

TME 2.1	<pre> k = 8 freq = 5 top = 30  sequences = readFasta("Sequence_by_Peaks_3.fasta") getTopMotifs(removeLowComplexity(searchMotifs(k, sequences)), top) </pre> <p>✓ 0.0s</p> <p>Output exceeds the <a href="#">size limit</a>. Open the full output data <a href="#">in a text editor</a></p> <pre> 'TCCCCCCC': 15, 'ATTTGTAT': 13, 'TATTGTAT': 12, 'TTTCTTTT': 12, 'TTCCCCCC': 11, 'CGATGAGC': 11, 'TATTGTGA': 11, 'CTCATTGG': 11, 'CTGATTGG': 11, 'CTTCCCCC': 10, 'CTCCCCCC': 10, </pre>
TME 2.2	<pre> bestMotif, bestDistance, motifDist = MedianStringSearch(kmersValid, sequences[0:3], k) print (bestMotif, bestDistance) </pre> <p>✓ 8.6s</p> <p>GGCACC GA 1</p>
TME 3.1	<pre> def readFasta(genome, n):     sequence = []     file = open(genome, "r")     sequence = []     for s in file:         if s[0] != "&gt;":             sequence.append(s.strip().upper())     sequenceStr = "".join(sequence)     #sequence = [sequenceStr]     sequence = [sequenceStr[i:i+n] for i in range(0, len(sequenceStr), n)]     sequenceRet = [x for x in sequence if x]     return sequenceRet  genome = "Sequence_by_Peaks_3.fasta"  sequencesChip = readFasta(genome, n) t = len(sequencesChip) print (sequencesChip[8], t, k) revSequences = [reverseComplement(m) for m in sequencesChip] sequencesChip = sequencesChip + revSequences sequencesRet = [x for x in sequencesChip if len(x) == n] s, consensus = GreedyProfileMotifSearchIter(sequencesRet, k, 20) printTopMotifsScore(consensus, 10) </pre> <p>✓ 6.8s</p> <pre> CTGATCAGCA 4616 8 AATATTTA AATAAATT AATAAATT AAAATATA AAAATTAT TAAATTAT AATTATTT ATAATTTA AAATATTA TAAATTAT </pre> <p>AATATTTA AATAAATT AATAAATT AAAATATA AAATTATT TAAATTAT AATTATTT ATAATTTA AAATATTA TAAATTAT</p>
TME 3.2	Sevag le fait
<p>TME 4</p> <p>LOGO :</p> <p><a href="https://weblogo.berkeley.edu/cache/file21eJOC.png">https://weblogo.berkeley.edu/cache/file21eJOC.png</a></p>	<pre> tree = construct_tree(sequences) motifTest = generateRandomSequence(k, false) print(motifTest) motifOccur_sorted = inexact_match(motifTest, sequences, tree, v) print(motifOccur_sorted) </pre> <p>✓ 32s</p> <p>GCAGTT</p> <pre> ('GTATTT': 38, 'TCATTT': 37, 'GAAATT': 27, 'GCTATT': 26, 'AAAGTT': 26, 'GCAGAG': 22, 'GTAGTT': 21, 'TCTGTT': 21, 'GAAGTT': 20, 'GCAGAA': 20, 'GCATTT': 19, 'TCAATT': 19, 'ACAGTT': 19, </pre>