Universidad Rey Juan Carlos

TESIS DOCTORAL

Técnicas de visualización interactiva para la exploración y análisis de datos complejos. Aplicación en el ámbito médico

Autor:

Iván Velasco González

Directores:

Sofía Bayona Beriso
Luis Pastor Pérez

Programa de Doctorado en Tecnologías de la Información y las Telecomunicaciones

Escuela Internacional de Doctorado

2023

Agradecimientos

Me gustaría agradecer en primer lugar a Óscar, compañero del grupo de investigación que me introdujo en esta andadura de la investigación allá por mi 3º año de carrera. Quién me iba a decir a mí que al final, entre todos, conseguirían engañarme para adentrarme en una carrera académica. Pero en gran parte, gracias a que él me dio esa primera oportunidad, he llegado hasta aquí.

Gracias a mis compañeros del despacho, que siempre están dispuestos a ayudar con cualquier duda. Especialmente gracias a Cosmin y a su inestimable ayuda con todo el papeleo relacionado con el doctorado y con los trámites tanto de la beca FPU como en el propio depósito de esta tesis. Mentiría si digo que no me ha salvado alguna vez de no dejarme un trámite sin realizar. Gracias a Aaron, por ser la cabeza del grupo en todo el tema de la burocracia de la carrera académica, el pobre ha experimentado todos los cambios en su propia piel y, gracias a su experiencia y consejos, el resto de los doctorandos hemos evitado muchos tropiezos.

Agradecer a mis directores de tesis, Sofía y Luis, que me han ayudado mucho en el desarrollo de la tesis. Qué decir de Luis, siempre anda hasta arriba de trabajo, y a la vez siempre es capaz de sacar un hueco para ayudarme con los asuntos complejos de estructura. Si no fuera por él, posiblemente este trabajo sería muy diferente. Sofía y yo nos conocemos desde que comencé a trabajar con ella, antes siquiera de empezar mi primer TFG, ¡cómo pasa el tiempo! La verdad que estoy muy agradecido de que la vida me pusiese a trabajar con ella. Nunca podré agradecerle lo suficiente el esfuerzo que ha invertido en el desarrollo de esta tesis, remando a contracorriente y contra reloj cuando a mí no me quedaban fuerzas para hacerlo. De verdad, gracias de corazón por apoyarme en los momentos difíciles.

Gracias a Tiberio y Nathalie, por todo lo que me habéis ofrecido y enseñado durante mi estancia en Colombia, que fue maravillosa gracias a vosotros. Agradeceros también por acogerme en vuestra casa. Nunca olvidaré el toque de campana por la mañana para el desayuno, gracias por hacerme sentir como uno más. A Alexis, por su soporte en cualquier duda que me surgiera en programación, por las tardes que hemos dedicado programando las herramientas y arreglando errores codo a codo, y, por qué no, también por tus videos explicando los conceptos de visualización y las herramientas relacionadas, gracias a ellos me fue más sencillo empezar a trabajar con las herramientas web.

A mi familia, que siempre me ha apoyado en todo lo que hago y me ha dado todo mientras me dedicaba a sacar mi doctorado adelante. Qué sería de mí sin su compresión y apoyo incondicional, o sin los *tuppers* de comida en épocas de exámenes, o de revisiones de artículos. Tampoco nos vamos a engañar, comer decentemente cuando no tienes tiempo ni de pensar es un privilegio.

A mis amigos por apoyarme en los momentos en los que estás quemado y te dan ganas de dejarlo todo. Si no fuese por ellos, posiblemente hubiese tirado la toalla en más de una ocasión, pero siempre están ahí con sus buenos consejos para animarme a no quedarme a mitad en el

camino. Al final, todos lo sabemos, la vida del doctorando es muy dura y es mejor hacer el viaje acompañado.

Agradecer también al Ministerio de Universidad por la ayuda FPU (FPU19/04516) de la que he disfrutado durante esta etapa de mi vida.

Por todo ello, muchas gracias a todos.

Resumen

En los últimos tiempos se ha producido un gran aumento en la cantidad de datos disponibles en muchos ámbitos. A su vez, los datos son cada vez más complejos debido a su heterogeneidad, que viene acompañada a veces de características temporales y espaciales, pudiendo presentar, además, tipos y patrones diversos junto con complicadas interrelaciones. Esta tesis busca aportar soluciones al desafío de la comprensión de los datos y la extracción del conocimiento. En particular, se centra en diseñar métodos y herramientas genéricos que permitan analizar, visualizar y comprender mejor los distintos tipos de datos, en particular en aquellos casos en los que la naturaleza espacial y temporal de los datos es fundamental. Así, el trabajo comienza centrándose en particular en datos de tipo morfológico filiformes que poseen una componente espacial intrínseca, para después abordar datos que incluyan la componente temporal.

Los datos de tipo morfológico filiformes son frecuentes en el campo de la medicina. Modelan estructuras delgadas y alargadas que incluyen ramificaciones. Los datos temporales, presentes prácticamente en todos los campos de la ciencia, pueden ser homogéneos, en cuyo caso todos los registros tienen información del mismo tipo de variables y cuentan con el mismo número de observaciones. También existen los datos temporales heterogéneos que modelan procesos complejos con datos cualitativos y cuantitativos, variables de distinto tipo, y distinto número de observaciones por registro. Aunque este trabajo se centra en mejorar la visualización y el análisis para estos tipos de datos de forma genérica, es necesario seleccionar algún dominio de aplicación que permita validar las soluciones aportadas. Se decidió escoger datos variados dentro del ámbito médico, como son los datos de morfologías filiformes neuronales, los datos temporales homogéneos de señales de electroencefalograma, y los datos temporales heterogéneos correspondientes al proceso de crecimiento de bebés prematuros, por su variedad y complejidad.

Respecto a los datos de tipo morfológico filiforme, se dispone de un conjunto de datos pertenecientes al dominio de la morfología neuronal. Estos datos están basados en un conjunto de mallas 3D y representan de manera precisa la forma de los elementos neuronales. No obstante, presentan la limitación de no disponer de información sobre la estructura y la jerarquía entre los mismos, lo que dificulta poder realizar análisis a nivel de la neurona completa. Para afrontar este problema, se propone generar una representación complementaria infiriendo la información de jerarquía, de modo que sea posible realizar dicho tipo de análisis. Actualmente, las representaciones neuronales incluyen nueva información sobre sus elementos. Sin embargo, la mayoría de las herramientas disponibles aún no ofrecen la posibilidad de aprovechar esta información y visualizarla. Por ello, se diseña una herramienta de visualización que incluya toda esta nueva información y que pueda representar toda la estructura al completo, todo ello sin perder el detalle de la forma de cada uno de los elementos.

Por otra parte, los datos de carácter temporal presentan problemáticas propias. Se basan en la monitorización de una o más variables durante un periodo determinado de tiempo y con un intervalo de medición, conociéndose como series temporales. Estas series temporales pueden tratarse con un enfoque univariante, donde cada variable se trata de forma individual, o bien con un enfoque multivariante, donde, además del análisis individual, se tiene en cuenta la relación entre las variables muestreadas. En este trabajo se proponen técnicas de análisis para datos temporales homogéneos, concretamente de clasificación, basándose en métodos multivariantes y en técnicas de extracción de características, obteniendo un excelente rendimiento.

Con todo, este tipo de análisis no es suficiente para comprender en profundidad datos complejos como pueden ser los datos temporales heterogéneos. En este caso, las técnicas de visualización centradas en el análisis exploratorio permitan al usuario familiarizarse con los datos por medio de la búsqueda de tendencias, la comprobación de hipótesis previas, y la formulación de nuevas hipótesis en base a la visualización. Con la intención de soportar estas funcionalidades, se propone una herramienta de visualización de datos temporales heterogéneos centrada en la creación y comparación de grupos de análisis. Además, debido a las necesidades cambiantes de la visualización, se diseña como una herramienta modular, altamente configurable, y reactiva que permita la fácil integración con otras herramientas, permitiendo así la adaptación a nuevas necesidades.

Los resultados obtenidos en las propuestas de visualización y análisis han sido muy positivos. Se ha conseguido generar de forma precisa la jerarquía de los datos morfológicos filiformes a partir de las mallas inconexas 3D y se han propuesto mejoras para la visualización de este tipo de datos de forma volumétrica. El clasificador propuesto para datos temporales homogéneos logró obtener un 100% de precisión, con la ventaja añadida de ser interpretable y explicable. Por su parte, la visualización centrada en datos temporales heterogéneos ha recibido muy buenas críticas de los expertos en el dominio, tanto en su versión individual, como cuando se incluye en un entorno de visualización en el que colaboran varias herramientas.

De estos trabajos se concluye que es posible desarrollar herramientas de visualización y análisis con un enfoque genérico que puedan ser utilizadas con distintos tipos de datos para mejorar la comprensión de grandes conjuntos de datos complejos de diversa naturaleza

Abstract

In recent times, there has been a large increase in the amount of data available in numerous domains. At the same time, data are becoming more complex due to their heterogeneity, which is sometimes combined with temporal and spatial characteristics, and may also present diverse types and patterns along with complicated interrelationships. This thesis aims to provide solutions to the challenge of data understanding and knowledge extraction. Specifically, it focuses on the design of generic methods and tools to improve the analysis, visualization and understanding of different types of data, especially when the spatial and temporal nature of the data is critical. Thus, the work begins by focusing specifically on morphological data that have an intrinsic spatial component, and then progresses to data that include the temporal component.

Filiform morphological data are common in the medical field. It models thin, elongated structures that include branching. Temporal data, present in practically all fields of science, can be homogeneous, meaning that all records have information on the same type of variables and have the same number of observations. There also exists heterogeneous temporal data that model complex processes with qualitative and quantitative data, variables of different types, and different numbers of observations per record. Although this work focuses on improving the visualization and analysis for these types of data in a generic way, it is necessary to select some application domain to validate the solutions proposed. We chose varied data within the medical domain, such as neuronal filiform morphology data, homogeneous temporal data of electroencephalogram signals, and heterogeneous temporal data corresponding to the growth process of premature infants, due to their variety and complexity.

Regarding the filiform morphological data, within the Human Brain Project, a data set belonging to the neuronal morphology domain was available. These data are based on a set of 3D meshes and accurately represent the shape of neuronal elements. However, the data suffer from the limitation of not having information about the structure and hierarchy between them, which makes it difficult to perform analyses at the level of the whole neuron. To address this problem, we propose to generate a complementary representation by inferring the hierarchy information, so as to make possible to perform this type of analysis. Currently, neural representations include new information about their elements. However, most of the available tools do not yet offer the possibility to take advantage of this information and visualize it. Therefore, a visualization tool is designed that includes all this new information and can represent the entire structure without losing the detail of the shape of each of the elements.

On the other hand, data of a temporal nature present their own problems. They are based on the monitoring of one or more variables over a given period of time and with a measurement interval, which is known as a time series. These time series can be treated with a univariate approach, where each variable is analyzed individually, or with a multivariate approach, where,

in addition to the individual analysis, the relationship between the sampled variables is taken into account. In this paper first we developed analysis techniques for homogeneous temporal data, specifically classification, based on multivariate methods and feature extraction techniques, obtaining an excellent performance.

However, this type of analysis is not sufficient for an extensive understanding of complex data such as heterogeneous temporal data. In this case, visualization techniques focused on exploratory analysis allow the user to become familiar with the data by searching for trends, testing previous hypotheses, and formulating new hypotheses based on the visualization. With the intention of supporting these functionalities, we propose a heterogeneous temporal data visualization tool focused on the creation and comparison of analysis groups. Furthermore, due to the changing needs of visualization, it is designed as a modular, highly configurable, and reactive tool that allows easy integration with other tools, allowing it to adapt to new requirements.

The results obtained in the visualization and analysis proposals have been very positive. The hierarchy of the filiform morphological data has been accurately generated from the 3D unconnected meshes and improvements have been made for the visualization of this type of data in a volumetric way. The proposed classifier for homogeneous temporal data achieved 100% accuracy, with the added advantage of being interpretable and explainable. The visualization focused on heterogeneous temporal data has received very good feedback from domain experts, both in its individual version and when included in a visualization environment in which several tools cooperate.

From these works, it is concluded that it is possible to develop visualization and analysis tools with a generic approach that can be used with different types of data to improve the understanding of large complex datasets of diverse nature.

Abreviaturas

AR: Autorregresivo (*Autoregressive*)
BCI: Interfaz cerebro-ordenador (*Brain Computer Interface*)
CD12: Coeficiente intelectual a los 12 meses
CD6: Coeficiente intelectual a los 6 meses
CMFMTS: Medidas de complejidad y características de las series temporales multivariantes (*Complexity Measures and Features for Multivariate Time Series*)
CNN: Red neuronal convolucional (*Convolutional Neural Network*)
CNN-LSTM: Convolutional Neural Network with Long Short-Term Memory
CSP: Patrones espaciales comunes (*Common Spatial Pattenrs*)
CWT: Transformada continua de ondículas (*Continous Wavelet Transform*)
D: Medida D de Hoefflin (*Hoefflin's D measure*)
DWT: Transformada Discreta de ondículas (*Discrete Wavelet Transform*)
ECG: Electrocardiograma (*Electrocardiogram*)
EEG: Electroencefalograma (*Electroencephalogram*)
ELM*:* Máquina de aprendizaje extremo *(Extreme Learning Machine)*
FEM: Método de elementos Finitos (*Finite Element Method*)
FFT: Transformada rápida de Fourier (*Fast Fourier Transformation*)
HBP: Proyecto cerebro humando (*Human Brain Project*)
IoT: Internet de las cosas (*Internet of Things*)
IQR: Rango intercuartílico (*Interquartile Range*)
KFCV: Validación cruzada de K-grupos (*K-fold Cross-Validation*)
KNN: K vecinos más cercanos (*K Nearest Neighbor*)
LDA: Análisis discriminante lineal (*Lineal Discriminant Analisys*)
LOOCV: Validación cruzada dejando uno fuera (*Leave-One-Out Cross-Validation*)
LTSM: Memoria a corto y largo plazo (*Long-Short Term Memory*)
MEMD: Descomposición modal empírica multivariante (*Multivariate Empirical Mode Decomposition*)
MODWT: Transformada wavelet discreta de máximo solapamiento (*Maximal Overlap Wavelet Transform*)
MRA: Análisis Multiresolución (*Multiresolution Analysis*)
OOBB: Caja contenedora orientada al objeto (*Object Oriented Bounding Box*)
PC: Perímetro Craneano.
PCA: Análisis de componentes principales (*Principal Component Analysis*)
PCC: Coeficiente de correlación de Pearson (*Pearson Correlation Coefficient*)
PE: Entropía de permutación (*Permutation Entropy*)
QDA: Análisis discriminante cuadrático (*Quadratic Discriminant Analisys*)

RCIU: Retardo de crecimiento intrauterino
RF: Bosque aleatorio (*Random Forest)*
SSVEP: Potenciales de estado estable evocados visualmente (*Steady-State Visual Evoked Potential*)
STFT: Transformada de Fourier de tiempo corto (*Short Term Fourier Transform*)
SVM: Máquina de vectores de soporte (*Support Vectot Machine*)
VRML: Lenguaje de modelado de realidad virtual (*Virtual Relity Modeling Language*)

Contenido

1. Introducción

1.1. Descripción del problema y motivación

Recientemente se ha podido observar un crecimiento explosivo de la cantidad de datos disponible en muchos ámbitos, gracias a los avances que se han venido produciendo en los métodos y dispositivos utilizados para su adquisición. Sin embargo, estos avances generan nuevos problemas, debido al aumento de complejidad asociado a este proceso. Por ejemplo, el gran volumen de datos disponible en la mayoría de los campos de aplicación dificulta su correcta compresión, así como su manejo por parte de los expertos del dominio, porque determinados aspectos críticos presentes entre los datos adquiridos pueden quedar sepultados entre otros que sean irrelevantes. Además, los datos masivos a menudo provocan que muchos métodos de análisis por computador tradicionales no funcionen de forma correcta debido a limitaciones de rendimiento.

Otro problema que dificulta el análisis y comprensión de los datos es su heterogeneidad, viniendo asociados a veces a aspectos temporales y espaciales, y pudiendo presentar tipos y patrones diversos, así como complicadas interrelaciones [1]. Para lidiar con estas nuevas problemáticas, las técnicas de análisis automático y visualización han demostrado ser muy útiles [2], [3]. Concretamente la visualización de datos se basa en el hecho de que nuestro sistema visual es especialmente hábil para detectar patrones en todo lo que vemos; en consecuencia, las técnicas de análisis visual aprovechan esta capacidad para facilitar la comprensión de los datos [4]. Por otro lado, el análisis automático, usado por ejemplo para realizar una clasificación automática de datos en categorías, permite obtener cierta información concreta de los datos de forma inmediata por parte de los expertos. Además, en ocasiones, lo aprendido al analizar unos datos de un tipo determinado ayuda porque también puede aplicarse para automatizar y optimizar el análisis de un nuevo conjunto de datos similares sin esfuerzo [2], [5].

En esta tesis se pretende desarrollar herramientas de visualización y análisis que puedan trabajar con datos de gran tamaño y de alta dimensionalidad. Además, se pretende diseñar herramientas genéricas válidas para múltiples dominios, de forma que puedan ser usadas con distintos tipos de datos con la intención de mejorar su impacto y aplicabilidad. Respecto a los tipos de datos considerados, se ha prestado especial atención a datos en los que los aspectos morfológicos o temporales resultan determinantes para su estudio.

1.1.1. Contexto de los datos: ámbito de la salud

Debido a que la tesis se ha enmarcado dentro de los trabajos del grupo de investigación VG-LAB [6] , que participa en un número de proyectos de investigación como el "*Human Brain Project*" (HBP) [7], la validación y las pruebas de las herramientas se han centrado en el ámbito de la salud. En particular, el objetivo del proyecto HBP es el estudio y compresión del cerebro humano [8]. Para ello, se está estudiando el cerebro a distintas escalas: desde el nivel celular e incluso de ultraescala, que analiza los datos desde la perspectiva de morfología neuronal estudiando la estructura de las neuronas, y siendo comúnmente utilizada para la identificación y la

clasificación de las mismas [9], hasta una perspectiva de mayor nivel de abstracción, que puede incluir simulaciones computacionales o estudios funcionales de zonas completas del cerebro. Los trabajos realizados en el HBP influyen necesariamente en el trabajo realizado en esta tesis; sin embargo, se ha intentado seguir un planteamiento genérico que pueda aislar los datos concretos de los que se dispone de su ámbito de aplicación, dentro de los condicionantes asociados al tipo de datos empleado.

A continuación, se describirá la problemática particular estudiada dentro del ámbito de este trabajo. Por una parte, se presentan algunos problemas relacionados con datos en los que la información espacial es importante, como en aquellos asociados a la morfología neuronal. Por otra parte, se han considerado también datos de tipo funcional, caracterizados por la relevancia de su dimensión temporal. Para ello, en esta tesis se han considerado también datos provenientes de Electroencefalografía (*Electroencephalography,* EEG), que son datos temporales homogéneos (es decir, del mismo tipo y tomados a intervalos regulares). También se ha querido trabajar con datos temporales heterogéneos que son más complejos, y en este caso, los datos utilizados para validar las técnicas provienen del desarrollo de bebés prematuros pertenecientes al programa madre canguro. Hay que señalar que se ha intentado buscar datos de características diferentes, con el fin de fomentar el carácter genérico de las técnicas que se desarrollen en la tesis. Los detalles tanto del tipo de datos como de los dominios a los que pertenecen se encuentran en el capítulo de estado del arte.

1.1.1.1. Análisis y visualización 3D de datos morfológicos.

Dentro del ámbito de la salud existen multitud de ejemplos en los que la información morfológica presente en los datos resulta esencial. En esta tesis se ha trabajado con datos del ámbito de la neurociencia, que presentan unas características peculiares, como la posibilidad de ser considerados como estructuras de tipo filiforme, así como la relevancia de sus patrones de arborización. Así, la estructura de las neuronas o sus redes a nivel micro y mesoescala se puede describir mediante conjuntos de elementos cuyo grosor varía de forma relativamente lenta con respecto a su longitud, dentro de los árboles axonal y dendrítico, presentando conexiones entre ellos y pudiendo asemejarse a estructuras en forma de árbol [10]. Esta misma estructura se puede encontrar, por ejemplo, en el sistema cardiovascular [11].

Desde el punto de vista de análisis y procesamiento, este tipo de datos normalmente puede verse representado de dos maneras diferentes: por un lado, con una representación volumétrica basada en mallas 3D, o, por el contrario, por una representación más abstracta basada en polilíneas, que únicamente definen las trayectorias de las estructuras, pero no su forma de manera precisa.

En este trabajo se disponía principalmente de datos de estructuras filiformes representados por medio de mallas 3D, lo que tiene una serie de ventajas e inconvenientes. Por un lado, las ventajas de este tipo de datos es que representan muy bien la superficie de la estructura filiforme obteniendo gran nivel de detalle. Por el contrario, tienen la gran desventaja de que no disponen de forma explícita de su estructura, no incluyendo información sobre las ramificaciones presentes en los datos, lo que hace muy complejo la realización de distintos tipos de análisis dependientes de esta información. Por lo tanto, para este trabajo es necesario disponer de datos con una jerarquía definida que faciliten la realización de las tareas de análisis y visualización posteriores. En este contexto, con jerarquía nos referimos a información sobre las relaciones padre-hijo entre distintas polilíneas a la hora de bifurcarse (por ejemplo, sea una polilínea A que se bifurca en dos polilíneas B y C, la polilínea A sería progenitora de las polilíneas B y C). Por otro

lado, es necesario tener en cuenta que ambas representaciones facilitan y poseen mejores características para realizar distintos tipos de análisis, por lo que, disponer de ambas sería lo ideal.

Por otra parte, aunque la visualización de estas estructuras representadas por medio de mallas 3D se encuentra prácticamente resuelta, no es el caso de las visualizaciones que utilizan la representación basada en polilíneas, que, aunque son muy útiles para comprender las trayectorias de las distintas ramificaciones, pueden omitir información importante, como por ejemplo su grosor. Para solucionar esta limitación, algunas herramientas permiten visualizar la estructura filiforme con representación basada en polilíneas de manera volumétrica, lo que permite apreciar de forma mucho más completa la estructura. Sin embargo, muchas de estas herramientas presentan algunas limitaciones a la hora de generar una malla 3D correcta. De este modo algunas herramientas generan mallas inconexas, que presentan agujeros, o superficies que no pueden existir en el mundo real, (es decir mallas de tipo *non-manifold*, lo que hace imposible utilizarlas para propósitos de análisis. Por el contrario, las mallas conexas y correctas de tipo *manifold* [12], [13] son aptas para ambas tareas (análisis y visualización).

1.1.1.2. Análisis y visualización de datos con carácter temporal

En el ámbito de la salud y la neurociencia, muchas veces se cuenta con datos cuya naturaleza temporal es esencial para los trabajos posteriores de análisis, lo que supone una serie de desafíos específicos. Algunos ejemplos de este tipo de datos pueden ser los datos provenientes de electroencefalograma (EEG), de electrocardiograma (ECG) o los datos cerebrales extraídos con la ayuda de electrodos diseñados específicamente para ser utilizados en interfaces cerebro-ordenador (*Brain Computer Interfaces,* BCI).

En los últimos tiempos se ha avanzado en diversas tareas de análisis. Así, se pueden mencionar los resultados obtenidos en la clasificación automática de este tipo de datos, demostrándose que puede ser muy útil para diferentes propósitos. Por ejemplo, en el caso de EEG, se han obtenido buenos resultados detectando enfermedades como epilepsia o Alzheimer [14]–[17], así como también ha resultado ser útil para determinar el nivel de conciencia de pacientes en estado vegetativo [18]. Por su parte, en el caso de ECG se ha demostrado que la clasificación puede utilizarse en diferentes ámbitos, como en la detección de infartos [19].

El análisis de datos temporales generado por múltiples elementos de adquisición se puede enfocar de dos maneras diferenciadas: considerando los datos como series temporales univariantes, donde cada componente de la señal se trata de manera aislada, sin tener en cuenta su relación con el resto de componentes, o considerándolos como una serie temporal multivariante, donde no solamente se tiene en cuenta cada componente de la señal de manera individual, sino también cómo se relaciona con el resto de los componentes que componen la señal.

En general, este enfoque multivariante es mucho más potente y útil que el enfoque univariante, debido a que permite estudiar las relaciones entre los distintos componentes que presenta la señal, obteniendo, por tanto, una información más completa. Sin embargo, este enfoque multivariante conlleva una serie de retos específicos que no están presentes en el enfoque univariante, por lo que esta ganancia en la información capturada tiene un coste asociado en cuanto a la complejidad y la dificultad del diseño de métodos de análisis. En esta tesis se ha trabajado en el diseño de un clasificador automático de series temporales basada en un enfoque

multivariante, de forma que pueda ser útil para arrojar pistas sobre posibles hipótesis (debido a que es completamente explicable e interpretable).

Aunque se ha avanzado mucho en la clasificación y análisis automático de señales temporales, hay ocasiones en las que resulta más apropiado realizar las tareas de análisis desde un planteamiento exploratorio o interactivo, que permite la realización de tareas de análisis también sobre datos temporales no necesariamente homogéneos. Así, en muchos ámbitos, es habitual contar con datos heterogéneos que incluyen distintos tipos de datos (categóricos, ordinales, numéricos, etc.). En estos casos, sigue siendo necesario proporcionar herramientas a los expertos en el dominio (que no tienen por qué ser expertos en Big Data, ni expertos en estadística o informática) que les ayuden a poder comprenderlos, visualizarlos y analizarlos. En esta tesis, como se describirá en el siguiente apartado de objetivos, también se busca proporcionar herramientas que permitan explorar grandes conjuntos de datos temporales heterogéneos, para que los expertos de cada dominio puedan contrastar posibles hipótesis de forma rápida, sencilla y visual.

1.2. Hipótesis y Objetivos.

En esta sección se establece la hipótesis de la tesis, así como los objetivos para su comprobación.

La hipótesis de la tesis es: **Es posible desarrollar herramientas de visualización y análisis que puedan mejorar la comprensión de grandes conjuntos de datos complejos, en particular en aquellos casos en los que la naturaleza espacial y temporal de los datos es fundamental. Igualmente, es posible diseñar estas técnicas siguiendo un enfoque genérico, de forma que puedan ser utilizadas con diferentes tipos de datos procedentes de distintos dominios**.

Por lo tanto, el objetivo general de la tesis es **desarrollar nuevas técnicas de visualización y análisis genéricas que sean capaces de adaptarse a datos de distintos dominios**. Este objetivo a su vez se divide en dos 2 subobjetivos en función de los tipos de datos con los que se trabaja:

- **Subobjetivo 1**: Diseñar una herramienta que facilite el análisis y posterior visualización de datos de tipo morfológico filiformes sin estructura. La herramienta deberá poder recibir datos de mallas sin estructura o información que las relacione entre sí y procesarlas para generar un nuevo tipo de datos que sí incluya una estructura, así como información jerárquica para poder ser utilizados en otras herramientas.
- **Subobjetivo 2**: Diseñar herramientas genéricas de análisis y visualización de datos multivariables de carácter temporal. En concreto se comenzará clasificando datos temporales multivariables homogéneos, para continuar realizando una herramienta más versátil que permita la visualización y el análisis exploratorio de datos temporales multivariables que además puedan ser heterogéneos.

1.3. Metodología.

Para la resolución de los problemas planteados en este trabajo, así como para la consecución de los objetivos descritos, se ha llevado a cabo un primer paso de familiarización con los conceptos del ámbito de las estructuras filiformes y de las series temporales.

Seguidamente se llevó a cabo un estudio de la literatura científica relacionada con las estructuras filiformes. El primer paso, consistió en la búsqueda de algún método para conseguir información estructurada a partir de la información no estructurada almacenada en la representación en mallas 3D, para facilitar su posterior análisis y visualización. Sin embargo, no se encontró ningún método para este propósito, haciéndose necesario desarrollar un método

propio. Respecto a los visualizadores de estructuras filiformes se encontraron varias propuestas, pero se detectaron problemas o posibles puntos de mejora que después se abordaron en el transcurso de la tesis.

Respecto al análisis y visualización de series temporales, en el estudio de la literatura relacionada con las series temporales univariantes se observa que ya existe mucho trabajo realizado. Sin embargo, el campo de las series multivariantes estaba teniendo avances significativos en el campo de la clasificación, en el que se podían realizar aportaciones relevantes.

Como se ha comentado anteriormente en la introducción, los principales objetivos de la tesis se centran en el desarrollo de técnicas genéricas de análisis y visualización. Sin embargo, para conseguir estos objetivos es necesario validar las herramientas en algún campo de estudio concreto con ayuda de expertos en ese dominio. En nuestro caso, los dominios escogidos han sido morfología neuronal, EEG y el desarrollo de bebes prematuros, siendo necesario realizar un proceso de aprendizaje de los conceptos básicos para poder comunicarnos con los expertos en un lenguaje común y poder atender a sus impresiones sobre las herramientas.

Respecto al proceso de desarrollo en sí mismo, se ha basado en una colaboración cercana con los expertos de los dominios de validación mostrándoles prototipos incrementales. De esta forma han podido testearse de forma continuada las herramientas desarrolladas para valorar tanto su utilidad como su facilidad de uso o posibles puntos de mejora. Además, esta realimentación por parte de los expertos también ha ayudado en gran medida a detectar posibles nuevas funciones para las herramientas que resulten de interés en el dominio de aplicación.

1.4. Contribuciones.

Como se ha comentado anteriormente, todas las herramientas desarrolladas, aunque genéricas para ciertos tipos de datos y no restringidas a un dominio concreto, han sido aplicadas a algún dominio dentro del ámbito de la medicina para comprobar su funcionamiento y utilidad. Respecto a las herramientas pertenecientes al subjetivo 1, han sido validadas con datos de morfología neuronal, aunque pueden aplicarse a todo tipo de datos filiformes, como por ejemplo para estructuras anatómicas vasculares. Por su parte, las herramientas del subjetivo 2 han sido validadas con datos de EEG, ECG y BCI para la herramienta de análisis, y con datos de ydesarrollo de bebés prematuros la herramienta de visualización.

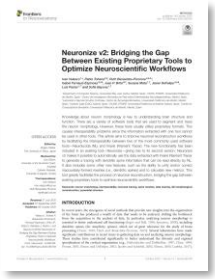
Respecto a las contribuciones aportadas al lograr los objeticos descritos anteriormente podríamos destacar:

1. Se ha desarrollado una herramienta que es capaz de obtener, partiendo de datos sin información jerárquica de estructuras filiformes, datos estructurados jerárquicamente con los que es más sencillo realizar posteriormente análisis.
2. Gracias a la herramienta del punto anterior, se ha logrado una mejora de interoperabilidad entre dos herramientas muy utilizadas en el dominio neurocientífico.
3. Se ha desarrollado también una herramienta de visualización que permite visualizar en 3D y con una malla correcta (de tipo *manifold*) estructuras filiformes representadas en forma de trazado. De esta forma, los expertos pueden comprender mejor la estructura filiforme visualizada gracias a poder ver su volumen.
4. Igualmente, se ha creado una herramienta de clasificación automática para datos temporales basada en un enfoque multivariante que consigue capturar las relaciones

entre los distintos componentes de la señal. Esta herramienta tiene la ventaja de que los resultados son completamente explicables e interpretables, permitiendo así la formulación de nuevas hipótesis.

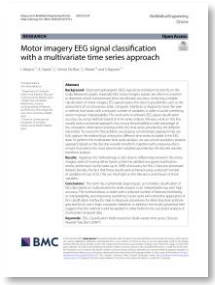
5. Se ha propuesto una librería que implementa este método que facilita que expertos de distintos dominios con sus respectivos datos puedan utilizar el método de forma sencilla, sin necesidad de tener amplios conocimientos matemáticos, estadísticos, ni informáticos.
6. Se ha generado también una herramienta de visualización de datos heterogéneos de carácter temporal, así como de procesos longitudinales que involucren un gran número de variables temporales, con un fuerte enfoque en la creación de grupos. La herramienta se ha diseñado pensando en la integración con otras herramientas, de tal forma que sigue un esquema modular y un enfoque reactivo que permite una alta interactividad entre las herramientas conectadas.

Cabe destacar que las contribuciones 1, 2 y 3 han dado como resultado el artículo de investigación:



Velasco, I., Toharia, P., Benavides-Piccione, R., Fernaud-Espinosa, I., Brito, J. P., Mata, S., DeFelipe, J., Pastor, L., & Bayona, S. (2020). Neuronize v2: Bridging the Gap Between Existing Proprietary Tools to Optimize Neuroscientific Workflows. *Frontiers in Neuroanatomy*, *14*. https://doi.org/10.3389/FNANA.2020.585793/FULL

A su vez, que la contribución 4 ha dado como resultado el siguiente articulo:



Velasco, I., Sipols, A., de Blas, C. S., Pastor, L., & Bayona, S. (2023). Motor imagery EEG signal classification with a multivariate time series approach. *BioMedical Engineering OnLine 2023 22:1*, *22*(1), 1–24. https://doi.org/10.1186/S12938-023-01079-X

Además, la contribución 5 ha dado como resultado un artículo en preparación que se enviará próximamente a *Journal of Statistical Software* y la contribución 6 ha resultado en otro artículo que se desea enviar a *IEEE Transactions on Visualization and Computer Graphics*.

2. Estado del Arte

En esta sección se presenta una revisión del estado actual de cada una de las problemáticas que se han comentado en la sección de introducción que aborda esta tesis.

La primera de estas problemáticas es la necesidad de generar una representación estructurada y jerarquizada, partiendo de datos morfológicos disponibles inconexos que presentan un gran nivel de detalle en cuanto a la morfología pero que no contienen información jerárquica al estar basados en representación de mallas 3D independientes. Además, también se hará un pequeño repaso a las herramientas utilizadas para obtener datos morfológicos, tanto aquellas enfocadas a conseguir representaciones 3D como aquellas orientadas a mostrar la morfología mediante polilíneas jerárquicas. Se explicará también, de cara a los expertos, las consecuencias que conllevan la existencia de esta dualidad de representaciones. Por último, se verán en detalle las herramientas de visualización actuales de información morfológica basadas en una representación de polilíneas.

En segundo lugar, se tratará el estado actual de los algoritmos de clasificación de series temporales, diferenciándolos por el enfoque adoptado: Series temporales univariantes, *deep learning*, teoría de la señal, etc. Por último, se hará un repaso de las diferentes herramientas de visualización de este tipo de datos de carácter temporal.

2.1. Representación y visualización de datos morfológicos filiformes

Este trabajo se centra en la visualización y análisis de datos morfológicos filiformes. Las estructuras filiformes tienen como característica principal su forma delgada y alargada, asemejándose así a hilos, pudiendo observarse en multitud de campos de la medicina como pueden ser la morfología neuronal [20] o el sistema vascular [21]. Aunque estas estructuras no tienen por qué tener bifurcaciones para ser consideradas como tal, las estructuras filiformes más interesantes suelen tener cierto grado de ramificación haciéndolas más complejas. Además, estas estructuras filiformes ramificadas pueden presentar forma de árbol o de grafo.

Respecto a la forma de representar estas estructuras, describiremos dos formas predominantes. Por un lado, existe una representación basada en una serie de mallas 3D de forma que se pueden ajustar para obtener una excelente precisión en cuanto a la forma que presentan, pero con la desventaja de que, al tratarse de mallas independientes entre sí, no se dispone de información sobre su conectividad o sobre la jerarquía entre unos elementos y otros (considerando, por ejemplo, a la malla del soma como la malla primera a partir de la cual parte el resto, se desconoce qué mallas cuelgan de qué otras). Por otro lado, la representación basada en polilíneas únicamente tiene en cuenta las trayectorias de las estructuras filiformes y tienen la ventaja de que contienen información detallada sobre su conectividad y jerarquía (se sabe qué de qué polilínea parte cada una de las polilíneas), además de ser archivos muy ligeros. Sin embargo, tienen el inconveniente de que incluyen información limitada sobre la forma y volumen. Como se puede intuir, cada una de estas representaciones permite realizar diferentes tipos de análisis que no es posible realizar con la otra representación ya que contienen distinta

información. Esta limitación en los análisis que se pueden llevar a cabo con cada una de las representaciones provoca que sea problemático el conseguir todos los análisis disponibles sobre una estructura filiforme al necesitar realizar trabajo para procesar y obtener cada una de ambas representaciones por separado, lo que suele requerir un tiempo considerable.

Como se ha comentado anteriormente, un campo que maneja datos de tipo morfológico filiforme es el dominio de la morfología neuronal, que ha sido seleccionado como dominio de aplicación. En este campo se manejan predominantemente dos tipos de representaciones: por un lado, la representación basada en mallas 3D de las distintas partes de los elementos neuronales, y, por otro lado, la representación basada en polilíneas conexas que es conocida como trazado neuronal. Dentro de este campo la representación más utilizada es la de trazado neuronal, sin embargo, utilizar la representación de mallas 3D ofrece más precisión a los expertos en cuanto a los detalles de la forma. Entre las herramientas que trabajan con la representación de tipo trazado, una de las más utilizadas es Neurolucida™ (NL; MicroBrightfield, VT, USA), un software comercial, propietario y muy cerrado. Sin embargo, también existen multitud de herramientas de software libre que trabajan con este tipo de representación, permitiendo generar estos trazados neuronales a partir de las imágenes en bruto obtenidas usando microscopios. A continuación, destacamos algunos ejemplos de estas aplicaciones. Snake [22] basa su enfoque de extracción en la detección de unos puntos semilla que luego evolucionan siguiendo unas fuerzas de deformación. La aplicacción APP2 [23] en primer lugar realiza una "*over-reconstruction*", es decir, selecciona en las imágenes de microscopía todos los pixeles con señal, para posteriormente realizar una poda de estos. Por su parte, flNeuronTool [24] parte de unos puntos semilla que posteriormente son expandidos por medio de semiesferas que detectan tanto el centro de las dendritas como las bifurcaciones. SmartTracing [25] utiliza una reconstrucción inicial, que puede haberse extraído con otro método, y seguidamente utiliza *machine-learning* para reconstruir el resto de la neurona. La herramienta NeuTube [23] se centra en la reconstrucción manual apoyando al experto con una serie de vistas 2D coordinadas con una vista 3D. Rivulet [26], para obtener el trazado, utiliza un algoritmo propio basado en un *back-tracking* iterativo. TreMap [27] realiza una segmentación manual de la neurona en varias vistas 2D que luego son ensambladas automáticamente para realizar la reconstrucción 3D. La herramienta NeuroGPS-Tree [28], al contrario que el resto, se centra en reconstruir poblaciones completas de neuronas en lugar de una única neurona. Ensemble Neuron Tracer [29] ofrece construir una serie de trazados con métodos automáticos, para seguidamente, combinar sus resultados, obteniendo así mejor precisión por medio de *machine-learning.* Por su parte, ShuTu [30] lleva a cabo una segmentación semiautomática, donde se realiza una primera aproximación de forma automática que luego el usuario puede refinar.

Respecto a las herramientas que trabajan con representaciones basadas en mallas 3D, quizá la más popular es Imaris™ (BitplaneAG, Zurich, Switzerland) que, al igual que Neurolucida™ es un software comercial, propietario y cerrado. La cantidad de herramientas disponibles que trabajen morfologías neuronales directamente con mallas 3D es bastante más limitada, por lo que suele usarse software genérico de reconstrucción de imágenes. Un ejemplo es el uso de ImageJ e ImageJ2 [31], [32], una herramienta que permite la segmentación y extracción de mallas a partir de *stacks* de imágenes genéricas de forma manual. Por otra parte, Fiji [33] es una versión de ImageJ centrada en la segmentación de imágenes biológicas de forma manual, aunque puede utilizarse con el plugin TrakEM2 [34] para obtener segmentaciones semiautomáticas.

Por tanto, dos de las herramientas más utilizadas en el ámbito (Neurolucida™ e Imaris™) están ligadas a un tipo diferente de representación. Esta diferencia en las representaciones que

manejan cada una de las herramientas provoca que, si los usuarios quieren disponer de todos los tipos de análisis posibles (que son distintos en ambas herramientas debido a la distinta naturaleza de las representaciones que manejan) deban realizar el proceso completo de obtención y refinamiento de la neurona en sus dos representaciones, lo que es un proceso muy costoso en tiempo y, además, necesita de personal experto.

En cuanto a la visualización de las neuronas, existen algunas herramientas que son capaces de visualizar las representaciones de tipo trazado de forma volumétrica generando una malla 3D a partir de la información limitada contenida en la representación del trazado. Una de las primeras herramientas en realizar esta tarea fue *NeuroConstruct* [35]. Sin embargo, la malla 3D generada por esta herramienta tenía una serie de limitaciones importantes: La primera de ellas es que la malla generada está conformada por una serie de cilindros rectos sin conectar, lo que no se aproxima a una neurona real donde el radio de las dendritas cambia de forma continua. Además, las dendritas en la realidad presentan curvas suaves que no se representaban correctamente. Una limitación importante es que la malla generada es solo útil para propósitos de visualización, ya que está conformada por mallas inconexas que encima se entrecruzan entre sí (Ilustración 1A). *Neuronize* [36] (Ilustración 1B) intenta paliar los problemas de *NeuroConstruct* generando un soma más realista partiendo de una esfera que seguidamente se deforma por medio de un sistema masa-muelle, donde las dendritas tiran del soma deformándolo y consiguiendo una representación más orgánica. Sin embargo, parte siempre de una esfera y no hace uso de la información de múltiples contornos que describe el soma de forma más detallada. En *Neuronize*, la malla generada para representar la neurona es completamente constante, respetando los grosores cambiantes de las neuritas, y las bifurcaciones están integradas dentro de la malla, de tal forma que toda la neurona forma parte de la misma malla. Por otra parte, también soporta la representación de espinas dendríticas, aunque de una forma muy básica, estas espinas se colocan en ubicaciones obtenidas en base a una función de distribución con una orientación aleatoria y sus geometrías provienen de unos modelos predefinidos. *NeuroMorphoVis* [37] (Ilustración 1C) sigue un enfoque muy similar a *Neuronize,* consiguiendo generar toda la neurona en una única malla y consiguiendo representaciones realistas del soma, aunque sin aprovechar la información de múltiples contornos. Además, tiene herramientas de reparado de trazados y soporta múltiples tipos de archivo de entrada, aunque tiene la gran limitación de no soportar la visualización de espinas. *NeuroTessMesh* [38] (Ilustración 1D) sigue sin aprovechar la información avanzada del soma de múltiples contornos, aunque mejora el modelo físico para obtener una aproximación por medio del método de elementos finitos (*Finite Element Method*, FEM). Tiene la gran ventaja de que el proceso de creación de la malla 3D se hace completamente en GPU en tiempo real soportando teselación adaptativa. Esta característica permite a la herramienta, a partir de los trazados neuronales, mostrar un gran conjunto de neuronas de forma simultánea adaptando su resolución en función de las necesidades. Sin embargo, el soporte para espinas es bastante simple, no aprovechando la información relativa ni a su posición, ni a su orientación ni a su geometría, utilizando en su lugar geometría predefinida.

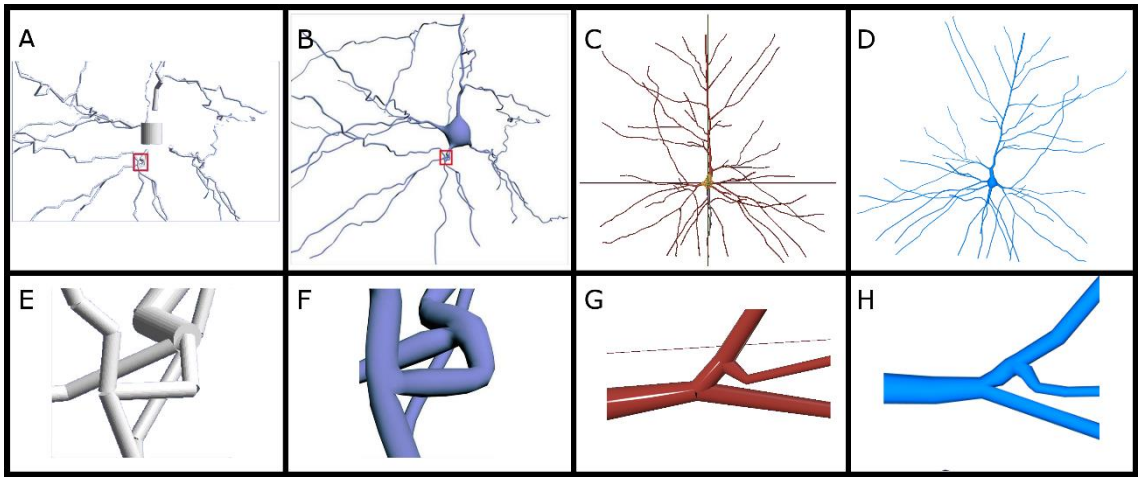


Ilustración 1: En la Ilustración puede apreciarse la representación volumétrica de neuronas a partir de una representación basada en polilíneas (también conocido como trazado neuronal en el ámbito) tanto a nivel de neurona completa, como de forma detallada. ***NeuroCostruct*** *(A, E)* ***Neuronize*** *(B, F)* ***NeuroMorphoVis*** *(C, G)* ***NeuroTessMesh*** *(D, H). Las imágenes relativas a NeuroConstruct y Neuronize han sido obtenidas de* [36]

En cierta medida esta generación de una malla 3D correcta (*manifold*) partiendo de una representación de tipo trazado puede verse como una solución al problema de interoperabilidad entre las dos representaciones. Sin embargo, la malla 3D generada a partir del trazado nunca podrá tener la precisión de una representación 3D "nativa" debido a que el formato de origen carece de esta información (por ejemplo, de la información sobre la forma detallada de una espina dendrítica concreta) y, por tanto, la malla 3D generada es solamente una aproximación.

Por tanto, tras la revisión bibliográfica se observa una carencia en cuanto a poder tener una representación jerárquica que a su vez presente un alto nivel de detalle en la geometría y la forma de los elementos neuronales. Así como la posibilidad de poder visualizar esta representación de la forma más fiel posible a la información disponible.

2.2. Análisis de series temporales.

Una serie temporal puede verse como un conjunto de datos derivados de realizar observaciones sucesivas sobre ciertas variables a lo largo del tiempo [39]. Este tipo de datos pueden observarse en multitud de campos distintos como el de las finanzas o el campo de la salud. Además, la proliferación del Internet de las cosas (*Internet of things,* IoT) ha provocado un gran aumento de este tipo de datos, al disponer de muchos sensores recogiendo datos a lo largo del tiempo [40]. Por otro lado, dentro del análisis de series temporales, las tareas que están recibiendo más atención son: la predicción (*forecasting*) de valores futuros basándose en las observaciones previas; la clasificación; la clusterización y la detección de anomalías [39].

Describiremos los principales trabajos en la literatura sobre clasificación de series temporales, más concretamente, dentro del campo de la salud. En cuanto a datos proporcionados por EEG, se han desarrollado técnicas de clasificación para: detectar convulsiones o diagnosticar epilepsia [14], [15], [41], [42]; detectar de forma automática EEGs anormales [43]–[46]; la detección del nivel de consciencia [47], el diagnóstico de Alzheimer [16] o el uso en BCI [48], [49]. Por otra parte, trabajando con datos derivados de ECG ha sido posible, por ejemplo, detectar infartos [50].

Históricamente para la realización de estos clasificadores de series temporales se ha seguido un enfoque univariante, de tal forma que se tienen en cuenta los componentes de la señal por

separado sin tener en cuenta las relaciones entre estos. Este enfoque permite conocer ciertas características de la serie temporal como la tendencia, ciclos, estacionalidades o hacer predicciones. Sin embargo, su utilidad es limitada debido a que no es capaz de aprovechar la información relativa a cómo se relacionan las distintas señales entre sí, lo que en algunos casos es muy importante [18]. Por ello, la tendencia está cambiando a utilizar un enfoque multivariante, que, aunque tiene la ventaja de tener en cuenta más información también tiene una serie de retos asociados [51].

Muchos de los métodos actuales se basan en la extracción de características. Así, se extrae una serie de características de la serie temporal original (ya sea con un enfoque univariante o multivariante), y después se utilizan estas características calculadas para alimentar a algún tipo de clasificador. Estos métodos de extracción de características pueden basarse o bien en el dominio del tiempo, o en el dominio de la frecuencia.

Respecto a las alternativas basadas en el dominio del tiempo, algunas propuestas consisten en extensiones de modelos autorregresivos (AR). Por ejemplo, en [52] a la hora de clasificar distintas tareas mentales por medio de las señales de EEG utilizan AR junto con la entropía aproximada para construir un vector de características por cada electrodo de forma independiente. Seguidamente se utilizan los vectores de características junto con un clasificador de tipo "*Extreme Learning Machine*" (ELM) para obtener la clasificación. Por otra parte, en [53] comienzan realizando un proceso de separación de señal con la intención de eliminar artefactos (*artifacts)* e interferencias (como las producidas por movimientos musculares) en los datos de EEG capturados. Seguidamente, utilizando AR extraen una serie de características que son utilizadas para clasificar junto con una red bayesiana. Otros trabajos utilizan la descomposición de Hermite para este propósito. Por ejemplo, [54] utiliza esta descomposición junto con algoritmos de optimización evolutivos para obtener los parámetros de descomposición más favorables, y seguidamente, utilizar los coeficientes de Hermite obtenidos con estos parámetros para entrenar una serie de clasificadores (ELM [55], árboles de decisión [56] , KNN [57], etc.). En su trabajo, los mejores resultados los obtienen con ELM.

Por otra parte, las alternativas basadas en el dominio de la frecuencia están basadas en la trasformada rápida de Fourier (*Fast Fourier Transform,* FFT). Por ejemplo, [58] propone utilizar FFT para obtener el rango de frecuencias deseado (para su caso entre 8 y 22Hz) y reducir el número de variables utilizando análisis de componentes principales (*Principal Component Analysis*, PCA) para, en último lugar, alimentar a un clasificador de tipo máquina de vectores de soporte (*Support Vector Machine,* SVM). Por otro lado, en [59] realizan un sistema BCI en función de potenciales de estado estable evocados visualmente (*Steady-State Visual Evoked Potential*, SSVEP) que consiste en que, cuando un humano observa un objeto parpadeando a cierta frecuencia, esta frecuencia presenta un pico en la actividad cerebral. Utilizan FFT para identificar qué frecuencias presentan mayor intensidad y así deducir a qué objeto está mirando el usuario.

Estas aproximaciones basadas en el dominio del tiempo o de la frecuencia por separado son en algunas ocasiones poco efectivas debido a la falta de información temporal o espectral. Para solucionar estos problemas se han desarrollado también los métodos híbridos (conocidos como métodos tiempo-frecuencia) que permiten realizar un análisis multiresolución en los dos dominios. Las alternativas más comunes pasan por utilizar una transformada de Fourier de tiempo corto (S*hort-Time Fourier Transform, STFT*). Por ejemplo, [60] utiliza STFT para generar una imagen partiendo de la información de la señal, utilizando dicha imagen después para alimentar una red neuronal convolucional (*Convolutional Neural Network,* CNN) para realizar la

clasificación. Tian y Liu en [61] utilizan un enfoque similar salvo que, en lugar de alimentar la red con toda la información disponible, seleccionan únicamente las bandas óptimas de frecuencia para la clasificación. Otra de las opciones es utilizar la transformada continua de ondículas (C*ontinuous Wavelet Transform,* CWT). Por ejemplo, Ieracitano et al. en [62] en un primer paso descomponen la señal utilizando CWT, y partiendo de esta señal descompuesta generan una serie de métricas de cada uno de los componentes de la señal descompuesta: media, desviación estándar, asimetría (*skewness)*, curtosis y entropía. Además, también generan características basadas en biespectros que son capaces de obtener relaciones no lineales entre las frecuencias entre distintos componentes de la señal. Sin embargo, no capturan las relaciones entre los distintos componentes de la señal. En algunos trabajos se sustituye el uso de CWT por el de la transformada discreta de ondículas (*Discrete Wavelet Transform*, DWT) al ser computacionalmente más eficiente, aunque en realidad realizan la misma operación.

Otros métodos muy utilizados y que han dado buenos resultados son los basados en *deep-learning*. Por ejemplo, Gao et al. [63] mezclan el uso de *deep-learning* con la extracción de características utilizando una red neuronal convolucional con memoria a largo-corto plazo (C*onvolutional Neural Network with Long Short-Term Memory* (CNN-LSTM) para obtener estas. Al contrario, otras propuestas son capaces de clasificar directamente las señales sin obtener características representativas, como por ejemplo el trabajo de Xie *et al.* [64], que utilizan una red CNN junto a varios *transformers* para obtener la clasificación directamente partiendo de las señales en bruto de EEG. Además, se han llevado a cabo algunos trabajos que han intentado adaptar los métodos univariantes basados en *deep-learning* a un enfoque multivariante. Por ejemplo, Karim *et al.* [65] proponen ampliar las redes LSTM-FNCs que han dado buenos resultados en análisis univariante al caso multivariante. Para ello añaden bloques de *Squeeze&Exicite* que en un primer momento generan información agregada del contexto, generando así una serie de métricas por componente de la señal. Seguidamente, esta información se utiliza en la etapa de *Exicite* que intenta capturar relaciones entre los distintos canales. Por otra parte, también existen nuevas propuestas centradas en este enfoque multivariante [66], [67]. Sin embargo, aun teniendo en cuenta los buenos resultados obtenidos y las altas tasas de precisión que aportan, los métodos basados en *deep-learning* tienen problemas inherentes al método, como el costoso proceso de entrenamiento y, sobre todo, la falta de interpretabilidad de los resultados. En esta línea de lidiar con la falta de interpretabilidad de los métodos basados en *deep-learning* existen algunos trabajos, como el de Baldán y Benítez [68], aunque todavía falta camino por recorrer.

En función de la bibliografía consultada, para la clasificación de series temporales homogéneas se plantea un enfoque híbrido y multiresolución empleando el método de DWT que permite reexpresar la serie temporal en una serie de coeficientes a distintos niveles de resolución tanto temporal, como espectral. Además, se combina con el enfoque de extracción de características para obtener tanto características univariantes (relativas únicamente a un rango de frecuencias de un componente) como multivariantes, que capturaran las relaciones entre los mismos rangos de frecuencias de distintos componentes.

2.3. Visualización de datos de carácter temporal

La visualización interactiva de datos de carácter temporal es una tarea de gran importancia y este tipo de datos se categoriza dentro de uno de los siete tipos principales de datos debido a los problemas particulares que presenta [69]. Debido a la gran importancia de este tipo de datos se han realizado muchos trabajos en esta línea como: *TimeSearcher* [70] (Ilustración 2A) permite explorar de forma visual datos cuantitativos de carácter temporal y ofrece la realización de

grupos por medio de rectángulos de selección conocidos tomo *TimeBoxes* (con manipulación directa), posibilitando, además, visualizar de forma detallada e individual las instancias seleccionadas. Sin embargo, la aplicación tiene una serie de limitaciones. Por un lado, las *TimeBoxes* tienen un comportamiento poco intuitivo, ya que en lugar de seleccionar todas las líneas que intersequen en algún punto con la *TimeBox*, selecciona únicamente las líneas que presenten todos los puntos de observación (los puntos que definen la polilínea) en el intervalo definido en el eje x dentro de la *TimeBox*. De esta forma, si una línea no tiene puntos de observación definidos en el eje x descrito por la *TimeBox* no se verá seleccionada, aunque visualmente la línea transcurra por la *TimeBox*. Además, no permite la creación de múltiples grupos de forma simultánea. *LifeLines* [71] (Ilustración 2B) está destinada a la visualización del historial médico de los pacientes a través de una visualización que permite observar el historial médico del paciente en un panel ordenado cronológicamente. En este panel pueden mostrarse tanto eventos puntuales (representados como cuadrados o círculos) así como procesos más duraderos en el tiempo (representados por una línea). Esta visualización permite a los expertos extraer relaciones entre los distintos eventos o patologías de forma sencilla al tenerlas alineadas a lo largo del tiempo. *LifeLines2* [72] (Ilustración 2C) está pensada para permitir la búsqueda, agregación y comparación con datos temporales categóricos. Para ello utilizan un método basado en una serie de pasos, el primer de ellos consiste en alinear todas las observaciones en función de la aparición de un evento (o de la aparición de varios de estos eventos). Seguidamente se realiza un filtrado en función de los eventos encontrados (además, este filtrado separa el *dataSet* en varios grupos que pueden ser comparados) y, por último, se agregan esos eventos de forma automática en función del zoom aplicado, permitiendo así tener una idea general. Además, la herramienta también presenta vistas coordinadas que permiten visualizar de forma sencilla la distribución de los eventos. *LifeFlow* [73] (Ilustración 2D) tiene como objetivo mezclar la propiedad de agregación de LifeLines2 junto con una visualización de tipo TreeMap para permitir visualizar de manera sencilla las distintas sucesiones de eventos, así como cuál es más común que otro. Para ello, utiliza una visualización en la que el eje horizontal representa el tiempo, mientras que el vertical representa el número de ocurrencias de esa sucesión de eventos en particular. Además, se utiliza una franja de cierto color para identificar el evento. De esta forma puede verse de forma sencilla la evolución entre los distintos eventos fijándose en la sucesión de colores, el tiempo de cada etapa por el espacio ocupado en posición horizontal, y lo común de esa secuencia de eventos por el alto.

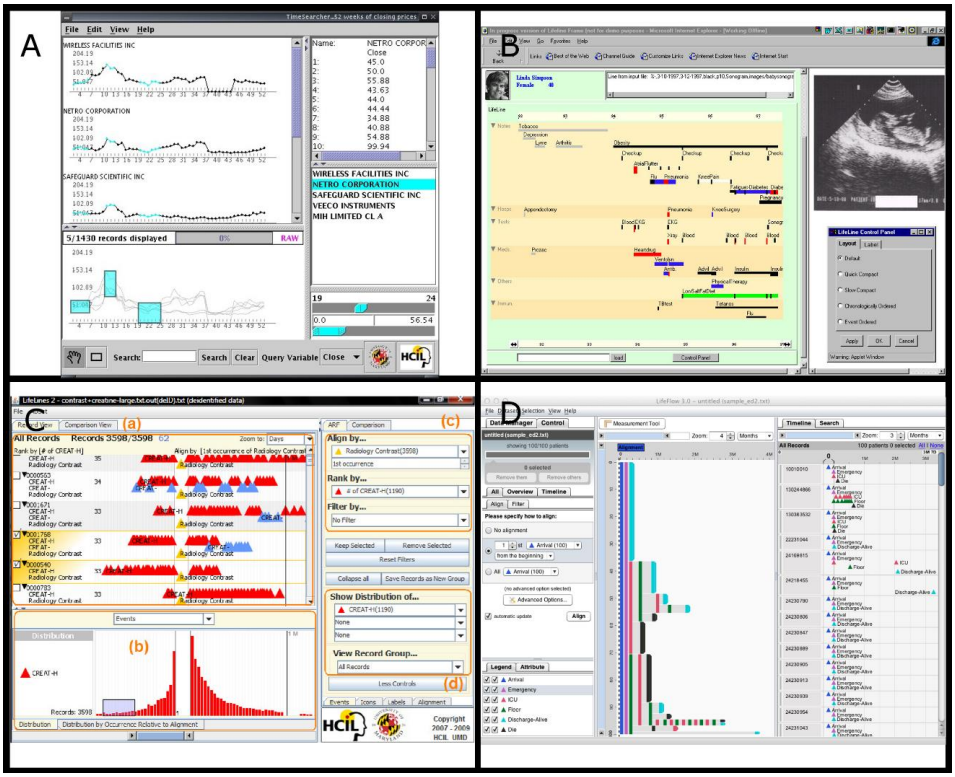


Ilustración 2: Ejemplo de las visualizaciones propuestas por TimeSearcher ***(A)****, LifeLines* ***(B)****, LifeLines2* ***(C)*** *y EventFlow* ***(D)***

Aunque estas herramientas han proporcionado grandes aportaciones, en la actualidad han caído en el desuso debido a que es muy difícil utilizarlas en entornos modernos. Esto se debe a que las herramientas se han quedado obsoletas y la reimplementación de estas es una ardua tarea [74]. Mientras, por el contrario, otras herramientas son muy utilizadas por la comunidad (como TreeMap [75], [76]) gracias a que están implementadas en tecnologías modernas como D3 [77]. Además, las herramientas modernas suelen seguir un enfoque basado en widgets, de tal forma que sea relativamente sencillo conectar varias herramientas entre sí para facilitar la creación de entornos de visualización más complejos. Por el contrario, las herramientas antiguas solían seguir un enfoque monolítico de tal forma que las herramientas estaban diseñadas para funcionar de manera aislada, sin ningún tipo de interacción con el exterior de la herramienta.

Dentro del programa canguro se han realizado varias herramientas de visualización, sin embargo, aunque algunas de ellas utilizan tecnologías modernas (como D3 o vega-lite) el enfoque adoptado para su desarrollo se parece más al enfoque monolítico. Esto ha provocado que las herramientas desarrolladas hayan quedado obsoletas rápidamente, debido a la imposibilidad de integrarlas con nuevas herramientas que complementen a las desarrolladas. Además, debido a este enfoque monolítico, modificar y actualizar las herramientas existentes se convierte en una tarea muy compleja, provocando que sea difícil modificar la herramienta para las necesidades cambiantes de los expertos. Un ejemplo de estos problemas podemos observarlos en “Analítica Canguro” [78]

Por otra parte, las tecnologías modernas siguen dos características básicas: modularidad y reactividad. Un buen ejemplo de estas características puede encontrarse en la plataforma Observable, una herramienta colaborativa basada en notebooks reactivos [79], lo que permite un rápido desarrollo y análisis de prototipos. Aunque esta herramienta se suele comparar con los *Jupyter Notebooks* [80], la principal diferencia es que cada celda en Observable es reactiva y

se ejecutan por orden topológico, de tal forma que las celdas se re-ejecutan de forma automática si se modifica el valor de alguna de sus dependencias. Además, presenta un inteligente sistema de *Inputs* reactivos y reutilizables, que permiten aprovechar de forma sencilla las aportaciones de otros usuarios al construir nuevas aplicaciones. Por estas características, entre otras, es una herramienta recomendada para la colaboración a la hora de explorar, analizar y compartir datos.

En base a la bibliografía revisada con respecto a la visualización de datos temporales y al problema que queremos afrontar de visualizar datos temporales multivariantes heterogéneos, nos aprovecharemos de la versatilidad de Observable y extenderemos la potente funcionalidad de la antigua herramienta *TimeSearcher* para proporcionar soluciones modernas, reactivas, útiles y extensibles a los expertos.

3. Análisis y visualización de datos morfológicos

Este capítulo se enfoca en aportar soluciones al primero de los subobjetivos, el desarrollo de herramientas de análisis y visualización de datos morfológicos filiformes. Cuando se parte de datos morfológicos de mallas 3D que no tienen definida una estructura o relación entre sí, se debe trabajar para obtener una representación estructurada, que contenga información de los elementos y las relaciones entre ellos, de modo que sea más apta para la realización de tareas de análisis y visualización.

Concretamente, se dispone de datos morfológicos filiformes (pertenecientes al campo de la morfología neuronal) que han sido minuciosamente revisados y procesados manualmente por los expertos del dominio para mejorar su validez. El problema es que dichos datos, pese a presentar un nivel de detalle alto en cuanto a su forma 3D, están en un formato basado en mallas 3D que no dispone de estructura ni jerarquía, lo que dificulta los análisis a nivel de toda la neurona. Por tanto, es primordial diseñar un método que permita inferir y generar esta información de estructura y jerarquía partiendo de la información espacial de mallas 3D que no la contiene.

Otro problema adicional es que, en el caso de los datos de morfología neuronal, la superficie de las mallas 3D no siempre representa correctamente la superficie de los elementos neuronales. Como se mencionó en el capítulo anterior, en la sección 2.1, los expertos utilizan la herramienta Imaris™ para extraer las espinas dendríticas con un alto nivel de detalle por medio de isosuperficies a distintos umbrales de señal. Aunque este método genera espinas muy ajustadas a la forma original de la espina (lo que es algo deseado) generan una malla individual por cada isosuperficie utilizada. Esto provoca que la espina esté representada por múltiples mallas que pueden cruzarse, intersecarse, o presentar problemas topológicos generando mallas *non-manifold* (mallas unas dentro de otras, con vértices repetidos o agujeros, por ejemplo) que no son aptas para el análisis ni para extracción de métricas puesto que podrían proporcionar medidas no fiables. Para resolver este problema se propone un método que es capaz de generar una nueva malla 3D *manifold* partiendo del conjunto de mallas generado por Imaris™, respetando al máximo posible la superficie exterior de la espina.

Por otro lado, dentro del campo de aplicación en morfología neuronal, de cara a poder visualizar una neurona completa con detalle, se detectó un problema de interoperabilidad entre formatos, que son distintos en función del método utilizado para la extracción y el refinamiento de los datos e incompatibles entre sí. Cuando se desea obtener datos muy detallados de las formas, los neurocientíficos pueden usar la herramienta Imaris™ [81] (obteniendo mallas 3D sueltas con formas muy ajustadas, pero con el problema de falta de estructura o relación entre las mallas. Cuando se desea obtener medidas en base al análisis de toda una neurona o visualizarla, es habitual utilizar la herramienta Neurolucida™ [82], que representa la neurona mediante su

trazado neuronal (una representación basada en polilíneas) que incluye información jerárquica. Sin embargo, de acuerdo con los expertos, Neurolucida™ no proporciona herramientas para obtener la forma de manera tan precisa al nivel de detalle requerido como sí lo hace Imaris™.

Esta tesis pretende ofrecer herramientas que permitan visualizar neuronas completas, y tener representaciones que incluyan tanto la información de jerarquía y estructura como la información de detalle de las formas. Por ello, el método para inferir información jerárquica de los datos que permita el análisis, planteará a su vez una solución al problema de interoperabilidad entre ambos formatos. Esto se hará manteniendo el potencial de los datos de mallas 3D refinadas conseguidas trabajando con Imaris™, a la vez que se infiere la información de estructura y jerarquía para poder obtener un trazado, todo ello sin perder el ajuste fino de la forma realizado por los neurocientíficos de forma prácticamente manual. Además, se aplicarán técnicas que generen mallas *manifold* para poder obtener medidas más fiables sobre los elementos neuronales. Todo ello conlleva la ventaja de que, una vez obtenido el trazado, este puede ser utilizado en Neurolucida™, ampliando así el tipo de análisis que se pueden realizar sobre dicha neurona.

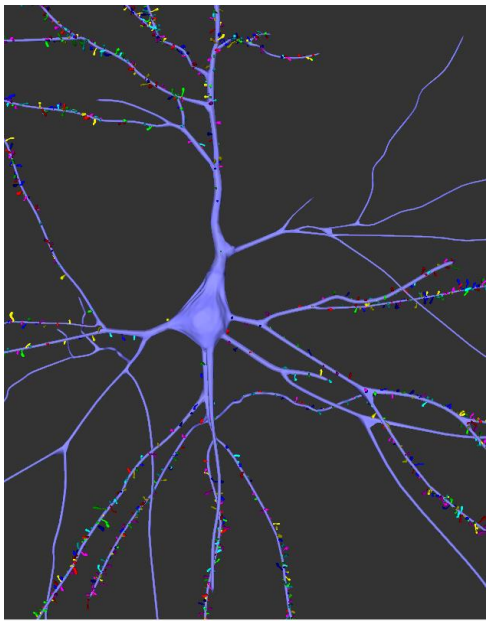


Ilustración 3: Ejemplo de la morfología de una neurona. En el centro puede apreciarse el soma. Partiendo de él, las ramificaciones o neuritas. Las pequeñas protuberancias que se muestran con distintos colores son las espinas.

A continuación, se describe otra limitación existente en la forma de representar la morfología neuronal. Para comprenderlo mejor, en la Ilustración 3 se muestra la morfología de una neurona. En el centro se aprecia el núcleo de la célula, denominado soma. Desde el soma parten ramas, llamadas neuritas. Las estructuras pequeñas sobre las neuritas marcadas con distintos colores se denominan espinas, uno de los elementos clave para que se produzcan conexiones sinápticas. En los últimos tiempos, los archivos de tipo trazado neuronal (que recogen información morfológica de la célula) han experimentado un gran avance en la información que incluyen. Concretamente se han añadido nuevas formas de representar el soma con mucha más precisión, así como información sobre la posición y orientación de las espinas dendríticas. Sin embargo, los visualizadores de estos datos morfológicos filiformes aún no son capaces de utilizar esta información para representar de forma más precisa y realista las estructuras. Por tanto, se desea que la herramienta de visualización que se desarrolle sí pueda permitir la visualización aprovechando al máximo toda la información disponible de la neurona.

En el siguiente apartado se explica el método desarrollado para facilitar el análisis de las estructuras filiformes con aplicación a morfología neuronal. Después se procede a explicar el proceso diseñado para corregir los datos sobre mallas 3D *non-manifold* y, por último, se detallan las aportaciones realizadas al visualizador de morfologías neuronales.

3.1. Generación de estructura y jerarquía.

En esta sección se abordará el método desarrollado para generar información sobre la estructura y jerarquía de datos morfológicos filiformes. El método se ha validado en el dominio de la morfología neuronal, por lo que tanto los datos de entrada como los de salida utilizaran los formatos propios de este ámbito. Sin embargo, el método en de generación puede aplicarse a otros datos espaciales sin estructura dando soporte para la lectura y escritura de sus formatos propios.

Respecto a la aplicación en morfología neuronal, el archivo generado será un trazado neuronal (en el formato de Neurolucida™), mientras que el archivo de entrada será una representación basada en mallas 3D obtenida con Imaris Filament Tracer™. En primer lugar, se muestra un resumen de los componentes de un trazado neuronal genérico. Después, se detalla el formato específico de Imaris Filament Tracer™.

3.1.1. Descripción de un trazado

Se puede encontrar una lista completa de los diferentes formatos utilizados para los trazados en [83]. Hace unos años, la mayoría de los archivos de tipo trazado no contenían información sobre las espinas dendríticas, y solían contener una información del soma muy básica, representándolo como una esfera. Sin embargo, en los últimos tiempos, nuevos formatos de trazados neuronales incluyen información más compleja del soma, que ahora se ve representado por una serie de contornos 2D, lo que permite inferir su forma 3D. Asimismo, también cuentan con información sobre la ubicación de las espinas dendríticas que anteriormente no solía recogerse.

Todos los formatos de tipo trazado incluyen una jerarquía entre los distintos puntos, de modo que cada punto sabe quién es su punto de trazado predecesor y, además, se guarda información de a qué tipo de estructura pertenece cada uno de los puntos (soma, dendrita apical, dendrita basal, espina, etc.). Así, un punto de trazado cualquiera tiene asociado, sus coordenadas 3D, el tipo de estructura que representa, un radio, que define el grosor de la estructura, y su punto de trazado predecesor que define la jerarquía.

Sin embargo, las representaciones basadas en mallas 3D (como, por ejemplo, los archivos proporcionados por Imaris Filament Tracer™) habitualmente no disponen de puntos de trazado ni de una jerarquía entre las mallas 3D que lo conforman, como se verá en la siguiente sección que trata sobre el formato de Imaris Filament Tracer™.

3.1.2. Formato de Imaris Filament Tracer™

Antes de describir cómo se genera el trazado neuronal a partir de las mallas 3D disconexas, se van a explicar las peculiaridades del formato basado en mallas 3D utilizado por la herramienta de morfología neuronal Imaris Filament Tracer™: el lenguaje de modelado de realidad virtual (*Virtual Relity Modeling Languag*e, VRML).

El formato VRML permite la inclusión de diferentes mallas 3D dentro en un mismo fichero, incluyendo así en un único archivo todas las mallas relativas a una neurona en concreto. Sin embargo, el archivo VRML generado por Imaris Filament Tracer™ no es correcto y presenta una

serie de errores, esto provoca que sea necesario realizar un pequeño preproceso de limpieza para poder visualizarlo correctamente.

Por otro lado, el archivo VRML generado por Imaris Filament Tracer™ tienen una serie de particularidades. Una de estas particularidades es que cada una de las mallas 3D ("VRML volume") tiene asociado un identificador que comienza por "FilamentSegment6" o "FilamentSegment7" en función de si se trata de mallas 3D asociadas a dendritas o espinas respectivamente. Sin embargo, hay que tener en cuenta que estos identificadores no tienen por qué ser únicos (son únicos respecto a la misma dendrita, pero no respecto a una misma neurona), por lo que se pueden encontrar dos mallas 3D con el mismo identificador.

En estos archivos VRML exportados por Imaris Filament Tracer™ los volúmenes (es decir las mallas 3D) están representados de una forma muy particular. Tanto los fragmentos de dendrita como las espinas están definidas por un conjunto de rodajas elípticas (*slices*), donde cada una de estas rodajas viene definida por 17 puntos. Esto permite extraer secciones completas de estos objetos simplemente procesando cada uno de los volúmenes dividiendo los puntos que lo conforman en grupos de 17 (ver Ilustración 4).

Ilustración 4: A la Izquierda puede observarse las secciones elípticas (cada una formada por un conjunto de 17 puntos) representando un fragmento dendrítico. Por otro lado, a la derecha se representa una espina dendrítica.

Para procesar estas estructuras se comienza leyendo las coordenadas de los puntos hasta que se han leído 17 puntos, obteniendo así una rodaja completa. De esta forma, para cada estructura, el método sigue los siguientes pasos: almacena todas las rodajas de la estructura; almacena el número total de rodajas; asigna un identificador único. Este procedimiento se aplica a cada una de las estructuras que componen el archivo leyéndolo de manera completa.

Una vez se han procesado todas las estructuras presentes en el archivo, se procede a extraer una polilínea que lo represente para utilizarla posteriormente en la construcción de la jerarquía del trazado. Para ello, por cada rodaja que compone un fragmento, se calcula su punto central y se le asigna un radio (que define el grosor de la estructura en ese punto) igual a la distancia desde el punto central calculado al punto más lejano de la rodaja. De esta forma, se obtiene una polilínea que discurre por la parte central de la estructura.

Por último, es necesario tener en cuenta que, en el archivo VRML proporcionado por Imaris™, algunas espinas están definidas por medio de dos estructuras: la geometría principal de la espina y una pequeña esfera con un cuello que corresponde con la inserción de la espina. Esta segunda geometría probablemente es añadida por la herramienta Imaris™ para prevenir que en la visualización las espinas se vean desconectadas de las dendritas (y aparezcan como colgadas en el aire), por lo tanto, al ser un añadido no extraído por los neurocientíficos dentro de la propia dendrita, se decidió ignorarlo, para que no perturbe las posteriores mediciones de las espinas.

3.1.3. Construcción de la jerarquía

Como se ha comentado anteriormente, los trazados se almacenan en ficheros estructurados donde los diferentes puntos que conforman el trazado tienen una estructura jerárquica. Es decir, aunque los puntos solo guarden información de su punto predecesor, con un procesado, para cada punto podemos saber cuál es su predecesor y su punto posterior o si, por el contrario, es el último punto de una estructura y no tiene puntos posteriores. Sin embargo, en la representación basada en mallas 3D (obviando las peculiaridades propias de Imaris Filament Tracer™) normalmente se cuenta con una malla 3D o un conjunto de mallas sin ningún tipo de información sobre las conexiones entre las distintas mallas, ni ningún tipo de relación jerárquica entre ellas.

Para lidiar con esta falta de información se ha desarrollado un método que es capaz de inferir un trazado (y toda la jerarquía necesaria) partiendo de un conjunto de mallas 3D inconexas. La jerarquía se infiere en base a la proximidad entre las distintas polilíneas que representan un fragmento de dendrita. Debido a que el archivo VRML de entrada podría no contener un soma, el método asume que, por cada dendrita, el primer fragmento encontrado es el más cercano a donde debería estar el soma. Además, hay que tener en cuenta que cada una de las dendritas es procesada de manera independiente, y que, aunque en la representación 3D la dendrita está fragmentada en multitud de partes, en la mayoría de los archivos de trazado es necesario que todos los puntos entre dos bifurcaciones pertenezcan a una única polilínea.

El método comienza centrándose en las dendritas. Para ello, añade todas las polilíneas disponibles (para una única dendrita) en una lista de polilíneas por procesar. Por convención, la primera polilínea representa la polilínea más cercana al soma y se establece como la primera polilínea de la dendrita. Seguidamente se lleva a cabo un algoritmo de búsqueda que calcula la distancia entre cada uno de los puntos de la polilínea que está siendo procesada con todos los puntos de las polilíneas sin procesar. Si alguna de estas distancias es menor que cierto umbral (conocido como "*connection Threshold*"), el algoritmo supone que los fragmentos están conectados. Una vez se han encontrado todas las conexiones de la polilínea actual, estas conexiones se clasifican y procesan en función de su tipo (existen 3 tipos principales de conexiones), y se añade la polilínea actual a la jerarquía. Seguidamente el método elige la conexión más cercana al comienzo de la polilínea (de todas las disponibles) y la escoge como la próxima polilínea para ser procesada. Nótese que al acabar de procesar esta polilínea se continuara con el resto de las conexiones. El método continúa de esta forma, procesando una polilínea a la vez, eliminándolas de la lista de polilíneas por procesar, calculando sus conexiones y, por último, añadiéndola a la jerarquía.

A continuación, se detallan los diferentes tipos de conexiones principales y cómo se resuelven:

1. El caso más simple ocurre cuando la polilínea actual no tiene ninguna conexión. Esto implica que es un fragmento terminal (el fin de una estructura). En este caso, simplemente se añade

la polilínea a la jerarquía en consecuencia con la polilínea previa a la que está conectada. Sin embargo, debido a la limitación comentada anteriormente de que todos los puntos entre bifurcaciones deben pertenecer a la misma polilínea, los puntos de la polilínea actual se añaden al final de la polilínea previa (Ilustración 5A)

2. Cuando la polilínea actual tiene únicamente una conexión posterior pueden darse dos subcasos:
   2.1. Si el primer punto de la conexión posterior corresponde con el último punto de la polilínea actual: La polilínea posterior se elimina de la lista de polilíneas por procesar y las dos polilíneas se unen en una sola. Esta nueva polilínea unida será la siguiente polilínea en ser procesada (Ilustración 5B).
   2.2. Si el primer punto de la conexión posterior corresponde a un punto intermedio de la polilínea actual: La polilínea actual se divide en dos subpolilíneas. Después, se añade a la jerarquía la primera subpolilínea (*First New Previous*), que va desde el punto inicial al punto de conexión. Seguidamente, el método continúa procesando ambas, la segunda subpolilínea (*New Current 1*) que va desde el punto de conexión hasta el final de la polilínea actual original y la polilínea de la conexión posterior (*New Current 2*), como se observa en Ilustración 5C.
3. Si la polilínea actual tiene dos conexiones posteriores y el punto inicial de estas dos conexiones corresponde al punto final de la polilínea actual: En esta situación, se trata de una bifurcación. La polilínea se va a añadir a la jerarquía a continuación de la polilínea previa y se continúa procesando las 2 polilíneas posteriores (Ilustración 5D).

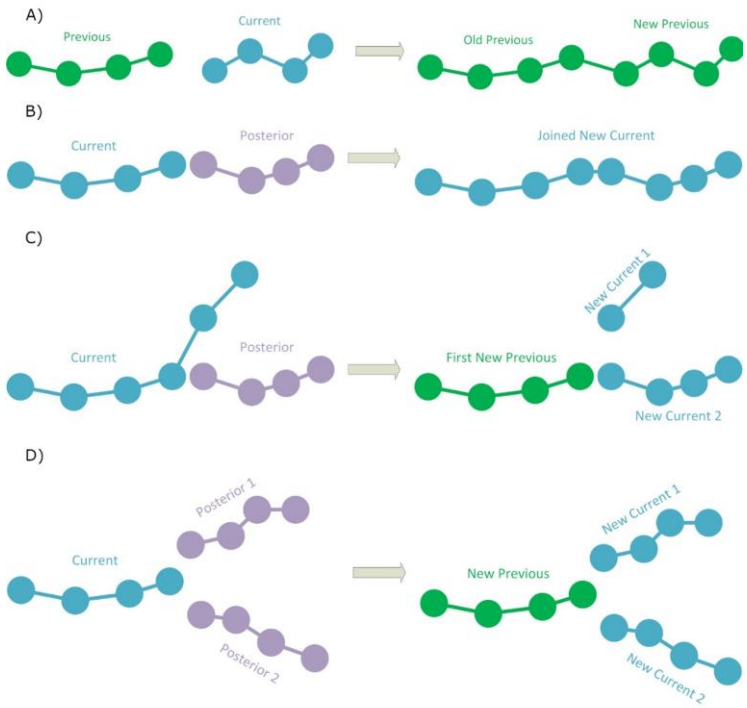


Ilustración 5: Proceso para el procesado de las polilíneas de cara a inferir una jerarquía entre ellas. (A) Caso en el que la polilínea actual no tiene conexiones posteriores, por lo que es un fragmento terminal. En este caso, el fragmento actual simplemente se añade a la jerarquía. El color verde muestra las polilíneas ya añadidas a la jerarquía, que no necesitan más procesamiento. (B) La polilínea actual tiene una conexión posterior en la que el último punto de la polilínea actual corresponde al primer punto de la conexión posterior. Las dos polilíneas se unen en una única polilínea y se procesará esta nueva polilínea (el color azul representa las polilíneas que se van a procesar). (C) La polilínea actual tiene una conexión posterior en algún punto medio. En este caso, la polilínea actual se divide en dos subpolilíneas. La primera subpolilínea se añade a la jerarquía (color verde). La segunda subpolilínea y la conexión posterior se

procesarán (color azul). (D) La polilínea actual tiene dos conexiones posteriores en su último punto. La polilínea actual se añade a la jerarquía (en color verde) y las dos conexiones posteriores se procesarán (en color azul)

Aunque estas son los distintos tipos de conexiones principales en los que se divide el algoritmo de generación de la jerarquía, existen dos casos especiales que es necesario tratar para convertirlos en uno de los tipos generales mencionados anteriormente:

1. En el caso en el que la polilínea tenga dos conexiones posteriores, donde al menos de una de estas conexiones se realice en un punto intermedio de la polilínea actual. En este caso, en primer lugar, se trata la conexión más cercana al inicio de la polilínea actual. Este punto de conexión se utiliza para dividir la polilínea actual en dos subpolilíneas: la primera de ellas discurre desde el inicio de la polilínea actual al punto de la primera conexión; la segunda subpolilínea transcurre desde el punto de conexión al final de la polilínea. Seguidamente se añade la primera de las subpolilíneas a la jerarquía y se procede a procesar la polilínea de la conexión que se está tratando y la segunda polilínea, por último, se añade la polilínea de la conexión que no se ha tratado para que sea encontrada en su momento cuando se procese la segunda subpolilínea (Ilustración 6A). De esta forma, este caso se transforma esta conexión especial a dos conexiones principales de tipo 3.
2. En el caso en el que la polilínea actual tenga dos conexiones posteriores justo en el mismo punto, es necesario aplicar un preproceso, ya que normalmente los archivos de tipo trazado no soportan más de una bifurcación en el mismo punto. Para lidiar con este problema simplemente se escoge una de las conexiones y se mueve su punto de inicio a otro punto cercano de la polilínea actual para que pueda ser tratado como en el caso anterior (Ilustración 6B).
3. En el caso en el que la polilínea actual tenga más de dos conexiones, simplemente se procede tratando cada una de las conexiones, empezando desde el inicio de la polilínea actual de acuerdo con los casos vistos anteriormente. De esta forma, la polilínea actual se irá subdividiendo en función de los tipos de conexiones encontradas.

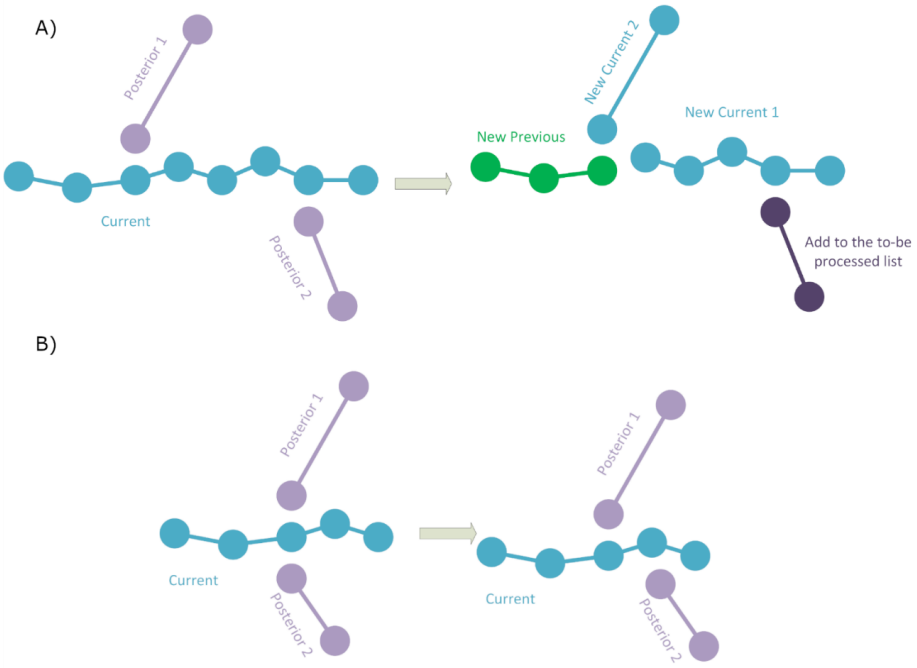


Ilustración 6: Casos especiales a tratar para construir la jerarquía: (A) La polilínea actual tiene dos conexiones posteriores en algunos puntos intermedios. La polilínea actual se divide en dos subpolilíneas, la primera subpolilínea se añade a la jerarquía (verde). Seguidamente se procesarán la segunda subpolilínea y la primera conexión posterior (azul). Observe que la segunda conexión posterior (en negro) se añade a la lista de polilíneas a procesar. (B) La polilínea actual tiene dos conexiones posteriores en el mismo punto intermedio. Aquí, una de las conexiones posteriores se modifica para cambiar su punto inicial al siguiente punto más cercano diferente en la polilínea actual.

Este método presupone que las polilíneas están construidas desde el centro de la estructura; en el caso de la morfología neuronal partiendo del soma hacia los extremos siguiendo su forma natural de árbol. Sin embargo, aunque esta sería la forma más "natural" de proceder, algunas de las polilíneas se encuentran invertidas, lo que ocasiona problemas en el algoritmo de construcción de la jerarquía, además, los formatos de tipo trazado no permiten este modo de construcción invertido. Por lo tanto, se ha añadido una etapa después de la búsqueda de conexiones, de tal forma que, si el punto de conexión inicial de una polilínea se encuentra en la segunda mitad de esta, se detecta que la polilínea se encuentra invertida y se corrige para continuar normalmente con el algoritmo.

Este proceso se repite para cada una de las dendritas de la neurona hasta que se completa todo el archivo. Sin embargo, debido a que en el archivo de entrada algunas de las mallas 3D se pueden intersecar existen puntos del trazado resultante que pueden estar repetidos o muy juntos. Para solucionar este problema se realiza un postproceso en el que los puntos del trazado que estén a una distancia menor que cierto umbral son considerados como repetidos y son eliminados.

3.1.1. Umbral de conexión

Como se puede deducir de la sección anterior, la correcta elección del umbral de conexión es clave para el correcto funcionamiento del algoritmo, debido a que es el que determina si dos polilíneas cualesquiera deben estar conectadas o no. Nótese que la elección de este parámetro no es trivial ya que valores demasiado pequeños pueden provocar que algunas de las polilíneas no se conecten. Por el contrario, un valor grande asegura la conexión de todas las polilíneas, pero puede ocasionar conexiones incorrectas y una mala gestión de las bifurcaciones.

Para evitar que los expertos tengan que lidiar directamente con este parámetro de configuración, el método busca de forma automática un valor adecuado para la mayoría de los casos. El proceso consiste en procesar el fichero partiendo de un valor de umbral de conexión pequeño e ir aumentándolo progresivamente en caso de encontrar polilíneas desconectadas. Esto tiene un impacto en el tiempo en el que se procesan los archivos, debido a que en ocasiones es necesario reprocesarlos varias veces, pero asegura (en la gran mayoría de los casos) un resultado correcto.

3.1.2. Generación del soma para el trazado

Una vez se han construido todas las dendritas con su correspondiente jerarquía es necesario añadir el soma al archivo de trazado. Este soma puede provenir de una malla 3D exportada desde Imaris™, o bien es posible que en los datos no exista ninguna información sobre el soma. En el caso de que no se disponga de información relativa al soma neuronal es necesario inferirla, para ello se define el soma como una esfera donde su centro es el baricentro de todos los puntos de inicio dendríticos y su radio es la distancia más corta entre el baricentro y estos puntos de inicio.

Finalmente, una vez se dispone de información del soma (ya sea por medio de un archivo externo o inferida a partir de las dendritas) es necesario transformar la representación 3D a una representación basada en una serie de contornos 2D que es el formato soportado por el formato de trazado (ver Ilustración 7). Nótese que, como se verá más adelante, esta primera aproximación del soma es refinada y, los contornos 2D se actualiza en consecuencia.

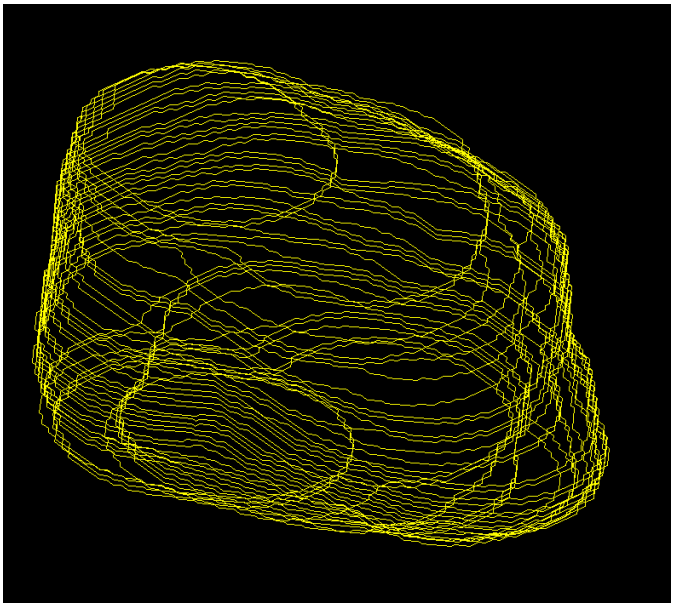


Ilustración 7: Ejemplo de soma conformado por múltiples contornos 2D.

3.1.3. Añadiendo espinas al trazado

En el momento en el que ya se dispone de la representación de tipo trazado jerárquico de la morfología neuronal, incluyendo soma y dendritas, se procede a añadir las espinas dendríticas. Estas espinas pueden encontrarse en el mismo archivo que las dendritas o en otros archivos de tipo mallas 3D, y siguen una estructura similar formada por rodajas elípticas. Alternativamente, pueden obtenerse espinas de otro archivo que contenga el esqueleto de las espinas como polilíneas.

En caso de que las espinas se encuentren en el mismo archivo, la polilínea que representa cada una de las espinas se obtiene de forma similar a como se obtienen las polilíneas que representan los fragmentos dendríticos. Por el contrario, si las espinas son proporcionadas en un archivo aparte, que contiene una polilínea representando cada espina, se supone que el primer punto de esta polilínea es el punto referente a la cabeza de la espina, mientras que el último punto referencia el punto de inserción de esta en la dendrita.

Una vez se ha obtenido toda la información de las espinas de alguno de los dos archivos soportados, estas son añadidas jerárquicamente al trazado previamente generado. Para ello, por cada espina, se calcula la distancia desde el punto de inserción de la espina (el principio de la espina) a todos los puntos del trazado generado. Si la menor distancia calculada es menor que el umbral de conexión ("Connection Threshold"), se supone que la espina se encuentra conectada a ese punto con el que tiene la menor distancia de todas y se escoge este punto como punto inicial de la espina. Nótese que este planteamiento es un poco menos preciso que añadir al trazado el punto inicial de la espina, pero las desviaciones introducidas por esto son mínimas si el trazado está suficientemente muestreado, como es el caso de los trazados generados a partir de mallas provenientes de Imaris.

Por último, aunque en la representación interna de la herramienta las espinas se almacenen de forma completa como polilíneas, en la actualidad el soporte para espinas en los formatos de trazado es bastante limitado. En concreto muchos de los formatos de trazado no soportan la representación de espinas, y los que los soportan, normalmente, únicamente admiten representar la espina como una única línea, desde un punto de inserción al punto final. Debido a esta limitación y con el fin de maximizar la compatibilidad del trazado generado, a la hora de exportar el fichero únicamente se tiene en cuenta el punto inicial y final de la polilínea obviando el resto de los puntos.

Una vez se ha terminado este paso, se dispone de una conversión total de la representación original basada en mallas 3D a una representación basada en polilíneas (que dispone de estructura y jerarquía).

3.2. Reparación y unificación de geometría

Respecto a los datos disponibles sobre las espinas dendríticas, los expertos en morfología neuronal tienen dos formas de digitalizarlas. Por un lado, pueden utilizar Filament Tracer para obtener una geometría correcta de las espinas, pero con poca precisión (este tipo es el utilizado en el generador de esqueletos) o, por otro lado, pueden obtenerlas utilizando la herramienta Imaris™. Con esta herramienta la precisión de las espinas es muy elevada, pero las espinas generadas así presentan una serie de problemas.

A la hora de digitalizar las espinas utilizando Imaris™, los expertos deben definir varis isosuperficies que acaben modelando la espina completa. Aunque esta forma de proceder arroja unos resultados muy precisos en cuanto a la forma presenta el problema de que cada una de estas isosuperficies es una malla completamente independiente del resto de mallas que conforman la espina. De esta forma, las distintas mallas que conforman el archivo no están conectadas entre sí, pudiendo estas intersecarse mutuamente, o presentar agujeros, lo que ocasiona que los análisis realizados sobre estas mallas (como por ejemplo el cálculo de su superficie o del volumen) sean incorrectos (ver Ilustración 8).

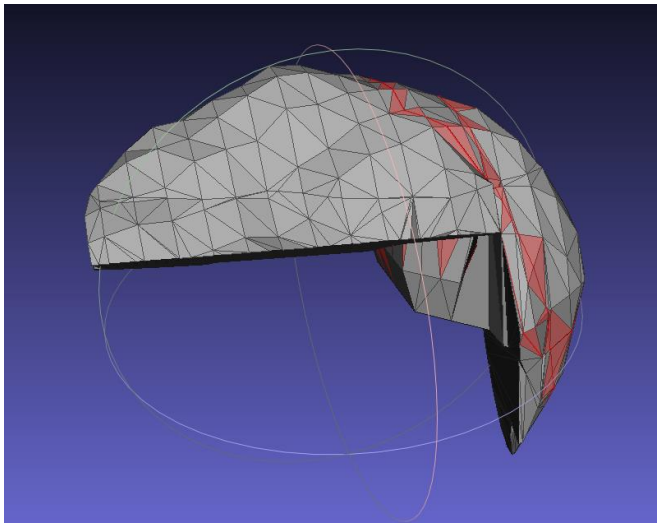


Ilustración 8: Ejemplo de una espina extraída utilizando Imaris™, las caras rojas presentan errores por múltiples motivos (caras interiores, geometrías con espesor cero, etc.…)

Para solucionar estos problemas se ha desarrollado una herramienta que es capaz de reparar estos archivos con múltiples mallas, dando como resultado una única malla correcta (*manifold*). Además, una vez reparadas, nuestra herramienta también calcula de manera automática una serie de métricas sobre las mallas corregidas como superficie, volumen, etc.

El método consiste en una adaptación del propuesto por [84] pero ha sido extendido para reparar no únicamente mallas de espinas, sino que ha sido adaptado para reparar mallas de mayor tamaño, como por ejemplo mallas de fragmentos dendríticos, de somas o, en general, cualquier tipo de malla 3D. En primer lugar, el método realiza una voxelización transformando la representación de superficie a una representación volumétrica. Seguidamente, se realiza un proceso de dilatación-erosión [85] con la intención de conectar fragmentos de malla inconexos y de cerrar posibles agujeros. Posteriormente, se realiza un suavizado gaussiano a la voxelización, eliminando así posibles picos producidos por errores, por último, se vuelve a la representación de superficie por medio de la técnica de *marching cubes* [86].

Aunque con este método se obtuvieron muy buenos resultados para mallas relativamente pequeñas (como las de las espinas, ver Ilustración 15), presentaba problemas a la hora de gestionar la memoria para mallas más grandes (como la de los fragmentos dendríticos). El principal problema radicaba en el proceso de voxelización, que voxeliza todo lo contenido en la caja contenedora (*bounding box)* definida por la malla, de tal manera que para ciertas mallas (como las de los fragmentos dendríticos) exista un número enorme de voxels, a pesar de que estén la gran mayoría vacíos (ver Ilustración 9). Para solucionar este problema, se decidió orientar las mallas con los ejes de coordenadas para reducir al máximo el número de voxels vacíos. El método escogido consiste en utilizar el algoritmo de análisis de componentes principales (*Principal Component Analysis, PCA*) para obtener los ejes con mayor dispersión y alinear estos con los ejes de coordenadas, formando así una caja contenedora alineada al objeto (*Object Oriented Boundig Box, OOBB)* que se ajuste mejor a la malla a procesar).

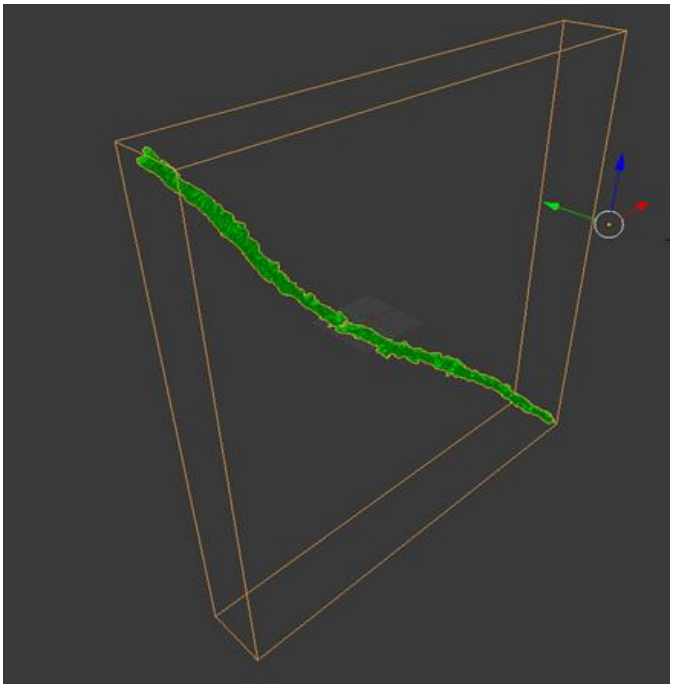


Ilustración 9: Ejemplo de un fragmento dendrítico que se encuentra colocado en una posición en la que su caja contenedora es muy grande en relación con el espacio ocupado por el fragmento dendrítico. Esto provoca que el algoritmo de reparación tenga que lidiar con una gran cantidad de voxels, aunque la mayoría de ellos estén vacíos.

Respecto al caso particular en el que se ha aplicado este método, es decir a morfología neuronal, existen una serie de dificultades a la hora de leer de forma correcta las mallas de las espinas y los fragmentos dendríticos, debido a los formatos de exportación utilizados por estas herramientas. Concretamente existen dos formatos utilizados por esta herramienta, VRML e IMX.

Aunque el formato VRML es abierto, la forma en la que está construido por parte de Imaris™ contiene gran cantidad de información innecesaria y las estructuras propias del formato no se emplean correctamente, lo que hace imposible abrir este tipo de archivo propietario con programas estándar. Para solucionar este problema se ha diseñado un algoritmo que es capaz de "limpiar" este archivo haciendo que contenga únicamente la información relevante, es decir las mallas de espinas y fragmentos dendríticos, y que siga una estructura correcta.

Por otra parte, el formato IMX (propietario de Imaris™) contiene mucha información no relacionada con la geometría de la neurona, pero, sin embargo, no contiene ninguna etiqueta (o no ha sido posible encontrarla) que permita diferenciar el tipo de objeto entre espinas y fragmentos dendríticos. Para superar este escollo, se llevó a cabo un análisis estadístico sobre los vértices que presentaban las espinas dendríticas, determinando así, a partir de este análisis, que las estructuras que tuvieran su número de vértices en un rango de [10, 10.000] serían consideradas como espinas, mientras que las estructuras con más de 10.000 vértices serían consideradas fragmentos dendríticos.

3.2.1. Herramienta de comparación de mallas.

Aunque a simple vista los resultados obtenidos por el reparador de mallas 3D parecían prometedores, y puesto que este trabajo se centra en mejorar la visualización y el análisis, se deseaba poder comprobar la correctitud del método y visualizar de forma sencilla los cambios producidos a la hora de ajustar los parámetros. Por lo tanto, se decidió realizar una herramienta

visual de comparación de mallas 3D que permitiese evaluar de una manera sencilla las diferencias entre dos mallas con la intención de comparar la malla original con la malla reparada.

La herramienta presenta dos opciones de visualización, por un lado, una primera opción que permite visualizar las dos mallas en una vista *side-by-side*, en la que se pueden comparar dos mallas cualesquiera. Por otro lado, y quizá la opción más interesante, es una vista *side-by-side* avanzada que permite comparar las diferencias entre dos mallas (pensada para comparar mallas originales y reparadas) por medio de colores y en donde las vistas están coordinadas de tal forma que, aunque el usuario mueva el punto de vista sobre una de las mallas, al estar coordinadas, siempre se estén viendo ambas mallas desde el mismo punto de vista. Un ejemplo con la comparación de las mallas de dos espinas dendríticas puede verse en la Ilustración 10.

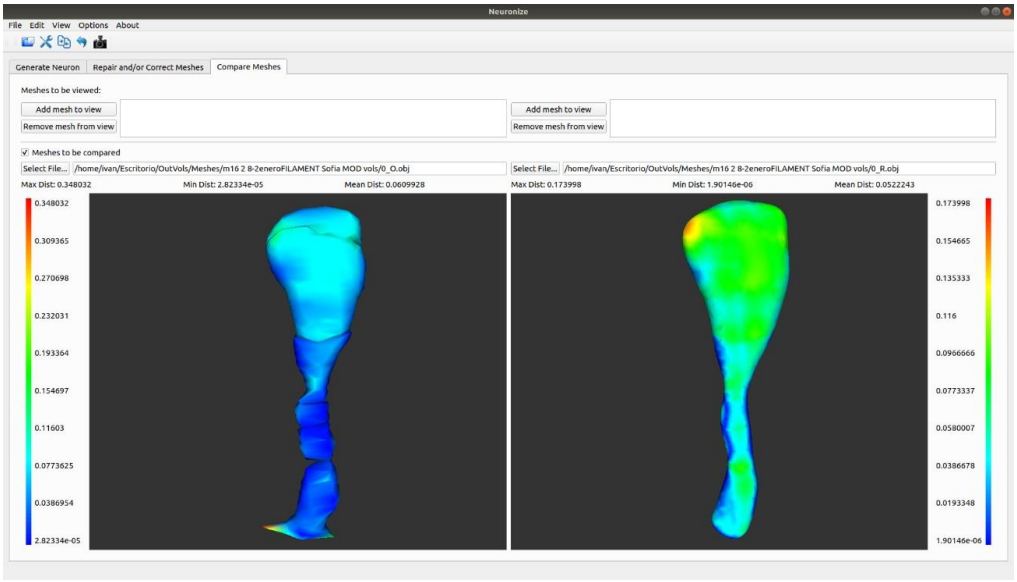


Ilustración 10: Ejemplo de la visualización diseñada para comparar las mallas originales con las reparadas aplicado a espinas dendríticas. Los colores azules denotan una discrepancia menor, mientras que los colores cálidos denotan mayor discrepancia. Es importante notar que la coloración de las dos mallas no es igual, tanto debido a que utilizan escalas de color como a que el método para el cálculo de distancias no es simétrico.

Para calcular la distancia entre las dos mallas y poder colorearla se ha utilizado la distancia de Hausdorff [87] entre las mallas. Esta distancia no es conmutativa, siendo necesario calcular la distancia de una malla a la otra y viceversa (de ahí que haya dos valores de distancia y dos escalas de colores, como se aprecia en la Ilustración 10). Una vez obtenidas las distancias, se utiliza una función de transferencia para mapear los valores de distancia a distintos colores que serán utilizados en la visualización. Además, en la propia interfaz se pueden ver una serie de métricas relevantes, como la distancia máxima, mínima y la media.

3.3. Mejoras del visualizador de datos morfológicos filiformes

En este apartado se detallarán las mejoras realizadas a un visualizador de estructuras filiformes, con aplicación a morfología neuronal, ya existente previamente [36] aprovechando, por una parte, los métodos desarrollados comentados anteriormente, así como la nueva información incluida en los archivos de tipo trazado relativos a morfología neuronal.

3.3.1. Generación del soma

En las estructuras filiformes pertenecientes al campo de la morfología neuronal, existen distintas estructuras que es necesario tratar. Una de estas estructuras es el soma, el cual se encuentra en el centro de la neurona, siendo su núcleo, y es de donde emergen las dendritas.

Tradicionalmente este soma se ha representado por medio de una esfera simple del que emergían las dendritas. Algunos trabajos han intentado mejorar esta aproximación utilizando métodos de simulación física para deformar ese soma esférico en función del grosor y distancia de las dendritas, por ejemplo, Neuronize [36] utilizaba un enfoque basado en masa-muelle mientras que NeuroTessMesh [38] utiliza una deformación basada en elementos finitos. Sin embargo, desde que se crearon estas herramientas cada vez se dispone de más información sobre la forma del soma.

En el software comercial Neurolucida™ (y en su formato ASC) se dispone de información precisa sobre la forma del soma. Esta información viene representada por una serie de contornos 2D que en su conjunto definen la forma 3D del soma. Sin embargo, para poder visualizar, y más importante analizar, este soma se necesita su representación en forma de malla 3D, para conseguirla se aplica un algoritmo de *Convex Hull* al conjunto de contornos 2D obteniendo asi una malla 3D. Si bien la malla 3D generada por el algoritmo es correcta, esta presenta muy pocos triángulos y de tamaño muy variable, lo que no es óptimo para llevar a cabo simulaciones físicas sobre la malla (como el proceso de deformado), por lo tanto, la malla 3D obtenida es procesada hasta conseguir una malla isotrópica con una mayor densidad de triángulos y con mayor uniformidad entre ellos. En la Ilustración 11 puede observarse un ejemplo de un soma conformado por múltiples contornos.

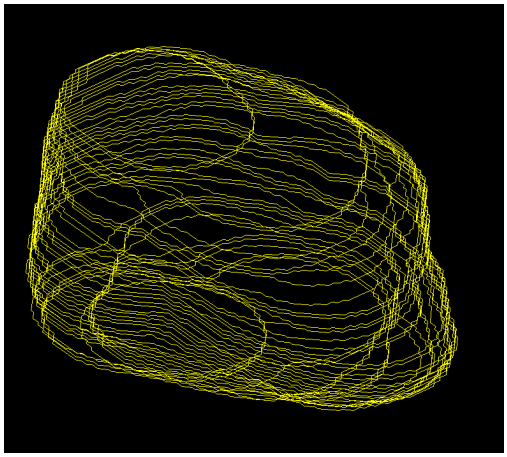


Ilustración 11: A la derecha se observa un soma definido por un conjunto de múltiples contornos 2D. A la derecha se observa la reconstrucción de una geometría representada por una malla 3D partiendo de los contornos 2D.

Por otra parte, es necesario tener en cuenta que en ocasiones el inicio de las dendritas se encuentra dentro del propio soma. Para solucionar esta inconsistencia se supone que la información correcta es la que representa el soma, por lo tanto, se eliminan todos los puntos de trazado del interior de este, considerando como nuevo inicio de la dendrita la intersección entre esta y la malla del soma.

3.3.2. Nuevos sistemas de colocación de espinas

Respecto a las espinas dendríticas se da una situación similar que, con el soma, gracias a los algoritmos desarrollados en esta tesis ahora se dispone de nueva información que puede ser utilizada para mejorar la visualización de estas estructuras. Además, los formatos de tipo trazado también han avanzado a la hora de representar estas estructuras incluyendo ahora información sobre su posición y orientación.

En la primera versión del visualizador no había disponible esta información sobre las espinas, ni relativa a su geometría ni a su posición y orientación. Por lo tanto, esta primera versión tenía

incluidas una serie de geometrías de espinas por defecto (15) que eran colocadas siguiendo una función de distribución y siguiendo una orientación aleatoria.

Ahora, gracias a la nueva información contenida en los ficheros de tipo trazado, se puede conocer la posición y orientación de cada una de las espinas, sin embargo, no es posible conocer su geometría. Aprovechando esta nueva información la herramienta es capaz de colocar las espinas siguiendo esas posiciones y orientaciones definidas en el archivo de entrada, mientras que para la geometría se utilizan espinas procedentes de una base de datos de espinas previamente extraídas de representaciones 3D.

En caso de tener información sobre la geometría de las espinas, debido a que se dispone de un fichero de representación 3D, simplemente se añaden estas a la malla previamente generada. Además, se realizan una serie de transformaciones que permiten almacenar la espina de una manera homogénea con la intención de ser almacenada en una base de datos para su posterior uso en neuronas que no contengan la geometría de las espinas.

3.4. Resultados

En esta sección se detallarán los resultados obtenidos en las distintas tareas realizadas para mejorar los procesos de análisis y visualización de datos morfológicos filiformes, aplicados al ámbito de la morfología neuronal.

3.4.1. Generación de estructura y jerarquía.

Unas de las principales limitaciones que se ha encontrado en este trabajo a la hora de realizar las tareas de análisis y visualización es que se disponía de los datos en un formato de malla 3D inconexas y sin información de jerarquía, extraídas con Imaris™, lo que dificultaba la etapa de análisis. Por lo tanto, se ha desarrollado un método que es capaz de generar la información de estructura y jerarquía partiendo del conjunto de mallas 3D inconexas. Además, para la validación de la herramienta se han escogido datos provenientes del dominio de la morfología neuronal.

Para validar el correcto funcionamiento de la herramienta se solicitó a los expertos en el campo un gran abanico de neuronas para las cuales disponían tanto de la representación basada en mallas 3D como la representación de tipo trazado neuronal. Así, se pudo comprobar que el método desarrollado generaba una representación similar por medio de la comparación visual entre el trazado generado por las herramientas y el trazado extraído manualmente por los expertos (ver Ilustración 12). Sin embargo, para algunas de estas neuronas se podían detectar ciertas inconsistencias entre el trazado generado por la herramienta y el extraído manualmente, tras consultarlo con los expertos, se llegó a la conclusión de que al ser ambos procesos de extracción manuales podían existir ciertas discrepancias. No obstante, los expertos estaban más satisfechos del trazado neuronal extraído por nuestra herramienta que el extraído manualmente, debido a que con Imaris™ pueden ser más precisos (ver Ilustración 12).

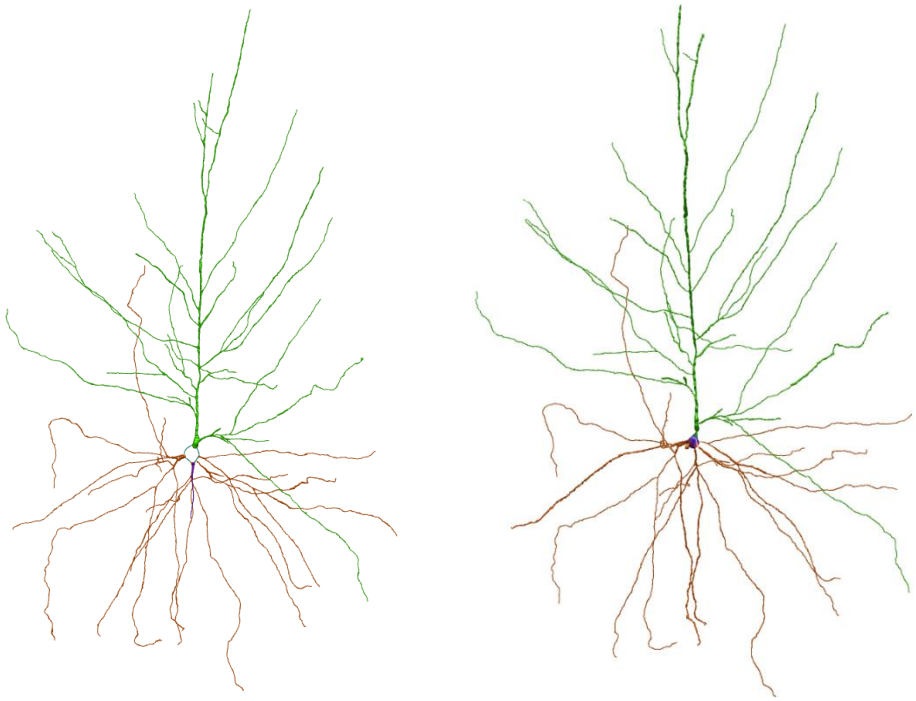


Ilustración 12: Comparativa entre el trazado neuronal generado manualmente con Neurolucida™ (Izquierda) y el trazado generado por la herramienta desarrollado partiendo del conjunto de mallas 3D inconexas generados por Imaris™. Como se puede observar en la imagen existen ciertas discrepancias en la parte superior de la neurona, debido a que ambos conjuntos de datos de prueba (tanto el trazado obtenido con Neurolucida™, como el conjunto de mallas obtenido con Imaris™) se obtienen de forma manual.

Por otro lado, el método desarrollado para transformar un soma representado por medio de una malla 3D (proveniente de Imaris™, o de algún método de generación) ha dado muy buenos resultados como se puede observar en la Ilustración 13.

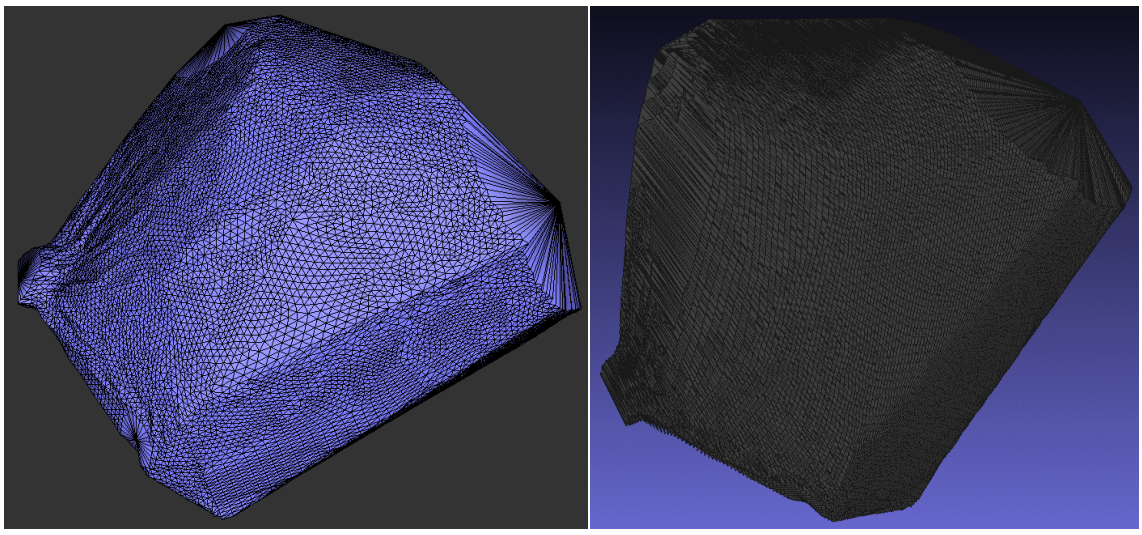


Ilustración 13: A la derecha puede observar un soma neuronal con una representación basada en malla 3D. A la derecha puede observarse el mismo soma después de haber sido convertido a un conjunto de contornos 2D, que serán almacenados en el fichero de trazado neuronal.

Además, el proceso de adición de las espinas al trazado también ha dado buenos resultados. Sin embargo, debido a la falta de neuronas de las que se dispone a la vez de la representación de mallas 3D y de un trazado neuronal con espinas (debido a que la extracción de espinas con

Neurolucida™ es muy costoso) no ha sido posible realizar una comparación como para el trazado sin espinas. Por lo tanto, ha sido necesario comparar directamente con la imagen de microscopia y con la representación basada en mallas 3D gracias a las mejoras realizadas al visualizador de estructuras morfológicas filiformes.

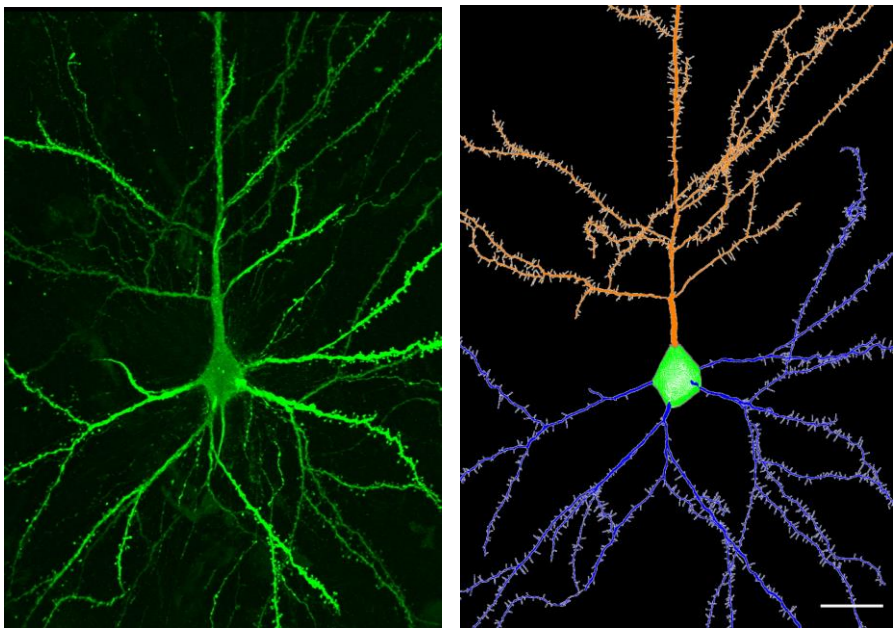


Ilustración 14: Comparativa entre la imagen de microscopia de una neurona (derecha) y el trazado con espinas generado partiendo de la representación de mallas 3D (derecha). Nótese que, las pequeñas líneas que emergen de las dendritas representan las distintas espinas dendríticas. Además, también puede observarse el soma formado por múltiples contornos 2D en el trazado neuronal (derecha).

Respecto al problema de interoperabilidad encontrado entre las herramientas Imaris™ y Neurolucida™ ha sido solucionado de manera satisfactoria. Por un lado, se ha generado una herramienta que es capaz de leer los formatos propietarios de Imaris™ (VRML, IMX), y, aprovechando el algoritmo de generación de la jerarquía, crear un archivo de tipo trazado neuronal en el formato propietario de Neurolucida™ (ASC). Además, se han añadido algunas funcionalidades de cara a la usabilidad por parte de los usuarios. La primera de estas funcionalidades es el método automático del cálculo del umbral de conexión abstrayendo de estos detalles a los usuarios, además, también se ha añadido un modo por lotes que es capaz de convertir una gran cantidad de archivos sin intervención del usuario. Por otro lado, con el objetivo de comprobar la usabilidad de la herramienta se llevaron a cabo varias pruebas con usuarios, y, gracias al *feedback* de estos la herramienta pudo mejorarse sustancialmente.

3.4.2. Reparación y unificación de geometría.

Otro de los problemas que presentan los datos disponibles radica en que algunas estructuras están generadas en base a un método de isosuperficies. Este método genera una malla 3D por cada isosuperficie seleccionada, lo que ocasiona que el archivo resultante este conformado por un conjunto de mallas 3D (que pueden tener errores como agujeros, triángulos mal formados, etc.…), que se entrecruzan entre si (pudiendo generar caras internas) siendo así una malla *non-manifold* no apta para el análisis.

Para solucionar este problema se ha propuesto un método que es capaz de generar una nueva malla 3D de tipo *manifold* en base a un conjunto de mallas 3D *non-manifold,* como es el caso de las espinas de alta precisión generadas por Imaris™. Además, el algoritmo presenta una serie de

parámetros de configuración que le permiten adaptarse a distintos tipos de mallas o a personalizar los resultados obtenidos. En la Ilustración 15 puede observarse una comparativa entre una espina original y 2 espinas reparas con distinta selección de parámetros.

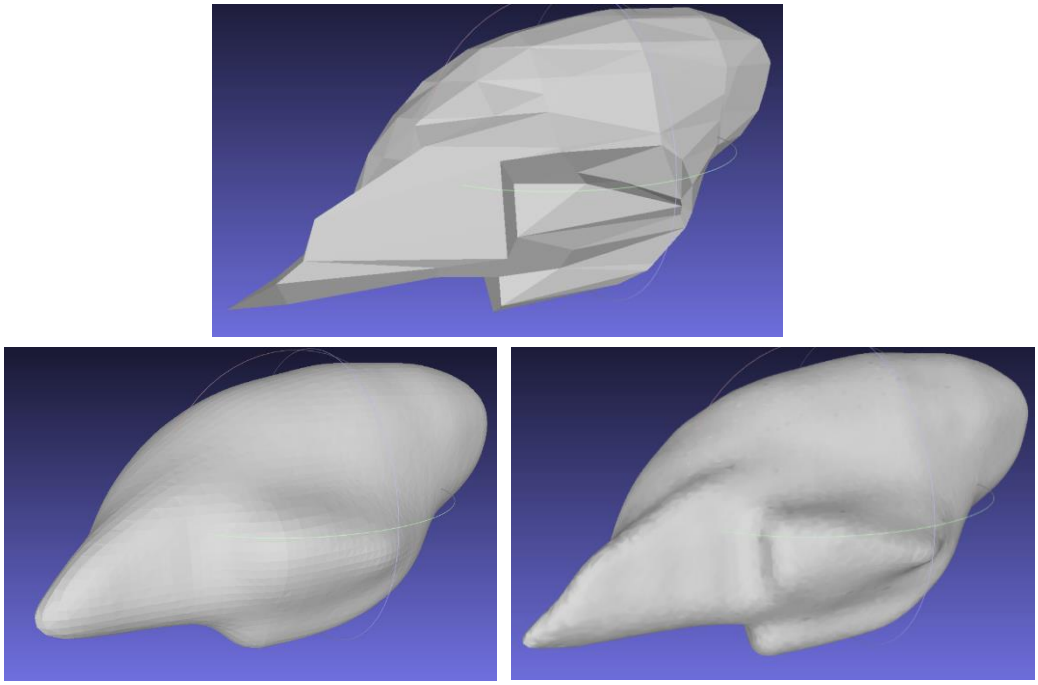


Ilustración 15: Comparativa entre una malla generada por medio de isosuperficies con la herramienta Imaris™ (arriba) presenta caras interiores, etc.…) y la misma malla reparada con el método desarrollado, pero con distintos parámetros de configuración (abajo).

Como se puede observar en la Ilustración 15 el proceso de reparación de las mallas es exitoso para mallas de pequeño tamaño. Sin embargo, para mallas de más tamaño se tenía el problema de que el algoritmo consumía grandes cantidades de memoria al lidiar con *bounding box*es de gran tamaño, que en el caso de los fragmentos dendríticos se encuentra mayoritariamente vacía. Para solucionar este problema se desarrolló un método que es capaz de alinear los ejes de mayor dispersión de la malla con los ejes de coordenadas provocando así que la *bounding box* se ajuste mejor a la malla 3D (ver Ilustración 16).

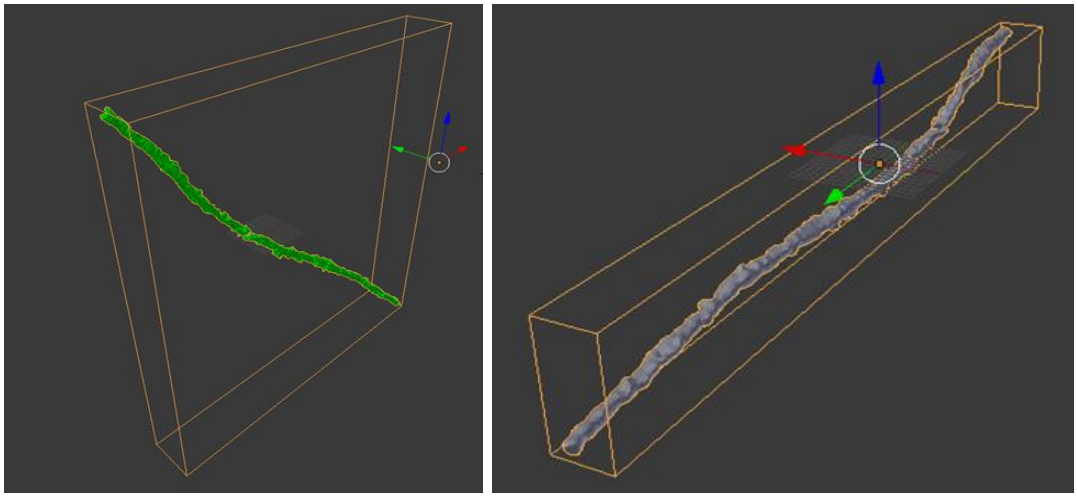


Ilustración 16: A la Izquierda puede verse un fragmento dendrítico en la posición original. Por el contrario, a la derecha puede verse el mismo fragmento dendrítico después del proceso de alineado de los ejes de mayor dispersión del modelo con los ejes de coordenadas.

Sin embargo, aunque el método propuesto para reducir el uso de memoria funciona de forma correcta y ha reducido considerablemente los requisitos de memoria, los resultados obtenidos reparando fragmentos dendríticos no son del todo satisfactorios como puede verse en la Ilustración 17.

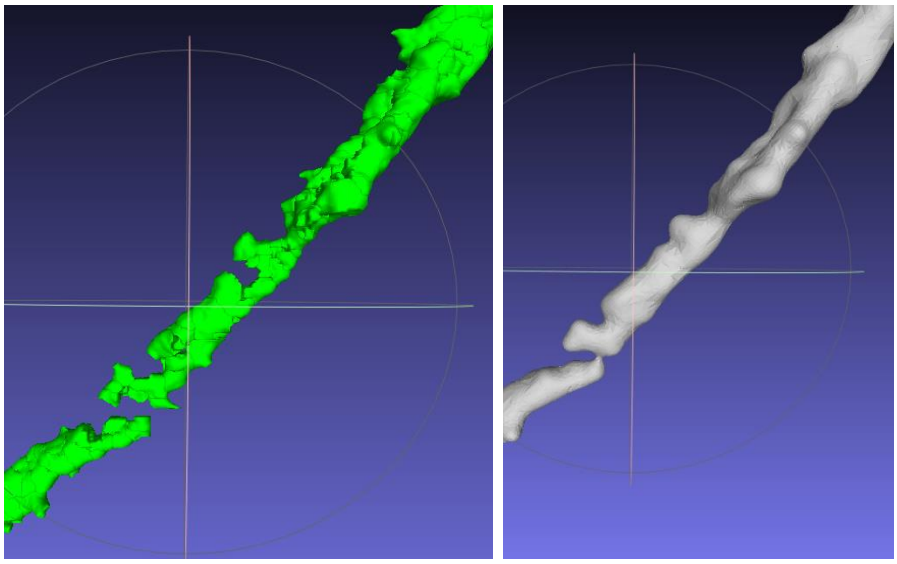


Ilustración 17: Comparativa entre un fragmento dendrítico extraído utilizando Imaris™ y el mismo fragmento reparado por la herramienta propuesta.

Por otro lado, para comprobar la correctitud del algoritmo y poder comparar como se comportaba con diferentes parámetros de configuración se desarrolló una herramienta de visualización que permite comparar dos mallas de manera simultánea, y visualizar las diferencias por medio de colores (Ilustración 17), lo que ha resultado muy útil durante el proceso de validación del algoritmo.

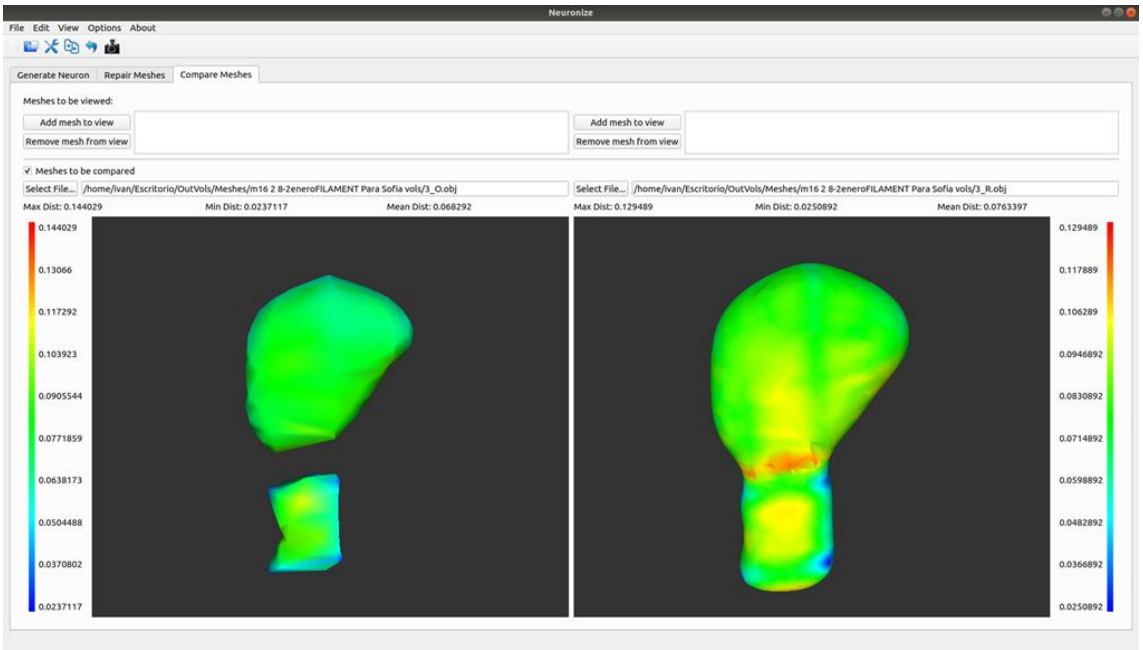


Ilustración 18: Ejemplo de la herramienta de comparación de mallas 3D, comparando una espina original generada por Imaris™ (izquierda) y la espina reparada por el método desarrollado (derecha).

3.4.3. Mejoras del visualizador de datos morfológicos filiformes

Respecto a los visualizadores de datos morfológicos filiformes, y concretamente en el apartado de la morfología neuronal tenían una serie de limitaciones al representar el soma y las espinas dendríticas debido a que no eran capaces de utilizar la nueva información más detallada que incluían los archivos de trazado neuronal.

En la nueva versión de los archivos de trazado neurona el soma se representa por medio de un conjunto de contornos 2D que representa de forma bastante precisa la forma del soma. Sin embargo, esta representación basada en contornos 2D no es apta para ser utilizada en una vista 2D necesitando, por tanto, generar una malla 3D a partir de estos contornos. En la Ilustración 19 puede observarse el resultado de aplicar el método desarrollado a un soma de múltiples contornos 2D generando una malla 3D. La representación es muy buena, aunque pueden observarse algunas diferencias debido a la imposibilidad de capturar zonas convexas debido a limitación del método que serán abordadas en el apartado de discusión.

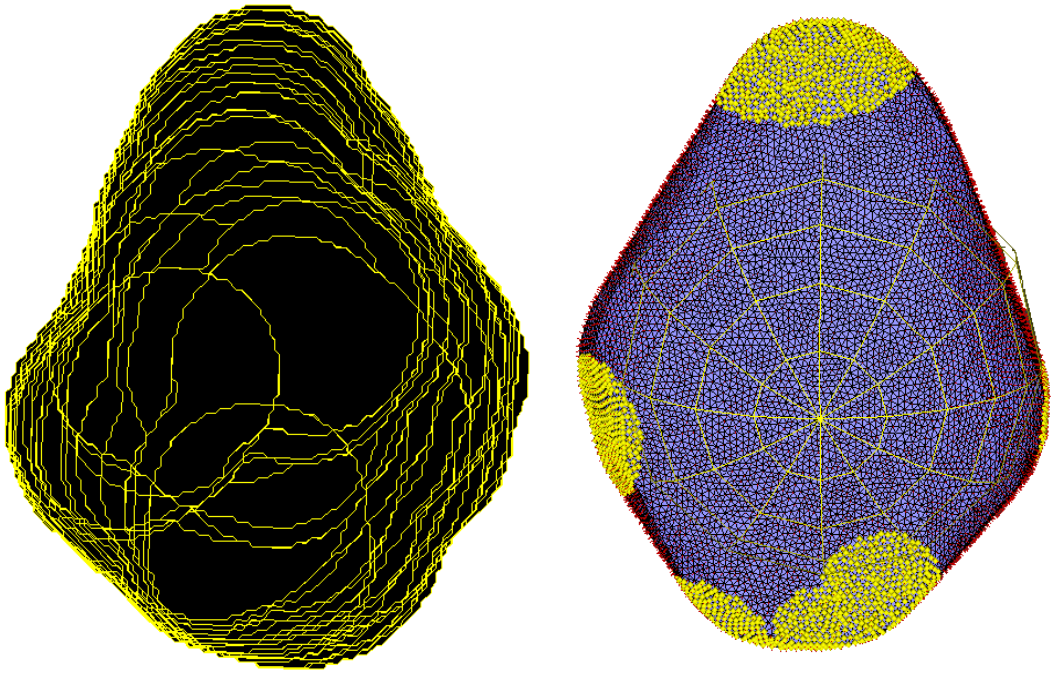


Ilustración 19: A la derecha se observa un soma definido por un conjunto de múltiples contornos 2D. A la derecha se observa la reconstrucción de una geometría representada por una malla 3D partiendo de los contornos 2D con el método propuesto.

Respecto a los nuevos métodos de colocación de espinas, que aprovechan: o bien la información completa sobre la malla de estas si se encuentra disponible, o únicamente la información referente a su posición y orientación han dado muy buenos resultados. En la Ilustración 20 puede observarse una neurona representada por medio de múltiples mallas 3D, y la visualización generada por la herramienta propuesta una vez aplicado el método de generación de jerarquía. En esta imagen, se observa tanto el buen resultado obtenido con el nuevo método de colocación de espinas, como el resultado del generador de la jerarquía, ya que, se ha aplicado como preproceso para poder realizar esta comparación.

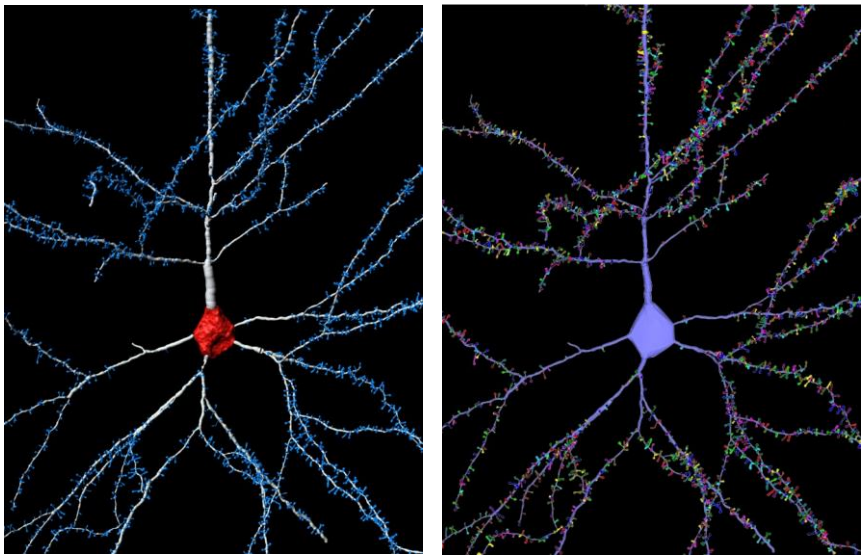


Ilustración 20: Comparativa de una neurona representada por medio de mallas 3D que no presenta estructura ni jerarquía (izquierda), y la conversión a una representación de polilíneas conseguida por el método diseñado. Nótese que la representación de polilíneas se está visualizando de manera volumétrica para facilitar su comparación.

3.5. Discusión.

En este capítulo se han propuesto distintos métodos para mejorar el análisis y visualización de datos morfológicos filiformes, pasando por generar nueva información de jerarquía que facilite la realización de análisis posteriores y la corrección de algunos datos basados en mallas 3D no aptas para el análisis. Por otro lado, se han realizado mejoras en las herramientas de visualización permitiendo obtener representaciones más realistas gracias al uso de información más detallada, así como una nueva herramienta de visualización que permite comparar mallas 3D entre sí, permitiendo validar los resultados obtenidos. Además, todas las soluciones propuestas han sido validadas dentro del campo de la morfología neuronal

En la medida de nuestro conocimiento no existía anteriormente ninguna herramienta o método, en el campo de los datos morfológicos filiformes, que fuese capaz de generar información sobre estructura y jerarquía partiendo de datos representados por medio de mallas 3D que carecen de esta. La gran ventaja de disponer de esta información sobre la jerarquía es que se tiene disponible de forma sencilla toda la información sobre las bifurcaciones, y cuando ocurren estas, siendo una de las características principales de las estructuras filiformes, lo que facilita enormemente su análisis. Además, gracias al trabajo realizado en esta dirección se ha solucionado un problema de interoperabilidad entre dos de las herramientas más utilizadas en el dominio de la morfología neuronal (Imaris™ y Neurolucida™). Esto ha provocado que la cantidad de datos disponibles para el análisis en este campo haya aumentado sustancialmente al poder obtenerlos de forma automática. Por otra parte, debido a que los procesos de digitalización, tanto en Imaris™ como en Neurolucida™ sean en menor o mayor medida manuales, provocaba que nunca se tuviesen exactamente los mismos datos, dificultando así su comparación.

Por otra parte, también se dispone de datos sobre espinas dendríticas con alto nivel de detalle. Sin embargo, estos datos basados en una representación de malla 3D, presentan ciertos errores a la hora relacionados con la forma de construcción de las mallas debido a su proceso de creación basado en isosuperficies. Esta forma de obtener los archivos provoca que estén conformados por múltiples mallas 3D no conectadas entre sí, que pueden intersecarse, estar

unas dentro de otras, o presentar problemas a nivel individual como agujeros, es decir, es una malla de tipo *non-mainifold* no apta para el análisis. Para poder utilizar esta información en análisis es necesario obtener una representación con una única malla 3D correcta, es decir de tipo *manifold*. Esto se ha realizado por medio de un algoritmo de reparación que ha obtenido muy buenos resultados, como se puede apreciar gracias a la herramienta de comparación de mallas desarrollada, que ha permitido visualizar de manera sencilla en que puntos se estaba cometiendo más error, así como seleccionar los mejores parámetros para la reparación. Además, esta herramienta de comparación también ha sido muy útil para demostrar a los expertos en el dominio que la herramienta de reparación funcionaba bien para su tipo de datos, y que el error cometido era mínimo, generando así una gran confianza en los usuarios que pueden comprobar por sí mismos la calidad de la reparación. Respecto a las mejoras implementadas en la gestión de memoria de la herramienta de reparación, por medio de ajustar mejor la malla a la caja contenedora, se ha conseguido reducir en gran medida los requisitos de esta al reparar mallas grandes como la de los fragmentos dendríticos. Sin embargo, los resultados obtenidos al reparar los fragmentos dendríticos no son del todo satisfactorios, aunque posiblemente podría haberse conseguido una mejor reparación modificando los parámetros del algoritmo.

Las mejoras implementadas en el visualizador de datos morfológicos filiformes ahora permiten una representación más precisa tanto del soma como de las espinas dendríticas aprovechando la nueva información presente en los nuevos formatos de trazado neuronal. Esta información es una representación basada en contornos 2D del soma que permite generar una malla 3D que aproxime a estos contornos, e información sobre la posición y orientación de las espinas dendríticas. Respecto a otras herramientas que también visualizan datos morfológicos filiformes provenientes del campo de la morfología neuronal cabe destacar: NeuroMorphoVis [37]que sigue un enfoque similar tanto para la creación de la malla 3D, como en el proceso de inferencia del soma a partir de una esfera, que la herramienta original en la que se basa este trabajo[36], además, propone métodos para editar el trazado neuronal, así como herramientas de reparación. Sin embargo, no es capaz de aprovechar la información de contornos 2D generando somas menos realistas y no tiene soporte para espinas dendríticas. Por otro lado, NeuroTessMesh [38] propone una versión mejorada del método de generación del soma cambiando el método de deformación por el método de elementos finitos que es más fácil de configurar y genera superficies más suaves. Sin embargo, tampoco es capaz de aprovechar la información de múltiples contornos derivando en un soma menos preciso. Respecto a la generación de la malla 3D propone un método basado en teselación adaptativa con múltiples niveles de detalle que le permite mostrar un gran conjunto de neuronas, en lugar de una única neurona como la herramienta propuesta. Sin embargo, el soporte para espinas es limitado y no aprovecha la información sobre posición y orientación de las espinas dendríticas, generando así espinas de menor precisión.

El método de generación de la malla 3D del soma partiendo de una representación de contornos 2D presenta una limitación importante, al estar basado en el algoritmo de *convex-hull* el método únicamente es capaz de generar somas que sean completamente convexos. De forma intuitiva el algoritmo puede entenderse como recubrir con un material elástico todos los contornos 2D, de esta forma el material elástico no es capaz de ajustarse de manera eficiente a las partes cóncavas, generando así errores de aproximación. Aunque para esta aplicación en concreto no es muy problemático, ya que los somas son en su mayor parte convexos, podría mejorarse el algoritmo para que lidie mejor con las partes cóncavas.

4. Análisis y clasificación de series temporales homogéneas

Esta sección se centra en el análisis de series temporales homogéneas multivariantes, y más concretamente en la tarea de clasificación siguiendo un enfoque genérico que pueda ser aplicado a distintos dominios, aunque en este trabajo se valida utilizando datos del dominio EEG.

Como se ha comentado en la sección de introducción los datos de carácter temporal presentan una dificultad y complejidad suficientes como para tener su propia categoría. Este tipo de datos puede observarse en prácticamente todos los campos y aplicaciones lo que le otorga una gran importancia. Concretamente este capítulo se centra en las series temporales, las cuales consisten en la monitorización de una o más variables en un periodo temporal, donde se toman valores cada cierto tiempo (siendo normalmente este intervalo entre mediciones constante). Las series temporales pueden analizarse principalmente mediante dos enfoques: por un lado, con un enfoque univariante que únicamente tiene en cuenta el comportamiento de cada señal de forma individual y, por otro lado, con un enfoque multivariante, que aparte de tener en cuenta cada señal de forma individual también tiene en cuenta las relaciones que tiene las distintas señales entre sí.

Aunque el método se desarrolla de forma genérica pensando en cualquier tipo de serie temporal, en este trabajo se aplicado a datos provenientes de EEG. Este tipo de datos puede verse como una serie temporal multivariante homogénea, donde cada uno de los electrodos utilizados corresponde con una señal de la serie temporal completa y cada electrodo recoge el mismo número de variables.

Para el análisis de estas series temporales se ha decidido utilizar un enfoque basado en la extracción de características y el análisis multiresolución. A grandes rasgos, el método comienza realizando un análisis multiresolución (*Multiresolution Analysis*, MRA) descomponiendo cada una de las señales que el electroencefalograma completo en varios niveles teniendo cada nivel distinta resolución para frecuencias y tiempo. Seguidamente, partiendo de estas señales de nivel, se extraen una serie de características que buscan definir dichas señales y las relaciones entre ellas. Sin embargo, para un tipo de datos como el de EEG o el de ECG, por ejemplo, el número de características extraídas (del orden de decenas de miles para el EEG) es demasiado grande para utilizarlo en un clasificador puesto que provocaría problemas de sobreajuste (*overfitting*, por lo que se seleccionan las características más relevantes por medio de un discriminante por pasos (*Stepwise Discriminant*). Por último, estas características seleccionadas son las que se utilizan para entrenar un clasificador. En la Ilustración 21 se muestra un diagrama con cada una de las etapas.

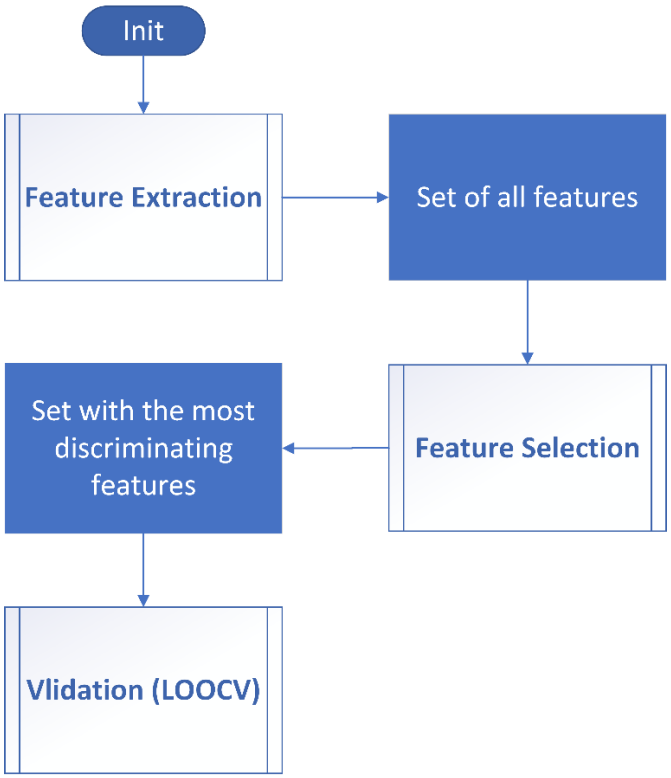


Ilustración 21: Esta figura explica mediante un diagrama de flujo cada una de las etapas principales del método de clasificación propuesto que serán explicadas en más detalle en secciones posteriores.

Como se ha comentado anteriormente, aunque el método ha sido diseñado de manera genérica para funcionar con cualquier juego de datos con carácter temporal, independientemente del ámbito, se valida con datos del dominio EEG. En particular, los datos de EEG que se usan están relacionados con dos tareas diferentes de imagen motoras (la persona imagina que realiza un movimiento): el movimiento de manos y el movimiento de pies que son las clases entre las cuales se clasificarán los datos.

En las siguientes subsecciones se detalla el funcionamiento de cada una de las etapas que conforman el método. Seguidamente, se comentan los resultados obtenidos de la validación del método aplicado a datos de EEG, para terminar, explicando cómo se ofrece una librería desarrollada en base al método para facilitar su utilización.

4.1. Descomposición de la señal: *MODWT*

El primer paso del método consiste en descomponer cada una de las señales de la serie temporal multivariante. Sin embargo, para asegurar una aportación igualitaria de cada una de las señales que componen la serie temporal, se lleva aplica un proceso de normalización a cada una de ellas aplicando la siguiente expresión $\frac{(x-\bar{X})}{\sigma_x}$. Después de normalizar las señales que componen la serie temporal, se comienza con el proceso de descomposición utilizando el método de *Discrete Wavelet Transform* (DWT) [88] que es capaz de re-expresar la señal original en una serie de coeficientes que están asociados con un tiempo concreto, un filtro específico a utilizar, y con una escala diádica concreta [89]. Esta escala diádica permite controlar la sensibilidad del método a un cierto rango de frecuencias dadas, donde valores de escalas más pequeños se utilizan para obtener información de alta frecuencia, mientras que valores más elevados se utilizan para la información de baja frecuencia. Por este motivo, el método realiza un MRA en el dominio de la frecuencia como en el del tiempo.

Aunque DWT es un gran método para realizar MRA, tiene una serie de limitaciones. La limitación más importante para este trabajo es que el número de coeficientes generados se reduce a la mitad por cada aumento del parámetro de escala. Esta reducción en el número de coeficientes provoca que la transformación sea incapaz de mantener la propiedad invariante en el tiempo de la serie temporal original, además también hace bastante complicado asociar los coeficientes de cierta escala con los valores originales de la serie temporal. Para solucionar estas limitaciones se usa una variante del método, la transformada wavelet discreta de máximo solapamiento (*Maximal Overlap Discrete Wavelet Transform,* MODWT). Esta variante genera en cada escala el mismo número de coeficientes que el número de observaciones de la serie original.

Dado un filtro wavelet concreto ($\tilde{h}_{j,l}$) y un filtro de escala ($\tilde{g}_{j,l}$) (donde j es el nivel de descomposición, los coeficientes de la aplicación de MODWT a una señal $\widetilde{W}_j$, así como sus coeficientes de escala $\widetilde{V}_j$ se definen como una transformación de la serie temporal $x = x_t,\ t = 0,1,2,3 \dots, N-1$, (donde N es el número total de observaciones en la serie temporal), de la siguiente forma:

$$\widetilde{W}_{X,j,t} = \sum_{l=0}^{L_j-1} \widetilde{h}_{j,l} X_{t-l mod N}$$

$$\widetilde{V}_{X,j,t} = \sum_{l=0}^{L_j-1} \widetilde{g}_{j,l} X_{t-l mod N}$$

Donde $L_j = (2^j - 1)(L-1) + 1$, siendo L el tamaño del filtro seleccionado.

En relación con el filtro específico utilizado para procesar la señal, existen diferentes familias, siendo cada una de ellas más útil para ciertos análisis que otras. Las familias utilizadas en este trabajo son las siguientes [90]:

- Haar: Este filtro wavelet es el más básico. Presenta una forma cuadrada, lo que tiene de inconveniente que no es diferenciable. Sin embargo, esto puede ser una ventaja para el análisis de señales con cambios repentinos como es el caso de las señales digitales [91].
- Daubechies: Esta familia de wavelets es una extensión de la wavelet de Haar, donde cada una de las wavelets se define por el número de *Vanishing Moments. Estos Vanishing Moments* determinan el polinomio de mayor grado que la wavelet puede reconstruir, de esta forma un número mayor de *Vanishing moments* permite lidiar con señales más complejas. 41ea ejemplo *db2* es una wavelet de Daubechies con 2 *Vanishing Moments*. Nótese que la wavelet *db1* coincide con la wavelet de Haar.
- Symlets: Esta familia de wavelets es una modificación de la familia de Dauchebies con el objetivo de presentar una familia de wavelets casi simétrica. Esta propiedad de simetría puede ser útil en ciertos contextos donde el error es menos importante si se presenta de forma simétrica (como por ejemplo en imágenes [90]). Además, también tienen la ventaja de que es más sencillo lidiar con los extremos de la señal con wavelets simétricas
- Coiflets: Esta familia de wavelets intenta maximizar el número de *Vanishing Moments.* Esto es útil para tareas de compresión de información debido a que maximiza el número

de coeficientes que son próximos a 0, que podrían descartarse al no proveer de mucha información.

Por otro lado, se ha comentado anteriormente que MODWT opera en base a distintos niveles de descomposición, donde valores pequeños de escala hacen referencia a información de alta frecuencia, mientras que valores mayores lidian mejor con información de baja frecuencia. Sin embargo, existe un límite en el nivel de descomposición que se puede aplicar que depende del tamaño de la serie temporal (N), de tal forma que el nivel máximo viene definido por la expresión $J \leq \log_2(\mathrm{N})$. Sin embargo, este límite genérico todavía puede originar ciertos problemas dependiendo de la longitud del filtro seleccionado (L), por ello para asegurar un correcto funcionamiento el máximo nivel de descomposición seguro se define como $J \leq log\_2(N/(L-1)+1$.

Una vez se ha descompuesto cada una de las señales que conforma la serie temporal multivariante en el número de niveles deseado, se procede a utilizar los coeficientes obtenidos para extraer distintas características relevantes.

4.2. Extracción de características

Aunque la descomposición de las señales realizadas por MODWT contiene una gran cantidad de información es necesario sintetizar esta información en una serie de características que sean más manejables en un proceso de clasificación tradicional. Para ello, se calculan a partir de estas señales descompuestas tanto características univariantes únicamente relativas a la señal individual, como características que intentan resumir la relación entre los niveles de descomposición de las señales que componen la serie temporal. Las características con un enfoque univariante obtenidas son las siguientes (nótese que, por simplicidad, los coeficientes de un cierto nivel serán representados por W, 42ealizars que V es la varianza):

- Varianza:

$$V_{X,j}^2 = \frac{1}{M_j} \sum_{t=L_j-1}^{N-1} W_{j,t}^2,$$

  donde $M_j = N - L_j + 1$ç
- Rango intercuartílico (IQR):

$$IQR_{X,j} = P_{75}(W_{X,j,t}) - P_{25}(W_{X,j,t})$$

- Entropía de permutación (*Permutation Entropy, PE*):

$$PE = \frac{-\sum_{j=1}^{m!} p_j \, ln(p_j)}{ln(m!)},$$

  donde $p_j = \frac{n_j}{\sum_{j=1}^{m!} nj}$, m es la dimensión de integración, n_j es el número de veces que la permutación Jth ocurre.

Respecto a las características que involucran más de una señal, es decir las que obtienen información sobre las relaciones entre estas, es necesario tener en cuenta que únicamente se calcula esta relación para un mismo nivel de descomposición en ambas señales.

- Coeficiente de correlación de Pearson (Pearson Correlation Coefficient, PCC):

$$\rho'_{XY,j} = \frac{\sum_{t=L_j-1}^{N-1}(W_{X,j,t} - \bar{W}_{X,j})(W_{Y,j,t} - \bar{W}_{Y,j})}{\sqrt{V_{X,j}} * \sqrt{V_{Y,j}}}$$

- Medida D de Hoefflin (*Hoefflin's D measure*, D):

$$D = 30(\frac{(N-2)(N-3)D_1 + D_2 - 2(N-2)D_3}{N(N-1)(N-2)(N-3)(N-3)},$$

donde $D_1 = \sum_{i=1}^{N} Q_i(Q_i - 1)$, $D_2 = \sum_{i=1}^{N} R_i(R_i - 1)(R_i - 2)(S_i - 1)(S_i - 2)$, $D_3 = \sum_{i=1}^{N}(R_i - 2)(S_i - 2)Q_i$. R_i y S_i son rangos de $W_{X,j,t}$ y de $W_{Y,j,t}$ respectivamente. Por otro lado, Q_i son los rangos bivariados.

Respecto al número total de variables que se extraen de la serie temporal original y que, algunas de ellas, se usarán para la clasificación, hay que tener en cuenta, por un lado, las variables relacionas con características que involucran únicamente una señal y, por otro, las variables relacionadas con las características que involucran dos señales de nivel. En relación con las características que involucran una sola señal, hay que tener en cuenta que se obtienen J variables por cada una de las señales y por cada una de las características univariantes comentadas. De esta forma, el número total de variables relacionadas con características univariantes serían $3 * (J * N)$. Por otro lado, el número de variables relacionadas con características que involucran más de una señal (es decir las multivariantes) es mucho mayor que las univariantes, debido a que es necesario calcular estas para cada par de señales. Así, el número de variables relacionadas con una característica multivariante dependerá del número de posibles combinaciones de dos señales y el nivel de descomposición. De esta forma, el Número total de variables relacionadas con características multivariantes obtenidas por el presente método serían $2 * (\binom{N}{2} * j)$. Como puede apreciarse, el número total de variables a tener en cuenta para la clasificación puede ser demasiado grande para la mayoría de los casos, provocando problemas de *overfitting*.

En la Ilustración 22 puede verse un diagrama de los pasos realizados por el algoritmo hasta obtener el conjunto de todas las variables que pueden ser utilizadas para la clasificación, que serán filtradas en el próximo paso.

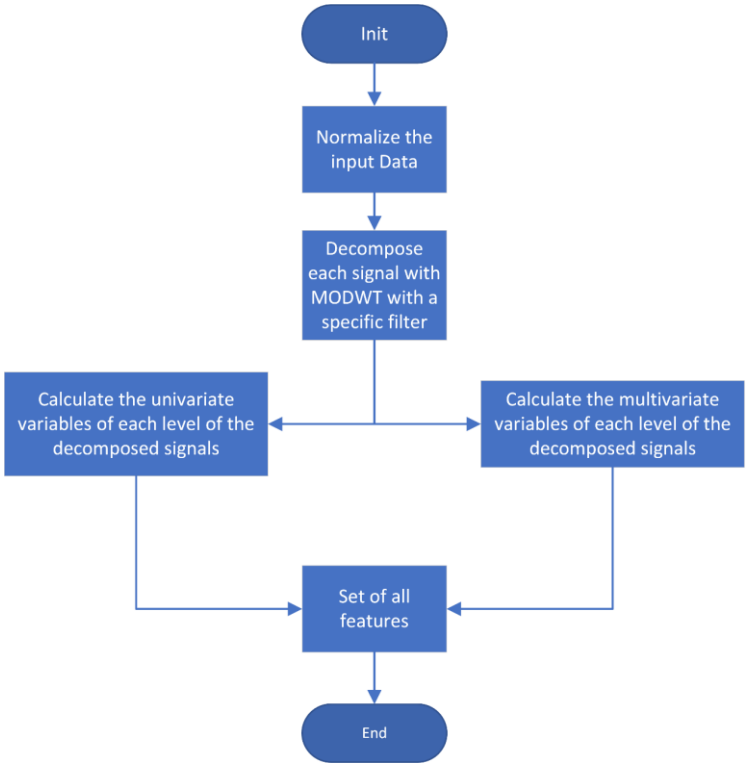


Ilustración 22: Diagrama que muestra en detalle la etapa de extracción de características, que es la encargada de obtener todo el conjunto de variables

4.3. Selección de características: discriminante por pasos

Como se ha comentado anteriormente el gran número de variables que calcula el proceso de extracción de variables puede ser problemático de cara a los algoritmos de clasificación ya que puede provocar problemas de *overfitting*.

Para solucionar este problema se lleva a cabo un algoritmo iterativo discriminante por pasos (*stepwise discriminant)* que selecciona un subconjunto de todas las variables calculadas en el paso anterior en función de su poder discriminante. El método consiste en introducir de forma iterativa una variable a la vez en el conjunto de variables más discriminantes, observando cómo afecta su introducción al poder discriminante del conjunto y escogiendo la variable que maximice este poder discriminante. Este proceso se repite hasta obtener el número de variables deseado como se indica en [92]. En la Ilustración 23 pueden observarse las etapas del algoritmo.

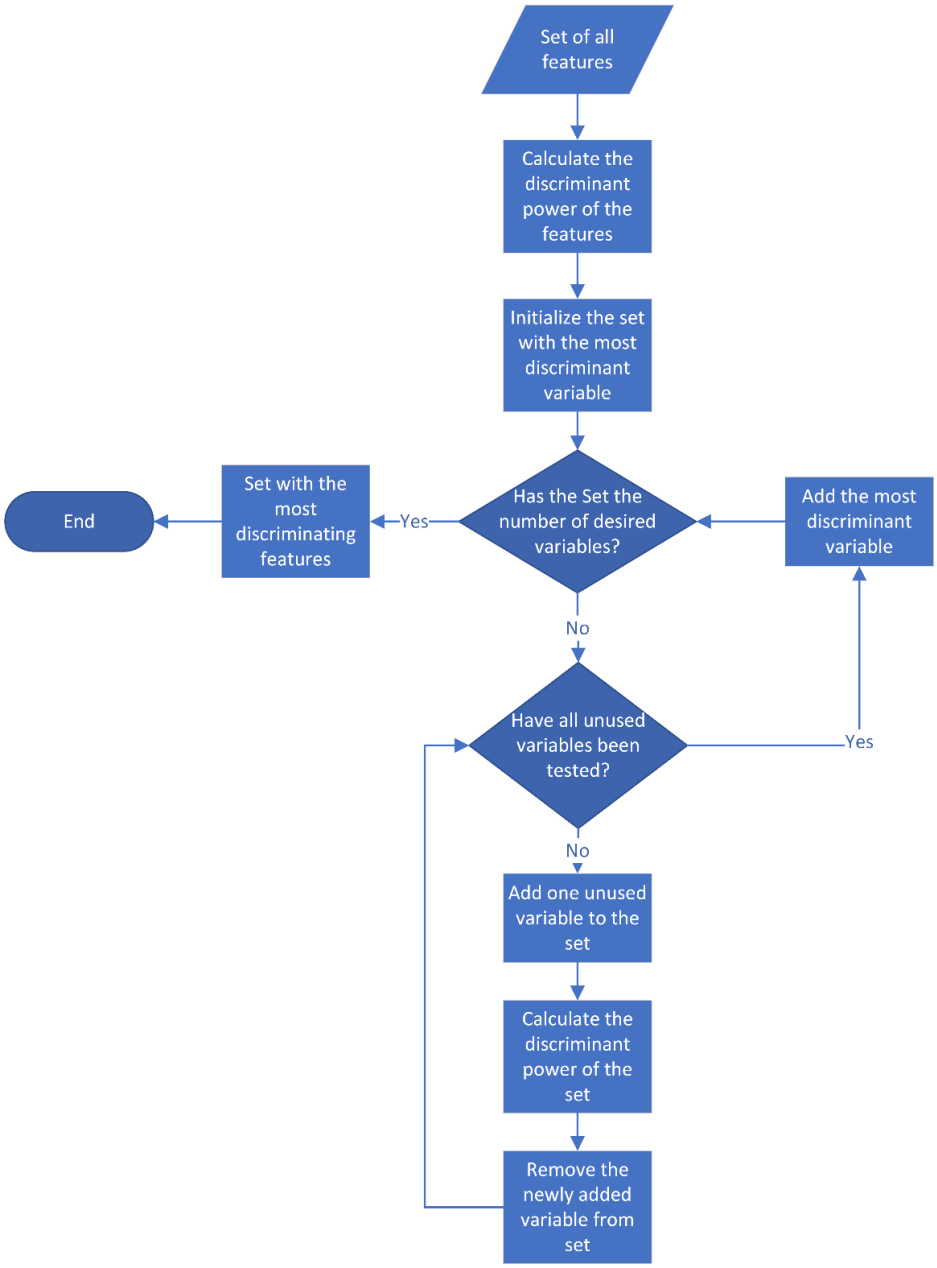


Ilustración 23: Vista en detalle de la etapa de selección de características. Esta etapa parte del conjunto completo de variables calculadas por la etapa de extracción de características, y selecciona las que presentan mayor poder discriminante. De esta forma se obtiene un nuevo conjunto de variables reducido con las variables más explicativas.

Para determinar el poder discriminante del subconjunto de variables, se utiliza la traza de Lawlley-Hotteling descrita por la siguiente ecuación:

$$V = (n-g)\sum_{i=1}^{p'}\sum_{j=1}^{p'} a_{ij}\sum_{k=1}^{g} n_k(\overline{X_{ik}} - \overline{X_i})(\overline{X_{jk}} - \overline{X_j}),$$

donde:

- n = Número de observaciones
- g = Número de grupos
- p’ = Número de variables discriminantes

- n_k = Número de casos en el grupo k
- $\overline{X_{ik}}$ = Media de la variable i en el grupo k
- $\overline{X_i}$ = Media de la variable i en todos los grupos
- a_{ij} = Un elemento de la matriz inversa de la suma de productos cruzados dentro de los grupos (también llamada W [93]):

$$W_{ij} = \sum_{k=1}^{g} \sum_{m=1}^{n_k} \left(X_{ikm} - \overline{X_i k}\right)\left(X_{jkm} - \overline{X_{jk}}\right),$$

donde X_{ikm} es el valor de la variable *i* para el caso *m* en el grupo *k*.

Esta ecuación mide la distancia entre los centroides de los grupos discriminados (es decir, la distancia entre las medias de los grupos), pero no tiene en consideración la cohesión entre estos grupos [93].

4.4. Clasificación

Una vez se ha seleccionado un número determinado de variables que se consideran las más discriminantes, se procede a utilizarlas para entrenar un clasificador y poder predecir así nuevas observaciones.

Para este proceso de clasificación se han utilizado dos tipos de clasificadores (aunque podría utilizarse cualquier otro), por un lado, un discriminador lineal (*Lineal Discriminant Analysis,* LDA) y, por otro lado, un discriminador cuadrático (*Quadratic Discriminant Analysis,* QDA). El clasificador utilizado dependerá del tipo de datos a tratar, ya que su efectividad dependerá de estos.

Con respecto de los métodos de validación cruzada utilizados para comprobar el rendimiento del algoritmo, se dispone por una parte de la validación cruzada dejando uno fuera (*Leave-One-Out Cross-Validation*, LOOCV). El método consiste en utilizar todos los datos disponibles salvo uno para entrenamiento dejando ese único caso para validación, repitiéndose este proceso para todos los datos del conjunto. Este método tiene la ventaja de presentar una estimación más robusta del rendimiento del algoritmo evaluado a costa de un gran coste computacional. El otro método disponible es la validación cruzada de K-grupos (*K-fold Cross-Validation*, KFCV) que divide el conjunto de datos en K subgrupos, de tal forma, que uno de los subgrupos será seleccionado para validación mientras que el resto de los subgrupos serán utilizados para entrenamiento. Este método tiene la ventaja de ser menos costoso computacionalmente (al tener que entrenar solamente K modelos), pero tiene menos precisión que LOOCV.

4.5. Resultados

En esta sección se detallarán tanto el proceso de validación del algoritmo, como los resultados obtenidos aplicando el método a datos provenientes del dominio de EEG.

4.5.1. Datos utilizados

Los datos utilizados para probar el rendimiento del algoritmo desarrollado pertenecen al dominio de EEG. Para comprender mejor la naturaleza de estos datos se va a detallar el diseño del experimento en el que se obtuvieron estos datos.

Los datos parten de un total de 20 sujetos adultos neurológicamente sanos (11 mujeres, 9 hombres) con un rango de edad de 25-66 años. Las señales de EEG son registradas durante 5.5 segundos con un gorro de electrodos de 64 canales, donde el electrodo del ojo ha sido descartado para el proceso de clasificación. El experimento consiste en seguir una serie de

órdenes de tal forma que los participantes deben realizar 8 tareas organizadas en bloques aleatorios.

- Cuatro ejercicios en los que el participante tiene que imaginar que cerraba el puño y después relajaba la mano.
- Cuatro ejercicios en los que el participante tiene que imaginar movimiento de los dedos de los pies seguido de relajación de estos.

Para realizar el experimento cada participante debe completar 15 ensayos del ejercicio requerido después de escuchar el sonido de un silbato. Cada participante completa secuencialmente un total de 8 bloques presentados de forma pseudoaleatoria de tal forma que nunca se muestren más de 2 bloques del mismo tipo de forma consecutiva, con un descanso de 1 a 2 minutos entre bloques. Cada bloque comienza con la presentación auditiva de las instrucciones de la tarea para ese bloque donde se pide a los sujetos que realicen una acción (o bien apretar su mano derecha en un puño y después relajarla, o mover todos los dedos de ambos pies) cada vez que se escuche un pitido. Los sujetos deben realizar la acción en cuanto oyen el pitido. Después de 5 segundos, una vez finalizadas las instrucciones, se presentan 15 pitidos de forma binaural (600 Hz durante 60 ms) con un intervalo aleatorio entre 4.5 y 9.5 segundos. El bloque finaliza con una instrucción de relajación. El experimento completo incluye 400 ejercicios, siendo la mitad de estos movimientos de manos y los otros movimientos de pies.

Respecto a las señales de EEG, estas han sido registradas con un equipo de OSG que tiene 2 amplificadores Schwarzer AHNS de 44 canales con una frecuencia de muestreo de 1000 Hz. Los datos han sido pasados por un filtro de paso alto a 0.27 Hz y no se ha aplicado ningún filtro de paso bajo. La etapa de post-proceso se ha llevado a cabo con el software Cartool [94], específicamente la línea base se ha seleccionado como los 500 ms previos al estímulo. Los datos han sido recalculados de forma individual basándose en la media de referencia, y se ha aplicado un filtro paso banda en el rango de frecuencias 1 Hz – 40 Hz. Seguidamente, un proceso automático elimina los ensayos en los que se observa una amplitud superior a 100 mV en alguno de los electrodos. Finalmente, los ensayos se han inspeccionado manualmente para quitar parpadeos, movimientos y artefactos musculares. Los datos faltantes debidos a estos artefactos se interpolan utilizando un algoritmo de *spline* 3D.

Se puede encontrar una descripción completa del diseño del experimento en [47].

4.5.2. Resultados de clasificación

En esta sección se muestran los resultados obtenidos de aplicar el método desarrollado a los datos de EEG previamente comentados. Los datos constan de un total de 63 señales (electrodos) registradas durante 5.5 segundos a una frecuencia de 1000 Hz (5500 muestras) para un total de 400 ensayos. Es importante destacar que el clasificador ha sido entrenado con los datos de todos los sujetos al mismo tiempo, siendo por tanto un clasificador multi-sujeto.

Como se ha comentado, el método desarrollado tiene una serie de parámetros que pueden afectar al rendimiento del algoritmo. Estos parámetros están relacionados por un lado con DWT en sí, como el tipo de filtro seleccionado o el número total de variables a tener en cuenta en la clasificación, mientras que otros parámetros están más relacionados con el tipo de datos a tratar como el tipo de características a calcular (varianza, correlación, etc.) o el método de clasificación utilizado (lineal o cuadrático).

Diferentes combinaciones de características y métodos de clasificación pueden afectar el rendimiento general del método, así como proporcionarnos ciertas pistas sobre el comportamiento de las señales analizadas. Por ello, los resultados se han generado para cualquier posible configuración de los parámetros del método (filtro, características y número de variables) ejecutando el método por cada una de estas posibles combinaciones. Estos resultados se pueden observar en la Ilustración 24 y en la Ilustración 25, donde cada figura muestra la precisión obtenida en función de las variables tenidas en cuenta. Cada figura tiene tres gráficas, dependiendo de si el algoritmo ha utilizado como característica la varianza (Vars), la correlación (Cors) o ambas (Vars & Cors). Estas gráficas se han calculado utilizando tanto el discriminante lineal como el cuadrático.

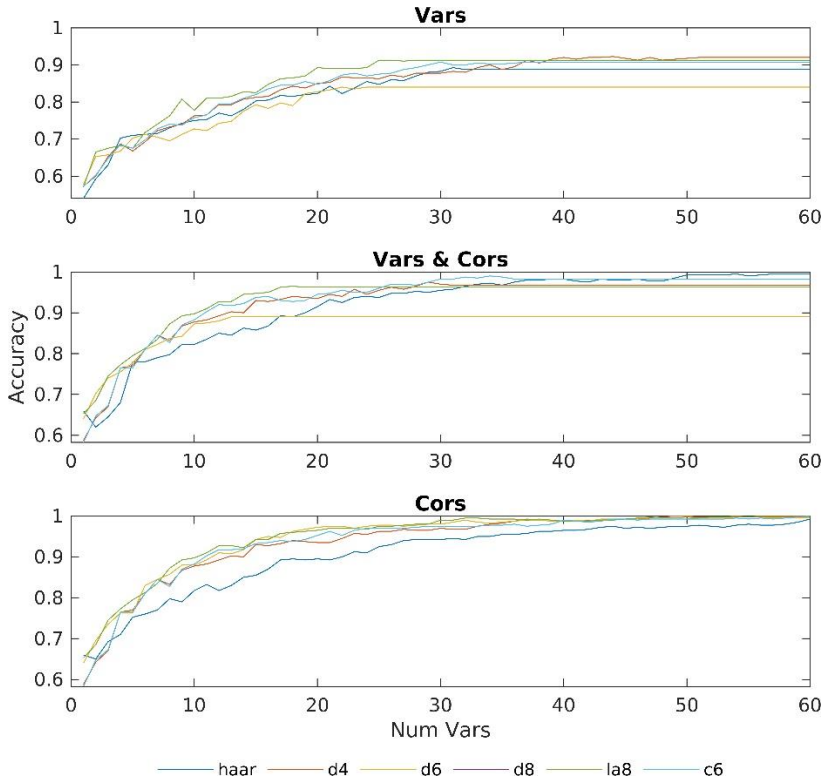


Ilustración 24: Precisión obtenida utilizando un discriminante lineal con subconjuntos de entre 1 y 60 de las variables más discriminantes. Imagen incluida en [18]

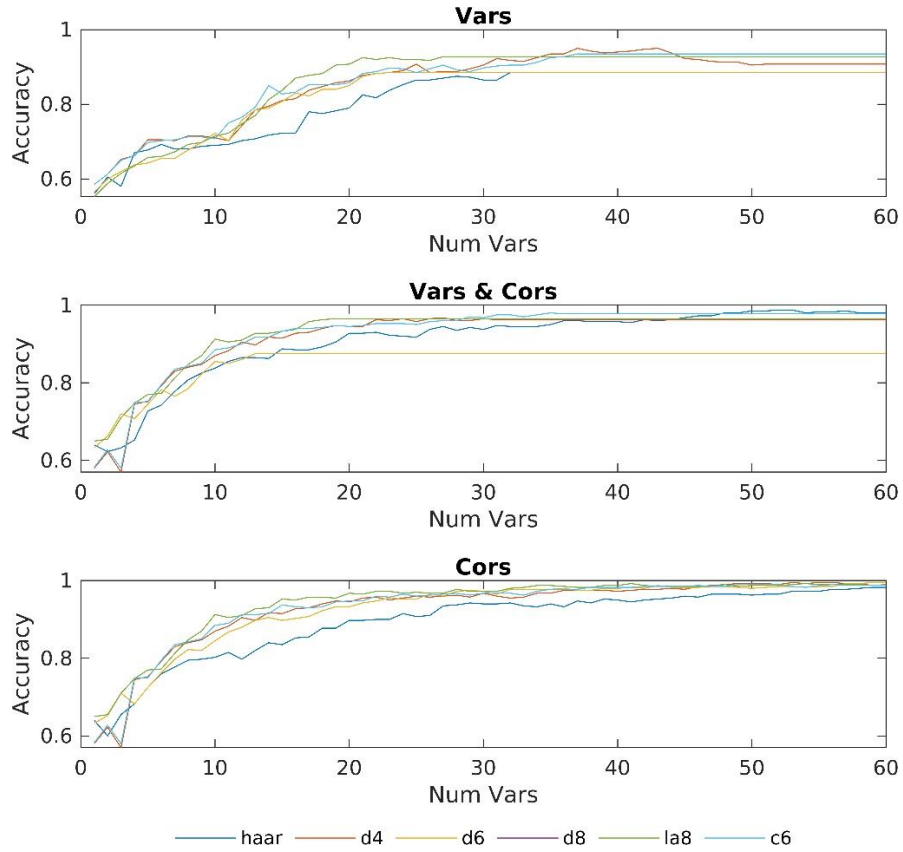


Ilustración 25: Precisión obtenida utilizando un discriminante cuadrático con subconjuntos de entre 1 y 60 de las variables más discriminantes [18]

Como puede observarse en la Ilustración 24, con un discriminante lineal y utilizando únicamente 55 variables o más para la clasificación, se obtiene una precisión en la clasificación del 100% utilizando el filtro *d4*. Además, para 50 variables con el mismo filtro ya se obtienen precisiones muy cercanas al 100%. En relación con los otros filtros wavelets utilizados, se puede observar como todos ellos, excepto Haar, consiguen una precisión de entre el 98% y el 100% utilizando 40 variables, mientras que utilizando 45 todos ellos exceden el 99% de precisión.

Por otro lado, si comparamos el rendimiento del discriminante lineal contra el discriminante cuadrático, se puede observar cómo, para estos datos, el discriminante lineal es más preciso. También se puede observar cómo, para estos datos en concreto, la correlación entre las distintas señales (electrodos) es más importante que las varianzas, indicando que la relación en la activación de las diferentes señales es más importante que la activación individual de cada una de ellas. Además, también se observa cómo el rendimiento utilizando varianzas y correlación es inferior al obtenido utilizando únicamente correlaciones. Esto parece indicar que el algoritmo de selección de las variables más discriminantes no realiza una selección óptima en el caso de estudio escogido.

Respecto al tiempo de ejecución del algoritmo (ejecutado en con un Ryzen 1600x y 16Gb de RAM), el proceso de obtener la precisión de la clasificación para todas las combinaciones posibles de parámetros ha requerido 1 día. Sin embargo, es importante destacar que este proceso únicamente debe ser llevado a cabo una vez para cada problema de clasificación con la intención de encontrar los mejores parámetros de configuración del método para el caso concreto, siendo necesarios posteriormente únicamente unos minutos para obtener los resultados de una combinación concreta de parámetros.

*Tabla 1: Valores medios obtenidos en el proceso de clasificación utilizando los filtros: haar, d4, d6, d8, la8 y c6 para un determinado tamaño de conjunto de variables para clasificación **(Size)**, un determinado clasificador **(Linear o Quadratic)** y unas determinadas características **(Var, Var&Cor o Cor)***

	Size	Linear			Quadratic		
		Var	Var&Cor	Cor	Var	Var&Cor	Cor
Accuracy	20	0,86	0,94	**0,95**	0,86	0,94	0,94
	40	0,90	0,96	**0,98**	0,92	0,95	**0,98**
	60	0,90	0,96	**1,00**	0,91	0,95	0,99
Sensitivity	20	0,87	0,94	**0,96**	0,86	0,93	0,94
	40	0,89	0,96	**0,99**	0,92	0,95	0,97
	60	0,89	0,96	**1,00**	0,92	0,95	0,99
Specificity	20	0,84	0,93	0,94	0,87	0,94	0,95
	40	0,90	0,96	**0,98**	0,91	0,95	**0,98**
	60	0,90	0,96	**1,00**	0,91	0,95	0,99
F-Measure	20	0,85	0,94	**0,95**	0,86	0,94	0,94
	40	0,91	0,95	**0,98**	0,90	0,96	**0,98**
	60	0,91	0,95	**1,00**	0,90	0,96	0,98

La Tabla 1 muestra un resumen de los diferentes ratios e indicadores derivados de las matrices de confusión. Los mejores resultados en términos de precisión, sensibilidad y especificad, se obtienen utilizando como característica las correlaciones y clasificando en base a un clasificador lineal. Por otro lado, puede observarse que para n=60 (siendo n el número de variables escogidas para la clasificación), los valores de precisión, sensibilidad y especificidad alcanzan el 100% para todas las pruebas realizadas.

Con relación al tipo de clasificador utilizado, se observa que el clasificador cuadrático también obtiene buenos resultados utilizando como característica las correlaciones. Este no es sorprendente y coincide con los resultados en [17], donde ya se indicaba que el clasificador lineal era la mejor opción en su marco de investigación en comparación con otros clasificadores más complejos. Además, el uso del clasificador lineal tiene ciertas ventajas como que garantiza la convergencia y la robustez ante nuevos datos. Además, los modelos lineales son más fáciles de explicar e interpretar.

4.5.3. Análisis de importancia de electrodos

En esta sección se lleva a cabo un pequeño estudio sobre que electrodos son los más importantes para la tarea de clasificación.

El método desarrollado genera un gran número de variables a utilizar para la clasificación pudiendo causar problemas de *overfitting*, siendo necesario un método que seleccione las variables más discriminantes. Dado que cada una de estas variables está vinculada a uno (o dos) electrodos, es interesante analizar qué electrodos están involucrados en la tarea de clasificación, ya que podría indicar que dicho electrodo tiene cierta influencia en la tarea analizada. En este caso, se analiza cuáles de los electrodos seleccionados pertenecen a la corteza motora, la cual se monitorea mediante 18 electrodos (de un total de 63).

Para este análisis se escogen unos valores de configuración del método que ofrece una buena precisión con un número reducido de variables:

- Numero de variables = 20
- Características = Correlaciones
- Filtro = *d6*
- Niveles de descomposición = 12
- Clasificador = lineal

Con estos parámetros el método obtiene una precisión del 97.25% con un número reducido de variables (20). Sin embargo, como cada una de las variables hace referencia a la correlación entre el nivel de descomposición de un electrodo con otro electrodo, cada una de las variables referencia dos electrodos diferentes. Esta configuración, por tanto, presenta un total de 40 electrodos, pero si se eliminan los electrodos repetidos quedan 33 electrodos únicos, de los cuales 9 pertenecen a la corteza motora.

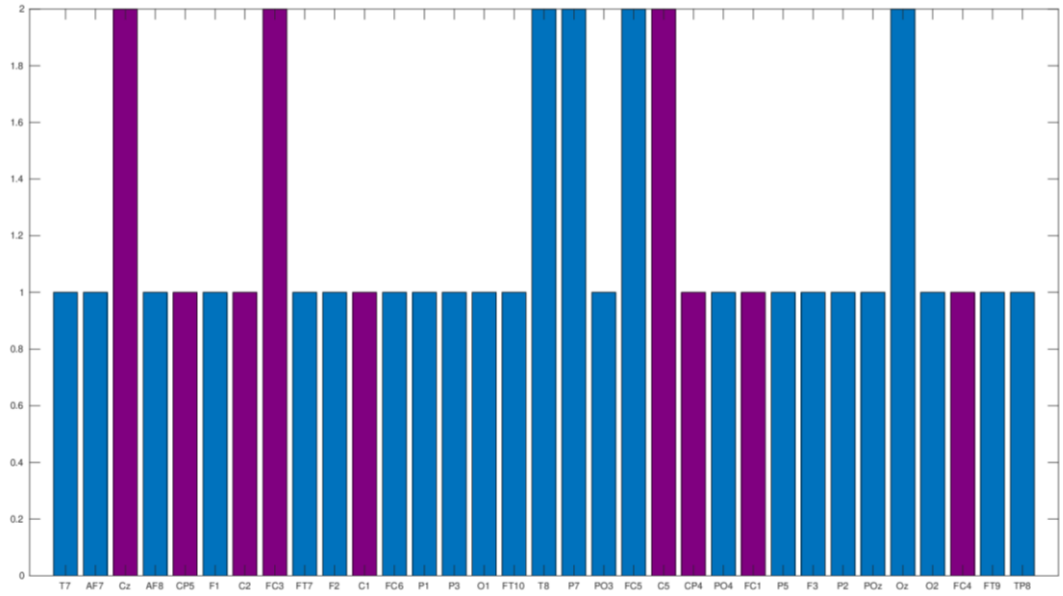


Ilustración 26: Análisis de frecuencia de aparición de los electrodos que intervienen en el cálculo de alguna de las 20 variables más discriminantes seleccionadas por el discriminante por pasos. El color morado representa electrodos no asociados a la corteza motora, mientras que el color azul representa los electrodos que sí están asociados con la corteza motora.

Como puede verse en la Ilustración 26, 7 de los electrodos son más relevantes al estar involucrados en dos correlaciones. De estos electrodos, cuatro no están relacionados con la corteza motora (T8, P7, FC5, Oz), mientras que 3 de ellos (Cz, FC3, C5) sí están relacionados con la corteza motora. Nótese que si únicamente se tienen en cuenta los electrodos relacionados con la corteza motora la precisión cae al 90%.

Con la intención de comparar el rendimiento de los electrodos asociados a la corteza motora con los que no están asociados a esta, y verificar la selección de los electrodos por parte del algoritmo, se ha llevado a cabo un test de diferencias de proporciones significativas. Es necesario tener en cuenta que, por cada experimento, se considera una prueba Bernouilli donde los posibles resultados son acierto (success) o fallo (failure) en la clasificación. Los 400 diferentes experimentos realizados han permitido calcular la precisión media, así como el intervalo de confianza de la proporción correcta para cada caso, considerando un nivel de significación 51eali

= 0,05. La Ilustración 27 muestra la frecuencia de éxito de cada combinación de electrodos. El test de comparación de proporciones concluye que no existen diferencias significativas en la combinación de electrodos relacionados con la corteza motora y no relacionados con esta, con un valor de p = 0.204.

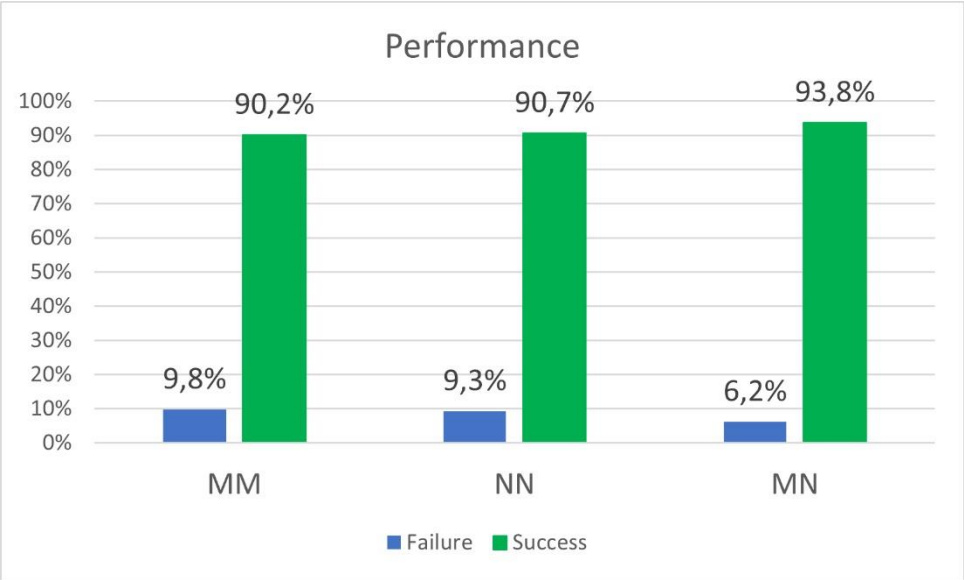


*Ilustración 27: El grafico muestra el porcentaje de aciertos y fallos al clasificar cada uno de los 400 experimentos con una combinación diferente de electrodos: únicamente electrodos de la corteza motora **(MM)**, únicamente electrodos no pertenecientes a la corteza motora **(NN)** y una combinación de estos (**MN)***

4.5.4. Librería de fácil utilización.

Como se ha demostrado, el método consigue buenos resultados en el campo de imaginación de movimiento dentro del ámbito EEG. Además, en la literatura queda demostrado que también es válido para otras tareas dentro del ámbito de EEG y en el ámbito de los ECG, lo que parece indicar cierta capacidad general para tratar con datos de carácter temporal independiente del dominio. Por lo tanto, se ha considerado útil desarrollar una librería que implementase el método desarrollado.

Esta librería tiene como objetivo principal que sea muy sencilla de utilizar para los usuarios no expertos en informática o estadística (pero que sí son expertos en su campo), lo que conlleva algunas decisiones de diseño a la hora de desarrollar la librería. La primera de estas decisiones es el lenguaje de programación. En este caso se escoge R al ser ampliamente utilizado por investigadores de todos los ámbitos para llevar a cabo sus análisis estadísticos. Además, aunque el objetivo principal, son los usuarios no expertos quienes necesitan facilidad de uso, también hay que considerar a los usuarios expertos que busquen personalizar o modificar etapas del proceso.

De cara a los usuarios no expertos, se han desarrollado una serie de funciones de muy fácil utilización que únicamente necesitan los datos y ciertos parámetros sencillos de configuración para llevar a cabo todo el proceso de clasificación. Además, también se proporciona un método que prueba, de forma automática, todas las posibles combinaciones de parámetros para ayudar a los usuarios a escoger los que proporcionen mejores resultados.

Por otro lado, de cara a los usuarios expertos, se han propuesto métodos altamente personalizables que implementan cada una de las etapas del algoritmo por separado. Así, los

expertos pueden, o bien personalizar cada uno de los pasos a su antojo modificando los valores de los parámetros previstos, o directamente sustituir de forma completa una etapa por una etapa suya propia, lo que es muy probable que ocurra con cierta frecuencia en la etapa de clasificación.

Además, también se ha optimizado la propia ejecución del algoritmo por medio de la paralización de varios de los pasos, obteniendo grandes mejoras en el tiempo de ejecución. Sin embargo, el consumo de memoria en el entorno paralelo es muy elevado provocando que sea un limitante en el número de tareas que se pueden llevar a cabo.

4.6. Discusión.

En este capítulo se ha propuesto un método de clasificación de series temporales multivariantes basado en un enfoque de extracción de características lo que lo hace explicable. Este método se ha utilizado con un conjunto de datos proveniente del dominio de EEG para validar su funcionamiento, aunque también ha demostrado ser útil con datos de ECG.

El método ha conseguido muy buenos resultados en la tarea de clasificación, con una precisión cercana al 100%. Aun así, se observan unas diferencias significativas de rendimiento en función de la wavelet seleccionada para realizar la clasificación. En este caso de análisis en concreto, con la wavelet *d4* se ha obtenido mejores resultados que utilizando la wavelet *Haar*. El buen funcionamiento de la wavelet *d4* puede deberse a que la wavelet es casi simétrica y, además, cuenta con 4 *Vanishing Moments*, que son característicos de las señales de EEG. Por el contrario, es posible que el mal funcionamiento de la wavelet *Haar* se deba a su naturaleza no continua, lo que provoca una peor aproximación de las señales de EEG.

Respecto a otros trabajos realizados en el ámbito de la clasificación de señales de EEG para determinar consciencia en el trabajo de Henriques *et al.*[47] se realizó esta misma tarea de clasificación, exactamente con los mismos datos y se obtuvo únicamente un 67% de precisión. Por el contrario, Cruse et al. [95] con un conjunto de datos similar obtuvieron únicamente un 50% de precisión. Por lo tanto, nuestra propuesta ha obtenido unos resultados mucho mejores que los obtenidos hasta el momento para este tipo de datos. Por otra parte, con fines comparativos, se han probado otros métodos de clasificación de series temporales multivariantes con este mismo conjunto de datos utilizado en este trabajo. Concretamente, se ha probado el método de descomposición modal empírica multivariante (*Multivariate Empirical Mode Decomposition*, MEMD) [96], que consiste en extraer las funciones de modo intrínseco para reconstruir una señal de EEG mejorada y, posteriormente, extraer las mejores características por medio de patrones espaciales comunes (*Common Spatial Pattenrs*, CSP) [97]. También se han probado las redes neuronales MLSTM-FNC y MALSTM-FNC que han sido adaptadas al enfoque multivariante por medio de bloques *Squeeze&Excite* [65]. Por último, se ha escogido para la comparación el método de medidas de complejidad y características de las series temporales multivariantes (*Complexity Measures and Features for Multivariate Time Series*, CMFMTS) que consiste en la extracción de características para cada componente individualdonde, seguidamente, estas características son ensambladas formando una representación multivariante que es utilizada con múltiples clasificadores (SVM, *Random forest* (RF) y C5.0).

*Tabla 2: Tabla comparativa de nuestra propuesta frente a otros métodos. La primera fila de la tabla hace referencia al método utilizado para llevar a cabo la clasificación. La segunda fila indica los parámetros de configuración escogidos para cada método, siendo estos para nuestra propuesta **(our proposal)** el número de variables seleccionadas por el discriminante por pasos, en el caso de **FNCs** la red neuronal concreta que se ha utilizado, para **CMFTS** el clasificador seleccionado y **MEMD** solamente se ha aplicado con una configuración.*

Method	**Our proposal**			**FNCs**				**CMFMTS**			**MEMD**
Configuration	**20**	**40**	**60**	**LSTM**	**ALSTM**	**MLSTM**	**MALSTM**	**C5.0**	**RF**	**SVM**	**MEMD**
Accuracy	0.95	0.98	1	0.86	0.82	0.71	0.78	0.89	0.9	0.79	0.73

Los resultados de la comparativa con todos estos métodos se muestra en la Tabla 2. Como se puede observar, el método propuesto supera en precisión a todos los métodos con una mejora de entre un 10 y un 20% dependiendo de la configuración seleccionada. Sin embargo, lo más importante es que, además de obtener un buen rendimiento, el método propuesto es el más interpretable de todos. Esto se debe al bajo número de variables que necesita para obtener buenos resultados de clasificación, y a que cada una de esas variables está ligadas a un rango de frecuencias concreto de un electrodo, o a la relación entre el mismo rango de frecuencias de dos electrodos diferentes. Por el contrario, los métodos basados en *54eal-learning* son muy difíciles de interpretar, y aun con los esfuerzos de Baldan *et al.* [68] para lidiar con esta falta de interpretabilidad su método tiene en cuenta demasiadas variables para la clasificación (2584 contra las 20-60 de nuestro método).

Por otra parte, nuestro método es el más eficiente computacionalmente. Se necesita únicamente 1 día para probar todas las posibles combinaciones de parámetros para determinar cuál se ajusta mejor, y únicamente unos minutos cuando ya se tiene seleccionada la configuración optima. Por el contrario, MEMD requirió 4 días de procesamiento, mientras que las alternativas en *54eal-learning* tardaron un tiempo aproximado de 16 horas. Sin embargo, esta aproximación requiere tarjetas gráficas de alto rendimiento para poder ejecutarse. Concretamente se han utilizado 2 Nvidia 1080 ti y ha sido necesario reducir el tamaño de los *batches (*la cantidad de datos procesados a la vez*)* debido a limitaciones de memoria.

Una de las desventajas del método es que es muy complicado saber a priori qué combinación de parámetros va a obtener buenos resultados (wavelet, características y número de variables) y que el número de posibles combinaciones de parámetros son bastante elevadas. Aunque se puede obtener estos valores de configuración óptimos por medio de fuerza bruta, es posible que para datos con muchas componentes o con muchas observaciones el coste computacional sea demasiado elevado. Esto se debe principalmente al paso de selección de las variables más discriminantes que tiene un coste cuadrático en función del número de variables. Además, la inclusión de nuevas características también incrementaría sustancialmente el tiempo de ejecución al aumentar considerablemente las variables calculadas. Sin embargo, estas limitaciones únicamente aplican cuando se necesita obtener la mejor configuración de parámetros para un problema de clasificación determinado.

Debido a las grandes ventajas que aporta el método en la clasificación de series temporales multivariantes se decidió desarrollar una librería desarrollada en R de fácil utilización que implementa el método propuesto para ser utilizada por la comunidad. Se decidió realizar una

publicación científica sobre esta librería pero lamentablemente el artículo fue rechazado, aunque se encuentra en proceso de revisión para volver a ser enviado.

5. Visualización y análisis exploratorio de datos multivariantes temporales heterogéneos

En esta sección se abordará el diseño e implementación de una herramienta de visualización que permita tanto la visualización, como la realización de análisis exploratorio de datos de carácter temporal (que incluyen series temporales y procesos longitudinales).

Los datos de carácter temporal son un tipo de datos muy importantes que tienen una serie de problemáticas propias. Este tipo de datos se encuentra en multitud de campos diferentes, por lo que es muy importante brindar herramientas de análisis y visualización que ayuden a comprender y extraer conocimiento de estos datos. Aunque que las herramientas de análisis son muy útiles para comprobar hipótesis realizadas a priori, no siempre facilitan la exploración de los datos para descubrir nuevas características que pudieran conducir a plantear nuevas hipótesis. Para realizar esta tarea de descubrimiento de los datos es más apropiada la visualización, concretamente la visualización y el análisis exploratorio. El análisis exploratorio se centra en la idea de que cuanto mejor se conozcan los datos, más sencilla será la extracción de conocimiento a partir de estos, así como la comprobación de hipótesis. En este proceso de conocimiento de los datos se suelen utilizar resúmenes estadísticos, búsqueda de tendencias y formulación y comprobación de nuevas hipótesis [98], [99]. Para conseguir todos estos objetivos, el análisis exploratorio aprovecha las fortalezas del sistema visual humano que tiene la capacidad de detectar patrones, tendencias, grupos, etc.

Como se vio en la sección 2.1, pese a haber varios trabajos en esta línea, muchas de las herramientas han quedado obsoletas y otras carecen de características que son deseables para la visualización exploratoria, como son la facilidad de uso, la rapidez en la interacción, el permitir a los usuarios agrupar de forma sencilla los datos en distintos grupos, el poder visualizar en el momento las tendencias de los grupos, o la reactividad.

Además, se desea que la herramienta diseñada se base en tecnologías actuales, pueda ser modular y extensible, con el fin de permitir ampliar su funcionalidad. Otro requisito imprescindible es que pueda colaborar con otras herramientas existentes o futuras.

La herramienta debe tener un enfoque genérico de forma que se pueda usar con diferentes tipos de datos heterogéneos que provengan de distintos dominios.

En base a estos requisitos, se toman las siguientes decisiones de diseño para la herramienta:

- Ofrecer un entorno de visualización sencillo
- Permitir el filtrado de datos en función de parámetros temporales y cuantitativos

- Ofrecer facilidad en la creación de grupos sobre la propia visualización, con solo unos clicks de ratón
- Mostrar información sobre los grupos creados y que se puedan ver sus tendencias, compararlos o exportarlos.
- Permitir a los usuarios la exploración de nuevas hipótesis de forma rápida e intuitiva, pudiendo realizar cambios en tiempo real.
- Alta personalización: la herramienta debe ser altamente configurable para adaptarse a datos temporales de distintos ámbitos, así como a tareas de análisis diferentes.
- Altamente escalable, permitiendo manejar de forma interactiva conjuntos de datos con cientos de atributos de distintos tipos, centrándose en los temporales, cuantitativos y categóricos, a la vez que gestiona miles de registros.
- La herramienta debe comportarse como un componente reutilizable y reactivo de forma que pueda utilizarse en conjunción con otras herramientas.
- Permitir la interacción con múltiples variables cuantitativas de forma simultánea.

A continuación, se describirán algunas de las características principales de la herramienta, de modo que sea fácil comprender su funcionamiento para, en secciones posteriores, detallar los distintos aspectos.

La herramienta, para conjuntos de datos temporales, conocida como *TimeSearcher+*, permite que el usuario elija cualquier variable y, para cada registro, dibuja una polilínea que muestre la evolución de los valores a lo largo del tiempo. Dichas líneas se pueden dibujar con cierto nivel de transparencia para que puedan observarse tendencias generales cuando se representen grandes cantidades de datos.

Por otro lado, para conseguir la funcionalidad de la creación de grupos, se utiliza una técnica de manipulación directa basada en el concepto de *TimeBox* [93]. Una *TimeBox* se define como una caja contenedora (*bounding box)* que selecciona todas las polilíneas que intersecan con ella en algún momento. Además, la herramienta permite crear conjuntos asociados de varias *TimeBoxes* de tal manera que se puedan hacer grupos que no se basen en valores en un único instante o intervalo de tiempo (de una única *TimeBox*), sino que se harían grupos con aquellos registros que intersecan con todas las cajas contenedoras de un mismo conjunto asociado, por lo que un usuario podría seleccionar datos en función de sus tendencias. Un ejemplo se muestra en la Ilustración 30, donde el conjunto de cajas contenedoras asociadas coloreadas en azul incluye a aquellos registros con valores bajos al principio que luego se recuperan hasta llegar a valores alrededor de la media o superiores. Por su parte, el conjunto de cajas contenedoras asociadas naranja representa aquellos registros que comienzan con valores bajos y siguen manteniendo valores bajos a lo largo del tiempo. De este modo, es posible seleccionar registros en función de la tendencia de los valores, así como comparar de forma sencilla diferentes grupos.

Las cajas contenedoras se pueden definir haciendo click con el ratón y arrastrando para crear un rectángulo en un primer paso y, cuando sea necesario, ajustando al valor exacto utilizando unos cuadros de texto (ver Ilustración 28). Los colores y tamaños de las cajas, así como el número de cajas asociadas en cada conjunto es configurable en función de las necesidades del usuario. Además, la herramienta también ofrece líneas de referencia y las medias de cada grupo para facilitar la comparación. En la Ilustración 30 puede observase un ejemplo de la interfaz con todas las partes marcadas.

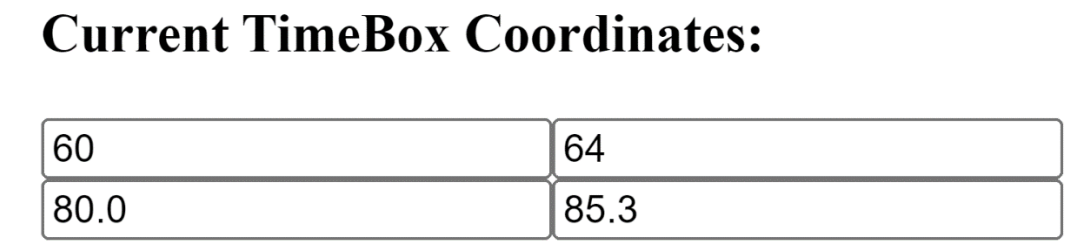


Ilustración 28: Ejemplo de los cuadros de texto utilizados para ajustar la posición de las TimeBoxes

También es posible utilizar la herramienta con múltiples variables cuantitativas a la vez. Para esta funcionalidad se ha seguido el enfoque multivariante, de tal forma que la herramienta deberá comportarse como una única entidad que permite interactuar con varias variables a la vez (ver Ilustración 29). De esta manera se tiene una única interfaz común para todas las variables, que comparten la definición de grupos, de tal forma que se pueda hacer un filtrado progresivo de los datos en estas variables. Por otro lado, también se pueden resaltar los registros seleccionados a partir de una variable sobre la que se han realizado *TimeBoxes*, en otra variable diferente, permitiendo así ver la selección desde múltiples perspectivas.

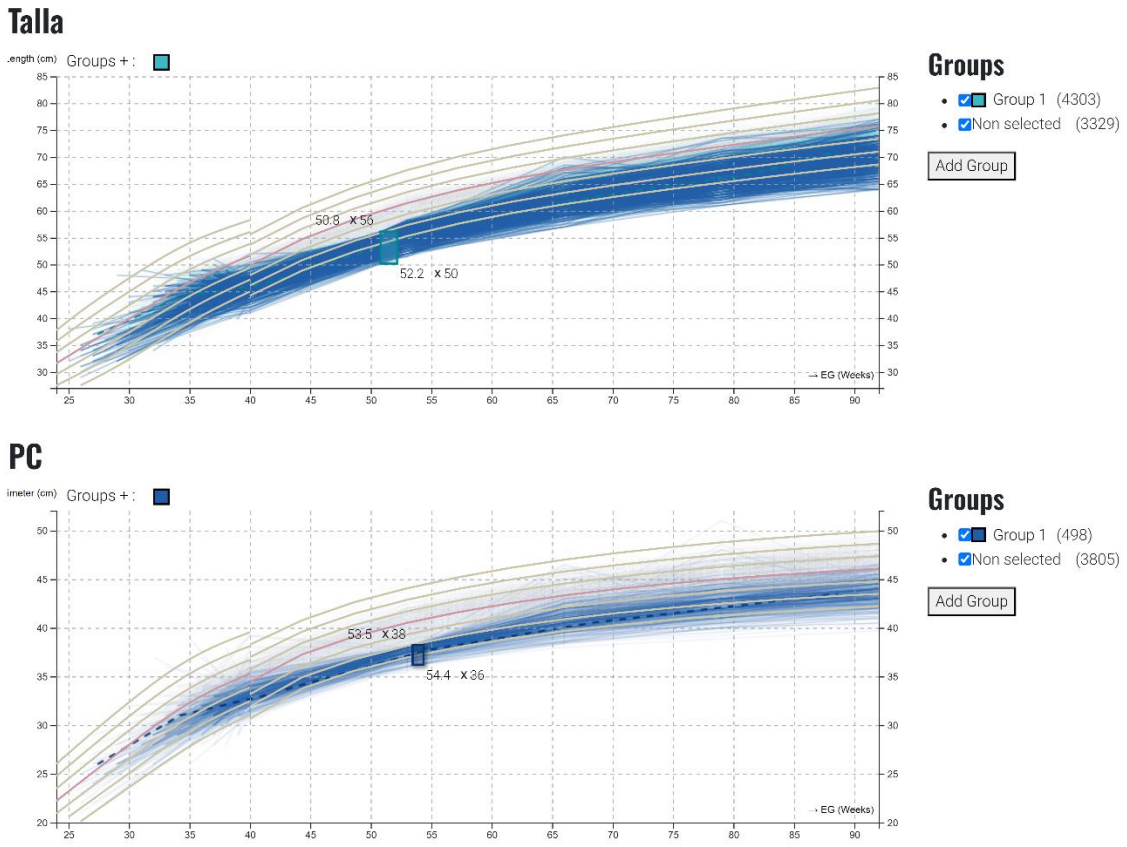


Ilustración 29: Ejemplo de utilización de la herramienta con varias variables a la vez (Talla y perímetro craneano) donde además se muestra la funcionalidad de resaltado entre variables. Concretamente se están resaltando en la variable Talla los elementos seleccionados en función de la variable PC..

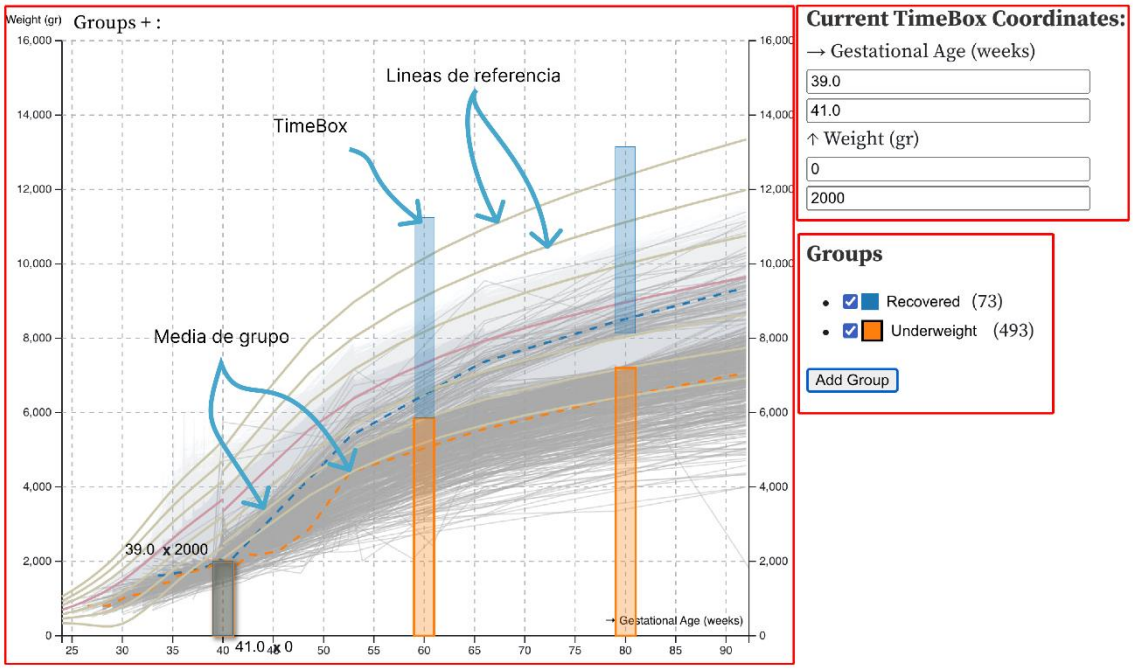


Ilustración 30: Ejemplo de la visualización propuesta para la herramienta de análisis de datos de carácter temporal con un fuerte enfoque en la definición de grupos (TimeSearcher+). Esta visualización corresponde a valores del peso de bebés prematuros a lo largo del tiempo. El usuario ha creado dos conjuntos de cajas contenedoras asociadas que, a su vez, determinan dos grupos. El grupo azul representa a bebés con peso bajo a las 40 semanas que lograron recuperarse a medida que pasaba el tiempo. El grupo naranja, por su parte, incluye a los bebés prematuros que se mantuvieron con peso bajo a lo largo del tiempo.

Una vez introducida la herramienta, las siguientes secciones detallan a más bajo nivel la gran variedad de posibilidades en cuanto a la gestión de grupos, así como las técnicas utilizadas para optimizar su rendimiento. Después, se explica el proceso seguido para renderizar toda la interfaz y la gran cantidad de datos que se manejan. Seguidamente, se detallará el proceso seguido para hacer la aplicación modular y reactiva, lo que le permite integrarse con otras herramientas de forma sencilla, aumentando así su utilidad. Por último, se ve un caso especial en el que la herramienta es capaz de visualizar e interactuar con múltiples variables cuantitativas a la vez.

5.1. Selección de grupos

Como se ha comentado anteriormente, la herramienta requiere un enfoque muy basado en la creación de grupos, así como una alta interactividad. Para soportar esta interactividad, el usuario tiene la capacidad en cualquier momento de crear nuevas *TimeBoxes* asociadas a un determinado grupo, así como eliminarlas, o modificarlas. Para crear una *TimeBox*, se presentan dos posibilidades. Por un lado, el usuario puede utilizar el método de manipulación directa con el ratón que tiene la ventaja de ser muy intuitivo y rápido, pero tiene la desventaja de ser poco preciso. Por otro lado, puede utilizar la entrada por teclado en dos interfaces diferentes, una que está incluida dentro de la propia visualización que se descubre al seleccionar una *TimeBox*, o bien con un widget aparte (que además es personalizable como se verá en la sección 5.3). Así, el usuario pincha con el ratón donde quiere situar uno de los extremos de la *TimeBox* y desplaza el ratón en función del conjunto de datos que quiere incluir. Alternativamente, puede introducir los valores exactos de las esquinas de dicha caja. Una vez creada una *TimeBox*, para modificarla puede hacerlo con el ratón, ya sea cambiando el tamaño de la *TimeBox* pinchando sobre el borde de la caja o bien manteniendo el tamaño de la *TimeBox,* pero eligiendo desplazarlo y moverla

sobre el conjunto de datos. Por otro lado, es posible crear conjuntos de *TimeBoxes* asociadas en donde todas las *TimeBoxes* de ese conjunto se representan con un mismo color. Por ejemplo, las *TimeBoxes* azules pertenecen todas al mismo conjunto asociado de *TimeBoxes* (ver Ilustración 30). Estos conjuntos asociados de *TimeBoxes* pueden crearse por medio del botón "+" situado en la parte superior izquierda de la visualización o bien utilizando el *Widget* de gestión de conjuntos asociados a *TimeBoxes* que ofrece funcionalidades adicionales como eliminación completa del conjunto, cambiar el nombre del conjunto, o activar y desactivar los conjuntos. Además, en todo momento se muestra información sobre el número de registros o elementos seleccionados por el conjunto de *TimeBoxes* asociadas.

La selección de grupos de registros se apoya en los conjuntos de *TimeBoxes* de tal forma que cada conjunto de *TimeBoxes* seleccionará un grupo de registros. Esta selección depende del número de *TimeBoxes* que tenga el conjunto. En caso de que un conjunto solamente tenga una *TimeBox,* los registros seleccionados serán todos aquellos que intersequen en algún momento con la caja. Por el contrario, si el conjunto de *TimeBoxes* tiene más de una caja, la selección consistirá en los registros que intersequen con todas las cajas, generando por tanto una consulta de tipo *AND.* Por otra parte, la aplicación también debe de ser capaz de mostrar el resultado de la interacción con estos sistemas en tiempo real (y enviar el resultado, junto con datos de estado, a otras aplicaciones). Por lo tanto, es necesario disponer de una alta escalabilidad lo que obliga al uso de técnicas de optimización para que la aplicación presente un buen rendimiento aun cuando se está interactuando con una gran cantidad de datos.

Para conseguir esta alta escalabilidad deseada es necesario un sistema que tenga que recalcular el menor número de elementos posibles al realizar alguna modificación. Con esta idea en mente, cada una de las *TimeBoxes* almacena los elementos con los que interseca de manera individual, de tal forma que para calcular los elementos que pertenecen a un cierto grupo sea necesario calcular los elementos que están presentes en todas las *TimeBoxes* del grupo. Sin embargo, este enfoque tiene la ventaja de que al modificar una *TimeBox* únicamente es necesario recalcular sus intersecciones, no siendo necesario recalcular las intersecciones de todas las *TimeBoxes* del grupo lo cual sería muy costoso.

Aun teniendo en cuenta que el algoritmo únicamente necesita recalcular las intersecciones de la *TimeBox* que se modifica, este proceso de cálculo de intersecciones puede ser demasiado costoso para un número grande de elementos. Por lo tanto, se ha propuesto una técnica de optimización basada en partición espacial que proporciona un buen rendimiento en el cálculo de intersecciones como se describirá a continuación.

5.1.1. Partición espacial

El método escogido para acelerar el cálculo de las intersecciones entre todas las polilíneas y las *TimeBoxes* se basa en los métodos de partición espacial, que consisten en dividir el espacio con la intención de comprobar únicamente las regiones de interés y no en todo el espacio completo.

El método implementado consiste en subdividir el espacio en celdas de un tamaño constante que seguidamente serán inicializadas con las polilíneas que crucen esas celdas, con la intención de comprobar las colisiones únicamente con los segmentos de polilínea que se encuentren cercanas en el espacio a la *TimeBox* a comprobar. En la Ilustración 31 puede verse un ejemplo del funcionamiento del algoritmo.

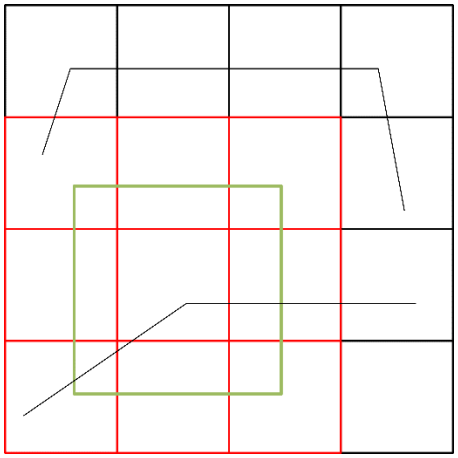


Ilustración 31: En la imagen puede verse un ejemplo del método de partición espacial para una división de 4 tanto en vertical como en horizontal. Se puede observar cómo, gracias al método de división, únicamente es necesario comprobar las colisiones con los segmentos contenidos en las celdas rojas (el cuadrado verde la TimeBox seleccioda), no siendo necesario comprobar los segmentos del resto de celdas.

Este algoritmo de partición tiene dos partes bien diferenciadas: por un lado, la construcción del espacio particionado que rellene todas las celdas con los segmentos que las atraviesan; por otro lado, el cálculo de qué celdas intersecan con una *TimeBox* dada. Nótese que el proceso de construcción del espacio particionado es sumamente costoso, pero únicamente es necesario realizarlo como preproceso al iniciar la herramienta y, a cambio, se obtiene una consulta de las intersecciones mucho más rápida.

5.1.2. Construcción del espacio particionado.

El primer paso para utilizar el algoritmo de partición espacial para la detección de colisiones consiste en construir el espacio particionado. Este espacio está constituido por celdas con un tamaño constante, siendo configurable el número de celdas que tiene el espacio tanto en vertical como en horizontal (de forma separada). Una vez se han creado e inicializado todas las celdas presentes en el espacio, se procede a rellenar cada una de las celdas con las polilíneas (datos) que las atraviesan.

Antes de comenzar a rellenar del espacio particionado, es necesario tener en cuenta que calcular la celda a la que pertenece un punto cualquiera es una tarea trivial gracias a que es un particionado uniforme. De esta forma, para calcular la celda a la que pertenece un punto simplemente es necesario dividir cada una de las coordenadas del punto, por el tamaño de cada una de las celdas despreciando la parte decimal, obteniendo así el índice de la celda en los dos ejes ($xIndex = floor(CordenadaX \,/\, TamañoX)$). Ahora que se conoce cómo se puede mapear un punto a su respectiva celda, se detallará el proceso de rellenado del espacio particionado. Este proceso se realiza de forma independiente por cada polilínea de los datos de entrada. Por cada polilínea, el primer paso consiste en añadir el primer punto de esta a su celda correspondiente, seguidamente se trata cada uno de los segmentos que componen la polilínea pudiéndose dar dos casos:

- Que el punto final del segmento pertenezca a la misma celda que el punto inicial, en este caso, simplemente se añade el punto a la celda.
- Que el punto final del segmento pertenezca a una celda distinta a la del punto inicial. En este caso es necesario tener en cuenta que la celda a la que pertenece el punto final no tiene por qué ser contigua a la celda del punto inicial, y que es necesario que todas las celdas entre ambas almacenen ese segmento. Para lidiar con este caso, el algoritmo

calcula con qué celdas interseca el segmento a tratar, añadiendo el segmento completo a cada una de las celdas. Nótese que el cálculo de la intersección entre el segmento y la celda está muy optimizado al estar la celda alineada con los ejes.

En la Ilustración 32 puede verse un pequeño ejemplo del resultado del espacio particionado al añadir una polilínea a este.

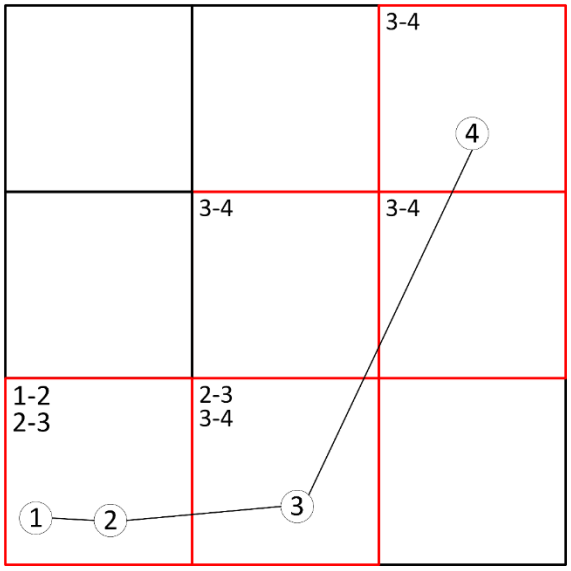


Ilustración 32: En la imagen puede observarse cómo se produce la inicialización del espacio particionado. Las celdas marcadas en rojo son las celdas que son atravesadas por la polilínea definida. Además, en la esquina de cada celda pueden verse los segmentos que contiene, siendo estos los segmentos para los que se deberá realizar la comprobación de si la TimeBoxinterseca con ellos en caso de que la TimeBox interseque con la celda.

5.1.3. Detección de colisiones

Una vez el espacio particionado ha sido creado y rellenado se puede proceder a realizar las consultas sobre dicho espacio, lo que requiere del que requiere el algoritmo de detección de colisiones entre las *TimeBoxes* y las polilíneas de los datos de entrada.

Para hacer una de estas consultas de intersección, el primer paso consiste en calcular con qué celdas interseca una *TimeBox* dada. Para realizar este proceso simplemente se calcula a qué celdas pertenece cada una de las esquinas de la *TimeBox*; de esta forma se delimita un cuadrado formado por las celdas que intersecan con la *TimeBox*. Seguidamente, se comprueban las colisiones únicamente de los segmentos contenidos en esas celdas con la *TimeBox*, obviando el resto de los segmentos, ya que no es posible que produzcan intersección. Un ejemplo de esto puede verse en la Ilustración 33.

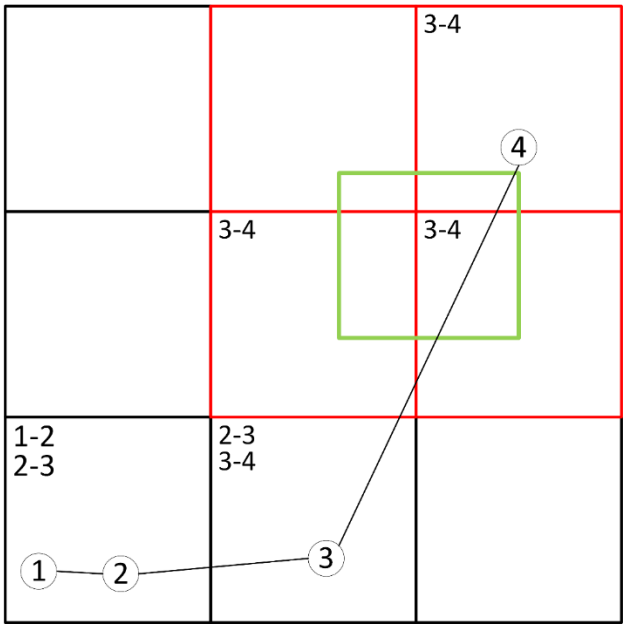


Ilustración 33: En la imagen puede observarse un ejemplo de funcionamiento del algoritmo para un espacio particionado ya inicializado. La TimeBox (en verde) interseca con 4 celdas (marcadas en rojo) y únicamente debe comprobar los segmentos que están contenidos en ella, que en este caso es el segmento 3-4. De esta forma únicamente es necesario comprobar las colisiones de la TimeBox con un segmento y no con los 3 segmentos que conforman la polilínea.

Respecto al algoritmo utilizado para calcular la intersección entre los segmentos correspondientes a los valores de los elementos o registros y las *TimeBoxes,* se ha diseñado teniendo en cuenta que las *TimeBoxes* siempre son paralelas a los ejes, pudiendo de esta forma optimizarlo para este caso en concreto. El algoritmo consiste en comprobar si el segmento de un registro interseca con alguno de los 4 segmentos delimitadores que componen la *TimeBox*. Para realizar esta comprobación el primer paso consiste en mirar si los puntos inicial y final del segmento de un registro se encuentran a ambos lados del segmento delimitador, ya que de lo contrario no puede existir colisión. En caso de que se encuentren a ambos lados, se procede a calcular el punto de intersección entre una recta infinita definida por segmento delimitador y el segmento del registro. Seguidamente se comprueba si este punto se encuentra sobre del segmento delimitador. Si este es el caso, el segmento del registro interseca con el segmento delimitador y, por tanto, con la *TimeBox*. Nótese que este es el mismo proceso utilizado para calcular las intersecciones con las celdas en el proceso de relleno del espacio particionado.

5.2. Proceso de renderizado

Otro de los aspectos importantes para la herramienta es cómo llevar a cabo el proceso de renderizado de la forma más eficiente posible, ya que el gran número de elementos a mostrar en pantalla puede provocar una degradación del rendimiento de forma severa si no se realiza de forma correcta.

Para intentar paliar estos problemas se ha optado por un enfoque híbrido que combina elementos que son considerados estáticos, es decir no van a cambiar en el transcurso de la aplicación, y elementos considerados dinámicos que presentaran gran cantidad de cambios mientras el usuario interactúe con ellos.

Además, también es necesario tener en cuenta que se dispone de toda una vista de detalle (opcional) que implica tener una gráfica completamente independiente para cada uno de los

datos que estén seleccionados. Esta gran cantidad de gráficas pueden sobrecargar fácilmente el navegador por la gran cantidad de código HTML que se requiere.

En el siguiente apartado se explicará en detalle cada parte del proceso de renderizado por separado.

5.2.1. Renderizado de la interfaz (estático)

Como se ha comentado anteriormente, hay partes de la visualización que nunca van a cambiar (o que sus cambios son muy esporádicos); como, por ejemplo, todos los elementos que facilitan la interacción, pero no así los datos seleccionados fruto de esa interacción, que sí serian dinámicos. Por lo tanto, para este tipo de elementos se ha decidido utilizar para su renderizarlo elementos SVG de HTML con la ayuda de D3 para la creación y actualización de estos elementos.

Los elementos principales que son considerados estáticos por la aplicación son los ejes de coordenadas, y sobre todo las *TimeBoxes* que se apoyan en una funcionalidad de D3 conocida como *brush.* Esta funcionalidad que permite hacer rectángulos de selección y proporciona toda la parte de interacción con el usuario, de tal forma que cada *TimeBox* está representada por un *brush.* Aunque la funcionalidad de *brush* también contempla la selección, esta no se ha usado al estar limitada a la selección de puntos. Para el renderizado de los ejes de coordenadas, se hace uso de una de las funcionalidades de D3 que es capaz de generar cada uno de estos ejes en base al número de píxeles disponibles para la visualización, así como al rango de los datos. Sin embargo, esta funcionalidad ha sido modificada para añadir una cuadrícula visual que ayude a identificar los valores de las líneas. Para ello, se han seleccionado las marcas generadas por D3 en los propios ejes de coordenadas y, a partir de ellas se han añadido líneas, verticales u horizontales según corresponda, formando así la cuadrícula.

Respecto al renderizado de los rectángulos que modelan y proporcionan la interacción de las *TimeBoxes con el usuario*, aunque su funcionalidad se apoye en los *brushes* proporcionados por D3, en la funcionalidad original de D3 únicamente se puede tener un *brush* por visualización. Debido a esta limitación, y puesto que nuestro diseño requiere que pueda haber más de un *TimeBox* por visualización toda la gestión de renderizado de los *brushes* se hace de forma externa con el objetivo de tener un número ilimitado de *brushes* funcionales en la visualización.

El primer paso para comprender cómo se gestiona el renderizado de los *brushes* es tener en cuenta que estos se componen de dos elementos principales: un elemento *overlay* que ocupa toda el área de visualización y tiene la funcionalidad de crear un nuevo *brush* al hacer la acción de arrastrar, y un elemento *selection* que representa el cuadrado de selección formado por el *brush.* Por lo tanto, para tener varios *brushes* activos, el primer paso consiste en poder generar más de uno sin que sus componentes *overlay* entren en conflicto. Para ello, la aplicación siempre mantiene disponible un *brush* sin inicializar (es decir, un *brush* que no tiene elemento *selection* debido a que todavía no se ha realizado la acción de arrastrar), de tal forma que será el que se use para añadir una nueva *TimeBox*. Por el contrario, todos los *brushes* en los que se haya realizado una selección, tendrán su componente *overlay* con los eventos desactivados para que no generen conflicto. Sin embargo, en este punto todavía existe un conflicto con el elemento *overlay* del *brush* sin inicializar (que será el que se use para añadir una nueva *TimeBox*) y los elementos selección (*selection)* de los *brushes* inicializados que están “detrás” del elemento *overlay* y, por lo tanto, no se puede interactuar con ellos. Para solucionar este problema, se ha decidido utilizar la propiedad *tab-index,* que permite especificar qué elementos están por “encima” de otros. De esta forma, los elementos selección (*selection)* siempre se encuentran

por encima del *overlay.* En la Ilustración 34 puede verse un ejemplo del código HTML generado siguiendo este enfoque para un total de 2 *brushes* inicializados.

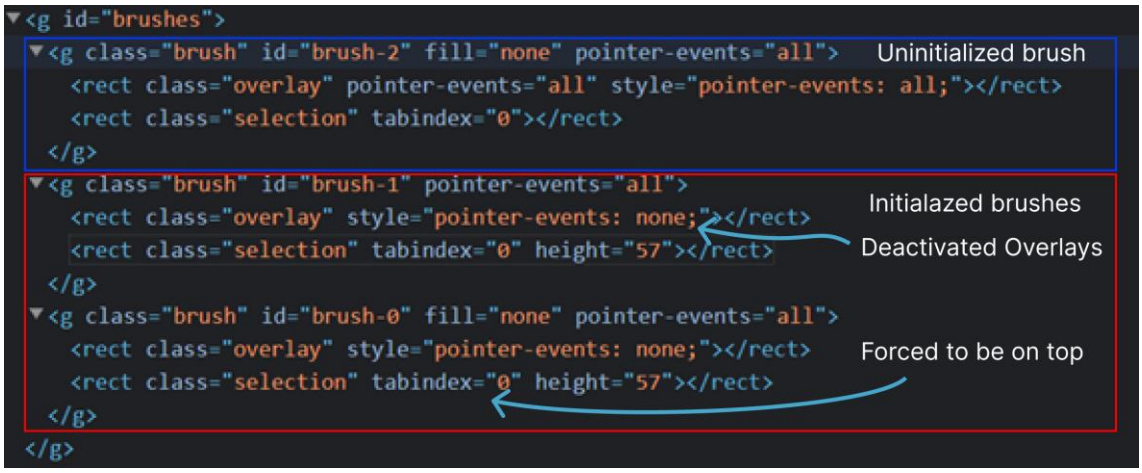


Ilustración 34: En la ilustración puede verse un fragmento de código HTML de la aplicación para un total de 2 brushes inicializados (sección roja), además del brush sin inicnializar (sección azul). Por otro lado, también puede verse un esquema señalando las propiedades relevantes de estos elementos.

Por otra parte, como inicialmente la funcionalidad de D3 soportaba solamente un *brush,* ha sido necesario añadir todos los elementos visuales para poder trabajar con más de un *brush* de manera efectiva. Con este objetivo en mente, se han añadido colores a los *brushes* para identificar a qué conjunto asociado pertenecen; una sombra para identificar el *brush* con el que se está interactuando; y un borde de líneas discontinuas para selección múltiple. Además, para indicar qué *brushes* pertenecen al conjunto activo (siendo los *brushes* del conjunto activo los únicos que permiten interacción), se ha utilizado el ancho del borde del *brush.* La Ilustración 35 muestra un ejemplo de todas las claves visuales utilizadas.

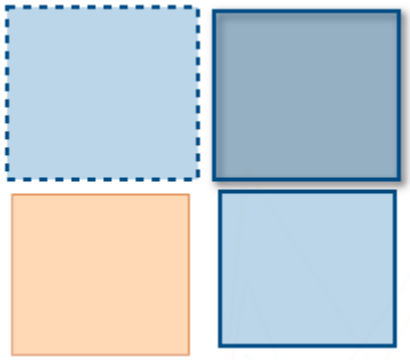


Ilustración 35: En la Ilustración puede observarse un ejemplo brushes indicando todas sus claves visuales: Color—Conjunto; Borde discontinuo – Selección; Ancho borde ampliado – Pertenencia al conjunto activo; Sombra – Brush con el que se está interactuando.

Además, también deben existir elementos que permitan interactuar con el sistema de conjuntos y modificar las coordenadas de las *TimeBoxes* utilizando el teclado para obtener más precisión. Respecto al sistema de conjuntos, se dispone de dos elementos que permiten interactuar con él. Por un lado, una vista integrada en la visualización que permite funciones básicas y, por otro, un elemento más complejo que ofrece todas las funcionalidades (ver Ilustración 36). Nótese que pueden utilizarse ambas a la vez y que se encuentran coordinadas. Respecto a modificar las coordenadas de las *TimeBoxes* de manera precisa por teclado, se proporcionan también dos

posibilidades. La primera de ellas está integrada en la propia visualización. Consiste en mostrar las coordenadas de las esquinas de la *TimeBox* con la que se está interactuando, permitiendo modificarlas. La segunda posibilidad es acceder a un elemento externo que tiene la misma funcionalidad, aunque los cambios se aplican de manera automática (ver Ilustración 37).

Ilustración 36: En la imagen pueden verse las dos formas de interactuar con el sistema de conjuntos. A la izquierda la vista integrada que permite funciones básicas (cambiar conjunto activo, añadir nuevo conjunto). A la derecha la vista compleja que permite: cambiar conjunto activo, cambiar nombre del conjunto, habilitar o deshabilitar conjuntos y eliminar conjuntos.

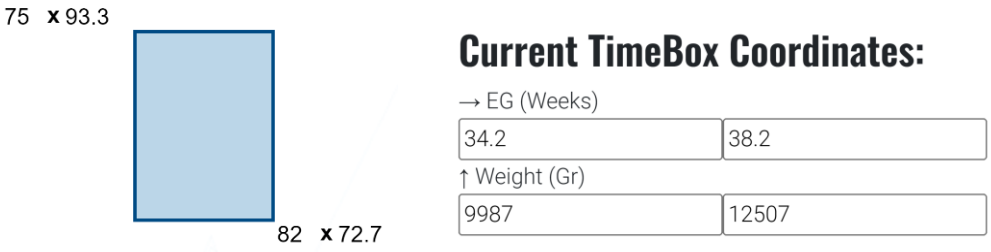


Ilustración 37: En la imagen puede verse a la izquierda la vista integrada en la propia visualización para modificar las coordenadas de una TimeBox, modificándolas directamente introduciendo los datos en las esquinas de la *TimeBox. A la derecha puede verse una vista "externa" que tiene la misma funcionalidad, aunque en esta vista los cambios se aplican automáticamente*.

Otro de los elementos que entran dentro del renderizado estático son las líneas de referencia, que son proporcionadas por el usuario. Estas líneas se dibujan una única vez en el SVG de tal manera que queden siempre por encima de las líneas de los elementos (que forman parte del renderizado dinámico), y por debajo de las *TimeBoxes*.

5.2.2. Renderizado de los datos (dinámico)

Como se ha visto en la sección anterior, toda la parte de interacción e "interfaz" de la aplicación se renderiza de forma estática por medio de SVG y D3. Sin embargo, a la hora de visualizar los datos en sí, este enfoque no es viable debido a problemas de rendimiento del navegador a la hora de interpretar el HTML generado. Esto se debe a que cada "línea" es un elemento HTML, y, por lo tanto, el DOM se satura con relativa facilidad.

Para solucionar estos inconvenientes, para dibujar la parte relativa a los datos se ha optado por utilizar *canvas*, un elemento HTML que permite dibujar gráficos de manera eficiente gracias a la aceleración por hardware (uso de la GPU) por medio de código *JavaScript,* de tal forma que se necesita un único elemento para todos los elementos gráficos.

El primer paso para el proceso de dibujado consiste en pre-generar la información relativa a las líneas 2D ("*path2D*") que van a ser pintadas al comienzo de la ejecución, agilizando así el proceso de dibujado posterior. Una vez se dispone de toda la información relativa a las líneas a dibujar pre-calculadas, se procede al proceso de dibujado en sí, pudiendo distinguirse dos casos: que no exista ninguna selección activa, o que sí exista selección. En el caso de que no exista ninguna selección activa, se dibujan todos los datos con un color y una transparencia por defecto. En caso de que exista una selección, los datos se dividen en dos: los seleccionados y los no seleccionados. Los elementos no seleccionados se dibujan primero utilizando un color y

transparencia para este grupo. Por el contrario, los datos seleccionados son dibujados con un color en base al grupo al que pertenecen y una transparencia específica. Este proceso se repite de forma completa cada vez que se produce una modificación en el estado de la aplicación que implique re-dibujar la sección de datos (cambios en la selección, desactivación de un grupo, etc.).

Otro de los elementos que se dibuja en el *canvas* son las líneas que indican las medias de los grupos. De esta forma, estas líneas se dibujan discontinuas, más anchas que el resto y, sobre todo, en último lugar, para que queden por encima del resto de líneas de los datos.

Nótese que, aunque a nivel conceptual el renderizado de los datos es bastante sencillo, la complejidad viene dada por la necesidad de un muy buen rendimiento, ya que es necesario dibujar una gran cantidad de datos en tiempo real sin mermar la interactividad. Para conseguir este rendimiento la parte más importantes es pre-calcular las líneas a dibujar. Sin embargo, también se han realizado varias pruebas para escoger el orden optimo a la hora de dibujar para maximizar el rendimiento minimizando los cambios de contexto.

5.2.3. Vista de detalle

Aparte de la vista general que permite interactuar con todos los datos, también se dispone de una vista de detalle opcional, que permite visualizar cada uno de los elementos seleccionados por separado. Un ejemplo de esta vista puede verse en la Ilustración 38.

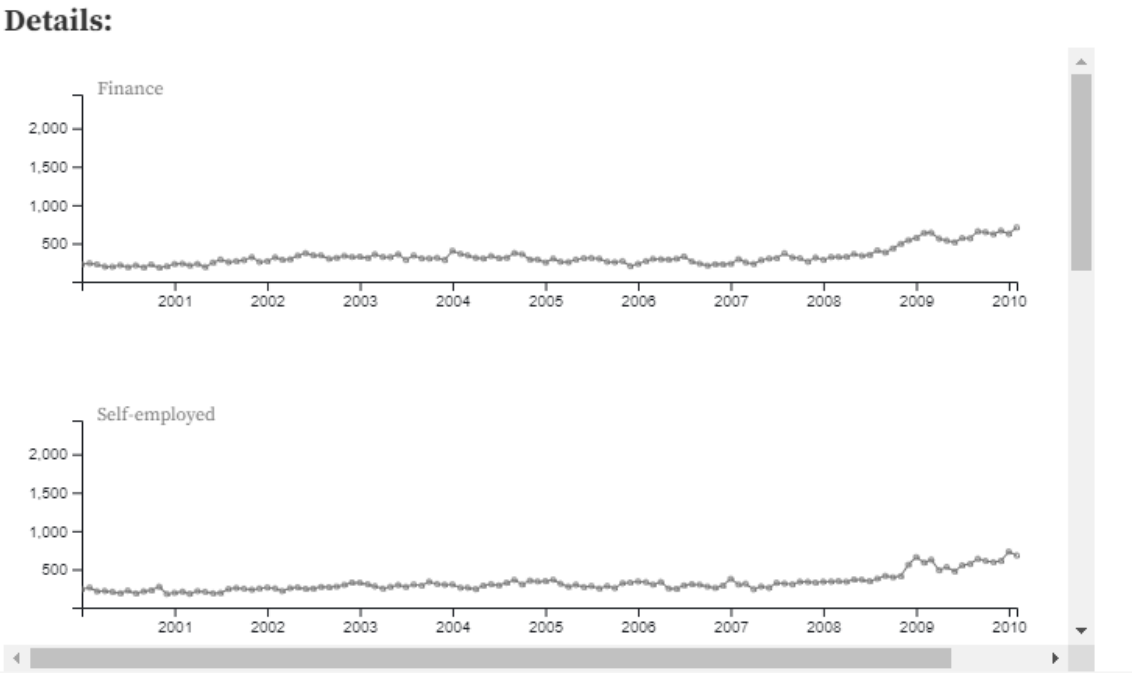


Ilustración 38: Ejemplo de la vista de detalle. Datos de los elementos "Finance" y "Self-employed"

En esta vista de detalle, cada uno de los elementos que la componen puede verse como una vista principal, pero en la que únicamente se representa un elemento determinado. Esto permite a los expertos analizar casos atípicos o de interés de forma sencilla y sin el ruido generado por el resto de los elementos.

Esta vista presenta el reto de la gran cantidad de elementos que se pueden tener seleccionados a la vez (del orden de miles), ya que cada elemento seleccionado va a requerir una visualización propia que puede llegar a consumir bastantes recursos. En una primera iteración se decidió pre-

renderizar todas las vistas de detalle de manera individual, para seguidamente ir añadiendo y eliminando estas vistas en función de los elementos seleccionados. Sin embargo, aunque el tiempo de ejecución era bastante bueno, existía el problema de saturación del DOM al tener demasiados elementos, ralentizando el navegador. Además, el consumo de memoria de la solución era demasiado elevado. Para resolver estos problemas, se optó por un enfoque diferente, en el cual únicamente se renderizan las vistas de los grupos seleccionados que son actualmente visibles en la pantalla. Para ello se ha utilizado la funcionalidad de *IntersectionObserver,* que genera eventos cada vez que un elemento aparece o desparece al realizar *scroll*. Aprovechando esta funcionalidad, se ha generado un contenedor que contiene elementos vacíos del tamaño definido para cada vista de detalle, de tal manera que cuando estos entren en el espacio visible, se completen con la información necesario (un ejemplo de esto puede verse en la Ilustración 39). Gracias a este método, se pasa de tener cargadas miles de vistas de forma simultánea a únicamente en torno a una decena.

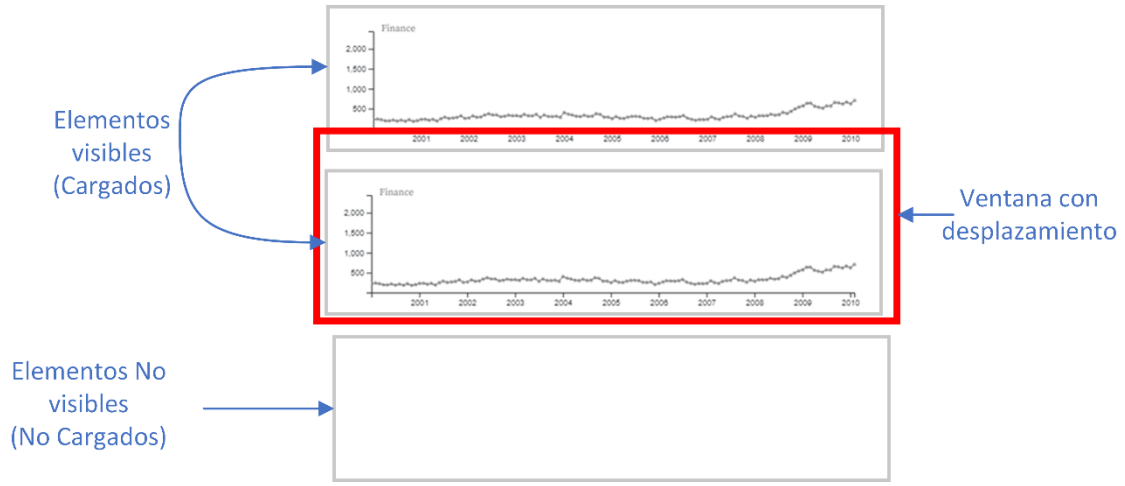


Ilustración 39: En este ejemplo de la funcionalidad de IntersectionObserver puede observase la ventana visible con desplazamiento (rojo) y los diferentes elementos que pueden mostrarse al realizar el desplazamiento de la ventana. Cuando uno de estos elementos entra en la ventana, pasa a ser un elemento visible y se procede a cargar la información relativa a ese elemento. Por el contrario, cuando un elemento sale de la ventana pasa a ser un elemento no visible y se descarga toda la información relativa a ese elemento para ahorrar recursos.

5.3. Modularidad y Reactividad

Como se ha comentado anteriormente, una parte muy importante de la herramienta es el enfoque modular que permite integrarla de forma sencilla en entornos de visualización más complejos. Además, la reactividad también es un punto importante en el diseño de esta herramienta. Así, no solo se integran varias herramientas en un entorno, sino que también se permite que cada una de ellas reaccione a los cambios de las otras de forma coordinada, obteniendo así un entorno altamente interactivo.

Respecto a la modularidad, la herramienta se encuentra disponible en un paquete NPM, lo que permite ser importada en un proyecto de forma sencilla y rápida. Además, cada uno de los componentes que presenta la herramienta (vista principal, vista de detalle, gestor de grupos, coordenadas de la *TimeBox*), pueden ser renderizados en cualquier elemento DIV que el usuario desee, permitiendo así variar el *layout* (o incluso ocultarlos directamente). Otro de los puntos para conseguir esta modularidad es la alta personalización de la herramienta, la cual contiene una gran cantidad de parámetros, permitiendo así que pueda ajustarse a diferentes necesidades de forma sencilla.

Respecto a la reactividad, esta consiste en la capacidad de la herramienta para reaccionar a cambios producidos por otras herramientas (o que estas reaccionen a sus cambios) en tiempo real. Para conseguir esto, se ha seguido el modelo propuesto por *Observable*, el cual consta de dos pasos: por un lado, se actualiza el valor de salida del elemento con el que se ha interactuado a través de su propiedad *value* y seguidamente se lanza un evento *Input*, que indica a todas las herramientas relacionadas que el valor de esa herramienta ha cambiado, para que actualicen su estado. Por lo tanto, cuando la herramienta detecta que se ha producido un cambio en los elementos seleccionados (y no un cambio en la herramienta en sí, en este caso no se actualiza para ahorrar recursos) se lleva a cabo un proceso de actualización. Este proceso consiste en actualizar la propiedad *value* e indicar al resto de herramientas el cambio por medio del evento *input.* Además, también se ha incluido un sistema de *callbacks* que permiten llegar al mismo resultado sin la utilización de *Observable*.

Sin embargo, la forma en la que se trata la reactividad en Observable tiene ciertas dificultades. Una de estas dificultades radica en que cada vez que otra aplicación lanza un evento *Input, Observable* vuelve a recalcular la celda completa en lugar de actualizar únicamente los datos, pero no elimina los elementos del DOM asociados a la celda. Este comportamiento provoca que, si no se tiene cuidado con la adición de elementos al DOM, pueda haber elementos repetidos inesperados provocando un comportamiento errático de la herramienta. Además, también es necesario tener en cuenta que algunos de los elementos del DOM de la herramienta han podido ser definidos por el usuario como un *target*, es decir un elemento del DOM donde la herramienta renderizará alguno de sus componentes. Con todo esto en mente, se ha decidido usar una estrategia en la cual se comprueba al inicio si se ha recibido un *target* en concreto por parte del usuario que será usado en ese caso. Si no, se comprueba si ya existía un elemento creado para el *target* y se reutiliza. Por último, si no existe *target* definido, se crea como comportamiento por defecto. Seguidamente, una vez creados (o reciclados) todos los *targets,* se procede o bien a crear todos los elementos necesarios para la visualización dentro de estos *targets,* o bien se reciclan los elementos que sea posible, con el objetivo de aumentar el rendimiento a la hora de mostrar los cambios.

5.4. Múltiples vistas enlazadas.

Una de las limitaciones que tiene la herramienta con lo explicado hasta este punto es que no sería posible explorar de manera eficiente más de una variable. Aunque podría utilizarse la modularidad y reactividad para conectar varios TimeSearchers, este enfoque no tendría el nivel de coordinación necesario para exprimir todo su potencial. Por ello, se decidió trabajar en un concepto de *TimeSearcher+* multivariante, el cual debe verse como un único *TimeSearcher+*, pero que permite explorar más de una variable a la vez. De esta forma, aunque en la visualización se vean varios *TimeSearchers*, estos están completamente coordinados y comparten datos, comportándose como un único elemento.

Para realizar este *TimeSearcher+* multivariado se han considerado varias opciones tanto de interacción como de concepto, llegando a las siguientes decisiones de diseño:

- Topología jerárquica: Con el fin de facilitar la compresión de la visualización por parte de los expertos, las múltiples vistas tendrán definida una relación padre-hijo de manera lineal. De esta forma, el filtrado realizado en una vista con respecto a una determinada variable (primer nivel) será la entrada de la vista siguiente (es decir su hijo) correspondiente a otra variable (segundo nivel), permitiendo así hacer un refinamiento progresivo de la selección. Nótese que *TimeSearcher+* ofrece la posibilidad de hacer

grupos en base a más de dos niveles o variables. En un primer momento, se decidió que el *TimeSearcher+* multivariado no tuviera ninguna jerarquía, de tal manera que se pudiese interactuar en cada una de las vistas sin ningún "orden", lo que permitía más libertad. Sin embargo, a los expertos les parecía muy confuso y no entendían muy bien el funcionamiento de la herramienta, provocando que perdieran rápidamente el interés.

- Coordinación de grupos: Para remarcar el hecho de que se trata de un único *TimeSearcher+* multivariado (y facilitar el uso), todas las vistas comparten la definición completa de los grupos creados, variando únicamente en el número de elementos seleccionados.
- *Highlight* ascendente: Otra característica que puede resultar interesante es visualizar la selección realizada en un *TimeSearcher+* de nivel inferior en los *TimeSearchers* superiores. Con esta característica es posible ver cómo se comporta una selección en una variable, en otra variable distinta, manteniendo además el contexto (ver Ilustración 40). Para realizar esto, el usuario únicamente tiene que seleccionar un *TimeBox* en alguna de las visualizaciones, provocando así el *Highlight* del grupo de la *TimeBox* en el resto de las visualizaciones
- Gestión de color: Uno de los puntos más importantes para conseguir este concepto de *TimeSearcher+* multivariado es tener una cohesión entre los colores elegidos para cada *TimeSearcher+* individual, pero que a su vez permitan diferenciar cada una de las variables. Con este objetivo en mente se ha decido utilizar una paleta de colores de la misma familia para cada uno de los grupos (ver Ilustración 41).

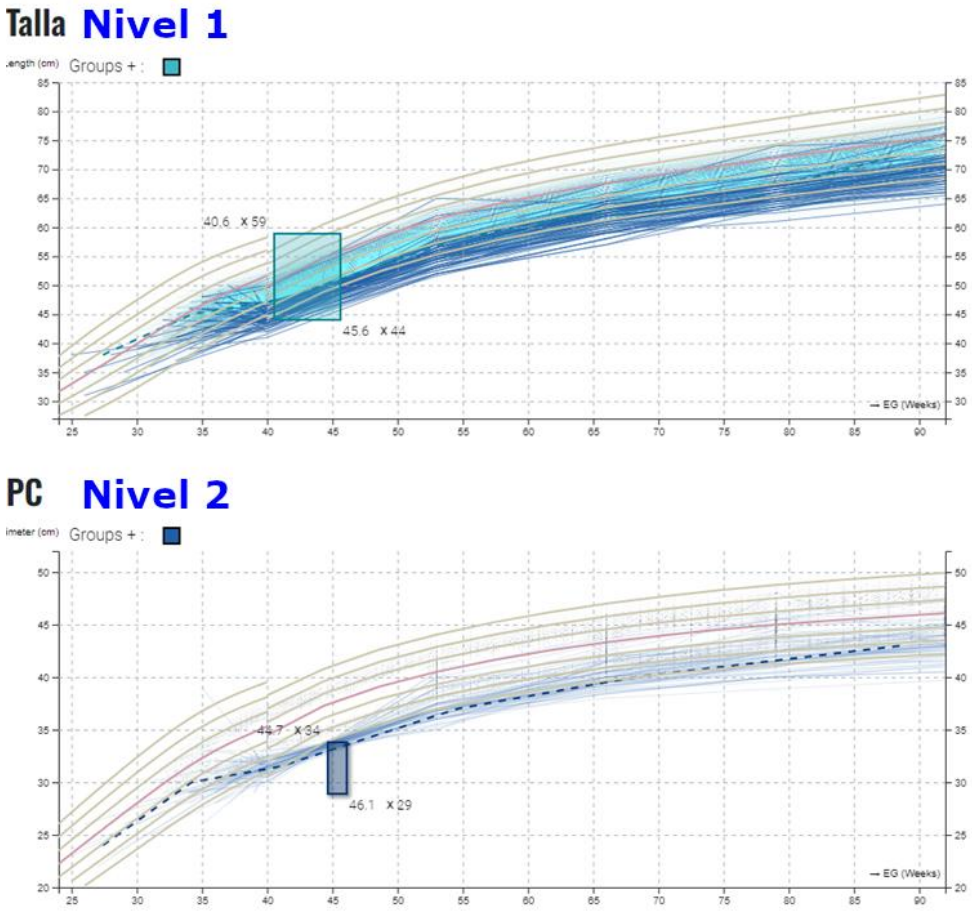


Ilustración 40: Ejemplo de visualización de la selección de una variable en otra. En la imagen puede verse cómo se comportan los elementos seleccionados en la variable PC (Nivel 2) en la visualización de la variable Talla (Nivel 1), de

tal forma que en la visualización de Talla se puede ver elementos de color azul claro que son los seleccionados por la variable talla, pero no por PC, y elementos azules oscuros que son los elementos seleccionados por Talla como por PC. Esto permite ver como los elementos que están seleccionados en PC, presentan valores bajos en la variable Talla

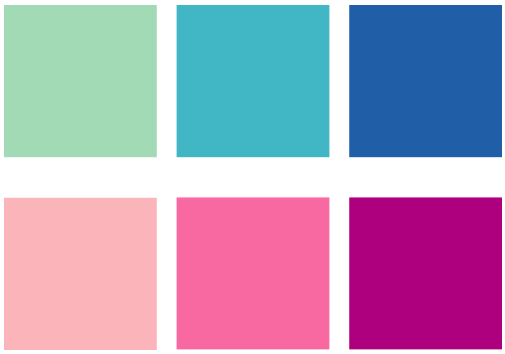


Ilustración 41: Ejemplo de paletas elegidas para un TimeSearcher+ multivariados con 2 grupos posibles y 3 niveles.

A nivel de implementación, este concepto de *TimeSearcher+* multivariante se gestiona como múltiples *TimeSearchers+* que se comunican entre sí por medio de un sistema de eventos. Además, cada uno de los *TimeSearchers+* conocen qué posición ocupa en la jerarquía, así como el número total de niveles (ver Ilustración 42). Para facilitar el uso por parte de los usuarios, lo único necesario para definir el sistema de jerarquía es indicar quién es el padre de cada uno de los *TimeSearcher+* individuales.

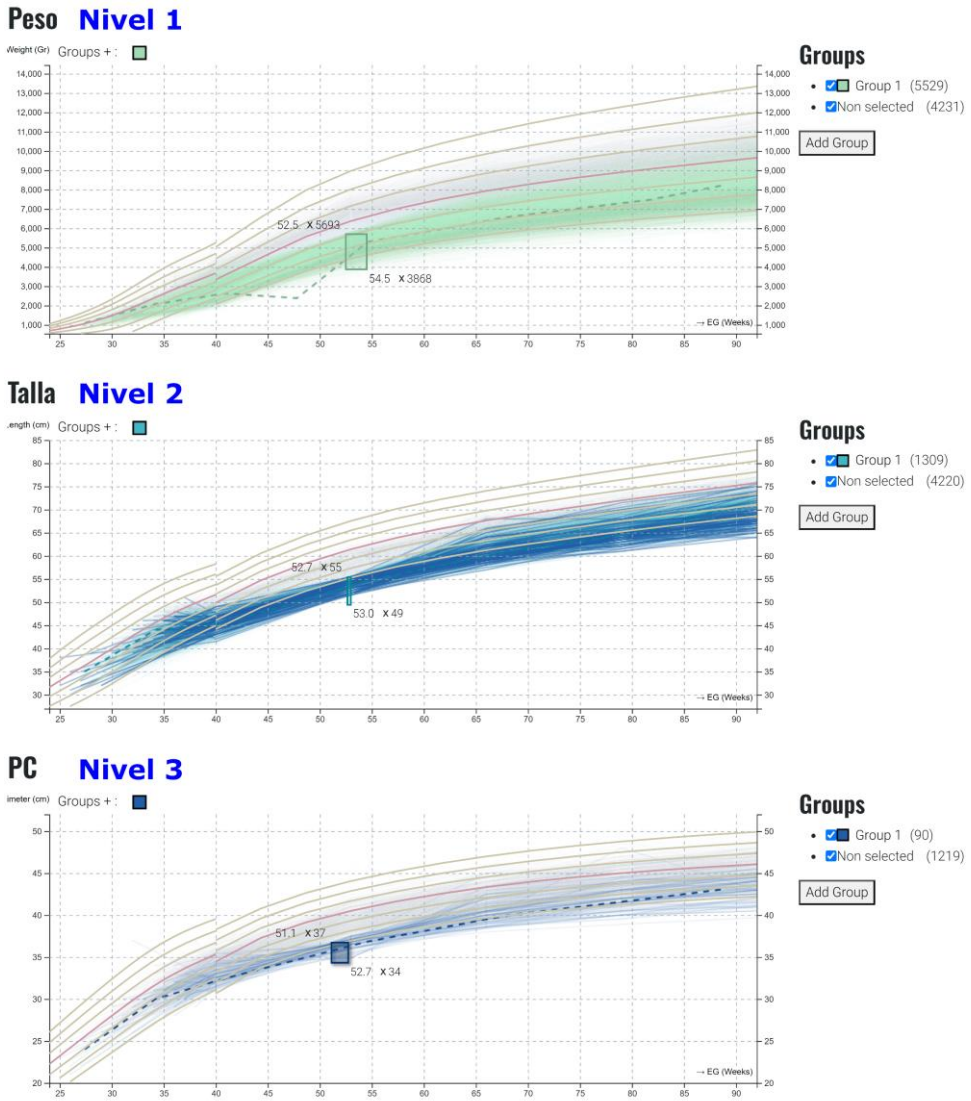


Ilustración 42:Ejemplo de la jerarquía de TimeSearcher multivariado con 3 variables. En este caso, se seleccionan unos registros en función de su Peso en la gráfica superior. De esos registros, se selecciona un subconjunto, en función de la Talla (gráfica intermedia), por último, se selecciona otro subconjunto en base a su PC (perímetro craneano). Así, los elementos finalmente seleccionados, son los que cumplen pertenecer a esos tres rangos a la vez de PC, Talla y Peso respectivamente. Nótese que la jerarquía de las variables va de arriba a abajo.

Gran parte de los eventos que maneja el *TimeSearcher+* multivariado son bastante triviales (como la gestión de grupos), ya que simplemente es necesario propagar la acción del usuario al resto de *TimeSearchers+* individuales. Sin embargo, el evento relacionado con la selección de elementos es bastante complejo, debido principalmente a la necesidad de obtener un muy buen rendimiento para que siga manteniéndose interactivo. Para conseguir esto, se ha seguido un enfoque en el cual todos los *TimeSearchers* tienen acceso a la selección realizada por el resto; así, cuando se actualiza un *TimeSearcher* únicamente es necesario recalcular su selección y propagar los cambios necesarios. En la Ilustración 43 puede observarse un ejemplo del funcionamiento de los eventos en el caso de que cambie la selección de algunos de los *TimeSearchers*.

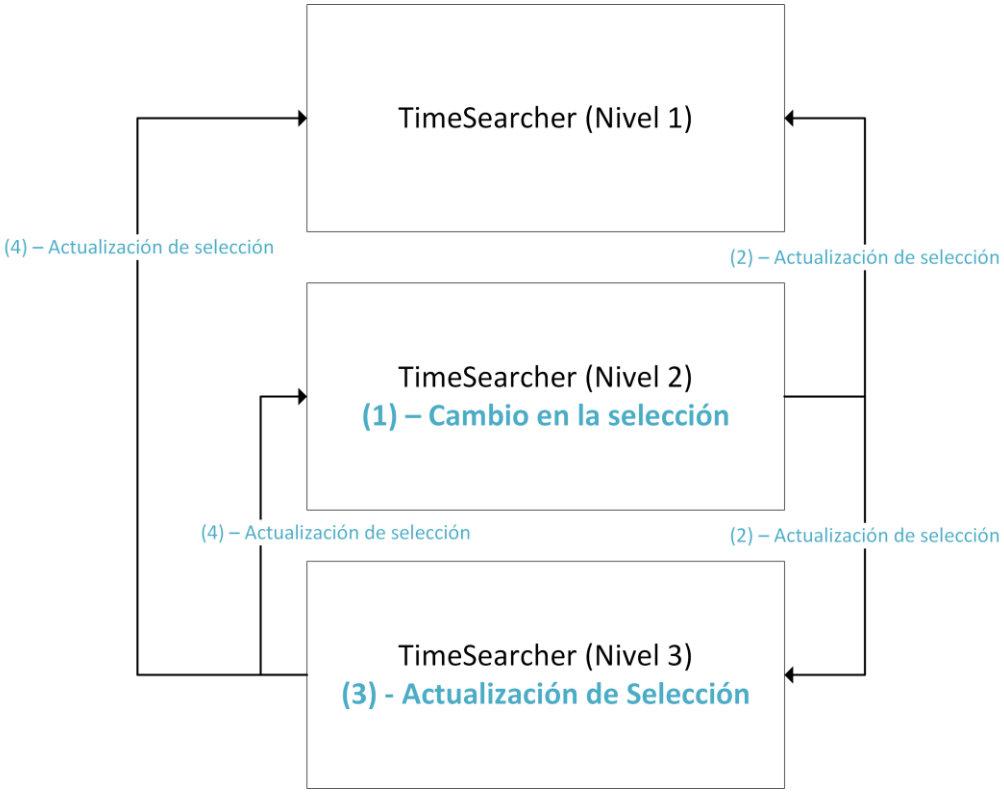


Ilustración 43: En la imagen puede observarse el proceso de envío de mensajes y actualizaciones en un TimeSearcher multivariado de 3 niveles al realizar una modificación en el segundo nivel. (1) Se produce un cambio en la selección del nivel 2. (2) Se envían eventos de actualización de selección a los niveles 1 y 3. (3) El nivel 3 recalcula su selección final en base a la nueva selección del nivel 2. (4) El Nivel 3 envía su selección final a los niveles 1 y 2.

5.5. Resultados

En esta sección se comentarán los resultados obtenidos derivados de las pruebas de usabilidad de la herramienta con expertos del programa madre canguro. Concretamente, los datos analizados constan de 50.000 sujetos a los cuales se les ha realizado un seguimiento exhaustivo hasta el año de edad, con un total de 27.000 variables. Sin embargo, en estos estudios los expertos estaban interesados sobre todo en las medidas antropométricas: peso, talla y perímetro craneano (PC), junto con las variables del desarrollo cerebral: coeficiente intelectual a los 6 meses (CD6) y coeficiente intelectual a los 12 meses (CD12).

Las pruebas de la herramienta se llevaron a cabo de forma incremental, empezando por una única instancia de *TimeSearcher+* por separado. A continuación, se probó un entorno integrado basado en las necesidades de los expertos del programa canguro y, por último, se integró el concepto de *TimeSearcher+* multivariante en el entorno de visualización. En las siguientes secciones se detallarán los resultados obtenidos en cada una de estas sesiones de evaluación con los usuarios.

5.5.1. TimeSearcher+

Para la sesión de validación con los expertos de *TimeSearcher+* se desarrolló una pequeña herramienta que permite explorar la distribución de pesos de los bebés del programa canguro y compararlos con ciertas líneas de referencia. En la Ilustración 44 puede observarse la página web diseñada para la prueba de la herramienta, donde puede verse la vista principal (1), la vista de detalle (2) y la gestión de grupos (3).

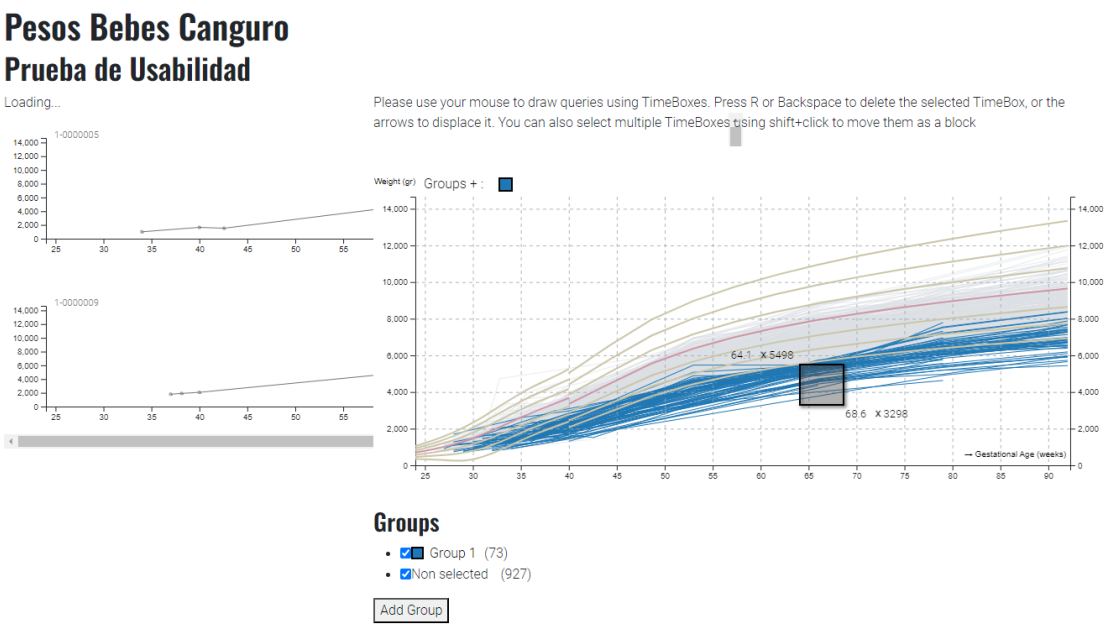


Ilustración 44: Herramienta utilizada para llevar a cabo la prueba de usabilidad de TimeSearcher+ de forma individual. En la imagen pueden observarse 3 componentes de TimeSearcher+: (1) Vista principal (izquierda); (2) Vista de detallada (derecha); (3) vista de gestión de grupos (abajo).

Se realizó una prueba de usuario con una experta, familiarizada con los datos del programa canguro. En la prueba de usuario de esta versión de la herramienta, la experta se encontró muy cómoda utilizando la herramienta, no teniendo ningún problema en conformar los diferentes grupos que se le propusieron. Así, gracias a esta prueba, pudo comprobarse que la interacción basada en *TimeBoxes* era intuitiva y fácil de utilizar. Sin embargo, aunque a la experta le parecía sencillo colocar las *TimeBoxes* e interactuar con ellas, echaba de menos mayor precisión a la hora de colocarlas con el ratón, ya que, en su campo trabajan con valores concretos.

Otra parte de la prueba consistió en dejar a la experta interactuar con la herramienta para que explorase sus propias hipótesis. En esta fase, la usuaria quería seleccionar todos los bebés que se encontraran por debajo de una curva de referencia entre unas fechas dadas. Sin embargo, el modelo de selección basado en cajas donde solo se seleccionan las líneas que pasen por todas las *TimeBoxs* no es compatible con este tipo de selecciones, por lo que la usuaria no pudo llevar a cabo su análisis de forma satisfactoria. Por lo tanto, de esta prueba se deduce la necesidad de hacer una selección en base a las líneas de referencia y que las líneas de referencia no actúen solamente como elemento visual, sino que sirvan para la interacción.

Por otra parte, la experta también comparó la herramienta propuesta con otra herramienta que utilizan de forma asidua, *InterGrow* para ver cómo se comporta el bebé en función de las curvas de referencia. La experta comentaba que la herramienta resultaba bastante tediosa de utilizar debido a que tenían que introducir los datos para la creación de las *TimeBoxes* a mano (puesto que no quería usar el ratón por ser más impreciso y no detectó la funcionalidad de poder escribir los datos en las propias esquinas de la *TimeBox*). Además, resaltaba que con *TimeSearcher+* podía hacer una selección de los bebés prematuros de características similares para poder hacerse una idea de la evolución esperada del bebé en base a casos vistos anteriormente. Esto, en su opinión, daría como resultado un seguimiento del bebé más preciso, debido a que se permite una comparación con bebés de características similares. En la actualidad, sin embargo, lo habitual es comparar simplemente con las curvas de referencia genéricas desarrolladas para bebés del norte de Europa, que no tienen por qué tener exactamente las mismas métricas que

en otras regiones. La experta resaltó que *TimeSearcher* +, por tanto, supera esa limitación. Además, también comentó que *TimeSearcher*+ podría ser una herramienta muy útil para poder mostrar a los padres de estos bebés cual es la tendencia esperada, evitando así el desaliento producido al comparar al bebé con otros que nacieron con un peso mayor y cuyos valores y evolución serán con toda probabilidad diferentes, por ejemplo.

Ilustración 45: Ejemplo de la herramienta utilizada anteriormente por la experta en el programa madre canguro, para comprobar el crecimiento del bebé prematuro con las curvas de referencia genéricas.

Además, de esta sesión, la experta también puntualizó algunas cosas mejorables de la interacción de la herramienta que le podrían ser de utilidad:

- Permitir la acción de que, al mover el ratón a un punto determinado en una línea de la gráfica y permanecer ahí un tiempo, se muestre el dato preciso de dicho punto.
- Seleccionar únicamente una línea haciendo *click*, permitiendo así la comparación sencilla de esa línea (ese bebé) con el resto.

5.5.2. KMC-Explorer

La segunda prueba de usabilidad que se llevó a cabo partía del enfoque modular de *TimeSearcher*+ y se quería comprobar cómo de útil resultaba en combinación con otras herramientas. Con este objetivo en mente, se desarrolló un entorno de visualización y análisis enfocado en las necesidades de los expertos del programa canguro denominado *KMC-Explorer*. Se contó con dos expertas en el programa canguro y familiarizadas con el significado de las variables recogidas en los datos. En este caso, las tareas de análisis a resolver serían del tipo de seleccionar dos grupos en base a su peso, e intentar ver diferencias en el desempeño al año de edad.

Este entorno (ver Ilustración 46) constaba de varias herramientas conectadas entre sí que debían ser utilizadas en un orden especifico. A continuación, se enumeran:

1. Sistema de filtrado que permite definir dos grupos de análisis a priori (los cuales se mostrarán con diferentes colores) en base a cualquier variable del juego de datos.

2. *TimeSearcher+* con el cual se pueden hacer subgrupos que pueden ser explorados por las otras herramientas.
3. Módulo estadístico que calcula diferentes métricas de las variables de interés seleccionadas.
4. Diagrama de violines que muestra la distribución de cada uno de los subgrupos seleccionados, así como cada una de las muestras individuales.
5. Selección de variables de interés.

Nótese que, aunque la aplicación está pensada para ser utilizada siguiendo un orden, puede modificarse cualquier parámetro en cada uno de los 5 componentes, y el entorno al completo será actualizado en consecuencia, manteniendo el estado del resto de componentes.

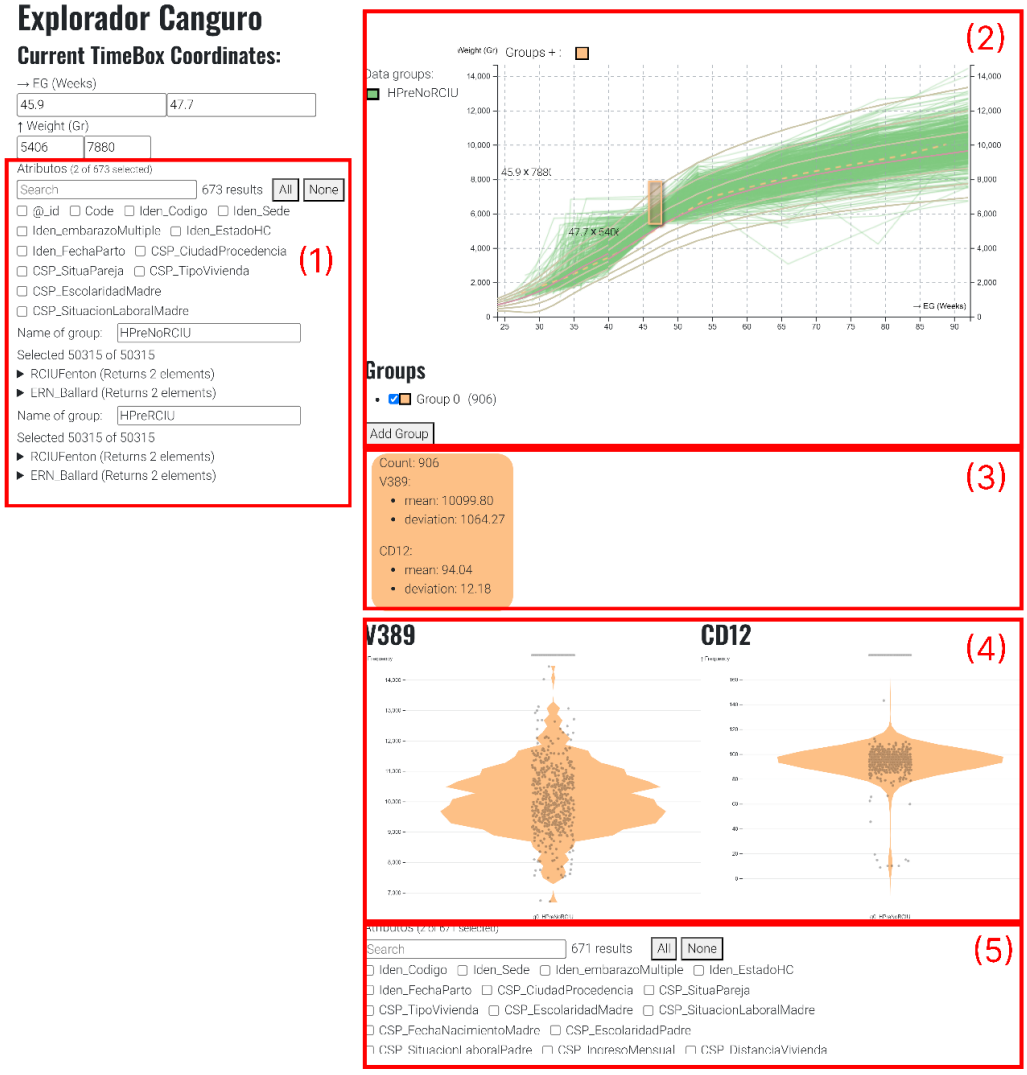


Ilustración 46: Ejemplo de KMC-Explorer, el entorno de visualización desarrollado para la prueba de usabilidad de TimeSearcher+ en conjunto con otras herramientas. (1) Módulo de filtrado de grupos a priori; (2) TimeSearcher+; (3) Módulo estadístico (4) Diagramas de violines; (5) Selección de variables de interés.

En la prueba de usabilidad, los expertos debían de realizar algunas tareas propuestas con el objetivo de comprobar la sencillez de uso de la aplicación y si los distintos elementos diseñados cumplían correctamente su función. El primer paso de la tarea a realizar consistía en dividir la muestra en dos grupos seleccionando por un lado los bebés prematuros (menos de 36 semanas) con Retardo de Crecimiento Intrauterino (RCIU) y por otro lado los bebés prematuros que no presentaban retardo de crecimiento. Al realizar este proceso, los usuarios tuvieron algunos

problemas para seleccionar los valores correctos por dos razones: el módulo de filtrado no disponía de entrada por teclado; y, además, el módulo enviaba una gran cantidad de eventos *input* de manera innecesaria, provocando latencias exageradas. En la Ilustración 47 puede verse el resultado de este filtrado en *TimeSearcher+*, donde el color verde representa los bebés prematuros sin RCIU, mientras que el morado representa los bebés prematuros con RCIU.

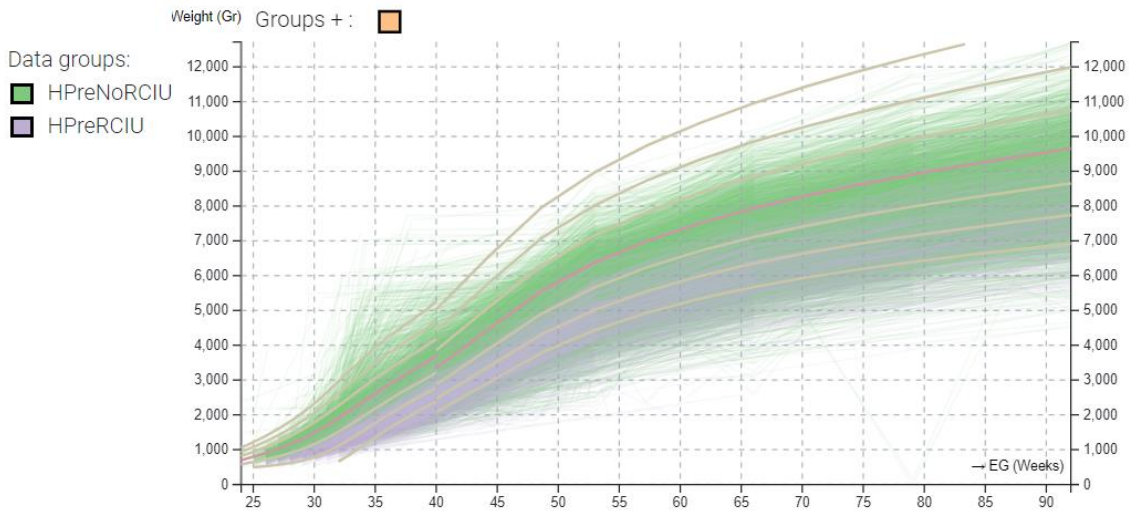


Ilustración 47: Resultado mostrado en TimeSearcher+ al filtrar los datos del programa madre canguro. En verde niños prematuros (menos de 36 semanas de gestación) sin RCIU. En morado niños prematuros con RCIU.

En la imagen los expertos pudieron darse cuenta rápidamente que, como era de esperar, los bebés que presentaban RCIU (en morado) tenían una peor evolución en peso que los bebés que no presentaban RCIU. Sin embargo, aunque queda claro que tienen una evolución peor en peso, a los expertos les surgió la pregunta de si el coeficiente intelectual al año de edad se vería afectado en este grupo. Así, antes de realizar ninguna acción en *TimeSearcher+* se dirigieron al módulo de Diagrama de violines (4), descubriendo que, aunque la diferencia en peso era notoria, la diferencia en CD no parecía tan relevante (ver Ilustración 48).

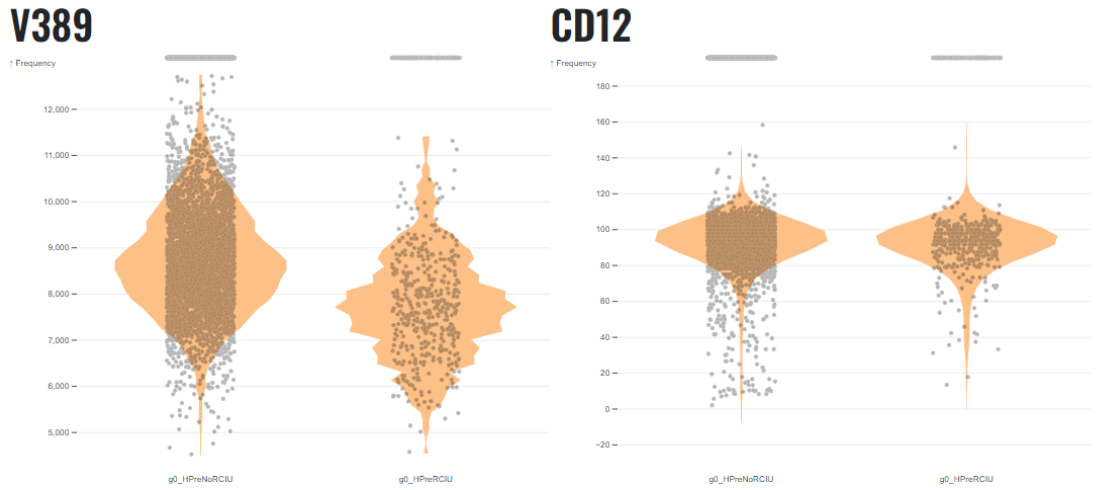


Ilustración 48: Comparativa en peso (V389) y en coeficiente intelectual a los 12 meses de edad (CD12). En cada gráfica, el violín de la izquierda representa a bebés prematuros sin RCIU y el de la derecha a bebés prematuros con RCIU. Puede observarse que, aunque las diferencias de peso son notables, las diferencias en CD12 son mucho menores.

Seguidamente, los expertos querían comparar dos grupos. El primer grupo en naranja lo componen bebés que habían comenzado bien el proceso (peso por encima de 2 desviaciones típicas a la semana 40) pero que habían acabado mal (peso por debajo de 2 desviaciones típicas al año de edad) . El segundo grupo en azul lo conformaban bebés que habían comenzado mal, pero que acabaron bien. Con esto, los expertos querían comprobar el peso del comienzo, durante el proceso, y el resultado final a las 90 semanas. En la Ilustración 49 puede verse el filtrado realizado, y en la Ilustración 50 el resultado proporcionado por el módulo estadístico y el módulo de *Diagramas de violines.*

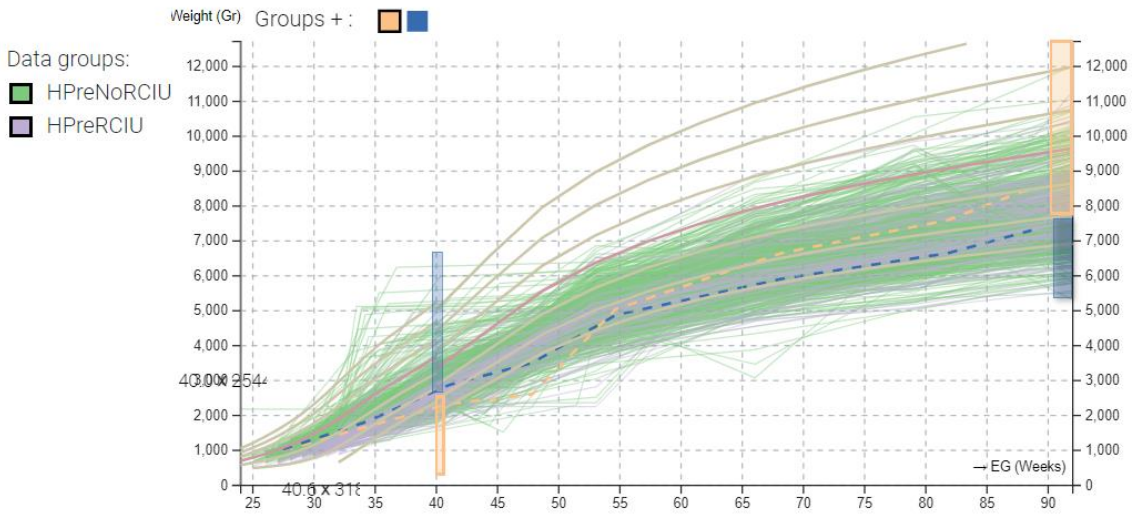


*Ilustración 49: Filtrado realizado por los expertos seleccionando, por un lado, bebés con un peso inferior a dos desviaciones típicas a las 40 semanas de edad, y un peso superior a las 2 desviaciones típicas al año de edad (**grupo naranja**). Por otro, bebés con un peso superior a las dos desviaciones típicas a la semana 40, pero con un peso inferior a 2 desviaciones típicas al año de edad (**grupo azul**)*

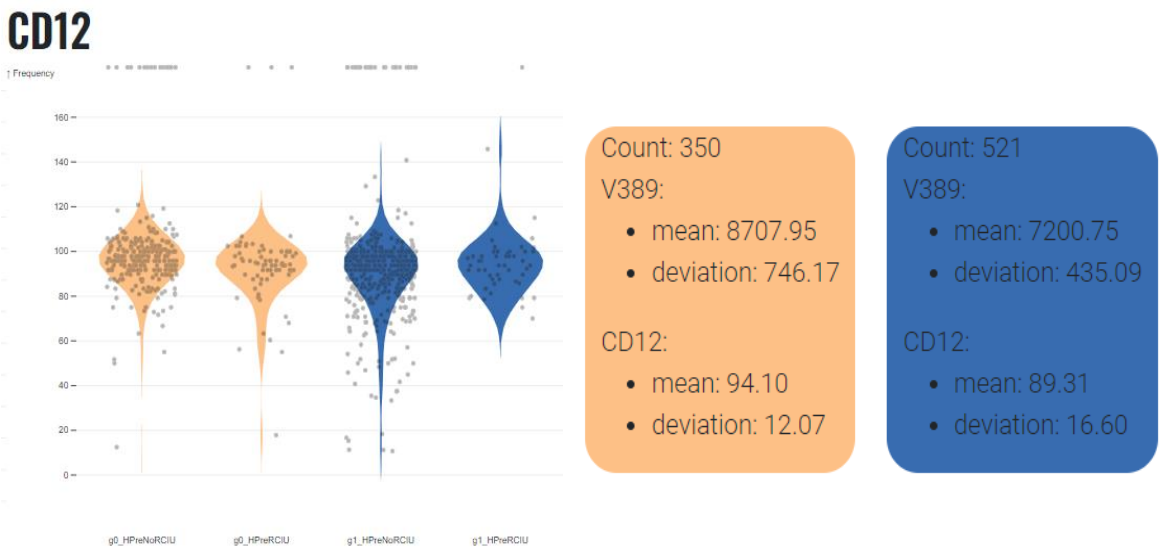


Ilustración 50: Resultados obtenidos para el filtrado mostrado en la Ilustración 49. A la izquierda, el módulo de diagrama de violines para CD12. A la derecha el módulo estadístico que muestra algunas métricas sobre los dos grupos. Se puede observar cómo los bebés que comenzaron con poco peso, pero se recuperaron (grupo naranja) presentan un mayor coeficiente intelectual a los 12 meses que los bebés que comenzaron con buen peso, pero tuvieron

problemas en el proceso (grupo azul). Esto refuerza la hipótesis de que el método canguro ayuda al desarrollo cerebral de los niños prematuros con bajo peso.

Los expertos al ver los resultados del módulo estadístico concluyeron que en realidad lo más importante para el desarrollo cerebral no era tanto el estado del bebé en su nacimiento, sino el proceso de crecimiento y el valor de peso final conseguido. Esto se evidencia al observar el módulo estadístico, donde los bebés del grupo naranja, que nacieron en peores condiciones, pero se recuperaron presentan un coeficiente intelectual mayor que los bebés que nacieron en buenas condiciones, pero que empeoraron durante el proceso de crecimiento. Se demostró así que el periodo que pasan los bebés siguiendo el programa canguro es crucial para el desarrollo cerebral del bebé prematuro. Así, en un par de minutos, los expertos pudieron comprobar sobre la marcha la pregunta de investigación que les había surgido (a falta de corroborar la hipótesis con una *suite* estadística ahora que ya disponían de los grupos creados por ellos mismos). Respecto a las conclusiones que los expertos pudieron obtener del módulo de diagrama de violines, simplemente pudieron ver que la dispersión de los coeficientes intelectuales era mayor en el grupo azul que el naranja. Además, también encontraron posibles inconsistencias en los datos con bebés con menos de 20 de CD, los cuales parecen claramente errores.

Respecto a las cosas positivas que las expertas destacaron de la herramienta, se encuentra en primer lugar la reactividad. Las expertas valoraron muy positivamente el hecho de que pudiesen modificar cualquier parte de la aplicación en tiempo real y esos cambios se viesen reflejados de inmediato en el resto de los componentes. Así, la modificación más típica consistía en añadir nuevas variables de interés para intentar encontrar una variable que explicara la diferencia entre dos grupos (por ejemplo, bebés que comenzaron de forma similar, pero tuvieron resultados muy dispares al año de edad). Además, también modificaban de forma recurrente la definición de los grupos a priori modificando, por un lado, la edad gestacional de nacimiento de los bebés, y en otro análisis, las fechas en las que los datos fueron recolectados con el objetivo de comprobar si el método había mejorado con el tiempo. Por otro lado, aunque no lo comentaron explícitamente, las expertas hacían un gran uso de la alta interactividad de la herramienta, realizando pequeñas modificaciones de forma constante y observando cómo afectaban los cambios en tiempo real a las variables de interés que tenían definidas en ese momento. De esta forma, las expertas exploraban de forma rápida hipótesis que les iban surgiendo sobre la marcha al interactuar con la herramienta guiadas por los resultados arrojados por esta. Esto es un punto muy positivo ya que era el objetivo principal de la herramienta, permitir explorar distintas hipótesis de manera rápida e intuitiva. Además, en esta línea, las expertas también comentaron que, aunque las tareas que estaban realizando con la herramienta las podrían haber realizado en su *suite* estadística (SPSS [100]), con la herramienta los cambios eran mucho más rápidos, mientras que introducir modificaciones en el análisis en SPSS es bastante más costoso y el ver los resultados inmediatamente de forma visual, también les hizo comprobar preguntas que hasta ahora no habían corroborado en su *suite* estadística. Sin embargo, las expertas no se sentían lo suficientemente seguras de la herramienta como para publicar un estudio basándose solo en la evidencia proporcionada por esta, sino que una vez encontraran algo interesante con la herramienta, querían cerciorarse exportando los datos y utilizando SPSS. Esto no es nada sorprendente, ya que como se ha comentado anteriormente, la herramienta no busca poder validar hipótesis de manera formal (aunque se podría conectar de forma sencilla con herramientas que proporcionen test estadísticos).

Por otro lado, las expertas también valoraron de forma muy positiva la inclusión de la media de los grupos en la visualización, lo que les permitía comparar de un vistazo el proceso de desarrollo

de los grupos seleccionados. Así, lo que más les interesaba era poder ver dónde los grupos empezaban a divergir con la intención de encontrar alguna explicación a esto, y poder así, mejorar estos aspectos del método madre canguro. También, encontraron muy útil la nueva funcionalidad de ajustar las coordenadas de una *TimeBox* por teclado (tal y como solicitó la primera experta en la validación del apartado 5.5.1) Aunque la interfaz integrada en la visualización les pareció algo confusa y no la vieron al inicio, el *Widget* con cajas de edición (*SpinBoxes)* les pareció más sencillo de utilizar (ver Ilustración 51),

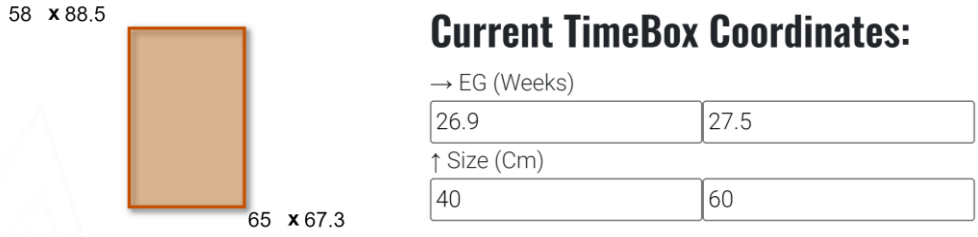


Ilustración 51: Ejemplo de las diferentes formas de modificar las coordenadas de la TimeBox seleccionada. A la Izquierda método integrado en la propia visualización en donde se pueden cambiar los valores numéricos en las propias esquinas de la TimeBox. A la Derecha Widget externo basado en 4 cajas de edición (SpinBoxes).

Respecto a las cosas negativas que las expertas encontraron en la herramienta, la más grave tiene que ver con la elección del canal de codificación de los grupos. Como se puede observar en esta versión de la herramienta se tienen dos tipos de grupo: los grupos a priori definidos mediante el módulo de filtrado y que dan color a las líneas (verde y morado), y los grupos creados en *TimeSearcher*+ (naranja y azul). El hecho de que existan dos tipos de grupos diferentes y que, además, utilicen el mismo canal de codificación visual hace confuso para el usuario ver estos grupos como dos elementos separados. Además, el hecho de que en el módulo estadístico no aparezcan los 4 subgrupos (2 a priori, divididos a su vez por *TimeSearcher*+) fomenta esta confusión, más aún cuando en el módulo de diagrama de violines si se tienen en cuenta los 4 subgrupos.

Otro de los aspectos negativos que se pudieron capturar de la prueba con usuarias es que las expertas se ceñían al *layout* y las visualizaciones propuestas, no pidiendo en ningún momento un cambio de las visualizaciones en sí, sino únicamente mejoras a los módulos y visualizaciones mostrados. Aunque este comportamiento era esperable porque los expertos en el método madre canguro no son expertos en visualización, evidencia la necesidad de que un experto en visualización ayude a los expertos en el dominio a seleccionar las visualizaciones más útiles para la tarea. Esto resta un poco de valor al enfoque modular ya que los mismos usuarios no van a modificar o encontrar deficiencias en los componentes propuestos, aunque el problema puede ser fácilmente solucionable por el experto en visualización.

5.5.3. KMC-Explorer MultipleViews

En la última prueba de usabilidad llevada a cabo en la herramienta se quería probar el desempeño del *TimeSearcher*+ multivariado, ya que, en anteriores pruebas de usabilidad los usuarios reclamaban el poder interactuar con más de una variable a la vez. Sin embargo, se intuía que el trabajo necesario para conseguir una visualización consistente, coherente y, sobre todo, útil y comprensible era muy elevado. Por este motivo se decidió dejar esta característica para un estado más avanzado, donde todos los sistemas de *TimeSearcher*+ se encontrasen más refinados y probados.

El entorno de visualización diseñado para esta prueba, *KMC-Explorer MultipleViews* es muy similar al mostrado en el apartado anterior con la diferencia de que, ahora, se tiene un *TimeSearcher*+ multivariado con 3 niveles o variables: Peso, Talla y PC (ver Ilustración 52). Es importante tener en cuenta que el orden de los niveles es importante ya que el filtrado se va refinando en cada uno de los niveles.

Por otra parte, en este *TimeSearcher*+ multivariado los colores que definen los grupos son más una familia que un color en específico, de esta forma cada uno de los componentes del *TimeSearcher*+ multivariado tiene su propio color dentro de la familia para poder distinguir sus selecciones. Sin embargo, esta sobrecarga en el número de colores a utilizar, en el ejemplo 6, es incompatible con el uso de los grupos a priori (grupos establecidos fuera de *TimeSearcher+)* tal y como estaban definidos, debido a que generarían mucha confusión. Por lo tanto, por el momento, se ha escogido no utilizar grupos a priori en el *TimeSearcher*+ multivariado (aunque a nivel de código está implementado).

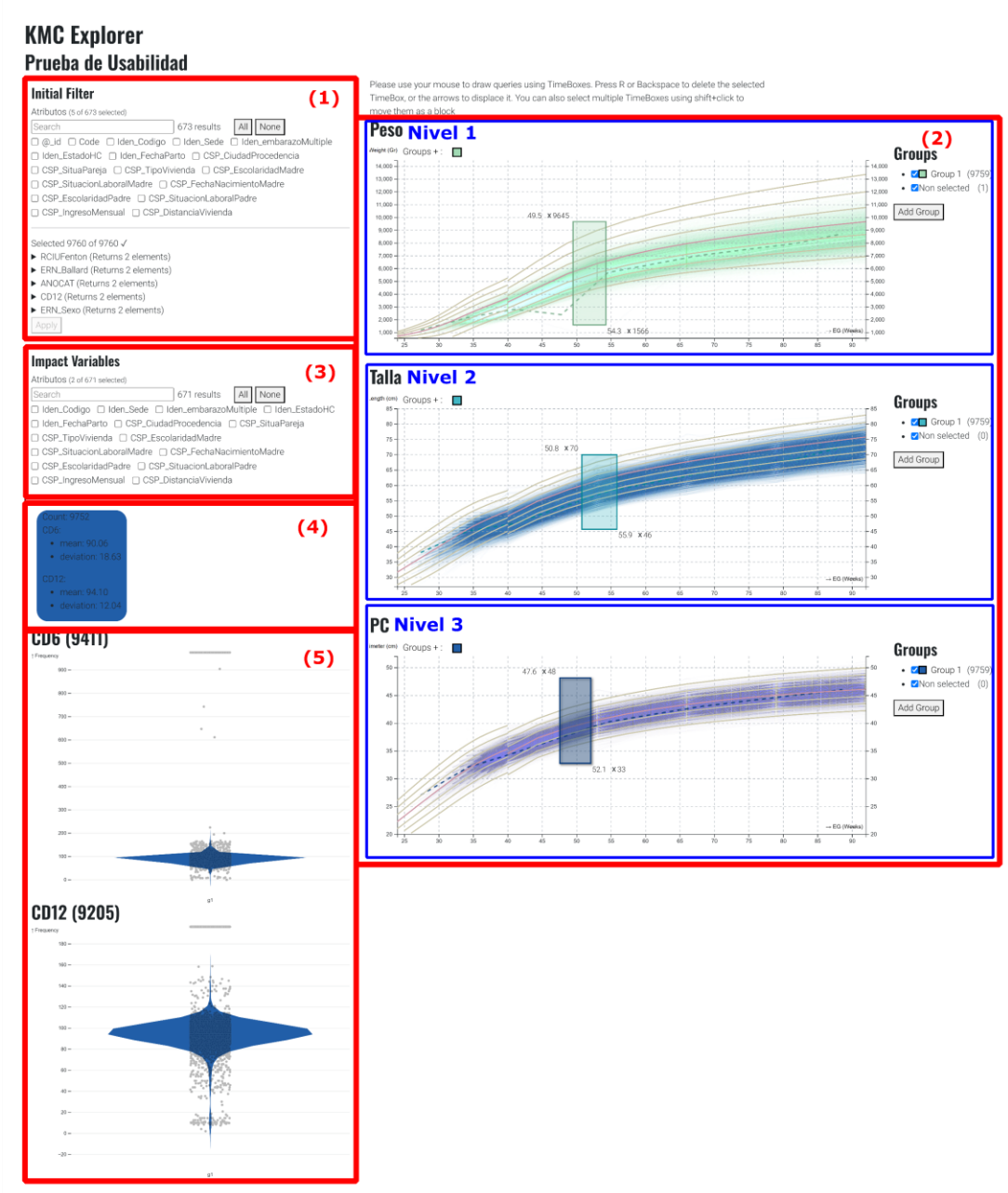


Ilustración 52: Entorno de visualización KMC-Explorer Multiview propuesto para el TimeSearcher+ multivariado que se encuentra formado por una serie de componentes: (1) Módulo de filtrado; (2) TimeSearcher+ multivariado; (3) Módulo de selección de variables de impacto; (4) Módulo estadístico; (5) Diagrama de violines. Todos los componentes citados se encuentran coordinados entre sí.

Respecto al *feedback* recibido por los usuarios (2 expertas del programa canguro), este fue bastante positivo, ya que se mantenían todos los elementos que les gustaban de la versión anterior (alta interactividad, manipulación intuitiva, actualizaciones en tiempo real) junto con el hecho de que podían tener más de una variable antropométrica de manera simultánea.

Además, también se añadió un pequeño modulo estadístico que permite comparar grupos entre sí, mostrando varias métricas como diferencia de medias o un test ANOVA para comprobar si son significativamente diferentes en una variable de interés dada. Esta nueva adición les resultó útil e interesante, utilizándola una vez habían realizado todo el proceso de análisis de forma visual para comprobar si las diferencias que se observaban en la visualización o en el otro

módulo estadístico eran significativas (aunque se les avisó que el objetivo del test ANOVA era dar una idea general, no sustituir a SPSS).

Respecto al uso del *TimeSearcher*+ multivariado, al principio les fue un poco confuso de entender el esquema de colores escogido, y el hecho de que cada componente individual no tenía sus propios grupos, sino que funcionaban como un todo (seguramente debido a cómo se presenta la interfaz, que es muy similar al caso de un único *TimeSearcher*+). Sin embargo, una vez superados estos escollos, la interacción y uso les resultó sencilla e intuitiva valorando muy positivamente el enfoque de refinamiento progresivo (ya que encaja muy bien con sus datos). Por otro lado, la función de resaltado (*Highlight)* también les resultó muy interesante, ya que les permitía ver de un vistazo cómo se comportaba la selección realizada en una variable en las otras variables, teniendo además acceso al contexto de estas variables.

Aunque la prueba de usabilidad al final transcurrió de forma satisfactoria, al principio hubo un problema grave. Debido a la gran cantidad de elementos a mostrar en 3 visualizaciones diferentes, en total unas 60.000 curvas, que se tienen que modificar en tiempo real, se observaron problemas de rendimiento. Además, estos problemas de rendimiento se ven agravados por la forma escogida para la interacción, donde una excesiva latencia provoca que las *TimeBoxes* se muevan a "saltos" más o menos grandes en función del número de curvas en la zona de interacción. Estos saltos provocan que sea imposible colocar una *TimeBox* con precisión, lo que provocó cierto rechazo por parte de los usuarios, ya que la herramienta no se comporta como ellos esperan y, sobre todo, se pierde una parte crucial del enfoque reactivo haciendo que los cambios no se vean en tiempo real. Para solucionar este problema se decidió reducir el número de líneas por componente teniendo un total de 30.000 líneas pudiendo así realizar la prueba de usabilidad centrándonos en los conceptos de visualización y no en el rendimiento de la herramienta. Sin embargo, después de la prueba se llevó a cabo un proceso de optimización de *TimeSearcher*+ multivariado pudiendo mostrar las 60.000 líneas manteniendo la interactividad.

5.6. Discusión

En este trabajo se ha desarrollado una herramienta para el análisis exploratorio centrada en la generación de múltiples grupos de análisis. Para ello se han utilizado técnicas de manipulación directa que facilitan la interacción por parte de los usuarios.

Esta herramienta está fuertemente influenciada por la herramienta *TimeSearcher* original [70] original, de la cual se ha heredado el concepto de *TimeBox*, y parte del enfoque del diseño en cuanto al objetivo de la visualización. Sin embargo, este trabajo no es solamente una reimplementación con tecnologías modernas (como se comentará más adelante), sino que se han mejorado algunas de sus funcionalidades originales, o se han añadido nuevas funciones que puedan cumplir con nuevos casos de uso.

El primero de estos cambios es la interacción de la *TimeBox* con las polilíneas generadas por los datos. En el trabajo original, esta interacción era poco intuitiva, ya que requería que todos los puntos que definían la polilínea en el intervalo en X definido por la *TimeBox* se encontrasen dentro de esta. De esta forma, únicamente se seleccionan las polilíneas que se encuentren dentro de la *TimeBox* para todo el intervalo X que define. Además, la implementación no tenía en cuenta lo que sucedía entre los puntos de definición de la polilínea (probablemente por problemas de rendimiento), pudiendo provocar problemas con polilíneas con poca cantidad de puntos. Por otro lado, esto puede causar una inconsistencia de cara a la visualización, donde el

usuario puede ver como una polilínea dada se encuentra contenida en la *TimeBox*, pero no es seleccionada al no tener puntos de definición en ese espacio. En cambio, en nuestra propuesta se ha optado por que la *TimeBox* seleccione todas las polilíneas que intersequen en algún momento con la TimeBox haciendo más sencillo de comprender su funcionamiento. En nuestro caso, además, sí se tiene en cuenta la línea generada entre dos puntos de definición.

Por otra parte, la propuesta original únicamente permitía definir un único grupo a la vez, que, aunque es útil para ciertos casos, no permite la comparación de una manera sencilla dentro de la propia herramienta. Para solucionar esta limitación, la nueva aplicación es capaz de generar varios grupos, donde cada uno de ellos tiene asociadas una serie de *TimeBoxes*. Además, gracias a que las líneas seleccionadas se colorean en función del grupo, es muy sencillo distinguir diferencias entre los distintos grupos escogidos.

Adicionalmente, también se da soporte a grupos generados a priori fuera de la herramienta, lo que permite compararlos de forma visual directamente, y dividirlos en subgrupos gracias a las *TimeBoxes*.

Algunas herramientas, aunque aportaron gran valor a la comunidad, han caído en desuso debido a que se han quedado anticuadas en cuanto a las tecnologías utilizadas en su desarrollo. Esto es debido a que es complicado realizar mejoras directamente en el código, y a que, en su momento, no estuvieron diseñadas para ser utilizadas en conjunción con otras herramientas, sino como una entidad aislada. Gracias al uso de tecnologías modernas y a los nuevos enfoques de modularidad y reactividad, se soluciona en gran parte estas problemáticas, ya que, aunque siga sin ser sencillo modificar la herramienta a nivel de código, sí que es muy sencillo comunicar distintas herramientas entre sí, complementando así sus funcionalidades.

Por otra parte, la funcionalidad de alinear las distintas observaciones en función de un evento determinado como proponen *Lifelines* y *Lifelines2* [71], [72] parece una característica muy interesante a incluir en *TimeSearcher+*. Esta funcionalidad podría ayudar a comparar la progresión de los datos estudiados partiendo desde un punto de evolución similar, lo que facilita en gran medida la comparación al encontrarse todas las observaciones alineadas respecto al eje X. Sin embargo, en este momento, al no soportar esta alineación, únicamente se puede comparar de manera realmente eficiente datos en los que todos los valores comienzan en el mismo instante temporal, o en los que el proceso de alineación se haya realizado de forma externa.

Debido al gran potencial que presenta *TimeSearcher+* se decidió intentar realizar una publicación en el *IEEEVIS 2023* [101], pero a pesar de las buenas críticas de los revisores el artículo fue rechazado, por lo que se encuentra en proceso de revisión para ser enviado a otro foro.

6. Conclusiones

Este capítulo comienza resumiendo, en base a los resultados, cuáles son las principales conclusiones obtenidas en esta tesis. Después, se detallan las contribuciones y se establecen líneas de trabajos futuros.

Esta tesis aborda aportar soluciones a distintas problemáticas de análisis y visualización cuando se trabaja con grandes conjuntos de datos complejos o con alta dimensionalidad. La hipótesis de partida, por tanto, es: "**Es posible desarrollar herramientas de visualización y análisis que puedan mejorar la comprensión de grandes conjuntos de datos complejos, en particular en aquellos casos en los que la naturaleza espacial y temporal de los datos es fundamental. Igualmente, es posible diseñar estas técnicas siguiendo un enfoque genérico, de forma que puedan ser utilizadas con diferentes tipos de datos procedentes de distintos dominios"**.

Como se ha detallado en los capítulos de la tesis, en este trabajo se han diseñado técnicas y métodos genéricos que han sido implementados posteriormente en herramientas para que puedan ser aplicados y utilizados por expertos de distintos dominios sin necesidad de que tengan grandes conocimientos matemáticos, estadísticos, ni informáticos y sin necesidad de que tengan nociones de visualización. En particular, en la introducción de esta tesis se motivó el interés de trabajar en dos casos concretos por su especial naturaleza, como son los casos de la visualización de datos morfológicos espaciales, así como con datos temporales. Estas herramientas han sido validadas con distintos ejemplos provenientes del ámbito médico. A continuación, se detallan las conclusiones con respecto a las distintas líneas de trabajo/aplicación que se han llevado a cabo en el desarrollo de esta tesis, siendo estas líneas las siguientes:

- Facilitar el análisis y posterior visualización de datos morfológicos filiformes sin estructura, así como la obtención de una malla *manifold* que permita obtener medidas correctas.
- Añadir mejoras en la visualización de datos morfológicos filiformes representados por medio de polilíneas aprovechando la nueva información disponible.
- Diseño de un clasificador basado en series temporales multivariantes para datos de carácter temporal homogéneos.
- Diseño de una visualización basada en tecnologías modernas que permita el análisis exploratorio y la creación interactiva de grupos para datos de carácter temporal heterogéneos.

Con relación al análisis de datos morfológicos, se partió de un conjunto de datos representados por medio de mallas 3D que no presentaban ninguna estructura jerárquica. Esta falta de estructura generaba una serie de inconvenientes a la hora de realizar ciertos tipos de análisis que son dependientes de esta información. Así, uno de los objetivos de esta tesis (subobjetivo 1) es facilitar el análisis y posterior visualización de datos morfológicos sin estructura. Para conseguir este objetivo se ha desarrollado una herramienta que es capaz de transformar de

forma completamente automática datos que se encuentran representados mediante mallas 3D inconexas a una representación de tipo trazado que contiene información jerárquica, apta para posteriores análisis. Además, teniendo en cuenta que esta herramienta podría tener un alto valor para la comunidad, y no únicamente para esta investigación en particular, se ha tenido en cuenta que los expertos en el dominio médico no suelen ser expertos en informática, y a menudo se sienten abrumados por herramientas demasiado complejas. Por tanto, la herramienta fue diseñada con una interfaz muy sencilla y completamente guiada, de modo que se oculta la complejidad de los parámetros por defecto, a menos que el usuario desee consultarlos o modificarlos de forma explícita.

Por otra parte, aunque el objetivo para la investigación era obtener un tipo de datos con mayores posibilidades de análisis, se consiguió paliar un problema de interoperabilidad en el dominio neurocientífico como efecto colateral. Concretamente, cada una de las representaciones utilizadas en el ámbito (mallas 3D o trazado neuronal) se encuentran asociadas a paquetes de software comerciales que ofrecen distintos tipos de análisis. Así, varios de los análisis que se pueden realizar sobre estas estructuras quedan ligados a la representación específica utilizada, no pudiendo realizarse dichos análisis directamente sobre la otra representación. Este problema provocaba que, en caso de querer obtener todos los análisis, los expertos tuvieran que realizar el trabajo de digitalización de los datos por duplicado, siendo esta tarea altamente costosa en tiempo y, además, propensa a errores. Por lo tanto, existía una clara ineficiencia a la hora de trabajar y analizar este tipo de estructuras. Igualmente, aunque los expertos realizasen el proceso de digitalización por duplicado nunca podrían tener exactamente los mismos datos en ambas herramientas (al ser un proceso manual) dificultando así su análisis y comparación. Ahora, con la transformación automática, una vez refinada la representación de mallas 3D, automáticamente se puede obtener la representación correspondiente de trazado neuronal, de modo que las dos representaciones sean congruentes y se puedan realizar los análisis disponibles en ambas herramientas y obtener sus resultados.

Adicionalmente, la herramienta permite ser ejecutada en modo *batch* para poder transformar automáticamente un gran conjunto de neuronas sin requerir la intervención del usuario.

Para comprobar el correcto funcionamiento de la herramienta desarrollada, se han realizado pruebas con multitud de archivos diferentes provenientes de laboratorios especializados en el estudio de la morfología neuronal, comparando los resultados obtenidos con la herramienta desarrollada tanto con el archivo original, como con las digitalizaciones realizadas por los expertos. Además, con el objetivo de que la herramienta no fuese únicamente de uso interno, se realizaron pruebas de usabilidad donde los expertos comentaban en voz alta las dudas o dificultades para utilizar la herramienta. En una primera versión, se les mostró un prototipo que permitía una alta configuración por medio de parámetros, y en el que la mayoría de la funcionalidad se encontraba en una misma ventana. Aunque este enfoque tenía las ventajas de ser altamente configurable y de ser muy rápido de utilizar, los expertos en el dominio neurocientífico (que no son expertos en informática) se sintieron abrumados por la gran cantidad de opciones disponibles, no siendo capaces de utilizar la herramienta de forma correcta. Al final, tras un proceso conjunto de diseño de la interfaz centrado en el usuario y varias iteraciones de revisión de la interfaz, se llegó a una versión en el que el proceso estaba profundamente guiado por medio de una serie de ventanas secuenciales, que permitían seleccionar todos los casos posibles diferentes de archivos de entrada.

De esta herramienta, de su utilización y validación, se concluye que es posible realizar desarrollos que sean capaces de generar, partiendo de datos morfológicos sin estructura, una información de estructura y jerarquía, facilitando así el posterior análisis de los datos. Es posible realizar todo ello por medio de un proceso completamente automático que es capaz de generar datos con estructura de forma coherente con los datos de entrada. Por otra parte, también se puede concluir que el hecho de disponer de los mismos datos representados de ambas maneras y de forma congruente aporta una serie de ventajas. Por ejemplo, al tener ambas representaciones disponibles, se mejora la fiabilidad de los análisis al poder extraer resultados consistentes de ambas representaciones, lo que mejora el análisis y entendimiento global de la estructura analizada.

Respecto a la sesión de validación de la herramienta por parte de los usuarios, se pueden extraer conclusiones interesantes sobre la presentación de la interfaz. Un usuario con más habilidad en el mundo de la informática prefiere una interfaz que le permita configurar el resultado a su gusto (aun teniendo en cuenta que puede no entender todos los parámetros de manera profunda hasta no dominar la herramienta). Sin embargo, para los usuarios no expertos en informática, esto supone una barrera de entrada muy grande, ya que se sienten abrumados por la gran cantidad de opciones disponibles, y sienten la necesidad de entender de forma profunda todos los parámetros antes de comenzar a utilizar la herramienta. Algo similar sucede con la velocidad de uso de la herramienta, mientras que un usuario experto priorizaría la rapidez de utilización, el usuario no experto prefiere una versión más guiada, aunque sea algo más lenta de utilizar.

Otro de los problemas que tenían los datos disponibles en morfología neuronal era que no se podían obtener métricas precisas de las espinas con alto nivel de detalle. Esto se debe a que los expertos en el campo utilizaban una herramienta (Imaris™), que funciona por medio de isosuperficies, lo que implica que el archivo de salida contenga mallas inconexas que pueden cruzarse e intersecar entre sí. En general, estos archivos son muy útiles para visualizar, pero no válidos para analizar, lo que impedía obtener métricas correctas y suponía un grave problema para los expertos en morfología neuronal. Para solucionarlo, se desarrolló una herramienta que es capaz de generar una malla 3D correcta partiendo de ese conjunto de mallas incorrectas que se cruzan e intersecan entre sí, es decir se diseñó un reparador de mallas 3D.

Además, aunque los resultados obtenidos con el reparador de mallas parecían correctos a simple vista, para proporcionar evidencia sobre la calidad de las correcciones, se deseaba comprobar de forma más precisa cuál era el error cometido por el método de reparación y en qué condiciones se producía más o menos error. Con estos objetivos en mente, se desarrolló una herramienta de visualización que permite comparar las diferencias entre dos mallas cualesquiera. Dicha herramienta se utilizó, en este caso, para comparar las mallas 3D originales con las reparadas. Gracias a esta herramienta, pudo comprobarse de manera visual que los errores cometidos por el reparador de mallas eran, según la opinión de los expertos, despreciables.

Del proceso de creación de la herramienta de reparación de mallas y de la herramienta de comparación se puede concluir que es posible realizar herramientas que sean capaces de reparar y fusionar un conjunto de mallas 3D incorrectas, generando una malla apta para analizar. Además, también se puede concluir que es importante que los resultados obtenidos tras la adquisición de datos no sean solamente aptos para la visualización, sino que también permitan realizar análisis y obtener métricas sobre estos para poder ampliar la compresión de los datos.

Por otro lado, se ha hecho patente la necesidad de poder comprobar la corrección de los resultados de una manera sencilla. La posibilidad de poder comprobar los resultados obtenidos generó una mayor confianza en el algoritmo (por parte de los expertos en el dominio) . Además, la herramienta, pese a estar diseñada para ofrecer una interfaz sencilla y guiar cada paso del proceso, también permite a usuarios más avanzados o expertos en informática poder ajustar todos los parámetros de los algoritmos en caso de ser necesario.

Respecto a los visualizadores de estructuras neuronales en el dominio neurocientífico que parten de una representación de tipo trazado, existen multitud de opciones disponibles en la literatura. Sin embargo, estas opciones tienen algunas limitaciones, como, por ejemplo, que no son capaces de generar una malla 3D conexa y correcta de toda la neurona, o que no aprovechan toda la información disponible, provocando que no sean todo lo precisas que podrían ser.

Por otro lado, con el tiempo las representaciones de la morfología neuronal han ido incluyendo más información, especialmente en lo referente al soma (describiendo su forma mediante un conjunto de contornos 2D) y a las espinas (para las que ahora se conoce su ubicación y orientación reales). Por ello, se han realizado mejoras en un visualizador existente para que toda esta información se aproveche y la visualización sea más completa y precisa.

El segundo tipo de datos que ha sido objeto de una especial atención en esta tesis es el de los datos de carácter temporal cuyo análisis es particularmente complejo. En primer lugar, dentro del proceso de análisis visual, se ha trabajado en un clasificador de series temporales, siendo estas una sucesión de muestras (de una o más variables) con un intervalo de muestreo, normalmente, constante. Aunque en la literatura se encuentran multitud de trabajos en este campo, la gran mayoría de ellos siguen un enfoque univariante, es decir que tratan las distintas señales que componen la serie temporal de forma independiente, perdiendo así la información contenida en las relaciones entre los distintos componentes de la serie temporal. Este enfoque ha obtenido resultados razonablemente buenos para algunas aplicaciones, pero el hecho de no tener en cuenta estas relaciones entre los distintos componentes limita su utilidad. Para solucionar este problema se pueden utilizar enfoques multivariantes, que sí tienen en cuenta estas relaciones entre los distintos componentes de la serie temporal, aunque presentan una serie de retos propios inherentes a su mayor complejidad. Siguiendo este enfoque multivariante, la gran mayoría de propuestas se basan en el uso de *deep-learning* que, aunque está ofreciendo muy buenos resultados, tiene la desventaja de que los resultados son poco interpretables y es complejo comprender cómo se ha llevado a cabo una determinada clasificación.

Con la intención de solucionar los problemas presentados, se ha desarrollado un método de clasificación de series temporales con un enfoque multivariante, que ha mostrado muy buenos resultados. El método se basa en MRA y la extracción de características, de tal forma que se utiliza MODWT para descomponer las señales en distintos niveles en función de las frecuencias más relevantes. Seguidamente, partiendo de esta descomposición, se obtienen una serie de variables para la clasificación, donde ciertas variables tienen en cuenta las señales de forma individual (propias del análisis univariante), mientras que otras tienen en cuenta la relación entre 2 señales (propio del análisis multivariante). El problema es que el número de variables de clasificación obtenidas puede ser demasiado alto y provocar problemas de *overfitting*. Por esta razón, el método selecciona las variables con mayor poder discriminante para utilizarlas en la clasificación.

Este método se ha validado utilizando datos provenientes del dominio de EEG. Concretamente, los datos consisten en una serie de sujetos que imaginan movimientos de manos o pies. Se

registra su actividad cerebral utilizando un total de 64 electrodos, aunque uno de ellos se descarta de cara a la clasificación, teniendo así un total de 63 señales y un total de 5500 muestras por señal. En la tarea de clasificar entre movimientos de pies y de manos con estos datos, se han obtenido precisiones cercanas al 100% usando entre 40 y 55 variables de clasificación seleccionadas (de las 22176 variables totales) llegando al 100% de aciertos en algunos casos.

Aunque los resultados de precisión obtenidos son similares a los obtenidos por otras propuestas basadas en *deep-learning*, el presente método tiene la gran ventaja de ser completamente explicable e interpretable, lo que puede ayudar a abrir nuevos caminos de investigación. Para confirmar esto, se ha llevado a cabo un pequeño estudio sobre los elementos más importantes que ha tenido en cuenta el clasificador, de tal forma que estos elementos puedan arrojar pistas sobre los datos. De este estudio se ha podido confirmar que las variables que proporcionan mayor poder discriminante son las que involucran a más de una señal, y esto parece indicar que es más importante la relación entre los electrodos que su comportamiento individual. Además, de entre los electrodos que el método ha escogido para realizar la clasificación, puede observarse cómo algunos de los electrodos que demuestran ser relevantes para la clasificación no está relacionado con la corteza motora, lo cual es un hecho intrigante que puede ser interesante investigar.

De este proceso de desarrollo y diseño pueden extraerse varias conclusiones: la primera de ellas es que es posible realizar un clasificador de series temporales con un enfoque multivariante que presente buenos resultados y que, además, sea completamente explicable e interpretable (al contrario que los clasificadores en *deep-learning*). Por otra parte, el hecho de que los métodos de análisis automáticos sean interpretables tiene un gran valor debido a que el análisis de sus resultados puede arrojar información interesante que sea el germen de futuras investigaciones.

No obstante, pese a que los métodos de análisis automáticos son muy útiles para comprobar hipótesis ya establecidas a priori, no son tan útiles a la hora de hacer un análisis cuando no se sabe muy bien qué se está buscando; es decir, cuando se necesita explorar los datos. Para esta tarea de análisis exploratorio, las herramientas de visualización interactiva han demostrado ser de gran utilidad para los expertos.

Por ello, se ha diseñado una herramienta de visualización genérica para series temporales y procesos longitudinales procedentes de cualquier dominio con un fuerte enfoque en la creación interactiva de grupos de análisis, que complemente el trabajo de clasificación comentado anteriormente. De esta forma, la herramienta no solamente permite visualizar los datos y compararlos entre ellos, sino que también es capaz, por medio de manipulación directa, de crear distintos grupos de datos en base a distintos criterios usando la herramienta *TimeSearcher*+, con la intención de que dichos grupos puedan compararse. Por otro lado, las tareas de análisis y visualización suelen ser bastante complejas y altamente dinámicas, por loque se descartó realizar una aplicación monolítica. En su lugar, se optó por un enfoque modular, lo que permite a la herramienta desarrollada integrarse con otras herramientas de forma sencilla para ampliar así su funcionalidad. Además, la herramienta también presenta un enfoque reactivo de tal forma que los cambios realizados en los datos de entrada de la herramienta provocan que las visualizaciones se actualicen de manera automática y, de la misma forma, los cambios realizados en la propia herramienta se propagan de forma instantánea al resto de herramientas, de tal forma que se puedan obtener vistas coordinadas y una interactividad muy elevada al integrar varias aplicaciones diferentes.

Esta herramienta ha sido probada dentro del programa canguro, el cual cuenta con datos de crecimiento de bebés prematuros desde su nacimiento hasta el año de edad, presentando un gran número de variables. Estos expertos proporcionaron un *feedback* muy positivo sobre la herramienta, ya que les permitía ver de forma muy sencilla las tendencias de los diferentes grupos. Además, con la herramienta pueden crear grupos interactivamente con la intención de compararlos más en profundidad. Adicionalmente, como se ha comentado anteriormente, la herramienta está diseñada para que pueda usarse en conjunto con otras herramientas, permitiendo así crear entornos de visualización más complejos. Aprovechando esta característica, se diseñó *KMC-Explorer*, un entorno dirigido a los expertos del método canguro, quienes destacaron la gran utilidad de poder cambiar cualquier parte de la aplicación en tiempo real y que estos cambios se vieran reflejados de manera inmediata en el resto de la aplicación, permitiendo por tanto explorar distintas hipótesis con grupos creados sobre la marcha de manera muy rápida. Sin embargo, es necesario tener en cuenta que la herramienta está diseñada para explorar distintas hipótesis visualmente, y ha de conectarse con un paquete estadístico para su comprobación formal.

Ante la demanda de los expertos de poder crear y visualizar interactivamente grupos de individuos en función de varias variables, se mejoró la herramienta para permitir esta funcionalidad. Se diseñó entonces *KMC-Explorer MultiViews* que satisfizo la necesidad de poder seleccionar sujetos que cumplieran distintos criterios que involucraban más de una variable. Así, por ejemplo, en este entorno es posible seleccionar bebés que tengan a la vez un peso, un perímetro craneano y una talla entre determinados valores específicos.

De esta parte del trabajo de la tesis se pueden extraer algunas conclusiones, como la importancia de disponer de herramientas que permitan una visualización exploratoria, incluso aunque aún no se tenga ninguna hipótesis preconcebida. Esto es especialmente útil cuando los datos son multivariables, temporales y complejos. Además, los expertos valoraron mucho la velocidad y la posibilidad de realizar ajustes de una manera muy rápida para observar sus hipótesis, o de introducir ligeras variaciones. Por otra parte, el enfoque modular ha permitido que sea más sencillo adaptar el entorno de visualización a las necesidades particulares de los expertos del programa canguro. Por otra parte, la inclusión de la reactividad ha permitido que la comunicación entre las distintas herramientas haya sido altamente interactiva, brindando así al usuario la posibilidad de modificar sus elecciones en cualquier parte de la aplicación y que los cambios se vean reflejados en tiempo real, permitiendo así la exploración de distintas hipótesis sobre datos temporales complejos a gran velocidad.

Por último, respecto a la hipótesis que ha dirigido el desarrollo de esta tesis: “**Es posible desarrollar herramientas de visualización y análisis que puedan mejorar la comprensión de grandes conjuntos de datos complejos, en particular en aquellos casos en los que la naturaleza espacial y temporal de los datos es fundamental. Igualmente, es posible diseñar estas técnicas siguiendo un enfoque genérico, de forma que puedan ser utilizadas con diferentes tipos de datos procedentes de distintos dominios”,** se puede concluir que es cierta, como se ha demostrado a lo largo de la tesis.

6.1. Contribuciones

En esta sección se resumirán las principales contribuciones aportadas.

Respecto al subobjetivo 1, el desarrollo de herramientas que faciliten la visualización y posterior análisis de datos de tipo morfológicos filiformes, este ha sido cumplido y ha proporcionado dos contribuciones principales:

- Se ha desarrollado una herramienta que es capaz de inferir la estructura y jerarquía partiendo de un conjunto de mallas 3D carente de estas, generando así un fichero correcto coherente con la representación original que ofrece, en consecuencia, más posibilidades de análisis de las estructuras a nivel de neurona completa.
- Se ha diseñado una herramienta de reparación de mallas que permite solventar errores tales como caras internas o agujeros y proporcionar una malla *manifold* que permita obtener medidas más fiables.
- Se ha solucionado un problema de interoperabilidad en el dominio neurocientífico donde cada una de las representaciones (mallas 3D inconexas y trazado neuronal) estaban asociadas a distintas tareas y paquetes de software comerciales. Esta situación dificultaba la obtención de todos los análisis disponibles e, incluso cuando se hacía por duplicado el proceso de extracción y refinado de los datos, no se podía garantizar que los resultados fueran congruentes, por estar ambos procesos hechos en paralelo de forma semi manual y ser susceptibles de errores. Ahora, por el contrario, es posible obtener las dos representaciones de forma que sean congruentes, lo que permite obtener todos los análisis disponibles y facilita aún más el proceso de análisis.
- Se han implementado una serie de mejoras en el visualizador de estructuras neuronales y filiformes que han conseguido aprovechar toda la información disponible tanto del soma como de las espinas para aportar una visualización más realista que la anteriormente existente. Esto permite que los expertos morfólogos cuenten con visualizaciones más precisas y completas de las neuronas.

Por otra parte, el subobjetivo 2 estaba centrado en el desarrollo de herramientas de análisis y visualización de datos de carácter temporal. Respecto a él, se puede afirmar que también se ha alcanzado, habiéndose producido las siguientes contribuciones:

- Se ha desarrollo un método de clasificación automático interpretable y explicable basado en el análisis de series temporales multivariantes. Destacablemente, el método ha obtenido muy buenos resultados en la aplicación a datos del dominio de EEG.
- Además, para facilitar la utilización de esta contribución, se ha desarrollado una librería en lenguaje R que implementa el método de tal forma que sea muy sencilla de utilizar para personas que son expertas en su dominio, pero que no están familiarizadas con la informática o la estadística.
- Una contribución notable es que se ha diseñado e implementado una herramienta de visualización de datos de carácter temporal, que permite la visualización exploratoria de grandes cantidades de datos temporales heterogéneos. Dicha herramienta está fuertemente enfocada hacia la creación interactiva de grupos de análisis para su comparación. Para su desarrollo se ha partido de una serie de principios de diseño específicos, entre los que se pueden citar su modularidad, interactividad y reactividad.

Todas estas contribuciones están disponibles para la comunidad científica por medio de 2 publicaciones científicas en revistas JCR de prestigio, de otras dos en preparación y, además las herramientas han sido publicadas para su libre utilización y distribución en modalidad *OpenSource* para que puedan ser útiles a toda la comunidad científica.

6.2. Trabajos futuros

En esta sección se detallan los posibles trabajos futuros que han ido surgiendo durante el desarrollo de la tesis pero que no han sido abarcados.

Respecto a la herramienta de reparación de mallas, es de reseñar que dejó muy satisfechos a los expertos cuando se aplicaba a espinas dendríticas. Sin embargo, comentaron que también querían utilizarlo para reparar fragmentos dendríticos completos. En esta línea se avanzó lo suficiente como para reparar los fragmentos dendríticos implicados. Sin embargo, aunque los resultados fueron bastante satisfactorios, en función de los datos de entrada, el algoritmo tenía problemas en generar una superficie lisa y en reconstruir algunos huecos presentes en las mallas. Posiblemente estos resultados podrían mejorarse realizando un ajuste fino de los parámetros del método para este tipo de estructuras, pero esto queda por estudiar. También queda pendiente la reparación de la malla de neuronas completas.

Respecto al visualizador de estructuras filiformes, se añadieron una serie de mejoras que aumentaron la precisión de la visualización aprovechando la nueva información contenida en los ficheros de tipo trazado. Sin embargo, la precisión del soma podría mejorarse más si se usase un método alternativo al *convex hull* que permitiese generar superficies con partes cóncavas. Aunque esto no afecta a una gran cantidad de somas (la gran mayoría son convexos), en el caso de somas con partes cóncavas esta característica disminuiría el error. Además, la última versión de Neurolucida™ (Neurolucida360™), dispone de información más precisa sobre las espinas, como su esqueleto, determinado por una polilínea, y, lo más importante, proporciona una malla 3D asociada. Por lo tanto, aprovechando esta nueva información, se podrían incluir modificaciones en la herramienta para aumentar la precisión de la visualización de espinas, aunque quizá se perdería en compatibilidad con otras herramientas previas.

Con relación al clasificador de datos de carácter temporal, se han obtenido muy buenos resultados al clasificar los datos de movimientos imaginarios de pies y manos. Sin embargo, es necesario tener en cuenta que estos datos estaban previamente preprocesados. Por lo tanto, sería interesante comprobar el funcionamiento del clasificador con otro conjunto de datos de movimientos imaginarios sin preprocesar, para ver si se obtienen resultados similares para obtener más evidencia sobre la generalidad el método.

Con respecto al visualizador interactivo de datos temporales heterogéneos que permite la creación interactiva de grupos, los resultados obtenidos en las pruebas de usabilidad con los expertos han sido muy satisfactorios. Sin embargo, todavía queda trabajo pendiente a la hora de perfeccionar algunas de las características relevantes de la herramienta, como son su reactividad, facilidad de uso y escalabilidad.

Como trabajo futuro, queremos mejorar la reactividad de *TimeSearcher+*, facilitar su uso y mejorar aún más la escalabilidad. También queremos incluir herramientas para el preprocesamiento de los datos. Para ello, planeamos incluir módulos que permitan al usuario alinear los datos por diferentes eventos, atributos o patrones y ofrecer operaciones para diferentes granularidades temporales.

7. Bibliografía

[1] X. Jin, B. W. Wah, X. Cheng, and Y. Wang, "Significance and Challenges of Big Data Research," *Big Data Research*, vol. 2, no. 2, pp. 59–64, Jun. 2015, doi: 10.1016/j.bdr.2015.01.006.

[2] C. W. Tsai, C. F. Lai, H. C. Chao, and A. V. Vasilakos, "Big data analytics: a survey," *J Big Data*, vol. 2, no. 1, p. 21, Dec. 2015, doi: 10.1186/s40537-015-0030-3.

[3] D. Keim, H. Qu, and K. L. Ma, "Big-data visualization," *IEEE Computer Graphics and Applications*, vol. 33, no. 4. pp. 20–21, 2013. doi: 10.1109/MCG.2013.54.

[4] S. M. Ali, N. Gupta, G. K. Nayak, and R. K. Lenka, "Big data visualization: Tools and challenges," in *Proceedings of the 2016 2nd International Conference on Contemporary Computing and Informatics, IC3I 2016*, Institute of Electrical and Electronics Engineers Inc., 2016, pp. 656–660. doi: 10.1109/IC3I.2016.7918044.

[5] S. Boyapati, S. R. Swarna, V. Dutt, and N. Vyas, "Big data approach for medical data classification: A review study," in *Proceedings of the 3rd International Conference on Intelligent Sustainable Systems, ICISS 2020*, Institute of Electrical and Electronics Engineers Inc., Dec. 2020, pp. 762–766. doi: 10.1109/ICISS49785.2020.9315870.

[6] "Visualization & Graphics Lab." Accessed: Sep. 12, 2023. [Online]. Available: https://vg-lab.es/

[7] "Human Brain Project." Accessed: Jun. 21, 2023. [Online]. Available: https://www.humanbrainproject.eu/en/

[8] K. Amunts *et al.*, "The Human Brain Project—Synergy between neuroscience, computing, informatics, and brain-inspired technologies," *PLoS Biol*, vol. 17, no. 7, Jul. 2019, doi: 10.1371/JOURNAL.PBIO.3000344.

[9] M. Halavi, K. A. Hamilton, R. Parekh, and G. A. Ascoli, "Digital reconstructions of neuronal morphology: Three decades of research trends," *Front Neurosci*, vol. 6, no. APR, p. 23417, Apr. 2012, doi: 10.3389/FNINS.2012.00049/BIBTEX.

[10] E. P. Cervantes, C. H. Comin, R. M. C. Junior, and L. da F. Costa, "Morphological Neuron Classification Based on Dendritic Tree Hierarchy," *Neuroinformatics*, vol. 17, no. 1, pp. 147–161, Jan. 2019, doi: 10.1007/s12021-018-9388-7.

[11] P. Reymond, F. Merenda, F. Perren, D. Rüfenacht, and N. Stergiopulos, "Validation of a one-dimensional model of the systemic arterial tree," *Am J Physiol Heart Circ Physiol*, vol. 297, no. 1, Jul. 2009, doi: 10.1152/ajpheart.00037.2009.

[12] D. Bonneau, P. M. DiFrancesco, and D. Jean Hutchinson, "Surface reconstruction for three-dimensional rockfall volumetric analysis," *ISPRS Int J Geoinf*, vol. 8, no. 12, Nov. 2019, doi: 10.3390/ijgi8120548.

[13] Herbert. Edelsbrunner and J. (John) Harer, *Computational topology : an introduction*. American Mathematical Society, 2010.

[14] U. R. Acharya, S. Vinitha Sree, G. Swapna, R. J. Martis, and J. S. Suri, "Automated EEG analysis of epilepsy: A review," *Knowl Based Syst*, vol. 45, pp. 147–165, Jun. 2013, doi: 10.1016/J.KNOSYS.2013.02.014.

[15] S. J. M. Smith, "EEG in the diagnosis, classification, and management of patients with epilepsy," *Neurology in Practice*, vol. 76, no. 2. BMJ Publishing Group Ltd, pp. ii2–ii7, Jun. 01, 2005. doi: 10.1136/jnnp.2005.069245.

[16] C. Lehmann *et al.*, "Application and comparison of classification algorithms for recognition of Alzheimer's disease in electrical brain activity (EEG)," *J Neurosci Methods*, vol. 161, no. 2, pp. 342–350, Apr. 2007, doi: 10.1016/J.JNEUMETH.2006.10.023.

[17] Q. Ge, Z. C. Lin, Y. X. Gao, and J. X. Zhang, "A Robust Discriminant Framework Based on Functional Biomarkers of EEG and Its Potential for Diagnosis of Alzheimer's Disease," *Healthcare 2020, Vol. 8, Page 476*, vol. 8, no. 4, p. 476, Nov. 2020, doi: 10.3390/HEALTHCARE8040476.

[18] I. Velasco, A. Sipols, C. S. De Blas, L. Pastor, and S. Bayona, "Motor imagery EEG signal classification with a multivariate time series approach," *Biomed Eng Online*, vol. 22, no. 1, pp. 1–24, Dec. 2023, doi: 10.1186/S12938-023-01079-X/FIGURES/10.

[19] E. A. Maharaj and A. M. Alonso, "Discriminant analysis of multivariate time series: Application to diagnosis based on ECG signals," *Comput Stat Data Anal*, vol. 70, pp. 67–87, Feb. 2014, doi: 10.1016/j.csda.2013.09.006.

[20] D. E. Donohue and G. A. Ascoli, "Automated reconstruction of neuronal morphology: An overview," *Brain Research Reviews*, vol. 67, no. 1–2. Elsevier, pp. 94–102, Jun. 24, 2011. doi: 10.1016/j.brainresrev.2010.11.003.

[21] M. K. Pugsley and R. Tabrizchi, "The vascular system: An overview of structure and function," *J Pharmacol Toxicol Methods*, vol. 44, no. 2, pp. 333–340, Sep. 2000, doi: 10.1016/S1056-8719(00)00125-8.

[22] Y. Wang, A. Narayanaswamy, C. L. Tsai, and B. Roysam, "A broadly applicable 3-D neuron tracing method based on open-curve snake," *Neuroinformatics*, vol. 9, no. 2–3, pp. 193–217, Sep. 2011, doi: 10.1007/s12021-011-9110-5.

[23] H. Xiao and H. Peng, "APP2: Automatic tracing of 3D neuron morphology based on hierarchical pruning of a gray-weighted image distance-tree," *Bioinformatics*, vol. 29, no. 11, pp. 1448–1454, 2013, doi: 10.1093/bioinformatics/btt170.

[24] X. Ming *et al.*, "Rapid reconstruction of 3D neuronal morphology from light microscopy images with augmented rayburst sampling," *PLoS One*, vol. 8, no. 12, pp. 1–10, 2013, doi: 10.1371/journal.pone.0084557.

[25] H. Chen, H. Xiao, T. Liu, and H. Peng, "SmartTracing: self-learning-based Neuron reconstruction," *Brain Inform*, vol. 2, no. 3, pp. 135–144, 2015, doi: 10.1007/s40708-015-0018-y.

[26] S. Liu, D. Zhang, S. Liu, D. Feng, H. Peng, and W. Cai, "Rivulet: 3D Neuron Morphology Tracing with Iterative Back-Tracking," *Neuroinformatics*, vol. 14, no. 4, pp. 387–401, 2016, doi: 10.1007/s12021-016-9302-0.

[27] Z. Zhou, X. Liu, B. Long, and H. Peng, "TReMAP: Automatic 3D Neuron Reconstruction Based on Tracing, Reverse Mapping and Assembling of 2D Projections," *Neuroinformatics*, vol. 14, no. 1, pp. 41–50, 2016, doi: 10.1007/s12021-015-9278-1.

[28] T. Quan *et al.*, "NeuroGPS-Tree: Automatic reconstruction of large-scale neuronal populations with dense neurites," *Nat Methods*, vol. 13, no. 1, pp. 51–54, Dec. 2015, doi: 10.1038/nmeth.3662.

[29] C. W. Wang, Y. C. Lee, H. Pradana, Z. Zhou, and H. Peng, "Ensemble Neuron Tracer for 3D Neuron Reconstruction," *Neuroinformatics*, vol. 15, no. 2, pp. 185–198, 2017, doi: 10.1007/s12021-017-9325-1.

[30] D. Z. Jin, T. Zhao, D. L. Hunt, R. P. Tillage, C. L. Hsu, and N. Spruston, "ShuTu: Open-Source Software for Efficient and Accurate Reconstruction of Dendritic Morphology," *Front Neuroinform*, vol. 13, no. October, pp. 1–19, 2019, doi: 10.3389/fninf.2019.00068.

[31] C. T. Rueden *et al.*, "ImageJ2: ImageJ for the next generation of scientific image data," *BMC Bioinformatics*, vol. 18, no. 1, p. 529, Nov. 2017, doi: 10.1186/s12859-017-1934-z.

[32] C. A. Schneider, W. S. Rasband, and K. W. Eliceiri, "NIH Image to ImageJ: 25 years of image analysis," *Nature Methods*, vol. 9, no. 7. Nature Publishing Group, pp. 671–675, Jul. 28, 2012. doi: 10.1038/nmeth.2089.

[33] J. Schindelin *et al.*, "Fiji: An open-source platform for biological-image analysis," *Nature Methods*, vol. 9, no. 7. Nature Publishing Group, pp. 676–682, Jul. 28, 2012. doi: 10.1038/nmeth.2019.

[34] A. Cardona *et al.*, "TrakEM2 Software for Neural Circuit Reconstruction," *PLoS One*, vol. 7, no. 6, p. e38011, Jun. 2012, doi: 10.1371/journal.pone.0038011.

[35] P. Gleeson, V. Steuber, and R. A. Silver, "neuroConstruct: A Tool for Modeling Networks of Neurons in 3D Space," *Neuron*, vol. 54, no. 2, p. 219, Apr. 2007, doi: 10.1016/J.NEURON.2007.03.025.

[36] J. P. Brito, S. Mata, S. Bayona, L. Pastor, J. DeFelipe, and R. Benavides-Piccione, "Neuronize: a tool for building realistic neuronal cell morphologies," *Front Neuroanat*, vol. 7, p. 15, Jun. 2013, doi: 10.3389/fnana.2013.00015.

[37] M. Abdellah *et al.*, "NeuroMorphoVis: A collaborative framework for analysis and visualization of neuronal morphology skeletons reconstructed from microscopy stacks," *Bioinformatics*, vol. 34, no. 13, pp. i574–i582, 2018, doi: 10.1093/bioinformatics/bty231.

[38] J. J. Garcia-Cantero, J. P. Brito, S. Mata, S. Bayona, and L. Pastor, "NeuroTessMesh: A Tool for the Generation and Visualization of Neuron Meshes and Adaptive On-the-Fly

Refinement," *Front Neuroinform*, vol. 11, p. 38, Jun. 2017, doi: 10.3389/fninf.2017.00038.

[39] J. Siebert, J. Groß, and C. Schroth, "A Systematic Review of Packages for Time Series Analysis," in *The 7th International conference on Time Series and Forecasting*, Basel Switzerland: MDPI, Jun. 2021, p. 22. doi: 10.3390/engproc2021005022.

[40] F. Dama and C. Sinoquet, "Time Series Analysis and Modeling to Forecast: a Survey," Mar. 2021, Accessed: Jul. 13, 2023. [Online]. Available: http://arxiv.org/abs/2104.00164

[41] U. R. Acharya, S. L. Oh, Y. Hagiwara, J. H. Tan, and H. Adeli, "Deep convolutional neural network for the automated detection and diagnosis of seizure using EEG signals," *Comput Biol Med*, vol. 100, pp. 270–278, Sep. 2018, doi: 10.1016/J.COMPBIOMED.2017.09.017.

[42] H. Işik and E. Sezer, "Diagnosis of epilepsy from electroencephalography signals using multilayer perceptron and Elman artificial neural networks and wavelet transform," *J Med Syst*, vol. 36, no. 1, pp. 1–13, Feb. 2012, doi: 10.1007/s10916-010-9440-0.

[43] S. Lopez, G. Suarez, D. Jungreis, I. Obeid, and J. Picone, "Automated Identification of Abnormal Adult EEGs," ... *IEEE Signal Processing in Medicine and Biology Symposium. IEEE Signal Processing in Medicine and Biology Symposium*, vol. 2015, Feb. 2015, doi: 10.1109/SPMB.2015.7405423.

[44] "Generalized EEG Waveform Abnormalities: Overview, Background Slowing, Intermittent Slowing." Accessed: Jul. 13, 2023. [Online]. Available: https://emedicine.medscape.com/article/1140075-overview

[45] J. W. C. Medithe and U. R. Nelakuditi, "Study of normal and abnormal EEG," *ICACCS 2016 - 3rd International Conference on Advanced Computing and Communication Systems: Bringing to the Table, Futuristic Technologies from Arround the Globe*, Oct. 2016, doi: 10.1109/ICACCS.2016.7586341.

[46] Ö. Yıldırım, U. B. Baloglu, and U. R. Acharya, "A deep convolutional neural network model for automated identification of abnormal EEG signals," *Neural Computing and Applications 2018 32:20*, vol. 32, no. 20, pp. 15857–15868, Nov. 2018, doi: 10.1007/S00521-018-3889-Z.

[47] J. Henriques *et al.*, "Protocol Design Challenges in the Detection of Awareness in Aware Subjects Using EEG Signals," *Clin EEG Neurosci*, vol. 47, no. 4, pp. 266–275, Oct. 2016, doi: 10.1177/1550059414560397.

[48] K. W. Ha and J. W. Jeong, "Motor Imagery EEG Classification Using Capsule Networks," *Sensors 2019, Vol. 19, Page 2854*, vol. 19, no. 13, p. 2854, Jun. 2019, doi: 10.3390/S19132854.

[49] S. Kundu and S. Ari, "Brain-Computer Interface Speller System for Alternative Communication: A Review," *IRBM*, vol. 43, no. 4, pp. 317–324, Aug. 2022, doi: 10.1016/J.IRBM.2021.07.001.

[50] E. A. Maharaj and A. M. Alonso, "Discriminant analysis of multivariate time series: Application to diagnosis based on ECG signals," *Comput Stat Data Anal*, vol. 70, pp. 67–87, Feb. 2014, doi: 10.1016/j.csda.2013.09.006.

[51] B. Dhariyal, T. Le Nguyen, S. Gsponer, and G. Ifrim, "An Examination of the State-of-the-Art for Multivariate Time Series Classification," in *IEEE International Conference on Data Mining Workshops, ICDMW*, IEEE Computer Society, Nov. 2020, pp. 243–250. doi: 10.1109/ICDMW51313.2020.00042.

[52] Y. Zhang, X. Ji, and Y. Zhang, "Classification of EEG signals based on AR model and approximate entropy," *Proceedings of the International Joint Conference on Neural Networks*, vol. 2015-Septe, Sep. 2015, doi: 10.1109/IJCNN.2015.7280840.

[53] R. Chai *et al.*, "Driver Fatigue Classification with Independent Component by Entropy Rate Bound Minimization Analysis in an EEG-Based System," *IEEE J Biomed Health Inform*, vol. 21, no. 3, pp. 715–724, May 2017, doi: 10.1109/JBHI.2016.2532354.

[54] S. Taran and V. Bajaj, "Drowsiness Detection Using Adaptive Hermite Decomposition and Extreme Learning Machine for Electroencephalogram Signals," *IEEE Sens J*, vol. 18, no. 21, pp. 8855–8862, Nov. 2018, doi: 10.1109/JSEN.2018.2869775.

[55] G. Bin Huang, Q. Y. Zhu, and C. K. Siew, "Extreme learning machine: A new learning scheme of feedforward neural networks," in *IEEE International Conference on Neural Networks - Conference Proceedings*, 2004, pp. 985–990. doi: 10.1109/IJCNN.2004.1380068.

[56] Y. Izza, A. Ignatiev, and J. Marques-Silva, "On Explaining Decision Trees," Oct. 2020, Accessed: Sep. 19, 2023. [Online]. Available: http://arxiv.org/abs/2010.11034

[57] K. Taunk, S. De, S. Verma, and A. Swetapadma, "A brief review of nearest neighbor algorithm for learning and classification," in *2019 International Conference on Intelligent Computing and Control Systems, ICCS 2019*, Institute of Electrical and Electronics Engineers Inc., May 2019, pp. 1255–1260. doi: 10.1109/ICCS45141.2019.9065747.

[58] R. Bousseta, I. El Ouakouak, M. Gharbi, and F. Regragui, "EEG Based Brain Computer Interface for Controlling a Robot Arm Movement Through Thought," *IRBM*, vol. 39, no. 2, pp. 129–135, Apr. 2018, doi: 10.1016/J.IRBM.2018.02.001.

[59] C. Yang, H. Wu, Z. Li, W. He, N. Wang, and C. Y. Su, "Mind control of a robotic arm with visual fusion technology," *IEEE Trans Industr Inform*, vol. 14, no. 9, pp. 3822–3830, Sep. 2018, doi: 10.1109/TII.2017.2785415.

[60] S. Chaudhary, S. Taran, V. Bajaj, and A. Sengur, "Convolutional Neural Network Based Approach Towards Motor Imagery Tasks EEG Signals Classification," *IEEE Sens J*, vol. 19, no. 12, pp. 4494–4500, Jun. 2019, doi: 10.1109/JSEN.2019.2899645.

[61] G. Tian and Y. Liu, "Simple convolutional neural network for left-right hands motor imagery EEG signals classification," *International Journal of Cognitive Informatics and Natural Intelligence*, vol. 13, no. 3, pp. 36–49, Jul. 2019, doi: 10.4018/IJCINI.2019070103.

[62] C. Ieracitano, N. Mammone, A. Hussain, and F. C. Morabito, "A novel multi-modal machine learning based approach for automatic classification of EEG recordings in dementia," *Neural Networks*, vol. 123, pp. 176–190, Mar. 2020, doi: 10.1016/J.NEUNET.2019.12.006.

[63] Z. Gao, T. Yuan, X. Zhou, C. Ma, K. Ma, and P. Hui, "A Deep Learning Method for Improving the Classification Accuracy of SSMVEP-Based BCI," *IEEE Transactions on Circuits and*

Systems II: Express Briefs, vol. 67, no. 12, pp. 3447–3451, Dec. 2020, doi: 10.1109/TCSII.2020.2983389.

[64] J. Xie *et al.*, "A Transformer-Based Approach Combining Deep Learning Network and Spatial-Temporal Information for Raw EEG Classification," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 30, pp. 2126–2136, 2022, doi: 10.1109/TNSRE.2022.3194600.

[65] F. Karim, S. Majumdar, H. Darabi, and S. Harford, "Multivariate LSTM-FCNs for time series classification," *Neural Networks*, vol. 116, pp. 237–245, Aug. 2019, doi: 10.1016/j.neunet.2019.04.014.

[66] H. Ismail Fawaz, G. Forestier, J. Weber, L. Idoumghar, and P. A. Muller, "Deep learning for time series classification: a review," *Data Min Knowl Discov*, vol. 33, no. 4, pp. 917–963, Jul. 2019, doi: 10.1007/s10618-019-00619-1.

[67] A. P. Ruiz, M. Flynn, J. Large, M. Middlehurst, and A. Bagnall, "The great multivariate time series classification bake off: a review and experimental evaluation of recent algorithmic advances," *Data Min Knowl Discov*, vol. 35, no. 2, pp. 401–449, Mar. 2021, doi: 10.1007/s10618-020-00727-3.

[68] F. J. Baldán and J. M. Benítez, "Multivariate times series classification through an interpretable representation," *Inf Sci (N Y)*, vol. 569, pp. 596–614, Aug. 2021, doi: 10.1016/j.ins.2021.05.024.

[69] B. Shneiderman, "The eyes have it: a task by data type taxonomy for information visualizations," in *Proceedings 1996 IEEE Symposium on Visual Languages*, IEEE Comput. Soc. Press, pp. 336–343. doi: 10.1109/VL.1996.545307.

[70] H. Hochheiser and B. Shneiderman, "Dynamic Query Tools for Time Series Data Sets: Timebox Widgets for Interactive Exploration," *Inf Vis*, vol. 3, no. 1, pp. 1–18, Mar. 2004, doi: 10.1057/palgrave.ivs.9500061.

[71] C. Plaisant, R. Mushlin, A. Snyder, J. Li, D. Heller, and B. Shneiderman, "LifeLines: using visualization to enhance navigation and analysis of patient records.," *Proceedings / AMIA ... Annual Symposium. AMIA Symposium*, pp. 76–80, Jan. 1998, doi: 10.1016/b978-155860915-0/50038-x.

[72] T. D. Wang *et al.*, "Temporal summaries: Supporting temporal categorical searching, aggregation and comparison," in *IEEE Transactions on Visualization and Computer Graphics*, Nov. 2009, pp. 1049–1056. doi: 10.1109/TVCG.2009.187.

[73] K. Wongsuphasawat, J. A. G. Gómez, C. Plaisant, T. D. Wang, S. Ben, and M. Taieb-Maimon, "LifeFlow: Visualizing an overview of event sequences," in *Conference on Human Factors in Computing Systems - Proceedings*, Association for Computing Machinery, 2011, pp. 1747–1756. doi: 10.1145/1978942.1979196.

[74] P. Buono and M. F. Costabile, "Insights on the development of visual tools for analysis of pollution data".

[75] B. Johnson Ben Shneiderman, bri anj, and cs md ed, "Tree-Maps: A Space-Filling Approach to the Visualization of Hierarchical Information Structures".

[76] B. Shneiderman, "Tree visualization with tree-maps," *ACM Transactions on Graphics (TOG)*, vol. 11, no. 1, pp. 92–99, Jan. 1992, doi: 10.1145/102377.115768.

[77] M. Bostock, V. Ogievetsky, and J. Heer, "D3 Data-Driven Documents," *IEEE Trans Vis Comput Graph*, vol. 17, no. 12, pp. 2301–2309, Dec. 2011, doi: 10.1109/TVCG.2011.185.

[78] D. J. Díaz Pérez, "Herramienta visual para exploración de datos en una cohorte de niños canguro durante su primer año de vida," Universidad de los Andes, 2017.

[79] J. M. Perkel, "Reactive, reproducible, collaborative: computational notebooks evolve," *Nature*, vol. 593, no. 7857, pp. 156–157, May 2021, doi: 10.1038/D41586-021-01174-W.

[80] H. Liu and C. North, "Case Study Comparison of Computational Notebook Platforms for Interactive Visual Analytics," *Proceedings - 2022 IEEE Visualization in Data Science, VDS 2022*, pp. 1–5, 2022, doi: 10.1109/VDS57266.2022.00005.

[81] "Neurolucida® - MBF Bioscience." Accessed: Sep. 22, 2023. [Online]. Available: https://www.mbfbioscience.com/products/neurolucida/

[82] "Imaris for Neuroscientists - Imaris - Oxford Instruments." Accessed: Sep. 22, 2023. [Online]. Available: https://imaris.oxinst.com/products/imaris-for-neuroscientists

[83] "NLMorphologyconverter - NLMorphologyConverter: Format Status." Accessed: Jul. 22, 2023. [Online]. Available: http://neuronland.org/NLMorphologyConverter/FormatStatus.html

[84] G. Eyal *et al.*, "Unique membrane properties and enhanced signal processing in human neocortical neurons," *Elife*, vol. 5, no. OCTOBER2016, Oct. 2016, doi: 10.7554/ELIFE.16553.

[85] R. M. Haralick, S. R. Sternberg, and X. Zhuang, "Image Analysis Using Mathematical Morphology," *IEEE Trans Pattern Anal Mach Intell*, vol. PAMI-9, no. 4, pp. 532–550, 1987, doi: 10.1109/TPAMI.1987.4767941.

[86] W. E. Lorensen and H. E. Cline, "Marching cubes: A high resolution 3D surface construction algorithm," *ACM SIGGRAPH Computer Graphics*, vol. 21, no. 4, pp. 163–169, Aug. 1987, doi: 10.1145/37402.37422.

[87] P. Cignoni, C. Rocchini, and R. Scopigno, "Metro: Measuring Error on Simplified Surfaces," *Computer Graphics Forum*, vol. 17, no. 2, pp. 167–174, 1998, doi: 10.1111/1467-8659.00236.

[88] I. Daubechies, "The Wavelet Transform, Time-Frequency Localization and Signal Analysis," *IEEE Trans Inf Theory*, vol. 36, no. 5, pp. 961–1005, 1990, doi: 10.1109/18.57199.

[89] D. B. Percival and A. T. Walden, "Wavelet Methods for Time Series Analysis," *Wavelet Methods for Time SeriesAnalysis*, 2000, doi: 10.1017/CBO9780511841040.

[90] I. Daubechies, "The Wavelet Transform, Time-Frequency Localization and Signal Analysis," *IEEE Trans Inf Theory*, vol. 36, no. 5, pp. 961–1005, 1990, doi: 10.1109/18.57199.

[91] R. S. Stankovir and B. J. Falkowski, "The Haar wavelet transform: its status and achievements," *Computers & Electrical Engineering*, vol. 29, no. 1, pp. 25–44, Jan. 2003, doi: 10.1016/S0045-7906(01)00011-8.

[92] E. A. Maharaj, P. D'Urso, and D. U. A. Galagedera, "Wavelet-based Fuzzy Clustering of Time Series," *J Classif*, vol. 27, no. 2, pp. 231–275, Sep. 2010, doi: 10.1007/S00357-010-9058-4/METRICS.

[93] W. R.Klecka, "Discriminant Analysis," 1980, doi: 10.4135/9781412983938.

[94] "Cartool Community." Accessed: Sep. 20, 2023. [Online]. Available: https://sites.google.com/site/cartoolcommunity/

[95] D. Cruse *et al.*, "Bedside detection of awareness in the vegetative state: A cohort study," *The Lancet*, vol. 378, no. 9809, pp. 2088–2094, Dec. 2011, doi: 10.1016/S0140-6736(11)61224-5.

[96] D. P. Mandic, N. Ur Rehman, Z. Wu, and N. E. Huang, "Empirical mode decomposition-based time-frequency analysis of multivariate signals: The power of adaptive data analysis," *IEEE Signal Process Mag*, vol. 30, no. 6, pp. 74–86, 2013, doi: 10.1109/MSP.2013.2267931.

[97] P. Gaur, R. B. Pachori, H. Wang, and G. Prasad, "A multivariate empirical mode decomposition based filtering for subject independent BCI," in *2016 27th Irish Signals and Systems Conference, ISSC 2016*, Institute of Electrical and Electronics Engineers Inc., Aug. 2016. doi: 10.1109/ISSC.2016.7528480.

[98] F. Hartwig and B. Dearling, *Exploratory Data Analysis*. 2455 Teller Road, Newbury Park California 91320 United States of America : SAGE Publications Inc., 1979. doi: 10.4135/9781412984232.

[99] G. J. Myatt, *Making Sense of Data*. Wiley, 2007. doi: 10.1002/0470101024.

[100] "IBM Corp. Released 2021. IBM SPSS Statistics for Windows, Version 28.0. Armonk, NY: IBM Corp."

[101] "Welcome to IEEE VIS 2023!" Accessed: Sep. 22, 2023. [Online]. Available: https://ieeevis.org/year/2023/welcome