An Efficient Motif Finding Algorithm

Chen-Tsun Chuang[a], Cheng-Chia Yang [b]

[a]Department of Computer Science & Information Engineering, National Formosa University

[b] Kuang Tien General Hospital

cct@nfu.edu.tw

論文摘要

在生物資訊的研究領域中如何尋找 Motif，一直是一個相當具有挑戰性的問題，尋找 Motif 也是多重序列排比的一項重要主題，因此一直吸引許多學者投入這方面的研究，並有一些不錯的成效，對生物學家搜尋 motif 具相當大的幫助。但由於此問題具相當的複雜性，因此仍存在很多問題極待克服，隨著定序速度的成長，此問題更加重要。詳盡搜尋的方式雖然可以準確的找到 motif，但是當基因序列和搜尋的 motif 長度過長的時候，在運算時間容易呈指數的倍數成長，因此，在本研究中，我們構思了一種快速而有效的多重序列排比方法，稱為『樣式區塊配對演算法(pattern block matching algorithm)』，此方法改善已往之字元循序比對技巧，因此，能快速集合候選之 Motif，進而，迅速搜尋 Motif，並且可以避免因序列長度(N)增加時所造成搜尋 motif 執行效率和準確度下降等問題，又能夠兼顧效率及精準的方式來找到最好的結果。

關鍵詞：生物資訊，共同短序列，樣式區塊配對演算法，多重序列排比。

Abstract

Recently, motif finding became a very popular area in bioinformatics, thus more and more researches are interested in discover motif. However, many algorithms can not solve the Pevnzer and Szes' challenge problem. It motivates my algorithm to purpose construct an exhaustive method to improve the motif finding performance in discover signals such as: (9,2),(11,3),(13,4),(15,5), and (17,6) and to solve the problem that accuracy will be descending while sequence length is increasing.

Although exhaustive search method could find motifs accuracy, it still needs to face the problem that computing time will be grown exponentially by length increasing of genomic sequence and motif. The research through provide assist skills not only to avoid the length of sequence increasing effect computing time, but also efficiency and accuracy to discover the optimal result. We could expect the research will bring well performance and apply in other bioinformatics domain related motif finding is expandable.

Keywords：Bioinformatics, Motif, Pattern Block Matching Algorithm, Multiple Sequence Alignment**.**

Introduction

The sequencing data of genomic sequences is growing rapidly day-by-day; however gene annotation data are not growing with the genomic sequencing data. The massive amount of raw sequences generated, the more laboratory cost and time consuming for biologists, so that lead bioinformatics to play an important role in genomic sequence analysis.

Organism's evolution and mutate continuously, but some important genomic fragment will be conserved through the evolution from the ancestor, this conserved fragment is called motif. Motif is a consensus short pattern, it can be detected in protein, DNA, and RNA sequences that could be predicted some molecular function, structure property and related protein families. Many problems lead to identify motif becoming more complex, so that a lot of motif finding strategies were developed. Motif can be at different positions randomly in each sequence. However the signals may weak, not significant enough in each gene upstream region sequence, such as promoters and splicing sites.

In 2000, Pevzner PA indicate a challenge problem to implant (*l, d*)-Motif. Let motif be a fixed but unknown short consensus sequence M of length l. Suppose that M occurs once in each of T genomic sequences of common length n, but that each occurrence of M is corrupted by exactly d nucleotide substitutions in positions chosen independently at random. T sequences are given which each of them contain short consensus sequence M.

Each fragment of length L which is similar to motif M and d base pairs (bp) mutate, denoted *(L,d)*-motif. The challenge problem is to identify *(15, 4)*-motif which means 4 bp mutate in motif of length 15 bp ,the motif in 20 sequences and each sequence of length is 600 bp. In figure 1, described how to identify *(8, 2)*-motif in 4 sequences, each sequence have a consensus motif M=ACAGGATC, all homology fragments have 2 bp mutation with M.

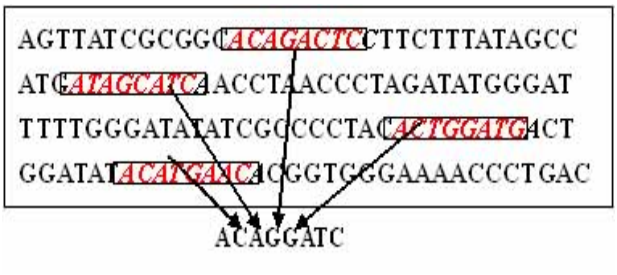

Figure 1 the Challenge Problem of example

The motif finding algorithms such as Gibbs[7],CONSENSUS[6],MEME[2],TEIRESIAS[11], WINNOWER[10],SP-STAR[10],PROJECTION[5]etc.
In the past, some algorithms were evaluated by performance coefficient, to find *(15, 4)*-motif in 20 sequences and each length is 100-1000bp[10] CONSENSUS, MEME to find *(15, 4)*-motif in 20 sequences, the performance coefficients is weak by length increasing [12]; moreover SP-STAR and WINNOWER are helpful to find *(15, 4)*-motif, but SP-STAR not performed very well in (15, 4)-motif finding in sequences of each length 1000 bp and WINNOWER also not performed very well in (15, 4)-motif finding when discover motif in sequences of each length over 1300 bp [6].
In 2002, Buhler discuss different kinds of *(L,d)*-motif in 20 sequences and each sequence length is 600 bp [5], compared the algorithms such as Gibbs, WINNOWER, SP-STAR, PROJECTION. The research resulted algorithms such as Gibbs, WINNOWER, SP-STAR to search *(14, 4)-*, *(16, 5)-*, and *(18, 6)*-motif weakly. PROJECTION could successfully to find *(15, 4)*-motif in 20 sequences and each sequence length is 2000 bp, but to find *(9,2)-* , *(11,3)-* , *(13,4)-*, *(15,5)-*, and *(17,6)*-motif were still needed to improve the performance.
The purpose of our algorithm is to construct a new exhaustive algorithm to solve the challenge problem. The exhaustive algorithm always finds motif accuracy, but the computing time usually grows as exponential by genomic sequence or the length of motif increasing. Our exhaustive algorithm could provide efficiency and accuracy optimal result by our assist skills and avoid the problem of computing time grown as exponential by sequence length increasing. We tried to solve the PROJECTION algorithm not performed very well in *(9,2)-* , *(11,3)-* , *(13,4)-*, *(15,5)-*, and *(17,6)*-motif finding, and to solve the problem of sequence length (N) increasing.

Sample and Tool

Our approach is to create a new algorithm to find motif and write programs by Visual Basic.NT.
We choose 20 sequences randomly and each length is 1000bp, the artificial motif of length 15bp implanted in each sequence randomly and at random position of sequence, as the motif implanted, 4 positions of motif will switch randomly. The algorithm is developed for solving the Challenge Problem.
Another sample is to discover transcription factor regulation element in eukaryotic gene upstream region which is evidenced contain transcription factor regulation element sequences in orthologous sequences, these gene are from as follows:

1. Preproinsulin: there are two kinds of preproinsulin signals, those from TRANSFAC database [13] and CT II element [5].
2. Dihydrofolate reductase(DHFR): the known motif of DHFR is non-TATA transcription start signal[8].
3. Metallothioneins: there are three kinds of known motifs of metallothionein those are MREa promoter, MREd promoter, and MREf promoter [3].
4. c-fos gene upstream region: 3' end of c-fos serum response element [9].
5. S. cerevisiae genes are SWI4, CLN3, CDC6, CDC46, and CDC47. The promoter of these gene is called ECB element, that contain cell-cycle-dependent promoter[3]

Method

Many scientists are interested in motif finding, thus many algorithms are developed, and those algorithms are expected to find motif rapidly, accuracy and not lose their performance by length increasing. Motif length and switched base pair in signals made motif finding to become more difficult, we tried to increasing computing efficiency and didn't lose our accuracy to find motif by our approach.
The data structure of long DNA sequences is numeral through our transformation initially [5]. Each DNA sequence could be divided into several patterns; each pattern width of length *k* by exhaustive algorithm and each pattern shift S bp by user.All pattern of length *k* are sets of each DNA sequence. We construct a table to presentation the *K-mer* frequency and positions and design assist skills to increase performance efficiency and to reduce computer memory by our new approach, these reasons are as follow:

1. We transformed DNA data structure to be quantity. It could reduce computer memory; though this data structure aligns patterns could also be more rapidly.
2. When pattern length is 6bp, there will generate 4^6 kinds of patterns, if one base pair switches, the *(6,1)-pattern* will generate more kinds of patterns. The research only analyzes pattern occurring in sequence, it doesn't evaluate entire possible patterns.
3. Two assist skills are designed to discard patterns not correspond with this rule; these are total number of patterns and G+C% of patterns.
4. We admit 2d to switch in motif to avoid the unspecified motif missing easily. For next filtering out patterns not correspond with this rule, we design the number of match between patterns must N-1, it is very accuracy to find motif.

Each DNA sequence could be divided into several patterns of length *k* and each pattern shifts 1 bp and forms overlapping words. The number of occurrence of patterns and position of patterns in sequences are detected and this information is collected to find motif, the steps are described as follows:

Step1: The DNA sequences $S_{i,\ i=1,2...,N,}$ contain motif of length L, P_k defined pattern of length K, $P_k=\{w_1w_{2...}w_K\}$, $w_K \ni \{A, T, G, C\}$, the pattern of length K could be

defined by user, such as *(15,4)-motif*, *K*=15, each S_i contains (L-K+1) kinds of patterns.

Step2: The sequence S_i is divided into several patterns and each pattern shifts 1 bp and forms overlapping words in sequences by hash method to record and to find occurrence and position of pattern in sequences, the observed number of occurrence of each pattern denotes count, the position of pattern is designed to look for the distribution of patterns. Each sequence performs our program only once; it can get the observed occurrence and position of patterns of length k in S_i. If pattern occurs twice above, the next same pattern will cumulate the number of occurrence to the first same record of pattern.

Step3: Calculate the total number of patterns in sequence S_i, and count for G+C% in each sequence, G+C%= occurrence of base G+ base C of total number of collected patterns/ total number of collected pattern*K. Such as there is a sequence AATCG, if K=3, the patterns are AAT, ATC, TCG, and G+C% of pattern is 3/(3*3)=3/9.

Step4: Aligned the least total number of patterns in a sequence with the less total number of patterns in a sequence, if both of total number of patterns are the same, G+C% can help to identify the differences of patterns and choose the most different one to aligned with. If the G+C% couldn't identify the similarity, such as two sequences have the same G+C%, we will choose sequences randomly and find the qualify patterns to align with. We construct a Pattern Match Table to record pattern occurring in sequence by each pattern alignment and calculate the cumulate number of matches of these candidate patterns.

Xor is the method for discovering *(l,d)-motif* to match the same bit of each two pattern alignment ,admit 2d to switch in pattern and collect all the pattern meet the "AND-OR"

$$[result=] \text{pattern in Sequence } S_{i\,i=1..n} \text{ Xor pattern in Sequence } S_{ii=1..n}$$

Each two patterns are aligned continuously in the other sequences, when candidate pattern doesn't meet *(l,d)*-rule in next sequence, then discard it from Pattern Match Table and iterative the step until all candidate pattern alignment is finished.

If the same pattern occurs twice, the second record will be merged to the first one and cumulate the number of matches of the second one to the first one.

After the alignment finished, the cumulate number of matches must over N-1, N defined the number of sequence S_i, and this threshold assumed homology pattern in each sequence certainly. If candidate pattern doesn't qualify this threshold then discard it; if threshold is satisfied, that will be the predicted motif.

Research method of example:

To assume there are six sequences S_1=AGTTGTATATCGTG, S_2=TAATATATAATATA, S_3=TATATCCCCAGCTG, S_4=GTGTGTGTAGATAG, S_5=TATCTATATCTATA, S_6=CCCTATACAGGCCG, and each length is 14bp, these sequences are implanted a motif TATATA, each implant switch one base pair at random position, denoted *(6,1)-motif*, our purpose is to discover *(6,1)-motif*.

Step1 and Step2: We transform data structure of sequences to be numeral, each pattern unit is k=6 for discovering *(6,1)-motif* and record position and the number of occurrence of pattern in six sequences(table 1).

Table 1 pattern record table

S1			S2			S3		
Pattern	Count	Position	Pattern	Count	Position	Pattern	Count	Position
AGTTGT	1	1	TAATAT	2	1,8	TATATC	1	1
GTTGTA	1	2	AATATA	2	2,9	ATATCC	1	2
TTGTAT	1	3	ATATAT	1	3	TATCCC	1	3
TGTATA	1	4	TATATA	1	4	ATCCCC	1	4
GTATAT	1	5	ATATAA	1	5	TCCCCC	1	5
TATATC	1	6	TATAAT	1	6	CCCCAG	1	6
ATATCG	1	7	ATAATA	1	7	CCCAGC	1	7
TATCGT	1	8				CCAGCT	1	8
ATCGTG	1	9				CAGCTG	1	9
S4			S5			S6		
Pattern	Count	Position	Pattern	Count	Position	Pattern	Count	Position
GTGTGT	2	1,3	TATCTA	2	1,7	CCCTAT	1	1
TGTGTG	1	2	ATCTAT	2	2,8	CCTATA	1	2
TGTGTA	1	4	TCTATA	2	3,9	CTATAC	1	3
GTGTAG	1	5	CTATAT	1	4	TATACA	1	4
TGTAGA	1	6	TATATC	1	5	ATACAG	1	5
GTAGAT	1	7	ATATCT	1	6	TACAGG	1	6
TAGATA	1	8				ACAGGC	1	7
AGATAG	1	9				CAGGCC	1	8
						AGGCCG	1	9

Step3: S_5 is the least number of patterns in these six sequences, the number of patterns in S_5 is 6, and the others are S_2 and S_4, the number of patterns in S_2 is 7 and the number of patterns in S_4 is 8, and the number of patterns in rest sequences, each is 8. The G+C% are: S_1=15/54, S_2=0, S_3=33/54, S_4=18/48, S_5=6/36, S_6=27/54.

Step 4: According the thresholds of our approach, we pick the sequence have the least number of patterns initially, and finding *(6,1)-motif* through align S_5 and S_2(figure 2). We admit 2d positions to mutate at random position, so that there are 4 positions to mutate at 6 positions randomly, collect each pattern number of matches meet *(l, d)-rule*, if pattern miss this rule, then discard it.

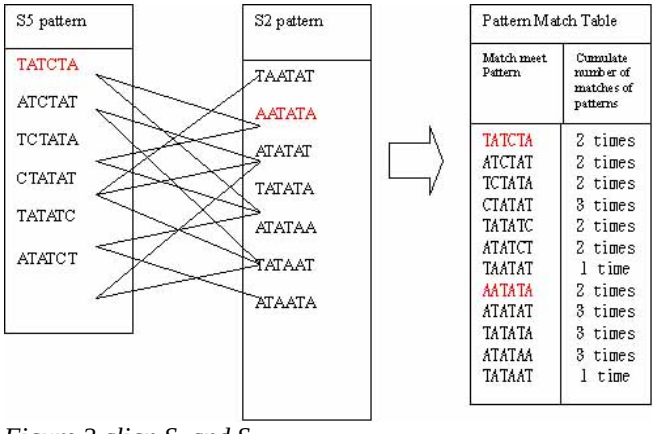

Figure 2 align S_5 and S_2.

Collect patterns after S_5 and S_2 alignment, then align with S_4(figure 3), previous collected pattern discarded through align with S_4, and the record of the discarded pattern in Pattern Match Table will delete, too. The new finding pattern will add to Pattern Match Table.

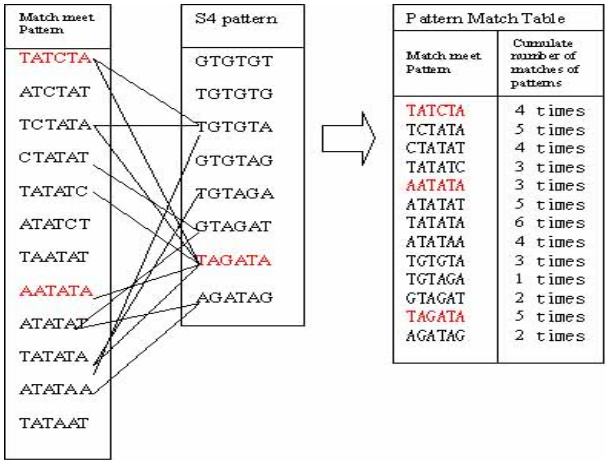

Figure 3 Collect patterns align with S_4

The collected patterns through S_5, S_2 and S_4 alignment will align with the next sequence which has the less number of patterns. S_1, S_3 and S_6 have the same number of patterns, so that G+C% helped to discover the differences of these sequences. The G+C% of patterns meet *(6, 1)-rule* in S_5, S_2 and S_4 is 13/78, this value compared with G+C% of patterns meet *(6, 1)-rule* in S_1 is 15/54, G+C% of patterns meet *(6, 1)-rule* in S_3 is 33/54 and S_6 is 27/54. The result of the most difference sequence is S_3, the alignment order is S_3, S_6 and S_1.

The previous resulted pattern align with S_3(figure 4),we found a pattern TATATC which repeated, we cumulated each the number of matches of pattern TATATC to the same record, and then align with S_6(figure 5).

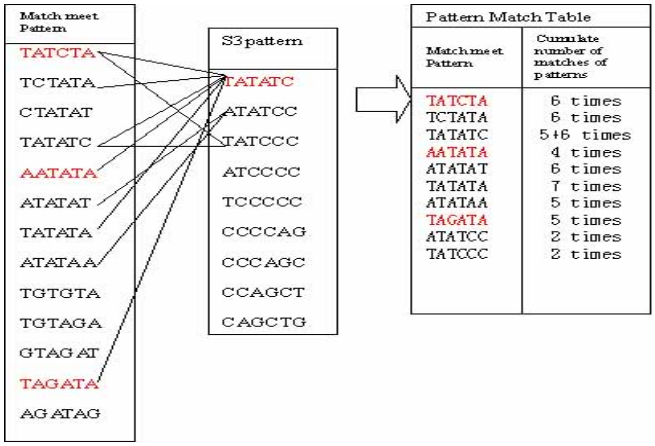

Figure 4 Collect patterns align with S_3

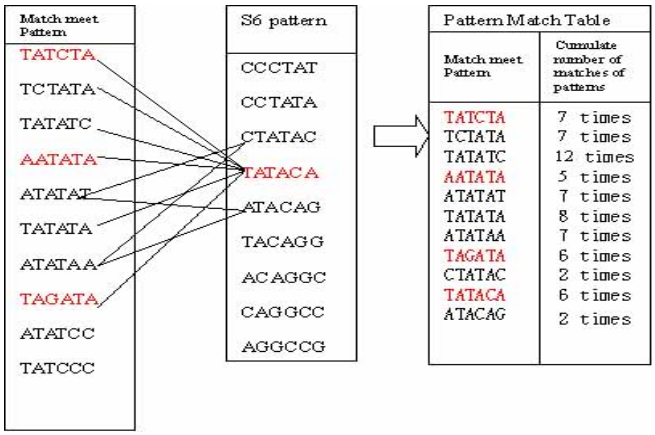

Figure 5 Collect patterns align with S_6

Align with S_1 (figure 6) also have a pattern TATATC repetitive and the number of matches also cumulated.

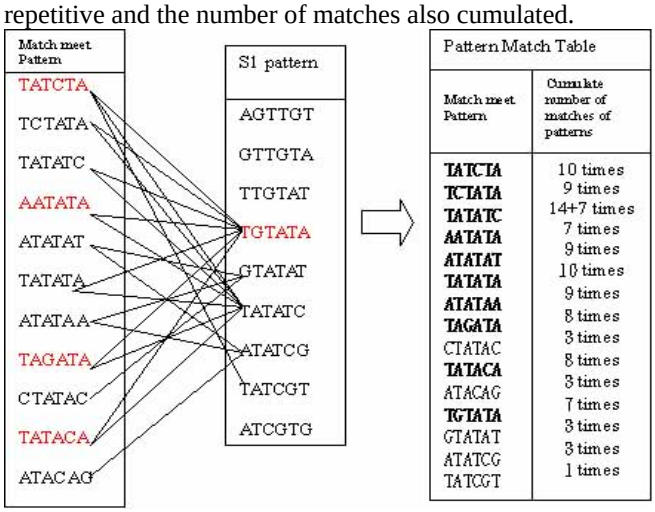

Figure 6 Collect patterns align with S_1

The final predicted motifs will result in our initial implanted motif are TGTATA, AATATA, TATATC, TAGATA, TATCTA, and TATACA. They are also the implanted motif. So the approach can find motif successfully (Table 2).

Table 2 the final result

Pattern Match Table	
Match meet Pattern	Cumulate number of matches of patterns
TATCTA	10 times
TCTATA	9 times
TATATC	21 times
AATATA	7 times
ATATAT	9 times
TATATA	10 times
ATATAA	9 times
TAGATA	8 times
TATACA	8 times
TGTATA	7 times

Conclusion and remarks:

The algorithm is to analyze the global properties of DNA sequences, because we align the entire candidate pattern to discover the most interesting local information, it exhaustive and precise. Computer user doesn't need to make many conditions previously and only need to define pattern length k, it can more simplify to user. Our approach can focus different kind of motifs, such as *(9,2)-motif, (11,3) -motif, (13,4) -motif, (15,5) –motif* and provide accuracy method to search motif, the algorithm only perform once can acquire the accuracy result and doesn't need to refine the result iteratively, the accuracy won't decreasing by length increasing. Our method successfully to improve the weakness of exhaustive algorithm, not only the method is very accuracy but also save cost and time. The transformed data structure of sequences is also efficiency to store in system and save our system resources.

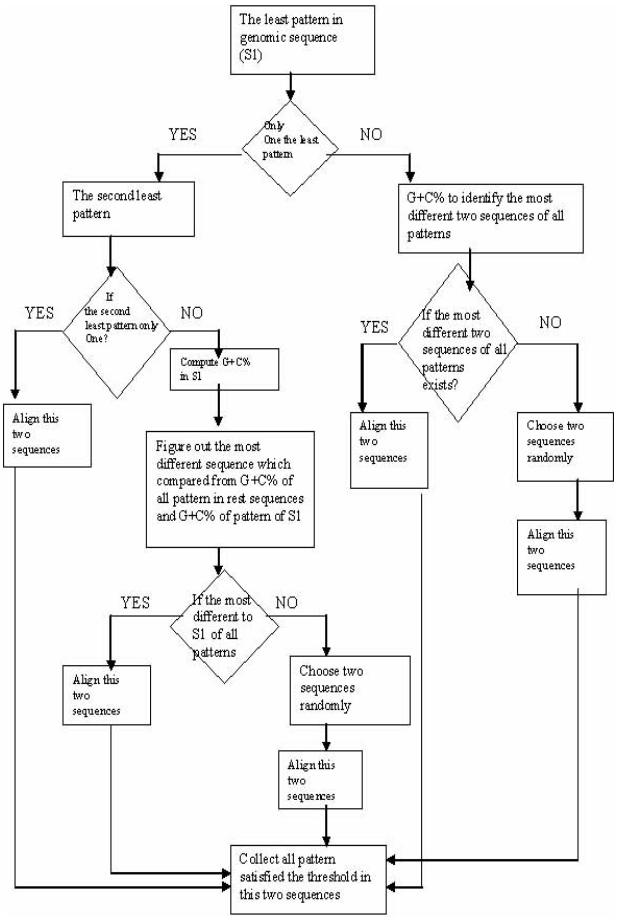

Figure 8 step process 1

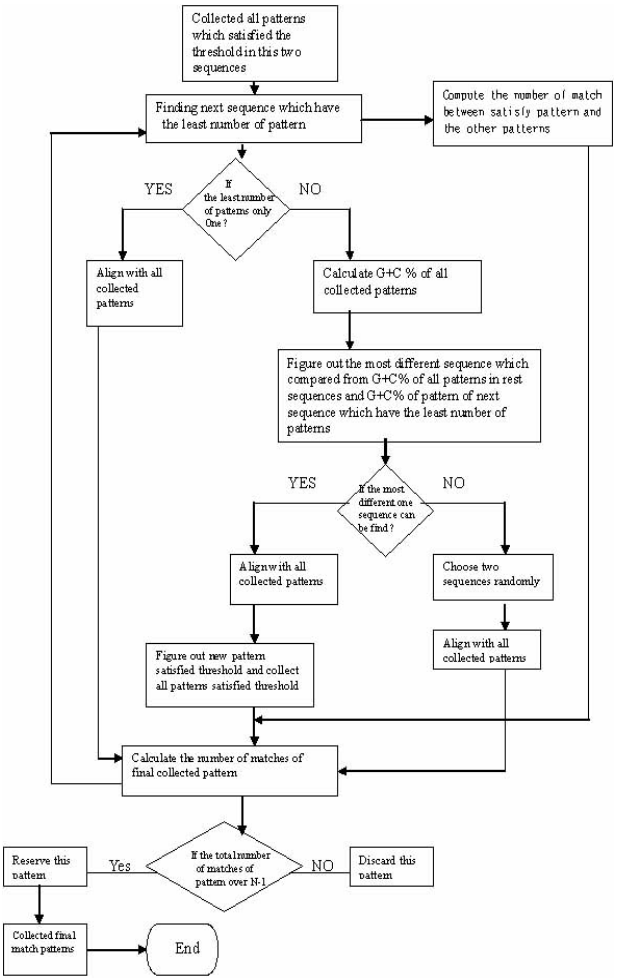

Figure 9 step process 2

Reference

1. Andersen, R.D., Taplitz, S.J., Wong, S., Bristol, G., Larkin, B., and Herschman, H.R. Metal-dependent binding of a factor in vivo to the metal-responsive elements of the metallothionein 1 gene promoter. *Molecular and Cellular Biology* 7, 3574–81. 1987.

2. Bailey, T., and Elkan, C. Unsupervised learning of multiple motifs in biopolymers using expectation maximization. Machine Learning; 21:51-80. 1995.

3. Boam, D.S.W., Clark, A.R., and Docherty, K. Positive and negative regulation of the human insulin gene by multiple trans-acting factors. *J. Biological Chem.* 265, 8285–96. 1990.

4. Buhler J. and Tompa M.. Finding motifs using random projections. *J Comput Biol*,;9(2):225-42. 2002.

5. Chuang ,C.T.、Yang,C.C.、Huang,M.F. , Statistical Analysis of the Genomic Sequence of Human Chromosome 22, Tzu Chi Medical Journal， 2004.6，16(3)，151 - 158。

6. Hertz, G., and Stormo, G. Identifying DNA and protein patterns with statistically significant alignments of multiple sequences. Bioinformatics;15:563-577, 1999.

7. Lawrence, C.; Altschul, S.; Boguski, M.; Liu, J.; Neuwald, A.; and Wootton, J. Detecting subtle sequence signals: a Gibbs sampling strategy for mul-tiple alignment. Science..262:208-214. 1993.

8. McInerny, C.J., Partridge, J.F., Mikesell, G.E., Creemer, D.P., and Breeden, L.L. A novel Mcm1-dependent element in the SWI4, CLN3, CDC6, and CDC47 promoters activates M/G1-speci. c transcription. *Genes and Development* 11, 1277–88 1997

9. Means, A.L., and Farnham, P.G. Transcription initiation from the dihydrofolate reductase promoter is positioned by *hip1* binding at the initiation site. *Mol. Cell. Biol.* 10, 653–61 1990

10. Pevzner PA, Sze SH. Combinatorial approaches to finding subtle signals in DNA sequences Proc Int Conf Intell Syst Mol Biol.;8:269-78 2000.

11. Rigoutsos, I. and Floratos, A. Combinatorial pattern discovery in biological sequences: The TEIRESIAS algorithm. Bioinformatics, Published erratum appears in Bioinformatics. 14(1):55~67 1998.

12. Sze.S. Gelfand.M. and Pevzner.P. Finding weak motifs in DNA sequences. *In proceedings of Pacitic Symposium on Biocomputing*.;235-246 2002.

13. Wingender, E., Dietze, P., Karas, H., and Knüppel, R. TRANSFAC: A database on transcription factors and their DNA binding sites. *Nucl. Acids Res.* 24, 238–41 1996.