*1780–1789* *Nucleic Acids Research, 2003, Vol. 31, No. 6*
*DOI: 10.1093/nar/gkg254*

# ZCURVE: a new system for recognizing protein-coding genes in bacterial and archaeal genomes

**Feng-Biao Guo, Hong-Yu Ou and Chun-Ting Zhang***

Department of Physics, Tianjin University, Tianjin 300072, China

Received October 11, 2002; Revised and Accepted January 20, 2003

## ABSTRACT

**A new system, ZCURVE 1.0, for finding protein-coding genes in bacterial and archaeal genomes has been proposed. The current algorithm, which is based on the Z curve representation of the DNA sequences, lays stress on the global statistical features of protein-coding genes by taking the frequencies of bases at three codon positions into account. In ZCURVE 1.0, since only 33 parameters are used to characterize the coding sequences, it gives better consideration to both typical and atypical cases, whereas in Markov-model-based methods, e.g. Glimmer 2.02, thousands of parameters are trained, which may result in less adaptability. To compare the performance of the new system with that of Glimmer 2.02, both systems were run, respectively, for 18 genomes not annotated by the Glimmer system. Comparisons were also performed for predicting some function-known genes by both systems. Consequently, the average accuracy of both systems is well matched; however, ZCURVE 1.0 has more accurate gene start prediction, lower additional prediction rate and higher accuracy for the prediction of horizontally transferred genes. It is shown that the joint applications of both systems greatly improve gene-finding results. For a typical genome, e.g. *Escherichia coli*, the system ZCURVE 1.0 takes ~2 min on a Pentium III 866 PC without any human intervention. The system ZCURVE 1.0 is freely available at: http://tubic.tju.edu.cn/Zcurve_B/.**

## INTRODUCTION

By June 2002, the whole genomic sequences of more than 60 bacteria and archaea were available in the GenBank/EMBL/DDBJ databases. More and more bacterial genome-sequencing projects are currently underway. The fast increasing pace of the bacterial genome-sequencing projects leads to a need for automatic genome annotation. One of the most important tasks of annotation is to recognize protein-coding genes in genomes. Gene recognition is a necessary step to fully understand the functions, activities and roles of genes in cellular processes. Although the gene-finding issue for bacterial and archaeal genomes is relatively easier than that for eukaryotic species, problems have not yet been completely solved. There exist some well-known algorithms and systems for gene-finding in bacterial and archaeal genomes currently, such as GeneMark (1,2), Glimmer (3,4), ORPHEUS (5) and GeneHacker Plus (6). Most of the above algorithms were either based on the higher-order Markov chain models or the hidden Markov chain model. For example, GeneMark used a fifth-order model (1,2), whereas Glimmer used a *k*-order model, where $0 \leqslant k \leqslant 8$ (3,4). Higher-order Markov chain models are particularly effective in extracting local statistical characteristics of coding sequences. Consequently, high recognition accuracy is generally achieved. However, the main disadvantage of such Markov chain models is that thousands of parameters are needed in practical use. For example, for a fifth-order Markov model, a total of $3 \times 4^6 = 12\,288$ parameters are needed. When the size of the genome is not large enough, the gene recognition by such models may be less reliable.

An attempt is made in this paper to put forward an alternative approach for gene recognition in bacterial and archaeal genomes to overcome the above shortcoming. Although the correlation of dinucleotides is considered here, the algorithm is mainly sustained by considering the global statistical characteristics of coding sequences. The methodology adopted here is based on the Z curve representation of DNA sequences (7). The method has been used to recognize genes in budding yeast (8) and *Vibrio cholerae* genomes (9). Although the method is phrased in the language of Z curve, essentially it is based on the compositional asymmetry of three codon positions in coding sequences. The idea was pioneered in the work of Fickett (10) and Staden (11) about 20 years ago in the pre-genome era. The algorithm presented here is an improved version of our previous work in two respects. First, in previous studies known genes were used to predict unknown ones (8,9), whereas the new algorithm is an *ab initio* gene-finding system, i.e. the un-annotated genomic sequences are the only input data. Secondly, in addition to considering the occurrence frequencies of single nucleotides, those of dinucleotides are taken into account. Compared with the Markov chain models, the algorithm presented is much simpler because only 33 parameters are needed. Therefore, this algorithm is basically different from those used in GeneMark (1,2), Glimmer (3,4) and GeneHacker Plus (6). Generally speaking, the former and latter lay stress on global and local statistical features of coding sequences, respectively. Thus, the two approaches are essentially complementary. It is

*To whom correspondence should be addressed. Tel: +86 22 2740 2987; Fax: +86 22 2740 2697; Email: ctzhang@tju.edu.cn

 

shown that the joint utilizations of both approaches lead to better gene-finding results in bacterial and archaeal genomes.

## MATERIALS AND METHODS

### The database

The bacterial and archaeal genomes and related annotation information were downloaded from the GenBank Release 129.0.

### Seeking all ORFs and the 'seed' ORFs from bacterial or archaeal genomes

The gene-finding method consists of a number of steps. The first step is to seek all ORFs from the genome being studied. An ORF is defined as a fragment of DNA sequence beginning with one of the codons ATG, CTG, GTG or TTG and ending with one of the three stop codons. In this paper, the default minimum length of ORFs studied is 90 bp. All the possible ORFs equal to or longer than 90 bp in each of the six frames of the double-strand DNA are extracted. For genomes with the G+C content <56%, another set of ORFs with length longer than 500 bp, named seed ORFs, are also extracted, which do not overlap with any other ORFs. These ORFs are very likely to be protein-coding genes (3,4). For the genomes with G+C content >56%, the method to find seed ORFs will be discussed in another section below.

### The core algorithm

The methodology adopted here is based on the Z curve (7), which is another representation of DNA sequences. Here the algorithm is presented briefly as follows. The frequencies of bases A, C, G and T occurring in an ORF or a fragment of DNA sequence with bases at positions 1, 4, 7, …; 2, 5, 8, …; 3, 6, 9, ..., are denoted by $a_1$, $c_1$, $g_1$, $t_1$; $a_2$, $c_2$, $g_2$, $t_2$; $a_3$, $c_3$, $g_3$, $t_3$, respectively. They are in fact the frequencies of bases at the first, second and third codon positions. Based on the Z curve (7), $a_i$, $c_i$, $g_i$, $t_i$ are mapped onto a point $P_i$ in a three-dimensional space $V_i$, $i$ = 1, 2, 3. The coordinates of $P_i$, denoted by $x_i$, $y_i$, $z_i$, are determined by the Z-transform of DNA sequence (7):

$$\begin{cases} x_i = (a_i + g_i) - (c_i + t_i) \\ y_i = (a_i + c_i) - (g_i + t_i), \; x_i, y_i, z_i \in [-1, 1], \; i = 1, 2, 3 \\ z_i = (a_i + t_i) - (g_i + c_i). \end{cases} \quad \mathbf{1}$$

The Z-transform of DNA sequence transforms the four frequencies of DNA bases into the coordinates of a point in a three-dimensional space. In addition to the frequencies of codon-position-dependent single nucleotides, we need to consider the frequencies of phase-specific dinucleotides. Let the frequencies of the 16 dinucleotides AA, AC, …, and TT occurring at the codon positions 1–2 and 2–3 of an ORF or a fragment of DNA sequence be denoted by $p_{12}$(AA), $p_{12}$(AC), …, $p_{12}$(TT); $p_{23}$(AA), $p_{23}$(AC), …, and $p_{23}$(TT), respectively. Using the Z-transform (7), we find:

$$\begin{cases} x_k^X = [p_k(\mathrm{XA}) + p_k(\mathrm{XG})] - [p_k(\mathrm{XC}) + p_k(\mathrm{XT})] \\ y_k^X = [p_k(\mathrm{XA}) + p_k(\mathrm{XC})] - [p_k(\mathrm{XG}) + p_k(\mathrm{XT})], \\ z_k^X = [p_k(\mathrm{XA}) + p_k(\mathrm{XT})] - [p_k(\mathrm{XG}) + p_k(\mathrm{XC})] \\ \qquad \mathrm{X} = \mathrm{A}, \mathrm{C}, \mathrm{G}, \mathrm{T}; \; k = 12, 23 \end{cases} \quad \mathbf{2}$$

where $x_k^X$, $y_k^X$ and $z_k^X$ are the coordinates, X = A, C, G, T and $k$ = 12, 23. Let the three-dimensional space $V_k^X$ be spanned by $x_k^X$, $y_k^X$ and $z_k^X$. The direct-sum of the subspaces $V_1$, $V_2$, $V_3$, $V_{12}^A$, $V_{12}^C$, $V_{12}^G$, $V_{12}^T$, $V_{23}^A$, $V_{23}^C$, $V_{23}^G$ and $V_{23}^T$ is denoted by a 33-dimensional space V, i.e. $V = V_1 \oplus V_2 \oplus V_3 \oplus V_{12}^A \oplus \ldots \oplus V_{23}^T$, where the symbol $\oplus$ denotes the direct-sum of two subspaces. The 33 components of the space V, i.e. $u_1$, $u_2$, …, $u_{33}$, are defined as follows:

$$\begin{cases} u_1 = x_1, \; u_2 = y_1, \; u_3 = z_1 \\ u_4 = x_2, \; u_5 = y_2, \; u_6 = z_2 \\ u_7 = x_3, \; u_8 = y_3, \; u_9 = z_3 \end{cases} \quad \mathbf{3}$$

$$\begin{cases} u_{10} = x_{12}^A, \; u_{11} = y_{12}^A, \; u_{12} = z_{12}^A \\ u_{13} = x_{12}^C, \; u_{14} = y_{12}^C, \; u_{15} = z_{12}^C \\ u_{16} = x_{12}^G, \; u_{17} = y_{12}^G, \; u_{18} = z_{12}^G \\ u_{19} = x_{12}^T, \; u_{20} = y_{12}^T, \; u_{21} = z_{12}^T \end{cases} \quad \mathbf{3'}$$

$$\begin{cases} u_{22} = x_{23}^A, \; u_{23} = y_{23}^A, \; u_{24} = z_{23}^A \\ u_{25} = x_{23}^C, \; u_{26} = y_{23}^C, \; u_{27} = z_{23}^C \\ u_{28} = x_{23}^G, \; u_{29} = y_{23}^G, \; u_{30} = z_{23}^G \\ u_{31} = x_{23}^T, \; u_{32} = y_{23}^T, \; u_{33} = z_{23}^T \end{cases} \quad \mathbf{3''}$$

Therefore, an ORF or a fragment of DNA sequence can be represented by a point or a vector in the 33-dimensional space V. Note that $u_i \in [-1, +1]$, $i$ = 1, 2, …, 33. Therefore, the space V is a 33-dimensional super-cube with the side length of 2.

To complete the algorithm, we need two groups of samples. One is a set of the positive samples corresponding to the so-called seed ORFs (regarded as protein-coding genes); the other is a set of control (negative) samples corresponding to the non-coding sequences. The two groups of samples constitute the training set, used in the Fisher discriminant algorithm described below. Before calculating the Fisher coefficients, the strategy to produce the negative samples needs to be mentioned. It is a rather difficult problem to prepare an appropriate set of non-coding sequences in bacterial genomes, because the amount of non-coding DNA sequences is too few to be used. For example, in the genome of *V.cholerae*, <12% (14%) of the whole DNA sequences in the larger (smaller) chromosome is non-coding (9). In some bacterial genomes, the fraction of the non-coding sequences is even less than 10%. Therefore, non-coding sequences in most bacterial or archaeal genomes are too limited to be used as negative samples. Additionally, the quality of non-coding samples so produced is questionable, because the intergenic sequences are generally dominated by structural RNA sequences or other functional elements. To solve the problem, a method to produce negative samples is presented here. A negative sample is just derived from a seed ORF. Generally, a coding DNA sequence is the result of the organism's evolution in a very long history. The selection pressure is so strong that only few DNA sequences are lucky enough to be selected as coding sequences coding for native proteins. Therefore, coding sequences have stringent regular structures (7,12). If the regular structure of a coding sequence is completely destroyed, the coding sequence is transformed into a non-coding one. Therefore, the negative sample corresponding to the positive one (seed ORF) may be simply obtained by shuffling the coding sequence sufficiently. A simple Monte Carlo program was written to shuffle each coding sequence for

at least 20 000 times. The complementary sequence of the resulting random sequence was used as a non-coding sequence. Consequently, the coding and the non-coding sequence so produced have the same length, but with different base composition. The major difference is that the former has some regular structure, while the latter does not.

The Fisher linear equation for discriminating the positive and negative samples in the 33-dimensional space V represents a super-plane, described by a vector **c** which has 33 components $c_1$, $c_2$, …, and $c_{33}$. For more detail, refer to, for example, Zhang and Wang (8) and Mardia *et al.* (13). Based on the data in the training set (including the positive and negative samples), an appropriate threshold $c_0$ is determined to make the coding/non-coding decision. The threshold $c_0$ is uniquely determined by making the false negative rate and the false positive rate equal. Once the vector **c** and the threshold $c_0$ are obtained, the decision of coding/non-coding for each ORF is simply made by the criterion of $\mathbf{c}\cdot\mathbf{u} > c_0$ / $\mathbf{c}\cdot\mathbf{u} < c_0$, where $\mathbf{c} = (c_1, c_2, \ldots, c_{33})^T$, $\mathbf{u} = (u_1, u_2, \ldots, u_{33})^T$, and 'T' indicates the transpose of a matrix.

The criterion of $\mathbf{c}\cdot\mathbf{u} > c_0$ / $\mathbf{c}\cdot\mathbf{u} < c_0$ for making the decision of coding/non-coding can be rewritten as $Z(\mathbf{u}) > 0$ / $Z(\mathbf{u}) < 0$, where $Z(\mathbf{u}) = \mathbf{c}\cdot\mathbf{u} - c_0$. $Z(\mathbf{u})$ is called the Z score or Z index for an ORF or a fragment of DNA sequence.

### Strategy to deal with overlapping ORFs

Once the vector **c** and the threshold $c_0$ are derived from the set of seed ORFs and the corresponding set of negative samples, the next step for finding genes is to apply the Z score to all the ORFs found in the first step of the method. The ORFs with Z scores >0 are kept, and those with Z scores <0 are discarded. For the ORFs obtained by this way, in many cases two ORFs overlap with each other. The two overlapping ORFs may be either situated at the same strand or at different strands. If the fraction of overlapping part of two ORFs is greater than one-fifth of the whole length of any of the two overlapping ORFs, the longer one that has a larger Z score is recognized as a gene, and the shorter that has a smaller Z score is a non-coding one. Otherwise, both are recognized as genes. In a few cases, if the Z score for the shorter ORF is remarkably greater than that of the longer ORF, the shorter ORF is recognized as a gene, and the longer is recognized as non-coding. After resolving the two ORF's overlap, the overlap for three ORFs needs to be resolved, because three ORFs overlap occasionally. If the Z scores of the first and third ORFs are much greater than that of the second, the second ORF is rejected. Otherwise, the first second and the second third are dealt with separately according to the method for resolving two overlapping ORFs. Generally speaking, the procedure to deal with overlapping ORFs in ZCURVE 1.0 is in an iteration mode. The iteration procedure will stop when no new task for resolving overlapping ORFs is needed.

### Method to predict gene starts

The prediction of gene starts in bacterial and archaeal genomes is a difficult problem. Since the pioneering work of Stormo *et al.* (14), the issue has been the subject of intensive studies during the past years (15–18). Here the same problem is tackled using the Z curve method. It was observed that (data not shown) the behavior of the Z curve at the vicinity of gene starts is notably different from that of non-gene starts. Based on this fact, a method to predict gene starts is proposed, which includes two steps. The first is a filtering step, in which the seed ORFs are filtered such that the gene starts of the ORFs filtered have the maximum fidelity in some sense. For simplicity, each component of the Z curve (7) at the vicinity of a start codon is approximately fitted by a straight line:

$$\begin{cases} x_n = v_1 \times n + b_x, \; n \in [-13, -7] \\ y_n = v_2 \times n + b_y, \; n \in [-13, -7] \\ z_n = v_3 \times n + b_z, \; n \in [-36, +20] \end{cases} \quad \mathbf{4}$$

where $n$ is the base position, and $v_1$, $v_2$ and $v_3$ are the slopes of the straight lines determined by the least square fitting, while $b_x$, $b_y$ and $b_z$ are the corresponding intercepts. Note that the first nucleotide of a start codon is located at the position zero. Also note that the two parameters $v_1$ and $v_2$ are calculated relying on the same region, whereas $v_3$ is computed in a different region. To calculate the fourth parameter, the flanking fragments of the upstream region –90 ~ –1, and the downstream region 0 ~ 89 are considered. For each of the two flanking fragments, calculate the Z score, and denote them by $Z_{up}$ and $Z_{down}$ for the upstream and downstream fragment, respectively. Note that usually the Z score is computed for an ORF, and now it is computed for a fragment of 90 bp in length. Then $v_4 = -Z_{up} \times Z_{down}$. Here the new sets of positive and negative samples are needed, which are different from those defined in the section 'The core algorithm'. The set of positive samples consists of the start codons of all seed ORFs in a genome. The set of negative samples consists of all false start codons, either upstream or downstream of each start codon. For each start codon, either in the positive or negative sample set, the four parameters $v_1$, $v_2$, $v_3$ and $v_4$ are computed, which correspond to a point in a four-dimensional space. Consequently, there are two kinds of points in the four-dimensional space, corresponding to the positive and negative sample set, respectively. Using the Fisher discriminant algorithm again, a Fisher discriminant function (similar to the Z score defined previously) is calculated for each start codon, including the false ones. At the final stage, only the seed ORF, whose start codon has the maximum Fisher discriminant function value, is retained, and all other seed ORFs are discarded. Accordingly, the retained seed ORFs are called confident seed ORFs, in which the positions of their starts are accurate with the maximum fidelity.

Once the set of confident seed ORFs is established, the training step begins. Similar to the filtering step described above, for each start codon, either in the positive or negative sample set derived from the confident seed ORFs, calculate the four parameters $v_1$, $v_2$, $v_3$ and $v_4$ (in the program, the region for calculating $v_1$ and $v_2$ is slightly adjusted depending on different genomes; to avoid tedious description, the detail is omitted). In addition, another two parameters, $v_5$ and $v_6$, need to be introduced. If the start codon is ATG, then $v_5(\mathrm{ATG}) = 0.78$, otherwise, $v_5(\mathrm{GTG}) = 0.14$, $v_5(\mathrm{TTG}) = 0.07$ and $v_5(\mathrm{others}) = 0.01$. The sixth parameter is defined by $v_6 = e^{-l/l_0}$, where $l_0$ denotes the length of the longest ORF and $l$ the distance between the start codon and the most-left start codon in the confident seed ORF studied. Based on the six parameters and positive and negative sample sets derived from the confident seed ORFs, the Fisher discriminant algorithm is used again. Consequently, six Fisher coefficients and an appropriate threshold are calculated. Then the training step is finished.

**Table 1.** Comparison of the numbers of annotated and additional genes found by ZCURVE 1.0 and Glimmer 2.02, respectively, for 18 complete bacterial or archaeal genomes[a]

| Information in GenBank Organism | GenBank accession no. | G+C content (%) | No. of genes annotated | ZCURVE 1.0 Annotated genes found (%)[b] | ZCURVE 1.0 Additional genes found (%)[c] | Glimmer 2.02 Annotated genes found (%)[b] | Glimmer 2.02 Additional genes found (%)[c] |
|---|---|---|---|---|---|---|---|
| *Buchnera* | BA000003 | 26.31 | 564 | 563 (99.8) | 84 (14.9) | 564 (100) | 85 (15.1) |
| *C.perfringens* | BA000016 | 28.57 | 2660 | 2648 (99.5) | 126 (4.7) | 2639 (99.2) | 160 (6.0) |
| *R.prowazekii* | AJ235269 | 29.00 | 834 | 826 (99.0) | 124 (14.9) | 830 (99.5) | 219 (26.3) |
| *C.acetobutylicum* | AE001437 | 30.93 | 3672 | 3630 (98.9) | 332 (9.0) | 3619 (98.6) | 552 (15.0) |
| *R.conorii* | AE006914 | 32.44 | 1374 | 1331 (96.9) | 221 (16.1) | 1328 (96.7) | 429 (31.2) |
| *L.lactis* | AE005176 | 35.33 | 2266 | 2245 (99.1) | 343 (15.1) | 2226 (98.2) | 387 (17.1) |
| *L.monocytogenes* | AL591824 | 37.98 | 2846 | 2835 (99.6) | 264 (9.3) | 2828 (99.4) | 229 (8.0) |
| *C.pneumoniae* | AE001363 | 40.58 | 1052 | 1035 (98.4) | 138 (13.1) | 1036 (98.5) | 248 (23.6) |
| *C.trachomatis* | AE001273 | 41.31 | 894 | 883 (98.8) | 124 (13.9) | 887 (99.2) | 194 (21.7) |
| *A.aeolicus* | AE000657 | 43.48 | 1522 | 1514 (99.5) | 311 (20.4) | 1511 (99.3) | 261 (17.1) |
| *B.subtilis* | AL009126 | 43.52 | 4100 | 4036 (98.4) | 799 (19.5) | 4026 (98.2) | 1092 (26.6) |
| *T.acidophilum*[d] | AL139299 | 45.99 | 1478 | 1437 (97.2) | 297 (20.1) | 1438 (97.3) | 314 (21.2) |
| *M.thermoautotrophicum*[d] | AE000666 | 49.54 | 1869 | 1825 (97.6) | 295 (15.8) | 1814 (97.1) | 236 (12.6) |
| *E.coli* | U00096 | 50.79 | 4289 | 4210 (98.2) | 1012 (23.6) | 4157 (96.9) | 1005 (23.4) |
| *S.meliloti* | AL591688 | 62.73 | 3341 | 3316 (99.3) | 1239 (37.1) | 3287 (98.4) | 2077 (62.2) |
| *P.aeruginosa* | AE004091 | 66.56 | 5565 | 5466 (98.2) | 1206 (21.7) | 5503 (98.9) | 3144 (56.5) |
| *R.solanacearum* | AL646052 | 67.04 | 3440 | 3349 (97.4) | 1020 (29.7) | 3346 (97.3) | 2058 (59.8) |
| *S.coelicolor* | AL645882 | 72.12 | 7512 | 7269 (96.8) | 1377 (18.3) | 7186 (95.7) | 4427 (58.9) |
| Average (14)[e] | – | – | – | 98.64 ± 0.91 | 15.03 ± 5.04 | 98.44 ± 1.07 | 18.92 ± 7.24 |
| Average (4)[e] | – | – | – | 97.93 ± 1.08 | 26.70 ± 8.42 | 97.58 ± 1.42 | 59.35 ± 2.36 |
| Average (18)[e] | – | – | – | 98.48 ± 0.96 | 17.62 ± 7.54 | 98.24 ± 1.17 | 27.91 ± 18.44 |

[a]The names of the bacteria or archaea are listed in the ascending order of their genomic G+C content. The abbreviated names of bacteria or archaea are used. For example, *Bacillus subtilis* is abbreviated as *B.subtilis*, and so forth. According to our definition, the length of an ORF includes the stop codon, whereas Glimmer does not. Therefore, the minimum ORF length, 90 bp adopted in our program, corresponds to 87 bp in Glimmer. When running Glimmer 2.02, except the minimum ORF length which is assigned to 87 bp, all other parameters are assigned by default.
[b]The percentage in the parenthesis is the accuracy.
[c]The percentage in the parenthesis is the additional prediction rate.
[d]Archaeal genomes.
[e]The values are averaged over the first 14, last four and all the 18 genomes, respectively. The figure following ± is the standard deviation.

Based on the six Fisher coefficients and the threshold obtained, the Fisher discriminant function (similar to the Z score defined previously), is used as a post-processor to relocate the gene starts for gene-finding output. Only the start codon in a gene with the maximum Fisher discriminant function value is predicted to be the true start codon.

## RESULTS AND DISCUSSION

### Indices to evaluate the algorithm

To test the algorithm, the program was run for some bacterial or archaeal genomes available in the GenBank Release 129.0. Evaluation of the algorithm is based on the comparison between the results of gene-finding by the method presented and the annotation in GenBank for each genome. Two independent indices are used to evaluate the performance of the algorithm. The first is called 'accuracy' or 'sensitivity' defined by:

$$\text{Accuracy} = \frac{\text{Number of genes predicted correctly by the algorithm in a genome}}{\text{Number of genes annotated in GenBank for the genome studied}}, \quad \mathbf{5}$$

and the second is called 'additional prediction rate' defined by:

$$\text{Additional prediction rate} = \frac{\text{Number of genes predicted that do not appear in the annotation}}{\text{Number of genes annotated in GenBank for the genome studied}}. \quad \mathbf{6}$$

### Comparison with Glimmer 2.02 (I): all annotated genes and function-known genes

In GenBank Release 129.0, most bacterial or archaeal genomes were annotated completely or partially with Glimmer. For a fair comparison with Glimmer, only the genomes not annotated by Glimmer were selected. Consequently, 18 bacterial or archaeal genomes were used for the comparison, whose names are listed as follows: *Buchnera* sp. APS, *Clostridium perfringens*, *Rickettsia prowazekii* strain Madrid E, *Clostridium acetobutylicum* ATCC824, *Rickettsia conorii* Malish 7, *Lactococcus lactis* subsp. *lactis* IL1403, *Listeria monocytogenes* strain EGD, *Chlamydophila pneumoniae* CWL029, *Chlamydia trachomatis*, *Aquifex aeolicus*, *Bacillus subtilis*, *Thermoplasma acidophilum*, *Methanobacterium thermoautotrophicum* delta H, *Escherichia coli* K12, *Sinorhizobium meliloti* 1021, *Pseudomonas aeruginosa* PA01, *Ralstonia solanacearum* and *Streptomyces coelicolor* A3(2). Their abbreviated names and the GenBank accession numbers are listed in Table 1, where the genomes are listed in the order in which the genomic G+C content is ascending. Accordingly, ZCURVE 1.0 and Glimmer 2.02 were run for

**Table 2.** Comparison of the numbers of function-known genes found by ZCURVE 1.0 and Glimmer 2.02, respectively, for 18 complete bacterial or archaeal genomes[a]

| Organism | No. of function-known genes[a] | Function-known genes found by ZCURVE 1.0 (%) | Function-known genes found by Glimmer 2.02 (%) |
|---|---|---|---|
| *Buchnera* | 477 | 99.8 | 100 |
| *C.perfringens* | 853 | 99.4 | 99.8 |
| *R.prowazekii* | 488 | 99.4 | 100 |
| *C.acetobutylicum* | 2092 | 99.8 | 99.9 |
| *R.conorii* | 545 | 99.6 | 100 |
| *L.lactis* | 1441 | 99.2 | 98.3 |
| *L.monocytogenes* | 244 | 98.8 | 100 |
| *C.pneumoniae* | 593 | 98.5 | 99.2 |
| *C.trachomatis* | 569 | 98.9 | 99.6 |
| *A.aeolicus* | 859 | 99.5 | 99.3 |
| *B.subtilis* | 1224 | 98.5 | 98.6 |
| *T.acidophilum* | 515 | 99.6 | 100 |
| *M.thermoautotrophicum* | 885 | 99.1 | 99.4 |
| *E.coli*[b] | 811 | 98.5 | 99.0 |
| *S.meliloti* | 187 | 98.9 | 97.3 |
| *P.aeruginosa*[c] | 405 | 98.8 | 99.5 |
| *R.solanacearum* | 92 | 100 | 98.9 |
| *S.coelicolor* | 792 | 99.4 | 98.9 |
| Average[d] | – | 99.21 ± 0.48 | 99.32 ± 0.73 |

[a]Function-known genes were obtained from the annotated files in GenBank, except those noted specially below.
[b]For the *E.coli* genome, 811 experimentally verified genes were obtained from the EcoGene database (http://bmb.med.miami.edu/) (20).
[c]405 experimentally verified *P.aeruginosa* genes were downloaded from the PseudoCAP database (http://www.pseudomonas.com) (23).
[d]The values are averaged over all the 18 genomes listed. The figure following ± is the standard deviation.

each of the 18 bacterial or archaeal genomes. Parameters in Glimmer 2.02 were used by default settings. The results are also listed in Table 1. For the 14 bacterial or archaeal genomes with the G+C content <56% in Table 1, the average accuracy of ZCURVE 1.0 and Glimmer 2.02 is 98.64 ± 0.91% and 98.44 ± 1.07%, respectively. The additional prediction rates of ZCURVE 1.0 and Glimmer 2.02 are 15.03 ± 5.04% and 18.92 ± 7.24%, respectively. For the four bacterial genomes with the G+C content >56% in Table 1, the average accuracy of ZCURVE 1.0 and Glimmer 2.02 is 97.93 ± 1.08% and 97.58 ± 1.42%, whereas the average additional prediction rates of ZCURVE 1.0 and Glimmer 2.02 are 26.70 ± 8.42% and 59.35 ± 2.36%, respectively. Obviously, the additional prediction rate of ZCURVE 1.0 is much lower than that of Glimmer. If the average is performed over 18 genomes, the average accuracy is 98.48 ± 0.96% and 98.24 ± 1.17%, whereas the average additional prediction rates are 17.62 ± 7.54% and 27.91 ± 18.44%, respectively, for ZCURVE 1.0 and Glimmer 2.02. In summary, the prediction accuracy of both systems is well matched, but Glimmer 2.02 has much higher additional prediction rate (~10% higher) than ZCURVE 1.0.

Since the annotations are not 100% accurate, further comparisons were performed based on the function-known genes which have more reliable annotations. For each of the 18 genomes, the genes with known biological functions were selected. The gene-finding accuracy of these genes for both systems is listed in Table 2. Note that the accuracy presented here is defined as the ratio of the number of function-known genes predicted correctly by the algorithm in the genome over the total number of function-known genes selected in the genome. Also note that the additional prediction rates are meaningless in this case. As we can see from Table 2, the average accuracy and standard deviation for ZCURVE 1.0 and Glimmer 2.02 are 99.21 ± 0.48% and 99.32 ± 0.73%, respectively. The comparison further supports the conclusion that the gene-finding accuracy of both systems is well matched.

## Comparison with Glimmer 2.02 (II): short and horizontally transferred genes

The cutting edge of the gene-finding issue is better prediction of the difficult cases—short and horizontally transferred genes. The short genes are defined as those with length ⩽300 bp. For short genes, the difficulty of finding them comes from the fact that statistical features are usually not remarkable due to the shortness of sequence length. Horizontally transferred genes usually have unusual base composition, G+C content and codon usage, resulting in problems in gene-finding algorithms. Therefore, it is important to examine the performance of the new system for finding short and horizontally transferred genes. To have the statistical reliability, the genomes with more than 100 annotated short genes are studied here. Of the 18 genomes listed in Table 1, only 12 genomes meet this condition. Their names are listed in Table 3. The accuracy of finding these genes for each genome by ZCURVE 1.0 and Glimmer 2.02, respectively, are listed in the fourth and fifth columns of Table 3. As indicated in the last line of Table 3, the average accuracy with ZCURVE 1.0 is ~4% higher than that with Glimmer 2.02. Since the annotation is not perfect, further comparison based on short and

**Table 3.** Comparison of the numbers of short annotated genes found by ZCURVE 1.0 and Glimmer 2.02, respectively, for 12 complete bacterial or archaeal genomes

| Organism | No. of short genes annotated (≤300 bp) | Short genes found by ZCURVE 1.0 (%) | Short genes found by Glimmer 2.02 (%) |
|---|---|---|---|
| *C.perfringens* | 246 | 95.5 | 92.3 |
| *C.acetobutylicum* | 408 | 90.4 | 87.3 |
| *R.conorii* | 383 | 91.9 | 91.9 |
| *L.lactis* | 252 | 92.1 | 87.7 |
| *L.monocytogenes* | 237 | 95.8 | 94.9 |
| *B.subtilis* | 476 | 88.7 | 88.4 |
| *M.thermoautotrophicum* | 228 | 86.4 | 82.0 |
| *E.coli* | 382 | 84.0 | 77.7 |
| *S.meliloti* | 273 | 96.0 | 89.4 |
| *P.aeruginosa* | 319 | 93.7 | 90.6 |
| *R.solanacearum* | 320 | 85.0 | 80.9 |
| *S.coelicolor* | 565 | 81.8 | 69.9 |
| Average (8)[a] | – | 90.60 ± 4.14 | 87.78 ± 5.65 |
| Average (4)[a] | – | 89.13 ± 6.80 | 82.70 ± 9.56 |
| Average (12)[a] | – | 90.11 ± 4.90 | 86.08 ± 7.18 |

[a]The values are averaged over the first eight, last four and all the 12 genomes, respectively. The figure following ± is the standard deviation.

**Table 4.** Comparison of the numbers of short and function-known genes found by ZCURVE 1.0 and Glimmer 2.02, respectively, for four bacterial genomes

| Organism | No. of short function-known genes (≤300 bp) | Short function-known genes found by ZCURVE 1.0 (%) | Short function-known genes found by Glimmer 2.02 (%) |
|---|---|---|---|
| *C.acetobutylicum* | 93 | 96.8 | 96.8 |
| *L.lactis* | 123 | 91.1 | 86.2 |
| *B.subtilis* | 88 | 83.0 | 81.8 |
| *E.coli* | 111 | 75.7 | 78.4 |
| Average | – | 86.65 ± 9.24 | 85.80 ± 8.00 |

function-known genes was performed. Only four genomes have more than 80 short and function-known genes. The accuracy of finding these genes is listed in Table 4. As can be seen, the average accuracy of ZCURVE 1.0 is still higher than that of Glimmer 2.02. A database of all the short and function-known genes in the 18 genomes listed in Table 1 has been constructed and is accessible from the web site: http://tubic.tju.edu.cn/Zcurve_B/Appendix/.

Recently, a database of horizontally transferred genes in prokaryotic genomes was established (19). The genomes with more than 100 horizontally transferred genes identified are studied here. Of the 18 genomes listed in Table 1, only 10 genomes meet this condition. Their names and gene-finding accuracy are listed in Table 5. Although the horizontally transferred genes in the database are only putative, the data have high reliability (19). As we can see from Table 5, the average accuracy of finding horizontally transferred genes with ZCURVE 1.0 is ~2% higher than that with Glimmer 2.02. Based on the two tests shown above, the performance of ZCURVE 1.0 for finding short and horizontally transferred genes is generally better than that of Glimmer 2.02. The results may be explained by the fact that in ZCURVE 1.0, only 33 parameters are used to characterize the coding sequences, therefore it gives better consideration to both typical and atypical cases, whereas in high-order Markov-model-based methods, e.g. Glimmer 2.02, thousands of parameters are trained, which may result in less adaptability.

## Comparison with Glimmer 2.02 (III): gene start prediction

To evaluate the performance of gene start prediction of the new system, some reliable data sets were used. For *E.coli*, 195 genes were taken from Link *et al.* (18), whose start sites were experimentally validated by protein N-terminal sequencing. In addition, 811 *E.coli* genes in the EcoGene database (20), whose starts were also verified by protein N-terminal sequencing, were used as another test set. For *B.subtilis*, 58 genes were taken from Yada *et al.* (6), whose starts were confirmed with homologous sequences. Consequently, for the 195 *E.coli* genes, all (100%) genes were found by ZCURVE 1.0, of which 180 (180/195 = 92.3%) gene starts were precisely predicted by the new system. For the 811 *E.coli* genes in the EcoGene database, 799 (799/811 = 98.5%) genes were found, of which 719 (719/811 = 88.7%) gene starts were precisely predicted by ZCURVE 1.0. For the 58 *B.subtilis* genes, 57 genes (57/58 = 98.3%) were found, of which 54 (54/58 = 93.1%) gene starts were precisely predicted by the program ZCURVE 1.0. Refer to Table 6 for a summary. Compared with the Glimmer system, it was reported (16) that Glimmer 2.0 correctly predicted 68% of gene starts out of the 195 *E.coli* genes; 66% of gene starts out of the genes in an early version of the EcoGene data set. After post-processing by RBSfinder (16), the accuracy of gene start prediction increased to 92 and 88% for the 195 *E.coli* genes and genes in the EcoGene data set, respectively. When running Glimmer 2.02 for the genome of *B.subtilis*, of the 58 genes validated, 57 (57/58 = 98.3%) were found, of which 40 (40/58 = 69.0%) gene starts were precisely predicted by Glimmer 2.02 without switching to the gene start prediction subroutine. If the subroutine was switched on, the accuracy of gene start prediction increased to 44/58 = 75.9%. In summary, the performance of gene start prediction of the new system ZCURVE 1.0 is much better than that of the Glimmer system,

**Table 5.** Comparison of the numbers of horizontally transferred genes found by ZCURVE 1.0 and Glimmer 2.02, respectively, for 10 complete bacterial or archaeal genomes[a]

| Organism | No. of horizontally transferred genes | Horizontally transferred genes found by ZCURVE 1.0 (%) | Horizontally transferred genes found by Glimmer 2.02 (%) |
|---|---|---|---|
| *C.acetobutylicum* | 146 | 99.3 | 98.6 |
| *L.monocytogenes* | 184 | 99.5 | 97.8 |
| *B.subtilis* | 552 | 98.0 | 94.2 |
| *T.acidophilum* | 145 | 97.2 | 93.1 |
| *M.thermoautotrophicum* | 178 | 93.8 | 86.5 |
| *E.coli* | 359 | 97.8 | 89.4 |
| *S.meliloti* | 179 | 97.8 | 94.4 |
| *P.aeruginosa* | 307 | 87.6 | 96.1 |
| *R.solanacearum* | 356 | 91.6 | 93.3 |
| *S.coelicolor* | 541 | 92.4 | 90.9 |
| Average | – | 95.50 ± 3.95 | 93.43 ± 3.73 |

[a]The data of horizontally transferred genes were downloaded from the HGT-DB database (http://www.fut.es/~debb/HGT/) (19).The figure following ± is the standard deviation.

**Table 6.** Summary of the accuracy of gene start prediction with ZCURVE 1.0[a]

| Test set | No. of genes | Genes found[b] | Start prediction accuracy[c] |
|---|---|---|---|
| Set 1 (*E.coli*) | 195 | 195/195 = 100% | 180/195 = 92.3% |
| Set 2 (*E.coli*) | 811 | 799/811 = 98.5% | 719/811 = 88.7% |
| Set 3 (*B.subtilis*) | 58 | 57/58 = 98.3% | 54/58 = 93.1% |

[a]The data of test sets 1, 2 and 3 are obtained from Link *et al.* (18), Rudd (20) and Yada *et al.* (6), respectively.
[b]Denotes the case where the 3′ end prediction (and not necessarily the 5′ end prediction) matches the annotation in the test set.
[c]Denotes the case where both the 3′ end and 5′ end predictions match the annotation in the test set.

and slightly better than the performance of the same system after the post-processing by RBSfinder (16).

### Comparison with Glimmer 2.02 (IV): about the false positive prediction rate

One of the remarkable differences between ZCURVE 1.0 and Glimmer 2.02 is that the former has a much lower additional prediction rate than that of the latter, especially for the genomes with high G+C content (e.g. G+C content >0.56). Since the annotation is not perfect, in extreme cases, two different explanations may be derived from this fact: either Glimmer's predictions are correct, i.e. the annotation under-predicts many genes, or ZCURVE's predictions are correct, i.e. Glimmer has a high false positive prediction rate. To critically examine which possibility is right, the genome of *P.aeruginosa* is studied here as an example. The numbers of genes annotated, predicted by ZCURVE 1.0 and by Glimmer 2.02 for this genome are 5565, 6672 and 8647, respectively. Out of the 6672 (8647) genes predicted by ZCURVE 1.0 (Glimmer 2.02), 5466 (5503) match the annotated genes in GenBank. Accordingly, 1206 and 3144 genes are additionally predicted by ZCURVE 1.0 and Glimmer 2.02, respectively. On the other hand, it was observed by many researchers that for GC-rich prokaryotic genomes, such as *P.aeruginosa,* the G+C content at the third codon position ($GC_3$) is generally greater than that at the first codon position ($GC_1$) for most protein-coding genes (21,22). Figure 1 shows the distributions of $GC_3$ versus $GC_1$ for 405 function-known genes verified experimentally (23), 1206 and 3144 genes additionally predicted by ZCURVE 1.0 and Glimmer 2.02, respectively. As can be seen clearly, the points corresponding to the function-known genes verified experimentally are situated almost all at the region of $GC_3 > GC_1$, whereas those for the 1206 and 3144 genes additionally predicted by ZCURVE and Glimmer are mainly situated at regions of $GC_3 > GC_1$ and $GC_3 < GC_1$, respectively. This fact indicates that most of the 3144 genes additionally predicted by Glimmer 2.02 are very unlikely to code for proteins, implying that Glimmer 2.02 has a high false positive prediction rate for this genome. Not only for GC-rich genomes, but also for other genomes [as shown in Wang and Zhang (9)], Glimmer 2.02 generally has a high false positive prediction rate. Therefore, the lower (higher) additional prediction rate of ZCURVE 1.0 (Glimmer 2.02) generally implies a lower (higher) false positive prediction rate, especially for the genomes with G+C content >0.56.

### Seeking seed ORFs for genomes with the G+C content greater than 56%

It should be pointed out that the organizations of genomes with relatively higher G+C content appear to be different to those with relatively lower G+C content. To demonstrate one of the differences, we define a new parameter called 'overlapping ratio of long ORFs' in a genome, denoted by $p$:

$$p = \frac{\text{The total number of ORFs longer than 500 bp in a genome}}{\text{The number of ORFs longer than 500 bp that do not overlap with others}}. \quad \mathbf{7}$$

Obviously, $p \geqslant 1$. The average value of $p$ over the 18 bacterial or archaeal genomes studied here is 52.69, whereas the average $p$ value over 14 genomes with relatively lower G+C content is only 1.77. Generally speaking, the value of $p$ is different for different genomes. The distribution of $p$ as a function of the G+C content for the 18 bacterial or archaeal genomes studied here is shown in Figure 2. As we can see, the values of $p$ for genomes with relatively lower G+C content are almost a constant ($p \approx 2$). Fitting the points in Figure 1 by an exponential curve, we have found that the curve has a turning point at about G+C = 56%, starting from which the value of $p$

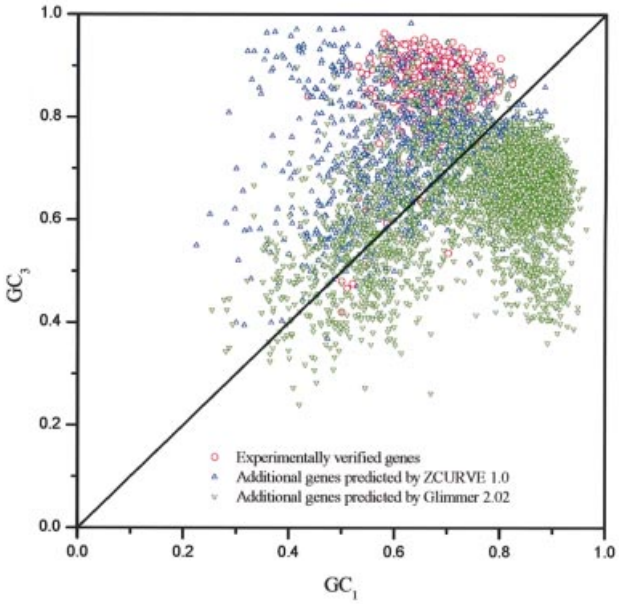


**Figure 1.** Distributions of points of $GC_3$ versus $GC_1$ corresponding to 405 function-known genes verified experimentally (23), 1206 and 3144 genes additionally predicted by ZCURVE 1.0 and Glimmer 2.02, respectively, for the genome of *P.aeruginosa*. Here $GC_3$ and $GC_1$ denote the GC content at the third and first codon positions, respectively. Note that the points corresponding to the function-known genes verified experimentally are situated almost all at the region of $GC_3 > GC_1$, whereas those for the 1206 and 3144 genes additionally predicted by ZCURVE and Glimmer are situated mainly at the regions of $GC_3 > GC_1$ and $GC_3 < GC_1$, respectively. This fact indicates that most of the 3144 genes additionally predicted by Glimmer 2.02 are very unlikely to code for proteins, implying that Glimmer 2.02 has a high false positive prediction rate for this genome.

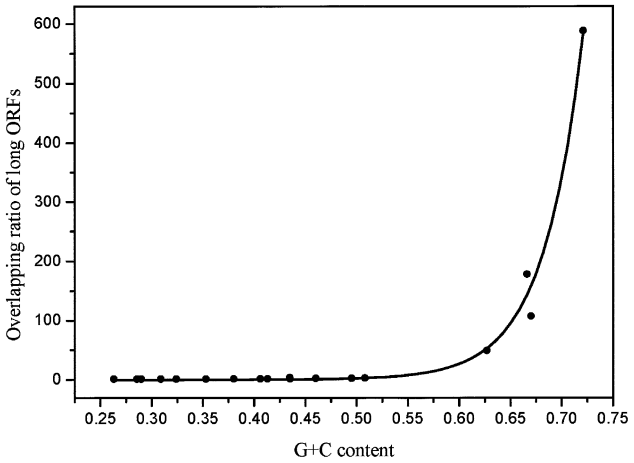


**Figure 2.** Relation between the overlapping ratio of long ORFs defined in equation **7** and the G+C content. The mean overlapping ratio averaged over 18 bacterial or archaeal genomes studied here is 52.69, whereas the mean overlapping ratio averaged over 14 bacterial or archaeal genomes with relatively lower G+C content is only 1.77. Fitting the points by an exponential curve, it is found that the curve has a turning point at about G+C = 56%, starting from which the value of $p$ increases remarkably.

**Table 7.** The coordinates of the super-sphere center in the nine-dimensional space spanned by $u_1 \sim u_9$

| | First codon position | Second codon position | Third codon position |
|---|---|---|---|
| $x$ | $u_1^0 =$ 0.230501 | $u_4^0 =$ –0.081112 | $u_7^0 =$ –0.139324 |
| $y$ | $u_2^0 =$ –0.072228 | $u_5^0 =$ 0.039828 | $u_8^0 =$ 0.111210 |
| $z$ | $u_3^0 =$ –0.382034 | $u_6^0 =$ 0.077345 | $u_9^0 =$ –0.754366 |

increases remarkably. Consequently, when the G+C content of a genome is ⩾56%, the number of ORFs longer than 500 bp that do not overlap with any other ORFs is too limited to be used. For example, a total of 14 284 ORFs longer than 500 bp are found in the genome of *P.aeruginosa* PA01 (the G+C content is 66.56%). Of the 14 284 ORFs, only 80 ORFs do not overlap with any others, which are not sufficient to be used as seed ORFs. Therefore, the strategy of seeking seed ORFs based on the method of 'long and non-overlapping ORFs' does not work in genomes with the G+C content ⩾56%. We have to look for a new way to seek seed ORFs in these genomes. The technique we adopted is the so-called 'method of nine-dimensional super-sphere', which is explained as follows.

First of all, the coordinates of a center in the nine-dimensional space are calculated, which consists of the following steps: (i) starting from the annotation files of *Caulobacter crescentus*, *Deinococcus radiodurans* R1 chromosome 1, *Halobacterium* sp. NRC-1 and *P.aeruginosa* PA01 in GenBank, select all of the function-known genes longer than 500 bp; (ii) calculate the nine parameters defined in equation **3** for each of the genes selected; (iii) calculate the average values of these nine parameters over all the genes selected, and the result is listed in Table 7. From a geometrical point of view, the nine average values represent the coordinates of a center O in the nine-dimensional space. We should point out that although the nine average values are derived from the annotation files of GenBank, they are treated as universal constants in the algorithm. In other words, these constants are applicable to all bacterial or archaeal genomes with a G+C content ⩾56%. Secondly, the procedure to look for the seed ORFs consists of the following steps: (i) calculate the nine parameters defined in equation **3** for each of the ORFs in a genome to be studied; (ii) those ORFs whose mapping points are situated within the nine-dimensional super-sphere centered at O with a radius $r$ are treated as seed; (iii) increase the $r$ value gradually until the number of seed ORFs selected is greater than or equal to 250. Consequently, more than 250 seed ORFs for a genome are found by this method.

## Joint applications of ZCURVE 1.0 and Glimmer 2.02

As mentioned previously, ZCURVE 1.0 and Glimmer 2.02 are based on different principles. Glimmer is a Markov-chain-based method, which reflects local statistical characteristics of coding sequences, whereas ZCURVE is based mainly on global statistical characteristics of coding sequences. Therefore, the two algorithms should be essentially complementary. For joint applications of both systems, a jury-decision algorithm may be adopted. According to the jury-decision algorithm, genes predicted by both systems simultaneously are finally predicted as genes, otherwise, the one predicted by any individual system is considered to be non-coding. To demonstrate the effect of this joint application of both systems, the genomes of *B.subtilis, E.coli* K12, *P.aeruginosa* PA01 and *S.meliloti* were used to test the jury-decision algorithm. The result is listed in Table 8. As we can see from Table 8, although the accuracy is decreased slightly, the additional prediction rate has been

**Table 8.** Joint applications of ZCURVE 1.0 and Glimmer 2.02 for *B.subtilis*, *E.coli*, *S.meliloti* and *P.aeruginosa*, respectively[a]

| Organism | | *B.subtilis* | *E.coli* | *S.meliloti* | *P.aeruginosa* |
|---|---|---|---|---|---|
| ZCURVE 1.0 | Annotated genes found (%) | 98.4 | 98.2 | 99.3 | 98.2 |
| | Additional genes found (%) | 19.5 | 23.6 | 37.1 | 21.7 |
| Glimmer 2.02 | Annotated genes found (%) | 98.2 | 96.9 | 98.4 | 98.9 |
| | Additional genes found (%) | 26.6 | 23.4 | 62.2 | 56.5 |
| Both | Annotated genes found (%) | 97.5 | 96.2 | 97.8 | 97.5 |
| | Additional genes found (%) | 8.5 | 6.1 | 14.5 | 7.1 |

[a]'Both' means that joint applications of ZCURVE 1.0 and Glimmer 2.02 are based on a jury-decision algorithm. Genes predicted by both systems simultaneously are finally predicted as genes, otherwise, the one predicted by any individual system is considered to be non-coding. Note that the additional prediction rate has been greatly reduced by using this joint method.

greatly reduced by the joint method, compared with any individual system. Therefore, joint applications of both systems lead to better gene-finding results in bacterial and archaeal genomes.

## CONCLUSION

Although the gene-finding issue in bacterial and archaeal genomes is relatively easier than that in eukaryotic genomes, many problems have not yet been completely solved. The new system proposed here represents an alternative effort to solve the issue. Compared with the Markov-model-based methods, the Z curve method is much simpler, as reflected by the fact that only 33 parameters are needed to describe coding sequences, whereas usually more than 10 000 parameters are used in, say, the GeneMark system. As a result, the new system essentializes the statistical properties of the coding sequences and needs less data to be trained than the Markov-model-based methods. Therefore, the new system ZCURVE 1.0 is more reliable, especially for small genomes. Additionally, high gene-finding accuracy and low additional prediction rates are achieved, and, in particular, the performance of ZCURVE 1.0 for recognizing short and horizontally transferred genes is satisfactory. Furthermore, the joint applications of ZCURVE 1.0 with some Markov-model-based systems, say, Glimmer 2.02, lead to better gene-finding results in bacterial and archaeal genomes. All these features indicate that the new gene-finding system ZCURVE 1.0 will be a useful tool for annotation pipelines for bacterial and archaeal genomes.

## ACKNOWLEDGEMENTS

We thank Dr Salzberg for kindly sending us Glimmer 2.02 used for comparison with the new system ZCURVE 1.0. We also thank Dr Ju Wang for checking the early version of the program. Discussions with Ren Zhang are gratefully acknowledged. The present study was supported in part by the 973 Project grant G1999075606 of China.

## REFERENCES

1. Borodovsky,M. and McIninch,J. (1993) GenMark: parallel gene recognition for both DNA strands. *Comput. Chem.*, **17**, 123–133.
2. Besemer,J., Lomsadze,A. and Borodovsky,M. (2001) GeneMarkS: a self-training method for prediction of gene starts in microbial genomes. Implications for finding sequence motifs in regulatory regions. *Nucleic Acids Res.*, **29**, 2607–2618.
3. Salzberg,S.L., Delcher,A.L., Kasif,S. and White,O. (1998) Microbial gene identification using interpolated Markov models. *Nucleic Acids Res.*, **26**, 544–548.
4. Delcher,A.L., Harmon,D., Kasif,S., White,O. and Salzberg,S.L. (1999) Improved microbial gene identification with GLIMMER. *Nucleic Acids Res.*, **27**, 4636–4641.
5. Frishman,D., Mironov,A., Mewes,H.W. and Gelfand,M. (1998) Combining diverse evidence for gene recognition in completely sequenced bacterial genomes [published erratum appears in *Nucleic Acids Res.*, **26**, 3870]. *Nucleic Acids Res.*, **26**, 2941–2947.
6. Yada,T., Totoki,Y., Takagi,T. and Nakai,K. (2001) A novel bacterial gene-finding system with improved accuracy in locating start codons. *DNA Res.*, **8**, 97–106.
7. Zhang,C.-T. and Zhang,R. (1991) Analysis of distribution of bases in the coding sequences by a diagrammatic technique. *Nucleic Acids Res.*, **19**, 6313–6317.
8. Zhang,C.-T. and Wang,J. (2000) Recognition of protein coding genes in the yeast genome at better than 95% accuracy based on the Z curve. *Nucleic Acids Res.*, **28**, 2804–2814.
9. Wang,J. and Zhang,C.-T. (2001) Identification of protein-coding genes in the genome of *Vibrio cholerae* with more than 98% accuracy using occurrence frequencies of single nucleotides. *Eur. J. Biochem.*, **268**, 4261–4268.
10. Fickett,J.W. (1982) Recognition of protein coding regions in DNA sequences. *Nucleic Acids Res.*, **10**, 5303–5318.
11. Staden,R. (1984) Measurements of the effects that coding for a protein has on a DNA sequence and their use for finding genes. *Nucleic Acids Res.*, **12**, 551–567.
12. Zhang,C.-T. and Chou,K.-C. (1994) A graphic approach to analyzing codon usage in 1562 *Escherichia coli* protein coding sequences. *J. Mol. Biol.*, **238**, 1–8.
13. Mardia,K.V., Kent,J.T. and Bibby,J.M. (1979) *Multivariate Analysis*. Academic Press, London, UK.
14. Stormo,G.D., Schneider,T.D., Gold,L. and Ehrenfeucht,A. (1982) Use of the 'Perceptron' algorithm to distinguish translational initiation sites in *E. coli*. *Nucleic Acids Res.*, **10**, 2997–3011.
15. Hannenhalli,S.S., Hayes,W.S., Hatzigeorgiou,A.G. and Fickett,J.W. (1999) Bacterial start site prediction. *Nucleic Acids Res.*, **27**, 3577–3582.
16. Suzek,B.E., Ermolaeva,M.D., Schreiber,M. and Salzberg,S.L. (2001) A probabilistic method for identifying start codons in bacterial genomes. *Bioinformatics*, **17**, 1123–1130.
17. Frishman,D., Mironov,A. and Gelfand,M. (1999) Starts of bacterial genes: estimating the reliability of computer predictions. *Gene*, **234**, 257–265.
18. Link,A.J., Robison,K. and Church,G.M. (1997) Comparing the predicted and observed properties of proteins encoded in the genome of *Escherichia coli* K-12. *Electrophoresis*, **18**, 1259–1313.
19. Garcia-Vallve,S., Guzman,E., Montero,M.A. and Romeu,A. (2003) HGT-DB: a database of putative horizontally transferred genes in prokaryotic complete genomes. *Nucleic Acids Res.*, **31**, 187–189.

20. Rudd,K.E. (2000) EcoGene: a genome sequence database for *Escherichia coli* K-12. *Nucleic Acids Res.*, **28**, 60–64.
21. Muto,A. and Osawa,S. (1987) The guanine and cytosine content of genomic DNA and bacterial evolution. *Proc. Natl Acad. Sci. USA*, **84**, 166–169.
22. Majumdar,S., Gupta,S.K., Sundararajan,V.S. and Ghosh,T.C. (1999) Compositional correlation studies among the three different codon positions in 12 bacterial genomes. *Biochem. Biophys. Res. Commun.*, **266**, 66–71.
23. Stover,C.K., Pham,X.Q., Erwin,A.L., Mizoguchi,S.D., Warrener,P., Hickey,M.J., Brinkman,F.S., Hufnagle,W.O., Kowalik,D.J., Lagrou,M., Garber,R.L., Goltry,L., Tolentino,E., Westbrock-Wadman,S., Yuan,Y., Brody,L.L., Coulter,S.N., Folger,K.R., Kas,A., Larbig,K., Lim,R., Smith,K., Spencer,D., Wong,G.K., Wu,Z., Paulsen,I.T., Reizer,J., Saier,M.H., Hancock,R.E., Lory,S. and Olson,M.V. (2000) Complete genome sequence of *Pseudomonas aeruginosa* PAO1: an opportunistic pathogen. *Nature*, **406**, 959–964.