

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

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

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

13 by

14 Duranne Duran
15 Yvonne Lin
16 Daniella Pailden

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

19 January 1, 2026

Abstract

21 Next-generation sequencing (NGS) platforms have advanced research but re-
22 main susceptible to artifacts such as PCR-induced chimeras that compromise
23 mitochondrial genome assembly. These artificial hybrid sequences are prob-
24 lematic for small, circular, and repetitive mitochondrial genomes, where they
25 can generate fragmented contigs and false junctions. Existing detection tools,
26 such as UCHIME, are optimized for amplicon-based microbial community ana-
27 lysis and depend on reference databases or abundance assumptions unsuitable
28 for organellar assembly. To address this gap, this study presents MitoChime,
29 a machine learning pipeline for detecting PCR-induced chimeric reads in *Sar-*
30 *dinella 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

⁴³ **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, Feature Selection, and Interpretation	29

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

⁹⁴ List of Figures

⁹⁵	3.1 Process diagram of the study workflow.	20
⁹⁶	4.1 Boxplots of key features by class	38
⁹⁷	4.2 Test F1 of all baseline classifiers, showing that no single model	
⁹⁸	clearly dominates and several achieve comparable performance. . .	40
⁹⁹	4.3 Comparison of test F1 (left) and ROC–AUC (right) for baseline	
¹⁰⁰	and tuned models.	41
¹⁰¹	4.4 Confusion matrices for the four representative models on the held-	
¹⁰²	out test set.	44
¹⁰³	4.5 ROC (left) and precision–recall (right) curves for the four represen-	
¹⁰⁴	tative models on the held-out test set.	45
¹⁰⁵	4.6 Permutation-based feature importance for four representative clas-	
¹⁰⁶	sifiers.	47
¹⁰⁷	4.7 Aggregated feature family importance across four models.	49

111 List of Tables

₁₂₀ **Chapter 1**

₁₂₁ **Introduction**

₁₂₂ **1.1 Overview**

₁₂₃ The rapid advancement of next-generation sequencing (NGS) technologies has
₁₂₄ transformed genomic research by enabling high-throughput and cost-effective
₁₂₅ DNA analysis (Metzker, 2010). Among current platforms, Illumina sequencing
₁₂₆ remains the most widely adopted, capable of producing millions of short reads
₁₂₇ that can be assembled into reference genomes or analyzed for genetic variation
₁₂₈ (Bentley et al., 2008; Glenn, 2011). Despite its high base-calling accuracy,
₁₂₉ Illumina sequencing is prone to artifacts introduced during library preparation,
₁₃₀ particularly polymerase chain reaction (PCR)-induced chimeras, which are ar-
₁₃₁ tificial hybrid sequences that do not exist in the true genome (Judo, Wedel, &
₁₃₂ Wilson, 1998).

₁₃₃ 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

₁₅₉ solution for improving mitochondrial genome reconstruction.

₁₆₀ 1.2 Problem Statement

₁₆₁ Chimeric reads can distort assembly graphs and cause misassemblies, with par-
₁₆₂ ticularly severe effects in mitochondrial genomes (Boore, 1999; Cameron, 2014).
₁₆₃ Existing assembly pipelines such as GetOrganelle, MITObim, and NOVOPlasty
₁₆₄ assume that sequencing reads are free of such artifacts (Dierckxsens et al., 2017;
₁₆₅ Hahn et al., 2013; Jin et al., 2020). At PGC Visayas, several mitochondrial as-
₁₆₆ semblies have failed or yielded incomplete contigs despite sufficient coverage, sug-
₁₆₇ gesting that undetected chimeric reads compromise assembly reliability. Mean-
₁₆₈ while, existing chimera detection tools such as UCHIME and VSEARCH were
₁₆₉ developed primarily for amplicon-based community analysis and rely heavily on
₁₇₀ reference or taxonomic comparisons (Edgar, Haas, Clemente, Quince, & Knight,
₁₇₁ 2011; Rognes, Flouri, Nichols, Quince, & Mahé, 2016). These approaches are un-
₁₇₂ suitable for single-species organellar data, where complete reference genomes are
₁₇₃ often unavailable.

₁₇₄ 1.3 Research Objectives

₁₇₅ 1.3.1 General Objective

₁₇₆ This study aims to develop and evaluate a machine learning-based pipeline (Mi-
₁₇₇ toChime) that detects PCR-induced chimeric reads in *Sardinella lemuru* mito-

₁₇₈ chondrial sequencing data in order to improve the quality and reliability of down-
₁₇₉ stream mitochondrial genome assemblies.

₁₈₀ **1.3.2 Specific Objectives**

₁₈₁ Specifically, the study aims to:

- ₁₈₂ 1. construct simulated *Sardinella lemuru* Illumina paired-end datasets contain-
₁₈₃ ing both clean and PCR-induced chimeric reads,
- ₁₈₄ 2. extract alignment-based and sequence-based features such as k-mer compo-
₁₈₅ sition, junction complexity, and split-alignment counts from both clean and
₁₈₆ chimeric reads,
- ₁₈₇ 3. train, validate, and compare supervised machine learning models for classi-
₁₈₈ fying reads as clean or chimeric,
- ₁₈₉ 4. determine feature importance and identify indicators of PCR-induced
₁₉₀ chimerism,
- ₁₉₁ 5. integrate the optimized classifier into a modular and interpretable pipeline
₁₉₂ deployable on standard computing environments at PGC Visayas.

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

₁₉₄ This study focuses solely on PCR-induced chimeric reads in *Sardinella lemuru*
₁₉₅ mitochondrial sequencing data, with the species choice guided by four consid-
₁₉₆ erations: (1) to limit interspecific variation in mitochondrial genome size, GC

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

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

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

222 patterns.

223 1.5 Significance of the Research

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

²³⁴ **Chapter 2**

²³⁵ **Review of Related Literature**

²³⁶ This chapter presents an overview of the literature relevant to the study. It
²³⁷ discusses the biological and computational foundations underlying mitochondrial
²³⁸ genome analysis and assembly, as well as existing tools, algorithms, and techniques
²³⁹ related to chimera detection and genome quality assessment. The chapter aims to
²⁴⁰ highlight the strengths, limitations, and research gaps in current approaches that
²⁴¹ motivate the development of the present study.

²⁴² **2.1 The Mitochondrial Genome**

²⁴³ Mitochondrial genome (mtDNA) is a small, typically circular molecule found in
²⁴⁴ most eukaryotes. It encodes essential genes involved in oxidative phosphorylation
²⁴⁵ and energy metabolism. Because of its conserved structure, mtDNA has become
²⁴⁶ a valuable genetic marker for studies in population genetics and phylogenetics
²⁴⁷ (Anderson et al., 1981; Boore, 1999). In animal species, the mitochondrial genome

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

253 **2.1.1 Mitochondrial Genome Assembly**

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

268 2.2 PCR Amplification and Chimera Formation

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

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

288 **2.3 Existing Traditional Approaches for Chimera**

289 **Detection**

290 Several computational tools have been developed to identify chimeric sequences in
291 NGS datasets. These tools generally fall into two categories: reference-based and
292 de novo approaches. Reference-based chimera detection, also known as database-
293 dependent detection, is one of the earliest and most widely used computational
294 strategies for identifying chimeric sequences in amplicon-based community studies.
295 These methods rely on the comparison of each query sequence against a curated,
296 high-quality database of known, non-chimeric reference sequences (Edgar et al.,
297 2011).

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

312 sequence correspond to distinct high-abundance sequences.

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

318 2.3.1 UCHIME

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

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

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

340 **2.3.2 UCHIME2**

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

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

357 2.3.3 CATch

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

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

374 Benchmarking in both reference-based and de novo modes demonstrated signif-
375 icant performance improvements. CATCh achieved sensitivities of approximately
376 85 percent in reference-based mode and 92 percent in de novo mode, with corre-
377 sponding specificities of approximately 96 percent and 95 percent. These results
378 indicate that CATCh detected 7 to 12 percent more chimeras than any individual
379 algorithm while maintaining high precision.

380 **2.3.4 ChimPipe**

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

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

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

398 2.4 Machine Learning Approaches for Chimera 399 and Sequence Quality Detection

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

411 2.4.1 Feature-Based Representations of Genomic Sequences

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

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

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

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

437 2.5 Synthesis of Chimera Detection Approaches

438 To provide an integrated overview of the literature discussed in this chapter, Ta-
439 ble 2.1 summarizes the major chimera detection studies, their methodological
440 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.

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

⁴⁵³ **Chapter 3**

⁴⁵⁴ **Research Methodology**

⁴⁵⁵ This chapter outlines the steps involved in completing the study, including data
⁴⁵⁶ gathering, generating simulated mitochondrial Illumina reads, preprocessing and
⁴⁵⁷ indexing the data, developing a feature extraction pipeline to obtain read-level fea-
⁴⁵⁸ tures, applying machine learning algorithms for chimera detection, implementing
⁴⁵⁹ feature selection methods, and validating and comparing model performance.

⁴⁶⁰ **3.1 Research Activities**

⁴⁶¹ As illustrated in Figure 3.1, this study carried out a sequence of procedures to
⁴⁶² detect PCR-induced chimeric reads in mitochondrial genomes. The process began
⁴⁶³ with collecting a mitochondrial reference sequence of *Sardinella lemuru* from the
⁴⁶⁴ National Center for Biotechnology Information (NCBI) database, which was used
⁴⁶⁵ as a reference for generating simulated clean and chimeric reads. These reads
⁴⁶⁶ were subsequently indexed and mapped. The resulting collections then passed

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

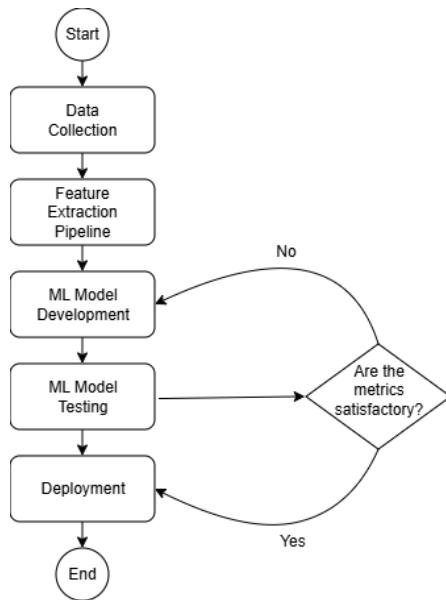


Figure 3.1: Process diagram of the study workflow.

473 3.1.1 Data Collection

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

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

479 proximately one (1) week.

480 **Data Preprocessing**

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

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

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

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

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

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

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

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

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

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

```
513  samtools view -bS clean.sam -o clean.bam  
514  samtools view -bS chimeric.sam -o chimeric.bam  
515  
516  samtools sort clean.bam -o clean.sorted.bam  
517  samtools index clean.sorted.bam  
518  
519  samtools sort chimeric.bam -o chimeric.sorted.bam  
520  samtools index chimeric.sorted.bam
```

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

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

531 **3.1.2 Feature Extraction Pipeline**

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

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

543 total duration of approximately two (2) weeks.

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

549 **Supplementary Alignment and Alignment-Structure Features**

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

557 After parsing, the function **sa_feature_stats** was applied to establish the fun-
558 damental split indicators, **has_sa** and **sa_count**. Along with these initial counts,
559 the function aggregated metrics related to the structure and reliability of the
560 split alignments, including the number of alignment segments, strand consistency,
561 minimum, maximum, and mean distance between split segments, and summary
562 statistics of mapping quality and mismatch counts across segments.

563 **K-mer Composition Difference**

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

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

576 **Microhomology**

577 The process of extracting the microhomology feature also started by using
578 `infer_breakpoints` to identify a candidate junction. Once a breakpoint was
579 established, the script scanned a ± 40 base-pair window surrounding the break-
580 point and applied the function `longest_suffix_prefix_overlap` to identify the
581 longest exact suffix-prefix overlap between the left and right read segments. This
582 overlap, representing consecutive bases shared at the junction, was recorded as
583 `microhomology_length` in the dataset. The 40 base-pair window was chosen
584 to ensure that short shared sequences at or near the breakpoint were captured

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

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

595 3.1.3 Machine Learning Model Development

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

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

607 (20%) subsets using stratified sampling to preserve the 1:1 ratio of clean to
608 chimeric reads. Model development and evaluation were implemented in Python
609 (Version 3.11) using the `scikit-learn`, `xgboost`, `lightgbm`, and `catboost` li-
610 braries. A broad panel of classification algorithms was then benchmarked on the
611 training data to obtain a fair comparison of different model families under identical
612 feature conditions. The panel included a trivial dummy classifier, L_2 -regularized
613 logistic regression, a calibrated linear support vector machine (SVM), k -nearest
614 neighbours, Gaussian Naïve Bayes, decision-tree ensembles (Random Forest, Ex-
615 tremely Randomized Trees, and Bagging with decision trees), gradient boosting
616 methods (Gradient Boosting, XGBoost, LightGBM, and CatBoost), and a shallow
617 multilayer perceptron (MLP).

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

626 **3.1.4 Model Benchmarking, Hyperparameter Optimiza- 627 tion, and Evaluation**

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

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

Second, hyperparameter optimization was conducted for each of the ten selected models using randomized search with five-fold stratified cross-validation (`RandomizedSearchCV`). For tree-based ensembles, the search space included the number of trees, maximum depth, minimum samples per split and per 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

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

656 **3.1.5 Feature Importance, Feature Selection, and Inter-**
657 **pretation**

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

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

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

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

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

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

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

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

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

3.1.6 Validation and Testing

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

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

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

726 **3.1.7 Documentation**

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

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

738 For manuscript preparation and supplementary materials, Overleaf (\LaTeX)
739 was used to produce publication-quality formatting and consistent referencing.

740 **3.2 Calendar of Activities**

741 Table 3.1 presents the project timeline in the form of a Gantt chart, where each
742 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	• • • •	• • • •	• • • •	• • • •	• • • •	• • • •	• • • •

743 **Chapter 4**

744 **Results and Discussion**

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

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

754 **4.1 Descriptive Analysis of Features**

755 **4.1.1 Summary Statistics Per Class**

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

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

765 **Alignment and Supplementary Alignment Features**

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

`772 Clipping-Based Features`

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

`778 K-mer Distribution Features`

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

`783 Microhomology Features`

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

`789 Overall, the summary statistics indicate that alignment-based and clipping-`
`790 based features provide the strongest class separation, k-mer features contribute`
`791 limited but complementary signal, and microhomology features exhibit minimal`

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

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

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

794 4.1.2 Boxplots By Class

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

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

801 of split alignments.

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

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

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

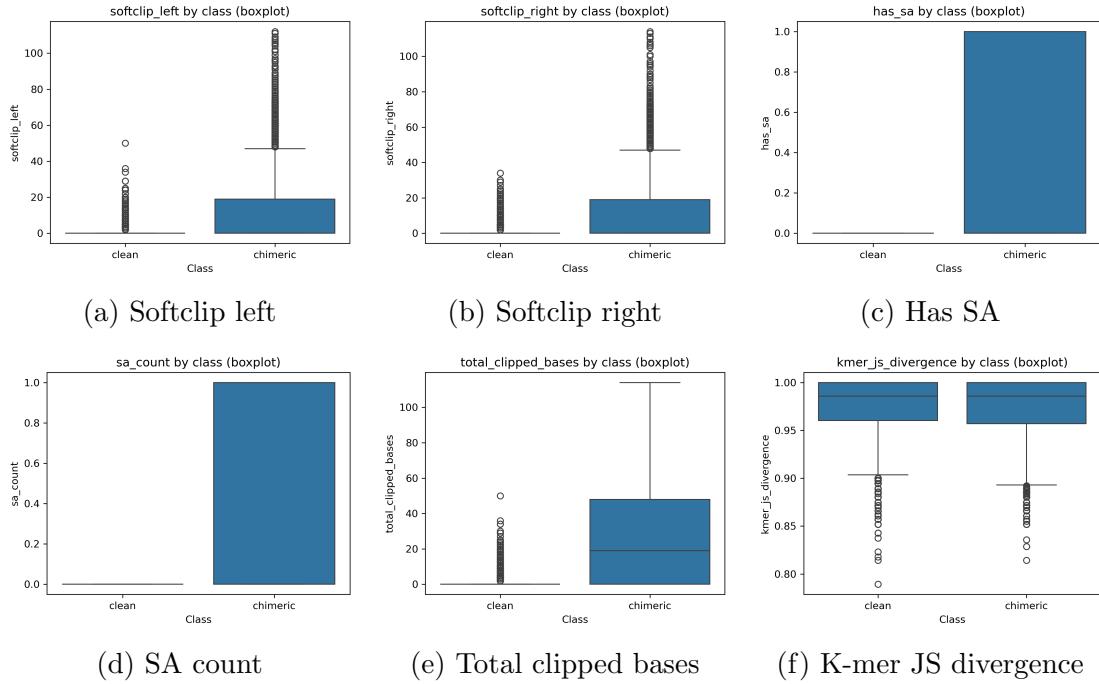


Figure 4.1: Boxplots of key features by class

809 4.2 Baseline Classification Performance

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

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

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

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

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

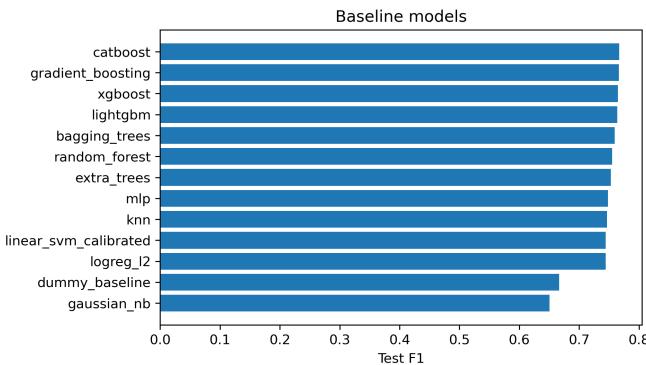


Figure 4.2: Test F1 of all baseline classifiers, showing that no single model clearly dominates and several achieve comparable performance.

826 4.3 Effect of Hyperparameter Tuning

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

831 or slightly worse.

832 CatBoost, gradient boosting, LightGBM, XGBoost, random forest, bagging
833 trees, and MLP all experienced small increases in test F1 (typically $\Delta F1 \approx 0.002$ –
834 0.009) and ROC–AUC (up to $\Delta AUC \approx 0.008$). After tuning, CatBoost remained
835 the best performer with test accuracy 0.80, precision 0.92, recall 0.66, F1-score
836 0.77, and ROC–AUC 0.84. Gradient boosting achieved almost identical perfor-
837 mance (F1 0.77, AUC 0.84). Random forest and bagging trees also improved to
838 F1 scores around 0.76 with AUC ≈ 0.84 .

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

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

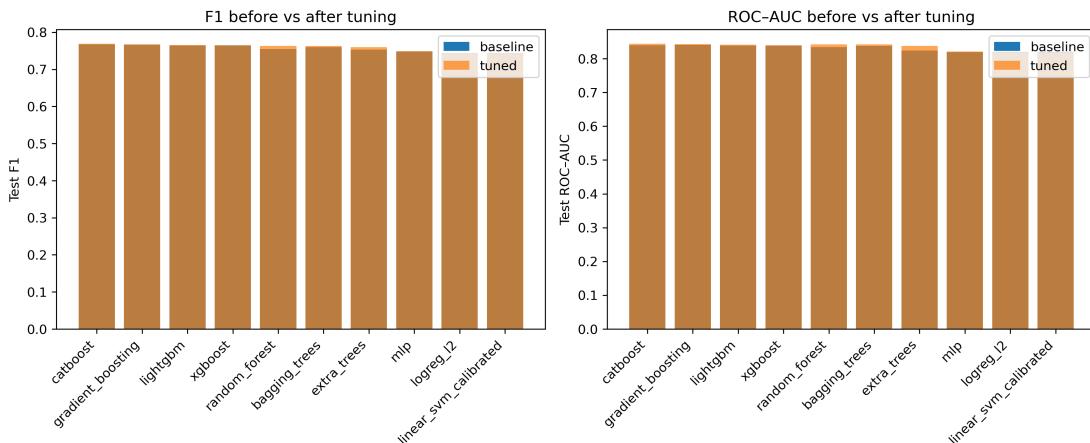


Figure 4.3: Comparison of test F1 (left) and ROC–AUC (right) for baseline and tuned models.

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

842 **4.4 Detailed Evaluation of Representative Mod- 843 els**

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

850 **4.4.1 Confusion Matrices and Error Patterns**

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

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

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

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

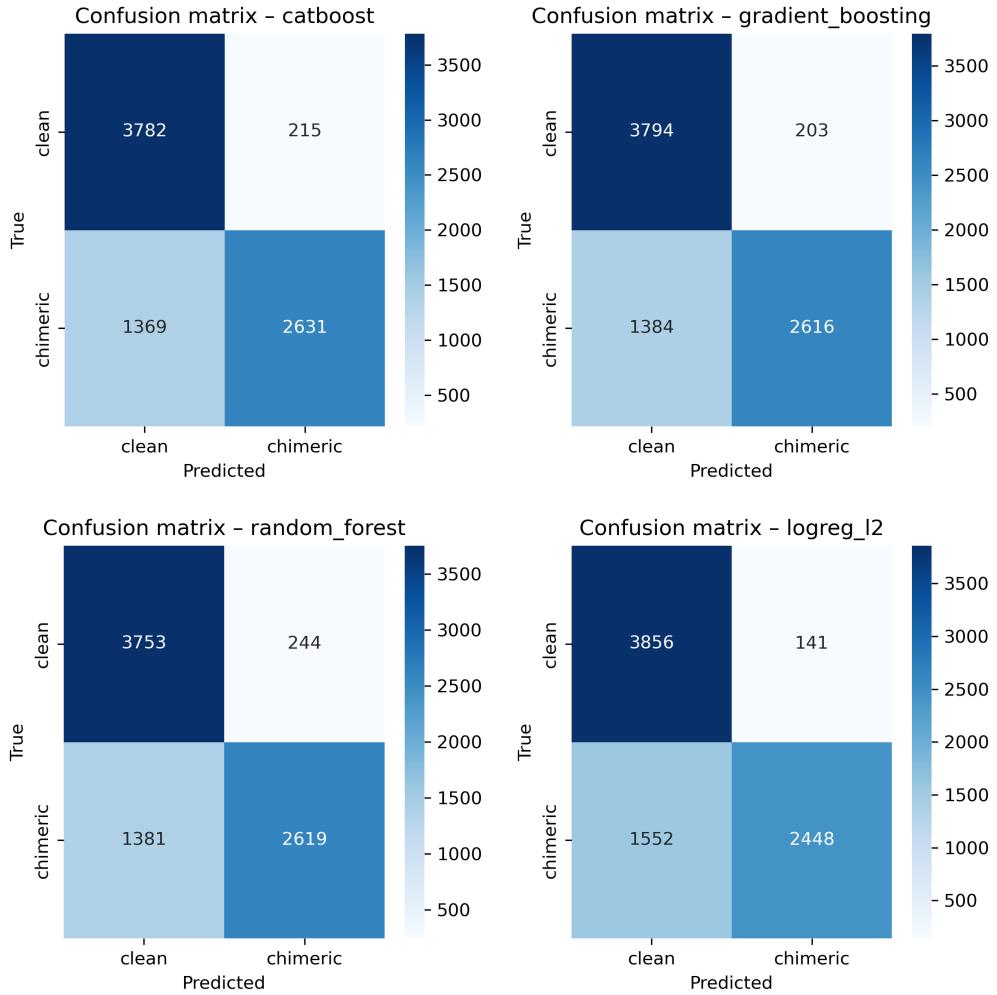


Figure 4.4: Confusion matrices for the four representative models on the held-out test set.

⁸⁶⁹ 4.4.2 ROC and Precision–Recall Curves

⁸⁷⁰ Receiver operating characteristic (ROC) and precision–recall (PR) curves as
⁸⁷¹ shown in Figure 4.5 further support the similarity among the top models. The
⁸⁷² three tree-based ensembles (CatBoost, gradient boosting, random forest) achieved
⁸⁷³ ROC–AUC values of approximately 0.84 and average precision (AP) around 0.88.

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

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

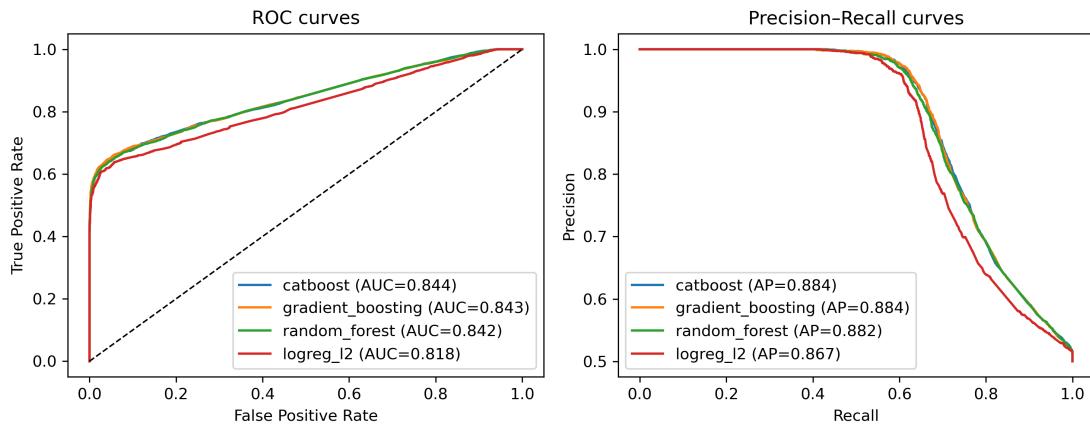


Figure 4.5: ROC (left) and precision–recall (right) curves for the four representative models on the held-out test set.

881 4.5 Feature Importance

882 4.5.1 Permutation Importance of Individual Features

883 To understand how each classifier made predictions, feature importance was quan-
884 tified using permutation importance. This analysis was applied to four represen-
885 tative models: CatBoost, Gradient Boosting, Random Forest, and L₂-regularized

886 Logistic Regression.

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

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

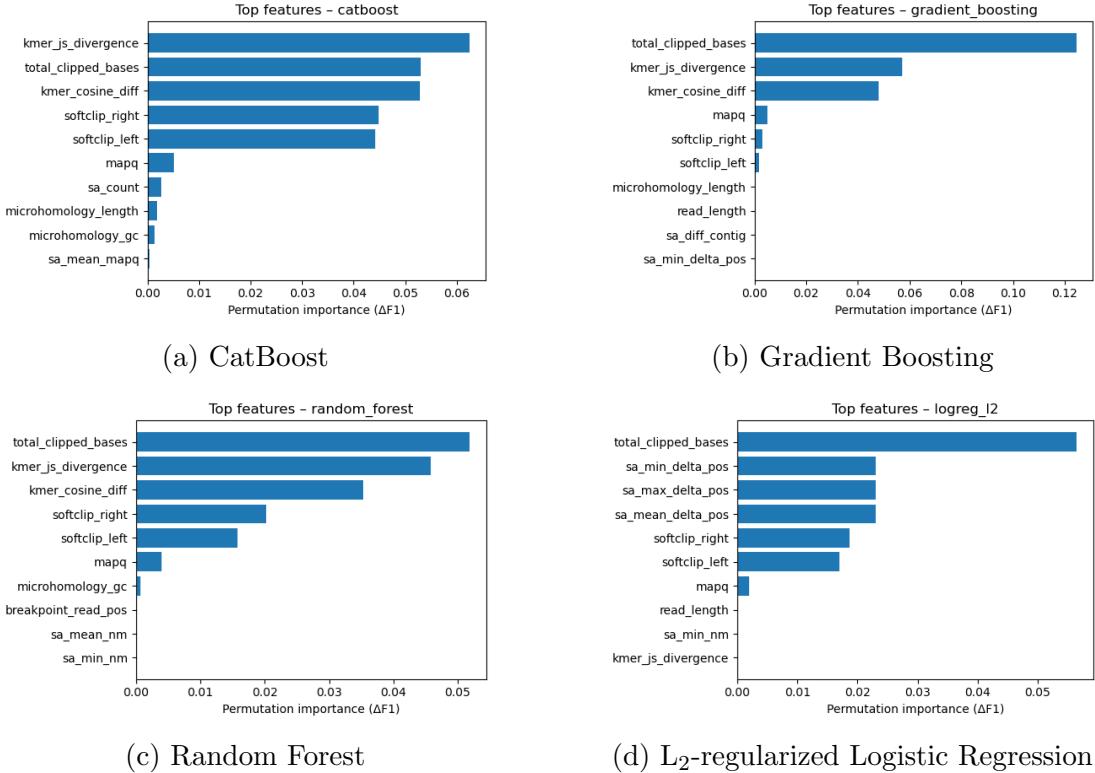


Figure 4.6: Permutation-based feature importance for four representative classifiers.

901 4.5.2 Feature Family Importance

902 To evaluate the contribution of broader signals, features were grouped into
 903 five families: SA_structure (supplementary alignment and segment met-
 904 rics, e.g., has_sa, sa_count, sa_min_delta_pos, sa_mean_nm, etc.), Clipping
 905 (softclip_left, softclip_right, total_clipped_bases, breakpoint_read_pos),
 906 Kmer_jump (kmer_cosine_diff, kmer_js_divergence), Micro_homology ((
 907 microhomology_length, microhomology_gc), and Other (e.g., mapq).

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

910 by Kmer_jump (0.12), with Other features contributing minimally (0.005) and
911 SA_structure (0.003) and Micro_homology (0.003) providing minimal predictive
912 power. Gradient Boosting shows a similar trend, with Clipping (0.13) domi-
913 nating, Kmer_jump (0.11) secondary, and the remaining families contributing
914 negligibly. Random Forest integrates both Clipping (0.088) and Kmer_jump
915 (0.08) effectively, while SA_structure, Micro_homology, and Other remain minor
916 contributors. L₂-regularized Logistic Regression emphasizes Clipping (0.09)
917 and SA_structure (0.07), with Kmer_jump and Micro_homology having minimal
918 impact.

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

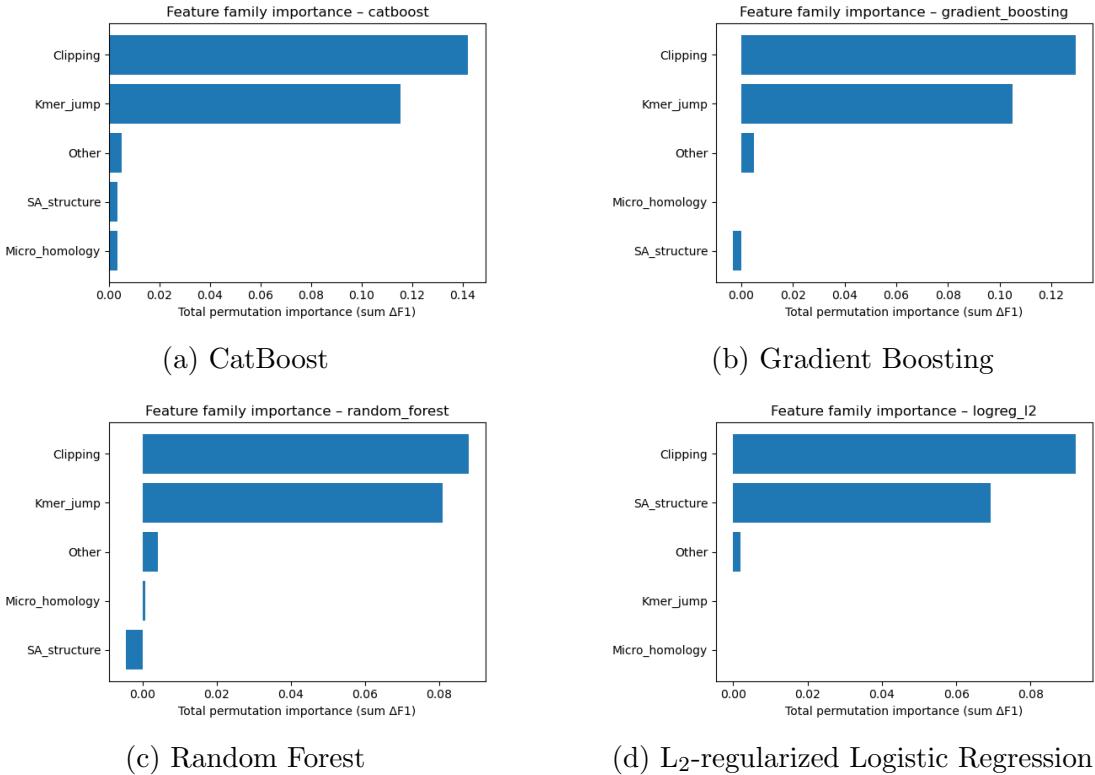


Figure 4.7: Aggregated feature family importance across four models.

924 4.6 Feature Selection

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

930 4.6.1 Cumulative Importance Curve

931 The cumulative importance curve was computed using the tuned CatBoost clas-
932 sifier. Figure 4.8 illustrates the contribution of features sorted by importance.
933 The curve rises steeply for the first few features and then gradually plateaus,
934 indicating that a small number of features capture most of the model’s pre-
935 dictive power. A cumulative importance of 95% is reached at $k = 4$ features,
936 which are `total_clipped_bases`, `kmer_js_divergence`, `kmer_cosine_diff`, and
937 `softclip_left`.

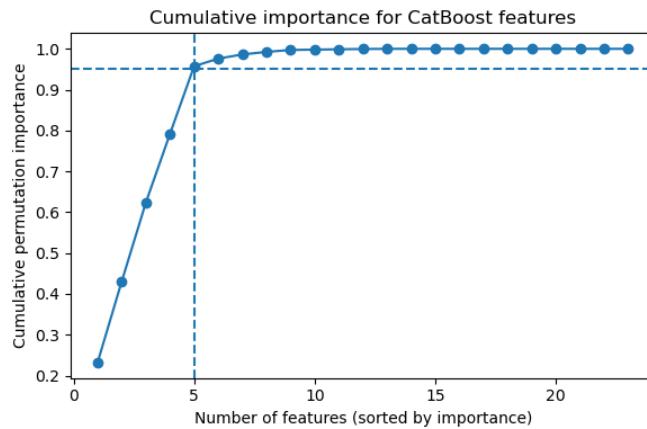


Figure 4.8: Cumulative importance curve of features sorted by importance.

938 4.6.2 Performance Comparison Across Feature Sets

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

944 equivalent performance with an F1 of 0.7670 and a ROC–AUC of 0.8353. An
 945 ablation model excluding microhomology features (`microhomology_length` and
 946 `microhomology_gc`) also performed comparably, with an F1 of 0.7679 and ROC–
 947 AUC of 0.8447. These results indicate that clipping and k-mer features capture
 948 almost all predictive signal, while microhomology features are largely redundant
 949 in this dataset.

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

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

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

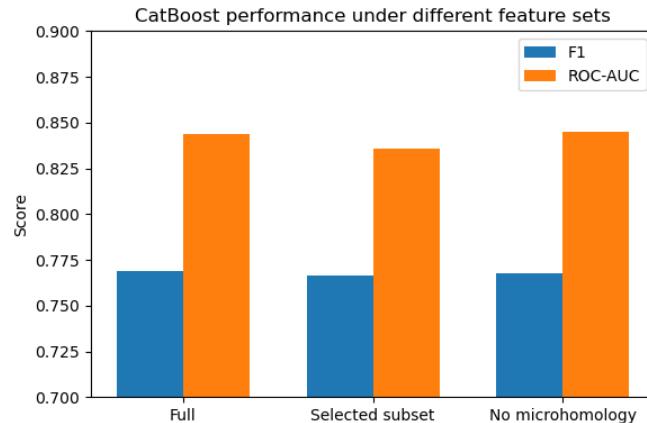


Figure 4.9: Comparison of F1 and ROC–AUC for the full, top-4 selected, and no-microhomology feature set variants.

953 **4.6.3 Interpretation and Final Feature Set Choice**

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

961 **4.7 Summary of Findings**

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

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

⁹⁷⁴ **Appendix A**

⁹⁷⁵ **Complete Per-Class Summary**

⁹⁷⁶ **Statistics**

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

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

Continued on next page

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

Continued on next page

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

978 References

- 979 Anderson, S., Bankier, A., Barrell, B., Bruijn, M., Coulson, A., Drouin, J., ...
980 Young, I. (1981, 04). Sequence and organization of the human mitochondrial
981 genome. *Nature*, *290*, 457-465. doi: 10.1038/290457a0
982 Arango, G., Garner, E., Pruden, A., Heath, L., Vikesland, P., & Zhang, L. (2018,
983 02). Deeparg: A deep learning approach for predicting antibiotic resistance
984 genes from metagenomic data. *Microbiome*, *6*. doi: 10.1186/s40168-018
985 -0401-z
986 Bentley, D. R., Balasubramanian, S., Swerdlow, H. P., Smith, G. P., Milton, J.,
987 Brown, C. G., ... Smith, A. J. (2008). Accurate whole human genome
988 sequencing using reversible terminator chemistry. *Nature*, *456*(7218), 53–
989 59. doi: 10.1038/nature07517
990 Boore, J. L. (1999). Animal mitochondrial genomes. *Nucleic Acids Research*,
991 *27*(8), 1767–1780. doi: 10.1093/nar/27.8.1767
992 Cameron, S. L. (2014). Insect mitochondrial genomics: Implications for evolution
993 and phylogeny. *Annual Review of Entomology*, *59*, 95–117. doi: 10.1146/
994 annurev-ento-011613-162007
995 Dierckxsens, N., Mardulyn, P., & Smits, G. (2017). Novoplasty: de novo assembly
996 of organelle genomes from whole genome data. *Nucleic Acids Research*,

- 997 45(4), e18. doi: 10.1093/nar/gkw955
- 998 Edgar, R. C. (n.d.). *Uchime in practice*. Retrieved from https://www.drive5.com/usearch/manual7/uchime_practical.html
- 1000 Edgar, R. C. (2016). Uchime2: improved chimera prediction for amplicon sequencing. *bioRxiv*. Retrieved from <https://api.semanticscholar.org/CorpusID:88955007>
- 1003 Edgar, R. C., Haas, B. J., Clemente, J. C., Quince, C., & Knight, R. (2011).
1004 Uchime improves sensitivity and speed of chimera detection. *Bioinformatics*,
1005 27(16), 2194–2200. doi: 10.1093/bioinformatics/btr381
- 1006 Glenn, T. C. (2011). Field guide to next-generation dna sequencers. *Molecular
1007 Ecology Resources*, 11(5), 759–769. doi: 10.1111/j.1755-0998.2011.03024.x
- 1008 Gonzalez, J. M., Zimmermann, J., & Saiz-Jimenez, C. (2004, 09). Evaluating
1009 putative chimeric sequences from pcr-amplified products. *Bioinformatics*,
1010 21(3), 333-337. Retrieved from <https://doi.org/10.1093/bioinformatics/bti008> doi: 10.1093/bioinformatics/bti008
- 1012 Gray, M. W. (2012). Mitochondrial evolution. *Cold Spring Harbor perspectives
in biology*, 4. Retrieved from <https://doi.org/10.1101/cshperspect.a011403> doi: 10.1101/cshperspect.a011403
- 1015 Hahn, C., Bachmann, L., & Chevreux, B. (2013). Reconstructing mitochondrial
1016 genomes directly from genomic next-generation sequencing reads—a baiting
1017 and iterative mapping approach. *Nucleic Acids Research*, 41(13), e129. doi:
1018 10.1093/nar/gkt371
- 1019 Jin, J.-J., Yu, W.-B., Yang, J., Song, Y., dePamphilis, C. W., Yi, T.-S., & Li,
1020 D.-Z. (2020). Getorganelle: a fast and versatile toolkit for accurate de
1021 novo assembly of organelle genomes. *Genome Biology*, 21(1), 241. doi:
1022 10.1186/s13059-020-02154-5

- 1023 Judo, M. S. B., Wedel, W. R., & Wilson, B. H. (1998). Stimulation and sup-
1024 pression of pcr-mediated recombination. *Nucleic Acids Research*, *26*(7),
1025 1819–1825. doi: 10.1093/nar/26.7.1819
- 1026 Labrador, K., Agmata, A., Palermo, J. D., Ravago-Gotanco, R., & Pante, M. J.
1027 (2021). Mitochondrial dna reveals genetically structured haplogroups of
1028 bali sardinella (sardinella lemuru) in philippine waters. *Regional Studies in*
1029 *Marine Science*, *41*, 101588. doi: 10.1016/j.rsma.2020.101588
- 1030 Li, H. (2018, 05). Minimap2: pairwise alignment for nucleotide sequences. *Bioin-*
1031 *formatics*, *34*(18), 3094-3100. Retrieved from <https://doi.org/10.1093/bioinformatics/bty191>
1032 doi: 10.1093/bioinformatics/bty191
- 1033 Liang, Q., Bible, P. W., Liu, Y., Zou, B., & Wei, L. (2020, 02). Deepmi-
1034 crobes: taxonomic classification for metagenomics with deep learning. *NAR*
1035 *Genomics and Bioinformatics*, *2*(1), lqaa009. Retrieved from <https://doi.org/10.1093/nargab/lqaa009> doi: 10.1093/nargab/lqaa009
- 1036 Metzker, M. L. (2010). Sequencing technologies — the next generation. *Nature*
1037 *Reviews Genetics*, *11*(1), 31–46. doi: 10.1038/nrg2626
- 1038 Mysara, M., Saeys, Y., Leys, N., Raes, J., & Monsieurs, P. (2015). Catch,
1039 an ensemble classifier for chimera detection in 16s rrna sequencing stud-
1040 ies. *Applied and Environmental Microbiology*, *81*(5), 1573-1584. Retrieved
1041 from <https://journals.asm.org/doi/abs/10.1128/aem.02896-14> doi:
1042 10.1128/AEM.02896-14
- 1043 Peccoud, J., Lequime, S., Moltini-Conclois, I., Giraud, I., Lambrechts, L., &
1044 Gilbert, C. (2018, 04). A survey of virus recombination uncovers canon-
1045 ical features of artificial chimeras generated during deep sequencing li-
1046 brary preparation. *G3 Genes—Genomes—Genetics*, *8*(4), 1129-1138. Re-
1047 trieved from <https://doi.org/10.1534/g3.117.300468> doi: 10.1534/
- 1048

- 1049 g3.117.300468
- 1050 Qin, Y., Wu, L., Zhang, Q., Wen, C., Nostrand, J. D. V., Ning, D., ... Zhou, J.
1051 (2023). Effects of error, chimera, bias, and gc content on the accuracy of
1052 amplicon sequencing. *mSystems*, 8(6), e01025-23. Retrieved from <https://journals.asm.org/doi/abs/10.1128/msystems.01025-23> doi: 10.1128/
1053 msystems.01025-23
- 1054
- 1055 Qiu, X., Wu, L., Huang, H., McDonel, P. E., Palumbo, A. V., Tiedje, J. M., &
1056 Zhou, J. (2001). Evaluation of pcr-generated chimeras, mutations, and het-
1057 eroduplexes with 16s rrna gene-based cloning. *Applied and Environmental*
1058 *Microbiology*, 67(2), 880–887. doi: 10.1128/AEM.67.2.880-887.2001
- 1059 Ren, J., Song, K., Deng, C., Ahlgren, N., Fuhrman, J., Li, Y., ... Sun, F. (2020,
1060 01). Identifying viruses from metagenomic data using deep learning. *Quan-*
1061 *titative Biology*, 8. doi: 10.1007/s40484-019-0187-4
- 1062 Rodriguez-Martin, B., Palumbo, E., Marco-Sola, S., Griebel, T., Ribeca, P.,
1063 Alonso, G., ... Djebali, S. (2017, 01). Chimpipe: Accurate detection of
1064 fusion genes and transcription-induced chimeras from rna-seq data. *BMC*
1065 *Genomics*, 18. doi: 10.1186/s12864-016-3404-9
- 1066 Rognes, T., Flouri, T., Nichols, B., Quince, C., & Mahé, F. (2016). Vsearch: a
1067 versatile open source tool for metagenomics. *PeerJ*, 4, e2584. doi: 10.7717/
1068 peerj.2584
- 1069 Sedlazeck, F., Rescheneder, P., Smolka, M., Fang, H., Nattestad, M., von Haeseler,
1070 A., & Schatz, M. (2018, 06). Accurate detection of complex structural
1071 variations using single-molecule sequencing. *Nature Methods*, 15. doi: 10
1072 .1038/s41592-018-0001-7
- 1073 Sfeir, A., & Symington, L. S. (2015). Microhomology-mediated end joining: A
1074 back-up survival mechanism or dedicated pathway? *Trends in Biochemical*

- 1075 *Sciences*, 40(11), 701-714. Retrieved from <https://www.sciencedirect.com/science/article/pii/S0968000415001589> doi: <https://doi.org/10.1016/j.tibs.2015.08.006>
- 1076
- 1077
- 1078 Vervier, K., Mahé, P., Tournoud, M., Veyrieras, J.-B., & Vert, J.-P. (2015, 11). Large-scale machine learning for metagenomics sequence classification. *Bioinformatics*, 32(7), 1023-1032. Retrieved from <https://doi.org/10.1093/bioinformatics/btv683> doi: 10.1093/bioinformatics/btv683
- 1079
- 1080
- 1081
- 1082 Willette, D., Bognot, E., Mutia, M. T., & Santos, M. (2011). *Biology and ecology of sardines in the philippines: A review* (Vol. 13; Tech. Rep. No. 1). NFRDI
- 1083
- 1084 Technical Paper Series. Retrieved from <https://nfrdi.da.gov.ph/tpjf/etc/Willette%20et%20al.%20Sardines%20Review.pdf>
- 1085