Biophysics, Vol. 44, No. 4, 1999, pp. 631–635. Translated from Biofizika, Vol. 44, No. 4, 1999, pp. 649–654.

Averaging of Site Recognition Results Enhances the Reliability of Human Genome Annotation

M. P. Ponomarenko[1], Yu. V. Ponomarenko[1], O. A. Podkolodnaya[1], A. S. Frolov[1], D. V. Vorob'ev[1], N. A. Kolchanov[1], and G. C. Overton[2]

[1] *Institute of Cytology and Genetics, Siberian Division, Russian Academy of Sciences, Novosibirsk, 630090 Russia*
[2] *University of Pennsylvania, Philadelphia, USA*

Received June 19, 1998

Abstract—The Central Limit Theorem is used to develop a "systemic" approach to recognition of functional sites with increased reliability. The approach is based on averaging a large number of "particular" predictions. A sufficiently large number of such particular procedures are obtained by recoding of the DNA sequence in 20 different alphabets and subsequent application of the standard consensus and profile recognizers. The method was tested on the binding sites of transcription factors GATA-1 and C/EBP. As expected, the averaged recognizer is more reliable than each of the particular recognizers.

Key words: functional site, increase of reliability, averaging

INTRODUCTION

Recognition of functional sites is one of the key problems of genome annotation [1]. There exist a large number of site prediction algorithms, reviewed in [2]. The most widely used methods are consensus and profiles based on evolutionary conservation of functional sites of the same type [3–7]. The tests of the existing genome annotation algorithms [8, 9] demonstrate both the fast progress in this area and the need for improved reliability, which still is a limiting factor in genome annotation.

In this connection we suggest a "systemic" approach to enhancing the site recognition reliability, based on the Central Limit Theorem. At the core of the approach is averaging of many different procedures for recognition of the same signal. Formal generation of such procedures uses the conventional consensus and profile methods, and to increase the number of generated procedures, we use 20 different "oligonucleotide recoding" alphabets. We have developed a computer system for automated generation of C programs for oligonucleotide consensus and profile recognizers given an aligned set of site sequences. This system has been used to analyze the binding sites of transcription factors GATA-1 and C/EBP. It has been demonstrated that the "averaged" recognizers produced more exact predictions than any of the 40 "particular" recognizers in each of these alphabets, as expected from the Central Limit Theorem.

EXPERIMENTAL

The proposed 20 "oligonucleotide recoding" alphabets are presented in the table. Each of them contains a complete set of 2, 4, 8, 16, 32, or 64 short oligonucleotides of fixed length from one through five nucleotides. One can see that at expected nucleotide frequencies $p(\mathrm{A}) = p(\mathrm{T}) = p(\mathrm{G}) = p(\mathrm{C}) = 0.25$, all oligonucleotides over the same alphabet have the same expected frequencies. The table also lists the critical oligonucleotide frequencies f_0. If the observed frequency of an oligonucleotide exceeds the critical frequency, the oligonucleotide is included into the site consensus. Figure 1 presents examples of automatically generated C texts implementing the procedures for site recognition by consensi and profiles using the alphabets from the table. One can see that both for consensus recognizers (Fig. 1a, b) and profile recognizers (Fig. 1c, d) the programs using the standard

(a)

```
double GATA1_1bp_Cons_ATGC (char *Seq){
double X=0.; char N, *s; s=&Seq[0];
if(strlen(Seq)<28) return (–99.);
N=s[11];switch(N){ case’G’: X++;break;}
N=s[12];switch(N){ case’A’: X++;break;}
N=s[13];switch(N){ case’T’: X++;break;}
N=s[14];switch(N){ case’A’: X++;break;}
N=s[15];switch(N){ case’A’: X++;break;}
N=s[16];switch(N){ case’G’: X++;break;}
N=s[17];switch(N){ case’G’: X++;break;}
return ( (X–3.7353)/ 1.8333);}
```

(b)

```
double GATA1_5bp_Cons_ATGC_X (char *Seq){
double X=0.; char N1, N2, N3, *s; s=&Seq[0];
if(strlen(Seq)<28) return (–95.);
N1=s[9]; N2=s[11]; N3=s[13];
if(N1== ‘C’ && N2==’G’ && N3== ‘Ò’)X++;
N1= s[10]; N2= s[12]; N3=s[14];
if(N1==’A’ && N2== ‘À’&& N3== ‘À’)X++,
if(N1==’T’ && N2==’À’ && N3==’A’)X++;
N1= s[11]; N2=s[13]; N3=s[15];
if(N1==’G’ && N2== ‘Ò’ && N3==’À’)X++;
N1= s[12]; N2=s[14]; N3=s[16];
if(N1==’À’ && N2==’À’ && N3=’G’)X++;
N1=s[13]; N2=s[15]; N3=s[17];
if(N1==’T’ && N2==’À’ && N3==’G’)X++,.
return ((X- 1.3039)/1.2843);}
```

(c)

```
double GATA1_1bp_ Freq_ATGC (char *Seq){
double A[28]={0.255, 0.073, 0.273, 0.345, 0.182, 0.164, 0.109, 0.309, 0.273, 0.145,
              0.491, 0.018, 0.909, 0.018, 0.764, 0.673, 0.127, 0.182, 0.255, 0.364,
              0.473, 0.418, 0.164, 0.218, 0.236, 0.200, 0.309, 0.291};
double T[28]={0.291, 0.291, 0.382, 0.255, 0.255, 0.200, 0.255, 0.255, 0.255, 0.255,
              0.418, 0.018, 0,018, 0.764, 0.164, 0.018, 0.164, 0.109, 0.255, 0.164,
              0.182, 0.182, 0.255, 0.436, 0.291, 0.164, 0.309, 0.255};
double G[28]={0.182, 0.236, 0.145, 0.164, 0.345, 0.400, 0.364, 0.145, 0.255, 0.200,
              0.055, 0.945, 0.036, 0.036, 0.055, 0.291, 0.618, 0.618, 0.182, 0.236,
              0.164, 0.273, 0.236, 0.109, 0.291, 0.473, 0.218, 0.200};
double C[28]={0.273, 0.400, 0.200, 0.236, 0.218, 0.236, 0.273, 0.291, 0.218, 0.400,
              0.036, 0.018, 0,036, 0,182, 0.018, 0.018, 0.091, 0.091, 0.309, 0.236,
              0.182, 0.127, 0.345, 0.236, 0.182, 0.164, 0.164, 0.255};
double X=0.; char N, *s; s=&Seq[0];if(strlen(Seq) < 28) return (–91.);
for(i=0;i<28;i++){ N=s[i];switch(N){case’A’: X+=A[i];break; case’T’: X+=T[i];break;
case’G’: X+=G[i];break; case’C’: X+=C[i];break; }}return ((X- 8.8250)/ 1.7754);}
```

(d)

```
double GATA1_3bp_ Freq_WS_X (char *Seq){
double WW[28]={0.364, 0.255, 0.364, 0.291, 0.164, 0.164, 0.218, 0.182, 0.491, 0.018,
              0.855, 0.018, 0.873, 0.545, 0.255, 0.127, 0.182, 0.182, 0.364, 0.400,
              0.273, 0.345, 0.164, 0.200, 0.364, 0.255, 0.001, 0.002};
double WS[28]={0.182, 0.109, 0.291, 0.309, 0.273, 0.200, 0.145, 0.382, 0.036, 0.382,
              0.055, 0.018, 0.055, 0.236, 0.673, 0.564, 0.109, 0.109, 0.145, 0.127,
              0.382, 0.255, 0.255, 0.455, 0.164, 0.109, 0.001, 0.002};
double SW[28]={0.291, 0.345, 0.073, 0.073, 0.200, 0.400, 0.309, 0.218, 0.418, 0.018,
              0.073, 0.764, 0.055, 0.145, 0.036, 0.164, 0.327, 0.345, 0.291, 0.200,
              0.145, 0.309, 0.364, 0.164, 0.255, 0.291, 0.001, 0.002};
double SS[28]={0.164, 0.291, 0.273, 0.327, 0.364, 0.236, 0.327, 0.218, 0.055, 0.582,
              0.018, 0.200, 0.018, 0.073, 0.036, 0.145, 0.382, 0.364, 0.200, 0.273,
              0.200, 0.091, 0.218, 0.182, 0.218, 0.345, 0.001, 0.002};
double X=0.; int i; char N1, N2, *s; s=&Seq[0];if(strlen(Seq)<28) return (–97.);
for(i=0;i<26;i++){ N1=s[i]; N2=s[i+2]; switch(N1){
case’A’: case’T’: switch(N2){case’A’: case’T’: X+= WW[i];break; case’G’: case’C’: X+=WS[i];break,} break;
case’G’: case’C’: switch(N2){case’A’: case’T’: X+=SW[i];break; case’G’: case’C’: X+= SS[i];break;} break;} }
return ((X- 8.0340)/ 1.6230);}
```

Fig. 1. Examples of automatically generated C programs for GATA-1 recognizers using consensi (a, b) and profiles (c, d) in various alphabets. The programs in the conventional alphabet {A, C, G, T} (a, c) insignificantly differ from the programs in other alphabets (b, d).

Oligonucleotide alphabets

Alphabet $E_n = \{e_1, ..., e_n\}$ of oligonucleotides of length L				Threshold, f_0	Used before
F	L	M=A/C, K=G/T, R=A/G, Y=T/C, W=A/T, S=G/C	n		
N_4	1	A, T, G, C	4	0.500	see review [2]
N_{16}	2	AA, AT, AG, AC, TA, TT, ..., GC, CA, CT, CG, CC	16	0.333	
N_{64}	3	AAA, AAT, AAG, ..., CGC, CCA, CCT, CCG, CCC	64	0.125	[10]
Nx_{16}	3	AxA, AxT, AxG, AxC, ..., CxA, CxT, CxG, CxC	16	0.333	
Nx_{64}	5	AxAxA, AxAxT, AxAxG, ..., CxCxT, CxCxG, CxCxC	64	0.125	
MK_4	2	MM, MK, KM, KK	4	0.500	
MK_8	3	MMM, MMK, MKM, MKK, ..., KMK, KKM, KKK	8	0.250	
KM_{16}	4	MMMM, MMMK, MMKM, ..., KKKM, KKKK	16	0.333	
MKx_4	3	MxM, MxK, KxM, KxK	4	0.500	
MKx_8	5	MxMxM, MxMxK, MxKxM, ..., KxKxM, KxKxK	8	0.250	
RY_4	2	RR, RY, YR, YY	4	0.500	
RY_8	3	RRR, RRY, RYR, RYY, YRR, YRY, YYR, YYY	8	0.250	
RY_{16}	4	RRRR, RRRY, RRYR, ..., YYRY, YYYR, YYYY	16	0.333	
RYx_4	3	RxR, RxY, YxR, YxY	4	0.500	
RYx_8	5	RxRxR, RxRxY, RxYxR, RxYxY, ..., YxYx R, YxYxY	8	0.250	
WS_4	2	WW, WS, SW, SS	4	0.500	
WS_8	3	WWW, WWS, WSW, WSS, SWW, SWS, SSW, SSS	8	0.250	
WS_{16}	4	WWWW, WWWS, WWSW, ..., SSSW, SSSS	16	0.333	
WSx_4	3	WxW, WxS, SxW, SxS	4	0.500	
WSx_8	5	WxWxW, WxWxS, WxSxW, ..., SxSxW, SxSxS	8	0.250	

{A, T, G, C} alphabet (Fig. 1a, c) are similar to the programs using the alphabets from the table (Fig. 1b, d).

If there exist K recognizers $\{f_k\}_{1 \le k \le K}$ for some signal, the results of application of these methods to a sequence S can be averaged:

$$F_K(S) = \sum_{k=1,K} f_k(S)/K, \qquad (1)$$

where all values $f_k(S)$ are normalized for N sites (Site) or random DNA (Rand):

$$\sum_{n=1,N} f_k(\mathrm{Site}_n)/N = 1, \qquad (1a)$$

$$\sum_{n=1,N} f_k(\mathrm{Rand}_n)/N = -1. \qquad (1b)$$

The heuristic rule for site recognition is

$$\{\text{if } F_K(S) > 0 \text{ then } S \text{ is a site}\}. \qquad (1c)$$

According to the Central Limit Theorem, one should expect that as the number of recognizers K increases, the distribution of the values F_K tends to the Gaussian distribution with the means 1 for sites and –1 for random sequences. The standard deviations are decreasing as $K^{-1/2}$.

Thus we suggest to use the constant difference between means and decreasing standard deviations of F_K for sites and random sequences as one of the possible ways to improve the reliability of site annotation in genomic DNA.

RESULTS AND DISCUSSION

Nucleotide sequences of experimentally determined binding sites of transcription factors GATA-1

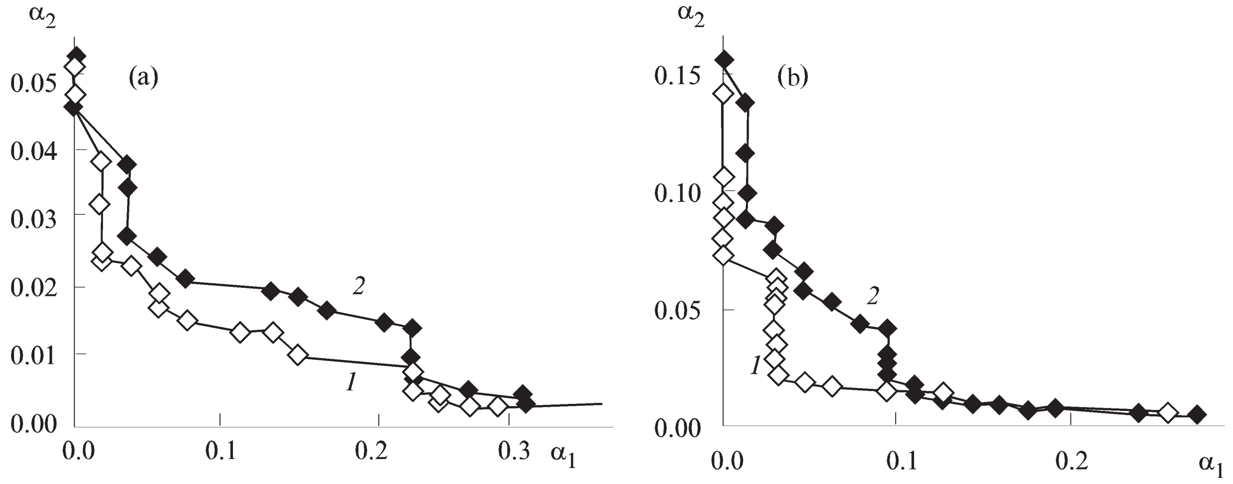

Fig. 2. Dependence between the type 1 and 2 errors (α_1 and α_2 respectively) of "averaged recognizer" (curves *1*) and profile recognizer (curves *2*) of GATA-1 (a) and C/EBP (b) sites.

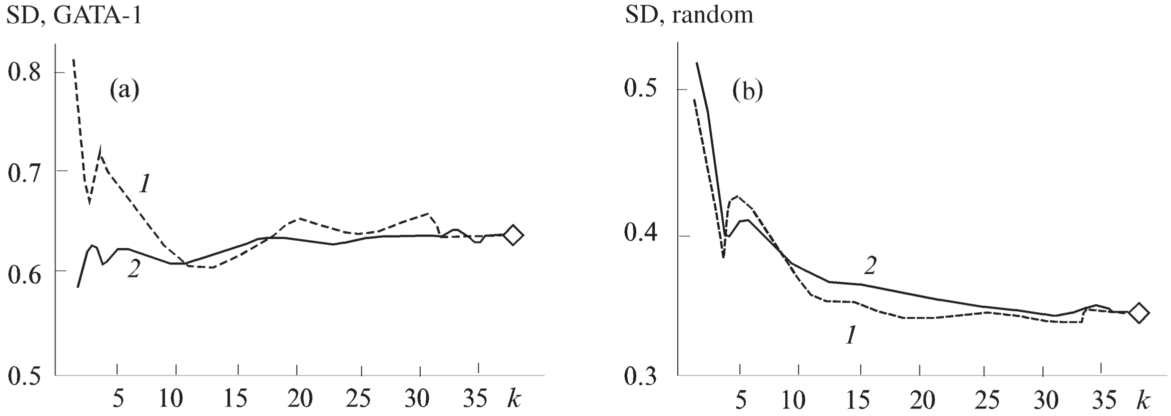

Fig. 3. Dependence of the standard deviation (SD) of the discrimination values F_K of the "averaged recognizer" on the number of particular recognizers *k* for the control subsample of GATA-1 sites (a) and 1000 random DNA sequences of the same length (b): *1*, without {A, T, G, C}; *2*, with {A, T, G, C}; diamond: complete averaging.

(102 sequences) and C/EBP (62 sequences) were taken from the database ALIGN available at http://wwwmgs.bionet.nsc.ru/. Each of the two samples was divided into two equal nonintersecting subsamples, training and control. The training subsample was used to generate C programs as described above. The examples for consensi are given in Fig. 1a, b; for profiles, in Fig. 1c, d.

The control subsamples were used to test the performance of the recognizers on independent experimental data. The type 1 and 2 errors (α_1 and α_2) were estimated, as well as the means and standard deviations of the discriminating values of these recognizers on sites and 1000 random nucleotide sequences of the same length. Figure 2a, b (curves *1*) presents the detailed comparison of dependencies between the levels of the type 1 and type 2 errors for "averaged recognizer" of GATA-1 (a) and C/EBP (b) sites. For comparison, the results produced by the conventional profile algorithm are given (curves *2*). One can see that the "averaged recognizer" has less type 2 errors at any level of the type 1 errors than the profile.

To clarify the causes of this improvement in performance, we determined the dependence between the standard deviation of the discrimination values F_K of the "averaged recognizer" on the number *K* of particular recognizers. At fixed value of *K* from 2 through 38, *K* out of 40 consensi and profiles (Table) were selected at random to obtain the "averaged recognizer" by formula (1). Ten independent random tests were performed. Figure 3a presents the dependences for the control subsample of GATA-1 sites. Curve *1* corresponds to the case when the standard alphabet {A, C, G, T} is excluded from the analysis, and curve *2*, to the case when both the consensus and the profile in this alphabet is used in all ten tests. Figure 3b features the same dependences for 1000 random nucleotide sequences of the same length.

In Fig. 3a one can see that if the standard consensus and profile in the alphabet {A, C, G, T} are used, addition of new recognizers does not influence the standard deviation of the discrimination values F_K (curve *2*), whereas in the case when the standard recognizers are not used (curve *1*) the standard deviation of F_K soon reaches a stationary level and then again does not depend on the number K of the particular recognizers being averaged. This means that the type 1 error substantially depends on the quality, size, homogeneity, and representativeness of the experimental data, in agreement with the common knowledge.

However, as demonstrated by Fig. 3b, the type 2 error has quite different properties. Independent of the incorporation (curve *2*) or excluding (curve *1*) of the conventional consensus and profile in the alphabet {A, T, G, C}, the standard deviation of the values F_K decreases proportionally to $K^{-1/2}$ according to the Central Limit Theorem. This means that the increase of the prediction reliability is caused by adecrease of the type 2 error determined mainly by the standard deviation of the discrimination values F_K on random DNA sequences.

Thus, the results presented show that the reliability of genomic DNA annotation can be achieved by averaging of many particular recognizers.

ACKNOWLEDGMENTS

This study was supported by NIH under grant 2-R01-RR04026-08A2; Russian Foundation for Basic Research under grants 97004-49740, 96-04-50006, 97-07-90309, 98-07-90126; State program Human Genome; Siberian Division of the Russian Academy of Sciences under grant IG-97/13.

REFERENCES

1. Fickett, J.W., *Trends Genet.*, 1996, vol. 12, pp. 316–320.
2. Gelfand, M.S., *J. Comput. Biol.*, 1995, vol. 2, pp. 87–115.
3. Bucher, P., *J. Mol. Biol.*, 1990, vol. 212, pp. 563–578.
4. Karlin, S. and Brendel, V., *Science*, 1992, vol. 257, pp. 39–49.
5. Quandt, K., Frech, K., *et al.*, *Nucl. Acids Res.*, 1995, vol. 23, pp. 4878–4884.
6. Uberbacher, E.C., Xu, Y., and Mural, R.J., *Meth. Enzymol.*, 1996, vol. 266, pp. 259–281.
7. Chen, Q.K., Hertz, G.Z. and Stormo, G.D., *Comput. Appl. Biosci.*, 1997, vol. 13, pp. 29–35.
8. Fickett, J.W. and Hatzigeorgiou, A.G., *Genome Res.*, 1997, vol. 7, pp. 861–878.
9. Burset, M. and Guigo, R., *Genomics*, 1996, vol. 34, pp. 353–367.
10. Kondrakhin, Y.V., Shamin, V.V. and Kolchanov, N.A., *CABIOS*, 1994, vol. 10, pp. 597–603.
11. Lawrence, C., *Comput. Chem.*, 1994, vol. 18, pp. 255–258.
12. Solovyev, V., Salamov, A. and Lawrence, C., *Nucl. Acids Res.*, 1994, vol. 22, pp. 5156–5163.
13. Guigo, R. and Fickett, J.W., *J. Mol. Biol.*, 1995, vol. 253, pp. 51–60.
14. Kondrakhin, Y.V., *et al.*, *CABIOS*, 1995, vol. 11, p. 477–488.