



UNIVERSIDAD DE VALLADOLID

ESCUELA TÉCNICA SUPERIOR DE INGENIEROS DE TELECOMUNICACIÓN

TRABAJO FIN DE GRADO

GRADO EN TECNOLOGÍAS DE LA TELECOMUNICACIÓN

**IDENTIFICACIÓN BIOMÉTRICA CON
RESONANCIA MAGNÉTICA DE DIFUSIÓN:
ANÁLISIS DE LA INDIVIDUALIDAD DE LA
CONECTIVIDAD CEREBRAL**

Autor:

D. Daniel Bueno Pacheco

Tutor:

Dr. D. Rodrigo De Luis García

Valladolid, Julio de 2017

TÍTULO: IDENTIFICACIÓN BIOMÉTRICA CON RESONANCIA
MAGNÉTICA DE DIFUSIÓN: ANÁLISIS DE LA
INDIVIDUALIDAD DE LA CONECTIVIDAD
CEREBRAL

AUTOR: D. Daniel Bueno Pacheco

TUTOR: Dr. D. Rodrigo De Luis García

DEPARTAMENTO: Teoría de la Señal y Comunicaciones e Ingeniería
Telemática

TRIBUNAL

PRESIDENTE: Dr. D. Carlos Alberola López

VOCAL: Dr. D. Santiago Aja Fernández

SECRETARIO: Dr. D. Rodrigo De Luis García

FECHA: Julio de 2017

CALIFICACIÓN:

Resumen

Hoy en día la identificación mediante características fisiológicas del ser humano está muy avanzada. Este tipo de identificación, también llamado identificación biométrica, permite distinguir a las personas mediante atributos que son únicos en cada uno. Muchas de esas características son muy conocidas, entre las que destacamos la huella dactilar, la cara o el iris del ojo humano. En este trabajo se plantea el objetivo de utilizar resonancia de difusión desde un punto de vista de identificación biométrica. En particular, se emplearán conectomas obtenidos a partir de esta modalidad de resonancia. Un conectoma es un mapa de conexiones neuronales del cerebro. El trabajo pretende por lo tanto responder a la pregunta de si el conectoma de un individuo es único como lo son, por ejemplo, sus huellas dactilares. Tras desarrollar e implementar un flujo de procesado a partir de las adquisiciones de resonancia originales para obtener las matrices de conectividad, se ha diseñado también un método de identificación a partir de dichas matrices. Los experimentos realizados sobre un conjunto de datos procedente de más de 200 sujetos han proporcionado tasas de acierto muy elevadas, cercanas en algunos casos al 100%.

Palabras clave

DTI, MRI, difusión, conectómica, conectoma, BIS, tractografía, verificación, sustancia blanca.

Abstract

Nowadays identification by using physiological features of the human being has become very common, and the associated technology is very advanced. This kind of identification, also called biometric identification, allows to tell apart people through attributes that are unique in each person. Many of these features are well known, among which we can highlight the fingerprints, the face or the iris of the human eye. In this

work, we plan to use diffusion magnetic resonance imaging from the point of view of biometric identification. We used connectomes obtained from this kind of resonance, being a connectome a map of the neuronal connections of the brain. Thus, the work seeks to answer the question of whether the connectome of a person is as unique as, for example, his fingerprints are. After developing and working out a processing work flow from the original magnetic resonance acquisitions to obtain the connectivity matrices, we have designed a method for performing identification from such matrices. Experiments performed on a data set from more than 200 subjects have provided high accuracy rates, in some cases close to 100%.

Keywords

DTI, MRI, diffusion, connectomics, connectome, BIS, tractography, verification, white matter.

Agradecimientos

En primer lugar me gustaría agradecer a mis padres y a mi hermano su apoyo durante los 4 años de carrera, sin ellos no hubiese conseguido todo los logros que tengo ahora.

En segundo lugar, quisiera agradecer a todos los profesores que me han transmitido sus conocimientos, con el fin de mejorar como persona y formarme académicamente para llegar a donde estoy ahora. Entre estos profesores quisiera agradecer personalmente a César Gutiérrez Vaquero que me aconsejó y me dió ideas sobre mi Trabajo de Fin de Grado, y especialmente a mi tutor, Rodrigo, el cuál, me ha brindado su apoyo y guía a lo largo de este estudio que me permitió realizar, así como su comprensión y paciencia que ha tenido conmigo.

En último lugar quisiera agradecer a amigos y voluntarios que me han ayudado a la realización de mi trabajo, bien con su apoyo y ofrecimiento para completar este estudio satisfactoriamente, sobre todo a Leticia Bueno y a Alicia De Porras las cuales me han ayudado más de lo que ellas podrían imaginar. También a Sergio Serrano el cual me proporcionó la plantilla para escribir detalladamente mi trabajo.
A todos ellos, muchas gracias.

Índice

Índice de figuras	11
Índice de tablas	13
Capítulo 1. Introducción.....	15
1.1. Motivación.....	15
1.2. Objetivos	17
1.3. Medios.....	17
1.4. Fases.....	19
1.5. Organización de la memoria.....	19
Capítulo 2. Resonancia de difusión en el cerebro	21
2.1. Introducción.....	21
2.2. Principio físico de la difusión	21
2.3. Difusión isótropa y difusión anisótropa	22
2.4. Medición de la difusión con MRI	23
2.5. Modelos de difusión anisótropa.....	24
2.5.1. DTI	24
2.5.2. HARDI	26
2.6. Tractografía	26
2.7. Conectómica	27
Capítulo 3. Identificación biométrica	29
3.1. Introducción.....	29
3.2. Identificación vs Verificación	31
3.3. Identificación biométrica como problema de reconocimiento de patrones	31
3.4. Identificación biométrica mediante resonancia de difusión	32
Capítulo 4. Procesado para la obtención de matrices de conectividad	35
4.1. Medios de partida.....	35
4.1.1. Datos	35

4.1.2.	Software.....	35
4.1.3.	Estructura de directorios.....	37
4.2.	Estructura del procesado	37
4.3.	Preprocesado.....	38
4.4.	Procesado de T1.....	39
4.4.1.	Conversión de formato y volumen de la imagen T1	39
4.4.2.	Eliminación de materia no cerebral en el volumen T1	41
4.4.3.	Segmentación y parcelación	42
4.5.	Procesado de DWI	45
4.5.1.	Conversión de formato de la imagen de difusión DWI.....	45
4.5.2.	Obtención de una máscara del cerebro a partir de la imagen de difusión DWI	46
4.5.3.	Obtención de un mapa de diferentes tejidos cerebrales	48
4.5.4.	Estimación de la "Response Function" y de la "Fibre Orientation Distributions"	49
4.5.5.	Obtención de tensores y medidas escalares asociadas	52
4.5.6.	Adaptar la segmentación cortical para calcular la conectómica....	54
4.5.7.	Registrado para proyectar la segmentación cortical sobre el volumen tensorial.	56
4.5.8.	Realización de una tractografía completa del cerebro	60
4.5.9.	Obtención de la matriz de conectividad	63
Capítulo 5.	Método de identificación biométrica	67
5.1.	Introducción.....	67
5.2.	Datos de partida	67
5.3.	Método propuesto	67
Capítulo 6.	Trabajo experimental	71
6.1.	Introducción.....	71
6.2.	Conjunto de datos y esquema de validación	71
6.3.	Resultados	75
6.4.	Discusión.....	77
Capítulo 7.	Conclusiones y trabajo futuro.....	79
7.1.	Conclusiones.....	79
7.2.	Líneas de trabajo futuro	80
Referencias		83
Apéndice A.	Scripts	87
A.1.	Script 1: funcion_global_identificacion_conectomicaV3	87
A.2.	Script 2: normaliza_o_estandariza.....	98
A.3.	Script 3: compara_indices_iguales_encontrados.....	101

A.4. Script 4: calcula_umbral_optimo 102

Índice de figuras

Figura 3-1 Modelo general de un sistema biométrico	30
Figura 4-1: Visualización de una imagen anatómica T1 mediante MRtrix	40
Figura 4-2: Visualización de la extracción de materia no cerebral de la imagen anatómica T1 mediante MRtrix.....	42
Figura 4-3: Segmentación cortical realizada sobre la imagen anatómica T1.....	44
Figura 4-4: Imagen hecha por una resonancia magnética de difusión en formato .nrrd.	46
Figura 4-5: Contraposición de la máscara generada con la imagen de difusión mediante MRtrix.....	47
Figura 4-6: Visualización de una imagen 3D obtenida de una imagen 5TT.....	49
Figura 4-7: Visualización de la función de respuesta estimada.....	51
Figura 4-8: Visualización de la FOD en superposición con la imagen de difusión.	52
Figura 4-9: Visualización con MRtrix de la FA en cada región de la imagen de difusión	54
Figura 4-10: Adaptación de la segmentación cortical mediante un cambio de su 'tabla de búsqueda'	56
Figura 4-11: Visualización superpuesta de la FA(negro y gris) con la segmentación cortical(en colores rojo y naranja) producida por Freesurfer mediante mrview	59
Figura 4-12: Visualización superpuesta de la FA(negro y gris) con la segmentación cortical(en colores rojo y naranja) en formato 5TT mediante mrview.	60
Figura 4-13: Visualización de una tractografía superpuesta con la imagen de difusión mediante mrview.....	63
Figura 4-14: Visualización del conectoma en contraposición con la segmentación cortical obtenida con la parcelación del apartado 3.5.6 mediante mrview	64
Figura 6-1: Visualización de una matriz de conectómica basado en el número de fibras entre regiones.....	72
Figura 6-2: Visualización de una matriz de conectómica basado en la FA media entre regiones	72
Figura 6-3: Visualización de los parámetros a variar antes de ejecutar el script.....	73
Figura 6-4: Gráfica del número de conexiones válidas en función del valor del umbral.	74
Figura 6-5: Gráfica de la representación de las distancias entre casos, así como el umbral y el porcentaje de comparaciones.	76

Índice de tablas

Tabla 6.1: Extracto de la tabla general de casos empleados	72
Tabla 6.2 Tasa de acierto para cada iteración.....	75

Capítulo 1. Introducción

1.1. Motivación

En el mundo moderno, la tecnología y la medicina han evolucionado considerablemente de tal manera que hoy en día podemos [2][3] mapear el sistema nervioso de un organismo mediante una serie de conexiones de una manera eficiente y bastante precisa. Estos mapas de conexiones en el sistema nervioso se llaman conectomas y la rama que estudia estos conectomas es la conectómica.

Para obtener estos mapas de conexiones, hemos utilizado la resonancia magnética de difusión. Es un método de resonancia magnética que utiliza la difusión de las moléculas de agua para generar contraste en las imágenes de la resonancia magnética de sujetos in vivo y ex vivo, gracias a que existen tejidos biológicos sensibles a las características locales de la difusión molecular. La difusión molecular en los tejidos no es libre, pero refleja interacciones con muchos obstáculos, como macromoléculas, fibras y membranas. Por lo tanto, los patrones de difusión de moléculas de agua pueden revelar detalles microscópicos sobre la arquitectura del tejido, ya sea normal o en un estado alterado o patológico. Los movimientos de las moléculas de agua dentro de los tejidos pueden ser magnéticamente etiquetados usando pulsos de gradiente de campos magnéticos bipolares, haciendo que los núcleos de hidrógeno del agua devuelvan unas señales que dependen del tipo de tejido y de esa difusión de las moléculas de agua. Nosotros vamos a usar la resonancia de difusión en el cerebro, lo cual nos va a permitir el estudio de la conectividad de las fibras nerviosas en él.

Existen dos tipos de difusión molecular, que son la difusión isotrópica y anisotrópica[4][3]. En el caso de la difusión isotrópica, la movilidad de las moléculas desde el origen es aproximadamente igual en todas las direcciones del espacio, debido principalmente a que el agua se ha difundido en un medio sin barreras. Si el agua se difunde en un medio con barreras, la difusión no será igual en todas las direcciones del espacio, la cual llamaremos difusión anisotrópica. La difusión anisotropa en la materia blanca se utiliza para estimar la organización de fibras cerebrales. Esto es debido a que las moléculas de agua se difunden más fácilmente en la dirección de las fibras en vez de perpendicularmente porque en la sustancia blanca del cerebro la mielina de los tractos nerviosos actúa como barrera, de forma que dificultan el movimiento a lo largo de ellas. Rápidamente se descubrió que esta característica podría ser usada para mapear la orientación en el espacio de los tractos nerviosos de la sustancia blanca y asumiendo que la dirección de mayor difusión indicaría la total orientación de las fibras. Como

veremos, hemos utilizado dos modelos de difusión anisótropa, ya que en el cerebro predomina la difusión anisótropa. Uno de ellos es la imagen por tensores de difusión (DTI), la cual se utiliza ampliamente para asignar la tractografía de la materia blanca en el cerebro. Para [1] caracterizar la difusión en medios anisótropos se emplean los tensores de difusión, de los cuales se derivan medidas escalares como la FA ya que los tensores son entidades matemáticas algo complejas.

La tractografía[2] es un procedimiento que se usa para poner de manifiesto los tractos neuronales, representándolos mediante líneas en el espacio tridimensional en el que se encuentra la adquisición de resonancia. Es decir, podemos estimar las trayectorias de los haces de fibras en el cerebro y representarlas. Gracias a la tractografía y una segmentación de la corteza cerebral en diferentes regiones de interés podemos realizar un estudio de la conectómica. La conectómica obtiene medidas de conectividad entre las distintas regiones del cerebro, entre las cuales destacan el número de fibras entre dos regiones o la FA media. A partir de los datos resultantes se puede crear una matriz de conectividad que las represente, dependiendo de una serie de parámetros que las caractericen, obteniendo una matriz diferente por cada característica.

Tras obtener las matrices de conectómica mediante un método propio queremos ver si somos capaces de identificar personas biométricamente con el fin de estudiar qué es lo que hace único al conectoma de una persona, si es que es único. A la hora de identificar biométricamente a una persona, se tienen en cuenta una serie de aspectos como la seguridad, facilidad de uso, aceptación del usuario, coste, etc que influyen a la hora de elegir el tipo de identificación. También es importante distinguir entre los modos de identificación y verificación, habiendo utilizado nosotros el segundo de ellos. En verificación, uno tiene una comparación de dos casos, y tiene que decidir si corresponden al mismo sujeto. La respuesta es simplemente sí o no. Hay que considerar dos tipos de errores: decir que son el mismo cuando son distintos, y decir que son distintos cuando son iguales. En identificación pura, la situación es que un sujeto proporciona su huella dactilar, por ejemplo, y el sistema tiene que averiguar quién es ese sujeto comparándola con su base de datos, o bien decidir que la huella no pertenece a ningún sujeto incluido en la base de datos. En este caso, lo que se hace es buscar, de entre la base de datos, cuál es el caso con el que la distancia es mínima, y elegir ese caso como pareja si supera un umbral. Si no se supera, se decide que no hay pareja. Nosotros hemos implementado el modo de verificación, pero no el de identificación. Además un sistema de identificación biométrica (BIS) también puede estudiarse como un sistema de reconocimiento de patrones en los que para poder diseñarlo se describen una serie de pasos generales entre los que destacan la adquisición de datos, preprocesado, obtención del vector de características, selección o extracción y la clasificación de este vector entre una serie de clases.

Mediante un método propio vamos a obtener matrices de conectómica partiendo de una serie de datos de resonancia. Gracias a una serie de programas entre los que destacan FreeSurfer, MRtrix, FSL, Slicer o Matlab vamos a realizar un procesado utilizando diversas herramientas de estos programas. Matlab lo utilizaremos principalmente para obtener una serie de scripts que nos permitirán realizar el procesado de varios casos en paralelo y ahorrarnos tiempo de procesamiento.

Utilizando un método de identificación biométrica exclusivo para nuestro estudio, creamos un algoritmo en Matlab para poder obtener el principal objetivo de

nuestro trabajo. Este método se caracteriza principalmente por estar diseñado para un estudio que disponga pocos datos de difusión, además de diseñar un esquema de "majority voting" que consiste en ir calculando las distancias entre casos comparando uno a uno los elementos del vector de características con otro vector, con el fin de mejorar nuestros resultados. Para calcular objetivamente el umbral, hemos seguido un procedimiento en el que cuando el sistema está en su fase de entrenamiento, tenemos una distribución de distancias para comparaciones entre el mismo sujeto, y otra distribución de distancias para comparaciones entre sujetos distintos. Podemos aproximar ambas distribuciones por Gaussianas, y calcular teóricamente cuál es el umbral óptimo para separar esas dos distribuciones.

Este método nos da una serie de resultados satisfactorios con una tasa de acierto a la hora de identificar individuos de casi el 100%, pero también se destaca que existen una serie de factores a tener en cuenta en este estudio que podrían servir para mejorar la tasa de resultados a la hora de identificar biométricamente.

1.2. Objetivos

El objetivo principal de este trabajo consiste en realizar un estudio, mediante la implementación de un sistema de identificación biométrica sobre las matrices de conectividad obtenidas con resonancia de difusión, acerca de la individualidad de la conectividad cerebral entre personas. El propósito de dicho análisis no es en principio avanzar hacia un sistema de identificación biométrica de uso práctico basado en resonancia de difusión (lo cual no parece tener mucho sentido), sino estudiar en qué medida el conectoma humano es único en cada persona.

El objetivo general descrito puede dividirse en varios objetivos particulares:

- Poner a punto un flujo de procesamiento de datos capaz de obtener matrices de conectividad a partir de las adquisiciones de resonancia anatómica y de difusión.
- Obtener una base de datos formada por resonancias de difusión (y anatómicas) de un conjunto suficiente de sujetos, incluyendo sujetos con varias adquisiciones.
- Diseñar e implementar un sistema de identificación biométrica basado en resonancia de difusión.
- Diseñar un esquema experimental y obtener resultados acerca del funcionamiento del sistema implementado.

1.3. Medios

Para la realización del estudio se dispone de datos de resonancia de difusión correspondientes a 226 sujetos. Dichos datos fueron adquiridos en una unidad de resonancia magnética Philips Achieva 3 Tesla, propiedad de la Universidad de Valladolid. El protocolo de adquisición empleado incluye una adquisición de tipo anatómico, ponderada en T1, y una adquisición de difusión.

Los parámetros de la primera secuencia incluyen: secuencia Turbo Field Echo (TFE), matriz de tamaño 256x256, resolución espacial de 1x1x1 mm³ y 160 cortes sagitales cubriendo todo el cerebro.

Los parámetros de la adquisición de difusión incluyen: 61 direcciones de gradiente más un volumen "baseline" sin ponderación en difusión, b-value=1000 s/mm², matriz de tamaño 128x128, resolución espacial de 2x2x2 mm³ y 66 cortes axiales cubriendo todo el cerebro.

La duración total de la adquisición es de 19 minutos.

La base de datos empleada está formada por sujetos pertenecientes a estudios pasados desarrollados en el Laboratorio de Procesado de Imagen (LPI), incluyendo voluntarios sanos y también pacientes de esquizofrenia y cefaleas. Además, se han realizado, expresamente para la realización de este trabajo, adquisiciones dobles sobre el mismo sujeto, de forma que sea posible entrenar y evaluar el método de identificación.

Entre las herramientas software que hemos utilizado para llevar a cabo el desarrollo del trabajo se encuentran:

- Matlab: Este programa genera varios scripts con todos los comandos que van ejecutando los distintos pasos del procesado para obtener las matrices. Además creamos un algoritmo en el lenguaje de programación de Matlab con el fin de cumplir la función final de este Trabajo de Fin de Grado (método de identificación biométrica)
- Slicer versión 4.5: En nuestro análisis de la conectividad, hemos utilizado principalmente este programa para la visualización de imágenes de resonancia y para la conversión de formatos de estas con el fin de poderlas visualizar de una manera más detallada.
- FSL: Es un biblioteca de herramientas con las cuales hemos realizado las siguientes funciones
 - Extracción de materia no cerebral de la imagen anatómica.
 - Obtención de tensores y medidas escalares asociadas.
 - Registrado para proyectar la segmentación cortical sobre el volumen tensorial, que es el espacio donde esta calculada la tractografía.
- MRtrix: Hemos utilizado este programa para las siguientes partes del procesado:
 - Obtención de una máscara del cerebro.
 - Obtención de una segmentación de los diferentes tipos de tejido cerebral en formato 5TT.
 - Estimación de la "response function"(RF) y de la "fibre orientation distribution"(FOD).
 - Realización de la tractografía completa del cerebro.
 - Adaptar la segmentación cortical generada por Freesurfer para calcular la conectómica.
 - Obtención de las matrices de conectividad.
- Freesurfer: Es un software que hemos utilizado para las siguientes partes del procesado:
 - Generación de una segmentación cortical.
 - Conversión del tamaño del volumen de la imagen anatómica T1 a un tamaño 256x256x256 usando la herramienta mri_convert.
 - Conversión de la segmentación cortical al formato .nii para su posterior procesado.
 - Parcelación y etiquetado de las regiones corticales y subcorticales.
- MRICConvert: Es un conversor de ficheros de imágenes médicas que hemos utilizado para convertir del formato DICOM de la imagen T1 a un formato .nii utilizando la herramienta mconvert.

El hardware utilizado para realizar las distintas partes del procesado, han sido dos servidores del Laboratorio de Procesado de Imagen(LPI) la Universidad de Valladolid: tanis.lpi.tel.uva y tebas.lpi.tel.uva.es que trabajan con un sistema operativo Ubuntu 16.04.2. A ambos servidores del Laboratorio nos hemos conectado de manera remota utilizando la herramienta ssh desde el mismo laboratorio con el ordenador ceres.lpi.tel.uva.es que utiliza un sistema operativo Linux.

1.4. Fases

Este trabajo se puede dividir en varias fases. En primer lugar, un estudio acerca del funcionamiento de la resonancia de difusión y una descripción de dos modelos de difusión, así como una serie de procedimientos generales para la obtención de matrices de conectómica.

En segundo lugar, se ha recolectado y estudiado una documentación acerca de la identificación biométrica, destacando el funcionamiento general, así como factores a tener en cuenta a la hora de diseñar un sistema de identificación.

En tercer lugar, se trata del procesado realizado para la obtención de matrices de conectividad mediante una serie de herramientas. Se divide principalmente en un procesado de la imagen anatómica y un procesado de la imagen de difusión en el que resalta la obtención de un mapa de conexiones neuronales y con él la obtención de matrices de conectómica.

En cuarto lugar, el diseño e implementación del método de identificación biométrica usado, seguido del trabajo experimental realizado sobre el cual mostramos los resultados obtenidos, así como una discusión acerca de las implicaciones y limitaciones del estudio.

1.5. Organización de la memoria

En la presente memoria se detallan y resumen los pasos seguidos de la siguiente manera:

En el Capítulo 2 se explican los fundamentos de la resonancia de difusión en el cerebro(2.1), así como el principio físico de la difusión(2.2), la distinción entre difusión anisótropa e isotropa(2.3), cómo se mide la difusión mediante MRI(2.4), los modelos de difusión anisótropa(2.5), y los conceptos de tractografía(2.6) y conectómica(2.7).

En el Capítulo 3 detallamos los fundamentos de la identificación biométrica(3.1), así como una distinción entre identificación y verificación(3.2), la descripción de una serie de características generales de un sistema de identificación biométrica como un sistema de reconocimiento de patrones(3.3) y aspectos de la identificación biométrica mediante resonancia de difusión(3.4).

En el Capítulo 4 y el Capítulo 5 describimos el método propuesto, detallando los pasos seguidos para el procesado llevado para la obtención de matrices de conectividad y el método de identificación biométrica usado respectivamente.

El Capítulo 6 consiste en describir el trabajo experimental realizado, así como una descripción de sus resultados y una discusión acerca de una serie de parámetros que se pueden tener en cuenta.

En el Capítulo 7 se recogen las conclusiones del estudio realizado, así como una serie de líneas de trabajo futuro propuestas.

Por último, en el Apéndice A se muestra una descripción del código de los scripts empleados para realizar el análisis de identificación.

Capítulo 2. Resonancia de difusión en el cerebro

2.1. Introducción

La resonancia de difusión[1] [2] [3] es una técnica in vivo de resonancia magnética que introduce un método distinto al que utiliza la resonancia magnética convencional, de forma que es sensible a los movimientos microscópicos de las moléculas de agua dentro de los tejidos. Una de las principales características de este método es que la compleja organización estructural de la materia blanca del cerebro puede ser representada en gran detalle con los diferentes métodos que abarca la resonancia de difusión, lo cual nos servirá en nuestro método propuesto más adelante.

El funcionamiento principal de la resonancia de difusión está basado en señales obtenidas de los núcleos de hidrógeno(protones). Hay muchas moléculas que contienen protones de hidrógeno en nuestro cuerpo, pero a partir de ahora hablaremos de que la señal obtenida por la resonancia proviene de los protones del agua, que es muy abundante en los tejidos biológicos. Los movimientos de las moléculas de agua dentro de los tejidos pueden ser etiquetados magnéticamente usando pulsos de gradiente de campos magnéticos bipolares, lo cual crea imágenes sensibles a la difusión en determinadas direcciones llamadas DWIs(Diffusion Weighted Images) con el fin de generar un contraste en estas imágenes.

La difusión molecular se refiere al movimiento aleatorio de estas moléculas(de agua), también llamado movimiento Browniano, que es una consecuencia de la energía térmica que contienen estas moléculas. El éxito de esta técnica se basa principalmente en el concepto de que durante el movimiento aleatorio, las moléculas de desplazamiento impulsadas por la difusión permiten obtener información de la estructura del tejido a una escala microscópica más allá de la resolución de imagen habitual: durante tiempos de medición de difusión de unos 50 milisegundos, las moléculas de agua se mueven por el cerebro una distancia media de unos 10 μm , balanceándose, cruzando, o interactuando con componentes de tejidos como fibras, macromolécula y membranas.

También es necesario destacar que la difusión es un proceso de tres dimensiones. Por lo tanto, la movilidad de las moléculas puede no ser la misma en todas las direcciones. Esta anisotropía(apartado 2.3) puede ser debida a una disposición física del medio o de la presencia de obstáculos que limiten los movimientos moleculares en algunas direcciones. Solo los desplazamientos moleculares a lo largo de las direcciones del gradiente son visibles, ya que los efectos de la difusión anisótropa pueden ser detectados mediante la observación de las variaciones en las medidas de la difusión cuando las direcciones de los pulsos del grandiente son cambiados.

2.2. Principio físico de la difusión

La difusión[4] es un proceso por el cual la materia es transportada de una parte de un sistema a otro debido a movimientos térmicos moleculares o movimientos browniano.

La naturaleza de los dos procesos fue descubierta por Adolf Fick en 1855, el cual describió la difusión de manera cuantitativa adoptando la ecuación matemática de la conducción del calor derivada de Fourier. Estos movimientos vienen descritos por la primera ley de Fick (2.1), que declara que las diferencias locales en una concentración de soluto originará un flujo neto de moléculas de soluto desde las zonas de mayor concentración a las de concentración más baja.

$$J = -D \frac{\partial \Phi}{\partial x} \quad (2.1)$$

En la ecuación 2.1, J es el flujo difusivo, es decir, mide la cantidad de sustancia que fluye a través de una unidad de área durante un intervalo de una unidad de tiempo, D es el coeficiente de difusión o difusividad, el cual representa el movimiento térmico de moléculas de agua [3], Φ es la concentración [4], medida en cantidad de sustancia por unidad de volumen y x es la posición, dada en dimensiones de longitud.

Para dos o más dimensiones, se sustituye la siguiente derivada $\frac{\partial \Phi}{\partial x}$ por el gradiente $\nabla \Phi$, el cual es el gradiente de concentración.

D es proporcional a la velocidad de difusión al cuadrado de las partículas que están difundiendo, que a su vez depende de la temperatura, viscosidad del fluido y del tamaño de las partículas de acuerdo a la relación de Einstein-Stokes. Aunque como veremos en el apartado 2.4 este factor varía de otra forma distinta, ya que para describir la difusión anisótropa es preciso disponer de más parámetros.

2.3. Difusión isótropa y difusión anisótropa

Existen dos tipos de difusión molecular, que son la difusión isométrica y anisométrica [3].

En el caso de la difusión isométrica, la movilidad de las moléculas es aproximadamente igual en todas las direcciones del espacio. En otras ocasiones, sin embargo, la estructura de los tejidos hace que la difusión no se produzca por igual en todas las direcciones, y en este caso hablamos de difusión anisométrica. En el tejido cerebral de la sustancia blanca, la mielina de los tractos nerviosos actúa como una barrera que dificulta la difusión en la dirección perpendicular a dichas fibras [4], de forma que la difusión se produce predominantemente a lo largo de la orientación de las fibras[3].

Si bien para describir la difusión isométrica es suficiente (en principio) con estimar la constante de difusividad, D , en el caso de la difusión anisométrica es necesario utilizar un modelo más complejo, y la descripción de la difusión deberá utilizar herramientas matemáticas más elaboradas.

2.4. Medición de la difusión con MRI

Existen varios tipos de secuencias de resonancia magnética que son sensibles a la difusión de las moléculas de agua. Una de las más sencillas puede construirse a partir de una secuencia Spin Echo, de forma que la señal resultante es [3]:

$$S = PD(1 - e^{-TR/T1}) e^{-TE/T2} e^{-bD} \quad (2.2)$$

La ecuación [2.2](#) se llama también ecuación de Stejskal-Tanner, donde PD es la densidad de protones, $T1$ y $T2$ son los tiempos de relajación del tejido y TR y TE son el tiempo de repetición y tiempo de eco, respectivamente. La ponderación en difusión en esta secuencia viene dada por el parámetro b , conocido habitualmente como b -value. Finalmente, D es la constante de difusión que se pretende estimar (suponemos por ahora el caso isotrópico).

Debido a que en MRI el movimiento molecular generado por los gradientes de concentración no puede ser diferenciado del movimiento molecular debido a otros factores[4], es habitual referirse a la constante de difusión de la ecuación [2.2](#) como el coeficiente aparente de difusión (ADC), que es el coeficiente que verdaderamente se puede medir mediante DWI. En este coeficiente la intensidad de cada píxel es proporcional al grado de difusión y de otros parámetros como la duración del gradiente.

Ahora vamos a adaptar [4] la ecuación [2.2](#) para el caso de tensores de difusión y obtenemos la ecuación [2.4](#). Destacamos que vamos a considerar que una serie de constantes conocidas, pasarán a llamarse S_0 por simplificación. ($S_0 = PD(1 - e^{-TR/T1}) e^{-TE/T2}$ ya que los valores de los que depende los consideramos conocidos)

$$S_k = S_0 e^{-bg_k'Dg_k} \quad (2.4)$$

La ecuación [2.4](#), también llamada ecuación de Stejskal-Tanner para el caso anisótropo contiene los siguientes cambios: $\mathbf{D} = \mathbf{D} * \mathbf{I}$ siendo \mathbf{I} el tensor identidad, g_k es un vector gradiente normalizado, S_0 es la intensidad de la señal en ausencia de un gradiente de difusión ($g_0=0$), S_k es la intensidad medida tras aplicar el k -ésimo gradiente de difusión en la dirección g_k , $g_k'Dg_k$ representa el coeficiente de difusión en la dirección g_k y b es el factor de ponderación de difusión hablado en la ecuación [2.2](#).

Para estimar el tensor, se tienen que hacer al menos seis mediciones además de saber de antemano el parámetro S_0 (ausencia de gradiente). Para obtener las intensidades de cada señal $\{S_1, S_2, \dots, S_6\}$ se aplican seis ponderaciones de difusión y direcciones de gradiente. Resolviendo el sistema de ecuaciones resultante de aplicar la ecuación [2.4](#) generalizada se obtienen los vectores de difusión correspondientes. Este sistema se puede resolver de distintas maneras entre las que destaca la manera más habitual que es por mínimos cuadrados. Ya que las medidas obtenidas suelen estar en superposición con un ruido bastante alto, se suelen obtener más gradientes para estimar mejor el tensor como hemos hecho en la práctica.

2.5. Modelos de difusión anisótropa

2.5.1. DTI

Para caracterizar la difusión en medios anisótropos, donde los desplazamientos de las moléculas no son iguales en todas direcciones, es preciso emplear modelos matemáticos más complejos que el visto en la ecuación 2.2. DTI(Diffusion Tensor Imaging) [3] [4] es una técnica de DWI, que modela la difusión anisótropa como un elipsoide en el espacio, es decir, como un tensor [1] que es una matriz 3x3 simétrica ($D_{ij} = D_{ji}$, con $i, j = x, y, z$) y definida positiva.

$$\mathbf{D} = \begin{pmatrix} D_{xx} & D_{xy} & D_{xz} \\ D_{yx} & D_{yy} & D_{yz} \\ D_{zx} & D_{zy} & D_{zz} \end{pmatrix} \quad (2.3)$$

El coeficiente de difusión [1], \mathbf{D} , en DWI esta descrito mediante un parámetro escalar. Cabe destacar que la difusión en el cerebro es anisótropa (varía en diferentes direcciones), particularmente en la sustancia blanca, por lo que es necesario expresar las ecuaciones en función de vectores y matrices y no de escalares. Por lo que en DTI, este parámetro no se puede estimar mediante un escalar, ya que la difusión varía con la dirección debido a las barreras de mielina en los axones, lo que hace que las moléculas de agua se difundan a lo largo de las fibras. Por lo que para describir \mathbf{D} se requiere un tensor que describa la movilidad de las moléculas a lo largo de cada dirección y la correlación entre estas direcciones.

Para estimar la orientación de las fibras[3] se estiman tres cantidades de difusión a lo largo de los ejes X,Y y Z(estas medidas concuerdan con la forma de un elipsoide) ya que en la realidad las orientaciones de las fibras son casi siempre oblicuas a los ejes. Por lo tanto, nuestro tensor contiene seis parámetros que son tres autovalores λ_1 , λ_2 y λ_3 correspondientes al eje más largo, el mediano y el más pequeño respectivamente y tres autovectores v_1 , v_2 y v_3 que corresponden a la orientación de cada eje de los autovalores. El elipsoide representa la distancia media de difusión, siendo el eje principal del elipsoide correspondiente con la dirección de mayor difusión e indica la orientación del eje de la fibra. Además como es muy difícil visualizar vectores 3D se suele convertir esta información a un espacio de colores y generar un mapa de código de colores.

En muchas ocasiones (como para visualizar o para realizar comparaciones numéricas) resulta conveniente obtener a partir de los tensores de difusión medidas escalares que nos proporcionen información acerca de diferentes aspectos de la forma o tamaño del tensor de difusión[1].

Traza (Trace)

Se calcula como la suma de los elementos diagonales del tensor [4], lo que equivale a la suma de los autovalores:

$$\text{Tr}(\mathbf{D}) = D_{xx} + D_{yy} + D_{zz} = \lambda_1 + \lambda_2 + \lambda_3 \quad (2.5)$$

Difusividad media (MD, Mean Diffusivity)

La MD [1] es una medida que representa la cantidad de difusión en un voxel y se obtiene de la siguiente manera:

$$MD = \frac{Tr(D)}{3} = \frac{\lambda_1 + \lambda_2 + \lambda_3}{3} \quad (2.6)$$

Es preciso destacar que la MD y la traza son realmente la misma medida, excepto un factor de escala.

Anisotropía fraccional (FA, Fractional Anisotropy)

La FA describe el grado de anisotropía en el proceso de difusión. Es un escalar entre 0 y 1, en el que un valor de 0 significa que la difusión es totalmente isotrópica y 1 que la difusión es puramente lineal.

$$FA = \frac{1}{\sqrt{2}} \frac{\sqrt{(\lambda_1 - \lambda_2)^2 + (\lambda_2 - \lambda_3)^2 + (\lambda_1 - \lambda_3)^2}}{\sqrt{\lambda_1^2 + \lambda_2^2 + \lambda_3^2}} = \frac{\sqrt{3}}{\sqrt{2}} \frac{|D - \frac{1}{3}trace(D)I|}{|D|} \quad (2.7)$$

donde **I** es el tensor identidad.

Anisotropía relativa (RA, Relative Anisotropy)

La RA [1] está definida como la desviación estándar de los autovalores dividido por la media de los autovalores y al igual que la FA [4] es una medida de la anisotropía que se puede calcular sin conocer explícitamente los autovalores:

$$RA = \frac{1}{\sqrt{2}} \frac{\sqrt{(\lambda_1 - \lambda_2)^2 + (\lambda_2 - \lambda_3)^2 + (\lambda_1 - \lambda_3)^2}}{\lambda_1 + \lambda_2 + \lambda_3} = \frac{\sqrt{3}}{\sqrt{2}} \frac{|D - \frac{1}{3}trace(D)I|}{trace(D)} \quad (2.8)$$

Mapeado de la orientación de la fibra (Fiber Orientation Mapping)

El último parámetro escalar [1] del que hablaremos será el mapeo de la orientación en el espacio de la estructura de los tejidos. La suposición es que la dirección de las fibras es colineal con la dirección del autovector asociado a la mayor difusividad. Esta aproximación abre una nueva manera de conseguir información directa de sujetos in vivo en la organización de los tejidos orientados como los músculos, el cerebro o la materia blanca espinal. Para mostrar la orientación de la fibra se utiliza mapas de colores y representación de elipsoides.

Como ya hemos hablado, DTI [3] nos proporciona dos nuevos contrastes, que son la difusión anisotrópica y la orientación de la fibra, pero también tiene una importante desventaja: La suposición de que el eje de mayor difusión corresponde a la orientación de la fibra no es verdadera si existen dos poblaciones de fibra. Una orientación no puede representar la orientación de dos diferentes trazas neuronales que viajan en diferentes direcciones que se cruzan entre sí. Además la mayoría de las fibras se cruzan entre sí en la materia blanca del cerebro.

2.5.2. HARDI

El modelo del tensor de difusión, si bien resulta de gran utilidad, tiene también sus limitaciones. La más importante es que no permite modelar adecuadamente situaciones en las cuales existen varias direcciones predominantes dentro de un mismo voxel. Esto sucede particularmente en aquellos lugares del cerebro en los que se producen cruces de fibras.

Para solventar dicho problema, se han propuesto diferentes modelos más avanzados, que reciben habitualmente el nombre común de técnicas HARDI (High Angular Resolution Diffusion Imaging) que engloba varias técnicas. En general las técnicas HARDI[40] consisten en medir la señal de difusión usando un número mucho mayor de direcciones de gradiente de difusión uniformemente distribuidas que las requeridas para DTI, con el fin de capturar la información de difusión completa que no está modelada adecuadamente mediante un solo tensor de difusión.

Una de las técnicas de HARDI es la llamada DSI (Diffusion Spectrum Imaging) [2] que se caracteriza por la adquisición de un gran número de imágenes a las que se aplican fuertes gradientes magnéticos que sensibilizan el movimiento en múltiples direcciones. Esto no sólo resulta en largos tiempos de adquisición sino que también requiere hardware de alto rendimiento, por lo que por esta razón DTI es altamente utilizada actualmente. DSI es mayoritariamente utilizado para realizar tractografías de las fibras.

2.6. Tractografía

La tractografía [2] es un método algorítmico para deducir la trayectorias de los haces de fibras a través de la materia blanca basándose en los datos de la resonancia de difusión.

Partiendo de DTI, por ejemplo, si suponemos que [3] la orientación del eje más largo(v_1) del tensor de difusión representa la orientación de la fibra, podemos reconstruir en 3D las tractos neuronales. Puesto que el tamaño de las fibras es mucho menor que el de los voxels empleados, lo que se obtiene en realidad son aproximaciones de conjuntos de fibras.

Es posible definir dos tipos principales de métodos de tractografía [5] : la tractografía determinista (también llamado *streamlines tractography*) y la tractografía probabilística. La tractografía determinista asume que el principal autovector v_1 es paralelo a la fibra dominante en cada voxel y forma una tangente al espacio curvo trazado por el tracto de materia blanca. La evolución del espacio curvo es realizada por la propagación de una ruta bidireccional desde un punto de origen, el cual es el centro del voxel normalmente. Además da una estimación de los parámetros del modelo, eligiendo la trayectoria más probable. Pero la tractografía determinista se encuentra con una importante desventaja: Las medidas provienen de voxels que son mucho más largos que los axones y cada punto en el cerebro no tiene una conexión sino varias. Como nuestras medidas suelen contener errores y como nuestros modelos nunca pueden ser perfectos, siempre será incierto representar la ruta.

La tractografía probabilística estima las orientación de las fibras con mayor probabilidad, pero también estimamos la distribución, representando cómo de probable es que cada orientación se encuentre a lo largo de una fibra. Luego, cuando trazamos las

conexiones, las trazamos miles de veces desde el punto de origen, resultando un conjunto de múltiples fibras que pasan por el punto origen y cada vez usamos una orientación diferente de acuerdo con su verosimilitud. El conjunto de todos estos caminos diferentes son entonces colectivamente una medida de la probabilidad de conexión o probabilidad. El resultado final es un conjunto de fibras probables, que no necesariamente tienen que existir.

En ambos casos, se puede analizar directamente la estimación de los tractos neuronales o pueden ser comparados con un umbral de tal forma que las conexiones solo se consideran existentes si existe un número mínimo de líneas(streamlines) o se alcanza una mínima probabilidad (tractografía probabilística). Estas 'streamlines' (como vamos a llamarlas a partir de ahora) son obtenidas conectando un conjunto de píxeles en base a sus direcciones preferidas y es un ejemplo de tractografía determinista.

2.7. Conectómica

Una importante [2] aplicación dentro de la neurociencia es la conectómica, que es el estudio y la producción de los conectomas. Los conectomas [5] son un conjunto de conexiones en una red neuronal, es decir, describen las redes de conexiones sinápticas entre las neuronas del cerebro. Para estudiarlas [2] se definen distintas regiones de interés en la estructura cerebral que son los nodos de la red y, mediante tractografía, se analizan las conexiones de tractos de fibras entre ellos. A partir de los datos resultantes se puede crear una matriz de conectividad que las represente, aunque estas matrices obtenidas con tractografía varían de acuerdo a la modalidad de imágenes de difusión utilizadas para obtener los datos.

La obtención de las matrices de conectómica depende del proceso usado en cada caso pero generalmente se siguen los siguientes pasos [2] [5] [7]:

- Procesado de la imagen anatómica(T1): A partir de esta imagen obtenida con una gran resolución del cerebro, hay que generar una segmentación cortical de la materia blanca y la materia gris del cerebro. También hay que realizar una parcelación de la segmentación en varias regiones anatómicas.
- Procesado de la imagen de difusión: Utilizando la imagen dada, se realiza una estimación de los tractos de fibras que componen la materia blanca mediante una tractografía.
- Elaboración de un mapa de la conexiones neuronales del cerebro: Con la información obtenida de los dos pasos anteriores y la realización de una segmentación del córtex cerebral en varias regiones y subdividir las en pequeñas regiones de interés, sabiendo dónde empiezan y terminan las fibras en cada una de ellas gracias a la tractografía, obtenemos la red o mapa neuronal.
- Obtención de una matriz de conectividad: Se obtiene computando las conexiones de las fibras entre pares de regiones de interés. Para cada conjunto de fibras entre dos regiones se pueden obtener una serie de parámetros que las caractericen, como el número de fibras que la forman o el valor medio de la FA. Con cada característica se obtiene una matriz de conectividad diferente.

Capítulo 3. Identificación biométrica

3.1. Introducción

La biometría [41] es la ciencia que se dedica a identificar personas mediante características fisiológicas. La identificación biométrica es un tipo de identificación basada en características biométricas, las cuales a diferencia de la identificación tradicional basada en información o pertenencias de cada individuo, son rasgos individuales de cada persona. Este tipo de identificación es mucho más seguro que el convencional, además de poder identificar a la gente a través de varias características, ya que el cuerpo humano dispone de muchas. Para poder identificar biométricamente a un individuo se utilizan los llamados sistemas de identificación biométrica (BISs, Biometric Identification Systems) que pueden ser considerados como sistemas de reconocimiento de patrones que establecen la autenticidad de una característica fisiológica o de comportamiento perteneciente a un usuario. Las técnicas biométricas más utilizadas son el reconocimiento automático de huellas, caras, retina, iris, geometría de la mano, voz, etc.

Hay que tener en cuenta que aunque nosotros estamos aplicando identificación a las resonancias de difusión, en la práctica no tiene mucho sentido ya que como vamos a describir, algunos criterios como la facilidad de uso y la aceptación del usuario, no se cumplen para este tipo de BISs. Lo estamos haciendo como ejercicio para estudiar la individualidad del conectoma en una persona.

Una serie de criterios para permitir caracterizar diferentes BISs son:

- Seguridad: Debido al ruido en los sensores, limitaciones en el procesamiento y la variabilidad en las características biométricas es posible que no se pueda garantizar la correcta autenticación por un BIS. Para aplicar una tecnología biométrica a una aplicación de identificación de una persona es importante comprender y evaluar la tecnología en el contexto de la aplicación objetivo y de la población objetivo.
- Facilidad de uso: Hay un equilibrio entre la complejidad del uso del sistema y el nivel de seguridad que se debe asegurar.
- Aceptación del usuario: La mayoría de usuarios no aceptarían sistemas incómodos de usar y considerarían que son difíciles de usar. Pero para aplicaciones de alta seguridad, un BIS complicado e incluso engorroso de usar puede no ser un obstáculo y estar aceptado si proporciona una sensación de seguridad. Además hay un importante problema, que es si los BIS pueden ser usados para violar los derechos de privacidad de las personas. La tecnología biométrica puede prevenir este problema como cualquier otra técnica de identificación.
- Facilidad de implementación: La tecnología biométrica necesita estar fabricada de manera accesible para la integración de los sistemas y su implementación. Además debido a la falta de estándares en la industria de los BISs, el aprovechamiento de la tecnología biométrica no es sencillo.
- Coste: El equipamiento, instalación, costes de entrenamiento, software, mantenimiento del sistema y costes de operación pueden ser problemas a la hora de considerar el coste total para desplegar un sistema. En un futuro cercano, debido al aumento de la disponibilidad de potencia de procesado más barata y una gran producción de sensores baratos, puede ser posible hacer que la biometría sea accesible a todo el mundo.

Vamos a describir un modelo general de un sistema de identificación biométrica mediante la [figura 3-1](#):

- Subsistema de recopilación de datos: Este subsistema captura la característica biométrica a analizar mediante un sensor. Además hay que tener en cuenta el grado de colaboración del usuario y el entorno donde la recopilación de datos tiene lugar a la hora de diseñar un BIS con el fin de obtener la menor variación a la hora de obtener los datos.
- Subsistema de transmisión: A veces la recopilación de datos biométricos y el procesamiento están almacenados en diferentes localizaciones, por lo que es necesario la transmisión de los datos.
- Subsistema de procesamiento de señal: Este subsistema convierte los datos originales en un vector de características, intentando preservar toda la información discriminante necesaria y eliminando toda la información no relevante para poder distinguir dos individuos diferentes.
- Subsistema de almacenamiento: Contiene las plantillas correspondientes a los usuarios registrados en el sistema.
- Subsistema de decisión: Obtiene unos valores numéricos de la comparación de un vector de características con una plantilla almacenada. Es necesario aplicar alguna estrategia para decidir si el usuario está registrado en la base de datos. Hay múltiples formas de realizar el proceso de decisión, pero por ejemplo la más simple es comparar la distancia con un umbral.

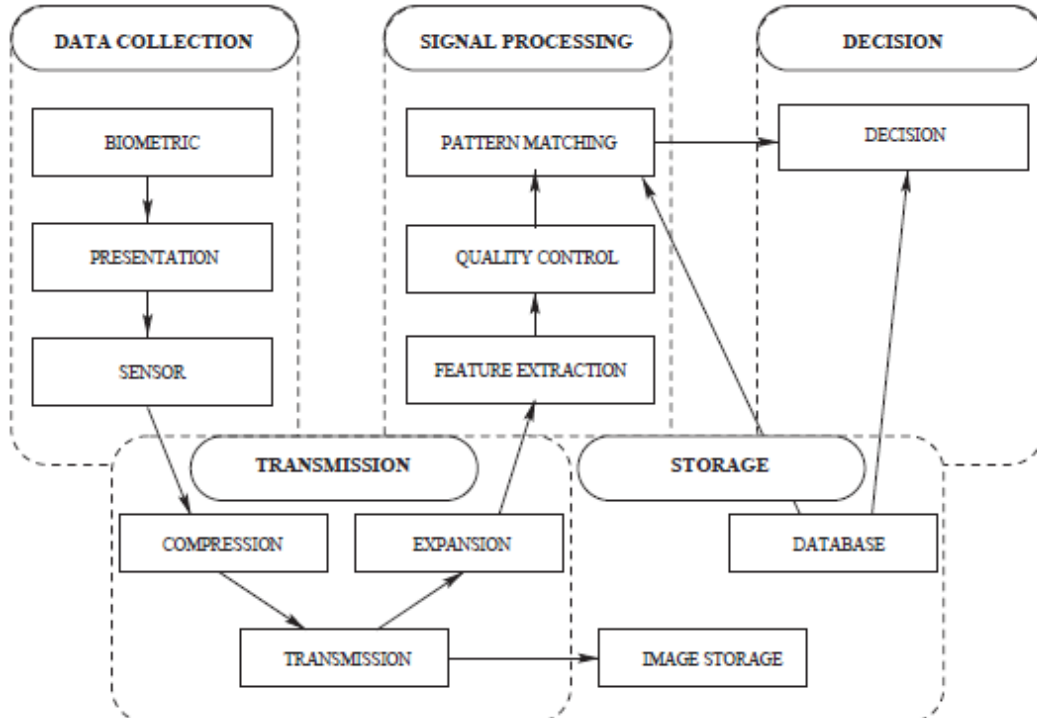


Figura 3-1 Modelo general de un sistema biométrico

3.2. Identificación vs Verificación

Los BISs [41] pueden operar de dos maneras diferentes que son la verificación o autenticación y la identificación. En el proceso de verificación el usuario demanda ser una persona, y el sistema acepta o rechaza esta petición después de comparar la característica biométrica de este individuo con la de otros usuarios almacenados en una base de datos. Este proceso implica conocer presuntamente la identidad del individuo a autenticar, por lo tanto, dicho individuo ha presentado algún tipo de credencial que le identifique, que después del proceso de autenticación biométrica será validada o no. Se pueden tener dos tipos de errores a la hora de decidir la clasificación del individuo acerca de si es quién dice ser: la tasa de aceptación falsa (FAR, false accept rate), que es la probabilidad de aceptar la petición de que el usuario es la persona que él dice ser, cuando este usuario no es esa persona y la tasa de rechazo falso (FRR, false reject rate), que es la probabilidad de rechazar la petición de que un usuario es la persona que dice ser, cuando verdaderamente este usuario es esa persona.

En el proceso de identificación, el usuario accede al sistema de identificación biométrica, el cual extrae características biométricas del usuario y las compara con los otros patrones almacenados en la base de datos para decidir si esta persona está entre los usuarios registrados en el sistema, y cuál es su identidad. Este proceso implica no conocer a priori la identidad presunta del individuo y el resultado de este proceso es la identidad del individuo, mientras que en el proceso de verificación es un valor verdadero o falso.

El proceso de identificación es más exigente que el de autenticación o verificación biométrica, sobre todo cuando el número de usuarios es elevado. Esto es debido a que la necesidad de procesamiento y comparaciones es más reducida en el proceso de verificación. Por esta razón, es habitual usar autenticación cuando se quiere validar la identidad de un individuo desde un sistema con capacidad de procesamiento limitada o se quiere un procesado muy rápido.

3.3. Identificación biométrica como problema de reconocimiento de patrones

Como ya hemos dicho, un BIS puede ser considerado un sistema de reconocimiento de patrones. Esta disciplina [42] incluye muchas aplicaciones pero nos centramos en la biometría que es la que nos interesa. Un patrón podría ser una cara humana, una imagen de una huella dactilar, una señal de voz, etc. Para reconocer o clasificar un patrón se escogen entre dos procedimientos: Clasificación supervisada, donde el patrón de entrada es identificado como un miembro de una clase ya predefinida y clasificación no supervisada, donde el sistema agrupa los patrones sin ningún criterio establecido a priori durante el diseño del sistema. Así pues, estas clases han sido previamente definidas por el diseñador del sistema (clasificación supervisada) o fueron aprendidas basándose en la similitud de los patrones (clasificación no supervisada). En identificación biométrica el patrón de entrada puede ser una huella, la cara, el iris, etc mientras que las clases de patrones corresponden a los usuarios autorizados para el control de acceso.

Para diseñar un sistema de reconocimiento de patrones se necesita:

- 1) Adquisición de los datos: Gracias a un dispositivo(un sensor) obtenemos una serie de datos, los cuales vamos a clasificar más tarde. Por ejemplo, en nuestro documento hemos obtenido imágenes de difusión y anatómicas mediante una resonancia magnética.
- 2) Preprocesado: A veces es necesario corregir posibles deficiencias en los datos debido a errores en el proceso de adquisición de ellos, o bien para preparar los datos de cara a posteriores procesos en las etapas de extracción de características o clasificación. Por ejemplo, en el Capítulo 4 hay varios apartados en los que hemos cambiado el formato de algunos datos, para procesarlos de una manera más eficiente.
- 3) Extracción de características: Es el proceso de generar vectores de características que puedan ser usadas en el proceso de clasificación de los datos. Por ejemplo, nosotros hemos convertido las imágenes dadas por la resonancia, en matrices de conectómica, mediante un procesado del cual hablaremos en el Capítulo 4. También es habitual realizar un procedimiento de selección de características, por el cual elegimos solamente parte del conjunto inicial de características para llevar a cabo la clasificación posterior. Esta selección de características permite reducir la dimensionalidad del problema, y descartar características que no son suficientemente discriminantes.
- 4) Clasificación: La clasificación trata de asignar las diferentes partes del vector de características a grupos o categorías, basándose en las características extraídas. Existen muchos tipos de clasificación, entre las que destacan la clasificación supervisada y no supervisada de la cual hemos hablado.

3.4. Identificación biométrica mediante resonancia de difusión

Hace años se descubrió que la función cerebral y su estructura es bastante variable entre sujetos sanos[46][43]. Recientemente se ha demostrado que la variación entre las conexiones cerebrales en la materia blanca en individuos distintos es suficiente para realizar una identificación entre estos sujetos de una manera robusta y segura.

Para ello hemos utilizado la resonancia de difusión[45] para medir patrones microscópicos de difusión en la materia blanca y definir la estructura de regiones del conectoma, pero en lugar de mapear cada región con conexiones podríamos haber utilizado el conectoma local (Local Connectome) que es una técnica que supera las limitaciones de la resonancia de difusión. El conectoma local está definido como el grado de conectividad entre vóxeles adyacentes en la materia blanca. Mediante el concepto de SDF (Spin Distribution Function), que mide la densidad de agua que se está difundiendo en una cierta dirección, podemos utilizarlo para cuantificar el conectoma local. Para esa dirección usa, en cada voxel, la orientación dada en ese voxel por un atlas. Este concepto nos permite caracterizar mediante un vector de características el mapa de conexiones del conectoma local, y mediante la distancia euclídea entre dos vectores de diferentes personas cada uno, podemos utilizarla para distinguirlos. La principal diferencia entre nuestro método de difusión y la SDF, es que en el primer caso la difusión cuantifica cuánto se difunde el agua a lo largo del tejido dependiendo de la estructura de los haces de fibra. Y la SDF cuantifica cómo el agua se difunde a lo

largo de los tractos de las fibras siendo sensible a una serie de características de la materia blanca como la densidad de los haces de fibra.

También destacamos que se ha demostrado en numerosos estudios que si el intervalo de tiempo en el que hemos hecho dos resonancias de difusión de un individuo aumenta, disminuye la semejanza entre la conectividad cerebral del sujeto dado.

Aunque queremos resaltar que otros estudios[44] han tenido en cuenta otra serie de características con el fin de identificar individuos mediante la conectómica como son la edad mental, inteligencia, polimorfismo genético y personalidad entre otras, ya que algunas pueden estar directamente relacionadas con las conexiones cerebrales de cada sujeto porque como bien sabemos muchas de ellas contribuyen a hacer única cada persona.

Capítulo 4. Procesado para la obtención de matrices de conectividad

4.1. Medios de partida

4.1.1. Datos

Asumimos[8] que tenemos una adquisición correspondiente al protocolo habitual de DTI:

- Adquisición de una imagen T1 en formato DICOM, con $1 \times 1 \times 1$ mm³ de resolución. El formato DICOM es el formato estándar internacional para las imágenes médicas y la información relacionada.
- Adquisición de DWIs en formato DICOM, con $2 \times 2 \times 2$ mm³ de resolución, $b\text{-value}=1000$ s/mm² y 61 gradientes.

Este Capítulo contiene la descripción de los pasos seguidos para procesar tanto la imagen anatómica (T1) como la imagen de difusión (DWIs), todo ello con el propósito de calcular matrices de conectividad.

El procedimiento consiste en ejecutar funciones de Matlab que generan a su vez varios scripts con todos los comandos que van ejecutando los distintos pasos. Luego hay que ejecutar esos scripts, con la salvedad de que pueden necesitar ser ejecutados en diferentes máquinas (en local y en el servidor).

4.1.2. Software

El estudio acerca de la obtención de matrices de conectómica se ha realizado de manera remota en dos servidores de la Universidad de Valladolid: `tanis.lpi.tel.uva` y `tebas.lpi.tel.uva.es` que trabajan con un sistema operativo Ubuntu 16.04.2. Además de las herramientas propias del sistema para el manejo de ficheros y directorios, se utilizan distintos softwares especializados en el análisis de imágenes de resonancia magnética y herramientas matemáticas.

Software necesario para llevar a cabo el procesado:

- Matlab: Este programa desarrollado por Mathworks[10], es una herramienta de software matemático que ofrece un entorno de desarrollo integrado (IDE) con un lenguaje de programación propio y de alto nivel. Entre sus numerosas funciones, Matlab permite analizar datos, trabajar con matrices, desarrollar algoritmos y crear aplicaciones. En nuestro procedimiento para obtener las matrices de conectividad, Matlab realizará las siguientes funciones[8]:
 - Genera varios scripts con todos los comandos que van ejecutando los distintos pasos del procesado para obtener las matrices.
 - Crearemos un algoritmo en el lenguaje de programación de Matlab con el fin de cumplir la función final de este Trabajo de Fin de Grado.
- Slicer versión 4.5[11]: Es una plataforma de software libre para el procesamiento de imágenes y visualización 3D de ellas. Slicer proporciona, entre otras, una gran variedad de herramientas para el estudio de imágenes de resonancia magnética y de difusión, además de un entorno gráfico para la representación de los volúmenes. En nuestro análisis de la conectividad, hemos utilizado principalmente a Slicer para la visualización de imágenes de resonancia

y para la conversión de formatos de éstas con el fin de poderlas visualizar de una manera más detallada.

- FSL[12]: Es una biblioteca de herramientas para el análisis de imágenes de resonancia magnética funcional y de difusión. También, proporciona una interfaz gráfica para la visualización de los datos (fslview). Las herramientas más destacadas que hemos utilizado en el procesado son[8]:
 - Extracción del cerebro, eliminando cráneo y elementos no pertenecientes al cerebro.
 - Obtención de tensores y medidas escalares asociadas(FA, MD,...)
 - Registrado para proyectar la segmentación cortical sobre el volumen tensorial, que es el espacio donde está calculada la tractografía.
- MRtrix[7]: Esta herramienta proporciona una gran cantidad de herramientas para el procesamiento, análisis y visualización de imágenes, concentrándose en el análisis de la materia blanca utilizando la resonancia de difusión, en un algoritmo probabilístico para la tractografía sobre la materia blanca y el análisis de la conectividad estructural principalmente. Hemos utilizado a MRtrix para las siguientes partes del procesado [8]:
 - Obtención de una máscara del cerebro.
 - Obtención de una segmentación de los diferentes tipos de tejido cerebral. Para ello, generamos la imagen 5tt(5 tipos de tejido) mediante la parcelación de la segmentación cortical generada previamente con Freesurfer.
 - Estimación de la "response function"(RF) y de la "fibre orientation distribution"(FOD), la cual es necesaria para realizar la tractografía. Se estima la distribución de las orientaciones de la fibra(FOD) a partir de los datos de difusión usando una deconvolución esférica. Para realizar el proceso de la deconvolución, uno de los pasos necesarios es hallar la función de respuesta (RF) que es la señal esperada para un voxel que contiene un único haz de fibras orientado coherentemente en la materia blanca.
 - Realización de la tractografía completa del cerebro.
 - Adaptar la segmentación cortical generada por Freesurfer para calcular la conectómica.
 - Obtención de las matrices de conectividad.
- Freesurfer[13]: Es un software de código abierto utilizado para el procesamiento, análisis y visualización de imágenes cerebrales obtenidas mediante resonancia magnética. Entre sus principales herramientas, destacamos que sirve para el registrado de la imagen, para la segmentación cortical y subcortical y para la realización de tractografías. Hemos utilizado Freesurfer para las siguientes partes del procesado:
 - Generación de una segmentación cortical.
 - Conversión del tamaño del volumen de la imagen anatómica T1 a un tamaño 256x256x256 usando la herramienta mri_convert.
 - Conversión de la segmentación cortical al formato .nii para su posterior procesado.
 - Parcelación y etiquetado de las regiones corticales y subcorticales.
- MRICConvert [14]: Es un conversor de ficheros de imágenes médicas que convierte los ficheros en formato DICOM a varios formatos de volúmenes que son FSL NifTi, SPM Analyze, Meta Image, NifTi, Analyze 7.5 y BrainVoyager. Hemos utilizado MRICConvert para la siguiente parte del procesado:

- Conversión del formato DICOM de la imagen T1 a un formato .nii utilizando la herramienta mcverter.

4.1.3. Estructura de directorios

Antes de empezar el procesado es necesario la generación de la estructura de directorios necesaria para almacenar cada uno de los ficheros obtenidos por los distintos programas utilizados[8].

Se muestra un ejemplo con el nombre de caso *genérico* caso001:

- datos_MRI_cerebral
 - 3T
 - difusión
 - DWIs
 - tensores_con_FSL
 - MRTRIX_brain_mask
 - MRTRIX_spherical_deconvolution
 - MRTRIX_whole_brain_tractography
 - MRTRIX_conectomica
 - estructural
 - T1s
 - T1s_256x256x256
 - Freesurfer_cortical_parcelation
 - Freesurfer_segmentacion_cortical_resultado
 - originales_dicom
 - caso001
 - DICOM
 - IM_0001 (T1)
 - IM_0003 (DWIs)
- scripts_procesar_3T

Desde el directorio `scripts_procesar_3T` se ejecutarán todos los comandos y scripts, así que nos situamos en la carpeta de los scripts antes de comenzar el procesado. Para ello ejecutamos lo siguiente en la consola de comandos:

cd datos_MRI_cerebral/3T/scripts_procesar_3T

Una vez situados comenzará el procesado para la obtención de las matrices de conectómica.

4.2. Estructura del procesado

Vamos a realizar un análisis llamado conectómica, que consiste en calcular medidas de una red neuronal, obteniendo para ello las conexiones entre las distintas regiones de interés de la corteza cerebral.

Para realizar este análisis vamos a describir brevemente el esquema del procesado a seguir[8]:

1. Preparación del entorno: Fijar el valor de las variables de entorno de los programas y actualizar la variable PATH con la ruta de directorios de cada programa.
2. Procesado de la imagen anatómica (T1), que incluye:

- a. Conversión de la imagen T1 en formato DICOM a formato .nii. Después se convierte esta imagen en .nii a un volumen $256 \times 256 \times 256 \text{ mm}^3$.
 - b. Extracción de lo que no es cerebro de la imagen T1 con el nuevo volumen y formato.
 - c. Generación de una segmentación cortical del volumen del cerebro y parcelación de la superficie cortical en distintas regiones de interés. Tras obtener el resultado de la segmentación se cambia al formato .nii que no da problemas en el procesado.
3. Procesado de la imagen de difusión (DWIs), que incluye:
- a. Conversión de la imagen de difusión en formato DICOM a formato .nii para su posterior procesado y formato .nrrd para su visualización y creación de dos ficheros que contienen el $b\text{-value}=1000 \text{ s/mm}^2$ y las 61 direcciones de gradientes.
 - b. Obtención de una máscara del cerebro a partir de la imagen DWI y los ficheros con las direcciones de gradiente y los b-values.
 - c. Obtención de una segmentación de los diferentes tipos de tejido cerebral. Para ello, generamos la imagen 5tt(5 tipos de tejido) mediante la parcelación de la segmentación cortical generada previamente.
 - d. Estimación de la "response function"(RF) y de la "fibre orientation distribution"(FOD).
 - e. Obtención de tensores y medidas escalares asociadas(FA, MD,...)
 - f. Adaptar la segmentación cortical generada por Freesurfer para calcular la conectómica.
 - g. Registrado para proyectar la segmentación cortical sobre el volumen tensorial, que es el espacio donde esta calculada la tractografía.
 - h. Realización de la tractografía completa del cerebro.
 - i. Obtención de las matrices de conectividad.

4.3. Preprocesado

Antes de empezar a ejecutar los programas, nos situamos en uno de los servidores disponibles y realizamos las siguientes partes:

1. Fijar el valor de las variables de entorno de los programas. Para ello escribimos en la consola de comandos las siguientes líneas:


```
export FSLOUTPUTTYPE="NIFTI_GZ"
export FSLDIR="/opt/shared/fsl"
export FREESURFER_HOME="/opt/shared/freesurfer"
export SUBJECTS_DIR="/datos/pruebas_daniel/estructural/Freesurfer_cortical_parcelation"
```
2. Actualizar la variable PATH con la ruta de directorios de cada programa. Para ello escribimos:


```
PATH="$PATH:/opt/shared/MRConvert"
PATH="$PATH:/opt/shared/fsl/bin"
PATH="$PATH:/opt/shared/slicer/SuperBuild-Debug/Slicer-build/lib/Slicer-4.3/cli-modules/"
PATH="$PATH:/opt/shared/MRtrix/mrtrix3/release/bin:/opt/shared/MRtrix/mrtrix3/scripts"
source $FREESURFER_HOME/SetUpFreeSurfer.sh
```

Tras la ejecución de los anteriores pasos, estamos listos para utilizar los programas necesarios para el análisis de la resonancia de difusión mediante la conectómica.

4.4. Procesado de T1

4.4.1. Conversión de formato y volumen de la imagen T1

1. Conversión de los volúmenes T1 de formato DICOM a formato .nii

El volumen T1[8] al igual que el DWI están en formato DICOM, dado por la resonancia magnética de la universidad. Este formato da varios problemas durante el procesado por lo que es necesaria la conversión.

Para ello utilizamos 'mcverter'[14], que es una utilidad para línea de comandos de la herramienta MRIConvert, de la Universidad de Oregon. Escribimos:

```
mcverter -o ../estructural/T1s/caso001 -f nifti -n -F salida  
../originales_dicom/caso001/DICOM/IM_0001
```

Explicamos detalladamente las opciones utilizadas por el comando:

- o: ruta donde se va a almacenar el fichero de salida
- f: formato de salida
- n: Guardar los ficheros como .nii
- F: Nombre del fichero de salida

Ahora vamos a cambiar el nombre del fichero de salida dado por mcverter con el fin de que tenga un nombre más descriptivo:

```
mv /datos/pruebas_daniel/estructural/T1s/caso001/salida.nii  
/datos/pruebas_daniel/estructural/T1s/caso001/caso001_T1.nii
```

Vamos a mostrar en la figura 4-1 un ejemplo de la imagen anatómica T1 en formato .nii con el siguiente comando de MRtrix[15]: ***mrview ../estructural/T1s/caso001_T1.nii***

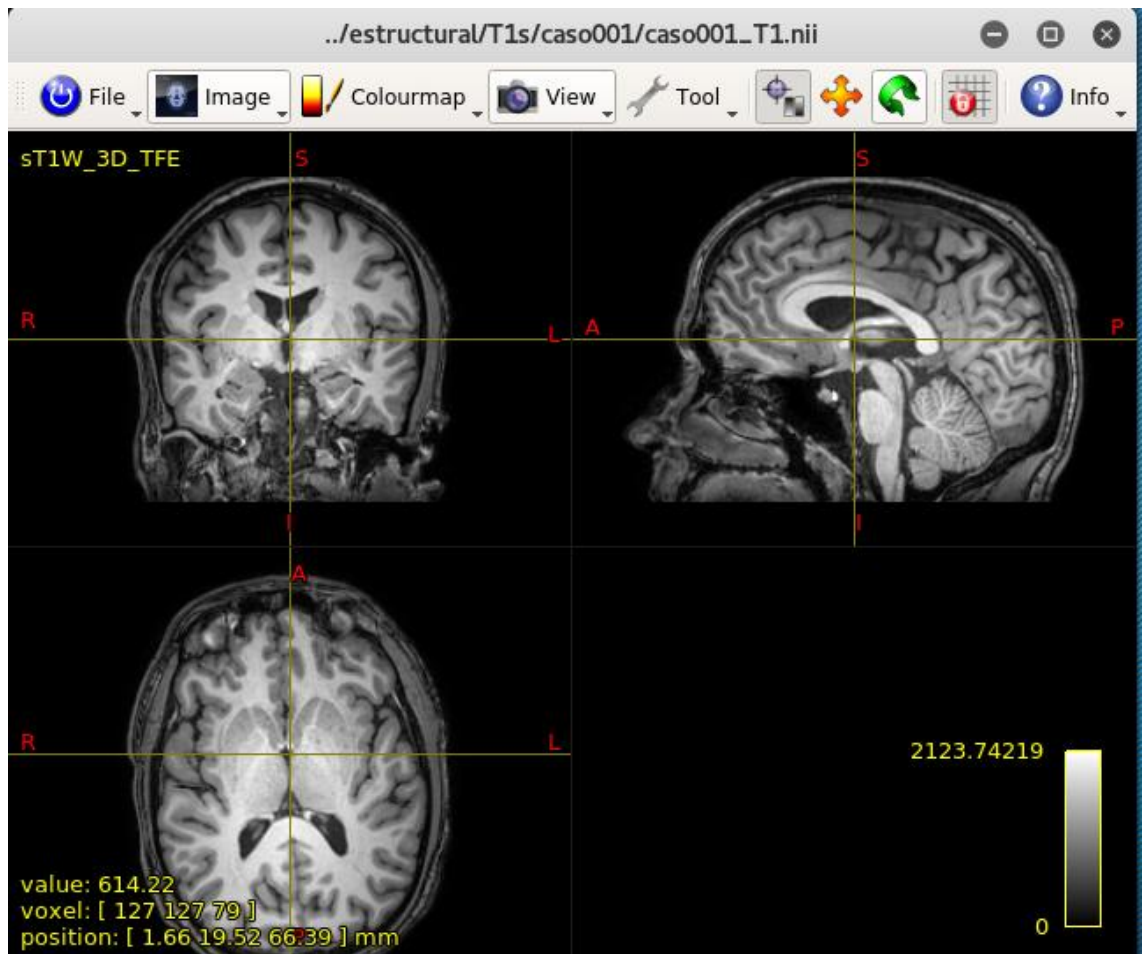


Figura 4-1: Visualización de una imagen anatómica T1 mediante MRtrix

2. Conversión de los volúmenes T1 a un tamaño 256x256x256. Esto es necesario porque Freesurfer genera una parcelación cortical con tamaño 256x256x256, y luego tendremos que realizar un registrado, así que necesitamos hacerlo a partir de un volumen que tenga el mismo tamaño que la segmentación realizada con Freesurfer. Para ello usamos la herramienta `mri_convert` de Freesurfer.

i) Creamos una carpeta del caso correspondiente, que es donde almacenaremos el resultado de la imagen T1 con un volumen 256x256x256

```
mkdir ../estructural/T1s_256x256x256/caso001
```

ii) Convertimos el volumen actual del T1 a 256x256x256 con la herramienta `mri_convert`[16] de Freesurfer la cual cambia el formato de diferentes ficheros.

Explicamos detalladamente la opción utilizada por el comando:
 -c: Se ajusta al tamaño de 1 mm el tamaño del voxel en el hueso coronal. Corte en la dirección de 256x256x256 o más.

```
mri_convert -c ../estructural/T1s/caso001/caso001_T1.nii  

../estructural/T1s_256x256x256/caso001/caso001_T1_256.nii.gz
```


4.4.2. Eliminación de materia no cerebral en el volumen T1

Para ello utilizamos la herramienta de fsl, llamada bet. Esto servirá para más adelante en la parte de conectómica. Usamos la opción -R con el comando bet[17] porque si no, hace mal la extracción del cerebro en algunos casos.

Descripción de la opción:

-R: Esta opción realiza una estimación del centro del cerebro más robusta. Su principal propósito es de mejorar la extracción del cerebro cuando los datos de entrada contienen mucho tejido no cerebral(más abundante aún cuando hay una gran parte de cuello incluido en la imagen de entrada).

La herramienta realiza la extracción del cerebro, obteniendo el fichero caso001_T1_256_masked.nii.gz que solo contiene cerebro:

```
bet ../estructural/T1s_256x256x256/caso001/caso001_T1_256.nii.gz  
../estructural/T1s_256x256x256/caso001/caso001_T1_256_masked.nii.gz -R
```

Vamos a mostrar en la figura 4-2 un ejemplo de la imagen anatómica T1 en formato .nii y con volumen 256x256x256 tras haber eliminado materia no cerebral con el siguiente comando de MRtrix[15]:

```
mrview ../estructural/T1s_256x256x256/caso001/caso001_T1_256_masked.nii.gz
```

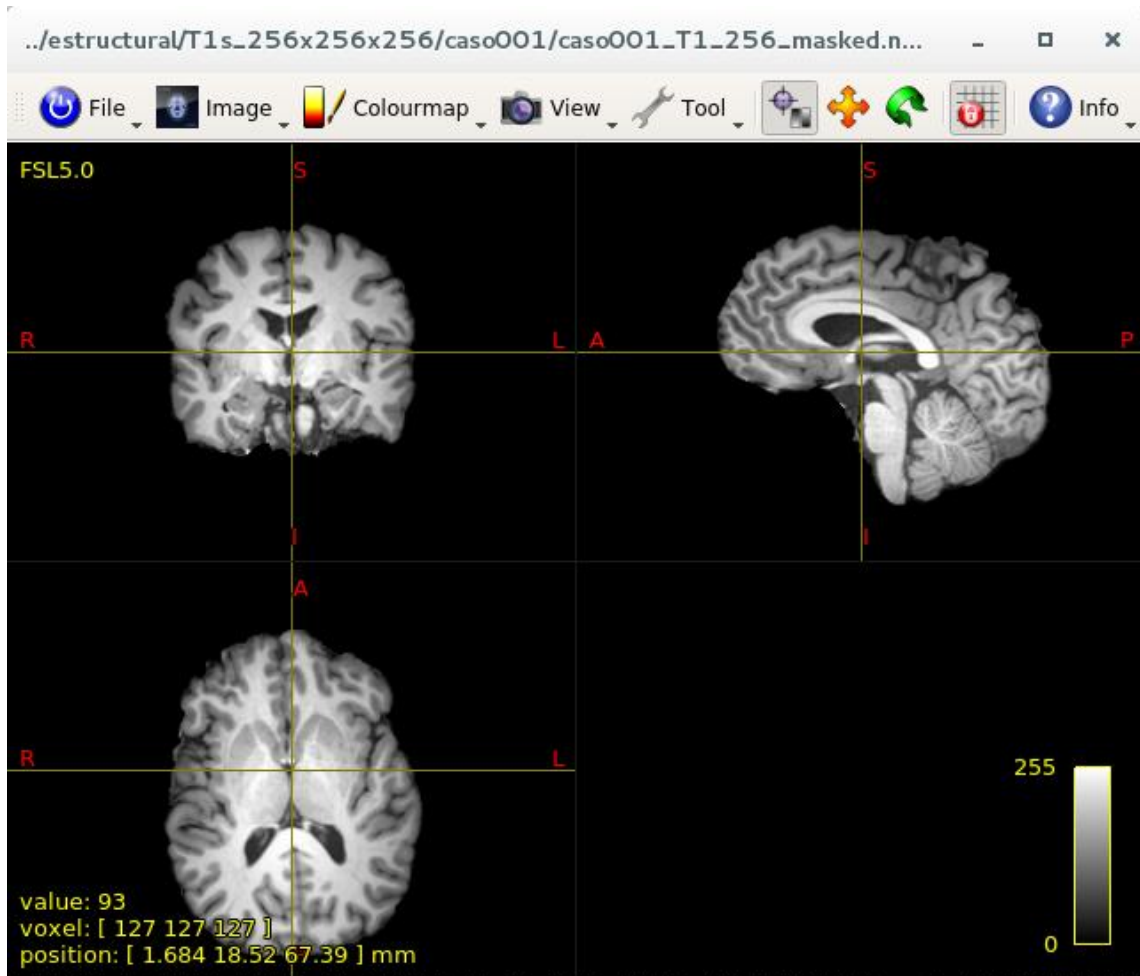


Figura 4-2: Visualización de la extracción de materia no cerebral de la imagen anatómica T1 mediante MRtrix.

4.4.3. Segmentación y parcelación

La segmentación y la parcelación[19][20] consisten en el etiquetado de cada voxel de las estructuras cerebrales usando una variedad de información que incluye intensidades de imagen, posición global dentro del cerebro y posición relativa a las estructuras cerebrales vecinas. La segmentación divide el cerebro en varias regiones distintas entre las que se encuentra la corteza cerebral, materia gris, materia blanca, etc. La parcelación divide la superficie cortical en varias regiones de interés y las etiquetas asignadas a cada voxel pueden tomar varios valores, identificando qué tipo de región o tejido pertenece al voxel.

Tras haber realizado la conversión de formato de la imagen T1 procedemos con el siguiente paso del análisis de la conectómica, el cual consiste en la parcelación y segmentación de esta imagen anatómica en el nuevo formato. Para ello vamos a utilizar la herramienta *recon-all* [18] de Freesurfer que realiza una segmentación cortical y subcortical del cerebro, dividiéndolo en varias regiones anatómicas. Esta herramienta realiza múltiples pasos[21]: Primero partiendo de la imagen T1 y utilizando un atlas predeterminado de cerebro, Freesurfer genera y elimina el cráneo y el cuello de la misma así como reduciendo los efectos del movimiento y normalizando la intensidad de

las imágenes. Luego, alinea el volumen con el atlas ya dicho y procede a la segmentación, separando la materia blanca y la materia gris. A continuación, reconstruye diferentes superficies y volúmenes que permiten visualizar y estudiar distintas estructuras cerebrales, como la materia blanca o la corteza cerebral. Finalmente, realiza la parcelación de la superficie cortical y genera tablas con información sobre las distintas regiones.

Ahora procedemos a mostrar las líneas de los comandos escritos en la consola para tal fin:

1. Generación de una segmentación cortical usando FreeSurfer

Descripción de las opciones utilizadas de *recon-all*[18]:

-s : nombre del sujeto, del que se creará una nueva carpeta de directorios con ese nombre en la ruta ../estructural/Freesurfer_cortical_parcelation

-i: entrada específica en .nii

-all: hace todo, incluyendo la segmentación subcortical.

i) Hay que explicarle a FreeSurfer cómo se va a llamar el caso generado, y dónde está el T1 en formato .nii, ya que el formato DICOM da problemas a la hora de generar la segmentación. Para ello escribimos:

```
recon-all -s caso001 -i ../estructural/T1s/caso001/caso001_T1.nii
```

ii) Ahora procedemos a lanzar la segmentación propiamente dicha:

```
recon-all -s caso001 -all
```

Tras terminar el procesado, en la carpeta Freesurfer_cortical_parcelation se creará una serie de carpetas. Entre ellas está la carpeta para el procesado del siguiente apartado.

2. Conversión del resultado de la segmentación cortical a formato .nii

Esto lo podemos hacer con la utilidad mri_convert de FreeSurfer, pero antes ejecutamos los siguientes comandos:

i) Creamos una carpeta del caso correspondiente, que es donde almacenaremos el resultado de la segmentación cortical en formato .nii.

```
mkdir ../estructural/Freesurfer_segmentacion_cortical_resultado/caso001
```

ii) Copiamos la imagen volumétrica aparc+aseg.mgz a la carpeta previamente creada en el apartado i) , ya que es el fichero que usaremos para definir los nodos de nuestra conectómica.

```
cp ../estructural/Freesurfer_cortical_parcelation/caso001/mri/aparc+aseg.mgz  
../estructural/Freesurfer_segmentacion_cortical_resultado/caso001/caso001_cortical  
_seg.mgz
```

Ahora vamos a realizar la conversión de la segmentación cortical del formato mgz a formato nii con el comando mri_convert[16], ya que más tarde usaremos esta segmentación cortical para obtener la imagen 5tt con un comando del programa MRtrix que trabaja mejor con el tipo de formato nii.

```
mri_convert
```

```
../estructural/Freesurfer_segmentacion_cortical_resultado/caso001/caso001_cortical  
_seg.mgz  
../estructural/Freesurfer_segmentacion_cortical_resultado/caso001/caso001_cortical  
_seg.nii
```

Vamos a mostrar en la figura 4-3 un ejemplo de la segmentación cortical realizada en formato .mgz con el siguiente comando de MRtrix[15]:

```
mrview  
../estructural/Freesurfer_segmentacion_cortical_resultado/caso001/caso001_cortical_seg.mgz
```

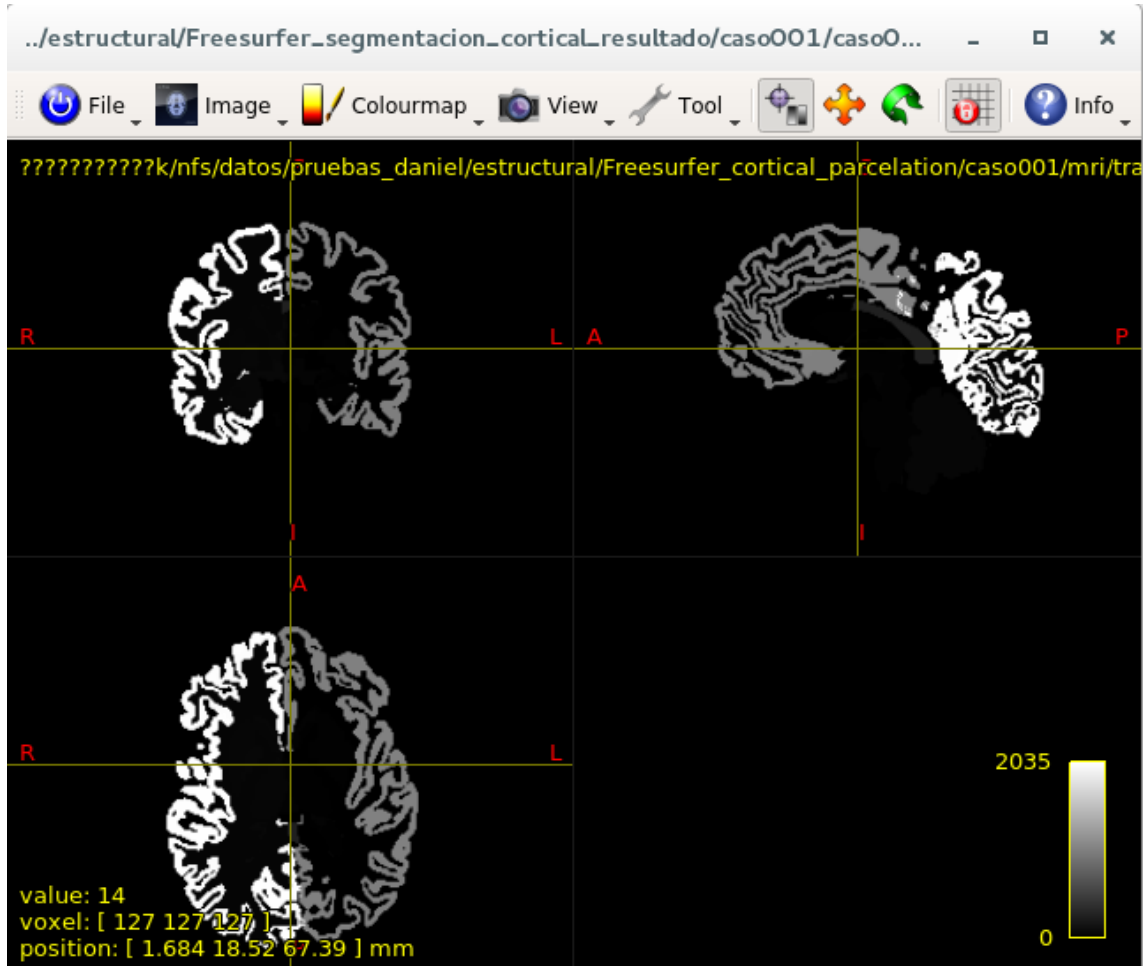


Figura 4-3: Segmentación cortical realizada sobre la imagen anatómica T1.

4.5. Procesado de DWI

4.5.1. Conversión de formato de la imagen de difusión DWI

Se usa la herramienta DWIConvert[22] a través de Slicer 4.5 en línea de comandos. Esta herramienta convierte imágenes de difusión hechas por resonancia magnética en formato DICOM a formato nrrd. Este programa también analiza el encabezado DICOM para extraer la información necesaria sobre el marco de medición, direcciones de ponderación de difusión, valores b, etc.

Vamos a describir las opciones utilizadas con este programa:

--inputDicomDirectory: Directorio que almacena el fichero en formato DICOM.

--outputVolume: Nombre del fichero de salida

--conversionMode: Define el formato de salida al que se va a convertir el fichero con el formato de entrada

Ahora vamos a describir los pasos realizados en este apartado:

i) Creamos una carpeta del caso correspondiente, que es donde almacenaremos el resultado de la imagen DWI en los formatos nrrd y nii.

```
mkdir ../difusion/DWIs/caso001
```

ii) Convertimos la imagen DWI del formato DICOM al formato nrrd para su posterior visualización con el programa Slicer 4.5 (hemos situado el ejecutable DWIConvert en la ruta que vemos)

```
../DWIConvert --inputDicomDirectory ../originales_dicom/caso001/DICOM/IM_0003  
--outputVolume ../difusion/DWIs/caso001/caso001_dwi.nrrd --conversionMode  
DicomToNrrd
```

iii) Convertimos la imagen DWI del formato DICOM al formato nrrd para su procesado más tarde

```
../DWIConvert --inputDicomDirectory ../originales_dicom/caso001/DICOM/IM_0003  
--outputVolume ../difusion/DWIs/caso001/caso001_dwi.nii --conversionMode  
DicomToFSL
```

También en la misma carpeta se crean para este caso dos ficheros correspondientes a el b-value=1000 s/mm² y las 61 direcciones de gradientes porque son los que da la resonancia magnética de la universidad que son:

```
caso001_dwi.bval caso001_dwi.bvec
```

Vamos a mostrar en la figura 4-4 un ejemplo de la imagen hecha por una resonancia magnética de difusión en formato .nrrd abriéndolo con el programa Slicer[11]:

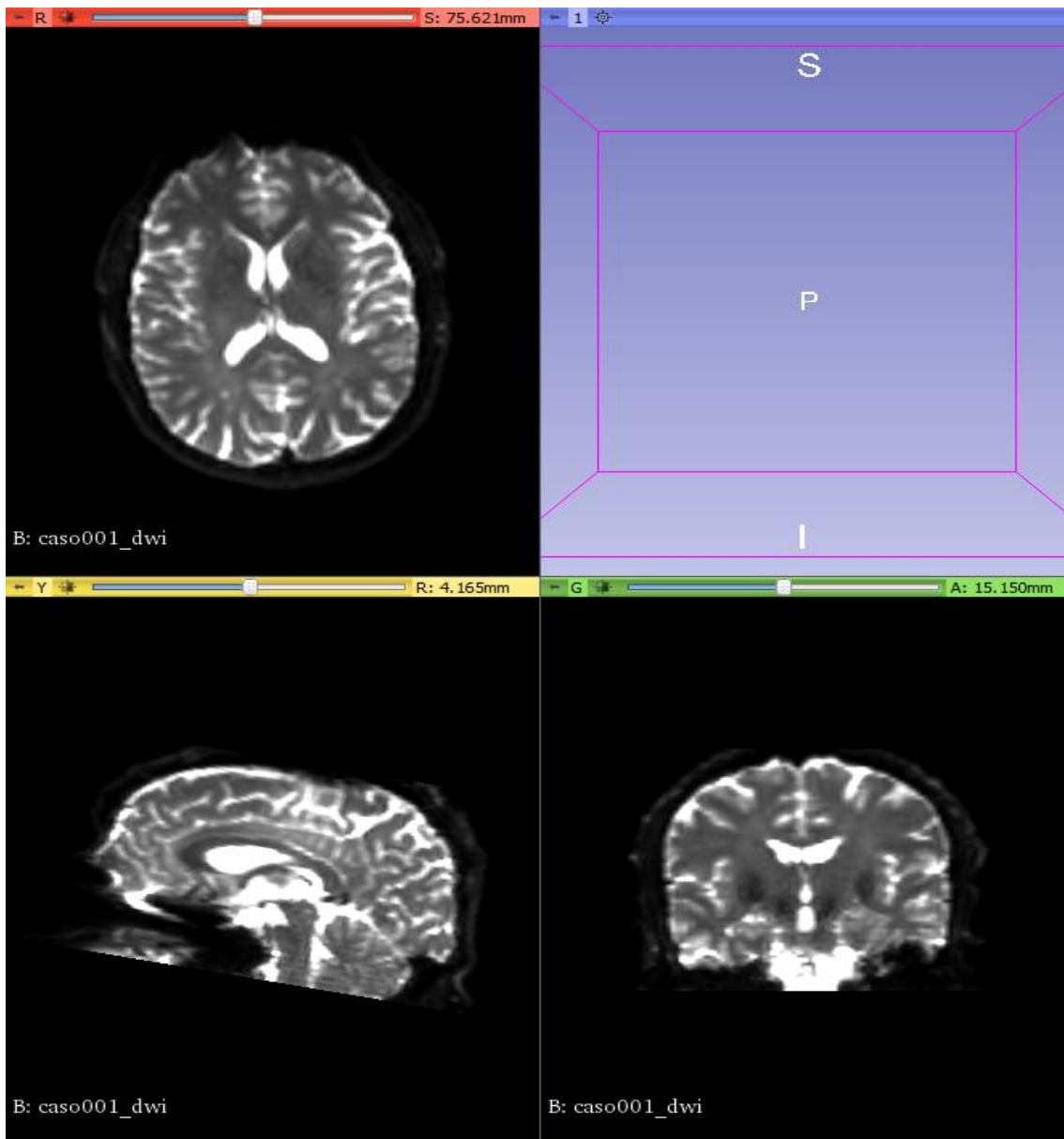


Figura 4-4: Imagen hecha por una resonancia magnética de difusión en formato .nrrd

4.5.2. Obtención de una máscara del cerebro a partir de la imagen de difusión DWI

Se usa MRtrix en línea de comandos. Hace falta introducir los ficheros con las direcciones de gradiente y los b-values. Como los que se generan automáticamente con DWIConvert tienen el problema de que están colocados en columna en lugar de estar colocados en fila, usamos unos ficheros precalculados que sirven para el caso de 61 gradientes.

Explicación del comando a utilizar y sus opciones:

dwi2mask: Genera una máscara entera del cerebro de una imagen DWI. Los ficheros con la dirección de los gradientes y los volúmenes con valor de $b=0$ son usado para obtener una máscara que incluye tanto tejido cerebral como CSF(fluido cerebroespinal) -fslgrad bvecs bvals : Especifica las direcciones de gradiente y los valores de b usados en la adquisición en formato FSL bvecs/bvals.

i) Creamos una carpeta del caso correspondiente, que es donde almacenaremos la máscara de la imagen DWI.

```
mkdir ../difusion/MRTRIX_brain_mask/caso001
```

ii) Generamos una máscara entera del cerebro de la imagen DWI del caso correspondiente con los ficheros de las direcciones de gradiente y los valores de b.

```
dwi2mask ../difusion/DWIs/caso001/caso001_dwi.nii
```

```
../difusion/MRTRIX_brain_mask/caso001/caso001_MRTRIX_brain_mask.nii
```

```
-fslgrad ../difusion/b_values_61_direcciones.bvec
```

```
../difusion/b_values_61_direcciones.bval
```

iii) Para comprobar que la máscara se ha realizado sin ningún problema visualizamos un ejemplo en contraposición con la imagen DWI en la Figura 4-5 mediante MRtrix [15]:

```
mrview ../difusion/DWIs/caso001/caso001_dwi.nii -roi.load
```

```
../difusion/MRTRIX_brain_mask/caso001/caso001_MRTRIX_brain_mask.nii
```

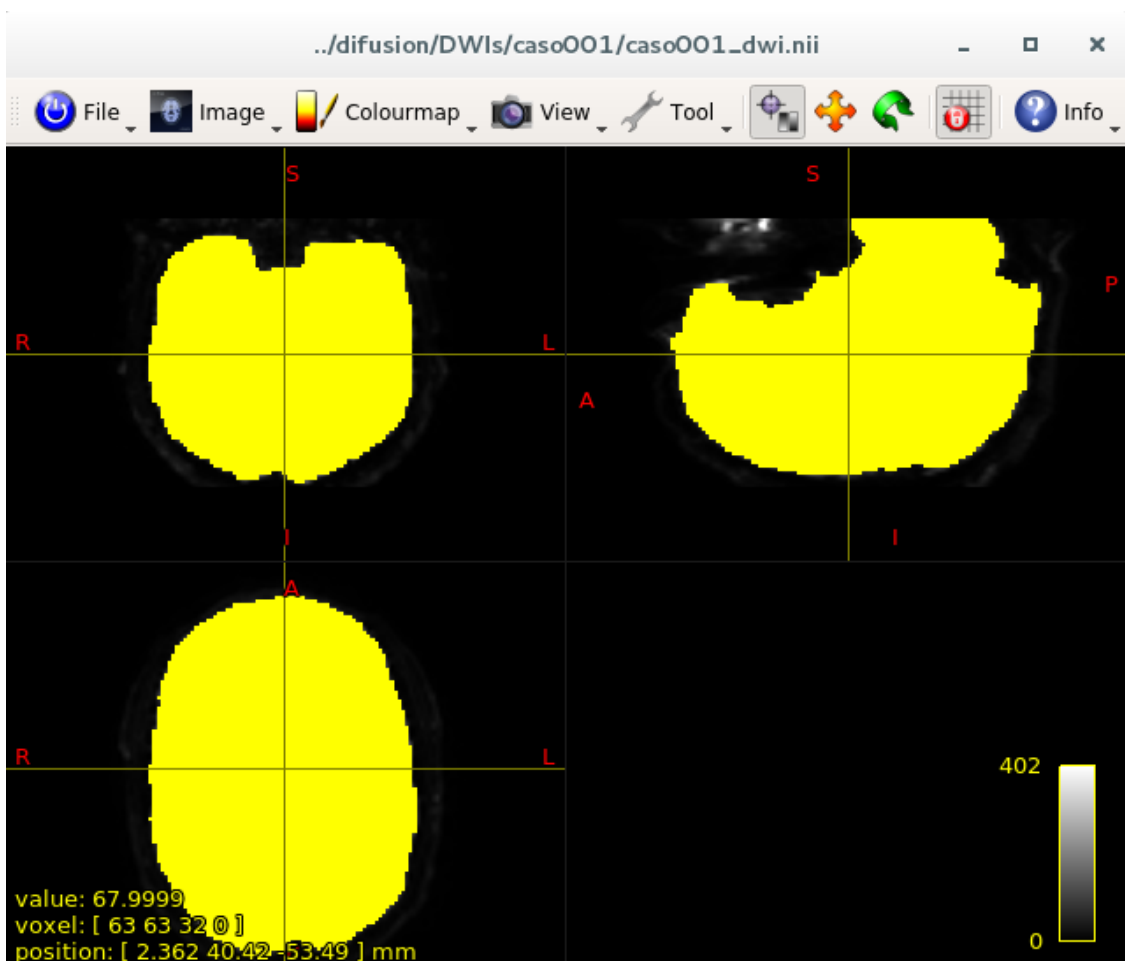


Figura 4-5: Contraposición de la máscara generada con la imagen de difusión mediante MRtrix.

4.5.3. Obtención de un mapa de diferentes tejidos cerebrales

Vamos a obtener una segmentación[23][25] de los diferentes tipos de tejido cerebral. Para ello, generamos la imagen 5TT(5-Tissue-Type, 5 Tipos de Tejido) mediante la parcelación de la segmentación cortical generada por Freesurfer previamente con el comando *5ttgen freesurfer*. Esta imagen 5TT es una imagen 4D, una imagen de punto flotante de 32 bits, donde la dimensión del cuarto eje es 5, que son los 5 volúmenes 3D en la imagen. Estos 5 volúmenes corresponden a los 5 tipos de tejidos. En todos los vóxeles del cerebro, la suma de estos 5 volúmenes debería ser 1.0, y fuera del cerebro debería ser 0. Estos 5 tipos de tejido deben aparecer en el siguiente orden para que los prioridades anatómicas se apliquen correctamente durante la tractografía:

1. Materia gris cortical.
2. Materia gris subcortical.
3. Materia blanca.
4. Líquido cefalorraquídeo(CSF).
5. Tejido patológico.

Los primeros cuatro son componentes cerebrales que se han separado en 4 tipos para que alterasen lo menos posible los resultados de la tractografía entre las distintas regiones a evaluar, ya que la intensidad de algunas estructuras como las pertenecientes a la materia gris subcortical está en algún lugar entre la materia gris cortical y la materia blanca y conviene por lo tanto separar estos tipos de tejido. El quinto puede ser usado opcionalmente para delinear manualmente las regiones del cerebro donde la arquitectura del tejido presente no está clara, y por lo tanto el tipo de prioridades anatómicas que se aplicarán son no conocidas. Para cada 'streamline' que entra en la región, ninguna prioridad anatómica será aplicada hasta que el 'streamline' exista en esa región o se detiene debido a algún otro criterio de terminación de 'streamlines'.

Este tipo de parcelación lo necesitamos para realizar una tractografía ACT, de la cual hablaremos más adelante.

Ahora vamos a explicar el comando *5ttgen freesurfer* de MRtrix[24] y las opciones que hemos usado:

5ttgen freesurfer [options] input output

input: La parcelación de la segmentación cortical generada por Freesurfer.

output: La imagen 5TT de salida

-nocrop: No recorta la imagen 5TT resultante para reducir su tamaño (mantiene las mismas dimensiones que la imagen de entrada).

i) Ahora generamos la segmentación de los 5 tipos de tejido:

5ttgen freesurfer -nocrop

../estructural/Freesurfer_segmentacion_cortical_resultado/caso001/caso001_cortical_seg.nii ../estructural/T1s_256x256x256/caso001/caso001_5tt.nii.gz

Para comprobar que hemos realizado bien la segmentación usamos el comando *5tt2vis* [26] el cual vamos a explicar:

5tt2vis [options] input output: generamos una imagen 3D para su visualización a partir de una imagen anatómica segmentada en el formato 5TT para ACT.

input: la imagen 4D segmentada en el formato 5TT

output: la imagen 3D de salida para su visualización

ii) Ahora escribimos el comando en la consola:

*5tt2vis ../estructural/T1s_256x256x256/caso001/caso001_5tt.nii.gz
../estructural/T1s_256x256x256/caso001/caso001_5tt_vis.nii.gz*

Para comprobar que la imagen 3D se ha realizado sin ningún problema visualizamos un ejemplo en la Figura 4-6 mediante MRtrix [15]:

mrview ../estructural/T1s_256x256x256/caso001/caso001_5tt_vis.nii.gz

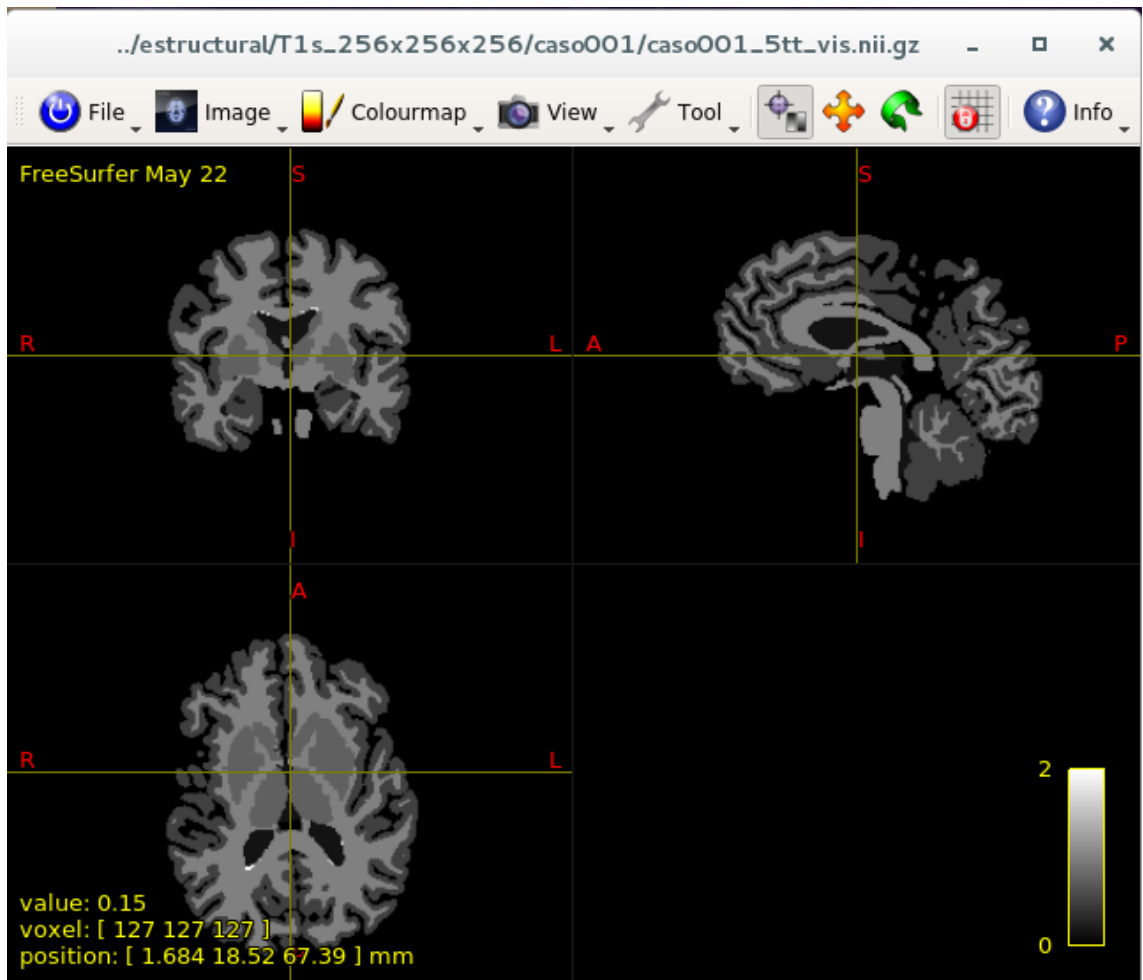


Figura 4-6: Visualización de una imagen 3D obtenida de una imagen 5TT

4.5.4. Estimación de la "Response Function" y de la "Fibre Orientation Distributions"

Vamos a estimar las "fibre orientation distributions"[27][28][29](FOD, distribuciones de la orientación de la fibra) presentes en cada voxel de la imagen de difusión usando una deconvolución esférica. Un paso obligatorio en la deconvolución esférica es derivar la "response function"(RF, función de respuesta) que es usada como el núcleo durante el paso de deconvolución. Para la materia blanca, esta es la señal esperada de un voxel que contiene un único haz de fibras coherentemente orientado. MRtrix prefiere derivar esta función directamente de los datos de la imagen de difusión en vez de usar una función de plantilla ad-hoc, y para ello se promedia típicamente la señal de difusión de un conjunto de voxels de fibra única determinados empíricamente. La FOD puede ser obtenida mediante la operación inversa, es decir, la deconvolución esférica de la señal ponderada por difusión por la función de respuesta. Los métodos de deconvolución son muy sensibles a los efectos de ruido, pero la robustez de la operación puede ser

mejorada enormemente por una restricción no negativa, conduciendo a una determinación robusta de las orientaciones de las fibras en un tiempo clínicamente aceptable.

Antes de iniciar la estimación creamos la carpeta donde almacenaremos la RF y la FOD:
mkdir ../difusion/MRTRIX_spherical_deconvolution/caso001

La RF es necesaria para estimar la FOD y para ello hemos usado el siguiente comando con una serie de opciones que vamos a explicar:

dwi2response tournier [options] input output: Utiliza un algoritmo iterativo para la selección de un voxel de fibra única y su estimación de la función de respuesta.

input: la imagen de difusión DWI

output: El fichero de texto de salida de la función de respuesta.

-mask: Proporciona una máscara inicial para la selección del voxel de respuesta.

-fslgrad bvecs bvals : Especifica las direcciones de gradiente y los valores de b usados en la adquisición en formato FSL bvecs/bvals.

i) Ahora vamos a estimar la función de respuesta escribiendo el comando siguiente anteriormente descrito[28]:

***dwi2response tournier -fslgrad ../difusion/b_values_61_direcciones.bvec
../difusion/b_values_61_direcciones.bval ../difusion/DWIs/caso001/caso001_dwi.nii
../difusion/MRTRIX_spherical_deconvolution/caso001/caso001_sph_dec_response -
mask ../difusion/MRTRIX_brain_mask/caso001/caso001_MRTRIX_brain_mask.nii***

Para comprobar que hemos estimado correctamente la función de respuesta, mostramos un ejemplo con el comando ***shview***[30] en la figura 4-7:
shview

../difusion/MRTRIX_spherical_deconvolution/caso001/caso001_sph_dec_response

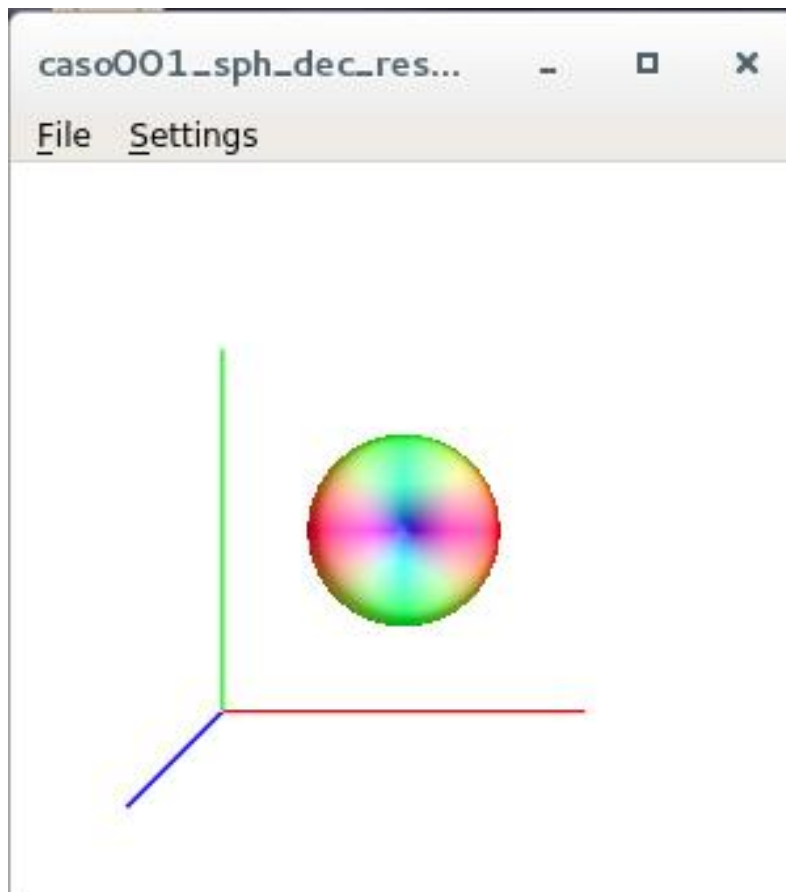


Figura 4-7: Visualización de la función de respuesta estimada

Ahora vamos a estimar la FOD, la cual es necesaria para realizar la tractografía más tarde. Para ello hemos utilizado el comando *dwi2fod csd* [27] con una serie de opciones que vamos a describir:

```
dwi2fod [ options ] csd dwi response odf [ response odf ... ]
```

dwi: Imagen de difusión.

response odf: Pares de respuesta de tejido de entrada y salida de imágenes FOD

-mask image: Sólo realizan cálculos dentro de la imagen de máscara cerebral binaria especificada.

-fslgrad bvecs bvals : Especifica las direcciones de gradiente y los valores de b usados en la adquisición en formato FSL bvecs/bvals.

Ahora escribimos en la consola el comando que hemos descrito:

```
dwi2fod csd -fslgrad ../difusion/b_values_61_direcciones.bvec  
../difusion/b_values_61_direcciones.bval ../difusion/DWIs/caso001/caso001_dwi.nii  
../difusion/MRTRIX_spherical_deconvolution/caso001/caso001_sph_dec_response  
../difusion/MRTRIX_spherical_deconvolution/caso001/caso001_sph_dec_output_fod.  
nii -mask  
../difusion/MRTRIX_brain_mask/caso001/caso001_MRTRIX_brain_mask.nii
```

Para ver si hemos realizado la FOD correctamente vamos a visualizar un ejemplo de la imagen de difusión junto con la FOD en la figura 4-8. Escribimos el siguiente comando de MRtrix[15] para ello:

```
mrview ../difusion/DWIs/caso001/caso001_dwi.nii -odf.load_sh  
../difusion/MRTRIX_spherical_deconvolution/caso001/caso001_sph_dec_output_fod.  
nii
```

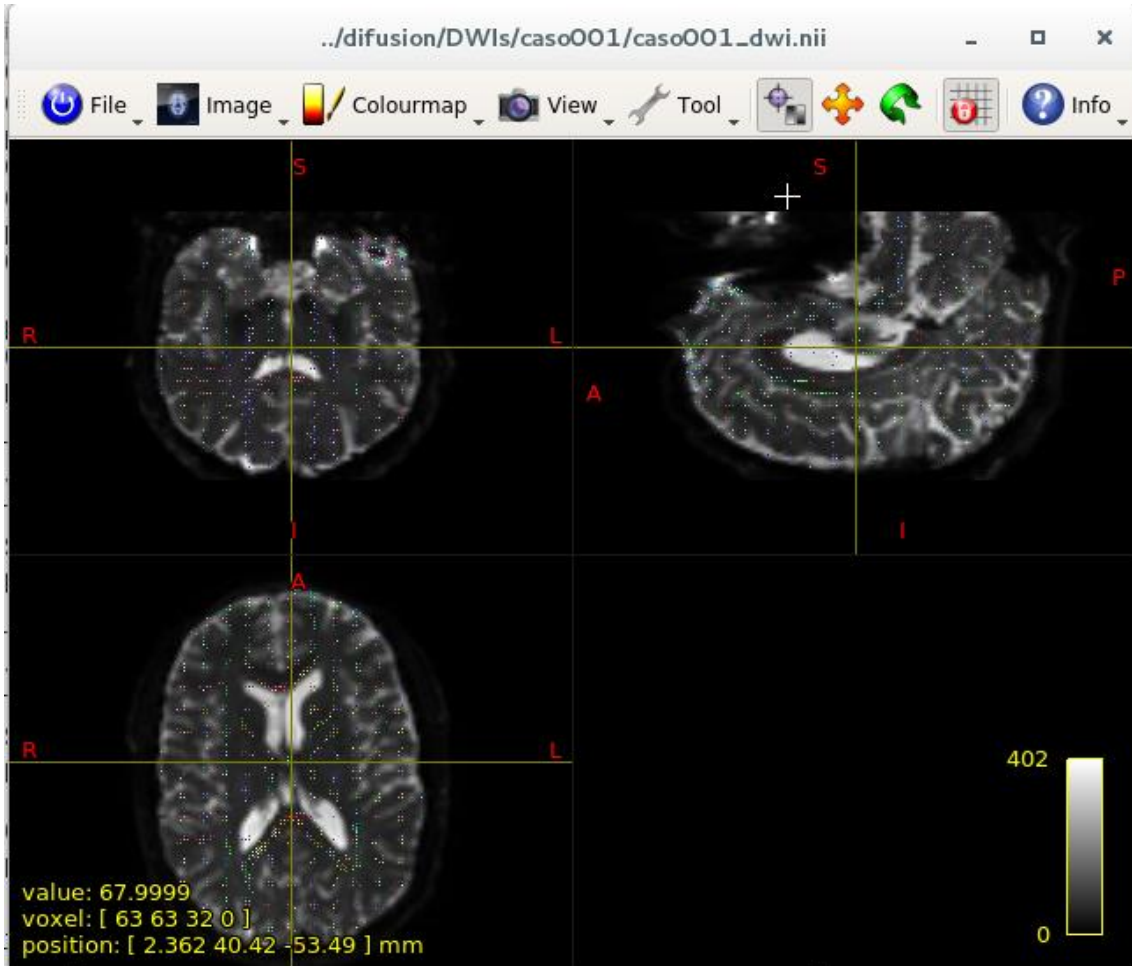


Figura 4-8: Visualización de la FOD en superposición con la imagen de difusión.

4.5.5. Obtención de tensores y medidas escalares asociadas

Vamos a obtener una serie de tensores y volúmenes escalares asociados. Se usa la utilidad *dtifit* [31], de FSL. Previamente necesitamos una máscara del cerebro, usamos la calculada anteriormente con *dwi2mask* de MRtrix que calculamos a partir de los DWIs. Necesitamos también los b-values y vectores de los gradientes. Los que crea DWIConvert tienen un problema, así que usamos unos que tenemos preparados, y que almacenamos en el directorio difusión.

i) Primero creamos el directorio donde almacenaremos los tensores y las medidas escalares.

```
mkdir ../difusion/tensores_con_FSL/caso001
```

ii) Ahora vamos a describir el comando a utilizar y sus opciones:

DTIFIT: Se ajusta a un modelo de tensor de difusión en cada voxel.

- k Imagen de difusión
- o Nombre de base de salida
- m fichero que contiene la máscara del cerebro
- r fichero que contiene los gradientes
- b fichero que contiene los b-values

Y ahora escribimos en la consola:

```
dtifit -k ../difusion/DWIs/caso001/caso001_dwi.nii -o  
../difusion/tensores_con_FSL/caso001/caso001 -m  
../difusion/MRTRIX_brain_mask/caso001/caso001_MRTRIX_brain_mask.nii -r  
../difusion/b_values_61_direcciones.bvec -b ../difusion/b_values_61_direcciones.bval
```

De entre todas las medidas escalares y tensores obtenidos, vamos a utilizar la llamada ***caso001_FA*** que representa la FA en las regiones de la imagen de difusión y que vemos en la figura 4-9 mediante MRtrix[15] con el siguiente comando:

```
mrview ../difusion/tensores_con_FSL/caso001/caso001_FA.nii.gz
```

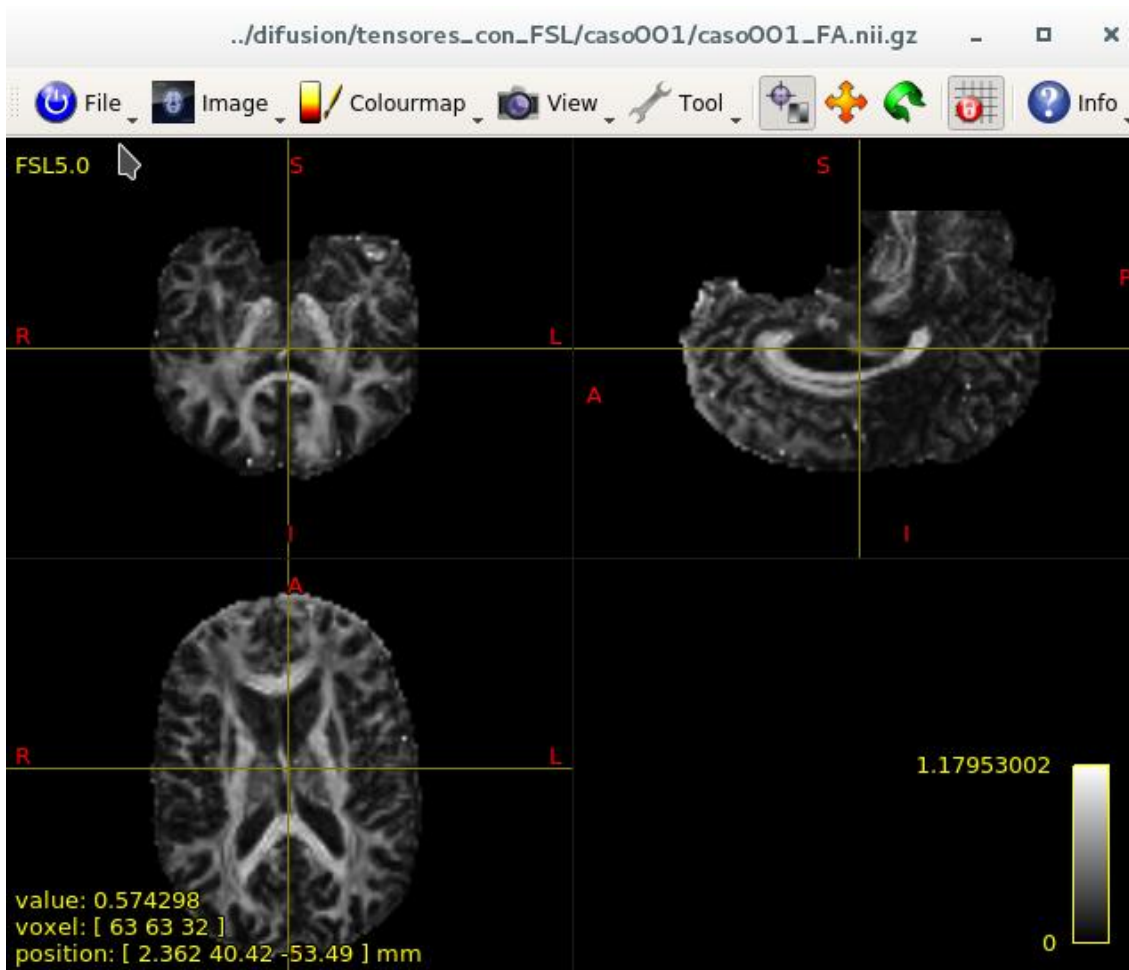


Figura 4-9: Visualización con MRtrix de la FA en cada región de la imagen de difusión

4.5.6. Adaptar la segmentación cortical para calcular la conectómica

Hay que adaptar la segmentación cortical y parcelación realizada por Freesurfer en pasos anteriores a lo que necesitamos para el cálculo de la conectómica[8]. Se trata de cumplir dos requisitos:

- a) Que tenga valores consecutivos (lo necesita la utilidad de conectómica de MRtrix)
- b) Que agrupe las zonas corticales que queremos. Esto es dependiente de la aplicación concreta(estudio de esquizofrenia u otras cosas)

Las parcelaciones[32] se proporcionan típicamente como una imagen entera, donde cada número entero corresponde a un nodo particular, y los voxels donde no hay ningún nodo de la parcelación tienen un valor de 0. Sin embargo, la mayoría de los esquemas de la parcelación, los valores utilizados para los nodos no aumentan monótonicamente a partir de 1, sino que siguen una distribución no lineal. Entonces se proporciona un archivo de texto (o 'tabla de búsqueda') que vincula índices de nodos a nombres de estructura. Sin embargo, esto es indeseable para la construcción de conectores y sería preferible que los índices de nodo aumenten monótonicamente a partir de 1, de manera que cada valor entero corresponde a una posición de fila / columna en la matriz de conexión. Al hacer este cambio de tabla nos ahorramos[33] generar un conectoma enorme y mayormente lleno de ceros. Por lo tanto, lo que preferiríamos es mapear el

índice entero único de esta estructura a un índice de fila / columna particular del conectoma. Esto debe hacerse de tal manera que todas las estructuras de interés tengan un valor entero único entre 1 y N, donde N es el número de nodos en el conectoma.

Esta funcionalidad se proporciona con el comando *labelconvert*[32]. Toma como entrada una imagen de parcelación que ha sido proporcionada por algún otro paquete de software(en nuestro caso Freesurfer), y convierte los índices de la etiqueta. Esto se hace de modo que el código que genera realmente el conectoma puede ser 'tonto y ciego', es decir, que los valores enteros en los extremos de la 'streamline' corresponden a la fila y columna de la matriz del conectoma que debe incrementarse. Además, este diseño de cadenas de procesamiento proporciona flexibilidad tanto en lo que se refiere a la fuente de los datos de parcelación como a la forma en que el usuario desea personalizar el diseño de su conexión.

Comando utilizado:

labelconvert [options] path_in lut_in lut_out image_out : Convierte una imagen[34] de nodo del conectoma de una tabla de búsqueda a otra. El uso típico es convertir una imagen de parcelación proporcionada por algún otro software, basado en la tabla de búsqueda proporcionada por ese software, para ajustarse a una nueva tabla de consulta, particularmente aquella en la que los índices de nodo aumentan de 1, en preparación para la construcción del conectoma.

1. path_in: imagen de entrada.
2. lut_in: tabla de búsqueda del conectoma de la imagen de entrada.
3. lut_out: tabla de búsqueda nueva del conectoma de la imagen de salida.
4. image_out: imagen de salida.

Ahora vamos a describir las tablas de búsqueda[33]:

FreeSurferColorLUT.txt: Esta es la tabla de búsqueda de la imagen de entrada. Nos da un nombre significativo para esta estructura basada en el índice entero. También nos da una cierta información de color, no necesaria ahora.

fs_default.txt: Nuestra meta entonces es determinar un nuevo índice entero para esta estructura, que determinará la fila / columna de nuestra matriz del conectoma a la cual esta estructura corresponde. Esto se trata asignando los índices de estructura de esta tabla de búsqueda a una nueva tabla de búsqueda, y para ello usaremos la tabla de búsqueda predeterminada de MRtrix fs_default.txt.

El comando *labelconvert* funciona de la siguiente manera en el orden dado por los pasos:

1. Lee el valor entero de cada voxel de la imagen de entrada.
2. Convierte los valores enteros a un 'string'(cadena de texto), basado en la tabla de búsqueda de entrada (FreeSurferColorLUT.txt) .
3. Encontramos el 'string' en la tabla de búsqueda nueva de salida (fs_default.txt) .
4. Escribimos el índice entero almacenado en la tabla de búsqueda de salida para esta estructura en el voxel de la imagen de salida.

Escribimos en la consola:

labelconvert

```
../estructural/Freesurfer_segmentacion_cortical_resultado/caso001/caso001_cortical_seg.nii $FREESURFER_HOME/FreeSurferColorLUT.txt /opt/shared/MRtrix/mrtrix3/src/connectome/tables/fs_default.txt
```

`../estructural/Freesurfer_segmentacion_cortical_resultado/caso001/caso001_cortical_seg_lut.nii`

Para comprobar que la adaptación de la segmentación cortical para calcular la conectómica se ha realizado sin ningún problema, visualizamos un ejemplo en la Figura 4-10 mediante MRtrix [15] :

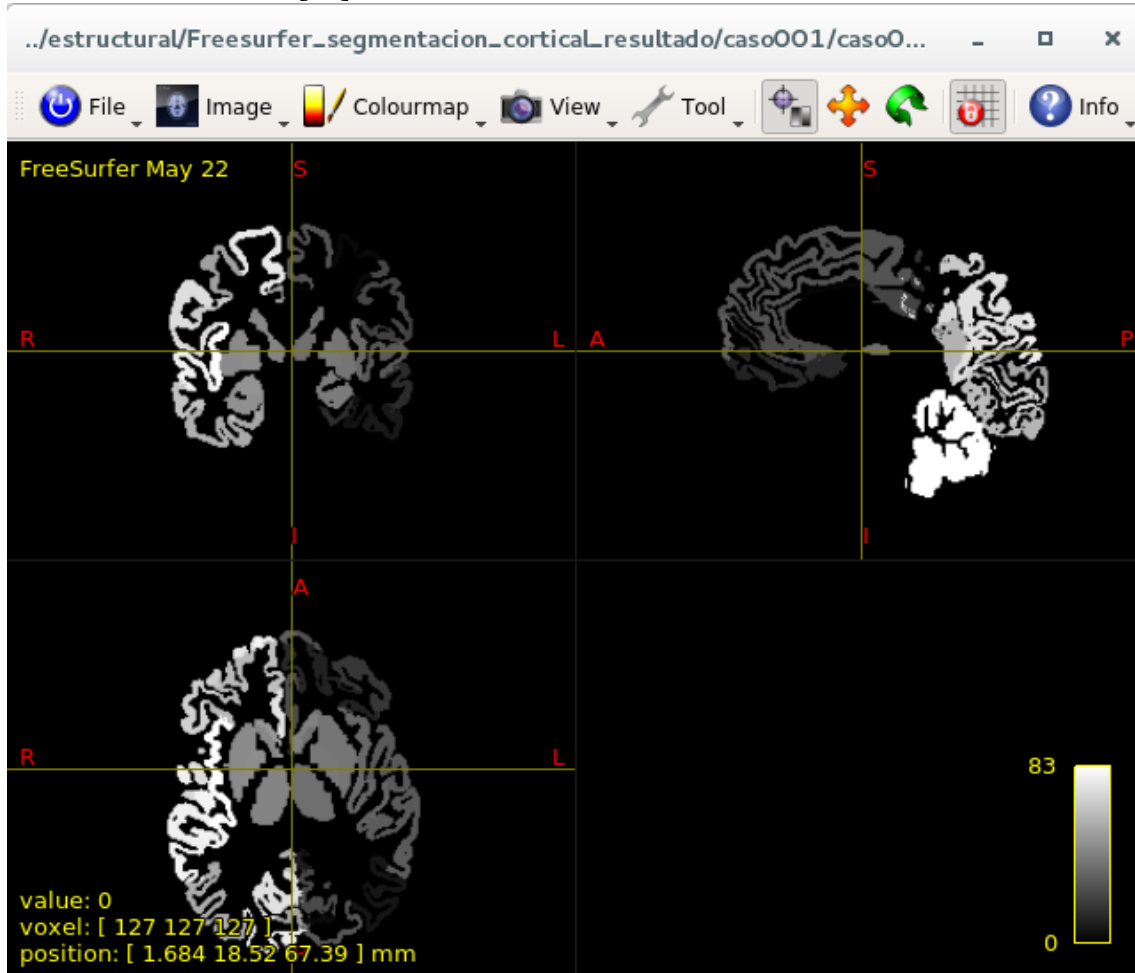


Figura 4-10: Adaptación de la segmentación cortical mediante un cambio de su 'tabla de búsqueda'

4.5.7. Registrado para proyectar la segmentación cortical sobre el volumen tensorial.

Se realiza este registrado entre la imagen T1 y la FA[8] para proyectar la segmentación cortical sobre el volumen tensorial, que es el espacio donde esta calculada la tractografía. Tenemos que hacer este registrado para luego aplicar la deformación resultante a la segmentación cortical obtenida con Freesurfer. Esto lo hacemos con la herramienta flirt de FSL.

Detalles importantes: Los DWIs y cantidades derivadas que se crean a partir de los DWI.nii están mal colocados. Esto hace que las orientaciones del T1 y de la FA sean

muy diferentes. Por eso, es necesario considerar ángulos de búsqueda en el registrado de entre -180 y 180 grados (por defecto se usa 90). Si no, el registrado no saldrá bien.

En general[8][35], es mejor al hacer un registrado tomar como referencia la imagen de más calidad. En este caso esa imagen es el T1. Sin embargo, luego queremos deformar la segmentación cortical en el sentido inverso, así que necesitamos invertir la transformación resultante (hay un comando de flirt para ello).

Como paso previo, es muy conveniente quitar lo que no sea cerebro del T1, puesto que en la FA se ha quitado lo que no es cerebro y así el registrado funcionará mejor. (Realizado ya con la herramienta bet en 4.4.2)

i) Primero creamos la carpeta donde se almacenará el registrado:

```
mkdir ../difusion/MRTRIX_conectomica/caso001
```

ii) Ahora para realizar el registrado usamos una serie de comandos de FSL con sus respectivas opciones:

flirt: Es el principal programa[35] que realiza el registro afín. Las opciones principales son: entrada (-in) y un volumen de referencia (-ref). La transformación afín calculada que registra la entrada a la referencia se guarda como una matriz afín 4x4 (-omat). El volumen de salida (-out) donde la transformación se aplica al volumen de entrada para alinearla con el volumen de referencia. Además, FLIRT también se puede utilizar para aplicar una transformación guardada a un volumen (-applyxfm, -init y -out) o para aplicar una transformación que alinea las coordenadas NIFTI mm (-applyxfm, -userqform y -out, pero no -init). Para estos usos el volumen de referencia debe ser especificado ya que establece las dimensiones de voxel y de imagen del volumen resultante.

-dof: número de transformaciones dofs (Degree of freedom, grados de libertad) (por defecto 12).

-searchrx <min_angle><max_angle> (ángulos en grados: por defecto es -90 90)

-searchry <min_angle><max_angle> (ángulos en grados: por defecto es -90 90)

-searchrz <min_angle><max_angle> (ángulos en grados: por defecto es -90 90)

-init <matrix-filename> (matriz afín 4x4 de entrada)

-interp {trilinear,nearestneighbour,sinc,spline}: Interpolation - selecciona el método de interpolación usado en la transformación final

-applyxfm (aplica transformadas - requiere -init)

convert_xfm [36]: Es una utilidad que se utiliza para convertir entre diferentes formatos de ficheros de transformación. Puede leer y escribir matrices ascii 4x4. Además, se puede utilizar para concatenar dos transformaciones (usando -concat con la segunda transformación) o para encontrar la transformación inversa (usando -inverse).

Vamos a proceder con los siguientes pasos al registrado[8]:

i) Realizamos la transformación afín que registra la FA al volumen de referencia T1 con las opciones de 12 grados de libertad, que es el comportamiento por defecto y recomendado por fsl. Esta transformación la almacenamos en una matriz con formato .mat como vemos:

```
flirt -in ../difusion/tensores_con_FSL/caso001/caso001_FA -ref  
../estructural/T1s_256x256x256/caso001/caso001_T1_256_masked -out  
../difusion/MRTRIX_conectomica/caso001/caso001_FA_over_T1 -omat  
../difusion/MRTRIX_conectomica/caso001/caso001_transf_FA_over_T1.mat -dof 12  
-searchrx -180 180 -searchry -180 180 -searchrz -180 180
```

ii) Ahora hallamos la inversa de la matriz que almacena la transformación realizada en el paso anterior, con el fin de obtener una matriz con una transformación afín para que realice este registrado entre la imagen T1 y la FA.

```
convert_xfm -omat
```

```
../difusion/MRTRIX_conectomica/caso001/caso001_transf_inv_FA_over_T1.mat -  
inverse ../difusion/MRTRIX_conectomica/caso001/caso001_transf_FA_over_T1.mat
```

iii) Realizado el registrado entre la imagen T1 y la FA, se proyecta la segmentación cortical sobre el volumen tensorial. Utilizamos un método de interpolación nearestneighbour [36]:

```
flirt -in
```

```
../estructural/Freesurfer_segmentacion_cortical_resultado/caso001/caso001_cortical  
_seg_lut.nii -ref ../difusion/tensores_con_FSL/caso001/caso001_FA -out  
../difusion/MRTRIX_conectomica/caso001/caso001_cortical_seg_lut_warped -init  
../difusion/MRTRIX_conectomica/caso001/caso001_transf_inv_FA_over_T1.mat -  
interp nearestneighbour -applyxfm
```

iv) Ahora se proyecta la segmentación cortical en formato 5TT sobre el volumen tensorial. Utilizamos un método de interpolación nearestneighbour [36]:

```
flirt -in ../estructural/T1s_256x256x256/caso001/caso001_5tt.nii.gz -ref  
../difusion/tensores_con_FSL/caso001/caso001_FA -out  
../estructural/T1s_256x256x256/caso001/caso001_5tt_warped -init  
../difusion/MRTRIX_conectomica/caso001/caso001_transf_inv_FA_over_T1.mat -  
interp nearestneighbour -applyxfm
```

Vamos a visualizar en las Figuras 4-11 y 4-12 un ejemplo del resultado de realizar la proyección de una segmentación cortical en el formato por defecto y otro en el formato 5TT sobre la FA, respectivamente. Para ello mostramos superpuestos la FA con la segmentación cortical resultante en ambos casos mediante MRtrix[15]:

```
mrview ../difusion/tensores_con_FSL/caso001/caso001_FA -overlay.load  
../difusion/MRTRIX_conectomica/caso001/caso001_cortical_seg_lut_warped
```

```
mrview ../difusion/tensores_con_FSL/caso001/caso001_FA -overlay.load  
../estructural/T1s_256x256x256/caso001/caso001_5tt_warped
```

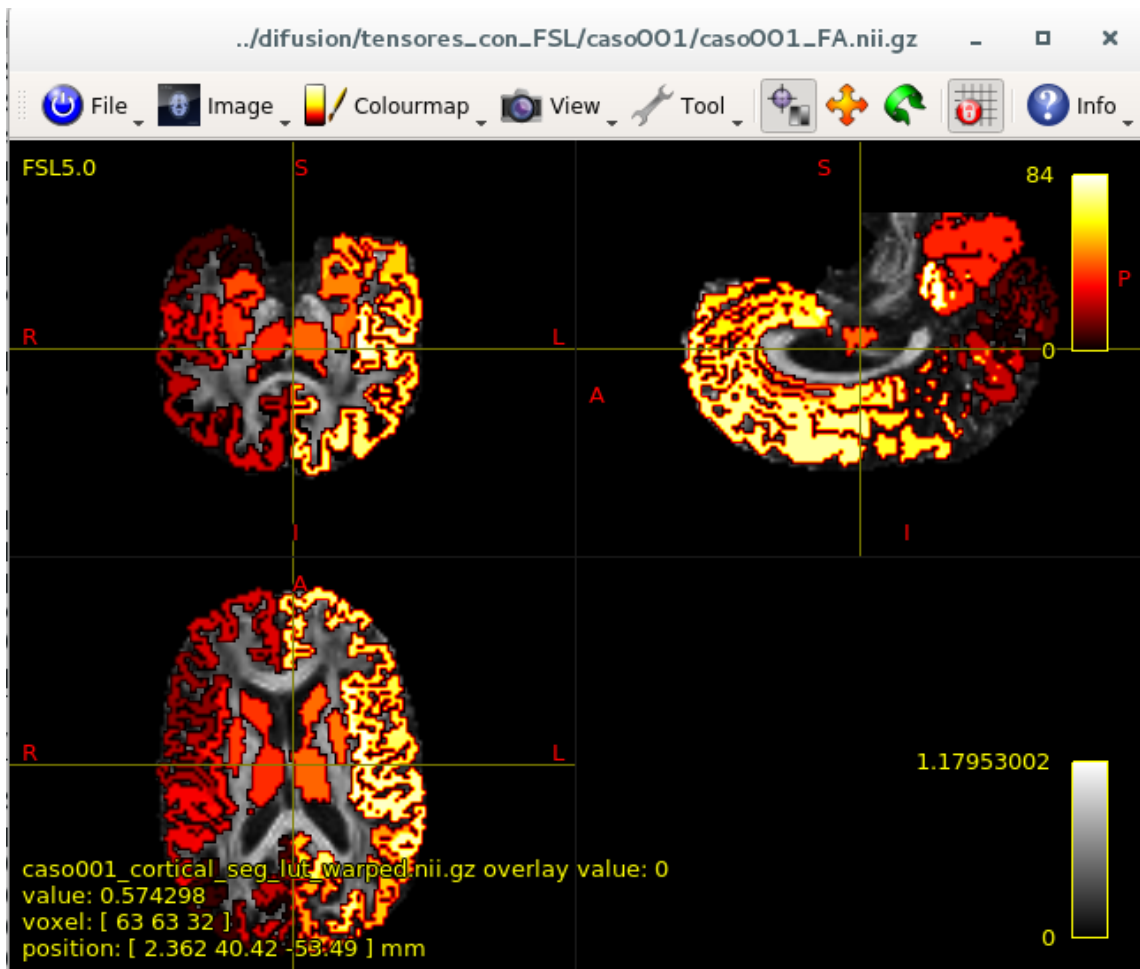


Figura 4-11: Visualización superpuesta de la FA(negro y gris) con la segmentación cortical(en colores rojo y naranja) producida por Freesurfer mediante mrview



Figura 4-12: Visualización superpuesta de la FA (negro y gris) con la segmentación cortical (en colores rojo y naranja) en formato 5TT mediante mrview.

4.5.8. Realización de una tractografía completa del cerebro

Usamos "Anatomically Constrained Tractography" (ACT, tractografía anatómicamente restringida). A medida que se propaga una 'streamline'[25], la posición actual de la 'streamline' se asigna a una posición precisa en la imagen de difusión relevante (utilizando el inverso de la transformación de imagen) y los valores apropiados se dibujan utilizando por ejemplo una interpolación tri-lineal para adquirir una estimación local del modelo de difusión. Por lo tanto, la única información típicamente disponible para un algoritmo 'streamline' para determinar la dirección apropiada para el seguimiento es este modelo de difusión interpolado localmente y la tangente de la 'streamline' en el paso anterior. Estos algoritmos tienen ahora una fuente adicional de información: a saber, las fracciones de volumen parcial estimadas interpoladas localmente de los cuatro tipos de tejido constituyentes. La posición actual de la 'streamline' se correlaciona con una localización en la imagen anatómica segmentada (utilizando la inversa de la transformación de dicha imagen), y un valor interpolado tri-lineal extraído de esta imagen, para obtener una estimación del tejido constituyente en el punto actual.

Nuestra tarea entonces es utilizar esta información adicional tan eficazmente como sea posible, usando la información anatómica prioritaria que es sana desde una perspectiva

biológica. No influenciarnos la dirección de seguimiento de ninguna manera, ya que el modelo de difusión proporciona esta información. Influenciamos las 'streamlines' en dos maneras. Los mecanismos por los cuales las 'streamlines' terminan y los criterios por los cuales las 'streamlines' son aceptadas o rechazadas en base a su plausibilidad biológica. Las estructuras que se están reconstruyendo usando la tractografía de 'streamlines' son los haces axonales neuronales largos de la sustancia blanca. Estas estructuras conectan áreas de procesamiento espacialmente distantes del sistema nervioso central (la materia gris) donde realizan la sinapsis con otras neuronas y usan la materia blanca como medio de conducción para proporcionar esta conexión. Como tal, debemos esperar que la mayoría de los puntos extremos de estas estructuras se produzcan dentro de la materia gris (cortical o sub-cortical). Algunos pueden proyectarse a la columna vertebral (como el tracto cortico-espinal), pero nunca deben terminar en la sustancia blanca normal. Además, los axones no entran en las regiones que contiene líquido cefalorraquídeo del cerebro. En resumen, básicamente, lo que hace es aplicar prioridades a la tractografía para que acabe donde se espera que acabe (básicamente en la sustancia gris o en la materia blanca subcortical). Para hacer esto, hace falta una segmentación de esos diferentes tipos de tejido en el cerebro en un formato llamado 5TT ya explicado.

Se usa MRtrix en la línea de comandos. Aquí hay un parámetro crítico, que es el número de fibras que se generan (opción -number). Vamos a usar 2000000, un valor aceptable ya que influye tanto en el tiempo que tardará en ejecutarse como en el tamaño del fichero resultante (y la dificultad para manejarlo después).

MRtrix genera un fichero con formato .tck. Para poder usarlo con otros programas, lo convertimos después a .vtk como Slicer (aunque no tendrá información sobre FA y por lo tanto las fibras saldrán de una manera poco vistosa).

El comando que hemos utilizado se llama *tckgen* [37] que vamos a describir junto con una serie de opciones:

tckgen [options] source tracks: realiza una tractografía determinística

source: imagen que contiene los datos de entrada

tracks: fichero de salida que contiene los tractos generados.

-seed_image image: 'Siembra' las 'streamlines' de manera aleatoria dentro de la máscara de la imagen de difusión.

-number: trazos establecidos a número deseado de trazos. El programa continuará generando trazos hasta que el número de trazos haya sido seleccionado y escrito al fichero de salida.

-act image: Utiliza la tractografía anatómicamente restringida (ACT) durante el trazado. La imagen proporcionada debe estar en el formato 5TT.

i) Primero creamos la carpeta donde se almacenará el fichero de salida dado por la tractografía:

```
mkdir ../difusion/MRTRIX_whole_brain_tractography/caso001
```

ii) Realizamos la tractografía ACT escribiendo en la consola:

```
tckgen -act ../estructural/T1s_256x256x256/caso001/caso001_5tt_warped.nii.gz  
../difusion/MRTRIX_spherical_deconvolution/caso001/caso001_sph_dec_output_fod.  
nii  
../difusion/MRTRIX_whole_brain_tractography/caso001/caso001_act_2000000_fibra  
s.tck -seed_image  
../difusion/MRTRIX_brain_mask/caso001/caso001_MRTRIX_brain_mask.nii -  
number 2000000
```

iii) Ahora con el comando *tckconvert* realizamos la conversión de datos de trazas de un formato a otro, así como aplicar transformaciones entre convenciones espaciales. Para ello escribimos:

tckconvert

```
../difusion/MRTRIX_whole_brain_tractography/caso001/caso001_act_2000000_fibras.tck
```

```
../difusion/MRTRIX_whole_brain_tractography/caso001/caso001_act_2000000_fibras.vtk
```

iv) Vamos a usar el comando *tckinfo* para chequear por si acaso:

tckinfo

```
../difusion/MRTRIX_whole_brain_tractography/caso001/caso001_act_2000000_fibras.tck
```

Vamos a visualizar un ejemplo de una tractografía realizada superpuesta con la imagen de difusión mediante MRtrix[15] en la figura 4-13:

```
mrview ../difusion/DWIs/caso001/caso001_dwi.nii -tractography.load
```

```
../difusion/MRTRIX_whole_brain_tractography/caso001/caso001_act_2000000_fibras.tck
```

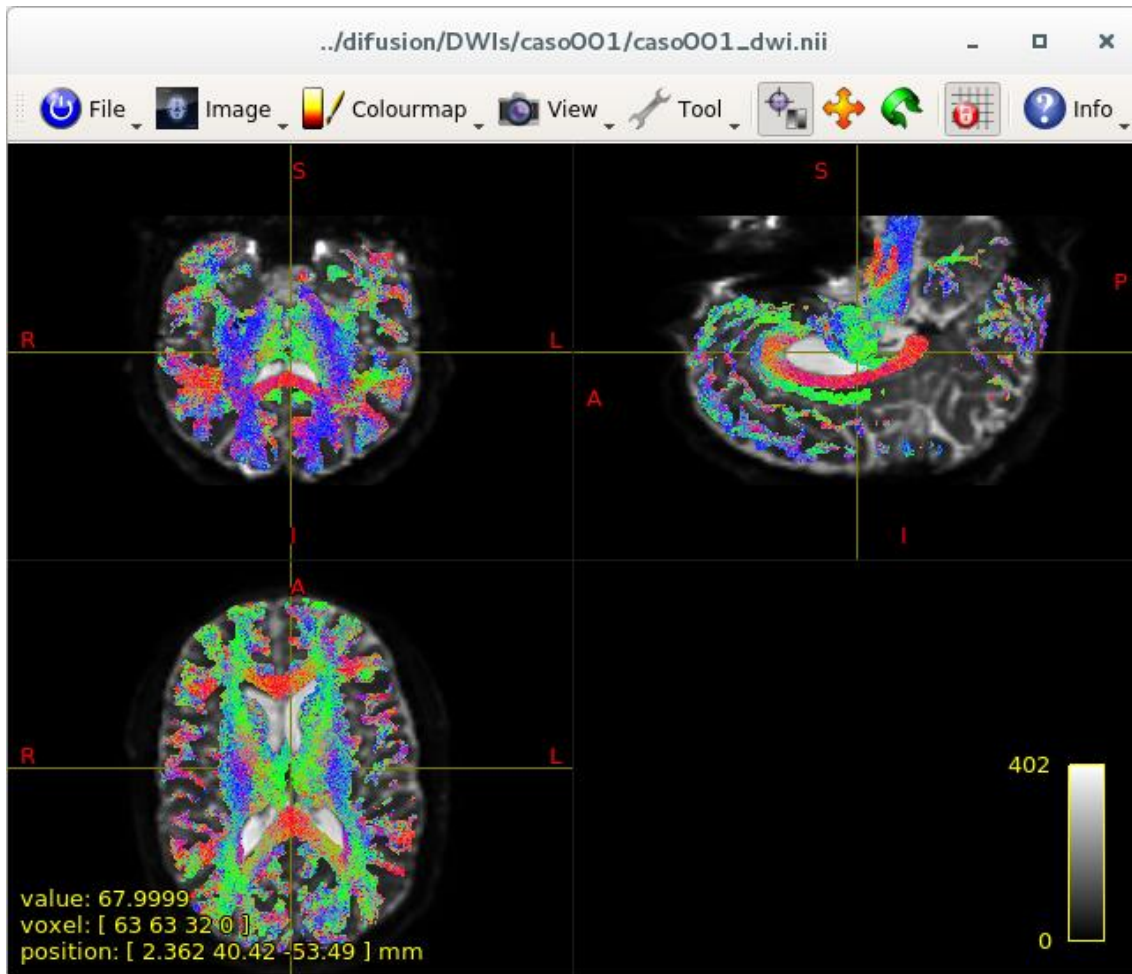


Figura 4-13: Visualización de una tractografía superpuesta con la imagen de difusión mediante mrview.

4.5.9. Obtención de la matriz de conectividad

Un factor[32] en la producción del conectoma estructural comúnmente pasado por alto en la literatura es el mecanismo utilizado para asignar 'streamlines' a las parcelas de materia gris. Si se hace incorrectamente esto puede tener una gran influencia en los conectomas resultantes. Este es un aspecto en el que realmente brilla la Tractografía Anatómicamente Restringida (ACT) porque las 'streamlines' sólo puede terminar precisamente en la materia gris - interfaz de la sustancia blanca, dentro de la materia gris subcortical, o en el borde inferior de la imagen donde esta asignación se hace relativamente trivial.

Para generar[38] una matriz de conectividad desde un tractograma, basado en una imagen de parcelación de nodos, vamos a usar la utilidad *tck2connectome* de MRtrix, que nos proporciona un fichero .csv con las matriz de conexiones. Vamos a describir el comando con una serie de opciones

tck2connectome [options] tracks_in nodes_in connectome_out

- tracks_in: fichero de trazos de entrada .
- nodes_in: imagen de entrada que contiene la parcelación de los nodos.
- connectome_out: el fichero de salida .csv con la matriz de conexiones.

Escribimos en la consola:

tck2connectome

```
../difusion/MRTRIX_whole_brain_tractography/caso001/caso001_act_2000000_fibra  
s.tck
```

```
../difusion/MRTRIX_conectomica/caso001/caso001_cortical_seg_lut_warped.nii.gz  
../difusion/MRTRIX_conectomica/caso001/caso001_conectomica_fiber_count.csv
```

Ahora vamos a ver un ejemplo de la representación de una matriz de conectómica que contiene el número de fibras en cada nodo o región en contraposición con la segmentación cortical realizada después de la parcelación del apartado 4.5.6 y añadiendo también la imagen de la segmentación con la parcelación del 4.5.6 para inicializar la herramienta del conectoma con MRtrix[15] en la figura 4-14:

mrview

```
../estructural/Freesurfer_segmentacion_cortical_resultado/caso001/caso001_cortical  
_seg_lut.nii -connectome.init
```

```
../estructural/Freesurfer_segmentacion_cortical_resultado/caso001/caso001_cortical  
_seg_lut.nii -connectome.load
```

```
../difusion/MRTRIX_conectomica/caso001/caso001_conectomica_fiber_count.csv
```

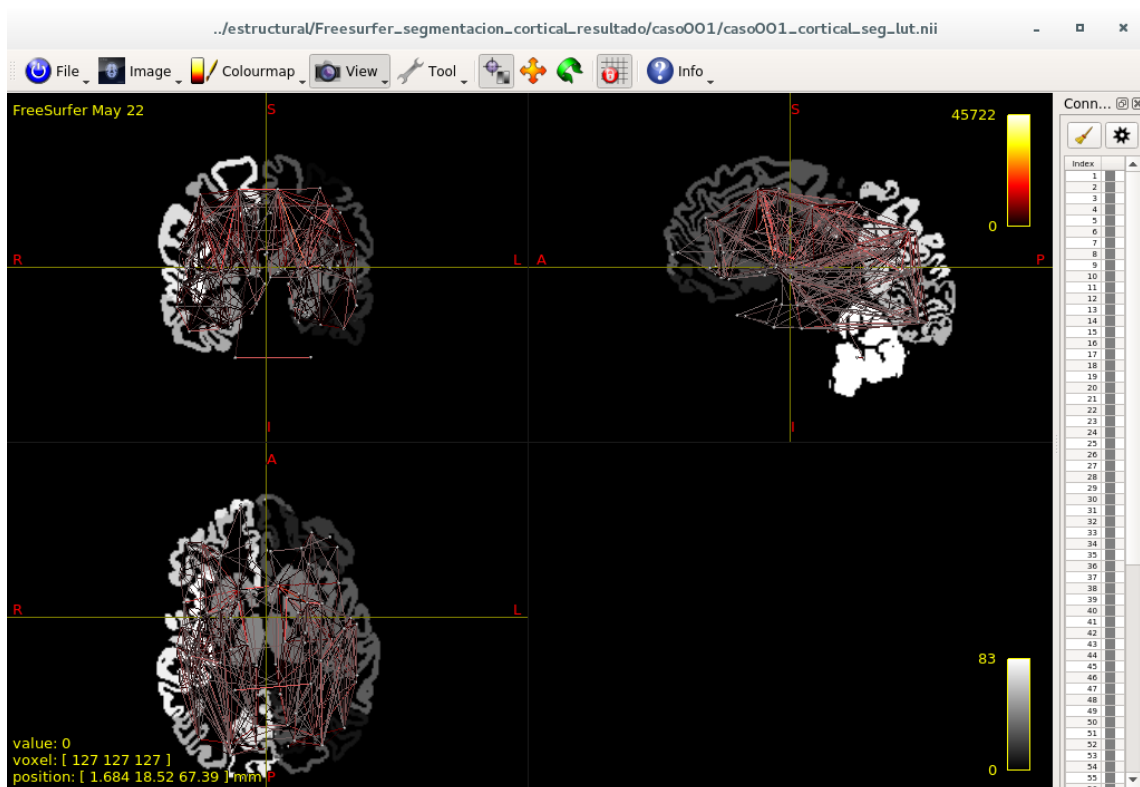


Figura 4-14: Visualización del conectoma en contraposición con la segmentación cortical obtenida con la parcelación del apartado 3.5.6 mediante mrview

Si queremos usar otras medidas de conectividad que no sean el número de fibras, tenemos que modificar las opciones de uso de *tck2connectome* y utilizando un nuevo

comando llamado *tcksample*[39] como paso intermedio. Por ejemplo, si quisiéramos la FA media. Se hace con los dos siguientes comandos:

tcksample

```
../difusion/MRTRIX_whole_brain_tractography/caso001/caso001_act_2000000_fibra  
s.tck ../difusion/tensores_con_FSL/caso001/caso001_FA.nii.gz  
../difusion/MRTRIX_conectomica/caso001/caso001_mean_scalars.csv -stat_tck mean
```

tck2connectome

```
../difusion/MRTRIX_whole_brain_tractography/caso001/caso001_act_2000000_fibra  
s.tck  
../difusion/MRTRIX_conectomica/caso001/caso001_cortical_seg_lut_warped.nii.gz  
../difusion/MRTRIX_conectomica/caso001/caso001_conectomica_mean_FA.csv -  
scale_file ../difusion/MRTRIX_conectomica/caso001/caso001_mean_scalars.csv -  
stat_edge mean
```

Con este último procedimiento tenemos un fichero .csv en el que contiene la FA media entre dos nodos o regiones.

Capítulo 5. Método de identificación biométrica

5.1. Introducción

El método de verificación empleado en este estudio se va a basar en tres partes. Primero vamos a describir los datos que tenemos para aplicar nuestro método de verificación así como una serie de parámetros, los cuales tendrán un valor numérico, y que servirán para que nuestro algoritmo se realice de una manera específica. Segundo, describiremos detalladamente el esquema de entrenamiento y tercero, explicaremos el esquema de validación.

5.2. Datos de partida

Explicamos en primer lugar la naturaleza de los datos de los que partimos para realizar la identificación. Para cada adquisición (también llamada caso), disponemos de las matrices de conectómica resultantes del procesado de las adquisiciones de resonancia magnética descrito en el capítulo anterior. Para cada caso tenemos dos matrices de conectividad:

- Matriz de conectividad en la que la descripción de cada conexión consiste en el número de "fibras" o "streamlines" encontrados conectando a cada par de regiones corticales considerado. Esta medida recibirá en adelante el nombre de FC (FiberCount).
- Matriz de conectividad en la que la descripción de cada conexión se realiza a través del valor medio de la FA (Fractional Anisotropy) a lo largo de la región volumétrica que ocupan las fibras encontradas entre cada par de regiones.

Puesto que la segmentación cortical realizada dentro del esquema de procesado de datos divide la corteza cerebral en 84 regiones, las matrices de conectividad (tanto la de FC como la de FA) tienen un tamaño de 84x84. Estas matrices de conectividad son simétricas, puesto que las medidas de conectividad también lo son. Sólo tiene por lo tanto considerar la parte superior (o inferior) de la matriz. Además, tiene sentido considerar la conectividad de una región cortical consigo misma (la diagonal de la matriz), porque sucede habitualmente que a partir de una región cortical existen fibras que se adentran en la sustancia blanca para luego regresar a la misma región cortical. Debido a estos dos factores, el número de valores de interés que se pueden considerar a partir de la matriz de conectividad es de 3570 elementos: $(84 \cdot 84 - 84) / 2 + 84$.

5.3. Método propuesto

El método de identificación propuesto puede dividirse en las siguientes etapas:

Selección de características

El número inicial de características es de 3570 si solamente se considera la FA o el FC, o bien 7140 si se consideran ambos descriptores. Este número de características es excesivo por dos motivos. En primer lugar, porque aumenta demasiado la dimensionalidad del problema teniendo en cuenta el tamaño del conjunto de datos de

entrenamiento disponibles. En segundo lugar, porque existen muchas conexiones cerebrales para las que el número de "streamlines" es muy pequeño en muchos de los casos. Para estas conexiones, los valores de FC y FA son muy variables entre casos, lo que probablemente se traduzca en muy poca fiabilidad a la hora de ser empleadas para el método de verificación. Por estos motivos, se reduce el número de características a emplear, introduciendo un parámetro, N, que es el número de fibras para cada conexión que el sistema exige. De esta manera, al iniciar el proceso de entrenamiento, una cierta conexión solamente se considera válida (y es por lo tanto seleccionada) si se cumple que para todos los casos de entrenamiento el número de "streamlines" para todos los casos al menos igual a N. Por otra parte, el sistema permite el empleo como características solamente de la FC, solamente de la FA, o de ambas conjuntamente. Como se verá con detalle en el capítulo siguiente, con N=500 el número de conexiones válidas se reduce a aproximadamente 200 (el número exacto depende de cada conjunto de entrenamiento).

Comparaciones individuales

En entrenamiento, es conocido si un cierto caso es un caso individual o bien se trata de un caso "con pareja", es decir, existe otra adquisición correspondiente a la misma persona. Teniendo en cuenta este hecho, para cada una de las características del vector de características se calculan las distancias de todas las combinaciones posibles de parejas de casos. Partiendo de la suposición de que las distancias serán en general más pequeñas cuando se comparan casos correspondientes al mismo sujeto que cuando se comparan casos correspondientes a sujetos distintos, es posible calcular un umbral de manera teórica tras asumir que las distribuciones de distancias en los casos de comparaciones entre el mismo sujeto y comparaciones de distintos sujetos son Gaussianas.

Majority voting

Los resultados de las comparaciones individuales pueden proporcionar decisiones individuales acerca de la comparación entre dos casos. Para alcanzar una única decisión se ha optado por una estrategia de majority voting. Esta estrategia consiste, en primer lugar, en asignar un valor binario a la decisión sobre cada característica individualmente (1 si la distancia es menor que el umbral calculado, 0 si la distancia es mayor). De esa forma, se obtiene un vector formado por 1s y 0s cuya longitud será igual al número de características utilizadas. Si la comparación se está realizando entre dos casos pertenecientes a un mismo individuo, es de esperar que el número de 1s (comparaciones positivas) sea mayor que en el caso de que la comparación se realice entre casos pertenecientes a distintos individuos. Por este motivo, se buscará un segundo umbral en el recuento de comparaciones positivas, que será el punto medio entre el máximo de las comparaciones negativas y el mínimo de las comparaciones positivas. La comparación con este segundo umbral determinará la decisión final del sistema de identificación biométrica.

El proceso de entrenamiento, por lo tanto, consiste en la determinación de los siguientes elementos:

1. Si se utiliza la FC, la FA o ambas.
2. Cuántas conexiones se consideran válidas, y cuáles son esas conexiones.
3. Cuáles son los umbrales de decisión individuales para las comparaciones correspondientes a cada una de las conexiones válidas.
4. Cuál es el umbral global sobre el porcentaje de comparaciones positivas que se utilizará en decisión.

En modo test, el método realizará los siguientes pasos para realizar una comparación entre dos casos y determinar si corresponden o no al mismo sujeto:

1. Extraer las conexiones válidas, y crear un vector de características para cada caso utilizando la FC, FA o ambas.
2. Calcular la distancia para cada uno de los valores de esos vectores de características, y compararla con el umbral correspondiente calculado en entrenamiento. Obtener por lo tanto un 1 o un 0 para el vector de comparaciones positivas.
3. Comparar el número de comparaciones positivas con el segundo umbral calculado en entrenamiento. Si es superior, entonces la verificación será positiva (se trata del mismo individuo). Si es inferior, entonces la verificación será negativa (se trata de distinto individuo).

Capítulo 6. Trabajo experimental

6.1. Introducción

Ya hemos expuesto el objetivo que persigue este estudio, además de describir las bases teóricas y el método propuesto de entre todos los que hemos investigado. Ahora vamos a describir más detalladamente el trabajo experimental realizado en el programa Matlab, opciones elegidas y los dilemas que se han encontrado en el desarrollo de estos.

En primer lugar se exponen los parámetros utilizados de los cual dependen los resultados obtenidos, además de describir detalladamente el esquema de validación. En segundo lugar vamos a exponer los resultados que nos ha dado nuestro script con los datos dispuestos, mostrándolo en una serie de gráficas, las cuales explicaremos. Y en tercer lugar vamos a realizar una discusión acerca de como una serie de criterios utilizados, los cuales no se han explicado detalladamente, pueden influir a la hora de evaluar los resultados de nuestro trabajo.

6.2. Conjunto de datos y esquema de validación

Vamos a describir brevemente el conjunto de datos disponibles. A partir de un tabla excel, la cual, mostraremos una parte de ella en la [tabla 6.1](#) al ser una tabla de 226 filas, se describen algunos casos que corresponden a casos con pareja y sujetos individuales. Es decir tenemos 226 casos, de los cuales 32 corresponden a casos repetidos, por lo que tenemos 16 sujetos a los que hemos hecho dos resonancias. Como podemos ver algunos casos repetidos se han obtenido de una manera distinta, bien con un tiempo de diferencia, o bien dentro de la misma sesión y realizando una única adquisición de resonancia de tipo T1, que será utilizada en ambos casos para calcular la conectómica.. Además es preciso destacar que los casos de una sola persona, no son todos de sujetos sanos, lo cual este criterio influirá en nuestro experimento.

caso212		
caso213		
caso214		
caso216		
caso217		
caso218		
caso219		
caso221		
caso223		
caso224		
caso226		
caso303		
caso304		
caso191	caso272	(Repetido con meses de diferencia)

caso188	caso271	(Repetido con meses de diferencia)
caso225	caso270	(Repetido con meses de diferencia)
caso220	caso259	(Repetido con meses de diferencia)
caso288	caso289	(mismo T1)
caso290	caso291	(mismo T1)
caso294	caso295	(mismos DWIs pero distinto T1)
caso296	caso297	(distintos T1s y distintos DWIs, pero adquiridos en la misma sesión)
caso298	caso299	(mismo T1)

Tabla 6.1: Extracto de la tabla general de casos empleados

También vamos a mostrar en la [figura 6.1](#) y la [figura 6.2](#) una parte de las matrices de conectómica, ya que son matrices 84x84 bastante grandes.

	VarName1	VarName2	VarName3	VarName4	VarName5	VarName6	VarName7	VarName8
	NUMBER	NUMBER	NUMBER	NUMBER	NUMBER	NUMBER	NUMBER	NUMBER
1	1453	0	0	0	0	86	1445	169
2	0	1061	3	0	0	0	0	0
3	0	0	6972	1	0	9	7	14
4	0	0	0	5004	0	4	134	0
5	0	0	0	0	459	131	0	414
6	0	0	0	0	0	13732	39	5134
7	0	0	0	0	0	0	10073	201
8	0	0	0	0	0	0	0	10613

Figura 6-1: Visualización de una matriz de conectómica basado en el número de fibras entre regiones

	VarName1	VarName2	VarName3	VarName4	VarName5	VarName6	VarName7	VarName8
	NUMBER	NUMBER	NUMBER	NUMBER	NUMBER	NUMBER	NUMBER	NUMBER
1	0.2689913962	0	0	0	0	0.5260853341	0.3534417834	0.3770478807
2	0	0.1869311929	0.3868607084	0	0	0	0	0
3	0	0	0.2832555547	0.4526613355	0	0.4362638791	0.4077519604	0.4350367337
4	0	0	0	0.2601802129	0	0.4150941521	0.3607047901	0
5	0	0	0	0	0.2459356045	0.306933572	0	0.2603156963
6	0	0	0	0	0	0.2637424513	0.4575670583	0.2770391882
7	0	0	0	0	0	0	0.2691328922	0.4668194917
8	0	0	0	0	0	0	0	0.2548489433

Figura 6-2: Visualización de una matriz de conectómica basado en la FA media entre regiones

Los datos anteriores se han almacenado en un fichero `.mat` llamado `datos_conectomica_para_id_26_junio.mat` que contiene una tabla con la lista de casos, y dos matrices que contienen los datos de manera vectorial para cada caso con las dos características de la FC y FA, siendo el número de filas correspondiente al número de casos y el número de columnas correspondiente al vector de características de cada

caso, que en este procesado tiene un tamaño de 3570. A partir de este .mat comienza el trabajo experimental.

En el Apéndice A hemos escrito detalladamente el código que hemos utilizado, el cuál tras la explicación general que realizaremos en este apartado, se puede consultar para obtener más detalles concretos.

Hay varios parámetros que se pueden modificar en el código como vemos en la [figura 6.3](#) para ver cómo varían los resultados:

-M: Es el número de veces que repetimos el proceso de generar datos de entrenamiento y de test, y luego entrenar, validar y obtener resultados.

-N: Es el número de fibras para cada conexión que exigimos que tengan todos los casos para considerarla conexión válida.

-A1 y A2: Número de casos solitarios y pareados que apartamos para el entrenamiento en cada iteración.

-opción: Sirve para decidir si usamos solamente la FA, solamente la FC o las dos cosas.

-subopción: Sirve para decidir si normalizamos los datos, estandarizamos, o no hacemos nada. Como las comparaciones se hacen ahora individualmente, no será necesaria en nuestro método, pero la destacamos por si se puede utilizar en líneas futuras.

```
load datos_conectomica_para_id_26_junio.mat
%load datos_conectomica_para_id.mat
M=10;    % Numero de veces que separamos casos de entrenamiento y de test y
        % hacemos todo el proceso (entrenamiento y test) hasta obtener
        % resultados sobre la tasa de acierto del invento. Se hace al azar
        % este procedimiento
A1=30;   % Numero de casos solitarios que usamos en entrenamiento
A2=16;   % Numero de casos pareados que usamos en entrenamiento

N=500;  % Numero de fibras que exigimos a cada conexion para que sea valida
opcion=3;
subopcion=3;
```

Figura 6-3: Visualización de los parámetros a variar antes de ejecutar el script.

En este método hemos usado conjuntamente las características de la FA y la FC(opcion=3), las cuales hemos estandarizado(subopcion=3), aunque no sea necesario porque se las comparaciones se hacen individualmente. El resto de parámetros son un ejemplo, pero se pueden utilizar otros valores perfectamente. En el parámetro M, damos un valor de 10, para poder visualizar los parámetros que muestran por pantalla detalladamente. En el número de casos solitarios cogemos un valor grande pero no demasiado, y en cuestión de los casos pareados a utilizar en entrenamiento, hemos cogido la mitad de los disponibles, ya que en el esquema de validación no podremos utilizar estos casos. Y acerca del número de fibras hemos escogido un parámetro no demasiado alto ni demasiado bajo, acorde con la [figura 6.4](#) en la cual comparamos el número de conexiones válidas con el valor del umbral que obtenemos. En ella como podemos ver que para un valor mayor de 500, el umbral empieza a aumentar a un ritmo considerable. Por lo que cogemos ese valor que está en el límite cuando el umbral cambia su pendiente.

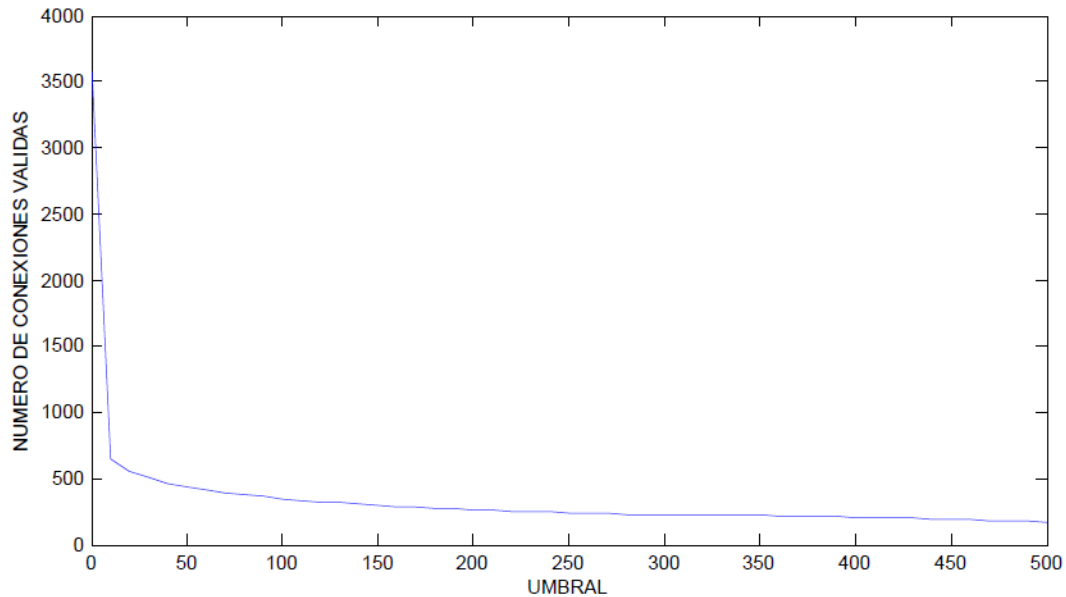


Figura 6-4: Gráfica del número de conexiones válidas en función del valor del umbral.

Tras haber descrito el conjunto de datos y los parámetros del script, vamos a describir los apartados del código del apartado A.1:

- 1) Obtenemos los índices de los casos de entrenamiento y los casos de test: Averiguamos los índices de los casos que son únicos y los casos que tienen pareja. Guardamos esa información en un vector, con un valor para cada caso. Si es un caso único, ponemos 0. Si es un caso con pareja, es decir, la misma persona corresponde a otro caso también, ponemos el índice de la pareja. Luego aleatoriamente para cada iteración escogemos una serie de casos solitarios y otros tantos en pareja. Con los vectores de índices obtenemos cuatro matrices que contienen los casos escogidos para sujetos individuales y en pareja para la FC y la FA en cada caso.
- 2) Decidimos con qué conexiones nos quedamos para analizarlas. Para ello decidimos cuál debe ser el umbral de número de fibras, el cual hemos escogido según la [gráfica 6.4](#). Con este apartado las matrices que contienen la información de la FC y la FA para casos solitarios y en pareja, se reducen en tamaño.
- 3) Dependiendo de la opción elegida, nos quedamos con la FA, FC o las dos, y normalizamos, estandarizamos o no hacemos nada. Para ello hemos creado una función, la cual está descrita en el apartado A.2 que realiza las dos funciones propiamente dichas.
- 4) Recorremos cada una de las características del vector de características, y estudiamos las distancias calculadas cuando los casos pertenecen a sujetos que son la misma persona y cuando son distintos. Hemos comparado la distancia entre dos casos con un umbral óptimo asumiendo que las dos distancias tienen funciones de densidad de probabilidad gaussianas. Tenemos por lo tanto que igualar las funciones de densidad de probabilidad, porque el umbral óptimo se encuentra en la intersección de ambas curvas. El umbral óptimo se ha hallado con la función descrita en el apartado A.4. Por lo tanto con la función de A.4 se obtendrá un vector con los umbrales (vector T), en la que cada umbral se ha

obtenido mediante el método descrito entre dos casos. Ahora calculamos la probabilidad de error, que será la suma de probabilidades acumuladas por cada gaussiana al otro lado del umbral. Con eso calculamos el porcentaje de comparaciones que superan el umbral en un caso, y en otro. Entonces calculamos el umbral óptimo final(umbral_comparaciones_positivas) como el punto medio entre el mínimo del porcentaje de comparaciones positivas en casos iguales y el máximo de las comparaciones positivas en casos distintos. Este es el valor que tendremos que utilizar en el conjunto de test para evaluar si el porcentaje de comparaciones positivas nos permite afirmar que el caso pertenece a un sujeto solitario o con pareja.

- 5) Test: Vamos a comparar todos los casos entre sí. Recorremos cada una de las características del vector de características, y estudiamos las distancias que se obtienen al hacer la comparación. Ahora vemos si las comparaciones que hemos hecho han superado o no el umbral. Con eso calculamos la tasa de acierto para ambos casos, que son la tasa de acierto cuando hemos dicho que dos casos pertenecen a la misma persona, y efectivamente, pertenecen a la misma persona y la tasa de acierto cuando hemos dicho que dos casos no pertenecen a la misma persona y realmente no pertenecen al mismo sujeto.

6.3. Resultados

Vamos a ver en la tabla 6.2 la tasa de acierto cuando hemos ejecutado nuestro código para un cada iteración, habiendo un número de iteraciones $M=7$.

Tasa de acierto cuando hemos dicho que dos casos pertenecen a la misma persona, y efectivamente, pertenecen a la misma persona.	Tasa de acierto cuando hemos dicho que dos casos no pertenecen a la misma persona, y efectivamente, no pertenecen a la misma persona.
1.0000	0.9999
0.8750	0.9999
0.7500	0.9999
1.0000	0.9999
0.8750	0.9999
0.7500	0.9999
1.0000	1.0000

Tabla 6.2 Tasa de acierto para cada iteración

Como podemos observar nuestra tasa de acierto es muy alta, pero sin embargo existe la posibilidad de equivocarnos a la hora de determinar si un caso pertenece al mismo sujeto que otro y viceversa. Vamos a ver en la [figura 6.5](#) más detalladamente los casos de nuestro estudio, en función de probabilidades, umbral, etc para 100 iteraciones.

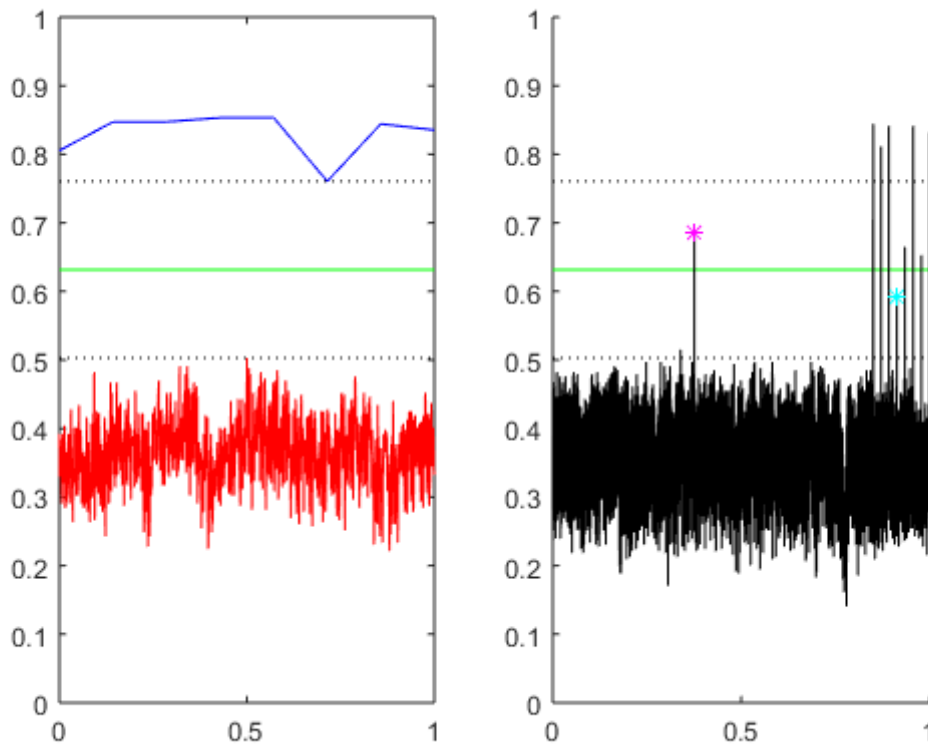


Figura 6-5: Gráfica de la representación de las distancias entre casos, así como el umbral y el porcentaje de comparaciones.

Vamos a describir detalladamente las dos gráficas. En la primera gráfica representamos los casos utilizados en el esquema de entrenamiento. En rojo tenemos representado el porcentaje de comparaciones positivas para casos distintos, es decir, para comparaciones entre casos en los que al sujeto al que pertenecen solo existe ese caso y que hemos distinguido perfectamente de otros casos. También como una línea punteada, se representa el valor máximo del porcentaje de estas comparaciones. En azul, en la misma gráfica, tenemos representado el porcentaje de comparaciones positivas para casos iguales, es decir, para comparaciones entre casos en los que al sujeto al que pertenecen existen dos casos de ese individuo y que hemos identificado como pertenecen al mismo satisfactoriamente. También como una línea punteada, se representa el valor mínimo del porcentaje de estas comparaciones. Como hemos descrito antes, el umbral óptimo lo representamos en verde, siendo el umbral el punto medio entre el mínimo del porcentaje de comparaciones positivas en casos iguales y el máximo del las comparaciones positivas en casos distintos.

En la siguiente gráfica representamos los casos utilizados en el esquema de test, habiendo hallado previamente el umbral óptimo, y representándolo en verde, además del mínimo y el máximo de los porcentajes de comparaciones descritos en el anterior apartado. En negro tenemos representado el porcentaje de comparaciones entre los casos utilizados en el esquema de validación. Como podemos observar la mayoría de comparaciones son acertadas, aunque tenemos dos casos en los que no. En rosa con un asterisco representamos la comparación entre dos casos en los que son distintos, pero que al superar el umbral, nos detecta que pertenecen a la misma persona los dos casos, cosa errónea. En azul con un asterisco representamos la comparación entre dos casos

que pertenecen a la misma persona, pero que al no superar el umbral, nos detecta que no pertenecen a la misma persona los dos casos, cosa errónea. Como vemos estos dos errores son debidos a que la tasa de acierto no es del 100% para ambos casos. En la discusión veremos algunos detalles que pueden ayudarnos a mejorar la tasa de acierto.

6.4. Discusión

En el anterior apartado, hemos obtenido una serie de resultados bastante buenos, pero que, sin embargo, no nos ofrecen una tasa de acierto a la hora de identificar y distinguir individuos del 100%. Para poder aumentar esta tasa de acierto, hay una serie de parámetros que se pueden tener en cuenta. Nosotros hemos considerado, que el vector de distancia entre casos de la misma persona y casos de distintas personas se representan mediante una función de densidad de probabilidad gaussiana. Por ejemplo, también, se podría haber aproximado la función como una t-student y podría haber dado resultados distintos aunque el procesado sea más difícil. Además a la hora de comparar casos, hemos escogido casos de una única persona que tienen alguna enfermedad, con lo que a la hora de comparar los casos, este parámetro ha sido importante. También conviene destacar que a la hora de obtención de dos resonancias de difusión del cerebro para una persona, se han hecho de diferentes maneras. Cuando hemos realizado dos resonancias a un sujeto, se ha utilizado la misma imagen anatómica o la misma imagen de difusión para la obtención de las matrices de conectómica. En estos casos se ha hecho especial hincapié en que la similitud entre ellos es mayor que para otros casos, en los que se han realizado dos resonancias con un intervalo de tiempo entre ellas, de varios meses. A estos últimos casos, hemos comprobado que sus distancias son mayores que para los casos en los que tenemos la misma imagen anatómica o de difusión. Por lo tanto, esta serie de factores son importantes a la hora de identificar biométricamente individuos a través del conectoma.

Capítulo 7. Conclusiones y trabajo futuro

7.1. Conclusiones

En este Trabajo de Fin de Grado se ha realizado un análisis de la individualidad de la conectividad cerebral. El objetivo fundamental de este trabajo era identificar individuos mediante resonancia de difusión. Con la resonancia se obtenían una serie de matrices de conectómica, las cuales hemos utilizado para poder distinguir personas biométricamente. Aunque en la práctica no tiene mucho sentido identificar a una persona metiéndola en una resonancia, pero desde un punto de vista de identificación, hemos estudiado que es lo que hace único al conectoma de una persona.

Primero para entender el trabajo experimental realizado, es necesaria una etapa de documentación en la que hemos descrito la resonancia de difusión y la identificación biométrica. Como hemos explicado a lo largo de los capítulos anteriores, en la resonancia de difusión hemos detallado acerca del funcionamiento de una resonancia magnética, su funcionamiento y cuales son los pasos necesarios para obtener matrices de conectómica. Acerca de la identificación biométrica, hemos dado una definición extensa, así como una distinción entre verificación, el cuál es el método usado. También hemos destacado los pasos generales mediante reconocimiento de patrones usados, así como resaltar otros estudios parecidos al nuestro que se han llevado a cabo.

Hemos seguido un procedimiento propio para la obtención de matrices de conectividad y para la identificación biométrica. Para el procesado de las matrices, hemos partido de una serie de datos de resonancia de difusión obtenidos mediante la resonancia magnética de la Universidad de Valladolid. Gracias a una serie de programas gratuitos, instalados en servidores del departamento del Laboratorio de Procesado de Imagen, hemos podido obtener matrices de conectómica. Cabe destacar que el tiempo de procesamiento de cada resonancia ha sido muy lento, por lo que se han realizado muchos casos en paralelo para obtener las matrices en el menor tiempo posible. Además, el coste del tiempo empleado para ello ha sido importante. Y, acerca del método de identificación biométrica utilizado, al disponer de pocos datos de resonancia de difusión, recalamos que no es infalible, pero da unos resultado muy buenos.

Hemos creado un algoritmo, el cual realiza todo el procedimiento de identificación, con un coste de procesamiento destacable, pero no tan elevado como el procesado para la obtención de matrices de conectómica. Resaltamos que los resultados obtenidos, son realmente prometedores y que aunque ha habido algunas limitaciones, el objetivo de identificar individuos a través de las conexiones cerebrales es posible.

Además, para realizar los distintos análisis del Capítulo 4 y el Capítulo 6, se han creado varios scripts que llevan a cabo todos los pasos necesarios de manera automática sin apenas ser necesaria la intervención del usuario. En el Capítulo 4 se ha descrito paso a paso cada comando utilizado, pero nosotros hemos juntado cada comando del procesado en un script, con el fin de realizar el procesado de cada paso en paralelo con otros casos y poder ahorrar tiempo de procesamiento. El script creado en el Capítulo 6, pretende

que solo conociendo una serie de parámetros del script, podamos realizar la verificación de individuos solamente ejecutando el programa.

También cabe destacar que nuestro estudio presenta una serie de limitaciones. En primer lugar, destacamos que disponemos de muy pocos individuos a los que se les ha hecho dos resonancias de difusión, concretamente 16 sujetos. Además es preciso destacar el número dispar que tenemos de sujetos con dos resonancias de difusión comparado con el número de individuos a los que se les han hecho una resonancia de difusión, que son 194 personas. Esto limita bastante nuestra investigación, ya que normalmente, uno hace una división de sus datos entre conjunto de entrenamiento y conjunto de validación, y luego simplemente calcula resultados. Sin embargo, en nuestro caso si haces una división al azar entre conjunto de entrenamiento y test, es posible que tus resultados estén sesgados. Por eso, lo que hemos hecho es plantear un esquema en el que haces el proceso de entrenamiento y test muchas veces, cada vez eligiendo conjuntos de entrenamiento y test distintos. Vas calculando resultados, y luego haces el promedio.

Una limitación del estudio es el hecho de que no se han considerado factores adicionales, como puede ser la edad, salud, tiempo entre la obtención de las resonancias de cada sujetos e incluso algunos parámetros que ya hemos discutido acerca del proceso del trabajo experimental realizado con una serie de scripts detallados del Apéndice A que influyen en la conectividad.

Otra limitación, que afecta al análisis de conectividad cerebral, es que nosotros hemos realizado el procesado de obtención de matrices de conectómica mediante una serie de programas, los cuales algunos son experimentales y aún no están perfeccionados. Si hubiésemos escogido otros programas o incluso hubiesemos utilizado otras herramientas de los programas que hemos utilizado, nuestro análisis variaría.

7.2. Líneas de trabajo futuro

Se pueden plantear varias líneas futuras del estudio que hemos realizado, teniendo en cuenta algunas limitaciones y aspectos que nos hemos encontrado.

El primer aspecto sería conseguir más datos de resonancia. En este estudio hemos realizado una división al azar entre conjunto de test y entrenamiento, pero lo normal sería dividir los datos entre esos dos conjuntos de manera fija. Para ello sería necesario adquirir más datos de resonancia de difusión, es decir, conseguir más sujetos que quieran hacerse dos resonancias magnéticas o bien conseguir una base de datos pública que nos permita tener datos de difusión. Si consiguiéramos más datos, nos permitiría ampliar nuestro estudio, y conseguir unos resultados probablemente diferentes, que nos ayudarían a determinar nuestro objetivo final de la identificación biométrica.

El segundo aspecto a tener en cuenta, serían factores como edad, salud o el tiempo entre adquisición de la primera y segunda resonancia de un sujeto. Hay casos que pertenecen a sujetos sanos que solo se han hecho una resonancia, pero también hay otros casos que pertenecen a personas con enfermedades como cefaleas o esquizofrenia, que no hemos tenido en cuenta a la hora de comparar sus matrices de conectómica con otros casos. En

una línea futura se podría separar los casos de sujetos sanos y no sanos con el fin de avanzar en el estudio de los conectomas de estos individuos. También es importante destacar que es más difícil distinguir que dos casos pertenecen a una misma persona si esa persona se ha hecho dos resonancias magnéticas con un tiempo de diferencia de meses, que dos casos de una misma persona, en las que sus resonancias se han realizado de manera seguida e incluso compartiendo la misma imagen anatómica o de difusión para los dos casos. Por eso este último factor es importante a destacar ya que determina que el conectoma humano depende del tiempo entre adquisiciones de resonancia.

El tercer factor, es el procesado para la obtención de matrices de conectividad. Nosotros hemos obtenido las matrices mediante una serie de herramientas de varios programas. Por lo tanto es posible argumentar que si hubiésemos obtenido las matrices de conectividad mediante otras herramientas de los mismos programas o hubiésemos utilizado otros programas, nuestras matrices serían diferentes a las utilizadas ahora.

Y finalmente, el último aspecto que debemos comentar, es el método de identificación usado. Nosotros al disponer de pocos datos de difusión, además de considerar varios aspectos en el programa como que las distancias entre casos están representadas como funciones de probabilidad gaussiana, hemos obtenido una tasa de acierto alta a la hora de identificar individuos. Pero si cambiasemos estos dos aspectos, e incluso otros que consideramos importantes, como la obtención del umbral, nuestro estudio podría haber dado resultados diferentes. Por lo tanto, partiendo de los mismos datos, se podría comparar diferentes métodos de identificación, y ver cual arroja mejores resultados.

Referencias

- [1] D. LeBihan, J.-F. Mangin, C. Poupon, C. A. Clark, S. Pappata, N. Molko, and H. Chabriet. Diffusion Tensor Imaging: Concepts and Applications. *Journal of Magnetic Resonance Imaging*, 13:534–546, 2001.
- [2] Patric Hagmann, Leila Cammoun, Xavier Gigandet, Stephan Gerhard, P. Ellen Grant, Van Wedeen, Reto Meulia, Jean-Philippe Thiran, Christopher J. Honey, Olaf Sporns. MR connectomics: Principles and challenges. *Journal of Neuroscience Methods*, 1:34-45, 2010.
- [3] Susumu Mori, Jiangyang Zhang. Principles of Diffusion Tensor Imaging and Its Applications to Basic Neuroscience Research. *Neuron*, 51:527-539, 2006.
- [4] C-F. Westin, S. E. Maier, H. Mamata, A. Nabavi, F. A. Jolesz, and R. Kikinis. Processing and visualization for diffusion tensor MRI. *Medical Image Analysis*, 6:93–108, 2002.
- [5] Derek K. Jones. Studying connections in the living human brain. *Cortex*, 44:936-952, 2008.
- [6] Marcus Kaiser. A tutorial in connectome analysis: Topological and spatial features of brain networks. *Neuroimage*, 57:892–907, 2011.
- [7] MRtrix. MRtrix 3.0 documentation. <http://mrtrix.readthedocs.io/en/latest/>. Último acceso: 2017-06-20.
- [8] Rodrigo De Luis García, (2016). Pipeline_procesado_dti_conectómica. UVA, Valladolid, España.
- [9] DICOM(nema). DICOM: About DICOM. <http://medical.nema.org/Dicom/about-DICOM.html>. Último acceso: 2017-06-20.
- [10] Mathworks. Matlab: El lenguaje del cálculo técnico. <https://es.mathworks.com/products/matlab.html>. Último acceso: 2017-06-20.
- [11] Slicer. 3D Slicer. <http://www.slicer.org/>. Último acceso: 2017-06-20.
- [12] FSL. FSL - FslWiki. <http://fsl.fmrib.ox.ac.uk/fsl/fslwiki/>. Último acceso: 2017-06-20.
- [13] Laboratory for Computational Neuroimaging, Athinoula A. Martinos Center. FreeSurfer. <http://surfer.nmr.mgh.harvard.edu/>. Último acceso: 2017-06-20.
- [14] University of Oregon. MRIConvert and mcverter. <http://lcn.uoregon.edu/downloads/mriconvert/mriconvert-and-mcverter>. Último acceso: 2017-06-20.
- [15] MRtrix. <http://mrtrix.readthedocs.io/en/latest/reference/commands/mrview.html>. Último acceso: 2017-06-20.
- [16] Freesurfer. https://surfer.nmr.mgh.harvard.edu/pub/docs/html/mri_convert.help.xml. Último acceso: 2017-06-20.
- [17] FSL. <https://fsl.fmrib.ox.ac.uk/fsl/fslwiki/BET/UserGuide>. Último acceso: 2017-06-20.

- [18] Freesurfer. recon-all - Free Surfer Wiki. <https://surfer.nmr.mgh.harvard.edu/fswiki/recon-all> . Último acceso: 2017-06-20.
- [19] B. Fischl, D.H. Salat, E. Busa, M. Albert, M. Dieterich, C. Haselgrove, A. van der Kouwe, R. Killiany, D. Kennedy, S. Klaveness, A. Montillo, N. Makris, B. Rosen, and A.M. Dale. Whole brain segmentation: Automated labeling of neuroanatomical structures in the human brain. *Neuron*, 33:341–355, 2002.
- [20] B. Fischl, A. van der Kouwe, C. Destrieux, E. Halgren, F. Segonne, D. Salat, E. Busa, L. Seidman, J. Goldstein, D. Kennedy, V. Caviness, N. Makris, B. Rosen, and A.M. Dale. Automatically parcellating the human cerebral cortex. *Cerebral Cortex*, 14:11–22, 2004.
- [21] FreeSurfer. ReconAllDevTable - Free Surfer Wiki. <https://surfer.nmr.mgh.harvard.edu/fswiki/ReconAllDevTable> . Último acceso: 2017-06-20.
- [22] Slicer. Documentation/4.5/Modules/DWIconverter - SlicerWiki. <https://www.slicer.org/wiki/Documentation/4.5/Modules/DWIconverter> . Último acceso: 2017-06-20.
- [23] MRtrix. Anatomically-Constrained Tractography (ACT) . http://mrtrix.readthedocs.io/en/latest/quantitative_structural_connectivity/act.html . Último acceso: 2017-06-20.
- [24] MRtrix. 5ttgen. <http://mrtrix.readthedocs.io/en/latest/reference/scripts/5ttgen.html> . Último acceso: 2017-06-20.
- [25] Smith, R. E., Tournier, J.-D., Calamante, F., & Connelly, A. Anatomically-constrained tractography: Improved diffusion MRI streamlines tractography through effective use of anatomical information. *NeuroImage*, 62:1924–1938, 2012.
- [26] MRtrix. 5tt2vis. <http://mrtrix.readthedocs.io/en/latest/reference/commands/5tt2vis.html> . Último acceso: 2017-06-20.
- [27] MRtrix. dwi2fod. <http://mrtrix.readthedocs.io/en/latest/reference/commands/dwi2fod.html> . Último acceso: 2017-06-20.
- [28] MRtrix. Response function estimation . http://mrtrix.readthedocs.io/en/latest/constrained_spherical_deconvolution/response_function_estimation.html . Último acceso: 2017-06-20.
- [29] Chantal M.W. Tax, Ben Jeurissen, Sjoerd B. Vos, Max A. Viergever, Alexander Leemans. Recursive calibration of the fiber response function for spherical deconvolution of diffusion MRI data. *NeuroImage*, 86:67–80, 2014.
- [30] MRtrix. shview. <http://mrtrix.readthedocs.io/en/latest/reference/commands/shview.html> . Último acceso: 2017-06-20.
- [31] FSL. FDT/UserGuide - FslWiki. <https://fsl.fmrib.ox.ac.uk/fsl/fslwiki/FDT/UserGuide#DTIFIT> . Último acceso: 2017-06-20.
- [32] MRtrix. Structural connectome construction . http://mrtrix.readthedocs.io/en/latest/quantitative_structural_connectivity/structural_connectome.html . Último acceso: 2017-06-20.

- [33] MRtrix. labelconvert: Explanation & demonstration. http://mrtrix.readthedocs.io/en/latest/quantitative_structural_connectivity/labelconvert_tutorial.html . Último acceso: 2017-06-20.
- [34] MRtrix. labelconvert. <http://mrtrix.readthedocs.io/en/latest/reference/commands/labelconvert.html> . Último acceso: 2017-06-20.
- [35] FSL. FLIRT - FslWiki. <https://fsl.fmrib.ox.ac.uk/fsl/fslwiki/FLIRT> . Último acceso: 2017-06-20.
- [36] FSL. FLIRT/UserGuide - FslWiki https://fsl.fmrib.ox.ac.uk/fsl/fslwiki/FLIRT/UserGuide#convert_xfm . Último acceso: 2017-06-20.
- [37] MRtrix. tckgen. <http://mrtrix.readthedocs.io/en/latest/reference/commands/tckgen.html> . Último acceso: 2017-06-20.
- [38] MRtrix. tck2connectome. <http://mrtrix.readthedocs.io/en/latest/reference/commands/tck2connectome.html> . Último acceso: 2017-06-20.
- [39] MRtrix. tcksample. <http://mrtrix.readthedocs.io/en/latest/reference/commands/tcksample.html> . Último acceso: 2017-06-20.
- [40] J.-Donald Tournier, Chun-Hung Yeh, Fernando Calamante, Kuan-Hung Cho, Alan Connelly, Ching-Po Lin. Resolving crossing fibres using constrained spherical deconvolution: Validation using diffusion-weighted imaging phantom data. *Neuroimage* , 42:617-625, 2008.
- [41] Rodrigo de Luis García , Carlos Alberola Lopez, Otman Aghzout, Juan Ruiz Alzola. Biometric identification systems. *Signal Processing*. 83:2539-2557, 2003.
- [42] Anil K. Jain, Robert P.W. Duin, Jianchang Mao. Statistical pattern recognition: a review. *IEEE*. 22:4-37, 2000.
- [43] Emily S Finn, Xilin Shen, Dustin Scheinost, Monica D Rosenberg, Jessica Huang, Marvin M Chun, Xenophon Papademetris, R Todd Constable. Functional connectome fingerprinting: identifying individuals using patterns of brain connectivity. *Nature Neuroscience*. 18:1664-1671, 2015.
- [44] Julien Dubois, Ralph Adolphs. Building a Science of Individual Differences from fMRI. *Trends in Cognitive Sciences*. 20:425-443, 2016.
- [45] Fang-Cheng Yeh , Jean M. Vettel, Aarti Singh, Barnabas Poczos, Scott T. Grafton, Kirk I. Erickson, Wen-Yih I. Tseng, Timothy D. Verstynen. Quantifying Differences and Similarities in Whole-Brain White Matter Architecture Using Local Connectome Fingerprints. *Plos*. 2015.
- [46] Greg D Parker , George J.A. Evans , and Derek K Jones. Structural Fingerprinting of the Human Brain: How unique is tract shape to the individual?. *ISMRM*. 2016.

Apéndice A. Scripts

Se han creado varios scripts para realizar los análisis descritos en el Capítulo 6 . Estos scripts se han escrito en el lenguaje matemático de Matlab. Para comenzar el procesado, se ha ejecutado una función principal llamada *funcion_global_identificacion_conectomicaV3.m* que realizará el análisis completo de identificación de sujetos a partir de matrices de conectómica. esta función principal esta compuesta por varias funciones que también se describirán a continuación.

A.1. Script 1: *funcion_global_identificacion_conectomicaV3*

```
function funcion_global_identificacion_conectomicaV3

% V3: 26 de Junio de 2017
% A partir del conjunto total de casos, hacemos un bucle general para
% probar diferentes combinaciones de conjuntos de entrenamiento y de
% test.
% En lugar de hacer una búsqueda exhaustiva de todas las
% combinaciones,
% simplemente hacemos un numero (M) de combinaciones

% V2: Reformamos cuestiones de indices porque estaba demasiado
% complicado

% Funcion gorda que hace todo lo necesario para la identificacion con
% conectomica.

% Fases:
% 1- Decidimos cuales son los datos de entrenamiento y cuales son los
% datos
% de test
% 2- Decidimos cual debe ser el umbral de numero de fibras
% 3-

% Opciones:
% 1.- Utilizamos solamente la FC. Podriamos utilizar los datos
% originales,
% los normalizados, o los estandarizados
% 2.- Utilizamos solamente la FA. Podriamos utilizar los datos
% originales,
% los normalizados, o los estandarizados
% 3.- Utilizamos la FC junto con la FA. En este caso es necesario
% normalizar o estandarizar
%
% Subopciones:
% 1.- Características sin normalizar o estandarizar
% 2.- Características normalizadas
% 3.- Características estandarizadas

load datos_conectomica_para_id_26_junio.mat
%load datos_conectomica_para_id.mat
M=10; % Numero de veces que separamos casos de entrenamiento y de
test y
% hacemos todo el proceso (entrenamiento y test) hasta obtener
```

```

        % resultados sobre la tasa de acierto del invento. Se hace al
azar
        % este procedimiento
A1=30;    % Numero de casos solitarios que usamos en entrenamiento
A2=16;    % Numero de casos pareados que usamos en entrenamiento

N=500; % Numero de fibras que exigimos a cada conexion para que sea
valida
opcion=3;
subopcion=3;

% Averiguamos los indices de los casos que son unicos y los casos que
% tienen pareja. Guardamos esa info en un vector, con un valor para
cada
% caso. Si es un caso unico, ponemos 0. Si es un caso con pareja,
ponemos
% el indice de la pareja
for i=1:length(tabla_lista_casos)
    if(length(char(tabla_lista_casos{i,2}))==0)    % Es un caso unico
        indice_caso_pareja(i)=0;
    else    % Es un caso con
pareja
        nombre_caso_complementario=char(tabla_lista_casos{i,2})
        for j=1:length(tabla_lista_casos)

if(strcmp(tabla_lista_casos{j,1},nombre_caso_complementario)==1)
            indice_caso_pareja(i)=j;
        end
    end
end
end
indice_caso_pareja

cuantos_casos_total=length(indice_caso_pareja)
cuantos_casos_solitarios=length(find(indice_caso_pareja==0))
cuantos_casos_pareados=length(find(indice_caso_pareja>0))

A1    % Numero de casos solitarios que usamos en
entrenamiento
A2    % Numero de parejas que usamos en entrenamiento
B1=cuantos_casos_solitarios-A1 % Numero de casos solitarios que
usamos en test
B2=cuantos_casos_pareados-A2    % Numero de parejas que usamos en test

tamanyo_conjunto_entrenamiento=A1+A2
tamanyo_conjunto_test=B1+B2
%pause;

%ind=1:length(tabla_lista_casos)    % Todos los casos

% BUCLE PARA HACER LEAVE ONE OUT
disp('BUCLE GENERAL CON M REPETICIONES');
for i=1:M

    disp(' ');
    disp('-----');

```



```

disp('-----');
%disp('Extraemos los casos de entrenamiento y los casos de test');
i

% 1.- Obtenemos los indices de los casos de entrenamiento y los
casos de
% test
disp('OBTENCION DE LOS DATOS DE ENTRENAMIENTO Y DE LOS DATOS DE
TEST');

indices_solitarios=find(indice_caso_pareja==0);
indices_solitarios_random=randperm(length(indices_solitarios));
indices_casos_entrenamiento1=indices_solitarios_random(1:A1);
indices_casos_test1=indices_solitarios_random(A1+1:end);

indices_pareados=find(indice_caso_pareja>0)
% Como sabemos que los casos pareados estan consecutivos (la
pareja
% esta a continuacion del primer caso), elegimos al azar teniendo
en
% cuenta esto
indices_pareados_random=randperm(length(indices_pareados)/2)
contador=1;
for j=1:A2/2

indices_casos_entrenamiento2(contador)=2*indices_pareados_random(j)-1;

indices_casos_entrenamiento2(contador+1)=2*indices_pareados_random(j);
contador=contador+2;
end
indices_casos_entrenamiento2
contador=1;
for
j=A2/2+1:(A2+B2)/2%length(indices_pareados_random)/2+1:length(indices_
pareados_random)
j
indices_casos_test2(contador)=2*indices_pareados_random(j)-1;
indices_casos_test2(contador+1)=2*indices_pareados_random(j);
contador=contador+2;
end
indices_casos_test2
%pause;

indices_casos_entrenamiento2=indices_casos_entrenamiento2+indices_pare
ados(1)-1;
indices_casos_test2=indices_casos_test2+indices_pareados(1)-1;

indices_casos_entrenamiento=[indices_casos_entrenamiento1
indices_casos_entrenamiento2]
pareja_casos_entrenamiento=indice_caso_pareja(indices_casos_entrenamie
nto)
indices_casos_test=[indices_casos_test1 indices_casos_test2]
pareja_casos_test=indice_caso_pareja(indices_casos_test)

```

```

indices_nuevos_casos_entrenamiento=1:length(indices_casos_entrenamiento);
for j=1:length(indices_nuevos_casos_entrenamiento)

ind=find(pareja_casos_entrenamiento==indices_casos_entrenamiento(j));
if(length(ind)==1)
    indices_nuevos_pareja_entrenamiento(j)=ind;
else
    indices_nuevos_pareja_entrenamiento(j)=0;
end
end
indices_nuevos_casos_test=1:length(indices_casos_test);
for j=1:length(indices_nuevos_casos_test)
    ind=find(pareja_casos_test==indices_casos_test(j));
    if(length(ind)==1)
        indices_nuevos_pareja_test(j)=ind;
    else
        indices_nuevos_pareja_test(j)=0;
    end
end

indices_nuevos_casos_entrenamiento;
indices_nuevos_pareja_entrenamiento;
indices_nuevos_casos_test;
indices_nuevos_pareja_test;

%pause;

matriz_casos_FA_entrenamiento=matriz_casos_FA(indices_casos_entrenamiento,:);

matriz_casos_FC_entrenamiento=matriz_casos_FC(indices_casos_entrenamiento,:);
matriz_casos_FA_test=matriz_casos_FA(indices_casos_test,:);
matriz_casos_FC_test=matriz_casos_FC(indices_casos_test,:);

tamano_datos_entrenamiento=size(matriz_casos_FC_entrenamiento);
tamano_datos_test=size(matriz_casos_FC_test);

indice_pareja_entrenamiento=pareja_casos_entrenamiento;
indice_pareja_test=pareja_casos_test;

%pause;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% ENTRENAMIENTO %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% 2.- Decidimos con que conexiones nos quedamos para analizarlas
% Decidimos cual debe ser el umbral de numero de fibras
%N=500;
disp('EXTRACCION DE LAS CONEXIONES VALIDAS');
conexion_valida=ones(1,size(matriz_casos_FC_entrenamiento,2));

```

```

    for j=1:size(matriz_casos_FC_entrenamiento,2) % Recorremos todas
    las conexiones
        FC_min_conexiones(j)=min(squeeze(matriz_casos_FC_entrenamiento(:,j)));
        if(FC_min_conexiones(j)<N)
            conexion_valida(j)=0;
        end
    end

    num_conexiones_validas=sum(conexion_valida);
    %pause;

    % Nos quedamos solamente con las conexiones validas
    indices_conexion_valida=find(conexion_valida==1);

matriz_casos_FC_test_validas=matriz_casos_FC_test(:,indices_conexion_v
alida);

matriz_casos_FA_test_validas=matriz_casos_FA_test(:,indices_conexion_v
alida);

matriz_casos_FC_entrenamiento_validas=matriz_casos_FC_entrenamiento(:,
indices_conexion_valida);

matriz_casos_FA_entrenamiento_validas=matriz_casos_FA_entrenamiento(:,
indices_conexion_valida);

    size(matriz_casos_FC_entrenamiento_validas);
    size(matriz_casos_FC_test_validas);

    % 3.- Dependiendo de la opcion elegida, nos quedamos con FA, FC o
    los
    % dos, y normalizamos o estandarizamos (o nada)
    disp('DECIDIMOS QUE CARACTERISTICAS QUEREMOS (FA, FC O AMBAS) Y
    NORMALIZAMOS O ESTANDARIZAMOS (O NADA)');
    [matriz_datos_entrenamiento,
matriz_datos_test]=normaliza_o_estandariza(matriz_casos_FA_entrenamien
to_validas,matriz_casos_FA_test_validas,matriz_casos_FC_entrenamiento_
validas,matriz_casos_FC_test_validas,opcion,subopcion);

    %size(matriz_datos_entrenamiento);
    %size(matriz_datos_test);

    %pause;

    % 4.- Recorremos cada una de las características del vector de
    % características, y estudiamos las distancias cuando los sujetos
    son
    % iguales y cuando son distintos
    disp('ESTUDIAMOS LAS DISTANCIAS CUANDO ES EL MISMO SUJETO O SUJETO
    DISTINTO PARA BUSCAR EL UMBRAL OPTIMO');

```

```

    for m=1:size(matriz_datos_entrenamiento,2) % Bucle para cada una
de las características del feature vector

vector_datos_entrenamiento=squeeze(matriz_datos_entrenamiento(:,m));
vector_datos_entrenamiento';
indices_nuevos_pareja_entrenamiento;

contador=0;
for p=1:length(vector_datos_entrenamiento)
    for r=1:length(vector_datos_entrenamiento)
        if (p>r)
            contador=contador+1;
            vector1=vector_datos_entrenamiento(p);
            vector2=vector_datos_entrenamiento(r);

            distancia(contador)=abs(vector1-vector2);
            if(indices_nuevos_pareja_entrenamiento(p)==r)
                [p,r];
                iguales(contador)=1;
                %squeeze(distancia(contador));
            else
                iguales(contador)=0;
            end
        end
    end
end

%if(m==1)
%contador
%pause;
%end

ind1=find(iguales==0);
medial=mean(distancia(ind1));
std1=std(distancia(ind1));

ind2=find(iguales==1);
media2=mean(distancia(ind2));
std2=std(distancia(ind2));

% Calculamos el umbral optimo asumiendo que las dos distancias
tienen
% pdfs gaussianas
dibuja=0;
T(m)=calcula_umbral_optimo(medial,std1,media2,std2,dibuja);

% Ahora calculamos la probabilidad de error, que sera la suma
de
% probabilidades acumuladas por cada gaussiana al otro lado
del umbral
P1=1/2;
P2=1/2;
if(medial<media2)
    P=(1/2)*(1-
normcdf(T(m),medial1,std1)+normcdf(T(m),media2,std2));
else
    P=(1/2)*(1-
normcdf(T(m),media2,std2)+normcdf(T(m),medial1,std1));

```

```

end

vector_probabilidades(m)=P;

% Ahora vemos si las comparaciones que hicimos superaron o no
% superaron el umbral (para comparaciones entre iguales y para
% comparaciones entre distintos). Con eso calculamos el
porcentaje
% de comparaciones que superan el umbral en un caso, y en otro
ind_iguales=find(iguales==1);
ind_distintos=find(iguales==0);

for t=1:length(distancia)
    if(distancia(t)<T(m))
        vector_comparaciones(t)=1;
    else
        vector_comparaciones(t)=0;
    end
end

matriz_comparaciones(:,m)=vector_comparaciones;
%matriz_comparaciones(:,m)=distancia<T(m);

%pause;

end
%vector_probabilidades'

%size(distancia)
%pause;

porcentaje_comparaciones_positivas=sum(matriz_comparaciones,2)/size(matriz_comparaciones,2);
size(matriz_comparaciones)
clear matriz_comparaciones

porcentaje_comparaciones_positivas(ind_iguales)
porcentaje_comparaciones_positivas(ind_distintos);
size(porcentaje_comparaciones_positivas(ind_distintos))

length(ind_distintos)

h=figure(3);
clf;
subplot(1,2,1);
set(h,'Color',[1 1 1]);

plot(linspace(0,1,length(porcentaje_comparaciones_positivas(ind_iguales))),porcentaje_comparaciones_positivas(ind_iguales),'b');
hold on;

plot(linspace(0,1,length(porcentaje_comparaciones_positivas(ind_distintos))),porcentaje_comparaciones_positivas(ind_distintos),'r');
axis([0 1 0 1]);

medial=mean(porcentaje_comparaciones_positivas(ind_iguales))

```

```

std1=std(porcentaje_comparaciones_positivas(ind_iguales))

media2=mean(porcentaje_comparaciones_positivas(ind_distintos))
std2=std(porcentaje_comparaciones_positivas(ind_distintos))

% Calculamos el umbral optimo como el punto medio entre el minimo
del
% porcentaje de comparaciones positivas en casos iguales y el
maximo
% del las comparaciones positivas en casos distintos

minimo_porcentaje_iguales=min(porcentaje_comparaciones_positivas(ind_i
guales));

maximo_porcentaje_distintas=max(porcentaje_comparaciones_positivas(ind
_distintos));

umbral_comparaciones_positivas=(minimo_porcentaje_iguales+maximo_porce
ntaje_distintas)/2
%pause;

figure(3);
hold on;
plot([0 1],ones(1,2)*minimo_porcentaje_iguales,'k:');
hold on;
plot([0 1],ones(1,2)*maximo_porcentaje_distintas,'k:');
hold on;
plot([0 1],ones(1,2)*umbral_comparaciones_positivas,'g');

% Al finalizar el bucle anterior, hemos obtenido:
% - El vector T, que es un vector con los umbrales a utilizar con
el
% conjunto de test para cada una de las características del vector
de
% características
% - El valor "umbral_comparaciones_positivas", que es el valor que
% tendremos que utilizar en el conjunto de test para evaluar si el
% porcentaje de comparaciones positivas nos permite afirmar que el
caso
% pertenece a un sujeto solitario o con pareja
%pause;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% TEST %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

disp('TEST');
% Para hacer el test, vamos a comparar todos con todos.
% 5.- Recorremos cada una de las características del vector de
% características, y estudiamos las distancias que se obtienen al
hacer
% la comparacion

for m=1:size(matriz_datos_test,2) % Bucle para cada una de las
características del feature vector
vector_datos_test=squeeze(matriz_datos_test(:,m));
vector_datos_test';
indices_nuevos_pareja_test;

```

```

    %pause;

    contador=0;
    for p=1:length(vector_datos_test)
        for r=1:length(vector_datos_test)
            if (p>r)
                contador=contador+1;
                vector1=vector_datos_test(p);
                vector2=vector_datos_test(r);

                distancia2(contador)=abs(vector1-vector2);
                if (indices_nuevos_pareja_test(p)==r)
                    [p,r];
                    iguales2(contador)=1;
                    %squeeze(distancia2(contador));
                else
                    iguales2(contador)=0;
                end
            end
        end
    end
    end
    if (m==1)
        %contador
        %pause
    end

    % Ahora vemos si las comparaciones que hicimos superaron o no
    % superaron el umbral (para comparaciones entre iguales y para
    % comparaciones entre distintos). Con eso calculamos el
porcentaje
    % de comparaciones que superan el umbral en un caso, y en otro
    ind_iguales=find(iguales2==1);
    ind_distintos=find(iguales2==0);

    matriz_comparaciones_test(:,m)=distancia2<T(m);

    %pause;

end

porcentaje_comparaciones_positivas=sum(matriz_comparaciones_test,2)/size(matriz_comparaciones_test,2);
clear matriz_comparaciones_test

indices_iguales_estimado=find(porcentaje_comparaciones_positivas>umbral_comparaciones_positivas);

indices_distintos_estimado=find(porcentaje_comparaciones_positivas<=umbral_comparaciones_positivas);
[indices_iguales_estimado
10000*porcentaje_comparaciones_positivas(indices_iguales_estimado)];

indices_iguales_verdadero=ind_iguales;

```

```

    indices_distintos_verdadero=ind_distintos;
    [indices_iguales_verdadero'
10000*porcentaje_comparaciones_positivas(indices_iguales_verdadero)]

    % Obtenemos resultados numericos
    [tasa_acierto_1(i)
tasa_acierto_2(i)]=compara_indices_iguales_encontrados(indices_iguales_
_estimado,indices_iguales_verdadero,indices_distintos_estimado,indices_
_distintos_verdadero)

    %pause;

    figure(3);
    subplot(1,2,2);
    set(h,'Color',[1 1 1]);

    %plot(linspace(0,1,length(porcentaje_comparaciones_positivas(indices_i
guales_estimado))),porcentaje_comparaciones_positivas(indices_iguales_
_estimado),'b');
    hold on;

    %plot(linspace(0,1,length(porcentaje_comparaciones_positivas(indices_d
istintos_estimado))),porcentaje_comparaciones_positivas(indices_distin
tos_estimado),'r');
    hold on;
    plot([0 1],ones(1,2)*minimo_porcentaje_iguales,'k:');
    hold on;
    plot([0 1],ones(1,2)*maximo_porcentaje_distintas,'k:');
    hold on;
    plot([0 1],ones(1,2)*umbral_comparaciones_positivas,'g');
    axis([0 1 0 1]);

    hold on;

    plot(linspace(0,1,length(porcentaje_comparaciones_positivas)),porcenta
je_comparaciones_positivas,'k');
    x=linspace(0,1,length(porcentaje_comparaciones_positivas));
    for s=1:length(porcentaje_comparaciones_positivas)
        if(
(porcentaje_comparaciones_positivas(s)>umbral_comparaciones_positivas)
&& iguales2(s)==0 )
            plot(x(s),porcentaje_comparaciones_positivas(s),'m*');
        elseif(
(porcentaje_comparaciones_positivas(s)<umbral_comparaciones_positivas)
&& iguales2(s)==1 )
            plot(x(s),porcentaje_comparaciones_positivas(s),'c*');
        end
    end

    %porcentajes_comparaciones_encontradas=porcentaje_comparaciones_positi
vas(indices_iguales_estimado)

```



```

%porcentajes_comparaciones_verdaderas=porcentaje_comparaciones_positiv
as(indices_iguales_verdadero)
    %indices_iguales_verdadero'

    disp('Final iteracion');
    %pause;
end

[tasa_acierto_1' tasa_acierto_2']

mean(tasa_acierto_1)
mean(tasa_acierto_2)

```

```

if(0)

    % Para cada caso de test, comparamos la distancia con cada uno de
los
    % datos de entrenamiento (y la pareja de test, si la hubiera).
    if(indice_pareja_test==0)
        % Es un caso unico
        disp('Caso unico');
        matriz_datos_comparacion=matriz_datos_entrenamiento;
    else
        % Es un caso con pareja
        disp('Caso con pareja');

matriz_datos_comparacion=[matriz_datos_entrenamiento;matriz_datos_test
(2,:)];
    end

    for k=1:size(matriz_datos_comparacion,1) % Bucle de comparacion
con cada uno de los casos de comparacion

        comparaciones_positivas=0;

```

```

        %size(matriz_datos_comparacion,2)
        %pause;
        for m=1:size(matriz_datos_comparacion,2) % Bucle para cada
una de las características del feature vector

            vector1=squeeze(matriz_datos_test(1,m));
            vector2=squeeze(matriz_datos_comparacion(k,m));

            distancia(m)=abs(vector1-vector2);
            if(distancia(m)<T(m))
                comparaciones_positivas=comparaciones_positivas+1;
            end
        end
        total_comparaciones_positivas(k)=comparaciones_positivas;

porcentaje_positivas(k)=total_comparaciones_positivas(k)/size(matriz_d
atos_comparacion,2);
    end

disp('-----');

porcentaje_positivas'
nombre_caso_test1=char(tabla_lista_casos{i,1})
maximo_porcentaje=max(porcentaje_positivas)
if(indice_pareja_test~=0)
    disp('Caso con pareja');
    pause;
end

pause;

end

```

A.2. Script 2: normaliza_o_estandariza

```

function [matriz_datos_entrenamiento
matriz_datos_test]=normaliza_o_estandariza(matriz_FA_entrenamiento,mat
riz_FA_test,matriz_FC_entrenamiento,matriz_FC_test,opcion,subopcion)

% Con esta funcion hacemos dos cosas:
% 1- Nos quedamos solamente con la FA, solamente con la FC o con las
dos
% cosas dependiendo de la variable "opcion" seleccionada
% 2- Dejamos los datos como estan, o bien los normalizamos o
estandarizamos. Normalizarlos significa colocarlos entre 0 y 1.
% Estandarizarlos significa hacer que tengan media nula y desviacion
% unitaria.
% Hay que tener en cuenta dos cosas importantes:
% - Lo que hacemos es normalizar o estandarizar individualmente cada
una
% de las características del vector de características (considerando
el

```

```

% vector formado por una caracteristica a lo largo de todos los casos
de
% entrenamiento).
% - Para normalizar o estandarizar los datos de test, lo hacemos
utilizando
% los valores que obtuvimos para los datos de entrenamiento (porque si
no
% estaríamos realizando una normalizacion/estandarizacion diferente
para
% los datos de entrenamiento y para los datos de test)

for i=1:size(matriz_FA_entrenamiento,2)

    if(opcion==1)          % Solamente usamos la FC

        if(subopcion==1)      % No normalizar o estandarizar

matriz_datos_entrenamiento(:,i)=matriz_FC_entrenamiento(:,i);
            matriz_datos_test(:,i)=matriz_FC_test(:,i);

        elseif(subopcion==2)    % Normalizamos

matriz_datos_entrenamiento(:,i)=(matriz_FC_entrenamiento(:,i)-
min(matriz_FC_entrenamiento(:,i)))/(max(matriz_FC_entrenamiento(:,i))-
min(matriz_FC_entrenamiento(:,i)));
            matriz_datos_test(:,i)=(matriz_FC_test(:,i)-
min(matriz_FC_entrenamiento(:,i)))/(max(matriz_FC_entrenamiento(:,i))-
min(matriz_FC_entrenamiento(:,i)));

        elseif(subopcion==3)    % Estandarizamos

matriz_datos_entrenamiento(:,i)=(matriz_FC_entrenamiento(:,i)-
mean(matriz_FC_entrenamiento(:,i)))/std(matriz_FC_entrenamiento(:,i));
            matriz_datos_test(:,i)=(matriz_FC_test(:,i)-
mean(matriz_FC_entrenamiento(:,i)))/std(matriz_FC_entrenamiento(:,i));

        end

    elseif(opcion==2)    % Solamente usamos la FA

        if(subopcion==1)      % No normalizar o estandarizar

matriz_datos_entrenamiento(:,i)=matriz_FA_entrenamiento(:,i);
            matriz_datos_test(:,i)=matriz_FA_test(:,i);

        elseif(subopcion==2)    % Normalizamos

matriz_datos_entrenamiento(:,i)=(matriz_FA_entrenamiento(:,i)-
min(matriz_FA_entrenamiento(:,i)))/(max(matriz_FA_entrenamiento(:,i))-
min(matriz_FA_entrenamiento(:,i)));
            matriz_datos_test(:,i)=(matriz_FA_test(:,i)-
min(matriz_FA_entrenamiento(:,i)))/(max(matriz_FA_entrenamiento(:,i))-
min(matriz_FA_entrenamiento(:,i)));

        elseif(subopcion==3)    % Estandarizamos

matriz_datos_entrenamiento(:,i)=(matriz_FA_entrenamiento(:,i)-
mean(matriz_FA_entrenamiento(:,i)))/std(matriz_FA_entrenamiento(:,i));

```

```

        matriz_datos_test(:,i)=(matriz_FA_test(:,i)-
mean(matriz_FA_entrenamiento(:,i)))/std(matriz_FA_entrenamiento(:,i));

    end

    elseif(opcion==3)    % Usamos tanto FC como FA

        if(subopcion==1)        % No normalizar o estandarizar

matriz_datos1_entrenamiento(:,i)=matriz_FC_entrenamiento(:,i);
        matriz_datos1_test(:,i)=matriz_FC_test(:,i);

matriz_datos2_entrenamiento(:,i)=matriz_FA_entrenamiento(:,i);
        matriz_datos2_test(:,i)=matriz_FA_test(:,i);

        elseif(subopcion==2)    % Normalizamos

matriz_datos1_entrenamiento(:,i)=(matriz_FC_entrenamiento(:,i)-
min(matriz_FC_entrenamiento(:,i)))/(max(matriz_FC_entrenamiento(:,i))-
min(matriz_FC_entrenamiento(:,i)));
        matriz_datos1_test(:,i)=(matriz_FC_test(:,i)-
min(matriz_FC_entrenamiento(:,i)))/(max(matriz_FC_entrenamiento(:,i))-
min(matriz_FC_entrenamiento(:,i)));

matriz_datos2_entrenamiento(:,i)=(matriz_FA_entrenamiento(:,i)-
min(matriz_FA_entrenamiento(:,i)))/(max(matriz_FA_entrenamiento(:,i))-
min(matriz_FA_entrenamiento(:,i)));
        matriz_datos2_test(:,i)=(matriz_FA_test(:,i)-
min(matriz_FA_entrenamiento(:,i)))/(max(matriz_FA_entrenamiento(:,i))-
min(matriz_FA_entrenamiento(:,i)));

        elseif(subopcion==3)    % Estandarizamos

matriz_datos1_entrenamiento(:,i)=(matriz_FC_entrenamiento(:,i)-
mean(matriz_FC_entrenamiento(:,i)))/std(matriz_FC_entrenamiento(:,i));
        matriz_datos1_test(:,i)=(matriz_FC_test(:,i)-
mean(matriz_FC_entrenamiento(:,i)))/std(matriz_FC_entrenamiento(:,i));

matriz_datos2_entrenamiento(:,i)=(matriz_FA_entrenamiento(:,i)-
mean(matriz_FA_entrenamiento(:,i)))/std(matriz_FA_entrenamiento(:,i));
        matriz_datos2_test(:,i)=(matriz_FA_test(:,i)-
mean(matriz_FA_entrenamiento(:,i)))/std(matriz_FA_entrenamiento(:,i));

    end

end

end

if(opcion==3)

matriz_datos_entrenamiento=[matriz_datos1_entrenamiento,matriz_datos2_
entrenamiento];
    matriz_datos_test=[matriz_datos1_test,matriz_datos2_test];
end

```

A.3. Script 3: compara_indices_iguales_encontrados

```
function [tasa_acierto_1
tasa_acierto_2]=compara_indices_iguales_encontrados(indices_iguales_es
timado,indices_iguales_verdadero,indices_distintos_estimado,indices_di
stintos_verdadero)

% Comparamos los hallazgos de casos iguales que hemos encontrado con
los
% que son ciertos. Describimos dos tipos de acierto:
% - tasa_acierto_1: Porcentaje de casos en los que habia una pareja y
en
% efecto
% la hemos clasificado como pareja (dividimos entre num de casos con
% pareja)
% - tasa_acierto_2: Porcentaje de casos en los que la comparacion era
entre
% casos distintos, y en efecto decidimos que eran distintos

contador=0;
for i=1:length(indices_iguales_verdadero)
    mi_indice=indices_iguales_verdadero(i);
    ind=find(indices_iguales_estimado==mi_indice);
    if(length(ind)>=1) % Esa pareja sí se encontro
        contador=contador+1;
    else % Esa pareja no se encontro
    end
end
tasa_acierto_1=contador/length(indices_iguales_verdadero);

contador=0;
for i=1:length(indices_distintos_verdadero)
    mi_indice=indices_distintos_verdadero(i);
    ind=find(indices_distintos_estimado==mi_indice);
    if(length(ind)>=1) % Lo que se encontro era cierto
        contador=contador+1;
    else % Lo que se encontro NO era cierto
    end
end
tasa_acierto_2=contador/length(indices_distintos_verdadero);
```

A.4. Script 4: calcula_umbral_optimo

```
function T=calcula_umbral_optimo(media1,sigma1,media2,sigma2,dibuja)

% Calculamos el umbral optimo para separar dos pdfs gaussianas,
% asumiendo
% que la probabilidad a priori de ambas pdfs es la misma. Tenemos por
% lo
% tanto que igualar las pdfs, porque el umbral optimo se encuentra en
% la
% interseccion de ambas curvas
% Utilizo una expresion tomada de (pag. 6)
% http://www.math.tau.ac.il/~turkel/notes/threshold.pdf

if(dibuja==1)
    %Dibujo auxiliar, si fuera necesario
    x1=linspace(media1-5*sigma1,media1+5*sigma1,1000);
    x2=linspace(media2-5*sigma2,media2+5*sigma2,1000);
    y1=normpdf(x1,media1,sigma1);
    y2=normpdf(x2,media2,sigma2);
    h=figure(1);
    clf;
    set(h,'Color',[1 1 1]);
    plot(x1,y1);
    hold on;
    plot(x2,y2,'r');
end

P1=1/2;
P2=1/2;

% Tenemos una ecuacion cuadratica, donde los coeficientes son lo
% siguiente:
A=sigma1^2 - sigma2^2;
B=2*(media1*sigma2^2 - media2*sigma1^2);
C=sigma1^2*media2^2 - sigma2^2*media1^2 +
2*sigma1^2*2*sigma2^2*log((sigma2*P1)/(sigma1*P2));

if(abs(A)<eps)
    T=-C/B;
else
```

```
% Resolvemos la ec. cuadratica
T1=(-B+sqrt(B^2-4*A*C))/(2*A);
T2=(-B-sqrt(B^2-4*A*C))/(2*A);

% Una de las dos soluciones estara entre las dos medias, asi que
sera la
% buena
if( (T1>(min(media1,media2))) & (T1<max(media1,media2)) )
    T=T1;
else
    T=T2;
end
end
```