# Deduper Part One

*David Degnan*

*October 19th, 2018*

## 1. Define the problem

Illumina sequencing libaries contain duplicates due to PCR biases, as certain sequences are more likely to be amplified than others. This creates a problem during downstream analyses (differential expression, for example) as certain reads will be over-represented. These PCR duplicates need to be removed.

## 2. Write examples

Examples have been moved to the end of the file for R Markdown readability.

## 3. Develop your algorithm using psuedocode

*Step One:* The files will undergo Samtools sort and be organized by chromosome and position.

*Step Two:* A 30 line "sliding window" will be generated and stored as a dictionary (key will be line number, and the line itself will be the value, split with the python .split() function). Reads will either be written to a discard file (Step Three) or written to a new file (Step Four). This will continue until the end of the file is reached.

*Step Three:* 30 lines will be read at a time. A line will be written to a discard file (with "_discarded" at the end of the file name before the extension) if: * If it contains any N's in the UMI. * The chromosome number, start position, and UMI are the same as any of the following 30 lines. * Note that start positions will be adjusted according to soft clipping, insertion, and deletion. This information will be taken from the cigar string. * The bitwise flag will keep track of which strand it's on. If it's only on a different strand once, it is not a duplicate and will not be written to the _discarded file.

*Step Four:* If no conflicts are found, the reads will be written to an output file that will contain the name of the input file with a "_deduped" at the end of it (before the extension). *Step Five:* The sliding window will then slide down to the next read. In other words, the first entry in the dictionary will become the 31st line of the file (this can be tracked with a modulus, ie 31 % 30 = 1). The second entry in the dictionary will then be compared to all the other lines, testing for any conflicts (Step Three) and if none found, being written to the "_deduped" file (Step Four). This continues for each line of the file until the end is reached.

## 4. Determine high level functions

### STEP ONE: Samtools Sort

Use samtools to sort the file

```
purge the modules
load samtools
samtools sort input file
convert sorted input file to a sam file
```

## STEP TWO: Fill the sliding window

Open input file, and generate the two output files

```
open all three of the files, allowing an argparse for the input file
```

Initialize sliding window and read in the first thirty lines

```
intialize an open window
for i in range(30):
  entry number is key and line is file
```

## STEP THREE/FOUR: Start testing

UMIs with any N's will need to be tossed

```
define test UMI as a function that
  returns true if the UMI has any Ns
  returns false if the UMI does not
TEST: UMI with an N and an UMI without
```

Bitwise flag checker. If line passes, this function adds an R to UMI. A singular R will be written to deduped, and multiple will be written to duplicate.

```
if the 5th bit is on:
  then the read is on the reverse strand
  add an R to the UMI

TEST: With 5th bit set and without
```

Adjust start position based on soft clipping

```
given the number preceeding the s:
  add to the position number whatever preceeded the s

TEST: given an S adjust the position
```

If necessary, adjust for insertions or deletions.

```
for deletion:
  add to the position number
for insertion:
  subtract from the position number

TEST: given an I or a D, adjust the position
```

Given chromosome, start position, and UMI, determine if they all match any other sequence

```
if chromosome matches any in sliding window
and start position matches the same line in the sliding window
and UMI matches the same line in the sliding window:
  write file to duplicate
else:
  write file to deduped

TEST: give matches and ones that do not match
```

**STEP FIVE: Read in next line**

```
position in window = line number % 30
save line into that position as value
```

## 2. Examples

Table 1: Input Sam File

| QNAME | FLAG | NAME | POS | MAPQ | CIG | R | P | LEN | SEQ | QUAL |
|---|---|---|---|---|---|---|---|---|---|---|
| NS5:S:I:1:N:T:I:GTGATGTC | 0 | 2 | 3286131 | 36 | 5M | * | 0 | 0 | CCCCC | ORI |
| NS5:S:I:1:N:T:I:GTGATGTC | 0 | 2 | 3286131 | 36 | 5M | * | 0 | 0 | CCCCC | DUP |
| NS5:S:I:1:N:T:I:GTGATGTC | 0 | 2 | 3286131 | 36 | 2S3M | * | 0 | 0 | CCCCC | ORI |
| NS5:S:I:1:N:T:I:GTGATGTC | 0 | 2 | 3286131 | 36 | 5M | * | 0 | 0 | CCCCC | DUP |
| NS5:S:I:1:N:T:I:GTGATGTC | 0 | 2 | 3286131 | 36 | 5M | * | 0 | 0 | CCCCC | DUP |
| NS5:S:I:1:N:T:I:GTGATGTC | 16 | 2 | 3286131 | 36 | 5M | * | 0 | 0 | AAAAA | ORI |
| NS5:S:I:1:N:T:I:GTGATGTC | 16 | 2 | 3286131 | 36 | 5M | * | 0 | 0 | AAAAA | DUP |
| NS5:S:I:1:N:T:I:GTGATGTC | 0 | 3 | 5 | 36 | 5M | * | 0 | 0 | CTCTC | ORI |
| NS5:S:I:1:N:T:I:GTGATGTC | 16 | 3 | 5 | 36 | 5M | * | 0 | 0 | AGAGA | ORI |
| NS5:S:I:1:N:T:I:GTGATGTC | 16 | 3 | 5 | 36 | 5M | * | 0 | 0 | AGAGA | DUP |
| NS5:S:I:1:N:T:I:GTGATGTC | 16 | 3 | 5 | 36 | 1S4M | * | 0 | 0 | GAGA | ORI |

Table 2: Output Deduped File

| QNAME | FLAG | NAME | POS | MAPQ | CIG | R | P | LEN | SEQ | QUAL |
|---|---|---|---|---|---|---|---|---|---|---|
| NS5:S:I:1:N:T:I:GTGATGTC | 0 | 2 | 3286131 | 36 | 5M | * | 0 | 0 | CCCCC | ORI |
| NS5:S:I:1:N:T:I:GTGATGTC | 0 | 2 | 3286131 | 36 | 2S3M | * | 0 | 0 | CCCCC | ORI |
| NS5:S:I:1:N:T:I:GTGATGTC | 16 | 2 | 3286131 | 36 | 5M | * | 0 | 0 | AAAAA | ORI |
| NS5:S:I:1:N:T:I:GTGATGTC | 0 | 3 | 5 | 36 | 5M | * | 0 | 0 | CTCTC | ORI |
| NS5:S:I:1:N:T:I:GTGATGTC | 16 | 3 | 5 | 36 | 5M | * | 0 | 0 | AGAGA | ORI |
| NS5:S:I:1:N:T:I:GTGATGTC | 16 | 3 | 5 | 36 | 1S4M | * | 0 | 0 | GAGA | ORI |

Table 3: Output Duplicate File

| QNAME | FLAG | NAME | POS | MAPQ | CIG | R | P | LEN | SEQ | QUAL |
|---|---|---|---|---|---|---|---|---|---|---|
| NS5:S:I:1:N:T:I:GTGATGTC | 0 | 2 | 3286131 | 36 | 5M | * | 0 | 0 | CCCCC | DUP |
| NS5:S:I:1:N:T:I:GTGATGTC | 0 | 2 | 3286131 | 36 | 5M | * | 0 | 0 | CCCCC | DUP |
| NS5:S:I:1:N:T:I:GTGATGTC | 0 | 2 | 3286131 | 36 | 5M | * | 0 | 0 | CCCCC | DUP |
| NS5:S:I:1:N:T:I:GTGATGTC | 16 | 2 | 3286131 | 36 | 5M | * | 0 | 0 | AAAAA | DUP |
| NS5:S:I:1:N:T:I:GTGATGTC | 16 | 3 | 5 | 36 | 5M | * | 0 | 0 | AGAGA | DUP |