

## siSig

・メンバ変数

data : 信号を保持

trig : トリガデータを保持

time : 時刻データを保持 (unit : s)

props : vhdrデータ等に含まれる計測条件を保持

etc : siSigクラスによって行われた処理のパラメータを保持

### ***siSig(string filename)***

パス内に含まれる, filenameのvhdr, vmrk, eegファイルをインポート

filenameには拡張子を含めない

e.g.

filename = '20210729\_B50\_online\_0001'

### ***siSig(siSig array)***

複数セッションの信号を結合し, 処理したい場合に用いる.

引数で渡されたsiSigのデータを結合

### ***plot()***

multichanplot関数を用いた可視化

### ***filtfilt(double A, double B)***

filtfilt関数を用いたゼロ位相フィルタの適用

### ***zf = filter(double A, double B)***

filter関数を用いたフィルタリング

フィルタ適用後のタップの最終状態をzfとして返す

・フィルタについて

$$Y(z) = \frac{b(1) + b(2)z^{-1} + \dots + b(n_b + 1)z^{-n_b}}{1 + a(2)z^{-1} + \dots + a(n_a + 1)z^{-n_a}} X(z)$$

MATLABのフィルタの伝達関数は上のようになる。ここで、 $a(n)$ をすべて0とし、分母を1とするとFIRフィルタとなる。この、A、Bは書籍等により逆に表記されている場合があるので注意。

MATLABのsignal processing toolboxを使ってフィルタ設計をする場合、戻り値のA、Bはおそらくそのまま使える。

***[trig\_list, num\_trig] = get\_trig\_list()***

trig\_list : 1行目にサンプル番号, 2行目にトリガ番号が格納された行列

num\_trig : 1行目にトリガ番号, 2行目にそのトリガの出現数が格納された行列

# siEpoch

・メンバ変数

data : 信号を保持

time : 時刻データを保持 (unit : s)

num : エポックデータ数

props : 引数として渡されたsiSigクラスのインスタンスが保持していたパラメータ

etc : siEpochクラスによって行われた処理のパラメータを保持

## ***siEpoch(siSig signal , trig\_num, range, baseline)***

siSigクラスで信号を結合, フィルタ処理しsiEpochのコンストラクタに渡すことで, エポッキング, ベースライン補正が行われます.

signalには, siSigクラスのインスタンス

trig\_num : エポッキングしたいトリガ番号

range : トリガ発生時刻を0としたエポックデータ切り出し区間 (unit : s) [1x2 array]

baseline : トリガ発生時刻を0としたベースライン区間 (unit : s) [1x2 array]

また, ベースライン処理を行いたくないときは, baselineに空行列 [] を渡してください

e.g.

```
epoch = siEpoch(siSig, 2, [-0.1 0.5], [-0.05 0])
```

## ***rej\_th(ch, th)***

閾値thを指定し, その範囲を超えたエポックデータを棄却します. chで任意のチャンネルを指定することができます.

ch : 閾値を用いた棄却判断をするチャンネル e.g. [1 3 5 8]

th : 閾値 [min max] のフォーマットで指定 (unit : micro volts) e.g. [-100 100]

e.g.

siEpoch.rej\_th(1:64, [-100 100]) -> 1:64チャンネルがプラマイ100uVを超えたら棄却

siEpoch.rej\_th(65:66, [-300 300]) -> 65,66チャンネルがプラマイ300uVを超えたら棄却

脳波と眼電それぞれに, 異なる閾値を指定したりするときに使います

# siPlot

ERPプロット用ライブラリ

## ***siPlot(div, varargin)***

div : [div\_row div\_culumn]

matlabのsubplot(a, b, n)のa,b部分を指定

表示されるfigureを行, 列方向に何分割するかを指定します.

e.g.

```
plt = siPlot([3 4])
```

• option

‘axis’ : ‘ij’ を指定すると, y軸が反転する

## ***id = plot(loc, epoch, time, options)***

データをプロット

loc : コンストラクタ実行時に指定したdivのうち, 何番目にプロットするかを指定

matlabのsubplot(a, b, c)のcを指定

epoch : エポックデータを渡します. time \* N

time : 渡したエポックデータの時刻データ (unit : s)

options :

• plt\_mode

指定しない場合, averageになります

‘average’ : 加算平均してプロット (エポックデータの2次元の方向に加算平均)

‘butterfly’ : バタフライプロット

• matlabのplot関数が戻すLineオブジェクトのプロパティを指定することが可能

つまり, matlabのplot関数のオプションはそのまま記述可能. また, Lineオブジェクトのプロパティを直接指定することもできる.

e.g.

```
plot(loc, epoch, time, ‘plt_mode’, ‘average’, ‘color’, ‘r’, ‘linewidth’, 2);
```

• 戻り値のidについて

各loc内にplotしたlineオブジェクトのid番号を返します.

このidは, 各loc内にplotした順番です.

## ***axis\_x(loc, options)***

locで指定した場所のプロットにx軸を表示

matlabのplot関数を用いてLineオブジェクトを生成し, 軸を表示

loc : コンストラクタ実行時に指定したdivのうち, 何番目にプロットするかを指定

matlabのsubplot(a, b, c)のcを指定します  
options : matlabのplot関数を用いて描画されるLineオブジェクトのプロパティを指定  
plot関数のオプション, Lineオブジェクトのプロパティを指定

e.g.

```
axis_x(loc, 'color', 'r', 'linewidth', 2)
```

## ***axis\_y(loc, options)***

locで指定した場所のプロットにy軸を表示

loc : コンストラクタ実行時に指定したdivのうち, 何番目にプロットするかを指定

matlabのsubplot(a, b, c)のcを指定します

options : matlabのplot関数を用いて描画されるLineオブジェクトのプロパティを指定  
plot関数のオプション, Lineオブジェクトのプロパティを指定

e.g.

```
axis_y(loc, 'color', 'r', 'linewidth', 2)
```

## ***ttest(loc, id\_1, id\_2, alpha, options)***

2標本間のt検定を算出し, 有意差有りと判定された時刻を網掛けする.

loc : 何番目のplotかを指定

id\_1 : 標本1とするデータのidを指定 -> siPlot.plot() 関数が戻す値

id\_2 : 標本2とするデータのid

alpha : t検定の有意水準を指定 -> 0.01で1%の危険率

'mode' : 'mesh' or 'line' -> 指定しない場合meshになる

'mesh'を選択すると, グレーの網掛けで検定結果が表示される

'line'を選択すると, 線で検定結果が表示される

'height', 及び'pos'オプションが指定できる

height : プロットのy軸の何%の高さのラインを表示するか指定 -> defaults : 5

pos : プロットの上端から何%の位置にラインを表示するか指定 -> defaults : 1%間隔で表示

e.g.

```
ttest(loc, id_1, id_2, 0.01, 'mode', 'line', 'height', 5, 'pos', 10)
```

y軸の5%の高さの線が, 上端から10%の位置に表示される.

options : matlabのfill関数を用いて適用される網掛けのオプションを指定

fill関数のオプションがそのまま適用できるほか, fill関数が戻すPatchオブジェクトのプロパティを直接指定することができる

## ***plot\_stim(loc, range, varargin)***

指定した時刻をマーキング. 刺激提示時刻の表示を念頭に作成.

loc : 何番目のplotか指定

range : n by 2 の行列. 1行のフォーマットは, [range\_1 range\_2] (unit : s)

・オプション

height : プロットのy軸の何%の高さのラインを表示するか指定 -> defaults : 5

pos : プロットの上端から何%の位置にラインを表示するか指定 -> defaults : 1%間隔で表示

## ***change\_props(arg, field, val)***

内部処理用

フィールドのプロパティを変更する.

fieldで指定するフィールド名は大文字小文字を問わない.

arg : 変更したいフィールドを保持する構造体を渡す

field : field名をcharで指定

val : fieldの値

## ***opt\_lim(loc) or opt\_lim(loc, rate)***

内部処理用

指定した場所のプロットのylimをデータに合わせる.

既定では, プロットされているデータの最大, 最小値を取得し, 5%大きくしたylimを指定

rateを引数として指定すると, 何%大きくするか指定できる.

rate = 1にすると, データの最大, 最小と同じ

rate = 1.05にすると, 5%大きくなる