

(GIS) OSmOSE

Open Science meets
Ocean Sound Explorers

User guide of OSmOSE analytics platform

OSmOSE Working Report

Authorship This document was drafted by

- Dorian Cazau¹⁾

belonging to the following institute (at the time of their contribution): 1) Lab-STICC, ENSTA Bretagne.

Document Review Though the views in this document are those of the authors, it was reviewed by a panel of acousticians before publication. This enabled a degree of consensus to be developed with regard to the contents, although complete unanimity of opinion is inevitably difficult to achieve. Note that the members of the review panel and their employing organisations have no liability for the contents of this document.

The Review Panel consisted of the following experts (listed in alphabetical order):

- Maelle Torterotot¹⁾

belonging to the following organisms / research institutes (at the time of their contribution): 1) Lab-STICC, IMT Atlantique.

Last date of modifications March 25, 2022

Recommended citation "User guide for OSmOSE notebooks", OSmOSE working report (version dating from March 25, 2022, distributed openly on <https://osmose.xyz/>)

Future revisions Revisions to this document will be considered at any time, as well as suggestions for additional material or modifications to existing material, and should be communicated to Dorian Cazau (dorian.cazau@ensta-bretagne.fr).

Document and code availability This document has been made open source under a Creative Commons Attribution-Noncommercial-ShareAlike license (CC BY-NC-SA 4.0). All associated codes have also been released in open source and access under a GNU General Public License and are available on github (<https://github.com/Project-ODE>).

Contents

1	Pre-requirements	4
1.1	Getting your logins	4
1.2	Connecting to Datarmor	4
1.3	Executing a notebook on Datarmor	4
2	How using our platform ?	6
2.1	Use case 1 : Importing a new dataset	6
2.2	Use case 2 : Creating a set of spectrograms for an APLOSE campaign	7
2.3	Use case 3 : Creating a long-term averaged spectrogram	7
3	How contributing to the development of our platform ?	7
3.1	Contribution 1 : Proposing new features and reporting bugs	7

1 Pre-requirements

1.1 Getting your logins

Before, you need to have an account on Datarmor, that will provide you with

- extranet logins (username + password)
- intranet logins (username + password)

To get an account, please contact your IT assistant if you are from an institute partner of Datarmor, otherwise contact dorian.cazau@ensta-bretagne.fr.

1.2 Connecting to Datarmor

Datarmor can be accessed in two ways.

- via VPN and ssh following the document “[Extranet access to DATARMOR.](#)” provided in annex ;
- via a user-friendly portal at https://domicile.ifremer.fr/dana-na/auth/url_default/welcome.cgi.

Be sure you can connect to Datarmor through both accesses before continuing.

1.3 Executing a notebook on Datarmor

Most of our services require the use of Jupyter notebooks deployed on Datarmor. So let's see how to use them.

Step 1: launch a jupyterhub session

Now you can click on JUPYTER on the portal, enter your intranet logins and click on Start My Server. You will then have to select a job profile, corresponding to the computer resources on which you wish to run your jupyterhub. By default, select **Datarmor-Notebook-4cores,8GBRAM,2hours**. You are now on the Jupyter-Hub of Datarmor!

Step 2: copy and paste OSmOSE notebooks in your datahome

OSmOSE notebooks are located on our workspace at `/home/datawork-osmose/`. You will have to make a copy of them into your personal workspace. One way of doing that is to open a terminal from your jupyterhub portal (see figure 1) and run the following command:

```
>> cp -R /home/datawork-osmose/osmoseNotebooks_v0/ . ; chmod -R 700 osmoseNotebooks_v0/
```

Step 3: use a notebook

To open a notebook, just double click on it. If your notebook kernel is not automatically set to `allohaEnv` (see Fig. 2), change it in Kernel → Change Kernel → Python [conda env:allohaEnv]

Step 4: list of must-do and things to keep in mind

Except that, the use of our notebooks should be straight forward even for users without basic skills in Python. Just be careful with a few “Must and Must not do” :

- Do not implement computationally expensive processing codes on notebooks. This should be done with job batch - better that you contact dorian.cazau@ensta-bretagne.fr before ;

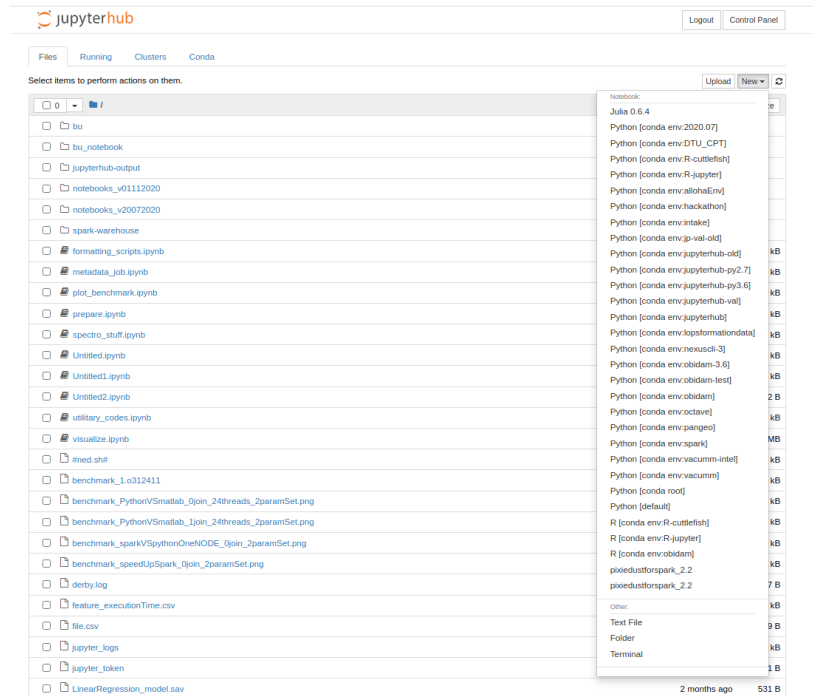


Figure 1: Open a terminal from jupyterhub.

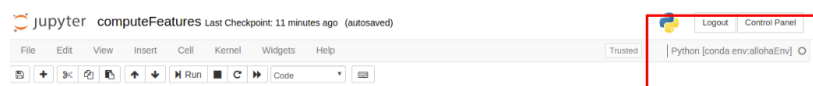


Figure 2: Notebook kernel need to be set to allohaEnv.

- Avoid underusing your notebook, especially do not forget to close your notebook session once you are done. To do so, follow this procedure :

1. on the Jupyter Hub main page → click tab Running → click on all Shutdown
2. then click on the button Logout on the Jupyter Hub main page

- Avoid running several notebooks simultaneously ;
- Only run one analysis at a time on a same dataset !

And keep in mind that :

- In case of a long computation (for example, resampling high-frequency data), your notebook might close itself due to its limited wall-time. This is not a problem, all your computation will keep running safely, however you will not be able to track progress of them through the notebook ... maybe it is time for you to learn the following command to be run in the jupyter hub terminal :

```
>> qstat - u myusername
```

2 How using our platform ?

2.1 Use case 1 : Importing a new dataset

Let's start with some constraints you will have to comply with :

- all your audio files must be .wav files ;
- all your audio files must have the same characteristics, especially duration and sample frequency ;
- so far, we are dealing with files with a maximal duration of 388 minutes and a maximal volume of 230 MB ; if your files happen to exceed drastically one or both of these values, please contact dorian.cazau@ensta-bretagne.fr before going further.

Then follow these instructions to import your dataset :

1. build the dataset folder `/home/datawork-osmose/dataset/mydatasetname/raw/audio/original` , where you have to replace `mydatasetname` by your dataset name (can be done through FileZilla or SSH terminal) ;
2. upload your audio files in this folder. This step **MUST** be done through FileZilla using datacopy as host ;
3. download the file `/home/datawork-osmose/templates/timestamp.csv`, fill it for your campaign respecting the exact same format, and then put it in `/home/datawork-osmose/dataset/mydatasetname/raw/audio/original` (can be done through FileZilla or SSH terminal). Check our code snippets in Annex for this operation!
4. in case you have a mobile hydrophone, download the file `/home/datawork-osmose/templates/gps_depth.csv`, fill it for your campaign respecting the exact same format **and variable names** (depth is not mandatory), and then put it in `/home/datawork-osmose/dataset/mydatasetname/raw/auxiliary/` after creating the corresponding folder (can be done through FileZilla or SSH terminal) ;
5. in case you have auxiliary data, download the file `/home/datawork-osmose/templates/ECMWF_1H.csv`, fill it for your campaign respecting the exact same format, and then put it in `/home/datawork-osmose/dataset/mydatasetname/raw/auxiliary/` after creating the corresponding folder (can be done through FileZilla or SSH terminal) ;

6. in case you have any other data or documents you want to store along with your dataset¹, please put them in `/home/datawork-osmose/dataset/mydatasetname/other/` after creating the corresponding folder (can be done through FileZilla or SSH terminal) ;
7. open and execute the notebook `build_dataset.ipynb` from our notebook package (for information on notebooks see section 1.3).

2.2 Use case 2 : Creating a set of spectrograms for an APLOSE campaign

This will be done in two steps :

1. use the notebook `fileScaleAnalysis.ipynb` to create the set of spectrograms for your campaign
2. without waiting the end of the generation of your spectrograms, use the notebook `buildDatasetsCSV_temporaryNotebook.ipynb` to create an internal file that will be used to import your dataset into our annotation tool APLOSE ;

2.3 Use case 3 : Creating a long-term averaged spectrogram

Use the notebook `datasetScaleAnalysis.ipynb` to create any spectrogram at a time scale longer than your audio file duration. You can also use this notebook to recover raw features at a time scale longer than your audio file duration, just set `newFeaturesG0` to `True` and set your time aggregation scale with `timeResoAggregation`. You can also join auxiliary variables along your spectrograms and/or features through `aux_variable`.

3 How contributing to the development of our platform ?

3.1 Contribution 1 : Proposing new features and reporting bugs

Any proposition of new features and bug reporting must be done through the following documents hosted on our google drive :

- for the backend : <https://docs.google.com/document/d/1U7Y4YEQaowNztZXBjCXHdH0VnXzjwp4pIa0JrZ0mYGI/edit#>
- for the frontend : https://docs.google.com/document/d/1jZ_aUuW6qAok_8Yku1yB0iHv2cVYD6ZT/edit

In case you need access to these documents, please send a mail to Dorian Cazau (dorian.cazau@ensta-bretagne.fr).

ANNEX

Code snippets

We provide the following set of code snippets here `/home/datawork-osmose/codeSnippets/`, useful for different operations on our platform:

- `create_timestampCSV.ipynb` : to be used in Use Case 1 Step 3, i.e. to generate the file `timestamp.csv` from your imported audio files.

¹Better to keep it all in one place!

Extranet access to DATARMOR

When you are outside your institute (IFREMER/SHOM/IUEM/UBO/ENSTA Bretagne/Ecole Naval) you have to use Pulse Secure to be able to use a ssh connection to DATARMOR.

There are 3 steps :

- Installation
- Using Pulse Secure
- Access to datarmor

1 Installation

You can download Pulse Secure here : <https://cloud.ifremer.fr/index.php/s/WC9GArY8Eo51yZE>

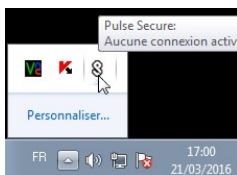
There is a version for :

- Windows 64 bits : JunosPulse.x64.msi
- Windows 32 bits : JunosPulse.x86.msi
- Mac OS X 64 bits : JunosPulse.dmg
- Linux 64 bits
 - CentOS 6.4 / Fedora 23 : pulse-8.1R11.1.i386.rpm
 - Fedora 24 or 25 : ps-pulse-linux-5.2r6.0-b977-centos-rhel-installer.rpm
 - ***rpm -ivh pulse-xxxx.rpm***
 - Ubuntu 14.04 : pulse-8.1R11.1.i386.deb
 - ***dpkg -i pulse-xxxx.deb***
 - *With ubuntu, you'll probably have to install these libraries :*
 - ***apt-get install libc6-i386***
 - ***apt-get install lib32z1***
 - Ubuntu 17.04 is not officially supported but it works with : pulse-8.2R5.i386.deb
 - ***sudo dpkg -i pulse-xxxx.deb***
 - ***sudo apt-get install libstdc++6:i386***

2 Using Pulse Secure

For Windows and MacOS

For the first time, you'll have to configure the connection :
Execute Pulse Secure :

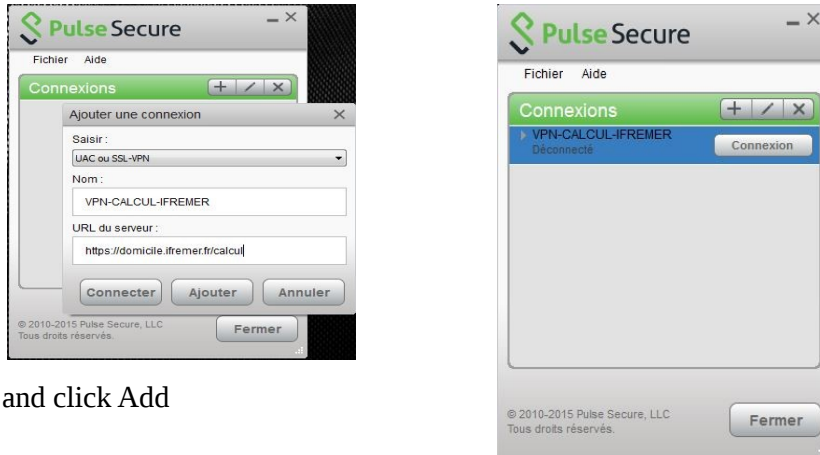


Windows : launch Pulse Secure



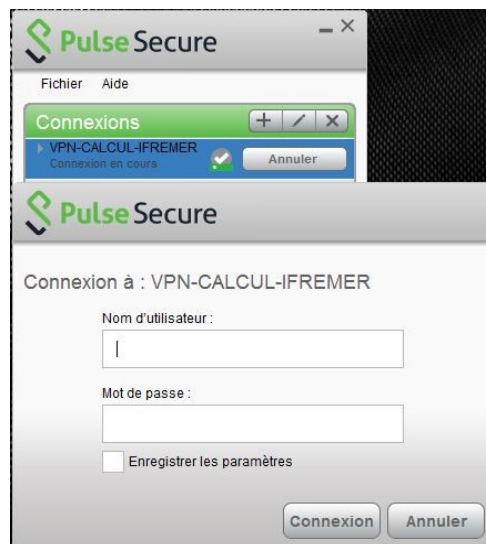
MacOs : launch Pulse Secure

and add a new connection with « + » like this :

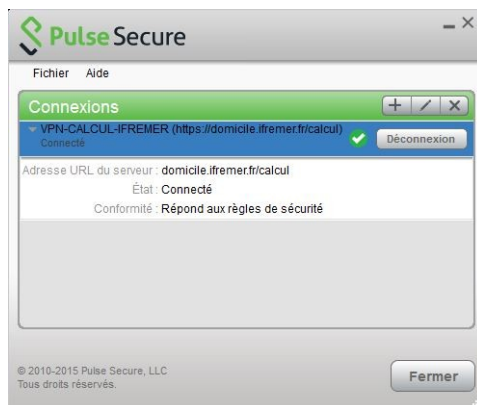


and click Add

Click Login and enter your extranet login and password



You are connected :



At the end, just click disconnect.

For Linux

To connect, just use the command line (Think about logging off in firefox before) :
/usr/local/pulse/PulseClient.sh -h domicile.ifremer.fr -u your_extranet_login -U https://domicile.ifremer.fr/calcul -r vpn-Ifremer

VPN Password:

Give your password and you should be connected.
(With Ubuntu, you have to use "sudo /usr/local/pulse/PulseClient.sh" just for the 1st connection)

To check the status, you can use this command in an another terminal window :

/usr/local/pulse/PulseClient.sh -S

Connection Status :

connection status : Connected
bytes sent : 584
bytes received : 0
Connection Mode : ESP
Encryption Type : AES128/SHA1
Comp Type : None
Assigned IP : 134.246.222.xxx

To disconnect, simply do "Control + C" on the 1st connection window or launch from another Terminal window:

/usr/local/pulse/PulseClient.sh -K

3 Access to datarmor

When you are connected with pulse secure, you can connect to datarmor via ssh :

ssh datarmor.ifremer.fr

or

ssh datarmor0-10g.ifremer.fr (same as **ssh 134.246.184.4**)

ssh datarmor1-10g.ifremer.fr (same as **ssh 134.246.184.5**)

ssh datarmor2-10g.ifremer.fr (same as **ssh 134.246.184.6**)

ssh datarmor2-10g.ifremer.fr (same as **ssh 134.246.184.7**)

For windows, you can use a ssh client like putty or mobaXterm