A Comparison of two Compositional Segmentation Algorithms for Genomic Sequences

AP Research

May 17th, 2021

Word Count: 4,325

ABSTRACT

Of the various segmentation algorithms created to predict the locations of compositionally homogeneous domains within genomic sequences, two of the most widely used algorithms are IsoPlotter (Elhaik et al. 2010b) and IsoSegmenter (Cozzi et al. 2015). However, these two algorithms yield significantly different predictions, and no study to date has thoroughly examined their differences. Here, I present a detailed comparison of the IsoPlotter and IsoSegmenter algorithms, using a library of simulated random genomic sequences as a benchmark to test algorithm performance and accuracy. Each simulated genomic sequence consisted of multiple simulated compositional domains which were assigned distinct guanine-cytosine (GC) percentages based on the isochore families model (Bernardi 2000). Of the 2,000 simulated sequences generated in this study, 1,100 consisted of domains assigned equal lengths, and the other 900 sequences contained domains assigned variable lengths based on a power-law distribution. My results show that IsoPlotter significantly outperforms IsoSegmenter under a variety of test scenarios, and that IsoSegmenter consistently predicts the existence of large (>200,000bp) domains regardless of underlying genomic architecture. However, there is room for both algorithms to be improved upon, such as IsoPlotter's tendency to underpredict compositional domain sizes.

INTRODUCTION

In 1976, Gabriel Macaya and colleagues centrifuged mouse and human DNA in a CsCl density gradient, and after analyzing the banding patterns produced by this density centrifugation technique, these researchers put forth a theory regarding the compositional structures of genomes. Their conclusions outlined the basis of what would eventually be named the isochore theory: that the major banding components existed in large units, estimated to be greater than 300kb (300,000 base pairs) in length in the mouse genome (Macaya et al. 1976). Later, these large chromosomal sequences were found to contain homogeneous amounts of G + C (guanine + cytosine) content within the regions, and were eventually named "isochores", meaning similar regions (Cuny et al. 1981). Over the years, a general definition for isochores has emerged:

1. Isochores are long, generally homogeneous chromosomal DNA segments greater than 300kb in length (Macaya et al. 1976, Bernardi 1993, Bernardi 2000, Bernardi 2001)
2. Isochores are composed of a homogeneous GC content (Bernardi 2000)
3. Isochores are organized into distinct isochore families defined by unique GC content ranges. (Bernardi et al. 1985, Bernardi 1993, Bernardi 2000, Bernardi 2001) For instance, the human genome was said to consist of 5 isochore families: 2 "GC-poor" families L1 and L2, and 3 "GC-rich" families H1, H2, and H3. The 5 families had GC contents of <37%, 37%–42%, 42%–47%, 47%–52%, and >52%, respectively. (Bernardi 2000)
4. Boundaries between isochores were marked by sharp changes in GC content (Fukagawa et al. 1994).

This definition of isochores can be summarized in the following sentence: isochores are "...long (> 300 kb), compositionally homogeneous DNA segments that can be subdivided into a small number of families characterized by different GC levels...." (Bernardi 2000). The creators of the isochore theory suggested that vertebrate genomes consist of "mosaics" of different isochores with different GC contents (Bernardi 2000).

Testing isochore theory with genomic sequencing data:

Many research papers published before genome sequencing became widely available used the GC content of the third codon positions in protein-coding genes to approximate the total GC content of a region (Bernardi et al. 1985, Michaux et al. 2001, Belle et al. 2002). The main drawback of this method is that protein-coding genes were estimated to only comprise 1.5% of the entire human genome, causing such third-codon estimates not representative of the majority of the genome at large (Lander et al. 2001). However, the publication of complete genome sequences created an opportunity to more thoroughly assess isochore theory (Lander et al. 2001). While the human genome did show long range variation in GC content, an analysis of 300kb windows resulted in a standard deviation that was too large to be consistent with a model of homogeneous isochores (Lander et al. 2001). Put directly by the researchers of Lander et. al (2001), "...isochores do not appear to merit the prefix 'iso'", and "...it is likely to be worth redefining the concept…". The publishing of the cow genome introduced additional evidence against isochore theory. An analysis of the distribution of bovine GC content found that various

compositionally homogeneous segments do exist in the cow genome; however, only 3% of the homogeneous GC domains identified had a length greater than the 300kb isochore cutoff (Bovine Genome Sequencing and Analysis Consortium 2009).

In all, genomic analysis did find compositionally homogeneous domains, but very few of these domains met the narrow criteria put forward by isochore theory. This led to an alternative hypothesis, the compositionally homogeneous domain theory, which theorized that mammalian genomes are composed of various homogeneous domains that vary in size from 1,000 base pairs to 10,000,000 base pairs in length, and that small domains should not be ignored, as they normally are under isochore theory (Elhaik et al. 2014).

Nonetheless, proponents of isochore theory disputed many of these findings. Bernardi (2001) took issue with Lander et al.'s definition of isochores as "strictly homogeneous", and others objected to the use of the Binomial Test to test for isochore homogeneity (Li et al. 2003). These conflicting interpretations that arose after the study of sequencing data incited the modern debate over isochores.

The current debate over isochores:

In 2005, Cohen et al. analyzed the isochore composition of the human genome using a segmentation algorithm, and found that segments greater than 300kb only covered 41% of the human genome and made up only 3.9% of the total segments generated. They also noticed discrepancies between the 5-family isochore model (Bernardi 2000) and the models generated from their segmentation data, leading them to conclude overall that "...isochore theory has reached the limit of its usefulness…" Further studies of mammalian genomes using another segmentation algorithm called IsoPlotter (Elhaik et al. 2010b) supported these results (Elhaik et al. 2014). It was found that homogeneous sequences greater than 300kb (isochores) covered less than 28% of total mammalian genomes and constituted less than 2% of all domains identified in this study (Elhaik et al. 2014).

However, Clay and Bernardi (2005) rebutted the claims of Cohen et al. (2005), arguing that the segmentation algorithm over-segmented putative isochores into small segments that were later discarded. They also accused Cohen et al.'s cutoff of 300kb for isochoric segments as being too narrow, despite the numerous papers shown earlier that explicitly stated that isochoric segments are greater than 300kb in length. Additionally, Costantini et al. (2006) used a sliding-window algorithm rather than a segmentation algorithm, and yielded completely opposite results from Cohen et al. (2005). These researchers identified nearly 3,000 isochoric sequences that covered 85% of the genome, and plots of the distribution of GC sequences in these identified isochores depicted five distinct isochore families (Costantini et al. 2006).

After looking through the literature, these differences can be traced to two main variables: strictness of the definition of isochores, and algorithm style used. Proponents of isochore theory tend to favor a flexible, ever-widening definition of isochores (Bernardi 2001, Clay and Bernardi 2005), whereas those seeking to disprove isochore theory opt to follow the exact definition of isochores according to the literature (Cohen et al. 2005, Elhaik et al. 2010b, Elhaik et al. 2014). In terms of algorithms, proponents of isochore theory have consistently relied upon sliding-window algorithms, while opponents of isochore theory started with sliding-window algorithms (Lander et al. 2001) but later shifted towards recursive segmentation algorithms (Cohen et al. 2005, Elhaik et al. 2010b, Elhaik et al. 2014).

Algorithms that Identify Compositionally Homogeneous Domains:

As mentioned above, a key contributor to the isochore debate is the difference in algorithm choice. Many different types of algorithms have been developed by researchers to try and delineate the boundaries of homogeneous domains (Elhaik et al. 2010a), but the most commonly used algorithms are the sliding-window method and the segmentation method. The sliding-window method involves splitting the genome into multiple small fragments of a chosen length known as windows, and computing the GC content of each individual window (Costantini et al. 2006). On the other hand, segmentation algorithms spit the entire genome into two halves at a point that maximize the GC content difference between both halves, and then each half is recursively split into even more halves until a stopping point is reached (Cohen et al. 2005).

A. **IsoSegmenter**: One prominent sliding-window algorithm is IsoSegmenter (Cozzi et al. 2015), which partitions the genome into non-overlapping 100kb length windows.(Costantini et al. 2006). Secondly, GC content is computed at each window, and based on GC content the window is assigned to one of the 5 isochore families (Bernardi 2000). Finally, nearby, compositionally similar windows are merged to form long, continuous segments.

B. **IsoPlotter:** A relevant example of a segmentation algorithm is the IsoPlotter algorithm (Elhaik et al. 2010b). IsoPlotter works exactly like the general segmentation algorithm described above, except for some important changes in stopping condition. Earlier segmentation algorithms stopped segmentation when the Jensen-Shannon Divergence statistic, which measured the difference in GC content on both sides of the divergence point, fell below a manually inputted threshold (Cohen et al. 2005). However, the IsoPlotter found the manual input of a stopping threshold problematic, since the wrong threshold choice could lead to too much or too little segmentation. Instead, IsoPlotter automatically calculates the right stopping threshold based upon the length and GC content of the sequence, eliminating any need for human input (Elhaik et al. 2010b).

Gap in Research:

When both IsoPlotter and IsoSegmenter were run on human chromosome 1, both algorithms produced drastically different results (Cozzi et al. 2015). Constantini et al. (2006) identified numerous isochores in mammalian genomes using IsoSegmenter, whereas Elhaik et al. (2014) used IsoPlotter data to conclude that isochores do not exist in mammalian genomes. Given how IsoPlotter produced multiple small domains and IsoSegmenter produced larger isochore-length segments (Figure 1), it is evident that differences in algorithm design are a major cause behind the conflict over isochore theory. To identify which of these two algorithms is producing the more accurate results, the accuracy, precision, and other key performance metrics of IsoPlotter and IsoSegmenter need to be compared side-by-side. Although the outputs of IsoSegmenter and IsoPlotter have been compared in the past (Cozzi et al. 2015), no paper thus far has compared the actual performance and accuracy of these two algorithms, and this creates a large gap in the research.

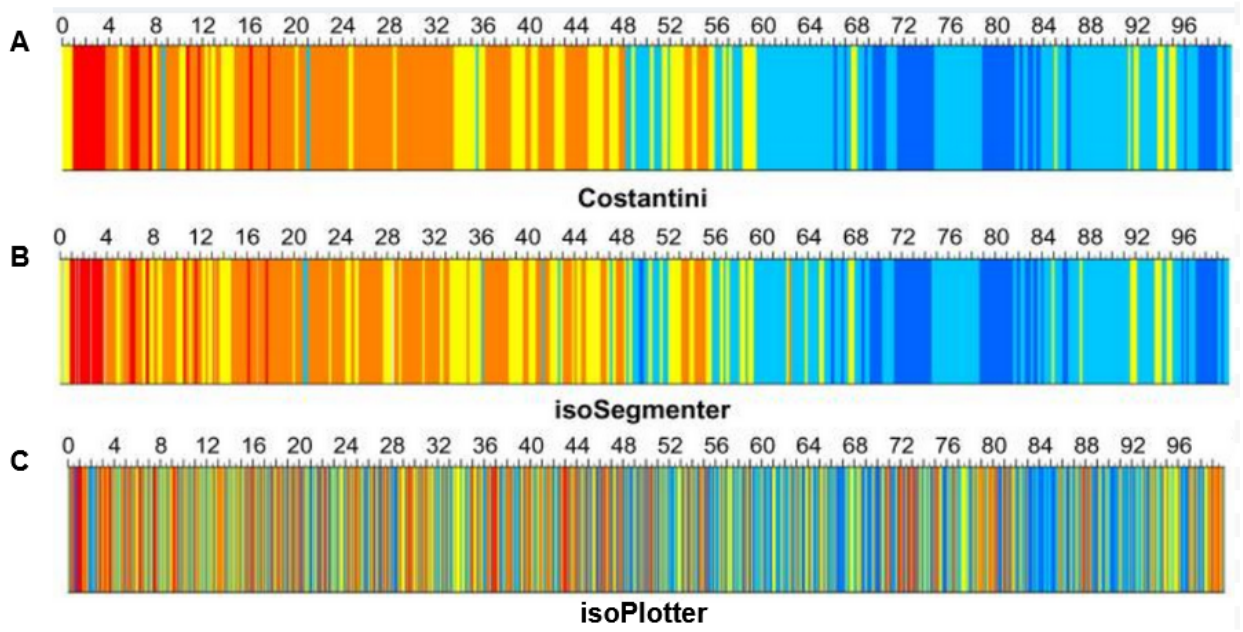


Figure 1: Isochore domain assignment for the first 100mb of human chromosome 1 by Costantini et al.'s (2006) initial sliding-window algorithm, IsoSegmenter, and IsoPlotter (Cozzi et al. 2015). The scale of the diagram is in MB (1 mb = 10^6 base pairs). The colors deep blue, light blue, yellow, orange, and red represent isochores belonging to families L1, L2, H1, H2, and H3, respectively.

Purpose and Hypothesis:

Given the conflicting results delivered by IsoPlotter and IsoSegmenter, the purpose of this study is to identify which of these two algorithms is producing more accurate, precise, efficient, and reliable results that better represent the compositional profile of a DNA sequence. My analysis of these two algorithms will focus on the comparison of key performance metrics such as sensitivity, specificity, precision, deviation from expected results, and more. Based on how IsoPlotter has completely eliminated the need for subjective user input (Elhaik et al. 2010b), and based on previous benchmarks that demonstrated the superiority of segmentation algorithms over sliding window algorithms (Elhaik et al. 2010a), I hypothesize that IsoPlotter will prove to be more effective than IsoSegmenter in compositional domain analysis.

MATERIALS AND METHODS

For my research methodology, I will compare the predictions of isoPlotter and isoSegmenter on two different types of simulated sequences. A similar approach was used by Elhaik et al. (2010a) to test a variety of isochore algorithms. Simulated sequences will be used since they enable easy, accurate, and reproducible calculation of the sensitivity, positive predictive value, and Jaccard similarity coefficient for isoPlotter and isoSegmenter. All code and data used in this study are freely accessible in an online repository[1] for replication by others.

Segments of Equal Length:

For the first analysis section (equal-length), I will generate simulated genomic sequences with a length of 1mb (1,000,000bp). Each simulated sequence will be evenly divided into varying amounts of equal-length simulated domains. The number of divisions/domains per segment will range between 1, 2, 4, 5, 10, 20, 25, 40, 50, 80, and 100. If, for example, a particular simulated sequence is divided into 10 domains, it will be composed of ten 100kb (100,000bp) equal-length domains. Each domain will be randomly assigned to one of the five isochore families (L1, L2, H1, H2, and H3) with frequencies of 22.8%, 33.2%, 22.7%, 11.2%, and 3.01%, respectively. (Schmidt and Frishman 2008). The precise GC content of each domain will be randomly selected from the GC content range of the assigned family, and adjacent domains will always be assigned to different isochore families (Schmidt and Frishman 2008).. Simulated sequences were compiled and converted into FASTA format using the Biopython package (Cock et al. 2009).

Segments of Varying Length:

For the second analysis section, I will better simulate the real-world genomic sequences by varying domain lengths within each sequence. According to Cohen et al. (2005), the lengths of homogeneous GC domains in the human genome follow a power-law distribution with an exponent of $\alpha = -2.55$. The properties of power-law distributions are summarized in Table 1:

[1] https://anonymous.4open.science/r/Segmentation-Algorithm-Comparison-78D2

Table 1: A summary of the properties of power-law distributions

Probability Density Function (PDF)	$p(x) = \frac{\alpha-1}{x_{min}}(\frac{x}{x_{min}})^{-\alpha}$
Cumulative Distribution Function (CDF)	$P(x) = 1 - (\frac{x}{x_{min}})^{1-\alpha}$
Inverse Cumulative Distribution Function (inverse CDF)	$P^{-1}(r) = x_{min}(1 - r)^{\frac{1}{1-\alpha}}$
Expected Value (μ, mean)	$\mu_x = x_{min}(\frac{1-\alpha}{2-\alpha}); \alpha > 2$

Note: x_{min} = lower bound, α = power law exponent (or scaling parameter), r = random uniformly distributed variable in the range [0,1). The PDF, CDF, and inverse CDF equations were sourced from Clauset et al. (2009) The expected value was derived by the author using the general definition of expected value for a continuous distribution: $\int_{xmin}^{\infty} xp(x)dx$

For this analysis section, I will generate simulated chromosome segments with a length of 5mb (5,000,000bp), and choices for the number of domains per segment will range between 2, 4, 6, 8, 10, 20, 30, 40, and 50. Using the *transformation method* described by Clauset et al. (2009), the lengths of each simulated domain will be randomly sampled from a power-law distribution with x_{min} = 10,000bp and α = -2.55 using an inverse CDF function (Table 1). Afterwards, the domain lengths will be normalized by their total to scale the overall sequence length to 5,000,000bp.

With the parameters x_{min} = 10,000bp and α = -2.55, the mean domain length generated from the function will be around 28,182bp (Table 1). The overall 5 mb segment length and the options for number of domains take this into account, as they were chosen to prevent the length normalization step from compressing domains to lengths smaller than 10,000bp.

Scoring of Predicted Domains for Simulated Sequences:

After generating these two types of simulated sequences, I will run both isoPlotter and isoSegmenter on each of the simulated sequences to predict compositional domains. The main advantage of using simulated sequences is that, unlike real data, the exact start and stop point of the isochore domains is predetermined. This enables me to directly compute the accuracy of the

domain predictions by isoPlotter and isoSegmenter. Both isoPlotter and isoSegmenter assign domains by predicting the *boundaries* between isochore domains.

Thus, a predicted domain will be deemed a *true positive (TP)* match if both predicted boundaries match the actual boundary positions of the domain within ±5% of the predicted domain's size, following the precedent established by Elhaik et al. (2010a) for benchmark testing. A *false positive (FP)* predicted domain would be a domain predicted by an algorithm for which one or both predicted boundaries do not match with ground truth boundaries. A *false negative (FN)* domain would be a ground truth domain which was not successfully predicted by the algorithm (Figure 2).

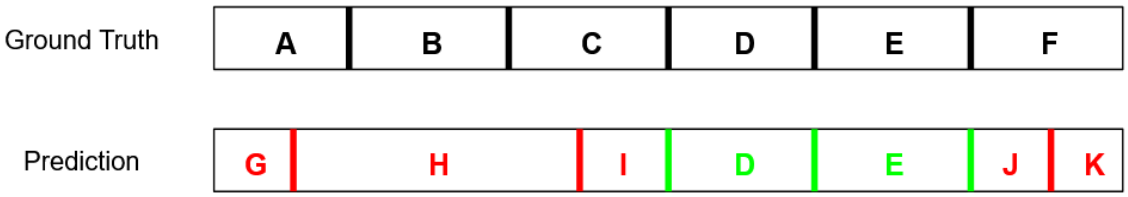


Figure 2: An example of how domain predictions will be scored. Boundaries in green indicate correct matches, and domains in green indicate a correctly predicted domain. Boundaries and domains in red indicate incorrect predictions. Notice that a domain is only considered a correct match if both boundaries align with the ground truth. In this example, the *true positives (TP)* are D and E, the *false positives (FP)* are G, H, I, J, K, and the *false negatives (FN)* are A, B, C, F.

Data Analysis:

After scoring the predicted domains, I will compare the performance of isoPlotter and isoSegmenter on the simulated sequences using the metrics of sensitivity, positive predictive value (PPV, or precision rate), and the Jaccard Index. The methods for calculating these three metrics are provided below:

$$Sensitivity = \frac{TP}{TP + FN}$$

$$Positive\ Predictive\ Value\ (PPV) = \frac{TP}{TP + FP}$$

$$Jaccard\ Index = \frac{TP}{TP + FP + FN}$$

where TP, FP, and FN stand for true positive, false positive, and false negative results, respectively, which will be determined using the process from the preceding subsection. All plots for analysis were generated using the ggplot2 R package (Wickham 2016).

RESULTS

Each category of equal-length and variable-length domain sequences was simulated for 100 trials, yielding a total of 1,100 sequences with equal domain lengths and 900 sequences with variable domain lengths. Figure 3 demonstrates sample GC profiles for an equal length and a variable length sequence generated in this study.

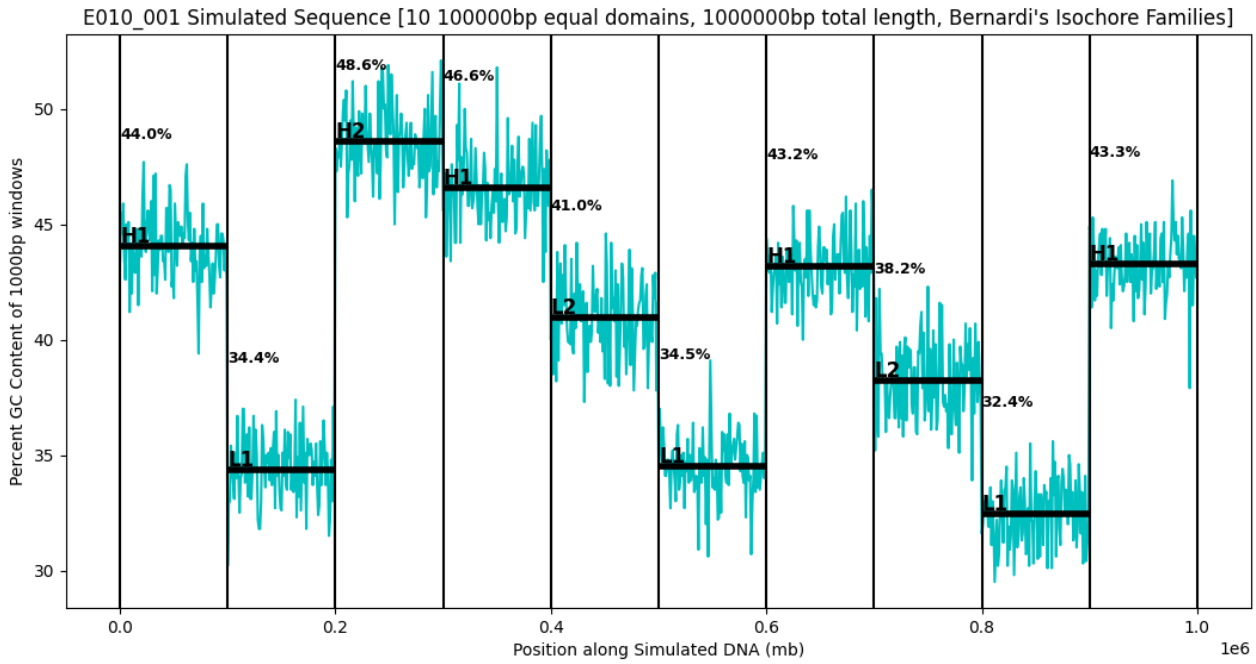
The 2,000 unique simulated sequences generated were segmented using both isoPlotter and isoSegmenter, and the predictions of each algorithm were compared to the ground truth and scored using the guidelines established in the Materials and Methods section**.** A summary of the overall performance of both algorithms is presented in Table 2, which contains the average sensitivity, precision, and Jaccard Index for each algorithm across all simulated sequences.

Table 2: Overall Performance of isoPlotter and isoSegmenter on the simulated sequence dataset

	Mean for isoPlotter	**Median for isoPlotter**	**Mean for isoSegmenter**	**Median for isoSegmenter**	**p-value**
Sensitivity	0.5362	0.5200	0.2714	0.0250	$p < 2.2x10^{-16}$***
Precision (PPV)	0.3122	0.2903	0.2854	0.0714	$p < 2.2x10^{-16}$***
Jaccard Index	0.2600	0.2321	0.2394	0.0196	$p < 2.2x10^{-16}$***

Table 2: p-values were computed using a two-sample Wilcoxon Rank Sum test, using H_0: equal medians for isoPlotter and isoSegmenter, and H_a: difference in medians between isoPlotter and isoSegmenter. Significance Codes: *$p \leq 0.05$, **$p \leq 0.01$, ***$p \leq 0.001$

A



B

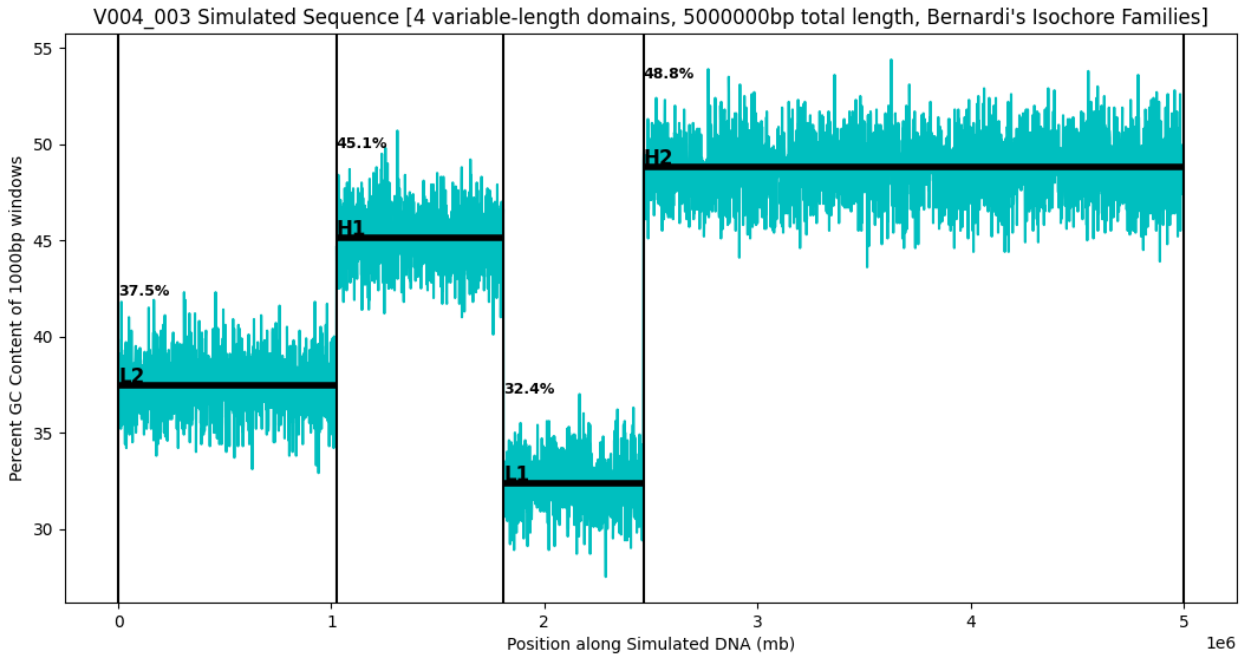


Figure 3: The GC content profile of a 1mb simulated sequence with 10 equal-length domains **(a)** and a 5mb simulated sequence with 4 variable-length domains **(b)**. Isochore family and mean GC percentage are indicated for each simulated domain. Both GC Profiles were generated using 1,000bp sliding windows.

It is important to note that the medians of all three performance metrics for isoSegmenter are much closer to zero than the mean, implying that isoSegmenter's mean sensitivity, precision and Jaccard Index are skewed upwards by outlier instances when isoSegmenter had much higher performance than average. The medians for isoPlotter's performance, however, were much closer to the means, indicating a more consistent performance lacking skew. To account for this upwards skew in isoSegmenter's performance, the Wilcoxon Test was used to compare the performance metrics of isoPlotter and isoSegmenter rather than the standard two-mean t-test. Overall, IsoPlotter had a higher median sensitivity, precision rate, and Jaccard Index compared to isoSegmenter, and the differences in these performance metrics were all statistically significant at a cutoff of $\alpha = 0.05$.

Influence of Domain Size on Performance of isoPlotter and isoSegmenter:

Figures 4 and 5 provide a closer look at the effect of domain size on the performance of isoPlotter and isoSegmenter. By design, an increase in the number of domains per sequence for both equal-length and variable-length simulated sequences is associated with a decrease in average domain length.

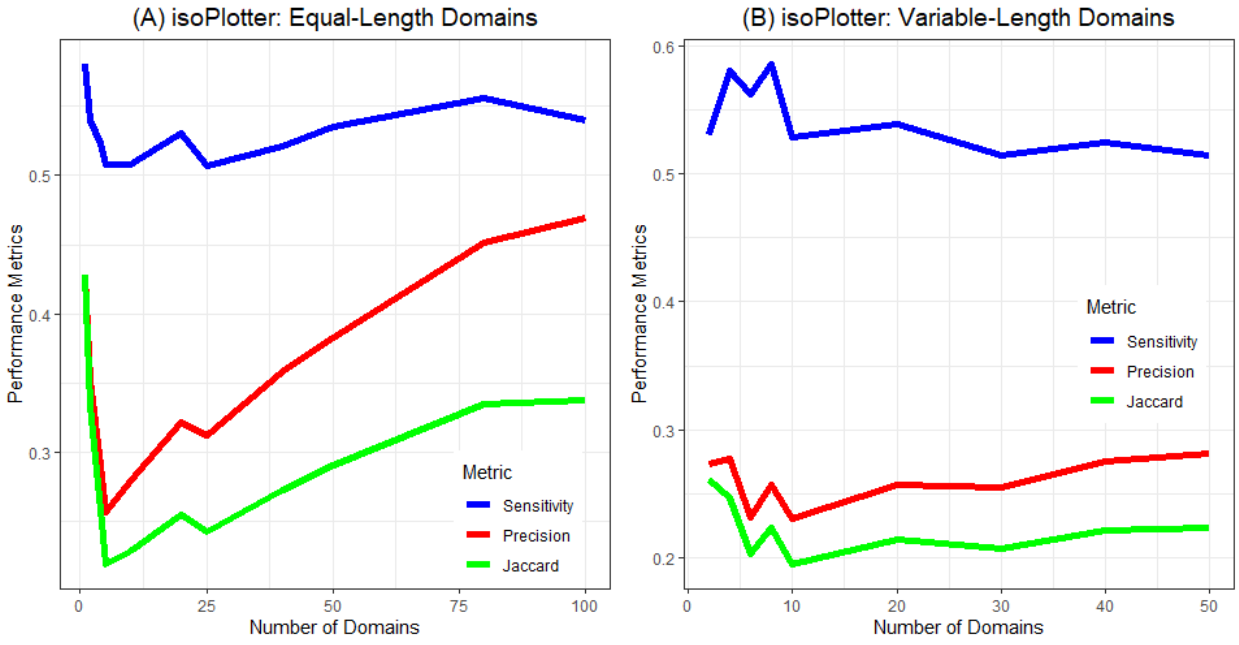


Figure 4: Trends in the performance of isoPlotter across varied numbers of domains per sequence for equal-length sequences **(a)** and variable-length sequences **(b)**.

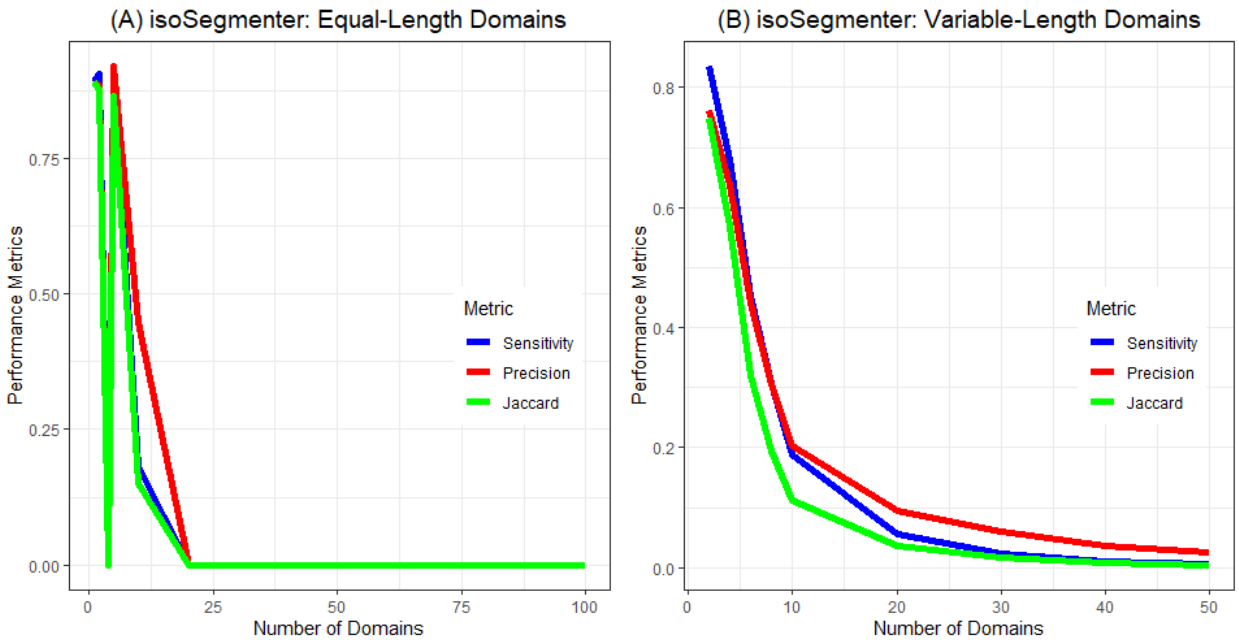


Figure 5: Trends in the performance of isoSegmenter across varied domain lengths for equal-length sequences **(a)** and variable-length sequences **(b)**.

With very large domain sizes, isoSegmenter had high performance, and correctly predicted nearly all the domains in this size range (Figure 5). However, as the number of domains increased beyond 20 (which is equivalent to an equal-length domain size of 50,000bp), all performance metrics for isoSegmenter rapidly declined to zero. On the other hand, isoPlotter displayed more consistent performance levels across all sequences tested (Figure 4). For isoPlotter, sensitivity remained between 50% and 60% across all types of equal and variable domain sequences. Interestingly, isoPlotter's precision rate and Jaccard Index increased as the number of domains increased (especially in the equal-length sequences), and both of these performance metrics for isoPlotter were higher on average among equal-length sequences compared to variable-length ones.

To better understand these performance changes across domain lengths, the average predicted domain length for both isoPlotter and isoSegmenter on the equal-length dataset was plotted against true domain size (Figure 6). Ideally, the predicted lengths would exactly match the ground truth, leading all data points to lie on a 45 degree line through the origin. Deviations of data points away from this reference line reflect prediction inaccuracies.

In Figure 6, it is apparent that the average prediction length of isoSegmenter does not decrease below 200,000 base pairs, even while segmenting sequences with domain lengths as small as 10,000 bp. In contrast, isoPlotter has a tendency to over-segment sequences into smaller domains than observed, and its predictions are closer to the reference 45-degree line at smaller domain sizes. However, unlike isoSegmenter, isoPlotter's average prediction lengths are correlated to true domain size, since the data points from isoPlotter follow the contour of the reference line. Figure 7 further displays this tendency of isoSegmenter to predict domains that are 200,000bp long. It also shows a more uniform distribution for isoPlotter's prediction lengths, with the exception of larger domain sizes, which isoPlotter predicted much more infrequently.

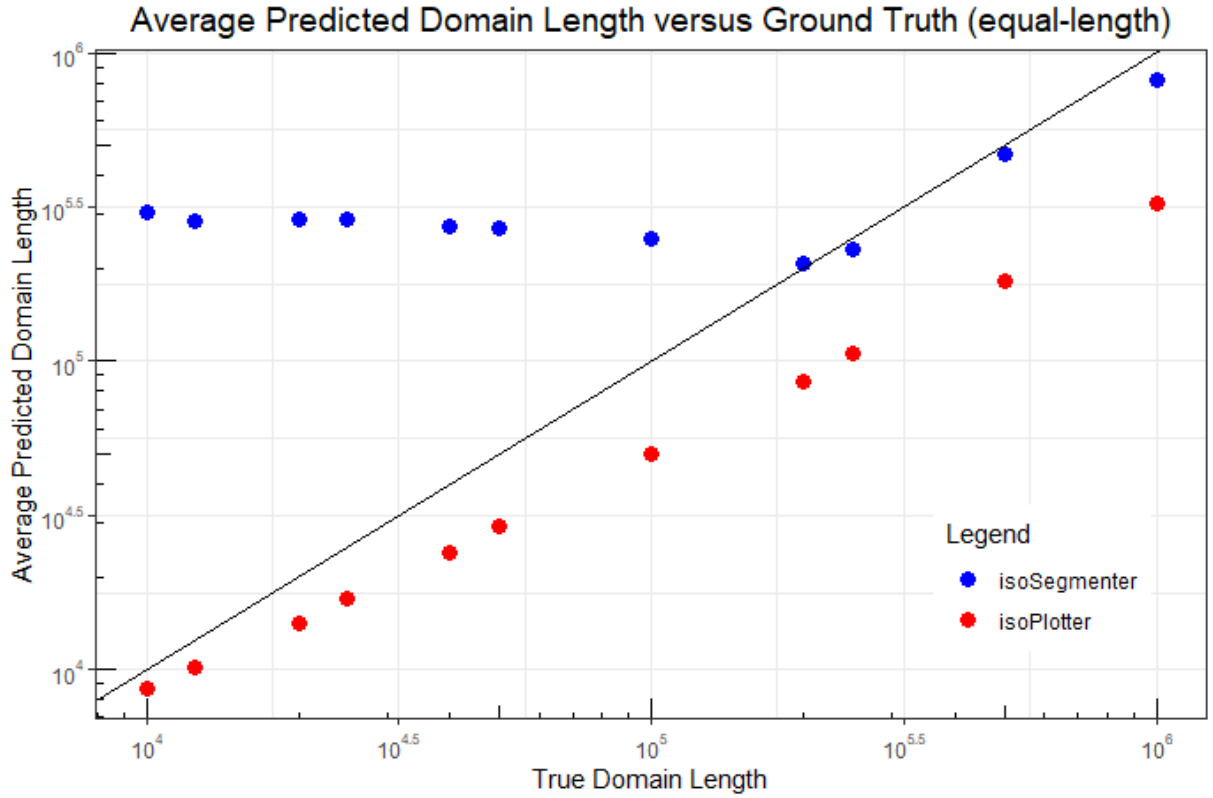


Figure 6: A comparison of average predicted domain length for isoPlotter and isoSegmenter to the actual domain length in the simulated sequence (data is for equal domain length sequences only).

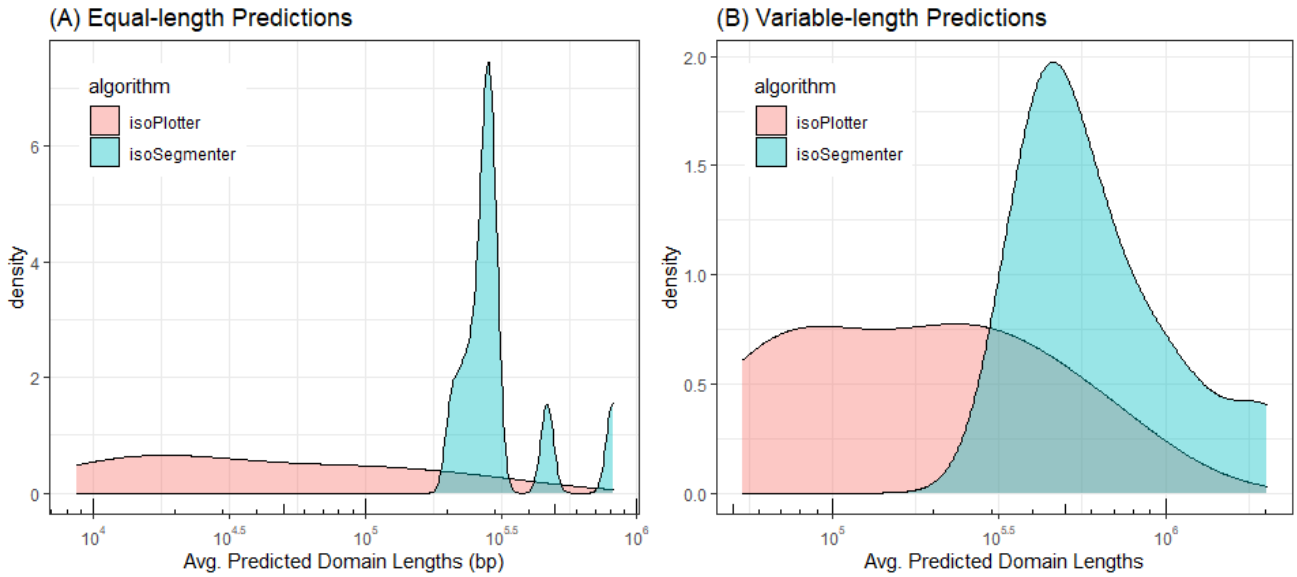


Figure 7: Distributions of the averages of predicted domain lengths for **(a)** each type of equal-length simulated sequence and **(b)** each type of variable-length sequence. Both distributions display a peak in average isoSegmenter predicted lengths near the 200,000-300,000 length range.

DISCUSSION

In this study, I used a series of controlled, randomly generated simulated sequences to conduct a benchmark comparison of isoPlotter and isoSegmenter, two compositional segmentation algorithms that deliver highly contradictory predictions. After testing the predictive power of these two algorithms under 11 equal-length domain scenarios and 9 variable-length domain scenarios, the results showed that the isoPlotter algorithm has a higher overall sensitivity, precision rate, and Jaccard Similarity Index, and that all three of these differences in performance between isoPlotter and isoSegmenter were statistically significant (Table 2). Additionally, in-depth analysis of performance across various domain-length scenarios revealed that isoSegmenter experiences a sharp drop in performance as domain lengths decrease below 500,000bp, due to its systematic tendency to predict large domains that are 200kb or longer (Figures 5, 6). On the other hand, isoPlotter exhibits a much more consistent, higher level of performance across all domain lengths, and although it has a systematic tendency to predict smaller domain sizes than the ground truth, isoPlotter achieves much higher performance metrics because its predictions are able to accommodate a wider range of domain lengths (Figure 7). Thus, after taking all of these observations into account, the results affirm my initial hypothesis that the isoPlotter algorithm is more accurate and effective when compared to isoSegmenter, and that isoPlotter's predictions are more reliable than those made by isoSegmenter.

Based on the findings of Elhaik et al. (2010a) for the Costantini et al. (2006) algorithm (the parent algorithm of isoSegmenter), I predicted that isoSegmenter's subjective default 100kb window length would cause its predictions to spike around domain lengths of 100kb and decline elsewhere. However, I observed that the majority of isoSegmenter's predictions were 200,000bp in length, rather than 100,000bp (Figure 7). This can be explained by a new isoSegmenter default setting added to the original Costantini algorithm, which requires all predicted domains to be at least two windows long. This default prediction size of two 100,000kb windows equates to the minimum, highly frequent prediction size of 200,000bp observed earlier.

My findings that isoSegmenter's median sensitivity and precision rates were close to zero (Table 2) agree with Elhaik et al.'s (2010a) benchmark sensitivity and precision analysis of isoSegmenter's parent algorithm created by Costantini et al. (2006). However, Elhaik and

colleagues found that the sensitivity and precision of isoPlotter's parent algorithm, the D_{JS} Jenson-Shannon Divergence algorithm, were above 90%, which was much higher than isoPlotter's performance in this study, with a median sensitivity and precision of 52% and 29%, respectively. Although my design approach to the simulated test sequences used in this study was very different from the design methods in Elhaik et al. (2010a), I used the same method of scoring and the same 5% boundary difference threshold established in their study. As such, it is possible that the reduction in median performance between the D_{JS} and the isoPlotter algorithms could be attributed to isoPlotter's new automatic thresholding features, however, further analysis is necessary to confirm this finding.

Despite the fact that all simulated domains were generated based upon Bernardi's Isochore Families (Bernardi et al. 2000), which isoSegmenter uses to classify its segmentations (Cozzi et al. 2015), isoSegmenter was still unable to properly segment these simulated sequences. In fact, isoSegmenter predicted 3-5 300,000bp long isochoric domains for a sequence composed of 100 distinct 10,000bp domains, a sequence that most definitely did not contain isochores. The feature of isoSegmenter which joins together nearby windows belonging to the same family helps explain how these artificial large-domain predictions arose (Cozzi et al. 2015). These results demonstrate isoSegmenter's clear, systematic bias towards the generation of large, isochore-length domains (>300kb), even when the actual genomic sequence does not contain any semblance of isochores.

Implications:

This systematic bias exhibited by isoSegmenter raises questions regarding the validity of the findings of Cozzi et al. (2015), as well as the validity of multiple other papers which used isoSegmenter as a key part of their computational genomic analyses (Afreixo et al. 2016, Arhondakis et al. 2020, Ayad et al. 2020, Bernardi 2015, Delage et al. 2020, Li et al. 2019, Mourad et al. 2020, Nacheva et al. 2017). In the paper which published the isoSegmenter algorithm, Cozzi and colleagues criticized other algorithms such as isoPlotter as being "...an exercise in DNA sequence segmentation with no biological relevance…" However, the first and most important priority of a segmentation algorithm is to make predictions that accurately reflect the underlying, true genomic sequence. A segmentation algorithm which superimposes

unrepresentative, artificial boundaries onto a genomic sequence with no grounds in the true compositional profile is the one which truly lacks biological relevance.

It is also worth noting that isoPlotter also consistently underpredicts the size of compositional domains (Figures 6, 7). However, every algorithm exhibits some form of compromise or shortcoming, and such shortcomings can be accepted given that the algorithm generates predictions which are representative of the ground truth. In this regard, isoPlotter far outperforms isoSegmenter, for its domain-length predictions clearly trend smaller as domain lengths decrease, unlike isoSegmenter, whose predictions do not reflect underlying domain sizes. Looking back at the differences in domain predictions between isoSegmenter and isoPlotter (Figure 2), this disparity is justified by my findings that isoSegmenter and isoPlotter have opposing tendencies to under-segment and over-segment sequences, respectively.

Limitations:

Although simulated sequences enable controlled benchmark studies and allow for the accurate calculation of performance metrics, they are not representative of real genomic sequences. Other studies of DNA compositional architecture did not identify any noticeable differences between the AT/GC skew behavior of random sequences and actual genomic sequences, however, real genomic sequences were found to have higher variation in A-T skew compared to G-C skew (Fimmel et al.). Additionally, while the variable-length simulated sequences provided a good approximation of true variation in domain lengths as reported by Cohen et al. (2005), it is reasonable to assume that not all inter-domain boundaries in real sequences will be as sharp and pronounced as the boundaries in my simulated sequences. Hence, while the insights from this study provide a good general idea of the performance of isoPlotter and isoSegmenter, these insights are incomplete without a thorough analysis of performance in real-world scenarios.

Future Recommendations and Conclusions:

Thus, in future studies I hope to build upon the trends and findings observed in this paper by analyzing the performance of these algorithms on real genomic sequences. Knowing of isoSegmenter's tendency to predict unrealistically large domains in genomic sequences, isochore data predicted by isoSegmenter or its parent algorithm (Costantini et al. 2006) should not be used as ground truth data, for this would lead to circular logic and results. Instead, future studies using real genomic data to compare isochore algorithms are advised to use criteria such as homogeneity of predicted domains to evaluate accuracy, in a manner similar to testing conducted by Elhaik et al. (2010a). I also would like to conduct an in-depth comparison between isoPlotter and its parent algorithm D_{JS} to rectify their differences in performance reported above. Furthermore, given that both isoSegmenter and isoPlotter had some form of bias in their predictions (Figure 6), future studies should also look into the design of newer segmentation algorithms which are capable of more accurate, representative predictions.

Above all, it is my hope that the results of this study can help inform the algorithm choices of future researchers in the field of genome compositional analysis. I also hope that the findings from this study justify some of the contradictory results obtained with regards to isochore theory, and that the choice of stronger, more accurate compositional segmentation algorithms aided by these results will contribute to increased consensus among future studies of the compositional architectures of vertebrate genomes.

REFERENCES

Afreixo V, Rodrigues JM, Bastos CAC, Silva RM. 2016. The exceptional genomic word symmetry along DNA sequences. BMC bioinformatics. 17(1):1–10.

Arhondakis S, Milanesi M, Castrignanò T, Gioiosa S, Valentini A, Chillemi G. 2020. Evidence of distinct gene functional patterns in GC-poor and GC-rich isochores in Bos taurus. Animal genetics. 51(3):358–368.

Ayad LAK, Dourou A-M, Arhondakis S, Pissis SP. 2020. IsoXpressor: a tool to assess transcriptional activity within isochores. Genome biology and evolution. 12(9):1573–1578.

Belle EMS, Smith N, Eyre-Walker A. 2002. Analysis of the phylogenetic distribution of isochores in vertebrates and a test of the thermal stability hypothesis. J Mol Evol. 55(3):356–363. eng. doi:10.1007/s00239-002-2333-1.

Benjamini Y, Hochberg Y. 1995. Controlling the False Discovery Rate: A Practical and Powerful Approach to Multiple Testing. Journal of the Royal Statistical Society: Series B (Methodological). 57(1):289–300. doi:10.1111/j.2517-6161.1995.tb02031.x.

Bernardi G. 1993. The vertebrate genome: isochores and evolution. Mol Biol Evol. 10(1):186–204. eng. doi:10.1093/oxfordjournals.molbev.a039994.

Bernardi G. 2000. Isochores and the evolutionary genomics of vertebrates. Gene. 241(1):3–17. eng. doi:10.1016/s0378-1119(99)00485-0.

Bernardi G. 2001. Misunderstandings about isochores. Part 1. Gene. 276(1-2):3–13. eng. doi:10.1016/s0378-1119(01)00644-8.

Bernardi G, Olofsson B, Filipski J, Zerial M, Salinas J, Cuny G, Meunier-Rotival M, Rodier F. 1985. The mosaic genome of warm-blooded vertebrates. Science. 228(4702):953–958. eng. doi:10.1126/science.4001930.

Bernardi G. 2015. Chromosome architecture and genome organization. PLoS One. 10(11):e0143739.

Bovine Genome Sequencing and Analysis Consortium. 2009. The genome sequence of taurine cattle: a window to ruminant biology and evolution. Science. 324(5926):522–528. eng. doi:10.1126/science.1169588.

Clauset A, Shalizi CR, Newman MEJ. 2009. Power-Law Distributions in Empirical Data. SIAM Rev. 51(4):661–703. doi:10.1137/070710111.

Clay O, Bernardi G. 2005. How not to search for isochores: a reply to Cohen et Al. Mol Biol Evol. 22(12):2315–2317. eng. doi:10.1093/molbev/msi231.

Clay OK, Bernardi G. 2011. GC3 of genes can be used as a proxy for isochore base composition: a reply to Elhaik et al. Mol Biol Evol. 28(1):21–23. eng. doi:10.1093/molbev/msq222.

Cock PJA, Antao T, Chang JT, Chapman BA, Cox CJ, Dalke A, Friedberg I, Hamelryck T, Kauff F, Wilczynski B, et al. 2009. Biopython: freely available Python tools for computational molecular biology and bioinformatics. Bioinformatics. 25(11):1422–1423. eng. doi:10.1093/bioinformatics/btp163.

Cohen N, Dagan T, Stone L, Graur D. 2005. GC composition of the human genome: in search of isochores. Mol Biol Evol. 22(5):1260–1272. eng. doi:10.1093/molbev/msi115.

Costantini M, Clay O, Auletta F, Bernardi G. 2006. An isochore map of human chromosomes. Genome Res. 16(4):536–541. eng. doi:10.1101/gr.4910606.

Cozzi P, Milanesi L, Bernardi G. 2015. Segmenting the Human Genome into Isochores. Evol Bioinform Online. 11:253–261. eng. doi:10.4137/EBO.S27693.

Cuny G, Soriano P, Macaya G, Bernardi G. 1981. The major components of the mouse and human genomes. 1. Preparation, basic properties and compositional heterogeneity. Eur J Biochem. 115(2):227–233. eng. doi:10.1111/j.1432-1033.1981.tb05227.x.

Delage W, Thevenon J, Lemaitre C. 2020. Towards a better understanding of the low discovery rate of short-read based insertion variant callers. In: JOBIM 2020. [place unknown]: [publisher unknown].

Elhaik E, Graur D. 2014. A comparative study and a phylogenetic exploration of the compositional architectures of mammalian nuclear genomes. PLoS Comput Biol. 10(11):e1003925. eng. doi:10.1371/journal.pcbi.1003925.

Elhaik E, Graur D, Josic K. 2010a. Comparative testing of DNA segmentation algorithms using benchmark simulations. Mol Biol Evol. 27(5):1015–1024. eng. doi:10.1093/molbev/msp307.

Elhaik E, Graur D, Josić K, Landan G. 2010b. Identifying compositionally homogeneous and nonhomogeneous domains within the human genome using a novel segmentation algorithm. Nucleic Acids Res. 38(15):e158. eng. doi:10.1093/nar/gkq532.

Fimmel E, Gumbel M, Karpuzoglu A, Petoukhov S. 2019. On comparing composition principles of long DNA sequences with those of random ones. Biosystems. 180:101–108. eng. doi:10.1016/j.biosystems.2019.04.003.

Fukagawa T, Sugaya K, Matsumoto K, Okumura K, Ando A, Inoko H, Ikemura T. 1995. A boundary of long-range G + C% mosaic domains in the human MHC locus: pseudoautosomal boundary-like sequence exists near the boundary. Genomics. 25(1):184–191. eng. doi:10.1016/0888-7543(95)80124-5.

Graur D, Sater AK, Cooper TF. 2016. Molecular and genome evolution. Sunderland, Massachusetts: Sinauer Associates. ISBN: 9781605354699.

Lander ES, and 254 others. 2001. Initial sequencing and analysis of the human genome. Nature. 409(6822):860–921. eng. doi:10.1038/35057062.

Li W, Bernaola-Galván P, Carpena P, Oliver JL. 2003. Isochores merit the prefix 'iso'. Comput Biol Chem. 27(1):5–10. eng. doi:10.1016/s1476-9271(02)00090-7.

Macaya G, Thiery JP, Bernardi G. 1976. An approach to the organization of eukaryotic genomes at a macromolecular level. J Mol Biol. 108(1):237–254. eng. doi:10.1016/s0022-2836(76)80105-2.

Michaux J, Reyes A, Catzeflis F. 2001. Evolutionary history of the most speciose mammals: molecular phylogeny of muroid rodents. Mol Biol Evol. 18(11):2017–2031. eng. doi:10.1093/oxfordjournals.molbev.a003743.

Mourad R. 2020. Studying 3D genome evolution using genomic sequence. Bioinformatics. 36(5):1367–1373.

Nacheva E, Mokretar K, Soenmez A, Pittman AM, Grace C, Valli R, Ejaz A, Vattathil S, Maserati E, Houlden H, et al. 2017. DNA isolation protocol effects on nuclear DNA analysis by microarrays, droplet digital PCR, and whole genome sequencing, and on mitochondrial DNA copy number estimation. PLoS One. 12(7):e0180467.

Schmidt T, Frishman D. 2008. Assignment of isochores for all completely sequenced vertebrate genomes using a consensus. Genome Biol. R104. eng. doi:10.1186/gb-2008-9-6-r104.

Thiery JP, Macaya G, Bernardi G. 1976. An analysis of eukaryotic genomes by density gradient centrifugation. J Mol Biol. 108(1):219–235. eng. doi:10.1016/s0022-2836(76)80104-0.

Wickham H. 2016. ggplot2: Elegant Graphics for Data Analysis. [place unknown]: Springer-Verlag New York. ISBN: 978-3-319-24277-4. https://ggplot2.tidyverse.org/.