# EEGLab function analysis

## newtimefbaseln.m

displaying using std\_plottf.m

parameters

1. INPUT: PPori(NxM-cell of 3D/4D-doubles; FreqxTimexTrials or ChansxFreqxTimesxTrials), timesout(1D-doubles)
2. paramsersp.powbase = double/NaN;
   * use an already defined baseline
3. paramsersp.verbose = ‘on’/’off’;
   * prints to command line
4. **paramsersp.singletrials = 'off'/’on’;**
   * **average power over trials (3rd/4th dimension of each cell in PPori)**
5. **paramsersp.commonbase = 'off'/’on’;**
6. **paramsersp.basenorm = 'off'/’on’;**
7. **paramsersp.trialbase = 'off'/’on’;**
8. **paramsersp.baseline = [TIME\_BOUND(1),TIME\_BOUND(end)]/NaN**

high-level understanding

1. IF powbase=NaN; IF singletrials=’on’ && trialbase=’off’, for each cell in **PPori** the mean across the 3rd dimension will replace each cell in **PPori**. Then, the mean across the 2nd dimension (time) will be taken to create the baseline spectrum. ELSE just the mean across time will be taken for each element in the 3rd dimension. ELSE powbase=double; double will be the baseline. THIS STEP CREATES **PP**, **allMbase**, and **allMstd**.
2. IF commonbase=’on’; for each baseline spectrum from (1) find the mean across those baselines. THIS STEP MODIFIES **allMbase** and **allMstd**
3. IF baseline=double; IF trialbase= ‘on’ && basenorm=’off’; divide all elements in PP by its respective element in **allMbase.** ELSEIF trialbase=’on’ && basenorm=’on’; subtract each element in allMbase from its respective element in PP *then* divide by allMstd.

NON-LOG INPUT PERMUTATIONS (RE-LOG BEFORE PLOTTING)

call:: allersp = newtimefbaseln(allersp, hardcode\_times, paramsersp);

1. nonlog-transformed TF data, doubles
2. NaN
3. ‘on’
4. **‘off’**
5. **‘off’**
6. **‘off’**
7. **‘off’**
8. **NaN**

A close-up of a chart

Description automatically generatedcall:: allersp = newtimefbaseln(allersp, hardcode\_times, paramsersp);

1. nonlog-transformed TF data, doubles
2. NaN
3. ‘on’
4. **‘off’**
5. **‘off’**
6. **‘off’**
7. **‘off’**
8. **[TIME\_BOUND(1),TIME\_BOUND(end)]**

A close-up of a screen

Description automatically generated

call:: allersp = newtimefbaseln(allersp, hardcode\_times, paramsersp);

1. nonlog-transformed TF data, doubles
2. NaN
3. ‘on’
4. **‘on’**
5. **‘off’**
6. **‘off’**
7. **‘off’**
8. **[TIME\_BOUND(1),TIME\_BOUND(end)]**

A comparison of a number of labels

Description automatically generated with medium confidence

call:: allersp = newtimefbaseln(allersp, hardcode\_times, paramsersp);

1. nonlog-transformed TF data, doubles
2. NaN
3. ‘on’
4. **‘on’**
5. **‘on’**
6. **‘off’**
7. **‘off’**
8. **[TIME\_BOUND(1),TIME\_BOUND(end)]**

A screenshot of a screen

Description automatically generated

call:: allersp = newtimefbaseln(allersp, hardcode\_times, paramsersp);

1. nonlog-transformed TF data, doubles
2. NaN
3. ‘on’
4. **‘on’**
5. **‘on’**
6. **‘on’**
7. **‘off’**
8. **[TIME\_BOUND(1),TIME\_BOUND(end)]**

A comparison of red squares

Description automatically generated

call:: allersp = newtimefbaseln(allersp, hardcode\_times, paramsersp);

1. nonlog-transformed TF data, doubles
2. NaN
3. ‘on’
4. **‘on’**
5. **‘on’**
6. **‘on’**
7. **‘on’**
8. **[TIME\_BOUND(1),TIME\_BOUND(end)]**

A close-up of a chart

Description automatically generated

call:: allersp = newtimefbaseln(allersp, hardcode\_times, paramsersp);

1. nonlog-transformed TF data, doubles
2. NaN
3. ‘on’
4. **‘on’**
5. **‘on’**
6. **‘off’**
7. **‘on’**
8. **[TIME\_BOUND(1),TIME\_BOUND(end)]**

A screenshot of a computer generated image

Description automatically generated

call:: allersp = newtimefbaseln(allersp, hardcode\_times, paramsersp);

1. nonlog-transformed TF data, doubles
2. NaN
3. ‘on’
4. **‘off’**
5. **‘on’**
6. **‘off’**
7. **‘on’**
8. **[TIME\_BOUND(1),TIME\_BOUND(end)]**

A comparison of a heat wave

Description automatically generated with medium confidence

call:: allersp = newtimefbaseln(allersp, hardcode\_times, paramsersp);

1. nonlog-transformed TF data, doubles
2. NaN
3. ‘on’
4. **‘off’**
5. **‘off’**
6. **‘off’**
7. **‘on’**
8. **[TIME\_BOUND(1),TIME\_BOUND(end)]**

**A close-up of a chart

Description automatically generated**

LOG INPUT PERMUTATIONS

call:: allersp = newtimefbaseln(allersp, hardcode\_times, paramsersp);

1. log-transformed TF data, doubles
2. NaN
3. ‘on’
4. **‘off’**
5. **‘off’**
6. **‘off’**
7. **‘off’**
8. **NaN**

A close-up of a chart

Description automatically generated

call:: allersp = newtimefbaseln(allersp, hardcode\_times, paramsersp);

1. log-transformed TF data, doubles
2. NaN
3. ‘on’
4. **‘off’**
5. **‘off’**
6. **‘off’**
7. **‘off’**
8. **[TIME\_BOUND(1),TIME\_BOUND(end)]**

A close-up of a screen

Description automatically generated

call:: allersp = newtimefbaseln(allersp, hardcode\_times, paramsersp);

1. log-transformed TF data, doubles
2. NaN
3. ‘on’
4. **‘on’**
5. **‘off’**
6. **‘off’**
7. **‘off’**
8. **[TIME\_BOUND(1),TIME\_BOUND(end)]**

A screenshot of a computer screen

Description automatically generated

call:: allersp = newtimefbaseln(allersp, hardcode\_times, paramsersp);

1. log-transformed TF data, doubles
2. NaN
3. ‘on’
4. **‘off’**
5. **‘on’**
6. **‘off’**
7. **‘off’**
8. **[TIME\_BOUND(1),TIME\_BOUND(end)]**

**A screenshot of a computer screen

Description automatically generated**

call:: allersp = newtimefbaseln(allersp, hardcode\_times, paramsersp);

1. log-transformed TF data, doubles
2. NaN
3. ‘on’
4. **‘off’**
5. **‘on’**
6. **‘off’**
7. **‘on’**
8. **[TIME\_BOUND(1),TIME\_BOUND(end)]**

**A screenshot of a computer screen

Description automatically generated**

(no change) permutation 5

call:: allersp = newtimefbaseln(allersp, hardcode\_times, paramsersp);

1. log-transformed TF data, doubles
2. NaN
3. ‘on’
4. ‘on’
5. ‘off’
6. ‘off’
7. ‘off’
8. NaN

A close-up of a chart

Description automatically generated

(no change) permutation 6

call:: allersp = newtimefbaseln(allersp, hardcode\_times, paramsersp);

1. log-transformed TF data, doubles
2. NaN
3. ‘on’
4. ‘on’
5. ‘on’
6. ‘off’
7. ‘off’
8. NaN

A close-up of a chart

Description automatically generated

(no change) permutation 7

call:: allersp = newtimefbaseln(allersp, hardcode\_times, paramsersp);

1. log-transformed TF data, doubles
2. NaN
3. ‘on’
4. ‘on’
5. ‘on’
6. ‘on’
7. ‘off’
8. NaN

A close-up of a chart

Description automatically generated

(no change) permutation 8

call:: allersp = newtimefbaseln(allersp, hardcode\_times, paramsersp);

1. log-transformed TF data, doubles
2. NaN
3. ‘on’
4. ‘on’
5. ‘on’
6. ‘on’
7. ‘on’
8. NaN

A screenshot of a computer generated image

Description automatically generated

(no change) permutation 9

call:: allersp = newtimefbaseln(allersp, hardcode\_times, paramsersp);

1. nonlog-transformed TF data, doubles
2. NaN
3. ‘on’
4. ‘on’
5. ‘off’
6. ‘off’
7. ‘off’
8. NaN

A close-up of a chart

Description automatically generated

NOTE: pretty sure without a baseline time (8) these options don’t do anything