CAP5510 Fall 2019, HW#2

Question no 1:

Differential genes can be evaluated using various methods. Among them, **t-test** is a well-known one but individual **p-value** doesn't work well for multiple testing. The value of False Positive might be very high in case of multiplicity as it considers all genes with **p-value < α** to be differential genes. So, this is considered to be less useful in significant findings. To overcome this issue, another approach called '**Bonferroni correction**' is used.

- **Bonferroni Correction:**

Genes with **p-value** less than **α / N** (N is number of tests) is considered to be differential genes in this approach. In this method the number of possible differential genes is reduced compared to the **t-test** method by using the **cut-off** value (**α/ N**). This approach can be used to evaluate the difference of gene expression under two grouped conditions as it considers the number of tests. The advantages and disadvantages are given below:

Advantage of Bonferroni Method:

1. Very low value of False Positive.
2. It doesn't assume that the tests are independent
3. As it takes a very selective set of candidates using a cutting-off process, the chances of the selected candidates being differential is very high.

Disadvantage of Bonferroni Method:

1. As the number of candidates decreases, the value of False Negative increases
2. The chances of high-throughput decreases as the number of possible candidate becomes very less (very conservative).

- **FDR Correction (Benjamini & Hochberg):**

This approach tries to solve the issues of **t-test** and **Bonferroni** Method by forming a balance between them. It considers a small portion of False Positives to be tolerable and controls this value while identifying the differential genes and thus increases the number of possible candidates. In this method, the value of False Positive is lower than t-test and value of False Negative is lower than Bonferroni Method.

Approach:

1. In this method, it ranks all the genes according to their p-values in ascending order (gene with lowest p-value is ranked 1^{st}). Ex: $p(_1)<p(_2)\ldots<p(_N)$
2. Differential genes $p(_i)$ are tested (beginning with the p-value) and selected if they satisfy the following formula:

$$\mathbf{p(_i) < i*(q/N)\ ;} \qquad \mathbf{1 \le i \le k}$$

Advantages:

1. Control False Discovery Rate (FDR)
2. It might be able to detect more differential genes compared to the Bonferroni Method.

Disadvantages:

1. This procedure is only valid when the tests are individual.

Question 2:

(1) Profile Representation for the motif x is given below:

a	0	2	2	3	1	1	2	4
g	0	0	2	1	0	0	3	1
t	3	0	0	1	4	0	0	0
c	2	3	1	0	0	4	0	0

(2) The Consensus representation for this motif, its score and total distance is given below:

- **Consensus:** t c a a t c g a [or, t c g a t c g a]
- **Score:** 3 + 3 + 2 + 3 + 4 + 4 + 3 + 4 = 26

Sequence 1: t c g a t c a a, distance from Consensus: 2
Sequence 2: t c a a t c g a, distance from Consensus: 0
Sequence 3: c a a g t c g a, distance from Consensus: 3
Sequence 4: t c c t a c a a, distance from Consensus: 4
Sequence 5: c a g a t a g g, distance from Consensus: 5

- **Total Distance:** 2 + 0 + 3 + 4 + 5 = 14

(3) Comparison between Two Algorithms is given below:

- **Computational Equivalence:**

Motif finding algorithm tries to maximize score by considering all the possible start positions for all the sequences. If the length of the motif is L and there are t number of sequences each with length n then number of possible starting positions for each sequence will be (n – L+1). In total, (n – L +1)*t sets of starting positions. For each set of starting position, number of scoring operations will be L. So, total complexity for this algorithm will be L * (n – l + 1) * t which is really high and next to impossible for long sequences.

Median string problem tries to calculate the Hamming distance between the possible motifs and all possible starting position sets. It selects the motif which has the minimum distance from

some set of starting position. As it considers all possible combinations of motifs of length L, it takes 4^L time. Comparing a possible motif with all possible starting positions in a sequence of length n takes (n – L + 1) time. So, for t sequences the time should be t * (n – L +1). Overall complexity for this algorithm is 4^L * t * (n – L +1). This complexity is also large but compared to L (n – L + 1)t , this is much less.

Example:

Suppose, t = 8, n = 1000, L = 10

For Motif Finding Algorithm using Score Maximization, time complexity = 10 * (1000 – 10 +1) 8

= 9.3E24

For Median String Problem, time complexity = 4 10 *8 * (1000 – 10 + 1)

= 8.3E9

The computational difference between these two algorithms is clearly visible. With increasing value of t and n, this difference will become more and more visible. So, computationally Median String Problem is undoubtedly better than Motif finding problem using score maximization.

- **Comparison of Output:**

These two algorithms are equivalent in respect of optimality. The motif finding problem basically tries to minimize the Hamming Distance of consensus from the motifs by selecting the consensus with the maximum score. The Median substring problem also tries to select motif with minimum Hamming distance. It means their output might be similar given the same set of sequences.

Question 3:

(1) Transcription Factor (TF) influences the expression of gene by binding to a specific region (promoter region) of that gene. The site of the gene where the TF binds to is called Transcription Factor Binding Site (TFBS). TFBS is also known as motif (pattern) as for a certain TF, that region has a specific pattern that attracts the TF to that specific site and bind it there. A certain TF (protein) has a specific sequence. It can be binded to TFBS of different genes. But to bind some specific TF to a TFBS, the sequence of TFBS should follow certain pattern so that it can bind the TF. If a group of genes are known to be regulated by same TF, then their binding sites must have some similar properties. It means there has to be some similarity between the patterns of their motif. So looking for similar kind of sequences in the upstream positions of the group of genes who are controlled by the same TF might be helpful. So, the assumption that “overrepresented sequence patterns are potential motifs” is reasonable.

Limitations to this Assumption:

Computationally the motifs of a group of genes controlled by the same TF can be found by finding out a set of sequences in each gene which have maximum similarity to each other. But there are lots of limitations to finding the possible motifs for genes, such as:

(i) We do not know the length of the motif in each sequence

(ii) The motifs can be located anywhere in the regulatory regions of the genes. So, the starting positions of the motifs are unknown.

(iii) We don't know what the motif sequence might actually look like.

(iv) Motifs might vary from each other but we do not have any idea about how much they might vary from each other. Also, the hamming distance between different motifs might not be same, it might also vary.

(v) Even if we consider that we know the motif length (L), the motif finding algorithm that tries to maximize consensus score will take around $O(L^* (n-L+1)^t) = O(Ln^t)$ time which will take billions of years to complete it. Here, n represents length of each DNA sequence and t represents number of DNA sequence. Another approach, Median string problem takes $O(4^L)$ time, this is better than the previous one but still very large. So, time complexity for finding motifs using computational method is a big issue at present.

(vi) Time complexity can be reduced if we compromise the optimality of the solution. For example, Greedy Motif Search works faster but it cannot guarantee optimal solution. Also, output varies based on the order of sequences (which ones we are considering at first). So, this might result in high value of False Positive.

(2) Given a group of sequences, **MEME** software can identify overrepresented sequence patterns among these sequences. It can identify the motifs among a group of DNA sequence controlled by the same Transcription Factor.

- **Running the Software:**

Input: upstream sequences of the target genes of a transcription factor

Number of DNA Sequence: 10

Length of each sequence: 800

Parameters:

1) Background: A 0-order background model generated from the supplied sequences. (Default)
2) Discovery Mode: Classic: optimizes the E-value of the motif information content (Default)
3) Site Distribution: Zero or one occurrence (of a contributing motif site) per sequence. (Default)
4) Motif Width: Between 6 wide and 50 wide (inclusive) (Default)

5) Motif Count: Searching for 3 motifs (top 3).
6) Type of sequence: DNA
7) Command Line: *-dna -oc . -nostatus -time 18000 -mod zoops -nmotifs 3 -minw 6 -maxw 50 -objfun classic -revcomp -markov_order 0*

Result:

Letter frequencies in dataset: **A** 0.323 **C** 0.177 **G** 0.177 **T** 0.323

- **Motif 1:**

TCGYGGCGAGACCMVCSBSTG

E-value= 4.1e-025; Relative Entropy: 30.5 ; for this one E-value and Relative Entropy is lowest, so this is the first preference

Multilevel Consensus Sequence:

It is calculated from the motif letter-probability matrix. For example, in 1st column the probability of T is higher than the probability of G. For the 2nd column, the probability of G is highest and probability is 0.0 for three other letters.

TCGCGGCGAGACCCACCCGTG
G TT AGAGGCCC
CTAT G

- **Motif 2:**

YTTSYCDDTTTTYKSMVAWCTTTTBGKTTTTTTYTTCGAWTTTTHVA

E-value = 6.5e-004; Relative Entropy: 35

Multilevel Consensus Sequence:

TTTCCCAGTTTTTTGAGATCTTTTCGGTTTTTTTTTCGATTTTTCAA
CC GTGGTC C CGCCAGAAC GTTAAC CCCA GACAG CGAC
TA C CT T T C T TG

- **Motif 3:**

GBGACASAYKYGAAACGAGTKTCMCCST

E-value = 1.5e-001; Relative Entropy: 39.1

Multilevel Consensus Sequence:

GCGACACACTTGAAACGAGTGTCACCGT
AGACGCGGTGCA CC A TCT TCG C
TC A