scaErcanLab_ChIP-seq_analysis_v12.1_slurm

Author: Sevinc Ercan
Implementation 01/2021: Diogo Mesquita - dam740@nyu.edu; Lena Street - las821@nyu.edu; Jun Kim - kimj50@nyu.edu
Update: 2/19/2025 Yuya Zhao v11.1 to v11.2 is switch of UCSC trackhubs to web server
Updated: 06/17/2025 Daniel Garbozo v11.2 to v12 is switch of single end to paired-end sequencing, implementation of sanity checks and low mapping diagnosis.
Updated: 10/14/2025 Daniel Garbozo v12 to v12.1 include single end pipeline. Included quality controls of both pipelines.

##############################
##Paired end Pipeline ##
##############################

1. Create a new directory *WD* for the data (*WD* for working directory)
   cmd: mkdir /scratch/**NYUID**/*NewChIPData*
   cmd: cd /scratch/**NYUID**/*NewChIPData*

2. Copy the new data (*.fastq.gz* files).
   cmd: cp /scratch/cgsb/gencore/out/Ercan/**DirectoryName**/**FileName*** /scratch/**NYUID**/NewChIPData/.

3. Rename the files WITHOUT SPACES
   cmd: mv Oldfilename.fastq Newfilename.fastq
   - conventional naming ChIP:
     <descriptive_name>_<extract>_<sequence_ID>_input_<input_ID>_<R1/2>.fastq.gz
     Example: 8WG16_Millipore_JEL1197_males_L2L3_ext725_DO103_KAPA_input_DO97_KAPA_R2.fastq.gz
   - conventional naming Input:
     input_<strain>_<stage>_<extract>_<input_ID>_<R1/2>.fastq.gz
     Example: input_JEL1197_males_L2L3_ext725_DO97_KAPA_R2.fastq.gz

4. Create a directory to store the script (*WD/script*)
   cmd: mkdir /scratch/**NYUID**/*NewChIPData/script*
   cmd: cd /scratch/**NYUID**/*NewChIPData/script*
5. Copy the required files to the *WD/script*
   - Script
   cmd: cp /scratch/cgsb/ercan/scripts/chip/slurm/run_chip_PE_DG_v2.sh .

   - Configuration file
     The example configuration file is named config.env and it can be found in the drive:
     https://drive.google.com/drive/folders/1qbLCBYKRDNqj9x-7F3v2IPxHMfEbUnaq?hl=es
     or you can get it directly in prince to your *WD/script* with:
   cmd: cp /scratch/cgsb/ercan/scripts/chip/slurm/config_PE.env .

     [IMPORTANT] After either downloading or copying the configuration file, you must edit it with the information about your working directory path that **MUST** include the fastq directory.
     config.env:

```
# === Project paths ===
WORK_DIR="/scratch/dg4739/chip_test" # ← Change this path to your working directory (WD)
ADAPTERS="/scratch/dg4739/adapters/TruSeq3-PE.fa"
INDEX_PREFIX="/scratch/cgsb/ercan/annot/forBowtie/c_elegans.WS220"
```

```
BLACKLIST="/scratch/cgsb/ercan/annot/ce10-blacklist.bed"
BLAST_DB="/vast/work/public/genomics/ncbi/blast/db/nt/nt"

# === Parameters ===
# Trimming
CROP=75
MINLEN=35
# Alignment
MAXINS=800
# Bamcoverage & Bamcompare
BIN_SIZE=10
# Fingerprint
FINGERPRINT_N=2000000
# Blast
SUBSAMPLE_SIZE=30000
```

If you edited the configuration locally on your PC transfer it to prince with:
cmd: scp path/to/config.env $NYUID@prince.hpc.nyu.edu:/scratch/NYUID/*NewChIPData*/script/.

6. Run the ChIP-seq pipeline and wait for it to finish
   (inside *WD/scripts/*)
   cmd: sbatch run_chip_PE_DG_v2.sh config_PE.env

   To check if it's still running type:
   cmd: squeue -u $NYUID

   The previous command also works to get the $JOBID of the job (in the output of the command look for the row with name "chip")
   Also the logs of the job (i.e. the printed messages) will be stored in /scratch/$NYUID/script/slurm_$JOBID_chip_blast.out

   As a sanity check you can look at the top of this log file to make sure the program read the configuration file correctly (**cmd**: less /scratch/$NYUID/script/slurm_$JOBID_chip_blast.out)
   You can look at this log file while the job is running

##############################
##Single end Pipeline ##
##############################

1. Create a new directory *WD* for the data (*WD* for working directory)
   cmd: mkdir /scratch/**NYUID**/*NewChIPData*
   cmd: cd /scratch/**NYUID**/*NewChIPData*

2. Copy the new data (*.fastq.gz* files).
   cmd: cp /scratch/cgsb/gencore/out/Ercan/**DirectoryName**/**FileName*** /scratch/**NYUID**/NewChIPData/.

3. Rename the files WITHOUT SPACES
   cmd: mv Oldfilename.fastq Newfilename.fastq
   - conventional naming ChIP:
     <descriptive_name>_<extract>_<sequence_ID>_input_<input_ID>.fastq.gz
     Example: 8WG16_Millipore_JEL1197_males_L2L3_ext725_DO103_KAPA_input_DO97.fastq.gz

- conventional naming Input:
  input_<strain>_<stage>_<extract>_<input_ID>.fastq.gz
  Example: input_JEL1197_males_L2L3_ext725_DO97.fastq.gz

4. Create a directory to store the script (*WD/script*)
   cmd: mkdir /scratch/**NYUID**/*NewChIPData/script*
   cmd: cd /scratch/**NYUID**/*NewChIPData/script*
5. Copy the required files to the *WD/script*
   - Script

   cmd: cp /scratch/cgsb/ercan/scripts/chip/slurm/run_chip_SE_DG_v4.sh .

   - Configuration file
     The example configuration file is named config.env and it can be found in the drive:
     https://drive.google.com/drive/folders/1qbLCBYKRDNqj9x-7F3v2IPxHMfEbUnaq?hl=es
     or you can get it directly in prince to your *WD/script* with:

   cmd: cp /scratch/cgsb/ercan/scripts/chip/slurm/config_SE.env .

   [IMPORTANT] After either downloading or copying the configuration file, you must edit it with the information about your working directory path that **MUST** include the fastq directory.
   config.env:

```
# Change this path to you working directory that MUST contain fastq directory
# fastq directory MUST naming "fastq" and it contains fastq files.
WORK_DIR="/scratch/cgsb/ercan/DanielG/2025_09_22-PEvsSE-testing/single_end"

# probably don't need to change the next lines
# === Project paths ===
INDEX_PREFIX="/scratch/cgsb/ercan/annot/forBowtie/c_elegans.WS220"
BLACKLIST="/scratch/cgsb/ercan/annot/ce10-blacklist.bed"
BLAST_DB="/vast/work/public/genomics/ncbi/blast/db/nt/nt"

#  Parameters
## Bamcoverage & Bamcompare
BIN_SIZE=10
## BLAST
SUBSAMPLE_SIZE=30000
```

   If you edited the configuration locally on your PC transfer it to prince with:
   cmd: scp path/to/config.env $NYUID@prince.hpc.nyu.edu:/scratch/NYUID/*NewChIPData*/script/.

6. Run the ChIP-seq pipeline and wait for it to finish
   (inside *WD/scripts/*)
   cmd: sbatch run_chip_PE_DG_v2.sh config_PE.env

   To check if it's still running type:
   cmd: squeue -u $NYUID

   The previous command also works to get the $JOBID of the job (in the output of the command look for the row with name "chip")

Also the logs of the job (i.e. the printed messages) will be stored in /scratch/$NYUID/script/slurm_$JOBID_chip_blast.out

As a sanity check you can look at the top of this log file to make sure the program read the configuration file correctly (**cmd**: less /scratch/$NYUID/script/slurm_$JOBID_chip_blast.out)
You can look at this log file while the job is running

##############################
##Checking ##
##############################

Check if the job ran successfully

a. check your email

You will get an email for the start and end of the job that the ChIP-seq pipeline spawns. Check the email that signals the end of a job and make sure that they have a COMPLETED status (as opposed to FAILED) and that the exit code is [0-0].

b. Read slurm_$JOBID_chip_blast.out to find errors:

- **cmd**: less /scratch/$NYUID/script/slurm_$JOBID_chip_blast.out

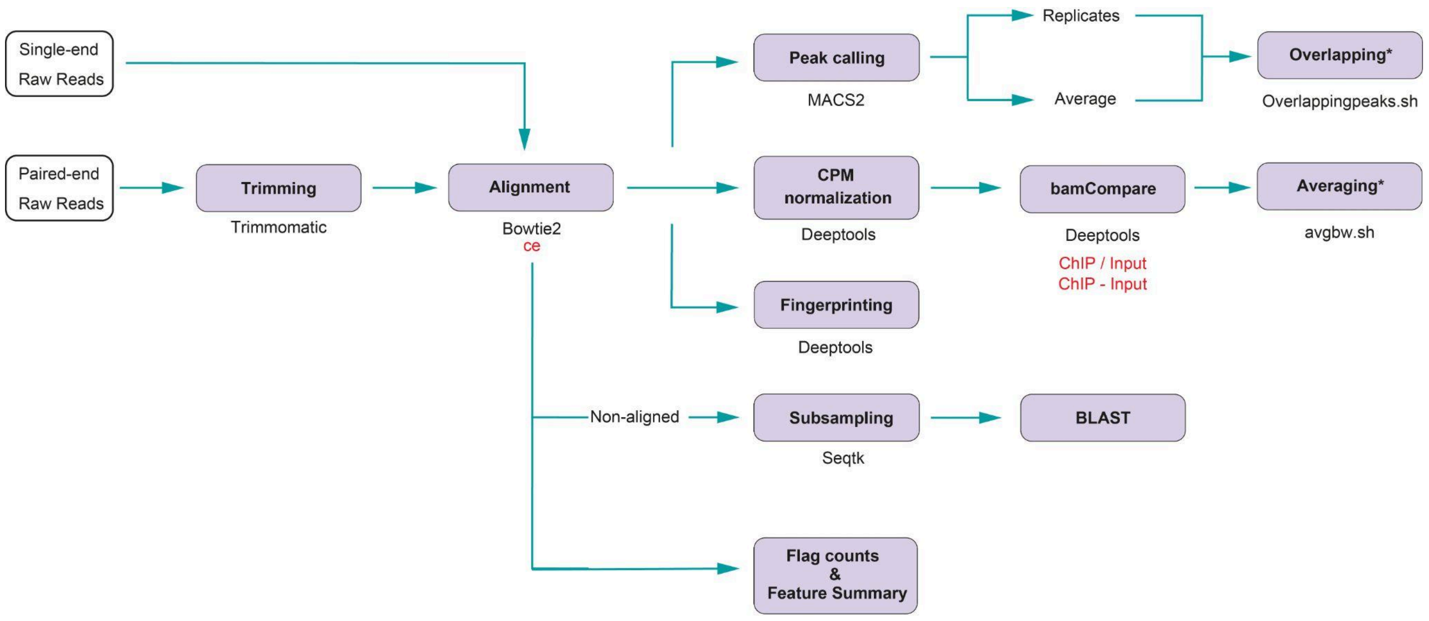
Some errors are not reported in the emails. And the “slurm” file might catch some of these. If you don’t notice any errors this file finished with **“✅ Pipeline version 2 completed”**, and many steps like something like this:

```
[INFO] Aligning raw-8WG16_Millipore_JEL1197_males_L2L3_ext725_D0103_KAPA_input_D097_KAPA
...
[INFO] Converting SAM to BAM and sort
[INFO] Rename chromosomes for UCSC:
8WG16_Millipore_JEL1197_males_L2L3_ext725_D0103_KAPA_input_D097_KAPA
[INFO] Indexing renamed BAM
[INFO] CPM normalization for 8WG16_Millipore_JEL1197_males_L2L3_ext725_D0103_KAPA_input_D097_KAPA
...
[INFO] Identifying ChIP with Input BAM file:
8WG16_Millipore_JEL1197_males_L2L3_ext725_D0103_KAPA_input_D097_KAPA_sorted.bam
  ↳ Captured extNumber: 725
  ✅ Matched Input BAM: input_JEL1197_males_L2L3_ext725_D097_KAPA_sorted.bam
[INFO] Running bamCompare ...
[INFO] Generated
8WG16_Millipore_JEL1197_males_L2L3_ext725_D0103_KAPA_input_D097_KAPA_sorted_subtract.bw
[INFO] Running plotFingerprint for
8WG16_Millipore_JEL1197_males_L2L3_ext725_D0103_KAPA_input_D097_KAPA_sorted.bam vs
input_JEL1197_males_L2L3_ext725_D097_KAPA_sorted.bam
[INFO] === Finished ChIP vs Input comparisons ===

[INFO] === CONTAMINATION DIAGNOSIS STEP - subsample and BLAST of 30000 random reads from non-aligned
8WG16_Millipore_JEL1197_males_L2L3_ext725_D0103_KAPA_input_D097_KAPA

Wed Jun 03 11:42:01 AM EDT 2025: ✅ Pipeline completed
```

If you find errors you will see the corresponding error messages (note that in any STEP of the pipeline are registered and indicate in which part of the pipeline the error occurred). For example, if there are errors you will see something like this:

```
[INFO] Running bamCompare --operation subtract
--bam1 8WG16_Millipore_JEL1197_males_L2L3_ext725_D0103_KAPA_input_D097_KAPA_sorted.bam
--bam2 input_JEL1197_males_L2L3_ext725_D097_KAPA_sorted.bam
--outFileName
8WG16_Millipore_JEL1197_males_L2L3_ext725_D0103_KAPA_input_D097_KAPA_sorted_subtract.bw
usage:  bamCompare -b1 treatment.bam -b2 control.bam -o log2ratio.bw
bamCompare: error: the following arguments are required: --bamfile1/-b1, --bamfile2/-b2
```

################################
##Graphic Pipeline   ##
################################



*: The next improvements will include these steps

At this point - you're all done processing the ChIP data. If you have followed this protocol in its entirety your scratch/**NYUID**/NewChIPData folder should look like this:

```
[dg4739@log-2 paired_end]$ ls
BAM  coverage  fastq  MACS2output  quality_control  script
[dg4739@log-2 single_end]$ ls
alignment  coverage  fastq  MACS2output  quality_control  script
```

Inside the directories there are:

- Single end

```
[dg4739@log-2 chip_test] find . -type d
./blast
./blast/06_03-blast_results
./blast/06_03-subsample_30000_aligned_reads_fasta
./coverage
./coverage/06_03-CPM_normalization
./coverage/06_03-bam_compare
./fastq
./quality_control
```

```
./quality_control/fingerprint
./quality_control/fingerprint/06_03-plots
./raw_alignment
./raw_alignment/06_03-raw_aligned
./scripts
./trimmed_alignment
./trimmed_alignment/06_03-aligned
./trimmed_alignment/06_03-non-aligned
./trimming
./trimming/06_03-trimmed-wo-adaptors
./trimming/06_03-unpaired
./trimming/log

[dg4739@log-2 single_end]$ find . -type
./script
./coverage
./coverage/2025_09_27-bam_compare
./coverage/2025_09_28-CPM_normalization
./coverage/2025_09_27-CPM_normalization
./coverage/2025_09_28-bam_compare
./fastq
./MACS2output
./alignment
./alignment/2025_09_27-non-aligned
./alignment/2025_09_28-non-aligned
./alignment/2025_09_28-BAM-aligned
./alignment/2025_09_27-BAM-aligned
./quality_control
./quality_control/2025_09_28-fingerprint
./quality_control/2025_09_28-counts_read
./quality_control/blast
./quality_control/blast/2025_09_27-blast_results
./quality_control/blast/2025_09_28-blast_results
./quality_control/blast/2025_09_27-subsample_30000_aligned_reads_fasta
./quality_control/2025_09_27-fingerprint
./quality_control/2025_09_27-counts_read
```

- Paired end

```
[dg4739@log-2 paired_end]$ find . -type d
./script
./BAM
./BAM/trimming
./BAM/trimming/2025_09_27-trimmed-wo-adaptors
./BAM/trimming/2025_10_06-unpaired
./BAM/trimming/log
./BAM/trimming/2025_10_06-trimmed-wo-adaptors
./BAM/trimming/2025_09_27-unpaired
./BAM/trimmed_alignment
./BAM/trimmed_alignment/2025_09_27-non-aligned
./BAM/trimmed_alignment/2025_09_27-aligned
./BAM/trimmed_alignment/2025_10_06-non-aligned
```

```
./BAM/trimmed_alignment/2025_10_06-aligned
./BAM/raw_alignment
./BAM/raw_alignment/2025_09_27-raw_aligned
./BAM/raw_alignment/2025_10_06-raw_aligned
./coverage
./coverage/2025_09_27-bam_compare
./coverage/2025_10_06-CPM_normalization
./coverage/2025_09_27-CPM_normalization
./coverage/2025_10_06-bam_compare
./fastq
./MACS2output
./quality_control
./quality_control/2025_10_06-counts_read
./quality_control/2025_09_27-blast
./quality_control/2025_09_27-blast/2025_09_27-blast_results
./quality_control/2025_09_27-blast/2025_09_27-subsample_30000_aligned_reads_fasta
./quality_control/2025_10_06-blast
./quality_control/2025_10_06-blast/2025_10_06-blast_results
./quality_control/2025_10_06-fingerprint
./quality_control/2025_09_27-fingerprint
./quality_control/2025_09_27-counts_read
```

##############################
## Low Mapping diagnosis   ##
##############################

1. Analyze BLAST results to identify predominant species in non-aligned reads (vs C. elegans reference)

→ Example: For input sample 'HW125', examine top BLAST hits for non-C.elegans sequence

```
wc -l
8WG16_Millipore_JEL1197_males_L2L3_ext725_DO103_KAPA_input_DO97_KAPA_subsample_30000_blastn_best_hit
_per_read.tsv
29489 ## Number of hits.

grep -c "Escherichi"
8WG16_Millipore_JEL1197_males_L2L3_ext725_DO103_KAPA_input_DO97_KAPA_subsample_30000_blastn_best_hit
_per_read.tsv
27997 ## Number of hits that have "Escherichi" term. --> 27997/29489 (~ 94.940%)

grep -i -c "homo\|human"
8WG16_Millipore_JEL1197_males_L2L3_ext725_DO103_KAPA_input_DO97_KAPA_subsample_30000_blastn_best_hit
_per_read.tsv
24  --> ## Number of hits that have "homo" or “human” or “Human”, etc terms. 24/29489 (~ 0.08%)
```

###################
#generate final peaks#
###################

The v11 pipeline doesn’t generate the “final” MACS peaks automatically. Use the following scripts to generate the “final” peaks where the avg peak profile is used as base, and only the peaks that show up in the majority of replicates are kept.

cmd: cd MACSoutput

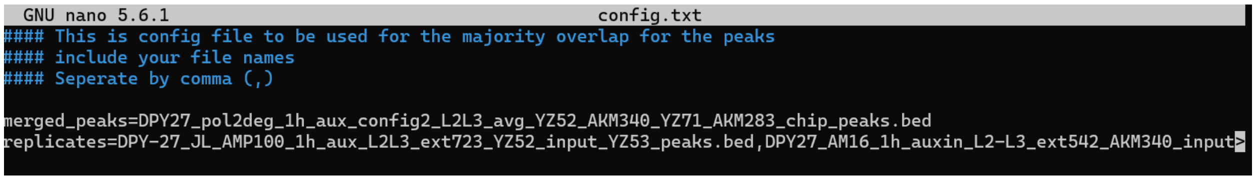
#move into the directory where MACS peak calling output is kept.

cmd: cp /scratch/cgsb/ercan/scripts/chip/slurm/getOverlappingPeaks_DO_config.txt /scratch/cgsb/ercan/scripts/chip/slurm/getOverlappingPeaks_DO.sh .

#copy the required scripts to your working directory

cmd: nano getOverlappingPeaks_DO_config.txt

#edit the config file as described



```
  GNU nano 5.6.1                                    config.txt
#### This is config file to be used for the majority overlap for the peaks
#### include your file names
#### Seperate by comma (,)

merged_peaks=DPY27_pol2deg_1h_aux_config2_L2L3_avg_YZ52_AKM340_YZ71_AKM283_chip_peaks.bed
replicates=DPY-27_JL_AMP100_1h_aux_L2L3_ext723_YZ52_input_YZ53_peaks.bed,DPY27_AM16_1h_auxin_L2-L3_ext542_AKM340_input>
```

# Note: Put ALL the replicates in ONE LINE separated by a comma (,) WITHOUT SPACE.

cmd: sh getOverlappingPeaks_DO.sh

#submit slurm job.

It should take just a few minutes to run the job. Edit the config file and rerun sh for each protein/condition.

########################
#generate average bigwigs: #
########################

Daniel method Jan 2021

The bash script is a wrapper for the avgbw.py file that loads the needed modules and also takes input for the file name of the average bigwig output.

The script will average values at each position for all replicates provided (from Jun's documentation).

cmd: srun -c1 -t0:30:00 --mem=4000 --pty /bin/bash

#request an interactive session.

cmd: cp /scratch/cgsb/ercan/scripts/chip/slurm/avgbw.sh ./

#copy script to your current working directory with the bigwigs you want to average

cmd: ./avgbw.sh rep1.bw rep2.bw rep3.bw

#run the script. it will prompt you to enter the file name of the average output

Jun method - retired Jan 2021:

The script will average values at each position for all replicates provided.

cmd: srun -c1 -t1:00:00 --mem=40000 --pty /bin/bash

#request an interactive session with lots of memory, the script is quick, but memory intensive

cmd: cd /scratch/**NYUID**/NewChIPData/InputSubtCoverage

cmd: cp /scratch/cgsb/ercan/scripts/chip/slurm/avgbw.py /scratch/**NYUID**/NewChIPData/InputSubtCoverage/.

#copy the script from ercan directory

cmd: module load pybigwig/0.3.17

#the script requires a module installed on HPC

cmd: python avgbw.py rep1.bw rep2.bw rep3.bw

#simply add the name of bigwig files you want to average, separated by spaces

#if you want to average all files starting with DPY27_N2, then simply type python avgbw.py DPY27_N2*

#if the input files are in different directories, you can provide the full PATH to the file.

Ex: python avgbw.py /scratch/kimj50/ChIPrun_1/DPY27_N2_rep1.bw /scratch/kimj50/ChIPrun_2/DPY27_N2_rep2.bw

output file: average.bw

make sure to rename the file (**cmd**: mv original_name new_name), if you plan to generate another average.bw file in the same directory

########################
##Create a track hub: ##
########################

1. Log into ap-static server using your NYU credentials
   cmd: ssh NYUID@ap-static.bio.nyu.edu
   Note: If you get access denied, ask Sevinc to request access for your HPC account.
2. Go into the folder with the UCSC files:
   cmd : cd /var/www/sites/ercan-tracks.bio.nyu.edu

3. (optional) make your own folder to hold the forUCSC files in ap-static
   cmd: mkdir NAME_OF_NEW_FOLDER
   cmd: cd NAME_OF_NEW_FOLDER
4. Download the forUCSC folder from greene to ap-static
   cmd: scp -r NYUID@greene.hpc.nyu.edu:PATH_TO_forUCSC .
   Note: You should copy **only the forUCSC folder**, not the entire directory generated by the ChIP seq pipeline!!
   (e.g.: /scratch/cgsb/ercan/yuya/chip/chip_tutorial/forUCSC)
5. Copy the required files to your forUCSC folder
   cmd: cd forUCSC
   cmd: cp /var/www/sites/ercan-tracks.bio.nyu.edu/scripts/config_trackhub.yaml /var/www/sites/ercan-tracks.bio.nyu.edu/scripts/make_trackhub.sh .

6. Edit the configuration file to match your analysis
   Again, the details on how to do this are specified in the configuration file itself

7. Create the trackhub
   cmd: source make_trackhub.sh

   Wait a few seconds

8. Add the track hubs to your UCSC browser
   https://data:sdc3&dpy27$@ercan-tracks.bio.nyu.edu/replicates/
   User name: data
   password: sdc3&dpy27$
   You now have several track hubs you can add to the UCSC browser.

   To add a trackhub go to https://genome.ucsc.edu . There select *My data* > *Track hubs* > *My hubs* and in the URL field input the urls that are detailed below.

   The first track hub you have is the one with all ChIP-seq experiments sequenced at one date. To add this trackhub use the following url:
   https://data:sdc3&dpy27$@ercan-tracks.bio.nyu.edu/replicates/folder_name/hub.txt Where ***date*** is the date of sequencing in *Ercan_Lab_Data_WS220* google sheet.

   Second, you have the track hubs of the averages. Here you will have one track hub for each protein (or sample) you chipped. Add one by one to the browser using the url:

/var/www/html/myHubs_v2/averages/***name***/hub.txt
https://data:sdc3&dpy27$@ercan-tracks.bio.nyu.edu/averages/***name***/hub.txt
Where ***name*** is the name you want to give your hub (e.g. David2020)

9. Add the track hub info to the TrackHubs tab of the *Ercan_Lab_Data_WS220* document in google drive

10. Adding additional files to the track hubs (e.g. when building a hub for a paper)
    a. Put the file in the correct folder
       cmd: scp $NYUID@greene.hpc.nyu.edu:/path/to/file
       /var/www/sites/ercan-tracks.bio.nyu.edu/scripts/averages/***name***/ce10

    b. Rerun the track hub configuration perl script
       cmd: cd /var/www/sites/ercan-tracks.bio.nyu.edu/averages/***name***/ce10
       cmd: ls *bw *bb > files.txt
       cmd: perl loadBWBBToUCSC.pl files.txt

    c. Add again the trackhub to the UCSC browser by following step 5 and using the url:
       /var/www/sites/ercan-tracks.bio.nyu.edu/***name***/hub.txt

11. Add the track hub info to the TrackHubs tab of the *Ercan_Lab_Data_WS220* document in google drive
    a. The link to the document is
       https://docs.google.com/spreadsheets/d/1xF8nNs5dqMsMv8Ot29Hhm4Yazjs5UTwtAditn3drjCc/edit#gid=21

12. Add the trackhub containing directory and its contents to the Ercan google drive folder to backup and share files with lab members. The google drive folder containing the lab trackhubs is located here:
    https://drive.googl.com/drive/u/1/folders/1bMq3g0psmllmVTGD8Treo-_mm_SMYDhb

######################
### V11 update notes ###
######################
The script to combine peaks of replicates after MACS does not work correctly. Instead of using the merged/average peaks as base, the script uses the last entered replicate as base. Therefore, the resulting “final” peaks will differ depending on the order of the config file. In v11, the “final” peak file is removed to avoid confusion. If the “final” peak file is needed, use a separate script to generate the “final” peaks.

Detailed notes of this update can be found chip_final_peaks_notes