**MIPSTR Custom Scripts Read-me**

All scripts are written for Shell or Python v. 2.7.2. You will also need NumPy, BWA, and SAMtools. Run these scripts from the directory that contains them.

**1. Make synthetic references**

Run python script *make\_ref\_standard.py*

* You must have a **reference locus master text file** in myfile = open().
  + This is a csv or text file with a header row and the following six columns: 1. STR locus ID

2. Sequence captured by MIP including TA sequences

3. Unit number in reference

4. Unit sequence in reference- *e.g.* CAA

5. Unit size in reference- *e.g.* trinucleotide = 3

6. STR start- number of bp in captured sequence where the STR starts

* You must tell python where you want the output fasta files to go by refseq = open().
  + For example, I put them into my folder 100\_repeat\_reference\_seqs.
  + The script will output one fasta file for each of your target STR loci named for the corresponding STR locus ID.

**2. Index synthetic references using BWA**

Run shell script *MakeIndex.sh*

* Set refDir to your folder containing all the fasta files from step one.

**Steps 1 and 2 only need to be done once for each target (i.e. any experiments done using the same MIPs can be mapped back to these references)**

**3. Sort reads to corresponding STR locus**

Run shell script *MIP\_sort\_scroll.sh*

* libDir should be set to the directory that contains all library folders, within which are the fastq files off the sequencer.
* You also need an **extension arm sequence file** (*e.g.* EAdict.txt). This file has two columns:
  + STR locus IDs
  + the sequence of the extension arm 5’ to 3.’

This shell script runs the python script *MIP\_sort\_by\_TA.py*

**4. Map to synthetic references (requires BWA and SAMtools)**

Run shell script *map\_process\_fqs.sh*

* fastqDir needs to be set to your directory containing the folders (different accessions or individuals) which have your fastq files now sorted by STR locus (previously called libDir in step 3).
* refDir needs to be set to the directory where you have the fasta reference files you made and indexed in steps 1 and 2
* This script will put all the standard BWA/SAMtools output files in the alignments folder for each locus.

**5. Get genotype and sort by degenerate tag**

Run shell script *degen\_count\_repeats.sh*

* fastqDir should be the same as in step 4.
* This script will make a directory in each library folder called degen\_counts where it will put .txt and .txt.raw files.
* It runs the python script *sorting\_degens.py*
  + It also opens a file called **master\_for\_calling.csv** which is a csv file with three columns:

1. STR locus ID
2. unit size
3. STR start

(all from locus master text file used in step 1)

* It includes unit number from a read if the alignment score A>=180
* If read ends within STR itself, it reports negative unit number

**6. Take mode of all reads from same degenerate tag**

Run shell script *scroll\_degen\_counts.sh*

* libDir is the same as fastqDir in 3 and 4…sorry for the inconsistency
* This will make a directory called multimodes for each library.
* This script runs python script *call\_by\_degen.py* on txt.raw files made in step 5
  + If there is a single mode and at least three reads in the tag-defined read group, it calls that mode as the unit number, otherwise it reports -1.
  + You can change how strict you want to make your calls by requiring the reads supporting the mode to represent 70%, 80%, etc of the total reads in a tag-defined read group (currently set to 66%)
  + This outputs text files **where you tell it** with STR Locus ID, unit number, and count of tag-defined read groups supporting that unit number into multimodes folder.