<table>
<thead>
<tr>
<th>Quick Links</th>
</tr>
</thead>
<tbody>
<tr>
<td>HPC Home</td>
</tr>
<tr>
<td>Getting an account</td>
</tr>
<tr>
<td>Gentle Introduction to using HPC</td>
</tr>
<tr>
<td>Getting started on Prince</td>
</tr>
<tr>
<td>Prince How-to Articles</td>
</tr>
<tr>
<td>Logging in</td>
</tr>
<tr>
<td>Windows</td>
</tr>
<tr>
<td>Mac / Linux</td>
</tr>
<tr>
<td>Clusters and Storage</td>
</tr>
<tr>
<td>Prince (HPC)</td>
</tr>
<tr>
<td>Dumbo (Hadoop)</td>
</tr>
<tr>
<td>Brooklyn (OpenStack)</td>
</tr>
</tbody>
</table>
Dalmata (NYU Abu Dhabi)

Transferring data to/from the clusters

Transferring data to/from Prince cluster using Globus

Submitting jobs with sbatch

Available software

Licensed Software Available on the HPC Cluster

Building Software

Slurm Tutorial

Tutorials

FAQs

Scratch Area Cleanup

Programming for Biologists
Basic R jobs

Multiple R versions exist in HPC environment. To check what are available, on Prince:

$ module avail r

<table>
<thead>
<tr>
<th>Package</th>
<th>Version</th>
<th>Location</th>
</tr>
</thead>
<tbody>
<tr>
<td>r/intel/3.0.3</td>
<td></td>
<td>ray/openmpi/intel/20160114</td>
</tr>
<tr>
<td>reprozip/intel/1.0.3</td>
<td></td>
<td>rpy2/intel/2.5.6</td>
</tr>
<tr>
<td>r/intel/3.1.2</td>
<td></td>
<td>rdkit/intel/201409.2</td>
</tr>
<tr>
<td>rsem/intel/1.2.15</td>
<td></td>
<td>requests/2.7.0</td>
</tr>
<tr>
<td>r/intel/3.2.0</td>
<td></td>
<td>rdp_classifier/2.11</td>
</tr>
<tr>
<td>ribopicker/0.4.3</td>
<td></td>
<td>rseqc/intel/2.3.9</td>
</tr>
<tr>
<td>r/intel/3.2.2</td>
<td></td>
<td>recon/intel/1.08</td>
</tr>
<tr>
<td>rstudio/0.98.1028</td>
<td></td>
<td>rmblast/2.2.28</td>
</tr>
<tr>
<td>randfold/intel/2.0</td>
<td></td>
<td>rendertoolbox3/2.1-18</td>
</tr>
<tr>
<td>rtax/0.984</td>
<td></td>
<td>rose/20151118</td>
</tr>
<tr>
<td>raxml/intel/7.3.0</td>
<td></td>
<td>repeat_masker/4.0.5</td>
</tr>
<tr>
<td>rosetta/intel/54167</td>
<td></td>
<td>ruby/gnu/2.1.1</td>
</tr>
<tr>
<td>raxml/intel/8.0.23</td>
<td></td>
<td>repeatmodeler/1.0.8</td>
</tr>
<tr>
<td>rosetta/openmpi/intel/2014.35.57232</td>
<td></td>
<td></td>
</tr>
<tr>
<td>raxml/intel/8.2.5</td>
<td></td>
<td>repeatscout/intel/1.0.5</td>
</tr>
<tr>
<td>rosetta/openmpi/intel/54167</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Suppose we want to use 3.3.2, run these commands:
We first clean up the environment by doing `module purge`. Then we load the R version selected, check what are available in current environment. We can see that R 3.3.2 is indeed loaded along with its dependency modules. Let's try this basic R example. We name it "example.R":

```r
df <- data.frame(x=c(1,2,3,1), y=c(7,19,2,2))
df
indices <- order(df$x)
order(df$x)
df[indices,]
df[rev(order(df$y)),]
```

Below is the screen scene while running it on Prince:
$ R

R version 3.3.2 (2016-10-31) -- "Sincere Pumpkin Patch"
Copyright (C) 2016 The R Foundation for Statistical Computing
Platform: x86_64-centos-linux-gnu (64-bit)
R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.
Natural language support but running in an English locale

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.
Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

> df <- data.frame(x=c(1,2,3,1), y=c(7,19,2,2))
> df
   x  y
1 1  7
2 2 19
3 3  2
4 1  2
> indices <- order(df$x)
> order(df$x)
[1] 1 4 2 3
> df[indices,]
   x  y
1 1  7
3 3  2
4 1  2
2 2 19
> df[rev(order(df$y)),]
   x  y
2 2 19
1 1  7
4 1  2
3 3  2
> quit()

What is shown above is a simple demo case on login nodes. For real interactive analysis scenario, users are encouraged to run on compute
nodes using the 'srun' command to request dedicated resources, e.g.:

$ srun --x11 --nodes=1 --ntasks-per-node=4 --mem=4000 -t2:00:00 --pty /bin/bash
$ xterm
$ module load r/intel/3.3.2
$ R

Besides running our analysis interactively, long running and big data crunching jobs ought to be submitted to the batch system slurm. The
"example.R" can be submitted to slurm to run in batch mode.

Copy example files to your newly created directory.
Below is how the example looks like:

```r
df <- data.frame(x=c(1,2,3,1), y=c(7,19,2,2))
indices <- order(df$x)
order(df$x)
df[indices,]
df[rev(order(df$y)),]
```

Then create a sbatch job script as:

```bash
#!/bin/bash
#SBATCH --job-name=RTest
#SBATCH --nodes=1
#SBATCH --tasks-per-node=1
#SBATCH --mem=2GB
#SBATCH --time=01:00:00

module purge
module load r/intel/3.3.2

cd /scratch/$USER/example
R --no-save -q -f example.R > example.out 2>&1
```

Once the sbatch script file is ready, it can be submitted to the job scheduler using sbatch. After successful completion of job, verify output log file for detail output information.

```bash
sbatch run-R.sbatch
```