual.append(np.mean(pet_time[np.where(image_manual_data >
0.1)]))
        tac_cluster.append(np.mean(pet_time[np.where(mask_label > 0.1)]))
        tac_brain.append(np.mean(pet_time[np.where(image_brain_data > 0.1)]))

    return tac_manual, tac_cluster, tac_brain

def TAC_linechart_CCA(pet, cluster, cluster_CCA, manual, brain,
mouse_index, max_label, max_label_CCA):

    # PET
    image_pet = nib.load(pet)
    image_pet_data = image_pet.get_fdata()

    x_dim = 192

    if mouse_index == 1:
        image_pet_data[0:round(x_dim/2),:,:] = 0
    else:
        image_pet_data[round(x_dim/2):x_dim,:,:] = 0

    # Manual
    image_manual = nib.load(manual)
    image_manual_data = image_manual.get_fdata()

    if mouse_index == 1:
        image_pet_data[0:round(x_dim/2),:] = 0
    else:
        image_pet_data[round(x_dim/2):x_dim,:] = 0

    # Cluster
    image_cluster = nib.load(cluster)
    image_cluster_data = image_cluster.get_fdata()

    # find the cluster with highest jaccard index
    # label_index = np.where(image_cluster_data == max_label)
    label_index = np.where((max_label - 0.5 < image_cluster_data) &
(image_cluster_data < max_label + 0.5))
```

```
    mask_label = np.zeros(image_cluster_data.shape, dtype=np.int32)
    mask_label[label_index] = 1#np.int32(1)

    # Cluster after CCA
    image_cluster_CCA = nib.load(cluster_CCA)
    image_cluster_CCA_data = image_cluster_CCA.get_fdata()

    # find the cluster with highest jaccard index
    # label_index = np.where(image_cluster_data == max_label)
    label_index_CCA = np.where((max_label_CCA - 0.5 <
image_cluster_CCA_data) & (image_cluster_CCA_data < max_label_CCA +
0.5))
    mask_label_CCA = np.zeros(image_cluster_CCA_data.shape,
dtype=np.int32)
    mask_label_CCA[label_index_CCA] = 1#np.int32(1)

    # find brain
    image_brain = nib.load(brain)
    image_brain_data = image_brain.get_fdata()
    x_dim = int(image_brain_data.shape[1])

    if mouse_index == 1:
        image_brain_data[0:round(x_dim/2),:,:] = 0
    else:
        image_brain_data[round(x_dim/2):x_dim,:,:] = 0


    tac_manual = []
    tac_cluster = []
    tac_cluster_CCA = []
    tac_brain = []
    for val in range(image_pet.shape[3]):
        pet_time = image_pet_data[:, :, :, val]
        tac_manual.append(np.mean(pet_time[np.where(image_manual_data >
0.1)]))
        tac_cluster.append(np.mean(pet_time[np.where(mask_label > 0.1)]))
        tac_brain.append(np.mean(pet_time[np.where(image_brain_data > 0.1)]))
        tac_cluster_CCA.append(np.mean(pet_time[np.where(mask_label_CCA >
0.1)]))

    return tac_manual, tac_cluster, tac_cluster_CCA, tac_brain

tac_manual1, tac_cluster1, tac_cluster1_CCA, tac_brain1 =
TAC_linechart_CCA(pet1, cluster1, cluster1_CCA, manual1, brain1,
mouse_index1, max_label1, max_label1_CCA)
```

```
dict1 = {'Manual' : tac_manual1,
        'Cluster': tac_cluster1,
        'Cluster_CCA': tac_cluster1_CCA,
        'Brain': tac_brain1}

df = pd.DataFrame(dict1)
# saving the dataframe
df.to_csv('data1.csv')
```