Instructions to reproduce the workflow of the paper “Modeling Brain Waves as a Mixture of Latent Processes”

**Rcpp main Codes:**

The following cpp codes contains the main functions to generate the MCMC to a periodogram the code also provides detailed description of the parameters used.

* BMARD\_V112020.cpp : main function to fit the BMARD method
* NDPrcpp\_pgramChoudhuri.cpp: Bayesian non-parmetric method of Choudhuri using Bernstein polynomials.

**BMARD fitting to Simulated time series:**

The 3000 (1000X 3 settings described section 3) periodogram fitted were implemented in a HPC cluster through the memory manager SLURM.

We provide the following job scripts

* simuljobAR2
* simuljobAR12
* simuljobMA4
* simuljobchoudhuri

The intention is to load the periodogram matrix and read 1 periodogram and run the MCMC. SLURM helps to run this process faster for all the time series with the following sentence.

for i in {1..1000}; do sbatch JOB\_FILE\_NAME $i; done

in case you only want to run in a local computer the R files containing the simulations codes are:

* simulationAR12.R
* simulationmixAR2.R
* simulationMA4.R
* chouduripgramcode.R

then you need to change line 6

argsbash<- commandArgs()[6]

to set the number of the simulation to fit argsbash={1,…,1000} and change the directories accordingly to the path where the files are located.

The file also indicates how to generate a new simulation in line 46.

**Main Components extraction:**

The BMARD method were run in the simulation using 6 MCMC chains and have to be summarized the following code contains the functions to extract the Maximum A Posteriori estimator of the number of components, the location, scale and weights parameters per chain and globally using all the chains.

* ExtractionBMARDmaincomponentsmodes.R

To accelerate the process the main components extraction was implemented in the HPC cluster with the next job files

* excompMA4job
* excompAR2job
* excompAR12job

these job files read the following R code (match the simulations name for each case)

* extractAR2mixmodes.R
* extractAR22modes.R
* extractMA4modes.R

In the case of the Bernstein polynomials method of Choudhuri (2014) the extraction just computes the pointwise mean, median, and quantile curves using the next job files:

* extcurvchoudAR2mix
* extcurvchoudAR12
* extcurvchoudMA4

these job files read the following R code (match the simulations name for each case)

* ExtractionmeancurvesCHOUDAR2mix.R
* ExtractionmeancurvesCHOUDAR12.R
* ExtractionmeancurvesCHOUDMA4.R

**Figure 2 Reproducibility:**

The First section loads the data form the shared matrices

Second section compute the kernel and spline estimator, this should be computed relatively quick.

Third section computes the standardized spectral density function of the three-setting described in section 3.

In section 4 is assumed the previous step were run and load the result data bases corresponding to the extracted main pointwise summary curves (mean, median and quantiles) from the simulations for BMARD and the Bernstein polynomial method of Choudhuri (2014). Then it is necessary locate manually the median of the integrated absolute error (IAE) to finally plot all the curves together using matplot().

**Table 1 and Table 2 reproducibility:**

These tables compute the differences (disparity) of the true values of the parameters specified in the Section 3 for the mixture of AR(2) processes and the location of the peaks of the AR(12) process.

Table 1 then localize the estimation that correctly have at least 3 components and compute the absolute difference between the MAP estimations and the true parameter values, the compute the average and standard deviation for each of the components.

Table 2 only consider the disparity between the true peaks locations and the estimated locations for the 5 peaks of the AR(12) process.