Genomics England

Running workflows on the HPC

Emily Perry, Zander Mears, Alex Ho, Ken Hanscombe

12th December 2023

Data security

- This training session will include data from the GEL Research Environment
- As part of your IG training you have agreed to not distribute these data in any way
- You are not allowed to:
  - Invite colleagues to watch this training with you
  - Take any screenshots or videos of the training
  - Share your webinar link (we will remove anyone who is here twice)
- We will record this training and distribute the censored video afterwards

Trainers

Emily Perry
Research Engagement Manager

Zander Mears
HPC Platform Engineer

Alex Ho
Bioinformatician - Research Services

Ken Hanscombe
Bioinformatician - Research Services

Questions

Your microphones are all muted

Use the Zoom Q&A to ask questions

Upvote your favourite questions: if we are short on time we will prioritise those with the most votes.

Agenda

1. Introduction and admin
2. What is the High Performance Cluster?
3. Why use the HPC?
4. Logging into the HPC and working in an interactive session
5. How to create and monitor jobs on the HPC
6. Tools and software available and how to load them
7. Interactive coding tools
8. Bringing in your own tools and software
9. Questions

2. What is the High Performance Computing Cluster?

What is High Performance Computing?

High Performance Computing or HPC is set of computing, networking and storage resources integrated with workload orchestration services for HPC applications. Example use cases are:

- Analytics for financial services
- Manufacturing
- Scientific visualization and simulation
- Genomic sequencing and medical research
- Oil & gas
- Weather prediction

GEL's research environment HPC is a number of computers and a Weka storage system linked together by high-speed Local Area Network (LAN). This is all managed by a scheduling software from IBM called Load Sharing Facility (LSF).

Collectively this is known as the 'Helix' Cluster.

Types of cluster hosts/nodes

Management/Master Hosts:

Coordinator of the cluster, scheduling and dispatching of jobs based on cluster configuration and load. Will not run jobs.

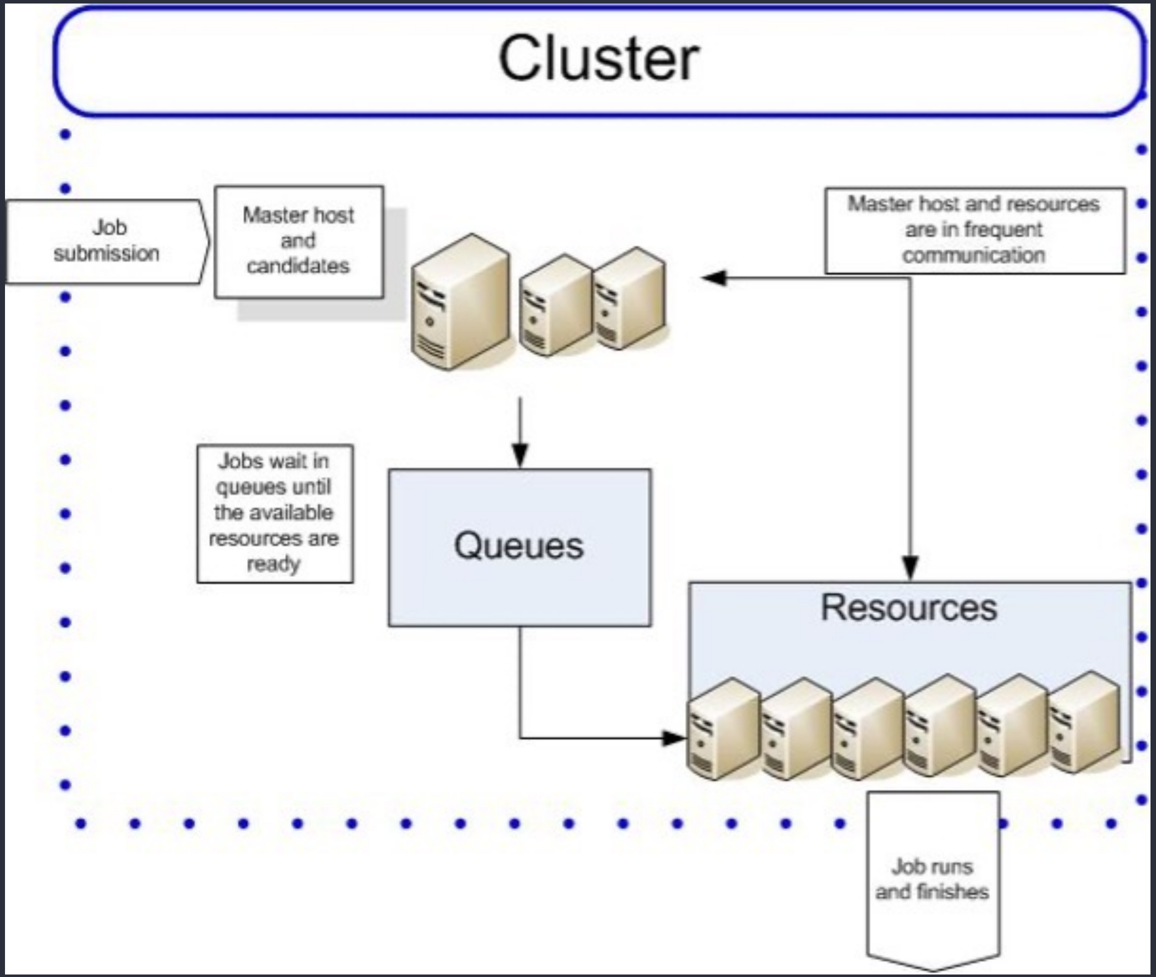
Client/Submission Hosts:

Submits jobs.

Execution/Server Hosts:

Runs jobs. These are the resources in the diagram.

Host and Node mean the same and can be used interchangeably.


Cluster
Job submission
Master host and candidates
Master host and resources are in frequent communication
Jobs wait in queues until the available resources are ready
Queues
Resources
Job runs and finishes

Helix

Three master nodes, primary, secondary and tertiary. Multi master setup in an active/passive configuration, automatic failover.

Four login nodes, used for different groups.

Three server nodes for interactive jobs. 50 job slots per node, maximum of five jobs per user.

54 server nodes for all other jobs. 34 job slots per node, maximum of 600GB memory per job.

Helix

- Each server node has dual 50Gb/s connections to the LAN
- Weka backend servers are connected via dual 100Gb/s
- LAN is non-blocking architecture
- Each connected host can run at full speed without the network throttling

Queues

Inter – Interactive jobs only

Pipeline – Pipeline divisional queue

Short – For jobs running up to four hours.

Medium – For jobs running up to 24 hours.

Long - For jobs running up to one week.

Job scheduling

When a job is submitted to the cluster the management node will allocate the job to a server immediately if the resources are available otherwise, they get placed into a queue waiting for dispatch.

There are many scheduling configurations and submission options which affect how long your job takes to get dispatched to a server node for execution.

Resource requirements can affect the time it takes for the job to be submitted to an execution host. More resources requested, busier the cluster, the longer it will take to be dispatched. We have configured the cluster such that a job should never be permanently waiting for resources. Nonetheless it is recommended to be conservative with jobs resource requirements and review usage in the job output file.

Queue prioritisation

Queues in order of dispatch priority:

Inter, Pipeline, Short, Medium, Long

Fairshare

LSF will dispatch jobs from queues based on user shares, most users have equal share. There are some exceptions who have slightly more.

Pre-emption

Pending jobs in higher priority queues can pre-empt and suspended (SSUP status) running jobs in lower priority queues in order of preference:

Pipeline -> Long, Medium the Short

Short -> Long then Medium

Medium -> Long

Guaranteed Resource Pool

Helix queues have a guaranteed minimum number of hosts for running jobs:

Pipeline: 5

Short: 20

Medium: 15

Long: 15

Connecting to the cluster for the first time

Create ssh keys:

```
ssh-keygen
```

Copy over the ssh key to the Helix login node:

```
ssh-copy-id
<username>@corp.gel.ac@<login node>.int.corp.
gel.ac
```

Enter password

Create ssh config file:

```
nano ~/.ssh/config
```

Add the following:

```
host helix-login
    Hostname <login node>.int.corp.gel.ac
    User <username>@corp.gel.ac
    IdentityFile ~/.ssh/id_rsa
```

Save file you will then be able to login via:

```
ssh helix-login
```

3. Why use the HPC?

Why use the HPC?

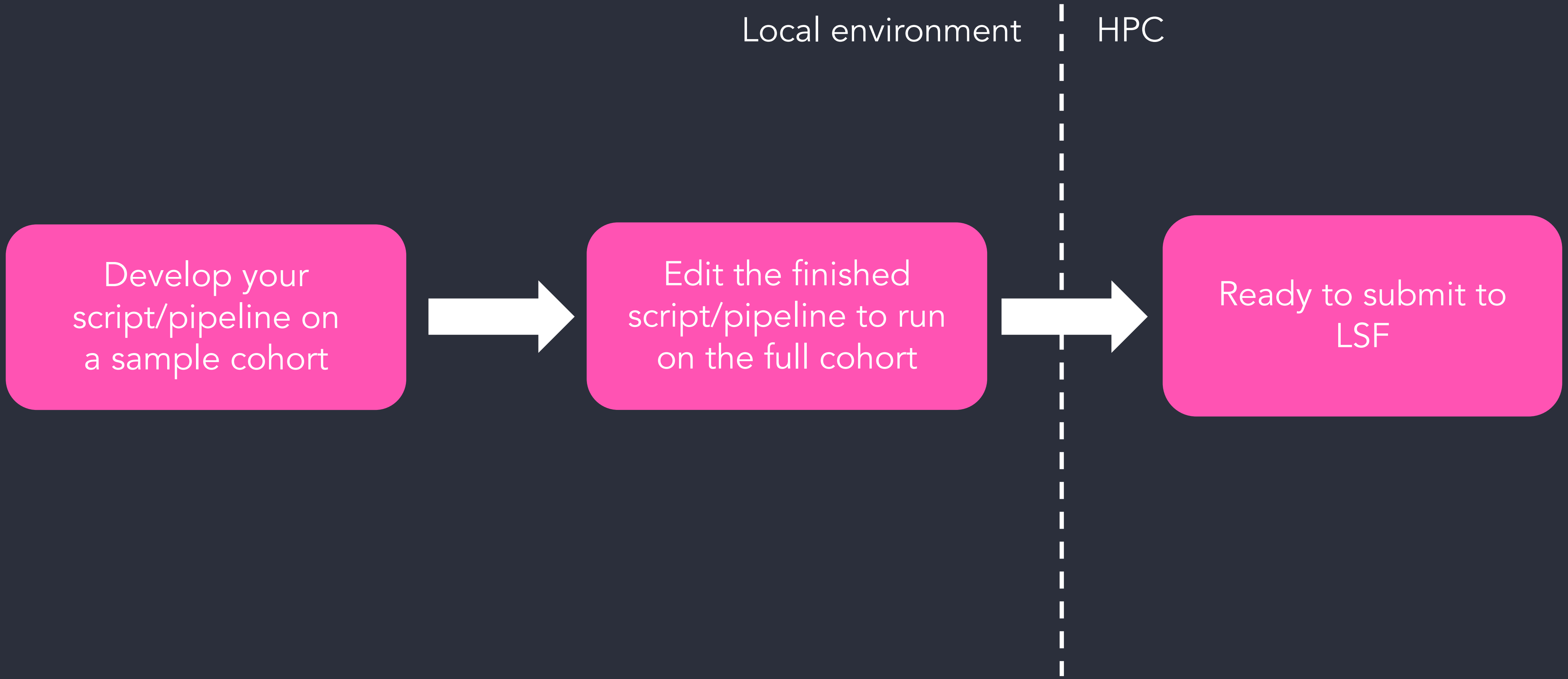
The HPC enables you to solve larger, more complex problems in far less time…

1. Increased performance when running complex scripts and pipelines
2. Avoid RAM cap in the Research Environment desktop

The local environment of the RE is designed for exploratory work.

The HPC is provided for "heavyweight" batch research.

Why use the HPC?

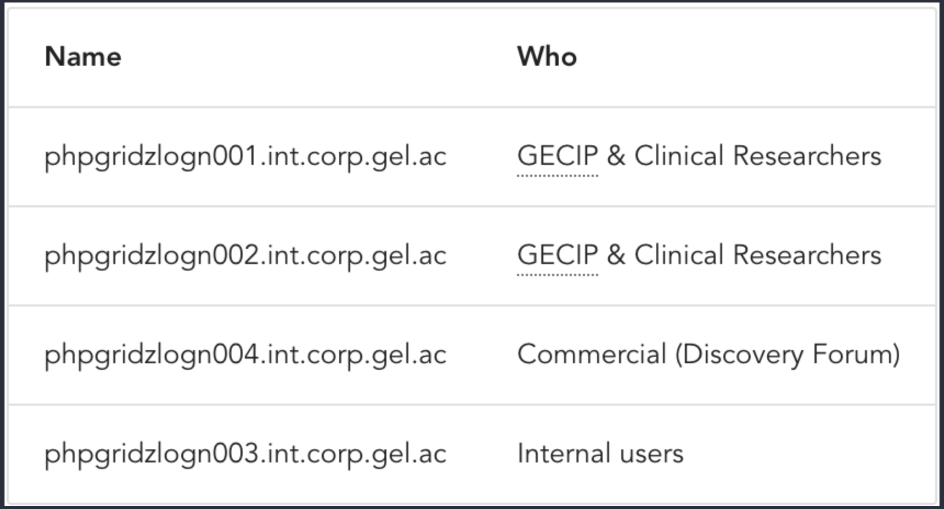

Local environment
HPC
Develop your script/pipeline on a sample cohort
Edit the finished script/pipeline to run on the full cohort
Ready to submit to LSF

4. Logging into the HPC and working in an interactive session

Logging into the HPC: the Login nodes

When you log into our HPC , you will be connected to one of the log in nodes:

```
ssh username@corp.gel.ac@phpgridzlogn001.int.corp.gel.ac

username@corp.gel.ac@phpgridzlogn001.int.corp.gel.ac's password:

>[username@corp.gel.ac@phpgridzlogn001]$
```



Name	Who
phpgridzlogn001.int.corp.gel.ac	GECIP & Clinical Researchers
phpgridzlogn002.int.corp.gel.ac	GECIP & Clinical Researchers
phpgridzlogn004.int.corp.gel.ac	Commercial (Discovery Forum)
phpgridzlogn003.int.corp.gel.ac	Internal users

Further reading: https://re-docs.genomicsengland.co.uk/hpc_access/

A word of caution…

- The login nodes act as a portal to the HPC for navigating our folders and submitting your jobs
- They have **not** been designed to handle the processing of data directly
- Unauthorised tools will not be permitted to run on the login nodes and may be terminated without warning

Further reading: https://re-docs.genomicsengland.co.uk/capping/

Command line coding on the HPC: the Interactive nodes

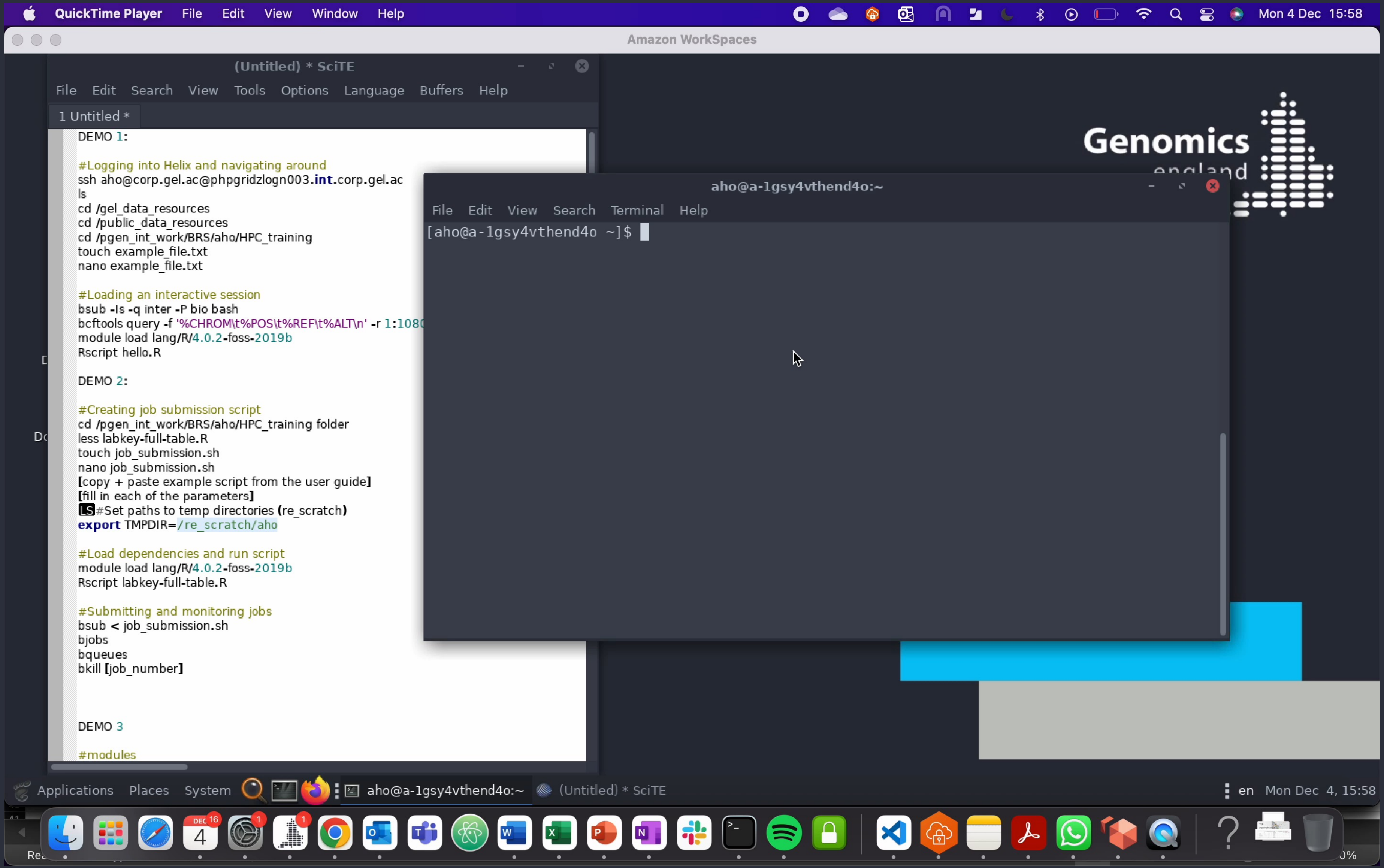
- Inter nodes: Access to HPC compute through a familiar terminal interface
- Useful for exploratory analysis and developing scripts that will later be scaled up

To start an interactive bash session:

```
bsub -Is -q inter -P <project code> bash

Job <931873> is submitted to queue <inter>.
<<Waiting for dispatch ...>>
<<Starting on phpgridzlsfe031>>
[username@corp.gel.ac@phpgridzlsfe031 ~]$
```

Further reading: https://re-docs.genomicsengland.co.uk/hpc_jobs/

Demo: Login and Inter nodes



```
DEMO 1:

#Logging into Helix and navigating around
ssh aho@corp.gel.ac@phpgridzlogn003.int.corp.gel.ac
ls
cd /gel_data_resources
cd /public_data_resources
cd /pgen_int_work/BRS/aho/HPC_training
touch example_file.txt
nano example_file.txt

#Loading an interactive session
bsub -Is -q inter -P bio bash
bcftools query -f '%CHROM\t%POS\t%REF\t%ALT\n' -r 1:108
module load lang/R/4.0.2-foss-2019b
Rscript hello.R

DEMO 2:

#Creating job submission script
cd /pgen_int_work/BRS/aho/HPC_training folder
less labkey-full-table.R
touch job_submission.sh
nano job_submission.sh
[copy + paste example script from the user guide]
[fill in each of the parameters]
LS#Set paths to temp directories (re_scratch)
export TMPDIR=/re_scratch/aho

#Load dependencies and run script
module load lang/R/4.0.2-foss-2019b
Rscript labkey-full-table.R

#Submitting and monitoring jobs
bsub < job_submission.sh
bjobs
bqueues
bkill [job_number]



DEMO 3

#modules
```

```
[aho@a-1gsy4vthend4o ~]$
```

5. How to create and monitor jobs on the HPC

Running larger analyses: LSF and the HPC Queues

- For most HPC use cases, your full analysis scripts should be submitted as "jobs" to the LSF

- The Load Sharing Facility balances job load, providing access to the HPCs full compute
  - Mature, commercial job scheduler by IBM
  - Takes your job requirements
  - Finds the best resources to run the job
  - Monitors its progress and provides detailed logging

- LSF jargon:
  - Job: A logical unit for application, environment and resources when computing a task
  - Cluster: A group of hosts running LSF that work together as a single unit
  - Queue: A scheduling entity where to-be-run and running jobs reside
  - Execution host: The actual compute resource (node) that runs a job
  - Priority: Priority at a technical level (not implying a job is more important than another)

Further reading: https://www.ibm.com/docs/en/spectrum-lsf/10.1.0?topic=overview-lsf-introduction

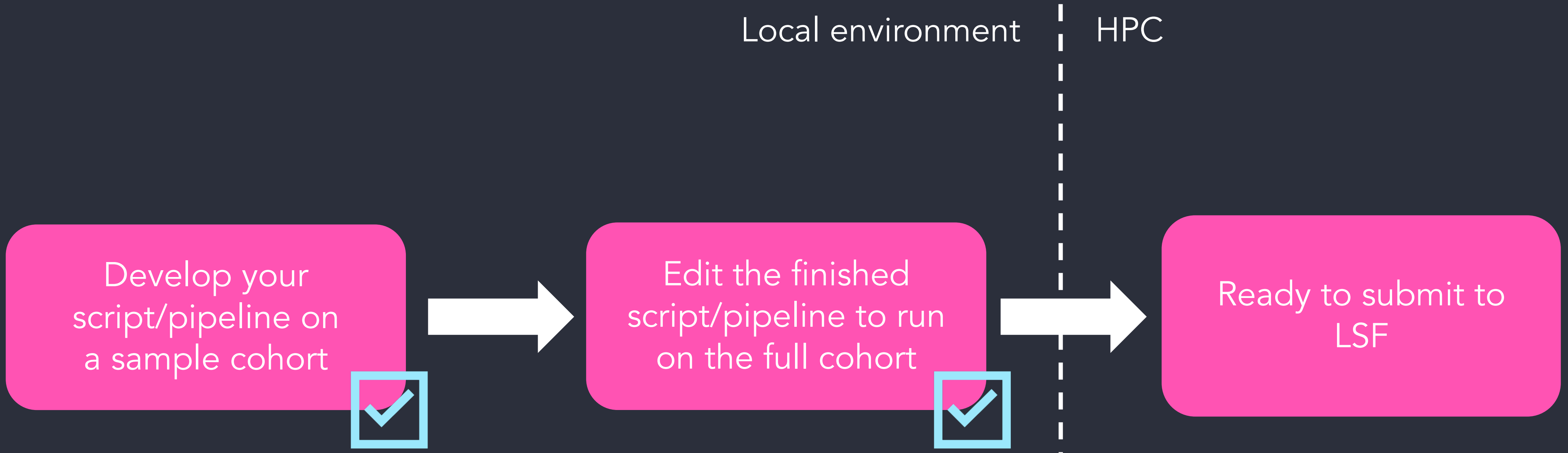
Running larger analyses: LSF and the HPC Queues

- There are four available queues you may specify to the LSF, select according to expected job length:

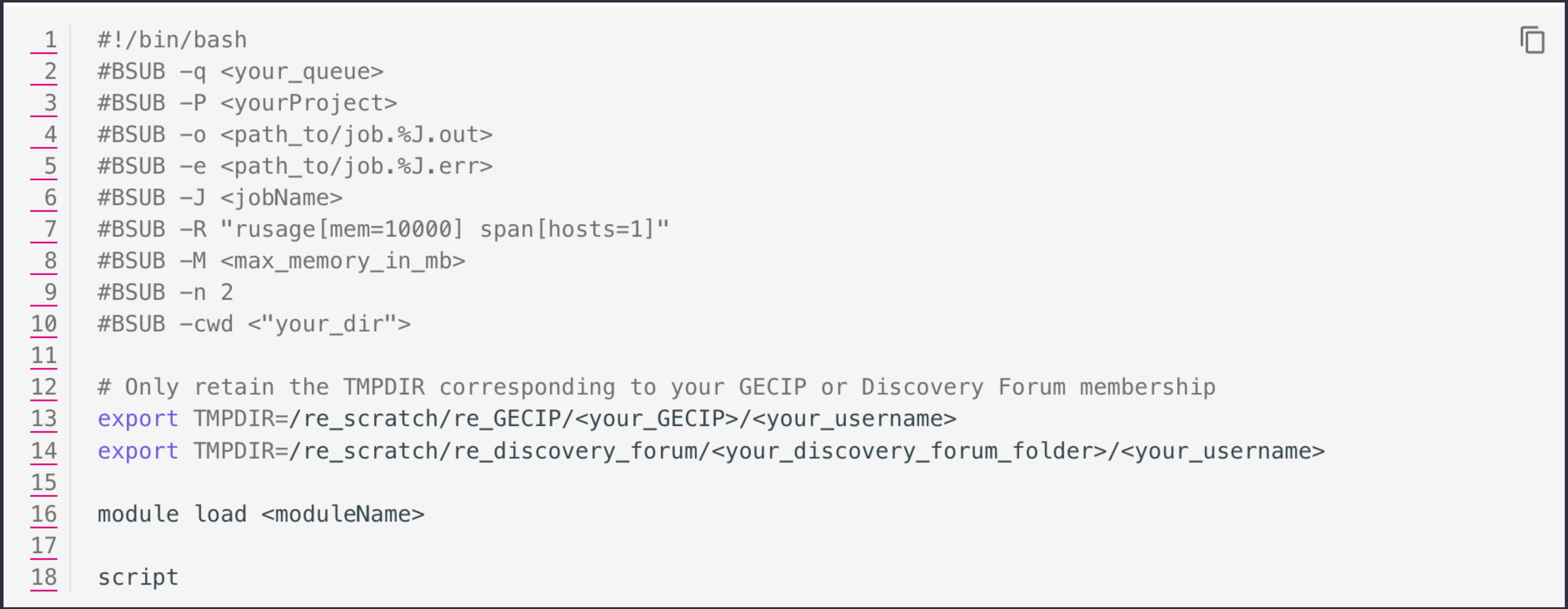
Inter	Interactive jobs: • Lightweight interactive or GUI tools • Require constant connection to the submission shell • Auto-killed after 2 weeks
	Batch jobs (disconnected from the submission shell):
Short	<4hrs
Medium	<24hrs
Long	<7days (or longer if explicitly configured)

Further reading: https://re-docs.genomicsengland.co.uk/hpc_jobs/

Running larger analyses: LSF and the HPC Queues


Local environment
HPC
Develop your script/pipeline on a sample cohort
Edit the finished script/pipeline to run on the full cohort
Ready to submit to LSF

Typical LSF submission



```
#!/bin/bash
#BSUB -q <your_queue>
#BSUB -P <yourProject>
#BSUB -o <path_to/job.%J.out>
#BSUB -e <path_to/job.%J.err>
#BSUB -J <jobName>
#BSUB -R "rusage[mem=10000] span[hosts=1]"
#BSUB -M <max_memory_in_mb>
#BSUB -n 2
#BSUB -cwd <"your_dir">

# Only retain the TMPDIR corresponding to your GECIP or Discovery Forum membership
export TMPDIR=/re_scratch/re_GECIP/<your_GECIP>/<your_username>
export TMPDIR=/re_scratch/re_discovery_forum/<your_discovery_forum_folder>/<your_username>

module load <moduleName>

script
```

bsub < submission_script.sh

Further reading: https://re-docs.genomicsengland.co.uk/hpc_jobs/

Typical LSF submission

```bash
#!/bin/bash
#BSUB -q <your_queue>
#BSUB -P <yourProject>
#BSUB -o <path_to/job.%J.out>
#BSUB -e <path_to/job.%J.err>
#BSUB -J <jobName>
#BSUB -R "rusage[mem=10000] span[hosts=1]"
#BSUB -M <max_memory_in_mb>
#BSUB -n 2
#BSUB -cwd <"your_dir">

# Only retain the TMPDIR corresponding to your GECIP or Discovery Forum membership
export TMPDIR=/re_scratch/re_GECIP/<your_GECIP>/<your_username>
export TMPDIR=/re_scratch/re_discovery_forum/<your_discovery_forum_folder>/<your_username>

module load <moduleName>

script
```

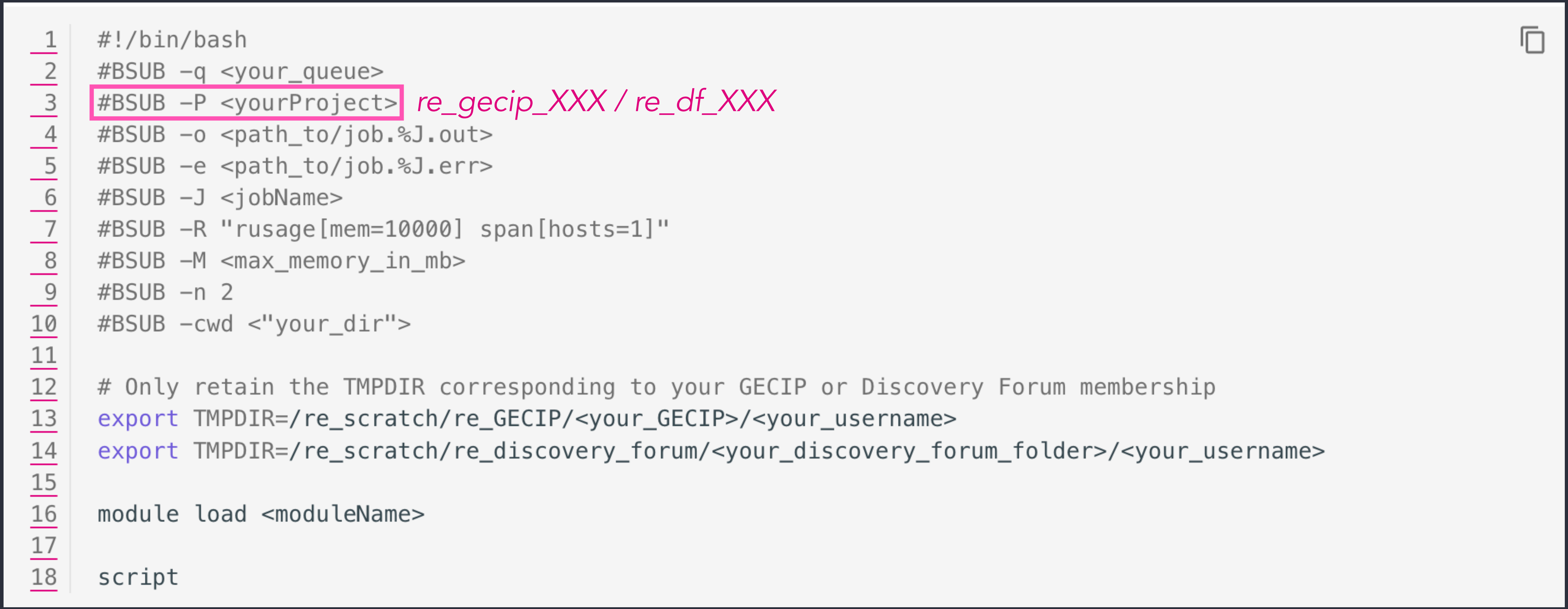
short, medium, long

bsub < submission_script.sh

Further reading: https://re-docs.genomicsengland.co.uk/hpc_jobs/

Genomics England

Typical LSF submission



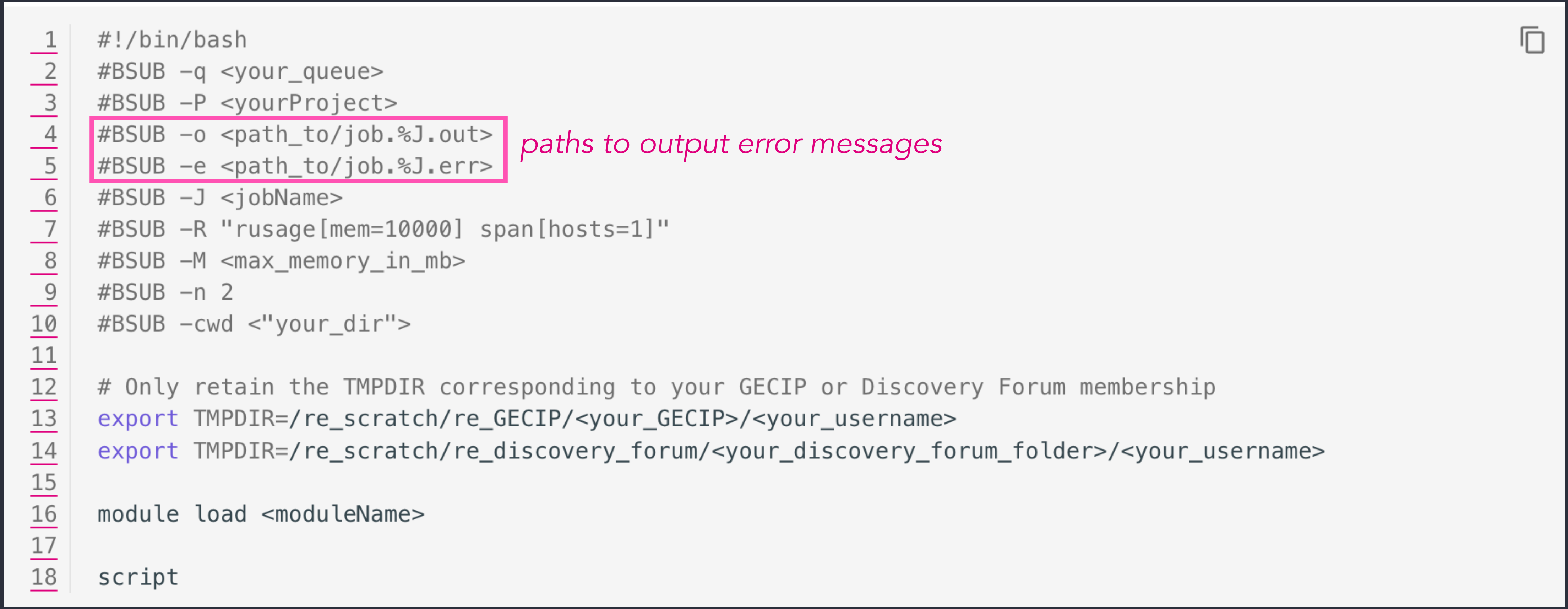
```
#!/bin/bash
#BSUB -q <your_queue>
#BSUB -P <yourProject>
#BSUB -o <path_to/job.%J.out>
#BSUB -e <path_to/job.%J.err>
#BSUB -J <jobName>
#BSUB -R "rusage[mem=10000] span[hosts=1]"
#BSUB -M <max_memory_in_mb>
#BSUB -n 2
#BSUB -cwd <"your_dir">

# Only retain the TMPDIR corresponding to your GECIP or Discovery Forum membership
export TMPDIR=/re_scratch/re_GECIP/<your_GECIP>/<your_username>
export TMPDIR=/re_scratch/re_discovery_forum/<your_discovery_forum_folder>/<your_username>

module load <moduleName>

script
```

re_gecip_XXX / re_df_XXX

bsub < submission_script.sh

Further reading: https://re-docs.genomicsengland.co.uk/hpc_jobs/

Typical LSF submission



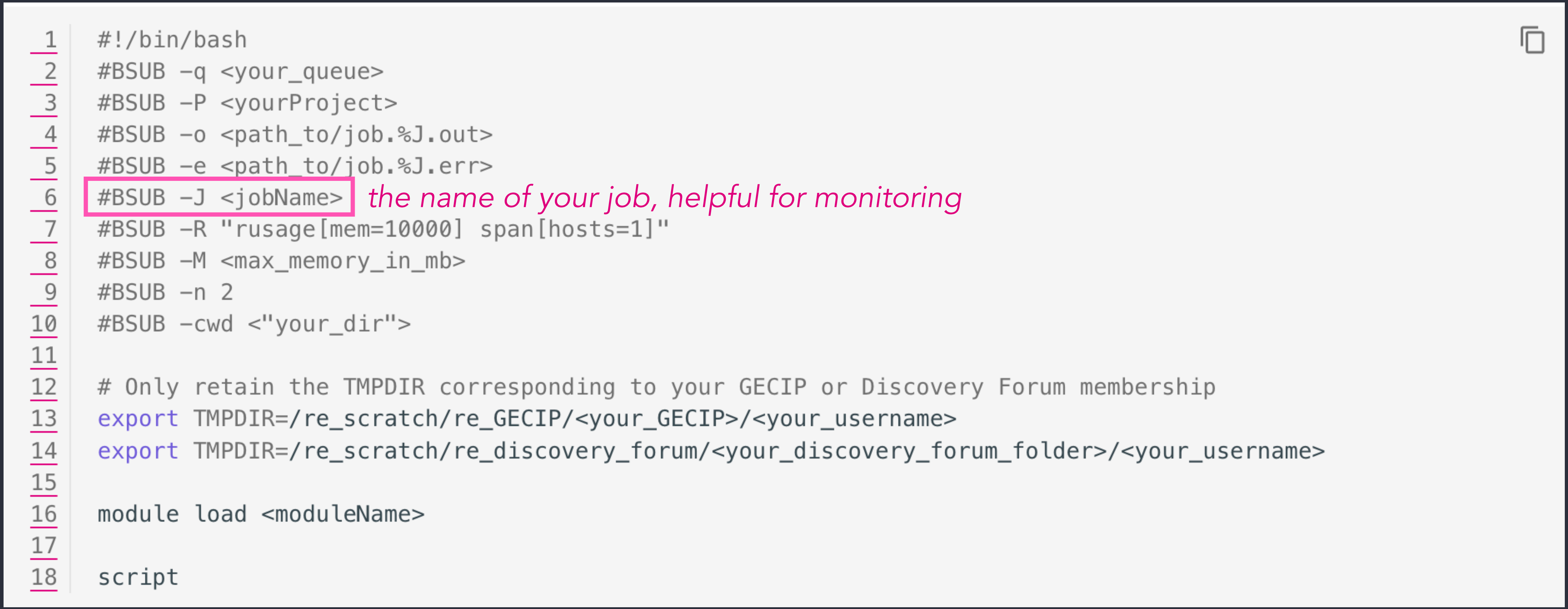
```
#!/bin/bash
#BSUB -q <your_queue>
#BSUB -P <yourProject>
#BSUB -o <path_to/job.%J.out>
#BSUB -e <path_to/job.%J.err>
#BSUB -J <jobName>
#BSUB -R "rusage[mem=10000] span[hosts=1]"
#BSUB -M <max_memory_in_mb>
#BSUB -n 2
#BSUB -cwd <"your_dir">

# Only retain the TMPDIR corresponding to your GECIP or Discovery Forum membership
export TMPDIR=/re_scratch/re_GECIP/<your_GECIP>/<your_username>
export TMPDIR=/re_scratch/re_discovery_forum/<your_discovery_forum_folder>/<your_username>

module load <moduleName>

script
```

paths to output error messages

bsub < submission_script.sh

Further reading: https://re-docs.genomicsengland.co.uk/hpc_jobs/

Typical LSF submission



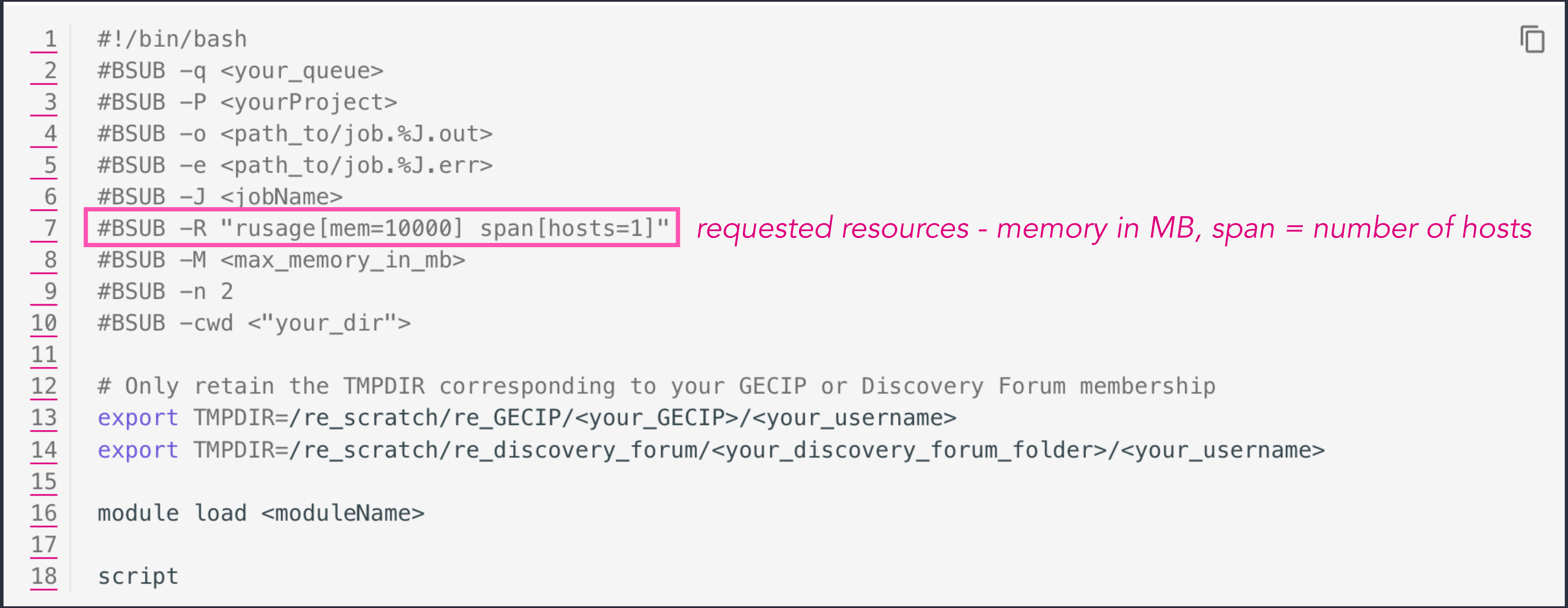
```
#!/bin/bash
#BSUB -q <your_queue>
#BSUB -P <yourProject>
#BSUB -o <path_to/job.%J.out>
#BSUB -e <path_to/job.%J.err>
#BSUB -J <jobName>
#BSUB -R "rusage[mem=10000] span[hosts=1]"
#BSUB -M <max_memory_in_mb>
#BSUB -n 2
#BSUB -cwd <"your_dir">

# Only retain the TMPDIR corresponding to your GECIP or Discovery Forum membership
export TMPDIR=/re_scratch/re_GECIP/<your_GECIP>/<your_username>
export TMPDIR=/re_scratch/re_discovery_forum/<your_discovery_forum_folder>/<your_username>

module load <moduleName>

script
```

the name of your job, helpful for monitoring

bsub < submission_script.sh

Further reading: https://re-docs.genomicsengland.co.uk/hpc_jobs/

Typical LSF submission



```
#!/bin/bash
#BSUB -q <your_queue>
#BSUB -P <yourProject>
#BSUB -o <path_to/job.%J.out>
#BSUB -e <path_to/job.%J.err>
#BSUB -J <jobName>
#BSUB -R "rusage[mem=10000] span[hosts=1]"
#BSUB -M <max_memory_in_mb>
#BSUB -n 2
#BSUB -cwd <"your_dir">

# Only retain the TMPDIR corresponding to your GECIP or Discovery Forum membership
export TMPDIR=/re_scratch/re_GECIP/<your_GECIP>/<your_username>
export TMPDIR=/re_scratch/re_discovery_forum/<your_discovery_forum_folder>/<your_username>

module load <moduleName>

script
```

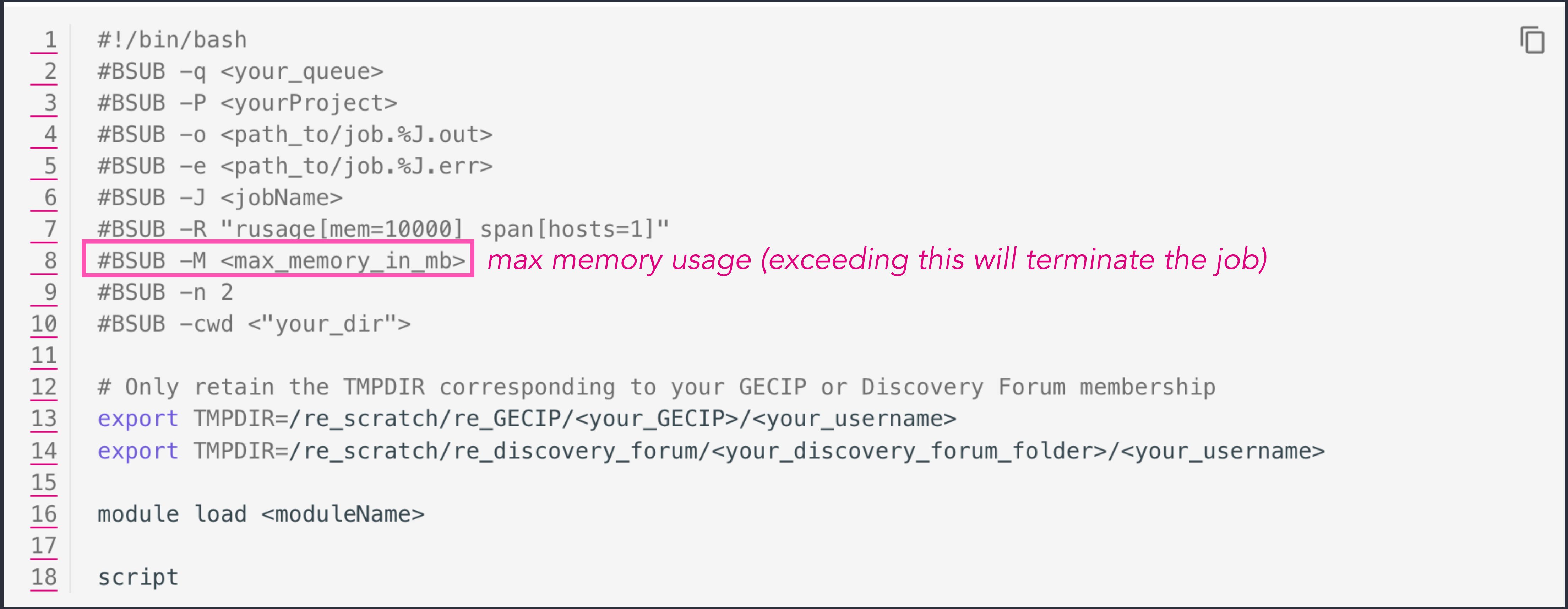
requested resources - memory in MB, span = number of hosts

bsub < submission_script.sh

Further reading: https://re-docs.genomicsengland.co.uk/hpc_jobs/

Genomics England

Typical LSF submission



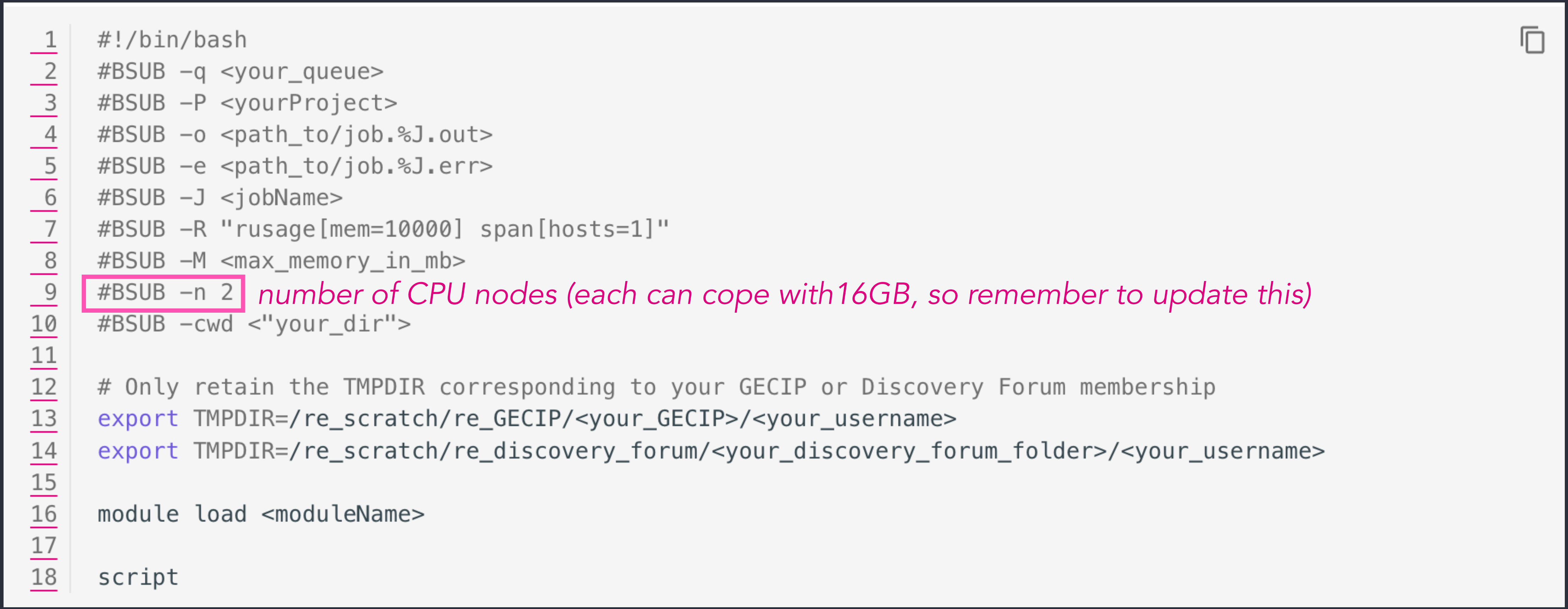
```
#!/bin/bash
#BSUB -q <your_queue>
#BSUB -P <yourProject>
#BSUB -o <path_to/job.%J.out>
#BSUB -e <path_to/job.%J.err>
#BSUB -J <jobName>
#BSUB -R "rusage[mem=10000] span[hosts=1]"
#BSUB -M <max_memory_in_mb>
#BSUB -n 2
#BSUB -cwd <"your_dir">

# Only retain the TMPDIR corresponding to your GECIP or Discovery Forum membership
export TMPDIR=/re_scratch/re_GECIP/<your_GECIP>/<your_username>
export TMPDIR=/re_scratch/re_discovery_forum/<your_discovery_forum_folder>/<your_username>

module load <moduleName>

script
```

max memory usage (exceeding this will terminate the job)

bsub < submission_script.sh

Further reading: https://research-help.genomicsengland.co.uk/display/GERE/4.c+How+to+submit+jobs+to+LSF

Typical LSF submission



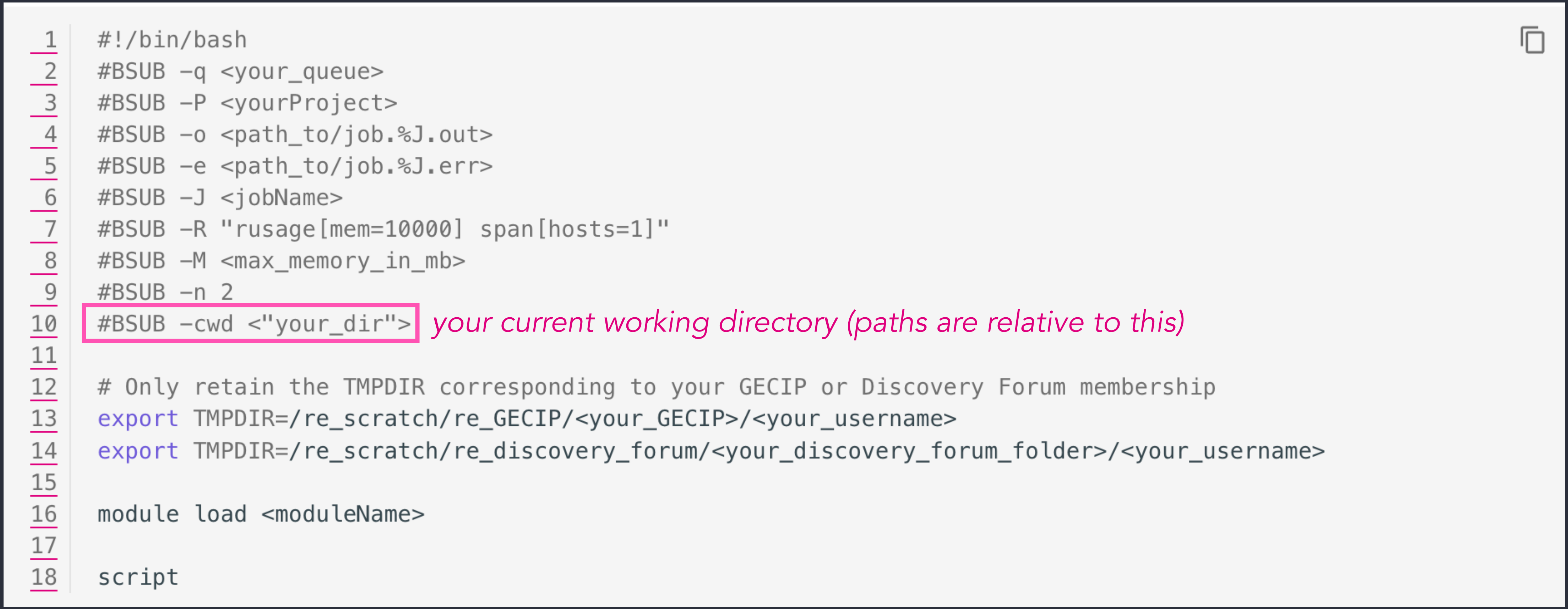
```bash
#!/bin/bash
#BSUB -q <your_queue>
#BSUB -P <yourProject>
#BSUB -o <path_to/job.%J.out>
#BSUB -e <path_to/job.%J.err>
#BSUB -J <jobName>
#BSUB -R "rusage[mem=10000] span[hosts=1]"
#BSUB -M <max_memory_in_mb>
#BSUB -n 2
#BSUB -cwd <"your_dir">

# Only retain the TMPDIR corresponding to your GECIP or Discovery Forum membership
export TMPDIR=/re_scratch/re_GECIP/<your_GECIP>/<your_username>
export TMPDIR=/re_scratch/re_discovery_forum/<your_discovery_forum_folder>/<your_username>

module load <moduleName>

script
```

number of CPU nodes (each can cope with16GB, so remember to update this)

bsub < submission_script.sh

Further reading: https://re-docs.genomicsengland.co.uk/hpc_jobs/

Typical LSF submission



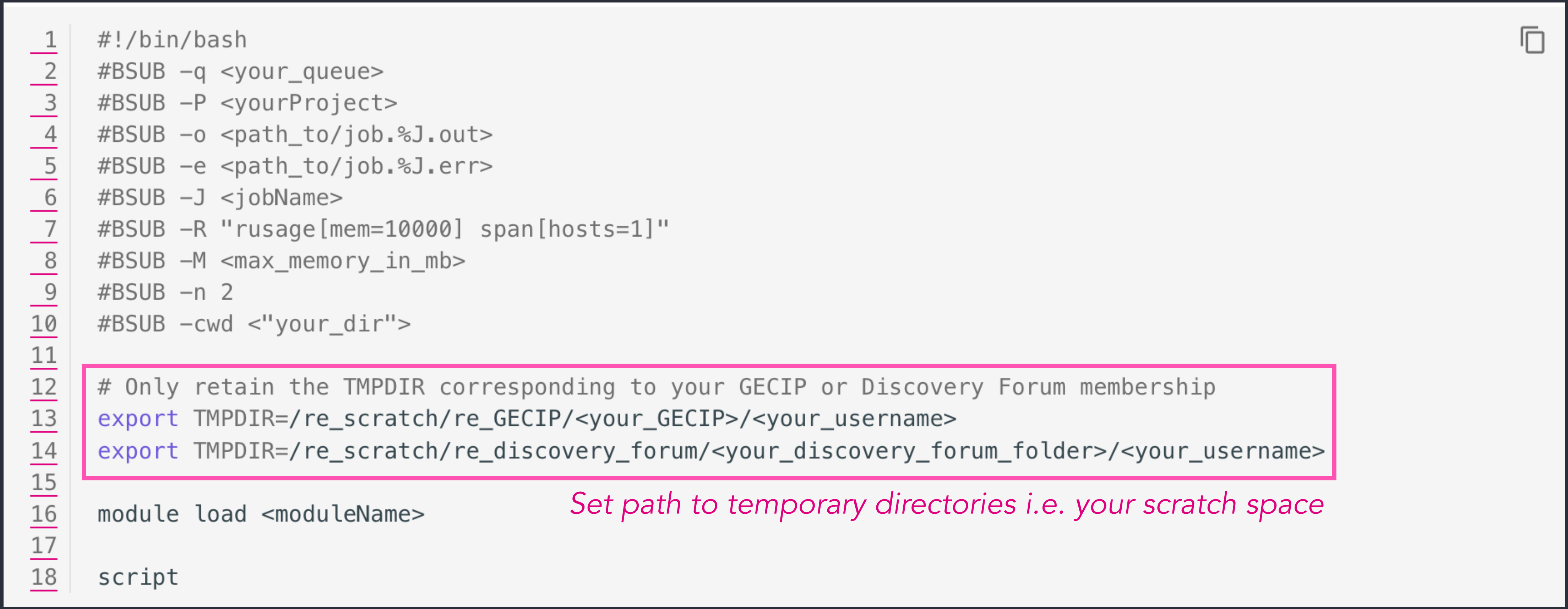
```
#!/bin/bash
#BSUB -q <your_queue>
#BSUB -P <yourProject>
#BSUB -o <path_to/job.%J.out>
#BSUB -e <path_to/job.%J.err>
#BSUB -J <jobName>
#BSUB -R "rusage[mem=10000] span[hosts=1]"
#BSUB -M <max_memory_in_mb>
#BSUB -n 2
#BSUB -cwd <"your_dir">

# Only retain the TMPDIR corresponding to your GECIP or Discovery Forum membership
export TMPDIR=/re_scratch/re_GECIP/<your_GECIP>/<your_username>
export TMPDIR=/re_scratch/re_discovery_forum/<your_discovery_forum_folder>/<your_username>

module load <moduleName>

script
```

your current working directory (paths are relative to this)

bsub < submission_script.sh

Further reading: https://re-docs.genomicsengland.co.uk/hpc_jobs/

Typical LSF submission



```bash
#!/bin/bash
#BSUB -q <your_queue>
#BSUB -P <yourProject>
#BSUB -o <path_to/job.%J.out>
#BSUB -e <path_to/job.%J.err>
#BSUB -J <jobName>
#BSUB -R "rusage[mem=10000] span[hosts=1]"
#BSUB -M <max_memory_in_mb>
#BSUB -n 2
#BSUB -cwd <"your_dir">

# Only retain the TMPDIR corresponding to your GECIP or Discovery Forum membership
export TMPDIR=/re_scratch/re_GECIP/<your_GECIP>/<your_username>
export TMPDIR=/re_scratch/re_discovery_forum/<your_discovery_forum_folder>/<your_username>

module load <moduleName>

script
```

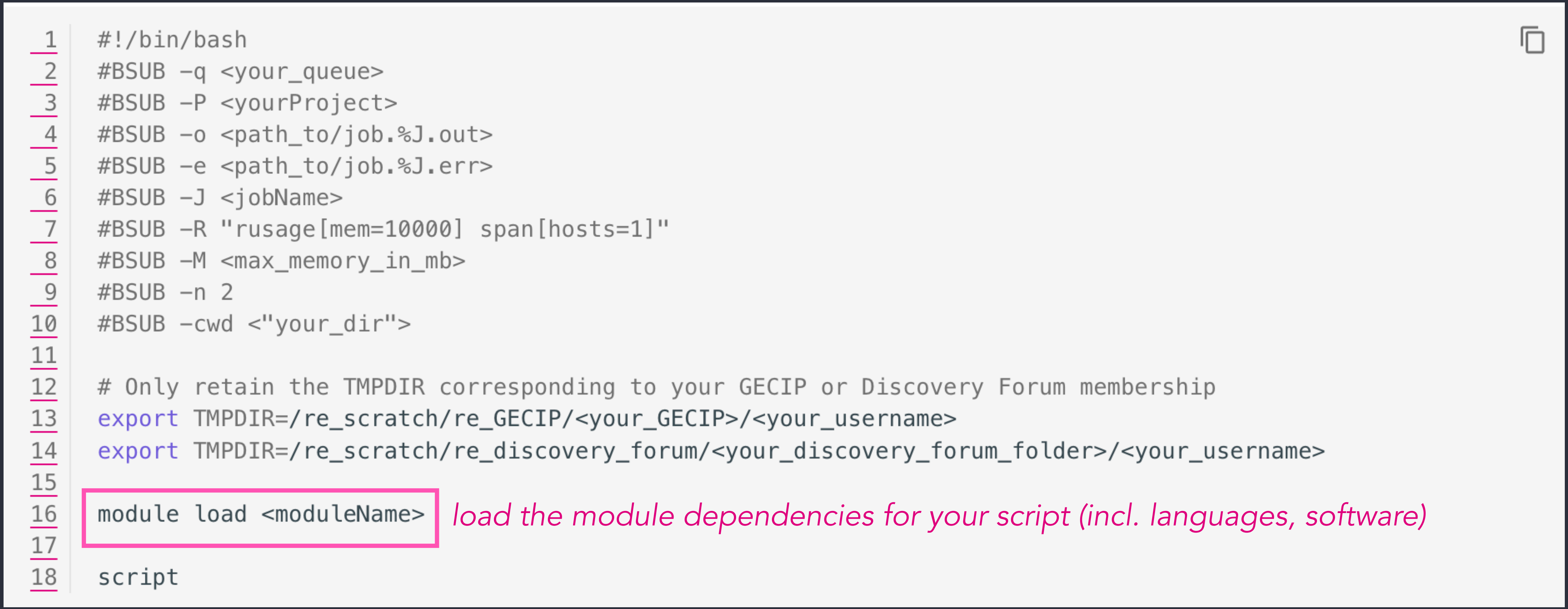
Set path to temporary directories i.e. your scratch space

bsub < submission_script.sh

Further reading: https://re-docs.genomicsengland.co.uk/hpc_jobs/

Genomics England

Typical LSF submission



```
#!/bin/bash
#BSUB -q <your_queue>
#BSUB -P <yourProject>
#BSUB -o <path_to/job.%J.out>
#BSUB -e <path_to/job.%J.err>
#BSUB -J <jobName>
#BSUB -R "rusage[mem=10000] span[hosts=1]"
#BSUB -M <max_memory_in_mb>
#BSUB -n 2
#BSUB -cwd <"your_dir">

# Only retain the TMPDIR corresponding to your GECIP or Discovery Forum membership
export TMPDIR=/re_scratch/re_GECIP/<your_GECIP>/<your_username>
export TMPDIR=/re_scratch/re_discovery_forum/<your_discovery_forum_folder>/<your_username>

module load <moduleName>

script
```

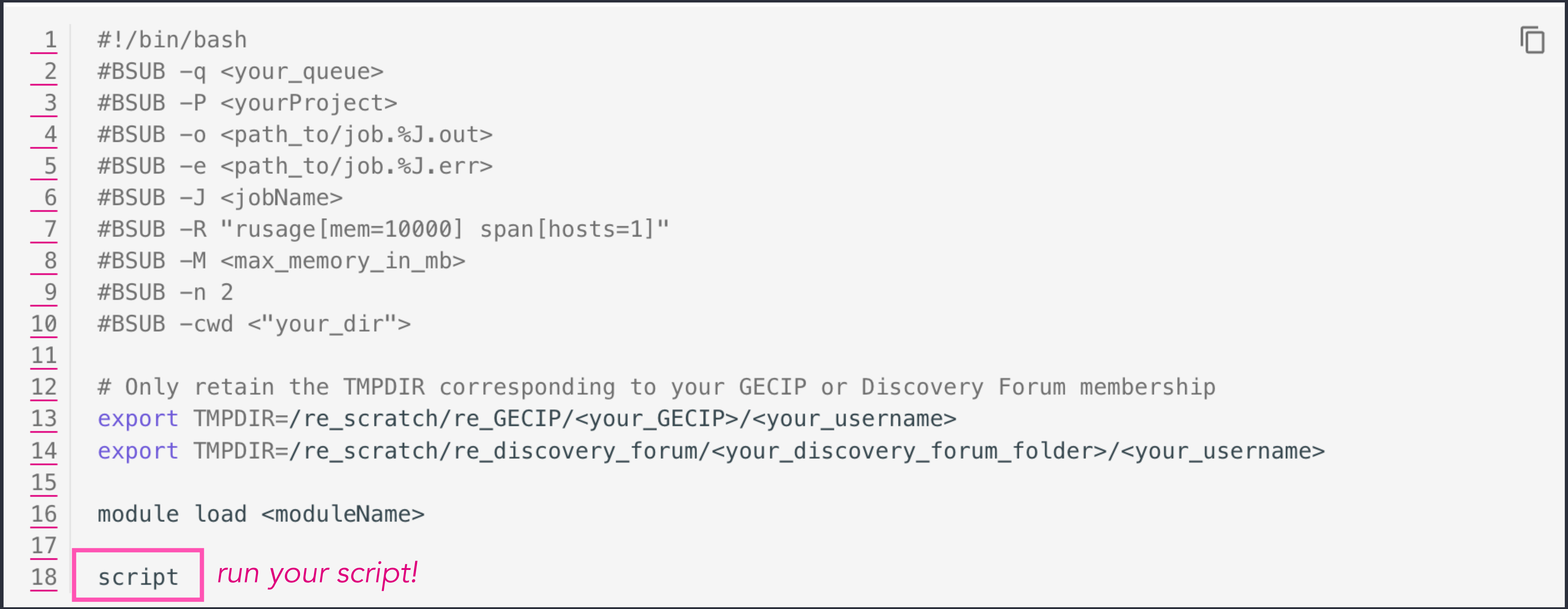
load the module dependencies for your script (incl. languages, software)

bsub < submission_script.sh

Further reading: https://re-docs.genomicsengland.co.uk/hpc_jobs/

Genomics England

Typical LSF submission



```
#!/bin/bash
#BSUB -q <your_queue>
#BSUB -P <yourProject>
#BSUB -o <path_to/job.%J.out>
#BSUB -e <path_to/job.%J.err>
#BSUB -J <jobName>
#BSUB -R "rusage[mem=10000] span[hosts=1]"
#BSUB -M <max_memory_in_mb>
#BSUB -n 2
#BSUB -cwd <"your_dir">

# Only retain the TMPDIR corresponding to your GECIP or Discovery Forum membership
export TMPDIR=/re_scratch/re_GECIP/<your_GECIP>/<your_username>
export TMPDIR=/re_scratch/re_discovery_forum/<your_discovery_forum_folder>/<your_username>

module load <moduleName>

script
```

run your script!

bsub < submission_script.sh

Further reading: https://re-docs.genomicsengland.co.uk/hpc_jobs/

Advice for more advanced jobs

Multicore jobs:

- A parallel job may span multiple hosts, with a number of processes allocated to each host.
- Specify number of CPUs with –n
- Schedule on to a single host to take advantage of its efficient shared memory: span[hosts=1]
- Spread out on to multiple hosts to take advantage of aggregate memory: span[hosts=>1]

Job dependencies:

- Sometimes, whether a job should start depends on the result of another job.
- To submit a job that depends on another job: use the -w option to bsub (it is lowercase w)

Job arrays:

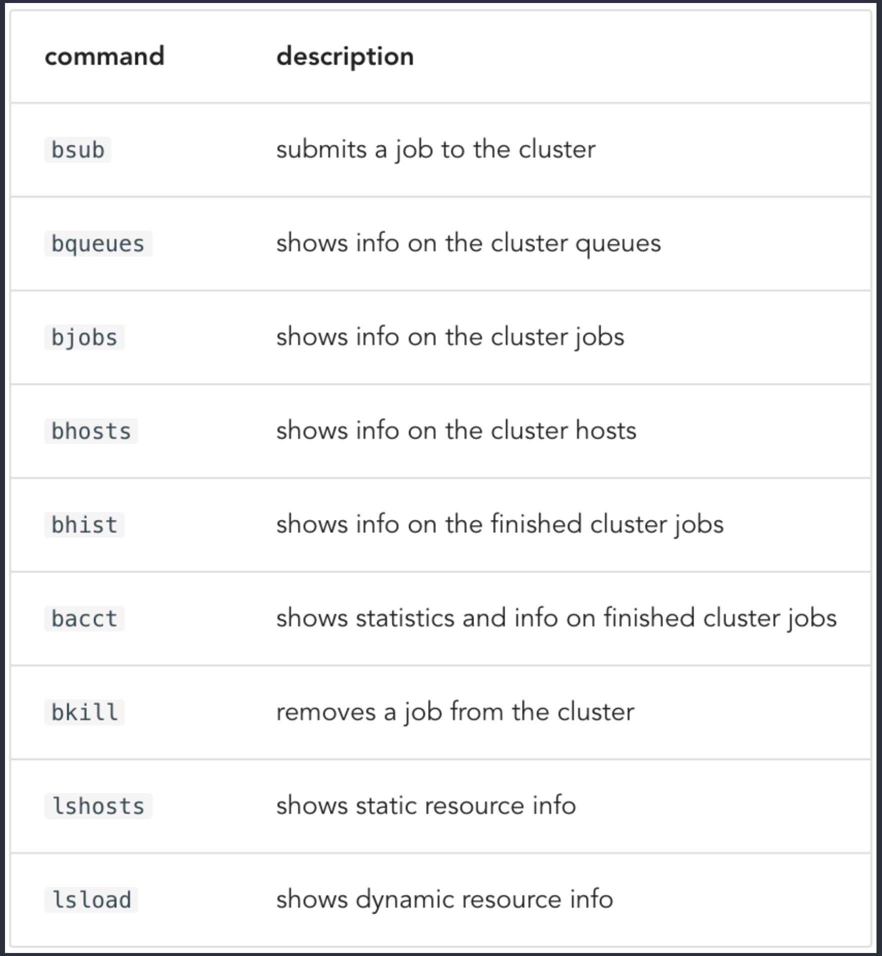
- Allows a sequence of jobs to share the same executable and have different inputs and outputs
- Syntax: bsub –J "ArrayName[index]" –i in.%I myjob

Throttling jobs:

- Control the number of concurrent running jobs, being mindful of other users in the cluster.
- bsub -q medium –J “myArray[1-1000]%50” <rest of the submission>

Further reading: https://re-docs.genomicsengland.co.uk/hpc_guidelines/

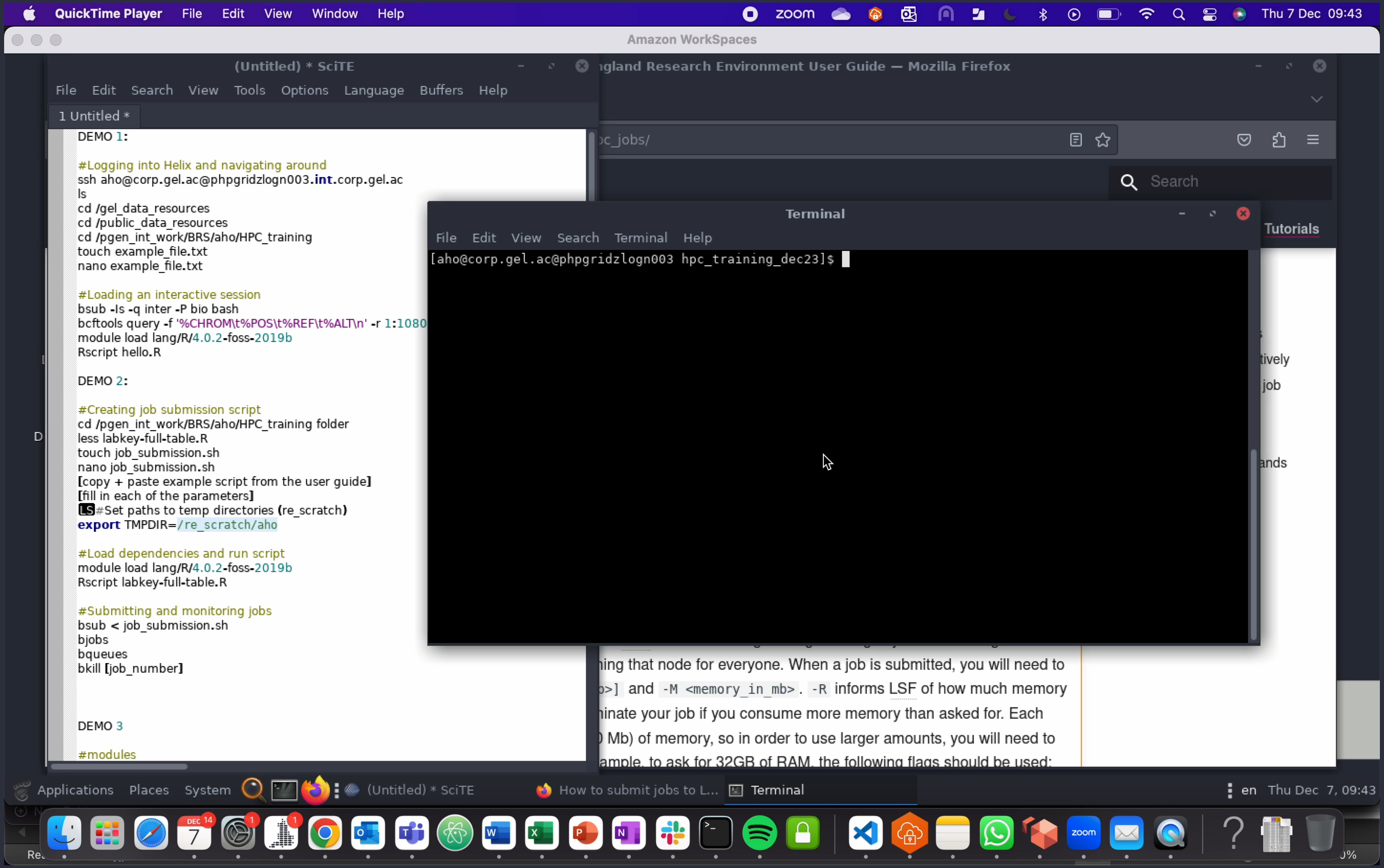
Monitoring your job



command	description
`bsub`	submits a job to the cluster
`bqueues`	shows info on the cluster queues
`bjobs`	shows info on the cluster jobs
`bhosts`	shows info on the cluster hosts
`bhist`	shows info on the finished cluster jobs
`bacct`	shows statistics and info on finished cluster jobs
`bkill`	removes a job from the cluster
`lshosts`	shows static resource info
`lsload`	shows dynamic resource info

Useful link to all the LSF commands:
https://www.ibm.com/docs/en/spectrum-lsf/10.1.0?topic=started-quick-reference

Demo: Submitting a job to LSF



```
DEMO 1:

#Logging into Helix and navigating around
ssh aho@corp.gel.ac@phpgridzlogn003.int.corp.gel.ac
ls
cd /gel_data_resources
cd /public_data_resources
cd /pgen_int_work/BRS/aho/HPC_training
touch example_file.txt
nano example_file.txt

#Loading an interactive session
bsub -Is -q inter -P bio bash
bcftools query -f '%CHROM\t%POS\t%REF\t%ALT\n' -r 1:1080
module load lang/R/4.0.2-foss-2019b
Rscript hello.R

DEMO 2:

#Creating job submission script
cd /pgen_int_work/BRS/aho/HPC_training folder
less labkey-full-table.R
touch job_submission.sh
nano job_submission.sh
[copy + paste example script from the user guide]
[fill in each of the parameters]
LS#Set paths to temp directories (re_scratch)
export TMPDIR=/re_scratch/aho

#Load dependencies and run script
module load lang/R/4.0.2-foss-2019b
Rscript labkey-full-table.R

#Submitting and monitoring jobs
bsub < job_submission.sh
bjobs
bqueues
bkill [job_number]


DEMO 3

#modules
```

```
[aho@corp.gel.ac@phpgridzlogn003 hpc_training_dec23]$ 
```

ing that node for everyone. When a job is submitted, you will need to
>] and `-M <memory_in_mb>`. `-R` informs LSF of how much memory
inate your job if you consume more memory than asked for. Each
) Mb) of memory, so in order to use larger amounts, you will need to
ample, to ask for 32GB of RAM, the following flags should be used:

QuickTime Player File Edit View Window Help
Thu 7 Dec 09:43
Amazon WorkSpaces
(Untitled) * SciTE
File Edit Search View Tools Options Language Buffers Help
1 Untitled *
ngland Research Environment User Guide — Mozilla Firefox
pc_jobs/
Search
Tutorials
Terminal
File Edit View Search Terminal Help
Applications Places System
(Untitled) * SciTE
How to submit jobs to L...
Terminal
en Thu Dec 7, 09:43

6. Tools and software available and how to load them

Modules

- Allows the HPC Helix to have multiple versions of software available
- Dynamic modification of a user's environment via ***module files***
- Chain module load commands to set up your environment for your research
- >1,000 different set of tools and versions available
  - Coding languages (Python, R, Perl)
  - Workflow languages (Cromwell, Nextflow)
  - Command line tools (bcftools, GATK, etc.)

For quick guidance: ***`module help`***

To list available modules: ***`module avail`***

To load a module: ***`module load XXX`***

e.g. ***`module load lang/R/4.0.2-foss-2019b`***

A quick mention on R versions & Libraries

- Several versions of R available (`module avail`): 3.6/4.0/4.1/4.2.1
- Each version has its own range of centrally installed packages
  - Self-service installation
  - Software requests (e.g. packages in GitHub and installed by devtools)
  - >library(<package_name>)
- Rstudio and VScode are installed for script development (more from Ken…)
- Bring in a container through Singularity (more from Ken…)

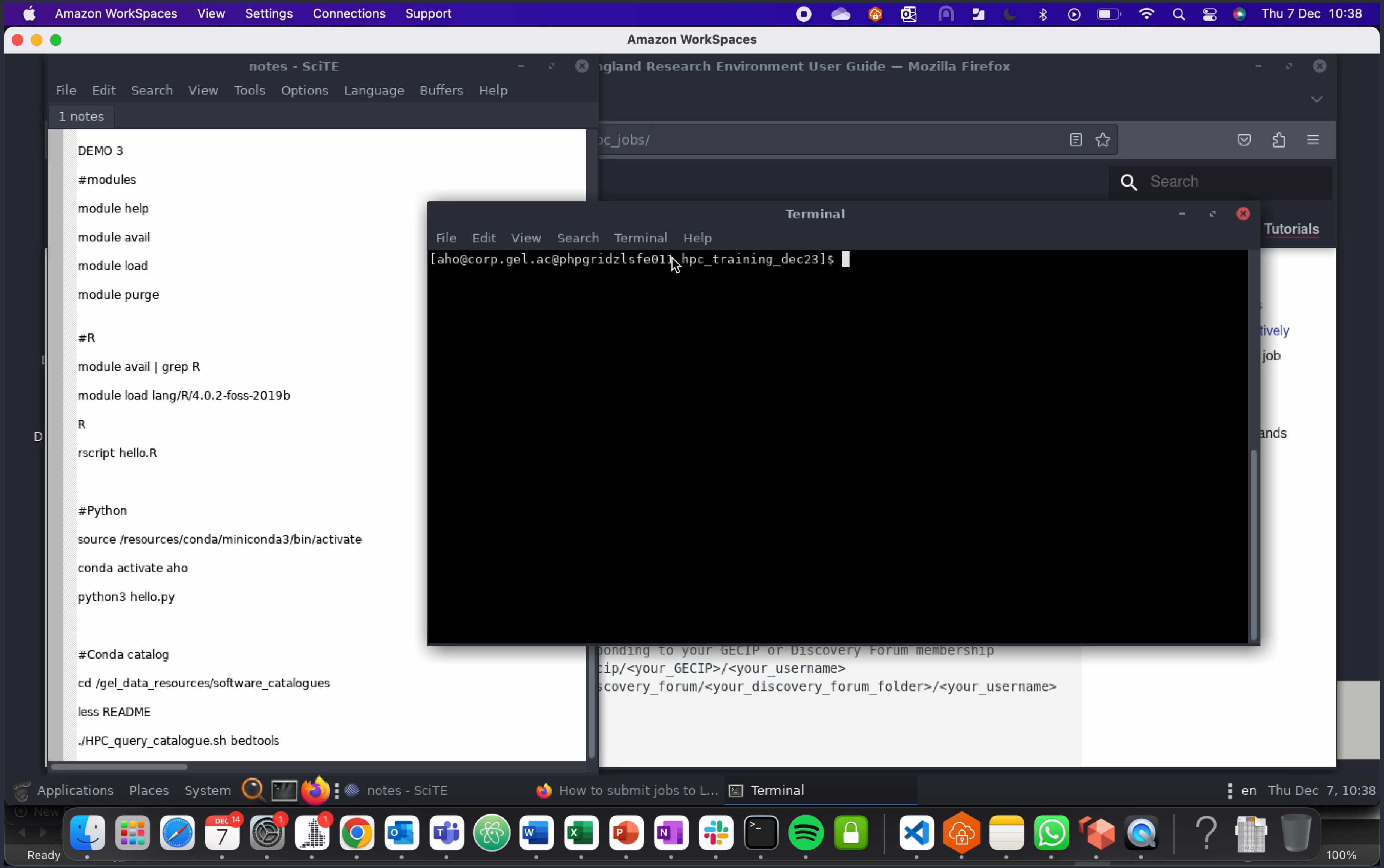
Further reading: https://re-docs.genomicsengland.co.uk/hpc_r/

Python packages & conda environments

- Several versions of python available (module avail): 2.7, 3.7, 3.10
- import <package> // import <package> as <alias>
- Conda* envs: /gel_data_resources/software_catalogues
  - See the README files for usage
- Create your own! (see user guide)
  - Copy .condarc to your working directory
  - `conda create python==<version_number> --prefix /path/to/env/location`
  - `source /resources/conda/miniconda3/bin/activate`
  - `conda activate /path/to/env/location`
  - `conda install -c conda-main <package_1>`
  - `pip install <package_name> --index-url https://artifactory.aws.gel.ac/artifactory/api/pypi/pypi/simple`
- Bring in a container through Singularity (more from Ken…)

Further reading: https://re-docs.genomicsengland.co.uk/python/

Demo: Modules, R and Python



```
DEMO 3

#modules

module help

module avail

module load

module purge

#R

module avail | grep R

module load lang/R/4.0.2-foss-2019b

R

rscript hello.R

#Python

source /resources/conda/miniconda3/bin/activate

conda activate aho

python3 hello.py

#Conda catalog

cd /gel_data_resources/software_catalogues

less README

./HPC_query_catalogue.sh bedtools
```

```
[aho@corp.gel.ac@phpgridzlsfe011 hpc_training_dec23]$
```

```
ponding to your GECIP or Discovery Forum membership
cip/<your_GECIP>/<your_username>
scovery_forum/<your_discovery_forum_folder>/<your_username>
```

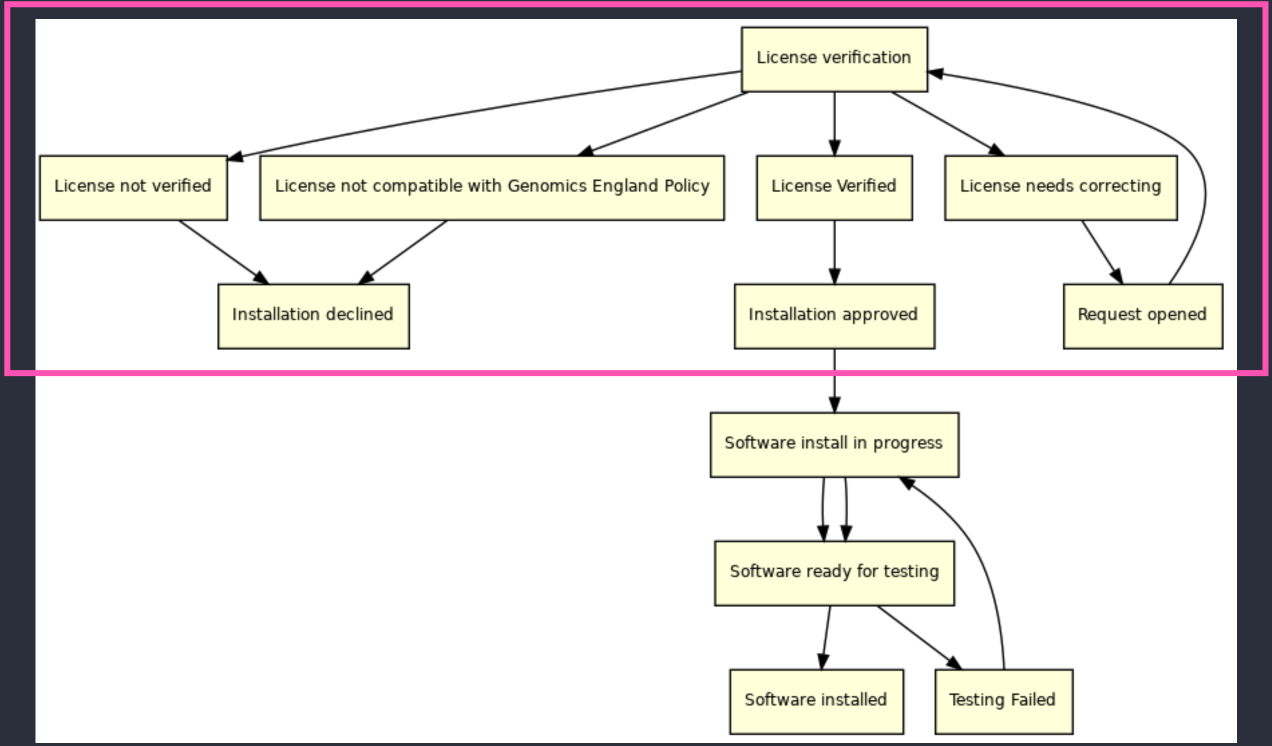
Amazon WorkSpaces View Settings Connections Support
Thu 7 Dec 10:38
notes - SciTE
File Edit Search View Tools Options Language Buffers Help
1 notes
ngland Research Environment User Guide — Mozilla Firefox
c_jobs/
Search
Terminal
File Edit View Search Terminal Help
Tutorials
Applications Places System
How to submit jobs to L...
en Thu Dec 7, 10:38

Genomics England

Software Requests

https://research-help.genomicsengland.co.uk/display/GERE/How+to+request+Software+installation+within+the+Research+Environment

- Software not already available? Self-installation not possible?
- Raise a ticket: https://jiraservicedesk.extge.co.uk/plugins/servlet/desk



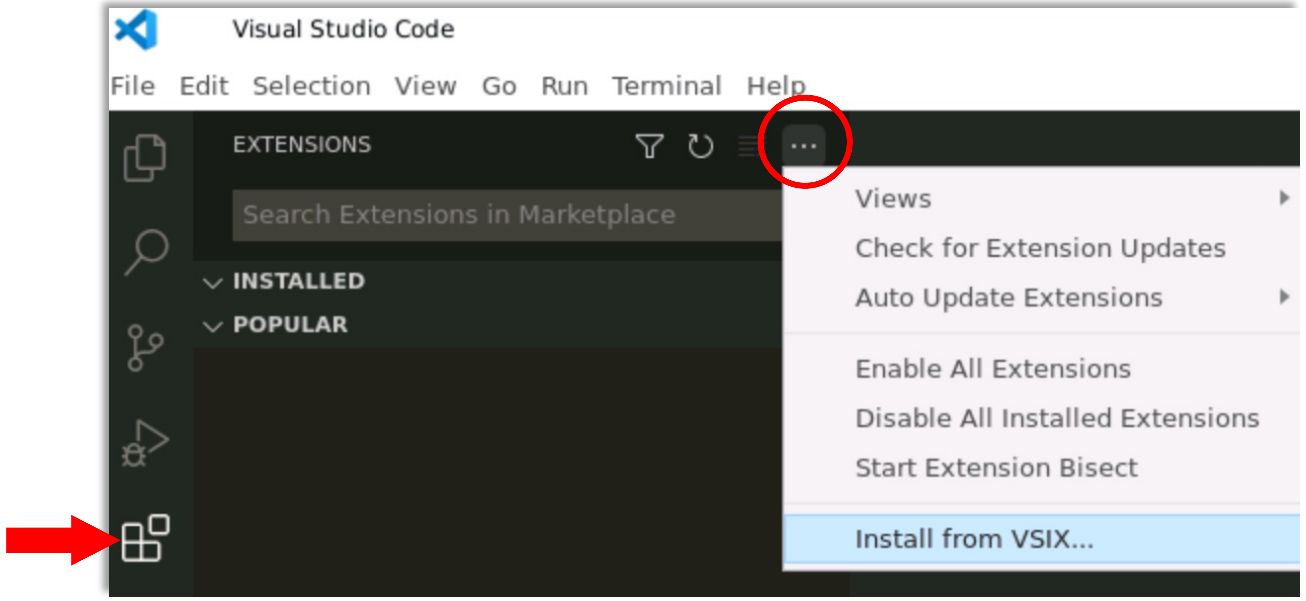
Further reading: https://re-docs.genomicsengland.co.uk/hpc_software_request/

7. Interactive coding tools

Text editors/ IDEs

- several text editors/ IDEs available (VS Code, GVim, Emacs, …)
  - any can be used to write and develop scripts for submission to the HPC

- VS Code has no direct access to VS Code Marketplace
  - curated set of extensions ~/public_data_resources/vscode_extensions/vsix



Visual Studio Code
File Edit Selection View Go Run Terminal Help
EXTENSIONS
Search Extensions in Marketplace
INSTALLED
POPULAR
Views
Check for Extension Updates
Auto Update Extensions
Enable All Extensions
Disable All Installed Extensions
Start Extension Bisect
Install from VSIX...

R configuration (prerequisite for using R)

- copy config files to your home directory (from Terminal)

```
# On the desktop
cp -a ~/gel_data_resources/example_config_files/Inuvika/. ./

# On the Helix/HPC
ssh <username>@corp.gel.ac@phpgridzlogn00<N>.int.corp.gel.ac
cp -a /gel_data_resources/example_config_files/Helix/. ./
```

- copies 3 files: .Renviron, .Rprofile, .netrc
    - edit .netrc login and password

```
machine labkey-embassy.gel.zone
login <username>
password <password>
```

⚠ **set permission to user only access**, (i.e. `-rw------- <username> <username> .netrc`)

```
chmod 600
```

R/RStudio (1)

- **R** log in to Helix, start an interactive session, load an R module

```
ssh <username>@corp.gel.ac@phpgridzlogn00<N>.int.corp.gel.ac
bsub -Is -q inter -P <project_code> bash
module avail lang/R
module load lang/R/4.0.2-foss-2019b
R
```

- **RStudio**
  - open RStudio (double-click desktop icon)
  - open a Terminal in RStudio (`Tools > Terminal > New Terminal` or Alt+Shift+R)
  - log into Helix, start an interactive session, load an R module

```
ssh <username>@corp.gel.ac@phpgridzlogn00<N>.int.corp.gel.ac
bsub -q inter -P <project_code> -Is -n 1 -R rusage[mem=16000] -M 16000 /bin/bash
module avail lang/R
module load lang/R/4.0.2-foss-2019b
R
```

R/RStudio (2)

- open a new script (`File > New File > R Script` or Ctrl+Shift+N), or an existing script
- send code to Terminal

Optional

Change the keyboard shortcut for sending code to the RStudio Terminal – recommended for Mac users
- `Tools > Modify Keyboard Shortcuts...`
- change "**Send Selection to Terminal**" (suggested: Ctrl+Cmd+Enter)

R packages

- make an R package folder

```
cd /path/to/personal_folder
mkdir Rpackages
```

- In R, install and load packages

```
install.packages("broom", lib = "/path/to/personal_folder/R_packages")
library("broom", lib.loc = "/path/to/personal_folder/R_packages")
```

- Optionally, mount library location to `.libPaths()` to load packages without `lib.loc`

```
.libPaths(c( .libPaths(), "/path/to/personal_folder/Rpackages"))
library(broom)
```

- see docs for `BioConductor::install`, `devtools::install_github`, and pre-installed packages
- *any problems raise a service desk ticket*

Jupyter Notebook/ Lab

- login to Helix HPC, start an interactive session

```
ssh <username>@corp.gel.ac@phpgridzlogn00<N>.int.corp.gel.ac
bsub -P <project_code> -M 25G -Is -q inter bash
```

- start Jupyter Lab in the conda environment `2021_base_clone` (note URL*)

```
source /resources/conda/miniconda3/bin/activate
conda activate 2021_base_clone
jupyter lab --no-browser --ip="*" --port=<remote_port>
```

- connect to your Jupyter Lab session (from another terminal)

```
#  Set <host_port> and <remote_port> to the same value. Don't use: 8888, 5000, 8000 or 9000
# NB. this is one line, just wrapped
ssh -4 -L <host_port>:phpgridzlsfe<N>.cluster:<remote_port>
<username>@corp.gel.ac@phpgridzlogn00<N>.int.corp.gel.ac
```

- start a browser, paste URL*

8. Bringing in your own tools and software

Software/ Containers

- *Use **containers** with the **singularity** container engine*
  - only raise a service desk ticket if you can't use a container for your needs

```
module load tools/singularity/3.8.3
singularity --help
```

- 2 whitelisted **container repositories** (we use Artifactory as a pull-through cache):
  - **dockerhub**: docker-remote.artifactory.aws.gel.ac
  - **quay.io**: docker-quay-io.artifactory.aws.gel.ac

- e.g. to use the latest bcftools in the RE, on Helix,

use url prefix *use artifactory* *choose tag/version*

```
# Official image url is quay.io/biocontainers/bcftools

singularity exec docker://docker-quay-io.artifactory.aws.gel.ac/biocontainers/bcftools:1.18--h8b25389_0 \
--bind /some/local/data:/data \
bcftools --version
```

map directories on your host system to directories within your container

read about software licensing (tl;dr, it's your responsibility)

GEL Workflows

We have **workflows** for a few common queries:

Association testing

- **Aggregate Variant Testing (AVT)** performs case-control rare variant association analyses
- **GWAS** performs case-control genome-wide association analysis

Variant screening

- **Small Variant** extracts and annotates variants within query genes
- **Structural Variant** extracts CNVs and SVs within a query regions defined by gene(s) or coordinates
- **Functional Annotation** VEP annotates VCFs

RE User Guide 〉 **Workflows, scripts and containers** https://re-docs.genomicsengland.co.uk/workflows/#workflows
RE User Guide 〉 **Workflows, scripts and containers** https://re-docs.genomicsengland.co.uk/workflows/#containers

GEL Workflows

- to run a workflow, copy the submission script (only) to your project folder

```
cd /re_gecip/me/my_workflows/my_small_variant_analysis
cp /pgen_int_data_resources/workflows/rdp_small_variant/main/submit.sh .
```

- edit the project code, query genes, and sample input

```
#BSUB -P <project_code>

small_variant='/pgen_int_data_resources/workflows/rdp_small_variant/main'

nextflow run "${small_variant}"/main.nf \
    --gene_input "${small_variant}"/input/gene_list.txt \
    --sample_input "${small_variant}"/input/sample_list.txt \
    --use_sample_input false \
    -profile cluster \
    -resume
```

- run the workflow

```
bsub < submit.sh
```

Best practices for using the HPC

DO...

- make use of interactive sessions for lightweight analysis and script development
- estimate the length of your job and choose the most appropriate queue
- consider queue priority/scheduling when requesting resources through LSF
- use singularity containers to import software to improve your analysis
- utilise the GEL workflows if they will help you achieve your research goals
- raise a "software request" only if you are unable to install your own software
- raise a "service request" if we can help you with anything
- use '-o' and '-e' options with job submissions as offers resource usage any output and errors for troubleshooting

DO NOT...

- use the log in nodes to run any analysis
- request excessive resources to run your jobs

9. Getting help and questions

Getting help

Check our documentation:

- https://re-docs.genomicsengland.co.uk/
- Click on the documentation icon in the environment

Contact our Service Desk:

- https://jiraservicedesk.extge.co.uk/plugins/servlet/desk

Questions

Your microphones are all muted

Use the Zoom Q&A to ask questions

Upvote your favourite questions: if we are short on time we will prioritise those with the most votes

Future sessions

9th Jan	Using the Research Environment for clinical diagnostic discovery
13th Feb	Importing data and tools to use in the RE
12th Mar	Building cancer cohorts and survival analysis
9th Apr	Introduction to the RE
14th May	Building rare disease cohorts with matching controls
11th Jun	Finding participants based on genotypes
9th Jul	Getting medical records for participants
10th Sep	Using GEL data for publications and reports
8th Oct	What tools and workflows should I use to fulfil an overall goal?
12th Nov	Running workflows on the HPC and Cloud
10th Dec	Introduction to the RE

Past training

In the User Guide:
- Redacted slides and videos
- Q&A from sessions

Sessions:
- What tools and workflows should I use to fulfil an overall goal?, November 2023
- Using GEL data for publications and reports, October 2023
- Getting medical history for participants, August 2023
- Finding participants based on genotypes, July 2023
- Building rare disease cohorts with matching controls, June 2023
- New datasets in the RE, May 2023
- Importing tools and data to use in the Research Environment, March 2023
- Using the GEL Research Environment for clinical genetic diagnosis, February 2023
- Introduction the Research Environment, January 2023
- Using the HPC to run jobs, November 2022
- Getting medical history for participants, September 2022
- Finding participants based on genotypes, July 2022
- Building a cohort based on phenotypes, May 2022
- Introduction the Research Environment, March 2022

Feedback

Thank you