Ribosome profiling pipeline (Add command examples)

1. Import data to working directory on Monty (manually)
   - .gz files
   - 4 files per sample (if ran on NextSeq500, has 4 lanes/chip)
   - File name structure: <SampleName>_<indexID>_L00<1, 2, 3 or 4>_R1_001.fastq.gz
     - e.g: D_S6_L003_R1_001.fastq.gz
     - "L003" stands for "lane 3" on sequencing chip
     - "indexID" is the same for all 4 files in a sample
   - Import to directory named "rawfastqgz" (see "Resulting file system" below)

Actual pipeline starts here (output files used in downstream steps are marked in bold):

2. Make one fastq.gz file per sample
   - Output directory: "rawfastqgz"
   - Concatenate the 4 fastq.gz files mentioned above
   - Store as "**<ExperimentName>.<SampleName>.fastq.gz**" in directory "rawfastqgz"

3. Assess read quality
   - Software: FastQC
   - Output directory: "qualityCheck"
   - Input file(s):
     - "<ExperimentName>.<SampleName>.fastq.gz" (from step 2)
   - Output file(s):
     - "<ExperimentName>.<SampleName>.fastQC.xml"
   - Settings:
     - -t <*number of files processed*> (number of threads)
     - -o <*path to directory "qualityCheck"*> (output directory)
   - Example command:
     - `fastqc -t 3 -o <path to directory "qualityCheck"> inputFile1 inputFile2 inputFile3`

4. Trim away adapter sequences (if possible, run all files in parallel)
   - Software: Cutadapt
   - Output directory: "cutadapt"
   - Input file(s):
     - "<ExperimentName>.<SampleName>.fastq.gz" (from step 2)
   - Output file(s):
     - "**<ExperimentName>.<SampleName>.cutadapt.fastq.gz"**
     - "<ExperimentName>.<SampleName>.tooShort.fastq.gz"
     - "<ExperimentName>.<SampleName>.untrimmed.fastq.gz"

- “<ExperimentName>.<SampleName>.cutadapt.report.txt”
- Settings:
  - -a *<adapter sequence>* (sequence to be aligned to reads)
  - -O *6* (do trim read if alignment overlap between read and adaptor is less than 6)
  - -m *6* (discard reads shorter than 6 bases after trimming)
  - -n *3* (remove up to 3 adapter sequences per read)
  - -e *0.15* (maximum allowed error rate)
  - --too-short-output=<ExperimentName>.<SampleName>.tooShort.fastq.gz
  - --untrimmed-output=<ExperimentName>.<SampleName>.noAdapter.fastq.gz
  - -o **<ExperimentName>.<SampleName>.cutadapt.fastq.gz**
  - > <ExperimentName>.<SampleName>.cutadapt.report.txt”
- Example command:
  - `cutadapt -a AGATCGGAAGAGCACACGTCT -O 6 -m 6 -n 3 -e 0.15 --too-short-output=PPE5.B.tooShort.fastq.gz --untrimmed-output=PPE5.B.noAdapter.fastq.gz -o` **`PPE5.B.cutadapt.fastq.gz`** `PPE5.B.fastq.gz > PPE5.B.cutadapt.report.txt`

5. Assess footprint-length distribution
   - Software: FastQC
   - Output directory: “lengthDistr”
   - Input files(s):
     - “<ExperimentName>.<SampleName>.cutadapt.fastq.gz” (from step 4)
   - Outputfile(s):
     - “<ExperimentName>.<SampleName>.cutadapt.fastQC.xml”
   - Settings:
     - -t *<number of files processed>* (number of threads)
     - -o *<path to directory “quality”>* (output directory)
   - Example command:
     - `fastqc -t 3 -o <path to directory “lengthDistr”> inputFile1 inputFile2 inputFile3`

6. Trim ends with low-quality base calls (if possible, run all files in parallel)
   - Software: Sickle
   - Output directory: “highQuality”
   - Input file(s):
     - “<ExperimentName>.<SampleName>.cutadapt.fastq.gz” (from step 4)
   - Output file(s):
     - “**<ExperimentName>.<SampleName>.quality.fastq**”
     - “<ExperimentName>.<SampleName>.quality.report.txt”
   - Settings:

- se (single-end-sequenced reads)
- -f <ExperimentName>.<SampleName>.cutadapt.fastq.gz (input file)
- -t *sanger* (quality score type)
- -q *20* (quality score threshold)
- -l *6* (discard reads shorter than 6 bases after trimming)
- -o **<ExperimentName>.<SampleName>.quality.fastq** (output file)
- > <ExperimentName>.<SampleName>.quality.report.txt (save report in text file)

- Example command:
  - `sickle se -f PPE5.B.cutadapt.fastq.gz -t sanger -q 20 -l 6 -o **PPE5.B.quality.fastq** > PPE5.B.quality.report.txt`

7. Remove rRNA and tRNA sequences
   - Software: Bowtie1
   - Output directory: “tANDrRNAremoval”
   - Input file(s):
     - /ssd/common/tools/bowtie1-1.1.2/indexes/rRNAtRNA.syn6803.NC000911 (reference sequence)
     - “<ExperimentName>.<SampleName>.quality.fastq” (from step 6)
   - Output file(s):
     - “**<ExperimentName>.<SampleName>.tANDrRNAdeplete.fastq**”
     - “<ExperimentName>.<SampleName>.tANDrRNA.sam”
     - “<ExperimentName>.<SampleName>.tANDrRNAdeplete.report.txt”
   - Settings:
     - -a (report all alignments for a read)
     - --best (report alignment for a read in best-to-worst order)
     - --strata (report only the best alignment, i.e. least mismatches)
     - -t (print searching times)
     - -n *2* (maximum amount of mismatches in seed alignment)
     - -l *28* (seed length)
     - -S (report alignments in SAM format)
     - -p *10* (number of threads)
     - --un **<ExperimentName>.<SampleName>.tANDrRNAdeplete.fastq** (nonaligned reads)
     - 2> <ExperimentName>.<SampleName>.tANDrRNAdeplete.report.txt
   - Example command:
     - `bowtie -a --best --strata -t -n 2 -l 28 -S -p 10 --un **PPE5.B.tANDrRNAdeplete.fastq** /ssd/common/tools/bowtie1-1.1.2/indexes/rRNAtRNA.syn6803.NC000911 PPE5.B.quality.fastq PPE5.B.tANDrRNA.sam 2> PPE5.B.tANDrRNAdeplete.report.txt`

8. Map reads to the genome (if possible, run all files in parallel)

- Software: Bowtie1
- Output directory: “mapped”
- Input file(s):
  - “/ssd/common/tools/bowtie1-1.1.2/indexes/syn6803.NC_000911 (reference sequence)
  - “<ExperimentName>.<SampleName>.tANDrRNAdeplete.fastq” (from step 7)
- Output file(s):
  - “**<ExperimentName>.<SampleName>.mapped.bwt1**”
  - “<ExperimentName>.<SampleName>.notmapped.fastq”
  - “<ExperimentName>.<SampleName>.moremapped.fastq”
  - “<ExperimentName>.<SampleName>.report.txt”

- Settings:
  - -a (report all alignments for a read)
  - --best (report alignment for a read in best-to-worst order)
  - --strata (report only the best alignment, i.e. least mismatches)
  - -m *1* (discard non-unique alignments, maximum 1 alignment/read allowed)
  - -n *2* (maximum amount of mismatches in seed alignment)
  - -l *28* (seed length)
  - -t (print searching times)
  - -p *10*
  - --un <ExperimentName>.<SampleName>.notmapped.fastq
  - --max <ExperimentName>.<SampleName>.moremapped.fastq
- Example commmand:
  - `bowtie -a --best --strata -m 1 -n 2 -l 28 -t -p 10 --un PPE5.B.notMapped.fastq --max PPE5.B.moreMapped.fastq /ssd/common/tools/bowtie1-1.1.2/indexes/syn6803.NC_000911 PPE5.B.tANDrRNAdeplete.fastq` **`PPE5.B.mapped.bwt`** `2> PPE5.B.report.txt`

9. Count the number of reads on read-occupied positions in genome (“center-weighted”) (if possible, run all files in parallel)

- Script: “/ssd/jan/ribprof/seqdata/scripts/readCountScript.max48.sup2.py”
- Output directory: “readcount”
- Input file(s):
  - “<ExperimentName>.<SampleName>.mapped.bwt” (from step 8)
- Output file(s):
  - “**<ExperimentName>.<SampleName>.readCount.p**”
  - “**<ExperimentName>.<SampleName>.readCount.m**”
- Command:
  - `python readCountScript.max48.sup2.py`

10. Calculate total number of mapped reads
    - Script: “/ssd/jan/ribprof/seqdata/scripts/totalNbrMappedReadsScript.sup3.py”
    - Output directory: “readcount”
    - Input file(s):
        - “<ExperimentName>.<SampleName>.readCount.p” (from step 9)
        - “<ExperimentName>.<SampleName>.readCount.m” (from step 9)
    - Output file(s):
        - “**<ExperimentName>.<SampleName>.totalNbrMappedReads**”
    - Command:
        - `python totalNbrMappedReadsScript.sup3.py`

11. Calculate RPM (reads per Million mapped reads) on read-occupied positions in genome
    - Script: “/ssd/jan/ribprof/seqdata/scripts/RPMscript.sup6.py”
    - Output directory: “RPM”
    - Input file(s):
        - “<ExperimentName>.<SampleName>.readCount.p” (from step 9)
        - “<ExperimentName>.<SampleName>.readCount.m” (from step 9)
        - “<ExperimentName>.<SampleName>.totalNbrMappedReads” (from step 10)
    - Output file(s):
        - “**<ExperimentName>.<SampleName>.RPM.p**”
        - “**<ExperimentName>.<SampleName>.RPM.m**”
    - Command:
        - `python RPMscript.sup6.py`

12. Complete RPM vs. genome-position list by assigning “0” to all unoccupied positions
    - Script: “/ssd/jan/ribprof/seqdata/scripts/RPMcompleteScript.NC000911.sup7.py”
    - Output directory: “RPM”
    - Input file(s):
        - “<ExperimentName>.<SampleName>.RPM.p (from step 11)
        - “<ExperimentName>.<SampleName>.RPM.m (from step 11)
    - Output file(s):
        - “**<ExperimentName>.<SampleName>.RPM0.p**”
        - “**<ExperimentName>.<SampleName>.RPM0.m**”
    - Command:
        - `python RPMcompleteScript.NC000911.sup7.py`

13. Count the number of reads on every gene
    - Script: “/ssd/jan/ribprof/seqdata/scripts/readsPerGeneScript.sup4.edited.py”
    - Output directory: “readsPerGene”
    - Input file(s):
        - “<ExperimentName>.<SampleName>.readCount.p” (from step 9)

    - “<ExperimentName>.<SampleName>.readCount.m” (from step 9)
    - “/ssd/jan/ribprof/seqdata/genelists/syn6803.NC000911.genelist_p”
    - “/ssd/jan/ribprof/seqdata/genelists/syn6803.NC000911.genelist_m”
- Output file(s):
    - “**<ExperimentName>.<SampleName>.readsPerGene.p**”
    - “**<ExperimentName>.<SampleName>.readsPerGene.m**”
- Command:
    - `python readsPerGeneScript.sup4.edited.py`

Settings variables (= default)

- Step 2 (general):
    - <ExperimentName> =
    - <SampleName> = <SampleName> in Illumina output file
- Step 4 (Cutadapt):
    - -a = AGATCGGAAGAGCACACGTCT
    - -O = 6
    - -m = 6
    - -n = 3
    - -e = 0.15
- Step 6 (Sickle):
    - -q = 20
    - -l = 6
- Step 7 (Bowtie1):
    - -n = 2
    - -l =28
    - -p = 10
- Step 8 (Bowtie1):
    - -m = 1
    - -n = 2
    - -l =28
    - -p = 10

Resulting file system

“YYYY.MM.DD.<Experiment name>”

- “rawfastqgz”
- “qualityCheck”
- “cutadapt”
- “lenghtDistr”

- “highQuality”
- “tANDrRNAremoval
- “mapped”
- “readcount”
- “RPM”
- “readsPerGene”