**Instructions**

Simulation script:

*ViEEG\_manuscript\_forward\_inverse\_simulation\_Beamforming.py*

**Python dependencies**

This script uses the following Python packages:

*numpy (v1.20.2)*, *scipy (v.1.6.2)*, *mne (v0.22.1)*, *hdf5storage (v0.1.17)*, *matplotlib.pylab (v1.20.2)*

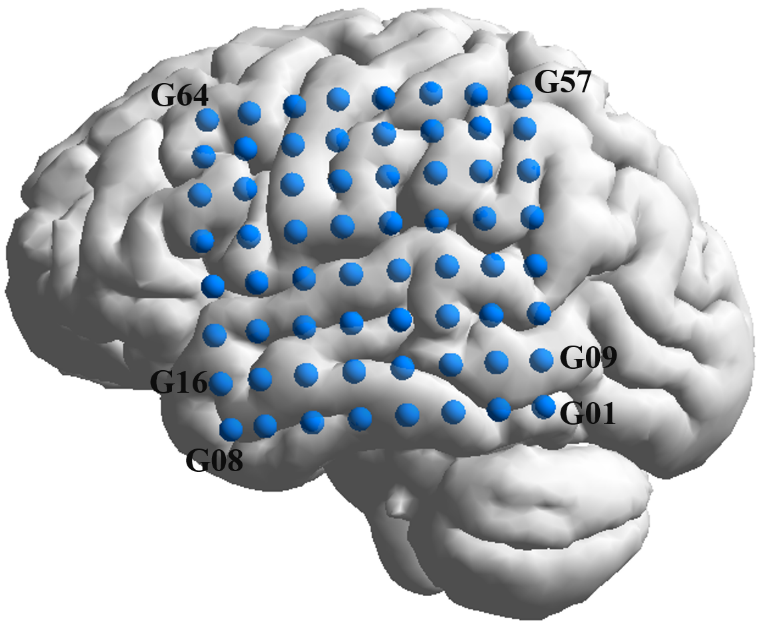
I guess you will probably need to install MNE-Python package with the latest stable version (v 0.22.1). Please follow the instruction from MNE-Python website <https://mne.tools/stable/install/mne_python.html> and I highly recommend installing MNE-Python using Anaconda (conda command) and its virtual environment through environment.yml file offered by MNE-Python (conda env create --name mne --file environment.yml). This enables a clean installation of MNE-Python environment with all necessary dependencies (and required versions) without contaminating your base Python environment.

**Some notes**

In the Python script, I used 64 Gaussian random signals as simulated/generated time-series and assigned them to 64 ViEEG electrodes. Then these Gaussian random signals were forward and inverse modelled using function *forward\_inverse\_modelling*. In this simulation, the number of time-series can be less than 64 but cannot be over 64, given the size (and location) of ViEEG grid is fixed. The current operating sample rate is fixed at 1000 Hz, so best that generated time-series have the same sample rate as 1000 Hz. However, I guess sample rate is not sensitive at all in the simulations that Reviewer 1 asked for.

Because beamformer technique requires to define a baseline, it means if you provide 90-second time-series it will use the first one-third of signals as baseline (define time zero at one-third of time points) and only use the first two-third as the epoch for source reconstruction using beamformers. In the script, when you specify tmin\_epoch (e.g., -5) and tmax\_epoch (e.g., 35), the reconstructed signals and the original signals will be cropped into [tmin\_epoch, tmax\_epoch] segment as the outputs of *forward\_inverse\_modelling* function. So, the longer signals you use in forward and inverse modelling the better.

A ViEEG grid with size of 8-by-8 is implemented in left temporal-parietal regions, as demonstrated in the figure below. This allows simulating a network of max. 64 nodes/time-series. The layout of ViEEG grid is given below with virtual electrode labels. This should allow to superimpose some specific connectivity structures with known ictogenecity.



*ViEEG grid with size of 8-by-8 (64 nodes/time-series in total) placed in left temporal-parietal regions. Labels are assigned in the same fashion that G01starts from posterior and inferior aspects.*