Greedy Motif Search

Madiraju Bharadwaj, Anugrah Nambiar, Akash Ranjan, Nandakishor P

Abstract

DNA motif discovery is an important problem in bioinformatics and it is essential for identifying transcription factor binding sites that play a key role in the gene expression process. Through this project report , we intend to demonstrate programming implementation of greedy motif search which is basically to find a set of motifs across a number of DNA sequences that match each other most closely. DNA motif identification is a critical subject in bioinformatics since it is required for identifying transcription factor binding sites that are significant in the gene expression process. Motifs are small patterns that repeat within a group of DNA sequences. However, finding them with an exhaustive search is computationally expensive and unfeasible. As a result, probabilistic and heuristic approaches can be employed to solve this problem. The focus of this research is on greedy building methods for locating DNA motifs. The goal is to find an array of t starting positions that maximizes the similarity value for a given t x n DNA sequence matrix.

Introduction

DNA carries the whole genetic information in it to develop and maintain a living organism. It exists in most cells and serves as a template for synthesizing proteins, which are essential for performing vital functions such as structuring cells, catalyzing metabolic reactions, providing a communication between cells, and replicating DNA. Although each cell that has DNA carries the whole genetic information in it, only a small part is active at the same time in order to be used in protein synthesis. This phenomenon is called gene expression and it allows cells to show unique characteristics by performing specific tasks according to the organ to which they belong.

Regulation of gene expression consists of two steps. i) transcription and ii) translation. In the transcription step, part of gene is copied and a molecule of ribonucleic acid, or RNA, is produced. Then in the translation step, RNA molecules take the role in producing a protein.A critical decision in gene expression is which specific part of the genes should be activated before the transcription process begins. This activation is carried out by special proteins called transcription factors, which bind a specific regulatory region of a gene. Hence it is important to identify transcription factor binding sites in genes to understand gene expression mechanisms better. These transcription factor binding sites are called motifs.

Although there is not much information about most of the transcription factor binding sites, it is known that they are generally short motifs which repeat and are over-represented among a set of regulatory regions of DNA sequences. Therefore, computational approaches can be employed to discover these motifs under the lack of prior assumption. A direct approach to find motifs within DNA sequences is brute force search in which it scans all possible motif candidates and guarantees to find the most desirable one. However, because of the exponential nature of the problem, this method is mostly impractical and a solution can not be obtained in a reasonable time.

Greedy algorithms identify the "most appealing" solution at each algorithm iteration. Most times Greedy algorithms fail to discover a precise solution to a problem. Instead, they frequently arrive at an approximation.

Working of the algorithm is represented below:

From a DNA set

🔽

Run through each possible k-mer in first dna string

🔽

Identify the most appropriate match for the first k-mer within the other DNA strings thereby forming a matrix.

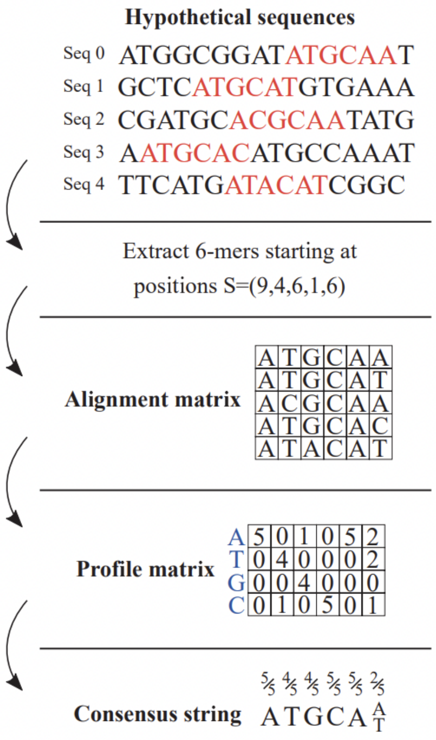
🔽

Score each motif set and select the best scoring set.

The Motif discovery problem

Here we have made use of python language for Greedy motif Algorithm. To evaluate motifs or common sequence parts in DNA, profile matrices can be constructed. A consensus string that represents the motif found is formed by taking the nucleotide that has the highest count for each column in the profile matrix. A greedy approach could be selecting the best possible solution component at a time. Then we find the best probability k-mer from the list of all k-mers and profile matrix. The greedy motif search algorithm's core premise is to locate the collection of motifs that most closely match each other across a number of DNA sequences. We have also made use of Laplace's rules in order to counter the zero issue of

In order to evaluate motifs or common sequence parts in DNA, profile matrices can be constructed. As a first step, we need to determine starting positions for each motif candidate and so on. Starting indices may take value between [0,n-l], where n is the sequence length and l is the motif length. After subsequences are extracted from each sequence, they are aligned and combined in an alignment matrix of size t x l, where l indicates motif length and t indicates the first motif candidate starting is extracted from sequence index 0.



A	T	G	C	A	A
A	T	G	C	A	T
A	C	G	C	A	A
A	T	G	C	A	C
A	T	A	C	A	T

A	5	0	1	0	5	2
T	0	4	0	0	0	2
G	0	0	4	0	0	0
C	0	1	0	5	0	1

Then,the number of each nucleotide(A:Adenine, T:Thymine, G:Guanine and C: Cytosine) per column is counted and written in the profile matrix. If there are equally frequent nucleotides that have the highest count are represented together in the consensus string. The quality of the consensus string can be measured by the score. Given a set of t sequences of nucleotide length n, a set of l nucleotide long parts(l-mers) from each of the sequences that results in maximum consensus score. The goal is to find an array of t starting positions that maximizes the similarity value for a given t x n DNA sequence matrix.

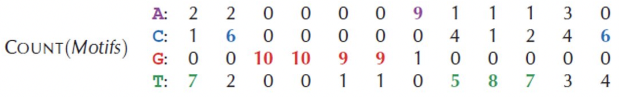
Motif Discovery using Greedy construction:

A greedy construction algorithm starts from an empty solution and adds a single solution component at each iteration according to a greedy condition provided until a valid solution is generated. In the motif discovery case, a solution component corresponds to one entry in a starting position array S and the greedy condition is the amount of similarity score increase when a component is added.

Methodology

Greedy algorithm which is used for selecting the most attractive move at each iteration of the algorithm ,can find an approximation solution but not exact solution to the motif problem.

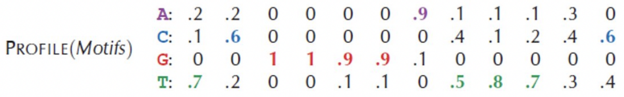
Here we are given a dna sequence and we slice it into multiple windows and create a dna motif matrix ,then using the dna motif matrix we count the no of A,C,G,T and making it into a count motif matrix


COUNT(*Motifs*)

A:	2	2	0	0	0	0	9	1	1	1	3	0
C:	1	6	0	0	0	0	0	4	1	2	4	6
G:	0	0	10	10	9	9	1	0	0	0	0	0
T:	7	2	0	0	1	1	0	5	8	7	3	4

The count matrix contains no of A, C,G,T bases sequence in each column of the dna motif matrix and produces a count matrix .

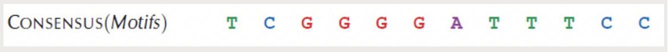
Here after using the count matrix ,we are trying to find the sum of all bases in one column and then divide each element by their sum to get a profile matrix .


PROFILE(*Motifs*)

A:	.2	.2	0	0	0	0	.9	.1	.1	.1	.3	0
C:	.1	.6	0	0	0	0	0	.4	.1	.2	.4	.6
G:	0	0	1	1	.9	.9	.1	0	0	0	0	0
T:	.7	.2	0	0	.1	.1	0	.5	.8	.7	.3	.4

By using the profile matrix ,we are making a consensus(motifs) string which is made by taking the maximum probable element of each column and finding which base it points to and take it as a part of the consensus string .For example here we can see

For the first column T has the maximum value(0.7) ,for 2nd column base C has the maximum value(0.6),for 3rd column base G has the maximum value (1).



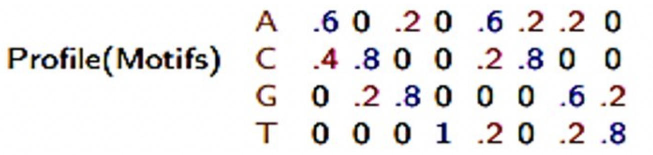
CONSENSUS(*Motifs*)	T	C	G	G	G	G	A	T	T	T	C	C

The probability that a profile matrix will produce a given string is given by the product of individual nucleotides the k-mer that tends to have a higher probability ,when it is more similar to the consensus string of the profile. The closer the k-mer is to the consensus string the larger is the p(k-mer profile) is using the profile matrix we are going to find the probability of every k-mer in a string text and find which k-mer in the dna sequence have highest pr(k-mer).

And suppose if there are multiple values in the profile most k-mers in dna sequence,we select the first such k-mer occurring in the dna string so what we are doing is to run through each possible k-mer and Identify the best matches for this initial k-mer within each out of the following dna strings, thus creating a set of motifs at each step.
Score each set of motifs to find and return the best scoring set.

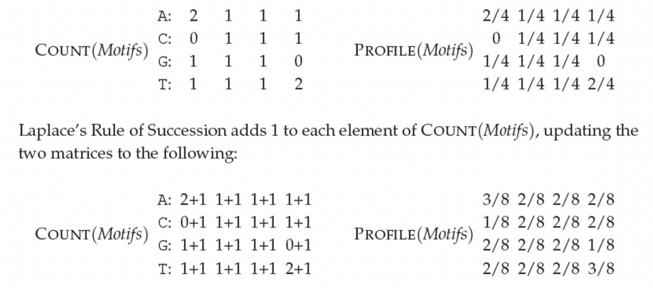
```
 If Score(Motifs) <Score(BestMotifs)
          bestMotifs  Motifs
```

there are some cases where if we get zero probability even though there is only one mismatch.



$$\text{Profile(Motifs)}\quad \begin{array}{cccccccccc} \text{A} & .6 & 0 & .2 & 0 & .6 & .2 & .2 & 0 \\ \text{C} & .4 & .8 & 0 & 0 & .2 & .8 & 0 & 0 \\ \text{G} & 0 & .2 & .8 & 0 & 0 & 0 & .6 & .2 \\ \text{T} & 0 & 0 & 0 & 1 & .2 & 0 & .2 & .8 \end{array}$$

According to Cromwell's rule, statistical maximstates that we should not use probabilities of 0 or 1 unless we are talking about logical statements that can only be true or false. We are going with laplace's rule of succession which states that we can use pseudocounts instead of counts to compute the probabilities.



$$\text{COUNT}(\textit{Motifs})\quad \begin{array}{lcccc} \text{A:} & 2 & 1 & 1 & 1 \\ \text{C:} & 0 & 1 & 1 & 1 \\ \text{G:} & 1 & 1 & 1 & 0 \\ \text{T:} & 1 & 1 & 1 & 2 \end{array} \qquad \text{PROFILE}(\textit{Motifs})\quad \begin{array}{cccc} 2/4 & 1/4 & 1/4 & 1/4 \\ 0 & 1/4 & 1/4 & 1/4 \\ 1/4 & 1/4 & 1/4 & 0 \\ 1/4 & 1/4 & 1/4 & 2/4 \end{array}$$

Laplace's Rule of Succession adds 1 to each element of COUNT(*Motifs*), updating the two matrices to the following:

$$\text{COUNT}(\textit{Motifs})\quad \begin{array}{lcccc} \text{A:} & 2{+}1 & 1{+}1 & 1{+}1 & 1{+}1 \\ \text{C:} & 0{+}1 & 1{+}1 & 1{+}1 & 1{+}1 \\ \text{G:} & 1{+}1 & 1{+}1 & 1{+}1 & 0{+}1 \\ \text{T:} & 1{+}1 & 1{+}1 & 1{+}1 & 2{+}1 \end{array} \qquad \text{PROFILE}(\textit{Motifs})\quad \begin{array}{cccc} 3/8 & 2/8 & 2/8 & 2/8 \\ 1/8 & 2/8 & 2/8 & 2/8 \\ 2/8 & 2/8 & 2/8 & 1/8 \\ 2/8 & 2/8 & 2/8 & 3/8 \end{array}$$

We use the same rules mentioned above and create another profile matrix and another probable k-mer for the next row of the dna matrix ,like this we are going to do the same steps and append the best motifs of each row.

RESULTS

Our Greedy motif algorithm, when implemented, ran error-free. It is understood that Greedy search is effective in reducing the time taken to compute the algorithm when compared to Brute-Force Algorithm. It will find a similar set of patterns which can be eligible candidates of Motifs in DNA. .

Conclusions

Even though Programmable implementation of the Greedy Motif Algorithm was successful, it is seen that they have a polynomial time complexity of O (ln^2 + nlt) where l is the length of the motif, n is the length of the DNA samples, and t is the number of DNA samples. Many times results are dependent on the initial random motif and it is unlikely to get a good result in a single run. Greedy algorithm is a good method to find motifs approximately but since it is looking one step at a time,it is not a good method compared to available options.

References

https://bioinformaticsalgorithms.com/data/debugdatasets/motifs/GreedyMotifSearch.pdf

https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9219366&tag=1

https://www.mrgraeme.com/greedy-motif-search/