CMSC423_Proj4_Notebook_JosephWu

December 11, 2015

1 Joseph Wu, 112801489, CMSC423

2 Question 1 Answer

Question 1: Yes, from the results of the experiment, the reads seem to correspond to a gene sequence with a mutation at codon 275. The output of the code from PileUp.py gave me a list of nucleotide positions where there are mismatching positions. Since 274 (accounting for 0-based indexing) is the codon position and each codon is made up from 3 nucleotides, the target nucleotide position would be at 822. The output from the code in that area is:
816 A C:1
818 T G:1
822 C T:10 <- target
831 G C:1
837 T G:1
Nucleotide position 822 mismatches from ‘C’ to ‘T’ frequently in the reference. At the position of 822 in the reference sequence it has codon CAC which codes for histidine and switching position 822 from ‘C’ to ‘T’ gives us TAC which codes for the amino acid Tyrosine. This shows that the reads correspond to a gene sequence with a mutation that changes an amino acid from histidine to tyrosine.

```
In [ ]: # question_1_script
        # this is the code given from project 4 page with some extra code to get files

        from pileup import PileUp
        from approximate_matcher import ApproximateMatcher
        import sys

        # import the fasta files
        f = open(sys.argv[1], ’r’)
        g = open(sys.argv[2], ’r’)
        reference  = ""
        for line in f:
            reference += line.rstrip()
        reads = (g.readline()).rstrip().split()

        reference.rstrip()
        print reference[822:835]
        # initialize object
        am = ApproximateMatcher(reference)
        pileup = PileUp(reference)
        d = 3

        for read in reads:
                # find matching positions for a given read
                # assumes positions is a list (even if only a single match is found)
                # with matching positions
                positions = am.get_matches(read, d)
                if len(positions) > 0:
                        # add to pileup object
                        pileup.insert(positions, read)

        # prints out mismatching positions
        # output is:
        # (<position>, <reference_character>, [(<variant_character>,
        # <num_times_aligned>)])
        # argument filters mismatch by frequency in which variant character
        # is observe, e.g., .01 means variant character has to be seen at least
        # once for every 100 aligned nucleotides
        pileup.print_mismatches(.01)
```

3 Output from question_1_script

Note: Jupyter cut out the newlines after each output

```
16 A T:1 18 A G:1 46 C G:1 55 T C:1 57 G C:1 61 A G:1 70 T C:1 73 A T:1 75 A G:1 77 T G:1 81 A T:1 85
T C:1 87 A C:1 102 A C:1 108 T A:1 125 T C:1 127 A C:1 131 T G:1 137 T C:1 138 G A:1 141 A G:1 150 C
A:1 153 A T:1 155 C G:1 159 A C:1 172 A C:1 174 A T:1 180 T C:1 182 G T:1 185 A C:1 186 A C:1 193 C
A:1 196 A G:1 203 C T:1 207 A T:1 215 C G:1 216 A C:1 219 T A:1 220 T A:1 228 G A:1 233 G T:1 234 T
A:1 248 G A:1 253 T G:1 257 G T:3 268 C A:1 277 C A:1 280 T G:1 282 A G:1 284 T G:1 285 G T:1 289 G
A:1 290 G T:1 294 A T:1 296 A G:1 297 T C:1 298 A G:1 299 C A:1 300 A C:1 310 A G:1 311 C A:2 312 A
G:1 313 G T:1 315 A T:1 318 A T:1 320 A G:1 328 C A:1 348 A C:1 365 A G:1 371 C T:1 376 C A:1 385 G
A:1 390 A T:1,G:1 402 A C:1 404 T C:1 412 C G:1 418 T G:1 421 A C:1,G:1 426 A T:1 427 A T:1,G:1 455
G T:1 458 C A:1 459 C A:1 477 A C:1 479 C G:1 481 G A:1 493 A G:1 497 T C:1 502 C T:1 503 T G:1 504
C G:1 509 C G:1 514 C T:1 523 A C:1 524 G T:1 527 A C:1 528 G C:1,T:1 530 C G:1 531 G A:1,T:1 533 T
G:1 535 G A:1 538 C A:1 572 A T:1 573 A C:1,G:1 574 C A:1,G:1 580 G A:1 594 G A:1 597 A T:1 598 A
C:1 607 T A:1 612 G T:1 618 A C:1 620 G A:1 628 G T:1 630 A C:1,T:1 633 A G:1 641 C G:1 642 A G:1
665 T C:1 680 A G:1 685 C T:1 687 G T:1 692 T G:1 693 G C:1 697 G C:1 701 A G:1 703 A G:1 709 C T:1
712 G C:1 715 T C:1 717 A T:1 718 C T:1,G:1 719 T G:1 725 G T:1 729 G T:1 731 T C:1 736 C T:1 738 A
C:1 753 T A:1 757 A T:1 760 A G:1 781 G T:1 785 G A:1 789 G C:1 808 A G:1 810 G C:1 816 A C:1 818 T
G:1 822 C T:10 831 G C:1 837 T G:1 841 G T:1 842 T C:1 843 T G:1 845 T C:1 847 C A:1 851 T A:1 859
A C:2 860 A G:1 862 T C:1 863 C T:1 867 T C:1 871 T A:1,G:1 874 G C:1 880 A G:1 881 T G:1 883 A C:1
884 C A:1 887 G C:1 889 A G:1 890 T G:1 893 C A:1 896 G T:1 897 A C:2 901 G A:1,T:1 904 C T:1 906
T G:1 907 G A:1,C:1 911 G A:1 917 C T:1,G:1 923 G A:1,C:1 928 T C:1,G:1 930 G T:2 936 C A:1 940 T
G:1 941 A T:1 942 G A:1 948 A C:1 950 A G:1 953 C A:1 954 A G:1 957 G C:1 975 C A:1 990 A T:1 995
A C:1 997 G C:1 1000 G T:1 1003 G T:2 1005 G A:1 1010 A G:1 1012 T A:1 1014 T C:1 1015 C A:1,T:1
1019 T C:1 1021 A T:1 1025 A C:1 1027 C T:1 1030 A C:1 1035 G T:1 1036 T A:1 1045 T A:1 1050 T A:1
1055 A C:1 1057 A T:1 1058 C A:1 1059 G A:1 1068 G C:1 1070 T C:1 1076 A C:1 1077 G A:2 1079 G A:1
1081 G A:1 1082 A G:1 1083 A T:1 1085 T C:1 1087 A G:1 1094 T G:1 1095 A C:1 1096 G C:2,T:1 1097
T G:2 1101 A T:1 1102 G A:1,T:1 1104 A C:1 1108 G A:1,C:1 1110 T C:1 1120 T C:1 1121 T G:1 1122 T
G:1 1123 G C:1 1124 G T:1 1128 C T:1 1129 C A:1 1131 A G:1 1132 A T:1 1139 G T:2 1145 G C:1 1146 A
G:1 1151 C G:1 1152 A T:1 1155 A C:1,T:1 1162 C A:1 1163 A C:1 1166 A C:1 1168 A G:1 1170 C A:1,T:1
1173 G C:2 1177 T A:1 1179 G T:1 1185 A G:1 1200 G C:1,T:1 1205 T C:1,G:1 1206 A C:1,T:1 1211 G T:1
1214 T G:1 1219 T C:1 1226 T C:1 1227 C A:1,T:1 1228 C T:1 1229 A G:1 1234 T C:1 1244 G C:1 1246 A
T:1 1249 G A:1 1256 A C:1 1257 C T:1 1261 G T:1 1262 C A:1 1265 C T:1 1270 T G:1 1272 G C:1 1276 T
C:1 1278 A T:1 1281 A T:1 1287 C A:1,G:1 1290 C T:1 1292 C A:1,T:1 1302 A G:1 1303 C A:1 1305 A T:1
1313 T A:1 1314 A T:1 1316 C T:1 1319 G C:1 1322 C T:1 1325 C T:1 1335 T G:1 1347 A C:1 1352 C T:1
1353 A C:1 1363 G T:1 1374 G T:1 1383 G C:1 1392 T C:1
```