

HOW TO RUN MATLAB on the HPC – Ugent

In this specific example I am running an SVM decoding analysis using:

- CosmoMVPA toolbox (<https://www.cosmomvpa.org>)
- libsvm library (<https://github.com/cjlin1/libsvm>)

Introductory part & data upload (same as you can find in the “HOW TO RUN fMRIprep on the HPC – Ugent”)

1. If you have never used the HPC before, you need to request an account. You can find the procedure and detailed information here: <https://docs.hpc.ugent.be/macOS/account/>
*If you have access to a VO (== virtual organization), that will give you much more storage space, so ask for it!
2. Access your HPC space → <https://login.hpc.ugent.be/> and after the login you should arrive to this page


HPC UGent

Files

Jobs

Clusters

Interactive Apps



OnDemand provides an integrated, single access point for all of your HPC resources.

Status

STEVIN HPC-UGent infrastructure status on Thu, 14 Mar 2024 09:10:02

cluster	full nodes	free nodes	partial nodes	total nodes	running jobs	queued jobs
skitty	3	0	57	68	191	10028
joltik	3	0	7	10	27	58
doduo	40	0	55	128	173	60
accelgor	5	0	3	9	20	764
donphan	0	0	15	16	15	13
gallade	3	0	12	16	27	141

Documentation is available at: <https://www.ugent.be/hpc/en/support/documentation.htm>

For a full view of the current loads and queues see: <https://hpc.ugent.be/clusterstate/>

Updates on current system status and maintenance can be found on: <https://www.ugent.be/hpc/en/infrastructure/status>

To contact the support team, send an email to hpc@ugent.be.

If you click on “Files” you will see all the possible folders and space where you will put your data and scripts.

Files

Jobs

Clusters

Interactive Apps

Home Directory

\$VSC_DATA /data/gent/473/vsc47358

\$VSC_SCRATCH /scratch/gent/473/vsc47358

\$VSC_SCRATCH_ARCANINE /arcanine/scratch/gent/473/vsc47358

In my case, I did not join a VO yet but if you did you should have also folders with VO at the end: e.g. \$VSC_DATA_VO. For now I will not use those, but if you have them I suggest you to use them (bigger storage).

As a general info:

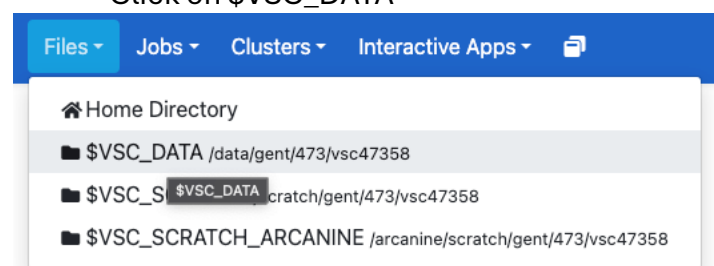
- the Home directory and the \$DATA are long-term storage slow filesystem → in the home directory you can store some scripts and configuration files, in the DATA folder (bigger space) you can store your data/results.
- \$VSC_SCRATCH is a fast temporary storage, e.g. for transient data like the temporary files created by fmripred during preprocessing. This is where we'll run our preprocessing analysis.

Here you can find a full description of all possible folders in the HPC:

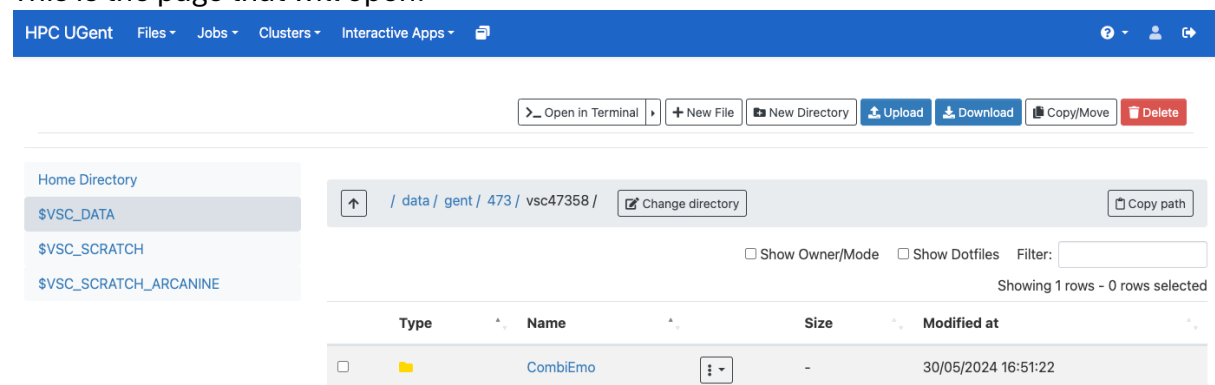
https://docs.hpc.ugent.be/macOS/running_jobs_with_input_output_data/?h=%24vsc+data

3. **Upload your data.** You need them to be in BIDS format in order to run fmripred. We will upload them on the folder \$VSC_DATA (or better \$VSC_DATA_VO if you have it).

Click on \$VSC_DATA

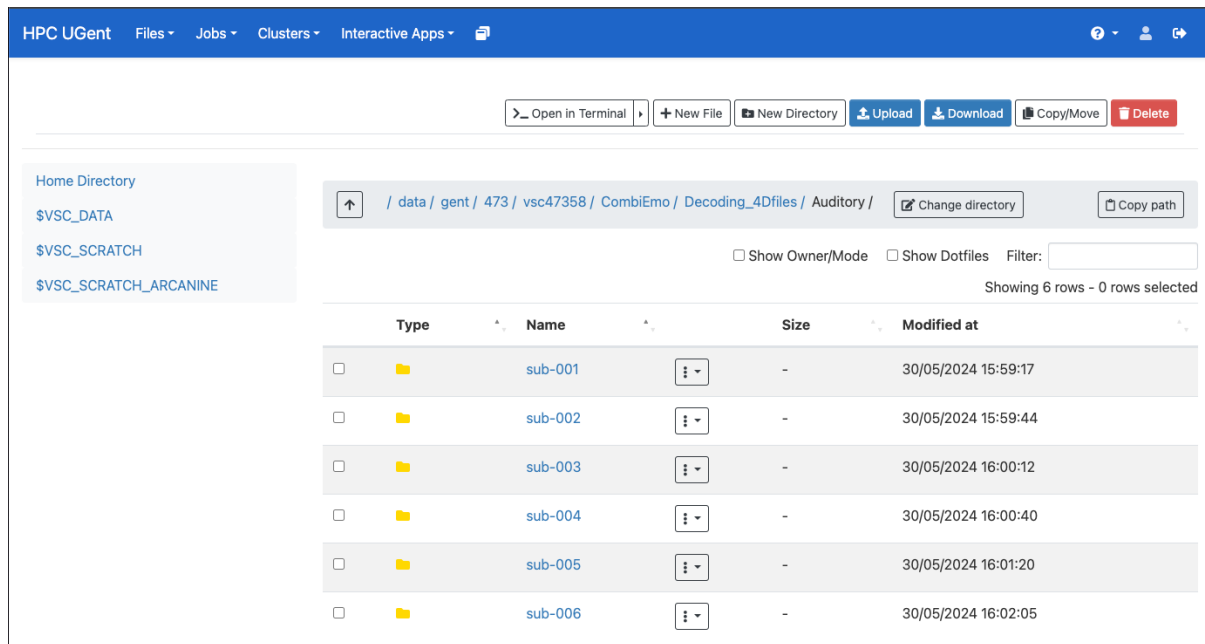


This is the page that will open:



I already have my data uploaded in the folder CombiEmo/Decoding_4Dfiles/

Here I have 3 folders: *Auditory*; *Visual*; *Bimodal*. Within each of these folders I have one folder per subject, and within each sub-folder I have a 4D file that I will use to run my searchlight decoding analysis: '4D_beta_2.nii'.



You have 2 options for uploading the data:

- a. Use the Upload button on top of the page



This will be slower and sometimes less smooth due to internet connection

- b. Use the terminal to upload the data. To use this option you need to set the SSH key access when setting your account (if you didn't do that, you can go back and add it now!)

- To use the terminal, open it in your local pc and type this line: **rsync -rv [path_to_data_on_your_pc_goes_here] [home_directory_goes_here]**
- In my specific case this looks like: **rsync -rv Desktop/Decoding_4Dfiles vsc47358@login.hpc.ugent.be:**
- You will be asked to enter your pass_phrase and then the copy procedure will start.
- Once it ends, you can open your Home Directory on HPC and you will find your data folder. In my case the 'Decoding_4Dfiles' folder is there.
- Now I just need to move it to the \$VSC_DATA: click on the folder and then click on the "Copy/Move" button on the top menu. A window on the left side will appear.

Now move to the folder where you want to copy your data: \$VSC_DATA. Once you are there, you can click on the red button "Move" in the left window and the data will be moved here.

4. I also created here an empty folder CombiEmo/Results where I will store the resulting .nii files from the decoding analyses (this path is specified in my .m script).

Prepare your environment (folders and scripts)

5. Now let's prepare the rest of the environment in the folder \$VSC_SCRATCH.

Here I created a folder using the top button 'New Directory' and I named it 'CombiEmo_script'. Inside this folder I uploaded these folders and script:

- a. **'CoSMoMVPA' folder**. This is a matlab toolbox that consist in a series of functions. To use my matlab script in the HPC I need all the functions that are used in the script to be also on the cluster. I will need to compile those together with my .m script (see below)
- b. **'masks' folder**. Where I uploaded the brain masks I will use for the searchlight (these masks are specified in my .m script)
- c. **'temporary_dir' folder** where some log files created during the analyses will be stored.
- d. **'Decoding_WithinModality_CombiEmo.m' →** my matlab script to run the decoding using CoSMoMVPA (you can find it in this Github folder).
- e. **'BASH_Matab.sh' →** I will launch this script (you can find it in this Github folder) from the HPC terminal to make running my matlab script.

Compile your MATLAB program

6. Many info and guidelines on why and how you compile your Matlab program to be able to use it in the HPC can be found here:

<https://docs.hpc.ugent.be/MATLAB>

In my case these are the steps I followed:

- a. Once in the project folder in the HPC (i.e. 'CombiEmo_script'), open the terminal and type `'module load MATLAB/2022b'` and enter → you need to load this specific version if you want to use the libsvm. For other kind of analyses there are other matlab versions also available (e.g. 2021b), but the libsvm is in the path of matlab 2022b!
- b. Then load also the libsvm module typing in the terminal `'module load LIBSVM-MATLAB/3.30-GCCcore-11.3.0-MATLAB-2022b-r5'`
- c. The next step is to compile your matlab program, and including in this compilation your script and the functions and library that you need to run the script. In my case I type this in the terminal: `mcc -mv Decoding_WithinModality_CombiEmo.m -a CoSMoMVPA -a $EBROOTLIBSVMINMATLAB`

At this point you should see something like this appearing in the terminal (maybe some other warnings as well):

```
[vsc47358@gligar08 CombiEmo_script]$ mcc -mv Decoding_WithinModality_CombiEmo.m -a CoSMoMVPA -a $EBROOTLIBSVMINMATLAB
Opening log file: /user/gent/473/vsc47358/java.log.42975
Warning: X does not support locale en_US.UTF-8
Compiler version: 8.5 (R2022b)
Analyzing file dependencies.
Parsing file "/kyukon/scratch/gent/473/vsc47358/CombiEmo_script/Decoding_WithinModality_CombiEmo.m"
(referenced from command line).
Generating file "/kyukon/scratch/gent/473/vsc47358/CombiEmo_script/readme.txt".
Generating file "run_Decoding_WithinModality_CombiEmo.sh".
```

In addition, a bunch of files will be created in the project folder (i.e. 'CombiEmo_script'), among those files there should be: 'run_Decoding_WithinModality_CombiEmo.sh' that will be used to run your script.

Prepare your **BASH_MATLAB.sh** script

7. This is the final step, you will launch this script on the terminal to run your .m script. You can find an example of mine in this Github repository.
In this script you will set some info about the number of nodes you will ask for, the walltime, you will set some directory_pats and finally you have the command to run your compiled script.
This is how it will look like:

```
#!/bin/bash
#PBS -N MATLAB-BetCat_5waydecoding_perm_AI
#PBS -l nodes=2:ppn=16
#PBS -l mem=25gb
#PBS -l walltime=24:00:00
#PBS -m abe
#
# Example (single-core) MATLAB job script
# see http://hpcugent.github.io/vsc_user_docs/
#
# make sure the MATLAB version matches with the one used to compile the MATLAB program!
module load MATLAB/2021b
# use temporary directory (not $HOME) for (mostly useless) MATLAB log files
# subdir in $TMPDIR (if defined, or /tmp otherwise)
export MATLAB_LOG_DIR=/scratch/gent/473/vsc47358/CombiEmo_script/temporary_dir
# configure MATLAB Compiler Runtime cache location & size (1GB)
# use a temporary directory in /dev/shm (i.e. in memory) for performance reasons
export MCR_CACHE_ROOT=/scratch/gent/473/vsc47358/CombiEmo_script/temporary_dir
export MCR_CACHE_SIZE=1024MB
# change to directory where job script was submitted from
cd /scratch/gent/473/vsc47358/CombiEmo_script
# run compiled example MATLAB program 'example', provide '5' as input argument to the program
# $EBROOTMATLAB points to MATLAB installation directory
#./run_setup_libsvm.sh $EBROOTMATLAB
./run_Decoding_WithinModality_CombiEmo.sh $EBROOTMATLAB
```

8. In the terminal type: **sh BASH_Matlab.sh**
Something like this should appear in the terminal:

```
[vsc47358@gligar08 CombiEmo_script]$ sh BASH_Matlab.sh
-----
Setting up environment variables
-----
LD_LIBRARY_PATH is ./apps/gent/RHEL8/zen2-1b/software/MATLAB/2022b-r5/runtime/glnxa64:/apps/gent/RHEL8/zen2-1b/software/MATLAB/2022b-r5/bin/glnxa64:/a
pps/gent/RHEL8/zen2-1b/software/MATLAB/2022b-r5/sys/os/glnxa64:/apps/gent/RHEL8/zen2-1b/software/MATLAB/2022b-r5/sys/openssl/lib/glnxa64
Opening log file: /scratch/gent/473/vsc47358/CombiEmo_script/temporary_dir/java.log.46612
Warning: X does not support locale en_US.UTF-8
```

And then, something similar of what you expect to see in the matlab command window. In my case the CosmoMCPA countdown:

```
+00:00:09 [#####] -00:00:00 mean size 99.6
+00:40:04 [#####-----] -01:11:11 [ ]
```

9. When the decoding analysis ends, the resulting .nii file is stored in the folder /data/gent/473/vsc47358/CombiEmo/results ready to be downloaded and inspected

