2004－ICS－138　(3)
2004／12／4

Developing Text Mining Based Algorithms for Classifying Biological Sequences

HOANG KIEM† and DO PHUC†

Abstract. The paper focuses on developing the algorithms for discovering the frequent motifs and the ordered co-occurrence set of frequent motifs supporting the classification of the family of biological sequences. AprioriBioSequence is the name of our proposed algorithm, which has been developed from the algorithms of discovering the frequent patterns in document sentences of text mining. AprioriBioSequence can discover the frequent motifs without specifying the length of discovered motifs. Besides, paper also deals with the algorithm for discovering the ordered set of the co-occurrence frequent motifs for classifying the biological sequences. The experimental results of the proposed algorithms with the E-Coli Promoter sequences are presented.

Keywords: biological sequences, co-occurrence, frequent motifs, text mining

1. Introduction

The problem of discovering the frequent motifs in a set of biological sequences is one of important problems of biological sequence analysis [1], [2], [3], [4], [5]. Based on the frequent motifs, we can discover the feature sets of a family of biological sequences [1], the conservation areas thru the evolution generations [5], the association between gene expression and gene functions, the classification rules for gene classification There are many research works related to this problem. In [3], Jason T. L. Wang and Steve Rozen have developed techniques for DNA Sequence Classification based on frequent patterns discovery. In [5] Timothy Lawrence Bailey has employed probabilistic model for discovering motifs in DNA & protein sequence. In [4] R. Durbin, S. Eddy, A. Korgh, G. Mitchison employed probabilistic models of protein and nucleic acids for biological sequence analysis. These methods employed probabilistic models such as EM model, Hidden Markov Model. With a large volume of biological sequences, the demand of developing the algorithms which can work with the large number of biological sequences is a significant problem.

† Center for Information Technology Vietnam National University, HCM city

In this paper, we have developed an algorithm based on text mining which can discover the frequent ordered terms in the sentences of document [3] for discovering the frequent motifs [2] and the classification rules based on the co-occurrence graph of frequent motifs to be discovered in biological sequences.

Our algorithm uses the idea of discovery the frequent sets in data mining. We believe that our approach will satisfy the demand of large DNA sequence volume. Paper is organized as follows 1) Introduction 2) The fundamental concepts 3) Problem statement and time complexity 4) Developing the algorithms for discovering the frequent motifs 5) Developing the algorithm for discovering the classification rules of the family of biological sequences based on the ordered set of co-occurrence frequent motifs 6) Experimental Results 7) Conclusions

2. The basic concepts

Definition 1: The biological alphabet

The biological alphabet A composes of basic biological letters. These biological letters correspond to 4 nucleotides of DNA sequences or 20 amino acids of Protein. For simplicity, in the following sections, the biological sequences is called as sequence.

Definition 2: *The biological sequences*

Biological sequence is the sequence to be created from the basic biological letters of the biological alphabet A. Let s be a biological sequence, then |s| be the length of s or number of letters in ***s***.

Definition 3: *Sub-sequences*
Given two sequences s, t with $s=s_1s_2 \ldots s_n$ and $t=t_1t_2 \ldots t_m$, sequence t is called as the subsequence of s or s contains t if:

(i) $m \le n$

(ii) $\exists k \in [1,\ldots,n-m+1]$, $t_h=s_{h+k-1}$, $\forall\ h \in [1,\ldots,m]$

Denote $t \subseteq_{seq} s$ if t is the sub-sequence of t.

Example 1: Given sequence s =ACACACAC and sequence t = CACA, then $t \subseteq_{seq} s$**.**

Definition 4: *List of sequences*
Let S be a set of sequences, P(S) be the set of all sub-sequences of all sequences of S. Define function ρ_M: $P(S) \rightarrow S$ such that $\forall\ u \in P(S)$, $\rho_M(u)=\{s \in S \mid u \subseteq_{seq} s\}$.
Where $\rho_M(u)$ corresponds to the sequences of S containing u. $\rho_M(u)$ is called as a list of sequences containing u.

Property 1: $s \subseteq_{seq} t \Rightarrow \rho_M(s) \supseteq \rho_M(t)$

Example 2**:** Given 3 sequences as follows:
s_1 = 'A**CGTTT**ATAAAGTCACACGTAGCCCCACGTACAGT'
s_2 = 'CGCGTCGAAGTCGACCGTGGGAAAGTCACACAGT'
s_3 = 'GGTCGATGCA**CGTTT**CTAAATCAGTCGCACACAGT'
If u ='**CGTTT',** then list of sequences containing u is $\rho_M(u) = \{s1, s_3\}$.

Definition 5: *Support of sub-sequences*
Let S be the set of sequences, and u is a sub-sequence, $u \in P(S)$, The support of sub-sequence u in S is defined as the ratio of the number of sequences of S containing u and the number of sequences in S. The support of sub-sequence u is denoted SPM(u) and calculated by $SPM(u)= |\rho_M(u)|/|S|$ where |S| is the number of sequences in S.

Definition 6: *The frequent motifs*
Let S be the set of biological sequences, $minsupp \in (0,1]$ be a given threshold. Given $u \in P(S)$, u is called a frequent motif with threshold minsupp iff $SPM(u) \ge minsupp$.
Denote FM(S, minsupp) be the set of frequent motif of S with threshold minsupp, it means that: $FM(S, minsupp) = \{u \in P(S) \mid SPM(u) \ge minsupp\}$

Example 3**:** Given 5 sequences as follows:
s_1 = 'A**CGT**TAT**AAA**GTCACACGTAGCCCCACGTACAGT'
s_2 = 'CGCGTCGAAGTCGAC**CGT**GGG**AAA**GTCACACAGT'
s_3 = 'GGTCGATGCA**CGT**TCT**AAA**TCAGTCGCACACAGT'
s_4 = 'A**CGT**TAG**AAA**GTAGCTACCCGTACGTCACACAGT'
s_5 = 'A**CGT**GTT**AAA**GTCAACGACGTACGTCGTCACAGT'
Some frequent motifs with threshold minsupp=1.0 are as AAA, CGT . . .

3. Problem Statement and Time Complexity

Problem Statement:
Let S be the set of biological sequences and $minsupp \in (0,1]$ be a given threshold, find FM (S, minsupp).

Time Complexity Analysis:
Let h be the length of discovered frequent motifs and $minsupp \in (0,1]$ be a given threshold, S contains n sequences with length m. The number of candidates of frequent motifs with threshold minsupp and length h is $|\mathcal{A}|^h$. The number of checked positions in n sequences is $(m-h+1)^n$, therefore the time complexity if the problem is $O(m-h+1)^n |\mathcal{A}|^h)$.

4. Developing the Algorithm for Discovering the Frequent motifs

Proposition 1: If $s \in FM(S, minsupp)$ and $t \subseteq_{seq} s$ then $t \in FM(S, minsupp)$.
Proof: According to the property of function ρ_M, we have that if $t \subseteq_{seq} s$ then $\rho_M(t) \supseteq \rho_M(s)$, thus:
$SPM(t) \ge SPM(s) \ge minsupp \Rightarrow t \in FM(S, minsupp)$.

Proposition 2: If $s \notin FM(S, minsupp)$ and $s \subseteq_{seq} t$ then $t \notin FM(S,minsupp)$.
Proof: According to the property of function ρ_M. we have that if $s \subseteq_{seq} t$ then $\rho_M(s) \supseteq \rho_M(t)$, therefore $minsupp \ge SPM(s) \ge SPM(t), \Rightarrow t \notin FM(S, minsupp)$.

Definition 7: *Set of the candidates of frequent motifs*
Let $Fk = \{u \in FM (S, minsupp) \mid |u| = k\}$ be the set of frequent motifs of S with threshold minsupp and length k and C ($\boldsymbol{F_k}$) be the set of the candidates of frequent motifs for Fk in the next step. Based on the result of proposition 1, C ($\boldsymbol{F_k}$) is defined as:
$C(\boldsymbol{F_k}) = \{w \in P(S) \mid |w| = k+1$ and $t \in \boldsymbol{F_k}, c \in \boldsymbol{F_1}$ and $w=t+c\}$
Example 4: Suppose that s=ACCA is in F4 and $F_1=\{A, C\}$, the set of candidates for the next step is $C(\boldsymbol{F_4})= \{ACCAA, ACCAC\}$

Algorithm 1. Find the set of frequent motifs

With the DNA where $|\mathcal{A}| = 4$, the search space is a 4^{th} degree tree (figure 1). The basic idea of algorithm is to sprout (expand the search space) to branches containing the subsequences to be contained in s and t. For example, if s and t contain

only letter A and letter C, tree is sprouted to the branch for letter A and branch for letter C. Then continuing to branches for AA and AC, branches for A and CA, CC for branch C, then AAA, AAC, ACA, ACC, CAA, CAC, CCA, CCC, . . .

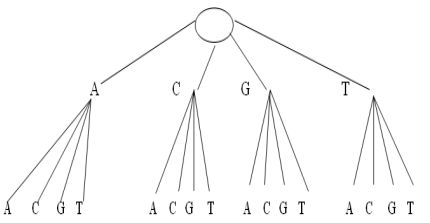


Figure 1: *The search space of problem*

AprioriBioSequence algorithm is designed based on this idea. Given a threshold minsupp, let $\boldsymbol{F}_k$ be the set of frequent motifs with threshold minsupp and length k and Ck be the set of the candidates for step k of the algorithm (step for discovering the frequent motifs with length k). The main steps of the **AprioriBioSequence** are as follows:

Input: Set of biological sequences S and threshold minsupp$\in(0,1]$
Output: FM(S, minsupp)
1) Create $C_1 = \{\{a\} \mid a \in \mathcal{A}\}$
2) k = 1
3) While $(Ck \neq \emptyset)$ do
4) Let Fk = $\{u \in C_k \mid SPM(u) \geq minsupp\}$;
5) k = k + 1;
6) Create Ck = C($\boldsymbol{F}_{k-1}$); // using the definition 7
7) End while
8) FM(S, minsupp) = ∅;
9) For (i=1; i < k; i++)
10) FM(S, minsupp) $\cup$= $\boldsymbol{F}_i$
11) Endfor
12) Return FM(S, minsupp)

Algorithm 2. Create set of the candidates of frequent motifs $C(F_k)$ for the step k.
Input: Set Fk and F1
Output: Set of the candidates of frequent motifs $C(F_k)$ for the step k.
The main steps of algorithm 2 are as follows:
1) $C(F_k) = \emptyset$
2) For each $u_y \in F_k$ do begin
3) For each $u_x \in F_1$ do begin
4) $u_t = u_y + u_x$ // sequence concatenation
5) $C(F_k) = C(F_k) \cup \{u_t\}$
6) Endfor
7) Endfor
8) Return $C(F_k)$

Theorem 1. Algorithm 2 can create a complete set of the candidates of frequent motifs for the step k of the AprioBioSequence Algoprithm

Proof: We have $C(F_k) = \{ w \in P(S) \mid |w| = k+1$ and $t \in F_k$, $c \in F_1$ and $w = t+c \}$. Two loops (line 3, line 4) guaranties to scan all elements of F_k and F_1, therefore algorithm 2 can create properly $C(F_k)$.

Theorem 2. Algorithm 1 (**AprioriBioSequence)** discover all frequent motifs with threshold minsupp
Proof: We prove this theorem by induction. With k = 1 then F_1 is correct because F_1 is created by checking each letter of $\mathcal{A}$ (line 1). Suppose that $\boldsymbol{F}_k$ is correct, we will prove the algorithm 1 can create $\boldsymbol{F}_{k+1}$ from $\boldsymbol{F}_k$.
In fact, if $u \in \boldsymbol{F}_k$ then $|u| = k$ and $SPM(u) \geq$ minsupp. Besides $\boldsymbol{F}_{k+1} \subseteq C(\boldsymbol{F}_k)$ (proposition 1), thus , $\boldsymbol{F}_{k+1}$ can be created from $\boldsymbol{F}_k$.
Theorem 3: The time complexity of algorithm 2 is $O(|F_k||F_1|)$.

5. Developing the Algorithms Supporting Biological Sequence Classification by using the Set of Co-occurrence Frequent Motifs

This section will focus on using the co-occurrence of frequent patterns in a set of biological sequences and their positions in sequences to classify the family of DNA sequences

Definition 8: Co-occurrence motif
Given two frequent motifs u, v ∈ FM(S,minsupp) and s s∈S. Two frequent motif u is called as co-occurrence motif with v in s if u and v are subsequences of s.

Definition 9: Measure of co-occurrence of frequent motifs
Let S be a set of biological sequences and minsupp ∈(0,1] be a given threshold, FM(S,minsupp) be the set of all frequent motifs with threshold minsupp of S.
Given u,v∈FM(S,minsupp). According to the definition 4, $\rho_M(u)$ is the set of sequences of S and these sequences contain u and $\rho_M(v)$ is the set of sequences of S and these sequences contain v. Therefore,$\rho_M(u) \cap \rho_M(v)$ is the set of sequences of S that contain both u and v.
The measure of occurrence of u, v in S is calculated by:

$$Co(u,v) = \frac{1}{2}\left(\frac{|\rho_M(u) \cap \rho_M(v)|}{|\rho_M(v)|} + \frac{|\rho_M(u) \cap \rho_M(v)|}{|\rho_M(u)|}\right)$$

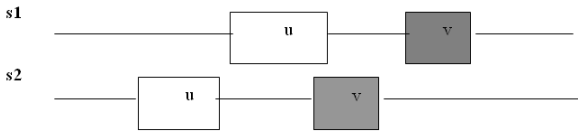
The measure of co-occurrence **Co(u,v)** expresses the possibility of co-occurrence of both u and v in S. The value of Co(u,v) is in [0,1], Co(u,v) approaches to 1 corresponding to the high degree of co-occurrence of u and v in S.

Definition 10: The ordered co-occurrence motif
Given u,v ∈ FM(S,minsupp), and s be a sequence of S containing both u and v. The frequent motif u is called an ordered co-occurrence motif of v in s iff:

i) u is co-occurrence motif with v

ii) u always appears before v in s and u does not overlap v in s.

Example 5. The frequent motif u is an ordered co-occurrence motif with v in two sequences s1 and s2.



In figure 2, the ordered set of frequent motifs {TCT, AAC, ACA} appears in some sequences of DNA E-Coli Promoter.

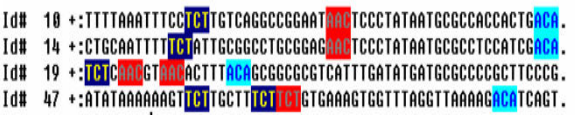


Figure 2. Map of the co-occurrence frequent motifs TCT→AAC→ACA

Definition 11: *The graph of co-occurrence of frequent motifs*
Let S be the set of a family of biological sequences, FM(S,minsupp) be the set of the frequent motifs with threshold minsupp of S, **Tco** be the threshold of popularity. Graph G(V,E) is called a **co-occurrence graph** in which V = FM(S,minsupp). With two vertexes v1, v2∈V, we have edge (v1,v2)∈E , if Co(v1,v2)≥ Tco and v1 is the co-occurrence frequent motifs of v2 .

Definition 12: *Paths*
Path p of graph G(V,E) is a sequence $\alpha_1 \rightarrow \alpha_2 \ldots \rightarrow \alpha_n$, where α_i and α_{i+1} are the vertexes of G(V,E). Path p= (α_1, α_2,..., α_n) expresses the order of appearance of frequent motifs in sequence, such that α_i is the co-occurrence of α_{i+1}.

Example 6: An example of ordered set of frequent motifs is ACA→ TTG→GGA→TCGA

Definition 13: *Sub paths*
Given two paths p=(α_1, α_2, . . ., α_n) and q=(β_1, β_2, . . ., β_m), path p is called as sub-path of q if exists the integers $t_1,t_2,\ldots, t_k$ such that :

$$\alpha_{t1} = \beta_{t1},\ \alpha_{t2} = \beta_{t2}, \ldots,\ \alpha_{tk} = \beta_{tk}$$

Example 7: Given two sequences
p: TTG→GGA and
q = ACA→ TTG→GGA→TCGA
Then p is the sub path of q.

Definition 14: *List of sequences containing path*
Let S be the set of biological sequences and G(V,E) be the co-occurrence graph of the ordered set of frequent motifs of S. Let Path(G(V,E)) be the set of all paths in G(V,E), we define function ρ_S: Path(G(V,E))→S such that ∀p∈Path(G(V,E)), $\rho_S(p)$ is the set of all sequences of S containing path p.

Definition 15: *Path length*
Path length is the number of vertexes in this path. Denote |p| as the path length of p.

Definition 16: *The support of path*
Let S be the set of biological sequences and G(V,E) be the co-occurrence graph of frequent motifs, Given p ∈ Path(G(V,E)), we define the support of path p in S is the ratio between the number of sequences in S and the number of sequences of S containing path p. The support of path p is denoted by SPP(p) and is calculated by SPP(p)=$|\rho_S(p)|/|S|$.
The path p is called as a frequent path with threshold minpath ∈ (0,1] iff SPP(p)≥minpath
Denote FP(G(V,E), minpath)={p ∈ Path(G(V,E)) | SPP(p)≥ minpath }

Definition 17*: The classification error of a path p*
Let p ∈ FP(G(V,E),minpath) and $\rho_S(p)$ be the set of sequences containing p. In the set of $\rho_S(p)$, there are sequences belonging to many classes. To define the class purity (belonging to the same class) of the sequences in $\rho_S(p)$, we use the Gini index [4]. [5]. The smaller of the value of Gini($\rho_S(p)$), corresponding to the bigger of the class purity. In practice, we define a threshold **Cerror** where 0 ≤ CError ≤1. The path p is called as high purity path iff Gini($\rho_S(p)$) ≤ CError. When Cerror=0, the sequences of $\rho_S(p)$ belong to the same class.

Property 2. If p is a frequent path with threshold minpath and q is a subsequence of path p, q is a frequent path with threshold minpath**.**

Property 3. If p is not a frequent path with threshold minpath and p is a subsequence of q, q is not a frequent path with threshold minpath.

Algorithm 3: Discovering the co-occurrence set of frequent motifs supporting the classification of biological sequences.

Let S_k be the set of frequent paths with threshold minpath and length k and C_k is the set of the candidates of frequent motifs with length k+1. The main steps of the algorithm 3 is as :

Input: The co-occurrence graph G(V,E) and threshold minpath

Output: FP(G(V,E),minpath)

1) S_1 = {path with length 1}
2) k=2
3) While ($S_{k-1} \neq \varnothing$) do
4) C_k=Create_Valid_Candidate_Path(S_{k-1})
5) Checking the validity of path
6) If valid_path then Save to S_k
7) Endwhile
8) FP(G(V,E),minpath) = $\varnothing$;
9) For (i=1 ; i < k ; i++)
10) FP(G(V,E), minpath) $\cup$= S_i
11) Endfor
12) Return FP(G(V,E), minpath)

In algorithm 2, we use **AprioBioSequence** for calculating S1.

Algorithm 4. Create_Valid_Candidates

Input: S_{k-1} (the set of path with length k-1).

Output: C_k (the set of path candidates with length k)

1) Create_Path_Candidate(S_{k-1})
2) $C_k = \varnothing$
3) For each p in S_{k-1}
4) For each q in S_{k-1}
5) If p can combine with q then
6) r = Combine(p, q);
7) Save r in C_k;
8) Endfor
9) Endfor
10)Return C_k

The criteria for concatenation of path p and path q with length k is that the suffix with length k-1 of path p is the same of the prefix with length k-1 of path q.

The result of concatenation of the path p and path q is a sequence with the length k+1 in which the first k letters are of path p and the last letter of result sequence is the last letter of sequence q (function **Combine**).

Example 8: p= 2,3,4,5 q= 3,4,5,7

The result of combining p and q is a path 2,3,4,5,7

Algorithm 5: Discovering the sequence with high class purity

Input: FP(G(V,E),minpath) and classification error threshold CError

Output: The high class purity

1) For each p $\in$ FP(G(V,E),minpath) do
2) If Gini (ρ_S(p)) $\leq$ CError then
3) Output the path with high class purity
4) Endif
5) Endfor

Theorem 4: Algorithm 3 guaranties to create a complete set of the candidates. Using the depth first search algorithm for finding the paths of graph G(V,E). From FP(G(V,E),minpath) we use the Gini index for checking the goodness of classification of each path in Path(G(V,E)).

Theorem 5: The time complexity of algorithm 3 is $O(|Sk|^2)$.

Theorem 6: The time complexity of the algorithm for discovering the valid path in the graph G(V,E) is O(|E|) and the memory space for the graph is O(|V|+|E|).

6. Experimental results

We use the data set of 106 DNA E-Coli promoter which is provided by Jude Shavlik at the Web site address http://www.ics.uci.edu. This set contains two groups: promoter and non promoter. The sequences from 1 to 53 belong to promoter group and the sequences from 54 to 106 belong to non-promoter group. Some sequences of promoter group are listed as follows:

TACTAGCAATACGCTTGCGTTCGGTGGTTAAGTATGTA
TAATGCGCGGGCTTGTCGT +
TGCTATCCTGACAGTTGTCACGCTGATTGGTGTCGTTAC
AATCTAACGCATCGCCAA +
GTACTAGAGAACTAGTGCATTAGCTTATTTTTTTGTTAT
CATGCTAACCACCCGGCG +

Some frequent motifs with threshold minsuppp= 0.1 in the data set are listed as:

CCTT, CCGC, CCGT, CCGG, CTAC, CTAG, CTCC, CTCG, CTTC, CTGC, CTGT, CGAA, CGAC, CGCG, CGTA, CGTC, CGTG, CGGA, CGGC, CGGT, TATC, TAGT, TAGG, TCAA, TCAT, TCAG, TCCA, TCCC, TCCT, TCCG, TCTA, TCTC, TCTG, TCGT, TTCA, TTCC, TTCG, TTGC, TGAT, TGAG, TGCA, TGCC, TGGC, TGGT, GAAG, GACT, GACG, GATA, GATC, GATT, GAGT, GCAC, GCCA, GCCC, GCCT, GCCG, GCTA, GCTG, GCGT, GCGG,

GTAG, GTCT, GTCG, GTTC, GTGA, GTGG, GGAA, GGAT, GGCC, CGCG, GGTA, GGTT, GGTG, AAAAC

Some ordered set of frequent motifs with classification error CError = 0.1 in the data set are listed as:

CGT->AAA->AAC; CGA->AAA->AAC;
CAG->AAA->AAC; CAT->AAA->AAC
CAC->AAA->AAC;CGT->AAA->AAC->AAT;
CGC->AAA->AAC->AAT
CTA->AAA; CCA->AAA;

7. Conclusions

Paper deals with the algorithms for discovering the frequent motifs in the set of biological sequences. AprioriBioSequence is the name of our proposed algorithm which is developed based on the data mining approach. AprioriBioSequence can discover the frequent
motifs without specifying the length and it can work with the large data set. Besides, paper also deals with the algorithms for discovering the set of co-occurrence of the frequent motifs for classifying the family of biological sequences.

References

1) Alvis Brazama, Inge Jonassen et al: Pattern Discovery in Bio-sequences, In Proc of 4th International Colloquium Grammatical inference (ICGI-98), Springer, 1998
2) H. Kiem, D. Phuc: Discovering the motif based association rules from set of DNA sequences, In Lecture Notes for computer science, Springer Verlag, 2005, Proc of the int'l conf. on Rough set, Data mining, RSCTC'2000, Banff, Canada, pp 348- 352,2000
3) Jason T. L. Wang, Steve Rozen: New Techniques for DNA Sequence Classification, New Jersey Institute of Technology, New York University, USA, 1997
4) R. Durbin, S. Eddy, A. Korgh, G. Mitchison: Biological sequence analysis probabilistic models of protein and nucleic acids, Cambridge University Press, England, 2000.
5) Timothy Lawrence Bailey: Discovering motifs in DNA & protein sequence, University of California, Ph. D. dissertation, San Diego, USA, 1999.