

Various BASH Scripts

```
#####
```

```
# 1) BASH Script for adding BAM tracks to JBrowse
```

```
#!/bin/bash
```

```
#[LIVE script for live server]
```

```
#This (live) script reads in a config file of BAM locations, makes soft links, and adds to JBrowse data/tracks.conf and private data_private/tracks.conf files.
```

```
#make dir to hold BAM directories
```

```
#first check if already exists
```

```
if [ -d "BAM_SoftLinks" ]
```

```
then
```

```
    echo "Can't make directory BAM_SoftLinks; directory already exists"
```

```
else
```

```
    mkdir BAM_SoftLinks
```

```
fi
```

```
#Read in config file of BAM locations: BAMsites.txt or similar
```

```
IFS=$'\t'    #set internal field separator to tab
```

```
while read ID CAT BAM BAI; do #read in IDs, Categories, BAM and BAI locations
```

```
    #check if dir already exists, make dir for ID if not
```

```
    if [ -d "BAM_SoftLinks/$ID" ]
```

```
    then
```

```
        echo "Can't make directory $ID; directory already exists"
```

```
    else
```

```
        mkdir "BAM_SoftLinks/$ID" #make directory for soft link
```

```
    fi
```

```
        #create soft links
```

```
        #create BAM soft link
```

```
        if [ -e "BAM_SoftLinks/$ID/$ID.bam" ]
```

```
        then
```

```
            echo "Can't make file $ID.bam; file already exists."
```

```
        else
```

```
            ln -s "$BAM" "BAM_SoftLinks/$ID/$ID.bam"
```

```
        fi
```

```
        #create BAI soft link
```

```
        if [ -e "BAM_SoftLinks/${ID}/${ID}.bam.bai" ]
```

```
        then
```

```

        echo "Can't make file $ID.bam.bai; file already exists."
    else
        ln -s "$BAI" "BAM_SoftLinks/$ID/$ID.bam.bai"
    fi

    #Archive tracks_test.conf in public and private data dir
    dt=`date '+%d-%m-%Y_%H-%M-%S'`
    cp data/tracks_test.conf "data/Archive/tracks_test_${dt}.conf"
    cp data_private/tracks_test.conf "data_private/Archive/tracks_test_${dt}.conf"

    #Add new section text via perl script call
    perl template_writer.pl "$ID" "$CAT"

done<BAMsites_PublicJBrowse_2020-05-20_CP.txt

echo "BAM script finished."

#####

# 2) Perl script for writing BAM specs to JBrowse config file

#!/usr/bin/perl -w

#perl script (live) to add JBrowse template text

#get ID value from command line call
my $ID = shift @ARGV;
#get CAT category value from command line call
my $CAT = shift @ARGV;

#open public config file for writing
my $config = "data/tracks.conf";
open(my $out, '>>', $config);
#open private config file for writing
my $config_priv = "data_private/tracks.conf";
open(my $out_priv, '>>', $config_priv);

#read in template file and replace on output
my $template = "data/tracks_section_template.txt";
open (IN, $template) or die "Can't open file $!";
print "Adding new section $ID to tracks config file.\n";

while(my $row = <IN>)      {
    chomp $row;

```

```

        #regex replacement - replace $ID in template with passed-in $ID value
        $row =~ s/\$ID/$ID/g;
        #replace $CAT in template with passed-in $CAT value
        $row =~ s/\$CAT/$CAT/g;
        #print to public config
        print $out "$row\n";
        #print to private config
        print $out_priv "$row\n";
    }
    close(IN);
    close $out;
    close $out_priv;

    print "Finished adding new section $ID to tracks config file.\n";

#####

# 3) File-transfer scripts

# 3A) Transfer BAM folder with checksum

#/bin/bash

#This script conducts a checksum on a full directory (in Jubail server)
#Run within full 'Muriel' directory 5-13-23
#Run WITHIN bam dir 5-14-23

#move to appropriate directory to execute command
#cd /scratch/share/cgsb/Muriel
cd /scratch/share/cgsb/Muriel/bams

#set the subdirectory to generate checksum for
#transfer="/scratch/share/cgsb/Muriel/ids_read1.sh"
#dir="Muriel"
dir="bams"

echo "Starting checksum of $dir"
find . -type f -exec md5sum {} + | LC_ALL=C sort | md5sum
echo "Finished checksum of $dir"

# 3B) Transfer folder of BAMs

#/bin/bash

```

```
#This script conducts a file transfer, pushes from Jubail to Greene using RSYNC
#5-8-23
```

```
#move to appropriate directory
cd /scratch/mid224/transtest_4-18-23
```

```
#set the directory or file to be transferred
transfer="/scratch/share/cgsb/Muriel/bams"
#set destination on Greene
GreeneDest="/scratch/mid224/data/alUla_data"
```

```
echo "Starting RSYNC of $transfer"
rsync -avrP $transfer mid224@greene.hpc.nyu.edu:"$GreeneDest"
echo "Finished RSYNC of $transfer"
```

```
# 3C) Transfer forward and reverse read files from manifest
```

```
#!/bin/bash
```

```
#This script conducts a file transfer, pulls from Jubail to Greene using RSYNC
```

```
#move to appropriate directory
cd /scratch/mid224/transfer_test_4-18-23
```

```
#Read in manifest of files
```

```
IFS=$'\t' #set internal field separator to tab
```

```
while read ID F R; do
```

```
    echo -e "Manifest contents: ID: ${ID} Forward file: ${F} Reverse file: ${R}\n"
```

```
    #copy over F and R files
```

```
    echo "Starting RSYNC of ${F}"
```

```
    rsync -avrP -e "ssh -p 4410" mid224@10.230.240.213: "/storage/illumina/dna/${ID}/${F}"
```

```
RSYNCcopies_test
```

```
    echo "Finished RSYNC of ${F}"
```

```
    echo "Starting RSYNC of ${R}"
```

```
    rsync -avrP -e "ssh -p 4410" mid224@10.230.240.213: "/storage/illumina/dna/${ID}/${R}"
```

```
RSYNCcopies_test
```

```
    echo "Finished RSYNC of ${R}"
```

```
done<manifest.3.2.2021.tsv
```

```
echo "Finished RSYNC of all files."
```