FAQ

*"If something requires **you** in the equation to work, it will not work for science."*

What is SpiSOP?

SpiSOP (abrev. for Spindles Slow Oscillation and Power-spectral-density) is an open source tool supporting detection and reporting of spindle or slow oscillation events, their co-occurance or respective matching non-events and power (density) of specific spectra or frequency bands in pre-scored (sleep stages) EEG and MEG data as well as simple automatic EMG artifact detection. Most detection methods and annotations reported in sleep EEG research are covered and thus can be replicated by minor adaptions in the standard parameters.

SpiSOP was designed to process large data quanta at once and multiple datasets in parallel.

Analyses is according to wished channels, their various combinations, specified dynamic sleep stages, various detection thresholds and further parameters.

Frequency peaks in the power band of slow oscillations, spindles or any other band of interest are automatically determined and visually confirmed or adapted by user.

SpiSOP also contains a data browser to sleep (re)score and view sleep EEG data and previously detected events, as well as aids in scoring and detection decisions.

It comes as a standalone command line tool for Windows, Linux

and Mac, or can be integrated as toolbox in Matlab.

Read in data and performing spectral analyses is based on the ***FieldTrip*** toolbox (GPLv2+, for details see: http://fieldtrip.fcdonders.nl/ or http://www.fieldtriptoolbox.org/).

In brief, SpiSOP is free software for most commonly used basic sleep EEG analyses in one easy tool, and facilitates the fast replication of results on large data quanta, as well as enables others to share and communicate the exact methods.

In long, and abstract-like form:

FAST REPLICATION AND SHARING OF BASIC SLEEP EEG ANALYIS IN ONE TOOL –
Replicating a comprehensive sleep EEG analysis is a challenge. But is it the only means to overcome the prevailing conflicts in reports on the mechanisms and function of hallmark sleep EEG features, including sleep spindles. Even minor methodological changes at the base of sleep EEG analysis can have vast consequences on conclusions and therefore complicate their communication to other researchers. To address the abundance of methodological variation and their implementation to extract any sleep EEG feature is an overload to any researcher. Those difficulties do not only hinder a researcher's investigation and focus, but also undermines the progression of the field. Furthermore, any sleep feature is hard to understand in isolation, and requires parallel assessment of related other features to investigate their putative relationships, i.e. temporal, phasic and topographical occurrence, co-occurrence and their sources. This calls for a free tool that is easy to use and share for rapid replication of sleep EEG analyses by other researchers using the same standards or alterations of them.
Here I introduce 'SpiSOP', a parallel computing tool for experts as well as non-experts to process large datasets quickly and exchange used methodology easily. Based on scored

sleep EEG data, this toolbox supports automated detection and reporting of (fast and slow) sleep spindles, slow oscillations, their co-occurrence, search for respective sleep stage matching randomized 'non-events', as well as power density of specific spectra or frequency bands. Automatized exclusion of typical confounding sleep EEG artifact epochs is also supported. Analyses are according to wished channels, their various combinations, specified dynamic sleep stages, various detection thresholds and further parameters. Visualization and comprehensive comparison between sleep scorers, as well as a tool that aids scoring decisions are also included. Methods are standardized and they support most basic features as they were reported in current sleep and memory research. Replication of results is fast. It requires only minimal intervention from the researcher with minor adjustments of vastly flexible standard parameters. 'SpiSOP' is open source software under the GPLv3 for non-commercial use. Core functions were implemented based on the FieldTrip toolbox to cope with almost any available EEG data and recording settings. It does not require any Matlab programming skills, but only a short introduction in handling. SpiSOP comes as standalone software and does not require a Matlab license. A complete standard EEG analysis can be set up, changed and started within minutes, and the setups can be shared to others to be replicated on other datasets. This represents a first step in analyzing large sleep data quanta across researchers, studies, designs, tasks and species, thus simplifies a precise understanding of the fine-tuned sleep EEG phenomena.

Why you should thank

FieldTrip and other contributers?

If you use SpiSOP, you are also using other open source software contributed by other people, most prominently the code of the many FieldTrip contributers! SpiSOP would not have possible without them or FieldTrip. In thus I want to thank them for their inspiring work, and encourage you to do the same. Since SpiSOP is just very basic analysis, usually at the beginning, I would encourage to use more sophisticated analyses using FieldTrip, take a look at http://www.fieldtriptoolbox.org/ and then cite them too, or at giving them "internet community support" by likes tweets etc., but at least invite them for bear, coffee, tea etc. when you get to know one of them.

How is the SpiSOP licensed?

The SpiSOP source code and compiled standalone are released under a GPLv2+ license (see COPYING file for details). However the full copyrights of the documentation, including the figures, remain to the author.

Is it for free and how can I get it?

YES! You can download it here, it is distributed under the GNU License.

Why should I use SpiSOP?

In brief, it is free, open source, easy and fast to set up and use, has established methods, calculates fast and automatically, is robust, does most of basic human sleep EEG analyses, allows you to replicate a lot of other analyses, lets you use any kinds of data format, makes it easy to repeat your analyses, just changing a parameter at a time and makes it easy to exchange and share your analyses. Thus someone else can perform and understand your analysis on their data. Did I mention that it can handle >1000 datasets as well as 1, and that the list goes on…

Why should I NOT use SpiSOP?

SpiSOP is obviously limited in functions and are tailored for most common sleep EEG analyses, nothing more.

If you like or heavily depend on graphical user interfaces (GUIs), where every step there is a click, SpiSOP is not what you are looking for (but see section: “Why is there no GUI (when I want to launch SpiSOP)?”).

If you want a great tailored support of the software or you cannot deal with limitations of open source software in development (e.g. putative bugs), then you should consider proprietary software (starting at about $10’000).

If you want to go beyond the basics and perform advanced EEG analyses, SpiSOP can be a good start, but not your ultimate goal.

If you want to do everything yourself, go ahead (but maybe you could consider contributing to SpiSOP instead).

Why is there no GUI (when I want to launch SpiSOP)?

SpiSOP is designed for fast, and replicateable processing of data and to be learned and shared easily in a sparse environment. Graphical User Interfaces complicate these goals:

- GUIs hide vital steps of the analysis protocol in redundant and repetitive clicks by the user, this complicates sharing, replication and is not fast.
- GUIs are not good in perfoming steps over and over, thus they complicate batch or pipeline processing and would require a second user interface for that (e.g. another command line) to also do that, this would require double functionality and is not a sparse environment.
- GUIs are in most of the cases (and belief it or not, this would be the case here) more complicated to handle, and thus to learn, especially to overview the full functionality of software, this makes it harder in sharing and fast progress too.

If you like arguments by authority, see the opinions of people “that get things done” using a computer (e.g. Linus Torvalds, the driving force behind Linux and Git). Part of their speed in doing things efficiently is not because they have learned a “complicated system of command line tools”, but because they focused their learning on the easy way of running software, that is with a few entered words from the command line, that are adaptable and reusable. Be couragious, hate the GUI, “Text trumps textures.” and “Closer to code is the easiest mode.”

That being said, GUIs are good for something, e.g. organizing functionality, or visualize complex data and processes or make decisions, and this is why the SpiSOP functions like browser,

freqpeaks, and hypcomp use a GUI.

Maybe in the future there will be a “launch GUI” at the start, but it hopefully this will focus on the “organizing the functionality” part.

How do I cite SpiSOP?

Frederik D. Weber (2013) SpiSOP tool(box) https://www.spisop.org

(peer reviewed publication is in preparation)

Can it do automated sleep scoring?

Not yet, but probably soon, please be patient. For now, SpiSOP includes a manual sleep-scoring program (*browser*). It aids scoring decisions by the scorer already by marking periods of slow-waves and spindles automatically to gain a faster decision on NonREM sleep stages. If you have any suggestions on open source work that exists and works already on automated sleep scoring, this would be great, just write shortly to spisop@spisop.org …. In the mean time please support for example Simon Kern to get an awesome sleep scoring software out and usable: project on reseachgate and github source.

What ADDITIONAL SOFTWARE is needed?

…For running the STANDALONE version:

1. MATLAB Compiler Runtime (MCR) version 8.2 (R2013b) is installed (an installer for your operating system like Windows, Mac, or linux) can be downloaded here. NOTE you do NOT need any Matlab license NOR do you need Matlab installed to run the standalone version. (but see https://www.mathworks.com/products/compiler/mcr/index.html)

…For running the Matlab SOURCE CODE:

- Matlab 2013a or greater (recommended is 64 bit version running on Linux,Windows or Mac)

Required toolboxes:

- *Signal Processing Toolbox* (signal)
- *Statistics Toolbox* (stats)
- *DSP System Toolbox* (dsp)

Optional toolboxes:

- *Parallel Computing Toolbox* (distcomp) is needed for parallel computing
- *Curve fitting Toolbox* (curvefit) is needed for faster and more memory efficient smoothing of signals.
- *Compiler* for compiling the source code as a standalone

Note that the SpiSOP toolbox can be made running in Matlab without these required toolboxes, however this reduces functionalities, especially in the used filters and the accumulation of output or an integrated use of low level functions within matlab scripts.

- (A release of the FieldTrip (fieldtrip) toolbox. ***A***

working version is already included, that was modified for use of SpiSOP toolbox)
- A nice text editor, however standard text editors are sufficient.

What additional FREEWARE is useful to install?

- (recommended) EDFbrowser – to view, check and edit edf files (source)
- (recommended) LibreOffice (Calc) – for opening the output files and convert to Excel files if necessary.
- ***[R]*** – for statistical analyses and plotting of the results
    - https://www.r-project.org/
    - Rstudio https://www.rstudio.com/
    - ggplot2 http://ggplot2.org/
    - Quick-R http://www.statmethods.net/index.html (by Robert I. Kabacoff)
- **Other related projects:**
    - Braininterface or BF++ to tool collection for extensive BCI and EEG analyses in a GUI
    - Polyman (like EDFbrowser, checks EDF files) EDF viewer and checker including sleep scoring, export of Scoring info to textfile over EDFbrowser (export of annotations), created by Marco Roessen and Bob Kemp. Since 2014 maintained by Diego Alvarez-Estevez (closed source)
    - Collection of free tools at http://www.edfplus.info/
    - SignalPlant – tool collection for EEG analyses in a GUI
    - sleepSMG – sleep scoring tool (requires Matlab and

EEGLab) by Stephanie Greer & Jared Saletin
- Somnonetz viewer, share and view EDFs in the browser instantly by Maximilian Beier
- EDF viewer web browser-based viewer (client side via or offline) by Bilal Zonjy source.
- EEGLab (requires Matlab) – all you need for EEG analyses, great tool
- pyrem by Quentin Gaissmann
- MODA by Benjamin Yetton
- wonambi formerly sleepscoring and phypno by Giovannni Piantoni and Jordan O'Byrne
- SWA toolbox by Armand Mensen (see the cool paper)
- FASST by Christophe Phillips
- SPINKY by Tarek Lajnef + Christian O'Reilly (see their nice paper)
- Sleep (user-friendly sleep analyses) by Etienne Combrisson + Raphael Vallat et al. (see their cool paper and contribute to the source)
- yasa yet another spindle algorithm (in python) by Raphael Vallat
- SEV (matlab tool for sleep scoring, spindle and power density, PLM analysis and more) by Hyatt More (IV) with source
- SpindleTool souce by Sara Mariani
- edfview an sleep scoring fork of the EDFbrowswer by Justus Schwabedal
- Spindler a EEGlab, Matlab based spindle detector by VisLab
- McSleep a Matlab based multichannel EEG detector by Ankit Parekh
- sleepr an [R] package for sleep analysis by Paul Bouchequet
- sleepanalyzer Matlab scripts for sleep EEG power analysis, event detection (spindles, ripples, slow waves, theta waves…), PAC etc. by Til Ole Bergmann
- Luna by Shaun Purcell
- also see more tools listed on sleepdata.org

What hardware is needed?

Computer with RAM to hold dataset at least two times the biggest dataset (recommended at least 4 GB RAM) and a fast hard drive (recommended solid state drive SDD). If using parallel computing consider the same amount of RAM is needed for every CPU! For STANDALONE version you need a 64bit version of your operating system running (like on most modern computers, e.g. Windows 7 64bit)

What is needed to run SpiSOP and in what format?

1. Datasets (i.e. the raw data, and their header files if they exists, and optionally, marker files) supported formats are listed here **http://fieldtrip.fcdonders.nl/dataformat** ALSO OpenBCIv3 SD card files can be converted using with the conversion tool in the subfolder *~/utilitiy/convert_data/OpenBCIv3/convert_SD_file*
2. Hypnogram files containing scored epochs for each data as scored with the program ***Schlafaus*** or with the ***spisop browser***:

Hypnogram contains two columns (separated by space, tab or comma, however tab is preferred) to mark sleep stages.

First column can contain values representing sleep stages, i.e.

0 – Wake,

1 – Stage 1,

2 – Stage 2,

3 – Stage 3,

4 – Stage 4,

5 – REM,

8 – Movement Time.

Second column can contain values marking movement arousals and artifacts, i.e.

0 – no arousals or artifacts,

1 – arousal or artifacts,

2 – arousal or artifacts due to Movement Time.

3 and greater – mark of artifact or “do not use”.

Here is an **example hypnogram file**

NOTE: If you scored in another Program there is a sample of conversion scripts to bring the scoring files in the required format (see folder ~/utilitiy/convert_sleep_scoring). This includes Domino (Somonmedics: *.txt files exported), Brainvision (*.vmrk marker files), SigmaSleep (*.sta files)

Hint: All filenames and filepaths should NOT contain commas (i.e. “,”) and avoid spaces (i.e. “ ”).

What if my computer has not

enough memory or I get an out of memory error?

Be sure to close all the other applications running on the computer and using the systems memory.

Not using the parallel computation option is then advised (or using less parallel processes at the same time).

Reducing the sampling rate of the data (in the parameters) lessens further memory load.

Restricting the number of channels to a specific subset of channels (one can later also append the results of those channel subsets).

One can also consider to detect in less sleep stages and later merge the results (e.g. if NonREM events are searched then first search S2 events, then SWS events, later merge the results of both analysis, here the hypvals function can help you cut your sleep scoring in different time periods).

What are the KEYBOARD SHORTCUTS for scoring in the BROWSER?

KEYBOARD SHORTCUTS ONLY WILL WORK IF MOUSE WAS CLICKED IN THE DISPLAY AREA!

(This is a known bug in MATLAB GUIs with no perfect solution was found yet)
Keyboard shortcuts are:

- Scoring:
    - 0 or W: Wake (W)
    - 1: Stage 1 (S1/N1)
    - 2: Stage 2 (S2/N2)
    - 3: Stage 3 (S3/N3)
    - 4: Stage 4 (S4)
    - 5 or R: REM (R)
    - 7 or D: Delete Stage
    - 8: Movement Time (MT)
    - 9: Movement Arousal/Epoch with artifact(s) (MA, additional)
- Control:
    - Left-arrow: go to previous epoch
    - Right-arrow: go to next epoch
    - Up-arrow: decrease scaling (zoom in all channels)
    - Down-arrow: increase scaling (zoom out all channels)
    - Y: Vertical Y-scaling basis (equivalent to scaling factor = 1) (dialog)
    - Shift + C: Channel settings select, color, scaling, order, focus on EEG, EMG, EOG channels
    - Shift + Up-arrow: skip up through undisplayed channels
    - Shift + Down-arrow: skip down through undisplayed channels
    - Shift + Q: Quit Program
    - P: Power spectrum of the current epoch
    - X: Hide/Display menue bar
    - T: Select epoch to jump to (dialog)
    - L: Set Thresholds (dialog)
    - Shift + H: Shortcut help
- Scoring aids:
    - M: enable/disable marking, cummulative time
    - N: enable/disable spindle signal display (EEG filtered in spindle band)
    - V (formerly E): enable/disable display of pre-readin Events, if processed already

        - Q: delete previously made marking
        - E (formerly R): enable/disable Ruler (aka “the Score-ship”)
        - J: enable/disable spindle marking (aka “the Score-ship”)
        - K: enable/[polartity-switch]/disable K-Complex
        - or Slow oscillation marking (aka “the Score-ship”)
        - G: enable/disable display of the time grid
        - Z: enable/disable zooming (double-click = return to full view)
            - left-click: zoom in
            - Alt + left-click zoom out
            - double-click: fully zoom out
        - A: switch to next artifact marking type
        - Ctrl + 1..9 OR Alt + 1..9: skip to next artifact type 1..9
    - Saving/Export:
        - Shift + S: Save Session (dialog)
        - Shift + O: Open Session (dialog)
        - Shift + I: Import Hypnogram (dialog)
        - Shift + E: Export Hypnogram (dialog)
    - Non-Scoring:
        - Shift + Left-arrow: decrease epoch length
        - Shift + Right-arrow: increase epoch length
        - H: Horizontal scaling
        - (S: switch highlighting/marking style, deprecated)

Sleep staging automatically skips to the next epoch if this epoch was not scored with a sleep stage of MT and was not scored before (just scoring MA will not skip automatically).

How can I help or contribute

to SpiSOP?

You want to help? Great! You can edit the source (pull requests), give constructive feedback on the use or things to improve, or encourage the authors to continue, please contact spisop@spisop.org

I found a BUG, what to do?

No Software is perfect, SpiSOP is no exception? Maybe soon there will be a bug report system. Until then, please try to describe the bug and please sent to spisop@spisop.org, the author will be pleased to get a good bug report. (when does the but occur, what is acutally the bug, how can I recreate it. and maybe include the error messages and command line output)

If something goes wrong or there are errors?

A suggestion, try to precede in this order:

1. read the error message and try to understand it (in the command line, the last statements saying "…error…"), it will probably tell you something or how to fix parameters!
2. only SOURCE version: check if *spisop_init.m* is correct and has all the values set correctly and if it has run line by line (try again if possible)
3. check if direct parameters of function are correct (e. g. filenames etc.)

4. check if (core) parameter file has correct parameters (e. g. filenames of datasets file etc.) and no spaces around the commas
5. ~~(deprecated)check if files given in parameter file have no empty lines inbetween → remove empty lines inbetween~~
6. ask someone you know with more or similar experience to look over (explaining the problem often helps)
7. ask Frederik D. Weber (aka Freddy) for help, he likes to help: spisop@spisop.org

How reliable, common or established are the Methods, and how do they compare to other work?

Science is progressing, and with it the approaches to tackle new arising questions. There are therefore no established Methods. However the methods expand on many previous reports of top of the edge researchers, and can be tuned to replicate most of basic and advanced sleep research reports. In addition, the parameters can be fast and flexibly configured to address robustness of the chosen approach. Even though differences to other algorithms exist, the author’s opinion is that there are no established gold standards, and that it is more important to have a robust measure that actually aids in answering research questions. For example investigating THAT spindle algorithms compare different to false gold standards like highly subjective visual scorings, or against each other is not of scientific value for itself. Instead investigating HOW algorithms differ is also deemed not helpful. However investigating WHAT are the consequences of changing algorithms AND their parameters in terms of answering research questions

is what should be done. Two algorithms can be very similar and result in different conclusions of a hypothesis (e.g. slow vs. fast spindles), and they can be very different but still result in the same conclusion (e.g. both results correlate equally to a behavioral measure). Finally, most algorithms differ on so many fundamental parameters that they are not worth being compared in their power to answer a research question. In science you only change one parameter in a controlled manner and observe its influence on the results before moving to the next. In essence, SpiSOP just provides this approach, you are welcome to also compare to other methods.

What to know about sleep data before starting any analyses?

SEE DOCUMENTATION