

1    **MitoChime: A Machine Learning Pipeline for**  
2            **Detecting PCR-Induced Chimeras in**  
3            **Mitochondrial Illumina Reads**

4                            A Special Project Proposal  
5                            Presented to  
6            the Faculty of the Division of Physical Sciences and Mathematics  
7                            College of Arts and Sciences  
8            University of the Philippines Visayas  
9                            Miagao, Iloilo

10                           In Partial Fulfillment  
11                           of the Requirements for the Degree of  
12            Bachelor of Science in Computer Science

13                           by

14                           Duranne Duran  
15                           Yvonne Lin  
16                           Daniella Pailden

17                           Adviser  
18                           Francis D. Dimzon, Ph.D.

19                           January 2, 2026

## Abstract

21 Next-generation sequencing (NGS) platforms have advanced research but re-  
22 main susceptible to artifacts such as PCR-induced chimeras that compromise  
23 mitochondrial genome assembly. These artificial hybrid sequences are prob-  
24 lematic for small, circular, and repetitive mitochondrial genomes, where they  
25 can generate fragmented contigs and false junctions. Existing detection tools,  
26 such as UCHIME, are optimized for amplicon-based microbial community ana-  
27 lysis and depend on reference databases or abundance assumptions unsuitable  
28 for organellar assembly. To address this gap, this study presents MitoChime,  
29 a machine learning pipeline for detecting PCR-induced chimeric reads in *Sar-*  
30 *dinella lemur* Illumina paired-end data without relying on external reference  
31 databases.

32 Using simulated datasets containing clean and chimeric reads, a feature  
33 set was extracted, combining alignment-based metrics (e.g., supplementary  
34 alignments, soft-clipping) with sequence-derived statistics (e.g., k-mer com-  
35 position, microhomology). A comparative evaluation of supervised learning  
36 models identified tree-based ensembles CatBoost and Gradient Boosting as top  
37 performers, achieving an F1-score of 0.77 and an ROC-AUC of 0.84 on held-  
38 out test data. Feature importance analysis highlighted soft-clipping and k-mer  
39 compositional shifts as the strongest predictors of chimerism, whereas micro-  
40 homology contributed minimally. Integrating MitoChime as a pre-assembly  
41 step can aid in streamlining mitochondrial reconstruction pipelines.

42 **Keywords:** Chimera detection, Mitochondrial genome,  
Assembly, Machine learning

# Contents

44	<b>1 Introduction</b>	<b>1</b>
45	1.1 Overview . . . . .	1
46	1.2 Problem Statement . . . . .	3
47	1.3 Research Objectives . . . . .	3
48	1.3.1 General Objective . . . . .	3
49	1.3.2 Specific Objectives . . . . .	4
50	1.4 Scope and Limitations of the Research . . . . .	4
51	1.5 Significance of the Research . . . . .	6
52	<b>2 Review of Related Literature</b>	<b>7</b>
53	2.1 The Mitochondrial Genome . . . . .	7
54	2.1.1 Mitochondrial Genome Assembly . . . . .	8

55	2.2	PCR Amplification and Chimera Formation . . . . .	9
56	2.3	Existing Traditional Approaches for Chimera Detection . . . . .	10
57	2.3.1	UCHIME . . . . .	11
58	2.3.2	UCHIME2 . . . . .	12
59	2.3.3	CATch . . . . .	13
60	2.3.4	ChimPipe . . . . .	14
61	2.4	Machine Learning Approaches for Chimera and Sequence Quality	
62		Detection . . . . .	15
63	2.4.1	Feature-Based Representations of Genomic Sequences . . .	15
64	2.5	Synthesis of Chimera Detection Approaches . . . . .	16
65	<b>3</b>	<b>Research Methodology</b>	<b>19</b>
66	3.1	Research Activities . . . . .	19
67	3.1.1	Data Collection . . . . .	20
68	3.1.2	Feature Extraction Pipeline . . . . .	23
69	3.1.3	Machine Learning Model Development . . . . .	26
70	3.1.4	Model Benchmarking, Hyperparameter Optimization, and	
71		Evaluation . . . . .	27
72	3.1.5	Feature Importance, Feature Selection, and Interpretation	29

73	3.1.6	Validation and Testing . . . . .	31
74	3.1.7	Documentation . . . . .	32
75	3.2	Calendar of Activities . . . . .	32
76	<b>4</b>	<b>Results and Discussion</b>	<b>34</b>
77	4.1	Descriptive Analysis of Features . . . . .	35
78	4.1.1	Summary Statistics Per Class . . . . .	35
79	4.1.2	Boxplots By Class . . . . .	37
80	4.2	Baseline Classification Performance . . . . .	39
81	4.3	Effect of Hyperparameter Tuning . . . . .	40
82	4.4	Detailed Evaluation of Representative Models . . . . .	42
83	4.4.1	Confusion Matrices and Error Patterns . . . . .	42
84	4.4.2	ROC and Precision–Recall Curves . . . . .	44
85	4.5	Feature Importance . . . . .	45
86	4.5.1	Permutation Importance of Individual Features . . . . .	45
87	4.5.2	Feature Family Importance . . . . .	47
88	4.6	Feature Selection . . . . .	49
89	4.6.1	Cumulative Importance Curve . . . . .	50

90	4.6.2	Performance Comparison Across Feature Sets . . . . .	50
91	4.6.3	Interpretation and Final Feature Set Choice . . . . .	52
92	4.7	Summary of Findings . . . . .	52
93	<b>A</b>	<b>Complete Per-Class Summary Statistics</b>	<b>53</b>
94	<b>B</b>	<b>Boxplots for All Numeric Features by Feature Family</b>	<b>56</b>
95	B.0.1	SA Structure (Supplementary Alignment and Segment Met-	
96		rics) . . . . .	56
97	B.0.2	Clipping-Based Features . . . . .	58
98	B.0.3	K-mer Features . . . . .	58
99	B.0.4	Microhomology Features . . . . .	59
100	B.0.5	Others . . . . .	59

# 101 List of Figures

102	3.1	Process diagram of the study workflow. . . . .	20
103	4.1	Boxplots of key features by class . . . . .	38
104	4.2	Test F1 of all baseline classifiers, showing that no single model	
105		clearly dominates and several achieve comparable performance. . .	40
106	4.3	Comparison of test F1 (left) and ROC–AUC (right) for baseline	
107		and tuned models. . . . .	41
108	4.4	Confusion matrices for the four representative models on the held-	
109		out test set. . . . .	44
110	4.5	ROC (left) and precision–recall (right) curves for the four represen-	
111		tative models on the held-out test set. . . . .	45
112	4.6	Permutation-based feature importance for four representative clas-	
113		sifiers. . . . .	47
114	4.7	Aggregated feature family importance across four models. . . . .	49

115	4.8	Cumulative importance curve of features sorted by importance. . .	50
116	4.9	Comparison of F1 and ROC–AUC for the full, top-4 selected, and	
117		no-microhomology feature set variants. . . . .	51
118	B.1	Boxplots of SA Structure features by class (1/2). . . . .	56
119	B.2	Boxplots of SA Structure features by class (2/2). . . . .	57
120	B.3	Boxplots of clipping-based features by class. . . . .	58
121	B.4	Boxplots of k-mer features by class. . . . .	58
122	B.5	Boxplots of microhomology features by class. . . . .	59
123	B.6	Boxplots of other numeric features by class. . . . .	59



# 124 List of Tables

125	2.1	Comparison of Chimera Detection Approaches and Tools . . . . .	17
126	3.1	Timetable of activities. . . . .	33
127	4.1	Summary statistics of selected key features by class. . . . .	37
128	4.2	Performance of baseline classifiers on the held-out test set. . . . .	40
129	4.3	Performance of tuned classifiers on the held-out test set. . . . .	41
130	4.4	Test set performance of three feature set variants using tuned Cat-	
131		Boost. . . . .	51
132	A.1	Complete per-class summary statistics for all extracted features. .	53

# Chapter 1

## Introduction

### 1.1 Overview

The rapid advancement of next-generation sequencing (NGS) technologies has transformed genomic research by enabling high-throughput and cost-effective DNA analysis (Metzker, 2010). Among current platforms, Illumina sequencing remains the most widely adopted, capable of producing millions of short reads that can be assembled into reference genomes or analyzed for genetic variation (Bentley et al., 2008; Glenn, 2011). Despite its high base-calling accuracy, Illumina sequencing is prone to artifacts introduced during library preparation, particularly polymerase chain reaction (PCR)-induced chimeras, which are artificial hybrid sequences that do not exist in the true genome (Judo, Wedel, & Wilson, 1998).

PCR chimeras form when incomplete extension products from one template

147 anneal to an unrelated DNA fragment and are extended, creating recombinant  
 148 reads (Qiu et al., 2001). In mitochondrial genome assembly, such artifacts are  
 149 especially problematic because the mitochondrial genome is small, circular, and  
 150 often repetitive (Boore, 1999; Cameron, 2014). Even a small number of chimeric  
 151 or misjoined reads can reduce assembly contiguity and introduce false junctions  
 152 during organelle genome reconstruction (Dierckxsens, Mardulyn, & Smits, 2017;  
 153 Hahn, Bachmann, & Chevreux, 2013; Jin et al., 2020). Existing assembly tools  
 154 such as GetOrganelle and MITObim assume that input reads are largely free of  
 155 such artifacts (Hahn et al., 2013; Jin et al., 2020). Consequently, undetected  
 156 chimeras may produce fragmented assemblies or misidentified organellar bound-  
 157 aries. To ensure accurate reconstruction of mitochondrial genomes, a reliable  
 158 method for detecting PCR-induced chimeras before assembly is essential.

159 This study focuses on mitochondrial sequencing data from the genus *Sar-*  
 160 *dinella*, a group of small pelagic fishes widely distributed in Philippine waters.  
 161 Among them, *Sardinella lemuru* (Bali sardinella) is one of the country’s most  
 162 abundant and economically important species, providing protein and livelihood  
 163 to coastal communities (Labrador, Agmata, Palermo, Ravago-Gotanco, & Pante,  
 164 2021; Willette, Bognot, Mutia, & Santos, 2011). Accurate mitochondrial as-  
 165 semblies are critical for understanding its population genetics, stock structure,  
 166 and evolutionary history. However, assembly pipelines often encounter errors or  
 167 fail to complete due to undetected chimeric reads. To address this gap, this  
 168 research introduces MitoChime, a machine learning pipeline designed to detect  
 169 PCR-induced chimeric reads using both alignment-based and sequence-derived  
 170 statistical features. The tool aims to provide bioinformatics laboratories, partic-  
 171 ularly the Philippine Genome Center Visayas (PGC Visayas), with an efficient

172 solution for improving mitochondrial genome reconstruction.

## 173 1.2 Problem Statement

174 Chimeric reads can distort assembly graphs and cause misassemblies, with par-  
175 ticularly severe effects in mitochondrial genomes (Boore, 1999; Cameron, 2014).  
176 Existing assembly pipelines such as GetOrganelle, MITObim, and NOVOPlasty  
177 assume that sequencing reads are free of such artifacts (Dierckxsens et al., 2017;  
178 Hahn et al., 2013; Jin et al., 2020). At PGC Visayas, several mitochondrial as-  
179 semblies have failed or yielded incomplete contigs despite sufficient coverage, sug-  
180 gesting that undetected chimeric reads compromise assembly reliability. Mean-  
181 while, existing chimera detection tools such as UCHIME and VSEARCH were  
182 developed primarily for amplicon-based community analysis and rely heavily on  
183 reference or taxonomic comparisons (Edgar, Haas, Clemente, Quince, & Knight,  
184 2011; Rognes, Flouri, Nichols, Quince, & Mahé, 2016). These approaches are un-  
185 suitable for single-species organellar data, where complete reference genomes are  
186 often unavailable.

## 187 1.3 Research Objectives

### 188 1.3.1 General Objective

189 This study aims to develop and evaluate a machine learning-based pipeline (Mi-  
190 toChime) that detects PCR-induced chimeric reads in *Sardinella lemuru* mito-

191 chondrial sequencing data in order to improve the quality and reliability of down-  
192 stream mitochondrial genome assemblies.

### 193 **1.3.2 Specific Objectives**

194 Specifically, the study aims to:

- 195 1. construct simulated *Sardinella lemuru* Illumina paired-end datasets contain-  
196 ing both clean and PCR-induced chimeric reads,
- 197 2. extract alignment-based and sequence-based features such as k-mer compo-  
198 sition, junction complexity, and split-alignment counts from both clean and  
199 chimeric reads,
- 200 3. train, validate, and compare supervised machine learning models for classi-  
201 fying reads as clean or chimeric,
- 202 4. determine feature importance and identify indicators of PCR-induced  
203 chimerism,
- 204 5. integrate the optimized classifier into a modular and interpretable pipeline  
205 deployable on standard computing environments at PGC Visayas.

## 206 **1.4 Scope and Limitations of the Research**

207 This study focuses solely on PCR-induced chimeric reads in *Sardinella lemuru*  
208 mitochondrial sequencing data, with the species choice guided by four consid-  
209 erations: (1) to limit interspecific variation in mitochondrial genome size, GC

210 content, and repetitive regions so that differences in read patterns can be at-  
211 tributed more directly to PCR-induced chimerism, (2) to align the analysis with  
212 relevant *S. lemuru* sequencing projects at PGC Visayas, (3) to take advantage of  
213 the availability of *S. lemuru* mitochondrial assemblies and raw datasets in public  
214 repositories such as the National Center for Biotechnology Information (NCBI),  
215 which facilitates reference selection and benchmarking, and (4) to develop a tool  
216 that directly supports local studies on *S. lemuru* population structure and fisheries  
217 management.

218 The study emphasizes **wgsim**-based simulations and selected empirical mito-  
219 chondrial datasets from *S. lemuru*. It excludes naturally occurring chimeras, nu-  
220 clear mitochondrial pseudogenes (NUMTs), and large-scale assembly rearrange-  
221 ments in nuclear genomes. Feature extraction is restricted to low-dimensional  
222 alignment and sequence statistics, such as k-mer frequency profiles, GC con-  
223 tent, soft and hard clipping metrics, and split-alignment counts rather than high-  
224 dimensional deep learning embeddings. This design keeps model behaviour inter-  
225 pretable and ensures that the pipeline can be run on standard workstations at  
226 PGC Visayas. Testing on long-read platforms (e.g., Nanopore, PacBio) and other  
227 taxa is outside the scope of this project.

228 Other limitations in this study include the following: simulations with vary-  
229 ing error rates were not performed, so the effect of different sequencing errors on  
230 model performance remains unexplored; alternative parameter settings, including  
231 k-mer lengths and microhomology window sizes, were not systematically tested,  
232 which could affect the sensitivity of both k-mer and microhomology feature de-  
233 tection; and the machine learning models rely on supervised training with labeled  
234 examples, which may limit their ability to detect novel or unexpected chimeric

235 patterns.

## 236 1.5 Significance of the Research

237 This research provides both methodological and practical contributions to mito-  
238 chondrial genomics and bioinformatics. First, MitoChime detects PCR-induced  
239 chimeric reads prior to genome assembly, with the goal of improving the con-  
240 tiguity and correctness of *Sardinella lemuru* mitochondrial assemblies. Second,  
241 it replaces informal manual curation with a documented workflow, improving au-  
242 tomation and reproducibility. Third, the pipeline is designed to run on computing  
243 infrastructures commonly available in regional laboratories, enabling routine use  
244 at facilities such as PGC Visayas. Finally, more reliable mitochondrial assemblies  
245 for *S. lemuru* provide a stronger basis for downstream applications in the field of  
246 fisheries and genomics.

## 247 Chapter 2

## 248 Review of Related Literature

249 This chapter presents an overview of the literature relevant to the study. It  
250 discusses the biological and computational foundations underlying mitochondrial  
251 genome analysis and assembly, as well as existing tools, algorithms, and techniques  
252 related to chimera detection and genome quality assessment. The chapter aims to  
253 highlight the strengths, limitations, and research gaps in current approaches that  
254 motivate the development of the present study.

### 255 2.1 The Mitochondrial Genome

256 Mitochondrial genome (mtDNA) is a small, typically circular molecule found in  
257 most eukaryotes. It encodes essential genes involved in oxidative phosphorylation  
258 and energy metabolism. Because of its conserved structure, mtDNA has become  
259 a valuable genetic marker for studies in population genetics and phylogenetics  
260 (Anderson et al., 1981; Boore, 1999). In animal species, the mitochondrial genome



261 ranges from 15–20 kilobase and contains 13 protein-coding genes, 22 tRNAs, and  
262 two rRNAs arranged compactly without introns (Gray, 2012). In comparison to  
263 nuclear DNA, the ratio of the number of copies of mtDNA is higher and has  
264 simple organization which make it particularly suitable for genome sequencing  
265 and assembly studies (Dierckxsens et al., 2017).

### 266 **2.1.1 Mitochondrial Genome Assembly**

267 Mitochondrial genome assembly refers to the reconstruction of the complete mito-  
268 chondrial DNA (mtDNA) sequence from raw or fragmented sequencing reads. It is  
269 conducted to obtain high-quality, continuous representations of the mitochondrial  
270 genome that can be used for a wide range of analyses, including species identi-  
271 fication, phylogenetic reconstruction, evolutionary studies, and investigations of  
272 mitochondrial diseases. Because mtDNA evolves rapidly, its assembled sequence  
273 provides valuable insights into population structure, lineage divergence, and adap-  
274 tive evolution across taxa (Boore, 1999). Compared to nuclear genome assembly,  
275 assembling the mitochondrial genome is often considered more straightforward but  
276 still encounters technical challenges such as the formation of chimeric reads. Com-  
277 monly used tools for mitogenome assembly such as GetOrganelle and MITObim  
278 operate under the assumption of organelle genome circularity, and are vulnerable  
279 when chimeric reads disrupt this circular structure, resulting in assembly errors  
280 (Hahn et al., 2013; Jin et al., 2020).

## 2.2 PCR Amplification and Chimera Formation

PCR plays an important role in NGS library preparation, as it amplifies target DNA fragments for downstream analysis. However as previously mentioned, the amplification process can also introduce chimeric reads which compromises the quality of the input reads supplied to sequencing or assembly workflows. Chimeras typically arise when incomplete extension occurs during a PCR cycle. This causes the DNA polymerase to switch from one template to another and generate hybrid recombinant molecules (Judo et al., 1998). Artificial chimeras are produced through such amplification errors, whereas biological chimeras occur naturally through genomic rearrangements or transcriptional events.

In the context of amplicon-based sequencing, the presence of chimeras can inflate estimates of genetic or microbial diversity and may cause misassemblies during genome reconstruction. Qin et al. (2023) has reported that chimeric sequences may account for more than 10% of raw reads in amplicon datasets. This artifact tends to be most prominent among rare operational taxonomic units (OTUs) or singletons, which are sometimes misinterpreted as novel diversity, further causing the complication of microbial diversity analyses (Gonzalez, Zimmermann, & Saiz-Jimenez, 2004). As such, determining and minimizing PCR-induced chimera formation is vital for improving the quality of mitochondrial genome assemblies, and ensuring the reliability of amplicon sequencing data.

## 2.3 Existing Traditional Approaches for Chimera Detection

Several computational tools have been developed to identify chimeric sequences in NGS datasets. These tools generally fall into two categories: reference-based and de novo approaches. Reference-based chimera detection, also known as database-dependent detection, is one of the earliest and most widely used computational strategies for identifying chimeric sequences in amplicon-based community studies. These methods rely on the comparison of each query sequence against a curated, high-quality database of known, non-chimeric reference sequences (Edgar et al., 2011).

On the other hand, the de novo chimera detection, also referred to as reference-free detection, represents an alternative computational paradigm that identifies chimeric sequences without reliance on external reference databases. This method infer chimeras based on internal relationships among the sequences present within the dataset itself, making it particularly advantageous in studies of under explored or taxonomically diverse communities where comprehensive reference databases are unavailable or incomplete (Edgar, 2016; Edgar et al., 2011). The underlying assumption on this method is that during PCR, true biological sequences are generally more abundant as they are amplified early and dominate the read pool, whereas chimeric sequences appear later and are generally less abundant. The de novo approach leverage this abundance hierarchy, treating the most abundant sequences as supposed parents and testing whether less abundant sequences can be reconstructed as mosaics of these templates. Compositional and structural similarity are also evaluated to check whether different regions of a candidate

sequence correspond to distinct high-abundance sequences.

In practice, many modern bioinformatics pipelines combine both paradigms sequentially: an initial de novo step identifies dataset-specific chimeras, followed by a reference-based pass that removes remaining artifacts relative to established databases (Edgar, 2016). These two methods of detection form the foundation of tools such as UCHIME and later UCHIME2.

### 2.3.1 UCHIME

UCHIME is one of the most widely used tools for detecting chimeric sequences in amplicon-based studies and remains a standard quality-control step in microbial community analysis. Its core strategy is to test whether a query sequence ( $Q$ ) can be explained as a mosaic of two parent sequences, ( $A$  and  $B$ ), and to score this relationship using a structured alignment model (Edgar et al., 2011).

In reference mode, UCHIME divides the query into several segments and maps them against a curated database of non-chimeric sequences. Candidate parents are identified, and a three-way alignment is constructed. The algorithm assigns “Yes” votes when different segments of the query match different parents and “No” votes when the alignment contradicts a chimeric pattern. The final score reflects the balance of these votes. In de novo mode, UCHIME operationalizes the abundance-skew principle described earlier: high-abundance sequences are treated as candidate parents, and lower-abundance sequences are evaluated as potential mosaics. This makes the method especially useful when no reliable reference database exists.

347 Although UCHIME is highly sensitive, it faces key constraints. Chimeras  
348 formed from parents with very low divergence (below 0.8%) are difficult to detect  
349 because they are nearly indistinguishable from sequencing errors. Accuracy in ref-  
350 erence mode depends strongly on database completeness, while de novo detection  
351 assumes that true parents are both present and sufficiently more abundant, such  
352 conditions are not always met.

### 353 **2.3.2 UCHIME2**

354 UCHIME2 extends the original algorithm with refinements tailored for high-  
355 resolution sequencing data. One of its major contributions is a re-evaluation  
356 of benchmarking practices. Edgar (2016) demonstrated that earlier accuracy es-  
357 timates for chimera detection were overly optimistic because they relied on un-  
358 realistic scenarios where all true parent sequences were assumed to be present.  
359 Using the more rigorous CHSIMA benchmark, UCHIME2 showed the prevalence  
360 of “fake models” or real biological sequences that can be perfectly reconstructed  
361 as apparent chimeras of other sequences, which suggests that perfect chimera de-  
362 tection is theoretically unattainable. UCHIME2 also introduces several preset  
363 modes (e.g., denoised, balanced, sensitive, specific, high-confidence) designed to  
364 tune sensitivity and specificity depending on dataset characteristics. These modes  
365 allow users to adjust the algorithm to the expected noise level or analytical goals.

366 Despite these improvements, UCHIME2 must be applied with caution. The  
367 website manual explicitly advises against using UCHIME2 as a standalone  
368 chimera-filtering step in OTU clustering or denoising workflows because doing so  
369 can inflate both false positives and false negatives (Edgar, n.d.).

### 370 2.3.3 CATCh

371 As previously mentioned, UCHIME (Edgar et al., 2011) relied on alignment-based  
372 sequences in amplicon data. However, researchers soon observed that different al-  
373 gorithms often produced inconsistent predictions. A sequence might be identified  
374 as chimeric by one tool but classified as non-chimeric by another, resulting in  
375 unreliable filtering outcomes across studies.

376 To address these inconsistencies, Mysara, Saeys, Leys, Raes, and Monsieurs  
377 (2015) developed the Classifier for Amplicon Tool Chimeras (CATCh), which rep-  
378 resents the first ensemble machine learning system designed for chimera detection  
379 in 16S rRNA amplicon sequencing. Rather than depending on a single detec-  
380 tion strategy, CATCh integrates the outputs of several established tools, includ-  
381 ing UCHIME, ChimeraSlayer, DECIPHER, Pintail, and Perseus. The individual  
382 scores and binary decisions generated by these tools are used as input features for  
383 a supervised learning model. The algorithm employs a Support Vector Machine  
384 (SVM) with a Pearson VII Universal Kernel (PUK) to determine optimal weight-  
385 ings among the input features and to assign each sequence a probability of being  
386 chimeric.

387 Benchmarking in both reference-based and de novo modes demonstrated signif-  
388 icant performance improvements. CATCh achieved sensitivities of approximately  
389 85 percent in reference-based mode and 92 percent in de novo mode, with corre-  
390 sponding specificities of approximately 96 percent and 95 percent. These results  
391 indicate that CATCh detected 7 to 12 percent more chimeras than any individual  
392 algorithm while maintaining high precision.

### 393 2.3.4 ChimPipe

394 Among the available tools for chimera detection, ChimPipe is a pipeline developed  
395 to identify chimeric sequences such as biological chimeras. It uses both discordant  
396 paired-end reads and split-read alignments to improve the accuracy and sensitivity  
397 of detecting biological chimeras (Rodriguez-Martin et al., 2017). By combining  
398 these two sources of information, ChimPipe achieves better precision than meth-  
399 ods that depend on a single type of indicator.

400 The pipeline works with many eukaryotic species that have available genome  
401 and annotation data (Rodriguez-Martin et al., 2017). It can also predict multiple  
402 isoforms for each gene pair and identify breakpoint coordinates that are useful  
403 for reconstructing and verifying chimeric transcripts. Tests using both simulated  
404 and real datasets have shown that ChimPipe maintains high accuracy and reliable  
405 performance.

406 ChimPipe lets users adjust parameters to fit different sequencing protocols or  
407 organism characteristics. Experimental results have confirmed that many chimeric  
408 transcripts detected by the tool correspond to functional fusion proteins, demon-  
409 strating its utility for understanding chimera biology and its potential applications  
410 in disease research (Rodriguez-Martin et al., 2017).

## 411 **2.4 Machine Learning Approaches for Chimera** 412 **and Sequence Quality Detection**

413 Traditional chimera detection tools rely primarily on heuristic or alignment-based  
414 rules. Recent advances in machine learning (ML) have demonstrated that models  
415 trained on sequence-derived features can effectively capture compositional and  
416 structural patterns in biological sequences. Although most existing ML systems  
417 such as those used for antibiotic resistance prediction, taxonomic classification,  
418 or viral identification are not specifically designed for chimera detection, they  
419 highlight how data-driven models can outperform similarity-based heuristics by  
420 learning intrinsic sequence signatures. In principle, ML frameworks can integrate  
421 indicators such as k-mer frequencies, GC-content variation and split-alignment  
422 metrics to identify subtle anomalies that may indicate a chimeric origin (Arango  
423 et al., 2018; Liang, Bible, Liu, Zou, & Wei, 2020; Ren et al., 2020).

### 424 **2.4.1 Feature-Based Representations of Genomic Se-** 425 **quences**

426 Feature extraction converts DNA sequences into numerical representations suit-  
427 able for machine learning models. One approach is k-mer frequency analysis,  
428 which counts short nucleotide sequences within a read (Vervier, Mahé, Tournoud,  
429 Veyrieras, & Vert, 2015). High-frequency k-mers, including simple repeats such  
430 as “AAAAAA,” can highlight repetitive or unusual regions that may occur near  
431 chimeric junctions. Comparing k-mer patterns across adjacent parts of a read can  
432 help identify such regions, while GC content provides an additional descriptor of



433 local sequence composition (Ren et al., 2020).

434 Alignment-derived features further inform junction detection. Long-read tools  
435 such as Sniffles (Sedlazeck et al., 2018) use split alignments to locate breakpoints  
436 across extended sequences, whereas short-read aligners like Minimap2 (Li, 2018)  
437 report supplementary and secondary alignments that indicate local discontinu-  
438 ities. Split alignments, where parts of a read map to different regions, can reveal  
439 template-switching events. These features complement k-mer profiles and en-  
440 hance detection of potentially chimeric reads, even in datasets with incomplete  
441 references.

442 Microhomology, or short sequences shared between adjacent segments, is an-  
443 other biologically meaningful feature. Short microhomologies, typically 3–20 bp,  
444 are involved in template switching both in cellular repair pathways and during  
445 PCR, where they act as signatures of chimera formation (Peccoud et al., 2018;  
446 Sfeir & Symington, 2015). In PCR-induced chimeras, short identical sequences  
447 at junctions provide a clear signature of chimerism. Measuring the longest exact  
448 overlap at each breakpoint complements k-mer and alignment features and helps  
449 identify reads that are potentially chimeric.

## 450 **2.5 Synthesis of Chimera Detection Approaches**

451 To provide an integrated overview of the literature discussed in this chapter, Ta-  
452 ble 2.1 summarizes the major chimera detection studies, their methodological  
453 approaches, and their known limitations.

Table 2.1: Comparison of Chimera Detection Approaches and Tools

Method / Tool	Core Approach	Key Limitations
<b>Reference-based Detection</b>	Compares each query sequence against curated databases of verified, non-chimeric sequences; evaluates segment similarity to identify mosaic patterns.	Accuracy depends on database completeness; performs poorly for novel taxa or missing parents; limited sensitivity for low-divergence chimeras.
<b>De novo Detection</b>	Identifies chimeras using only internal dataset structure; leverages abundance hierarchy and compositional similarity to infer whether low-abundance sequences can be reconstructed from abundant parents.	Assumes true sequences are more abundant; fails when amplification bias distorts abundances; struggles when parental sequences are similarly abundant or highly similar.
<b>UCHIME</b>	Alignment-based model that partitions the query into segments, identifies parent candidates, and computes a chimera score via a three-way alignment; supports reference and de novo modes.	Reduced accuracy for very closely related parents (<0.8% divergence); sensitive to incomplete databases; de novo mode fails if parents are absent or not sufficiently more abundant.
<b>UCHIME2</b>	Updated UCHIME with improved benchmarking (CHSIMA) and multiple sensitivity/specificity presets; better handles incomplete references and dataset variability.	“Fake models” limit theoretical accuracy; genuine variants may mimic chimeras; not recommended as a standalone step in OTU or denoising pipelines due to increased false positives/negatives.
<b>CATCh</b>	First ensemble ML model for 16S chimera detection; integrates outputs of UCHIME, ChimeraSlayer, DECIPHER, Pintail, and Perseus using an SVM to boost overall prediction accuracy.	Performance constrained by underlying tools; ML model cannot capture features not present in component algorithms; may misclassify in highly novel or low-coverage datasets.
<b>ChimPipe</b>	Pipeline for detecting biological chimeras in RNA-seq using discordant paired-end reads and split-read alignments; identifies isoforms and breakpoint coordinates.	Requires high-quality genome and annotation; tailored to RNA-seq rather than amplicons; computationally intensive; limited to organisms with available reference genomes.

454 Across existing studies, no single approach reliably detects all forms of chimeric  
455 sequences, and the reviewed literature consistently shows that chimeras remain a  
456 persistent challenge in genomics and bioinformatics. Although the surveyed tools  
457 are not designed specifically for organelle genome assembly, they provide valu-  
458 able insights into which methodological strategies are effective and where current  
459 approaches fall short. These limitations collectively define a clear research gap:  
460 the need for a specialized, feature-driven detection framework tailored to PCR-  
461 induced mitochondrial chimeras. Addressing this gap aligns with the research  
462 objective outlined in Section 1.3, which is to develop and evaluate a machine  
463 learning-based pipeline (MitoChime) that improves the quality of downstream  
464 mitochondrial genome assembly. In support of this aim, the subsequent chapters  
465 describe the design, implementation, and evaluation of the proposed tool.

## 466 Chapter 3

# 467 Research Methodology

468 This chapter outlines the steps involved in completing the study, including data  
469 gathering, generating simulated mitochondrial Illumina reads, preprocessing and  
470 indexing the data, developing a feature extraction pipeline to obtain read-level fea-  
471 tures, applying machine learning algorithms for chimera detection, implementing  
472 feature selection methods, and validating and comparing model performance.

### 473 3.1 Research Activities

474 As illustrated in Figure 3.1, this study carried out a sequence of procedures to  
475 detect PCR-induced chimeric reads in mitochondrial genomes. The process began  
476 with collecting a mitochondrial reference sequence of *Sardinella lemuru* from the  
477 National Center for Biotechnology Information (NCBI) database, which was used  
478 as a reference for generating simulated clean and chimeric reads. These reads  
479 were subsequently indexed and mapped. The resulting collections then passed

480 through a feature extraction pipeline that computed k-mer profiles, supplementary  
 481 alignment (SA) features, and microhomology information to prepare the data  
 482 for model construction. The machine learning models were trained using the  
 483 processed input, evaluated using cross-validation and held-out testing, tuned for  
 484 improved performance, and then subjected to feature importance and feature  
 485 selection analyses before final validation.

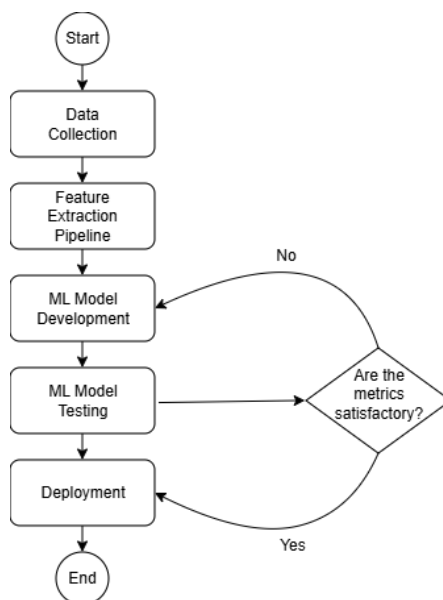


Figure 3.1: Process diagram of the study workflow.

### 486 3.1.1 Data Collection

487 The mitochondrial genome reference sequence of *S. lemuru* was obtained from the  
 488 NCBI database (accession number NC\_039553.1) in FASTA format and was used  
 489 to generate simulated reads.

490 This step was scheduled to begin in the first week of November 2025 and  
 491 expected to be completed by the end of that week, with a total duration of ap-

492 proximately one (1) week.

## 493 Data Preprocessing

494 All steps in the simulation and preprocessing pipeline were executed using a cus-  
495 tom script in Python (Version 3.11). The script runs each stage, including read  
496 simulation, reference indexing, mapping, and alignment processing, in a fixed se-  
497 quence.

498 `wgsim` (Version 1.13) was used to simulate 10,000 paired-end fragments, pro-  
499 ducing 20,000 reads (10,000 forward and 10,000 reverse) from the original refer-  
500 ence (`original_reference.fasta`) and designated as clean reads. The tool was  
501 selected because it provides fast generation of Illumina-like reads with controllable  
502 error rates, using the following command:

```
503 wgsim -1 150 -2 150 -r 0 -R 0 -X 0 -e 0.05 -N 10000 \  
504     original_reference.fasta ref1.fastq ref2.fastq
```

505 Chimeric sequences were then generated from the same reference FASTA  
506 file using a separate Python script. Two non-adjacent segments were ran-  
507 domly selected such that their midpoint distances fell within specified minimum  
508 and maximum thresholds. The script attempted to retain microhomology to  
509 mimic PCR-induced template switching. The resulting chimeras were written  
510 to `chimera_reference.fasta` and processed with `wgsim` to simulate 10,000  
511 paired-end fragments, generating 20,000 chimeric reads (10,000 forward reads in  
512 `chimeric1.fastq` and 10,000 reverse reads in `chimeric2.fastq`) using the same  
513 command format as above.

514       Next, a `minimap2` index of the reference genome was created using:

```
515 minimap2 -d ref.mmi original_reference.fasta
```

516       Minimap2 (Version 2.28) was used to map simulated clean and chimeric reads  
517 to the original reference. An index (`ref.mmi`) was first generated to enable efficient  
518 alignment, and mapping produced the alignment features used as input for the  
519 machine learning model. The reads were mapped using the following commands:

```
520 minimap2 -ax sr -t 8 ref.mmi ref1.fastq ref2.fastq > clean.sam
```

```
521 minimap2 -ax sr -t 8 ref.mmi \  
522   chimeric1.fastq chimeric2.fastq > chimeric.sam
```

523       The resulting clean and chimeric SAM files contain the alignment positions of  
524 each read relative to the original reference genome. These files were then converted  
525 to BAM format, sorted, and indexed using `samtools` (Version 1.20):

```
526 samtools view -bS clean.sam -o clean.bam
```

```
527 samtools view -bS chimeric.sam -o chimeric.bam
```

528

```
529 samtools sort clean.bam -o clean.sorted.bam
```

```
530 samtools index clean.sorted.bam
```

531

```
532 samtools sort chimeric.bam -o chimeric.sorted.bam
```

```
533 samtools index chimeric.sorted.bam
```

534 The total number of simulated reads was expected to be 40,000. The final col-  
535 lection of reads contained 19,984 clean reads and 20,000 chimeric reads (39,984 en-  
536 tries in total), providing a roughly balanced distribution between the two classes.  
537 After alignment with `minimap2`, only 19,984 clean reads remained because un-  
538 mapped reads were not included in the BAM file. Some sequences failed to align  
539 due to the error rate defined during `wgsim` simulation, which produced mismatches  
540 that caused certain reads to fall below the aligner’s matching threshold.

541 This whole process was scheduled to start in the second week of November 2025  
542 and was expected to be completed by the last week of November 2025, with a total  
543 duration of approximately three (3) weeks.

### 544 **3.1.2 Feature Extraction Pipeline**

545 This stage directly followed the alignment phase, utilizing the resulting BAM files  
546 (specifically `chimeric.sorted.bam` and `clean.sorted.bam`). A custom Python  
547 script was created to efficiently process each primary-mapped read to extract  
548 the necessary set of features, which were then compiled into a structured feature  
549 matrix in TSV format. The pipeline’s core functionality relied on the `Pysam`  
550 (Version 0.22) library for parsing BAM structures and `NumPy` (Version 1.26) for  
551 array operations and computations. To ensure correctness and adherence to best  
552 practices, bioinformatics experts at PGC Visayas were consulted to validate the  
553 pipeline design, feature extraction logic, and overall data integrity.

554 This stage of the study was scheduled to begin in the last week of Novem-  
555 ber 2025 and conclude by the first week of December 2025, with an estimated



556 total duration of approximately two (2) weeks.

557     The pipeline focused on three feature families that collectively capture bi-  
558 ological signatures associated with PCR-induced chimeras: (1) supplementary  
559 alignment (SA) and alignment-structure metrics, (2) k-mer composition differ-  
560 ence, and (3) microhomology around putative junctions. Additional alignment  
561 quality indicators such as mapping quality were also included.

## 562 **Supplementary Alignment and Alignment-Structure Features**

563 Split-alignment information was derived from the SA tag embedded in each pri-  
564 mary read of the BAM file. This tag is typically associated with reads that map to  
565 multiple genomic locations, suggesting a chimeric structure. To extract this infor-  
566 mation, the script first checked whether the read carried an **SA:Z** tag. If present,  
567 the tag string was parsed using the function `parse_sa_tag`, yielding metadata for  
568 each alignment containing the reference name, mapped position, strand, mapping  
569 quality, and number of mismatches.

570     After parsing, the function `sa_feature_stats` was applied to establish the fun-  
571 damental split indicators, `has_sa` and `sa_count`. Along with these initial counts,  
572 the function aggregated metrics related to the structure and reliability of the  
573 split alignments, including the number of alignment segments, strand consistency,  
574 minimum, maximum, and mean distance between split segments, and summary  
575 statistics of mapping quality and mismatch counts across segments.

## 576 **K-mer Composition Difference**

577 Comparing k-mer frequency profiles between the left and right halves of a read  
578 allows for the detection of abrupt compositional shifts, independent of alignment  
579 information.

580 The script implemented this by inferring a likely junction breakpoint using the  
581 function `infer_breakpoints`, prioritizing the boundaries defined by soft-clipping  
582 operations. If no clipping was present, the midpoint of the alignment or the read  
583 length was used as a fallback. The read sequence was then divided into left and  
584 right segments at this inferred breakpoint, and k-mer frequency profiles ( $k =$   
585 6) were generated for both halves, ignoring any k-mers containing ambiguous N  
586 bases. The resulting k-mer frequency vectors were normalised and compared using  
587 the functions `cosine_difference` and `js_divergence` to quantify compositional  
588 discontinuity across the inferred breakpoint.

## 589 **Microhomology**

590 The process of extracting the microhomology feature also started by using  
591 `infer_breakpoints` to identify a candidate junction. Once a breakpoint was  
592 established, the script scanned a  $\pm 40$  base-pair window surrounding the break-  
593 point and applied the function `longest_suffix_prefix_overlap` to identify the  
594 longest exact suffix-prefix overlap between the left and right read segments. This  
595 overlap, representing consecutive bases shared at the junction, was recorded as  
596 `microhomology_length` in the dataset. The 40 base-pair window was chosen  
597 to ensure that short shared sequences at or near the breakpoint were captured

without including distant sequences that are unlikely to be mechanistically related.

Additionally, the GC content of the overlapping sequence was calculated using the function `gc_content`, which counts guanine (G) and cytosine (C) bases within the detected microhomology and divides by the total length, yielding a proportion between 0 and 1 that was stored under the `microhomology_gc` attribute. Microhomology was quantified using a 3–20 bp window, consistent with values reported in prior research on PCR-induced chimeras. A k-mer length of 6 was used to capture patterns within the 40 bp window surrounding each breakpoint, providing sufficient resolution to detect informative sequence shifts.

### 3.1.3 Machine Learning Model Development

After feature extraction, the per-read feature matrices for clean and chimeric reads were merged into a single dataset. Each row corresponded to one paired-end read, and columns encoded alignment-structure features (e.g., supplementary alignment count and spacing between segments), CIGAR-derived soft-clipping statistics (e.g., left and right soft-clipped length, total clipped bases), k-mer composition discontinuity between read segments, microhomology descriptors near candidate junctions, and alignment quality (e.g., mapping quality). The resulting feature set comprised 23 numeric features and was restricted to quantities that can be computed from standard BAM/FASTQ files in typical mitochondrial sequencing workflows.

The labelled dataset was randomly partitioned into training (80%) and test

620 (20%) subsets using stratified sampling to preserve the 1:1 ratio of clean to  
 621 chimeric reads. Model development and evaluation were implemented in Python  
 622 (Version 3.11) using the `scikit-learn`, `xgboost`, `lightgbm`, and `catboost` li-  
 623 braries. A broad panel of classification algorithms was then benchmarked on the  
 624 training data to obtain a fair comparison of different model families under identical  
 625 feature conditions. The panel included a trivial dummy classifier,  $L_2$ -regularized  
 626 logistic regression, a calibrated linear support vector machine (SVM),  $k$ -nearest  
 627 neighbours, Gaussian Naïve Bayes, decision-tree ensembles (Random Forest, Ex-  
 628 tremely Randomized Trees, and Bagging with decision trees), gradient boosting  
 629 methods (Gradient Boosting, XGBoost, LightGBM, and CatBoost), and a shallow  
 630 multilayer perceptron (MLP).

631 For each model, five-fold stratified cross-validation was performed on the train-  
 632 ing set. In every fold, four-fifths of the data were used for fitting and the remaining  
 633 one-fifth for validation. Mean cross-validation accuracy, precision, recall, F1-score  
 634 for the chimeric class, and area under the receiver operating characteristic curve  
 635 (ROC-AUC) were computed to summarize performance and rank candidate meth-  
 636 ods. This baseline screen allowed comparison of linear, probabilistic, neural, and  
 637 ensemble-based approaches and identified tree-based ensemble and boosting mod-  
 638 els as consistently strong performers relative to simpler baselines.

### 639 **3.1.4 Model Benchmarking, Hyperparameter Optimiza-** 640 **tion, and Evaluation**

641 Model selection and refinement proceeded in two stages. First, the cross-validation  
 642 results from the broad panel were used to identify a subset of competitive mod-

643 els for more detailed optimization. Specifically, ten model families were carried  
644 forward:  $L_2$ -regularized logistic regression, calibrated linear SVM, Random For-  
645 est, ExtraTrees, Gradient Boosting, XGBoost, LightGBM, CatBoost, Bagging  
646 with decision trees, and a shallow MLP. This subset spans both linear and non-  
647 linear decision boundaries, but emphasizes ensemble and boosting methods, which  
648 showed superior F1 and ROC-AUC in the initial benchmark.

649 Second, hyperparameter optimization was conducted for each of the ten se-  
650 lected models using randomized search with five-fold stratified cross-validation  
651 (`RandomizedSearchCV`). For tree-based ensembles, the search space included the  
652 number of trees, maximum depth, minimum samples per split and per leaf, and  
653 the fraction of features considered at each split. For boosting methods, key hyper-  
654 parameters such as the number of boosting iterations, learning rate, tree depth,  
655 subsampling rate, and column subsampling rate were tuned. For the MLP, the  
656 number and size of hidden layers, learning rate, and  $L_2$ -regularization strength  
657 were varied. In all cases, the primary optimisation criterion was the F1-score of  
658 the chimeric class, averaged across folds.

659 For each model family, the hyperparameter configuration with the highest  
660 mean cross-validation F1-score was selected as the best-tuned estimator. These  
661 tuned models were then refitted on the full training set and evaluated once on the  
662 held-out test set to obtain unbiased estimates of performance. Test-set metrics in-  
663 cluded accuracy, precision, recall, F1-score for the chimeric class, and ROC-AUC.  
664 Confusion matrices and ROC curves were generated for the top-performing mod-  
665 els to characterise common error modes, such as false negatives (missed chimeric  
666 reads) and false positives (clean reads incorrectly labelled as chimeric). The final  
667 model or small set of models for downstream interpretation was chosen based on

668 a combination of test-set F1-score and ROC-AUC.

### 669 **3.1.5 Feature Importance, Feature Selection, and Inter-** 670 **pretation**

671 To relate model decisions to biologically meaningful signals, feature-importance  
672 analyses were performed on the best-performing tree-based models. Two comple-  
673 mentary approaches were used. First, built-in importance measures from ensemble  
674 methods (e.g., split-based importances in Random Forest and Gradient Boosting)  
675 were examined to obtain an initial ranking of features based on their contribution  
676 to reducing impurity. Second, model-agnostic permutation importance was com-  
677 puted on the test set by repeatedly permuting each feature column while keeping  
678 all others fixed and measuring the resulting decrease in F1-score. Features whose  
679 permutation led to a larger performance drop were interpreted as more influential  
680 for chimera detection.

681 For interpretability, individual features were grouped into conceptual families:  
682 (i) supplementary alignment and alignment-structure features (e.g., SA count,  
683 spacing between alignment segments, strand consistency), (ii) soft-clipping fea-  
684 tures (e.g., left and right soft-clipped length, total clipped bases, inferred break-  
685 point position), (iii) k-mer composition discontinuity features (e.g., cosine dis-  
686 tance and Jensen-Shannon divergence between k-mer profiles of read segments),  
687 (iv) microhomology descriptors (e.g., microhomology length and local GC content  
688 around putative breakpoints), and (v) other alignment quality features (e.g., map-  
689 ping quality). This analysis provided a basis for interpreting the trained models  
690 in terms of known mechanisms of PCR-induced template switching and for iden-

691 tifying which alignment-based and sequence-derived cues are most informative for  
692 distinguishing chimeric from clean mitochondrial reads.

693 Building on these importance results, an explicit feature selection step was  
694 implemented using CatBoost as the reference model, since it was among the top-  
695 performing classifiers. Permutation importance scores were re-estimated for Cat-  
696 Boost on the held-out test set using the F1-score of the chimeric class as the  
697 scoring function. Negative importance scores, which indicate that permuting a  
698 feature did not reliably harm performance, were set to zero and interpreted as  
699 noise. The remaining non-negative importances were sorted in descending order  
700 and converted into a cumulative importance curve by expressing each feature's  
701 importance as a fraction of the total positive importance.

702 A compact feature subset was then defined by selecting the smallest number of  
703 features whose cumulative importance reached at least 95% of the total positive  
704 importance. This procedure yielded a reduced set of four strongly predictive  
705 variables dominated by soft-clipping and k-mer divergence metrics (for example,  
706 total clipped bases and k-mer divergence between read halves).

707 To quantify the impact of this reduction, CatBoost was retrained using only  
708 the selected feature subset, with the same tuned hyperparameters as the full 23-  
709 feature model, and evaluated on the held-out test set. Performance of the reduced  
710 model was then compared to that of the full model in terms of F1-score and ROC-  
711 AUC to assess whether dimensionality could be reduced without appreciable loss  
712 in predictive accuracy.

713 In addition, an ablation experiment was performed to specifically evaluate  
714 the contribution of explicit microhomology features. The microhomology vari-

ables (`microhomology_length` and `microhomology_gc`) were removed from the full feature set to obtain a 21-feature configuration. CatBoost was refitted on this microhomology-ablated feature set, using the same tuned hyperparameters, and evaluated on the held-out test set. Comparing the full, reduced-subset, and microhomology-ablated variants allowed the study to quantify both the degree of redundancy among features and the practical contribution of microhomology to classification accuracy.

Taken together, the feature importance and feature selection analyses provided a more parsimonious model variant and a clearer interpretation of which alignment-based and sequence-derived signals are most informative for detecting PCR-induced chimeras.

### 3.1.6 Validation and Testing

Validation involved both internal and external evaluations. Internal validation was achieved through five-fold stratified cross-validation on the training data to verify model generalization and reduce variance due to random sampling. External testing was performed on the 20% hold-out dataset from the simulated reads, providing an unbiased assessment of model generalization. Feature extraction and preprocessing were applied consistently across all splits.

Comparative evaluation was performed across all candidate algorithms and CatBoost feature-set variants to determine which models demonstrated the highest predictive performance and computational efficiency under identical data conditions. Their metrics were compared to identify which algorithms and feature



737 configurations were most suitable for further refinement and potential integration  
738 into downstream mitochondrial assembly workflows.

### 739 **3.1.7 Documentation**

740 Comprehensive documentation was maintained throughout the study to ensure  
741 transparency and reproducibility. All stages of the research, including data gath-  
742 ering, preprocessing, feature extraction, model training, feature selection, and  
743 validation, were systematically recorded in a **README** file in the GitHub reposi-  
744 tory. For each analytical step, the corresponding parameters, software versions,  
745 and command line scripts were documented to enable exact replication of results.

746 The repository structure followed standard research data management prac-  
747 tices, with clear directories for datasets and scripts. Computational environments  
748 were standardised using Conda, with an environment file (**environment.yml**)  
749 specifying dependencies and package versions to maintain consistency across sys-  
750 tems.

751 For manuscript preparation and supplementary materials, Overleaf (L<sup>A</sup>T<sub>E</sub>X)  
752 was used to produce publication-quality formatting and consistent referencing.

## 753 **3.2 Calendar of Activities**

754 Table 3.1 presents the project timeline in the form of a Gantt chart, where each  
755 bullet point corresponds to approximately one week of planned activity.

Table 3.1: Timetable of activities.

Activities (2025)	Nov	Dec	Jan	Feb	Mar	Apr	May
Data Collection and Simulation	• • • •						
Feature Extraction Pipeline	•	•					
Machine Learning Development		•	• •	• • • •	• • • •	• •	
Testing and Validation						• •	• • • •
Documentation	• • • •	• • • •	• • • •	• • • •	• • • •	• • • •	• • • •

## 756 Chapter 4

# 757 Results and Discussion

758 This chapter presents the performance of the proposed feature set and machine  
759 learning models for detecting PCR-induced chimeric reads in simulated mitochon-  
760 drial Illumina data. The behaviour of the main features is first described, followed  
761 by a comparison of baseline classifiers, an assessment of the effect of hyperparam-  
762 eter tuning, and an analysis of feature importance in terms of individual variables  
763 and feature families.

764 The final dataset contained 31,986 reads for training and 7,997 reads for test-  
765 ing, with classes balanced (approximately 4,000 clean and 4,000 chimeric reads in  
766 the test split).

## 767 4.1 Descriptive Analysis of Features

### 768 4.1.1 Summary Statistics Per Class

769 Summary statistics were computed separately for clean reads (class 0) and  
770 chimeric reads (class 1) to characterize the distributional behavior of the features.  
771 For each feature, the mean, standard deviation, median, first and third quartiles  
772 (Q1, Q3), interquartile range (IQR), minimum, maximum, and sample size ( $n$ )  
773 were calculated.

774 Only a subset of the features is summarized in the main text to highlight key  
775 trends, and not all summary statistics columns are shown for brevity. The com-  
776 plete set of per-class summary statistics for all features is provided in Appendix A  
777 (Table A.1).

### 778 Alignment and Supplementary Alignment Features

779 Features related to supplementary alignments show strong separation between  
780 classes. Chimeric reads exhibit supplementary alignments, as reflected by higher  
781 values of `has_sa`, `sa_count`, and `num_segments`, whereas clean reads consistently  
782 show a single alignment segment with no supplementary mappings. This behavior  
783 is consistent with the expected structure of chimeric reads and indicates that  
784 alignment-based features are highly informative.

## 785 **Clipping-Based Features**

786 Clipping-related features, including `softclip_left`, `softclip_right`, and  
787 `total_clipped_bases`, display higher means and broader distributions in chimeric  
788 reads. Clean reads are dominated by zero or near-zero clipping, while chimeric  
789 reads exhibit increased clipping and greater variability, which reflects the presence  
790 of split alignments.

## 791 **K-mer Distribution Features**

792 K-mer-based features, such as `kmer_js_divergence` and `kmer_cosine_diff`, show  
793 only modest differences between clean and chimeric reads. Chimeric reads show  
794 slightly higher average divergence, but substantial overlap with clean reads means  
795 this feature alone cannot reliably distinguish the classes.

## 796 **Microhomology Features**

797 Microhomology-related features (`microhomology_length` and `microhomology_gc`)  
798 exhibit nearly identical summary statistics across both classes. The majority of  
799 reads in both classes contain short or zero-length microhomologies, resulting in  
800 minimal separation. This means that microhomology serves as a weak standalone  
801 indicator and is more appropriately treated as supporting evidence.

802 Overall, the summary statistics indicate that alignment-based and clipping-  
803 based features provide the strongest class separation, k-mer features contribute  
804 limited but complementary signal, and microhomology features exhibit minimal

discriminative power on their own. These observations motivate the combined multi-feature approach used in subsequent modeling and evaluation.

Table 4.1: Summary statistics of selected key features by class.

Feature	Class	Mean	Std	Median	IQR
has_sa	chimeric	0.406	0.491	0.0	1.0
has_sa	clean	0.000	0.000	0.0	0.0
num_segments	chimeric	1.406	0.491	1.0	1.0
num_segments	clean	1.000	0.000	1.0	0.0
softclip_left	chimeric	12.55	21.90	0.0	19.0
softclip_left	clean	0.23	1.54	0.0	0.0
softclip_right	chimeric	12.90	22.12	0.0	19.0
softclip_right	clean	0.21	1.51	0.0	0.0
total_clipped_bases	chimeric	25.44	25.48	19.0	48.0
total_clipped_bases	clean	0.44	2.16	0.0	0.0
kmer_js_divergence	chimeric	0.974	0.025	0.986	0.043
kmer_js_divergence	clean	0.976	0.025	0.986	0.040
kmer_cosine_diff	chimeric	0.974	0.026	0.986	0.042
kmer_cosine_diff	clean	0.976	0.025	0.986	0.041
microhomology_length	chimeric	0.458	0.755	0.0	1.0
microhomology_length	clean	0.462	0.758	0.0	1.0
microhomology_gc	chimeric	0.172	0.361	0.0	0.0
microhomology_gc	clean	0.172	0.361	0.0	0.0

### 4.1.2 Boxplots By Class

Boxplots were generated for each feature, with the x-axis representing the class clean reads and chimeric reads and the y-axis representing the feature value. Figure 4.1 presents a panel of selected key features, while boxplots for all numeric features are provided in Appendix B.

For clipping-related features, chimeric reads exhibit higher medians and longer upper whiskers than clean reads, indicating increased variability and the presence

814 of split alignments.

815       Supplementary alignment features show that clean reads are largely zero,  
816       whereas chimeric reads display a wider distribution, reflecting frequent supple-  
817       mentary alignments.

818       K-mer metrics show a slight upward shift for chimeric reads, but substantial  
819       overlap with clean reads indicates modest discriminative power.

820       Microhomology features have nearly overlapping distributions for both classes,  
821       consistent with their low standalone predictive importance.

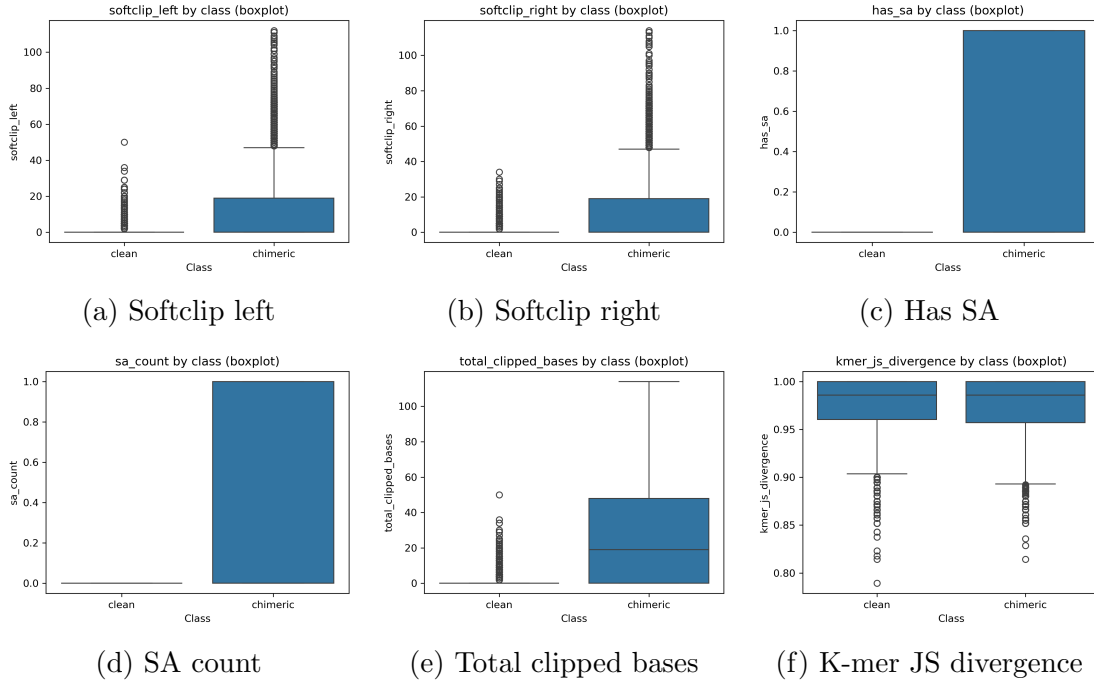


Figure 4.1: Boxplots of key features by class

## 822 4.2 Baseline Classification Performance

823 Table 4.2 summarises the performance of eleven classifiers trained on the engi-  
824 neered feature set using five-fold cross-validation and evaluated on the held-out  
825 test set. All models were optimised using default hyperparameters, without ded-  
826 icated tuning.

827 The dummy baseline, which always predicts the same class regardless of the  
828 input features, achieved an accuracy of 0.50 and test F1-score of 0.67. This re-  
829 flects the balanced class distribution and provides a lower bound for meaningful  
830 performance.

831 Across other models, test F1-scores clustered in a narrow band between ap-  
832 proximately 0.74 and 0.77 and ROC-AUC values between 0.82 and 0.84. Gradi-  
833 ent boosting, CatBoost, LightGBM, XGBoost, bagging trees, random forest, and  
834 multilayer perceptron (MLP) all produced very similar scores, with CatBoost and  
835 gradient boosting slightly ahead (test F1  $\approx$  0.77, ROC-AUC  $\approx$  0.84). Linear  
836 models (logistic regression and calibrated linear SVM) performed only marginally  
837 worse (test F1  $\approx$  0.74), while Gaussian Naive Bayes lagged behind with substan-  
838 tially lower F1 ( $\approx$  0.65) despite very high precision for the chimeric class.



Table 4.2: Performance of baseline classifiers on the held-out test set.

model	test_accuracy	test_precision	test_recall	test_f1	test_roc_auc
dummy_baseline	0.500000	0.500000	1.000000	0.667000	0.500000
logreg_l2	0.789000	0.945000	0.614000	0.744000	0.821000
linear_svm_calibrated	0.789000	0.945000	0.614000	0.744000	0.820000
random_forest	0.788000	0.894000	0.654000	0.755000	0.834000
extra_trees	0.788000	0.901000	0.647000	0.753000	0.824000
gradient_boosting	0.802000	0.936000	0.648000	0.766000	0.840000
xgboost	0.800000	0.929000	0.650000	0.765000	0.839000
lightgbm	0.799000	0.926000	0.650000	0.764000	0.838000
catboost	0.803000	0.936000	0.650000	0.767000	0.839000
knn	0.782000	0.892000	0.642000	0.747000	0.815000
gaussian_nb	0.741000	0.996000	0.483000	0.651000	0.819000
bagging_trees	0.792000	0.900000	0.657000	0.760000	0.837000
mlp	0.789000	0.931000	0.625000	0.748000	0.819000

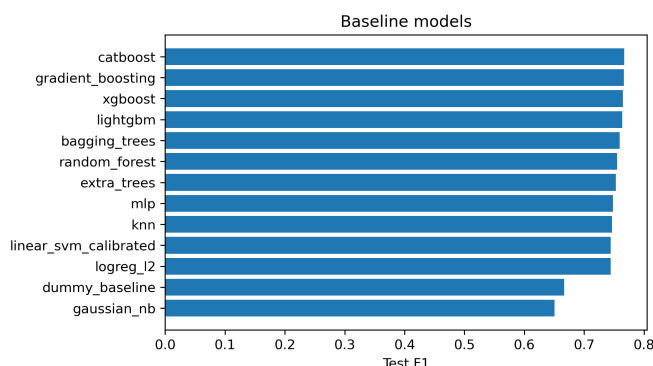


Figure 4.2: Test F1 of all baseline classifiers, showing that no single model clearly dominates and several achieve comparable performance.

### 839 4.3 Effect of Hyperparameter Tuning

840 To assess whether performance could be improved further, ten model families un-  
841 derwent randomised hyperparameter search. The tuned metrics are summarised  
842 in Table 4.3. Overall, tuning yielded modest but consistent gains for tree-based en-  
843 sembles and boosting methods, while leaving linear models essentially unchanged

844 or slightly worse.

845 CatBoost, gradient boosting, LightGBM, XGBoost, random forest, bagging  
846 trees, and MLP all experienced small increases in test F1 (typically  $\Delta F1 \approx 0.002$ –  
847 0.009) and ROC–AUC (up to  $\Delta AUC \approx 0.008$ ). After tuning, CatBoost remained  
848 the best performer with test accuracy 0.80, precision 0.92, recall 0.66, F1-score  
849 0.77, and ROC–AUC 0.84. Gradient boosting achieved almost identical perfor-  
850 mance (F1 0.77, AUC 0.84). Random forest and bagging trees also improved to  
851 F1 scores around 0.76 with  $AUC \approx 0.84$ .

Table 4.3: Performance of tuned classifiers on the held-out test set.

model	test_accuracy	test_precision	test_recall	test_f1	test_roc_auc
logreg_l2_tuned	0.788000	0.946000	0.612000	0.743000	0.818000
linear_svm_calibrated_tuned	0.788000	0.944000	0.612000	0.743000	0.818000
random_forest_tuned	0.797000	0.915000	0.655000	0.763000	0.842000
extra_trees_tuned	0.794000	0.910000	0.652000	0.760000	0.837000
gradient_boosting_tuned	0.802000	0.928000	0.654000	0.767000	0.843000
xgboost_tuned	0.799000	0.922000	0.653000	0.765000	0.839000
lightgbm_tuned	0.801000	0.930000	0.651000	0.766000	0.842000
catboost_tuned	0.802000	0.924000	0.658000	0.769000	0.844000
bagging_trees_tuned	0.798000	0.922000	0.650000	0.763000	0.842000
mlp_tuned	0.790000	0.934000	0.625000	0.749000	0.821000

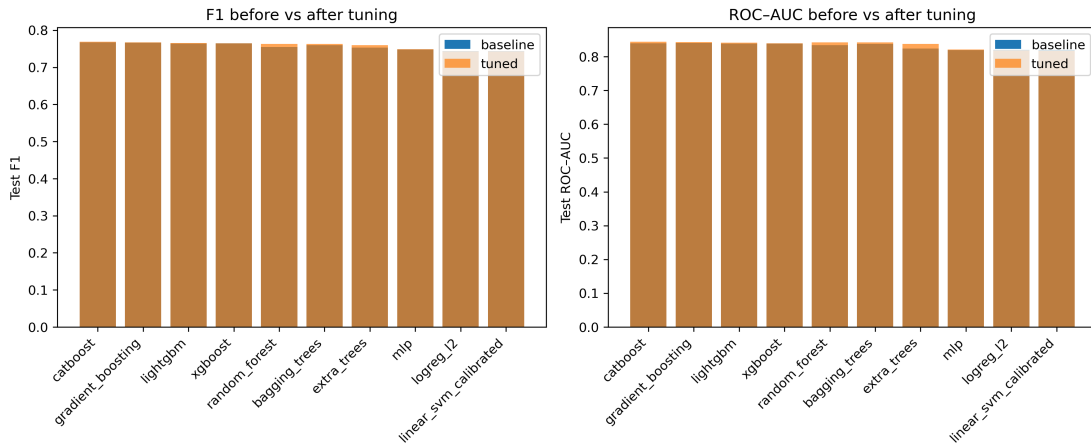


Figure 4.3: Comparison of test F1 (left) and ROC–AUC (right) for baseline and tuned models.

852 Because improvements are small and within cross-validation variability, tun-  
853 ing was interpreted as stabilising and slightly refining the models rather than  
854 completely altering their behaviour or their relative ranking.

## 855 4.4 Detailed Evaluation of Representative Mod- 856 els

857 For interpretability and diversity, four tuned models were selected for deeper  
858 analysis: CatBoost (best-performing boosted tree), scikit-learn gradient boost-  
859 ing (canonical gradient-boosting implementation), random forest (non-boosted  
860 ensemble baseline), and  $L_2$ -regularised logistic regression (linear baseline). All  
861 models were trained on the engineered feature set and evaluated on the same  
862 held-out test data.

### 863 4.4.1 Confusion Matrices and Error Patterns

864 Classification reports and confusion matrices for the four models reveal consistent  
865 patterns. CatBoost and gradient boosting both reached overall accuracy of ap-  
866 proximately 0.80 with similar macro-averaged F1 scores ( $\sim 0.80$ ). For CatBoost,  
867 precision and recall for clean reads were 0.73 and 0.95, respectively, while for  
868 chimeric reads they were 0.92 and 0.66 ( $F1 = 0.77$ ). Gradient boosting showed  
869 nearly identical trade-offs.

870 Random forest attained slightly lower accuracy (0.80) and chimeric F1 (0.76),  
871 whereas logistic regression achieved the lowest accuracy among the four (0.79)

872 and chimeric F1 (0.74), although it provided the highest chimeric precision (0.95)  
873 at the cost of lower recall (0.61).

874 Across all models, errors were asymmetric. False negatives (chimeric reads pre-  
875 dicted as clean) were more frequent than false positives. For example, CatBoost  
876 misclassified 1,369 chimeric reads as clean but only 215 clean reads as chimeric.  
877 This pattern indicates that the models are conservative and prioritise avoiding  
878 false chimera calls at the expense of missing some true chimeras. Consultation  
879 with PGC Visayas indicated that this conservative behavior is generally accept-  
880 able, though further evaluation and testing will be required to assess its impact  
881 on downstream analyses.

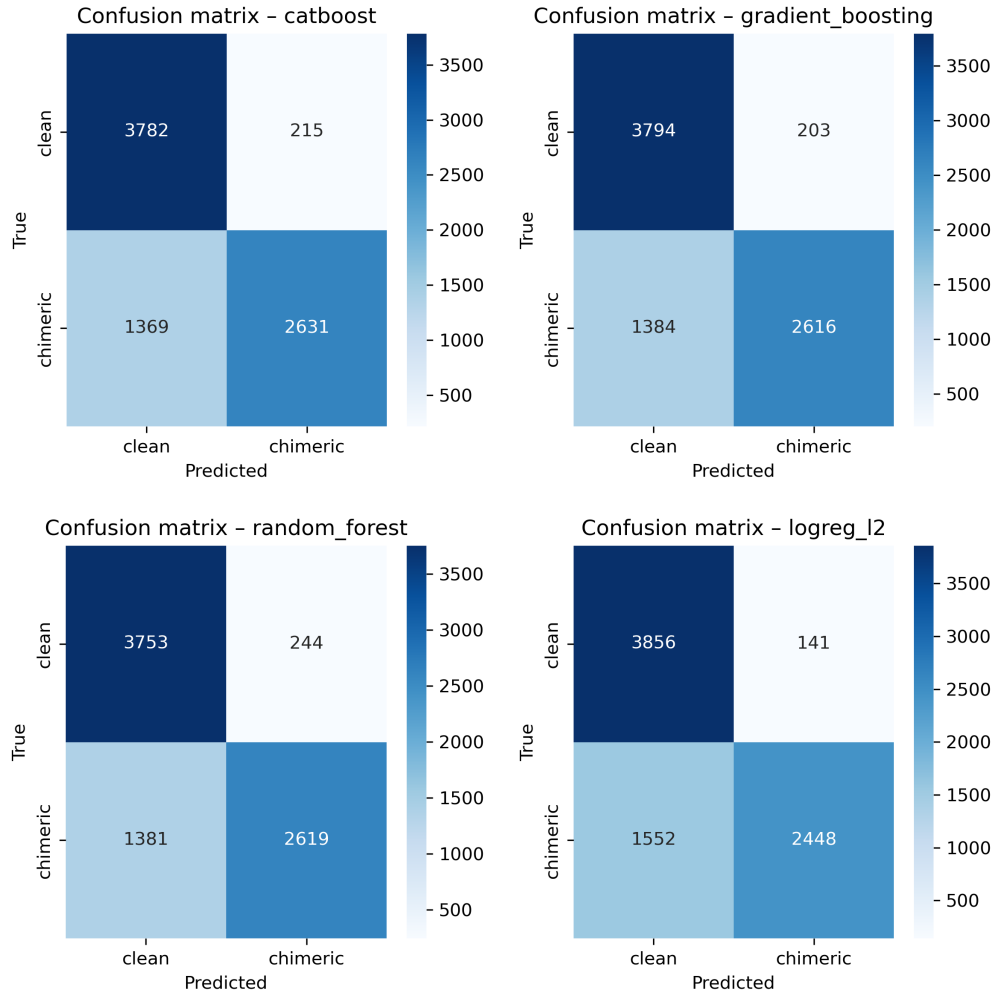


Figure 4.4: Confusion matrices for the four representative models on the held-out test set.

## 882 4.4.2 ROC and Precision–Recall Curves

883 Receiver operating characteristic (ROC) and precision–recall (PR) curves as  
 884 shown in Figure 4.5 further support the similarity among the top models. The  
 885 three tree-based ensembles (CatBoost, gradient boosting, random forest) achieved  
 886 ROC–AUC values of approximately 0.84 and average precision (AP) around 0.88.

887 Logistic regression performed slightly worse ( $AUC \approx 0.82$ ,  $AP \approx 0.87$ ) but still  
 888 substantially better than the dummy baseline.

889 The PR curves show that precision remains above 0.9 across a broad range  
 890 of recall values (up to roughly 0.5–0.6), after which precision gradually declines.  
 891 This behaviour indicates that the models can assign very high confidence to a  
 892 subset of chimeric reads, while more ambiguous reads can only be recovered by  
 893 accepting lower precision.

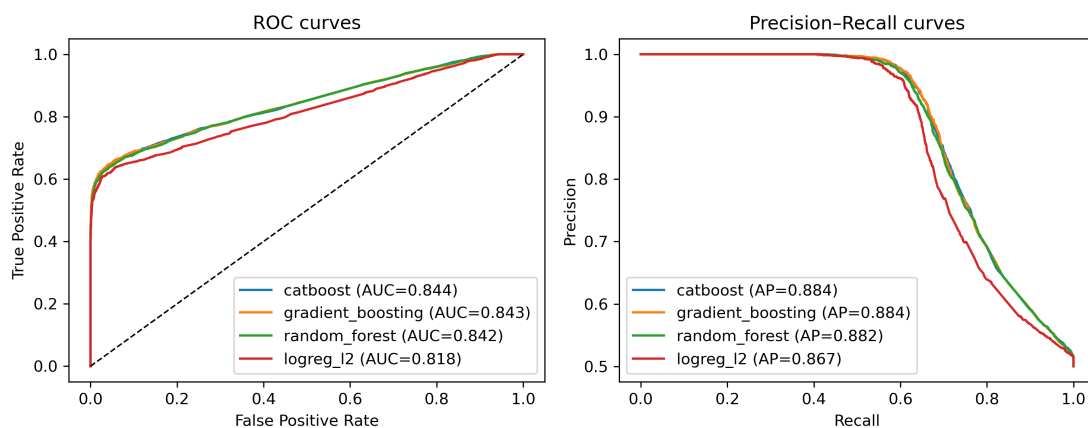


Figure 4.5: ROC (left) and precision–recall (right) curves for the four representative models on the held-out test set.

## 894 4.5 Feature Importance

### 895 4.5.1 Permutation Importance of Individual Features

896 To understand how each classifier made predictions, feature importance was quan-  
 897 tified using permutation importance. This analysis was applied to four represen-  
 898 tative models: CatBoost, Gradient Boosting, Random Forest, and  $L_2$ -regularized

899 Logistic Regression.

900 As shown in Figure 4.6, the total number of clipped bases consistently pro-  
901 vides a strong predictive signal, particularly in Random Forest, Gradient Boosting,  
902 and L<sub>2</sub>-regularized Logistic Regression. CatBoost differs by assigning the highest  
903 importance to k-mer divergence metrics such as `kmer_js_divergence`, which cap-  
904 ture subtle sequence changes resulting from structural variants or PCR-induced  
905 chimeras. Soft-clipping features (`softclip_left` and `softclip_right`) provide  
906 more information around breakpoints, complementing these primary signals in all  
907 models except Gradient Boosting. L<sub>2</sub>-regularized Logistic Regression relies more  
908 on alignment-based split-read metrics.

909 Overall, these results indicate that accurate detection of chimeric reads relies  
910 on both alignment-based signals and k-mer compositional information. Explicit  
911 microhomology features contribute minimally in this analysis, and combining both  
912 alignment-based and sequence-level features enhances model sensitivity and speci-  
913 ficity.

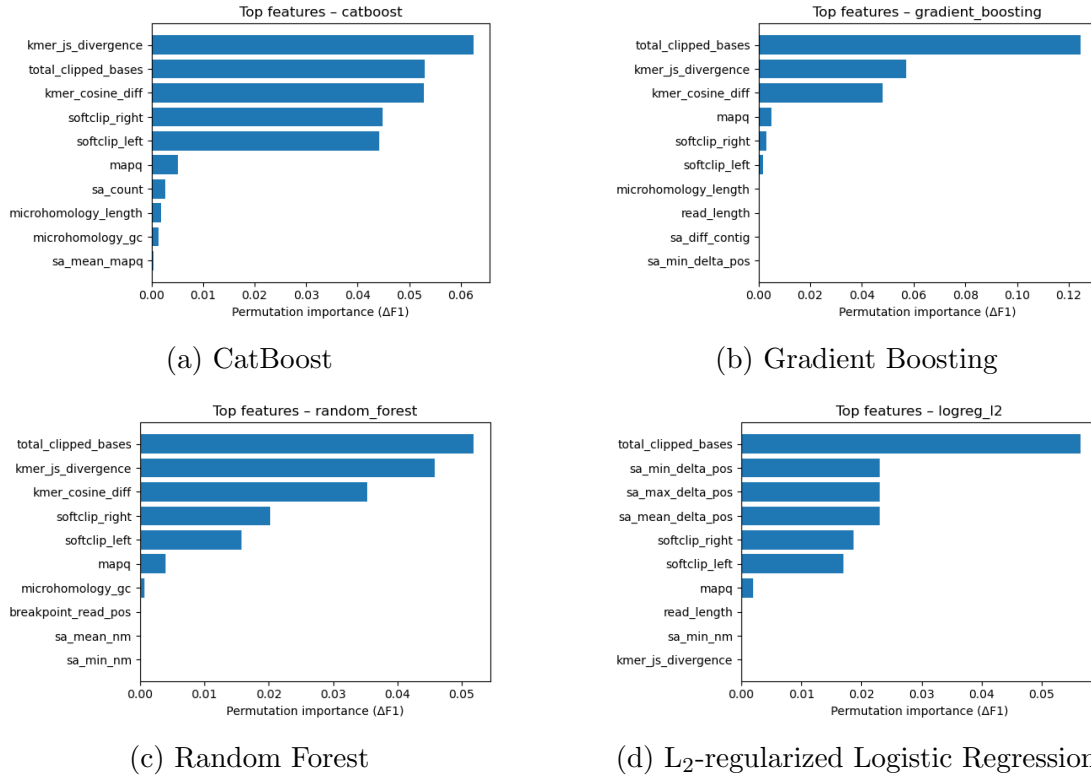


Figure 4.6: Permutation-based feature importance for four representative classifiers.

## 914 4.5.2 Feature Family Importance

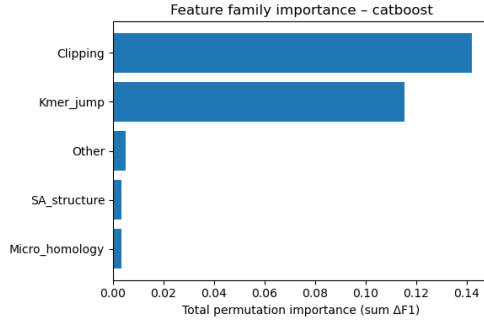
915 To evaluate the contribution of broader signals, features were grouped into  
 916 five families: SA\_structure (supplementary alignment and segment met-  
 917 rics, e.g., `has_sa`, `sa_count`, `sa_min_delta_pos`, `sa_mean_nm`, etc.), Clipping  
 918 (`softclip_left`, `softclip_right`, `total_clipped_bases`, `breakpoint_read_pos`),  
 919 Kmer\_jump (`kmer_cosine_diff`, `kmer_js_divergence`), Micro\_homology (  
 920 `microhomology_length`, `microhomology_gc`), and Other (e.g., `mapq`).

921 Aggregated analyses reveal consistent patterns across models. In CatBoost,  
 922 the Clipping family has the largest cumulative contribution (0.14), followed

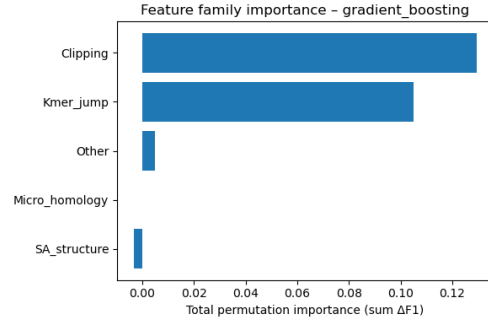


923 by Kmer\_jump (0.12), with Other features contributing minimally (0.005) and  
924 SA\_structure (0.003) and Micro\_homology (0.003) providing minimal predictive  
925 power. Gradient Boosting shows a similar trend, with Clipping (0.13) domi-  
926 nating, Kmer\_jump (0.11) secondary, and the remaining families contributing  
927 negligibly. Random Forest integrates both Clipping (0.088) and Kmer\_jump  
928 (0.08) effectively, while SA\_structure, Micro\_homology, and Other remain minor  
929 contributors. L<sub>2</sub>-regularized Logistic Regression emphasizes Clipping (0.09)  
930 and SA\_structure (0.07), with Kmer\_jump and Micro\_homology having minimal  
931 impact.

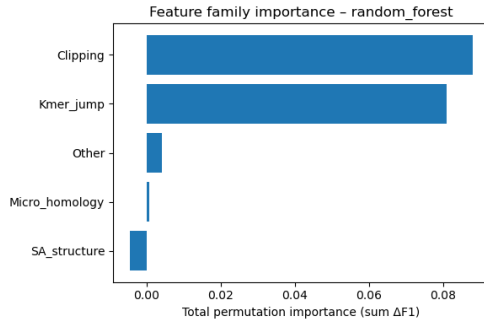
932 Both feature-level and aggregated analyses indicate that detection of chimeric  
933 reads in this dataset relies primarily on alignment irregularities (Clipping) and  
934 k-mer compositional shifts (Kmer\_jump), which often arise from PCR-induced  
935 template switching events, while explicit microhomology features contribute min-  
936 imally.



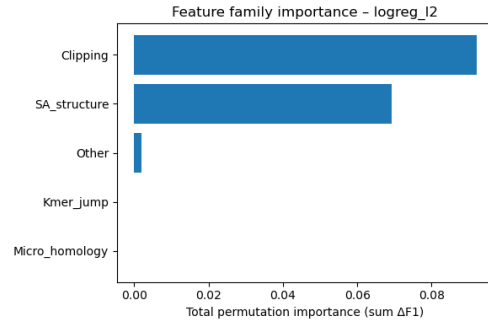
(a) CatBoost



(b) Gradient Boosting



(c) Random Forest



(d)  $L_2$ -regularized Logistic Regression

Figure 4.7: Aggregated feature family importance across four models.

## 937 4.6 Feature Selection

938 Feature selection was performed to identify the smallest subset reaching 95% cu-  
 939 mulative importance. Three models were evaluated as references: the full model  
 940 with all 23 features, a reduced model with the top- $k$  features, and an ablation  
 941 model excluding microhomology features, using a tuned CatBoost classifier to  
 942 assess feature contributions and overall classification performance.

### 4.6.1 Cumulative Importance Curve

The cumulative importance curve was computed using the tuned CatBoost classifier. Figure 4.8 illustrates the contribution of features sorted by importance. The curve rises steeply for the first few features and then gradually plateaus, indicating that a small number of features capture most of the model's predictive power. A cumulative importance of 95% is reached at  $k = 4$  features, which are `total_clipped_bases`, `kmer_js_divergence`, `kmer_cosine_diff`, and `softclip_left`.

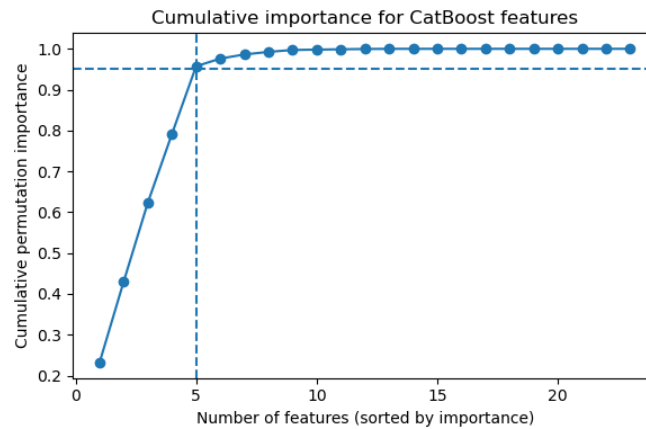


Figure 4.8: Cumulative importance curve of features sorted by importance.

### 4.6.2 Performance Comparison Across Feature Sets

Classification performance was compared across three feature sets using a tuned CatBoost classifier. The full model, incorporating all 23 engineered features, achieved an F1 score of approximately 0.7686 and a ROC-AUC of 0.8436. A reduced model using only the top four features (`total_clipped_bases`, `kmer_js_divergence`, `kmer_cosine_diff`, and `softclip_left`) achieved nearly

957 equivalent performance with an F1 of 0.7670 and a ROC-AUC of 0.8353. An  
 958 ablation model excluding microhomology features (`microhomology_length` and  
 959 `microhomology_gc`) also performed comparably, with an F1 of 0.7679 and ROC-  
 960 AUC of 0.8447. These results indicate that clipping and k-mer features capture  
 961 almost all predictive signal, while microhomology features are largely redundant  
 962 in this dataset.

Table 4.4: Test set performance of three feature set variants using tuned CatBoost.

Variant	No. of Features	Test F1	ROC-AUC
Full CatBoost	23	0.7686	0.8436
Selected (top-4)	4	0.7670	0.8353
No microhomology	21	0.7679	0.8447

963 Figure 4.9 presents a bar chart comparing F1 and ROC-AUC across the three  
 964 variants, with the x-axis showing the model variants and two bars per group  
 965 representing the F1 and ROC-AUC values.

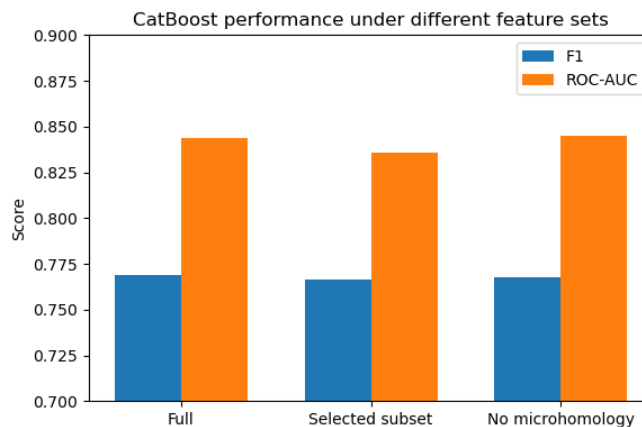


Figure 4.9: Comparison of F1 and ROC-AUC for the full, top-4 selected, and no-microhomology feature set variants.

### 966 4.6.3 Interpretation and Final Feature Set Choice

967 The full 23-feature model is retained as the primary configuration for the re-  
968 mainder of the study, while the four-feature subset serves as a lightweight al-  
969 ternative. Clipping features reflect alignment junctions and mapping disruptions  
970 typical of chimeric reads, and k-mer divergence captures changes in sequence com-  
971 position across breakpoints. Microhomology features appear largely redundant,  
972 as their signal is either indirectly represented by clipping and k-mer features or  
973 not strongly expressed in the simulation dataset.

## 974 4.7 Summary of Findings

975 All models performed substantially better than the dummy baseline, with test  
976 F1-scores around 0.76 and ROC-AUC values near 0.84. Hyperparameter tuning  
977 yielded modest improvements, with boosting methods, particularly CatBoost and  
978 gradient boosting, achieving the highest performance. Confusion matrices and  
979 precision-recall curves indicate that the models prioritize precision over recall for  
980 chimeric reads, minimizing false positives.

981 Feature importance analysis highlighted alignment breakpoints, such as clip-  
982 ping, and abrupt shifts in k-mer composition as the main contributors to predic-  
983 tive power. Microhomology metrics and supplementary alignment features had  
984 minimal impact. These findings suggest that alignment-based and k-mer-based  
985 features alone are sufficient for training classifiers to detect mitochondrial PCR-  
986 induced chimeric reads under the conditions tested.

# Appendix A

## Complete Per-Class Summary Statistics

Table A.1: Complete per-class summary statistics for all extracted features.

Feature	Class	Mean	Std	Median	IQR
breakpoint_read_pos	chimeric	75.000	0.000	75.000	0.000
breakpoint_read_pos	clean	75.000	0.000	75.000	0.000
has_sa	chimeric	0.406	0.491	0.000	1.000
has_sa	clean	0.000	0.000	0.000	0.000
kmer_cosine_diff	chimeric	0.974	0.026	0.986	0.042
kmer_cosine_diff	clean	0.976	0.025	0.986	0.041
kmer_js_divergence	chimeric	0.974	0.025	0.986	0.043
kmer_js_divergence	clean	0.976	0.025	0.986	0.040
mapq	chimeric	59.987	0.355	60.000	0.000

*Continued on next page*

Feature	Class	Mean	Std	Median	IQR
mapq	clean	59.663	2.036	60.000	0.000
mean_base_quality	chimeric	40.000	0.000	40.000	0.000
mean_base_quality	clean	13.000	0.000	13.000	0.000
microhomology_gc	chimeric	0.172	0.361	0.000	0.000
microhomology_gc	clean	0.172	0.361	0.000	0.000
microhomology_length	chimeric	0.458	0.755	0.000	1.000
microhomology_length	clean	0.462	0.758	0.000	1.000
num_segments	chimeric	1.406	0.491	1.000	1.000
num_segments	clean	1.000	0.000	1.000	0.000
read_length	chimeric	150.000	0.000	150.000	0.000
read_length	clean	150.000	0.000	150.000	0.000
ref_start_1based	chimeric	8428.635	4248.348	8433.000	6773.250
ref_start_1based	clean	8200.121	4626.918	8240.000	7926.000
sa_count	chimeric	0.406	0.491	0.000	1.000
sa_count	clean	0.000	0.000	0.000	0.000
sa_diff_contig	chimeric	0.000	0.000	0.000	0.000
sa_diff_contig	clean	0.000	0.000	0.000	0.000
sa_max_delta_pos	chimeric	1573.531	2364.996	0.000	2826.250
sa_max_delta_pos	clean	0.000	0.000	0.000	0.000
sa_max_mapq	chimeric	14.104	21.424	0.000	27.000
sa_max_mapq	clean	0.000	0.000	0.000	0.000
sa_mean_delta_pos	chimeric	1573.531	2364.996	0.000	2826.250
sa_mean_delta_pos	clean	0.000	0.000	0.000	0.000

*Continued on next page*

Feature	Class	Mean	Std	Median	IQR
sa_mean_mapq	chimeric	14.104	21.424	0.000	27.000
sa_mean_mapq	clean	0.000	0.000	0.000	0.000
sa_mean_nm	chimeric	0.022	0.319	0.000	0.000
sa_mean_nm	clean	0.000	0.000	0.000	0.000
sa_min_delta_pos	chimeric	1573.531	2364.996	0.000	2826.250
sa_min_delta_pos	clean	0.000	0.000	0.000	0.000
sa_min_nm	chimeric	0.022	0.319	0.000	0.000
sa_min_nm	clean	0.000	0.000	0.000	0.000
sa_opp_strand_count	chimeric	0.000	0.000	0.000	0.000
sa_opp_strand_count	clean	0.000	0.000	0.000	0.000
sa_same_strand_count	chimeric	0.406	0.491	0.000	1.000
sa_same_strand_count	clean	0.000	0.000	0.000	0.000
softclip_left	chimeric	12.546	21.898	0.000	19.000
softclip_left	clean	0.225	1.543	0.000	0.000
softclip_right	chimeric	12.896	22.123	0.000	19.000
softclip_right	clean	0.212	1.513	0.000	0.000
total_clipped_bases	chimeric	25.442	25.481	19.000	48.000
total_clipped_bases	clean	0.437	2.157	0.000	0.000



## 991 Appendix B

### 992 Boxplots for All Numeric 993 Features by Feature Family

#### 994 B.0.1 SA Structure (Supplementary Alignment and Seg- 995 ment Metrics)

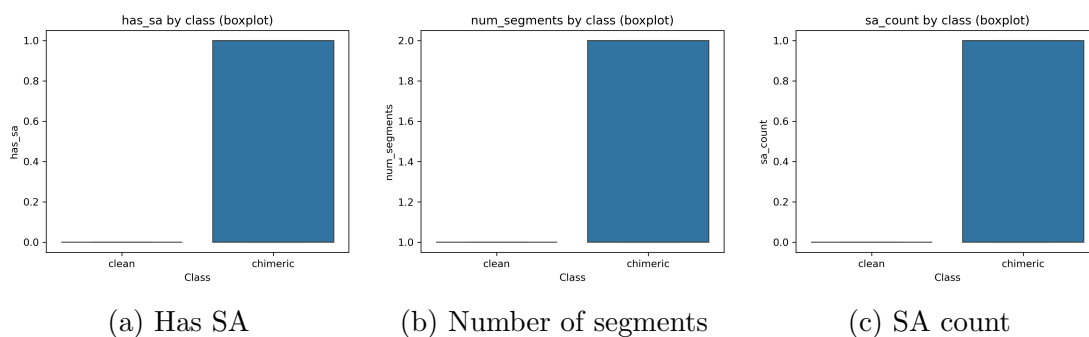
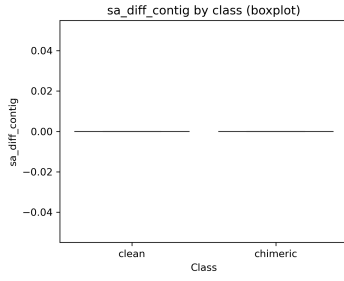
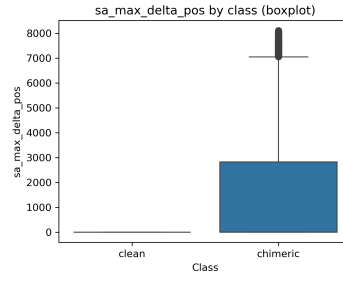


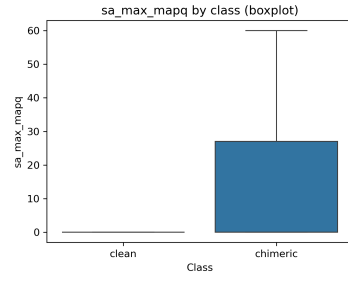
Figure B.1: Boxplots of SA Structure features by class (1/2).



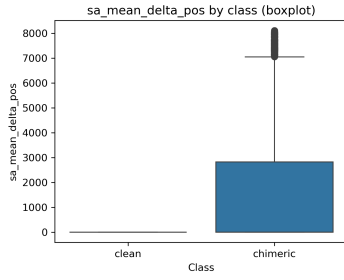
(a) SA different contig



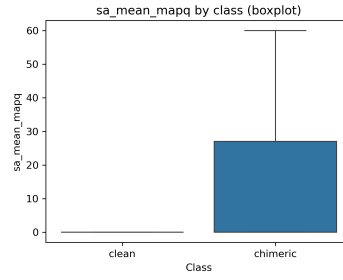
(b) SA max  $\Delta$  position



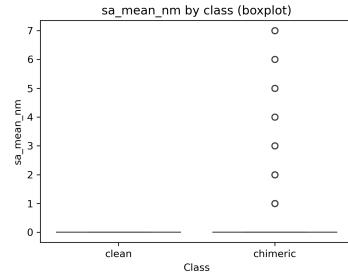
(c) SA max MAPQ



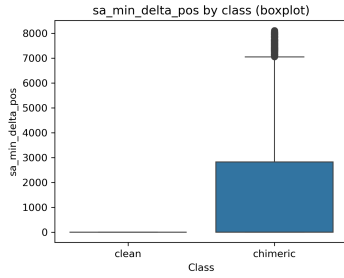
(d) SA mean  $\Delta$  position



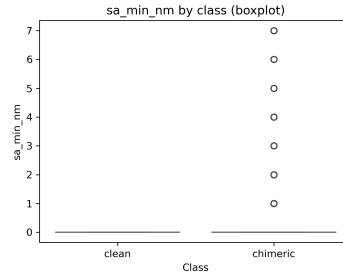
(e) SA mean MAPQ



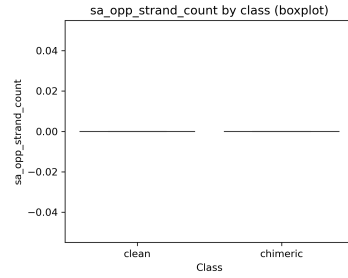
(f) SA mean NM



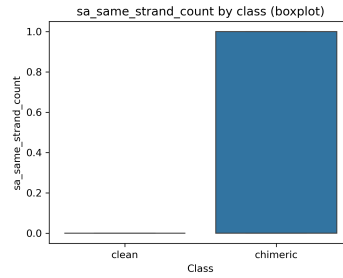
(g) SA min  $\Delta$  position



(h) SA min NM



(i) SA opposite strand count



(j) SA same strand count

Figure B.2: Boxplots of SA Structure features by class (2/2).

## 996 B.0.2 Clipping-Based Features

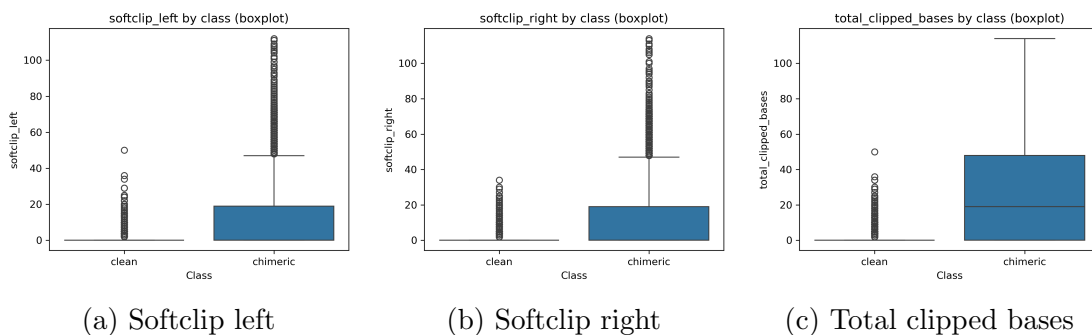


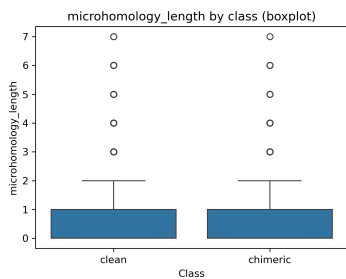
Figure B.3: Boxplots of clipping-based features by class.

## 997 B.0.3 K-mer Features

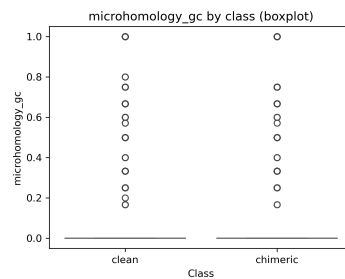


Figure B.4: Boxplots of k-mer features by class.

## 998 B.0.4 Microhomology Features



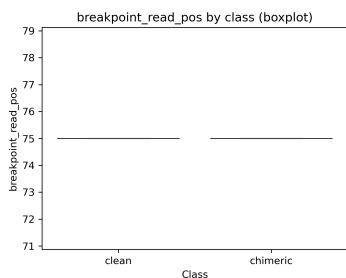
(a) Microhomology length



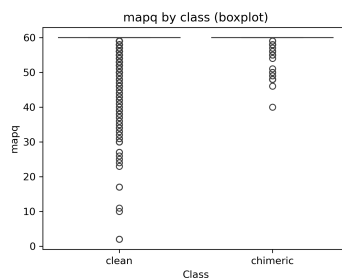
(b) Microhomology GC

Figure B.5: Boxplots of microhomology features by class.

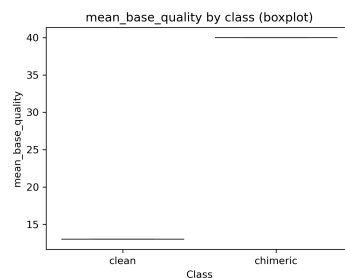
## 999 B.0.5 Others



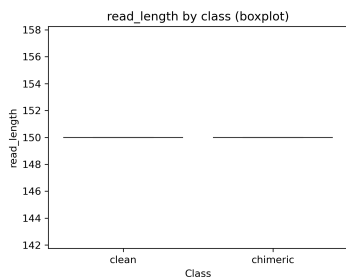
(a) Breakpoint read position



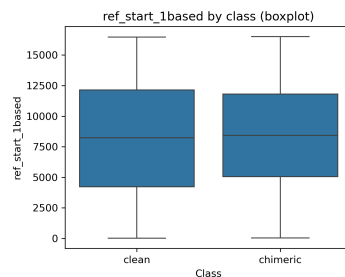
(b) MAPQ



(c) Mean base quality



(d) Read length



(e) Reference start (1-based)

Figure B.6: Boxplots of other numeric features by class.

# References

- Anderson, S., Bankier, A., Barrell, B., Bruijn, M., Coulson, A., Drouin, J., ...  
Young, I. (1981, 04). Sequence and organization of the human mitochondrial  
genome. *Nature*, 290, 457-465. doi: 10.1038/290457a0
- Arango, G., Garner, E., Pruden, A., Heath, L., Vikesland, P., & Zhang, L. (2018,  
02). Deeparg: A deep learning approach for predicting antibiotic resistance  
genes from metagenomic data. *Microbiome*, 6. doi: 10.1186/s40168-018  
-0401-z
- Bentley, D. R., Balasubramanian, S., Swerdlow, H. P., Smith, G. P., Milton, J.,  
Brown, C. G., ... Smith, A. J. (2008). Accurate whole human genome  
sequencing using reversible terminator chemistry. *Nature*, 456(7218), 53–  
59. doi: 10.1038/nature07517
- Boore, J. L. (1999). Animal mitochondrial genomes. *Nucleic Acids Research*,  
27(8), 1767–1780. doi: 10.1093/nar/27.8.1767
- Cameron, S. L. (2014). Insect mitochondrial genomics: Implications for evolution  
and phylogeny. *Annual Review of Entomology*, 59, 95–117. doi: 10.1146/  
annurev-ento-011613-162007
- Dierckxsens, N., Mardulyn, P., & Smits, G. (2017). Novoplasty: de novo assembly  
of organelle genomes from whole genome data. *Nucleic Acids Research*,

1019 45(4), e18. doi: 10.1093/nar/gkw955

1020 Edgar, R. C. (n.d.). *Uchime in practice*. Retrieved from [https://www.drive5](https://www.drive5.com/usearch/manual7/uchime_practical.html)

1021 [.com/usearch/manual7/uchime\\_practical.html](https://www.drive5.com/usearch/manual7/uchime_practical.html)

1022 Edgar, R. C. (2016). Uchime2: improved chimera prediction for amplicon se-

1023 quencing. *bioRxiv*. Retrieved from [https://api.semanticscholar.org/](https://api.semanticscholar.org/CorpusID:88955007)

1024 [CorpusID:88955007](https://api.semanticscholar.org/CorpusID:88955007)

1025 Edgar, R. C., Haas, B. J., Clemente, J. C., Quince, C., & Knight, R. (2011).

1026 Uchime improves sensitivity and speed of chimera detection. *Bioinformatics*,

1027 27(16), 2194–2200. doi: 10.1093/bioinformatics/btr381

1028 Glenn, T. C. (2011). Field guide to next-generation dna sequencers. *Molecular*

1029 *Ecology Resources*, 11(5), 759–769. doi: 10.1111/j.1755-0998.2011.03024.x

1030 Gonzalez, J. M., Zimmermann, J., & Saiz-Jimenez, C. (2004, 09). Evalu-

1031 ating putative chimeric sequences from pcr-amplified products. *Bioin-*

1032 *formatics*, 21(3), 333–337. Retrieved from [https://doi.org/10.1093/](https://doi.org/10.1093/bioinformatics/bti008)

1033 [bioinformatics/bti008](https://doi.org/10.1093/bioinformatics/bti008) doi: 10.1093/bioinformatics/bti008

1034 Gray, M. W. (2012). Mitochondrial evolution. *Cold Spring Harbor perspectives*

1035 *in biology*, 4. Retrieved from [https://doi.org/10.1101/cshperspect](https://doi.org/10.1101/cshperspect.a011403)

1036 [.a011403](https://doi.org/10.1101/cshperspect.a011403) doi: 10.1101/cshperspect.a011403

1037 Hahn, C., Bachmann, L., & Chevreux, B. (2013). Reconstructing mitochondrial

1038 genomes directly from genomic next-generation sequencing reads—a baiting

1039 and iterative mapping approach. *Nucleic Acids Research*, 41(13), e129. doi:

1040 10.1093/nar/gkt371

1041 Jin, J.-J., Yu, W.-B., Yang, J., Song, Y., dePamphilis, C. W., Yi, T.-S., & Li,

1042 D.-Z. (2020). Getorganelle: a fast and versatile toolkit for accurate de

1043 novo assembly of organelle genomes. *Genome Biology*, 21(1), 241. doi:

1044 10.1186/s13059-020-02154-5

- 1045 Judo, M. S. B., Wedel, W. R., & Wilson, B. H. (1998). Stimulation and sup-  
1046 pression of pcr-mediated recombination. *Nucleic Acids Research*, 26(7),  
1047 1819–1825. doi: 10.1093/nar/26.7.1819
- 1048 Labrador, K., Agmata, A., Palermo, J. D., Ravago-Gotanco, R., & Pante, M. J.  
1049 (2021). Mitochondrial dna reveals genetically structured haplogroups of  
1050 bali sardinella (*sardinella lemuru*) in philippine waters. *Regional Studies in*  
1051 *Marine Science*, 41, 101588. doi: 10.1016/j.rsma.2020.101588
- 1052 Li, H. (2018, 05). Minimap2: pairwise alignment for nucleotide sequences. *Bioin-*  
1053 *formatics*, 34(18), 3094–3100. Retrieved from [https://doi.org/10.1093/](https://doi.org/10.1093/bioinformatics/bty191)  
1054 [bioinformatics/bty191](https://doi.org/10.1093/bioinformatics/bty191) doi: 10.1093/bioinformatics/bty191
- 1055 Liang, Q., Bible, P. W., Liu, Y., Zou, B., & Wei, L. (2020, 02). Deepmi-  
1056 crobes: taxonomic classification for metagenomics with deep learning. *NAR*  
1057 *Genomics and Bioinformatics*, 2(1), lqaa009. Retrieved from [https://](https://doi.org/10.1093/nargab/lqaa009)  
1058 [doi.org/10.1093/nargab/lqaa009](https://doi.org/10.1093/nargab/lqaa009) doi: 10.1093/nargab/lqaa009
- 1059 Metzker, M. L. (2010). Sequencing technologies — the next generation. *Nature*  
1060 *Reviews Genetics*, 11(1), 31–46. doi: 10.1038/nrg2626
- 1061 Mysara, M., Saeys, Y., Leys, N., Raes, J., & Monsieurs, P. (2015). Catch,  
1062 an ensemble classifier for chimera detection in 16s rna sequencing stud-  
1063 ies. *Applied and Environmental Microbiology*, 81(5), 1573–1584. Retrieved  
1064 from <https://journals.asm.org/doi/abs/10.1128/aem.02896-14> doi:  
1065 10.1128/AEM.02896-14
- 1066 Peccoud, J., Lequime, S., Moltini-Conclois, I., Giraud, I., Lambrechts, L., &  
1067 Gilbert, C. (2018, 04). A survey of virus recombination uncovers canon-  
1068 ical features of artificial chimeras generated during deep sequencing li-  
1069 brary preparation. *G3 Genes—Genomes—Genetics*, 8(4), 1129–1138. Re-  
1070 trieved from <https://doi.org/10.1534/g3.117.300468> doi: 10.1534/

g3.117.300468

Qin, Y., Wu, L., Zhang, Q., Wen, C., Nostrand, J. D. V., Ning, D., ... Zhou, J. (2023). Effects of error, chimera, bias, and gc content on the accuracy of amplicon sequencing. *mSystems*, 8(6), e01025-23. Retrieved from <https://journals.asm.org/doi/abs/10.1128/msystems.01025-23> doi: 10.1128/msystems.01025-23

Qiu, X., Wu, L., Huang, H., McDonel, P. E., Palumbo, A. V., Tiedje, J. M., & Zhou, J. (2001). Evaluation of pcr-generated chimeras, mutations, and heteroduplexes with 16s rna gene-based cloning. *Applied and Environmental Microbiology*, 67(2), 880–887. doi: 10.1128/AEM.67.2.880-887.2001

Ren, J., Song, K., Deng, C., Ahlgren, N., Fuhrman, J., Li, Y., ... Sun, F. (2020, 01). Identifying viruses from metagenomic data using deep learning. *Quantitative Biology*, 8. doi: 10.1007/s40484-019-0187-4

Rodriguez-Martin, B., Palumbo, E., Marco-Sola, S., Griebel, T., Ribeca, P., Alonso, G., ... Djebali, S. (2017, 01). Chimpipes: Accurate detection of fusion genes and transcription-induced chimeras from rna-seq data. *BMC Genomics*, 18. doi: 10.1186/s12864-016-3404-9

Rognes, T., Flouri, T., Nichols, B., Quince, C., & Mahé, F. (2016). Vsearch: a versatile open source tool for metagenomics. *PeerJ*, 4, e2584. doi: 10.7717/peerj.2584

Sedlazeck, F., Rescheneder, P., Smolka, M., Fang, H., Nattestad, M., von Haeseler, A., & Schatz, M. (2018, 06). Accurate detection of complex structural variations using single-molecule sequencing. *Nature Methods*, 15. doi: 10.1038/s41592-018-0001-7

Sfeir, A., & Symington, L. S. (2015). Microhomology-mediated end joining: A back-up survival mechanism or dedicated pathway? *Trends in Biochemical*



1097 *Sciences*, 40(11), 701-714. Retrieved from <https://www.sciencedirect>  
 1098 [.com/science/article/pii/S0968000415001589](https://www.sciencedirect.com/science/article/pii/S0968000415001589) doi: <https://doi.org/>  
 1099 [10.1016/j.tibs.2015.08.006](https://doi.org/10.1016/j.tibs.2015.08.006)  
 1100 Vervier, K., Mahé, P., Tournoud, M., Veyrieras, J.-B., & Vert, J.-P. (2015,  
 1101 11). Large-scale machine learning for metagenomics sequence classifica-  
 1102 tion. *Bioinformatics*, 32(7), 1023-1032. Retrieved from <https://doi.org/>  
 1103 [10.1093/bioinformatics/btv683](https://doi.org/10.1093/bioinformatics/btv683) doi: [10.1093/bioinformatics/btv683](https://doi.org/10.1093/bioinformatics/btv683)  
 1104 Willette, D., Bognot, E., Mutia, M. T., & Santos, M. (2011). *Biology and ecology*  
 1105 *of sardines in the philippines: A review* (Vol. 13; Tech. Rep. No. 1). NFRDI  
 1106 Technical Paper Series. Retrieved from <https://nfrdi.da.gov.ph/tpjf/>  
 1107 [etc/Willette%20et%20al.%20Sardines%20Review.pdf](https://nfrdi.da.gov.ph/tpjf/etc/Willette%20et%20al.%20Sardines%20Review.pdf)