Identifying compositionally homogeneous and nonhomogeneous domains within the human genome using a novel segmentation algorithm

Eran Elhaik1,*, Dan Graur2, Krešimir Josić3 and Giddy Landan2

1McKusick - Nathans Institute of Genetic Medicine, Johns Hopkins University School of Medicine, Baltimore, MD, 21205, 2Department of Biology and Biochemistry, University of Houston, Houston, TX, 77204-5001 and 3Department of Mathematics, University of Houston, Houston, TX, 77204-3008, USA

Received April 20, 2010; Revised May 23, 2010; Accepted May 26, 2010

ABSTRACT

It has been suggested that the mammalian genome is composed mainly of long compositionally homogeneous domains. Such domains are frequently identified using recursive segmentation algorithms based on the Jensen–Shannon divergence. However, a common difficulty with such methods is deciding when to halt the recursive partitioning and what criteria to use in deciding whether a detected boundary between two segments is real or not. We demonstrate that commonly used halting criteria are intrinsically biased, and propose IsoPlotter, a parameter-free segmentation algorithm that overcomes such biases by using a simple dynamic halting criterion and tests the homogeneity of the inferred domains. IsoPlotter was compared with an alternative segmentation algorithm, DJS, using two sets of simulated genomic sequences. Our results show that IsoPlotter was able to infer both long and short compositionally homogeneous domains with low GC content dispersion, whereas DJS failed to identify short compositionally homogeneous domains and sequences with low compositional dispersion. By segmenting the human genome with IsoPlotter, we found that one-third of the genome is composed of compositionally nonhomogeneous domains and the remaining is a mixture of many short compositionally homogeneous domains and relatively few long ones.

INTRODUCTION

Mammalian guanine–cytosine (GC) content is known to have a complex internal compositional organization (1). For example, the human genome is known to contain long compositionally homogeneous domains whose GC contents range from ~33% to ~60%. Evidence for the non-uniformity and non-randomness of nucleotide composition was first discovered several decades ago when bulk DNA sequences that had been randomly sheared into long fragments were separated by their base composition using thermal melting and gradient centrifugation (2). The fragments were grouped into a small number of classes distinguished by their buoyant densities that correlate with GC content. These findings led Bernardi and co-workers (3–5) to propose the isochore theory for the structure of homeotherm genomes.

The isochore theory posits that mammalian genomes are mosaics of isochores: long (>300 kb), relatively homogeneous regions, each with a typical GC content (6). The theory further posits that these regions are separated by boundaries of sharp GC content changes (7) and that all isochores can be divided into a handful of compositional domain classes or families (8).

With the advent of mammalian genomics, it became feasible to attempt to detect isochores using a segmentation algorithm with the genomic sequence as the sole input. Indeed, many methods were proposed to detect isochores by partitioning genomic sequences into compositional domains according to predefined criteria (9–14).

In a previous study, we proposed a benchmark for testing the abilities of segmentation algorithms to identify compositionally homogeneous regions and isochores within genomic sequences (15). Surprisingly, the various segmentation methods, such as sliding-window (16,17), recursive segmentation (10,12,18) and least-square segmentation (14,19), yielded inconsistent results. Recursive segmentation algorithms based on the Jensen–Shannon divergence (20), such as DJS (10,11), significantly outperformed all other segmentation algorithms.
Recursive segmentation methods find cutting points (known also as segmentation points or partition points) that maximize the difference in base compositions between adjacent subsequences. Because there is at least one position in the sequence that maximizes the difference in base composition of any two subsequences, the recursive partitioning process can (in theory) continue until the number of segments equals the number of nucleotides (11). Therefore, a central part of all segmentation algorithms is the criterion used to halt the segmentation when the differences between adjacent segments become ‘insignificant’.

Criteria that are too stringent or relaxed can lead to under- and over-segmentation, respectively (15). For instance, the halting criterion of the DJS algorithm works as follows: first, the threshold is set by partitioning multiple homogeneous sequences of a certain size and composition and obtaining the maximal Jensen–Shannon divergence statistic \( D_{JS} \). Next, a fixed threshold for the \( D_{JS} \) entropy statistic is set \textit{a priori} by establishing a minimum statistical significance level for the Jensen–Shannon entropy below which segmentation cannot take place (11). Finally, the \( D_{JS} \) statistic is calculated over all possible cutting points along the candidate sequence, and the maximal \( D_{JS} \) value is compared to the threshold value. The candidate sequence is partitioned in the position of the maximal \( D_{JS} \) if it is higher than the threshold. Segmentation continues recursively for segments for which \( D_{JS} \) exceeds the given threshold. Unfortunately, any choice of an \textit{a priori} threshold affects the lengths of the inferred domains (15). For example, lowering the threshold to improve short domain inference decreases the ability to detect large domains. Although this problem has been reported previously (18,21,22), no solution has been proposed. The problem is caused by the initial choice of sequences used to determine the threshold and, therefore, cannot be solved by changing their properties.

To overcome this difficulty, we introduce IsoPlotter, an improved recursive segmentation algorithm that employs a ‘dynamic threshold’ that takes into account the composition and size of each segment. IsoPlotter calculates the \( D_{JS} \) statistic over all possible cutting points and compares its maximum to a dynamic threshold. In contrast to the standard DJS algorithm, the threshold is not determined \textit{a priori}, but separately for each segment to be partitioned. The length and standard deviation of GC content averaged over small windows along the segment are used to determine the dynamic threshold. If the maximum \( D_{JS} \) statistic exceeds this dynamically determined threshold, the segment is partitioned and segmentation continues recursively.

In this study, we tried to avoid some of the semantic confusion regarding isochores by simulating two sets of genomic sequences containing predetermined compositionally homogeneous domains, each separated from adjacent domains by sharp changes in GC content. These simulated domains should be considered isochores and may be used to compare the detection capability of DJS (10,11) and IsoPlotter. The first set consisted of tri-domain sequences with a short central-domain flanked by two long domains. In the second set, multi-domain sequences were generated with varying standard deviations of GC content for each domain. In the last part of our study, we studied the compositional architecture of the human genome and compared the results obtained by using both IsoPlotter and DJS.

**METHODS**

The DJS and IsoPlotter segmentation algorithms

The DJS is a binary, recursive segmentation algorithm that splits a DNA sequence by finding a point that maximizes the difference in GC content between adjacent subsequences. The resulting subsequences are then recursively segmented until a halting condition is satisfied (11).

Briefly, a sequence of length \( L \), GC content \( F_{GC} \) and AT content \( F_{AT} = 1 - F_{GC} \), is divided into two continuous subsequences \((s = \text{left}, \text{right})\) of length \( l_s \), GC content \( f_{s,\text{GC}} \), and AT content \( f_{s,\text{AT}} = 1 - f_{s,\text{GC}} \). These subsequences are split at the point \( i \) that maximizes the entropic measure \( D_{JS} \) defined as the difference between the overall entropy of the sequence \( H_{\text{tot}} \) and the weighted sum of the entropies of both subsequences \( H_{\text{left}} \) and \( H_{\text{right}} \):

\[
D_{JS}(i) = H_{\text{tot}} - \left( \frac{l_{\text{left}}}{L} H_{\text{left}} + \frac{l_{\text{right}}}{L} H_{\text{right}} \right) \quad (1 < i < L)
\]

where the entropy of the right and left subsequences is

\[
H_s = -f_{s,\text{GC}} \log_2 f_{s,\text{GC}} - f_{s,\text{AT}} \log_2 f_{s,\text{AT}} \quad (0 \leq f \leq 1)
\]

and the entropy of the whole sequence is

\[
H_{\text{tot}} = -F_{\text{GC}} \log_2 F_{\text{GC}} - F_{\text{AT}} \log_2 F_{\text{AT}} \quad (0 \leq F \leq 1)
\]

The maximal \( D_{JS} \) value is denoted by \( \hat{D}_{JS} = \max D_{JS}(i) \) as the point of maximum difference between the left and right subsequences. The process of segmentation is terminated when \( \hat{D}_{JS} \) is smaller than a predetermined threshold. We used a threshold of \( 5.8 \times 10^{-5} \) (Dr. Tal Dagan, University of Düsseldorf, personal communication). Instead of comparing \( \hat{D}_{JS} \) to a predetermined threshold, the IsoPlotter algorithm compares it to a dynamic threshold computed from the length and composition of the candidate subsequence.

The algorithms were implemented in Matlab 7.5 and are available at http://code.google.com/p/isoplotter/.

**Simulated data analyses**

To test the capabilities of IsoPlotter and DJS to detect compositionally homogeneous domains, we performed two analyses. First, we simulated 1000 tri-domain sequences, each composed of two 10-Mb-long compositionally homogeneous domains at the 5’ and 3’ ends and a short, compositionally homogeneous, central-domain of length 100 kb. Domain mean GC contents were chosen from a normal distribution with a mean of 50% and a standard deviation of 5%. Negative values of the mean GC contents or those higher than 100% were unlikely
given the low standard deviation. We repeated the simulation with central-domains of 300 kb and 1 Mb in size.

In the second analysis, we simulated 1000 multi-domain sequences composed of 13 equally sized compositionally homogeneous domains, each 10 kb in length. Domain mean GC contents were chosen from a normal distribution with a mean of 50% and a standard deviation of 1%. The domain GC content standard deviation (σGC) ranged from 10−2 to 10−1 with the between-domain variation and the within-domain variation increasing accordingly.

We used simulated sequences comprising of compositionally homogeneous domains within predefined borders separated from adjacent domains by sharp changes in GC content. These simulated domains should be recognized as isochores. Domains were composed of 32 bp non-overlapping windows to reduce computation time without compromising accuracy. The GC content GCwindow and standard deviation σGCwindow of each window i were calculated from a uniform distribution and its full-width standard deviation as

\[
GC_{\text{window}}(i) = \left( GC_{\text{domain}} - \frac{\sum σ_{GC_{\text{window}}}}{n} \right) + σ_{GC_{\text{window}}}
\]

(4)

\[
σ_{GC_{\text{window}}}(i) = σ_{GC} \times \sqrt{12 \times R_{\text{uniform}}}
\]

(5)

where GCdomain and σGC are the domain GC content and standard deviation, respectively, n is the number of windows in a domain and Runiform is a random variable drawn from a uniform distribution between 0 and 1; for example, a domain of size 160 bp with a mean GC content of 54% and a GC content standard deviation of 5% composed of five 32-bp windows with mean GC contents of 49%, 58%, 45%, 60% and 58%. The segmentation algorithms were applied to the resulting sequences of GC frequencies.

In our analyses, we chose to work on short windows of 32 bp rather than on single nucleotides to save computation time without sacrificing accuracy (10,15). To exclude a possibility of a bias due to window size, we repeated the second analysis without using windows.

After sequences were partitioned using IsoPlotter and DJJS we tested the performance of the two algorithms by computing domain inferences, defined as the number of domains whose borders were identified within <1000 bp or a distance of <5% of their size, whichever is smaller (15). To evaluate the segmentation results, we used two statistics: sensitivity and precision. Sensitivity is the proportion of correctly inferred domains out of all predetermined domains. Precision quantifies the probability for positive prediction of the algorithm, i.e., the proportion of correctly inferred domains out of all reported domains. For example, if a sequence composed of 10 compositionally homogeneous domains was partitioned by an algorithm that inferred 50 domains but only 25 of them were correctly inferred, then the algorithm sensitivity would be 25% and its precision would be 50%. Sensitivity and precision quantify an algorithm’s accuracy and prediction power, respectively. To test whether the differences between the algorithms are significant, we used the one-tailed Wilcoxon rank-sum test with α = 0.01 (23) and the false discovery rate (FDR) correction for multiple tests (24). All reported results were obtained with a minimum domain length of 3 kb for both IsoPlotter and DJJS, consistent with the literature (6,8,25).

Segmentation of the human genome

The human genome assembly (build 36) was obtained from the NCBI FTP website ftp://ftp.ncbi.nlm.nih.gov/genomes/. Each chromosome was divided into non-overlapping windows of 32 bp in length, and their GC content was calculated.

The genome was partitioned by using both IsoPlotter and DJJS with a minimum domain length of 3 kb. Because the actual genome segmentation is unknown, the algorithms could not be evaluated for accuracy. Only their results can be compared.

In order to compare the output of the two algorithms, we divided the detected domains into nine groups. The cutoffs (lc) that define the boundaries between the groups were chosen as 3 kb, 10 kb, 50 kb, 100 kb, 200 kb, 300 kb, 500 kb, 1 Mb and 10 Mb. For each group, the ‘genome coverage’ was calculated by dividing the sum of the domain lengths by the genome length. The homogeneity of the domains was assessed using a homogeneity test (see below). The domain lengths and the proportion of compositionally homogeneous domains were compared between the two algorithms using a one-tailed t-test with α = 0.05 (23).

Homogeneity test

Inferred domains were classified into two types, compositionally homogeneous and nonhomogeneous, based on their homogeneity relative to the chromosome (Figure 1). We used the F-test to compare the GC content variance of a domain with that of the sequence on which it resides (26). The GC content for each 32-bp non-overlapping window was calculated for the domain in question and for the entire sequence. Because the F-test assumes the data are normally distributed, we followed Cohen et al. (10) and applied the arcsine-root
transformation to the GC content values of the windows within each domain (and sequence) before calculating the variance.

A one-tailed F-test with a null hypothesis $H_0$: $\sigma^2_{\text{domain}} \geq \sigma^2_{\text{sequence}}$, and an alternative hypothesis, $H_1$: $\sigma^2_{\text{domain}} < \sigma^2_{\text{sequence}}$, were applied with $n_1 - 1$ and $n_2 - 1$ degrees of freedom, where $n_1$ and $n_2$ are the numbers of windows in the domain and in the sequence, respectively. If the variance over a domain turned out to be significantly lower ($P < 0.05$) than that of the corresponding sequence, then the domain was considered homogeneous compared to the sequence. We improved the procedure proposed by Cohen et al. (10) by adjusting for multiple comparisons using the FDR correction (24).

RESULTS

Modeling the dynamic threshold, $D_t$

Ideally, the segmentation halting threshold should be calculated analytically from the distribution of the $D_{JS}$ entropy statistic (11). Because the theoretical distribution of the $D_{JS}$ entropy is unknown, the threshold had to be obtained empirically. This was previously done by simulating uniform (homogeneous) regions of a certain length and standard deviation of the GC content ($\sigma_{GC}$), obtaining the maximal $D_{JS}$ entropy statistic ($\hat{D}_{JS}$) for every sequence, calculating the cumulative distribution of $D_{JS}$, and choosing a threshold value corresponding to some type I error rate (10,11). For example, Cohen et al. (10) obtained the threshold using 100,000 sequences of size 1 Mb and $\sigma_{GC}$ of 1%. This practice leads to biases in the length and GC variability of inferred domains.

To demonstrate the relationship between $\hat{D}_{JS}$ and the sequence length, we generated a random sequence of length 1 Mb with a GC content of 50% and GC content standard deviation $\sigma_{GC}$ of 1%. We then calculated $D_{JS}$ for the entire sequence and subsequences of lengths 100 kb and 10 kb (Figure 2). The resulting $\hat{D}_{JS}$ values show a 10-fold increase with every 10-fold decrease in sequence length. A similar relationship exists between $\hat{D}_{JS}$ and the standard deviation $\sigma_{GC}$ of the sequence. No relationship was found between $\hat{D}_{JS}$ and the sequence GC content (data not shown).

Reducing this bias in IsoPlotter required modeling the dependencies between a segment length and its standard deviation $\sigma_{GC}$ on the one hand, and the $\hat{D}_{JS}$ statistic on the other hand. We generated 10,000 sequences for each of 13 length parameters $L$ ranging from 1 kb to 1 Mb and for each of five GC content standard deviation parameters $\sigma_{GC}$ ranging from 1% to 10%. For each simulation setting, we calculated $\hat{D}_{JS}$ by allowing IsoPlotter to partition the sequence once and obtained the threshold ($D_t$) from the top 0.01% percentile of the cumulative $D_{JS}$ distribution (Figure 3). A log-scale plot of $D_t$ as a function of sequence length and variability reveals a near perfect linear relationship (Figure 4). A log-scale linear regression on the length ($L$) and GC content dispersion ($\sigma_{GC}$) fits the empirical data extremely well ($r^2 = 0.995, P < 10^{-16}$) with the following coefficients:

$$\ln D_t = -0.97 \ln L + 0.7 \ln \sigma_{GC} + 4.42$$  \hspace{1cm} (6)

The threshold $D_t$ depends on the length and GC content dispersion for each subsequence. We therefore refer to it is a ‘dynamic threshold’.

Comparing segmentation results for simulated sequences

To gauge domain homogeneity in the simulated sets, we compared the variance in GC content within each domain
to that of the sequence on which it resides (see the ‘Homogeneity test’ section). Domains in all the sequences were constructed to be homogeneous, and the two algorithms were expected to detect them.

We first tested the abilities of IsoPlotter and DJS to detect a short compositionally homogeneous central-domain (100 kb, 300 kb and 1 Mb) within two large compositionally homogeneous domains (10 Mb). Inferences of central-domain borders were divided into three types: no detection, partial detection (one border detected) and full detection (Table 1). IsoPlotter identified at least one domain border in all sequences and both domain borders in more than 86% of the sequences. By contrast, DJS missed both domain borders in 15–20% of the sequences, identified one domain border in 46–75% of the sequences, and fully identified the two domain borders in <40% of the sequences. IsoPlotter inferences were unaffected by central-domain size, while DJS inference and sensitivity were highest for longer domains. Interestingly, both algorithms had high precision (97–100%) despite of DJS’s poor performances. IsoPlotter performances were significantly higher than those obtained by DJS (Wilcoxon rank-sum test, p < 0.01).

Next, we tested the abilities of IsoPlotter and DJS to infer domains with differing GC content standard deviations and fixed lengths. An example of a simulated sequence is shown in Figure 5a. The mean sensitivity for each domain $\sigma_{\text{GC}}$ is shown in Figure 5b. For these data, IsoPlotter sensitivity was 98%, with a precision approaching 100%. The DJS sensitivity was 19% and strongly dependent on $\sigma_{\text{GC}}$, while the precision was near 100%. IsoPlotter significantly outperformed DJS for every domain variation tested (Wilcoxon rank-sum test, $p < 0.01$). Results were robust to the choice of window size that composed the compositionally homogeneous domains.

### Segmentation of the human genome

One of the main premises of isochore theory is that nearly the entire genome of homeotherms (warm-blooded animals) consists of compositionally homogeneous domains that exceed 300-kb in length (4–6,27). Figure 6 presents the genome coverage as a function of domain length. IsoPlotter ‘isochoric’ domains ($\geq$300 kb) cover <30% of the human genome, while DJS’s ‘isochoric’ domains cover <50% of the genome.

**Table 1. Proportion of central-domain detection inferred by IsoPlotter and DJS**

<table>
<thead>
<tr>
<th>Segmentation algorithm</th>
<th>Central-domain Sensitivity</th>
<th>Precision</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Size</td>
<td>No detection (%)</td>
</tr>
<tr>
<td>IsoPlotter</td>
<td>100 kb</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>300 kb</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>1 Mb</td>
<td>0</td>
</tr>
<tr>
<td>DJS</td>
<td>100 kb</td>
<td>20</td>
</tr>
<tr>
<td></td>
<td>300 kb</td>
<td>20</td>
</tr>
<tr>
<td></td>
<td>1 Mb</td>
<td>15</td>
</tr>
</tbody>
</table>

![Figure 4](https://example.com/f4.png)

**Figure 4.** The threshold $D_t$ values obtained from simulated sequences with various lengths and GC standard deviations are plotted against each sequence length $L$ and variability $\sigma_{\text{GC}}$ on a log-log scale. The threshold chosen by Cohen et al. (10) is marked by a large dot.

![Figure 5](https://example.com/f5.png)

**Figure 5.** (a) An illustration of the GC content spatial distribution along a 130-kb simulated sequence that contains 13 equally sized domains (dotted bars). Domain mean GC content is marked by black lines. (b) The mean sensitivities (with error bars) of IsoPlotter (green) and DJS (red) are shown for every domain (dotted bars).
Classifying domains according to their lengths shows that overall IsoPlotter inferred a higher proportion of short to medium-size domains than DJS (10–200 kb), while DJS inferred more long domains (≥300 kb), including very long domains (≥10 Mb). Overall, domains inferred by DJS are significantly longer than those inferred by IsoPlotter (t-test, α < 0.05) illustrating the bias of DJS toward long domains.

Classifying domains inferred by both algorithms according to their homogeneity shows that for each length-cutoff group 70% of IsoPlotter inferences were homogeneous compared to only 53–67% of DJS. Moreover, the ratio of homogeneous/nonhomogeneous inferences for IsoPlotter is significantly higher than that of DJS (t-test, α < 0.05) for all length cutoffs (Table 2) except \( l_c = 10 \text{ Mb} \) (see Supplementary Tables S1 and S2). These results suggest that the low proportion of compositionally homogeneous domains detected by DJS is an outcome of the segmentation quality and does not depend on domain lengths.

In terms of coverage, 19% of the human genome is composed of 820 compositionally homogeneous domains longer than 300 kb (Table 2). Furthermore, these domains constitute only 1% of the total number of compositionally homogeneous domains (117,391). Therefore, restricting genome compositional studies to ‘isochoric’ domains, as is commonly done (e.g., 17, 28, 29), necessitates ignoring 99% of all domains that cover over 80% of the genome.

### Compositional domain ideograms of the human genome

Using IsoPlotter inferences, three compositional domain ideograms of the human genome were drawn (Figures 7a–c). The ideograms illustrate our major findings, and allow us to compare compositional patterns among chromosomes. The first ideogram shows that compositionally homogeneous domains cover between 62% (chromosome 4) and 78% (chromosome 11) of the chromosomes (Figure 7a). Dividing compositionally homogeneous domains to long (≥300 kb) and short domains reveals that ‘isochoric domains’ are heterogeneously distributed along chromosomes covering between 5% (chromosome 19) and 29% (chromosomes 5) of the chromosomes (Figure 7b). By contrast, short compositionally homogeneous domains cover between 38% (chromosome 4) and 72% (chromosome 22) of the chromosomes. ‘Isochoric domains’ can further be classified into low GC domains ranging from 20% to 40% (574 domains) and high GC domains ranging from 40% to 60% (245 domains) with a single rich GC domain (61%) in chromosome 16 (Figure 7c). Thus, 70% of all ‘isochoric domains’ are AT rich. We find no evidence for the five-family division proposed by Bernardi et al. (4).

### DISCUSSION

The study of genome composition has been hampered for decades by conflicting results and uncertain methodology. Schmidt and Frishman (30) proposed to address this problem by using a consensus method based on the results of several algorithms (such as 12, 17, 19). This approach is problematic because combining correct inferences with incorrect ones only serves to dilute the truth. Instead, we proposed a benchmark to test the performance of different segmentation algorithms (15). We showed that recursive segmentation algorithms based on the Jensen–Shannon divergence (20) performed significantly better than all other segmentation algorithms.

However, even recursive segmentation algorithms can perform poorly because of their use of fixed thresholds as halting criteria. Here, we show that the \( D_{JS} \) entropy measure is correlated with sequence length and the standard deviation of its GC content (\( \sigma_{GC} \)), and that

![Figure 6](https://academic.oup.com/nar/article-abstract/38/15/e158/2409738)

**Figure 6.** Cumulative spatial coverage for human genome using all domains inferred by IsoPlotter (blue) and DJS (red). Seventy percent of IsoPlotter inferences were homogeneous compared to 67% for DJS.

Table 2. Comparison of IsoPlotter and DJS segment results for the human genome as a function of length cutoff (\( l_c \))

<table>
<thead>
<tr>
<th>Length cutoff (( l_c ))</th>
<th>Domains longer than ( l_c )</th>
<th>Compositionally homogeneous domains longer than ( l_c )</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Number (%)</td>
<td>Genome coverage (%)</td>
</tr>
<tr>
<td>IsoPlotter</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3 kb</td>
<td>117,391 (100) 100</td>
<td>82,186 (70) 70</td>
</tr>
<tr>
<td>10 kb</td>
<td>45,466 (39) 86</td>
<td>31,867 (27) 60</td>
</tr>
<tr>
<td>50 kb</td>
<td>9,661 (8) 60</td>
<td>6,880 (6) 42</td>
</tr>
<tr>
<td>100 kb</td>
<td>4,691 (4) 48</td>
<td>3,265 (3) 33</td>
</tr>
<tr>
<td>200 kb</td>
<td>2,081 (2) 35</td>
<td>1,431 (1) 24</td>
</tr>
<tr>
<td>300 kb</td>
<td>1,199 (1) 27</td>
<td>820 (1) 19</td>
</tr>
<tr>
<td>500 kb</td>
<td>561 (1) 19</td>
<td>386 (1) 13</td>
</tr>
<tr>
<td>1 Mb</td>
<td>163 (&lt;1) 9</td>
<td>113 (&lt;1) 7</td>
</tr>
<tr>
<td>10 Mb</td>
<td>0 (0) 0</td>
<td>0 (0) 0</td>
</tr>
<tr>
<td>DJS</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3 kb</td>
<td>87,794 (100) 100</td>
<td>58,554 (67) 64</td>
</tr>
<tr>
<td>10 kb</td>
<td>33,470 (38) 89</td>
<td>22,365 (26) 57</td>
</tr>
<tr>
<td>50 kb</td>
<td>6,986 (8) 70</td>
<td>4,600 (5) 44</td>
</tr>
<tr>
<td>100 kb</td>
<td>3,337 (4) 61</td>
<td>2,180 (3) 38</td>
</tr>
<tr>
<td>200 kb</td>
<td>1,598 (2) 53</td>
<td>1,019 (1) 33</td>
</tr>
<tr>
<td>500 kb</td>
<td>1,042 (1) 48</td>
<td>667 (1) 30</td>
</tr>
<tr>
<td>1 Mb</td>
<td>572 (&lt;1) 35</td>
<td>208 (&lt;1) 21</td>
</tr>
<tr>
<td>10 Mb</td>
<td>15 (&lt;1) 9</td>
<td>8 (&lt;1) 5</td>
</tr>
</tbody>
</table>
Figure 7. Ideograms of compositional domains inferred by IsoPlotter and mapped to chromosomes. Using data mining approach, the ideograms uncover the compositional patterns of long homogeneous domains (‘isochoric’) in three layers (from top to bottom): (a) compositionally homogeneous domains and nonhomogeneous domains; (b) long compositionally homogeneous domains (≥300 kb), short compositionally homogeneous domains (<300 kb) and compositionally nonhomogeneous domains; and (c) long compositionally homogeneous domains (≥300 kb) color coded by their mean GC content and all other domains (short compositionally homogeneous and nonhomogeneous domains).
this dependence introduces biases in the segmentation. Consequently, recursive segmentation algorithms employing a fixed threshold, such as $D_{JS}$, cannot be expected to perform well on sequences containing isochores of different lengths and compositions. To overcome this problem, we modeled these relationships to create a dynamic-threshold algorithm.

The log-linear relation between the threshold value, $D$, and sequence length, $L$ [Equation (6)] is not surprising. Let us consider an idealized model of genomic sequence as a series of Bernoulli trials with $P$ the probability of a $G$ or a $C$ nucleotide. The mean GC content of a sequence of length $N$ is, therefore, a random variable that approximately follows a normal distribution with mean $P$ and standard deviation $\sqrt{P(1-P)/N}$. The standard deviation of the mean GC content decreases with sequence length.

Since the entropy $H_{\text{tot}}$ is defined as a function of the mean GC content, $H_{\text{tot}}$ is itself a random variable. In the case of a Bernoulli sequence, $H_{\text{tot}}$ is approximately normally distributed as it is a function of mean GC content, and the standard deviation of the mean GC content is relatively small. Furthermore, the $D_{JS}$ statistic is a function of $H_{\text{tot}}$. Hence, $D_{JS}$ is also a random variable with variance that decreases with sequence length. Although real genomic sequences cannot be expected to be modeled by perfect Bernoulli trials, the decrease of the standard deviation of the mean GC content with the increase of sequence size is expected due to the central limit theorem (31), if the correlations between nucleotide content along the sequence are not too strong.

The main idea proposed here is expected to hold despite the mild long-range correlations described in genomic sequences (11,32–35), which may increase the standard deviation. These correlations reduce domain homogeneity and produce a higher proportion of false positive inferences (type I errors) because of the fluctuations in nucleotide composition. Moreover, since recursive segmentation algorithms compare the composition of subsequences residing on adjacent subsequence to each other, the resulting domains are not necessarily homogeneous compared to the whole sequence. For these reasons, it is essential to assess the homogeneity of the inferred domains using a homogeneity test.

The high sensitivity and precision obtained using the dynamic threshold were demonstrated in two analyses in which IsoPlotter successfully detected compositionally homogeneous domains of various lengths and GC content standard deviations $\sigma_{GC}$. IsoPlotter’s high sensitivity and precision are a direct result of its dynamic threshold. Such results are not achievable with other segmentation algorithms. Moreover, these results suggest that segmentation approaches that filter or concatenate ‘short’ segments to eliminate GC content fluctuations (e.g., 12,28) may be misleading.

A homogeneity test was applied to the domains inferred in the human genome. A classification of these domains revealed, surprisingly, that the majority of the genome (70%) consisted of compositionally homogeneous domains, but only 19% of them can be considered isochores in the traditional sense (8). We note that IsoPlotter was not artificially tuned to detect domains of a particular length (e.g., 28) and, unlike $D_{JS}$, it is not biased toward short domains (Table 1).

Cohen et al. (10) used the $D_{JS}$ algorithm to partition the human genome and then classified the inferred domains according to their lengths using a length cutoff, $l_c$. A comparison of our results with those obtained by Cohen et al. (10) reveals two major differences. First, Cohen et al. (10) reported that the proportion of domains found to be ‘putatively homogeneous’ out of all inferred domains was dependent on the domain length cutoff, $l_c$. That is, the longer domain was more likely considered the ‘putatively homogeneous’ domain. In contrast, our results show that the proportion of compositionally homogeneous domains out of all inferred domains slightly decreased with the increase in length cutoff, $l_c$. Second, they reported a possible bias in their homogeneity test, which qualified almost all long domains (>50 kb) as ‘putatively homogeneous’. However, we did not observe this bias using our homogeneity test. The difference in our results can be explained by our improved homogeneity test that corrects for multiple comparisons.

Segmenting the human genome with IsoPlotter revealed a new genomic compositional architecture consisting of a mixture of compositionally nonhomogeneous domains with numerous short compositionally homogeneous domains and relatively few long ones (Figures 7a–c). A preliminary analysis of eight species indicates that this salient description holds for other mammalian genomes (Elhaik E. and Graur D., unpublished data). To understand how such structures emerged in an evolutionary perspective, a comparative analysis using different genomes is currently underway. Using IsoPlotter, we now have the ability to use the same analytical tool on genomes that were heretofore considered too heterogeneous to be partitioned, such as the yeast genome (36).

SUPPLEMENTARY DATA
Supplementary Data are available at NAR Online.

FUNDING
Funding for open access charge: The National Science Foundation (grant number DBI-0543342 to D.G.); University of Houston (Small Grant award number 1098048 to D.G.); US National Library of Medicine (grant number LM010009-01 to D.G. and G.L.); National Science Foundation (grant numbers DMS-0604429, DMS-0817649); Texas Advanced Research Program and the Advanced Technology Program (Project no. 003652-0024-2007 to K.J.).

Conflict of interest statement. None declared.

REFERENCES


