Specification for pss-bam

Inputs:
1. a bam file of sequences aligned to a genome
2. a fasta file of the genome

Outputs:
text describing the observed the base context of alignment data (where reads start and stop) and mismatches of the aligned reads to the genome. An example of the output format is shown in the example below

```
### pss-bam.pl v 0.061
### /data/genomes/hs37d5.fa
### aln/JK774.M.bam
### Format of table:
### Counts of how often a read base and genome base were seen at
### each position in the aligned reads.
### First base is what was seen in the read.
### Second base is what was in the genome at that position.
### POS AA AC AG AT CA CC CG CT GA GC GG GT TA TC TG TT
### Forward read substitution counts and base context
-2 36643 0 0 0 0 32369 0 0 0 0 29212 0 0 0 0 31615
-1 61901 0 0 0 0 17967 0 0 0 0 36380 0 0 0 0 13591
0 26339 31 48 34 39 36885 16 4357 29 30 20335 50 156 58 23 41410
1 36474 21 43 31 37 29643 8 3374 28 21 21463 23 78 58 11 38525
2 35745 15 48 17 21 34994 15 2780 39 15 21430 25 40 53 10 34580
3 39707 22 54 20 24 32749 6 2045 36 17 22156 24 66 44 8 32851
4 38541 12 61 18 25 30517 11 1693 28 4 24692 42 26 46 12 34068
5 36180 11 53 16 23 30573 9 1714 24 20 25976 30 43 47 9 35082
6 36887 7 61 17 26 30608 12 1599 31 11 25618 31 38 57 13 34785
7 38081 11 64 12 16 30830 14 1469 33 8 25730 33 32 37 11 33424
8 37800 16 74 22 16 31638 12 1470 36 17 24974 28 39 38 9 33607
9 37526 9 56 18 16 32555 11 1401 26 14 24707 30 33 45 4 33352
10 36848 6 70 17 33 33062 22 1324 33 12 24008 27 30 50 7 34259
11 37043 14 49 14 19 32682 10 1261 25 11 24987 27 20 32 12 33602
12 36744 12 61 24 15 32753 12 1152 23 15 25792 36 19 32 10 33091
13 37632 9 53 24 23 31890 9 1148 18 11 25555 36 27 56 8 33294
14 37411 15 50 22 18 31651 9 1132 35 13 25972 26 28 42 2 33359


### Reverse read substitution counts and base context
14 37268 12 65 20 18 31898 10 1237 17 10 25397 33 26 43 8 33731
13 36912 10 49 18 17 31780 10 1144 25 13 26017 34 21 46 9 33677
12 36936 8 39 25 14 31733 14 1171 23 8 25793 37 27 44 10 33901
11 37492 12 56 24 19 31509 12 1263 30 21 25457 37 24 39 12 33791
10 37416 15 46 15 19 32136 13 1375 27 17 24942 33 23 32 8 33686
9 38236 10 44 19 16 31473 10 1447 26 10 24302 33 26 40 11 34112
8 38166 12 45 20 18 31127 10 1540 30 17 24342 29 31 55 11 34347
7 38413 18 61 30 19 30790 8 1507 29 14 23685 32 28 45 10 35120
6 37237 10 48 29 10 31711 13 1546 34 15 24606 29 22 54 9 34424
5 37437 15 64 24 23 31618 15 1526 22 17 26063 41 27 46 12 32855
4 35077 22 53 30 21 33029 12 1624 31 7 26532 23 22 40 12 33296
3 32160 14 48 19 18 36828 16 1921 29 16 25106 37 29 56 12 33523
2 31569 19 67 23 23 36087 14 1929 28 21 26605 26 39 55 13 33318
1 30656 16 57 22 30 39800 19 2809 29 24 25873 30 30 47 24 30374
0 26594 32 52 36 38 40976 11 3431 36 17 20397 28 35 58 12 38087
1 66660 0 0 0 0 15063 0 0 0 0 34419 0 0 0 0 13697
2 38037 0 0 0 0 27142 0 0 0 0 32762 0 0 0 0 31897
```

Program options to implement:

-r <region length; default = 15> The length, in basepairs, into the interior of alignments to report on
-l <minimum length of read to report; default = 0>
-L <maximum length of read to report; default = 250000000>
-q <map quality filter of read to report>
-U <upstream context base filter; first base before alignment must be one of these>
-D <downstream context base filter; first base after alignment must be one of these>
-m <only consider merged reads>

Code base to use:
fasta-genome-io: C code for reading and accessing a genome in fasta format. Loads all of genome into physical memory for fast access
sam-parse: C code for line-parsing SAM format alignment data

Program overview:
1. Parse input genome in fasta format
2. For each line of sam input
    Determine if the alignment is to be included
        MQ
        minimum length
        maximum length
        upstream base context
        downstream base context
        merged read (if -m)
        primary alignment
        no gaps, soft or hard masking (Cigar string is just one number and M)
    If it is included, retrieve corresponding genome segment, with context bases
    Reverse complement genome and read if alignment is reverse complement
    Add counts to relevant results data structure
3. Output results table