Delft University of Technology

ORCA: A tool for chromosomal origin of replication prediction in eubacteria

Author Zoya van Meel

Supervisor Dr. Jasmijn Baaijens

July 11, 2022

Abstract

The mechanisms of chromosomal DNA replication in eubacteria are well-understood and researched. DNA replication starts with unwinding of the DNA at the origin of replication (*oriC*). The proximity of genes on a sequence to the *oriC* plays a role in replication and transcription related processes [1]. This makes the origin a location of significant interest. However, due to difficulty of the procedures involved, the experimental determination and verification of chromosomal origins has only been done for 23 bacteria.

This paper proposes ORCA, Origin of Chromosomal Replication Assessment, a Python tool for the visualisation of nucleotide disparities and the prediction of the location of *oriC*s based on the analysis of nucleotide disparities, *dnaA*-box regions, and *dnaA* and *dnaN* gene positions. Other bacterial *oriC*-predicting tools have been proposed in the past, but none are available anymore. Due to unavailability of the some of the sequences, ORCA was tested on fifteen of the 23 experimentally verified *oriCs*. ORCA is able to predict the *oriC*s of these chromosomes with a precision of 100 % and a recall of 85.71 %.

It is hoped ORCA can be a useful *in silico* method to help researchers in experimental determination of more *oriC*s with its prediction and visualisation capabilities. ORCA is available on GitHub: `www.github.com/ZoyavanMeel/ORCA`.

1 Introduction

DNA replication is an important step in the bacterial cell cycle. It is the process through which bacteria duplicate their genetic material before they divide [2, 3]. The starting point for chromosomal replication in bacterial DNA is called the origin of replication (*oriC*) [4]. Most bacterial species have circular chromosomes whose replication starts at the *oriC* and progresses bidirectionally around the DNA and terminates at the *terC*-region [5]. Knowing where on the DNA the *oriC* is, can be valuable information to most biology-related fields. For example, the use of the location of the *oriC* has become more interesting, since it was found that the *oriC*-proximal distance of genes plays a significant role in the control of replication and transcription related processes [1, 6, 7]. *oriC* positions can also be used in other contexts, like in comparative gene location analyses across species as a consistent measuring point.

Unfortunately, experimental determination of the *oriC* is not often done. That is because finding one is not simple. There are multiple *in vivo* and *in vitro* methods for experimentally determining the *oriC* based on protein interaction. However these are usually used alongside *in silico* methods and each other, since they can be impractical and sometimes provide contradictory results. While *in vivo* methods provide native conditions for studying *oriC*-interacting proteins, they are restricted in their throughput and can lack quantitative assessment of the protein-*oriC* interaction. On the other hand, *in vitro* methods provide an easier way of quantifying the interaction of a protein with an *oriC*, but its results can be highly dependent on assay conditions. [8, 9, 10]

The most widely used *in silico* tool used to be Ori-Finder 1 [9, 11]. Ori-Finder 1 was an online tool which makes use of several skew-analyses and the location of certain genes on the chromosome to predict where the *oriC* could be [12]. There are two websites that host Ori-Finder 1, however both of these throw errors when trying to use the tool. A download of the software can be requested, however it fails to produce any output. A few other *oriC*-predicting tool have been developed, however these tools are usually meant for visualisation rather than position prediction and/or have been discontinued [13, 14, 15]. Tools like OriLoc provide plots that can help visualise characteristics of the given DNA, but it is only meant to be used in combination with experimental verification [15, 16]. The research group that created Ori-Finder 1 went on to make Ori-Finder 2 and 3 for the prediction of origins in archaea and *Saccharomyces cerevisiae* subspecies, respectively [17, 18]. Only Ori-Finder 3 is still functioning.

This paper proposes ORCA (Origin of Chromosomal Replication Assessment). ORCA is an *in silico* *oriC*-finding tool available on GitHub. The core package consists only of three scripts that make use of five popular well-maintained Python libraries: NumPy, SciPy, Scikit-learn, and Biopython, as well as Matplotlib for visualisation. Because it was written in Python-3.9, ORCA works on any operating system with minimal effort, making it easily accessible.

The aim of this project is to provide an easy-to-use, accessible, and high-throughput alternative to Ori-Finder 1 for predicting *oriC*s in circular bacterial chromosomes. ORCA's standard parameters can be easily adjusted for fine-tuned predictions on single organisms, but it can also be readily applied onto large datasets for mass-annotation. ORCA makes use of the same principles and approaches as Ori-Finder 1, namely: Z-curve analysis, GC-skew, and *dnaA*(-box) locations to predict *oriC*s [11, 19, 20]. ORCA extracts positions that are characteristic to *oriC*-regions from the Z-curve analysis. These positions are filtered and ranked based on their proximity to genes and protein targets related to chromosomal DNA replication.

ORCA can work with only an accession number provided. It will fetch the needed FASTA-file for parsing the chromosomal DNA directly from NCBI as well as a FASTA-file containing the sequences and locations of all annotated genes on the chromosome. These files can also be loaded manually. ORCA outputs a dictionary that can easily be used with the provided plotters to visualise the characteristics of the DNA. This output includes, the Z-curve, the GC-skew, the predicted

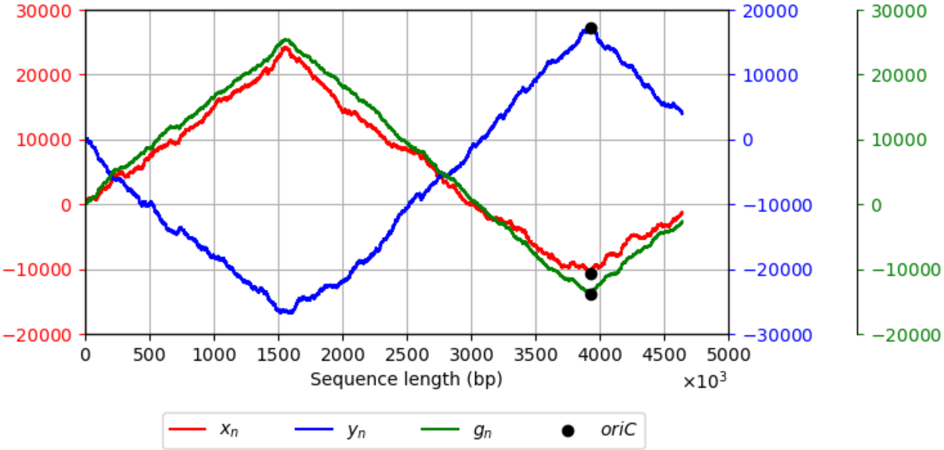

Figure 1: x_n, y_n, and g_n of the chromosome of *E. coli* K12 (Accession: NC_000913.3). The *oriC* is located at 3925744..3925975 bp [21, 22].

origin(s), and the confidence of ORCA its assessment of each origin.

2 Methods

2.1 Disparity Curves

Lobry [23] showed that minima in the GC-disparity (also called: GC-skew) show changes in polarity in the DNA that are characteristic to *oriC*s. The GC-disparity of a DNA sequence (g_n) is defined as the distribution of guanine over cytosine. Equation 1 shows the formula of the GC-disparity. N corresponds to the length of the DNA sequence in nucleotides. G_n and C_n are the cumulative occurrences of the corresponding bases between the first and n^{th} base (zero-indexed). For example, $G_{99} = 45$, means that guanine appeared 45 times in the first one hundred bases of the sequence. The direction of the GC-disparity curve changes from negative to positive at the origin of replication [24]. Frank and Lobry [15] used this concept to create OriLoc by analysing DNA sequences for these minima.

$$\begin{gathered} g_n = (G_n - C_n) \\ g_n \in [-N, N], n = 0, 1, 2, ..., N. \end{gathered} \tag{1}$$

Zhang and Zhang [25] showed that the GC-disparity curve of a DNA sequence is a special case of the Z-curve. The Z-curve is a unique three dimensional representation of a DNA sequence which is constructed by calculating three different distributions of two nucleotide bases over the others.

The Z-curve is made up of a series of nodes, P. Each node, $P_n = (x_n, y_n, z_n)$, is determined by the Z-Transform shown in Equation 2. Like in Equation 1, N corresponds to the length of the DNA sequence in nucleotides. A_n, T_n, C_n, and G_n are the cumulative occurrences of the corresponding bases between the first and n^{th} base (zero-indexed). Gao and Zhang [20] used the Z-curve's relation to the GC-disparity to show that the minima of the x-component and the maxima of the y-component of a sequence's Z-curve could also be used to identify *oriC*-like regions in the DNA sequence.

$$\begin{gathered} P_n = \begin{cases} x_n = (A_n + G_n) - (T_n + C_n) \\ y_n = (A_n + C_n) - (T_n + G_n) \\ z_n = (A_n + T_n) - (C_n + G_n) \end{cases} \\ x_n, y_n, z_n \in [-N, N], n = 0, 1, 2, ..., N. \end{gathered} \tag{2}$$

The definitions of each component of the Z-curve are based on the chemical properties of each base. They are defined as follows:

- x_n shows the distribution of puRine/pYrimidine (RY-disparity):

$$\text{Bases} \begin{cases} \text{Purine,} & \text{R} = \text{A, G,} \\ \text{Pyrimidine,} & \text{Y} = \text{C, T.} \end{cases}$$

- y_n shows the distribution of aMino/Keto (MK-disparity):

$$\text{Bases} \begin{cases} \text{Amino,} & \text{M} = \text{A, C,} \\ \text{Keto,} & \text{K} = \text{G, T.} \end{cases}$$

- z_n shows the distribution of Strong/Weak hydrogen bonds (SW-disparity):

$$\text{Bases} \begin{cases} \text{Weak,} & \text{W} = \text{A, T,} \\ \text{Strong,} & \text{S} = \text{C, G.} \end{cases}$$

ORCA uses the concepts of Z-curve and GC-disparity analysis in the initial part of the prediction

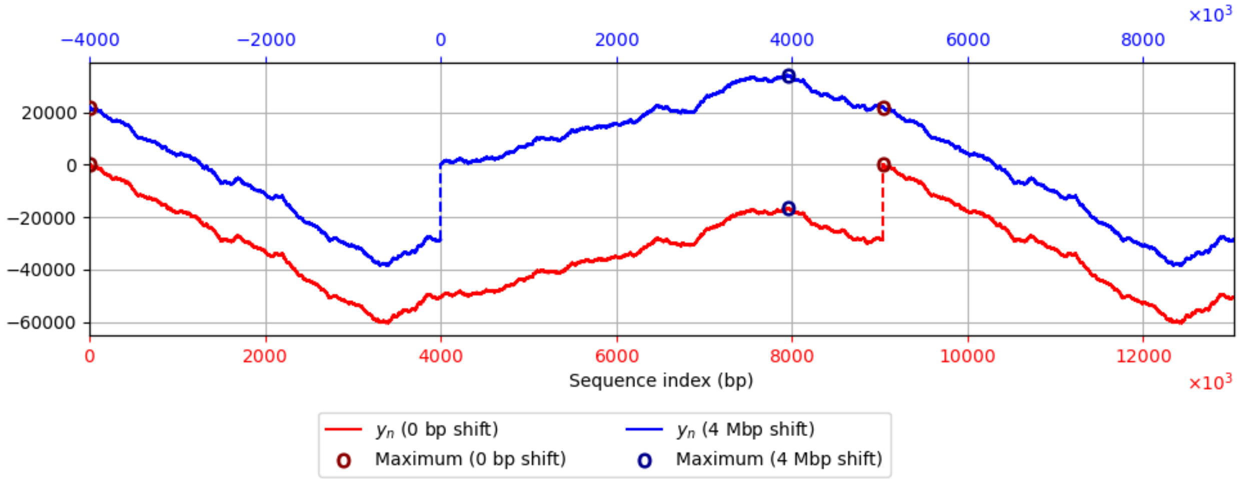

Figure 2: y_n (MK-disparity) of the chromosome of *N. koreensis* GR20-10 (Accession: NC_016609.1). In red as read from the sequence available in RefSeq and in blue with a 4 Mbp shift towards the 3'-side of the sequence. Where a sequence starts to get read, can influence its global extremes. The *oriC* of *N. koreensis* GR20-10 is not experimentally determined. Ori-Finder 1 predicts the origin to be at 7836170..7836739 bp, which corresponds to the blue maximum.

process (Sec. 2.3) to find contenders for *oriC* locations. As an example of this analysis, Figure 1 shows the RY-, MK-, and GC-disparity curves for the chromosome of *Escherichia coli* (*E. coli*) K12. The *oriC* of *E. coli* is located at 3925744..392597 bp [21, 22]. This corresponds perfectly with the minima of the RY- and GC-disparities and the maximum of the MK-disparity.

However, finding the *oriC* is not as simple as looking for global extremes. Due to the circular nature of bacterial chromosomes, it is up to the researcher which base is considered the starting point of the sequence.[1] This, in combination with the definition of the Z-curve as starting on $(0, 0, 0)$, makes it so a disparity curve can have a different global extreme based on which base gets chosen as the starting point of the sequence.

For example, the MK-disparity of *Niastella koreensis* (*N. koreensis*) GR20-10 is shown in Figure 2. This figure shows two versions of the disparity. One starting the calculation at the same base as the sequence in RefSeq; the other starting the calculation 4 Mbp later in the sequence. The second one is equally correct, because the chromosome is circular. However it leads to a different global maximum. Therefore, one has to account for local extremes when searching for potential *oriC*-locations.

The global extreme of one disparity curve also does not necessarily have to correspond to the global extreme of another. Even, in seemingly perfectly aligned extremes, like in *E. coli* K12 (Fig. 1), the extremes are not at the exact same indices. For this reason, there should be allowed for misalignment of the disparity curves. This is determined by a window as explained further in Section 2.3.1.

2.2 Gene & target locations

The position of certain genes and gene target sites on a DNA sequence can give information on the location of the *oriC*. This can be used to help narrow down the locations of interest found with the disparity curves. Namely, *dnaA* (helicase promotor) and *dnaN* (sliding clamp) are two genes that are often located in close proximity to the *oriC*, since they are heavily involved in the initiation of the replication cycle [12, 26, 27, 28]. Extracting the positions of these genes and comparing them to the locations of interest can help in identifying the *oriC*.

Additionally, the binding-site of the *dnaA* protein is of interest in the determination of the *oriC* location. The *dnaA* protein binds on specific target-sites called *dnaA*-boxes. The binding of *dnaA* to the *dnaA*-box promotes the unwinding of DNA with helicase *dnaB*. [4, 16, 28]

dnaA-boxes are species-specific [16, 26, 29]. However, *E. coli*'s most common *dnaA*-box is used a lot as consensus sequence for *dnaA*-boxes in all eubacteria: 5'-TTATNCACA-3' [30, 31]. In case the species-specific *dnaA*-box is not known or the consensus sequence is not present, some allow for one mismatch of the consensus [26, 32].

2.3 The prediction process

ORCA makes use of the Z-curve theory proposed by Zhang and Zhang [25], GC-disparity analysis, and gene (target) analysis for determining the *oriC* of a given chromosome. ORCA's source code is freely available on GitHub (Sec. 6). The sequence analysis process happens in four major steps.

[1] There is an informal rule that when a circular sequence is linearised, it is done so on the *oriC* [15, `http://pbil.univ-lyon1.fr/software/Oriloc/howto.html`]. However, this is not done often and when it is, it is not noted or annotated.

2.3.1 Z-curve analysis

ORCA uses a self-defined `Peak` class that makes the analysis of potential *oriC*s easier. Each `Peak` represents an index on the DNA sequence and has a window. A `Peak`'s window is the area surrounding the `Peak`'s index. For example, an index at $n = 5$ with a window of 4 bp, includes everything between $n = 2$ and $n = 7$. This window also accounts for the circularity of the sequence.

(I) SciPy is used to find initial peaks in the three relevant disparity curves [33]. Scipy's peak finding function is set to look for peaks around $\frac{1}{12}^{\text{th}}$ of the sequence length from each other. Twelve local minima and twelve local maxima along gives 24 peaks. Setting the minimal distance smaller lead to inconclusive origin selection. A larger minimal distance led to some origins not being considered in the analysis. The initial peaks are filtered based on the following two criteria:

  (i) Check if a peak is the global extreme of its own window. If not, reject the peak.

  (ii) Check if two peaks have intersecting windows. If so, reject the less extreme peak.

(II) Once this is done for each of the three relevant disparity curves, the accepted peaks get matched to each other. This means, for example, that if the indices of a peak found in x_n and a peak in y_n are close enough to each other (based on their windows), they get averaged into a new peak. This happens for every combination of curves and their peaks. If a peak has no matching peak in another curve, it gets rejected.

(III) The process in steps (I) and (II) get repeated with three varying sizes for the windows. These sizes are 1 %, 3 %, and 5 % of the chromosome's length. This way the windows are scaled with the size of the chromosome. This procedure accounts for uncertainty in choosing the optimal window size.

(IV) To match the peaks obtained from the three window sizes, a connected groups in an undirected graph problem is solved. An adjacency matrix is computed and with a depth-first-search the connected groups are found based on a threshold distance of the largest window size (5 % of the chromosome's length). In case the within distance of a group is larger than three times the threshold, that group gets reconnected using a threshold half of the original to split it up. The within distance is defined as the largest distance between two points in a group.

(V) The average index of each group is then calculated and these are the potential *oriC*s used for the rest of the analysis. A Z-score is defined as the fraction of peaks in a group over the total number of peaks. The Z-score for each potential *oriC* is calculated. This score is used as a measure for how confident ORCA is in a given index being the true *oriC*.

2.3.2 Gene location analysis

The gene location information of the *dnaA* and *dnaN* genes are parsed from a gene info file. The gene info file is the coding DNA sequence nucleotide FASTA for the chromosome and is a direct result of the PGAP. This file is obtained using Entrez Direct's EFetch with these parameters: `db='nuccore'`, `rettype='fasta_cds_na'`. Each peak obtained from the Z-curve analysis is ranked on the average distance to the average index of both genes. This ranking is called the G-score and is also used as a confidence measure like the Z-score.

2.3.3 *dnaA*-box analysis

All midpoints of 9-mers in the chromosome sequence that match the consensus box, 5'-TTATNCACA-3' perfectly are collected and each peak is checked on how many of these points are within its window. Ranking is based on the fraction of the amount of *dnaA*-boxes each peak contains over the total number of *dnaA*-boxes that were contained in a peak. This ranking is known as the D-score and is again used as a confidence measure for ORCA in the peak.

2.3.4 Score-based *oriC* selection

The three scores (Z, G, and D), as defined above, are all normalised between 0 and 1. They were used to make two versions of ORCA, mean-ORCA and SVC-ORCA. Mean-ORCA takes the mean of these scores for each potential origin and chooses the one with the highest value. The mean score is then used as a measure of confidence in the origin being the true origin.

SVC-ORCA uses the three scores as features in a Support Vector Classifier (SVC). The SVC predicts based on these scores whether a potential *oriC* is a true origin or whether it should be rejected. SVCs are maximum-margin machine learning models. This means they work by constructing a hyperplane between two groups in the given data that provides that maximum margin on both sides of the plane. The benefit of using SVCs is their high performance with relatively little tuning and their memory efficiency upon query. [34, 35]

The SVC is trained and tested using a 75:25 train:test-split. Its hyperparameters were tuned on precision (Eq. 3) using Scikit-learn's `GridsearchCV` with a 3-fold cross validation [36]. The decision function of the model is called on the three scores. The output of this function shows on which side of the hyperplane generated by the SVC the input is. One downside of using an SVC is that its results are non-probabilistic [37]. However, this output can be interpreted as the confidence the model -and consequently ORCA- has in its prediction [38]. The further this output is from zero, the more confident the model is in its prediction. Negative outputs, mean the scores do not belong to a true origin and *vice versa*. All parameters used for training the SVC can be found on GitHub. The link is available in Section 6.

2.4 Datasets

Seventeen[2] bacterial chromosomes have had their *oriC* experimentally determined and verified [11, 16, 39], but only two of these have their *oriC*(s) annotated in NCBI's Reference Sequence collection (RefSeq) [40]. Every entry in RefSeq gets annotated using their prokaryotic genome annotation pipeline (PGAP) [41]. Unfortunately, this pipeline does not perform any analyses for the location of the *oriC*, nor does it allow for any annotation outside of the PGAP. So, even if the *oriC* of a chromosome is known, it will not be annotated.

The only RefSeq entries that could have annotated origins are the chromosomes of organisms with 'reference' status and plasmids [39]. However, this does not mean that a 'reference' chromosome with known origins automatically gets annotated. For example, the *oriC*s of *Bacillus subtilis* 168, a widely used model bacterium, have been known for over twenty years, but are not annotated in RefSeq [22, 42, 43].

Luo *et al.* [11] compiled the predictions they made with Ori-Finder 1 in DoriC. DoriC is an online database with with the predicted *oriC*s of 9669 chromosomes and plasmids [44]. Extracting only the circular bacterial chromosomes leaves 7580 samples; 7288 of which are still available in RefSeq, excluding the seventeen verified ones. The 7288 DoriC entries are used as the first dataset; the seventeen chromosomes with verified origins are used as a second dataset. Even though the DoriC dataset consists entirely of predictions, for the purposes of this paper, they will be seen as ground truths.

All RefSeq entries are identified by an accession and version number. Accession numbers correspond to DNA sequences. Version numbers show which version an accession is on. DoriC does not show the version of the sequence that was used for all predictions in the database. An example of why version numbers are important, is that of *Pandoraea pnomenusa* (NC_023018). NC_023018.2 is differently linearised than NC_023018.1. DoriC uses version 1, but this is not noted. This means that if a correct prediction is made on version 2, but evaluated on version 1, it would be considered off by more than 40 % of the chromosome's length

2.5 Evaluation metrics

To evaluate the performance of ORCA, its predictions are compared against the 7288 predictions in DoriC and the seventeen *oriC*s that have been experimentally verified. ORCA's performance is evaluated on its precision (Eq. 3) and recall (Eq. 4) on both of these sets. A True Positive is defined as a prediction that has a maximum distance of 2.5 % of the length of the chromosome to the ground truth. DoriC entries will be seen as ground truths even though the database consists entirely of predictions.

Through DoriC and the verified dataset, ORCA's performance can be compared against Ori-Finder 1. Most of the predictions by Ori-Finder 1 on the verified data used outdated accessions, even for the time. Because the effect of the use of the wrong version could be significant, an overview of the discrepancies in versions of the experimentally verified dataset can be found in Table 4 in Appendix A. For the measurements on the verified set, it is possible to use the same versions as DoriC for all except two accessions: NC_002947.1 and NC_002971.1. These sequences are not available in RefSeq anymore and the current versions of these accessions are version 4. The reason only the first versions of these accessions were used, is not known.

$$\text{Precision} = \frac{\text{True Positive}}{\text{True Positive} + \text{False Positive}} \cdot 100\,\% \quad (3)$$

$$\text{Recall} = \frac{\text{True Positive}}{\text{True Positive} + \text{False Negative}} \cdot 100\,\% \quad (4)$$

3 Results

3.1 Feature Importance

ORCA produces three scores (features) which it uses to evaluate each of its computed potential origins. It is important to understand the effect of each score on the precision and recall of ORCA to determine a minimum. The importance of the Z-, G-, and D-scores in SVC-ORCA cannot be measured, since the SVC makes use of an approximation of a Radial Basis Function (RBF) called Random Kitchen Sinks as its kernel [45]. The coefficients and/or importance of a feature in a model trained with this kernel cannot be computed [36].

3.1.1 Mean-ORCA

To determine the effects of each score on mean-ORCA, each score was used as a confidence measure separately and plotted. Figure 3 shows the precision and recall of mean-ORCA against different decision boundaries, e.g. if mean-ORCA is set to only call an origin true if it had a minimum confidence of 50 %, then its precision and recall would be 90.75 % and 41.35 %, respectively.

[2] Luo *et al.* [11] found 23 chromosomes with verified *oriC*s, see Table A in Appendix A.

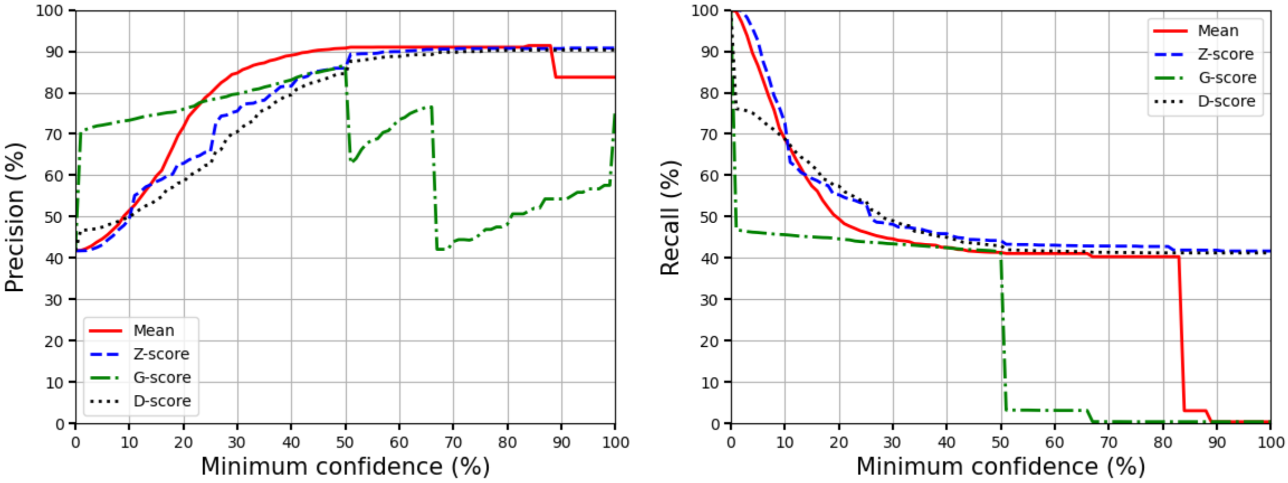

Figure 3: Precision (left) and recall (right) of mean-ORCA at different confidence cut-offs on the DoriC dataset. ‘Mean’ shows the confidence as a mean of the three scores. The others use only one score to measure confidence.

It can be seen that the Z- and D-scores have a very similar behaviour. The higher the Z- or D-score, the better the precision and the worse the recall. The G-score, however, behaves very differently. It serves as a good indicator of precision in the first minimum 50 %, however after that it drops twice. These drops can be seen in the recall graph as well. Even a minimum confidence of 1 % drops the recall to 46.86 %. The effects of the G-score can also be seen in the precision and recall when taking the mean of the scores. It is better for mean-ORCA not to use the G-score in its confidence score.

3.1.2 SVC-ORCA

The behaviour of the scores in mean-ORCA was taken into consideration when training the SVC. Three different SVC-models were trained that make use of variations on the G-score:

- Standard model: the same G-scores as used in mean-ORCA as a feature (3 features),
- No-G model: trained without the G-score as a feature (2 features),
- Separate-G model: the G-score is split in two, the GA-score (feature) for the distance of an *oriC* to *dnaA* and the GN-score (feature) for its distance to *dnaN* (4 features).

Table 1: Precision and recall of the three SVC models on the 25 % DoriC test set.

Model	Precision (%)	Recall (%)
Standard	93.64	90.05
No-G	93.08	87.93
Separate-G	92.20	83.89

The precision and recall of all three models can be found in Table 1. Interestingly, the No-G model outperformed the Separate-G model. It seems that it is better for the SVC to have no information on the genes’ location than it is for it to have more specific information available. As Table 1 shows, the Standard model performs best on both evaluation metrics, therefore it was chosen for further analysis.

3.2 Mean-ORCA vs. SVC-ORCA

Figures 4 and 5 were used to determine if it was better to use ORCA with or without a SVC model. Mean-ORCA does not take the G-score into consideration anymore when calculating its confidence in a potential origin. Figure 4 shows a precision against recall graph for both ORCA versions. This graph shows that even though mean-ORCA can have a higher recall than SVC-ORCA, it goes at the cost of precision.

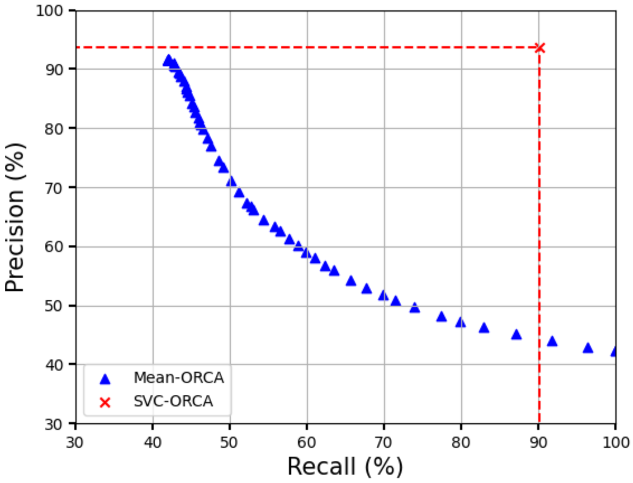

Figure 4: The precision against recall for both ORCAs. The points for mean-ORCA are the precision/recall at different decision boundaries between 0 and 100 % confidence.

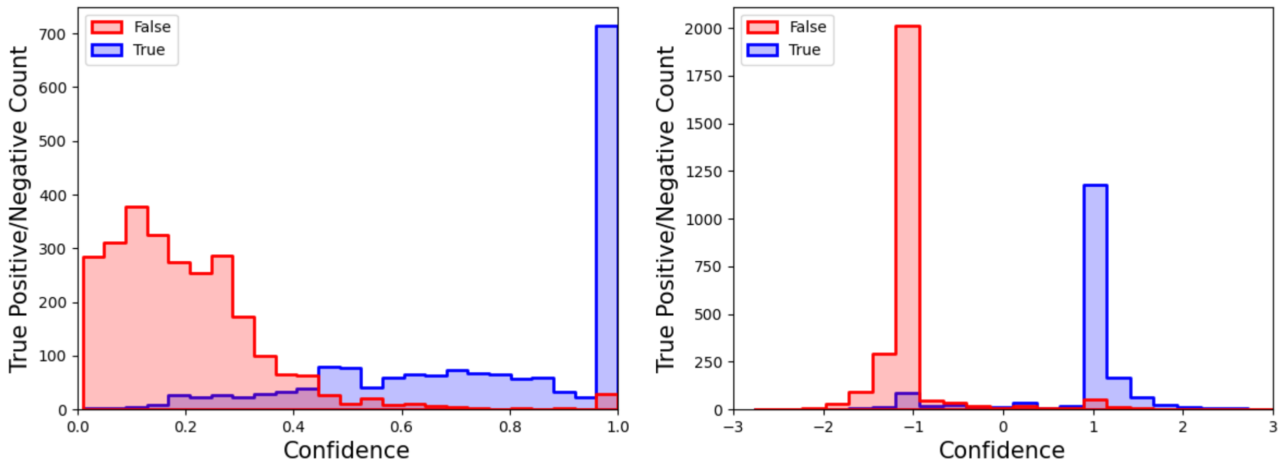

Figure 5: Histograms of the True Positive/Negative rate per confidence bin. Left: mean-ORCA, right: SVC-ORCA. Both ORCAs were tested on the 25 % DoriC testing set.

Figure 5 shows the distribution of True Positives/Negatives based on the confidence each model has in the prediction. SVC-ORCA has a confidence score centered on zero. Any sample with a confidence below zero gets classified as a false origin. The histogram shows that what SVC-ORCA classifies as a true/false origin is very much in line with the True Positives/Negatives.

Due to the nature of SVCs, the samples are separated from each other with minimal overlap. The samples that receive a confidence score in this interval, are samples that violate the soft-margin of the decision boundary. Any sample in this margin still gets classified based on which side of the decision boundary its on, but gets a much lower confidence score.

The confidence of mean-ORCA ranges from zero to one. This histogram, along with Figure 4 can help decide where the decision boundary for mean-ORCA should be, depending on whether precision or recall is being optimised. For these experiments, precision and recall were considered equally important. This would set mean-ORCA's decision boundary at 16.40 %, resulting in a precision and recall of 59.44 %. Since there is no point at which mean-ORCA has both a higher precision and recall than SVC-ORCA, SVC-ORCA was chosen to be used in the final version.

3.3 Experimental dataset

From the previous analyses it is clear that ORCA with the use of a SVC is the best choice. A new SVC was trained using the same parameters as before, but only keeping the seventeen experimentally verified accessions for testing. Due to the unavailability of two of the correct accessions, the precision and recall were measured twice. One set included NC_002947.4 and NC_002971.4 (17-set), the other did not (15-set). Due to the small size of the dataset and the large effect alternate versions can have on the results, it was deemed necessary to make this distinction.

The precision and recall on both sets are shown in Table 2. Both the precision and recall increase with the exclusion of these two samples. ORCA's recall is lower in both cases than when tested on the 25 % DoriC set, dropping from 90.05 % (Tab. 1) to 85.71 %. The precision and recall of Ori-Finder 1 are both 100 % on the 15- and 17-set.

Table 2: The precision and recall of ORCA on the experimental datasets.

Dataset	Precision (%)	Recall (%)
15-set	100.00	85.71
17-set	92.86	81.25

ORCA had two False Negatives on the 15-set: NC_003272.1 and NC_010546.1. Separate analysis of these two samples revealed that they both generated a large amount of potential origins of eleven and thirteen, respectively. On average three to four potential origins are found and judged by the SVC. The Z-score is a fraction of the total amount of potential origins, so when more origins are found, the maximum Z-score an origin can get, is lower than when fewer potential origins are found in the Z-curve analysis.

Mean-ORCA as also tested on the experimental datasets. One hundred confidence levels for were tested to determine the bounds of the precision and recall. The precision and recall extremes are relatively closely to each other. Mean-ORCA reaches maximum precision of 100 % on both datasets at a decision boundary of 23 % confidence this corresponds to a recall of 38.89 % and 35.00 % on the 17- and 15-set respectively. Its maximum recall of 100 % drops starting at 3 %. At a boundary of 3 %, its precision is 38.89 % and 35.00 % on the 17- and 15-set, respectively. Interestingly, the extremes of the precision and recall are exactly the same.

Table 3: Breakdown of time spent processing each part of the prediction. The middle column sample is the average of all entries in DoriC and does not correspond to any one sample.

Accession	**NC_018417.1**	**N/A**	**NC_010162.1**
Sequence length (bp)	157 543	3 719 519	13 033 779
Total genes	178	3406	9622
Disparity calculation (s)	0.63	22.59	53.97
Gene file parsing (s)	0.40	9.17	28.33
Other calculations (s)	0.43	4.1	12.67
Total time (s)	1.46	35.86	94.97

3.4 Processing time

One of ORCA's advantages is high throughput and ease of use. The time needed to analyse a single sequence is useful to know, especially when processing large datasets. Table 3 shows the breakdown of the analysis of the average DoriC sample. This analysis was done using an Intel®Core™i7-7600U CPU at 2.80 GHz. It is shown that the processing time increases with the amount of genes that have to be read through in the gene info file as well as well as the sequence length. The time it takes to read the average amount of genes is 9.17 seconds. The rest of the time is spent on the rest of the calculations.

Table 3 also shows processing time of the smallest and largest chromosomes in DoriC. The smallest and largest samples are not representative of the average bacterial chromosome, but they serve as an indication of the minimum and maximum processing time for a single sample. It can be seen that most of the processing time is spent on the calculation of the disparity curves. This operation is very costly and is the bottleneck for the process.

4 Discussion

4.1 Current performance

The performances of several versions of ORCA were tested and it was found that ORCA with a SVC performs best with a precision of 93.64 % and recall of 90.05 % on the DoriC testing set. Its precision increases on the experimental dataset (15-set) to 100.00 %, like Ori-Finder 1. However, ORCA's recall drops to 85.71 %. This drop in recall is likely due to the small size of the dataset. A single sample switching between True Positive and False Negative can sway the recall between 78.57 % and 92.86 %.

It was interesting to find that the No-G SVC model performed better on both precision and recall than the Separate-G SVC model (Tab. 1). Whether the *dnaA* and/or *dnaN* location is close to the *oriC* depends highly on the species, since these genes are not always close to each other either [16, 26]. The SVC in ORCA does not know this information, which likely leads to the model being less able to interpret when a high GA- or GN-score is more important.

It seems that the location data does help the performance of the SVC to some degree, because the Standard model still performed better than the No-G model, which did not use the gene locations in its predictions. As shown in Figure 3, the G-score does give some indication as to the precision. It is likely that the SVC in the Standard model was able to interpret this score correctly. Unfortunately, due to the kernel used, it is not possible to assess the importance of the G-score in the final model (Sec. 3.1), but only that it is useful for increasing precision and recall.

4.2 DoriC dataset

ORCA was trained on DoriC. There are two problems with this. One is the lack of full version notation for the used accessions. Almost all versions that have been notated are all version 1, regardless of the version available at the time of making Ori-Finder 1. Almost all, because there are two samples that do use a newer version. DoriC is easy to maintain if all entries are based on the same version, but it relies on RefSeq to support obsoleted versions of sequences indefinitely. RefSeq removes sequences that are too old, like with the two samples not used in the 15-set, but also regularly removes accessions entirely. This is why only 7305 of the 7580 circular chromosomes were still available for download.

The second problem is bias created by using DoriC as a training set. This experiment assumes the DoriC origins to be the same as the true origins. This was necessary due to the lack of experimental data, but this trains the SVC to be like Ori-Finder 1, rather than a true origin finder. It is important to realise the biases inherent to the datasets. A future experiment could be done, using only the experimentally verified *oriC*s as training data.

For these two reasons, it remains possible to use ORCA without a machine learning model. The SVC is simply an optional parameter that can be loaded in the origin-finding function. The Z-, G-, and D-score values are always available for interpretation by the user. The SVC is provided, but can easily be replaced by a user trained model, or not at all. The precision and recall scores of mean-ORCA showed that. depending on the user's priorities, use of ORCA without a model is also a viable option. This removes any biases towards DoriC origins.

4.3 Usability & Future improvements

While ORCA works relatively well, achieving the same precision on the 15-set as Ori-Finder 1, there are still optimisations possible. The two False Negatives on the 15-set are interesting examples. A beneficial extra feature for the SVC could be the number of other potential origins found in the sequence. Currently, each set of Z-, G-, and D-scores are seen as completely separate samples. Relating these scores to the total number of potential origins found in a sequence could help ORCA to better judge them.

Another DNA characteristic that could be considered in the future, is its bendability. DNA has to be able to bend and unravel easily for helicases and sliding clamps to be loaded on to it. It was found that the bendability of DNA is characteristically higher around the *oric* than it is elsewhere in the sequence [46, 47]. this could be of use in future versions of ORCA.

Furthermore, ORCA might be considering too many possible options for the locations of *oriC*s. All verified origins have been found in intergenic regions. These are non-coding regions in the DNA between two genes. Gao and Zhang [12] only looked for intergenic regions for locations of origins, that is why Ori-Finder 1 was able to produce definite boundaries for its predictions. It classifies the most likely intergenic region as the origin. Whether this holds true for most or all eubacteria requires more research.

One downside of using an SVC is that its results are non-probabilistic [37]. As shown in Figure 5, only a few sample get a classified with a confidence of $-0.9 < confidence < 0.9$. In a perfect SVC on data without outliers, this overlap does not occur, because no samples violate the margin surrounding the SVC hyperplane. This model is not perfect and neither is the data. Since it was shown the margin violation is minimal, future versions of ORCA could look to interpret the confidence score differently. This could be done by labeling the potential origins whose confidence falls in a certain interval as 'uncertain'. This would leave the interpretation of the origin up to the user. This would not work for large analyses, but could provide nuance when analysing a single organism.

SVCs are not the best learning machines, but they perform very well, despite their relative simplicity [34]. The SVC in used in this experiment is provided with ORCA, as are the datasets. This means it is easily possible to retrain a different model, if the user wishes.

The bulk of the processing time is spent on the disparity curve calculations. This calculation could be be sped up in several ways. One could lower the resolution of the curve by implementing a sliding window. This is one of the simplest and fastest ways to speed the process up. However, the precision would most likely suffer depending on how much resolution is lost.

In analysis of large datasets, it is possible to do the disparity calculations only once and store them. Depending on the dataset, this could require a significant amount of storage, but could be easily implemented. Another way is to run the analysis in parallel processes, but this does depend on the amount of processors available.

5 Conclusion

In conclusion, ORCA provides an effective and accessible alternative to Ori-Finder 1. ORCA's precision and recall on the samples that were experimentally verified showed that ORCA's performance is competitive with that of Ori-Finder 1. Since other *oriC*-finding tools are no longer available, ORCA provides a good solution and alternative to them.

ORCA's accessibility is apparent in its ease of use and modification. Python provides the perfect flexibility for a user to keep ORCA as-is or change it to fit their specific needs. ORCA can be employed in single organism analysis or in high-throughput pipelines with ease. A user only has to provide an accession to receive usable output for either exploratory research or final analysis.

ORCA is also useful solely as a visualisation tool. The built-in plotting functions allow for visualisation of single or multiple overlaying disparity curves. ORCA can still be improved upon, but its process allows for straightforward implementation of new steps. This way ORCA can be useful for many years to come.

6 Software & Data Availability

- ORCA: `www.github.com/ZoyavanMeel/ORCA`,
- DoriC 10.0 [44]: `www.tubic.org/doric`.

References

[1] J. Slager and J.-W. Veening, "Hard-wired control of bacterial processes by chromosomal gene location," *Trends in microbiology*, vol. 24, no. 10, pp. 788–800, 2016.

[2] A. Costa, I. V. Hood, and J. M. Berger, "Mechanisms for initiating cellular DNA replication," *Annual review of biochemistry*, vol. 82, p. 25, 2013.

[3] B. Alberts *et al.*, "Cells and genomes," in *Molecular biology of the cell*. Garland, New York, 2015, pp. 1–42.

[4] ——, "DNA replication, repair, and recombination," in *Molecular biology of the cell*. Garland, New York, 2015, pp. 237–298.

[5] ——, "DNA, chromosomes, and genomes," in *Molecular biology of the cell*. Garland, New York, 2015, p. 180.

[6] Y. Mileyko, R. I. Joh, and J. S. Weitz, "Small-scale copy number variation and large-scale changes in gene expression," *Proceedings of the National Academy of Sciences*, vol. 105, no. 43, pp. 16 659–16 664, 2008.

[7] A. Soler-Bistué, M. Timmermans, and D. Mazel, "The proximity of ribosomal protein genes to oriC enhances Vibrio cholerae fitness in the absence of multifork replication," *MBio*, vol. 8, no. 1, e00097–17, 2017.

[8] L. M. Hellman and M. G. Fried, "Electrophoretic mobility shift assay (EMSA) for detecting protein–nucleic acid interactions," *Nature protocols*, vol. 2, no. 8, pp. 1849–1861, 2007.

[9] C. Song, S. Zhang, and H. Huang, "Choosing a suitable method for the identification of replication origins in microbial genomes," *Frontiers in microbiology*, vol. 6, p. 1049, 2015.

[10] M. J. Buck and J. D. Lieb, "ChIP-chip: Considerations for the design, analysis, and application of genome-wide chromatin immunoprecipitation experiments," *Genomics*, vol. 83, no. 3, pp. 349–360, 2004.

[11] H. Luo, C.-L. Quan, C. Peng, and F. Gao, "Recent development of Ori-Finder system and DoriC database for microbial replication origins," *Briefings in Bioinformatics*, vol. 20, no. 4, pp. 1114–1124, 2019.

[12] F. Gao and C.-T. Zhang, "Ori-Finder: A web-based system for finding oriCs in unannotated bacterial genomes," *BMC bioinformatics*, vol. 9, no. 1, pp. 1–6, 2008.

[13] C.-A. H. Roten, P. Gamba, J.-L. Barblan, and D. Karamata, "Comparative genometrics (cg): A database dedicated to biometric comparisons of whole genomes," *Nucleic Acids Research*, vol. 30, no. 1, pp. 142–144, 2002.

[14] P. Worning, L. J. Jensen, P. F. Hallin, H.-H. Stærfeldt, and D. W. Ussery, "Origin of replication in circular prokaryotic chromosomes," *Environmental microbiology*, vol. 8, no. 2, pp. 353–361, 2006.

[15] A. Frank and J. Lobry, "Oriloc: Prediction of replication boundaries in unannotated bacterial chromosomes," *Bioinformatics*, vol. 16, no. 6, pp. 560–561, 2000.

[16] N. V. Sernova and M. S. Gelfand, "Identification of replication origins in prokaryotic genomes," *Briefings in Bioinformatics*, vol. 9, no. 5, pp. 376–391, 2008.

[17] H. Luo, C.-T. Zhang, and F. Gao, "Ori-Finder 2, an integrated tool to predict replication origins in the archaeal genomes," *Frontiers in microbiology*, vol. 5, p. 482, 2014.

[18] D. Wang, F.-L. Lai, and F. Gao, "Ori-Finder 3: A web server for genome-wide prediction of replication origins in saccharomyces cerevisiae," *Briefings in bioinformatics*, vol. 22, no. 3, bbaa182, 2021.

[19] R. Zhang and C.-T. Zhang, "A brief review: The Z-curve theory and its application in genome analysis," *Current genomics*, vol. 15, no. 2, pp. 78–94, 2014.

[20] F. Gao and C.-T. Zhang, "DoriC: A database of oriC regions in bacterial genomes," *Bioinformatics*, vol. 23, no. 14, pp. 1866–1867, 2007.

[21] A. Oka, K. Sugimoto, M. Takanami, and Y. Hirota, "Replication origin of the Escherichia coli K-12 chromosome: The size and structure of the minimum DNA segment carrying the information for autonomous replication," *Molecular and General Genetics MGG*, vol. 178, no. 1, pp. 9–20, 1980.

[22] NCBI Resource Coordinators, "Database resources of the national center for biotechnology information," *Nucleic acids research*, vol. 44, no. Database issue, p. D7, 2016.

[23] J. R. Lobry, "A simple vectorial representation of DNA sequences for the detection of replication origins in bacteria," *Biochimie*, vol. 78, no. 5, pp. 323–326, 1996.

[24] J. M. Freeman, T. N. Plasterer, T. F. Smith, and S. C. Mohr, "Patterns of genome organization in bacteria," *Science*, vol. 279, no. 5358, pp. 1827–1827, 1998.

[25] R. Zhang and C.-T. Zhang, "Z-curves, an intutive tool for visualizing and analyzing the dna sequences," *Journal of Biomolecular Structure and Dynamics*, vol. 11, no. 4, pp. 767–782, 1994.

[26] P. Mackiewicz, J. Zakrzewska-Czerwińska, A. Zawilak, M. R. Dudek, and S. Cebrat, "Where does bacterial replication start? Rules for predicting the oriC region," *Nucleic acids research*, vol. 32, no. 13, pp. 3781–3791, 2004.

[27] A. J. Oakley, P. Prosselkov, G. Wijffels, J. L. Beck, M. C. Wilce, and N. E. Dixon, "Flexibility revealed by the 1.85 Å crystal structure of the β sliding-clamp subunit of Escherichia coli DNA polymerase III," *Acta Crystallographica Section D: Biological Crystallography*, vol. 59, no. 7, pp. 1192–1199, 2003.

[28] J. L. Slonczewski and J. W. Foster, "Genomes and chromosomes," in *Microbiology: An evolving science: Third international student edition*. WW Norton & Company, 2013, pp. 237–275.

[29] B. Lafay, P. M. Sharp, A. T. Lloyd, M. J. McLean, K. M. Devine, and K. H. Wolfe, "Proteome composition and codon usage in spirochaetes: Species-specific and dna strand-specific mutational biases," *Nucleic acids research*, vol. 27, no. 7, pp. 1642–1649, 1999.

[30] S. Ishikawa *et al.*, "Distribution of stable DnaA-binding sites on the bacillus subtilis genome detected using a modified chip-chip method," *DNA research*, vol. 14, no. 4, pp. 155–168, 2007.

[31] R. S. Fuller, B. E. Funnell, and A. Kornberg, "The dnaA protein complex with the E. coli chromosomal replication origin (oriC) and other DNA sites," *Cell*, vol. 38, no. 3, pp. 889–900, 1984.

[32] M. Picardeau, J. R. Lobry, and B. J. Hinnebusch, "Physical mapping of an origin of bidirectional replication at the centre of the Borrelia burgdorferi linear chromosome," *Molecular microbiology*, vol. 32, no. 2, pp. 437–445, 1999.

[33] P. Virtanen *et al.*, "SciPy 1.0: Fundamental algorithms for scientific computing in Python," *Nature methods*, vol. 17, no. 3, pp. 261–272, 2020.

[34] D. Meyer, F. Leisch, and K. Hornik, "The support vector machine under test," *Neurocomputing*, vol. 55, no. 1-2, pp. 169–186, 2003.

[35] T. Joachims, "Making large-scale support vector machine learning practical," in *Advances in kernel methods: support vector learning.* MIT press, 1999, p. 169.

[36] F. Pedregosa *et al.*, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[37] J. Platt *et al.*, "Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods," *Advances in large margin classifiers*, vol. 10, no. 3, pp. 61–74, 1999.

[38] Scikit-learn developers, *Support vector machines*, 2022. [Online]. Available: `https://scikit-learn.org/stable/modules/svm.html#scores-probabilities` (visited on 07/05/2022).

[39] J. Kans, "Entrez direct: E-utilities on the UNIX command line," in *Entrez Programming Utilities Help [Internet]*, National Center for Biotechnology Information (US), 2022.

[40] N. A. O'Leary *et al.*, "Reference sequence (RefSeq) database at NCBI: Current status, taxonomic expansion, and functional annotation," *Nucleic acids research*, vol. 44, no. D1, pp. D733–D745, 2016.

[41] W. Li *et al.*, "RefSeq: Expanding the Prokaryotic Genome Annotation Pipeline reach with protein family model curation," *Nucleic acids research*, vol. 49, no. D1, pp. D1020–D1028, 2021.

[42] P. J. Lewis and J. Errington, "Direct evidence for active segregation of oriC regions of the Bacillus subtilis chromosome and co-localization with the Spo0J partitioning protein," *Molecular microbiology*, vol. 25, no. 5, pp. 945–954, 1997.

[43] J. Errington and L. T. van der Aart, "Microbe profile: Bacillus subtilis: Model organism for cellular development, and industrial workhorse," *Microbiology*, vol. 166, no. 5, p. 425, 2020.

[44] H. Luo and F. Gao, "DoriC 10.0: An updated database of replication origins in prokaryotic genomes including chromosomes and plasmids," *Nucleic acids research*, vol. 47, no. D1, pp. D74–D77, 2019.

[45] A. Rahimi and B. Recht, "Random features for large-scale kernel machines," *Advances in neural information processing systems*, vol. 20, 2007.

[46] W. Chen, P. Feng, and H. Lin, "Prediction of replication origins by calculating dna structural properties," *FEBS letters*, vol. 586, no. 6, pp. 934–938, 2012.

[47] F.-Y. Dao, H. Lv, F. Wang, and H. Ding, "Recent advances on the machine learning methods in identifying dna replication origins in eukaryotic genomics," *Frontiers in Genetics*, vol. 9, p. 613, 2018.

A Supplementary Material

Table 4: Overview of current accession versions of the sequences with experimentally verified *oriC*s against the ones used in DoriC. Version pre-DoriC shows which version was the standard before the development of Ori-Finder 1 (2007) if it differs from the version that was used in DoriC. The gray rows are the accessions that were used in the experimental dataset. *Genbank accession number.

Accession	**DoriC**	**Current**	**Version**	**Notes**
	Version	**Version**	**pre-DoriC**	
NC_000913	1	3	2 (2004)	
NC_000921	N/A	1		Mentioned on the website, but not in DoriC.
NC_000962	1	3	2 (2005)	
NC_000964	1	3		
NC_002696	1	2	2 (2001)	
NC_002947	1	4	2 (2002)	Version 1 not available anymore.
NC_002971	1	4	3 (2005)	Version 1 not available anymore.
NC_003047	1	1		
NC_003272	1	1		
NC_003869	1	1		
NC_003888	1	3	2 (2002)	Obsoleted. Do not use.
NC_005090	1	1		
NC_005363	1	1		
NC_006461	1	1		
NC_007575	N/A	1		Mentioned on the website, but not in DoriC.
NC_007604	1	1		Obsoleted. Do not use.
NC_007633	1	1		
NC_008255	1	1		
NC_009850	1	1		
NC_010546	1	1		
NC_011916	1	1		
NZ_CP007757	N/A	1		Mentioned on the website, but not in DoriC.
CP003904*	N/A	1		Mentioned on the website, but not in DoriC.