## Putting all pieces together - an R example

<table>
<thead>
<tr>
<th>Quick Links</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>HPC Home</strong></td>
</tr>
<tr>
<td><strong>Getting an account</strong></td>
</tr>
<tr>
<td><strong>Getting started on Prince</strong></td>
</tr>
<tr>
<td><strong>Prince How-to Articles</strong></td>
</tr>
<tr>
<td><strong>Logging in</strong></td>
</tr>
<tr>
<td><strong>Windows</strong></td>
</tr>
<tr>
<td><strong>Mac / Linux</strong></td>
</tr>
<tr>
<td><strong>Clusters and Storage</strong></td>
</tr>
<tr>
<td><strong>Prince (HPC)</strong></td>
</tr>
<tr>
<td><strong>Dumbo (Hadoop)</strong></td>
</tr>
<tr>
<td><strong>Brooklyn (OpenStack)</strong></td>
</tr>
<tr>
<td><strong>Dalmatia (NYU Abu Dhabi)</strong></td>
</tr>
</tbody>
</table>
# Running jobs on the Prince Cluster

Accessing the Prince Cluster
- From Windows workstation
- From Mac workstation

Software and Environment Module

Job script and resource request
- Introduction to job scheduling
- Submitting jobs with `sbatch`
- Requesting resources
- Using computing nodes interactively

Monitoring batch jobs
- Monitoring batch jobs - `squeue`
Pulling it all together - Preparing, submitting and monitoring a job on Prince

In this section we will prepare, submit and monitor a small R job. Our test case comes from the NYU Data Services “Introduction to R” tutorial.

Exercise
Start a terminal session on Prince and replicate this example in it.

Choose your own example
After - or instead of - following this example through, prepare and submit a run of something genuinely relevant to your research. This way, if you are doing this tutorial in a classroom, the presenter will be available should you have questions or strike difficulties.

We're using R, so first we'll look for available modules. On Prince:

$ module avail r

--------------------------------------------- /share/apps/modulefiles
---------------------------------------------
gstreamer/intel/1.10.2  mothur/intel/1.35.1  r/intel/3.3.2

There's a few modules starting with r, and a couple of versions of R. We'll use the latest version, 3.1.2.

$ module purge
$ module list
No Modulefiles Currently Loaded.
$ module load r/intel/3.3.2

Take a look at what it did:
... clearly, R uses a lot of other packages. The modulefile has looked after loading the correct ones.

```lua
$ module list
Currently Loaded Modulefiles:
  1) jdk/1.8.0_111  2) intel/17.0.1  3) openmpi/intel/2.0.1  4) r/intel/3.3.2
```

For our example, we'll get some code and data from /share/apps/examples:

```bash
$ mkdir /beegfs/$USER/R-example
$ cd !$
$ cp /share/apps/examples/r/basic/* .
```

Take a look at the job script:
$ cat my_R_job.s
#!/bin/bash
#SBATCH --nodes=1
#SBATCH --ntasks=1
#SBATCH --time=5:00
#SBATCH --mem=1GB
#SBATCH --job-name=myRtest
#SBATCH --mail-type=END
#SBATCH --mail-user=bob.smith@nyu.edu

module purge
module load r/intel/3.3.2

RUNDIR=$SCRATCH/R-example
mkdir -p $RUNDIR

cp /share/apps/examples/r/basic/example.R $RUNDIR

cd $RUNDIR
srun R CMD BATCH example.R example.out

There are a few steps we can try here:

1. Start an interactive batch session, and run the example.R script interactively
2. Close the interactive session, and submit the batch script as a job:

   $ sbatch my_R_job.s

   You'll get a job id returned.

   Is it running yet?

   $ squeue -u $USER

   You could watch the output in the run directory:

   $ ls -l ${SCRATCH}/R-example

   Finally, when the job finishes, you should see a .out file in the directory you submitted from.

**Exercise**

Experiment with sbatch options for the job name, output and error file merging and location, resource limits.