Universidad Nacional Autónoma de México

Instituto de Neurobiología

Algoritmo para la clasificación automática de potenciales de acción registrados extracelularmente

Tesis que para obtener el grado de
Maestro en Ciencias (Neurobiología)
presenta
el I.S.E. Jorge Arturo Gámez de León.

Director de Tesis
Dr. Hugo Merchant Nancy

Campus Juriquilla, Querétaro. Diciembre 2010

Universidad Nacional Autónoma de México

Instituto de Neurobiología

Los miembros del Comité Tutoral certificamos que la tesis elaborada por: Jorge Arturo Gámez de León, cuyo título es: "Algoritmo para la clasificación automática de potenciales de acción registrados extracelularmente" se presenta como uno de los requisitos para obtener el grado de Maestría en Ciencias (Neurobiología) y cumple con los criterios de originalidad y calidad requeridos por la División de Estudios de Posgrado de la Universidad Nacional Autónoma de México.

Firma

Presidente
Dr. Ranulfo Romo Trujillo _____

Secretario (Tutor)
Dr. Hugo Merchant Nancy _____

Vocal
Dr. Gerardo Rojas Piloni _____

Suplente
Dr. Rogelio Arellano Ostoa _____

Suplente
Dr. Fernando Peña Ortega _____

Aprobado por el comité académico

_____

Coordinador del programa

Resumen

ALGORITMO PARA LA CLASIFICACIÓN AUTOMÁTICA DE POTENCIALES DE ACCIÓN REGISTRADOS EXTRACELULARMENTE

La detección y análisis de la actividad neuronal es el punto de partida para entender diversas funciones cerebrales. Hasta ahora, los registros extracelulares son la opción más práctica para las preparaciones in vivo. En el Sistema Nervioso Central, los electrodos extracelulares reciben señales de varias neuronas al mismo tiempo, por lo que es de suma importancia poder asignar cada espiga registrada a la neurona que la generó.

En este trabajo, se propone el uso de un algoritmo de detección basado en la transformada de ondeleta y dos algoritmos de clasificación segmentados en el tiempo basados en el algoritmo de Expectación-Maximización (EM); así como diferentes versiones de métodos para la estimación del número de grupos presentes en un conjunto de datos. Se utilizó un algoritmo genético para encontrar los parámetros óptimos en cada uno de los pasos del algoritmo de detección. Posteriormente, se realizaron comparaciones entre diferentes algoritmos, tanto en señales artificiales como en registros extracelulares que se obtuvieron de la corteza premotora medial de monos rhesus (*Macaca mulatta*). Una metodología propuesta para la comparación de algoritmos de discriminación se implementó en un programa para Matlab. Este programa nombrado Sort Lab, tiene entre sus funciones crear y editar algoritmos de discriminación, medir y comparar el rendimiento de estos algoritmos, y crear señales artificiales basadas en simulaciones de redes neuronales, para ser usadas en la evaluación de los algoritmos.

El uso del Sort Lab mostró que el algoritmo basado en la transformada de ondeleta fue el mejor en la detección de espigas, tanto en la señales artificiales, como en la reales. Mientras que, el algoritmo de EM segmentado en el tiempo por agrupación de centroides propuesto en este trabajo, obtuvo en promedio los mejores resultados en la etapa de clasificación.

Summary

ALGORITHM FOR AUTOMATIC SORTING OF EXTRACELLULAR ACTION POTENTIALS.

The detection and analysis of the neuronal activity is the starting point for the understanding of various brain functions. Until now, extracellular recordings are the most practical option for in vivo preparations. In the Central Nervous System, extracellular electrodes can receive information from several neurons at the same time, therefore it is of utmost importance to assign each spike to the neuron that generated it.

In this work, we propose the use of a spike detection algorithm based on the wavelet transform, and two time sliced algorithms based on the Expectation-Maximization (EM) algorithm; in addition to different methods to estimate the number of groups present in a dataset. We used a genetic algorithm to find the optimal parameters for each of the steps of the detection algorithm. Subsequently, we compared the performance among different detection and classification algorithms, using artificial and real signals. The real signals were extracellular recordings from the medial premotor cortex of a rhesus monkey (*Macaca mulatta*). We implemented in Matlab a methodology to benchmark the performance of spike sorting algorithms. This software was named Sort Lab, it can create and edit spike sorting algorithms, measure their performance, and create artificial signals using the information from a neural network simulator.

Using Sort Lab, we found out that our wavelet transform based algorithm was the best detection algorithm in artificial and real signals. Moreover, the time sliced EM algorithm by centroids clustering, was in average the best algorithm for the classification phase.

Agradecimientos

A la Dirección General de Estudios de Posgrado de la UNAM. Número de cuenta: 509003223.

Al Consejo Nacional de Ciencia y Tecnología. Becario número: 220938.

Al Dr. Hugo Merchant por su dirección y sus enseñanzas.

A mi familia por su apoyo incondicional.

Al personal de la Unidad de Enseñanza, especialmente a la M.C. Leonor Casanova Rico, Jefa de la Unidad de Enseñanza.

Al personal de la Biblioteca del campus UNAM Juriquilla que es encabezado por el Dr. Francisco Javier Valles Valenzuela como coordinador.

Índice general

Capítulo 1

Introducción

La detección y análisis de la actividad neuronal es el punto de partida para entender diversas funciones cerebrales. La mayoría de las neuronas se comunican con potenciales de acción. Esta actividad puede ser registrada a través de electrodos ya sean intracelulares o extracelulares. Siendo estos últimos la única opción práctica para preparaciones *in vivo* hasta ahora. La detección y clasificación de las espigas en los registros extracelulares han demostrado ser todo un reto técnico (Lewickiy, 1998; Brown et al., 2004; Thakur et al., 2007). Los electrodos usados en estos registros usualmente reciben información de varias neuronas vecinas, por lo que es de suma importancia poder asignar estas espigas a la neurona correspondiente. La suma de las señales de muy baja intensidad provenientes de otras neuronas que son registradas únicamente como ruido, aunado a las fuentes de campos electromagnéticos presentes al momento del registro, aumentan la complejidad del análisis de las señales. Se pueden usar métodos manuales para medir la actividad neuronal en cada canal del registro. Sin embargo, la clasificación automática puede reducir de manera significativa el tiempo dedicado a esta actividad y al mismo tiempo mejorar la precisión del análisis. Una ventaja adicional de los algoritmos de clasificación de espigas, es la capacidad de distinguir la actividad de varias neuronas aun cuando estas disparen al mismo tiempo. Esta característica es especialmente importante en la investigación de los códigos neuronales que se basan en los tiempos de disparo (Lewickiy, 1998).

Es importante entender como se generan las señales eléctricas en el sistema nervioso. Las células nerviosas son las unidades de la transmisión de señales en el sistema nervioso. Una neurona típica se compone de cuatro regiones principales: el cuerpo o soma, las dendritas, el axón y los botones terminales. El soma es el centro metabólico de la célula, contiene el núcleo donde se almacena la información genética, así como el retículo endoplasmático, donde se sintetizan las proteínas de la célula. Del cuerpo de la célula se proyectan dos procesos: ramificaciones de dendritas y una axón largo. Las dendritas cumplen con la acción de recibir señales de otras neuronas, en cambio el axón se proyecta lejos del soma y es la unidad principal encargada de transmitir la señal a otras neuronas. Un axón puede llevar señales eléctricas en distancias desde 0.1mm hasta 3m. Estas señales eléctricas son llamadas potenciales de acción, son impulsos rápidos, temporales y responden a una regla de "todo o nada", tienen una amplitud de 100mV y una duración de 1ms aproximadamente. Los potenciales de acción se inician en una región especializada del axón, llamado el cono axonal. Los potenciales de acción constituyen las señales por las cuales el cerebro recibe, analiza y procesa información. Estas señales son muy parecidas en todo el sistema nervioso, aun cuando sean iniciadas por una gran variedad de eventos, de aquí que la información que lleva el potencial de acción no es determinada por la forma de la señal, sino por el circuito que activa, así como por la estructura temporal de los trenes de potenciales de acción.

El potencial de acción es definido principalmente por las propiedades de los canales de sodio y potasio (ver figura 1). La despolarización de la membrana causa que los canales de sodio se abran rápidamente, lo que se traduce en un aumento en la permeabilidad de la membrana al sodio, generando un corriente de sodio de entrada. Esta corriente descarga la capacitancia de la membrana, lo que resulta en una mayor despolarización que a su vez provoca la apertura de más canales de sodio, por lo que se genera un aumento de la corriente entrante. Este proceso lleva al potencial de membrana hacia el voltaje de equilibrio del sodio,

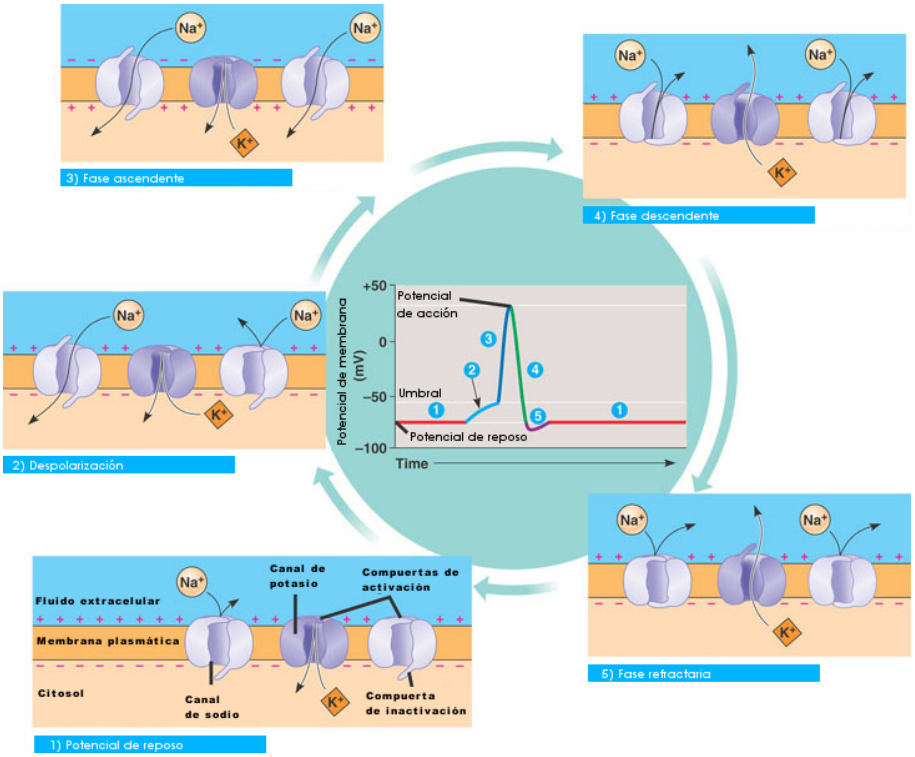


FIGURA 1. Esquema de la generación de un potencial de acción (modificado de Gallant (2008)). 1) La membrana se encuentra en su potencial de reposo. 2) La membrana se despolariza, se empiezan a abrir los canales de sodio, provocando una corriente de entrada de sodio. 3) Se genera una despolarización mayor de la membrana lo que incrementa el número de canales de sodio abierto y en consecuencia un aumento de la corriente entrante de sodio. Este aumento en la corriente de sodio, lleva a la membrana hacia el potencial de reposo del sodio. 4) Los canales de sodio se inactivan y se abren los canales de potasio dependientes de voltaje. Se detiene la corriente de entrada de sodio y aparece una corriente de salida de potasio. 5) La corriente de salida de potasio lleva a la membrana a un estado de hiperpolarización.

causando la fase ascendente del potencial de acción. El estado de despolarización del potencial se limita por la inactivación gradual de los canales de sodio y la apertura de los canales de potasio dependientes de voltaje, incrementando la permeabilidad de la membrana al potasio. Consecuentemente, la corriente de entrada de sodio es seguida por una corriente de salida del potasio que tiende a repolarizar la membrana. En la mayoría de las células nerviosas el estado de despolarización es seguido por una hiperpolarización de la membrana. El cual se debe a que los canales abiertos de potasio se cierran hasta cierto tiempo después de que el potencial de membrana ha regresado a su estado de reposo. El potencial de acción es seguido por un breve periodo donde se disminuye la excitabilidad de la célula, conocido como periodo refractario, durante el cual la generación de un nuevo potencial de acción requiere de un estímulo de mayor magnitud del que normalmente sería requerido. Este periodo refractario es causado por la inactivación residual de los canales de sodio y un aumento en los canales abiertos de potasio (Kandel et al., 2000).

La primer relación entre la comunicación neuronal y las señales eléctricas fue planteada por Luigi Galvani en 1791, pero no fue hasta la década de 1920 cuando los impulsos nerviosos pudieron ser medidos, amplificando las señales registradas por medio de electrodos. El circuito básico de registro consiste en amplificar la diferencia de potencial entre una referencia y un micro-electrodo. Los cambios de potencial medidos en la punta del electrodo reflejan un flujo de corriente en el medio extracelular. Usualmente el mayor componente de esta corriente es el generado por el potencial de acción, pero puede haber otras señales involucradas. Señales que lucen muy parecidas a los potenciales de acción son generadas por las fibras de conjuntos de axones. Estas son mucho más localizadas y pequeñas que los potenciales de acción. Otra fuente de señales es el potencial de campo, el cual usualmente aparece en las estructuras organizadas en capas y es resultado del flujo sincronizado de corriente en los conjuntos de dendritas paralelas. Esta señal tiene un ancho de banda angosto, por lo que es fácil filtrarla del registro.

En los registros extracelulares la forma de la punta del electrodo usado afecta la señal obtenida, entre más grande la punta mayor será el número de señales registradas. Si la punta del electrodo es demasiado grande, puede ser imposible aislar la señal de una neurona específica. En cambio, si la punta es demasiado pequeña puede ser difícil registrar una señal. Un electrodo de vidrio con una punta redonda obtendrá una señal diferente que uno de platino-iridio (Pt-Ir) cubierto de vidrio con forma de "bala". Una vez que se ha obtenido una señal del electrodo, el siguiente paso del proceso es amplificarla y finalmente realizar un análisis de la misma, que puede hacerse manualmente con ayuda de un detector de nivel o con algoritmos de discriminación de espigas (Lewickiy, 1998).

También existen los registros intracelulares, donde se inserta un electrodo en el cuerpo de la neurona o se realiza la técnica de *patch-clamp* para tener acceso directo al interior de la célula. Estas técnicas permiten obtener información sobre las variaciones sub-umbrales del potencial de membrana. Además de que con la información obtenida, se puede identificar la morfología de la neurona registrada. Sin embargo, en las técnicas intracelulares sólo se obtienen registros de una célula a la vez.

Existe una relación entre la señal registrada intracelularmente y el registro extracelular de la misma célula, tal como lo demostró Heneze et al. (2000). La fase ascendente del registro extracelular sigue a la primera derivada del potencial de acción intracelular. Sin embargo, la fase descendente es más prolongada en el registro extracelular debido a una suma de corrientes adicionales que no son proporcionales a la primer

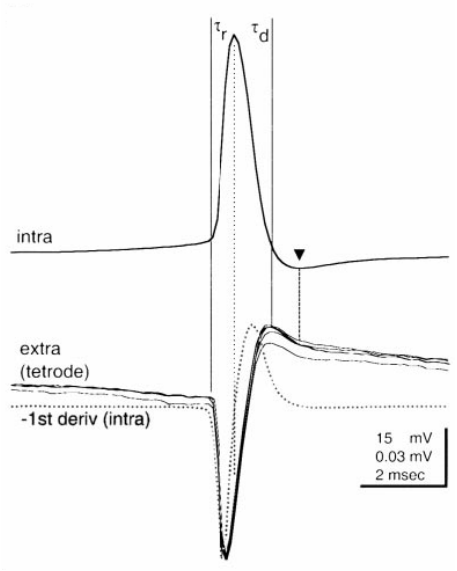


FIGURA 2. Comparación de un registro intracelular, extracelular y la primer derivada de la señal intracelular (tomado de Heneze et al. (2000)).

derivada de la señal intracelular, como las de las corrientes activas iónicas y las corrientes de "fuga" (ver figura 2).

El registro de la señal extracelular tiene variaciones dependiendo de la localización del electrodo con respecto a la región más cercana de la neurona (ver figura 3). La señal tiene una mayor amplitud entre más cerca del soma se haga el registro y disminuye cerca de los árboles dendríticos. Heneze et al. (2000) sugiere que el potencial de acción en las neuronas del hipocampo usualmente se inicia en el soma y se propaga a las dendritas, pudiendo haber situaciones en que los potenciales de acción se inicien en las dendritas durante periodos de alta actividad sináptica.

El registro extracelular se puede usar para medir la longitud de una espiga, siendo esta información útil para estimar si la neurona registrada es de tipo piramidal o se trata de una interneurona. Estas últimas generalmente tienen potenciales de acción más angostos que las células piramidales registradas en condiciones similares (Csicsvari et al., 1999).

Existen una gran variedad de algoritmos para la discriminación de espigas, sin embargo no existe un consenso de cuál es el mejor. Diferentes algoritmos aplicados al mismo conjunto de datos pueden generar diferentes resultados (Brown et al., 2004). Los principales problemas en el proceso de clasificación de las espigas tienen que ver con el ruido en la señal y con la presencia y variabilidad de las señales de las neuronas registradas. Las características de las espigas de una neurona cambian en un mismo registro debido a las características intrínsecas neuronales y las condiciones experimentales. Debido a la variabilidad en las formas de las espigas, puede haber un traslape en las características de las mismas, por lo que la clasificación de espigas tiene una tasa de error mayor a cero. Sin embargo, la clasificación manual es más propensa a errores que la clasificación automática (Harris et al., 2000).

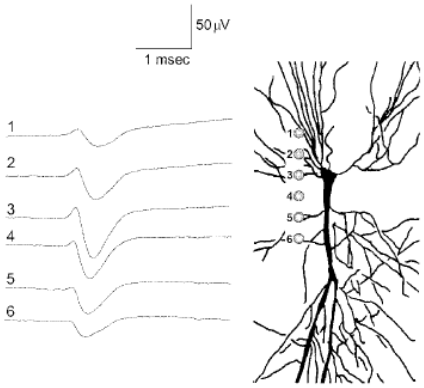


FIGURA 3. Comparación de registros extracelulares obtenidos en cercanía de diferentes regiones de una neurona (tomado de Heneze et al. (2000)).

Capítulo 2

Antecedentes

Las fases principales de un algoritmo de clasificación de espigas son: 1) detección de espigas, 2) selección de las características de las espigas y 3) agrupamiento de las características de las espigas seleccionadas (Quian Quiroga y Nadasdy, 2004). Sin embargo, un paso anterior a la clasificación de espigas es el preproceso de la señal. En este paso se busca mejorar la señal para su posterior análisis. Principalmente se realizan dos acciones: filtrado de la señal, para quitar los artefactos del medio ambiente que hayan afectado el registro, con métodos como ondeletas (Chan et al., 2008b); y el aislamiento de la misma, para eliminar el ruido en modo común que puede aparecer en todos los canales, con métodos como el Análisis de Componentes Independientes o ICA por sus siglas en inglés (Snellings et al., 2006; Hermle et al., 2005; Mamlouk et al., 2005; Takahashi y Sakurai, 2005; Takahashi et al., 2003b).

Muchos de los métodos de clasificación se basan en la discriminación por las características de las espigas (Thakur et al., 2007; Chan et al., 2008a; Wood y Black, 2007; Vargas-Irwin y Donoghue, 2007; Horton et al., 2007). En un inicio se clasificaron las espigas utilizando características seleccionadas con anterioridad como la amplitud, duración o amplitud pico-pico. Sin embargo, el asumir cuales características serán importantes para la clasificación suele producir resultados pobres. Un método comúnmente usado para seleccionar las características importantes es el Análisis de Componentes Principales o PCA por sus siglas en inglés, cuya idea de fondo es ordenar las diferentes características de las espigas por vectores ortogonales que representan las direcciones en los datos de mayor variación (Lewickiy, 1998; Shlens, 2005). Una vez realizado el análisis de las espigas se procede a clasificarlas en grupos, esto se puede hacer por algoritmos relativamente sencillos como son:

- El *K-medias*, donde se define la localización del grupo como la media de los datos pertenecientes al grupo (Chan et al., 2008a; Thakur et al., 2007; Vargas-Irwin y Donoghue, 2007; Takahashi et al., 2003a; Chan et al., 2008b).
- El agrupamiento bayesiano, que es un modelo estadístico donde se clasifica la posibilidad de que una espiga pertenezca al grupo (Bar-Hillel et al., 2006).
- Los métodos de comparación con plantillas de espigas, donde se generan o se seleccionan previamente las espigas "ideales", para cada espiga del registro se busca el mayor parecido con las plantillas ya sea por la distancia euclidiana o por una clasificación bayesiana (Vargas-Irwin y Donoghue, 2007; Delescluse y Pouzat, 2006; Wood y Black, 2007).
- Los algoritmos basados en filtros, donde se busca generar un conjunto de filtros que sean capaces de discriminar un conjunto de espigas entre ellas y del ruido de fondo, cada filtro se construye de manera que genere la mayor respuesta a la forma de espiga esperada y la menor al ruido de fondo,

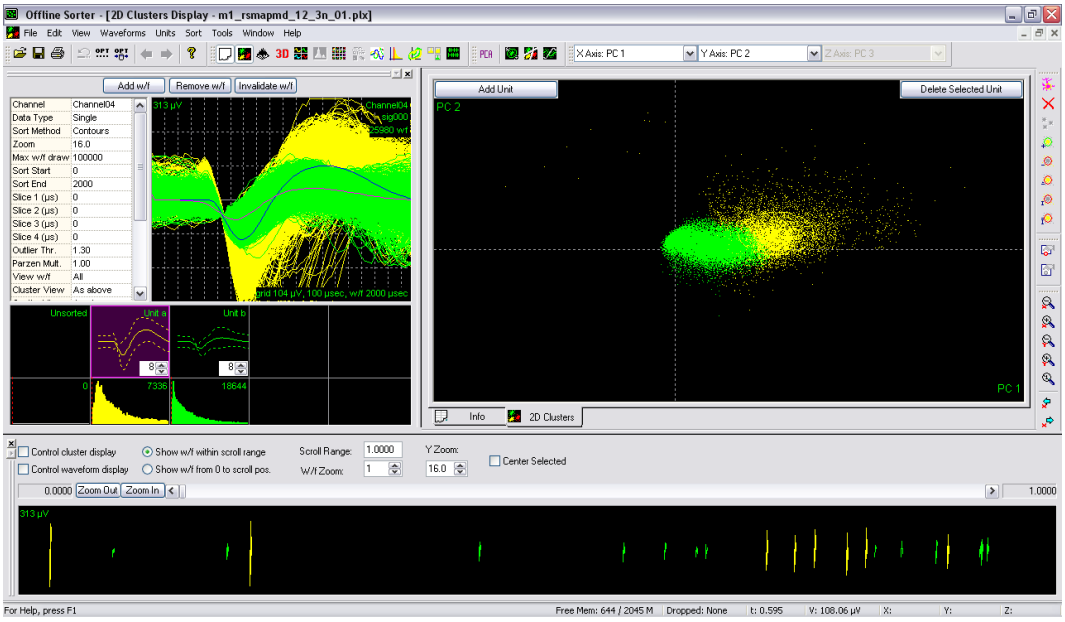


FIGURA 4. Imagen del programa de clasificación de espigas semiautomático Plexon.

al pasar la señal por los filtros se asigna la espiga al filtro con el que se obtenga la mayor respuesta (Lewickiy, 1998).

También existen los algoritmos de clasificación basados en redes neuronales, donde la red se entrena para reconocer características de las espigas y se obtiene la clasificación en grupos como salida (Hermle et al., 2005; Horton et al., 2007).

Hasta ahora es común que la clasificación de espigas se lleve a cabo por medios manuales o semiautomáticos con ayuda de programas especializados. Es común utilizar PCA para convertir las espigas a un espacio de características, donde se pueden representar en dos o tres dimensiones utilizando los dos o tres componentes principales respectivamente. Después se puede aplicar un algoritmo de agrupamiento, como *K-medias* para obtener grupos estimados, finalmente el operador terminará de clasificar las espigas de manera visual (ver figura 4)(Plexon, 2006).

2.1. Filtrado de la señal

El filtrado de una señal usualmente tiene el objetivo de remover las partes de la señal que son consideradas ruido. El funcionamiento de un filtro se puede describir en el dominio del tiempo o la frecuencia, sin embargo su comportamiento en frecuencia es más fácil de comprender. En el dominio de la frecuencia un filtro es un atenuador de ciertas frecuencias indeseables, mientras que permite el paso de otras. Idealmente las frecuencias deseadas no deben ser atenuadas, mientras que las indeseables deben ser completamente eliminadas y la transición entre una región y la otra debe ser de longitud cero. Sin embargo, de manera práctica, las bandas de paso sufren cierta atenuación, mientras que las bandas de corte no son completamente eliminadas. De igual forma, la proporción de amplitud puede mostrar ondulaciones a diferentes

frecuencias y la región de transición tiene una longitud mayor a cero (van Drongelen, 2007). Todos estos efectos provocan que la señal de salida real tenga alteraciones que pueden ser visibles en el dominio del tiempo.

De acuerdo con su comportamiento en el dominio de la frecuencia, un filtro se puede clasificar como: pasa-bajas, el cual permite el paso de frecuencias menores a cierta frecuencia de corte; pasa-altas, que permite el paso de frecuencias mayores a la frecuencia de corte; pasa-banda, que permite el paso de cierto rango de frecuencias; y rechaza-banda, que impide el paso de cierto rango de frecuencias (van Drongelen, 2007). Los filtros y sus parámetros deben ser sujetos a un análisis de acuerdo a las características de la señal que será filtrada. En el caso de las señales electrofisiológicas del sistema nervioso se busca la máxima atenuación del ruido y la mínima modificación de las formas de onda de las espigas. Es importante evitar en lo posible la alteración de la forma de onda de las espigas, debido a que se ha propuesto que las diferencias entre ellas se pueden usar para distinguir entre neuronas piramidales e inhibitorias (Csicsvari et al., 1999).

El comportamiento de un filtro se puede analizar gráficamente por medio de un diagrama de Bode, éste consiste en una gráfica del comportamiento de un sistema con respecto a la frecuencia. Un diagrama de Bode usualmente consiste en dos gráficas separadas. Una gráfica representa la amplitud de la respuesta del sistema con respecto a la frecuencia y en una segunda gráfica se muestra la fase de la función de transferencia que define al sistema. Es común utilizar el primer tipo de gráfica para analizar el comportamiento de un filtro con respecto a la frecuencia de la señal de entrada. En la figura 5 se puede ver una comparación de las características de ganancias de diversos filtros pasa-bajas en un rango de frecuencias de 0 a π radianes.

2.1.1. Filtros Butterworth. La forma general de la respuesta a la frecuencia de la magnitud de un filtro Butterworth es

$$| H(\omega) | = \frac{1}{\sqrt{1 + (\frac{\omega}{\omega_c})^{2N}}} \tag{2.1.1}$$

donde N es conocida como el "orden" del filtro. En otras palabras N es el orden de la ecuación diferencial necesaria para describir el comportamiento dinámico del filtro en el dominio del tiempo (Phillips et al., 2007).

Los filtros Butterworth son llamados máximamente planos en la banda de paso, debido a que, para un orden dado, tienen la mayor pendiente descendente después de la frecuencia de corte, sin inducir picos en su gráfica de Bode. Un filtro con una mayor pendiente permite que se filtren las frecuencias no deseadas con mayor precisión, mientras que la ausencia de ondulaciones en la banda de paso disminuye las deformaciones en la señal de salida (ver figura 6).

2.1.2. La transformada continua de ondeleta. La transformada de ondeleta es una función matemática que permite analizar datos al dividirlos en segmentos de frecuencia-tiempo (Saavedra et al., 2006). En la transformada continua de ondeleta (CWT por sus siglas en inglés) se utiliza una función, conocida como ondeleta madre, que es escalada y trasladada para calcular su convolución con la función original (ver figura 7). La transformada continua de ondeleta se define como:

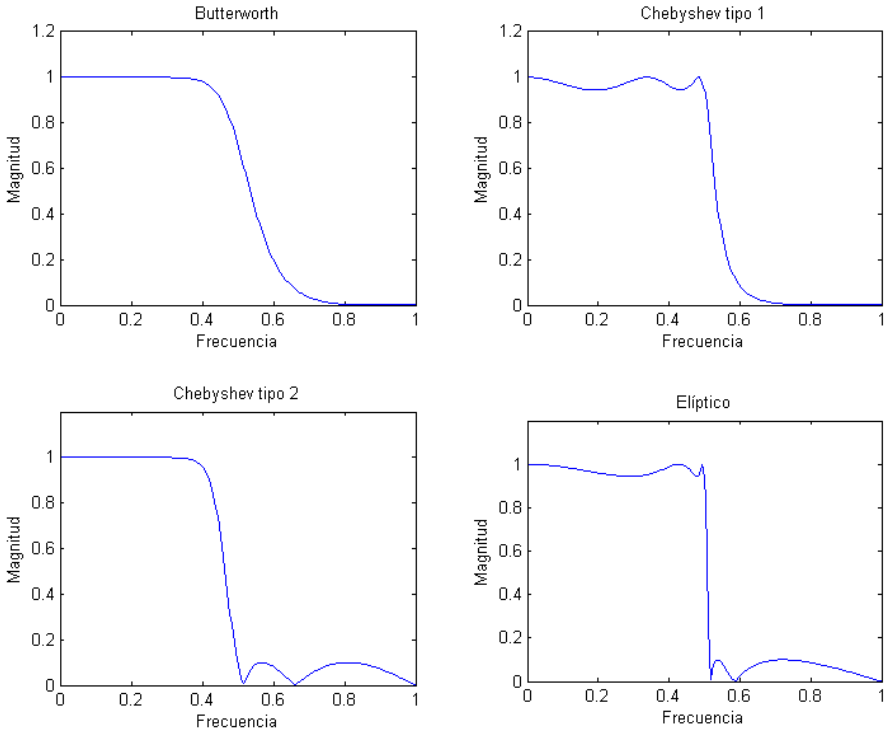


FIGURA 5. Comparación de respuesta en frecuencia de diversos filtros de quinto orden con frecuencia de corte normalizada en $\frac{\pi}{2}$. El filtro Butterworth tiene la mayor pendiente sin inducir picos en su gráfica de frecuencia.

$$CWT_\psi f(a,b) = W_f(b,a) = |a|^{-\frac{1}{2}} \int_{-\infty}^{\infty} f(t)\psi^*(\frac{t-b}{a})dt \tag{2.1.2}$$

Donde $a, b \, \epsilon \mathbb{R}$, $a \neq 0$. a y b son conocidos como coeficientes de dilatación y traslación respectivamente. La función $\psi(t)$ es la ondeleta madre. La transformada continua de ondeleta descompone una señal en diferentes escalas con diferentes niveles de resolución. Es esta descomposición lo que permite obtener la función original acotada en el dominio de la frecuencia, de tal forma que el resultado es un filtro pasa-banda basado en la transformada de ondeleta.

2.2. Fourier

La transformada de Fourier permite convertir señales continuas en sus componentes sinusoidales. La descomposición de una señal en sus componentes sinusoidales, permite conocer el espectro de frecuencias que contiene la señal. Esta información es muy usada con el fin de aislar los componentes de la señal deseada del ruido.

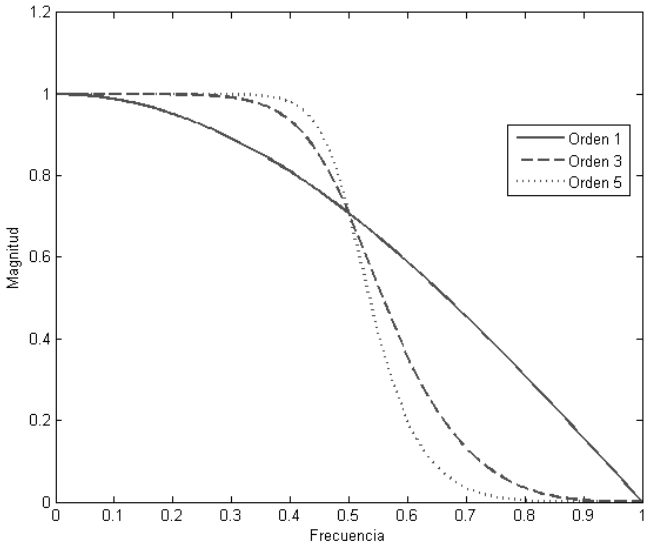


FIGURA 6. Comparación del comportamiento de filtros Butterworth de orden 1,3 y 5.

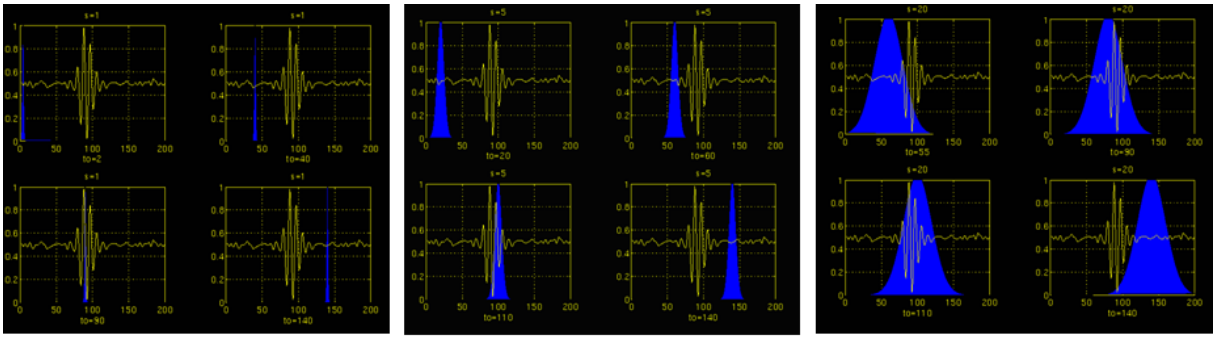


FIGURA 7. Transformada continua de ondeleta (tomado de Polikar (2001)). Ejemplo gráfico de la transformada continua de ondeleta, donde la onda azul es la ondeleta madre y la señal original está en amarillo. La ondeleta madre se escala incrementando su tamaño y se traslada en el tiempo sobre toda la señal original para cada escala.

Una serie de Fourier es la representación de una función periódica $f(x)$ como una combinación lineal de todas las funciones coseno y seno que tienen el mismo periodo, como en

$$f(x) = \frac{1}{2}a_0 + \sum_{n=1}^{\infty} a_n cos\frac{n\pi x}{l} + \sum_{n=1}^{\infty} b_n sin\frac{n\pi x}{l}. \tag{2.2.1}$$

La misma serie puede ser escrita en su forma compleja como

$$f(x) = \sum_{n=-\infty}^{\infty} c_n e^{\frac{in\pi x}{l}}, \tag{2.2.2}$$

donde $c_n = \frac{1}{2}(a_n - b_n)$ para toda n, si a_{-n} significa a_n y b_{-n} significa $-b_n$, de tal manera que b_0= 0(Lighthill, 1958).

La integral de Fourier se puede considerar como el límite formal de la serie de Fourier cuando el periodo tiende a infinito. De tal forma, que si $f(x)$ es cualquier función de x en el rango $(-\infty, \infty)$, se puede formar la función $f_l(x)$ de periodo $2l$ que coincide con $f(x)$ en el rango $(-l, l)$. La serie de Fourier (2.2.2) de $f_l(x)$ se puede escribir como

$$f_1(x) = \sum_{n=-\infty}^{\infty} e^{\frac{in\pi x}{l}} g_l(\frac{n}{2l})\frac{I}{2l}\,, \text{ donde } g_l(y) = \int_{-l}^{l} e^{-2\pi ixy} f(x)dx. \tag{2.2.3}$$

El limite formal cuando $l \to \infty$, donde en la serie $\frac{n}{2l}$ se escribe como y, y la diferencia entre valores sucesivos de y se escribe como dy, es

$$f(x) = \int_{-\infty}^{\infty} e^{2\pi ixy} g(y)dy\,, \text{ donde } g(y) = \int_{-\infty}^{\infty} e^{-2\pi ixy} f(x)dx, \tag{2.2.4}$$

dado que en el límite la función periódica $f_l(x)$ se convierte en $f(x)$.

Bajo estas circunstancias la función $g(y)$ es comúnmente llamada la transformada de Fourier de $f(x)$ (Lighthill, 1958).

Sin embargo las señales de los registros electrofisiológicos son digitales, por lo que se utiliza la contraparte digital de la transformada de Fourier llamada transformada de Fourier discreta (DFT). La transformada rápida de Fourier (FFT) es un método eficiente para calcular la DFT de una señal.

2.3. Teoría de detección de señales

El beneficio práctico más inmediato de la teoría de detección de señales (TDS), es que provee un conjunto de medidas útiles del desempeño en situaciones de toma de decisiones (McNicol, 2005). Las raíces de la teoría se remontan al trabajo de Neyman y Pearson en 1933, donde estudiaron pruebas de hipótesis e inferencias estadísticas. Los primeros trabajos donde se aplicó la teoría de detección de señales fueron en señales de radar, aunque rápidamente también fue aplicada en las áreas de psicología y medicina. La TDS se utiliza cuando una decisión debe ser tomada en un contexto de incertidumbre con respecto a las variables

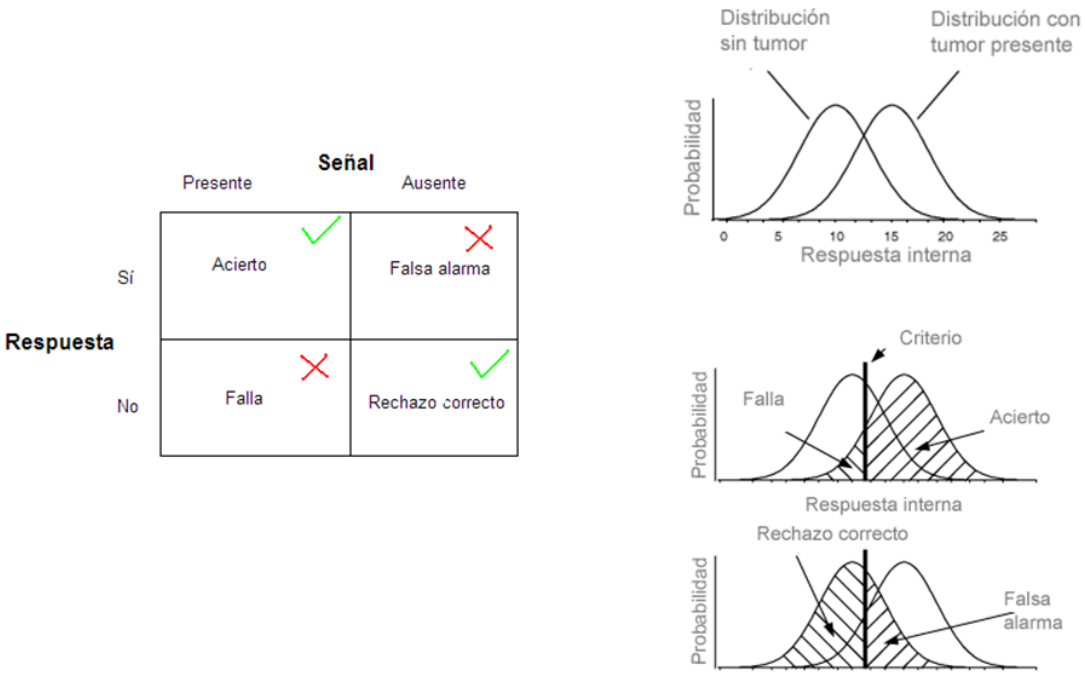


FIGURA 8. Matriz de confusión y modelo de la teoría de detección de señales. En este ejemplo la señal es una tomografía y el observador debe clasificar entre imágenes que contienen un tumor de aquellas que no lo contienen. Del lado izquierdo está la matriz de confusión con las posibles combinaciones de la respuesta contra la presencia de la señal. Del lado derecho están las gráficas de las distribuciones de la señal y el ruido, así como el criterio de decisión.

que afectan la decisión. Green y Swets (1988), aplicaron la TDS en el área de la percepción humana y en el proceso de toma de decisiones. En su estudio, los participantes debían discriminar entre un estímulo con cierta característica objetivo (señal) y el estímulo sin la característica (ruido). La TDS fue usada en un experimento auditivo, donde se les presentaba a los sujetos dos intervalos de observación con una señal de ruido de base, en alguno de los intervalos se le agregaba una señal sinusoidal al ruido. Después de escuchar ambos intervalos, el sujeto tenía que seleccionar aquel en el que creía que estaba presente la señal. El sujeto se veía forzado a dar una respuesta forzada. Cuando la energía de la señal era mucho mayor que la del ruido, las tasas de detección alcanzaban el 100 %. Sin embargo, cuando el coeficiente entre la energía de la señal y la energía del ruido llegaba a cierto valor mínimo, la tasa de detección era del 50 % (sensibilidad). Esto mostraba que el sujeto estaba dando una respuesta aleatoria que no tomaba en cuenta la información sensorial (respuesta sesgada).

Este enfoque de sensibilidad/respuesta sesgada dentro del marco de la TDS fue extendido a diversas áreas como el reconocimiento de memorias, detección de mentiras, selección de personal, diagnósticos médicos, inspecciones industriales, entre otras (Deshmukh y Rajagopalan, 2006).

La TDS puede ser caracterizada como un modelo que ayuda a los tomadores de decisiones o clasificadores a discriminar entre señal y ruido. La teoría asume que existe un sobrelape entre las distribuciones de la señal

y el ruido y que cualquier observación particular puede surgir de ambas distribuciones. En la figura 8 el eje vertical representa la probabilidad, mientras que el eje horizontal representa una variable que es usada por el clasificador para tomar la decisión. Para cierta observación, si el valor de la variable de decisión es alto, entonces el tomador de decisiones responderá con un "señal presente", mientras que si el valor de la variable es bajo el clasificador responderá con "ruido presente". El valor umbral que es lo suficientemente alto para tomar la decisión de "señal presente" es llamado criterio. Dado que las distribuciones de señal y ruido se sobrelapan, la decisión tomada tiene uno de cuatro posibles resultados. Si la observación provino de la distribución de señal y la decisión tomada la identifica como señal, entonces se tiene un acierto o verdadero positivo (TP). En cambio, si esta observación fue identificada como ruido, entonces la decisión es una falla o falso negativo (FN). En caso de que la observación provenga de la distribución de ruido y sea identificada como tal, se tiene una identificación correcta o verdadero negativo (TN). Mientras que si esta observación es identificada como señal, se tiene una falsa alarma o falso positivo (FP). La frecuencia de aciertos es conocida como proporción de aciertos (TPR) y es igual al número de observaciones clasificadas como aciertos, entre el número total de posibles observaciones de señal correctas. Mientras que la proporción de falsas alarmas (FPR) es igual al número de observaciones de ruido que son clasificadas como señal entre el número total de observaciones de ruido (Deshmukh y Rajagopalan, 2006).

2.3.1. Curvas ROC. La modificación del valor umbral del criterio, cambia las características del clasificador. Si se disminuye el valor del criterio aumentará el número de aciertos, pero también aumentara el número de falsas alarmas. Mientras que si el valor del criterio se incrementa, el número de aciertos disminuye al igual que el número de falsas alarmas. La elección del valor umbral es dependiente de la distribución de datos subyacente. Una forma de representar gráficamente el comportamiento del clasificador es conocida como curva de Características Operantes del Receptor (curva ROC). La curva ROC es una descripción bidimensional del comportamiento de un clasificador. Se grafica un cuadro donde ambos ejes van de 0 a 1, siendo el eje x la proporción de falsos positivos y el eje y la proporción de verdaderos positivos (figura 9). Sin embargo, para realizar comparaciones entre diferentes clasificadores es más sencillo tener un valor escalar único. Un método común para obtener un valor escalar es calcular el área bajo la curva ROC (AUC). Dado que el AUC es una porción del área del cuadro unitario, su valor siempre se encontrará entre 0 y 1. Un clasificador que siempre clasifique todo correctamente tendrá un área de 1, un clasificador que siempre clasifique todo incorrectamente tendrá un área de 0, mientras que un clasificador completamente aleatorio producirá una línea diagonal con un área de 0.5 (Fawcett, 2006). Las curvas ROC son una herramienta que ha sido utilizada para comparar el comportamiento de diferentes métodos de detección de eventos electrofisiológicos en una señal (Maccione et al., 2009).

2.3.2. El Coeficiente de Correlación de Matthews (MCC). Una medida estándar usada por los estadistas es el coeficiente de correlación, también conocido como el coeficiente de correlación de Pearson. En el contexto de predicción de estructuras secundarias, también es conocido en la literatura como el Coeficiente de Correlación de Matthews (ecuación 2.3.1). El coeficiente de correlación toma siempre un valor entre -1 y +1. Es una medida de cómo dos variables normalizadas tienden a tener el mismo signo y magnitud. Un valor de -1 indica una completa discrepancia, mientras que un valor de +1 significa una completa concordancia.

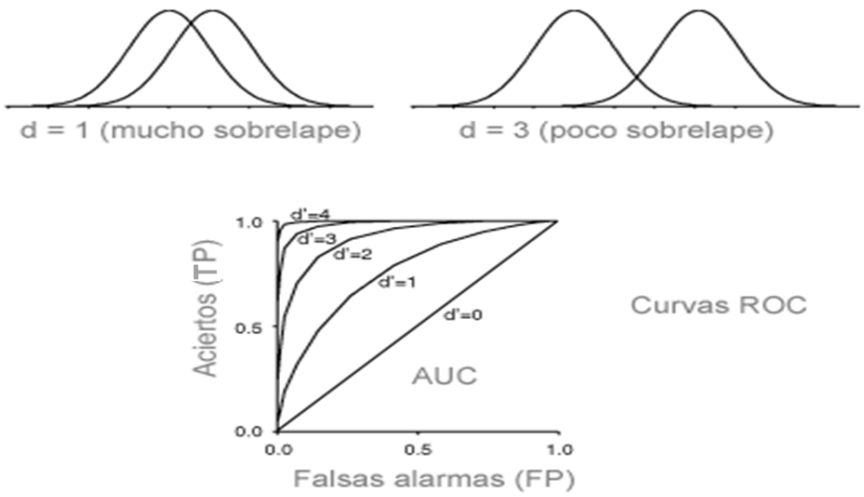


FIGURA 9. Ejemplo de curvas de características operantes del receptor (ROC) para distribuciones con diferentes distancias. Entre mayor sea la distancia entre las distribuciones de la señal y el ruido, el área bajo la curva de la ROC será mayor.

El coeficiente de correlación toma el valor de 0 ante una relación completamente aleatoria (Baldi et al., 2000).

$$MCC = \frac{TP \times TN - FP \times FN}{\sqrt{(TP + FN)(TP + FP)(TN + FP)(TN + FN)}} \tag{2.3.1}$$

El MCC usa los cuatro valores obtenidos de una clasificación: TP, TN, FP y FN. Por lo que frecuentemente, provee una evaluación mejor balanceada que la predicción basadas en las medidas de proporciones (TPR y FPR), como la AUC de las curvas ROC (ver sección 2.3.1). Sin embargo, existen situaciones donde, aún el coeficiente de correlación, no es capaz de proveer una evaluación adecuada. El MCC será relativamente alto en los casos en que el resultado de un clasificador contenga pocos o ningún falso positivo, pero al mismo tiempo contenga pocos verdaderos positivos.

2.4. Transformada Discreta de Ondeleta

La transformada de ondeleta es capaz de proveer simultáneamente el tiempo y la frecuencia de eventos instantáneos particulares en una señal, generando una representación tiempo-frecuencia de la misma. La forma discreta de la transformada continua de ondeleta permite calcular la CWT en una computadora, sin embargo no es realmente una transformada discreta. Lo que genera información altamente redundante para la reconstrucción de la señal. Esta redundancia implica un desperdicio de tiempo de computación y recursos. La transformada discreta de ondeleta (DWT), provee suficiente información para tanto el análisis como la síntesis de la señal, reduciendo los recursos necesarios para su cálculo (Polikar, 2001).

La representación digital de la señal se obtiene usando filtros digitales. Mientras que la transformada continua se calcula cambiando la escala de la ventana de análisis y el desplazamiento de la ventana en

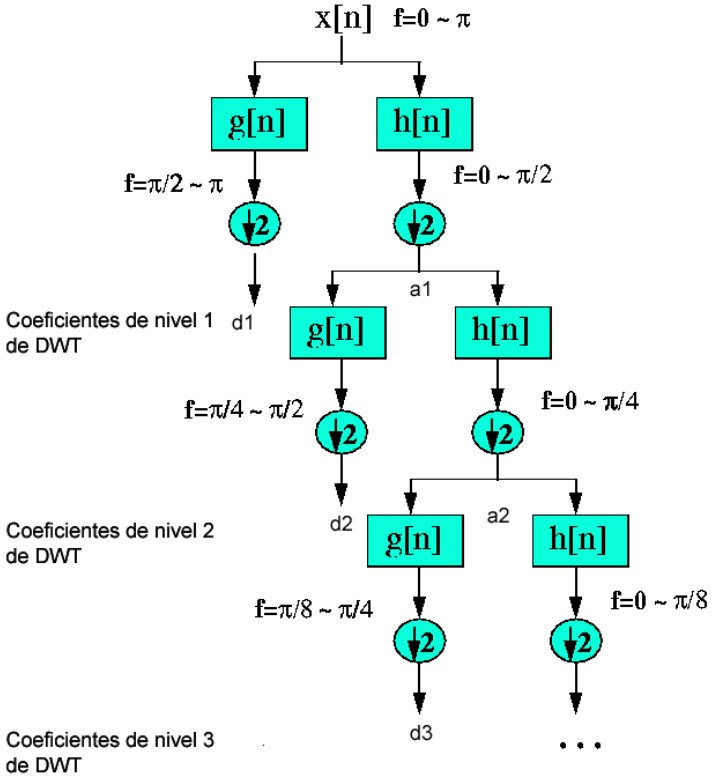


FIGURA 10. Transformada discreta de ondeleta. La señal digital original x[n] se descompone, a través de un filtro de ondeleta pasa-altas *(g[n])* y un filtro de escala pasa-bajas (*h(n)*), en sus componentes de alta (coeficientes de nivel d) y baja frecuencia (coeficientes de aproximación a) respectivamente. Debido a que cada conjunto de coeficientes contiene únicamente la mitad del contenido de frecuencias, se puede bajar la frecuencia de muestreo a la mitad sin perder información. Estos pasos se repiten de manera iterativa para cada nivel de la transformada usando los coeficientes de aproximación del nivel anterior como entrada. Finalmente, se obtiene un conjunto de coeficientes de nivel d para cada nivel de la descomposición y un conjunto de coeficientes de aproximación a para el último nivel.

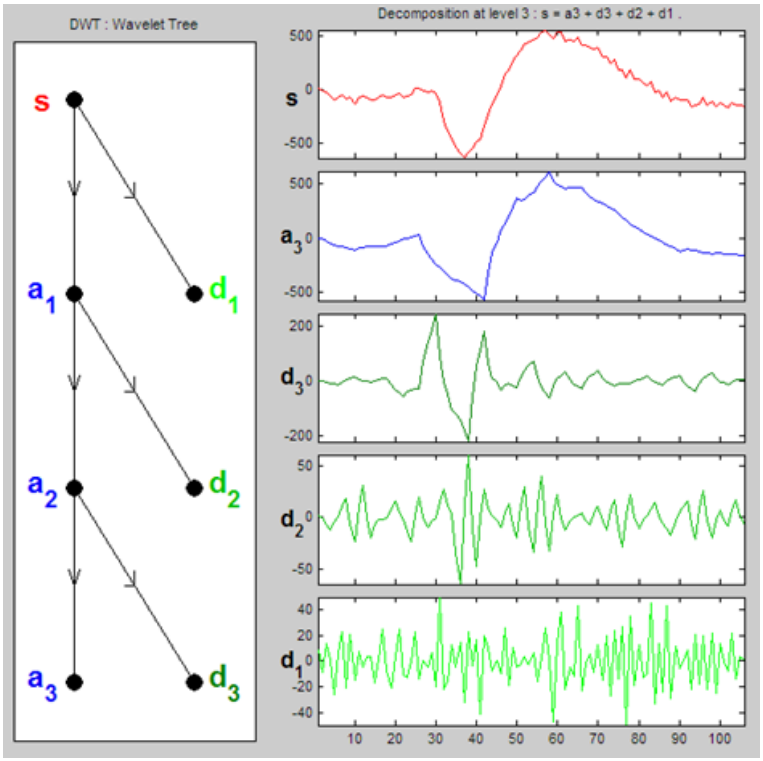


FIGURA 11. Ejemplo de la descomposición de una señal por la transformada discreta de ondeleta. Se muestra la señal original *(s)*, los coeficientes de aproximación del tercer nivel de descomposición (a_3), así como los coeficientes de nivel *(d)* del tercero al primer nivel de descomposición.

el tiempo, multiplicándola por la señal e integrando para todo tiempo (ver sección 2.4). La transformada discreta se obtiene en diferentes escalas al pasar la señal por una serie de filtros con diferentes frecuencias de corte (ver figura 10). La señal se filtra a través de filtros pasa-altas *(g[n])* para analizar las frecuencias altas y se filtra a través de una serie de filtros pasa-bajas *(h[n])* para analizar las frecuencias bajas (Polikar, 2001). Estos filtros están determinados por la ondeleta a utilizar.

La resolución de la señal es alterada por la operación de filtrado, mientras que la escala es modificada por operaciones de sobremuestreo o submuestreo. Submuestrear la señal genera una reducción en la frecuencia de muestreo, al eliminar algunas muestras de la señal. Mientras que sobremuestrear corresponde a un aumento en la frecuencia de muestreo al agregar nuevas muestras a la señal, usualmente por algún método de interpolación (Polikar, 2001).

La transformada de ondeleta tiene la propiedad de poder caracterizar ciertos eventos en las señales, siendo la familia y escala de la ondeleta madre los parámetros que afectan esta propiedad (ver figura 11).

2.4.1. La Transformada Estacionaria Discreta de Ondeleta. La transformada estacionaria discreta de ondeleta (SWT) es una versión de la transformada discreta de ondeleta (ver sección 2.4) donde

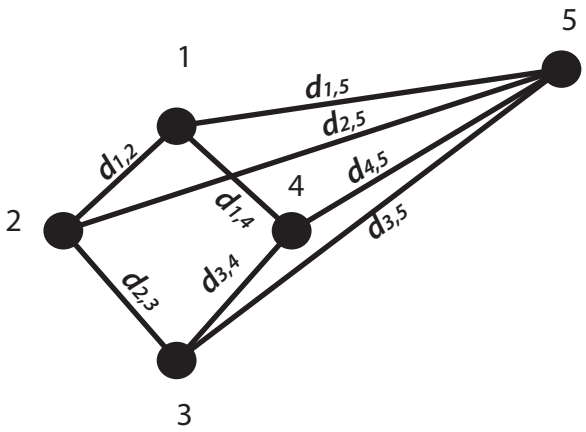


FIGURA 12. La geometría de distancia busca encontrar las coordenadas de un conjunto de puntos, a partir de las distancias entre ellos. En este ejemplo se tienen 5 puntos cuyas coordenadas se obtuvieron a partir de las distancias *d*. (modificado de Cui et al. (2004)).

no se hace la reducción de la frecuencia de muestreo después de pasar la señal por los filtros pasa-altas y pasa-bajas, lo que permite mantener la estructura temporal de los datos originales sin alteraciones.

2.5. Disminución dimensional y extracción de propiedades de una forma de onda de una espiga

La representación de las espigas en los registros electrofisiológicos es un conjunto multivariado de datos. Dependiendo de la frecuencia de muestreo se pueden tener decenas de variables que describen la forma de onda de una espiga. Debido a su complejidad, el costo de procesamiento requerido para analizar estos datos puede llegar a ser muy alto. Las técnicas de reducción dimensional de datos buscan mantener la mayor cantidad de información del conjunto original, que permita diferenciar y agrupar las espigas de acuerdo a la neurona que las generó, disminuyendo la complejidad de los datos para reducir el costo de procesamiento. Existen diversos métodos que permiten reducir el número de dimensiones de un conjunto de datos.

2.5.1. Análisis de Componentes Principales. El Análisis de Componentes Principales (PCA) es un método matemático para reorganizar la información en un conjunto de muestras. El PCA descubre nuevas variables, conocidas como Componentes Principales (PCs), que son responsables de la mayor parte de la variabilidad en los datos. Esto permite describir la información con un número menor de variables de las que estaban presentes originalmente. El primer Componente Principal (PC1) es la dirección a través de los datos que explica la mayor variabilidad en ellos. El segundo Componente Principal (PC2) y los subsecuentes Componentes Principales deben ser ortogonales al PC anterior y deben describir la mayor cantidad de la variabilidad restante. Una vez conocidas las direcciones de los PCs, los valores de las muestras individuales se pueden expresar como una suma lineal de PCs multiplicados por un coeficiente que describe cada PC (Davies y Fearn, 2004). Esto se logra calculando los eigenvectores de los datos y ordenándolos por sus eigenvalores correspondientes.

2.5.2. Geometría de distancia . La geometría de distancia se ha utilizado ampliamente en el análisis de las conformaciones atómicas de una molécula. El problema de encontrar las coordenadas de los átomos que forman una molécula, dado un conjunto de distancias entre pares de átomos, se puede entender como un problema de geometría de distancia (ver figura 12). De manera más general, el problema se puede definir como la búsqueda de las coordenadas para un conjunto de puntos en un espacio euclidiano R_k de k dimensiones para cualquier k, dado un conjunto de distancias entre pares de puntos en R_k. Una forma de resolver este problema es a través de un algoritmo de descomposición de valor unitario (Cui et al., 2004), el cual se basa en la idea de que dadas unas distancias $d_{i,j}$ entre los átomos i y j para todo $i, j = 0, 1, ..., n$, tenemos que:

$$\|x_i - x_j\| = d_{i,j},\ i, j = 0, 1, \ldots, n,$$

o de manera equivalente

$$\|x_i - x_j\|^2 = d_{i,j}^2,\ i, j = 0, 1, \ldots, n.$$

Se encuentre $x_n$localizado en el origen. Entonces

$$\|x_i\|^2 = d_{i,0}^2,\ i = 1, 2, \ldots, n.$$

$$\|x_i\|^2 - 2x_i^T x_j + \|x_j\|^2 = d_{i,j}^2,\ i, j = 1, 2, \ldots, n.$$

Se obtiene

$$d_{i,0}^2 - d_{i,j}^2 + d_{j,0}^2 = 2x_i^T x_j,\ i, j = 1, 2, \ldots, n.$$

Sea $\frac{D_{i,j}=(d_{i,0}^2 - d_{i,j}^2 + d_{j,0}^2)}{2}$. Se puede definir una matriz $D = [D_{i,j}]$. Sea X una matriz de n x *3 y* $X = [x_1, x_2, \ldots, x_n]^T$. Con lo que se obtiene

$$D = XX^T.$$

Si existe una solución para esta ecuación, la matriz D debe de ser de un rango ≤ 3. Por lo tanto, se puede hacer una descomposición de valor unitario para D y obtener así

$$D = U\Sigma U^T,$$

donde U es una matriz ortogonal de n x 3 y Σ es una matriz diagonal de 3 x 3 siendo los elementos diagonales σ_1, σ_2 y σ_3 aquellos valores unitarios mayores de D. Una solución para $D = XX^T$se puede obtener de

$$X = U\Sigma^{\frac{1}{2}}.$$

De manera similar a la utilizada por Cui et al. (2004) en la localización de los átomos que conforman una molécula, se puede aplicar la geometría de distancia a las distancias euclidianas entre las formas de

onda de las espigas, para obtener una representación tridimensional de las formas de onda.

2.6. Clasificación

Los métodos de clasificación y agrupamiento de datos se basan en el supuesto de que los datos provienen de conjuntos de clases independientes, donde cada una de estas clases puede ser descrita por un modelo relativamente simple. Este supuesto coincide con el caso de la discriminación de espigas, donde cada potencial es generado por una neurona diferente. El primer paso de la clasificación es describir tanto la localización del grupo, así como la variabilidad de los datos alrededor de esa localización. El segundo paso es, dada la descripción de un grupo, clasificar los datos nuevos. Existen muchos métodos para realizar el agrupamiento (Lewickiy, 1998).

2.6.1. K-Medias. El algoritmo de agrupamiento del vecino más cercano o de k-medias, consiste en definir la localización de un grupo como la media de los datos que conforman al grupo. Se inicializa el algoritmo con una estimación aleatoria de los grupos. Una espiga es asignada al grupo cuya media sea más cercana, usando distancia Euclidiana. Después de terminar la reasignación de espigas, se vuelve a iniciar otra iteración del algoritmo, calculando las medias de los grupos nuevamente, para después volver a calcular las distancias de cada espiga a las medias de los grupos, reasignándolas en caso de ser necesario. El proceso termina cuando la distancia de todas las espigas a la media de su grupo asignado es menor a cierto umbral o ya no se modifica el grupo de pertenencia de ninguna espiga. Este proceso define un conjunto de fronteras de decisión implícitas que separan los grupos usando sólo información sobre las medias e ignorando la distribución de los datos en el grupo (Lewickiy, 1998).

2.6.2. Expectación - Maximización para Mezclas Gaussianas. El algoritmo de Expectación-Maximización (EM) es un método que permite encontrar el estimador de máxima verosimilitud (ML) de un parámetro de una distribución probabilística. Por ejemplo, si suponemos que la temperatura ambiental para cada una de las 24 horas del día $x \epsilon \mathbb{R}^{24}$ depende de la estación de año $\theta \epsilon$ {primavera, verano, otoño, invierno} y que la distribución de la temperatura en cada estación es conocida $p(x|\theta)$. Pero, supongamos que sólo podemos medir la temperatura promedio de un día dado $y = \overline{x}$ y que queremos adivinar a partir de este dato a que estación del año θ pertenece este día. El estimador de máxima verosimilitud (ML) de θ, maximiza la probabilidad $p(y|\theta)$. El algoritmo de EM toma los datos observados y, de manera iterativa hace conjeturas acerca de los datos completos x y de igual manera encuentra la θ que maximiza la $p(x|\theta)$ sobre θ. De esta manera, el algoritmo EM intenta encontrar el ML de θ dada y (Chen y Gupta, 2010).

Cada iteración del algoritmo de EM consiste en dos procesos: el paso E y el paso M. En la expectación, o paso E (ecuación 2.6.1), los datos faltantes son estimados dados los datos observados y la estimación actual de los parámetros del modelo.

$$Q(\theta \mid \theta^{(t)}) = E_{Z|x,\theta^{(t)}} \left[\log L \left(\theta; x, Z \right) \right]. \tag{2.6.1}$$

En el paso M (ecuaciones 2.6.2 y 2.6.3), la función de probabilidad es maximizada bajo la suposición de que los datos faltantes son conocidos. Los estimados de los datos faltantes del paso E son usados en lugar de los verdaderos datos faltantes (McLachlan y Krishnan, 1996).

$$Q^{(t+1)} = \arg_\theta \max Q(\theta|\theta^{(t)}) \tag{2.6.2}$$

$$Q^t(\Theta) \triangleq \langle \log P(U, J|\Theta) \rangle \tag{2.6.3}$$

Bajo la suposición de que nuestros datos siguen una distribución normal, la ecuación L a maximizar está definida en la ecuación 2.6.4.

$$L(\theta; x, z) = \prod_{i=1}^{n} \sum_{j=1}^{k} I(z_i = j)\tau_j f(x_i; \mu_j, \Sigma_j). \tag{2.6.4}$$

El algoritmo de EM termina cuando no existe un punto que pueda maximizar la ecuación L en comparación con el estado actual.

2.6.3. Estimación del número de grupos. Los algoritmos de EM y de k-medias requieren que se les proporcione como parámetro el número de grupos buscados. Los resultados de ambos algoritmos cambian significativamente de acuerdo a este parámetro. Es por esto, que es de suma importancia tener un estimado del número de grupos presentes lo más preciso posible.

2.6.3.1. Criterio Bayesiano de Información (BIC). Uno de los aspectos más complicados del agrupamiento, es elegir el número de clases o grupos existentes. En el enfoque bayesiano, es posible estimar la probabilidad de cada modelo dados los datos observados. Si los supuestos del modelo son acertados, estos maximizarán las probabilidades relativas de diferentes números de clases. Este procedimiento selecciona el número de clases más probable dado las datos y no siempre favorece modelos con un mayor número de clases. BIC es un criterio de probabilidad penalizado por la complejidad del modelo (el número de parámetros en el modelo).

Sea $\chi = \{x_i \epsilon R^d : i = 1, ..., N\}$ el conjunto de datos que se quieren agrupar. Sea $C_k = \{c_i : i = 1, ..., k\}$ el agrupamiento que tiene k grupos. Modelamos cada grupo c_i como una distribución multivariada gaussiana $N(\mu_i, \Sigma_i)$ donde μ_i puede ser estimado como el vector promedio de la muestra y Σ_i puede ser estimada como la matriz de covarianza de la muestra. Entonces el número de parámetros para cada grupo es $d + \frac{1}{2}d(d+1)$. Sea n_i el número de muestras en el cluster c_i. Se define el BIC como

$$BIC(C_k) = \sum_{i=1}^{k} \{-\frac{1}{2} n_i log\,|\Sigma_i|\} - Nk(d + \frac{1}{2}d(d+1)). \tag{2.6.5}$$

Se calcula el BIC para cada modelo generado por el algoritmo de agrupamiento para un rango de un número de grupos. Este rango debe ser proporcionado *a priori*. Finalmente se elige el modelo con el número

FIGURA 13. Operación de entrecruzamiento de un algoritmo genético.

FIGURA 14. Operación de mutación de un algoritmo genético.

de grupos que maximiza el criterio del BIC (Chen y Gopalakrishnan, 1998).

2.7. Algoritmo genético

Un algoritmo genético es una técnica usada en computación para encontrar una solución exacta o aproximada a un problema de búsqueda y optimización. Es necesario que el dominio de soluciones tenga una representación genética y que exista una función de aptitud, con la cual se pueda evaluar el dominio de la solución.

Existen un conjunto de operaciones que debe llevar a cabo un algoritmo genético:

- Inicialización, debe de generar un conjunto de soluciones iniciales. Es común que esta inicialización se lleve a cabo de manera aleatoria.
- Selección, en cada iteración del algoritmo genético se seleccionan un porcentaje de las soluciones que obtengan los mejores valores de la función de aptitud.
- Reproducción, después de realizar la selección de las posibles soluciones, se generan soluciones viables de manera aleatoria y se combinan con las soluciones seleccionadas en el paso anterior. La combinación se hace por medio de:
  - Entrecruzamiento, se intercambian el inicio de una representación genética de una solución con el final de otra solución en un punto aleatorio (ver figura 13).
  - Mutación, la mutación se lleva a cabo con una probabilidad muy baja y consiste en cambiar aleatoriamente un valor binario de la representación de una solución (ver figura 14).
- Terminación, el algoritmo se repite hasta obtener una solución que tenga un nivel de error preestablecido o después de que se lleven a cabo cierto número de iteraciones.

2.8. Recursos informáticos para el análisis y simulación de registros electrofisiológicos

Existen diversas herramientas informáticas para los diferentes aspectos del análisis de datos electrofisiológicos. En la primer parte de esta sección hay dos ejemplos de programas que sirven para la clasificación automática de espigas: KlustaKwik y OSort. Estos programas se utilizaron para realizar comparaciones con los algoritmos presentados en este trabajo. En la siguiente parte de esta sección se presenta el programa sigTOOL, que es una plataforma de análisis de señales biológicas. Aprovechando su diseño modular, se creó

una herramienta para la creación y comparación de algoritmos de discriminación de espigas nombrada Sort Lab (ver sección 7.7). Uno de los módulos de esta herramienta permite crear señales artificiales para ser usadas en pruebas de comparación de algoritmos de discriminación. Estas señales artificiales son creadas a partir de los archivos generados por PyNN, el cuál es un programa que unifica la interfaz entre la creación de modelos de redes neuronales y diferentes simuladores de estas redes. NEST es uno de los simuladores soportados por PyNN, este simulador se utilizó para generar los archivos bases usados en la creación de las señales artificiales, debido a su enfoque en la topología de las redes neuronales.

2.8.1. KlustaKwik. Es un programa escrito en C++ para la clasificación de datos continuos multi-dimensionales. Originalmente se creó para la clasificación automática de formas de ondas de potenciales de acción neuronales. KlustaKwik está basado en el algoritmo de clasificación EM de Celeux y Govaert (1992). Fue creado con los siguientes objetivos:

- Ajustar una mezcla de gaussianas con matrices de covarianza libres.
- Automáticamente elegir el número de componentes que forman la mezcla.
- Ser robusto contra el ruido.
- Reducir el problema de los mínimos locales.
- Ejecución rápida en conjuntos grandes de datos.

KlustaKwik permite que se ajusten un número variable de grupos, penalizados por el criterio de información de Akaike (AIC). El programa prueba si al partir o juntar grupos se obtiene una mejor solución, lo que también ayuda al algoritmo a escapar de mínimos locales y a disminuir su sensibilidad en el número inicial de grupos (Harris, 2000).

2.8.2. OSort. OSort es un programa para Matlab, desarrollado para la detección y clasificación de potenciales de acción en línea. Las espigas que se originan de diferentes neuronas son diferenciadas por el contorno de su forma de onda y las diferencias en sus amplitudes, estas características son únicas a las neuronas individuales. El algoritmo actualiza de forma iterativa el modelo y asigna las espigas a grupos. Por lo tanto, no requiere de una fase de aprendizaje y es capaz de detectar nuevas neuronas que aparezcan en el registro durante el experimento (Rutishauser et al., 2006).

2.8.3. *sigTOOL*. *sigTOOL* es un programa de código libre para el análisis de señales biológicas. Es desarrollado por Malcolm Lidierth en el Wolfson Centre for Age-Related Diseases del Guy's Hospital Campus en el King's College London. Este programa corre bajo Matlab y provee un ambiente de programación y análisis para el procesamiento de datos neurológicos. Por medio de una interfaz gráfica, este ambiente, permite el análisis de formas de onda y de trenes de espigas (Lidierth, 2009).

El programa permite la incorporación de extensiones creadas por el usuario, que son integradas sin necesidad de modificar el código existente. *sigTOOL* se distribuye bajo una licencia de código libre GNU General Public License. Entre las funciones para el trabajo con formas de onda incluye: medias y medianas, auto correlación, cros-correlación, análisis espectral, estimación de coherencia, filtros digitales (IIR y FIR), re-muestreo, distribución de amplitud e ICA. Mientras que para el análisis de trenes de espigas tiene funciones

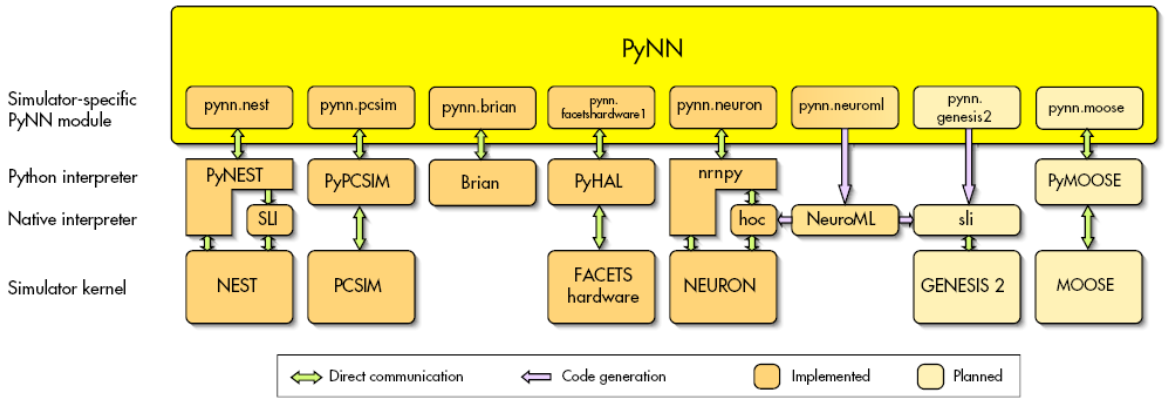


FIGURA 15. Arquitectura de PyNN (tomado de Davison et al. (2009)). En el nivel superior PyNN interpreta una definición general de una red neuronal. El nivel intermedio, convierte esta definición general en las instrucciones específicas de cada simulador. En el nivel más bajo, los simuladores llevan a cabo la simulación y regresan los resultados a PyNN, para ser traducidos a un formato común.

para: distribuciones de intervalos inter-espiga, gráficas de Poincaré, histogramas temporales peri-evento dirigidos por estímulos, *rasters*, etc.

2.8.4. PyNN . Existen diversos simuladores de redes neuronales. Aún cuando todos permiten simular redes, su enfoque suele ser diferente. Existen simuladores que están orientados a las simulaciones a gran escala, mientras que otros se especializan en simular las características de neuronas individuales. PyNN es un programa para Python (van Rossum y de Boer, 1991) que funciona como una interfaz de programación común para múltiples simuladores de redes neuronales. PyNN permite escribir un programa de una red neuronal una sola vez, utilizando el lenguaje Python, y ejecutarlo sin modificaciones en cualquier simulador soportado (entre los que están NEURON, NEST, PCSIM, Brian y el hardware neuromórfico VLSI Heidelberg). La finalidad de PyNN es incrementar la productividad del modelado de redes neuronales al proveer un nivel de abstracción de alto nivel, promover el intercambio y reutilización de código fuente y proveer las bases para realizar análisis independientes de las herramientas de simulación, visualización y manejo de datos.

2.8.5. Neural Simulation Tool NEST . NEST es un programa para la simulación de grandes redes heterogéneas de neuronas puntuales o con un número pequeño de compartimentos. Este simulador está orientado a los modelos que se enfocan en la dinámica, tamaño y estructura de los sistemas neuronales en lugar de en el detalle morfológico o biofísico de las propiedades de neuronas individuales.

Una simulación con NEST permite construir un sistema neuronal creando uno o más modelos de neuronas y conectándolos a través de sinapsis para formar una red. Es posible mezclar diferentes tipos de modelos de neuronas en una misma red. De igual manera, se pueden combinar diferentes modelos de mecánicas sinápticas. El simulador permite crear redes estructuradas en capas, áreas o subredes. Se pueden obtener mediciones de la red a través de dispositivos virtuales como voltímetros en la membrana celular de las

neuronas y detectores de espigas. NEST está diseñado para correr de manera paralela en computadoras individuales con múltiples procesadores o en grupos de computadoras conectadas juntas (Gewaltig y Diesmann, 2007).

NEST tiene un lenguaje propio llamado SLI que permite crear y manipular las redes y sus parámetros, así como crear funciones y realizar cálculos matemáticos. Sin embargo, también existe una extensión llamada PyNEST, que permite manipular el simulador desde el lenguaje Python (van Rossum y de Boer, 1991). Este lenguaje ha cobrado especial importancia en los últimos años en el análisis de datos científicos, gracias a la gran cantidad de librerías matemáticas de software libre que hay disponibles.

Capítulo 3

Justificación

El entendimiento de los códigos neuronales, requiere el registro simultaneo de un gran número de neuronas (Pouzat et al., 2002). Los registros extracelulares siguen siendo el único método práctico para preparaciones in vivo que es capaz de proveer información de la actividad eléctrica de unidades individuales dentro de poblaciones neuronales. Sin embargo, la información de los registros extracelulares de múltiples neuronas sólo es útil si las espigas generadas por las diferentes neuronas pueden ser discriminadas y clasificadas correctamente (Pouzat et al., 2002). La clasificación de las espigas se puede hacer de manera manual por un experto. Sin embargo, la clasificación manual tiene un índice de error mayor que los algoritmos automáticos (Harris et al., 2000). El tiempo dedicado al procesamiento de los registros toma especial importancia ante los avances técnicos que permiten el registro de decenas o hasta cientos de neuronas al mismo tiempo. Es por esto, que existe un gran interés en encontrar un algoritmo que sea capaz de realizar la discriminación de manera automática y eficaz. Diversos algoritmos de clasificación de espigas se han creado en los últimos años, tales como *Osort (Rutishauser et al., 2006), SOM* (Chelaru y Jog, 2005) y *DGCC* (Vargas-Irwin y Donoghue, 2007). Sin embargo no se ha logrado obtener consenso sobre cuál es el mejor trabajando de manera automática con la variedad de registros de diferentes laboratorios. Otro problema para la evaluación de los diversos algoritmos existentes, es que no existe una plataforma de pruebas estandarizada, por lo que es difícil realizar una comparación objetiva de los algoritmos. Un algoritmo que funcione de manera automática, eficiente y que sea adaptable a las diferentes condiciones de registro de diferentes laboratorios sería una herramienta sumamente valorada por la comunidad científica que utiliza registros electrofisiológicos de actividad unitaria.

Capítulo 4

Hipótesis

Un algoritmo de clasificación de espigas por épocas de tiempo tendrá un mejor desempeño en la detección y asignación, en comparación con los algoritmos tradicionales.

Capítulo 5

Objetivos

Diseñar un algoritmo que funcione de manera automática y que sea capaz de:

1. Identificar los segmentos de señal que contengan espigas.
2. Identificar el número de neuronas que aparecen en un registro.
3. Realizar la asignación de las espigas a la neurona que las haya generado.
4. Minimizar los errores en la clasificación.
5. Generar un estimado cuantitativo de la calidad de la señal y de los algoritmos de clasificación.

Capítulo 6

Métodos

Este proyecto se realizó principalmente en Matlab 7 en sus versiones para Windows y para Linux. La base de datos relacional usada para el almacenamiento de los datos de diversos análisis (apéndice A) y para el control del sistema de computo distribuido (ver apéndice B) fue MySQL. Se utilizó Perl 5 para la conversión de archivos de datos entre diferentes formatos. Finalmente, la simulación de la red neuronal para la generación de la señal artificial se llevó a cabo utilizando NEST (ver sección 2.8.5) a través de PyNN (ver sección 2.8.4).

El proyecto se llevó a cabo siguiendo las siguientes etapas:

1. Acondicionamiento y caracterización del contenido de frecuencias de señales de registros extracelulares.
2. Diseño de un algoritmo de detección y clasificación, por medio de comparaciones de diferentes métodos para cada etapa.
3. Selección de medidas de calidad de los algoritmos usados, así como de la señal de entrada.
4. Optimización de los algoritmos utilizados, usando un algoritmo genético para encontrar los parámetros que maximicen la detección y clasificación de espigas.
5. Generación de una señal artificial que refleje los problemas que se encuentran en los registros reales: artefactos por ruido de modo común, artefactos por movimiento del sujeto, ruido, múltiples neuronas presentes en el registro, traslape en el disparo de múltiples neuronas, variabilidad de la forma del potencial de acción con el tiempo.
6. Comparación de los resultados obtenidos por los algoritmos propuestos, así como con KlustaKwik (Harris, 2000) y OSort (Rutishauser et al., 2006) en señales artificiales y reales.

6.1. Acondicionamiento y caracterización del contenido de frecuencias de señales de registros extracelulares

6.1.1. Acondicionamiento de datos. La conversión de los archivos planos de datos electrofisiológicos a la estructura de una base de datos relacional (ver apéndice A), consistió en leer la información de 12 archivos DDT, que es el formato binario en el que se almacena los datos electrofisiológicos de los sets de registro. Cada archivo DDT representa una corrida que es parte de un ensayo. El archivo es de longitud variable y consiste en la información paralela de 7 canales de registro, donde cada canal corresponde a un electrodo. La información de cada canal se dividió en segmentos de 10 segundos de duración. En la base de datos se relacionó el nombre del archivo original, número de canal y bloque temporal para la identificación específica de cada bloque de datos almacenado.

Calidad	Características
0	Sólo ruido
1	Baja señal/alto ruido
2	Señal media/alto ruido
3	Señal media/ruido medio
4	Señal media/bajo ruido
5	Alta señal/bajo ruido

TABLA 1. Clasificación de la calidad de señales en registros extracelulares.

Cada canal de datos se clasificó de acuerdo a su calidad. Esta clasificación se hizo en base a las anotaciones hechas por expertos al momento de realizar los registros. La escala de clasificación usada se muestra en la tabla 1.

Con el fin de diferenciar las secciones de cada señal que corresponden a espigas o ruido, se utilizó el programa OSort (Rutishauser et al., 2006). En esta fase se utilizó este programa para poder tener una referencia de clasificación sin prejuicio de su rendimiento. La salida del programa OSort es un vector que contiene los tiempos en que ocurre el pico máximo de una espiga. Utilizando el vector de tiempos se extrajeron 0.8 ms de señal anteriores al tiempo del pico máximo y 2.2 ms posteriores al mismo. Se guardaron las espigas en la base de datos, de igual manera se guardaron los segmentos de señal que no contenían espigas y se marcaron como ruido.

6.1.2. Caracterización del espectro de frecuencias. Se utilizó la transformada rápida de Fourier (*FFT,* ver sección 2.2) para conocer el contenido de frecuencias en los registros extracelulares. Para ver la variación del espectro de frecuencias en el tiempo, se dividió una señal, que contenía tanto ruido como espigas, en segmentos de 10 segundos a los cuales se les aplicó la *FFT.*

Se analizó el espectro de frecuencias para señales electrofisiológicas que contenían tanto ruido como espigas y segmentos de señal que contenían espigas o ruido de manera excluyente. Se obtuvieron los valores promedio y de desviación estándar de los espectros de frecuencia para cada grupo de señales de acuerdo a su nivel de calidad.

Se realizó un análisis estadístico de las frecuencias donde aparecían picos en los espectros de las señales de diferentes calidades (ver tabla 1), con el fin de detectar aquellas frecuencias que aparecieran exclusivamente en las señales que contengan espigas en contraposición de las que contengan sólo ruido. Para cada uno de los espectros de frecuencia de las señales de los diferentes niveles de calidad se buscaron las frecuencias en las que aparecían picos. Donde se definió un pico como aquel valor máximo cuyas frecuencias inmediatas aledañas tuvieran un valor menor que una desviación estándar que el valor actual. Se contaron el número de señales del mismo nivel de calidad que tuvieran picos en las mismas frecuencias.

6.2. Diseño del algoritmo

El algoritmo se implementó con una estructura modular que permite probar diferentes combinaciones de métodos para las distintas etapas (ver figura 16). Los módulos que conforman la implementación son los siguientes:

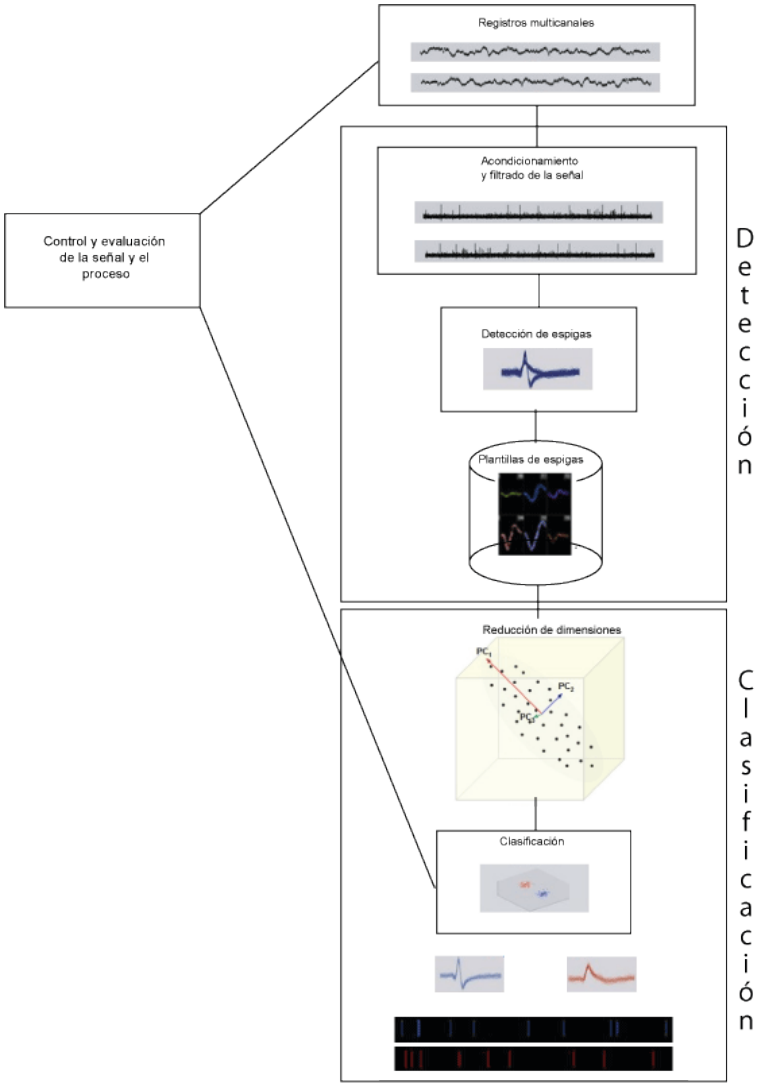


FIGURA 16. Esquema general del algoritmo de clasificación de espigas (tomado y modificado de Quian Quiroga (2007)).

1. Filtrado de la señal. El objetivo de esta etapa es eliminar el ruido la señal por medio de filtros. butterworth (ver sección 2.1.1) y por ondeletas (ver sección 2.1.2) (Hulata et al., 2002; Chan et al., 2008a), comparando la deformación de las formas de onda en el tiempo y la composición de frecuencias de la señal filtrada contra la original.
2. Detección de espigas. Se implementó el método de detección por ondeletas (ver sección 2.1.2).
3. Filtrado de las formas de onda detectadas por medio de un umbral de la distancia euclidiana entre la forma de onda detectada y una base de datos de plantillas de formas de espigas.
4. Reducción de dimensiones. Se implementó usando el análisis de componentes principales (ver sección

2.5.1). También se utilizaron medidas representativas de las formas de onda tales como la amplitud pico-valle, la proporción pico-valle y la distancia a la plantilla más cercana.

5. Clasificación de las espigas. Se utilizaron los algoritmos de clasificación de *Expectación-Maximización* (ver sección 2.6.2) *y Expectación-Maximización fraccionado en el tiempo* (ver sección 6.2.5.2).
6. Evaluación de la calidad de la señal y del comportamiento del algoritmo con respecto a un vector objetivo (ver sección 7.3).

6.2.1. Filtrado de la señal. La señales electrofisiológicas tienen componentes de diferentes frecuencias que modifican las formas de onda de las espigas. En especial se pueden ver componentes de alta frecuencia que aparecen como ruido aleatorio en las señales y componentes de baja frecuencia que puede generar un efecto de onda envolvente. Al filtrar una señal se busca eliminar el ruido no deseado. Sin embargo, es común que los filtros provoquen un desplazamiento de la fase de la señal, lo que se traduce en una deformación de la forma de onda de la señal en el tiempo. Se realizaron comparaciones de filtros Butterworth y filtros basados en ondeletas, para buscar los parámetros adecuados que eliminan el ruido, sin deformar la forma de onda de las espigas.

6.2.1.1. Butterworth. Para ver el efecto de un filtro Butterworth de tercer orden (ver sección 2.1.1) en la forma de onda de una espiga, se hicieron mediciones de la amplitud y la longitud de la espiga con diferentes frecuencias de corte. Este análisis se realizó para las siguientes frecuencias de corte: 100Hz, 300Hz, 500Hz, 600Hz, 700Hz, 800Hz, 900Hz, 1000Hz, 2000Hz, 3000Hz, 4000Hz y 5000 Hz.

Se definieron tres puntos en cada espiga analizada, se eligieron estos tres puntos pues se pueden definir de manera objetiva y son capaces de caracterizar la silueta de una espiga:

- t0, el tiempo en que se encuentre el valor mínimo de la señal antes del máximo de la misma.
- t1, el tiempo en el cual la señal toma un valor menor que el que tuviera en t0 después del máximo de la espiga.
- t2, el tiempo en que la señal regresa nuevamente a un valor mayor al de t0 después del mínimo de la espiga.

A partir de estos tres tiempos se definió el tiempo T.A. como la diferencia entre t1 - t0. Mientras que el tiempo T.B. se definió como la diferencia entre t2 - t1. Se analizaron 31 espigas aleatorias, para calcular los cambios promedios del tiempo T.A. y T.B. para cada frecuencia de corte, con respecto a la señal original.

6.2.1.2. Ondeletas. Una desventaja de la CWT (ver sección 2.1.2) es que la representación de la señal usualmente es redundante, dado que a y b son continuos sobre $\mathbb{R}$ (los reales). Sin embargo, es gracias a esta característica que la CWT se puede usar como un filtro para eliminar el ruido en una señal (Tsai, 2002). Los pasos para la eliminación del ruido son los siguientes:

1. Se aplica la transformada continua de ondeleta a la señal con ruido para generar los coeficientes del nivel mínimo en que se describa satisfactoriamente la señal.
2. Se selecciona un umbral en cada nivel y se igualan a cero los coeficientes menores a este.
3. Se realiza la transformada inversa de ondeleta con los coeficientes obtenidos.

6.2.2. Detección de espigas.

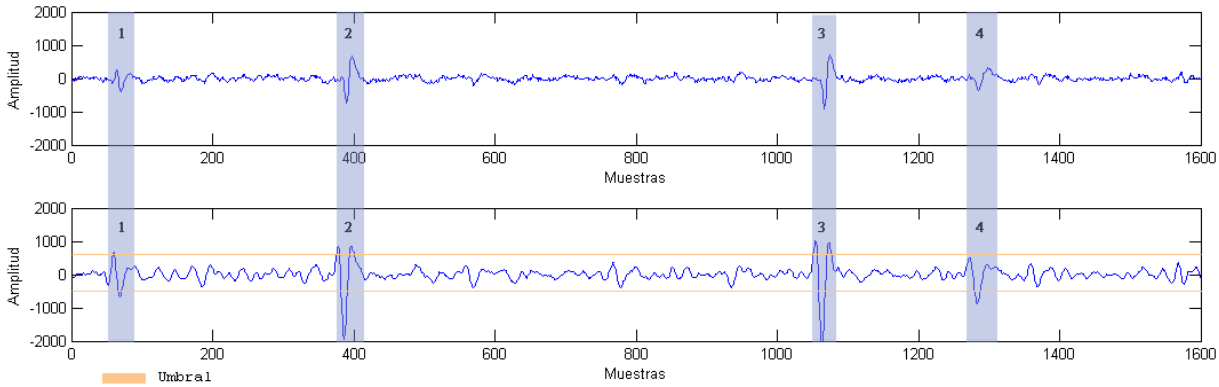


FIGURA 17. Detección de espigas en una señal por medio de la transformada estacionaria discreta de ondeleta. En el canal superior se presenta un segmento de 40ms de la señal original. En el canal inferior se muestran los coeficientes de nivel 4 de la transformada estacionaria discreta de ondeleta usando una ondeleta rbio1.1. Las áreas sombreadas muestran los segmentos de la señal original con forma de espigas y su transformación en formas de onda de una mayor amplitud, aunque con forma diferente y con polaridad invertida, en la señal procesada por la transformada de ondeleta.

6.2.2.1. Transformada Discreta de Ondeleta . La familia de la ondeleta madre y la escala son los parámetros de la transformada estacionaria discreta de ondeleta (ver sección 2.4.1) que determinan la forma de los eventos que será capaz de detectar la *SWT*, así como el valor umbral afectará la sensibilidad de detección (ver figura 17).

Se clasificó en cuatro eventos las diferentes partes de una señal de registro extracelular, de acuerdo a su contenido:

1. Ruido ambiente, es el ruido que aparece durante todo el registro y se encuentra en toda la banda de frecuencias, con una distribución de frecuencias combinada entre un ruido blanco de baja amplitud y un ruido rosa que domina las bajas frecuencias. El ruido blanco es una señal cuyo espectro es plano en cualquier rango de frecuencias, en otras palabras su densidad espectral de potencia es una constante. Mientras tanto, el ruido rosa se caracteriza por tener una gran magnitud en las frecuencias bajas, que va decayendo de manera exponencial a mayores frecuencias.
2. Espigas, señales generadas por los disparos de las neuronas con forma ondulatoria. La forma de onda de una espiga está compuesta por ondulaciones en frecuencias en el rango de los 500Hz, sin embargo la frecuencia de disparo entre espigas es un orden de magnitud más baja.
3. Artefactos por movimientos, son de baja frecuencia y tienen una escala de duración en segundos.
4. Ruido común, son artefactos que aparecen de manera simultánea en todos o casi todos los canales.

Se utilizó una estrategia iterativa con el objetivo de buscar los parámetros que permitan detectar de manera confiable los cuatro tipos de eventos mencionados. Recorriendo cada familia de ondeletas soportada por Matlab y un rango de escalas de 1 hasta 15, se realizaron *SWT* de señales donde se había detectado anteriormente los tiempos en que ocurrían cada tipo de evento. Se estableció un parámetro umbral, que

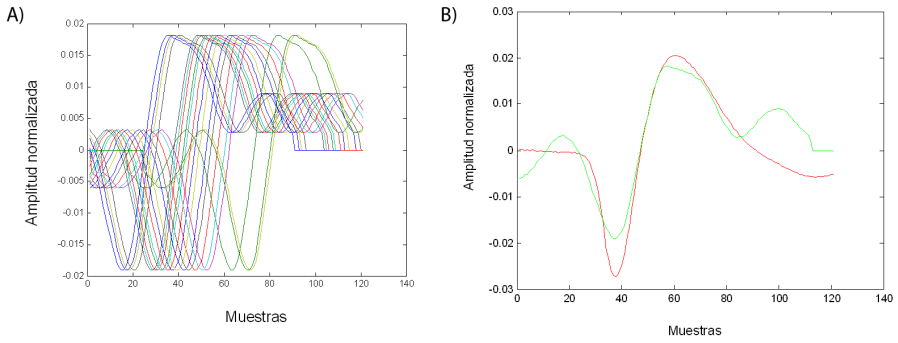


FIGURA 18. Alineación y filtrado de una forma de onda usando la distancia euclidiana a una plantilla. A) Alineación de una forma de onda desplazándola con respecto de una plantilla. B) Filtrado de la forma de onda al comparar la distancia euclidiana entre la forma de onda (verde) y una plantilla (rojo).

indica el valor mínimo que debe tomar la SWT cuando se detecta un evento. Se tomó el valor máximo del SWT y se dividió en veinte partes, cada valor se utilizó como umbral para generar un vector de tiempos donde presuntamente se detectó el evento. Se comparó el vector de tiempos generado con los tiempos reales en que se habían registrado los eventos. A partir de esta comparación se estableció la proporción de verdaderos positivos (TPR) y falsos positivos (FPR). Recorriendo los veinte posibles valores para el umbral se generó su correspondiente curva de características operantes del receptor. Sin embargo, utilizar una proporción del valor máximo de la señal como valor umbral resultó en valores que no eran transferibles entre señales diferentes. Por lo que, posteriormente se cambió el valor base para calcular el umbral. Utilizando la desviación estándar de la señal, en lugar del valor máximo de la misma.

Debido a la gran cantidad de cálculos que son necesarios para llevar a cabo los análisis en todas las familias de ondeletas y 15 escalas para cada una de ellas, se creó un sistema de computo distribuido basado en Matlab y una base de datos relacional descrito en el apéndice B.

6.2.3. Filtrado de formas de onda.

6.2.3.1. Filtrado por plantillas. La detección de espigas genera un número considerable de falsos positivos, por lo que es necesario realizar un tamizaje posterior de las formas de onda. Uno de los métodos que ofrece mejores resultados para filtrar estas formas de onda es la utilización de plantillas (Lewickiy, 1998). El primer paso es normalizar el área bajo la curva de la espiga putativa, con el fin de reducir la variabilidad en la magnitud de las formas de onda al ser comparadas con las plantillas. Se calcula la alineación de la forma de onda que minimice la distancia euclidiana entre la forma de onda normalizada y cada una de las plantillas registradas en el catálogo generado previamente. Se genera una matriz de distancias, comparando las formas de onda desplazadas con cada plantilla del catálogo. Utilizando esta matriz, se elige la plantilla y el desplazamiento con la distancia más corta (ver figura 18). Cuando la distancia entre la espiga putativa y la plantilla es menor que cierto umbral predefinido se acepta la forma de onda como una espiga, de no ser así se rechaza (ver tabla 2).

Desplazamiento	0	-1	+1	-2	+2	-3	+3
Distancia	0.059	0.123	0.161	0.115	0.124	0.133	0.192

TABLA 2. Distancia euclidiana entre una forma de onda y una plantilla para diferentes desplazamientos. Se alinea la forma de onda al desplazamiento que presente la menor distancia.

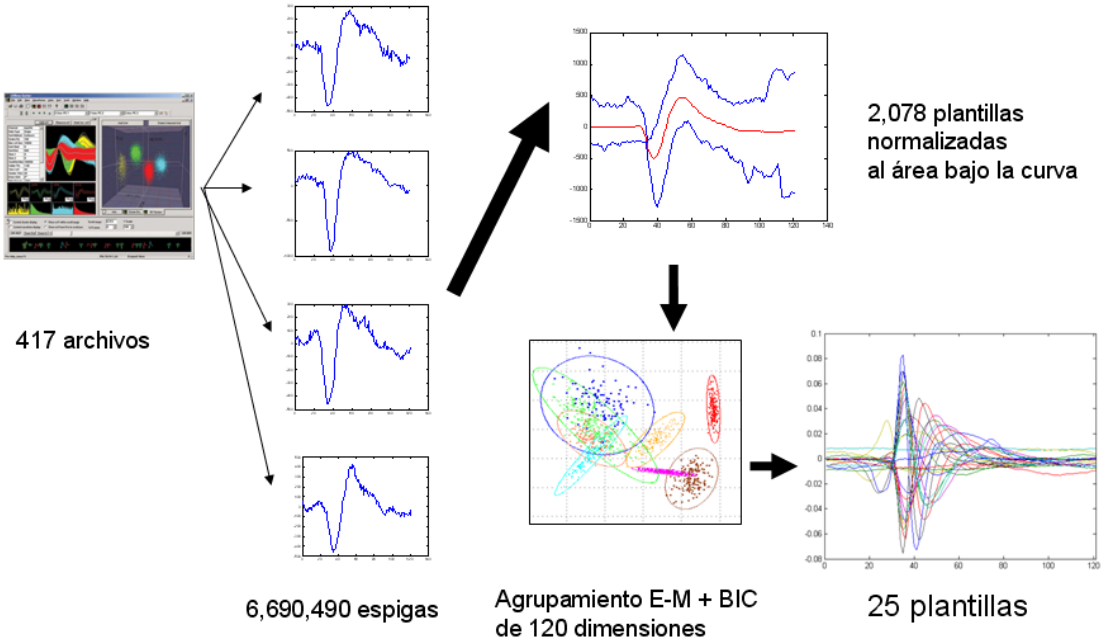


FIGURA 19. Esquema del proceso para la creación de la base de datos de plantillas. Se extrajeron las espigas de 417 archivos de registros extracelulares que fueron discriminados por expertos humanos. Se calcularon las medias de cada unidad discriminada. Las formas de ondas se agruparon utilizando el algoritmo de Expectación-Maximización con el criterio bayesiano de información. Finalmente, se obtuvieron 25 plantillas que correspondían a los centroides de las espigas agrupadas.

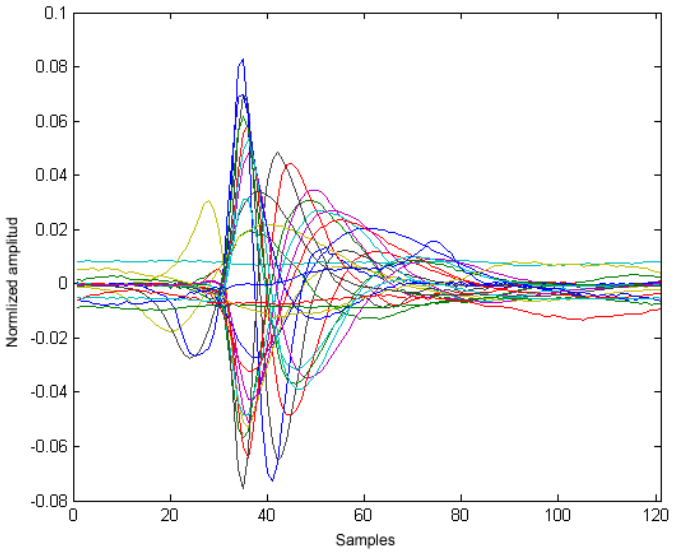


FIGURA 20. Formas de onda normalizadas del conjunto original de 25 plantillas tipo E.

6.2.3.2. Generación del catálogo de plantillas. Se generaron dos tipos de plantillas: M y E. El catálogo de plantillas tipo M se generó basado en los archivos disponibles de registros. La discriminación de las espigas presentes en los registros se llevó a cabo manualmente por expertos, usando el programa Plexon (Plexon, 2006). Se extrajo por separado cada unidad neuronal identificada y sus correspondientes formas de onda. Se calculó una plantilla como la media de las formas de onda pertenecientes a cada unidad. De esta manera se generaron 2078 plantillas. Se guardó en una base de datos la forma de la espiga de cada plantilla.

Se obtuvo otro catálogo de plantillas tipo E, aplicando una técnica de agrupamiento por Expectación-Maximización al catálogo de plantillas tipo M (ver sección 2.6.2). Se encontraron 25 plantillas E, como centroides de los grupos identificados de plantillas M (ver figura 20). Aproximadamente la mitad de las plantillas eran la representación invertida de la otra mitad.

Finalmente, se seleccionó un subconjunto de las plantillas M, llamadas plantillas F. La selección se realizó de manera visual, se mantuvo aquellas plantillas que fueran más representativas de las formas de ondas más comunes. La selección se limitó a aquellas plantillas que tuvieran una polaridad determinada, asumiendo que las mismas plantillas se pueden usar para las formas de onda de polaridad contraria, simplemente invirtiéndolas.

6.2.4. Extracción de características representativas las espigas y reducción dimensional. La representación de las espigas encontradas es una matriz multidimensional, cuyo tamaño depende de la frecuencia de muestreo a la que se haya registrado la señal. Sin embargo, una gran cantidad de esta información es redundante para los fines de la clasificación de las espigas. Existen varios métodos para extraer la información más representativa para la clasificación de cada espiga. Se utilizaron los 3 componentes

principales (ver sección 2.5.1) de las formas de onda para reducir la dimensión de los datos que se utilizaron como entrada para la siguiente etapa del algoritmo.

6.2.5. Clasificación de espigas. La clasificación es un paso esencial para obtener la actividad unitaria de las neuronas involucradas en un registro. Se realizó una comparación entre varios algoritmos de agrupamiento tradicionales (sección 2.6.2) y las variaciones propuestas en las secciones siguientes. En estos algoritmos de agrupamiento el número de grupos es un parámetro de entrada que afecta de manera importante los resultados obtenidos. Es por esto, que también se probaron diversos algoritmos para estimar el número de grupos presentes.

6.2.5.1. Agrupamiento por plantillas. Se calculó la distancia euclidiana entre las formas de onda a clasificar y las espigas del catálogo de plantillas (ver sección 6.2.3.2). Cada plantilla del catálogo representa un grupo. Se asignó la forma de onda al grupo de la plantilla que tuviera la menor distancia. La mayoría de los grupos de las plantillas quedaron vacíos.

Se realizaron pruebas utilizando registros extracelulares reales, donde las espigas habían sido clasificadas por expertos humanos de manera manual. Se compararon gráficamente los resultados de los expertos con los obtenidos por el agrupamiento por plantillas.

6.2.5.2. Expectación-Maximización segmentado en el tiempo. Un problema que se presenta en los registros de larga duración, es el cambio en las características de la señal en el tiempo. En especial, este cambio puede ser determinante en los registros crónicos o semi-crónicos, donde los cambios pueden incluir variaciones en las unidades neuronales presentes, así como en las características del ruido de la señal. Los movimientos de la posición del electrodo pueden generar cambios en las formas de las espigas, lo que afecta su representación en métodos como el análisis de componentes principales (PCA). El algoritmo de Expectación-Maximización segmentado en el tiempo, toma en cuenta estos cambios al dividir la señal en épocas, que son analizadas por Expectación-Maximización por separado. Al dividir el análisis en diferentes épocas surge la necesidad de encontrar una correspondencia de los grupos presentes en cada época y evaluar la estabilidad del modelo generado.

6.2.5.3. Expectación-Maximización segmentado en el tiempo por agrupación de centroides . Se inicia dividiendo las señales en el tiempo en épocas, donde cada época es analizada en primer lugar de manera independiente con el algoritmo de agrupamiento seleccionado. Se obtiene un modelo de las unidades neuronales presentes en esa época del registro. Los centroides de todas las épocas son agrupados con el mismo algoritmo que se utilizó para las espigas. Los grupos cuyos centroides sean agrupados juntos se consideran pertenecientes a un mismo grupo. Para cada época se calcula el valor de la función de aptitud del modelo generado para la época y el valor de la función para el modelo generado para la época anterior. Se comparan los valores de ambas épocas y se selecciona el modelo que maximice la función de aptitud, con cierto valor de preferencia por el modelo de la época anterior. Se repite el mismo procedimiento para cada época.

6.2.5.4. Expectación-Maximización segmentado en el tiempo por comparación de épocas. En este algoritmo se divide la señal en épocas de tiempo de manera similar al presentado en la sección 6.2.5.2. Se aplica el algoritmo de EM para la primera época y se obtienen el número de grupos en los que se segmentaron los datos. La salida del algoritmo de EM de la época anterior se utiliza como inicialización para la siguiente

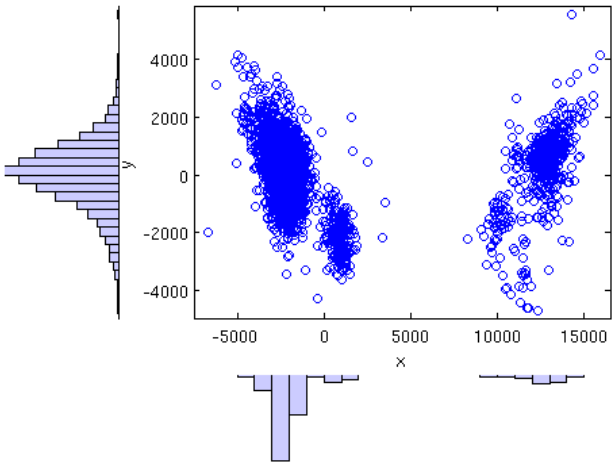


FIGURA 21. Estimación unidimensional gaussiana (EUC). Se calcula la proyección de los grupos en cada una de las dimensiones de los datos. Se busca el número de picos en cada una de las proyecciones. El mayor número de picos encontrados es considerado el número de grupos presentes.

época, de tal forma que, si la señal es estable el algoritmo convergerá al mismo número de grupos más rápidamente. De igual forma se calcula la asignación por EM con valores de inicialización aleatorios. Para cada época se calcula el valor de la función de verosimilitud para la asignación del algoritmo de EM con los valores de inicialización de la época anterior (*Q_fija*) y para la asignación en el caso de los valores iniciales aleatorios (*Q_libre*). Si la *Q_libre* es k veces mayor que la *Q_fija*, entonces el algoritmo selecciona el modelo generado por el algoritmo con los valores aleatorios. Esto permite que en caso de haber cambios en la señal, como apariciones o desapariciones de neuronas, el algoritmo sea capaz de detectar en que tiempo ocurrieron estos cambios y adaptarse.

6.2.5.5. Estimación del número de grupos presentes. Se utilizaron diversos métodos para estimar el número de grupos presentes en un conjunto de datos. El primer método probado fue el criterio bayesiano de información (ver sección 2.6.3.1), el cual es parecido al criterio de información de Akaike, utilizado por KlustaKwik (ver sección 2.8.1). También se probaron los siguientes métodos adicionales:

Estimación unidimensional gaussiana (EUC) . Este método busca picos en cada dimensión de los datos. Para cada una de las dimensiones que componen a los datos se calcula una convolución con una función gaussiana para difuminar las características de los datos. Después se buscan los picos en los datos, donde un pico es un valor máximo cuyos valores adyacentes son una desviación estándar menores al valor actual. Se cuentan el número de picos encontrados. Una vez contabilizados el número de picos en cada dimensión, el número de grupos presentes en los datos es el mayor número de picos encontrado (ver figura 21).

Localización de centroides en espiral (LSP) . La localización de centroides en espiral, permite estimar el número de grupos presentes y obtener una aproximación de las coordenadas de los centroides. Este algoritmo

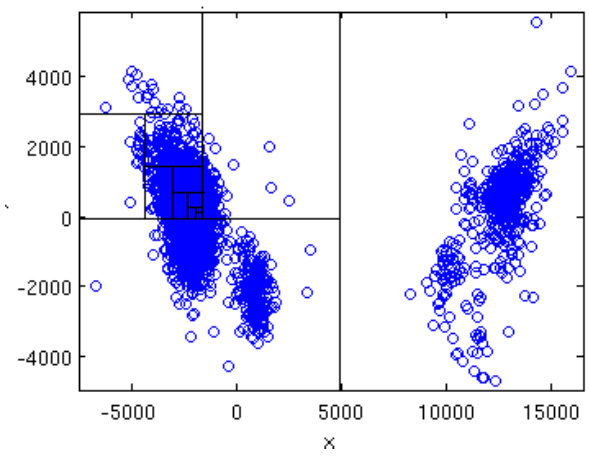


FIGURA 22. Localización de centroides en espiral (LSP). Es un proceso iterativo, donde para cada una de las dimensiones de los datos se busca el punto intermedio entre el valor máximo y el mínimo. Se cuentan el número de datos que caen en un sector y en el otro. Se repite el proceso para el sector que tenga la mayor cantidad de datos. Los pasos se repiten hasta que quede un dato en el área seleccionada. El dato resultante será el centroide.

se compone de tres pasos:

1. Búsqueda a profundidad, este paso se basa en un algoritmo de búsqueda binaria, donde se busca la coordenada que queda a la mitad entre el máximo y el mínimo de la dimensión del conjunto de datos. Se cuentan el número de datos que caen de uno y otro lado de la coordenada central. Se selecciona el cuadrante que contenga un mayor número de datos, o se selecciona de manera aleatoria alguno de los cuadrantes si ambos tienen el mismo número de datos. Se repite la operación para la siguiente dimensión de los datos (ver figura 22). Este paso del algoritmo termina cuando el cuadrante seleccionado hay un dato únicamente y la coordenada de ese dato es el centroide encontrado.
2. Búsqueda lateral, después de concluir el primer paso se ha encontrado el centroide del grupo que contiene la mayor cantidad de datos. El paso siguiente es buscar si existen otros grupos, esto se realiza inspeccionando los valores que se utilizaron para ir tomando la decisión de que sector elegir (el que contenía una mayor cantidad de datos), si la proporción entre los datos el sector seleccionado y el rechazado se encuentra entre un valor k y 0.5, se vuelve a realizar el primer paso en el sector que fue rechazado en la primer ocasión. Este valor k es la sensibilidad del algoritmo. Estos dos primeros pasos se repiten de manera recurrente, hasta haber explorado todas las posibles combinaciones de sectores que tengan una longitud mínima de d. Este valor de d especifica la distancia mínima que deben tener dos grupos para ser considerados independientes.
3. El último paso, es comprobar que los centroides encontrados pertenecen a grupos independientes. Se calcula un área de longitud d en cada una de las dimensiones de cada uno de los centroides. Utilizando los datos que caen dentro de esta área, se realiza nuevamente el paso 1, se verifica si el nuevo centroide se encuentra a una distancia menor de d de cualquier de los otros centroides, de ser

así se elimina el centroide y se repite este paso para los centroides restantes. Si el nuevo centroide encontrado se encuentra a una distancia mayor que d de todos los otros centroides, pero menor que $\frac{d}{4}$ del centroide original se considera que el algoritmo ha convergido y el centroide es aceptado.

6.3. Medidas de calidad de la señal y del desempeño de los algoritmos

6.3.1. Coeficiente señal-ruido (SNR). Es común utilizar el coeficiente señal-ruido (SNR) para la medición de la calidad de una señal. Sin embargo, no existe un consenso en cómo debe ser calculado este coeficiente. En diferentes trabajos donde se comparan algoritmos de clasificación, se utilizan definiciones diferentes del SNR y en algunos ni siquiera se define como se calculó el SNR. Esto complica el poder obtener una visión objetiva de la calidad de las señales que se usaron para realizar las comparaciones y por consiguiente de los algoritmos de detección y clasificación.

Otro problema del SNR es que es una medición *a priori* de la calidad de la señal, pues debemos conocer como está distribuida nuestra señal con respecto al ruido para poderlo calcular. Esto es especialmente problemático en los algoritmos de detección de señales, dado que el objetivo del algoritmo es determinar donde hay señal y donde hay ruido, por lo que calcular el SNR *a posteriori* de la detección lo convierte en una medida sesgada. Es por esto que sólo se reportan mediciones del SNR de las señales artificiales generadas, donde es posible aislar la señal del ruido antes de correr los algoritmos de detección.

La definición del SNR utilizado fue

$$SNR = \frac{A_{señal}}{6\, S_{ruido}}, \tag{6.3.1}$$

donde $A_{señal}$ es la amplitud máxima de la señal sin ruido y S_{ruido} es la desviación estándar del ruido.

Dado que la amplitud de las espigas de cada neurona será inversamente proporcional a la distancia a la que se encuentre del electrodo, el SNR de cada neurona en la señal tendrá un valor diferente. Para poder calcular el SNR total de la señal, se calculó el promedio del SNR de cada neurona ponderado al número de espigas de esa neurona presentes en la señal.

6.3.2. Medición del desempeño de un clasificador. Para las mediciones de la calidad de los clasificadores se realizó una comparación entre el AUC (ver sección 2.3) y el MCC (ver sección 2.3.2).

Se utilizó un vector objetivo generado por el simulador de redes neuronales para las señales artificiales o por un experto para el caso de las señales biológicas. Se considera un acierto cuando el tiempo de una espiga detectado por un algoritmo se encuentra hasta 1 ms antes o después del tiempo de una espiga presente en el vector objetivo, se utilizó como vector objetivo los tiempos de disparo generados por el simulador de redes neuronales para las señales artificiales (ver sección 6.5) o la información de la discriminación realizada por un experto para el caso de las señales biológicas.

Se implementó una versión modificada para el cálculo de un MCC capaz de medir el desempeño de los algoritmos de agrupamiento. El MCC está definido para la clasificación entre dos estados de una señal buscada: presente o ausente. Sin embargo, en el caso de los algoritmos de agrupamiento, se tiene un conjunto de asignaciones de miembros a sus grupos putativos, identificados por etiquetas. Para calcular un MCC de

Grupos algoritmo	Combinaciones				
	Grupo 1 real	Grupo 2 real	MCC Individual		MCC Grupal
	1	2	0.0057	0.896	0.7766
	1	3	0.0057	0.0006	0.00114
	2	1	0.3772	0	0.18664
	2	3	0.3772	0.0006	0.18104
	3	1	0.5062	0	0.24498
	3	2	0.5062	0.896	0.81606

TABLA 3. Ejemplo del cálculo del MCC para varios grupos. En este ejemplo el conjunto de datos original tiene dos grupos, mientras que el algoritmo de agrupamiento generó tres grupos. En cada renglón de la tabla se hace las combinaciones de los tres grupos del algoritmo en los dos grupos originales. Se calcula el MCC grupal al sumar aquellos datos que coincidan entre los grupos originales y los calculados. Se selecciona la combinación que tenga el mayor MCC. En este ejemplo el mejor resultado se obtiene cuando el grupo original 1 coincide con el grupo 3 del algoritmo y el grupo original 2 coincide con el grupo 2 del algoritmo.

los resultados de los algoritmos de agrupamiento, se definió un conjunto de datos objetivo, que contiene el resultado óptimo a obtener. Se creó una matriz de correspondencias entre las etiquetas de los grupos de los datos objetivo y la salida de los algoritmos, de tal forma que se tuvieran todas las posibles combinaciones. Esto con el fin de evitar que el mismo grupo tuviera identificadores diferentes en ambos conjuntos y fuera marcado como un falso positivo. De manera individual para cada grupo dentro de cada una de las combinaciones del arreglo, se compararon los miembros del grupo contra los de los datos objetivo, calculando el total de aciertos, falsas alarmas, fallas y rechazos correctos. Después, se sumaron estas estadísticas para todos los grupos dentro de una combinación de etiquetas. Sobre estas estadísticas se calculó el MCC con la ecuación 2.3.1. Para cada combinación de grupos se obtuvo su MCC, al final se compararon todas las MCC de las diferentes combinaciones y se seleccionó la que tuviera el mejor MCC, dado que esta es la combinación en la que coinciden la mayor cantidad de miembros entre el algoritmo y los datos objetivo con esa combinación de etiquetas de grupos (ver tabla 3).

6.4. Optimización

La discriminación y agrupamiento de espigas es un problema complejo, que puede ser abordado usando diferentes algoritmos. La aplicación de un algoritmo específico en un paso del proceso puede afectar la salida del paso siguiente. Esta complejidad aumenta conforme se agregan pasos al proceso. Se diseñó un algoritmo genético capaz de buscar los parámetros óptimos del conjunto de pasos necesarios para la discriminación de espigas. En primera instancia se implementó un algoritmo genético basado en una base de datos relacional. Donde las relaciones entre los componentes del algoritmo, así como los resultados de cada posible solución, eran almacenados en esta base de datos. Esta implementación requería la instalación y mantenimiento de

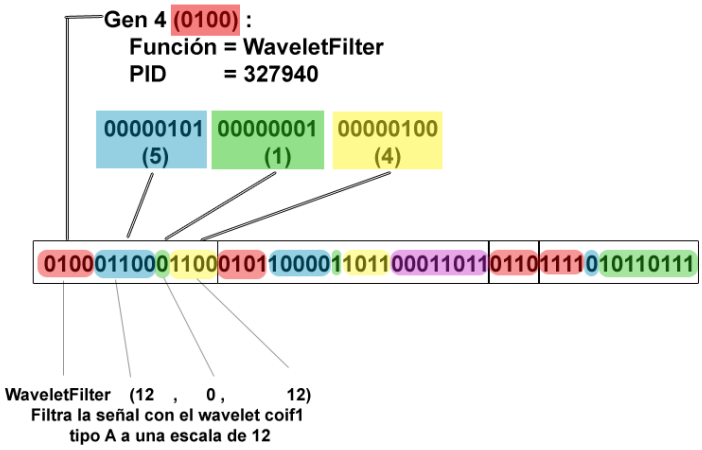


FIGURA 23. Definición genética de las soluciones de un algoritmo genético. Un gen contiene la definición de un alelo, el cual a su vez, contiene la definición de los parámetros de una función. El algoritmo genético busca la mejor combinación de parámetros de la función dada para obtener cierto resultado.

un servidor de base de datos, el cual no era apropiado para el objetivo del programa para la creación y comparación de algoritmos de discriminación Sort Lab (ver sección 7.7). Por lo que, se creó una segunda versión del algoritmo genético. La cual tuvo como objetivo poder ejecutarse en una computadora individual, sin necesidad de un servidor de base de datos adicional. Esta versión se basó en las librerías de algoritmos genéticos de Matlab.

6.4.1. Algoritmo genético basado en una base de datos relacional. La representación genética de las soluciones se hace en forma de una cadena binaria, con el fin de permitir la búsqueda dinámica de la mejor función y sus parámetros para la resolución del problema. Se definió una cadena binaria denominada gen. El gen contiene la definición de las funciones disponibles, así como de los parámetros que recibe esta función. La solución específica, función y parámetros específicos, se encuentran definidos en un alelo (ver figura 23). Es a este alelo al que se le aplica las operaciones de reproducción del algoritmo genético (ver sección 2.7).

Un genoma es un conjunto de soluciones para el problema de la clasificación de espigas, está compuesto por un conjunto de alelos, que representan las funciones necesarias para llevar a cabo esta solución, así como de los parámetros ideales de cada función. La función de este algoritmo genético es generar dinámicamente series de funciones capaces de llevar a cabo la discriminación de las espigas maximizando el MCC. La flexibilidad que permite la definición de genes y alelos, permite agregar nuevas funciones que pueden ser usadas en el dominio de soluciones utilizado por el algoritmo genético.

Esta versión de algoritmo genético se implementó con un depósito de datos en una base de datos relacional, donde las relaciones entre las entidades que componen al algoritmo genético: genes, alelos y

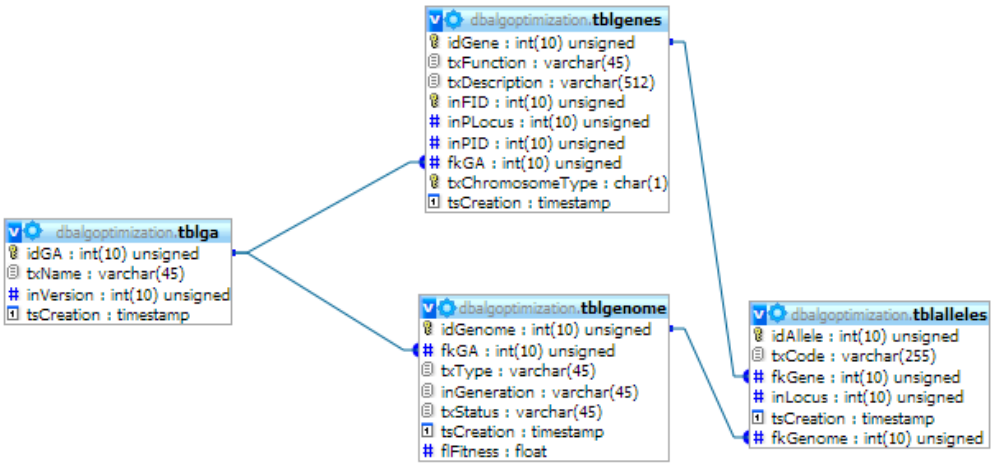


FIGURA 24. Diagrama de entidad-relación de la base de datos del algoritmo genético. Cada cuadro representa una tabla que contiene ordenados en campos y las líneas representan las relaciones entre estas tablas de datos.

genomas, se pueden representar fácilmente. El diagrama de entidad-relación se puede ver en la figura 24. Una de las razones más importantes para la implementación en esta plataforma fue que fuese compatible con el sistema de computo distribuido antes creado (ver apéndice B). Esto con el fin de poder distribuir la carga de trabajo de la búsqueda de soluciones entre diferentes computadoras, previendo que la ejecución de una solución tomaría una gran cantidad de tiempo.

6.4.2. Algoritmo genético usando las librerías de Matlab. Esta versión del algoritmo genético está basada en las librerías de Matlab, lo que permite que se ejecute en una computadora aislada. Se implementó con el fin de tener un método automático para optimizar los algoritmos de discriminación de espigas. En esta versión no existe un catálogo de funciones y parámetros preestablecido. En su lugar, cada función que forma parte de un algoritmo o programa para la discriminación, debe ser capaz de proveer la definición de la cadena binaria necesaria para la optimización de sus parámetros Un programa para la discriminación de espigas se compone de diversas funciones (ver sección 7.7). Donde cada función debe funcionar en alguna de tres modalidades, de acuerdo al número de parámetros que recibe:

1. Si no recibe parámetros, la función debe regresar las definiciones de los parámetros que requiere para su funcionamiento. Cada parámetro debe definir su nombre, su valor por defecto, una descripción y la longitud de la cadena binaria necesaria para cubrir el rango de valores que recibe. 2. Si únicamente recibe una estructura con los valores de sus parámetros de entrada, la función debe verificar que cada uno de los parámetros que contiene la estructura sean traducibles a un valor usable por la función y que estos valores se encuentren dentro de los rangos definidos por la función. 3. Si recibe una variable con datos y una estructura con los valores de los parámetros a utilizar, así como cualquier parámetro adicional. La función debe usar los parámetros provistos para realizar el análisis necesario de los datos.

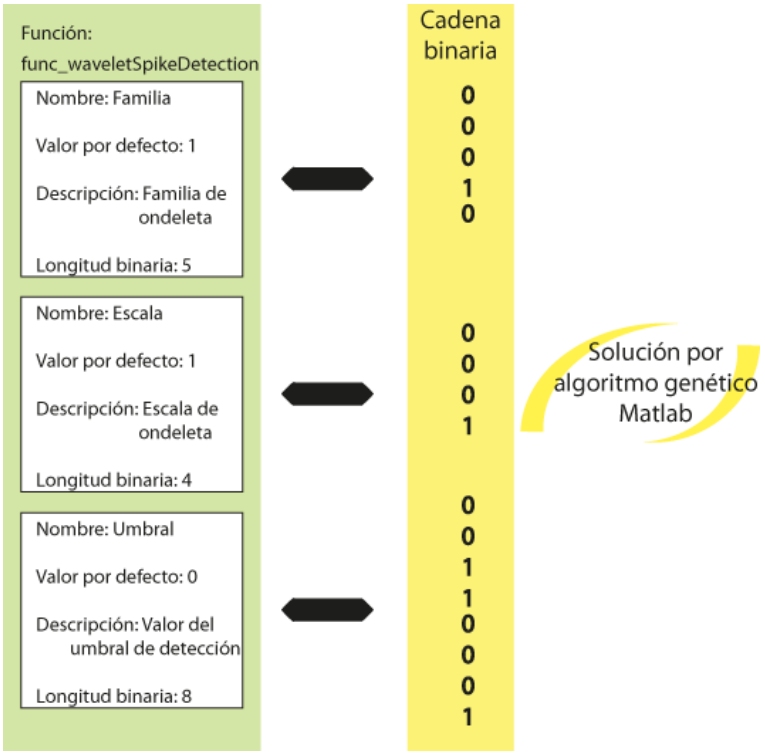


FIGURA 25. Ejemplo de la creación dinámica de una cadena binaria para la optimización de parámetros por un algoritmo genético. Cada función a optimizar por el algoritmo genético, es capaz de regresar la definición de la cadena binaria que requiere para representar sus parámetros.

Esta estructura modular permite que se genere la cadena binaria necesaria por el algoritmo genético de manera dinámica para cada programa creado, por lo que se pueden agregar nuevos pasos a un programa y su definición genética se actualizará automáticamente.

6.5. Señal artificial

El uso de una señal artificial es una práctica común para evaluar el comportamiento de un algoritmo de detección y clasificación de espigas. Usualmente se utilizan plantillas de formas de onda de espigas obtenidas de registros electrofisiológicos, a las cuales se les agrega ruido ya sea artificial o de segmentos de registros donde no hay espigas, para generar señales artificiales. Sin embargo, estas señales no reflejan la complejidad de las variaciones en la formas de onda de las espigas en un registro electrofisiológico real. Tomando como referencia el método para generar una señal artificial que se describe en Martinoia et al. (2004), se implementó un módulo para generar señales artificiales. Este módulo toma como entrada una señal de los voltajes de membrana de una o varias poblaciones de neuronas artificiales. La señal base de las neuronas artificiales es generada en NEST (ver sección 2.8.5) utilizando el lenguaje PyNN para definir el modelo de la red neuronal (ver sección 2.8.4).

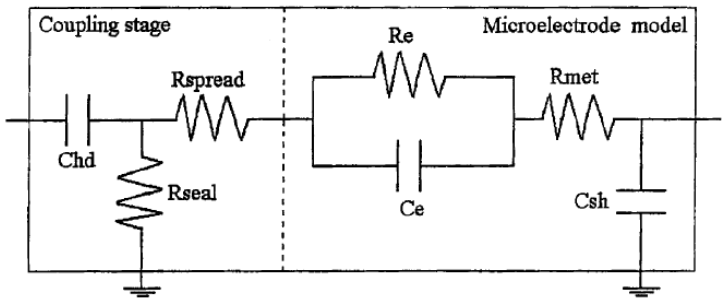


FIGURA 26. Modelo de la interfase célula-microelectrodo para un registro extracelular (tomado de Martinoia et al. (2004)). Donde C_{hd} es la capacitancia membrana celular-electrolito, R_{seal} es la resistencia de sellado entre la célula y el microelectrodo, R_{spread} es la resistencia de propagación, $R_e$es la resistencia de fuga, C_e es la capacitancia de la interfase electrolito-microelectrodo, R_{met} es la resistencia de la conexión metálica el microelectrodo y C_{sh} es un término que incluye todas las capacitancias a tierra del circuito.

Simulando la interfase entre una neurona y un electrodo, como un sistema lineal invariante con el tiempo (SLIT), se pueden obtener las formas de onda de cada una de las neuronas como serían recibidas sin ruido por un electrodo extracelular. En Martinoia et al. (2004) se utilizó SPICE, un simulador de circuitos electrónicos, para modelar estas deformaciones. Sin embargo, en este trabajo se utilizó la función de transferencia del SLIT para modelar la deformación que sufre la señal intracelular en el registro extracelular. Este método tiene la ventaja de que se pueden modelar los cambios de la señal desde Matlab, sin necesidad de usar programas adicionales.

La función de transferencia se obtuvo a partir de la descripción del circuito electrónico del modelo de la figura 26. Se utilizó la herramienta para Matlab SCAM (Cheever, 2009) para obtener la ecuación de la función de transferencia correspondiente a este circuito. Esta herramienta toma como entrada un archivo con la definición de los componentes y la conexión de los nodos que forman un circuito electrónico y genera como salida la representación simbólica de las ecuaciones de los voltajes y corrientes del circuito. La ecuación de la función de transferencia se obtiene de dividir las ecuaciones del voltaje de entrada entre el voltaje de salida. Es importante notar, que el modelo de la interfase célula-microelectrodo de Martinoia et al. (2004) está hecho para un arreglo multielectrodos de mediciones *in vitro*. Sin embargo, incluye los mismos componentes que los modelos de electrodos para registros *in vivo*. De cualquier manera, es relativamente trivial cambiar el circuito del modelo y encontrar una nueva función de transferencia gracias al uso de SCAM.

La función de transferencia del modelo célula-electrodo de la figura 26 obtenida fue:

$$H(s) = \frac{as}{b_1 s^3 + b_2 s^2 + b_3 s + 1}, \tag{6.5.1}$$

donde

$$
\begin{aligned}
a &= C_{hd}R_{seal}(1 + sC_eR_e),\\
b_1 &= C_{hd}R_{seal}R_{spread}C_eR_eC_{sh} + C_{hd}R_{seal}C_eR_eR_{met}C_{sh},\\
b_2 &= C_{hd}R_{seal}R_eC_{sh} + C_{hd}R_{seal}C_eR_e + C_eR_{spread}R_eC_{sh} + R_{seal}C_eR_eC_{sh}\\
&\quad + C_{hd}R_{seal}C_{sh}R_{met} + C_eR_eR_{met}C_{sh} + C_{hd}R_{seal}R_{spread}C_{sh},\\
b_3 &= R_eC_{sh} + R_{seal}C_{sh} + R_{spread}C_{sh} + C_eR_e + C_{sh}R_{met} + C_{hd}R_{seal}.
\end{aligned}
$$

La variable R_{seal} es dependiente de la distancia entre la célula y el electrodo, por lo que la función de transferencia es dependiente, de igual forma, de esta distancia. Gracias a esto, se puede simular la señal de salida de diferentes neuronas que se encuentren a distancias distintas del electrodo. Utilizando la función *lsim* de Matlab y la ecuación de la función de transferencia, se ajustaron los parámetros de acuerdo a la distancia de cada neurona y se obtuvo la señal de salida de cada una de las neuronas. Las señales de todas las neuronas se sumaron para obtener una señal combinada, de manera similar a como la recibiría un electrodo extracelular.

Finalmente, se le agregó ruido de diferentes fuentes a la señal artificial:

- Ruido rosa, que simula el ruido característico en los sistemas electrónicos.
- Ruido gaussiano, simulando ruido ambiente.
- Segmentos aleatorios de señales sinusoidales, para simular artefactos por microfonía y movimientos del electrodo.

6.6. Comparación de los algoritmos

6.6.1. Generación de señales artificiales de prueba. Para comparar el rendimiento de los algoritmos diseñados (ver sección 6.2), se generó un conjunto de cuatro señales, cada una de 300s de duración (ver sección 6.5) con diferentes niveles de calidad (ver sección 7.3). Todas las señales de 300s se crearon con la misma información base de los potenciales de membrana obtenida de PyNN de la red neuronal artificial. También se generó una señal de 15 s de duración con características similares, pero las conexiones de la red se realizaron de manera aleatoria, por lo que la información de los potenciales de membrana fue distinta que la usada para las cuatro señales de 300s. Esta señal de 15 s, se utilizó como entrada del algoritmo genético para la optimización de los parámetros de las funciones.

Para la simulación de las cuatro señales de 300s en PyNN, se creó una red de 49 neuronas: 2 neuronas inmediatas, cuyas señales deben ser detectadas y discriminadas por los algoritmos; 7 neuronas cercanas a una distancia de 200 µm, cuyas señales distorsionadas se pueden llegar a apreciar en los registros; y 40 neuronas lejanas a una distancia de 400 µm, cuyos potenciales de acción no son discernibles de manera individual, pero aparecen como ruido biológico en la señal artificial. Las tres poblaciones de neuronas están conectadas de manera convergente. Se conectó un generador de espigas con una distribución de Poisson en la población más lejana. Se crearon conexiones excitatorias (70 %) e inhibitorias (30 %) entre las poblaciones y se ajustaron los pesos de las conexiones de tal forma que las tres poblaciones tuvieran una tasa de disparo entre 10 y 20 Hz durante la simulación. Todas las señales se generaron con los mismos niveles de ruido (ver

Tipo de ruido	Parámetros	
Rosa	Amplitud	6
Gaussiano	Amplitud	6
Ondas sinusoidales	Amplitud	6
	Presencia (porcentaje de la señal)	20 %
	Frecuencia	200Hz
	Duración por segmento	0.4 s

TABLA 4. Parámetros de generación de ruido en las señales artificiales.

Señal	PP1		PP2	PP3	PIE	VE	SNR	
Alta	Neurona	Posición	(200,0)	(400,0)	(0,0)	(0,0)	1	2.5947
	1	(5,0)					2	2.2439
	2	(20,0)					Promedio	2.3081
Media	Neurona	Posición	(200,0)	(400,0)	(0,0)	(0,0)	1	1.6125
	1	(45,0)					2	1.4047
	2	(60,0)					Promedio	1.4412
Baja	Neurona	Posición	(200,0)	(400,0)	(0,0)	(0,0)	1	1.0134
	1	(100,0)					2	0.9505
	2	(115,0)					Promedio	0.96206
Movimiento	Neurona	Posición	(200,0)	(400,0)	(0,0)	(1,1)	1	2.5947
	1	(5,0)					2	2.2439
	2	(20,0)					Promedio	2.3081

TABLA 5. Parámetros de distancia neurona-electrodo y SNR de señales artificiales. Donde PP1 es la posición para la población 1 de neuronas, PP2 es la posición para la población 2, PP3 es la posición para la población 3, PIE es la posición inicial del electrodo, VE es la velocidad del electrodo, SNR es el coeficiente señal-ruido (para cada neurona de la población 1 y el promedio para la población en general).

tabla 4), únicamente se modificó la distancia a la que se colocaron las 2 neuronas inmediatas, que son las que generan las señales a discriminar, obteniendo señales con un menor SNR a una mayor distancia. En la simulación de tres de las señales, se mantuvieron estáticas las distancias de las neuronas al electrodo durante todo el registro, pero en la cuarta señal se simuló que el electrodo se desplazaba cada diez segundos con una velocidad fija. Las características para cada señal se pueden ver en la tabla 5.

6.6.2. Algoritmos de detección y clasificación. Se creó un algoritmo de detección (DET_NORM) de tres pasos: filtrado de la señal por ondeletas (ver sección 6.2.1), detección de espigas por ondeletas (ver sección 6.2.2.1) y filtrado de formas de onda por plantillas (ver sección 6.2.3.1). Se utilizó el algoritmo genético, descrito en la sección (ver sección 2.7), con la señal artificial de 15 s para obtener los parámetros de cada uno de los pasos del algoritmo. Se creó una versión rápida del algoritmo de detección (DET_RAP),

que solamente tenía el paso de la detección de espigas por ondeletas. Se realizó la comparación en el tiempo de ejecución de ambas versiones.

Se crearon un conjunto de algoritmos de clasificación, todos utilizan el algoritmo DET_NORM para la detección, agregando una etapa de clasificación de espigas, que consistió en los pasos de reducción dimensional y agrupamiento. Se usaron los primeros 3 componentes del PCA para la reducción dimensional. Se crearon combinaciones de los algoritmos de agrupamiento y métodos de estimación de grupos descritos en la sección 6.2.5 (ver tabla 6). Se realizaron comparaciones entre los resultados obtenidos por los algoritmos segmentados en el tiempo, sobre las mismas señales, utilizando épocas de 45s y de 20s de longitud.

Usando cada unas de las 4 señales de prueba generadas de 300 segundos (ver sección 6.6.1), se calculó el MCC (ver sección 6.3.2) para las dos versiones de los algoritmos de detección y para cada algoritmo de clasificación creado. De igual manera se evaluó el desempeño de los programas KlustaKwik (ver sección 2.8.1) y OSort (ver sección 2.8.2) con cada una de las señales. El programa KlustaKwik sólo realizá la etapa de la clasificación de espigas, por lo que se utilizó el algoritmo DET_NORM para la fase de detección. Sin embargo, OSort lleva a cabo las etapas de detección y clasificación con sus propios algoritmos. Se probó utilizar el algoritmo DET_NORM para la detección como entrada para la etapa de clasificación de OSort, pero los resultados fueron pobres.

Finalmente, se evaluó el desempeño de todos los algoritmos con tres señales de registros extracelulares reales, las cuales fueron obtenidas del área de la corteza premotora medial de un monos rhesus (*Macaca mulatta*) mientras realizaba una tarea dirigida. Estos registros fueron clasificados como de nivel 1, 3 y 5 de acuerdo a lo descrito en la sección 6.1.1. Estas señales fueron discriminadas de manera manual por un experto. Se utilizaron los resultados de la discriminación del experto como el vector objetivo para calcular el MCC de cada algoritmo (ver sección 6.3.2).

Abreviación	Algoritmo de clasificación	Sección	Método de estimación de grupos	Sección
EM_EUC	Expectación-Maximización	2.6.2	Estimación unidimensional gaussiana	6.2.5.5
EM_BIC	Expectación-Maximización	2.6.2	Criterio bayesiano de información	2.6.3.1
EM_LSP	Expectación-Maximización	2.6.2	Localización de centroides en espiral	6.2.5.5
EM_CC_EUC	Expectación-Maximización segmentado en el tiempo por agrupación de centroides	6.2.5.3	Estimación unidimensional gaussiana	6.2.5.5
EM_CC_BIC	Expectación-Maximización segmentado en el tiempo por agrupación de centroides	6.2.5.3	Criterio bayesiano de información	2.6.3.1
EM_CC_LSP	Expectación-Maximización segmentado en el tiempo por agrupación de centroides	6.2.5.3	Localización de centroides en espiral	6.2.5.5
EM_CEP_EUC	Expectación-Maximización segmentado en el tiempo por comparación de épocas	6.2.5.4	Estimación unidimensional gaussiana	6.2.5.5
EM_CEP_BIC	Expectación-Maximización segmentado en el tiempo por comparación de épocas	6.2.5.4	Criterio bayesiano de información	2.6.3.1
EM_CEP_LSP	Expectación-Maximización segmentado en el tiempo por comparación de épocas	6.2.5.4	Localización de centroides en espiral	6.2.5.5
KLUSTA	KlustaKwik	2.8.1		
OSORT	OSort	2.8.2		

TABLA 6. Combinaciones de los algoritmos de clasificación probados. Los primeros 9 algoritmos fueron programados de acuerdo a lo descrito en la sección 6.2. Los últimos dos algoritmos, KlustaKwik y OSort, son programas de discriminación de espigas elaborados por otros grupos de trabajo.

Capítulo 7

Resultados

7.1. Acondicionamiento y caracterización del contenido de frecuencias de señales de registros extracelulares

Utilizando el método descrito en la sección 6.1.1 se obtuvieron 9,709 segmentos de señal electrofisiológica real, de 10 segundos de duración, que fueron clasificados de acuerdo a su nivel de calidad del 0 al 5. También se obtuvieron las formas de onda de 587,851 espigas y, prácticamente, igual número de segmentos de señal entre las espigas clasificados como ruido.

Se calcularon los espectros de frecuencias de cada uno de los tipos de señales (ver figura 27). Se encontró que el espectro de frecuencias es muy parecido en los tres casos (señal con espigas y ruido, señal con sólo ruido y señal de una espiga). Sin embargo, se puede observar que el espectro de la espiga tiene una menor resolución que los otros dos, esto se debe a que el número de muestras de una espiga está limitado por el tamaño del segmento de 3 ms que se utilizó (121 muestras).

En los incisos *a* y *b* de la figura 28 se obtuvieron los *FFT* de dos segmentos de 10 segundos no contiguos de una misma señal. Se puede observar que el contenido de frecuencias no cambia de manera significativa durante un mismo registro. Sin embargo, también encontramos que el *FFT* sí cambia entre diferentes ensayos.

El análisis estadístico de la presencia de picos en el espectro de frecuencias de señales con calidad 0 y con calidad 5 se muestra en la figura 30. Se puede observar que no existen picos con un porcentaje de aparición del 100 %, y que los picos presentes aparecen tanto en señales identificadas como de sólo ruido, así como en las señales de calidad 5. Esto sugiere que los picos encontrados se deben al ruido ambiental en las señales y no a las espigas que aparecen en ellas. En conclusión el FFT no es una medida confiable para caracterizar los registros con o sin actividad neuronal, debido a que, aún en las señales con actividad

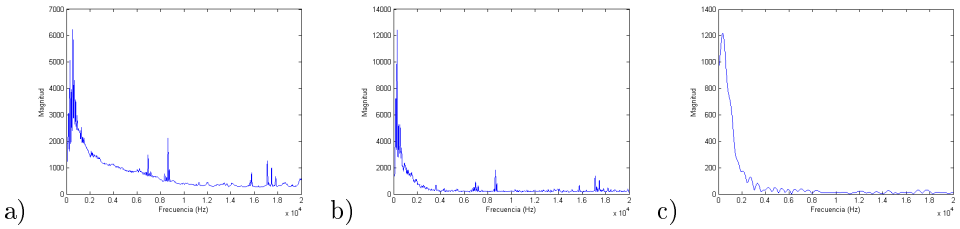


FIGURA 27. a) *FFT* de un segmento de 10 segundos de señal electrofisiológica que contiene espigas y ruido b) *FFT* de un segmento de señal identificada como ruido c) *FFT* del segmento de señal correspondiente a una espiga.

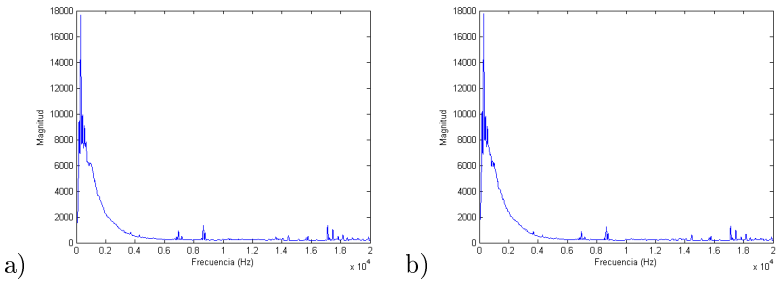


FIGURA 28. El espectro de frecuencias de la señal no cambia significativamente en el tiempo en un mismo ensayo. a) *FFT* del segmento del segundo 0 al 10 de una señal de registro b) *FFT* del segmento del segundo 500 al 510 de la misma señal.

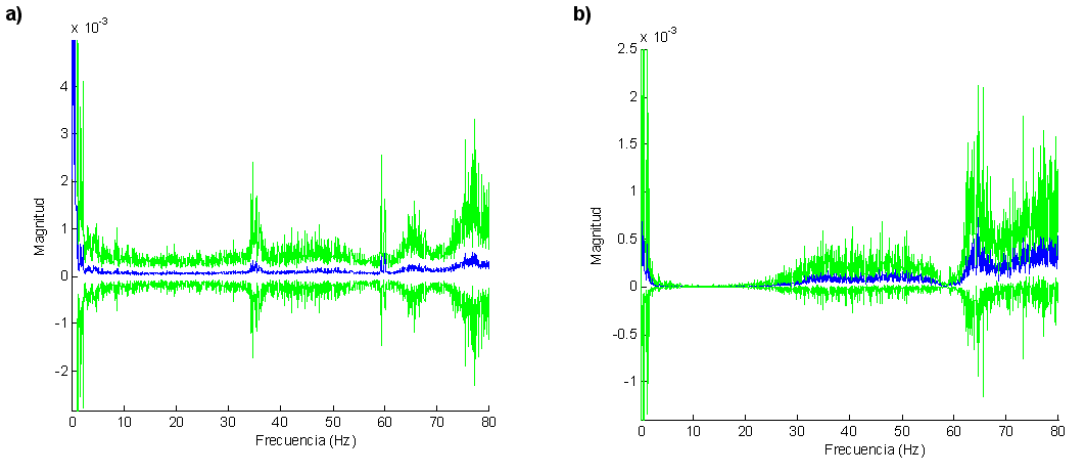
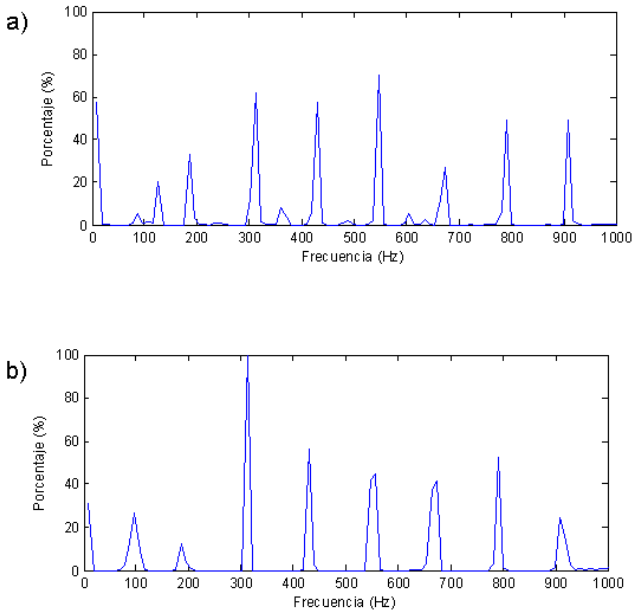


FIGURA 29. Espectros de frecuencia promedio (azul) y su desviación estándar (verde) para el rango de frecuencias de 0 a 80 Hz para a) conjunto de señales de calidad 0 y b) conjunto de señales de calidad 5.



c)

F. (Hz)	C=0 (%)	C=5 (%)
0	57.21	30.95
88	0.7870	26.43
300	**62.27**	**99.82**
420	**57.31**	**56.26**
540	70.64	42.26
550	0.23	44.86
650	12.04	37.56
660	27.04	41.47
780	**49.18**	**52.60**
1260	49.58	34.42

FIGURA 30. Porcentaje de presencia de picos en el FFT de señales de diferentes calidades. a) Señales de calidad 0 (sólo ruido) b) Señales de calidad 5 (espigas de gran amplitud y presencia de ruido de baja amplitud) c) Tabla de comparación de porcentajes de picos en señales de calidad 0 y 5 (en rojo aquellos picos de frecuencias que se encuentran presentes en más del 50 % de las señales de calidad 5).

neuronal, la proporción de la señal en la que hay espigas presentes es mucho menor que donde hay sólo ruido, por lo que el contenido de frecuencias del ruido domina el espectro de ambas.

Se filtraron las señales con un filtro Butterworth con una frecuencia de corte de 2000Hz, con el fin de eliminar los ruidos de alta frecuencia. Las señales filtradas fueron analizadas nuevamente para buscar los picos en sus *FFT*, sin embargo se obtuvo un resultado en la banda de frecuencias de interés similar al anterior. No se encontró una diferencia en la aparición de picos en el espectro de frecuencias de señales que contienen espigas, de aquellas que contienen únicamente ruido.

7.2. Pasos del algoritmo de discriminación de espigas

7.2.1. Comparación de filtros.

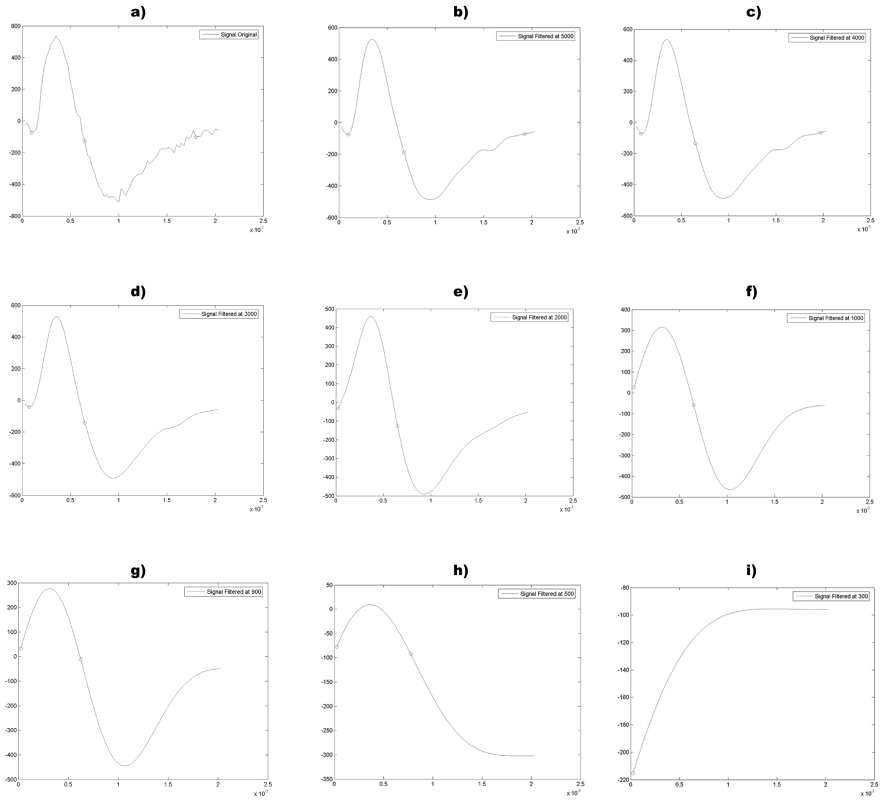


FIGURA 31. Comparación de formas de onda resultantes de procesar la señal de una espiga a través de filtros Butterworth con diferentes frecuencias de corte. a) Espiga original; espiga procesada con un filtro Butterworth con frecuencias de corte de b)5000Hz c)4000Hz d)3000Hz e)2000Hz f)1000Hz g)900Hz h)500Hz i)300Hz.

7.2.1.1. Filtros Butterworth . Aplicando filtros Butterworth (ver sección 6.2.1.1) con diferentes frecuencias de corte a una forma de onda de una espiga (ver figura 31) se obtuvieron los datos de la tabla 7.

En base a la información obtenida se determinó que la frecuencia de corte mínima que debe ser usada para que la espiga mantenga su forma de onda en cuanto a longitud y amplitud es de 2000 Hz.

7.2.1.2. Filtros por ondeletas. Se realizó el filtrado de una forma de onda de una espiga por medio de ondeletas (ver sección 6.2.1.2). En la figura 32 se puede ver la comparación de la forma de onda resultante de procesar una espiga con un filtro basado en ondeletas, se observa que se elimina el ruido de alta frecuencia sin modificar la forma de la espiga en longitud ni amplitud. Es por estas características que se decidió utilizar este tipo de filtros en la fase de detección del algoritmo de discriminación, en lugar de un filtro Butterworth.

7.2.2. Detección de espigas. Utilizando el método de detección por medio de la transformada estacionaria discreta de ondeleta (ver sección 6.2.2.1), se creó una base de datos donde se relacionan las AUC de las curvas ROC (ver sección 2.3) para cada familia y escala de ondeletas. Ordenando los registros

Frecuencia de corte	T.A.	T.B.	Amplitud
Original	0.037 ms	0.073ms	1020
5000 Hz	0.036 ms	0.073 ms	992
4000 Hz	0.037 ms	0.11 ms	990
3000 Hz	0.038 ms	0.12 ms	987
2000 Hz	0.026 ms	0.22 ms	930
1000 Hz	0.03 ms	0.23 ms	780
900 Hz	0.03 ms	0.48 ms	750
800 Hz	0.078 ms	0.52 ms	680
700 Hz	0.21 ms	1.9 ms	490
600 Hz	NA	NA	405
500 Hz	NA	NA	310
300 Hz	NA	NA	118
100 Hz	NA	NA	100

TABLA 7. Variaciones en las formas de onda de espigas, por medio de la medición del tiempo A (T.A.), despolarización y repolarización hasta el cruce en cero y el tiempo B (T.B.), desde el cruce de cero de la repolarización hasta el regreso al nivel basal, al ser procesadas por filtros Butterworth de frecuencia de corte *Fc*. Los valores NA significan que no fue posible calcular los tiempos, debido a que no se encontraron los puntos de referencia en la forma de onda obtenida en los intervalos de tiempo establecidos.

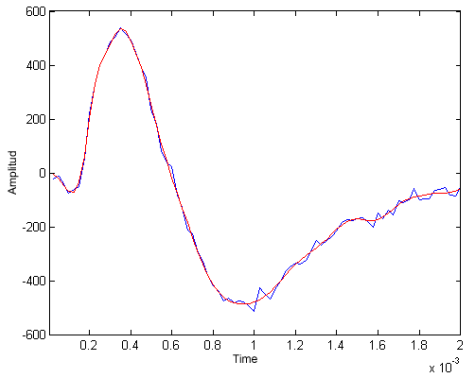


FIGURA 32. Señal original de una espiga con ruido (azul) sobrepuesta sobre la señal obtenida al realizar el proceso de eliminación de ruido por ondeletas utilizando una ondeleta madre Daubechies 7 con un nivel 2 (rojo).

	Espigas		Ruido	
	Familia y escala	AUC	Familia y escala	AUC
1	db2 escala 1 (A)	0.996155	db3 escala 4 (A)	0.997851
2	db3 escala 1 (A)	0.996128	rbio3.9 escala 5 (D)	0.997684
3	bior3.7 escala 1 (A)	0.996071	db5 escala 5 (D)	0.997363
4	bior6.8 escala 1 (A)	0.996056	db4 escala 5 (D)	0.997048
5	coif1 escala 1 (A)	0.996045	db5 escala 6 (D)	0.996723
6	dmey escala 1 (A)	0.996038	db4 escala 6 (D)	0.996696
7	rbio6.8 escala 3 (A)	0.996028	bior3.7 escala 5 (D)	0.996561
8	rbio6.8 escala 1 (A)	0.996017	db6 escala 5 (D)	0.996265
9	coif1 escala 2 (A)	0.996013	db4 escala 4 (A)	0.996083
10	db2 escala 2 (A)	0.996012	db3 escala 7 (D)	0.995905

TABLA 8. 10 ondeletas con mayor AUC para detección de espigas y ruido ambiente.

de manera descendente por su valor de *AUC*, se obtuvo una lista de los mejores clasificadores. La lista de las 10 mejores ondeletas madres y sus escalas para clasificar espigas y ruido se puede ver en la tabla 8.

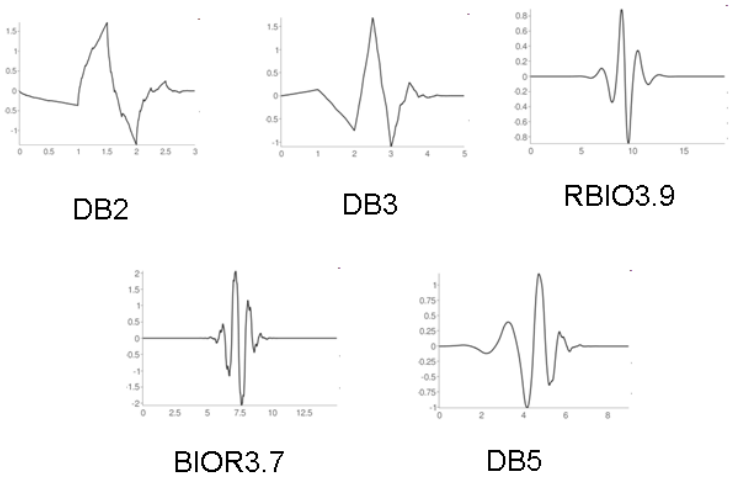


FIGURA 33. Formas de onda de cinco familias de ondeletas.

En la figura 34 se puede ver un segmento de señal con las espigas marcadas en rojo, así como una gráfica de tiempos de los eventos encontrados por la ondeleta rbio6.8 con escala 3 para espigas y la ondeleta bior3.7 con escala 5 para el ruido ambiente.

7.2.3. Generación del catálogo de espigas. Utilizando el método descrito en la sección 6.2.3.2 se obtuvieron una base de datos de 9 plantillas de formas de onda (ver figura 35), que se utilizan para el filtrado de las formas de onda.

7.2.3.1. Representación de plantillas en un espacio tridimensional. Utilizando el algoritmo descrito en la sección 2.5.2, se obtuvieron las coordenadas para cada plantilla representadas en la figura 36.

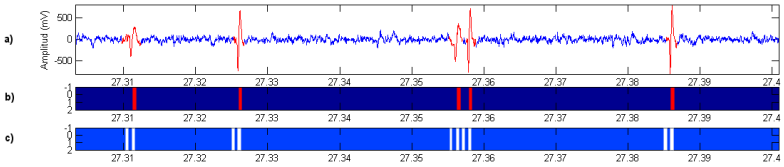


FIGURA 34. Detección de espigas por ondeletas en una señal de registro extracelular. a) Señal original con espigas en rojo b) en rojo detección de espigas con una ondeleta rbio6.8 con escala 3 c) en azul claro detección de ruido con una ondeleta bior3.7 con escala 5.

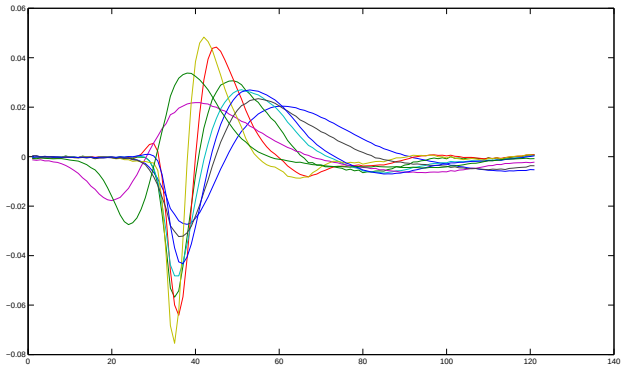


FIGURA 35. Catálogo de espigas. Formas de ondas normalizadas con conjunto de 9 plantillas tipo F.

Utilizando el método de PCA (ver sección 2.5.1) se obtuvo una representación tridimensional del catálogo de plantillas, de manera similar a lo realizado en la sección 2.5.2. En la figura 37 se pueden ver los resultados obtenidos. Se compararon las coordenadas obtenidas por el método de geometría de distancias y por PCA. Las coordenadas obtenidas por PCA resultaron más convenientes, debido a que la proyección de la primer coordenada generaba un conjunto de datos distribuido de manera más uniforme sobre el eje en el que se proyectan las plantillas.

7.2.4. Clasificación de espigas.

7.2.4.1. Agrupamiento por plantillas. En la figura 38 se puede observar un ejemplo de la clasificación que se puede obtener utilizando este método (ver sección 6.2.5.1). En esta figura se pueden observar tres unidades que fueron discriminadas manualmente (verde, azul y rojo), se observa que la amplitud es un parámetro que permite separar las distribuciones de las tres unidades, siendo la unidad identificada con el color verde la de mayor amplitud, seguida por la unidad azul y finalmente la unidad roja. Sin embargo, se observa que no es posible diferenciar la actividad de cada unidad de acuerdo a la plantilla en la que se clasificó, pues aparecen espigas de las tres unidades en las cuatro plantillas asignadas.

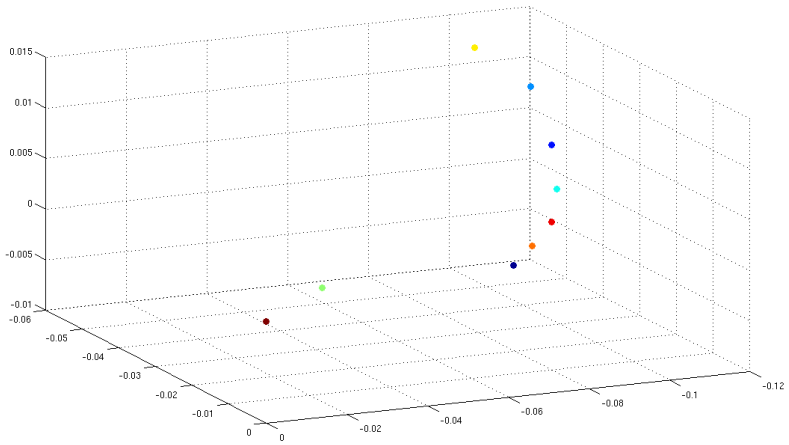


FIGURA 36. Coordenadas tridimensionales de plantillas obtenidas por geometría de distancias. Cada punto representa una plantilla diferente.

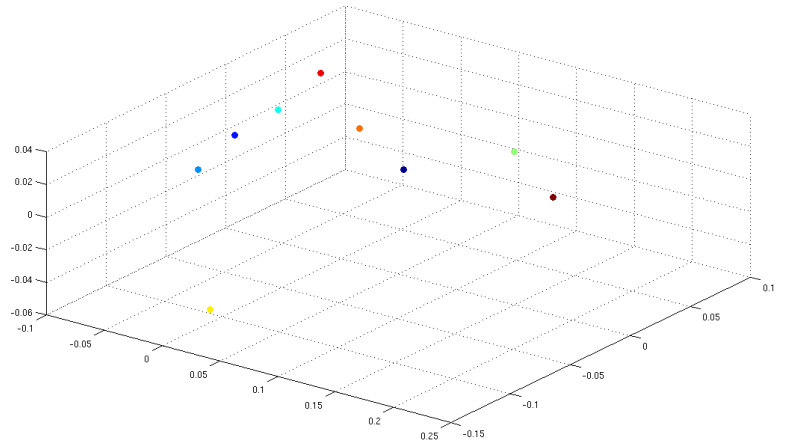


FIGURA 37. Coordenadas tridimensionales de plantillas obtenidas por PCA. Cada punto representa una plantilla diferente.

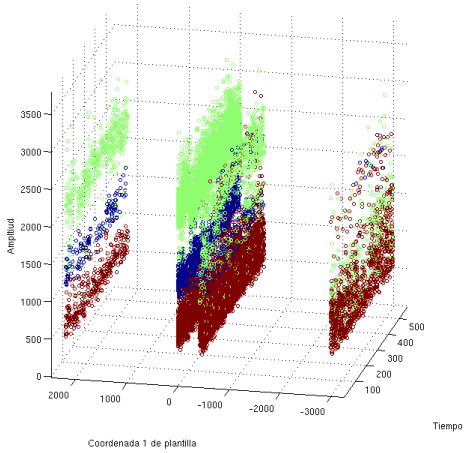


FIGURA 38. Comparación gráfica en el tiempo entre la clasificación manual de tres unidades (verde, azul y rojo) y la clasificación automática por la amplitud de la espiga. Cada superficie de puntos contiene las formas de onda asignadas a una plantilla, utilizando la menor distancia euclidiana entre ambas.

En conclusión, la asignación de una espiga a una plantilla dependiendo de la distancia euclidiana entre ambas, no es un buen parámetro para usar como clasificador. Sin embargo, es posible que la amplitud de la espiga sea de utilidad como una variable para otro algoritmo de clasificación.

7.2.4.2. Expectación-Maximización Gaussiana. En la figura 39 se pueden ver seis grupos de datos aleatorios con una distribución gaussiana, los cuales fueron generados usando la función *randn* en Matlab, con el fin de probar el algoritmo EM (ver sección 2.6.2) con la estimación por grupos BIC (ver sección 2.6.3.1). En la figura también se pueden ver los resultados del agrupamiento. Se puede observar que el algoritmo sobrestima el número de grupos presentes, sin embargo obtiene una asignación de datos muy cercana a la original.

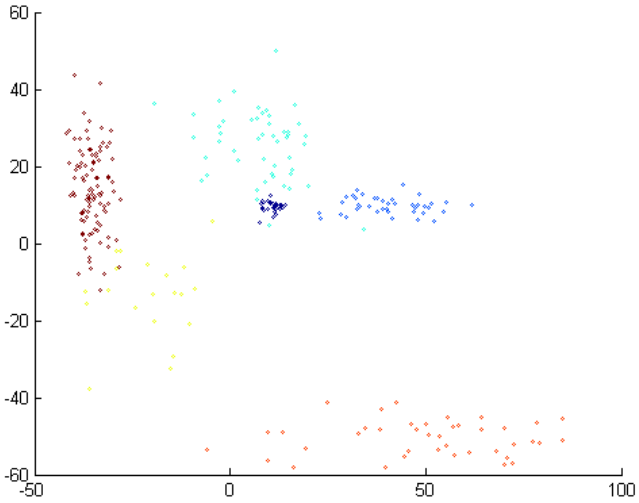
7.2.4.3. Comparación gráfica de los estimadores de grupos presentes en datos artificiales. En la figura 40 se puede ver gráficamente el resultado de aplicar el algoritmo de Expectación-Maximización (ver sección 2.6.2) sobre el PCA de un conjunto de datos reales. En la figura se puede observar que existen tres grupos, sin embargo BIC sobreestima el número de grupos presentes, mientras que EUC lo subestima.

7.3. Medidas de calidad de un clasificador

Las medidas más usadas para identificar la calidad de un clasificador son la proporción de verdaderos positivos (TPR) y la proporción de falsos positivos (FPR). La combinación de ambos se puede utilizar para crear una curva ROC (ver sección 2.3), cuya información se puede condensar en un valor escalar calculando el área bajo la curva (AUC).

En la tabla 9 se pueden ver los resultados de la comparación de la capacidad para representar la calidad de un detector de espigas por ondeletas utilizando la medición del TPR y FPR, contra el MCC (ver sección

a)



b)

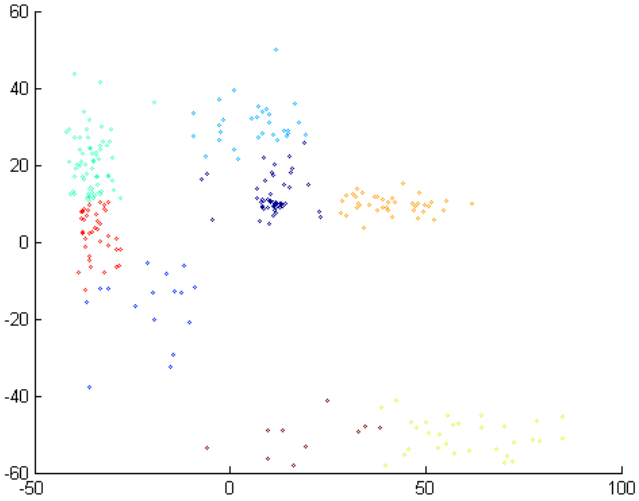


FIGURA 39. Grupos de datos artificiales aleatorios bidimensionales con una distribución gaussiana. a) Distribución original de los datos en grupos. b) Clasificación de los datos en grupos por el algoritmo de EM y BIC. La estimación del número de grupos presentes por el criterio bayesiano de información tiende a sobreestimar el número de grupos, en especial cuando la distancia entre grupos es pequeña.

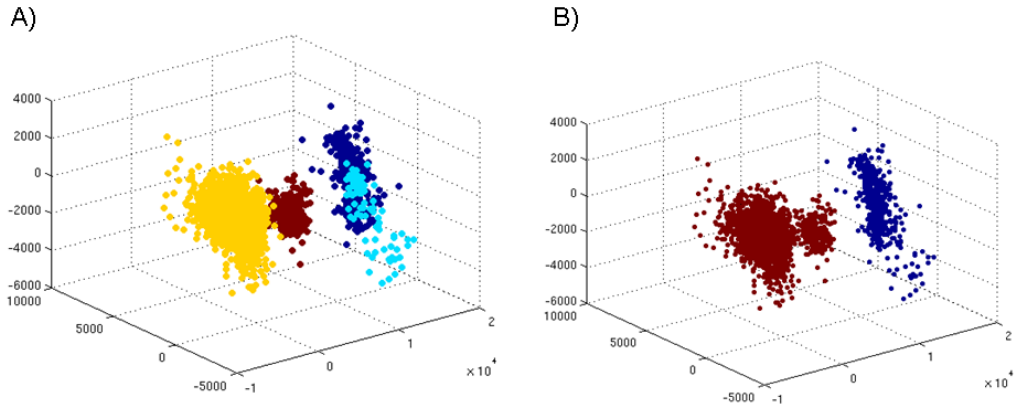


FIGURA 40. Comparación de la clasificación del algoritmo EM con estimación del número de grupos por BIC y EUC.

Familia	Escala	Tipo	Umbral	Aciertos	Falsas alarmas	Rechazos correctos	Fallas	TPR	FPR	MCC
bior3.7	5	D	0.2	52	2191	397,757	0	1	0.0055	0.1518
coif1	3	D	0	52	198,775	201,173	0	1	0.497	0.0114
bior3.7	5	D	0	52	199,450	200,498	0	1	0.4987	0.0114
rbio6.8	4	D	0	57	199,792	200,151	0	1	0.4996	0.0119
coif1	1	D	-0.2	52	378,365	21,583	0	1	0.946	0.0027

TABLA 9. Comparación de valores TPR y FPR contra el MCC.

6.3.2). Esto implica que la AUC no es una medida confiable para medir el desempeño de los algoritmos de discriminación de espigas porque sólo toma en cuenta dos de los cuatro valores de la matriz de confusión y la cantidad de rechazos es varias órdenes de magnitud mayor que la cantidad de aciertos. En cambio el MCC es un parámetro robusto para representar la calidad de un detector de espigas porque toma en cuenta los cuatro parámetros de la matriz de confusión, por lo que las diferencias en el tamaño de las distribuciones no alteran su capacidad de representación.

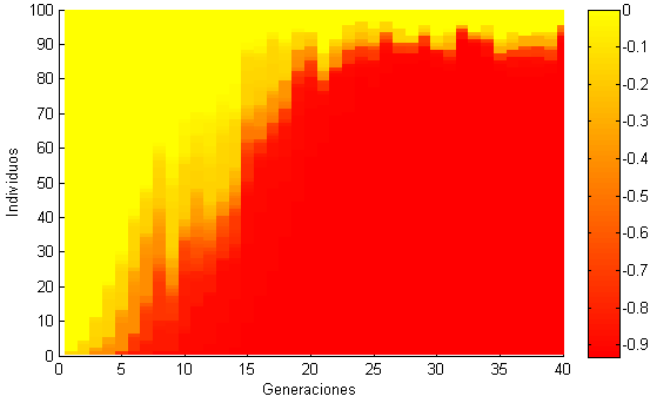


FIGURA 41. Función de aptitud del algoritmo genético para la optimización de la detección de espigas. El algoritmo generó 40 generaciones de soluciones, donde cada generación constó de 100 individuos. La función de aptitud tiene un rango de 0 a -1, donde valores menores representan una mejor clasificación.

7.4. Optimización

Utilizando el algoritmo genético descrito en la sección 6.4.2 se optimizaron los parámetros de las funciones para la detección de espigas (DET_NORM, ver sección 6.6.2). En la figura 41 se puede ver el proceso de convergencia del algoritmo genético. Se puede observar como en las primeras generaciones la mayoría de las soluciones creadas obtuvieron valores de cero en la función de aptitud, muchos de estos individuos en realidad tenían valores de parámetros en alguna función que estaban fuera del rango aceptado por la función, por lo que automáticamente recibían una aptitud de cero. Eventualmente el algoritmo genético empezó a generar soluciones que eran expresables por las funciones del algoritmo de detección y rápidamente sus características se propagaron a los otros individuos de la población.

Los parámetros de las funciones del algoritmo de detección (DET_NORM) encontrados por el algoritmo genético se pueden ver en la tabla 10. Por lo tanto podemos sugerir que un algoritmo genético es un método eficaz para encontrar los parámetros que optimizan las funciones para la detección de espigas en una señal electrofisiológica.

7.5. Señal artificial

En la figura 42 se puede ver una comparación de las señales de entrada de las dos neuronas inmediatas y las señales de salida con diferentes SNR generadas de acuerdo a lo descrito en la sección 6.5. Se generaron tres señales de 300s donde las distancias entre las neuronas y el electrodo se quedan fijas y una señal de 300s donde la posición del electrodo cambia cada 10 s.

7.6. Comparación de los algoritmos

Siguiendo el procedimiento descrito en la sección 6.6 se obtuvieron los resultados que se muestran en la tabla 11. El primer resultado que se puede apreciar, es que el mejor algoritmo de detección es DET_NORM.

Función	
Filtrado por ondeletas	Parámetro: Valor Familia: bior3.7 Coeficientes: A Escala: 4
Detección por ondeletas	Parámetro: Valor Familia: daubechies5 Coeficientes: A Escala: 3 Umbral: 28
Filtrado de formas de onda	Parámetro: Valor Umbral: 134

TABLA 10. Parámetros optimizados de las funciones del algoritmo de detección (DET_NORM) encontrados por el algoritmo genético.

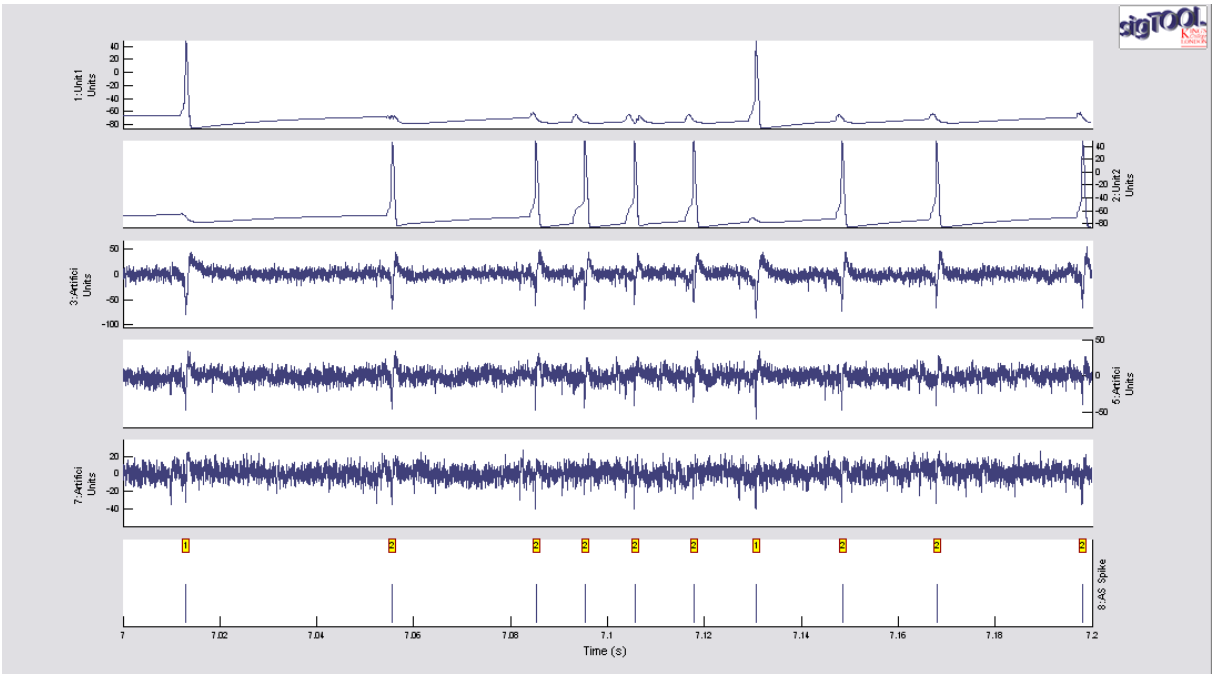


FIGURA 42. Entrada y salida del generador de señales artificiales. En el canal 1 y 2 se muestran las gráficas de los potenciales de membrana de las dos neuronas artificiales inmediatas. Del canal 3 al 5 se muestran las señales de salida con SNR combinados de 2.3, 1.44 y 0.96 respectivamente. En el canal 6 se muestran los tiempos de disparo de las dos unidades inmediatas.

Sin embargo, es una orden de magnitud más lento que su versión rápida DET_RAP. En las señales con un valor alto del coeficiente señal-ruido, la diferencia entre el MCC de ambos algoritmos es prácticamente despreciable. Sin embargo, conforme el coeficiente disminuye, el rango del MCC entre ambos algoritmos aumenta. DET_NORM es más lento debido a que, además de una fase de detección usando la transformada discreta de ondeleta, también lleva a cabo un filtrado de la señal y una comparación de las formas de onda contra una base de datos de plantillas para eliminar artefactos.

En la figura 43 y 44 una comparación gráfica de los resultados de los algoritmos para las señales artificiales y biológicas respectivamente. Al comparar ambas gráficas se puede apreciar que los resultados de casi todos los algoritmos fueron mejores en las señales artificiales que en las biológicas. Se puede obervar que OSort obtuvo los peores resultados en las señales artificiales de todos los algoritmos, de hecho en la señal artificial media marcó a todas las espigas como artefactos y las eliminó, por lo que obtuvo un MCC de cero. Este comportamiento parece ser un error en el código de OSort, dado que no se repitió en ninguna de las otras señales. No obstante, los malos resultados en las señales artificiales, parecen indicar que este algoritmo no es bueno diferenciando espigas cuyas formas de onda sean muy parecidas, como es el caso en estas las señales artificiales. Sin embargo, OSort obtuvo dos de los tres mejores resultados en las señales biológicas. El mejor algoritmo en las señales artificiales fue el EM_CC_LSP, mientras que en las señales biológicas fue EM_CC_BIC.

Los valores del MCC de los algoritmos de clasificación en la señal biológica de calidad alta obtuvieron una calificación menor que en la señal biológica de calidad media, es por esto que se realizó un examen más detallado de la señal biológica de calidad alta. En esta señal la discriminación manual mostró la existencia de dos neuronas: una neurona con una amplitud grande y una con una amplitud pequeña. Se calculó el MCC para el algoritmo de EM_LSP a cada una de las neuronas por separado y se obtuvo un valor de 0.7208 para la unidad con amplitud grande, mientras que el valor de la unidad de amplitud pequeña fue de 0.3595. En la figura 45 un segmento de la señal, junto con la discriminación hecha por el experto.

En la figura 46 se compara el algoritmo de Expectación-Maximización tradicional contra los algoritmos de EM segmentados en el tiempo por agrupación de centroides y por comparación de épocas, para cada método de estimación de grupos. El algoritmo de EM por agrupación de centroides obtiene resultados iguales o mejores que las otras dos versiones. Los métodos de estimación de grupos LSP y EUC obtuvieron resultados similares con todas las versiones del EM. Sin embargo, en promedio, el algoritmo de EM segmentado en el tiempo por agrupación de centroides usando el método del criterio bayesiano de información obtuvo los mejores resultados.

Los resultados de modificar el tamaño de la época que utilizan los algoritmos de EM segmentados en el tiempo se muestran en la figura 47. Todos los algoritmos, excepto el EM_CEP_BIC, tuvieron un mejor desempeño al reducir la época de 45s a 20s. En general los algoritmos segmentados en el tiempo por agrupamiento de centroides obtuvieron un mayor incremento en sus MCC promedios, en comparación con aquellos basados en la comparación de épocas. El algoritmo que obtuvo el mayor incremento con el cambio en la longitud de la época fue el EM_CC_LSP. Es posible que esto se deba a que el EM por agrupación por centroides, se beneficia de tener más épocas, pues esto significa que al final hay más centroides de grupos, lo que se traduce en una mayor cantidad de puntos por agrupar.

	Señal	Artificial Alta	Artificial Media	Artificial Baja	Artificial Movimiento	Promedio Artificial	Biológica Alta (calidad 5)	Biológica Media (calidad 3)	Biológica Baja (calidad 1)	Promedio Biológica	Promedio Total
	MCC										
Detección	DET_NORM	0.9992	0.9585	0.7436	0.9973	0.9247	0.6824	0.7706	0.5197	0.6576	0.8102
	DET_RAP	0.9979	0.8976	0.6886	0.9900	0.8935	0.7323	0.7597	0.5016	0.6645	0.7954
	OSORT	0.9693	0.8785	0.5268	0.9618	0.8341	0.5874	0.7309	0.4617	0.5933	0.7309
DET_NORM + Clasificación	EM_EUC	0.9970	0.7022	0.4802	0.5290	0.6771	0.4556	0.5136	0.2728	0.4140	0.5643
	EM_BIC	0.7508	0.5638	0.3468	0.6202	0.5704	0.4683	0.6535	0.2850	0.4689	0.5269
	EM_LSP	0.9970	0.7022	0.4802	0.5515	0.6827	0.4967	0.5136	0.3289	0.4464	0.5814
	KLUSTA	0.9877	0.6254	0.3559	0.8161	0.6963	0.4491	0.6071	0.3787	0.4783	0.6029
	OSORT	0.7356	0.0000	0.2878	0.7228	0.4366	0.5550	0.5088	0.3923	0.4854	0.4575
	EM_CC_EUC	0.9816	0.7027	0.4802	0.6716	0.7090	0.4440	0.5163	0.2727	0.4110	0.5813
	EM_CC_BIC	0.9589	0.6136	0.5039	0.7102	0.6967	0.4954	0.6009	0.3742	0.4902	0.6082
	EM_CC_LSP	0.9816	0.7105	0.4800	0.7515	0.7309	0.4503	0.4118	0.2830	0.3817	0.5812
	EM_CEP_EUC	0.9970	0.7022	0.4802	0.5651	0.6861	0.5182	0.5136	0.2897	0.4405	0.5809
	EM_CEP_BIC	0.6760	0.6125	0.3551	0.6750	0.5797	0.4157	0.4896	0.2862	0.3972	0.5014
	EM_CEP_LSP	0.9970	0.6125	0.4802	0.5651	0.6637	0.4814	0.5374	0.3114	0.4434	0.5693
	Tiempo de procesamiento (s)										
Detección	DET_NORM	158.5	172.9	217.6	162.0	177.7	108.5	148.8	137.1	131.5	157.9
	DET_RAP	15.6	15.6	17.4	13.7	15.6	14.2	19.3	15.8	16.4	15.9
	OSORT	2.2	2.8	1.9	2.5	2.3	1.3	1.4	1.7	1.5	2.0
DET_NORM + Clasificación	EM_EUC	174.6	187.3	234.3	164.5	190.2	119.7	146.4	138.3	134.8	166.4
	EM_BIC	171.1	199.3	225.3	184.8	195.1	98.3	159.2	146.2	134.6	169.2
	EM_LSP	154.2	190.3	212.5	163.7	180.2	104.9	147.1	140.4	130.8	159.0
	KLUSTA	170.4	200.0	229.4	169.1	192.2	112.7	156.5	153.5	140.9	170.2
	OSORT	6.3	5.2	2.8	6.1	5.1	2.5	3.9	4.4	3.6	4.5
	EM_CC_EUC	152.8	176.1	220.7	161.3	177.7	110.5	138.5	141.8	130.3	157.4
	EM_CC_BIC	173.4	202.6	211.8	181.6	192.3	119.5	137.0	171.0	142.5	171.0
	EM_CC_LSP	172.7	174.3	223.8	186.5	189.3	116.5	138.4	147.1	134.0	165.6
	EM_CEP_EUC	164.2	174.2	215.0	169.7	180.8	113.3	131.7	135.5	126.8	157.7
	EM_CEP_BIC	190.0	182.7	227.0	192.9	198.2	120.8	160.9	138.2	140.0	173.2
	EM_CEP_LSP	178.2	179.7	233.9	175.7	191.9	117.3	158.6	136.1	137.3	168.5

TABLA 11. Resultados de MCC obtenidos y tiempo de procesamiento de los algoritmos probados (los algoritmos de EM segmentados en el tiempo usaron una época de 20s). Se marca en verde el mejor valor y en rojo el peor valor para cada combinación de medida, señal y algoritmo. Cada columna es una señal diferente, que se encuentra dividida en cuatro secciones, dos de valores del MCC (detección y clasificación) y dos del tiempo de procesamiento (detección y clasificación). Las columnas de las señales están agrupadas de acuerdo a su origen: artificiales o biológicas. Al final de cada grupo de columnas hay una columna con el promedio del grupo. La última columna de la tabla es el promedio general de cada algoritmo. El algoritmo con el mejor MCC promedio para la fase de detección fue DET_NORM y para la fase de clasificación fue EM_CC_BIC. Mientras que el algoritmo más rápido tanto en la detección como en la clasificación fue OSort, sin embargo también es el que obtuvo los peores resultados en promedio.

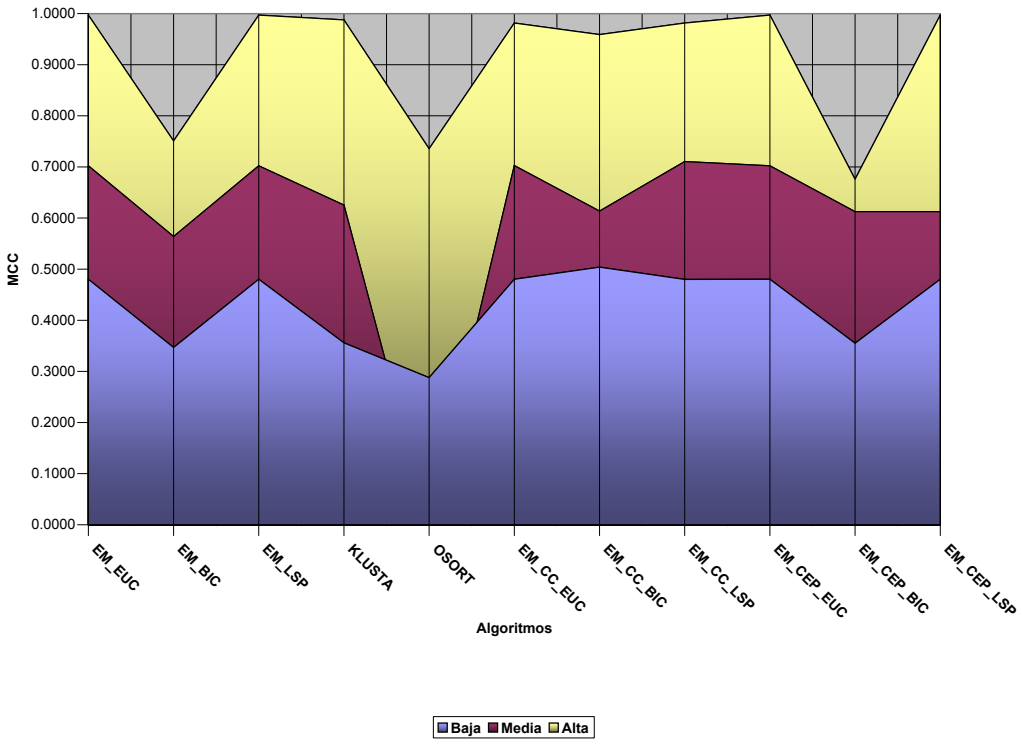


FIGURA 43. Gráfica del MCC obtenido por los resultados de los algoritmos de clasificación en señales artificiales con diferentes niveles de calidad. El mejor algoritmo fue el EM_CC_LSP, mientras que el peor fue OSort.

En la fase de clasificación, el mejor algoritmo en promedio fue el de Expectación-Maximización segmentado en el tiempo por agrupación de centroides usando el criterio bayesiano de información. Cabe mencionar que KlustaKwik demostró ser un programa robusto, pues obtuvo buenos resultados en todas las señales, tanto artificiales como biológicas. Esto lo llevó a colocarse como el algoritmo con el segundo mejor MCC promedio.

7.7. Sort Lab, implementación de un programa para la discriminación de espigas.

Con el fin de integrar todos los algoritmos necesarios para la clasificación de espigas se implementó una serie de módulos en Matlab llamados Sort Lab. Se seleccionó el sistema *sigTOOL* (ver sección *2.8.3*) pues es un programa para el análisis de datos electrofisiológicos, que permite integrar módulos adicionales a su funcionamiento. Se integraron los módulos en una extensión para este sistema.

Las funciones con las que cuenta Sort Lab son las siguientes:

- Creación de programas para la discriminación de espigas (ver figura 48). Esta función muestra los diferentes algoritmos disponibles para los diferentes pasos de la discriminación de espigas (ver sección

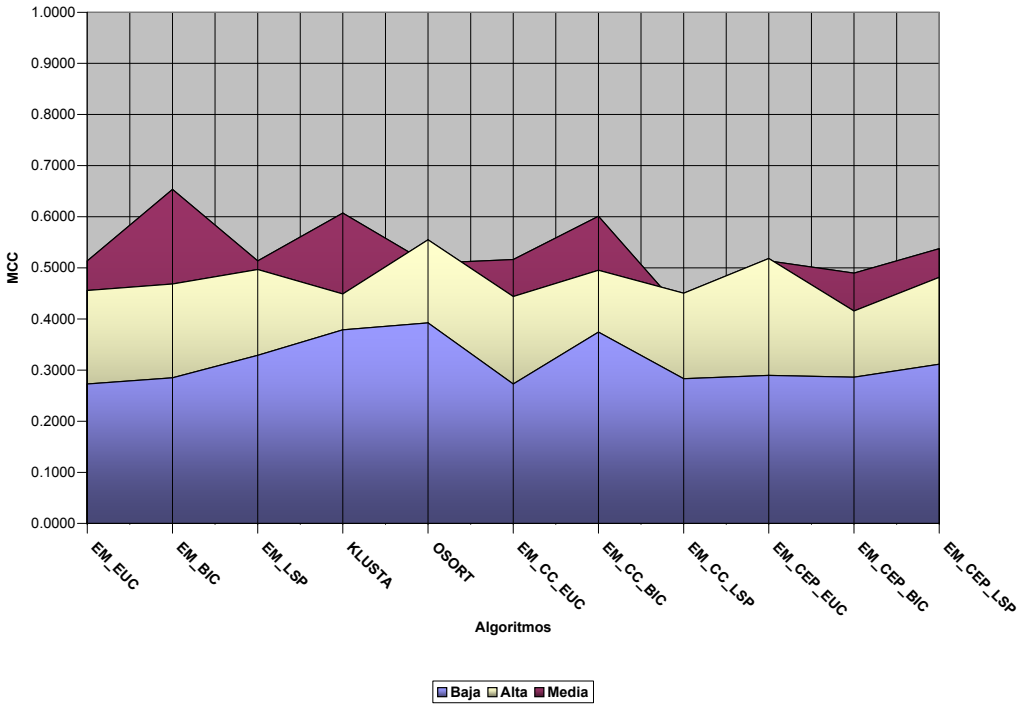


FIGURA 44. Gráfica del MCC obtenido por los resultados de los algoritmos de clasificación en señales biológicas con tres diferentes niveles de calidad. El mejor algoritmo fue el EM_CC_BIC, mientras que el peor fue EM_CC_LSP.

6.2). Este módulo permite crear programas seleccionando las funciones que lo conforman, así como los parámetros de cada función. También incluye la implementación del algoritmo genético para la optimización de los parámetros de las funciones seleccionadas (ver sección 6.4.2). Este algoritmo genético busca los mejores parámetros a partir de un archivo de entrenamiento. Este archivo de entrenamiento puede obtenerse de una señal artificial generada con Sort Lab o de algún registro que se haya discriminado manualmente.

- Crear una señal artificial (ver figura 49 y 50). Este módulo permite cargar datos de las variaciones de voltaje de la membrana celular de una neurona y simular como se recibiría la señal por un electrodo extracelular. El módulo toma en cuenta la distancia de las neuronas al electrodo, así como diferentes parámetros de ruido que permiten generar una señal más realista. Este módulo toma como entrada los archivos de voltajes de membrana de redes neuronales artificiales generados por PyNN (ver sección 6.5).
- Evaluar y comparar la eficiencia y calidad de uno o varios programas (ver figura 51). El comportamiento de los programas generados pueden ser evaluados o comparados entre sí, calculando el MCC del resultado de los algoritmos comparado con un archivo de entrenamiento (ver sección 7.3).

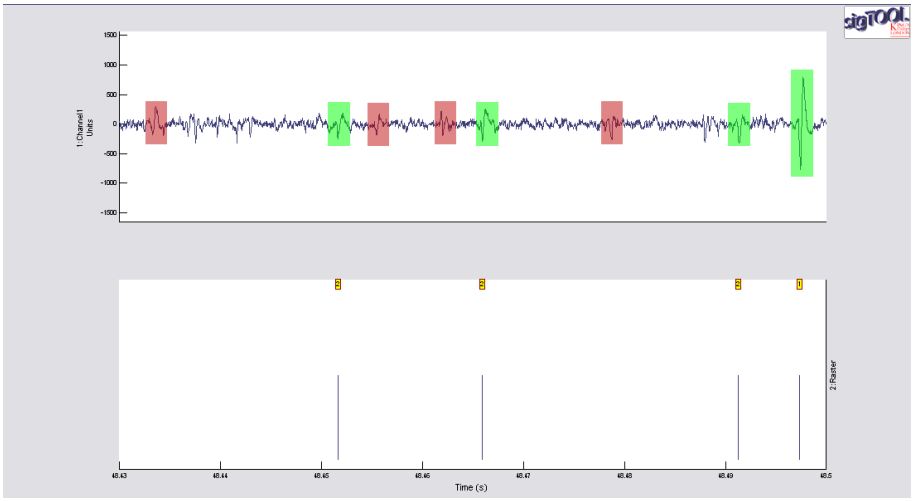


FIGURA 45. Segmento de señal biológica de alta calidad (nivel 5). En el primer canal se muestra la señal electrofisiológica, en verde se marcaron las formas de onda discriminadas como espigas por un expero humano, en rojo se marcaron las formas de onda que posiblemente sean espigas pero que no fueron marcadas como tales por el experto. En el segundo canal se muestra la discriminación hecha por el experto.

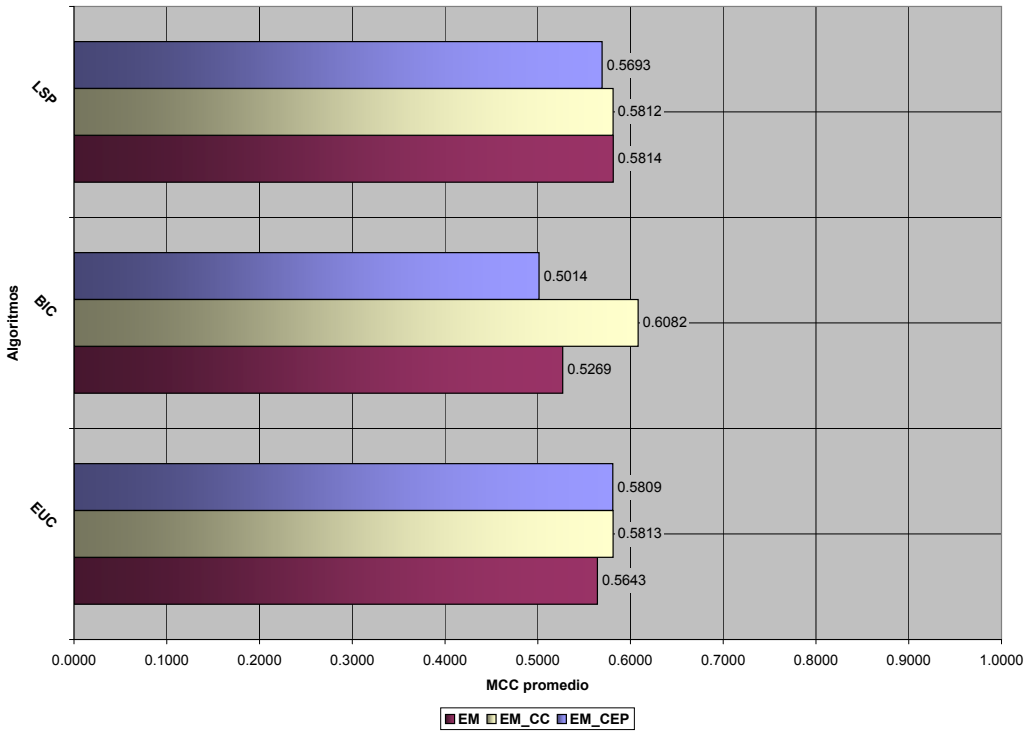


FIGURA 46. MCC promedio del algoritmo EM contra los EM segmentados en el tiempo para BIC, EUC y LSP.

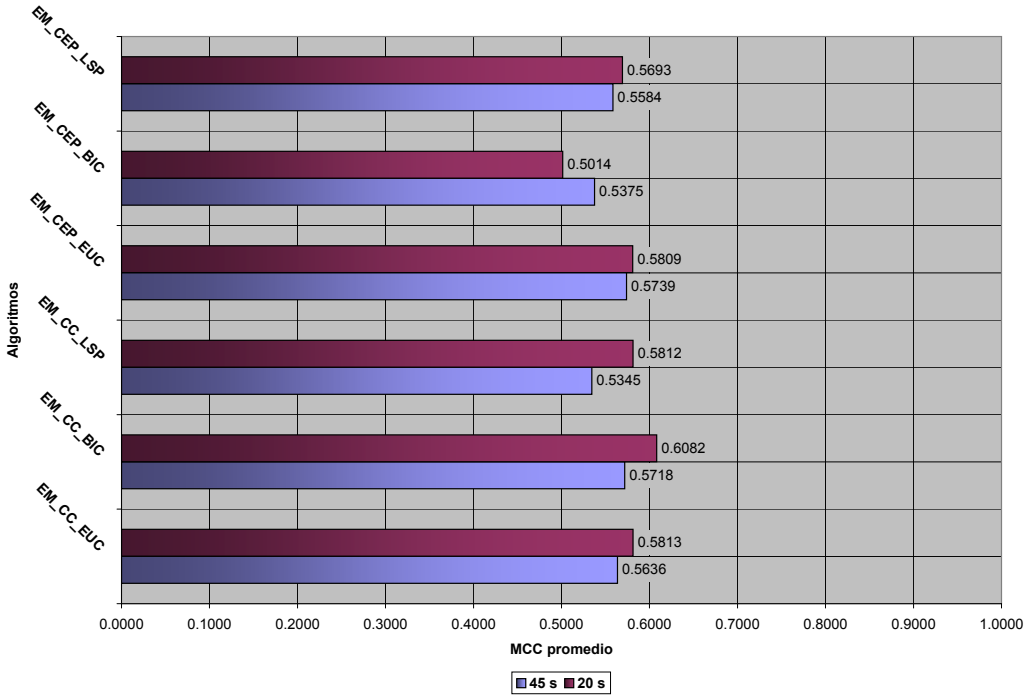


FIGURA 47. MCC promedio obtenidos por los algoritmos de EM segmentados en el tiempo con épocas de 20s y 45s para cada combinación de los diferentes métodos de estimación del número de grupos.

- Correr programas (ver figura 52). Una vez generado un programa que cumpla con los objetivos establecidos, se puede utilizar para discriminar las espigas contenidas en otros archivos de registros electrofisiológicos.

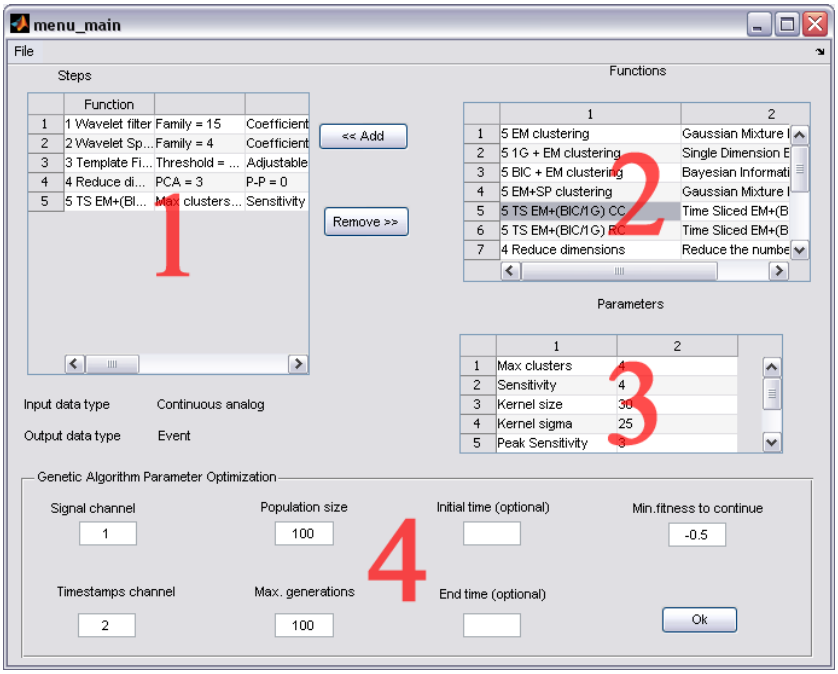


FIGURA 48. Pantalla de creación y edición de programas de Sort Lab. 1) Funciones que componen al programa que se está editando. 2) Catálogo de funciones registradas en Sort Lab. 3) Parámetros para la función seleccionada en el cuadro del catálogo de funciones. 4) Parámetros para el algoritmo genético, este algoritmo busca los parámetros óptimos de las funciones del programa actual tomando como referencia la señal que se encuentre en la pantalla principal de sigTOOL.

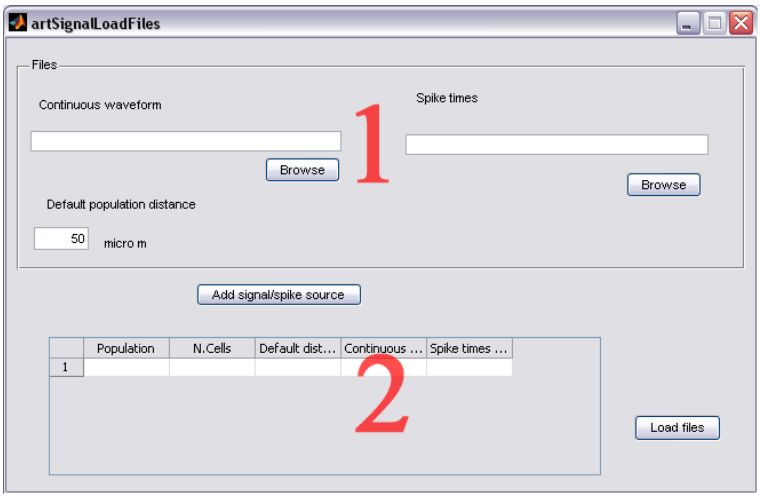


FIGURA 49. Pantalla de carga de archivos de PyNN de Sort Lab. 1) Selección de archivos de PyNN ".v" con los voltajes de membrana de la red neuronal simulada y archivos ".ras" con los tiempos de disparo de las neuronas. 2) Lista de los archivos seleccionados.

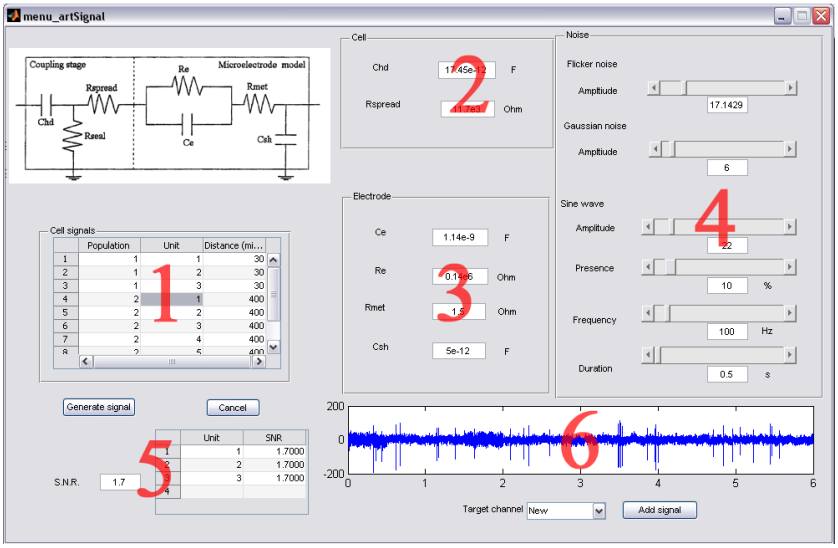


FIGURA 50. Pantalla de creación de una señal artificial de Sort Lab. 1) Listado de unidades registradas para la generación de la señal artificial, con la distancia entre cada unidad y el electrodo. 2) Propiedades de los componentes del lado celular del modelo de la interfase célula-electrodo (ver sección 6.5). 3) Propiedades de los componentes del lado del electrodo del modelo de la interfase célula-electrodo. 4) Parámetros del ruido artificial. 5) SNR de la señal artificial generada y de cada unidad presente en la señal. 6) Gráfica de la señal generada.

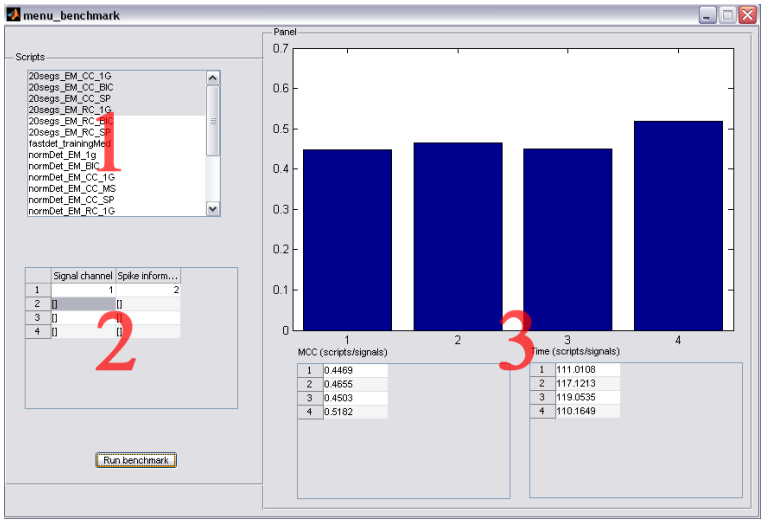


FIGURA 51. Pantalla de comparación de programas de Sort Lab. 1) Lista de selección de programas guardados de Sort Lab a comparar. 2) Selección de los canales del archivo de sigTOOL a usar para realizar las comparaciones. 3) Panel de salida de las comparaciones, compuesta de: gráfica de comparación del MCC, lista de valores del MCC y tiempo de ejecución de cada programa.

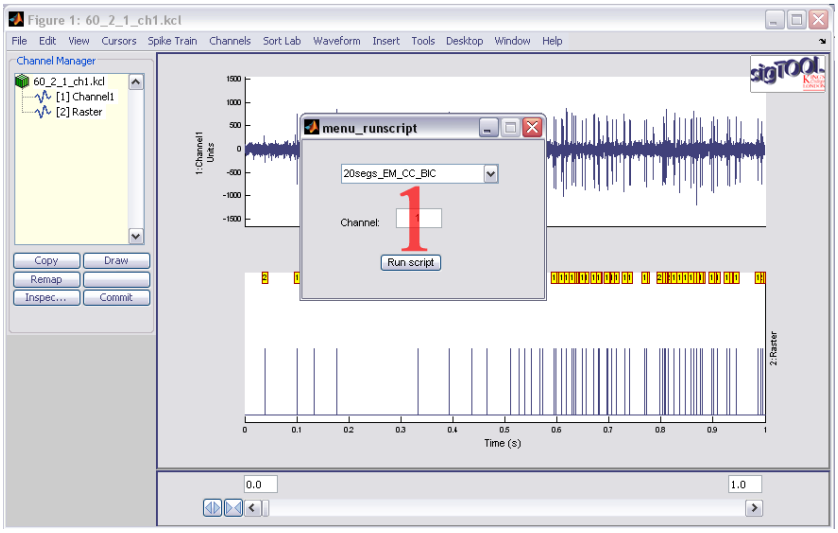


FIGURA 52. Pantalla de ejecución de programas de Sort Lab, sobre la pantalla principal de sigTOOL. 1) Cuadro de selección del programa a correr y selección del canal sobre el cual trabajará el programa.

Capítulo 8

Discusión

La discriminación de espigas es un proceso complejo que se puede dividir en dos etapas principales: detección y clasificación. Las cuáles, a su vez se pueden subdividir en un conjunto de métodos o algoritmos. En este trabajo se probaron diferentes combinaciones de métodos para llevar a cabo la detección y discriminación de espigas. La combinación que obtuvo los mejores resultados en la detección fue:

1. Filtrado de la señal, por un método basado en la transformada de ondeleta, el cual demostró deformar menos la forma de onda de las espigas.
2. Detección de espigas, por un umbral de la transformada estática digital de ondeleta.
3. Filtrado de las formas de onda, al compararlas contra una base de datos de formas de onda.

Esta combinación de algoritmos obtuvo los mejores resultados en la detección. Sin embargo, también es la combinación que requiere un mayor tiempo de cálculo. En las señales donde el coeficiente señal-ruido es grande, utilizar únicamente el paso de detección de espigas, permite obtener resultados parecidos. Sin embargo, los otros pasos permiten limpiar señales ruidosas y eliminar artefactos que pueden ser detectados como espigas. En comparación con OSort, que para efectos de las pruebas utilizó un método de umbral sencillo, la fase de detección por ondeletas permitió obtener mejores resultados. Esto se debe a que la transformada de ondeleta toma en cuenta la forma de la señal y no únicamente la amplitud para la detección de espigas. Los parámetros de la transformada de ondeleta determinan la forma de onda que detectará. Un algoritmo genético, demostró ser un método eficaz para encontrar estos parámetros. Dicho algoritmo es capaz de encontrar la familia y escala de la ondeleta que se debe usar para detectas las espigas en un registro electrofisiológico, tomando como entrada una señal marcada con los eventos que se desean encontrar. Este algoritmo genético podría ser útil para cualquier caso donde se tenga una señal en la que ocurren eventos puntuales que se quieren detectar y que son identificables por su forma de onda. Como podría ser el caso de la búsqueda de paroxismos en señales del EEG para detectar actividad epiléptica.

La clasificación de las espigas detectadas es el paso más complicado de la discriminación de espigas. Existe una gran variedad de algoritmos cuyo objetivo es resolver este problema. Sin embargo, la falta de un método estandarizado de comparación complica la evaluación de los algoritmos. En esta fase el algoritmo que obtuvo los mejores resultados fue el método de Expectación-Maximización segmentado en el tiempo, usando el criterio bayesiano de información para estimar el número de grupos presentes. El coeficiente promedio de MCC de este método ganó de manera marginal a los resultados de KlustaKwik, el cual demostró ser un algoritmo robusto que obtuvo buenos resultados en todas las señales. Este método también está basado en una modificación del algoritmo de EM, pero adicionalmente hace una búsqueda iterativa de los grupos que se puedan unir o separar por sus semejanzas, lo que le confiere su robustez. OSort, un programa realizado

por otro grupo, tuvo un tiempo de procesamiento mucho menor que cualquier otro algoritmo, pero obtuvo resultados pobres en varias de las pruebas. Esta limitación de OSort, parece estar ligada a su método de clasificación por plantillas, donde entre mayor diferencia haya entre los grupos de formas de onda a clasificar, el programa obtendrá mejores resultados.

La combinación de los diferentes métodos necesarios para llevar a cabo la clasificación dificulta el análisis del efecto de cada método en el resultado final. Sin embargo, el método de Expectación-Maximización segmentado en el tiempo por agrupación de centroides, demostró obtener mejores resultados que las otras versiones de los algoritmos de EM. Esto se debe a que este método busca el modelo más estable (de las neuronas involucradas) que se conserva durante todo el registro. Por un lado, esto lo vuelve más robusto a cambios temporales en el registro. Sin embargo, también le impide adaptarse a cambios en el número de neuronas presentes. Dividir el análisis de un registro en épocas, demostró arrojar mejores resultados que el analizar el registro en toda su longitud al mismo tiempo, en especial en las señales que tienen variaciones en la forma de las espigas durante el tiempo. Esto se debe a que los métodos de disminución de dimensiones y representación de las formas de las espigas, obtienen grupos más parecidos a una mezcla gaussiana. Los cambios en las formas de onda provocan que las representación de los grupos se difumine de manera asimétrica, lo que complica encontrar el modelo de mezclas gaussianas que lo generó.

El método de EM para encontrar un modelo de una mezcla gaussiana es sensible al parámetro del número de grupos presentes. Es por esto, que los métodos de estimación del número de grupos juegan un papel determinante en la calidad de la clasificación. Los métodos de estimación unidimensional gaussiana y la localización de centroides en espiral, obtuvieron resultados parecidos en las diferentes versiones de los algoritmos de EM. Sin embargo, el método del criterio bayesiano de información, obtuvo resultados malos en el EM tradicional, pero mejoró de manera importante en el EM segmentado en el tiempo por agrupación de centroides. El BIC es un método que tiende a sobreestimar el número de grupos presentes cuando los grupos no están bien separados. Al utilizar la segmentación en tiempo, se simplificó la representación de los datos, lo que permitió que el BIC mejorara de manera significativa sus estimaciones. La longitud de la época de tiempo de los algoritmos EM segmentados en el tiempo es importante, pues determina el número de espigas que serán analizadas por el algoritmo de EM al mismo tiempo. Una época demasiado larga no ayudará a simplificar los datos, mientras que una época demasiado corta no representará de manera adecuada las poblaciones neuronales que las componen. En este trabajo disminuir la época de 45s a 20s permitió mejorar los resultados de los algoritmos basados en el EM segmentado en el tiempo. Sin embargo, no se estudió cuál sería la época ideal para los registros electrofisiológicos. Esto se podría llevar a cabo utilizando el algoritmo genético que se usó en la fase de detección. Aunque, quedaría por comprobarse si es transferible el tamaño de la época ideal de un registro electrofisiológico a otros registros.

En este trabajo, se hizo gran énfasis en la creación de un método estandarizado para la comparación de algoritmos de discriminación de espigas, que se implementó en el programa Sort Lab. Se seleccionaron y propusieron, cuando hizo falta, medidas que permiten evaluar la calidad de la discriminación y de la señal a ser discriminada. Se encontró que el coeficiente de correlación de Mathews representa mejor la calidad de un discriminador, en comparación con el área bajo la curva de una ROC. Debido a que el MCC toma en cuenta los cuatro valores de la matriz de confusión, mientras que el AUC se ve sesgada cuando se

comparan poblaciones con tamaños muy diferentes. Sort Lab integra estos métodos de medición, además de proveer una plataforma modular para la creación de nuevos algoritmos y programas de discriminación. Esta modularidad, permite que se puedan modificar de manera sencilla los programas de discriminación, para ser adaptados a diferentes conjuntos de datos. Estos conjuntos de datos son determinados por los equipos o por las herramientas informáticas de análisis que son utilizados, ya sea en el mismo o en diferentes laboratorios.

Sort Lab, también incluye un módulo de creación de señales artificiales. El uso de una señal artificial es una herramienta útil para la evaluación de algoritmos de discriminación. La característica más importante de una señal artificial es que se conoce el contenido de la misma sin lugar a dudas. Es de suma importancia que la señal artificial sea capaz de simular los diferentes componentes que contiene la señal real de la manera más fidedigna. El uso de un simulador de redes neuronales, permite probar escenarios donde la morfología neuronal o las interconexiones de la red juegan un papel importante; la simulación de la interfase neurona-electrodo, permite simular diferentes características de los equipos de registro, al poder variar como será alterada la forma de la señal por los dispositivos electrónicos y la configuración de las diferentes fuentes de ruido. Todo lo anterior, se traduce en la posibilidad de generación de una gran gama de señales electrofisiológicas para probar diferentes aspectos de los algoritmos de discriminación. El generador de señales artificiales también puede ser útil para probar otros tipos de algoritmos, como podrían ser algoritmos de análisis de datos o de identificación morfológica de las neuronas involucradas.

En este trabajo, se desarrolló una metodología para la comparación de algoritmos de discriminación, así como la implementación de un programa modular con este fin. Después de realizar una serie de comparaciones, se propuso un algoritmo que logró obtener los mejores resultados en las señales probadas. Sin embargo, por los límites de tiempo impuestos por el programa de maestría, no se pudo estudiar la influencia de todos los parámetros involucrados en la discriminación de espigas. Sería necesario realizar pruebas en una mayor cantidad de señales con diferentes características, para poder concluir cual es la robustez de los algoritmos propuestos.

Capítulo 9

Conclusiones

La clasificación de espigas en registros electrofisiológicos extracelulares es un proceso complejo, donde la variabilidad de las señales biológicas es un factor determinante para la clasificación. Los registros extracelulares tienen un espectro de frecuencias que se mantiene durante todo el registro, existiendo variaciones entre diferentes registros. Además, no existen picos en las bandas de frecuencias que permitan diferenciar de manera consistente un registro que contiene espigas, de aquel que sólo contiene ruido.

La búsqueda del algoritmo ideal para discriminar espigas requiere de herramientas estandarizadas que permitan evaluar su desempeño y establecer comparaciones entre diferentes algoritmos. El Coeficiente de Correlación de Mathews es una medida que representa mejor la calidad de un clasificador, en comparación con el Área Bajo la Curva de una Curva de Características Operantes del Receptor. Mientras que, el Coeficiente Señal-Ruido es una medida útil para representar la calidad de una señal cuyo contenido se conoce a priori.

La discriminación de espigas requiere de una combinación de diferentes algoritmos. El algoritmo de detección que obtuvo los mejores resultados fue el DET_NORM, el cual es una combinación de: -filtrado de la señal por ondeletas -un método de detección basado en la transformada discreta de ondeleta y -un filtrado de las formas de onda para eliminar artefactos.

En este trabajo, se utilizó un algoritmo genético para encontrar el conjunto de parámetros óptimos de los métodos de detección. Se encontró que hay un balance entre la velocidad de ejecución y el desempeño de los métodos de detección. Siendo el método de discriminación DET_NORM el más lento de los métodos probados, pero también el más robusto ante el ruido.

La clasificación de espigas también requiere de una combinación de métodos. El método de Expectación-Maximización segmentado en el tiempo por agrupación de centroides obtuvo mejores resultados que el método de Expectación-Maximización tradicional. La longitud de la época utilizada en el método segmentado en el tiempo, tiene un efecto en la calidad de la clasificación. De igual manera, los métodos de EM son sensibles al parámetro del número de grupos existentes en un conjunto de datos. El criterio bayesiano de información en conjunto con el EM segmentado en el tiempo por agrupación de centroides obtuvo, en promedio, los mejores resultados de clasificación.

Sort Lab demostró ser una plataforma idónea para la comparación de algoritmos de discriminación de espigas. Su estructura modular permite que sea fácilmente extensible y que los programas creados en ella sean rápidamente adaptados a diferentes situaciones. El módulo de creación de señales artificiales permite realizar comparaciones con un conocimiento absoluto del contenido de la señal. Permitiendo que los resultados obtenidos con estas señales sean aplicables a registros extracelulares reales.

Bibliografía

Bagui S y Earp R. 2003. Database design using entity-relationship diagrams. CRC Press.

Baldi P, Brunak S, Chauvin Y, Andersen C y Nielsen H. 2000. Assessing the accuracy of prediction algorithms for classification: an overview. Bioinformatics 16(5), 412–24.

Bar-Hillel A, Spiro A y Stark E. 2006. Spike sorting: Bayesian clustering of non-stationary data. J Neurosci Meth 157, 303–316.

Brown E, Kass R y Mitra P. 2004. Multiple neural spike train data analysis: state-of-the-art and future challenges. Nat Neurosci 7(5), 456–461.

Celeux G y Govaert G. 1992. A classification em algorithm for clustering and two stochastic versions. Comput Stat Data An 14, 315–332.

Chan H, Lin M, Wu T, Lee S, Tsai Y y Chao P. 2008a. Detection of neuronal spikes using an adaptive threshold based on the max-min spread sorting method. J Neurosci Meth 172, 112–121.

Chan H, Wu T, Lee S, Fang S, Chao P y Lin M. 2008b. Classification of neuronal spikes over the reconstructed phase space. J Neurosci Meth 168, 203–211.

Cheever E. 2009. Scam, symbolic circuit analysis in matlab. http://www.swarthmore.edu/NatSci/echeeve1/Ref/mna/MNA6.h

Chelaru M y Jog M. 2005. Spike source localization with tetrodes. J Neurosci Meth 142, 305–315.

Chen S y Gopalakrishnan P. 1998. Clustering via the bayesian information criterion with applications in speech recognition. En Proceedings of the 1998 IEEE International Conference on Acoustics, Speech and Signal Processing.

Chen Y y Gupta M. 2010. Em demystified: An expectation-maximization tutorial. Rep. Téc. UWEETR-2010-0002, Department of Electrical Engineering, University of Washington.

Csicsvari J, Hirase H, Czurkó A, Mamiya A y G B. 1999. Oscillatory coupling of hippocampal pyramidal cells and interneurons in the behaving rat. J Neurosci 19, 274–287.

Cui F, Bibi T y Wu Z. 2004. Mtmm - a matlab toolbox for macromolecular modeling. En Advances in Bioinformatics and its applications, vol. 8, (pp. 319–328).

Davies A y Fearn T. 2004. Back to basics: the principles of principal component analysis. Spectroscopy Europe (pp. 20–23).

Davison A, Brüderle D, Eppler J, Kremkow J, Muller E, Pecevski D, Perrinet L y Yger P. 2009. Pynn: a common interface for neuronal network simulators. Front Neuroinform 2, 1–10.

Delescluse M y Pouzat C. 2006. Efficient spike-sorting of multi-state neurons using inter-spike intervals information. J Neurosci Meth 150, 16–29.

Deshmukh A y Rajagopalan B. 2006. Performance analysis of filtering software using signal detection theory. Decision Support Systems 42(2), 1015 – 1028.

DuBois P. 2003. MySQL Cookbook. O'Reilly.

Fawcett T. 2006. An introduction to roc analysis. Pattern Recognit Lett 27, 861–874.

Gallant T. 2008. Action potential generation. `http://kvhs.nbed.nb.ca/gallant/biology/action_potential_generation.html`.

Gewaltig MO y Diesmann M. 2007. Nest (neural simulation tool). Scholarpedia 2(4), 1430.

Green D y Swets J. 1988. Signal Detection Theory And Psychophysics. Peninsula Publishing.

Harris K. 2000. Klustakwik. http://klustakwik.sourceforge.net/.

Harris K, Heneze D, Csicsvari J, Hirase H y Buzsáki G. 2000. Accuracy of tetrode spike separation as determined by simultaneous intracellular and extracellular measurements. J Neurophysiol 84, 401–414.

Heneze D, Borhegyi Z, Csicsvari J, Mamiya A, Harris K y Buzsáki G. 2000. Intracellular features predicted by extracellular recordings in the hippocampus in vivo. J Neurophysiol 84, 390–400.

Hermle T, Schwarz C y Bogdan M. 2005. Employing ica and som for spike sorting of multielectrode recordings from cns. J Physiol 98, 349–356.

Horton P, Nicol A, Kendrick K y Feng J. 2007. Spike sorting based upon machine learning algorithms (soma). J Neurosci Meth 160, 52–68.

Hulata E, Segev R y Ben-Jacob E. 2002. A method for spike sorting and detection based on wavelet packets and shannon's mutual information. J Neurosci Meth 117, 1–12.

Kandel E, Schwartz J y Jessell T. 2000. Principles of neural science. McGraw-Hill Medical, 4 ed.

Krawetz S y Womble D. 2003. Introduction to bioinformatics: a theoretical and practical approach. Humana Press.

Lewickiy M. 1998. A review of methods for spike sorting: the detection and classification of neural action potentials. Netw Comput Neural Syst 9, 53–78.

Lidierth M. 2009. sigtool: a matlab-based environment for sharing laboratory-developed software to analyze biological signals. J Neurosci Meth 178, 188–196.

Lighthill M. 1958. An Introduction to Fourier Analysis and Generalised Functions (Cambridge Monographs on Mechanics). Cambridge University Press.

Maccione A, Gandolfob M, Massobrio P, Novellino A, Martinoia S y Chiappalone M. 2009. A novel algorithm for precise identification of spikes in extracellularly recorded neuronal signals. J Neurosci Meth 177, 241–249.

Mamlouk A, Sharp H, Menne K, Hofmann U y Martinetz T. 2005. Unsupervised spike sorting with ica and its evaluation using genesis simulations. Neurocomputing 65-66, 275–282.

Martinoia S, Massobrio P, Bove M y Massobrio G. 2004. Cultured neurons coupled to microelectrode arrays: Circuit models, simulations and experimental data. IEEE T BIO-MED ENG 51, 859–864.

McLachlan G y Krishnan T. 1996. The EM Algorithm and Extensions. New York: John Wiley & Sons.

McNicol D. 2005. A primer of signal detection theory. Lawrence Erlbaum Associets, Inc., Publishers.

Phillips C, Parr J y Riskin E. 2007. Signals, Systems, and Transforms, 4/E. Prentice Hall.

Plexon. 2006. Offline Sorter^{TM} Offline Spike Extraction and Sorting Software. Plexon Inc., Neurotechnology Research Systems.

Polikar R. 2001. The wavelet tutorial. Obtenido de http://users.rowan.edu/ polikar/WAVELETS/WTtutorial.html.

Pouzat C, Mazor O y Laurent G. 2002. Using noise signature to optimize spike-sorting and to assess neuronal classification quality. J Neurosci Meth 122, 43–57.

Quian Quiroga R. 2007. Spike sorting. `http://www.scholarpedia.org/article/Spike_sorting`.

Quian Quiroga R y Nadasdy Z. 2004. Unsupervised spike detection and sorting with wavelets and superparamagnetic clustering. Neural Comput 16, 1661–1687.

Rutishauser U, Schuman E y Mamelak A. 2006. Online detection and sorting of extracellularly recorded action potentials in human medial temporal lobe recordings, in vivo. J Neurosci Meth 154, 204–224.

Saavedra V, Fernández T, Harmony T y Castaño V. 2006. Ondeletas en ingeniería. principios y aplicaciones. Ingeniería, Investigación y Tecnología 3, 185.

Shlens J. 2005. A Tutorial on Principal Component Analysis. La Jolla: Salk Institute for Biological Studies.

Smith L, Austin J, Baker S, Borisyuk R, Eglen S, Feng J, Gurney K, Jackson T, Kaiser M, Overton P, Panzeri S, Quian Quiroga R, Schultz S, Sernagor E, Smith V, Smulders T, Stuart L, Whittington M y Ingram C. 2007. The carmen e-science pilot project: Neuroinformatics work packages. En UK e-Science All Hands Meeting 2007, (pp. 591–598), Nottingham, UK.

Snellings A, Anderson D y Aldridge J. 2006. Improved signal and reduced noise in neural recordings from close-spaced electrode arrays using independent component analysis as a preprocessor. J Neurosci Meth 150, 254–264.

Takahashi S, Anzai Y y Sakurai Y. 2003a. Automatic sorting for multi-neuronal activity recorded with tetrodes in the presence of overlapping spikes. J Neurophysiol 89, 2245–2258.

Takahashi S, Anzai Y y Sakurai Y. 2003b. A new approach to spike sorting for multi-neuronal activities recorded with a tetrode - how ica can be practical. Neurosci Res 46, 265–272.

Takahashi S y Sakurai Y. 2005. Real-time and automatic sorting of multi-neuronal activity for sub-millisecond interactions in vivo. Neuroscience 134, 301–315.

Thakur P, Lu H, Hsiao S y Johnson K. 2007. Automated optimal detection and classification of neural action potentials in extra-cellular recordings. J Neurosci Meth 162, 364–376.

Tsai S. 2002. Power Transformer Partial Discharge (PD) Acoustic Signal Detection using Fiber Sensors and Wavelet Analysis, Modeling, and Simulation. Tésis de Maestría, Virginia Tech.

van Drongelen W. 2007. Signal processing for neuroscientists: introduction to the analysis of physiological signals. Academic Press.

van Rossum G y de Boer J. 1991. Interactively testing remote servers using the python programming language. CWI Quarterly 4, 283–303.

Vargas-Irwin C y Donoghue J. 2007. Automated spike sorting using density grid contour clustering and subtractive waveform decomposition. J Neurosci Meth 164, 1–18.

Wood F y Black M. 2007. A nonparametric bayesian alternative to spike sorting. J Neurosci Meth .

Índice de figuras

Índice de tablas

Apéndice A

Base de datos

El uso de tecnologías de administración de información para datos de registro electrofisiológico ha tomado una mayor importancia ante el aumento de la cantidad de información generada por los arreglos multicanales, así como el número de herramientas comerciales y de código libre que han surgido para el análisis de estos datos. La complejidad de la interacción entre las diferentes herramientas incrementa la necesidad de garantizar el acceso a los datos, usando estándares abiertos que permitan la creación de interfases con un repositorio centralizado para el almacenamiento y administración de la información.

Esta necesidad ha llevado al desarrollo de proyectos de neuroinformática en diversos centros alrededor del mundo. Es común encontrar una base de datos centralizada como un repositorio neutral para el intercambio de datos con aplicaciones que pueden estar en distintos lenguajes y plataformas (Smith et al., 2007). El uso de estas tecnologías no debería reservarse a grandes centros con decenas de proyectos, pues los beneficios en la accesibilidad de los datos y los análisis realizados, así como el almacenamiento estructurado de ambos, puede repercutir en el aumento de la productividad científica.

La aplicación de estas ideas al proyecto actual permite la realización de una gran cantidad de análisis de los datos electrofisiológicos, independientemente de las herramientas a utilizar. De igual manera, las bases de datos centralizadas (repositorios de datos) y los sistemas de administración central de código (repositorios de código) se vuelven una herramienta indispensable ante la necesidad de distribuir la carga de trabajo de aquellos análisis que requieran una gran cantidad de tiempo de procesamiento.

A.1. Diagrama Entidad-Relación

Una base de datos es una colección de registros de contenido similar que pueden ser fácilmente accedidos y administrados. Existen dos tipos principales de bases de datos: planas y relacionales. Una base de datos plana es aquella que mantiene los registros de los datos sin una relación estructurada entre ellos, por ejemplo el archivo de una hoja de cálculo. Mientras que una base de datos relacional es aquella que está formada por tablas que contienen datos y las relaciones que tienen las tablas entre si. Estas diferencias se vuelven muy claras cuando se quiere obtener información de la base de datos: mientras que en la base de datos plana se debe buscar secuencialmente cada registro de interés, en la base de datos relacional se puede usar información maestra para obtener el subconjunto de datos buscado (Krawetz y Womble, 2003).

Una de las herramientas más utilizadas para el diseño de bases de datos relacionales son los diagramas entidad-relación (diagramas ER). En estos se conceptualiza la estructura de una base de datos (Bagui y Earp, 2003). Las partes que componen estos diagramas son:

- Tablas, que contienen la estructura y los tipos de datos en los que se almacenará la información.

- Relaciones, indicando la manera en la que se relacionan las tablas entre si.

Un diagrama ER contiene la información suficiente para implementar una base de datos de manera independiente al sistema donde se pondrá en ejecución.

El sistema de base de datos seleccionado para el proyecto fue *MySQL* debido a que es un sistema de administración de base de datos relacional (RDBMS) de código abierto; con millones de instalaciones alrededor del mundo; basado en SQL (Structured Query Language); es relativamente fácil de instalar, usar y administrar; y tiene una gran cantidad de herramientas para su interconexión con otros lenguajes y aplicaciones (DuBois, 2003).

En la figura 53 se puede ver el diagrama de entidad-relación que se utilizó para almacenar los datos de las señales electrofisiológicas y los resultados de los análisis aplicados a ellos. La conexión con Matlab se realizó por medio de *mYm*, que es una interfaz programada en C++ para este propósito. Esta aplicación fue seleccionada por su soporte para datos tipo *BLOB* (Binary Large Object), pues es un requisito de acuerdo al diseño de base de datos que se utilizó.

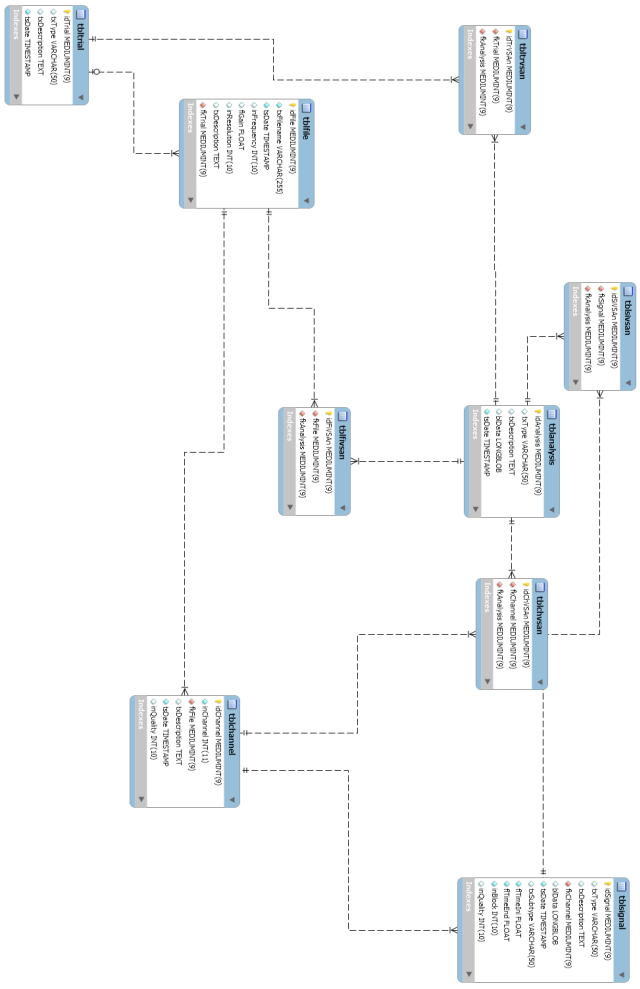


FIGURA 53. Diagrama de entidad-relación de la base de datos relacional para almacenaje y análisis de señales.

Apéndice B

Cómputo distribuido

El tiempo de procesamiento necesario para los algoritmos de análisis de datos es proporcional al tamaño del conjunto de datos por analizar y a la complejidad de los algoritmos usados. Un archivo que contiene 5 minutos de registros extracelulares esta compuesto por millones de muestras. Debido a la longitud de los datos por analizar y la complejidad de los algoritmos usados, los tiempos de procesamiento pueden ser de varios días. Es por esto que surgió la necesidad de distribuir las tareas que componen un algoritmo entre varias computadoras con el fin de reducir el tiempo necesario para llevar a cabo un análisis.

El sistema para distribución de tareas se basa en un modelo productor-consumidor, donde el productor es el servidor generador de tareas y los consumidores son las computadoras clientes que realizan las tareas. La sincronización del sistema se logra gracias a una base de datos relacional *MySQL*, que por medio de transacciones garantiza la atomicidad de la asignación de las tareas. Este último punto es importante para evitar que se asigne a dos o más clientes la misma tarea.

El sistema de cómputo distribuido está basado en tres módulos:

1. Base de datos para distribución y control de tareas (ver figura 54). Esta base de datos fue creada en *MySQL* 5.1 sobre una plataforma *Windows XP*. Es la encargada de mantener en orden la asignación de tareas, así como el estado de las mismas.
2. Servidor repositorio de archivos. Se utilizó un servidor para la administración de códigos llamado *VisualSVN Server*, el cual permite llevar un control de las versiones de los archivos de código y su acceso desde los clientes. En las computadoras clientes se utilizó el programa *TortoiseSVN* para poder acceder al repositorio de código y realizar la sincronización de los archivos. Las tareas asignadas a los clientes son funciones de Matlab descritas en estos archivos.
3. Funciones para clientes de cómputo distribuido. Se realizaron un conjunto de funciones para Matlab encargadas de acceder a la base de datos de tareas y ejecutar las tareas asignadas de forma secuencial.

El uso de los módulos mencionados, permitió distribuir las tareas entre los clientes de manera automática. Las ventajas de este sistema son: es relativamente sencillo de mantener, se pueden agregar o quitar clientes dinámicamente, los datos de trabajo pueden residir en bases de datos locales en diferentes sitios. Entre sus limitantes están: sólo debe ser usado con problemas que puedan dividirse en tareas que no tengan relación entre si, la división de las tareas debe indicarse manualmente al sistema.

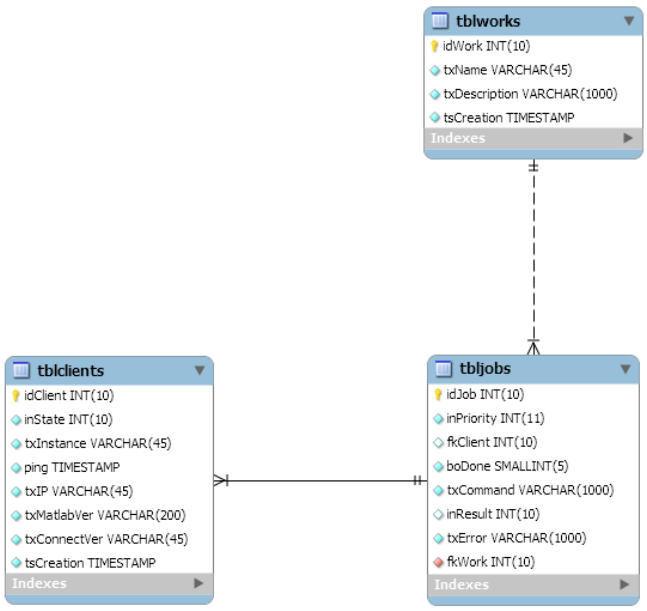


FIGURA 54. Diagrama de entidad-relación de la base de datos relacional para el sistema de control de cómputo distribuido.