Demultiplexing

Assignment the First

7-26-2023

We will be de-multiplexing sequences with dual-matched indexes (same index on both ends of the read).
N=unknown and put the read into “unknown” category for the simplest version of quality filtering.
Index hopping has occurred when index reverse complement does not match but is actually another index from the list of provided indexes known to be in the multiplexed samples.

Notes:
With open files as fhread1, fhread2, fhindex1, fhindex2
If startswith (@) in fh:
Read1 = fhread1.readline
Read 2 = fhread2.readline
Index 1 = fhindex1.readline
Index 2 = fhindex2.readline
Compare index to list of indexes
Append to header: “index - RC of index”
Eg, TCTTCGAC-TTCTCGAC (index seq and reverse complement of index seq: easy to look at and see that they match).

Bash commands for initial file exploration:

```
>>zcat <filename> | head -4 | grep -A1 "^@" --no-group-separator | grep -v "^@" | wc -m
102
```

102 -1(newline character) = 101nt per read.
9-1(newline character)=8nt per index.

Determine the encoding (and Q-score binning):
Illumina 1.8+ uses Phred+33 encoding for quality scores (J scores are present).

Made test files

```
>> zcat <filename> | head -40 > <testfilename>
```

Installed matplotlib to my bgmp_py311 conda environment so I can run matplotlib on it.

First sbatch didn’t work because I didn’t unzip the files. Added gzip open to .py
Second sbatch didn’t work because I was trying to write to the bgmp shared folder. Oops.

Ran a third time and it worked.
Ran a fourth time to include the read name specifically in the title of my plots.

Resource summary:
For R1
Maximum resident set size (kbytes): 68000
Percent of CPU this job got: 99%
Elapsed (wall clock) time (h:mm:ss or m:ss): 1:58:31
Exit status: 0

For R2
Maximum resident set size (kbytes): 70352
Percent of CPU this job got: 98%
Elapsed (wall clock) time (h:mm:ss or m:ss): 16:09.91
Exit status: 0

For R3
Maximum resident set size (kbytes): 65888
Percent of CPU this job got: 98%
Elapsed (wall clock) time (h:mm:ss or m:ss): 15:51.88
Exit status: 0

For R4
Maximum resident set size (kbytes): 70264
Percent of CPU this job got: 99%
Elapsed (wall clock) time (h:mm:ss or m:ss): 1:58:10
Exit status: 0

Assignment the Third

Environment: bgmp_py311
Scripts: demux.srun, demux.py
Required modules: matplotlib.pyplot 3.7.2, bioinfo.py v0.6, numpy 1.25.2
Collaborators: Temi, Evelyn, Tam, Dove

**Note: to check versions for matplotlib and numpy, open an interactive python session in the terminal:*

```
>>> import numpy
>>> numpy. __version__
'1.25.2'
>>> import matplotlib
>>> matplotlib. __version__
```

`'3.7.2'`
8-2-2023

Wrote rev_comp and makedict functions and added them to my bioinfo module.
Wrote demux.py (working draft)

8-3-2023

Goals for today:
Fix “B1” to actual index string - done, this is typical nomenclature for demultiplexing
Make index dictionary from the indexes.txt file - done
Make sample:index dictionary from indexes.txt - done
Add “sample” section for “Percentage of reads from each sample” - done
Heat map graph output file - done
Add functionality for zipped files and run large files (at end) - done

Not completed:
Comment and annotate code for readability (what are all my dictionaries??)
Nicer looking markdown file with results?

Began my first sbatch run on Talapas (demux.srun is a wrapper that runs demux.py) with a quality cutoff of 33 for indexes (job 24573). Exit status 1; division by 0 error because I have NO index pairs identified. I fixed the code so that it can still return 0 percent; and I also think my cutoff is too restrictive. I ran it again with 30 instead of 33.

Resource summary (job 24577):
Maximum resident set size (kbytes): 247876
Percent of CPU this job got: 87%
Elapsed (wall clock) time (h:mm:ss or m:ss): 1:20:36
Exit status: 0

I’ll have to run this again anyway to generate my plots and with a nicer markdown file.
I installed numpy to the bgmp_py311 environment to do this (matplotlib is already on there)
`conda install numpy -c conda-forge`

Anyway looks good so far, I’m generating the correct number of files and they’re all being populated. Summary stats look reasonable, but really conservative with my cutoff:

Demultiplexing Summary
Hopped: 330975
Matching: 226,715,602
Unknown: 136,200,158
Total: 363246735
`>>> zcat <filename> | wc -l`

`1452986940`

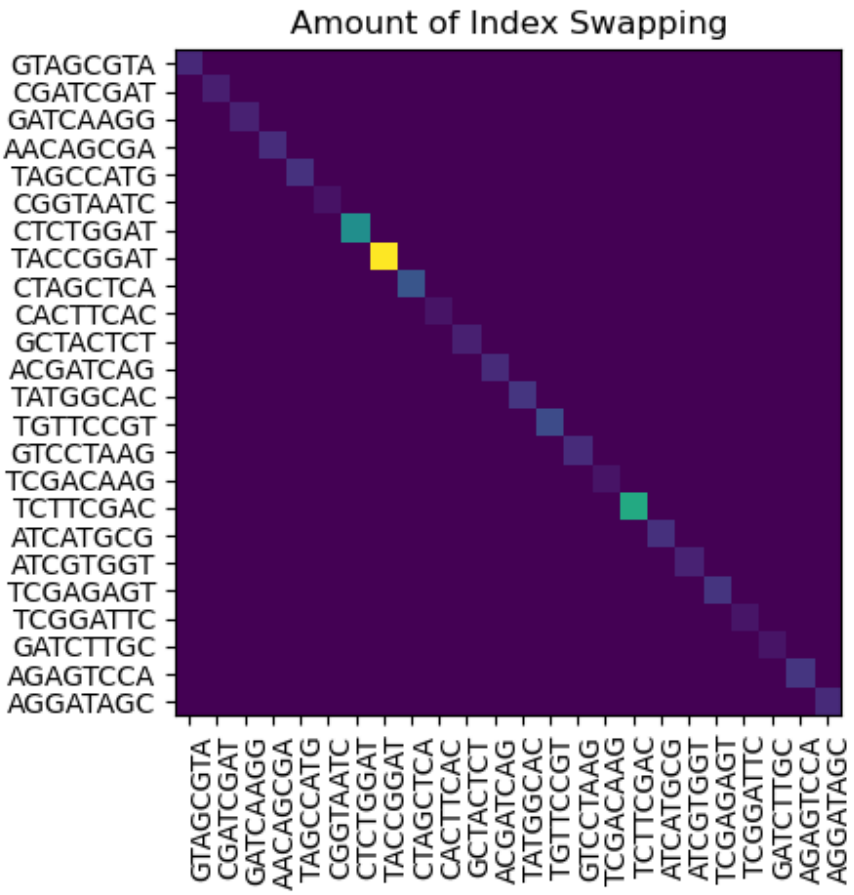
Divide by 4 lines per record: 363246735 = 363246735. Confirm the same number of records as my sequence files? YES. I'm capturing everything.

Sample	Read index	Percent
1	GTAGCGTA	2.546996744
2	CGATCGAT	1.869237919
3	GATCAAGG	2.041410454
4	AACAGCGA	2.808868884
6	TAGCCATG	3.152916225
7	CGGTAATC	1.055516682
8	CTCTGGAT	10.81312525
10	TACCGGAT	21.91595001
11	CTAGCTCA	5.749190124
14	CACTTCAC	1.136960129
15	GCTACTCT	1.897230699
16	ACGATCAG	2.617167918
17	TATGGCAC	3.374920796
19	TGTTCCGT	5.050624615
21	GTCCTAAG	2.734762383
22	TCGACAAG	1.166333493
23	TCTTCGAC	13.27198514
24	ATCATGCG	3.055752202
27	ATCGTGGT	2.086318259
28	TCGAGAGT	3.285204871
29	TCGGATTC	1.267808644
31	GATCTTGC	1.162836601
32	AGAGTCCA	3.353392062
34	AGGATAGC	2.585489904

I really need to clean up my local file organization, it looks horrible.
I also need to link Talapas to my github.
Sidenote:
https://bmcgenomics.biomedcentral.com/articles/10.1186/s12864-018-4703-0

Graphs for index swapping ideas; these are way too in depth for current projects (identifying swapping hotspots on the flow cell). I think this would be really cool if I had the time to do it, I'll have to note this and revisit it in a later project.

8-4-2023
Ran sbatch for producing a heatmap graph output in addition to the .md file in Demultiplex/demux_8-4-2023.



Haha. It is an ineffective graph for what I wanted it to show. All that shows up are the correctly paired indexes, since there are so many more of them than hopped indexes. Even then, the variation in the amount of each sample is so large that the paired indexes with less representation in the overall reads are very dim.

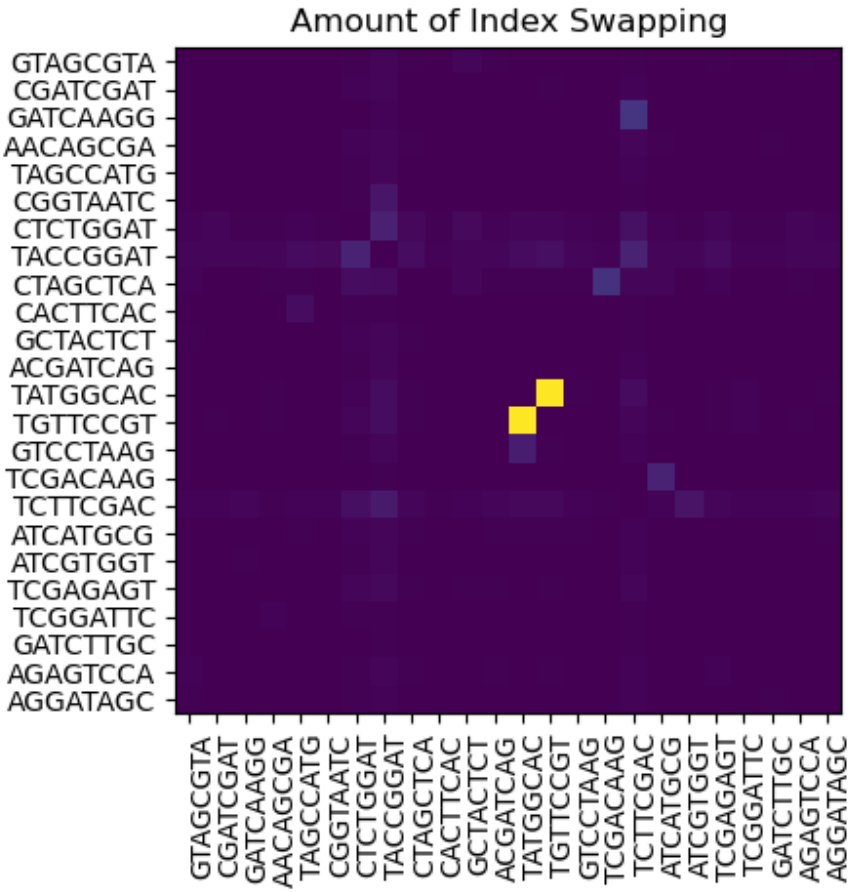
I think for overall amount of index swapping, I'll have to calculate swapped / (swapped + paired). The overall amount would be 330975/(330975+226715602) = 0.146%.
I need to add this to my .md file.

8-6-2023

Fixed the heatmap so it only displays indexes that are hopped

Fixed the output file so it only displays indexes that are hopped in tsv format
Fixed the .md file to have the overall percent of index swapping
Ran another sbatch.

This time the heatmap is much more useful. It clearly highlights the two indexes that swapped with each other the most:



Running a final time to generate an updated markdown file.

8-10-2023
The overall amount of index swapping with a cutoff of 2 (inclusive) should be a total of 707740 - I will rerun mine with a new cutoff to see if that works.
With a cutoff of 2 (which I input as -c 3, because my code is exclusive for quality cutoff), it is returning 707740 swapped indexes. Yay!

Final submission has a cutoff of 30. (Anything below 30 not counted).