CIBM Annual Symposium 2022

Campus Biotech, Geneva | 30th November

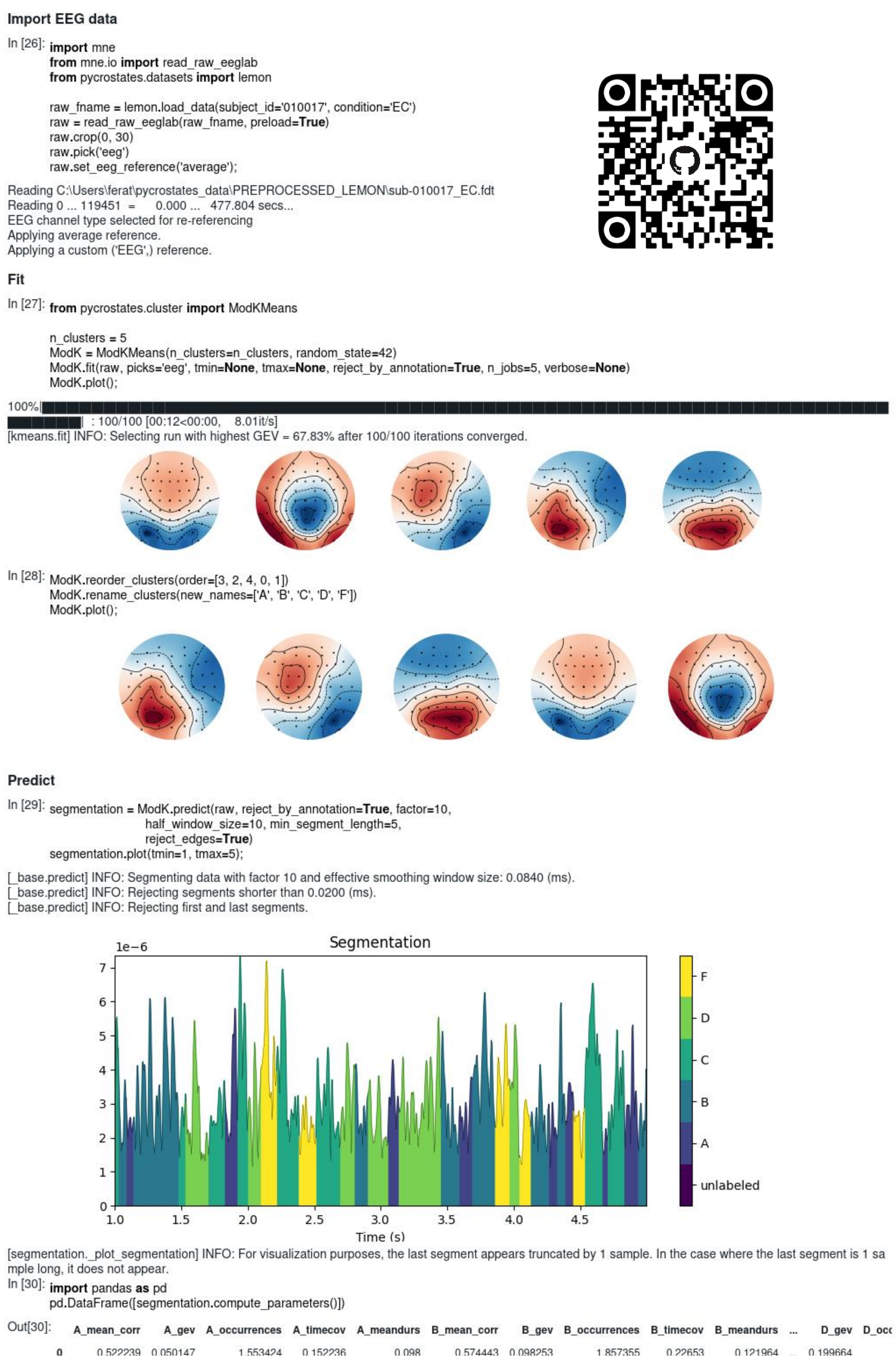
Pycrostates: a Python library to study EEG microstates

Victor Férat [1*] , Mathieu Scheltienne [2*], Denis Brunet [1,3], Tomas Ros [1,3], and Christoph Michel [1,3]

[1] Functional Brain Mapping Laboratory, Department of Basic Neurosciences, Campus Biotech, University of Geneva, Geneva, Switzerland
[2] Human Neuroscience Platform, Fondation Campus Biotech Geneva, Geneva, Switzerland
[3] Centre for Biomedical Imaging (CIBM) Lausanne-Geneva, Geneva

BACKGROUND

Microstates (MS) analysis is a method for investigating the spatiotemporal characteristics of EEG recordings. It consists of decomposing the multichannel EEG signal into a sequence of quasi-stable states, each state being characterized by a spatial distribution of EEG scalp potentials, also known as microstate maps or topographies.



Import EEG data

```
In [26]: import mne
         from mne.io import read_raw_eeglab
         from pycrostates.datasets import lemon

         raw_fname = lemon.load_data(subject_id='010017', condition='EC')
         raw = read_raw_eeglab(raw_fname, preload=True)
         raw.crop(0, 30)
         raw.pick('eeg')
         raw.set_eeg_reference('average');

Reading C:\Users\ferat\pycrostates_data\PREPROCESSED_LEMON\sub-010017_EC.fdt
Reading 0 ... 119451  =      0.000 ...   477.804 secs...
EEG channel type selected for re-referencing
Applying average reference.
Applying a custom ('EEG',) reference.
```

Fit

```
In [27]: from pycrostates.cluster import ModKMeans

         n_clusters = 5
         ModK = ModKMeans(n_clusters=n_clusters, random_state=42)
         ModK.fit(raw, picks='eeg', tmin=None, tmax=None, reject_by_annotation=True, n_jobs=5, verbose=None)
         ModK.plot();

100%|██████████████████████████████████████████████████████████████
██████████|  : 100/100 [00:12<00:00,    8.01it/s]
[kmeans.fit] INFO: Selecting run with highest GEV = 67.83% after 100/100 iterations converged.
```

```
In [28]: ModK.reorder_clusters(order=[3, 2, 4, 0, 1])
         ModK.rename_clusters(new_names=['A', 'B', 'C', 'D', 'F'])
         ModK.plot();
```

Predict

```
In [29]: segmentation = ModK.predict(raw, reject_by_annotation=True, factor=10,
                                     half_window_size=10, min_segment_length=5,
                                     reject_edges=True)
         segmentation.plot(tmin=1, tmax=5);

[_base.predict] INFO: Segmenting data with factor 10 and effective smoothing window size: 0.0840 (ms).
[_base.predict] INFO: Rejecting segments shorter than 0.0200 (ms).
[_base.predict] INFO: Rejecting first and last segments.
```

```
[segmentation._plot_segmentation] INFO: For visualization purposes, the last segment appears truncated by 1 sample. In the case where the last segment is 1 sa
mple long, it does not appear.
In [30]: import pandas as pd
         pd.DataFrame([segmentation.compute_parameters()])

Out[30]:   A_mean_corr     A_gev  A_occurrences  A_timecov  A_meandurs  B_mean_corr     B_gev  B_occurrences  B_timecov  B_meandurs  ...     D_gev  D_occ
        0     0.522239  0.050147       1.553424   0.152236       0.098     0.574443  0.098253       1.857355    0.22653    0.121964  ...  0.199664
```

* https://github.com/vferat/pycrostates

METHODS

Pycrostates implements several core functions: preprocessing tools (global field power peaks extraction, resampling), a clustering algorithm (modified version of the kmeans algorithm), clustering quality indices (Silhouette, Dunn), which are needed to perform MS analysis. It is aimed at researchers wishing to design their own microstate analyses with Python, and complements similar tools using other software, such as Cartool or the Matlab EEGLab Microstate toolbox. **Pycrostates** was built to fit seamlessly within the python scientific environment (numpy, scipy, pandas) and more particularly scikit-learn and MNE-python from which it is inspired in its philosophy and its implementation.

RESULTS

Pycrostates takes advantage of continuous integration and delivery tools such as auto-test (92.05% coverage), automatic code review (grade A) and is tested against several python versions and operating systems.

In addition, this library comes with extensive documentation including descriptions of all its algorithms and functions as well as several tutorials to help researchers to get started. Finally, Pycrostates is provided under the new BSD license allowing code reuse, even within commercial applications.

Function Library:

Datasets
- pycrostates.datasets.lemon.data_path
- pycrostates.datasets.lemon.standardize

File I/O
- pycrostates.io.ChInfo
- pycrostates.io.ChData
- pycrostates.io.read_cluster

Preprocessing
- pycrostates.preprocessing.extract_gfp_peaks
- pycrostates.preprocessing.resample

Cluster
- pycrostates.cluster.ModKMeans
- pycrostates.cluster.AAHCluster

Metrics

Segmentation
- pycrostates.segmentation.RawSegmentation
- pycrostates.segmentation.EpochsSegmentation

Visualisation
- pycrostates.viz.plot_cluster_centers
- pycrostates.viz.plot_raw_segmentation
- pycrostates.viz.plot_epoch_segmentation

CONCLUSION

- Pycrostates is a new python library for EEG microstates analysis
- Pycrostates is integrated within the scientific environment of python, especially MNE-python.
- Pycrostates takes advantage of modern development tools such as automatic testing, code quality review, which allows it to easily evolve over time.
- Pycrostates is open source and available on github *. We welcome any person wishing to participate in its development.