Masters thesis
Master's program in materials research
Medical physics and biophysics

Determining the partial volumes of brain tissue from spectroscopy voxels

Johannes Halkoaho

February 9, 2022

Supervisor(s): Antti Hakkarainen, M.Sc(Tech.), Sauli Savolainen, prof

Examiner(s): FT Marjut Timonen, Sauli Savolainen, prof

University of Helsinki
Faculty of Science

PL 64 (Gustaf Hällströmin katu 2a)
00014 Helsingin yliopisto

HELSINGIN YLIOPISTO — HELSINGFORS UNIVERSITET — UNIVERSITY OF HELSINKI

Tiedekunta — Fakultet — Faculty	Koulutusohjelma — Utbildningsprogram — Degree programme
Faculty of Science	Master's program in materials research Medical physics and biophysics

Tekijä — Författare — Author

Johannes Halkoaho

Työn nimi — Arbetets titel — Title

Determining the partial volumes of brain tissue from spectroscopy voxels

Työn laji — Arbetets art — Level	Aika — Datum — Month and year	Sivumäärä — Sidantal — Number of pages
Masters thesis	February 9, 2022	39

Tiivistelmä — Referat — Abstract

MRS or magnetic resonance spectroscopy is an imagining technique which can be used to gain information about the metabolite concentration within a certain volume of interest. This can be used for example in brain imagining.

The brain consists of three main types of tissue: cerebrospinal fluid, white and gray matter. It is important to know the different volume fractions of these tissues as the resolution in MRS is significantly lower than that of magnetic resonance imagining (MRI). The tissues all have different metabolite profiles and in order to get meaningful data the volume fractions need to be taken into account. This information can be gained from the segmentation of an image formed by using MRI.

In this work a software tool was created to find these volume fractions with the input of a .rda file that is created by the scanner and Nifti file. The Nifti file is the image formed by using MRI and the .rda file is the manufacturers raw data format for spectroscopy data which has the relevant information about the volumes of interest. The software tool was created using Python and JavaScript programming languages and different functions of FSL. FSL is a comprehensive library of analysis tools used in brain imaging data processing.

The steps for the software tool are: determining the coordinates of the volume of interest in FSL voxel coordinates, creating a mask in the correct orientation and location, removing non-brain tissue from the image using FSL's tool tailored for that purpose (BET), segmenting the image using FSL's segmenting tool (FAST), registering the mask on the segmented images and calculating the volume fractions.

The software tool was tested on imaging data that was obtained at Meilahti Kolmiosairaala for the purpose of the testing. The testing data set included five different spectroscopy volumes from different parts of the brain and a T_1 weighted image. The software tool was given the relevant information about the volume of interest in the form of a .rda file and the T_1 weighted image in the form of a Nifti file. The software tool then determined the different volume fractions from all of the five volumes of interest. There is variation on the volume fraction of different brain areas within different brains and it is not possible to have an absolute reference value. The results of the test corresponded to the possible volume fractions that can be expected from the volumes in question.

Avainsanat — Nyckelord — Keywords

LaTeX

Säilytyspaikka — Förvaringsställe — Where deposited

Muita tietoja — Övriga uppgifter — Additional information

HELSINGIN YLIOPISTO — HELSINGFORS UNIVERSITET — UNIVERSITY OF HELSINKI

Tiedekunta — Fakultet — Faculty	Koulutusohjelma — Utbildningsprogram — Degree programme
Matemaattis-luonnontieteellinen tiedekunta	Materiaalitutkimuksen maisteriohjelma Lääketieteellinen fysiikka ja biofysiikka

Tekijä — Författare — Author

Johannes Halkoaho

Työn nimi — Arbetets titel — Title

Determining the partial volumes of brain tissue from spectroscopy voxels

Työn laji — Arbetets art — Level	Aika — Datum — Month and year	Sivumäärä — Sidantal — Number of pages
Maisterin tutkielma	February 9, 2022	39

Tiivistelmä — Referat — Abstract

MS tai magneettispektroskopia on kuvantamistekniikka, jolla saadaan tietoa kudosten metaboliittikonsentraatioista tietystä tilavuudesta. Näistä konsentraatioarvoista voidaan tehdä potilaan diagnostiikkaan vaikuttavia johtopäätöksiä. Tekniikkaa voidaan käyttää myös aivokuvantamisessa.

Aivot koostuvat pääasiassa kolmesta kudoksesta, jotka ovat selkäydinneste, valkoinen ja harmaa aine. MS:n kannalta on tärkeä tietää näiden aivokudosten suhteelliset tilavuusosuudet, koska tekniikan resoluutio on huomattavasti pienempi kuin perinteisen magneettikuvantamisen. Jokaisella kudostyypillä on erilainen metaboliittiprofiili. Saadakseen merkityksellistä dataa, on otettava huomioon kudosten suhteelliset määrät. Tämä tieto voidaan saada segmentoimalla magneettikuvauksella muodostetut kuvat.

Tässä työssä luotiin ohjelma, joka pystyy määrittämään suhteelliset tilavuusosuudet kuvauslaitteen tuottamasta .rda-tiedostosta ja Nifti-tiedostosta. Nifti-tiedosto on magneettikuvauksella muodostettu kolmiulotteinen kuvatiedosto ja .rda-tiedosto on valmistajan raakadataformaatti spektroskopiadatalle, jossa on tarvittava informaatio valitusta tilavuudesta. Ohjelma on toteutettu käyttäen Python ja JavaScript ohjelmointikieliä sekä FSL:n eri toimintoja. FSL on kattava kirjasto analyysityökaluja, joka on tarkoitettu aivojen kuvantamisdatan käsittelyyn.

Ohjelman toimintavaiheet ovat: valitun tilavuuden koordinaattien määrittäminen FSL:n vokseli koordinaateissa, maskin luominen oikeaan orientaatioon ja sijaintiin, aivokudoksen erottaminen käyttäen FSL:n siihen tarkoitettua työkalua (BET), kuvan segmentointi käyttäen FSL:n segmentointi työkalua (FAST), maskin rekisteröinti segmentoituihin kuviin ja suhteellisten tilavuuksien laskeminen.

Ohjelmaa testattiin Meilahden Kolmiosairaalassa tätä työtä varten kuvattuun dataan. Dataan kuului viisi eri spektroskopiatilavuutta eri osista aivoja ja T_1-painoitteinen aivokuva. Ohjelmalle annettiin tarvittavat tiedot .rda-tiedoston muodossa ja T_1-painoitteinen aivokuva Nifti-tiedoston muodossa. Ohjelma määritti näiden tiedostojen perusteella suhteelliset tilavuusosuudet jokaisesta viidestä tilavuudesta. Aivojen eri alueiden suhteellisissa tilavuusosuuksissa on yksilöllisiä vaihteluita, joten absoluuttista vertailuarvoa ei voi määrittää. Tulokset vastasivat valittujen tilavuuksien mahdollisia suhteellisia tilavuusosuuksia. Kehitettyä ohjelmaa voidaan käyttää tutkimuksissa, jotka tarvitsevat aivokudosten suhteelliset tilavuusosuudet spektroskopiatilavuudesta. Kliinisen käytön mahdollisuus vaatisi ohjelmalta DICOM-standardi tukea.

Avainsanat — Nyckelord — Keywords

LaTeX

Säilytyspaikka — Förvaringsställe — Where deposited

Muita tietoja — Övriga uppgifter — Additional information

Contents

1. Introduction

The physical phenomenon called nuclear magnetic resonance (NMR) was first described by Isidor Rabi in 1938[17] as a part of a new method for measuring nuclear magnetic moments of molecules. It took until the year 1973 for the first publication about image using the method was published [12].

The method for taking images using NMR was later named magnetic resonance imaging (MRI) and later another imaging technique was developed called magnetic resonance spectroscopy (MRS).

These techniques are used in brain imagining in order to get information about the health of the subject. The images formed from MRI need to be processed before any detailed information can be gained. The processing steps for these images include image preprocessing and image segmentation.

For MRS it is important to know the different volume fractions of the brain tissues present as the resolution for MRS is significantly worse than the resolution of the image formed by MRI. This information can be gained from the segmentation of the image formed by MRI.

In this work a software tool is created to find these volume fractions of different volumes of interest (VOI) with the input of a .rda file which has the relevant information about the volumes and .nii file which is the image formed by MRI. The software tool is tested on imaging data that was obtained at Meilahti Kolmiosairaala for the purpose of this work.

The purpose of the software tool is to make it simpler and faster to gain the different volume fractions from spectroscopy volumes as due to the variations in volume fractions between subject the process has to be done in each case separately. The purpose of this work is to explain the relevant theory, the methods for the creation of the software tool and testing whether the software tool gives reasonable results.

Table 1.1: List of abbreviations

List of abbreviations	
NMR	Nuclear magnetic resonance
MRI	Magnetic resonance imagining
MRS	Magnetic resonance spectroscopy
VOI	Volume of interest
CSF	Cerebrospinal fluid
WM	White matter
GM	Gray matter
MP-RAGE	Magnetization prepared rapid gradient echo
PRESS	Point Resolved Spectroscopy
TFE	Turbo field echo
T_E	Echo time
T_R	Repetition time
BET	Brain extraction tool
CNN	Convolutional neural network
FAST	FMRIB's Automated Segmentation Tool

Table 1.2: List of used data formats

Used data formats	
.rda	Input file for the software tool. Contains spectroscopy voxel coordinate and orientation informa
.nii	Nifti file format and a input file for the software tool. Contains the neuroimaging data set.

2. Theory

2.1 Nuclear magnetic resonance

The main physical phenomenon responsible for the signals detected in MRI and MRS is called nuclear magnetic resonance (NMR). The basis work for reviewing the theory of NMR is [3] and [14].

2.1.1 Atomic nuclei in a constant magnetic field

All protons have an intrinsic property called the spin which can be compared to the classical angular momentum of a spinning sphere for proton the value is 1/2.

The spin and the magnetic moment have the following relation:

$$\vec{\mu} = \gamma \vec{S}$$

where γ is the gyromagnetic ration. So all atomic nuclei that have a non-zero spin have a non-zero magnetic moment.

With the addition of the constant magnetic field the atomic nuclei will begin to feel a torque that will attempt to align the magnetic dipole moments along the magnetic lines. But as the spin is comparable to the angular momentum the conservation of angular momentum applies and will stop the moment from aligning with the magnetic field. Instead the magnetic dipole moment will precess about the axis of the magnetic field with a frequency called the larmor frequency.

$$\omega = -\gamma B_0$$

In addition to this the spin has two different state (1/2 and -1/2) and these states have a energy difference of:

$$\Delta E = \gamma \hbar B_0 = 2\vec{\mu} \cdot \vec{B}_0$$

which will cause one state to be high energy and the other lower energy.

2.1.2 Radiating the system with radio waves

If the system is radiated with radio waves with a frequency of the Larmor frequency the small part of the lower energy spin states will flip to higher energy states. Causing the total magnetic moment to be more in the negative z-direction. The radio waves also cause the nuclei to precess in alignment with each other creating a x-y component for the total magnetic moment vector. As this process requires the right frequency it is called nuclear magnetic resonance.

2.2 MRI and MRS

Both MRI and MRS are based on nuclear magnetic resonance. MRS uses same basic principles as MRI and the difference is in the source for the signal.

2.2.1 T_1 and T_2 relaxation

As radiating of the system with radiowaves stops the systems total magnetic moment will start to decay to its original value. There are two different mechanisms for this decay called T_1 relaxation and T_2 relaxation [14].

T_1 relaxation is also called spin-lattice relaxation. During T_1 relaxation the nuclei that were exited into a higher energy state start to relax back into a lower energy state. And as that happens the total magnetic moment vector starts to be realigned with the outer magnetic field. This change in magnetic moment induces a current in a coil that can be measured.

T_2 relaxation is also called spin-spin relaxation. During this process the nuclei that were precessing in alignment due to the radiofrequency pulse start to fall out of alignment with each other and canceling the x-y plane components of the magnetic moment vector. This changes the magnetic moment vector and can be measured with the detector.

2.2.2 T_1 and T_2 weighted images

During the T_1 and T_2 relaxations the total magnetic moment vector changes and the change can be measured by a receiver coil. The relaxation processes are both independent of each other and simultaneous. The time it takes differs depending on the tissue. The relaxation time for the processes are referred to as T_1 and T_2 relaxation times. The definition of T_1 relaxation time is the time it takes for the longitudinal magnetization to reach 63% of its original value [14] and for T_2 relaxation time it is the time it takes for the transverse magnetization to decay to 37% of its initial value [14].

T_1 weighted image refers to an image constructed from the signal produced by the

decay of longitudinal magnetization of different tissues. The reason for the contrast between the tissues is the differing decay time. White matter has a very short T_1 time and cerebrospinal fluid (CSF) has a long T_1 time. Gray matter has a intermediate T_1 time [14]. In those cases white matter would contribute to lighter pixels, grey matter to intermediate ones and CSF the darkest pixels as seen in figure 2.1.

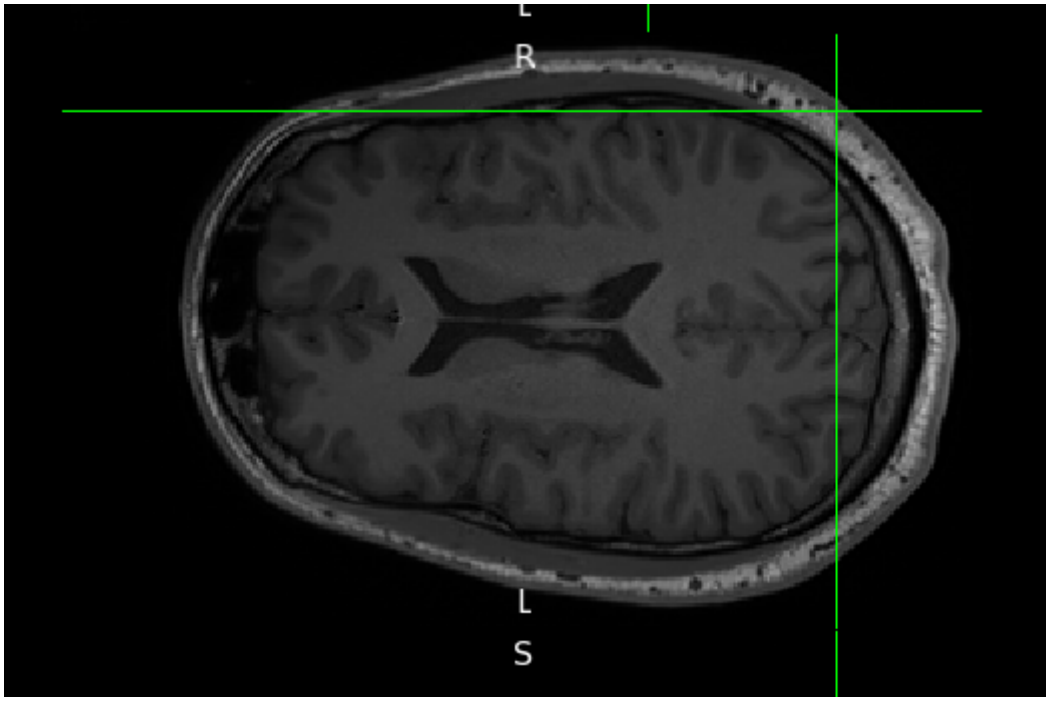

Figure 2.1: White matter is the lightest, gray matter intermediate and cerebrospinal fluid is the darkest parts

T_2 weighted image refers to an image constructed from the signals caused by the T_2 relaxation so by the dephasing of the nuclei and the decay of the transverse magnetization in different tissues. T_2 time differs by tissue creating the contrast. White matter has a short one and dephases rapidly, CSF has a long one and gray matter has an intermediate one [14]. In this case CSF is associated with lighter pixels, grey matter with intermediate and white matter with the darkest pixels as seen in figure 2.2.

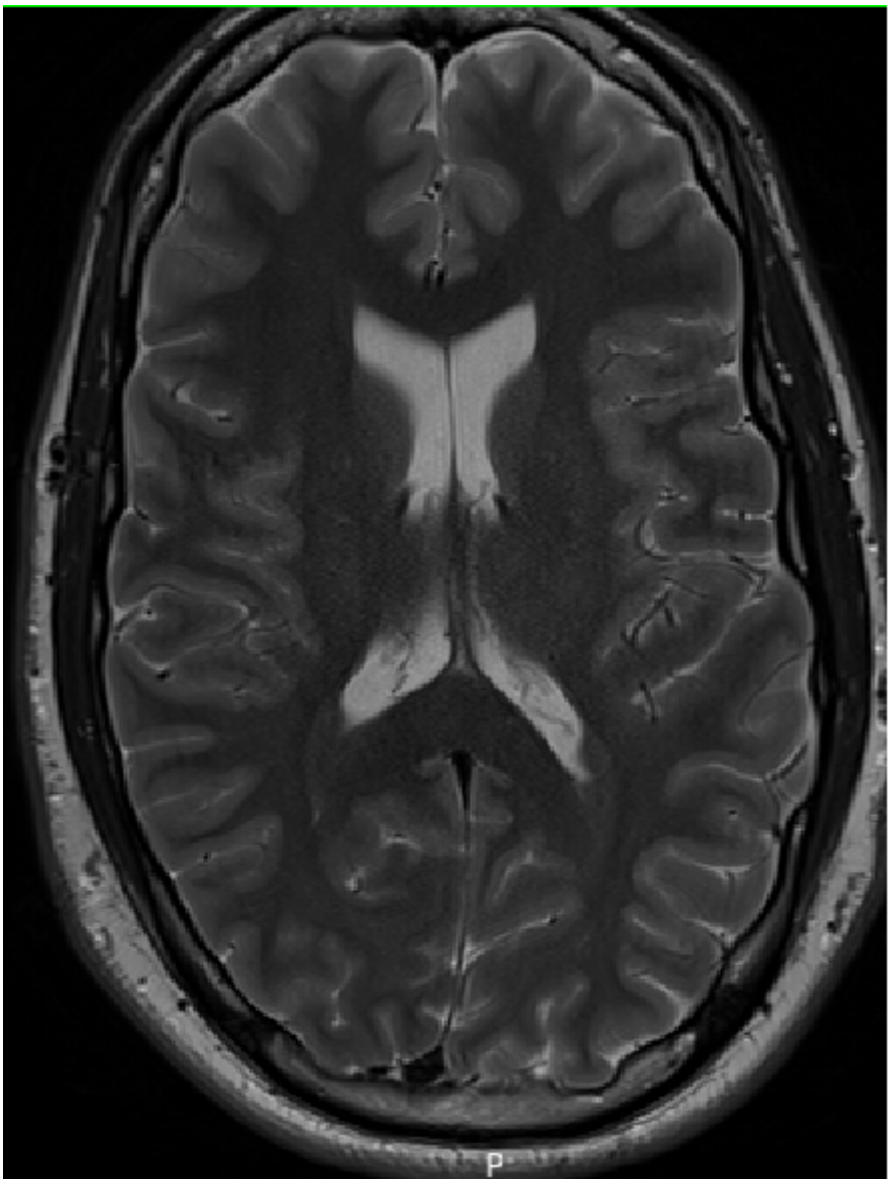

Figure 2.2: Cerebrospinal fluid is the lightest, grey matter intermediate and white matter the darkest parts.

2.2.3 Used Pulse sequences

Two pulse sequences were used in this work. 3D MPRAGE (3D magnetization prepared rapid gradient echo) to get the T_1 image and PRESS (Point Resolved Spectroscopy) for the spectroscopy imaging. PRESS utilizes spin echo and MPRAGE gradient echo.

Spin echo

The simplest sequence is the spin-echo sequence. It can produce T_1, T_2 or proton density weighted images [3].

It consists of a 90° radio frequency pulse which is called the excitation pulse. The

pulse flips the total magnetization into the x-y plane. Then after some time it is followed by a 180° degree pulse which is called inversion pulse.

At first after the excitation pulse the spins will become aligned creating a transverse magnetisation maximum and then start to dephase but then after the inversion pulse the spins will begin to realign again creating another maximum for the transverse magnetisation. The time between these two maximums is called the echo time or T_E. The inversion pulse comes at half the value of T_E. Another element is called T_R or repetition time which is the time between two different sequences.

A short T_E and a intermediate T_R would produce T_1 weighted image as the short T_E minimizes T_2 weighting and the intermediate T_R maximizes the T_1 weighting. For T_2 it would be long T_E (maximal T_2 weighting) and long T_R (minimal T_1 weighting). And for proton density weighted images a short T_E and a long T_R. The dependencies can be seen from the following formula for approximating the signal intensity of a voxel from [3]:

$$S_{SE} = PD * (1 - e^{-\frac{T_R}{T_1}}) * e^{-\frac{T_E}{T_2}}$$

where $1 - e^{-\frac{T_R}{T_1}}$ is the T_1 factor and the $e^{-\frac{T_E}{T_2}}$ is the T_2 factor. The PD is the proton density dependence which can't be varied.

Gradient recalled echo

Gradient recalled echo is an alternative for spin-echo sequence. In gradient recalled echo the initial flip angle is less than 90° [14]. This has some advantages over the spin-echo based sequences. It has increased signal at short T_R times and shorter T_E times as sometimes the inversion pulse is also omitted. It also has the disadvantage of image artifacts due to local field inhomogeneities and chemical shift effects which is caused by the omission of the inversion pulse.

Using the gradient fields it is possible to produce gradient echos which replace the echo caused by the inversion pulse.

Magnetization prepared rapid gradient echo (MP-RAGE)

MP-RAGE is a modification of T_1 fast field echo which is a gradient echo sequencing technique with a low-flip angle excitation. In this case the sequence is preceded by an inversion or saturation pulse to establish T_1 weighing creating a Turbo field echo or TFE sequence. MP-RAGE is the 3D version of TFE.

Point Resolved Spectroscopy (PRESS)

PRESS is a localization method that is a double spin-echo method. The sequence consists of three slice-selective RF-pulses; one excitation pulse and two refocusing pulses. The

pulses are applied in three different orthogonal gradients. The PRESS signal at echo time is the spin echo derived only from the volume that experienced all 3 pulses [10]. Meaning that the method localizes the signal to the cuboid-shaped volume created by the intersection of the three planes.

2.2.4 Chemical shift

The Larmor frequency is determined by the local magnetic field at the position of the nuclei and not by the external macroscopic magnetic field. The relation between the two magnetic fields is given by [3]:

$$\omega = \gamma B_{local} = \gamma(1 - \sigma)B_0$$

where σ is the dimensionless shielding constant that gives the relative resonance frequency shift and is independent of the magnitude of the magnetic field. The shift depends on the distribution of electrons around the nucleus and has different values for different molecules. This is the basis for magnetic resonance spectroscopy.

The chemical shift is determined as the relative frequency shift compared to a reference substance frequency ω_R. The difference $(\omega - \omega_R)$ is expressed as the dimensionless constant [3]:

$$\delta = \frac{\omega - \omega_R}{\omega_0} * 10^6 \cong (\sigma_R - \sigma) * 10^6$$

relative to the frequency $\omega_0 = \gamma B_0$ in part per million. For proton surrounded by 1 electron the chemical shift δ is around 10 ppm and for atoms with several electrons like ^{13}C nucleus it can be several hundreds of ppm [3]. The chemical shift provides information about the structure of the molecules and is a useful tool for the investigation of biochemical processes.

2.2.5 Difference between MRI and MRS

In MRI the main source of the signal is water and fat [14]. In brain MRS the source for the signal are metabolites. The metabolite concentrations are about 3-4 magnitudes smaller than that of water which also leads to poor temporal and spatial resolution. So the approach in MRI is quantitative and in MRS it is qualitative.

While MRI uses the bulk magnetic properties of the tissue that is being imaged, MRS uses the phenomenon of chemical shift to detect and quantify nuclei of different chemical groups based on the slightly different Larmor frequencies. In vivo MRS of the brain the signal from water and fat is suppressed and it is possible to gain information about the metabolites that are comparatively low in number.

In restricting the signal detection to a spectroscopy voxel volume, it is possible to remove the unwanted signals from outside the volume of interest (VOI) [10]. The

localization of the measurement decreases the effect of the variations of the magnetic fields as the variations are reduced over small volumes giving narrower spectral lines and more uniform signal excitation and reception.

2.2.6 Importance of volume fractions

Knowing the volume fractions of the spectroscopy voxel is important because of the low resolution of spectroscopy. The different brain tissues have different metabolic profiles. As the voxel size in spectroscopy is in order of cm and a typical voxel in brain images is of the order of mm it is important to know the different brain tissues in order to get meaningful information from spectroscopy standpoint.

In addition the volume fraction of cerebrospinal fluid can be used as a diagnostic tool. Pathological brain states often include brain tissue atrophy which causes the volume fraction of CSF to increase in cortical regions. The amount of CSF increases also with age [18].

2.3 Segmentation of images

Segmentation of images consists of a few preprocessing steps and then the image segmentation.

2.3.1 Preprocessing

Preprocessing of the image acquired is required for the segmentation of the images. The most important preprosessing steps are: MRI bias field correction, removal of nonbrain tissue and image registration in the case of multimodal image analysis.

MRI bias field correction

The automatic segmentation of MR scans can be hindered by acquisition-related artifacts. One of which is the lack of homogeneity of the radiofrequency which is known as the bias field. It consists of multiplicative variation of intensity levels across the image. The artifact is mostly unnoticeable to the naked eye but can degrade the volumetric quantification of cerebral tissue. Which is why bias field correction algorithms are used to reduce the classification error rates of the segmented tissue types.

The artifact can be caused by many different things. For example lack of uniform sensitivity of the radiofrequency emitting and receiving coils, the static field inhomogeneities or the magnetic susceptibility of the tissue [6]. As such the strength of the bias field is dependent on the strength of the magnetic field.

Due to the fact that it is difficult to correct for the bias field while taking the image, the correction of the bias field is done by post-acquisition mathematical algorithms. The algorithms can be divided into the ones that use segmentation and the ones that don't. The ones that use segmentation involve the following steps: First they classify the image voxels into a set of predefined tissue classes. Then calculate the bias-free image by assigning voxels in each tissue class their mean tissue intensity. After that they calculate the residual image as the substraction of the bias free image from the original image. Then they estimate the bias field by smoothing the residual image and finally it corrects the original image using this estimated bias field. This process is iterated until no desired improvement is gained.

Removal of non-brain tissue

As the nonbrain tissues such as fat, skull or neck have intensities that overlap with the intensities of brain tissues it is important to remove the nonbrain tissue before image segmentation. The point is to classify the voxel into either brain or nonbrain voxels. The result can be a binary mask or a new image with just the voxels that were classified as brain.

A very used method for brain extraction is the brain extraction tool (BET) [19] which was also used in this work.

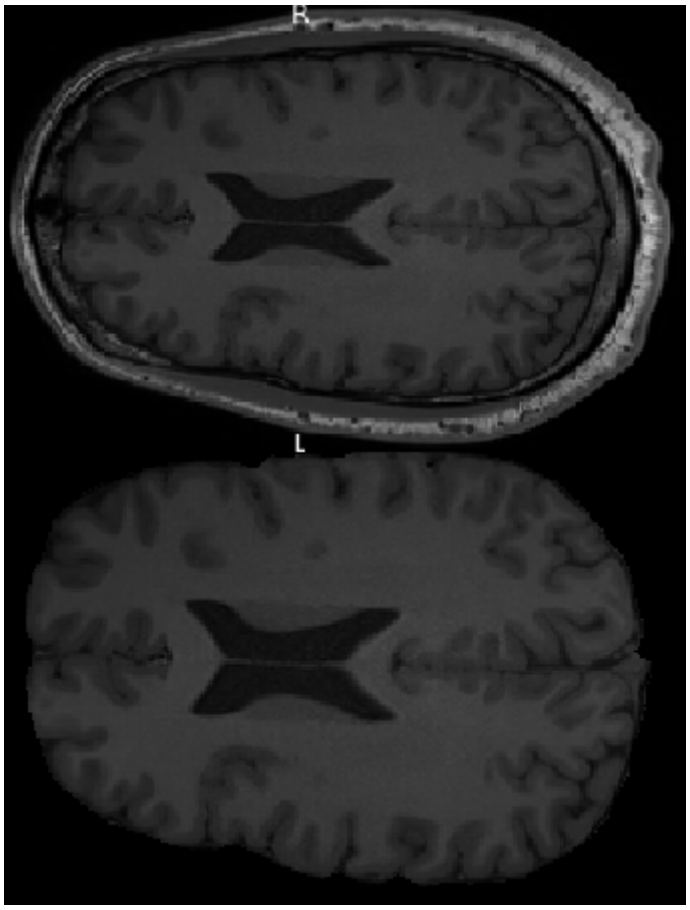

Figure 2.3: The image with the skull remaining and the image of the extracted brain using BET.

Image registration

Image registration is the process of overlaying two or more images of the same scene taken at different times from different viewpoints and/or by different sensors. Image registration is useful in all image analysis tasks which require for the desired information to be gained from the combination of various data sources. In medicine it is used for example to combine computer tomography and MR data to obtain more information about the patient, monitoring tumour growth and comparing patient's data with anatomical atlases [22].

2.3.2 Image segmentation methods

There are several different segmentation methods that have been developed over the years[4]. One way to divide the methods is in to the following 6 groups:

- Manual segmentation
- Intensity-based methods

- Atlas-based methods
- Surface-based methods
- Hybrid segmentation methods
- Convolutional neural network (CNN) segmentation methods

Manual segmentation

Manual segmentation is the process of labeling image pixels and segmenting the image by human operator by hand. Manual segmentation methods are usually the most trusted methods but are very prone to human error. Even the same human operator has trouble reproducing the segmentation in a reliable and objective manner[4].

The biggest problem with manual segmentation is how it is generally time consuming and challenging. When it comes to large sets of images automated augmentation methods are required. Manual segmentation methods are still used to establish ground truths which are used to evaluate new segmentation methods and in creating brain atlases.

Intensity-based methods

Intensity-based segmentation methods rely on the different intensities of the pixels. Different tissue classes have different pixel/voxel intensities depending on the chosen weighing. Intensity-based methods classify the different tissues into their own classes based on the intensity of the pixels/voxels[4].

The methods are very susceptible to error due to artifacts and overlapping intensities. The most simple method for classifying pixels/voxels based on their intensity is to use a intensity histogram and the fit a Gaussian function to its distribution. The probability function is the used to classify the probabilities that a voxel/pixel is of a certain tissue type. By incorporating neighbour information, meaning the information about the surrounding other pixels, the methods will give preference to homogeneous regions which can reduce the miss-classification due to random noise. Ready made atlases can be used to increase the probability of success by giving information about what kind of tissues are to be expected in a certain areas of the brain.

Atlas-based methods

Atlas-based methods can be a very useful tool if there exists a specific atlas or a template for the human brain of the population of interest is available. Atlases contain anatomical information about the human brain which can be overlaid on top of the image gained. Meaning that registration methods have to be used and thus the quality of the atlas-based

methods is directly dependant on the quality of the registration method[4]. Using such methods it is possible to segment the brain without any additional costs.

As the brain atlases are based on the brains of healthy individuals the method can give inaccurate results or fail when used with patients with brain lesions or a brain anatomy that differs from the atlas.

Surface-based methods

Surface-based methods use the surface of the brain for segmentation. There are many surface-based methods such as deformable models. Deformable models are a fusion of geometry, physics and approximation theory [8]. The models use closed parametric curves or surfaces to define a region of the brain and the curves/surfaces deform under the influence of forces. Geometry represents the shape, physics define the constraints of the varying of the shape and approximation theory fits the models to data. The brain extraction tool (BET) uses such a deformable model.

Hybrid segmentation methods

Hybrid segmentation methods are a combination of different methods. There are many situation where the segmentation method that should be used is not clear. So a combination of several different complementary techniques may be required to reach the desired outcome. The point would be to use methods that complement each others weak points to increase the segmentation accuracy. The drawback for such methods is the increased complexity compared to single method ones. Hybrid segmentation methods are often used in tandem with neural networks.

Convolutional neural network segmentation methods

With the advent of neural networks being commonly used in detecting what objects are in a image. The interest in designing neural network based methods for for medical image segmentation has increased.

A convolutional neural network (CNN) is a type of artificial neural network that is used in image recognition and processing that is optimized for pixel data. CNNs are machine learning pipelines modeled after the biological process of neurons and synapses. CNNs are the foundation for AI image segmentation.

Deep learning CNNs have already been used for the difficult job of segmenting neonatal brains [5] as the brains tend to vary more compared to adult brains. The neural networks were trained using over 24 pairs of neonatal T_1 and T_2 data set and validated against another eight pairs against ground truth. The research shows that both neural networks could segment the brain but not with 100% accuracy.

CNNs show promise in segmenting brain tumours [1]. Machine learning is a fast developing field and it will have a continuous impact on image segmentation method development in the future.

2.4 Medical image file formats

Image file formats are a central part of medical imagining.

There are some different formats used when processing medical images. Some of the most used ones are: Analyze, Neuroimagining informatics Technology Initiative or Nifti, Minc and Digital Imagining and communications in Medicine or Dicom [11]. This work will focus on Dicom and Nifti file formats as they are the most used starting with Nifti. This work uses a rda file format. Which is simply Siemens Magnetic Resonance Spectroscopy RDA file and is the scanners own file format replacing in this case Dicom.

Medical file formats can be divided in two different catagories [11]. The first category is intended to standardize the images generated by diagnostic modalities for example the Dicom format. The second category has the aim to facilitate and strengthen post-prosessing analysis for example the Nifti format.

Medical image files are usually divided into a header and the image file. In the header there is meta information about the file. In some formats the metadata and the image are stored in the same file and in others they are two seperate files.

2.4.1 Neuroimagining informatics Technology Initiative (Nifti)

Nifti is a file format created as a successor to Analyze format[11]. The intent was to create a format that keeps Anylyzes strengths but gets rid of the weaknesses. While analyze used the two-files paradigm, meaning that there were two seperate files for metadata and image (.hdr and .img), Nifti usually uses a single .nii file to store both of them. Although the format still allows separate file storage.

Nifti as well as Analyze is designed for multidimensional data. Meaning that it is possible to store in one file 3D or 4D data. In Nifti the first three dimensions are the three spatial dimensions and the fourth dimension is used for the temporal dimension.

The most major problem in Analyze was the lack of proper information about orientation in space. Leading to the result that the stored data could not be unambiguously interpreted. Nifti allows two different methods for storing orientation. First one comprises of a rotation and a translation and is used to map voxel coordinates to the scanner frame of reference. The second method is used to save the 12 parameters of a general linear transformation and aligning the image volume to a standard or template-based coordinate system.

The Nifti format has replaced Analyze in neuroimaging research. It is adopted as the default format by some of the most used software packages and segmentation programs such as FSL which is used in this work.

2.4.2 Digital Imagining and communications in Medicine (Dicom)

Dicom is a very important part of medical imaging. It is not only a image format but also a network communication protocol [11]. This work focuses on the image format part.

Dicom files contain in its header the personal information of the patient such as name, gender, age, weight and height and the type of procedure that led to the formation of the image itself. Dicom stresses the part that when the image is seperate from metadata the image becomes meaningless as a medical image. Dicom uses the one file system as in the header and the image are both in a single file. The header contains the description of the entire procedure in terms of acquisition protocol and scanning parameters.

Dicom works with 2-dimensional slices that can be reconstructed into a full 3D image.

2.5 FSL a library of analysis tools

The software tool that was created for this work heavily relies on FSL [9]. FSL is a comprehensive library of analysis tools for dealing with MRI brain imagining data. The software tool uses many features of FSL including fslmaths for creating a VOI mask, makerot for creating rotational matrices, convert_xfm for combining rotational matrices, flirt for rotating the created mask to the right orientation using the rotational matrices, robustfov for the removal of the neck, BET is used for brain extraction, FAST for brain segmentation and fslstats for determining the volume fractions inside the VOI. This section will go over in more detail the extraction method used in BET and the segmentation method used in FAST.

2.5.1 Brain extraction tool (BET)

The brain extraction tool (BET) [19] uses a deformable model which evolves to fit the brains surface using locally adaptive model forces.

BET starts by finding the centre-of-gravity of the head and the rough size of the head in the image. Then it forms a sphere with the spheres surface being triangular tessellation which is allowed to deform one vertex at a time while attempting to move towards the brain's edge. If a clean solution is not found the process is re-run with higher

smoothness constraints. BET can have trouble with neck pixels so removing residual neck tissue before extraction is recommended [15]. An example of an extracted brain using BET is found in figure 2.3.

2.5.2 FMRIB's Automated Segmentation Tool (FAST)

FMRIB's Automated Segmentation Tool or FAST [21] is a segmentation tool included in the FSL package. FAST does the bias field correction while segmenting.

The segmentation method used in FAST is based on a Hidden Markov Random Field model or HMRF model that is optimized using the Expectation Maximization algorithm. The model takes into account the intensity of the voxel and the spatial neighbourhood information (from the Markov Random Field) to determine the voxels tissue type. The final probability of a MRF and a Gaussian based model is then dependent on the intensity and the chosen weight of the neighbourhood information. If the weight of the neighbourhood information is too large the segmentation algorithm will start preferring for everything to be the same tissue which can also lead to a false result. The balance between the two contributions is important which is done by the Expectation Maximization algorithm.

FAST begins with a rough initial segmentation and the starts to iterate the estimation of the bias field and the estimation of the segmentation. As the segmentation is dependant on the value of the bias field and the calculation of the bias field is dependent on the intensity of the voxels this iteration is necessary. Finally FAST applies a partial volume system model with its own separate MRF to determine for each voxel to what fraction of the different tissues each voxel is and creates as an output the partial volume estimates as three different files inputfilename_pve_x with x=0 for CSF, x=1 for gray matter and x=2 for white matter.

3. Conducted imaging

The imagining was conducted at Meilahti Kolmiosairaala at 23.9.2021 and the subject scanned was the author. The measurements were done using a 3 Tesla clinical imager (model: Skyra, manufacturer: Siemens, Erlangen, Germany) with 64 channel head/neck coil. The image used in this work is a T_1 weighted image and in this section will go through the image and the data gained for analysis.

3.1 Imagining

The T_1 scan in question used MP-RAGE sequence which is a magnetization prepared rapid gradient echo. The parameters of MP-RAGE sequence were (T_R/T_E=3.15/1.37 ms, flip angle=8°, image voxel size =0.8x0.8x0.8 mm^3, acquisition matrix = [0,160,160,0], field of view = 260 x 260 mm, acquisition time 2 minutes 22.6 seconds).

3.2 Volumes of interest (VOI)

In order to use and test the software tool 5 volumes of interest were chosen. The volumes are as follows: a 20x20x20 mm volume from the frontal cortex, a 30x15x15 mm volume from the anterior cingulate cortex, two 20x20x20 mm volumes of areas bit off of centrum semiovale so that the volume is mostly gray matter one on the right side one on the left and lastly a 20x30x20 mm volume from the frontal-right side of the brain. The volumes of interest and their orientations can be seen in figure 3.1

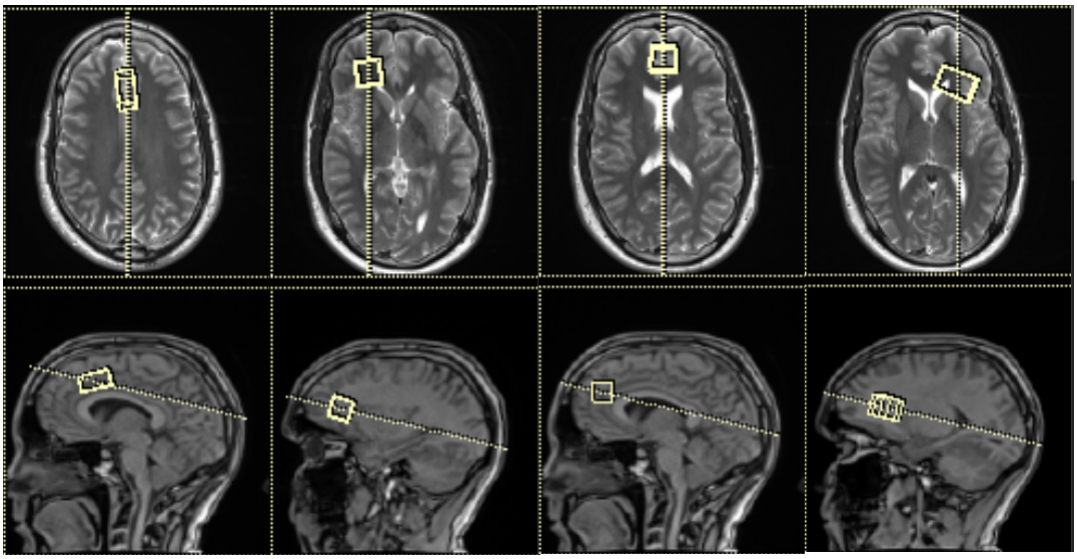

Figure 3.1: From left to right: anterior cingulate cortex, one of the centrum semiovale, frontal cortex, and the last one is a volume from the frontal-right side of the brain

3.3 Segmented images

The segmentation was done using BET and FAST. The result is shown in figure 3.2.

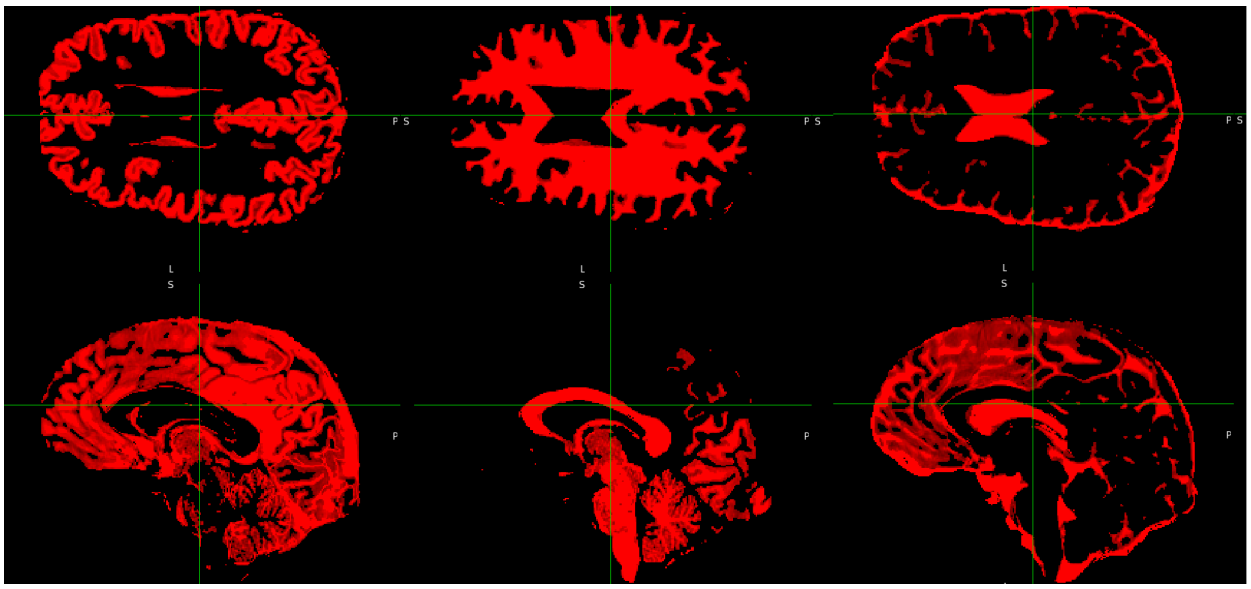

Figure 3.2: From left to right: gray matter mask, white matter mask and the cerobrospinal fluid mask

4. Methods

In order to determine the volume fractions a software tool was created utilizing existing tools and methods and with the help of [16]. The software tool was mostly written in python and heavily uses tools provided by FSL. It starts by asking the input .rda file and then .nii file. After that it uses the FSL function robustfov to remove most of the neck from the image and sends a command to utilize a JavaScript script to read the input .rda file.

4.1 Reading the input .rda file

So the first step is to read the relevant information from the input file which in this case was a .rda file. Python has trouble reading this file type but JavaScript was able to read the file. Because of this the software tool utilizes JavaScript in order to read the input .rda file. This requires for the user to have node installed which can be easily done by using the terminal command "pip install node". The JavaScript code takes the parameters from the files and creates .txt files that are then opened in the main python code.

The relevant information from the file for the creation of the software tool are found in table 4.1:

Table 4.1: List of parameters and their descriptions

Parameters and the descriptions			
RowVector[0]	RowVector[1]	RowVector[2]	the row vector
ColumnVector[0]	ColumnVector[1]	ColumnVector[2]	the column vector
VOIPositionSag	VOIPositionCor	VOIPositionTra	SVS voxel translations
VOINormalSag	VOINormalCor	VOINormalTra	normal vector
VOIThickness	VOIPhaseFOV	VOIReadoutFOV	voxel dimensions
VOIRotationInPlane			rotation in plane in radians

4.2 Calculating the coordinates of the VOI

In order to get the coordinates of the VOI all the parameters found in table 4.1 are needed with the exception of VOIRotationInPlane. The software tool reads the files provided by the JavaScript script and creates arrays out of them. Then it uses the parameters to calculate coordinates for the corners of the spectroscopy voxel. The method for calculating the coordinates of the corners is the same as in a Matlab script created by Dr. Michael Lindner [13].

Then it calculates 8 different arrays with 4 data elements from the parameters. The first three array elements are of the form: VOIThickness/2*RowVector[i]+VOIReadoutFOV/2*ColumnVector[i]+VOIPhaseFOV/2*NormalVector[i] with i=0,1,2 matching the index and the fourth element being 1. The arrays are all of this form but with differing signs in front of row,column and normal vector components with all 8 different combinations.

After creating these arrays the software tool combines them into one 8x4 matrix. Then the software tool takes its transpose and creates a matrix called oc1T. Then it forms a matrix from the voxel positions.

$$V = \begin{bmatrix} 1 & 0 & 0 & VOIPositionSag \\ 0 & 1 & 0 & VOIPositionCor \\ 0 & 0 & 1 & VOIPositionTra \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

It creates a new matrix by multiplying oc1T and V together;

$$f = V * oc1T$$

. And takes its transpose and calls it fT.

The the first three elements of each row of the created matrix consists of the coordinates of the VOI corners as defined around the origin of the coordinate system but with the opposite sign ie.

$$corner_i = \begin{bmatrix} -fT[i,0] & -fT[i,1] & -fT[i,2] \end{bmatrix}$$

with i going from 0 to 7 to get all the 8 corners.

The midpoint of the VOI can be calculated by choosing a corner and subtracting the coordinates from all the other corners and seeing where the difference is the largest which will be the opposite corner. Once the opposite corner is found the midpoint coordinates in anatomical coordinates is calculated by taking the average of the coordinates of the two corners.

Following [16] in order to get the voxel coordinates the anatomical coordinates must be transformed using a transformation matrix $(M_{MRI})^{-1}$ which is the inverse anatomical image transformation matrix. This can be obtained from the .nii file that was taken as the other input. The matrix is gained using the python library nibabel[2] which can be installed using pip install nibabel. Using the command nibabel.load(.nii file) and setting a parameter img=nibabel.load(.nii file). The nii-file header information can be gained with header=img.header. The M_{MRI} matrix is gained with the command M_{MRI}=header.get_sform().

The voxel coordinates of the midpoint of the VOI are calculated by multiplying the anatomical midpoint coordinates with the inverse of M_{MRI}:

$$MpointCoord_{voxel} = (M_{MRI})^{-1} * MpointCoord_{anatomical}$$

4.3 Creating a mask

To create a mask in these coordinates the software tool uses the FSL util fslmaths to first create a spot at the midpoint coordinates and then using the fslmath -kernel box (in the case of a cube) and fslmaths -kernel boxv3 (in the case if the sides are of differing length) to create a mask. The kernel box command takes the dimensions in mm which are direct values of the voxel dimensions from the .rda file. The kernel boxv3 in takes the values in voxels. Therefore to get the correct values the dimension must be divided with the pixel size which can be gained with from the nii-file header with the command header.get_zooms() using nibabel. This means that if the result of the sides is even the mask will be off by one pixel as the mask has to be created from a centre point and expanded out evenly in both directions for all sides.

To manually rotate the mask to correct orientation the software tool uses makerot command and sets the centre of rotation at the midpoint coordinates. as the coordinates must be in mm the midpoint coordinates are multiplied by the pixel dimension. Makerot requires the rotation angles with regard to the normal vector. The software tool uses the normal vectors (1,0,0) ,(0,1,0) and (0,0,1) separately creating 3 different rotation matrices and then combining them using the convert_xfm command. In order to get the rotation angles in relation to the normal vectors in question the software tool needs the following parameters from the .rda file: VOIRotationInPlane, VOINormalSag and VOINormalTra. The rotation angle in relation to the normal vector (1,0,0) in degrees is $\theta = arcsin(VOINormalSag) * \frac{180}{\pi}$, for the normal vector (0,1,0) it is $\theta = VOIRotationInPlane * \frac{180}{\pi}$ and for the normal vector (0,0,1) it is $\theta = arcsin(VOINormalCor) * \frac{180}{\pi}$. Then the software tool uses the flirt function in FSL to get the rotated version of the mask with the input being the non-rotated version and

the final rotational matrix.

4.4 Brain extraction and segmentation and the volume fractions

The commands to qualify the tissue classes within the SVS voxel are the same as they are in [16] "Example commands 1" except for the brain extraction which has had its parameters changed.

In the beginning the neck is removed with the robustfov command. This creates a new file to be extracted with BET. The parameters where changed according to the suggestions of [15]. The new command is "bet file.nii outputfile.nii.gz -B -f 0.1 -g 0". -B is the command for removing residual neck voxels as those effect the efficiency of BET.

Next FAST is used to segment the brain with the command "fast -t 1 -n 3 -H 0.1 -I 4 -l 20.0 -o outputfile.nii.gz". With outputfile.nii being the outputfile from the brain extraction. This creates the outputfile_pve_x files (x=0 for CSF, x=1 for gray matter, x=2 for white matter).

The last step is to use fslstats to get the volume fractions. The command is used for each pve_x file. The command is "fslstats -t outputfile_pve_x.nii.gz -k RotatedMask.nii.gz -m".

In the end the software tool prints out the volume fractions of each type.

4.5 The use of the developed software tool

The use of the software tool requires the installation of FSL, node, nibabel and numpy.

When the software tool is run it will ask for the input .rda file. For example if the file name is file.rda the input is "file.rda". Then it will ask to give the input .nii file and the input is given a file name file.nii, "file.nii".

The software tool will run all the steps mentioned above with the brain extraction and segmentation taking up most of the run time and give as the output the volume fractions.

5. Using software tool for the acquired data

In this chapter the software tool is used on the T_1 image and the volumes of interest discussed in chapter 3.

5.1 Checking the location and rotations of the VOI

The orientation and the location of the VOI can be verified by comparing the mask created with the software tool with the one reconstructed from the accompanying DICOM files and that are seen in fig 3.1.

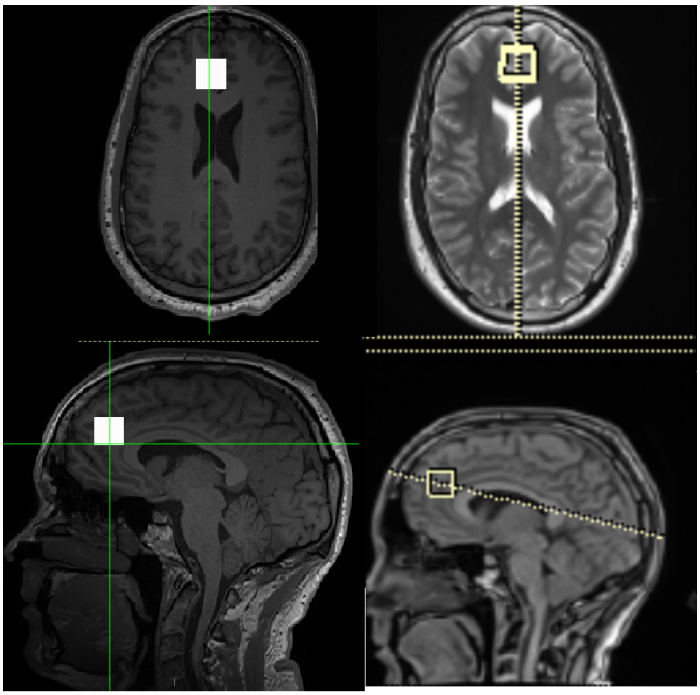

Figure 5.1: Frontal cortex VOI. On the left are the masks created by the software tool and on the right the original ones.

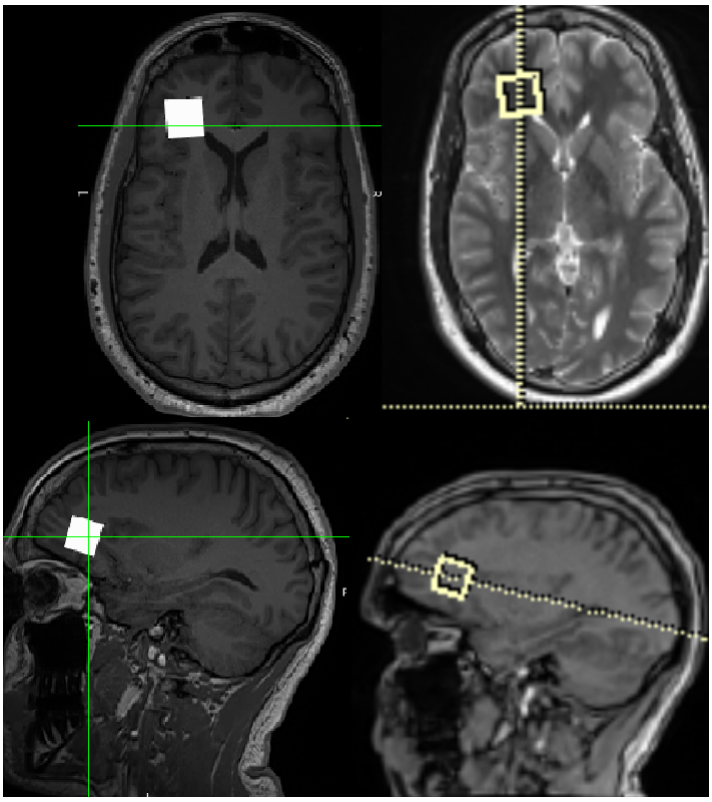

Figure 5.2: One of the centrum semiovale VOIs. On the left are the masks created by the software tool and on the right the original ones.

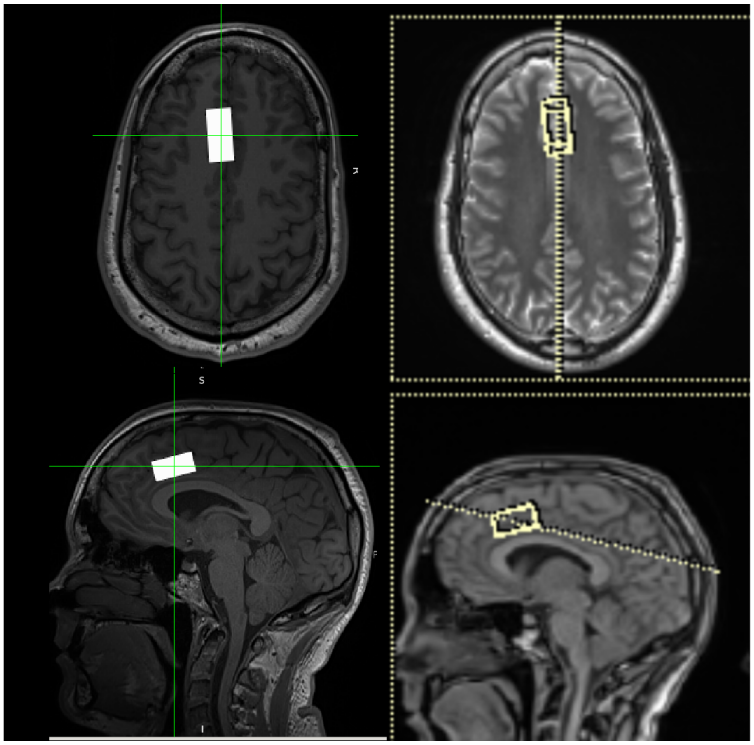

Figure 5.3: Anterior cingulate cortex VOI. On the left are the masks created by the software tool and on the right the original ones.

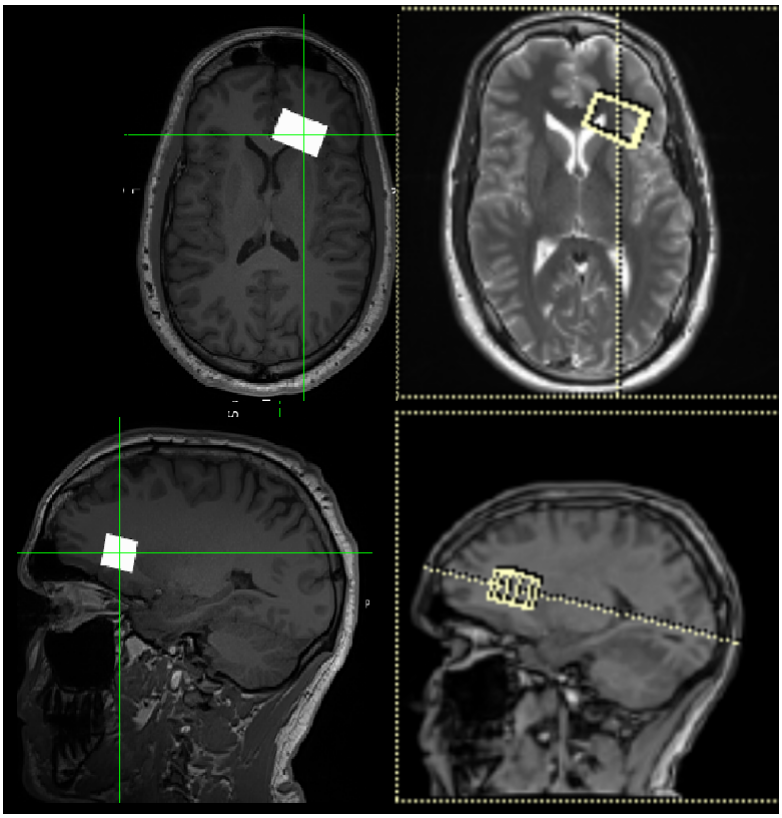

Figure 5.4: Volume of interest from the frontal-right side of the brain. On the left are the masks created by the software tool and on the right the original ones.

Comparing the figures the masks location and orientation seem to be the same.

5.2 The volume fractions for each VOI

The output volume fractions of the software tool for each of the .rda files created for testing is seen in the 5.1:

Table 5.1: The volume fractions of different volumes of interest gained from using the software tool

Different values for the volume fractions in specific VOI			
VOI in question	Cerebospinal fluid	Gray matter	White matter
Frontal cortex VOI	0.137	0.572	0.291
Anterior cingulate cortex VOI	0.151	0.653	0.196
Centrum semiovale (left) VOI	0.0155	0.198	0.787
Centrum semiovale (right) VOI	0.0178	0.233	0.749
VOI Frontal-right brain	0.0700	0.222	0.708

6. Discussion

MRI and MRS are integral part of brain imagining. MRI has proved to be a powerful diagnostic tool and even though MRS is quite new it has shown potential in diagnostics and learning about different diseases in the brain with new methods. For example functional MRS or fMRS can be used to track the concentration of neurotransmitter glutamate and γ-aminobutyric acid or GABA. Glutamate is the dominant neurotransmitter for most of cortical and hippocampal excitatory neurons and GABA is the neurotransmitter for the inhibitory neurons [20]. This could possibly be used for learning about psychiatric disorders and age-related cognitive dysfunctions.

Brain imagining and segmentation as a whole is a wide field of research and the methods discussed in this work represent only a small portion of the available tools. There are multiple different softwares/libraries for brain segmentation such as FreeSurfer or the matlab based spm. At least for spm there are similar software tools created. The software tool created in this work should be simple to use with FSL as it only takes as the input the .rda-file and a nifti-file which seems to make the created software tool easier to use than the other software tools. Medical files have been designed by many people for different uses and as such they don't always work ideally in all applications. Which lead to a strange difficulty from trying to read the medical files with python for which the solution was a combination of programming languages. JavaScript is a versatile language so it could be useful for other applications as well.

There are many possible sources of error for the results. For instance the movement of the subject[7], the bias field correction could be flawed, the segmentation method used could cause error in estimating the intensities or over or undervaluing neighbourhood information, the placement of the mask is slightly off and the removal of the nonbrain tissue could be incomplete. All of which can effect the results for the volume fractions. However, the results that were gained seem to realistically represent the areas of the brain the VOI were in. With the inner parts of the brain being mostly white matter and the more outer parts being more gray matter. In the cortical regions the percentage of gray matter was around 60%. The volume fractions will differ depending on the scanned brain due to for example age or individual differences which increases the importance of the created software tool.

The software tool could be expanded to include Dicom support. The reading of the relevant information on the Dicom file lead to the unexpected result of the information differing from the .rda file and the masks that were created based on the information were in slightly different place. If the software tool is to be expanded this needs to be solved. The software tool could be used in research with the .rda file input but for clinical use Dicom support is necessary.

7. Conclusion

In this work a software tool was created for determining the partial volumes of brain tissue from spectroscopy voxels.

The theory section briefly went over the theory of nuclear magnetic resonance and its applications MRI and MRS along with the relevant pulse sequences and short section on the differences between MRI and MRS. Then continued with the basics of image segmentation going through the preprocessing steps and the basic segmentation techniques along with the growing field of CNN segmentation methods. Finishing the theory with a short introduction to medical image file formats and FSL and its basic tools.

The conducted imagining done at Meilahti Kolmiosairaala was detailed. After which 4 different volumes of interest where discussed and the imaged T_1 weighted image was segmented.

In the methods section the creation of the software tool was detailed and the code is attached as an appendix.

Lastly before a short discussion the software tool was used on the image data in order to test the functionality of the software tool and the results were presented. The results where as one would expect from the regions in question. There is no set value for the fractions as they differ based on the subject but for example the percentage of gray matter in the cortical regions was 60% which corresponds to the expected fractions of the regions. In conclusion the software tool gives results of what is expected of the regions in question.

Determining the partial volumes of brain tissues is essential for spectroscopy data processing and interpreting the data. Given that the partial volume fractions differ between subjects it is important to determine the fractions in each case separately. The created software tool will help and quicken the process. The software tool could be used in research that concerns the subject matter.

Bibliography

[1] Abhishta Bhandari, Jarrad Koppen, and Marc Agzarian. "Convolutional neural networks for brain tumour segmentation". In: 11.1 (June 2020). DOI: 10.1186/s13244-020-00869-4. URL: https://doi.org/10.1186/s13244-020-00869-4.

[2] Matthew Brett et al. *nipy/nibabel: 3.2.1*. Version 3.2.1. Nov. 2020. DOI: 10.5281/zenodo.4295521. URL: https://doi.org/10.5281/zenodo.4295521.

[3] Gunnar Brix et al. "Basics of Magnetic Resonance Imaging and Magnetic Resonance Spectroscopy". In: *Magnetic Resonance Tomography*. Ed. by Maximilian F. Reiser, Wolfhard Semmler, and Hedvig Hricak. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008, pp. 3–167. ISBN: 978-3-540-29355-2. DOI: 10.1007/978-3-540-29355-2_2. URL: https://doi.org/10.1007/978-3-540-29355-2_2.

[4] Ivana Despotović, Bart Goossens, and Wilfried Philips. "MRI Segmentation of the Human Brain: Challenges, Methods, and Applications". In: *Computational and Mathematical Methods in Medicine* 2015 (2015), pp. 1–23. DOI: 10.1155/2015/450341. URL: https://doi.org/10.1155/2015/450341.

[5] Yang Ding et al. "Using Deep Convolutional Neural Networks for Neonatal Brain Image Segmentation". In: *Frontiers in Neuroscience* 14 (2020), p. 207. ISSN: 1662-453X. DOI: 10.3389/fnins.2020.00207. URL: https://www.frontiersin.org/article/10.3389/fnins.2020.00207.

[6] Juan D. Gispert et al. "Method for bias field correction of brain T1-weighted magnetic resonance images minimizing segmentation error". In: 22.2 (Mar. 2004), pp. 133–144. DOI: 10.1002/hbm.20013. URL: https://doi.org/10.1002/hbm.20013.

[7] Inger Havsteen et al. "Are Movement Artifacts in Magnetic Resonance Imaging a Real Problem?—A Narrative Review". In: *Frontiers in Neurology* 8 (May 2017). DOI: 10.3389/fneur.2017.00232. URL: https://doi.org/10.3389/fneur.2017.00232.

[8] Ravindra Hegadi, Kop Arpana, and Mallikarjun Hangarge. "A Survey on Deformable Model and its Applications to Medical Imaging". In: *International Journal of Computer Applications* 2 (Jan. 2010).

[9] Mark Jenkinson et al. "FSL". In: *NeuroImage* 62.2 (2012). 20 YEARS OF fMRI, pp. 782–790. ISSN: 1053-8119. DOI: https://doi.org/10.1016/j.neuroimage.2011.09.015. URL: https://www.sciencedirect.com/science/article/pii/S1053811911010603.

[10] Stephen F Keevil. "Spatial localization in nuclear magnetic resonance spectroscopy". In: *Physics in Medicine and Biology* 51.16 (July 2006), R579–R636. DOI: 10.1088/0031-9155/51/16/r01. URL: https://doi.org/10.1088/0031-9155/51/16/r01.

[11] Michele Larobina and Loredana Murino. "Medical Image File Formats". In: 27.2 (Dec. 2013), pp. 200–206. DOI: 10.1007/s10278-013-9657-9. URL: https://doi.org/10.1007/s10278-013-9657-9.

[12] P. C. LAUTERBUR. "Image Formation by Induced Local Interactions: Examples Employing Nuclear Magnetic Resonance". In: *Nature* 242.5394 (Mar. 1973), pp. 190–191. DOI: 10.1038/242190a0. URL: https://doi.org/10.1038/242190a0.

[13] Lindner Michael. *MRSParVolCo*. Jan. 2019. URL: https://github.com/DrMichaelLindner/MRSParVolCo (visited on 12/16/2021).

[14] Robert A. Pooley. "Fundamental Physics of MR Imaging". In: 25.4 (July 2005), pp. 1087–1099. DOI: 10.1148/rg.254055027. URL: https://doi.org/10.1148/rg.254055027.

[15] V. Popescu et al. "Optimizing parameter choice for FSL-Brain Extraction Tool (BET) on 3D T1 images in multiple sclerosis". In: 61.4 (July 2012), pp. 1484–1494. DOI: 10.1016/j.neuroimage.2012.03.074. URL: https://doi.org/10.1016/j.neuroimage.2012.03.074.

[16] Scott Quadrelli, Carolyn Mountford, and Saadallah Ramadan. "Hitchhiker'S Guide to Voxel Segmentation for Partial Volume Correction of in Vivo Magnetic Resonance Spectroscopy". In: 9 (Jan. 2016), MRI.S32903. DOI: 10.4137/mri.s32903. URL: https://doi.org/10.4137/mri.s32903.

[17] I. I. Rabi et al. "A New Method of Measuring Nuclear Magnetic Moment". In: *Phys. Rev.* 53 (4 Feb. 1938), pp. 318–318. DOI: 10.1103/PhysRev.53.318. URL: https://link.aps.org/doi/10.1103/PhysRev.53.318.

[18] L. Sakka, G. Coll, and J. Chazal. "Anatomy and physiology of cerebrospinal fluid". In: *European Annals of Otorhinolaryngology, Head and Neck Diseases* 128.6 (Dec. 2011), pp. 309–316. DOI: 10.1016/j.anorl.2011.03.002. URL: https://doi.org/10.1016/j.anorl.2011.03.002.

[19] Stephen M. Smith. “Fast robust automated brain extraction”. In: 17.3 (Nov. 2002), pp. 143–155. DOI: 10.1002/hbm.10062. URL: https://doi.org/10.1002/hbm.10062.

[20] Jeffrey A. Stanley and Naftali Raz. “Functional Magnetic Resonance Spectroscopy: The “New” MRS for Cognitive Neuroscience and Psychiatry Research”. In: 9 (Mar. 2018). DOI: 10.3389/fpsyt.2018.00076. URL: https://doi.org/10.3389/fpsyt.2018.00076.

[21] Y. Zhang, M. Brady, and S. Smith. “Segmentation of brain MR images through a hidden Markov random field model and the expectation-maximization algorithm”. In: *IEEE Transactions on Medical Imaging* 20.1 (2001), pp. 45–57. DOI: 10.1109/42.906424.

[22] Barbara Zitová and Jan Flusser. “Image registration methods: a survey”. In: 21.11 (Oct. 2003), pp. 977–1000. DOI: 10.1016/s0262-8856(03)00137-9. URL: https://doi.org/10.1016/s0262-8856(03)00137-9.

A. Python code

```
import os
import nilearn
import numpy as np
from numpy import linalg
import nibabel as nib
import scipy.io as sio



def SignPermutations(x,y,z):
    c=[x*(VoiT/2)*(row[0]) + y*(VoiR/2)*(col[0]) + z*(VoiP/2)*(normal[0]),
    x*(VoiT/2)*(row[1]) + y*(VoiR/2)*(col[1]) + z*(VoiP/2)*(normal[1]),
    x*(VoiT/2)*(row[2]) + y*(VoiR/2)*(col[2]) + z*(VoiP/2)*(normal[2]),1]
    return c


file = input("Enter rda file name : ")
T1=input("Enter nii file name : ")
os.system('node work.js ' + file)
os.system('robustfov -i '+T1+' -r T1neckremoved.nii')
T1='T1neckremoved.nii.gz'
row=[]
col=[]
SVST=[]
VoxelDim=[]
with open('Voxel Dimensions.txt') as dims:
    lines=dims.readline()
dimss=lines.split(",")
for line in dimss:
    VoxelDim.append(((float(line))))
```

```
with open('RowVectors.txt') as rows:
    lines=rows.readline()
rowss=lines.split(",")
for line in rowss:
    row.append(((float(line))))
with open('ColumnVectors.txt') as cols:
    lines=cols.readline()
colss=lines.split(",")
for line in colss:
    col.append(((float(line))))
with open('SVS voxel Translations.txt') as SVS:
    lines=SVS.readline()
SVSTS=lines.split(",")
for line in SVSTS:
    SVST.append(((float(line))))
SVST.append(1)
scannerCoord=[]
with open('PositionVector.txt') as poss:
    lines=poss.readline()
poss=lines.split(",")
for line in poss:
    scannerCoord.append(((float(line))))
scannerCoord.append(1)
normal=[]
with open('Normal.txt') as Norma:
    lines=Norma.readline()
Norma=lines.split(",")
for line in Norma:
    normal.append(((float(line))))

VoiT=VoxelDim[0]
VoiP=VoxelDim[1]
VoiR=VoxelDim[2]
VoiPosSag=SVST[0]
VoiPosCor=SVST[1]
VoiPosTra=SVST[2]

c1=SignPermutations(1,1,1)
```

```
c2=SignPermutations(-1,1,1)
c3=SignPermutations(-1,-1,1)
c4=SignPermutations(1,-1,1)
c5=SignPermutations(1,1,-1)
c6=SignPermutations(-1,1,-1)
c7=SignPermutations(-1,-1,-1)
c8=SignPermutations(1,-1,-1)
oc1=np.zeros((8,4))
for i in range(4):
    oc1[0,i]=c1[i]
    oc1[1,i]=c2[i]
    oc1[2,i]=c3[i]
    oc1[3,i]=c4[i]
    oc1[4,i]=c5[i]
    oc1[5,i]=c6[i]
    oc1[6,i]=c7[i]
    oc1[7,i]=c8[i]
voxpos=np.zeros((4,4))
for i in range(4):
    voxpos[i,i]=1
voxpos[0,3]=VoiPosSag
voxpos[1,3]=VoiPosCor
voxpos[2,3]=VoiPosTra
oc1T=oc1.transpose()
f=np.matmul(voxpos,oc1T)

fT=f.transpose()
for i in range(8):
    fT[i,1]=-fT[i,1]
corner1=[]
corner2=[]
corner3=[]
corner4=[]
corner5=[]
corner6=[]
corner7=[]
corner8=[]
for i in range(3):
```

```
    corner1.append(fT[0,i])
    corner2.append(fT[1,i])
    corner3.append(fT[2,i])
    corner4.append(fT[3,i])
    corner5.append(fT[4,i])
    corner6.append(fT[5,i])
    corner7.append(fT[6,i])
    corner8.append(fT[7,i])

maxdif=[0,0,0]
oppositeCorner=[0,0,0]
for i in range(8):
    for j in range(3):
        if(abs(fT[0,j]-fT[i,j])>maxdif[j]):
            maxdif[j]=abs(fT[0,j]-fT[i,j])
            oppositeCorner[j]=fT[i,j]

midpoint=[]
for i in range(3):
    midpoint.append((corner1[i]+oppositeCorner[i])/2)
midpoint.append(1)

img=nib.load(T1)
hdr=img.header
pixelSize=hdr.get_zooms()
x=pixelSize[0]
MMRI=hdr.get_sform()
MMRIII=linalg.inv(MMRI)
MRSVoxelCoord=np.matmul(MMRIII,midpoint)

degrees1=np.arcsin(normal[0])*180/(np.pi)
degrees2=VoxelDim[3]*180/(np.pi)
degrees3=np.arcsin(normal[1])*180/(np.pi)

os.system('fslmaths '+T1+
' -mul 0 -add 1 -roi '+str(MRSVoxelCoord[0])+' 1 '+ str(MRSVoxelCoord[1])+
 ' 1 ' + str(MRSVoxelCoord[2])+ ' 1 0 1 ACCpoint -odt float')
```

```
if(VoxelDim[0]==VoxelDim[1]==VoxelDim[2]):
    os.system('fslmaths ACCpoint -kernel box ' + str(VoxelDim[1]) +
    ' -fmean ACCVoxel -odt float')
else:
    os.system('fslmaths ACCpoint -kernel boxv3 ' + str(VoxelDim[1]/x) +
     ' ' + str(VoxelDim[0]/x) + ' ' + str(VoxelDim[2]/x) +
      ' -fmean ACCVoxel -odt float')

os.system('fslmaths ACCVoxel.nii.gz -bin ACCVoxel_bin.nii.gz')

os.system('makerot --centre=' +str((MRSVoxelCoord[0]*pixelSize[0]))+
','+str((MRSVoxelCoord[1]*pixelSize[0]))+','+str((MRSVoxelCoord[2]*pixelSize[0]))+
' -a 1,0,0 --theta='+str(degrees1)+' -o matrix1.mat')

os.system('makerot --centre=' +str((MRSVoxelCoord[0]*pixelSize[0]))+
','+str((MRSVoxelCoord[1]*pixelSize[0]))+','+str((MRSVoxelCoord[2]*pixelSize[0]))+
' -a 0,1,0 --theta='+str(degrees2)+' -o matrix2.mat')

os.system('makerot --centre=' +str((MRSVoxelCoord[0]*pixelSize[0]))+
','+str((MRSVoxelCoord[1]*pixelSize[0]))+','+str((MRSVoxelCoord[2]*pixelSize[0]))+
' -a 0,0,1 --theta='+str(degrees3)+' -o matrix3.mat')

os.system('convert_xfm -omat matrix4.mat -concat matrix1.mat matrix2.mat')
os.system('convert_xfm -omat matrix.mat -concat matrix3.mat matrix4.mat')
os.system('flirt -in ACCVoxel_bin.nii.gz -ref ACCVoxel_bin.nii.gz' +
' -out ACCVoxel_bin_rot.nii.gz -applyxfm -init matrix.mat')
os.system('bet '+T1+' extractedbrain.nii.gz -B -f 0.1 -g 0')
os.system('fast -t 1 -n 3 -H 0.1 -I 4 -l 20.0 -o extractedbrain.nii.gz')
print("Volume fraction of CSF:")
os.system('fslstats -t extractedbrain_pve_0.nii.gz -k ACCVoxel_bin_rot.nii.gz -m')
print("Volume fraction of gray matter:")
os.system('fslstats -t extractedbrain_pve_1.nii.gz -k ACCVoxel_bin_rot.nii.gz -m')
print("Volume fraction of white matter:")
os.system('fslstats -t extractedbrain_pve_2.nii.gz -k ACCVoxel_bin_rot.nii.gz -m')
```

B. JavaScript code

```
const fs = require('fs');
const file = process.argv.slice(2)[0];
const content = fs.readFileSync(file, 'utf8');
const array = content.split(/\r?\n/);
let row=[]
let column=[]
let dim=[]
let SVST=[]
let pos=[]
let nor=[]
for (let index = 0; index < array.length; index++) {
    const ads = array[index].split(/\s+=\s+|: /);
    if (array[index].includes('RowVector')){
        row.push(ads[1])
    }
    if (array[index].includes('ColumnVector')){
        column.push(ads[1])
    }
    if (array[index].includes('PositionVector')){
        pos.push(ads[1])
    }
    if (array[index].match('V(o|O)I')) {
        if (array[index].match('Position')) {
            SVST.push(ads[1])
        }
        if (array[index].match('Phase')) {
            dim.push(ads[1])
        }
        if (array[index].match('Readout')) {
            dim.push(ads[1])
```

```
        }
        if (array[index].match('Thickness')) {
            dim.push(ads[1])
        }
        if (array[index].match('InPlane')) {
            dim.push(ads[1])
        }
        if (array[index].match('NormalSag')) {
            nor.push(ads[1])
        }
        if (array[index].match('NormalCor')) {
            nor.push(ads[1])
        }
        if (array[index].match('NormalTra')) {
            nor.push(ads[1])
        }
    }
}
fs.writeFile('Normal.txt', nor,function (err){
    if(err) throw err;
});
fs.writeFile('Voxel Dimensions.txt', dim,function (err){
    if(err) throw err;
});
fs.writeFile('SVS voxel Translations.txt', SVST,function (err){
    if(err) throw err;
});
fs.writeFile('ColumnVectors.txt', column,function (err){
    if(err) throw err;
});
fs.writeFile('RowVectors.txt', row,function (err){
    if(err) throw err;
});
fs.writeFile('PositionVector.txt', pos,function (err){
    if(err) throw err;
});
```