Imaging Multimodality Techniques in Epilepsy

Grau en Enginyeria Biomèdica

Celia Sánchez Laorden

December 28, 2020

Contents

1 Introduction

In Nuclear Medicine, it has been developed a program to localize with high precision the epileptogenic focus. This has been motivated by the need of surgery procedure to remove the origin of crisis in drug-resistant epilepsy. The results from DTI (MRI), video-EEG, SPECT and PET are analysed by a multidisciplinary committee formed by the nuclear doctor, the neuroradiologist, the neuropsychologist, the neurosurgeon, among others. The group of Biomedical Images from the University of Barcelona and the Epilepsy Unit of Hospital Clínic of Barcelona have been using the SISCOM methodology. The first approach was to combine programs such as SPM and MRIcro. Launched in 2008, VPHTk project from CIBER-BBN provided a platform for the implementation of medical applications to provide a friendly user interface with the SISCOM post-processing complex algorithms to the doctors. FocusDET is the program currently in use that enables SISCOM, PET and PISCOM analysis. From now on this report is focused on SISCOM analysis.

It is known that at the beginning of a crisis (ictal state) the blood flow at the focal region increases compared to interictal or basal state. Using [Tc99m]-HMPAO or [Tc99m]-ECD blood perfusion radiotracers, capable to go through the blood-brain barrier and stay inside, is obtained an instantaneous image of the injection or ictal moment using SPECT after the crisis. After few days with no crisis another SPECT is done to record the interictal state. The subtraction of these images remarks the focus. This information is merged with the RM that will provide us with anatomical information. This procedure is known as SISCOM technique (Subtraction of Ictal SPECT CO-registered with MRI). After the post-processing of the results the evaluation is done with simulated images, Monte Carlo projections, comparing a theoretical focus with the experimental focus.

SISCOM has some limitations that should be considered in the post-processing. First, the two SPECT images have been taken in different days and this difficult the process of realignment. Second step is the coregistration of SPECT-MR. Before doing the difference image, a normalization of the activity must be done to equalize acquisitions at different doses of radiotracer due to its time decay. At this point, the background needs to be deleted and what is done is to select the epileptogenic zone (EZ). Finally, the merged EZ is obtained.

This project has been divided into two parts; the first one using SPM12 software has been guided and the second part has been done individually using Python.

2 Methodology

2.1 First Part

First part consisted of the realignment of interictal and ictal SPECT and coregistration of the two SPECT's with the MR. The first step was downloading a folder containing the SPECT interictal and ictal and the MR image in nifti format. The image analysed is number 13. Before doing anything more it had been checked that the origin is at the centre of the brain. With the cursor, the centre of the image was selected, and the values of "mm" were given to the variables "right (mm)", "forward (mm)" and "up (mm)" but changed the sign. This step was the reorientation. The SPM12 option "Realign Estimated Reslice" was selected for calculating the operation and perform the changes. With this step, the ictal image was moved above the interictal, that is why it was selected "Register to first". Finally, in "Resliced Images" option was selected "Images 2...n". For coregistering the option "Coregister Estimated Reslice" was selected. Reference image corresponds to the MR and the source image to the basal SPECT. Additionally, the "rictal" file result of the realignment was passed as "Other images". The images obtained at the end of the sequence were "rrictal", "rinterictal" and "RM".

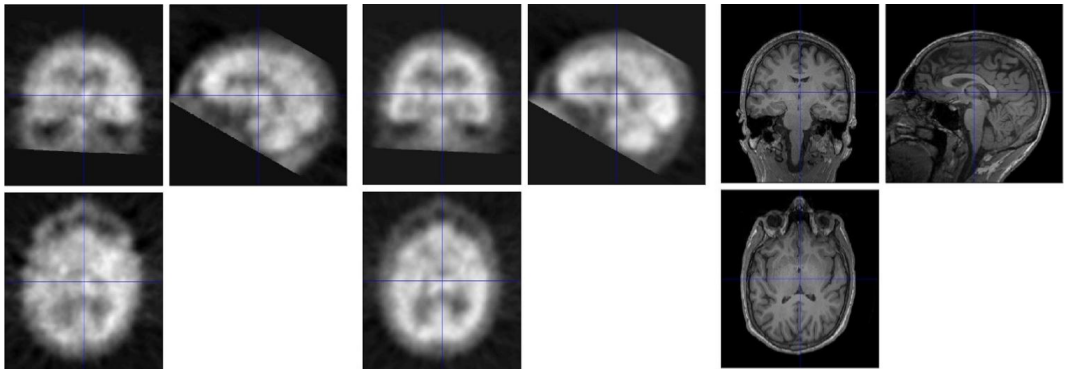


Figure 1: rrictal, rinterictal and RM images.

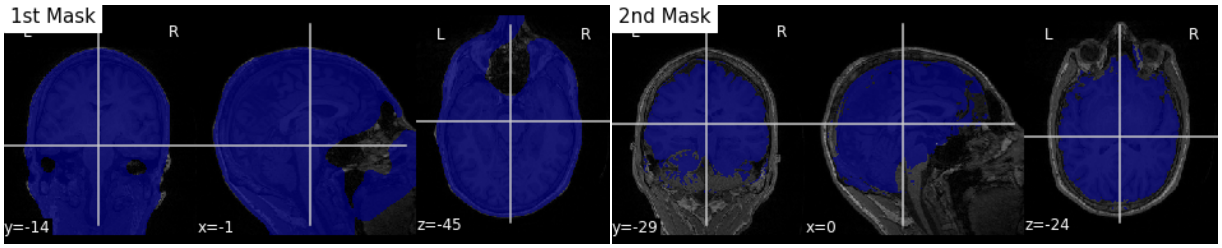
2.2 Second Part

The second part consisted of normalization of intensity, obtaining the difference image, and finally making the selection of the epileptogenic zone and merging the results. The whole process has been carried out with Python using the following packages; NiBabel, os.path, NumPy, Matplotlib, Nilearn, SciPy and DIPY.

To achieve the final aim, first a brain mask has been designed and applied. In order to remove the background and keep only the parts that are in the brain we can use a function called `median_otsu` from the *segment.mask* module of the DIPY package in Pyhton. It uses a median filter to smooth the original magnetic resonance and then Otsu's method. This is an automated histogram method to separate the brain (foreground) from its background. The output of the function are two 3D arrays. The one we want to proceed with normalization is the first one, the mask, an array of 0's and 1's. The values passed to the function are as `median_radius` 4,

radius in voxels of the applied median filter (kernel), and 4 as `num_pass`, number of pass of the median filter. However, `median_otsu` does not eliminate the skull correctly and so a second brain mask is created using MRIcro. Using the different tools that this software includes, the brain is saved as a ROI and applied to the previous obtained masked magnetic resonance scan. The final mask is applied multiplying, as it was a filter, to the SPECT ictal and interictal scans.

Figure 2: Comparisson between RM wihout and with the 2 masks.



(a) First Mask using Python. **(b)** Second Mask using MRIcro.

Figure 3: Plot cuts of mask 1 and 2.

To normalize different levels of radioisotope uptake, retention, and decay, different methods have been tested and compared using the histograms of the two SPECT scans, before and after. The following plot is the histogram of the ictal, in green, and the interictal scan, in red, without normalization: Before calculating the factor of intensity normalization, the outlier

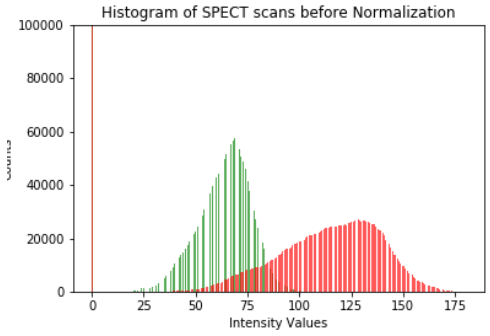
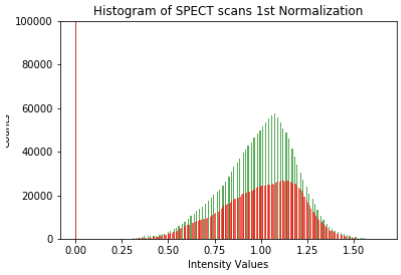


Figure 4: Histogram of the intensity values distribution of SPECT scans before Normalization.

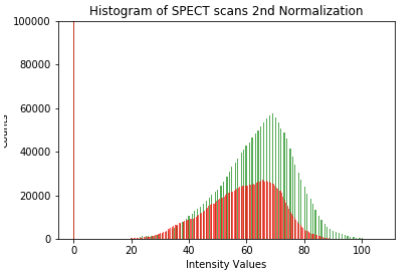
values from the ictal scan should be eliminated. These outliers are caused by high levels of hyperperfusion. At first glance, the distribution of intensity values from the ictal scan seems normal and therefore triple de standard deviation could be used to eliminate the outliers, both at the top and the bottom. However, after using two statistical tests it is concluded that the sample does not follow a Gaussian distribution. The statistical tests used to evaluate the ictal

sample are D'Agostino's K^2 test and Anderson-Darling Test, coming the last one from a more sophisticated statistical test called the Kolmogorov-Smirnov test. These tests are given by the Stats module from SciPy package. The final choice is to use percentiles 5 and 95 to find the outliers using a NumPy function.

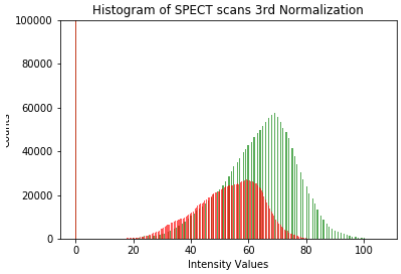
The first method consists of calculating the mean value of all non-zero voxels, mean in range, throughout the entire masked volume images. Voxel intensities are rescaled to the Mean in Range and multiplied by 100. This method was reported in a guide to assist the process of SISCOM method [**1**]. The second method that has been used is a normalization based on the total counts. First, the ratio of the total counts of ictal and interictal SPECT scans is calculated, and then the interictal data array is multiplied with it. This method was used to normalize the intensity in a paper dedicated to the study difference images calculated from ictal and interictal Technetium-99m-HMPAO SPECT scans of epilepsy [**2**]. The third method is based on the maximum intensity ratio and this ratio is applied, as in the previous method, to the interictal array. This method has been though discussing the previous method with another student. The last normalization tested consists of getting the maximum value of each volume and divide each voxel of this volume by this value. The following plots are the histograms obtained from the ictal and interictal SPECTs once normalized. The best normalization are obtained in the last two methods.



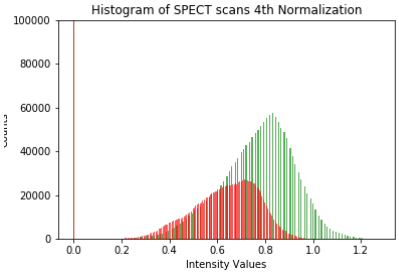
(a) Mean in Range Normalization.



(b) Total Counts Ratio Normalization.



(c) Maximum intensity ratio Normalization.



(d) Maximum intensity re-scale Normalization.

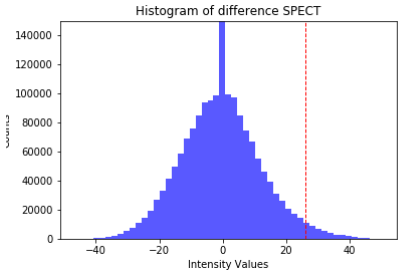
Figure 5: Histograms of the distribution of ictal and interictal SPECT after Normalization.

Before normalizing, the difference image is obtained. However, we cannot plot it yet because we will not see the epileptogenic focuses. That is why, a threshold has been defined to eliminate the intensities that are below it. The rule is to multiply the standard deviation by 6 and sum the median of the difference image. You can find the histograms for the difference image in the Results section. Once the intensities are filtered, the results are overlayed with the resonance

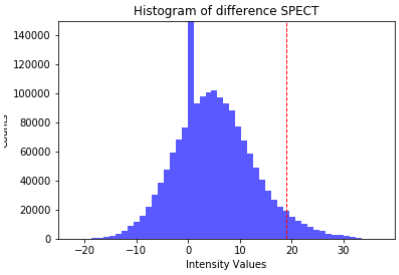
image. You can find all the Python code commented in the Annex.

3 Results

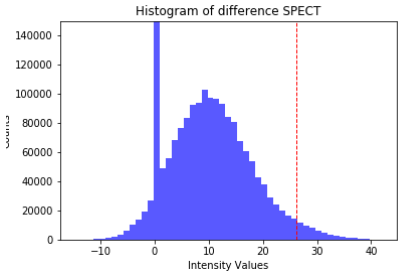
As mentioned in the previous section, a threshold has been defined to determine the epileptic focuses. In the following histograms, the distribution of intensity values of the difference images is shown and the threshold is marked in red:



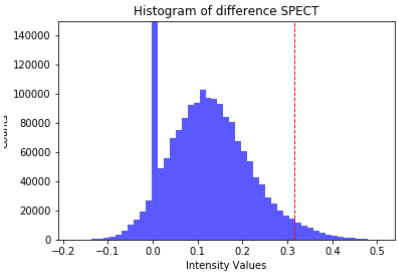
(a) Mean in Range Normalization.



(b) Total Counts Ratio Normalization.



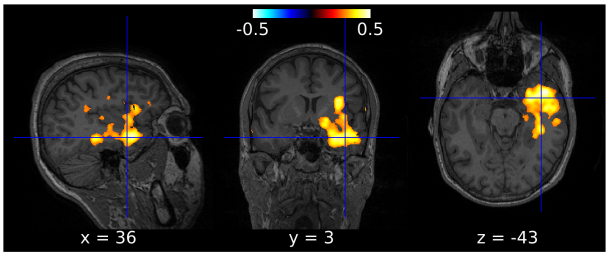
(c) Maximum intensity ratio Normalization.



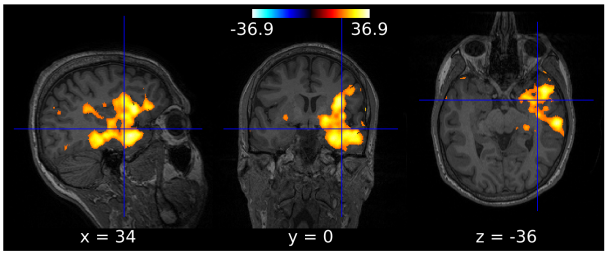
(d) Maximum intensity re-scale Normalization.

Figure 6: Histograms of the difference SPECT image with the threshold to determine significant diferences.

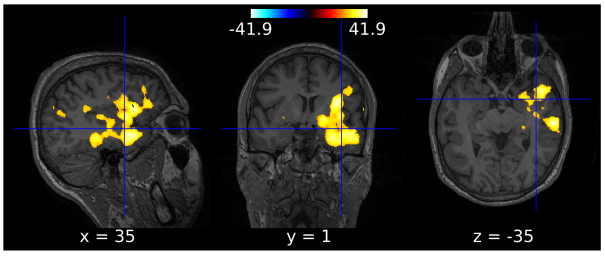
Below, the resulting images are presented.



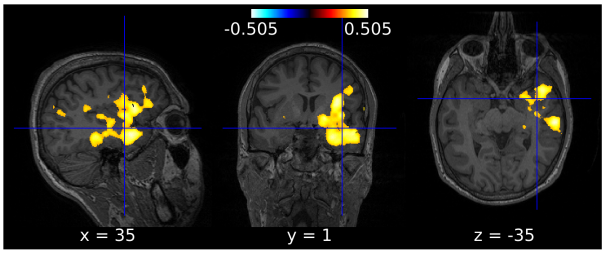
(a) Mean in Range Normalization.



(b) Total Counts Ratio Normalization.



(c) Maximum intensity ratio Normalization.



(d) Maximum intensity re-scale Normalization.

Figure 7: Difference SPECT images overlayed with RM image.

Using the Mean in Range method a good intensity normalization is achieved over the two SPECT scans. However, we should avoid using the mean or the median to normalize because the ictal intensity values of the focus led to bias as can be observed in Figure 7a). The final choice has been the Maximum Intensity Ratio Normalization. We can observe that the intensity values before the normalization and subtraction of SPECTs are the highest. This can be translated in an advantage for visualizing the focuses. The same criteria has been used to determine the threshold in the four cases, and the images that have smaller areas of higher concentration are the ones from the two last normalization, maximum ratio and maximum re-scaling. To normalize one SPECT intensity, the fourth method does not take into account the other SPECT. For the last reason, the third normalization is better than the fourth one. As with maximum intensity ratio normalization we are getting a normalization factor, this method can be easily implemented to different scenarios returning only one parameter. The following is a plot of the ROI's with a colour scale where the warm colour represent the higher intensities.

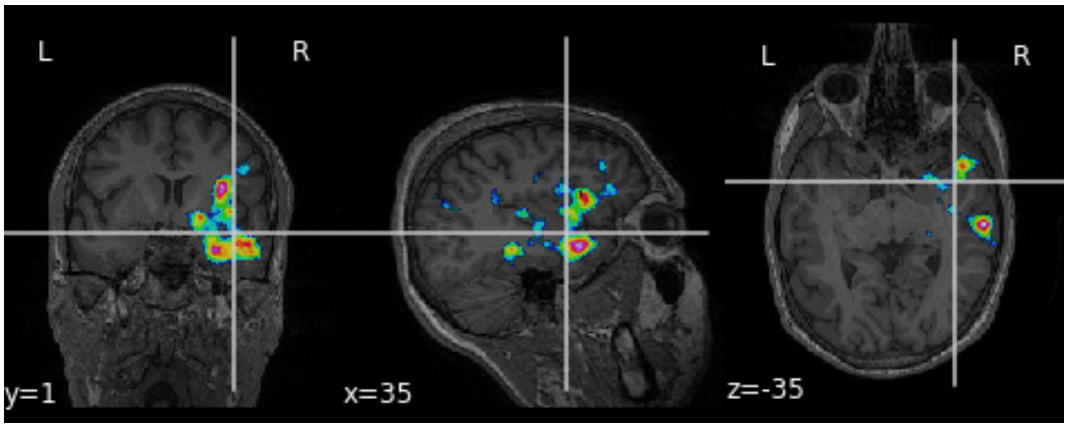


Figure 8: Plot cuts of ROI's from the difference SPECT image.

References

[1] SISCOM (Subtraction Ictal SPECT CO-registered to MRI, http://www.synapticom.net

[2] I G Zubal, S S Spencer, K Imam, J Seibyl, E O Smith, G Wisniewski, P B Hoffer. *Difference images calculated from ictal and interictal technetium-99m-HMPAO SPECT scans of epilepsy.* Journal of Nuclear Medicine. Department of Diagnostic Radiology, University of Yale, New Haven, Connecticut, 1995.

Appendix A Python Code for the Second Part

```
#packages
import nibabel as nib
import os.path
import numpy as np
import matplotlib
import matplotlib.pyplot as plt
from nilearn import plotting
from nilearn.masking import compute_epi_mask, compute_background_mask
from nilearn.plotting import plot_roi
from dipy.core.histeq import histeq
from dipy.segment.mask import median_otsu
from dipy.data.fetcher import fetch_scil_b0, read_siemens_scil_b0
from scipy.stats import normaltest, anderson

#functions
def import_image(path):
    if os.path.isfile(path):
        #nii image
        img_in = nib.nifti1.load(path)
        #data volume
        data_in = img_in.get_fdata()
        header_in = img_in.header
    else:
        print('no se encuentra archivo: '+ path)

    return img_in, data_in, header_in

def save_image(path, img):
    if os.path.isfile(path):
        nib.save(img, path)
    else:
        print('no se encuentra archivo: '+ path)

#paths, here you should include yours
ictal_path = r"C:\Users\Celia\Desktop\P-2\rICTAL.nii"
interictal_path = r"C:\Users\Celia\Desktop\P-2\rrINTERICTAL.nii"
RM_path = r"C:\Users\Celia\Desktop\P-2\RM.nii"
```

```
ictal_img, ictal_data, ictal_header = import_image(ictal_path)
interictal_img, interictal_data, interictal_header = import_image(interictal_path)
RM_img, RM_data, RM_header = import_image(RM_path)

#plotting the images of SPECT and RM
plotting.plot_img(ictal_img, title = 'ictal SPECT')
plotting.plot_img(interictal_img, title = 'interictal SPECT')
plotting.plot_img(RM_img, title = 'RM')

# 1st Brain mask in RM using Median Otsu's Method
img = RM_img
data = np.squeeze(RM_data)
b0_mask, mask = median_otsu(data, 4, 4)
#we used 4 as median_radius radius in voxels of the applied median filter and
#4 as num_pass Number of pass of the median filter

#Representation
mask_img = nib.Nifti1Image(mask.astype(np.float32), img.affine) #mask edited
b0_img = nib.Nifti1Image(b0_mask.astype(np.float32), img.affine) #mask edited
plot_roi(mask_img, RM_img, title = "1st Mask", output_file = "first_mask.png")

#save results from the fist mask approach, only for making the second mask
#fname = 'se_1.5t'
#nib.save(mask_img, fname + '_binary_mask.nii.gz')
#nib.save(b0_img, fname + '_mask.nii.gz')

#2nd Brain mask in RM using MRIcro
#here we upload the new mask done with MRIcro software
path_mask = nib.nifti1.load(r"C:\Users\Celia\Desktop\P-2\mse_1.5t_mask_2.hdr")
mask_edited = path_mask.get_fdata()
mask_edited_binary = np.where(mask_edited!=0, 1, mask_edited)
#plotting the new mask
mask_edited_binary_img = nib.Nifti1Image(mask_edited_binary.astype(np.float32), img.affine)
plot_roi(mask_edited_binary_img, RM_img, title = "2nd Mask", output_file =
"second_mask.png")
#comparisson between RM wihout and with the 2 masks
sli = data.shape[2] // 2
plt.figure('Brain segmentation')
plt.subplot(1, 3, 1).set_axis_off()
```

```
plt.imshow(histeq(data[:, :, sli].astype('float')).T,
           cmap='gray', origin='lower')

plt.subplot(1, 3, 2).set_axis_off()
plt.title("RM without and with 1rst and 2nd mask")
plt.imshow(histeq(b0_mask[:, :, sli].astype('float')).T,
           cmap='gray', origin='lower')

plt.subplot(1, 3, 3).set_axis_off()

plt.imshow(histeq(mask_edited[:, :, sli].astype('float')).T,
           cmap='gray', origin='lower')
plt.savefig("comparisson_rm.png", format = 'png')

plt.show()

#applying the mask as a filter
ictal_data_mask = np.multiply(mask_edited_binary, ictal_data)
interictal_data_mask = np.multiply(mask_edited_binary, interictal_data)

#histogram representation of SPECT ictal and interictal
plt.figure('histogram')
result = ictal_data_mask.flatten()
n, bins, patches = plt.hist(result, bins = 256, facecolor = 'green', alpha = 0.65)
#bandwith = 1
result2 = interictal_data_mask.flatten()
n, bins, patches = plt.hist(result2, bins = 256, facecolor = 'red', alpha = 0.65)
plt.ylim(0, 100000)
plt.xlabel('Intensity Values')
plt.ylabel('Counts')
plt.title('Histogram of SPECT scans before Normalization')
plt.savefig("hist_before.png", format = 'png')
plt.show()

#Pre-normalization steps to the ictal SPECT
#It seems that ictal scan has a normal distribution, and so we can use standard deviation
#to elimante the outliers the outliers come from the hyperperfusion values
#We are going to test normal distribution first
```

```
#We elimate the zeroes and we order the dataset of ictal scan
ictal_no_zeroes = ictal_data_mask.flatten()
ictal_no_zeroes = ictal_no_zeroes[ictal_no_zeroes != 0]
sort_ictal = np.sort(ictal_no_zeroes.flatten())

# D'Agostino and Pearson's Test
# normality test
stat, p = normaltest(sort_ictal)
print('Statistics=%.3f, p=%.3f' % (stat, p))
# interpret
alpha = 0.05
if p > alpha:
    print('Ictal array looks Gaussian (fail to reject H0)')
else:
    print('Ictal array does not look Gaussian (reject H0)')

#Anderson test
# normality test
result = anderson(sort_ictal)
print('Statistic: %.3f' % result.statistic)
p = 0
# interpret
for i in range(len(result.critical_values)):
    sl, cv = result.significance_level[i], result.critical_values[i]
    if result.statistic < result.critical_values[i]:
        print('%.3f: %.3f, Ictal array looks normal (fail to reject H0)' % (sl, cv))
    else:
        print('%.3f: %.3f, Ictal array does not look normal (reject H0)' % (sl, cv))

#We cannot use the standard deviation criteria to eliminate the outliers
#Instead we use percentiles, we calculate the values greater than 95%
#and smaller than 0.05%
print("Percentile 5 ", np.percentile(sort_ictal,5))
print("Percentile 95 ", np.percentile(sort_ictal,95))
ictal_data_quantiles = np.delete(sort_ictal, np.where(sort_ictal <
np.percentile(sort_ictal,5)))
ictal_data_quantiles = np.delete(ictal_data_quantiles, np.where(ictal_data_quantiles >
np.percentile(sort_ictal,95)))
```

```
#here starts the Normalization
#First normalization:
#consists of calculating the mean in range of each SPECT and normalize by these values.

#from now on, in each normalization, we use the ictal_data_quantiles array to calculate
#the normalization factor
mean_ictal = ictal_data_quantiles.mean()

interictal_data_mask_non_zero = np.ma.masked_equal(interictal_data_mask,0)
#the mean in range calculates the media without 0's
mean_interictal = interictal_data_mask_non_zero.mean()

print("Mean in Range for Ictal SPECT: ",round(mean_ictal,2))
print("Mean in Range for Interictal SPECT: ",round(mean_interictal,2))

ictal_norm = np.divide(ictal_data_mask, mean_ictal) #normalizing by the Mean in Range
interictal_norm = np.divide(interictal_data_mask, mean_interictal)

norm_result = np.subtract(ictal_norm,interictal_norm) #subtraction
norm_result_100 = np.multiply(np.subtract(ictal_norm,interictal_norm),100)

plt.figure('histogram1')
result = ictal_norm.flatten()
n, bins, patches = plt.hist(result, bins = 256, facecolor = 'green', alpha = 0.65)
result2 = interictal_norm.flatten()
n, bins, patches = plt.hist(result2, bins = 256, facecolor = 'red', alpha = 0.65)
plt.ylim(0, 100000)
plt.xlabel('Intensity Values')
plt.ylabel('Counts')
plt.title('Histogram of SPECT scans 1st Normalization')
plt.savefig("hist_1.png", format = 'png')
plt.show()

#here a threshold is set calculating the standard deviation and multiplying by 6
std_1 = np.std(norm_result_100)*6
thr_1 = np.median(norm_result_100.flatten())+std_1
print("The intensities smaller than ",thr_1,"are eliminated.")
plt.figure('histogram2')
result = norm_result_100.flatten()
```

```
n, bins, patches = plt.hist(result, bins = int(256/5), facecolor = 'blue', alpha = 0.65)
#bandwith = 5
plt.xlabel('Intensity Values')
plt.ylabel('Counts')
plt.title('Histogram of difference SPECT')
plt.xlabel('Intensity Values')
plt.ylabel('Counts')
plt.axvline(x=thr_1, color='r', linestyle='dashed', linewidth=1)
plt.ylim(0, 150000)
plt.savefig("hist_1_1.png", format = 'png')
plt.show()

norm_result_100 = np.where(norm_result_100<thr_1, 0, norm_result)
norm_img = nib.Nifti1Image(norm_result_100.astype(np.float32), img.affine)
#plot_roi(norm_img, bg_img=RM_img, output_file = "first_norm.png")
html_view = plotting.view_img(norm_img, bg_img=RM_img)
html_view.save_as_html('norm1.html')

#Second Normalization:
#consists of calculating the ratio of total number of counts between ictal and
#interictal scan. This normalization is applied to the interictal scan.
ratio_ictal_interictal = np.sum(ictal_data_quantiles)/np.sum(interictal_data_mask)
interictal_ratio = np.multiply(interictal_data_mask,ratio_ictal_interictal)
difference_norm_ratio = np.subtract(ictal_data_mask, interictal_ratio)
print("The ratio of the sum of ictal and interictal ", ratio_ictal_interictal)

plt.figure('histogram2')
result = ictal_data_mask.flatten()
n, bins, patches = plt.hist(result, bins = 256, facecolor = 'green', alpha = 0.65)
result2 = interictal_ratio.flatten()
n, bins, patches = plt.hist(result2, bins = 256, facecolor = 'red', alpha = 0.65)
plt.ylim(0, 100000)
plt.xlabel('Intensity Values')
plt.ylabel('Counts')
plt.title('Histogram of SPECT scans 2nd Normalization')
plt.savefig("hist_2.png", format = 'png')
plt.show()

std_2 = np.std(difference_norm_ratio)*6
```

```
thr_2 = np.median(difference_norm_ratio.flatten())+std_2
print("The intensities smaller than ",thr_2,"are eliminated.")
plt.figure('histogram2')
result = difference_norm_ratio.flatten()
n, bins, patches = plt.hist(result, bins = int(256/5), facecolor = 'blue', alpha = 0.65)
plt.xlabel('Intensity Values')
plt.ylabel('Counts')
plt.title('Histogram of difference SPECT')
plt.xlabel('Intensity Values')
plt.ylabel('Counts')
plt.axvline(x=thr_2, color='r', linestyle='dashed', linewidth=1)
plt.ylim(0, 150000)
plt.savefig("hist_2_1.png", format = 'png')
plt.show()

difference_norm_ratio = np.where(difference_norm_ratio<thr_2, 0,
difference_norm_ratio)
dif_norm_ratio_img =
nib.Nifti1Image(difference_norm_ratio.astype(np.float32), img.affine)
plot_roi(dif_norm_ratio_img, bg_img=RM_img, output_file =
"second_norm.png")
html_view2 = plotting.view_img(dif_norm_ratio_img, bg_img=RM_img)
html_view2.save_as_html('norm2.html')

#Third Normalization:
#consists of calculating the ratio of maximum intensity between the one
#from ictal and the one from the interictal scan. This normalization is
#applied to the interictal scan.

ratio_ictal_interictal_max = np.nanmax(ictal_data_quantiles)/
np.nanmax(interictal_data_mask)
interictal_ratio_max = interictal_data_mask*ratio_ictal_interictal_max
difference_norm_ratio_max = np.subtract(ictal_data_mask,
interictal_ratio_max)
print("The ratio of the maximum intensity of ictal and interictal ",
ratio_ictal_interictal_max)

plt.figure('histogram3')
result = ictal_data_mask.flatten()
```

```
n, bins, patches = plt.hist(result, bins = 256, facecolor = 'green', alpha = 0.65)
result2 = interictal_ratio_max.flatten()
n, bins, patches = plt.hist(result2, bins = 256, facecolor = 'red', alpha = 0.65)
plt.xlabel('Intensity Values')
plt.ylabel('Counts')
plt.title('Histogram of SPECT scans 2nd Normalization')
plt.xlabel('Intensity Values')
plt.ylabel('Counts')
plt.title('Histogram of SPECT scans 3rd Normalization')
plt.ylim(0, 100000)
plt.savefig("hist_3.png", format = 'png')
plt.show()

std_3 = np.std(difference_norm_ratio_max)*6
thr_3 = np.median(difference_norm_ratio_max.flatten())+std_3
print("The intensities smaller than ",thr_3,"are eliminated.")
plt.figure('histogram')
result = difference_norm_ratio_max.flatten()
n, bins, patches = plt.hist(result, bins = int(256/5), facecolor = 'blue', alpha = 0.65)
plt.xlabel('Intensity Values')
plt.ylabel('Counts')
plt.title('Histogram of difference SPECT')
plt.xlabel('Intensity Values')
plt.ylabel('Counts')
plt.axvline(x=thr_3, color='r', linestyle='dashed', linewidth=1)
plt.ylim(0, 150000)
plt.savefig("hist_3_1.png", format = 'png')
plt.show()

difference_norm_ratio_max = np.where(difference_norm_ratio_max<
thr_3, 0, difference_norm_ratio_max)
dif_norm_ratio_max_img =
nib.Nifti1Image(difference_norm_ratio_max.astype(np.float32),
img.affine)
plot_roi(dif_norm_ratio_max_img, bg_img=RM_img, output_file =
"third_norm.png")
html_view3 = plotting.view_img(dif_norm_ratio_max_img, bg_img=RM_img)
html_view3.save_as_html('norm3.html')
```

```
#Fourth Normalization:
#consists on calculating the maximum value of each volume and divide
#each voxel of this volume by this value.
max_ictal = np.max(ictal_data_quantiles)
max_interictal = np.max(interictal_data_mask)
norm_max_ictal = np.divide(ictal_data_mask, max_ictal)
norm_max_interictal = np.divide(interictal_data_mask, max_interictal)
norm_max = np.subtract(norm_max_ictal, norm_max_interictal)

plt.figure('histogram4')
result = norm_max_ictal.flatten()
n, bins, patches = plt.hist(result, bins = 256, facecolor = 'green', alpha = 0.65)
result2 = norm_max_interictal.flatten()
n, bins, patches = plt.hist(result2, bins = 256, facecolor = 'red', alpha = 0.65)
plt.xlabel('Intensity Values')
plt.ylabel('Counts')
plt.title('Histogram of SPECT scans 4th Normalization')
plt.ylim(0, 100000)
plt.savefig("hist_4.png", format = 'png')
plt.show()

std_4 = np.std(norm_max)*6
thr_4 = np.median(norm_max.flatten())+std_4
print("The intensities smaller than ",thr_4,"are eliminated.")
plt.figure('histogram')
result = norm_max.flatten()
n, bins, patches = plt.hist(result, bins = int(256/5), facecolor = 'blue', alpha = 0.65)
plt.xlabel('Intensity Values')
plt.ylabel('Counts')
plt.title('Histogram of difference SPECT')
plt.xlabel('Intensity Values')
plt.ylabel('Counts')
plt.axvline(x=thr_4, color='r', linestyle='dashed', linewidth=1)
plt.ylim(0, 150000)
plt.savefig("hist_4_1.png", format = 'png')
plt.show()

norm_max = np.abs(np.where(norm_max<thr_4, 0, norm_max))
norm_max_img = nib.Nifti1Image(norm_max.astype(np.float32), img.affine)
```

```
plot_roi(norm_max_img, bg_img=RM_img, output_file = "fourth_norm.png")
html_view4 = plotting.view_img(norm_max_img, bg_img=RM_img)
html_view4.save_as_html('norm4.html')
```

Appendix B Link to GitHub Repository with the downloadable Jupyter Notebook

GitHub Repository link