Algorithm Description

Gibbs sampling is a randomized algorithm based on the statistical method of iterative sampling. It is the basis of general local search algorithms. Although Gibbs sampling could not guarantee optimal performance, it works well in practice, which is also the main reason that we choose it for our project.

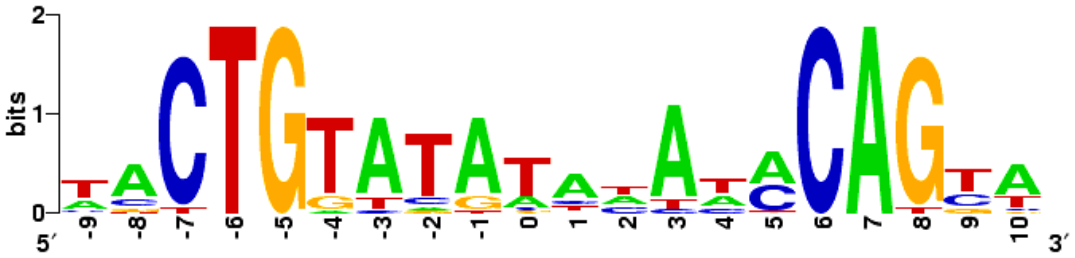

Figure 1. Catabolite Activator Protein binding sites (From weblogo.berkeley.edu/examples.html)

The goal of "Motif finding" is to identify a set of common substrings in different DNA sequence since these substrings are more likely to be binding sites for transcription factors, just like Figure1. To start this process, sequences and length of motif should be inputted first as material and parameter. And then, set random number (s1,...st) as the initial motif starting positions in each input sequence. In this mini project, we choose the first sequence for sampling and pick a random motif position for each unchosen position.

- TGGATGACAG
- CAGTATACCT
- CAGCTTACTA
- CGGAATGCAT

Next, we need to build up a matrix to show the frequency count of unchosen sequences (Table 1). In this way, the weight for each position can be calculated and then normalized based on the data provided in the matrix.

Nucleotide	1	2	3	4	5	6	7
A	0	0.5	0.5	0.75	0.5	0.25	0
C	0	0.25	0	0	0.25	0.5	0.25
G	0.75	0.25	0	0	0	0.25	0
T	0.25	0	0.5	0.25	0.25	0	0.75

Table 1. Frequency of unchosen sequences

After that, we should randomly choose one position using the weight above as the motif and set it as the motif we found. It is necessary to clarify that we should not choose the most possible position. For example, we can have the 26% possibility to choose the position 2 for the motif position (Table 2). Then repeat this process until all sequences have ever been chosen and repeat all these processes until motif positions do not change anymore.

	1	2	3	4
Ai	0.14	0.26	0.3	0.3

Table 2. The weight for position be chosen

Finally, the program should be stopped when the positions do not change anymore. In this algorithm, we grade the results according to ICPC, if the score of the results does not satisfy the requirement, the program will return to the last step and continue running until the grade is good enough. However, sometimes the algorithm will "get locked" into a "local optimum" which means it is running all the time and could not stop. Thus, the program was set up by us to return automatically when it ran 20000 iterations. When it "get locked" into a "local optimum" which means it will continue and do not stop. So we will return automatically return the result after 20000 iterations. We found that it will be meaningless to iterate too many times.

To sum up, Gibbs sampling repeatedly leaves one sequence out and optimizes the motif location in the left-out sequence. One thing deserves to be noticed is that the algorithm has no "stop criterion", so we should find an appropriate way to stop. It is significant to find a balance between accuracy and runtime in terms of the design demand. Besides, it has O(*N*) linear run-time which means you can get a good result if you do not have the high requirement for accuracy. This algorithm is not able to ensure true maximum score, but there is a space to make some "updates" to decrease the score, for instance, making "s^t" contains more similar substrings seems to discover the matched substring easier. Another limitation is that the algorithm also needs to estimate the length of motif so that you can use this algorithm to do the prediction.