

UNIVERSIDAD DE ALCALÁ

Escuela Politécnica Superior

**GRADO EN INGENIERÍA ELECTRÓNICA DE
COMUNICACIONES**

Trabajo Fin de Grado

**Detección y análisis de eventos
postsinápticos obtenidos de registros
intracelulares de neuronas espinales**

Autor: David Sánchez Salvador

Tutores: Sira Elena Palazuelos Cagigas y José Luis Martín Sánchez

Tribunal:

Presidente: D. Rafael Barea Navarro

Vocal 1: D. Raúl Mateos Gil

Vocal 2: D. José Luis Martín Sánchez

CALIFICACIÓN:

FECHA:

Agradecimientos

Ante todo agradezco a mis padres el sacrificio que han hecho durante años para darme la posibilidad de estudiar lo que realmente me gustaba, sin dejar de apoyarme en ningún momento.

A mi familia, siempre orgullosa de lo que quería hacer en mi vida, ojalá pudiera compartirlo con todos.

A mis amigos, que tanto me animaron a seguir, que me hicieron creer que podía lograrlo, sin ellos no lo hubiera conseguido.

A mis hermanos, que me ayudaron a recordar que una noche, ante un fuego tranquilo, cuando la claridad desaparece y los ruidos se apagan hice una Promesa.

A Iván y a mis tutores José Luis y Sira, les agradezco de corazón todo su trabajo y dedicación por hacer posible este proyecto.

Muchas gracias a todos.

Índice de contenidos

Introducción.....	13
Capítulo 1	
Estado del arte	17
1.1 Algoritmos basados en análisis de correlación.....	18
1.2 Algoritmos basados en umbrales.....	20
1.3 Algoritmos basados en plantillas.....	21
1.3.1 Análisis mediante plantillas fijas	21
1.3.2 Análisis mediante plantillas escaladas óptimamente	22
1.3.3 Ajuste ponderado de mínimos cuadrados con varias plantillas	27
1.4 Algoritmo de detección automática por pasos (FAUST)	34
1.5 Algoritmo de deconvolución ciega	38
1.6 Software para el análisis de señales biológicas: <i>Spike2</i>	43
Capítulo 2	
Desarrollo de la herramienta <i>PS_Analysis</i> (PostSinaptic Analysis).....	45
2.1 Data handling.....	48
2.2 Plotting.....	49
2.3 Preprocess.....	52
2.4 Reconstruction	53
2.5 Detection.....	55
2.6 Classification.....	58
Capítulo 3	
Resultados	59
3.1 Evaluación de las diferentes configuraciones de <i>PS_Analysis</i>	59
3.2 Comparativa <i>PS_Analysis</i> con la detección visual y <i>Spike2</i>	68
3.3 Histogramas de los eventos detectados.....	72
Capítulo 4	
Conclusiones y trabajos futuros	73
Pliego de condiciones	75
Presupuesto	77
Manual de usuario de <i>PS_Analysis</i>	79
Bibliografía	87
Anexo	89

Lista de acrónimos

mEPSC	miniature Excitatory Postsynaptic Current
sEPSC	spontaneous Excitatory Postsynaptic Current
mIPSC	miniature Inhibitory Postsynaptic Current
sIPSC	spontaneous Inhibitory Postsynaptic Current
EPSP	Excitatory Postsynaptic Potential
IPSP	Inhibitory Postsynaptic Potential
OLS	Ordinary Least Squares
WLS	Weighted Least Squares
SSE	Sum of Squared Errors
RSD	Residual Standard Deviation
NS	Normalized Scale

Resumen

Las neuronas integradas en circuitos neuronales generan para su comunicación eventos postsinápticos. Los electrofisiólogos estudian los eventos generados para poder conocer cómo se relacionan entre ellas y cómo se comportan ante ciertos factores externos, como por ejemplo la presencia de fármacos.

Cuando se hace un registro intracelular, se captan los eventos de un conjunto de sinapsis de una neurona y otras próximas. Para que esta información sea de utilidad, es necesario identificar los eventos postsinápticos generados frente al ruido presente en el registro. Además se trata de caracterizar dichos evento para poder analizar el efecto de los experimentos.

El objetivo de este proyecto es desarrollar una aplicación que permita a los electrofisiólogos detectar, clasificar y extraer distintas características de registros intracelulares procedentes de distintos circuitos neuronales. Se requiere que sea una herramienta fiable, con bajo porcentaje de resultados erróneos, además de rápida y automática en comparación con las técnicas que utilizan los electrofisiólogos.

Para la implementación de la herramienta ha sido necesario un período de investigación sobre los algoritmos o técnicas más utilizados en el análisis de registros intracelulares. Tras este estudio se han incluido en el programa herramientas basadas en algunas de las técnicas estudiadas en la investigación.

Finalmente, tras la realización de este proyecto, se ha conseguido desarrollar un programa, *PS_Analysis*, que automatiza y agiliza en gran medida el proceso de detección y análisis de eventos postsinápticos en registros intracelulares.

Abstract

Integrated neurons in neural circuits generate postsynaptic events for their communication. Electrophysiologists study events generated in order to know how they relate to each other and how they behave to certain external factors, such as the presence of drugs.

When an intracellular recording is done, the events of a set of synapses of a neuron and other nearby are captured. For this information to be useful, it is necessary to identify the postsynaptic events generated against noise present in the record. Furthermore it is about to characterize these events analyzing the effect of the experiments.

The target of this project is to develop an application that allows electrophysiologists to detect, classify and extract some features of intracellular recordings from different neural circuits. It is required to be a reliable tool, with low percentage of wrong results, also fast and automatic compared to the techniques used by electrophysiologists.

For the implementation of the tool was necessary a period of research on algorithms and techniques commonly used in the analysis of intracellular recordings. After this study, has been included in the program the tools based on some of the techniques studied in the research.

Finally, after the performance of this project, has been developed a program, *PS_Analysis*, that automates and greatly speed up the process of detecting and analyzing postsynaptic events in intracellular recordings.

Introducción

La médula espinal constituye el primer centro de integración de la información procedente de los receptores sensoriales. La información procedente de receptores activados por estímulos nocivos también se procesa en gran medida en la médula espinal y modificando la actividad neuronal se puede influir en la percepción del dolor como ocurre con algunos analgésicos y anestésicos que actúan a nivel espinal. Por eso el estudio del funcionamiento de la médula espinal es interesante en el campo del dolor.

La herramienta desarrollada en este proyecto servirá de apoyo a los electrofisiólogos en el estudio de la médula espinal. El procesamiento de la información a nivel espinal depende de las conexiones sinápticas y de cómo están formados los circuitos neuronales. Un circuito neuronal es un conjunto de múltiples conexiones sinápticas ordenadas que se produce como resultado de la unión entre diferentes neuronas.

El registro electrofisiológico de las neuronas integradas en un circuito a menudo revela una gran cantidad de eventos postsinápticos espontáneos debidos a la activación de diversos elementos presinápticos. El análisis de estos eventos puede permitir inferir el funcionamiento de un circuito en base a la información que recibe uno de sus elementos. Para adquirir estos registros se utiliza la técnica de fijación de voltaje, o técnica de *patch-clamp*. Esta técnica se detalla en [1] y es comúnmente empleada para estudiar el funcionamiento de las sinapsis.

La duración y la amplitud de estos eventos espontáneos reflejan la cinética de los receptores ionotrópicos, que permiten la apertura de canales, para las moléculas neurotransmisoras y las tasas de liberación de esos neurotransmisores. Las diferencias en estas propiedades se pueden utilizar para identificar las clases específicas de entradas sinápticas de una neurona. Por otra parte, la forma de onda de los eventos representa la unidad básica o un evento cuántico, de las cuales las respuestas evocadas más grandes se componen. Para estimar con precisión y de forma fiable los parámetros de los eventos se requiere la grabación de cientos o miles de eventos durante períodos de muchos minutos, incluso horas.

Aunque la detección de estos eventos puede hacerse visualmente, la detección automática aumenta en gran medida el número de eventos que se pueden analizar y reduce el tiempo requerido, además de proporcionar una caracterización más completa de estos eventos. Además los algoritmos automatizados definen de forma inequívoca los criterios de selección, por lo que la detección y caracterización es más objetiva y reproducible.

Los eventos postsinápticos de los registros intracelulares pueden ser registrados en cualquier neurona (en corriente o en voltaje) y se dividen en espontáneos o miniatura y excitatorios o inhibitorios.

En la figura I, extraída de [2], podemos observar una representación de los distintos tipos, que se describen a continuación:

1. *spontaneous Postsynaptic Current, sPSC*: Eventos resultantes de un potencial de acción espontáneo en una neurona pre sináptica con la consecuente liberación de un número indeterminado de vesículas sinápticas.
2. *miniature Postsynaptic Current, mPSC*: Eventos procedentes de la liberación de una sola vesícula sináptica en ausencia de potenciales de acción, debido a la liberación estocástica, aleatoria. Ante este tipo de eventos se bloquean los potenciales de acción con un fármaco y tienen menor amplitud que los anteriores.
3. *Excitatory postsynaptic current, EPSC*: eventos que despolarizan la membrana y favorecen que la célula se dispare, aumentando la probabilidad de que se alcance el umbral para generar un potencial de acción.
4. *Inhibitory postsynaptic current, IPSC*: Eventos que hiperpolarizan la célula y dificultan la generación de potenciales de acción.

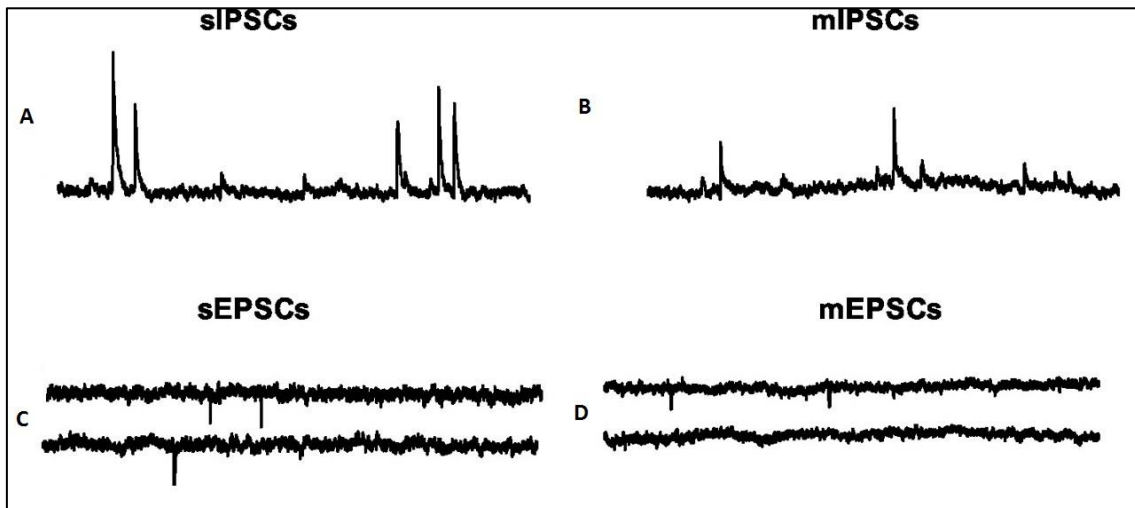


Figura I. Representación de los distintos eventos postsinápticos en corriente.
 (A) *spontaneous Inhibitory Postsynaptic Current*. (B) *miniature Inhibitory Postsynaptic Current*. (C) *spontaneous Excitatory Postsynaptic Current*. (D) *miniature Excitatory Postsynaptic Current*. Fuente: [2]

En este proyecto, de aplicación en el ámbito de la fisiología, se realizará el estudio detallado de diferentes algoritmos que permiten hacer un análisis automático y se diseñará e implementará una plataforma, que haciendo uso de dichos algoritmos, facilite al máximo y con los mejores resultados posibles el análisis de las señales registradas.

Los resultados se interpretarán posteriormente en el contexto del funcionamiento de la médula espinal. La intensidad y las características de estas respuestas vienen determinadas por las conexiones entre los distintos elementos que constituyen los circuitos espinales. Cambios en estas conexiones o en el funcionamiento de los contactos sinápticos a consecuencia de procesos de sensibilización central pueden ayudar a entender cómo estos circuitos se adaptan en situaciones de dolor crónico.

Adicionalmente se analizarán estos registros de células individuales para estudiar la posible presencia de ritmos y/o sincronía en la actividad subumbral evocada por la actividad sináptica y su relación con respuestas del circuito.

Capítulo 1

Estado del arte

En este capítulo se detallarán algunos de los algoritmos, automáticos y semiautomáticos, más utilizados para la detección de eventos en registros electrofisiológicos. Estos algoritmos se pueden dividir en cuatro categorías según las técnicas que implementen, aunque estas categorías no serán excluyentes sino que se apoyan sobre las anteriores:

- Algoritmos que buscan un evento a partir de umbrales de amplitud o a través de la primera derivada.
- Aquellos que comparan una plantilla, fija o escalable, con un registro transitorio en busca de una coincidencia óptima.
- Algoritmos que se basan en medidas de similitud, mediante correlaciones cruzadas, entre una base conocida y el conjunto de datos o los propios datos entre sí.
- Algoritmos basados en modelos estadísticos que mediante iteraciones buscan converger en una solución óptima.

El objetivo de estos algoritmos es identificar los eventos de interés que pueda haber en los registros neuronales. En estos registros encontraremos multitud de eventos, individuales y solapados, además del ruido de fondo y algunos artefactos de diferente origen. En general, estas técnicas tratan de encontrar características, como la amplitud de pico y la forma de las señales, con distintas distribuciones para los eventos y el ruido que permitan hacer una detección correcta.

1.1 Algoritmos basados en análisis de correlación

Inicialmente el análisis de datos biomédicos era realizado de forma manual pero poco a poco se fue incluyendo el uso de algoritmos en esta tarea, lo que agilizó el trabajo de los investigadores.

En [3] se describe el funcionamiento de uno de los primeros programas de ordenador para analizar respuestas intracelulares. Éste recibía los datos del programa de reconocimiento de la respuesta neuronal, analizaba los datos en términos de varios procedimientos estadísticos, y presentaba los resultados de una forma reducida, y era de esperar que fuera fiable.

El programa atendía a cuatro cuestiones:

- Intervalos de tiempo de los potenciales postsinápticos excitatorios reconocidos (EPSP), de los potenciales postsinápticos inhibitorios reconocidos (IPSP) y de los potenciales de acción.
- Probabilidad de que tras una de las anteriores respuestas neuronales se produzca un segundo tipo de respuesta dentro de un intervalo tiempo de retardo determinado, independientemente de que las respuestas sean contiguas.
- Histogramas de amplitud de los EPSP e IPSP reconocidos.
- Probabilidad de que dado un tipo particular de respuesta de un rango determinado de amplitud, se produzca un tipo concreto de respuesta de la misma, o diferente, amplitud dentro de un intervalo de tiempo de retardo determinado.

El programa de reconocimiento de eventos era capaz de reconocer estas respuestas con una gran precisión teniendo en cuenta la tecnología del momento. Proporcionando a la salida una lista con el tipo de evento, el instante de inicio, el tiempo de subida y de su amplitud.

Si el evento reconocido era un potencial de acción proporcionaba también su duración. El propósito de estos datos de salida era facilitar un análisis estadístico.

Una forma común de análisis estadístico preliminar, asociado con los datos de respuestas neuronales intracelulares, es contar el número de veces que una determinada variación se produce dentro de una muestra. Para los datos neuronales, hay dos tipos de variación que son de interés según se indica en [3], las variaciones de tiempo, a partir de las cuales se generan histogramas de intervalos de tiempo y las variaciones de amplitud, generando de este modo un histograma de amplitud, donde sólo aparecen las variaciones de EPSPs e IPSPs. Además, sólo se tienen en cuenta los eventos que han alcanzado su pico antes del comienzo de otro evento.

Tiempo de correlación de eventos

En probabilidad y estadística, la correlación indica la fuerza y la dirección de una relación lineal y la proporcionalidad entre dos variables estadísticas. Se considera que dos variables cuantitativas están correlacionadas cuando los valores de una de ellas varían sistemáticamente con respecto a los valores homónimos de la otra. En esencia, la correlación de tiempo de los eventos neuronales es similar a analizar la distribución de frecuencia o los histogramas de dichos eventos.

Mientras que la duración temporal de las respuestas estaba relacionada con los eventos contiguos y con los eventos similares, esta forma de analizar los eventos es más independiente. Este programa, conocido como el programa de correlación, mide la probabilidad (o correlaciona) de que, dado un determinado tipo de evento (EPSP, IPSP o potencial de acción), habrá otro evento de un tipo concreto dentro de un intervalo de tiempo de retardo determinado. Los dos eventos de referencia y de referencias cruzadas, no necesitan ser adyacentes, ni tienen que ser del mismo tipo.

Este análisis es mucho más significativo que analizar la distribución de frecuencia, ya que los eventos observados (EPSP, IPSP) son por lo general de cada una de las múltiples entradas de la neurona que se está midiendo y debido a las diferentes señales de las entradas los eventos pueden ser asíncronos.

Se emplean dos tipos de correlaciones en este método:

- Las correlaciones tipo-A, independientes de la amplitud, tanto del evento de referencia como del evento de referencia cruzada. Todos los eventos incluidos en los histogramas de intervalos de tiempo se incluyen en este tipo.
- Las correlaciones tipo-B, donde se llevan a cabo con una condición de amplitud impuesta. Sólo los eventos subumbrales (IPSP y EPSP) que cumplan los requisitos para su inclusión en los histogramas de amplitud son considerados para este tipo de correlación. Todos los potenciales de acción se pueden incluir en cualquiera de los dos tipos.

Este programa, desarrollado en [3], es uno de los primeros métodos de análisis de eventos postsinápticos que nos encontramos. No se contemplaba una clasificación de eventos detectados, bien sea por falta de datos fiables para una clasificación o por que las características extraídas no eran suficientes. A pesar de ello se consiguen las primeras estimaciones sobre algunos eventos (los EPSP ocurren próximos entre ellos y espaciados unos 50 ms, establece una relación entre los IPSP y EPSP).

En definitiva, este primer método de análisis basado en histogramas de amplitud y correlación de tiempos nos da una idea del comportamiento de las respuestas neuronales intracelulares.

1.2 Algoritmos basados en umbrales

La detección automática de eventos sinápticos espontáneos, o eventos miniatura, generalmente se ha basado en técnicas de detección por umbral. En estos algoritmos el umbral se fija en un nivel absoluto o bien en un nivel proporcional a la línea base o nivel de ruido del registro.

En concreto, en el método de detección más simple que utiliza umbrales de amplitud, según se describe en [4], un evento será detectado cuando la amplitud de la señal postsináptica registrada supere cierto nivel umbral que variará según la procedencia del registro. El valor de este umbral se obtiene a partir de una estimación de registros anteriores.

Estas técnicas tienen la ventaja de ser relativamente rápidas y fáciles de implementar, pero dado que estos algoritmos utilizan únicamente información acerca de la amplitud de los eventos postsinápticos su rendimiento es peor, sobre todo porque estos eventos pueden sufrir atenuaciones.

Una alternativa son los algoritmos de detección basados en umbrales de la primera derivada de la señal como se indica en los artículos [5] y [6]. Normalmente, los datos se suavizan con un filtro antes de aplicar el algoritmo de la primera derivada. Estos algoritmos tienen una gran resolución temporal y son más robustos frente al ruido de baja frecuencia presente en el registro, pero por lo general muestran una alta tasa de detección de falsos positivos. Los falsos positivos (FP) son producidos por el ruido de la línea base, amplificado por el proceso de derivar la señal, mientras que los falsos negativos (FN) serán producidos por eventos postsinápticos con tiempos de subida lentos, que no serán detectados.

Cualquier ruido o artefacto transitorio que sea lo suficientemente grande o rápido será detectado, independientemente de su forma. Algunas de las técnicas de umbral intentan evitar este problema mediante la aplicación de criterios de selección adicionales. En [6] por ejemplo, comparando la evolución en el tiempo de cada evento detectado con el comportamiento temporal de un evento sináptico esperado, rechazando así los eventos que no coincidan con este criterio.

La realización de cualquier algoritmo de detección por umbral depende fundamentalmente de las condiciones del registro y el valor umbral seleccionado. Estas técnicas se ven comprometidas por el ruido de alta frecuencia, y los datos deben ser generalmente procesados con un filtrado adicional antes de que el algoritmo se pueda aplicar. Por lo tanto, los eventos postsinápticos sufrirán atenuaciones debidas al procesamiento de la señal. El filtrado y umbral óptimos deben ser determinados por un método de ensayo y error, distinto para cada conjunto de datos.

El rendimiento de ambos métodos (umbral de amplitud y de primera derivada) es bastante inferior frente al análisis manual de un investigador experimentado en el que, a priori, el conocimiento acerca de la evolución en el tiempo de los eventos postsinápticos permite distinguir eventos reales de artefactos experimentales.

1.3 Algoritmos basados en templates o plantillas

Los eventos postsinápticos espontáneos pueden ser difíciles de detectar mediante técnicas de umbral cuando sus valores de amplitud son similares al nivel de ruido de los registros. En la documentación consultada se plantean técnicas de detección automáticas para eventos pequeños y asíncronos basados en plantillas. Estos algoritmos simplemente utilizan una o varias plantillas con la forma de onda típica de un evento postsináptico, y van recorriendo los registros, comparando la plantilla con los posibles eventos. Si se cumplen ciertos criterios o umbrales de similitud se considera una detección.

Hay muchas variaciones de la técnica de detección basada en plantillas. Por ejemplo, los algoritmos de plantillas fijas, que utilizan una única plantilla para recorrer los registros y encontrar eventos similares, algoritmos basados en plantillas escaladas que utilizan tanto la amplitud como la similitud para la detección de eventos o las técnicas de correlación cruzada capaces de detectar eventos de diferente amplitud con formas similares a la plantilla.

En cualquier caso, la plantilla consiste básicamente en una línea de base previa y el evento. La duración o la longitud de la plantilla (en ms) se determinan de forma experimental. La longitud óptima de la plantilla debe alcanzar un compromiso entre el tiempo suficiente para que un evento decaiga de nuevo a la línea de base, sin que interfiera con la detección de otro evento posterior que potencialmente podría ocurrir en un pequeño intervalo de tiempo entre eventos.

La plantilla resultante tendrá unidades de tiempo (ms) y amplitud (pA o mV, según sea un registro en corriente o en voltaje) aunque se suele normalizar respecto al pico de amplitud del evento promedio, por eso es más útil pensar en la plantilla en términos de unidades adimensionales de tiempo y amplitud.

1.3.1 Análisis mediante plantillas fijas

Los algoritmos de plantillas fijas utilizan una única plantilla, o varias, para escanear datos y encontrar eventos similares a dicha plantilla tanto en forma como en amplitud. Inicialmente se emplearon para detecciones en registros intracelulares, aunque se utilizan mayormente para detectar potenciales de acción en registros extracelulares. Son muy restrictivas para los eventos de interés: los artefactos y ruidos transitorios son rechazados automáticamente ya que no coinciden con el rango temporal de la plantilla. Sin embargo, estas técnicas utilizan plantillas con pocas amplitudes fijas, mientras que los eventos postsinápticos espontáneos que nos interesa detectar presentan una gran variación de amplitudes.

Por esta razón, las técnicas de plantillas fijas están en desuso a la hora de detectar eventos postsinápticos espontáneos.

1.3.2 Análisis mediante plantillas escaladas óptimamente

En el estudio realizado en [6] se describe un método simple y sensible para la detección de eventos postsinápticos espontáneos. Es muy selectivo para los eventos que nos interesan y se basa en una plantilla postsináptica con una amplitud variable. Se puede aplicar directamente sobre los datos en bruto, es decir, sin necesidad de realizar un filtrado adicional e incorpora un ajuste automático para que el nivel de ruido presente en el registro influya lo menos posible en los umbrales de decisión.

El rendimiento se ha evaluado en [6] bajo una variedad de condiciones y para el correcto funcionamiento del algoritmo es necesario determinar el ajuste óptimo de los parámetros que intervienen en la plantilla.

Esta técnica consiste en ir recorriendo nuestro registro, de corriente (mEPSC) o tensión (mEPSP), con una plantilla de señal de duración similar al tiempo de un evento postsináptico típico.

Primero la plantilla se va escalando de la mejor forma posible para ajustarse en cada posición a los datos de nuestro registro. Después se calcula un criterio de detección basándose en el factor de escala óptimo y la calidad del ajuste. Finalmente, un evento será detectado cuando supere el nivel umbral determinado, que dependerá del criterio de detección calculado previamente.

A grandes rasgos, un evento es detectado cuando la plantilla óptimamente escalada se ajusta a la traza del registro.

La sensibilidad y selectividad de éste método, que se ha probado en [6] con datos reales y simulados, es comparable con la detección visual y mejora los resultados obtenidos con las técnicas de detección por umbral únicamente. Aun así, el ajuste de los parámetros de las plantillas utilizadas afecta a los resultados obtenidos.

A pesar de que este método utiliza plantillas para la detección de eventos sigue influyendo el ruido presente en el registro, aunque el algoritmo trata de compensar de forma automática los cambios que se produzcan en el nivel de ruido. Los eventos postsinápticos en miniatura se solapan con el ruido de fondo del registro, lo que puede ocultar pequeños eventos y dar así mayor importancia a los eventos más grandes.

Un incremento en la amplitud media de los eventos o una atenuación del ruido del registro puede aumentar la tasa de detección de eventos pequeños, es decir, con un buen procesamiento de la señal podemos obtener mejores resultados.

Según el estudio realizado en [6], en condiciones normales se detectan todos los eventos postsinápticos rápidos (con pequeños tiempos de subida) con amplitudes de al menos tres veces la desviación estándar del ruido de fondo ($3\cdot\sigma$) y el 75% de los eventos con amplitudes superiores a $2\cdot\sigma$.

Función de la plantilla

En [6] se propone, para la búsqueda de eventos postsinápticos espontáneos, una función de plantilla que tiene una línea de base nula, seguida de un evento sináptico idealizado que consiste en una función exponencial de subida y otra de bajada:

$$TEMPLATE(t) = \begin{cases} 0 & si\ t \leq 0 \\ NORM \cdot \left(1 - e^{-\frac{t}{RISE}}\right) \cdot e^{-\frac{t}{DECAY}} & si\ t > 0 \end{cases} \quad (1)$$

Donde t es el tiempo desde el inicio del evento postsináptico ideal, **NORM** es el factor de escala que se emplea para normalizar la amplitud de pico, **RISE** es la constante de tiempo de la fase de subida de la plantilla, y **DECAY** es la constante de tiempo de la fase de bajada de la plantilla.

El algoritmo empleado en la detección no depende, ni está limitado, a una única plantilla ya que la expresión (1) es una idealización sencilla de un evento postsináptico y cualquier otra forma de onda que simule dicho tipo de evento podría ser utilizada.

Los parámetros temporales de la plantilla se pueden seleccionar tras una estimación obtenida a partir de varios eventos sinápticos identificados de forma manual. Aunque **RISE** y **DECAY** son dos parámetros importantes, la precisión temporal de la plantilla no es crítica, pero afecta a los resultados.

Después se debe ajustar la plantilla ideal de la expresión (1) a los datos de nuestro registro, para ello se realiza un escalado, se aplica un offset y después se compara, para cada posición, la diferencia entre la plantilla ajustada y los datos del registro que queremos evaluar.

El ajuste de los parámetros **SCALE** y **OFFSET** se basa en el uso de mínimos cuadrados ordinarios (OLS) y se calculan de tal forma que se minimice la suma de errores cuadrados (SSE) entre la plantilla ajustada y los datos. La suma de errores o residuos cuadrados es una medida de la similitud entre los datos y un modelo estimado. Un valor pequeño de SEE indica un buen ajuste del modelo a los datos. Esto se puede expresar como:

$$SSE(i) = \sum_{k=1}^N [SCALE \cdot TEMPLATE(k) + OFFSET - DATA(i + k)]^2 \quad (2)$$

Según se indica en [6] el criterio de detección que se utiliza para la separación de los eventos se calcula a partir del factor de escala de la plantilla (**SCALE**) y de la exactitud del ajuste de la plantilla a los datos, lo que se denomina como desviación del error estándar (RSD) y se define según la expresión (3), donde N es el número de puntos o muestras que forman la plantilla.

$$RSD = \sqrt{\frac{SSE}{N - 1}} \quad (3)$$

A partir de la ecuación **(3)** se calcula el valor del criterio de detección o escala normalizada (**NS**).

$$NS = \frac{SCALE}{RSD} \quad (4)$$

Esta escala normalizada nos servirá para determinar si la plantilla una vez ajustada a los datos se corresponde con un evento postsináptico.

El parámetro **SCALE** puede aumentar o disminuir con el fin de ajustar la plantilla a los máximos de amplitud de los eventos que buscamos y por tanto también variará la desviación del error estándar (**RSD**), que se intentará reducir al mínimo. Puesto que el criterio de detección depende de estos dos parámetros, éste tomará un valor diferente para cada fragmento de registro con el que trabajemos. Por tanto, si **NS** supera un determinado valor umbral, en la traza del registro que se ha comparado con la plantilla ajustada tendremos un evento. A partir de la ecuación **(4)** se puede observar cómo este criterio de detección dependerá del nivel de ruido de fondo y de la poca similitud entre la plantilla y los datos, debido a las diferencias en sus respectivas formas de onda. Por lo tanto, este criterio de detección está estrechamente relacionado con la relación señal-ruido de cada evento detectado.

Según se indica en [6], es poco probable que el ruido de fondo supere 4 veces la desviación estándar del ruido ($4 \cdot \sigma$), por tanto un umbral de detección de este orden se aproximará a la sensibilidad de detección óptima, garantizando al mismo tiempo una baja tasa de detecciones erróneas de eventos o falsos positivos.

El mismo nivel de umbral se puede utilizar para cualquier conjunto de datos, independientemente del nivel de ruido, ya que el nivel de umbral se calculará a partir de la desviación estándar del ruido presente en el registro. Es decir, el algoritmo de detección se ajusta automáticamente para compensar las variaciones de ruido que nos encontraremos en los diferentes registros y así mantener un rendimiento óptimo.

Un evento será considerado como tal cuando el criterio de detección supere el nivel de umbral seleccionado, alcance un máximo y después caiga de nuevo por debajo del umbral. El criterio de detección muestra típicamente un pico agudo, sin ambigüedades cuando la plantilla está perfectamente alineada con un evento postsináptico. El máximo absoluto en el criterio de detección da una indicación precisa del instante en el que se produce el evento.

A continuación se muestra un ejemplo sencillo donde se utiliza el criterio de detección, calculado a partir de la plantilla escalada, para localizar pequeños eventos. Se utiliza una traza de datos que contiene únicamente dos eventos postsinápticos (mEPSC) superpuestos con el ruido presente en el registro, el registro tiene una desviación estándar del ruido (σ) de -1 pA según se indica en [6].

Primero una plantilla con una cinética similar a los eventos, en la figura 1.1 se representa una plantilla de ejemplo creada con MATLAB a partir de la expresión (1), recorre la traza mostrada en la figura 1.2A y el criterio de detección se calcula para cada muestra de la traza (Figura 1.2B).

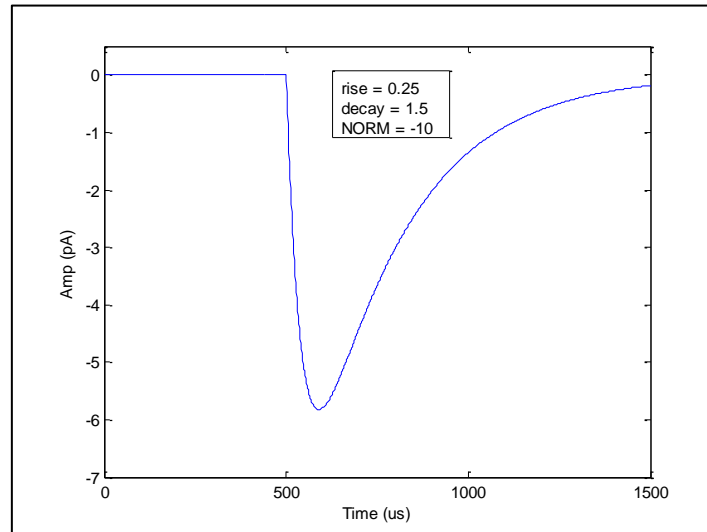


Figura 1.1. Plantilla de ejemplo representada con MATLAB.

Se observa que, como se ha citado anteriormente, el criterio de detección presenta picos, negativos en este caso, cuando la plantilla se ha alineado de forma óptima con el evento postsináptico. Finalmente, utilizando un umbral de -4 pA, equivalente a 4σ , ambos eventos serán detectados.

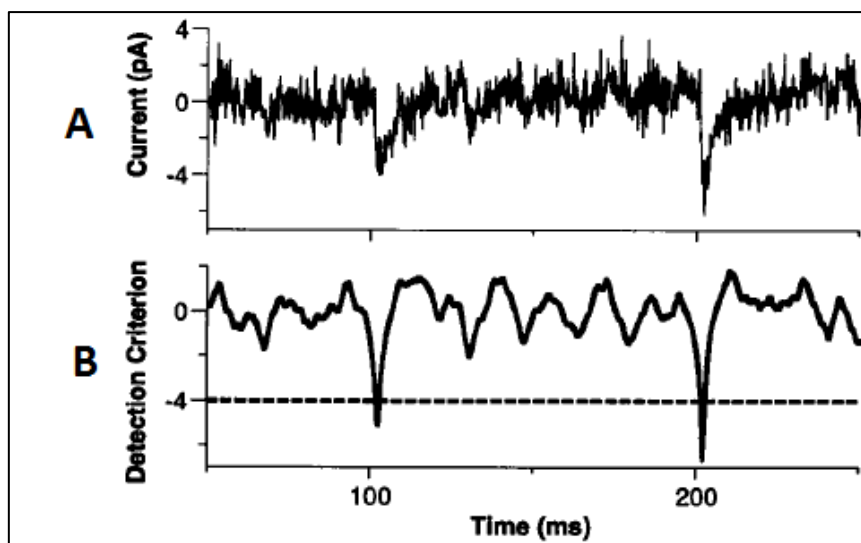


Figura 1.2. (A) Traza de datos con dos eventos y (B) el criterio de detección obtenido.

Fuente: [6]

Este método es bastante selectivo para los eventos que se ajusten al comportamiento temporal de la plantilla. Sin embargo, también se detectan eventos que se desvían de esta evolución en el tiempo, sobre todo si son más grandes temporalmente que la plantilla. Esto puede ser una ventaja, ya que la duración de los potenciales o corrientes en miniatura (mEPSC y mEPSP) puede variar de un evento a otro según se indica en [6]. Por otro lado pueden surgir problemas en la detección y la caracterización si los eventos se encuentran solapados con alguna otra señal externa o con otros eventos, aunque sean similares temporalmente. Para obtener mejores resultados se propone aplicar criterios de selección adicionales, como parámetros basados en la forma de onda por ejemplo. En esta técnica se contempla además el rechazo opcional de eventos con grandes errores típicos como tiempos de subida excesivamente rápidos o lentos, o grandes errores estándar de referencia o de offset.

En [6] el algoritmo de detección fue probado con datos simulados, aunque esto no aporta excesiva veracidad al algoritmo, y también con datos reales. Se utilizan dos medidas para analizar el rendimiento del método de detección automático propuesto, el porcentaje de eventos identificados correctamente (sensibilidad de detección) y la tasa de falsos positivos (FP), sobre estas medidas influye el ruido, el umbral de detección, la longitud del registro analizado, el solapamiento de eventos y la elección de los parámetros temporales de la plantilla frente a la variabilidad temporal de los eventos postsinápticos.

En conclusión, este algoritmo es, al menos, tan sensible como la detección de eventos visual cuando se aplica a los datos electrofisiológicos reales. Se observa una mayor sensibilidad y selectividad que las técnicas de umbral previamente descritas. El enfoque de las plantillas escaladas también tiene dos ventajas prácticas respecto a otras técnicas automatizadas: los datos no tienen que ser filtrados previamente y además, antes de que se analicen, el algoritmo de detección compensa de forma automática los cambios en el nivel de ruido, por lo que, el mismo ajuste de umbral se puede utilizar para cualquier conjunto de datos. La detección fiable se puede lograr mediante el uso de un procedimiento sistemático sencillo para seleccionar los parámetros de plantilla y el nivel de umbral. Para extraer un rendimiento óptimo, puede ser necesario algún ajuste adicional de estos parámetros. Las dos desventajas principales de este enfoque son su procesamiento relativamente lento debido a la gran cantidad de cálculos numéricos necesarios, y su incapacidad para detectar eventos solapados o que estén separados en el tiempo por menos de la longitud de la plantilla utilizada. Aunque la velocidad de procesamiento no tiene por qué ser un problema grave, ya que depende del tamaño de los registros que queremos analizar y de las condiciones de trabajo de los electrofisiólogos. Estas desventajas pueden minimizarse relativamente mediante la selección de una plantilla corta.

1.3.3 Ajuste ponderado de mínimos cuadrados con varias plantillas

La técnica de plantillas escaladas de forma óptima explicada anteriormente es capaz de detectar pequeños eventos comparables con el ruido de fondo. Sin embargo, la tasa de detección se reduce significativamente, como ya se ha justificado, debido a la superposición de eventos. En el trabajo realizado en [7] se utilizan múltiples plantillas combinadas con la técnica de mínimos cuadrados ponderados (WLS) para la detección de los dos tipos de eventos con los que nos podemos encontrar, aislados y superpuestos.

Para detectar automáticamente los eventos espontáneos en [7] se ha desarrollado una técnica en la que se utiliza un algoritmo de mínimos cuadrados ponderados (WLS) en lugar de los mínimos cuadrados ordinarios (OLS) utilizados en el estudio realizado en [6] que se explica en el apartado anterior.

Además de la ponderación, se utilizan varias plantillas de forma simultánea para escanear los datos del registro y aumentar así la precisión del ajuste entre los datos y las plantillas. Una mejora importante que se consigue con la técnica de las plantillas ponderadas es que la tasa de detección deja de ser tan sensible a la longitud de las plantillas y además esta técnica es capaz de realizar una detección de eventos superpuestos de forma fiable.

Del mismo modo que se utilizaba OLS en el algoritmo anterior, con la técnica WLS también se busca la minimización de la suma errores. Sin embargo, en lugar de considerar todos los puntos con la misma igualdad, WLS pondera los puntos de tal forma que las regiones seleccionadas de la plantilla hacen una mayor contribución al ajuste. En particular, haciendo un mayor hincapié en el ajuste de la línea base inicial y la fase de subida de la plantilla para el evento hemos encontrado.

La técnica de mínimos cuadrados ponderados permite una mayor precisión en la detección, este método mejora la detección de ráfagas de eventos compuestas de múltiples eventos individuales superpuestos.

a) Función de la plantilla

Este algoritmo utiliza para la plantilla una función $TEMPLATE(t)$ similar a la expresión (1) del apartado anterior. Pero además añade la función de ponderación, que da prioridad al ajuste con la línea base y la fase de subida. Se calcula de acuerdo a la expresión:

$$WEIGHT(t) = \begin{cases} \beta & si t \leq 0 \\ \beta \cdot e^{-\frac{t}{m}} & si t > 0 \end{cases} \quad (5)$$

donde m es una constante que determina la tasa de pérdida de la ponderación y β es un factor de escala que hace que se cumpla la siguiente expresión, siendo N el número de muestras de la plantilla:

$$\sum_{t=0}^{N-1} WEIGHT(t) = N$$

Ahora, igual que en el método de plantillas escaladas, se deben ajustar los parámetros de la plantilla de tal forma que se minimice la suma de errores cuadrados (SSE) entre la plantilla ajustada y los datos, pero utilizando mínimos cuadrados ponderados no se le da el mismo peso a cada par de datos y plantilla. Por lo tanto, el parámetro $WEIGHT(k)$ se introduce en la ecuación (2), que ahora se convierte en:

$$SSE(i) = \sum_{k=0}^{N-1} WEIGHT(k) \cdot [SCALE \cdot TEMPLATE(k) + OFFSET - DATA(i + k)]^2 \quad (7)$$

Además de las escalas normalizadas, según se indica en [7] también se utilizan la correlación cruzada y la desviación estándar residual para eliminar las fluctuaciones de ruido presentes en los diferentes eventos detectados.

Para la detección de eventos, la plantilla puede ser generada a partir de un promedio de eventos identificados visualmente en un registro concreto o puede ser simulada de acuerdo a la ecuación (1) con valores de **RISE** y **DECAY** basados en la estimación de eventos identificados visualmente.

b) Detección de eventos aislados

Para ilustrar el método de detección de eventos con múltiples plantillas, en [7] se realiza una detección con tres plantillas sobre datos simulados con eventos aislados (Figura 1.3). El uso de los datos simulados para la evaluación de diferentes estrategias de detección es ventajoso ya que el número total de eventos dentro del conjunto de datos es conocido. Con datos reales, incluso un investigador experimentado en la detección visual de eventos puede fallar para identificar un número indeterminado.

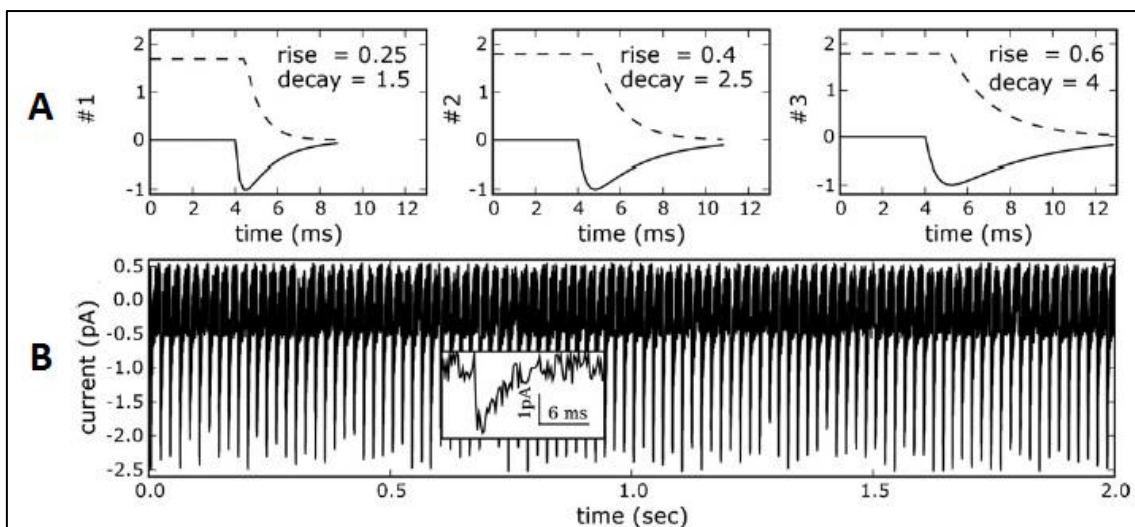


Figura 1.3. (A) Plantillas con diferentes parámetros y (B) datos simulados. Fuente: [7]

Las plantillas y las ponderaciones se han generado con los parámetros indicados en la figura 1.3A de acuerdo a las ecuaciones (1) y (5) respectivamente. Los datos, según se indica en [7], se han generado mediante la superposición de 100 eventos postsinápticos simulados, uniformemente espaciados, sobre ruido blanco (Figura 1.3B).

En este ejemplo utilizado en [7], todos los eventos se detectan correctamente y no hay falsas detecciones.

c) Detección de eventos solapados

La técnica de plantillas escaladas con OLS desarrollada en [6] no es capaz de detectar eventos que se superponen. En [7] se ha realizado un análisis comparativo de ambos métodos (OLS y WLS).

Primero se ha utilizado el método OLS, con una $\sigma = 3$, con las plantillas que aparecen en la figura 1.3A, sobre un registro con trenes de eventos superpuestos (Figura 1.4). Cada tren contiene tres eventos. El primer evento de cada tren tiene una clara línea base previa al evento, el segundo carece tanto de pre-evento de referencia y una fase de caída completa, el tercero carece de una línea de base pre-evento, pero tiene una descomposición completa a la línea de base.

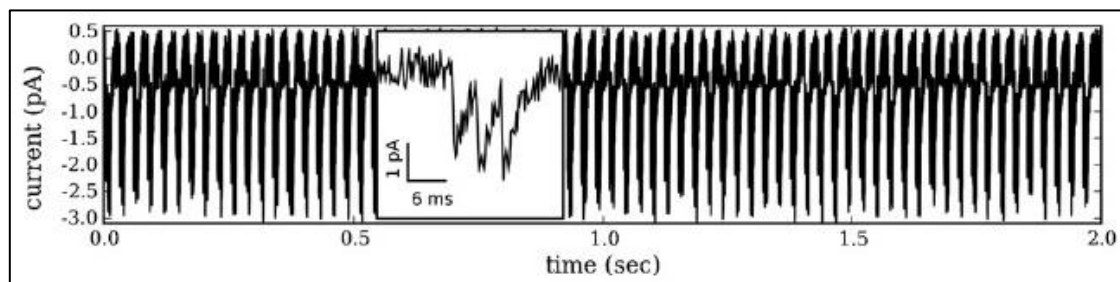


Figura 1.4. Registro con eventos solapados. Fuente: [7].

La cantidad de eventos detectados, cuando hay solapamiento, utilizando OLS depende notablemente del tamaño de la línea base de la plantilla. Reduciendo la longitud del pre-evento se aumenta la sensibilidad de este método. Cuando la línea base se elimina por completo de la plantilla se alcanza el máximo de detecciones pero también aumenta la detección de falsos positivos.

En la figura 1.5 se puede apreciar un ejemplo de lo que ocurre utilizando OLS, sólo se detectan los últimos eventos de cada ráfaga, ya que son los que tienen la fase de caída completa, además se producen detecciones erróneas, por ejemplo en la tercera ráfaga (Figura 1.5A) se detecta dos veces el mismo evento. En cuanto al criterio de detección (Figura 1.5B), utilizando un nivel umbral de 3σ , se muestra que sólo el último evento de cada ráfaga produce valores de NS mayores que el umbral.

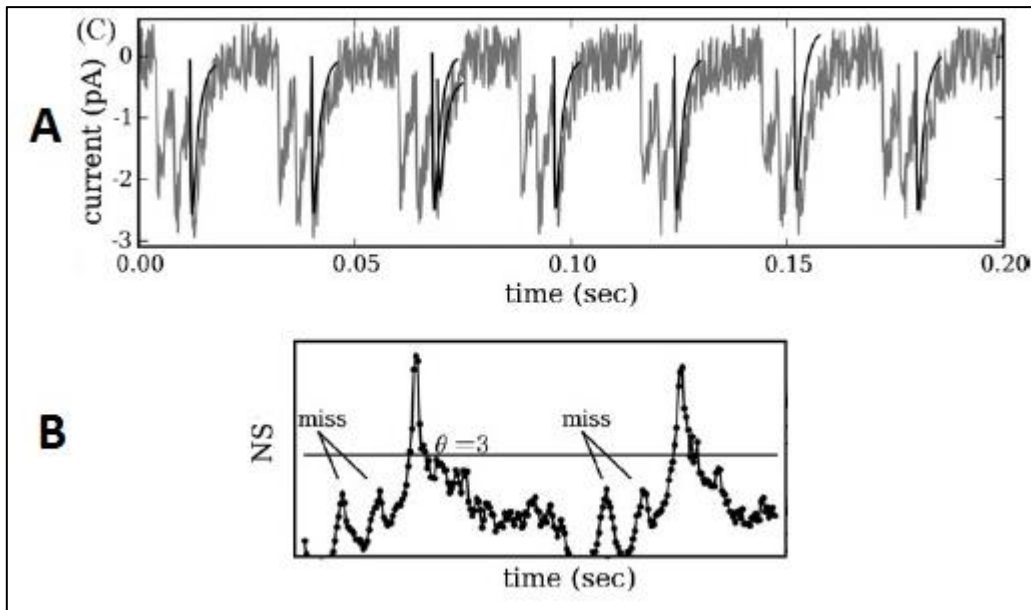


Figura 1.5. (A) Eventos detectados por plantillas sin línea base y (B) criterio de detección.
Fuente: [7]

La incapacidad para detectar los eventos anteriores dentro de la ráfaga se debe a la falta de coincidencia con la fase de caída de la plantilla. Por tanto, para mejorar la detección de estos eventos anteriores en [7] se plantea reducir la duración del tiempo de bajada de la plantilla.

Cuando se acortan las fases de caída de las plantillas, se mejora la correspondencia entre la caída del evento y el template de ajuste. Sin embargo, ya que no se puede predecir el intervalo entre dos eventos solapados para registros reales, es difícil decidir sobre la duración de las fases de caída de plantillas específicas.

Los resultados obtenidos con eventos solapados en [7] al utilizar el método OLS no son del todo satisfactorios ya que hay falta de selectividad entre el ruido y los eventos, encontramos demasiadas detecciones erróneas y las posibles soluciones como modificar las plantillas no son óptimas.

En conjunto, estos resultados ponen de manifiesto la mala detección de múltiples eventos superpuestos cuando se utiliza la detección con plantillas basada en OLS.

A continuación se muestra (Figura 1.6) la mejora de los resultados con la técnica WLS para detectar eventos que se solapan. En este caso se han utilizado plantillas (Figura 1.6A) sin línea base previa con el ajuste de ponderación adecuado sobre los datos de la figura 1.4. Sin embargo, debido a la falta de línea de base, muchas fluctuaciones del ruido de fondo son falsamente detectadas como eventos (Figura 1.6B). En la figura 1.6C se muestran algunos valores de NS obtenidos en el proceso de detección.

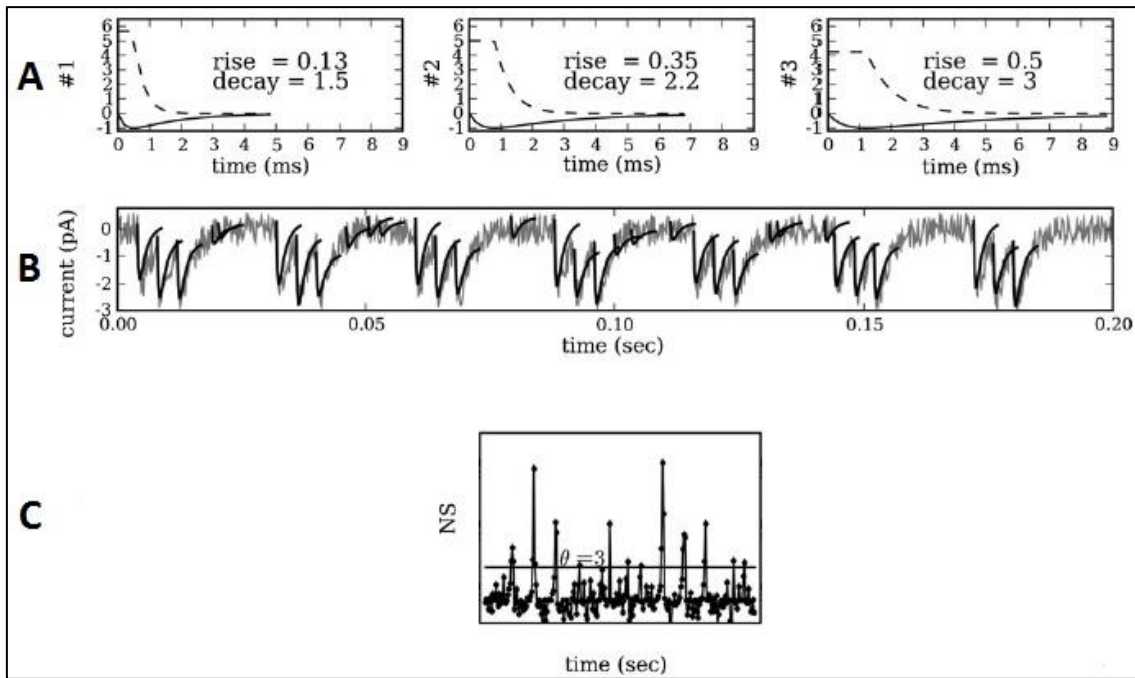


Figura 1.6. (A) Plantillas ponderadas. (B) Traza del registro con las plantillas escaladas. (C) Valores de la escala normalizada (NS) de la mejor plantilla. Fuente: [7]

Con este análisis se producen muchas más detecciones aunque varias son falsas debido a la detección errónea de las fluctuaciones del ruido como eventos. Gracias al uso de datos simulados se pueden distinguir las detecciones correctas y las erróneas ya que se conoce el número de eventos y el instante en que se producen dentro del registro.

Si se observan detalladamente algunos parámetros como SCALE, NS o la desviación del error revelan algunas diferencias que pueden ser útiles para distinguir los eventos correctos de las detecciones erróneas. La mayor diferencia entre detecciones y falsos positivos se encuentra en los valores del parámetro SCALE, esto se debe a que las plantillas utilizadas tienen valores de amplitud pequeños, muy similares a los máximos que encontramos en el ruido. Si umbralizamos según el factor de escala todos los eventos que se han detectado en la prueba podemos eliminar las detecciones erróneas causadas por las fluctuaciones del ruido.

Una consecuencia inmediata de esto es que el evento detectable más pequeño será aproximadamente igual a la amplitud del ruido presente en el registro de datos, aunque en general esto también ocurrirá al detectar eventos a simple vista. En este análisis, realizado en [7], la diferencia en los valores del parámetro SCALE es tan clara porque los datos se han generado artificialmente, con eventos de amplitudes uniformes. Sin embargo, la distribución de las amplitudes de los eventos en datos biológicos es obviamente aleatoria. Por tanto, no se espera que los valores de SCALE, o cualquier otro parámetro, de eventos y ruido estén claramente separados sino que tendrán un considerable solapamiento.

Para comprobar las implicaciones de esto en [7] se realiza un análisis con un registro simulado (Figura 1.7A) con eventos cuyas amplitudes tienen más aleatoriedad que el registro que se analizaba anteriormente. Los resultados obtenidos con WLS son similares a los del análisis anterior (Figura 1.6), hay un gran número de detecciones pero también falsos positivos, sin embargo, la umbralización que se realizaba anteriormente para identificarlos ya no es válida puesto que ya no existe tanta diferencia en los parámetros y la selección del valor umbral no es inmediata.

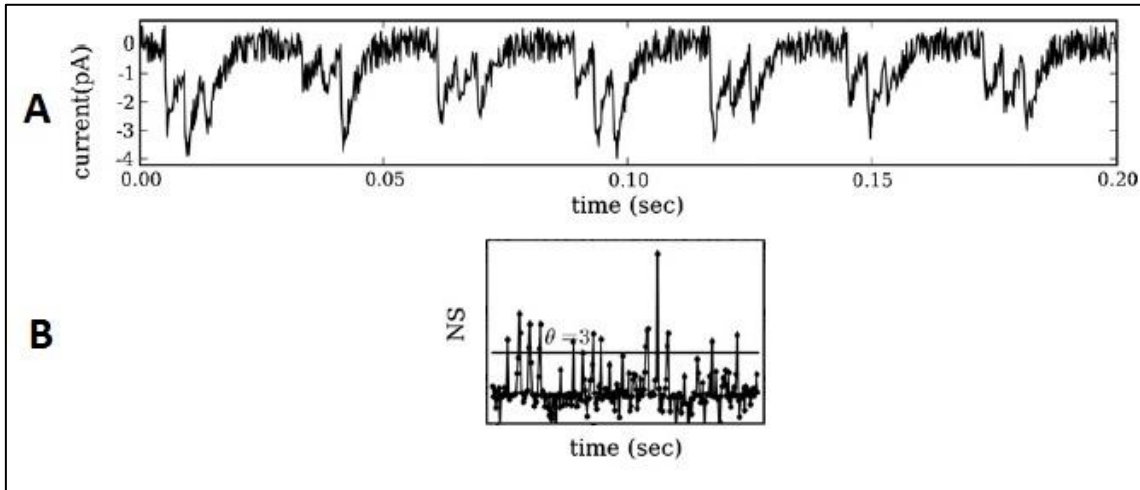


Figura 1.7. (A) Traza con eventos de amplitud aleatoria. (B) Criterio de detección (NS).
Fuente: [7]

Una forma de seleccionar un umbral que permita eliminar los falsos positivos es escanear una parte de los datos que carezca de eventos y estimar un umbral a partir de algún parámetro que pueda ser crítico. Por ejemplo en el estudio que se realiza en [7], se parte de la distribución del parámetro SCALE para seleccionar el umbral que permita distinguir entre las detecciones correctas y las erróneas. Aunque se utiliza el mismo parámetro que antes, el procedimiento para elegir el valor umbral es distinto.

En la figura 1.8 se pueden observar los valores de NS y SCALE de cada evento detectado para los dos análisis realizados con la técnica WLS, en negro se representan las detecciones correctas y en blanco los falsos positivos.

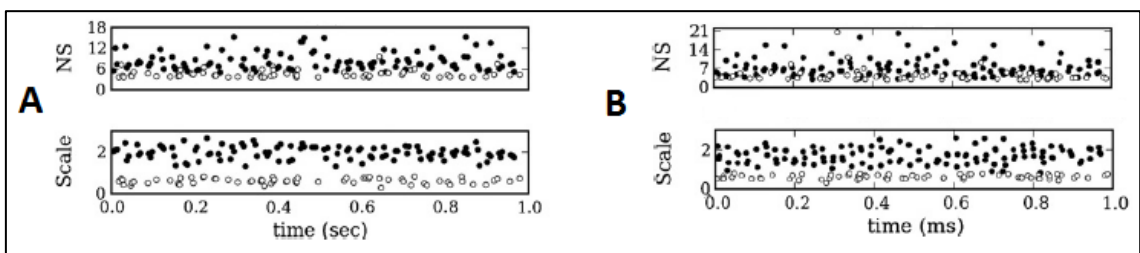


Figura 1.8. Representación de NS y SCALE de los análisis de registros con eventos de (A) amplitudes uniformes y (B) amplitudes aleatorias. Fuente: [7]

En resumen, las características que se utilizan para la separación de eventos se pueden clasificar en dos categorías, características de amplitud y características de similitud. Más en concreto, la extracción de características de similitud depende normalmente del ajuste de la plantilla. Sin embargo, cuando se analizan registros con eventos superpuestos es difícil encontrar un ajuste óptimo para la plantilla y los eventos solapados y esto implica un error en las características de similitud.

El análisis con plantillas escaladas óptimamente, desarrollado en el apartado 1.2.2, es capaz de detectar pequeños eventos aislados, aunque falla ante la superposición de eventos. Otro enfoque es ajustar eventos individuales con plantillas ponderadas sin línea de base para obtener una estimación de la similitud y la amplitud para la separación de eventos. Esta técnica no es demasiado precisa, sin embargo, debido a la tolerancia en cuanto a la variación de la forma, el análisis con plantillas utilizando WLS presenta una alta tasa de detección, como se ha comprobado en el análisis anterior. Aunque el uso de plantillas ponderadas sin línea de base implica un problema, la información sobre el ruido de fondo se pierde en cada ajuste y el factor de escala normalizada (NS) pierde su función de supresión de ruido de fondo y sólo refleja la similitud entre la plantilla ponderada y los datos.

Además la distinción de eventos frente al ruido basada en NS falla ante pequeñas fluctuaciones de ruido que se ajustan a las plantillas. Sin embargo, estos falsos positivos pueden ser eliminados umbralizando a partir de algún parámetro.

En los análisis anteriores, se utiliza la escala normalizada (NS) para seleccionar las mejores plantillas, así como para separar los eventos del ruido. Hay, sin embargo, más de una forma de seleccionar el mejor ajuste y el criterio de detección. En el estudio realizado en [7] se plantea utilizar la correlación cruzada en lugar de la escala normalizada para definir las mejores plantillas, así como para separar eventos. En el análisis de registros pueden utilizarse múltiples criterios de detección simultáneamente para eliminar falsos eventos, esto es importante sobre todo con registros de datos de gran tamaño si además se utilizan plantillas sin línea base ya que normalmente se detectan numerosos eventos erróneos debido a la fluctuación del ruido de fondo y se requiere umbralizar después para eliminar la falsa detección.

En conclusión, una de las ventajas de utilizar múltiples plantillas es que se pueden obtener características de los eventos con más precisión que con las plantillas individuales y estas características, tales como la correlación cruzada y la desviación estándar residual, pueden ser útiles para eliminar las detecciones erróneas. El uso de varias plantillas facilita la elección, tras un ensayo previo, de los parámetros de la plantilla.

Actualmente no existe un criterio para la elección de la técnica de plantillas idónea para la detección de eventos postsinápticos, sin embargo, la técnica de múltiples plantillas ponderadas que se plantea en [7] proporciona un método sencillo para detectar eventos tanto aislados como superpuestos.

1.4 Algoritmo de detección automática por pasos (FAUST)

En el estudio realizado en [8] se ha desarrollado un algoritmo para la detección rápida y precisa de eventos postsinápticos espontáneos. El algoritmo realiza la búsqueda de los eventos a partir de unos criterios de amplitud y pendiente. La característica más importante de este algoritmo es su búsqueda paso a paso, con interés en los puntos de inicio y de pico de cada evento.

La ventaja de este de algoritmo es que es simple y rápido. La desventaja es que el rendimiento de la detección de este método depende de la frecuencia de muestreo y de la relación señal ruido del registro que se analiza, debido a que este tipo de algoritmos se perturban a menudo por el ruido de alta frecuencia. Por eso estos métodos no son muy adecuados para la detección de eventos en miniatura (mEPSC) cuyas amplitudes están más cerca del nivel de ruido. En concreto, este algoritmo está enfocado para eventos espontáneos (sEPSC) que siguen a los potenciales de acción o para eventos en miniatura relativamente aislados.

La variabilidad en la amplitud y complejidad en la forma de onda de los sEPSCs hace que el análisis mediante un algoritmo convencional, mediante plantillas por ejemplo, sea muy complicado. El algoritmo desarrollado en [8] se basa en umbrales de pendiente y amplitud y es capaz de detectar simultáneamente múltiples picos en registros con eventos espontáneos complejos y pequeños eventos en miniatura prácticamente aislados y con bajos niveles de ruido.

Desarrollo del algoritmo

El proceso de detección del algoritmo se puede dividir en cinco pasos. En la primera etapa (Figura 1.9A), el algoritmo hace una estimación aproximada de los posibles eventos presentes en la traza que se está analizando y determina un instante "temporal" de inicio del evento. El siguiente paso (Figura 1.9B) es la detección de la muestra de inicio correcta del evento y después se localiza el punto temporal donde está el pico del evento, que probablemente se encontrará varias muestras después del instante de inicio (Figura 1.9C). La cuarta etapa (Figura 1.9D) es una búsqueda hacia atrás para detectar el verdadero máximo del evento. El paso final (Figura 1.9E) es comprobar si la amplitud del evento detectado excede cierto umbral. Esta búsqueda paso a paso permite la detección precisa de los puntos de pico de los eventos.

En [8] se ha desarrollado también una herramienta, **FAUST**, que facilita el uso del algoritmo. Esta herramienta requiere cinco parámetros iniciales para realizar las detecciones:

- Un umbral de amplitud, **th**, que determina la amplitud mínima de los eventos detectados por el algoritmo.
- La pendiente mínima, **slope**, de los eventos. Este valor se traduce en un parámetro temporal, **wi**, que se define como **th/slope**.
- El parámetro **bin**, que determina el número de puntos promediados.
- El periodo temporal, **pp**, para la búsqueda de picos.
- El parámetro **n**, que indica el ruido de línea base, y puede ser ajustado automáticamente por el programa.

El algoritmo internamente traduce los parámetros temporales en muestras y su diagrama de flujo se representa en la figura 1.10.

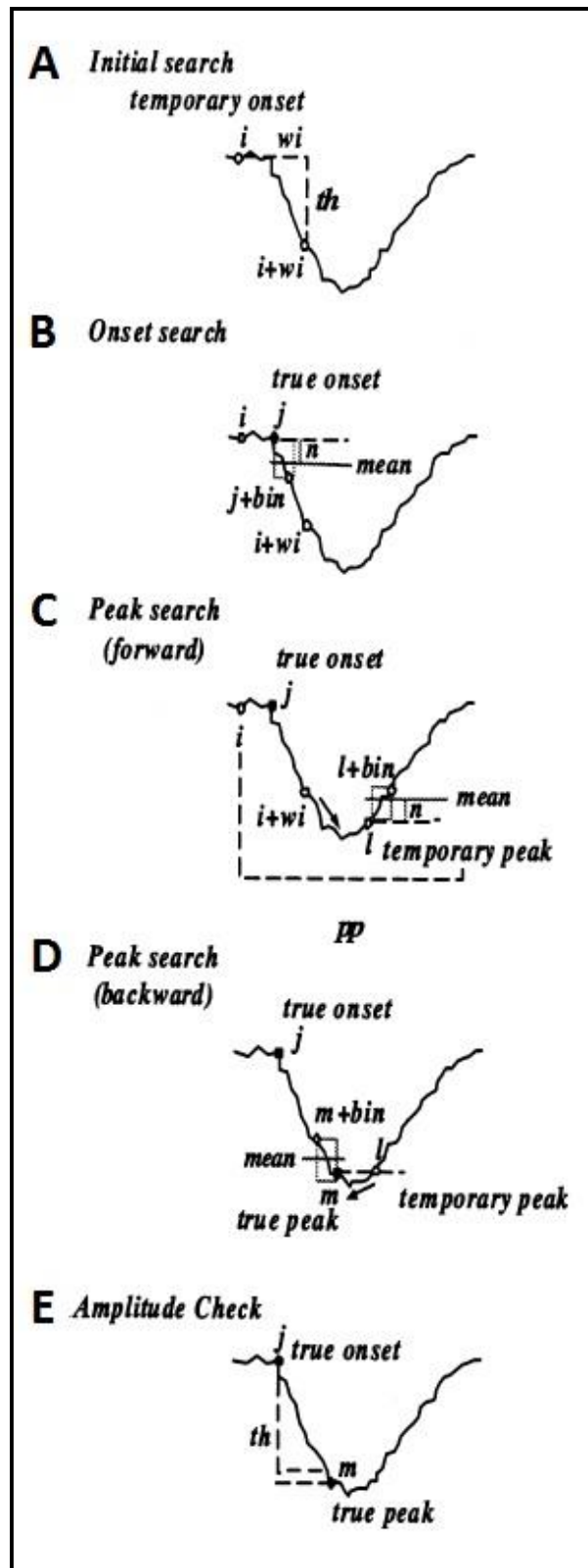


Figura 1.9. Esquema de la detección de eventos. Cuando no se cumple ningún criterio en el paso **A** el resto se omiten. Fuente: [8]

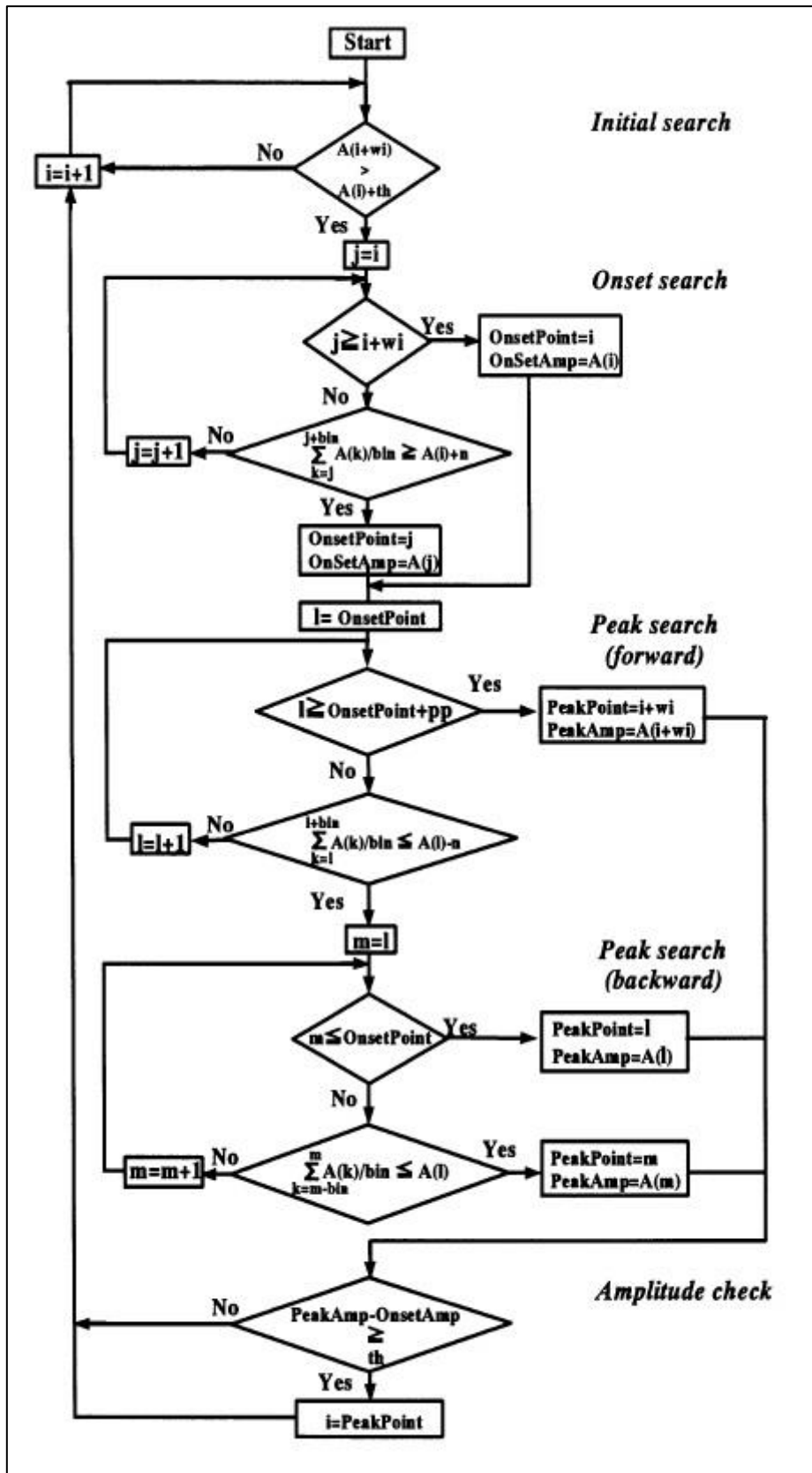


Figura 1.10. Diagrama de flujo del algoritmo formado por bucles y sentencias IF.

Fuente: [8]

En [8] se ha probado el algoritmo y los resultados han sido satisfactorios para los registros que se han procesado, consiguiendo detecciones de eventos espontáneos en un amplio rango de frecuencia y amplitud. Además el algoritmo se ha utilizado para la detección de otros tipos de señales electrofisiológicas como por ejemplo potenciales de acción extracelulares. Esto es posible gracias a que el algoritmo no depende de una plantilla particular sino que se basa en pendientes y amplitudes para la detección.

Para comprobar la eficiencia del algoritmo (**FAUST**), en [8], se compara la detección hecha por programas comerciales que emplean otro tipo de algoritmos: **Mini analysis** (Synaptsoft), y **Axograph** (Axon instruments).

Para la comparativa se analiza un registro de datos simulados que contiene en total 1800 sEPSCs complejos solapados con el ruido de fondo. La ventaja de utilizar un registro simulado es que se conocen todos los eventos presentes y se puede realizar mejor una comparativa. En la figura 1.11 podemos observar los resultados de los diferentes análisis representados en histogramas.

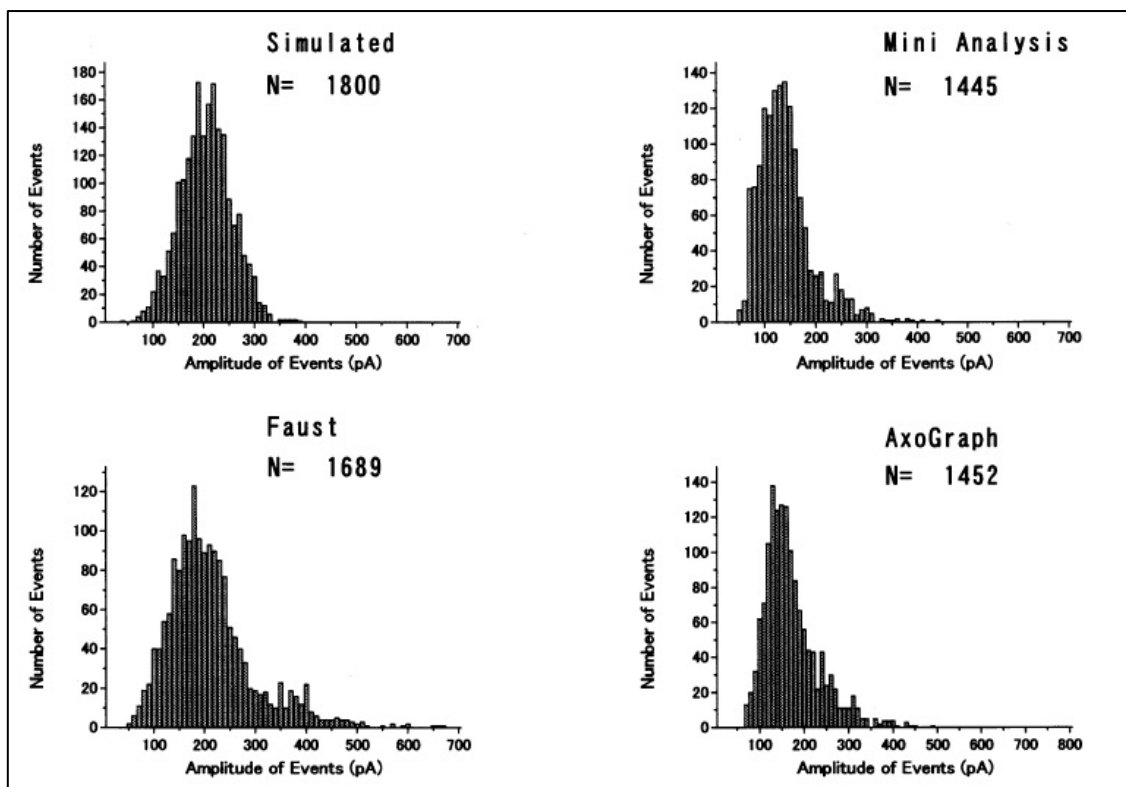


Figura 1.11. Histogramas de amplitud de los N eventos detectados. Fuente: [8]

Con la herramienta **FAUST** basada en este algoritmo se detectan el 93% de los eventos del registro simulado, mientras que con **Mini Analysis** se detectaron el 80.2% y con **AxoGraph** el 80.6%.

En conclusión, se obtienen mejores resultados con este algoritmo frente a otros que utilizan los programas comerciales. Es relativamente sencillo frente a las técnicas de ajuste de plantillas que se han explicado anteriormente y la precisión en la detección no es menor que estos métodos, si los parámetros para la detección se ajustan de forma óptima.

1.5 Algoritmo de deconvolución ciega

Según se describe en [9], se ha desarrollado un algoritmo de deconvolución ciega de máxima probabilidad para detectar eventos en registros con EPSC complejos. El algoritmo es capaz de caracterizar la cinética de los EPSC típicos, así como delimitar eventos individuales con una resolución temporal más alta que otros métodos existentes. El enfoque también sirve para registros con bajas relaciones señal a ruido y aquellos con solapamientos importantes entre eventos. En [9] se demuestra la eficacia del algoritmo en registros reales y datos simulados de alta complejidad.

En [9] se parte de una premisa indicada en la expresión **(8)**: los registros que queremos analizar se pueden describir como un sistema formado por la convolución de una señal con una función de respuesta de impulso. Por tanto, se ha desarrollado un algoritmo de deconvolución que ayuda en el análisis de estos registros electrofisiológicos complejos.

Según se define en [10], la deconvolución ciega se refiere a la recuperación de la señal corrompida por otra fuente, sin conocimiento explícito de la función de respuesta al impulso utilizada en la convolución. Esto se consigue normalmente haciendo suposiciones apropiadas de la entrada para estimar la respuesta de impulso mediante el análisis de la salida.

El objetivo que se plantea en [9] es mejorar los algoritmos existentes de varias maneras. En primer lugar, tratando de distinguir los eventos individuales, incluso cuando la proximidad temporal de los eventos contiguos se acerca al tiempo de subida del EPSC. En segundo lugar, la variabilidad en las condiciones experimentales exige que el algoritmo se adapte automáticamente a cada situación de registro sin necesidad de que el usuario proporcione configuraciones concretas que dependan de las condiciones del registro. Finalmente, se desea un algoritmo ciego que pueda estimar la forma de onda del EPSC unitario directamente a partir de los EPSCs detectados en cada registro. Se espera que la obtención de una buena estimación de la forma unitaria del EPSC, la función de la respuesta al impulso, implique un aumento de la precisión temporal del algoritmo, lo que permite un mejor análisis de las señales con solapamiento de eventos. Con estos objetivos se ha desarrollado e implementado el algoritmo, que después se evaluará su rendimiento sobre datos reales y simulados. En concreto, si observamos la figura 1.12, el propósito del algoritmo planteado es conseguir la reconstrucción de la primera señal, a partir de la observación del registro que se está analizando (en negro) y aplicando la función de la respuesta al impulso (Figura 1.12B).

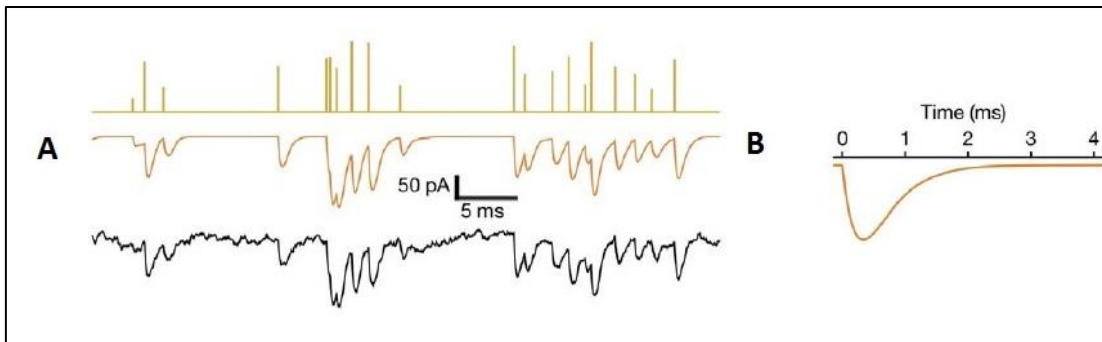


Figura 1.12. (A) Señal de densidad de amplitud, señal filtrada y registro original, de corrientes postsinápticas y (B) respuesta al impulso del sistema. Fuente: [9]

En [9] se define la expresión **(8)** para el algoritmo de deconvolución ciega. Se trata de una ecuación general que nos sirve para representar matemáticamente los registros que queremos analizar. En la expresión, los datos observados (\mathbf{x}) son la suma de la señal (\mathbf{s}) filtrada a través de una convolución con la respuesta al impulso (\mathbf{f}) y el ruido de fondo (\mathbf{n}).

$$\mathbf{x} = \mathbf{s} * \mathbf{f} + \mathbf{n} \quad \mathbf{(8)}$$

A partir de los datos registrados, \mathbf{x} , se pretende con este algoritmo obtener la señal \mathbf{s} y la respuesta al impulso, \mathbf{f} . Para ello será necesario un proceso de filtrado no lineal para separar \mathbf{s} y \mathbf{f} .

El desarrollo del algoritmo se puede descomponer en dos partes. Durante el primer paso, la deconvolución, se busca la señal óptima dados los datos y una estimación preliminar del filtro o respuesta al impulso. En el segundo paso, la estimación del filtro, se mejora la estimación previa del filtro dados los datos y la estimación actual de la señal. Estos dos pasos se pueden alternar sin que eso afecte a los resultados obtenidos.

a) Deconvolución

A partir de los datos que se identifican en \mathbf{x} y un filtro \mathbf{f} , podemos encontrar una representación dispersa de la señal mediante la optimización del valor de un regularizador de máxima entropía.

El principio de máxima entropía, según se describe en [11], se basa en la premisa de que cuando se estima la distribución de probabilidad, hay que seleccionar la distribución que tenga mayor incertidumbre, es decir, la entropía máxima. Teniendo en cuenta las posibles limitaciones. De esa manera no se introduce ninguna hipótesis adicional en los cálculos.

Un regularizador es una función de coste introducida en el procedimiento de optimización para asegurar la suavidad, la magnitud, o alguna otra característica de la representación que se busca.

Encontrar el mejor valor del parámetro de regularización (α) es con frecuencia un desafío en problemas de deconvolución, ya que la optimización se consigue sólo con el valor correcto de α . Cuando α es demasiado grande, la señal obtenida carece de la resolución completa disponible a partir de los datos. Cuando α es demasiado pequeño, se produce demasiada influencia del ruido de fondo.

La estimación óptima de la señal se consigue mediante la maximización de una función de probabilidad que depende de \mathbf{x} , \mathbf{f} y el parámetro de regularización (α) con respecto a $\hat{\mathbf{s}}$, para obtener la estimación $\hat{\mathbf{s}}$.

La presencia de ruido y la naturaleza de la probabilidad a priori significan que $\hat{\mathbf{s}}$ será casi, pero no exactamente, cero en regiones donde no haya eventos, pero si haya ruido. Cualquier estimación puntual se verá alterada ante la presencia de eventos diminutos en $\hat{\mathbf{s}}$.

Esta representación dispersa ($\hat{\mathbf{s}}$) es el resultado del primer paso del algoritmo, la deconvolución ciega, y nos permite continuar perfeccionando la estimación de la respuesta al impulso \mathbf{f} .

b) Estimación de la respuesta al impulso

La calidad de la estimación \hat{s} que se obtiene en el proceso de deconvolución depende fundamentalmente de la estimación de f que hagamos.

El algoritmo se ha planteado en [9] de tal manera que la respuesta al impulso es invariante, aunque la amplitud y la localización en el tiempo de cada evento es diferente. Condicionalmente a la estimación de la señal anterior, se estima la respuesta al impulso, de forma no paramétrica, con una función de probabilidad similar a la que se utiliza para \hat{s} .

También podemos utilizar la estimación \hat{s} obtenida en el paso anterior y optimizar f . Cuando los eventos están bien separados, este proceso es similar a la detección de los eventos aislados y la búsqueda de la respuesta al impulso que mejor se adapte a ellos.

c) Convergencia de las iteraciones

Este algoritmo se basa en iteraciones, aunque la aplicación sucesiva de los pasos de deconvolución y estimación de la respuesta al impulso converge rápidamente en la señal y el filtro óptimos.

A pesar de la posibilidad de obtener unas estimaciones óptimas tras realizar los dos pasos del algoritmo, el solapamiento entre los eventos implica que la optimización directa de las amplitudes y tiempos de los eventos podría converger en un resultado erróneo.

En general, también puede ser conveniente ejecutar el algoritmo en diferentes condiciones iniciales, con el fin de comprobar que el fenómeno en estudio no sea demasiado dependiente de artefactos que se hayan podido introducir durante la deconvolución. En el estudio realizado en [9], tras explorar diferentes condiciones iniciales con distintos conjuntos de datos, se concluyó que el algoritmo siempre convergía a las mismas soluciones.

d) Rendimiento en datos reales

En [9] primero se ha tratado de demostrar la capacidad del algoritmo para identificar EPSCs individuales en registros reales complejos. Para los registros largos y estables que se analizaron (Figura 1.13 A), el algoritmo recupera fácilmente la forma de onda característica de los EPSCs (Figura 1.13 B), que se representa matemáticamente como una suma de exponenciales.

La distribución de amplitud (Figura 1.14A) de los eventos detectados es similar a la distribución típica que se ha encontrado en la documentación y la distribución de los intervalos de tiempo entre eventos (Figura 1.14B) puede aproximarse mediante funciones exponenciales. Podemos ver los resultados de la prueba realizada en las figuras 1.13 y 1.14, donde se representan tres registros diferentes para los que se detectaron respectivamente 7799, 825, y 3800 EPSCs

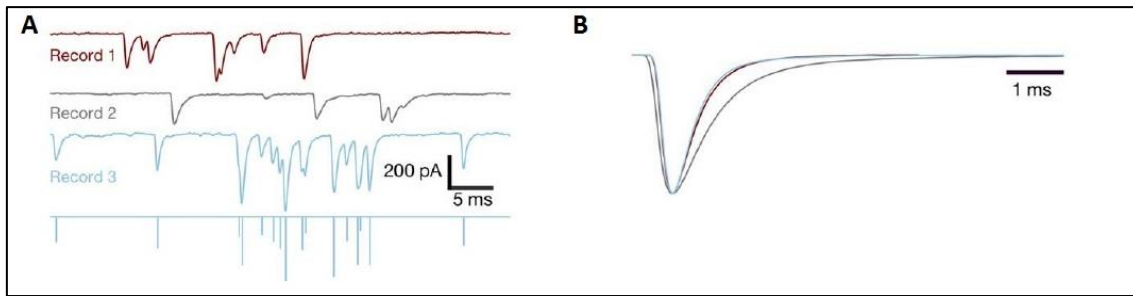


Figura 1.13. (A) Trazas de tres registros con eventos típicos. La traza inferior representa la deconvolución del tercer registro. (B) Estimaciones de la respuesta al impulso para cada registro que muestran la variabilidad en la cinética de los EPSCs. Fuente: [9]

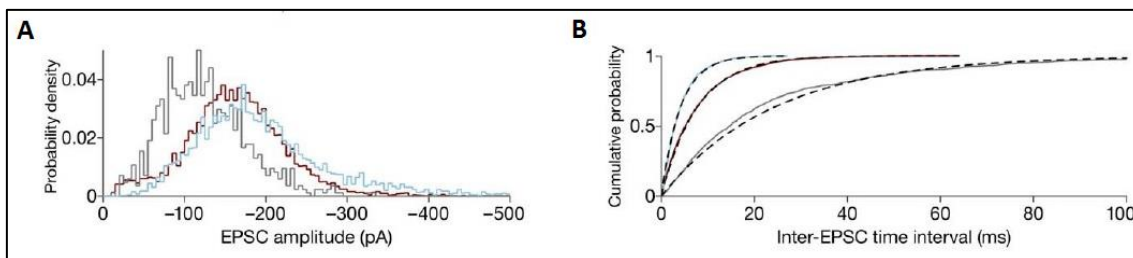


Figura 1.14. (A) Distribuciones de probabilidad de las amplitudes detectadas para los tres registros y (B) distribuciones de probabilidad de los tiempos entre eventos. Fuente: [9]

e) Rendimiento en datos simulados

También se ha probado el rendimiento del algoritmo mediante el análisis de datos simulados. Se han generado registros con EPSCs con amplitudes, tiempos, formas, y espectro de ruido completamente controlados y se ha probado la capacidad del algoritmo para extraer de un registro complejo la respuesta al impulso característica del EPSC, así como los tiempos y la amplitud de los eventos individuales. Además se ha comparado el rendimiento del algoritmo frente al de MiniAnalysis, un programa comercial que se ha citado anteriormente y es utilizado para el análisis de registros fisiológicos.

MiniAnalysis plantea un enfoque heurístico que proporciona flexibilidad a costa de requerir asistencia de un operador. Por el contrario, el enfoque global de la deconvolución es automático y permite que el programa se ejecute con menos intervención.

En [9] se ha tratado de demostrar la aplicabilidad del algoritmo de deconvolución con registros postsinápticos complejos. Tras realizar una serie de pruebas utilizando datos simulados, se concluye en el estudio que el algoritmo realiza favorablemente el análisis en comparación con el programa MiniAnalysis ampliamente utilizado.

Finalmente, en la herramienta final desarrollada en [9] se siguen distintas estrategias numéricas que permiten que el algoritmo se ejecute de manera eficiente. Aunque el algoritmo se basa en un modelo estadístico, en la aplicación se utilizan varias aproximaciones. Gracias a esto, la herramienta requiere un tiempo de ejecución aproximadamente proporcional al tamaño del registro que se pretende analizar. Por ejemplo, el análisis de un registro de unas $2 \cdot 10^5$ muestras requiere menos de un minuto en un equipo de trabajo de gama media, según las pruebas realizadas en [9].

Por otro lado, la mayoría de los algoritmos requieren parámetros por parte del usuario o ajustados de forma experimental. Este algoritmo realiza de forma automática la estimación de la respuesta de impulso, el ruido, y las constantes de regularización. Sin embargo, estos ajustes automáticos pueden ser anulados para los usuarios que requieren un control manual del algoritmo.

Además, la respuesta al impulso no necesita ser estimada a partir de los datos si se ha calculado en algún análisis anterior.

Una tarea complicada para los algoritmos en este tipo de análisis es separar los eventos múltiples superpuestos de los eventos individuales que simplemente tienen variabilidad en su cinética.

En definitiva, en [9] se ha desarrollado una herramienta flexible para realizar la deconvolución de series de datos de una forma eficiente y automatizada, donde los usuarios pueden determinar estadísticas de los eventos, su ubicación y amplitud, o las formas de onda de los eventos, incluso si estos se encuentran superpuestos y la relación señal ruido es baja.

1.6 Software para el análisis de señales biológicas: *Spike2*

Spike2 es un sistema de adquisición útil para la captura sencilla de datos o un paquete completo para el control de experimentos, el registro y el análisis y tiene además muchas ventajas sobre otros sistemas comerciales de adquisición de datos. Es un programa utilizado por los electrofisiólogos, para los que se va a desarrollar la herramienta de detección y análisis de eventos postsinápticos, para el registro de las señales que se pretende analizar. Toda la documentación sobre el funcionamiento del programa se ha obtenido de [12].

Incluye funciones de registro y análisis válidos para una amplia gama de aplicaciones de investigación. Es potente y flexible para las aplicaciones más exigentes, incluso para las que requieran equipos más especializados.

Tiene las configuraciones de muestreo y análisis que la mayoría de los investigadores necesitan y también dispone de un lenguaje de script incorporado que proporciona una flexibilidad mucho más allá de la disponible en muchos productos alternativos, lo que permite la automatización de tareas repetitivas y proporciona herramientas adicionales para los investigadores que prefieren describir sus propias rutinas.

Permite importar datos registrados por muchos otros sistemas, se trata de un programa versátil para analizar los datos existentes. Además se pueden exportar datos a hojas de cálculo, texto, archivos binarios y Matlab

Es una herramienta potente también para la representación gráfica de múltiples canales registrados en un mismo experimento, ya que permite situar cursores, ajustar el tiempo mostrado en cada canal y la amplitud de la señal representada para mostrar zonas de interés.

Spike2 identifica y clasifica la actividad de unidades individuales o múltiples, tanto on-line como off-line y permite marcar eventos utilizando umbrales simples o clasificar hasta 32 canales en línea con el ajuste de plantillas de trazas individuales sobre registros completos. Para registros múltiples, *Spike2* tiene herramientas para clasificar los eventos en base a la forma de onda. Por ejemplo, una combinación de emparejamiento de plantillas y separación en clases basado en el Análisis de Componentes Principales (PCA), a partir de medidas definidas por el usuario como correlaciones o errores.

También permite hacer detecciones o clasificaciones de forma interactiva, simplemente trazando una línea umbral a través de los picos de los eventos que se pretende identificar. ***Spike Collision Analysis*** es un complemento de *Spike2* que permite separar las colisiones mediante la comparación de la forma de onda de los eventos con pares de plantillas existentes.

Capítulo 2

Desarrollo de la herramienta *PS_Analysis* (PostSinaptic Analysis)

El objetivo de este proyecto es desarrollar una herramienta que sirva de ayuda a los electrofisiólogos en el análisis de registros neuronales, en concreto con señales intracelulares.

Actualmente, los electrofisiólogos a los que va destinada la herramienta realizan el análisis de los registros de forma manual. A partir de *Spike2* y mediante umbrales realizan las detecciones de eventos. Después extraen características de dichos eventos, como la amplitud o sus cinéticas. Esto implica un problema ya que la detección es muy tediosa y pueden encontrarse numerosos eventos erróneos, así como perder algunos correctos. Para ello se ha implementado *PS_Analysis* que complementará a *Spike2* en el trabajo de los electrofisiólogos. Se trata de una herramienta desarrollada en Matlab que permite analizar rápidamente registros neuronales y los resultados que se obtienen se pueden exportar a *Spike2*.

En general, los pasos que sigue el programa para analizar los registros se basan en algunos de los algoritmos explicados en el capítulo 1.

Primero, tras un procesamiento previo, se emplea una herramienta, *fade-master*, basada en un algoritmo de deconvolución ciega para reconstruir una señal de información, es decir, formada únicamente por los eventos del registro. Ya que esta reconstrucción no es del todo exacta y las grandes fluctuaciones de ruido pueden ser consideradas como eventos, se aplica a continuación otra herramienta, *detectPSPs*, basada en un algoritmo por umbral, calculado a partir del ruido de fondo del registro original, para realizar la detección correctamente. Una vez identificados los eventos en el registro original, se extraen características (amplitudes y cinéticas) que nos permitan realizar una clasificación. Esta clasificación se basa en un algoritmo típico, *kmeans*, que busca formar clases basadas en distancias.

Las herramientas *fade-master*, *detectPSPs* y *kmeans* que se utilizan en este programa se han elegido por los buenos resultados que permiten conseguir, por la facilidad de uso respecto a otras herramientas consultadas y por estar disponibles de forma libre en Matlab.

El diseño de la interfaz de usuario se ha basado en el de otra herramienta Desarrollada en un trabajo fin de carrera previo [13] en colaboración con el mismo equipo de electrofisiólogos al que va destinado para los mismos electrofisiólogos a los que va

destinado *PS_Analysis*. Se ha elegido este diseño porque facilita el trabajo y es sencillo de utilizar.

En la figura 2.1 se muestra el diagrama de bloques del programa. A continuación se detallan brevemente los bloques que forman la herramienta.

- **Data handling:** Bloque encargado del manejo de los datos, carga, guardado y exportación.
- **Plotting:** Representación de los datos y los eventos identificados.
- **Preprocess:** Procesamiento previo de la señal para mejorar los resultados, se aplican técnicas de filtrado y submuestreo.
- **Reconstruction:** Bloque que utiliza un algoritmo de deconvolución para reconstruir la señal de interés a partir del registro original o procesado.
- **Detection:** Identificación de los eventos mediante un algoritmo basado en derivadas y umbrales y extracción de características.
- **Classification:** Agrupamiento de los eventos detectados según los parámetros obtenidos con un algoritmo basado en distancias.

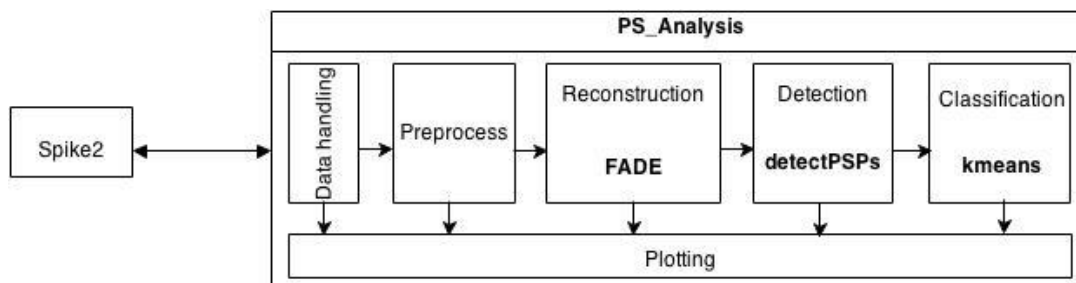


Figura 2.1. Diagrama de bloques del sistema *PS_Analysis*.

El usuario interactuará con el programa, que debe ejecutar desde Matlab, a través de la ventana principal (Figura 2.2) que se inicia con el comando *PS_Analysis*. Esta función se encuentra en la función principal del programa, *PS_Analysis.m*, a través del cual se accede a todas las funciones de la herramienta y donde se realiza la gestión de los bloques.

Otras funciones importantes que forman el programa son *FADE.m*, *detect_locs.m* y *parameters.m*.

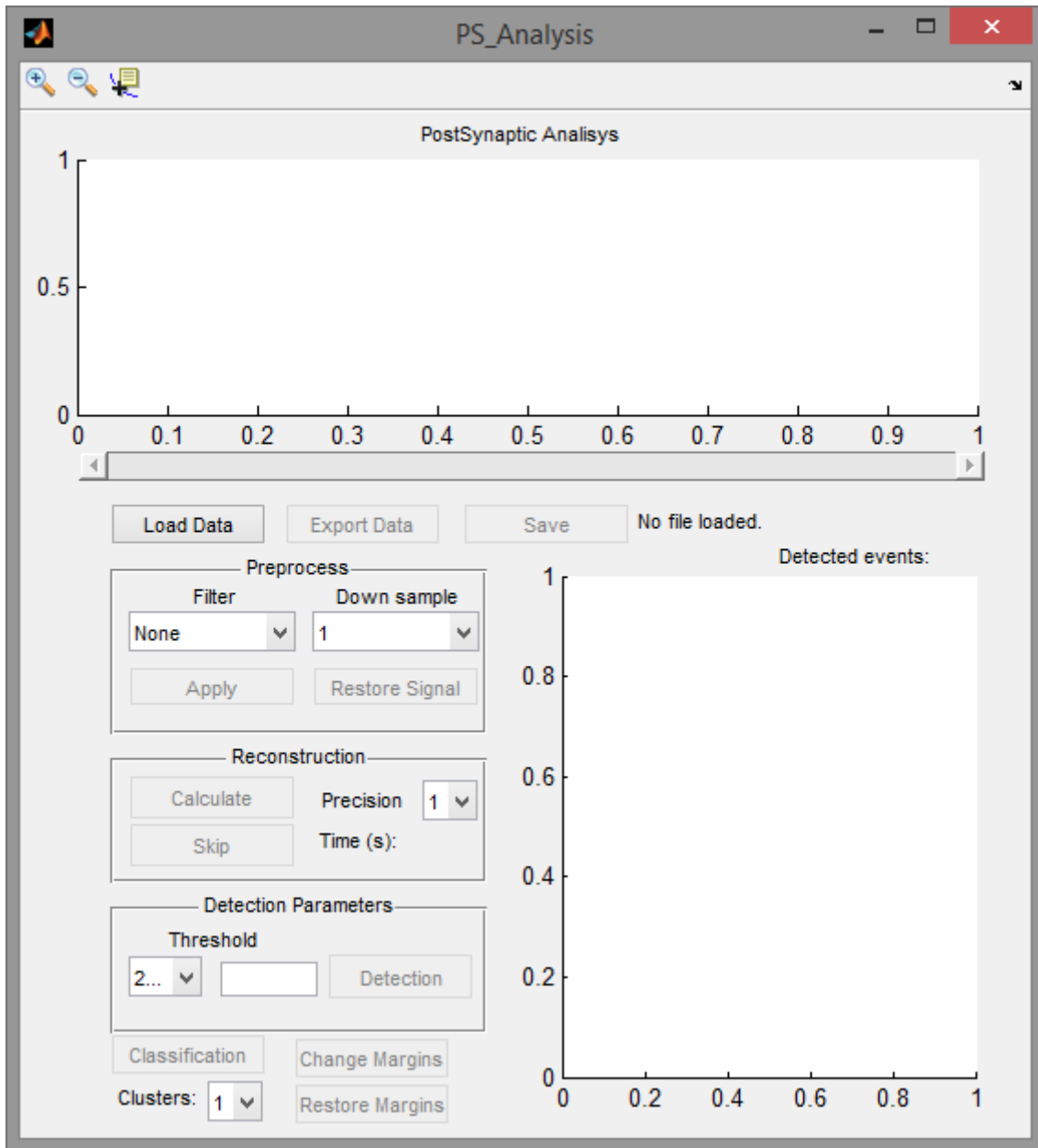


Figura 2.2. Ventana principal de la herramienta *PS_Analysis* sin datos cargados.

Desde esta interfaz gráfica el usuario gestionará todas las opciones del programa *PS_Analysis*. El programa diseñado tiene una ejecución secuencial sencilla de manejar gracias a la ventana principal. Puesto que la ejecución de los distintos bloques del programa debe seguir un orden, los botones de la interfaz de usuario, inicialmente deshabilitados, se irán activando según el usuario vaya pasando por cada uno de ellos.

2.1 Data handling

Este bloque se encarga de gestionar los datos de entrada y de salida del programa. Por una parte adapta los registros procedentes de *Spike2* cargados por el usuario en formato *.mat. Es importante que el usuario proporcione el fichero de forma adecuada para evitar errores en el sistema.

También permite exportar en formato *.txt los resultados obtenidos para poder trabajar de nuevo con ellos en *Spike2*, ya que es uno de los principales programas que utilizan los electrofisiólogos. Además se pueden guardar los resultados en tablas para trabajar con ellos sobre una hoja de cálculo.

Los botones de la ventana principal que forman este bloque son **“Load Data”**, **“Export Data”** y **“Save”**.

A continuación se describen las callbacks que se activan al pulsar los diferentes botones que aparecen en la ventana principal de la interfaz gráfica usuario, contenidas en *PS_Analysis.m*:

a) *load_data_Callback*

Se inicia al pulsar el botón **“Load Data”**. Primero desactiva todos los botones de la interfaz de usuario y elimina todas las representaciones de las gráficas. Después solicita al usuario el archivo *.mat que contiene el registro con la señal original exportado desde *Spike2*.

A continuación, guarda en la estructura *handles* las variables *folderpath*, *filename*, *original_data*, *data*, *to*, *units*, *ts*, *original_ts*, *Xlimit*, *Xmin* y *Xmax*.

Después activa el botón *Apply*, que permitirá aplicar las diferentes opciones del recuadro *Preprocess*.

Finalmente, llama a la función *Plot_d* para representar en la gráfica superior de la ventana principal la señal del registro cargado en valor absoluto y sin valor medio.

b) *save_Callback*

Esta función se inicia con el botón **“Save”**, permite guardar en una tabla datos obtenidos del análisis del registro. Tras la detección se calculan cuatro parámetros de cada evento, guardados en la variable *handles.params*, que servirán para la clasificación (amplitud, tiempo de subida, tiempo de bajada, máximo). Estos parámetros, junto con la posición temporal del evento se guardan en un fichero *.xls.

La estructura del fichero generado, donde cada fila corresponde con un evento, es la siguiente:

- La primera columna se corresponde con la amplitud.
- La segunda con el tiempo de subida, desde el inicio del evento hasta el máximo.
- La tercera con el tiempo de bajada, desde el máximo hasta el fin del evento.
- En la cuarta columna se representa el valor máximo del evento.
- Y en la última columna el instante de inicio del evento.

c) export_data_Callback

La opción de exportar los datos obtenidos en el análisis se inicia con el botón “**Export Data**”. Esta función es importante ya que nos va a permitir llevar nuestro trabajo a *Spike2*. Para que se active este botón debemos haber pasado por todos los bloques de la herramienta. La función carga los resultados obtenidos en la clasificación, que se encuentran en la variable *handles.I_aux*, solicita al usuario la carpeta de destino y el nombre que desea asignar al nuevo archivo *.txt de salida y genera el citado archivo.

El formato del archivo de salida es el siguiente:

En la primera línea, el período de muestreo de los eventos y su longitud en número de muestras. En el resto de líneas, donde cada fila se corresponde con un evento distinto, una matriz con el siguiente formato:

- En la primera columna, el grupo al que pertenece cada evento tras la clasificación.
- En la segunda columna, el instante de tiempo (en segundos) en el que se produce la primera muestra de cada evento.
- En el resto de columnas, las muestras que componen cada evento.

2.2 Plotting

Este bloque es el encargado de trazar gráficamente las señales que más interesan a los electrofisiólogos para el análisis. La representación de estas señales se realiza en las dos gráficas de la ventana principal y en una nueva ventana que se creará.

En la gráfica superior se traza la señal cargada, que se actualizará según el ajuste que se realice en los bloques *Preprocess* y *Reconstrucion* (cambios en el registro original o superposición de la señal reconstruida). En la gráfica inferior se representan superpuestos todos los eventos detectados por la herramienta, alineados correctamente.

Por último, en la ventana emergente se representan, en tantas gráficas como clases se hayan seleccionado en la opción “**Clusters**”, todos los eventos clasificados, a partir del método *kmeans*. Además se representa en la misma ventana una gráfica con los parámetros de cada evento que se han utilizado para la clasificación.

En la ventana principal tenemos dos botones que permiten interactuar con la gráfica inferior, son “**Change Margins**” y “**Restore Margins**”. Además en la gráfica superior hay una barra de desplazamiento que permite visualizar de una forma más cómoda registros muy largos.

La función que se encarga de manejar todas las representaciones citadas anteriormente es *Plot_d*, aunque también intervienen algunas callbacks que se describen a continuación:

a) *Plot_d*

Esta es la función más importante de este bloque y es llamada desde distintos botones.

Las representaciones, o cambios, en la gráfica superior se realizan con los botones "**Load Data**", "**Reconstruction**", "**Apply**" y "**Restore Signal**", además de la opción "*Threshold*".

En la gráfica inferior se representarán datos cuando se pulse el botón "Detection" y se ajustará el eje temporal con los botones "**Change Margins**" y "**Resore Margins**".

La ventana donde se representan los eventos clasificados se iniciará cuando se pulse el botón "**Classification**". A continuación, en las figuras 2.3 y 2.4 se muestran las diferentes gráficas y representaciones que tiene la herramienta.

Como se puede apreciar, esta función controla diferentes opciones, para organizar las acciones se utiliza una estructura *switch-case* con cuatro tipos de representación que se seleccionan en sus respectivas callbacks. Las opciones de la función son:

- El primer caso (Figura 2.3A) representa, en valor absoluto, los datos cargados por el usuario, originales o procesados, en la ventana superior de la interfaz. Además activa a barra de desplazamiento si la longitud del registro supera el valor de *handles.Xlimit*.
- En el segundo caso (Figura 2.3B) se representa la variable *handles.reconstruction* en la gráfica superior, solapada con la señal original. Esta traza es útil para comprobar si la señal resultante del bloque *Reconstruction* es válida para continuar con el proceso de análisis.
- El tercer caso (Figura 2.3C) permite trazar todos los eventos detectados en la gráfica inferior. Se representan los eventos en una ventana de 30 ms alineados respecto a los picos. Aunque el registro cargado se haya modificado con un filtro o un submuestreo en el bloque *Preprocess*, en esta opción se representan los eventos que se han detectado con las mismas muestras que los formaban en el registro original.
El procesamiento que se haga sobre el registro simplemente facilita la detección, los eventos que se clasifiquen, y sus características, deben proceder del registro original.
- En el cuarto caso (Figura 2.4) se representan los eventos clasificados en distintas clases. Además se presentan los parámetros que se han extraído de los eventos y han servido para la clasificación. Los eventos de cada gráfica se trazan con diferentes colores al igual que sus correspondientes parámetros.

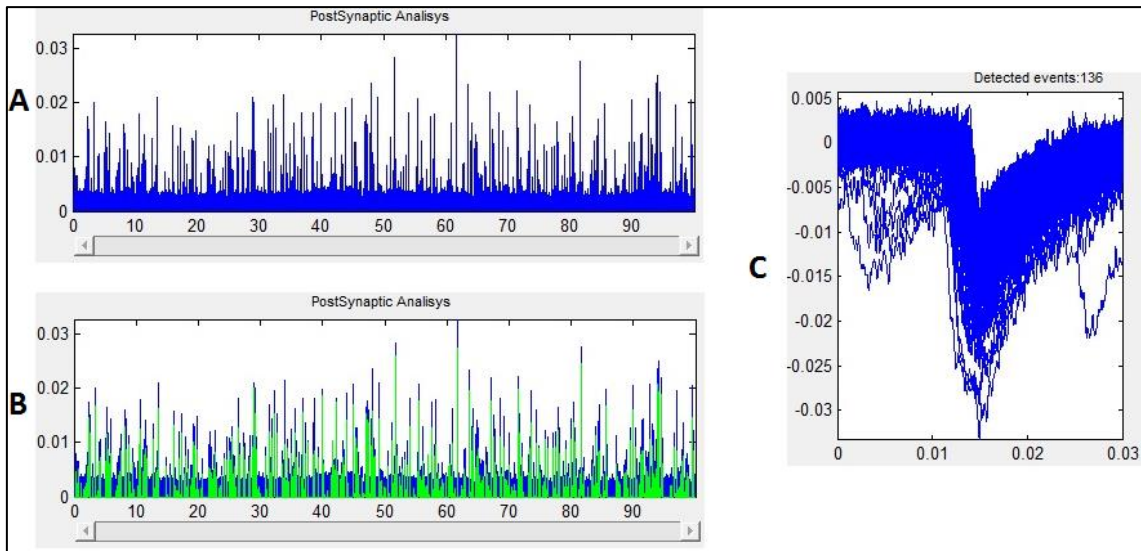


Figura 2.3. Gráficas de la herramienta PS_Analysis.

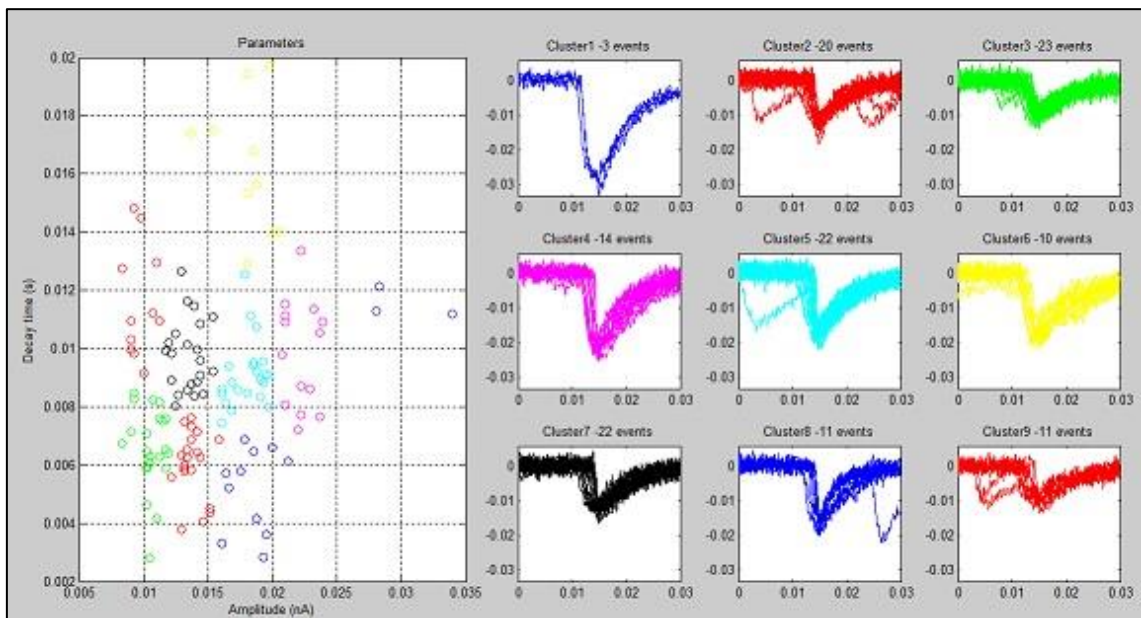


Figura 2.4. Gráficas con eventos clasificados y parámetros empleados PS_Analysis.

b) change_margins_Callback

Se activa al pulsar el botón **“Change Margins”**. Toma los valores de *thrr* y *thrl* fijados por el usuario manualmente en la gráfica inferior de la venta principal de la interfaz de usuario y ajusta el eje temporal. Habilita además el botón **“Restore Margins”** si está desactivado.

c) restore_margins_Callback

Se activa al pulsar el botón **“Restore Margins”** y revierte los cambios producidos por la función anterior. Ahora el eje temporal de la gráfica inferior tiene la configuración por defecto.

2.3 Preprocess

En este bloque se permite al usuario aplicar un procesamiento previo al registro de interés para facilitar la detección de los eventos. Este paso es importante, para registros con mucho ruido o una frecuencia de muestreo demasiado grande, pero no imprescindible. Sobre la señal original se puede aplicar un filtrado y/o un submuestreo según lo indique el usuario.

Las funciones de filtrado y submuestreo son propias de Matlab y en el código fuente de la herramienta se encuentran en *apply_Callback*, que es la función principal del bloque *Preprocess*.

Los botones de la ventana principal que forman este bloque son *“Apply”* y *“Restore Signal”*, además de los menús desplegables *“Filter”* y *“Down Sample”* que permiten al usuario configurar el preprocesamiento.

Estudiamos a continuación las funciones principales:

a) *apply_Callback*

Esta función se ejecuta cuando se pulsa el botón *“Apply”* pero antes deben configurarse las opciones de *Filter* y *Down Sample*.

Primero se elimina la componente continua de la señal original con la función *detrend* de Matlab. Después se filtra la señal, pero antes se comprueba la opción seleccionada en el menú desplegable. El tipo de filtro que se propone es un filtro de suavizado que proporciona Matlab [14], *smooth*. El usuario simplemente selecciona el método que utiliza la función *smooth* o puede elegir no aplicar ningún filtro (opción *None*). Los métodos disponibles en la herramienta se explican brevemente a continuación, estas especificaciones han sido extraídas de la ayuda de Matlab.

- *‘Moving’*: Media móvil. Cada nuevo punto es el promedio del subconjunto de los puntos contiguos. El número de puntos del subconjunto es el valor de *span*.
- *‘Lowess’*: Regresión local utilizando mínimos cuadrados lineales ponderados y un modelo polinomial de primer grado.
- *‘Loess’*: Regresión local utilizando mínimos cuadrados lineales ponderados y un modelo polinomial de segundo grado.
- *‘Sgolay’*: Filtro de Savitzky-Golay. Una media móvil generalizada con coeficientes determinados por un ajuste lineal por mínimos cuadrados y un modelo polinómico de un grado especificado.

La selección del método a emplear por la función *smooth* depende del registro que se esté analizando. No existe un criterio concreto para escoger un método u otro, sino que se propone utilizar aquel que, visualmente, suavice mejor la señal sin influir en los eventos de interés.

Se ha elegido este tipo de filtro porque es útil en señales bioeléctricas donde existe ruido de fondo. Además este tipo de filtrado es utilizado por los electrofisiólogos a los que va destinada la herramienta para analizar los registros.

Tras filtrar la señal se aplica un submuestreo, si la frecuencia de muestreo del registro es muy elevada es recomendable aplicarlo ya que el análisis será más rápido y los resultados no se verán afectados. Se utiliza la función *downsample* de Matlab y el usuario puede seleccionar en la interfaz gráfica de 2 a 10 muestras para realizar el submuestreo, si se selecciona el valor 1 la señal original no sufre cambios. El valor elegido en *downsample* se guardará en la variable *handles.nsamples* para utilizarlo más adelante. Puesto que hemos reducido el número de muestras de la señal es importante que actualicemos las variables *handles.ts* y *handles.sRate*, que representan el tiempo entre muestras y la frecuencia de muestreo respectivamente, pero guardaremos los parámetros iniciales (*handles.ts_original*) porque más adelante necesitaremos recuperar la señal original. Cada vez que se pulse el botón **“Apply”** los cambios se realizarán sobre el registro original, no sobre la señal ya procesada y se representará en la gráfica superior de la interfaz. Cuando se realiza el preprocesamiento se habilitan los botones **“Calculate”** y **“Skip”** para poder continuar con el análisis.

restore_signal_Callback

Esta función simplemente elimina la señal procesada con sus parámetros y representa la original en la gráfica superior. También se eliminan los eventos representados en la gráfica inferior.

Además deshabilita los botones siguientes para que el usuario repita el bloque de *Preprocess*.

2.4 Reconstruction

El bloque *Reconstruction* es la base del programa. En esta parte se utiliza la herramienta ***fade-master*** (Fast Automated Deconvolution of EPSCs), descargada de internet [15]. Esta herramienta implementa un algoritmo de deconvolución ciega, desarrollado en el punto 1.6 y detallado en [9].

Como se explicaba en el capítulo anterior, el propósito de este algoritmo es obtener, a partir de un registro intracelular, la señal que contenga la información de los eventos únicamente. Para ello alternaba entre dos pasos, estimar un filtro y realizar una deconvolución. Para poder implementar esta herramienta en el programa *PS_Analysis* ha sido necesario adaptar varios parámetros iniciales, que se explicarán más adelante, así como modificar la forma de trabajo para optimizar el tiempo de cómputo de ***fade-master***.

Los botones de la ventana principal que permiten al usuario interactuar con este bloque son **“Calculate”** y **“Skip”**, además del menú desplegable **“Precision”** que permite al usuario configurar el número de iteraciones que se realizan en la reconstrucción. Cuanto mayor sea el nivel de precisión requerido por el usuario, mayor será el tiempo que empleará la herramienta. Ya que el tiempo es un factor importante, en el recuadro *Reconstruction*, se ha incluido el indicador *Time (s)* que mostrará el tiempo que emplea el programa en realizar la reconstrucción. Este tiempo, como se citaba en el punto 1.6, depende de la longitud en muestras del registro a analizar, de la complejidad de la reconstrucción, que depende del ruido presente en el registro, y del nivel de precisión o número de iteraciones que se seleccione.

A continuación se describen las callbacks que se activan al pulsar los diferentes botones que aparecen en la ventana principal de la interfaz gráfica usuario, contenidas en *PS_Analysis.m*:

a) *calculate_reconstruction_callback*

En esta función se desarrolla todo el proceso de reconstrucción. Primero se definen varios parámetros iniciales, importantes para el correcto funcionamiento del programa:

- *FE*: Factor de Escala. Tras realizar pruebas con la herramienta ***fade-master***, se ha llegado a la conclusión de que es necesario aplicar un factor de escala, de 10^4 , a nuestros registros para que no haya errores en el programa.
- *size_data*: Tamaño máximo, en muestras, de la señal que queremos analizar. Tras diversas ejecuciones se ha comprobado que el algoritmo trabaja mejor y más rápido con señales de, como máximo, $2 \cdot 10^5$ muestras. Por eso, si la señal que queremos analizar es mayor se segmenta para conseguir mejores resultados.
- *size_obj*: Tamaño máximo, en muestras, del fragmento de señal que se va a utilizar para estimar el filtro. Para que la estimación sea óptima y precisa se toma un fragmento de 10^4 muestras, o un segundo de duración si la frecuencia de muestreo (*handles.sRate*) es menor que *size_obj*, de la señal que queremos analizar.

Para implementar esta función se ha seguido un script de ejemplo incluido en ***fade-master*** que ayuda a entender el funcionamiento de esta herramienta.

Primero se realiza la estimación de un filtro, con la función *FADE* integrada en la herramienta ***fade-master***, necesario para la deconvolución de la señal de interés. Se utiliza un segmento de señal de *size_obj* muestras o de 1 segundo de duración. Para la estimación del filtro se realizan 6 iteraciones, que garantizan una buena estimación. También es importante en la estimación que el segmento de la señal contenga eventos identificables y poco ruido.

Después se calcula la reconstrucción de la señal, también con la función *FADE* pero esta vez con un filtro estimado. Si la señal tiene más muestras que *size_data* se segmenta para agilizar el proceso. Se realizarán tantas reconstrucciones como segmentaciones se hagan en la señal y después se reagruparán para formar una única reconstrucción correspondiente a la señal completa que estamos analizando. En el proceso se iterará tantas veces como se indique en el menú desplegable *Precision*.

Durante este proceso aparecerán dos ventanas propias del programa (Figura 2.5). La primera corresponde a la estimación del filtro y la segunda muestra la deconvolución de la señal a partir del filtro estimado en el proceso anterior.

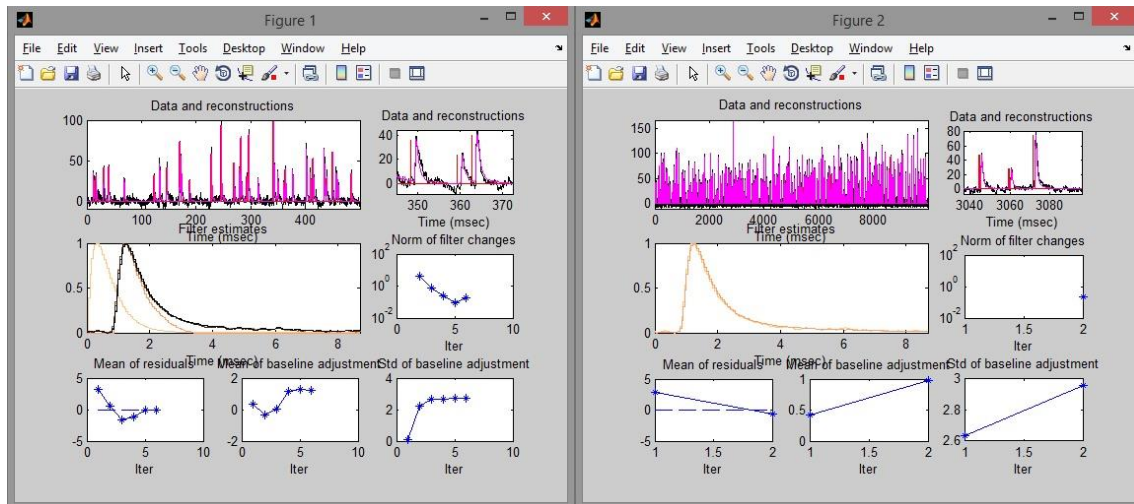


Figura 2.5. Gráficas del proceso de reconstrucción con registros de prueba. Fuente: *FADE*.

La señal reconstruida se guardará en la variable *handles.reconstruction* y también es importante guardar *handles.residual* que contiene información del ruido de fondo del registro. Antes se debe dividir estas variables por el fondo de escala (*FE*) utilizado.

Por último, se guardan varios parámetros de interés como *handles.rec_time* que nos indica el tiempo que se ha empleado en la reconstrucción y *handles.sigma*, que se obtiene a partir de *handles.residual* y representa la desviación estándar del ruido, este parámetro será importante para la detección por umbral.

Finalmente se llama a la función *Plot_d* y se representa, en la gráfica superior, la señal reconstruida sobre la señal original o procesada.

b) skip_reconstruction_callback

Esta función permite al usuario evitar el proceso de reconstrucción. Puede ser útil para registros con bajos niveles de ruido y eventos muy bien definidos. El valor de *handles.sigma* se obtiene ahora de la distribución estándar del registro.

2.5 Detection

En este bloque se realizará la detección y extracción de parámetros de los eventos presentes en el registro. Para la detección de los eventos se utiliza la herramienta *detectPSPs*, descargada desde *Mathworks* [16] y modificada para el programa bajo el nombre *detect_locs*. Para la extracción de características se ha creado una función llamada *parameters*.

En la ventana principal tenemos un botón, un menú desplegable y una caja de texto que permiten controlar la detección que se representará en la gráfica inferior, son ***“Detection”*** y ***“Threshold”***. Hasta que no se selecciona una opción en el menú desplegable, no se activa el botón que inicia la detección.

A continuación se describen las callbacks que se activan al pulsar los botones del recuadro *Detection* de la interfaz de usuario y las funciones creadas para detección y extracción de características que se lanzan desde las callbacks:

detectPSPs

La herramienta ***detectPSPs*** está concebida como un sistema completo de detección de eventos postsinápticos. Básicamente la señal se filtra y después se utiliza su derivada, también filtrada. En los comentarios del código fuente de [16] podemos ver los pasos que sigue: se crea una señal, a partir de la derivada de la señal inicial, en la que cada punto es igual a la suma del punto anterior y su derivada sólo si la derivada es menor que cero (para eventos descendentes), si no la función es cero en ese punto. Después se busca los puntos donde la derivada de esta nueva señal cambia de signo y si estos puntos superan cierto umbral se puede considerar que es el inicio de un evento.

Una vez encontrados los puntos de inicio de cada evento del registro se trata de conseguir una aproximación no lineal de cada evento que facilite el análisis. Por último se calculan cuatro parámetros característicos -amplitud, tiempo de subida, tiempo de bajada, es decir, la cinética del evento y un parámetro de aproximación- de cada evento identificado a partir de la función no lineal ajustada al evento.

El procedimiento que realiza esta herramienta es algo complicado y lento, debido a la aproximación no lineal de cada evento, por eso se ha adaptado para nuestro programa.

En ***detect_locs*** simplemente se proporciona como parámetro de entrada la variable *handles* del programa *PS_Analysis* y se sigue el mismo procedimiento que en ***detectPSPs***, salvo que, en vez de aproximar funciones a los eventos y calcular sus parámetros, devuelve los instantes de inicio y de pico de cada evento detectado, que se guardan en las variables *whereStarts* y *wherePeaks*. De esta forma se ha conseguido detectar eventos siguiendo el método utilizado en ***detectPSPs*** y se ha evitado el cálculo de funciones no lineales que ralentizaba el proceso.

Hay que tener en cuenta que con este método de derivadas se realizan detecciones de eventos superpuestos, ya que considera cambios en las pendientes de la señal para realizar las detecciones.

d_thr_Callback

Esta función gestiona el menú desplegable *Threshold* y proporciona el valor del umbral que se utilizará para decidir si los eventos identificados por ***detect_locs*** son correctos o son fluctuaciones de ruido.

Hay cuatro opciones para elegir el valor del umbral *handles.thr*:

- Que tome un valor de $2\cdot\sigma$ ($2\cdot handles.sigma$), es decir, dos veces la desviación estándar del ruido de fondo.
- Valor $3\cdot\sigma$, más restrictivo.
- Elegir del umbral manualmente sobre la gráfica superior de la ventana principal.
- Introducir el valor exacto del umbral deseado en la caja de texto, controlado con la función ***edit_thr_Callback***.

Cuando se seleccione el umbral se dibujará una línea horizontal en la gráfica superior que representa el valor de *handles.thr* elegido.

detection_Callback

Esta función primero se encarga de llamar a la función ***detect_locs*** para calcular los puntos de inicio y pico de los posibles eventos, almacenados en *whereStarts* y *wherePeaks*. Después recorre todos los posibles eventos detectados, considerando eventos correctos aquellos cuya amplitud (diferencia entre el valor de pico y el valor del punto de inicio) sea mayor que el umbral seleccionado en ***d_thr_Callback*** y guarda en *handles.starts* los instantes de inicio de los eventos considerados como correctos.

A continuación invoca a la función ***parameters***, guarda las características extraídas en *handles.params*, llama a *Plot_d* para representar los eventos detectados en la gráfica inferior de la ventana del programa y actualiza la cadena de texto "*Detected events*" situada sobre la gráfica inferior con el número de eventos detectados.

parameters

Con esta función se extraen los parámetros de los eventos que nos van a permitir realizar una clasificación o son de interés para los electrofisiólogos. Primero se calculan los instantes de inicio de los eventos detectados sobre el registro original. Si no se ha hecho ningún submuestreo en la señal, coinciden con *handles.starts*, sino se corresponden con *handles.starts*handles.nsamples*, siendo *handles.nsamples* el valor del submuestreo realizado.

Después se calculan los parámetros indicados por los electrofisiólogos: la amplitud del evento, como diferencia entre el valor en el inicio y el máximo, el valor de pico, los tiempos de subida y bajada con las funciones *risetime* y *falltime* respectivamente, estas funciones son propias de Matlab y tienen en cuenta la diferencia en tiempo o muestras que hay entre el instante de inicio o fin del evento hasta el máximo. Para evitar errores, en el cálculo de parámetros se han acotado los eventos en ventanas, de máximo 8 ms, que van desde el inicio de un evento hasta el inicio del siguiente.

2.6 Classification

Este último bloque es el encargado de clasificar los eventos detectados correctamente en distintas clases según los parámetros extraídos en *parameters*. El usuario debe indicar el número de grupos en los que se van a clasificar los eventos.

Para controlar el bloque *Classification* se utilizará el botón "*Classification*" y el menú desplegable "*Clusters*" de la interfaz de usuario.

A continuación se describen las callbacks y las funciones que intervienen en el desarrollo de este bloque:

Kmeans

Según se describe en [17] es un método de agrupamiento basado en distancias, que tiene como objetivo la clasificación de n objetos en k grupos en el que cada objeto pertenece al grupo más cercano al centroide que representa a cada clase. El algoritmo trata de minimizar suma de las distancias de los objetos al centroide de la clase a las que pertenezcan y se debe conocer el número, k , de clases.

Se ha utilizado la función *kmeans* de Matlab para la clasificación y los parámetros extraídos de cada evento son: amplitud, tiempo de subida y tiempo de bajada. Por defecto se utilizarán la amplitud y el tiempo de bajada, ya que según el criterio de los electrofisiólogos son los más discriminatorios, pero se pueden utilizar las tres características o combinarlas.

La función *kmeans* recibe como parámetros de entrada: una matriz con las características extraídas de los eventos, la variable *handles.k* donde el usuario indica el número de clases y que toma el valor del menú desplegable "*Clusters*" y el parámetro '*Replicates*' donde se configura el número de veces que se repite la clasificación, cada vez con un nuevo conjunto de centroides iniciales. Esto es importante ya que los centroides iniciales se eligen aleatoriamente y se van actualizando. El resultado de la clasificación puede variar dependiendo de la disposición inicial, pero si se introduce la opción '*Replicates*', con un valor de 5 por ejemplo, se ha comprobado que las distintas agrupaciones terminan convergiendo en la misma solución. La función devuelve la variable *handles.label* que contiene únicamente los índices de la clase a la que pertenece de cada evento.

classification_Callback

Esta función se activa al pulsar el botón "*Classification*" y se encarga de agrupar los parámetros que se van a utilizar en una matriz y llamar a la función *kmeans*.

Después de realizar la clasificación y a partir de *handles.label*, se genera la variable *handles.I_aux*, que contendrá todas las muestras de cada evento con su índice de clase correspondiente. Esta variable es útil para representar los eventos de cada grupo en distintas gráficas cuando se invoca a la función *Plot_d*. Por último se habilita el botón "*Export*" de la interfaz.

Capítulo 3

Resultados

En este capítulo se presentarán los resultados obtenidos tras analizar algunos registros con la herramienta *PS_Analysis*. Se utilizarán diferentes configuraciones del programa sobre un mismo registro para poder apreciar los efectos de los cambios. Además se compararán los resultados de este registro concreto con el análisis visual realizado por los técnicos y con un análisis hecho con *Spike2*. Este análisis consiste en aplicar un suavizado a la señal y derivarla, después se fija un umbral manualmente para detectar eventos. Con esta comparación se pretende extraer un porcentaje de detecciones correctas (que determine la fiabilidad de la herramienta), así como una tasa de falsos positivos que nos permita determinar que método es mejor para el análisis de estos registros.

También hay que destacar que la validez de esta herramienta no sólo se mide por los resultados obtenidos, sino por su utilidad para los electrofisiólogos y por el tiempo que les ahorrará cuando trabajen con ella.

Las pruebas han sido realizadas con registros intracelulares reales obtenidos por los electrofisiólogos. Aunque los registros capturados tengan una duración de 5 a 60 minutos, para el análisis con el programa se han utilizado segmentos del registro de una duración de 100 a 400 segundos. Los segmentos analizados no han sido elegidos al azar, sino que los han indicado los electrofisiólogos por ser zonas de interés que contenían los distintos eventos que se pretende estudiar.

3.1 Evaluación de las diferentes configuraciones de *PS_Analysis*

El registro a analizar será “2-11-10-cel1-4”, un registro intracelular, en corriente, de 3900 segundos de duración donde los electrofisiólogos han realizado un proceso de control al principio del registro y después han aplicado un fármaco para analizar los efectos. En este registro, previamente etiquetado por expertos, encontraremos mEPSCs típicos con un amplio rango de amplitudes y zonas con eventos superpuestos. En la siguiente gráfica (Figura 3.1) se muestra el registro completo en *nA* y, marcado en rojo, una ampliación (Figura 3.2) de 5 segundos para que se aprecien con más detalle los eventos. A simple vista se podría determinar que hay más de 30 eventos en este fragmento, pero esta estimación debe ser comprobada por un experto, ya que muchos eventos pueden ser confundidos con fluctuaciones del ruido o viceversa. Este fragmento nos da una idea también de la dificultad de analizar este tipo de registros, en sólo 5 segundos de señal encontramos numerosos eventos, algunos de ellos complicados de identificar, posiblemente solapados y con presencia de ruido.

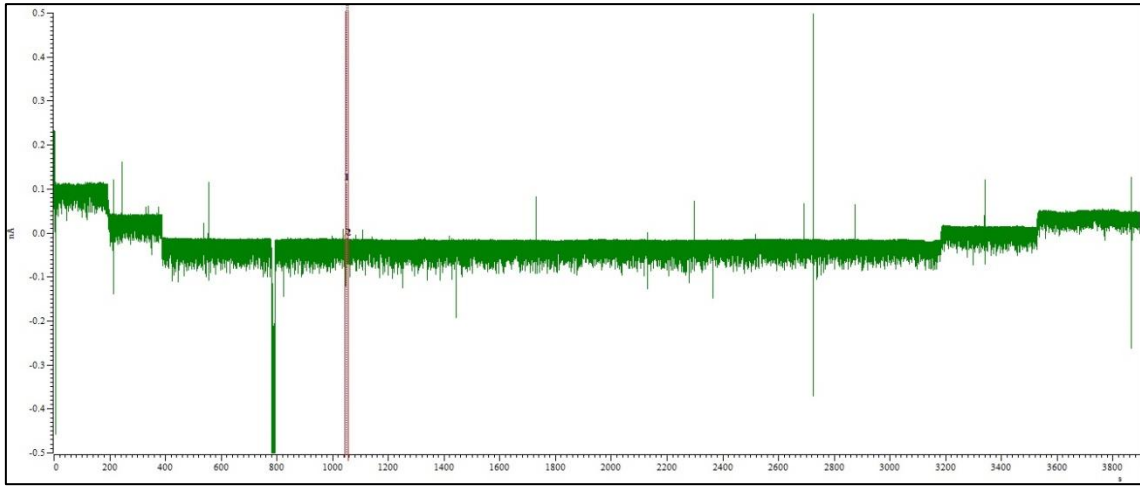


Figura 3.1. Registro 2-11-10-cel1-4 completo. Fuente: *Spike2*.

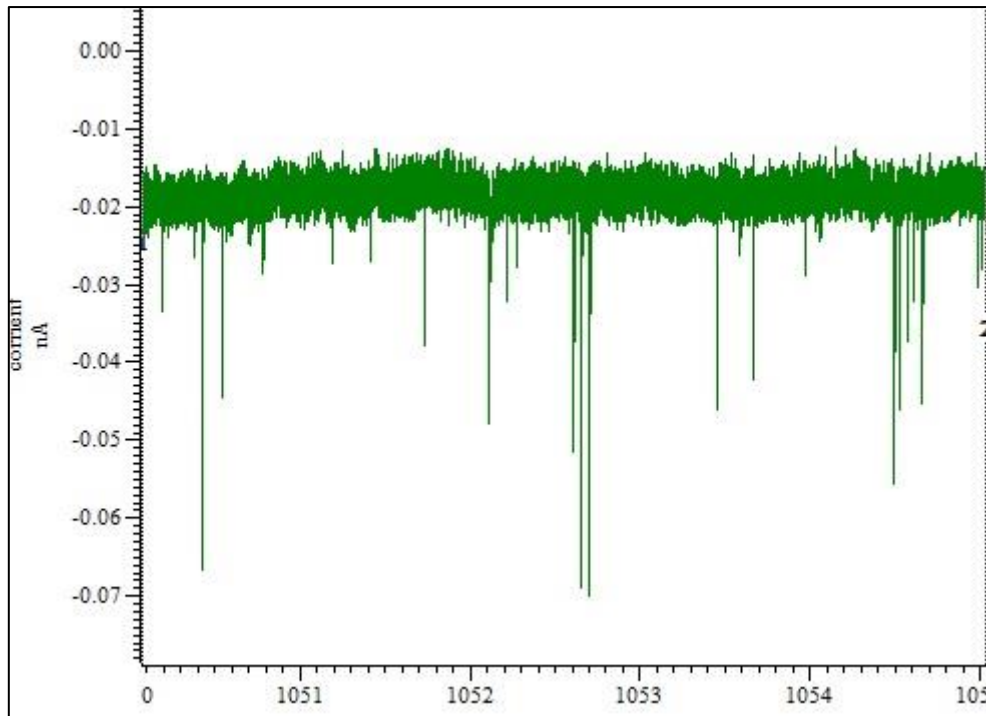


Figura 3.2. Ampliación del registro 2-11-10-cel1-4 en una gráfica de 5 s. Fuente: *Spike2*.

Estas capturas se han obtenido de *Spike2*, a partir de los cursores 1 y 2 (marcados en rojo en la figura 3.1) se ha ampliado la señal. Aunque desde *Spike2* se pueden procesar las señales, con filtros, suavizados o derivadas por ejemplo, durante las pruebas realizadas con *PS_Analysis* no se ha aplicado ningún cambio con *Spike2* sobre el registro original.

Para realizar las primeras pruebas se ha exportado desde *Spike2* un fragmento de 100 segundos de señal del registro "2-11-10-cel1-4".

En concreto, se ha exportado el fragmento de 1000 a 1100 segundos del registro, lo que significa en Matlab una señal de 1250000 muestras ya que el tiempo entre muestras es de 0.08 ms. Las gráficas que se muestran a continuación son capturas de la interfaz de usuario, la amplitud de la señal tiene unidades de nA y el tiempo se representa en segundos. Todas las pruebas se han realizado con un submuestreo de 8 muestras, por tanto la señal procesada tendrá 156250 muestras, lo que agiliza el análisis con *PS_Analisis*.

A continuación se detallan las diferentes pruebas realizadas. Para cada prueba se indica el tipo de filtrado aplicado, el nivel de precisión (número de iteraciones) de la reconstrucción y el tiempo empleado, los valores umbral $2\cdot\sigma$ y $3\cdot\sigma$ que dependen del ruido de fondo de registro, el número de eventos detectados y el porcentaje de detecciones comparado con el registro etiquetado por los expertos. Además se muestra en gráficas el ajuste de la reconstrucción a la señal que se está analizando.

El análisis del registro etiquetado visualmente por los electrofisiólogos será la base para comprobar la fiabilidad de los resultados obtenidos, en total se han etiquetado 614 eventos. Por otro lado, en la búsqueda realizada con *Spike2* se han obtenido 695 detecciones. Es útil comparar también la detección de *Spike2* con el etiquetado visual para apreciar las ventajas que puede ofrecer, en cuanto a los resultados, la herramienta *PS_Analisis*.

Para la evaluación de la herramienta, únicamente se han comparado los métodos que conseguían un número de detecciones similar al conseguido tras el análisis visual realizado por los electrofisiólogos. Por eso, en las tablas que se muestran a continuación sólo se ha calculado un porcentaje de detecciones correctas para las configuraciones: 'None' con precisión 2, 'Moving' con precisión 1 y 2 y 'Lowess' con precisión 2.

Para contrastar los resultados se han comparado uno por uno todos los eventos detectados en cada una de estas configuraciones con el registro etiquetado que sirve de base. También se han comparado las detecciones de *Spike2*, así se ha obtenido la tasa de falsos positivos y de detecciones perdidas de cada método.

3.1.1 Pruebas sin filtrado ('None')

Primero se realizan las pruebas sin aplicar ningún suavizado a la señal original. La reconstrucción se hace relativamente rápido como se muestra en la tabla 1 pero es bastante mala (Figura 3.3), los eventos de la reconstrucción tienen poca amplitud. Además los valores de **Threshold** son muy altos y esto implica que se detecten pocos eventos.

Precisión	Tiempo (s)	Threshold (nA)	Nº de detecciones	Detectado (%)	
1	94.93	2·σ	0.003669	440	-
		3·σ	0.005509	374	-

Tabla 1. Resultados del análisis sin filtrado con nivel 1 de precisión.

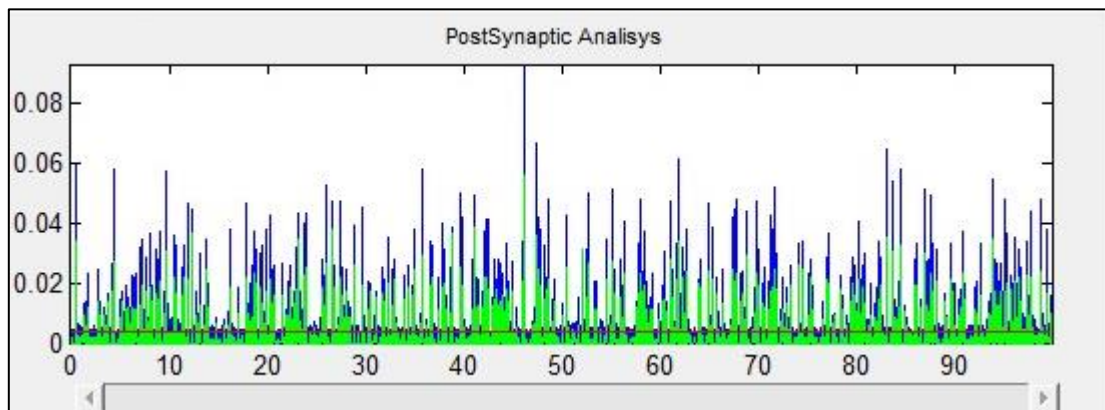


Figura 3.3. Reconstrucción sin filtrado con nivel 1 de precisión.

Al aumentar la precisión se mejora la reconstrucción, a costa de que aumente el tiempo de ejecución, y por tanto se producen más detecciones. Si el umbral es muy bajo, el ruido se puede interpretar como eventos, ya que no se ha filtrado. En este caso es probable que haya bastantes falsos positivos.

Precisión	Tiempo (s)	Threshold (nA)	Nº de detecciones	Detectado (%)	
2	129	2·σ	0.002958	644	92.19%
		3·σ	0.004442	535	-

Tabla 2. Resultados del análisis sin filtrado con nivel 2 de precisión.

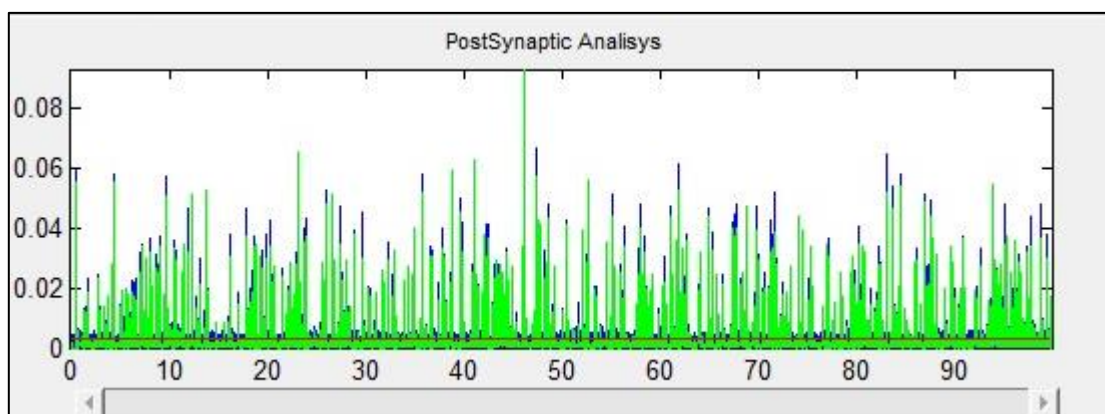


Figura 3.4. Reconstrucción sin filtrado con nivel 2 de precisión.

3.1.2 Pruebas con filtrado de media móvil ('Moving')

En este caso se utiliza un filtro para suavizar la señal. La reconstrucción es mejor que antes pero no es idónea, aunque el número de detecciones es considerable. Cabe añadir que al haber suavizado la señal, la reconstrucción ha sido mucho más rápida. El ruido se ha suavizado y por tanto σ , como se indica en la tabla 3, tiene un valor más pequeño. Si **Threshold** es muy pequeño aumenta la probabilidad de detectar falsos positivos (FP).

Precisión	Tiempo (s)	Threshold (nA)		Nº de detecciones	Detectado (%)
1	66.6	$2\cdot\sigma$	0.002037	619	92.51%
		$3\cdot\sigma$	0.003056	554	-

Tabla 3. Resultados del análisis con 'Moving' y nivel 1 de precisión.

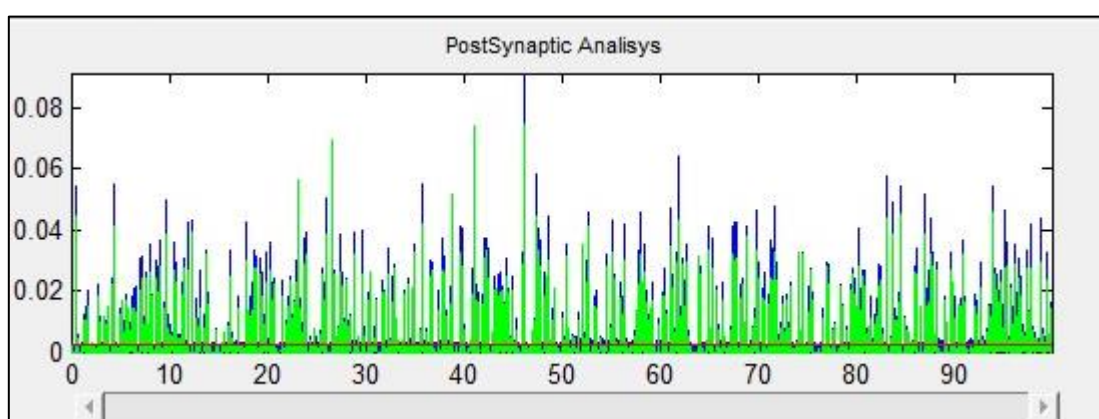


Figura 3.5. Reconstrucción con filtro de media móvil y nivel 1 de precisión.

Con más precisión conseguimos una reconstrucción (figura 3.6) casi perfecta pero se debe prestar atención al valor del umbral.

Precisión	Tiempo (s)	Threshold (nA)		Nº de detecciones	Detectado (%)
2	200.26	$2\cdot\sigma$	0.002487	636	93.49%
		$3\cdot\sigma$	0.003731	544	-

Tabla 4. Resultados del análisis con 'Moving' y nivel 2 de precisión.

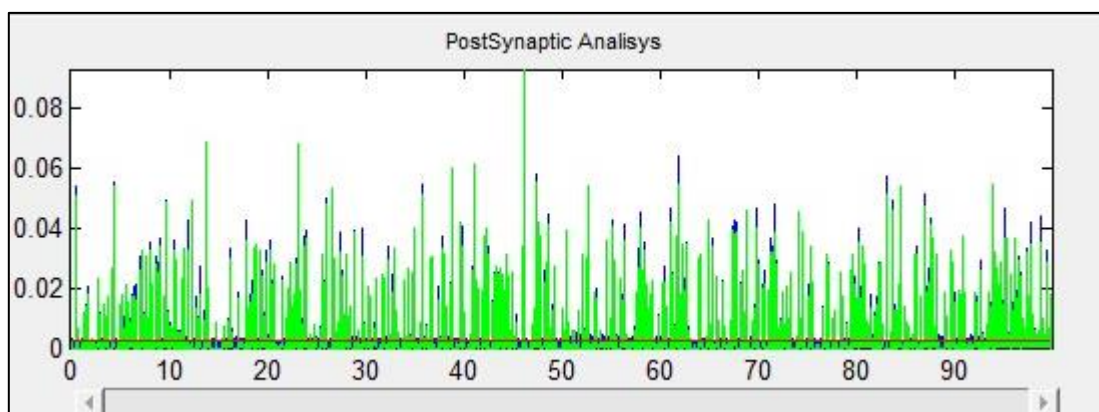


Figura 3.6. Reconstrucción con filtro de media móvil y nivel 2 de precisión.

3.1.3 Pruebas con filtrado 'Lowess'

Con este método de suavizado se consigue una reconstrucción (figura 3.7) similar a la del apartado anterior pero el valor de σ es mayor porque no se ha conseguido filtrar tanto el ruido, por eso el número de detecciones es menor (tabla 5).

Precisión	Tiempo (s)	Threshold (nA)		Nº de detecciones	Detectado (%)
1	112.42	$2\cdot\sigma$	0.00303	545	-
		$3\cdot\sigma$	0.00454	475	-

Tabla 5. Resultados del análisis con 'Lowess' y nivel 1 de precisión.

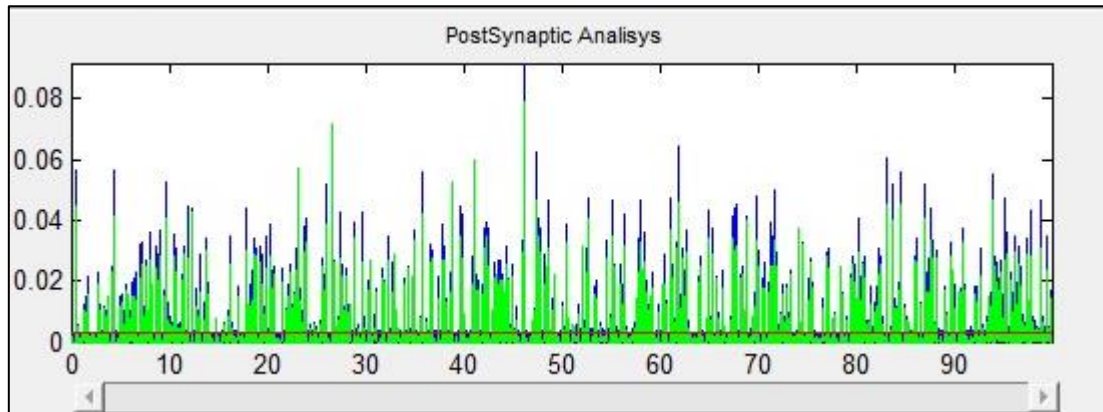


Figura 3.7. Reconstrucción con filtro 'Lowess' y nivel 1 de precisión.

Nuevamente al aumentar la precisión se mejora la reconstrucción utilizando este método, a pesar de que el tiempo aumenta aunque no es crítico. La detección de eventos es aceptable pero el valor de σ es muy pequeño. Esto implica que se pueden detectar más fluctuaciones de ruido como eventos, por eso hay más detecciones.

Precisión	Tiempo (s)	Threshold (nA)		Nº de detecciones	Detectado (%)
2	140.59	$2\cdot\sigma$	0.002077	679	-
		$3\cdot\sigma$	0.003115	588	91.37%

Tabla 6. Resultados del análisis con 'Lowess' y nivel 2 de precisión.

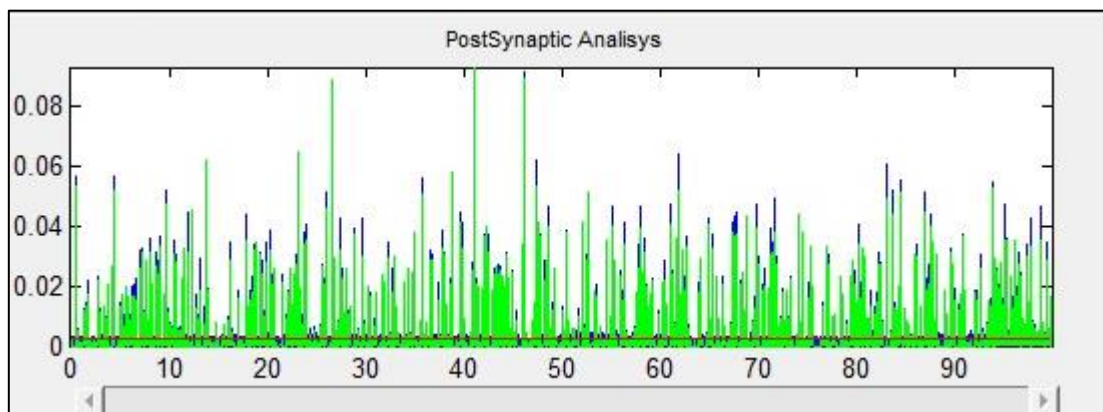


Figura 3.8. Reconstrucción con filtro 'Lowess' y nivel 2 de precisión.

3.1.4 Pruebas con filtrado 'Loess'

Con este método de suavizado la reconstrucción es más rápida pero no se ajusta a la señal. Además los valores de **Threshold** son grandes y se pierden bastantes eventos en la detección.

Precisión	Tiempo (s)	Threshold (nA)		Nº de detecciones	Detectado (%)
1	85.68	$2\cdot\sigma$	0.002914	519	-
		$3\cdot\sigma$	0.004371	475	-

Tabla 7. Resultados del análisis con 'Loess' y nivel 1 de precisión.

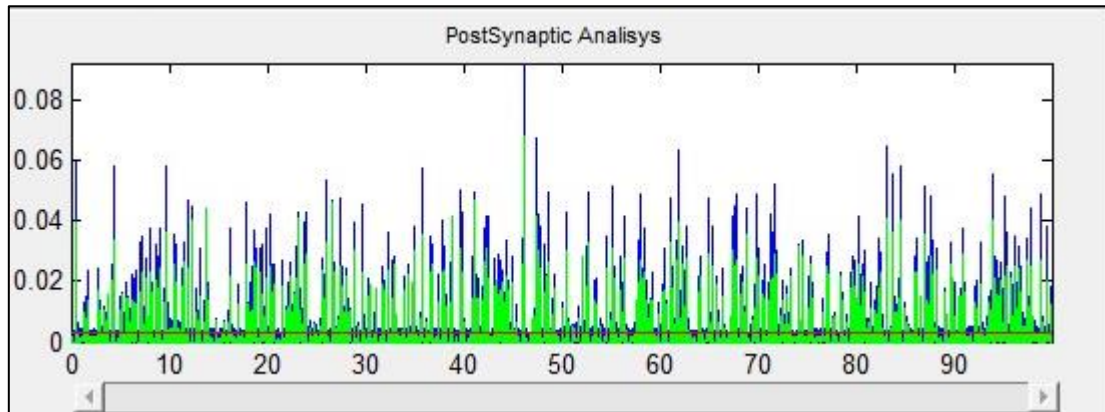


Figura 3.9. Reconstrucción con filtro 'Loess' y nivel 1 de precisión.

Al aumentar la precisión se mejora la reconstrucción pero no es idónea, σ sigue siendo grande porque no se ha filtrado el ruido correctamente y el número de detecciones aumenta respecto al caso anterior pero sigue habiendo pérdidas de eventos.

Precisión	Tiempo (s)	Threshold (nA)		Nº de detecciones	Detectado (%)
2	112.85	$2\cdot\sigma$	0.003617	576	-
		$3\cdot\sigma$	0.005426	499	-

Tabla 8. Resultados del análisis con 'Loess' y nivel 2 de precisión.

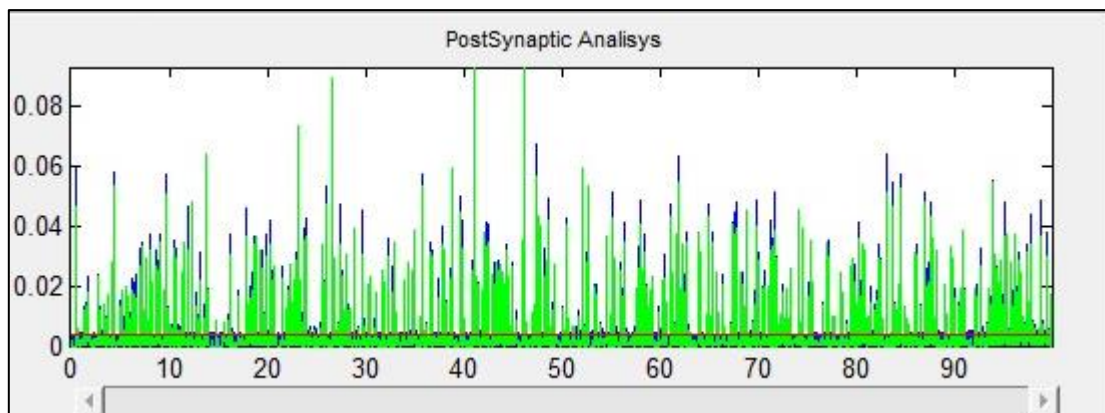


Figura 3.10. Reconstrucción con filtro 'Loess' y nivel 2 de precisión.

3.1.5 Pruebas con filtrado 'Sgolay' (Savitzky-Golay)

En este caso la reconstrucción es bastante pobre y al igual que en el apartado anterior, hay pérdidas en la detección porque σ tiene un valor muy grande.

Precisión	Tiempo (s)	Threshold (nA)	Nº de detecciones	Detectado (%)	
1	98.79	$2\cdot\sigma$	0.002767	531	-
		$3\cdot\sigma$	0.004150	471	-

Tabla 9. Resultados del análisis con el método 'Savitzky-Golay' y nivel 1 de precisión.

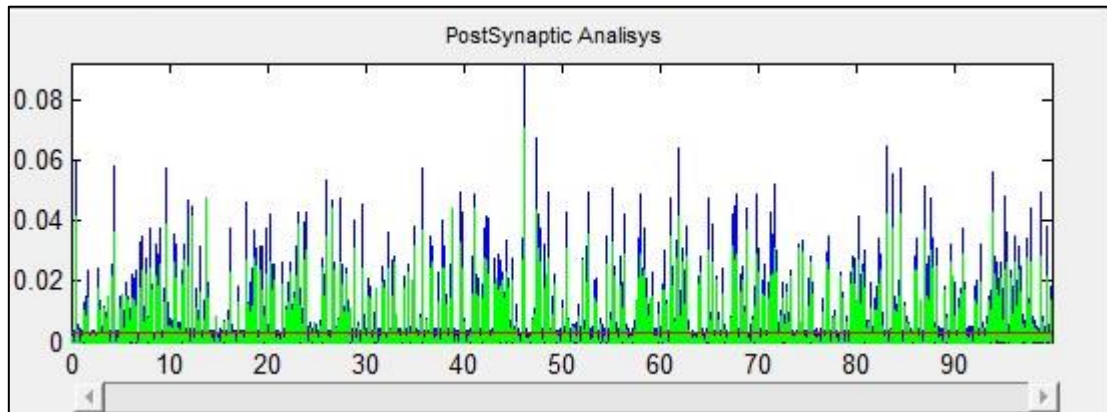


Figura 3.11. Reconstrucción con el método de 'Savitzky-Golay' y nivel 1 de precisión.

Ahora la reconstrucción ha mejorado considerablemente pero también se ha incrementado el tiempo empleado. El número de detecciones ha aumentado pero sigue sin ser válido.

Precisión	Tiempo (s)	Threshold (nA)	Nº de detecciones	Detectado (%)	
2	128.01	$2\cdot\sigma$	0.003628	583	-
		$3\cdot\sigma$	0.005442	501	-

Tabla 10. Resultados del análisis con el método 'Savitzky-Golay' y nivel 2 de precisión.

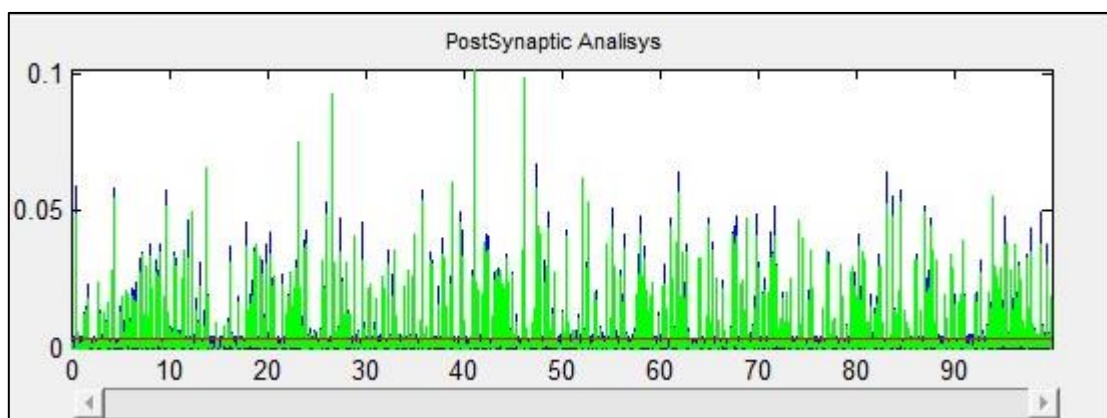


Figura 3.12. Reconstrucción con filtro de 'Savitzky-Golay' y nivel 2 de precisión.

En conclusión, tras realizar pruebas en la señal con las distintas configuraciones se puede afirmar que aparentemente los mejores resultados se obtienen por un lado, aumentando la precisión, y por otro utilizando el método con el que se consiga la mejor reconstrucción de la señal original.

<i>Filtrado</i>	<i>Precisión</i>	<i>Tiempo (s)</i>	<i>Threshold (nA)</i>		<i>Nº de detecciones</i>	<i>Detect. (%)</i>
<i>'None'</i>	1	94.93	$2\cdot\sigma$	0.003669	440	-
			$3\cdot\sigma$	0.005509	374	-
	2	129	$2\cdot\sigma$	0.002958	644	92.19%
			$3\cdot\sigma$	0.004442	535	-
<i>'Moving'</i>	1	66.6	$2\cdot\sigma$	0.002037	619	92.51%
			$3\cdot\sigma$	0.003056	554	-
	2	200.26	$2\cdot\sigma$	0.002487	636	93.49%
			$3\cdot\sigma$	0.003731	544	-
<i>'Lowess'</i>	1	112.42	$2\cdot\sigma$	0.00303	545	-
			$3\cdot\sigma$	0.00454	475	-
	2	140.59	$2\cdot\sigma$	0.002077	679	-
			$3\cdot\sigma$	0.003115	588	91.37%
<i>'Loess'</i>	1	85.68	$2\cdot\sigma$	0.002914	519	-
			$3\cdot\sigma$	0.004371	475	-
	2	112.85	$2\cdot\sigma$	0.003617	576	-
			$3\cdot\sigma$	0.005426	499	-
<i>'Sgolay'</i>	1	98.79	$2\cdot\sigma$	0.002767	531	-
			$3\cdot\sigma$	0.004150	471	-
	2	128.01	$2\cdot\sigma$	0.003628	583	-
			$3\cdot\sigma$	0.005442	501	-

Tabla 11. Recopilación de los resultados obtenidos con los distintos métodos.

3.2 Comparativa *PS_Analysis* con la detección visual y *Spike2*

Para contrastar la fiabilidad de los resultados obtenidos con cada configuración se han comparado visualmente uno por uno todos los eventos detectados en cada una de ellas con el registro etiquetado por los electrofisiólogos. También se han comparado las detecciones obtenidas con *Spike2*. En la figura 3.13 se muestra cómo se ha realizado la comparativa.

En la gráfica **A** se representan con marcas los instantes de inicio de los eventos detectados con *Spike2*, en **B** utilizando el método '*Lowess*', en **C** y **D** el filtrado '*Moving*' con distintas precisiones, en **E** sin filtrado, '*None*', en **F** se representan las marcas realizadas por los electrofisiólogos visualmente y en **G** el registro original.

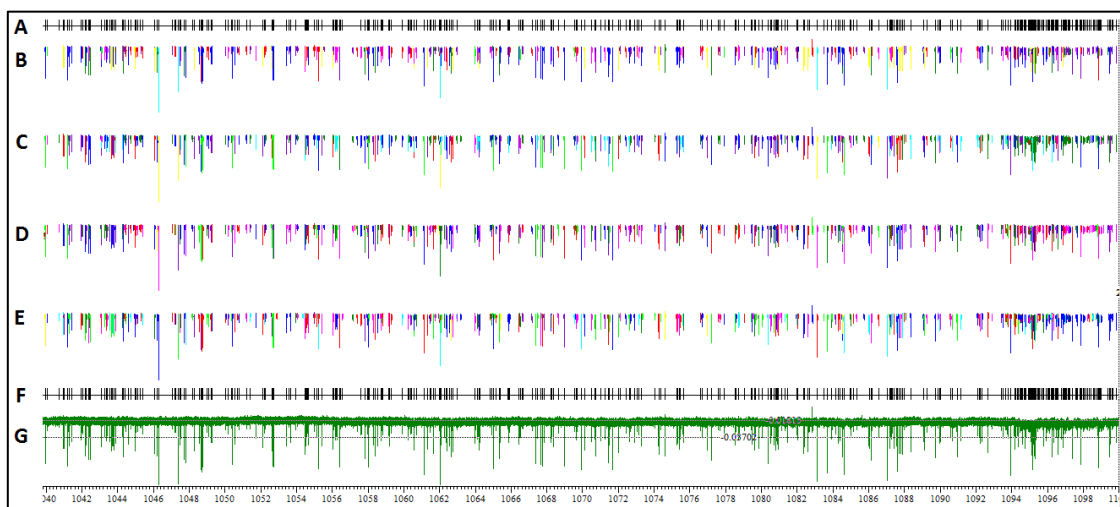


Figura 3.13. Análisis de las distintas técnicas de detección.

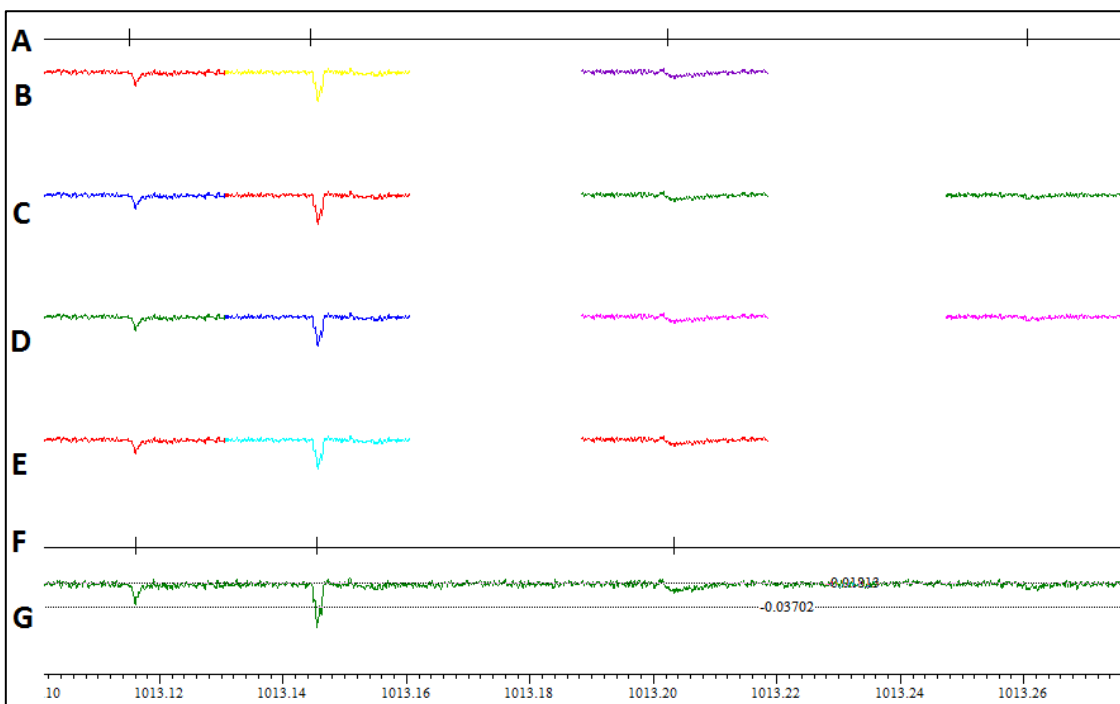


Figura 3.14. Detecciones correctas y falsos positivos en las gráficas A, C y D.

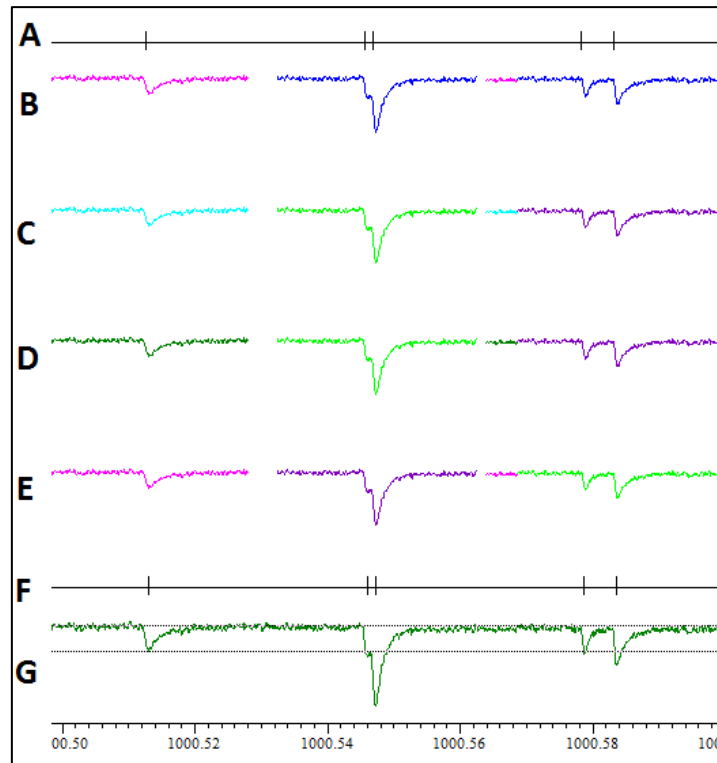


Figura 3.15. Detecciones correctas salvo en los eventos solapados.

En las figuras 3.14 y 3.15 se muestran distintas ampliaciones de la figura 3.13 que sirven para hacernos una idea del análisis realizado.

Tras analizar estos resultados se ha obtenido una tasa de falsos positivos y de detecciones perdidas de cada método, en comparación con el etiquetado visual de los electrofisiólogos que sirve de base. En la tabla 12 se presentan los resultados obtenidos.

<i>Gráfica</i>	<i>Método</i>	<i>Precisión</i>	<i>Threshold</i>	<i>Nº de detecciones</i>	<i>Falsos positivos (%)</i>	<i>Detecciones correctas (%)</i>
A	Detección <i>Spike2</i>	-	-	695	15.9%	94.8%
E	'None'	2	$2\cdot\sigma$	644	9.1%	92.19%
D	'Moving'	1	$2\cdot\sigma$	619	6.46%	92.51%
C		2	$3\cdot\sigma$	636	7.3%	93.49%
B	'Lowess'	2	$3\cdot\sigma$	588	3.57%	91.37%

Tabla 12. Número total de detecciones, tasa de FP y porcentaje de detecciones de los métodos evaluados visualmente.

Los resultados se han evaluado con los electrofisiólogos a los que va destinada la aplicación. Se ha concluido que es preferible tener una tasa de falsos positivos baja antes que un gran porcentaje de detección de los eventos. Las distintas configuraciones de *PS_Analysis* evaluadas tienen mejor tasa de FP que la detección con *Spike2*, aunque realizan menos detecciones.

Por otro lado, los falsos positivos de nuestra herramienta pueden ser eliminados a posteriori con alguna de las funcionalidades de *Spike2*. Cabe destacar que la gran mayoría de los eventos no detectados por *PS_Analysis* se tratan de eventos solapados o muy próximos con otros. Esto podría solucionarse modificando algún parámetro del algoritmo de detección que utiliza la herramienta.

En conclusión, para el registro “2-11-10-cel1-4” analizado en este capítulo, las distintas configuraciones de *PS_Analysis* utilizadas consiguen unos resultados razonables para el trabajo de los electrofisiólogos. Más concretamente, con la configuración del filtrado ‘*Lowess*’ empleada se obtiene un porcentaje de detección aceptable (91.37%) y una tasa de falsos positivos muy baja (3.57%). A continuación, en las figuras 3.16 y 3.17, se muestran las capturas de la herramienta utilizando esta configuración.

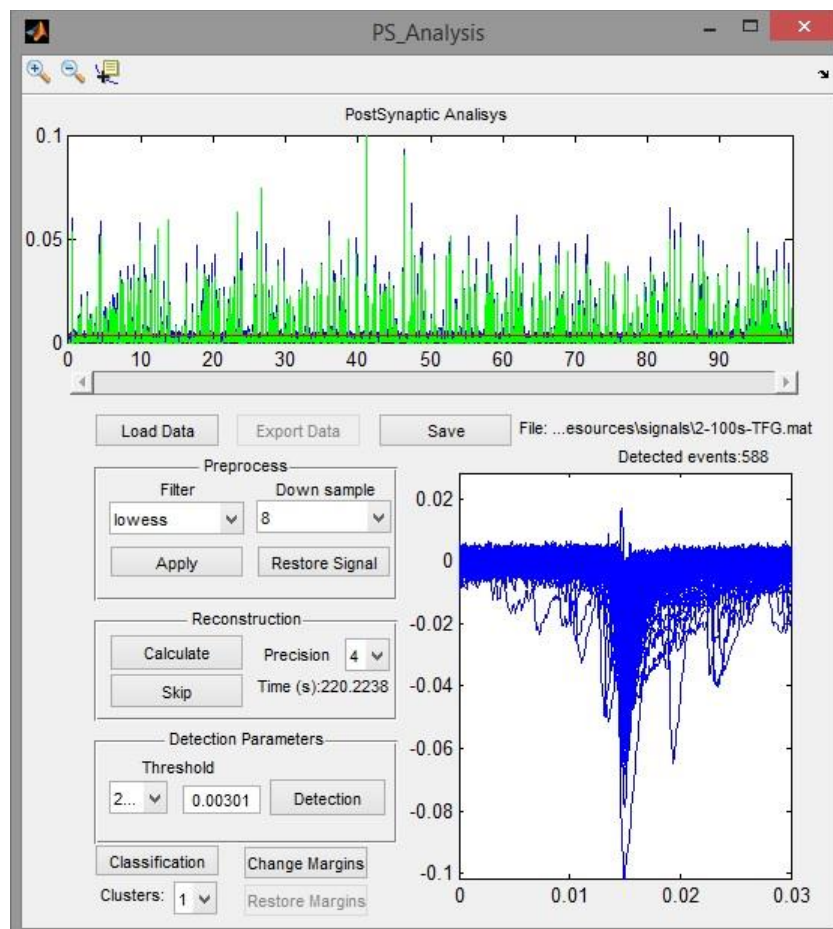


Figura 3.16. Detección de *PS_Analysis* con filtrado ‘*Lowess*’ y nivel 2 de precisión.

En la figura 3.16 se puede observar que la reconstrucción calculada se ajusta bastante bien a la señal, un buen ajuste de la reconstrucción mejora la detección de eventos. En la gráfica inferior de esta figura se representan todos los eventos detectados.

Finalmente, tras la detección se realiza la clasificación (Figura 3.17) de los eventos en nueve clases para poder observar mejor la distribución de los eventos. La complejidad para clasificar los eventos de este tipo de registros radica en el amplio rango de amplitudes que podemos encontrar, además las cinéticas de estos eventos no son parámetros excesivamente críticos para determinar una clasificación.

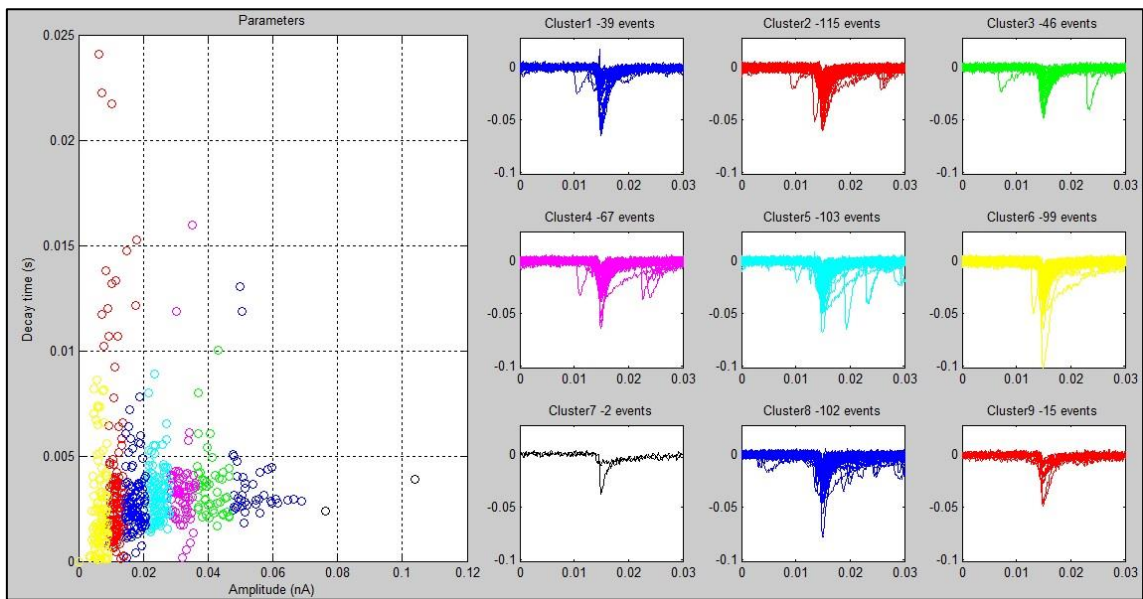


Figura 3.17. Clasificación de los eventos postsinápticos detectados.

3.3 Histogramas de los eventos detectados

Para el trabajo de los electrofisiólogos es útil representar en histogramas los resultados obtenidos. Es una forma gráfica de interpretar la distribución de las características de los eventos. A continuación se muestran los histogramas de amplitud y de la cinética (tiempo de subida + tiempo de bajada) de los eventos detectados con la configuración 'Lowess' citada en el punto anterior. Los datos se han extraído de *PS_Analysis* con la opción de guardado ("*Save*") y se encuentran en el anexo.

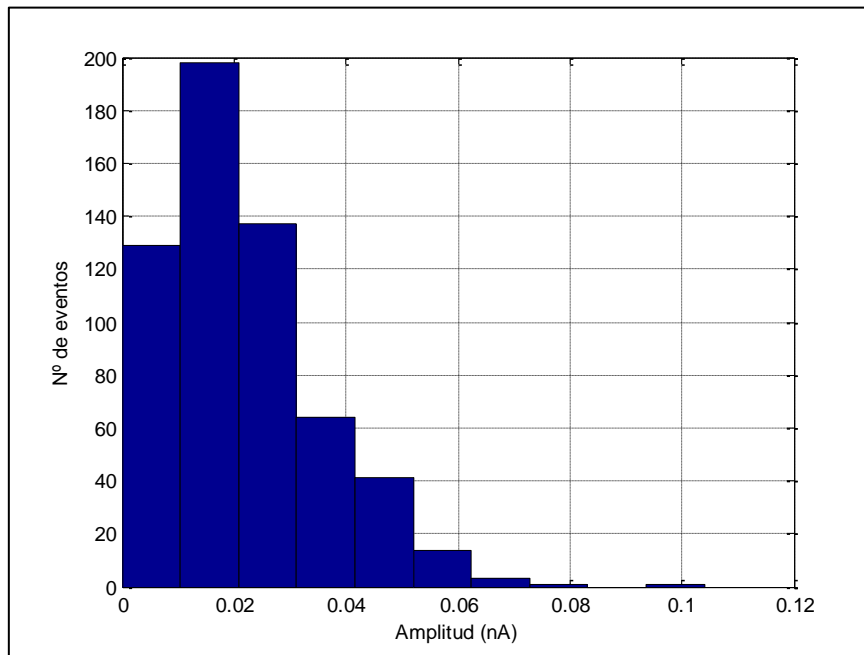


Figura 3.18. Histograma de amplitud, en valor absoluto, de los eventos mEPSC detectados.

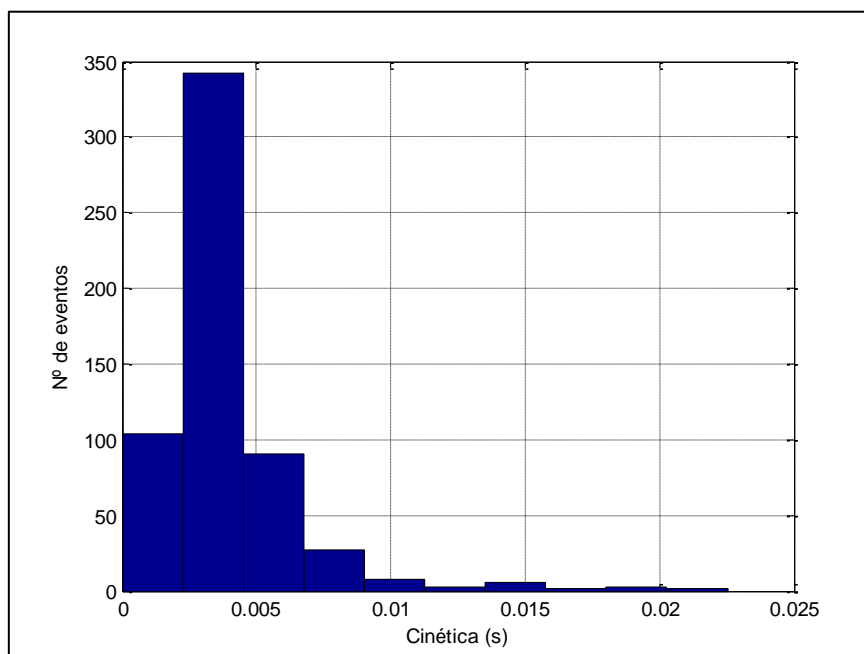


Figura 3.19. Histograma de tiempos de duración de los eventos mEPSC detectados.

Capítulo 4

Conclusiones y trabajos futuros

Por último, en este capítulo se presentan las conclusiones del proyecto, así como diversas líneas de trabajo futuro sobre las cuales se puede continuar con la investigación llevada a cabo hasta el momento.

En este trabajo se ha desarrollado una aplicación, *PS_Analysis*, capaz de detectar y clasificar eventos postsinápticos de forma rápida y precisa. Para determinar con qué algoritmos implementar el programa se ha hecho un estudio teórico de las técnicas más utilizadas en la actualidad, con el objetivo de encontrar algún algoritmo desarrollado en Matlab. Finalmente se utilizaron las herramientas *fade-master* y *detectPSPs* que combinadas se emplean en la identificación de eventos postsinápticos.

La aplicación implementada está destinada a los electrofisiólogos de la Universidad de Alcalá. Este proyecto trata de simplificar el trabajo de los técnicos ya que el análisis de registros intracelulares se realiza de forma manual y este proceso es largo y tedioso.

La dificultad radica en la naturaleza de los registros que se pretenden estudiar. Como se ha podido observar son señales con bastante ruido de fondo y multitud de eventos con un amplio rango de amplitudes, algunos tan pequeños que se confunden con el ruido y otros muy próximos entre sí. Esto conlleva fallos en la detección. Por eso cuesta analizar estos registros, más aún de forma visual, y no siempre los resultados son correctos.

Durante el desarrollo de la aplicación se trabajó estrechamente con los electrofisiólogos y se utilizaron sus registros para adaptar la herramienta a sus condiciones y necesidades.

Finalmente, tras evaluar el programa utilizando registros etiquetados por los expertos, se puede concluir que *PS_Analysis* puede servir de gran ayuda en el trabajo de laboratorio ya que aporta rapidez y cierta fiabilidad en los análisis.

Líneas de trabajo futuro

En este apartado se comentan diversas líneas de trabajo a seguir para continuar con la mejora de sistemas de análisis de registros intracelulares:

- Mejorar la aplicación desarrollada en alguna de las siguientes líneas:
 - Conseguir un análisis más automático modificando los bloques de preprocesamiento y detección.
 - Conseguir más precisión en el análisis mejorando los algoritmos de reconstrucción y detección.
 - Tratar de extraer más parámetros de los eventos detectados para poder utilizar un algoritmo de clasificación automático
- Adaptar la herramienta para analizar registros con diferentes tipos de eventos (excitadores e inhibidores).
- Mejorar el rendimiento de *PS_Analysis* ante registros con actividad espontánea que conllevan un mayor número de eventos y la posibilidad de que los eventos se solapen o se sumen.
- Añadir un sistema que permita a la aplicación realizar análisis en tiempo real.
- Combinar funcionalidades de *Spike2* (procesamiento, detección, umbralización, clasificación) con la herramienta.

Pliego de condiciones

En este capítulo se describen los materiales y programas que han sido utilizados para la realización de este proyecto, así como los requisitos software y hardware para la utilización del programa *PS_Analysis*, desarrollado en este trabajo.

1. Material empleado en la realización del proyecto

Para la realización de este proyecto se han empleado los siguientes recursos:

- Hardware:
 - Ordenador portátil Toshiba Satellite L50-A-18R con las siguientes especificaciones:
 - Procesador Intel Core i7 4700MQ a 2.4 GHz.
 - Memoria RAM de 4 GB a 1600 MHz con tecnología ddr3.
 - Tarjeta gráfica GeForce GT 740M de 2 GB.
 - Disco duro de 750 GB a 5400 rpm con interfaz serial ata.
- Software:
 - Sistema operativo Windows 8.1 Home Premium de 64 bits.
 - *Spike2* versión 7.00 (CED, Cambridge, Reino Unido).
 - Matlab R2012a.
 - Signal Processing Toolbox de Matlab.
 - Fade-master (Fast automated blind deconvolution of EPSCs).
 - detectPSPs (Postsynaptic Potential Detector).
 - Procesador de textos Microsoft Office Professional Plus 2010.

2. Material necesario para la ejecución de *PS_Analysis*

Para la ejecución del programa son necesarios los siguientes recursos:

- Hardware:
 - El programa se ha ejecutado con éxito empleando el ordenador portátil utilizado para la realización del proyecto, cuyas especificaciones se han descrito en la sección anterior. Para mejorar la ejecución del programa se recomienda utilizar una memoria RAM superior.
- Software:
 - Sistema operativo Windows 8.1.
 - Matlab 2012a.

Presupuesto

Para completar la memoria se presenta el presupuesto total del proyecto, donde se incluyen las herramientas utilizadas, tanto hardware como software, y la mano de obra. A continuación se desglosan dichos costes:

a) Costes materiales

Concepto	Unidades	Precio Unitario	IVA (21%)	Total
Ordenador Toshiba Satellite	1	578.5 €	121.5 €	700 €
Matlab R2012a	1	2000 €	420 €	2420€
Signal Processing Toolbox (Matlab)	1	1000 €	210 €	1210 €
Spike2 v7.00	1	1936.26 €	406.61 €	2342.87 €
			Subtotal	6672.87€

b) Costes de desarrollo

En este caso, se ha considerado adecuado fijar un coste de 50 €/hora para las actividades de investigación, evaluación de algoritmos y desarrollo de aplicaciones; y de 20 €/hora para la redacción de documentación.

Concepto	Número de horas	Precio Unitario	Total
Estudio y análisis del problema	50	50 €/h	2500 €
Investigación y evaluación	150	50 €/h	7500 €
Programación	200	50 €/h	10000 €
Pruebas y ajustes	20	50 €/h	1000 €
Redacción	200	20 €/h	4000 €
		Subtotal	25000 €

En total el proyecto tendría un coste total de 31672.87 €.

Manual de usuario de *PS_Analysis*

Introducción

PS_Analysis (*Post Synaptic Analysis*) es una aplicación desarrollada para su ejecución en Matlab. Ésta permite la utilización de archivos con extensión *.mat exportados desde *Spike2* para llevar a cabo el análisis de registros intracelulares utilizando además las herramientas implementadas en Matlab:

- ***fade-master***: disponible públicamente en <https://github.com/andorardo/fade>.
- ***detectPSPs***: descargable desde el repositorio de Matlab <http://www.mathworks.com/matlabcentral/fileexchange/19380-postsynaptic-potential-detector>.

Instalación

Para la instalación del programa se debe guardar la carpeta *PS_Analysis* en la localización deseada. Después, hay que abrir Matlab y seleccionar en la ventana “*Current Folder*” la localización de la carpeta del programa que hemos guardado. El programa ya está instalado. Para ejecutarlo, debe teclearse *PS_Analysis* en la ventana de comandos de Matlab y pulsar la tecla “**Enter**”.

Por otro lado, en la carpeta se incluye un archivo llamado “*ImpWM2SMR.s2s*” que contiene un programa desarrollado por la empresa *Cambridge Electronic Design Ltd*. El programa permite importar el archivo *.txt con el resultado generado por *PS_Analysis* a un canal de tipo en *Spike2*. Para poder utilizar el programa cómodamente, es recomendable incluirlo en la barra de programas de *Spike2*. Para ello, debe seleccionarse en el menú Script de la ventana principal de *Spike2* la opción “**Script Bar List...**”. Después se pulsa el botón “**Add...**” y se selecciona el archivo “*ImpWM2SMR.s2s*”. Finalmente, se pulsa “**Ok**” y aparecerá un nuevo botón en la barra de programas de *Spike2* que habrá que pulsar cada vez que el usuario desee importar el resultado de *PS_Analysis*.

Exportación de datos desde *Spike2*

PS_Analysis utiliza datos de Matlab. Para poder utilizar datos procedentes de *Spike2*, es necesario exportarlos antes en formato *.mat, formato de archivos de Matlab.

Para exportar un registro desde *Spike2*, se debe abrir el archivo con la señal a analizar. Si sólo se va a analizar parte del canal, es recomendable situar dos cursores que delimiten la zona bajo estudio. Para llevar a cabo la exportación, se debe seleccionar en el menú “**File**” la opción “**Export As...**”. En la nueva ventana que aparece, se selecciona la carpeta en la que se desea guardar el nuevo archivo y se le asigna un nombre. En el campo “**Tipo**”, debe seleccionarse la opción *MATLAB data (*.mat)*. Una vez hecho esto, aparece una nueva

ventana. En el campo *Channel or channel list* se debe seleccionar un único canal, que es el que se quiere analizar. En los campos "*Time range*" y "*To*", se puede elegir *0.0* y *MaxTime()* si se desea exportar el canal completo o bien si se ha marcado un fragmento del registro con dos cursores 1 y 2, se debe seleccionar *Cursor(1)* y *Cursor(2)*. Es importante asegurarse de que el campo "*Time shift data so that first exported range starts at 0*" no está marcado. En caso contrario, cuando se importe el resultado de vuelta a Spike2, los eventos detectados y clasificados aparecerán desplazados en tiempo. Una vez hecho esto, se pulsa primero el botón "*Add*" y luego "*Export*".

Interfaz de usuario

a) Carga de datos

Al ejecutar el programa desde la línea de comandos de Matlab aparece la ventana principal donde sólo se encuentra activo el botón "*Load Data*". Al pulsar sobre el botón, se abre una nueva ventana en la que se debe seleccionar el archivo exportado a formato *.mat desde Spike2 que se quiere utilizar. Una vez cargado el archivo, se activa el botón "*Apply*" del bloque *Preprocess*. Además, en la gráfica superior aparece una representación de la señal cargada. En la gráfica sólo se muestra la señal en valor absoluto.

Tras la carga del registro la ventana del programa será similar a la figura 1.

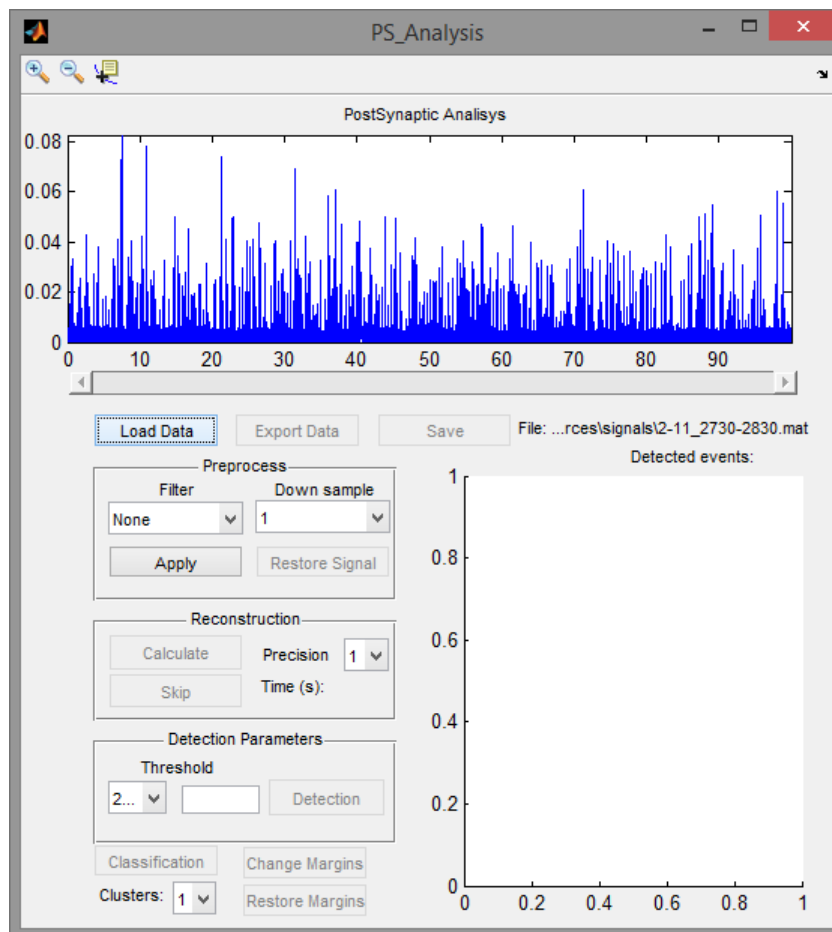


Figura 1. Ventana principal de la herramienta *PS_Analysis* con datos cargados.

b) Preprocesamiento de la señal

Una vez cargados los datos podemos filtrar o submuestrear la señal original. El filtrado nos permite suavizar los niveles de ruido del registro y al submuestrear se reduce el número de muestras de la señal. De cada n muestras que indiquemos nos quedaremos con una.

En el bloque *Preprocess* elegiremos el tipo de filtro y el número de muestras n a través de los menús desplegables. Cuando hayamos hecho una selección deberemos pulsar el botón **“Apply”** para que los cambios surtan efecto, así se representará en la gráfica superior la señal con los cambios que hemos aplicado. Cada vez que pulsamos el botón **“Apply”** los cambios seleccionados se aplican sobre la señal original, es decir, no podemos filtrar y/o muestrear dos veces.

En la figura 2 se pueden apreciar los cambios producidos al aplicar un filtrado **‘Moving’** y un submuestreo de 8 sobre la señal cargada en la figura 1.

Para revertir los cambios efectuados por el bloque *Preprocess* simplemente debemos pulsar el botón **“Restore Signal”** con el que recuperaremos la señal original que se volverá a representar en la gráfica superior de la ventana.

NOTA: En el preprocesamiento es recomendable utilizar una configuración que no distorsione demasiado los eventos de interés de la señal. Si no utilizamos ningún filtro habrá mucho ruido y si realizamos un submuestreo excesivo perderemos mucha información de la señal, aunque si no reducimos el número de muestras la herramienta tardará mucho en analizar la señal e incluso se bloqueará.

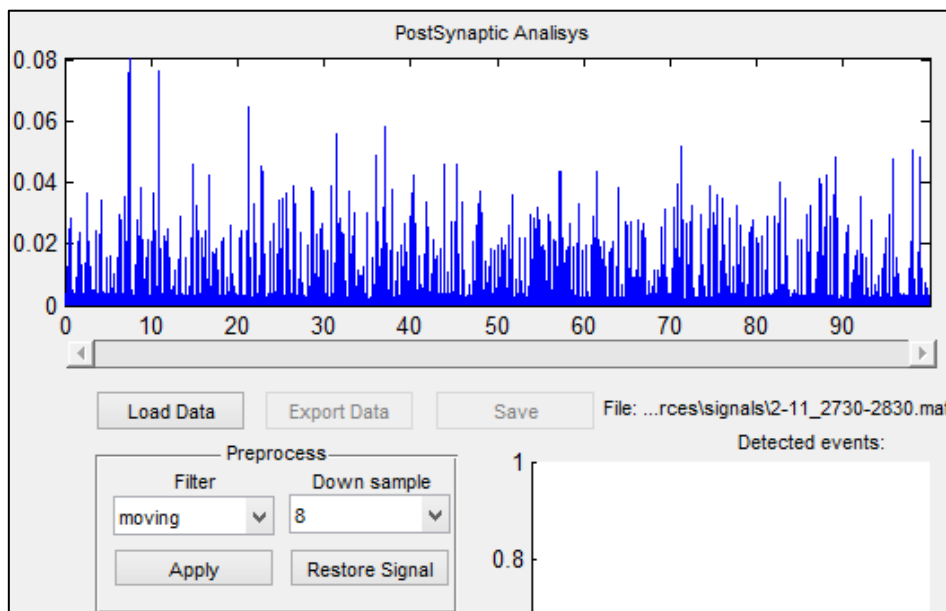


Figura 2. Señal procesada con un filtro **‘Moving’** y un submuestreo de 8.

c) Reconstrucción de la señal

Este paso no es imprescindible para el proceso de detección y clasificación pero sí altamente recomendable. Con el objetivo de mejorar el rendimiento en el proceso de detección, es necesario reconstruir, a partir de la señal preprocesada, una señal que sólo contenga eventos. Para ello debemos elegir un nivel de precisión del menú desplegable "**Precision**" y pulsar el botón "**Reconstrucción**", si no queremos realizar este paso podemos pulsar el botón "**Skip**" y el programa nos habilitará directamente las opciones siguientes sin realizar la reconstrucción.

El nivel de precisión influye en la calidad de la señal que se obtiene en la reconstrucción pero también en el tiempo que tarda en calcularla. Es recomendable utilizar un nivel dos o superior, aunque para señales con poco ruido con un nivel uno es suficiente. Para señales ruidosas la reconstrucción tardará más tiempo en calcularse.

El tiempo que tarda en calcularse la reconstrucción lo podemos ver en el apartado *Time (s)* del recuadro *Reconstruction*.

En la figura 3 podemos ver el resultado de la reconstrucción de la señal anterior.

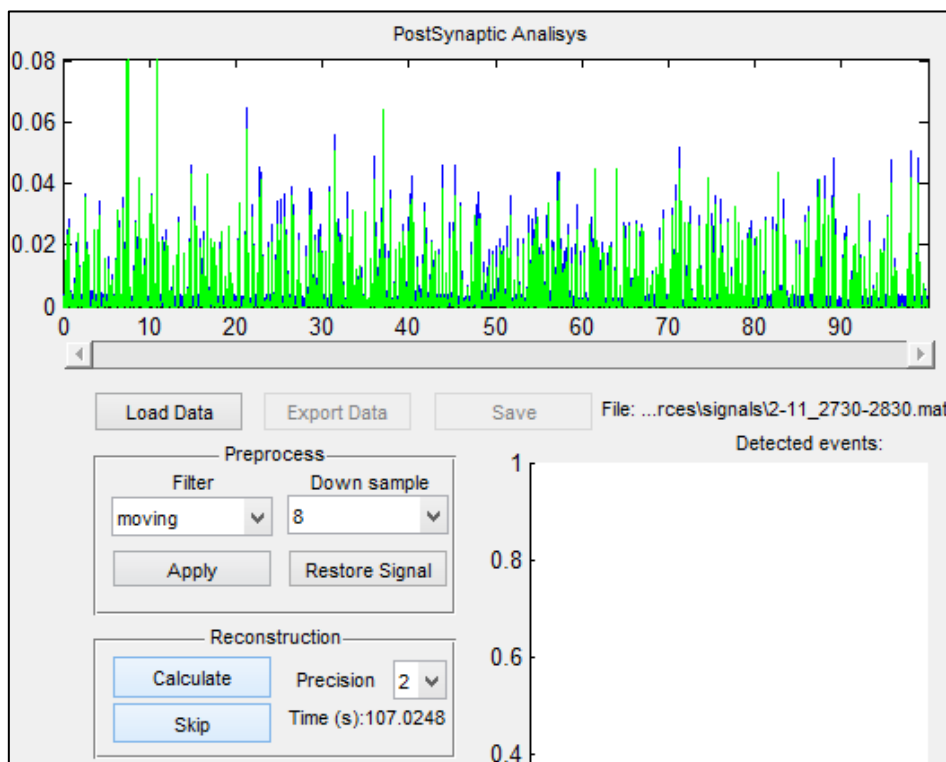


Figura 3. Señal procesada con un filtro '*Moving*' y un submuestreo de 8.

d) Detección de los eventos

El siguiente paso es realizar la detección de los eventos. Si se han realizado correctamente los dos pasos anteriores (*Preprocess* y *Reconstruction*) la detección será mejor.

Primero se debe seleccionar del menú desplegable "**Threshold**" el umbral que queremos utilizar. Podemos elegir $2\cdot\sigma$ o $3\cdot\sigma$ si queremos un umbral que dependa del nivel de ruido del registro (σ es la desviación estándar del ruido de fondo) o bien la opción "*Manual*" si queremos elegir el umbral desde la gráfica superior de la interfaz de usuario.

NOTA: Cuanto más pequeño sea el valor del umbral elegido más detecciones se realizarán, pero será más probable que se detecten falsos positivos. En cambio si el umbral es muy grande estaremos perdiendo pequeños eventos presentes en el registro. Por tanto hay que llegar a una solución intermedia, si los umbrales de σ no son adecuados es recomendable utilizar la opción manual.

Cuando hemos elegido el valor del umbral podemos pulsar el botón "**Detection**" para comenzar la búsqueda de eventos. En la figura 4 se muestra el resultado de la detección de eventos para la señal con la que estamos trabajando. En la gráfica inferior se representan superpuestos todos los eventos detectados y en el campo "*Detected events*" se muestra el número de detecciones.

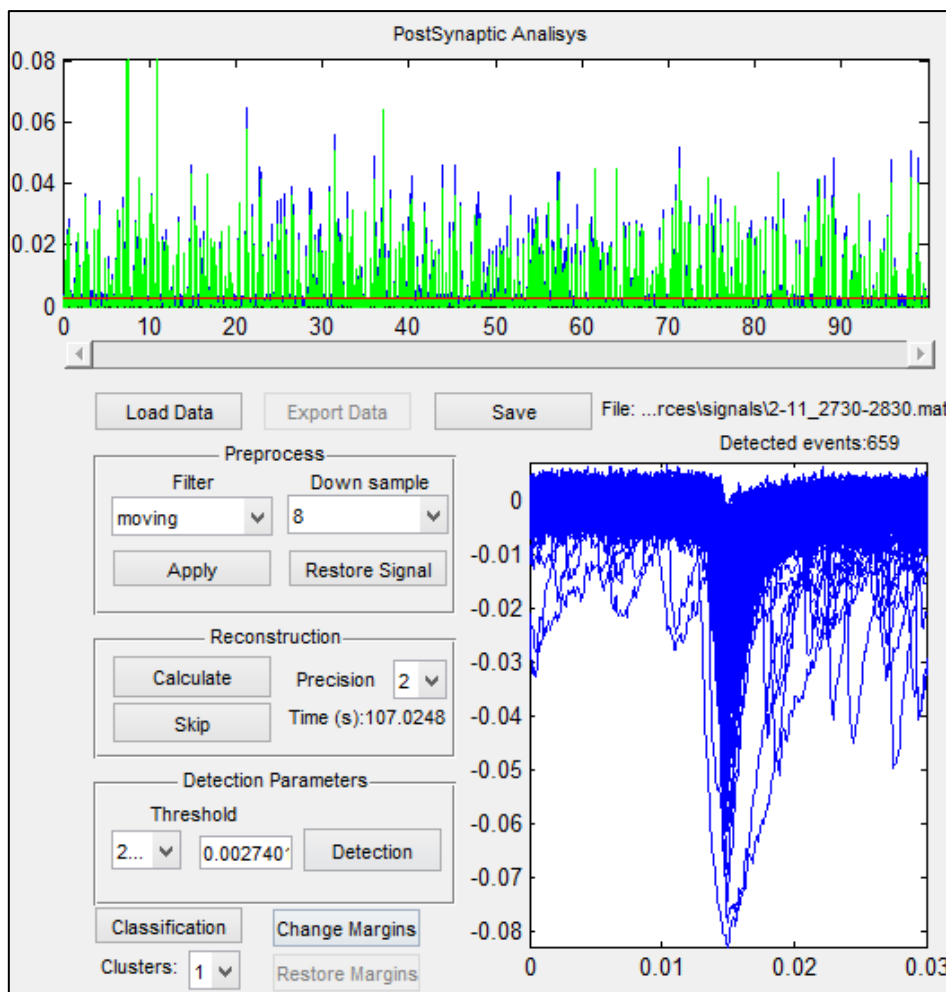


Figura 4. Detección de eventos con un umbral de $2\cdot\sigma$.

e) Clasificación

El último paso del análisis que realiza la herramienta es la clasificación de los eventos detectados. Esta clasificación se hace en base a los parámetros de amplitud y tiempos de subida o bajada de los eventos, siendo más utilizado el tiempo de bajada por ser más crítico.

Para la clasificación se utiliza un algoritmo que busca agrupar cada evento según sus parámetros, pero antes se debe indicar el número de grupos en los que queremos que se clasifique. Utilizaremos el menú desplegable “*Clusters*” para indicarlo y después pulsaremos el botón “*Classification*” para ejecutar el algoritmo.

Cuando termina la clasificación aparece una nueva ventana con los diferentes grupos en los que se ha clasificado y una gráfica con la distribución de los parámetros de cada evento. En la figura 5 podemos ver el resultado cuando elegimos 9 grupos.

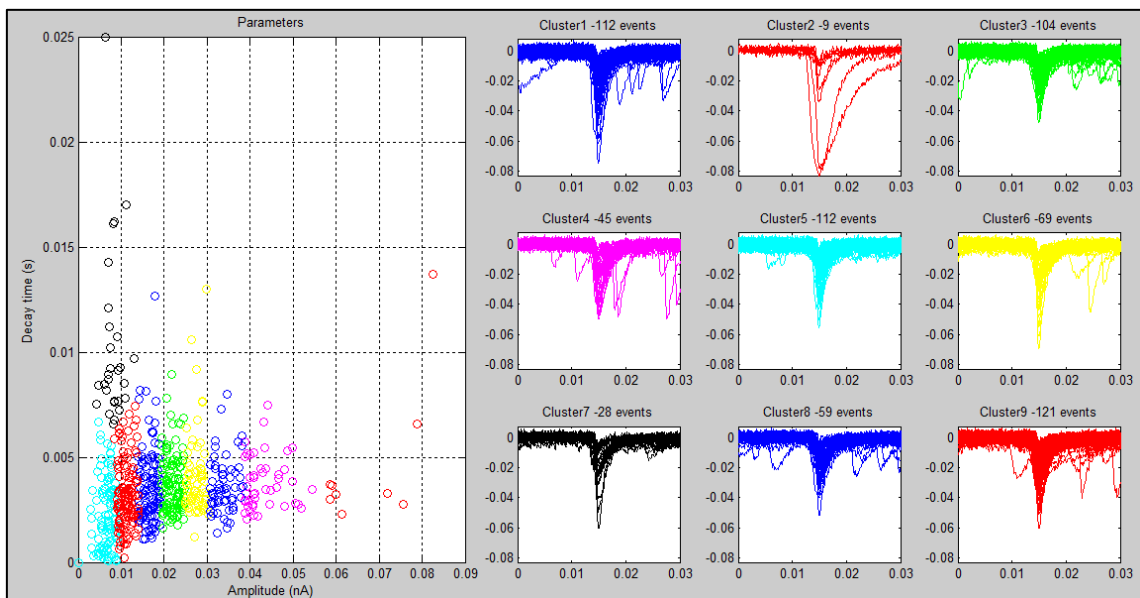


Figura 5. Clasificación de los eventos detectados a partir de los parámetros de amplitud y tiempo de caída.

Una vez terminado el proceso de clasificación se pueden guardar resultados obtenidos durante el análisis de la señal. Para ello se activan los botones “*Export*” y “*Save*”

e) Guardado de los resultados

La aplicación permite dos opciones de guardado tras completar el análisis de la señal:

- **“Save”**: Pulsando este botón podremos guardar en una tabla las características extraídas de cada evento en el bloque *Detection*. Aparecerá una ventana donde podremos elegir el nombre y la carpeta de destino donde se guardará el fichero *.xls. La tabla que se genera tiene una fila por cada evento detectado y cada columna se corresponde con un parámetro diferente siendo el orden: Amplitud, tiempo de bajada, tiempo de subida, valor de pico, instante de pico. Además, cuando se guarde la tabla, aparecerá otra ventana que nos permitirá guardar la variable *handle*, que maneja la aplicación en cada procesamiento, en un archivo *.mat. Esta variable contiene datos del análisis realizado y puede servir para retomar el trabajo con *PS_Analysis*.
- **“Export”**: Pulsando este botón habilitaremos la opción de exportar el resultado de nuestro análisis en un fichero *.txt para su posterior importación con *Spike2*. Aparecerá una ventana donde podremos seleccionar un nombre y una carpeta de destino para el archivo que se va a generar con los resultados. Este archivo contendrá las muestras de todos los eventos y los grupos en los que han sido clasificados.

Importación del resultado de *PS_Analysis* a *Spike2*

Para importar en *Spike2* el archivo *.txt que se genera como resultado al pulsar el botón **“Export”** en la aplicación *PS_Analysis*, es necesario seleccionar el programa *ImpWM2SMR* de la barra de programas de *Spike2*. En primer lugar, el programa solicita el archivo *.smr al que se desea importar el resultado (es recomendable que sea el mismo archivo que se utilizó para extraer la señal analizada). Después debe indicarse el archivo *.txt con el resultado de la clasificación. Una vez hecho esto, el programa crea un canal con los eventos clasificados, asignando a cada grupo un código y un color.

Bibliografía

[1] Dirección web:

<http://www.ncbi.nlm.nih.gov/books/NBK10841/box/A229/>

[2] Dirección web:

<http://jn.physiology.org/content/107/1/29>

[3] F. F. Hiltz (1965), "Computer correlation analysis of intracellular neuronal responses". *Applied Physics Laboratory, The Johns Hopkins University. Silver Spring, Maryland.*

[4] Soltesz, I., D. K. Smetters, and I. Mody. 1995. Tonic inhibition originates from synapses close to the soma. *Neuron*. 14:1273–1283.

[5] Alejandro Javier Pernía-Andrade, Sarit Pati Goswami, Yvonne Stickler, Ulrich Fröbe, Alois Schlögl, and Peter Jonas (2012). "A Deconvolution-Based Method with High Sensitivity and Temporal Resolution for Detection of Spontaneous Synaptic Currents In Vitro and In Vivo". *Biophysical Journal*, Volume 103, pp. 1429–1439.

[6] J. D. Clements and J. M. Bekkers (1997), "Detection of Spontaneous Synaptic Events with an Optimally Scaled Template". *Biophysical Journal*, vol. 73, pp. 220-229.

[7] Guo-Hua Li, Michael F. Jackson, John F. MacDonald (2007). "Weighted least squares fitting with multiple templates for detection of small spontaneous signals". *Journal of Neuroscience Methods*, volume 164, pp. 139–148.

[8] Suguru N. Kudoh, Takahisa Taguchi (2002). "A simple exploratory algorithm for the accurate and fast detection of spontaneous synaptic events". *Biosensors and Bioelectronics*, vol. 17, pp. 773-782.

[9] Daniel Andor-Ardo, Erica C. Keen, A. J. Hudspeth, Marcelo O. Magnasco (2012). "Fast, Automated Implementation of Temporally Precise Blind Deconvolution of Multiphasic Excitatory Postsynaptic Currents".

[10] Dirección web:

<http://www.inf.fu-berlin.de/lehre/WS05/Mustererkennung/infomax/infomax.pdf>

[11] Dirección web:

<http://www-mtl.mit.edu/Courses/6.050/2003/notes/chapter10.pdf>

[12] Manual de usuario Spike2 for Windows Version 7. Disponible en:

<http://www.ced.co.uk/img/Spike7.pdf>.

[13] Trabajo Fin de Carrera: Detección y clasificación de espigas en registros extracelulares de múltiples neuronas. Autor: Alberto González Egea (2013).

[14] Documentación de Matlab. Disponible en:

<http://www.mathworks.es/es/help/curvefit/smooth.html?refresh=true>

[15] Dirección web:

<https://github.com/andorardo/fade>

[16] Dirección web:

<http://www.mathworks.com/matlabcentral/fileexchange/19380-postsynaptic-potential-detector/content/detectPSPs.m>

[17] Transparencias de Sistemas de Visión Artificial: *“Tema 4. Técnicas de reconocimiento (3a parte)”*. Sira E. Palazuelos Cagigas, Luis M. Bergasa Pascual (2014). Universidad de Alcalá de Henares. Departamento de electrónica.

Anexo

Datos empleados en los histogramas del punto 3.3 del capítulo de resultados, pertenecientes a los eventos extraídos con *PS_Analysis* del registro “2-11-10-cel1-4”, del fragmento [1000-1100] (s).

Amplitud (pA)	Tiempo de bajada (ms)	Tiempo de subida (ms)	Instante de pico (s)
18,55463446	3,583388094	0,619198822	1000,51306
59,81437632	2,775017476	1,620904686	1000,54738
19,53118599	0,789601405	0,32681684	1000,57882
23,92573919	4,013487169	0,249780278	1000,58362
6,103464419	2,766898908	0,275996669	1000,69154
13,42767037	1,868966831	0,781110053	1001,40122
14,89252143	2,620069607	0,389066183	1001,57114
21,7284571	2,550407812	0,268532606	1001,75426
16,35737067	1,411246594	0,567199146	1001,7693
11,23041023	4,006549902	0,832285233	1001,8125
25,1464222	1,955320493	0,441110423	1002,65938
14,64837166	2,106748331	0,510474218	1002,75746
16,84565192	4,528831945	0,452597929	1003,0077
11,71868782	2,486819489	0,755197819	1003,28018
11,71870977	1,915315888	0,209332197	1003,5877
16,11320627	3,281218938	0,945064175	1003,74154
7,324167544	1,685378694	0,575999341	1003,8653
26,6112696	3,996734648	0,802665112	1004,20162
53,71088995	2,935129869	0,311466468	1004,34194
59,32611335	2,876178698	0,377325364	1004,37634
11,71869696	0,874517885	0,31563541	1004,45482
10,49800298	2,277374742	0,676665498	1004,46082
17,82220162	2,677577414	0,807198244	1004,91162
18,31049201	2,577570721	0,74526282	1005,10194
14,4041652	3,219098991	3,856665066	1005,23906
10,98626229	1,160079378	0,81666579	1005,48706
8,544876155	2,75843299	0,67599749	1005,6097
21,97259407	4,748708606	0,639999693	1005,64178
18,06599477	6,770029939	3,476791735	1005,84906
7,568300853	0,984057909	0,413864511	1005,90594
14,64838812	1,755501573	0,828051379	1006,01562
9,277257796	1,680115797	1,659196325	1006,06778
21,97260687	1,877453273	0,285881724	1006,16794
10,00971625	2,764889151	0,437331722	1006,2185
24,41401129	2,373733626	0,272803847	1006,32866

12,45111152	4,001158458	0,672632607	1006,5473
26,61127143	4,200158344	0,525961344	1006,64442
34,66791206	1,991343552	0,404284979	1007,12506
38,08588629	2,085496035	0,413466411	1007,16578
35,15619696	3,049344957	0,452869113	1007,2785
16,35736701	3,544339496	0,250998575	1007,35282
31,00579537	3,846780777	0,60489924	1007,55674
26,8554212	2,010716876	0,349562698	1007,58098
11,23040108	3,047271234	1,391120367	1007,8481
38,81830085	3,859208419	0,392354899	1008,16066
20,75190009	4,017823128	0,665332848	1008,33642
29,78509773	2,064290624	0,291377435	1008,57474
13,67181648	4,046532861	0,972888261	1008,72066
33,93549384	5,708401532	0,288799681	1008,92522
13,42766488	3,243457563	0,734101517	1009,02386
37,10931831	2,486849488	0,410899967	1009,06426
14,89251595	3,062117808	0,544397604	1009,3909
17,82220345	2,319811523	0,876189896	1009,48882
36,865185	3,02411342	0,38833285	1009,60986
16,11321358	5,813382248	0,918397268	1009,67338
25,14643317	1,873422664	0,469718797	1009,7661
61,0351215	2,939677116	0,332841997	1009,7705
14,40424384	2,586117595	0,221558094	1009,85258
11,71869696	1,317744993	0,418741837	1010,19434
10,74213446	0,902904996	0,773332504	1010,30314
44,67768317	2,262444904	0,31582561	1010,45922
33,69135687	2,187622971	0,39365029	1010,69018
10,74213812	2,215239213	0,261998627	1010,84378
7,812441478	1,215444766	0,35733024	1011,17122
28,32025764	2,174425861	0,376725423	1011,31698
33,93549018	3,021131504	0,542132151	1011,46106
45,89838081	2,281874699	0,233872078	1011,96474
11,23041571	2,912055462	0,972083428	1011,97738
47,60732678	5,078716362	2,263570374	1012,35106
22,46088446	1,560939313	0,381947454	1012,81706
13,42768134	1,551049599	0,407108819	1012,9425
14,89252143	1,094325562	0,428554012	1013,1161
34,42377143	0,882480955	0,631953394	1013,14554
7,81242319	8,150479644	1,594970159	1013,20338
17,57803356	0,417299955	2,743997193	1013,53106
9,521424025	2,235347832	0,240331165	1013,75562
21,24016671	3,698036313	0,906474333	1013,80714
35,15618782	15,97925564	0,607732653	1013,83762
22,46088264	3,579299359	0,292524709	1013,99922
11,47454354	1,525718637	1,659427838	1014,85282

3,417917544	0,153335985	0,150855729	1014,86946
9,765573794	4,576053055	0,505141494	1014,8765
8,544859696	0,817830903	1,428568523	1015,56818
7,568295367	8,104162103	0,835997371	1015,65082
11,23040291	2,730717341	0,57636093	1015,99906
35,6444709	3,720809246	0,32972005	1016,24714
10,25385321	0,916243741	0,796798719	1016,36138
6,835868006	1,207433913	1,040443315	1017,12818
20,26361884	3,91703492	0,364144333	1017,13898
8,789014951	4,037589429	0,407199091	1017,77362
48,09564277	3,136893613	0,448483453	1017,83306
14,64836069	4,498053197	1,319998799	1018,04498
13,67179819	2,797809285	0,649331219	1018,31954
13,4276722	0,210136347	0,534919332	1018,33938
17,57806465	2,122117851	0,329399169	1018,35082
26,12298835	1,925049535	0,395036741	1018,53826
40,28315375	3,658605734	0,429149787	1018,70426
17,57798967	1,276575719	4,138666246	1018,72466
7,080006802	1,350567746	0,237701082	1018,73402
23,68158027	2,705135757	0,341331791	1018,73994
22,94916571	1,687257893	0,3127268	1018,9105
31,00580634	2,923897512	0,392472162	1018,96106
9,033140946	0,545146987	0,339042705	1018,97682
34,17963264	3,520134967	0,625713585	1019,23762
19,77533393	1,708905434	0,790399106	1019,3081
21,72845162	7,973112147	0,874998527	1019,3289
16,84564643	2,528896609	0,668953964	1019,4105
31,00581183	2,211158031	0,278486763	1019,42866
4,394465413	2,842204953	0,815994427	1019,64642
15,62494514	2,121235979	0,614399077	1019,7445
40,5272962	2,702232007	0,419663549	1019,8421
43,21282662	3,316094182	0,383999368	1020,23322
5,371051688	6,731191045	0,271465189	1020,36538
25,63470345	3,059126405	0,455237259	1020,48658
23,68158759	2,237574695	0,614656727	1020,6833
19,28705817	1,956914105	0,265338708	1020,73442
24,41400032	3,200488194	0,584934021	1020,80786
23,43744696	2,513449626	0,287132553	1021,44298
26,85541937	1,467505822	0,301521075	1021,54074
10,9862696	2,712221876	0,510767838	1021,86178
17,57806282	3,084806125	0,895997363	1022,17746
20,75188546	4,342915807	0,553329019	1022,1873
26,61126595	3,506425986	0,826764182	1022,31506
23,19329537	2,995585175	0,269443565	1022,36058
15,13666754	1,906033931	0,577162924	1022,55906

4,394434324	4,01449044	0,203195978	1022,61098
18,79875863	3,542297818	0,891065568	1022,71602
32,95892402	2,464489952	0,509777363	1022,89418
43,21281564	10,03559668	0,905998391	1023,25194
19,77532845	4,881931946	0,535887373	1023,32754
10,25384407	1,972820756	0,274398697	1023,49714
39,3065821	2,824238774	0,417742912	1023,68514
8,300726386	1,812849354	1,00700903	1023,71634
44,67767402	4,21809048	0,473175759	1024,03338
8,544826777	0,134536677	1,075998284	1024,3613
8,300698954	3,069362811	1,233554036	1024,89578
8,789003978	2,393379723	0,338665065	1025,44962
28,07610238	3,815034065	0,784997066	1025,53458
32,47065009	3,126554783	0,312993034	1025,63306
31,98236884	2,773532287	0,384621686	1025,9057
53,95501777	3,63852079	0,326962007	1025,95786
49,80462166	13,04006075	0,800241978	1026,62946
24,41400398	2,325005988	0,324285116	1026,8237
30,76165108	3,682904457	0,919998429	1026,92138
6,103436987	1,326736749	0,614997169	1027,05946
11,47453988	0,430639771	1,081064883	1027,21522
48,82806465	2,631701313	0,285714004	1027,36842
27,83197821	3,405302775	0,325526238	1027,60874
11,2304084	3,25661392	0,495198443	1027,84394
15,13666206	2,10632463	0,698899644	1028,0025
25,63470162	4,443974467	0,738569846	1028,05722
11,71868599	1,162613095	0,705903076	1028,2537
18,3104847	1,568482331	0,82999874	1028,35986
17,33391671	3,04215817	1,019662949	1028,5009
20,75189095	2,050211772	0,421332428	1028,79698
46,38666389	4,025356905	0,513683671	1028,8997
14,64838446	6,076048261	0,41445302	1028,92906
6,591727381	2,43629245	1,268999145	1029,28242
50,0487696	2,919956842	0,336998343	1029,7805
13,67181831	0,682119505	0,320532174	1029,91738
23,19330634	2,014286558	0,341390691	1030,2289
19,77535039	3,396636553	0,408999867	1030,2341
11,47455634	1,179211543	0,727998592	1030,31914
22,70503058	4,159085909	0,343029866	1030,3509
17,57805002	12,15887303	1,600726841	1030,51018
8,789007636	1,228116731	0,215998757	1030,91634
19,28699965	3,105600492	2,314998992	1031,09474
8,544857867	13,77421085	0,680950485	1031,10234
16,35736335	1,352903379	1,286856328	1031,17122
5,615170367	1,546353718	0,416911446	1031,18234

28,56439643	2,957220663	0,351050707	1031,76698
9,033146432	1,836718258	0,709384107	1031,78938
12,93938363	2,967187094	0,754498372	1031,8277
25,14642585	3,932888731	0,548776543	1031,97506
39,30658759	2,176093012	0,301377249	1032,35882
10,49795361	3,590490647	0,314988126	1032,65434
28,07611518	2,239496483	0,290755065	1032,79418
32,22650946	3,261879401	0,492705121	1032,98666
12,93937266	5,232358022	0,925197042	1033,18434
24,16986152	2,843826609	0,783330779	1033,97962
25,14641122	3,243249063	0,481510233	1034,2925
6,347588584	0,100112953	1,907110981	1034,61418
23,43742319	3,443634359	1,117997906	1034,72658
11,71868782	1,763049075	0,807031382	1034,81682
5,126159426	0,167715416	0,10336867	1034,9685
39,7948597	5,393152353	0,495650964	1035,03954
9,521422196	2,209586235	0,316119282	1035,70578
20,50775215	2,386780812	0,725942365	1035,7985
14,16009041	2,494959385	1,009744693	1035,87746
59,81439643	4,471832088	0,430444017	1035,9065
10,74213629	1,592002167	0,137066046	1036,13538
9,033100712	1,56462332	1,014795714	1036,14402
8,300717242	2,279024248	1,373713359	1036,31834
37,10932562	2,552947802	0,379635899	1036,56538
34,17962532	3,595502099	0,48266605	1036,92306
24,1698597	2,828256658	0,493999253	1037,39154
8,056583932	3,019154758	0,799330299	1037,54234
43,45697273	3,940431457	0,434587527	1037,7581
10,98627875	3,860240539	0,603999266	1037,89786
33,69134956	2,489014728	0,479618586	1038,04154
17,333935	2,190647987	0,356626945	1038,3401
9,277232193	6,427210279	0,819196396	1038,60714
28,80854437	4,086821394	0,296567429	1038,6281
40,52726694	6,059237157	1,25653242	1038,78034
25,63471625	2,28251133	0,294405236	1039,67642
57,3729975	2,533487575	0,360757151	1039,7161
47,85150581	2,447559328	0,370129661	1039,95066
11,23037182	9,211473324	1,801140121	1040,0497
30,27338264	3,310612495	0,273959123	1040,9733
30,76166023	3,346739833	0,336746187	1040,9933
50,53705085	11,89756452	0,565265825	1041,20514
7,324039528	1,600115857	0,466653	1041,26482
26,85541937	4,001034995	0,358430498	1041,30458
24,65814095	3,096626407	0,838648488	1041,43506
6,591752984	2,354152853	0,181331642	1041,95714

23,43744331	3,475060841	0,670059719	1041,97178
14,40424018	2,95816523	0,787995221	1042,04858
37,59760139	4,217105642	0,384728944	1042,18098
26,36713629	3,496600366	0,529453766	1042,24802
42,72453622	2,780731756	1,34935155	1042,37506
18,79877875	1,793964539	0,245954458	1042,40026
41,5038459	2,415793783	0,321448718	1042,45258
25,63470527	3,556928213	0,433503834	1043,06186
16,60151129	3,056316768	0,848379863	1043,08194
5,859272587	15,52646725	2,910931244	1043,10978
18,55462349	3,530026574	1,15333137	1043,29178
21,48430733	3,731151229	1,085599075	1043,37706
12,69525215	1,406673464	0,60599859	1043,41194
15,38079902	5,177684961	0,670712765	1043,4273
23,43744879	3,705842045	0,378310041	1043,55106
31,49409125	3,185359182	0,450789	1043,67258
30,02924933	1,988611899	0,47156611	1043,71538
27,34367868	1,526168687	0,214535145	1043,76826
34,17963812	3,628887856	0,548230597	1043,8025
22,7050196	3,554286216	0,469664652	1043,8861
18,55463629	2,110534227	0,634153232	1044,27938
40,03901312	3,948087549	0,432365792	1044,31194
9,277270598	3,296394672	0,517998105	1044,38722
12,69525032	0,804405189	0,671236521	1044,4629
25,39056831	2,979471994	0,708847563	1044,58746
10,74213629	2,9164659	1,006932116	1044,72162
31,49409125	2,969130778	0,33976634	1044,95194
12,20697821	1,984360662	0,95618798	1045,03466
10,00967784	1,736235719	2,114627434	1045,09634
5,857701648	0,915447239	0,138517701	1045,31354
13,18353706	1,504264661	0,33199907	1045,38554
6,347586756	0,414140713	1,136529604	1046,0613
32,95892768	4,128028349	0,565127849	1046,07034
41,2597071	4,935575041	0,388178068	1046,2221
104,0038532	3,908194887	0,618576566	1046,2801
6,591738353	4,911254877	0,588443225	1047,0477
11,71868782	2,64004896	0,690284124	1047,19378
13,42766122	2,182055486	0,685089809	1047,2493
68,84759224	2,854508432	0,329157816	1047,38906
26,12297372	4,613354474	0,67692183	1047,47394
19,04289926	5,598013845	0,996266145	1047,52818
44,43353523	3,888041509	0,536075387	1047,7325
46,38666754	2,120507542	0,348257718	1047,80194
29,54095527	1,963231138	0,259193439	1048,27402
24,16985787	3,600002597	0,968379939	1048,5513

53,22260687	2,896416641	0,334355519	1048,6697
11,23040474	2,983531863	0,766397784	1048,68186
56,15229071	2,275487536	0,35804851	1048,6961
51,26947273	3,717920464	0,312380494	1048,77122
13,91595162	1,623398112	0,867947201	1048,9993
14,64838264	2,249070943	0,27031519	1049,07122
25,39057379	2,646740585	0,439876152	1049,21058
23,43742319	5,095751386	1,477332832	1049,27762
9,765562821	0,894439893	0,9359976	1050,03666
15,62494331	1,60805697	0,548798813	1050,19066
6,103449788	4,441023886	1,273329197	1050,38434
48,82807562	3,456640808	0,361412886	1050,4265
24,90228523	1,786430326	0,514283822	1050,54594
5,371000481	0,284500045	0,221329535	1050,70362
10,74212898	1,499891355	0,855312411	1050,7853
10,25385139	1,267291107	0,660797595	1051,19218
6,835864348	0,166858807	0,859197551	1051,4157
20,99600963	0,806222061	0,797359777	1051,73106
29,29681282	9,808086773	0,618570808	1052,10898
13,91596259	1,50036428	0,379422401	1052,21674
10,0097071	1,165494243	0,497332446	1052,2733
32,95891854	2,632922912	0,400496136	1052,6109
50,5370472	3,970132719	0,413374772	1052,65554
51,26946907	3,45767241	0,556922621	1052,70354
27,83198187	2,250225158	0,375199321	1053,4505
4,882755807	4,83343424	0,911110435	1053,58842
25,39056831	2,690819521	0,356647249	1053,66138
9,033155576	1,856120511	0,299332581	1053,96882
5,615192313	2,521643972	0,149998436	1054,0601
38,08587898	3,550909941	0,591124991	1054,49474
18,3104847	3,712026185	0,289449544	1054,5045
29,78508127	4,110064861	1,120797645	1054,52426
18,79876046	1,673350136	0,368490961	1054,57386
12,69525946	1,974157916	0,243157128	1054,60522
25,39056465	3,087285328	0,552532581	1054,65994
11,71869696	3,058043952	0,520887426	1054,98218
12,20697273	1,297548338	0,902220704	1055,00994
16,84564643	1,318200161	0,656499508	1055,15706
50,5370472	2,624878007	0,287129345	1055,21042
10,25385504	1,694082355	0,399999502	1055,22914
12,69524849	1,908606353	0,836797424	1055,24698
28,32026129	4,083001815	0,555427796	1055,42114
5,126887288	1,174250339	1,216997714	1055,5517
31,24995245	1,704600209	0,363076576	1056,03218
8,300726386	2,629672343	0,160531022	1056,0709

19,53119148	3,728676342	0,331622972	1056,17986
19,28705085	1,448903326	0,628362094	1056,19394
16,84564643	2,283814989	0,567655586	1056,22138
19,77532845	2,050211732	0,615843349	1056,29138
48,09564826	2,551879029	0,423532872	1056,41778
7,324140112	4,402082353	1,101426763	1056,43802
5,126889117	0,584124084	0,884039485	1056,57754
10,98624766	0,264057042	1,395163155	1057,39554
15,8690693	2,85514738	0,929998685	1057,61786
42,72455451	1,673117579	0,377175284	1057,80786
14,4042347	2,411533721	0,618132243	1057,95346
12,9393946	2,735263392	0,994932439	1057,99106
49,31634956	2,987406154	0,426959257	1058,00954
6,59174384	3,832924744	0,514797915	1058,05034
23,43744148	3,793560644	0,871198875	1058,28482
37,10931465	3,402298324	0,335364606	1058,40138
27,5878321	2,36193802	0,967053814	1058,5549
20,26361884	1,489964808	0,290499304	1058,7185
20,75188729	3,960905251	0,298631794	1058,7389
10,9862696	3,631253032	0,8177767	1058,78018
16,11320993	2,250003469	0,454854959	1058,82938
11,23041571	1,249290145	0,268797622	1059,15434
15,62492502	2,889083827	1,462799099	1059,16426
24,9022962	3,020509637	0,239417721	1059,20082
13,42767037	5,484867248	0,98499887	1059,31418
13,91595527	2,572086064	0,376950548	1059,92522
18,31049384	3,621015311	0,68749952	1060,2513
7,324163886	2,032149935	0,479996519	1060,3109
12,45112433	5,155165358	0,426398111	1060,40338
13,42766122	5,829755196	1,593141299	1060,41474
32,4706446	1,73301342	0,535832018	1060,5517
20,01947639	2,447104218	0,255598381	1060,57394
15,38072221	3,602278889	0,894656837	1060,69594
61,52338629	2,751096812	0,336050178	1061,12354
28,32025946	3,593226018	0,414856308	1061,33674
8,789011294	5,555213538	0,430132735	1061,4589
7,568309997	0,76268377	0,282969186	1061,4793
4,394469071	1,341295278	0,542220938	1061,50866
36,62103523	4,526692428	0,676824594	1061,6685
9,765571965	1,683863238	0,653332623	1061,76874
11,9628321	2,690443707	1,031313274	1061,97442
76,41596259	1,829241301	0,296719807	1062,01578
12,9393818	2,608848253	0,709331731	1062,06266
22,21674201	1,404766672	0,150918793	1062,0997
18,55462532	2,809585828	0,872683692	1062,28114

14,89252692	1,344022943	0,429990573	1062,3613
32,22649483	3,996003321	0,50879572	1062,39274
10,49799018	4,853485542	0,731109548	1062,40442
26,12299018	2,451674989	0,330856277	1062,57722
37,10931648	6,075392136	0,426338098	1062,67074
32,95892585	2,012059417	0,416702743	1062,73602
7,812446965	1,879826577	0,325332234	1062,96426
7,568277079	0,870694803	2,151997273	1062,9905
24,9022011	18,87263353	1,761824954	1063,99346
18,55463629	3,638872357	0,673142258	1064,13986
29,29680916	3,621581332	0,77799945	1064,20514
22,94916754	4,115482914	0,366798657	1064,24994
6,591740182	3,272603125	0,896490528	1064,8281
15,62494331	1,8974436	0,561110515	1064,8985
42,23627509	4,064010376	0,459564712	1065,00138
51,51361884	2,327013699	0,332629594	1065,02378
11,47448867	3,616089085	3,592364126	1065,16626
42,72455451	3,293510811	0,421586795	1065,3509
24,65814826	3,641693832	0,353776474	1065,41426
9,033139117	1,605117394	0,879321388	1065,8377
22,46086983	5,198169492	0,603477303	1065,88586
11,71869148	3,575485358	0,749088213	1065,93218
23,43743782	2,078032452	0,351046486	1066,45586
18,06635139	15,26019962	0,467691384	1066,50418
5,371044372	0,926817929	0,166797552	1066,62106
7,080015946	0,813366619	0,635866191	1066,6485
10,98624949	7,751865484	1,776663781	1067,16698
46,63080451	4,420158022	0,465284648	1067,38418
19,04290657	4,062705384	0,411197343	1067,48498
48,82806831	4,023589221	0,370423418	1067,66522
48,58392402	2,534728859	0,327223155	1067,76074
22,21673287	3,187956257	0,994266199	1067,84154
25,39057196	3,800052722	0,452363012	1068,24426
19,04291206	2,18980715	0,593329494	1068,27642
25,39055368	3,225207016	0,705903756	1068,36794
11,47455085	3,696052183	1,017890355	1068,63298
49,07221442	4,735432253	0,479237627	1068,96418
10,25385139	21,69871708	0,82559836	1069,53522
10,25383493	3,571350912	1,667998238	1069,62746
18,0663587	2,326549404	0,280399546	1069,64242
11,47456365	3,838636554	0,265331594	1069,71274
55,66400581	2,970605976	0,429536453	1069,93338
30,5175196	3,354247398	0,475546289	1070,1101
8,544859696	1,420623951	1,064442986	1070,50578
18,79876229	7,801824403	0,293413847	1070,51298

20,26361518	2,914428552	0,458054225	1070,68346
35,40033942	1,331671615	0,2329433	1070,71266
6,103438815	7,403299161	1,474284415	1070,72402
22,94915657	4,53761965	0,664265675	1071,0229
17,333935	1,636921898	0,326190088	1071,23162
45,41009224	3,01066941	0,378016797	1071,42682
41,99213446	2,283336276	0,30436848	1071,4353
22,70501777	3,177834757	0,495499192	1071,44458
23,68158576	3,003088786	0,697732591	1071,5109
52,00189643	2,686852511	0,290712992	1071,68442
30,76166389	3,390065069	0,561118451	1072,01778
7,324162057	3,117685556	0,525712949	1072,09466
13,42768317	2,774728951	0,716110085	1072,41506
22,70501777	1,676701262	0,941999302	1072,6145
6,835878978	1,49792286	0,392531034	1072,69314
22,46088629	3,603161446	0,684266118	1072,89506
13,67182014	1,159833331	0,379729385	1073,04946
15,38080817	1,862473436	0,256846467	1073,16194
10,98609587	5,235199865	7,867427793	1073,2953
26,85541571	4,48015781	0,421960103	1073,3041
6,591740182	0,75814984	0,616999454	1074,05178
34,17961983	6,088853378	1,018666115	1074,25642
9,033139117	3,053829391	1,041332842	1074,3721
37,59756115	2,881218091	1,171759298	1074,61986
25,39057196	1,688227858	0,337435028	1075,26058
23,19330451	2,59723513	0,518998657	1075,33234
11,47453805	1,162503457	0,82199798	1075,41114
26,6112696	3,588055207	0,723996573	1075,43042
16,84563546	5,346699421	0,78196229	1075,48138
14,89251229	0,176608298	0,841398737	1075,61762
21,72846625	3,146679972	0,462797522	1075,65402
10,74213081	4,587741325	0,98959916	1076,59898
15,38080268	6,370700736	0,43306634	1076,67386
10,25385504	1,070501237	0,493332322	1076,70498
30,76166023	4,252094885	0,516189901	1076,96386
44,18940375	2,205181032	0,301589356	1077,21178
10,98626777	1,760403928	1,439997843	1077,58026
9,521422196	1,524120129	0,935332671	1077,6725
12,20690141	10,69291351	3,901332245	1077,92314
27,09954537	5,768463859	0,700094325	1078,53402
9,765568307	3,529496453	0,409998526	1078,5977
17,08978889	4,441637466	0,813331952	1078,74546
9,521422196	2,78409394	0,773332034	1079,05042
7,568282565	2,494694986	1,505887937	1079,4389
23,68159125	5,480936157	0,537671922	1079,47802

30,76165291	4,601493802	0,754132783	1079,61378
14,16009407	7,125226399	0,795197563	1079,68722
27,34368599	2,719893998	0,495058249	1079,84538
16,11321541	1,924349268	0,825903582	1080,06306
45,89838081	3,029545192	0,382666105	1080,36538
23,43743416	3,375026702	0,340887891	1080,49698
16,35734872	1,156873848	1,244798653	1080,56322
25,39055733	1,933772948	0,587998889	1080,69866
28,32025946	2,39003702	0,536846501	1080,77994
30,76165108	3,373972207	1,15110953	1080,80874
21,24015025	4,173431049	1,876856505	1080,8229
20,50775215	2,090428534	0,42387563	1080,86218
23,19330085	3,537567403	0,412665443	1080,89666
31,00580085	4,176118191	0,392664514	1080,93186
12,20697821	1,68005664	0,991998778	1081,11274
7,812435992	6,570062994	1,00266319	1081,32314
20,75190192	2,091939548	0,304834376	1081,4525
9,765573794	1,705079521	0,306663779	1081,96194
11,96284856	7,108366384	0,181199374	1081,98906
21,48423052	4,279550345	1,009773316	1082,04602
34,9120472	3,02671362	0,275199225	1082,36178
28,07611335	3,332693161	0,585713403	1082,3981
31,00580817	3,21784319	0,345333034	1082,6205
10,49799933	2,516837954	0,658199136	1082,7093
66,40618965	2,957043397	0,350823648	1083,13546
8,056589419	1,865687508	0,323187972	1083,49282
54,93157662	2,713379347	0,333060933	1083,7033
17,82221625	2,500784536	0,279332726	1083,92218
8,789003978	2,471118563	0,438283995	1083,9501
12,20698919	3,723625584	0,20830877	1084,14586
24,16986884	6,366700206	0,726662543	1084,31658
24,16985604	3,052293744	2,56266699	1084,39458
36,8651722	3,633370216	0,385554437	1084,51346
26,1229975	2,322502259	0,329007594	1084,55514
56,1522834	4,213382465	0,419272103	1084,61962
8,056527239	0,181027449	3,707428293	1084,94514
10,98626595	3,893167563	1,096666041	1084,9877
11,23041937	3,010486173	0,312966298	1085,1561
8,788983862	1,345644871	1,090665145	1085,32954
5,615168538	1,634957577	1,25266389	1085,7437
28,80853706	3,957069496	0,807770161	1086,02554
40,28315009	4,165503254	0,312544933	1086,15274
12,45111335	2,487921798	1,02570656	1086,18562
17,82218699	2,125828108	1,241902674	1086,55322
64,69721259	2,897229715	0,345019008	1087,03466

29,29681282	2,136037742	0,290792989	1087,17074
17,8222071	2,036088792	0,915191064	1087,21154
15,86906747	7,189070959	0,801996741	1087,27298
13,67179088	6,562435869	2,158475402	1087,31954
30,02923835	2,280158414	0,240849596	1087,41954
27,09954171	6,541783409	0,922561466	1087,56842
51,26948004	2,809640763	0,282903512	1087,5901
34,17963812	2,880693143	0,3029889	1087,68154
33,44721076	1,836698803	0,45393897	1087,80914
32,47067752	2,613984254	0,427449783	1087,81298
35,40032845	4,059438017	0,451332619	1087,89074
7,568308169	5,274821578	0,462817326	1087,97986
5,859307334	0,955634477	1,276566622	1088,0849
35,40032845	3,488054623	0,846663362	1088,35346
21,48430916	4,534976487	1,247999121	1089,05186
31,9813758	0,169529346	0,291525313	1089,10498
11,47454354	3,096403023	0,667887503	1089,24962
39,06245062	2,91782879	0,314175894	1089,7257
6,591729209	1,391123105	1,299799142	1089,87626
4,150332103	1,033471265	0,729998493	1089,89634
21,72842967	5,275519132	0,954410861	1089,97498
21,72846076	2,738288933	0,320035805	1090,02674
14,892494	14,76408939	1,548532507	1090,59826
21,48423601	3,202751197	1,091194123	1090,68898
35,88855849	3,630039647	0,567287589	1090,94386
20,50775032	3,106743162	0,425845145	1091,17074
14,16009956	2,615516874	0,541764135	1092,08634
18,55462898	5,63681372	0,902398179	1092,17282
23,92573004	2,368416364	0,407137687	1092,23138
22,46087715	2,040449339	0,291265754	1092,35658
36,13275215	3,504045635	0,378233661	1092,68618
8,789013122	0,818050299	0,791272689	1092,89754
9,033139117	1,226352942	0,341198116	1093,43962
18,55463081	3,040490032	1,025599389	1093,52186
5,615146593	5,081761457	0,522169333	1093,6813
13,18352243	0,116613817	0,662180628	1093,75034
17,57806099	2,075955181	0,349713133	1093,78778
58,59369514	4,369894388	0,400859867	1093,9333
13,91592784	0,247852748	0,152124346	1094,13762
26,85541023	3,00834979	0,660219017	1094,19458
14,64838446	2,071864202	0,459787219	1094,22186
14,1600959	2,974025508	0,928265955	1094,31626
27,0995472	3,00804564	0,54039895	1094,35986
10,98618914	0,615308949	0,513321786	1094,45666
9,277276084	4,948833992	1,842793274	1094,48874

23,92572639	4,040072461	0,765998979	1094,5893
8,544843237	2,751572527	1,816664771	1094,69786
26,61125863	9,519633038	0,836132336	1094,74098
18,0663459	1,21529214	0,81955448	1094,79754
6,591731038	5,124275186	2,216999997	1094,94242
9,277265112	10,66331701	1,091196595	1094,98778
19,04291937	2,557916559	0,311826372	1095,06714
33,69135139	4,162344055	0,757646602	1095,11154
51,02533393	2,529370994	0,330923314	1095,15066
7,080001315	2,391117781	1,316219331	1095,21746
20,7518946	1,70505207	0,353998908	1095,23714
29,29681648	4,063389886	0,227973384	1095,24482
39,30658759	3,569503898	0,593660822	1095,33546
11,23034073	6,410104065	4,331333877	1095,47378
4,394441639	0,460697424	0,922054929	1095,56682
23,19329537	5,517157167	0,845097548	1095,7737
7,812401245	4,910938451	1,879992974	1095,93842
21,48432928	1,502387033	0,511713207	1095,95082
9,277131609	12,02045987	6,067997461	1096,07266
12,93940192	2,37291616	0,829998785	1096,12898
38,08588264	3,820810034	0,605932514	1096,25138
8,300698954	2,370706719	1,969598374	1096,29226
22,46088446	1,546212729	0,292932599	1096,33642
15,62493782	2,749924965	0,907998891	1096,48274
6,103447959	1,308150631	0,458179239	1096,53674
32,47058974	2,018010216	0,624285621	1096,59058
4,88262962	8,172488226	0,323804087	1096,61474
18,31038594	4,540061162	1,337328093	1096,79618
29,05266854	2,418327235	0,69742678	1096,87458
11,47453439	2,061537665	1,157996768	1096,9217
9,521416709	3,288405544	0,965327637	1097,0321
10,25369777	13,16805654	5,898663875	1097,07474
9,033038533	2,441017556	1,497593614	1097,16498
8,544801174	1,014429232	0,870254129	1097,18418
4,394472728	0,477178312	0,866665092	1097,2109
4,882699114	6,994007477	0,159996434	1097,2389
8,544856038	1,45403281	1,342221422	1097,26882
37,10923784	8,002412609	0,34079839	1097,39418
9,277299859	2,450047564	0,559821504	1097,6141
16,35736152	3,01362682	1,138131786	1097,6833
46,38666206	2,156414453	0,334666044	1097,86362
7,568267935	0,151484346	2,459731982	1097,89522
10,49793166	2,595128071	3,426714123	1098,00186
6,103429671	1,657455258	1,962856074	1098,25402
8,056066382	0,544500847	7,987983254	1098,33874

8,056569302	0,618717342	1,597198244	1098,44026
7,08002509	2,607443148	1,649999851	1098,56234
6,835844231	3,804400574	2,847998896	1098,7113
15,86908027	3,006013683	1,219999066	1098,79754
50,53705085	3,146201439	0,333939019	1098,8749
10,49794446	2,176087252	4,143000637	1098,89858
7,079959253	11,72535306	2,140505658	1099,02154
7,812406731	0,450060486	2,469329445	1099,38522
37,10916835	2,468146929	0,381025448	1099,45354
25,87884956	5,537587673	0,412569763	1099,53018
8,788923511	0,463830059	4,578285272	1099,62146
22,94916937	2,763108525	0,450766433	1099,85242

