Universidad de Valladolid

FACULTAD DE CIENCIAS

GRADO EN ESTADÍSTICA

Análisis de señales neuronales del proyecto Blue Brain con modelos oscilatorios

---

Autor
Marta Fernández Poyatos

Tutores:

Alejandro Rodríguez Collado
Cristina Rueda Sabater

Agradecimientos

Al grupo de Inferencia Estadística con Restricciones, en especial a mis tutores Cristina y Alejandro. Gracias por vuestra constante dedicación y por transmitirme vuestra pasión por el trabajo que hacéis día tras día. Gracias por todo el conocimiento que me habéis transmitido en estos meses.

A los profesores del Grado en Estadística que me han acompañado estos últimos cinco años. De todos y cada uno de ellos me llevo un gran aprendizaje.

A mis amigos y compañeros de clase por hacer de estos años una etapa preciosa de la que me llevo grandes recuerdos.

Por último y más importante, a mis padres y a mi pareja. A mis padres por su apoyo incondicional y todos los valores que me han inculcado desde pequeña. Y a mi pareja por creer en mi e inspirarme siempre a seguir adelante.

Resumen

El cerebro es la estructura más compleja y a la vez más desconocida de nuestro cuerpo. Este desconocimiento hace que aún hoy haya muchos interrogantes acerca de las causas de múltiples trastornos mentales como el Parkinson o la esquizofrenia.

Existen diferentes técnicas para alcanzar un conocimiento más profundo del cerebro, entre las que se encuentra el estudio de su actividad neuronal mediante señales oscilatorias de diferencia de potencial. Estas señales se han analizado tradicionalmente con diferentes modelos, como el modelo Cosinor o el modelo de Fourier. Sin embargo, estos modelos caracterizan las señales de una forma bastante imprecisa y dan como resultados unos parámetros no interpretables.

El objetivo principal de este trabajo es el análisis de las señales de la base de datos del proyecto Blue Brain con diferentes modelos oscilatorios, entre ellos, el modelo Frequency Modulated Möbius (FMM). Es un modelo paramétrico que consigue unos mejores resultados que otros modelos oscilatorios en el ajuste de señales electrofisiológicas, además de proporcionar parámetros interpretables. Estos revelan diferencias existentes entre las señales de las distintas clases de neuronas. Por otro lado, incorporados como características a procedimientos de aprendizaje, permitirán discriminar los diferentes tipos neuronales, lo que será clave para el tratamiento y diagnóstico de múltiples enfermedades y trastornos mentales.

Palabras clave: señal oscilatoria, FMM, neuronas, genes.

Abstract

The brain is the most complex and the most unknown structure of our body. This lack of knowledge makes that even today there are still many questions about the causes of many mental disorders such as Parkinson's disease or schizophrenia.

There are different techniques for getting a deeper understanding of the brain, including the study of the oscillatory signals of their neuronal activity through oscillatory signals of potential difference. These signals have traditionally been analysed with different models, such as the Cosinor model or the Fourier model. However, these models characterise the signals in a rather imprecise way and give us uninterpretable parameters.

The main objective of this work is the analysis of the signals of the Blue Brain project database with different oscillatory models, among them, the Frequency Modulated Möbius model (FMM). It is a parametric model that achieves better results than other oscillatory models in fitting electrophysiological signals, as well as providing interpretable parameters. These reveal differences between the signals of different classes of neurons. On the other hand, incorporated as features in learning procedures, they make possible to discriminate between different neuronal types, which will be essential for the treatment and diagnosis of multiple diseases and mental disorders.

Key words: scillatory signal, MMF, neurons, genes.

Índice general

Índice de figuras

Índice de tablas

Capítulo 1

Introducción

Actualmente aún hay un gran desconocimiento del funcionamiento del cerebro. Uno de los métodos que nos permiten comprenderlo mejor es la neurociencia electrofisiológica, es decir, el estudio de la actividad eléctrica de las neuronas. Estas señales eléctricas se recogen mediante electrodos, obteniéndose señales que se caracterizan por ser oscilatorias.

Existen múltiples propuestas para el estudio de las señales electrofisiológicas, las más utilizadas son los modelos Cosinor y de Fourier, sin embargo, ofrecen resultados poco precisos y sus parámetros no son interpretables. Recientemente en el grupo de investigación de Inferencia Estadística con Restricciones de la Universidad de Valladolid, se ha desarrollado el modelo FMM (Frequency Modulated Möbius) [16]. Este modelo es más flexible que los armónicos de los modelos Cosinor y de Fourier. Además, es un modelo paramétrico, cuyos parámetros son interpretables, permitiendo cuantificar características de la señal de forma directa con un número reducido de parámetros. El modelo de Fourier, para caracterizar formas muy asimétricas, requiere de muchas componentes, mientras que el modelo FMM tan solo una. El uso de muchas componentes en el modelo de Fourier puede dar lugar a problemas de sobreajuste. El modelo FMM no solo está diseñado para el estudio de señales electrofisiológicas y la caracterización de la actividad eléctrica de las neuronas, sino también otro tipo de señales, como los electrocardiogramas, permitiendo identificar enfermedades cardiovasculares. También tiene aplicaciones en el campo de la cronobiología y de la astrofísica. Además, en este modelo, cada onda tiene una correspondencia directa con un proceso fisiológico.

En este trabajo nos proponemos estudiar los conceptos fundamentales de la neurociencia electrofisiológica y la caracterización molecular de las células, así como los diferentes modelos para el estudio de las señales oscilatorias: el modelo Cosinor, el modelo de Fourier, y en especial, el modelo FMM.

Se estudiará la base de datos del proyecto Blue Brain, que contiene registros eléctricos de las neuronas de una rata de la especie *Rattus norvegicus*, tomados mediante el software IGOR Pro. Se

emplea una librería específica de R para la lectura de los datos tomados mediante este software, además de seleccionarse las señales estimuladas eléctricamente con un único *peak*. Estas señales se recortarán en un mismo intervalo de tiempo y se eliminarán los artefactos provocados por ruido en la medición. Una vez que las señales hayan sido procesadas, se analizarán mediante el modelo FMM y los parámetros resultantes del ajuste se incorporarán como características a diferentes procedimientos de aprendizaje supervisado, en busca de una discriminación de las señales según su tipo genético.

Capítulo 2

Contexto

En este capítulo se procederá a introducir los conceptos biológicos necesarios para entender el problema que se va a tratar: neurociencia electrofisiológica y caracterización molecular de las células.

2.1. Neurociencia electrofisiológica

La **neurociencia** es una ciencia que estudia aspectos celulares conductuales, computacionales, funcionales, evolutivos, moleculares y médicos del sistema nervioso. Abarca múltiples disciplinas, como la fisiología, biología molecular, anatomía, citología, biología del desarrollo, informática y modelado matemático, entre otras.

La **electrofisiología neuronal** es una rama de la neurociencia que explora la actividad eléctrica de las neuronas vivas e investiga los procesos moleculares que rigen su señalización. Las neuronas se comunican usando señales eléctricas y químicas. Las técnicas utilizadas en la electrofisiología miden la actividad eléctrica de estas señales y buscan descifrar sus mensajes intercelulares e intracelulares.

Como podemos apreciar en la figura 2.1, existen tres posibles técnicas electrofisiológicas para medir las señales enviadas por las neuronas, definidas según dónde se coloca el instrumento de medición en la neurona:

- **Extracelular**: el electrodo se sitúa justo fuera de la neurona de interés.
- **Intracelular**: el electrodo se inserta dentro de la neurona.

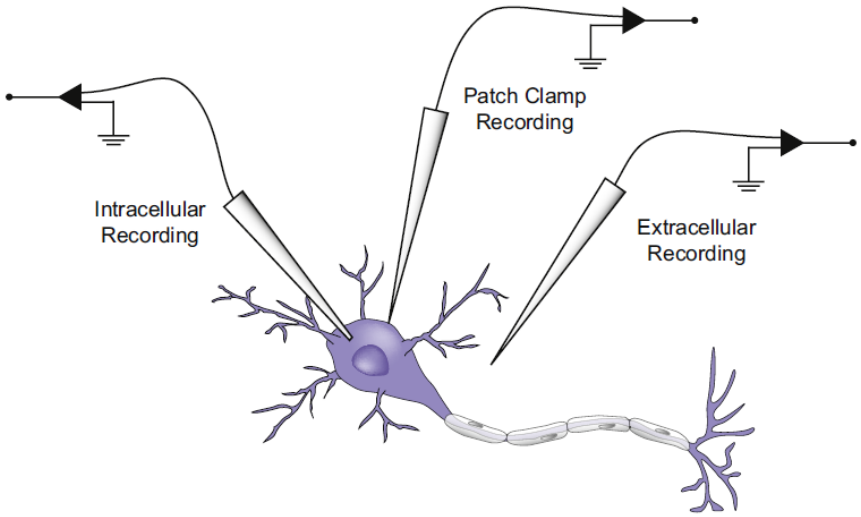


Figura 2.1: Tipos de medición [1]

- **Técnicas "patch clamp"**: el electrodo se sitúa pegado a la membrana de la neurona.

Estas técnicas de medición se utilizan para examinar las propiedades eléctricas de las neuronas *in vitro* (fuera de un organismo vivo) e *in vivo* (dentro de un organismo vivo).

2.1.1. Propiedades eléctricas de las neuronas

La actividad eléctrica de una neurona se basa en la concentración del gradiente relativo y en los gradientes electrostáticos dentro de la célula, así como en los tipos de canales iónicos que están presentes en las neuronas.

Una propiedad fundamental de las neuronas y que será básica para el desarrollo de este proyecto es la **diferencia de potencial**, es decir, el potencial eléctrico producido por la diferencia de carga entre la parte intracelular y extracelular de la membrana de la neurona. El potencial en reposo de las neuronas suele rondar los -70 mV aproximadamente.

Las neuronas se comunican produciendo cambios en el potencial de la membrana de otras neuronas. A estos cambios podemos denominarlos **AP** (Action Potential), *spike* o *peak*. Los APs son unidades de información entre las neuronas, y su número y forma determinan el perfil morfológico, funcional, y genético de la célula. El AP mide la fluctuación del potencial de una neurona, es decir, la diferencia entre el potencial eléctrico dentro y fuera de la neurona debido a estímulos externos. Cuando un AP ocurre, podemos observar tres etapas bien distinguidas:

- **Depolarización**: Es la primera etapa, se produce un incremento repentino del potencial.
- **Repolarización**: Es la segunda etapa, el potencial decrece de forma súbita hasta el potencia el reposo.
- **Hiperpolarización**: Es la última etapa, el descenso del potencial continúa para posteriormente recuperar paulatinamente el potencial en reposo.

Tras estas tres etapas se pasa a un **periodo de refracción**, en el que el potencial se mantiene en reposo.

La forma típica de un AP y las tres etapas explicadas previamente, las podemos observar en la figura 2.2.

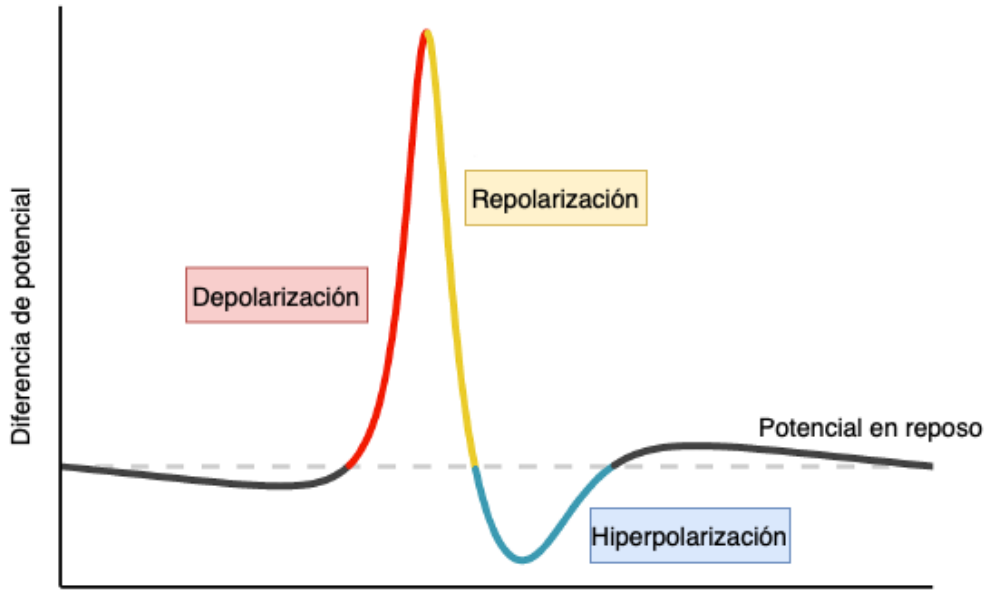


Figura 2.2: Fases del AP de una neurona [15]

Tras producirse la etapa de depolarización (y por tanto, un *peak*) debido a un potencial positivo elevado, se pasa a una **etapa refractaria** en la que no puede producirse otro *peak*. Si se produce un aumento del potencial, pero no lo suficiente, entonces no se desencadenará el *peak*.

2.2. Caracterización molecular de una célula

Las neuronas tienen diferentes propiedades a nivel morfológico, molecular, electrofisiológico y de características funcionales [2]. Para poder llegar a comprender la estructura y el funcionamiento del cerebro y de las conexiones neuronales, es necesario agrupar las neuronas en diferentes tipos y posteriormente, analizar las características de estos. Sin embargo, resulta todo un reto llegar a realizar esta clasificación. El primer intento de clasificación morfológica de las neuronas fue alrededor de hace 100 años, realizado por Ramón y Cajal en su libro [3]. Múltiples propuestas han sucedido a la de Ramón y Cajal a lo largo de los años debido a que la clasificación de las neuronas resulta ser indispensable para llegar a comprender el funcionamiento del cerebro. Sin embargo, hasta hace pocos años ha habido grandes dificultades técnicas, ya que no se disponían de los métodos necesarios para el conocimiento de las células.

Resulta muy complejo agrupar las células en diferentes tipos, debido a que cada neurona es diferente, sin embargo se ha conseguido llegar a tres tipos de clasificaciones de las neuronas, de acuerdo con sus características morfológicas, fisiológicas y moleculares. La **morfología** de las células neuronales estudia la forma de las dendritas, axones o patrones de ramificación, entre otras características. La **fisiología** estudia el potencial en reposo de las neuronas, las propiedades biofísicas y la frecuencia de activación. Entre las **propiedades moleculares** encontramos la composición proteica y la composición de ARNm.

También podemos utilizar como criterio de clasificación de las neuronas, las **propiedades de conectividad**, aunque son difíciles de medir.

En nuestro caso particular, estudiaremos las neuronas de la corteza cerebral, que contiene múltiples áreas sensoriales y motoras. La mayoría de los estudios sobre tipos neuronales se han centrado en estudiar la corteza somatosensorial y visual de las ratas.

Las neuronas pueden dividirse en dos clases fundamentales: neuronas **glutamatérgicas** (principalmente excitatorias) y neuronas **GABAérgicas** (principalmente inhibitorias), ambas clases con múltiples subclases.

Las neuronas **glutamatérgicas** son específicas de un área concreta del cerebro y son las encargadas de propagar las señales dentro y entre las células [7]. Desarrollan un papel fundamental en procesos de aprendizaje, cognición y en la memoria. Una transmisión irregular de las señales por parte de estas neuronas puede provocar diferentes afecciones neurológicas.

Las neuronas **GABAérgicas** (principalmente **excitatorias**), no son específicas de un área concreta del cerebro y controlan el flujo de las señales. A pesar de representar tan solo entre un 10 % y un 15 % de las neuronas de la corteza cerebral, se cree que este tipo de neuronas tiene un papel fundamental en muchas de las funciones de la corteza cerebral, como por ejemplo, el control de los circuitos corticales y el mantenimiento del equilibro excitatorio e inhibitorio necesario para

la transferencia de información [8]. Un mal funcionamiento de estas neuronas se ha llegado a asociar con ciertos tipos severos de epilepsia, esquizofrenia, trastornos de ansiedad y autismo.

Dependiendo del autor, podemos encontrar diferentes formas de agrupar las neuronas, como en [4], [5] y [8]. Según [8], existen tres grupos fundamentales no solapados de marcadores moleculares, que representan casi el 100 % de las neuronas GABAérgicas de la corteza somatosensorial primaria:

- **Pvalb**(*Parvalbumin*). Está relacionado con el déficit de memoria de trabajo (memoria a corto plazo), la flexibilidad cognitiva y la sociabilidad en animales con enfermedades psiquiátricas. En enfermedades como la esquizofrenia, un tratamiento de modulación de este tipo de neuronas, puede tener un mejor desempeño que el uso de antipsicóticos [9].

- **SOM** (*Neuropeptide somatostatin*). Ciertos trastornos neurodegenerativos y neuropsiquiátricos, como el Alzheimer, el Parkinson, la enfermedad de Huntington, trastornos depresivos, trastorno bipolar y la esquizofrenia, están relacionados con una disminución de neuronas de este tipo [10].

- **5HT3aR** (*5-hydroxytryptamine receptor 3A*): Dentro de ese grupo podemos encontrar dos subgrupos, según si se expresa el gen *vasoactive intestinal polypeptide* (**VIP**) o no. El subgrupo de neuronas VIP está asociado a los ritmos cicardianos de las neuronas, tanto la ausencia de estas neuronas VIP como una cantidad excesiva, puede provocar una asincronía en los ritmos cicardianos celulares [11].

Además de PV, SOM, 5HT3aR y VIP, existen otros marcadores moleculares utilizados frecuentemente para estudiar y etiquetar las neuronas GABAérgicas. A diferencia de los anteriores marcadores, estos pueden expresarse superpuestos: **CB** o **Calb1** (*Calbindin*), **CR** o **Calb2** (*Calretinin*), **Cck** (*Cholecystokinin*), **NPY** (*Neuropeptide Y*). También, existe un pequeño número de células que no pueden ser agrupadas ya que no expresan ningún marcador molecular.

Es un gran reto crear una taxonomía neuronal completa, debido a la gran heterogeneidad existente entre las neuronas, a nivel genético, electrofisiológico y morfológico. En la figura 2.3 se muestra una de las taxonomías mas complejas existentes hoy en día. Se observa que las neuronas GABAérgicas son las pertenecientes a la rama de la izquierda del árbol, correspondientes con tonos más rosados, mientras que las neuronas glutamatérgicas son las que pertenecen a la rama central del árbol, correspondiéndose con los colores más verdosos y azules. Por último, las células no neuronales se corresponden con la rama derecha del árbol, con tonos grisáceos.

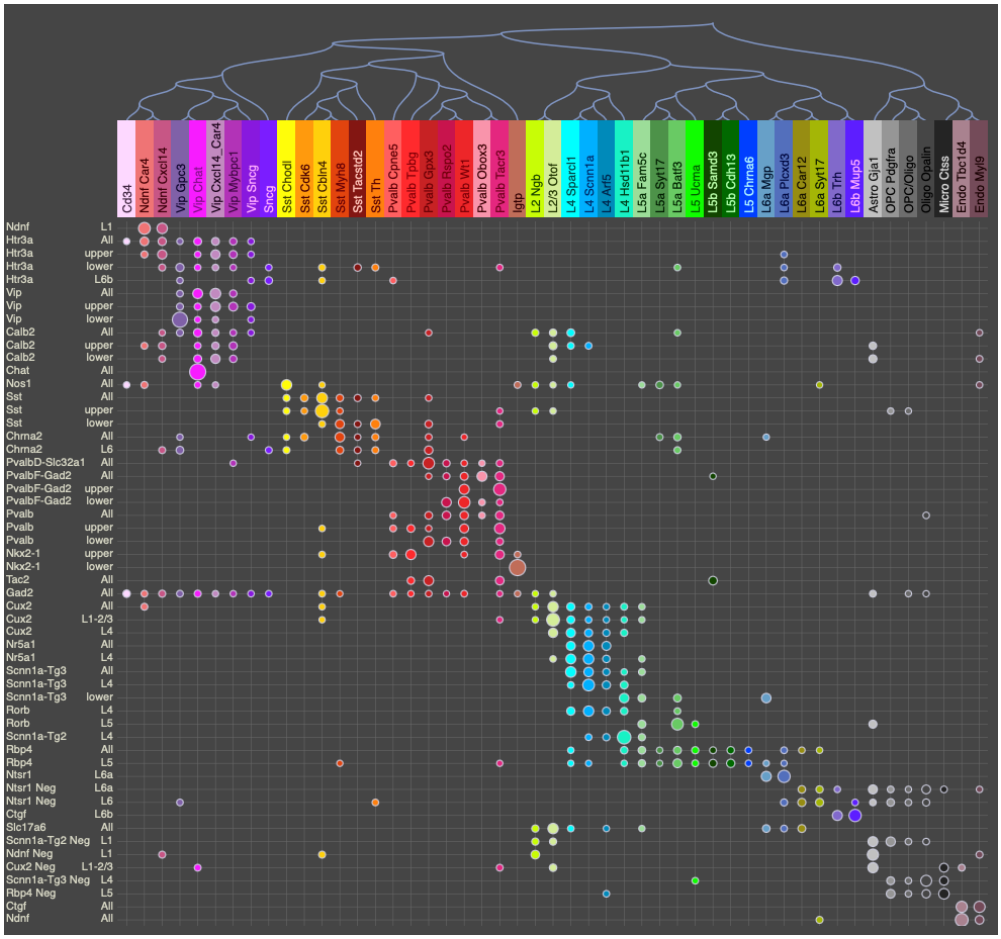


Figura 2.3: Taxonomía neuronal Allen Brain Atlas [6]

Capítulo 3

Métodos

En este capítulo se desarrollarán diferentes modelos para el análisis de señales oscilatorias, además de hacer un estudio comparativo entre ellos. Por otro lado, se introducirán los conceptos teóricos de las técnicas de aprendizaje supervisado que posteriormente se van a utilizar para el análisis de los resultados obtenidos tras el ajuste de señales neuronales mediante modelos oscilatorios.

3.1. Modelos oscilatorios

Existen diferentes modelos para el análisis de señales oscilatorias. Entre los más frecuentes se encuentran el modelo Cosinor y el modelo de Fourier, que se explican a continuación. Posteriormente, se explica un nuevo modelo, el modelo FMM, mucho más flexible que los anteriores y con múltiples aplicaciones en el ámbito de la biología.

Suponemos que $X(t_i)$, $t_1 < \ldots < t_n \in [0, 2\pi]$ es un vector de observaciones en el tiempo. $X(t_i)$ representa la diferencia de potencial en la membrana de la neurona en cada instante de tiempo observado $(t_i, i = 1, \ldots, n)$. Podemos definir estos modelos oscilatorios de la siguiente forma:

$$X(t_i) = \mu(t_i) + \varepsilon(t_i) \tag{3.1}$$

Siendo $\mu(t_i)$ la señal y $\varepsilon(t_i)$ el ruido (aleatorio e independiente de la señal), con $i = 1, \ldots, n$.

3.1.1. Modelo Cosinor (COS)

El modelo cosinor es el modelo más sencillo para el análisis de señales oscilatorias. Se utiliza para describir señales simétricas (sinusoidales).

Podemos definir el modelo Cosinor de la siguiente forma:

$$X(t_i) = \mu(t_i) + \varepsilon(t_i) = M + A \cdot \cos(t_i + \varphi) + \varepsilon(t_i); i = 1, \ldots, n \tag{3.2}$$

1. $M \in \mathbb{R}$, $A \in \mathbb{R}^+$
2. $\varphi \in [0, 2\pi]$
3. $(\varepsilon(t_1), \ldots, \varepsilon(t_n))' \sim N_n(0, \sigma^2 I)$

Siendo M el parámetro MESOR (Midline Statistic of Rhythm, una media ajustada al ritmo) [13], A la amplitud de la onda, y φ la acrofase.

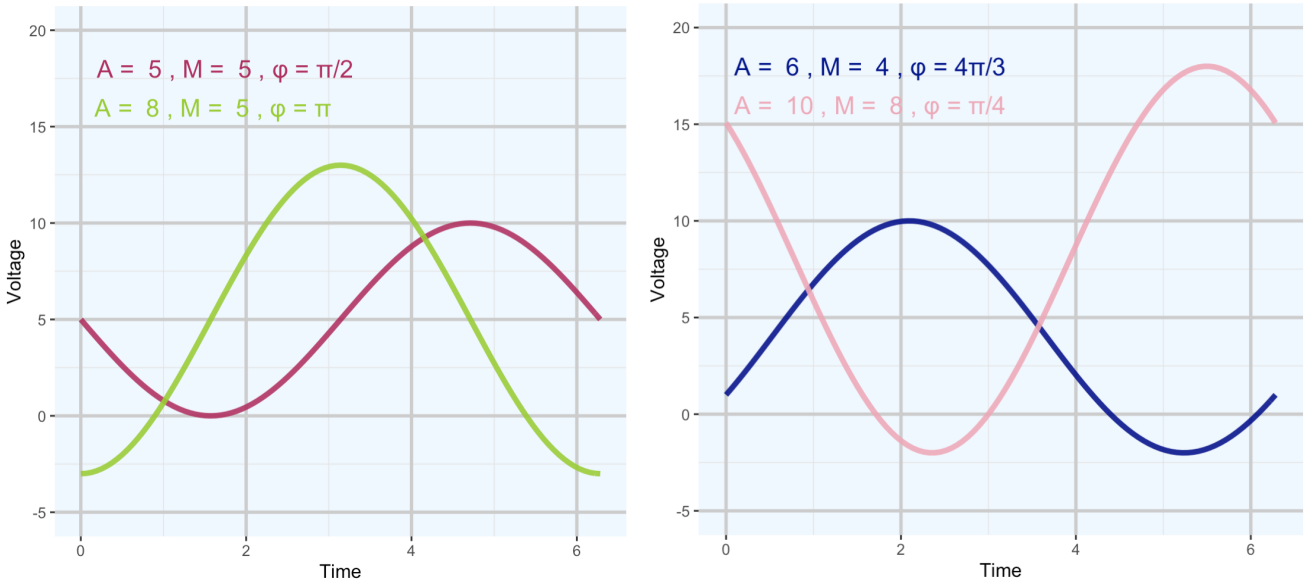


Figura 3.1: Representación del modelo cosinor para diferentes configuraciones paramétricas

En las figuras 3.1 podemos ver la representación de $\mu(t_i)$ del modelo Cosinor con diferentes configuraciones paramétricas. Observamos como la amplitud de la onda aumenta al aumentar el valor de A, siendo mayor en el gráfico en color verde de la figura de la izquierda y en el de color

rosa en el gráfico de la derecha. También vemos cómo el parámetro M cambia la posición de la onda, es decir, con un mayor valor del parámetro M, el valor de la onda en el eje y es mayor. Por último, si nos fijamos en el parámetro φ, se aprecia cómo cambia el momento de máxima elevación de la señal.

3.1.2. Modelo de Fourier (FD)

El modelo de Fourier es un modelo aditivo basado en la descomposición de Fourier. Permite describir modelos algo más complejos que el modelo Cosinor. Este modelo puede definirse de la siguiente forma:

$$X(t_i) = \mu(t_i) + \varepsilon(t_i) = A_0 + \sum_{n=1}^{N} \Big(A_n \cdot \cos(n \cdot t_i) + B_n \cdot \sin(n \cdot t_i) \Big) + \varepsilon(t_i);\ i = 1, \ldots, n \quad (3.3)$$

1. $A_0 \in \mathbb{R};\ A_1, \ldots, A_n,\ B_1, \ldots, B_n \in \mathbb{R}^+$
2. $(\varepsilon(t_1), \ldots, \varepsilon(t_n))' \sim N_n(0, \sigma^2 I)$

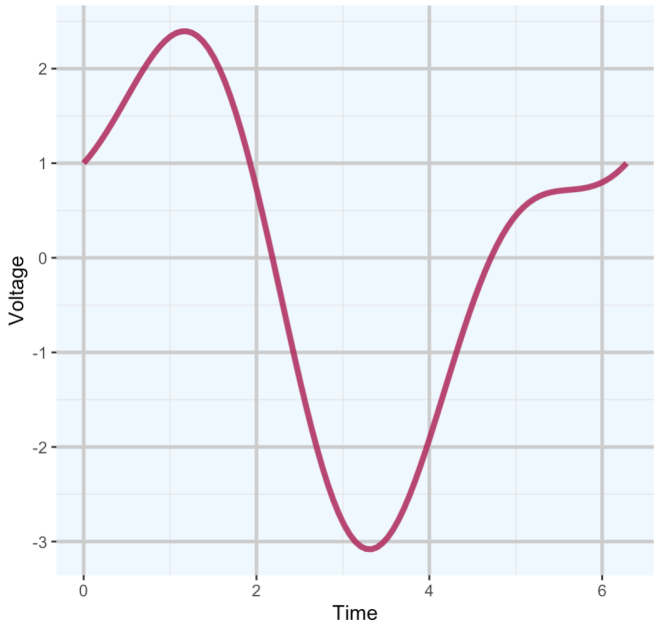
El modelo Cosinor explicado en el apartado anterior es un caso particular del modelo de Fourier con 1 armónico (FD^1).

Un ejemplo muy habitual del modelo de Fourier es FD^2, es decir, el modelo de Fourier con dos armónicos. Podemos definirlo de la siguiente forma:

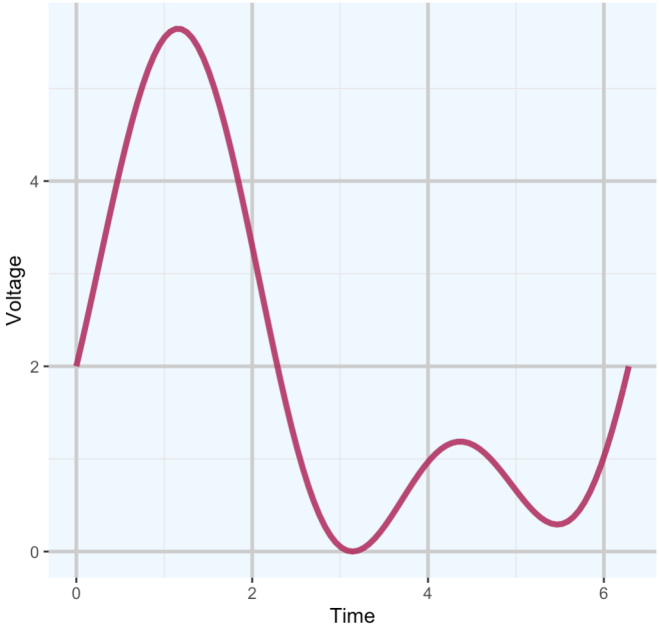
$$X(t_i) = A_0 + A_1 \cdot \cos(t_i) + B_1 \cdot \sin(t_i) + A_2 \cdot \cos(2t_i) + B_2 \cdot \sin(2t_i) + \varepsilon(t_i);\ i = 1, \ldots, n \quad (3.4)$$

1. $A_0 \in \mathbb{R};\ A_1, A_2, B_1, B_2 \in \mathbb{R}^+$
2. $(\varepsilon(t_1), \ldots, \varepsilon(t_n))' \sim N_n(0, \sigma^2 I)$

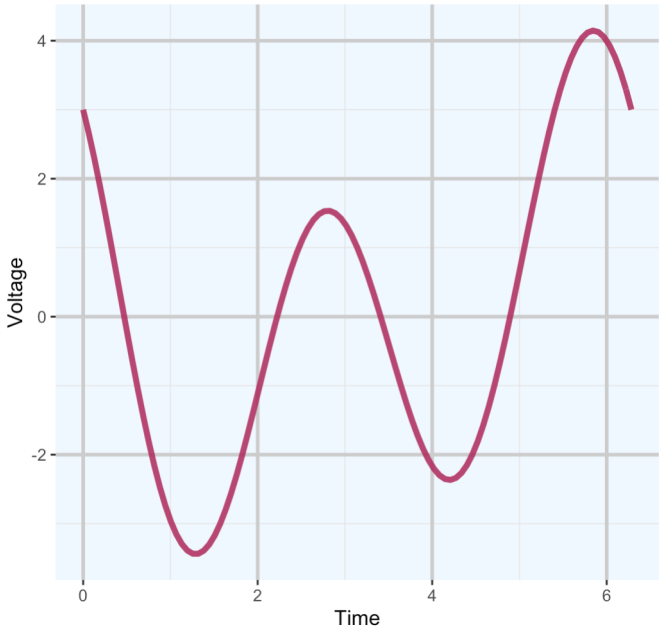
A continuación, en las figuras 3.2 se muestra la representación del modelo de Fourier FD^2 con diferentes configuraciones paramétricas.



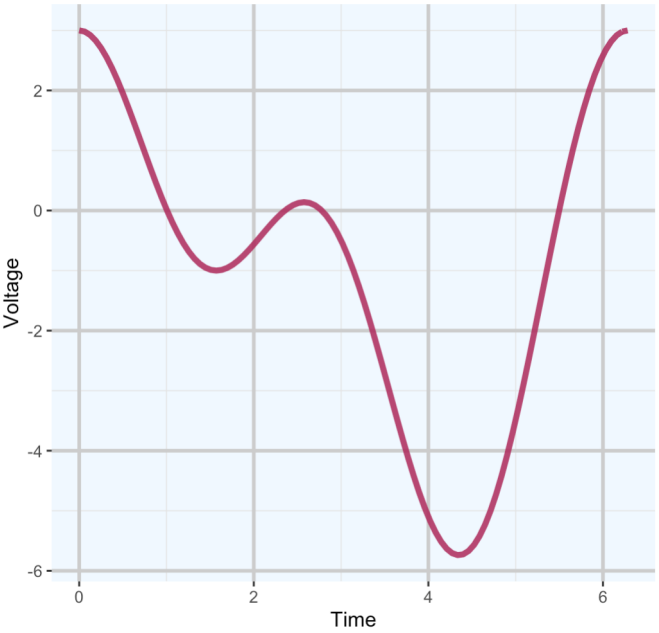
$A_0 = 0,\ A_1 = 2,\ A_2 = -1,\ B_1 = 1,\ B_2 = 0$



$A_0 = 2,\ A_1 = 1,\ A_2 = -1,\ B_1 = 2,\ B_2 = 1$



$A_0 = 0,\ A_1 = 1,\ A_2 = 2,\ B_1 = -1,\ B_2 = -2$



$A_0 = -1,\ A_1 = 2,\ A_2 = 2,\ B_1 = 2,\ B_2 = -1$

Figura 3.2: Modelo de Fourier: diferentes configuraciones de sus parámetros

Fourier permite describir formas más complejas que el modelo Cosinor, al contar con más parámetros y, por lo tanto, más grados de libertad.

3.1.3. El modelo FMM

El modelo FMM (Frequency Modulated Möbius) es un modelo paramétrico muy flexible, que permite deformaciones sinusoidales para acomodarse a asimetrías en múltiples aplicaciones [14]. La estimación de los parámetros de este modelo es sencilla y sus parámetros tienen una interpretación directa. Las principales fortalezas de este modelo respecto a los modelos de Fourier y Cosinor son: una formulación paramétrica sencilla, interpretabilidad y flexibilidad de los parámetros que describen las ondas de las señales, la identificabilidad y precisión de los estimadores y la robustez frente al ruido.

Las señales oscilatorias se definen en el dominio del tiempo. Se asume que el tiempo está en el intervalo $[0, 2\pi)$. En otro caso, se debe transformar el tiempo $t' \in [t_0, T+t_0]$ mediante: $t = \frac{(t'-t_o)2\pi}{T}$, siendo t_0 el tiempo inicial y T el periodo.

El modelo FMM puede estar formado por la suma de varias ondas o solo por una. Podemos definir una onda de la siguiente forma:

$$W(t, v) = A \cdot cos(\phi(t; \alpha, \beta, w)) \tag{3.5}$$

Los diferentes parámetros que componen esta onda se explicarán en el próximo apartado.

3.1.3.1. El modelo FMM_1

El modelo FMM_1, o FMM monocomponente [15], es el modelo FMM más simple, ya que consta de una señal formada por una única onda. Este modelo es capaz de describir patrones no sinusoidales, gracias a la inclusión de la función de enlace de Möbius como fase $\phi(t; \alpha, \beta, w)$.

Podemos definir el modelo FMM monocomponente de la siguiente forma:

$$X(t_i) = \mu(t_i) + \varepsilon(t_i) = M + A \cdot cos(\phi(t; \alpha, \beta, w)) + \varepsilon(t_i), i = 1, \ldots, n. \tag{3.6}$$

Donde:

1. $M \in \mathbb{R}$, $A \in \mathbb{R}^+$

2. $\phi(t_i;\alpha,\beta,\omega) = \beta + 2 \cdot arctan(w \cdot \tan(\frac{t_i-\alpha}{2}));\ \alpha,\ \beta \in [0, 2\pi],\ w \in [0, 1]$

3. $(e(t_1), \ldots e(t_n))' \sim N_n(0, \sigma^2 I)^2$

3.1.3.1.1. Parámetros del modelo

Los cinco parámetros del modelo FMM_1 mostrados en la ecuación 3.6 son:

- $\boldsymbol{M}$: intercept ($M \in \mathbb{R}$)
- $\boldsymbol{A}$: amplitud de la onda ($A \in \mathbb{R}^+$)
- $\boldsymbol{\alpha}$: parámetro de localización de fase ($\alpha \in [0, 2\pi]$)
- $\boldsymbol{\beta}$ y $\boldsymbol{\omega}$: parámetros de forma de la onda
  - $\boldsymbol{\beta}$: asimetría de la onda ($\beta \in [0, 2\pi]$)
  - $\boldsymbol{\omega}$: apuntamiento de la onda ($\omega \in [0, 1]$)

En la figura 3.3 podemos ver, para valores fijos de los parámetros M, A, β y ω ($M = 0$, $A = 1$, $\beta = \pi$ y $\omega = 1$), distintos patrones que describe una onda FMM al variar el parámetro α.

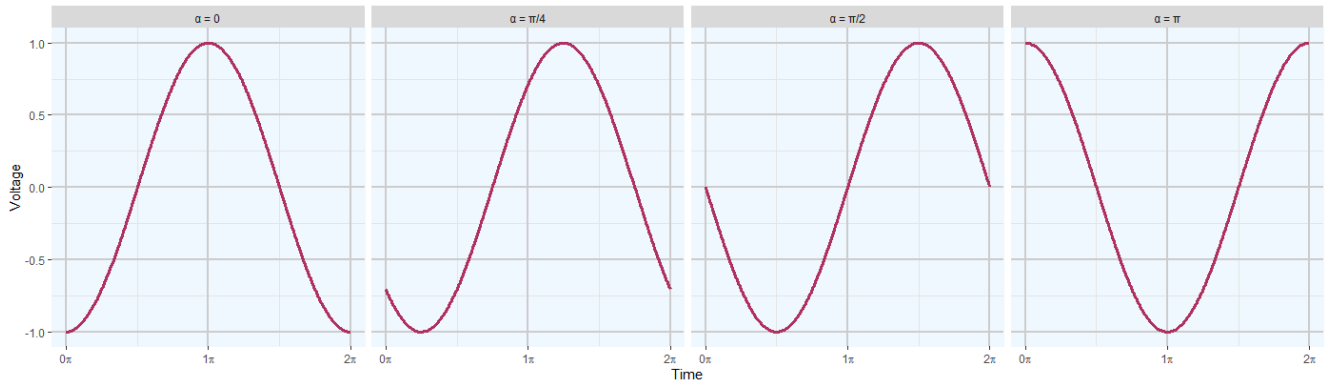


Figura 3.3: Modelos FMM_1 con $M = 0$, $A = 1$, $\beta = \pi$, $\omega = 1$

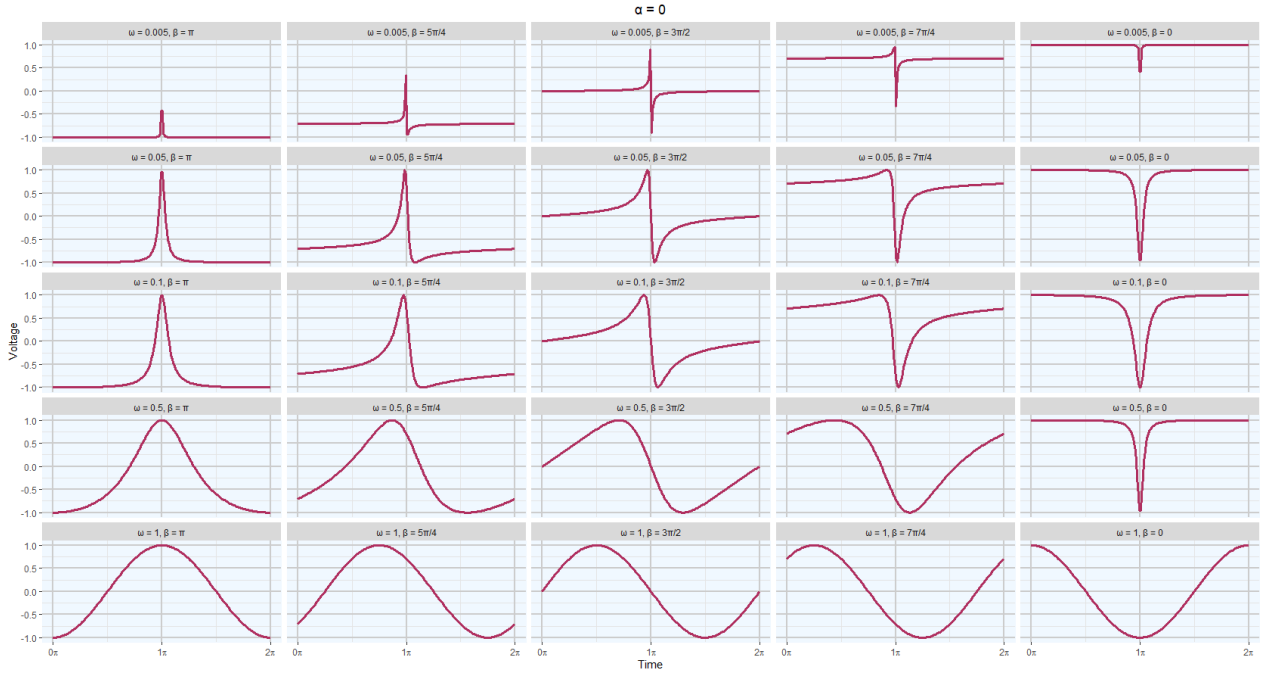


Figura 3.4: Modelos FMM_1 con $M = 0$, $A = 1$ y $\alpha = 0$

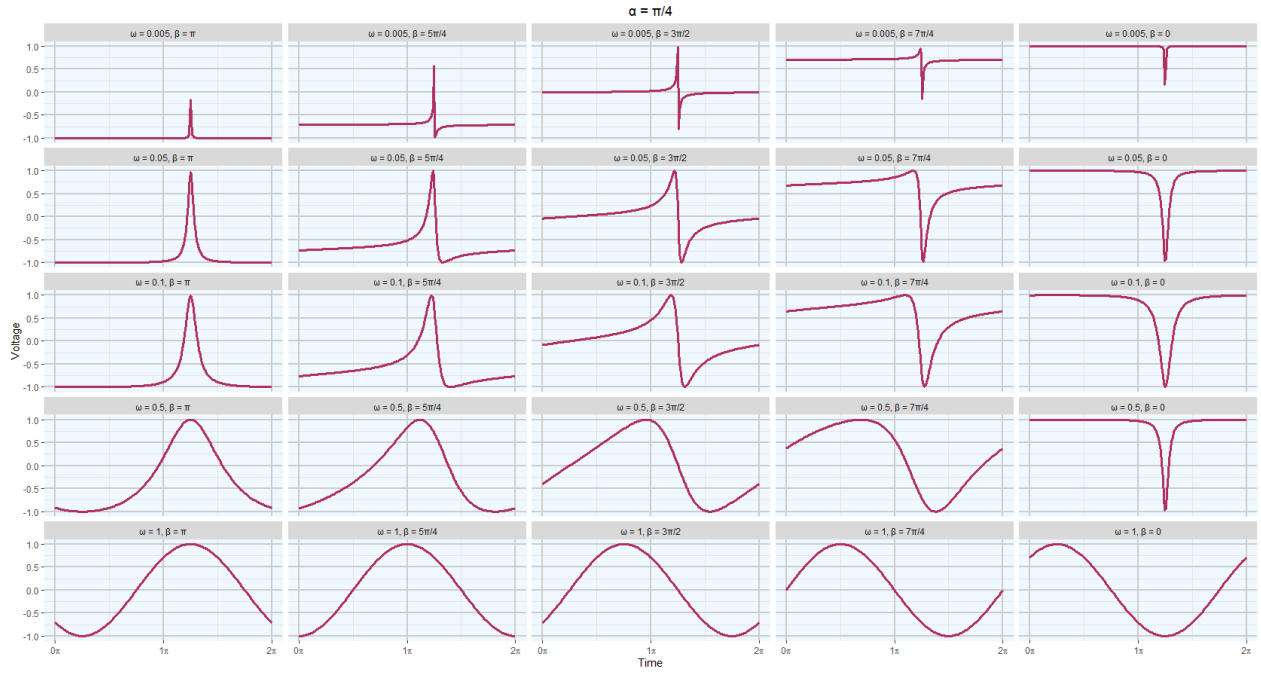


Figura 3.5: Modelos FMM_1 con $M = 0$, $A = 1$ y $\alpha = \frac{\pi}{4}$

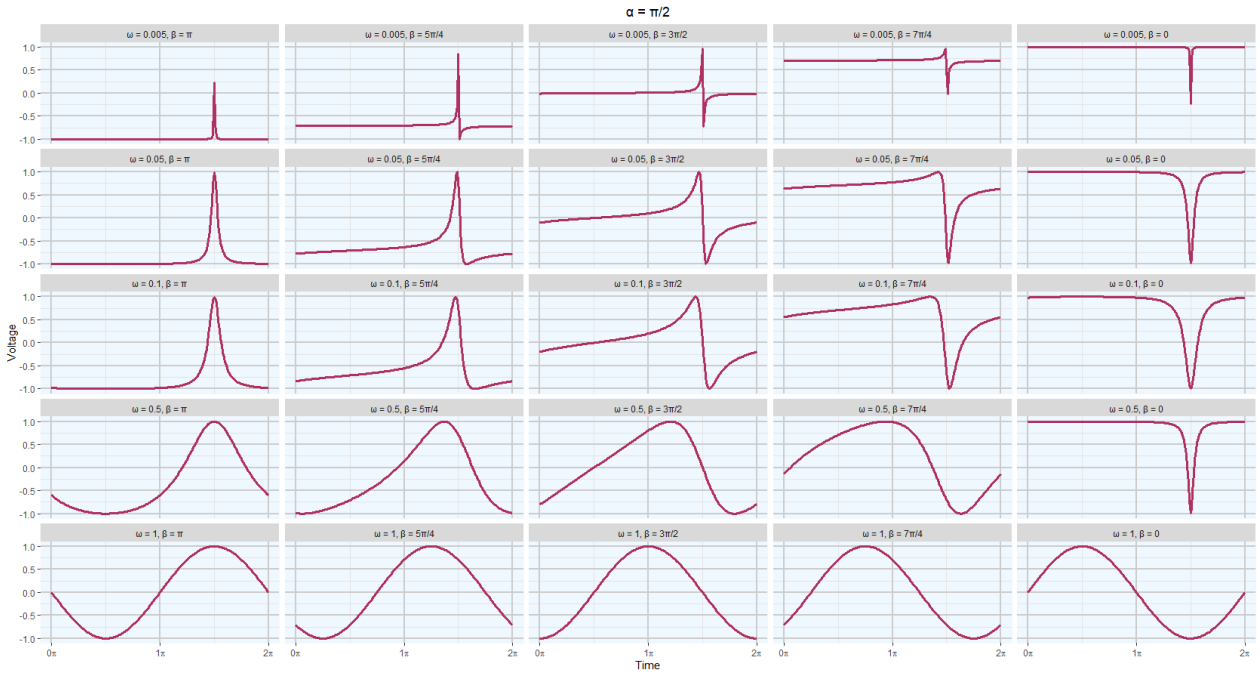


Figura 3.6: Modelos FMM_1 con $M = 0$, $A = 1$ y $\alpha = \frac{\pi}{2}$

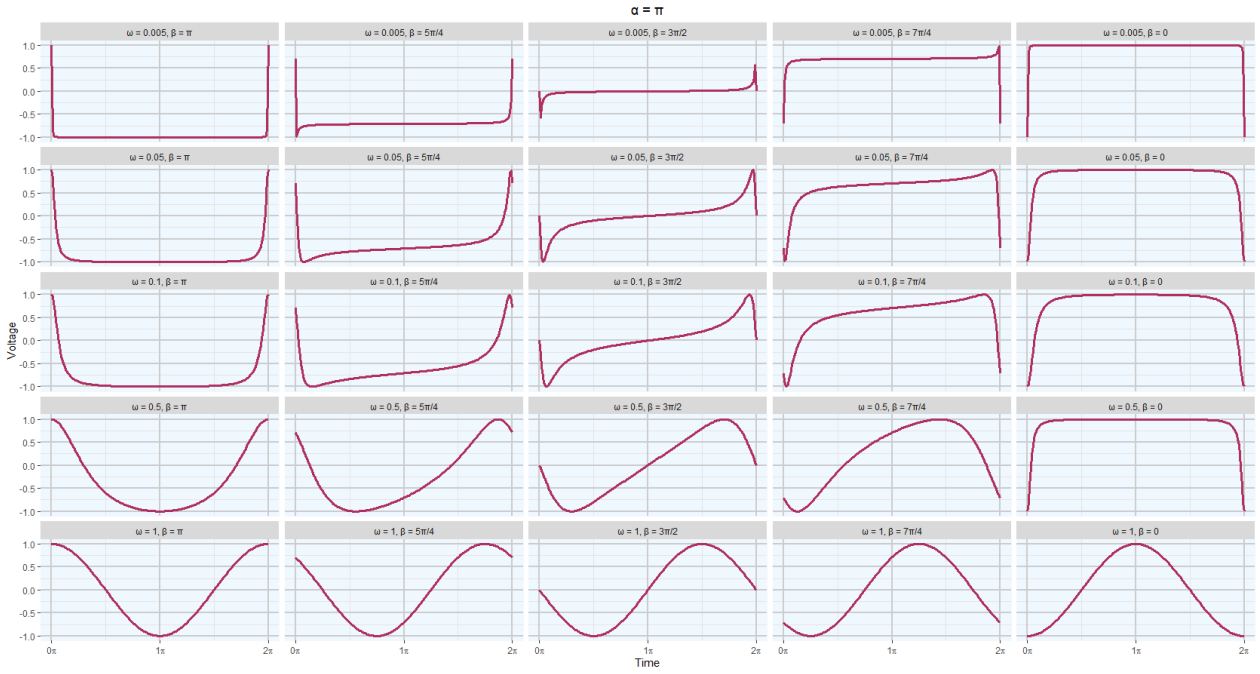


Figura 3.7: Modelos FMM_1 con $M = 0$, $A = 1$ y $\alpha = \pi$

En las figuras 3.4, 3.5, 3.6 y 3.7 se muestran diferentes patrones que describe una onda FMM_1 para distintos valores de los parámetros β y ω, con valores de α: 0 (figura 3.4), $\pi/4$ (figura 3.5), $\pi/2$ (figura 3.6) y π (figura 3.7) y el resto de parámetros con valores fijos.

El parámetro β describe la asimetría de la onda. Para valores de $\beta = 0$ y $\beta = \pi$, la onda es completamente simétrica. Sin embargo, para $\beta = 3\pi/2$, la señal es muy asimétrica. Una señal con $\alpha = 0$ y $\beta' = \beta - \pi$, describe la señal inversa a la de β que se muestra analíticamente con la igualdad trigonométrica: $cos(x - \pi) = -cos(x)$, $x \in [0, 2\pi]$.

El parámetro ω describe el apuntamiento de la onda. Valores de ω cercanos a 0 describen patrones muy afilados. A medida que ω incrementa su valor, los patrones se van suavizando, correspondiéndose con una curva sinusoidal al alcanzar un valor de 1. En este caso, el modelo FMM se corresponde con el modelo Cosinor, en el que la acrofase es: $\phi = \beta - \alpha$. Cuando ω es pequeño, los cambios en los patrones de la onda son sencillos de ver, pero cuando $\omega > 0.5$, las ondas se suavizan mucho y los cambios no son tan intuitivos.

Además de estos parámetros básicos, podemos definir otros dos parámetros de gran utilidad práctica: tiempo en el que la onda alcanza su mínimo (t_L) y su máximo (t_U). Estos parámetros se definen de la siguiente forma:

$$t_U = \alpha + 2 \cdot \arctan\left(\frac{1}{w}\tan\left(\frac{-\beta}{2}\right)\right) \tag{3.7}$$

$$t_L = \alpha + 2 \cdot \arctan\left(\frac{1}{w}\tan\left(\frac{\pi - \beta}{2}\right)\right) \tag{3.8}$$

Los valores de la señal en estos puntos se obtienen de la siguiente forma:

$$Z_u = X(t_U) = M + A \tag{3.9}$$

$$Z_L = X(t_L) = M - A \tag{3.10}$$

3.1.3.1.2. Estimador Máximo Verosímil

El estimador máximo verosímil (MLE) [17] de los parámetros del modelo FMM_1 es la solución del problema de optimización:

$$\hat{\theta} = \underset{\theta \in \Theta}{\operatorname{argmin}} \sum_{i=1}^{n} (X(t_i) - \mu(t_i, \theta))^2 \tag{3.11}$$

Θ es el espacio paramétrico de θ, siendo $\Theta = \mathbb{R} \times \mathbb{R}^+ \times [0, 2\pi) \times [0, 2\pi) \times [0, 1]$.

Una medida de la proporción de varianza explicada por el modelo es el R^2, que puede definirse de la siguiente forma:

$$R^2 = 1 - \frac{\sum_{i=1}^{n} (X(t_i) - \mu(t_i, \hat{\theta}))^2}{\sum_{i=1}^{n} (X(t_i) - \overline{X})^2} \tag{3.12}$$

3.1.3.1.3. Algoritmo de estimación

Para la estimación de los parámetros del modelo FMM_1 es necesario reformularlo de la siguiente forma [17]:

$$\mu(t_i, \theta) = M + A \cdot \cos(t_i^* + \varphi) \tag{3.13}$$

Donde: $t_i^* = \alpha + 2 \cdot \arctan\left(\omega \cdot \tan\left(\frac{t_i - \alpha}{2}\right)\right)$ y $\varphi = \beta - \alpha$ para $i = 1, \ldots, n$.

Mediante la identidad trigonométrica de la suma de ángulos podemos reescribir el modelo como:

$$X_i = M + \delta z_i + \gamma \omega_i + \varepsilon_i \tag{3.14}$$

Donde, $\delta = A \cdot \cos(\varphi)$, $\gamma = -A \cdot \sin(\varphi)$, $z_i = \cos(t_i^*)$, $\omega_i = \sin(t_i^*)$ y $\varepsilon_i \sim N(0, \sigma^2)$ para $i = 1, \ldots, n$.

Considerando unos valores iniciales fijos y conocidos para α y ω (suponiendo por tanto, valores de conocidos de t_i^* para $i = 1, \ldots, n$), el problema de estimación se reduce a resolver un problema de mínimos cuadrados. Así, la estimación de los parámetros M, A y β se realiza de la siguiente forma:

$$\hat{M} = \overline{X} - \hat{\delta} \sum_{i=1}^{n} z_i - \hat{\gamma} \sum_{i=1}^{n} \omega_i \tag{3.15}$$

$$\hat{A} = \sqrt{\hat{\delta}^2 + \hat{\gamma}^2} \tag{3.16}$$

$$\hat{\beta} = \alpha + \varphi \tag{3.17}$$

Teniendo en cuenta todo lo anterior, se describen los dos pasos del algoritmo de estimación:

1. Se define un *grid* de valores para (α, ω), se obtienen los estimadores de $\hat{M}$, $\hat{A}$ y $\hat{\beta}$ mediante mínimos cuadrados y se selecciona la solución de máxima verosimilitud.
2. Se realiza una optimización de Nelder-Mead. Este es un método heurístico de búsqueda que se utiliza para minimizar una función objetivo en un espacio multidimensional, dados los valores iniciales de los puntos a ser estimados. Busca de forma aproximada una solución óptima local variando la función a minimizar suavemente.

El estimador de la varianza (σ^2) se obtiene de los residuos del modelo, de la misma forma que se hace en los modelos lineales normales, de la siguiente forma:

$$\hat{\sigma^2} = \frac{\sum_{i=1}^{n}(X_i - \hat{\mu}_i)^2}{n-k} \tag{3.18}$$

Siendo k el número de parámetros libres utilizados para estimar μ.

3.1.3.2. El modelo FMM$_m$

El modelo FMM$_m$, también llamado modelo FMM multicomponente, se define como un modelo paramétrico, aditivo, más un error. Este modelo descompone una señal en una suma de ondas. Podemos definirlo de la siguiente forma:

$$X(t_i) = \mu(t_i, \theta) + \varepsilon(t_i) \tag{3.19}$$

Donde:

$$\mu(t_i, \theta) = M + \sum_{J=1}^{m} W(t_i, \nu_J) \tag{3.20}$$

$$W(t_i, \nu_J) = \sum_{J=1}^{m} A_J \cdot \cos(\phi_J(t_i)) \tag{3.21}$$

Siendo:

- $\nu = (A_J, \alpha_J, \beta_J, \omega_J)$
- $\theta = (M, \nu_1, \ldots, \nu_m)$, verificando:
  - $M \in \mathbb{R}$; $\nu_J \in \Theta_J = \mathbb{R}^+ \times [0, 2\pi) \times [0, 2\pi) \times [0, 1]$; $J = 1, \ldots, m$
  - $\alpha_1 \leq \alpha_2 \leq \cdots \leq \alpha_m \leq \alpha_1$
  - $A_1 = max_{1 \leq J \leq m} A_J$
- $(e(t_i), \ldots, e(t_n))' \sim N_n(0, \sigma^2 I)$

Los parámetros del modelo son identificables incluyendo en el modelo las restricciones que se acaban de comentar. α_1 no es necesariamente igual a $min_{1 \leq J \leq m} \alpha_J$ debido a que el orden es circular.

3.1.3.2.1. Estimador Máximo Verosímil

Al igual que en la estimación de los parámetros del modelo FMM_1, los parámetros del modelo FMM_m son la solución del problema de optimización:

$$\hat{\theta} = \underset{\theta \in \Theta}{\operatorname{argmin}} \sum_{i=1}^{n} (X(t_i) - \mu(t_i, \theta))^2 \tag{3.22}$$

Θ es el espacio paramétrico de θ, siendo $\Theta = \mathbb{R}^+ \times [0, 2\pi) \times [0, 2\pi) \times [0, 1]$.

3.1.3.2.2. Algoritmo de estimación

La estimación de los parámetros del modelo FMM_m se realiza mediante un algoritmo de *backfitting* ajustando los residuos del modelo [19]. Sea $\hat{W}_J^{(k)}(t_i)$ los valores ajustados por la onda J-ésima del modelo FMM en t_i, $i = 1, \ldots, n$ en la k-ésima iteración. Los pasos que sigue el algoritmo son los siguientes:

1. Inicializar $\hat{W}_1^{(0)}(t_i) = \cdots = \hat{W}_m^{(0)}(t_i) = 0$

2. Etapa de *backfitting*. Para $J = 1, \ldots, m$, calcular:

$$r_J^{(k)}(t_i) = X(t_i) - \sum_{I=1}^{J-1} \hat{W}_I^{(k)}(t_i) - \sum_{I=J+1}^{m} \hat{W}_I^{(k-1)}(t_i) \tag{3.23}$$

   Ajustando un modelo FMM_1 a $r_J^{(k)}(t_i)$, obteniendo: $\hat{\alpha}_J^{(k)}$, $\hat{\beta}_J^{(k)}$, $\hat{\omega}_J^{(k)}$ y $\hat{W}_J^{(k)}(t_i)$

3. Se repite el paso 2 hasta que se alcanza un criterio de parada. Este criterio se define como la diferencia entre la variabilidad explicada en dos iteraciones consecutivas ($R_k^2 - R_{k-1}^2 \leq C$). R_k^2 es la proporción de varianza explicada por el modelo en la k-ésima iteración y está definido en 3.12. C es una constante.

4. $\hat{M}$ y $\hat{A}_J$ se calculan resolviendo:

$$\min_{M \in \mathbb{R}; A_j \in \mathbb{R}^+} \sum_{i=1}^{n} (X(t_i) - M - \sum_{J=1}^{m} A_j \cdot \cos(\hat{\phi}_J(t_i)))^2 \tag{3.24}$$

3.1.4. Comparación COS, FD y FMM

En esta sección analizamos dos ejemplos de ajuste de los modelos Cosinor, Fourier y FMM a señales reales y establecemos comparaciones. Estos datos pertenecen al conjunto de datos de señales neuronales de ratas que analizaremos en detalle más adelante.

	Modelo COS	Modelo FD^2	Modelo FMM_1
R^2	0.2826213	0.4792064	0.953504
MSE	160.5018	116.5191	10.41046

Tabla 3.1: Comparación R^2 modelos COS, FD^2 y FMM_1

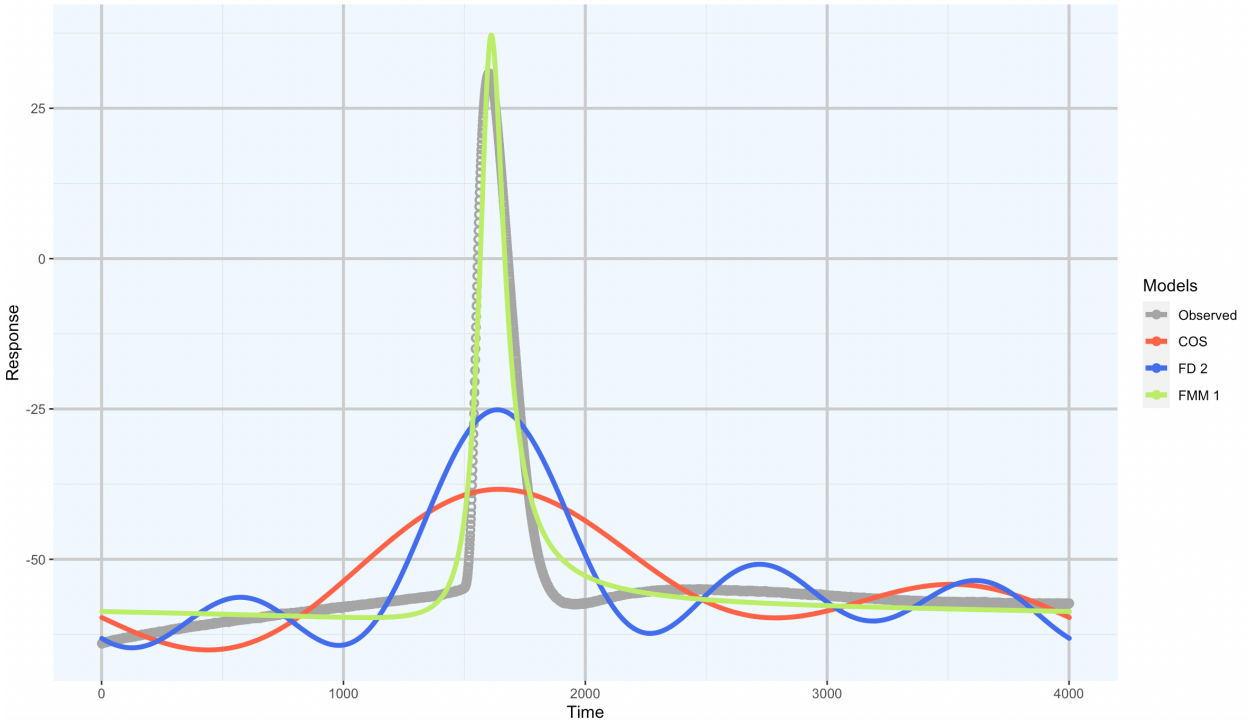


Figura 3.8: Comparación modelos COS, FD^2 y FMM_1

En la figura 3.8 podemos ver en un ejemplo real, las predicciones de los modelos Cosinor (rojo), Fourier con dos armónicos (azul) y FMM monocomponente (verde), junto a los datos observados (gris). El motivo de comparar estos dos últimos modelos es que ambos tienen el mismo número de parámetros, 5. Observando el gráfico, vemos que el modelo Cosinor no es bueno para el ajuste de estos datos. Además, como se muestra en la tabla 3.1, explica una variabilidad (R^2) de tan solo el 28 %. La predicción del modelo de Fourier con dos armónicos (FD^2) parece que ajusta algo mejor que el modelo Cosinor, aunque está lejos de ser un buen ajuste. Explica tan solo el 48 % de la variabilidad de los datos. La predicción del modelo FMM_1 ajusta de forma bastante precisa los datos, mucho mejor que los modelos Cosinor y FD^2. La variabilidad explicada es de más del 95 %. Si comparamos el MSE (*Mean Squared Error*) de los tres modelos, vemos que con diferencia, es más bajo en el caso del modelo FMM_1. Por todos estos motivos, podemos decir que para el ajuste de estos datos, el modelo FMM_1 es el mejor.

A continuación, vamos a ajustar los mismos datos con el modelo FMM_3, modelo que se estudiará en detalle más adelante. Este modelo tiene 13 grados de libertad (1 parámetro M, 3 parámetros A, 3 parámetros α, 3 parámetros β y 3 parámetros ω). Vamos a compararlo con un modelo Cosinor y con un modelo FD^7. Se ha elegido este modelo para tener aproximadamente el mismo número de grados de libertad. En este caso, el modelo FD^7 tiene 14 (2· nº armónicos, es decir, $2 \cdot 7$).

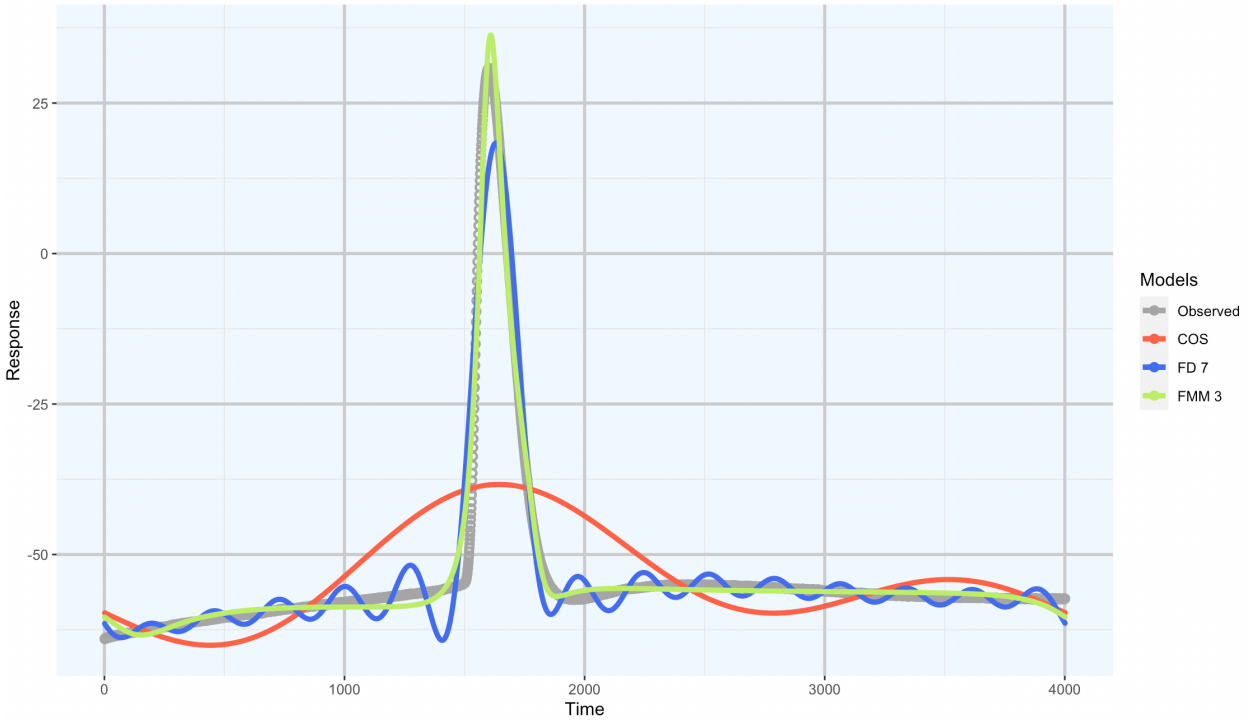


Figura 3.9: Comparación modelos COS, FD^7 y FMM$_3$

	Modelo COS	Modelo FD^7	Modelo FMM$_3$
R^2	0.2826213	0.9385833	0.9824824
MSE	160.5018	13.74099	3.919267

Tabla 3.2: Comparación R^2 modelos COS, FD^7 y FMM$_3$

En la figura 3.9 podemos ver en gris la señal observada. Claramente el modelo que mejor se ajusta es el modelo FMM$_3$ (en verde). Gracias a su gran flexibilidad, permite adaptar su forma casi a la perfección a los datos. Podemos ver cómo el modelo Cosinor (en rojo) es demasiado sencillo para ajustar los datos y que el modelo de Fourier (azul), ajusta bastante bien el *peak* de los datos, pero sobreajusta mucho en los momentos de reposo de la neurona. Si nos fijamos en la tabla 3.2 podemos ver el R^2 y el MSE de cada uno de estos modelos oscilatorios. El model Cosinor predice muy mal estos datos, con un R^2 de tan solo el 28 % y con un MSE muy superior a los otros dos modelos. El modelo FMM$_3$ vuelve a destacar por ser el modelo con un mejor R^2 y un menor MSE, aunque el modelo de Fourier con 7 armónicos también explica mucha variabilidad y tiene un MSE bastante bajo. Cabe destacar que el modelo FD^7 también es más complejo ya que

tiene un grado de libertad más, por lo que a la hora de tener que elegir un modelo de entre estos tres, nos quedaríamos con el modelo FMM_3.

3.2. Técnicas de aprendizaje supervisado

El aprendizaje supervisado es la técnica más común de *machine learning*. Permite deducir una respuesta a partir de un conjunto de datos, sabiendo cuál debe ser la salida correcta. Podemos clasificar los problemas de aprendizaje supervisado en dos tipos: **regresión** y **clasificación**. En los problemas de regresión se trata de predecir un valor numérico dentro de un valor finito de posibles resultados. En los problemas de clasificación se trata de predecir el valor de una clase, es decir, un valor discreto.

3.2.1. Clasificadores

En este trabajo utilizaremos solo salidas discretas, por lo que tan solo hablaremos de un problema de clasificación. Trataremos los siguientes tipos de clasificadores: discriminante lineal, árboles de clasificación, *random forest* y SVM (*Support Vector Machine)*. Esta sección se ha escrito siguiendo [20], [21]. Para más detalle respecto a estos procedimientos, se recomienda la lectura de los textos citados.

3.2.1.1. Discriminante lineal (LDA)

Es una herramienta utilizada no solo para clasificación, sino también para reducción de la dimensionalidad y visualización de datos. Es un modelo muy simple que permite obtener resultados robustos e interpretables. Permite expresar los datos como combinaciones lineales de las variables. De esta forma se puede ver cómo y en qué medida influye cada variable en el modelo.

Este método está muy relacionado con el análisis de componentes principales (PCA), ya que ambos buscan combinaciones lineales de variables que explican los datos de la mejor forma posible. Sin embargo, LDA intenta modelar de forma explícita la diferencia entre las clases, mientras que PCA no tiene en cuenta las diferencias de clase.

En un problema genérico de clasificación con K clases, el objetivo es construir una regla discriminante que divida el espacio de los datos en K regiones disjuntas (una región por clase). De esta forma, la clasificación mediante LDA simplemente significa que asignamos una instancia a una clase si está en su región. Para conocer en qué región cae, se utiliza una regla bayesiana de clasificación.

3.2.1.2. Árboles de clasificación

Los árboles de clasificación son una herramienta muy útil para problemas de decisión. Permiten visualizar y representar de forma explícita la toma de decisiones de un clasificador.

En la tarea de clasificación, únicamente utilizan variables categóricas. Por lo tanto, si existen variables numéricas continuas en el conjunto de datos, deben discretizarse. El algoritmo de clasificación de un árbol comienza eligiendo una variable y obteniendo un punto de corte (`c`) de tal forma que las observaciones con un valor de la variable menor o igual a `c` van a un nodo, y las que tienen un valor mayor, van a otro. Cada nodo puede volver a dividirse eligiendo una variable y un punto de corte (esta variable puede haberse utilizado anteriormente). Podremos considerar a un nodo como terminal si ya no se puede dividir más.

Para la clasificación de instancias no vistas, recorremos el árbol de decisión hasta llegar a un nodo terminal.

Para evitar que se produzca sobreajuste, se parte de árboles muy desarrollados y se podan algunas ramas finales. Es preferible que los datos estén explicados por árboles de poca profundidad, debido a que las hipótesis más simples son capaces de generalizar mejor, ya que contendrán menos atributos irrelevantes.

3.2.1.3. Random Forest

Es un método de *ensemble learning* (*bagging*) que combina un gran número de árboles de clasificación. El modelo de *random forest* es muy popular debido a que es un método muy sencillo de entrenar. Cada árbol de clasificación individual hace una predicción de clase. En problemas de regresión, se calcula la media de la salida predicha y esta será la predicción obtenida por el *random forest*. En problemas de clasificación, la clase más elegida se convierte en la predicción del modelo, a esto se le llama "votación de la mayoría".

Al igual que el resto de métodos de *bagging*, es un algoritmo formado por la combinación de algoritmos más simples que se ejecutan de forma paralela e independiente, y se elige como salida, el voto mayoritario de los algoritmos que lo componen.

Los árboles de clasificación que hemos comentado antes son propensos a sobreajustar el conjunto de entrenamiento. Los *random forests* permiten compensar el sobreajuste y generalizar bien. También son buenos para conjuntos de datos de alta dimensionalidad.

Los parámetros que se pueden ajustar, entre otros, son:

- Número de árboles que componen el *random forest*.

- Número de características por árbol, habitualmente se calcula de la siguiente forma:

$$\text{mtry} = \sqrt{p-1} \tag{3.25}$$

  Siendo `p` el número total de predictores del modelo.

3.2.1.4. Support Vector Machine (SVM)

Es un método de clasificación supervisada que permite encontrar clasificadores lineales en espacios transformados. Su objetivo es encontrar un hiperplano multidimensional que separe las instancias de los datos. Existen multitud de posibles hiperplanos que separen los datos, pero el objetivo de SVM es encontrar el plano que tenga el mayor margen, es decir, la mayor distancia al hiperplano respecto a los puntos de las diferentes clases. En la figura 3.10 podemos ver una representación clara de cómo es el hiperplano y el margen máximo para separar las instancias de dos clases diferentes.

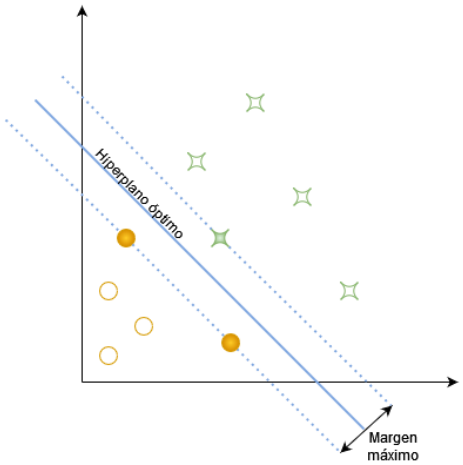


Figura 3.10: Hiperplano SVM

Las únicas observaciones que contribuyen a calcular los coeficientes del hiperplano máximo son las que se encuentran en los márgenes, se denominan **puntos soporte** o **vectores soporte**. En la figura 3.10 los puntos soporte son los puntos coloreados que se encuentran sobre los márgenes.

En el caso de tener clases no linealmente separables, se debe realizar una transformación no lineal del espacio de entrada. Estas transformaciones se llevan a cabo aplicando un *kernel* a los datos, normalmente polinomial o radial. A este último se le llama RBF (Radial Basis Function).

En el *kernel* polinomial, se deben tener en cuenta tres parámetros:

- **Grado**: es el grado del polinomio.
- **Escala**: factor de escala de los pesos.
- **C** o **coste**: penalización del error del modelo.

En el *kernel* radial se deben tener en cuenta dos parámetros:

- **Sigma**: determina el alcance de una instancia de entrenamiento.
- **C** o **coste**: penalización del error del modelo.

Es un algoritmo de clasificación rápido y fiable que funciona muy bien con una cantidad limitada de datos a analizar. Es muy efectivo en espacios de alta dimensión incluso cuando el número de dimensiones es más grande que el número de muestras. Es resistente al sobreajuste gracias a la regularización en función del valor del parámetro C. Un inconveniente de este modelo es que no se puede saber con certeza qué variables han sido determinantes en las predicciones, por eso habitualmente se le llama modelo de "caja negra".

3.2.2. Evaluación de resultados y métricas utilizadas

Para la evaluación de los resultados de cada uno de los diferentes tipos de clasificadores utilizaremos una técnica bien conocida llamada **validación cruzada**. Esta herramienta se utiliza habitualmente en *machine learning* cuando no se tienen suficientes datos como para aplicar otras técnicas más eficientes como la división de los datos en tres conjunto: entrenamiento, validación y test.

En la validación cruzada con k particiones, en primer lugar, se mezclan las instancias del conjunto de datos, de tal forma que el orden de las entradas y de las salidas es completamente aleatorio, evitando posibles sesgos en los datos. A continuación, se divide el conjunto de datos en k grupos (habitualmente 10) del mismo tamaño. Se utilizará un grupo para test y el resto para entrenamiento, y se repetirá este proceso k veces, cada vez, con un grupo de test diferente. Al final de la ejecución de la validación cruzada, tendremos k resultados. Para calcular el resultado final, simplemente, se calcula la media de los k resultados obtenidos.

Además de la típica métrica de **tasa de acierto**, es decir, el porcentaje de veces que el clasificador acierta en una predicción, podemos calcular:

- **Matriz de confusión**: es una herramienta que nos permite visualizar el desempeño de un clasificador. En la figura 3.11 podemos ver un ejemplo de matriz de confusión para un clasificador binario. En las filas representamos los valores que predice el clasificador, y en las columnas representamos los valores reales de las instancias. En la diagonal de la matriz podemos ver el número de instancias bien clasificadas, es decir, cuando los valores reales y predichos coinciden.

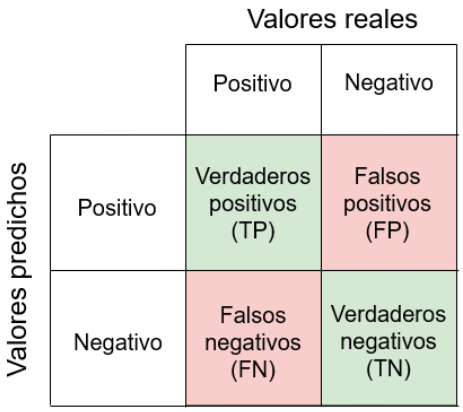


Figura 3.11: Matriz de confusión

- **Sensibilidad**: es una medida que nos indica cual es la probabilidad de que la predicción sea positiva siendo el valor real positivo. Se calcula de la siguiente forma:

$$\text{sensibilidad} = \frac{TP}{TP + FP} \tag{3.26}$$

- **Especificidad**: es una medida que nos indica cual es la probabilidad de que la predicción sea negativa siendo el valor real negativo. Se calcula de la siguiente forma:

$$\text{especificidad} = \frac{TN}{TN + FP} \tag{3.27}$$

Capítulo 4

Extracción y procesamiento de las señales

En este capítulo se explica todo el procesamiento de los datos utilizados, desde su extracción de la base de datos del proyecto Blue Brain, hasta el preprocesamiento de las señales para ajustar el modelo FMM y FMM_3 y la asignación de las ondas según su forma.

4.1. Extracción de datos

En este apartado comentaremos la base de datos del proyecto Blue Brain, de la que se obtienen todos los datos que vamos a utilizar. También comentaremos los datos y la librería utilizada para su extracción.

4.1.1. Blue Brain Project

El proyecto iniciado en la Escuela Politécnica Federal de Lausana (Suiza), tiene como objetivo principal el estudio de la estructura del cerebro de mamíferos, creando una simulación a nivel molecular de todo el cerebro. Mediante ingeniería inversa, se trata de profundizar en el entendimiento del funcionamiento del cerebro y sus disfunciones.

En este proyecto se establece la neurociencia de simulación como un enfoque complementario a la neurociencia experimental, clínica y teórica para poder entender el cerebro. Para ello, están creando la primera reconstrucción del mundo del cerebro de una rata. Ofrece un enfoque completamente nuevo para comprender la estructura y la función multinivel del cerebro.

La base de datos del proyecto está disponible en [22]. En este trabajo se analizará el conjunto de datos: "**Rat Somatosensory Cortex Neurons (hind limb)**", en el cual se estudian, como su nombre indica, las neuronas de la corteza somatosensorial de la rata (en su miembro posterior).

Estos datos contienen registros eléctricos de las neuronas de una rata de la especie *Rattus norvegicus*, de la cepa *Han wistar*, entre el día 13 y 16 después de nacer.

Los datos se han tomado mediante el software **IGOR Pro**, un software de gráficos científicos y análisis de datos que recoge la respuesta de voltaje en milivoltios (mV) de neuronas individuales al ser sometidas a estímulos de corriente en picoamperios(pA), mediante la técnica "patch clamp", en la que los electrodos de medición se sitúan pegados a la membrana de la neurona.

Nuestro conjunto de datos consiste en un conjunto de señales recogidas de 235 tipos de neuronas diferentes, siendo un total de 5728 señales a analizar. También, contaremos con la expresión genética celular de 179 de estas neuronas.

Dentro de estas señales, hay dos grupos distinguidos: señales de neuronas en reposo y señales de neuronas estimuladas eléctricamente. En nuestro caso, solo vamos a trabajar con el segundo grupo.

4.1.2. IgorR

IgoR es una librería de R que nos permite la lectura de datos de ficheros binarios generados por el software Igor Pro. Este es un potente software que permite la creación de gráficos, análisis de datos, procesamiento de imágenes y programación. Este software además, permite la recopilación de datos de la herramienta NIDAQ Tools MX. Esta herramienta permite la lectura de ondas analógicas, en nuestro caso, las ondas cerebrales de las ratas.

IgoR permite la lectura de ficheros con formato de salida `.ibw` (Igor Binary Wave). También ficheros `.pxp` (Packed Experiment) y ficheros generados por la herramienta Neuromatic, también del software Igor Pro. En este trabajo todos los ficheros que contienen las ondas neuronales a analizar, se encuentran en formato `.ibw`.

Se ha empleado la versión 0.7.2 de IgoR, disponible en el repositorio de GitHub [23]. Para la lectura de datos, utilizamos el comando `read.ibw(file)`. Nos devuelve un vector con la señal que ya podremos utilizar en R. También nos proporciona otro tipo de información sobre la señal: las unidades de los datos, el nombre de la señal, su longitud, etc. Para extraer estas características, podemos usar la siguiente instrucción: `attr(señal, ''tipoAtributo'')$atributo`. Por ejemplo, para obtener el nombre de la señal, tendríamos que escribir lo siguiente: `attr(wave, "WaveHeader")$WaveName)`. Para extraer información extra sobre la frecuencia a la que se han tomado los datos, podemos utilizar la instrucción: `tsp.igorwave(señal)`.

4.1.3. Expresión genética de las células

La información de la expresión genética de las células se recoge en formato `.txt` para cada célula. La lectura de datos es simple, mediante la instrucción: `read.table(fichero.txt)`.

Cada fichero contiene la expresión genética de cada célula. Cuenta con siete variables dicotómicas que nos dicen si siete genes diferentes se han activado o no.

Estos siete genes son: PV (*Parvalbumin positive*), CB (*Calbindin*), SOM (*Somatostatin*), VIP (*Vasoactive Intestinal Peptide*), NPY (*Neuropeptide Y*), CCK (*Cholecystokinin*) y CR (*Calretinin*).

4.2. Preprocesamiento de los datos

Ante las 5728 señales a analizar pertenecientes a 235 diferentes tipos de células, vamos a necesitar un importante preprocesamiento de los datos antes de poder aplicar el modelo FMM.

Para ello, vamos estudiar todas las señales y clasificarlas por el número de peaks que tengan, quedándonos con aquellas que tienen un único *peak*. Vamos a crear una tabla resumen en la que almacenemos cuáles son estas señales y la célula a la que pertenecen, junto con la información de la expresión genética de cada célula.

Posteriormente, vamos a recortar la señal en un intervalo de (2 ms, 3 ms), representándola y almacenándola en un fichero .csv, para su uso posterior. También vamos a estudiar ciertos artefactos que quedan en las señales una vez recortadas, al igual que antes, representando cada una de ellas y almacenándolas en un fichero .csv. Una vez tengamos este procesamiento de todas las señales hecho, podremos ajustarlas mediante el modelo FMM.

4.2.1. Detección de *peaks*

Para la detección de *peaks* se ha desarrollado un algoritmo que detecta cuándo se produce este pico en la estimulación eléctrica de la neurona. Para etapas posteriores nos interesa saber cuántos *peaks* tiene cada señal y la posición de estos *peaks* en el intervalo de tiempo de medición de la señal.

En primer lugar, para cada señal, si detectamos que es una neurona en reposo, simplemente se descarta, no nos interesa estudiarla, ya que no tiene *peaks*. Para el resto de señales que sí que tienen *peaks*, estudiamos los casos en los que el voltaje es elevado (debido a que que con voltajes bajos no se desencadenan los *peaks*). Buscamos los puntos en los que haya un cambio de pendiente

en la señal, que pase de ser creciente a decreciente, para ello estudiamos la diferencia de voltaje entre cada instante de tiempo y los dos instantes anteriores y los dos siguientes.

En ocasiones, debido a la alta frecuencia de muestreo con la que se han tomado los datos, el número de puntos es muy alto, por lo que las técnicas habituales de detección de *peaks* no sirven. La estrategia seguida en estos casos es, antes de considerar a un punto como *peak*, estudiamos si hay algún otro punto en un muy breve espacio de tiempo que ya hayamos seleccionado como *peak*. En este caso, elegimos el punto con un mayor voltaje.

Una vez que hemos detectado el número de *peaks* y la posición de cada uno de ellos en la señal, procedemos a almacenar esta información en una tabla (en formato `.csv`). Además, crearemos otro fichero en el que se almacenan únicamente las señales con un *peak*, porque son las señales que nos interesa estudiar. Se almacenará el nombre de la señal, el tipo de neurona al que pertenece y la posición en la que se encuentra el *peak*, junto con la información de la caracterización genética de cada tipo de neurona. De las 5728 señales totales de la base de datos del proyecto Blue Brain, el número de señales con un único *peak* es 115, aunque de estas solo tenemos información genética de 105, por lo que este es el número de instancias del conjunto de datos que se utilizará a partir de ahora.

4.2.2. Recorte de la señal

A partir del fichero que se ha generado con toda la información de las señales con un único *peak*, cargamos las señales y procedemos a recortarlas. Habitualmente el tiempo en el que se segmenta un AP se define como:

$$[t_s - 2k, t_s + 3k] \tag{4.1}$$

Siendo t_s el instante de tiempo [14] en el que el que se produce el *peak*; d, el tiempo necesitado por cada neurona para producir un *peak* cuando recibe un estímulo eléctrico; y k, una cantidad de tiempo concreta en milisegundos. En nuestro caso, emplearemos $k = 9$.

En las figuras 4.1 y 4.2 podemos ver un ejemplo de una señal original (4.1) y una señal recortada (4.2). Se aprecia que en la figura de la izquierda, la señal completa, tiene ruido en los extremos debido a que justo son esos instantes en los que se colocó el electrodo sobre la neurona. En la figura de la derecha podemos ver que, además de eliminar el problema existente, el *peak* se encuentra perfectamente recortado según el intervalo que se mencionó en la ecuación 4.1.

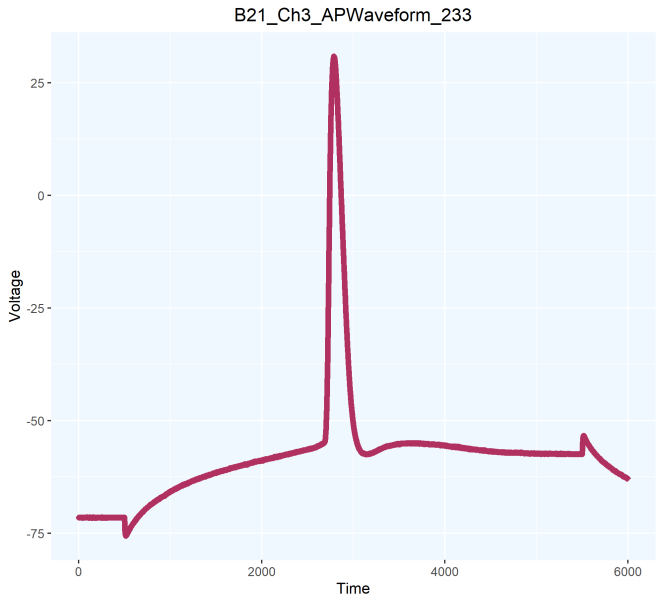


Figura 4.1: Señal completa

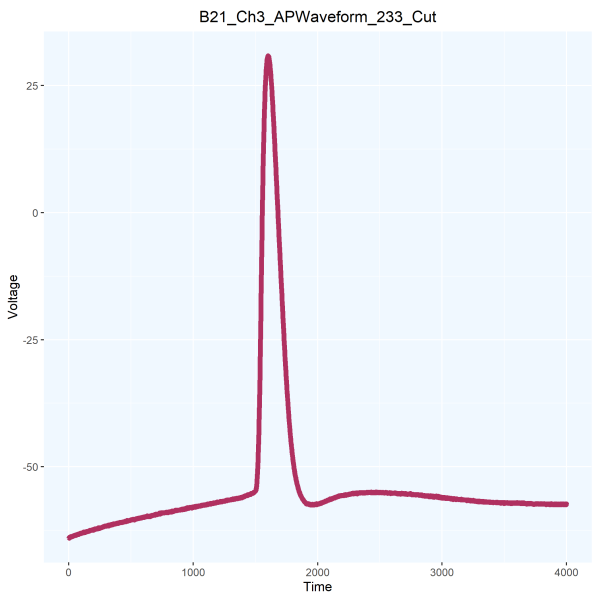


Figura 4.2: Señal recortada

4.2.3. Eliminación de artefactos

En ocasiones, debido a problemas en la toma de los datos, no se consiguen eliminar por completo los artefactos en los extremos de la señal al recortarla, debido probablemente, a que el electrodo se retiró de la neurona antes de que terminara la etapa de hiperpolarización. También puede ocurrir que, las señales de un único *peak* de la base de datos, ya hayan sido previamente recortadas de otras señales con varios *peaks*, de forma que queden restos de otros en sus extremos.

Para ajustar los modelos a las señales, debemos eliminar diversos artefactos. Para ello, en los momentos en los que cada señal se encuentra en estado de reposo, se estudia la existencia de pequeños *peaks*. Una vez detectados, se buscan los puntos en los que aproximadamente se inician estos pequeños *peaks*, y se sustituye por una secuencia generada con los valores del punto inicial y final de los artefactos detectados. Como en algunos casos, los puntos iniciales y finales ya llevan consigo un cambio de tendencia en la señal, indicativa de este aumento de potencial, se ha decido que la sustitución de los valores de voltaje de los artefactos por la secuencia generada sea desde unos pocos puntos antes y después.

Además, en algunos casos, estos pequeños *peaks* se detectan justo al final de la señal, por lo que no podríamos hacer lo que se ha comentado previamente. La solución elegida para estos casos ha sido trazar una línea recta desde el extremo izquierdo del artefacto y hasta el final de la señal.

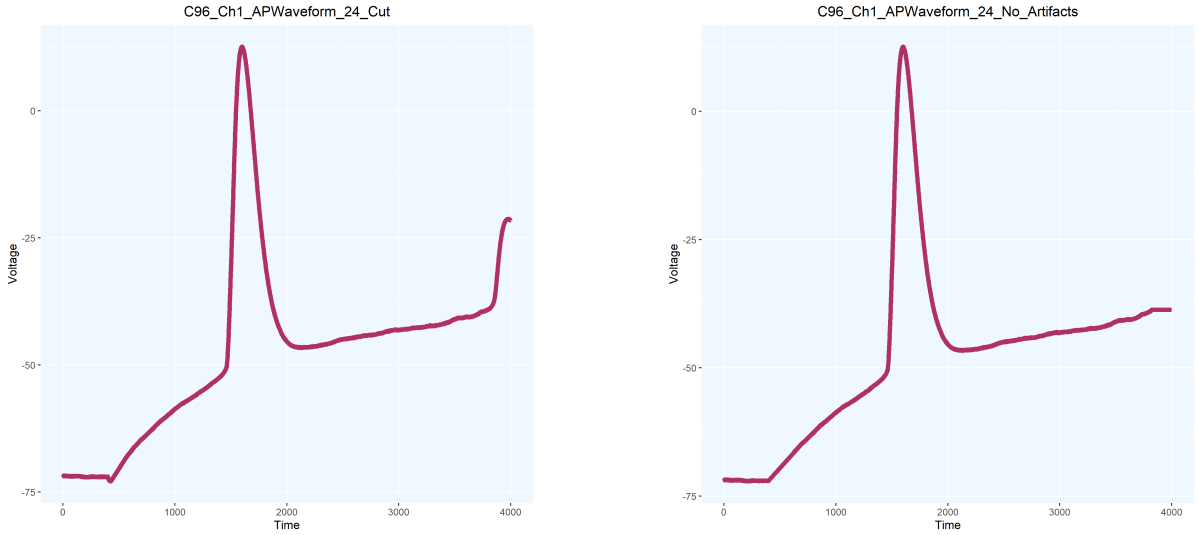


Figura 4.3: Señal con artefactos y sin artefactos

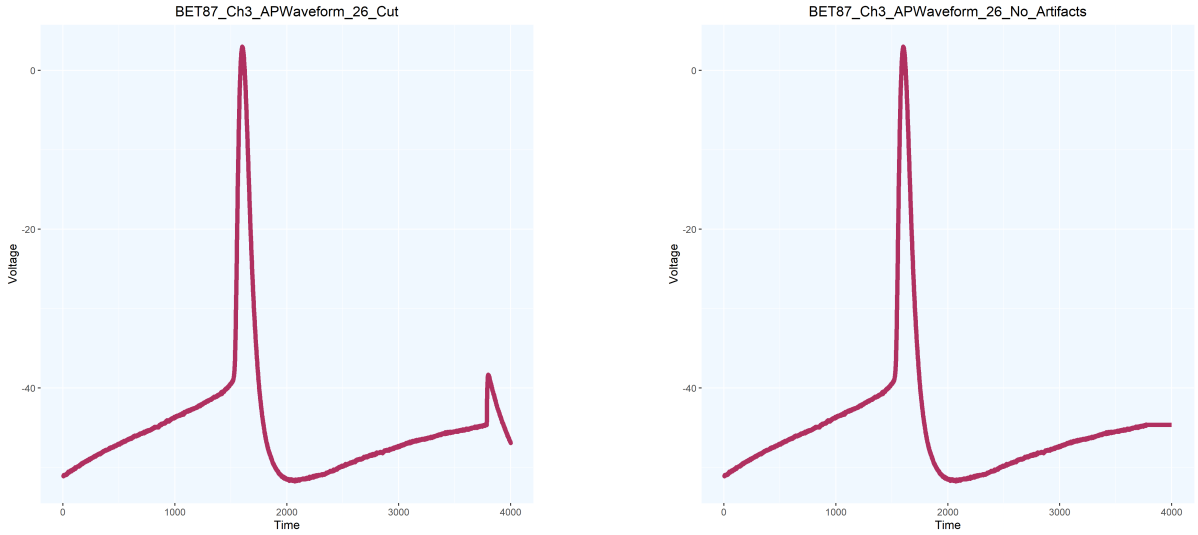


Figura 4.4: Señal con artefactos y sin artefactos

En las figuras 4.3 y 4.4 podemos ver dos ejemplos de señales con artefactos y cómo se ven después de eliminarlos. En el primer caso, vemos una señal con un *peak* detectado justo al final de la señal. Se ha eliminado simplemente continuando con los valores del potencial que se daban justo antes de producirse este pequeño *peak*. En el segundo caso, vemos un ejemplo habitual de artefacto, fácilmente eliminable uniendo los puntos iniciales y finales del *peak*.

4.3. Procesamiento de los datos

En este apartado se va a comentar cómo se ha ajustado el modelo FMM a las señales neuronales de las ratas que ya han sido preprocesadas.

4.3.1. Paquete FMM

El paquete FMM es una librería de R que permite ajustar modelos de Möbius con una o varias componentes a datos con una gran variedad de patrones rítmicos presentes en señales oscilatorias. Este paquete está disponible en CRAN [24].

Las principales funcionalidades de esta librería son: ajustar el modelo FMM de una o de varias componentes a unos datos, generar datos de forma artificial mediante el modelo FMM y visualizar gráficamente los resultados de un ajuste del modelo FMM a unos datos. Además, el paquete proporciona varios conjuntos de datos biológicos con los que se puede ajustar también el modelo FMM.

Todos las señales electrofisiológicas de las neuronas de ratas que solo tienen un *peak* y que han sido preprocesadas, son ajustadas mediante el modelo FMM, para ello, se utiliza la siguiente instrucción:

```
fitFMM(vData, nback, parallelize)
```

Siendo:

- `vData`: los datos observados de la señal oscilatoria que se quiere ajustar, en formato de vector numérico.
- `nback`. número que indica el número de componentes que se quieren ajustar.
- `parallelize`: valor booleano que permite el uso de todos los núcleos del ordenador para agilizar el ajuste del modelo FMM.

El objeto del ajuste del modelo contiene, entre otros, los valores ajustados por el modelo (`fittedValues`), los parámetros resultantes del ajuste del modelo (`M`, `A`, `alpha`, `beta` y `omega`), la suma de cuadrados del error (`SSE`) y el R^2 (`R2`).

Una vez que se han ajustado todas las señales, se almacena la información relevante de estos ajustes en un fichero `.csv`: el nombre de la señal, el valor de sus parámetros M, A, α, β y ω, la varianza explicada por el modelo, la tendencia de cada onda y la información relativa a la

expresión genética de cada una de las ondas. Para el caso del modelo FMM_3, se deben almacenar los valores mencionados previamente, por componentes.

Tras el ajuste del modelo, se procede a realizar una representación gráfica de este. Para ello se utiliza la instrucción:

```
plotFMM(objFMM, use_ggplot2 , components , textExtra )
```

Siendo:

- `objFMM`: el objeto resultante del ajuste del modelo FMM a los datos observados.
- `use_ggplot2`: valor booleano que permite elegir una representación mediante la librería de `ggplot2` o mediante la librería base de R.
- `components`: valor booleano que permite elegir entre dos tipos de representaciones:
  - `True`: permite dibujar las componentes ajustadas por el modelo FMM.
  - `False`: permite dibujar la composición de las ondas sobre los datos observados.
- `textExtra`: permite añadir un comentario en el título del gráfico. En este trabajo se utilizará el nombre de cada señal.

No se devuelve ningún objeto, simplemente se representa el ajuste del modelo FMM.

Otra de las instrucciones utilizadas de esta librería es:

```
generateFMM(M, A , alpha , beta, omega, from, to, length.out, plot, outvalues,
    sigmaNoise)
```

Esta instrucción permite simular datos del modelo FMM, definiendo sus parámetros M, A, α, β y ω. El significado del resto de parámetros es:

- `from`: valor numérico que indica el valor inicial de los datos.
- `to`: valor numérico que indica el valor final de los datos.
- `length.out`: valor numérico que indica la longitud del vector de datos que se desea simular.
- `plot`: valor booleano que indica si queremos hacer una representación automática de los datos generados.
- `outvalues`: valor lógico para indicar si queremos devolver el vector numérico generado.

- `sigmaNoise`: valor numérico que contiene la desviación estándar de ruido gaussiano que se desea añadir.

Nos devuelve una lista compuesta por: un vector numérico con los datos simulados (`y`), un vector numérico con los datos del tiempo en los que se realiza la simulación (`t`) y una lista con los parámetros de entrada M, A, α, β y ω (`input`).

4.3.2. Algoritmo de asignación de ondas con el modelo FMM_3

El ajuste del modelo FMM_3 a los datos observados, asigna las ondas en función de la variabilidad explicada, es decir, la primera onda será siempre la que más variabilidad explique, y la tercera, la que menos. Por lo tanto, tras el ajuste de cada modelo FMM_3, debemos hacer una reasignación de los valores de los parámetros presentes en las tres ondas. Nos interesa mantener la primera onda: aquella que más variabilidad recoge, pero queremos hacer que la tercera componente sea siempre la que ajuste la tendencia, debido a que resulta de un menor interés para el estudio de las características de las ondas. Consideramos la existencia de tendencia al encontrar una diferencia de potencial mayor a 5 mV entre el primer y el último instante de la señal. Resulta fundamental para el estudio posterior de los parámetros del modelo que todas las componentes ajusten el mismo tipo de características de la onda. En la figura 4.5 se muestra un ejemplo de dos señales con (izquierda) y sin tendencia (derecha).

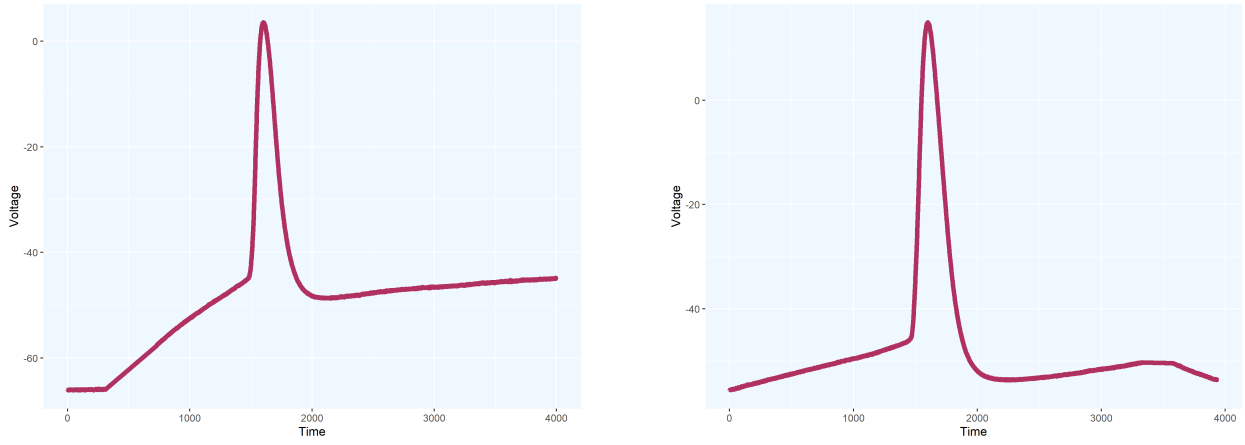


Figura 4.5: Ejemplo de dos señales con (izquierda) y sin (derecha) tendencia

Para la reasignación de ondas, en primer lugar, debemos mirar si la onda tiene tendencia. En caso afirmativo, se busca el parámetro α con un valor próximo a π. Esta será la componente que está ajustando la tendencia, asignándose a la tercera componente. Cuando nos encontramos

con una onda que no tiene tendencia, debemos asignar como segunda componente a la onda que cumpla el siguiente criterio:

$$\text{onda } 2 = argmin(1 - cos(\alpha_1 - \alpha_i)), \forall i \in \{2, 3\} \tag{4.2}$$

Una vez se han asignado correctamente las ondas respecto a los dos criterios que acaban de mencionarse, se procede a representar gráficamente las tres ondas ajustadas y reasignadas y se almacena correctamente el valor de los parámetros del modelo para su estudio posterior.

Capítulo 5

Resultados

En este capítulo se comentan los resultados del ajuste de las señales neuronales procesadas previamente mediante el modelo FMM_1 y mediante el modelo FMM_3. Se comparan los resultados de estos dos modelos. También se realiza una caracterización de los diferentes parámetros del modelo FMM_3 y se aplican técnicas de aprendizaje supervisado para la clasificación de los distintos tipos de genes de acuerdo con los resultados obtenidos del ajuste de este modelo.

5.1. Ajuste del modelo FMM_1

Tras el la selección de las señales de interés y de el preprocesamiento de estas, procedemos a realizar el ajuste de un modelo FMM_1 (modelo FMM con una única componente). El ajuste de este modelo ya ha sido descrito en el capítulo 3, concretamente en la ecuación 3.6. La librería utilizada también ha sido comentada en el capítulo 4.

En las figuras 5.1 y 5.2 podemos ver el ejemplo de cuatro señales ajustadas por el modelo FMM_1. En el caso de la señal de la izquierda de la figura 5.1 vemos que incluso con una alta tendencia, el modelo es capaz de explicar con tan solo 5 parámetros, una variabilidad del 92 %.

En las otras tres figuras vemos que con una sola componente ya se explica una variabilidad mayor del 95 % ya que la señal se ajusta casi a la perfección con los datos observados, a pesar de tener formas diferentes. Con tan solo cinco parámetros, el modelo demuestra ser muy flexible a la hora de ajustar este tipo de datos.

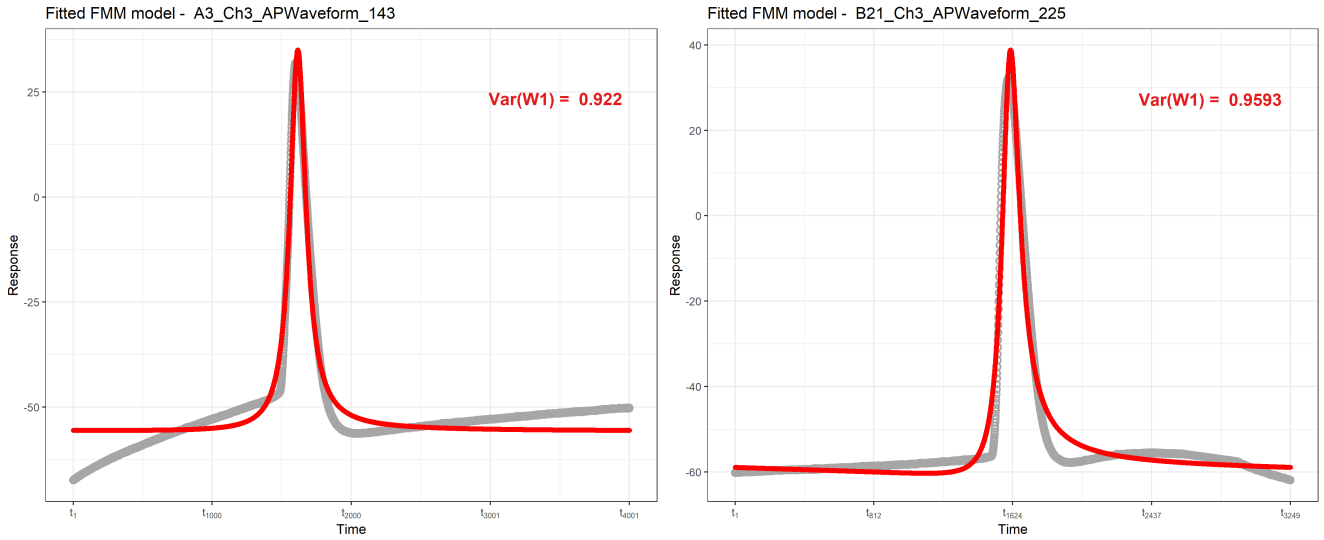


Figura 5.1: Aplicación del modelo FMM_1

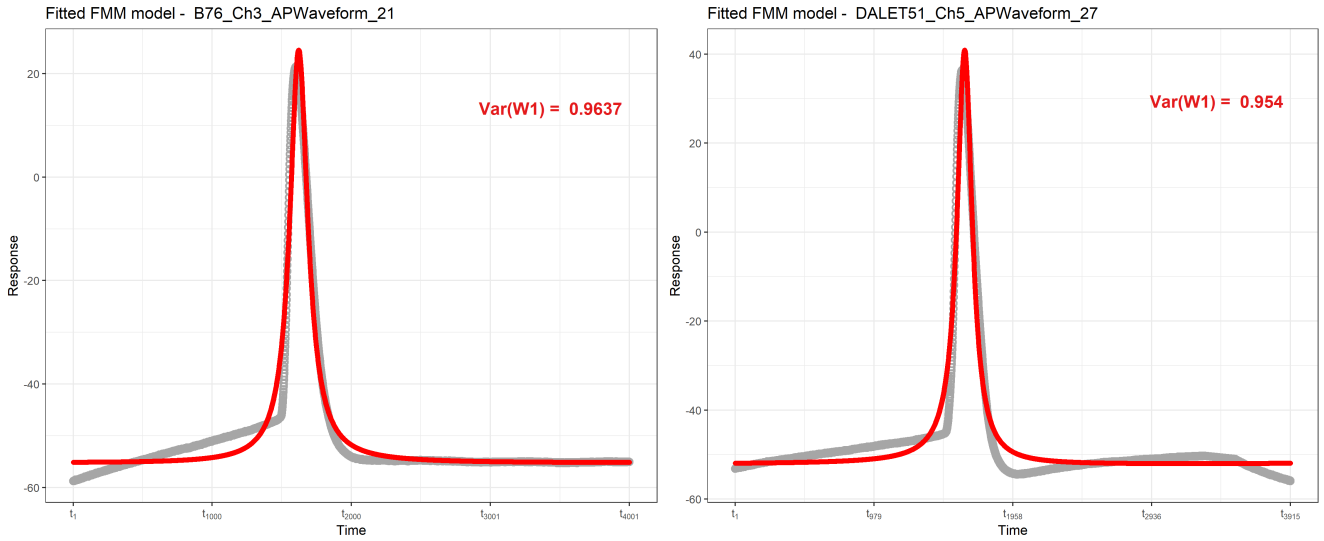


Figura 5.2: Aplicación del modelo FMM_1

5.2. Ajuste del modelo FMM_3

Tras el la selección de las señales de interés y del preprocesamiento de estas, procedemos a realizar el ajuste de un modelo FMM_3 (modelo FMM con tres componentes).

A continuación se mostrarán unos ejemplos de señales ajustadas mediante el modelo FMM_3. Para cada señal se mostrarán los datos observados junto con las tres ondas que ajustan el modelo FMM_3, siendo la onda roja, la que explica una mayor variabilidad de la onda, la onda azul, la segunda que caracteriza a la señal, y la onda verde, es la que recoge la tendencia de la señal. También se mostrará otro gráfico con los datos observados (gris) frente la composición de las tres ondas (rojo).

En la figura 5.3 podemos ver que los datos que se están ajustando tienen tendencia y que efectivamente, la tercera componente, la verde, es la que está ajustando la tendencia. En las figuras 5.4, 5.5 y 5.6 no existe tendencia, por eso se elige la segunda componente como ya se ha comentado antes, la onda con un α más cercano al de la primera componente. Podemos ver que en los cuatro casos, la variabilidad explicada por la primera componente ya es muy alta, en ningún caso bajando del 91 %. La segunda y tercera componente ajusta muy poca variabilidad, pero si nos fijamos en la composición de las tres ondas, podemos ver en todos los casos que se ajusta casi a la perfección con los datos.

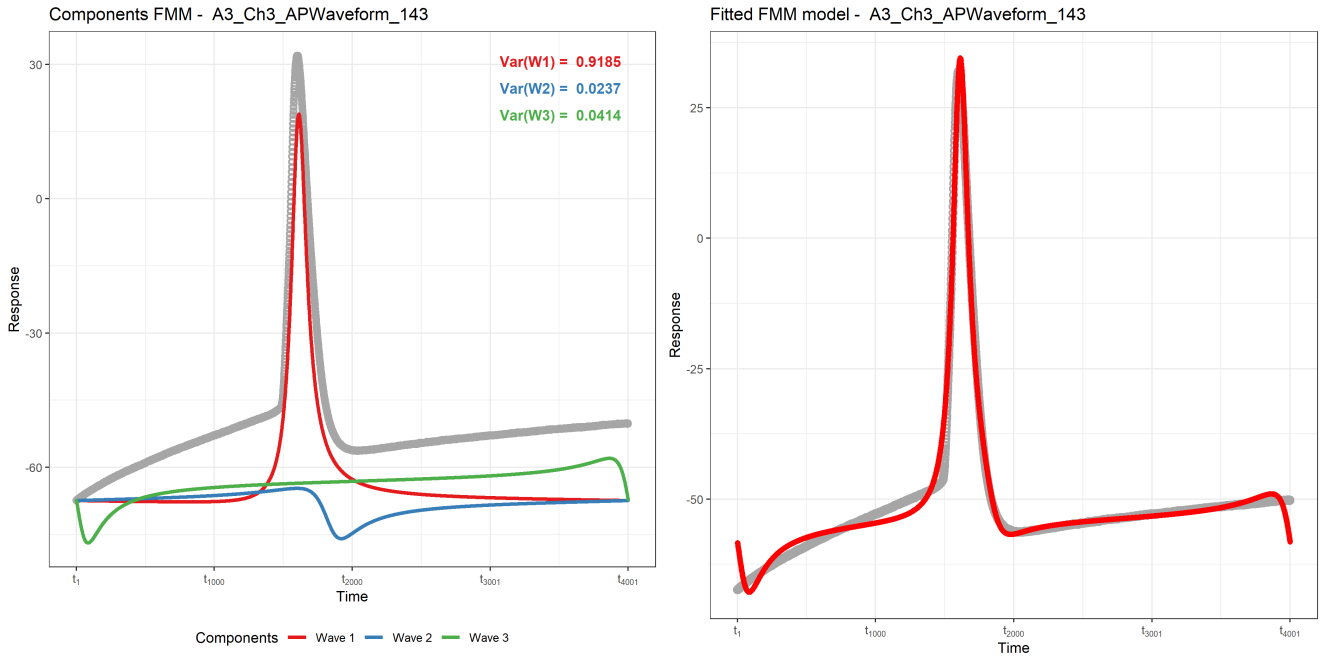


Figura 5.3: Aplicación del modelo FMM_3

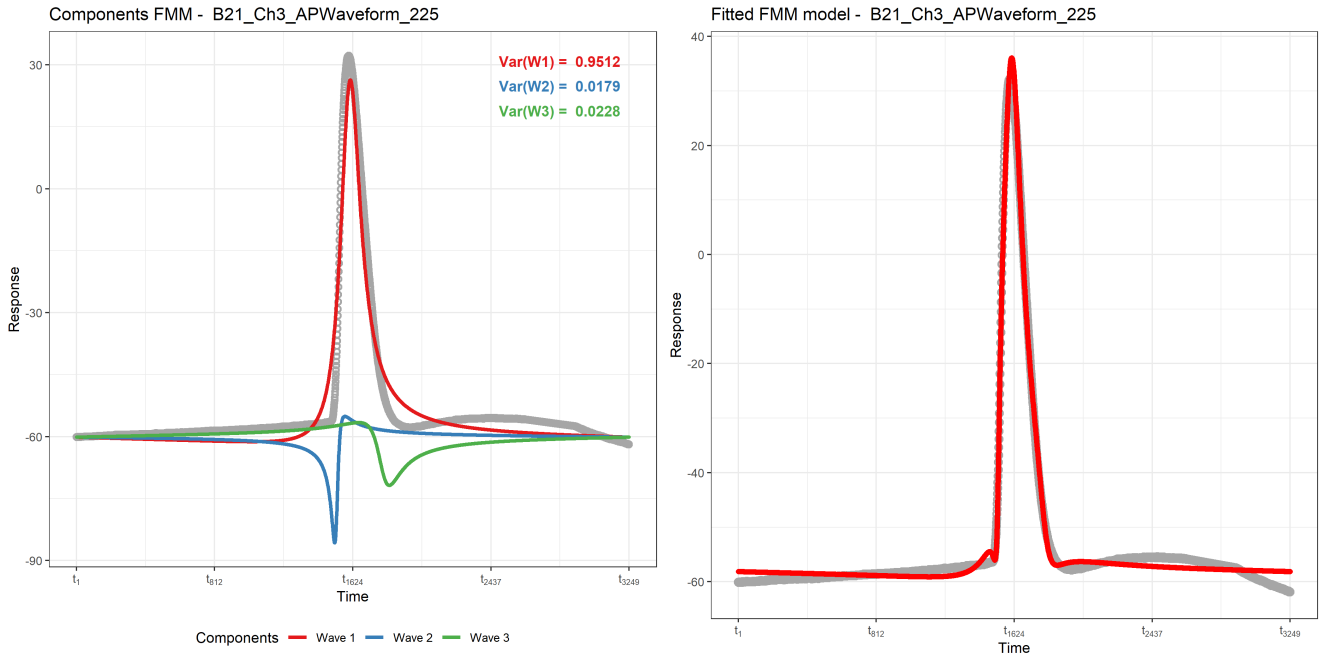


Figura 5.4: Aplicación del modelo FMM_3

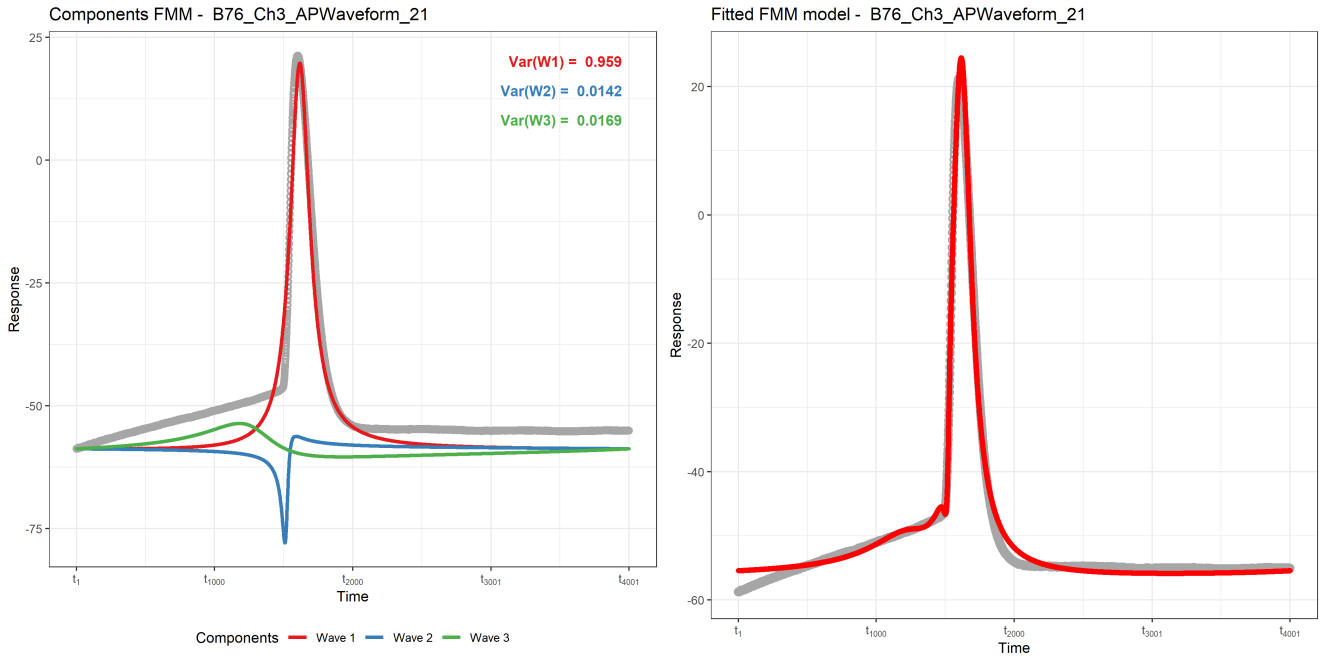


Figura 5.5: Aplicación del modelo FMM_3

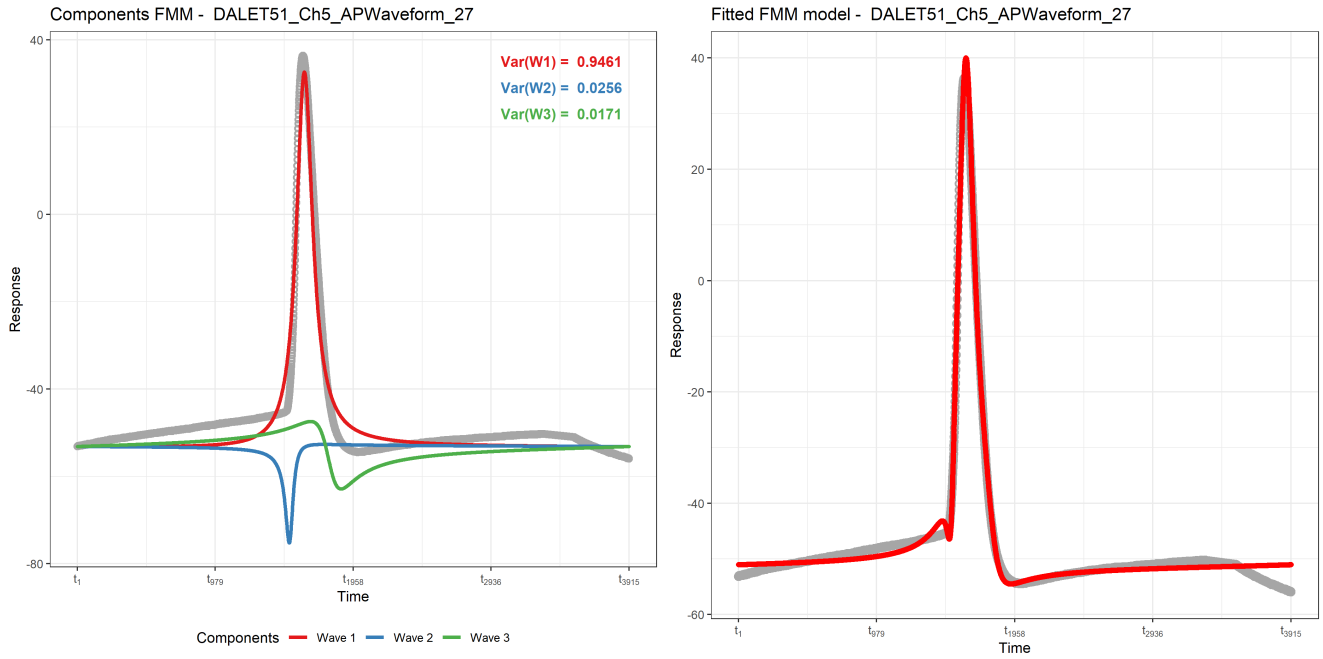


Figura 5.6: Aplicación del modelo FMM_3

5.3. Comparación de modelos

Vamos a estudiar la media y la mediana del R^2 de los modelos FMM_1 y FMM_3. Además, en el caso del modelo FMM_3, estudiaremos estas métricas para cada componente.

	R^2
Media	0.912
Mediana	0.938

Tabla 5.1: Media y mediana del R^2 del modelo FMM_1

	R^2_1	R^2_2	R^2_3	R^2_{global}
Media	0.906	0.025	0.052	0.983
Mediana	0.930	0.024	0.026	0.983

Tabla 5.2: Media y mediana del R^2 del modelo FMM_3

Los resultados del ajuste de las señales son muy buenos, caracterizan de manera bastante precisa las ondas, como podemos ver en las tablas 5.1 5.2. Para el modelo con una componente, vemos que se obtiene una media de un 91 % para el R^2 y una mediana del 93 %. Si nos fijamos en el modelo de las tres componentes, estos datos suben. Podemos apreciar que la tanto la media como la mediana del R^2 de las tres ondas es de 0.98, es decir, el modelo caracteriza casi por completo las señales. Además, con la primera componente ya se consigue un R^2 de media del 90 %, y más de la mitad de las señales tienen un R^2 mayor del 93 %, por lo que, como se ve, la segunda y tercera componente apenas recogen variabilidad. Podemos apreciar que la media y la mediana para la segunda componente son prácticamente las mismas, a diferencia de la tercera componente, en la que la media del R^2 duplica a la mediana. Esto se debe a que las ondas que ajustan la tendencia, recogen más variabilidad que las que no tienen tendencia, haciendo subir la media.

Media				
	R^2_1	R^2_2	R^2_3	R^2_{global}
PV	0.946	0.015	0.024	0.985
CB	0.900	0.022	0.061	0.983
SOM	0.923	0.027	0.033	0.984
VIP	0.902	0.023	0.058	0.983
NPY	0.937	0.028	0.021	0.986
CCK	0.876	0.023	0.083	0.982
CR	0.883	0.026	0.073	0.981

Mediana				
	R^2_1	R^2_2	R^2_3	R^2_{global}
PV	0.946	0.015	0.024	0.985
CB	0.939	0.020	0.023	0.983
SOM	0.934	0.027	0.022	0.983
VIP	0.92	0.022	0.043	0.983
NPY	0.944	0.027	0.020	0.989
CCK	0.905	0.023	0.053	0.983
CR	0.872	0.026	0.076	0.98

Tabla 5.3: Media y mediana del R^2 para cada tipo de gen

En la tabla 5.3 podemos ver estas mismas métricas para el modelo FMM_3, medidas por cada gen activado al producirse el *peak* en la neurona. Podemos ver que tanto la media como la mediana del R^2, en todos los casos, es mayor de 0.98, alcanzando el valor más alto en el caso del gen NPY. Los genes cuya media y mediana del R^2 en la primera componente son más pequeñas, son CCK y CR. Podemos ver que para estos mismos genes, la tercera componente explica de media una mayor variabilidad, por lo que se intuye que en estos genes, hay tendencia.

5.4. Caracterización de los diferentes tipos paramétricos

En este apartado se realiza un estudio gráfico de las posibles diferencias en los parámetros obtenidos al aplicar el modelo FMM_3 a las distintas señales cerebrales de las ratas, según el tipo de gen que se se active cuando la neurona recibe un impulso eléctrico. En la tabla 5.4 podemos ver el número de señales que tienen cada tipo de gen activado o no activado.

	PV	CB	SOM	VIP	NPY	CCK	CR
Presencia	1 (0.95 %)	18 (17.14 %)	29 (27.62 %)	18 (17.14 %)	4 (3.81 %)	36 (34.29 %)	12 (11.43 %)
Ausencia	104 (99.05 %)	87 (82.86 %)	76 (72.38 %)	87 (82.86 %)	101 (96.19 %)	69 (65.71 %)	93 (88.57 %)

Tabla 5.4: Proporción de ondas en las que se activa cada gen

En primer lugar, vamos a estudiar el perfil mediano de cada uno de los genes, es decir, para cada tipo de gen activado, vamos a calcular la mediana de todos los parámetros de todas las señales y vamos a simular unos datos con estos nuevos parámetros. De esta forma se obtienen siete nuevas ondas que vamos a representar y a comparar.

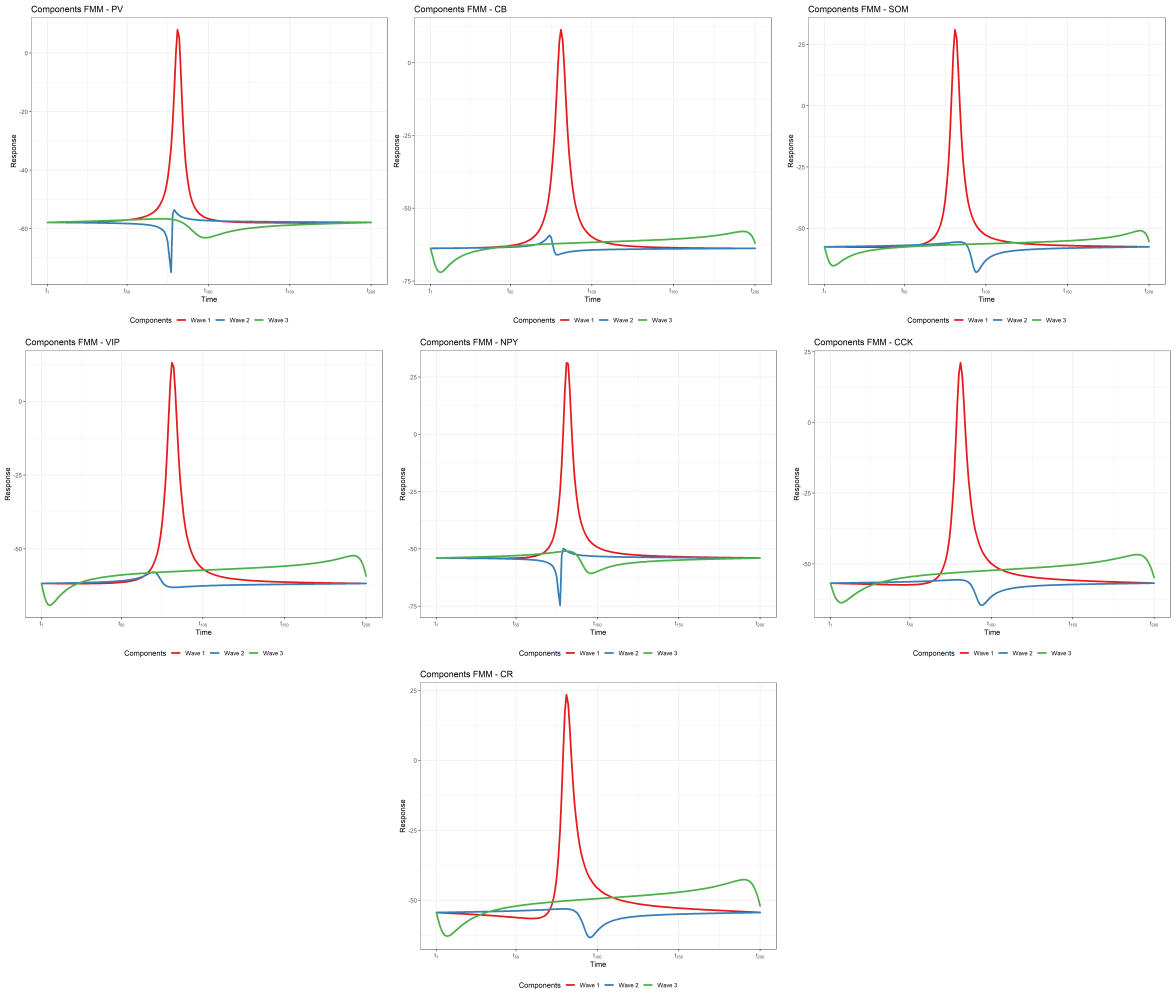


Figura 5.7: Perfil mediano de cada tipo de gen ajustado mediante el modelo FMM_3

En la figura 5.7 podemos ver la representación del perfil mediano de las tres ondas de cada gen por separado. A simple vista ya podemos observar que en los genes de PV y NPY, no existe tendencia, porque sus valores de α_3 son muy diferentes a los del resto, en los que se ve que claramente la tercera onda (verde) tiene un valor de α próximo a π. También se puede apreciar que en la segunda onda, el valor de ω parece mayor en ciertos casos, como en el gen CR, mientras que otros genes, por ejemplo PV, presentan unas ondas con mucho más apuntamiento. A simple vista, parece que la primera onda para todos los genes, es muy parecida.

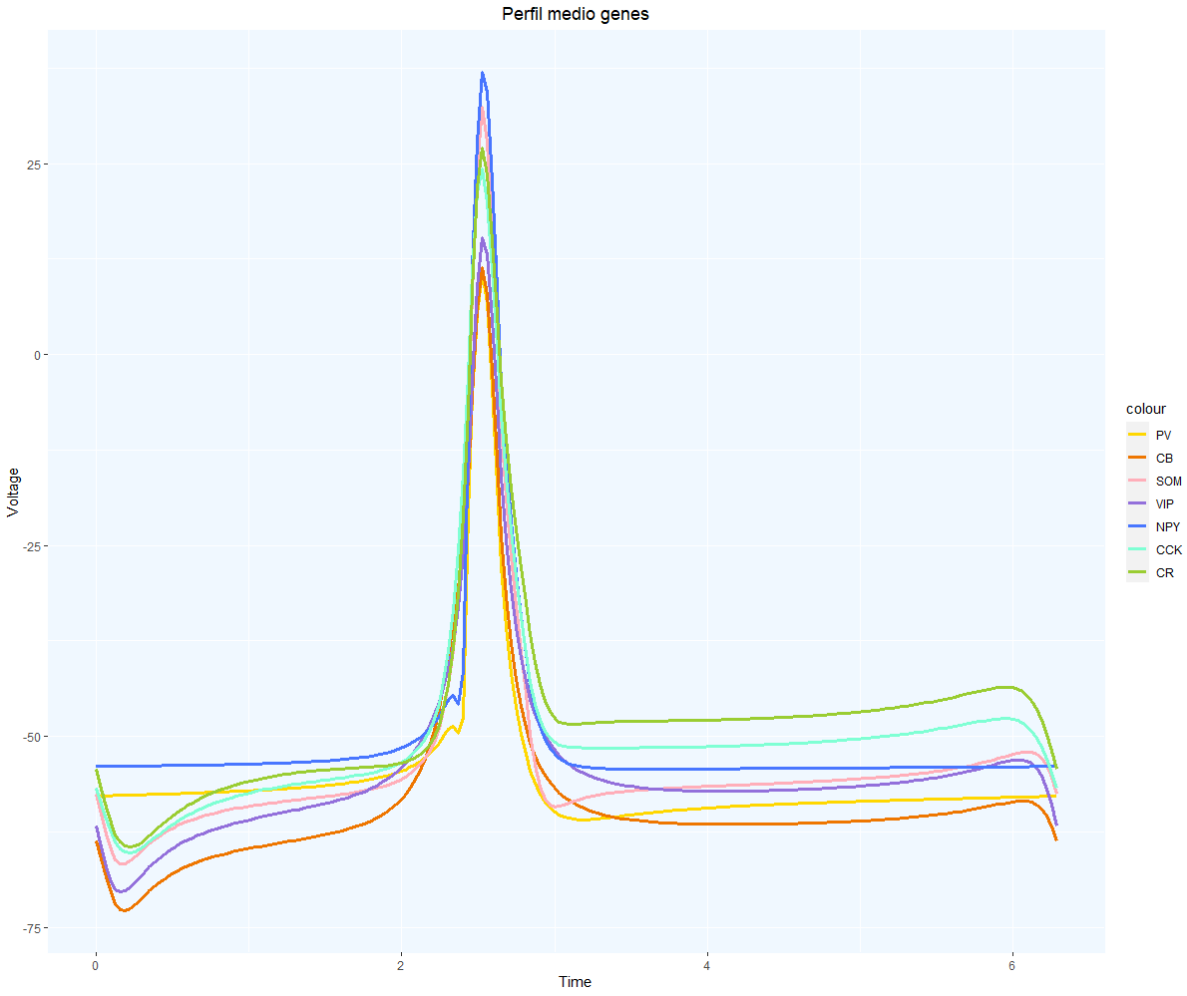


Figura 5.8: Perfil mediano de los genes

En la figura 5.8 podemos ver que se ha representado la suma de las tres ondas de cada tipo de gen, distinguiéndolos por colores diferentes. Podemos ver que el valor de α_1 es prácticamente el mismo en todos los casos, pero también es significativo ver que el parámetro A_1, la amplitud del *peak*, es diferente según el tipo de gen, siendo NPY el de mayor amplitud y CB el que menos.

Además, podemos fijarnos que el apuntamiento de las ondas que tienen un menor A_1 es mayor, es decir ω_1.

A continuación, vamos a estudiar estos parámetros que parecen distinguirse entre los diferentes genes, mediante boxplots. Vamos a representar tan solo los boxplots de seis de los siete genes, debido a que, el gen PV, tan solo se encuentra presente en una de las señales de nuestro conjunto de datos.

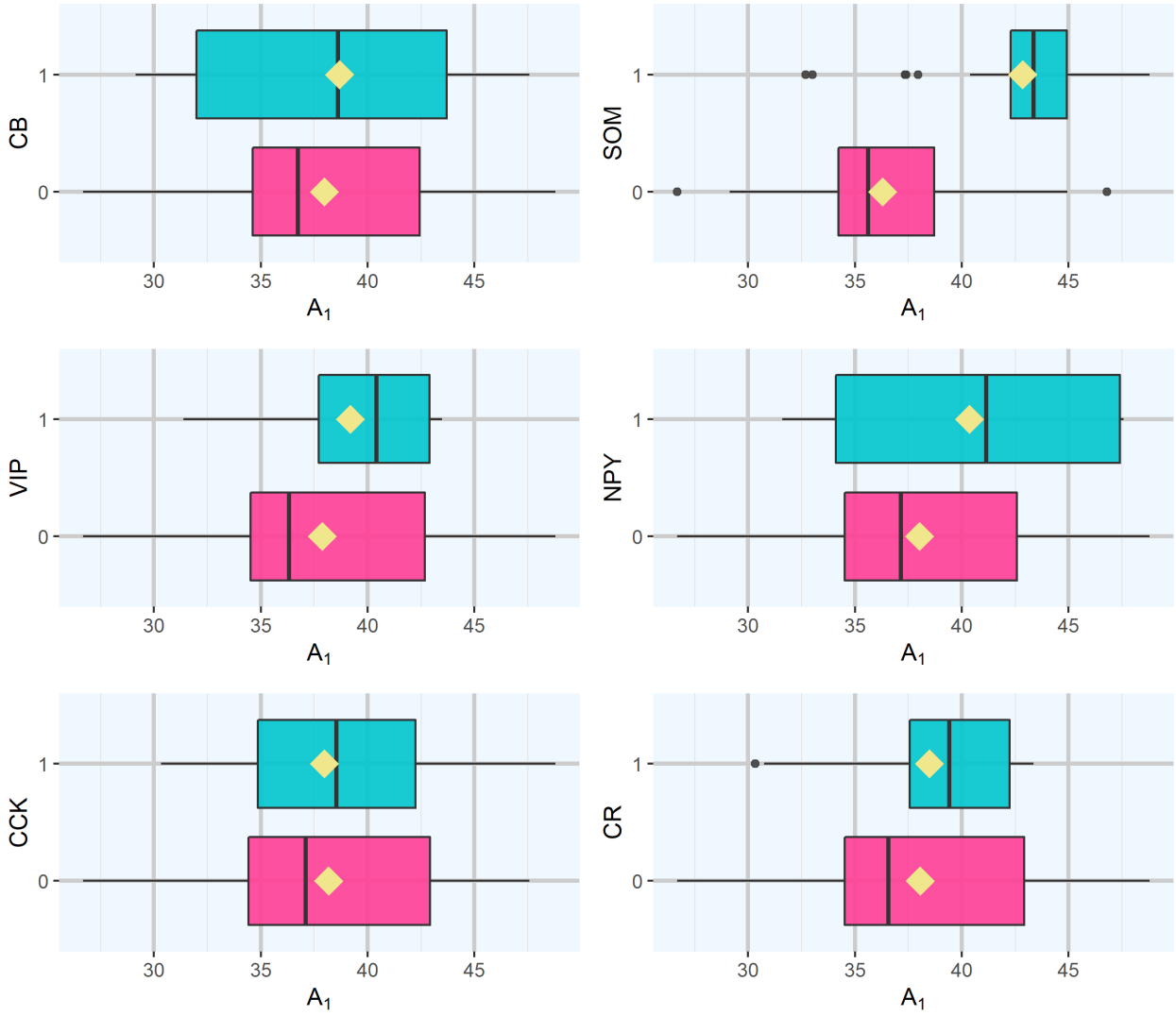


Figura 5.9: Boxplots del parámetro A_1 según la activación de los genes (azul) o no (rosa) y la media de cada uno (amarillo)

En la figura 5.9 podemos apreciar como sí existe una diferencia bastante marcada entre las

ondas en las que se activa el gen SOM, teniendo unos valores de A_1 bastante más altos que en las ondas en las que no se activa. También vemos que en otros casos como el gen CB, apenas hay diferencias, sus medias son muy similares y las cajas de los boxplots coinciden. En casos como en el gen NPY, parece que las medianas son bastante diferentes, aunque no tanto como en el caso de SOM, pero las cajas de los boxplots prácticamente coinciden, en parte debido a la alta varianza que existe en las ondas con presencia del gen.

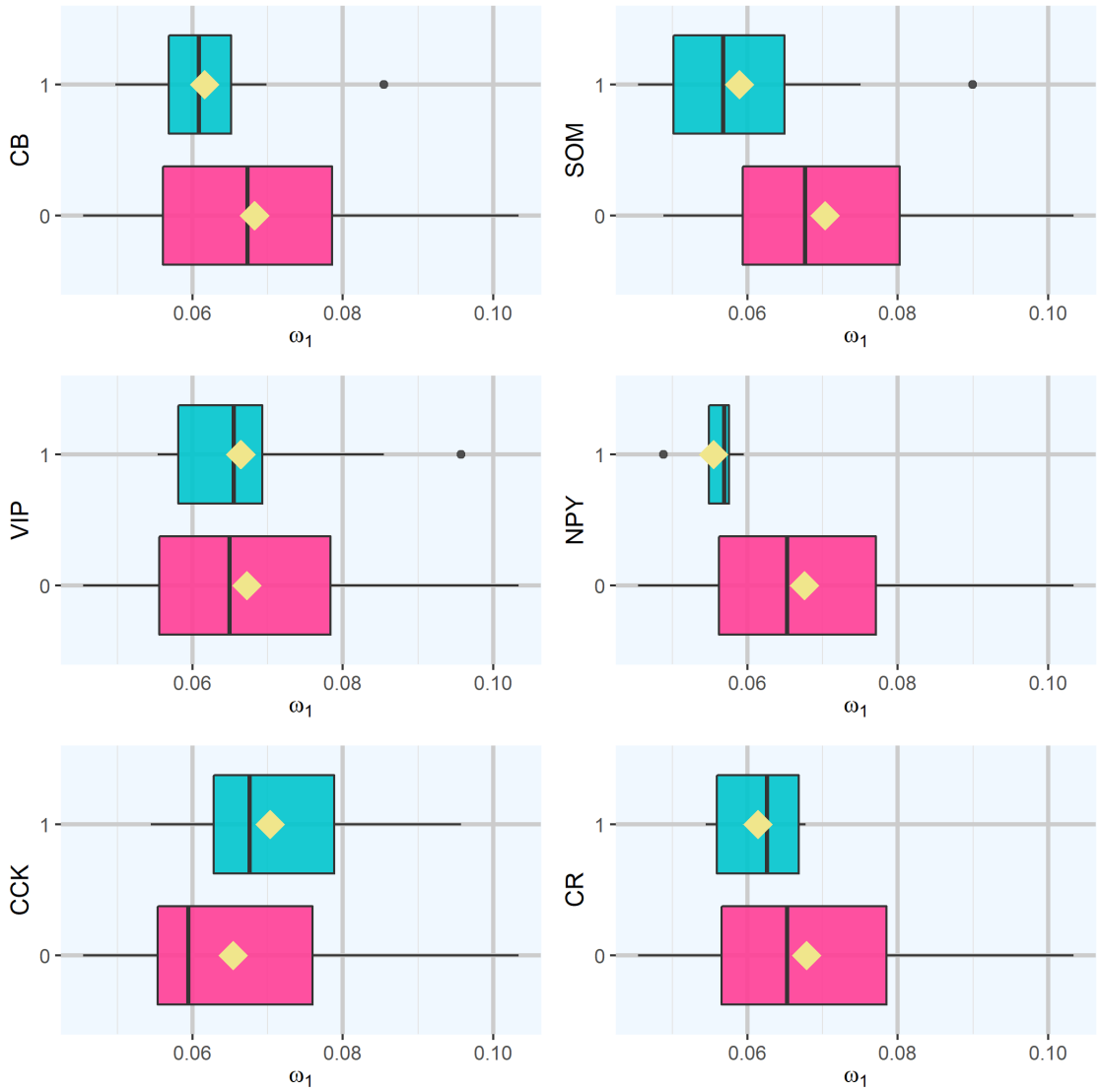


Figura 5.10: Boxplots del parámetro ω_1 según la activación de los genes (azul) o no (rosa) y la media de cada uno (amarillo)

En la figura 5.10 se muestra una representación similar a la que vimos en la figura 5.9, pero para el parámetro ω_1. En este caso se ha decidido eliminar dos outliers que distorsionaban las escalas de los gráficos. Podemos ver que en el gen SOM vuelve a haber una diferencia destacable entre los parámetros ω_1 en la presencia o ausencia del gen. Lo mismo ocurre para el gen NPY, las señales con presencia de este gen tienen valores de ω_1 menores que las señales en las que no se manifiesta este gen. Para el resto de genes, aparentemente no existe una diferencia remarcable entre la presencia o ausencia de cada gen.

El parámetro β_2 es un parámetro circular, por lo que sus valores se encuentran en un espacio circular, con valores en un intervalo $[0, 2\pi]$, no en el espacio euclídeo, como es habitual. Por lo tanto, no podemos hacer las representaciones usuales, como el boxplot utilizado para los parámetros A_1 y ω_1, sino que vamos a representar su seno frente a su coseno.

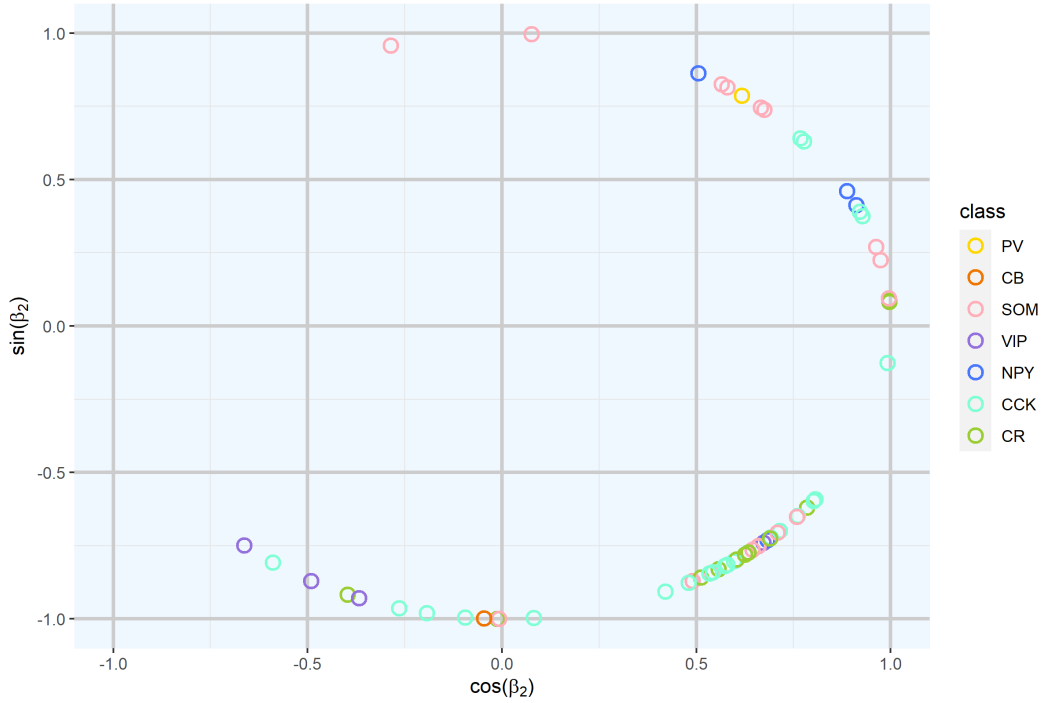


Figura 5.11: Representación circular de β_2

En la figura 5.11 vemos, como acabamos de comentar, el coseno de β_2 en el eje X, y el seno en el eje Y. En este gráfico podemos apreciar cómo de parecido es el parámetro β_2 para cada gen. Se aprecia, por ejemplo, que la mayoría de observaciones del gen NPY se encuentran bastante lejos de las observaciones en las que se manifiesta el gen VIP. Lo mismo ocurre con las observaciones en las que se manifiesta el gen SOM, parecen, en la mayoría de los casos, bastante alejadas de las observaciones en las que se manifiesta el gen CCK. Para la única observación del gen PV parece

que su parámetro β_2 es muy similar a observaciones del gen SOM. En el caso de las señales en las que se manifiesta el gen CR y CB, no parece que exista una diferencia clara del resto de genes ni tampoco un parecido claro con un único gen.

5.5. Discriminación de señales según tipo genético

En esta sección nos proponemos resolver problemas binarios de clasificación supervisada de señales para cada uno de los grupos genéticos, identificando los parámetros que más influyen en la clasificación. No estudiaremos la presencia de los genes PV y NPY debido a que el tamaño muestral es muy bajo, contando con tan solo una y cuatro instancias respectivamente. Para el ajuste de los modelos se utiliza la librería `caret` [25]. Esta librería nos permite de forma sencilla ajustar múltiples modelos mediante validación cruzada. Previo al ajuste de los modelos, se han estandarizado las variables. Los modelos que se van a estudiar son: discriminante lineal (LDA), árboles de clasificación, *random forest* y *support vector machine* (SVM).

Los clasificadores LDA y SVM son métodos basados en distancias, por lo que asumen que las variables tienen distribuciones euclideas. Las variables β_1, β_2 y β_3 no pueden ser estudiadas directamente con estos métodos debido a que se miden en el espacio circular. Se reemplazan por las siguientes variables: $\sin(\beta_1)$, $\cos(\beta_1)$, $\sin(\beta_2)$, $\cos(\beta_2)$, $\sin(\beta_3)$, $\cos(\beta_3)$. En el caso de los árboles de clasificación y *random forest*, sí que se pueden utilizar β_1, β_2 y β_3, debido a que estos métodos discretizan las variables, por lo que son capaces de explotar correctamente las distribuciones de estos parámetros, a pesar que sus valores estén entre 0 y 2π. Además de las variables ya mencionadas, los cuatro métodos utilizarán también las variables: M, A_1, A_2, A_3, $\text{dist}(\alpha_1, \alpha_2)$, $\text{dist}(\alpha_1, \alpha_3)$, $\text{dist}(\alpha_2, \alpha_3)$, ω_1, ω_2 y ω_3.

	LDA	Árboles	Random Forest	SVM
CB	**0.7905** (0.8495, 0.3333)	**0.7905** (0.8218, 0)	**0.8857** (0.8866, 0.875)	**0.9333** (0.9444, 0.9667)
SOM	**0.8667** (0.8875, 0.8)	**0.8476** (0.8571, 0.8095)	**0.8952** (0.9114, 0.8462)	**0.9048** (0.8929, 0.9524)
VIP	**0.8** (0.8511, 0.3636)	**0.8095** (0.8526, 0.4)	**0.8857** (0.8866, 0.875)	**0.9333** (0.9255, 1)
CCK	**0.8** (0.8158, 0.7586)	**0.7905** (0.8615, 0.675)	**0.8952** (0.8919, 0.9032)	**0.9048** (0.8734, 1)
CR	**0.8762** (0.8922, 0.3333)	**0.8857** (0.8857, 0)	**0.9619** (0.9588, 1)	**0.9619** (0.9588, 1)

Tabla 5.5: Resultados de discriminación por clasificador y gen: **tasa de acierto**, (sensibilidad, especifidad)

En la tabla 5.5 se presentan los resultados del ajuste de los cuatro clasificadores para cinco genes: CB, SOM, VIP, CCK y CR. Se muestran los resultados de la tasa de acierto, la sensibilidad y la especifidad de cada modelo. Todos los modelos han sido entrenados mediante validación cruzada de 10 particiones y todos los resultados, incluyendo las matrices de confusión, son el resultado del promedio de estas 10 particiones.

Para el análisis de discriminante lineal se ha utilizado como probabilidades a priori, la distribución de las clases, es decir, el número de instancias que tienen o no un gen activado. Como podemos ver en la tabla 5.5, LDA junto con los árboles de clasificación son los métodos que nos ofrecen peores resultados, debido en parte a que son los métodos más sencillos. La mejor tasa de acierto del modelo LDA es la de los genes SOM y CR, con una sensibilidad similar en ambos casos, sin embargo, la especifidad para la clasificación del gen CR es bastante peor, tan solo 0.33. Esto nos dice que el modelo no es tan bueno al calcular la probabilidad de que se active el gen, cuando realmente el gen se ha activado. Esto significa que el modelo tiende habitualmente a clasificar más instancias como si no se hubiera activado el gen. Para ver por qué se produce esto, vamos a fijarnos en sus matrices de confusión.

En la matriz de confusión para el gen CR de la figura 5.12 podemos ver que la mayoría de señales que sí que tienen activado este gen, están siendo clasificadas como si no estuviera activado. Esto explica la baja especifidad que se consigue para este gen. Si lo comparamos con la matriz de confusión para el gen SOM, se puede ver que en este caso el porcentaje de señales que se clasifican correctamente como gen sí activado es mucho mayor, de esta forma la especifidad es mucho más alta.

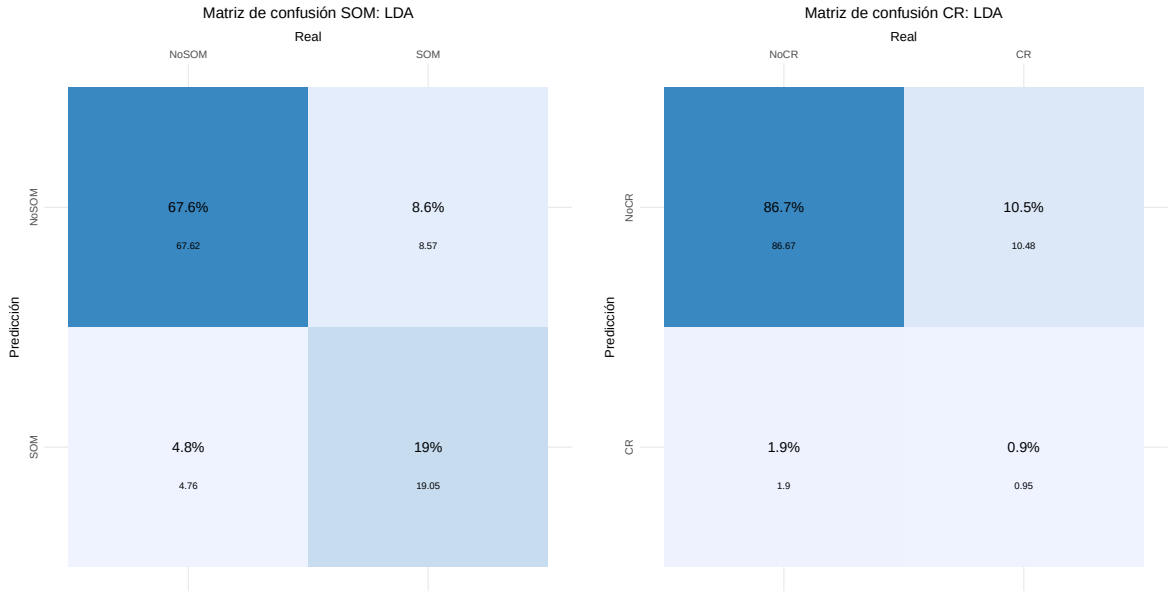


Figura 5.12: Matrices de confusión para el discriminante lineal de los genes SOM y CR

Fijándonos en los coeficientes de los discriminantes lineales para cada tipo de gen, vemos que para los cinco tipos de genes, las dos variables que más influencia tienen en los modelos son: $\text{dist}(\alpha_1, \alpha_3)$ y $\text{dist}(\alpha_2, \alpha_3)$.

Volviendo a la tabla 5.5, nos fijamos en los resultados de los árboles de clasificación. Vemos que las tasas de acierto de los cinco clasificadores tienen valores entre 79 % y 88 %. La sensibilidad de todos los clasificadores también es alta, entre un 82 % y un 88 %, sin embargo, la especifidad es más problemática, en el caso del gen CB y del gen CR, es 0. Si nos fijamos en las matrices de confusión de la figura 5.13 vemos el motivo. En la matriz de confusión para el gen CB, vemos que la celda en la que las señales con el gen CB activado y clasificado como tal, está vacía, lo que significa que todas las señales que se deberían clasificar como si tuvieran este gen activado, se están clasificando erróneamente. En la matriz de confusión del gen CR, vemos que dos celdas están vacías, las relacionadas como predichas por el clasificador como si tuvieran este gen activado, lo que significa que el árbol de clasificación predice todas las señales como si no tuvieran el gen CR activado.

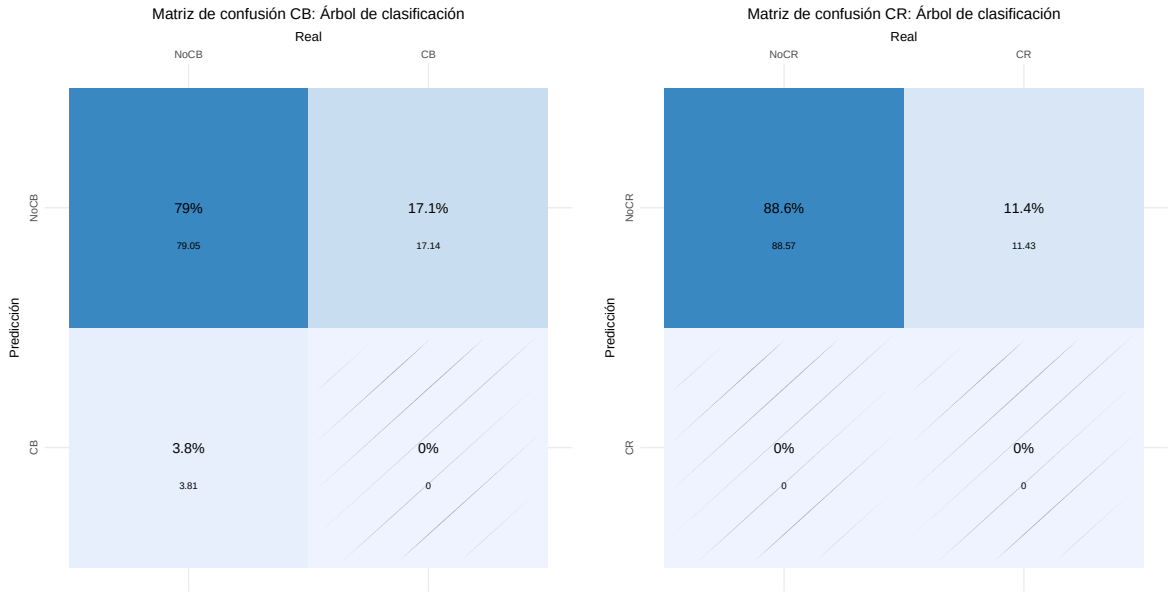


Figura 5.13: Matrices de confusión de los árboles de clasificación

Para el estudio de los árboles como clasificadores, además de las métricas estudiadas para el resto de clasificadores, vamos a visualizar los árboles, para ver cuáles han sido las variables que se han utilizado para decidir a qué clase asignar cada instancia. En la figura 5.14 podemos ver los árboles para los clasificadores de los genes SOM y CCK. Se observa que las variables que se han utilizado para decidir a qué clase asignar las instancias son A_1 y A_2 para el primer caso, y β_1, ω_1 y β_3 para el segundo. No se representan los árboles de clasificación de los genes CB, VIP y

CR debido a que los árboles tan solo tienen un nodo raíz: A_1 para los dos primeros y β_1 para el tercero.

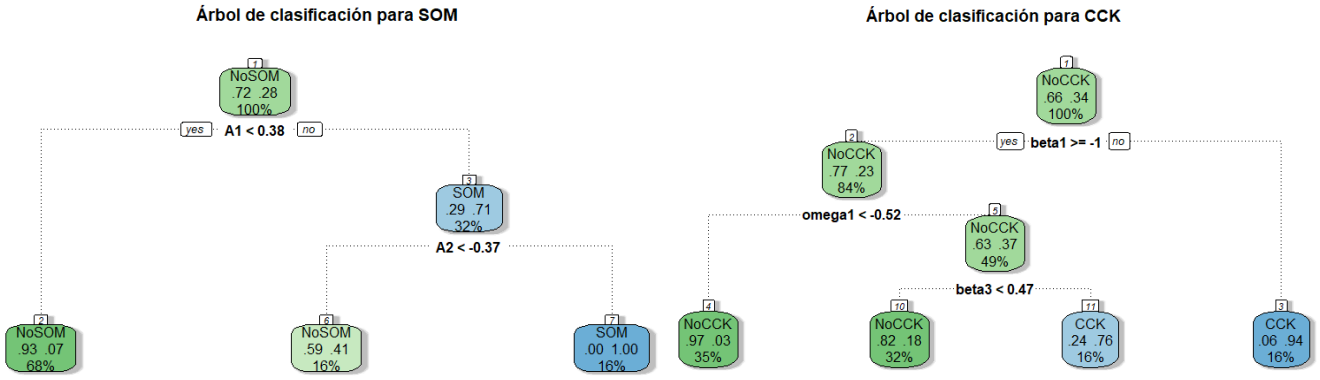


Figura 5.14: Árboles de clasificación

Los métodos de *random forest* y *support vector machine* son bastante más complejos que los métodos de discriminante lineal y árboles de clasificación. Por este motivo, se va a hacer un ajuste de los parámetros de interés de los modelos, eligiendo los que den mejores resultados. Para los *random forest* se han probado diferentes valores para los parámetros de número de árboles (`ntree`) y número de características por árbol (`mtry`). Para el parámetro de `ntree` se ha probado con los valores: 5, 10, 15, 20, 25 y 30. Para el parámetro `mtry` se ha probado con los valores: 2, 3, 4, 5 y 6.

	Random Forest	SVM
CB	ntree: 30 mtry: 3	kernel: polinomial degree: 2, scale: 1, C: 1
SOM	ntree: 30 mtry: 5	kernel: radial sigma: 1, C: 1
VIP	ntree: 20 mtry: 2	kernel: radial sigma: 1, C: 10
CCK	ntree: 25 mtry: 3	kernel: radial sigma: 1, C: 10
CR	ntree: 15 mtry: 5	kernel: polinomial degree: 4, scale: 0.1, C: 0.1

Tabla 5.6: Parámetros de los métodos *random forest* y SVM

En el caso de los *support vector machine* se han probado dos tipos diferentes de *kernel*: polinomial y radial (RBF). Para el caso de *kernel* polinomial, se prueban diferentes valores para el grado

del polinomio (`degree`), la escala (`scale`), y el coste (`C`). Para el grado del polinomio probaremos con: 2, 3, 4 y 5, para la escala probaremos: 0.01, 0.1 y 1 y para el coste: 0.1, 1, 10 y 100. En el caso de *kernel* radial, probaremos los parámetros sigma (`sigma`) y coste (`C`). Para sigma probaremos los siguientes valores: 1, 10 y 100; para el coste probaremos con: 0.1, 1, 10 y 100. En la tabla 5.6 podemos ver los parámetros que se han elegido para cada uno de los clasificadores, tanto para *random forest* como para SVM.

Los resultados obtenidos por los *random forest* para los distintos tipos de genes se encuentran en la tabla 5.5. Podemos ver que todos los clasificadores consiguen una tasa de acierto muy cercana al 90 %, incluso llegando a 96 % el clasificador para el gen CR. La sensibilidad en bastante alta, estando en todos los genes entre un 88 % y un 95 %. La gran mejora respecto a los clasificadores anteriores LDA y árboles de clasificación es que la especifidad mejora mucho en todos los casos, siempre superior al 84 % y llegando a alcanzar el 100 % en el caso del gen CR.

El método de *random forest* nos permite ver cuáles han sido las variables que más han influido en las decisiones de los árboles de clasificación que lo componen. En la tabla 5.7 se muestran cuáles han sido las tres variables más importantes para cada tipo de gen. Como ya se había observado en el análisis estadístico de los parámetros del modelo FMM_3, se aprecia que las variables A_1, ω_1 y β_2 son relevantes en la clasificación de los diferentes grupos genéticos.

	CB	**SOM**	**VIP**	**CCK**	**CR**
1º	A_2	A_1	A_1	ω_1	β_1
2º	β_2	A_2	β_1	β_1	A_3
3º	A_1	ω_1	β_2	β_2	β_2

Tabla 5.7: Variables más importantes en *random forest*

El método de *support vector machine*, al igual que el método de *random forest*, es más complejo que LDA y los árboles de clasificación. También es un método considerado de "caja negra". Se han ajustado los parámetros que se han comentado previamente y se han utilizado los que mejores resultados han generado. Estos parámetros se encuentran en la tabla 5.6.

En la tabla 5.5 podemos ver los resultados del método de SVM para cada tipo de gen, utilizando los parámetros de la tabla 5.6. Si los comparamos con los del resto de clasificadores, se observa que se obtienen los mejores resultados, con una tasa de acierto que en ningún caso baja del 90 %. La tasa de acierto más alta se alcanza con el clasificador para el gen CR. La sensibilidad alcanzada para todos los genes es muy alta, y la especifidad también, siendo incluso en tres casos del 100 %.

Capítulo 6

Conclusiones y líneas futuras

6.1. Conclusiones

En este trabajo se han estudiado diferentes modelos oscilatorios para ajustar señales electrofisiológicas de neuronas de ratas. Todas estas señales han sido extraídas de la base de datos abiertos del proyecto Blue Brain.

El ajuste de estas señales mediante diferentes modelos oscilatorios: Cosinor, Fourier y FMM nos ha permitido ver las ventajas de este último frente al resto. El modelo FMM monocomponente ha obtenido un R^2 de media de un 91 % y el modelo FMM con tres componentes ha conseguido elevar esta cifra al 98 %, adaptándose casi a la perfección a los datos, mientras que los modelos Cosinor y de Fourier han explicado un porcentaje de variabilidad mucho menor.

La discriminación de señales según su tipo genético mediante diferentes métodos de clasificación supervisada ha obtenido muy buenos resultados, destacando los modelos de *support vector machine* y *random forest*. De esta forma, los parámetros del modelo FMM permiten relacionar las señales electrofisiológicas con los diferentes tipos genéticos de las células y con diferentes trastornos asociadas a estos, debido a que la presencia o abundancia de ciertas expresiones genéticas en determinadas células está relacionado con enfermedades neurológicas como la esquizofrenia.

Los resultados obtenidos en este trabajo y las características del modelo FMM, especialmente, la interpretabilidad biológica de sus parámetros, nos hacen ver el gran potencial de este modelo en un futuro esperemos no muy lejano, en el que gracias a este se pueda ampliar el conocimiento actual del cerebro, en especial, del cerebro humano.

6.2. Líneas futuras

En las señales electrofisiológicas de neuronas de ratas, al desencadenarse un *peak*, muchas veces se activan varios genes. Quedan por estudiar las posibles relaciones entre los distintos tipos genéticos activados simultáneamente y los parámetros del modelo FMM resultantes tras el ajuste de las señales.

En la base de datos del proyecto Blue Brain quedan muchas señales por estudiar, únicamente se han analizado las que tienen un único *peak*. Fácilmente se pueden segmentar las señales con dos o tres *peaks*, ampliando así los datos utilizados para la clasificación genética. Con estos datos ampliados, podría elaborarse una clasificación circular de las expresiones genéticas que se han estudiado, como ya se ha realizado en otros proyectos como [14].

Existen otras muchas bases de datos con señales neuronales que podrían estudiarse a través del modelo FMM con el propósito de diagnosticar enfermedades, como puede ser la epilepsia. También podrían estudiarse bases de datos con características genéticas y morfológicas, en vez de neuronales, para probar el funcionamiento del modelo FMM con este nuevo tipo de señales.

Siglas

AP *Action Potential*. XI, 4, 5, 32

CB *Calbindin*. 31, 44, 46, 48, 50, 51, 52, 53, 54

CCK *Cholecystokinin*. 31, 44, 49, 50, 51, 52, 53, 54

CR *Calretinin*. 31, 44, 46, 50, 51, 52, 53, 54

FMM *Frequency Modulated Möbius*. 1, 2, 14, 17, 19, 21, 29, 31, 35, 36, 39, 55, 56

FMM$_1$ *Frequency Modulated Möbius con una componente*. IX, XII, XIII, 13, 14, 17, 18, 20, 21, 22, 39, 40, 43

FMM$_3$ *Frequency Modulated Möbius con 3 componentes*. IX, XI, XII, XIII, 22, 23, 24, 29, 36, 37, 39, 40, 41, 42, 43, 44, 45, 54

FMM$_m$ *Frequency Modulated Möbius con m componentes*. 19, 20

NPY *Neuropeptide Y*. 31, 44, 46, 48, 49, 50

PV *Parvalbumin positive*. 31, 44, 46, 47, 49, 50

SOM *Somatostatin*. 31, 44, 48, 49, 50, 51, 52, 53, 54

VIP *Vasoactive Intestinal Peptide*. 31, 44, 49, 50, 51, 52, 53, 54

Bibliografía

[1] Carter, M., & Shieh, J. (2015). Guide to Research Techniques in Neuroscience (Second Edition). Academic Press.

[2] Zeng, H., & Sanes, J. (2017). Neuronal cell-type classification: challenges, opportunities and the path forward. Nature Reviews Neuroscience, 18(9), 530-546.

[3] Ramón y Cajal, S. (1899 - 1904). Histología del sistema nervioso del hombre y de los vertebrados.

[4] Gouwens, N., Sorensen, S., Baftizadeh, F., Budzillo, A., Lee, B., & Jarsky, T. et al. (2020). Integrated Morphoelectric and Transcriptomic Classification of Cortical GABAergic Cells. Cell, 183(4), 935-953.e19.

[5] Gonchar, Y., & Wang, Q. (2008). Multiple distinct subtypes of GABAergic neurons in mouse visual cortex identified by triple immunostaining. Frontiers In Neuroanatomy, 1.

[6] Tasic, B., Menon, V., Nguyen, T., Kim, T., Jarsky, T., & Yao, Z. et al. (2016). Adult mouse cortical cell taxonomy revealed by single cell transcriptomics. Nature Neuroscience, 19(2), 335-346.

[7] Disorders, F., Pankevich, D., Davis, M., & Altevogt, B. (2011). Glutamate-related biomarkers in drug development for disorders of the nervous system. Washington, D.C.: National Academies Press.

[8] Tremblay, R., Lee, S., & Rudy, B. (2016). GABAergic Interneurons in the Neocortex: From Cellular Properties to Circuits. Neuron, 91(2), 260-292.

[9] Ferguson, B., & Gao, W. (2018). PV Interneurons: Critical Regulators of E/I Balance for Prefrontal Cortex-Dependent Behavior and Psychiatric Disorders. Frontiers In Neural Circuits, 12.

[10] Song, Y., Yoon, J., & Lee, S. (2021). The role of neuropeptide somatostatin in the brain and its application in treating neurological disorders. Experimental &Amp; Molecular Medicine, 53(3), 328-338.

[11] Ono, D., Honma, K., & Honma, S. (2021). Roles of Neuropeptides, VIP and AVP, in the Mammalian Central Circadian Clock. Frontiers In Neuroscience, 15.

[12] A Cellular Taxonomy of the Mouse Visual Cortex: Allen Institute for Brain Science. (2022). Recuperado el 19 de Abril de 2022, de http://casestudies.brain-map.org/celltax#section_explorea

[13] Cornelissen, G. (2014). Cosinor-based rhythmometry. Theoretical Biology And Medical Modelling, 11(1).

[14] Rodríguez-Collado, A., & Rueda, C. (2019). Electrophysiological and Transcriptomic Features Reveal a Circular Taxonomy of Cortical Neurons. Frontiers In Human Neuroscience, 15.

[15] Rueda, C., Rodriguez-Collado, A., & Larriba, Y. (2021). A Novel Wave Decomposition for Oscillatory Signals. IEEE Transactions On Signal Processing, 69, 960-972.

[16] Rueda, C., Larriba, Y., & Peddada, S. (2019). Frequency Modulated Möbius Model Accurately Predicts Rhythmic Signals in Biological and Physical Sciences. Scientific Reports, 9(1).

[17] Rueda, C., Larriba, Y., & Peddada, S. (2019). Supplementary Material for "Frequency Modulated Möbius Model Accurately Predicts Rhythmic Signals in Biological and Physical Sciences". Scientific Reports, 9(1).

[18] Rodríguez-Collado, A., & Rueda, C. (2021). A simple parametric representation of the Hodgkin-Huxley model. PLOS ONE, 16(7), e0254152.

[19] I. Fernández, A. Rodríguez-Collado, Y. Larriba, A. Lamela, C. Canedo & C. Rueda.(2021). FMM: An R Package for Modeling Rhythmic Patterns in Oscillatory Systems.

[20] Hastie, T., Tibshirani, R., & Friedman, J. (2001). The elements of statistical learning (Second Edition). Springer.

[21] Bishop, C. (2016). Pattern Recognition and Machine Learning. Springer.

[22] Neuronal Electrical Recordings - Blue Brain Portal. Recuperado el 8 de Marzo de 2022, de https://portal.bluebrain.epfl.ch/resources/data/neuronal-electrical/

[23] Jefferis, G. (2017). IgorR. Recuperado el 9 de Marzo de 2022, de https://github.com/cran/IgorR

[24] Fernandez, I., Rodriguez-Collado, A., Larriba, Y., Lamela, A., Canedo, C., & Rueda, C. (2021). CRAN - Package FMM. Recuperado el 8 de Mayo de 2022, de: https://cran.r-project.org/web/packages/FMM/index.html

[25] Kuhn M., Win J., Weston S., Williams A., Keefer C., Engelhardt A., Cooper T., Mayer Z., Kenkel B., Benesty M., Lescarbeau R., Ziem A., Scrucca L., Tang Y., Candan C. & Hunt T. (2022) CRAN - Package caret. Recuperado el 19 de Mayo de 2022 de: https://cran.r-project.org/web/packages/caret/index.html