

Introduction

We can use the Dice similarity coefficient to obtain a numerical measure of the similarity of two MRI images. The document describes how to use a Python script that calculates the Dice coefficient between two anatomical images in BrainVoyager (VMRs).

This can be used to evaluate image registration, image segmentation or EPI distortion correction. For EPI distortion correction, create a VTC, click “Show VTC Vol” on the 3D Volume tools and save as secondary VMR via the BrainVoyager File menu option.

The Dice similarity coefficient (or Sørensen-Dice coefficient) is the ratio of the overlap of images A and B :

$$\frac{2|A \cap B|}{|A| + |B|}$$

which is in words two times the size of the overlap between images A and B divided by the size of image A plus the size of image B , where ‘size’ is here interpreted as the number of marked voxels.

Please note that there are advantages and disadvantages of using the Dice coefficient and for some situations, another metric might be preferred. Another regularly used measure for image is the Jaccard index. For multi-modal image similarity, the normalised gradient field (NGF) by Modersitzki & Haber, mutual information (MI) by Viola & Wells, normalised mutual information (NMI) by Studholme et al, correlation ratio (CR) by Roche et al or the Woods function by Woods et al.

Latest update: 22-03-21

Manual approach

In this example, we use the Getting Started Guide 4.0 example data¹ and normalize this to MNI space, using the ICBM template². We calculate two coefficients: one before and one after MNI normalization.

First Dice coefficient: before MNI normalization

Binarize the images: reduce to two colours We reduce the image to two colours, where each voxel either represents brain or background. To binarize the images (*.vmr), we mark every non-relevant voxel black (= 0) and all other voxels blue (= 240). First, keep all voxels larger than 60 via 3D Volume Tools → Segment → Range → Keep marked (Figure 1). Save the *.vmr and click again ‘Range’ to mark with color 240 (blue) (see Figure 2 and Figure 3); save and close the volumes.



Figure 1: Reload marked to preserve just the marked voxels

¹set 0009, CG_3DT1MPR_SCRIPT_IHHC_SAG.vmr

²ICBM452-IN-MNI152-SPACE_BRAIN.vmr

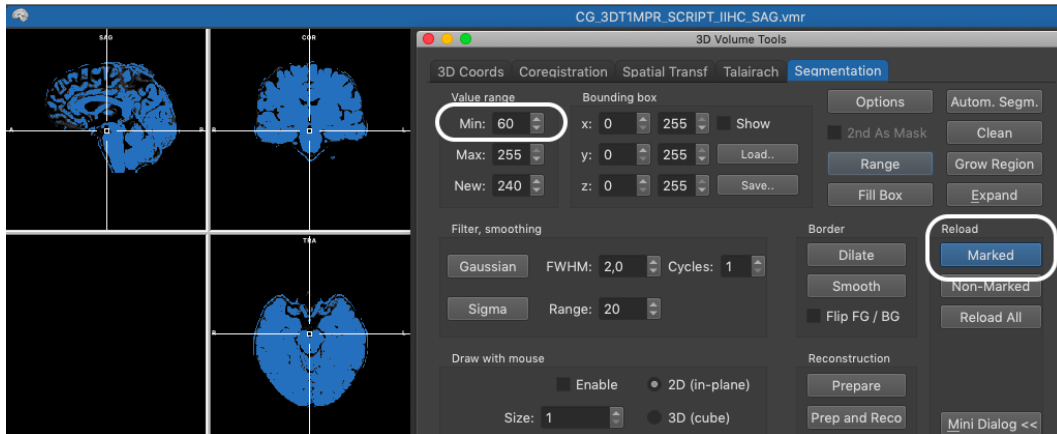


Figure 2: Marked anatomical image

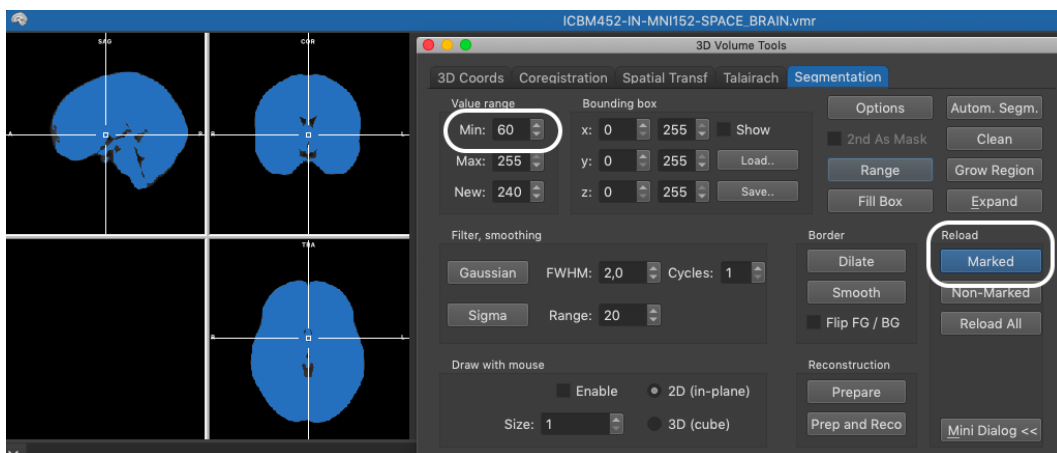


Figure 3: Marked template image

Merge the binarised images To obtain the size of the joint images, we merge the binarised images. First, load the anatomical image (*.vmr) we would like to compare to. Then, we go to Segment → Options → Mask and select first the registered, marked, volume and click the AND button. Save as ICBM452_*.*_marked_merged_with_IIHC_SAG.vmr.

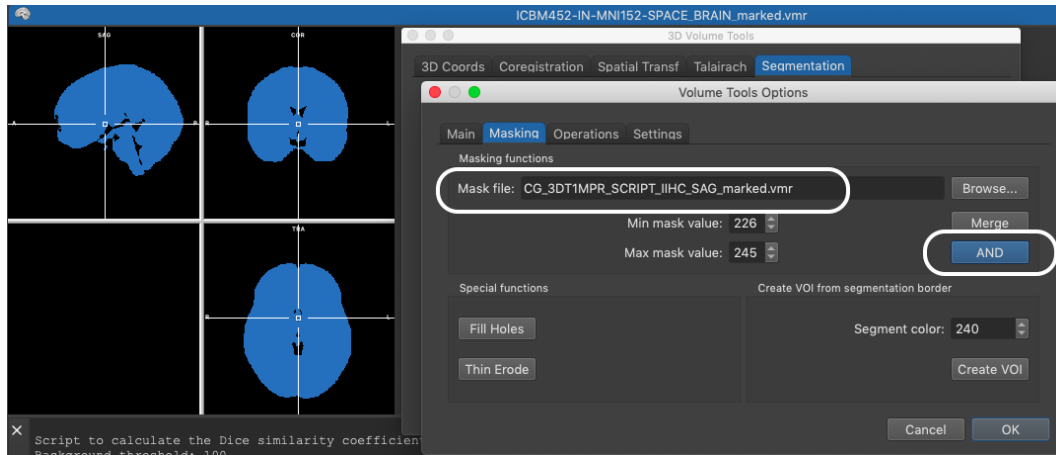


Figure 4: Merged images

Count voxels of merged and binarised images Count the voxels of the merged image via 3D Volume Tools → Options → Volumetry → Count (Figure 5). Load the marked anatomical image and count the voxels (Figure 6), as well as the voxels in the target *.vmr (Figure 7).

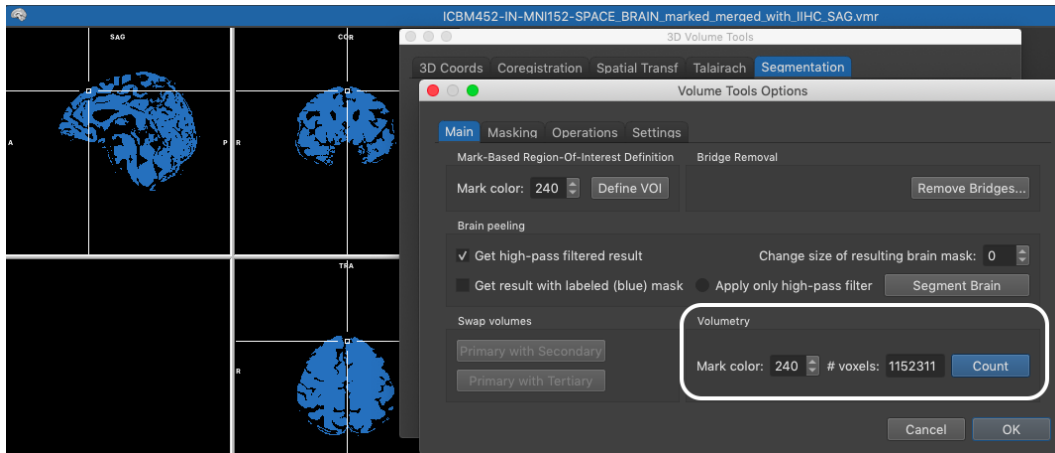


Figure 5: Count voxels of merged images

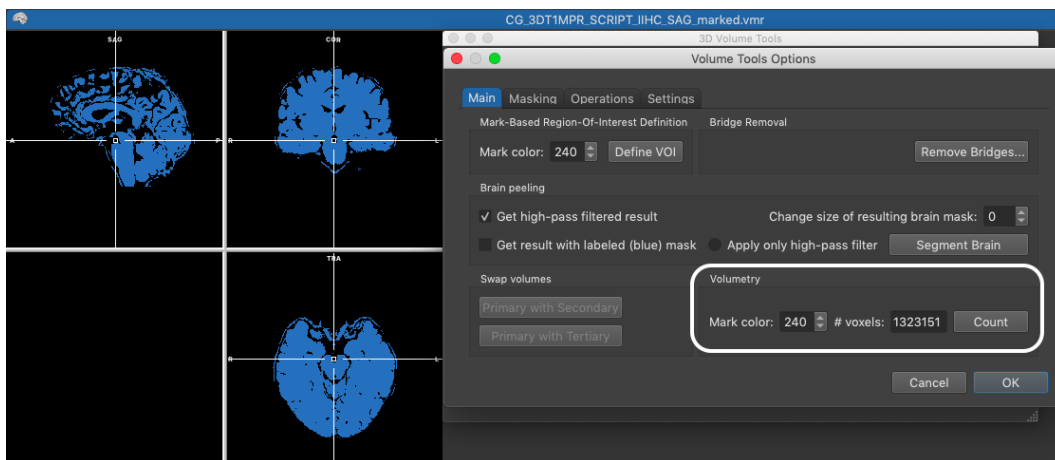


Figure 6: Count voxels of anatomical image

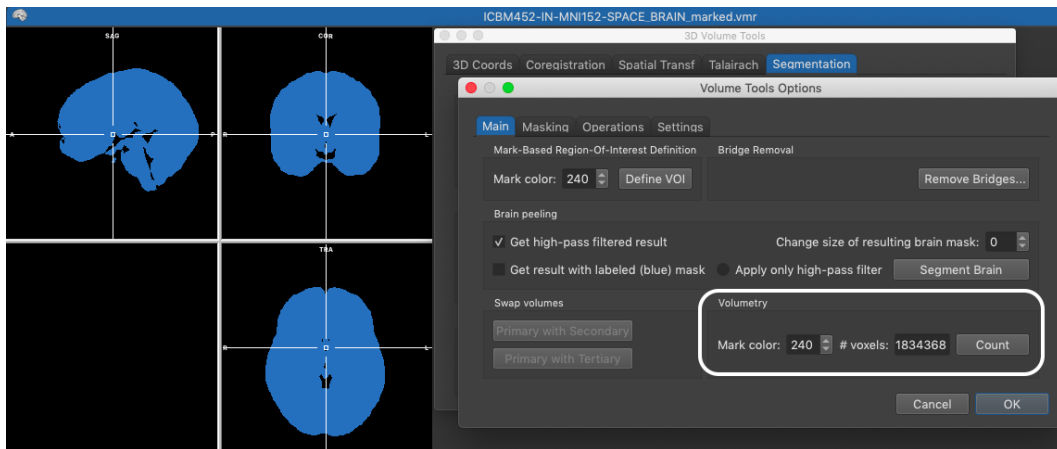


Figure 7: Count voxels of template image

Calculate the Sørensen-Dice coefficient We have 1152311 voxels in the merged image. In the marked image in native space, we count 1323151. The marked ICBM image has 1834368 voxels. We use the formula:

$$\frac{2|A \cap B|}{|A| + |B|} = \frac{2 \times 1152311}{(1323151 + 1834368)} = 0.7299$$

Second Dice coefficient: after MNI normalization

Now we repeat the binarization, marking, merging and counting for normalized image and the template MNI image.

For the merged images in MNI space with threshold 60 we count 1387075 voxels. In the normalized image we count 1420266 voxels.

So the calculation is:

$$\frac{2|A \cap B|}{|A| + |B|} = \frac{2 \times 1387075}{(1420266 + 1834368)} = 0.8524$$

Hence the normalization of the anatomical image from native space to MNI space improved the Dice similarity from 0.7299 to 0.8524.

Using an existing Python script

First, place the script in the appropriate folder (for BrainVoyager 22, this is /Documents/BrainVoyager/Extensions/PythonScripts/). We will compare the Dice coefficients between a non-normalized image and an MNI template, and a normalized image and an MNI template. Therefore we have to run the script twice, to obtain two Dice coefficients.

Start BrainVoyager after copying the script to the Python scripts folder. Click the Python icon (see Figure 8) in the top right corner of the BrainVoyager window.



Figure 8: Activate the Python editor via the Python icon

This will open the Python script editor. Load the script `image_registration_evaluation_with_dice_coefficient_bv22_v01.py` via the “Load” button. Click “Run”. We can select a threshold for binarising the image. In this case, we choose threshold 60 to cover as much of the brain as possible without including background noise (see Figure 9).

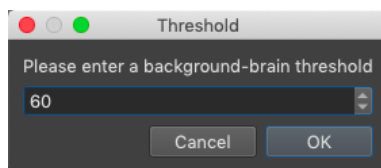
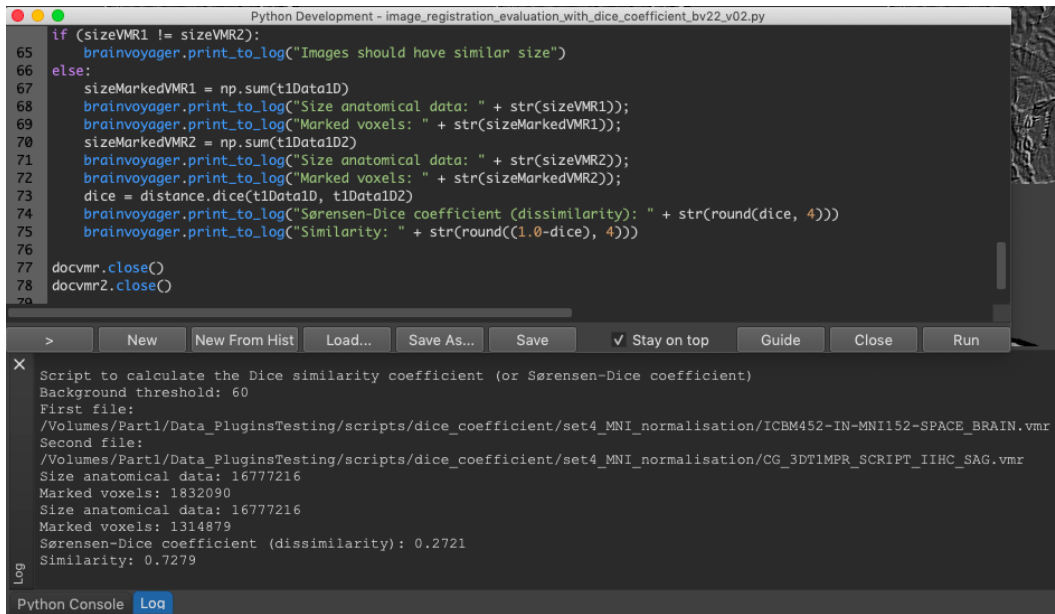


Figure 9: Select a threshold

File dialogs will appear two times, once for each VMR. In this example, for the first file we select an MNI template, copied from the folder “MNITemplates” in the BrainVoyager application directory³. The similarity between the MNI template and the non-normalized Getting Started Guide image is printed to the BrainVoyager Log tab (see Figure 10).

³On macOS, right-click on the BrainVoyager icon in the Finder to show the package contents; the image “ICBM452-IN-MNI152-SPACE.BRAIN.vmr” can be found in folder /Contents/Resources/MNITemplates/.



```
Python Development - image_registration_evaluation_with_dice_coefficient_bv22_v02.py
65 if (sizeVMR1 != sizeVMR2):
66     brainvoyager.print_to_log("Images should have similar size")
67 else:
68     sizeMarkedVMR1 = np.sum(t1Data1D)
69     brainvoyager.print_to_log("Size anatomical data: " + str(sizeVMR1));
70     brainvoyager.print_to_log("Marked voxels: " + str(sizeMarkedVMR1));
71     sizeMarkedVMR2 = np.sum(t1Data1D2)
72     brainvoyager.print_to_log("Size anatomical data: " + str(sizeVMR2));
73     brainvoyager.print_to_log("Marked voxels: " + str(sizeMarkedVMR2));
74     dice = distance.dice(t1Data1D, t1Data1D2)
75     brainvoyager.print_to_log("Sørensen-Dice coefficient (dissimilarity): " + str(round(dice, 4)))
76     brainvoyager.print_to_log("Similarity: " + str(round((1.0-dice), 4)))
77
78 docvmr.close()
79 docvmr2.close()
80
```

```
X Script to calculate the Dice similarity coefficient (or Sørensen-Dice coefficient)
Background threshold: 60
First file:
/Volumes/Part1/Data_PluginsTesting/scripts/dice_coefficient/set4_MNI_normalisation/ICBM452-IN-MNI152-SPACE_BRAIN.vmr
Second file:
/Volumes/Part1/Data_PluginsTesting/scripts/dice_coefficient/set4_MNI_normalisation/CG_3DT1MPR_SCRIPT_IHC_SAG.vmr
Size anatomical data: 16777216
Marked voxels: 1832090
Size anatomical data: 16777216
Marked voxels: 1314879
Sørensen-Dice coefficient (dissimilarity): 0.2721
Similarity: 0.7279
Log
Python Console Log
```

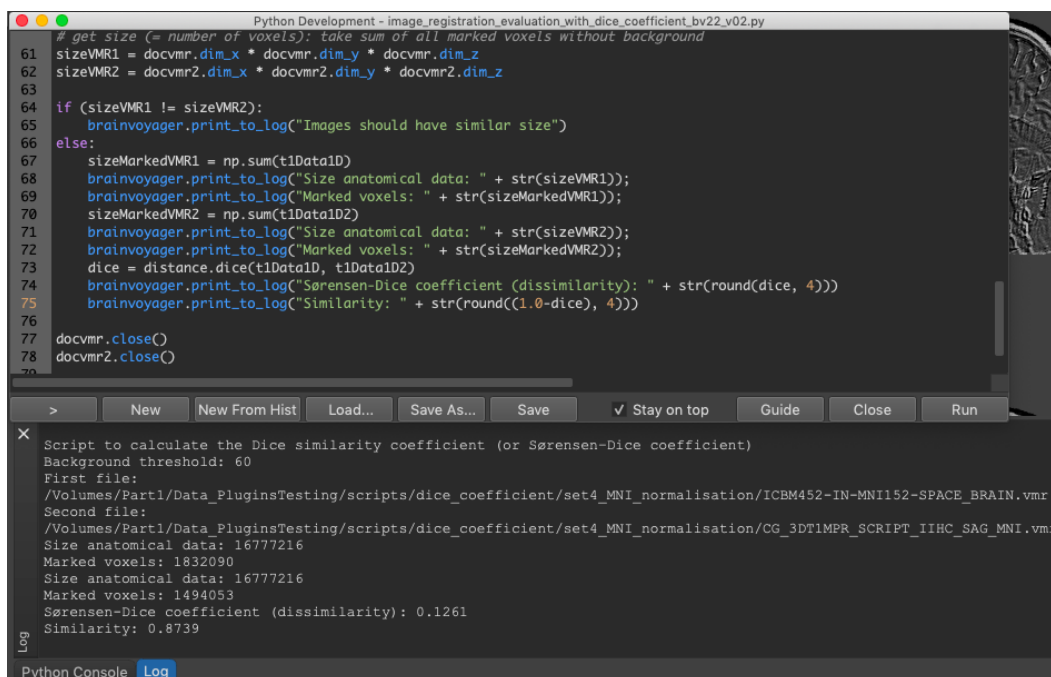
Figure 10: The similarity between the MNI template and the non-normalized Getting Started Guide image is printed to the BrainVoyager Log tab

As second VMR file, we select an inhomogeneity corrected VMR in standard 256 x 256 x 256 size from the Getting Started Guide v4.0. The script calculates the Sørensen-Dice coefficient using SciPy⁴ and prints the value to the BrainVoyager Log tab.

The script gives a similarity of 0.7279 (or dissimilarity of 0.2721) between the image in native space and the ICBM template in MNI space. In the next step we normalize the Getting Started Guide anatomical image to MNI space via Volumes → Normalize to MNI Template Space using the standard settings.

⁴<https://docs.scipy.org/doc/scipy/reference/generated/scipy.spatial.distance.dice.html>

After normalizing the anatomical image we run the script again, with both images in MNI space. The similarity between the MNI template and the normalised Getting Started Guide image is printed to the BrainVoyager Log tab (see Figure 11).



```
Python Development - image_registration_evaluation_with_dice_coefficient_bv22_v02.py
# get size (= number of voxels): take sum of all marked voxels without background
61 sizeVMR1 = docvnr.dim_x * docvnr.dim_y * docvnr.dim_z
62 sizeVMR2 = docvnr2.dim_x * docvnr2.dim_y * docvnr2.dim_z
63
64 if (sizeVMR1 != sizeVMR2):
65     brainvoyager.print_to_log("Images should have similar size")
66 else:
67     sizeMarkedVMR1 = np.sum(t1Data1D)
68     brainvoyager.print_to_log("Size anatomical data: " + str(sizeVMR1));
69     brainvoyager.print_to_log("Marked voxels: " + str(sizeMarkedVMR1));
70     sizeMarkedVMR2 = np.sum(t1Data1D2)
71     brainvoyager.print_to_log("Size anatomical data: " + str(sizeVMR2));
72     brainvoyager.print_to_log("Marked voxels: " + str(sizeMarkedVMR2));
73     dice = distance.dice(t1Data1D, t1Data1D2)
74     brainvoyager.print_to_log("Sørensen-Dice coefficient (dissimilarity): " + str(round(dice, 4)))
75     brainvoyager.print_to_log("Similarity: " + str(round((1.0-dice), 4)))
76
77 docvnr.close()
78 docvnr2.close()
79

> New New From Hist Load... Save As... Save Stay on top Guide Close Run
X Script to calculate the Dice similarity coefficient (or Sørensen-Dice coefficient)
Background threshold: 60
First file:
/Volumes/Part1/Data_PluginsTesting/scripts/dice_coefficient/set4_MNI_normalisation/ICBM452-IN-MNI152-SPACE_BRAIN.vmr
Second file:
/Volumes/Part1/Data_PluginsTesting/scripts/dice_coefficient/set4_MNI_normalisation/CG_3DT1MPR_SCRIPT_IHHC_SAG_MNI.vmr
Size anatomical data: 16777216
Marked voxels: 1832090
Size anatomical data: 16777216
Marked voxels: 1494053
Sørensen-Dice coefficient (dissimilarity): 0.1261
Similarity: 0.8739
Log
Python Console Log
```

Figure 11: The similarity between the MNI template and the normalised Getting Started Guide image is printed to the BrainVoyager Log tab

We obtain a similarity of 0.8739 (or dissimilarity of 0.1261), so the overlap between the anatomy and the MNI template increased after normalizing to MNI space.

It is also possible to evaluate the quality of segmentations using the Dice coefficient script. Please note that the similarity calculation requires the sizes of the images to be the same (for example both $256 \times 256 \times 256$ voxels).

Python script for BrainVoyager 22

```
# image_registration_evaluation_with_dice_coefficient_bv22_v02.py
#
# This script evaluates image registration by calculating the overlap using the Sorensen-Dice coefficient
# using SciPy: https://docs.scipy.org/doc/scipy/reference/generated/scipy.spatial.distance.dice.html
# In this first version, we need two VMRs.
#
# Usage: load script in BrainVoyager Python Script Editor, click 'Run'. The threshold to remove
# background from brain will be asked via an input dialog with default 100.
# Two file dialogs will appear to select two *.vmr files. Files will be closed when script is finished.
# The Dice similarity coefficient will be written to the BrainVoyager Log tab.
#
# https://en.wikipedia.org/wiki/S%C3%B8rensen%E2%80%93Dice_coefficient
# Please note that the Dice similarity coefficient has many alternatives which might or might not be more appropriate.
#
# HB 030321, 090321, 190321

import numpy as np
import PythonQt.Qt as qt
from scipy.spatial import distance
import bva

def createInputDialog():
    dlg = qt.QInputDialog()
    return dlg

def getNumber():
    dlg = qt.QInputDialog()
    title = "Threshold"
    label = "Please enter a background-brain threshold"
    default = 100
    minimum = 0
    maximum = 255
    number = dlg.getInt(dlg, title, label, default, minimum, maximum)
    if number != '':
        return number

brainvoyager.print_to_log("\nScript to calculate the Dice similarity coefficient (or Srensen-Dice coefficient)")

thresholdVMR = getNumber()
brainvoyager.print_to_log('Background threshold: ' + str(thresholdVMR))

# access data
vmrfilename = brainvoyager.choose_file("Please select the first VMR file", "*.vmr")
docvmr = brainvoyager.open_document(vmrfilename)
t1Data1D = bva.vmrarray(docvmr)
brainvoyager.print_to_log('First file:\n' + vmrfilename)

vmrfilename2 = brainvoyager.choose_file("Please select the second VMR file", "*.vmr")
docvmr2 = brainvoyager.open_document(vmrfilename2)
t1Data1D2 = bva.vmrarray(docvmr2)
brainvoyager.print_to_log('Second file:\n' + vmrfilename2)

# segment into binary images
marked = 1 # default BV segmentation colour is 240, but we need a logical vector with only 0s and 1s
t1Data1D[t1Data1D <= thresholdVMR] = 0 # first this one, otherwise everything will be black
t1Data1D[t1Data1D > thresholdVMR] = marked # segment
t1Data1D2[t1Data1D2 <= thresholdVMR] = 0
t1Data1D2[t1Data1D2 > thresholdVMR] = marked

# get size (= number of voxels): take sum of all marked voxels without background
sizeVMR1 = docvmr.dim_x * docvmr.dim_y * docvmr.dim_z
sizeVMR2 = docvmr2.dim_x * docvmr2.dim_y * docvmr2.dim_z

if (sizeVMR1 != sizeVMR2):
    brainvoyager.print_to_log("Images should have similar size")
else:
    sizeMarkedVMR1 = np.sum(t1Data1D)
    brainvoyager.print_to_log("Size anatomical data: " + str(sizeVMR1));
    brainvoyager.print_to_log("Marked voxels: " + str(sizeMarkedVMR1));
    sizeMarkedVMR2 = np.sum(t1Data1D2)
    brainvoyager.print_to_log("Size anatomical data: " + str(sizeVMR2));
    brainvoyager.print_to_log("Marked voxels: " + str(sizeMarkedVMR2));
    dice = distance.dice(t1Data1D, t1Data1D2)
    brainvoyager.print_to_log("Srensen-Dice coefficient: " + str(round(dice, 4)))
    brainvoyager.print_to_log("Similarity: " + str(round((1.0-dice), 4)))
```

```
docvmr.close()  
docvmr2.close()
```