

# Entregable Fase 3

## Segmentación de materia blanca en imágenes de Resonancia Magnética del cerebro. ITK-SNAP

Antonio Delgado Bejarano - 22 de mayo de 2019

### Introducción

En este último documento vamos a explicar como hemos validado el algoritmo implementado. Para ello hemos desarrollado un pequeño script en MATLAB que nos permite obtener una serie de medidas para posteriormente calcular una serie de índices que nos permitirán valorar objetivamente el rendimiento de nuestro algoritmo.

### Validación en MATLAB

El script comienza convirtiendo la imagen segmentada a formato binario (estaba en formato entero). Hay que precisar el corte de partida común pues la verdad de referencia cuenta con muchos menos cortes que la imagen segmentada, por lo que es muy importante que los cortes que se van a comparar sean los mismos en los dos casos.

Para realizar las medidas se procede primero a calcular cuatro valores: Los verdaderos positivos, los falsos positivos, los verdaderos positivos y los verdaderos negativos. Para ello construimos primeramente una máscara restando la imagen segmentada con la verdad de referencia, de forma que los píxeles que se quedaran a 1 (estaban a 1 en la imagen segmentada y a 0 en la verdad de referencia) constituirían los falsos positivos y los píxeles que tomaban valor -1 (estaban a 0 en la imagen de la segmentación y a 1 en la verdad de referencia) constituirían los falsos negativos. El inconveniente de este método es que no consigue discriminar los verdaderos positivos de los verdaderos negativos (en el caso de restar 0 - 0 y 1 - 1 obtenemos el mismo resultado). Para solucionar esto recurrimos a una segunda máscara formada por la suma de la imagen segmentada y la verdad de referencia, de forma que los píxeles que se quedaban a 0 (suman un 0 en cada imagen) constituyen los verdaderos negativos, y los píxeles que tomaran valor 2 (suman un 1 en cada imagen) forman los verdaderos positivos).

Mediante un bucle calculamos estos cuatro valores para todos los cortes, de forma que al final se acumulaba el valor total de FP, FN, TP y TN para todo el volumen.

```

im=logical(X);
mask=zeros(size(GROUNDTRUTH,3));
FP=0;
FN=0;
TP=0;
TN=0;

for i=1:size(GROUNDTRUTH,3)
    mask = im(:,:,:i+92) - GROUNDTRUTH(:,:,:i);
    mask2 = im(:,:,:i+92) + GROUNDTRUTH(:,:,:i);

    FP=FP+length(find(mask==1));
    FN=FN+length(find(mask==0));
    TP=TP+length(find(mask2==2));
    TN=TN+length(find(mask2==0));
end

```

Fig. 1.- Cálculo de FP, FN, TP y TN.

Aclarar que se añade 92 en el índice que selecciona el corte concreto para que coincida el corte de nuestra segmentación con la verdad de referencia por lo que hemos explicado al principio (para cada imagen hubo que buscar el valor concreto para hacerlos coincidir).

Una vez tenemos los valores de FP, FN, TP y TN para todos los cortes del volumen calcularemos los siguientes parámetros:

- Sensibilidad:  $TP/(TP+FN)$   Porcentaje de pacientes con la enfermedad que reciben un resultado positivo del algoritmo.
- Especificidad:  $TN/(TN+FP)$   Porcentaje de pacientes sin la enfermedad que reciben un resultado negativo del algoritmo.
- PPV:  $TP/(TP+FP)$   Probabilidad de que un paciente tenga la enfermedad si el algoritmo dice que la tiene.
- NPV:  $TN/(TN+FN)$   Probabilidad de que un paciente no tenga la enfermedad si el algoritmo dice que no la tiene.

También se han calculado los índices Jaccard y Dice que mide la similitud entre dos imágenes. El cálculo de estos índices ya estaba implementado en MATLAB por lo que se ha recurrido a las funciones existentes.

Los resultados obtenidos para las diferentes segmentaciones fueron los siguientes:

	Imagen 1	Imagen 2	Imagen 3	Imagen 4	Imagen 5
Sensitividad	0.9703	0.9289	$\frac{0.9058}{0.6671}$	0.9940	0.8972
Especificidad	0.9853	0.9996	$\frac{0.9869}{0.9791}$	0.9848	0.9996
PPV	0.8892	0.9968	$\frac{0.9001}{0.8089}$	0.9022	0.9968
NPV	0.9963	0.9907	$\frac{0.9848}{0.9569}$	0.9991	0.9865
Jaccard	0.8656	0.9261	$\frac{0.8056}{0.5763}$	0.8973	0.8946
Dice	0.9280	0.9617	$\frac{0.8929}{0.7312}$	0.9459	0.9444

*Fig. 2 - Resultados obtenidos*

Aclarar que en la imagen 3 tenía una imagen potenciada en T1 (valor superior de los dos en la tabla) y otra potenciada en T2 (valor inferior). En general todos los índices son muy cercanos a 1, lo cual indica que el algoritmo trabaja muy bien. La imagen potenciada en T2 es un caso algo diferente pues tuve más problemas para segmentarla y no me quedó del todo bien, lo cual se ve reflejado al ser la que peores índices consigue.

Con estos resultados podemos decir que el algoritmo en general funciona bastante bien para el caso a estudiar.