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

Duran, Lin, Pailden

University of the Philippines Visayas

December 14, 2025

Outline

1. Introduction
2. Problem Statement & Proposed Solution
3. Objectives
4. Scope and Limitations
5. Methodology

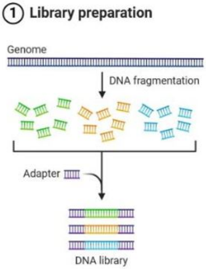
Real-World Problem

Next Generation Sequencing (NGS)

Source: University of the Philippines Visayas, 2022

Illumina Seq Workflow

Step 1. Library Preparation



Source: Microbe Notes, 2024

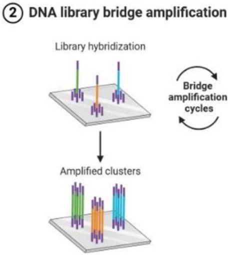
PGC
University of the Philippines Visayas
Philippine Genome Center Visayas
@PGCVisayas

Source: Philippine Genome Center Visayas, 2025

Real-World Problem

Illumina Seq Workflow

Step 2. Library Bridge Amplification (PCR)



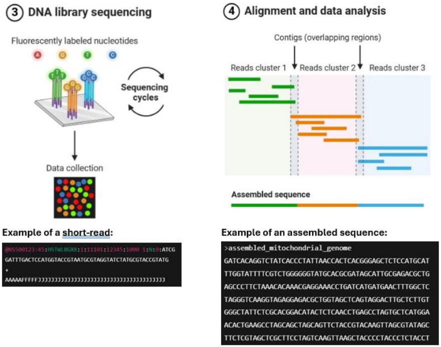
Source: Microbe Notes, 2024

PGC
PHILIPPINE GENOME CENTER VISAYAS
University of the Philippines Visayas
Philippine Genome Center Visayas

Source: Philippine Genome Center Visayas, 2025

Illumina Seq Workflow

Step 3. Sequencing and Alignment



Example of a short-read:

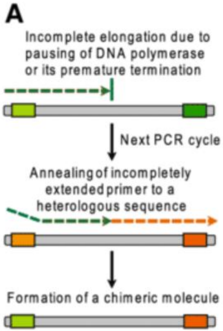
```
@NS500123:45:H5TWLBGXX:1:11101:12345:1000 1:N:0:ATCG
GATTTGACTCCATGGTACCGTAATGCGTAGGTATCTATGCGTACCGTATG
+
AAAAFFFFFJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJ
```

Example of an assembled sequence:

```
>assembled_mitochondrial_genome
GATCACAGGTCTATCACCCTATTAACCACTCACGGGAGCTCTCCATGCAT
TTGGTATTTTCGTCTGGGGGGTATGCACGCGATAGCATTGCGAGACGCTG
AGCCCTTCTAAACACAAACGAGGAAACCTGATCATGATGAACTTTGGCTC
TAGGGTCAAGGTAGAGGAGACGCTGGTAGCTCAGTAGGACTTGCTCTTGT
GGGCTATTCTCGCACGGACATACTCTCAACCTGAGCCTAGTGCTCATGGA
ACACTGAAGCCTAGCAGCTAGCAGTTCTACCGTACAAGTTAGCGTATAGC
TTCTCGTAGCTCGCTTCCTAGTCAAGTTAAGCTACCCCTACCCTCTACCT
```

Real-World Problem

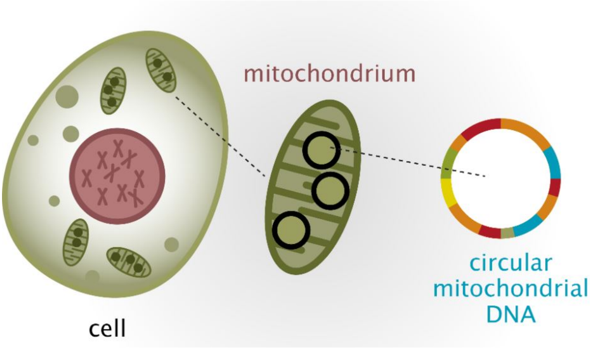
PCR-Chimera Formation


A
Incomplete elongation due to pausing of DNA polymerase or its premature termination
Next PCR cycle
Annealing of incompletely extended primer to a heterologous sequence
Formation of a chimeric molecule

Source: Omelina et al., 2024

Real-World Problem

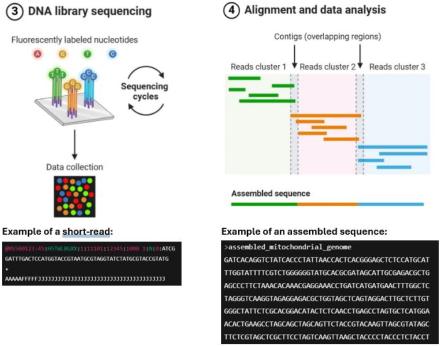
The Mitochondrial Genome



Source: UZ Bruzzel., 2020

Real-World Problem

Disrupts Genome Assembly



Example of a short-read:

```
@NS500123:45:H5TWLBGXX:1:11101:12345:1000 1:N:0:ATCG
GATTTGACTCCATGGTACCGTAATGCGTAGGTATCTATGCGTACCGTATG
+
AAAAFFFFFJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJ
```

Example of an assembled sequence:

```
>assembled_mitochondrial_genome
GATCACAGGTCTATCACCCTATTAACCACTCACGGGAGCTCTCCATGCAT
TTGGTATTTTCGTCTGGGGGGTATGCACGCGATAGCATTGCGAGACGCTG
AGCCCTTCTAAACACAAACGAGGAAACCTGATCATGATGAACTTTGGCTC
TAGGGTCAAGGTAGAGGAGACGCTGGTAGCTCAGTAGGACTTGCTCTTGT
GGGCTATTCTCGCACGGACATACTCTCAACCTGAGCCTAGTGCTCATGGA
ACACTGAAGCCTAGCAGCTAGCAGTTCTACCGTACAAGTTAGCGTATAGC
TTCTCGTAGCTCGCTTCCTAGTCAAGTTAAGCTACCCCTACCCTCTACCT
```

Existing Approaches

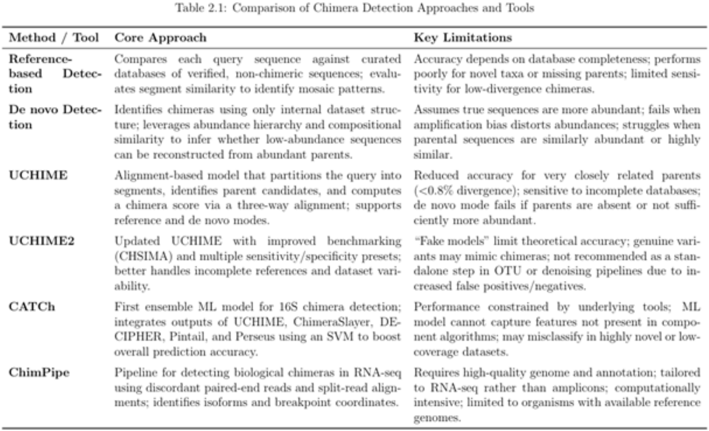


Table 2.1: Comparison of Chimera Detection Approaches and Tools

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.

Problem Statement & Proposed Solution

- **Problem Statement:** Chimeric sequencing reads can disrupt mitochondrial genome assembly, but current assembly pipelines assume artifact-free input and existing chimera detection tools are not designed specifically for organellar, particularly mitochondrial datasets, leaving assemblies vulnerable to undetected artifacts.
- **Proposed Solution:** A machine-learning pipeline designed to detect PCR-induced chimeric reads using both alignment-based and sequence-derived features to improve the quality and reliability of downstream mitochondrial genome assemblies.

General Objective

- Develop and evaluate a machine-learning pipeline (MitoChime) to detect PCR-induced chimeric reads in *S. lemuru* mitochondrial sequencing data to improve downstream assembly quality.

Specific Objectives

1. Construct simulated *Sardinella lemuru* Illumina paired-end datasets contain ing both clean and PCR-induced chimeric reads.
2. Extract alignment-based and sequence-based features such as k-mer composition, microhomology, and split-alignment counts from both clean and chimeric reads
3. Train, validate, and compare supervised machine learning models for classifying reads as clean or chimeric.
4. Determine feature importance and identify indicators of PCR-induced chimerism.
5. Integrate the optimized classifier into a modular and interpretable pipeline deployable on standard computing environments at PGC Visayas.

Scope of the Study

- Focuses on PCR-induced chimeric reads in *Sardinella lemuru* mitochondrial sequencing data to:
    - to limit interspecific variation in mitochondrial genome size, GC content, and repetitive regions so that differences in read patterns can be attributed more directly to PCR-induced chimerism
    - to align the analysis with relevant *S. lemuru* sequencing projects at PGC Visayas
    - to take advantage of the availability of *S. lemuru* mitochondrial assemblies and raw datasets in public repositories such as the National Center for Biotechnology Information (NCBI), which facilitates reference selection and benchmarking
    - to develop a tool that directly supports local studies on *S. lemuru* population structure and fisheries management produce tools applicable to local population and fisheries studies

Scope of the Study

- Uses wgsim-based simulations and selected empirical mitochondrial datasets
- Analysis targets low-dimensional alignment and sequence features (k-mers, GC content, clipping, split alignments) to maintain interpretability and computational accessibility
- Long-read platforms and other taxa are not included

Key Exclusions

- Naturally occurring chimeras
- NUMTs
- Large-scale nuclear genome rearrangements
- High-dimensional deep learning embeddings

Other Limitations

- No simulations with variable sequencing error rates
- No testing of alternative parameter settings (k-mer length, microhomology windows)
- Reliance on supervised machine learning may limit detection of novel/unknown chimeric patterns

Methodology

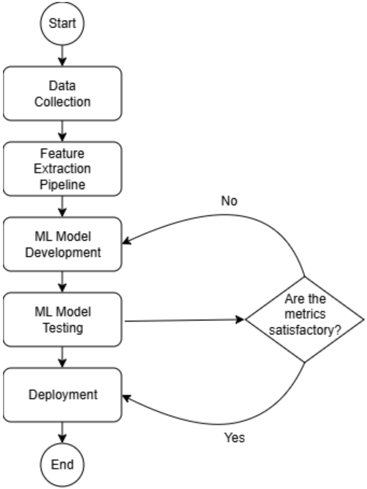


Figure: Process Diagram of the Special Project

Data Collection

The *S. lemuru* mitochondrial reference genome (NCBI: NC_039553.1) was downloaded in FASTA format and used as the basis for generating simulated reads.

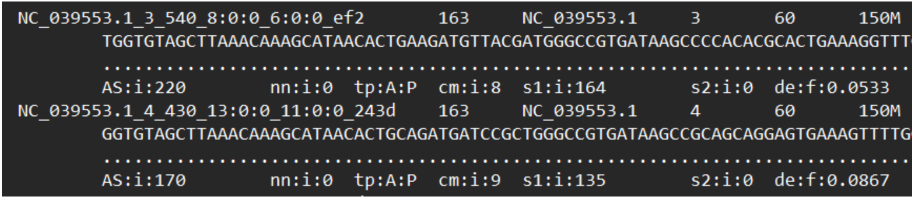
Data Preprocessing

- A Python script was used to generate the reads.
- Clean reads were produced with wgsim from the reference genome with error rate set to 5%.
- A chimeric reference was created by creating a custom script to combine non-adjacent segments with microhomology
- Chimeric reads were simulated with wgsim.
- All reads were mapped with minimap2 to extract alignment information.
- SAM/BAM files were converted, sorted, and indexed with samtools.

Data Preprocessing

- Final dataset: 40k reads, roughly balanced between clean and chimeric (19,984 clean reads and 20,000 chimeric).
- Some of the clean reads failed to align due to the set error rate.

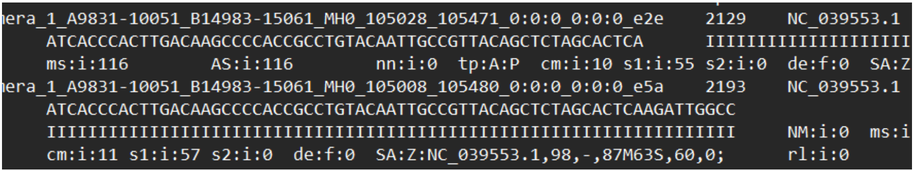
Data Preprocessing



```
NC_039553.1_3_540_8:0:0_6:0:0_ef2       163     NC_039553.1     3       60      150M
        TGGTGTAGCTTAAACAAAGCATAACACTGAAGATGTTACGATGGGCCGTGATAAGCCCCACACGCACTGAAGGTTT
        .............................................................................
        AS:i:220        nn:i:0  tp:A:P  cm:i:8  s1:i:164        s2:i:0  de:f:0.0533
NC_039553.1_4_430_13:0:0_11:0:0_243d    163     NC_039553.1     4       60      150M
        GGTGTAGCTTAAACAAAGCATAACACTGCAGATGATCCGCTGGGCCGTGATAAGCCGCAGCAGGAGTGAAAGTTTTG
        .............................................................................
        AS:i:170        nn:i:0  tp:A:P  cm:i:9  s1:i:135        s2:i:0  de:f:0.0867
```

Figure: SAM File of Clean Reads

Data Preprocessing



```
era_1_A9831-10051_B14983-15061_MH0_105028_105471_0:0:0_0:0:0_e2e    2129    NC_039553.1
    ATCACCCACTTGACAAGCCCCACCGCCTGTACAATTGCCGTTACAGCTCTAGCACTCA      IIIIIIIIIIIIIIIIIIII
    ms:i:116        AS:i:116        nn:i:0  tp:A:P  cm:i:10 s1:i:55 s2:i:0  de:f:0  SA:Z
era_1_A9831-10051_B14983-15061_MH0_105008_105480_0:0:0_0:0:0_e5a    2193    NC_039553.1
    ATCACCCACTTGACAAGCCCCACCGCCTGTACAATTGCCGTTACAGCTCTAGCACTCAAGATTGGCC
    IIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIII    NM:i:0  ms:i
    cm:i:11 s1:i:57 s2:i:0  de:f:0  SA:Z:NC_039553.1,98,-,87M63S,60,0;      rl:i:0
```

Figure: SAM File of Chimeric Reads

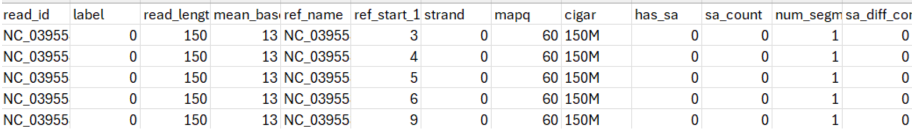
Feature Extraction Pipeline

- BAM files were processed with a Python script (`extract_features.py`) to build a TSV feature matrix.
- Used Pysam for parsing alignments and NumPy for computation.

Feature Extraction Pipeline

- Focused on three features linked to PCR-induced chimeras:
  1. **Supplementary Alignment (SA)**: Detects split alignments; counts and metrics extracted from SA tags
  2. **K-mer Composition Difference**: Breakpoints inferred; left/right segments compared using cosine and JS metrics.
  3. **Microhomology**: Overlap at junction quantified (length + GC content) within a defined window.
- Pipeline design and outputs to be validated by experts.

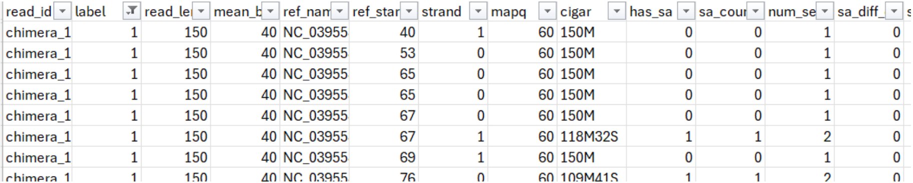
Feature Extraction Pipeline



read_id	label	read_lengt	mean_bas	ref_name	ref_start_1	strand	mapq	cigar	has_sa	sa_count	num_segm	sa_diff_co
NC_03955	0	150	13	NC_03955	3	0	60	150M	0	0	1	0
NC_03955	0	150	13	NC_03955	4	0	60	150M	0	0	1	0
NC_03955	0	150	13	NC_03955	5	0	60	150M	0	0	1	0
NC_03955	0	150	13	NC_03955	6	0	60	150M	0	0	1	0
NC_03955	0	150	13	NC_03955	9	0	60	150M	0	0	1	0

Figure: TSV Dataset showing Clean Reads

Feature Extraction Pipeline



read_id	label	read_le	mean_b	ref_nam	ref_star	strand	mapq	cigar	has_sa	sa_cour	num_se	sa_diff_	s
chimera_1	1	150	40	NC_03955	40	1	60	150M	0	0	1	0	
chimera_1	1	150	40	NC_03955	53	0	60	150M	0	0	1	0	
chimera_1	1	150	40	NC_03955	65	0	60	150M	0	0	1	0	
chimera_1	1	150	40	NC_03955	65	0	60	150M	0	0	1	0	
chimera_1	1	150	40	NC_03955	67	0	60	150M	0	0	1	0	
chimera_1	1	150	40	NC_03955	67	1	60	118M32S	1	1	2	0	
chimera_1	1	150	40	NC_03955	69	1	60	150M	0	0	1	0	
chimera_1	1	150	40	NC_03955	76	0	60	109M41S	1	1	2	0	

Figure: TSV Dataset showing Chimeric Reads

Dataset construction and split

- Simulated feature tables:
  - Clean reads (label 0)
  - PCR-induced chimeras (label 1)
- `build_datasets.py`:
  - Concatenate tables
  - Shuffle rows (avoid file-order artefacts)
- 80/20 **stratified** train–test split
- Test set held out and used **only once** at the end

Validation strategy

- Layer 1: 80/20 stratified train–test split
- Layer 2: 5-fold stratified cross-validation on training set
  - Train on 4 folds, validate on 1
  - Rotate so each fold is validation once
- Layer 3: Final evaluation on held-out test set
- Hyperparameter tuning:
  - `RandomizedSearchCV` inside CV for top models
- Goal: stable estimates and **unbiased** test performance

Model zoo and preprocessing pipeline

- **Baseline:** dummy majority-class classifier
- **Linear models:** logistic regression, calibrated linear SVM
- **Tree ensembles:**
  - Random Forest, Extra Trees
  - Gradient Boosting, XGBoost, LightGBM, CatBoost
- **Others:** bagging trees, k-NN, Gaussian NB, shallow MLP
- Common scikit-learn pipeline:
  - Median imputation (numeric missing values)
  - Standardisation (zero mean, unit variance)
- Ensures a **fair comparison** across models

Effect of hyperparameter tuning

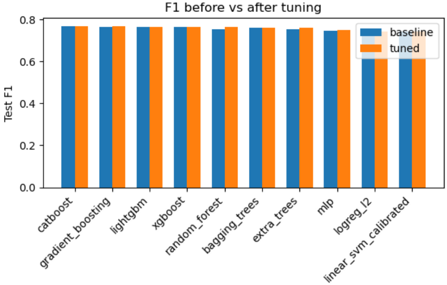


Figure: Test F1: baseline vs tuned.

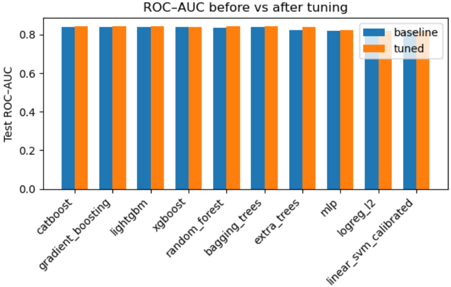


Figure: Test ROC–AUC: baseline vs tuned.

- Tuning done with `RandomizedSearchCV` on training set
- Small but consistent gains (ΔF1, ΔAUC $\approx$ 0.001–0.01)
- Top-ranked models remain the same (CatBoost, Gradient Boosting, LightGBM)

ROC and precision–recall curves

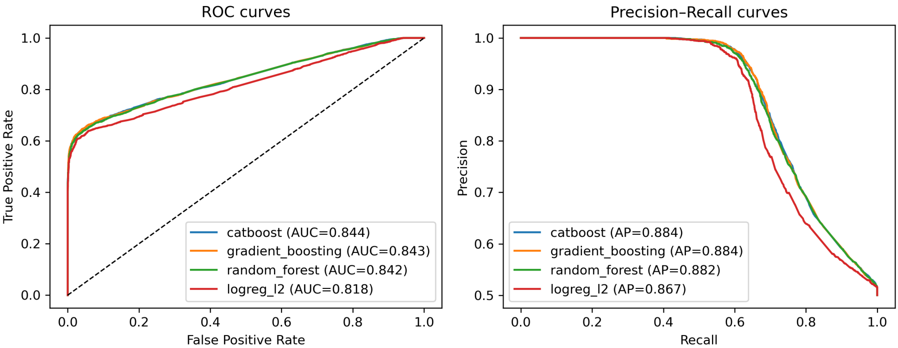


Figure: ROC (left) and PR (right) curves for CatBoost, Gradient Boosting, Random Forest, and logistic regression.

- Ensembles: ROC–AUC ≈ 0.84; logreg: ≈ 0.82
- Average precision ≈ 0.88 for ensembles
- Precision > 0.9 up to recall ≈ 0.5–0.6

Confusion matrix: CatBoost (test set)

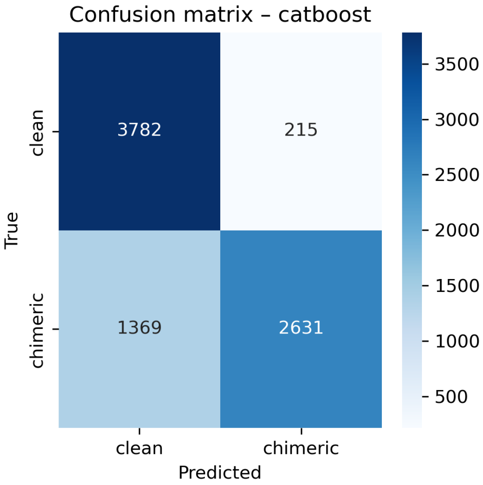
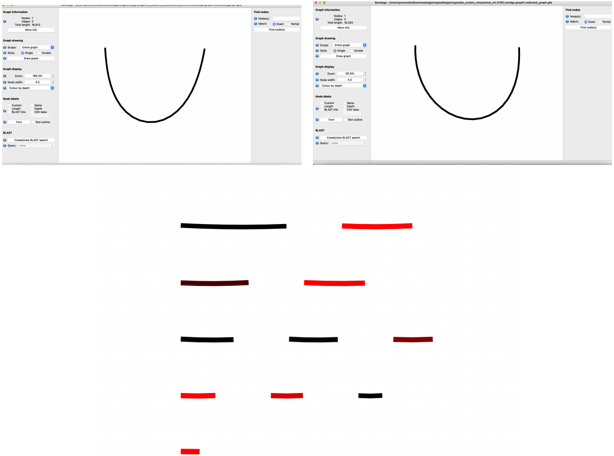


Figure: Confusion matrix heatmap for CatBoost.

- Clean reads (*negative class*):
  - Specificity ≈ 0.95 (TN = 3782 / 3997)
  - False positive rate ≈ 0.05 (FP = 215 / 3997)
- Chimeric reads (*positive class*):
  - Precision ≈ 0.92 (TP = 2631 / (2631 + 215))
  - Recall ≈ 0.66 (TP = 2631 / 4000)
- Behaviour at default threshold:
  - **Conservative chimera filter:** low FP, higher FN
  - Misses ∼34% of chimeras (FN = 1369 / 4000)

Downstream impact of residual chimeras (GetOrganelle)



Bandage - /Users/yvonnelin/Downloads/getorganelle/getorganelle_out/mt_mix/animal_mt.K105.contigs.graph1.selected_graph.gfa

Graph information
Nodes: 1
Edges: 0
Total length: 16,613

Graph drawing
Scope: Entire graph
Style: ● Single ○ Double

Graph display
Zoom: 186.3%
Node width: 4.5
Colour by depth

Node labels
□ Custom □ Name
□ Length □ Depth
□ BLAST hits □ CSV data:
□ Text outline

BLAST
Query: none

Find nodes
Node(s):
Match: ● Exact ○ Partial

Graph information
Nodes: 1
Edges: 0
Total length: 16,593

Graph drawing
Scope: Entire graph
Style: ● Single ○ Double

Graph display
Zoom: 191.8%
Node width: 4.5
Colour by depth

Node labels
□ Custom □ Name
□ Length □ Depth
□ BLAST hits □ CSV data:
□ Text outline

BLAST
Query: none

Find nodes
Node(s):
Match: ● Exact ○ Partial

Figure: Clean, 50% mixed, and chimera-only assemblies.

- **Clean:** 1 contig (16,613 bp)
- **Mixed 50%:** 1 contig (16,593 bp)
- **Chimera-only:** 11 contigs (∼39.7% mapped)
- **Implication:** missed chimeras can degrade assembly completeness/contiguity.

Top features for CatBoost

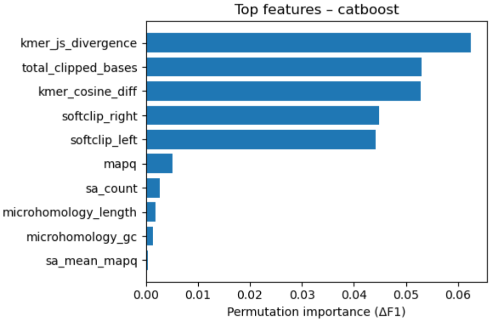


Figure: Permutation importance (ΔF1) for CatBoost.

- Strongest predictive signals:
  - `kmer_js_divergence` (within-read composition shift)
  - `total_clipped_bases` (junction-like clipping)
  - `kmer_cosine_diff` (windowed k-mer change)
- Also informative:
  - Left/right soft-clipping
  - Mapping quality (MAPQ)
  - SA count (supplementary alignments)
- Consistent with PCR chimera breakpoints and split mappings

Feature family importance

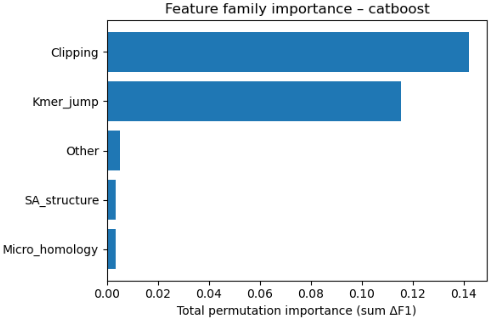


Figure: Aggregated permutation importance by feature family (CatBoost).

- Aggregated importance:
  - **Clipping** features dominate
  - **K-mer jump** features are also strong
- Smaller contributions:
  - SA structure
  - Micro-homology
  - Other alignment context
- Similar ranking observed for Gradient Boosting and Random Forest

Summary of findings

- Tree-based ensembles (CatBoost, Gradient Boosting, LightGBM) outperform linear baselines.
- Best performance on held-out reads:
  - F1 ≈ 0.77
  - ROC–AUC ≈ 0.84
- Most predictive signals match chimera junction patterns:
  - within-read k-mer composition shifts (*k-mer jump*)
  - extensive soft-clipping / clipped bases
- At the default threshold, CatBoost is **conservative**:
  - specificity ≈ 0.95 (keeps clean reads)
  - recall ≈ 0.66 (misses some chimeras)

Next steps

- **Error analysis:** characterize FP vs FN cases (focus on false negatives).
- **Calibration:** adjust threshold / use cost-sensitive objective to increase recall while controlling FP.
- **Biological validation:** compare assemblies before vs after filtering (contig count, length, coverage).
- **Exploratory extension:** sequence models (CNN / Transformer / RNN) for subtle breakpoint patterns.

Thank you!