

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                     Miag-ao, 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                    December 5, 2025

# Contents

21	<b>1 Introduction</b>	<b>1</b>
22	1.1 Overview . . . . .	1
23	1.2 Problem Statement . . . . .	3
24	1.3 Research Objectives . . . . .	4
25	1.3.1 General Objective . . . . .	4
26	1.3.2 Specific Objectives . . . . .	4
27	1.4 Scope and Limitations of the Research . . . . .	5
28	1.5 Significance of the Research . . . . .	6
29	<b>2 Review of Related Literature</b>	<b>7</b>
30	2.1 The Mitochondrial Genome . . . . .	7
31	2.1.1 Mitochondrial Genome Assembly . . . . .	8

32	2.2	PCR Amplification and Chimera Formation . . . . .	9
33	2.3	Existing Traditional Approaches for Chimera Detection . . . . .	10
34	2.3.1	UCHIME . . . . .	11
35	2.3.2	UCHIME2 . . . . .	12
36	2.3.3	CATch . . . . .	13
37	2.3.4	ChimPipe . . . . .	14
38	2.4	Machine Learning Approaches for Chimera and Sequence Quality	
39		Detection . . . . .	15
40	2.4.1	Feature-Based Representations of Genomic Sequences . . .	15
41	2.5	Synthesis of Chimera Detection Approaches . . . . .	16
42	<b>3</b>	<b>Research Methodology</b>	<b>19</b>
43	3.1	Research Activities . . . . .	19
44	3.1.1	Data Collection . . . . .	20
45	3.1.2	Feature Extraction Pipeline . . . . .	24
46	3.1.3	Machine Learning Model Development . . . . .	26
47	3.1.4	Model Benchmarking, Hyperparameter Optimization, and	
48		Evaluation . . . . .	28
49	3.1.5	Feature Importance and Interpretation . . . . .	29

50	3.1.6 Validation and Testing . . . . .	30
51	3.1.7 Documentation . . . . .	31
52	3.2 Calendar of Activities . . . . .	32
53	<b>4 Results and Discussion</b>	<b>33</b>
54	4.1 Descriptive Analysis of Features . . . . .	33
55	4.1.1 Exploratory Data Analysis . . . . .	34
56	4.2 Baseline Classification Performance . . . . .	35
57	4.3 Effect of Hyperparameter Tuning . . . . .	37
58	4.4 Detailed Evaluation of Representative Models . . . . .	38
59	4.4.1 Confusion Matrices and Error Patterns . . . . .	39
60	4.4.2 ROC and Precision–Recall Curves . . . . .	40
61	4.5 Feature Importance and Biological Interpretation . . . . .	42
62	4.5.1 Permutation Importance of Individual Features . . . . .	42
63	4.5.2 Feature Family Importance . . . . .	43
64	4.6 Summary of Findings . . . . .	45
65	<b>A Histograms of Key Features</b>	<b>47</b>

# 66 List of Figures

67	3.1	Process Diagram of Special Project . . . . .	20
68	4.1	Feature correlation heatmap showing relationships among alignment-	
69		derived and sequence-derived variables. . . . .	35
70	4.2	Test F1 of all baseline classifiers, showing that no single model	
71		clearly dominates and several achieve comparable performance. . .	36
72	4.3	Comparison of test F1 (left) and ROC–AUC (right) for baseline and	
73		tuned models. Hyperparameter tuning yields small but consistent	
74		gains, particularly for tree-based ensembles. . . . .	38
75	4.4	Confusion matrices for the four representative models on the held-	
76		out test set. All models show more false negatives (chimeric reads	
77		called clean) than false positives. . . . .	40
78	4.5	ROC (left) and precision–recall (right) curves for the four represen-	
79		tative models on the held-out test set. Tree-based ensembles cluster	
80		closely, with logistic regression performing slightly but consistently	
81		worse. . . . .	41

82	4.6	Permutation-based feature importance for four representative clas-	
83		sifiers. Clipping and k-mer composition features are generally the	
84		strongest predictors, whereas microhomology and other alignment	
85		metrics contribute minimally. . . . .	43
86	4.7	Aggregated feature family importance across four models. Clipping	
87		and k-mer compositional shifts are consistently the dominant con-	
88		tributors, while SA_structure, Micro_homology, and other features	
89		contribute minimally. . . . .	45
90	A.1	Histogram plots of six key features comparing clean and chimeric	
91		reads. . . . .	48

# 92 List of Tables

93	2.1	Comparison of Chimera Detection Approaches and Tools . . . . .	17
94	3.1	Timetable of Activities . . . . .	32
95	4.1	Performance of baseline classifiers on the held-out test set. . . . .	36
96	4.2	Performance of tuned classifiers on the held-out test set. . . . .	37

# 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



111 anneal to an unrelated DNA fragment and are extended, creating recombinant  
 112 reads (Qiu et al., 2001). In mitochondrial genome assembly, such artifacts are  
 113 especially problematic because the mitochondrial genome is small, circular, and  
 114 often repetitive (Boore, 1999; Cameron, 2014). Even a small number of chimeric  
 115 or misjoined reads can reduce assembly contiguity and introduce false junctions  
 116 during organelle genome reconstruction (Dierckxsens, Mardulyn, & Smits, 2017;  
 117 Hahn, Bachmann, & Chevreux, 2013; Jin et al., 2020). Existing assembly tools  
 118 such as GetOrganelle and MITObim assume that input reads are largely free of  
 119 such artifacts (Hahn et al., 2013; Jin et al., 2020). Consequently, undetected  
 120 chimeras may produce fragmented assemblies or misidentified organellar bound-  
 121 aries. To ensure accurate reconstruction of mitochondrial genomes, a reliable  
 122 method for detecting and filtering PCR-induced chimeras before assembly is es-  
 123 sential.

124 This study focuses on mitochondrial sequencing data from the genus *Sar-*  
 125 *dinella*, a group of small pelagic fishes widely distributed in Philippine waters.  
 126 Among them, *Sardinella lemuru* (Bali sardinella) is one of the country’s most  
 127 abundant and economically important species, providing protein and livelihood  
 128 to coastal communities (Labrador, Agmata, Palermo, Ravago-Gotanco, & Pante,  
 129 2021; Willette, Bognot, Mutia, & Santos, 2011). Accurate mitochondrial assem-  
 130 blies are critical for understanding its population genetics, stock structure, and  
 131 evolutionary history. However, assembly pipelines often encounter errors or fail  
 132 to complete due to undetected chimeric reads. To address this gap, this research  
 133 introduces MitoChime, a machine learning pipeline designed to detect and filter  
 134 PCR-induced chimeric reads using both alignment-based and sequence-derived  
 135 statistical features. The tool aims to provide bioinformatics laboratories, partic-

136 ularly the Philippine Genome Center Visayas (PGC Visayas), with an efficient  
137 solution for improving mitochondrial genome reconstruction.

## 138 1.2 Problem Statement

139 While NGS technologies have revolutionized genomic data acquisition, the ac-  
140 curacy of mitochondrial genome assembly remains limited by artifacts produced  
141 during PCR amplification. These chimeric reads can distort assembly graphs and  
142 cause misassemblies, with particularly severe effects in small, circular mitochon-  
143 drial genomes (Boore, 1999; Cameron, 2014). Existing assembly pipelines such  
144 as GetOrganelle, MITObim, and NOVOPlasty assume that sequencing reads are  
145 free of such artifacts (Dierckxsens et al., 2017; Hahn et al., 2013; Jin et al., 2020).  
146 At PGC Visayas, several mitochondrial assemblies have failed or yielded incom-  
147 plete contigs despite sufficient coverage, suggesting that undetected chimeric reads  
148 compromise assembly reliability. Meanwhile, existing chimera detection tools such  
149 as UCHIME and VSEARCH were developed primarily for amplicon-based com-  
150 munity analysis and rely heavily on reference or taxonomic comparisons (Edgar,  
151 Haas, Clemente, Quince, & Knight, 2011; Rognes, Flouri, Nichols, Quince, &  
152 Mahé, 2016). These approaches are unsuitable for single-species organellar data,  
153 where complete reference genomes are often unavailable. Therefore, there is a  
154 pressing need for a reference-independent, data-driven tool capable of detecting  
155 and filtering PCR-induced chimeras in mitochondrial sequencing datasets.

## 156 1.3 Research Objectives

### 157 1.3.1 General Objective

158 This study aims to develop and evaluate a machine learning-based pipeline (Mi-  
159 toChime) that detects PCR-induced chimeric reads in *Sardinella lemuru* mito-  
160 chondrial sequencing data in order to improve the quality and reliability of down-  
161 stream mitochondrial genome assemblies.

### 162 1.3.2 Specific Objectives

163 Specifically, the study aims to:

- 164 1. construct simulated *Sardinella lemuru* Illumina paired-end datasets contain-  
165 ing both clean and PCR-induced chimeric reads,
- 166 2. extract alignment-based and sequence-based features such as k-mer compo-  
167 sition, junction complexity, and split-alignment counts from both clean and  
168 chimeric reads,
- 169 3. train, validate, and compare supervised machine-learning models for classi-  
170 fying reads as clean or chimeric,
- 171 4. determine feature importance and identify indicators of PCR-induced  
172 chimerism,
- 173 5. integrate the optimized classifier into a modular and interpretable pipeline  
174 deployable on standard computing environments at PGC Visayas.

## 1.4 Scope and Limitations of the Research

This study focuses on detecting PCR-induced chimeric reads in Illumina paired-end mitochondrial sequencing data from *Sardinella lemuru*. The decision to restrict the taxonomic scope to a single species is based on four considerations: to limit interspecific variation in mitochondrial genome size, GC content, and repetitive regions so that differences in read patterns can be attributed more directly to PCR-induced chimerism; to align the analysis with relevant *S. lemuru* sequencing projects at PGC Visayas; to take advantage of the availability of *S. lemuru* mitochondrial assemblies and raw datasets in public repositories such as the National Center for Biotechnology Information (NCBI), which facilitates reference selection and benchmarking; and to develop a tool that directly supports local studies on *S. lemuru* population structure and fisheries management.

The study emphasizes `wgsim`-based simulations and selected empirical mitochondrial datasets from *S. lemuru*. It excludes naturally occurring chimeras, nuclear mitochondrial pseudogenes (NUMTs), and large-scale assembly rearrangements in nuclear genomes. Feature extraction is restricted to low-dimensional alignment and sequence statistics, such as k-mer frequency profiles, GC content, read length, soft and hard clipping metrics, split-alignment counts, and mapping quality, rather than high-dimensional deep learning embeddings. This design keeps model behaviour interpretable and ensures that the pipeline can be run on standard workstations at PGC Visayas. Testing on long-read platforms (e.g., Nanopore, PacBio) and other taxa is outside the scope of this project; the implemented pipeline is evaluated only on short-read *S. lemuru* datasets.

Other limitations in this study include the following: simulations with varying

199 error rates were not performed, so the effect of different sequencing errors on model  
200 performance remains unexplored; alternative parameter settings, including k-mer  
201 lengths and microhomology window sizes, were not systematically tested, which  
202 could affect the sensitivity of both k-mer and microhomology feature detection as  
203 well as the identification of chimeric junctions; and the machine-learning models  
204 rely on supervised training with labeled examples, which may limit their ability  
205 to detect novel or unexpected chimeric patterns.

## 206 1.5 Significance of the Research

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

## 217 Chapter 2

## 218 Review of Related Literature

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

### 225 2.1 The Mitochondrial Genome

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

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

### 236 **2.1.1 Mitochondrial Genome Assembly**

237 Mitochondrial genome assembly refers to the reconstruction of the complete mito-  
238 chondrial DNA (mtDNA) sequence from raw or fragmented sequencing reads. It is  
239 conducted to obtain high-quality, continuous representations of the mitochondrial  
240 genome that can be used for a wide range of analyses, including species identi-  
241 fication, phylogenetic reconstruction, evolutionary studies, and investigations of  
242 mitochondrial diseases. Because mtDNA evolves rapidly, its assembled sequence  
243 provides valuable insights into population structure, lineage divergence, and adap-  
244 tive evolution across taxa (Boore, 1999). Compared to nuclear genome assembly,  
245 assembling the mitochondrial genome is often considered more straightforward but  
246 still encounters technical challenges such as the formation of chimeric reads. Com-  
247 monly used tools for mitogenome assembly such as GetOrganelle and MITObim  
248 operate under the assumption of organelle genome circularity, and are vulnerable  
249 when chimeric reads disrupt this circular structure, resulting in assembly errors  
250 (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

295 sequence correspond to distinct high-abundance sequences.

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

### 301 **2.3.1 UCHIME**

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

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

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

### 323 **2.3.2 UCHIME2**

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

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

### 340 2.3.3 CATCh

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

346 To address these inconsistencies, Mysara, Saeys, Leys, Raes, and Monsieurs  
347 (2015) developed the Classifier for Amplicon Tool Chimeras (CATCh), which rep-  
348 resents the first ensemble machine learning system designed for chimera detection  
349 in 16S rRNA amplicon sequencing. Rather than depending on a single detec-  
350 tion strategy, CATCh integrates the outputs of several established tools, includ-  
351 ing UCHIME, ChimeraSlayer, DECIPHER, Pintail, and Perseus. The individual  
352 scores and binary decisions generated by these tools are used as input features for  
353 a supervised learning model. The algorithm employs a Support Vector Machine  
354 (SVM) with a Pearson VII Universal Kernel (PUK) to determine optimal weight-  
355 ings among the input features and to assign each sequence a probability of being  
356 chimeric.

357 Benchmarking in both reference-based and de novo modes demonstrated signif-  
358 icant performance improvements. CATCh achieved sensitivities of approximately  
359 85 percent in reference-based mode and 92 percent in de novo mode, with corre-  
360 sponding specificities of approximately 96 percent and 95 percent. These results  
361 indicate that CATCh detected 7 to 12 percent more chimeras than any individual  
362 algorithm while maintaining high precision.

### 363 2.3.4 ChimPipe

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

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

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

## 381 2.4 Machine Learning Approaches for Chimera 382 and Sequence Quality Detection

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

### 394 2.4.1 Feature-Based Representations of Genomic Se- 395 quences

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

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

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

412 Microhomology, or short sequences shared between adjacent segments, is an-  
413 other biologically meaningful feature. Its length, typically a few to tens of base  
414 pairs, has been linked to microhomology-mediated repair and template-switching  
415 mechanisms (Sfeir & Symington, 2015). In PCR-induced chimeras, short iden-  
416 tical sequences at junctions provide a clear signature of chimerism. Measuring  
417 the longest exact overlap at each breakpoint complements k-mer and alignment  
418 features and helps identify reads that are potentially chimeric.

## 419 **2.5 Synthesis of Chimera Detection Approaches**

420 To provide an integrated overview of the literature discussed in this chapter, Ta-  
421 ble 2.1 summarizes the major chimera detection studies, their methodological  
422 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.



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

## 435 Chapter 3

# 436 Research Methodology

437 This chapter outlines the steps involved in completing the study, including data  
438 gathering, generating simulated mitochondrial Illumina reads, preprocessing and  
439 indexing the data, developing a feature extraction pipeline to extract key features,  
440 applying machine learning algorithms for chimera detection, and validating and  
441 comparing model performance.

## 442 3.1 Research Activities

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

449 through a feature extraction pipeline that extracted k-mer profiles, supplementary  
 450 alignment (SA) features, and microhomology information to prepare the data for  
 451 model construction. The machine learning model was trained using the processed  
 452 input, and its precision and accuracy were assessed. It underwent tuning until it  
 453 reached the desired performance threshold, after which it proceeded to validation  
 454 and will undergo testing.

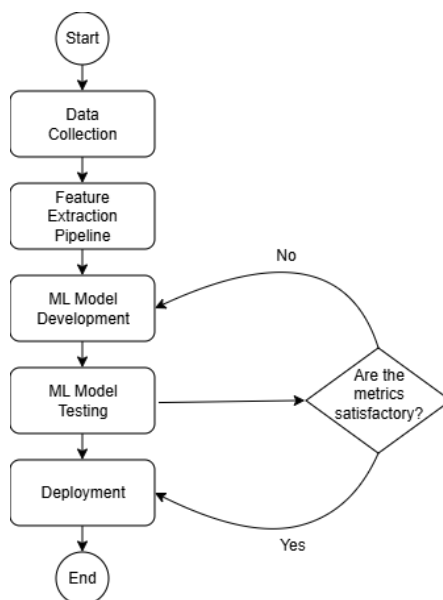


Figure 3.1: Process Diagram of Special Project

### 455 3.1.1 Data Collection

456 The mitochondrial genome reference sequence of *S. lemuru* was obtained from the  
 457 NCBI database (accession number NC\_039553.1) in FASTA format. This sequence  
 458 served as the basis for generating simulated reads for model development.

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

461 proximately one (1) week.

## 462 Data Preprocessing

463 To reduce manual repetition, all steps in the simulation and preprocessing pipeline  
464 were executed using a custom script in Python (Version 3.11). The script runs  
465 each stage, including read simulation, reference indexing, mapping, and alignment  
466 processing, in a fixed sequence.

467 Sequencing data were simulated from the NCBI reference genome using `wgsim`  
468 (Version 1.13). First, a total of 10,000 paired-end fragments were simulated,  
469 producing 20,000 reads (10,000 forward and 10,000 reverse) from the the original  
470 reference (`original_reference.fasta`) and and designated as clean reads using  
471 the command:

```
472 wgsim -1 150 -2 150 -r 0 -R 0 -X 0 -e 0.001 -N 10000 \  
473         original_reference.fasta ref1.fastq ref2.fastq
```

474 The command parameters are as follows:

- 475 • `-1` and `-2`: read lengths of 150 base pairs for each paired-end read.
- 476 • `-r`, `-R`, `-X`: mutation rate, fraction of indels, and indel extension probability,  
477 all set to a default value of 0.
- 478 • `-e`: base error rate, set to 0.001 to simulate realistic sequencing errors.
- 479 • `-N`: number of read pairs, set to 10,000.

480 Chimeric sequences were then generated from the same NCBI reference using a  
481 separate Python script. Two non-adjacent segments were randomly selected such  
482 that their midpoint distances fell within specified minimum and maximum thresh-  
483 olds. The script attempts to retain microhomology, or short identical sequences  
484 at segment junctions, to mimic PCR-induced template switching. The resulting  
485 chimeras were written to `chimera_reference.fasta`, with headers recording seg-  
486 ment positions and microhomology length. The `chimera_reference.fasta` was  
487 processed with `wgsim` to simulate 10,000 paired-end fragments, generating 20,000  
488 chimeric reads (10,000 forward reads in `chimeric1.fastq` and 10,000 reverse reads  
489 in `chimeric2.fastq`) using the command format.

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

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

492 Minimap2 (Version 2.28) is a tool used to map reads to a reference genome.  
493 The index `ref.mmi` of the original reference sequence is required by `minimap2` for  
494 efficient read mapping. Mapping allows extraction of alignment features from each  
495 read, which were used as input for the machine learning model. The simulated  
496 clean and chimeric reads were then mapped to the reference index as follows:

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

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

500 Here, `-ax sr` specifies short-read alignment mode, and `-t 8` uses 8 CPU

501 threads. The resulting clean and chimeric SAM files contain the alignment posi-  
502 tions of each read relative to the original reference genome.

503 The SAM files were then converted to BAM format, sorted, and indexed using  
504 **samtools** (Version 1.20):

```
505 samtools view -bS clean.sam -o clean.bam
506 samtools view -bS chimeric.sam -o chimeric.bam
507
508 samtools sort clean.bam -o clean.sorted.bam
509 samtools index clean.sorted.bam
510
511 samtools sort chimeric.bam -o chimeric.sorted.bam
512 samtools index chimeric.sorted.bam
```

513 BAM files are the compressed binary version of SAM files, which enables faster  
514 processing and reduced storage. Sorting arranges reads by genomic coordinates,  
515 and indexing allows detection of SA as a feature for the machine learning model.

516 The total number of simulated reads was expected to be 40,000. The final col-  
517 lection of reads contained 19,984 clean reads and 20,000 chimeric reads (39,984 en-  
518 tries in total), providing a roughly balanced distribution between the two classes.  
519 After alignment with **minimap2**, only 19,984 clean reads remained because un-  
520 mapped reads were not included in the BAM file. Some sequences failed to align  
521 due to the 5% error rate defined during **wgsim** simulation, which produced mis-  
522 matches that caused certain reads to fall below the aligner's matching threshold.

523 This whole process is scheduled to start in the second week of November 2025

524 and is expected to be completed by the last week of November 2025, with a total  
525 duration of approximately three (3) weeks.

### 526 **3.1.2 Feature Extraction Pipeline**

527 This stage directly follows the previous alignment phase, utilizing the resulting  
528 BAM files (specifically `chimeric.sorted.bam` and `clean.sorted.bam`). A custom  
529 Python script was created to efficiently process each primary-mapped read to  
530 extract the necessary set of analytical features, which are then compiled into a  
531 structured feature matrix in TSV format. The pipeline's core functionality relies on  
532 libraries, namely `Pysam` (Version 0.22) for the robust parsing of BAM structures and  
533 `NumPy` (Version 1.26) for array operations and computations. To ensure correctness  
534 and adherence to best practices, bioinformatics experts at the PGC Visayas will  
535 be consulted to validate the pipeline design, feature extraction logic, and overall  
536 data integrity. This stage of the study was scheduled to begin in the third week  
537 of November 2025 and conclude by the first week of December 2025, with an  
538 estimated total duration of approximately three (3) weeks.

539 The pipeline focuses on three features that collectively capture biological sig-  
540 natures associated with PCR-induced chimeras: (1) Supplementary alignment flag  
541 (SA count), (2) k-mer composition difference, and (3) microhomology.

#### 542 **Supplementary Alignment Flag**

543 Split-alignment information was derived from the SA (Supplementary Alignment)  
544 tag embedded in each primary read of the BAM file. This tag is typically asso-

545 ciated with reads that map to multiple genomic locations, suggesting a chimeric  
546 structure. To extract this information, the script first checked whether the read  
547 carried an `SA:Z` tag. If present, the tag string was parsed using the function  
548 `parse_sa_tag`, yielding a structure for each alignment containing the reference  
549 name, mapped position, strand, mapping quality, and number of mismatches.

550 After parsing, the function `sa_feature_stats` was applied to establish the fun-  
551 damental split indicators, `has_sa` and `sa_count`. Along with these initial counts,  
552 the function synthesized a summarization by aggregating metrics related to the  
553 structure and reliability of the split alignments.

## 554 **K-mer Composition Difference**

555 Chimeric reads often comprise fragments from distinct genomic regions, resulting  
556 in a compositional discontinuity between segments. Comparing k-mer frequency  
557 profiles between the left and right halves of a read allows for the detection of such  
558 abrupt compositional shifts, independent of alignment information.

559 The script implemented this by inferring a likely junction breakpoint using  
560 the function `infer_breakpoints`, prioritizing the boundaries defined by soft-  
561 clipping operations in the `CIGAR` string. If no clipping was present, the midpoint  
562 of the alignment or the read length was utilized as a fallback. The read sequence  
563 was then divided into left and right segments at this inferred breakpoint, and  
564 k-mer frequency profiles ( $k = 5$ ) were generated for both halves, ignoring any  
565 k-mers containing ambiguous 'N' bases. The resulting k-mer frequency vectors  
566 will be normalized and compared using the functions `cosine_difference` and  
567 `js_divergence`.



## 568 Microhomology

569 The process of extracting the microhomology feature started by utilizing the func-  
570 tion `infer_breakpoints` similar to the k-mer workflow. Once a breakpoint was es-  
571 tablished, the script scanned a  $\pm 40$  base pair window surrounding the breakpoint  
572 and used the function `longest_suffix_prefix_overlap` to identify the longest  
573 exact suffix-prefix overlap between the left and right read segments. This overlap,  
574 which represents consecutive bases shared at the junction, was recorded as the  
575 `microhomology_length` in the dataset. The 40-base pair window was chosen to  
576 ensure that short shared sequences at or near the breakpoint were captured, with-  
577 out including distant sequences that are unrelated. Additionally, the GC content  
578 of the overlapping sequence was calculated using the function `gc_content`, which  
579 counts guanine (G) and cytosine (C) bases within the detected microhomology  
580 and divides by the total length, yielding a proportion between 0 and 1, and was  
581 stored under the `microhomology_gc` attribute. Short microhomologies, typically  
582 3-20 base pairs in length, are recognized signatures of PCR-induced template  
583 switching (Peccoud et al., 2018).

584 A k-mer length of 6 was used to capture patterns within the same 40-base pair  
585 window surrounding each breakpoint. These profiles complement microhomology  
586 measurements and help identify junctions that are potentially chimeric.

### 587 3.1.3 Machine Learning Model Development

588 After feature extraction, the per-read feature matrices for clean and chimeric  
589 reads were merged into a single dataset. Each row corresponded to one paired-

590 end read, and columns encoded alignment-structure features (e.g., supplementary  
591 alignment count and spacing between segments), CIGAR-derived soft-clipping  
592 statistics (e.g., left and right soft-clipped length, total clipped bases), k-mer com-  
593 position discontinuity between read segments, and microhomology descriptors  
594 near candidate junctions. The resulting feature set was restricted to quantities  
595 that can be computed from standard BAM/FASTQ files in typical mitochondrial  
596 sequencing workflows.

597 The labelled dataset was randomly partitioned into training (80%) and test  
598 (20%) subsets using stratified sampling to preserve the 1:1 ratio of clean to  
599 chimeric reads. Model development and evaluation were implemented in Python  
600 (Version 3.11) using the `scikit-learn`, `xgboost`, `lightgbm`, and `catboost` li-  
601 braries. A broad panel of classification algorithms was then benchmarked on the  
602 training data to obtain a fair comparison of different model families under identical  
603 feature conditions. The panel included: a trivial dummy classifier, L2-regularized  
604 logistic regression, a calibrated linear support vector machine (SVM),  $k$ -nearest  
605 neighbours, Gaussian Naïve Bayes, decision-tree ensembles (Random Forest, Ex-  
606 tremely Randomized Trees, and Bagging with decision trees), gradient boosting  
607 methods (Gradient Boosting, XGBoost, LightGBM, and CatBoost), and a shallow  
608 multilayer perceptron (MLP).

609 For each model, five-fold stratified cross-validation was performed on the train-  
610 ing set. In every fold, four-fifths of the data were used for fitting and the remaining  
611 one-fifth for validation. Mean cross-validation accuracy, precision, recall, F1-score  
612 for the chimeric class, and area under the receiver operating characteristic curve  
613 (ROC-AUC) were computed to summarize performance and rank candidate meth-  
614 ods. This baseline screen allowed comparison of linear, probabilistic, neural, and

ensemble-based approaches and identified tree-based ensemble and boosting models as consistently strong performers relative to simpler baselines.

### 3.1.4 Model Benchmarking, Hyperparameter Optimization, and Evaluation

Model selection and refinement proceeded in two stages. First, the cross-validation results from the broad panel were used to identify a subset of competitive models for more detailed optimization. Specifically, ten model families were carried forward: L2-regularized logistic regression, calibrated linear SVM, Random Forest, ExtraTrees, Gradient Boosting, XGBoost, LightGBM, CatBoost, Bagging with decision trees, and a shallow MLP. This subset spans both linear and non-linear decision boundaries, but emphasizes ensemble and boosting methods, which showed superior F1 and ROC-AUC in the initial benchmark.

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

For each model family, the hyperparameter configuration with the highest mean cross-validation F1-score was selected as the best-tuned estimator. These tuned models were then refitted on the full training set and evaluated once on the held-out test set to obtain unbiased estimates of performance. Test-set metrics included accuracy, precision, recall, F1-score for the chimeric class, and ROC-AUC. Confusion matrices and ROC curves were generated for the top-performing models to characterise common error modes, such as false negatives (missed chimeric reads) and false positives (clean reads incorrectly labelled as chimeric). The final model or small set of models for downstream interpretation was chosen based on a combination of test-set F1-score, ROC-AUC, and practical considerations such as model complexity and ease of deployment within a feature extraction pipeline.

### 3.1.5 Feature Importance and Interpretation

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

For interpretability, individual features were grouped into four conceptual

660 families: (i) supplementary alignment and alignment-structure features (e.g., SA  
661 count, spacing between alignment segments, strand consistency), (ii) CIGAR-  
662 derived soft-clipping features (e.g., left and right soft-clipped length, total clipped  
663 bases), (iii) k-mer composition discontinuity features (e.g., cosine distance and  
664 Jensen–Shannon divergence between k-mer profiles of read segments), and (iv) mi-  
665 crohomology descriptors (e.g., microhomology length and local GC content around  
666 putative breakpoints). Aggregating permutation importance scores within each  
667 family allowed assessment of which biological signatures contributed most strongly  
668 to the classifier’s performance. This analysis provided a basis for interpreting the  
669 trained models in terms of known mechanisms of PCR-induced template switching  
670 and for identifying which alignment- and sequence-derived cues are most informa-  
671 tive for distinguishing chimeric from clean mitochondrial reads.

### 672 **3.1.6 Validation and Testing**

673 Validation will involve both internal and external evaluations. Internal valida-  
674 tion was achieved through five-fold cross-validation on the training data to verify  
675 model generalization and reduce variance due to random sampling. External vali-  
676 dation will be achieved through testing on the 20% hold-out dataset derived from  
677 the simulated reads, which will be an unbiased benchmark to evaluate how well  
678 the trained models generalized to unseen data. All feature extraction and prepro-  
679 cessing steps were performed using the same feature extraction pipeline to ensure  
680 consistency and comparability across validation stages.

681 Comparative evaluation was performed across all candidate algorithms, in-  
682 cluding a trivial dummy classifier, L2-regularized logistic regression, a calibrated

683 linear SVM, k-nearest neighbours, Gaussian Naïve Bayes, decision-tree ensembles,  
684 gradient boosting methods, and a shallow MLP. This evaluation determined which  
685 models demonstrated the highest predictive performance and computational effi-  
686 ciency under identical data conditions. Their metrics were compared to identify  
687 which algorithms were most suitable for further refinement.

### 688 **3.1.7 Documentation**

689 Comprehensive documentation was maintained throughout the study to ensure  
690 transparency and reproducibility. All stages of the research, including data gath-  
691 ering, preprocessing, feature extraction, model training, and validation, were sys-  
692 tematically recorded in a `.README` file in the GitHub repository. For each ana-  
693 lytical step, the corresponding parameters, software versions, and command line  
694 scripts were documented to enable exact replication of results.

695 The repository structure followed standard research data management prac-  
696 tices, with clear directories for datasets and scripts. Computational environments  
697 were standardized using Conda, with an environment file (`environment.arm.yml`)  
698 specifying dependencies and package versions to maintain consistency across sys-  
699 tems.

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

## 702 3.2 Calendar of Activities

703 Table 3.1 presents the project timeline in the form of a Gantt chart, where each  
 704 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	• • • •	• • • •	• • • •	• • • •	• • • •	• • • •	• • • •

# Chapter 4

## Results and Discussion

### 4.1 Descriptive Analysis of Features

This chapter presents the performance of the proposed feature set and machine-learning models for detecting PCR-induced chimeric reads in simulated mitochondrial Illumina data. We first describe the behaviour of the main features, then compare baseline classifiers, assess the effect of hyperparameter tuning, and finally analyse feature importance in terms of individual variables and biologically motivated feature families.

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



### 717 4.1.1 Exploratory Data Analysis

718 An exploratory data analysis (EDA) was conducted on the extracted feature ma-  
719 trix to characterize general patterns in the data and gain preliminary insight into  
720 which variables might meaningfully contribute to classification. Histograms of  
721 key features indicated that alignment-based variables showed clear class separa-  
722 tion as chimeric reads have higher frequencies of split alignments and and no-  
723 ticeably broader long-tailed distribution on soft-clipped regions (`softclip_left`  
724 and `softclip_right`). In contrast, sequence-based variables such a microhomol-  
725 ogy length and k-mer divergence displayed substantial overlap between classes,  
726 suggesting more limited discriminative value. The complete set of histograms is  
727 provided in Appendix 4.6.

728 As shown in Figure 4.1, the feature correlation heatmap shows that alignment-  
729 derived variables form a strongly correlated cluster, whereas sequence-derived  
730 measures show weak correlations with both the alignment-based features and with  
731 one another. This heterogeneity indicates that no single feature family captures  
732 all relevant signal sources.

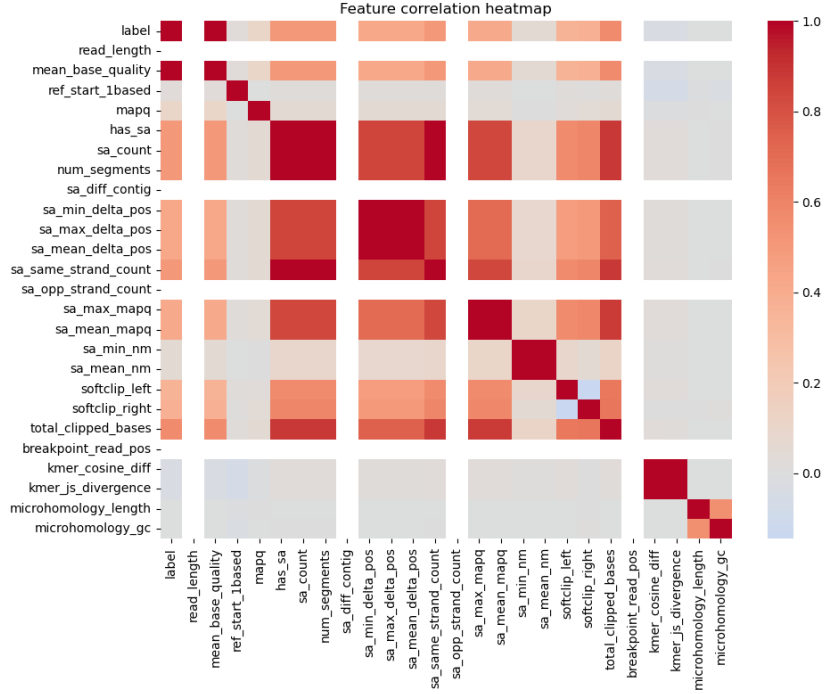


Figure 4.1: Feature correlation heatmap showing relationships among alignment-derived and sequence-derived variables.

## 4.2 Baseline Classification Performance

Table 4.1 summarises the performance of eleven classifiers trained on the engineered feature set using five-fold cross-validation and evaluated on the held-out test set. All models were optimised using default hyperparameters, without dedicated tuning.

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

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

Table 4.1: 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

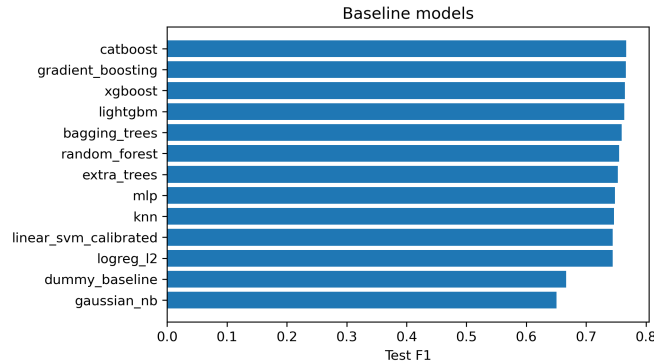


Figure 4.2: Test F1 of all baseline classifiers, showing that no single model clearly dominates and several achieve comparable performance.

### 4.3 Effect of Hyperparameter Tuning

To assess whether performance could be improved further, ten model families underwent randomised hyperparameter search (Chapter 3). The tuned metrics are summarised in Table 4.2. Overall, tuning yielded modest but consistent gains for tree-based ensembles and boosting methods, while leaving linear models essentially unchanged or slightly worse.

CatBoost, gradient boosting, LightGBM, XGBoost, random forest, bagging trees, and MLP all experienced small increases in test F1 (typically  $\Delta F1 \approx 0.002$ – $0.009$ ) and ROC–AUC (up to  $\Delta AUC \approx 0.008$ ). After tuning, CatBoost remained the best performer with test accuracy 0.802, precision 0.924, recall 0.658, F1-score 0.769, and ROC–AUC 0.844. Gradient boosting achieved almost identical performance (F1 0.767, AUC 0.843). Random forest and bagging trees also improved to F1 scores around 0.763 with  $AUC \approx 0.842$ .

Table 4.2: 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

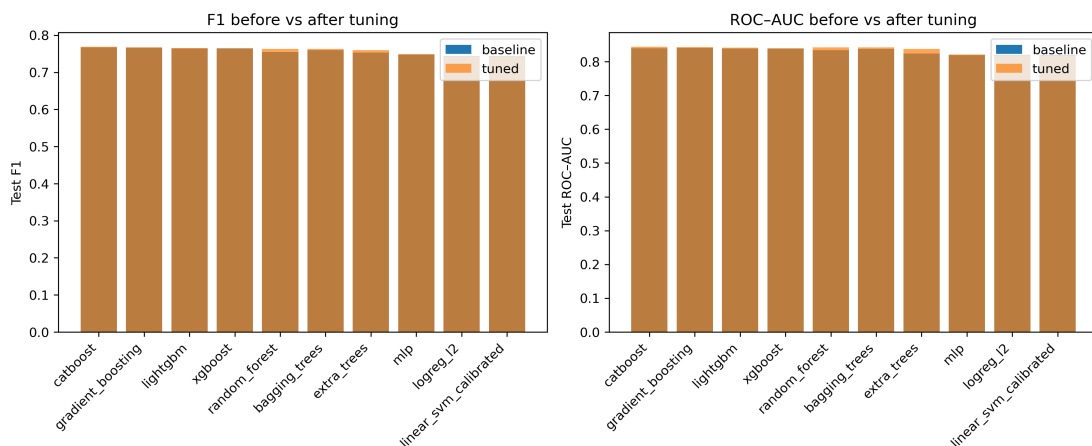


Figure 4.3: Comparison of test F1 (left) and ROC-AUC (right) for baseline and tuned models. Hyperparameter tuning yields small but consistent gains, particularly for tree-based ensembles.

Because improvements are small and within cross-validation variability, we interpret tuning as stabilising and slightly refining the models rather than fundamentally altering their behaviour or their relative ranking.

## 4.4 Detailed Evaluation of Representative Models

For interpretability and diversity, four tuned models were selected for deeper analysis: CatBoost (best-performing boosted tree), scikit-learn gradient boosting (canonical gradient-boosting implementation), random forest (non-boosted ensemble baseline), and L2-regularised logistic regression (linear baseline). All models were trained on the engineered feature set and evaluated on the same held-out test data.

#### 774 4.4.1 Confusion Matrices and Error Patterns

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

781 Random forest attained slightly lower accuracy (0.80) and chimeric F1 (0.76),  
782 whereas logistic regression achieved the lowest accuracy among the four (0.79)  
783 and chimeric F1 (0.74), although it provided the highest chimeric precision (0.95)  
784 at the cost of lower recall (0.61).

785 Across all models, errors were asymmetric. False negatives (chimeric reads  
786 predicted as clean) were more frequent than false positives. For example, CatBoost  
787 misclassified 1 369 chimeric reads as clean but only 215 clean reads as chimeric.  
788 This pattern indicates that the models are conservative: they prioritise avoiding  
789 spurious chimera calls at the expense of missing some true chimeras. Depending on  
790 downstream application, alternative decision thresholds or cost-sensitive training  
791 could be explored to adjust this balance.

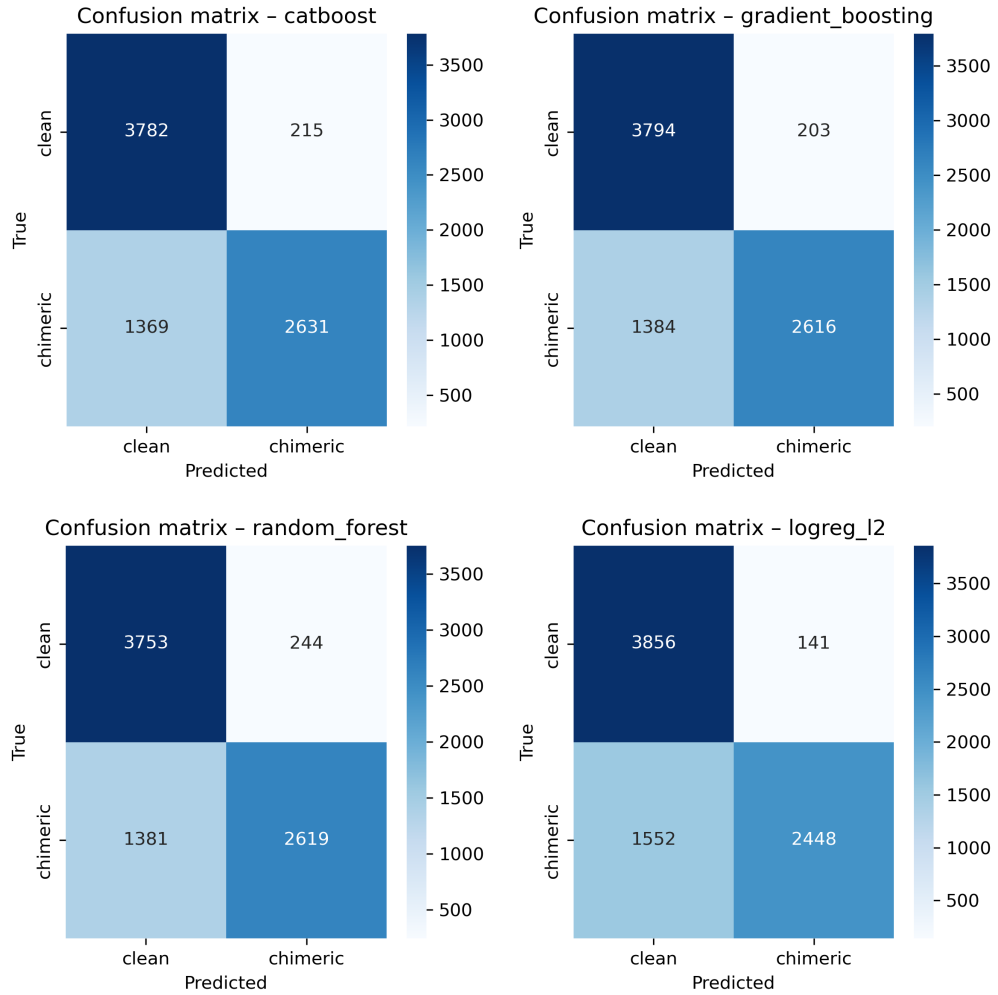


Figure 4.4: Confusion matrices for the four representative models on the held-out test set. All models show more false negatives (chimeric reads called clean) than false positives.

#### 792 4.4.2 ROC and Precision–Recall Curves

793 Receiver operating characteristic (ROC) and precision–recall (PR) curves (Fig-  
 794 ure 4.5) further support the similarity among the top models. The three tree-based  
 795 ensembles (CatBoost, gradient boosting, random forest) achieved ROC–AUC val-  
 796 ues of approximately 0.84 and average precision (AP) around 0.88. Logistic re-

gression performed slightly worse ( $AUC \approx 0.82$ ,  $AP \approx 0.87$ ) but still substantially better than random guessing.

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

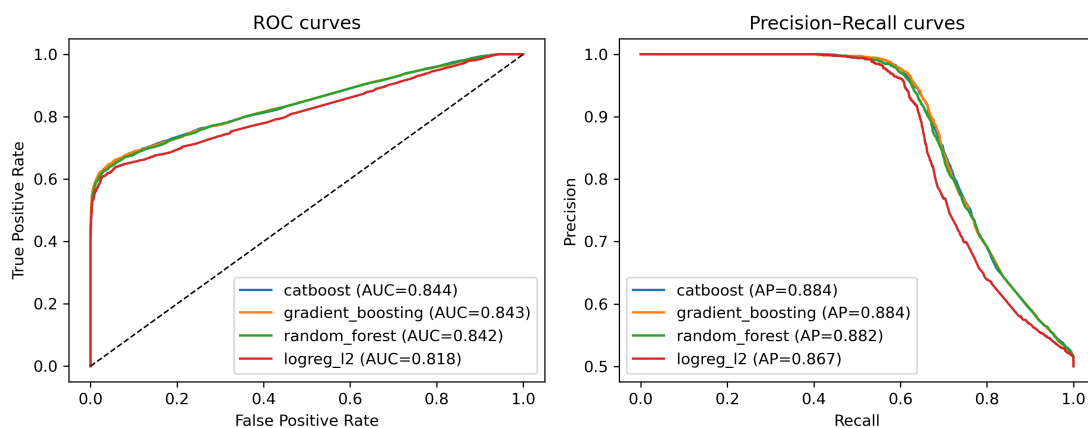


Figure 4.5: ROC (left) and precision–recall (right) curves for the four representative models on the held-out test set. Tree-based ensembles cluster closely, with logistic regression performing slightly but consistently worse.



## 804 4.5 Feature Importance and Biological Interpre- 805 tation

### 806 4.5.1 Permutation Importance of Individual Features

807 To understand how each classifier made predictions, feature importance was quan-  
808 tified using permutation importance. In this approach, the values of a single fea-  
809 ture are randomly shuffled, and the resulting drop in  $F_1$  score ( $\Delta F_1$ ) reflects how  
810 strongly the model depends on that feature. Greater decreases in  $F_1$  indicate  
811 stronger reliance on that feature. This analysis was applied to four representa-  
812 tive models: CatBoost, Gradient Boosting, Random Forest, and  $L_2$ -regularized  
813 Logistic Regression.

814 As shown in Figure 4.6, the total number of clipped bases consistently pro-  
815 vides a strong predictive signal, particularly in Random Forest, Gradient Boosting,  
816 and  $L_2$ -regularized Logistic Regression. CatBoost differs by assigning the highest  
817 importance to k-mer divergence metrics such as `kmer_js_divergence`, which cap-  
818 ture subtle sequence changes resulting from structural variants or PCR-induced  
819 chimeras. Soft-clipping features (`softclip_left` and `softclip_right`) provide  
820 additional context around breakpoints, complementing these primary signals in  
821 all models except Gradient Boosting.  $L_2$ -regularized Logistic Regression relies  
822 more on alignment-based split-read metrics when breakpoints are simple, but it is  
823 less effective at detecting complex rearrangements that introduce novel sequences.

824 Overall, these results indicate that accurate detection of chimeric reads relies  
825 on both alignment-based signals and k-mer compositional information. Explicit

826 microhomology features contribute minimally in this analysis, and combining both  
 827 alignment-based and sequence-level features enhances model sensitivity and speci-  
 828 ficity.

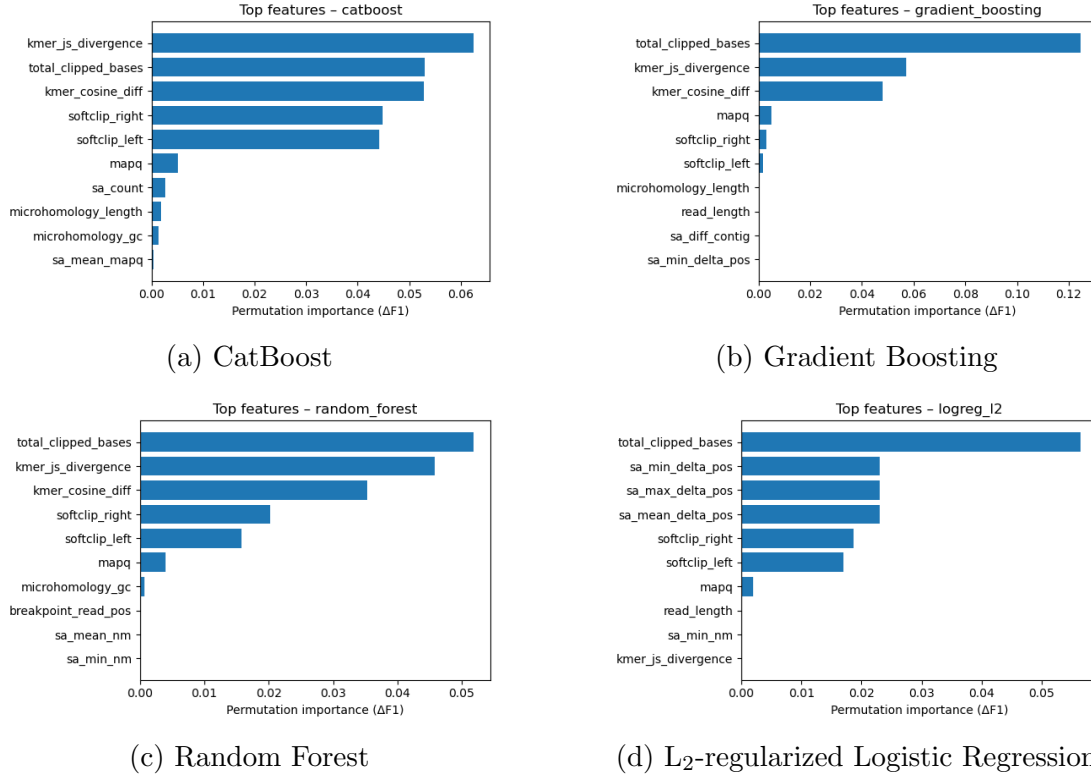


Figure 4.6: Permutation-based feature importance for four representative classifiers. Clipping and k-mer composition features are generally the strongest predictors, whereas microhomology and other alignment metrics contribute minimally.

## 829 4.5.2 Feature Family Importance

830 To evaluate the contribution of broader biological signals, features were  
 831 grouped into five families: SA\_structure (supplementary alignment and seg-  
 832 ment metrics, e.g., `has_sa`, `sa_count`, `sa_min_delta_pos`, `sa_mean_nm`), Clipping  
 833 (`softclip_left`, `softclip_right`, `total_clipped_bases`, `breakpoint_read_pos`),

834 Kmer\_jump (`kmer_cosine_diff`, `kmer_js_divergence`), `Micro_homology`, and  
835 Other (e.g., `mapq`).

836 Aggregated analyses reveal consistent patterns across models. In CatBoost,  
837 the Clipping family has the largest cumulative contribution (0.14), followed  
838 by Kmer\_jump (0.12), with Other features contributing modestly (0.005) and  
839 SA\_structure (0.003) and Micro\_homology (0.003) providing minimal predictive  
840 power. Gradient Boosting shows a similar trend, with Clipping (0.13) domi-  
841 nating, Kmer\_jump (0.11) secondary, and the remaining families contributing  
842 negligibly. Random Forest integrates both Clipping (0.088) and Kmer\_jump  
843 (0.08) effectively, while SA\_structure, Micro\_homology, and Other remain minor  
844 contributors. L<sub>2</sub>-regularized Logistic Regression emphasizes Clipping (0.09)  
845 and SA\_structure (0.07), with Kmer\_jump and Micro\_homology having minimal  
846 impact.

847 Both feature-level and aggregated analyses indicate that detection of chimeric  
848 reads in this dataset relies primarily on alignment disruptions (Clipping) and  
849 k-mer compositional shifts (Kmer\_jump), which often arise from PCR-induced  
850 recombination events, while explicit microhomology features contribute minimally.

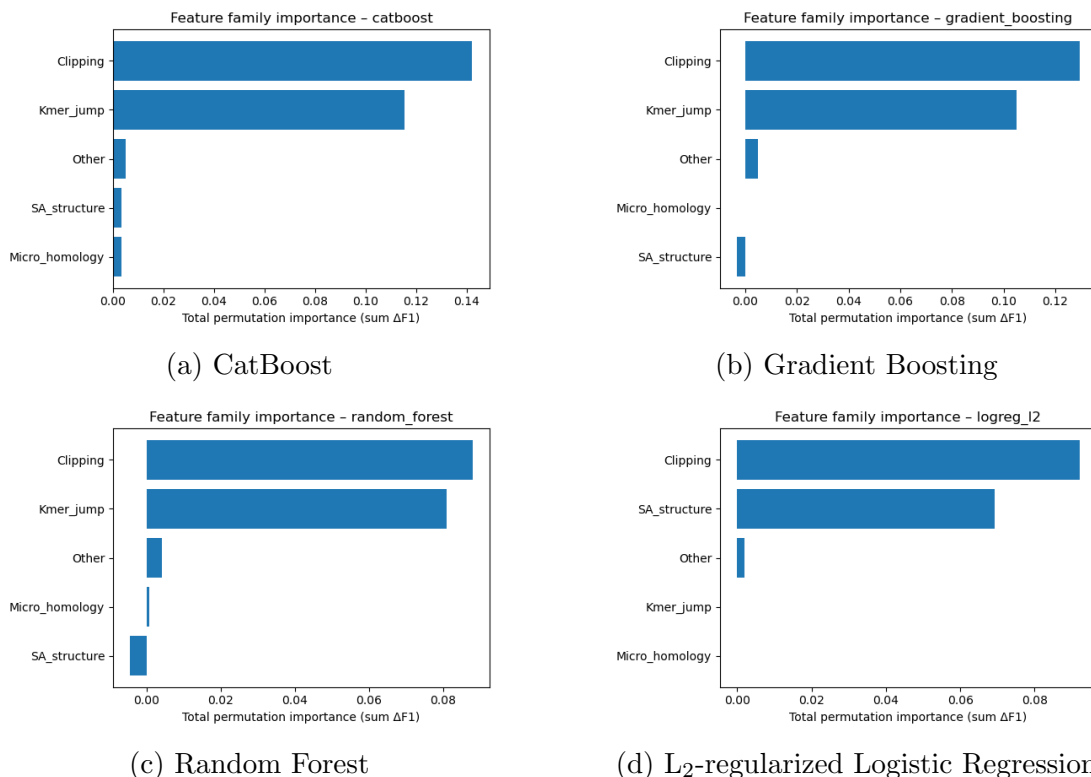


Figure 4.7: Aggregated feature family importance across four models. Clipping and k-mer compositional shifts are consistently the dominant contributors, while SA\_structure, Micro\_homology, and other features contribute minimally.

## 851 4.6 Summary of Findings

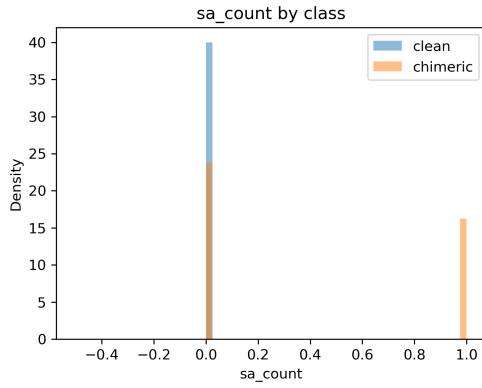
852 After removing trivially discriminative metadata, all models performed substan-  
853 tially better than the dummy baseline, with test F1-scores around 0.76 and ROC-  
854 AUC values near 0.84. Hyperparameter tuning yielded modest improvements,  
855 with boosting methods, particularly CatBoost and gradient boosting, achieving  
856 the highest performance. Confusion matrices and precision-recall curves indicate  
857 that these models prioritise precision for chimeric reads while accepting lower re-  
858 call, which is a conservative strategy appropriate for scenarios where false positives

859 are costly.

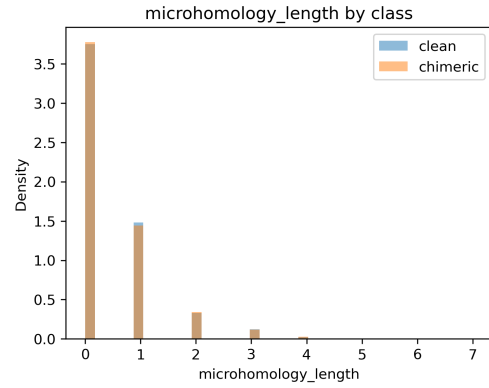
860 Feature importance analyses revealed that alignment disruptions, such as clip-  
861 ping, and abrupt k-mer composition changes accounted for most predictive power.  
862 In contrast, microhomology metrics and supplementary alignment descriptors con-  
863 tributed minimally. These results indicate that features based on read alignment  
864 and k-mer composition are sufficient to train classifiers for detecting mitochon-  
865 drial PCR-induced chimera reads, without needing additional quality-score or  
866 positional information in the conditions tested.

## <sup>867</sup> **Appendix A**

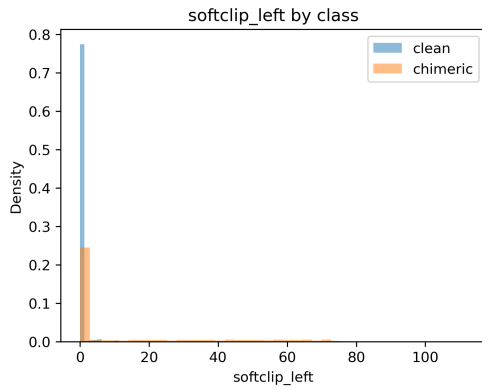
### <sup>868</sup> **Histograms of Key Features**



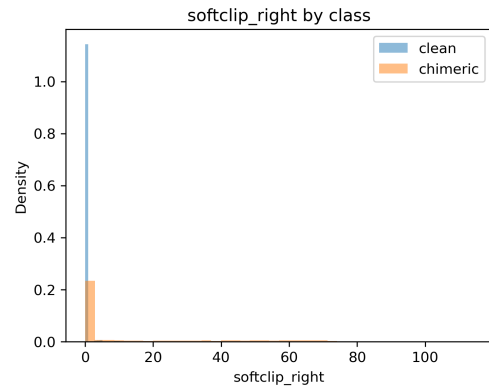
(a) sa\_count



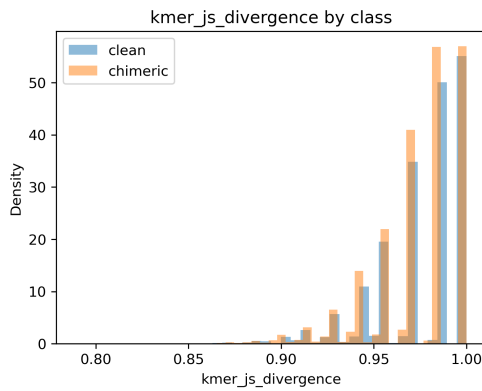
(b) Microhomology length



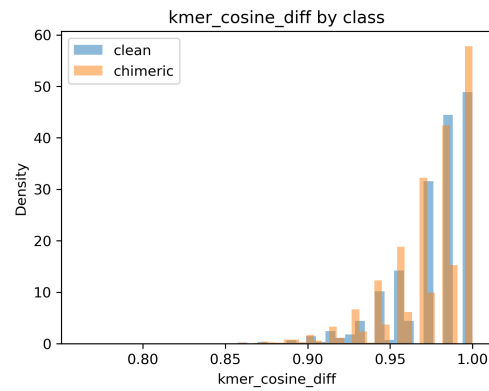
(c) softclip\_left



(d) softclip\_right



(e) k-mer Jensen-Shannon divergence



(f) k-mer cosine difference

Figure A.1: Histogram plots of six key features comparing clean and chimeric reads.

## 869 References

- 870 Anderson, S., Bankier, A., Barrell, B., Bruijn, M., Coulson, A., Drouin, J., ...  
871 Young, I. (1981, 04). Sequence and organization of the human mitochondrial  
872 genome. *Nature*, *290*, 457-465. doi: 10.1038/290457a0
- 873 Arango, G., Garner, E., Pruden, A., Heath, L., Vikesland, P., & Zhang, L. (2018,  
874 02). Deeparg: A deep learning approach for predicting antibiotic resistance  
875 genes from metagenomic data. *Microbiome*, *6*. doi: 10.1186/s40168-018  
876 -0401-z
- 877 Bentley, D. R., Balasubramanian, S., Swerdlow, H. P., Smith, G. P., Milton, J.,  
878 Brown, C. G., ... Smith, A. J. (2008). Accurate whole human genome  
879 sequencing using reversible terminator chemistry. *Nature*, *456*(7218), 53–  
880 59. doi: 10.1038/nature07517
- 881 Boore, J. L. (1999). Animal mitochondrial genomes. *Nucleic Acids Research*,  
882 *27*(8), 1767–1780. doi: 10.1093/nar/27.8.1767
- 883 Cameron, S. L. (2014). Insect mitochondrial genomics: Implications for evolution  
884 and phylogeny. *Annual Review of Entomology*, *59*, 95–117. doi: 10.1146/  
885 annurev-ento-011613-162007
- 886 Dierckxsens, N., Mardulyn, P., & Smits, G. (2017). Novoplasty: de novo assembly  
887 of organelle genomes from whole genome data. *Nucleic Acids Research*,



888 45(4), e18. doi: 10.1093/nar/gkw955

889 Edgar, R. C. (2016). Uchime2: improved chimera prediction for amplicon se-  
 890 quencing. *bioRxiv*. Retrieved from [https://api.semanticscholar.org/](https://api.semanticscholar.org/CorpusID:88955007)  
 891 CorpusID:88955007

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

894 Edgar, R. C., Haas, B. J., Clemente, J. C., Quince, C., & Knight, R. (2011).  
 895 Uchime improves sensitivity and speed of chimera detection. *Bioinformatics*,  
 896 27(16), 2194–2200. doi: 10.1093/bioinformatics/btr381

897 Glenn, T. C. (2011). Field guide to next-generation dna sequencers. *Molecular*  
 898 *Ecology Resources*, 11(5), 759–769. doi: 10.1111/j.1755-0998.2011.03024.x

899 Gonzalez, J. M., Zimmermann, J., & Saiz-Jimenez, C. (2004, 09). Evalu-  
 900 ating putative chimeric sequences from pcr-amplified products. *Bioin-*  
 901 *formatics*, 21(3), 333-337. Retrieved from [https://doi.org/10.1093/](https://doi.org/10.1093/bioinformatics/bti008)  
 902 [bioinformatics/bti008](https://doi.org/10.1093/bioinformatics/bti008) doi: 10.1093/bioinformatics/bti008

903 Gray, M. W. (2012). Mitochondrial evolution. *Cold Spring Harbor perspectives*  
 904 *in biology*, 4. Retrieved from [https://doi.org/10.1101/cshperspect](https://doi.org/10.1101/cshperspect.a011403)  
 905 [.a011403](https://doi.org/10.1101/cshperspect.a011403) doi: 10.1101/cshperspect.a011403

906 Hahn, C., Bachmann, L., & Chevreux, B. (2013). Reconstructing mitochondrial  
 907 genomes directly from genomic next-generation sequencing reads—a baiting  
 908 and iterative mapping approach. *Nucleic Acids Research*, 41(13), e129. doi:  
 909 10.1093/nar/gkt371

910 Jin, J.-J., Yu, W.-B., Yang, J., Song, Y., dePamphilis, C. W., Yi, T.-S., & Li,  
 911 D.-Z. (2020). Getorganelle: a fast and versatile toolkit for accurate de  
 912 novo assembly of organelle genomes. *Genome Biology*, 21(1), 241. doi:  
 913 10.1186/s13059-020-02154-5

- 914 Judo, M. S. B., Wedel, W. R., & Wilson, B. H. (1998). Stimulation and sup-  
915 pression of pcr-mediated recombination. *Nucleic Acids Research*, 26(7),  
916 1819–1825. doi: 10.1093/nar/26.7.1819
- 917 Labrador, K., Agmata, A., Palermo, J. D., Ravago-Gotanco, R., & Pante, M. J.  
918 (2021). Mitochondrial dna reveals genetically structured haplogroups of  
919 bali sardinella (*sardinella lemuru*) in philippine waters. *Regional Studies in*  
920 *Marine Science*, 41, 101588. doi: 10.1016/j.rsma.2020.101588
- 921 Li, H. (2018, 05). Minimap2: pairwise alignment for nucleotide sequences. *Bioin-*  
922 *formatics*, 34(18), 3094–3100. Retrieved from [https://doi.org/10.1093/](https://doi.org/10.1093/bioinformatics/bty191)  
923 [bioinformatics/bty191](https://doi.org/10.1093/bioinformatics/bty191) doi: 10.1093/bioinformatics/bty191
- 924 Liang, Q., Bible, P. W., Liu, Y., Zou, B., & Wei, L. (2020, 02). Deepmi-  
925 crobes: taxonomic classification for metagenomics with deep learning. *NAR*  
926 *Genomics and Bioinformatics*, 2(1), lqaa009. Retrieved from [https://](https://doi.org/10.1093/nargab/lqaa009)  
927 [doi.org/10.1093/nargab/lqaa009](https://doi.org/10.1093/nargab/lqaa009) doi: 10.1093/nargab/lqaa009
- 928 Metzker, M. L. (2010). Sequencing technologies — the next generation. *Nature*  
929 *Reviews Genetics*, 11(1), 31–46. doi: 10.1038/nrg2626
- 930 Mysara, M., Saeys, Y., Leys, N., Raes, J., & Monsieurs, P. (2015). Catch,  
931 an ensemble classifier for chimera detection in 16s rna sequencing stud-  
932 ies. *Applied and Environmental Microbiology*, 81(5), 1573–1584. Retrieved  
933 from <https://journals.asm.org/doi/abs/10.1128/aem.02896-14> doi:  
934 10.1128/AEM.02896-14
- 935 Peccoud, J., Lequime, S., Moltini-Conclois, I., Giraud, I., Lambrechts, L., &  
936 Gilbert, C. (2018, 04). A survey of virus recombination uncovers canon-  
937 ical features of artificial chimeras generated during deep sequencing li-  
938 brary preparation. *G3 Genes—Genomes—Genetics*, 8(4), 1129–1138. Re-  
939 trieved from <https://doi.org/10.1534/g3.117.300468> doi: 10.1534/

940 g3.117.300468

941 Qin, Y., Wu, L., Zhang, Q., Wen, C., Nostrand, J. D. V., Ning, D., ... Zhou, J.  
 942 (2023). Effects of error, chimera, bias, and gc content on the accuracy of  
 943 amplicon sequencing. *mSystems*, 8(6), e01025-23. Retrieved from [https://](https://journals.asm.org/doi/abs/10.1128/msystems.01025-23)  
 944 [journals.asm.org/doi/abs/10.1128/msystems.01025-23](https://journals.asm.org/doi/abs/10.1128/msystems.01025-23) doi: 10.1128/  
 945 msystems.01025-23

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

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

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

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

960 Sedlazeck, F., Rescheneder, P., Smolka, M., Fang, H., Nattestad, M., von Haeseler,  
 961 A., & Schatz, M. (2018, 06). Accurate detection of complex structural  
 962 variations using single-molecule sequencing. *Nature Methods*, 15. doi: 10  
 963 .1038/s41592-018-0001-7

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

966 *Sciences*, 40(11), 701-714. Retrieved from <https://www.sciencedirect>  
 967 [.com/science/article/pii/S0968000415001589](https://www.sciencedirect.com/science/article/pii/S0968000415001589) doi: [https://doi.org/](https://doi.org/10.1016/j.tibs.2015.08.006)  
 968 [10.1016/j.tibs.2015.08.006](https://doi.org/10.1016/j.tibs.2015.08.006)  
 969 Vervier, K., Mahé, P., Tournoud, M., Veyrieras, J.-B., & Vert, J.-P. (2015,  
 970 11). Large-scale machine learning for metagenomics sequence classifica-  
 971 tion. *Bioinformatics*, 32(7), 1023-1032. Retrieved from [https://doi.org/](https://doi.org/10.1093/bioinformatics/btv683)  
 972 [10.1093/bioinformatics/btv683](https://doi.org/10.1093/bioinformatics/btv683) doi: 10.1093/bioinformatics/btv683  
 973 Willette, D., Bognot, E., Mutia, M. T., & Santos, M. (2011). *Biology and ecology*  
 974 *of sardines in the philippines: A review* (Vol. 13; Tech. Rep. No. 1). NFRDI  
 975 Technical Paper Series. Retrieved from [https://nfrdi.da.gov.ph/tpjf/](https://nfrdi.da.gov.ph/tpjf/etc/Willette%20et%20al.%20Sardines%20Review.pdf)  
 976 [etc/Willette%20et%20al.%20Sardines%20Review.pdf](https://nfrdi.da.gov.ph/tpjf/etc/Willette%20et%20al.%20Sardines%20Review.pdf)