How to run *R* on the HPC

June 25, 2009

Secure copy of bash script file to '$HOME' on HPC server following '$ scp source host:destination' structure, e.g.:

```
$ scp script.sh user@login.cx1.hpc.ic.ac.uk:/home/user/whatever/script.sh
```

Open a secure shell (ssh):

```
$ ssh user@login.cx1.hpc.ic.ac.uk
```

Check for available modules:

```
$ module avail
```

The script you will run needs a sha-bang (telling it what shell to run, usually bash), you need to allocate resources to PBS (such as walltime, number of processors, and memory[1], using the '#PBS' command), and tell it what *R* script to run. The script could look something like this:

```
#!/bin/bash
#PBS walltime=01:00:00#1 hour

module load R/2.8.1#Or whichever relevant version; default is old
module load intel-suite

R CMD BATCH $HOME/whatever/myR.r $HOME/whatever/myR.r.output
```

where 'myR.r.output' shows the output printed onto the R-console. The modules can also be loaded in the shell in order to test *R* codes directly.

In your *R* code you need to set the environment so that it knows its working directory and where to output files:

```
home <- Sys.getenv('HOME')
..
save(object, file='home/whatever/object.RData')
```

The job then needs to be queued using qsub (PBS):

[1]HPC uses Quad-core processors, with each processor having 1GB of RAM available. PBS defaults to 800MB of memory (thus allowing an extra 200MB of overhead which are usually added to the specified memory required). If more than 1GB is specified or used, that inhabilitates another processor, even if 1 is only used to process the script.

```
$ qsub -j eo script.sh
```

where '-j eo' is an option to join both output and error into one file. Running the script will produce a job output (anything that is printed on shell (e.g. 'echo')), and an error file (related to whether the script was successful or not), in the form of {scriptname}.o{job id} and {scriptname}.e{jobid}.

The 'qstat' command provides information on the job being submitted (which queue (short, medium, long), status, etc.) as well as information on all queues available (-q, -Q).