



**UNIVERSIDAD DE CONCEPCIÓN
FACULTAD DE INGENIERÍA
DEPARTAMENTO DE INGENIERÍA ELÉCTRICA**



USO DE AUTOENCODER PARA LA SEGMENTACIÓN Y SIMULACIÓN DE FIBRAS CEREBRALES

POR

Sebastián Andrés Navarrete Caro

Informe Final Memoria de Título presentada a la Facultad de Ingeniería de la Universidad de Concepción para optar al grado académico de Ingeniero Civil Biomédica

Profesora guía
Prof. Pamela Guevara
Prof. Cecilia Hernández

Comisión
Dr. Sebastián Godoy

Agosto 23, 2023
Concepción
(Chile)

© 2023 Sebastián Andrés Navarrete Caro

© 2023 Sebastián Andrés Navarrete Caro

Ninguna parte de esta memoria puede reproducirse o transmitirse bajo ninguna forma o por ningún medio o procedimiento, sin permiso por escrito del autor.

Agradecimientos

A mis padres, por todo el esfuerzo que hicieron durante toda mi vida para poder brindarme una buena vida y educación. Mis logros son el reflejo de sus sacrificios.

A mis amigos de toda la vida, gracias por siempre estar ahí, por ayudarme a despejar la mente y sacarme una sonrisa en los días más abrumadores.

A Natalia, por ser una maravillosa persona que alegra mis días con tu amor. Tenerte a mi lado mientras realizaba este trabajo me daba la energía necesaria para avanzar.

A la profesora Pamela Guevara, por el conocimiento brindado y por proporcionarme las herramientas necesarias para llevar a cabo este trabajo. De otra forma, este trabajo no sería posible.

Resumen

El uso de neuroimágenes ha experimentado un gran avance en las últimas décadas, con una amplia variedad de técnicas e investigaciones que respaldan el estudio y diagnóstico de trastornos cerebrales en diversas enfermedades. Entre estas técnicas, las tractografías cerebrales, obtenidas mediante resonancia magnética por difusión, permiten visualizar en tres dimensiones las fibras cerebrales y estudiar la conectividad entre diferentes regiones del cerebro. Sin embargo, el procesamiento de las tractografías resulta desafiante debido a la alta dimensionalidad y complejidad de la información, ya que un conjunto de datos de tractografías puede presentar cientos de miles o incluso millones de fibras cerebrales con formas diversas.

El Autoencoder es un modelo generativo de aprendizaje profundo (Deep Learning) que se entrena para reducir la dimensionalidad de los datos a un espacio latente y luego reconstruirlos a partir de este. Este tipo de modelo tiene diversas aplicaciones, y una de ellas es el procesamiento y optimización de problemas a través del espacio latente. En esta memoria de título, se propone entrenar un Autoencoder utilizando las tractografías de la base de datos ARCHI, que contiene 79 tractografías de adultos sanos, con el objetivo de segmentar fascículos y simular fibras cerebrales realistas utilizando el espacio latente.

La segmentación de los fascículos se logró usando el algoritmo de Machine Learning llamado Radius Neighbor Classifier, el cual permite predecir la etiqueta de nueva información comparándola con la distancia radial de los datos de entrenamiento. El modelo se entrenó utilizando la información en el espacio latente de las fibras cerebrales de 5 tractografías segmentadas por medio del algoritmo [1].

La simulación de los fascículos se realizó mediante la selección de fibras semilla y la interpolación entre ellas en el espacio latente. Se emplearon técnicas de clustering para obtener centroides e interpolar las fibras con respecto a estos, variando así el radio de curvatura de los fascículos. Además, se usaron algoritmos de clustering para obtener distintos fascículos de los datos de tractografía para poder simularlos. Para llevar la simulación desde el espacio latente al formato original se usa el decoder del modelo.

Los resultados mostraron que el algoritmo de segmentación propuesto obtiene, en promedio, una mayor cantidad de fibras por fascículo en comparación con el algoritmo de [1], mejorando la segmentación para 25 de 38 fascículos del atlas de los cuales se puede observar una mejora significativa para los fascículos talámicos, cingulados y arqueados. Si bien el modelo obtiene una menor cantidad de fibras para 13 de los 38 fascículos, se observa una mejora en la segmentación para algunos de estos, como los fascículos del cuerpo calloso, debido a que el modelo obtiene una menor cantidad de fibras atípica y ruidosas que no forman parte del grupo.

Abstract

The use of neuroimaging has experienced significant advancements in recent decades, with a wide variety of techniques and research supporting the study and diagnosis of brain disorders in various diseases. Among these techniques, brain tractography obtained through diffusion magnetic resonance imaging allows for the visualization of brain fibers in three dimensions and the study of connectivity between different regions of the brain. However, processing tractography data is challenging due to the high dimensionality and complexity of the information, as a healthy individual can have hundreds of thousands or even millions of diverse-shaped brain fibers.

An Autoencoder is a generative Deep learning model that is trained to reduce the dimensionality of data to a latent space and then reconstruct it from there. This type of model has various applications, and one of them is processing and optimizing problems through the latent space. In this thesis, the proposal is to train an Autoencoder using tractography data from the ARCHI database, which contains 79 tractography scans from healthy adults, with the aim of segmenting fascicles and simulating realistic brain fibers using the latent space.

Segmentation of the fascicles was achieved using a machine learning algorithm called Radius Neighbor Classifier, which predicts the label of new information by comparing it with the radial distance of the training data. The model was trained using information from the latent space of the brain fibers of 5 segmented tractography scans through the algorithm [1].

Simulation of the fascicles was performed by selecting seed fibers and interpolating between them in the latent space. Clustering techniques were used to obtain centroids and interpolate the fibers with respect to these centroids, thus varying the curvature radius of the fascicles. Additionally, clustering algorithms were used to obtain different fascicles from the tractography data for simulation. The decoder of the model is used to bring the simulation from the latent space to the original format.

The results showed that the proposed segmentation algorithm obtains, on average, a greater number of fibers per fascicle compared to the algorithm in [1], improving the segmentation for 25 out of 38 fascicles in the atlas. A significant improvement can be observed for thalamic, cingulate, and arcuate fascicles. Although the model yields a smaller number of fibers for 13 out of the 38 fascicles, an improvement in segmentation is observed for some of these, such as the fascicles of the corpus callosum. This is because the model obtains a smaller number of atypical and noisy fibers that are not part of the group.

Tabla de contenidos

CAPITULO 1. INTRODUCCIÓN.....	14
1.1 INTRODUCCIÓN GENERAL	14
1.2 OBJETIVOS	15
1.2.1 <i>Objetivo General</i>	15
1.2.2 <i>Objetivos Específicos</i>	15
1.3 ALCANCES Y LIMITACIONES	15
1.4 METODOLOGÍA	15
1.4.1 <i>Entrenamiento del Autoencoder</i>	16
1.4.2 <i>Segmentación de Fascículos</i>	17
1.4.3 <i>Simulación de fibras cerebrales</i>	17
1.5 TEMARIO	18
CAPITULO 2. MARCO TEÓRICO	19
2.1 INTRODUCCIÓN	19
2.2 SISTEMA NERVIOSO CENTRAL.....	19
2.2.1 <i>Conceptos generales</i>	19
2.2.2 <i>Neuronas</i>	19
2.2.3 <i>Materia Blanca y Materia Gris</i>	20
2.2.4 <i>Fibras Cerebrales</i>	21
2.3 IMAGEN DE RESONANCIA MAGNÉTICA POR DIFUSIÓN	22
2.3.1 <i>Imágenes de Tensor de Difusión (Diffusion Tensor Imaging, DTI)</i>	23
2.3.2 <i>Imágenes de Alta Resolución Angular (High Angular Resolution Diffusion Imaging, HARDI)</i>	24
2.4 SISTEMAS DE COORDENADAS	25
2.5 TRACTOGRAFÍA	26
2.5.1 <i>Segmentación de Fascículos</i>	27
2.6 ALGORITMOS DE AGRUPAMIENTO DE DATOS (ALGORITMOS DE CLUSTERING).....	27
2.7 INTELIGENCIA ARTIFICIAL	29
2.7.1 <i>Redes Neuronales Artificiales (Artificial Neural Networks, ANN)</i>	31

2.7.2	<i>Redes Neuronales Convolucionales (Convolutional Neural Networks, CNNs)</i>	32
2.7.2.1	<i>Capa Convolutiva</i>	33
2.7.2.2	<i>Capa de Agrupación</i>	35
2.7.2.3	<i>Capa Densa (Fully connected layer)</i>	36
2.7.2.4	<i>Capa de Función de activación</i>	36
2.8	HIPERPARÁMETROS	37
2.9	AUTOENCODER.....	38
2.10	TRABAJOS PREVIOS	38
2.10.1	<i>Simulación de Fibras</i>	38
2.10.2	<i>Autoencoder</i>	40
2.11	DISCUSIÓN	42
CAPITULO 3. BASE DE DATOS Y CÓDIGOS EXISTENTES.....		43
3.1	INTRODUCCIÓN	43
3.2	BASE DE DATOS	43
3.2.1	<i>Datos ARCHI</i>	43
3.2.2	<i>Datos de adolescentes con y sin autismo</i>	43
3.2.3	<i>Atlas de Fascículos de Materia Blanca Profunda</i>	44
3.3	CÓDIGOS EXISTENTES	45
3.4	DISCUSIÓN	45
CAPITULO 4. DISEÑO DEL AUTOENCODER Y ENTRENAMIENTO DEL MODELO		46
4.1	INTRODUCCIÓN	46
4.2	METODOLOGÍA.....	46
4.2.1	<i>Procesamiento de Bases de Datos</i>	46
4.2.2	<i>Programación de la Estructura del Autoencoder</i>	48
4.2.3	<i>Optimización de Hiperparámetros</i>	49
4.2.4	<i>Entrenamiento del Autoencoder</i>	50
4.3	DISCUSIÓN	52
CAPITULO 5. SIMULACIÓN DE FIBRAS		53
5.1	INTRODUCCIÓN	53
5.2	METODOLOGÍA.....	53
5.3	SELECCIÓN DE FASCÍCULOS.....	53
5.4	VARIACIÓN DE RADIO DE CURVATURA DEL FASCÍCULO.....	56
5.5	SIMULACIÓN DE FIBRAS CEREBRALES.....	58

5.6 DISCUSIÓN	59
CAPITULO 6. SEGMENTACIÓN DE FIBRAS MEDIANTE CLASIFICADOR	60
6.1 INTRODUCCIÓN	60
6.2 CREACIÓN DE BASE DE DATOS PARA EL CLASIFICADOR	60
6.3 CLASIFICADOR	61
6.4 DISCUSIÓN	65
CAPITULO 7. RESULTADOS	66
7.1 INTRODUCCIÓN	66
7.2 SIMULACIÓN DE FIBRAS CEREBRALES	66
7.3 SEGMENTACIÓN DE FIBRAS CEREBRALES	68
7.3.1 Base de Datos ARCHI	68
7.3.2 Base de Datos de sujetos con autismo	71
CAPITULO 8. CONCLUSIONES.....	73
8.1 DISCUSIÓN	73
8.2 CONCLUSIONES	74
8.3 TRABAJO FUTURO	75
CAPITULO 9. GLOSARIO	76
CAPITULO 10. REFERENCIAS.....	77
ANEXO A.	83
A. I SEGMENTACIÓN DE FASCÍCULOS	83
A. II MÉTRICAS DE EVALUACIÓN PARA LA SIMULACIÓN DE FIBRAS	86
A. III RESULTADOS.....	87

Índice de Tablas

4.1	Estructura del Autoencoder. Para todas las capas de convolución se usa un tamaño de kernel de 3, padding de 1 y dilatation de 1.	49
4.2	Combinaciones de hiperparámetros seleccionados por Optuna y su desempeño.	50
6.3	Cantidad de fibras segmentadas por fascículos	63
7.4	Distancia media y Puntaje BA de las simulaciones de fascículos	67
7.5	Comparación de la cantidad de fibras para los dos métodos de segmentación.	72
A.1	Cantidad de fibras segmentadas por fascículo.	83

Índice de Figuras

1.1	Diagrama de trabajo.	16
2.2	Anatomía de una neurona.	20
2.3	Materia Blanca y Gris en el Sistema Nervioso Central.	20
2.4	Ilustración de algunos fascículos del cerebro.	21
2.5	Secuenciaciones de imágenes de resonancia magnética (MRI)	22
2.6	Ejemplo ilustrativo de las distintas imágenes de resonancia magnética por difusión necesarias para poder calcular el tensor de difusión.	23
2.7	Estimación de la orientación de distintas configuraciones de fibras para valores del tensor de difusión (DT) y función de distribución de orientación de fibras (fODF).	24
2.8	Ilustración de ejemplo que muestra como un mismo punto en MNI puede ser dos puntos distintos en Tailairach.	25
2.9	Tractografía de fibras cerebrales	26
2.10	Atlas de fascículos de materia blanca superficial y segmentación de un sujeto.	27
2.11	Comparación de distintos algoritmos de clustering de fibras cerebrales	29
2.12	Diagrama de los conceptos de Inteligencia Artificial, Machine Learning y Deep Learning.	30
2.13	(a) Estructura de un perceptron. (b) Red de perceptrones.	32
2.14	Ejemplo de estructura básica de modelo de CNN.	33
2.15	Ejemplo de un filtro 3x3 a una imagen de 6x6 píxeles.	33
2.16	Explicación visual de los parámetros de las capas de convolución. a) Stride. b) Padding. C) Dilatation.	34
2.17	Aplicación de capa de agrupación con técnica del valor máximo y promedio.	35
2.18	Gráfica y ecuación de las funciones de activación Sigmoidea y ReLU.	36
2.19	Estructura general de un Autoencoder.	38
2.20	Procesos para la selección de fibras a simular del algoritmo de Poo et al. [2].	39
2.21	Esquema generar para la simulación de fascículos del algoritmo de Poo et al. [2].	39
2.22	Estructura del Autoencoder LSTM [3].	40
2.23	Estructura del Autoencoder de FINTA	41

	XI
2.24 Identificación de fibras cerebrales no-plausibles de algoritmo FINTA [4]	42
3.25 Atlas de fascículos de materia blanca profunda	44
4.26 Diagrama de procesos del diseño y entrenamiento del Autoencoder	46
4.27 Remuestro de fibras cerebrales	47
4.28 Estructura del Autoencoder de FINTA modificada a 128 valores en el espacio, latente . .	48
4.29 Gráfica de la pérdida del modelo vs. las épocas	51
4.30 Comparación entre fibras orinales y la reconstrucción de las mismas por el Autoencoder.	52
5.31 Diagrama de procesos para la simulación de fibras.	53
5.32 Método del codo para determinar el valor óptimo de <i>epsilon</i>	54
5.33 Fascículos obtenidos mediante DBSCAN y visualizados usando t-SNE.	55
5.34 Distintas perspectivas de algunos fascículos obtenidos por DBSCAN en el espacio latente.	56
5.35 Ilustración de la obtención de centroides del fascículo Corticoespinal izquierdo.	56
5.36 Fascículo Corticoespinal con radio de curvatura variado.	58
5.37 Diagrama de procesos desde la selección de fascículo hasta el uso del decoder.	59
6.38 Representación t-SNE de la base de datos de segmentaciones.	61
6.39 Ejemplo ilustrativo de la clasificación de un nuevo dato usando ' <i>distance</i> ' como función del parámetro <i>weights</i> . a) Clasificación del nuevo dato en color negro. b) Nuevo dato clasificado con el color azul debido a que los datos más cercanos aportan un mayor peso a la clasificación.	62
6.40 Nuevas fibras que se obtienen para cada fascículo del hemisferio izquierdo al aumentar el radio de búsqueda del clasificador.	64
6.41 Segmentación de los fascículos Corticoespinal (CST), fascículos arqueado (AF) y cuer- po calloso (CC) del mismo sujeto, obtenido de distintos grupos de investigación con protocolos y algoritmos distintos.	65
7.42 Gráfico de cajas comparando las cantidades de fibras por fascículos para ambas segmen- taciones.	69
7.43 Visualización de los fascículos segmentados del sujeto 020 y 022 para ambos métodos .	70
A.1 Segmentación de fascículos talámicos, cingulados y arqueados del hemisferio izquierdo para distintos valores de radio de búsqueda.	84
A.2 Segmentación del fascículo corticoespinal, uncinados, fornix, frontoocipital, longitudinal del hemisferio izquierdo y de los fascículos del cuerpo calloso, para distintos valores de radio de búsqueda.	85
A.3 Valores BA para distintos fascículos con θ igual a 5 y 15.	86

A.4	Imágenes de las simulaciones del fascículo corticoespinal izquierdo. El recuadro en negro muestra el fascículo original en rojo y la simulación en color verde	87
A.5	Imágenes de las simulaciones del fascículo arqueado anterior izquierdo. El recuadro en negro muestra el fascículo original en rojo y la simulación en color verde	88
A.6	Imágenes de las simulaciones del fascículo del cuerpo calloso GENU. El recuadro en negro muestra el fascículo original en rojo y la simulación en color verde	89
A.7	Imágenes de las simulaciones del fascículo fronto-occipital inferior derecho. El recuadro en negro muestra el fascículo original en rojo y la simulación en color verde	90
A.8	Imágenes de las simulaciones del fascículo longitudinal inferior izquierdo. El recuadro en negro muestra el fascículo original en rojo y la simulación en color verde	91
A.9	Visualización de los fascículos segmentados del sujeto 011 para ambos métodos	92
A.10	Visualización de los fascículos segmentados del sujeto 026 y 050 para ambos métodos	93
A.11	Visualización de los fascículos segmentados del sujeto 052 y 060 para ambos métodos	94
A.12	Visualización de los fascículos segmentados del sujeto hc100303 y lk090309 para ambos métodos	95
A.13	Visualización de los fascículos segmentados del sujeto nl100461 y ru100340 para ambos métodos	96

Capítulo 1. Introducción

1.1 Introducción General

El cerebro es el órgano más complejo e importante de nuestro sistema, responsable del funcionamiento y la estabilidad de todas las áreas de nuestro cuerpo. Comprender cómo actúa el cerebro y desvelar los misterios que se encuentran en su interior ha sido motivo de incontables noches sin dormir para cientos de miles de investigadores a lo largo de los siglos. Los primeros investigadores que intentaron comprender el cerebro debían diseccionar animales y humanos post mortem en busca de pistas. En la actualidad, gracias a la resonancia magnética y sus distintas secuencias, es posible analizar in vivo la anatomía, funcionalidad y conectividad de las diferentes áreas del cerebro. Mediante la resonancia magnética por difusión, es posible determinar parcialmente la estructura de la materia blanca de este, generando un mapa de las fibras cerebrales conocido como tractografía.

Debido a la gran cantidad de fibras en el cerebro, los investigadores utilizan algoritmos de clustering y segmentación para agrupar conjuntos de fibras. La segmentación tiene como objetivo agrupar las fibras con ubicación y forma similares en fascículos conocidos que conectan dos áreas específicas del cerebro [1]. Mientras que los algoritmos de clustering de fibras cerebrales agrupan las fibras en grupos según características y métricas comunes [5] [6] [7], pero la validación de estos es complicada debido a que los conjuntos de tractografías no presentan un etiquetado de las fibras. Por esto se han creado algoritmos de simulación de fibras cerebrales con el propósito de simular datos de tractografías de cerebro completo etiquetadas [3] [2].

Gracias a la resonancia magnética y al procesamiento de estas a través de diversos algoritmos, se ha logrado un mejor entendimiento del funcionamiento y la conectividad del cerebro, pero aún existen muchas complicaciones y desafíos por superar para comprender este complejo sistema. Con los grandes avances en inteligencia artificial en los últimos años, algunos investigadores han propuesto utilizar los Autoencoder, un tipo de algoritmo de aprendizaje profundo, para reducir la dimensionalidad de los conjuntos de tractografías y procesar la información en este espacio [4] [8].

En este trabajo se propone utilizar un Autoencoder basado en la estructura de FINTA [4], entrenado a partir de los conjuntos de tractografía de la base de datos ARCHI [9] con el objetivo de crear un nuevo método de segmentación y simulación de fibras cerebrales a partir del procesamiento de los datos de tractografías en el espacio latente del Autoencoder entrenado.

1.2 Objetivos

1.2.1 Objetivo General

Implementar un Autoencoder con la finalidad de procesar la información de los datos de tractografía, con el propósito de obtener un nuevo y mejorado método para la segmentación y simulación de fascículos, basándose en el procesamiento de datos mediante modelos de aprendizaje profundo.

1.2.2 Objetivos Específicos

- Entrenar un modelo de Autoencoder capaz de preservar la integridad de la información original durante la reconstrucción.
- Implementar un simulador de fibras cerebrales mediante Autoencoder y el uso del espacio latente.
- Implementar un método de segmentación de fascículos basados en un atlas, usando la información en el espacio latente de los datos de tractografías.
- Evaluar y comparar la efectividad de la segmentación propuesta con otros algoritmos de segmentación consolidados.

1.3 Alcances y Limitaciones

El Autoencoder entrenado debe simular fibras cerebrales con estructura y forma similares a las fibras de un individuo sano. Al aplicar el algoritmo de segmentación a los datos de tractografía, se espera que pueda clasificar los 38 fascículos presentes en el atlas de materia blanca profunda. Se compararán los resultados del algoritmo de segmentación propuesto con el algoritmo utilizado por el grupo de análisis de imágenes médicas de la Universidad de Concepción [1]. En cuanto al simulador de fibras, se requerirá el uso de al menos dos métricas para poder cuantificar los resultados de la simulación.

1.4 Metodología

Si bien este trabajo sigue una metodología lineal, para poder entender de mejor manera las distintas etapas y procesos, el diagrama de trabajo fue separado en las tres estructuras más importantes, como se

muestra en la Fig. 1.1. A partir de esta figura, se puede observar que al finalizar el entrenamiento del modelo, el diagrama de trabajo se bifurca en dos bloques distintos. Estos bloques funcionan de manera independiente y permiten la segmentación y simulación de fibras.

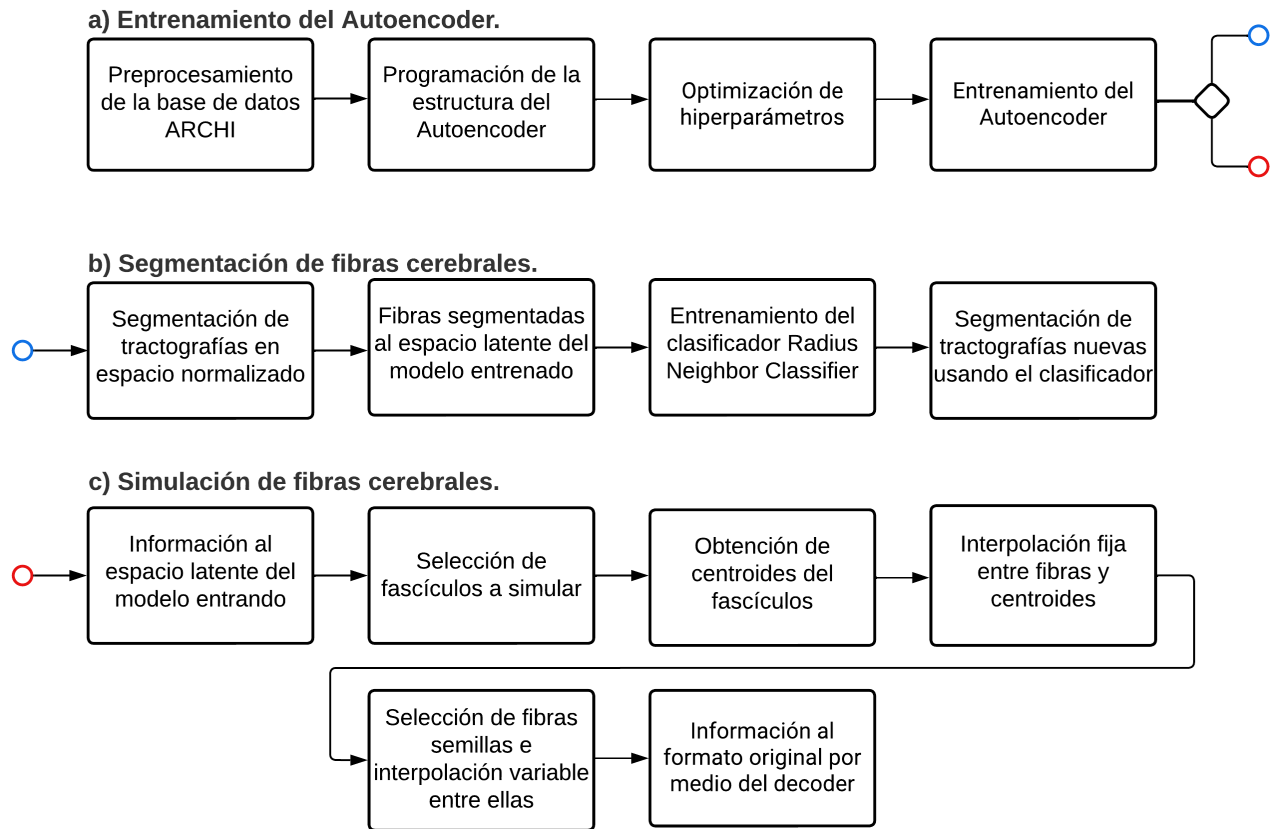


Fig. 1.1: Diagrama de trabajo.

1.4.1 Entrenamiento del Autoencoder

Para poder entrenar correctamente el Autoencoder, primero es necesario preprocesar los datos de las tractografías de la base de datos ARCHI. Esto implica estandarizar el tamaño de las fibras y mejorar el rendimiento del modelo. El preprocesamiento consta de los siguientes pasos:

- Remuestrear las fibras cerebrales a 256 puntos por fibra.
- Eliminar las fibras con una longitud menor a 15 mm.
- Reorientar las fibras en relación con el extremo que tenga la menor distancia al inicio.

En segundo lugar, se programa la estructura que tendrá el Autoencoder en Python. Posteriormente, se optimizan los hiperparámetros del Autoencoder para mejorar la eficacia y eficiencia del modelo.

Por último, se procede a entrenar el modelo utilizando 5 conjuntos de datos de tractografías previamente preprocesadas, durante 100 épocas cada uno.

1.4.2 Segmentación de Fascículos

La segmentación de los fascículos se logra utilizando el clasificador Radius Neighbor Classifier. Para entrenar este modelo de aprendizaje automático (ML), es necesario contar con una base de datos de tractografías segmentadas. Para ello, se utiliza el código de segmentación proporcionado por el laboratorio de visualización [1].

Antes de llevar a cabo la segmentación, es necesario normalizar el conjunto de tractografías y el atlas de segmentación a un espacio estandarizado. Posteriormente, se vuelve a llevar el conjunto de tractografías al espacio original y se procesa la información a través del codificador (encoder) del Autoencoder para obtener la información en el espacio latente.

El clasificador Radius Neighbor Classifier se entrena utilizando la información del espacio latente. Por último, se obtiene una nueva segmentación utilizando datos nuevos mediante el uso de este clasificador.

1.4.3 Simulación de fibras cerebrales

Una vez que el modelo ha sido entrenado, se lleva la información al espacio latente y se obtienen fascículos utilizando el algoritmo de clustering DBSCAN. A continuación, se selecciona el fascículo que se desea simular. Luego, se utiliza el algoritmo de clustering K-means para agrupar las fibras en sub-fascículos y obtener los centroides de cada grupo. Esto permite interpolar todas las fibras con relación a su respectivo centroide, lo que permite la variación del radio de curvatura del fascículo.

Luego, se seleccionan N fibras semilla y se interpolan entre ellas para obtener nuevas fibras. Este proceso se repite hasta obtener la cantidad de fibras correspondiente al fascículo que se desea simular. Por último, se añade ruido gaussiano a las fibras en el espacio latente para introducir pequeñas variaciones, y luego se lleva la información de vuelta a su formato original utilizando el decodificador del modelo, completando así la simulación.

1.5 Temario

Este trabajo presenta el siguiente temario:

- (i) Capítulo 1. Introducción: Introducción en forma general del trabajo, indicando los objetivos y alcances del proyecto.
- (ii) Capítulo 2. Marco Teórico: Se presenta el estudio bibliográfico con respecto a la anatomía del cerebro y técnicas de resonancia magnética para la obtención de tractografías, algoritmos de clustering e inteligencia artificial, enfatizando en el aprendizaje profundo o Deep learning. Además, se explican los trabajos en el estado del arte con respecto al uso de Autoencoder en las tractografías y simulación de fibras cerebrales.
- (iii) Capítulo 3. Base de Datos y Códigos Existentes: Se exponen las bases de datos de conjuntos de tractografías que se utilizaron para este trabajo y los códigos existentes usados.
- (iv) Capítulo 4. Diseño del Autoencoder y Entrenamiento del modelo: Se presentan los procesamientos que se llevaron a cabo para la programación y optimización de parámetros del Autoencoder junto al entrenamiento de este.
- (v) Capítulo 5. Simulación de Fibras: Se expone la implementación del Autoencoder para la simulación de fibras cerebrales y todas las etapas que se llevaron a cabo para simular fascículos con forma realista. Además, se detallan el método empleado para la obtención de fascículos.
- (vi) Capítulo 6. Segmentación de Fibras mediante clasificador: Se presentan la implementación del Autoencoder para segmentar fibras empleando el clasificador Radius Neighbors Classifier. Se explica la creación de la base de datos que se utilizó como entrenamiento del modelo y los procesos para el ajuste de parámetros del clasificador.
- (vii) Capítulo 7. Resultados: Se exponen los resultados obtenidos al usar el espacio latente para la simulación y segmentación.
- (viii) Capítulo 8. Conclusión: Se presentan las conclusiones finales del trabajo, junto con discusiones y posibles trabajos a futuro.

Capítulo 2. Marco Teórico

2.1 Introducción

En este capítulo se presenta una revisión conceptual con respecto a los Autoencoder, Inteligencia artificial, dMRI, tractografías y algoritmos de clustering, además, se realizó una revisión de trabajos previos con respecto a la simulación de fibras cerebrales y el uso de Autoencoder para el procesamiento de los datos de tractografías, con el objetivo de identificar posibles problemas y limitaciones de los métodos existentes, así como identificar áreas de mejora.

2.2 Sistema Nervioso Central

2.2.1 Conceptos generales

El Sistema Nervioso Central (SNC) está compuesto por el encéfalo y la médula espinal, ambos están recubiertos por tres capas de tejido llamadas meninges, las cuales los protegen. Además, se encuentran sumergidos en el líquido cefalorraquídeo, el cual desempeña funciones de protección, nutrición, regulación y eliminación de desechos.

2.2.2 Neuronas

Desde un punto de vista histológico, el SNC está compuesto por células nerviosas o neuronas y células gliales. Las neuronas son la unidad básica funcional del SNC, conformadas por el cuerpo celular o soma que contiene el núcleo, prolongaciones cortas llamadas dendritas, que generalmente reciben información hacia la célula, y prolongaciones más largas llamadas axones, encargados de transmitir la mayor parte de la información de salida. En algunas ocasiones, los axones están recubiertos por una vaina de mielina generada por células gliales especializadas, lo que aumenta la velocidad de transmisión de la información a lo largo de los axones. La comunicación y el intercambio de información entre dos neuronas se produce en las sinapsis, regiones donde la información se transmite desde el axón de una célula a las dendritas de otras células [10]. La Figura 2.2 muestra una ilustración de la anatomía mencionada.

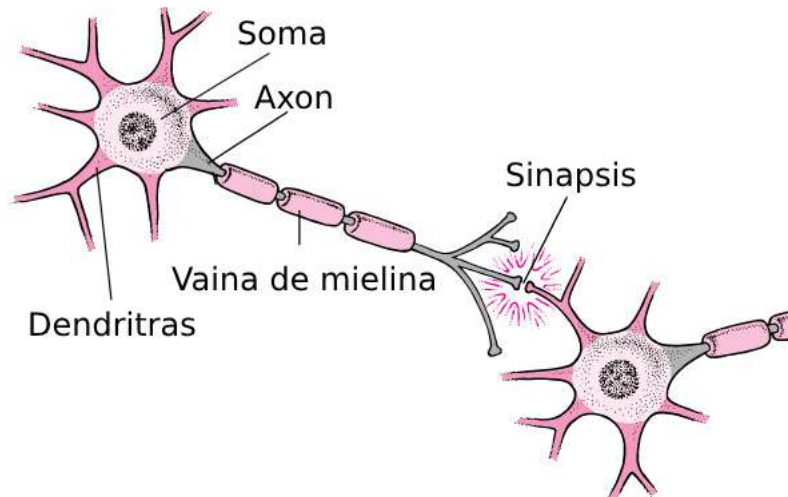


Fig. 2.2: Anatomía de una neurona.

2.2.3 Materia Blanca y Materia Gris

Las áreas del sistema nervioso central que están compuestas mayoritariamente por los somas de las neuronas se denominan materia gris, mientras que las zonas que presentan una mayor cantidad de axones con vainas de mielina se llaman materia blanca. Los hemisferios cerebrales están cubiertos por un manto de materia gris conocido como corteza cerebral. En la materia gris ocurre la mayoría de las sinapsis locales, mientras que en la materia blanca se transmite la información a distancias más amplias. Como se puede observar en la Figura 2.3, en los hemisferios cerebrales la materia gris de la corteza se encuentra en los exteriores, mientras que en la espina dorsal es la materia blanca la que está en la periferia [10].

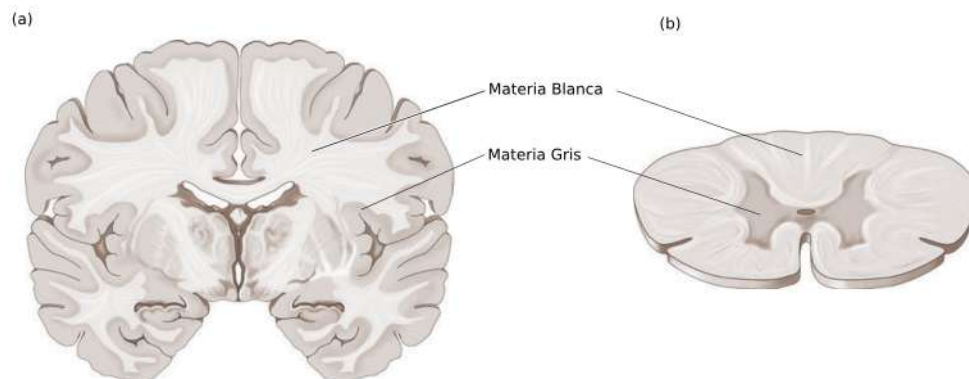


Fig. 2.3: Materia Blanca y Gris en el Sistema Nervioso Central. a) Hemisferios cerebrales. b) Medula espinal.

Imagen adaptada de [10], se reordenaron las figuras

2.2.4 Fibras Cerebrales

Se denomina fibras cerebrales a los axones de las neuronas que componen la Materia Blanca (White Matter, WM)) del cerebro. Estas fibras pueden tener longitudes muy variadas, pero generalmente se agrupan en fibras cortas, que se encuentran principalmente en la parte más superficial, y en fibras largas, que se ubican en la zona más profunda de la WM.

Los conjuntos de fibras cerebrales que tienen forma y ubicación similar, que conectan dos áreas del cerebro, se denominan fascículos. Tradicionalmente, los fascículos se clasifican en tres grupos. Los fascículos comisurales, que interconectan áreas corticales relacionadas de hemisferios distintos, como la comisura anterior y el cuerpo caloso, que es uno de los fascículos más voluminosos del cerebro. Los fascículos de asociación, los cuales conectan áreas corticales dentro del mismo hemisferio, como los fascículos cingulados, uncinados, frontooccipitales y longitudinales. Por último, están los fascículos de proyección, que conectan áreas corticales con los centros subcorticales, como el fascículo corticoespinal, que se conecta con la médula espinal [11]. La Fig. 2.4 muestra una ilustración de algunos de los fascículos mencionados.

Existen distintas técnicas que permiten analizar y entender la estructura y funcionamiento de la materia blanca. Actualmente, se estudia la estructura de la materia blanca mediante técnicas de Imagen de Resonancia Magnética por Difusión.

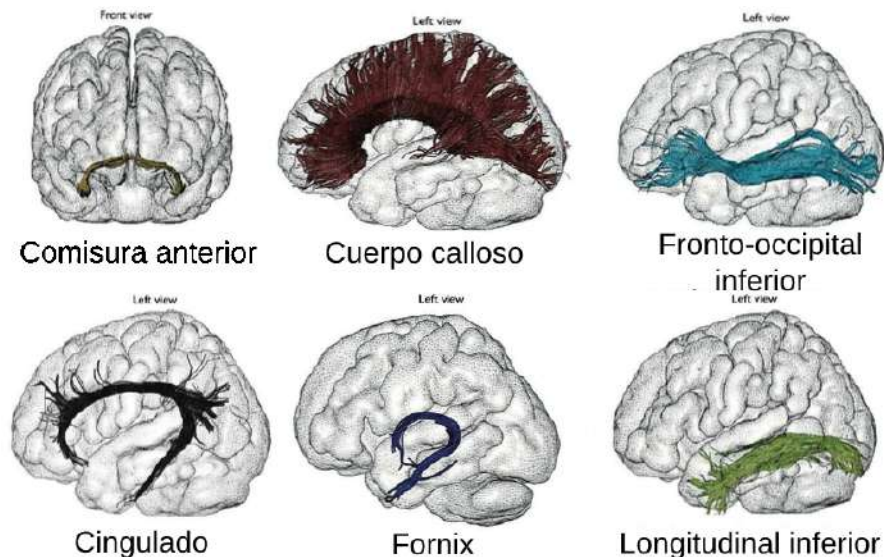


Fig. 2.4: Ilustración de algunos fascículos del cerebro.

Imagen adaptada de [12], se cortó la imagen y tradujo los nombres de los fascículos

2.3 Imagen de Resonancia Magnética por Difusión

La Imagen de Resonancia Magnética por Difusión o DWI es un tipo específico de secuencia de Imagen de Resonancia Magnética (Magnetic Resonance Imaging, MRI), como se puede observar en la Figura 2.5, la cual se utiliza extensamente en el estudio de la materia blanca (WM, por sus siglas en inglés). Esta técnica no invasiva in vivo se basa en el movimiento aleatorio de las partículas de agua en su entorno, conocido como movimiento Browniano [13]. Aunque las partículas de agua se mueven de manera aleatoria, en el cerebro se encuentran en un entorno restringido o anisotrópico, lo que significa que su trayectoria depende de la forma de los axones mielinizados. Por lo tanto, es posible estimar la trayectoria de las fibras cerebrales y de otros tejidos al calcular la orientación de las moléculas de agua.

Para calcular la direccionalidad de la difusión o la difusión anisotrópica, se usan métodos de difusión. Existen distintos métodos de difusión, cuyo uso varía dependiendo de la aplicación. Por ejemplo, las Imágenes de tensor de difusión (Diffusion Tensor Imaging, DTI) se emplean mayoritariamente en el ámbito clínico debido a su rapidez de adquisición, mientras que en la investigación se tienden a utilizar métodos más completos y complejos como la Imágenes de alta resolución angular (High Angular Resolution Diffusion Imaging, HARDI).

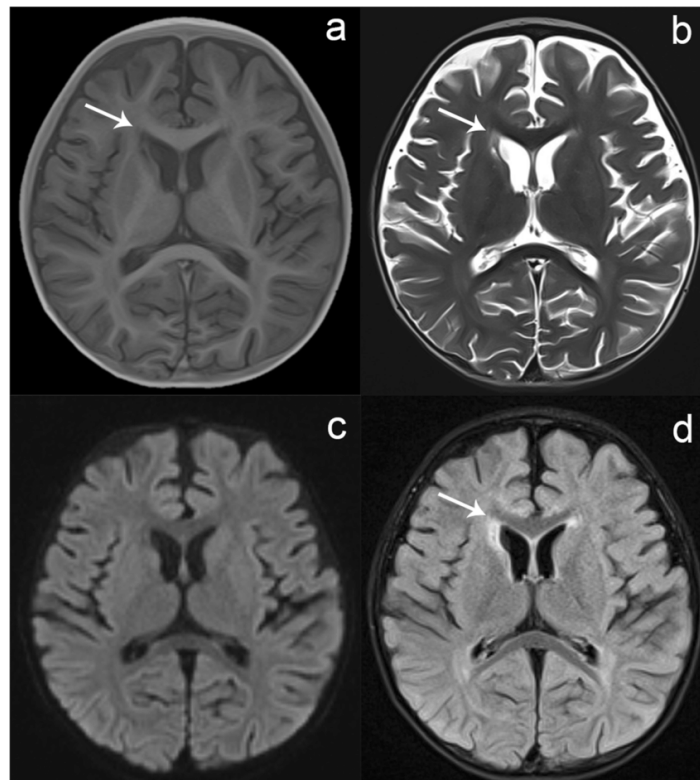


Fig. 2.5: Secuenciaciones de imágenes de resonancia magnética (MRI): a) T1; b) T2; c) DWI; d) T2 Flair.

2.3.1 Imágenes de Tensor de Difusión (Diffusion Tensor Imaging, DTI)

Las imágenes de tensor de difusión permite describir la difusión de las moléculas de agua asumiendo que siguen una distribución Gaussiana tridimensional mediante el tensor de difusión. Este tensor tiene una estructura simétrica de 3x3, con 6 incógnitas que corresponden a los valores propios λ_1 , λ_2 y λ_3 , y los vectores propios asociados, como se muestra en la ecuación 2.3.1. El valor propio más grande representa la difusión del agua a lo largo de la fibra, mientras que los otros dos valores propios representan la difusión perpendicular a la fibra [14]. Para calcular el tensor de difusión, se requieren al menos 7 imágenes de DWI distintas, donde una de ellas debe tener un valor bajo de difusión [15], como se muestra en la Figura 2.6.

$$\mathbf{DT} = \begin{bmatrix} D_{xx} & D_{xy} & D_{xz} \\ D_{yx} & D_{yy} & D_{yz} \\ D_{zx} & D_{zy} & D_{zz} \end{bmatrix} \longrightarrow \lambda_1, \lambda_2, \lambda_3, \mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3 \quad (2.3.1)$$

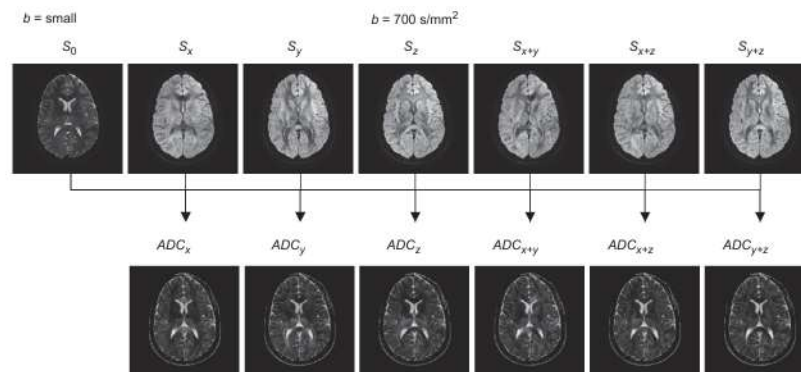


Fig. 2.6: Ejemplo ilustrativo de las distintas imágenes de resonancia magnética por difusión necesarias para poder calcular el tensor de difusión.

Imagen extraída de [15]

El tensor de difusión permite calcular el valor de la Anisotropía Fraccional (Fractional Anisotropy, FA), que describe el grado de anisotropía del tejido. Cuando el valor de FA es cercano a cero, el tejido presenta difusión isotrópica, es decir, la difusión es igual en todas las direcciones. Por otro lado, cuando el valor de FA se acerca a 1, el tejido presenta difusión altamente anisotrópica, lo que indica una dirección preferencial de la difusión.

2.3.2 Imágenes de Alta Resolución Angular (High Angular Resolution Diffusion Imaging, HARDI)

Las imágenes de alta resolución angular (High Angular Resolution Diffusion Imaging, HARDI) se desarrollaron con el objetivo de superar las limitaciones de DTI, como la naturaleza no Gaussiana de la difusión y la incapacidad de caracterizar adecuadamente la información cuando varias fibras se cruzan en el mismo voxel [16]. Además, estas imágenes proporcionan nuevas medidas de anisotropía más allá de FA y el tensor de difusión.

A diferencia de DTI, HARDI permite obtener estimaciones más robustas de la orientación de las fibras mediante el cálculo de la Función de Distribución de Orientación (ODF, por sus siglas en inglés) como se muestra en la Fig.2.7. La ODF es una representación matemática que describe la distribución de probabilidades de las orientaciones de las fibras dentro de un voxel. Proporciona una forma de modelar y cuantificar la complejidad de las orientaciones de las fibras dentro de un voxel [17]. Sin embargo, una desventaja de Imágenes de alta resolución angular es que requieren una mayor cantidad de imágenes ponderadas en difusión y un esquema específico de muestreo, como single-shell y multi-shell, lo cual implica un tiempo de adquisición más prolongado en comparación con DTI.

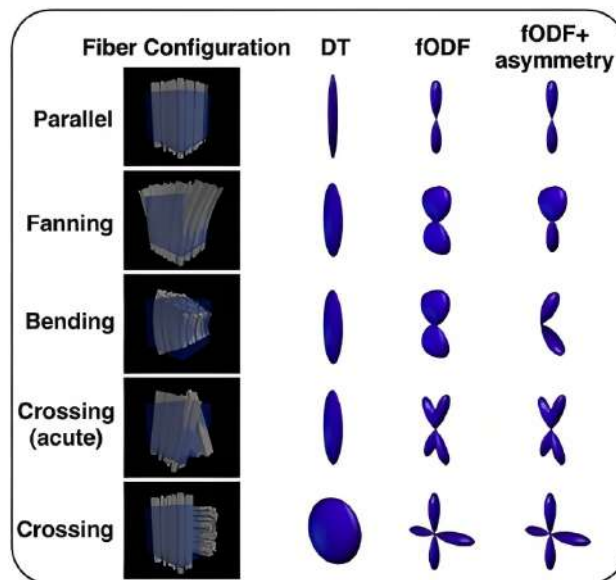


Fig. 2.7: Estimación de la orientación de distintas configuraciones de fibras para valores del tensor de difusión (DT) y función de distribución de orientación de fibras (fODF).

2.4 Sistemas de Coordenadas

Al obtener imágenes de resonancia magnética de diferentes sujetos o en diferentes ocasiones, es común encontrar que estas no están completamente alineadas y presentan variabilidad o movimiento. Además, los cráneos y, por ende, los cerebros de las personas pueden mostrar una alta variabilidad en sus proporciones. Por lo tanto, se requiere aplicar transformaciones a estas imágenes para llevarlas a un espacio de coordenadas común y poder analizar la información de manera adecuada.

Los dos espacios de coordenadas comúnmente utilizados son el espacio Talairach y el espacio MNI (Montreal Neurological Institute). El espacio Talairach fue creado en el año 1967 por Jean Talairach y Gabor Sziklia, y posteriormente actualizado en 1988 por Talairach y Tournoux. Este espacio se creó a partir de la disección del cerebro post-mortem de un solo sujeto. El sistema de coordenadas tiene su origen en la comisura anterior (AC), con los ejes x e y en el plano horizontal y el eje z en el plano vertical. El eje y está definido por la línea que conecta la parte inferior de la comisura posterior (PC) con la parte superior de la AC. El eje x está definido por la línea que pasa por el punto AC y es ortogonal a la línea AC-PC. El eje z está definido por la línea que pasa por la fisura interhemisférica y el punto AC [19].

En 1993 se creó el espacio MNI con el objetivo de obtener una representación más precisa del cerebro. Este espacio se generó al promediar las imágenes de resonancia magnética de cientos de sujetos. A lo largo de los años, se han desarrollado diversos atlas MNI, siendo el primero de ellos MNI305 [19].

Aunque los sistemas de coordenadas Talairach y MNI son ampliamente utilizados, al analizar información de distintos sujetos, ambos tienen que estar en el mismo espacio de coordenadas, ya que un mismo punto en el espacio MNI puede corresponder a diferentes puntos en el espacio Talairach para diferentes sujetos, como se ilustra en la Fig 2.8.

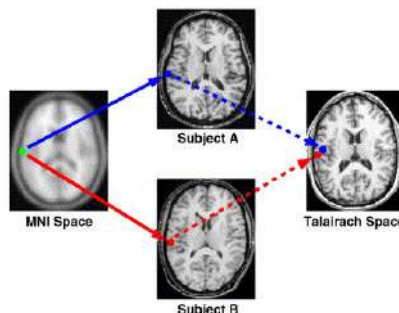


Fig. 2.8: Ilustración de ejemplo que muestra cómo un mismo punto en MNI puede ser dos puntos distintos en Talairach.

Imagen adaptada de [19], se extrajo de la figura original una porción relevante de la imagen

2.5 Tractografía

Las tractografías corresponden a conjuntos de algoritmos que permiten estimar la trayectoria de las fibras cerebrales a partir de la información obtenida de los métodos de difusión. Como resultado, se obtiene la ubicación de las fibras de la materia blanca en un espacio tridimensional. Los datos de las tractografías se pueden visualizar en una imagen 3D, como se observa en la Fig. 2.9, donde las distintas curvas representan las fibras cerebrales. Sin embargo, es importante tener en cuenta que estas representaciones son una aproximación de la trayectoria real debido a la presencia de ruido y/o problemas en la adquisición de los datos. Los datos de tractografías se utilizan para estudiar la conectividad de las distintas áreas del cerebro, con el objetivo de obtener un mayor entendimiento de su funcionamiento [20].

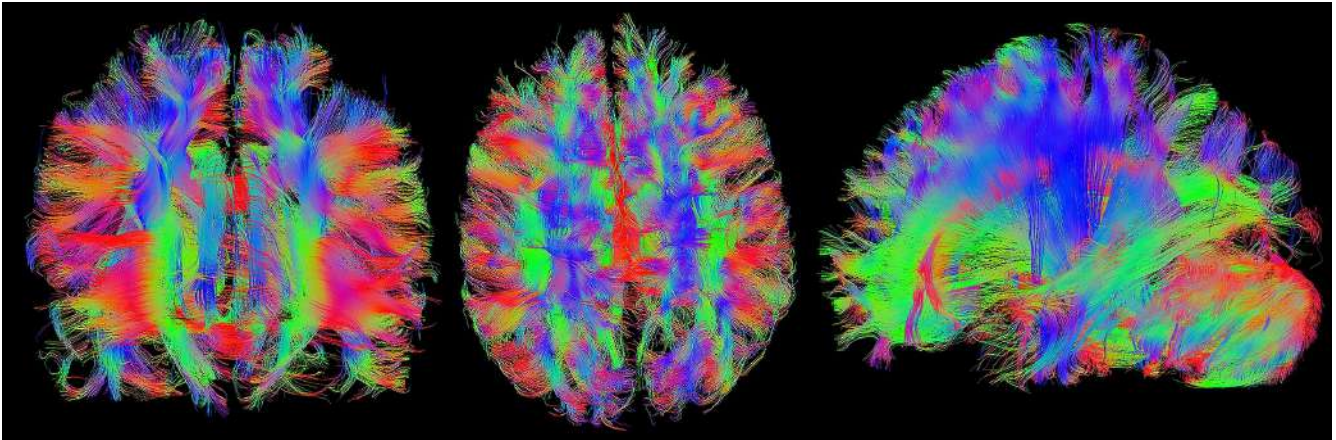


Fig. 2.9: Tractografía de fibras cerebrales

Además del estudio del cerebro, los datos de tractografías recientemente se han implementado junto con algoritmos de segmentación de fibras cerebrales en distintas neurocirugías, con el objetivo de obtener un mapa de las conexiones cerebrales del paciente para mejorar la planificación de la intervención y reducir posibles daños provocados. Además, se han comenzado a utilizar en la implementación de la estimulación cerebral profunda (Deep brain stimulation, DBS) para la selección preoperatoria de la ubicación del implante. En el futuro, también podrían utilizarse para identificar si los pacientes son aptos para recibir el implante [21].

2.5.1 Segmentación de Fascículos

Los métodos de segmentación etiquetan las fibras cerebrales en distintos fascículos relevantes. Existen 2 tipos de estrategias distintas para la segmentación. La primera elige diferentes regiones de interés anatómicas para identificar las fibras cerebrales que interconectan a estas. Este tipo de segmentación se puede lograr de forma automática o manual. La segunda estrategia elige un conjunto de fibras y utiliza un umbral de distancia para identificar las fibras con forma y posición similar [1]. Para este tipo de método, se suelen utilizar atlas de fascículos para poder identificar las fibras similares Fig. 2.10.

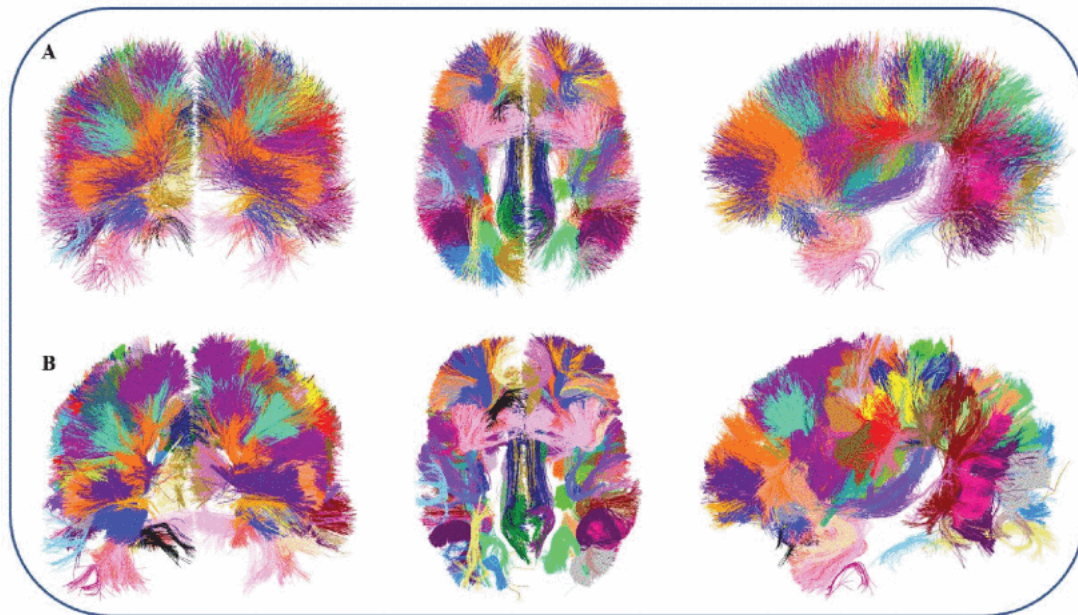


Fig. 2.10: Atlas de fascículos de materia blanca superficial y segmentación de un sujeto. A) Atlas de fascículos de materia blanca, este está compuesto por 50 fascículos por hemisferio. B) Segmentación de fibras de un sujeto.

Imagen extraída de [1]

2.6 Algoritmos de Agrupamiento de Datos (Algoritmos De Clustering)

En un mundo donde el acceso y obtención de información aumenta rápidamente, son necesarias herramientas para poder procesar y darle uso a esta data. Los algoritmos de agrupamiento o clustering son un tipo de clasificación no supervisada que tiene como objetivo determinar el agrupamiento natural de los datos, de tal forma que los datos de un mismo grupo tienen patrones análogos entre sí [22]. Existen variadas técnicas para agrupar la información en la cuales cada una utiliza algoritmos y principios distintos. A continuación se explican los tipos algoritmos de clustering más utilizados.

- **Jerárquico:** Este método busca construir una estructura con forma de árbol o dendrograma donde la información se agrupa según niveles sucesivos de similitud, basados en la proximidad de los datos. Al visualizar la agrupación en forma de árbol se tiene el nivel más alto o raíz, donde toda la información se encuentra en un solo grupo, y niveles más bajos donde hay una serie de nodos, hasta llegar a las hojas, donde cada hoja forma parte de su propio grupo. Este árbol o dendrograma se puede cortar en niveles intermedios para obtener distintos números de grupos.

Existen dos tipos de algoritmos de agrupación jerárquica: los aglomerativos y los divisivos. En los algoritmos jerárquicos aglomerativos, al inicio cada dato presenta su propio grupo y se van emparejando los grupos en función de la cercanía de distancia según un umbral establecido. Por otro lado, en los algoritmos jerárquicos divisores, toda la información forma parte del mismo grupo y se segmenta en diferentes clusters en función de la distancia que existe entre los datos[23][24].

- **Agrupamiento por partición:** Método de agrupación de datos no jerárquico e iterativo que tiene como objetivo segmentar la información en K grupos por medio de variadas técnicas, donde en general K es definido por el usuario previamente. Entre las técnicas más utilizadas están: (i) K-means que obtiene los grupos a partir de K centroides, los cuales son ubicados de manera tal que minimicen la suma de distancias entre cada dato y el centroide de su grupo o clúster, (ii) K-Medoid que funciona de manera similar a K-means, pero en vez de seleccionar un punto en el espacio como centroide utiliza la ubicación de los datos reales y selecciona como mediodo al dato de un grupo cuya disimilitud media a todos los objetos en el grupo es mínima. Este método presenta mejores resultados ante conjuntos de datos atípicos. Los algoritmos de agrupación por partición son de los más utilizados debido a simplicidad y efectividad [24].

- **Basados en densidad:**

Estos métodos, a diferencia de otros, no utilizan la cercanía de los datos para agrupar, sino que se basan en el grado de densidad que existe en un punto del espacio. Este grado de densidad se calcula principalmente a partir de dos parámetros: la densidad mínima de datos requerida para considerarse un grupo y la máxima distancia permitida entre dos datos para que sean considerados del mismo grupo.

Los algoritmos basados en densidad desempeñan un papel fundamental, ya que son ampliamente utilizados para agrupar información con estructura no lineal y excluyen el ruido debido a la baja densidad del mismo. Los dos algoritmos de agrupación basados en densidad más conocidos y utilizados son el DBSCAN (Density-Based Spatial Clustering of Applications with Noise) y el DENCLUE (Density-Based Clustering) [23][24].

En el ámbito de la Biomedicina, más específicamente en el campo de las tractografías y fibras cerebrales, se utilizan técnicas de clustering para obtener automáticamente fascículos de fibras cerebrales que presenten características anatómicas o estructurales comunes. Esto permite analizar y observar cómo las distintas áreas del cerebro están conectadas entre sí. En la siguiente Fig 2.11 se comparan los algoritmos de clustering de fibras FFClust [5], Guevara [25], QuickBundle [7] y QuickBundleX [6] con respecto a los fascículos del atlas de fascículos de fibras largas DWM [26].

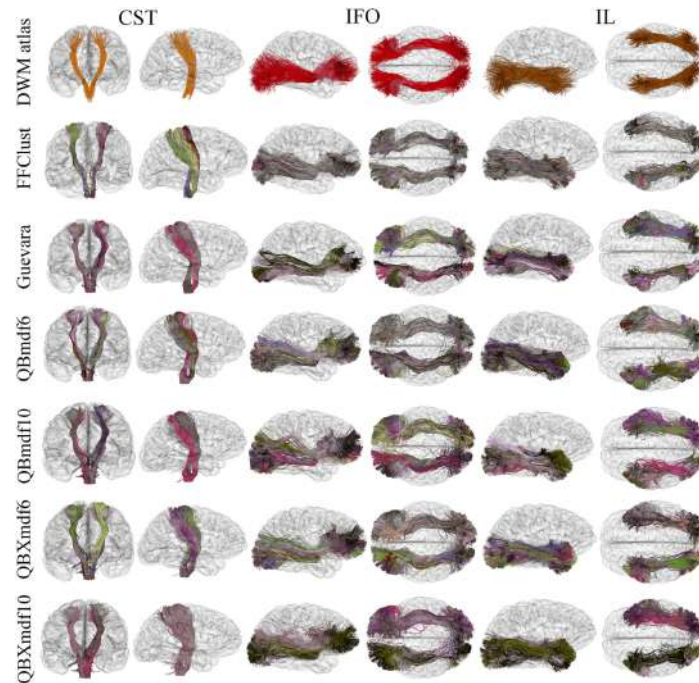


Fig. 2.11: Comparación de distintos algoritmos de clustering de fibras cerebrales para los fascículos cerebrospinal (CST), fascículo fronto-occipital (IFO) y fascículo longitudinal inferior (IL) del atlas de fascículos de fibras largas DWM.

Imagen extraída de [5]

2.7 Inteligencia Artificial

La mayor limitación de la mente humana es su capacidad para procesar una gran cantidad de datos en un corto período de tiempo. Los software de los ordenadores, a través de la aplicación de algoritmos, pueden adquirir una gran cantidad de “experiencia” en poco tiempo. De esta premisa nació la Inteligencia Artificial (Artificial Intelligence, IA), que en términos simples es la capacidad de los ordenadores para imitar el razonamiento humano [27]. La IA es la base de los conceptos de Aprendizaje de Máquinas (Machine Learning, ML) y Aprendizaje profundo, como se mostró en la Fig. 2.5.

Machine Learning es la aplicación de la IA que permite que el computador/máquina a través de algoritmos de programación pueda automáticamente aprender y mejorar a través de la experiencia. Para que algoritmo de ML pueda aprender de la experiencia es necesario una base de datos del cual pueda extraer características. Dependiendo de si la base de datos está etiquetada o no se puede dividir los algoritmos de ML en algoritmos de aprendizaje supervisado y no supervisado respectivamente [27].

El Aprendizaje de Máquinas (Machine Learning, ML) es la aplicación de la IA que permite que el ordenador/máquina aprenda y mejore automáticamente a través de la experiencia utilizando algoritmos de programación. Para que un algoritmo de ML pueda aprender de la experiencia, es necesario tener una base de datos de la cual extraer características. Dependiendo de si la base de datos está etiquetada o no, se pueden clasificar los algoritmos de ML como supervisados o no supervisados, respectivamente [27].

El Aprendizaje profundo (Deep Learning, DL) es una subclase de Aprendizaje de Máquinas (Machine Learning, ML) que presenta una estructura más compleja, inspirada en la biología de las Redes neuronales del cerebro humano. Las Redes Neuronales Artificiales (Artificial Neural Networks, ANNs) son la base de los algoritmos de DL, las cuales consisten en un conjunto de nodos o neuronas artificiales entrelazadas y en distintas capas de profundidad, que resuelven una serie de ecuaciones para llegar a una respuesta. Estas neuronas artificiales [27].

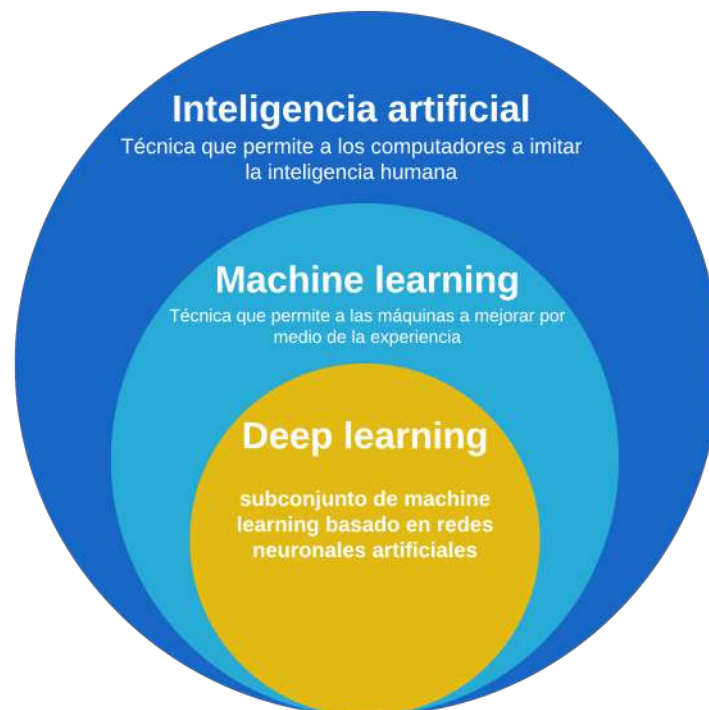


Fig. 2.12: Diagrama de los conceptos de Inteligencia Artificial, Machine Learning y Deep Learning.

2.7.1 Redes Neuronales Artificiales (Artificial Neural Networks, ANN)

La unidad base de las ANN es el perceptrón, también conocido coloquialmente como neurona artificial, el cual fue desarrollado en 1957 por Frank Rosenblatt [28]. Se observa la estructura de esta en la Fig. 2.13 (a).

La ecuación del perceptrón se define de la siguiente manera:

$$y = f \left(\sum_{i=1}^D x_i w_i + b \right) \quad (2.7.1)$$

Donde:

- D es la dimensionalidad de la información de entrada.
- x_i son las variables de entrada, que pueden ser, por ejemplo, los valores de los píxeles de una imagen.
- w_i es el peso asociado a un valor de entrada específico.
- b es el sesgo (bias) de la neurona.
- f es la función de activación, que es una función no lineal utilizada para permitir que el modelo resuelva problemas no lineales.

Cuando se unen varios perceptrones en paralelo y en serie en diferentes niveles de profundidad, donde la salida de un perceptrón se convierte en la entrada de otro perceptrón en una capa más profunda, se obtiene una red de perceptrones llamada Red Neuronal Artificial Fig.2.13(b).

La estructura de las ANN consta de una capa de entrada, donde inicialmente pasa la información, seguida de una o varias capas ocultas (hidden layers), donde la información se procesa para obtener las características más relevantes. Finalmente, se encuentra una capa de salida, donde la información se presenta en forma de resultado. Es importante destacar que también es posible tener una única capa que actúe como capa de entrada, hidden layer y salida [29][30].

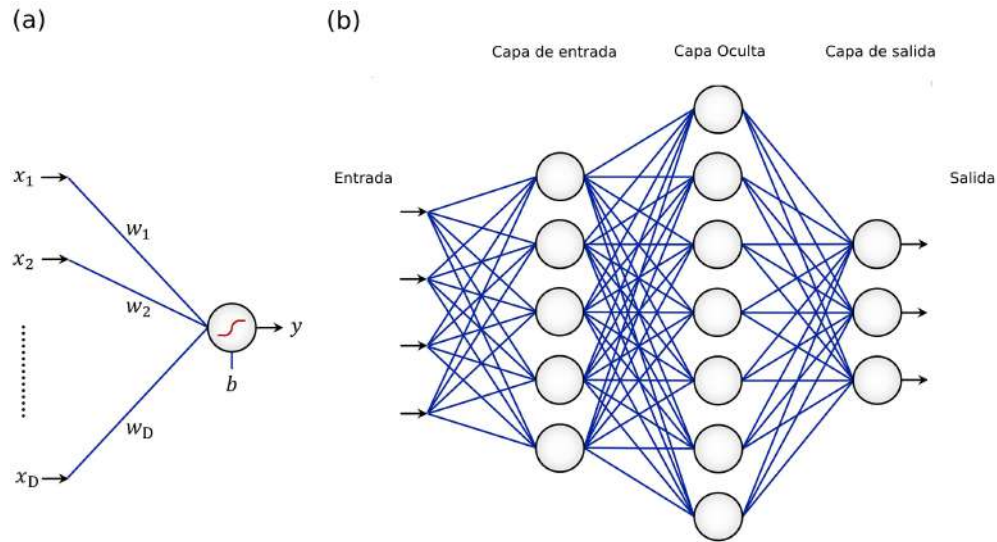


Fig. 2.13: (a) Estructura de un perceptron. (b) Red de perceptrones.

Imagen extraída de [30], se le añadió texto en español

2.7.2 Redes Neuronales Convolucionales (Convolutional Neural Networks, CNNs)

Las redes neuronales convolucionales, al igual que las ANN, son un tipo de algoritmo DL, pero en este caso, la información de entrada es una imagen que se convoluciona a través de filtros o kernels con el objetivo de extraer las características principales de la imagen. Estos filtros o kernels son las capas del modelo y los valores asociados a ellos son los pesos [31]. Las CNN tienen la ventaja de que los filtros y sus respectivos pesos se entrenan para detectar características sin importar su posición espacial en la imagen, por lo que no se requiere un procesamiento previo de los datos en comparación con las ANN [32].

Las CNN también presentan la arquitectura de capa de entrada, capas ocultas o hidden layers y capa de salida, pero este tipo de modelo cuenta con tipos de capas únicas necesarias para su correcto funcionamiento. Entre las más conocidas e importantes se encuentran la capa de convolución, la capa de agrupación o pooling layer, la capa densa o fully connected y la capa de función de activación. En este tipo de modelo usualmente se le denomina como hidden layer al conjunto de capas en serie compuesto por una capa de convolución, una capa de función de activación, otra capa de agrupación y una capa densa [33] [34]. En la siguiente Fig. 2.14 se muestra un ejemplo de la estructura básica de una CNN para identificar si una imagen corresponde a un perro o no, donde ReLU layer es un tipo de capa de función de activación.

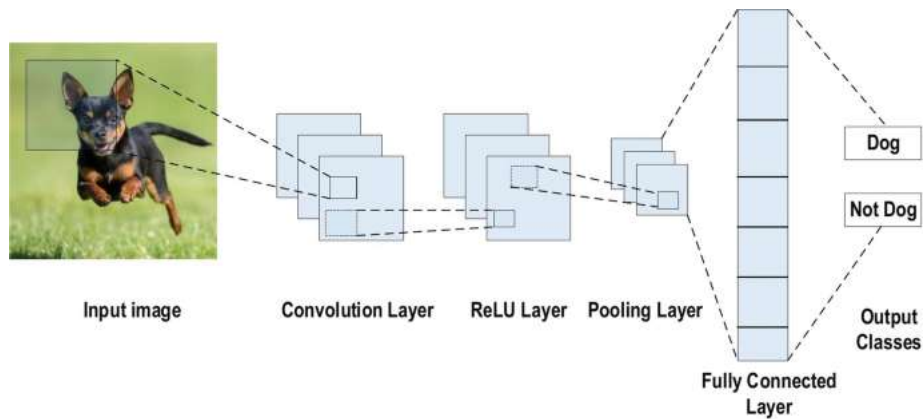


Fig. 2.14: Ejemplo de estructura básica de modelo de CNN.

Imagen extraída de [34]

2.7.2.1 Capa Convolutiva

La capa convolutiva es la responsable de extraer características de la imagen. Esto se logra mediante la convolución de los valores de los píxeles de la imagen de entrada con un kernel, que se puede definir como una malla de valores discretos. El kernel se desliza desde el inicio de la imagen hasta el último píxel, obteniendo un tensor de salida como se muestra en la Fig. 2.15. Un solo filtro produce un tensor de salida. En una capa convolutiva, normalmente se aplican varios filtros para obtener un mapa de características [34][35].

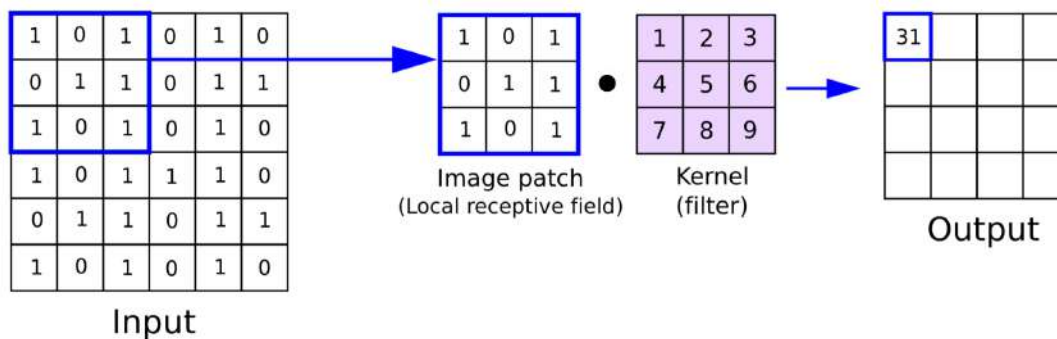


Fig. 2.15: Ejemplo de un filtro 3x3 a una imagen de 6x6 píxeles.

Imagen extraída de [34]

Como se observa en la Fig. 2.15, la aplicación de la convolución resulta en un tensor de tamaño menor que la imagen original. Las capas convolucionales presentan una serie de parámetros que permiten modificar el funcionamiento del kernel y la estructura de la entrada o salida, con el objetivo de darle la forma deseada al tensor de salida y/o modificar el resultado obtenido.

Entre los parámetros más importantes se encuentran el tamaño del kernel, el stride (que indica el tamaño del salto que realiza el kernel al deslizarse por la imagen de entrada), el padding (que es el relleno aplicado en los bordes de la imagen de entrada o salida, generalmente con valores iguales a 0) y la dilatación (que controla el espaciamiento entre los valores del kernel). Estos parámetros no son únicos a las capas convolucionales y también se utilizan para la capa de agrupación[34].

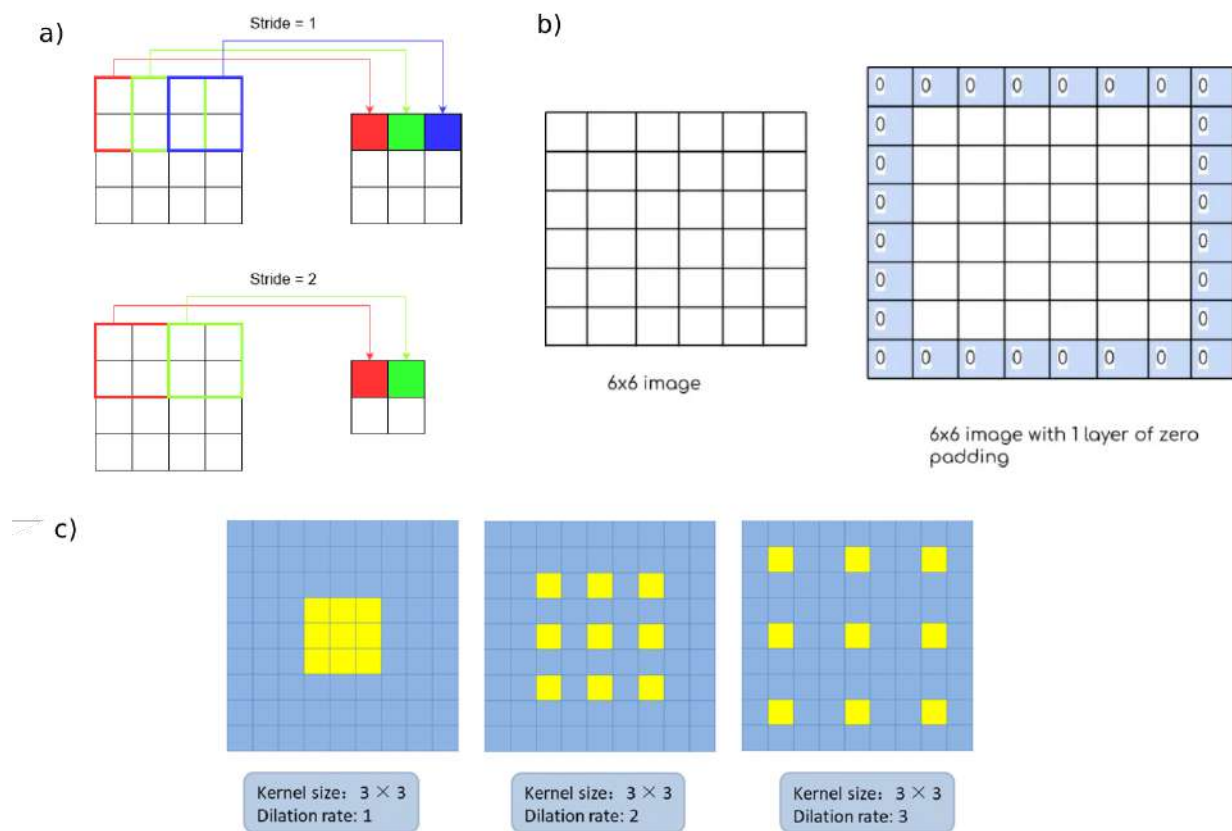


Fig. 2.16: Explicación visual de los parámetros de las capas de convolución. a) Stride. b) Padding. C) Dilatation.

Entender cómo funcionan estos parámetros y cómo impactan el tamaño de la imagen de salida es de suma importancia en aplicaciones donde se quiere reducir o aumentar la dimensionalidad de la información a un tamaño en específico. Para una imagen de dimensiones $H \times W$, donde H es la altura y W es el ancho, se utiliza la siguiente fórmula para determinar las dimensiones que tendrá la imagen después de pasar por la capa convolucional.

$$H_{out} = \left\lfloor \frac{H_{in} + 2P - D \cdot (K - 1) - 1}{S} + 1 \right\rfloor \quad (2.7.2)$$

$$W_{out} = \left\lfloor \frac{W_{in} + 2P - D \cdot (K - 1) - 1}{S} + 1 \right\rfloor \quad (2.7.3)$$

Donde:

- W_{in} y H_{in} son el ancho y la altura de la imagen antes de la capa convolucional.
- W_{out} y H_{out} son las dimensiones de salida de la imagen después de la capa convolucional.
- K es el tamaño del filtro o kernel utilizado en la convolución.
- P es el relleno (padding).
- D es el factor de dilatación.
- S es el paso (stride).

2.7.2.2 Capa de Agrupación (Pooling Layer)

Las capas de agrupación se utilizan para disminuir la dimensionalidad espacial de los mapas de características, con el propósito de reducir la cantidad de parámetros que el modelo debe aprender. Además, permiten introducir invarianza espacial, es decir, que el modelo sea menos sensible a variaciones en la posición de las características.

Estas capas funcionan seleccionando o calculando un valor a partir de una porción específica de la imagen de entrada. La técnica de agrupación más utilizada es la agrupación por valor máximo, donde se selecciona la celda con el valor más alto. En la Fig. 2.17 se muestra un ejemplo con agrupación por valor máximo y promedio. Sin embargo, existen muchas otras técnicas, como la agrupación por promedio, la agrupación estocástica, la agrupación espacial piramidal, entre otras [34][35].

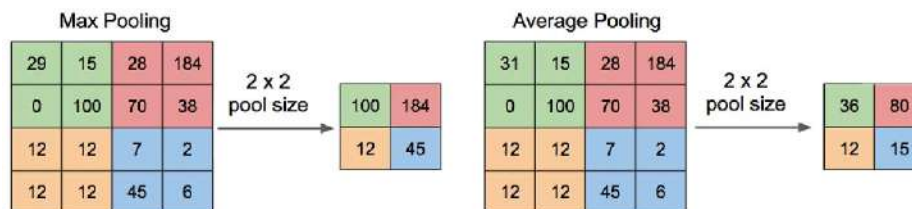


Fig. 2.17: Aplicación de capa de agrupación con técnica del valor máximo y promedio.

2.7.2.3 Capa Densa (Fully connected layer)

La capa densa o fully connected layer funciona de manera similar a los modelos de Redes Neuronales Artificiales (Artificial Neural Networks, ANNs). Se utiliza una vez que el modelo ha extraído las características y reducido su dimensionalidad a través de las capas de convolución y la capa de agrupación, respectivamente. Sin embargo, antes de pasar por la capa densa, la información debe pasar por la capa de aplanamiento, la cual, como su nombre indica, transforma la información en un vector 1D. Esto se debe a que los ANNs no toman como entrada un tensor, sino una variable, como los píxeles de una imagen [32][34].

Las capas densas se utilizan porque los ANNs permiten identificar características distintas a las obtenidas mediante las capas convolucionales. Además, permiten mapear la información a una capa densa de salida, donde los nodos finales se pueden utilizar para la clasificación. Por ejemplo, en la Fig. 2.14, se muestra una capa de salida compuesta por una capa densa de 2 nodos, la cual permite identificar si la imagen es de un perro o no [35][34].

2.7.2.4 Capa de Función de activación

Las funciones de activación son funciones no lineales que permiten que los modelos generen resultados no lineales. Estas funciones normalmente se ubican después de una capa de convolución, pero también es necesario colocar una capa de función de activación específica, dependiendo del problema que se quiere resolver, antes de la salida del modelo [34]. Por ejemplo, en un clasificador binario es necesario emplear la función de activación sigmoidea, la cual convierte las variables de salida en valores entre 0 y 1. La función de activación más utilizada en las capas ocultas es la función ReLU debido a su eficiencia y desempeño general. Esta función iguala todos los valores negativos a 0 y mantiene los valores positivos, como se puede observar en la Figura 2.18.

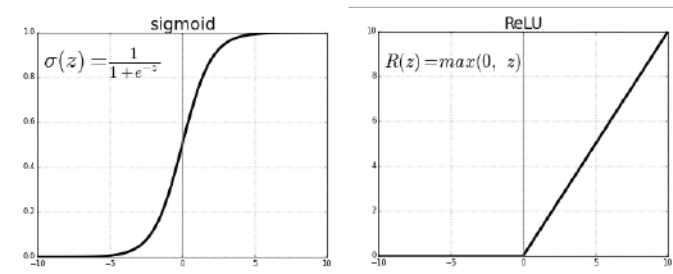


Fig. 2.18: Gráfica y ecuación de las funciones de activación Sigmoidea y ReLU.

2.8 Hiperparámetros

Los hiperparámetros en DL son configuraciones que pueden ser ajustadas para optimizar y mejorar el desempeño del algoritmo. A continuación se describen algunos de los hiperparámetros más relevantes.

- La tasa de aprendizaje o learning rate, determina la velocidad en que el modelo ajusta su peso durante la etapa de entrenamiento. Un learning rate alto puede llevar a una convergencia más rápida, pero también puede generar una divergencia del modelo. En cambio, utilizar un learning rate muy bajo conlleva a una convergencia lenta.
- La función de pérdida, también conocida como función de costo, calcula el valor del error del modelo que consiste en la diferencia entre la salida del algoritmo y la salida deseada o Verdad Absoluta (Ground Truth, GT). Existen distintas funciones de pérdida de las cuales la más usada es la función Error medio cuadrático (Mean Squar Error, MSE).
- Weight Decay o L_2 Regularization es una técnica de regularización, que tiene como propósito añadir un termino adicional a la función de pérdida con el objetivo de penalizar valores altos de los pesos. El uso de este parámetro es fundamental para evitar sobre entrenamiento o overfitting del modelo.
- La cantidad de épocas indica el número de veces que la base de datos es pasado por el modelo, y el tamaño del lote o batch size establece el número de datos que el algoritmo procesa antes de la actualización de los pesos.

Poder encontrar los valores óptimos para los distintos hiperparámetros del modelo puede ser una tarea complicada si se hace de forma manual (probando cada alternativa uno por uno). Para ello se utilizan métodos de ajuste de hiperparámetros, los cuales entrenan y evalúan el modelo con distintas combinaciones de hiperparámetros para poder determinar el más eficiente [36][37].

2.9 Autoencoder

Los Autoencoder (AE) son un tipo de Deep Learning que permite reducir la dimensionalidad de la información (o los datos) a un espacio denominado espacio latente, a través del entrenamiento de las distintas capas de sus redes neuronales (pueden ser convolucionales o artificiales), para la posterior reconstrucción de la información a partir de espacio latente [38]. La estructura general de un AE típico está formada por el Encoder por el cual pasa la información original y reduce las características de los datos a un espacio latente y el Decoder el cual reconstruye la información del espacio latente Fig. 2.19.

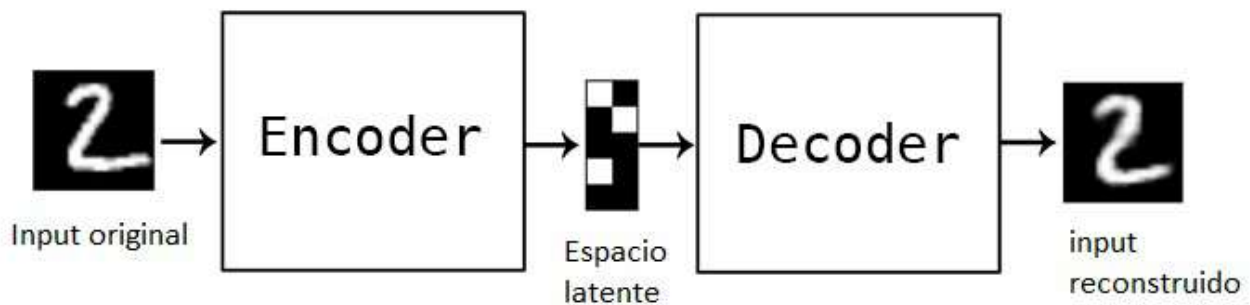


Fig. 2.19: Estructura general de un Autoencoder.

Dependiendo de la forma en que se entrene y la estructura del Autoencoder, este podrá tener aplicaciones distintas. Las aplicaciones más comunes en las cuales se utilizan los AE son para la reducción de dimensionalidad de los datos, reducción o eliminación de ruido, extracción de características y simulación o generación de nuevos datos.

2.10 Trabajos Previos

2.10.1 Simulación de Fibras

Los datos de tractografías no presentan un etiquetado o Verdad Absoluta (Ground Truth, GT) con respecto a qué fibras corresponden a qué fascículos, por lo que es complejo determinar la efectividad de los algoritmos de clustering de fibras cerebrales. Es por esto que algunos investigadores han propuesto la simulación de fibras cerebrales para generar tractografías de cerebro completo con sus datos etiquetados.

De la revisión bibliográfica, se puede observar la existencia de dos algoritmos que logran simular fibras cerebrales para evaluar algoritmos de clustering. El algoritmo de Poo et al. [2] permite obtener simulaciones de cerebro completo a partir de unas pocas fibras meticulosamente seleccionadas. El proceso de selección se muestra en la Fig. 2.20, donde primero se aplica el algoritmo de clustering FFClust para reducir la dimensionalidad de la información, y se filtran las fibras con una longitud menor a 50 milímetros. Posteriormente, se selecciona aleatoriamente una cierta cantidad de centroides de los grupos obtenidos mediante FFClust. Por último, se selecciona el conjunto de centroides que presenten la mayor distancia entre ellos.

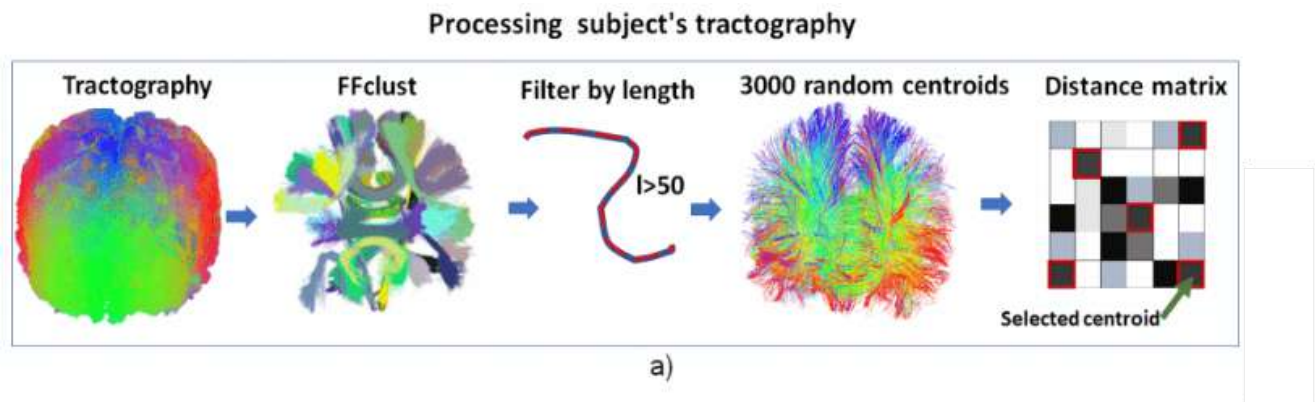


Fig. 2.20: Procesos para la selección de fibras a simular del algoritmo de Poo et al. [2].

Para simular un fascículo, el algoritmo selecciona un centroide y genera copias ligeramente desplazadas de este, creando así un fascículo con forma cilíndrica. Se le añaden curvas exponenciales al fascículo cilíndrico, como se muestra en la Fig. 2.21, con el objetivo de obtener simulaciones más realistas.

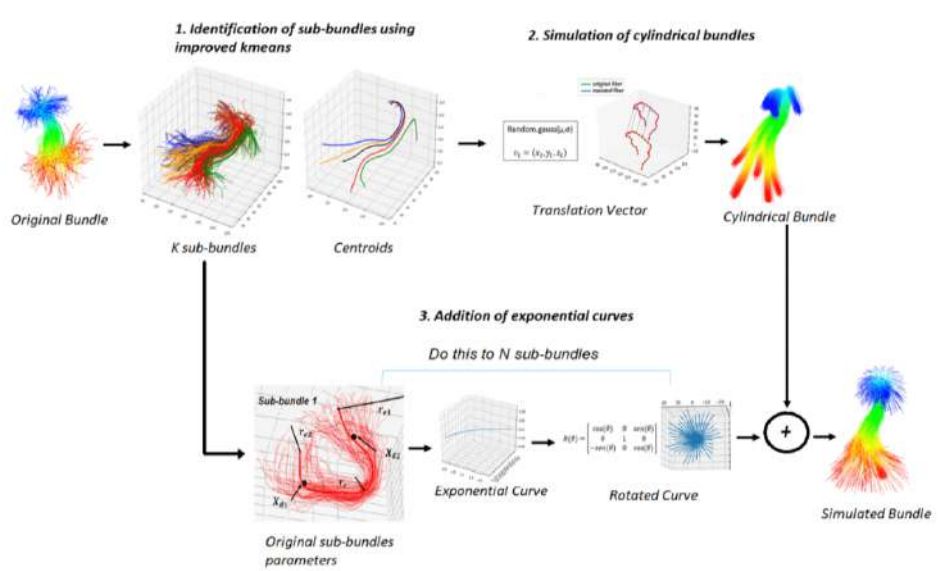


Fig. 2.21: Esquema generar para la simulación de fascículos del algoritmo de Poo et al. [2].

El otro algoritmo de simulación de fibras es el propuesto por Huerta et al. [3], el cual utiliza un Autoencoder LSTM con 7 valores en el espacio latente Fig. 2.22. La simulación se realiza llevando la información de una o varias fibras seleccionadas (generalmente los centroides de los fascículos) al espacio latente utilizando el encoder del modelo. Luego, se generan copias de estas fibras y se les añade ruido gaussiano a los valores en el espacio latente. Al pasar por el decoder, el modelo genera fibras ligeramente modificadas basadas en la información del espacio latente.

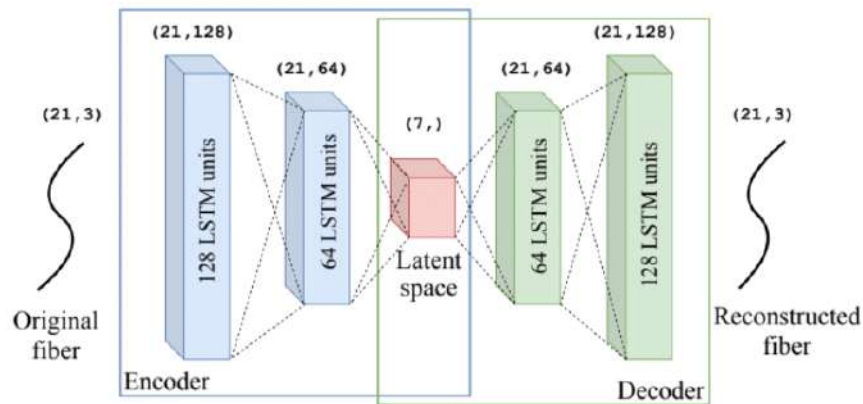


Fig. 2.22: Estructura del Autoencoder LSTM [3].

Este método puede simular fibras cerebrales. Sin embargo, tiene limitaciones para recrear la forma más compleja que presentan las fibras en sus extremos. Esto se debe a que solo utiliza el ruido gaussiano como parámetro, el cual tiene restricciones en términos de variabilidad y complejidad de los fascículos simulados.

2.10.2 Autoencoder

En el área de la neurociencia, específicamente en el procesamiento de datos de tractografía, se ha comenzado recientemente a implementar los Autoencoder con el objetivo de mejorar algunos procesos. Si bien existen varios artículos que han utilizado los Autoencoder, destacan los trabajos del grupo de investigación de la Universidad de Sherbrooke, quienes han desarrollado hasta la fecha los algoritmos FINTA [4] y GESTA [8], los cuales utilizan la misma estructura de Autoencoder. Estos algoritmos permiten filtrar las fibras cerebrales no plausibles de los conjuntos de tractografías (FINTA) y aumentar la cantidad de fibras cerebrales en las tractografías (GESTA).

El algoritmo FINTA [4] permite identificar las fibras cerebrales no-plausibles de los datos de tractografía, procesando la información en el espacio latente del Autoencoder con estructura de redes neuronales convolucionales Fig. 2.23. El Autoencoder consta de 6 capas convolucionales en el encoder, seguidas de capas de función de activación ReLU, lo que permite reducir la dimensionalidad de los datos y obtener resultados no lineales. Luego, se utiliza una capa de aplanamiento y una capa densa para llevar la información al espacio latente, que consta de 32 valores. Finalmente, el decoder realiza el procedimiento inverso al del encoder, utilizando capas de upsampling para aumentar la dimensionalidad de los datos y reconstruirlos en su formato original.

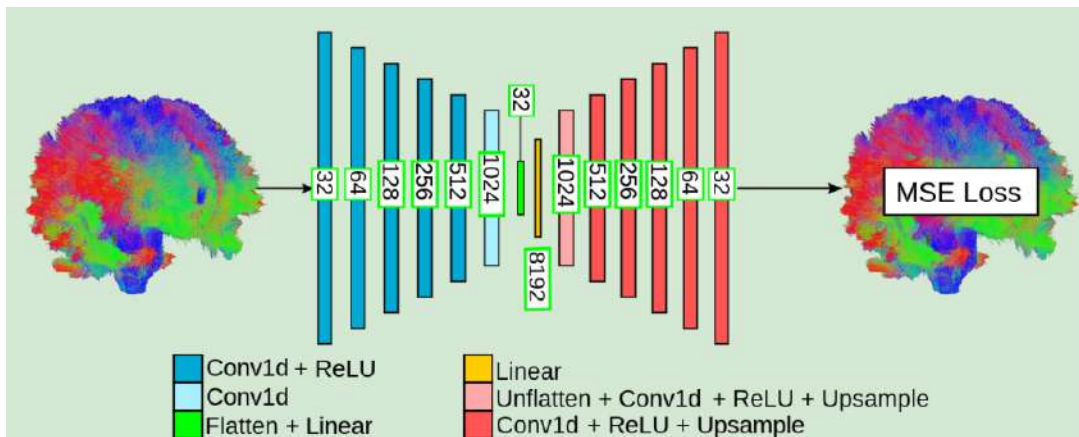


Fig. 2.23: Estructura del Autoencoder de FINTA.

Imagen extraída de [39]

Para identificar las fibras cerebrales implausibles, el grupo de investigación entrenó el Autoencoder utilizando fibras cerebrales de varios conjuntos de datos de tractografía. Una vez entrenado el Autoencoder, llevaron la información de los conjuntos de tractografía al espacio latente, donde observaron que, en promedio, las fibras implausibles presentaban una distancia mayor a su vecino más cercano. Por lo tanto, calcularon un umbral de distancia para descartar estas fibras del conjunto de tractografías como muestra la Fig. 2.24

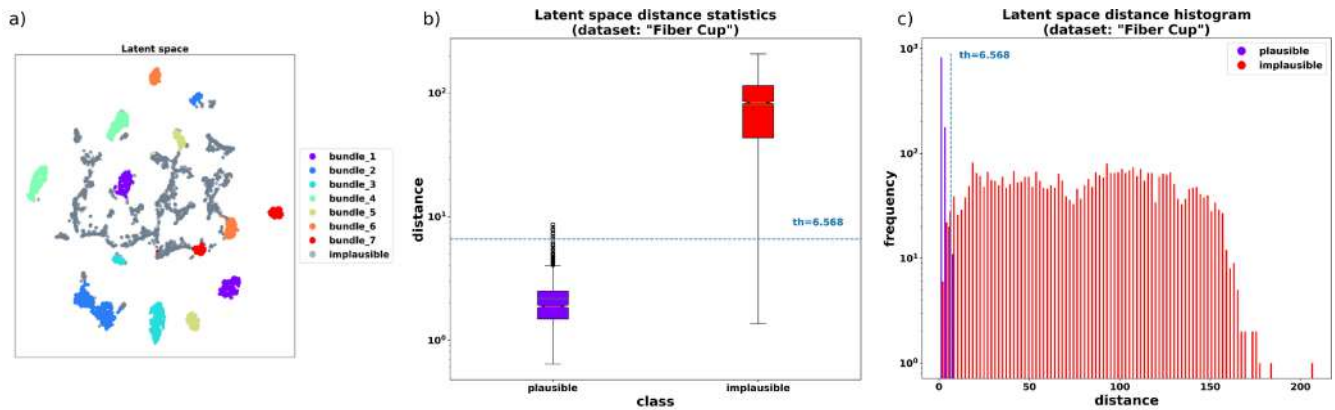


Fig. 2.24: Identificación de fibras cerebrales no-plausibles de algoritmo FINTA [4]. a) Representación t-SNE del espacio latente de una porción de la base de datos "Fiber Cup". b) Distancia al vecino más cercano para las fibras plausibles e implausibles visualizado mediante gráficos de caja. c) Distancia al vecino más cercano para las fibras plausibles e implausibles visualizado mediante histograma.

Imagen adaptada de [4], se juntó y reorientó de varias figuras

2.11 Discusión

La capacidad de las computadoras para procesar información y poder identificar características y patrones a partir de pequeñas variaciones en la información original ha creado su propia rama de la ciencia. La cual cada año genera modelos más y más complejos con el objetivo de automatizar y/o resolver problemas en todos los ámbitos posibles, desde los problemas de la vida cotidiana hasta el entendimiento del funcionamiento del cerebro, el cual irónicamente ha inspirado los fundamentos de los algoritmos de aprendizaje profundo.

Del estudio bibliográfico se observa que se han mejorado considerablemente las herramientas para poder entender la anatomía, funcionalidad y conectividad del cerebro, pero aún queda mucho por mejorar y comprender en todos los ámbitos del estudio cerebral. Desde los métodos de difusión, los equipos de resonancia magnética y sus sensores, los algoritmos de tractografía, la histología de las propias neuronas, la variabilidad del cerebro entre las personas, entre muchos otros. Los problemas que busca abordar esta memoria de título, como la simulación de fibras y la segmentación de fascículos por medio del espacio latente, son puntos muy específicos entre todos los problemas que existen, pero aun así son clave para poder entender mejor este complejo sistema llamado cerebro.

Capítulo 3. Base de Datos y Códigos Existentes

3.1 Introducción

En este capítulo se describen las bases de datos que se utilizaron en este trabajo, así como los códigos existentes, los cuales fueron proporcionados por el grupo de análisis de imágenes médicas de la Universidad de Concepción.

3.2 Base De Datos

3.2.1 Datos ARCHI

Esta cuenta con datos de dMRI de 79 sujetos sanos, que corresponden a datos HARDI de la base de datos ARCHI. El protocolo de adquisición incluye imágenes anatómicas T1 y una adquisición HARDI SS-EPI single-shell con una resolución espacial isotrópica de $1,7 \times 1,7 \times 1,7 \text{ mm}^3$, 60 direcciones de difusión optimizadas con $b = 1500 \text{ s/mm}^2$. Se realizó un preprocesamiento de los datos utilizando el software BrainVISA/Connectomist-2.0. Se aplicó el modelo analítico de Q-ball para obtener campos de ODF en cada voxel. Se calculó la tractografía determinística de streamlines para la máscara de todo el cerebro basada en la imagen T1 (sin umbral de FA), con un paso de 0,1 mm y un ángulo máximo de curvatura de 30° [9].

Además de los conjuntos de tractografía, la base de datos dispone transformadas afines de cada uno de los sujetos para la normalización T1-Talairach y T2-T1.

3.2.2 Datos de adolescentes con y sin autismo

Base de datos HARDI que consta de 37 adolescentes, de los cuales 18 son controles y 19 son pacientes diagnosticados con autismo [40]. Esta base de datos fue facilitada por la profesora Pamela Guevara y proporcionada por colaboradores del Centro Neurospin en Francia. La base de datos contiene una amplia cantidad de información, pero en este trabajo específico, se utilizarán únicamente las transformadas afines para llevar la información a un espacio normalizado y los datos de tractografía para evaluar el algoritmo de segmentación propuesto.

3.2.3 Atlas de Fascículos de Materia Blanca Profunda

Atlas de fascículos de la Materia Blanca Profunda (Deep White Matter, DWM), muestra la variabilidad de posición y forma entre sujetos. El atlas está compuesto por 36 fascículos y un total 22,158 fibras cerebrales etiquetadas Fig. 3.25 [26].

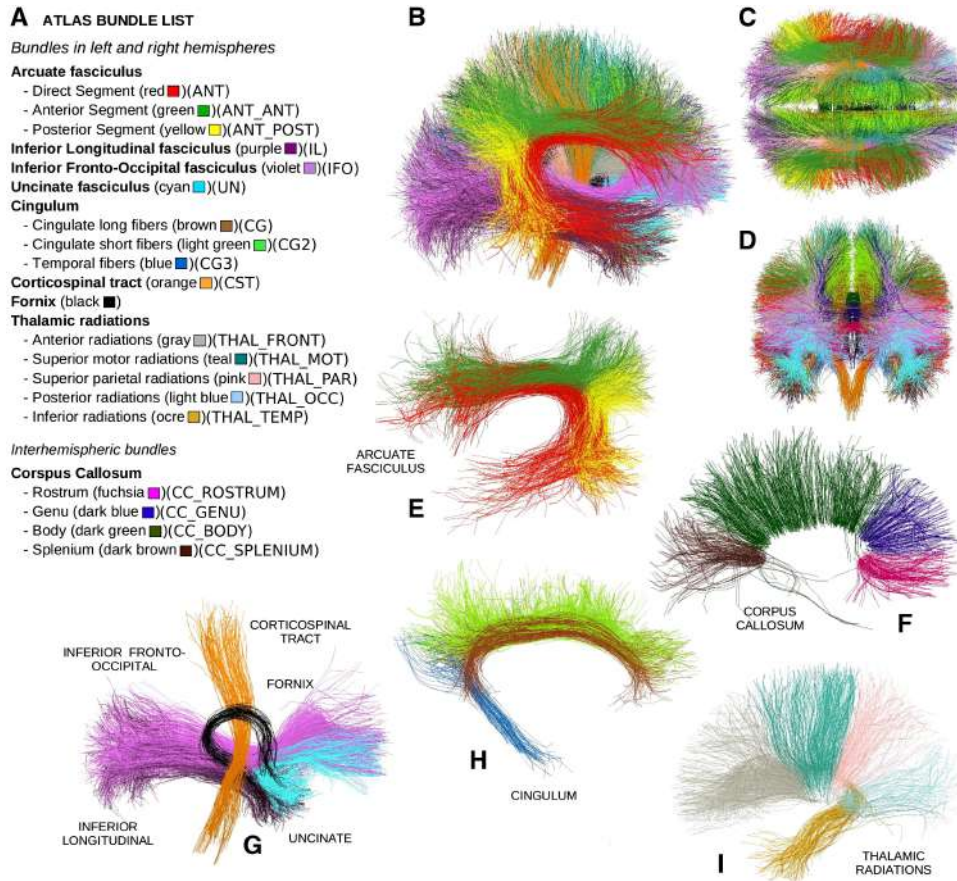


Fig. 3.25: Atlas de fascículos de materia blanca profunda. A) Lista de los fascículos con sus respectivos colores y abreviaciones. B) Plano sagital del atlas. C) Plano axial del atlas. D) Plano coronal del atlas. E) Fascículos arqueados. F) Plano sagital de los fascículos del cuerpo calloso. G) Plano sagital de los fascículos fronto-occipital inferior, longitudinal inferior, fórnix, corticoespinal y uncinado. H) Plano sagital de los fascículos cingulados. I) Plano sagital de los fascículos talámicos.

Imagen extraída de [26]

3.3 Códigos Existentes

Para llegar a cabo algunos procesos se utilizaron códigos existentes que fueron proporcionados por el grupo de análisis de imágenes médicas de la Universidad de Concepción.

- bundlesTools: Código escrito en lenguaje Python. Presenta distintas funciones que permite la lectura y escritura de archivos bundles.
- Transformación espacial de tractografías: Código escrito en lenguaje Python. Presenta 3 funciones que trabajan en conjunto para poder transformar el espacio de coordenadas de un dato de tractografía a otro por medio de una matriz afín.
- Segmentación de fascículo: Código escrito en lenguaje C. Permite segmentar las fibras de un sujeto al calcular la distancia Euclidiana máxima entre cada fibra con N-puntos y todos los centroides del respectivo atlas que se usó como referencia [1].

3.4 Discusión

Se seleccionó una base de datos de tractografías de sujetos con alguna patología (en este caso autismo) para respaldar los resultados de la segmentación propuesta debido a que estos sujetos pueden presentar una variabilidad en su materia blanca profunda, lo que dificulta la segmentación de fibras.

Si bien las bases de datos de tractografías no están en el mismo espacio de coordenadas, debido a que fueron adquiridos con métodos y por grupos de adquisición distintos, ambas tienen las respectivas transformadas afines, las cuales permiten llevarla la información a espacio normalizado y poder comparar los resultados correctamente.

Capítulo 4. Diseño del Autoencoder y Entrenamiento del Modelo

4.1 Introducción

En este capítulo se explican los procesamientos que se deben realizar en la base de datos ARCHI para lograr un entrenamiento óptimo del Autoencoder. Se describe la estructura que tendrá el modelo y las bibliotecas que se utilizaron, así como la optimización de los hiperparámetros. Además, se presentan los resultados obtenidos durante el entrenamiento del Autoencoder.

4.2 Metodología

En la siguiente Figura 4.26 se muestra los distintos procesos que se llevan a cabo en el diseño y entrenamiento del Autoencoder.

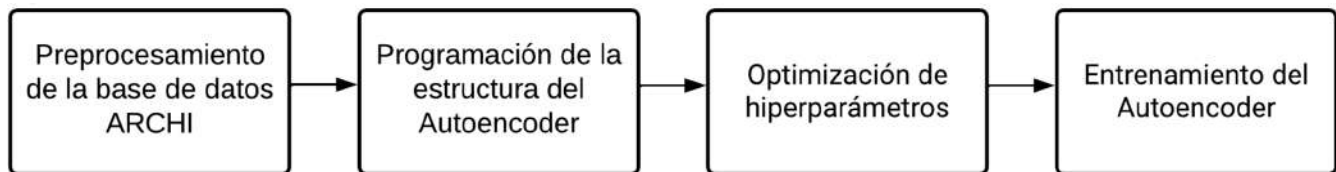


Fig. 4.26: Diagrama de procesos del diseño y entrenamiento del Autoencoder

4.2.1 Procesamiento de Bases de Datos

Para el correcto funcionamiento del modelo de aprendizaje profundo y para mejorar su rendimiento, es necesario aplicar tres preprocesamientos a la base de datos.

En primer lugar, debido a que los datos de tractografía de cada sujeto de la base de datos están segmentados en tres archivos .bundles, es necesario unificarlos en un mismo archivo. Además, se debe remuestrear cada fibra a 256 puntos equidistantes utilizando la biblioteca Dipy [41], como se muestra en la Figura 4.27.

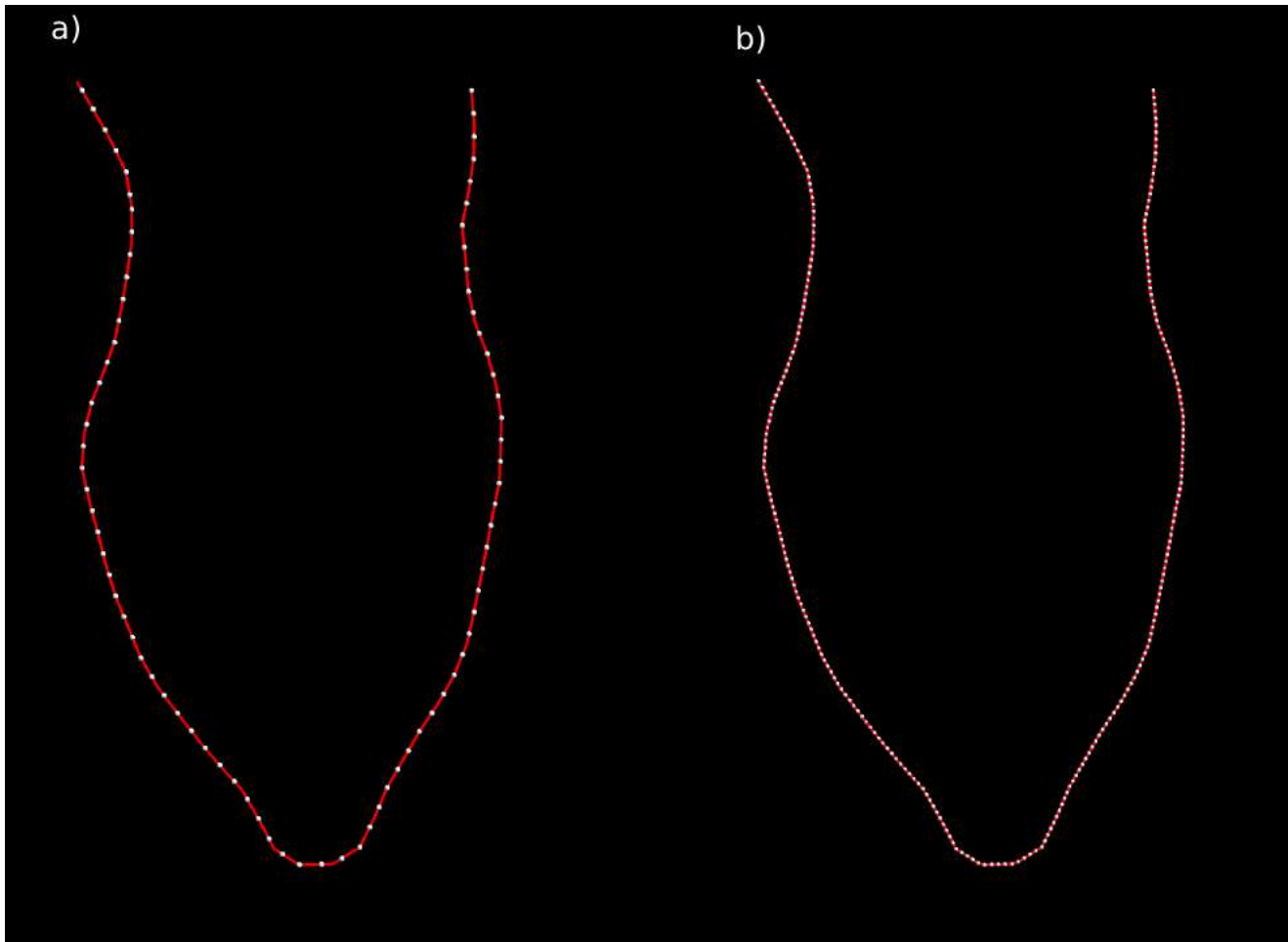


Fig. 4.27: Remuestro de fibras cerebrales. Fibras cerebrales en rojo y puntos por fibra en blanco. a) Fibra cerebral original con 78 muestras. b) Fibra cerebral remuestrada a 256 puntos equidistantes.

En segundo lugar, es importante que todas las fibras tengan la misma orientación. Esto implica que el primer punto de la fibra debe estar más cercano al punto de referencia $(0,0,0)$ en términos de distancia Euclidiana que el último punto de la fibra. En caso de que el último punto de la fibra esté más cercano al punto de referencia que el primero, se debe cambiar la orientación de la fibra.

Por último, se deben eliminar las fibras cerebrales que tengan una longitud menor a 15 mm, ya que afectan al rendimiento del modelo debido a que suelen ser fibras ruidosas.

4.2.2 Programación de la Estructura del Autoencoder

Para la programación del Autoencoder se usa Python en conjunto con la biblioteca PyTorch [42], la cual contiene las herramientas necesarias para crear un algoritmo de Deep Learning.

Para el Autoencoder se emplea la estructura del paper FINTA [4], con la diferencia de que se ha modificado el espacio latente para tener una mayor dimensionalidad, con 128 valores en lugar de 32, con el fin de tener una mayor cantidad de características por fibra para mejorar el procesamiento de esta información. La estructura modificada de FINTA se observa en la Fig. 4.28 la cual utiliza capas convolucionales unidimensionales en el encoder para extraer características y reducir la dimensionalidad mediante la selección de parámetros específicos como el tamaño de kernel, stride, dilatación y padding, seguido de una capa de función de activación ReLU. Para llevar la información al espacio latente, se utiliza una capa de aplanamiento seguida de una capa densa que genera el vector de valores necesario, en este caso, 128. En el decoder, se realiza el proceso inverso al encoder, empleando capas de Upsample que aumentan la dimensionalidad de la imagen para devolver la información al espacio original.

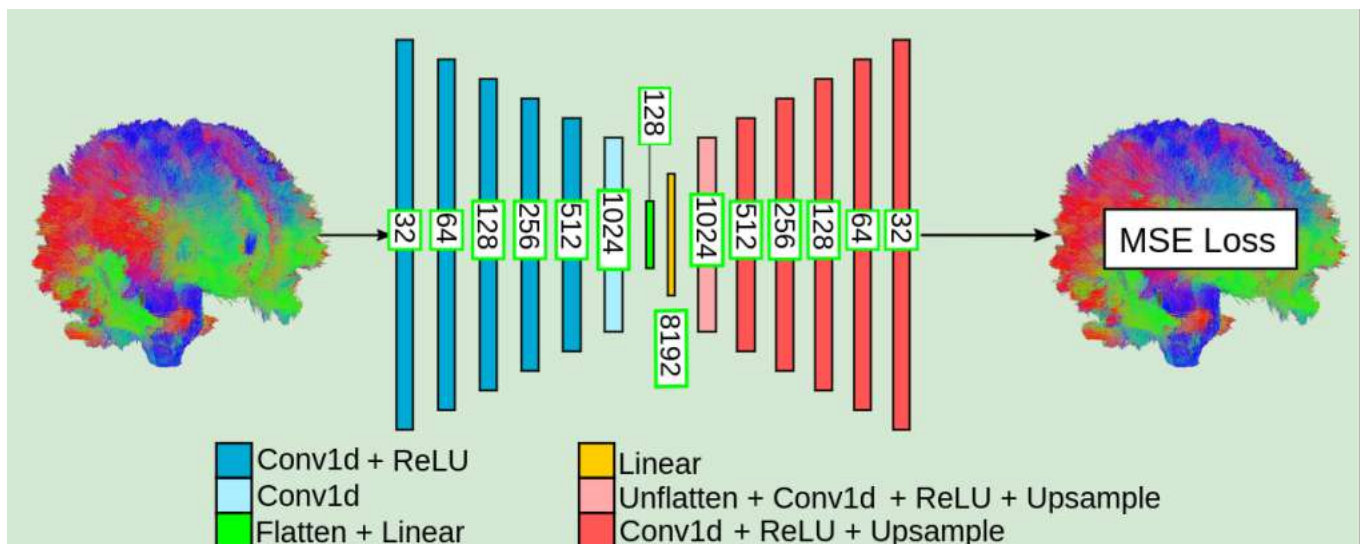


Fig. 4.28: Estructura del Autoencoder de FINTA modificada a 128 valores en el espacio, latente

Imagen adaptada de [39], se modificó el valor del espacio latente

La tabla 4.1 muestra de mejor forma cómo el tamaño y cantidad de canales de la información va cambiando al pasar por las distintas capas del Autoencoder.

Tabla 4.1: Estructura del Autoencoder. Para todas las capas de convolución se usa un tamaño de kernel de 3, padding de 1 y dilatation de 1.

Seccion	Tipo de capa	Canales	Tamaño
Entrada	–	3	256
Encoder	1D conv (Stride = 1)	32	256
	1D conv (Stride = 2)	64	128
	1D conv (Stride = 2)	128	64
	1D conv (Stride = 2)	256	32
	1D conv (Stride = 2)	512	16
	1D conv (Stride = 2)	1024	8
	Aplanamiento	8192	–
Espacio Latente	Capa densa	128	–
Decoder	Capa densa	8192	–
	desaplanamiento	1024	8
	upsampling + 1D conv (Stride = 1)	1024	8
	upsampling + 1D conv (Stride = 1)	512	16
	upsampling + 1D conv (Stride = 1)	256	32
	upsampling + 1D conv (Stride = 1)	128	64
	upsampling + 1D conv (Stride = 1)	64	128
	upsampling + 1D conv (Stride = 1)	32	256
Salida	–	3	256

4.2.3 Optimización de Hiperparámetros

La optimización de los hiperparámetros y la correcta elección de sus valores pueden marcar la diferencia entre un modelo que falle o que tarde demasiado tiempo en alcanzar el objetivo y uno que cumpla con los requisitos de forma rápida y eficiente. Para la optimización de los hiperparámetros, se utilizó la biblioteca Optuna [43], que permite el ajuste automático de los hiperparámetros con métodos de vanguardia y trabaja en conjunto con la mayoría de bibliotecas de ML, incluyendo PyTorch [42].

Para optimizar los hiperparámetros de ‘Weight decay’ y ‘Learning Rate’ de nuestro modelo, se seleccionaron aleatoriamente 350.000 fibras de los datos tractografía del sujeto 005 de la base de datos ARCHI y se usó Optuna para ajustar la combinación de hiperparámetros de tal manera que se minimice el valor de pérdida del modelo al finalizar el entrenamiento durante 100 épocas (se empleó el optimizador Adam). Optuna se programó para buscar 12 combinaciones distintas de los parámetros utilizando el modelo de búsqueda por defecto TPESampler. El muestreador Tree-structured Parzen Estimator (TPE) funciona mediante un proceso iterativo en el cual utiliza el historial de resultado de las distintas combinaciones de hiperparámetros para sugerir una nueva combinación de hiperparámetros por medio de modelos probabilísticos. De la búsqueda de hiperparámetros, se obtuvo los resultados mostrados en la tabla 4.2, donde la segunda combinación presentó el mejor desempeño.

Tabla 4.2: Combinaciones de hiperparámetros seleccionados por Optuna y su desempeño.

Combinación #	Learning Rate	Weight Decay	Perdida
1	5.7691×10^{-4}	1.2142×10^{-5}	0.31911
2	2.0413×10^{-5}	1.5157×10^{-5}	0.20529
3	4.7646×10^{-4}	33129×10^{-4}	0.37576
4	5.4565×10^{-5}	25392×10^{-4}	341.98
5	2.9901×10^{-3}	3.0562×10^{-5}	10.224
6	3.5084×10^{-3}	1.2003×10^{-5}	9744.9
7	2.6625×10^{-3}	3.7843×10^{-5}	2.1098
8	2.8617×10^{-3}	1.2627×10^{-5}	2.8580
9	2.1553×10^{-5}	4.9199×10^{-5}	8.2516
10	4.8101×10^{-5}	1.0791×10^{-5}	1197.1
11	7.0578×10^{-5}	1.5922×10^{-5}	0.24475
12	1.5757×10^{-5}	1.0516×10^{-5}	0.21067

4.2.4 Entrenamiento del Autoencoder

Para poder entrenar el Autoencoder, primero es necesario cargar los datos de tractografías en Python y modificar su formato para que pueda ser utilizado por la biblioteca PyTorch [42]. Al cargar los datos de tractografías en Python utilizando la biblioteca bundleTools, la información se encuentra en un formato de lista de listas, donde cada lista contiene las coordenadas de las fibras en los 3 ejes. Sin embargo, PyTorch no permite la lectura directa de este formato como base de datos para el entrenamiento del Autoencoder, por lo que es necesario realizar algunas modificaciones.

En primer lugar, se convierte la lista de listas en una matriz tridimensional (i, j, k), donde ‘i’ representa el número de fibras por tractografía, ‘j’ representa el número de puntos por fibra y ‘k’ representa el número de ejes de coordenadas (equivalente a 3). Posteriormente, se debe intercambiar el orden de los ejes ‘j’ y ‘k’, obteniendo así una matriz en el formato (i, k, j). Este intercambio es necesario para que los datos estén en el formato adecuado para el procesamiento con PyTorch. Finalmente, se convierte la estructura de la matriz en un tensor, que es el formato de datos requerido por PyTorch para su correcto funcionamiento.

El modelo fue entrenado usando un ordenador con sistema operativo Windows 10 Pro, procesador 12th Gen Intel(R) Core(TM) i5-12400F 2.50 GHz, 16 GB de RAM y tarjeta de vídeo NVIDIA GeForce RTX 3060 Ti. El entrenamiento del Autoencoder se realizó utilizando 5 conjuntos de tractografías preprocesadas, con una duración de 100 épocas para cada conjunto. En total, el entrenamiento tomó aproximadamente 325 minutos.

En la gráfica de pérdida del modelo (Fig. 4.29) se puede observar que el Autoencoder llega a valores de pérdida bajos de forma rápida debido a la optimización de parámetros por medio de Optuna. Además, se puede observar un aumento de la pérdida en las épocas en las que se cambió de conjunto de tractografía debido a la variabilidad de fibras entre sujetos. El modelo llegó en su última época a un valor de pérdida de 0.0392408448.

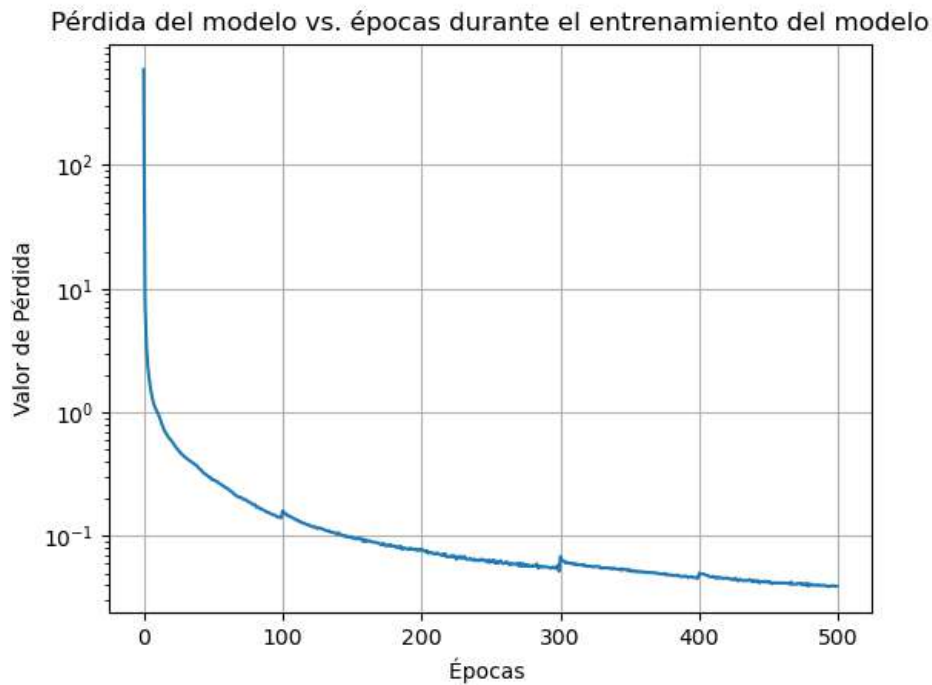


Fig. 4.29: Gráfica de la pérdida del modelo vs. las épocas

En la Figura 4.30 se muestran 25 fibras en color verde indicando las fibras originales y en rojo las mismas fibras, pero pasadas por el Autoencoder ya entrada, es decir, el modelo comprimió la información al espacio latente y posteriormente la reconstruyó. Se puede observar que las figuras fueron reconstruidas casi perfectamente, por lo que el modelo está apto para el procesamiento de datos a partir de este.

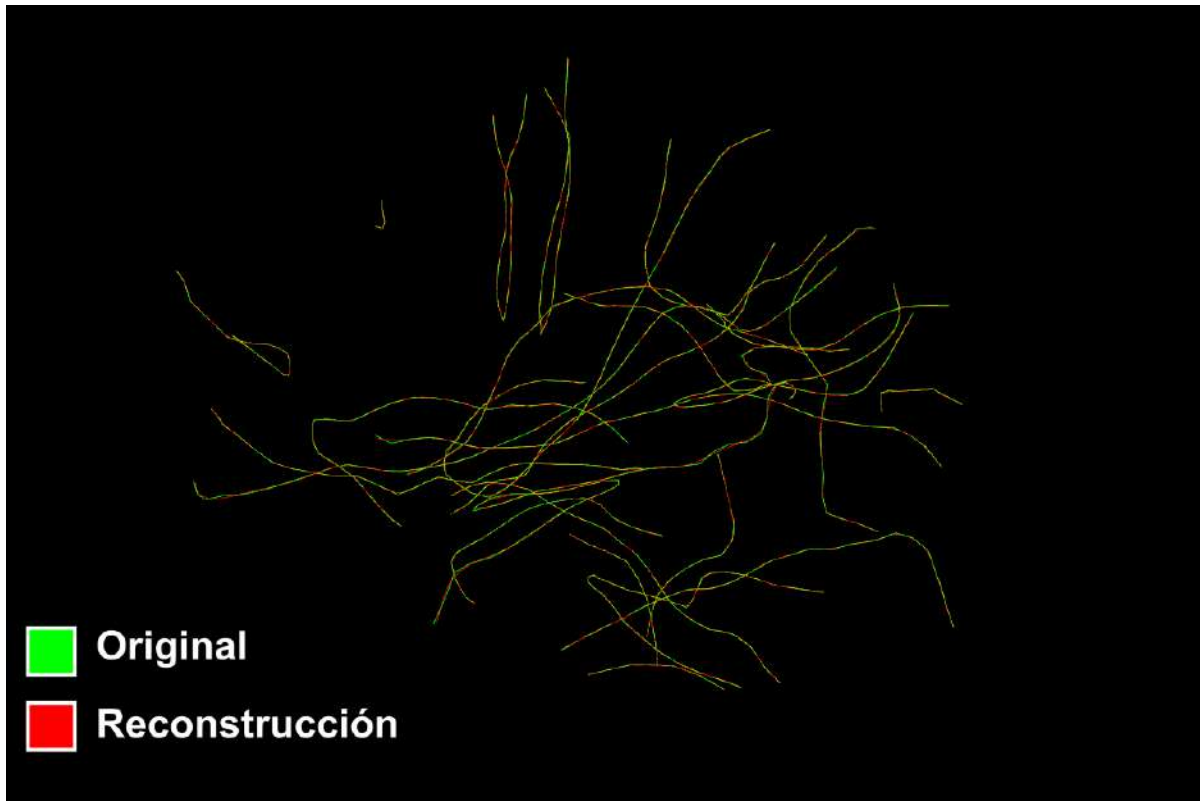


Fig. 4.30: Comparación entre fibras orinales y la reconstrucción de las mismas por el Autoencoder.

4.3 Discusión

El procesamiento de la base de datos, específicamente la eliminación de las fibras con longitud menor a 15 milímetros y la optimización de hiperparámetros mediante Optuna, resultaron ser pasos esenciales para lograr un entrenamiento óptimo y rápido del Autoencoder. Inicialmente, se intentó entrenar el modelo con la totalidad de las fibras y utilizando los hiperparámetros utilizados en FINTA [4]. Sin embargo, este enfoque resultó en tiempos de entrenamiento prolongados y los resultados no cumplieron con las expectativas.

La estructura del Autoencoder, aunque se basó en gran medida de FINTA [4], requirió varios meses para poder ser implementada en un código funcional. Inicialmente, se estudiaron los fundamentos de los algoritmos de aprendizaje profundo, en particular los Autoencoder y redes neuronales convolucionales. Luego, se realizó la tarea de convertir el modelo en un código funcional en lenguaje Python, lo cual implicó aprender a utilizar las bibliotecas PyTorch y Optuna, las cuales presentan una curva de aprendizaje significativa.

Capítulo 5. Simulación de Fibras

5.1 Introducción

Con el Autoencoder ya entrenado se procede a simular fibras cerebrales por medio del espacio latente del Autoencoder. Para la simulación se utilizaron algoritmos de clustering con el objetivo de agrupar las fibras cerebrales en fascículos y sub-fascículos. Con la finalidad de poder generar fascículos con curvatura variada, se interpolan las fibras con respecto al centroide de cada sub-fascículo. La simulación se logra interpolando algunos valores en el espacio latente y añadiéndole un ruido gaussiano para posteriormente pasarlo por el decoder del modelo.

5.2 Metodología

Para poder simular fibras cerebrales por medio del Autoencoder entrenado, es necesario pasar por una serie de procesos que se ven plasmados en la Fig. 5.31.

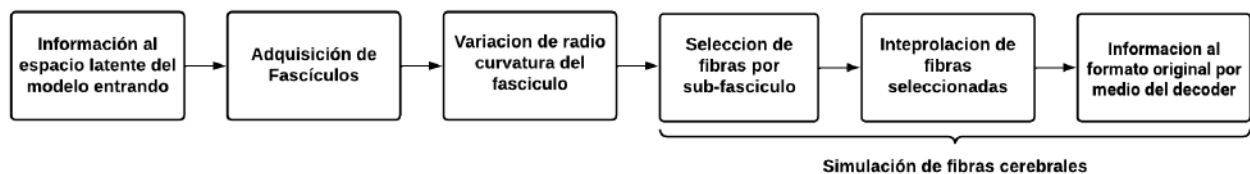


Fig. 5.31: Diagrama de procesos para la simulación de fibras.

5.3 Selección de Fascículos

La Selección de fascículos se logró procesando la información de las fibras cerebrales en el espacio latente usando el algoritmo de clustering DBSCAN de la biblioteca scikit-learn [44]. Las fibras originales se transforman al espacio latente por medio del encoder del Autoencoder entrenado. Este procesamiento toma las fibras con formato (256,3) y las convierte en un vector unidimensional de 128 valores.

El algoritmo DBSCAN tiene varios parámetros que se pueden optimizar para mejorar los resultados. Dos de los parámetros más importantes son *epsilon* y *minimum samples*. *epsilon* se refiere a la máxima distancia entre dos puntos para que sean considerados vecinos y sean agrupados juntos. Por otro lado, *minimum samples* indica la cantidad mínima de puntos necesarios para formar un grupo.

Para el ajuste de estos parámetros se utilizó el método propuesto por [45]. Este método sugiere obtener el valor de *epsilon* utilizando el método del codo, el cual consiste en ordenar de forma ascendente la distancia de cada dato a su vecino más cercano y posteriormente seleccionar el valor donde se produce un cambio significativo en la pendiente de la curva. En cuanto al *minimum samples*, se recomienda utilizar un valor mayor a la dimensionalidad de la información. Esto se debe a que se espera que los grupos sean lo suficientemente densos y representativos de la información.

Se aplicó el método mencionado a los valores del espacio latente del sujeto 079. De acuerdo con la Fig. 5.32, se determinó que el valor de *epsilon* óptimo a utilizar es de 54 (redondeado). Dado que la dimensionalidad del espacio latente es de 128, se decidió utilizar un valor de 130 para el parámetro *minimum samples*.

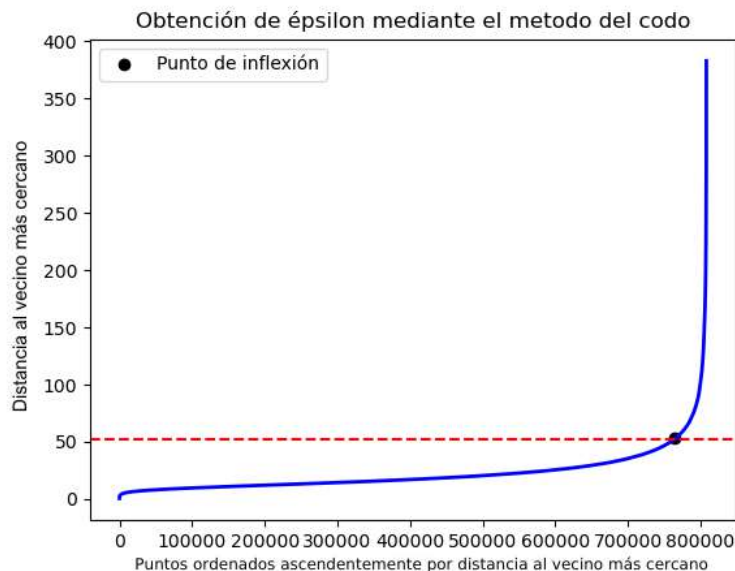


Fig. 5.32: Método del codo para determinar el valor óptimo de *epsilon*

Al implementar el algoritmo DBSCAN con los parámetros ajustados a la base de datos, se observa que la información es agrupada en un total de 391 grupos distintos, los cuales están etiquetados del -1 al 390. Es importante mencionar que las fibras con etiqueta -1 corresponden a fibras ruidosas que no pertenecen a ningún grupo específico.

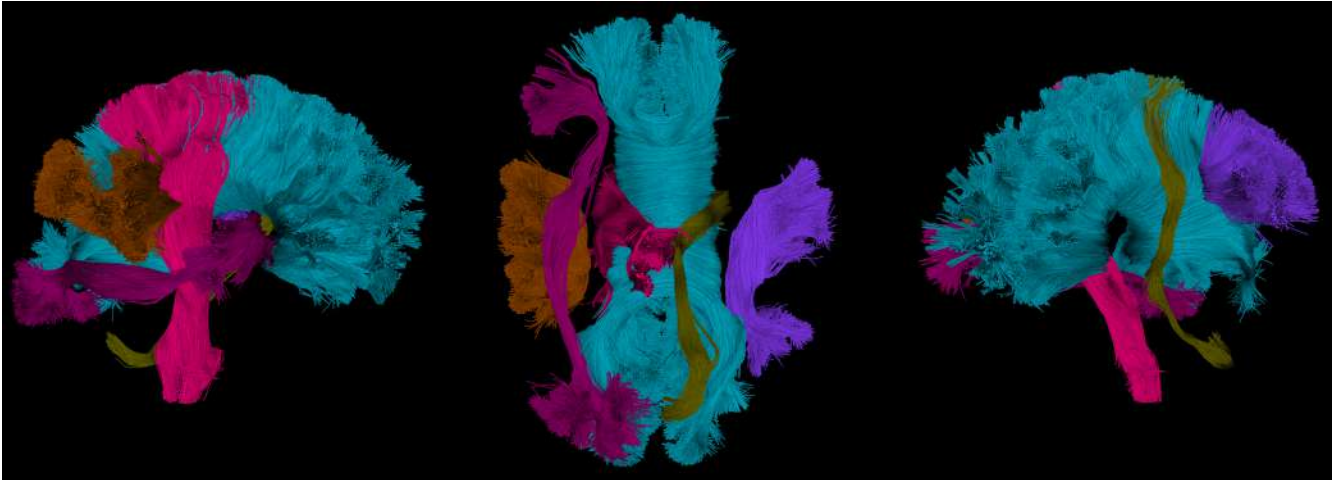


Fig. 5.34: Distintas perspectivas de algunos fascículos obtenidos por DBSCAN en el espacio latente.

5.4 Variación de Radio de Curvatura del Fascículo

Con los fascículos de los datos de tractografía ya seleccionados, se aplica el algoritmo de K-means de la biblioteca scikit-learn [44] sobre el espacio latente de cada fascículo para agruparlos en subfascículos y obtener sus centroides, lo que permitirá variar el radio de curvatura. La cantidad de centroides obtenidos depende del parámetro $n_clusters$ de K-means. En la Fig. 5.35 se muestra la obtención de centroides del fascículo corticoespinal izquierdo para $n_clusters$ igual a 4.

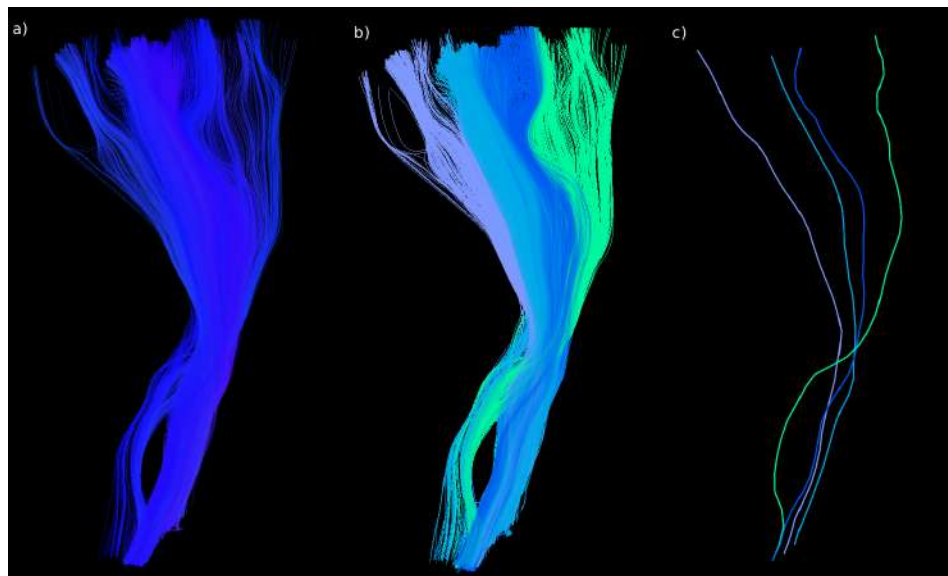


Fig. 5.35: Ilustración de la obtención de centroides del fascículo Corticoespinal izquierdo. a) Fascículo Corticoespinal izquierdo. b) Subfascículos coloreados obtenidos por medio de K-means. c) Centroides de cada grupo.

La variación del radio de curvatura de los fascículos se logra mediante la interpolación de cada fibra con su respectivo centroide. La interpolación es un método que se utiliza para estimar valores desconocidos a partir de valores conocidos.

Es importante destacar que existe una diferencia significativa entre la interpolación en el espacio latente y la interpolación en la información original. El modelo ha sido entrenado para reconstruir la información en el espacio latente y tiene conocimiento sobre la forma de las fibras. Por lo tanto, la generación de datos en el espacio latente permite crear fibras con formas realistas y coherentes.

El grado de curvatura de los fascículos está dado según el factor de interpolación f_i que se le aplica a la diferencia entre el centroide y la fibra, como se muestra en la ecuación 5.4.1. Si se aplica un f_i menor a 1, se obtendrán fascículos con radio menor al original, mientras que si se utilizan f_i mayores a 1, el radio será mayor.

$$y = y_c + (y_c - y_i) \cdot f_i \quad (5.4.1)$$

Donde:

- y es el valor en el espacio latente de la fibra interpolada.
- y_c es el valor en el espacio latente del centroide.
- y_i es el valor en el espacio latente de la fibra a interpolar.
- f_i es el factor de interpolación.

La figura 5.36 muestra el fascículo corticoespinal con distintos radios de curvatura, obtenidos al interpolar las fibras de un grupo con su respectivo centroide para distintos valores de f_i .

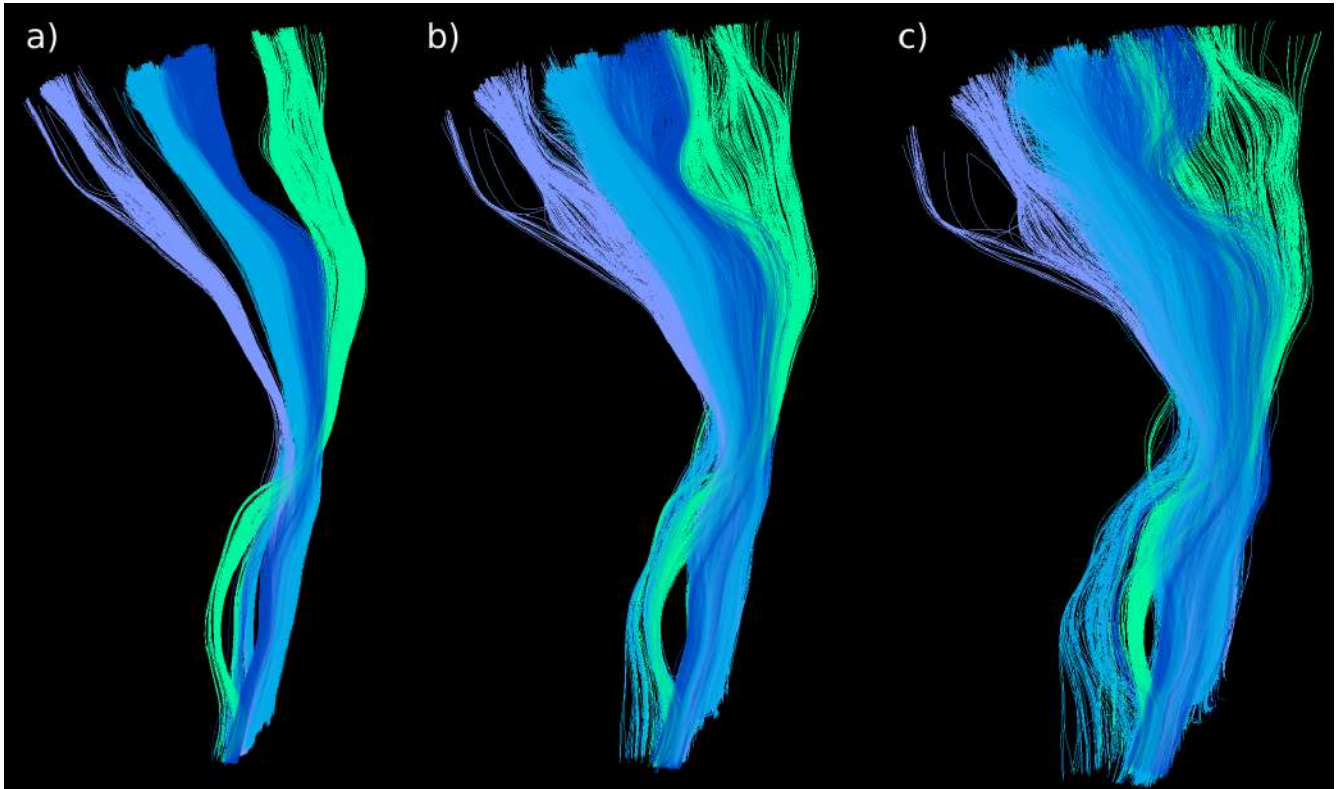


Fig. 5.36: Fascículo Corticoespinal con radio de curvatura variado. a) Interpolación con $f_i=0.5$. b) Interpolación con $f_i=1$. c) Interpolación con $f_i=1.5$. Los colores representan los distintos subfascículos

5.5 Simulación de Fibras Cerebrales

Una vez que se tiene el fascículo con el radio de curvatura variado, se seleccionan k fibras por sub-fascículo y se interpolan entre sí con un f_i variado desde 0.2 hasta 1.0, hasta completar la cantidad de fibras que tenga el sub-fascículo. Dependiendo de la cantidad de fibras que se seleccionen, la simulación de fibras será más parecida al fascículo original. Debido a que el objetivo es generar fascículos con la misma forma, pero no igual al original, se optó por utilizar 15 fibras por fascículo.

Por último, se le añade un ruido gaussiano a los valores del espacio latente de las fibras seleccionadas y a las nuevas fibras obtenidas de la interpolación, con el objetivo de modificar levemente las fibras para mejorar la simulación. Posteriormente, los valores del espacio latente se pasan por el decoder del Autoencoder entrenado para volver a su formato de información original. El proceso desde la selección del fascículo hasta pasar por el decoder se muestra en la Fig. 5.37.

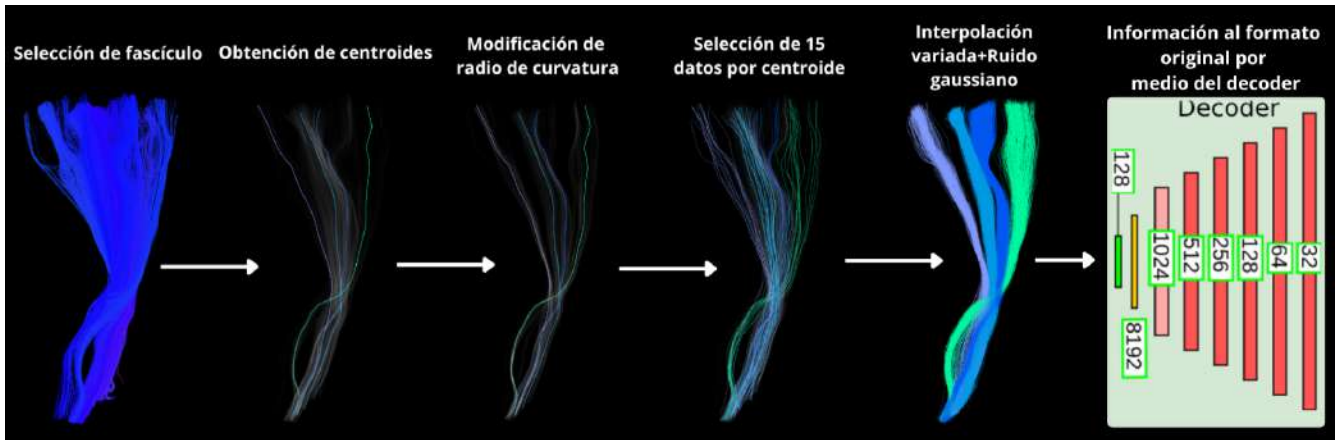


Fig. 5.37: Diagrama de procesos desde la selección de fascículo hasta el uso del decoder. Todos los procesos. Se rastrearon los índices de las fibras al espacio original para obtener las figuras.

5.6 Discusión

Inicialmente, se consideró simular los fascículos a partir de centroides mediante la generación de copias y la adición de ruido gaussiano. Sin embargo, este enfoque resultó en fascículos con una forma demasiado tubular, lo cual no era realista para algunos de ellos. Por lo tanto, se optó por utilizar la selección e interpolación de fibras para la simulación, lo que permitió luego variar el radio de curvatura de los fascículos, añadiendo así otro nivel de modificación de los fascículos.

En la implementación, se buscaba tener varios parámetros para poder modificar ampliamente el resultado de la simulación del fascículo. Sin embargo, una complicación que surgió al tener diferentes parámetros, como el ruido gaussiano, el grado de interpolación, la cantidad de centroides y la cantidad de fibras seleccionadas para interpolar, fue la dificultad para lograr un equilibrio entre ellos. Ya que no se busca generar un fascículo idéntico o igual al original, sino más bien obtener otra representación del mismo. Por ejemplo si se seleccionan demasiadas fibras para interpolar, la simulación resultante se asemejaba demasiado al fascículo original. Por otro lado, si se utilizaban muy pocas fibras, el resultado sería pobre. Lo mismo sucedía con cada parámetro, siendo necesario encontrar un equilibrio adecuado entre ellos.

Capítulo 6. Segmentación de Fibras mediante clasificador

6.1 Introducción

El método propuesto en este trabajo segmenta los datos de tractografías en fascículos conocidos mediante el clasificador Radius Neighbors Classifier. Este se entrenó utilizando una base de datos que contiene información en el espacio latente de distintos conjuntos de tractografías segmentadas mediante el algoritmo [1], proporcionado por el grupo de análisis de imágenes médicas de la Universidad de Concepción. En este capítulo se describirá la implementación del segmentador de fibras y se explicarán los distintos procesos que se llevaron a cabo para crear la base de datos con la cual se entrenó el modelo.

6.2 Creación de Base de Datos para el Clasificador

Con el objetivo de entrenar el clasificador, se creó una base de datos con la información en el espacio latente de los datos de tractografías segmentadas de 5 sujetos, mediante el algoritmo de segmentación [1]. Para obtener la segmentación, es necesario que la información de los sujetos y la del atlas estén en un espacio común. Debido a que el atlas está en el espacio Talairach y los sujetos en su propio espacio, se les aplicó las respectivas matrices de traslación a cada sujeto para llevarlos al espacio Talairach. Con la información en este espacio común, se procedió a segmentar los datos de tractografía utilizando el algoritmo [1]. La Tabla A.1 muestra la cantidad de fibras que tiene cada fascículo para los 5 sujetos.

Debido a que el Autoencoder fue entrenado en el espacio propio de los sujetos, se rastrearon los índices de los fascículos segmentados en Talairach al espacio original. Por último, se llevó la información de los fascículos segmentados al espacio latente mediante el encoder del Autoencoder. La Fig. 6.38 muestra la representación t-SNE de la base de datos creada.

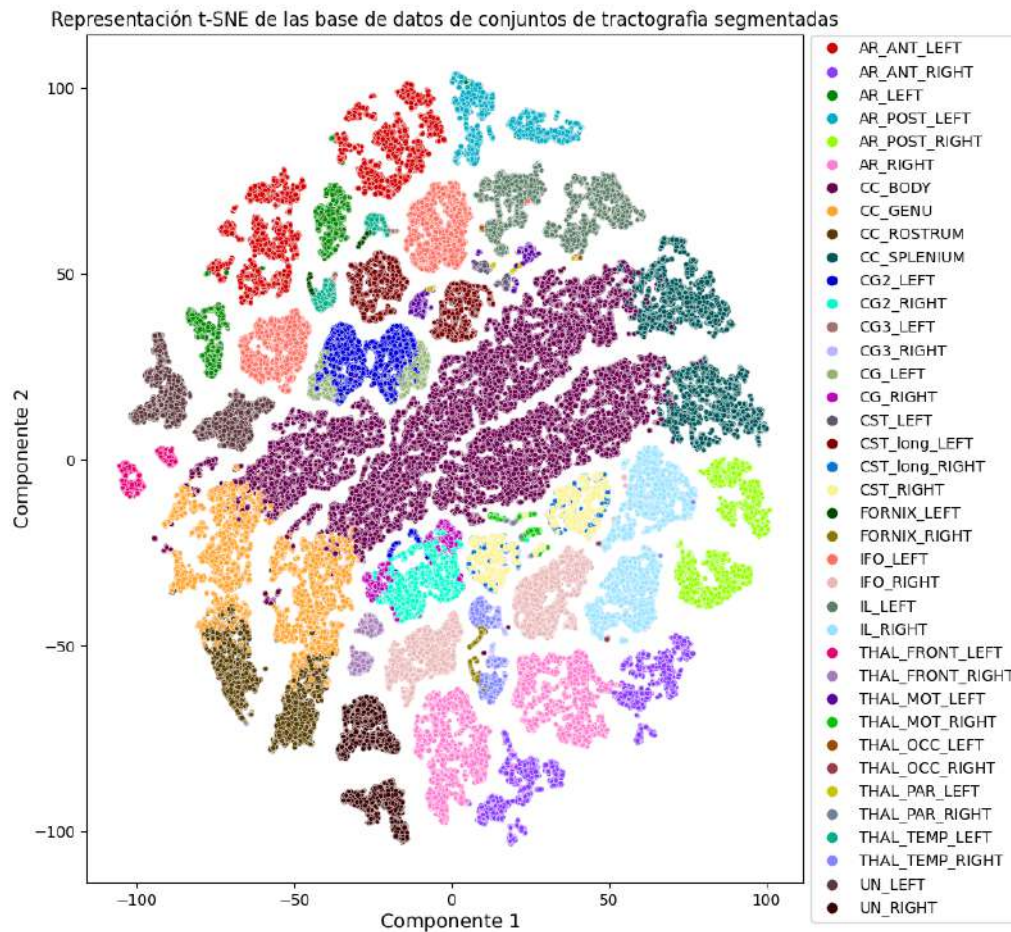


Fig. 6.38: Representación t-SNE de la base de datos de segmentaciones. Para la obtención de esta imagen se utilizó perplexity = 160, exaggeration = 1, radom state = 42.

6.3 Clasificador

El clasificador que se empleó fue el Radius Neighbors Classifier de la biblioteca scikit-learn [44]. Este clasificador permite predecir la etiqueta de la información nueva al compararla con la distancia radial de los datos de entrenamiento. Se utilizó este clasificador en lugar del K-vecinos más cercanos (K-Nearest Neighbors, K-NN) debido a que permite encontrar vecinos dentro de una distancia de búsqueda determinada. Además, si la información a clasificar no presenta ningún vecino dentro del radio de búsqueda, se le asigna la etiqueta -1, evitando así la clasificación errónea de fibras que no son parte de ningún fascículo

a segmentar.

Dado que algunos fascículos, como los distintos fascículos del cuerpo calloso, arqueados o cingulados, están ubicados cerca unos de otros y presentan una variación en la cantidad de fibras, se configuró el parámetro *weights* del clasificador con la función '*distance*'. Esto implica que los vecinos más cercanos tienen mayor peso en la clasificación que los vecinos más alejados. Al asignar un peso mayor a los vecinos más cercanos, se otorga una mayor influencia a la información proveniente de las regiones próximas, lo que ayuda a mejorar la precisión en la clasificación de estos fascículos. La Figura 6.39 muestra un ejemplo ilustrativo del funcionamiento de '*distance*'.

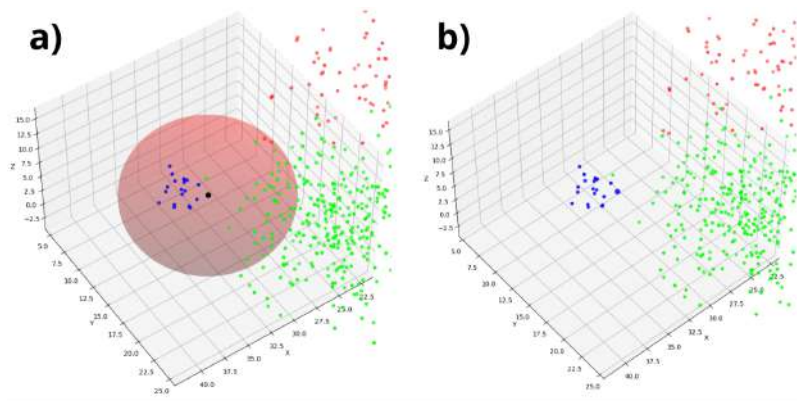


Fig. 6.39: Ejemplo ilustrativo de la clasificación de un nuevo dato usando '*distance*' como función del parámetro *weights*. a) Clasificación del nuevo dato en color negro. b) Nuevo dato clasificado con el color azul debido a que los datos más cercanos aportan un mayor peso a la clasificación.

El parámetro más importante a ajustar es el de la distancia de búsqueda o *radio*. Si se elige un valor muy pequeño, el método segmentará muy pocas fibras por fascículo. Por otro lado, si se le asigna un valor demasiado grande, segmentará fibras que no forman parte de los respectivos fascículos. La optimización de este parámetro se llevó a cabo comparando los resultados de la segmentación del sujeto 074, el cual no pertenece al conjunto de entrenamiento del clasificador, para los valores de radio 45, 60, 75, 80, 85, 90 y 120.

La tabla 6.3 muestra la cantidad de fibras obtenidas por fascículo para los distintos valores de la distancia de búsqueda. Además, en la Figura A.1 y Figura A.2 se visualiza la segmentación de los fascículos del hemisferio izquierdo y del cuerpo calloso para diferentes valores de la distancia de búsqueda. A partir de estas imágenes y la información de la tabla, se determinó que los valores 45 y 60 para la distancia de búsqueda no son óptimos debido a la escasa segmentación de fibras, especialmente para el fascículo arqueado directo derecho (AR_RIGHT).

Tabla 6.3: Cantidad de fibras segmentadas por fascículos

Fascículos	Distancia de búsqueda						
	45	60	75	80	85	90	120
AR_ANT_LEFT	11	1329	5546	7199	8596	9674	14982
AR_ANT_RIGHT	114	2391	10233	12595	14364	15527	20055
AR_LEFT	0	228	2260	3336	4371	5313	7519
AR_POST_LEFT	143	1546	4475	5565	6446	7268	13333
AR_POST_RIGHT	123	1473	5327	6684	8026	9499	17979
AR_RIGHT	0	3	176	332	588	961	3319
CC_BODY	3704	14928	27316	30253	32035	33152	36235
CC_GENU	421	4678	9457	10403	11054	11613	11950
CC_ROSTRUM	504	2791	4195	4556	4961	5445	8246
CC_SPLENIUM	6	1208	3703	4617	5411	6043	8283
CG2_LEFT	560	2584	5640	6810	8054	9096	14219
CG2_RIGHT	1035	3835	7165	7988	8564	9039	11183
CG3_LEFT	21	251	840	1048	1255	1471	2687
CG3_RIGHT	91	310	734	862	998	1149	2238
CG_LEFT	19	189	967	1165	1331	1473	1674
CG_RIGHT	72	534	1559	1810	2030	2141	2174
CST_LEFT	1430	3628	5570	5923	6128	6271	7825
CST_long_LEFT	42	97	58	142	217	237	516
CST_long_RIGHT	1309	5515	6749	6997	7214	7381	8914
CST_RIGHT	48	829	1975	2314	2685	2967	3994
FORNIX_LEFT	12	42	110	156	197	221	363
FORNIX_RIGHT	47	208	353	388	412	448	714
IFO_LEFT	0	98	1642	2985	4418	5776	10106
IFO_RIGHT	1	167	1047	1632	2433	3443	10676
IL_LEFT	18	1335	5460	7198	9102	11131	22711
IL_RIGHT	34	619	4129	6266	8898	11760	25612
THAL_FRONT_LEFT	83	1239	3468	4061	4563	5039	7175
THAL_FRONT_RIGHT	355	1527	2894	3429	4039	4761	8980
THAL_MOT_LEFT	695	1440	1681	1848	1981	2156	2858
THAL_MOT_RIGHT	789	1790	2397	2546	2671	2786	3302
THAL_OCC_LEFT	2	102	456	646	795	929	1201
THAL_OCC_RIGHT	0	20	68	100	161	213	598
THAL_PAR_LEFT	18	104	198	231	248	265	402
THAL_PAR_RIGHT	18	88	243	305	310	341	847
THAL_TEMP_LEFT	570	1269	2010	2310	2597	2909	4777
THAL_TEMP_RIGHT	711	1648	2434	2810	3273	3730	6278
UN_LEFT	108	599	1447	1817	2229	2647	5598
UN_RIGHT	452	2115	3544	3972	4372	4823	7978
TOTAL	13566	62757	137526	163299	187027	209098	317501

Para determinar cuál de los valores restantes es el mejor, se visualización las segmentaciones de los fascículos con el objetivo de identificar las fibras que anatómicamente no forman parte de ellos. Debido a que al aumentar el valor del radio, la cantidad de fibras por fascículo aumenta considerablemente, lo que dificulta el análisis. Se optó por observar únicamente las nuevas fibras obtenidas al aumentar el radio, como se muestra en la Figura 6.40.

Del análisis de estas imágenes, se observó que para los valores de radio 120 y 90, el modelo tiene dificultades para clasificar las fibras de los fascículos talámicos parietal y occipital, confundiendo la clasificación entre ambos. Además, en los fascículos arqueados y frontoocipitales inferiores se comienzan a segmentar fibras que se proyectan fuera de las regiones correspondientes a estos fascículos.

Debido a que los valores de radio 75, 80 y 85 muestran una buena segmentación, se optó por utilizar el valor de radio 85, ya que permite segmentar más fibras por fascículo, mejorando así la representación de los fascículos en la segmentación.

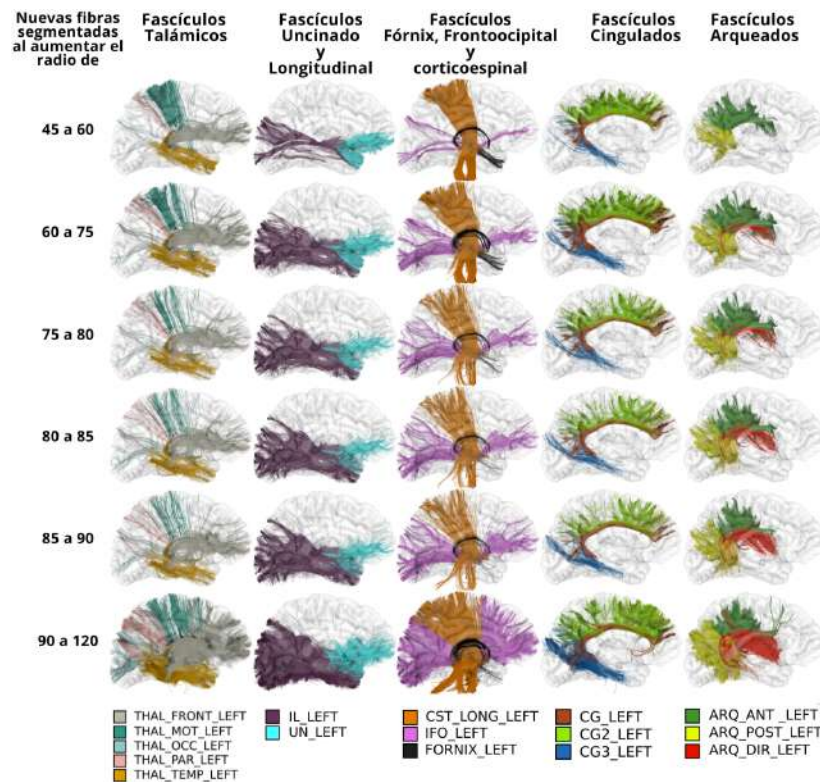


Fig. 6.40: Nuevas fibras que se obtienen para cada fascículo del hemisferio izquierdo al aumentar el radio de búsqueda del clasificador.

Es importante destacar que la definición de la forma que deben tener los distintos fascículos y la determinación de si las fibras efectivamente forman parte de ellos es un proceso difícil y altamente subjetivo, que depende de la experiencia y el conocimiento de la persona. Diferentes grupos de investigación, protocolos de adquisición de datos de tractografía y algoritmos pueden obtener segmentaciones distintas para el mismo fascículo de un mismo sujeto [47].

La Figura 6.41 muestra la segmentación de los fascículos arqueado, corticoespinal y del cuerpo calloso realizada por diferentes grupos de investigación para el mismo sujeto. En esta figura se puede observar que cada grupo obtiene una segmentación parcialmente distinta.

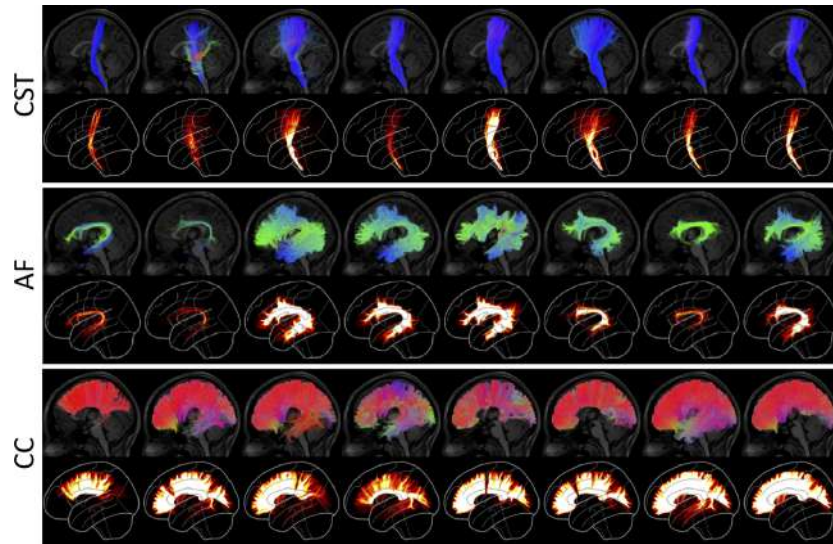


Fig. 6.41: Segmentación de los fascículos Corticioespinal (CST), fascículos arqueado (AF) y cuerpo calloso (CC) del mismo sujeto, obtenido de distintos grupos de investigación con protocolos y algoritmos distintos.

Imagen extraída de [47]

6.4 Discusión

El modelo de segmentación propuesto es capaz de clasificar correctamente los 38 fascículos con una buena cantidad de fibras, dependiendo del radio de búsqueda. Sin embargo, se ha observado que el proceso de clasificación puede ser más lento en comparación con otros métodos. Esto se debe principalmente a la gran cantidad de fibras utilizadas como entrenamiento para el clasificador, lo cual requiere una selección más meticulosa de estas fibras para abordar este problema.

Uno de los principales desafíos de los algoritmos de segmentación, incluyendo este método, es la selección del valor límite o umbral, que corresponde al valor del radio de búsqueda para el algoritmo propuesto. Para mejorar significativamente la segmentación, es necesario desarrollar un modelo automático de selección del valor límite que pueda ajustarse individualmente para cada sujeto a segmentar. Durante la implementación de la segmentación, se ha observado que este valor límite podría automatizarse mediante un cálculo basado en la densidad de las fibras y la distancia al vecino más cercano en el espacio latente. Sin embargo, se requiere más tiempo e investigación para lograr resultados concretos en este aspecto.

Capítulo 7. Resultados

7.1 Introducción

En este capítulo se presentan los resultados obtenidos al implementar el simulador de fibras cerebrales en 5 fascículos con formas y estructuras complejas. Se utilizaron las métricas de Bundle Adjacency y la distancia Euclidiana media a la fibra más cercana para obtener valores cuantitativos de los resultados, además de mostrar imágenes de las simulaciones. También se muestran los resultados del algoritmo de segmentación y se realiza una comparación con el algoritmo propuesto por López-López et al. [1] en 7 sujetos de la base de datos ARCHI y 4 sujetos de la base de datos de jóvenes con y sin autismo.

7.2 Simulación de Fibras Cerebrales

Debido a que los algoritmos de simulación de fibras cerebrales actuales presentan problemas para generar fascículos más complejos, se evaluará el modelo propuesto para los siguientes fascículos: arqueado anterior, corticoespinal, fronto-occipital inferior, longitudinal inferior y el fascículo Genu del cuerpo calloso. Estos 5 fascículos presentan una forma intrincada debido a que no tienen una forma cilíndrica única, ya que presentan varios subfascículos con formas distintas.

En la simulación se mantuvieron fijos los parámetros de radio de curvatura ($f_i = 1$) y ruido gaussiano (media=0, desviación estándar=0.8). Se varió la cantidad de centroides (5 y 10) y el número de fibras seleccionadas (10, 20, 30), obteniendo 6 simulaciones por cada fascículo. Para cuantificar los resultados de la simulación, se utilizaron las métricas de evaluación de Bundle Adjacency (BA) A. II con $\theta = 3$ mm y la distancia Euclidiana media a la fibra más cercana, las cuales están explican con mayor detalle en el anexo.

La tabla 7.4 muestra los valores obtenidos de las métricas de evaluación para cada fascículo con las 6 combinaciones de parámetros. Se observa que al aumentar los valores de los parámetros, el puntaje BA aumenta considerablemente, lo que indica una mayor similitud entre el fascículo original y el simulado, y la distancia media de las fibras disminuye, junto con la desviación estándar. También se observa que el parámetro que más influye en la simulación es la cantidad de centroides.

Tabla 7.4: Distancia media y Puntaje BA de las simulaciones de fascículos

Fascículos	Parámetros del simulador		Metricas	
	Número de centroides	Fibras seleccionadas	Puntaje BA ($\theta = 3$)	Distancia media \pm desviacion estandar [mm]
Arqueado anterior	5	10	0.545	9.95 \pm 5.03
	5	20	0.610	8.01 \pm 5.10
	5	30	0.643	7.11 \pm 4.44
	10	10	0.660	6.70 \pm 4.58
	10	20	0.755	5.44 \pm 3.98
	10	30	0.778	4.83 \pm 3.47
Corticoespinal	5	10	0.714	7.07 \pm 3.48
	5	20	0.860	5.25 \pm 3.19
	5	30	0.876	4.67 \pm 2.98
	10	10	0.735	5.67 \pm 3.42
	10	20	0.911	4.43 \pm 2.93
	10	30	0.947	3.72 \pm 2.62
Fronto-occipital inferior	5	10	0.316	12.9 \pm 5.21
	5	20	0.369	12.9 \pm 5.79
	5	30	0.391	12.15 \pm 5.67
	10	10	0.477	9.49 \pm 3.99
	10	20	0.508	8.71 \pm 4.02
	10	30	0.543	8.22 \pm 3.95
Longitudinal inferior	5	10	0.381	12.1 \pm 6.77
	5	20	0.408	11.2 \pm 6.98
	5	30	0.402	10.5 \pm 6.87
	10	10	0.528	9.92 \pm 6.77
	10	20	0.540	8.93 \pm 6.92
	10	30	0.550	8.65 \pm 6.35
Cuerpo calloso GENU	5	10	0.505	10.4 \pm 6.27
	5	20	0.543	8.78 \pm 6.27
	5	30	0.613	8.44 \pm 6.40
	10	10	0.604	7.69 \pm 4.48
	10	20	0.713	7.17 \pm 4.90
	10	30	0.771	6.86 \pm 5.12

Los resultados cuantitativos muestran que el simulador presenta un buen desempeño considerando la estructura y forma compleja de los fascículos seleccionados. El fascículo corticoespinal obtuvo el mejor rendimiento, con un BA de 0.947 y una distancia media de 3.72 en la simulación con 10 centroides y 30 fibras seleccionadas. Le siguen el arqueado anterior con un BA de 0.778 y una distancia media de 4.83, y el cuerpo calloso Genu con un BA de 0.771 y una distancia media de 6.86. Estos resultados de rendimiento se pueden apreciar claramente al visualizar la simulación en las Fig. A.4, A.5 y A.6

Los fascículos fronto-occipital inferior y longitudinal inferior (Fig. A.7 y Fig. A.8) son los que presentan menor desempeño. Esto se podría deber a que las fibras de estos fascículos están demasiado dispersas y los fascículos no presenta una forma única.

7.3 Segmentación de Fibras Cerebrales

Para evaluar la segmentación propuesta en este trabajo se compararon los resultados del algoritmo con respecto al algoritmo de segmentación [1] proporcionado por el grupo de análisis de imágenes médicas de la Universidad de Concepción para 7 sujetos de la base de datos ARCHI [9]. Al aplicar algoritmos de Deep learning y machine learning es posible que los resultados estén sesgados debido al sobre-ajuste de los modelos obteniendo resultados que solo son aplicables para la base de datos original, por lo que también se compararon los resultados de ambas segmentaciones en 4 sujetos con autismo de la base de datos de sujetos con y sin autismo.

7.3.1 Base de Datos ARCHI

Se seleccionaron aleatoriamente 7 sujetos de la base de datos ARCHI, excluyendo aquellos utilizados para el entrenamiento del clasificador, con el fin de realizar una comparación de la segmentación entre ambos métodos. La Figura 7.42 muestra la cantidad de fibras segmentadas para cada fascículo utilizando ambos métodos, representadas mediante gráficos de caja. Se puede observar que el método propuesto en este trabajo permite obtener, en promedio, una mayor cantidad de fibras en los fascículos talámicos, cingulado de fibras largas (CG2), cingulado de fibras cortas (CG3) y en la mayoría de los fascículos arqueados. Sin embargo, se obtiene, en promedio, una menor cantidad de fibras para los fascículos del cuerpo calloso y el fascículo fronto-occipital inferior.

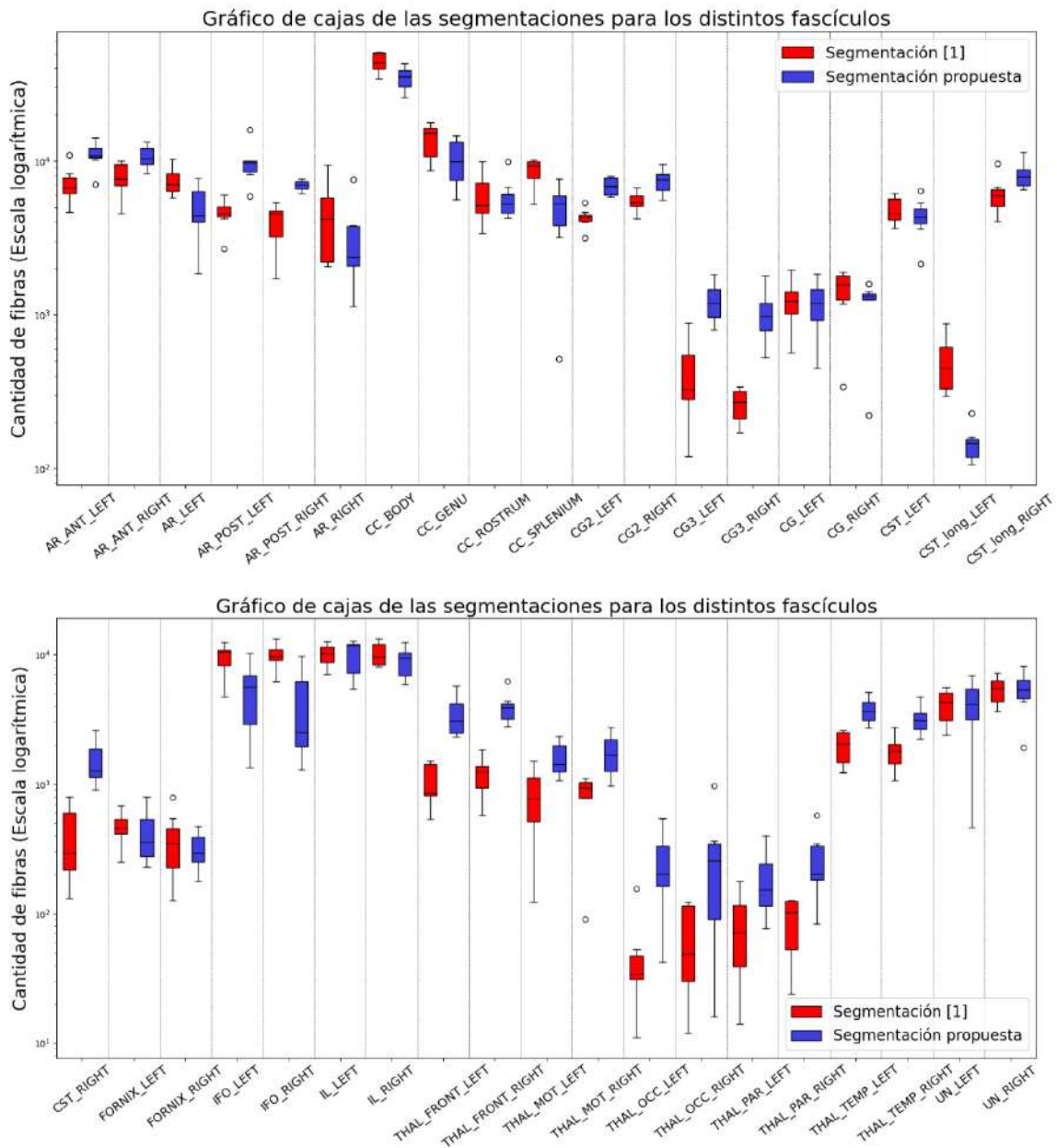


Fig. 7.42: Gráfico de cajas comparando las cantidades de fibras por fascículos para ambas segmentaciones.

Al visualizar las segmentaciones para ambos métodos Fig. 7.43, A.9, A.10, A.11 (se despliegan solamente los fascículos del hemisferios izquierdo debido a que no existe una gran diferencia entre hemisferios) se observa el claro aumento de fibras en los fascículos talámicos y una mejor segmentación para el fascículo arqueado directo (color rojo). Si bien el modelo obtiene una menor cantidad de fibras en los fascículos del cuerpo calloso se puede observar una segmentación más pulida sin tantas fibras outlier, espacialmente en el fascículo body del cuerpo calloso (color verde oscuro).

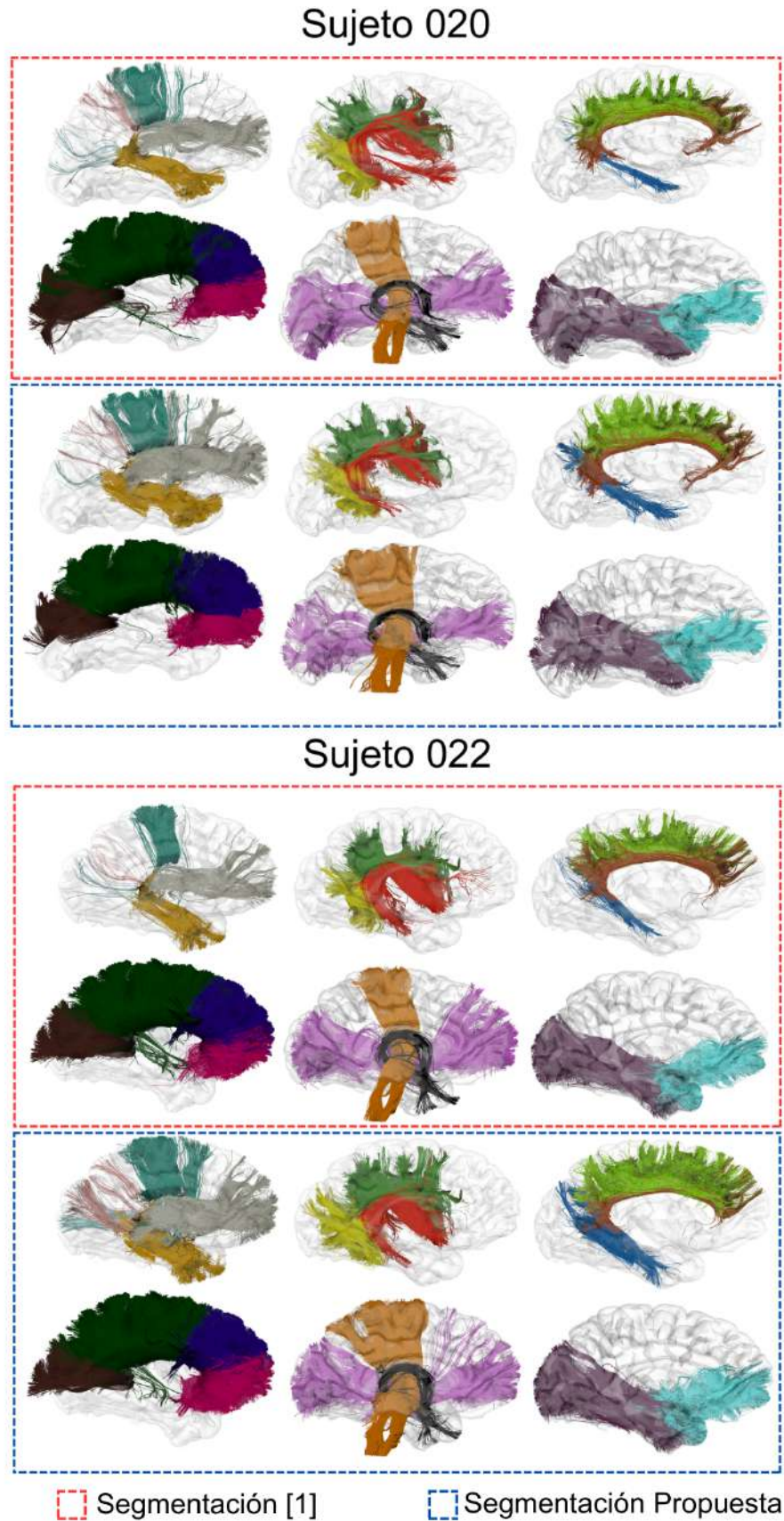


Fig. 7.43: Visualización de los fascículos segmentados del sujeto 020 y 022 para ambos métodos

7.3.2 Base de Datos de sujetos con autismo

Debido a que los conjuntos de tractografías de diferentes bases de datos presentan su información en espacios de coordenadas distintos, se llevó a cabo el entrenamiento de otro Autoencoder en el espacio estandarizado de Talairach. Esto se hizo para garantizar el correcto funcionamiento de la segmentación en conjuntos de tractografías provenientes de diversas bases de datos.

Dado que el espacio Talairach está centrado en el punto (0,0,0), las coordenadas de las fibras cerebrales pueden tener valores negativos en este espacio. Por lo tanto, se reemplazaron las primeras y últimas funciones de activación ReLU (Rectified Linear Unit) del modelo por funciones Leaky ReLU, que permiten la propagación de valores negativos a través de las capas subsiguientes.

En la tabla 7.5 se muestra la cantidad de fibras segmentadas para cada fascículo utilizando ambos métodos, así como la diferencia de fibras entre los métodos. Se observa que el método propuesto logra segmentar un 20 % más de fibras en 25 de los 38 fascículos analizados, lo cual representa una mejora significativa en términos de la cantidad de fibras segmentadas. Se observa una mejora notable en la segmentación de fascículos con una menor cantidad de fibras, como los talámicos y los uncinados. Sin embargo, se obtiene una menor cantidad de fibras para algunos fascículos específicos, como el fronto-occipital inferior izquierdo y derecho, el cingulado de fibras largas izquierdo y derecho, y los fascículos del cuerpo calloso (body, genu y splenium). En el anexo están las visualizaciones comparativas de ambos métodos para los 4 sujetos seleccionados Fig. A.12 A.13

Tabla 7.5: Comparación de la cantidad de fibras para los dos métodos de segmentación.

Fascículos	nl100461			hc100303			lk090309			ru100340		
	S1	S2	S2-S1	S1	S2	S2-S1	S1	S2	S2-S1	S1	S2	S2-S1
AR_ANT_LEFT	1536	6497	4961	6118	8356	2238	6215	8889	2674	6391	7954	1563
AR_ANT_RIGHT	4299	9153	4854	13253	18805	5552	5051	11799	6748	10648	14645	3997
AR_LEFT	8115	8571	456	9039	7537	-1502	1993	1296	-697	6924	2744	-4180
AR_POST_LEFT	2581	12016	9435	3834	11077	7243	4780	11298	6518	1387	4062	2675
AR_POST_RIGHT	6437	11120	4683	5428	9908	4480	2763	6434	3671	3075	4977	1902
AR_RIGHT	1863	1884	21	2848	2889	41	359	239	-120	2612	715	-1897
CC_BODY	43371	41005	-2366	40819	36438	-4381	62128	59433	-2695	47080	34309	-12771
CC_GENU	10459	7696	-2763	9596	6716	-2880	14028	10236	-3792	12042	7330	-4712
CC_ROSTRUM	4400	6597	2197	7474	8467	993	10826	12542	1716	5739	7837	2098
CC_SPLENIUM	6466	3910	-2556	8726	5965	-2761	13182	10371	-2811	5670	156	-5514
CG2_LEFT	3602	6384	2782	6027	8691	2664	5820	8775	2955	5774	8178	2404
CG2_RIGHT	5016	7014	1998	5051	9687	4636	6698	12006	5308	6243	9665	3422
CG3_LEFT	869	1593	724	711	1405	694	1183	2398	1215	87	712	625
CG3_RIGHT	202	787	585	806	1350	544	720	1449	729	123	384	261
CG_LEFT	3126	2341	-785	1588	470	-1118	3759	3073	-686	1214	473	-741
CG_RIGHT	2562	1596	-966	1447	655	-792	1957	1043	-914	504	138	-366
CST_LEFT	4436	4836	400	3061	3489	428	4440	5326	886	5095	5345	250
CST_long_LEFT	232	515	283	391	364	-27	127	197	70	387	208	-179
CST_long_RIGHT	4803	5780	977	4438	5204	766	2684	3242	558	3076	3693	617
CST_RIGHT	370	3140	2770	167	2070	1903	1256	3002	1746	2123	2752	629
FORNIX_LEFT	1020	1159	139	332	204	-128	31	16	-15	329	138	-191
FORNIX_RIGHT	177	434	257	191	140	-51	21	26	5	4	13	9
IFO_LEFT	8980	6695	-2285	8456	4680	-3776	6094	1781	-4313	7189	1580	-5609
IFO_RIGHT	6284	2960	-3324	8398	6724	-1674	11134	6905	-4229	8920	4228	-4692
IL_LEFT	13884	14358	474	10353	8512	-1841	13503	9549	-3954	9432	2224	-7208
IL_RIGHT	8331	10002	1671	9567	10765	1198	13249	13654	405	6956	5194	-1762
THAL_FRONT_LEFT	969	3641	2672	793	3297	2504	1353	4304	2951	423	2928	2505
THAL_FRONT_RIGHT	736	3181	2445	1492	4312	2820	1723	5035	3312	1867	5516	3649
THAL_MOT_LEFT	1588	2912	1324	65	709	644	385	739	354	1727	3014	1287
THAL_MOT_RIGHT	1170	2199	1029	72	1064	992	613	1121	508	587	3021	2434
THAL_OCC_LEFT	108	361	253	397	753	356	55	43	-12	166	309	143
THAL_OCC_RIGHT	242	453	211	36	375	339	251	427	176	7	482	475
THAL_PAR_LEFT	382	896	514	57	338	281	269	292	23	1019	798	-221
THAL_PAR_RIGHT	418	810	392	84	1085	1001	215	661	446	169	201	32
THAL_TEMP_LEFT	1886	3931	2045	2150	4344	2194	2048	4248	2200	1964	4252	2288
THAL_TEMP_RIGHT	587	1876	1289	3288	5602	2314	1125	3261	2136	524	2332	1808
UN_LEFT	3763	6575	2812	2512	5182	2670	3296	6440	3144	994	2860	1866
UN_RIGHT	4888	8833	3945	5545	7937	2392	4510	8282	3772	2208	4494	2286
TOTAL	170158	213711	43553	184610	215566	30956	209844	239832	29988	170679	159861	-10818

S1 representa el método [1]. S2 es el método propuesto en este trabajo. S2-S1 muestran la diferencia de fibras de estos dos métodos. Los valores coloreados de rojos indican que la segmentación S1 presentan más fibras en ese fascículo. Los valores coloreados en verde indican que la segmentación de S2 obtiene 20 % más de fibras para ese fascículo.

Capítulo 8. Conclusiones

8.1 Discusión

La estructura del Autoencoder utilizado permitió preservar la información original para ambas bases de datos de tractografía durante la reconstrucción una vez que el modelo fue entrenado, con un ligero cambio para la base de datos en el espacio de coordenadas Talairach. El procesamiento de las bases de datos y, específicamente, la optimización de los hiperparámetros, fueron clave para que el modelo obtuviera resultados de forma robusta y que el entrenamiento tarde el menor tiempo posible.

Si bien la simulación de fibras cerebrales mediante Autoencoder no es una idea nueva, ya que fue implementada previamente por Huerta et.al [3], el uso de otros parámetros como la selección e interpolación de fibras y la selección de centroides para simular los fascículos a nivel subfascículos fueron un punto clave para poder mejorar las simulaciones de fibra cerebrales. Al examinar las métricas de los resultados obtenidos para la simulación de fascículos complejos, se observa que la cantidad de centroides es el parámetro que tiene una mayor influencia en la mejora de la simulación de estos fascículos. Esto se debe a que los fascículos complejos no presentan una forma única, sino que están compuestos por distintos grupos o subfascículos con formas similares. Al aumentar la cantidad de centroides, el algoritmo tiene una mayor capacidad para capturar las variaciones y formas específicas de los subfascículos, lo que resulta en una simulación más precisa y similar al fascículo original.

En el diseño se explicó extensamente que la interpolación de fibras se puede también emplear para variar el radio de curvatura de los fascículos simulados, pero este parámetro no fue usado (se mantuvo fijo) para la obtención de resultados. Esto se debió a que se buscaba cuantificar la similitud entre el fascículo original y su simulación. Además, este parámetro fue creado para ser empleado en la evaluación de efectividad de algoritmos de clustering.

El modelo de segmentación propuesto permite obtener en promedio una mayor cantidad de fibras cerebrales tanto para sujetos sanos como para adolescentes con autismo, lo que permite mejorar el estudio de la conectividad de las áreas del cerebro y poder relacionar la variabilidad de estos fascículos con patologías. Esta mejora de la segmentación se debe a que la información contenida en el espacio latente no se limita únicamente a la ubicación de cada punto de una fibra para sus 3 ejes, sino que incluye las características esenciales que definen cada fibra en particular. Por lo que dos fibras cercanas entre sí en el espacio latente comparten más características que simplemente una ubicación y forma similares.

Debido a que el modelo propuesto de segmentación presentan una estructura simple, se puede modificar rápidamente la información de entrenamiento del clasificador para segmentar otros fascículos de materia blanca profunda y/o fascículos de materia blanca superficial. Si bien la estructura de este procedimiento no es compleja, en los ajustes de sus parámetros y los datos de entrenamiento del clasificador yace la clave para mejorar aún más la segmentación.

8.2 Conclusiones

Se logró implementar un Autoencoder que permite el procesamiento adecuado de la información de los conjuntos de datos de fibras cerebrales, tanto en el espacio propio de la base de datos ARCHI [9] como en el espacio normalizado de Talairach.

Utilizando el espacio latente, se logró desarrollar un simulador de fibras cerebrales que emplea algoritmos de clustering y técnicas de interpolación. Este simulador permite generar fascículos con una gran similitud respecto a los fascículos originales, a partir de cuatro parámetros distintos. Estos parámetros otorgan una gran versatilidad al modelo propuesto. De los resultados obtenidos en la simulación, se concluye que el parámetro *n_clusters* es el más relevante para simular fascículos complejos. Esto podría deberse a que estos fascículos complejos no presentan una forma única, por lo que dividir el fascículo en más grupos permite una reconstrucción más precisa de las diferentes formas de fascículos.

Además, se logró desarrollar un nuevo método de segmentación de fascículos basado en un atlas. Este enfoque utiliza el clasificador Radius Neighbor Classifier y la información del espacio latente. Gracias a ello, se logra recuperar al menos un 20% más de fibras para 25 de los 38 fascículos segmentados, tanto en sujetos sanos como en adolescentes con autismo. Estos resultados superan al método de segmentación actualmente empleado por el grupo de análisis de imágenes médicas de la Universidad de Concepción. Se considera que los resultados obtenidos en este trabajo son prometedores, lo que podría permitir un avance significativo tanto en la segmentación de fascículos como en la simulación de fibras cerebrales. Estos avances podrán ser validados con una mayor cantidad de datos y comparaciones en el futuro. Asimismo, se cree que estos resultados alentadores también pueden incentivar a otros investigadores a aplicar autoencoders y el uso del espacio latente en otros procesos relacionados con datos de tractografía, como por ejemplo la parcelación.

8.3 Trabajo Futuro

Si bien los resultados de este trabajo son positivos, existen varios puntos donde se podría profundizar el trabajo y mejorar diversos aspectos. Además, debido a limitaciones de tiempo, algunas ideas no pudieron ser implementadas, pero podrían ser interesantes para futuros trabajos. A continuación, se mencionan algunas de estas ideas:

- Entrenar el Autoencoder con datos de tractografía de sujetos jóvenes, adultos y con diferentes patologías en distintos espacios de coordenadas. Esto permitiría desarrollar un modelo inter-espacial capaz de procesar fibras de todos los tipos y tamaños.
- Emplear el simulador de fibras cerebrales para crear una tractografía simulada de cerebro completo. Esta simulación sería útil para evaluar la efectividad de diferentes algoritmos de clustering.
- Mejorar la selección de datos para el entrenamiento del clasificador y automatizar el proceso de selección del valor de radio de búsqueda para cada sujeto.
- Implementar el proceso de segmentación de fibras cerebrales para más fascículos de materia blanca profundos y para fascículos de materia blanca superficial.

Capítulo 9. Glosario

AE	Autoencoder
ANN	Red Neuronal Artificial , en inglés, Artificial Neural Network
ANNs	Redes Neuronales Artificiales , en inglés, Artificial Neural Networks
DL	Aprendizaje profundo , en inglés, Deep Learning
dMRI	Imagen de Resonancia Magnética por Difusión , en inglés, Diffusion Magnetic Resonance Imaging
DTI	Imágenes de tensor de difusión , en inglés, Diffusion Tensor Imaging
DWM	Materia Blanca Profunda , en inglés, Deep White Matter
FA	Anisotropía Fraccional , en inglés, Fractional Anisotropy
GT	Verdad Absoluta , en inglés, Ground Truth
HARDI	Imágenes de alta resolución angular , en inglés, High Angular Resolution Diffusion Imaging
IA	Inteligencia Artificial , en inglés, Artificial Intelligence
K-NN	K-vecinos más cercanos , en inglés, K-Nearest Neighbors
ML	Aprendizaje de Máquinas , en inglés, Machine Learning
MRI	Imagen de Resonancia Magnética , en inglés, Magnetic Resonance Im- aging
MSE	Error medio cuadrático , en inglés, Mean Squar Error
SNC	Sistema Nervioso Central
TPE	Tree-structured Parzen Estimator
WM	Materia Blanca , en inglés, White Matter

Capítulo 10. Referencias

- [1] A. Vázquez, N. López-López, N. Labra, M. Figueroa, C. Poupon, J.-F. Mangin, C. Hernández, and P. Guevara, “Parallel optimization of fiber bundle segmentation for massive tractography datasets,” in *2019 IEEE 16th International Symposium on Biomedical Imaging (ISBI 2019)*. IEEE, 2019, pp. 178–181.
- [2] E. Poo, C. Kublik, J. Houenou, C. Poupon, J.-F. Mangin, C. Hernández, and P. Guevara, “Fiber bundles simulator using exponential curves to validate fiber clustering algorithms,” in *18th International Symposium on Medical Information Processing and Analysis*, J. Brieva, P. Guevara, N. Lepore, M. G. Linguraru, L. Rittner, and E. R. C. M.D., Eds., vol. 12567, International Society for Optics and Photonics. SPIE, 2023, p. 125670J. [Online]. Available: <https://doi.org/10.1117/12.2669811>
- [3] I. Huerta, A. Vázquez, N. López-López, J. Houenou, C. Poupon, J.-F. Mangin, P. Guevara, and C. Hernández, “Inter-subject clustering of brain fibers from whole-brain tractography,” in *2020 42nd Annual International Conference of the IEEE Engineering in Medicine Biology Society (EMBC)*, 2020, pp. 1687–1691.
- [4] J. Legarreta, L. Petit, F. Rheault, G. Theaud, C. Lemaire, M. Descoteaux, and P.-M. Jodoin, “Filtering in tractography using autoencoders (FINTA),” *Medical Image Analysis*, vol. 72, p. 102126, 06 2021.
- [5] A. Vázquez, N. López-López, A. Sánchez, J. Houenou, C. Poupon, J.-F. Mangin, C. Hernández, and P. Guevara, “FFclust: Fast fiber clustering for large tractography datasets for a detailed study of brain connectivity,” *NeuroImage*, vol. 220, p. 117070, 2020. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S1053811920305565>
- [6] E. Garyfallidis, M.-A. Côté, F. Rheault, and M. Descoteaux, “Quickbundlesx: Sequential clustering of millions of streamlines in multiple levels of detail at record execution time,” 01 2016.
- [7] E. Garyfallidis, M. Brett, M. Correia, G. Williams, and I. Nimmo-Smith, “QuickBundles, a method for tractography simplification,” *Frontiers in Neuroscience*, vol. 6, p. 175, 12 2012.
- [8] J. H. Legarreta, L. Petit, P.-M. Jodoin, and M. Descoteaux, “Generative sampling in bundle tractography using autoencoders (GESTA),” *Medical Image Analysis*, vol. 85, p. 102761, 2023. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S1361841523000221>

- [9] C. Poupon and et al, “Connect/archi: an open database to infer atlases of the human brain connectivity,” *ESMRMB*, vol. 272, p. 2012, 2012.
- [10] H. Blumenfeld, *Neuroanatomy Through Clinical Cases*, 2nd ed. Sunderland, MA: Sinauer Associates, May 2010.
- [11] B. J. Jellison, A. S. Field, J. Medow, M. Lazar, M. S. Salamat, and A. L. Alexander, “Diffusion tensor imaging of cerebral white matter: a pictorial review of physics, fiber tract anatomy, and tumor imaging patterns,” *AJNR Am. J. Neuroradiol.*, vol. 25, no. 3, pp. 356–369, Mar. 2004.
- [12] P. B. Guevara Alvez, “Inference of a human brain fiber bundle atlas from high angular resolution diffusion imaging,” Theses, Université Paris Sud - Paris XI, Oct. 2011. [Online]. Available: <https://theses.hal.science/tel-00638766>
- [13] D. Le Bihan and M. Iima, “Diffusion magnetic resonance imaging: What water tells us about biological tissues,” *PLoS biology*, vol. 13, p. e1002203, 07 2015.
- [14] L. J. O’Donnell and C.-F. Westin, “An introduction to diffusion tensor image analysis,” *Neurosurg. Clin. N. Am.*, vol. 22, no. 2, pp. 185–96, viii, Apr. 2011.
- [15] “Chapter 5 - mathematics of diffusion tensor imaging,” in *Introduction to Diffusion Tensor Imaging (Second Edition)*, second edition ed., S. Mori and J.-D. Tournier, Eds. San Diego: Academic Press, 2014, pp. 33–37. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/B9780123983985000059>
- [16] “Chapter 8 - moving beyond DTI: High angular resolution diffusion imaging (hardi),” in *Introduction to Diffusion Tensor Imaging (Second Edition)*, second edition ed., S. Mori and J.-D. Tournier, Eds. San Diego: Academic Press, 2014, pp. 65–78. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/B9780123983985000084>
- [17] M. Descoteaux, *High Angular Resolution Diffusion Imaging (HARDI)*. John Wiley Sons, Ltd, 2015, pp. 1–25. [Online]. Available: <https://onlinelibrary.wiley.com/doi/abs/10.1002/047134608X.W8258>
- [18] S. N. Sotiropoulos and A. Zalesky, “Building connectomes using diffusion MRI: why, how and but,” *NMR Biomed.*, vol. 32, no. 4, p. e3752, Apr. 2019.
- [19] W. Chau and A. R. McIntosh, “The Talairach coordinate of a point in the MNI space: how to interpret it,” *NeuroImage*, vol. 25, no. 2, pp. 408–416, 2005. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S1053811904007554>

- [20] P. Basser, S. Pajevic, C. Pierpaoli, J. Duda, and A. Aldroubi, "In vivo fiber tractography using dt-mri data," *Magnetic resonance in medicine : official journal of the Society of Magnetic Resonance in Medicine / Society of Magnetic Resonance in Medicine*, vol. 44, pp. 625–32, 11 2000.
- [21] W. I. Essayed, F. Zhang, P. Unadkat, G. R. Cosgrove, A. J. Golby, and L. J. O'Donnell, "White matter tractography for neurosurgical planning: A topography-based review of the current state of the art," *NeuroImage: Clinical*, vol. 15, pp. 659–672, 2017. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S2213158217301444>
- [22] A. K. Jain, "Data clustering: 50 years beyond k-means," *Pattern Recognition Letters*, vol. 31, no. 8, pp. 651–666, 2010, award winning papers from the 19th International Conference on Pattern Recognition (ICPR). [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0167865509002323>
- [23] K. Bindra and A. Mishra, "A detailed study of clustering algorithms," in *2017 6th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO)*, 2017, pp. 371–376.
- [24] J. Oyelade, I. Isewon, O. Oladipupo, O. Emebo, Z. Omogbadegun, O. Aromolaran, E. Uwoghiren, D. Olaniyan, and O. Olawole, "Data clustering: Algorithms and its applications," in *2019 19th International Conference on Computational Science and Its Applications (ICCSA)*, 2019, pp. 71–81.
- [25] P. Guevara, C. Poupon, D. Rivière, Y. Cointepas, M. Descoteaux, B. Thirion, and J.-F. Mangin, "Robust clustering of massive tractography datasets," *NeuroImage*, vol. 54, no. 3, pp. 1975–1993, 2011. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S1053811910013200>
- [26] P. Guevara, D. Duclap, C. Poupon, L. Marrakchi-Kacem, P. Fillard, D. Le Bihan, M. Leboyer, J. Houenou, and J.-F. Mangin, "Automatic fiber bundle segmentation in massive tractography datasets using a multi-subject bundle atlas," *Neuroimage*, vol. 61, no. 4, pp. 1083–1099, Jul. 2012.
- [27] Y. Mintz and R. Brodie, "Introduction to artificial intelligence in medicine," *Minim. Invasive Ther. Allied Technol.*, vol. 28, no. 2, pp. 73–81, Apr. 2019.
- [28] F. Rosenblatt, "The perceptron - a perceiving and recognizing automaton," Cornell Aeronautical Laboratory, Ithaca, New York, Tech. Rep. 85-460-1, January 1957.
- [29] H. Ahmed and A. K. Nandi, *Artificial Neural Networks (ANNs)*, 2019, pp. 239–258.

- [30] S. Razavi, "Deep learning, explained: Fundamentals, explainability, and bridgeability to process-based modelling," *Environmental Modelling Software*, vol. 144, p. 105159, 2021. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S1364815221002024>
- [31] R. Chauhan, K. K. Ghanshala, and R. Joshi, "Convolutional neural network (cnn) for image detection and recognition," in *2018 First International Conference on Secure Cyber Computing and Communication (ICSCCC)*, 2018, pp. 278–282.
- [32] S. Albawi, T. A. Mohammed, and S. Al-Zawi, "Understanding of a convolutional neural network," in *2017 International Conference on Engineering and Technology (ICET)*, 2017, pp. 1–6.
- [33] D. Dai, "An introduction of cnn: Models and training on neural network models," in *2021 International Conference on Big Data, Artificial Intelligence and Risk Management (ICBAR)*, 2021, pp. 135–138.
- [34] L. Alzubaidi, J. Zhang, A. J. Humaidi, A. Al-Dujaili, Y. Duan, O. Al-Shamma, J. Santamaria, M. A. Fadhel, M. Al-Amidie, and L. Farhan, "Review of deep learning: concepts, CNN architectures, challenges, applications, future directions," *Journal of Big Data*, vol. 8, no. 1, Mar. 2021. [Online]. Available: <https://doi.org/10.1186/s40537-021-00444-8>
- [35] A. Ajit, K. Acharya, and A. Samanta, "A review of convolutional neural networks," in *2020 International Conference on Emerging Trends in Information Technology and Engineering (ic-ETITE)*, 2020, pp. 1–5.
- [36] W.-Y. Lee, S.-M. Park, and K.-B. Sim, "Optimal hyperparameter tuning of convolutional neural networks based on the parameter-setting-free harmony search algorithm," *Optik*, vol. 172, pp. 359–367, 2018. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0030402618310167>
- [37] R. Yamashita, M. Nishio, R. K. G. Do, and K. Togashi, "Convolutional neural networks: an overview and application in radiology," *Insights Imaging*, vol. 9, no. 4, pp. 611–629, Aug. 2018.
- [38] G. E. Hinton and R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *Science*, vol. 313, no. 5786, pp. 504–507, 2006, cited By :12839. [Online]. Available: www.scopus.com
- [39] F. Dumais, J. H. Legarreta, C. Lemaire, P. Poulin, F. Rheault, L. Petit, M. Descoteaux, and P.-M. Jodoin, "Fiesta: Autoencoders for accurate fiber segmentation in tractography," 2022.

- [40] J. Katz, M.-A. d’Albis, J. Boisgontier, C. Poupon, J.-F. Mangin, P. Guevara, D. Duclap, N. Hamdani, J. Petit, D. Monnet, P. Le Corvoisier, M. Leboyer, R. Delorme, and J. Houenou, “Similar white matter but opposite grey matter changes in schizophrenia and high-functioning autism,” *Acta Psychiatr. Scand.*, vol. 134, no. 1, pp. 31–39, Jul. 2016.
- [41] E. Garyfallidis, M. Brett, B. Amirbekian, A. Rokem, S. Van Der Walt, M. Descoteaux, and I. Nimmo-Smith, “Dipy, a library for the analysis of diffusion mri data,” *Frontiers in Neuroinformatics*, vol. 8, 2014. [Online]. Available: <https://www.frontiersin.org/articles/10.3389/fninf.2014.00008>
- [42] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, “Pytorch: An imperative style, high-performance deep learning library,” in *Advances in Neural Information Processing Systems 32*. Curran Associates, Inc., 2019, pp. 8024–8035.
- [43] T. Akiba, S. Sano, T. Yanase, T. Ohta, and M. Koyama, “Optuna: A next-generation hyperparameter optimization framework,” in *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2019.
- [44] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, “Scikit-learn: Machine learning in Python,” *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.
- [45] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu, “A density-based algorithm for discovering clusters in large spatial databases with noise,” in *Knowledge Discovery and Data Mining*, 1996.
- [46] L. van der Maaten and G. Hinton, “Visualizing data using t-SNE,” *Journal of Machine Learning Research*, vol. 9, pp. 2579–2605, 2008. [Online]. Available: <http://www.jmlr.org/papers/v9/vandermaaten08a.html>
- [47] K. G. Schilling, F. Rheault, L. Petit, C. B. Hansen, V. Nath, F.-C. Yeh, G. Girard, M. Barakovic, J. Rafael-Patino, T. Yu, E. Fischi-Gomez, M. Pizzolato, M. Ocampo-Pineda, S. Schiavi, E. J. Canales-Rodríguez, A. Daducci, C. Granziera, G. Innocenti, J.-P. Thiran, L. Mancini, S. Wastling, S. Coccozza, M. Petracca, G. Pontillo, M. Mancini, S. B. Vos, V. N. Vakharia, J. S. Duncan, H. Melero, L. Manzanedo, E. Sanz-Morales, Ángel Peña-Melián, F. Calamante, A. Attyé, R. P. Cabeen, L. Korobova, A. W. Toga, A. A. Vijayakumari, D. Parker, R. Verma, A. Radwan,

S. Sunaert, L. Emsell, A. De Luca, A. Leemans, C. J. Bajada, H. Haroon, H. Azadbakht, M. Chamberland, S. Genc, C. M. Tax, P.-H. Yeh, R. Srikanchana, C. D. Mcknight, J. Y.-M. Yang, J. Chen, C. E. Kelly, C.-H. Yeh, J. Cochereau, J. J. Maller, T. Welton, F. Almairac, K. K. Seunarine, C. A. Clark, F. Zhang, N. Makris, A. Golby, Y. Rathi, L. J. O'Donnell, Y. Xia, D. B. Aydogan, Y. Shi, F. G. Fernandes, M. Raemaekers, S. Warrington, S. Michielse, A. Ramírez-Manzanares, L. Concha, R. Aranda, M. R. Meraz, G. Lerma-Usabiaga, L. Roitman, L. S. Fekonja, N. Calarco, M. Joseph, H. Nakua, A. N. Voineskos, P. Karan, G. Grenier, J. H. Legarreta, N. Adluru, V. A. Nair, V. Prabhakaran, A. L. Alexander, K. Kamagata, Y. Saito, W. Uchida, C. Andica, M. Abe, R. G. Bayrak, C. A. G. Wheeler-Kingshott, E. D'Angelo, F. Palesi, G. Savini, N. Rolandi, P. Guevara, J. Houenou, N. López-López, J.-F. Mangin, C. Poupon, C. Román, A. Vázquez, C. Maffei, M. Arantes, J. P. Andrade, S. M. Silva, V. D. Calhoun, E. Caverzasi, S. Sacco, M. Lauricella, F. Pestilli, D. Bullock, Y. Zhan, E. Brignoni-Perez, C. Lebel, J. E. Reynolds, I. Nestratil, R. Labounek, C. Lenglet, A. Paulson, S. Aulicka, S. R. Heilbronner, K. Heuer, B. Q. Chandio, J. Guaje, W. Tang, E. Garyfallidis, R. Raja, A. W. Anderson, B. A. Landman, and M. Descoteaux, "Tractography dissection variability: What happens when 42 groups dissect 14 white matter bundles on the same dataset?" *NeuroImage*, vol. 243, p. 118502, 2021. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S1053811921007758>

- [48] B. Q. Chandio, S. L. Risacher, F. Pestilli, D. Bullock, F.-C. Yeh, S. Koudoro, A. Rokem, J. Harezlak, and E. Garyfallidis, "Bundle analytics, a computational framework for investigating the shapes and profiles of brain pathways across populations," *Sci. Rep.*, vol. 10, no. 1, p. 17149, Oct. 2020.

Anexo A

A. I Segmentación de fascículos

Tabla A.1: Cantidad de fibras segmentadas por fascículo.

Fascículos	Sujetos				
	079	078	077	076	075
AR_ANT_LEFT	5100	5615	2055	10152	9894
AR_ANT_RIGHT	8166	10528	3904	15464	15723
AR_LEFT	6716	12852	9342	10828	8028
AR_POST_LEFT	6449	3707	5792	5293	2067
AR_POST_RIGHT	2537	2312	5314	5047	2839
AR_RIGHT	5468	2256	3616	2097	3980
CC_BODY	43403	41404	34204	62102	39363
CC_GENU	10962	11712	12364	14683	12442
CC_ROSTRUM	3075	5527	6382	5966	6227
CC_SPLENIUM	12107	8034	7592	7972	9883
CG2_LEFT	3330	3641	3518	4650	5636
CG2_RIGHT	4404	5933	3781	5577	5738
CG3_LEFT	567	213	96	312	219
CG3_RIGHT	111	393	165	310	218
CG_LEFT	1315	1267	923	1921	832
CG_RIGHT	1494	1253	1058	3428	1287
CST_LEFT	5025	3882	4061	6201	3683
CST_long_LEFT	443	632	283	591	321
CST_long_RIGHT	4310	4567	4336	7546	4302
CST_RIGHT	166	204	583	1404	317
FORNIX_LEFT	690	201	380	931	1013
FORNIX_RIGHT	451	383	143	430	1053
IFO_LEFT	9145	6947	5400	10117	9788
IFO_RIGHT	9907	7876	6257	6023	10766
IL_LEFT	8876	7176	9593	7707	2980
IL_RIGHT	5878	8276	7048	4821	9682
THAL_FRONT_LEFT	2096	850	1112	1479	825
THAL_FRONT_RIGHT	1631	1293	1191	1192	1606
THAL_MOT_LEFT	868	1001	708	641	986
THAL_MOT_RIGHT	1114	2694	639	903	323
THAL_OCC_LEFT	83	26	45	22	38
THAL_OCC_RIGHT	191	50	11	100	70
THAL_PAR_LEFT	223	60	125	183	48
THAL_PAR_RIGHT	76	204	15	401	30
THAL_TEMP_LEFT	2870	1447	2195	1224	2017
THAL_TEMP_RIGHT	1580	782	1244	1210	1703
UN_LEFT	4521	4871	5563	3658	2635
UN_RIGHT	4603	4098	6240	6187	6387
TOTAL	179951	174167	157278	218773	184949

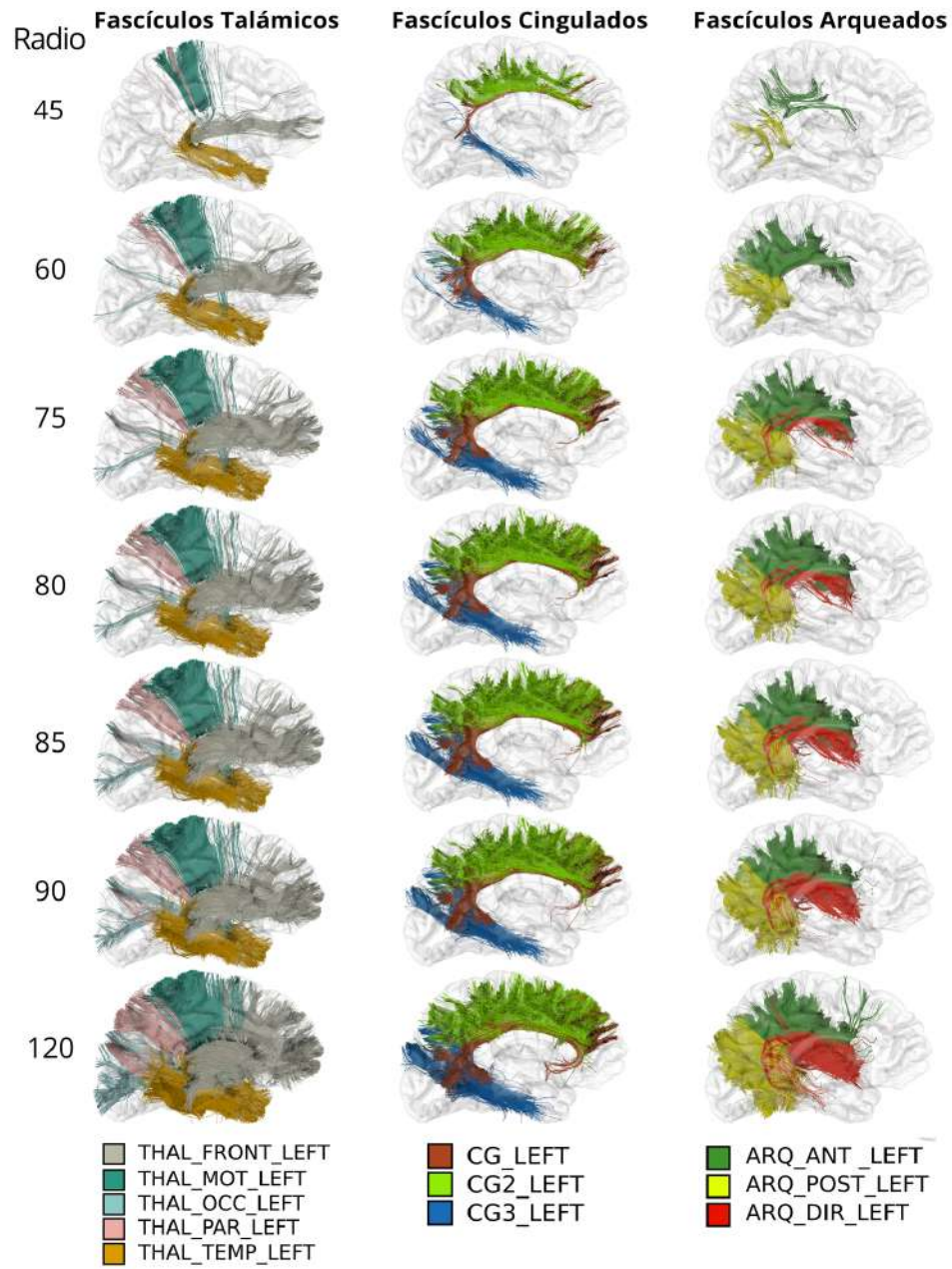


Fig. A.1: Segmentación de fascículos talámicos, cingulados y arqueados del hemisferio izquierdo para distintos valores de radio de búsqueda.

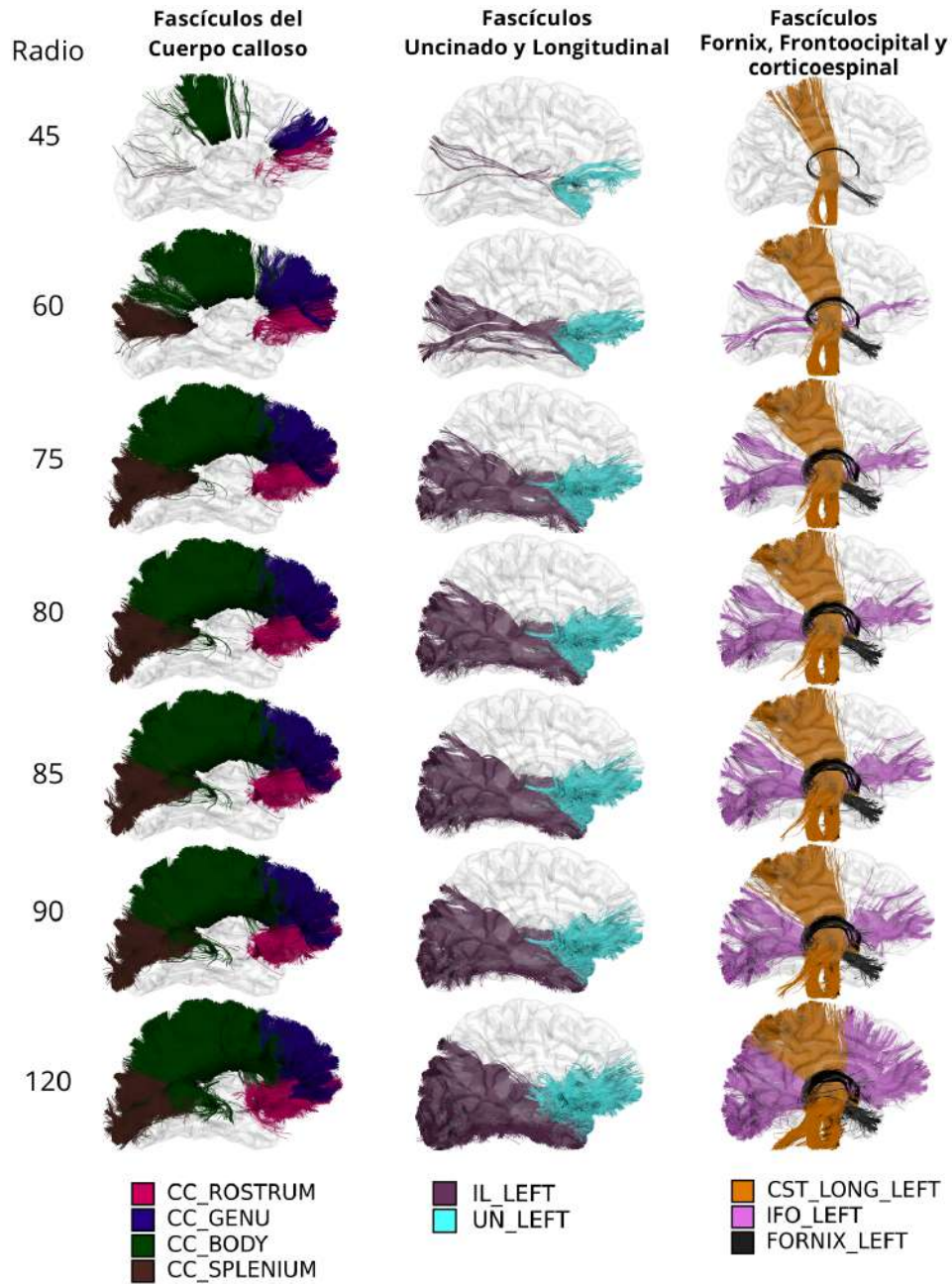


Fig. A.2: Segmentación del fascículo corticoespinal, uncinados, fornix, frontooccipital, longitudinal del hemisferio izquierdo y de los fascículos del cuerpo calloso, para distintos valores de radio de búsqueda.

A. II Métricas de Evaluación para la Simulación de Fibras

Para poder cuantificar los resultados de la simulación, se calculó el promedio de la distancia Euclidiana de las fibras cerebrales simuladas a la fibra original más cercana, restringiendo que más de una fibra simulada utilice la misma fibra original para el cálculo de la distancia, ya que esto alteraría los resultados. Por ejemplo, si el fascículo simulado estuviera conformado por copias de una fibra original, la distancia calculada sin esta restricción sería 0.

También se usó la métrica *bundle adjacency* (BA) [48], la cual está implementada como una función en la biblioteca de Dipy [41]. *Bundle adjacency* (BA) utiliza la distancia mínima directa y su inversa (MDF) para calcular la distancia entre dos fibras. *Bundle adjacency* (BA) se obtiene dado un umbral de adyacencia $\theta > 0$, y se dice que un conjunto de fibras $b_1 \in B_1$ es adyacente a B_2 si hay al menos una fibra $b_2 \in B_2$ con $MDF(b_1, b_2) \leq \theta$. La cobertura de B_1 por B_2 se define como la fracción de B_1 que es adyacente a B_2 . La cobertura varía entre 0 (cuando todas las fibras en B_1 están demasiado lejos de B_2) y 1 (cuando cada fibra en B_1 es adyacente a B_2) [48].

$$BA(B_1, B_2) = 0.5 (\text{cobertura}(B_1, B_2) + \text{cobertura}(B_2, B_1)) \quad (\text{A. II.1})$$

Para el cálculo de BA se necesitan los conjuntos de fibras y el valor de θ , donde θ representa la distancia en milímetros utilizada para calcular la cobertura entre fascículos. Si se elige un valor de θ demasiado alto, BA tenderá a ser 1, como se muestra en la Figura A.3.

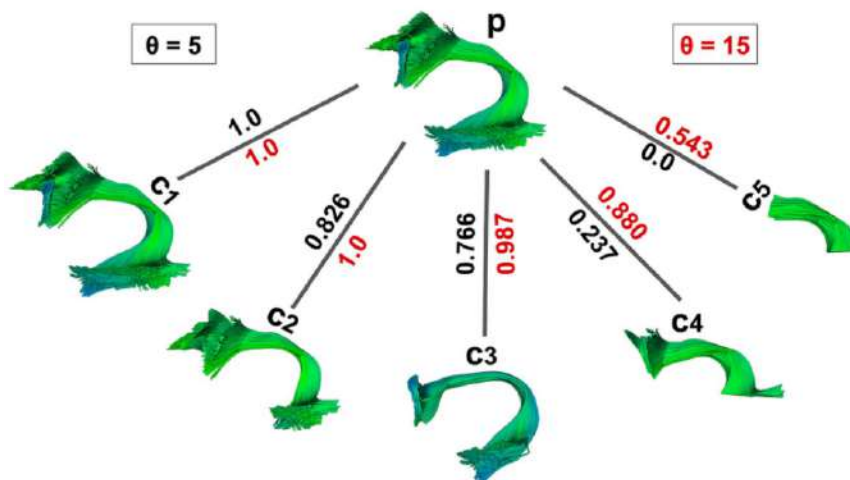


Fig. A.3: Valores BA para distintos fascículos con θ igual a 5 y 15.

Imagen extraída de [48]

A. III Resultados

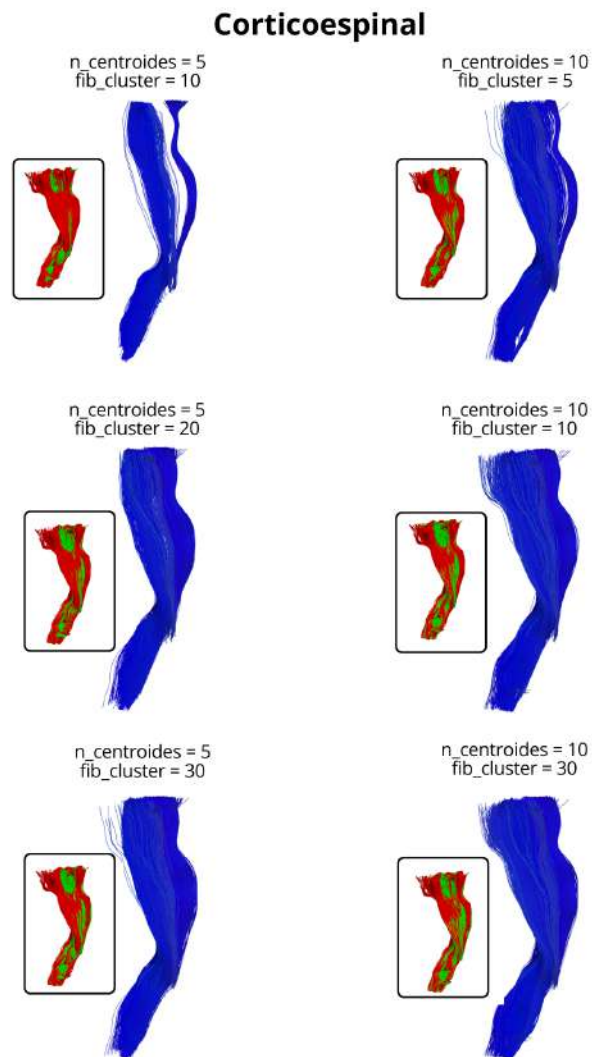


Fig. A.4: Imágenes de las simulaciones del fascículo corticoespinal izquierdo. El recuadro en negro muestra el fascículo original en rojo y la simulación en color verde

Arqueado anterior

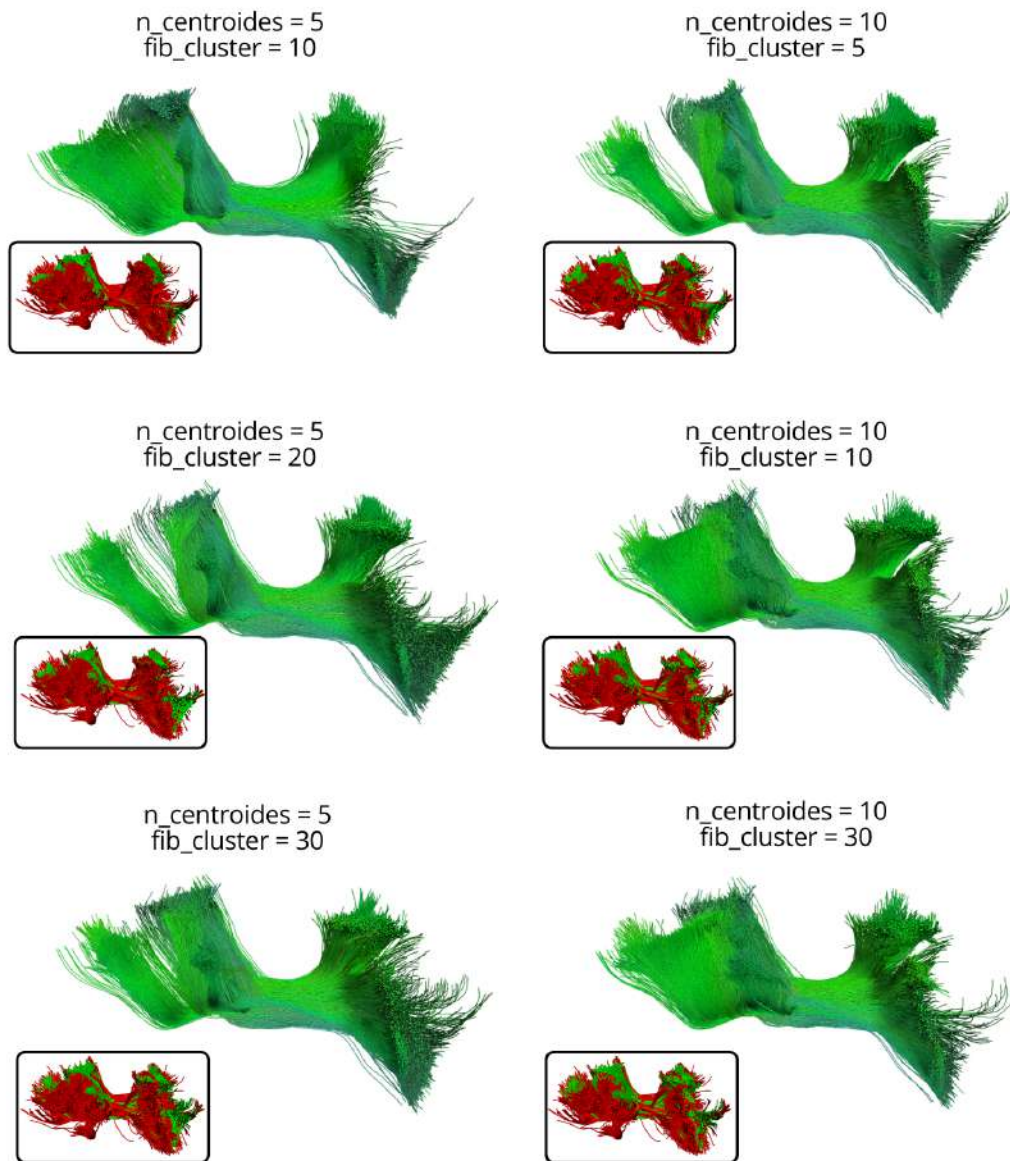


Fig. A.5: Imágenes de las simulaciones del fascículo arqueado anterior izquierdo. El recuadro en negro muestra el fascículo original en rojo y la simulación en color verde

CC_GENU

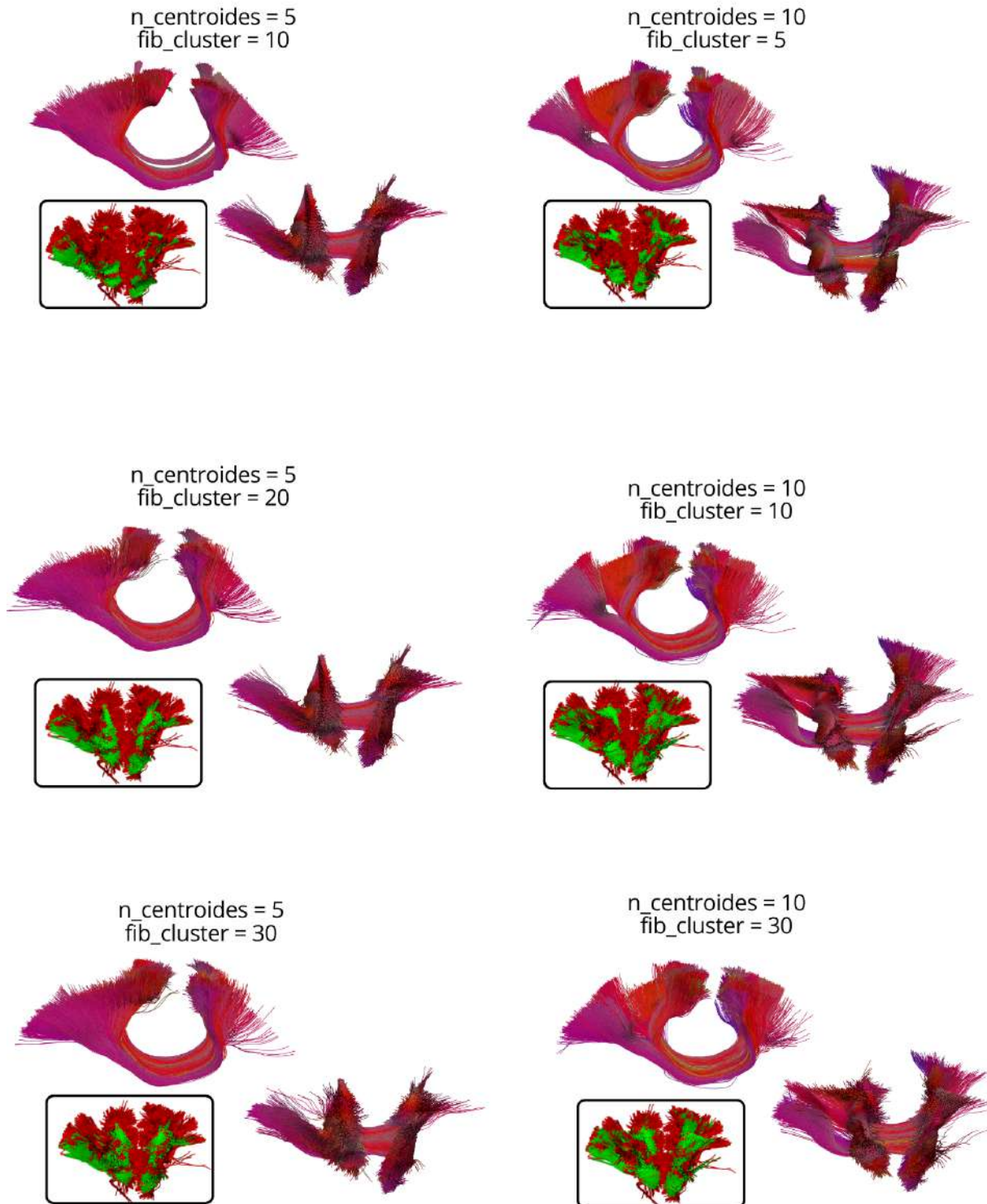


Fig. A.6: Imágenes de las simulaciones del fascículo del cuerpo calloso GENU. El recuadro en negro muestra el fascículo original en rojo y la simulación en color verde

Fronto-occipital inferior

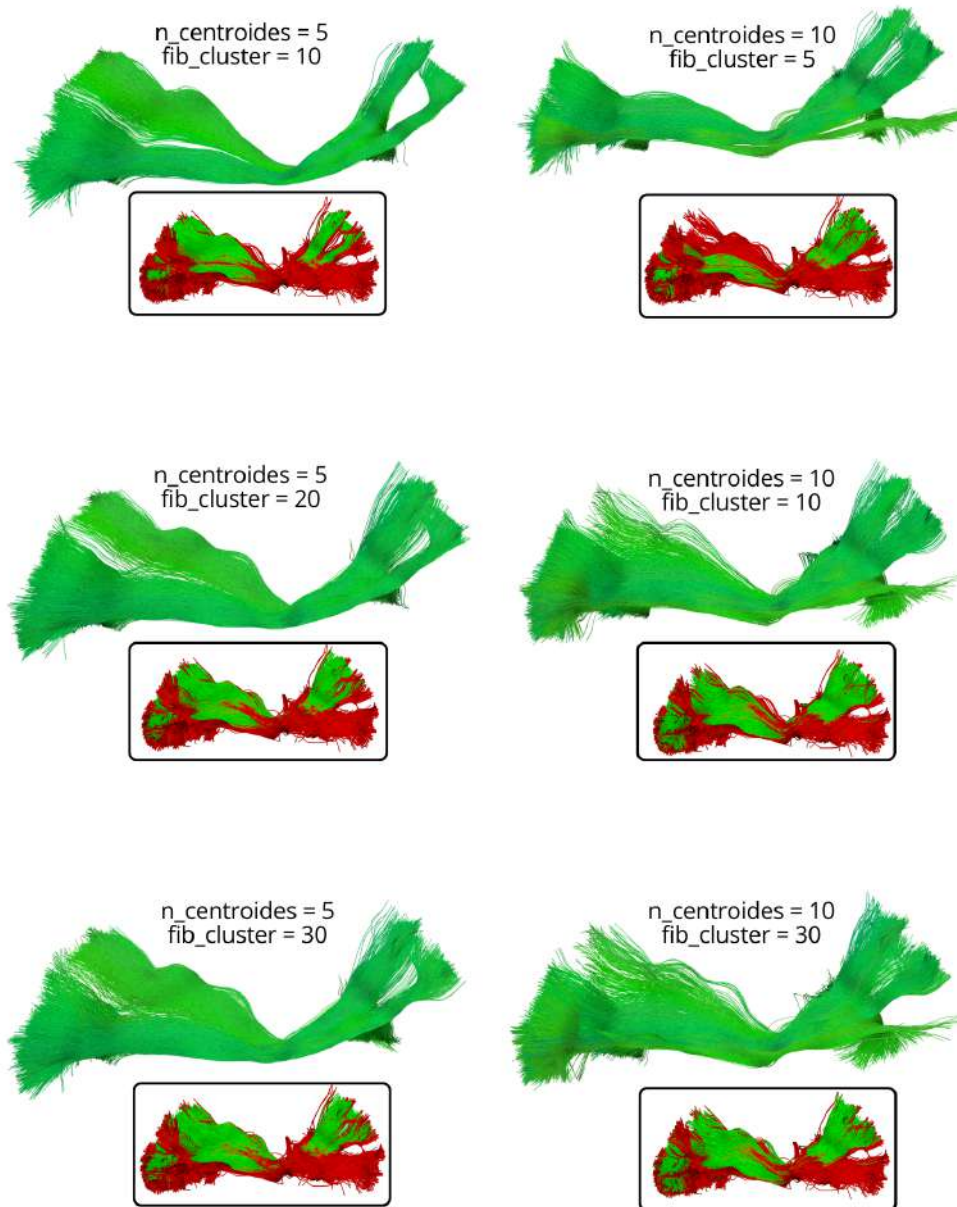


Fig. A.7: Imágenes de las simulaciones del fascículo fronto-occipital inferior derecho. El recuadro en negro muestra el fascículo original en rojo y la simulación en color verde

Longitudinal inferior

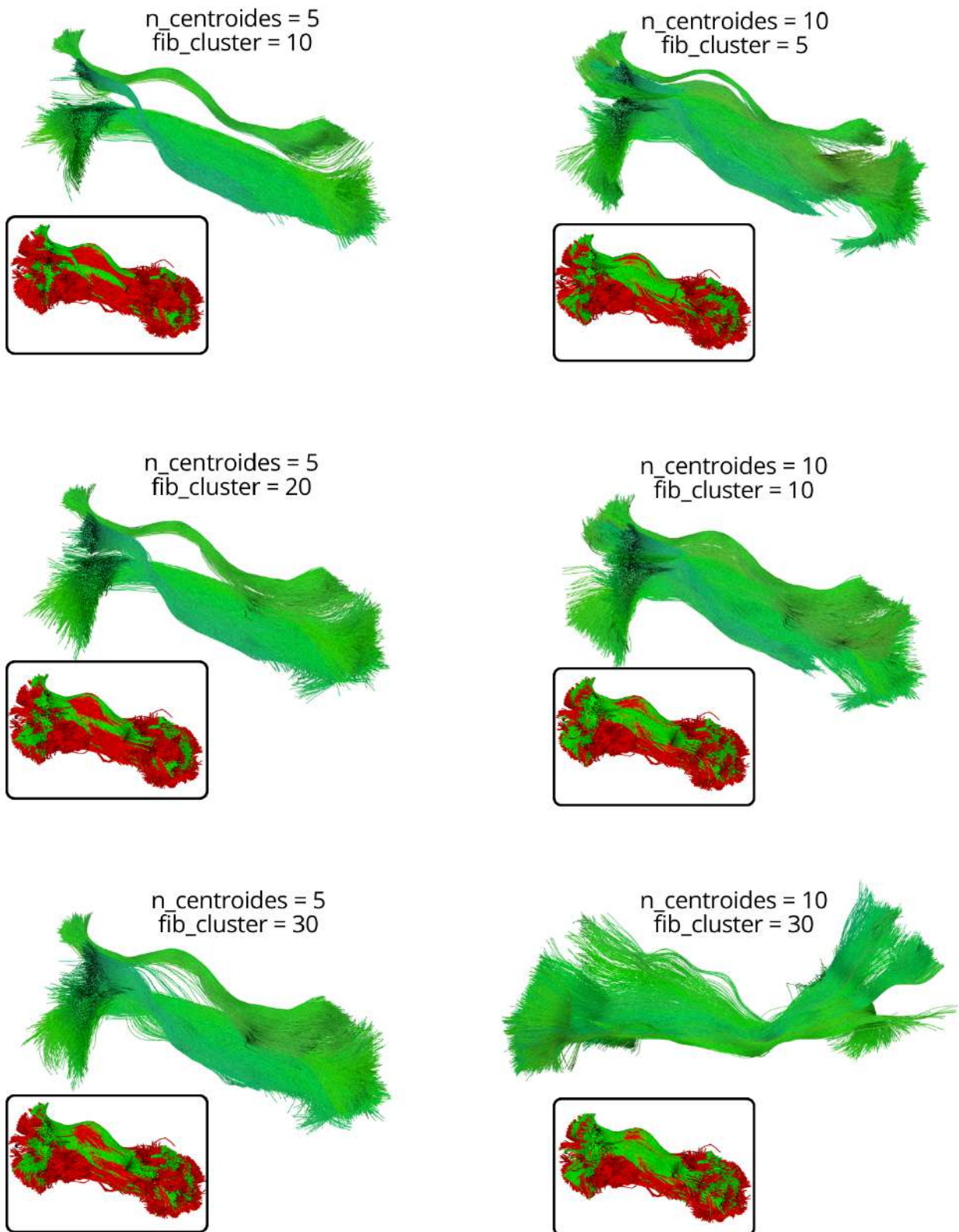


Fig. A.8: Imágenes de las simulaciones del fascículo longitudinal inferior izquierdo. El recuadro en negro muestra el fascículo original en rojo y la simulación en color verde

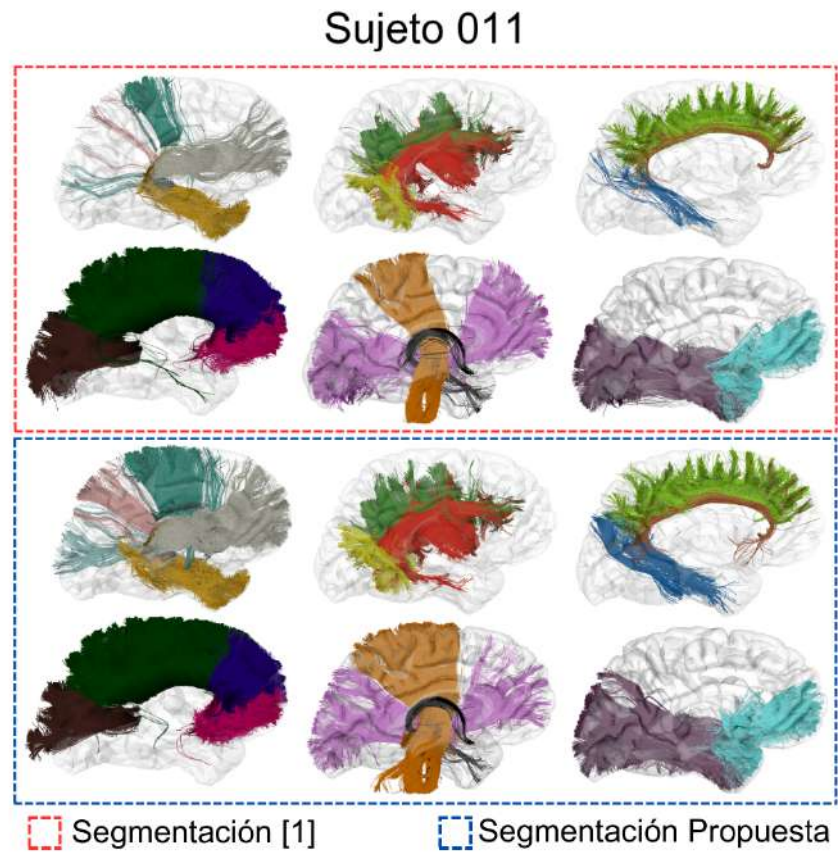
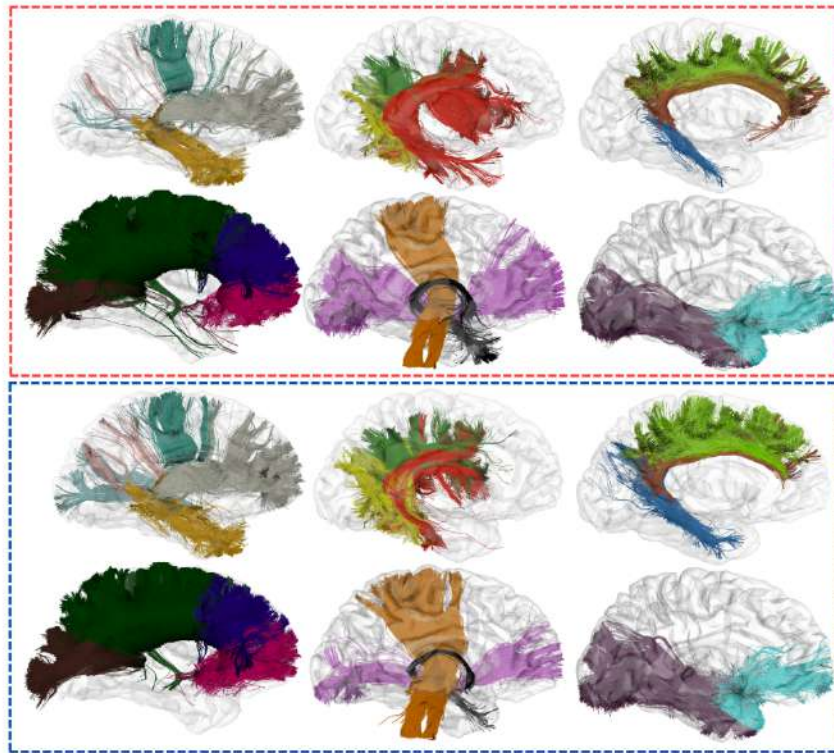
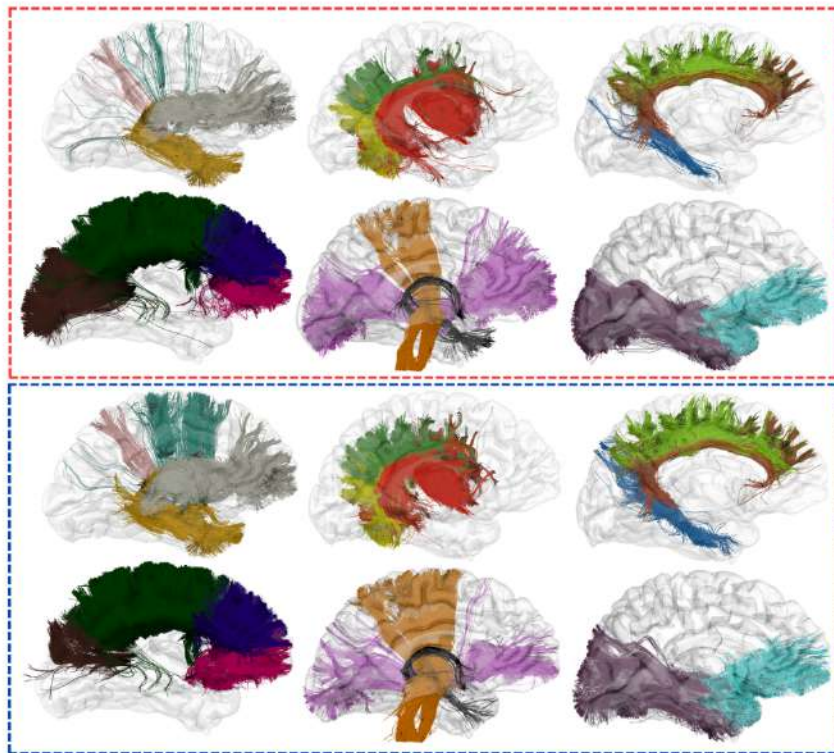


Fig. A.9: Visualización de los fascículos segmentados del sujeto 011 para ambos métodos

Sujeto 026



Sujeto 050

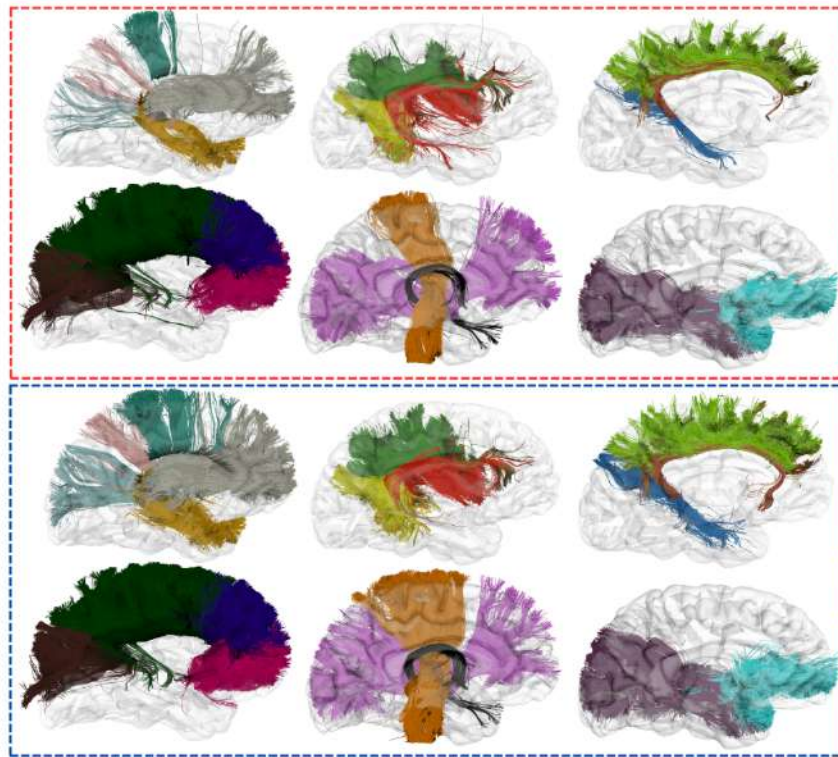


Segmentación [1]

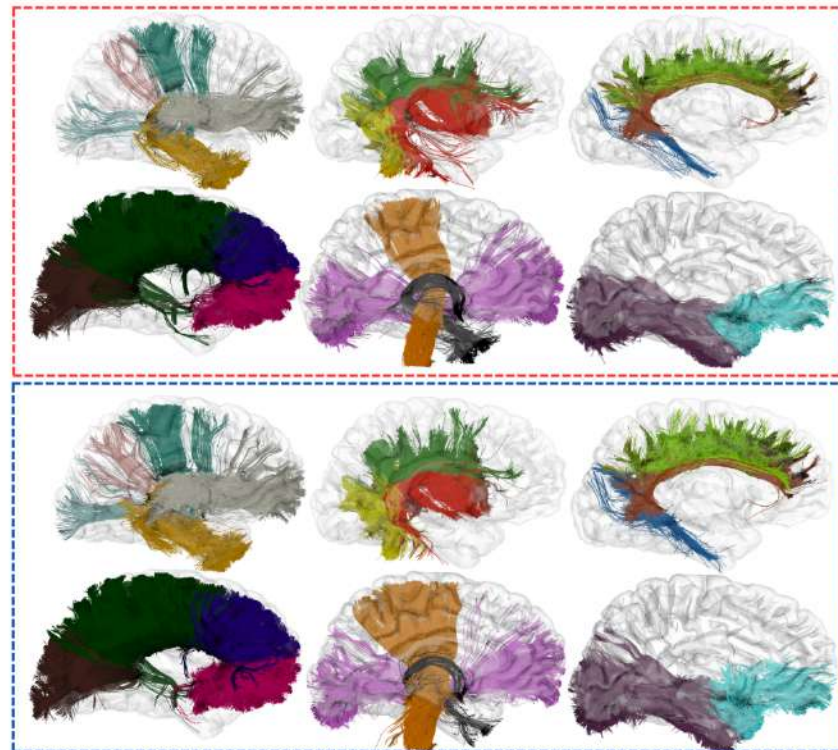
Segmentación Propuesta

Fig. A.10: Visualización de los fascículos segmentados del sujeto 026 y 050 para ambos métodos

Sujeto 052



Sujeto 060



Segmentación [1]

Segmentación Propuesta

Fig. A.11: Visualización de los fascículos segmentados del sujeto 052 y 060 para ambos métodos

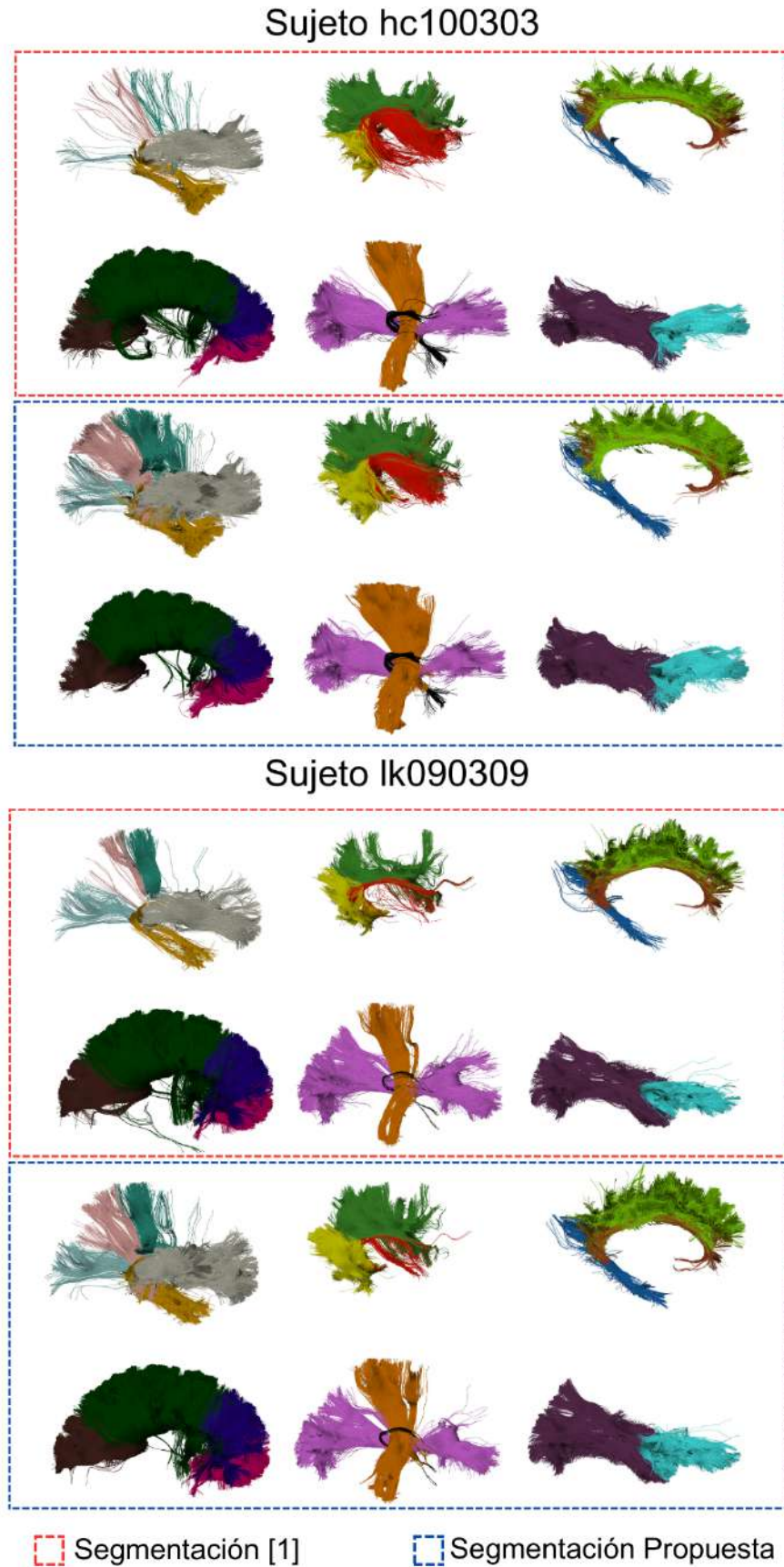
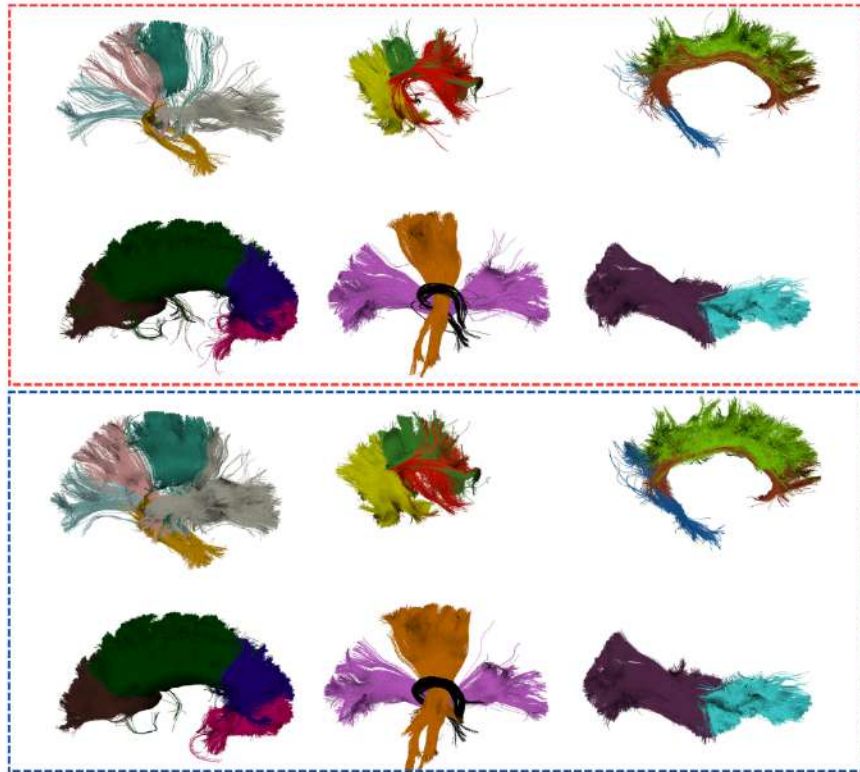
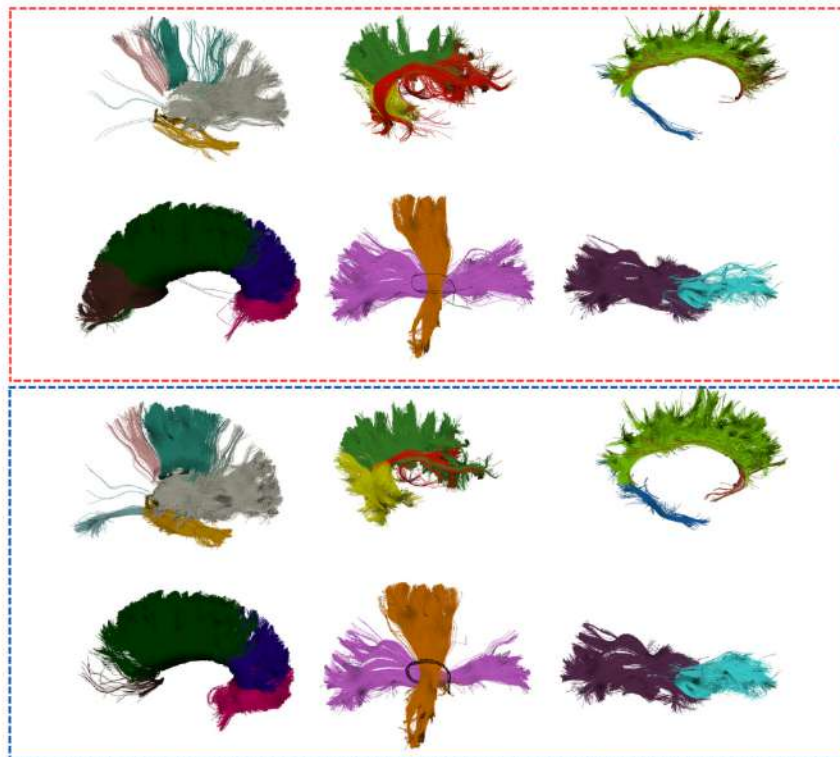


Fig. A.12: Visualización de los fascículos segmentados del sujeto hc100303 y lk090309 para ambos métodos

Sujeto nl100461



Sujeto ru100340



Segmentación [1]

Segmentación Propuesta

Fig. A.13: Visualización de los fascículos segmentados del sujeto nl100461 y ru100340 para ambos métodos

UNIVERSIDAD DE CONCEPCION – FACULTAD DE INGENIERIA
RESUMEN DE MEMORIA DE TITULO

Departamento	: Departamento de Ingeniería Eléctrica
Carrera	: Ingeniería Civil Biomédica
Nombre del memorista	: Sebastián Andrés Navarrete Caro
Título de la memoria	: Uso de Autoencoder para la segmentación y simulacion de fibras cerebrales
Fecha de la presentación oral	: 25 de Agosto 2023
Profesor(es) Guía	: Pamela Guevara A.
Profesor(es) Revisor(es)	: Cecilia Hernandez, Sebastián Godoy
Concepto	:
Calificación	:

Resumen

En las últimas décadas, el uso de neuroimágenes ha avanzado en el diagnóstico de trastornos cerebrales, destacando las tractografías cerebrales obtenidas por resonancia magnética, que visualizan fibras en tres dimensiones y permiten analizar la conectividad cerebral. El procesamiento de estas tractografías es complejo debido a la diversidad de formas y alta dimensionalidad de las fibras.

Los Autoencoders, modelos de aprendizaje profundo, se emplean para reducir la complejidad al transformar datos en un espacio latente y luego reconstruirlos. Aquí, se propone entrenar un Autoencoder con tractografías ARCHI para segmentar fascículos y simular fibras realistas mediante el espacio latente.

La segmentación de fascículos utiliza el algoritmo Radius Neighbor Classifier, que predice etiquetas según la distancia radial en datos de entrenamiento. La simulación de fascículos implica seleccionar fibras semilla e interpolar en el espacio latente, con técnicas de clustering para variar la curvatura.

Los resultados indican mejoras en la segmentación propuesta para 25 de 38 fascículos, incluyendo talámicos, cingulados y arqueados. Aunque en 13 de 38 hay menos fibras, la segmentación mejora, como en el cuerpo calloso. Con respecto a la simulación el modelo pude recrear con gran similitud fascículo con forma compleja.