



Laboratorio de Análisis en Neurociencias  
Versión en prueba v.0.1.4.1

Pablo Billeke  
Rodrigo Henriquez  
Francisco Zamorano  
[LAN.toolbox@gmail.com](mailto:LAN.toolbox@gmail.com)

29 de agosto de 2013

# Índice general

|          |   |           |
|----------|---|-----------|
| <b>1</b> | <b>Introducción</b>                     | <b>2</b>  |
| 1.1      | Instalación                             | 2         |
| 1.2      | Estructura del toolbox / Manual         | 2         |
| <b>2</b> | <b>Datos</b>                            | <b>4</b>  |
| 2.1      | Estructura simple LAN                   | 4         |
| 2.2      | Estructura grupal GLAN                  | 6         |
| 2.3      | Estructura COR                          | 7         |
| <b>3</b> | <b>Ejemplos</b>                         | <b>8</b>  |
| 3.1      | Importación y pre-procesamiento         | 8         |
| 3.2      | Potenciales relacionados a eventos      | 12        |
| 3.3      | Análisis Psicofísicos                   | 14        |
| <b>4</b> | <b>Interfaz Gráfica</b>                 | <b>15</b> |
| <b>5</b> | <b>Módulos</b>                          | <b>16</b> |
| 5.1      | Módulo Básico                           | 16        |
| 5.2      | Módulo de Segmentación                  | 17        |
| 5.3      | Módulo de Preprocesamiento              | 18        |
| 5.4      | Módulo de Potenciales Evocados          | 20        |
| 5.5      | Módulo de Análisis de Frecuencias       | 22        |
| 5.6      | Modulo de Redes                         | 25        |
| 5.7      | Módulo de Estadística                   | 26        |
| 5.8      | Modulo de Modelos y Tiempos de Reacción | 27        |
| 5.8.1    | Funciones entre R-Matlab                | 29        |
| 5.9      | Otras                                   | 30        |
| <b>6</b> | <b>Algunas justificaciones teóricas</b> | <b>31</b> |
| 6.1      | Entropía                                | 31        |
| 6.2      | Estadística                             | 31        |

```

-----
      /°   #/|
matlab: > LAN  toolbox
      \____#\|

```

# Capítulo 1

## Introducción

**LAN: Laboratorio de Análisis en Neurociencia**, es un toolbox para MATLAB, para el análisis de datos neurocientíficos (EEG y tiempos de reacción por el momento), cuyo propósito es establecer un lenguaje común entre procedimientos y algoritmos de otros softwares del área (como Fieldtrip, Eeglab, Cronux, Brainstorm etc) que posibilite el desarrollo de análisis experimentales para el usuario medio del área.

```
>> disp_lan

TOOLBOX:          <°LAN><]

      laboratorio de análisis en neurociencia

>>
```

### 1.1 Instalación

**LAN** se puede obtener en nuestra WIKI: <http://www.lantoolbox.wikispaces.com>, donde se dispone de una versión revisada y la versión actual de los desarrolladores. Para instalarlo solo hay que descargarlo y dejar la carpeta en el path de Matlab. Se puede solicitar un versión a [LAN.toolbox@gmail.com](mailto:LAN.toolbox@gmail.com).

Para realizar los modelos algunas funciones utilizan el programa R (se pretende en algún momento no requerir de este). Por este motivo se necesita que este programa este instalado y que también este instalado el script littler. Lamentablemente este modulo esta solo implementado para sistemas UNIX. Este script esta disponible gratuitamente en <http://dirk.eddelbuettel.com/code/littler/>. LAN, en la carpeta LAN/RT/littler/, dispone de la versión 0.1.3 (ver también <http://dirk.eddelbuettel.com/code/littler.html>). Instalar siguiendo las instrucciones en INSTALL.

### 1.2 Estructura del toolbox / Manual

El proyecto aun se encuentra en fase de prueba. El primer capítulo (Capítulo 2) muestra como se estructuran los datos en el sistema. Existen principalmente datos individuales (Sujetos para el análisis electroencefalográfico) y grupales (Grupo de sujetos). Se especifican los campos necesario y optativos para cada uno. Luego (Capítulo 3) se presentan ejemplos de como utilizar el toolbox para diferentes procesamientos. En el capítulo 5 se presentan los módulos que componen el Toolbox. En la versión actual se han implementado siete módulos. El primer módulo (Módulo básico) incluye funciones esenciales del sistema. El módulo de segmentación incluye herramientas de

importación de datos y segmentación. El módulo de pre-procesamiento incluye funciones para realizar detección de artefactos, interpolas canales y una visualización (GUI) de datos segmentados. El módulo de ERP, incluye estadísticas no-paramétrica y corrección basada clusters. El modulo de tiempo-frecuencia, incluye varios métodos de análisis (Fourier por ventanas, Hilberts y la implementación de Fieldtrip para *multitaper* y *wavelet*), estadística no paramétrica y una interfaz gráfica para la exploración de los resultados. En proceso de implementación están los análisis de fase y sincronía. El modulo de estadística, incluye las funciones básicas usadas en el modulo ERP y tiempo-frecuencias. Por último el modulo de modelos y tiempos de reacción esta en fase de implementación para realización de análisis más complejos entre los datos conductuales y electrofisiológicos (incluye hasta el momento realización de modelos mixtos usando R ). Finalmente, se desarrolla un capitulo teórico (Capítulo 6) que respalda algunos de los algoritmos implementados.

# Capítulo 2

## Datos

Estructura de los datos: Existen dos tipos de estructuras, las estructuras simples, que representan los análisis de un sujeto, designada generalmente por las iniciales del nombre del sujeto (en este manual se representan por la sigla LAN). También existen las estructuras múltiples que representan promedios de sujetos para los análisis grupales y estadísticos (en este manual representadas por GLAN)

### 2.1 Estructura simple LAN

Campos

| Campos Obligatorios |  |
|---------------------|--|
| LAN.srate           | = Frecuencia de muestreo   |
| LAN.data            | = Matriz del EEG (tres dimensiones: electrodo, tiempo, ensayos)<br>= o celda {ensayos} con matrices 2D [electrodo, tiempo]                               |
| LAN.time            | = Tiempo inicial(en segundos) - Tiempo final (en segundos) - Punto cero (en punto)<br>= e.g: [-1 0.5 3457]<br>= Para épocas variables [épocas x timepos] |
| LAN.event           | = Evento en formato EEGLAB   |
| LAN.delete          | = Guarda información eliminada del data original   |

| Campos Secundarios         |   |
|----------------------------|---|
| LAN.nbchan                 | = Número de canales   |
| LAN.pnts                   | = bins temporales por trials, vector cuando trials son de duración variable   |
| LAN.trials                 | = Número de ensayos   |
| LAN.accept                 | = vector logico con los trials buenos   |
| LAN.tag                    | = estructura de tag por trials y canal  |
| LAN.tag.labels             | = Etiqueta de la marcas o tag   |
| LAN.tag.mat                | = matriz de tag [nbchan x trials]   |
| LAN.trials_latency         | = Duración de cada trials   |
| LAN.freq                   | = Estructura con los datos del análisis de frecuencia   |
| LAN.cond                   | = Condición en texto ['Ojos Abiertos'],<br>= si existe una condición por estructura<br>= o estas están separadas en diferentes celdas.  |
| LAN.conditions             | = Estructura que delimita las diferentes condiciones que existen en la estructura LAN.<br>= Esta forma es preferible cuando existen varias condiciones y algunas se sobrelapan<br>= (es decir, hay ensayos que pueden pertenecer a más de una condición), ver más adelante. |
| LAN.name                   | = Nombre del sujeto ['Pedro']   |
| LAN.group                  | = Nombre del grupo al que pertenece el sujeto ['Control']   |
| LAN.xmax                   | = Tiempo final de registro “epoquiado”  |
| LAN.xmax_c                 | = Tiempo final de registro continuo   |
| Condiciones LAN.conditions |   |
| .name                      | = Celdas con los textos de los nombre de las condiciones  |
| .ind                       | = Celdas con matrices que presentan el índice lógico de que ensayo es de cada condición   |
| LAN.event                  |   |
| event.latency_aux          | = Latencia del evento con respecto la registro continuo   |
| event.latency              | = Latencia del evento con respecto al registro epoquiado  |
| event.type                 | = Tipo de evento  |
| event.duration             | = Duración del evento   |
| LAN.freq                   |   |
| freq.fourierp              | = matriz densidad espectral por fourier (sin tiempo)  |
| freq.powspctrm             | = matriz de poder inducido  |
| freq.fourierspctrm         | = matriz de coeficientes de fourier inducido  |
| freq.time                  | = Linea de tiempo   |
| freq.freq                  | = Eje de las frecuencias  |
| freq.evo.powspctrm         | = Matriz del poder evocado  |
| freq.cfg                   | = Configuraciones   |

## 2.2 Estructura grupal GLAN

Esta estructura guarda análisis grupales, los datos de cada individuo deben estar idealmente en la misma carpeta o en el **path**, y los nombres deben ser especificados en el campo **GLAN.subject** y deben estar adecuadamente estructurados según el tipo de análisis a realizar. En algunas funciones esta habilitado el parametro de **file.prefix** donde se puede usar el path específico de la archivo **.mat** de los sujetos (ver por ejemplo [timefreq\\_stata.m](#))

Campos

| Campos Obligatorios               |   |
|-----------------------------------|---|
| <b>GLAN.subject</b>               | = Sujetos que representa, en string con los nombre de los archivos <b>.mat</b><br>= [{ 'str' } , { 'str2' } ; ...]                                  |
| <b>GLAN.cond</b>                  | = vector con el índice de las condiciones en las estructuras LAN<br>= [n,n,...]   |
| <b>GLAN.group</b>                 | = vector con el indice de los grupos de los sujetos, debe coincidir con el número<br>= de filas de <b>subject</b><br>= [{ 'str' } , { 'str2' } ...] |
| <b>GLAN.srate</b>                 | = Frecuencia de muestreo.   |
| <b>GLAN.time</b>                  | = Tiempo.   |
| ERP GLAN.erp.                     |   |
| <b>erp.data</b>                   | = celdas de vector o matriz de los ERP promedios de los sujetos por condición<br>= [{erp condición 1},{erp condición 2}]                            |
| <b>erp.comp</b>                   | = celdas con las condiciones comparadas para la estadística<br>= [{[n1 n2]},{[n1 n3]},...]  |
| <b>erp.pval</b>                   | = matrices de los valores p, en celdas por comparación realizada  |
| <b>erp.hh</b>                     | = matrices lógicas de significancia, en celdas por comparación realizada  |
| <b>erp.hhc</b>                    | = matrices lógicas de significancia, corregida por comparaciones múltiples,<br>= en celdas por comparación realizada                                |
| <b>erp.stat</b>                   | = Resultados del estadístico utilizado, en celdas por comparación realizada   |
| <b>erp.cluster</b>                | = Índice de los cluster de electrodos, en celdas por comparación realizada  |
| Tiempo-frecuencias GLAN.timefreq. |   |
| <b>timefreq.data</b>              | = celdas de vector o matriz de los cartas tiempo frecuencias promedios<br>= de los sujetos por condición<br>= [{TF cond 1},{TF cond 2}]             |
| <b>timefreq.comp</b>              | = celdas con las condiciones comparadas para la estadística<br>= [{[n1 n2]},{[n1 n3]},...]  |
| <b>timefreq.pval</b>              | = matrices de los valores p, en celdas por comparación realizada  |
| <b>timefreq.hh</b>                | = matrices lógicas de significancia, en celdas por comparación realizada  |
| <b>timefreq.stat</b>              | = Resultados del estadístico utilizado, en celdas por comparación realizada   |
| <b>timefreq.cluster</b>           | = Índice de los cluster de electrodos, en celdas por comparación realizada  |

## 2.3 Estructura COR

Estructura para guardar datos para realizar correlaciones entre datos EEG y conductuales.

| RT. tiempos de reacción                                    |  |
|--|--|
| RT.rt  | = tiempos de reacción (unidades en <code>RT.conf.unit</code> ) |
| RT.laten   | = Latencia de aparición del estímulo en el experimento         |
| RT.est   | = Código de estímulo   |
| RT.resp  | = Código de respuestas   |
| FREQ. frecuencias de ventanas tiempo-frecuencia de interés |  |
| FREQ.  | = ...  |
| OTHER. campos dependientes del experimento                 |  |



# Capítulo 3

## Ejemplos

### 3.1 Importación y pre-procesamiento

EN utiliza funciones exportadas de EEGLAB para importar datos a matrices de matlab. En el ejemplo se importa un archivo .egg de Neuroscan utilizando la función `eeg2eeglab2lan.m`. En cada celda de la variable LAN dejamos cada condición de sujeto. Podemos configurar el ID del sujeto, el nombre de la condición y el grupo. También se pueden cargar posiciones de electrodos que se encuentran en el directorio de EN. En este caso utilizamos la gorra de 40 canales NuAmp de Neuroscan, que se encuentra en el archivo `chanlocs_40neuroscan_nuamp(40).mat`

```
1 load chanlocs_40neuroscan_nuamp(40) % importar chanlocs con la posicion de los electrodos
2 LAN = [] ;
3 LAN{1} = eeg2eeglab2lan('CON_1.eeg'); % Condicion uno
4     LAN{1}.name = 'S01'; % ID del sujeto
5     LAN{1}.cond = 'CON1'; % Nombre de la condicion
6     LAN{1}.group = 'CASO'; % Nombre del grupo
7     LAN{1}.chanlocs = chanlocs; % posicion de los electrodos
8 LAN{2} = eeg2eeglab2lan('CON_2.eeg'); % Condicion dos
9     LAN{2}.name = 'S01'; % ID del sujeto
10    LAN{2}.cond = 'CON2'; % Nombre de la condicion
11    LAN{2}.group = 'CASO'; % Nombre del grupo
12    LAN{2}.chanlocs = chanlocs; % posicion de los electrodos
```

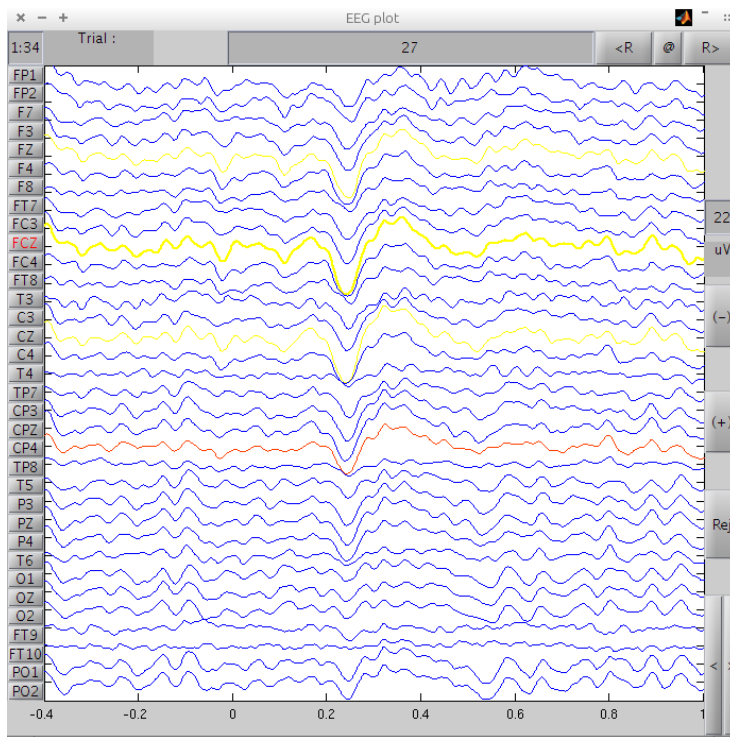
En el preprocesamiento se pueden eliminar electrodos no necesario como los oculares y las referencias, usando `electrode_lan.m`. Luego se pueden aplicara funciones para detectar artefactos. Están habilitados umbrales de voltaje y umbrales de amplitud por frecuencias. Ver las funciones `vol_thr_lan.m` y `fftamp_thr_lan.m`.

```
1 LAN = electrode_lan(LAN, 'REF'); % Eliminar los electrodos de referencia
2 %LAN = electrode_lan(LAN, 'T6'); % Eleminar un electrodos particular
3 LAN = electrode_lan(LAN, 'EOG'); % Eliminar los electrodos oculares
4 LAN = vol_thr_lan(LAN, 75, 'bad:V'); % Marac electrodos/esayos con diferencia de amplitud
5                                     % mayores a 75 micro volts
6 cfgF = [];
7 cfgF.frange = [1 30]; % rango de frecuencias
8 cfgF.method = 'ft'; % metodo, mutitaper
9 cfgF.thr = [3 0.2]; % Umbral: desviaciones stadar, porcetaje de frecuencias sobre el umbral
10 cfgF.cat = 1; % Calcula la desviacion standar concatenando todas las condiciones
11 LAN = fftamp_thr_lan(LAN, cfgF);
```

Para realizar un inspección ocular de los canales/ensayos marcados, épocas rechazadas, y realizar cambios de algunos parámetros, utilizar la función `prepro_plot.m`.

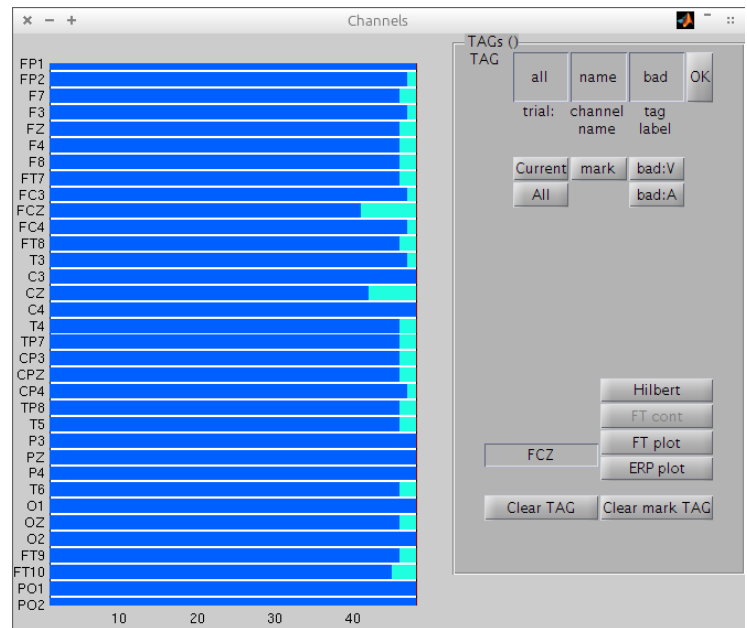
```
1 prepro_plot(LAN) % GUI para inspeccion visual
```

Esta GUI consta de tres ventanas. La primera, es la GUI de controles, donde se muestra como esta constituida la estructura LAN. Las condiciones con sus nombres, el ID del sujeto y a que grupo pertenece. También se pueden modificar estos nombres.

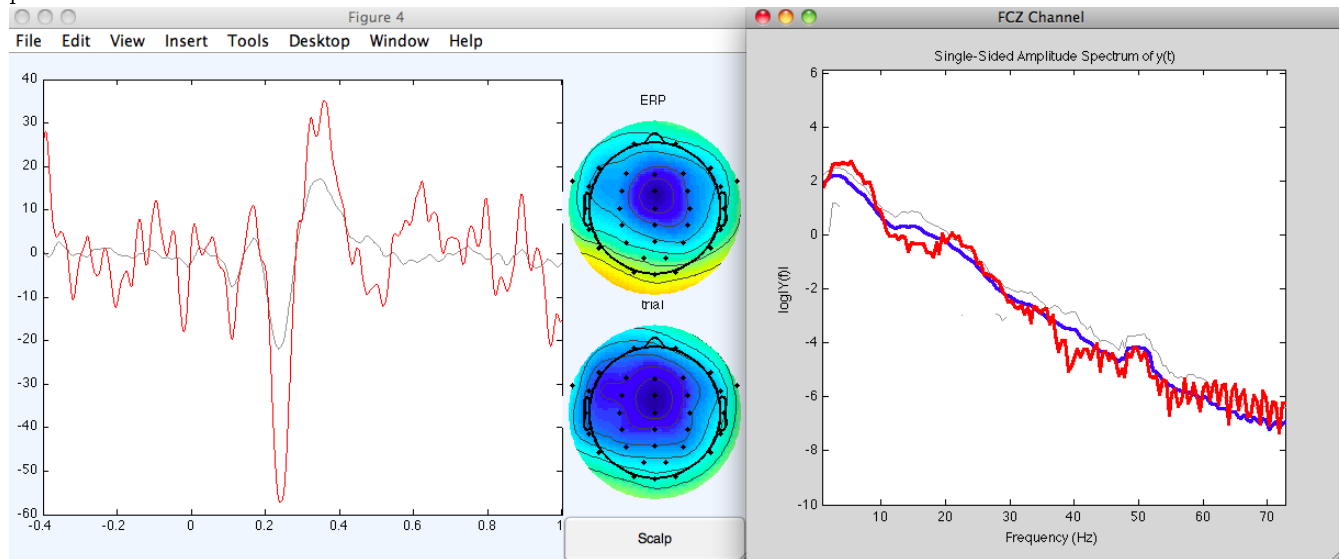


La segunda GUI (EEG plot), se pueden revisar el registro segmentado por ensayo. Los ensayos se pueden cambiar por los botones **<R** y **R>**, o directamente poniendo en número del ensayo que se desea en la cuadro editable **Trial:**. Cada canal se puede resaltar con el boton que indica su nombre, y con eso etiquetar o desetiquetar usando al tercera GUI (**Channels**).

La tercera GUI (**Channels**) es la herramienta para etiquetar las ensayos/canales. Además está la posibilidad de ver los ERP y la distribución topográfica por ensayo, y el gráfico de la amplitudes por frecuencias. Para este ultimo es necesario haber realizado previamente un detección de umbral de amplitud con al función `fftamp_thr_lan.m`, o usando el equivalente en la GUI de controles.



A continuación se muestra el resultado de ERP plot y de FT plot para el canal/ensayo señalado en el ejemplo.



Así, resulta fundamental toda esta visualización para evaluar si lo detectado por la automáticamente es realmente artefactual o no. En este caso representa solamente un sujeto que presenta en un ensayo único un prominente N2, por lo cual no es un artefacto. Usando los botones **clear TAG** se borran todas las etiquetas del ensayo, o usando **clear mark TAG** se borran solo las etiquetas de los canales resaltados o marcados.

El panel de control TAG que se encuentra en esta ventana es para cambiar o agregar TAG por canal y ensayo. En la ventana muestran los TAG disponibles, pero se pueden agregar los que uno quiera. Se pueden cambiar el TAG de canal solo del ensayo a del todo la condición, usando el primer casillero ('c', solo el ensayo actual; 'all', todos los ensayos). La lógica de esto es poder marcar ciertos canales/ensayos para después poder realizar un proceso en estos. Por ejemplo se pueden dejar como 'bad' los canales que se quieran interpolar. Luego de terminado el trabajo en la GUI se guarda los cambios (la variable) en el espacio de trabajo, mediante el botón **save (WS)**. Luego simplemente

cerramos la GUI con el boton `Close` (debe antes guardar en el espacio de trabajo la variable para no perder la información!). Luego se puede ocupar la funcion `lan_interp.m` para interpolrar todos los canales/ensayos marcado de la siguiente forma:

```
1 cfgI = [];  
2 cfgI.label = 'bad';  
3 LAN = lan_interp(LAN, cfgI);
```

Finalmente, guradamos la estructura LAN del sujeto.

```
1 save LAN LAN
```

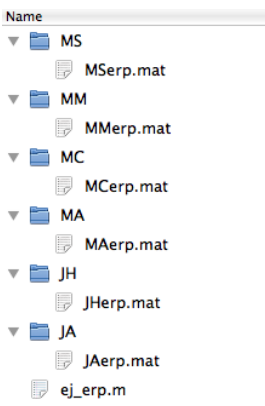
## 3.2 Potenciales relacionados a eventos

Para realizar potenciales relacionados a evento se requiere que todas las matrices individuales LAN estén guardadas en archivos individuales `.mat`. Además el orden de los archivos y carpetas tienen que presentar cierta lógica para poder dar esta descripción a las funciones. La función principal de esta etapa es `erp_stata.m` y la función de visualización `erp_plot.m`.

Lo primero es realizar las configuraciones en la estructura `cfg` para la función `erp_stata.m`, por ejemplo la siguiente:

```
1 % creando estructura de configuracion
2 cfg=[];
3 cfg.subject = [ ... % ID de cada sujeto
4               {'JA', 'JH', 'MA', 'MC', 'MM', 'MS'},...
5               ]; %
6 cfg.laplace=false; % opcion para realizar transformada de laplace
7 %cfg.H = H; % opciones necesarias para la transformacion
8 %cfg.G=G; % de laplace ver: LAN_LAPLACE.m
9 %cfg.chanlocs = chanlocs;
10 cfg.comp = [3 4]; % indice de las condiciones a comparar
11 % en este caso 3 = rechazos esperados
12 % 4 = rechazos inesperados
13 cfg.alpha = 0.05; % alfa para el primer test no parametrico
14 cfg.s='d'; % relacion entre las muestras
15 % en este caso dependiente por ser entre los mismo sujetos
16 cfg.group = [1 1]; % grupo al que pertenecen el sujeto
17 %cfg.groupname ={'controles'} % ID del grupo
18 % no necesita especificar cuando hay un solo grupo
19 cfg.bl = [ ]; % linea de base para la normalizacion
20 cfg.stata = true; % realizar estadistica
21 cfg.mcp = true; % realizar correccion
22 cfg.mcpMt = 'CBP'; % correccion por permutacion
23 cfg.nrandom = 2000; % numero de permutaciones a realizar
24 cfg.savesub = true; % guardar ERP de cada sujeto
25 cfg.delectrode = [1 2 5 6 27 33];
26 % indice de los electrodos a eliminar del analisis
```

En esta configuración se debe especificar donde la función va a buscar los archivos `mat` de cada sujeto, y como se llama la variable dentro del archivo. Esto se realiza con dos parámetros `cfg.filename = 'nombre del archivo'` y `cfg.matname = 'nombre de la variable'`. Ambos parámetros son caracteres o string. Existen ciertos caracteres especiales como `'%S'` que la función va a reemplazar por el ID de cada sujeto, y `'%G'` que es reemplazado por el nombre del grupo de cada sujeto. A continuación un ejemplo de como organizar los archivos. En este caso cada archivo se llama como el ID de cada sujeto más el sufijo `'erp'` y cada variable se llama como el ID de cada sujeto más el sufijo `'h'`.



```
1 cfg.filename = '%S/%Serp.mat';
2 % nombre y path de los archivos .mat de cada sujeto
3 % '%S' es reemplazado por el ID de cada sujeto
4 % '%G' es reemplazado por el ID del grupo de cada sujeto
5 cfg.matname = '%Sh';
```

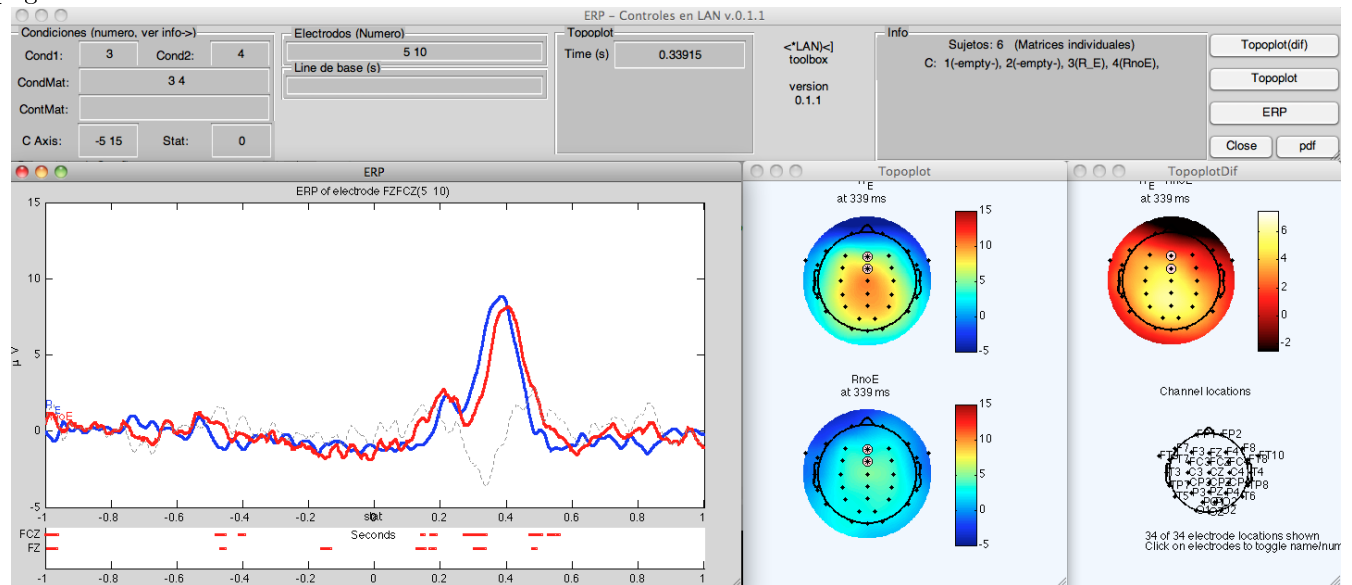
Luego se lanza la función, la que crea la estructura GLAN que contiene los ERP grupales y la estadística.

```
1 ERP = erp_stata([],cfg);
```

Luego se lanza la función `erp_plot.m` para visualizar los resultados.

```
1 erp_plot(ERP)
```

En la GUI se pueden elegir las condiciones, los electrodos y el tiempo que se quiere visualizar en los gráficos topográficos.



### 3.3 Análisis Psicofísicos

En [\[4\]](#) se han implementados algunas funciones para realizar análisis de datos psicofísicos, como aciertos y tiempos de reacción. Junto con lo anterior se esta implementado funciones para realizar análisis cruzados entre estos datos y datos electrofisiológicos.

La función de lectura es [rt\\_read.m](#), desde donde se crea la estructura RT a partir de los archivos del programa de presentación de estímulos.

## Capítulo 4

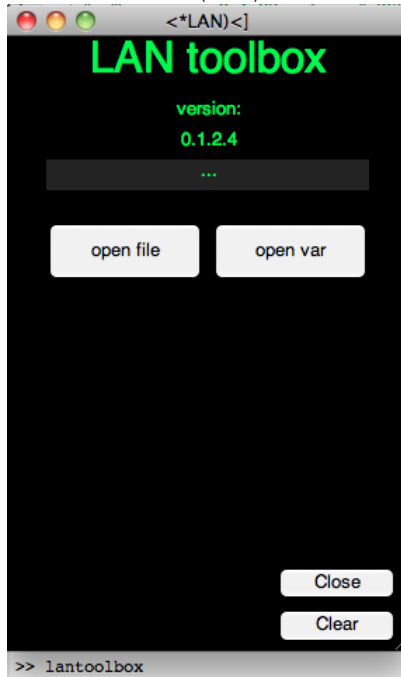
# Interfaz Gráfica

Estamos creando una interfaz gráfica con el fin de facilitar los primeros análisis, y poder facilitar la exploración de los resultados. Usando:

```
>> lantoolbox
```

o con la variable LAN o GLAN:

```
>> lantoolbox(LAN)
```





# Capítulo 5

## Módulos

### 5.1 Módulo Básico

#### **lan\_check.m** [v.1.1.1](#)

Chequea los campo de la estructura necesario, y crea los que se pueden deducir a partir de los datos. Se puede incluir en el parametro **delr=1**, elimina los ensayos no aceptados y los incorrectos según **LAN.accept** y **LAN.correct**.

```
>> LAN = lan_check(LAN)
>> LAN = lan_check(LAN,1)
```

#### **create\_lan.m** [v.1.0.0](#)

Crea una estructura LAN vacía.

```
>> LAN = craete_lan
>> LAN = craete_lan(nbchan,srate)
```

#### **merge\_lan.m** [1.1.3](#)

Une dos LAN segmentados, sino deja el primero con primera época y el segundo como segunda época. Une en forma ponderara las cartas tiempo frecuencia. Para reordenar los ensayos ocupar con estructura **cfg**:

|  |  |
|--|--|
| <code>cfg.LAN1=LAN1</code>                                 | Estructura LAN o ' <b>str</b> ' con el nombre de variable en workspace                           |
| <code>    = 'LAN1'</code>                                  |  |
| <code>cfg.LAN2=LAN2</code>                                 | idem   |
| <code>    = 'LAN2'</code>                                  |  |
| <code>cfg.sort=</code>                                     | vector del orden de épocas con 0 sin data,1 del lan1 2 del lan2, eg:<br>[0 1 1 2 1 1 1 2 2 2 2 ] |
| <code>&gt;&gt; LAN = merge_lan(LAN1, LAN2,LAN3,...)</code> |  |
| <code>&gt;&gt; LAN = merge_lan(LAN)</code>                 |  |
| <code>&gt;&gt; LAN = merge_lan(cfg)</code>                 |  |

#### **eeg2eeglab2lan.m** [v.0.0.2](#)

Importa de archivo **.eeg** de NEUROSCAN.

|  |   |
|--|---|
| <code>cfg.filename='filename'</code>                   | Nombre del archivo, con su extensión!                               |
| <code>cfg.where='path'</code>                          | Directorio donde está el archivo.                                   |
| <code>cfg.delbad=false</code>                          | elimina los ensayos malo, o solo los marca como tales (si es =true) |
| <code>&gt;&gt; LAN = eeg2eeglab2lan('filename')</code> |   |
| <code>&gt;&gt; LAN = eeg2eeglab2lan(cfg)</code>        |   |

### **eeglab2lan.m** v.0.0.7

Pasa del formato de EEGLAB a LAN. si nonQ=0 no realiza preguntas si faltan datos como en nombre del sujeto, condición y grupo. Divide como épocas segmentos cortados en EEGLAB marcados con `event.type = 'boundary'`.

```
>> LAN = eeglab2lan(EEG,nonQ)
```

### **erplab2lan.m** v.0.1

Carga archivo .erp (ERPLAB) con su correspondiente archivo .set (EEGLAB), y los transforma a una estructura LAN.

|                                       |  |
|---------------------------------------|--|
| <code>erpfile=['filename.erp']</code> | Nombre (y ruta) del archivo ERPLAB   |
| <code>setfile=['filename.set']</code> | Nombre (y ruta) del archivo EEGLAB, si no se menciona busca el guardado en .erp, si no despliega un menú |

```
>> LAN = erplab2lan('filename.erp','filename.set')
>> LAN = eeglab2lan('filename.erp')
```

### **ploteeglab.m** v.0.0.1

Scroll plot usando EEGLAB. **marca** opción con el código del evento que se desea numerada para distinguir épocas. Esta función aun requiere que este EEGLAB en el path. Usar de preferencia **prepro\_plot.m**.

```
>> ploteeglab(LAN, marca)
```

### **resample\_lan.m** v.0.0.02

Realiza resample de los datos, en **newsrate** se configura la nueva tasa de muestreo.

```
>> LAN = resample_lan(LAN,newsrate)
```

### **electrode\_lan.m** v.0.9.5

Elimina electrodo y los guarda en LAN.delete.data. Arregla estructura LAN.chanlocs y matrix LAN.chanlocs.electrodeamat cuando es posible. El segundo argumento puede ser el índice del electrodo ( [ 1,2,5,6]), el label del electrodo ('VEOR', 'VEOL', 'HEOU', 'HEUL'), o el tipo de electrodo ('EOG'). Para que estos dos ultimos parametros funciones, deben estar adecuadamente puestos en la estructura **chanlocs**.

```
>> LAN = electrode_lan(LAN, [32,33,34])
>> LAN = electrode_lan(LAN, 'EOG')
```

### **add\_field.m** **reduce\_field\_lan.m** **only\_field\_lan.m** v.1.0.5

Agregan, eliminan o solo dejan un campo en la estructura LAN.

```
>> LAN = add_field(LAN,'freq.cfg.step = 1');
>> LAN = reduce_field_lan(LAN,'freq');
```

### **label2idx\_elec.m** v.0.0.1

Busca los índices de los electrodos dado por la etiquetas: label='Cz' o label={'Cz', 'Oz'}.

```
>> idx = label2idx_electrode(chanlocs,label);
>> idx = label2idx_electrode(LAN.chanlocs,{'AP2', 'AP1'});
```

### **filt\_eeglab.m**

Filtro según script de EEGLAB (requiere este en el path).

```
>> LAN = filt_eeglab(LAN,low,hi)
```

## 5.2 Módulo de Segmentación

### **lan\_latency.m** v.0.0.1

Genera LAN.time a partir de eventos señalados, y segmenta. (adaptación de la antigua función latency.m, sin más

actualizaciones)

|  |   |
|--|---|
| <code>cfg.res=</code>                              | si existe el código de evento marcador único de referencia distinto para las épocas, puede ser más de una |
| <code>cfg.parametres=[ iniciales ; finales]</code> | matriz con el número de los eventos   |
| <code>cfg.epoch=true</code>                        | segmenta el dato continuo   |
| <code>cfg.segmente=true</code>                     | divide las condiciones en diferentes LAN estructuras  |

```
>> [ LAN latency] = lan_latency(LAN, cfg)
```

#### **del\_time**

Borra (o separa si `guardar = 1`) épocas de la matriz `LAN.time`, y genera nueva estructura LAN con éstas. Requiere la posición de las épocas a borrar o separar.

```
>> [ LAN LAN2] = del_time(LAN, pos, guardar)
>> [ LAN LAN2] = del_time(LAN, [1 2 6 10], 1)
```

#### **del\_epo.m**

Borra épocas, y las guarda en `LAN.delete.data_epoch`.

```
>> LAN = del_epo(LAN, [1 2 6 10])
```

#### **mod\_time**

Modifica la matriz `LAN.time`.

```
>> LAN = mod_time(LAN, time, mod, fix)
>> LAN = mod_time(LAN, [-0.5], [2], 1)
```

#### **epoch\_lan.m**

Crea épocas a partir de data continua, requiere que exista `LAN.epoch` con las 3 celdas, o input.

```
>> time = [ {[initial time x trials]} {[final time x trials]} {[ points0 x trials]} ]
>> LAN = epochs_lan(LAN)
>> LAN = epochs_lan(LAN, time, 2)
```

#### **eval\_time\_epoch.m**

Evalúa la duración de las épocas y las compara con un limite de referencia ('ref') en segundos, siendo verdadero si es mayor o igual. Este resultado lógico se guarda en `LAN.cfg.time.eval`, y 'ref' en `LAN.cfg.par_time`. Si 'cut' = 1, re-segmenta épocas mayores que el parámetro en duración de 'ref', a partir del final. Las menores las elimina y las guarda en `LAN.delete`.

```
>> LAN = eval_time_epoch(LAN, ref, cut)
```

```
>> LAN = eval_time_epoch(LAN, 3.0, 1)
```

#### **dividir\_eeglab2lan.m**

Divide una estructura LAN o EEGLAB en LAN con celdas, según el número de épocas por cada celda señalada. Requiere datos segmentados en matriz de 3D. (Para matriz de 2D usar `dividir_eeglab2lan_old`)

```
>> LAN = dividir_eeglab2lan(EEG, [75 7], [{ 'controles'}, { 'objetos'}])
```

#### **dividir\_eeglab2lan\_old.m**

Pasa de EEGLAB a LAN dividido por condiciones. En el EEG debe existir el campo `.time` para marcar los tiempos si es un EEG continuo.

```
>> [LAN] = dividir_eeg2lan(EEG, pr, sg, op, filename1, filename2)
```

## 5.3 Módulo de Preprocesamiento

**vol\_thr\_lan.m** v.0.0.01

Busca canales, ensayos malos mediante un umbral de voltage, y los etiqueta en la estructura TAG. parámetros son:

|                                 |  |
|---------------------------------|--|
| <code>thr=[uV]</code>           | Umbral en microvols                        |
| <code>tagname=['labels']</code> | Etiqueta, por defecto <code>'bad:V'</code> |

```
>> LAN = vol_thr_lan(LAN,thr,'tagname')
```

#### **fftamp\_thr\_lan.m v.0.0.01**

Busca canales y épocas donde la amplitud por frecuencia sobrepasa un umbral. Ver [fourierp.m](#)

|                                |   |
|--------------------------------|---|
| <code>thr=[1 0.2]</code>       | Umbral para detectar épocas x canales malos. Vector de la forma <code>[STD p]</code> , donde STD son las cantidad de desviaciones estandares y p la proporción de frecuencias que se sobrepasan el umbral para marcar en ensayo como malo |
| <code>tagname='tagname'</code> | string con el nombre (label) del TAG para marcar las épocas. Por defecto <code>'bad:A'</code> .   |
| <code>frange=[f1 f2]</code>    | rango de frecuencia a explorar.   |
| <code>method='mt'</code>       | String con el método de descomposición tiempo frecuencia. Si <code>'mt'</code> se realiza multitapers, sino se realiza fourier simple.  |
| <code>tapers=[TW K]</code>     | Definición de tapers para <code>'mt'</code> , donde TW es el producto timepo-ancho de banda y K es el número de tapers ( $k \leq 2 \cdot TW - 1$ )  |
| <code>cat=1</code>             | Si es 1, se considera la desviación estándar de todos las condiciones, sino se calcula por condición.   |
| <code>nch=[elec]</code>        | Electrodos a evaluar, por defecto todos ( <code>'all'</code> ).   |

```
>> LAN = fftamp_thr_lan(LAN,cfg)
```

**lan\_interp.m v.0.0.01** Realiza interpolación de canales malos, indicados individualmente, o a través de etiquetas de la estructura TAG (LAN.tag; ver capítulo 2).

|   |   |
|---|---|
| <code>type='label'</code>                 | Etiqueta de los canales a interpolar.                 |
| <code>bad_elec=[elec1 , elec2, ..]</code> | Vector de los canales a interpolar.                   |
| <code>bad_trial=[t1 , t2, ..]</code>      | Vector de los ensayos a interpolar.                   |
| <code>method='method'</code>              | Método a utilizar. Por defecto <code>'invdist'</code> |

```
>> LAN = lan_interp(LAN,cfg)
```

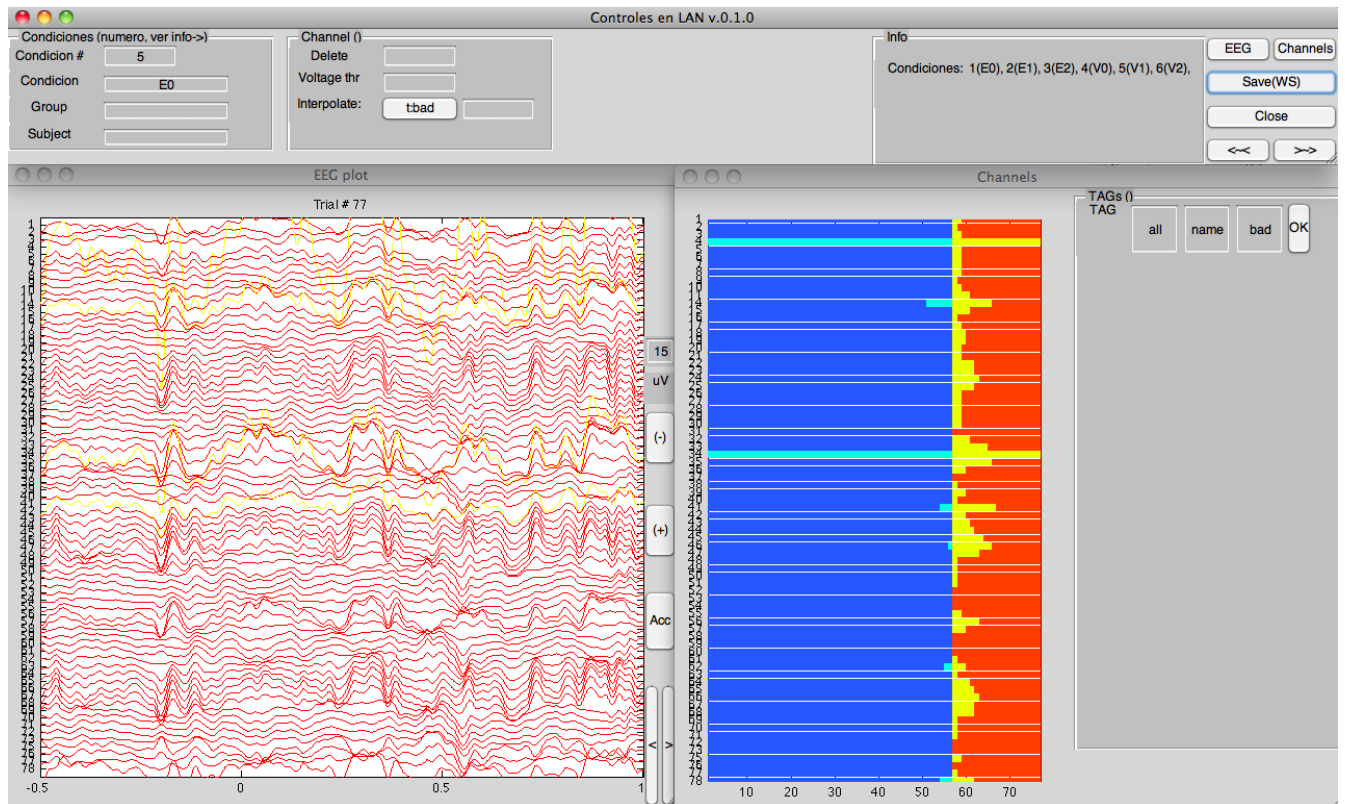
**lan\_laplace.m v.0.0.1** Realiza transformación esferica usando lapalceanos, segun los algoritmo descritos por () y implementados en CDStoobox (). Más información en <http://psychophysiology.cpmc.columbia.edu/Software/CSDtoolbox>

|                                    |   |
|------------------------------------|---|
| <code>cfg.header=10</code>         | Radio de la esfera, por defecto 10.   |
| <code>cfg.lambda=0.00001</code>    | parametro de suavizado.   |
| <code>cfg.H=H</code>               | Matrix de tranformación H precalculado, si no se encuentra la calcula para el montaje de electrodos tipo <code>'EEG'</code> . |
| <code>cfg.H=G</code>               | Matrix de tranformación G precalculado, si no se encuentra la calcula para el montaje de electrodos tipo <code>'EEG'</code> . |
| <code>cfg.chanlocs=chanlocs</code> | Estructura de localización de los electrodos para cacular las matrices H y G.   |

```
>> LAN = lan_laplace(LAN,cfg)
```

**prepro\_plot.m v.0.0.12** GUI para realizar preprocesamiento de datos previamente segmentados o visualización de datos continuos. Se puede aplicar umbral de voltaje, etiquetar canales-ensayos e interpolar.

```
>> prepro(LAN)
```



## 5.4 Módulo de Potenciales Evocados

### **erp\_lan.m v.0.0.4**

Realiza potencial evocado promedio y lo gráfica (`ifplot=1`) por electrodo o grupos de electrodos definidos en vector `roi`, la linea de base se define en `bl = [s1 s2]`. `hh` matriz lógica de la estadística.

```
erp_mean = erp_lan(LAN,roi,bl,ifplot,hh)
```

### **erp\_stata.m v.0.1.3**

Realiza estadística no paramétrica a los grupos de estructuras LAN con el `.data` con ERP (segmentados). Realiza un primer test de sensibilidad no parametrico (Wilcoxon, Mayor información en [nonparametric.m](#)), luego realiza correcciones por comparaciones múltiples a través del método de permutaciones basado en clusters (ver [Maris & Oostenveld, 2007](#)).

. Se requiere parámetro `cfg.` o `GLAN.erp.cfg.` con:

```
subject={'str' , 'str' , .. }]
```

Nombres de los sujetos para identificar archivos `.mat`. Si en el experimentos hay más de un grupo (caso y controles), poner cada una en una celda dentro de `subject = { {casos} , ... {controles} }`.

```
cond=[n1 n2];
```

índice de las condiciones a comparar, si compara grupos en las mismas condicion repirtirla(`cond = [1,1]`).

|  |  |
|--|--|
| <code>matdif=[1 -1 2 -2]</code>                                    | matriz de diferencias, del mismo tamaño que <code>cfg.comp</code> . donde los 1 corresponden al primer grupo, -1 el que se va arrestas a 1, los mismo para 2 y -2.   |
| <code>group=[n1 n2];</code>  | Grupos de individuos a comparar (indice dentro de <code>cfg.subject</code> ).  |
| <code>groupname={'casos', 'controles' };</code>                    | Nombre de los grupos de individuos a comparar, para identificar archivos   |
| <code>alpha=0.05;</code>   |  |
| <code>s='d';</code> OR <code>'i'</code>                            | relación entre las muestras.   |
| <code>bl=[ s1 s2];</code>  | Límites de la línea de base en segundos.   |
| <code>mcp=1</code> or <code>0</code>                               | Para realizar corrección por comparaciones múltiples, por test de permutaciones.   |
| <code>nrandom=n</code>   | Número de randomizaciones para el test de permutaciones. Por defecto 100.  |
| <code>savesub=0</code>   | Guarda erp por sujetos, por defecto=0;   |
| <code>stata=1</code>   | Realiza estadística, por defecto=1;  |
| <code>srata=1000</code>  | Chequea la tasa de muestreo de las estructuras LAN, y re-muestra las necesarias. Ver <a href="#">resample_lan.m</a>  |
| <code>laplace=false</code>   | Si es true, realiza transformación de laplace con parametros por defecto, usando CSD toolbox, ver <a href="#">lan_laplace.m</a> .  |
| <code>reref=[4 75]</code>  | Rreferencia a los indices o etiquetas de los canales. Es preferible rereferenciar usando el modulo de prerprocesamiento. Precaución al usar <code>delelectrode</code> , chequear los índices de los electrodos |
| <code>mcpMt='CBP'</code>   | Método de comparaciones. defecto:'CBP' cluter-based permutation;   |
| <code>delelectrode=[elec<sub>1</sub> elec<sub>n</sub> ... ]</code> | 'CRP' cluster-based randomization.<br>elimina electrodos del análisis.   |
| <code>filename='str'</code>  | Especifica el nombre del archivo mat, ejemplo <code>./%S/%S</code> . Por defecto es <code>./%S.mat</code>  |
| <code>matname='str'</code>   | Especifica el nombre de la matriz. Por defecto es <code>%S</code> .  |
| <code>= '%S'</code>  | Señala el nombre del sujeto en <code>cfg.subject</code> .  |
| <code>= '%G'</code>  | Señala el nombre del grupo en <code>cfg.groupname</code> .   |

`GLAN = erp_stata(GLAN, cfg)`

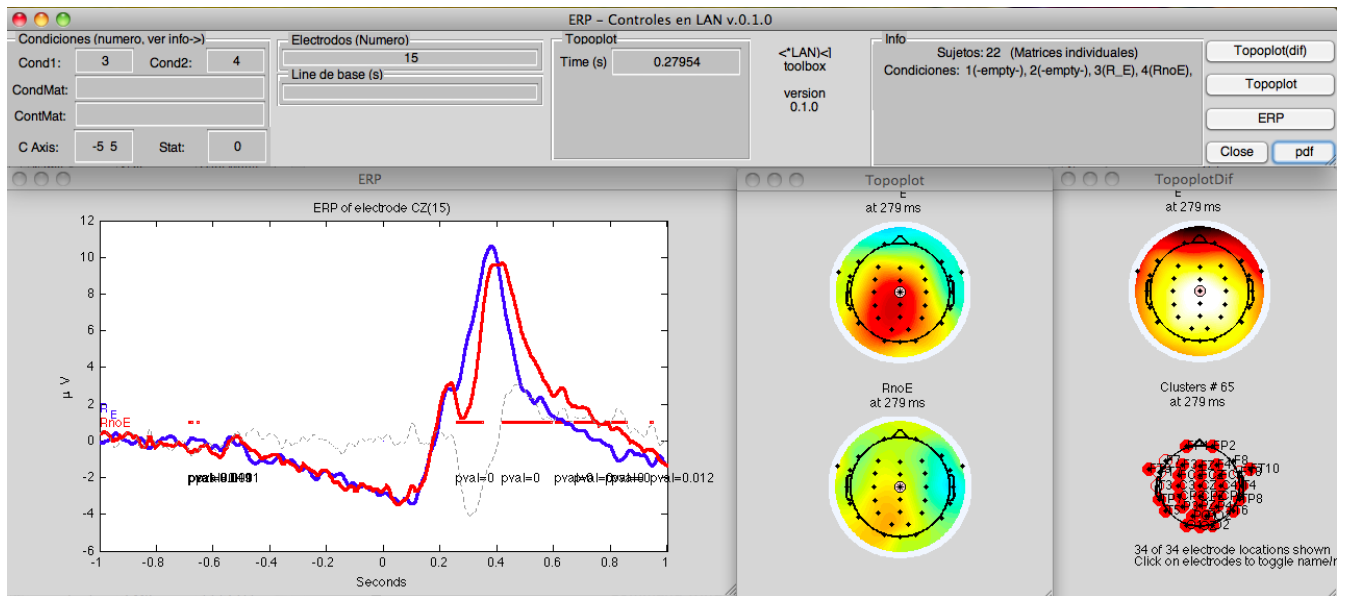
`GLAN = erp_stata(GLAN)`

`GLAN = erp_stata([], cfg)`

#### **erp\_plot.m** v.0.0.4

GUI para graficar potenciales, realizar estadística, gráficos topográficos, y estadística. Versión actual solo estadística de muestras dependientes, con corrección por comparaciones múltiples aproximada. En electrodos se pone el número del electrodo o roi de electrodos, o 'all' para graficar todos en un arreglo. Botón pdf exporta graficos visibles a pdf (requiere ps2pdf Latex), si no solo en ps. Esta función esta basado en [erp\\_glan.m](#) (ver `help(erp_glan)`)

`erp_plot(GLAN)`



#### erp\_glan.m v.0.1.7

Remplazada por [erp\\_plot.m](#), sin más actualizaciones. Realiza plot interactivo de erp de estructura GLAN por electrodo definido en vector *roi*, en *comp* se define el índice de la comparación a graficar. Si existe GLAN.chanlocs se habilita el gráfico topográfico de las condiciones.

```
erp_lan(GLAN,roi,comp)
```

## 5.5 Módulo de Análisis de Frecuencias

#### fourierp.m v.0.0.1

Calcula espectro de fourier inducido por canal. Se guarda en LAN.freq.fourierp .

|                                      |  |
|--------------------------------------|--|
| <code>chn=[elec1,elec2, ... ]</code> | canales a realizar fourier. Por defecto 'all'. |
| <code>= 'all'</code>                 |  |
| <code>ifplot=1</code>                | Realiza plot del espectro.                     |

```
>> LAN = fourierp(LAN)
>> LAN = fourier(LAN,cfg)
```

#### plot\_fourierp v.0.0.1

Realiza gráfico de frecuencias mostrando la media, el limite superior (una desviación estandar) el ensayo cuando se especifica *t* de uno o varios canales *nch*.

```
>> fourier(LAN,nch,t)
>> fourier(LAN,'Cz')
>> fourier(LAN,[10 11],3)
```

#### freq\_lan.m v.0.1.3

Realiza descomposición en tiempo y frecuencia. Parámetros se definen en LAN.freq.cfg o en parámetro *cfg*, siendo :

|                          |   |
|--------------------------|---|
| <code>type='str';</code> | el algoritmo a usar. 'Hilbert' , 'Fourier' , 'MultiTaper', 'Morlet' hasta el momento. |
|--------------------------|---|

|                              |   |
|------------------------------|---|
| rang=[f1 f2]                 | rango de frecuencias a calcular (e.g. [5 100]).   |
| bin=[r];                     | la resolución por frecuencia.   |
| win=[n];                     | ventana de hamming en puntos para Fourier o números de ciclos para Wavelet Morlet.  |
| fwin=[r];                    | ventana temporal por frecuencias para 'MultiTaper'. Si el largo del vector es 1, se asume como número de ciclos en cada frecuencia (ventanas variables), Si largo del vector es igual al numero de frecuencia a calcular, ventana temporal por frecuencias en segundos. |
| step=[n]                     | pasos entre ventanas en puntos para Fourier.  |
| tapsmofrq=[n]                | ancho de smooth de frecuencias, por frecuencias para 'MultiTaper'. Si largo del vector es 1, se concidera (n*Hz)  |
| resample=[n n]               | fracción para resamplear la señal para Hilbert. ([1 8], re-muestrea a 1/8).   |
| keeptrials='yes','no','file' | Opción para guardar la descomposición tiempo-frecuencia por cada ensayo. Útil para realizar estadística y modelos. opción 'file' los guarda en archivos separados para mejor manejo de memoria. Es preferible usar esta última opción.                                  |

Habilitada la opción de interfaz GUI [pregunta\\_lan.m](#) para configurar `cfg`. si éste no se encuentra.

```
>> LAN = freq_lan(LAN)
>> LAN = freq_lan(LAN, cfg)
```

#### **timefreq\_stata.m v.0.0.7**

Realiza estadística no peramétrica a los grupos de estructuras LAN con el `.freq` con cartas tiempo-frecuencias . Mayor información en [nonparametric.m](#). Se requiere parámetro `cfg`. o `GLAN.timefreq.cfg`. con:

|  |  |
|--|--|
| subject=[{ 'str' },{ 'str' } ... ]           | Nombres de los archivos .mat de los sujetos (o en GLAN.subject ). Si los sujetos están asu vez agrupados en grupos en celdas, se entiende que estos son los diferentes grupos a comparar. por ejemplo, grupo control <code>cfg.subject{1} = {'s1','s2',...}</code> ; y los casos <code>cfg.subject{2} = {'sn1','sn2',...}</code> ; |
| comp=[n1 n2 ..];                             | índice de las condiciones a comparar.  |
| group=[g1 g2 ..];                            | Si existe grupos de individuos son los índices de estos grupos (número de columna donde están en <code>cfg.subject</code> )  |
| matdif=[1 -1 2 -2 ..];                       | Si existe grupos de individuos son los índices de estos grupos (número de columna donde están en <code>cfg.subject</code> )  |
| matdif_transform={ 'none', 'log', 'log10' }; | Transformación a aplicar antes de hacer el contraste, por defecto ninguna 'none'   |
| alpha=0.05;                                  | alfa   |
| s='d'; OR = 'i'                              | relación entre las muestras.   |
| bl=[ s1 s2];                                 | Limites de la linea de base en segundos.   |
| norma='mdB';                                 | Tipo de normalización, ver <a href="#">normal_z.m</a> .  |
| mcp=1 or 0                                   | Para realizar corrección por comparaciones múltiples, por test de permutaciones(ver <a href="#">Maris &amp; Oostenveld, 2007</a> ).  |
| nrandom=n                                    | Número de permutaciones para el test de permutaciones  |
| delelectrode=[n1 n2 ... ]                    | Elimina electrodos del análisis  |
| savesub=0                                    | Guarda carta T-F por sujetos, por defecto=0.   |
| stata=1                                      | Realiza estadística, por defecto=1.  |



```
GLAN = timefreq_stata(GLAN,cfg)
GLAN = timefreq_stata(GLAN)
GLAN = timefreq_stata([],cfg)
```

#### freq\_plot.m v.0.0.1

Realiza grafico de cartas tiempo-frecuencias de estructuras lan. configuración cfg.:

b1 = [s1 s2] Limite en segundos de la linea de base para normalización.  
nor = 'z' or 'mdb' Tipo de normalización.

```
freq_plot_glan(LAN,cfg)
```

#### freq\_plot\_glan.m v.0.0.1

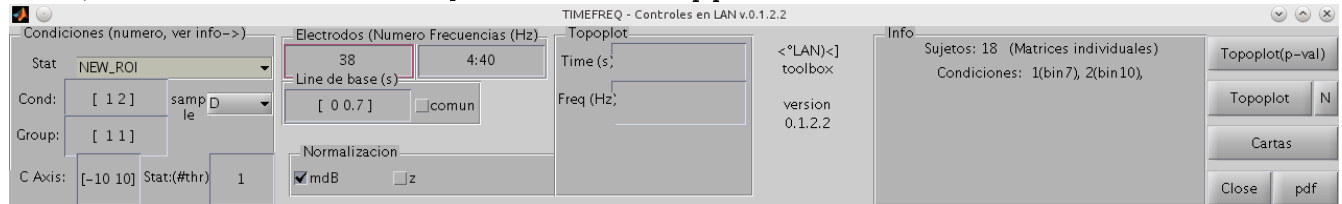
Realiza grafico de cartas tiempo-frecuencias de estructuras grupales glan. configuración cfg.:

b1 = [s1 s2] Limite en segundos de la linea de base para normalización.  
nor = 'z' or 'a' Tipo de normalización.  
comp= n Comparación a graficar, anula parámetro cond.  
cond= n Condición a graficar si no existe comp.  
hh= 0/1 grafica solo áreas estadísticamente significativas.

```
freq_plot_glan(GLAN,cfg)
```

#### timefreq\_plot.m v.0.1.1

GUI para graficar cartas tiempo frecuencia , estadística, gráficos topográficos. Realiza estadística de ROI de electrodos, usando corrección de cluster aproximada. **timefreq\_plot(GLAN)**



#### filter\_hilbert.m

Esta función no esta adaptada directamente para estructura LAN. Realiza filtro de hilber, que consiste en filtro pasa-banda, más la trasformación de Hilbert. Da como resultado la señal analítica de la señal centrada en la frecuencia de interés, de la forma:

$$s_a = s(t) + iH(s(t)) \quad (5.1)$$

Siendo  $H(s(t))$  la transformada de Hilbert de la señal  $s$ , obtenida de con la formula:

$$(Hs)(x) = i \int_{-\infty}^{+\infty} F(s)(u) \cdot \text{sgn}(u) \cdot e^{2\pi i u x} du \quad (5.2)$$

La señal analítica usando el principio de Phasor, para poder linearizar la combinación de ondas, dejando solo la amplitud (estática) y la fase (Sin información de frecuencia).

|         |  |
|---------|--|
| x=      | Senal, el tiempo en la priemra diemnsión.  |
| Fs=     | Frecuencia de muestre.   |
| Fp1=    | Frecuencia inferior del filtro mpasa banda   |
| Fp2=    | Frecuencia superior del filtro mpasa banda   |
| norbin= | Opción de realizar un blanqueo (Whitening) de la señal. Se indica el Por defecto en 0, sin blanqueo. |

```
>> xend = filter_hilbert(x,Fs,Fp1,Fp2,norbin)
```

## 5.6 Modulo de Redes

En este modulo se encuentran funciones para relaizar diversosos anailis de covarianza o sincronia entre pares de electrodos, así como análisis de redes.

**lan\_syn\_net.m** v.0.0.3.4

Función para realizar sincronía entre pares de electrodos.

|  |  |
|--|--|
| <code>cfg.freq_method='Method'</code>  | Método para calcular la fase y la amplitud instantanea. Existen habilitados: <code>'Wavelet'</code>  |
| <code>cfg.algorithm='Algoritmo'</code> | Algoritmo para calcular la sincronía de fase. Existen habilitados: <code>'PLV'</code> phase locking value, (Lachaux et al., 1999)  |
| <code>cfg.across='trials'</code>       | Realiza el calculo de sincronía a través de los ensayos: <code>'trials'</code> o a través del tiempo : <code>'time'</code>   |
| <code>cfg.step=[ n ]</code>            | Paso para el calculo de sincronia a travez de los ensayos. Ventana para el calculo de la sincronía por tiempo.   |
| <code>cfg.frange=[f1 f2]</code>        | Rango de frecuencias para analizar la sincronía.   |
| <code>cfg.ResHz=[ n ]</code>           | Resolución en bin por Hz de frecuencias  |
| <code>cfg.ncomponents=</code>          | Número de componentes si se realiza el análisis de sincronía en ele espacio de la fuentes. Si los componentes son más que uno por vertex, se reducen a uno usando en cross-espectro de la parte real, para no alterar la fase de la fuentes subyacentes. |
| <code>cfg.ifdiv=</code>                | ....   |
| <code>cfg.ndiv=</code>                 | ...  |

```
>> LAN = lan_syn_net(LAN, cfg)
```

**syn\_hilbert\_lan.m**

Realiza cálculo de diferencia de fase y sincronía entre electrodos, a través de transformada de Hilbert en la frecuencia especificada. Por defecto ocupa ventana de análisis de 6 ciclos.

$$PLV = Syn(t) = \sqrt{\langle Cos(\phi_1(t') - \phi_2(t')) \rangle_{T(t)}^2 + \langle Sin(\phi_1(t') - \phi_2(t')) \rangle_{T(t)}^2} \quad (5.3)$$

Donde T(t) es la ventada de tiempo, centrada t, donde  $t' \in T$ . Siendo el largo de T dependiente de  $\omega$ .

Además presenta opciones en formato .cfg (símil al usado en Fieltrip):

- Estadística
  - Si se encuentra en la estructura LAN.phase.cfg el campo `.stata = 'boot'`, se realiza un una muestra bootstrap de la fase, con `n = LAN.phase.cfg.nboot`, 100 por defecto. Entrega como resultado matriz con el valor  $p$  de la distribución bootstrap de una cola, en celda `{epocas[pares x tiempo]}` (en `LAN.phase.p_val`).
  - Si se encuentra en la estructura LAN.phase.cfg el campo `.stata = 'suro'`, se realiza un surogate de la fase con `n = LAN.phase.cfg.nsuro`, 100 por defecto. Entrega como resultado matriz con el valor  $p$  de la distribución bootstrap de una cola, en celda `{epocas[pares x tiempo]}` (en `LAN.phase.p_val`).
- Algoritmos
  - Sí existe en LAN.phase.cfg el campo `.pli = 1`, realiza el indice de diferencia de fase (Phase lag index, (ver: Stam et al., 2007)), según la siguiente formula:

$$PLI = Syn(t) = |\langle Sgn(\Delta\phi(t_k)) \rangle| \quad (5.4)$$

```
[LAN] = syn_hilbert_lan(LAN,frec,win_cycles);
```

#### **net\_syn\_lan.m** ;

Calcula matriz de sincronía en periodos de tiempo definidos, sin dinámica temporal para análisis de redes. Parámetro de definen en **par**, de la siguiente forma:

```
par{1} = phase.cfg.net_freq = f
— frecuencia para calculo de sincronía.
par{2} = phase.cfg.net_time = [t1 t2]
— rango temporal para en cálculo de sincronía.
par{3} = phase.cfg.net_algo = 'PLI' ó 'PLV' ó 'BOTH'
— algoritmo para el cálculo de las matrices de sincronía.
par{4} = phase.cfg.net_stata = 'SURO' ó 'BOOT'
— métodos de permutación para cálculo de significancia.
par{5} = phase.cfg.net_permute = n
— número de permutaciones calculadas.
LAN = net_syn_lan(LAN)
LAN = net_syn_lan(LAN,par)
```

#### **plot\_syncro** var **plot\_syncro\_s** var **plot\_syncro\_g**

Diagrama sincronía entre electrodos, por sujeto o promedios grupales.

#### **freq\_inter\_band.m** v 0.0.2a

Estudios interbandas, en fase experimental. Ordenamiento de amplitud de bandas de frecuencias en relación a otras, por electrodos. Parámetros en cell-array *par* de la forma siguiente:

```
.freq.cfg.inter_fr1 = par{1} = f1:f2
- primera banda de frecuencias.
.freq.cfg.inter_fr1 = par{2} = f3:f4
- segunda banda de frecuencia.
.freq.cfg.intert_fr1_w = par{3}: 'phase' , 'amplitud', 'both'
- que se va a compara de la primera banda, o banda base
.freq.cfg.what_fr2_w = par{4}: 'amplitud'
- que se va a compara de la segunda banda, (solo amplitud por el momento).
.freq.cfg.inter_time = par{5} = [t1 t2]
- tiempo para el calculo inter-frecuencia es segundos.
```

Si no se define el **cfg** ni el **par**, se despliega GUI para configuración. Por defecto busca primero los valores de amplitud si ya se han calculado en **LAN.freq.ind**, si no existe este campo, calcula la fase y la envolvente mediante la trasformada de hilbert, esto resulta más lento.

```
LAN = freq_inter_band(LAN,par)
LAN = freq_inter_band(LAN)
```

## 5.7 Módulo de Estadística

#### **lan\_nonparametric.m** v.0.1.0

Función para realizar estadística no paramétrica, en vías de remplazar definitivamente a **nonpametric.m**. data son los datos a comparara, en cell-array, en cada celda las condiciones a compara, última dimensión son los sujetos.

|                      |   |
|----------------------|---|
| <b>method='rank'</b> | Método,solo habilitado hasta el momento <b>'rank'</b> . |
| <b>paired=true</b>   | Relación entre las muestras                             |

```
>> [pval stats] = lan_nonparametric(data,cfg)
```

#### **nonparametric.m v.0.0.3**

Compara matrices (a b) de tres o cuatro dimensiones, suponiendo que los sujetos están en la ultima dimensión. **alpha** es nivel de significancia, **m** es la relación entre las muestras: 'i' independientes y 'd' dependiente. **means** = 1 se quiere calcular solo la significancia del promedio del área de a y b.

```
[pval, hh, stat] = nonparametric(a,b,alpha,m,means)
```

#### **parametric.m v.0.0.1**

Compara matrices (a b) de tres o cuatro dimensiones, suponiendo que los sujetos están en la ultima dimensión. **alpha** es nivel de significancia, **m** es la relación entre las muestras: 'i' independientes y 'd' dependiente. **means** = 1 se quiere calcular solo la significancia del promedio del área de a y b.

```
[pval, hh, stat] = parametric(a,b,alpha,m,means)
```

#### **FDRlan.m v.0.0.1**

Realiza corrección por comparaciones multiples, usando tasa de falsas alarma (false discovery rate) de los valores p dados por columna. Da como resultado el p umbral

```
pv = FDRlan(p,alpha)
```

#### **freq\_bootstraping.m v.0.0.1**

Realiza test de permutación bootstrapping en matrices de tres dimensiones, a través de la tercera dimensión.

**freq** = [freq x elect x tiempo] Matriz tiempo-frecuencia.

**b1** = [n n] Limites de la linea de base en puntos, por defecto [].

**alpha** = 0.05 Nivel de significancia, por defecto 0.05

```
[hh] = freq_bootstraping(freq,b1,alpha)
```

#### **cor\_freq\_lan.m**

Realiza gráfico y estadística (Wilcoxon de suma de los rangos, muestras independientes) de los  $\rho$  de correlaciones entre frecuencias por trials hechas por **freq\_lan.m**.

```
[ bs,mbs,p,hc] = cor_freq_lan(LAN)
```

#### **freq\_correlation.m**

Realiza correlaciones entre frecuencias por trial . Requiere en LAN.freq.cfg .corfr = [l] (1= si, 0 = no); .corfr1 = {f1:f2} {} Rangos de frecuencia a correlacionar con corfr2; .corfr2 = {f1:f2} {} Rangos de frecuencia a correlacionar con corfr1; .cortime = [t1 t2]; Tiempo a correlaciona en segundos.

```
LAN = freq_correlation(LAN)
```

## 5.8 Modulo de Modelos y Tiempos de Reacción

Serie de funciones en implementacion para realizar análisis de tiempos de reacción, correlaciones y modelos. Por el momento para el funcionamiento de los modelos se requiere que el software R este instalado y el script littler (Ver sección 2.3).

#### **rt\_read.m v.0.0.8**

Lee archivos de texto (.log , .ev2) de la presentación de estímulos para calcular tiempos de reacción.

```
cfg.type='formato'
```

Formato del archivo a leer. Puede ser:

'presentation': para archivos de presentation .log

'neuroscan': para archivos de neuroscan .ev2

|  |  |
|--|--|
| <code>cfg.filename='nombredearchivo.ext'</code>                  |  |
| <code>cfg.delim=[ est, resp;<br/>          =est2, resp2 ]</code> | matriz con estímulos y respuesta(s) correcta(s) para cada estímulo, -99 se ocupa para cuadrar las matrices.<br>Si en el experimento no hay respuestas correctas, usar las siguientes dos opciones: |
| <code>cfg.est=[est1,est2,..]</code>                              | Estímulos.   |
| <code>cfg.resp=[resp1, resp2, ..]</code>                         | Posibles respuestas.   |
| <code>cfg.rw=[s]</code>  | ventana de respuesta en <i>s</i> o <i>ms</i> según <code>cfg.unit</code>   |
| <code>cfg.stop=[event]</code>                                    | evento que marca fin del tiempo de respuesta, puede ponerse aquí la respuesta errónea.   |
| <code>cfg.unit='ms'</code>                                       | unidades empleadas para <code>cfg</code> y resultados ' <i>s</i> ', ' <i>ms</i> '  |
| <code>cfg.iflbc=0</code>   | Iniciar las latencias en cero (desde primer estímulo).   |
| <code>cfg.miss=1</code>  | Dejar respuestas omitidas separadas de estímulos contestados, usando <code>miss2rt.m</code> . Si no, deja respuestas omitidas como -99.  |
| <code>cfg.invert=0</code>  | Busca estímulos en relación a la respuesta y no al revés.  |

Ciertos formatos requieren configuraciones adicionales, como '*Neuroscan*' que requiere los siguientes parámetros:

|                             |   |
|-----------------------------|---|
| <code>cfg.srate=1000</code> | Tasa de muestreo del archivo donde se extrajeron los eventos.   |
| <code>cfg.ifr=false</code>  | Dependiendo del diseño del experimento, las respuestas del sujeto se guardan en forma diferente a los estímulos. Si este es el caso, se debe dejar este parámetro como: <code>true</code> . |

```
COR.RT = rt_read(cfg)
```

#### **miss2rt.m** v.0.0.1

Transforma eventos omitidos (missed) de estructuras RT a `rt=-99`.

#### **rt\_merge.m** v.0.0.3 y **rt\_merge\_block.m**

Une estructuras de tiempos de reacción RT.

#### **rt\_fixlaten** v.0.0.1

Arregla los tiempo de reacción comparando dos archivos de latencias. El primer argumento es la estructura a arreglar proveniente del programa de registro, y la segundo es la referencia que proviene del programa de estimulación.

|                               |   |
|-------------------------------|---|
| <code>cfg.f=n</code>          | Índice de la primera latencia de referencia para empezar a comparar, por defecto 1.             |
| <code>cfg.laten=</code>       | Vector con las latencias de referencia, del programa de estimulación                            |
| <code>cfg.est=</code>         | Vector con estímulos de referencia, del programa de estimulación                                |
| <code>cfg.resp=</code>        | Vector con respuestas de referencia, del programa de estimulación                               |
| <code>cfg.rt=</code>          | Vector con tiempos de reacción de referencia, del programa de estimulación                      |
| <code>cfg.RTfix=RT</code>     | Estructura RT de referencia si se quiere obviar los 4 parámetros anteriores                     |
| <code>cfg.dw_delta=100</code> | Delta tolerado para la comparación entre latencias  |
| <code>cfg.up_delta=200</code> | Delta sobre el cual clasifica aun estímulo como perdido, y usa el correspondiente de referencia |
| <code>cfg.ifplot=true</code>  | grafico de los delta, delta acumulados y los estímulos perdidos                                 |

```
>> RT = rt_fixlaten(RT, cfg)
```

```
>> RT = rt_fixlaten(RT1,RT2)
```

#### **cor\_merge.m v.0.0.1**

Une estructuras COR.

```
>> COR = cor_merge(COR1,COR2)
```

#### **cor\_add\_other.m v.0.0.1**

Añade un campo en **COR.other** con un valor determinado.

```
>> COR = cor_add_other(COR,field,date)
```

```
>> COR = cor_add_other(COR,'Subject','ID')
```

#### **cor\_stata.m v.0.0.2**

Realiza estadísticas básicas en los datos de tiempos de reacción, en la estructura COR. Requiere que en COR estén los datos de todos los sujetos del experimento, distinguidos con el ID en COR.OTHER.subject.

```
cfg.analysis={ 'all' }
```

Análisis a realizar, puede ser más de uno. Los parametros pueden ser  
'correct': realiza análisis de proporciones de las respuestas correctas por estímulo

'rt:mean': realiza análisis de las medias de los tiempos de reacción  
'rt:density': realiza análisis de densidad de los tiempos de reacción

'all': realiza todo los análisis

```
>>cor_stata(COR)
```

```
>> COR = cor_stata(COR, cfg)
```

## 5.8.1 Funciones entre R-Matlab

Este grupo de funciones requiere un sistema GNU y que este instalado en el sistema el software R y el script littler que esta se distribuye con LAN (~\rt\r\littler\).

#### **COR2tableR.m v.0.0.1**

Crea base de datos legible para R a partir de la estructura COR

```
filename='nombredearchivo.txt'
```

Nombre del archivo a escribir.

```
where='path/to/write'
```

Directorio, solo si es diferente al actual directorio de trabajo.

```
>> COR2tableR(COR, cfg)
```

#### **modelr.m v.0.0.11**

Realiza modelo con los datos en COR, usando el programa R.

```
model='rt ~ BETA'
```

Formula del modelo siguiendo la sintaxis de R.

```
command='lme'
```

Commando a usar en R, dependiendo de la librería a utilizar.

'lme': Comando para modelos mixtos usando librería nlme: Linear and Nonlinear Mixed Effects Models.

'lmer': Comando para modelos mixtos usando librería lme4: Linear mixed-effects models using S4 classes.

```
radom='1|sujeto'
```

Especificación de los efectos aleatorios del modelo.

```
cfg.electrode= [1:32]
```

Indice de los electrodos donde se aplicará el modelo.

```
conditions={'rt>50'; 'est==20' }
```

Condiciones que tiene que cumplir los datos para ser ingresados al modelo.

|  |   |
|--|---|
| <code>newvar={'newRT=D\$rt - 100 '};...</code> | Instrucciones o modificaciones de variables que se quieran ingresar en el modelo usando comando de R. |
| <code>pathtemp='path/to/write'</code>          | Directorio temporal donde guarada los datos que se requieren pasar enter matlab y R.                  |

Esta función escribe tados en el disco duro para traspasar la información entre matlab y R. Esto hace que no se requiera mucho RAM, a costa de que le proceso se enlentes. Para optimizar este proceso a las condiciones de cada experimento en particular, la función constan te tres etapas que se pueden realizar parcialmente, de la siguiente forma:

|                         |   |
|-------------------------|---|
| <code>onlyW=true</code> | Solo escribe los datos por electrodos en tablas que las pueda leer R. Se guardan en la carpeta especificada en <code>cfg.pathtemp</code> . Por defecto lastablas por electrodo se guarda como <code>borrameX.txt</code> , donde X es el número del electrodo en cuestión. Esto es útil cuando de los mismo datos se quieren probar diferentes modelos.  |
| <code>onlyR=true</code> | Solo realia los modelos en Rbasandose en las tablas previamente estritas en los archivos <code>borrameX.txt</code> . Los resultados se escriben en los archivos <code>borrameYX.txt</code> , donde Y es es resultado dependiendo de los modelos (Rc:coeficientes, Rp: p value,etc.) y X el electrodo en cuestion. Luego se estraen los resultados de estor archivos y se dejan en la estructura <code>COR.models</code> |
| <code>onlyE=true</code> | Solo se estraen los resultados de estor archivos <code>borrameYX.txt</code> y se dejan en la estructura <code>COR.models</code>   |

```
>> COR = modelr(COR, cfg)
```

## 5.9 Otras

### sound2event.m

De un canal de audio, selecciona picos (Respuestas) para transformarlos en eventos. `channel`, es el canal de audio, `code` es el código de los nuevos eventos, `threshold` es el umbral en desviaciones estándar para detectar el sonido, `win` es la ventana refractaria después de detectar una respuesta.

```
[LAN, cuantos] = sound2event(LAN, channel, code, threshold, win);
[LAN, cuantos] = sound2event(LAN, 1, 1, 4, 1000);
```

### pregunta\_lan.m v.0.1.3

Interfaz gráfica (GUI) para configurar archivos `cfg`.

`campos` campos de la estructura `cfg` a configurar, en cell-array con string `['campo1' 'campo2']`.

`opciones` Opciones que desplegar en una matriz de Cell-array con string, campos x opciones.

```
cfg = pregunta_lan(cfg, campos, opciones, label)
```

## Capítulo 6

# Algunas justificaciones teóricas

### 6.1 Entropía

sea  $x$  una variable aleatoria, se define incerteza

$$I(A_k) = \ln\left(\frac{1}{p_k}\right) = -\ln p_k$$

Se Define entropia de X a  $IE(i(x))$

$$-\sum_{k=1}^n p_k \ln(p_k)$$

$\log_2 \text{bits}$   
 $\ln \text{nats}$

#### Entropía Relativa

Si,  $p_k = PX = k$ , y  $k \in \text{Rec}X$ ,  $p_k$  se llama función de probabilidad de X.

Entonses la entopia relativa de dos funciones de probabilidad p y q, se define como, (la entropia de q con respecto a p):

$$H(p, q) = \sum_{k=1}^n p_k \ln\left(\frac{1}{q_k}\right) - H_X$$
$$= \sum_{k=1}^n p_k \ln\left(\frac{p_k}{q_k}\right) > 0$$

### 6.2 Estadística

Estadística paramétrica busca los parámetros de la distribución de la variables aleatorias, por lo tanto la distribución de la variable es conocida (Por ejemplo, distribución normal). La estadística para no- parametricas se usa cuando uno no conoce la distribución de la variable aleatoria. (Se puede usar como sinónimo de no paramétrica: distribución libre.)

S(x) es la función aleatoria de distribución empírica; que es

$$\frac{\text{número } x \leq a}{n}$$



La media muestral  $\bar{X} = \frac{1}{n} + \sum_{i=1}^n X_i$  Varianza muestral

Bootstrap se sacan muestras nuevas con reposición del mismo número de elementos de la muestra y el intervalo de confianza se saca con estadístico de orden, corresponde a  $(nb * 1 \pm \frac{\alpha}{2})$

Estadísticas consta de: Estimaciones Intervalos de confianza Test de hipótesis

Test de hipótesis simple: si aceptarla conduce a una única función de probabilidad: estimador con valor único  
complejas: si lleva más de una función de probabilidad

- Error I : se rechaza siendo verdadera.
- Error II: se acepta siendo falsa.

Significancia  $\alpha$  es la probabilidad máxima de rechazar  $H_0$  siendo verdadera

Estadístico, en una VA con la que se toma la regla de decisión Región crítica es donde se rechaza  $H_0$

La distribución nula es la distribución de prop de  $H_0$  siendo verdadera.

Potencia es  $1 - \beta$ , y es rechazar la hipótesis  $H_0$  falsa.

EL Valor  $p$  es el nivel de significancia más pequeño para el cual la hipótesis  $H_0$  puede ser rechazada.

Los test paramétricos la variable observada tiene distribución conocida y en la mayoría es normal. test es insesgado es cuando la  $p$  de rechazar  $H_0$  cuando es falsa es  $\geq$  a la probabilidad de rechazar  $H_0$  cuando es falsa.

Selección de test consistente, es cuando la potencia tiende a 1 cuando la muestra tiende a infinito

Eficiencia relativa de un test de iguales parámetros es igual a  $\frac{n_1}{n_2}$

Test es conservador (conservativo), si el nivel de significancia real es más pequeño que el calculado.

Test no paramétricos

### Test de Wilcoxon (Rangos Man-Whitman)

Uno tiene dos muestras ( $X$  y  $Y$ , siendo  $X < Y$ ) provenientes de distintas poblaciones. Se hace el estadístico de orden de las dos muestras juntas ( $R(XY)$ ). Esto supone que la distribución de las muestras es continua, y no hay valores repetidos. El estadístico se calcula así:

$$T = \sum_{i=1}^n R(X_i)$$
$$T_1 = \frac{T - E(T)}{\sqrt{V(T)}}$$

El test de Wilcoxon es:

- insesgado ( $P(-H_0 | -H_0) \geq P(-H_0 | H_0)$ )
- Consistente (si  $P(x > y)$ , la potencia  $P(-H_0 | H_1) \approx 1$ )

Esto para comparar distribuciones, pero al comparar media es test de consistencia.

Condiciones:

- Las muestras son independientes entre sí, y entre ellas.
- La escala de medición es al menos ordinal

La distribución de  $T_1$  es conocida.

### Test de Wilcoxon de los Rangos con Signos

Test para muestras pareadas, con dos observaciones tipo  $(X_i, Y_i)$ , que se pasan a una sola variable  $D_i = X_i - Y_i$ .

Supuesto para  $D_i$

- Distribución simétrica (media = mediana)
- Independientes
- Misma media
- La escala de medida es al menos intervalos

$R_i$  Es el signo del rango. Entonces  $R_i$  es igual al rango de los  $|D_i|$  si este es positivo, y es menos en rango de  $D_i$  si es negativo. En este caso el estadístico es  $T^+ = \sum R_i$ , para todos los  $R_i$  positivos. La distribución de este estadístico está tabulada, cuando no existen las diferencias 0, y bajo la hipótesis que la  $D_i$  tiene media 0,

# Bibliografía

- Lachaux, J. P., Rodriguez, E., Martinerie, J., & Varela, F. J. (1999). Measuring phase synchrony in brain signals. *Hum Brain Mapp*, 8, 194–208.
- Maris, E. & Oostenveld, R. (2007). Nonparametric statistical testing of EEG- and MEG-data. *J Neurosci Methods*, 164, 177–190.
- Stam, C. J., Nolte, G., & Daffertshofer, A. (2007). Phase lag index: assessment of functional connectivity from multi channel EEG and MEG with diminished bias from common sources. *Hum Brain Mapp*, 28, 1178–1193.