The seizure detection algorithm for neonatal EEG

This algorithm outputs the binary decision and performance measures for 4 different seizure detection algorithms (SDA). For SDAs with a trained SVM-model (I1-I3), the algorithm also outputs the raw decision values and features. The main file for the execution of an SDA is seizure_detection.m. The user can run the algorithm with EEG data in the following input types:

D1) EEG with a similar montage used in [1]
D2) EEG stored in an EDF format
D3) EEG stored in a mat-format

Technical notes:

- The algorithm includes a notch filter on 50Hz. If the user's EEG signal has a DC component different to 50Hz, the EEG should be preprocessed with a corresponding notch filter before running the algorithm
- The processing times with the most accurate algorithm (SDA) are quite long and therefore, users are strongly advised to use parallelization (input n for number of parallel pools). Faster implementations are in development.
- The algorithms are built with Matlab 2017a

The implementation can be chosen from:
I1) SDA – The automated seizure detection algorithm proposed in [1]
I2) SDA_DB_mod – A modified version of the SDA_DB [2,3]
I3) SDA_T – An implementation of the SDA of Temko et al. [4]
I4) SDA_DB – An implementation of the SDA of Deburchgraeve et al. [2]

There is also a facility to reproduce the results of the IJNS paper [1], instructions later in this document.

The outputs are:
For all of the input EEG types:

- dec - the final decision/binary output annotation sampled at 1Hz

For implementations I1-I3:

- dec_raw - the raw SVM output for each EEG channel
- feats – features computed on each channel sampled every 4 seconds

Performance measures:
To compute performance measures for input types 1-3, run compute_results_new_detection.m after running seizure_detection.m

Due to limitations in data storage on GitHub, SVM model files have been placed in an alternate repository (Zenodo, see prerequisites below). Ensure the SVM model files are in the same folder as the scripts seizure_detection.m and reproduce_SDA.m.

Prerequisites:

- Download repository from **Github**
- Download SVM files from Zenodo (https://zenodo.org/record/1281146#.WxoxIXVubCJ)
- Download EEG data (https://zenodo.org/record/1280684#.Wxh3QkiFNaQ – if required)

Keep all files in a single directory or set the path (use addpath) in Matlab to point to Code and Data files. The SVM-files are need with implementations I1-13.

Examples of use

EEG input type D1:

Inputs:

- filename - the filename of an 18 channel EEG recording, which is identical to the one used in the EEG database (https://zenodo.org/record/1280684#.Wxh3QkiFNaQ)
- format – a scalar denoting data input, in this case **format=1**
- detector – the SDA implementation as a string variable, choose from I1-I4
- n – the number of parallel pools (stronly encouraged if available, with I1-I3), if no multiple cores available, define n=1

Example:
Run the algorithm on a EDF file from our database.
filename=’ eeg1.edf’; **format=1**; detector=’SDA’; n=4;
[dec, dec_raw, feat] = seizure_detection(filename, format, detector, n);

If you have an annotation of the data then you can compare the SDA output to the annotation using compute_results_new_detection.m:
[auc, tdr, fdr] = compute_results_new_detection(dec_raw, annotat, detector), which returns AUC, true detection rate and false detection rate for the implementation chosen (=detector). In addition to the detector, this function requires the raw decision values and the annotation corresponding the input EEG for seizure_detection-function.

When the seizure_detection.m is run with eeg1.edf and fullSVM_SDA (Zenodo), the resulting dec and max dec_raw and raw decision values per channel should look similar to figures 1 and 2, respectively.

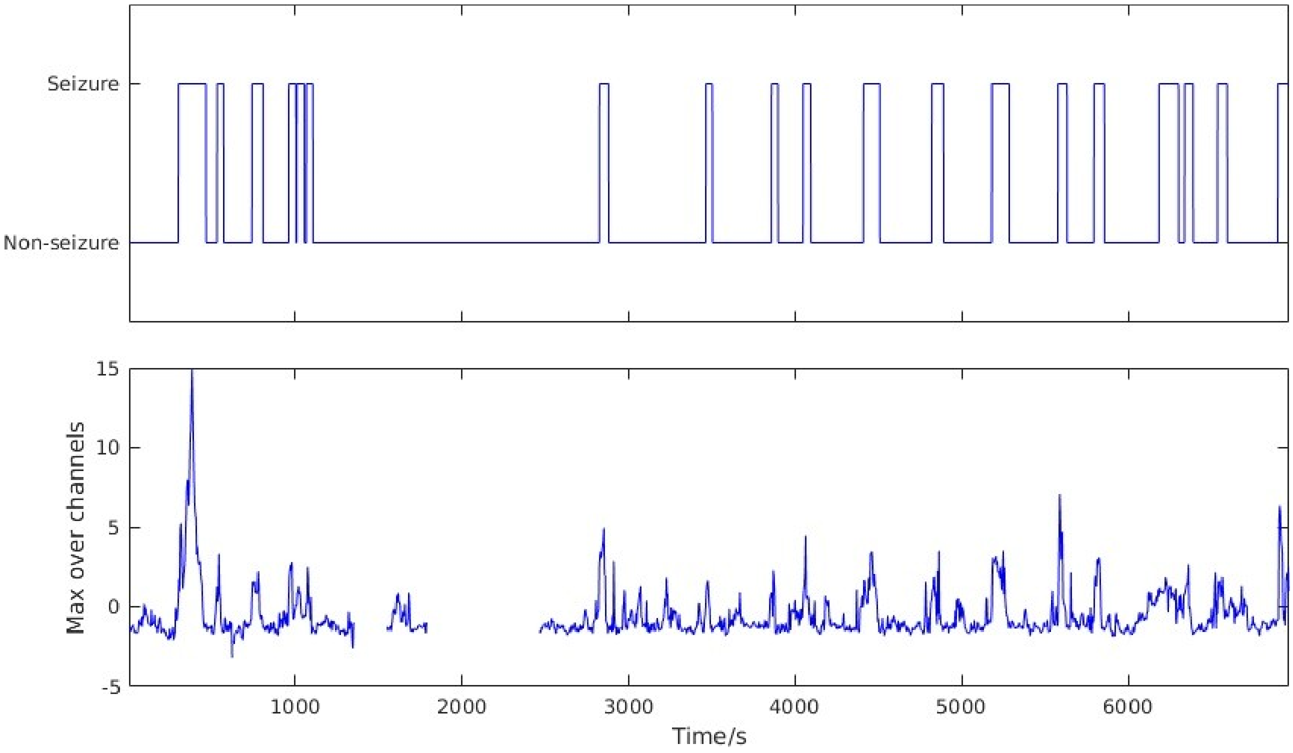

Figure 1. Top figure: Binary decision between seizure and non-seizure, bottom figure: maximum raw decision values over channels.

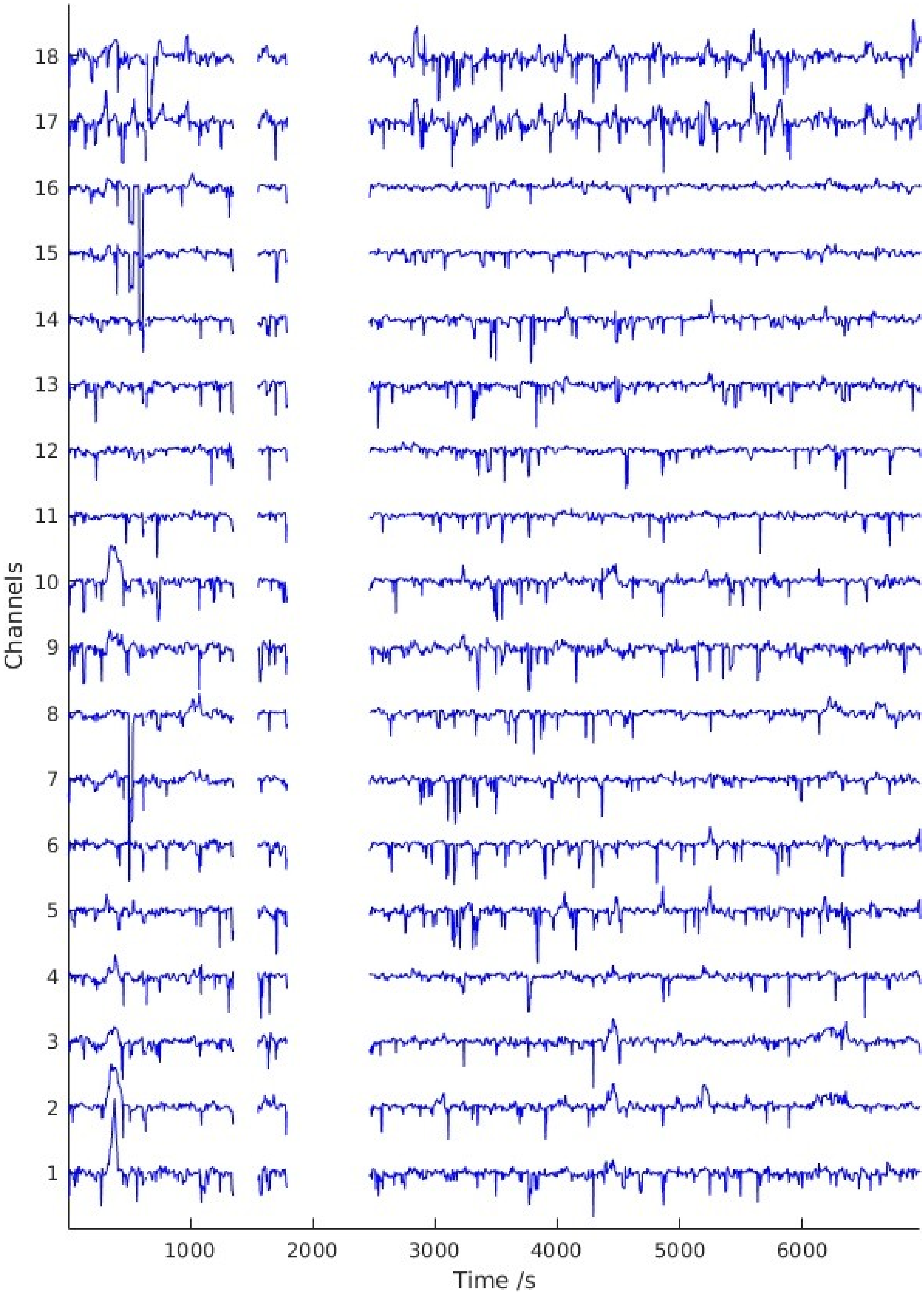

Figure 2. Raw decision values for each channel on eeg1.edf.

EEG input type D2:

Inputs:

- filename - the filename of an EEG recording including EEG data only in bipolar montage
- format – a scalar denoting data input, in this case **format=2**
- detector – the SDA implementation as a string variable, choose from I1-I4
- n – the number of parallel pools (stronly encouraged if available, with I1-I3), if no multiple cores available, define n=1

Example:

Run the algorithm on an arbitrary EDF file. Note the EDF file must be exported with the data montage to be used and every channel will be considered as an EEG channel (omit additional polysomnographic channels before export and export with desired EEG montage). If this is not the case use read_edf.m and build your own montage in Matlab.

```
filename=’ eeg1001.edf’; format=2; detector=’SDA_T’; n=4;
[dec, dec_raw, feat] = seizure_detection(filename, format, detector, n);
```

EEG input type D3:

Inputs:

- filename - a Matlab variable containing only EEG data (channel x time)
- format - a scalar denoting data input, in this case **format=3**
- detector – the SDA implementation as a string variable, choose from I1-I4
- n – the number of parallel pools (strongly encouraged if available, with I1-I3), if no multiple cores available, define n=1
- **fs – a scalar defining the sampling frequency (Hz) of the EEG data in the mat-file**

Example:

Run the algorithm on EEG data saved in a Matlab format.

```
filename='eeg.mat'; format=3; detector='SDA_DB'; n=1; fs=100;
dec = seizure_detection(filename, format, detector, n, fs);
```

Reproducing results of the paper [1]:

This case requires the download of the annotation file from Zenodo in addition to the EDF-files (https://zenodo.org/record/1280684#.Wxh3QkiFNaQ).

Inputs:

- group - the SDA implementation as a string variable, choose from I1-I4
- path - the path to the edf-files as a string variable
- annotat – the annotation mat-file
- n – scalar denoting the number of parallel pools (if non used, n=1)

Example:

Run the algorithm reproduce_SDA.m:

```
group=’SDA_DB_mod’; path=’path_to_edf_files’; n=10;
[dec,results,dec_raw,feats]=reproduce_SDA(group, path, annotat_new, n);
```

Compare_results to the orignal IJNS paper [1] by running:

```
disp_performance_measures(path_SDA,path_SDA_DB,path_SDA_DB_mod,path_SDA_T)
```

The inputs are paths to results-variables for each algorithm (all of them are required here).

References:

[1] K. Tapani, S. Vanhatalo and N. Stevenson, Time-varying EEG correlations improve automated neonatal seizure detection, International Journal of Neural Systems. (accepted for publication)

[2] K. Tapani, S. Vanhatalo and N. Stevenson, Incorporating spike correlations into an SVM-based neonatal seizure detector, EMBEC, 2017, pp. 322–325.

[3] W. Deburchgraeve, P. Cherian, M. De Vos, R. Swarte, J. Blok, G. Visser, P. Govaert and S. Van Huffel, Automated neonatal seizure detection mimicking a human observer reading EEG, Clin Neurophysiol 119(11) (2008) 2447–2454.

[4] A. Temko, E. Thomas, W. Marnane, G. Lightbody and G. Boylan, EEG-based neonatal seizure detection with support vector machines, Clin Neurophysiol, 122(3) (2011) 464–473.