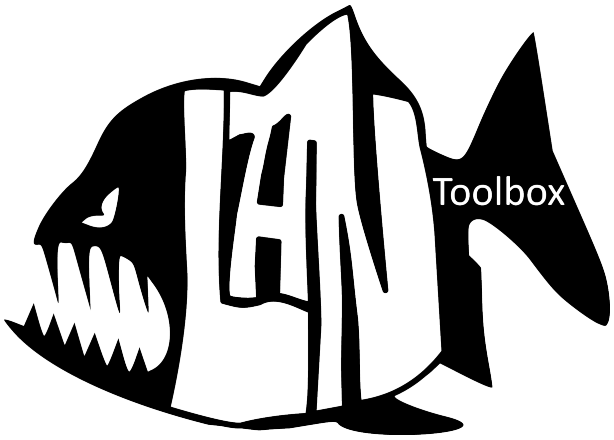


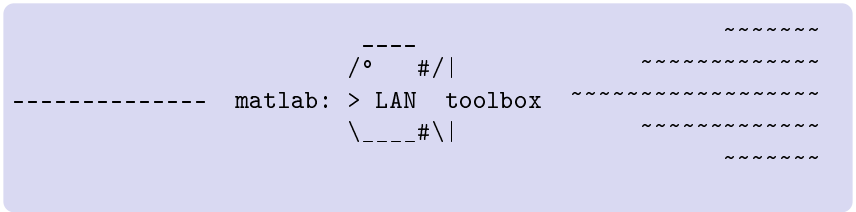
Analysis Laboratory in Neuroscience

beta version v.0.1.4.1

Pablo Billeke
Rodrigo Henriquez
Francisco Zamorano
LAN.toolbox@gmail.com

August 29, 2013

Contents



Chapter 1

Introduction

LAN: Analysis Laboratory in Neuroscience, is a Matlab toolbox for neuroscientist data (EEG and reaction time for the time being).The aim of this proyect is to create a shared lenguage among differents algorithms and softwares in the field (e.g. Fieldtrip, Eeglab, Chronux, Brainstorm, etc), in order to facilitate the implementation of experimental analysis by the users.

```
>> disp_lan

 TOOLBOX:            <°LAN)<]

      laboratorio de análisis en neurociencia

>>
```

1.1 Installation

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.

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 electrofiiológicos (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.

Chapter 2

Datos

Estructura de los datos: Existen dos tipos de estructuras, las estructuras simples, que representas 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 representanta 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)
	= o celda `{ensayos}` con matrices 2D `[electrodo, tiempo]`
LAN.time	= Tiempo inicial(en segundos) - Tiempo final (en segundos) - Punto cero (en punto)
	= e.g: `[-1 0.5 3457]`
	= 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']`,
	= si existe una condición por estructura
	= o estas están separadas en diferentes celdas.
LAN.conditions	= Estructura que delimita las diferentes condiciones que existen en la estructura LAN.
	= Esta forma es preferible cuando existen varias condiciones y algunas se sobrelapan
	= (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 especifico de la archivo .mat de los sujetos (ver por ejemplo **timefreq_stata.m**)

Campos

Campos Obligatorios	
GLAN.subject	= Sujetos que representa, en string con los nombre de los archivos `.mat`
	= [{'str'} , {'str2'} ; ...]
GLAN.cond	= vector con el índice de las condiciones en las estructuras LAN
	= [n,n,...]
GLAN.group	= vector con el indice de los grupos de los sujetos, debe coincidir con el número
	= de filas de `subject`
	= [{'str'} , {'str2'}...]
GLAN.srate	= Frecuencia de muestreo.
GLAN.time	= Tiempo.

ERP GLAN.erp.	
erp.data	= celdas de vector o matriz de los ERP promedios de los sujetos por condición
	= [{erp condición 1},{erp condición 2}]
erp.comp	= celdas con las condiciones comparadas para la estadística
	= [{[n1 n2]},{[n1 n3]},...]
erp.pval	= matrices de los valores p, en celdas por comparación realizada
erp.hh	= matrices lógicas de significancia, en celdas por comparación realizada
erp.hhc	= matrices lógicas de significancia, corregida por comparaciones múltiples,
	= en celdas por comparación realizada
erp.stat	= Resultados del estadístico utilizado, en celdas por comparación realizada
erp.cluster	= Índice de los cluster de electrodos, en celdas por comparación realizada

Tiempo-frecuencias GLAN.timefreq.	
timefreq.data	= celdas de vector o matriz de los cartas tiempo frecuencias promedios
	= de los sujetos por condición
	= [{TF cond 1},{TF cond 2}]
timefreq.comp	= celdas con las condiciones comparadas para la estadística
	= [{[n1 n2]},{[n1 n3]},...]
timefreq.pval	= matrices de los valores p, en celdas por comparación realizada
timefreq.hh	= matrices lógicas de significancia, en celdas por comparación realizada
timefreq.stat	= Resultados del estadístico utilizado, en celdas por comparación realizada
timefreq.cluster	= 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 `RT.conf.unit`)
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

Chapter 3

Ejemplos

3.1 Importación y pre-procesamiento

LAN 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 LAN. En este caso utilizamos la gorra de 40 canales NuAmp de Neuroscan, que se encuentra en el archivo `chanlocs_40neuroscan_nuamp(40).mat`

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

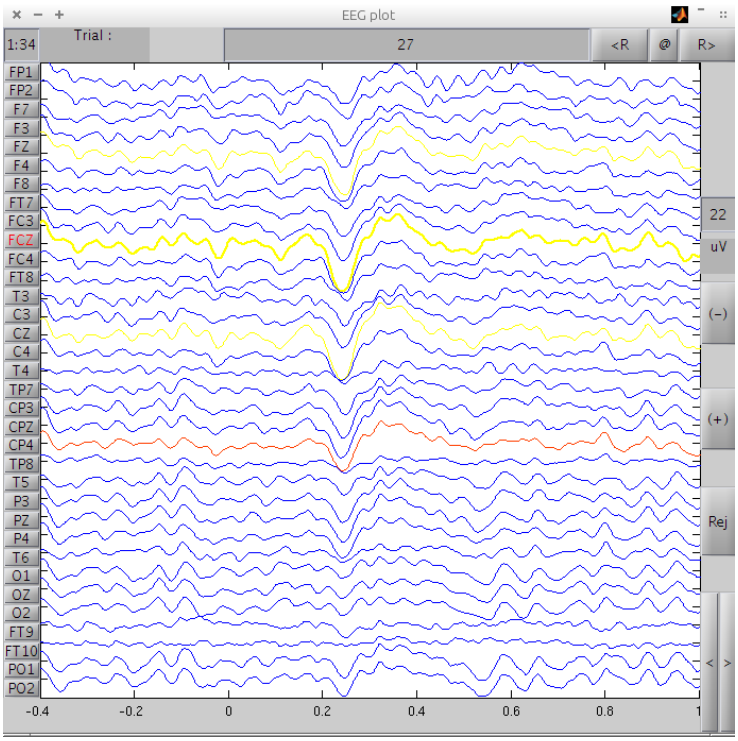
En el preprocesamiento se pueden eliminar electrodos no necesario como los oculare y las referencias, usando **electrode_lan.m**. Luego se pueden aplicara funciones para detectar artefactos. Están habilitados umbrales de voltaje y umbrales de amplitud por frecuencias. Ver las funciones **vol_thr_lan.m**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

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

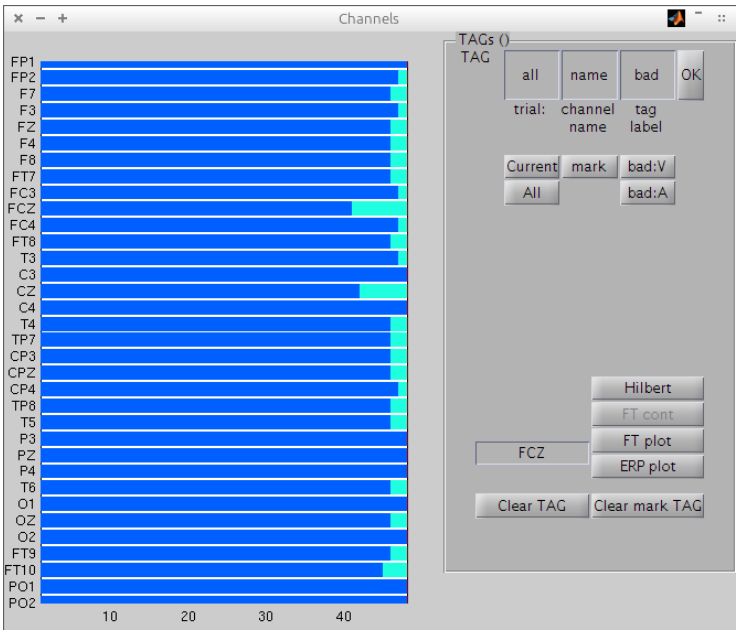
```
prepro_plot(LAN)  % GUI para inspeccion visual
```

Esta GUI consta de tres ventanas. La primera, el 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.

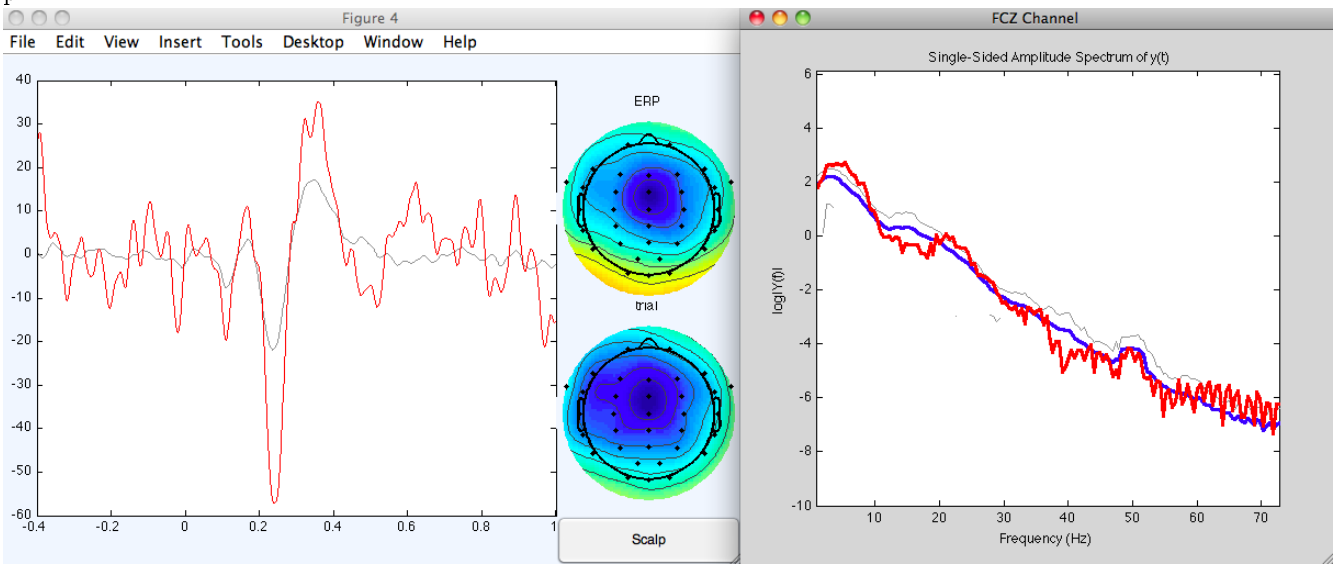


La segunda GUI (**EEG plot**), se pueden revisar el registro segmentado por ensayo. Los ensayos se pueden pueden cambiar por lo botones y , o directamente poniendo en número del ensayo que se desea en la cuadro editable **Trail:**. Cada canal se puede resaltar con el boton que indica su nombre, y con 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 meustra el resulatdo de `ERP plot` y de `FT plot` para el canl/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 muestar los TAG dismponible, pero se pueden agregar los que uno quiera. Se pueden cambiar el TAG de canal solo del esayo a del todo la condicion, usando el primer casillero (`'c'`, solo el ensayo actual; `'all'`, todos los ensayos) . La lógica de esto es poder marca cierto canales/ensayos para después poder realizar un proseso 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 cambio (la variable) en el espacio de trabajo, mediantes el boton `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 interporlar todos los canales/ensayos marcado de la siguiente forma:

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

Finalmente, guradamos la estructura LAN del sujeto.

```
 save LAN LAN
```

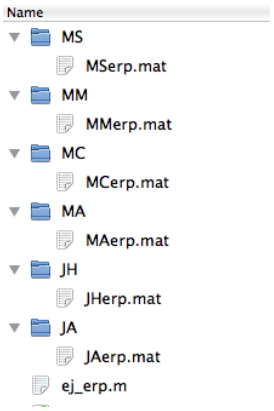
3.2 Potenciales relacionados a eventos

Para realizar potenciales relacionados a evento se requiere que todas las matricez 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 este 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:

```
% creando estructura de configuracion
cfg=[];
cfg.subject = [ ...  % ID de cada sujeto
                {'JA', 'JH' , 'MA' ,'MC' ,'MM' , 'MS'},...
               ];%
cfg.laplace=false; % opcion para realizar tranformada de lapalce
  %cfg.H = H;      % opctiones necesaria para la tranformacio
  %cfg.G=G;        % de lapalce ver: LAN_LAPLACE.m
  %cfg.chanlocs = chanlocs ;
cfg.comp = [3 4];  % indice de las condiciones a comparar
            % en este caso 3 = rechazos esperados
            % 4 = rechazos inesperados
cfg.alpha = 0.05;  % alfa para el primer test no parametrico
cfg.s='d';     % relacion entre las muestras
            % en este caso dependiente por ser entre los mismo sujetos
cfg.group = [1  1];% grupo al que pertencen el sujeto
  %cfg.groupname ={'controles'} % ID del grupo
  % no necesria especificar cuando hay un solo grupo
cfg.bl = [ ];      % linea de base para la normalizacion
cfg.stata = true;  % realizar estadistica
cfg.mcp = true;    % realizar correcion
cfg.mcpMt = 'CBP'; % correcion por permutacion
cfg.nrandom = 2000;% numero de permuatciones a realizar
cfg.savesub = true;% guardar ERP de cada suejto
cfg.delectrode = [1 2 5 6 27 33];
            % indice de los electrodos a elimianr 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. Ésto se realiza con dos parámetros `cfg.filename = 'nombre del archivo'` y `cfg.matname = 'nombre de la variable'`. Ambos parametros son caracteres o string. Existe ciertos caracteres especiales como `'%S'` que la funcioón va a remplazar por el ID de cada sujeto, y `'%G'` que es remplazado por el nombre del grupo de cada sujeto. A continuación un ejemplo de como organizar los archivos. En este casa 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'`.




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

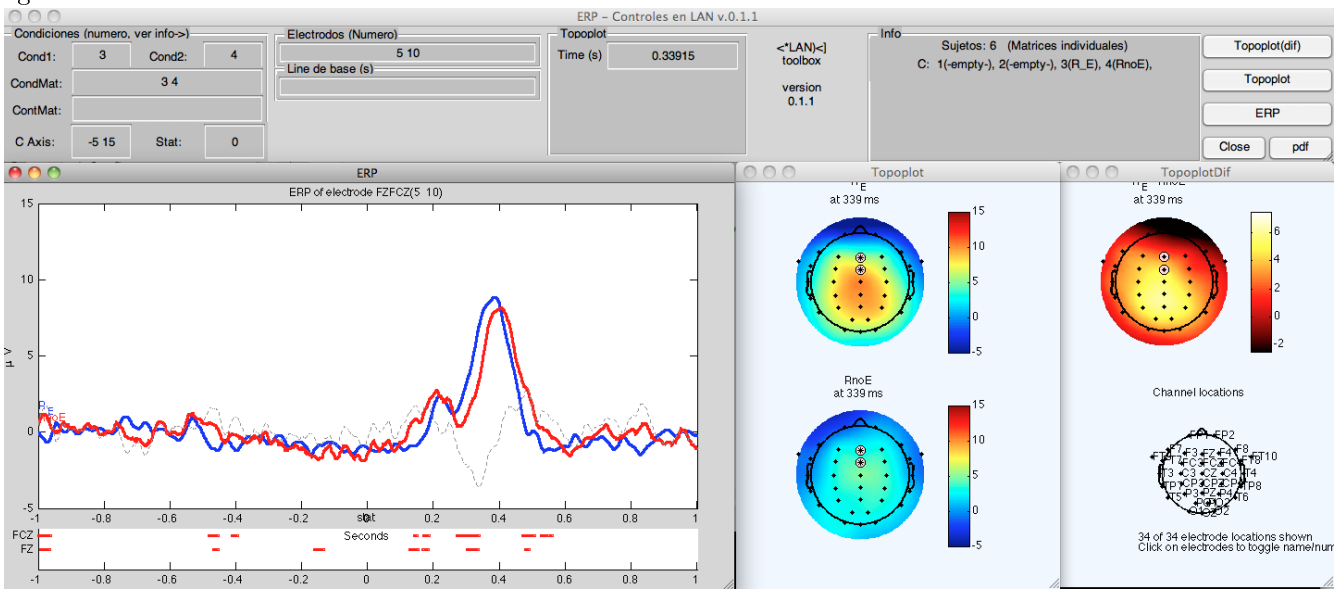
Luego se lanza la función, la que crea la estructura `GLAN` que contiene los ERP grupales y la estadística.

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

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

```
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 LAN 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.

Chapter 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 varaibel LAN o GLAN:

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

Chapter 5

Módulos

5.1 Basic Module

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**, elemina 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**:

`cfg.LAN1=LAN1` `='LAN1'`	Estructura LAN o `'str'` con el nombre de variable en workspace
`cfg.LAN2=LAN2` `='LAN2'`	idem
`cfg.sort=`	vector del orden de epocas con `0` sin data,`1` del lan1 `2` del lan2, eg: `[0 1 1 2 1 1 1 2 2 2 2 ]`

```
>> LAN = merge_lan(LAN1, LAN2,LAN3,...)
>> LAN = merge_lan(LAN)
>> LAN = merge_lan(cfg)
```

eeg2eeglab2lan.m v.0.0.2
Importa de archivo `.eeg` de NEUROSCAN.

`cfg.filenam='filename'`	Nombre del archivo, con su extención!.
`cfg.where='path'`	Directorio donde está el archivo.
`cfg.delbad=false`	elimina los ensayos malo, o solo los marca como tales (si es `=true`)

```
>> LAN = eeg2eeglab2lan('filename')
>> LAN = eeg2eeglab2lan(cfg)
```

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

Load a .erp file (ERPLAB) and the corresponding .set file (EEGLAB).

`erpfile=['filename.erp']`	Name (and path) of the ERPLAB file
`setfile=['filename.set']`	Name (and path) of the EEGLAB file. If there no existe the input search the file name in `.erp` file, and if there no existe the file display a pop up menu

```
>> 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.electrodemat 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 indices 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 lantency.m, sin más

actualizaciones)

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

```
>> [ 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 gurda 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, esayos malos mediante un umbral de voltage, y los etiqueta en la estructura TAG. paranetros son:

thr=[uV]	Umbral en microvols
tagname=['labels']	Etiqueta, por defecto 'bad:V'

```
>>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**

thr=[1 0.2]	Umbral para detectar épocas x canales malas. Vector de la forma [STD p] , 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
tagname='tagname'	string con el nombre (label) del TAG para marcar las épocas. Por defecto 'bad:A'.
frange=[f1 f2]	rango de frecuencia a explorar.
method='mt'	String con el método de descomposición tiempo frecuencia. Si 'mt' se realiza multitapers, sino se realiza fourier simple.
tapers=[TW K]	Definicion de tapers para 'mt', donde TW es el producto timepo-ancho de banda y K es el número de tapers (k <= 2*TW-1)
cat=1	Si es 1, se considera la desviación estándar de todos las condiciones, sino se calcula por condición.
nch=[elec]	Electrodos a evaluar, por defecto todos ('all').

```
>> 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).

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

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

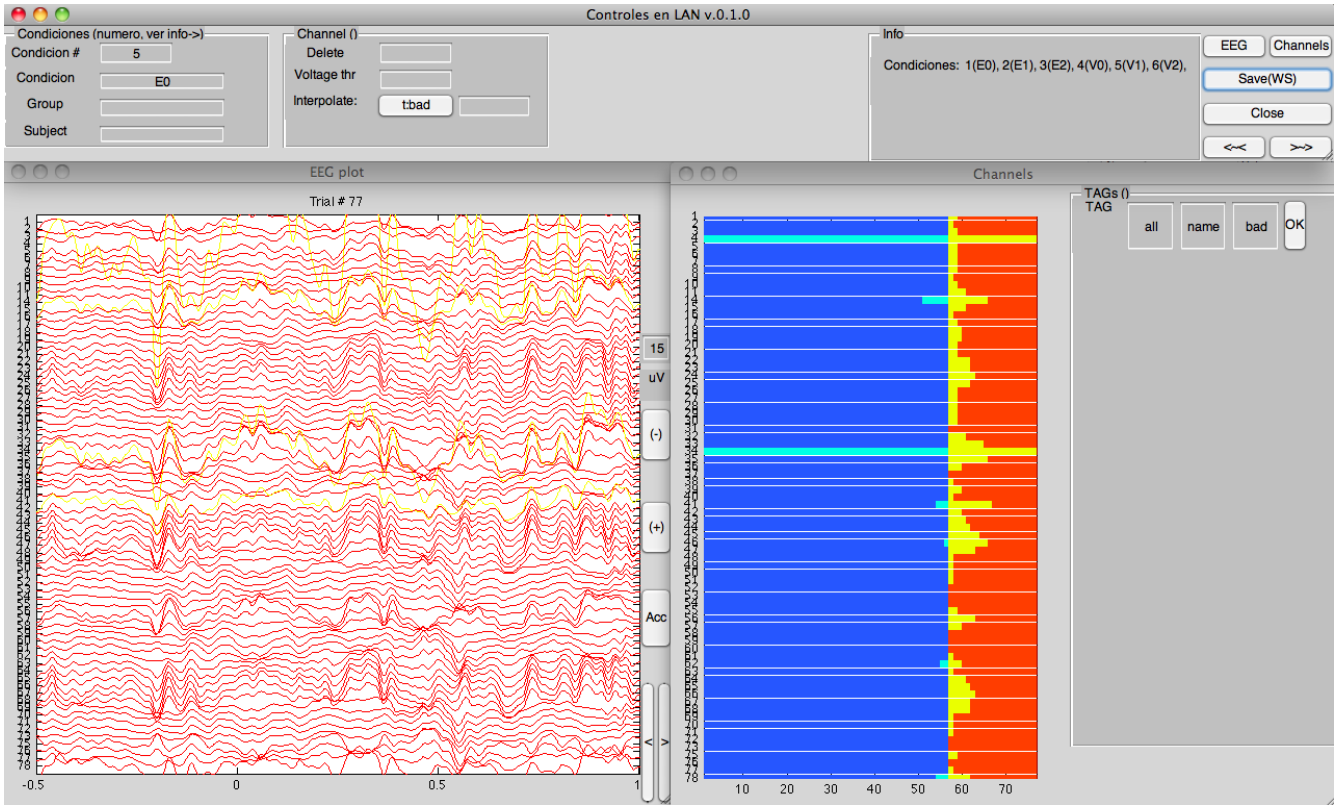
lan_laplace.m v.0.0.1 Realiza trandformación esferica usando lapalceanos, segun los algoritmo descritos por () y implementados en CDStoobox (). Más información en http://psychophysiology.cpmc.columbia.edu/Software/CSDtoolbox

cfg.header=10	Radio de la esfera, por defecto 10.
cfg.lambda=0.00001	parametro de suavizado.
cfg.H=H	Matrix de tranformación H precalculado, si no se encuentra la calcula para el montaje de electrodos tipo 'EEG'.
cfg.H=G	Matrix de tranformación G precalculado, si no se encuentra la calcula para el montaje de electrodos tipo 'EEG'.
cfg.chanlocs=chanlocs	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 peramétrica 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]`).

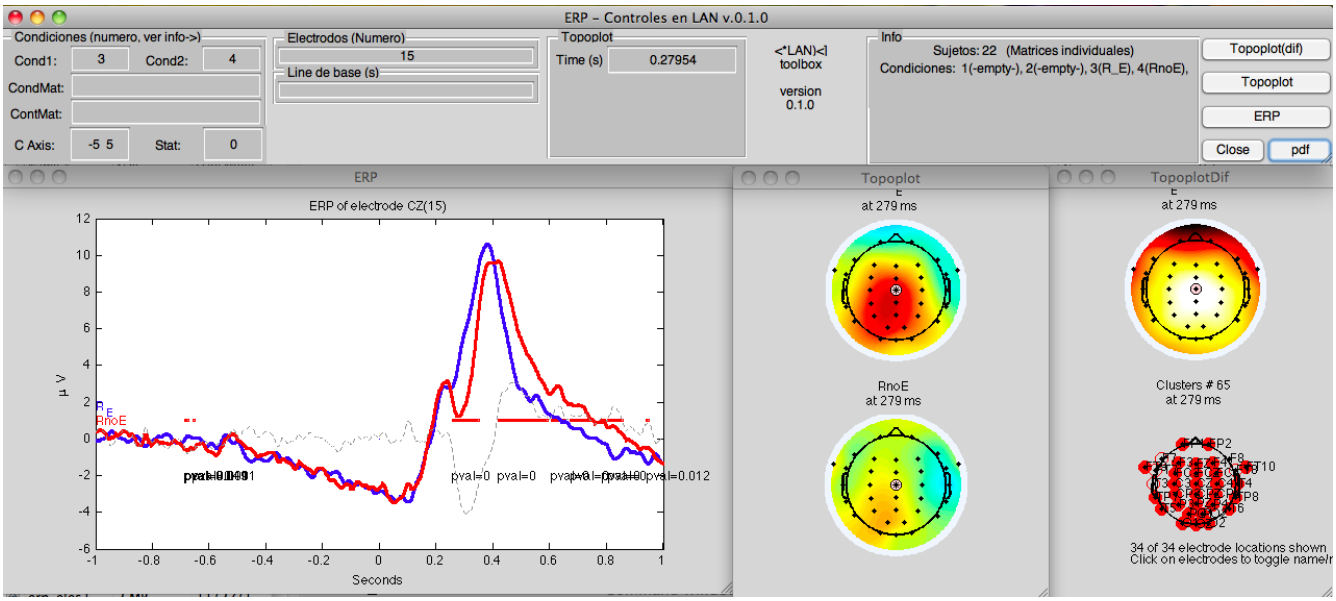
matdif=[1 -1 2 -2]	matriz de diferencias, del mismo tamaño que cfg.comp. donde los 1 corresponden al primer grupo, -1 el que se va arrestas a 1, los mismo para 2 y -2.
group=[n1 n2];	Grupos de individuos a comparar (indice dentro de cfg.subject).
groupname={'casos', 'controles' };	Nombre de los grupos de individuos a comparar, para identificar archivos
alpha=0.05;	
s='d'; OR 'i'	relación entre las muestras.
bl=[s1 s2];	Limites de la linea de base en segundos.
mcp=1 or 0	Para realizar corrección por comparaciones múltiples, por test de permutaciones.
nrandom=n	Número de randomizaciones para el test de permutaciones. Por defecto 100.
savesub=0	Guarda erp por sujetos, por defecto=0;
stata=1	Realiza estadística, por defecto=1;
srate=1000	Chequea la tasa de meustreo de las estructuras LAN, y re-muestrea las necesarias. Ver **resample_lan.m**
laplace=false	Si es true, realiza tranformación de laplace con parametros por defecto, usando CSD toolbox, ver **lan_laplace.m**.
reref=[4 75]	Rereferencia a los indices o etiquetas de los canales. Es preferible rereferenciar usando el modulo de prerprocesamiento. Precuación al usar delelctrode, chequear los índices de los electrodos
mcpMt='CBP'	Método de comparaciones. defecto:'CBP' cluter-based permutation; 'CRP' cluster-based randomization.
delelectrode=[$elec_1$ $elec_n$ ...]	elimina electrodos del análisis.
filename='str'	Especifica el nombre del archivo mat, ejemplo ./%S/%S. Por defecto es ./%S.mat
matname='str'	Especifica el mombre de la matriz. Por defecto es %S.
='%S'	Señala el nombre del sujeto en cfg.subject.
='%G'	Señala el nombre del grupo en cfg.groupname .

```
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` .

`chn=[elec1,elec2, ... ]` `='all'`	canales a realizar fourier. Por defecto `'all'`.
`ifplot=1`	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 :

type='str';	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 `cfg.subject{1} = {'s1','s2',..};` y los casos `cfg.subject{2} = {'sn1','sn2',..};`
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 `cfg.subject`)
matdif=[1 −1 2 −2 ..];	Si existe grupos de individuos son los índices de estos grupos (número de columna donde están en `cfg.subject`)
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 **normal_z.m**.
mcp=1 or 0	Para realizar corrección por comparaciones múltiples, por test de permutaciones(ver Maris & Oostenveld, 2007).
nrandom=n	Número de permutaciones para el test de permutaciones
delelctrode=[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.:
`bl = [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.:
`bl = [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:

$$s_a = s(t) + iH(s(t)) \tag{5.1}$$

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

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

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 diversoso 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.

`cfg.freq_method='Method'`	Método para calcular la fase y la amplitud instantanea. Exiten habilitados: `'Wavelet'`
`cfg.algoritm='Algoritmo'`	Algoritmo para calcular la sincronía de fase. Existen habilitados: `'PLV'` phase locking value, (Lachaux et al., 1999)
`cfg.across=' trials '`	Realiza el calculo de sincronía a través de los ensayos: `'trials'` o a través del tiempo : `'time'`
`cfg.step=[ n ]`	Paso para el calculo de syncronia a travez de los ensayos. Ventana para el calculo de la sincronía por tiempo.
`cfg.frange=[f1 f2]`	Rango de frecuencias para analizar la sincronía.
`cfg.ResHz=[ n ]`	Resolución en bin por Hz de frecuencias
`cfg.ncomponents=`	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.
`cfg.ifdiv=`	....
`cfg.ndiv=`	...

```
>> 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.

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

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

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:

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

```
[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.

`method='rank'`	Método,solo habilitado hasta el momento `'rank'`.
`paired=true`	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. `aplha` es nivel de significancia, `m` es la relación entre las muestras: 'i' independientes y 'd' dependiente. means = 1 se 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. `aplha` es nivel de significancia, `m` es la relación entre las muestras: 'i' independientes y 'd' dependiente. means = 1 se 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_bootstrapping.m v0.0.1

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.

`bl = [n n]` Limites de la linea de base en puntos, por defecto `[]`.

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

```
[hh] = freq_bootstrapping(freq,bl,alpha)
```

cor_freq_lan.m

Realiza gráfico y estadistica (Wilcoxon de suma de los rangos, muestras independientes) de los ρ de correlaciones entre frecuencias por trails 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 freciencia a correlacionar con corfr2; .corfr2 = {f1:f2} {} Rangos de freciencia 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 sofware R este instalado y el scrprit 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`

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

Ciertos formatos requieren configuraciones adicionales, como `'Neuroscan'` que requiere los siguentes parámetros:

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

```
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.

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

Adiciona 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 estructural `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 estimulo `'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 escript 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 del 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 libreria nlme: Linear and Nonlinear Mixed Effects Models. `'lmer'`: Comando para modelos mixtos usando libreria 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.

`newvar={'newRT=D$rt − 100 '; . . .}`	Instrucciones o modificaciones de variables que se quieran ingresar en el modelo usando comando de R.
`pathtemp='path/to/write'`	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 enlentese. 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 siguente forma:

`onlyW=true`	Solo escribe los datos por electrodos en tablas que las pueda leer R. Se guardan en la carpeta especificada en `cfg.pathtemp`. Por defecto lastablas por electrodo se guarda como `borrameX.txt`, donde X es el número del electrodo en cuestión. Esto es útil cuando de los mismo datos se quieren probar diferentes modelos.
`onlyR=true`	Solo realia los modelos en Rbasandose en las tablas previamente estritas en los archivos `borrameX.txt`. Los resultados se escriben en los archivos `borrameYX.txt`, 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 `COR.models`
`onlyE=true`	Solo se estraen los resultados de estor archivos `borrameYX.txt` y se dejan en la estructura `COR.models`

```
>> 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)
```

Chapter 6

Algunas justificaciones teóricas

6.1 Entropía

sea x una variable aleatoria, se define incerteza

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

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

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

$$log_2 bits$$

$$lnnats$$

Entropía Relativa

Si, $p_k = PX = k$, y $k \in RecX$, 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 $X = \frac{1}{n} + \sum_{i=1}^{n} X_i$ Varianaza muestral

Boostrap se sacan muestras nuevas con reposión del mismo número de elemtosde la muestar y el intervalo de confinza se saca con estadistico de orden, correspondeinete a $(nb * 1 \pm \frac{\alpha}{2})$

Estadisticas consta de: Estimaciones Intervalos de confianza Test de hipotesis

Test de hipotesis simple: si aceptarla conduce a una unica función de probabilidad: estimador con valor unico compuetas: si lleba más de una función de probabilidad

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

Significancia α es la probabilidad máxima de rechazar h_0 siendo verdadera

Estadistico, en una VA con la que se toma la regla de decición Región critica 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 rechachar la hipotesis H_0 falsa.

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

Los test parametricos la variable observada tiene districbución conocida y en la mayoria es normal. test es insesgado es cuando la p de rechazar H_0 cuando es falsa es $>=$ a la probabilidda de rechazar H_0 cuando es falsa.

Suseción de test consitente, es cuando la potencia tiende a 1 cuando la muestra tiende a infinito

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

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

Test no parametricos

Test de Wilcoxon (Rangos Man-Whitman)

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

$$T = \sum_{i=1}^{n} R(X_i)$$

$$T_1 = \frac{T - E(T)}{\sqrt{V(T)}}$$

El test de Wilcoxon es:

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

Esto para comparar distrubuciones, pero al comparar media es tes tierde consistencia.

Condiciones:

- Las muestras son independientes en si, y entre si.
- La escala de medición es almenos ordinal

La distribución de T1 es conocida.

Test de Wilcoxon de los Rangos con Signos

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

- Distribución simetrica (media = mediana)
- Independientes
- Misma media
- La escala de medida es almenos intervalos

R_i Es el signo del rango. Entonses 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 estadistico es $T^+ = \sum R_i$, para todos los R_i positivos. La distribución de este estadistico está tabulada, cuando no existen las diferencias 0, y bajo al hipotesis que la D_i tiene media 0,

Bibliography

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.