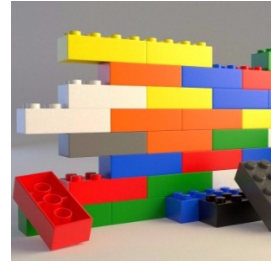


Oct 11, 2017 Version 1

Assembly with Megahit V.1

DOI

dx.doi.org/10.17504/protocols.io.j8tcrwn



James E Thornton Jr¹

¹Hurwitz Lab

Metafunc Course 2017



James E Thornton Jr

Hurwitz Lab

OPEN  ACCESS



DOI: dx.doi.org/10.17504/protocols.io.j8tcrwn

Protocol Citation: James E Thornton Jr 2017. Assembly with Megahit. **protocols.io**
<https://dx.doi.org/10.17504/protocols.io.j8tcrwn>

License: This is an open access protocol distributed under the terms of the **Creative Commons Attribution License**, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited

Protocol status: Working

Created: October 10, 2017

Last Modified: March 28, 2018

Protocol Integer ID: 8179

Abstract

Co-assembly using Megahit.



- 1 Log into the HPC.

Command

```
$ ssh hpc  
$ ocelote
```

- 2 From your home directory, open .bashrc file for editing.

Command

```
$ nano .bashrc
```

Note

Remember, you are already in your home directory after logging into ocelote.

- 3 Input the following line into your .bashrc file:

Command

This will allow us to execute tools found in /rsgroups/bh_class/bin without specifying the path name.

```
export PATH=/rsgroups/bh_class/bin:$PATH
```

- 4 Save and close the .bashrc file



- 5 Move into your project directory.

Command

```
$ cd /rsgrps/bh_class/username
```

- 6 Create a directory for assembly output. Then move into that directory.

Command

```
$ mkdir assembly  
$ cd !
```

- 7 Make directories for standard out and standard error.

Command

```
mkdir std-out std-err
```

- 8 Before we continue, determine if you have single end or paired end files. If you have two files per SRR number, you have paired end reads. Otherwise, you have single end reads.

1. If you have single end reads proceed to step 5.
2. If you have paired end reads, skip to step 6.

- 9 Assembly script for SINGLE END FILES

Create a script called run-assembly.sh

Command

```
#!/bin/bash
#PBS -W group_list=bh_class
#PBS -q windfall
#PBS -l select=1:ncpus=20:mem=40gb
#PBS -l pvmem=38gb
#PBS -l walltime=24:00:00
#PBS -l cput=48:00:00
#PBS -M netid@email.arizona.edu
#PBS -m bea
FASTQ_DIR='/rsgrps/bh_class/username/fastq'
ASSEM_DIR='/rsgrps/bh_class/username/assembly'
MIN_CONTIG_LEN=500
OUT_DIR='/rsgrps/bh_class/username/assembly/megahit-out'
cd $ASSEM_DIR
SINGLES=`ls $FASTQ_DIR/*.fastq | python -c 'import sys; print`
```

Note

OUT_DIR does NOT need to be created prior to running this script. Megahit will make the directory on its own.

10 Assembly script for PAIRED END FILES

Create a script called run-assembly.sh

**Command**

```
#!/bin/bash
#PBS -W group_list=bh_class
#PBS -q windfall
#PBS -l select=1:ncpus=20:mem=40gb
#PBS -l pvmem=38gb
#PBS -l walltime=24:00:00
#PBS -l cput=48:00:00
#PBS -M netid@email.arizona.edu
#PBS -m bea
FASTQ_DIR='/rsgrps/bh_class/username/fastq'
ASSEM_DIR='/rsgrps/bh_class/username/assembly'
MIN_CONTIG_LEN=500
OUT_DIR='/rsgrps/bh_class/username/assembly/megahit-out'
cd $ASSEM_DIR
R1s=`ls $FASTQ_DIR/*_1.fastq | python -c 'import sys; print
```

Note

OUT_DIR does NOT need to be created prior to running this script. Megahit will make the directory on its own.

- 11 Run the assembly:

Command

```
$ chmod +x run-assembly.sh
$ qsub -e std-err/ -o std-out/ run-assembly.sh
```

- 12 You can check the status of your job with the following command:



Command

```
$ qstat -u username
```

Note

Job runtime will vary depending on the size of your dataset.

- 13 Upon job completion, inspect the log file in the megahit-out folder to determine the quality of the assembly including: N50, number of total contigs, maximum/minimum lengths.