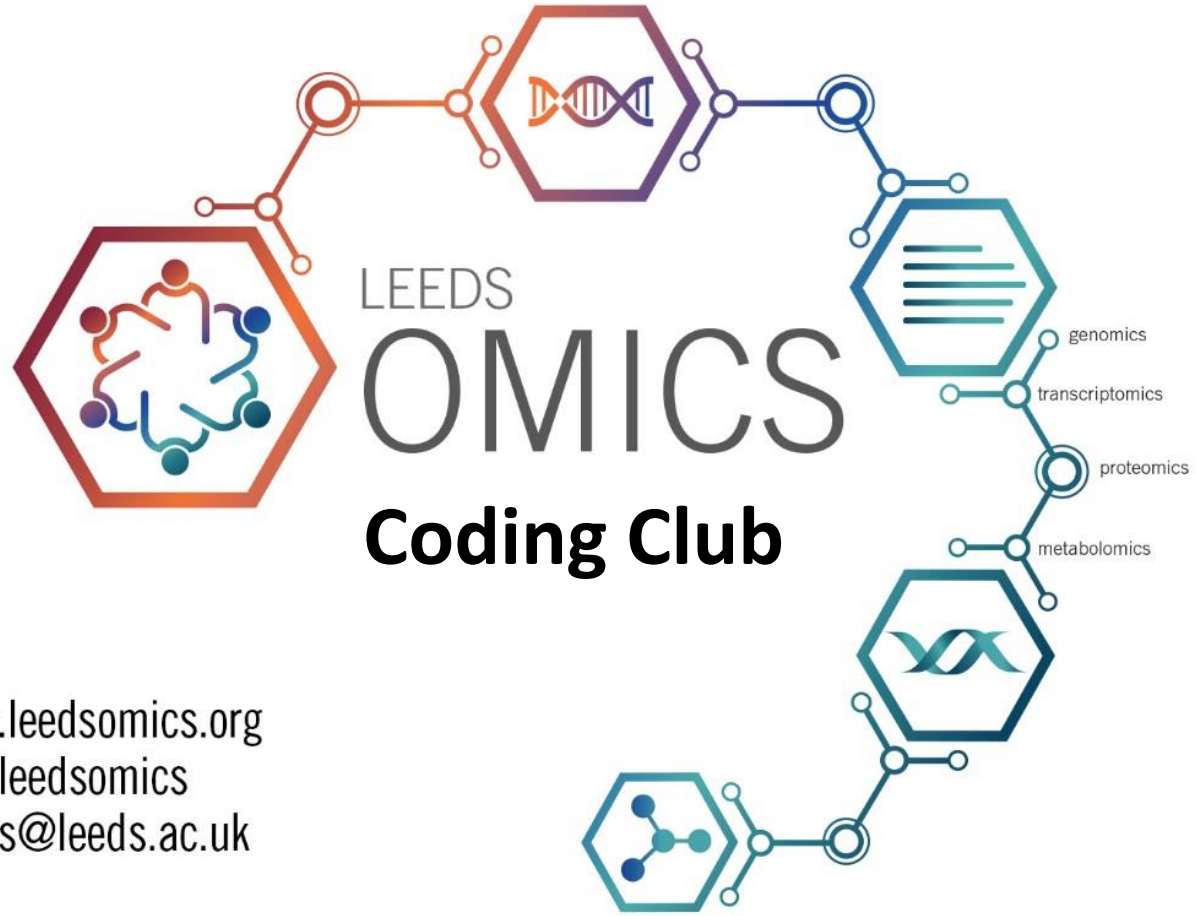

Coding Club

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

Using the HPC at UoL

Club Moderators: Elton Vasconcelos, Peter Mulhair, Euan McDonell, Chew Cheng, and Dapeng Wang

Topics to be addressed - **Survey Result**

https://arc.leeds.ac.uk/

ARC Advanced Research Computing

UNIVERSITY OF LEEDS

Home

We provide a comprehensive **Research Computing** support service for researchers in all faculties at the University of Leeds.

The ARC HPC systems

Service provided by Research Computing include:

A range of **Research Software Engineering**, consultancy and training services including:

- user guidance and **training**
- installing and supporting software applications on the HPC service
- improving software performance and handling large data sets
- guidance and support in meeting current and future research challenges with HPC
- how to incorporate HPC costings into grant applications

Typical header of a job script (job.sh)

```
#$ -cwd
#$ -V
#$ -l h_rt=16:00:00
#$ -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
```

To check your job status

To submit your job script

To delete your job from either the queue or running process

Common Grid Engine commands

`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. **Recomended**
`-V` ..... export current environment variables including modules. **Recomended**
`-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
w = 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)

Courtesy material from ARC training courses

Bring your issues on!