Useful links

1. https://userinfo.surfsara.nl/systems/lisa/user-guide/creating-and-running-jobs

Downloads

1. UvA VPN
    a. Accessing the UvA network
2. WinSCP:
    a. This enables the transfer of files between your local computer and the remote cluster using a GUI (Jorge: this is the cp command that I was referring to, you can use either this one or the sftp procotol).
    b. Host name: `csn-h0.science.uva.nl`
    c. Username & password: UvANetID & normal password
3. **Note:** Alternative to WinSCP: use `sftp` to access the cluster in a second command window:
   → easier way to copy and move files
    a. `put file.pdf [local → remote]`
    b. `get file.pdf [remote → local]`

Entering the cluster

1. Open `Windows Powershell`
2. `ssh {uvanetid}@csn-h0.science.uva.nl`
3. Same password
4. h0 is the entrance node

Navigating the cluster

1. Useful unix commands:

`man ln`	List of commands
`cd`	Change directory
`cd ..`	Go up one directory
`cd ~`	Return to home directory
`ls`	List of files in current working directory

`ls-lh`	Long list of files with readable file size
`ls-l`	List with long format - show permissions
`mkdir`	Make a new directory
`cp`	Copy file (`file_name new_directory`)
`mv`	Move (`file_name new_directory`)

2. `cd /data`
   `mkdir {name}`
   `cd {name}`
   `mkdir {data} / {scripts} / {results}`

3. Use WinSCP/ `sftp` to copy the data over

Sending a job to the cluster

1. Clone the repository into your scripts folder on the cluster
2. Be sure to use the command "`git pull`" to retrieve any changes committed to the GitHub
3. `squeue -` see list of current jobs that are in the queue (not running yet on SLURM)
   a. Alternative `top -u {user}` (these jobs are currently running on the cluster)
4. `sbatch {slurm file} - start job`
5. `scancel {jobnumber} - cancel if needed`
6. Using the Vim editor:

`Esc`	Switch to command mode
`i`	Switch to edit mode
`:`	Start writing a command
`:wq`	Save and exit file

7. Example `slurm`

```
#!/bin/sh
#SBATCH -J Ferret_stuffs
#SBATCH --output=/home/cbosman1/slurm/log/pupilMaster.out
#SBATCH --error=/home/cbosman1/slurm/log/pupilMaster.err
#SBATCH --mail-user conrado.bosman@me.com
#SBATCH --mail-type=ALL
#SBATCH --mem=64G

module load matlab
matlab -nodisplay -r "addpath /home/cbosman1/matlab/scripts/pupilProject/; cb_master_get_pupil_LFP;  quit"
```

Template:

```
#!/bin/sh
#SBATCH -J {identifier}
#SBATCH --output=/home/{name}/slurm/log/pupilMaster.out {where error file should be saved}
#SBATCH --error=/home/{name}/slurm/log/pupilMaster.err {where error file should be saved}
#SBATCH --mail-user {email address}
#SBATCH --mail-type=ALL
#SBATCH --mem=64G

module load matlab {version 2017a}

matlab -nodisplay -r "addpath /home/{path}/; {scriptname};  quit"
```

Working with different input parameters:

1. `sbatch` is not interactive
2. Call multiple main files in the slurm file

```
matlab -nodisplay -r "addpath ~/scripts/; main(1); main(2); main(3); quit"
```

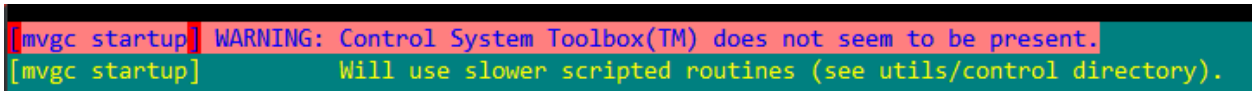
3. Alternative:

```
matlab -nodisplay -r "addpath ~/scripts/; main;  quit"
```

- The main.m file can call a “super_main.m” file: “super_main.m” will have a list of random variables that are output and stored somewhere as a variable.
- The main.m file then uses these stored variable as input

Setting up MVGC

1. Reference paper: The MVGC multivariate Granger causality toolbox: a new approach to Granger-causal inference

2. Add MVGC toolbox to personal folder in the cluster. Add to path and use start up in matlab script
3. This message may come up in your .err file:
    a. 

    b. "The Matlab Control System Toolbox function `dlyap` solves equations of the form (A.5); if the Control System Toolbox is not available, then the MVGC toolbox function `utils/dlyap_aitr` implements an efficient iterative solver"
    c. This is however a lot slower:
    d. One can add the Matlab Control System Toolbox to your folder on the cluster. Make sure you add it to path in the main script (see "`gc_parfor_cluster`" for an example).

In case of trouble with using 'startup`

```
origDir = pwd;
cd ~/matlab/MVGC
startup
cd(origDir)
```

Debugging in the cluster

It is possible to set up a testing `bash` environment to run the script. This can be set up to create an interactive matlab environment with a display.

Run the following commands to set up the testing environment:

```
srun --pty bash
module load matlab
matlab -r "addpath {path to script}; {scriptname};  quit"
```

Note the final command can be copied from the .slurm file. Be sure to remove the "-nodisplay" flag to be able to see your outputs