

ForiC-a Fast and Effective Method for Predicting the Origin of Bacterial Chromosome Replication

Stojićević, Marko

Abstract: Identifying the origin of replication (oriC) is fundamental to understanding bacterial genome organization and replication mechanisms. Traditional methods, such as GC-skew and cumulative GC-skew (CGC) analyses, are computationally simple but often imprecise. Advanced approaches, including DnaA-box detection and comparative genomics, can improve accuracy but are frequently resource-intensive and dependent on external biological information. This paper presents ForiC, a novel method that bridges this gap by combining the simplicity of GC- and CGC-based analyses with improved oriC-region localization performance. Beyond using CGC inflection points, ForiC integrates three complementary signals within a deterministic selection rule: the CGC global maximum, the CGC inflection point associated with that maximum, and its alignment with the GC global minimum. The resulting framework operates directly on the input genome sequence, without DnaA-box motifs, homology searches, or external databases. ForiC was calibrated and evaluated on a curated set of 36 annotated bacterial genomes using five phylum-stratified training/test splits. On one pre-defined stratified test subset selected as a representative balanced test subset, ForiC reached a hit accuracy of 90.91% using a 1% window and a symmetrically expanded two-window reporting interval, with complementary localization metrics (center-distance error, intersection-over-union) and one-window results reported in the main text. ForiC was benchmarked against classical skew-based methods (GC, CGC) and against a biologically informed reference tool (OriFinder). OriFinder achieves higher positional precision through homology-based refinement but may report multiple candidate oriC regions and depends on external databases such as BLAST. In contrast,

ForiC consistently produces a single, directly interpretable prediction per genome and runs substantially faster without external resources, making it complementary to reference-assisted tools and particularly suitable for large-scale, database-independent analyses. ForiC is available at <http://www.markostojicevic.com/foric>.

Index Terms: *algorithms, bioinformatics, computational genomics, GC skew, oriC prediction*

1. INTRODUCTION

DNA replication is one of the most important biological processes. This ensures the inheritance of attributes that define the phenotype of cells and the organism. In this complicated process, genetic information must be accurately copied to avoid mutations and ensure the survival of living organisms. Complicated as it is, the DNA replication process tends to differ in all three domains of life (the Bacteria, the Eukarya and the Archaea) and between organisms in the same domain [1], [2], [3], [4], [5].

Bacteria usually contain circular DNA chromosomes with a specific locus designated as the origin of replication (oriC). oriC contains three functional elements: a DNA repeat, an AT-rich region and regulatory protein recognition sites [6]. Moreover, the analysis of bacterial oriC regions revealed high-scale diversity, increasing the complexity of oriC sequence recognition [7], [8]. To better understand the mechanism of DNA replication initiation, it is important to identify the position of the replication origin.

DNA asymmetry is the most universal method of putative oriC identification in bacterial chromosomes [9]. The analysis revealed that the leading strand contains more Guanine (G) and Thymine (T), whereas the lagging strand contains more Cytosine (C) and Adenine (A) [10], [11]. The

¹ Manuscript submitted on August 2, 2026. Marko Stojićević, University of Belgrade, Faculty of Mathematics, Department of Informatics and Computer Science, Belgrade, Serbia, mstojicevic@gmail.com, ORCID: 0009-0005-8771-2012.

asymmetry of the base pairs that have been found in the strands is strong enough to identify the location of the oriC position via mathematical methods [12]. There are three main methods that measure nucleotide asymmetry: GC skew, cumulative GC (CGC) skew and the Z curve.

Although these approaches are widely used, each exhibits specific limitations. Basic GC skew is sensitive to window size and can produce multiple ambiguous asymmetry transitions, which reduces positional accuracy [13], [14]. Cumulative GC skew smooths the signal but often yields broad candidate regions rather than precise oriC locations, and its accuracy decreases in genomes where the GC skew signal itself is weak or where local compositional biases distort the cumulative curve [15]. The Z-curve provides richer compositional information but requires careful interpretation of multiple curve components, and in practice its accuracy typically depends on integration with additional biological signals such as DnaA-box motifs [16], [17]. These limitations motivate the development of methods that preserve the computational simplicity of skew-based analyses while increasing localization accuracy through additional, intrinsically derived features.

The GC skew method is based on GC asymmetry and is calculated as $(G-C)/(G+C)$ [10], [13], [14]. This method is based on the frequency of base pairs C and G in a given sequence of the genome. The genome is divided into equal-sized windows, and for each window, the imbalance between guanine (G) and cytosine (C) nucleotides is quantified as the GC skew value. These values are often represented in the form of a graph, where the x-axis shows the position (of the window) within the genome and the y-axis shows the GC skew value. The selected window size strongly influences the GC skew signal, affecting how clearly genomic patterns can be visualized and interpreted. This highlights the necessity of carefully selecting an appropriate window size to facilitate accurate analysis of the GC skew.

The second method, CGC skew, resolves most of the flaws of the GC skew method and is calculated as a sum of $(G-C)/(G+C)$ in adjacent windows from an arbitrary start to a given point in a sequence [18]. CGC skew also uses sliding windows, but each value reflects the running sum of GC differences from all preceding windows. As a result, CGC skew plots are typically smoother

and easier to interpret than standard GC skew plots, since the cumulative transformation reduces abrupt changes in curve curvature.

The third method is the Z-curve method, which represents a DNA sequence as a three-dimensional mathematical curve obtained by cumulating nucleotide counts along the genome [16], [19], [20], [21]. Its three coordinates capture complementary aspects of base composition: purine/pyrimidine (R/Y) asymmetry, amino/keto (M/K) asymmetry, and strong/weak hydrogen-bond (S/W) asymmetry, corresponding to the A+G vs. C+T, A+C vs. G+T, and G+C vs. A+T distinctions, respectively. Unlike the previous methods, the Z-curve does not require a sliding-window strategy; characteristic turning points of these curves have been used to infer candidate oriC positions.

Deeper analysis in various studies indicates that the switch in asymmetry does not always correspond to the position of oriC [22]. Various software tools implement methods such as GC, CGC, Z-curve, and other skew techniques [23]. Among the bacterial oriC predictors, one of the most widely used tools is OriFinder [17]. This algorithm comprises several steps, with the primary components being the determination of the Z-curve and analysis of DnaA-box distributions in the given bacterial genome. The predicted genomes are deposited in DoriC [24], a database of predicted oriC positions. In recent years, more species-specific indicator genes and functional elements of oriCs have been integrated into updated frameworks, leading to newer versions of OriFinder [25].

Recent studies have also introduced alternative oriC prediction frameworks, including open-source and machine-learning-based approaches [26], [27], [28]. However, these methods differ substantially in scope, target datasets, or implementation strategies. Therefore, OriFinder was retained in the present study as the principal comparative benchmark due to its broad recognition and established use in bacterial oriC prediction.

More specifically, existing models present several practical drawbacks. Basic GC and CGC skew methods often lack precision in predicting the exact location of oriC, frequently identifying broad candidate regions rather than the precise position of the oriC site. OriFinder, while regarded

as a highly accurate tool, depends heavily on BLAST for functional annotation of genes in unannotated genomes. This reliance introduces two key limitations: (1) the computational overhead of performing sequence alignments, which increases processing time and resource demands, especially for large-scale analyses, and (2) the dependence on the completeness and accuracy of external databases, which may vary over time and across research environments.

This study aims to propose a new, more effective method for oriC prediction that overcomes these limitations by utilizing an enhanced GC asymmetry-based approach. The ForiC (Fast oriC) platform eliminates the need for BLAST or extensive biological information such as DnaA box motifs. Instead, the method leverages a combination of GC and CGC skew analyses as the foundation. These methods are further refined by incorporating a mathematical framework that analyzes higher-order structural properties of CGC skew curves, rather than relying solely on local maxima, minima, or direct asymmetry transitions. In particular, the approach uses inflection points as informative features and combines them with additional genomic signals to improve oriC localization. Key parameters of the method, specifically the window size and prediction interval length, were calibrated empirically by evaluating candidate values on stratified training partitions and selecting the configuration with the most consistent performance. This approach differs from other methods by providing a faster, database-independent, and more versatile tool for bacterial origin prediction.

2. METHODS AND MATERIALS

2.1. *ForiC method description*

The ForiC method incorporates an initial input-validation phase to ensure that only biologically plausible genome sequences are subjected to detailed analysis. This preliminary screening includes basic sequence-length and GC-content sanity checks, together with simple compositional descriptors such as entropy, GC asymmetry, and nucleotide-count variance. These measures are intended to exclude trivial, highly atypical, or low-information inputs before the main prediction stage. Full definitions and implementation details

of the preliminary validation criteria are provided in [Additional file 1](#).

After this screening step, the validated genome sequence is processed using the core ForiC framework based on sliding-window GC and CGC skew analysis. Similar to GC and CGC skew methods, ForiC requires only the genome sequence as input and does not depend on external annotations or prior biological features. The computational complexity of the ForiC algorithm can be described as $O(n)$, where n denotes the number of base pairs in the genome sequence.

A defining feature of the ForiC algorithm is the use of specific inflection points (IPs) within the CGC graph (positions where the numerically approximated curvature changes sign) as an auxiliary positional cue alongside the CGC global maximum and the GC global minimum. The inflection point is not intended to replace the CGC extremum traditionally used to bracket oriC, but to refine the predicted interval when the global maximum alone may not unambiguously localize the oriC-containing region. Among these, the first inflection point following the CGC global maximum, denoted $IPCGC_{max}$, is particularly significant. This point was empirically observed to occur near annotated oriC regions in many genomes. To increase localization accuracy, the algorithm cross-references $IPCGC_{max}$ with the x-axis position of the GC global minimum. By combining these features, the ForiC algorithm is designed to improve oriC localization by integrating complementary signals from GC and CGC analyses. Figure 1 shows the CGC and GC graphs, highlighting the global extrema and inflection points used in the algorithm.



Figure 1: *Escherichia coli* GC and CGC graph with CGC inflection points. The graph shows (for the genome CP006027 *Escherichia coli* O145:H28 str. RM13514) plots for AT, CAT, GC and CGC measures, with noted global extreme values (minimum and maximum). The CGC plot also includes inflection points (shown as dots), where those of interest are those after the CGC global maximum value. The dashed black line denotes that the x-axis values overlap between the CGC inflection point and the GC global minimum value, whereas the dashed orange line denotes the oriC position.

It is appropriate to define an inflection point:

An inflection point on a differentiable curve $y = f(x)$ is a point (x_0, y_0) where the second derivative $f''(x)$ changes sign. At an inflection point, the concavity of the curve changes from concave up to concave down, or vice versa [29].

Importantly, the CGC skew does not define a curve $y = f(x)$ but rather generates a finite set of values $(x_1, x_2, \dots, x_n)$. These values can visually resemble the function's graph, but the method itself does not generate the (continuous) function's equation $f(x)$. Considering that there is no analytical definition that describes the set of x_i values, calculating the second derivative is not possible. The second derivative can be calculated only as a numerical approximation to obtain the inflection points.

The precondition of any of the calculations is to have a smooth function. The smoothness of the function depends on the number of windows used in the CGC method; as the window tends to be smaller, the number of windows (the set of x_i values) increases, leading to a smoother function. A point x_i is an inflection point whenever there is a polarity change in the numerical approximation of the second derivative. This approximation can be performed in multiple steps - first, to approximate the first derivative based on a set of x_i values and then, to calculate the approximated second derivative on the basis of the first derivative.

The first step is to approximate the first derivative using the sequence of values x_i . For each pair of consecutive windows, a new temporary variable q_i is defined as:

$$q_i = \frac{x_{i+1} - x_i}{m}, \quad i = 1, \dots, n-1$$

where m represents the size of the sliding window. The sequence $q_1, q_2, \dots, q_{n-1}$ represents an approximation of the first derivation of the function $f: f'(x)$.

The second derivative is then approximated by applying the same finite-difference procedure to consecutive values of q_i :

$$p_i = \frac{q_{i+1} - q_i}{m}, \quad i = 1, \dots, n-2$$

where m denotes the size of the sliding window.

Equivalently, this expression can be written directly in terms of the original values:

$$p_i = \frac{x_{i+2} - 2x_{i+1} + x_i}{m^2}$$

The sequence $p_1, p_2, \dots, p_{n-2}$ therefore represents an approximation of the second derivative, $f''(x)$.

On the basis of the previous equations, a point x_i is the inflection point whenever the condition $p_i * p_{i+1} < 0$ is met. The number of inflection points differs for different CGC calculations, but the number tends to decrease as the window size increases and vice versa.

CGC skew analysis is commonly used in conjunction with other genomic features to predict oriC in bacterial genomes. While various computational methods exist for oriC prediction, the transition points in CGC skew analysis, such as global maximum and global minimum values, are often considered indicative of genomic regions corresponding to oriC and the terminus, respectively. Based on the above, only inflection points around the CGC's global maximal value were important.

2.2. *ForiC Algorithm*

The algorithm applies a sliding-window strategy combining GC skew, CGC skew, and derived CGC inflection-point features. Specifically, it calculates graph inflection points, analyzes the derived data (GC, CGC, inflection points), and evaluates the correlation between these values. These steps are systematically implemented to improve the prediction precision. The ForiC algorithm is currently applied to circular bacterial genomes, and its structure is presented in the pseudocode shown in Figure 2.

Following the definition of these features, the initial stage of the algorithm involves calculating the GC skew and the CGC skew and identifying inflection points within the CGC data, which serve as critical features for subsequent analyses. As the algorithm progresses, the positioning of the oriC exhibits two distinct scenarios: either oriC is located near the origin of the circular genome's coordinate system or elsewhere within the genomic sequence.

The algorithm uses different measures for predicting the oriC position: (1) the position of the CGC inflection point associated with the CGC global maximum, (2) the overlap of the x-axis values between the CGC inflection points and the GC global minimum position, and (3) the position of the CGC global maximum.

The ForiC algorithm supports variable window sizes, allowing for adjustable granularity in the analysis of bacterial genomes. Window sizes were defined as proportions of the genome length. This flexibility facilitates interval-based predictions for oriC regions, with the output defined as a predicted genomic interval. In the standard one-window setting (1W), the predicted interval spans a single window length. In the extended two-window setting (2W), the reported oriC interval is not defined as two separately selected neighboring windows. Instead, the interval is symmetrically expanded around the initially predicted 1W interval center to a total span of two window lengths, allowing improved capture of oriC regions that extend beyond a single window. All parameters, including window size and interval length, were systematically evaluated to ensure robust performance. These

features enhance the algorithm's adaptability across diverse genomic contexts while maintaining computational efficiency.

The algorithm applies specific rules to determine the oriC position based on the distribution of inflection points and CGC global maximum values.

Because ForiC currently supports only circular bacterial genomes, predictions near the coordinate-system boundary (the junction between the final and initial genomic positions) require special handling. A boundary-centered oriC interval is assigned when either (i) the x-axis alignment between IPCGCmax and the GC global minimum occurs within the last two CGC windows, or (ii) the CGC global maximum lies within the first window. In such cases, the predicted oriC interval is centered on coordinate 0 and reported as a wrap-around interval across the genome boundary. In the standard one-window prediction setting, this corresponds to an interval extending by half a window length on each side of coordinate 0.

If neither condition is met, the algorithm applies the standard prediction rule by identifying the informative CGC inflection point located after the CGC global maximum and aligned with the GC minimum. The predicted oriC interval is then assigned to the immediately following window, while the aligned inflection point serves as a positional marker rather than the predicted interval itself.

```

Input: genome (string), window size (integer), interval mode (1W or 2W)
Output: Predicted oriC interval
FUNCTION CalculateOriginInterval(genome: String, windowSize: Integer, intervalMode: String) : array of Integer
var
  inflectionPoints, ipAfterCGCMax, ipAlignedWithGCMin, oriCPosition: array of Integer;
  gcSkew, cgSkew: array of Real;
begin
  // Step 1: Calculate initial skew values
  gcSkew := CalculateGCskew(genome, windowSize);
  cgSkew := CalculateCGskew(genome, windowSize);
  inflectionPoints := CalculateInflectionPoints(cgSkew);

  // Step 2: Identify Inflection points of interests
  ipAfterCGCMax := GetFirstInflectionPointAfterCGCMax(cgSkew, inflectionPoints);
  ipAlignedWithGCMin := GetInflectionAlignedWithGCMin(gcSkew, inflectionPoints);

  // Step 3: Define prediction center
  if (ipAfterCGCMax = nil or (ipAlignedWithGCMin["leftPosition"] > (genomeLength - 2*windowSize)) or
    (ipAlignedWithGCMin["leftPosition"] < windowSize))
  begin
    oriCPosition["leftPosition"] := genomeLength - windowSize/2;
    oriCPosition["rightPosition"] := windowSize/2;
  end
  else
  begin
    oriCPosition["leftPosition"] := ipAlignedWithGCMin["leftPosition"] + windowSize;
    oriCPosition["rightPosition"] := oriCPosition["leftPosition"] + windowSize;
  end
  end

  // Step 4: Expand interval if intervalMode is 2W
  if (intervalMode = "2W")
  begin
    oriCPosition := ExpandSymmetrically(oriCPosition, windowSize, genomeLength);
  end

  return oriCPosition;
end;

```

Figure 2: Pseudocode of the ForiC algorithm for interval-based oriC prediction in circular bacterial genomes.

2.2.1. Metrics for Assessing Algorithmic Prediction Performance

The performance of the ForiC algorithm was evaluated primarily using hit accuracy, reflecting its ability to detect the oriC region. Unlike standard classification models that assign discrete labels, ForiC predicts a genomic interval expected to contain the oriC region. Consequently, classification-based metrics such as precision, recall, sensitivity, and specificity are not directly applicable, as they require clearly defined positive and negative instances at the same resolution. Localization-oriented metrics described below were therefore adopted instead.

Accuracy (A) was defined as the proportion of correctly predicted oriC intervals relative to the total number of genomes:

$$A = \frac{N_{correct}}{N_{total}}$$

where $N_{correct}$ represents the number of genomes for which the annotated oriC position falls within the predicted interval, and N_{total} is the total number of genomes.

However, this metric alone may favor larger prediction windows, as broader intervals have a higher probability of overlapping the annotated oriC region. It is well established that accuracy alone may provide a misleading assessment in tasks where prediction granularity varies [30]. Therefore, additional localization-oriented metrics were introduced to complement detection performance and assess spatial precision.

The circular distance error between the centers of the predicted and reference oriC intervals was computed as:

$$d = \min(|c_p - c_r|, L - |c_p - c_r|)$$

where c_p and c_r are the centers of the predicted and reference (experimentally confirmed) intervals, respectively, and L is the genome length. This formulation follows the standard definition of distance on circular range [31] and quantifies how close the prediction is to the true oriC location, independently of the window size.

The intersection-over-union (IoU) score, commonly used in localization tasks to jointly assess detection and spatial precision [32], [33], was used to assess the degree of overlap between predicted and reference intervals:

$$IoU = \frac{|Pred \cap Ref|}{|Pred \cup Ref|}$$

This metric penalizes overly large predictions and rewards precise localization.

Each metric was computed separately for each genome based on the predicted and annotated oriC intervals. Hit accuracy was then obtained as the proportion of genomes with overlapping predicted and reference intervals, while circular distance error and IoU values were summarized across genomes using mean and median values, with particular emphasis on the median as a robust measure against outliers.

Due to the circular nature of bacterial genomes, intervals may span the coordinate

origin. Such cases were handled by representing intervals in a circular coordinate system, and for overlap-based calculations, intervals crossing the genome boundary were split into segments to ensure correct computation of intersection and union lengths.

Together, these metrics enable a balanced evaluation by distinguishing between detection performance (whether the *oriC* region is captured) and localization precision (how accurately it is localized).

2.3. ForiC Implementation

The ForiC's source code is publicly accessible and can be downloaded via a publicly available <http://bioinfo.matf.bg.ac.rs/home/downloads.waf?cat=Software&project=ForiC>.

Comprehensive instructions for the installation process, detailed usage options, and test cases are included in the provided link. Furthermore, ForiC is available online for testing purposes at <http://www.markostojicevic.com/foric>.

2.4. Materials

The dataset used in this study was retrieved from the NCBI database (<https://www.ncbi.nlm.nih.gov/nucleotide/>) in April 2026. Genome records were selected using a stringent query designed to include only bacterial chromosomes with experimentally validated origins of replication, annotated with the feature key “rep_origin” in GenBank format. Although the query targeted a broad range of bacterial genomes annotated with “rep_origin”, only 85 records were retrieved, reflecting the limited availability of such annotations.

From this set, only circular chromosomes were retained, as cumulative GC skew patterns are most consistently defined and interpreted in circular genomes with bidirectional replication [34], [35]. Additional filtering was applied to ensure a consistent and unambiguous benchmark: duplicate entries cross-listed in NCBI under multiple accession codes (i.e., the same sequence indexed under more than one accession), as well as genomes with multiple or ambiguous “rep_origin” annotations that could

not be mapped to a single continuous *oriC* region, were excluded. Furthermore, entries with computationally inferred *oriC* annotations (e.g., based on GC-skew analysis or automated annotation tools), sequence similarity-based annotations, or annotations lacking sufficient descriptive information were also excluded. The orientation of the *oriC* annotation (direct or complement) was not used as an exclusion criterion, as strand asymmetry in nucleotide composition remains consistent regardless of annotation direction [35].

Following this filtering procedure, 52 genomes were retained and classified into two groups. The primary dataset consisted of 36 genomes with experimentally validated *oriC* annotations and was used for parameter calibration and testing. The extended dataset comprised 16 additional genomes with curated but non-experimental *oriC* annotations, which were used exclusively to evaluate the robustness and generalizability of the method.

The dataset shows an uneven taxonomic distribution, with a predominance of Pseudomonadota genomes, reflecting the current availability of experimentally validated *oriC* annotations.

The primary dataset, used for parameter calibration and testing, is summarized in Table 1. The extended dataset is reported separately to avoid mixing annotations of different confidence levels. The complete list of genomes, including excluded entries and the specific reasons for their omission, is provided in **Additional file 2**. The relatively small size of the primary dataset reflects the limited availability of experimentally validated *oriC* annotations rather than an arbitrary selection bias, highlighting a broader limitation in current *oriC* benchmarking studies.

Table 1. Summary of the primary dataset (36 genomes), categorized by phylum

Phylum	N	Average length (bp)	Min. length (bp)	Max. length (bp)
Actinomycetota	1	9,239,851	9,239,851	9,239,851
Bacillota	3	3,128,281	1,991,969	5,346,304
Cyanobacteriota	1	8,680,294	8,680,294	8,680,294
Pseudomonadota	30	5,110,205	2,991,126	7,257,956
Spirochaetota	1	1,139,590	1,139,590	1,139,590
Total	36	5,048,631	1,139,590	9,239,851

2.4.1. Preparation of training and test datasets

The accuracy of ForiC was assessed by comparing predicted *oriC* positions to annotated reference positions obtained from GenBank records.

The primary dataset was divided into five stratified subsets based on phylum-level taxonomic distribution to improve representation of diverse bacterial groups across both training and test partitions. Five stratified splits were constructed to balance robustness of performance estimation against overreliance on any single data partition. In each split, 25 genomes (69.4%) were allocated to the training subset and 11 genomes (30.6%) to the testing subset. A 70:30 partitioning scheme was selected instead of the commonly used 80:20 ratio to

retain a more informative test subset given the relatively limited dataset size, while preserving adequate data for parameter calibration.

Stratification at the phylum level was applied to preserve taxonomic diversity across subsets and reduce potential bias toward overrepresented groups. Within each phylum, genomes were allocated to the training and test partitions using a manual balancing procedure designed to preserve representation while approximating the target 70:30 proportion. For phyla represented by only a small number of genomes, exact proportional allocation was not always feasible, and assignments were adjusted to ensure that each phylum was represented in both training and test partitions whenever possible. Because the allocation was

Table 2. Distribution of phyla per dataset, training/testing set

Phylum	DATASET 1		DATASET 2		DATASET 3		DATASET 4		DATASET 5	
	Training set	Testing set	Training set	Testing set	Training set	Testing set	Training set	Testing set	Training set	Testing set
Actinomycetota	0	1	1	0	1	0	0	1	1	0
Bacillota	2	1	2	1	2	1	2	1	2	1
Cyanobacteriota	1	0	0	1	1	0	1	0	0	1
Pseudomonadota	21	9	21	9	21	9	21	9	21	9
Spirochaetota	1	0	1	0	0	1	1	0	1	0
Total	25	11	25	11	25	11	25	11	25	11

deterministic and manually balanced rather than stochastic; no random seed was used, and the resulting splits were used consistently throughout all calibration and evaluation analyses, and their full composition is provided in [Additional file 2](#).

The partitioning enabled systematic evaluation of different parameter configurations, including window size and GC/CGC-related thresholds. Different window lengths, defined as proportions of the genome size, were evaluated, along with alternative strategies for interpreting predicted regions, including one-window

predictions and symmetrically expanded two-window reporting intervals.

Candidate window sizes (0.1%, 0.5%, 1%, 1.5%, and 2%) were calibrated using the training portions of each stratified subset. The final window-size setting was selected based on the most consistent overall performance across the training subsets and subsequently evaluated unchanged on the corresponding test subsets.

Evaluating parameter configurations across multiple stratified subsets reduced dependence on a single data partition and provided a more

reliable assessment of method performance across diverse genomic contexts.

Given the relatively limited size of the curated benchmark dataset (36 genomes) and the interval-localization nature of the task, performance was summarized using descriptive multi-metric comparisons across the five stratified subsets rather than formal null-hypothesis significance testing. This approach was considered more appropriate for emphasizing consistency of performance across multiple predefined partitions than inference from small-sample statistical tests.

3. RESULTS

3.1. *Quality of ForiC Predictions and Performance Analysis*

To validate the effectiveness of the ForiC method, a systematic evaluation was conducted on a dataset of bacterial genomes partitioned into five stratified datasets. The algorithm was applied using sliding window sizes ranging from 0.1% to 2% of the genome length. Performance was assessed using hit accuracy, defined as the proportion of cases in which the annotated oriC position falls within the predicted interval. All parameter tuning was performed exclusively on training subsets, while testing subsets were used for independent performance assessment, ensuring that the reported results were not influenced by parameter selection.

Performance was evaluated across all five stratified datasets and window sizes. Detailed accuracy values for both training and testing subsets are provided in Tables 3a and 3b, while additional evaluation metrics are summarized in [Additional file 3](#). Across datasets, identical hit-accuracy values were frequently observed at 1%, 1.5%, and 2% window sizes, indicating that performance had reached a plateau once the reporting interval became sufficiently broad. Therefore, a window size of 1% was selected as the smallest configuration that consistently achieved maximal or near-maximal hit accuracy while preserving better spatial resolution than larger windows.

Among the evaluated splits, Dataset 3 exhibited one of the most balanced training–testing profiles at the selected 1% window size,

with an absolute difference of 2.91 percentage points between training (88.00%) and testing (90.91%) hit accuracy under the expanded-interval criterion (Table 3b). At this configuration, Dataset 3 also maintained moderate center-distance values, typically in the range of approximately 6–30 kb (Additional file 3). Other datasets showed either larger train–test discrepancies at the same window size or less stable behavior across alternative window-size settings.

To illustrate these trends more clearly, Figure 3 presents the performance behavior for the test subset of Dataset 3. As shown in Figure 3a, hit accuracy for single-window predictions generally increased with window size. In contrast, when using the expanded prediction interval, hit accuracy increased up to a window size of 1% and then reached a plateau, with no further improvement observed for larger windows. Expanding the prediction interval to twice the window size, constructed by symmetric extension around the predicted center, increased hit accuracy under a broader interval criterion. In this configuration, hit accuracy reached up to 90.91% (10/11 genomes) in the testing subset, compared to substantially lower values for single-window predictions (e.g., 54.55% at 1% window size).

Additional evaluation metrics were used to assess localization performance. The center distance, summarized using the median, provided a robust measure of localization, whereas Intersection over Union (IoU), reported as the mean, remained low across configurations, generally on the order of 10^{-3} , reflecting the large size difference between predicted reporting intervals and experimentally annotated oriC regions (Additional file 3). The center distance remained unchanged between one-window and expanded-interval configurations, as the larger interval was generated by symmetric extension around the original prediction center. In contrast, IoU values were lower for the expanded interval configuration. Although a slightly lower center distance was observed at larger window sizes (e.g., 2%), this effect was not consistent across datasets and did not translate into consistently better overall performance. Accordingly, a window size of 1% was selected because it

provides the best balance between detection accuracy and spatial resolution.

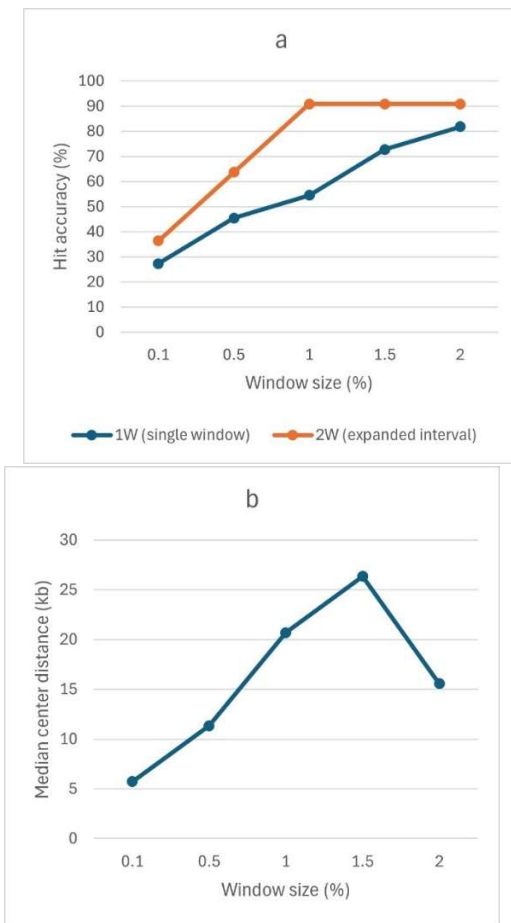


Figure 3: Effect of window size on oriC detection performance using the ForiC method, evaluated on the test subset of Dataset 3. (a) Hit accuracy for single-window (1W) predictions and for prediction intervals expanded to twice the window size (2W), constructed by symmetric extension around the predicted center. (b) Localization performance measured as median center distance between predicted and annotated oriC positions across different window sizes.

Based on these results, the 1% window-size configuration was selected for further analysis and comparison with existing methods.

Table 3a. Hit accuracy (%) for *oriC* prediction across five stratified datasets using single-window (1W) prediction intervals. Results are shown separately for training and testing subsets, along with the difference between them.

Win- dow size (%)	Dataset 1			Dataset 2			Dataset 3			Dataset 4			Dataset 5		
	Train- ing	Test	Absolute gap (pp)	Training	Test	Absolute gap (pp)	Train- ing	Test	Absolute gap (pp)	Train- ing	Test	Absolute gap (pp)	Train- ing	Test	Absolute gap (pp)
0.1	40.00%	63.64%	23.64%	44.00%	54.55%	10.55%	56.00%	27.27%	28.73%	40.00%	63.64%	23.64%	52.00%	36.36%	15.64%
0.5	60.00%	45.45%	14.55%	52.00%	63.64%	11.64%	60.00%	45.45%	14.55%	52.00%	63.64%	11.64%	64.00%	36.36%	27.64%
1	60.00%	36.36%	23.64%	52.00%	54.55%	2.55%	52.00%	54.55%	2.55%	40.00%	81.82%	41.82%	68.00%	18.18%	49.82%
1.5	64.00%	81.82%	17.82%	76.00%	54.55%	21.45%	68.00%	72.73%	4.73%	72.00%	63.64%	8.36%	76.00%	54.55%	21.45%
2	72.00%	72.73%	0.73%	80.00%	54.55%	25.45%	68.00%	81.82%	13.82%	72.00%	72.73%	0.73%	84.00%	45.45%	38.55%

Table 3b. Hit accuracy (%) for *oriC* prediction across five stratified datasets using prediction intervals expanded to twice the window size (2W). Results are shown separately for training and testing subsets, along with the difference between them.

Win- dow size (%)	Dataset 1			Dataset 2			Dataset 3			Dataset 4			Dataset 5		
	Train- ing	Test	Absolute gap (pp)	Train- ing	Test	Absolute gap (pp)	Train- ing	Test	Absolute gap (pp)	Train- ing	Test	Absolute gap (pp)	Train- ing	Test	Absolute gap (pp)
0.1	56.00%	90.91%	34.91%	64.00%	72.73%	8.73%	80.00%	36.36%	43.64%	60.00%	81.82%	21.82%	72.00%	54.55%	17.45%
0.5	76.00%	90.91%	14.91%	80.00%	81.82%	1.82%	88.00%	63.64%	24.36%	76.00%	90.91%	14.91%	84.00%	72.73%	11.27%
1	88.00%	90.91%	2.91%	92.00%	81.82%	10.18%	88.00%	90.91%	2.91%	88.00%	90.91%	2.91%	96.00%	72.73%	23.27%
1.5	88.00%	90.91%	2.91%	92.00%	81.82%	10.18%	88.00%	90.91%	2.91%	88.00%	90.91%	2.91%	96.00%	72.73%	23.27%
2	88.00%	90.91%	2.91%	92.00%	81.82%	10.18%	88.00%	90.91%	2.91%	88.00%	90.91%	2.91%	96.00%	72.73%	23.27%

3.2. *ForiC Comparison with Existing Models*

3.2.1. *Comparison with GC Skew and CGC Skew Baselines*

To ensure fair comparison, all methods were evaluated under a unified framework by representing their outputs as prediction intervals of matched length and assessing them using identical scoring criteria. This avoids bias toward methods producing broader predictions and enables direct comparison of detection and localization performance.

Further analysis revealed that ForiC generally outperformed the baseline methods, namely the GC and CGC skew analyses. Table 4 presents a comparative analysis of ForiC against these methods on the testing dataset. For fair comparison, GC- and CGC-based predictions were evaluated using the same window sizes as the final ForiC configuration and comparable interval-length settings. For each method, the primary candidate window was identified from the corresponding skew signal, and the final prediction interval was defined either as a single window (1W) or as an interval of twice the window size (2W). In the extended evaluation (2W), ForiC used symmetric expansion around its predicted center, whereas GC and CGC used adjacent two-window intervals derived from their respective skew extrema. This avoids bias due to unequal interval size while preserving the native prediction logic of each method.

Under these conditions, the GC method demonstrated a maximum accuracy of 72.73%, whereas CGC showed substantially lower performance (Table 4, [Additional file 4](#)). In contrast, ForiC achieved a higher accuracy of 90.91%. In addition, ForiC demonstrated better localization, as reflected by lower center-distance values compared with GC and CGC methods. Notably, CGC predictions showed very large localization errors, with a median center distance of 896.33 kb, indicating substantial displacement from validated oriC regions. This likely reflects a known limitation of CGC skew, in which the global maximum may occur far from oriC in genomes with asymmetric compositional patterns or irregular skew profiles.

A representative example further illustrates these differences. Although the final evaluation is based on the expanded-interval configuration, this example demonstrates localization capability using a single-window prediction. Using a window size of 1% and a single-window prediction interval, the genome CP006262 (length: 5,402,276 bp) was analyzed. The experimentally validated oriC region is located between positions 4,544,042 and 4,544,273. The GC method predicted oriC between positions 4,483,826 and 4,537,848, whereas the CGC method generated a prediction substantially displaced from the annotated oriC region (3,241,320–3,295,342). In contrast, ForiC localized the oriC region within the interval 4,537,848–4,591,870, demonstrating higher positional precision.

Under the evaluated framework, ForiC showed higher hit accuracy and lower center-distance values than GC and CGC methods.

3.2.2. *Comparison with OriFinder*

In addition to intrinsic skew-based baselines, ForiC was compared with OriFinder 2022 (Additional file 4), a widely used oriC prediction tool that integrates compositional asymmetry, motif detection, and homology-based signals within a ranking framework for putative oriC regions.

For fair comparison, all OriFinder predictions were converted into fixed-length evaluation windows centered on each reported candidate, matching the window size used for ForiC. This removes biases arising from differences in predicted interval lengths and enables direct comparison of localization metrics.

When two OriFinder candidate regions were reported, each candidate was evaluated independently, and a prediction was considered correct if at least one normalized window overlapped the validated oriC region (Additional file 4).

OriFinder achieved 100% accuracy under the candidate-based criterion, meaning that at least one reported candidate overlapped the validated oriC region in all analyzed genomes. However, multiple candidate regions were returned in 54.55% of cases, whereas only 45.45% of genomes received a single prediction. In contrast,

Table 4. Comparison of ForiC, GC, and CGC methods for oriC prediction on the test subset of Dataset 3. Results are reported for both single-window (1W) and symmetrically expanded two-window (2W) prediction intervals using a window size of 1% of the genome length. For GC and CGC methods, the primary candidate window was identified based on the corresponding skew signal, and prediction intervals were constructed using the same strategy as for ForiC. Metrics include hit accuracy (%), median center distance (kb), and Intersection over Union (IoU).

Method	Prediction interval length of 1 window (1W)			Prediction interval length of 2 windows (2W)		
	Accuracy (%)	Median center distance (kb)	Mean IoU	Accuracy (%)	Median center distance (kb)	Mean IoU
ForiC	54.55%	20.70	0.0050	90.91%	20.70	0.0035
GC	27.27%	27.36	0.0010	72.73%	27.36	0.0034
CGC	9.09%	896.33	0.0020	18.18%	896.33	0.0018

Table 5. Comparison of ForiC and OriFinder performance on the test subset of Dataset 3. Results are reported for both single-window (1W) and symmetrically expanded two-window (2W) prediction intervals, using a window size corresponding to 1% of the genome length. Accuracy is defined as the overlap between predicted and experimentally validated oriC regions. For OriFinder, accuracy is calculated using a candidate-based criterion (i.e., at least one predicted candidate overlaps the validated region). For distance-based metrics, the minimum distance among candidate predictions was used. All interval-based metrics were computed using normalized prediction windows of identical size to ensure comparability.

Method	Prediction interval length of 1 window (1W)			Prediction interval length of 2 windows (2W)		
	Accuracy (%)	Median center distance (kb)	Mean IoU	Accuracy (%)	Median center distance (kb)	Mean IoU
ForiC	54.55%	20.70	0.0050	90.91%	20.70	0.0035
OriFinder	100%	0.04	0.0078	100%	0.04	0.0039

ForiC consistently generated one prediction per genome and achieved 54.55% accuracy in the single-window (1W) setting and 90.91% under the expanded two-window (2W) criterion. OriFinder also showed markedly lower median center distance (0.04 kb versus 20.70 kb), indicating substantially higher positional precision (Table 5).

OriFinder also achieved higher IoU values in the single-window setting, although this difference decreased after interval expansion. Overall, the two methods reflect different practical trade-offs: OriFinder provides strong candidate recovery and high positional precision but depends on external resources (e.g., BLAST searches) and may return multiple predicted regions, whereas ForiC provides a single

intrinsically derived prediction without requiring prior annotation or database searches.

To further compare the behavior of both methods outside the benchmark dataset, an additional genome (AE000511, *Helicobacter pylori* 26695) was analyzed. This genome was not part of the training or evaluation datasets and does not have a precisely annotated oriC region in NCBI records. Both methods produced broadly similar predicted oriC locations, with interval centers at 1,642,783 for ForiC and 1,609,074 for OriFinder, indicating consistency in identifying candidate regions. However, substantial differences were observed in end-to-end execution time. The ForiC prediction was completed locally in 0.75 seconds on a standard workstation (Intel Core i7, 2.2 GHz, 8 GB RAM, SSD), whereas OriFinder, executed through its

public web interface, required approximately 1,680 seconds (28 minutes).

This difference likely reflects the more complex OriFinder workflow, which includes external searches such as BLAST-based homology detection and motif analysis, together with web-server execution overhead. In contrast, ForiC performs all computations directly on the input sequence without external dependencies.

Although limited to a single external genome, this example suggests that ForiC may be useful for rapid screening and large-scale analyses where lightweight execution is advantageous.

3.2.3. Robustness Under Local Sequence Perturbation

To further assess the robustness of the methods, selected genomes from the dataset were modified by removing nucleotide regions encompassing the experimentally validated oriC locations. The selected genomes include representatives from both the primary dataset (experimentally validated oriC annotations) and the extended dataset, allowing evaluation under both well-characterized and less certain annotation conditions. The specified subset of genomes was analyzed to illustrate different prediction behaviors under this perturbation.

The results (Table 6) indicate that ForiC maintains stable predictions even when the original oriC signal is partially removed, typically shifting the predicted region toward nearby compositional features. In contrast, OriFinder exhibits increased variability, often producing a larger number of candidate regions or shifting predictions to alternative high-scoring locations. In some cases, the number of reported candidates increased substantially, reflecting a stronger dependence on specific sequence features and external signals. While the table focuses on prediction stability and candidate variability, a complete overview of predictions for both original and modified sequences is available in [Additional file 5](#).

For genomes CP026681 and AE015450, ForiC predicted oriC-centered intervals at coordinate 0 in both original and modified sequences, consistent with the circular genome

representation in which coordinate 0 corresponds to the sequence boundary (wrap-around origin).

These findings suggest that ForiC appeared more stable in these selected perturbation examples to local disruptions in oriC-associated signals, while OriFinder's predictions are more sensitive to the presence of specific functional elements and homology-based cues.

3.3. Exploratory Consistency Analysis on the Extended Dataset

To further examine method behavior on genomes lacking consistently validated oriC annotations, an exploratory analysis was performed on the extended dataset. Because reliable reference coordinates were not uniformly available, formal accuracy assessment was not possible. Instead, positional agreement and prediction consistency across methods were examined.

In this analysis, ForiC was compared with GC skew, cumulative GC skew (CGC), and OriFinder to examine their behavior under less certain annotation conditions. The results are summarized in Table 7.

OriFinder demonstrated the highest level of agreement with ForiC, achieving 100% agreement and a very low mean distance of 3.26 kb, indicating that its predicted oriC locations are consistently close to those identified by ForiC. However, it produced an average of 1.73 candidate regions per genome, reflecting ambiguity in its predictions.

GC skew exhibited a relatively high agreement rate (80%), but with a substantially larger mean distance (202.65 kb), indicating the presence of significant outliers and reduced stability. CGC skew showed lower agreement (26.67%) and moderate distances, suggesting weaker consistency with ForiC predictions.

These exploratory results indicate that OriFinder and GC predictions were often positioned near ForiC predictions, whereas CGC showed lower positional consistency. Because experimentally validated references were unavailable for this dataset, these findings should be interpreted descriptively rather than as comparative accuracy rankings.

Table 6. Robustness analysis of ForiC and OriFinder predictions on selected genomes from the primary and extended datasets. Modified sequences were generated by removing genomic regions encompassing the experimentally validated oriC locations. Reported values correspond to the centers of predicted oriC regions, allowing direct comparison of prediction stability. While ForiC maintains relatively stable predictions with minor shifts in location, OriFinder exhibits increased variability, often producing a higher number of candidate regions after modification, reflecting its sensitivity to local sequence features and external signals.

Genome	Dataset	Removed region (bp)	ForiC center (original)	ForiC center (modified)	OriFinder candidates (original)	OriFinder candidates (modified)
CP006027	Primary	4,600,000-4,700,000	4,719,832	3,044,508	2	1
CP009273	Primary	3,870,000-3,940,000	3,959,847	3,899,997	2	3
CP026681	Extended	0-3,000	0	0	1	21
AE015450	Extended	0-7,000	0	0	2	4

Table 7. Comparative analysis of ForiC, GC skew, CGC skew, and OriFinder on the extended dataset. As experimentally validated oriC annotations are not consistently available, the evaluation is based on positional agreement relative to ForiC predictions. Mean and median distances represent circular distances between the centers of predicted oriC regions and the corresponding ForiC prediction centers. Agreement is defined as the percentage of predictions located within a window corresponding to 1% of the genome length centered at the ForiC prediction. For OriFinder, the minimum distance among candidate predictions was used, and the average number of candidates per genome is reported.

Metric	GC	CGC	OriFinder
Mean distance to ForiC (kb)	202.65	100.08	3.26
Median distance to ForiC (kb)	4.99	41.12	3.56
Agreement with ForiC (%)	80.00	26.67	100.00
Mean number of candidates	1	1	1.73

4. DISCUSSION

The performance metrics obtained during method development and testing indicate that ForiC achieves higher interval-based hit accuracy than GC and CGC skew baselines on the evaluated datasets. The method achieved a testing accuracy of 90.91%, exceeding that of the baseline approaches. Combined with comparable computational efficiency, these results suggest that the proposed method showed encouraging performance on held-out subsets and highlight the importance of integrating multiple genomic characteristics in oriC prediction.

Despite the high hit accuracy achieved, a key challenge remains in balancing interval size and positional precision. The improved performance observed with two-window reporting intervals indicates that the method can tolerate moderate positional uncertainty, which may be advantageous in practical screening applications. This improvement is also consistent with cases in which the oriC region spans neighboring sliding-window boundaries rather than being fully contained within a single window. However, the reported prediction intervals are intentionally designed for regional localization rather than exact base-pair delineation of oriC boundaries. The observed median center distance (20.70 kb) indicates that predicted oriC locations are typically placed within the reported interval,

supporting the stability of the approach. This reflects the design of the method, which prioritizes reliable regional localization of oriC-containing regions over nucleotide-level boundary prediction. Accordingly, ForiC is best viewed as a screening-oriented interval-localization framework that complements, rather than replaces, higher-resolution reference-assisted approaches. This interpretation is further supported by performance trends across different window sizes (Figure 3), where hit accuracy reaches a plateau beyond 1% of the genome length, while no consistent improvement in localization is observed. These results suggest diminishing returns with increasing window size, indicating that further interval expansion is unlikely to provide substantial benefit and may reduce spatial resolution.

While the previous analysis highlights the trade-off between interval size and positional precision, it is also important to consider how the proposed method compares with reference-based approaches such as OriFinder. While OriFinder demonstrated high positional precision and strong agreement with ForiC predictions, it frequently reported multiple candidate regions, requiring additional selection to identify the most likely oriC location. In contrast, ForiC consistently produced a single prediction per genome, providing a clear and directly interpretable result. This reflects a trade-off between positional resolution and decisiveness, where ForiC offers a streamlined and unambiguous prediction framework.

Performance varied across the taxonomically stratified benchmark subsets, which may reflect differences in genomic composition across bacterial lineages. Broader validation across additional phyla and larger balanced datasets is therefore required before stronger generalizability claims can be made.

The proposed method offers several practical advantages for computational genomics. In addition to improving upon simple compositional approaches such as GC and CGC skews, ForiC provides a computationally efficient and self-contained framework that does not rely on external databases or prior annotation. While reference-based tools such as OriFinder can achieve higher positional precision in certain

cases, this is partly attributable to their use of external biological resources and functional annotations. In contrast, ForiC focuses on identifying oriC-containing regions using intrinsic genomic signals, including compositional asymmetry and structural transitions reflected by inflection points. These features are consistent with known characteristics of bacterial replication origin regions, which are often associated with shifts in nucleotide composition and replication-related genome organization. The ability of ForiC to detect such signals across diverse genomes suggests that the method may capture biologically meaningful patterns related to DNA replication, although the precise role of individual features, particularly inflection points, requires further investigation.

Additional analyses on the extended dataset provide exploratory evidence of positional consistency across methods. Despite the absence of consistently validated oriC annotations, ForiC predictions showed general positional agreement with both compositional methods and OriFinder, indicating that the method captures biologically relevant signals beyond the training dataset.

Furthermore, robustness experiments involving the removal of oriC-associated regions suggested that ForiC predictions were comparatively stable in the selected perturbation experiments. These findings indicate that ForiC is less sensitive to local disruptions in sequence features and may provide more consistent predictions under uncertain or incomplete annotation conditions.

Nevertheless, several limitations should be acknowledged. The relatively modest number of genomes used in this study may limit the generalizability of the findings, highlighting the need for larger and more diverse datasets in future work. In addition, while ForiC effectively identifies oriC-containing regions, further improvements are needed to refine positional precision without compromising computational efficiency. The preliminary screening stage could also be extended in future work by incorporating sequence descriptors that account for nucleotide-order information. The proposed method relies on genomic features such as inflection points, whose biological interpretation and consistency across

different bacterial groups warrant further investigation. While these features contribute to improved prediction performance, their role in oriC localization is not yet fully understood, and the isolated contribution of the inflection-point feature was not separately evaluated in the present study. Future work should explore the relationship between inflection point patterns and underlying replication mechanisms, assess their independent predictive value, and evaluate their stability across larger and more diverse genomic datasets.

In conclusion, ForiC provides a lightweight intrinsic framework for oriC-region prediction that complements existing reference-assisted methods and may be particularly useful for rapid large-scale bacterial genome screening.

5. ACKNOWLEDGMENTS, STATEMENTS AND DECLARATIONS

5.1. Acknowledgments

I would like to thank my mentor, Dr Nenad Mitić, for reviewing the manuscript and providing valuable suggestions that significantly improved its quality.

5.2. Author Contributions

The author is solely responsible for all aspects of this manuscript. This includes conceptualization, methodology, formal analysis, investigation, data curation, writing – original draft, writing – review & editing, visualization, and project administration.

5.3. Data Availability Statement

The code developed for the method, which was used to produce the results presented in this study, is freely available at the following link: <http://bioinfo.matf.bg.ac.rs/home/downloads.waf?cat=Software&project=ForiC>.

The application itself is publicly accessible at <http://markostojicevic.com/foric/>.

The data are provided within the manuscript or supplementary information files. Additionally, the data and code can be obtained from the author upon reasonable request via email.

5.4. Funding

No funds, grants, or other support were received for conducting this study.

5.5. Competing Interests

The author(s) declare that they have no competing interests.

5.6. Ethics Approval and Consent to Participate

Not applicable.

5.7. Consent for Publication

Not applicable.

6. REFERENCES

- [1] S. P. Bell and K. Labib, "Chromosome Duplication in *Saccharomyces cerevisiae*," *Genetics*, vol. 203, no. 3, pp. 1027–1067, Jul. 2016, doi: 10.1534/genetics.115.186452.
- [2] C. Lou *et al.*, "sefOri: selecting the best-engineered sequence features to predict DNA replication origins," *Bioinformatics*, vol. 36, no. 1, pp. 49–55, Jan. 2020, doi: 10.1093/bioinformatics/btz506.
- [3] M. O'Donnell, L. Langston, and B. Stillman, "Principles and Concepts of DNA Replication in Bacteria, Archaea, and Eukarya," *Cold Spring Harb. Perspect. Biol.*, vol. 5, no. 7, pp. a010108–a010108, Jul. 2013, doi: 10.1101/cshperspect.a010108.
- [4] N. P. Robinson and S. D. Bell, "Origins of DNA replication in the three domains of life," *FEBS J.*, vol. 272, no. 15, pp. 3757–3766, Aug. 2005, doi: 10.1111/j.1742-4658.2005.04768.x.
- [5] R. R. Spinks *et al.*, "DnaB helicase dynamics in bacterial DNA replication resolved by single-molecule studies," *Nucleic Acids Res.*, vol. 49, no. 12, pp. 6804–6816, Jul. 2021, doi: 10.1093/nar/gkab493.
- [6] M. Wolanski, R. Donczew, A. Zawilak-Pawlik, and J. Zakrzewska-Czerwińska, "oriC-encoded instructions for the initiation of bacterial chromosome replication," *Front. Microbiol.*, vol. 5, Jan. 2015, doi: 10.3389/fmicb.2014.00735.
- [7] F. Bleichert, M. R. Botchan, and J. M. Berger, "Mechanisms for initiating cellular DNA replication," *Science*, vol. 355, no. 6327, p. eaah6317, Feb. 2017, doi: 10.1126/science.aah6317.
- [8] T. T. Richardson, D. Stevens, S. Pellicciari, O. Harran, T. Sperlea, and H. Murray, "Identification of a basal system for unwinding a bacterial chromosome origin," *EMBO J.*, vol. 38, no. 15, p. e101649, Aug. 2019, doi: 10.15252/embj.2019101649.
- [9] P. Mackiewicz, "Where does bacterial replication start? Rules for predicting the oriC region," *Nucleic Acids Res.*, vol. 32, no. 13, pp. 3781–3791, Jul. 2004, doi: 10.1093/nar/gkh699.

- [10] J. R. Lobry, "Asymmetric substitution patterns in the two DNA strands of bacteria," *Mol. Biol. Evol.*, vol. 13, no. 5, pp. 660–665, May 1996, doi: 10.1093/oxfordjournals.molbev.a025626.
- [11] E. R. M. Tillier and R. A. Collins, "The Contributions of Replication Orientation, Gene Direction, and Signal Sequences to Base-Composition Asymmetries in Bacterial Genomes," *J. Mol. Evol.*, vol. 50, no. 3, pp. 249–257, Mar. 2000, doi: 10.1007/s002399910029.
- [12] A. Marín and X. Xia, "GC skew in protein-coding genes between the leading and lagging strands in bacterial genomes: New substitution models incorporating strand bias," *J. Theor. Biol.*, vol. 253, no. 3, pp. 508–513, Aug. 2008, doi: 10.1016/j.jtbi.2008.04.004.
- [13] J. Lu and S. L. Salzberg, "SkewIT: The Skew Index Test for large-scale GC Skew analysis of bacterial genomes," *PLOS Comput. Biol.*, vol. 16, no. 12, p. e1008439, Dec. 2020, doi: 10.1371/journal.pcbi.1008439.
- [14] M. J. McLean, K. H. Wolfe, and K. M. Devine, "Base Composition Skews, Replication Orientation, and Gene Orientation in 12 Prokaryote Genomes," *J. Mol. Evol.*, vol. 47, no. 6, pp. 691–696, Dec. 1998, doi: 10.1007/PL00006428.
- [15] K. Arakawa and M. Tomita, "The GC Skew Index: A Measure of Genomic Compositional Asymmetry and the Degree of Replicational Selection," *Evol. Bioinforma.*, vol. 3, p. 117693430700300006, Jan. 2007, doi: 10.1177/117693430700300006.
- [16] H. Luo, C.-L. Quan, C. Peng, and F. Gao, "Recent development of Ori-Finder system and DoriC database for microbial replication origins," *Brief. Bioinform.*, vol. 20, no. 4, pp. 1114–1124, Jul. 2019, doi: 10.1093/bib/bbx174.
- [17] F. Gao and C.-T. Zhang, "Ori-Finder: A web-based system for finding oriC s in unannotated bacterial genomes," *BMC Bioinformatics*, vol. 9, no. 1, p. 79, Dec. 2008, doi: 10.1186/1471-2105-9-79.
- [18] A. Grigoriev, "Analyzing genomes with cumulative skew diagrams," *Nucleic Acids Res.*, vol. 26, no. 10, pp. 2286–2290, May 1998, doi: 10.1093/nar/26.10.2286.
- [19] H.-Y. Ou, F.-B. Guo, and C.-T. Zhang, "Analysis of nucleotide distribution in the genome of *Streptomyces coelicolor* A3(2) using the Z curve method," *FEBS Lett.*, vol. 540, no. 1–3, pp. 188–194, Apr. 2003, doi: 10.1016/S0014-5793(03)00263-1.
- [20] R. Zhang and C.-T. Zhang, "A Brief Review: The Z-curve Theory and its Application in Genome Analysis," *Curr. Genomics*, vol. 15, no. 2, pp. 78–94, Apr. 2014, doi: 10.2174/1389202915999140328162433.
- [21] R. Zhang and C.-T. Zhang, "Single replication origin of the archaeon *Methanosarcina mazei* revealed by the Z curve method," *Biochem. Biophys. Res. Commun.*, vol. 297, no. 2, pp. 396–400, Sep. 2002, doi: 10.1016/S0006-291X(02)02214-3.
- [22] A. Zawilak, "Identification of a putative chromosomal replication origin from *Helicobacter pylori* and its interaction with the initiator protein DnaA," *Nucleic Acids Res.*, vol. 29, no. 11, pp. 2251–2259, Jun. 2001, doi: 10.1093/nar/29.11.2251.
- [23] N. V. Sernova and M. S. Gelfand, "Identification of replication origins in prokaryotic genomes," *Brief. Bioinform.*, vol. 9, no. 5, pp. 376–391, Sep. 2008, doi: 10.1093/bib/bbn031.
- [24] M.-J. Dong, H. Luo, and F. Gao, "DoriC 12.0: an updated database of replication origins in both complete and draft prokaryotic genomes," *Nucleic Acids Res.*, vol. 51, no. D1, pp. D117–D120, Jan. 2023, doi: 10.1093/nar/gkac964.
- [25] M.-J. Dong, H. Luo, and F. Gao, "Ori-Finder 2022: A Comprehensive Web Server for Prediction and Analysis of Bacterial Replication Origins," *Genomics Proteomics Bioinformatics*, vol. 20, no. 6, pp. 1207–1213, Dec. 2022, doi: 10.1016/j.gpb.2022.10.002.
- [26] Y. Li and F. Gao, "OriV-Finder: a comprehensive web server for bacterial plasmid replication origin analysis," *Nucleic Acids Res.*, vol. 53, no. W1, pp. W451–W456, Jul. 2025, doi: 10.1093/nar/gkaf341.
- [27] S. M. Azim, Md. R. Haque, and S. Shatabda, "OriC-ENS: A sequence-based ensemble classifier for predicting origin of replication in *S. cerevisiae*," *Comput. Biol. Chem.*, vol. 92, p. 107502, Jun. 2021, doi: 10.1016/j.compbiolchem.2021.107502.
- [28] Z. Van Meel and J. A. Baaijens, "ORCA: Predicting replication origins in circular prokaryotic chromosomes," Mar. 30, 2024, *Bioinformatics*. doi: 10.1101/2024.03.28.587133.
- [29] J. Stewart, *Calculus: concepts and contexts*, Fourth edition, Enhanced edition. Boston, MA, USA: Cengage, 2019.
- [30] D. Powers, "Evaluation: From Precision, Recall and F-Measure to ROC, Informedness, Markedness & Correlation," *J. Mach. Learn. Technol.*, vol. 2, pp. 37–63, 2011.
- [31] K. V. Mardia and P. E. Jupp, *Directional Statistics*, 1st ed. in Wiley Series in Probability and Statistics. Wiley, 1999. doi: 10.1002/9780470316979.
- [32] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The Pascal Visual Object Classes (VOC) Challenge," *Int. J. Comput. Vis.*, vol. 88, no. 2, pp. 303–338, Jun. 2010, doi: 10.1007/s11263-009-0275-4.
- [33] M. A. Rahman and Y. Wang, "Optimizing Intersection-Over-Union in Deep Neural Networks for Image Segmentation," in *Advances in Visual Computing*, vol. 10072, G. Bebis, R. Boyle, B. Parvin, D. Koracin, F. Porikli, S. Skaff, A. Entezari, J. Min, D. Iwai, A. Sadagic, C. Scheidegger, and T. Isenberg, Eds., in Lecture Notes in Computer Science, vol. 10072. Cham: Springer International Publishing, 2016, pp. 234–244. doi: 10.1007/978-3-319-50835-1_22.
- [34] P. Worning, L. J. Jensen, P. F. Hallin, H. Stærfeldt, and D. W. Ussery, "Origin of replication in circular prokaryotic chromosomes," *Environ. Microbiol.*, vol. 8, no. 2, pp. 353–361, Feb. 2006, doi: 10.1111/j.1462-2920.2005.00917.x.
- [35] E. P. C. Rocha, "The replication-related organization of bacterial genomes," *Microbiology*, vol. 150, no. 6, pp. 1609–1627, Jun. 2004, doi: 10.1099/mic.0.26974-0.