



HAL
open science

Tools for convulsive seizures and interictal spikes detection

Daniel Diaz-Arce, Anis Ghouma, Pierre Guetschel, Théodore Papadopoulo, Massimo Mantegazza, Fabrice Duprat

► To cite this version:

Daniel Diaz-Arce, Anis Ghouma, Pierre Guetschel, Théodore Papadopoulo, Massimo Mantegazza, et al.. Tools for convulsive seizures and interictal spikes detection. Congrès Labex ICST, Nov 2021, Nantes, France. hal-03540822

HAL Id: hal-03540822

<https://inria.hal.science/hal-03540822>

Submitted on 24 Jan 2022

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

1 Introduction

Epilepsy is a very invalidating and common disorder, and the use of rodent models of epilepsy to test new anti-epileptic treatments is of great interest. The ability to detect epileptic seizures in these models with automated or semi-automated methods is still a major challenge.

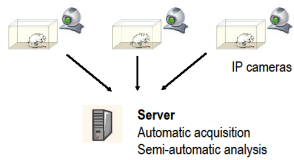
We have developed two new computer-based tools using Python language that,

- 1) enable the noninvasive semi-automated detection of convulsive seizures from videos,
- 2) enable the automatic detection of interictal spikes from Electrocorticogram recordings (ECoG).

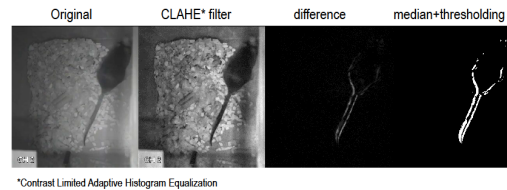
2 Detecting convulsive seizures from videos

Convulsive seizures are very serious phenotypic manifestations occurring in severe forms of epilepsy, they appear as clonic or tonic-clonic seizures with wild jumping. Our method is based on motion and its power spectral analysis calculated from video recording of freely moving mice. It helps the user to easily spot convulsive seizures within hundreds of hours of video recordings.

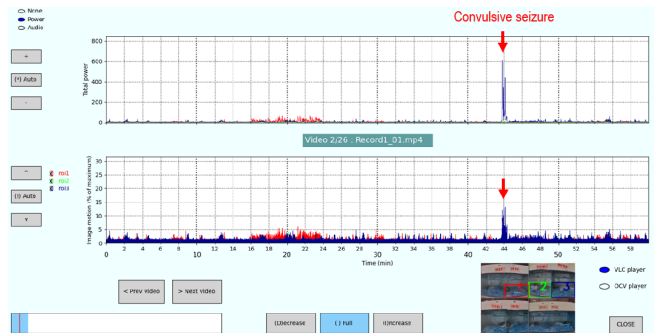
3 Non-invasive video-recording



4 Semi-automated analysis: image motion calculation



5 Display of image motion and its spectral power



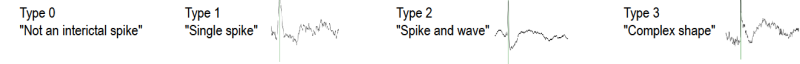
The main advantages of our method are 1) its noninvasiveness, the animals can be recorded in normal cages with minor modifications, 2) the high speed of the analysis with a good accuracy (88,8% accuracy in comparison to full video visual inspection, 93,2% accuracy in comparison to ECoG analysis), 3) the ability to acquire and analyze a very large number of files, thus long video lasting recordings, 4) the easy to use graphical interfaces, and 5) its cost-effectiveness

6 Detecting interictal spikes from ECoG signal using AI

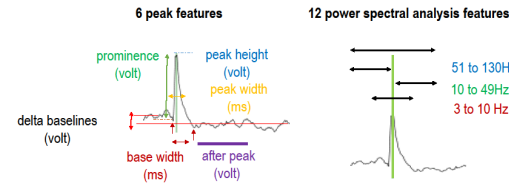
Interictal spikes is a typical feature of ECoG recordings and they are overrepresented in epileptic animals. The spikes are very short peaks (few tens of ms) and hence hard to spot within days of recordings. We have used a random forest classifier trained with 7000 manually classified peaks and using 18 peaks features to automatically classify thousands of peaks over long periods of acquisition.

7 Defining spikes shapes and features to train a classifier

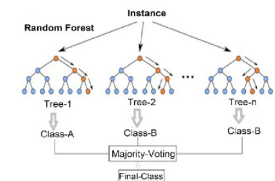
Spikes shapes



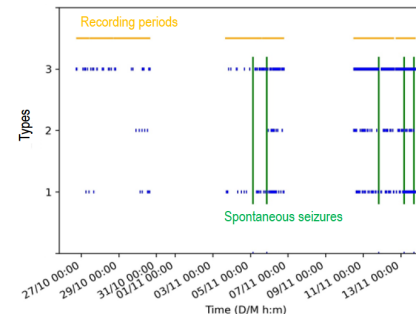
Spikes features



Training of a random forest classifier

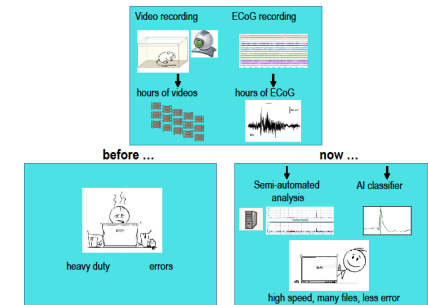


8 Using the classifier



When comparing the classifier results with manually classified spikes we obtained more than 80% accuracy

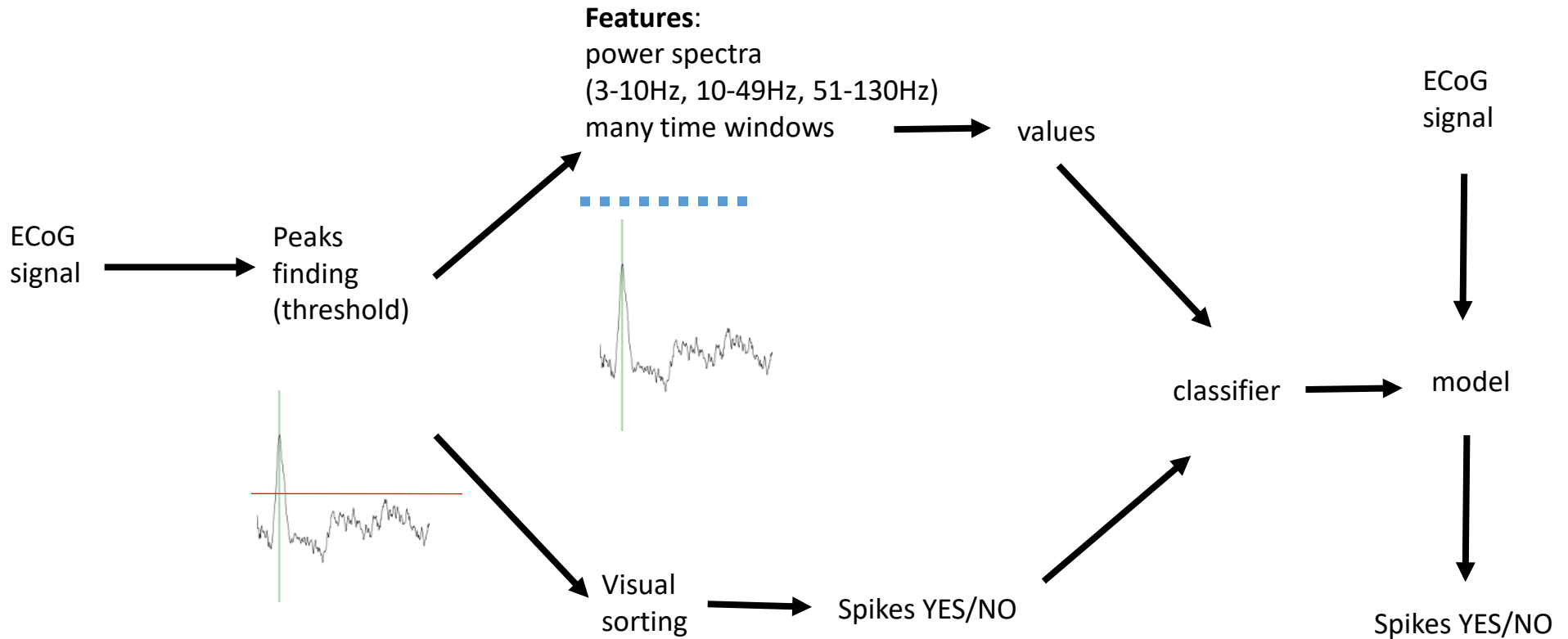
9 Conclusion



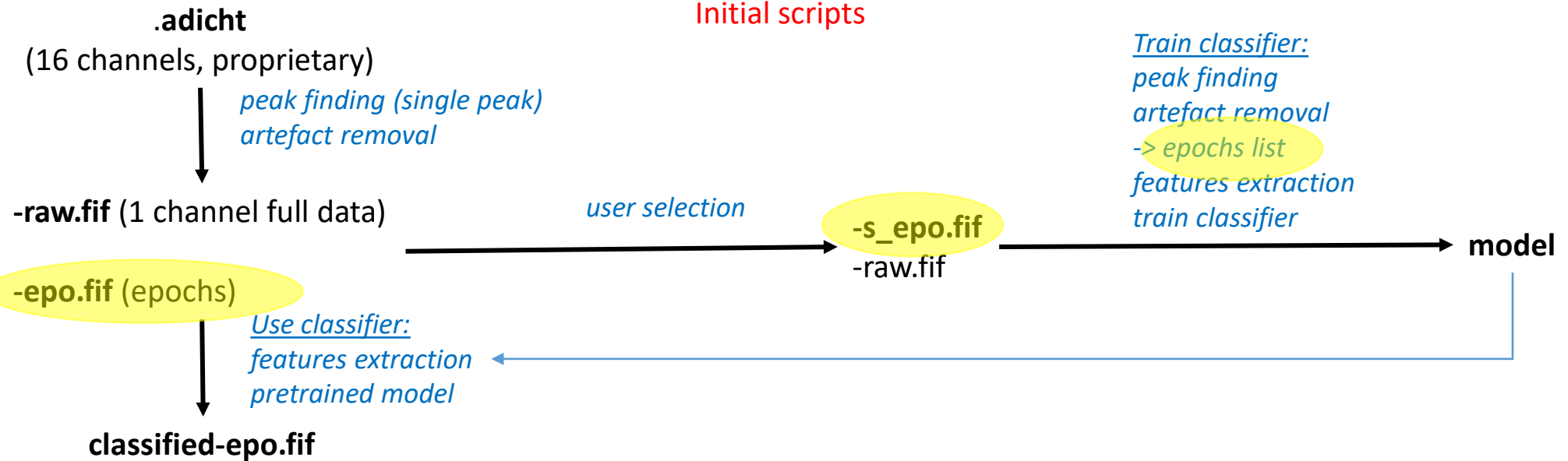
These new tools enable the analysis of hundreds of hours of video and ECoG recordings and give way to higher throughput analysis of our mice epileptic models.

Classifier for Interictal spikes search

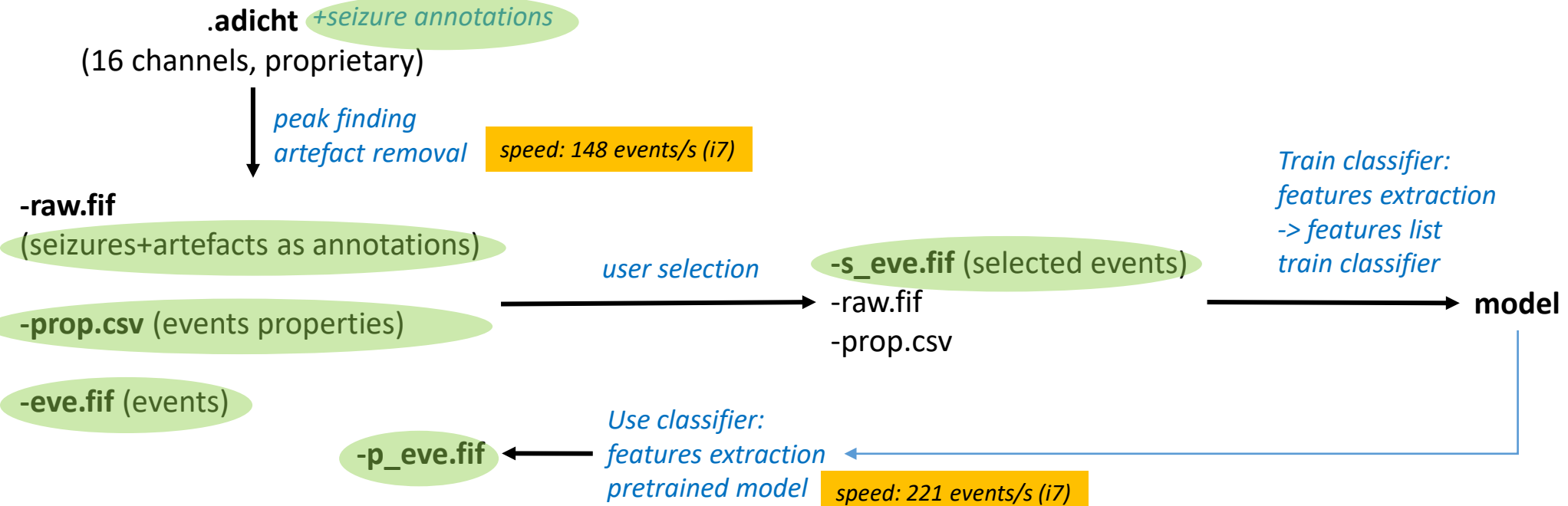
Initial scripts Pierre Guetschel, Théo Papadopoulo INRIA



Initial scripts



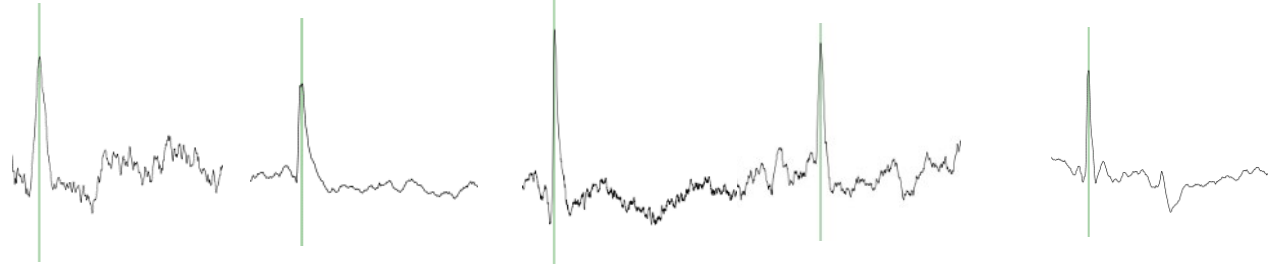
Modified scripts



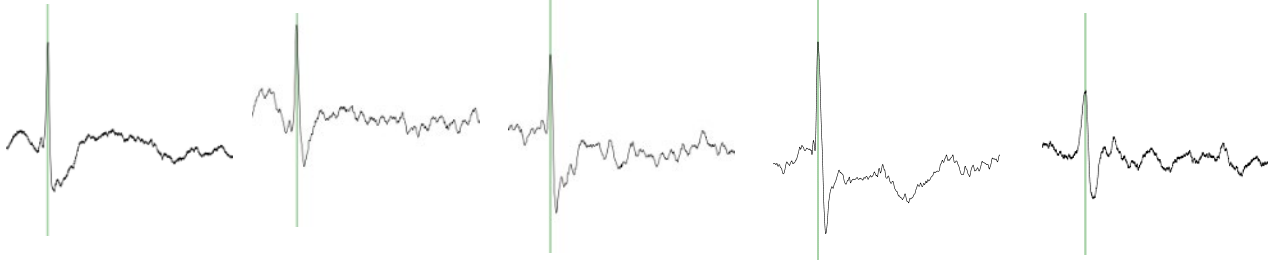
Redefining spikes types

Type 0 = not an interictal spike

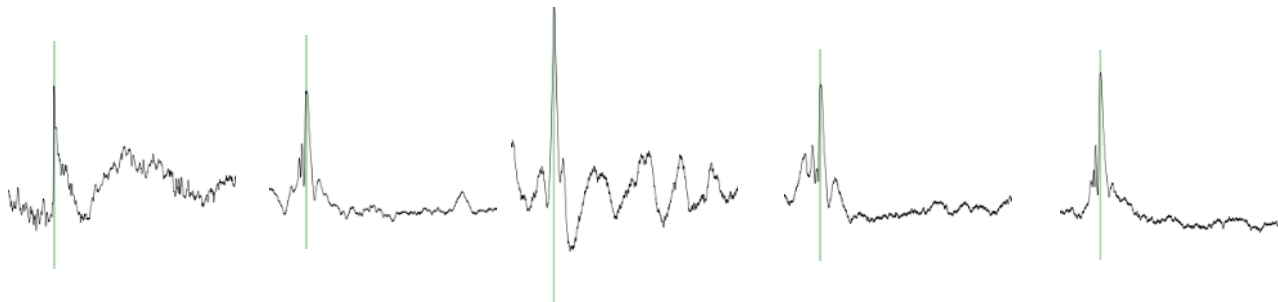
Type 1
"single spike"



Type 2
"spike and wave"



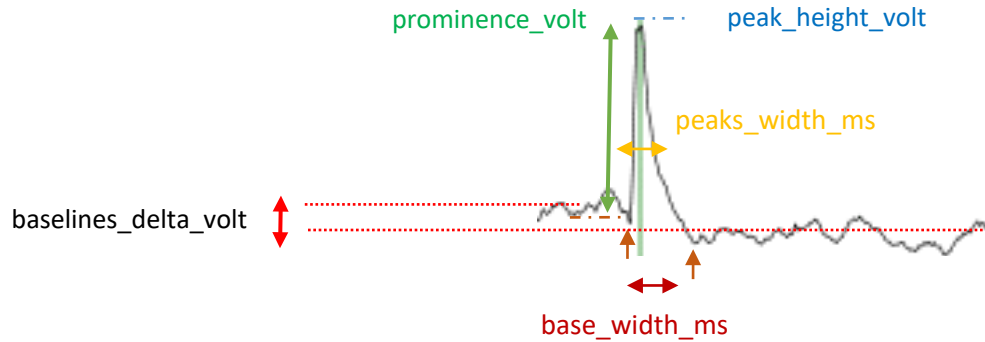
Type 3
"others"



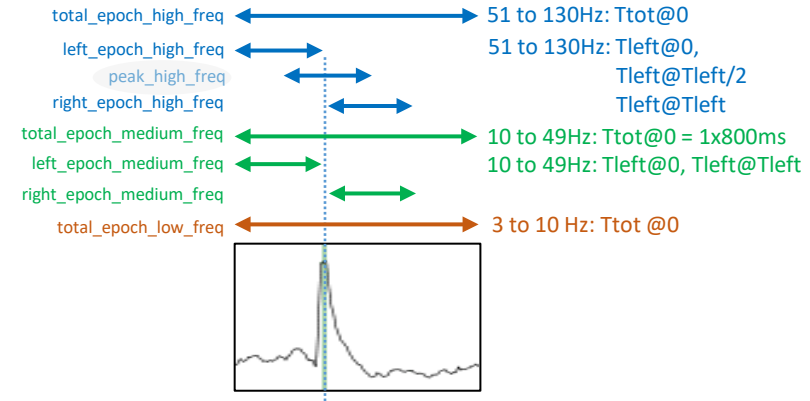
Refining features

13 features model =

5 peak features +



8 power spectral analysis optimised features



18 features model =

+ 1 peak feature



8 features model (peak centered)

+ last 5 spectral features



adicht_analysis.py

find events automatically
(if asked exclude artefact zones),
add seizure + artefacts as annotations

*adicht_functions, analysis_peaks, analysis_clusters,
channel_select, convert_to_matlab*

Input: `.adicht`
*channel, time selection
analysis type*
Return: `-raw.fif (+annotations)`
`-eve.fif (all type 1)`
`-prop.csv`
`0analysis_params.json`

show_models.py

display classifiers models parameters

Input: get all model files (.pkl)
Return: parameters of each model

raw_viewer.py

display raw + events as annotations

Input: `-raw.fif (+annotations)`
`-s_eve`
`0analysis_params.json`
Return: plot raw (mne)

Files types:
`-raw.fif` : signal in volts vs samples
`+annotations` : start and duration in sec, event names (bad artefacts, seizure)
`-eve.fif` : automatically detected events (index in samples, events index)
`-prop.csv` : events properties from automatic detection (width, ...)
`-s_eve.fif` : manually selected events (index in samples, events index)
`-p_eve.fif` : classifier predicted events (index in samples, events index)
`0analysis_params.json`: peak find, artefact correction, epochs creation parameters
`00mice_names_channels.json`: list of mice names and channel(s) for concatenation

add_annotations

events_modifier

events_bulk typer.py

bulk display (mne)
manual events classifying

Input: types names
`-raw.fif`
any eve.fif
`0analysis_params.json`
(-prop.csv)
Return: plot x epochs (mne) for each type
pyplot epochs overlay by type
`-s_eve.fif`

events_viewer.py

display events as mne epochs

Input: types names
`-raw.fif`
any eve.fif
`0analysis_params.json`
(-prop.csv)
Return: plot epochs (mne)
pyplot epochs overlay by type

epochs_viewer

display_artefact

start_train_classifier.py

train a classifier with manual events

Input: new model name
`-raw.fif`
`-s_eve.fif`
`-prop.csv`
`0analysis_params.json`
extract_features.py
Return: `model_name.pkl`
`0extracted_labels.pkl`
`0extracted_features.pkl`
plot confusion matrix

events_datetime_stats.py

display events against datetimes and stats

Input: `-raw.fif (+annotations)`
events file
Mouse&channel(s) names or json listing
Return: events + annotations / datetime
as pyplot + csv
Statistics as csv

start_events_classifiers.py

automatic events classifier

Input: existing model (.pkl)
`-raw.fif`
`-eve.fif`
`-prop.csv`
extract_features.py
Return: `-p_eve.fif`
stats classification results

events_retyper.py

display one by one (pyplot)
manual events classifying
after p- / s-eve comparison or not

Input: types names
`-p_eve.fif`
`-raw.fif`
`-s_eve.fif or not`
`0analysis_params.json`
Return: plot 1 epoch (pyplot)
button to select type
pyplot epochs overlay by type
`-s_eve.fif`

(or events_bulk typer.py)

Testing of various features

All trained with 5635 events and tested with 627

Model name	Nb of features	True type 0 events	True type 1 events	True type 2 events	True type 3 events	Total accuracy (%)
ef8 : Peak centered	8	43/63 68,3%	83/165 50,3%	73/105 69,5%	186/294 63,3%	61,4
ef13 : Peak and surrounding optimised	13	37/58 63,8%	84/164 51,2%	73/106 68,9%	210/299 70,2	64,4
ef18 : Peak and surrounding, all frequencies, afterpeak	18	45/63 71,4%	95/165 57,6%	84/105 80%	246/294 83,7%	75

ef8

Features	Importances
peaks_widths_ms	0.17
peak_low_freq	0.14
peak_medium_freq	0.13
bases_widths_ms	0.13
peak_high_freq	0.12
prominences_volt	0.11
peaks_heights_volt	0.1
baselines_delta_volt	0.09

ef13

Features	Importances
left_epoch_high_freq	0.12
peaks_widths_ms	0.11
peak_high_freq	0.09
total_epoch_medium_freq	0.08
left_epoch_medium_freq	0.08
prominences_volt	0.08
right_epoch_high_freq	0.07
total_epoch_low_freq	0.07
total_epoch_high_freq	0.07
bases_widths_ms	0.06
peaks_heights_volt	0.06
baselines_delta_volt	0.06
right_epoch_medium_freq	0.05

ef18

Features	Importances
afterpeak_volt	0.18
peaks_widths_ms	0.08
left_epoch_high_freq	0.08
peak_low_freq	0.07
peak_medium_freq	0.07
total_epoch_medium_freq	0.05
left_epoch_medium_freq	0.05
prominences_volt	0.05
right_epoch_high_freq	0.05
peak_high_freq	0.05
total_epoch_high_freq	0.04
right_epoch_low_freq	0.04
peaks_heights_volt	0.03
left_epoch_low_freq	0.03
total_epoch_low_freq	0.03
baselines_delta_volt	0.03
bases_widths_ms	0.03
right_epoch_medium_freq	0.03

Hyperparameters optimization

split random state=range(1, 100, 5), RF classifier random state=range(1, 200, 5):

Bests: accuracy 83,1% : split random_state=21, RF classifier random_state =101

Worse: accuracy 73,2% : split random_state =26, RF classifier random_state =116

NEUROMOD STUDENT : random search, grid search, Bayesian optimization, ... ?

n_estimators *int, default=100* The number of trees in the forest.

max_features {"auto", "sqrt", "log2"}, *int or float, default="auto"* The number of features to consider when looking for the best split

max_depth *int, default=None* The maximum depth of the tree

min_samples_split *int or float, default=2* The minimum number of samples required to split an internal node

min_samples_leaf *int or float, default=1* The minimum number of samples required to be at a leaf node.

criterion {"gini", "entropy"}, *default="gini"* The function to measure the quality of a split

bootstrap, *bool, default=True* Whether bootstrap samples are used when building trees. If False, the whole dataset is used to build each tree.

min_weight_fraction_leaf, *float, default=0.0* The minimum weighted fraction of the sum total of weights (of all the input samples) required to be at a leaf node.

max_leaf_nodes, *int, default=None* Grow trees with max_leaf_nodes in best-first fashion. Best nodes are defined as relative reduction in impurity

min_impurity_decrease, *float, default=0.0* A node will be split if this split induces a decrease of the impurity greater than or equal to this value.

oob_score *bool, default=False* Whether to use out-of-bag samples to estimate the generalization score. Only available if bootstrap=True.

random_state, *int, RandomState instance or None, default=None* Controls both the randomness of the bootstrapping of the samples used when building trees and the sampling of the features to consider when looking for the best split at each node (if max_features < n_features)

warm_start *bool, default=False* When set to True, reuse the solution of the previous call to fit and add more estimators to the ensemble, otherwise, just fit a whole new forest.

ccp_alpha, *non-negative float, default=0.0* Complexity parameter used for Minimal Cost-Complexity Pruning.

max_samples, *int or float, default=None* If bootstrap is True, the number of samples to draw from X to train each base estimator.

Select adicht

```
<210924.adicht> 744_Ch1_1a_ECoG_red (ch1) starting at 24/09/21 16:40:30 for 66h 47min 55s
<210924.adicht> 788_Ch3_1b_ECoG_red (ch3) starting at 24/09/21 16:40:30 for 66h 47min 55s
<210924.adicht> 789_Ch5_2a_red (ch5) starting at 24/09/21 16:40:30 for 66h 47min 55s
<210924.adicht> 789_Ch6_2_black (ch6) starting at 24/09/21 16:40:30 for 66h 47min 55s
<210924.adicht> 790_Ch7_2b_red (ch7) starting at 24/09/21 16:40:30 for 66h 47min 55s
<210924.adicht> 790_Ch8_2b_black (ch8) starting at 24/09/21 16:40:30 for 66h 47min 55s
<210924.adicht> 828_Ch9_3a_red (ch9) starting at 24/09/21 16:40:30 for 66h 47min 55s
<210924.adicht> 828_Ch10_3a_black (ch10) starting at 24/09/21 16:40:30 for 66h 47min 55s
<210924.adicht> 829_Ch11_3b_red (ch11) starting at 24/09/21 16:40:30 for 66h 47min 55s
<210924.adicht> 829_Ch12_3b_black (ch12) starting at 24/09/21 16:40:30 for 66h 47min 55s
```

Select raw

Reset selection lists

Options Save raw file Save epo file Save 2 channels Artefact correction Single peaks Start analysis

Quit Events viewer

IMPORTING <210924.adicht> file

SELECT CHANNEL(S) AND ONE BLOCK

	Block1	File end time:
All channels	Block start time: 24/09/21 16:40:29	27/09/21 11:28:25
<input type="checkbox"/> #1 744_Ch1_1a_ECoG_red	<input type="checkbox"/> 2nd channel 66h 47min 56s (2.0kHz)	
<input type="checkbox"/> #2 744_Ch2_1a_EMG_yellow_green	<input type="checkbox"/> 2nd channel 66h 47min 56s (2.0kHz)	
<input type="checkbox"/> #3 788_Ch3_1b_ECoG_red	<input type="checkbox"/> 2nd channel 66h 47min 56s (2.0kHz)	
<input type="checkbox"/> #4 788_Ch4_1b_EMG_yellow_green	<input type="checkbox"/> 2nd channel 66h 47min 56s (2.0kHz)	
<input type="checkbox"/> #5 789_Ch5_2a_red	<input type="checkbox"/> 2nd channel 66h 47min 56s (2.0kHz)	
<input type="checkbox"/> #6 789_Ch6_2_black	<input type="checkbox"/> 2nd channel 66h 47min 56s (2.0kHz)	
<input type="checkbox"/> #7 790_Ch7_2b_red	<input type="checkbox"/> 2nd channel 66h 47min 56s (2.0kHz)	
<input type="checkbox"/> #8 790_Ch8_2b_black	<input type="checkbox"/> 2nd channel 66h 47min 56s (2.0kHz)	
<input type="checkbox"/> #9 828_Ch9_3a_red	<input type="checkbox"/> 2nd channel 66h 47min 56s (2.0kHz)	
<input type="checkbox"/> #10 828_Ch10_3a_black	<input type="checkbox"/> 2nd channel 66h 47min 56s (2.0kHz)	
<input type="checkbox"/> #11 829_Ch11_3b_red	<input type="checkbox"/> 2nd channel 66h 47min 56s (2.0kHz)	
<input type="checkbox"/> #12 829_Ch12_3b_black	<input type="checkbox"/> 2nd channel 66h 47min 56s (2.0kHz)	
<input type="checkbox"/> #13 827_Ch13_4a_red	<input type="checkbox"/> 2nd channel 66h 47min 56s (2.0kHz)	
<input type="checkbox"/> #14 827_Ch14_4a_black	<input type="checkbox"/> 2nd channel 66h 47min 56s (2.0kHz)	
<input type="checkbox"/> #15 830_Ch15_4b_red	<input type="checkbox"/> 2nd channel 66h 47min 56s (2.0kHz)	
<input type="checkbox"/> #16 830_Ch16_4b_black	<input type="checkbox"/> 2nd channel 66h 47min 56s (2.0kHz)	

DEFINE TIME WINDOW TO ANALYSE (WITHIN SELECTED BLOCK)

Choose start datetime (dd/mm/yy hh:mm:ss): OR use file start datetime

Choose duration (hh:mm): OR up to end of block Save selection

Train classifier

Select new model name & events files, classifier is trained

Model name (_, letters and digits only) ?

default_model

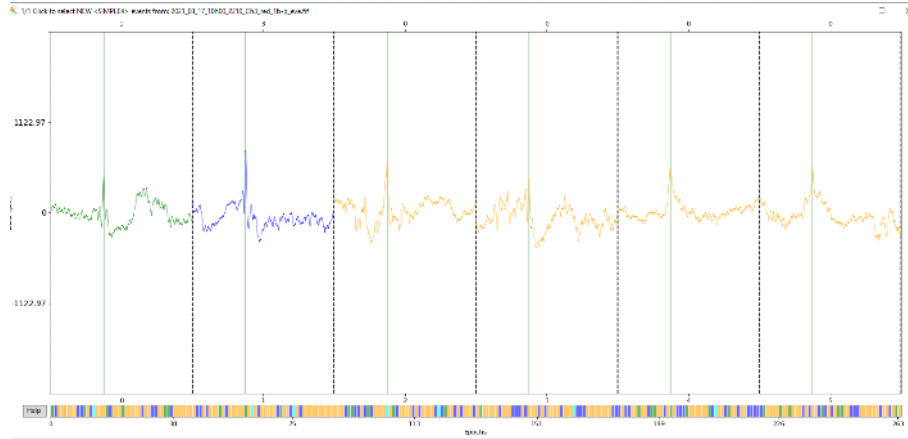
overwrite

Select s_eve files, and start

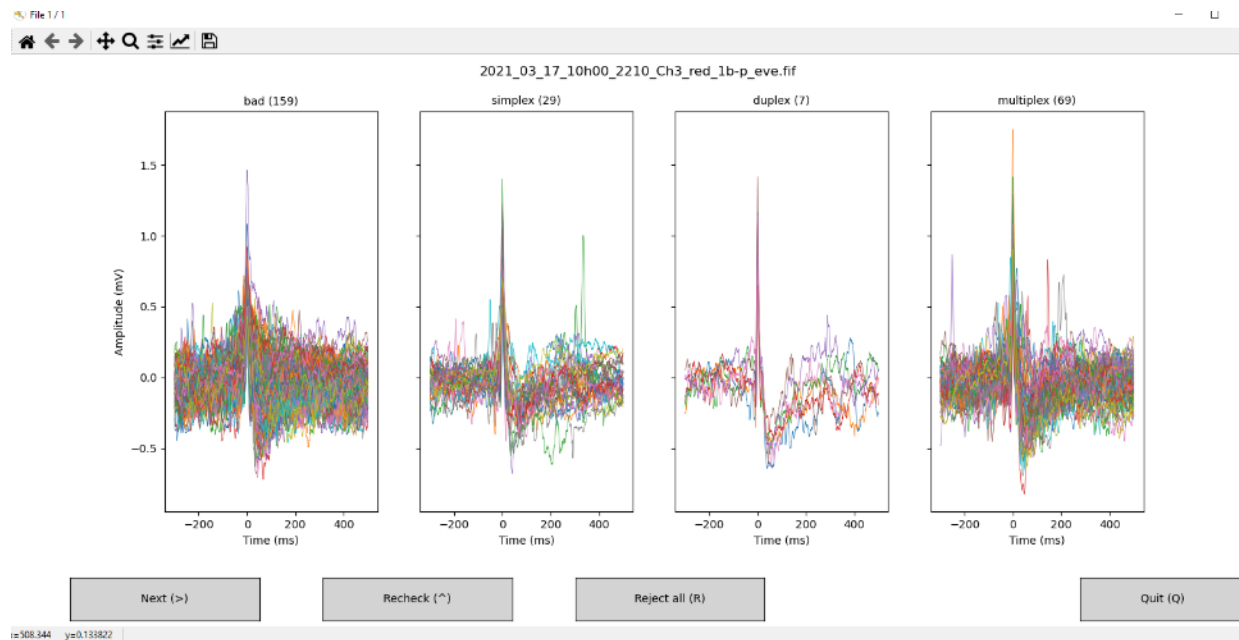
exit

Classifier model is saved to pkl file.

Events bulk typer



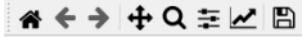
- 1/1 Click to select NEW <DUPLEX> events from: 2021_03_17_10h00_2210_Ch3_red_1b-p_eve.fif
- 1/1 Click to select NEW <MULTIPLEX> events from: 2021_03_17_10h00_2210_Ch3_red_1b-p_eve.fif
- 1/1 Click on NEW BAD EVENTS, to keep <simplex> <duplex> <multiplex> events from: :



Events typer

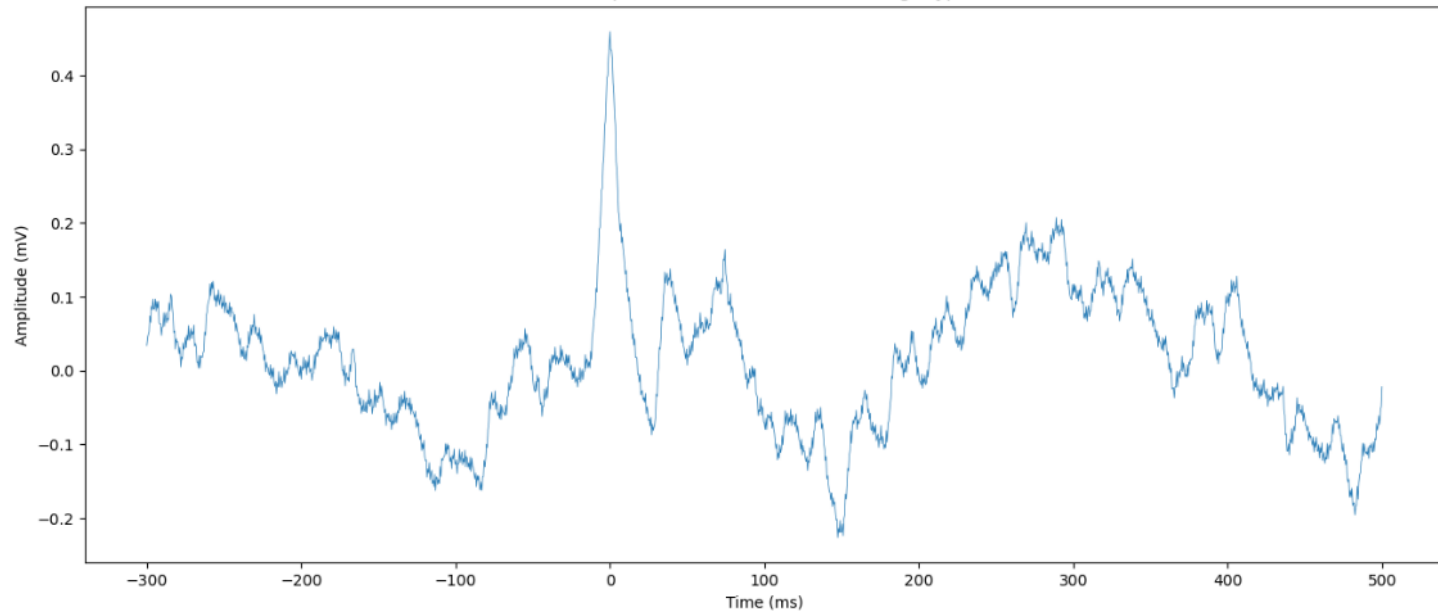
1/6: 210924_16h40_744_Ch1_1a_ECoG_red-s_eve.fif

- □ ×



Event #42 out of 576 (index 41 / 575 in file)

KEEP predicted = 1-SIMPLEX or change type



Keep (>)

0-bad

1-simplex

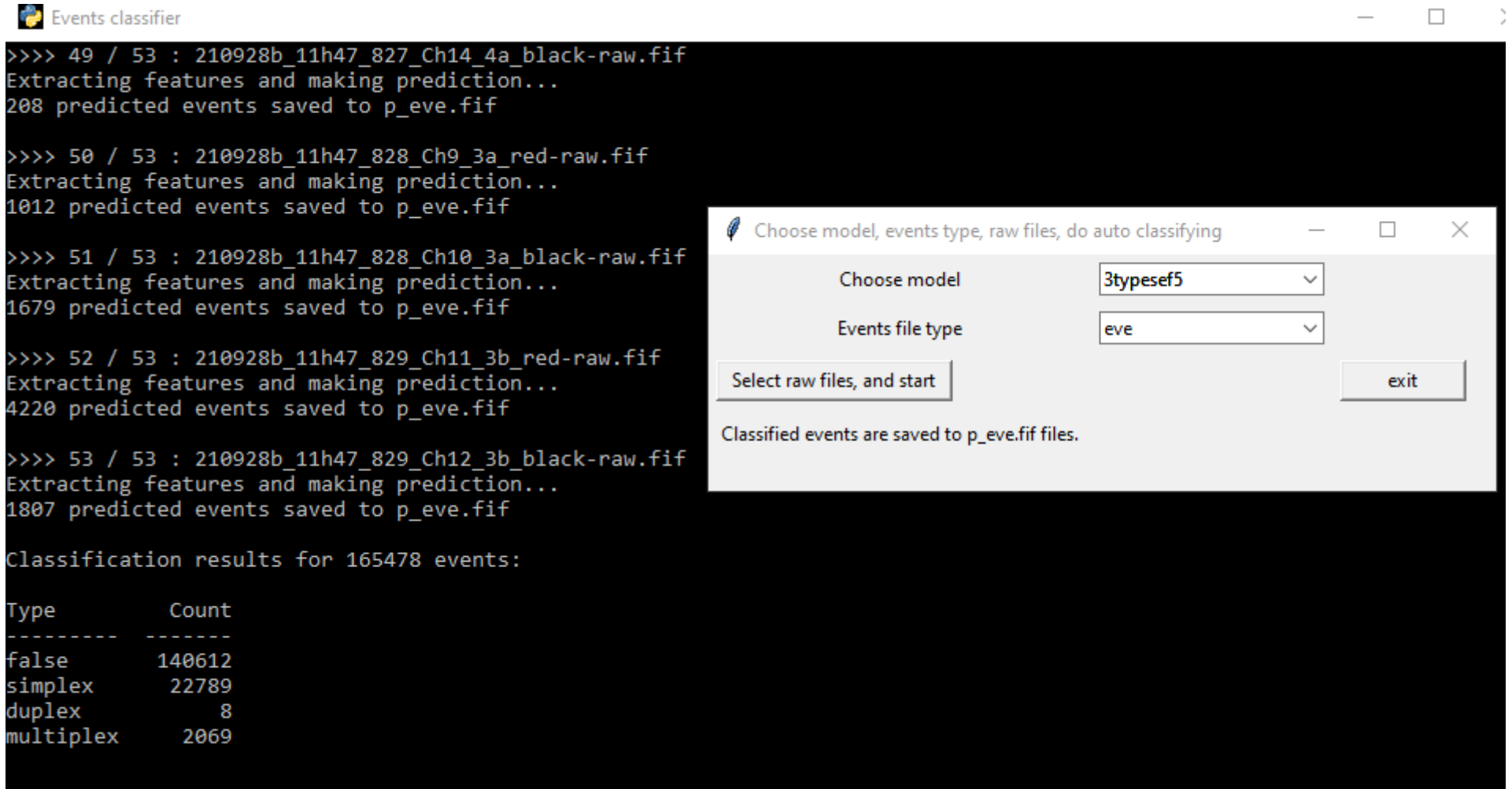
2-duplex

3-multiplex

Next file (N)

Quit (O)

Events classifier



The screenshot shows the 'Events classifier' application window. The main terminal area displays the following text:

```
>>>> 49 / 53 : 210928b_11h47_827_Ch14_4a_black-raw.fif
Extracting features and making prediction...
208 predicted events saved to p_eve.fif

>>>> 50 / 53 : 210928b_11h47_828_Ch9_3a_red-raw.fif
Extracting features and making prediction...
1012 predicted events saved to p_eve.fif

>>>> 51 / 53 : 210928b_11h47_828_Ch10_3a_black-raw.fif
Extracting features and making prediction...
1679 predicted events saved to p_eve.fif

>>>> 52 / 53 : 210928b_11h47_829_Ch11_3b_red-raw.fif
Extracting features and making prediction...
4220 predicted events saved to p_eve.fif

>>>> 53 / 53 : 210928b_11h47_829_Ch12_3b_black-raw.fif
Extracting features and making prediction...
1807 predicted events saved to p_eve.fif

Classification results for 165478 events:
```

Type	Count
false	140612
simplex	22789
duplex	8
multiplex	2069

Overlaid on the terminal is a configuration dialog box titled 'Choose model, events type, raw files, do auto classifying'. It contains two dropdown menus: 'Choose model' set to '3typesef5' and 'Events file type' set to 'eve'. There are two buttons: 'Select raw files, and start' and 'exit'. Below the buttons, it states 'Classified events are saved to p_eve.fif files.'

14 channels of 6811 minutes analysed in 19 min = 4915x

```
#### MODEL 3typesef5.pkl ####
```

```
>> find_peaks_args : {'height': [0.001, 0.003], 'threshold': [0, 4], 'prominence': 0.0005, 'wlen': 600, 'width': [10, 20], 'rel_height': 0.5, 'distance': 600}
```

```
>> find_artefacts_args : {'t_window': 3, 'max_ratio': 3, 'artefact_radius': 3.0, 'min_length': 20.0, 'online': True, 'preheat': 20.0, 'stride': 0.2}
```

```
>> extract_features_args : {'use_total_spectrum': True, 'use_left_spectrum': True, 'use_peak_spectrum': True, 'use_right_spectrum': True, 'use_peak_prop': True, 'freq_ranges': ((3.0, 10.0), (10.0, 49.0), (51.0, 130.0)), 'epoch_durations': (0.3, 0.5)}
```

```
>> clf : RandomForestClassifier(criterion='entropy', n_jobs=4, random_state=42)
```

```
>> tmin : -0.3
```

```
>> tmax : 0.5
```

```
>> sfreq : 2000.0
```

```
>> raw_file 88:
```

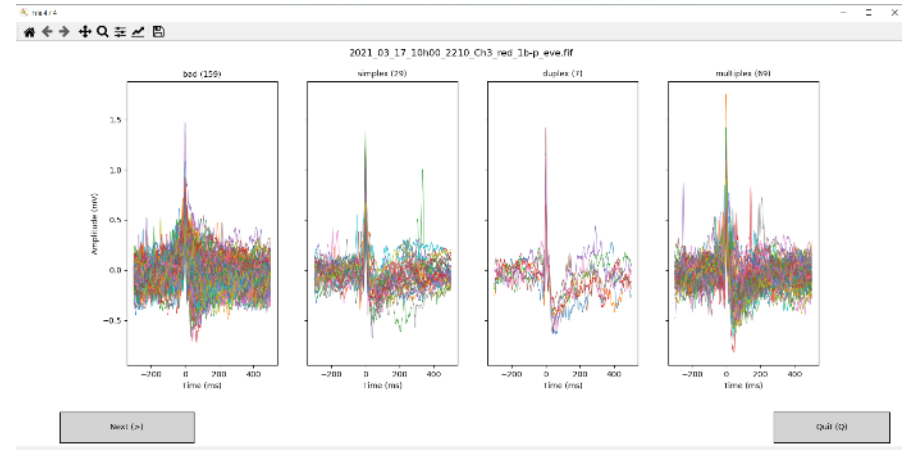
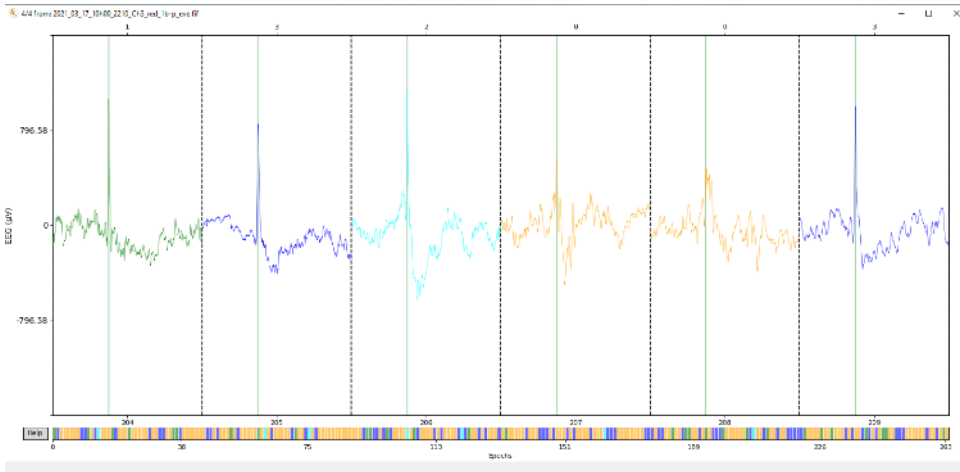
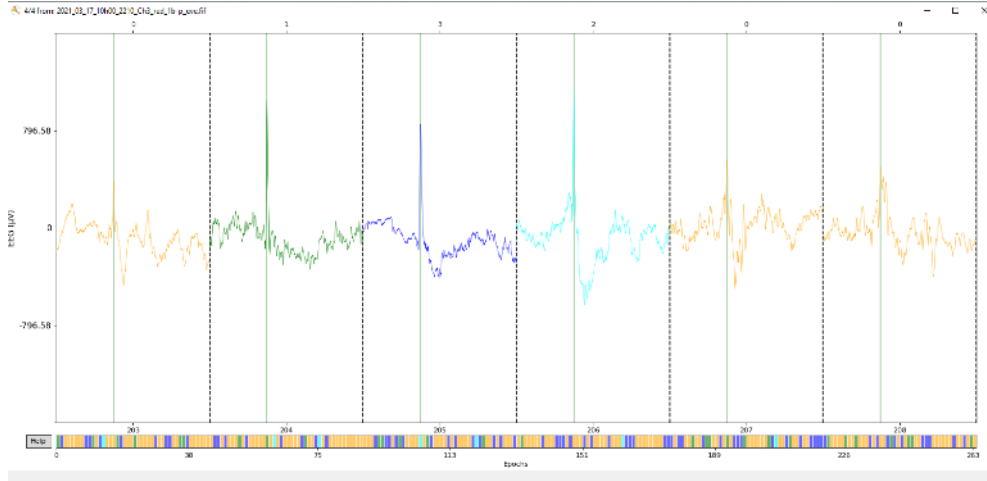
```
>> spike_reference_file 88:
```

```
>> features_names 18:
```

```
['total_epoch_low_freq', 'total_epoch_medium_freq', 'total_epoch_high_freq', 'left_epoch_low_freq', 'left_epoch_medium_freq', 'left_epoch_high_freq', 'peak_low_freq', 'peak_medium_freq', 'peak_high_freq', 'right_epoch_low_freq', 'right_epoch_medium_freq', 'right_epoch_high_freq', 'prominences_volt', 'peaks_heights_volt', 'peaks_widths_ms', 'bases_widths_ms', 'baselines_delta_volt', 'afterpeak_volt']
```

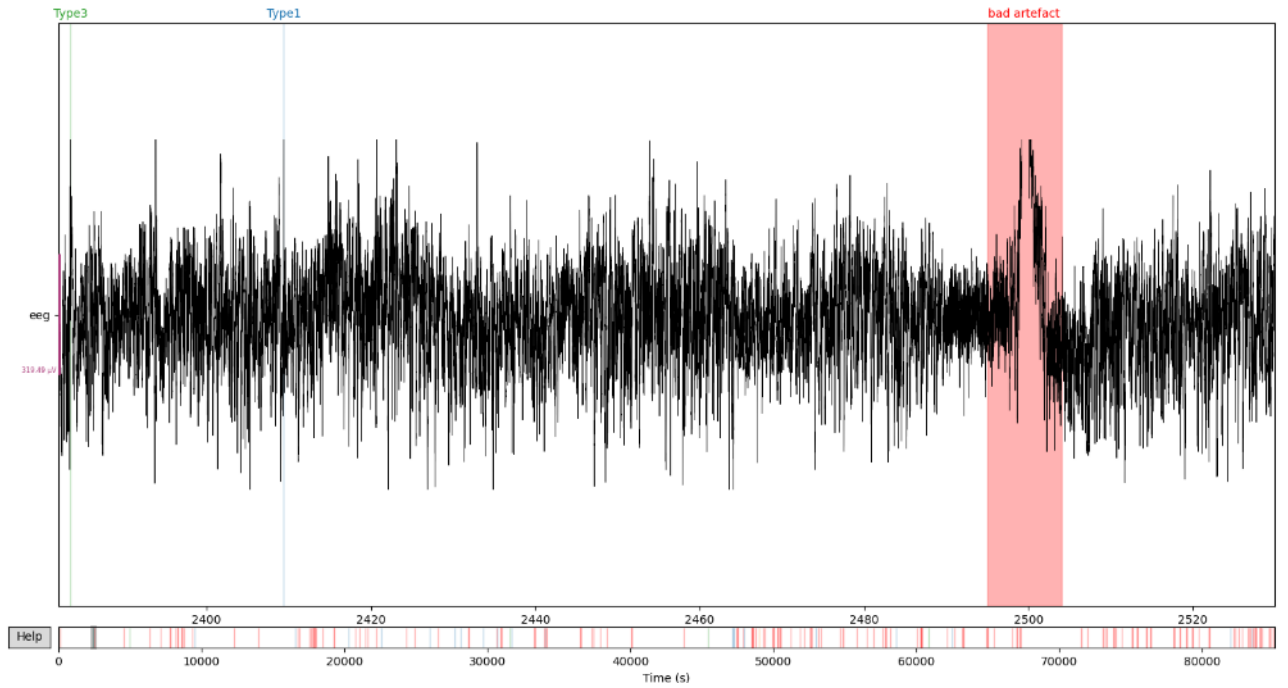
```
-----
Features          Importances
afterpeak_volt    0.18
peaks_widths_ms   0.08
left_epoch_high_freq 0.08
peak_low_freq     0.07
peak_medium_freq  0.07
total_epoch_medium_freq 0.05
left_epoch_medium_freq 0.05
prominences_volt 0.05
right_epoch_high_freq 0.05
peak_high_freq    0.05
total_epoch_high_freq 0.04
right_epoch_low_freq 0.04
peaks_heights_volt 0.03
left_epoch_low_freq 0.03
total_epoch_low_freq 0.03
baselines_delta_volt 0.03
bases_widths_ms   0.03
right epoch medium freq 0.03
```

Events viewer

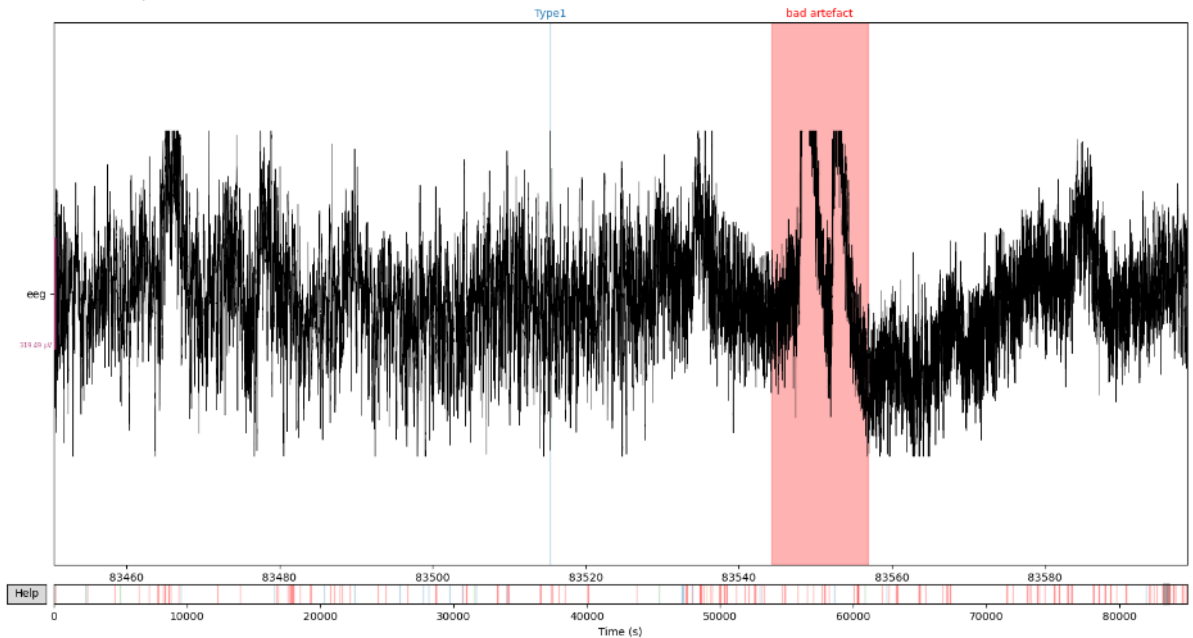


Raw viewer

E:\2021 TIMEM33 data\interictal_analysis\2021_03_17_10h00_2194_Ch5_red_2a-raw.ff



E:\2021 TIMEM33 data\interictal_analysis\2021_03_17_10h00_2194_Ch5_red_2a-raw.ff



Corresponding events file type to use:

Plot each file

Plot mouse&channels concatenated events

OPTION 1) SELECT A BATCH OF RAW FILES OR DIRECTORY, AND ONE MOUSE & CHANNELS NAMES

Mouse name: autofind all files in directory *

Channel name: compulsory for autofind

or optional

or optional

OPTION 2) SELECT JSON FILE CONTAINING MOUSE & CHANNELS NAMES (raw/eve files in the same directory)

← .json file

```

1 {"serie": "TMEM33",
2  "infos": [
3    ["2193", "Ch1", "Ch7", "Ch11"],
4    ["2193", "Ch2", "Ch8", "Ch12"],
5    ["2210", "Ch3", "", ""],
6    ["2210", "Ch4", "", ""],
7    ["2194", "Ch5", "", ""],
8    ["2194", "Ch6", "", ""],
9    ["2245", "Ch7", "", ""],
10   ["2245", "Ch8", "", ""],

```

.csv file

B	C	D	E	F	G	H	I	J
mouse_name	channel_name1	nb_files	duration_sec	nb_seizures	nb_type0	nb_type1	nb_type2	nb_type3
2210	Ch3	5	428261	0	748	198	19	327
2210	Ch4	5	428261	0	725	180	13	293
2245	Ch7	5	428261	0	749	23	0	28
2245	Ch8	5	428261	0	1887	31	1	26
2354	Ch11	3	257525	0	43	25	1	27
2354	Ch12	3	257525	0	309	36	0	29
2356	Ch15	5	428261	0	1209	151	0	90
2356	Ch16							
2358	Ch3	6	581477	1	899	59	1	31
2358	Ch4	7	673552	1	1384	273	1	85
2363	Ch7	7	673552	0	1917	47	4	164
2363	Ch8	7	673552	0	1842	70	4	481
2364	Ch11	7	673552	2	2421	649	0	78
2364	Ch12	7	673552	2	2678	356	0	71
2357	Ch13	7	673552	0	6443	27	0	105
2357	Ch14	7	673552	0	7617	24	0	150
2193	Ch1	4	340848	0	44	4	0	7
2193	Ch2	4	343376	0	113	12	0	5
2194	Ch5	5	428261	0	2387	124	2	49
2194	Ch6	5	428261	0	2395	226	3	69
2195	Ch9	4	340848	0	447	15	0	14
2195	Ch10	4	344939	0	469	9	0	3
2196	Ch13	5	428261	0	6452	49	0	26
2196	Ch14	5	428261	0	4437	38	0	20
2319	Ch1	7	673552	0	4167	88	3	75
2319	Ch2	7	673552	0	2201	60	0	75

Values for 2358_Ch4, from 7 p_eve files

.jpg file

