

# Segmenting Large Electron Microscopic Image Volumes

An Introduction to NBCR image analysis and segmentation tools



National Biomedical Computation Resource Summer Training Program @ UC San Diego

Chris Churas & Matthias Haberl

August 7<sup>th</sup>, 2017

[churas@ncmir.ucsd.edu](mailto:churas@ncmir.ucsd.edu) & [mhaberl@ucsd.edu](mailto:mhaberl@ucsd.edu)

## Hands-On Session 3

Running CHM Train via chmutil on Rocce Cluster

# Goals

By the end of this session, you will be able to:

1. Create and run CHM Job
2. Download data from Rocce

## 1. Upload data to Rocce

- A. Download the *download\_dataset\_3.zip* file from [https://github.com/CRBS/nbcrtrainingvm/blob/master/download\\_dataset\\_3.zip?raw=true](https://github.com/CRBS/nbcrtrainingvm/blob/master/download_dataset_3.zip?raw=true) and unzip and in a terminal enter it as before.
- B. The directory will contain two items:
  - i. A directory, *chmimages*, which contains EM data that we will be attempting to segment with CHM
  - ii. A directory, *model*, which contains a CHM trained model generated by following steps in previous hands on session 2.
- C. Upload the *chmimages* and *model* directory to Rocce using the commands below:

```
scp -i ~/.ssh/id_rsa -r chmimages <USER>@rocce.ucsd.edu:/data/scratch/<USER>/.
```

```
scp -i ~/.ssh/id_rsa -r model <USER>@rocce.ucsd.edu:/data/scratch/<USER>/.
```

## Directory Structure:

download\_dataset\_3

chmimages

model

```

~/download_dataset_3
Example Output
smithy@smith ~/download_dataset_3
$ scp -i ~/.ssh/id_rsa -r chmimages churastest@rocce.ucsd.edu:/data/scratch/churastest/
Could not create directory '/home/smithy/.ssh'.
The authenticity of host 'rocce.ucsd.edu (169.228.41.199)' can't be established.
RSA key fingerprint is 40:e0:56:fb:2b:38:8d:99:e9:4f:3e:7c:b8:a9:f3:e6.
Are you sure you want to continue connecting (yes/no)? yes
Failed to add the host to the list of known hosts (/home/smithy/.ssh/known_hosts).
Enter passphrase for key '/cygdrive/c/Users/smithy/.ssh/id_rsa':
.DS_Store 100% 6148 6.0KB/s 00:00
image.000.png 100% 132KB 132.4KB/s 00:00
image.001.png 100% 155KB 154.6KB/s 00:00
image.002.png 100% 156KB 156.3KB/s 00:00

smithy@smith ~/download_dataset_3
$ scp -i ~/.ssh/id_rsa -r model churastest@rocce.ucsd.edu:/data/scratch/churastest/
Could not create directory '/home/smithy/.ssh'.
The authenticity of host 'rocce.ucsd.edu (169.228.41.199)' can't be established.
RSA key fingerprint is 40:e0:56:fb:2b:38:8d:99:e9:4f:3e:7c:b8:a9:f3:e6.
Are you sure you want to continue connecting (yes/no)? yes
Failed to add the host to the list of known hosts (/home/smithy/.ssh/known_hosts).
Enter passphrase for key '/cygdrive/c/Users/smithy/.ssh/id_rsa':
MODEL_level0_stage1.mat 100% 361KB 360.9KB/s 00:00
MODEL_level0_stage2.mat 100% 438KB 438.0KB/s 00:00
MODEL_level1_stage1.mat 100% 2321KB 2.3MB/s 00:00
x.000.mat 100% 882KB 882.0KB/s 00:01
x.001.mat 100% 925KB 925.3KB/s 00:00
x.000.mat 100% 889KB 888.7KB/s 00:00
x.001.mat 100% 916KB 916.0KB/s 00:00
x.000.mat 100% 454KB 454.1KB/s 00:00
x.001.mat 100% 459KB 458.7KB/s 00:01
param.mat 100% 5088 5.0KB/s 00:00

smithy@smith ~/download_dataset_3
$

```

## 2. Create CHM job

We will be running the `chmutil` command `createchmjob.py` to create [CHM](#) job on the Rocce cluster. Here is a description of the arguments used in the command invocation below

`--chmbin` Path to [CHM Singularity](#) binary (use path as shown below)  
`--cluster` Tells tool what cluster is being used. (currently Rocce & Comet are supported)  
`--disablechmhisteq` Disables histogram equalization of images by CHM

- Connect to Rocce, change to `/data/scratch/<USER>` directory, and run `createchmjob.py` with arguments as shown below

```
ssh -i ~/.ssh/id_rsa <USER>@rocce.ucsd.edu
cd /data/scratch/<USER>
```

```
createchmjob.py chmimages model mychm --disablechmhisteq --cluster rocce --chmbin /data/churas/chm_s22.img
```

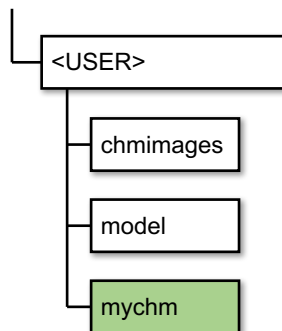
The above command will create a new directory named `mychm`.

If your train job from previous the hands on ran successfully then you can use it instead by replacing the directory path `model` with `mytrain/model`. The command will also output a command to run

For more information on how to use `createchmjob.py` just run the command with no arguments or with `-h` flag (Example: `createchmjob.py -help`)

<https://github.com/CRBS/chmutil/wiki/createchmjob.py>

## Directory Structure:



Note: If a mistake is made, you can use `rm -r` command to recursively remove the directory. The `mychm` directory and re run `createchmjob.py`

**NOTE:** `rm -r` has no undo & is dangerous, be careful

Ex: `rm -r mychm`

Type `man rm` for more information

```

churastest@login-0-0:/data/scratch/churastest

smithy@smith ~/download_dataset_3
$ ssh -i ~/.ssh/id_rsa churastest@rocce.ucsd.edu
Could not create directory '/home/smithy/.ssh'.
The authenticity of host 'rocce.ucsd.edu (169.228.41.199)' can't be established.
RSA key fingerprint is 40:e0:56:fb:2b:38:8d:99:e9:4f:3e:7c:b8:a9:f3:e6.
Are you sure you want to continue connecting (yes/no)? yes
Failed to add the host to the list of known hosts (/home/smithy/.ssh/known_hosts).
Enter passphrase for key '/cygdrive/c/Users/smithy/.ssh/id_rsa':
Last login: Mon Jul 24 16:45:40 2017 from coleslaw.camera.calit2.net
Rocks 6.2 (Sidewinder)
Profile built 10:47 03-Nov-2016
Kickstarted 11:05 03-Nov-2016

-----
Each user has 10Gb space in $HOME for no-volatile files and 100Gb
/data/scratch-your-account-name> for large input/output job files.
All jobs input/output must be in /data/*.
Additional disk space for large input/output jobs files can be available upon
request.

See example SGE submit scripts:
files ending with ".sub" in /share/apps/info directory

Information about using modules is /share/apps/info/modules

Please send requests and questions about this cluster to nbc-admin@ucsd.edu

-----
[churastest@login-0-0 ~]$ cd /data/scratch/churastest/
[churastest@login-0-0 churastest]$ createchmjob.py chmimages model mychm --disab
lechmhisteq --cluster rocce --chmbin /data/churas/chm_s22.img
Run this to submit job
/home/churastest/miniconda2/bin/checkchmjob.py "/data/scratch/churastest/mychm"
--submit
[churastest@login-0-0 churastest]$ |
  
```

## 3a. Processing loop

`createchmjob.py` will output a command to run. This starts the processing loop which comprises of alternating invocations of `checkchmjob.py` & `qsub`, for job submission, and `qstat` to see if jobs completed.

For more information visit: <https://github.com/CRBS/chmutil/wiki/createchmjob.py>

- Lets start the processing loop by running the command output by `createchmjob.py` from the previous step which will look like the following:

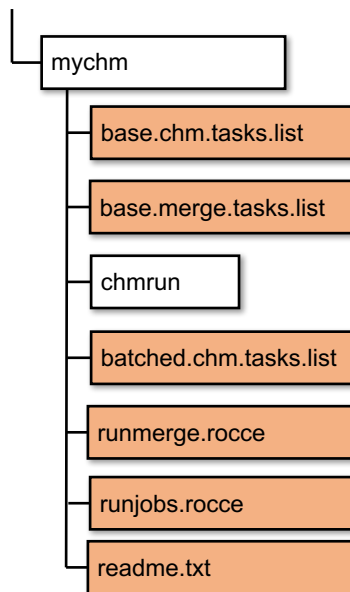
```
/home/<USER>/miniconda2/bin/checkchmjob.py "/data/scratch/<USER>/mychm" --submit
```

`checkchmjob.py` outputs in human readable form a current completion status of the job

If `--submit` is passed the script will create/update `batched.chm.tasks.list` and `batched.merge.chm.tasks.list` files (described later) and output a `qsub` command to run the jobs via OGE/SGE

**NOTE:** The `--submit` flag should be omitted if any jobs are running (job is visible in `qstat`) otherwise errors will occur.

#### Directory Structure:



```
churastest@login-0-0:/data/scratch/churastest
[churastest@login-0-0 churastest]$ createchmjob.py chmimages model mychm --disab
1echmhisteq --cluster rocce --chmbin /data/churas/chm_s22.img
Run this to submit job
/home/churastest/miniconda2/bin/checkchmjob.py "/data/scratch/churastest/mychm"
--submit
[churastest@login-0-0 churastest]$ /home/churastest/miniconda2/bin/checkchmjob.p
y "/data/scratch/churastest/mychm" --submit

Analyzing job. This may take a minute...

chmutil version: 0.8.0
Tiles: 512x512 with 0x0 overlap
Disable histogram equalization in CHM: True
Tasks: 50 tiles per task, 1 task(s) per node
Trained CHM model: /data/scratch/churastest/model
CHM binary: /data/churas/chm_s22.img

CHM tasks: 0% complete (0 of 3 completed)
Merge tasks: 0% complete (0 of 3 completed)

Run this:

  cd "/data/scratch/churastest/mychm"; qsub -t 1-3 runjobs.rocce

[churastest@login-0-0 churastest]$ |
```

The terminal window shows the execution of `createchmjob.py` and `checkchmjob.py`. The output includes job analysis details like tile size, histogram equalization, and task counts. It concludes with a command to run jobs: `cd "/data/scratch/churastest/mychm"; qsub -t 1-3 runjobs.rocce`. The text "Example Output" is at the bottom right.

## 3b. Processing loop

- A. When run successfully, `checkchmjob.py --submit` outputs a command to submit the jobs.

In the terminal copy the text line below "Run this:" and paste it in the terminal to submit the jobs.

In the first invocation of `checkchmjob.py` the text to copy will look similar to the text below where we will be submitting 3 jobs using the array job feature of SGE via `-t 1-3` option:

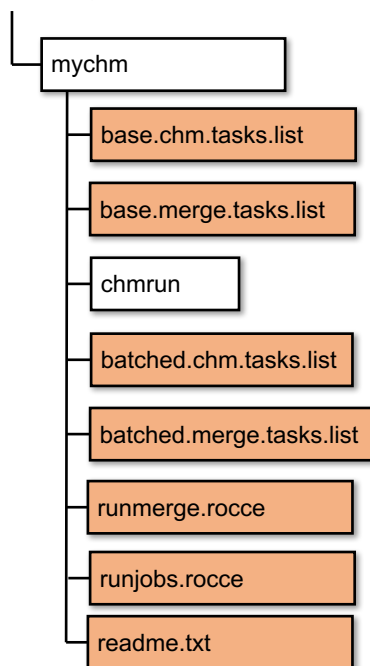
```
cd "/data/scratch/<USER>/mychm"; qsub -t 1-3 runjobs.rocce
```

The above command will put the terminal into `mychm` directory and invoke `qsub` to submit jobs to SGE.

The `;` character lets a user put multiple commands on a single line.

Run `man qsub` for more information

#### Directory Structure:



Run this:

```
cd "/data/scratch/churastest/mychm";qsub -t 1-3 runjobs.rocce
[churastest@login-0-0 churastest]$ cd "/data/scratch/churastest/mychm";qsub -t
1-3 runjobs.rocce
Your job-array 14740.1-3:1 ("chmjob") has been submitted
[churastest@login-0 mychm]$
```

When the *qsub* command is run, OGE/SGE will output a line of this format:

*Your job-array <JOBID>.<ARRAYID START>.<ARRAYID END>:1 ("<JOBNAME>") has been submitted*

The <JOBID> is the identifier for the job. This job is an array job so one <JOBID> will refer to multiple jobs that each have their own <ARRAYID> which is output in last column of *qstat* output.

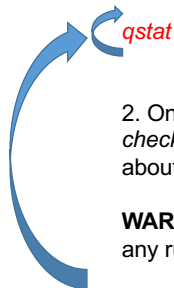
Files containing standard error/output for jobs can be found in *chmrun/stdout/<JOBID>.<ARRAYID>.out*.

Contents of these text files can be examined for debugging purposes.

## 3c. Processing loop

Completing the CHM job is now a repeat of the following operations.

1. Run *qstat* periodically and verify jobs have completed (jobs with ID should have 'c' complete state or not be listed)



2. Only **AFTER** *qstat* shows jobs have finished. Run *checkchmjob.py . --submit* and the *qsub* command. *checkchmjob.py* outputs to submit any remaining jobs. Once on a compute node CHM should only take about 3 minutes to run and the merge is just a few seconds

**WARNING:** Only add *--submit* to *checkchmjob.py* if all jobs have completed according to *qstat*, otherwise any running jobs will be run twice or will fail. It is safe to invoke *checkchmjob.py* without *--submit* anytime

*checkchmjob.py . --submit*

# Don't forget to run the *qsub* command output by *checkchmjob.py*

*checkchmjob.py* takes a directory path to job as its first argument. Since we are in *mychm* directory a period can be used which denotes current working directory.

For more information go to: <https://github.com/CRBS/chmutil/wiki/createchmjob.py>

How job 14740 with 3 tasks in queued state looks in qstat

Job <ARRAYID> this is task 3 for job 14740 and all jobs are running

Notice how qstat output nothing denoting jobs have finished

Once jobs finish, run checkchmjob.py again

checkchmjob.py showing CHM tasks have completed

```

churastest@login-0-0 mychm$ qstat
job-ID prior    name       user          state submit/start at   queue
-----
14740 0.00000 chmjob      churastest    qw   07/25/2017 14:38:23
[churastest@login-0-0 mychm]$ qstat
job-ID prior    name       user          state submit/start at   queue
-----
14740 0.50500 chmjob      churastest    r    07/25/2017 14:38:33 all.q@compute-
0-21.local 1 1
14740 0.50500 chmjob      churastest    r    07/25/2017 14:38:33 all.q@compute-
0-29.local 1 2
14740 0.50500 chmjob      churastest    r    07/25/2017 14:38:33 all.q@compute-
0-8.local  1 3
[churastest@login-0-0 mychm]$ qstat
[churastest@login-0-0 mychm]$ checkchmjob.py . --submit
Analyzing job. This may take a minute...

chmutil version: 0.8.0
Tiles: 512x512 with 0x0 overlap
Disable histogram equalization in CHM: True
Tasks: 50 tiles per task, 1 tasks(s) per node
Trained CHM model: /data/scratch/churastest/model
CHM binary: /data/churas/chm_s22.img

CHM tasks: 100% complete (3 of 3 completed)
Merge tasks: 0% complete (0 of 3 completed)

Run this:
cd "/data/scratch/churastest/mychm";qsub -t 1-3 runmerge.rocce
[churastest@login-0-0 mychm]$

```

Example Output

Output from checkchmjob.py when all jobs are complete

checkchmjob.py showing all tasks have completed

```

churastest@login-0-0 mychm$ cd "/data/scratch/churastest/mychm";qsub -t 1-3 ru
runmerge.rocce
Your job-array 14741.1-3:1 ("mergechmjob") has been submitted
[churastest@login-0-0 mychm]$ qstat
job-ID prior    name       user          state submit/start at   queue
-----
14741 0.00000 mergechmjo churastest    qw   07/25/2017 14:43:33
[churastest@login-0-0 mychm]$ qstat
job-ID prior    name       user          state submit/start at   queue
-----
14741 0.50500 mergechmjo churastest    r    07/25/2017 14:43:48 all.q@compute-
0-17.local 1 1
14741 0.50500 mergechmjo churastest    r    07/25/2017 14:43:48 all.q@compute-
0-24.local 1 2
14741 0.50500 mergechmjo churastest    r    07/25/2017 14:43:48 all.q@compute-
0-30.local 1 3
[churastest@login-0-0 mychm]$ qstat
[churastest@login-0-0 mychm]$ checkchmjob.py . --submit
Analyzing job. This may take a minute...

chmutil version: 0.8.0
Tiles: 512x512 with 0x0 overlap
Disable histogram equalization in CHM: True
Tasks: 50 tiles per task, 1 tasks(s) per node
Trained CHM model: /data/scratch/churastest/model
CHM binary: /data/churas/chm_s22.img

CHM tasks: 100% complete (3 of 3 completed)
Merge tasks: 100% complete (3 of 3 completed)

All jobs completed. Have a nice day!
[churastest@login-0-0 mychm]$

```

## 4. Understand CHM job structure

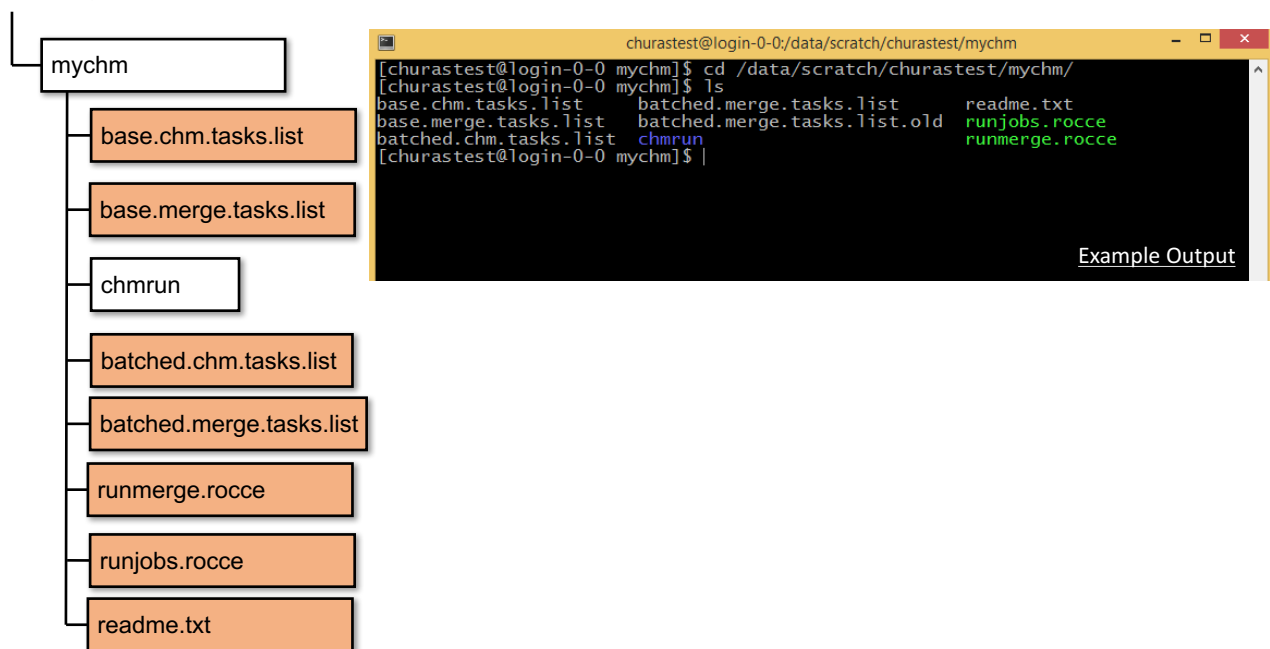
- A. Running `createchmjob.py` & `checkchmjob.py --submit` creates `mychm` directory with several files. Run `ls` to see a listing of files and directories.

`ls`

Below is a description of the files and directories seen from the `ls` command

<code>base.chm.tasks.list</code>	Main configuration file and contains CHM tasks to run
<code>batched.chm.tasks.list</code>	Configuration file that defines how CHM tasks in <code>base.chm.tasks.list</code> are batched onto individual compute nodes on the cluster
<code>batched.merge.tasks.list</code>	Configuration file that defines how Merge tasks in <code>base.merge.tasks.list</code> are batched onto individual compute nodes on the cluster
<code>base.merge.tasks.list</code>	Configuration file for Merge phase and contains merge tasks to run
<code>chmrun/</code>	Directory under which all job output is written. There are several directories within described later in this exercise
<code>readme.txt</code>	Contains information about this job including arguments used and how to get help
<code>runjobs.rocce</code>	OGE/SGE <code>qsub</code> submit script to run CHM tasks on Rocce cluster
<code>runmerge.rocce</code>	OGE/SGE <code>qsub</code> submit script to run Merge tasks on Rocce cluster

### Directory Structure:





## 4. Understand CHM job output structure

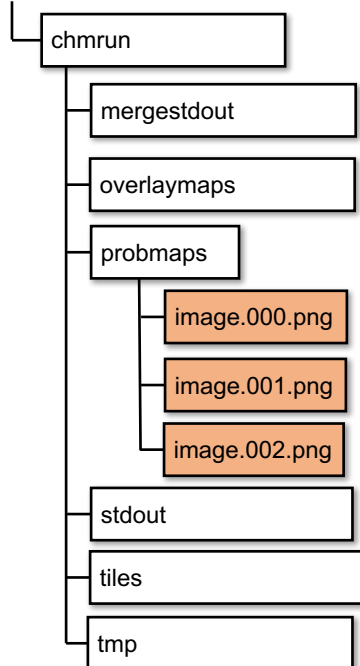
- A. Run `ls` on `chmrun/` to see the files and directories of CHM job output

`ls chmrun/probmaps`

Below is a description of the files and directories within `chmrun/` seen from the `ls` command

<code>mergestdout/</code>	Directory containing output from merge tasks. Merge tasks are directed to write to this path via <code>runmerge.rocce</code> queue submit script file.
<code>probmaps/</code>	Directory containing final output of merged probability maps. These images are created when the merge tasks are run.
<code>stdout/</code>	Directory containing output from CHM tasks. CHM tasks are directed to write to this path via <code>runjobs.rocce</code> queue submit script file.
<code>tiles/</code>	Directory containing directories for each input image. Within each of this input image directories the CHM tasks will write partial probability maps which will be merged by merge tasks in the SECOND phase of processing.
<code>tmp/</code>	Directory containing temp standard out and standard error files for <code>chmrunner.py</code> and <code>mergetilerunner.py</code> scripts. Once the task completes this directory and files will be removed.

### Directory Structure:



```
churastest@login-0-0:/data/scratch/churastest/mychm
[churastest@login-0-0 mychm]$ ls chmrun/
mergestdout overlaymaps probmaps stdout tiles tmp
[churastest@login-0-0 mychm]$ ls chmrun/probmaps
image.000.png image.001.png image.002.png
[churastest@login-0-0 mychm]$
```

Example Output

Final probability map images can be found under `probmaps/` directory with file names that match those of the input images.

## 9. Download data from Rocce

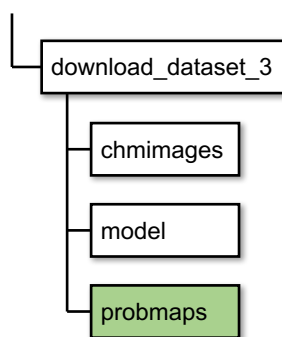
The merged probability map images are located in `chmrun/probmaps/<image name>`. The commands below exit the terminal from Rocce cluster and use `scp` to copy back the `probmaps` directory and images within

```
exit
cd ~/download_dataset_3
scp -i ~/.ssh/id_rsa -r <USER>@rocce.ucsd.edu:/data/scratch/<USER>/mychm/chmrun/probmaps .
```

Don't forget to include the period above at the end.

For more information go to: <https://github.com/CRBS/chmutil/wiki/createchmjob.py>

Directory Structure:



```
~/download_dataset_3
[churastest@login-0-0 mychm]$ exit
logout
Connection to rocce.ucsd.edu closed.

smithy@smith ~/download_dataset_3
$ cd ~/download_dataset_3

smithy@smith ~/download_dataset_3
$ scp -i ~/.ssh/id_rsa -r churastest@rocce.ucsd.edu:/data/scratch/churastest/mychm/chmrun/probmaps .
Could not create directory '/home/smithy/.ssh'.
The authenticity of host 'rocce.ucsd.edu (169.228.41.199)' can't be established.
RSA key fingerprint is 40:e0:56:fb:2b:38:8d:99:e9:4f:3e:7c:b8:a9:f3:e6.
Are you sure you want to continue connecting (yes/no)? yes
Failed to add the host to the list of known hosts (/home/smithy/.ssh/known_hosts).
Enter passphrase for key '/cygdrive/c/Users/smithy/.ssh/id_rsa':
image.000.png      100% 114kB 114.1kB/s   00:00
image.001.png      100% 130kB 130.4kB/s   00:00
image.002.png      100% 123kB 122.7kB/s   00:00

smithy@smith ~/download_dataset_3
$ ls
chmimages  model  probmaps

smithy@smith ~/download_dataset_3
$ |
```

Open probability map image files  
in favorite image viewer

For example, on Windows one  
could do the following to look at  
the first image by running the  
Following from the terminal:

*cd probmaps*

*mspaint.exe image.000.png*

