SeeDNA: A Visualization Tool for *K*-string Content of Long DNA Sequences and Their Randomized Counterparts

Junjie Shen[1], Shuyu Zhang[2], Hoong-Chien Lee[3], and Bailin Hao[2,4]*

[1] *Department of Computer Science, Zhejiang University, Hangzhou 310027, China;* [2] *T-Life Research Center, Fudan University, Shanghai 200433, China;* [3] *Department of Physics, National Central University, Chungli, Taiwan 320, China;* [4] *Institute of Theoretical Physics, Chinese Academy of Sciences, Beijing 100080, China.*

An interactive tool to visualize the *K*-string composition of long DNA sequences including bacterial complete genomes is described. It is especially useful for exploring short palindromic structures in the sequences. The SeeDNA program runs on Red Hat Linux with GTK+ support. It displays two-dimensional (2D) or one-dimensional (1D) histograms of the *K*-string distribution of a given sequence and/or its randomized counterpart. It is also capable of showing the difference of *K*-string distributions between two sequences. The C source code using the GTK+ package is freely available.

Key words: *K*-tuple, visualization, palindrome, randomized sequence

Introduction

The study of K-string composition of long DNA sequences including complete genomes is a natural extension of $G + C$ content, *i.e.*, $K = 1$, analysis. Using K values greater than 1 takes into account short-range (up to $K - 1$) correlations of nucleotides and enhances species-specific features in the sequence. Visualization of the K-string distribution on a computer screen using a crude color code is essentially a kind of coarse-graining that helps to highlight some prominent feature of the DNA sequence. For example, short palindromic strings of a certain type are avoided or under-represented in some bacterial genomes, leading to quite specific 2D histograms, while in mammalian genomic sequences the lower content of the dinucleotide CG as compared to GC dominates the picture (*1*, *2*). Two-dimensional portraits of the human chromosome 22 and the genomes of three bacteria are given in Figure 1. The similar patterns in the portraits of *Escherichia coli* and *Shigella flexneri* are caused by the under-representation of strings that contain $CTAG$ as substrings. The species-specific "avoidance signature" of complete bacterial genomes has eventually led to a new way of inferring phylogenetic relationship of prokaryotes without sequence alignment (*3*, *4*).

Studies of the 1D histograms of extant complete genomes in contrast to their random counterparts have revealed the existence of universal length in complete genomes that can be explained by a simple universal model for genome growth and evolution (*5*, *6*). In order to see that an observed feature does not occur in a random sequence, it is desirable to have randomization function built-in. A seemingly surprising effect is the appearance of fine structures in some randomized prokaryotic genomes with significant $G + C$ bias. In Figure 2, the 1D histograms for $K = 4$ to 9 are shown for *Mycobacterium tuberculosis* whose G and C make 65.6% of the genome. At fixed K, a total of $K + 1$ peaks can be seen in the histogram. This phenomenon has been fully understood. In particular, each peak may be well approximated by a Poisson distribution (*7*).

The visualization tool used by us to obtain the aforementioned results in the cited papers has been improved over the years. From a UNIX command line tool for 2D histograms based on Xlib and Xtoolkit (the old code is available at request to the corresponding author), it has evolved into a Linux software using the GTK+ package with a user-friendly graphic interface. We hereby describe this tool and put it into public domain.

Algorithm and Features

In order to count the frequency of occurrence of K-strings, a total of 4^K counters are needed. To visualize

*** Corresponding author.**
E-mail: hao@itp.ac.cn

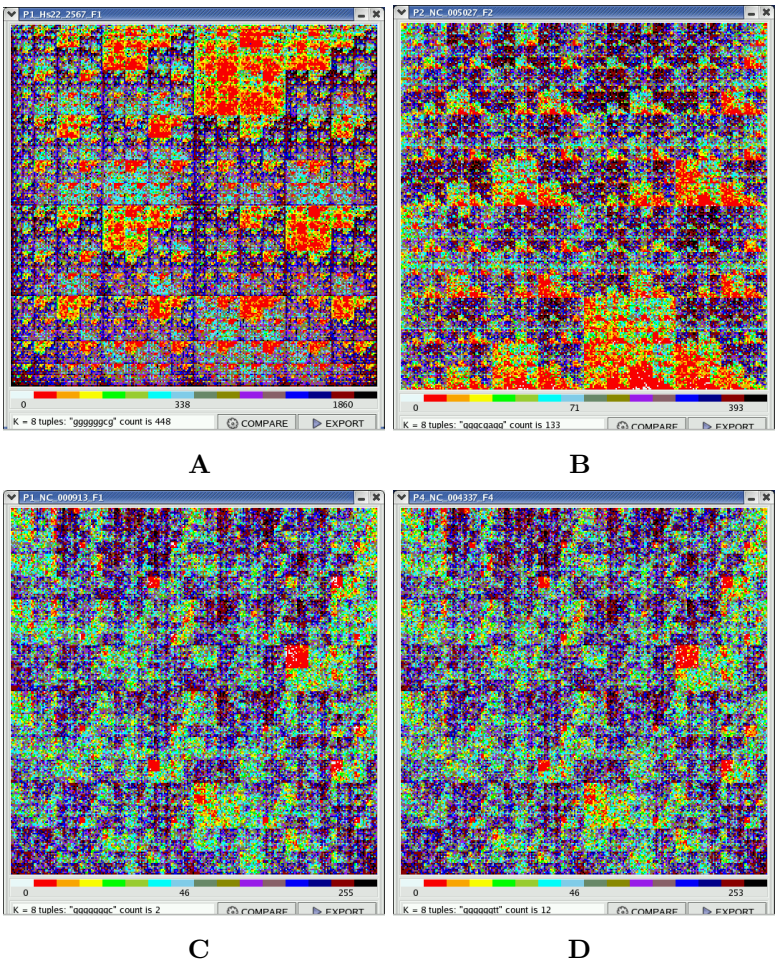


Fig. 1 2D portraits of the human chromosome 22 and the genomes of three bacteria. **A**. Human chromosome 22; **B**. *Pirellula* genome; **C**. *Escherichia coli* genome; **D**. *Shigella flexneri* 2a genome.

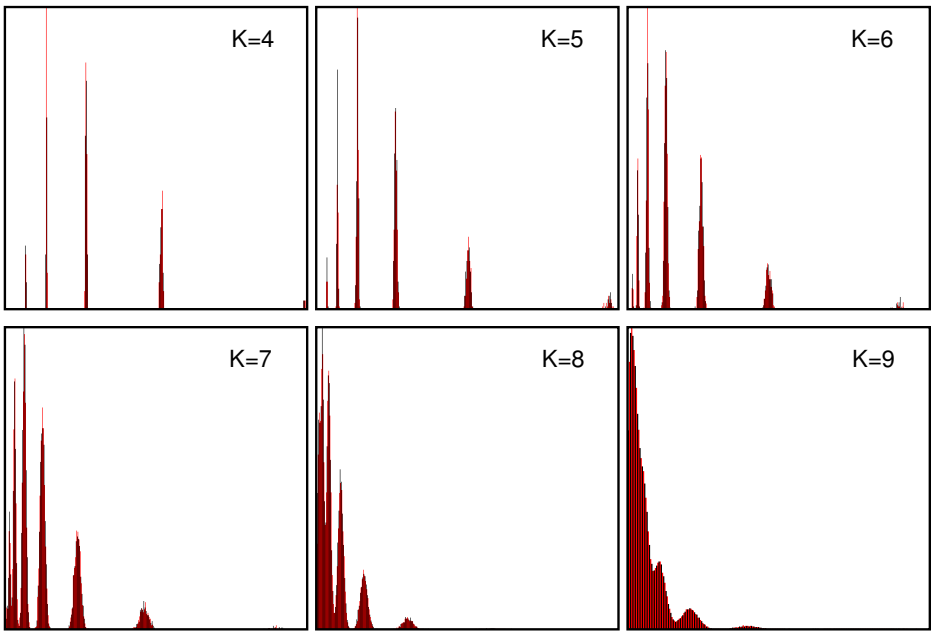


Fig. 2 Fine structure in the 1D histograms of *Mycobacterium tuberculosis* complete genome for $K = 4$ to 9.

the frequency distribution, these counters are allocated in a $2^K \times 2^K$ square matrix and a color code is used to show the range of counting. The allocation of counters is realized by taking the direct product of K copies of the 2×2 matrix (*1*)

$$\begin{pmatrix} G & C \\ A & T \end{pmatrix}.$$

We put G and C in the first row to make the effect of $G+C$ content readily visible. What we obtain is a 2D histogram or a "portrait" of the DNA sequence. Examples of bacterial portraits were given in the above reference and in Figure 1. Some combinatorial problems raised by these portraits have been solved rigorously (*2*). A 1D histogram is constructed by putting the counts along the abscissa from a minimal (may be zero) to a maximal count and the number of string types within a narrow range (a bin) of counts along the ordinate. To provide a reference for comparison, the program can randomize the input sequence, keeping the number of each type of nucleotides unchanged.

The User Graphic Interface and two sample histograms are shown in Figure 3. The program takes one or more DNA sequences in either GenBank or FASTA format as input. The user may form a list of sequences and then work with them to conduct comparative studies. SeeDNA displays 2D as well as 1D histograms of the designated sequence or its reverse-conjugate or both (by concatenating them) using the original input or its randomized counterpart. The string length K can be changed within the range $1 \sim 9$. Using K greater than 9 would extend the picture beyond the screen of most present-day computers. Both the 2D and 1D histograms are interactive. Moving and clicking the cursor will cause the designated string and its count (2D) or the count range and the number of string types whose counts fall in that range (1D) to be displayed.

The 2D histograms of closely related species show strong similarities in K-string composition. This is clearly seen in the two lower portraits of Figure 1 and the portrait of Figure 3C, as *E. coli*, *S. flexneri* and *S. typhi* all belong to the same family Enterobacteriaceae. Therefore, it makes sense to display the difference of counts for each string type. To put the comparison on equal footing, the counting results are normalized to that of 1 Mb. Then the two counts c_1 and c_2 for the same string type are used to calculate $(c_1 - c_2)/(c_1 + c_2)$. The last ratio is displayed using seven colors for the ranges $(-0.01, 0.01)$, $\pm(0.01, 0.1)$, $\pm(0.1, 0.5)$, and $\pm(0.5, 1)$. The string and its counts c_1 and c_2 are shown interactively at the bottom of the graph. This comparison feature is experimental for the time being and the way of showing the difference of "portraits" will be improved as more applications are implemented.

Some advanced options are also provided. For example, the user may change the color code or determine how many times the randomization procedure would be applied to the sequence before it is treated. At the users' choice a screen figure may be exported to a GIF file under a separate name for later manipulation.

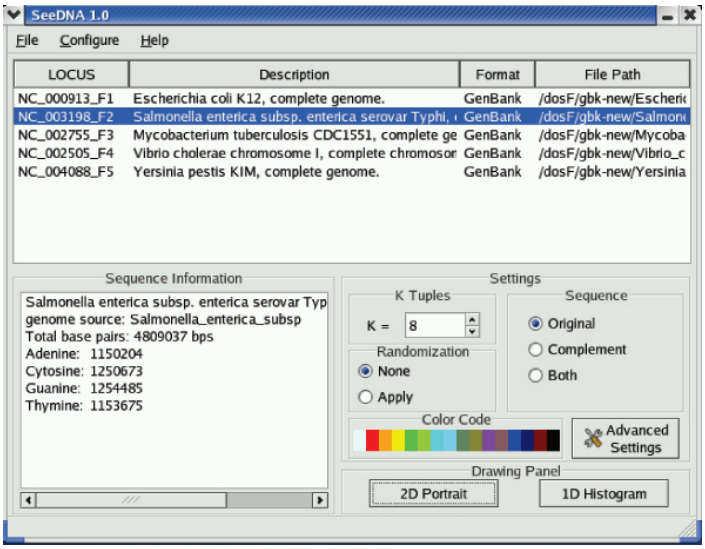
Implementation and Availability

Besides our old UNIX code, a very limited version of the 2D histogram was implemented at the European Bioinformatics Institute (EBI; http://industry.ebi.ac.uk/openBSA/bsa_viewers/) and the National Institute for Standard and Technology (NIST; http://math.nist.gov/~FHunt/GenPatterns/). These implementations did not provide built-in randomization, 1D histogram and comparison of 2D histograms.

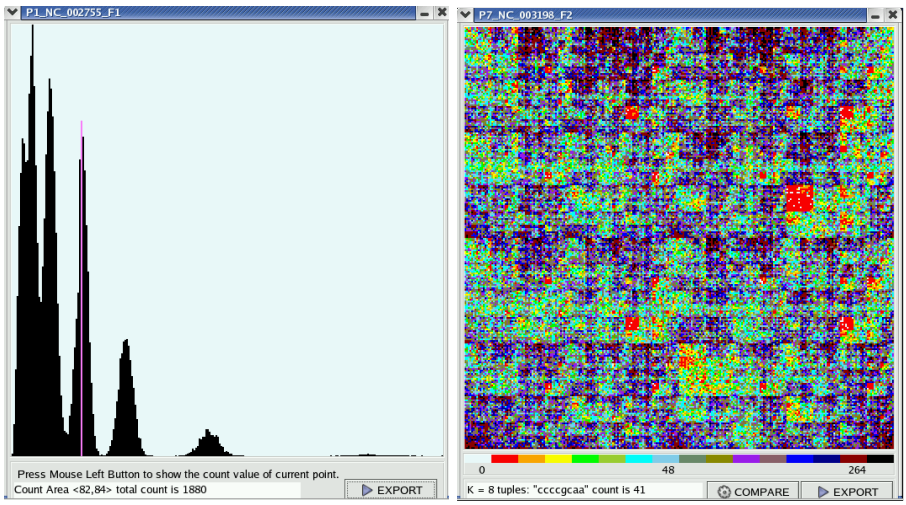
Our full-fledged SeeDNA program is written in C language using the GTK+ graphic package. A user-friendly interface makes the choice or combination of features a matter of clicking on buttons.

The 2D histograms, if shown only in black/white, look similar to the Chaos Game Representation (CGR) of DNA sequence (*8*, *9*). The CHAOS program in the free EMBOSS package (*10*) implements the CGR algorithm. The consistency of the "limiting measure" of CGR and SeeDNA algorithms has been analyzed by Tinó (*11*). However, the SeeDNA realizes the visualization of "density" in one pass, keeping the resolution K fixed. This additional information is displayed by using color codes as an extra dimension. This cannot be done in CGR without changing the algorithm and program. Moreover, due to finite resolution of the computer screen the actual K is out of control and it varies along different directions in the CGR. Therefore, SeeDNA may replace CHAOS entirely with many new features added (1D histogram, randomization, comparison of 2D histograms, *etc.*).

The source code of SeeDNA is freely available under the GNU General Public License at the authors' website (www.itp.ac.cn/~hao/SeeDNA.tar.gz; http://tlife.fudan.edu.cn/SeeDNA.tar.gz). Installation and running information as well as references are included as separate files in the above release package.



A



B **C**

Fig. 3 Sample output of SeeDNA. **A**. the UGI; **B**. 1D histogram of randomized *Mycobacterium tuberculosis* genome; **C**. 2D portrait of *Samonella typhi* genome.

Since the definition of direct product of matrices applies to rectangular matrices as well, the idea of using direct product of matrices to represent K-strings may also be extended to protein sequences. We may define a 4×5 matrix (*12*)

$$X = \begin{pmatrix} A & C & D & E & F \\ G & H & I & K & L \\ M & N & P & Q & R \\ S & T & V & W & Y \end{pmatrix},$$

where the matrix elements are the one-letter abbreviation of the amino acids. However, a similar visualization scheme would only work for $K \leq 4$ if one does not scroll the picture behind the screen. Furthermore, as protein sequences are much shorter than nucleic acids, the highlight of visualization must come from those strings that are present instead of those missing.

References

1. Hao, B.L., *et al.* 2000. Fractals related to long DNA sequences and bacterial complete genomes. *Chaos Solitons Fractals* 11: 825-836.
2. Hao, B.L. 2000. Fractals from genomes—exact solutions of a biology-inspired problem. *Physica A* 282: 225-246.

3. Qi, J., *et al.* 2004. Whole genome prokaryote phylogeny without sequence alignment: a K-string composition approach. *J. Mol. Evol.* 58: 1-11.
4. Hao, B.L. and Qi, J. 2004. Prokaryote phylogeny without sequence alignment: from avoidance signature to composition distance. *J. Bioinform. Comput. Biol.* 2: 1-19.
5. Hsieh, L.C., *et al.* 2003. Minimal model for genome evolution and growth. *Phys. Rev. Lett.* 90: 018101-018104.
6. Chang, C.H., *et al.* 2004. Shannon information in complete genomes. *IEEE Proc. Comput. Syst. Bioinform. Conf. (CSB2004)*: 20-30.
7. Xie, H.M. and Hao, B.L. 2002. Visualization of K-tuple distribution in prokaryote complete genomes and their randomized counterparts. *IEEE Proc. Comput. Syst. Bioinform. Conf. (CSB2002)*: 31-42.
8. Jeffrey, H.J. 1990. Chaos game representation of gene structure. *Nucleic Acids Res.* 18: 2163-2170.
9. Almeida, J.S., *et al.* 2001. Analysis of genomic sequences by chaos game representation. *Bioinformatics* 17: 429-437.
10. Rice, P., *et al.* 2000. EMBOSS: the European Molecular Biology Open Software Suite. *Trends Genet.* 16: 276-277.
11. Tinó, P. 2002. Multifractal property of Hao's geometric representations of DNA sequences. *Physica A* 304: 480-494.
12. Hao, B.L. 2002. "Spatial-temporal" patterns in prokaryote genomes. *Int. J. Bifurcat. Chaos* 12: 2625-2630.