BIOINFORMATICS

Vol. 18 no. 5 2002
Pages 679–688

# Sequence complexity profiles of prokaryotic genomic sequences: A fast algorithm for calculating linguistic complexity

Olga G. Troyanskaya [1,4], Ora Arbell [2], Yair Koren [2], Gad M. Landau [2,3] and Alexander Bolshoy [1,*]

[1] Genome Diversity Center, Institute of Evolution, University of Haifa, Haifa, Israel,
[2] Department of Computer Science, University of Haifa, Haifa, Israel and
[3] Department of Computer and Information Science, Polytechnic University, NY, USA

Received on January 29, 2001; revised on October 28, 2001; accepted on November 2, 2001

## ABSTRACT

**Motivation:** One of the major features of genomic DNA sequences, distinguishing them from texts in most spoken or artificial languages, is their high repetitiveness. Variation in the repetitiveness of genomic texts reflects the presence and density of different biologically important messages. Thus, deviation from an expected number of repeats in both directions indicates a possible presence of a biological signal. Linguistic complexity corresponds to repetitiveness of a genomic text, and potential regulatory sites may be discovered through construction of typical patterns of complexity distribution.

**Results:** We developed software for fast calculation of linguistic sequence complexity of DNA sequences. Our program utilizes suffix trees to compute the number of subwords present in genomic sequences, thereby allowing calculation of linguistic complexity in time linear in genome size. The measure of linguistic complexity was applied to the complete genome of *Haemophilus influenzae*. Maps of complexity along the entire genome were obtained using sliding windows of 40, 100, and 2000 nucleotides. This approach provided an efficient way to detect simple sequence repeats in this genome. In addition, local profiles of complexity distribution around the starts of translation were constructed for 21 complete prokaryotic genomes. We hypothesize that complexity profiles correspond to evolutionary relationships between organisms. We found principal differences in profiles of the GC-rich and other (non-GC-rich) genomes. We also found characteristic differences in profiles of AT genomes, which probably reflect individual species variations in translational regulation.

**Availability:** The program is available upon request from Alexander Bolshoy or at http://csweb.haifa.ac.il/library/#complex

**Contact:** bolshoy@research.haifa.ac.il

## INTRODUCTION

Prokaryotic and eukaryotic genomes contain multiple repetitive signals for transcription and replication regulation and other cellular functions. Genomic regions with high level of repetitiveness are called low-complexity zones (LCZ). Eukaryotic genomes are more repetitive than prokaryotic genomes; nevertheless, the latter also exhibit a number of repeats (Debrauwere *et al.*, 1997; van Belkum *et al.*, 1998). Moreover, compactness of prokaryotic genomes suggests full usage of low-complexity zones for varying biological purposes. The simplest, but frequent case of a low-complexity zone in prokaryotes, is a region of simple sequence repeats (SSRs), which usually consists of homopolymeric tracts, especially poly(A) and poly(T), of di-, tri-, and tetranucleotide repeats, and of more rare multimeric repeats of longer length. SSRs are encountered in many prokaryotes enabling genetic: consequently phenotypic flexibility. SSRs function at various levels of gene expression regulation (van Belkum *et al.*, 1998). The complete genome of *H. influenzae* (Fleischmann *et al.*, 1995) has been widely studied from this point of view (Field and Wills, 1998; Hood *et al.*, 1996; van Belkum *et al.*, 1998).

There are numerous available computer programs useful for searching repeats in biological sequences. For example, REPEATS (Benson, 1999; Benson and Waterman, 1994); two programs from the EMBOSS package (http://www.uk.embnet.org/Software/EMBOSS/) named etandem and equicktandem; Approximate Tandem Repeat Program (Landau *et al.*, 2001) at Haifa University (http://csweb.haifa.ac.il/library/appro_try1.html). These programs work by scanning a DNA sequence, looking

[*] To whom the correspondence should be addressed.

[4] O.T. was a summer student at Haifa University. Her present address is Stanford Medical Informatics, Stanford University, CA 94305, USA.

for tandemly repeated patterns. We propose an effective way to search complete genomic sequences for zones of lower complexity. As an example, we demonstrate an effectiveness of the linguistic complexity method, applying it to the complete genome of *H. influenzae* (Fleischmann *et al.*, 1995).

Genomic sequences can be successfully analyzed by linguistic measures, (e.g. Brendel *et al.*, 1986; Gelfand, 1993; Konopka, 1994; Pesole, 1994; Pesole *et al.*, 1996; Pietrokovski, 1994; Searls, 1997). One fundamental characteristic of linear symbolic sequences (texts, strings) is sequence complexity, which has been defined by many methods, based on either algorithmic complexity or Shannon entropy. These methods were adopted and used in genomic analysis (Konopka, 1990, 1994; Konopka and Chatterjee, 1988; Lauc *et al.*, 1992; Wan and Wootton, 2000; Wootton and Federhen, 1996).

This work utilizes linguistic complexity (LC), a measure different from the above-mentioned methods, to analyze genomic sequences. A genetic sequence is generally characterized by very high variability in repetitiveness. This feature leads to a straightforward definition of the sequence complexity as a richness of its vocabulary—how many different substrings of length $k$ ($k$-mers) appear in the sequence. The notion of linguistic complexity was introduced in 1990 (Trifonov, 1990), and we have previously used it in the studies of nucleosomal pattern and promoters (Bolshoy *et al.*, 1997; Gabrielian and Bolshoy, 1999). This measure was also described in Pesole *et al.* (1996) and implemented in the EMBOSS package (http://www.uk.embnet.org/Software/EMBOSS/). Here we used a modified version, wherein linguistic complexity (LC) is defined as the ratio of the number of substrings of any length present in the string to the maximum possible number of substrings. Our algorithm uses implicit suffix trees constructed by Ukkonen's algorithm (Ukkonen, 1995) to count the number of substrings in the string.

As we demonstrate below, our software provides an effective way to reveal variations in linguistic complexity of genomic texts. One useful application is in searching for lower-complexity zones. These regions are dispersed along the genome and may be revealed as exclusions from the typical picture of LC variation, as we demonstrate in our study of the complete genome of *H. influenzae*. Another application of the tool is to discover potential regulatory sites through the construction of typical patterns of LC distribution in certain regions. To demonstrate this, we examined the patterns of sequence complexity around the flanks of coding sequences. When the regions around the flanks of coding sequences in many available complete prokaryotic genomes are examined, well-defined profiles of LC are observed, which divide all prokaryotic genomes into a small number of distinct groups. Such gene-flanking LC distribution can serve as a genomic identifier of the coding regions or as a genomic profile. Presented prokaryotic genomic profiles carry the most general complexity properties typical for all studied AT-biased and AT-balanced prokaryotes, as well as reflection distinguishing individual properties of species.

## METHODS AND ALGORITHMS

### Linguistic complexity

Linguistic complexity (LC) is defined as the ratio of the number of subwords (substrings) present in the string of interest to the maximum number of subwords for a string of the same length over the same alphabet. (We use the terms words and subwords as synonyms of strings and substrings, respectively.) Let us count, for example, the number of different subwords for the DNA sequence S1 = ACGGGAAGCTGATTCCA. The length of S1 is equal to 17. It contains 15 of 16 possible different dinucleotides; 15 of 15 possible different trinucleotides in a string of length 17, maximum possible number of tetranucleotides (14), and so on. In summary, S1 contains $4 + 15 + 15 + 14 + \cdots + 1 = 119$ different subwords. LC(S1) = 119/120 = 0.992. Another sequence of length 17, S2 = ACACACACACACACACA, contains only two different mononucleotides, two different dinucleotides AC and CA, two different trinucleotides ACA and CAC, and so on. S2 contains $2+2+2+2+\cdots+1 = 33$ different subwords, and LC(S2) = 33/120 = 0.275. LC($A_{17}$) = 17/120 = 0.142 is a minimal LC-value for the strings of length 17.

Our program utilizes suffix trees to compute the number of substrings present in strings of DNA. Linguistic complexity is then calculated for each window, with window size ($m$) determined by the user (sliding window approach with a step size of 1). The results are outputted in the form that allows for easy plotting and analysis.

Because the sequence analyzed was a DNA sequence, the alphabet $\Sigma$ is defined as /A, T, C, G/, all other characters that occur in genomic sequence (s, w, n, *etc* and present badly sequenced nucleotides) are converted into the most likely equivalent out of $\Sigma$. In Table 1 a simple way of conversion, which we used in our computer program, is shown.

The maximum vocabulary over word sizes 1 to $m$ can be calculated according to the following formula (where $l$ is the alphabet size and $k$ is the word length):

$$\sum_{k=1}^{m} \min(\, l^k, m - k + 1)$$

### Counting the number of subwords with suffix trees

To calculate the number of subwords in a string, we utilize suffix trees (Gusfield, 1997). The key feature of a suffix tree is that the suffix of S, starting at position $i$ (S[$i \ldots m$]),

**Table 1.**

| Ambiguous code | S | W | | R | Y | M | K | | B | D | | H | V | N |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Nucleotide | **C, G** | **A, T** | | **A, G** | **C, T** | **A, C** | **G, T** | | **C, G, T** | **A, G, T** | | **A, C, T** | **A, C, G** | **A, C, G, T** |
| Probability | 1 | $\frac{1}{3}$ | $\frac{2}{3}$ | 1 | 1 | 1 | $\frac{1}{3}$ | $\frac{2}{3}$ | 1 | $\frac{1}{3}$ | $\frac{2}{3}$ | 1 | 1 | 1 |
| Conversion | **C** | **A** | **T** | **G** | **C** | **A** | **G** | **T** | **C** | **A** | **T** | **C** | **C** | **C** |

can be obtained by concatenation of the labels on the path from the root to leaf $i$. This property makes suffix trees a natural choice for the task of calculating the number of subwords in a string. The implicit suffix tree for S is a rooted directed tree, which has at most $m$ leaves and at most $m$ internal vertices, with each internal node having at least two children. Each edge of the tree is labeled with a substring of S and no two edges out of a node can have labels starting with the same character.

This work utilizes compressed suffix trees, where each edge is labeled with a pair of indices instead of explicitly written characters. The indices specify beginning and end positions of the substring within S. As a copy of S is always available, any character in S can be located in constant time. Because the number of edges in the suffix tree for a string of length $m$ is at most $2m-1$, compression of the edge labels allows the reduction of the suffix tree to only O($m$) space.

Since by definition, suffix tree T for string S contains all possible suffixes of S, every non-terminal character on the edge labels of T represents a prefix to some suffix in S. By suffix tree construction, repeated prefixes are only represented once in T and the number of subwords in S is equal to the number of prefixes in suffixes of S. Thus the number of characters on the edge labels of T is equal to the number of unique subwords in S. Therefore, the number of non-terminal characters represented on the edges of suffix tree T for string S is equal to the number of different subwords contained in S.

Several linear-time algorithms for suffix tree construction exist (Farach, 1997; McCreight, 1976; Ukkonen, 1995; Weiner, 1973). Our software utilized Ukkonen's algorithm (Ukkonen, 1995), which is based on constructing a sequence of implicit suffix trees, the last of which, Ť for S, is converted to a true suffix tree by adding the termination character. To obtain the number of characters on the edges of Ť, no separate counting procedure is necessary, as all explicit additions of characters to the tree can be counted in the process of suffix tree construction. This is possible because once a character is added to an edge label; it will never be removed within the same window. Therefore, this algorithm allows are to calculate the number of subwords in string S in the time needed for the construction of the suffix tree for that string, which is linear time.

## Performance

For each subsequence of the genome (window), the number of subwords can be calculated in linear time using the suffix tree construction. Vocabulary size for a given alphabet and maximum word length (window size) is only calculated once for a given genome, therefore this calculation doesn't have much impact on the performance of the program. Hence, each window of length $m$ is processed in O($m$) time. We then utilize the algorithm for efficient linear time sliding of the window described in (Larsson, 1996, 1999), which eliminates the necessity to build a separate suffix tree for each window. Larsson also provided the code we use in our software. Therefore, we can calculate linguistic complexity for a genome of size $n$ in time O($n$). Using an AMD 1200 MHz machine the running time was about 1 s per Mbase of DNA sequence. This running time does not take into account creating an output file for the results (complexity per window). Creating the file takes about 9 s per Mb of information (the size of the file is about 13 Mb per Mbase of DNA sequence). The running time, when testing the program on an Intel Pentium 550 MHz, was about 2 s per Mb of information (again, without time of creating an output file).

## RESULTS AND DISCUSSION

### 1. *Repetitive sequences in H. influenzae*

An effective way to find practically all simple sequence repeats in a long genomic sequence is to apply a measure of linguistic complexity to it for detecting all zones of low complexity. To illustrate this approach, we examined the complete genome of *H. influenzae* (Fleischmann *et al.*, 1995) using our software. We chose the *H. influenzae* genome to test our software because this genome has been well studied with respect to repeated sequences. In Figure 1, we present plots of complexity distribution along the genome using window sizes of 40, 100, and 2000 bases (further referred to as LC$_{40}$, LC$_{100}$, and

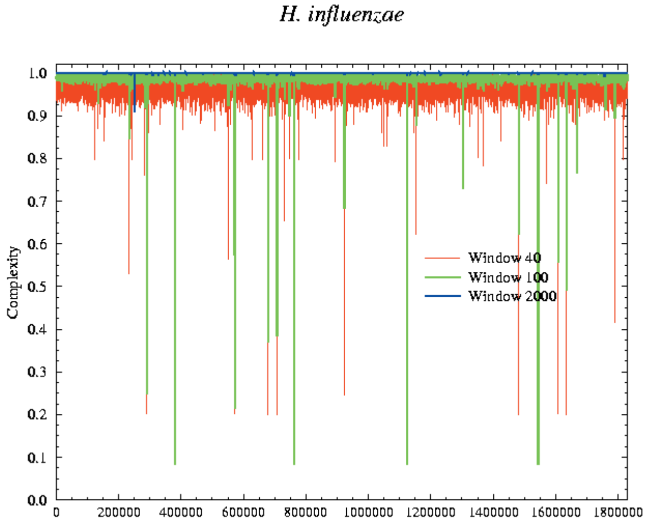

**Fig. 1.** Complexity distribution along the complete genome of *Haemophilus influenzae* (Fleischmann *et al.*, 1995). Red line corresponds to the window size of 40 nucleotides, green line corresponds to the window size of 100 bases, and blue line corresponds to the window size of 2000 bases. The size of *H. influenzae* is 1 830 138 bases. Variation of linguistic complexity measured with window of 40 bases ($LC_{40}$) rests in the range of 0.2 to 0.999; range of $LC_{100}$ is from 0.1 to 0.99; range of $LC_{2000}$ is from 0.91 to 1.0.

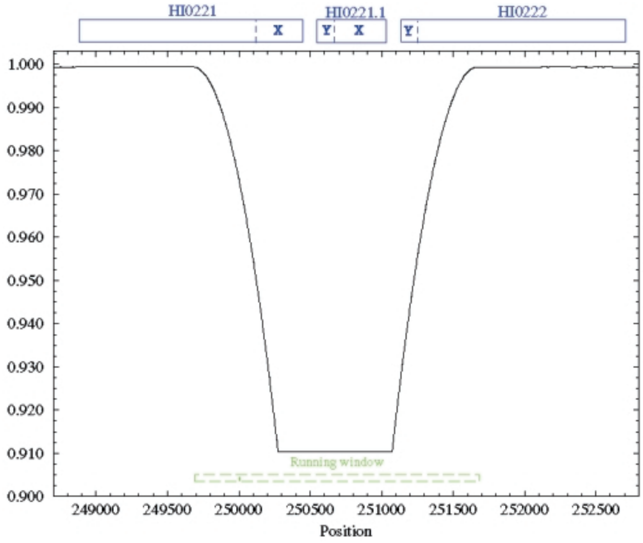

**Fig. 2.** Region in genome of *Haemophilus influenzae* including genes HI0221 (location 248948–250414), HI0221.1 (250524–251015), and HI0222 (251125–252696). http://research.haifa.ac.il/~genom/ComplexityArticle/Figures.html Distribution of sequence complexity measured with running window of 2000 bases in the region around the location 250 700 is shown. The predicted coding region HI0221.1 consists of two domains X and Y, where Y is the head domain of the gene *guaA* (GMP synthase), and X is identical to the tail of the gene *guaB* (inosine-5′-monophosphate dehydrogenase).

$LC_{2000}$). By varying the window size, we can obtain a more complete picture of the repeated region. A small-size sliding window reveals indication of relatively short repeats, while a big window may include long dispersed and degenerate repeats. The local minima indicate various repeats.

The minima regions revealed by $LC_{2000}$, a relatively large window size, should correspond to long and adjacent repeats. Let us investigate the region with the lowest value of $LC_{2000}$ (Figure 2). This global minimum is situated around the location 250,500. Three putative genes overlay this region: HI0221 (inosine-5’-monophosphate dehydrogenase—guaB), HI0221.1 (brute force ORF), and HI0222 (GMP synthase—guaA). HI0221.1 consists of two sequence modules identical to its nearest neighbors, which explains the low complexity of the region. Alignment of the appropriate regions of these three genes revealed that the first 50 amino acids of the hypothetical protein HI0221.1 (labeled Y in Figure 2) are 100% identical to the first 50 amino acids of the following protein HI0222 (guaA), and the rest 114 amino acids of HI0221.1 (notated X) are 100% identical to the tail of the preceding gene HI0221 (guaB). The head Y of this centaur came from guaA and the bottom part X is a tail of guaB. The intergenic regions (spacers of length 109 bases) between HI0221, HI0221.1, and HI0222 are also identical. Actually, the centaur HI0221.1 is a tandemly repeated region of the ‘X-spacer-Y’ of the total length $150 + 109 + 342 = 601$ nucleotides. Our procedure of mapping the genome using a large LC window allows one to recognize such potentially interesting outcomes of duplication (or artifacts of the sequence assembly procedure).

The complexity map of *H. influenzae* genome with the LC window of 100 bases is shown in green in Figure 1. The typical range of $LC_{100}$ values of *H. influenzae* is 0.97 to 0.99. The four global minima in Figure 1 all have an LC value equal to 0.08. Many local minima with $LC_{100}$ values lower than 0.95 are observed as well. These minima indicate low-complexity zones in *H. influenzae*. In Table 2, we describe 18 simplest regions of *H. influenzae* revealed by complexity mapping with $LC_{100}$ and corresponding to the local minima of the complexity map of $LC_{100}$ values in Figure 1. Notably, all simplest sequences that we identified are SSRs; practically all of them are perfect tandem repeats. However, SSRs # 8 and # 18, located around positions 746724 and 1,668,482 respectively, are imperfect long tandem repeats. Appendix 1 (http://research.haifa.ac.il/~genom/ComplexityArticle/

Appendix.doc) shows the locations and translations of the simplest regions of *H. influenzae*, with annotation of the regions both according to Fleischmann *et al.* (Fleischmann *et al.*, 1995) and Hood *et al.* (Hood *et al.*, 1996) the distribution of sizes of SSRs is of interest as well. Eleven SSRs in Table 2 are repeated tetranucleotides. The SSR #10 is a 12 times repeated pentamer; #1 and #4 are 12-mers, #12–a 15-mer, #13 and #8 are repeats 36 and 39 nucleotides in length, respectively. (More data may be found in Appendix 1.)

The linguistic complexity method thus allows us to identify biologically significant SSRs of varying sizes, as well as imperfect repeats, that may be missed by other methods. Analysis of complexity with larger window sizes (2000 base pairs) may allow identification of potentially interesting regions of duplications, as well as tagging potential genome assembly artifacts.

### **2.** *Linguistic complexity profiles around $5'$ and $3'$ ends of prokaryotic CDS*

Obviously, direct comparison of average complexity values calculated for different genomes may yield only limited results. However, typical distribution of sequence complexity around the flanks of genes exists and may serve as a kind of ‘genome signature’. To obtain complexity distributions around the ends of protein coding sequences typical to an individual prokaryotic genome, we used $LC_{50}$ distributions averaged over the CDSs with sufficient flanking non-coding regions. The window size of 50 nucleotides was empirically chosen among window sizes of 20, 50, 100, and 200 bases. This window size best highlighted the most common features for all individual genomes.

In Figure 3 (http://research.haifa.ac.il/∼genom/ComplexityArticle/Figures.html), we present the average distributions around the ends of CDS. We find a difference in the complexity profiles between the GC-rich genomes (*D. radiodurans*, *M. tuberculosis*, and *T. pallidum*) and other genomes (from here on denoted as AT genomes). All the AT genomes are rather similar in their main features: lower LC for non-coding regions with local complexity minima close to the ends of CDS (close to 0). In most AT genomes, the minimum of averaged LC distribution that is adjacent to the start of translation is very well pronounced. A small number of exclusions from this rule among AT genomes include *E. coli*, *H. influenzae*, and *M. pneumoniae*. The minimum of averaged LC distribution adjacent to the end of translation is less common or missing in *C. pneumoniae*, *E. coli*, *H. influenzae*, *M. pneumoniae*, *R. prowazekii*, *Synechocystis*, and *T. maritima*.

According to the LC profile, we can cluster genomes as ‘++’, ‘+-’, ‘-+’, and ‘–’, where ‘+’ refers to the presence of the minimum before or after the CDS, respectively. Group I, ‘++’, is characterized by both minima and contains: *A. fulgidus*, *A. aeolicus*, *M. jannaschii*, *P. abyssi*, *P. horikoshii*, *M. thermoautotrophicum*, *H. pylori*, and *M. genitalium*. Group IV, ‘–’, is characterized by absence of both minima and contains: *E. coli*, *H. influenzae*, and *M. pneumoniae*. Group II, ‘+-’, is characterized by the absence of a minimum adjacent to the end of CDS and contains: *Synechocystis*, *C. pneumonia*, *R. prowazekii*, and *T. maritima*. Group III, ‘-+’, is empty. According to this rough classification *B. subtilis*, *C. trachomatis*, and *B. burgdorferi* should be included in Group I. However, they show much wider minima in comparison with other members of the group, and resemble overall profile of Group IV.

In Figure 4, the prokaryotic representative linguistic complexity profile is shown. Seventeen LC profiles, 17 out of 21 genomes, were averaged to make a consensus (*D. radiodurans*, *M. tuberculosis*, *T. pallidum*, and *T. maritima* were excluded). Naturally, this plot mainly carries features of the dominating Group I. We obtained in a similar way consensus profiles for *Caenorhabditis elegans* and *Saccharomyces cerevisiae* (the study is in progress and data not shown). Interestingly, these profiles have common features missing in prokaryotic profiles—they belong to Group III.

Different complexity profiles thus roughly correspond to evolutionary relationships between organisms, and may reflect individual variations in translational regulation. As these profiles are conserved in the groups of organisms, the variations in sequence repetitiveness may indicate important regulatory sites. Linguistic complexity methods may therefore aid in comparative studies of translational regulation across genomes**.**

## CONCLUSIONS

This study demonstrates usefulness of global complexity profiles of DNA sequences as large as complete genomes or chromosomes. We show that the method of linguistic complexity is effective for description of typical genomic features through applying the LC measure to 21 entire prokaryotic genomes. We found principal differences in the complexity profiles of the GC-rich genomes (*D. radiodurans*, *M. tuberculosis*, and *T. pallidum*) and other (AT) genomes. Major features of profiles of AT prokaryotic genomes are the location of relatively simple regions of about 50 bp immediately before the starts of translation and immediately after the ends of CDS. In some prokaryotes, especially in hyperthermophiles, these minima are remarkably sharp. In fact, all Archea that we examined by the linguistic complexity method have similar LC profiles and belong to Group I described above. In some bacteria, for example *E. coli*, there are no significant minima after the end of translation, and there is an additional minimum of

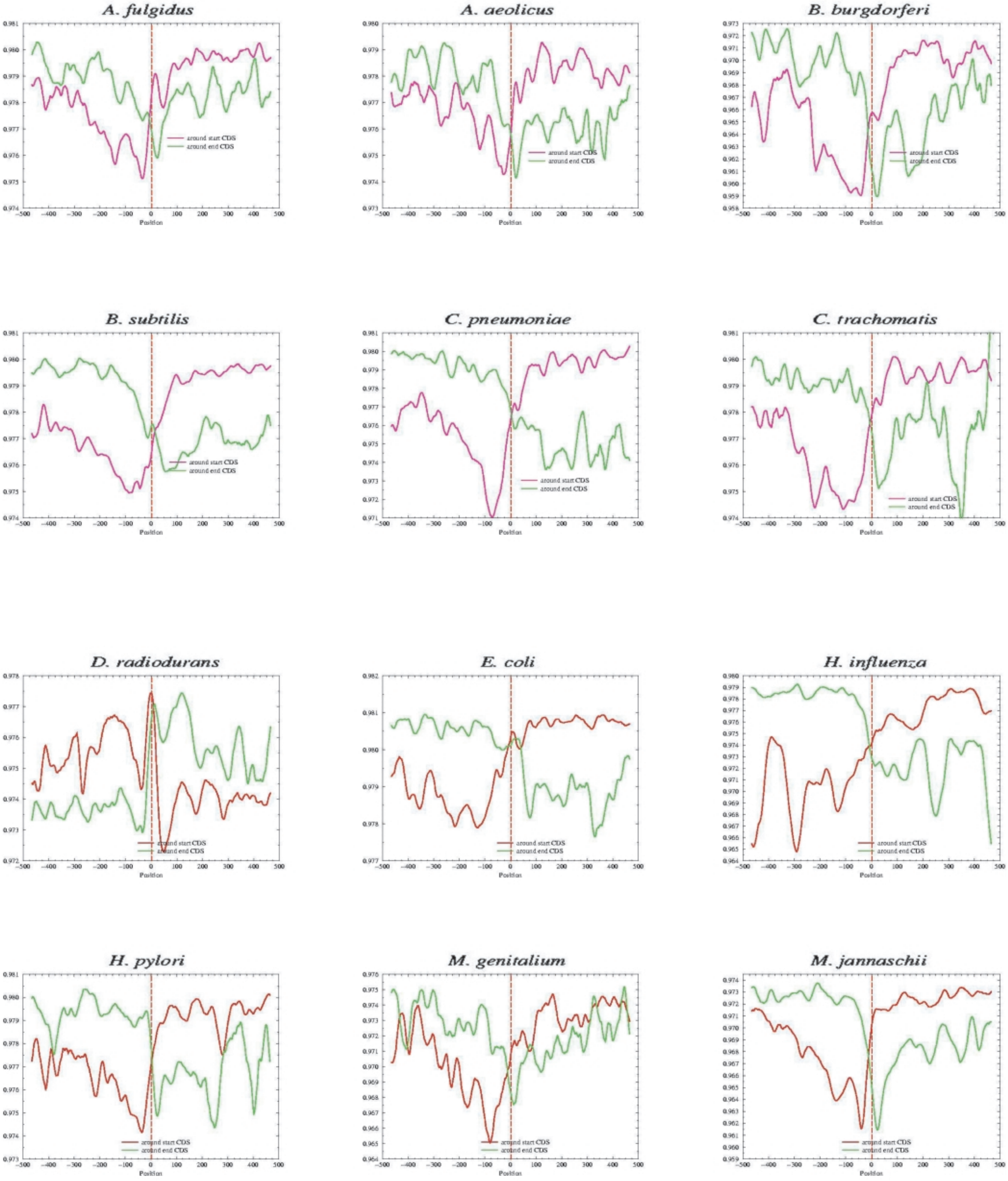
A. fulgidus
A. aeolicus
B. burgdorferi
B. subtilis
C. pneumoniae
C. trachomatis
D. radiodurans
E. coli
H. influenza
H. pylori
M. genitalium
M. jannaschii
around start CDS
around end CDS
Position
−500
−400
−300
−200
−100
0
100
200
300
400
500

Fig. 3. Cont.

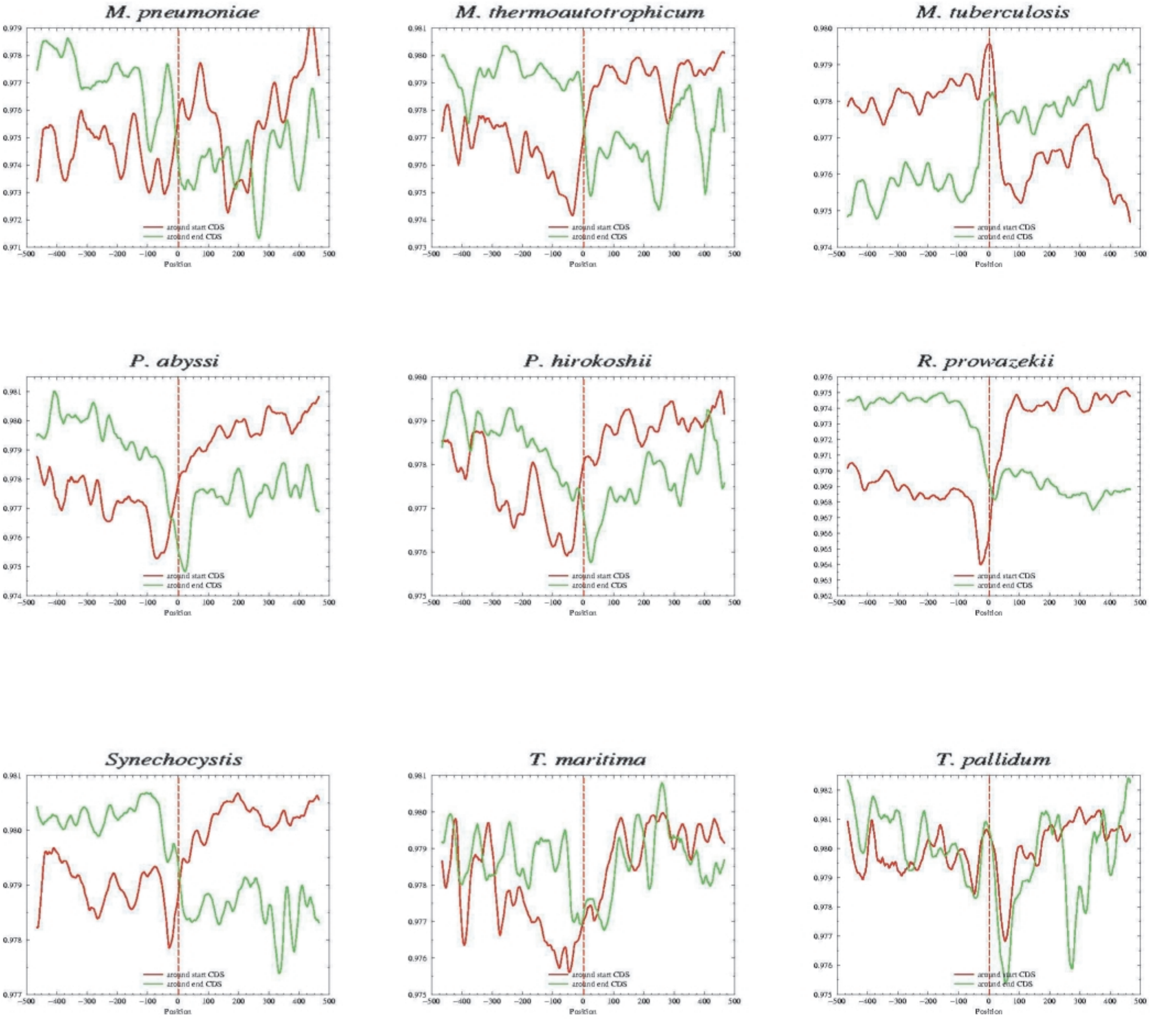

**Fig. 3.** Profiles of complexity distributions in the neighborhoods of the starts and ends of translation. http://research.haifa.ac.il/~genom/ComplexityArticle/Figures.html Complexity was measured by a running window of 50 bases. Only the starts of CDS longer than 125 nucleotides and flanked by upstream intergenic regions longer than 125 nucleotides were processed. Zero corresponds to 'the start of CDS' for complexity (LC) distribution around 5′-end of CDS (red line), and to 'the end of CDS' for LC distribution around 3′-end of CDS (green line). All the neighborhoods of the starts and ends of genes were constructed from strictly noncoding–coding and, correspondingly, coding–noncoding pairs of regions. The mean distributions (profiles shown) were obtained by averaging the distributions of all fragments from the same genome.

complexity in promoter regions—around −200–230 bp upstream from the start of translation. (Compare with Gabrielian and Bolshoy, 1999; Gabrielian *et al.*, 1999. It is interesting to note that prokaryotic LC profiles are distinct from those we observe for eukaryotic genomes (study in preparation), possibly reflecting differences in translational machinery. This topic requires further investigation and the study is currently in progress.

This 'complexity fingerprint' is also a fast and effective method to reveal low-complexity zones, which in many cases are SSRs. Application of the method of linguistic

**Table 2.** The simplest sequence repeats of *H. influenzae* revealed by linguistic complexity using a sliding window of 100 bp. Values in the column 'Complexity' are related to fragments of 100 bp size centered on locations indicated in the column 'Location'

| # | Location | Complexity | Repeat of repeat | Length SSR | Length of repeats | # of exact |
|---|---|---|---|---|---|---|
| 1 | 234001 | 0.85 | CTTACCAGCGAG | 12 | 52 | 4 |
| 2 | 289383 | 0.25 | CTGT | **4** | 108 | 22 |
| 3 | 380160 | 0.08 | TTGA | **4** | 135 | 33 |
| 4 | 550299 | 0.92 | AATTTAGGTTCA | 12 | 63 | 3 |
| 5 | 571431 | 0.21 | TTGA | **4** | 92 | 23 |
| 6 | 677767 | 0.37 | TTGG | **4** | 84 | 21 |
| 7 | 706532 | 0.38 | TTGG | **4** | 84 | 20 |
| 8 | 746724 | 0.90 | CATCTTCATCATCAA<br>AAAATTCCCCATCGT<br>CACCGTATT | 39 | 90 | 2 |
| 9 | 761165 | 0.08 | TTGG | **4** | 150 | 37 |
| 10 | 922713 | 0.68 | TTATC | 5 | 65 | 12 |
| 11 | 1123559 | 0.08 | TGAC | **4** | 128 | 32 |
| 12 | 1152772 | 0.88 | AAAGTTATAGAGAGG | 15 | 55 | 3 |
| 13 | 1303146 | 0.73 | CTTGTGCAGTAGTAT<br>CAGGAGCTGCTGCCT GTGGTG | 36 | 144 | 2 |
| 14 | 1481834 | 0.62 | AACC | **4** | 66 | 16 |
| 15 | 1543789 | 0.08 | TTGC | **4** | 100 | 25 |
| 16 | 1608642 | 0.56 | CAAT | **4** | 72 | 17 |
| 17 | 1633831 | 0.49 | CCAA | **4** | 76 | 19 |
| 18 | 1668482 | 0.77 | AGCAGATTTAGCTTT<br>GTCTGCACCGCATTT<br>GCCTTCACCACATTT ACCTTC | 51 | 160 | 2 |

complexity mapping to the complete genome of *H. influenzae* reveals all known SSRs; practically all of which are perfect tandem repeats. Our method is sensitive to long imperfect repeats, as well.

## ACKNOWLEDGEMENTS

We wish to thank Ed Trifonov and Valery Kirzhner for helpful discussions. We are grateful to Dr Larsson who permitted to use his code in our software. A.B. is a Guastaella Fellow, and is partially supported by the Fondation Sacta-Rachi, the Ancell Teicher Research Foundation for Molecular Genetics and Evolution, and by the FIRST Foundation of the Israel Academy of Science and Humanities. O.G.T. and O.A. are partially supported by the Israel Science Foundation, founded by the Israeli Academy of Sciences and Humanities. G.L. is partially supported by NSF grants CCR-9610238 and CCR-0104307, by NATO Science Programme grant PST.CLG.977017, by the Israel Science Foundation grants 173/98 and 282/01, by the FIRST Foundation of the Israel

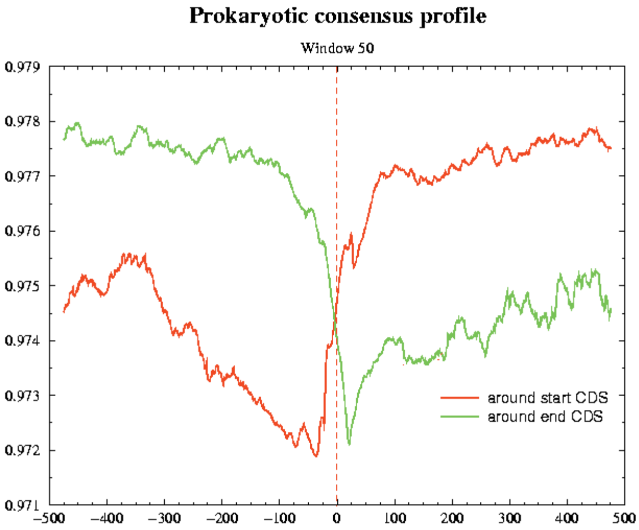

**Fig. 4.** Consensus prokaryotic genome profile. The averaged distributions presented in Figure 3 were averaged over 17 out of 21 genomes (*D. radiodurans*, *M. tuberculosis*, *T. pallidum*, and *T. maritima* were excluded) to obtain the combined consensus distributions.

Academy of Science and Humanities, and by IBM Faculty Partnership Award.

## REFERENCES

Benson,G. (1999) Tandem repeats finder: a program to analyze DNA sequences. *Nucleic Acids Res.*, **27**, 573–580.

Benson,G. and Waterman,M.S. (1994) A method for fast database search for all *k*-nucleotide repeats. *Nucleic Acids Res.*, **22**, 4828–4836.

Bolshoy,A., Shapiro,K., Trifonov,E.N. and Ioshikhes,I. (1997) Enhancement of the nucleosomal pattern in sequences of lower complexity. *Nucleic Acids Res.*, **25**, 3248–3254.

Brendel,V., Beckmann,J.S. and Trifonov,E.N. (1986) Linguistics of nucleotide sequences: morphology and comparison of vocabularies. *J. Biomol. Struct. Dyn.*, **4**, 11–21.

Debrauwere,H., Gendrel,C.G., Lechat,S. and Dutreix,M. (1997) Differences and similarities between various tandem repeat sequences: Minisatellites and microsatellites. *Biochimie*, **79**, 577–586.

Farach,M. (1997) Optimal suffix tree construction with large alphabets. *38th IEEE Symposium on Foundations of Computer Science*, pp. 137–143.

Field,D. and Wills,C. (1998) Abundant microsatellite polymorphism in *Saccharomyces cerevisiae*, and the different distributions of microsatellites in eight prokaryotes and *S. cerevisiae*, result from strong mutation pressures and a variety of selective forces. *Proc. Natl Acad. Sci. USA*, **95**, 1647–1652.

Fleischmann,R.D., Adams,M.D., White,O., Clayton,R.A., Kirkness,E.F., Kerlavage,A.R., Bult,C.J., Tomb,J.F., Dougherty,B.A., Merrick,J.M. and al,e. (1995) Whole-genome random sequencing and assembly of *Haemophilus influenzae* Rd. *Science*, **269**, 496–512.

Gabrielian,A.E. and Bolshoy,A. (1999) Sequence complexity and DNA curvature. *Comput. Chem.*, **23**, 263–274.

Gabrielian,A.E., Landsman,D. and Bolshoy,A. (1999) Curved DNA in promoter sequences. *In Silico Biol.*, **1**, 183–196.

Gelfand,M.S. (1993) Genetic language: metaphore or analogy? *Biosystems*, **30**, 277–288.

Gusfield,D. (1997) *Algorithms on Strings, Trees, and Sequences: Computer Science and Computational Biology*. Cambridge University Press, Cambridge [England].

Hood,D.W., Deadman,M.E., Jennings,M.P., Bisercic,M., Fleischmann,R.D., Venter,J.C. and Moxon,E.R. (1996) DNA repeats identify novel virulence genes in *Haemophilus influenzae*. *Proc. Natl Acad. Sci. USA*, **93**, 11121–11125.

Konopka,A.K. (1990) Towards mapping functional domains in indiscriminantly sequenced nucleic acids: a computational approach. In Sarma,R.H. and Sarma,M.H. (eds), *Structure & Methods*, vol. 1, Adenine Press, Albany, pp. 113–125.

Konopka,A.K. (1994) Sequences and codes: fundamentals of biomolecular cryptology. In Smith,D. (ed.), *BIOCOMPUTING: Informatics and Genome Projects*. Academic Press, San-Diego.

Konopka,A.K. and Chatterjee,D. (1988) Distance analysis and sequence properties of functional domains in nucleic acids and proteins. *Gene Anal. Tech.*, **5**, 87–93.

Landau,G.M., Schmidt,J.P. and Sokol,D. (2001) An algorithm for approximate tandem repeats. *J. Comput. Biol.*, **8**, 1–18.

Larsson,J.N. (1999) Structure of string matching and data compression. Doctoral, Lund University.

Larsson,N.J. (1996) Extended application of suffix trees to data compression. *IEEE Data Compression*, pp. 190–199.

Lauc,G., Ilic,I. and Heffer-Lauc,M. (1992) Entropies of coding and noncoding sequences of DNA and proteins. *Bioph. Chem.*, **42**, 7–11.

McCreight,E.M. (1976) A space-economical suffix tree construction algorithm based system and its evaluation. *J. ACM*, **23**, 262–272.

Pesole,G., Attimonelli,M. and Saccone,C. (1994) Linguistic approaches to the analysis of sequence information. *Trends Biotechnol.*, **12**, 401–408.

Pesole,G., Attimonelli,M. and Saccone,C. (1996) Linguistic analysis of nucleotide sequences: algorithms for pattern recognition and analysis of codon strategy. *Meth. Enzymol.*, **266**, 281–294.

Pietrokovski,S. (1994) Comparing nucleotide and protein sequences by linguistic methods. *J. Biotechnol.*, **35**, 257–272.

Searls,D.B. (1997) Linguistic approaches to biological sequences. *Comput. Appl. Biosci.*, **13**, 333–344.

Trifonov,E.N. (1990) Making sense of the human genome. In Sarma,R.H. and Sarma,M.H. (eds), *Structure & Methods*, vol. 1, Adenine Press, Albany, pp. 69–77.

Ukkonen,E. (1995) Online construction of suffix trees. *Algorithmica*, **14**, 249–260.

van Belkum,A., Scherer,S., van Alphen,L. and Verbrugh,H. (1998) Short-sequence DNA repeats in prokaryotic genomes. *Microbiol.*

*Mol. Biol. Rev.*, **62**, 275–293.

Wan,H. and Wootton,J.C. (2000) A global compositional complexity measure for biological sequences: AT-rich and GC-rich genomes encode less complex proteins. *Comput. Chem.*, **24**, 71–94.

Weiner,P. (1973) Linear pattern matching algorithm. *14th IEEE Symp. on Switching and Automata Theory*. pp. 1–11.

Wootton,J.C. and Federhen,S. (1996) Analysis of compositionally biased regions in sequence databases. *Meth. Enzymol.*, **266**, 554–571.