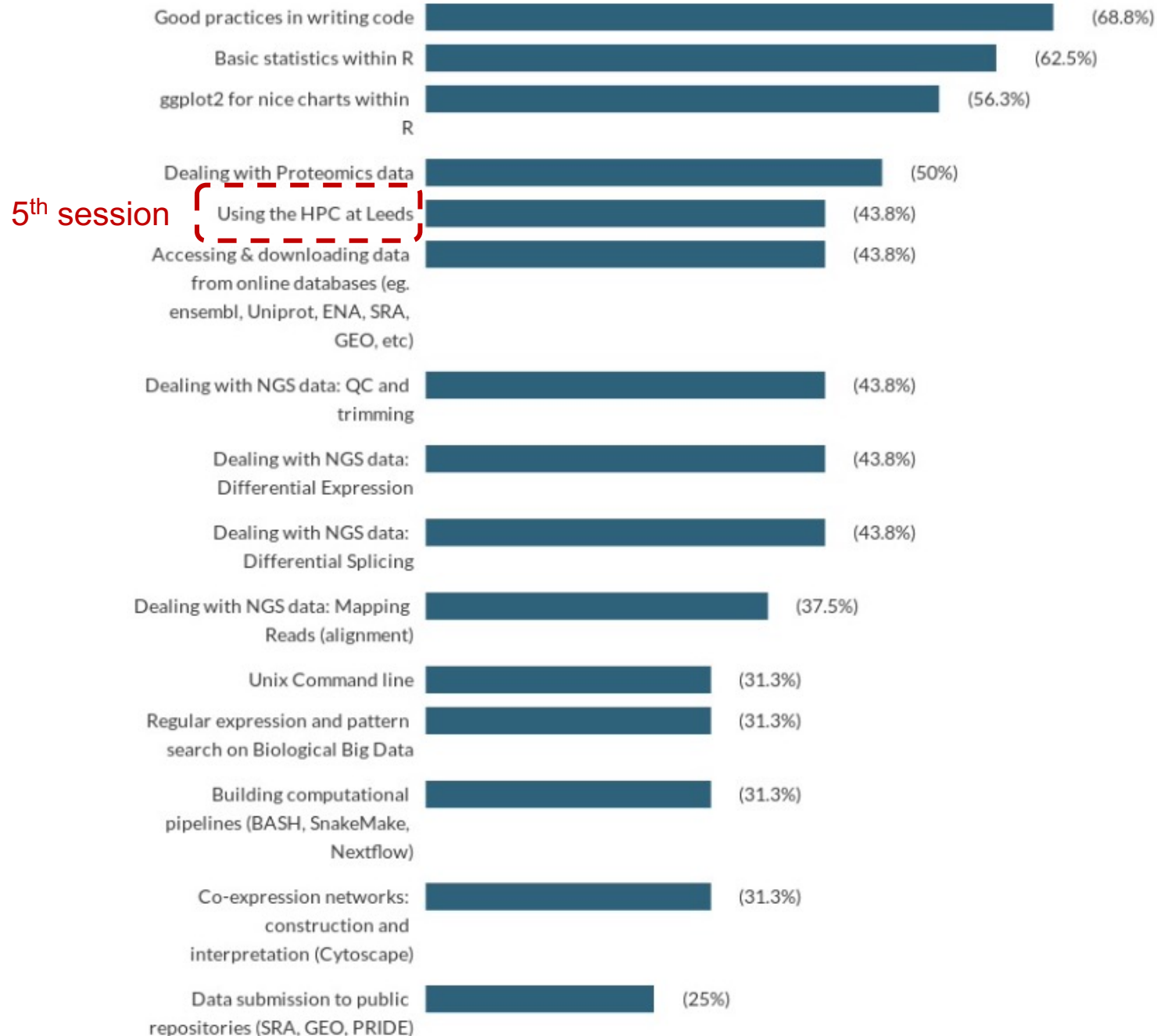


www.leedsomics.org
@leedsomics
omics@leeds.ac.uk

Unix command line and using HPC @ UoLeeds

Club Moderators: Elton Vasconcelos, Euan McDonell

Topics to be addressed - Survey Result (2021-22)



UNIX Tutorial for Beginners

A beginners guide to the **Unix** and **Linux** operating system. Eight simple tutorials which cover the basics of UNIX / Linux commands.

[Introduction to the UNIX Operating System](#)

- What is UNIX?
- Files and processes
- The Directory Structure
- Starting an UNIX terminal

[Tutorial One](#)

- Listing files and directories
- Making Directories
- Changing to a different Directory
- The directories . and ..
- Pathnames
- More about home directories and pathnames

[Tutorial Two](#)

- Copying Files
- Moving Files
- Removing Files and directories
- Displaying the contents of a file on the screen
- Searching the contents of a file

[Tutorial Three](#)

- Redirection
- Redirecting the Output
- Redirecting the Input
- Pipes

[Tutorial Four](#)

- Wildcards
- Filename Conventions
- Getting Help

[Tutorial Five](#)

- File system security (access rights)
- Changing access rights
- Processes and Jobs
- Listing suspended and background processes
- Killing a process

[Tutorial Six](#)

- Other Useful UNIX commands

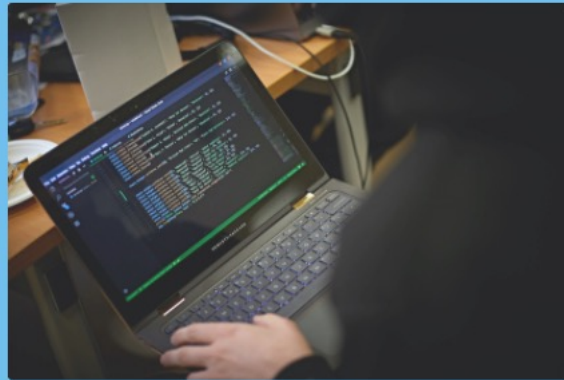


RESEARCH COMPUTING



Platforms

Learn about the Research Computing platforms available at Leeds including HPC, Cloud and more.



Consulting

Learn about our Research Software Engineer Consulting service, helping provide bespoke expertise to your research project.



Training

Read more about the wide range of training we offer from introduction to Linux to Advanced Python courses.

Typical header of a job script (job.sh)

```
## -cwd
## -V
## -l h_rt=16:00:00,h_vmem=2G
## -l nodes=1,ppn=24
## -m be
## -M user@leeds.ac.uk
## -l node_type=24core-768G    #Use just in case of a massive memory-consuming job

# Then add your job's command line(s) below
program1 -infile xxx.in -option1 -option2 -option3 -outfile xxx.out
program2 -infile xxx.out -option1 -option2 -option3 -outfile yyy.out
```


Common Grid Engine commands

To check your job status

To submit your job script

To delete your job from either the queue or running process

```
qstat .....list current user's jobs
qstat -u "*" ..... list all users jobs
qstat -g c ..... show available nodes and system load
qsub [options] job.sh .....submit a shell script job.sh to Grid Engine
qsh [options] ..... request queued interactive X-window session
qrsh [options] .....request a queued interactive "rsh" session
qdel job-id .....delete job
```

Commonly used options to qsub and queue configuration

Options can be specified after qsub or embedded in a shell script using "\$#". For a full list of options look at the manual page, `man qsub`. There is a single queue configured, with a maximum runtime of 48 hours.

```
-l h_rt=hh:mm:ss .....requested wall clock time, max is 48 hours. Required
-l h_vmem=memory ..... Sets limit of virtual memory/core. Default is 1G/core
-cwd .....run from current working directory. Recommended
-V ..... export current environment variables including modules. Recommended
-pe ib np ..... MPI parallel job on np cores
-pe smp np ..... MPI or OpenMP parallel job, on np shared memory cores
-m be ..... send mail at beginning and end of job to the owner
-M email_address ..... specify email address for the -m option
```

Node syntax (preferred for MARC1, ARC2 and Polaris)

Can be used instead of `-pe ib np` or `-pe smp np` to request exclusive access to nodes.

```
-l nodes=n ..... Request n full nodes (16 cores and 32GB memory)
-l np=n ..... Request n processes (16 gives 1 node, 32 gives 32 nodes etc.)
```

Alternatively, to specify a different number of cores per node (to give more memory per core) and/or threads per process:

```
-l nodes=<w>[,ppn=<y>][,tpp=<z>]
-l np=<x>[,ppn=<y>][,tpp=<z>]
```

Where:

```
w = number of nodes requested
x = number of processes requested
y = number of processes per node
z = number of threads per process
```

If y and z are omitted, the scheduler sets: y = number of cores in each node and

z = 1 If y is present and z is omitted, the scheduler sets: z = int (num cores / y)

If z is present and y omitted, the scheduler sets: y = int (num cores / z)

Bring your issues on!