Reproducible projects with python and snakemake – a subjective user perspective

Bi-weekly ESR meetings

25th Feb 2022 - David

Why interesting?

Research today:

- many projects in parallel, from first steps to publication, several years can pass
- people leave their lab after few years
- different labs work on the same research topic and need to share their data analysis workflows with the results to enable collaboration

➢ Reproducibility is needed

➢ Things must work out of the box to be re-used

Why interesting?

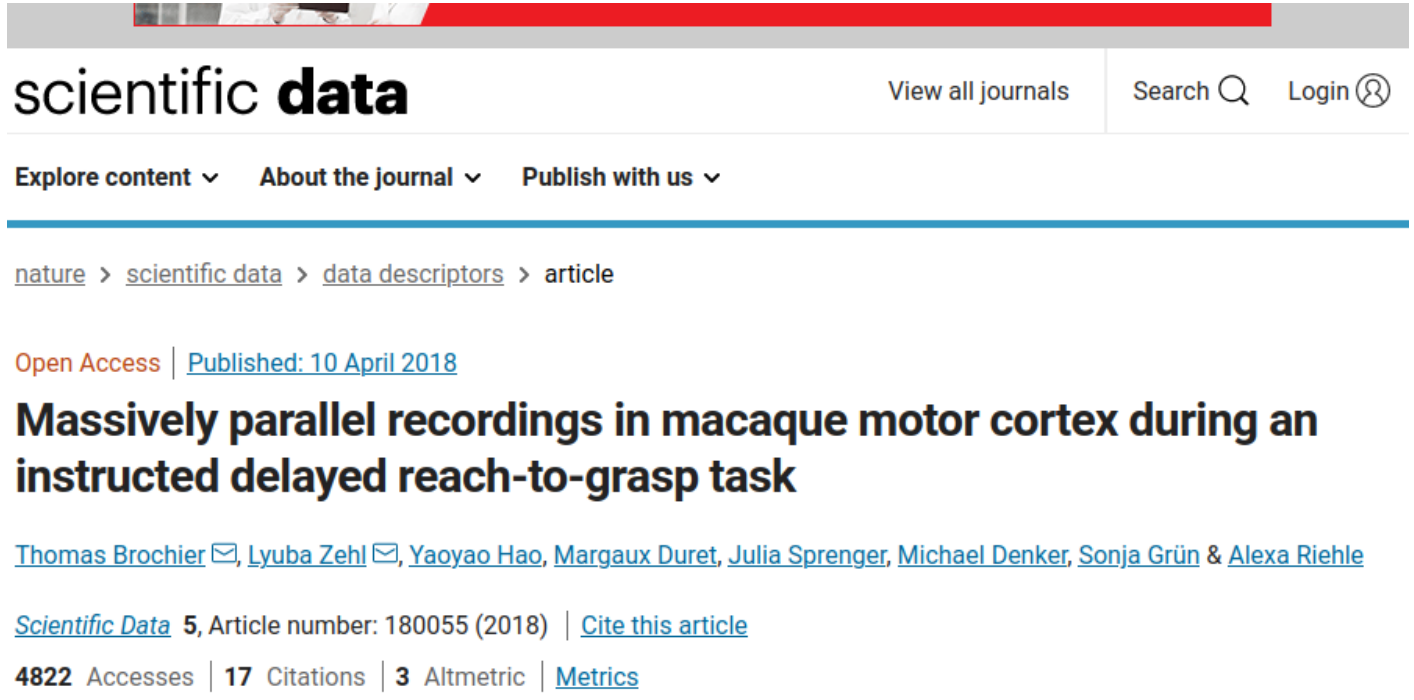
- Reproducibility is needed
- Things must work out of the box to be re-used

One solution: Snakemake workflows

- description via a human readable, Python based language
- creation of reproducible and scalable data analyses
- scaling from laptop to cluster computing without modification of the workflow description
- optionally manages the software environment to execute individual steps of the workflow (via conda, docker)

Example project

- Neural data from a Utah array (96 electrodes) in monkey motor cortex



scientific **data**

View all journals | Search | Login

Explore content | About the journal | Publish with us

nature > scientific data > data descriptors > article

Open Access | Published: 10 April 2018

Massively parallel recordings in macaque motor cortex during an instructed delayed reach-to-grasp task

Thomas Brochier, Lyuba Zehl, Yaoyao Hao, Margaux Duret, Julia Sprenger, Michael Denker, Sonja Grün & Alexa Riehle

Scientific Data **5**, Article number: 180055 (2018) | Cite this article

4822 Accesses | **17** Citations | **3** Altmetric | Metrics

Example project

- Neural data from a Utah array (96 electrodes) in monkey motor cortex

Table 3 Overview of files (names, size, and content) for each provided dataset of monkey L and N.

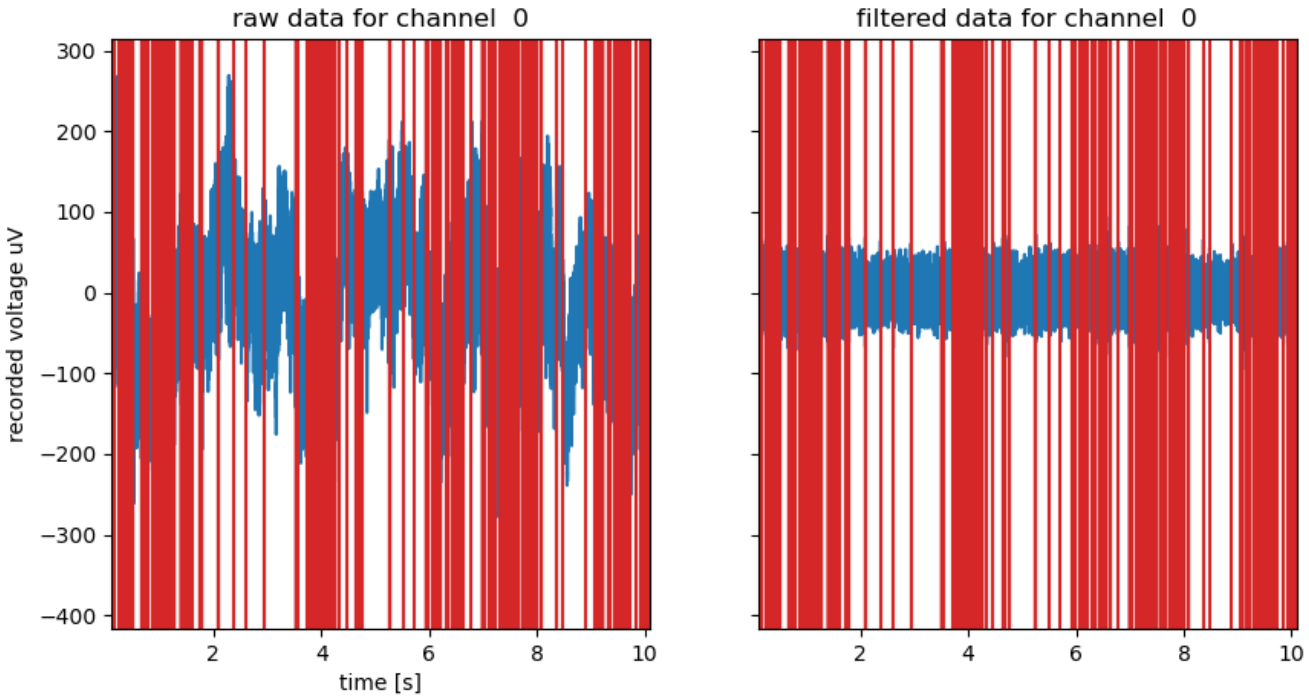
From: [Massively parallel recordings in macaque motor cortex during an instructed delayed reach-to-grasp task]

monkey L		
file names	**file size**	**file content**
l101210-001.ccf	108.2 kB	cerebus configuration file
l101210-001.nev	287.7 MB	digital events, unsorted spikes times & waveforms
l101210-001-02.nev	287.7 MB	digital events, sorted spikes times & waveforms
l101210-001.ns2	8.5 MB	analog signals of object sensors
l101210-001.ns5	4.1 GB	raw neuronal signal
l101210-001.odml	2.7 MB	metadata
monkey N		
file names	**file size**	**file content**
i140703-001.ccf	187.1 kB	cerebus configuration file
i140703-001.nev	168.3 MB	digital events, unsorted spikes times & waveforms
i140703-001-03.nev	168.3 MB	digital events, sorted spikes times & waveforms
i140703-001.ns2	204.7 MB	analog signals of object sensors and LFP signals
i140703-001.ns6	5.8 GB	raw neuronal signal
i140703-001.odml	2.3 MB	metadata

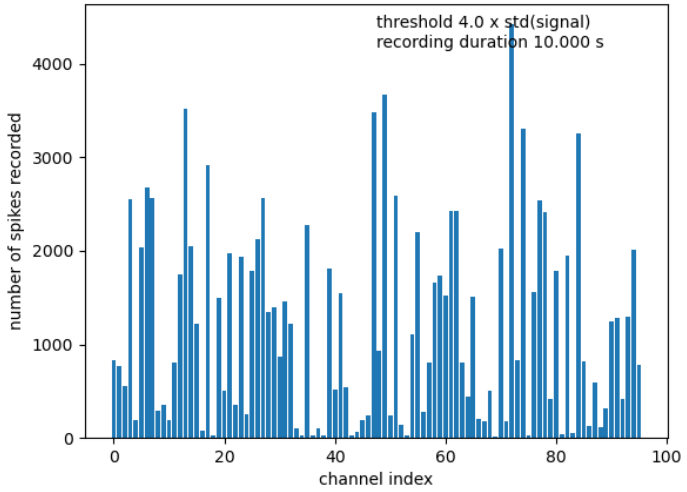
Outline

- Neural data from a Utah array (96 electrodes) in monkey motor cortex
- Workflow steps to perform on the dataset:
  - Download
    - Available on online repository
  - Preprocess
    - Bandpass filter the voltages recorded at all electrodes
  - Analyze
    - Extract spikes based on thresholding the filtered voltage signals
    - Count the number of spikes recorded per electrode
  - Plot results
    - Plot raw signal and filtered signal with spikes
    - Plot histogram of the number of spikes per channel
    -

Target



- Figure 1: raw signal, filtered signal, spikes
- Figure 2:
- Parameters:
- E.g. filename, filter frequencies, threshold for spike detection, file-format of the plots



Workflow Parameters

- Neural data from a Utah array (96 electrodes) in monkey motor cortex
- Workflow steps to perform on the dataset:
  - Download
    - Available on online repository
    - **filename**
  - Preprocess
    - Bandpass filter the voltages recorded at all electrodes
    - **lower & upper filter frequency**
  - Analyze
    - Extract spikes based on thresholding the filtered voltage signals
    - Count the number of spikes recorded per electrode
    - **threshold for spike detection**
  - Plot results
    - Plot raw signal and filtered signal with spikes
    - Plot histogram of the number of spikes per channel
    - **plot parameters, e.g. file format**

  (many more parameters not listed here)

Workflow Parameters

program to perform the step

- Neural data from a Utah array (96 electrodes) in monkey motor cortex
- Workflow steps to perform on the dataset:
  - Download **curl using bash shell command**
    - Available on online repository
    - **filename**
  - Preprocess **python script**
    - Bandpass filter the voltages recorded at all electrodes
    - **lower & upper filter frequency**
  - Analyze **python script**
    - Extract spikes based on thresholding the filtered voltage signals
    - Count the number of spikes recorded per electrode
    - **threshold for spike detection**
  - Plot results **python script**
    - Plot raw signal and filtered signal with spikes
    - Plot histogram of the number of spikes per channel
    - **plot parameters, e.g. file format**

  (many more parameters not listed here)

workflow recipe

in snakemake:

- Definition of target files
  - here we want to produce 2 plots:
  - Raw signal, filtered signal with spikes
  - Histogram of spike count per electrode
- Download **curl using bash shell command**
  - Available on online repository
  - **filename**
- Preprocess **python script**
  - Bandpass filter the voltages recorded at all electrodes
  - **lower & upper filter frequency**
- Analyze **python script**
  - Extract spikes based on thresholding the filtered voltage signals
  - Count the number of spikes recorded per electrode
  - **threshold for spike detection**
- Plot results **python script**
  - Plot raw signal and filtered signal with spikes
  - Plot histogram of the number of spikes per channel
  - **plot parameters, e.g. file format**

(many more parameters not listed here)

Open | Snakefile_heuristic.txt ~/Documents/meeting_presentations/2022-02-23_Entrain_ESR_meeting/my_example | Save

Snakefile_heuristic.txt | config.yml

```
configfile: "config.yml"

rule all:
    input:
        -> list of all target-filenames with their name including the employed parameters
        in snakemake syntax:
                                "wildcard"1          "wildcard"2                  "wildcard"3
             results/hist___file_{file_name}___filter{freq-range}___spike-threshold_{thresh}.png
        e.g. results/hist___file_i140703-001___filter [500,7500] ___spike-threshold_   3    .png

rule plot_figures:
    input:
        "{file_name}.ns6",
        "data/filtered_signal_RELEVANT-PARAMETERS-HERE.npy",
        "data/spiketrains_RELEVANT-PARAMETERS-HERE.npy",

    output:
        "results/signal__RELEVANT-PARAMETERS-HERE}.png",
        "results/hist___file_{file_name}___filter{freq-range}___spike-threshold_{thresh}.png"
    script:
        "scripts/plot.py"

rule analyze:
    input:
        "data/filtered_signal__RELEVANT-PARAMETERS-HERE.npy"
    output:
        "data/spiketrains_RELEVANT-PARAMETERS-HERE.npy"
    script:
        "scripts/analyze.py"

rule preprocess:
    input:
        "{file_name}.ns6"
    output:
        "data/filtered_signal_RELEVANT-PARAMETERS-HERE.npy"
    script:
        "scripts/preprocess.py"

rule download_data:
    output:
        "{file_name}.ns6"
    params:
        link = 'download-data-here.com'
    shell:
        "curl -L {params.link} > {output}"
```

workflow configuration

Select the parameters to execute your workflow

- **Filename** to download
- **lower & upper filter frequency** for the bandpass filtering
- **threshold for spike detection**
- **plot parameters, e.g. file format**

(many more parameters not listed here)

Open | config.yml ~/Documents/meeting_presentations/2022-02-23_Entrain_ESR_meeting/my_example | Save

Snakefile_heuristic.txt | config.yml

```yaml
file_name: ['i140703-001']

# slice loaded for analysis
tstart: [0.1]
tstop: [10.1]

# bandpass filter frequency bounds
lowcut: [500,500]
highcut: [5000,7500]

# threshold for spike detection (factor of std(filtered_signal))
spike_threshold_std: [3,4]
```

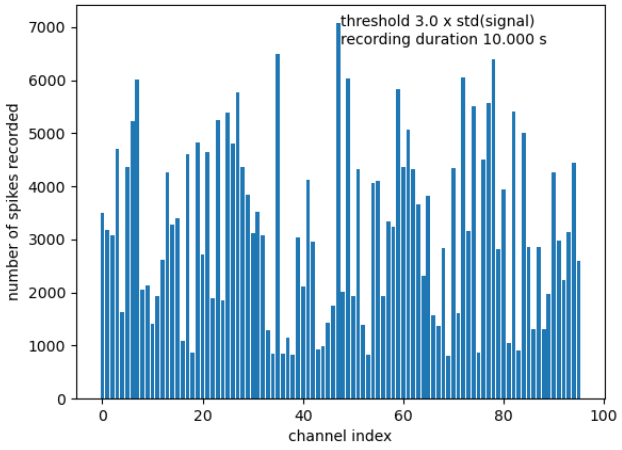
workflow execution

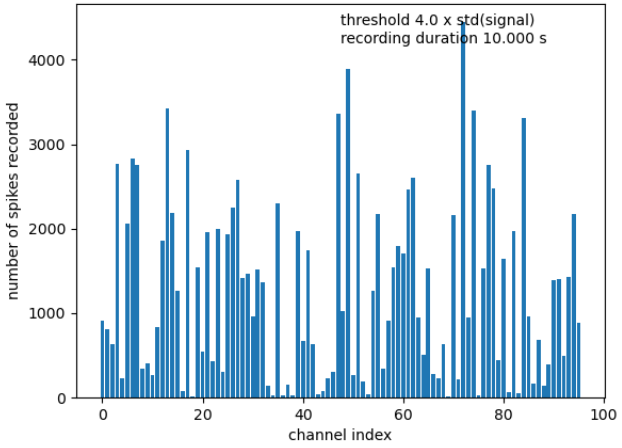
- Generation of all combinations of figures related to the parameter range
- When expanding the parameter range, snakemake automatically manages which parameter-sets have been already calculated and only performs the necessary steps

```
(snakemake_ephy_example) david@t490:~/Documents/meeting_presentations/2022-02-23_Entrain_ESR_meeting/
my_example $ ls
base-mypackage  data              results  Snakefile
config.yml      i140703-001.ns6   scripts
(snakemake_ephy_example) david@t490:~/Documents/meeting_presentations/2022-02-23_Entrain_ESR_meeting/
my_example $ snakemake
Provided cores: 1
Rules claiming more threads will be scaled down.
Job counts:
        count   jobs
        1       all
        4       analyze
        1       download_data
        4       plot_figures
        2       preprocess
        12
```

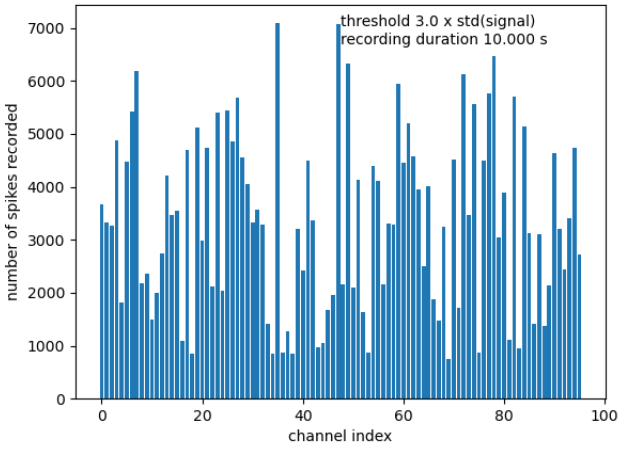
workflow results (only histograms)

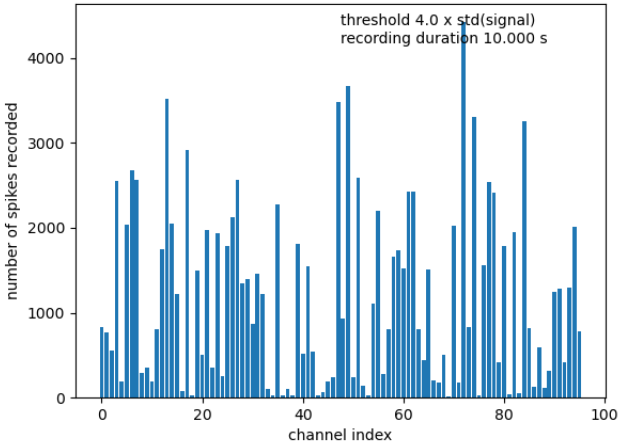
Bandpass filter range 500Hz-5kHz





Bandpass filter range 500Hz-7.5kHz





Additional features

- software management can be integrated
  - execution of workflow steps with individually defined conda-environments or docker containers
- Scalable from your laptop to a cluster via execution flags
  - E.g. execute steps in parallel on X cores:
    `snakemake –cores X`
  - Run your workflow on a computing cluster with job scheduling systems, e.g. slurm:
    `snakemake --cluster "sbatch -J snake" --jobs 95`

Thanks for your patience and interest ☺

For those interested, we can go into the details now ;-)