

PsychHP01 Cluster User Guide

VERSION 1

AYAN SENGUPTA

Contents

Introduction to the Cluster	2
Software Packages on the Cluster.....	2
Access the Cluster	10
Access through the Command Line.....	10
Access through the Graphical User Interface (GUI)	11
Data Structure on the Cluster.....	12
/MRIRaw	12
/MRIWork	13
/MRIArchive.....	14
Archive unused data on the cluster	14
Understanding the folder structure.....	14
How to compress the data and archive it.....	15
Check if the backup has worked.....	15
Delete the original folder.....	16
Retrieving the Backup	16
Movement of data to and from the cluster and local	17
Using Filezilla (SFTP).....	17
Using command line (SCP)	17
Running jobs on the job scheduler	19

Introduction to the Cluster

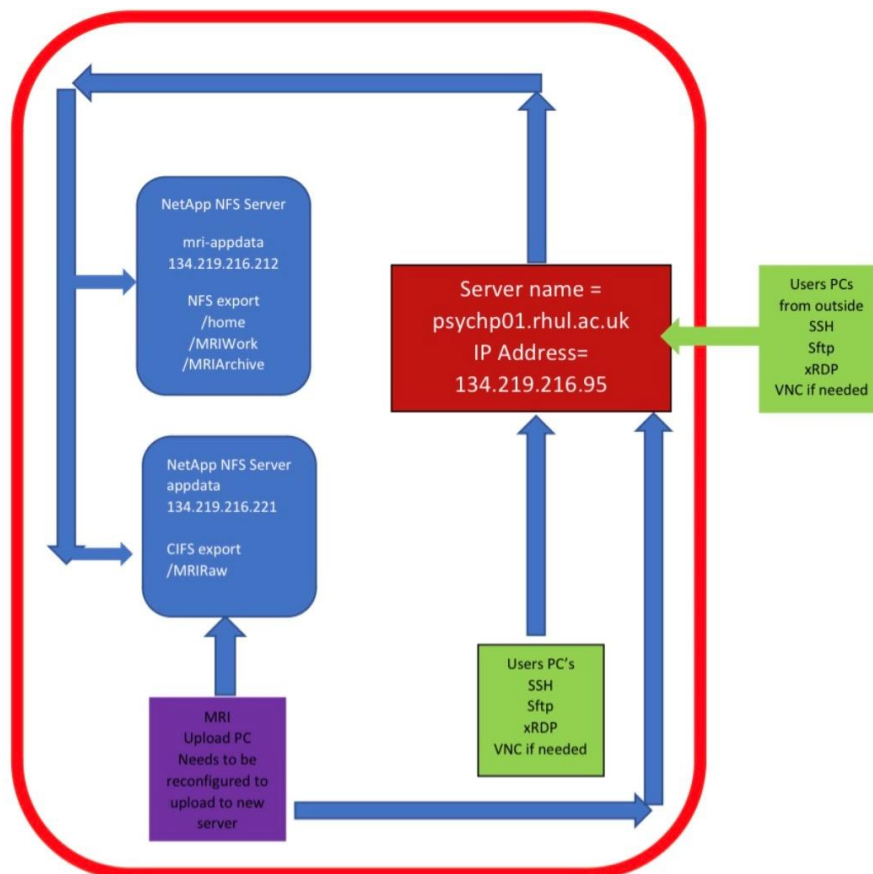
The PsychHP01 High Performance Cluster has been designed on the latest cutting-edge SSD based hardware platform of the RHUL Central IT. It is a fully backed-up virtualized system which is easily maintainable and also expandable for future research requirements. The cluster is also secure by the latest Firewall rules put in by the Central IT, however it is accessible to users over the college VPN for off-campus usage once the initial settings of the user has been done. The cluster has got one node with 128 cores of processing power and 256 GB of memory. It is accessible on campus at ***psychp01.rhul.ac.uk*** and also mounts the NFS file systems from the NetApps File Server. Please have a look at the Figure 1 for an overall idea of the architecture of the system. The access to the cluster is covered in the next chapter.

Software Packages on the Cluster.

PsychHP01 is a Linux-based platform and runs [Debian 11](#) Operating System on it. It has neuroimaging software suite NeuroDebian installed on it. All the software applications are installed locally on the cluster according to a decision taken by the Psychology IT Team. The software suite installed on the cluster is continuously evolving according to user needs and requirements. If a user needs a particular software for a study, the user is encouraged to get in touch with [Ayan Sengupta](#) at the earliest convenience. Ayan will work with the researcher to understand the requirements, run a feasibility analysis and install the required software as soon as possible. The timeline for the new software installation will be decided on a case-by-case basis involving the concerned researcher and utmost care would be taken not to

hamper the research workflow of all the users on the cluster. At the moment, following is a list of available software on the cluster and their corresponding path:

- **MATLAB R2023a** (*/usr/local/apps/MATLAB*)
- **GNU Octave 6.2.0** (*/usr/share/octave*)
- **Python 2.7** (*/usr/bin/python2*)
- **Python 3.11** (*/usr/bin/python3*)
- **SPM12** (*/usr/local/apps/SPM*)
- **FSI 6.0.7** (*/usr/local/apps/fsl_6.0.7*)
- **R 4.0.4** (*/usr/bin/R*)



Red outer line indicates VPN. Users from home are given firewall access so that they can connect to the server using SSH, SFTP, xRDP and VNC protocols.

Figure 1: Overview of the PsychHP01 cluster architecture

Below is a list of Matlab toolboxes that are licensed and available on the cluster to be used:

```
-----  
-----  
MATLABVersion: 9.14.0.2337262 (R2023a) Update 5  
MATLAB License Number: 935472  
Operating System: Linux 5.10.0-25-amd64 #1 SMP Debian  
5.10.191-1 (2023-08-16) x86_64  
JavaVersion: Java 1.8.0_202-b08 with Oracle Corporation Java  
HotSpot(TM) 64-Bit ServerVM mixed mode  
-----  
-----
```

MATLAB (R2023a)	Version 9.14
Simulink (R2023a)	Version 10.7
5G Toolbox (R2023a)	Version 2.6
AUTOSAR Blockset (R2023a)	Version 3.1
Aerospace Blockset (R2023a)	Version 6.0
Aerospace Toolbox (R2023a)	Version 4.4
Antenna Toolbox (R2023a)	Version 5.4
Audio Toolbox (R2023a)	Version 3.4
Automated Driving Toolbox (R2023a)	Version 3.7
Bioinformatics Toolbox (R2023a)	Version 4.17
Bluetooth Toolbox (R2023a)	Version 1.2
C2000 Microcontroller Blockset (R2023a)	Version 1.0
Communications Toolbox (R2023a)	Version 8.0
ComputerVision Toolbox (R2023a)	Version 10.4
Control System Toolbox (R2023a)	Version 10.13
Curve Fitting Toolbox (R2023a)	Version 3.9
DDS Blockset (R2023a)	Version 1.4
DSP HDL Toolbox (R2023a)	Version 1.2
DSP System Toolbox (R2023a)	Version 9.16

Database Toolbox (R2023a)	Version 11.0
Datafeed Toolbox (R2023a)	Version 6.4
Deep Learning HDL Toolbox (R2023a)	Version 1.5
Deep Learning Toolbox (R2023a)	Version 14.6
Econometrics Toolbox (R2023a)	Version 6.2
Embedded Coder (R2023a)	Version 7.10
Filter Design HDL Coder (R2023a)	Version 3.1.13
Financial Instruments Toolbox (R2023a)	Version 3.6
Financial Toolbox (R2023a)	Version 6.5
Fixed-Point Designer (R2023a)	Version 7.6
Fuzzy Logic Toolbox (R2023a)	Version 3.1
GPU Coder (R2023a)	Version 2.5
Global Optimization Toolbox (R2023a)	Version 4.8.1
HDL Coder (R2023a)	Version 4.1
HDLVerifier (R2023a)	Version 7.1
Image Acquisition Toolbox (R2023a)	Version 6.7.1
Image Processing Toolbox (R2023a)	Version 11.7
Industrial Communication Toolbox (R2023a)	Version 6.2
Instrument Control Toolbox (R2023a)	Version 4.8
LTE Toolbox (R2023a)	Version 3.9
Lidar Toolbox (R2023a)	Version 2.3
MATLAB Coder (R2023a)	Version 5.6
MATLAB Compiler (R2023a)	Version 8.6
MATLAB Compiler SDK (R2023a)	Version 7.2
MATLAB Report Generator (R2023a)	Version 5.14

MATLAB Test (R2023a)	Version 1.0
Mapping Toolbox (R2023a)	Version 5.5
Medical Imaging Toolbox (R2023a)	Version 1.1
Mixed-Signal Blockset (R2023a)	Version 2.4
Model Predictive Control Toolbox (R2023a)	Version 8.1
Motor Control Blockset (R2023a)	Version 2.0
Navigation Toolbox (R2023a)	Version 2.4
Optimization Toolbox (R2023a)	Version 9.5
Parallel Computing Toolbox (R2023a)	Version 7.8
Partial Differential Equation Toolbox (R2023a)	Version 3.10
Phased Array System Toolbox (R2023a)	Version 5.0
Powertrain Blockset (R2023a)	Version 2.0
Predictive Maintenance Toolbox (R2023a)	Version 2.7
RF Blockset (R2023a)	Version 8.5
RF PCB Toolbox (R2023a)	Version 1.3
RF Toolbox (R2023a)	Version 4.5
ROS Toolbox (R2023a)	Version 2.0
Radar Toolbox (R2023a)	Version 1.4
Reinforcement Learning Toolbox (R2023a)	Version 2.4
Requirements Toolbox (R2023a)	Version 2.2
Risk Management Toolbox (R2023a)	Version 2.2
Robotics System Toolbox (R2023a)	Version 4.2
Robust Control Toolbox (R2023a)	Version 6.11.3
Satellite Communications Toolbox (R2023a)	Version 1.4
Sensor Fusion and Tracking Toolbox (R2023a)	Version 2.5

SerDes Toolbox (R2023a)	Version 3.0
Signal Integrity Toolbox (R2023a)	Version 1.3
Signal Processing Toolbox (R2023a)	Version 9.2
SimBiology (R2023a)	Version 6.4.1
SimEvents (R2023a)	Version 5.14
Simscape (R2023a)	Version 5.5
Simscape Battery (R2023a)	Version 1.1
Simscape Driveline (R2023a)	Version 3.7
Simscape Electrical (R2023a)	Version 7.9
Simscape Fluids (R2023a)	Version 3.6
Simscape Multibody (R2023a)	Version 7.7
Simulink 3D Animation (R2023a)	Version 9.6
Simulink Check (R2023a)	Version 6.2
Simulink Code Inspector (R2023a)	Version 4.3
Simulink Coder (R2023a)	Version 9.9
Simulink Compiler (R2023a)	Version 1.6
Simulink Control Design (R2023a)	Version 7.0
Simulink Coverage (R2023a)	Version 5.6
Simulink Design Optimization (R2023a)	Version 3.13
Simulink DesignVerifier (R2023a)	Version 4.9
Simulink PLC Coder (R2023a)	Version 3.8
Simulink Real-Time (R2023a)	Version 8.2
Simulink Report Generator (R2023a)	Version 5.14
Simulink Test (R2023a)	Version 3.8
SoC Blockset (R2023a)	Version 1.8

Stateflow (R2023a)	Version 10.8
Statistics and Machine Learning Toolbox (R2023a)	Version 12.5
Symbolic Math Toolbox (R2023a)	Version 9.3
System Composer (R2023a)	Version 2.4
System Identification Toolbox (R2023a)	Version 10.1
Text Analytics Toolbox (R2023a)	Version 1.10
UAV Toolbox (R2023a)	Version 1.5
Vehicle Dynamics Blockset (R2023a)	Version 2.0
Vehicle Network Toolbox (R2023a)	Version 5.4
Vision HDL Toolbox (R2023a)	Version 2.7
WLAN Toolbox (R2023a)	Version 3.6
Wavelet Toolbox (R2023a)	Version 6.3
Wireless HDL Toolbox (R2023a)	Version 2.6
Wireless Testbench (R2023a)	Version 1.2

Have a look [here](#) for an updated list of Matlab toolboxes on the cluster. Additionally Python 3 packages that are currently installed on the cluster can be found [here](#) and Python 2 packages are found [here](#).

For creating custom and reproducible python virtual environments on the cluster the **anaconda** package has been installed. This is an open-source package and environment management system and a truly versatile tool. Some very useful resources are available like a [Cheat Sheet](#) , a detailed [documentation](#) on the entire package and a short [10min video](#) showing the basics of conda. Please to get in touch

with [Ayan Sengupta](#) if you are looking to use conda on the cluster and would require more assistance.

For advanced linux users who want to use containerized applications both [Docker](#) (version 24.0.6) and [Singularity](#) (2.6.1-dist) is installed on the cluster.

Access the Cluster

Access to the cluster can be made available to all staff members with a @rhul.ac.uk email address. To request for an account on the cluster, please send an email to itservicedesk@rhul.ac.uk. It is expected to take 2-3 days for creating your account and corresponding access to the system. Someone from the Psychology IT team will get in touch with you shortly and when you get confirmation, you will be able to connect to the cluster. As the cluster is a Linux based computational platform, It is strongly advised that new cluster users explore the many tutorials and documentation resources available on the web, for example: HPC [Cluster Tutorials](#) or our [RHULPsychology github page](#).

Access through the Command Line

After you have registered up for the cluster access, you will be provided with a local USERID and PASSWORD, different from your college login credentials, which you can use to log into psychp01 using SSH (secure shell). The location of the server for logging on to is ***psychp01.rhul.ac.uk***.

When you are connected to the campus network, you can directly log onto PsychHP01 from your computer whereas, if you are off campus, you will need VPN access to the campus network and then connect to PsychHP01. If you don't have VPN access, please submit a request to the IT services desk (itservicedesk@rhul.ac.uk).

To SSH to psychp01 from Linux or Apple OSX, you should open a terminal and type

```
ssh username@psychp01.rhul.ac.uk
```

Replace ``username" with your username. Your username and password will be provided to you by the IT when you request access to PsychHP01.

From Windows, you will need to have installed an SSH client. The cmdr console emulator works nicely with Windows 10/11 and is available open source from this [link](#). Please download the full version to get built-in installation of git-for-windows along with it. After you have downloaded the zip file, extract it in your preferred location and inside the extracted folder you would find a `cmdr.exe` executable. Please double click on the executable and your windows terminal will be ready for you, you should be able to run the SSH command shown before to log on to the cluster. If you have any queries please get in touch with [Ayan Sengupta](#).

For Interactive sessions, for example Matlab you should with add `-Y` to your ssh command:

```
ssh -Y username@psychp01.rhul.ac.uk
```

where the command `-Y` enables trusted X11 forwarding.

Access through the Graphical User Interface (GUI)

Alternatively, for allowing GUI access, the cluster comes installed with Remote Desktop Protocol (RDP). From a Windows and Apple computers, use the Microsoft Remote Desktop App. From Linux computers use Remmina. You should use the server location as `psychp01.rhul.ac.uk` and accept the certificates and use the local userid and password provided by Psychology IT for logging in through RDP. The

advantage of RDP is that your desktop will be preserved between logins, across network disconnects, etc, until the cluster is reset.

Data Structure on the Cluster

The data is organized on the cluster primarily under 3 folders namely, /MRIRaw, /MRIWork and /MRIArchive. In the following sections we are going to discuss the data structure used in these locations and corresponding allocated storage for proper data management on the cluster.

/MRIRaw

The /MRIRaw folder is designed to hold raw data coming from the MRI scanner. When you are scanning for any pilot experiments, data collection etc. the raw data in DICOM format will be made available to you in /MRIRaw. Under the /MRIRaw folder a raw folder space number would be allocated for each PI and underneath that individual project and scan sequence data will have subfolders created for that. Below is an example.

```
└─ MRIRaw
  └─ MRIRaw00
    └─ QA
      └─ 201901031153_19800101XXXX
          └─ Series_001_32 Channel Localizer
          └─ Series_002_QA_EPI_Standard_Grappa
          └─ Series_003_QA_EPI_Standard_NoGrappa
      └─ 201901081222_19800101XXXX
          └─ Series_001_32 Channel Localizer
          └─ Series_002_QA_EPI_Standard_Grappa
          └─ Series_003_QA_EPI_Standard_NoGrappa
```

Here, `MRIRaw00` is the raw data folder workspace allocated for the PI Ari Lingeswaran. The project the PI is collecting data for is `QA`. The participant ID / CUBIC ID that he has collected data for is `201901031153_19800101XXXX`, which is a phantom in this case. The sequences used in the data collection is `Series_001_32 Channel Localizer`, `Series_002_QA_EPI_Standard_Grappa` etc. The raw DICOM files are saved under each sequence folders.

`/MRIWork`

The `/MRIWork` folder is designed to hold the data analysis of different projects under PIs. When a PI asks for an account on the cluster a corresponding `/MRIWork` workspace is created for them. At the moment, each PI workspace is allocated 1.5TB of data storage. Below is an example of the folder structure in `/MRIWork`

```
└─ MRIWork
  └─ MRIWork00
    └─ AL
      ├── ari_lingeswaran
      │   └─ QA
      └─ ayan_sengupta
          └─ data_analytics
```

Here, `MRIWork00` is the data analysis workspace allocated for the PI Ari Lingeswaran. The PI, Ari Lingeswaran, has an analysis workspace for himself `/MRIWork/MRIWork00/AL/ari_lingeswaran` and also his lab member Ayan Sengupta is allocated the analysis workspace `/MRIWork/MRIWork00/AL/ayan_sengupta`. Individually Ari is analyzing data for the project `QA` and his lab member Ayan Sengupta is analyzing data for the project

data_analytics. The data in /MRIWork is regularly backed up and can be revived from Central IT upon request. You should submit a request to itservicedesk@rhul.ac.uk to get assistance.

/MRIArchive

The /MRIArchive folder is designed to archive the old data analysis of different projects under PIs. Any project which has been finished but the analysed data needs to be saved for future use and GDPR requirement of different grants requires to be archived in the /MRIArchive. The folder structure under /MRIArchive is similar to the /MRIWork folder. Below is an example of the folder structure in /MRIArchive

```
└─ MRIArchive
  └─ MRIWork00
    └─ AL
      ├── ari_lingeswaran
      │   └─ QA
      ├── ayan_sengupta
      │   └─ data_analytics
```

Archive unused data on the cluster

Following are the steps for archiving unused data on the cluster. If unsure, please get in touch with [Ayan Sengupta](#)

Understanding the folder structure

The archive space is located at /MRIArchive. It' folder structure has been organized similarly to the /MRIWork space. For example:

The data in folder `/MRIWork/MRIWork10/pv/<user_folder>/Old/` should be compressed and archived in `/MRIArchive/MRIWork10/pv/<user_folder>/Old.tar.gz` location as a compressed tarball gzip file. Individuals will have writing permissions to their respective archive folders.

How to compress the data and archive it

Use the `tar cvf` and `gzip -9` commands to achieve highest compression.

For example:

```
tar cvf /MRIWork/MRIWork10/pv/<user_folder>/Old/ | gzip -9 ->
/MRIArchive/MRIWork10/pv/<user_folder>/Old.tar.gz
```

Check if the backup has worked

Use the `Check_Backup_Content.sh` script in the location `/usr/local/apps/` to check whether the backed-up content is same as the original. It is done with the help of file identifiers called `md5sums`. The script takes 2 inputs (input 1 -> path to the original folder, input 2 -> path to the Tarball archive) and produces a report. For example:

```
bash
/usr/local/apps/Archive_Old_Workspace/Check_Backup_Content.sh
/MRIWork/MRIWork10/pv/<user_folder>/Old/
/MRIArchive/MRIWork10/pv/<user_folder>/Old.tar.gz
```


Check the Backup Report files thus created in /tmp/Backup_Report-XXXXXXXXXX/ to check if you have successfully retrieved your backed up data.

Delete the original folder

Please do not forget to remove your original folder. It would not work if you would delete it from the GUI, it will just go and sit in the trash folder. Use `rm -r` command for the removal of the directory. Be careful, this above process is non-reversible. Please double check if you have got the backup file properly. One simple way to do this is to check the size of the backup file with the `du` command. For example:

```
du -sh /MRIArchive/MRIWork10/pv/<user_folder>/Old.tar.gz
```

Once you are confident remove the folder like:

```
rm -r /MRIWork/MRIWork10/pv/<user_folder>/Old/
```

Retrieving the Backup

Please change the directory to where you want to retrieve your backup. Use the `tar xzf` command to extract it. For example:

```
cd /MRIWork/MRI-Scratch
```

```
tar xzf /MRIArchive/MRIWork10/pv/<user_folder>/Old.tar.gz
```

Movement of data to and from the cluster and local

For analysis and presentation purposes it might be necessary to transfer the data to and from the cluster and the local computer. The cluster is equipped to do this secure copy (SCP) and secure file transfer protocol (SFTP). For using the SFTP protocol we encourage you to use the Filezilla program. SCP protocol is to be used for command line operations. In the following sections we will describe the data transfer procedures.

Using Filezilla (SFTP)

Please download the application Filezilla from [here](#). This program enables you to navigate through the folders on the local computer and cluster. Please enter `psychp01.rhul.ac.uk` in the Host field and use the cluster local `Username` and `Password` and use port `22` for using SFTP. Please leave the other settings to default as you are installing Filezilla and logging into the cluster through it. You would be able to drag and drop files and folders bi-directionally between the local computer and cluster through Filezilla over the Secure File Transfer Protocol. In order to assist you in using Filezilla I have included a very handy [tutorial video](#) of Filezilla. Please have a look, if you have doubts in using Filezilla, reach out to [Ayan](#).

Using command line (SCP)

The SCP command help data transfer from the source to the destination location from the command line. The SCP command can be broadly represented as the following:

```
scp <source_location> <destination_location>
```

For transferring folders including subfolders and files please use

```
scp -r <source_location> <destination_location>
```

where 'r' flag stands for recursive.

For example, if I want to transfer a folder called `trial` from my local `Downloads` folder to the `/MRIWork/MRIWork00/AL/ayan_sengupta` folder on the cluster, the command will look like the following.

```
scp -r c:\users\ayan\Downloads\trial  
ayan_userid@psychp01.rhul.ac.uk:/MRIWork/MRIWork00/AL/aya  
n_sengupta/
```

Once this command is issued from the terminal, the system will prompt for my password on the cluster, on successful entry of the password, the data transfer will start.

Running jobs on the job scheduler

The cluster is equipped with a versatile job scheduling system called SLURM. The scheduler can be used to run jobs in parallel on the cluster making the jobs faster by several folds. The jobs scheduler runs a queue of jobs from multiple users. It allocates the required number of CPUS and the required amount of memory to individual jobs as soon as the resources become free. The jobs can be Matlab scripts, Python Scripts, R and Bash scripts etc. To be able to use the job scheduler a bash submit needs to be used where all the required resources are specified with the sbatch command. Below is an example of such a submit script to be used on the cluster to run a simple python script.

```
#!/bin/bash
# Script to submit jobs to the SLURM job scheduler on the cluster
# In this script we will submit a sample Python script to the job scheduler
# The sample Python script can be found at
# (https://github.com/RHULPsychology/PsychHP01_SLURM_Template/blob/main/sample_python_script.py)
# To be able to run a script properly through job scheduler SLURM and
# manage the input and output of the scripts
# SLURM requires 2 log files, one output log file saves/prints the Output
# of the script and the other error logfile saves/prints
# the errors that the script might generate. hence we need to create a log
# folder for the scripts
# The following 3 lines of code Create log folder for output and error
# files. It creates the SLURM_log folder in
# the users's home folder, the specific location is
# /home/<userid>/SLURM_log, where <userid> is the local userid
# If you need to have the output and error files to be saved in any other
# location, make sure to change the
# OUTPUT_LOG_DIR=~ /SLURM_log line of the code according to your
# requirement.

OUTPUT_LOG_DIR=~ /SLURM_log
mkdir -p $OUTPUT_LOG_DIR
echo "created log folders successfully".
```

```

# The following line of code submits the Matlab script through the sbatch
command. sbatch command configures the SLURM
# job scheduler to allocate the required memory and processing CPUs, time
to run on the queue etc.
# In this example sbatch allocates 1 processing node, 1 CPU per task, the
job is named as Sample_Python.job,
# job_scheduler output file is Sample_Python_job.out and error file is
Sample_Python_job.err under the OUTPUT_LOG_DIR created earlier.
# The maximum time this job can run on the scheduler will be 2 days and the
memory allocated is 1000MB
# The last line 'python sample_python_script.py Captain_Spark' actually
runs the sample_python_script.py file with the input argument Captain_Spark

```

```

sbatch --nodes=1 \
      --cpus-per-task=1 \
      --job-name=Sample_Python.job \
      --output=${OUTPUT_LOG_DIR}/Sample_Python_job.out \
      --error=${OUTPUT_LOG_DIR}/Sample_Python_job.err \
      --time=2-00:00 \
      --mem=1000 \
      python sample_python_script.py Captain_Spark

```

The sample python script that is run through the job scheduler is shown in the following paragraph.

```

#!/usr/bin/python
### This sample python script takes a user input from the command line and
prints a "Hello World" statement.
# import os and sys python packages are required for this basic
functionality
import os
import sys
# This line accepts the user input and assigns it to the variable subject
subject = sys.argv[1]

# The actual print statement
print ("Hello World Subject"+str(subject))

```

All the files described above are for demonstration purposes and can be used as a template for running individual analysis scripts on the cluster. These can be downloaded from the Psychology github page ([link](#)). Also some very useful resources and lessons for using SLURM can be found [here](#). If you plan to run any analysis scripts written in Matlab, FSL, Python or R please feel free to get in touch with [Ayan](#) at your earliest convenience.