CODEN IEJMAT *Internet Electronic Journal of Molecular Design* **2005**, *4*, 613–624 ISSN 1538–6414
BioChem Press http://www.biochempress.com

Inter*net* Electronic Journal of Molecular Design

September 2005, Volume 4, Number 9, Pages 613–624

Editor: Ovidiu Ivanciuc

Special issue dedicated to Professor Danail Bonchev on the occasion of the 65th birthday

Finding Protein Coding Genes in the Yeast Genome Based on the Characteristic Sequences

Ping–an He,[1] Chun Li,[2] and Jun Wang [2]

[1] Faculty of Science, Zhejiang Institute of Science and Technology, Hangzhou, Zhejiang 310018, P. R. China

[2] Department of Applied Mathematics, Dalian University of Technology, Dalian 116024, P. R. China

Received: November 9, 2004; Revised: February 21, 2005; Accepted: May 4, 2005; Published: September 30, 2005

Citation of the article:

P. He, C Li, and J Wang, Finding Protein Coding Genes in the Yeast Genome Based on the Characteristic Sequences, *Internet Electron. J. Mol. Des.* **2005**, *4*, 613–624, http://www.biochempress.com.

Inter*net* Electronic Journal of Molecular Design

BioChem Press
http://www.biochempress.com

Finding Protein Coding Genes in the Yeast Genome Based on the Characteristic Sequences

Ping–an He,[1,*] Chun Li,[2] and Jun Wang [2]

[1] Faculty of Science, Zhejiang Institute of Science and Technology, Hangzhou, Zhejiang 310018, P. R. China

[2] Department of Applied Mathematics, Dalian University of Technology, Dalian 116024, P. R. China

Received: November 9, 2004; Revised: February 21, 2005; Accepted: May 4, 2005; Published: September 30, 2005

Internet Electron. J. Mol. Des. **2005**, *4* (9), 613–624

Abstract

Motivation. Due to the rapid growth of DNA sequences data in various databases, the development of accurate algorithms for gene prediction is of great importance. The motivation of this paper is to suggest a numerical characterization algorithm specific for predicting protein–coding genes in the yeast genome.

Method. The characteristic sequences of a DNA sequence are a group of (0,1) sequences. Each of them is a reduced representation of the given DNA sequence, and two of them can uniquely reconstruct the sequence. Based on the numerical description of the characteristic sequences, a protein coding gene finding algorithm specific for the yeast genome was suggested.

Results. As a result, the accuracy of the prediction is better than 95%. Based on this, it is found that the total number of protein coding genes in the yeast genome is 5897, coincident with 5800–6000, which is widely accepted. The names of putative non–coding ORFs are listed here in detail.

Conclusions. The results presented in this paper show that this new method is a useful gene prediction algorithm, and can be extended to find genes with more complicated structures.

Keywords. DNA sequence; characteristic sequences; gene prediction; gene recognition; Yeast genome.

1 INTRODUCTION

With the development of biotechnologies, the analysis of sequences, especially, gene finding become more and more important in bioinformatics. Most gene–finding algorithms are based on the differences of statistical properties between DNA sequences in coding and non–coding regions [1–7,13–21]. The phases in one strand of a DNA double helix are heterogeneous in the coding regions, whereas homogeneous in the non–coding regions. This fact constitutes the basis of almost all gene–finding algorithms [1,2]. The prediction of coding sequences has garnered a lot of attention during

Dedicated on the occasion of the 65th birthday to Danail Bonchev. Presented in part at the Internet Electronic Conference of Molecular Design 2004, IECMD 2004.

* Correspondence author; E–mail: pinganhe@yahoo.com.cn.

the last decade [1–7,13–21]. We can distinguish two types of methods: one relies on training with sets of example and counter–example sequences, and the other exploits the intrinsic properties of the DNA sequences to be analyzed.

Currently, the most popular approach is to consider a set of candidate exons weighted by some statistical parameters and then construct the optimal gene, defined as a consistent chain of exons using dynamic programming [3,4,5]. The recognition of coding sequences is usually approached by measuring the positional and compositional biases imposed by the genetic code on the DNA sequences in protein–coding regions [6]. Recent developments in the prediction of coding sequences require computation of discriminant functions with parameters that are estimated with a training set composed of examples and counter–examples (coding and non–coding sequences) [6,7]. For example, Zhang *et al*. [1,2] suggested a gene finding algorithm based on the YZ score index. In their algorithm, a graphical approach was used to explore the difference between coding and non–coding sequences.

An ORF is a DNA sequence that potentially encodes a protein. They always have a start codon (ATG) at one end and a translation–terminating stop codon at the other end, with at least 300 bases in between. In bacterial DNA sequences, practically all ORFs are coding sequences, which make the gene recognition easy.

In a previous paper [8], the characteristic sequences were introduced to represent a DNA sequence and make comparisons of the similarity and dissimilarity of DNA sequences. Based on the ideas of the characteristic sequences and the Euclid distance discriminant method, we propose, in this paper an algorithm for the recognition of coding ORFs and non–coding ORFs sequences in the yeast *Saccharomyces cerevisiae* genome.

2 MATERIALS AND METHODS

2.1 The Database

The budding yeast *Saccharomyces cerevisiae* is an important model organism for the Human Genome Project. In this paper, we use the *S. cerevisiae* genome DNA sequences. The *S. cerevisiae* genome DNA sequences can be obtained from the Munich Information Center for Protein Sequences (MIPS), released in 1997 [9,11]. The data for classification of ORFs in the yeast genome were downloaded from http://mips.gsf.de, release, October 10, 2001. In the MIPS database, all the ORFs are classified into six classes, which correspond to known proteins, no similarity, questionable ORFs, similarity or weak similarity to known proteins, similarity to unknown proteins and strong similarity to known proteins, respectively. The 1st, 2nd, 3rd, 4th, 5th and 6th classes include 3410(18), 516, 471(8), 820(2), 1003 and 229, entries, respectively, where the figures in the parentheses indicate the numbers of ORFs in the mitochondrial genome. The mitochondrial ORFs

are excluded here since the samples are too few to have statistical significance. Therefore in each of the six classes, 3392, 516, 463, 818, 1003 and 229 ORFs are contained, respectively.

2.2 Computer Software

2.2.1 The characteristic sequences and their numerical characterization

Mathematically, a homomorphism in algebra represents and emphasizes a partial mirror of an algebraic system. With this idea in the mind, we introduce the concept of characteristic sequences of a DNA sequence as follows.

According to their chemical structures, there are two ways to divide the four bases A, C, G, T into two classes: purine $R = \{A, G\}$ and pyrimidine $Y = \{C, T\}$; amino group $M = \{A, C\}$ and keto group $K = \{G, T\}$. Besides these, the division can also be made according to the strength of the hydrogen bond, *i.e.*, weak H–bonds $W = \{A, T\}$ and strong H–bonds $S = \{C, G\}$.

By the three divisions we reduce a DNA sequence into three (0,1) sequences, which is stated in mathematical terms as follows. Given a DNA sequence $G = a_1 a_2 a_3 \cdots$, we define three homomorphic maps ϕ_i, $i = 1, 2, 3$ by $\phi_i(G) = \phi_i(a_1)\phi_i(a_2)\cdots\cdots$, where

$$\phi_1 = \begin{cases} 1 & if \quad a_i \in R \\ 0 & if \quad a_i \in Y \end{cases} \quad \phi_2 = \begin{cases} 1 & if \quad a_i \in M \\ 0 & if \quad a_i \in K \end{cases} \text{ and } \phi_3 = \begin{cases} 1 & if \quad a_i \in W \\ 0 & if \quad a_i \in S \end{cases}$$

The $\phi_i(G)$, $i = 1, 2$ and 3, are called (R, Y)–, (M, K)–, and (W, S)–characteristic sequences, respectively.

Given a (0,1)–sequence $S = a_1 a_2 a_3 \cdots$, we define its normalized height function $h_S(p)$ (or $h(p)$ for short) to be q/p, which denotes the frequency of 1's occurring in the prefix of length p of S, that is, q is the number of 1's in $a_1 a_2 \cdots a_p$. Let k be a fixed positive integer. If S has length n, then we can divide it into k segments and consider their normalized height functions $h([n/k])$, $h([2n/k])$, … , $h([n])$, where $[n/k]$ denotes the biggest integer less than or equal to n/k.

For a DNA sequence, we can construct its characteristic sequences according to the above three homomorphic maps. Then we can obtain $h_R([in/k])$, $h_M([in/k])$ and $h_W([in/k])$, $i = 1, 2, \cdots, k$, where R, M and W denote (R, Y)–, (M, K)– and (W, S)–characteristic sequences, respectively. Thus, we have $3k$ values (or a $3k$–dimensional real vector) to describe a DNA sequence. By comparing these values, we can obtain some information of the DNA sequence.

2.2.2 The gene–finding algorithm

In this section, we suggest a gene–finding algorithm based on the different statistical properties at the three codon positions between protein coding ORFs and non–coding ones. The subsequence in an ORF with bases at positions $3i + 1$ $(i = 0, 1, 2, \cdots)$ forms a phase–specific sequence, and we call it the 1–subsequence. Similarly, we can also define 2–, 3–subsequence with bases at positions

$3i + j$ $(i = 0, 1, 2, \cdots)$ and $j = 2$ or 3 in the ORF.

For each phase–specific subsequence, regarded as an ordinary DNA sequence, there are three characteristic sequences. For each of them, taking $k = 2$ and considering its normalized height function, we obtain a 6–dimensional real vector for the phase–specific subsequence. We denote the six components of the *i*–subsequence by R_{ni}^1, R_{ni}^2, M_{ni}^1, M_{ni}^2, W_{ni}^1, W_{ni}^2, $i = 1, 2, 3$. Making a union of the three 6–dimensional vectors, we can describe each ORF (or an intergenic DNA sequence) by a point in an 18–dimensional real space.

To complete the algorithm in a computer, we need two groups of samples. Let P denote the group of the positive samples consisting of true protein coding genes, and N the group of negative samples composed of non–coding DNA sequences. The two groups of samples form the training set used in the protein coding gene–finding algorithm. Let n approximate the number of samples in each group. In the positive samples the *k*–th true coding ORF is described by a vector $\left(u_{k1}^{\mathrm{P}}, u_{k2}^{\mathrm{P}}, \cdots, u_{k18}^{\mathrm{P}}\right)^T$, where u_{ki}^{P} 's are the i–component of the vector $(i = 1, 2, \cdots, 18)$, and "T" denotes the ordinary transpose operator of matrix. Similarly, a vector $\left(u_{k1}^{\mathrm{N}}, u_{k2}^{\mathrm{N}}, \cdots, u_{k18}^{\mathrm{N}}\right)^T$ describes the *k*–th non–coding DNA sequence in the negative samples.

We adopt the convention used by Zhang *et al*. [1]. By $\bar{U}^{\mathrm{P}}$ and $\bar{U}^{\mathrm{N}}$ we denote the geometric centers of the positive and negative samples in the 18–dimensional space, where

$$\bar{U}^{\mathrm{P}} = \left(\bar{u}_1^{\mathrm{P}}, \bar{u}_2^{\mathrm{P}}, \cdots, \bar{u}_{18}^{\mathrm{P}}\right)^T, \quad \bar{U}^{\mathrm{N}} = \left(\bar{u}_1^{\mathrm{N}}, \bar{u}_2^{\mathrm{N}}, \cdots, \bar{u}_{18}^{\mathrm{N}}\right)^T \tag{1}$$

$$\text{and } \bar{u}_k^{\mathrm{P}} = \frac{1}{n}\sum_{i=1}^{n} u_{ik}^{\mathrm{P}}, \quad \bar{u}_k^{\mathrm{N}} = \frac{1}{n}\sum_{i=1}^{n} u_{ik}^{\mathrm{N}}, \quad k = 1, 2, \cdots, 18. \tag{2}$$

By an 18–dimensional vector $\left(u_1, u_2, \cdots, u_{18}\right)^T$ we denote a query ORF. We calculate the Euclid distances $d\left(U, \bar{U}^{\mathrm{P}}\right)$ between U and $\bar{U}^{\mathrm{P}}$, and $d\left(U, \bar{U}^{\mathrm{N}}\right)$ between U and $\bar{U}^{\mathrm{N}}$ to judge whether or not this ORF is a true protein coding gene. Here

$$d\left(U, \bar{U}^{\mathrm{P}}\right) = \left[\sum_{k=1}^{18}\left(u_k - \bar{u}_k^{\mathrm{P}}\right)^2\right]^{1/2} \text{ and } d\left(U, \bar{U}^{\mathrm{N}}\right) = \left[\sum_{k=1}^{18}\left(u_k - \bar{u}_k^{\mathrm{N}}\right)^2\right]^{1/2} \tag{3}$$

A coding index Δ is defined as $\Delta = d\left(U, \bar{U}^{\mathrm{P}}\right) - d\left(U, \bar{U}^{\mathrm{N}}\right) + c$ (4), where c is a constant determined by making the false positive rate and false negative rate identical in the training set. If $\Delta > 0$, the query ORF is recognized as a true protein coding gene, otherwise, the ORF or DNA sequence is recognized as a non–coding sequence.

3 EVALUATION AND APPLICATION

3.1 Definitions of sensitivity, specificity and accuracy

Sensitivity and specificity measures are widely used to characterize the accuracy of an algorithm or a recognition function. Here, we adopt the definitions and notations in Burset and Guigo [10].

Let TP denote the number of coding ORFs that have been correctly predicted as coding, and FN the number of coding ORFs that have been predicted as non–coding. Then we define the sensitivity S_n as,

$$S_n = \frac{TP}{TP + FN} \tag{5}$$

That is, S_n is the proportion of coding ORFs that have been correctly predicted as coding ORFs. Similarly, denoted by TN the number of intergenic sequences that have been correctly predicted as non–coding, and denoted by FP the number of intergenic sequences that have been predicted as coding, we define the specificity S_p as,

$$S_p = \frac{TN}{TN + FP} \tag{6}$$

That is, S_p is the proportion of intergenic sequences that have been correctly predicted as non–coding. In addition to, we define the accuracy T as the average of the sensitivity and specificity, that is

$$T = \frac{1}{2}\left(S_n + S_p\right) \tag{7}$$

3.2 Self–consistency and cross–validation tests

Usually, the re–substitution and cross–validation tests are efficient methods to evaluate the algorithm. The former reflects the self–consistency, and the latter reflects the extrapolating effectiveness of the algorithm. In the references [1, 2], the authors used the first class in the MIPS database, and regarded them as the positive samples. From the 16 yeast chromosomes, they randomly selected about 6000 intergenic sequences with length longer than 300 bp, starting with ATG and ending with one of the stop codons, and then, from the 6000 intergenic sequences, they randomly selected 2958 sequences as the negative samples and randomly divided each sample into two samples: training set and test set. Using them, their algorithms were evaluated.

Following Zhang's methodology, in this paper, we still use the MIPS database to evaluate our algorithm. The first class includes 3392 known genes in the 16 yeast chromosomes in the MIPS database. There are some differences between our data and that in Zhang's [1] paper. Data used in treatment was of more recent origin than that used in the Zhang’s work.

In the MIPS database released in 2001, the first class included 3392 known genes. We randomly divide the 3392 genes into two parts, one of which includes 2000 genes and the other 1392 genes. The former is regarded as a training set and the latter is regarded as a test set. Using Zhang's [1] method, we randomly select 7691 intergenic sequences (non–coding sequence) from *S. cerevisiae* genome, and randomly select 2000 and 1392 sequences from the above 7691 sequences, which form the training and test sets of the negative samples, respectively. In summary, the training set includes 2000 positive samples (true genes) and 2000 negative samples (intergenic sequences), and the test set include 1392 positive samples (true genes) and 1392 negative samples (intergenic sequences).

Table 1 The accuracy of the algorithm for three different tests.

	Test1	Test2	Test3	Test4	Test5	Test6
Sensitivity(%)	95.9	94.6	96.6	95.9	95.7	94.4
Specificity(%)	94.8	95.8	94.3	95.0	95.5	96.4
Accuracy(%)	95.35	95.2	95.45	95.45	95.6	95.4

Table 2. The 126 ORFs of the 2nd class (no similarity)in the MIPS database, which are recognized as non–coding

yal037c–a	ydr029w	yfr042w	yjl028w	ylr265c	ynl150w
yal064w	ydr042c	ygl006w–a	yjl064w	ylr366w	ynl174w
yar030c	ydr065w	ygl138c	yjl077c	ylr381w	ynl179c
yar047c	ydr102c	ygl188c	yjl136w–a	ylr400w	ynl211c
yar053w	ydr179w–a	ygr026w	yjl215c	ylr404w	ynl303w
yar070c	ydr274c	ygr168c	yjr023c	yml084w	ynl303w
ybl048w	ydr278c	ygr226c	yjr157w	yml090w	yol159c
ybl071c	ydr344c	ygr290w	ykl044w	ymr003w	yol160w
ybr027c	ydr350c	ygr291c	ykl158w	ymr057c	yor024w
ybr056w–a	ydr396w	yhl005c	ykl162c	ymr082c	yor029w
ybr209w	ydr524w–a	yhl037c	ykr032w	ymr141c	yor097c
ybr292c	ydr535c	yhr078w	ykr073c	ymr148w	yor152c
ycl056c	yel010w	yhr095w	yll007c	ymr151w	yor248w
ycl058c	yel014c	yhr139c–a	yll030c	ymr163c	yor255w
ycr022c	yel059w	yhr173c	yll059c	ymr187c	yor364w
ycr025c	yer066c–a	yil012w	ylr111w	ymr252c	yor392w
ycr085w	yer091c–a	yil027c	ylr112w	ymr254c	ypl041c
ydl176w	yer135c	yil071c	ylr122c	ymr320w	ypl200w
ydl196w	yer172c–a	yir020c	ylr124w	ynl122c	ypr012w
ydr015c	yfl019c	yir020c–b	ylr145w	ynl143c	ypr153w
ydr024w	yfl021c–a	yjl027c	ylr264c–a	ynl146w	ypr170w–a

Using the sequences in the training set, the average vectors $\bar{U}^{\mathrm{P}}, \bar{U}^{\mathrm{N}}$ and the parameter c (see Eqs. (2) and (4)) are determined. Using these quantities, the accuracy of the gene–finding algorithm in the training and test sets is calculated. Repeating the above random division procedure six times, we perform six re–substitution and cross–validation tests. The results of the cross–validation tests are listed in Table 1. As we will see from Table 1, the accuracy in each cross–validation test is always greater than 95%.

Table 3. The 297 ORFs of the 3rd class (questionable ORFs) in the MIPS database, which are recognized as non–coding

yal026c–a	ydr149c	ygl088w	yil060w	ylr279w	ynr025c
yal031w–a	ydr154c	ygl109w	yil066w–a	ylr282c	yol013w–b
yal059c–a	ydr157w	ygl118c	yil068w–a	ylr294c	yol035c
ybl053w	ydr199w	ygl132w	yil071w–a	ylr302c	yol099c
ybl062w	ydr203w	ygl149w	yil100c–a	ylr317w	yol134c
ybl065w	ydr220c	ygl152c	yil163c	ylr322w	yol150c
ybl070c	ydr230w	ygl165c	yir017w–a	ylr334c	yor041c
ybl073w	ydr241w	ygl168w	yir023c–a	ylr358c	yor082c
ybl077w	ydr269c	ygl177w	yjl009w	ylr428c	yor102w
ybl094c	ydr271c	ygl182c	yjl015c	ylr434c	yor121c
ybl107w–a	ydr290w	ygl193c	yjl022w	ylr444c	yor146w
ybr051w	ydr355c	ygl204c	yjl032w	ylr458w	yor169c
ybr064w	ydr360w	ygl214w	yjl075c	ylr465c	yor170w
ybr089w	ydr401w	ygl217c	yjl086c	yml009c–a	yor199w
ybr090c	ydr417c	ygl218w	yjl120w	yml012c–a	yor200w
ybr109w–a	ydr426c	ygr011w	yjl135w	yml047w–a	yor225w
ybr116c	ydr431w	ygr018c	yjl142c	yml094c–a	yor235w
ybr178w	ydr445c	ygr039w	yjl150w	yml116w–a	yor263c
ybr206w	ydr467c	ygr050c	yjl175w	ymr046w–a	yor277c
ybr224w	ydr509w	ygr051c	yjl182c	ymr052c–a	yor282w
ybr226c	ydr521w	ygr069w	yjl202c	ymr075c–a	yor300w
ybr266c	ydr526c	ygr073c	yjr018w	ymr086c–a	yor309c
ybr277c	yel009c–a	ygr107w	yjr038c	ymr135w–a	yor325w
ycl041c	yel018c–a	ygr114c	yjr071w	ymr153c–a	yor331c
ycr018c–a	yel075w–a	ygr115c	yjr087w	ymr158c–a	yor345c
ycr041w	yer046w–a	ygr122c–a	ykl030w	ymr158w–b	yor379c
ycr064c	yer067c–a	ygr139w	ykl036c	ymr172c–a	ypl034w
ycr087w	yer076w–a	ygr151c	ykl053w	ymr193c–a	ypl035c
ydl009c	yer084w	ygr176w	ykl076c	ymr290w–a	ypl044c
ydl016c	yer084w–a	ygr182c	ykl083w	ymr304c–a	ypl073c
ydl026w	yer087c–a	ygr219w	ykl115c	ymr306c–a	ypl102c
ydl032w	yer133w–a	ygr228w	ykl118w	ymr316c–a	ypl114w
ydl050c	yer137w–a	ygr259c	ykl131w	ynl013c	ypl185w
ydl062w	yer138w–a	ygr265w	ykl136w	ynl028w	ypl205c
ydl068w	yer145c–a	yhl002c–a	ykl147c	ynl089c	ypl238c
ydl094c	yer148w–a	yhl006w–a	ykl202w	ynl105w	ypl261c
ydl151c	yer165c–a	yhl019w–a	ykr033c	ynl114c	ypr039w
ydl152w	yer181c	yhl030w–a	ykr047w	ynl120c	ypr050c
ydl158c	yfl012w–a	yhl046w–a	yll020c	ynl170w	ypr053c
ydl172c	yfl013w–a	yhr028w–a	ylr101c	ynl171c	ypr077c
ydl187c	yfl015w–a	yhr049c–a	ylr123c	ynl184c	ypr087w
ydl221w	yfl032w	yhr063w–a	ylr140w	ynl198c	ypr099c
ydr008c	yfr036w–a	yhr071c–a	ylr169w	ynl205c	ypr136c
ydr034c–a	yfr052c–a	yhr125w	ylr171w	ynl226w	ypr142c
ydr048c	yfr056c	yhr145c	ylr198c	ynl228w	ypr146c
ydr053w	ygl024w	yhr193c–a	ylr202c	ynl235c	ypr150w
ydr112w	ygl042c	yil020c–a	ylr230w	ynl266w	ypr177c
ydr114c	ygl052w	yil029w–a	ylr252w	ynl276c	
ydr133c	ygl072c	yil030w–a	ylr261c	ynl319w	
ydr136c	ygl074c	yil047c–a	ylr269c	ynr005c	

3.3 Application of the algorithm to find genes in the ORFs of the 2nd–6th classes

In this section, we recognize genes in the ORFs of the 2nd–6th classes in the MIPS database using the algorithm.

Firstly, we merge the training set and test set of the positive samples into a new training positive set, and randomly select 3392 sequences from the 7691 intergenic sequences as mentioned above to form a new training negative set. In order to counter the particularity of the selected samples, we repeat this process ten times, and every time we calculate the average vectors $\bar{U}_i^{\mathrm{P}}$, $\bar{U}_i^{\mathrm{N}}$ and the parameter c_i, so we obtain ten triples $\left(\bar{U}_i^{\mathrm{P}}, \bar{U}_i^{\mathrm{N}}, c_i\right)\ i = 1, 2, \cdots, 10$.

Secondly, by taking the average of the ten triples we obtain a new triple as follows:

$$\bar{U}^{\mathrm{P}} = (0.62111, 0.62825, 0.547480.54638, 0.49741, 049147, 0.48988, 0.49839, 0.62634, 0.63190, 0.57953, 0.57735, 0.47751, 0.47784, 0.60762, 0.60980, 0.48249, 0.48755) \quad (8)$$

$$\bar{U}^{\mathrm{N}} = (0.50238, 049925, 0.64094, 0.64316, 0.50307, 0.49982, 0.50059, 0.50398, 0.64064, 0.64235, 0.49962, 0.50252, 0.50898, 0.50913, 0.63127, 0.63606, 0.49709, 0.50002) \quad (9)$$

$$\text{and } c = 0.015360 \quad (10)$$

Thirdly, we judge each sequence in the ORFs of the 2nd–6th classes in the MIPS database based on the vectors $\bar{U}^{\mathrm{P}}, \bar{U}^{\mathrm{N}}$ and the parameter c listed in Eqs. (8)–(10), respectively. For each ORF, we calculate the vector $U = (u_1, u_2, \cdots, u_{18})^T$, where u_i are defined in section 2.2.2. Based on the vectors $U, \bar{U}^{\mathrm{P}}, \bar{U}^{\mathrm{N}}$ and the parameter c, we calculate each coding–ness index Δ using Eq. (7). If $\Delta > 0$, the query ORF is recognized as a coding gene, otherwise, non–coding. In each class, the ORFs recognized as non–coding ORFs are listed in Tables 2–6 corresponding to the 2nd–6th classes in the yeast genome, respectively.

Table 4. The 60 ORFs of the 4th class (similarity or weak similarity to known proteins)in the MIPS database, which are recognized as non–coding

yal066w	ydr205w	yfr057w	yil040w	ylr064w	ynl176c
ybl089w	ydr249c	ygl104c	yil088c	ylr184w	ynr059w
ybr293w	ydr307w	ygl160w	yjl091c	ylr283w	yol079w
ycr001w	ydr319c	ygr101w	yjl170c	ylr311c	yol107w
ydl073w	ydr366c	ygr284c	yjl193w	ylr365w	yol152w
ydl119c	ydr413c	yhl035c	ykr030w	yml023c	yol163w
ydl199c	ydr524c	yhr035w	ykr103w	ymr088c	yor053w
ydl206w	yel045c	yhr130c	yll005c	ymr245w	yor080w
ydr100w	yer097w	yhr181w	yll037w	ymr306w	yor286w
ydr115w	yfl040w	yil025c	ylr050c	ynl109w	yor350c

Table 5. The 140 ORFs of the 5th class (similarity to unknown proteins)in the MIPS database, which are recognized as non–coding

yal018c	ydl054c	yel033w	yhr067w	ykl225w	ynr062c
yar029w	ydl089w	yel053w–	yhr069c–a	ykr051w	yol002c
yar060c	ydl114w–a	ayel067c	yhr212c	ykr106w	yol003c
yar068w	ydl123w	yer074w–a	yhr214w–a	yll065w	yol047c
ybl029c	ydl159w–a	yer079c–a	yil029c	ylr036c	yol048c
–a ybl049w	ydl185c–a	yer140w	yil089w	ylr047c	yol101c
ybl108w	ydl240c–a	yfl015c	yil090w	ylr149c–a	yol159c–a
ybl109w	ydl247w–a	yfl062w	yil174w	ylr368w	yol162w
ybr004c	ydl248w	yfl068w	yil175w	ylr408c	yor044w
ybr096w	ydr018c	yfr012w	yir030w–a	ylr463c	yor147w
ybr099c	ydr066c	ygl010w	yir040c	yml007c–a	yor175c
ybr103c–a	ydr084c	ygl041c	yjl003w	yml047c	yor365c
ybr147w	ydr105c	ygl084c	yjl052c–a	yml132w	ypl162c
ybr168w	ydr126w	ygl260w	yjl097w	ymr010w	ypl165c
ybr191w–a	ydr131c	ygl263w	yjr013w	ymr013w–a	ypl246c
ybr300c	ydr210w	ygr004w	yjr044c	ymr071c	ypl264c
ybr302c	ydr275w	ygr016w	yjr054w	ymr119w	ypr016w–a
ycl002c	ydr367w	ygr149w	yjr161c	ymr326c	ypr071w
ycl005w	ydr437w	ygr295c	yjr162c	ynl008c	ypr074w–a
ycl065w	ydr438w	yhl034w–a	ykl018c–a	ynl067w–a	ypr114w
ycr038w–a	ydr459c	yhl041w	ykl106c–a	ynl162w–a	
ycr097w–a	ydr492w	yhl042w	ykl165c–a	ynl326c	
ycr102w–a	ydr504c	yhl044w	ykl219w	ynl336w	
ydl027c	ydr525w–a	yhl045w	ykl223w	ynr061c	

Table 6. The 5 ORFs of the 6th class (strong similarity to known proteins)in the MIPS database, which are recognized as non–coding.

ybr210w	yel004w	yll051c	ylr046c	ymr040w

Furthermore, we re–estimate the number of protein coding genes in the 16 yeast chromosomes based on the above results. For example, the total number of the 2nd class ORFs is 516, in which 126 are recognized as non–coding. Suppose both the sensitivity and specificity of our algorithm are 95%, we can obtain a system of four linear equations as follows:

$$\begin{cases} TP/(TP+FN)=0.95 \\ TN/(TN+FP)=0.95 \\ TN+FN=126 \\ TP+FN+TN+FP=516 \end{cases}$$

from which we obtain that $FP \approx 6, FN \approx 20, TP \approx 384, TN \approx 106$. The number of the real coding sequences of the 2nd class should be equal to $TP+FN=384+20=404$. For the 3rd–6th classes, we can treat them in the same way. For the 6th–class, however, the above system has negative solutions. The reason is that the number recognized as non–coding sequences is too small, which is only 5. In this case, taking $FP=FN=0$, we have $TP=224$ and $TN=5$. Then, we list the values of TP, FP, TN and FN in the 2nd–6th class ORFs in Table 7.

Table 7. The numbers of predicted coding and non–coding ORFs of the 2nd–6th classes

Class	2nd	3rd	4th	5th	6th	Total
Total number of ORFs	516	463	818	1003	229	3029
TP	384	151	757	858	224	2374
FN	20	8	40	45	0	113
TN	106	289	20	95	5	515
FP	6	15	1	5	0	27
Total number of coding ORFs	404	159	797	903	224	2487
Total number of noncoding ORFs	112	304	21	100	5	542
Percentage of noncoding ORFs	21.7%	65.7%	2.6%	10%	2.2%	17.9%

Thus, the total number of protein coding genes should be equal to 5897, the sum of the number of the 1st class (3410) and the number of those in the 2nd–6th classes recognized by the present algorithm (3410+404+159+797+903+224, see Table 7). Note that the accuracy is actually greater than 95%, so, this sum should be an upper bound of the number of the genes in the yeast genome. The above estimate of protein coding genes in the yeast genome is coincident with 5800–6000, which is widely accepted [9,11,12]. The above estimate is based on error analysis, i.e. we have considered the false positive and false negative events in the prediction for each class. So, it should be statistically reliable.

4 CONCLUSIONS

In this paper, we propose a method for distinguish coding ORFs and non–coding ORFs in the yeast genome. For complete the algorithm, we take the first class ORFs (known protein) as coding gene sequences and intergenic DNA sequence as non–coding sequences. Using them, we distinguish coding ORFs and non–coding ORFs for 2nd–6th classes ORFs in the yeast genome and obtain the number of coding ORFs in the 2^{nd}–6^{th} classes are at most 404,159, 797, 903 and 224, respectively. As a result, the total number of coding ORFs is estimated to be less than to 5897 in the 16 yeast chromosomes. Besides, we can also observe that the percentage of non–coding ORFs is 17.9% in 2^{nd}–6^{th} classes from Table 7. However, the percentages in the 2nd and 3rd classes are higher than others, 21.7% and 65.7%, respectively. According to classification of ORFs in the MIPS database, some of these ORFs neither their function nor homology are known. Therefore, their high percentage is no wonder. With the increase in known genes, the number and percentage should be decrease.

As we mentioned, the idea of characteristic sequences comes from algebra, which is a kind of reduced representation for a complicated objects. This idea is applied not only to DNA sequences, but also to protein sequences and others. In practice, we can also concentrate on a single characteristic sequence. For example, in gene–finding algorithm of this paper, we can replace the

18–dimensional real space by a 6–dimensional real space: R_{ni}^1, R_{ni}^2, $i = 1, 2, 3$, according to the purine–pyrimidine classification. Using the 6–dimensional space, we can perform the same algorithm on the yeast genome to research the biological function of purine–pyrimidine. Similarly, we can also take M_{ni}^1, M_{ni}^2 or W_{ni}^1, W_{ni}^2, $i = 1, 2, 3$, to research the biological functions of amino–keto groups and weak–strong H–bonds. This might provide a possibility to reveal the biological functions of purine–pyrimidine, amino–keto groups and weak–strong H–bonds, respectively.

Acknowledgment

We thank Dr. Ovidiu Ivanciuc for sending us some references. This work is supported in part by the National Natural Science Foundation of China and Shanghai Postdoctoral Science Foundation.

5 REFERENCES

[1] C. T. Zhang, J, Wang, Recognition of protein coding genes in the yeast genomeat better than 95% accuracy based an the Z curve, *Nucleic Acids Res.* **2000**, *28*, 2804–2814.
[2] C. T. Zhang, J, Wang, R. Zhang, Using a Euclid distance discriminant method to find protein coding genes in the yeast genome, *Comput. Chem.* **2002,** *26*, 195–206.
[3] R. Guigo, Assembling genes from prodicted exons in linear time with dynamic programming, *J. Comput. Biol.* **1998**, *5*, 681–702.
[4] R. Guigo, DNA composition, codon usage and exon prediction. In genetics databases, *Bishop M. J. (ed.*), **1999**, 54–80. Lodon: Academic Press.
[5] M. A. Roytberg, T. V. Astakhova, M. S. Gelfand, Combinatorial approaches to gene recognition, *Comput. Chem.* **1997**, *21*, 229–235.
[6] Y. Quentin, C. Voiblet, F. Martin, G. Fichant, Protein–coding region discovery in organisms under–represented in databases, *Comput. Chem*. **1999**, *23*, 209–217.
[7] R. Guigo, Computational gene identifcation: an open problem, *Comput. Chem*. **1997**, *21*, 215–222.
[8] P. A. He, J. Wang, Characteristic Sequences for DNA Primary Sequence, *J. Chem. Inf. Comput. Sci.* **2002**, *42*, 1080–1085.
[9] A. Goffeau, B. G. Barrel, H. Bussey, R. W. Davis, B. Dujon, H. Feldmann, F. Galibert, J. D. Hoheisel, C. Jacq, M. Johnston, E. J. Louis, H. W. Mewes, Y. Murakami, P. Philippsen, H. Tettlin, S. G. Oliver, Life with 6000 Genes, *Science* **1996**, *274*, 546.
[10] M. Burset, R. Guigo, Evaluation of gene structure prediction programs, *Genomics* **1996**, *34*, 353–367.
[11] H. W. Mewes, K. Albermann, M. Bahr, D. Frishman, A. Gleissner, J. Hani, K. Heumann, K. Kleine, A. Maierl, S. G. Oliver, F. Pfeiffer, A. Zollner, Overview of the yeast genome, *Nature (Suppl.)* **1997**, *387*, 7–8.
[12] E. A. Winzeler, R. W. Davis, Functional analysis of the yeast genome, *Curr. Opin. Genet. Dev.* **1997**, *7*, 771–776.
[13] M. L. Chiusano, F. Alvarez–Valin, M. D. Giulio, G. D'Onofrio, G. Ammirato, G. Colonna, G. Bernardi, Second codon positions of genes and the secondary structures of proteins. Relationships and implications for the origin of the genetic code, *Gene* **2000**, *261*, 63–69.
[14] J. W. Fickett, Finding genes by computer: the state of the art, *Trends Genet.* **1996**, *12*, 316–320.
[15] M. S. Gelfand, Prediction of function in DNA sequence analysis. *J. Computational Biol.* **1995**, *2*, 87–115.
[16] P. Mackiewicz, M. Kowalczuk, A. Gierlik, M. R. Dudek, S. Cebrat, Origin and properties of non–coding ORFs in the yeast genome, *Nucleic Acids Res.* **1999**, *27*, 3503–3509.
[17] S. V. Buldyrev, A. L. Goldberger, S. Havlin, R. N. Mantegna, M. E. Matsa, C. K. Peng, M. Simons, H. E. Stanley, Long–range correlation properties of coding and noncoding DNA sequences: GenBank analysis, *Phys. Rev. E.* **1995**, *51(5)*, 5084–5091.
[18] A. Salamov, V. Solovyev, Ab initio gene finding in Drosophila genomic DNA, *Genome Research* **2000**, *10*, 516–522.

[19] J. C. W. Shepherd, Method to determine the reading frame of a protein from thepurine/pyrimidine genome sequence and its possible evolutionary justi_cation, *Proc. Natl. Acad. Sci. USA* **1981**, *78*, 1596–1600.
[20] I. Z. Siemion, P. J. Siemion, The informational context of the third base in amino acid codons, *BioSystems* **1994**, *33*, 39–48.
[21] V. V. Solovyev, Fractal graphical representation and analysis of DNA and protein sequences, *BioSystems* **1993**, *30*, 137–160.
[22] A. Thomas, M. Skolnick, A probabilistic model for detecting coding regions in DNA sequences. *IMAJ. Math. Appl. Med. Biol* **1994**, *11*, 149–160.
[23] M. Q. Zhang, Identi_cation of protein coding regions in the human genome by quadratic discriminant analysis. *Proc. Natl. Acad. Sci. USA* **1997**, *94*, 565–568.

Biographies

Ping–an He is associate professor at the Faculty of Science, Zhejiang Institute of Science and Technology. His main research interests include combinatorics, graph theory and bioinformatics.

Chun Li is a PhD student of Applied Mathematics at the Dalian University of Technology. His main research interests include combinatorics, information theory and bioinformatics.

Jun Wang is a Professor of Applied Mathematics at the Dalian University of Technology, the advisor of the first two authors.