



Open  
Electrophysiology

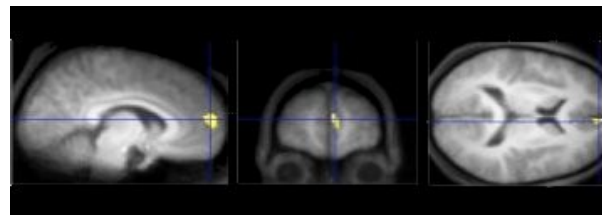
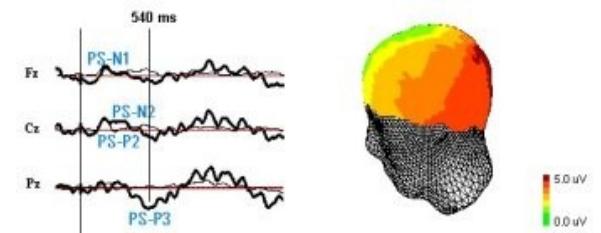
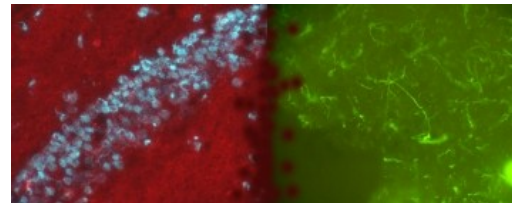


# Labo Neurosciences Sensoriel Comportement Cognition

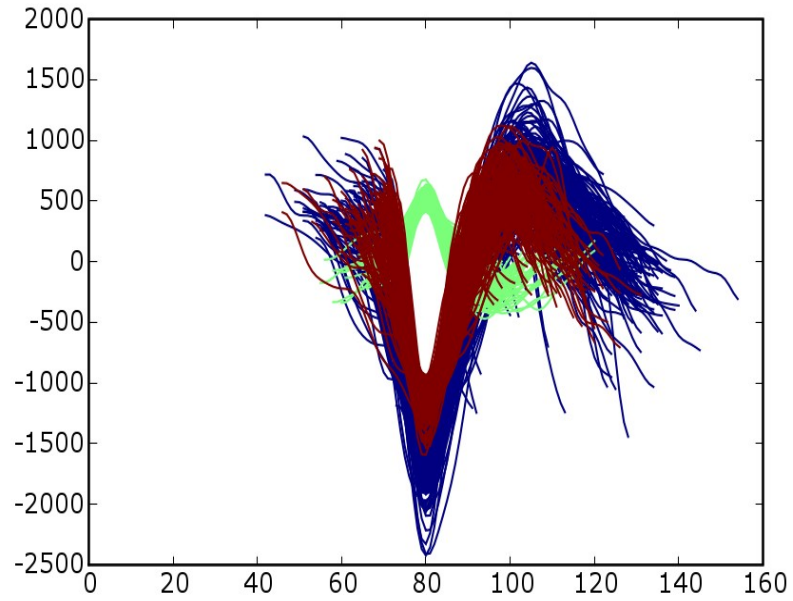
Director : Rémi Gervais

Samuel Garcia  
Data processing engineer, CNRS

Thèmes Labo : Olfaction and Audio

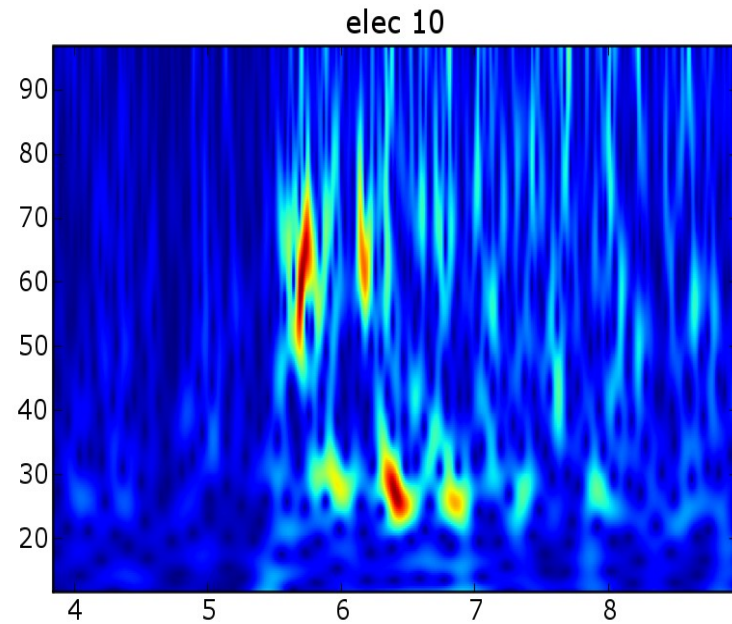


# Goal : mix 2 analysis



oscillations study

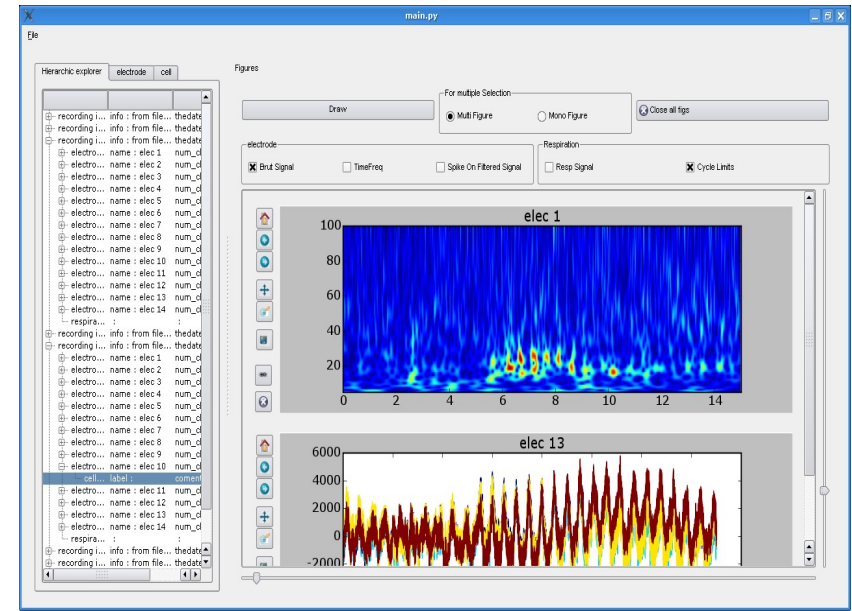
spikes study



# For who ?

## 2 levels for users :

a graphic user interface for non-scripters



A lower level : script with python + scipy + matplotlib + « simplified database storage »

```
visualise_max_oscillation.py - KWrite
Fichier Édition Affichage Signets Outils Configuration Aide

def visualise_max_oscillation(id_epoch = None) :
    """
    """
    ep = Epoch(id_epoch = id_epoch)
    trial = Trial()
    trial.load_from_db(ep.id_trial)

    query = """
    .         SELECT id_electrode
    .         FROM electrode
    .         WHERE electrode.id_trial = %s
    .         ORDER BY electrode.num_channel
    .         """
    id_electrodes, = sql(query,ep.id_trial)
    for e,id_electrode in enumerate(id_electrodes) :
        print 'id_electrode' , id_electrode
        fig = pylab.figure()

        elec =Electrode()
        elec.load_from_db(id_electrode)
        ind, = where( (elec.t() > ep.begin+temps_av) & (elec.t() < ep.begin+temps_ap))
        elec.signal = elec.signal[ind]
        elec.shift_t0 = temps_av

        ax1 = fig.add_subplot(2,2,1)
        elec.plot_bandwidth(ax = ax1, title = 'plein bande')
```

Based on :

MySQL : famous database server



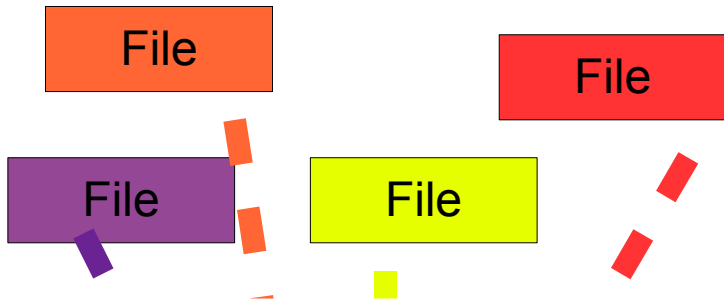
Scipy : scientific python module



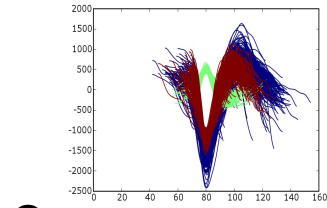
## •Avantages :

- 
- centralized ( Client-Server design)
- dataming (data and meta-data at the same place)
- efficient (MySQL queries)
- data sharing (you can open your server)
- analysis sharing (same database structure)
- free !
-

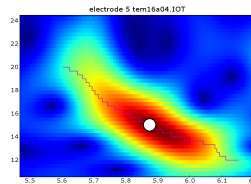
# Principle



Extraction of data and meta data

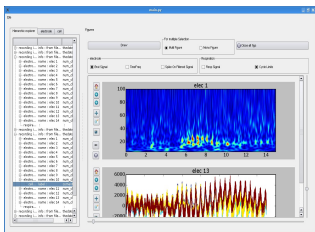


Spike detection



Oscillations detection

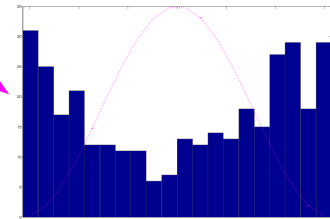
signal display



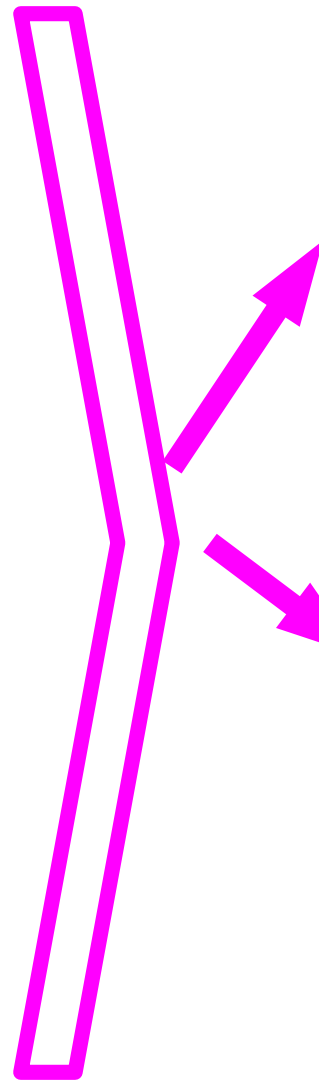
```
def visualise_mux_oscillation(id_epoch = None):  
    ...  
    ep = Epoch[id_epoch = id_epoch]  
    trial = Trial[]  
    trial.load_from_db(ep.id_trial)  
    query = ...  
    ...  
    id_electrodes = sql_query.ep.id_trial  
    for e_id_electrode in enumerate(id_electrodes):  
        print id_electrode - id_electrode  
        fig = pylab.figure()  
        elec = electrodes[  
            elec_load_from_db(id_electrode,  
                id = where('elec_id' = ep.begin_temp_wv' & elec_id = ep.begin_temp_wv')  
            elec_signal = elec.signal.load()  
            elec_shift_id = temp_wv  
            ax1 = fig.add_subplot(2, 1, 1)  
            elec_plot_bandwidth(ax = ax1, title = 'plain bande']
```



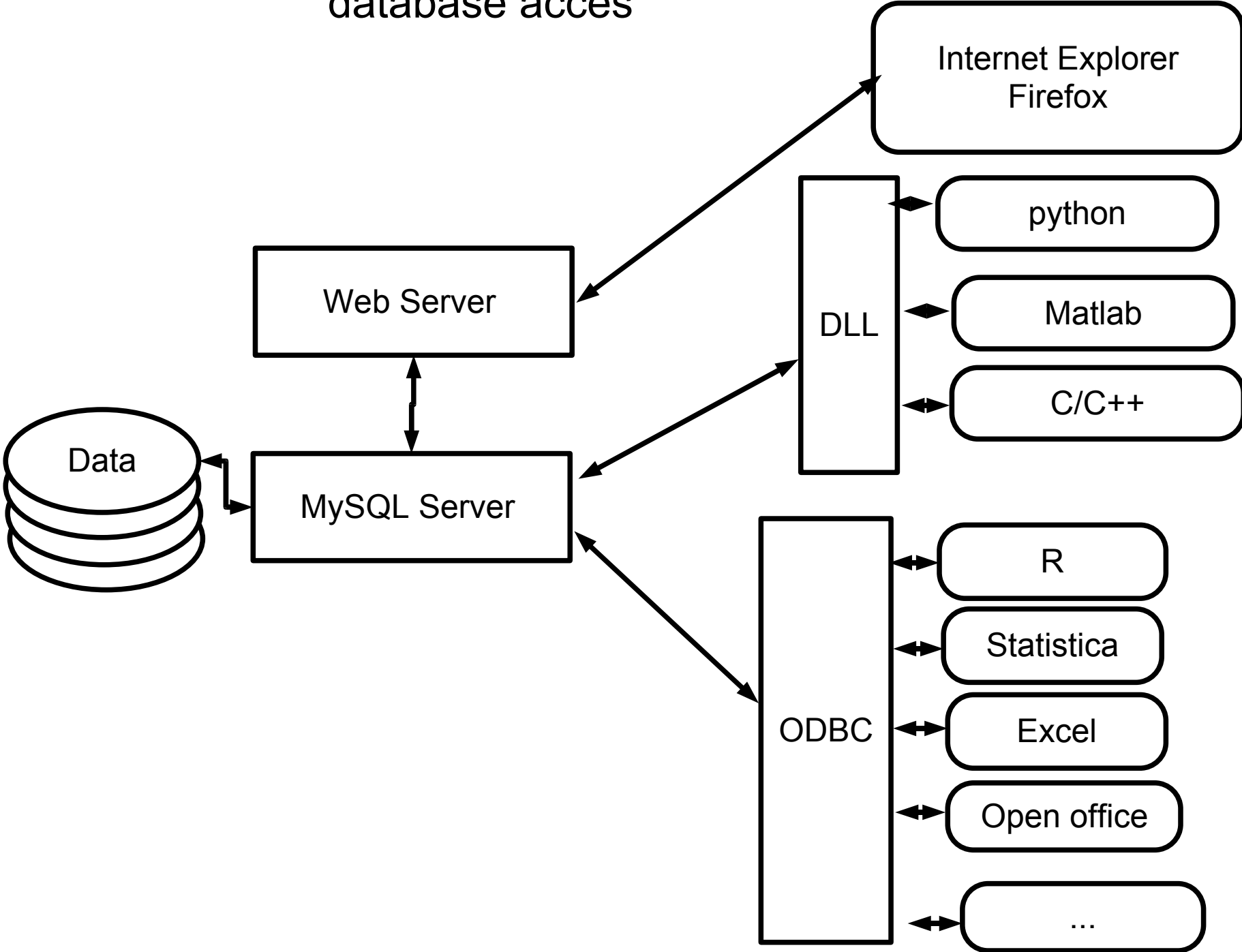
Analysis with python script (or other)



Analysis with the GUI

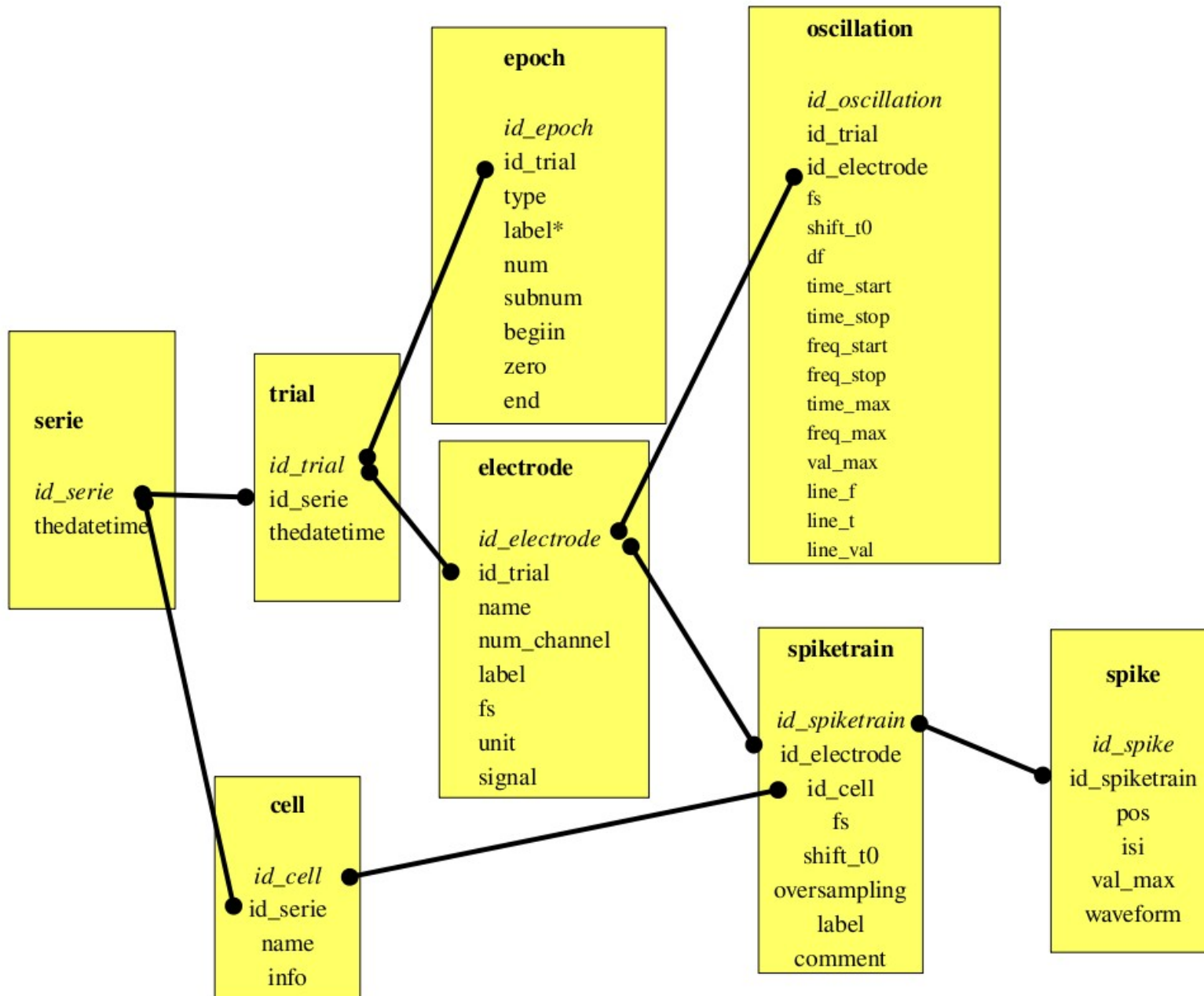


database acces





# Database Structure



## SQL Example :

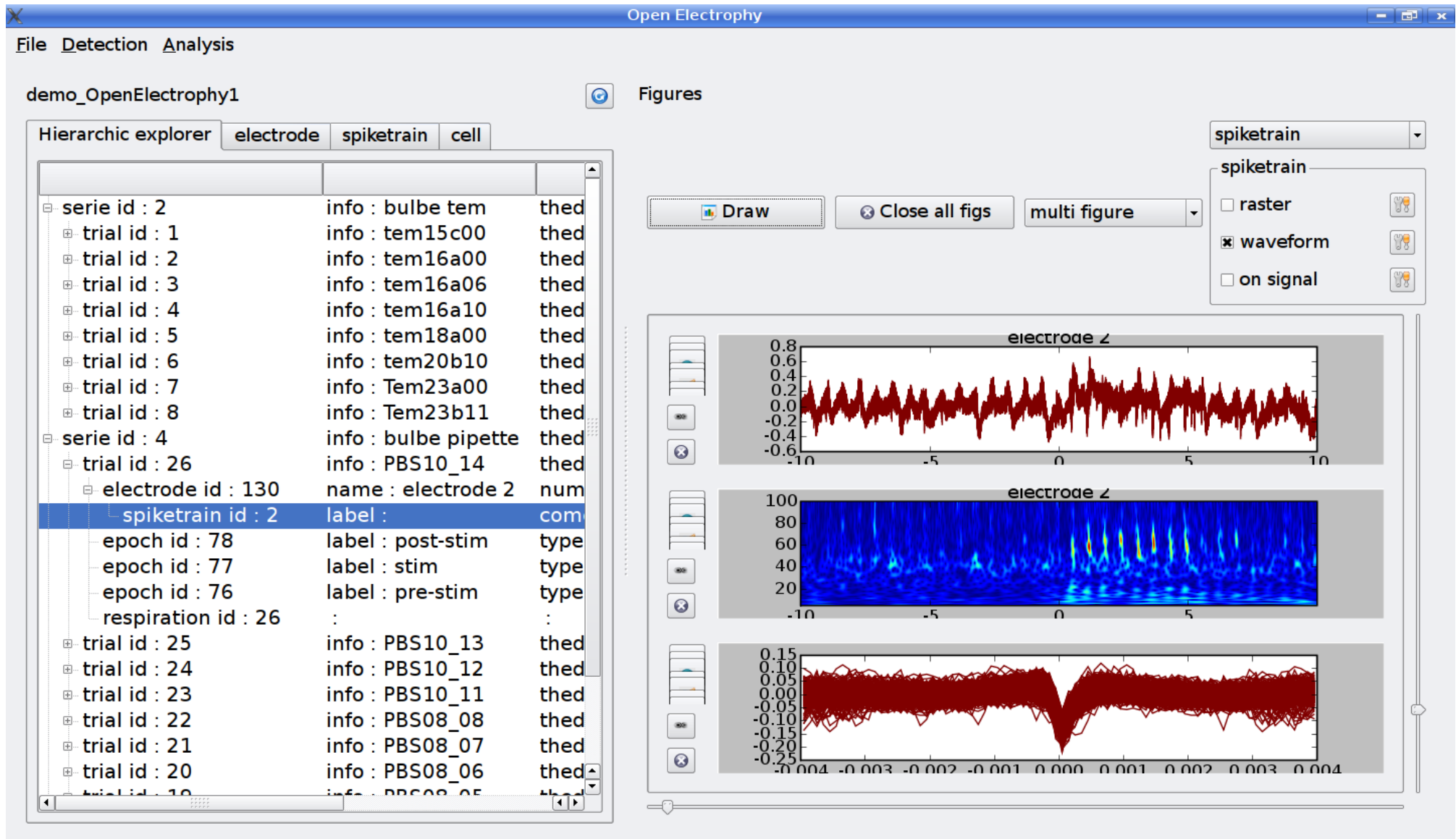
Select all electrode nb 5  
between 6<sup>th</sup> of june and 12<sup>th</sup> of  
july :

```
SELECT id_electrode  
FROM electrode , trial  
WHERE  
electrode.id_trial = trial.id_trial  
AND electrode.num_channel =5  
AND trial.thedatetime > 2007-06-10  
AND trial.thedatetime < 2007-07-12
```

take all the gamma  
oscillations :

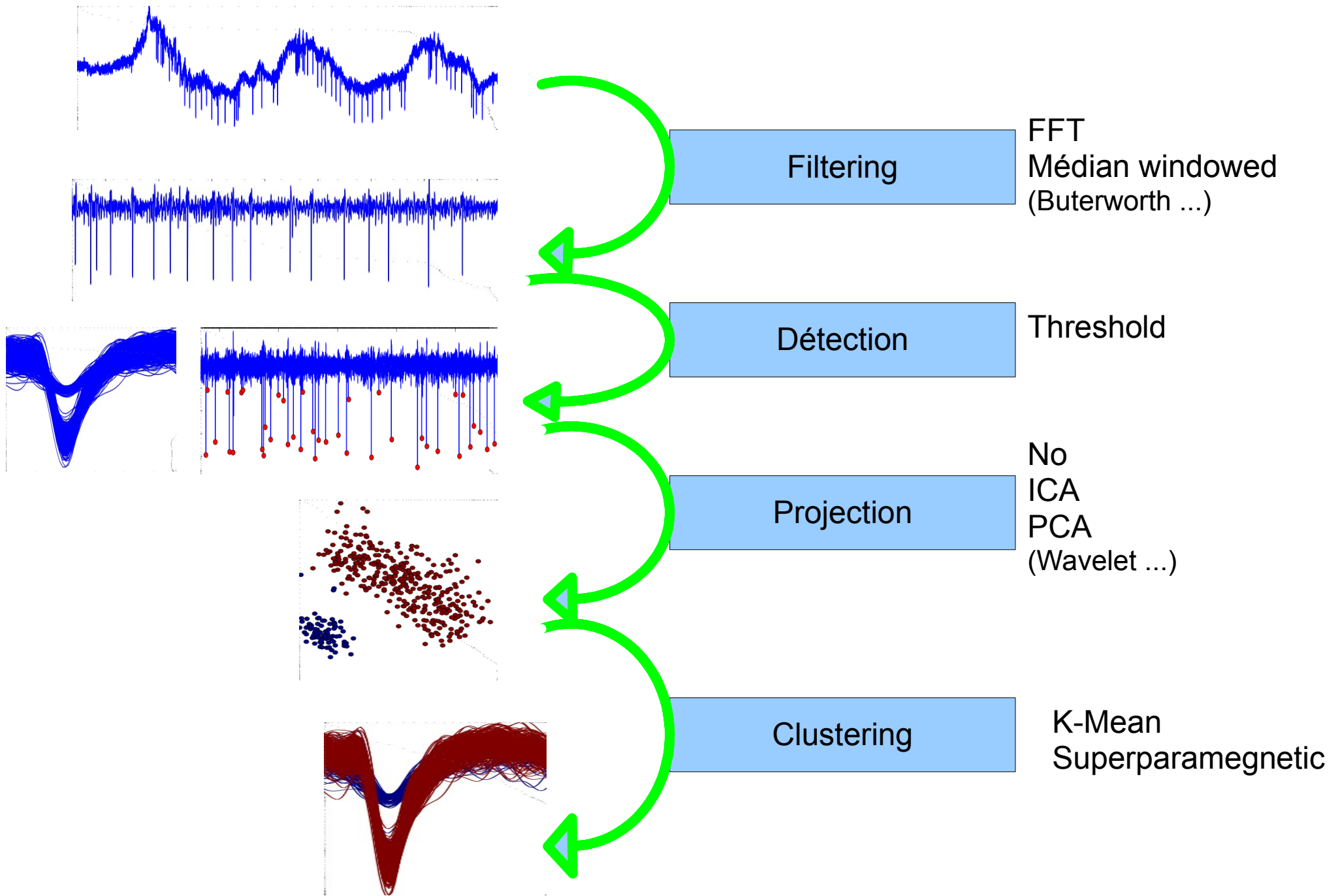
```
SELECT id_oscillation  
FROM oscillation  
WHERE  
oscillation.freq_max > 35  
AND oscillation.freq_max < 90
```

# Graphic User Interface

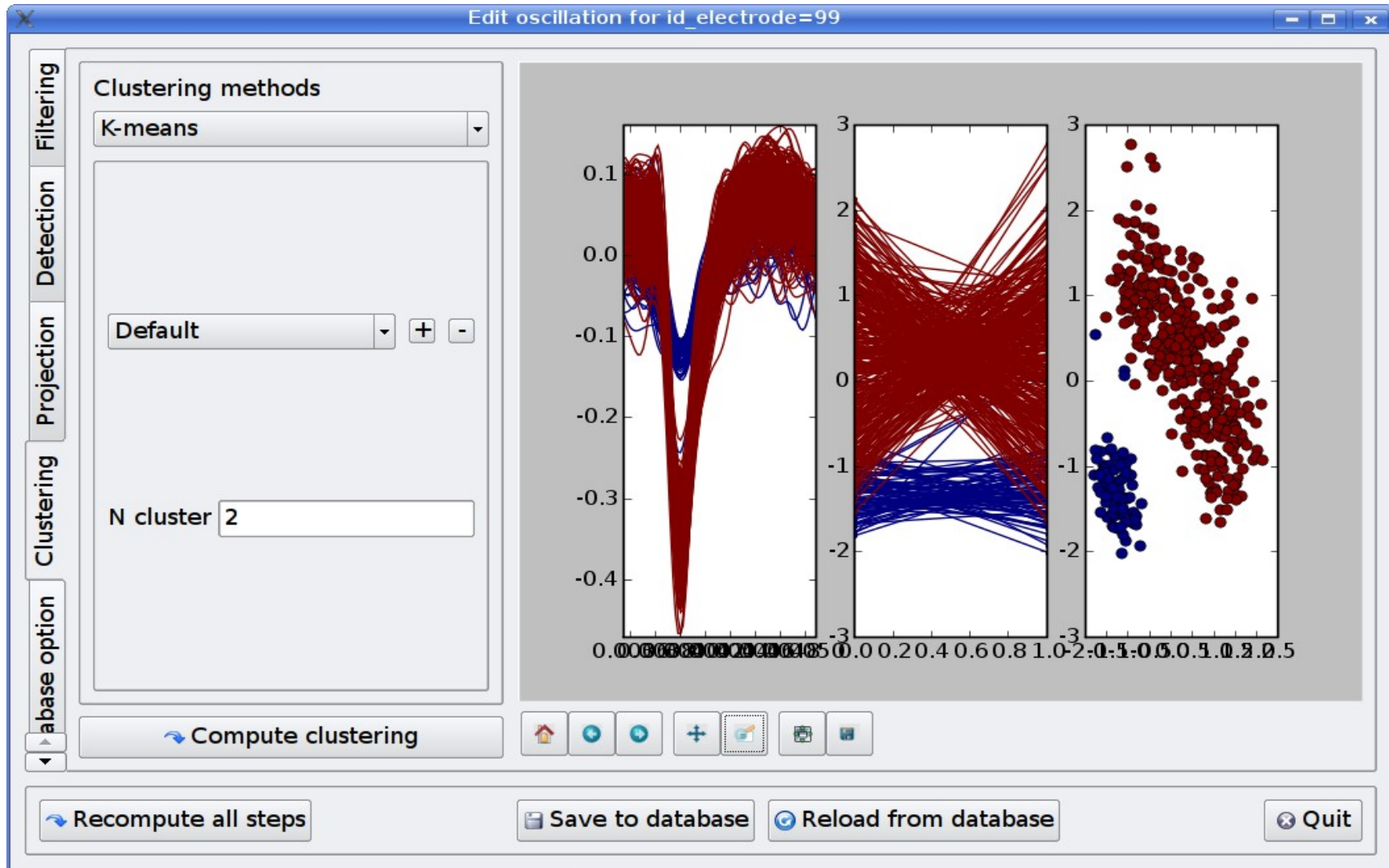


Spikes

# Spike detection and spike sorting : methods compilation



# Spike detection and sorting : Snapshot



Each spike is stored in the database

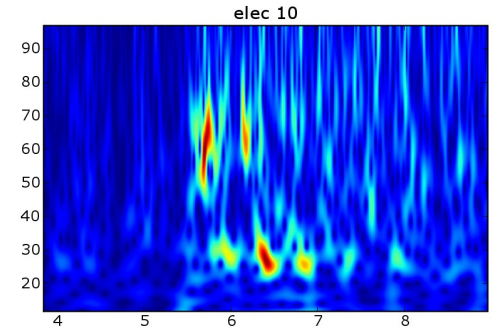
# Oscillations

# LFP Oscillations analysis : new approach

Classical analysis : Morlet wavelet time frequency map  
(scalogram)

New approach :

Use the scalogram for extracting oscillations  
Each oscillation is stored in the database



Avantage : Quantitative study (length, energy, phase , frequencie ...)

Article :

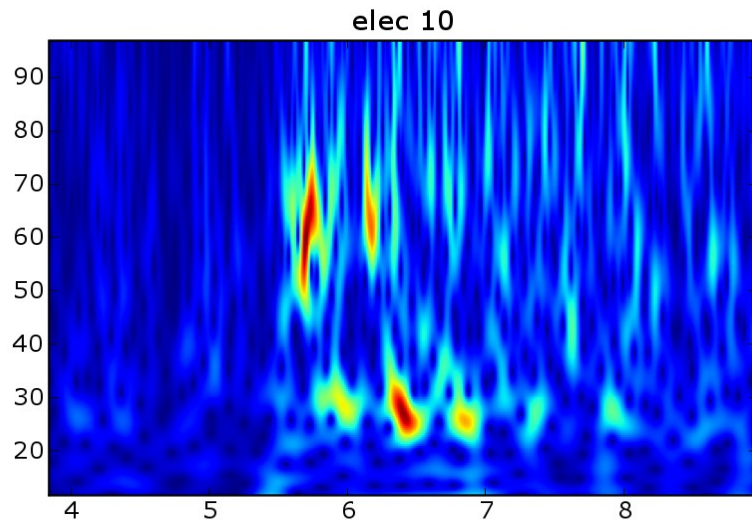
A wavelet-based method for local phase extraction from a multi-frequency oscillatory signal

J Neurosci Methods

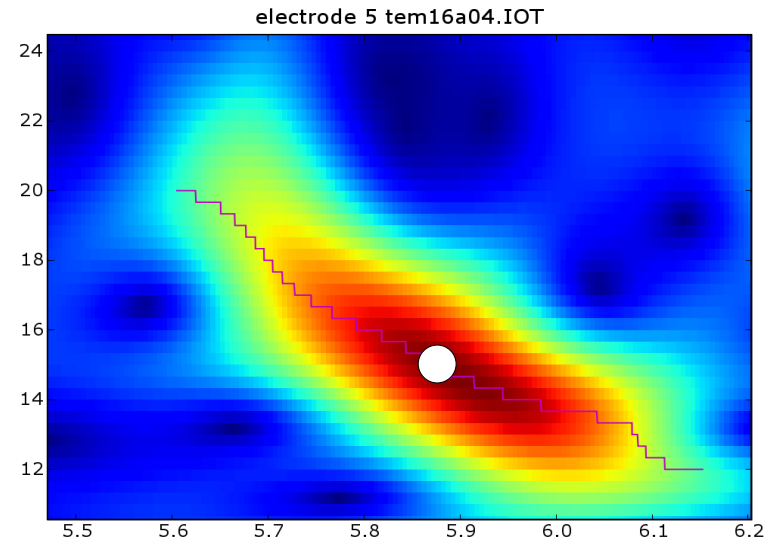
Stéphane G. Roux , Tristan Cenier, Samuel Garcia, Philippe Litaudon, Nathalie Buonviso



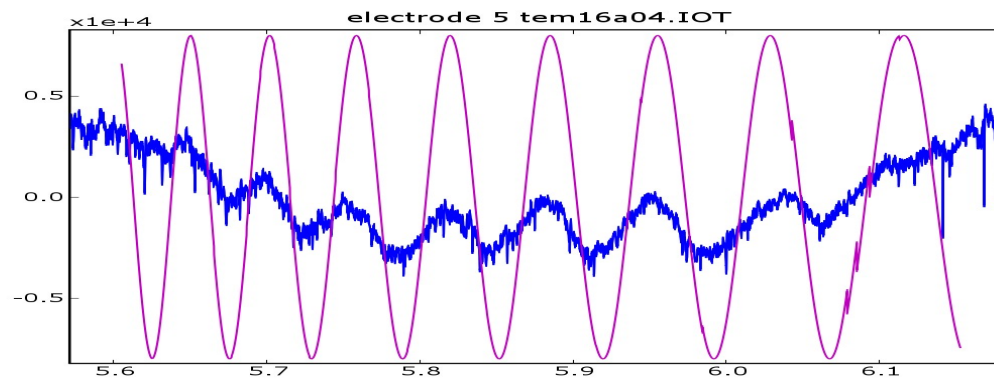
# Oscillation detection : principle



Morlet scalogram : local maxima extraction



Ridge extraction : time-frequency line



Oscillation in time domain

# Oscillation detection : GUI

Edit oscillation for id\_electrode=106

Scalogram Oscillation detection

Parameters for morlet :

Default

Freq start 10.0

Freq stop 90.0

Freq precision 0.5

Time start 0.0

Time stop inf

Time precision 0.0001

f0 2.5

Normalisation 0.0

Parameters for treshold :

Default

Abs treshold nan

Std tresh (if abs tresh is null) 7.0

Window ref.t1 -inf

Window ref.t2 0.0

Window ref.f1 10.0

Window ref.f2 90.0

Re reference tresh Equivalent 0.0479766820071

Parameters clean :

Default

Min nb of cycle 3.0

Eliminate simultaneous

Regroup overlap

Eliminate partial overlap

Recompute selected oscillation

Clean selected oscillation

Show new max

Recompute and add to list

Reset all, recompute and clean

oscillation	time_max	freq_max	val_max
oscilla...	0.55	49.0	0.0757505674445
oscilla...	0.8005	12.0	0.0588404436305
oscilla...	4.8265	61.5	0.0668908109512

0.000 0.008 0.016 0.024 0.032 0.040 0.048 0.056 0.064 0.072

0.8  
0.6  
0.4  
0.2  
0.0  
-0.2  
-0.4  
-0.6  
-0.8

0.3 0.4 0.5 0.6 0.7 0.8

electrode 8

70  
60  
50  
40  
30

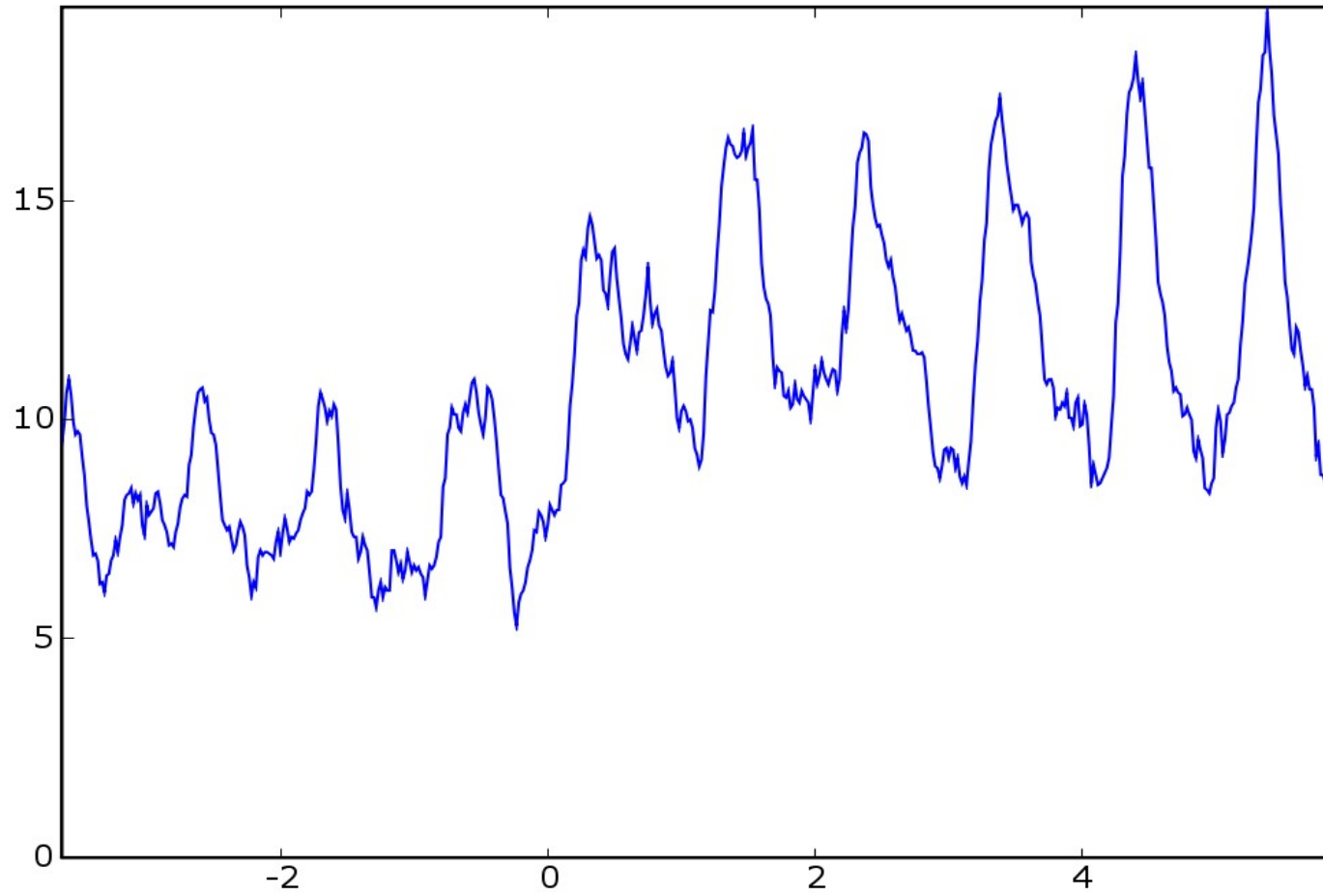
0.3 0.4 0.5 0.6 0.7 0.8

Save to database Reload from database Quit

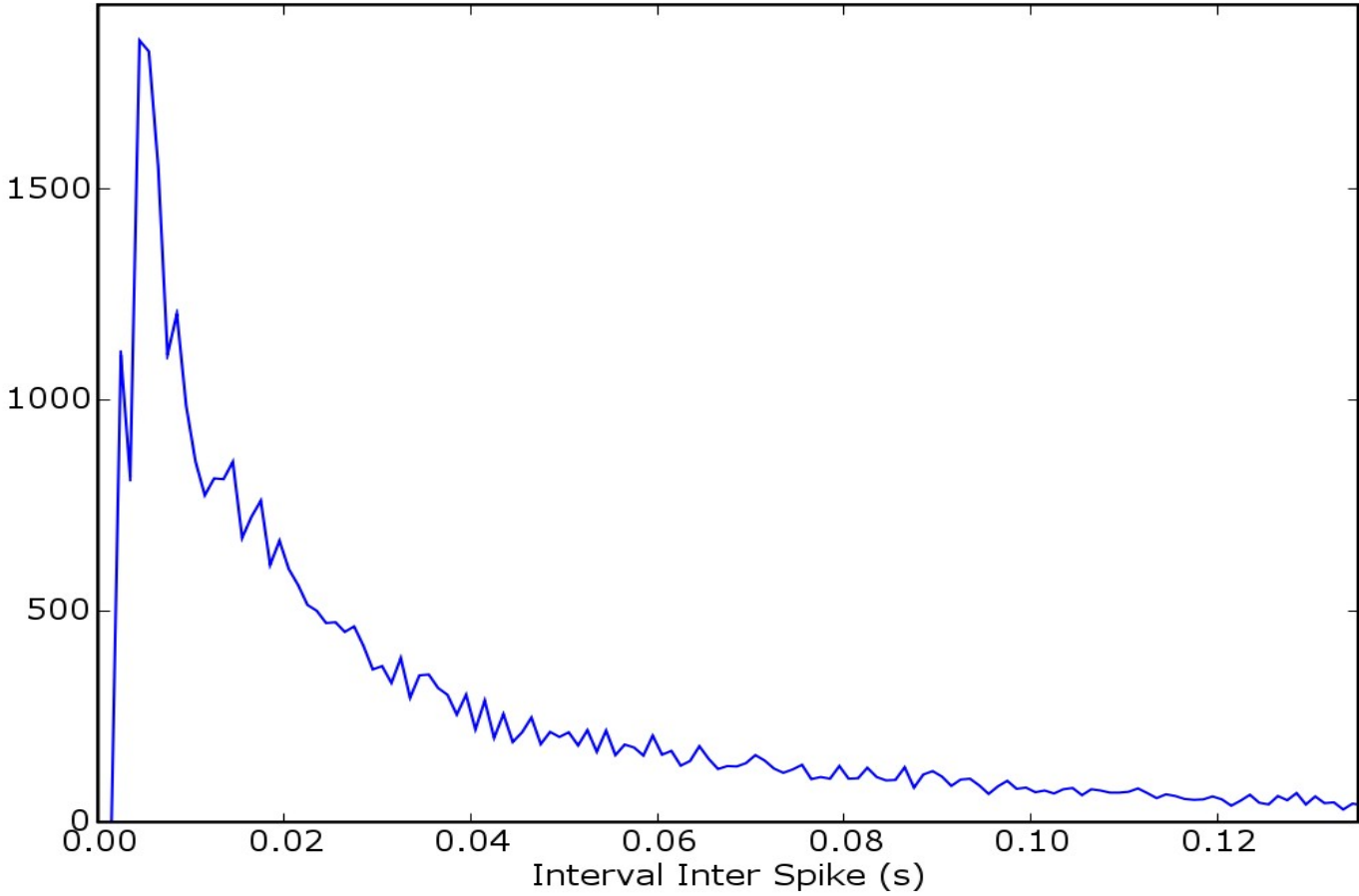
14:55

Some examples of analysis ...

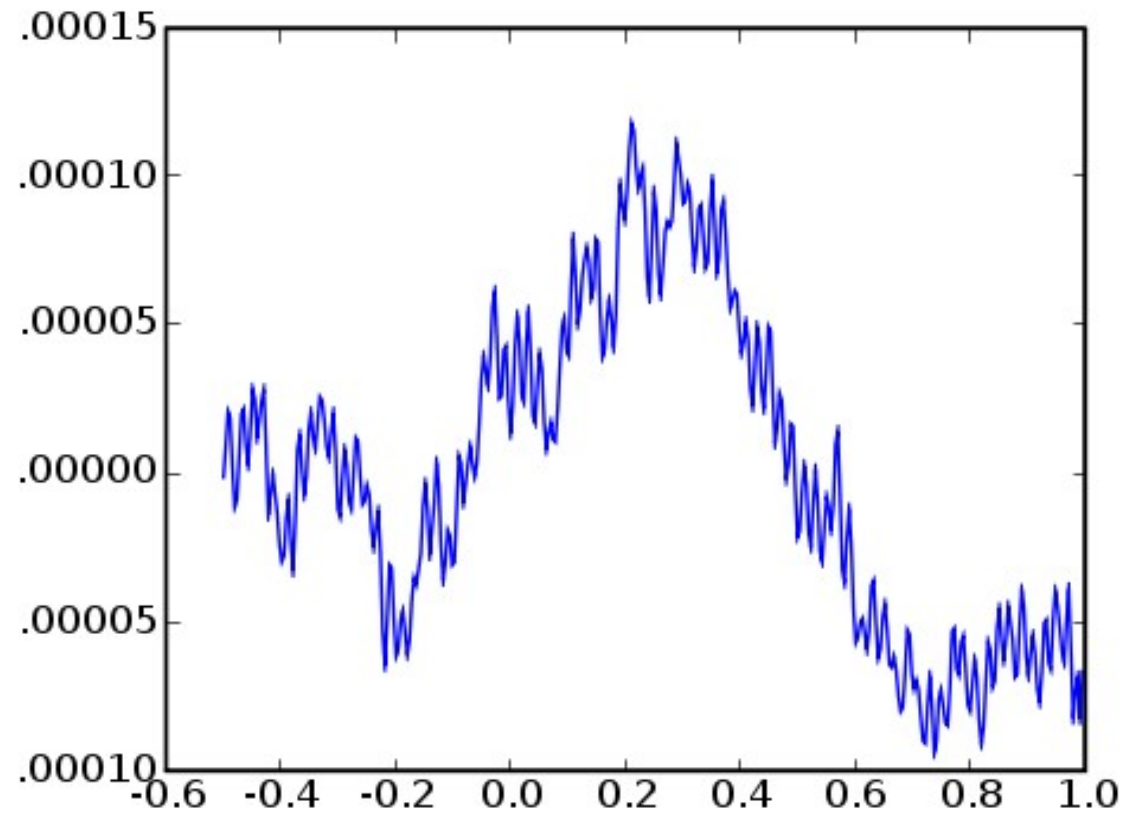
# Instantaneous windowed Spike rate



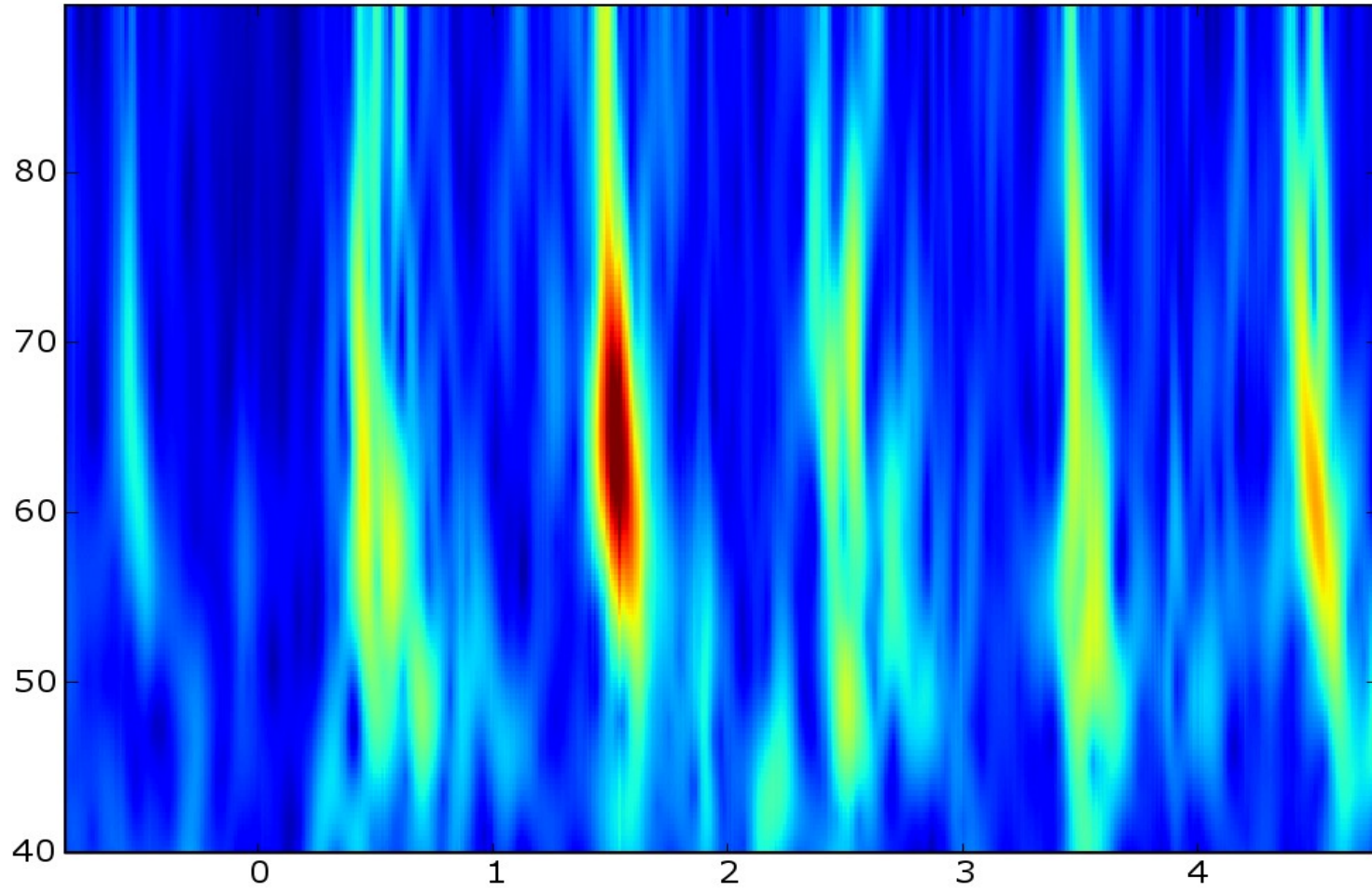
# Interval Inter spike Histogram



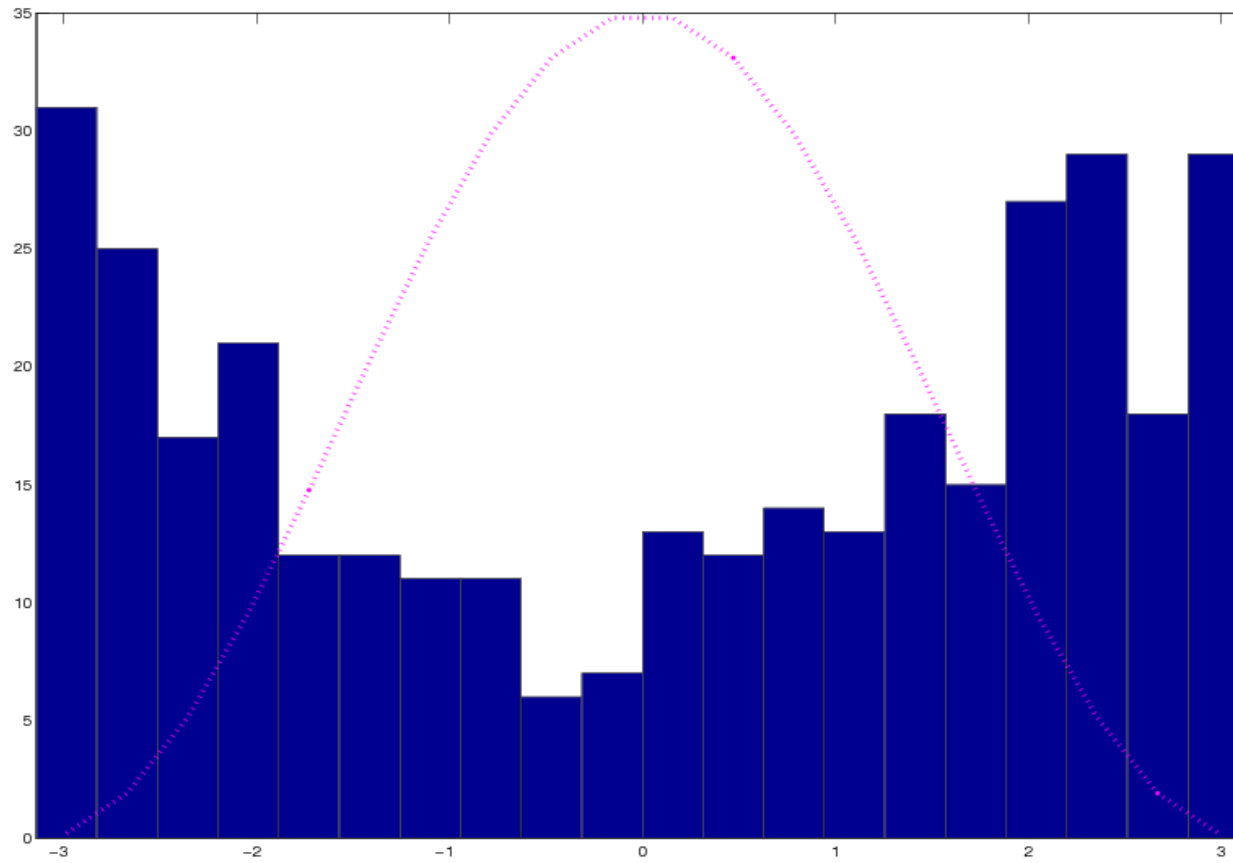
# Signal averaging



# Scalogram averaging

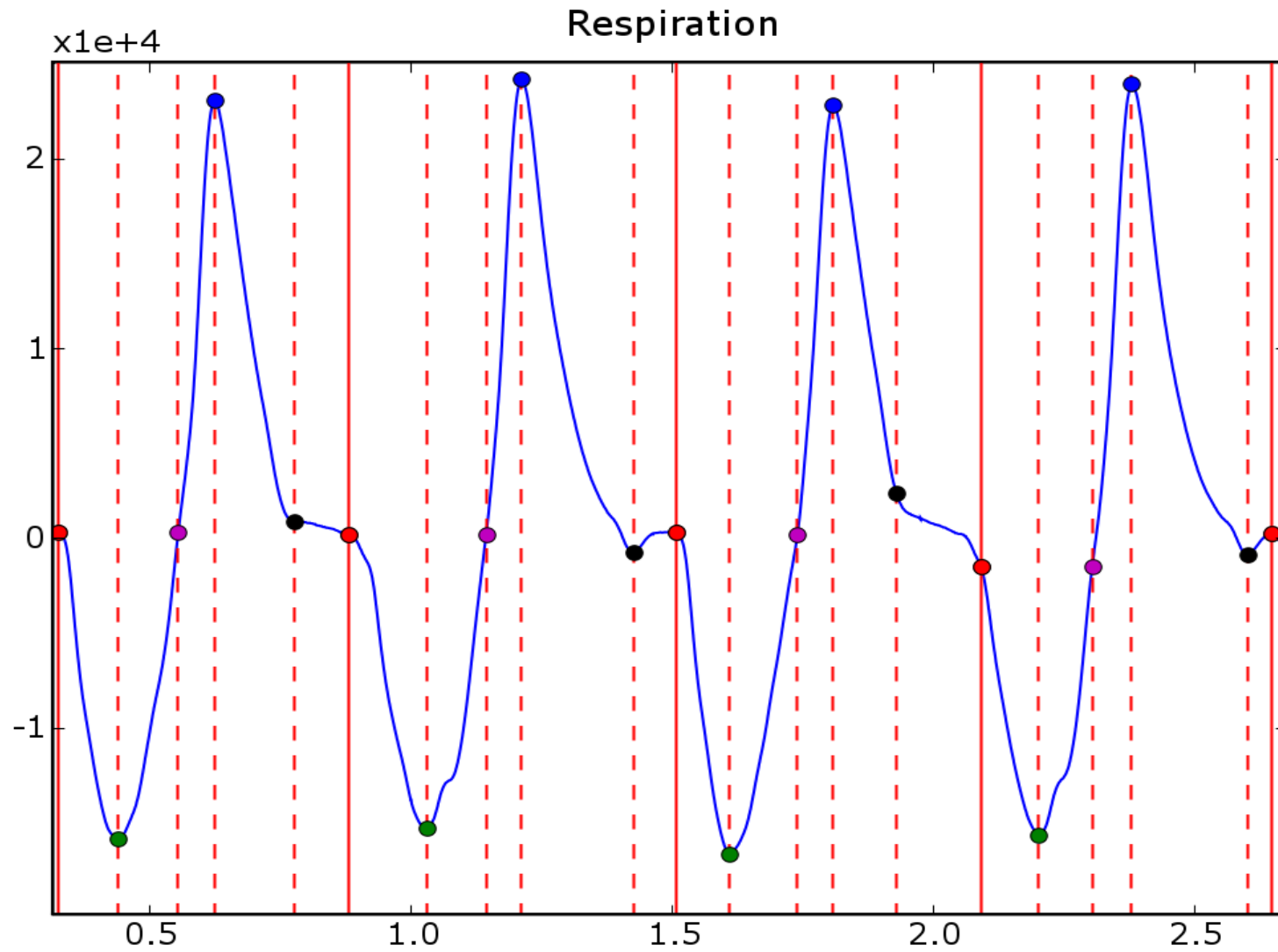


# Spike phase on oscillation





# Respiration detection



Link

<http://neuralensemble.org/trac/OpenElectrophy>

# Internal Code Design

MySQL is great for reading and exploring data but heavy for writing and modifying data

python Class (database\_storage) for :

- simplifying reading/writing/modifying data
- able to store numpy array
- delete recursively hierarchic data
- automagic table creation
- automagic update database structure

For Each table, you subclass and declare all fields.

And you can work immediately!

Example