Nucleotides homogeneity within eukaryotes genomes: A comparison of three methods

Eran Elhaik[1], Dan Graur[1], and Kresimir Josic[2]

[1] Department of Biology & Biochemistry, University of Houston, Houston, TX 77204-5001
[2] Department of Mathematics, University of Houston, Houston, TX 77204-3008

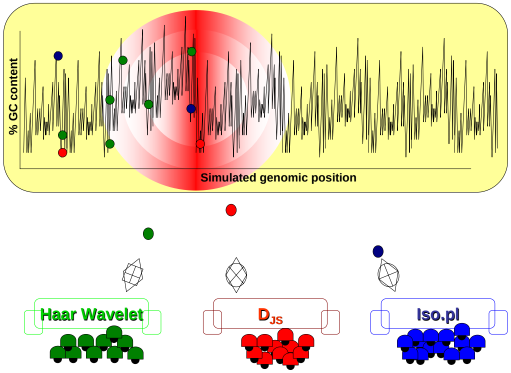
Introduction

The isochore theory, proposed three decades ago (Macaya et al. 1976. *J. Mol. Biol.* 108:237-254), depicts the mammalian genome as a mosaic of long, fairly homogeneous genomic regions called isochores, each with a characteristic GC content. Low-GC and high-GC segments alternate throughout the genome. Detecting isochores requires the use of segmentation algorithms that detect homogeneous sequences within the genome that are distinguishable in their GC content from adjacent sequences. Many such segmentation algorithms have been proposed in the literature, but none of those methods detected "isochores" in a reproducible manner. These failures raised doubts about the very existence of isochores to doubt the very existence of isochores (Lander et al. 2001. *Nature* 409:860–921; Haring and Kypr, 2001 *Mol. Biol. Rep.* 28:9-17; Cohen et al. 2005. *Mol. Biol. Evol.* 22:1260-1272).

Our purpose was to develop a benchmark to test the efficiency of three different segmentation algorithms to detect isochores. For this purpose we , we generated DNA sequences, with GC distribution differing between two predetermined borders. The GC distribution was generated from:
(1) a normal distribution with the added noise drawn from a uniform distribution, and (2) a uniform distribution with the added noise drawn from a normal distribution.

Three algorithms were evaluated by their ability to detect the borders with an accuracy of 100 bp and by their false positive detection of borders. The three algorithms are: (1) the Janson-Shannon divergence (D_{JS}) algorithm, a binary recursive segmentation procedure (Bernaola-Galvan et al. 1996. *Phys. Rev.* 53E:5181-5189), (2) a variant of D_{JS} algorithm (Iso.pl), a binary recursive segmentation procedure that differs from the original algorithm in its stopping criterion (Li et al. 2002. *Comput. Chem.* 25:491-510), and (3) our new DNA segmentation algorithm that is based on the one-dimension Haar decomposition function (Goupillaud et al. 1984. *Geoexploration.* 23:85-102). In this method, we define the mean GC content of any two adjacent nucleotides as a "signal" that may assume three values: 0, 0.5, and 1. Our Haar wavelet algorithm analyzes the signal and detects segments with similar GC fluctuations for any determined threshold value.

Our findings indicate that all three algorithms had a lower false negative detection when the GC distribution was drawn from a normal distribution with a standard deviation of > 0.3. When the standard deviation was 0.1 the false negative detections ranged between 20% and 50% and the false positive detections ranged between 0% and 40%.



Conclusions

Our benchmark simulation revealed the efficiency the of three segmentation algorithms. We showed that the D_{JS} and Haar wavelet algorithms work best in both normal and uniform distributions, however they both generate many false positives errors when the GC concentration was drawn from a uniform distribution. The Iso.pl algorithm had the lowest false positive errors in all the tests. It performed well when the GC concentration was drawn from a normal distribution, but poorly when the GC concentration was drawn from a uniform distribution.

Further work

The final purpose of the benchmark simulation is to lay a ground rule for the use of the best segmentation algorithm for the right genome sequence. For that purpose, we suggest to add more genome elements for the simulated sequences, such as outliers, GC islands, TE, and other elements. We also suggest developing an algorithm to analyze the genome GC distributions composition and suggest which algorithms are best to be used.

Methods

Generated sequences

The sequences were generated from two distributions: (1) a normal distribution with a mean that ranged between 0.1 and 0.9, a standard deviation that ranged between 0.1 and 0.9, and a noise that was drawn from a uniform distribution with a mean that ranged between 0.1 and 0.5, (2) a uniform distribution with a mean that ranged between 0.1 and 0.5, a noise that was drawn from a normal distribution with a mean that ranged between 0.1 and 0.9, a standard deviation that ranged between 0.1 and 0.9. The difference between adjacent subsegments of GC content was set to be higher than 0.5%. Results were considered "hits" when they were less than 100 bp distant from the border, and "miss" when they were not. Every test was repeated 10 times.

D_{JS}

In this algorithm, the sequence is recursively segmented by maximizing the difference in GC content between adjacent subsequences. The process of segmentation is terminated when the difference in GC content between two neighboring segments is no longer statistically significant *(Cohen et al. 2005. Mol. Biol. Evol. 22:1260-72)*.

Iso.pl

In this algorithm, the sequence is recursively segmented by maximizing the difference in GC content between adjacent subsequences. The stopping criteria use the Bayesian Information Criterion. The segmentation stops when $2ND_{JS} > log(N)K$, when *N* is the sequence size and *K* is the number of free parameters *(Li et al 2002. Comput. Chem. 25:491-510)*.

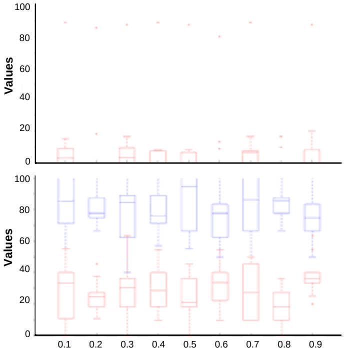
Haar wavelet

Our algorithm has four steps: (1) calculate the signal values. i.e., the mean GC content of any two adjacent nucleotides, (2) decompose the signal with a one-dimension Haar function, (3) reconstruct the signal using a threshold. Signal values higher than (1 – threshold) are assigned a value of 1, and (4) segment the reconstructed signal using a binary recursive algorithm (manuscript in preparation).

Figures legends

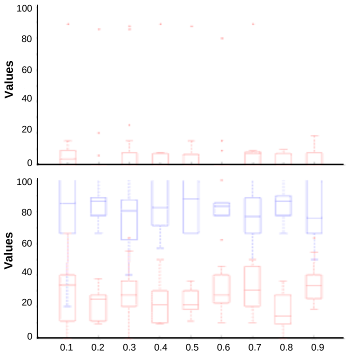
Evaluation of the segmentation results for all three algorithms. On the x-axis, the mean GC content for (top) normal distribution with a standard deviation of 0.1 and (bottom) uniform distribution with a noise that was drawn from a normal distribution with a standard deviation of 0.1 . The red boxplots show the false positive results (miss) and the blue boxplots show the true positive results (hits).

Haar Wavelet



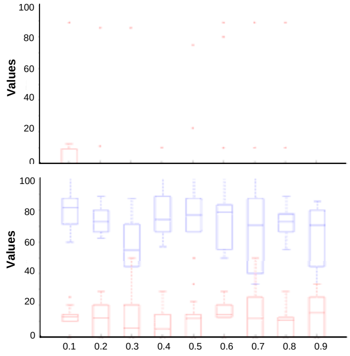
When the GC distribution was normal, Haar wavelet had a 99% mean hits score and a 3% false positive score. When the GC distribution was uniform, Haar wavelet had a 80% mean hits score and a 32% false positive score. Haar wavelet gained the highest hits scores in the normal distribution with standard deviation of 0.5.

D_{JS}



When the GC distribution was normal, D_{JS} had a 99% mean hits score and a 3% false positive score (miss). When the GC distribution was uniform, D_{JS} had 88% mean hits score and 32% false positive score. D_{JS} gained the highest hits scores in the normal distribution with standard deviation of 0.1, 0.3, 0.7, and 0.9.

Iso.pl



When the GC distribution was normal, Iso.pl had a 98% mean hits score and a 1% false positive score. When the GC distribution was uniform, Iso.pl had a 65% mean hits score and a 8% false positive score. Iso.pl gained the highest false positive scores in the normal distribution with standard deviation of 0.1 – 0.9.