

# MG\_HW4: Co-assembly using Megahit version 2

James Thornton

## Abstract

This protocol describes the procedure for performing a co-assembly of short reads to obtain contigs using the Megahit assembler. This procedure is performed on a node at the UoA HPC due to memory considerations.

**Citation:** James Thornton MG\_HW4: Co-assembly using Megahit. **protocols.io**

dx.doi.org/10.17504/protocols.io.fxwbppe

**Published:** 29 Sep 2016

## Guidelines

[UoA HPC: Using the Systems](#)

[Megahit github](#)

## Protocol

### Step 1.

Login to the HPC and move into Cluster(ICE).

```
cmd COMMAND
$ ssh hpc
$ ice
```

#### ■ ANNOTATIONS

**Bonnie Hurwitz** 30 Sep 2016

If you haven't disabled your menu, you will need to select "(3) Cluster/HTC/SMP"

### Step 2.

Move into your bh\_class directory.

```
cmd COMMAND
$ cd /rsgroups/bh_class/username
Use YOUR username to go to YOUR directory
```

### Step 3.

Assembly must be run on a node at UoAs HPC due to the high memory requirements of the job. Copy the below script into a new file called run-assembly.sh :

cmd **COMMAND**

```
#!/bin/bash

#PBS -W group_list=bh_class
#PBS -q windfall
#PBS -l select=1:ncpus=12:mem=23gb
#PBS -l pvmem=22gb
#PBS -l walltime=24:00:00
#PBS -l cput=24:00:00
#PBS -M netid@email.arizona.edu
#PBS -m bea

echo "my job_id is: ${PBS_JOBID}"

FASTA_DIR='/rsgrps/bh_class/username/fastq'
ASSEM_DIR='/rsgrps/bh_class/username/assembly/megahit-out'

cd $FASTA_DIR

FASTA=$(ls ./*.fasta | python -
c 'import sys; print ",".join([x.strip() for x in sys.stdin.readlines()])')

cd $ASSEM_DIR

megahit -r $FASTA --min-contig-len 1000 -t 12 -o $ASSEM_DIR
Make sure to replace netid and username. (username appears twice in this script) #PBS -l
select=1:ncpus=12:mem=23gb is the memory allocations for the job. 1 node, 12 CPUs, and 23gb
of RAM. FASTA=$(ls ./*.fasta) will find all files with the extension .fasta in your FASTA_DIR. and is
piped into the python command to join then on commas.
```

## ■ ANNOTATIONS

**Bonnie Hurwitz** 30 Sep 2016

don't forget to change the permissions so you can run this program:

```
$ chmod 755 run-assembly.sh
```

**Bonnie Hurwitz** 30 Sep 2016

You can also increase the number of cpu-threads to 12, because you are running on a machine with 12 cores.

```
-t 12
```

## Step 4.

Make a std-err and std-out directory.

```
cmd COMMAND
$ mkdir std-err
$ mkdir std-out
```

### Step 5.

Submit run-assembly.sh using qsub:

```
cmd COMMAND
$ qsub -e std-err/ -o std-out/ run-assembly.sh
```

Execute this command in your /rsgroups/bh\_class/username/assembly directory which is the same place where the run-assembly.sh script should be -e and -o declare where to print stdout and stderr.

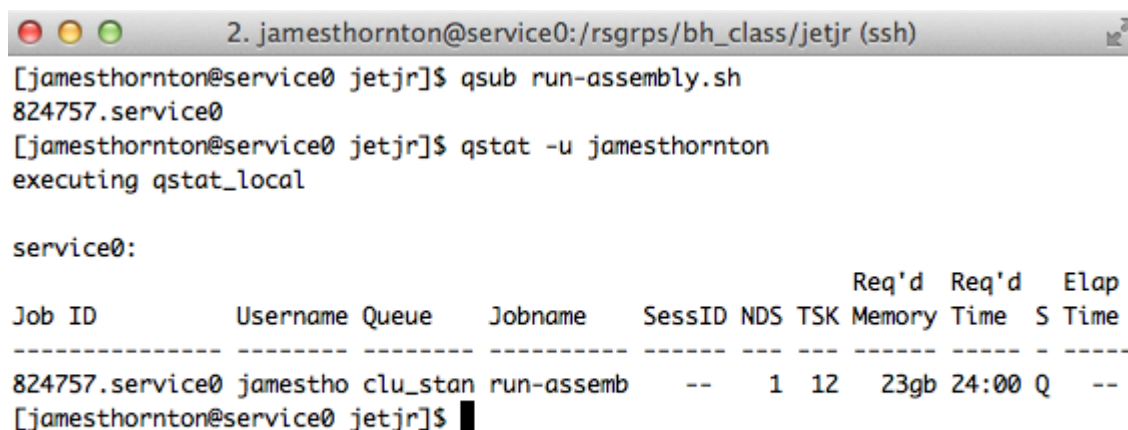
### Step 6.

Check the status of your job:

```
cmd COMMAND
$ qstat -u jamesthornton
```

use your netid username

#### EXPECTED RESULTS



```
2. jamesthornton@service0:/rsgroups/bh_class/jetjr (ssh)
[jamesthornton@service0 jetjr]$ qsub run-assembly.sh
824757.service0
[jamesthornton@service0 jetjr]$ qstat -u jamesthornton
executing qstat_local

service0:
```

Job ID	Username	Queue	Jobname	SessID	NDS	TSK	Req'd Memory	Req'd Time	Elap S	Time
824757.service0	jamesth	clu_stan	run-assembly	--	1	12	23gb	24:00	Q	--

```
[jamesthornton@service0 jetjr]$
```

### Step 7.

The status of the job will go from a 'Q' to 'R' when it is running. Once complete the list will be empty. You should receive email notifications once the job begins running and is complete.

### Step 8.

Once the job is complete move into the assembly directory and check its contents.

```
cmd COMMAND
$ cd /rsgroups/bh_class/username/assembly
$ ls
```

#### ANNOTATIONS

**Bonnie Hurwitz** 30 Sep 2016

you should also move the shell script there for future reference:

```
$ mv run-assembly.sh ./assembly
```

### Step 9.

Rename the final.contigs.fa to contigs.fa :

```
cmd COMMAND  
$ mv final.contigs.fa ./contigs.fa
```

### Step 10.

Check the log file and report number of contigs, min/max length, and N50 in your google doc.

```
cmd COMMAND  
$ tail log  
tail can be used since the information you need is at the bottom of the log file.
```

### ■ ANNOTATIONS

**Bonnie Hurwitz** 30 Sep 2016

Be sure to describe how you did the assembly. What program was used?