

Neuro-Integrative Connectivity (NIC) Workflow Component I: Data Processing and Transformation

Version 1.0

Section 0 — Licensing

NIC is licensed under a Creative Commons Attribution-NonCommercial-NoDerivs 4.0 Unported License. For a full copy of the license and terms, see <http://creativecommons.org/licenses/by-nc-nd/4.0/>.

Section 1 — Uses

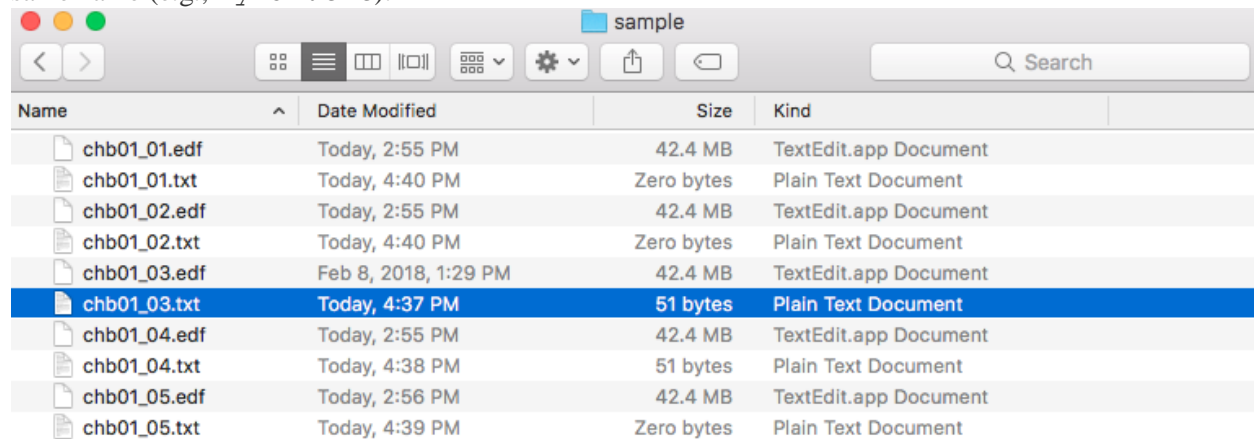
The software is intended to be used for the conversion of data files obtained from electroencephalography (EEG) in the European Data Format Plus Annotations (EDF+) into the Cloudwave Signal Format (CSF). CSF is intended to provide better human readability (based on JavaScript Object Notation), ease of computation with the companion CSF Computational Tool, and adaptability for use with parallel processing techniques.

Section 2 — Requirements

- Java SE 8 or higher is required to run the software; Java SE 10 is available at <http://www.oracle.com/technetwork/java/javase/downloads/index.html>.
- The software requires Javax JSON and Apache Commons Math JAR files. These should be contained in the JAR file for the converter. However, if there are issues, the required JAR files can be downloaded locally and added to the classpath during execution.
 - **JSON:** JAR files for JSON processing are included in Java EE 7 or higher; Java EE 8 is available at <http://www.oracle.com/technetwork/java/javaee/downloads/index.html>. Alternatively, specific JAR files may be obtained at http://download.oracle.com/otndocs/jcp/json_p-1_1-final-spec/index.html.
 - **Apache Commons:** The JAR file for Apache Commons Math is available at <https://commons.apache.org/proper/commons-math/>.
- EDF files should be separated into directories with one directory per patient.

Section 3 — Setup

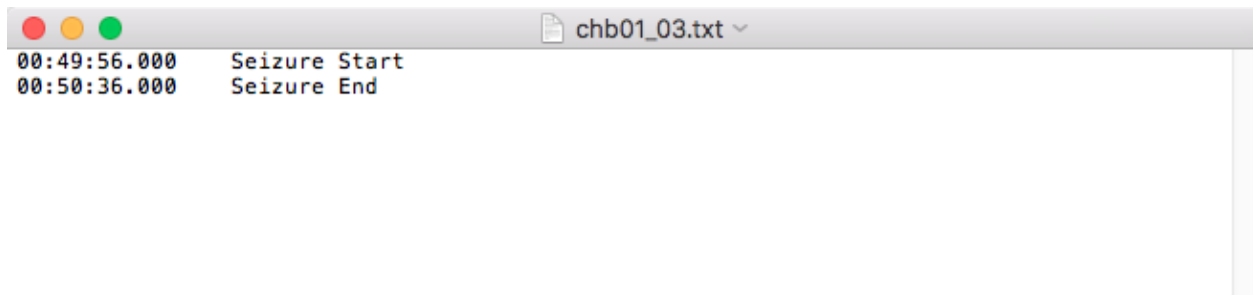
The folder must consist of EDF files (e.g., myEdf.edf) with corresponding annotation text files of the same name (e.g., myEdf.txt).



Name	Date Modified	Size	Kind
chb01_01.edf	Today, 2:55 PM	42.4 MB	TextEdit.app Document
chb01_01.txt	Today, 4:40 PM	Zero bytes	Plain Text Document
chb01_02.edf	Today, 2:55 PM	42.4 MB	TextEdit.app Document
chb01_02.txt	Today, 4:40 PM	Zero bytes	Plain Text Document
chb01_03.edf	Feb 8, 2018, 1:29 PM	42.4 MB	TextEdit.app Document
chb01_03.txt	Today, 4:37 PM	51 bytes	Plain Text Document
chb01_04.edf	Today, 2:55 PM	42.4 MB	TextEdit.app Document
chb01_04.txt	Today, 4:38 PM	51 bytes	Plain Text Document
chb01_05.edf	Today, 2:56 PM	42.4 MB	TextEdit.app Document
chb01_05.txt	Today, 4:39 PM	Zero bytes	Plain Text Document

(Note: The EDF files being used are from the CHB-MIT Scalp EEG Database, which is publicly available through the PhysioBank website.)

These text files must conform to the EDF+ specifications -- that is, they must be in a two-column format with the first column being an offset timestamp from the beginning of the file (the preferred format is HH:MM:SS.SSS), and the second column is a text annotation string. The file may indeed be empty (note that `chb01_01.txt` contains zero bytes), but it must exist.



Section 4 — Invocation

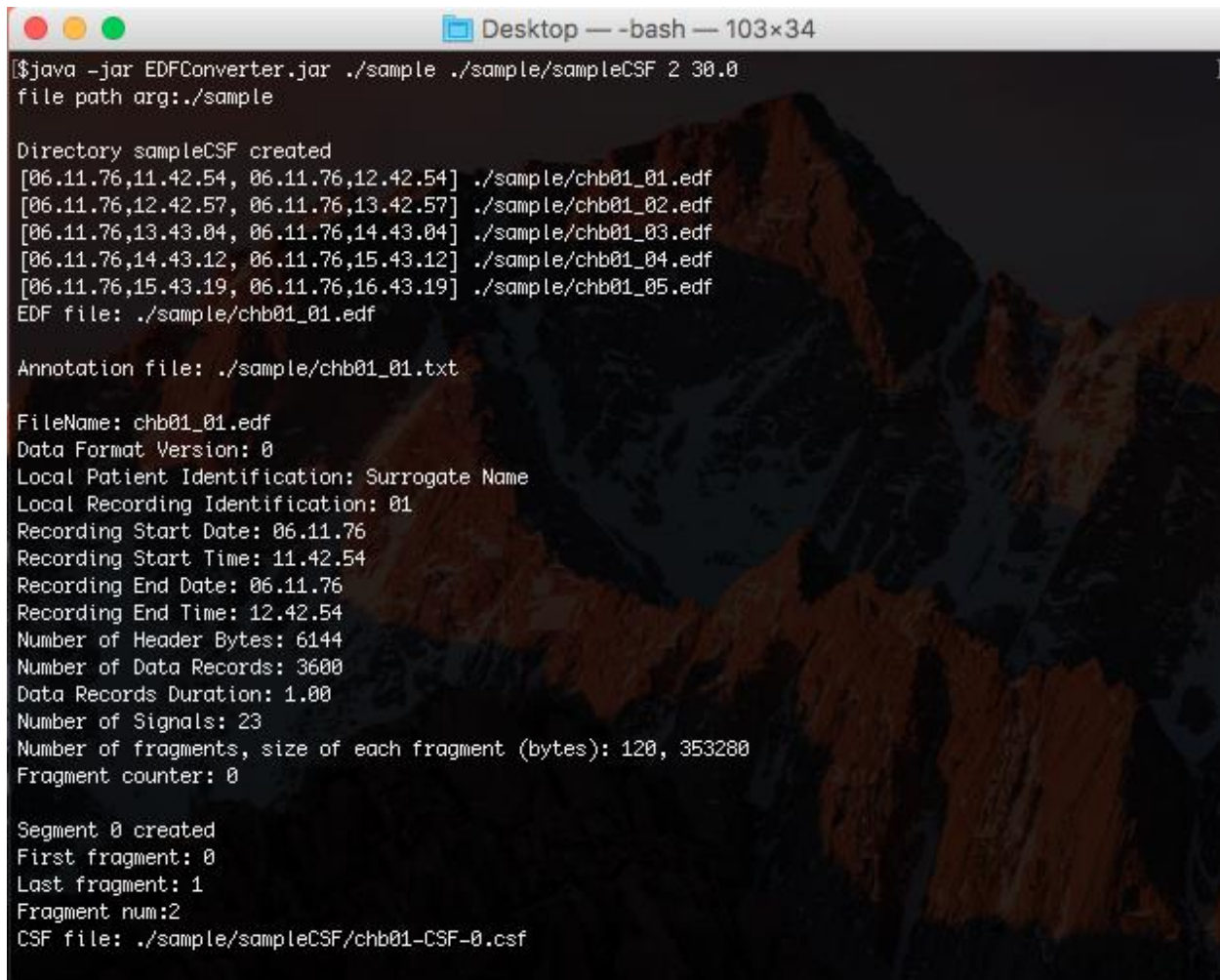
From the command line, navigate to where `EDFConverter.jar` is located and enter

```
java -jar EDFConverter.jar <EDFdir> <CSFDir> [frag {2}] [epoch {30.0}]
```

with the following arguments:

- **EDFdir (mandatory):** The String filepath of the directory containing the EDF files to be processed
- **CSFdir (mandatory):** The String filepath of the directory in which output CSF files will be placed; if the directory does not currently exist, it will be created
- **Frag (optional; default 2):** The integer that represents the maximum number of fragments to be included in each output CSF file
- **Epoch (optional; default 30.0):** The floating point value that represents the maximum time duration of each fragment in a CSF file

The program will begin by locating the destination folder for the CSF files, and creating it if it does not already exist. It will then sort the EDF files into chronological order. Then the program will process the EDF files in order, obtaining the EDF file and the associated annotation text file, and thereby create the corresponding CSF files. The program prints header information for each EDF file (this is for verification purposes) and summary statistics about the creation of each CSF file.



```
Desktop -- -bash -- 103x34
[$java -jar EDFConverter.jar ./sample ./sample/sampleCSF 2 30.0
file path arg:./sample

Directory sampleCSF created
[06.11.76,11.42.54, 06.11.76,12.42.54] ./sample/chb01_01.edf
[06.11.76,12.42.57, 06.11.76,13.42.57] ./sample/chb01_02.edf
[06.11.76,13.43.04, 06.11.76,14.43.04] ./sample/chb01_03.edf
[06.11.76,14.43.12, 06.11.76,15.43.12] ./sample/chb01_04.edf
[06.11.76,15.43.19, 06.11.76,16.43.19] ./sample/chb01_05.edf
EDF file: ./sample/chb01_01.edf

Annotation file: ./sample/chb01_01.txt

FileName: chb01_01.edf
Data Format Version: 0
Local Patient Identification: Surrogate Name
Local Recording Identification: 01
Recording Start Date: 06.11.76
Recording Start Time: 11.42.54
Recording End Date: 06.11.76
Recording End Time: 12.42.54
Number of Header Bytes: 6144
Number of Data Records: 3600
Data Records Duration: 1.00
Number of Signals: 23
Number of fragments, size of each fragment (bytes): 120, 353280
Fragment counter: 0

Segment 0 created
First fragment: 0
Last fragment: 1
Fragment num:2
CSF file: ./sample/sampleCSF/chb01-CSF-0.csf
```

The program will continue in this manner until all of the EDF files have been processed.

```
Desktop -- -bash -- 103x34
Segment 59 created
First fragment: 118
Last fragment: 119
Fragment num:2
CSF file: ./sample/sampleCSF/chb01-CSF-59.csf

first fragment: 118, last fragment: 119
EDF file: ./sample/chb01_02.edf

Annotation file: ./sample/chb01_02.txt

FileName: chb01_02.edf
Data Format Version: 0
Local Patient Identification: Surrogate Name
Local Recording Identification: 01
Recording Start Date: 06.11.76
Recording Start Time: 12.42.57
Recording End Date: 06.11.76
Recording End Time: 13.42.57
Number of Header Bytes: 6144
Number of Data Records: 3600
Data Records Duration: 1.00
Number of Signals: 23
Number of fragments, size of each fragment (bytes): 120, 353280
Fragment counter: 0

Segment 60 created
First fragment: 121
Last fragment: 122
Fragment num:2
CSF file: ./sample/sampleCSF/chb01-CSF-60.csf

first fragment: 121, last fragment: 122
Fragment counter: 2
```

```
Desktop -- -bash -- 104x35
Segment 296 created
First fragment: 593
Last fragment: 594
Fragment num:2
CSF file: ./sample/sampleCSF/chb01-CSF-296.csf

first fragment: 593, last fragment: 594
Fragment counter: 114

Segment 297 created
First fragment: 595
Last fragment: 596
Fragment num:2
CSF file: ./sample/sampleCSF/chb01-CSF-297.csf

first fragment: 595, last fragment: 596
Fragment counter: 116

Segment 298 created
First fragment: 597
Last fragment: 598
Fragment num:2
CSF file: ./sample/sampleCSF/chb01-CSF-298.csf

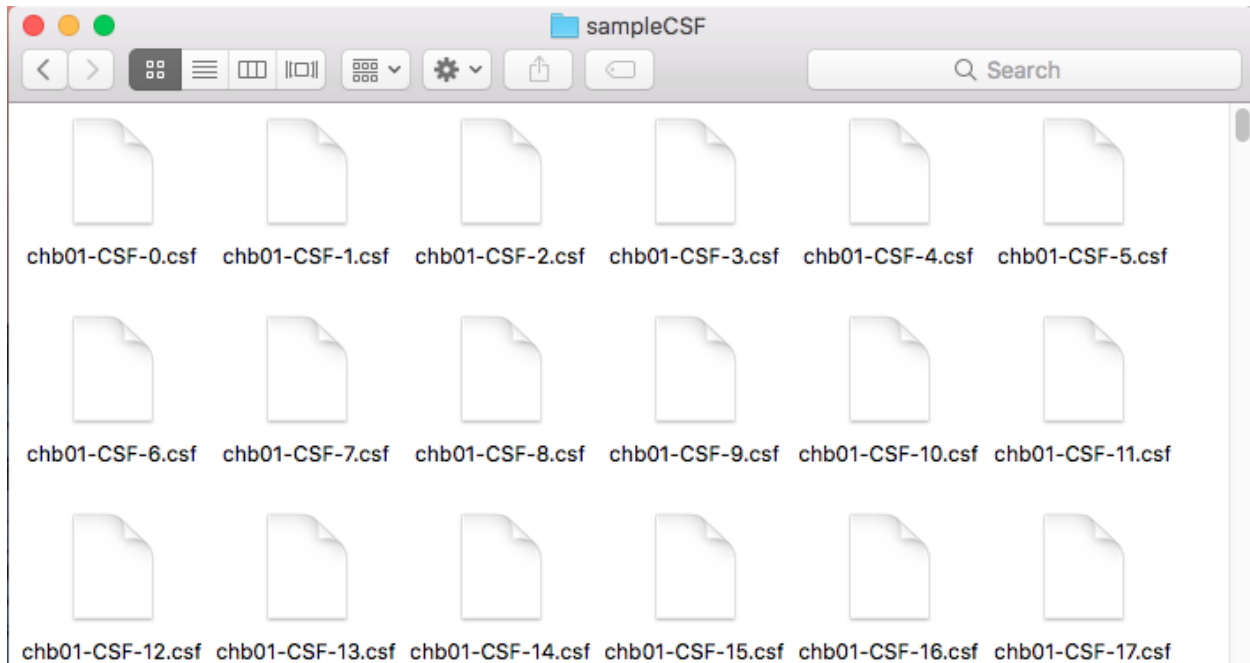
first fragment: 597, last fragment: 598
Fragment counter: 118

Segment 299 created
First fragment: 599
Last fragment: 600
Fragment num:2
CSF file: ./sample/sampleCSF/chb01-CSF-299.csf

first fragment: 599, last fragment: 600
$
```

Section 4 — Output

The output will be a directory populated with CSF files resulting from the processing of the directory of EDF files. The names of the CSF files are of the form `myEDF-CSF-#.csf`, where `myEDF` is the name common to the EDF files that occurs before the first underscore character, and the numbering of the files is chronological beginning at 0 (the EDF files are sorted chronologically before processing). Information about the correspondence between CSF files and EDF files is contained within the CSF files themselves, and is therefore not reflected in the filenames.



Credits

This software has been developed at Case Western Reserve University as part of a research project and includes contributions by the NIC development team members: Arthur Gershon, Pramith Devulapalli, Vimig Socrates, Haroon Khazi, and Satya Sahoo (Project PI).