

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

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

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

13 by

14 Duranne Duran
15 Yvonne Lin
16 Daniella Pailden

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

19 December 14, 2025

Abstract

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

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

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

Contents

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

55	2.2	PCR Amplification and Chimera Formation	9
56	2.3	Existing Traditional Approaches for Chimera Detection	10
57	2.3.1	UCHIME	11
58	2.3.2	UCHIME2	12
59	2.3.3	CATch	13
60	2.3.4	ChimPipe	14
61	2.4	Machine Learning Approaches for Chimera and Sequence Quality	
62		Detection	15
63	2.4.1	Feature-Based Representations of Genomic Sequences . . .	15
64	2.5	Synthesis of Chimera Detection Approaches	16
65	3	Research Methodology	19
66	3.1	Research Activities	19
67	3.1.1	Data Collection	20
68	3.1.2	Feature Extraction Pipeline	23
69	3.1.3	Machine Learning Model Development	26
70	3.1.4	Model Benchmarking, Hyperparameter Optimization, and	
71		Evaluation	27
72	3.1.5	Feature Importance and Interpretation	28

73	3.1.6 Validation and Testing	29
74	3.1.7 Documentation	30
75	3.2 Calendar of Activities	31
76	4 Results and Discussion	32
77	4.1 Descriptive Analysis of Features	32
78	4.1.1 Exploratory Data Analysis	33
79	4.2 Baseline Classification Performance	34
80	4.3 Effect of Hyperparameter Tuning	36
81	4.4 Detailed Evaluation of Representative Models	37
82	4.4.1 Confusion Matrices and Error Patterns	38
83	4.4.2 ROC and Precision–Recall Curves	39
84	4.5 Feature Importance	40
85	4.5.1 Permutation Importance of Individual Features	40
86	4.5.2 Feature Family Importance	42
87	4.6 Summary of Findings	44
88	A Exploratory Data Analysis	46
89	A.1 Histograms of Key Features	46

90 List of Figures

91	3.1	Process Diagram of Special Project	20
92	4.1	Feature correlation heatmap showing relationships among alignment-	
93		derived and sequence-derived features.	34
94	4.2	Test F1 of all baseline classifiers, showing that no single model	
95		clearly dominates and several achieve comparable performance. . .	35
96	4.3	Comparison of test F1 (left) and ROC–AUC (right) for baseline	
97		and tuned models.	37
98	4.4	Confusion matrices for the four representative models on the held-	
99		out test set.	39
100	4.5	ROC (left) and precision–recall (right) curves for the four represen-	
101		tative models on the held-out test set.	40
102	4.6	Permutation-based feature importance for four representative clas-	
103		sifiers.	42
104	4.7	Aggregated feature family importance across four models.	44

105	A.1 Histogram plots of six key features comparing clean and chimeric	
106	reads.	47

107 List of Tables

<small>108</small>	2.1 Comparison of Chimera Detection Approaches and Tools	17
<small>109</small>	3.1 Timetable of Activities	31
<small>110</small>	4.1 Performance of baseline classifiers on the held-out test set.	35
<small>111</small>	4.2 Performance of tuned classifiers on the held-out test set.	36

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 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 PCR-induced chimeric reads using both alignment-based and sequence-derived statistical features. The tool aims to provide bioinformatics laboratories, particularly the Philippine Genome Center Visayas (PGC Visayas), with an efficient

151 solution for improving mitochondrial genome reconstruction.

152 1.2 Problem Statement

153 Chimeric reads can distort assembly graphs and cause misassemblies, with par-
154 ticularly severe effects in mitochondrial genomes (Boore, 1999; Cameron, 2014).
155 Existing assembly pipelines such as GetOrganelle, MITObim, and NOVOPlasty
156 assume that sequencing reads are free of such artifacts (Dierckxsens et al., 2017;
157 Hahn et al., 2013; Jin et al., 2020). At PGC Visayas, several mitochondrial as-
158 semblies have failed or yielded incomplete contigs despite sufficient coverage, sug-
159 gesting that undetected chimeric reads compromise assembly reliability. Mean-
160 while, existing chimera detection tools such as UCHIME and VSEARCH were
161 developed primarily for amplicon-based community analysis and rely heavily on
162 reference or taxonomic comparisons (Edgar, Haas, Clemente, Quince, & Knight,
163 2011; Rognes, Flouri, Nichols, Quince, & Mahé, 2016). These approaches are un-
164 suitable for single-species organellar data, where complete reference genomes are
165 often unavailable.

166 1.3 Research Objectives

167 1.3.1 General Objective

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

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

172 **1.3.2 Specific Objectives**

173 Specifically, the study aims to:

- 174 1. construct simulated *Sardinella lemuru* Illumina paired-end datasets contain-
175 ing both clean and PCR-induced chimeric reads,
- 176 2. extract alignment-based and sequence-based features such as k-mer compo-
177 sition, junction complexity, and split-alignment counts from both clean and
178 chimeric reads,
- 179 3. train, validate, and compare supervised machine learning models for classi-
180 fying reads as clean or chimeric,
- 181 4. determine feature importance and identify indicators of PCR-induced
182 chimerism,
- 183 5. integrate the optimized classifier into a modular and interpretable pipeline
184 deployable on standard computing environments at PGC Visayas.

185 **1.4 Scope and Limitations of the Research**

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

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

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

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

214 patterns.

215 1.5 Significance of the Research

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

226 Chapter 2

227 Review of Related Literature

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

234 2.1 The Mitochondrial Genome

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

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

245 **2.1.1 Mitochondrial Genome Assembly**

246 Mitochondrial genome assembly refers to the reconstruction of the complete mito-
247 chondrial DNA (mtDNA) sequence from raw or fragmented sequencing reads. It is
248 conducted to obtain high-quality, continuous representations of the mitochondrial
249 genome that can be used for a wide range of analyses, including species identi-
250 fication, phylogenetic reconstruction, evolutionary studies, and investigations of
251 mitochondrial diseases. Because mtDNA evolves rapidly, its assembled sequence
252 provides valuable insights into population structure, lineage divergence, and adap-
253 tive evolution across taxa (Boore, 1999). Compared to nuclear genome assembly,
254 assembling the mitochondrial genome is often considered more straightforward but
255 still encounters technical challenges such as the formation of chimeric reads. Com-
256 monly used tools for mitogenome assembly such as GetOrganelle and MITObim
257 operate under the assumption of organelle genome circularity, and are vulnerable
258 when chimeric reads disrupt this circular structure, resulting in assembly errors
259 (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

304 sequence correspond to distinct high-abundance sequences.

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

310 **2.3.1 UCHIME**

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

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

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

332 **2.3.2 UCHIME2**

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

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

349 2.3.3 CATCh

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

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

366 Benchmarking in both reference-based and de novo modes demonstrated signif-
367 icant performance improvements. CATCh achieved sensitivities of approximately
368 85 percent in reference-based mode and 92 percent in de novo mode, with corre-
369 sponding specificities of approximately 96 percent and 95 percent. These results
370 indicate that CATCh detected 7 to 12 percent more chimeras than any individual
371 algorithm while maintaining high precision.

372 2.3.4 ChimPipe

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

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

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

390 2.4 Machine Learning Approaches for Chimera 391 and Sequence Quality Detection

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

403 2.4.1 Feature-Based Representations of Genomic Se- 404 quences

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

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

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

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

429 **2.5 Synthesis of Chimera Detection Approaches**

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

Table 2.1: Comparison of Chimera Detection Approaches and Tools

Method / Tool	Core Approach	Key Limitations
Reference-based Detection	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.
De novo Detection	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.
UCHIME	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.
UCHIME2	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.
CATCh	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.
ChimPipe	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.

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

445 Chapter 3

446 Research Methodology

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

452 3.1 Research Activities

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

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

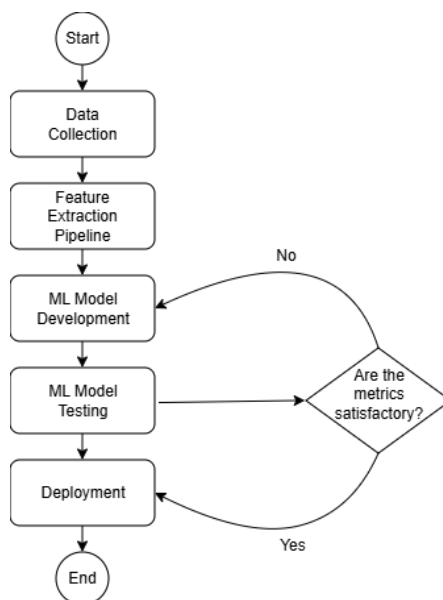


Figure 3.1: Process Diagram of Special Project

465 3.1.1 Data Collection

466 The mitochondrial genome reference sequence of *S. lemuru* was obtained from the
467 NCBI database (accession number NC_039553.1) in FASTA format and was used
468 to generate simulated reads.

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

471 proximately one (1) week.

472 Data Preprocessing

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

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

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

484 Chimeric sequences were then generated from the same reference FASTA
485 file using a separate Python script. Two non-adjacent segments were randomly
486 selected such that their midpoint distances fell within specified minimum and
487 maximum thresholds. The script attempts to retain microhomology to mimic
488 PCR-induced template switching. The resulting chimeras were written to
489 `chimera_reference.fasta` and was processed with `wgsim` to simulate 10,000
490 paired-end fragments, generating 20,000 chimeric reads (10,000 forward reads in
491 `chimeric1.fastq` and 10,000 reverse reads in `chimeric2.fastq`) using the same
492 command format above.

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

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

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

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

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

502 The resulting clean and chimeric SAM files contain the alignment positions of
503 each read relative to the original reference genome. These files were then converted
504 to BAM format, sorted, and indexed using `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 The total number of simulated reads was expected to be 40,000. The final col-
514 lection of reads contained 19,984 clean reads and 20,000 chimeric reads (39,984 en-
515 tries in total), providing a roughly balanced distribution between the two classes.
516 After alignment with `minimap2`, only 19,984 clean reads remained because un-
517 mapped reads were not included in the BAM file. Some sequences failed to align
518 due to the 5% error rate defined during `wgsim` simulation, which produced mis-
519 matches that caused certain reads to fall below the aligner’s matching threshold.

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

523 **3.1.2 Feature Extraction Pipeline**

524 This stage directly follows the previous alignment phase, utilizing the resulting
525 BAM files (specifically `chimeric.sorted.bam` and `clean.sorted.bam`). A cus-
526 tom Python script was created to efficiently process each primary-mapped read
527 to extract the necessary set of features, which are then compiled into a structured
528 feature matrix in TSV format. The pipeline’s core functionality relies on libraries,
529 namely `Pysam` (Version 0.22) for the robust parsing of BAM structures and `NumPy`
530 (Version 1.26) for array operations and computations. To ensure correctness and
531 adherence to best practices, bioinformatics experts at the PGC Visayas will be
532 consulted to validate the pipeline design, feature extraction logic, and overall data
533 integrity.

534 This stage of the study was scheduled to begin in the last week of November

535 2025 and conclude by the first week of December 2025, with an estimated total
536 duration of approximately two (2) weeks.

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

540 **Supplementary Alignment Flag**

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

548 After parsing, the function `sa_feature_stats` was applied to establish the fun-
549 damental split indicators, `has_sa` and `sa_count`. Along with these initial counts,
550 the function aggregated the metrics related to the structure and reliability of the
551 split alignments.

552 **K-mer Composition Difference**

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

556 The script implemented this by inferring a likely junction breakpoint using the
557 function `infer_breakpoints`, prioritizing the boundaries defined by soft-clipping
558 operations. If no clipping was present, the midpoint of the alignment or the read
559 length was utilized as a fallback. The read sequence was then divided into left and
560 right segments at this inferred breakpoint, and k-mer frequency profiles ($k = 6$)
561 were generated for both halves, ignoring any k-mers containing ambiguous 'N'
562 bases. The resulting k-mer frequency vectors are then normalized and compared
563 using the functions `cosine_difference` and `js_divergence`.

564 **Microhomology**

565 The process of extracting the microhomology feature started by utilizing the func-
566 tion `infer_breakpoints` similar to the k-mer workflow. Once a breakpoint was es-
567 tablished, the script scanned a ± 40 base pair window surrounding the breakpoint
568 and used the function `longest_suffix_prefix_overlap` to identify the longest
569 exact suffix-prefix overlap between the left and right read segments. This overlap,
570 which represents consecutive bases shared at the junction, was recorded as the
571 `microhomology_length` in the dataset. The 40-base pair window was chosen to
572 ensure that short shared sequences at or near the breakpoint were captured, with-
573 out including distant sequences that are unrelated. Additionally, the GC content
574 of the overlapping sequence was calculated using the function `gc_content`, which
575 counts guanine (G) and cytosine (C) bases within the detected microhomology
576 and divides by the total length, yielding a proportion between 0 and 1, and was
577 stored under the `microhomology_gc` attribute. Microhomology was quantified
578 using a (3–20) bp window, consistent with values reported in prior research on
579 PCR-induced chimeras. Moreover, a k-mer length of 6 was used to capture pat-

580 terns within the 40-bp window surrounding each breakpoint. This length is long
581 enough to detect informative sequence shifts at junctions.

582 3.1.3 Machine Learning Model Development

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

592 The labelled dataset was randomly partitioned into training (80%) and test
593 (20%) subsets using stratified sampling to preserve the 1:1 ratio of clean to
594 chimeric reads. Model development and evaluation were implemented in Python
595 (Version 3.11) using the `scikit-learn`, `xgboost`, `lightgbm`, and `catboost` li-
596 braries. A broad panel of classification algorithms was then benchmarked on the
597 training data to obtain a fair comparison of different model families under identical
598 feature conditions. The panel included: a trivial dummy classifier, L_2 -regularized
599 logistic regression, a calibrated linear support vector machine (SVM), k -nearest
600 neighbours, Gaussian Naïve Bayes, decision-tree ensembles (Random Forest, Ex-
601 tremely Randomized Trees, and Bagging with decision trees), gradient boosting
602 methods (Gradient Boosting, XGBoost, LightGBM, and CatBoost), and a shallow

603 multilayer perceptron (MLP).

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

612 **3.1.4 Model Benchmarking, Hyperparameter Optimiza-** 613 **tion, and Evaluation**

614 Model selection and refinement proceeded in two stages. First, the cross-validation
615 results from the broad panel were used to identify a subset of competitive mod-
616 els for more detailed optimization. Specifically, ten model families were carried
617 forward: L_2 -regularized logistic regression, calibrated linear SVM, Random For-
618 est, ExtraTrees, Gradient Boosting, XGBoost, LightGBM, CatBoost, Bagging
619 with decision trees, and a shallow MLP. This subset spans both linear and non-
620 linear decision boundaries, but emphasizes ensemble and boosting methods, which
621 showed superior F1 and ROC–AUC in the initial benchmark.

622 Second, hyperparameter optimization was conducted for each of the ten se-
623 lected models using randomized search with five-fold stratified cross-validation
624 (`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, and ROC-AUC.

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

648 to reducing impurity. Second, model-agnostic permutation importance was com-
649 puted on the test set by repeatedly permuting each feature column while keeping
650 all others fixed and measuring the resulting decrease in F1-score. Features whose
651 permutation led to a larger performance drop were interpreted as more influential
652 for chimera detection.

653 For interpretability, individual features were grouped into four conceptual
654 families: (i) supplementary alignment and alignment-structure features (e.g., SA
655 count, spacing between alignment segments, strand consistency), (ii) soft-clipping
656 features (e.g., left and right soft-clipped length, total clipped bases), (iii) k-mer
657 composition discontinuity features (e.g., cosine distance and Jensen–Shannon di-
658 vergence between k-mer profiles of read segments), and (iv) microhomology de-
659 scriptors (e.g., microhomology length and local GC content around putative break-
660 points). This analysis provided a basis for interpreting the trained models in
661 terms of known mechanisms of PCR-induced template switching and for identi-
662 fying which alignment-based and sequence-derived cues are most informative for
663 distinguishing chimeric from clean mitochondrial reads.

664 **3.1.6 Validation and Testing**

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

671 Comparative evaluation was performed across all candidate algorithms to de-
672 termine which models demonstrated the highest predictive performance and com-
673 putational efficiency under identical data conditions. Their metrics were compared
674 to identify which algorithms were most suitable for further refinement.

675 **3.1.7 Documentation**

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

682 The repository structure followed standard research data management prac-
683 tices, with clear directories for datasets and scripts. Computational environments
684 were standardized using Conda, with an environment file (`environment.yml`)
685 specifying dependencies and package versions to maintain consistency across sys-
686 tems.

687 For manuscript preparation and supplementary materials, Overleaf (L^AT_EX)
688 was used to produce publication-quality formatting and consistent referencing.

3.2 Calendar of Activities

Table 3.1 presents the project timeline in the form of a Gantt chart, where each 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. The behaviour of the main features is first described, followed by a comparison of baseline classifiers, an assessment of the effect of hyperparameter tuning, and an analysis of feature importance in terms of individual variables and 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).

704 4.1.1 Exploratory Data Analysis

705 An exploratory data analysis (EDA) was conducted on the extracted feature ma-
706 trix to characterize general patterns in the data and gain preliminary insight into
707 which variables might meaningfully contribute to classification. Histograms of key
708 features indicated that alignment-based variables showed clear class separation as
709 chimeric reads have higher frequencies of split alignments and broader long-tailed
710 distribution on soft-clipped regions (`softclip_left` and `softclip_right`). In
711 contrast, sequence-based variables such a microhomology length and k-mer di-
712 vergence displayed substantial overlap between classes, suggesting more limited
713 discriminative value. The complete set of histograms is provided in Appendix A.

714 As shown in Figure 4.1, the feature correlation heatmap shows that alignment-
715 derived features form a strongly correlated cluster, whereas sequence-derived fea-
716 tures show weak correlations with both the alignment-based features and with
717 one another. This heterogeneity indicates that no single feature family captures
718 all relevant signal sources.

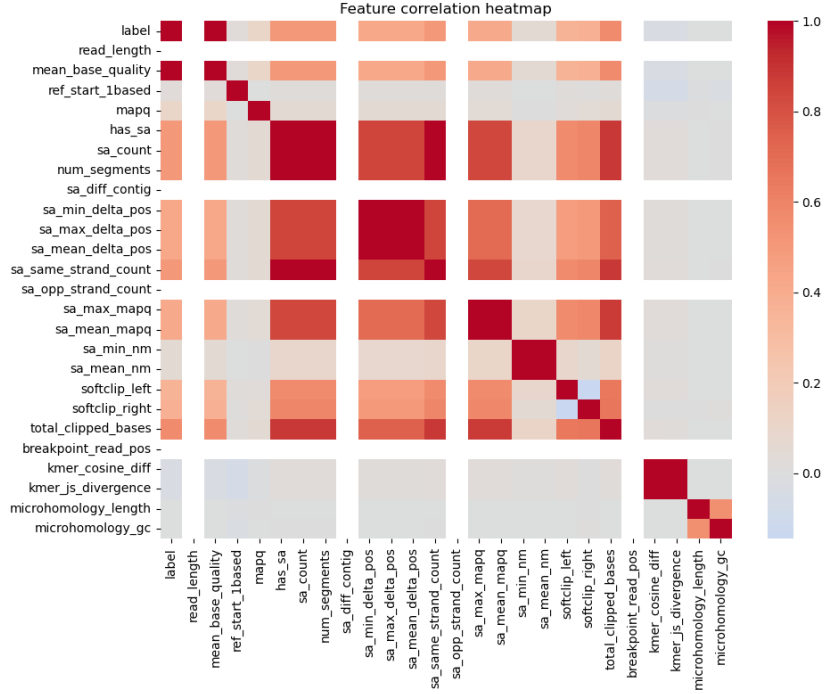


Figure 4.1: Feature correlation heatmap showing relationships among alignment-derived and sequence-derived features.

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.

728 Across other models, test F1-scores clustered in a narrow band between ap-
729 proximately 0.74 and 0.77 and ROC-AUC values between 0.82 and 0.84. Gradi-
730 ent boosting, CatBoost, LightGBM, XGBoost, bagging trees, random forest, and
731 multilayer perceptron (MLP) all produced very similar scores, with CatBoost and
732 gradient boosting slightly ahead (test F1 ≈ 0.77 , ROC-AUC ≈ 0.84). Linear
733 models (logistic regression and calibrated linear SVM) performed only marginally
734 worse (test F1 ≈ 0.74), while Gaussian Naive Bayes lagged behind with substan-
735 tially lower F1 (≈ 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

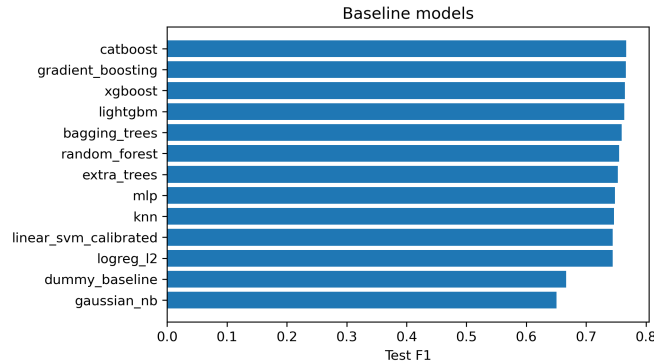


Figure 4.2: Test F1 of all baseline classifiers, showing that no single model clearly dominates and several achieve comparable performance.

4.3 Effect of Hyperparameter Tuning

To assess whether performance could be improved further, ten model families underwent randomised hyperparameter search. 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 $\Delta\text{F1} \approx 0.002$ – 0.009) and ROC–AUC (up to $\Delta\text{AUC} \approx 0.008$). After tuning, CatBoost remained the best performer with test accuracy 0.80, precision 0.92, recall 0.66, F1-score 0.77, and ROC–AUC 0.84. Gradient boosting achieved almost identical performance (F1 0.77, AUC 0.84). Random forest and bagging trees also improved to F1 scores around 0.76 with $\text{AUC} \approx 0.84$.

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

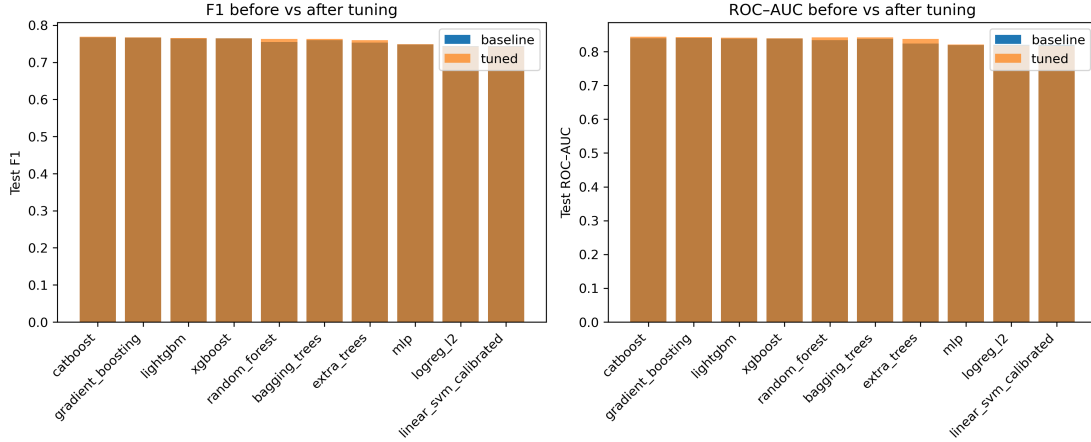


Figure 4.3: Comparison of test F1 (left) and ROC-AUC (right) for baseline and tuned models.

Because improvements are small and within cross-validation variability, tuning was interpreted as stabilising and slightly refining the models rather than completely 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 L_2 -regularised logistic regression (linear baseline). All models were trained on the engineered feature set and evaluated on the same held-out test data.

760 4.4.1 Confusion Matrices and Error Patterns

761 Classification reports and confusion matrices for the four models reveal consistent
762 patterns. CatBoost and gradient boosting both reached overall accuracy of ap-
763 proximately 0.80 with similar macro-averaged F1 scores (~ 0.80). For CatBoost,
764 precision and recall for clean reads were 0.73 and 0.95, respectively, while for
765 chimeric reads they were 0.92 and 0.66 ($F1 = 0.77$). Gradient boosting showed
766 nearly identical trade-offs.

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

771 Across all models, errors were asymmetric. False negatives (chimeric reads
772 predicted as clean) were more frequent than false positives. For example, CatBoost
773 misclassified 1,369 chimeric reads as clean but only 215 clean reads as chimeric.
774 This pattern indicates that the models are conservative and prioritise avoiding
775 false chimera calls at the expense of missing some true chimeras.

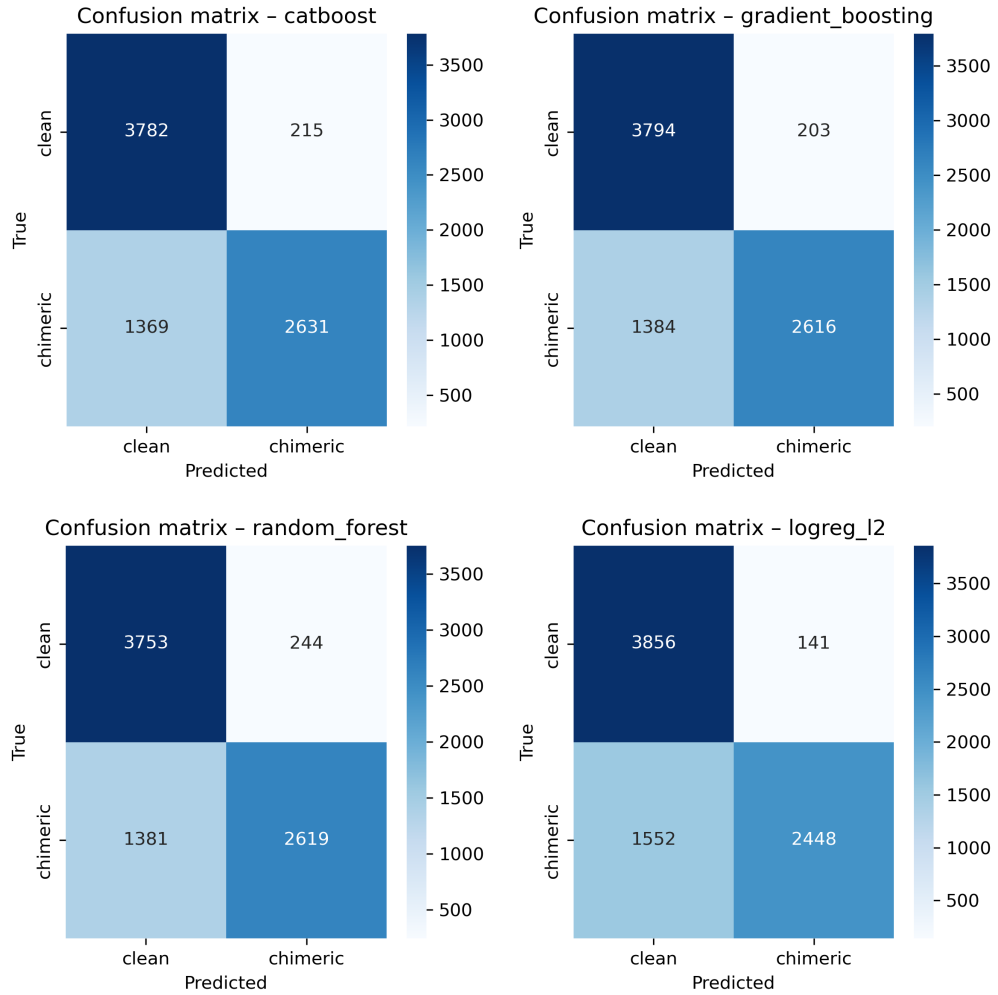


Figure 4.4: Confusion matrices for the four representative models on the held-out test set.

4.4.2 ROC and Precision–Recall Curves

Receiver operating characteristic (ROC) and precision–recall (PR) curves as shown in 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.

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

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

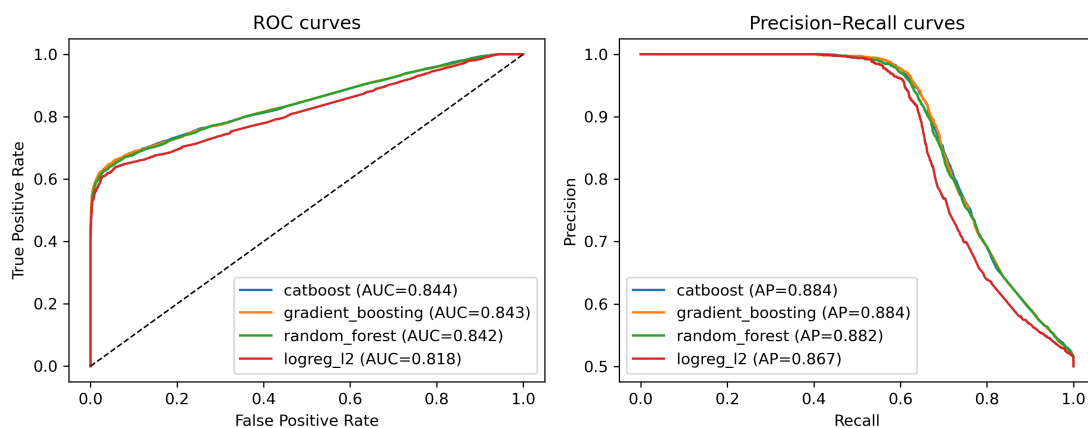


Figure 4.5: ROC (left) and precision–recall (right) curves for the four representative models on the held-out test set.

788 4.5 Feature Importance

789 4.5.1 Permutation Importance of Individual Features

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

793 Logistic Regression.

794 As shown in Figure 4.6, the total number of clipped bases consistently pro-
795 vides a strong predictive signal, particularly in Random Forest, Gradient Boosting,
796 and L₂-regularized Logistic Regression. CatBoost differs by assigning the highest
797 importance to k-mer divergence metrics such as `kmer_js_divergence`, which cap-
798 ture subtle sequence changes resulting from structural variants or PCR-induced
799 chimeras. Soft-clipping features (`softclip_left` and `softclip_right`) provide
800 more information around breakpoints, complementing these primary signals in all
801 models except Gradient Boosting. L₂-regularized Logistic Regression relies more
802 on alignment-based split-read metrics.

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

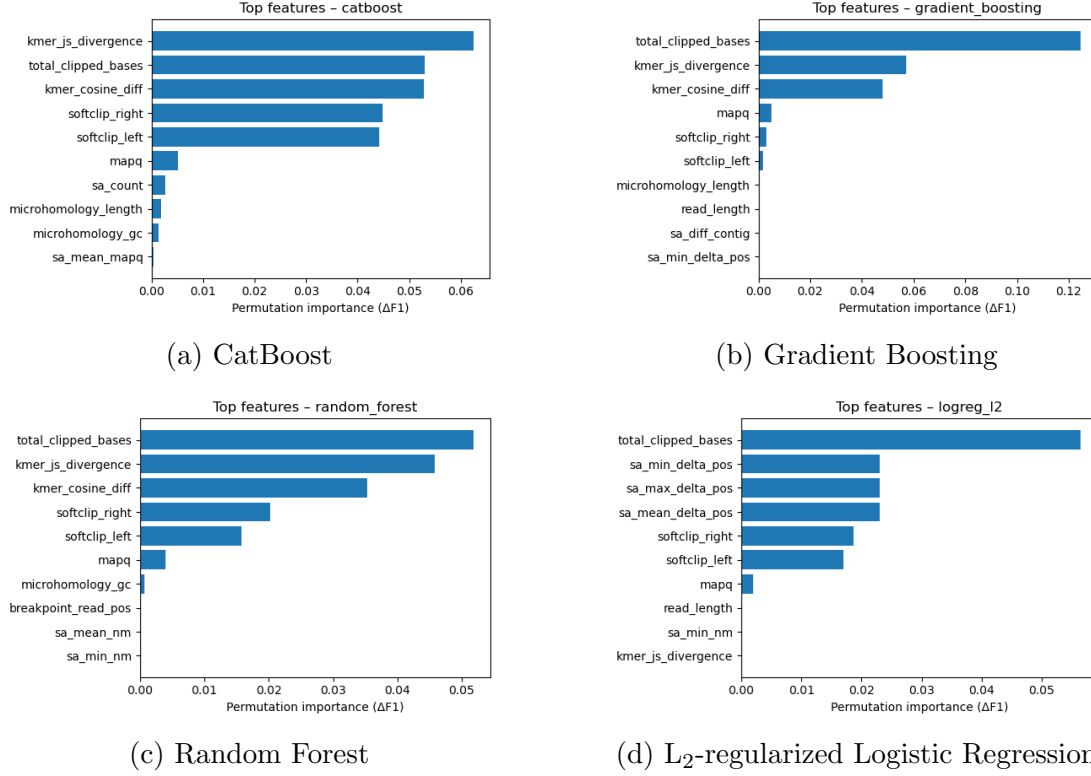


Figure 4.6: Permutation-based feature importance for four representative classifiers.

4.5.2 Feature Family Importance

To evaluate the contribution of broader signals, features were grouped into five families: SA-structure (supplementary alignment and segment metrics, e.g., `has_sa`, `sa_count`, `sa_min_delta_pos`, `sa_mean_nm`, etc.), Clipping (`softclip_left`, `softclip_right`, `total_clipped_bases`, `breakpoint_read_pos`), Kmer-jump (`kmer_cosine_diff`, `kmer_js_divergence`), Micro-homology (`microhomology_length`, `microhomology_gc`), and Other (e.g., `mapq`).

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

817 by Kmer_jump (0.12), with Other features contributing minimally (0.005) and
818 SA_structure (0.003) and Micro_homology (0.003) providing minimal predictive
819 power. Gradient Boosting shows a similar trend, with Clipping (0.13) domi-
820 nating, Kmer_jump (0.11) secondary, and the remaining families contributing
821 negligibly. Random Forest integrates both Clipping (0.088) and Kmer_jump
822 (0.08) effectively, while SA_structure, Micro_homology, and Other remain minor
823 contributors. L₂-regularized Logistic Regression emphasizes Clipping (0.09)
824 and SA_structure (0.07), with Kmer_jump and Micro_homology having minimal
825 impact.

826 Both feature-level and aggregated analyses indicate that detection of chimeric
827 reads in this dataset relies primarily on alignment irregularities (Clipping) and
828 k-mer compositional shifts (Kmer_jump), which often arise from PCR-induced
829 template switching events, while explicit microhomology features contribute min-
830 imally.

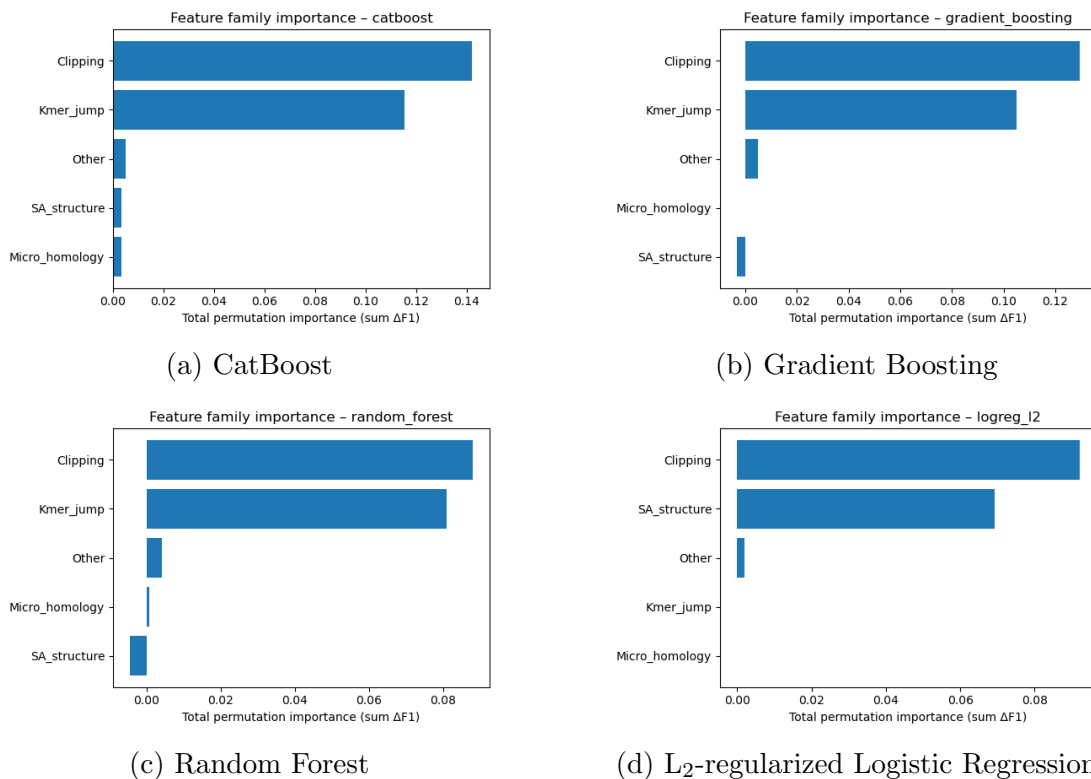


Figure 4.7: Aggregated feature family importance across four models.

831 4.6 Summary of Findings

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

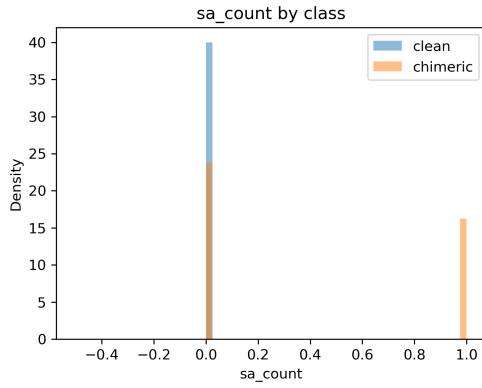
838 Feature importance analysis highlighted alignment breakpoints, such as clip-
 839 ping, and abrupt shifts in k-mer composition as the main contributors to predic-

840 tive power. Microhomology metrics and supplementary alignment features had
841 minimal impact. These findings suggest that alignment-based and k-mer-based
842 features alone are sufficient for training classifiers to detect mitochondrial PCR-
843 induced chimeric reads under the conditions tested.

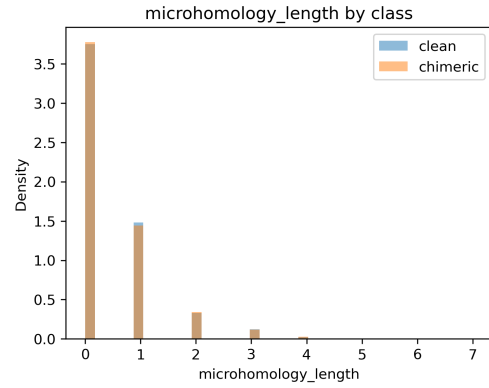
844 **Appendix A**

845 **Exploratory Data Analysis**

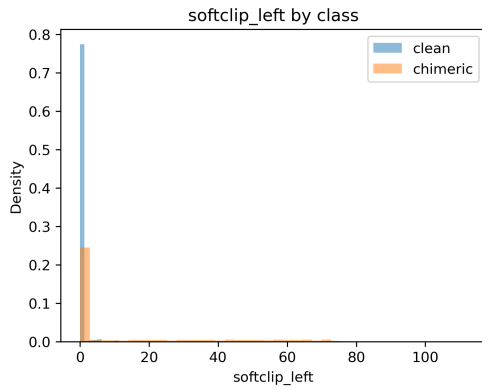
846 **A.1 Histograms of Key Features**



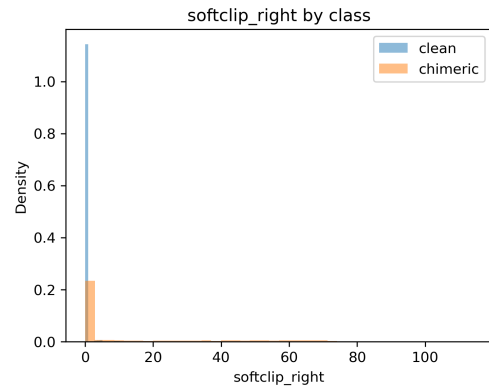
(a) sa_count



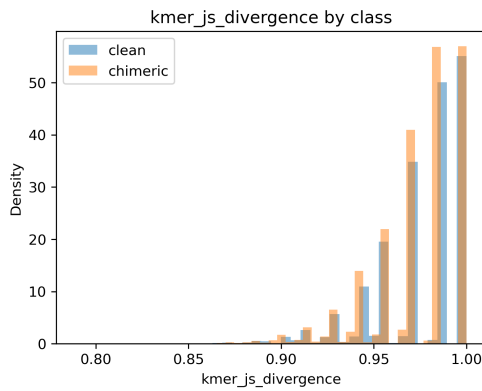
(b) Microhomology length



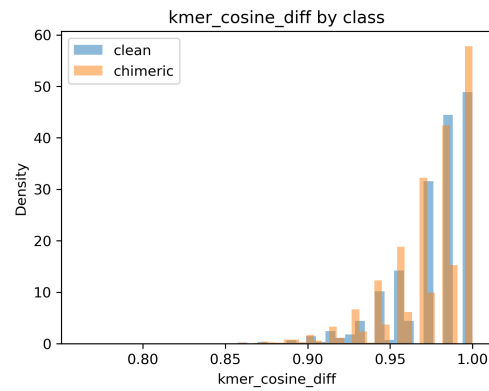
(c) softclip_left



(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.

847 References

- 848 Anderson, S., Bankier, A., Barrell, B., Bruijn, M., Coulson, A., Drouin, J., ...
849 Young, I. (1981, 04). Sequence and organization of the human mitochondrial
850 genome. *Nature*, 290, 457-465. doi: 10.1038/290457a0
- 851 Arango, G., Garner, E., Pruden, A., Heath, L., Vikesland, P., & Zhang, L. (2018,
852 02). Deeparg: A deep learning approach for predicting antibiotic resistance
853 genes from metagenomic data. *Microbiome*, 6. doi: 10.1186/s40168-018
854 -0401-z
- 855 Bentley, D. R., Balasubramanian, S., Swerdlow, H. P., Smith, G. P., Milton, J.,
856 Brown, C. G., ... Smith, A. J. (2008). Accurate whole human genome
857 sequencing using reversible terminator chemistry. *Nature*, 456(7218), 53–
858 59. doi: 10.1038/nature07517
- 859 Boore, J. L. (1999). Animal mitochondrial genomes. *Nucleic Acids Research*,
860 27(8), 1767–1780. doi: 10.1093/nar/27.8.1767
- 861 Cameron, S. L. (2014). Insect mitochondrial genomics: Implications for evolution
862 and phylogeny. *Annual Review of Entomology*, 59, 95–117. doi: 10.1146/
863 annurev-ento-011613-162007
- 864 Dierckxsens, N., Mardulyn, P., & Smits, G. (2017). Novoplasty: de novo assembly
865 of organelle genomes from whole genome data. *Nucleic Acids Research*,

866 45(4), e18. doi: 10.1093/nar/gkw955

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

868 [.com/usearch/manual7/uchime_practical.html](https://www.drive5.com/usearch/manual7/uchime_practical.html)

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

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

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

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

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

874 27(16), 2194–2200. doi: 10.1093/bioinformatics/btr381

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

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

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

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

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

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

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

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

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

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

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

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

887 10.1093/nar/gkt371

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

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

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

891 10.1186/s13059-020-02154-5

- 892 Judo, M. S. B., Wedel, W. R., & Wilson, B. H. (1998). Stimulation and sup-
893 pression of pcr-mediated recombination. *Nucleic Acids Research*, 26(7),
894 1819–1825. doi: 10.1093/nar/26.7.1819
- 895 Labrador, K., Agmata, A., Palermo, J. D., Ravago-Gotanco, R., & Pante, M. J.
896 (2021). Mitochondrial dna reveals genetically structured haplogroups of
897 bali sardinella (*sardinella lemuru*) in philippine waters. *Regional Studies in*
898 *Marine Science*, 41, 101588. doi: 10.1016/j.rsma.2020.101588
- 899 Li, H. (2018, 05). Minimap2: pairwise alignment for nucleotide sequences. *Bioin-*
900 *formatics*, 34(18), 3094–3100. Retrieved from [https://doi.org/10.1093/](https://doi.org/10.1093/bioinformatics/bty191)
901 [bioinformatics/bty191](https://doi.org/10.1093/bioinformatics/bty191) doi: 10.1093/bioinformatics/bty191
- 902 Liang, Q., Bible, P. W., Liu, Y., Zou, B., & Wei, L. (2020, 02). Deepmi-
903 crobes: taxonomic classification for metagenomics with deep learning. *NAR*
904 *Genomics and Bioinformatics*, 2(1), lqaa009. Retrieved from [https://](https://doi.org/10.1093/nargab/lqaa009)
905 doi.org/10.1093/nargab/lqaa009 doi: 10.1093/nargab/lqaa009
- 906 Metzker, M. L. (2010). Sequencing technologies — the next generation. *Nature*
907 *Reviews Genetics*, 11(1), 31–46. doi: 10.1038/nrg2626
- 908 Mysara, M., Saeys, Y., Leys, N., Raes, J., & Monsieurs, P. (2015). Catch,
909 an ensemble classifier for chimera detection in 16s rna sequencing stud-
910 ies. *Applied and Environmental Microbiology*, 81(5), 1573–1584. Retrieved
911 from <https://journals.asm.org/doi/abs/10.1128/aem.02896-14> doi:
912 10.1128/AEM.02896-14
- 913 Peccoud, J., Lequime, S., Moltini-Conclois, I., Giraud, I., Lambrechts, L., &
914 Gilbert, C. (2018, 04). A survey of virus recombination uncovers canon-
915 ical features of artificial chimeras generated during deep sequencing li-
916 brary preparation. *G3 Genes—Genomes—Genetics*, 8(4), 1129–1138. Re-
917 trieved from <https://doi.org/10.1534/g3.117.300468> doi: 10.1534/

g3.117.300468

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

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

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

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

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

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

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

944 *Sciences*, 40(11), 701-714. Retrieved from <https://www.sciencedirect>
 945 [.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)
 946 [10.1016/j.tibs.2015.08.006](https://doi.org/10.1016/j.tibs.2015.08.006)
 947 Vervier, K., Mahé, P., Tournoud, M., Veyrieras, J.-B., & Vert, J.-P. (2015,
 948 11). Large-scale machine learning for metagenomics sequence classifica-
 949 tion. *Bioinformatics*, 32(7), 1023-1032. Retrieved from [https://doi.org/](https://doi.org/10.1093/bioinformatics/btv683)
 950 [10.1093/bioinformatics/btv683](https://doi.org/10.1093/bioinformatics/btv683) doi: 10.1093/bioinformatics/btv683
 951 Willette, D., Bognot, E., Mutia, M. T., & Santos, M. (2011). *Biology and ecology*
 952 *of sardines in the philippines: A review* (Vol. 13; Tech. Rep. No. 1). NFRDI
 953 Technical Paper Series. Retrieved from [https://nfrdi.da.gov.ph/tpjf/](https://nfrdi.da.gov.ph/tpjf/etc/Willette%20et%20al.%20Sardines%20Review.pdf)
 954 [etc/Willette%20et%20al.%20Sardines%20Review.pdf](https://nfrdi.da.gov.ph/tpjf/etc/Willette%20et%20al.%20Sardines%20Review.pdf)