* ***All code + data should be on the server in /local/data/course/project/groups/[groupname]/FINAL***

**Run python script to copy, unzip and trim the files and perform fastq quality control on untrimmed and trimmed files:**

korte039@altschul:/local/data/course/project/groups/periwinkles/FINAL$ python3 unzipping\_trimming.py

**Check the results of the quality control by fastqc:**

korte039@altschul:/local/data/course/project/groups/periwinkles/FINAL/trimmedRNASeqReads/fastqc\_res$ firefox [file].html  
  
**Alignment with Bowtie2:**

**Run script “map\_reads\_forward.py”.**

The script will check for the presence of an earlier made .sam file and ask if this one should be used (y) or a new one should be created (n), in which case the current one will be converted into a backup.

The script will also check for the presence of index files for *Catharanthus roseus*, and create new ones if not found.

Lastly, the RNA-seq reads will be mapped to the transcriptome of the genome V2 and the .sam will be converted to a .bam.

Input:

The index files, symbolic link to the transcriptome and the .sam and .bam file can be found in /local/data/course/project/groups/periwinkles/FINAL/bowtie\_files

The true location for the transcriptome is

/local/data/course/project/genomes/Catharanthus\_roseus/genome\_version2/cro\_v2.transcripts.fasta

The reads can be found in /local/data/course/project/groups/periwinkles/FINAL/trimmedRNASeqReads

Output:

Newly created .sam, .bam and index files will be stored in

/local/data/course/project/groups/periwinkles/FINAL/bowtie\_files

**Running kallisto:  
Run python script “kallisto.py” to create an index and run kallisto quant on single end reads, also parses the output files and ultimately returns a file with the rounded tpm values for every condition for every target\_id.**Input:  
index: /local/data/course/project/genomes/Catharanthus\_roseus/genome\_version2/cro\_v2.transcripts.fasta  
quant:  
/local/data/course/project/groups/periwinkles/FINAL/trimmedRNASeqReads  
output:  
index: /local/data/course/project/groups/periwinkles/FINAL/kallisto\_results/kallisto\_index.idx

quant: /local/data/course/project/groups/periwinkles/FINAL/kallisto\_results/kallisto\_results\_  
rounded tpm values file: /local/data/course/project/groups/periwinkles/FINAL/deseq\_input.txt

**Making a multi-fasta file for blast:  
Run python script “blast.py” to extract alignment data from bowtie2 and write it to multiple multi-fasta files to be used in blastx.**input:  
bowtie sam file: /local/data/course/project/groups/periwinkles/FINAL/bowtie\_files/bt2\_forward.sam  
file with all differentially expressed genes:  
 /local/data/course/project/groups/periwinkles/FINAL/DiffEpressed.txt  
file with all differentially expressed genes without CK3:  
 /local/data/course/project/groups/periwinkles/FINAL/DiffExpressedNoCk3.txt  
file with all differentially expressed genes CK vs ET:  
 /local/data/course/project/groups/periwinkles/FINAL/CKvsETgenes.txt  
file with all differentially expressed genes CK vs MJ:  
 /local/data/course/project/groups/periwinkles/FINAL/CKvsMJgenes.txt  
output:  
multi-fasta files: /local/data/course/project/groups/periwinkles/FINAL/blast/input\_blast\_  
  
**Parsing the results from blastx:  
Run python script “blastx\_parser.py” to extract the accession numbers of all the top hits from the csv file gained from the blastx run.**input:  
/local/data/course/project/groups/periwinkles/FINAL/CK\_vs\_ET\_blastx\_result.csv  
/local/data/course/project/groups/periwinkles/FINAL/CK\_vs\_MJ\_blastx\_result.csv  
output:  
/local/data/course/project/groups/periwinkles/FINAL/CK\_vs\_ET\_blastx\_parsed\_result.txt  
/local/data/course/project/groups/periwinkles/FINAL/CK\_vs\_MJ\_blastx\_parsed\_result.txt