

Open in app ↗

Sign up

Sign In



Published in A Beginner's Guide to Brain-Computer Interfaces

You have **2** free member-only stories left this month.

[Sign up for Medium and get an extra one](#)



Tim de Boer

Follow

Aug 22, 2022 · 3 min read · ✨ · 🎧 Listen



Save



How To Easily Find Online EEG Datasets Using MOABB

A Beginner's Guide to Brain-Computer Interfaces (part 5)



7





EEG data is cool! But how to get it when you cannot measure the data yourself?

Starting a hobby project in the field of brain-computer interfaces can be hard. How to get your data? Buying an EEG device can already be a roadblock in your hobby project while you haven't even started yet with the fun part: coding!

Luckily, there are some great public EEG datasets freely available online. However, datasets useful for your project may be hard to find. Moreover, these datasets can be in a weird format, and some important additional information may not be given..

Luckily, these problems are solved by the great developers of **MOABB: The Mother of all BCI**

Benchmarks. MOABB is an open science project, aiming to build a benchmark of popular BCI algorithms applied on freely available EEG datasets. The code is available on [Github](#), and a [documentation](#) is available as well.

This blog post is meant to reach BCI enthusiasts struggling with starting their BCI project, who did not know about MOABB. I sure wish I had learned about MOABB earlier in my project, as I struggled a lot with quickly testing some ideas about my algorithms without the need of collecting data myself.

This post is not an exhaustive explanation about everything possible with MOABB. Only the bare essentials to get started with MOABB quickly are explained. A more extensive tutorial can be found on this [Github page](#), and for more information about MOABB, I gladly refer you to their [documentation](#).

Let's go over the basics and start coding!

Installation

Installation is simple. Write in your command prompt the following:

```
pip install moab
```

And you're set!

Getting the data

Say, you want to collect some data about left hand and right hand motor imagery. Datasets with these two classes are pretty popular, and MOABB has a build in paradigm for them. We can search for the available datasets of this paradigm using only 4 lines of code!

```
from moabb.paradigms import  
LeftRightImagery  
paradigm = LeftRightImagery()  
for dset in paradigm.datasets:  
    print(dset)
```

Giving as output:

```
<moabb.datasets.bnci.BNCI2014001 object at  
0x7fdf93cb5240>  
<moabb.datasets.bnci.BNCI2014004 object at  
0x7fdf93cb5278>  
<moabb.datasets.gigadb.Cho2017 object at  
0x7fdf93cb52b0>  
<moabb.datasets.Lee2019.Lee2019_MI object  
at 0x7fdf93cb5208>
```

```
<moabb.datasets.mpi_mi.MunichMI object at 0x7fdf93cb5320>
<moabb.datasets.physionet_mi.PhysionetMI
object at 0x7fdf93cb52e8>
<moabb.datasets.schirrmeister2017.Schirrme
ister2017 object at 0x7fdf93cb5390>
<moabb.datasets.bbc_i_eeg_fnirs.Shin2017A
object at 0x7fdf93cb5358>
<moabb.datasets.Weibo2014.Weibo2014 object
at 0x7fdf93cb5198>
<moabb.datasets.Zhou2016.Zhou2016 object
at 0x7fdf93cb5160>
```

Next, we load these datasets:

```
from moabb.datasets import (
    BNCI2014001,
    BNCI2014004,
    Cho2017,
    Lee2019_MI,
    MunichMI,
    PhysionetMI,
    Schirrmeister2017,
    Shin2017A,
    Weibo2014,
    Zhou2016
)

all_datasets = [BNCI2014001(),
                  BNCI2014004(),
                  Cho2017(),
                  Lee2019_MI(),
                  MunichMI(),
                  PhysionetMI(),
```

```
Schirrmeister2017(),  
Shin2017A(),  
Weibo2014(),  
Zhou2016()]
```

As these datasets differ a lot in their setup, you may benefit from reading the descriptions of the datasets in the [MOABB documentation](#).

Evaluation

Now we have all our data! Now, you can go ahead and try one of the most popular pipelines in motor imagery BCIs: CSP with LDA (More on CSP in later blog posts!)

```
from mne.decoding import CSP  
from sklearn.discriminant_analysis import  
LinearDiscriminantAnalysis as LDA  
from sklearn.pipeline import Pipeline  
from moabb import set_log_level  
from moabb.evaluations import  
CrossSessionEvaluation  
import warnings  
  
pipelines = {}  
pipelines["csp+lda"] = Pipeline(steps=  
[('csp', CSP(n_components=8)),  
  
('lda', LDA())])
```

```
set_log_level("error")
warnings.filterwarnings("ignore")

CSP_Eval= CrossSessionEvaluation(
    paradigm=paradigm,
    datasets=all_datasets,
    n_jobs=-1,
    suffix="workshop",
    overwrite=False
)
results = CSP_Eval.process(pipelines)
print(results)
```

Easy as that, you will have quick results about your newly developed algorithm or pipeline!

I hope this quick tutorial helped you getting starting in the field of BCI. Please check out the [publication page](#), where more practical BCI tutorials like this one will be posted in the future, and give me a follow to be notified for further posts!

Happy coding!

Neurotechnology

Brain Computer Interface

Eeg

Data Collection

Open Source

Enjoy the read? Reward the writer.^{Beta}

Your tip will go to Tim de Boer through a third-party platform of their choice, letting them know you appreciate their story.

Give a tip

Get an email whenever Tim de Boer publishes.

Your email



We couldn't process your request. Try again, or contact our support team.



Subscribe

By signing up, you will create a Medium account if you don't already have one. Review our [Privacy Policy](#) for more information about our privacy practices.

Get the Medium app

