Environment modules

In order to provide multiple versions of software compiled with different and varying libraries, we use modules. Modules allow us to add and remove software from our $PATH using a simple `module` command.

Common usage

- `module avail [name]` View all the modules on the HPC system filtered by `name`.
- `module list` View currently loaded modules
- `module load moduleName` Load the module denoted by `moduleName`
- `module purge` Unload all loaded modulefiles

Loading the R module

First we want to take a look at what R modules are available. To do that issue:

```
module avail R
```

`[john@discovery ~]$ module avail R`

`---------- /dartfs-hpc/admin/opt/modules/el7 ----------`

`R/4.0 R/4.0.4(default) R/4.1.2`

As you can see from this output there are multiple versions of R available. The one with (default) next to it is the default module and will be picked unless you specify the full path. For example, if I wanted R/4.1.2 I would issue:

```
module load R/4.1.2
```

If I did not care about what version I recieved then I can just issue module load R.

```
module load R
```

Once you have loaded the module you would like you can issue module list to see what you have currently loaded.

```
module list
```

```
[john@discovery ~]$ module load R
[john@discovery ~]$ module list
Currently Loaded Modulefiles:
  1) R/4.0.4
```

> Since I did not specify a version, the module command pulled in the (default) version of R, which happens to be R/4.0.4