

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

20

# 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 . . .	16
41	2.5	Synthesis of Chimera Detection Approaches . . . . .	18
42	<b>3</b>	<b>Research Methodology</b>	<b>21</b>
43	3.1	Research Activities . . . . .	21
44	3.1.1	Data Collection . . . . .	22
45	3.1.2	Feature Extraction Pipeline . . . . .	26
46	3.1.3	Machine Learning Model Development . . . . .	29
47	3.1.4	Model Benchmarking, Hyperparameter Optimization, and	
48		Evaluation . . . . .	30
49	3.1.5	Feature Importance and Interpretation . . . . .	32

50	3.1.6 Validation and Testing . . . . .	33
51	3.1.7 Documentation . . . . .	33
52	3.2 Calendar of Activities . . . . .	34
53	<b>4 Results and Discussion</b>	<b>35</b>
54	4.1 Descriptive Analysis of Features . . . . .	35
55	4.1.1 Univariate Distributions . . . . .	36
56	4.2 Baseline Classification Performance . . . . .	38
57	4.3 Effect of Hyperparameter Tuning . . . . .	39
58	4.4 Detailed Evaluation of Representative Models . . . . .	41
59	4.4.1 Confusion Matrices and Error Patterns . . . . .	42
60	4.4.2 ROC and Precision–Recall Curves . . . . .	43
61	4.5 Feature Importance and Biological Interpretation . . . . .	45
62	4.5.1 Permutation Importance of Individual Features . . . . .	45
63	4.5.2 Feature Family Importance . . . . .	46
64	4.6 Summary of Findings . . . . .	48

# 65 List of Figures

66	3.1	Process Diagram of Special Project . . . . .	22
67	4.1	Kernel density plots of six key features comparing clean and	
68		chimeric reads. . . . .	37
69	4.2	Test F1 of all baseline classifiers, showing that no single model	
70		clearly dominates and several achieve comparable performance. . .	39
71	4.3	Comparison of test F1 (left) and ROC–AUC (right) for baseline and	
72		tuned models. Hyperparameter tuning yields small but consistent	
73		gains, particularly for tree-based ensembles. . . . .	41
74	4.4	Confusion matrices for the four representative models on the held-	
75		out test set. All models show more false negatives (chimeric reads	
76		called clean) than false positives. . . . .	43
77	4.5	ROC (left) and precision–recall (right) curves for the four represen-	
78		tative models on the held-out test set. Tree-based ensembles cluster	
79		closely, with logistic regression performing slightly but consistently	
80		worse. . . . .	44

81	4.6	Permutation-based feature importance for four representative clas-	
82		sifiers. Clipping and k-mer composition features are generally the	
83		strongest predictors, whereas microhomology and other alignment	
84		metrics contribute minimally. . . . .	46
85	4.7	Aggregated feature family importance across four models. Clipping	
86		and k-mer compositional shifts are consistently the dominant con-	
87		tributors, while SA_structure, Micro_homology, and other features	
88		contribute minimally. . . . .	48

# 89 List of Tables

90	2.1	Comparison of Chimera Detection Methods . . . . .	19
91	3.1	Timetable of Activities . . . . .	34
92	4.1	Performance of baseline classifiers on the held-out test set. . . . .	39
93	4.2	Performance of tuned classifiers on the held-out test set. . . . .	40

# 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



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

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

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

## 135 1.2 Problem Statement

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

## 153 1.3 Research Objectives

### 154 1.3.1 General Objective

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

### 159 1.3.2 Specific Objectives

160 Specifically, the study aims to:

- 161 1. construct simulated *Sardinella lemuru* Illumina paired-end datasets contain-  
162 ing both clean and PCR-induced chimeric reads,
- 163 2. extract alignment-based and sequence-based features such as k-mer compo-  
164 sition, junction complexity, and split-alignment counts from both clean and  
165 chimeric reads,
- 166 3. train, validate, and compare supervised machine-learning models for classi-  
167 fying reads as clean or chimeric,
- 168 4. determine feature importance and identify indicators of PCR-induced  
169 chimerism,
- 170 5. integrate the optimized classifier into a modular and interpretable pipeline  
171 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: (1) 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; (2) to align the analysis with relevant *S. lemuru* sequencing projects at PGC Visayas; (3) 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 (4) 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.

## 195 1.5 Significance of the Research

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

## 206 Chapter 2

## 207 Review of Related Literature

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

### 214 2.1 The Mitochondrial Genome

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

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

### 225 **2.1.1 Mitochondrial Genome Assembly**

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

284 sequence correspond to distinct high-abundance sequences.

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

### 290 **2.3.1 UCHIME**

291 UCHIME is one of the most widely used computational tools for detecting chimeric  
292 sequences in amplicon sequencing data, as it serves as a critical quality control  
293 step to prevent the misinterpretation of PCR artifacts as novel biological diversity.  
294 The algorithm operates by searching for a model ( $M$ ) where a query ( $Q$ ) sequence  
295 can be perfectly explained as a combination of two parent sequences, denoted as  
296  $A$  and  $B$  (Edgar et al., 2011).

297 In reference mode, UCHIME divides the query into four chunks and maps  
298 them to a trusted chimeric-free database to identify candidate parents. It then  
299 constructs a three-way alignment to calculate a score based on “votes.” A “Yes”  
300 vote indicates the query aligns with parent  $A$  in one region and parent  $B$  in an-  
301 other, while a “No” vote penalizes the score if the query diverges from the expected  
302 chimeric model. In de novo mode, the algorithm operationalizes the abundance  
303 skew principle described in Section 2.3. Instead of using an external database,  
304 UCHIME dynamically treats the sample’s own high-abundance sequences as a  
305 reference database, testing if lower-abundance sequences can be reconstructed as

306 mosaics of these internal ancestors (Edgar et al., 2011).

307 Despite its high sensitivity, UCHIME has inherent limitations rooted in  
308 sequence divergence and database quality. The algorithm struggles to detect  
309 chimeras formed from parents that are very closely related, specifically when the  
310 sequence divergence between parents is less than roughly 0.8%, as the signal-to-  
311 noise ratio becomes too low to distinguish a crossover event from sequencing error  
312 (Edgar et al., 2011). Furthermore, in reference mode, the accuracy is strictly  
313 bound by the completeness of the database; if true parents are absent, the tool  
314 may fail to identify the chimera or produce false positives. Similarly, the de novo  
315 mode relies on the assumption that parents are present and sufficiently more  
316 abundant in the sample, which may not hold true in unevenly amplified samples  
317 or complex communities.

### 318 **2.3.2 UCHIME2**

319 Building upon the original algorithm, UCHIME2 was developed to address the  
320 nuances of high-resolution amplicon sequencing. A key contribution of the  
321 UCHIME2 study was the critical re-evaluation of chimera detection benchmarks.  
322 In the UCHIME2 paper (Edgar, 2016) and the UCHIME in practice website  
323 (Edgar, n.d), the author has noted that the accuracy results reported in the  
324 original UCHIME paper were “highly over-optimistic” because they relied on  
325 unrealistic benchmark designs where parent sequences were assumed to be 100%  
326 known and present. UCHIME2 introduced more rigorous testing (the CHSIMA  
327 benchmark), revealing that “fake models,” where a valid biological sequence  
328 perfectly mimics a chimera of two other valid sequences, are far more common

329 than previously assumed. This discovery suggests that error-free detection is  
330 impossible in principle (Edgar, 2016). Another notable improvement is the in-  
331 troduction of multiple application-specific modes that allow users to tailor the  
332 algorithm’s performance to the characteristics of their datasets. The following  
333 parameter presets: denoised, balanced, sensitive, specific, and high-confidence,  
334 enable researchers to optimize the balance between sensitivity and specificity  
335 according to the goals of their analysis.

336 However despite these advancements, the practical application of UCHIME2  
337 requires caution. The author explicitly advises against using UCHIME2 as  
338 a stand-alone tool in standard OTU clustering or denoising pipelines. Using  
339 UCHIME2 as an independent filtering step in these workflows is discouraged, as  
340 it often results in significantly higher error rates, increasing both false positives  
341 (discarding valid sequences) and false negatives (retaining chimeras) (Edgar,  
342 2016).

### 343 **2.3.3 CATch**

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

349 To address these inconsistencies, Mysara, Saeys, Leys, Raes, and Monsieurs  
350 (2015) developed the Classifier for Amplicon Tool Chimeras (CATCh), which rep-

resents the first ensemble machine learning system designed for chimera detection in 16S rRNA amplicon sequencing. Rather than depending on a single detection strategy, CATCh integrates the outputs of several established tools, including UCHIME, ChimeraSlayer, DECIPHER, Pintail, and Perseus. The individual scores and binary decisions generated by these tools are used as input features for a supervised learning model. The algorithm employs a Support Vector Machine (SVM) with a Pearson VII Universal Kernel (PUK) to determine optimal weightings among the input features and to assign each sequence a probability of being chimeric.

Benchmarking in both reference-based and de novo modes demonstrated significant performance improvements. CATCh achieved sensitivities of approximately 85 percent in reference-based mode and 92 percent in de novo mode, with corresponding specificities of approximately 96 percent and 95 percent. These results indicate that CATCh detected 7 to 12 percent more chimeras than any individual algorithm while maintaining high precision.

### 2.3.4 ChimPipe

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

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

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

## 384 **2.4 Machine Learning Approaches for Chimera** 385 **and Sequence Quality Detection**

386 Traditional chimera detection tools rely primarily on heuristic or alignment-based  
387 rules. Recent advances in machine learning (ML) have demonstrated that models  
388 trained on sequence-derived features can effectively capture compositional and  
389 structural patterns in biological sequences. Although most existing ML systems  
390 such as those used for antibiotic resistance prediction, taxonomic classification,  
391 or viral identification are not specifically designed for chimera detection, they  
392 highlight how data-driven models can outperform similarity-based heuristics by  
393 learning intrinsic sequence signatures. In principle, ML frameworks can integrate  
394 indicators such as k-mer frequencies, GC-content variation and split-alignment

395 metrics to identify subtle anomalies that may indicate a chimeric origin (Arango  
396 et al., 2018; Liang, Bible, Liu, Zou, & Wei, 2020; Ren et al., 2020).

### 397 **2.4.1 Feature-Based Representations of Genomic Se-** 398 **quences**

399 In genomic analysis, feature extraction converts DNA sequences into numerical  
400 representations suitable for ML algorithms. A common approach is k-mer fre-  
401 quency analysis, where normalized k-mer counts form the feature vector (Vervier,  
402 Mahé, Tournoud, Veyrieras, & Vert, 2015). These features effectively capture lo-  
403 cal compositional patterns that often differ between authentic and chimeric reads.  
404 In particular, deviations in k-mer profiles between adjacent read segments can  
405 serve as a compositional signature of template-switching events. Additional de-  
406 scriptors such as GC content and sequence entropy can further distinguish se-  
407 quence types; in metagenomic classification and virus detection, k-mer-based fea-  
408 tures have shown strong performance and robustness to noise (Ren et al., 2020;  
409 Vervier et al., 2015). For chimera detection specifically, abrupt shifts in GC or k-  
410 mer composition along a read can indicate junctions between parental fragments.  
411 Windowed feature extraction enables models to capture these discontinuities that  
412 rule-based algorithms may overlook.

413 Machine learning models can also leverage alignment-derived features such as  
414 the frequency of split alignments, variation in mapping quality, and local cover-  
415 age irregularities. Split reads and discordant read pairs are classical indicators  
416 of genomic junctions and have been formalized in probabilistic frameworks for  
417 structural-variant discovery that integrate multiple evidence types (Layer, Hall, &

418 Quinlan, 2014). Similarly, long-read tools such as Sniffles employ split-alignment  
419 and coverage anomalies to accurately localize breakpoints (Sedlazeck et al., 2018).  
420 Modern aligners such as Minimap2 (Li, 2018) output supplementary (SA tags) and  
421 secondary alignments as well as chaining and alignment-score statistics that can  
422 be summarized into quantitative predictors for machine-learning models. These  
423 alignment-signal features are particularly relevant to PCR-induced mitochondrial  
424 chimeras, where template-switching events produce reads partially matching dis-  
425 tinct regions of the same or related genomes. Integrating such cues within a  
426 supervised-learning framework enables artifact detection even in datasets lacking  
427 complete or perfectly assembled references.

428 A further biologically grounded descriptor is the length of microhomology at  
429 putative junctions. Microhomology refers to short, shared sequences, often in the  
430 range of a few to tens of base pairs that are near breakpoints where template-  
431 switching events typically happen. Studies of double strand break repair and  
432 structural variation have demonstrated that the length of microhomology corre-  
433 lates with the likelihood of microhomology-mediated end joining (MMEJ) or fork-  
434 stalled template-switching pathways (Sfeir & Symington, 2015). In the context of  
435 PCR-induced chimeras, template switching during amplification often leaves short  
436 identical sequences at the junction of two concatenated fragments. Quantifying  
437 the longest exact suffix-prefix overlap at each candidate breakpoint thus provides  
438 a mechanistic signature of chimerism and complements both compositional (k-  
439 mer) and alignment (SA count) features.



## 440   **2.5   Synthesis of Chimera Detection Approaches**

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

Table 2.1: Comparison of Chimera Detection Methods

Methods	Approach	Limitations
<b>Reference-based Chimera Detection</b>	Compares query sequences against curated, non-chimeric reference databases; identifies mosaic sequences by evaluating similarity to known templates.	Depends heavily on completeness and quality of reference databases; often fails when novel taxa or missing parent sequences are present; reduced accuracy for low-divergence chimeras.
<b>De novo Chimera Detection</b>	Identifies chimeras using only internal dataset relationships; relies on abundance patterns and compositional similarity; reconstructs sequences as mosaics of high-abundance parents.	Assumes true sequences are more abundant—fails when amplification bias distorts abundance; struggles with evenly abundant parental sequences; can misclassify highly similar true variants.
<b>UCHIME</b>	Alignment-based chimera detection; segments query sequence, identifies parent candidates, performs 3-way alignment, and computes chimera scores; supports both reference-based and de novo modes.	Accuracy inflated in original benchmarks; suffers under incomplete databases; poor performance on low-divergence chimeras; sensitive to sequencing errors; misclassifies when parents are missing.
<b>UCHIME2</b>	Improved initial UCHIME benchmarking; offers multiple sensitivity/specificity modes; more robust with incomplete references; higher sensitivity.	Cannot achieve perfect accuracy due to “perfect fake models”; genuine variants may be indistinguishable from artificial recombinants; theoretical detection limit remains.
<b>CATCH</b>	First ML ensemble tool for 16S chimera detection; integrates outputs of UCHIME, ChimeraSlayer, DECIPHER, Pintail, Perseus via SVM classifier; significantly improves sensitivity and specificity.	Depends on performance of underlying tools; ML model limited to features they output; ensemble can still misclassify in datasets with extreme novelty or low coverage.
<b>ChimPipe</b>	Pipeline for detecting fusion genes and transcript-derived chimeras in RNA-seq; uses discordant paired-end reads and split-alignments; predicts isoforms and breakpoint coordinates.	Designed for RNA-seq, not amplicons; needs high-quality genome and annotation; computationally heavier; limited to organisms with reference genomes.

444 Across existing studies, no single approach reliably detects all forms of chimeric  
445 sequences, particularly those generated by PCR-induced template switching in  
446 mitochondrial genomes. Reference-based tools perform poorly when parental se-  
447 quences are absent; de novo methods rely strongly on abundance assumptions;  
448 alignment-based systems show reduced sensitivity to low-divergence chimeras; and  
449 ensemble methods inherit the limitations of their component algorithms. RNA-  
450 seq-oriented pipelines likewise do not generalize well to organelle data. Although  
451 machine learning approaches offer promising feature-based detection, they are  
452 rarely applied to mitochondrial genomes and are not trained specifically on PCR-  
453 induced organelle chimeras. These limitations indicate a clear research gap: the  
454 need for a specialized, feature-driven classifier tailored to mitochondrial PCR-  
455 induced chimeras that integrates k-mer composition, split-alignment signals, and  
456 micro-homology features to achieve more accurate detection than current heuristic  
457 or alignment-based tools.

## 458 Chapter 3

# 459 Research Methodology

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

### 465 3.1 Research Activities

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

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

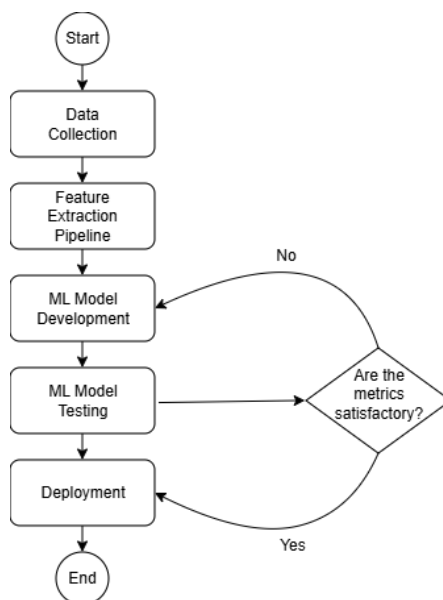


Figure 3.1: Process Diagram of Special Project

### 478 3.1.1 Data Collection

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

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

484 proximately one (1) week.

## 485 Data Preprocessing

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

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

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

497 The command parameters are as follows:

- 498 • `-1` and `-2`: read lengths of 150 base pairs for each paired-end read.
- 499 • `-r`, `-R`, `-X`: mutation rate, fraction of indels, and indel extension probability,  
500 all set to a default value of 0.
- 501 • `-e`: base error rate, set to 0.001 to simulate realistic sequencing errors.
- 502 • `-N`: number of read pairs, set to 10,000.

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

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

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

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

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

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

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

```
528 samtools view -bS clean.sam -o clean.bam
529 samtools view -bS chimeric.sam -o chimeric.bam
530
531 samtools sort clean.bam -o clean.sorted.bam
532 samtools index clean.sorted.bam
533
534 samtools sort chimeric.bam -o chimeric.sorted.bam
535 samtools index chimeric.sorted.bam
```

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

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

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



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

### 549 **3.1.2 Feature Extraction Pipeline**

550 A feature extraction pipeline will be developed and implemented to extract the  
551 necessary analytical features. This pipeline will function as a reproducible and  
552 modular workflow that accepts FASTQ and BAM/SAM file inputs, processes them  
553 using tools such as **samtools** and **jellyfish** (Version 2.3.1), and produces tabular  
554 feature matrices (TSV) for downstream machine learning. To ensure correctness  
555 and adherence to best practices, bioinformatics experts at the PGC Visayas will  
556 be consulted to validate the pipeline design, feature extraction logic, and overall  
557 data integrity. This stage of the study is scheduled to begin in the first week of  
558 January 2026 and conclude by the last week of February 2026, with an estimated  
559 total duration of approximately two (2) months.

560 The feature extraction pipeline focuses on three principal features from the  
561 simulated and aligned sequencing data: (1) supplementary alignment flag (SA  
562 count), (2) k-mer composition difference between read segments, and (3) micro-  
563 homology length at potential junctions. Each of these features captures a distinct  
564 biological or computational signature associated with PCR-induced chimeras.

### 565 **Supplementary Alignment Flag**

566 Supplementary alignment information will be assessed using the mapped and  
567 sorted BAM files (**clean.sorted.bam** and **chimeric.sorted.bam**) generated

568 from the data preprocessing stage. Alignment summaries will be checked using  
569 `samtools flagstat` to obtain preliminary quality-control statistics, including  
570 counts of primary, secondary, and supplementary (SA) alignments.

571 Both BAM files will be converted to SAM format for detailed inspection of  
572 reads in each file:

573 `samtools view -h clean.sorted.bam -o clean.sorted.sam`

574 `samtools view -h chimeric.sorted.bam -o chimeric.sorted.sam`

575 The SAM output will be checked for reads containing the `SA:Z` flag, as it  
576 denotes supplementary alignments. Reads exhibiting these or substantial soft-  
577 clipped regions will be considered strong candidates for chimeric artifacts. A  
578 custom Python script would be created to extract the alignment-derived features  
579 and relevant metadata including mapping quality, SAM flag information, CIGAR-  
580 based clipping, and alignment coordinates. These extracted attributes would then  
581 be organized and compiled into a TSV (`.tsv`) file.

## 582 **K-mer Composition Difference**

583 Chimeric reads often comprise fragments from distinct genomic regions, resulting  
584 in a compositional discontinuity between segments. Comparing k-mer frequency  
585 profiles between the left and right halves of a read allows detection of such abrupt  
586 compositional shifts, independent of alignment information. This will be obtained  
587 using Jellyfish, a fast k-mer counting software. For each read, the sequence will  
588 be divided into two segments, either at the midpoint or at empirically determined  
589 breakpoints inferred from supplementary alignment data, to generate left and right

sequence segments. Jellyfish will then compute k-mer frequency profiles (with  $k =$   
5 or 6) for each segment. The resulting k-mer frequency vectors will be normalized  
and compared using distance metrics such as cosine similarity or Jensen–Shannon  
divergence to quantify compositional disparity between the two halves of the same  
read. The resulting difference scores will be stored in a structured TSV file.

## Microhomology

The microhomology length was computed as part of the feature extraction  
pipeline. For each aligned read in the BAM files, the script first inferred a  
breakpoint using the function `infer_breakpoint`, which represents a junction  
between two segments. Breakpoints were determined primarily from soft-clipping  
patterns, where part of a read does not align to the reference and may indicate  
a junction. If no soft clips were present, SA tags were used to identify potential  
alignment discontinuities.

Once a breakpoint was established, the script scanned a  $\pm 40$  base pair window  
surrounding the breakpoint and used the function `longest_suffix_prefix_overlap`  
to identify the longest exact suffix-prefix overlap between the left and right read  
segments. This overlap, which represents consecutive bases shared at the junction,  
was recorded as the microhomology length. The 40-base pair window was  
chosen to ensure that short shared sequences at or near the breakpoint were  
captured, without including distant sequences that are unrelated. 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

613 and 1. Short microhomologies, typically 3-20 base pairs in length, are recognized  
614 signatures of PCR-induced template switching (Peccoud et al., 2018).

615 To provide additional sequence context around breakpoints, k-mer profiles  
616 were computed. K-mers are short sequences of consecutive nucleotides, and k-  
617 mer profiles quantify the frequency of each sequence. A k-mer length of 6 was  
618 used to capture patterns within the same 40-base pair window surrounding each  
619 breakpoint. High-frequency k-mers, such as repeated bases (e.g., "AAAAAA"),  
620 may indicate repetitive or unusual sequences at the junction. These profiles com-  
621 plement microhomology measurements and help identify junctions that are po-  
622 tentially chimeric.

### 623 **3.1.3 Machine Learning Model Development**

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

633 The labelled dataset was randomly partitioned into training (80%) and test  
634 (20%) subsets using stratified sampling to preserve the 1:1 ratio of clean to

chimeric reads. Model development and evaluation were implemented in Python (Version 3.11) using the `scikit-learn`, `xgboost`, `lightgbm`, and `catboost` libraries. A broad panel of classification algorithms was then benchmarked on the training data to obtain a fair comparison of different model families under identical feature conditions. The panel included: a trivial dummy classifier, L2-regularized logistic regression, a calibrated linear support vector machine (SVM),  $k$ -nearest neighbours, Gaussian Naïve Bayes, decision-tree ensembles (Random Forest, Extremely Randomized Trees, and Bagging with decision trees), gradient boosting methods (Gradient Boosting, XGBoost, LightGBM, and CatBoost), and a shallow multilayer perceptron (MLP).

For each model, five-fold stratified cross-validation was performed on the training set. In every fold, four-fifths of the data were used for fitting and the remaining one-fifth for validation. Mean cross-validation accuracy, precision, recall, F1-score for the chimeric class, and area under the receiver operating characteristic curve (ROC-AUC) were computed to summarize performance and rank candidate methods. 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

658 forward: L2-regularized logistic regression, calibrated linear SVM, Random For-  
659 est, ExtraTrees, Gradient Boosting, XGBoost, LightGBM, CatBoost, Bagging  
660 with decision trees, and a shallow MLP. This subset spans both linear and non-  
661 linear decision boundaries, but emphasizes ensemble and boosting methods, which  
662 showed superior F1 and ROC-AUC in the initial benchmark.

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

673 For each model family, the hyperparameter configuration with the highest  
674 mean cross-validation F1-score was selected as the best-tuned estimator. These  
675 tuned models were then refitted on the full training set and evaluated once on the  
676 held-out test set to obtain unbiased estimates of performance. Test-set metrics in-  
677 cluded accuracy, precision, recall, F1-score for the chimeric class, and ROC-AUC.  
678 Confusion matrices and ROC curves were generated for the top-performing mod-  
679 els to characterise common error modes, such as false negatives (missed chimeric  
680 reads) and false positives (clean reads incorrectly labelled as chimeric). The final  
681 model or small set of models for downstream interpretation was chosen based on  
682 a combination of test-set F1-score, ROC-AUC, and practical considerations such

683 as model complexity and ease of deployment within a feature extraction pipeline.

### 684 **3.1.5 Feature Importance and Interpretation**

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

695 For interpretability, individual features were grouped into four conceptual  
696 families: (i) supplementary alignment and alignment-structure features (e.g., SA  
697 count, spacing between alignment segments, strand consistency), (ii) CIGAR-  
698 derived soft-clipping features (e.g., left and right soft-clipped length, total clipped  
699 bases), (iii) k-mer composition discontinuity features (e.g., cosine distance and  
700 Jensen–Shannon divergence between k-mer profiles of read segments), and (iv) mi-  
701 crohomology descriptors (e.g., microhomology length and local GC content around  
702 putative breakpoints). Aggregating permutation importance scores within each  
703 family allowed assessment of which biological signatures contributed most strongly  
704 to the classifier’s performance. This analysis provided a basis for interpreting the  
705 trained models in terms of known mechanisms of PCR-induced template switching

706 and for identifying which alignment- and sequence-derived cues are most informa-  
707 tive for distinguishing chimeric from clean mitochondrial reads.

### 708 **3.1.6 Validation and Testing**

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

717 Comparative evaluation was performed across all candidate algorithms, in-  
718 cluding a trivial dummy classifier, L2-regularized logistic regression, a calibrated  
719 linear SVM, k-nearest neighbours, Gaussian Naïve Bayes, decision-tree ensembles,  
720 gradient boosting methods, and a shallow MLP. This evaluation determined which  
721 models demonstrated the highest predictive performance and computational effi-  
722 ciency under identical data conditions. Their metrics were compared to identify  
723 which algorithms were most suitable for further refinement.

### 724 **3.1.7 Documentation**

725 Comprehensive documentation was maintained throughout the study to ensure  
726 transparency and reproducibility. All stages of the research, including data gath-



727 ering, preprocessing, feature extraction, model training, and validation, were sys-  
 728 tematically recorded in a `.README` file in the GitHub repository. For each ana-  
 729 lytical step, the corresponding parameters, software versions, and command line  
 730 scripts were documented to enable exact replication of results.

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

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

## 738 3.2 Calendar of Activities

739 Table 3.1 presents the project timeline in the form of a Gantt chart, where each  
 740 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).

### 753 4.1.1 Univariate Distributions

754 The kernel density plots in Figures 4.1a–4.1f collectively show that alignment-  
755 based features provide the strongest separation between clean and chimeric reads.  
756 The distribution of `sa_count` (Figure 4.1a) is distinctly bimodal, with clean reads  
757 concentrated near zero and chimeric reads peaking around one, reflecting the  
758 frequent presence of supplementary alignments in chimeras. A similar pattern of  
759 clear separation is observed in `softclip_left` and `softclip_right` (Figures 4.1c  
760 and 4.1d), where clean reads cluster tightly at zero while chimeric reads display  
761 broad, long-tailed distributions, consistent with extensive soft clipping when  
762 a read spans multiple genomic locations. In contrast, `microhomology_length`  
763 (Figure 4.1b) shows substantial overlap between classes, with both distribu-  
764 tions sharply concentrated near zero and exhibiting smaller secondary peaks  
765 at short integer lengths, indicating limited discriminative value under the sim-  
766 ulated conditions. Finally, the k-mer-based features `kmer_js_divergence` and  
767 `kmer_cosine_diff` (Figures 4.1e and 4.1f) exhibit highly overlapping, multimodal  
768 distributions with both classes peaking near 1.0; although chimeric reads appear  
769 slightly less concentrated at the highest similarity values, the separation is weak  
770 overall.

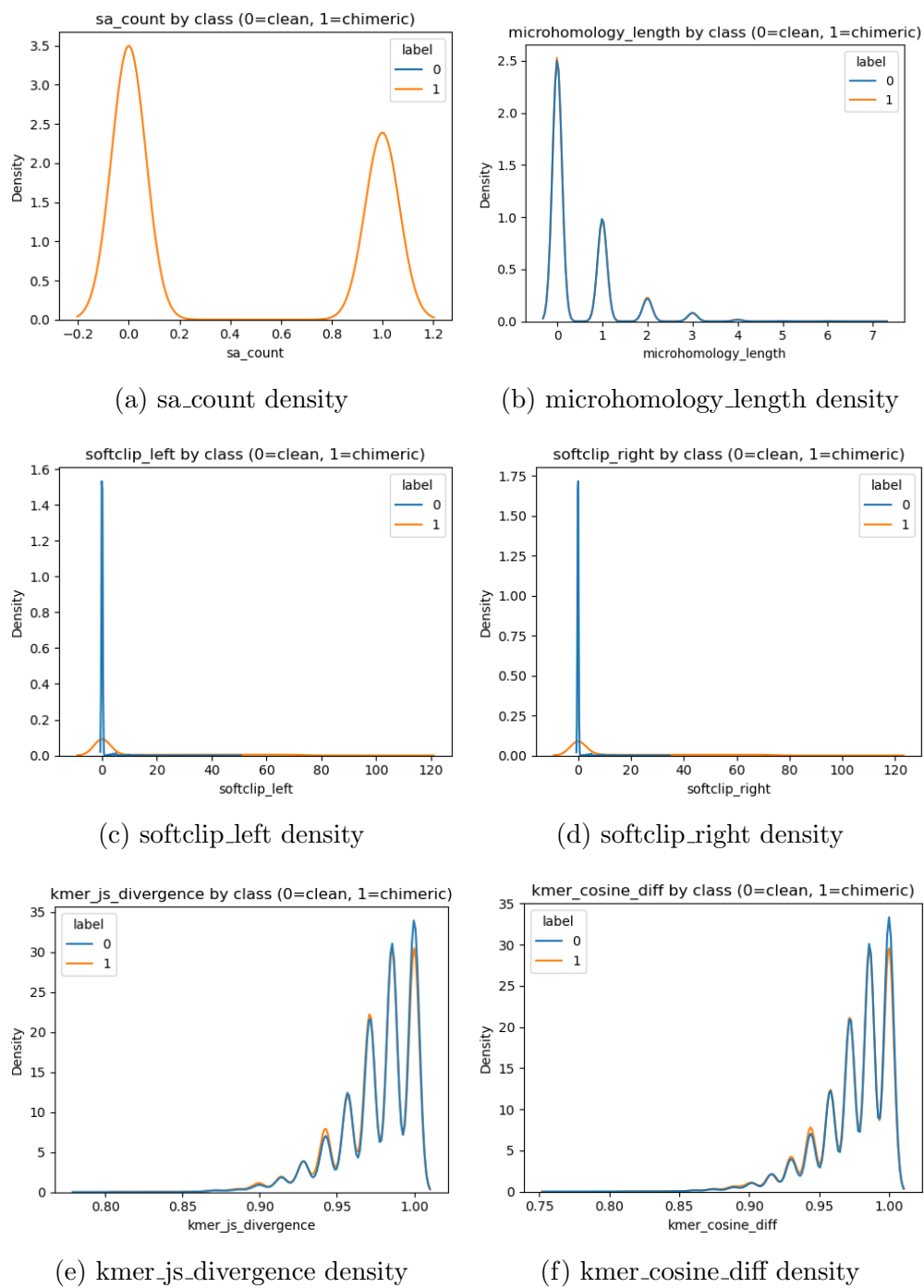


Figure 4.1: Kernel density plots of six key features comparing clean and chimeric reads.

## 771 4.2 Baseline Classification Performance

772 Table 4.1 summarises the performance of eleven classifiers trained on the engi-  
773 neered feature set using five-fold cross-validation and evaluated on the held-out  
774 test set. All models were optimised using default hyperparameters, without ded-  
775 icated tuning.

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

780 Across other models, test F1-scores clustered in a narrow band between ap-  
781 proximately 0.74 and 0.77 and ROC-AUC values between 0.82 and 0.84. Gradi-  
782 ent boosting, CatBoost, LightGBM, XGBoost, bagging trees, random forest, and  
783 multilayer perceptron (MLP) all produced very similar scores, with CatBoost and  
784 gradient boosting slightly ahead (test F1  $\approx$  0.77, ROC-AUC  $\approx$  0.84). Linear  
785 models (logistic regression and calibrated linear SVM) performed only marginally  
786 worse (test F1  $\approx$  0.74), while Gaussian Naive Bayes lagged behind with substan-  
787 tially 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

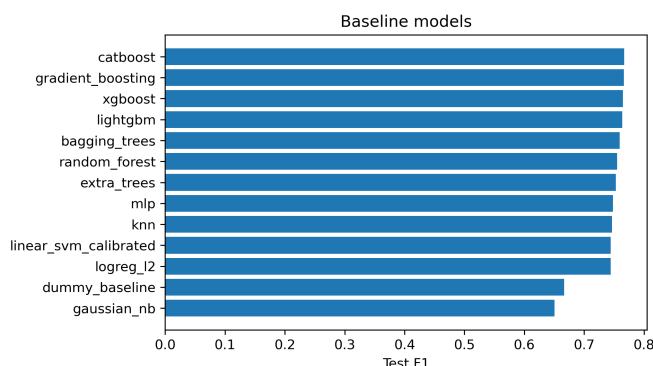


Figure 4.2: Test F1 of all baseline classifiers, showing that no single model clearly dominates and several achieve comparable performance.

## 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 essen-

793 tially unchanged or slightly worse.

794 CatBoost, gradient boosting, LightGBM, XGBoost, random forest, bagging  
795 trees, and MLP all experienced small increases in test F1 (typically  $\Delta F1 \approx 0.002$ –  
796 0.009) and ROC–AUC (up to  $\Delta AUC \approx 0.008$ ). After tuning, CatBoost remained  
797 the best performer with test accuracy 0.802, precision 0.924, recall 0.658, F1-score  
798 0.769, and ROC–AUC 0.844. Gradient boosting achieved almost identical perfor-  
799 mance (F1 0.767, AUC 0.843). Random forest and bagging trees also improved  
800 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

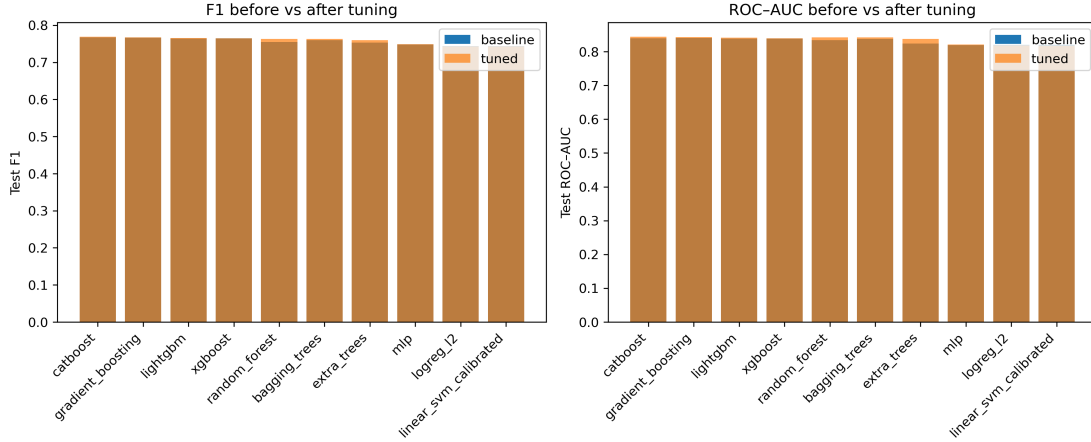


Figure 4.3: Comparison of test F1 (left) and ROC-AUC (right) for baseline and tuned models. Hyperparameter tuning yields small but consistent gains, particularly for tree-based ensembles.

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

## 804 4.4 Detailed Evaluation of Representative Mod- 805 els

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



#### 812 4.4.1 Confusion Matrices and Error Patterns

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

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

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

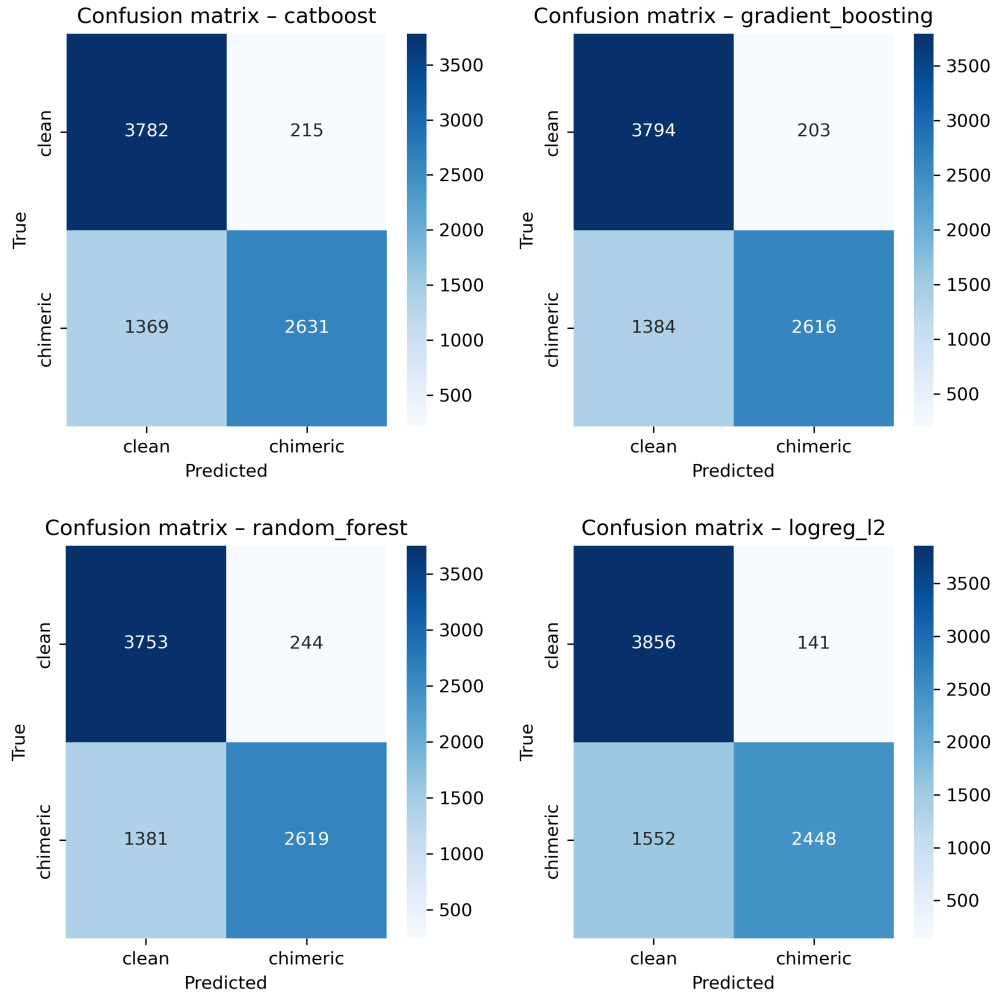


Figure 4.4: Confusion matrices for the four representative models on the held-out test set. All models show more false negatives (chimeric reads called clean) than false positives.

#### 4.4.2 ROC and Precision–Recall Curves

Receiver operating characteristic (ROC) and precision–recall (PR) curves (Figure 4.5) further support the similarity among the top models. The three tree-based ensembles (CatBoost, gradient boosting, random forest) achieved ROC–AUC values 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.

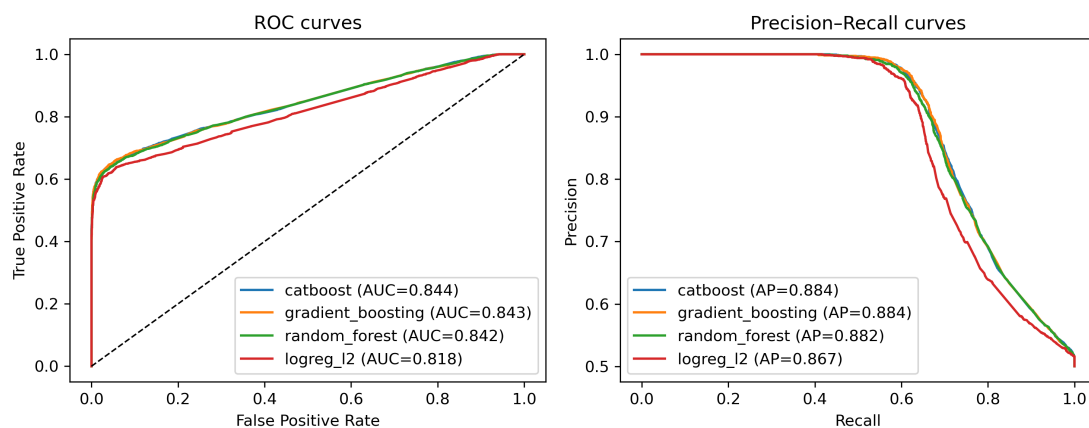


Figure 4.5: ROC (left) and precision–recall (right) curves for the four representative models on the held-out test set. Tree-based ensembles cluster closely, with logistic regression performing slightly but consistently worse.

## 842 4.5 Feature Importance and Biological Interpre- 843 tion

### 844 4.5.1 Permutation Importance of Individual Features

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

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

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

864 microhomology features contribute minimally in this analysis, and combining both  
 865 alignment-based and sequence-level features enhances model sensitivity and speci-  
 866 ficity.

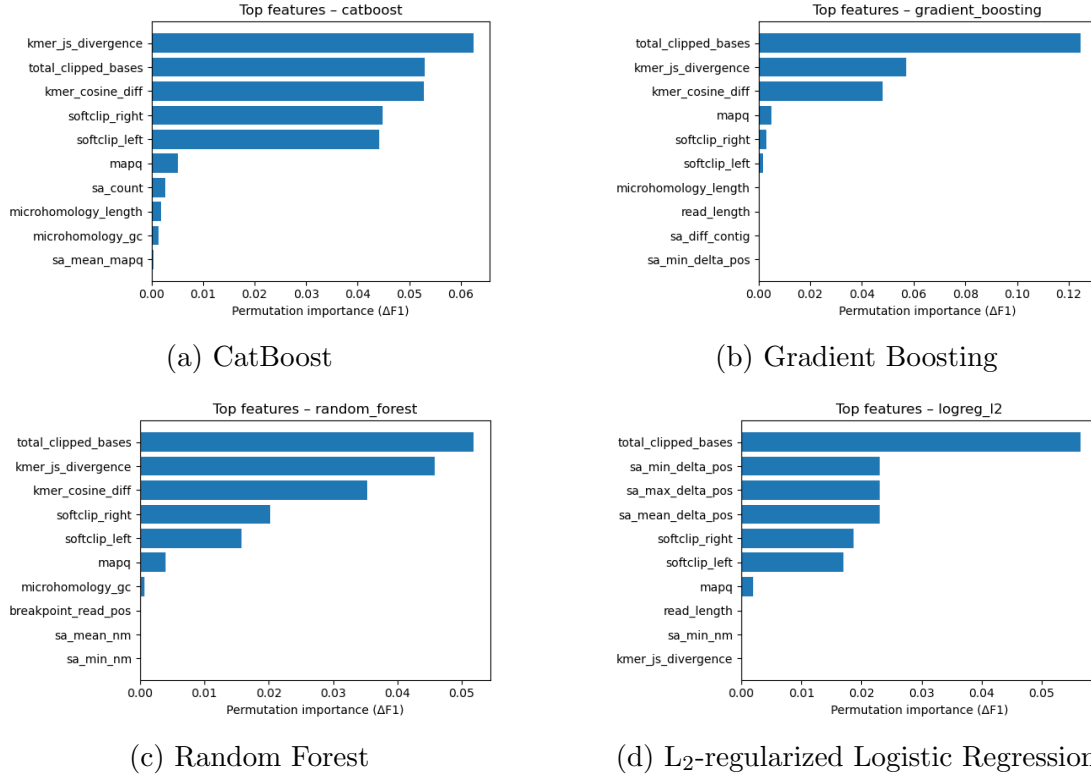


Figure 4.6: Permutation-based feature importance for four representative classifiers. Clipping and k-mer composition features are generally the strongest predictors, whereas microhomology and other alignment metrics contribute minimally.

## 867 4.5.2 Feature Family Importance

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

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

874     Aggregated analyses reveal consistent patterns across models. In CatBoost,  
875 the Clipping family has the largest cumulative contribution (0.14), followed  
876 by Kmer\_jump (0.12), with Other features contributing modestly (0.005) and  
877 SA\_structure (0.003) and Micro\_homology (0.003) providing minimal predictive  
878 power. Gradient Boosting shows a similar trend, with Clipping (0.13) domi-  
879 nating, Kmer\_jump (0.11) secondary, and the remaining families contributing  
880 negligibly. Random Forest integrates both Clipping (0.088) and Kmer\_jump  
881 (0.08) effectively, while SA\_structure, Micro\_homology, and Other remain minor  
882 contributors. L<sub>2</sub>-regularized Logistic Regression emphasizes Clipping (0.09)  
883 and SA\_structure (0.07), with Kmer\_jump and Micro\_homology having minimal  
884 impact.

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

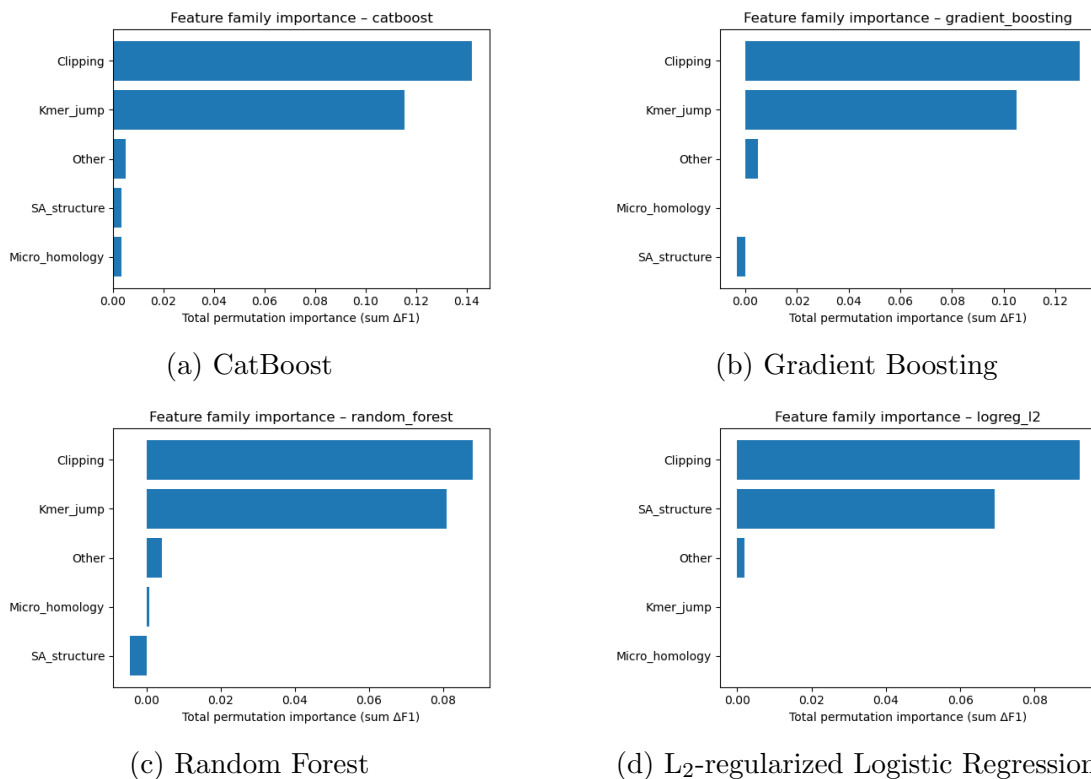


Figure 4.7: Aggregated feature family importance across four models. Clipping and k-mer compositional shifts are consistently the dominant contributors, while SA\_structure, Micro\_homology, and other features contribute minimally.

## 4.6 Summary of Findings

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

897 are costly.

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



## 905 References

- 906 Anderson, S., Bankier, A., Barrell, B., Bruijn, M., Coulson, A., Drouin, J., ...  
907 Young, I. (1981, 04). Sequence and organization of the human mitochondrial  
908 genome. *Nature*, *290*, 457-465. doi: 10.1038/290457a0
- 909 Arango, G., Garner, E., Pruden, A., Heath, L., Vikesland, P., & Zhang, L. (2018,  
910 02). Deeparg: A deep learning approach for predicting antibiotic resistance  
911 genes from metagenomic data. *Microbiome*, *6*. doi: 10.1186/s40168-018  
912 -0401-z
- 913 Bentley, D. R., Balasubramanian, S., Swerdlow, H. P., Smith, G. P., Milton, J.,  
914 Brown, C. G., ... Smith, A. J. (2008). Accurate whole human genome  
915 sequencing using reversible terminator chemistry. *Nature*, *456*(7218), 53–  
916 59. doi: 10.1038/nature07517
- 917 Boore, J. L. (1999). Animal mitochondrial genomes. *Nucleic Acids Research*,  
918 *27*(8), 1767–1780. doi: 10.1093/nar/27.8.1767
- 919 Cameron, S. L. (2014). Insect mitochondrial genomics: Implications for evolution  
920 and phylogeny. *Annual Review of Entomology*, *59*, 95–117. doi: 10.1146/  
921 annurev-ento-011613-162007
- 922 Dierckxsens, N., Mardulyn, P., & Smits, G. (2017). Novoplasty: de novo assembly  
923 of organelle genomes from whole genome data. *Nucleic Acids Research*,

924 45(4), e18. doi: 10.1093/nar/gkw955

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

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

927 CorpusID:88955007

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

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

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

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

932 27(16), 2194–2200. doi: 10.1093/bioinformatics/btr381

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

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

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

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

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

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

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

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

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

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

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

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

945 10.1093/nar/gkt371

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

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

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

949 10.1186/s13059-020-02154-5

950 Judo, M. S. B., Wedel, W. R., & Wilson, B. H. (1998). Stimulation and sup-  
951 pression of pcr-mediated recombination. *Nucleic Acids Research*, *26*(7),  
952 1819–1825. doi: 10.1093/nar/26.7.1819

953 Labrador, K., Agmata, A., Palermo, J. D., Ravago-Gotanco, R., & Pante, M. J.  
954 (2021). Mitochondrial dna reveals genetically structured haplogroups of  
955 bali sardinella (*sardinella lemuru*) in philippine waters. *Regional Studies in*  
956 *Marine Science*, *41*, 101588. doi: 10.1016/j.rsma.2020.101588

957 Layer, R., Hall, I., & Quinlan, A. (2014, 10). Lumpy: A probabilistic framework  
958 for structural variant discovery. *Genome Biology*, *15*. doi: 10.1186/gb-2014-  
959 -15-6-r84

960 Li, H. (2018, 05). Minimap2: pairwise alignment for nucleotide sequences. *Bioin-*  
961 *formatics*, *34*(18), 3094-3100. Retrieved from [https://doi.org/10.1093/](https://doi.org/10.1093/bioinformatics/bty191)  
962 [bioinformatics/bty191](https://doi.org/10.1093/bioinformatics/bty191) doi: 10.1093/bioinformatics/bty191

963 Liang, Q., Bible, P. W., Liu, Y., Zou, B., & Wei, L. (2020, 02). Deepmi-  
964 crobes: taxonomic classification for metagenomics with deep learning. *NAR*  
965 *Genomics and Bioinformatics*, *2*(1), lqaa009. Retrieved from [https://](https://doi.org/10.1093/nargab/lqaa009)  
966 [doi.org/10.1093/nargab/lqaa009](https://doi.org/10.1093/nargab/lqaa009) doi: 10.1093/nargab/lqaa009

967 Metzker, M. L. (2010). Sequencing technologies — the next generation. *Nature*  
968 *Reviews Genetics*, *11*(1), 31–46. doi: 10.1038/nrg2626

969 Mysara, M., Saeys, Y., Leys, N., Raes, J., & Monsieurs, P. (2015). Catch,  
970 an ensemble classifier for chimera detection in 16s rRNA sequencing stud-  
971 ies. *Applied and Environmental Microbiology*, *81*(5), 1573-1584. Retrieved  
972 from <https://journals.asm.org/doi/abs/10.1128/aem.02896-14> doi:  
973 [10.1128/AEM.02896-14](https://journals.asm.org/doi/abs/10.1128/aem.02896-14)

974 Peccoud, J., Lequime, S., Moltini-Conclois, I., Giraud, I., Lambrechts, L., &  
975 Gilbert, C. (2018, 04). A survey of virus recombination uncovers canon-

ical features of artificial chimeras generated during deep sequencing library preparation. *G3 Genes—Genomes—Genetics*, 8(4), 1129-1138. Retrieved 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

1002 .1038/s41592-018-0001-7

1003 Sfeir, A., & Symington, L. S. (2015). Microhomology-mediated end joining: A  
 1004 back-up survival mechanism or dedicated pathway? *Trends in Biochemical*  
 1005 *Sciences*, 40(11), 701-714. Retrieved from [https://www.sciencedirect](https://www.sciencedirect.com/science/article/pii/S0968000415001589)  
 1006 [.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)  
 1007 [10.1016/j.tibs.2015.08.006](https://doi.org/10.1016/j.tibs.2015.08.006)

1008 Vervier, K., Mahé, P., Tournoud, M., Veyrieras, J.-B., & Vert, J.-P. (2015,  
 1009 11). Large-scale machine learning for metagenomics sequence classifica-  
 1010 tion. *Bioinformatics*, 32(7), 1023-1032. Retrieved from [https://doi.org/](https://doi.org/10.1093/bioinformatics/btv683)  
 1011 [10.1093/bioinformatics/btv683](https://doi.org/10.1093/bioinformatics/btv683) doi: 10.1093/bioinformatics/btv683

1012 Willette, D., Bognot, E., Mutia, M. T., & Santos, M. (2011). *Biology and ecology*  
 1013 *of sardines in the philippines: A review* (Vol. 13; Tech. Rep. No. 1). NFRDI  
 1014 Technical Paper Series. Retrieved from [https://nfrdi.da.gov.ph/tpjf/](https://nfrdi.da.gov.ph/tpjf/etc/Willette%20et%20al.%20Sardines%20Review.pdf)  
 1015 [etc/Willette%20et%20al.%20Sardines%20Review.pdf](https://nfrdi.da.gov.ph/tpjf/etc/Willette%20et%20al.%20Sardines%20Review.pdf)