**Analysis of neuroarchiver data for spectral features and interictal events**

**Recording notes:** To simplify later analysis, it is recommended that data is acquired with clocks on BST. i.e. avoid recording across days when clocks change

**Repository details:**

* Download and install Git
* At the command line, navigate to a folder to be used for analysis and clone the github repository by typing:

“git clone <https://github.com/IrisOren/NeuroarchiverAnalysis.git>”

* The repository contains:
  + DataProcessing folder with files needed for the Data processing steps
  + NeuroarchiverAnalysisR with files and folders for post-processing in R
  + These instructions

**Eddie processing**

**Installation of LWDAQ on Eddie:** The processing of neuroarchiver data is performed on the Eddie cluster. The processing requires a version of neuroarchiver/lwdaq enabled on Eddie.

# Create a directory in your shared group area for lwdaq:

$ mkdir /exports/cmvm/eddie/sbms/groups/Oren/lwdaq

$ cd /exports/cmvm/eddie/sbms/groups/Oren/lwdaq

# Download the zip file using wget and unzip it (look what the latest version available is on the # neuroachiver website, here 8.5.22:

$ wget http://alignment.hep.brandeis.edu/Software/Download/LWDAQ\_8.5.22.zip

$ unzip LWDAQ\_8.5.22.zip

**Data processing:**

1. *On your datastore directory, prepare*
2. Data folder containing subfolders for each animal which contain all ndf data files for that animal
3. Output folder containing subfolder for each animal. Each subfolder contains:
   1. A copy of the Neuroarchiver processor**:** Neuroarchiver is able to analyse continuous EEG data by processing intervals in of a specified length. The processor calculates measures of interest (“metrics”) for each interval and writes them to a text file.

**The SCPP4V2.tcl** processor outputs the following metrics:

* Loss % (0-100)
* Spikes
* Bursts
* Delta power (0.1-3.9Hz)
* Theta power (4-12-Hz)
* Gamma1 power (20-80Hz)
* Gamma2 power (40-80Hz)
* Broadband1 power (5-160Hz)
* Broadband2 power (0.1-160Hz)
  1. config.tcl: The file specifies
* The processor to be used (e.g. SCPP4V2.tcl)
* The play\_interval length (8s used for SCPP4V2)
* The glitch\_threshold (200 used)
* **The channel to process (Remember to edit this for the specific animal!!!)**

1. *Copy data and output folders to Eddie*
2. From terminal (or mobaXterm on windows): ssh [uun@eddie3.ecdf.ad.ac.uk](mailto:uun@eddie3.ecdf.ad.ac.uk) (https://www.wiki.ed.ac.uk/display/ResearchServices/Quickstart)
3. Login to the staging environment which is used to copy files between Eddie and external locations: “qstaging –q login” (<https://www.wiki.ed.ac.uk/display/ResearchServices/Data+Staging>)
4. Navigate to source directory (either the data directory or output directory) which contains all the animal subfolders on datastore (eg. “cd /exports/cmvm/datastore/sbms/groups/Oren/Users/IrisOren/Neuroarchiver/J20) and copy to destination using the command: “rsync –rlva \* DESTINATIONDIRECTORY” (e.g /exports/cmvm/eddie/sbms/groups/Oren/Data/J20/)
5. Confirm that the files have been copied with correct permissions using “ls –l DATADESTINATIONDIRECTORY” and “ls –l OUTPUTDESTINATIONDIRECTORY”
6. *Setup job.sh file on Eddie. This is the Eddie job file. I save it in the parent directory on Eddie and edit it for each animalSubfolder, but you can create a different job file for each processing batch*
7. For each animal data folder, determine the number of ndf files to be processed using: “ls -1 DATADESTINATIONDIRECTORY/ANIMALSUBFOLDER/\*.ndf | wc –l”
8. Open the job.sh using “nano job.sh”
9. Edit line 12 to specify the number of ndf files to process
10. Edit line 19 to specify data SUBDIR to be processed to ANIMALSUBFOLDER
11. Edit line 25 to specify the INPUT\_BASE\_DIR (DATADESTINATIONDIRECTORY)
12. Edit line 26 to specify OUTPUT\_BASE\_DIR (OUTPUTDESTINATIONDIRECTORY)
13. Edit line 33 to specify CONFIG\_DIR (OUTPUTDESTINATIONDIRECTORY)
14. Check that the correct processor is specified in the line 44 as CONFIG2
15. Press CtrlX to exit and save (Y)
16. *Run job.sh*
17. From the command line submit the job by typing “qsub job.sh”
18. To monitor the progress of the job type “qstat –g d”
19. When the job has finished running, it will have created subfolders in the OUTPUTDESTINATIONDIRECTORY/ANIMALSUBFOLDER for each of the ndf files. Each subfolder contains the metric output file, and a copy of the processor and config file. These can be deleted using “rm OUTPUTDESTINATIONDIRECTORY/ANIMALSUBFOLDER/M\*/\*.tcl
20. Delete the job log files using “rm jobname\*”
21. *Copy files back to datastore*
22. Login to staging environment “qlogin –q staging”
23. “rsync –rlva SOURCEFILESONEDDIE DESTINATIONFOLDERONDATASTORE”

**Create RecordingInfo.csv and validate performance of spike counter:**

If analysing interictal spikes, then the performance of the spike counter needs to be validated. The spike counter might perform poorly in certain recordings due to noise or battery dying. To validate the performance, we can create an event list of identified spikes, and hop through events to quantify false positives.

1. Open neuroarchiver -> Tool -> Toolmaker
2. Load and run Spike\_ExtractSCPP4V2.tcl to create event list of all spikes and bursts
3. In neuroarchiver PickDir of ndf data directory
4. Pick the created event list, and hop through events (e.g. 100) to quantify false positive rate
5. Repeat for each animal
6. Save in spreadsheet NeuroarchiverAnalysisR/Data/RecordingInfo.csv with columns: AnimalID, Genotype, Channel, FalsePositive (Set to 0 if no information)

**Identify data for example traces**

If example traces are to be plotted, use the Export\_Signal.tcl as the neuroarchiver processor to export .txt files of each interval

If power spectra are to be plotted, use Spectrum\_Export.tcl as the neuroarchiver processer which exports fourier transform for each interval. Run this on the same intervals as exported signals.

**Video scoring**

* For each animal to be analysed, create a csv file with video scoring
* Files contain variables "Date", "Time", "S\_W", e.g. 19/08/2017,12:15:31,S.
* Each row specifies a behavioural state (“S” or “W”) and needs only to include the transition to the next behavioural state.
* The last entry must be the behavioural state at the end of the observation period.
* Save files in NeuroarchiverAnalysisR/Data/VideoScoring

**R post-processing: Summarise, plot and statistics per animal**

* The analysis is organised as an R project.

In RStudio, select: new project – Existing Directory, and select NeuroarchiverAnalysis/NeuroarchiverAnalysisR

* + NeuroArchiverAnalysis.rmd is a RMarkdown file. It calls several function contained in the R subfolder.
  + The required Data files should be saved in the Data folder.
    - RecordingInfo.csv: This should contain variables: AnimalID, Genotype, Channel, FalsePositive
    - VideoScoring files (see above)
    - Any exported traces and power spectra in .txt format
  + Output data is written to the Output folder. Running (aka. “knitting”) the RMarkdown file creates an HTML document of results “NeuroArchiverAnalysis.html”.
  + Before knitting the rmd file, set variables in VariableInitialisationChunk and SetThresholdChunk