Part 2:

My test reference was randomly generated. I chose the reads manually (1 aligns twice, 1 aligns nowhere, 3 align once) and ordered them randomly in the reads file. Testing 5 reads against a reference of length 10 doesn't prove that my code will work elsewhere, but it was useful for debugging.

No I shouldn't expect an exact distribution. (1) The number of reads may not be thus divisible, and (2) I'm assigning reads into groups using a random number generator.

I played around with it a bit before I started working seriously, but once I did this took probably 2-3 hours, with breaks

Part 3:

In two tests on the third (largest) set, the distributions agree up to the 5 decimal places I have the code print. It is possible that everything would be working perfectly and the ratios would not exactly match: there could be reads created to align once which, by chance, align in two places.

It reliably takes about .005 seconds to run on the first data set, .25 seconds on the second, and 29.5 seconds on the third. This is growth of 50x and ~100x respectively, while in each case the number of reads (to which the reference length is proportional) went up by 10x. Even being extremely generous and assuming linearity at 30 seconds

for 60,000 reads, 30x coverage of a human genome with ~3,000,000,000 base pairs with reads of length 50 means 1,800,000,000 reads, which gives us a run time estimate of 900,000 seconds or 10.5 days. The real number is probably orders of magnitude higher.

Writing this part probably took something like an hour.