Quality and Index Swapping

Our goal is to look through the lane of sequencing generated from your library preps and determine the level of index swapping and undetermined index-pairs, before and after quality filtering of index reads. All work should be done on Talapas. Test dataset can be found in `/projects/bgmp/test_data`

Part 1 – Quality Score Distribution per-nucleotide

1. You will receive four fastq files generated on the Illumina HiSeq 4000. Two of the fastq files will have very short "reads" which are your indexes. The other two fastq files will have longer reads which are your paired end reads containing your (Stacey's) biological data of interest. Using your old python script(s), generate a per-nucleotide quality distribution of each position in each of the four files.
2. By eye, compare the quality of your index reads versus your pair-end reads.

Part 2 – Quality filter based on index reads and split files by index

All your work in this section should be completed using the queuing system on Talapas. (See Nick's lecture notes on Talapas and https://hpcrcf.atlassian.net/wiki/display/TCP/How-to+Submit+a+Job to remind yourself how the queuing system works).

1. Write a python script which takes as input from the command line: four fastq file names, an integer which will represent the minimum quality score cutoff, and a fifth text file name which will have all known indexes (one per line).
    a. CHALLENGE – include logic which will allow you to pass either zipped or unzipped files to your script.
2. Your script will read in and store all known indexes from the `index.tsv` file in some data structure. You will use these indexes to compare each fastq record against, and retain counts of how many of each combination of indexes is observed.
    a. Recall how index swapping works. Build a data structure with every combination of known indexes, and initialize the count to 0 for each.
3. Your script will then look through both index reads. If ALL of the nucleotides of both index reads are above (or equal to) your minimum quality score, that record should be retained (output to new fastq files; see below).
4. When a record is to be retained, compare each index read to the list of possible indexes from part 2.2.
    a. If the observed indexes match a combination of indexes in your data structure, increment the counter for that index-pair, and add those indexes to the read name of all four fastq entries for that record. For example:
    `@NS500451:204:HH7GHBGXY:1:11101:7398:1048 4:N:0:0`
    would be renamed to
    `@NS500451:204:HH7GHBGXY:1:11101:7398:1048_index1_index2`
    b. Output those records to four fastq files that each indicate which index-pair was observed (i.e. fastq files named "`index1_index2_R1.fastq`", "`index1_index2_R2.fastq`", etc).

c. When the observed indexes don't match any index-pair from part 2.2, add the indexes to the read names, but output those to two fastq files named "`Undetermined_index_pair_R1.fastq`", "`Undetermined_index_pair_R2.fastq`"

Part 3 – Output Comparison

1. Run your script on the raw, unfiltered Illumina output, as well as with two different minimum quality cutoffs (to be determined when the lane of sequencing is finished).
2. Create clearly labeled per-nucleotide quality distribution plots for each dataset.
3. Create a table (in Excel or something else) with one row per index-pair, and one column per quality cutoff.
4. Think about each quality cutoff used and what those quality scores should mean. Does removing low quality records proportionally reduce the amount of index swapping? Does it proportionally reduce the number of "undetermined" index pairs? Why or why not?