Using ICS Software on Blanca

This tutorial assumes the following:

1) you are an ICS user (a user with access to "blanca-ics" on the Blanca computing cluster);
2) you have started a vncserver session on blogin01.rc.colorado.edu; and
3) (from your laptop or desktop computer) you have logged into a remote desktop on blogin01.rc.colorado.edu using the vncserver session.

Slides outlining these and other aspects of using Blanca can be found at:
https://github.com/ResearchComputing/CHANGE_2019

Overview

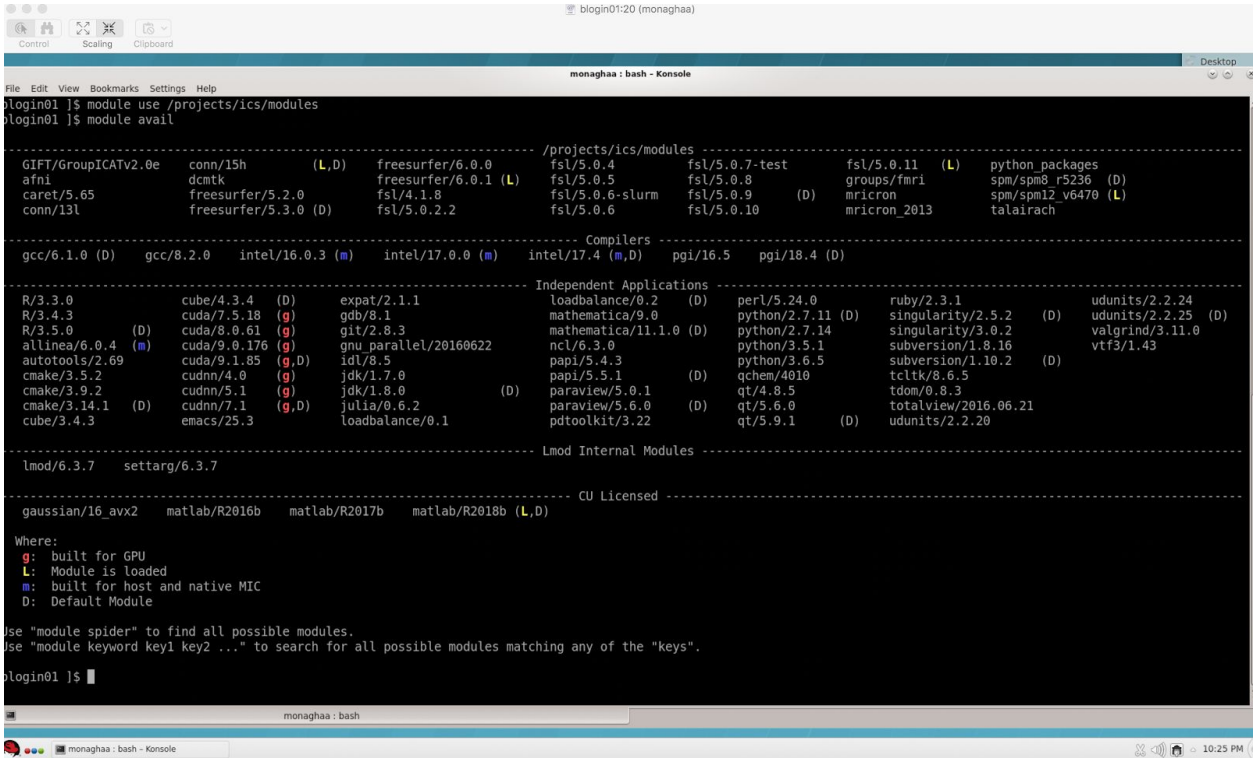
ICS has installed and manages many of their own software packages in /projects/ics/software. In order to enable the proper environment for each software package, ICS has created modules for each package that can be loaded just like any other module-based software on Blanca.

To enable the modules, one must first add them to the module stack as follows:

```
$ module use /projects/ics/modules
```

Now type the following to see the modules:

```
$ module avail
```



…you should see the usual modules that are available on Blanca, as well as an additional group of modules with the heading /projects/ics/modules that has modules for fsl, spm, freesurfer and other software packages.

Running ICS software on a blanca-ics compute node

If you'll be using ICS software for any sort of analysis, you should be doing the work on one of the blanca-ics compute nodes by running a job using the Slurm job manager. Otherwise, running tasks with software on the ICS head node, blogin01.rc.colorado.edu, will negatively impact your colleagues' ability to use Blanca.

A job can be submitted from blogin01.rc.colorado.edu. Jobs can be two types: 1) unattended tasks which are called batch jobs (submitted with '$ sbatch myjobscript.sh', or 2) tasks you interact with, called interactive jobs (submitted with '$ sinteractive --partition=blanca-ics --ntasks=<N>' where <N> is the number of cores you need).

In this tutorial, we will be interacting with the ICS software packages SPM12 and FSL via a graphical user interface (GUI). Therefore, we will start an interactive job before using the software. Let's start a 1-core job on a blanca-ics compute node:

$ sinteractive --partition=blanca-ics --ntasks=1

...sometimes additional memory (RAM) is needed to process large datasets. If a 1-core job crashes with an 'out-of-memory' error, try again but request more cores:

$ sinteractive --partition=blanca-ics --ntasks=2

...if you do not specify a blanca-ics node, you will randomly be assigned to one. The blanca-ics nodes include bnode0101, bnode0102, bnode0103, bnode0104, bnode0105, bnode0301, bnode0310, bnode0405, bnode0406, bnode0407, bnode0408, and bnode0409. The bnode03* and bnode04* nodes are newer and may provide better response times for GUI-based applications (compared to the bnode01* nodes). To specify one of these notes, you can add the --nodelist=<nodename> flag to the sinteractive commands above. Note that these are popular nodes among ICS users and may not always be available. An easy way to check whether there are some spare cores on any given node is as follows:

$ sinfo -a | grep ics

...which will yield some output like this:

```
blanca-ics      up 7-00:00:00     4    mix bnode[0101-0102,0104-0105]
blanca-ics      up 7-00:00:00     8  alloc bnode[0103,0301,0310,0405-0409]
```

...The "mix" nodes have at least one core available and can be used immediately. The "alloc" nodes have no cores presently available and therefore you would have to wait to use them.

Once you have started your interactive job on a blanca-ics compute node, you can use SPM12 and FSL per the steps provided below.

Note: For this tutorial, because we are simply learning how to invoke the ICS software but we won't be doing any computations, we will use the Blanca ICS head node, blogin01. Do not do this for your regular workflow or you will slow down blogin01 and negatively impact your other colleagues' ability to use Blanca.

Using SPM12

Step 1: From a terminal in your VNC desktop (and normally within an sinteractive job on a blanca-ics compute node), enable the ICS software modules and load matlab, spm12, and (optionally) the conn toolbox.

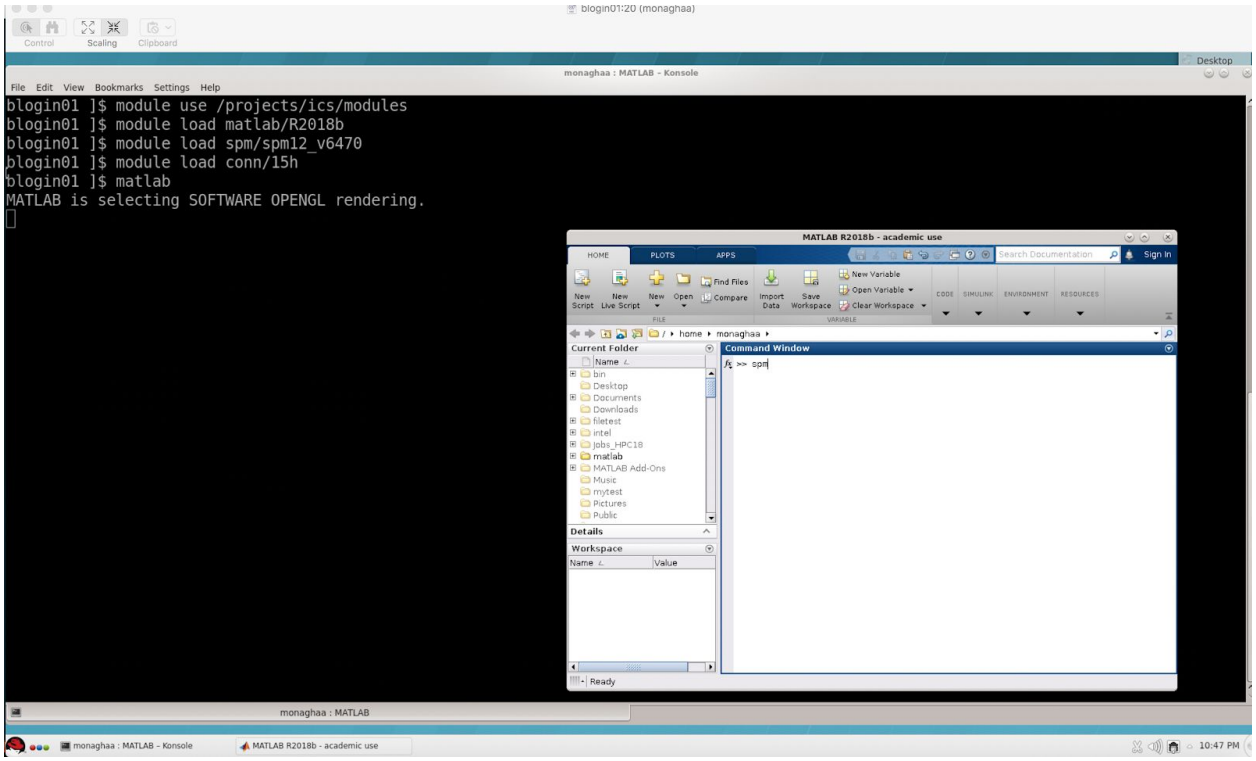
```
$ module use /projects/ics/modules
$ module load matlab/R2018b
$ module load spm/spm12_v6470
$ module load conn/15h
```

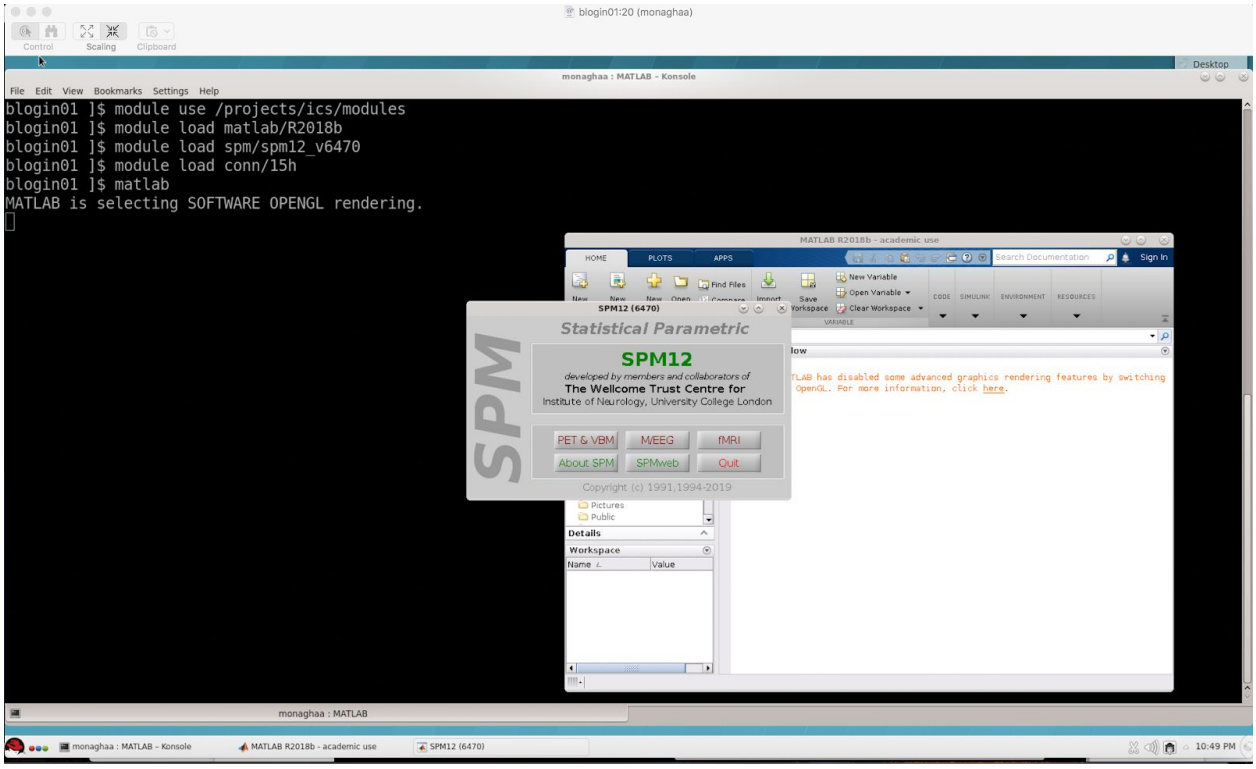
Step 2: Start matlab (this may take a few moments for the matlab GUI to initialize).

```
$ matlab
```

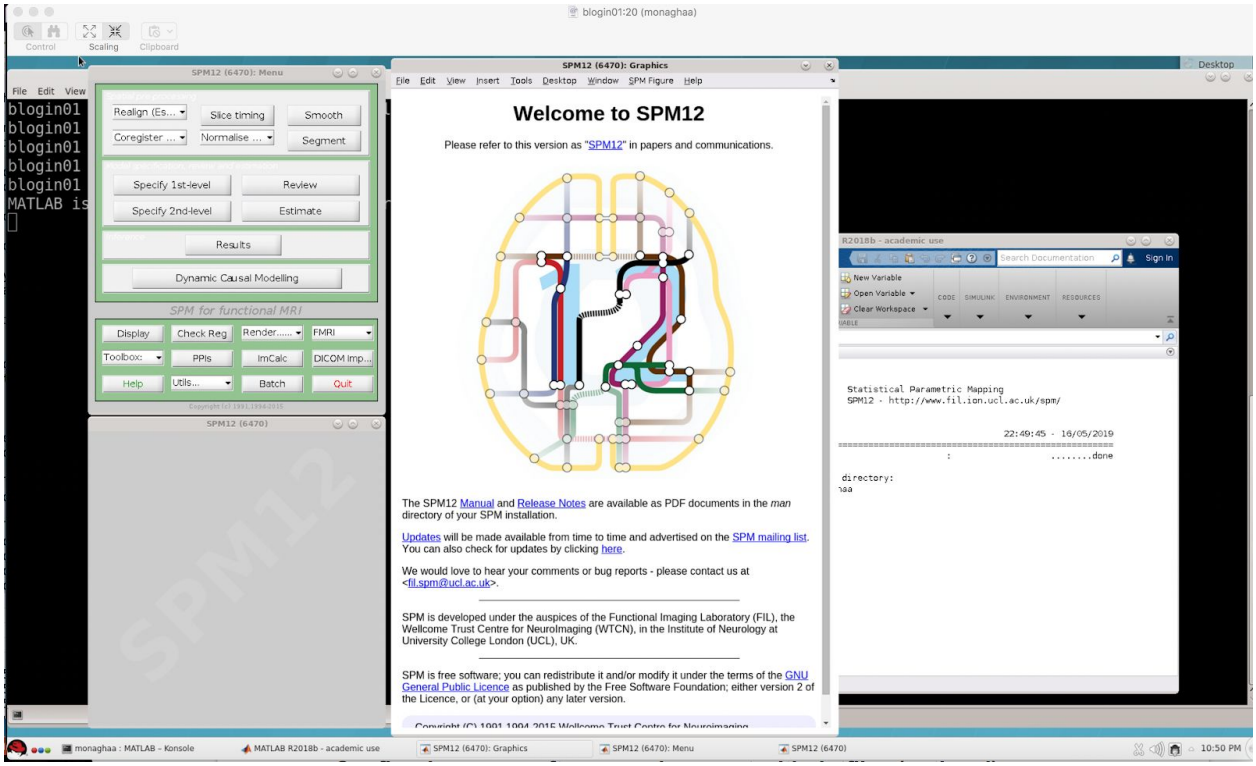
Step 3: From within matlab, invoke spm12

```
>> spm
```



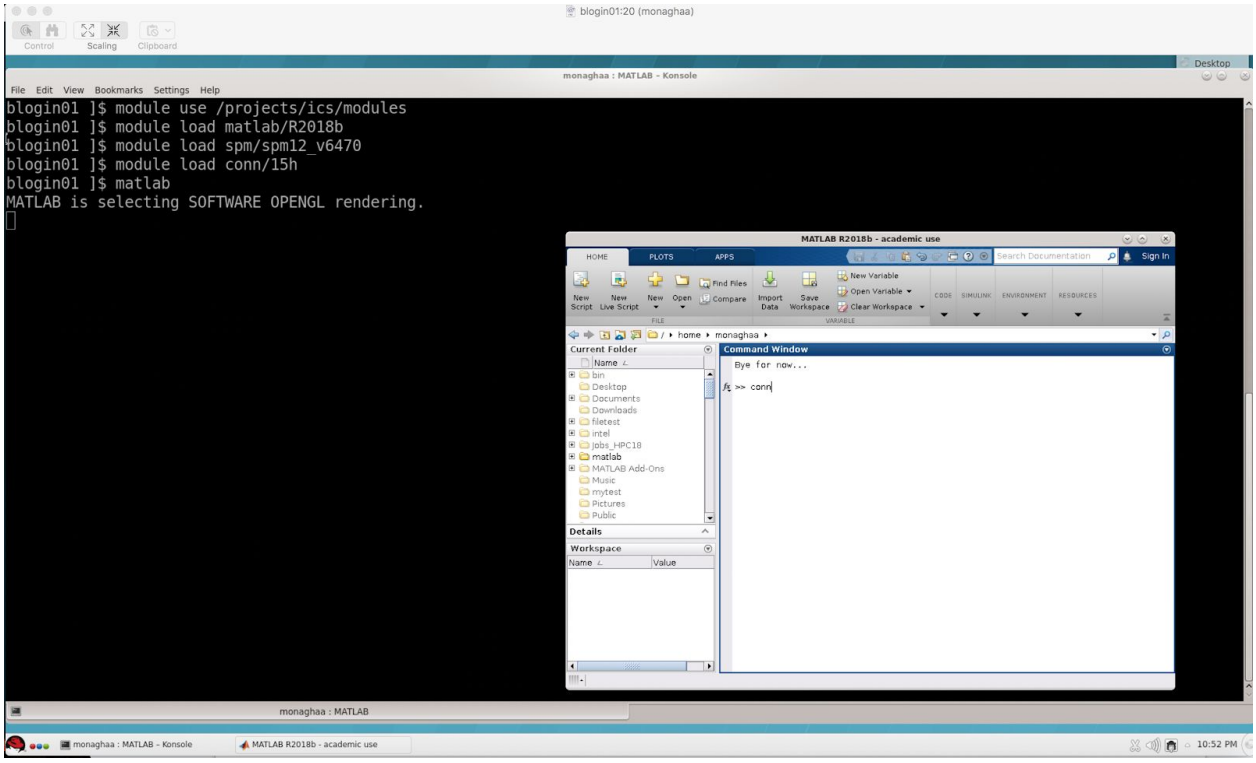


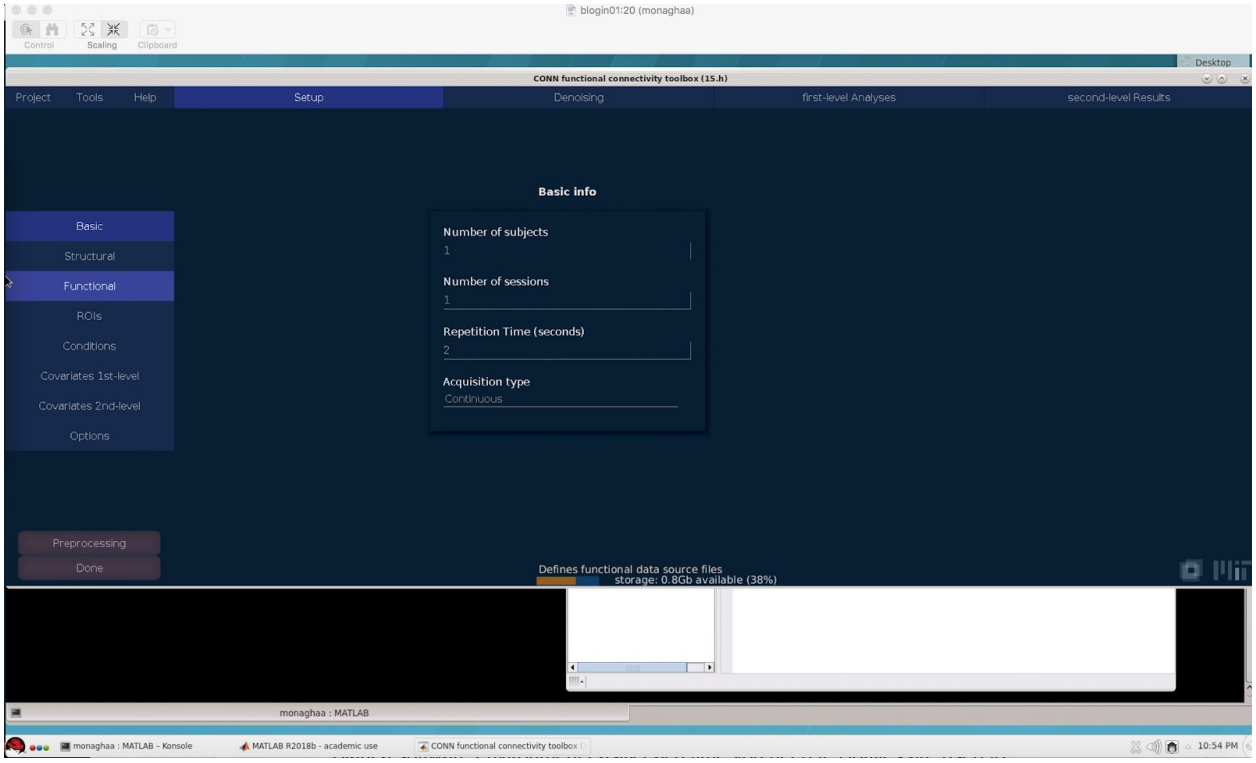
Step 4: From the spm GUI, start the package you desire by clicking on the desired button. For example, start the fMRI package:



If you wish to use the conn package, it can be invoked directly from the matlab command prompt, rather than from within spm:

```
>> conn
```





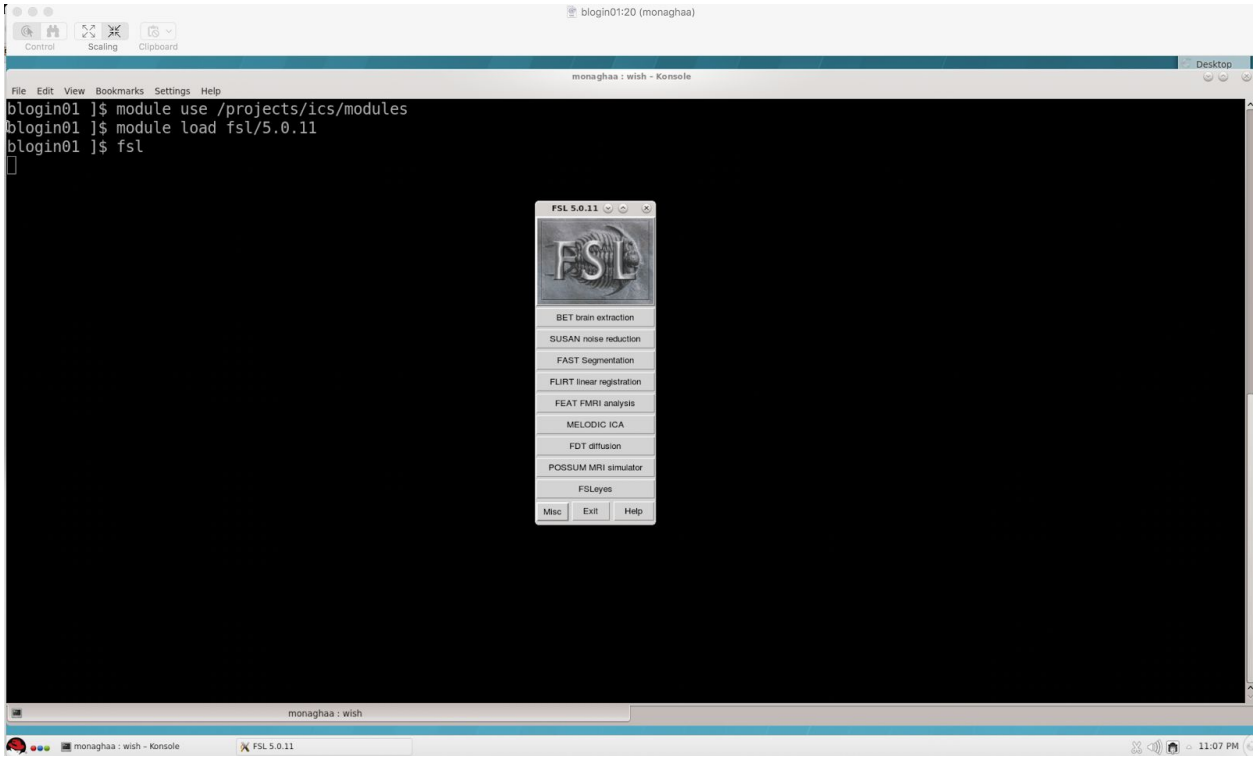
Using FSL (including MELODIC and fsleyes)

Step 1: From a terminal in your VNC desktop (and normally within an sinteractive job on a blanca-ics compute node), enable the ICS software modules and load fsl.

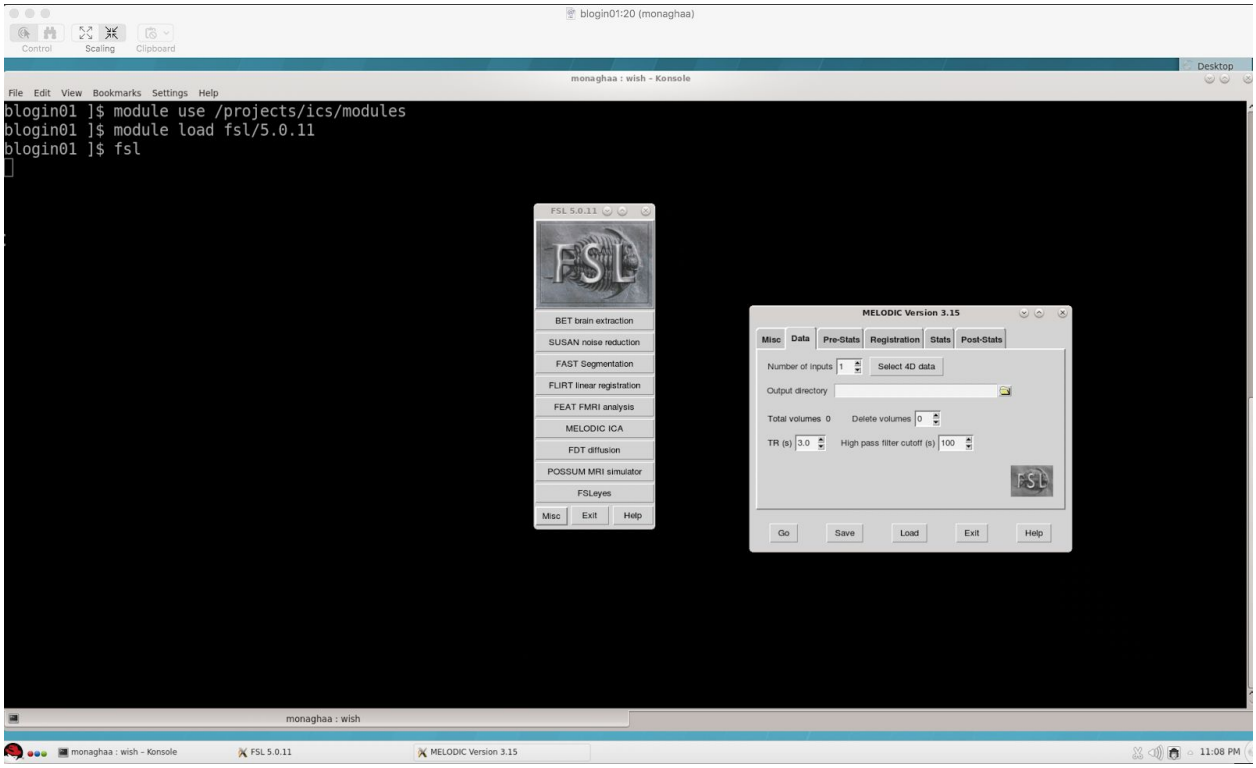
```
$ module use /projects/ics/modules
$ module load fsl/5.0.11
```

Step 2: Invoke fsl (it may take a few moments for the GUI to start)

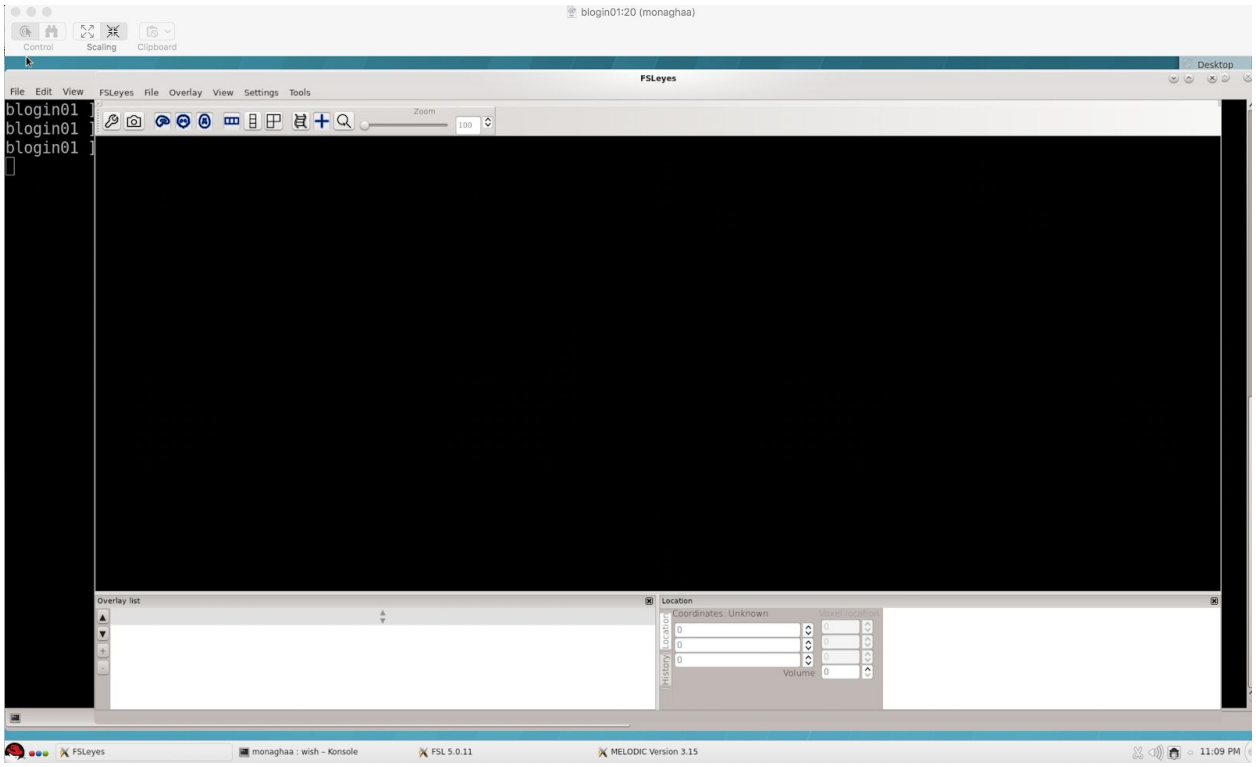
```
$ fsl
```



Step 3: click on the MELODIC button to start it.



Step 4: click on the FSLeyes button to start it.



Configuring your software environment with dotfiles (optional)

You can create a dotfile in your /home/$USER directory if you'd like to easily and quickly configure your Blanca software environment each time you start a new session. Dotfiles are used to configure all sorts of things in Linux environments; the name of the dotfile starts with a period (a dot). The period ensures the files stays hidden by default, to mitigate the chance that it will be inadvertently deleted.

Step 1: Use your favorite text editor to open a new dotfile in your /home/$USER directory (tip: the shorthand for your home directory is "~"). In this example we use nano and name the file ".myblancaenv".

```
$ nano ~/.myblancaenv
```

Step 2: Add the following lines to your new file, save and exit. You can customize the modules per your preferences (e.g, add, remove, or use different module versions).

```
#load matlab
module load matlab/R2018b

#enable the ics software modules
module use /projects/ics/modules

#load your favorite ics software
module load freesurfer/6.0.1
module load fsl/5.0.11
module load spm/spm12_v6470
module load conn/15h
```

Step 3: Now you can simply source your dotfile each time you login, and the ics module stack will be loaded, along with your most-used modules. “Sourcing” a file in linux executes the lines in it and makes sure that any changes apply to your current shell (terminal).

```
$ source ~/.myblancaenv
```

Final Advice

Chances are, just about anything you can do interactively in a GUI can be done in an unattended, non-interactive manner (via the command line). This likely applies to many of the tasks one can do in SPM12 and FSL. The true potential of high performance computing is often realized by learning how to do equivalent tasks at the command-line. This will, for example, enable you to submit Slurm batch jobs to process hundreds of tasks simultaneously, without the need for you to be present or interact with a GUI. Speaking with colleagues who have experience working with these software packages on Blanca is the best way to learn how to make your workflow more efficient and may save you countless hours of work.