



# Teoría de la información para EEG en MATLAB

---

## Visión general

El objetivo general de este proyecto es la creación de una herramienta de MATLAB (es decir, una librería) para facilitar el análisis de EEG (electroencefalografía) desde el ámbito de la Teoría de la Información.

Esta herramienta facilitará la importación de datos desde BrainVision en uno (o diversos) formatos, el acotado de esta información y el estudio de las diferentes características de las señales importadas.

Todas las partes de este documento irán siendo modificadas conforme el proyecto avance. Última modificación el 11 de marzo del 2020.

## Objetivos

1. Creación de la librería de análisis de señales de EEG
2. Complementación con análisis de otras señales (audio, vídeo...) mediante un sistema de estandarización de entrada
3. Integración «*plug & play*» con otras librerías como EEGLab, etc.

## Especificaciones

En este apartado se exponen las especificaciones necesarias de la aplicación, para poder hacer más fácil su implementación.

Hay que tener en cuenta que los datos que vamos a estar manejando son señales electroencefalográficas, que digitalmente se pueden tratar como cualquier otra señal digital.

## Carga de archivos de datos.

La herramienta tiene que tener la capacidad de leer archivos. En concreto, los que exporta el programa BrainVision. BrainVision exporta sus datos o bien en archivos .mat (los cuales, personalmente, no he manejado) o .vhdr.

Los archivos de formato vhdr, también denominados *header* (cabeceras), contiene la información de todos los datos exportados por BrainVision, tanto el nombre del resto de archivos, como la frecuencia de muestreo, el nombre de los canales, qué referencia se utiliza, etc.

Junto con las cabeceras, BrainVision exporta otros dos archivos: un archivo .dat que contiene directamente los canales del EEG:

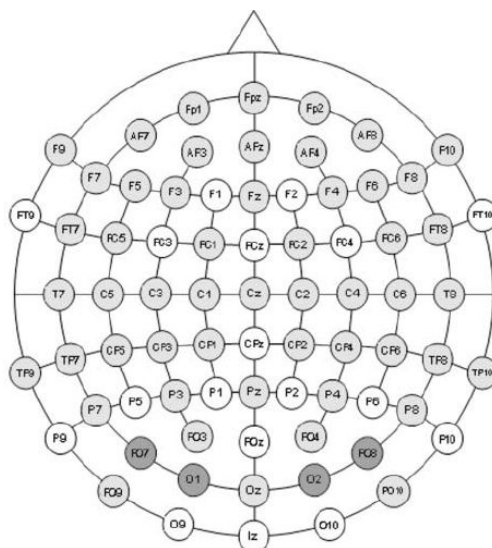
- Un archivo .vmrk (marcas) que contiene los puntos de los datos en los cuales hay marcas relacionadas con el experimento. Cada marca contiene un código, normalmente especificado por quien ha realizado el experimento, que indica algo sobre el desarrollo del mismo.
- Un archivo .dat (datos) que contiene todos los canales con sus valores de voltaje (no contiene el valor de las impedancias). El sistema de guardado es vectorial (cada línea, tiene primero la *cadena* que señala el nombre de canal y, posteriormente, los valores separados por espacios. Las unidades están marcadas en el archivo de cabecera.

Aun así, quizás sería bueno utilizar el programa de Python que ya existe para segmentar los experimentos. De esta manera la herramienta sólo tendría que cargar segmentos que, actualmente, son archivos .dat que contienen la cabecera con todos los metadatos y luego, línea a línea, todos los valores de los canales, separados por espacios y siendo el primer valor el nombre del canal.

*Posibilidad futura:* Que la propia herramienta pueda segmentártelo de una manera estándar a definir (creo que la que hice con Python no es nada mala, aunque mejorable).

## Esquema de la composición de canales

Una imagen de cómo están colocados los electrodos en la cabeza de los sujetos (en el caso de nuestro laboratorio, es el sistema 10-20 extendido) para acompañar los canales.



Este esquema necesitará que tenga **botones** vinculados a los distintos puntos de los canales, con lo cual de esta manera las personas que utilicen la herramienta puedan seleccionar los canales de interés para el análisis. Un **interruptor** o algo similar para seleccionar o deseleccionar todos los canales estaría bien, para facilitar la selección.

Esta selección de canales afectaría a cómo el programa selecciona los canales para un análisis posterior. Así también se podría implementar rangos de los segmentos a utilizar y otras cosas que sean útiles

*Posibilidad futura:* Definición (quizás una herramienta como pincel o distintos tipos de click) de zonas de interés (*ROIs*), para realizar análisis con la media de ellas como si fuera una única señal.

## Desglose por bandas de frecuencia

El poder definir bandas de frecuencia para realizar el análisis en distintas bandas de frecuencia es una gran característica. Poder decidir **cuántas** y los **distintos límites** de las mismas facilitaría que las personas que usan la herramienta tengan flexibilidad y puedan usarla para variedad de experimentos.

*Posibilidades futuras:*

- Se podría también hacer algo con las combinaciones de las bandas (como una media de las señales de varias bandas).

- Elección de los filtros a utilizar para la realización (por ahora lo estoy haciendo con un filtro Butterworth de 3er orden aplicado *forward and reverse*).

## Cálculo de la señal analítica de cada una de las señales

Mediante la transformada de Hilbert (que seguro que Matlab tiene una función para ella), podemos obtener la llamada «señal analítica» de la señal inicial. Con esta señal analítica, podemos calcular la envolvente de las señales, la fase instantánea (con la cual podemos obtener fácilmente la frecuencia instantánea)...

Entonces, el poder seleccionar el cálculo de la señal analítica mediante un **botón**, que de lugar a un **desplegable** o una **lista con seleccionables** que permita qué parámetros de la onda analítica estudiar (incluso, quizás, qué combinaciones tener en cuenta, pero eso para un futuro).

## Discretización

Para hacer cálculos dentro de la teoría de la información (una teoría que está definida para grupos discretos) deberíamos de discretizar los datos una vez preparados.

Poder mostrarle los límites máximos y mínimos de cada señal/grupo de señales al usuario y dejar que seleccione en cuántos grupos se separa, los límites de los mismos (si son grupos del mismo tamaño o diversos, si son grupos porcentuales, si cada canal tiene que discretizarse relativo a sus propios límites...).

Quizás este ámbito da para mucha prueba y error, pero también es algo muy importante a la hora de optimizar y jugar con la técnica de análisis, lo cual es muy importante.

## Tipos de cálculos de información

Los diversos cálculos que se harán. Ya bien sea sólo de la entropía de cada señal (para lo cual no se necesita más), o de cosas como probabilidades conjuntas, condicionales, información mutua...

En este aspecto hay mucha variedad y, quizás, dificultad de diseño intuitivo. Quizás una pequeña terminal donde puedas escribir mediante una sintaxis fácil podría ayudar a simplificar las cosas, si no, otro tipo de selección puede ser útil.

## Hitos

### I. Carga de segmentos y entropía directa de amplitud

Poder cargar un segmento, seleccionar canales (quizás todavía no un display con la imagen de la disposición de electrodos) y que te calcule la entropía de cada canal según los criterios dados (todavía sin filtrado de banda ni señal analítica, directamente en amplitudes, quizás con discretización de 0-25-50-100%).

### II. Desglose por bandas de frecuencias

Permitir que la aplicación te haga un filtro de una banda y calcular la entropía de esa banda.

Poder modificar los límites de la discretización, quizás en número de grupos.

### III. Señales analíticas

Calcular la señal analítica de todos los canales del segmento dado. Poder elegir qué parte de la señal estudiar, y continuar con la selección de discretización previa y las bandas de frecuencia.

Poder seleccionar varias bandas de frecuencia a estudiar.

### IV. Combinaciones de bandas

Como en Cogan y Poeppel (2011), el uso de bandas combinadas para calcular la sinergia y el *overlap* de la información de las bandas.