MATCH
Communications in Mathematical and in Computer Chemistry

MATCH Commun. Math. Comput. Chem. **60** (2008) 291-300

ISSN 0340 - 6253

A Numerical Representation of DNA Sequences and Its Applications

Weiyang Chen, Bo Liao[1], Yanshu Liu, Wen Zhu, Zhizhong Su
School of computer and communication, Hunan University,
Changsha, Hunan, 410082, China

(Received December 20, 2007)

Abstract

We introduced a sort of numerical coding method of DNA sequences. Based on this representation, we can reduce a DNA sequence into three binary digit sequence. Associating with the proposed coding rules, we can judge the mutation between bases and make sequence alignment easily.

1. Introduction

Bioinformatics data is mainly expressed by the form of sequence. Alignment of two sequences and mutation analysis are the most important tools of bioinformatics. Pairwise alignment helps to predict the functions of novel genes within any species. Particularly, these alignment methods allow us to determine the similarity between corresponding genome segments of two or more organisms belonging to the same genera. Additionally, such techniques can also be used to study hosts of other processes such as molecular evolution, RNA folding and gene regulation to name a few. On a broader scale these algorithms have also been used to determine homologies between proteins in order to predict structural and functional relationships.

Nowadays, the most of aligning methods are based on the original DNA sequences which are composed of A (adenine), G (guanine), T (thymine), and C (cytosine). For two sequences comparisons, there have many methods been used in

[1] Corresponding author: dragonbw@163.com

sequence alignment. But these methods are not easy to measure the mutation between bases. And in recent years, many authors have present different graphical representations of DNA sequences [1-17]. These graphical representations are also applied to the sequence alignment [1, 2] and mutation analysis [3, 4].

In this paper we describe a numerical coding method for DNA sequences. By this method, every DNA sequence can transform to three binary digit sequence. Associating with this coding method, we introduce an approach to make sequence alignment and judge the base mutations between sequences.

2. Numerical coding method for DNA sequence

Analysis and comparison DNA sequences should consider not only the structures of strings but also their chemical structures. In a DNA primary sequences, the four bases A, C, G and T can be classed into groups [4], purine {A, G}/pyrimidine {C, T}, amino {A, C}/keto {G, T}, and weak-H bond {A, T} /strong-H band {C, G}. In the following, we will outline a new numerical coding method of DNA sequences according to the three classifications of bases.

We will use the exclusive-OR operator. The exclusive-OR of x_1 and x_2 written $x_1 \oplus x_2$ is defined by Table 1.

Table1:The exclusive-OR

X_1	X_2	$X_1 \oplus X_2$
0	0	0
0	1	1
1	0	1
1	1	0

We will use a two bit binary digit to represent the four bases A, C ,G, and T, respectively. For the coding DNA sequence, the operating rules are defined by Table 2.

Table 2: The operating rules for the coding DNA sequences

X_1	X_2	$X_1 \oplus X_2$
00	00	00
00	01	01
00	10	10
00	11	11
01	00	01
01	01	00
01	10	11
01	11	10
10	00	10
10	01	11
10	10	00
10	11	01
11	00	11
11	01	10
11	10	01
11	11	00

There are three coding DNA sequences corresponding to the three classifications of bases.

(i) Corresponding the first classification: purine {A, G}/pyrimidine {C, T}, we define a coding rule satisfied $A \oplus G=11$, $C \oplus T=11$.

A:01, G: 10, C: 00, T: 11

For example, by the coding rule, the DNA sequence ACGT will be reduced into 01100011.

(ii) Corresponding the second classification: amino{A, C}/keto {G, T}, we define a coding rule satisfied $A \oplus C=11$, $G \oplus T=11$.

A:01, C: 10, G: 00, T: 11

(iii) Corresponding the third classification: weak-H bond {A, T}/strong-H bond {G, C}, we define a coding rule satisfied $A \oplus T=11$, $G \oplus C=11$.

A:01, T: 10, C: 00, G: 11

Based on these rules, we can obtain the following conclusions:

(1) A DNA sequence can be reduced into three binary digit sequence.

(2) For an arbitrary two bit binary digit x, $x \oplus x=00$. So, using our operating

rules, we can obtain the common subsequence of two DNA sequences by finding the regions of consecutive zero.

(3) For any class coding sequence, using our operating rules, we also can obtain the results 11, 10 and 01. These results contain different mutation information. The operating result 11 means that the mutation arises in the same class. That is to say, the mutation take place between A and G or between C and T. While the operating result 10 and 01 mean that the mutations occur in the different class.

For example, there are two sequences:

S1: A T G G T G C A C C T G A C T C C T G A

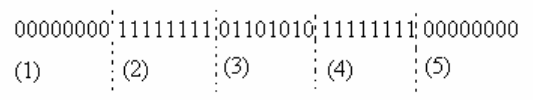
S2: A T G G C A T G A G A C G T C T C T G A

Corresponding the first classification, the numerical coding sequences of S1 and S2 based on the first coding rule are listed as follows:

S1: 011110101110000100001110010011000011100 1

S2: 011110100001111001100100101100110011100 1

Using our operating rules, we can obtain the following result:



Observing the result, we can find some interesting segments. The segments (1) and (5) are the regions of consecutive 0, so they have the same subsequence. The segments (2) and (4) are the regions of consecutive 11, so the mutation should be arise in the same class. That is to say, these mutations take place between A and G or between C and T. The segment (3) is the sequence of 01 or 10, so the mutations occur in the different class. That is to say, these mutations take place between purine and pyrimidine.

3. Sequence alignment and mutation analysis

Suppose two arbitrary sequences $L1=a_1a_2\cdots a_n$ and $L2=b_1b_2\cdots b_m$, where n and m are the length of sequences L1 and L2 respectively. Let $L1'=g(a_1)g(a_2)\cdots g(a_n)$ and

$L2'=g(b_1)g(b_2)\cdots g(b_m)$ be the coding sequences using a coding rule. We can obtain several theorems as follows:

Theorem 1: If $g(a_i) \oplus g(b_j)=00$, where $1 \le i \le n$, $1 \le j \le m$, then a_i should match b_j.

Theorem 2: If $g(a_i) \oplus g(b_i)=11$, where $1 \le i \le \min(n, m)$, then the mutation should be arise in the same class. That is to say, the mutation takes place between A and G or between C and T.

Theorem 3: If $g(a_i) \oplus g(b_i)=01$ or 10, where $1 \le i \le \min(n, m)$, then the mutation should be arise in the different class. That is to say, the mutation takes place between purine and pyrimidine.

Theorem 4: For any i, $1 \le i \le \min(n, m)$, if $g(a_i) \oplus g(b_i)=00$, then the sequence L1 is the subsequence of sequence L2 or the sequence L2 is the subsequence of sequence L1.

Theorem 5: For any i, j , $1 \leqslant i \leqslant n-d$, $1 \leqslant j \leqslant m-d$, $d>0$, if the subsequence $a_ia_{i+1}\cdots a_{i+d}$ match the subsequence $b_jb_{j+1}\cdots b_{j+d}$, and there is not any integer $x>d$ to make the new subsequence $b_tb_{t+1}\cdots b_{t+x}$ satisfied $g(a_{i+s}) \oplus g(b_{t+s})=00$, where $0 \leqslant s \leqslant x$, then sequence L1 and L2 should have the longest common subsequence $a_ia_{i+1}\cdots a_{i+d}$.

According to Theorem 5, we can obtain the longest common subsequence of arbitrary sequences L1 and L2. For two sequences $L1=a_1a_2\cdots a_n$ and $L2=b_1b_2\cdots b_m$, where n and m are the length of sequences L1 and L2 respectively. We can obtain the longest common subsequence of L1and L2: $T= a_ia_{i+1}\cdots a_{i+d}$ which corresponds $b_jb_{j+1}\cdots b_{j+d}$ in sequence L2 and satisfies $g(a_{i+k}) \oplus g(b_{j+k})=00$,for any $1 \leqslant i \leqslant n-d$, $1 \leqslant j \leqslant m-d$, $0 \leqslant k \leqslant d$. So the sequence L1 is divided into three parts $A1= a_1\ldots a_{i-1}$, $A2= a_ia_{i+1}\cdots a_{i+d}$ and $A3= a_{i+d+1}\ldots a_n$. The sequence L2 is also divided into three parts $B1= b_1\ldots b_{j-1}$, $B2= b_jb_{j+1}\cdots b_{j+d}$ and $B3= b_{j+d+1}\ldots b_m$. The subsequence A1 will align with the subsequence B1 and the subsequence A3 will align with the subsequence B3. Then, in the same way we can obtain the two longest common subsequences between the code of A1 and the code of B1, and between the code of A3 and the code of B3. Don't end this process until the two short subsequences match completely or don't match at all. In the result, we will obtain the optimal alignment.

The algorithm is implemented by using recursion strategy. The pseudocodes of the algorithm are described as follows:

```
Alignment (a, b)                // a and b represent the strings of both DNA
                                   sequences respectively.
{   A=Translate (a);            // translate strings into the set of binary sequence, A
                                   and B are arrays of 0 and 1.
    B=Translate (b);
    S=Max_match (A, B)          // S returned by Max_match is the best longest
                                   common binary sequence. Max_match is a
                                   function which finds the longest consecutive 00
                                   sequence by the ⊕ operator and shifting.
                                  // the old B is covered with B after moving.
    If S= = A or S= = φ  then
      End;                      // while matching completely or not matching
                                   anymore, end.
    If S ≠ A and S ≠ φ  then
      {   A1=Get_former (A, S);     // intercept A1 before S from A.
          A3=Get_latter (A, S);     // intercept A3 behind S from A.
          B1=Get_former (B, S);     // intercept B1 before S from B.
          B3=Get_latter (B, S);     // intercept B1 before S from B.
        a=re_translate (A1);        // the function can translate the set of
                                       binary sequence to the strings.
          b=re_translate (B1);
          Alignment (a,b);          // transferring recursively.
          a=Re_translate (A3);
          b=Re_translate (B3);
          Alignment (a,b);          // transferring recursively.
      }
}
```

In actual aligning process, there will be a gap inserting. We also can judge the alignment and mutation based on the proposed operation.

In this paper, we adopt add one bit binary digit 111 to express the gap. While, for the four bases A, C, G and T, we add one binary digit '0' in their former code. So the new coding rules are defined as follows:

(i) Corresponding the first classification: purine {A, G}/pyrimidine {C, T}, we define a coding rule satisfied A ⊕ G=011, C ⊕ T=011.

A:001, G: 010, C: 000, T: 011, "-": 111

For example, by the coding rule, the DNA sequence ACGT will be reduced into 001010000011.

(ii) Corresponding the second classification: amino{A, C}/keto {G, T}, we define a coding rule satisfied A ⊕ C=011, G ⊕ T=011.

A:001, C: 010, G: 000, T: 011, "-": 111

(iii) Corresponding the third classification: weak-H bond {A, T}/strong-H bond {G, C}, we define a coding rule satisfied A ⊕ T=011, G ⊕ C=011.

A:001, T: 010, C: 000, G: 011, "-": 111

There are two reasons for us to adopt this strategy.

(1) Every number do the ⊕ operation with '1', the result will be opposite to itself.

(2) In the process of pair wise alignment, there isn't the situation of two gaps aligning.

For example, suppose sequence L1=TAGGCCTCTGCCTAATCACACAG and sequence L2=CGGCCTCTGCCTTATTACACAA. The corresponding coding sequence based on the first coding rule as follows:

L1'=011001010010000000011000011010000000011001001011000001000001000001010;

L2'=000010010000000011000011010000000011011001011011001000001000001001.

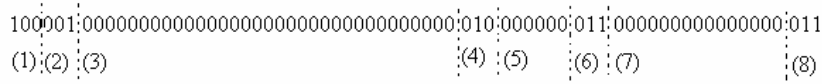
Then by shifting L2' right or left and the ⊕ operation we can obtain the longest regions of consecutive 000. It will correspond to the longest common subsequence. For this example, the longest common subsequence S=GGCCTCTGCCT. So the sequence L1 is divided into three parts, and it becomes L1=a_1a_2 S $a_{14}\cdots a_{23}$. The sequence L2 is divided into three parts too, and it becomes L2=b_1 S $b_{13}\cdots b_{22}$. Then we will do the same align in the first subsequence of L1 and the first subsequence of L2, the third subsequence of L1 and the third subsequence of L2. Until there isn't match at all or match completely.

Finally, we can obtain the aligning result as follows:

L1= TAGGCCTCTGCCTAATCACACAG

L2= - CGGCCTCTGCCTTATTACACAA

After the ⊕ operation based on the first coding rule, the coding is shown in the following:



From the result we can find every segment has different meaning. The segments (3), (5) and (7) are the regions of consecutive 000, so they have the same subsequence. The segments (6) and (8) are the regions of 011, so the mutation should be arise in the same class. That is to say, these mutations are between purine and purine, or between pyrimidine and pyrimidine. The segments (2) and (4) are the sequence of 001 or 010, so the mutations occur in the different class. That is to say, these mutations take place between purine and pyrimidine. The segment (1) is the sequence of 100 which belongs to the 1××, so this region denotes that there is a gap inserting. That is to say, the mutation should be insertion or deletion.

4. Conclusions

In this paper, we introduced a sort of numerical coding method of DNA sequences. By this method, every DNA sequence can transform into three binary digit sequence. Based on the proposed coding rules, we can do the sequence alignment and judge mutations. Compared to other alignment algorithms and mutation analysis methods, the advantage of our method is that it is simple and efficient. The time complexity of our alignment algorithm is linear.

Acknowledgment This work is supported in part by the National Nature Science Foundation of China(Grant 10571019) and the National Nature Science Foundation of Hunan province(Grant 07JJ5080). The authors thank the anonymous referees for many valuable suggestions, which have improved this manuscript.

References

[1] B. Liao, K. Q. Ding, Graphical approach to analyzing DNA sequences**,** *J. Comput. Chem.* **26** (2005)1519−1523.

[2] M. Randić, J. Zupan, D. Vikić-Topić, D. Plavšić, A novel unexpected use of a graphical representation of DNA: Graphical alignment of DNA sequences, *Chem. Phys. Lett.* **431** (2006) 375−379.

[3] B. Liao, A 2D graphical representation of DNA sequence *Chem. Phys. Lett.* **401** (2005) 196−199.

[4] Y. Yao, X. Nan, T. Wang, A new 2D graphical representation-classification curve and the analysis of similarity/dissimilarity of DNA sequences *J. Mol. Struct. (Theochem)* **764** (2006) 101−108.

[5] B. Liao, T. Wang, 3-D graphical representation of DNA sequences and their numerical characterization *J. Mol. Struct. (Theochem)* **681** (2004) 209–212.

[6] A. Nandy, A new graphical representation and analysis of DNA sequence structure: I. Methodology and application to globin genes, *Curr. Sci.* **66** (1994) 309–314.

[7] A. Nandy, P. Nandy, Graphical analysis of DNA sequences structure: II. Relative abundance of nucleotides in DNAs, gene evolution and duplication, *Curr. Sci.* **68** (1995) 75–85.

[8] M. Randić, M. Vračko, N. Lerš, D. Plavšić, Analysis of similarity/dissimilarity of DNA sequences based on novel 2-D graphical representation, *Chem. Phys. Lett.* **371** (2003) 202−207.

[9] G. Huang, B. Liao, W. Zhang, F. Gong. A novel method for sequence alignment and mutation analysis, *MATCH Commun. Math. Comput. Chem.* **59** (2008) 635−645.

[10] B. Liao, K. Ding, A 3D graphical representation of DNA sequences and its application, *Theor. Comp. Sci.* **358** (2006) 56−64.

[11] B. Liao, T. Wang, Analysis of similarity/dissimilarity of DNA sequences based on 3-D graphical representation, *Chem. Phys. Lett.* **388** (2004) 195−200.

[12] M. Randić, M. Vračko, N. Lerš, D. Plavšić, Novel 2-D graphical representation of DNA sequences and their numerical characterization, *Chem. Phys. Lett.* **368** (2003) 1−6.

[13] M. Randić, Graphical representations of DNA as 2-D map, *Chem. Phys. Lett.* **386** (2004) 468−471.

[14] M. Randić, M. Vračko, J. Zupan, M. Novič, Compact 2-D graphical representation of DNA, *Chem. Phys. Lett.* 373 (2003) 558−562.

[15] D. Bielinska-Waz, T. Clark, P. Waz, W. Nowak, A. Nandy, 2D dynamic representation of DNA sequences, *Chem. Phys. Lett.* **442** (2007) 140−144.

[16] D. Bielinska-Waz, W. Nowak, P. Waz, A. Nandy, T. Clark, Distribution moments of 2D-graphs as descriptors of DNA sequences, *Chem. Phys. Lett.* **443** (2007) 408−413.

[17] D. Bielinska-Waz, P. Waz, T. Clark, Similarity Studies of DNA Sequences Using Genetic Methods, *Chem. Phys. Lett.* **445** (2007) 68−73.