Fine-Tuning Oxford Nanopore Basecalling Models to Improve Single-Read Variant Calling in Antibody Libraries

A thesis

submitted in partial fulfilment

of the requirements for the degree

of

Master of Science (Research) in Computing Science

at

The University of Waikato

by

Kang Zhou

THE UNIVERSITY OF
WAIKATO
Te Whare Wānanga o Waikato

2026

Abstract

Accurate nucleic acid sequencing is essential for analysing protein variant libraries composed of closely related sequences. Oxford Nanopore sequencing offers long reads, high depth, and comparatively lower cost, but its single-read error rate limits its routine use for variant calling, where sequencing errors can be mistaken for true mutations. This thesis investigates whether domain-specific fine-tuning of a nanopore basecalling model can improve single-read variant-calling performance in an antibody library context, using a dual-site IgA Fc deep mutational scanning library as an example.

A series of nanopore basecalling models was fine-tuned using in-house wild-type IgA Fc nanopore data collected under different experimental sample-prep procedures. These models were evaluated on a held-out, variant-rich nanopore dataset against an orthogonal MGI-derived short-read reference dataset. The benchmarking framework assessed both read-level quality and population-level recovery, focusing on library-constrained variant detection and variant-abundance concordance. Fine-tuned models consistently outperformed the pretrained baseline, demonstrating that nanopore signals contain learnable domain-specific structure that can improve single-read variant calling.

The best model, `Alpha`, achieved the best observed balance of low primary error rate, high read retention, low framework noise, and strong abundance-aware concordance with the ground truth. The results showed that the experimental procedure used to collect training data had a greater influence on the fine-tuning outcome than either training data volume or the stringency of quality control. Lower validation loss observed during training did not reliably predict better downstream performance, highlighting the need to evaluate basecalling models using biologically meaningful metrics rather than training diagnostics alone. Model-weight analysis further showed that successful fine-tuning involved small, structured recalibration of the signal-interface and decoding components rather than broad model-level drift.

Overall, this thesis establishes a practical framework for fine-tuning and evaluating domain-specific nanopore basecallers for variant libraries. It shows that domain-specific fine-tuning can materially improve the recovery of variant populations from nanopore signals and supports future work on computational augmentation of WT IgA Fc nanopore signals, architecture-aware fine-tuning, UMI-linked molecule-level benchmarking, experimental validation, predictive modelling, and extension to longer antibody constructs.

Acknowledgements

I wish to acknowledge the people to whom I owe my sincerest gratitude.

First, I would like to thank my chief supervisor, Dr William Kelton. It has been my honour to work with you over the past two years, from my summer research through to the completion of my master's degree. Your patience, dedication, trust, and enthusiasm for research have been a guiding light, enabling me to pursue my goals and continue my work without interruption.

I also wish to thank my co-supervisor, Dr Geoffrey Holmes, who provided valuable advice and insight into the machine learning aspects of the project.

Special thanks go to Dr Kyrin Hanning, who supported me from the beginning of my summer project in late 2024 through the entirety of my master's research, offering invaluable guidance on both the biological and computational dimensions of the project. Without the preliminary work you established, I might never have had the opportunity to undertake this project, which I have thoroughly enjoyed, from the bench work to the bioinformatics pipeline.

I would also like to extend my thanks to Kevin Beijerling, who helped me find my footing in the lab, from preparing chemicals and growing cells to even knowing where to find the right tubes and vials. I have been consistently impressed by the depth of your knowledge and your patience in fielding my relentless questions. I wish you the best with your PhD.

A special mention goes to Patrick Wightman and Megan Koschany for generously sharing buffers and consumables when time was short, and for the occasional discussions that offered fresh perspectives and challenged my thinking.

Lastly, I want to thank everyone in the C2 lab for their support throughout my master's programme, including Dr Geetanjali Rai and Dr Judith Burrows, who welcomed me into the lab and were always on hand for daily inquiries and equipment issues.

Disclosure

During the preparation of this thesis, the author used generative AI tools, including `Claude Opus 4.6` and `ChatGPT 5.4`. Their use was restricted to discovering primary literature sources; developing a figure-plotting Python notebook; refining language; improving clarity and conciseness; assisting with formatting and presenting source code alongside associated documentation already developed by the author; and identifying redundant methodological descriptions across chapters for consolidation.

These tools were not used to generate the underlying research question, experimental work, primary data, core computational pipeline/analysis, or conclusions. All substantive interpretations, methodological decisions, validation of technical content, and final written work submitted in this thesis were reviewed and revised by the author, who takes full responsibility for the accuracy and integrity of the work.

Table of Contents

List of Figures

List of Tables

List of Equations

Glossary

A_{450}: Absorbance measured at 450 nm

AUC: Area under a curve

`BAM`: Binary Alignment Map

Basecalling: The process of converting a nanopore signal into a nucleotide sequence

Beam search: A heuristic search algorithm used in the decoding of nanopore model output

`Bonito`: Open-source research basecaller and training framework provided by Oxford Nanopore Technologies

CCS: Composite concordance score

CDR: Complementarity-determining region

CFU: Colony-forming unit

`CIGAR`: Compact Idiosyncratic Gapped Alignment Report

CRF: Conditional random field

CTC: Connectionist temporal classification

DMS: Deep mutational scanning

`Dorado`: The production basecaller provided by Oxford Nanopore Technologies

ECDF: Empirical cumulative distribution function

ELISA: Enzyme-Linked Immunosorbent Assay

Epistasis: A protein interaction in which the effect of one mutation depends on another

Fab: Fragment antigen binding

`FASTQ`: NGS sequencing readout containing an ID, sequence and quality scores

Fc: Fragment crystallisable

FcαRI (CD89): The cognate IgA Fc receptor

FcRn: Neonatal Fc receptor

Golden Gate assembly: Type IIS restriction enzyme-based cloning method

GPU: Graphics processing unit

IgA: Immunoglobulin A

Indel: An insertion or deletion of one or more nucleotides in a genetic sequence

Jensen–Shannon similarity: A bounded measure of how similar two distributions are

k-mer: A sequence context of k consecutive bases

Logit: An unnormalised output score by a neural network's output layer

LSTM: Long short-term memory

MGI: MGI Tech

MinION Mk1B: Portable sequencing device by ONT

MOI: Multiplicity of infection

Nanobody: A single-domain antibody fragment derived from camelid antibodies

NGS: Next-generation sequencing

NNK codon: A degenerate codon scheme in which N can be any base, and K is G or T

OLS: Ordinary least squares

ONT: Oxford Nanopore Technologies

HiFi: Pacific Biosciences high-fidelity sequencing

PCR: Polymerase chain reaction

Phagemid: A hybrid vector combining a plasmid backbone with an M13 phage origin

Phage display: A protein display technology using bacteriophage as a carrier

pIII: The minor coat protein of M13 phage

`POD5`: signal file format used to store nanopore current traces

Q-score/Phred quality score: Log-scaled measure of basecalling confidence

RNN: Recurrent neural network

scFv: Single-chain variable fragment

`SAM`: Sequence Alignment Map

Softmax: A function that converts a vector of logits into a normalised probability distribution

Transformer encoder: Attention-based neural network module used to encode signal features

UMI: Unique molecular identifier

Variant calling: The process of determining sequence changes relative to a reference

Viterbi algorithm: A dynamic programming method for finding the global optimal path

Weighted Jaccard similarity: A count-weighted overlap metric that compares datasets

WT: Wild-type

Chapter 1. Introduction

1.1 Protein Engineering

Protein engineering is the deliberate modification of protein amino acid sequences to alter their structure, function, stability, or specificity (Steiner & Schwab, 2012). The ability of proteins to bind and interact in diverse contexts has led to the development of subfields within protein engineering, including antibody engineering, enzyme optimisation, mapping fitness landscapes, de novo design, and sequence generation.

Altering a protein sequence can influence several linked properties. Binding affinity, often summarised by the dissociation constant (K_d), determines how tightly a protein binds its ligand. In contrast, catalytic efficiency (k_{cat}/K_m) reflects how effectively an enzyme converts substrate to product. However, to catalyse reactions and bind effectively, the protein must also remain stable in vivo: denaturing temperature (T_m) and the Gibbs free energy change (ΔG) are standard metrics in this context, although they do not capture the full complexity of in vivo protein stability, which also depends on unfolding rate, protease resistance, and propensity to aggregate. Stability and function matter in most contexts; in some cases, such as therapeutic proteins designed for in vivo use, immunogenicity also matters. Because these properties often trade off against one another, improving one feature can compromise another, making protein engineering a multi-parametric optimisation problem.

Historically, rational design in protein engineering has relied on prior knowledge of sequence-structure-function relationships, using biochemical assay data and high-resolution 3D structures obtained from X-ray crystallography, cryogenic electron microscopy, and nuclear magnetic resonance, to identify key residues responsible for protein function (Steiner & Schwab, 2012). This enables targeted approaches such as site-directed mutagenesis. However, structural information is not always available, and even when it is, it may remain unclear which regions are most relevant to engineering. These limitations motivated the development of directed evolution (Sellés Vidal et al., 2023), which was recognised with the 2018 Nobel Prize. Directed evolution commonly involves two linked steps: library generation and variant screening. A classical experimental design combines error-prone PCR, which introduces mutations using low-fidelity polymerase under altered reaction conditions, with protein display systems for screening.

In protein display systems, each variant is physically linked to the genotype that encodes it, allowing enriched binders or functional variants to be recovered after selection (i.e., affinity maturation). Protein display is therefore distinct from library construction: the library defines the available sequence diversity, whereas the display platform determines how that diversity can be screened. Across formats, display systems are categorised into cell-based and cell-free approaches (Jaroszewicz et al., 2022).

George P. Smith first demonstrated phage display in 1985 by displaying a cloned peptide on filamentous phage (Smith, 1985). The technique involves packaging library DNA into bacteriophages (usually M13), with the target protein/peptide fused to a coat protein and displayed on the phage surface. Phage display has limited folding capacity; it is not appropriate for complex peptides that require eukaryotic folding or post-translational modification (PTM) (Boder & Wittrup, 1997). Additionally, phage display can enrich for sticky/fast-growing clones, leading to screening artefacts (Zade et al., 2017). However, phage display has the advantage of a massive library size, often ~10^9 – 10^{11} in practice (Bashir & Paeshuyse, 2020). Phage screening, also known as 'panning', is simple and scalable. Additionally, affinity maturation consumables are cheaper than those for other display technologies.

Alternative protein display technologies, such as yeast display and mammalian display, were developed in the late 1990s and are commonly used in therapeutic research (Boder & Wittrup, 1997; Zhou et al., 2010). These technologies are better suited for eukaryotic proteins that require proper folding and post-translational modification. Their library sizes (~10^5 – 10^7) are usually smaller than those achievable with phage display (Slavny et al., 2024). This explains why many protein engineering libraries use smaller fragments that are easier to fold into the correct structure, thereby increasing the number of variants available for screening by phage display.

Cell-free display technologies, such as ribosome display and mRNA display, overcome the transformation bottleneck and can therefore achieve enormous library sizes (~10^{12} – 10^{14}) (Schaffitzel et al., 1999; H. Wang & Liu, 2014). Because they are performed entirely in vitro, these approaches are also useful for proteins that are toxic to living cells. As with other display methods, the gain in library size must be weighed against practical considerations such as folding, stability, and downstream assay design (Schaffitzel et al., 1999; H. Wang & Liu, 2014).

Despite advances in protein engineering, fundamental challenges persist: the sequence space is astronomically large (i.e., it increases exponentially due to the combinatorial nature of sequence space), variant fitness is multi-parametric, and laboratory measurements are noisy. These difficulties make accurate sequencing and variant calling essential, because sequencing errors can easily be mistaken for biological signals.

1.2 Deep Mutational Scanning

More recently, the deep mutational scanning (DMS) approach has been adapted to generate constrained variant libraries by pairing with protein display technologies (Fowler & Fields, 2014). Deep mutational scanning is a high-throughput experimental framework for measuring the effects of very large numbers of mutations in parallel. In a typical DMS experiment, a protein variant library is generated, subjected to a functional screen, and then quantified by high-throughput sequencing before and after selection. Variant frequencies from those two populations are used to compute an enrichment score, *E(v),* for each variant, which serves as a proxy for functional effect. More precisely, the *E(v)* is computed as the log of the frequency/count of the variant in the post-screening library divided by the frequency/count of the variant in the pre-screening library:

$$E(v) = log \frac{f(v)_{post}}{f(v)_{pre}} \tag{1.1}$$

Conceptually, DMS pairs designed mutagenesis libraries with deep sequencing so that variant abundance can be quantified before and after screening (Fowler & Fields, 2014). It therefore provides a bridge between library design, screening, and read-level measurement, while still relying on careful control over mutational diversity and assay design. This matters because excessive or uneven diversity spreads sequencing depth across too many variants, reducing confidence in enrichment estimates (Adams et al., 2016; Fowler & Fields, 2014). Therefore, despite the effectiveness of error-prone PCR in directed evolution experiments, it remains unsuitable for DMS library construction (Rasila et al., 2009).

In the past decade, improved methods for designing antibody libraries for DMS, such as site-saturation mutagenesis and combinatorial mutagenesis, have been proposed. In site-saturation mutagenesis, a chosen codon in a gene is diversified to encode all possible amino acids. Codon schemes such as NNK are widely used because they reduce stop-codon frequency and

redundancy relative to NNN (Sullivan et al., 2013). When using site-saturation mutagenesis in DMS, it is common for researchers to generate antibody libraries with single-site saturation at every possible residue in the paratope, which can be pooled into a single library for screening, deep sequencing, and enrichment analysis (Adams et al., 2016; Fowler & Fields, 2014).

The main limitation of single-site DMS libraries is that they cannot capture epistatic interactions between mutations. Epistasis occurs when the effect of one mutation depends on the presence of another. Epistasis can be broadly grouped into three classes, differentiated by the effects of combinations of mutations: positive epistasis, where two mutations together outperform expectations; negative epistasis, where combined mutations reduce function; and sign epistasis, where the effect of a mutation is reversed in the presence of another mutation. The key idea is that the phenotype depends on a combination of mutations, not on any single mutation. This matters in antibody engineering because improved variants often contain multiple coordinated substitutions rather than a single dominant mutation. Indeed, DMS studies have shown that high-affinity variants can accumulate several mutations across the paratope (Adams et al., 2019).

To capture epistasis, combinatorial mutagenesis introduces mutations at multiple positions simultaneously, generating variant combinations that can be screened directly. This enables measuring not only the effects of individual substitutions but also how mutations interact in the same sequence background (Kirby & Whitehead, 2022; Püllmann et al., 2019). The number of sequence variants ($N_{variant}$) in a site-saturated combinatorial library can be estimated as the number of position combinations multiplied by the number of choices at the selected positions:

$$N_{variant} = 20^m \frac{n!}{(n-m)!\,(m)!} \tag{1.2}$$

Where n is the number of selected positions, and m is the total number of positions. The size of a combinatorial library scales exponentially for each additional simultaneous mutation and possible mutational position. Therefore, in practice, a site-saturated combinatorial library can easily exceed the display limit of existing protein display technologies, and a good library design must be carefully formulated to ensure that the selected protein display technology encompasses uniformly distributed diversity with constraints (Hanning et al., 2022).

1.3 Variant Calling by Deep Sequencing

Within a deep mutational scanning workflow, accurately detecting 'true' variants is a key step. This step is known as 'variant calling': the computational procedure of determining mutations present in sequencing reads relative to a reference protein sequence. In DMS experiments, variant calling assigns each sequenced read to a specific protein variant, allowing accurate variant counting before and after selection and the subsequent calculation of enrichment scores (Fowler & Fields, 2014; Hanning et al., 2022; Wei & Li, 2023).

Despite the availability of many high-throughput NGS technologies (Slatko et al., 2018), each technology comes with its own trade-offs. Some offer exceptional accuracy but short read lengths. In contrast, others can sequence long molecules but lack sufficient single-read accuracy to call variants reliably or have limited depth for target applications.

Illumina sequencing is a widely used high-throughput sequencing technology that uses sequencing-by-synthesis with fluorescent nucleotides. DNA fragments attach to a flow cell, amplify into clusters, and each base incorporation is detected by fluorescence imaging (Bentley et al., 2008; Chen et al., 2013). A typical Illumina sequencing kit supports up to 2×300 bp paired-end reads with accuracy above Q30[1], and more recent chemistries can extend this read length to 2×500 bp. In a paired-end sequencing setup, fluorescent nucleotides are added to both the forward and reverse reads, thereby improving accuracy in overlapping results (Bloom, 2014; Tan et al., 2019). In general, Illumina performs exceptionally well in substitution-based variant detection. However, the technology's read-length limit makes it unsuitable for protein libraries with sequences longer than the paired-end read limit.

PacBio sequencing is another high-throughput platform that produces long reads, typically 10–25 kb (Hon et al., 2020). It uses single-molecule real-time sequencing inside tiny wells called zero-mode waveguides. DNA molecules are sequenced repeatedly in circular consensus mode and then collapsed into a single, highly accurate HiFi read. PacBio HiFi can achieve short-read-like accuracy, often reported at >99.5% and in some contexts around 99.8-99.9%[2] (B. Wang et al., 2025; Wenger et al., 2019). However, the main disadvantages are the cost and the

[1] For Illumina, Q30 denotes an estimated per-base error probability of 10^{-3}
[2] For PacBio HiFi, the accuracy typically reflects consensus accuracy after repeated passes.

lower throughput. This limits the usefulness of PacBio for very large DMS libraries, in which sequencing depth directly affects variant-frequency estimates (Rhoads & Au, 2015).

Barcode-based DMS has been developed to combat the sequencing depth issue with DMS. Instead of sequencing the full antibody variant in every screening round, a pool of short random DNA barcodes (15–20 nt) is attached to clones/variants before screening (Jann et al., 2026; Matreyek et al., 2018). A one-time mapping step, for example, using PacBio, is then used to associate each barcode with the full variant sequence (Çubuk et al., 2025). For subsequent rounds, only the barcode region needs to be sequenced, allowing variant frequencies to be inferred from barcode counts (using Illumina short-read sequencing). Barcode-linked DMS approaches are widely used to reduce sequencing cost and improve scalability, although they are described under different names across the literature (Weile et al., 2017). Their main limitation is that an initial barcode-to-variant map requires physical co-localisation of the barcode with the full coding sequence, which can increase construct and sequencing length requirements. In addition, barcode maps are most naturally suited to static or fixed libraries; in directed evolution, newly generated variants generally require re-barcoding or remapping (Johnson et al., 2023; Xie et al., 2018). Repeated PCR amplification of the barcode after each screening round introduces noise and can contribute to barcode-genotype unlinking over time (Faure et al., 2020).

1.4 Nanopore Sequencing

Oxford Nanopore Technologies (ONT) occupies an attractive middle ground between PacBio and Illumina. Sequencing is performed on a flow cell, while a reusable device records changes in ionic current as nucleic acids pass through membrane-embedded nanopores. A MinION flow cell contains 512 measurement channels, each of which is associated with multiple pores. The recorded ionic current needs to be translated computationally into a nucleotide sequence (Y. Wang et al., 2021; Zhang et al., 2024). ONT supports read lengths from short amplicons to ultra-long reads spanning megabases, while per-read accuracy depends strongly on pore chemistry, library prep, and basecalling model. The latest R10.4.1 chemistry can achieve very high read accuracy (90–99%)[3], but the error profile remains context-dependent (Sereika et al., 2022).

[3] For ONT, read accuracy here refers to single-read sequence accuracy (no consensus) after basecalling.

Many common ONT applications, including genome assembly, haplotype resolution, and targeted amplicon analysis, rely on read consensus to reduce the effects of per-read error (Sereika et al., 2022). This differs from classical DMS experiments, in which each molecule may correspond to a distinct variant that must be called individually. In practice, nanopore reads typically contain many indel and homopolymer errors, so quality control usually involves extensive length-based and quality-based filtering, as well as careful error correction, to avoid inflating apparent variant counts (Delahaye & Nicolas, 2021; Liu-Wei et al., 2024).

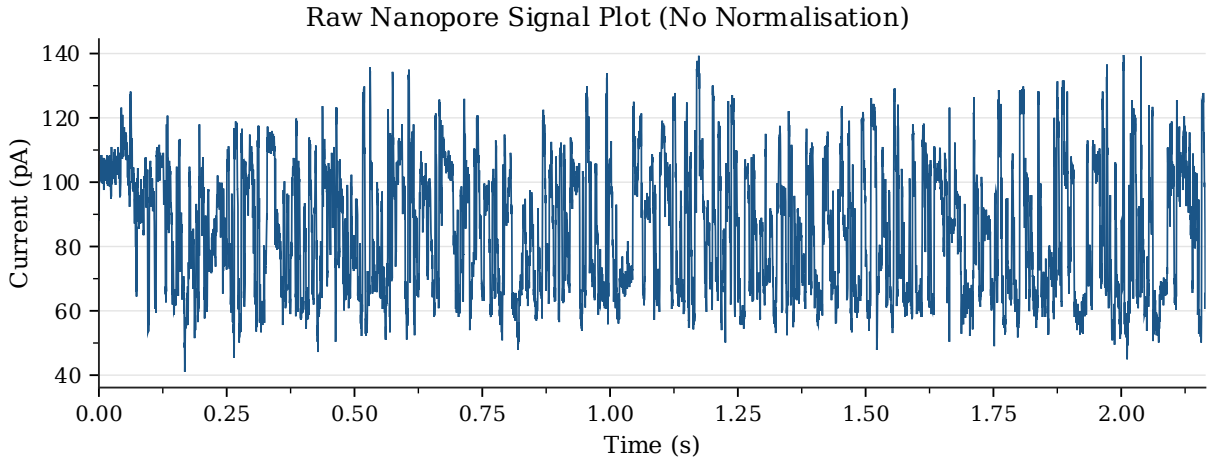

Figure 1. Representative raw nanopore ionic current trace for a WT IgA Fc amplicon. *Example raw electrical signal recorded for a single WT IgA Fc amplicon read on an Oxford Nanopore Technologies MinION Mk1B device using R10.4.1 flow-cell chemistry at a constant sampling rate of 5,000 Hz. The x-axis shows elapsed time (seconds), and the y-axis shows pore current (pA - picoampere). The stable current plateau from 0.00 to approximately 0.05 s corresponds to the open-pore baseline prior to strand capture. The fluctuating signal thereafter represents the translocation of the DNA strand through the nanopore at approximately 400 bases per second. A total of 10,825 current samples were recorded for this read; basecalling with the* `dna_r10.4.1_e8.2_400bps_sup@v5.2.0` *model recovered an 851-nt sequence, including the nanopore adapter. No current normalisation was applied prior to plotting. Plotted using* `Matplotlib` *(Hunter, 2007).*

An important strategy relevant to this project is UMI-based DMS (Deng et al., 2024; Doud & Bloom, 2016; Karst et al., 2021; Zurek et al., 2020). This involves incorporating unique molecular identifiers (UMIs), typically 8-16 ambiguous nucleotides, introduced during library preparation, often as dual UMIs in forward and reverse primers. PCR amplification is then used to generate multiple copies of each tagged molecule, and reads that share the same UMI pair are grouped to generate single-molecule consensus sequences. This substantially reduces sequencing error and can make ONT a viable approach for DMS. The trade-off is increased library-preparation complexity and a loss of effective depth, both of which matter for large libraries, especially in the pre-screened library. UMI diversity must also be carefully tuned: too little leads to collisions, whereas too much limits consensus formation (Andersson et al., 2024; Clement et al., 2018). In addition, PCR amplification is inherently biased (Aird et al., 2011), so poorly amplified molecules may drop out during quality control; certain sequence contexts

(GC-rich, secondary structure) form smaller families and are filtered out more readily; and some variants may be overamplified relative to others (Mamedov et al., 2008). As a result, the observed variant frequency can drift away from the true library frequency.

Most sequencing platforms achieve their best reported accuracy by leveraging redundancy and consensus. In contrast, this thesis addresses the more difficult problem of single-read variant calling in a constrained DMS library, where each read may represent a unique molecule and consensus-based correction can reduce effective depth or obscure genuine low-abundance variants.

1.5 Basecalling

Basecalling is the step that converts the raw electrical signal measured as nucleic acid moves through a pore into a nucleotide sequence and an associated quality string, which is saved in `FASTQ` or `BAM` format (Y. Wang et al., 2021). The raw signal is typically stored in `POD5` format as a time series of current measurements in pA, and `POD5` is a custom container that stores a series of Apache tables (e.g., metadata, table schema, and raw data). As with other sequencing technologies, higher base qualities are intended to indicate a lower expected error rate; in nanopore sequencing, however, those qualities are derived from a learned decoding model and therefore reflect both model internal confidence and calibration.

This distinction matters for the present thesis because per-base quality is not the same thing as downstream usefulness. A model may assign high confidence to a sequence that is still biologically implausible for a constrained mutational library, while a better-calibrated model may sacrifice some nominal quality to preserve the correct variant structure. For that reason, the model performance in this thesis is considered both a signal-decoding problem and a variant-calling problem.

Nanopore signals are time series with unique properties that make the decoding process challenging. At constant temperature and under fixed chemistry conditions, a motor protein ‘ratchets’ the strand through at a typical speed set by the enzyme. However, at the single-molecule level, motor-protein stepping is stochastic, producing variable dwell times, so the speed is not uniform along the read. In practice, the molecule moves through the pore with stochastic fluctuations that are non-uniform. For this reason, there is no clean alignment between the number of samples and the number of bases. In addition, the influence on the

current is determined by multiple DNA bases at once (k-mer context; 9-mer in R10.4 chemistry) (Samarakoon et al., 2025). Moreover, the baseline current varies across runs (current normalisation is incorporated into the latest basecalling model prior to feature extraction).

In fact, the nature of the nanopore signal decoding problem is analogous to speech recognition: it consists of mapping a variable-length signal to a variable-length sequence using a sequence encoder. ONT explicitly uses neural network architectures, including RNNs (recurrent neural networks) and transformers, in its latest basecaller, `Dorado` (Wan et al., 2022).

The development of basecalling software accelerated around 2016. Early basecallers commonly relied on event detection and probabilistic models, including hidden Markov model (HMM)-based approaches. `Nanocall` is the first freely available, open-source HMM-based basecaller for Oxford Nanopore sequence data using R7.3 chemistry (David et al., 2017). Neural-network basecalling emerged soon after, with `DeepNano` representing an early recurrent neural network (RNN)-based approach (Boža et al., 2017). Later, around 2018, `Chiron` became one of the first widely used end-to-end raw signal basecallers to apply connectionist temporal classification (CTC), an algorithm that enables direct mapping from unsegmented signal to nucleotide sequence without explicit event segmentation (Teng et al., 2018). Around 2019, neural network-based basecallers had mostly replaced HMM/event pipelines, greatly enhancing accuracy. Substantial research occurred during this period, focusing on neural network-based basecaller architectures (`Albacore/Guppy/Scrappie/Flappie` and others)(Wick et al., 2019). `Flappie` and `Guppy` are notable for their ‘flip-flop’ models, which improved homopolymer handling but often come at the cost of speed. Later in 2020, ONT introduced the research basecaller `Bonito`, which includes a conditional random field (CRF)-based decoder in place of a standard CTC decoder (Pagès-Gallego & de Ridder, 2023). This CTC-CRF-style architecture allowed the decoder to model dependencies between neighbouring output states, thereby improving transition modelling and alignment resolution. The latest architecture improvements come from incorporating a transformer encoder into the `Dorado sup` model, together with CTC-CRF decoding, improving contextual modelling even further at substantially increased computational cost, often requiring a graphics processing unit (GPU) for parallel signal processing.

The latest `Dorado` basecaller toolkit offers three model categories with different architectures, as shown in Figure 2. For the remainder of this review, the `Dorado` models discussed below are based on the v5.2.0 architecture. To our knowledge, Oxford Nanopore did not publicly describe `Bonito` or `Dorado` in a dedicated peer-reviewed paper; instead, both were primarily released as open-source software with publicly available model weights. Therefore, exact layer dimensions and number of parameters for `Dorado` v5.2.0 models were computed directly from the released model architecture files rather than taken from a peer-reviewed publication. The wider architectural tendencies, however, are consistent with published benchmarking work showing the importance of the RNN encoder, the Transformer encoder, and the CRF head in high-performing nanopore basecallers (Pagès-Gallego & de Ridder, 2023).

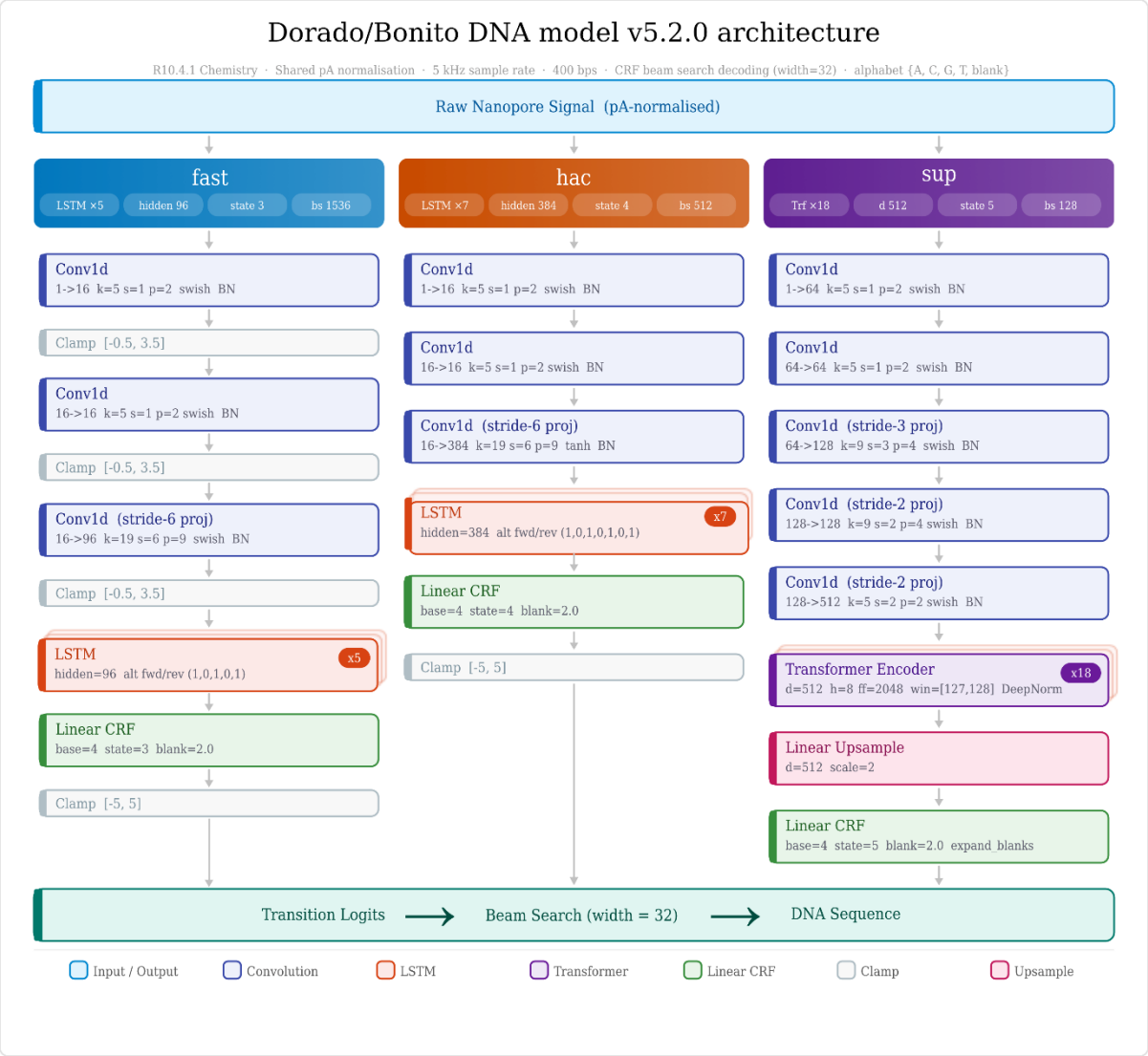

Figure 2. Comparison of `Dorado` model families. *Architecture diagram comparing the `Dorado` v5.2.0 `fast`, `hac`, and `sup` DNA basecalling models. From top to bottom, the schematic shows a shared pA-normalised raw-signal input; model-specific Conv1d stacks; recurrent LSTM blocks in `fast` and `hac`; a transformer encoder and a linear upsample block in `sup`; and a linear CRF decoding head in all three branches.*

All three models (`fast`, `hac`, `sup`) perform the same task: converting a 1D nanopore current trace into a nucleotide sequence. In each case, the raw signal is first normalised and then passed through a stack of Conv1D layers. These front-ends serve two purposes. First, it detects short local signal patterns. Second, it compresses the time axis into a shorter sequence of higher-level features. This step is necessary because nanopore signals are long, noisy, and only indirectly related to nucleotide identity. At any moment, the measured current reflects the combined influence of several neighbouring bases in the pore (i.e., 9-mer in R10.4.1 chemistry), and the number of signal samples associated with a given base varies because strand translation is not uniform. The convolutional layers, therefore, produce a more compact representation that the downstream sequence model can interpret.

The `fast` and `hac` models share the same overall Conv1D-LSTM-CRF design and differ mainly in scale. `Fast` is the smallest model, with 427,984 trainable parameters. Its convolutional feature extractor consists of two Conv1D layers that expand the signal from 1 to 16 channels, followed by a third Conv1D layer that expands to 96 channels while downsampling with a stride of 6. This feature sequence is then processed by five long short-term memory (LSTM) layers with alternating direction, providing bidirectional context (i.e., BiLSTM). The `hac` model followed the same overall design but is much wider and deeper, with 8,791,600 trainable parameters. Its main differences are a wider third convolution layer, seven alternating-direction LSTM layers, and a slightly larger CRF state space.

In both `fast` and `hac`, the recurrent encoder is important because the same nucleotide can produce different current patterns depending on its neighbours, and because variable dwell times mean that the mapping between signal samples and bases is not fixed. The LSTM layers integrate information over time, so that each output timestep reflects both the local signal shape and long-range sequence context. This is particularly important in nanopore basecalling, where signal interpretation depends not only on the current sample but also on what came before and what follows in the read.

In contrast, the `sup` model replaces the recurrent encoder with a deeper transformer encoder and is substantially larger, containing 78,719,936 trainable parameters. Its front end consists of five Conv1D layers rather than three, with later layers that both increase the channel dimension to 512 and aggressively compress the time axis. Eighteen transformer encoder blocks then process the compressed feature sequence. These blocks use multi-head self-attention, an output projection, a SwiGLU feed-forward network, RMSNorm, and residual connections. Unlike the

step-by-step recurrence used by LSTMs, self-attention allows each position in the feature sequence to draw information directly from other positions within the attention window. This is useful in basecalling because distinguishing homopolymers, indels, and irregular translocation events often requires evidence distributed across longer stretches of signal. In practice, the `sup` model uses local or windowed attention rather than full global attention, which would be computationally intractable for long nanopore signals. Therefore, prior convolutional downsampling plays an important role in making the transformer encoder tractable. After the transformer stack, a LinearUpSample layer restores temporal resolution before the final decoding step.

All three `Dorado` models end with a CRF head and therefore use the same decoding logic. At each encoder timestep, the encoder feature vector is projected to a set of transition scores over the CRF state graph. These scores represent the possible state transition from that timestep, conditioned on the current encoder representation.

Let n_{base} denote the number of canonical nucleotide symbols, and let the additional blank transition be included explicitly. For a CRF with state length s, the number of output scores per timestep is:

$$|V_l| = (n_{base} + 1)n_{base}^{s} \tag{1.3}$$

where n_{base}^{s} is the number of possible context states of length s, and $n_{base} + 1$ corresponds to the possible outgoing transition from each state, including the blank transition.

These transition scores are logits, not probabilities. They are produced by a linear projection of the encoder feature vector by the CRF head:

$$l = Wx \tag{1.4}$$

where x is the encoder feature vector (vector output from transformer encoder or LSTM encoder), W is the learned weights, and l is the transition scores. Since no activation function is applied to this output, the range of logit values is not restricted ($-\infty$, $+\infty$).

For example, the `Dorado` `sup` model uses a CRF state length of 5. This gives:

$$(4 + 1) \times 4^5 = 5120$$

output logits per timestep. Each logit is an unnormalised score for a particular nucleotide transition: either emitting nucleotide or taking the blank transition, conditioned on a state defined by the preceding 5-nucleotide context. (e.g., whether it makes more sense to transition from AACTT to ACTTG or AACTT to ACTTT).

During decoding, `Dorado` uses beam search to explore candidate paths through the CRF state graph whilst maintaining a fixed number of the highest-scoring partial paths, determined by the beam width (32 is the default in the `Dorado` decoding pipeline). The score of a complete path is the sum of its transition scores across timesteps:

$$Score(path) = \sum_{t} Score(t, t+1) \tag{1.5}$$

The decoded sequence corresponds to the path through the CRF state graph that maximises the cumulative score. However, because beam search is a heuristic approximation, the globally optimal path may be missed if it is pruned during search. Exact decoding in related-sequence models is often formulated by dynamic programming methods such as the Viterbi algorithm. It is possible to estimate the probability of each path using the softmax function by computing a partition function *Z* that sums the scores of all paths.

$$P(path) = \frac{e^{Score(path)}}{Z}, Z = \sum_{y \in allpaths} e^{Score(path_y)} \tag{1.6}$$

1.6 The dual-site IgA Fc DMS Library: An Example System

Antibody engineering provides a useful example of the single-read variant-calling problem addressed in this thesis. Engineered antibodies now constitute a major therapeutic platform, with over 150 clinically approved drugs for the treatment of cancer, autoimmune disease, and infectious disease. However, most development efforts have focused on variable domains and IgG-like scaffolds. IgA is interesting because it engages FcαRI on myeloid cells, thereby offering a distinct effector profile with therapeutic potential, particularly in settings where neutrophil recruitment is desirable (Ben Mkaddem et al., 2013; Boross et al., 2013; Kaplon et al., 2020). IgA is also abundant at mucosal surfaces, where early interception of pathogens is biologically important, yet it remains underdeveloped as a therapeutic class (Gao et al., 2024;

Macpherson et al., 2008). At the same time, native IgA has limitations such as a shorter serum half-life and incompletely optimised receptor-mediated activity, making IgA engineering a practical route for improving receptor engagement and functional performance (Kaplon et al., 2020; Meyer et al., 2016).

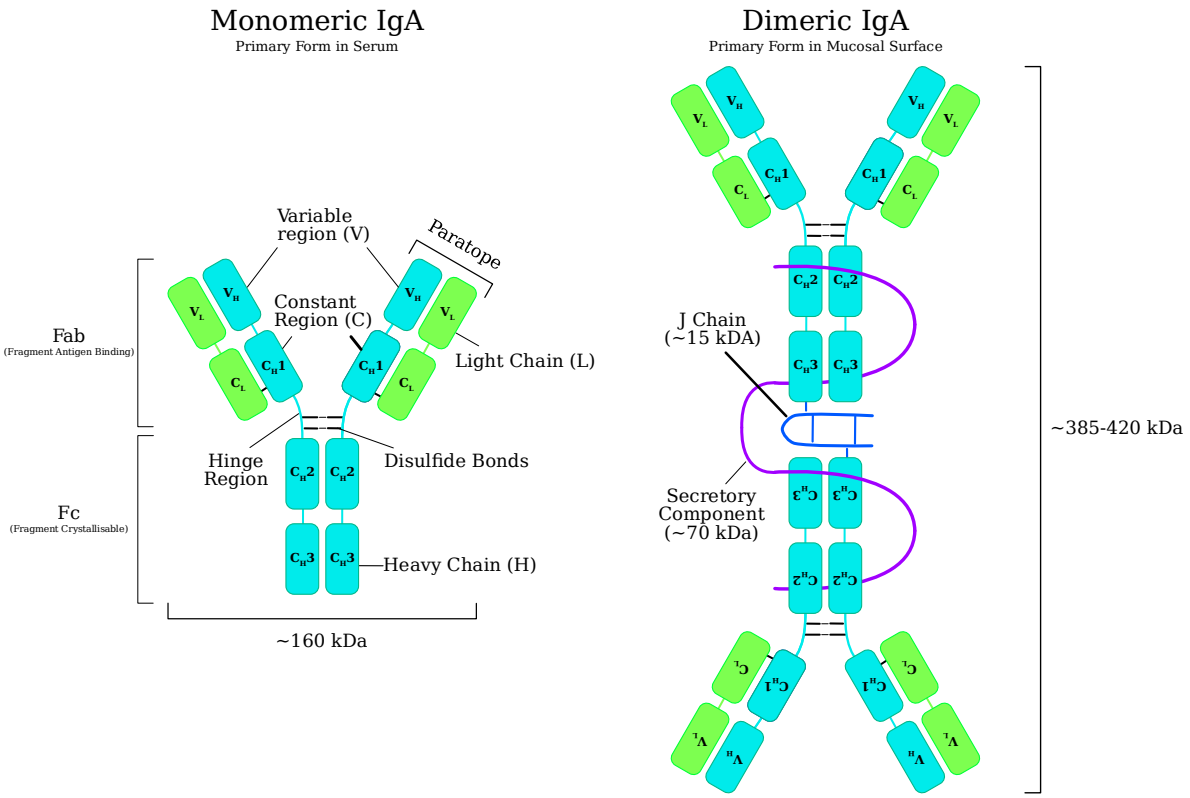

Figure 3. Structural organisation of the IgA antibody isotype. *Schematic representation of the two principal IgA forms found in humans: monomeric serum IgA and dimeric secretory IgA (sIgA). Monomeric IgA consists of two α-heavy chains and two light chains, structurally analogous to IgG. Dimeric sIgA comprises two IgA monomers joined by a J chain; during transcytosis across mucosal epithelial cells, the polymeric immunoglobulin receptor (pIgR) is cleaved, yielding the secretory component (SC), which stabilises the dimer and confers protease resistance at mucosal surfaces. Adapted from (de Sousa-Pereira & Woof, 2019).*

In this thesis, the example system is a dual-site IgA Fc DMS library developed by Dr Kyrin Hanning at the University of Waikato using oligo-based Golden Gate assembly (Hanning et al., 2025). Fc (fragment crystallisable) fragments refer to the ‘stem’ of the antibody and are composed entirely of heavy-chain constant domains (C_{H2} and C_{H3}). This provides a simple foundation for comparison and benchmarking fine-tuned nanopore basecalling models while avoiding the additional complexity of full antibody assembly. The dual-site DMS library was constructed by oligo-based Golden Gate assembly. During assembly, instead of relying on uncontrolled, error-prone mutagenesis, the library uses defined mutational parts, degenerate codons, and modular assembly, allowing molecules to be WT, single mutants (1-codon

mutations), or dual mutants (2-codon mutations) across predetermined regions of the IgA Fc sequence.

Golden Gate Assembly is well-suited to this design because Type IIS restriction enzymes cut outside their recognition sites, allowing ordered and scarless assembly of multiple parts in a single reaction (Bird et al., 2022; Engler et al., 2009). In the IgA Fc library, this enables the construction of defined mutation regions from paired oligonucleotide pools rather than synthesising each full double-stranded variant individually. The Fc sequence is partitioned into DMS regions and conserved framework regions, each carried by modular parts that assemble in a fixed order. This preserves clear constraints about where mutations are allowed and makes the resulting library especially suitable for downstream benchmarking.

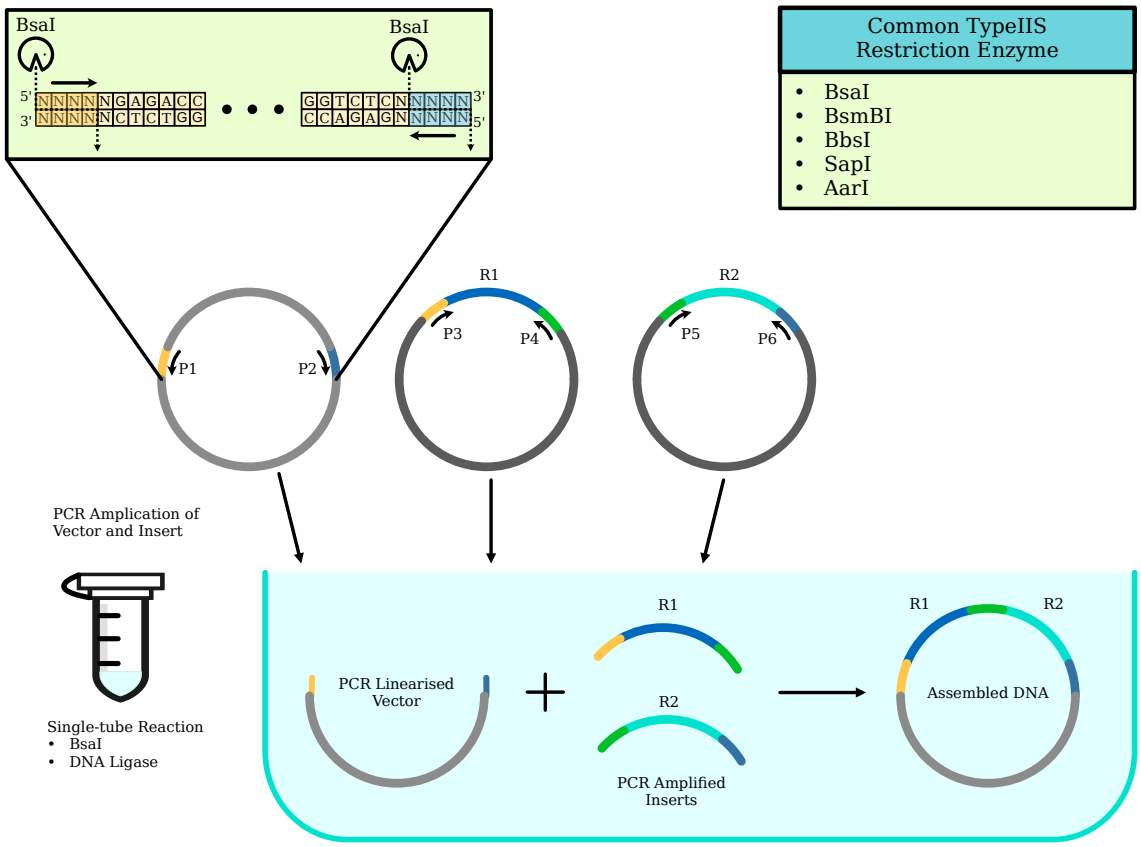

Figure 4. Principle of Golden Gate Assembly with two inserts. *Schematic illustrating the mechanism of Type IIS restriction enzyme-based Golden Gate cloning. Type IIS restriction enzymes recognise a defined sequence but cleave at a specified downstream offset from the recognition site, removing the recognition site from the ligation product. This design ensures that successfully ligated products cannot be recut and allows adjacent DNA fragments to be joined by unique four-nucleotide overhangs defined during construct design. Each insert requires two compatible overhangs, one per strand, to direct ordered, scar-free assembly into the destination vector. Adapted from (Bird et al., 2022).*

The WT Fc coding sequence spans 687 bp before the addition of sequencing adapters, and the predetermined DMS regions are separated across molecules. Preserving linkage between these

regions matters because the library contains both single- and dual-mutation variants, so sequencing must recover not just local substitutions but the full coverage of all DMS regions in a molecule. In practice, this exposes the thesis's central trade-off. PacBio HiFi can deliver excellent consensus accuracy, but its lower throughput and higher cost are less attractive when variant-frequency estimates depend on sequencing many molecules across a large variant library (Rhoads & Au, 2015; B. Wang et al., 2025; Wenger et al., 2019). Illumina paired-end sequencing is highly accurate, but the IgA Fc amplicon already pushes the comfortable overlap (<600 bp with a 2 x 300 kit), increasing sensitivity to end-of-read quality loss, especially in the reverse read (Bentley et al., 2008; Tan et al., 2019). The IgA Fc library, therefore, provides exactly the kind of example system in which nanopore sequencing becomes attractive, provided that per-read errors can be reduced enough for reliable single-read variant calling.

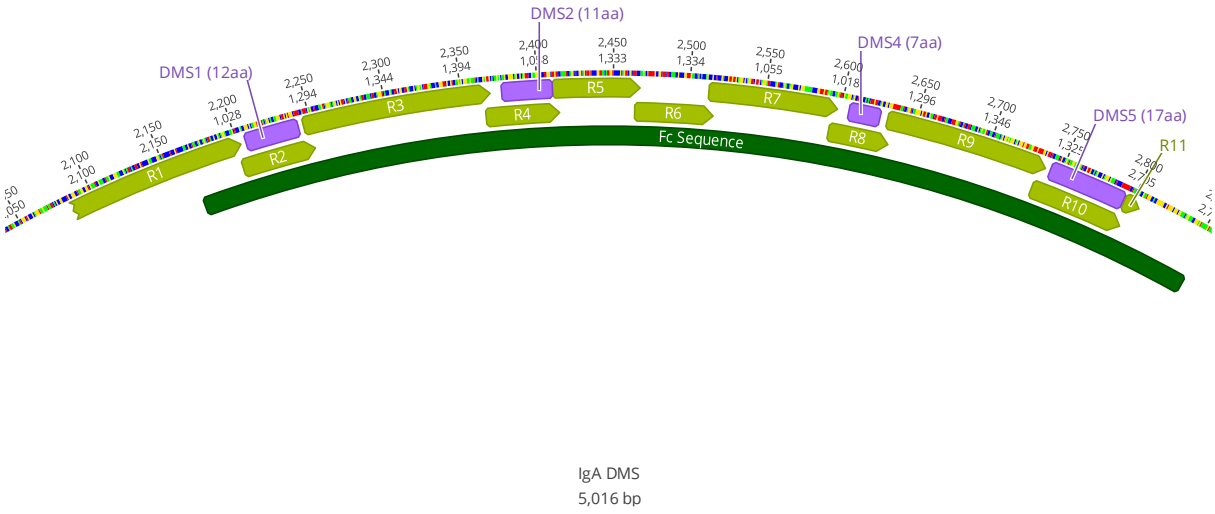

Figure 5. Annotated plasmid map of dual-site IgA Fc phagemid library construct. *The assembled construct comprises alternating deep mutational scanning (DMS) regions and framework regions within the Fc fragment. Light green segments indicate the Golden Gate assembly fragment, each defined by a unique four-nucleotide overhang that directs ordered ligation. Purple segments indicate the four DMS regions (positions 40-75, 211-243, 439-459, and 577-627 of the Fc reference) into which combinatorial nucleotide diversity was introduced using NNK degenerate codons. The dark green strip annotates the entire Fc reference (687 bp) used in this thesis. Any nucleotide along the Fc reference that is not in a DMS region is WT only. DMS3 is not assembled in this library due to the limitations of paired-end short-read length (i.e., a maximum of 2 x 300 at the time of library construction in 2023).*

1.7 Project Aim

To our knowledge, ONT has not publicly described the full composition of the training data used for `Dorado` basecalling models in peer-reviewed publications or in the main public software documentation. Public ONT materials describe the models and their continuing improvement, but not a complete training procedure. However, the models are trained using large collections of nanopore signal data paired with known reference sequences generated using specific sequencing chemistries. These datasets likely comprise diverse genomic samples and synthetic constructs spanning a wide range of sequence contexts. `Bonito` is ONT's open-source research basecaller and training platform for basecaller development. The GitHub repository was created in 2019 and includes tooling for training and evaluation, as well as a downloadable example training dataset; ONT currently recommends `Dorado` for routine basecalling and `Bonito` for training or method development.

Compared with the literature on basecaller benchmarking and inference, relatively few studies have focused on training or fine-tuning ONT basecalling models for domain-specific applications. Existing work nevertheless shows that custom training or adaptation can improve performance for taxa, repetitive regions, short reads, and modification-rich molecules (Ferguson et al., 2022; Vereecke et al., 2020; Z. Wang, Fang, et al., 2024; Z. Wang, Tu, et al., 2024; Wick et al., 2019).

The larger `sup` architecture is therefore a compelling candidate for transfer learning: its pretrained weights provide a general nanopore signal-to-sequence representation that can be adapted to the IgA Fc sequence context and library-preparation conditions examined in this thesis. This thesis aimed to test that premise by fine-tuning `sup` on experimentally collected WT IgA Fc nanopore datasets, then benchmarking the fine-tuned models on a held-out, variant-rich IgA Fc nanopore dataset against orthogonal short-read ground truth.

Chapter 2. Methods

2.1 Molecular & Cellular Biology Methods

2.1.1 Electrocompetent Cell Transformation

Six enriched IgA Fc variant plasmids were introduced into electrocompetent *Escherichia coli (E. coli)* XL1-Blue cells by electroporation. Fresh selection plates were prepared by supplementing molten LB agar at 50–55 °C with chloramphenicol to 25 µg/mL. SOC recovery medium (Super Optimal Broth supplemented with 20 mM glucose) was pre-warmed to 37 °C prior to use.

Electroporation cuvettes (0.2 cm gap) and microcentrifuge tubes were pre-chilled on ice. Each 40 µL of chilled electrocompetent E. coli aliquot was gently mixed with 2 µL of plasmid DNA in Type I water. The mixture was transferred to the pre-chilled cuvette, tapped gently to settle the contents, and the outer electrodes were wiped dry. Electroporation was performed using a MicroPulser (Bio-Rad, USA) according to the manufacturer's instructions (2.5 kV). The time constant was recorded and verified to fall within the expected range of 5.8–6.0 ms.

Immediately after the pulse, approximately 960 µL of pre-warmed (37 °C) SOC medium was added to the cuvette, and the suspension was gently mixed to prevent air bubbles. The entire volume was transferred to a 14 mL round-bottom culture tube and incubated at 37 °C with shaking at 225–250 rpm for 1 hour to allow phenotypic expression of the antibiotic resistance marker. Following recovery, 50 µL of each sample was plated onto pre-warmed LB agar selection plates and incubated overnight at 37 °C (16–18 hours). Plates were subsequently stored at 4 °C until inoculation.

2.1.2 Phage Production, Precipitation, and Titration

Six transformant E. coli XL1-Blue plates, each harbouring a unique IgA Fc phagemid variant, were used to inoculate liquid culture. A single colony from each plate was picked with a sterile micropipette, resuspended in 3 mL of LB medium supplemented with tetracycline (20 µg/mL), chloramphenicol (25 µg/mL), and 2% (w/v) dextrose, and then transferred to a 14 mL round-bottom culture tube. The culture was grown overnight at 37 °C with shaking at 200–250 rpm to serve as the starter culture.

The overnight culture was diluted 1:100 into fresh 2×YT medium supplemented with tetracycline (20 µg/mL), chloramphenicol (25 µg/mL), and 2% (w/v) glucose. The production culture was incubated at 37 °C with shaking until the OD_{600} reached 0.3–0.5. M13K07 helper phage (1×10^{13} CFU/mL stock) was added at a multiplicity of infection of 10. The infected culture was incubated at 30 °C for 20 minutes without shaking, then for 40 minutes with shaking. Kanamycin (35 µg/mL) and IPTG (0.1–1 mM) were then added to induce Fc-pIII-fusion expression, and the culture was incubated at 30 °C with shaking for 8–9 hours.

Phage was harvested by centrifugation at 7,000 × g for 10 minutes at 4 °C (fixed-angle rotor). The supernatant was filter-sterilised sequentially through 0.45 µm and 0.22 µm syringe filters. Phage precipitation was carried out entirely on ice. Chilled 5× PEG/NaCl solution (20% [w/v] PEG-8000, 2.5 M NaCl, autoclaved) was added at a 1:4 ratio (PEG/NaCl : supernatant) and mixed by inversion. The mixture was incubated on ice for 1 hour and centrifuged at 11,000 × g for 30 minutes at 4 °C (fixed-angle rotor). The supernatant was carefully removed, and the pellet resuspended in chilled PBS (pH 7.4) by gentle mixing, followed by a 30-minute incubation on ice and a clearing spin at 11,000 × g for 30 minutes at 4 °C (fixed-angle rotor). A second precipitation was performed identically, with a reduced incubation of 30 minutes. The final pellet was resuspended in chilled PBSG (PBS pH 7.4 supplemented with 50% [v/v] glycerol), cleared by centrifugation at 11,000 × g for 30 minutes at 4 °C, and stored at −20 °C.

Phage titres were determined by serial dilution. Each overnight culture of XL1-Blue was diluted into 2×YT medium and grown at 37 °C with shaking to an OD_{600} of approximately 0.3. In a 96-well plate, 90 µL of bacterial culture was dispensed per well. Ten microlitres of the phage suspension was added to the first well and serially diluted 1:10 across 12 wells in triplicate, changing tips between transfers. After 30 minutes of static incubation at 37 °C, 5 µL from dilutions 10^{-7} through 10^{-12} were spotted onto LB agar plates supplemented with kanamycin and chloramphenicol. Plates were incubated overnight at 37 °C, and colonies within the countable range were counted. The titre (CFU/mL) was calculated as the average colony count multiplied by 200 (to correct for the 5 µL spot volume relative to 1 mL) and divided by the dilution factor.

2.1.3 Phage-Display ELISA

Phage-display enzyme-linked immunosorbent assays (ELISA) were performed to assess the binding activity of IgA Fc phage variants and WT against the cognate receptor FcαRI. Two

ELISA formats were employed: a sandwich ELISA to measure relative binding affinity, and a direct ELISA to estimate phage concentration.

For the sandwich ELISA, 96-well plates were coated with recombinant FcαRI at 15 µg/mL in 50 mM sodium carbonate/bicarbonate coating buffer (pH 9.5) by adding 50 µL per well and incubating overnight at 4 °C. Wells were blocked with PBSMT (PBS supplemented with 2% [w/v] skim milk powder and 0.05% [v/v] Tween-20) for 1 hour at room temperature with shaking, then washed three times with PBST (PBS with 0.05% Tween-20). Phage samples were normalised to a starting concentration of approximately 3.4×10^{10} CFU/mL and serially diluted 1:3 down the plate in PBSM (PBS with 2% skim milk). After 1 hour of incubation at room temperature, the plates were washed three times with PBST. Bound phages were detected by incubating with an anti-M13 HRP-conjugated monoclonal antibody (Sino Biological, #11973-MM05T) diluted 1:20,000 in PBSM for 1 hour at room temperature. After five washes with PBST, 50 µL of TMB Ultra substrate was added to each well. The reaction was stopped after colour development with 50 µL of 1 M H_2SO_4, and absorbance was measured at 450 nm.

Binding was assessed at pH 7.4 and 5.8 using PBS-based buffers. Controls included M13K07 helper phage as a positive control for the anti-M13 antibody and wells without phage coating as negative controls.

For the direct ELISA, phage samples were serially diluted in 50 mM sodium carbonate/bicarbonate coating buffer (pH 9.5) and coated directly onto 96-well plates at 50 µL per well, starting from the same normalised concentration used in the sandwich format (approximately 3.4×10^{10} CFU/mL) and serially diluted 1:3 down the plate. Plates were incubated overnight at 4 °C and blocked with PBSMT for 1 hour at room temperature with shaking. After three washes with PBST, bound phage was detected using the same anti-M13 HRP-conjugated monoclonal antibody (Sino Biological, #11973-MM05T), diluted 1:20,000 in PBSM, for 1 hour at room temperature. Following five washes with PBST, 50 µL of TMB Ultra substrate was added per well, the reaction was stopped with 50 µL of 1 M H_2SO_4, and absorbance was measured at 450 nm. This format allowed independent estimation of relative phage concentrations across preparations, independent of target binding.

2.1.4 Colony Selection

The IgA Fc phage-display library was maintained as a glycerol stock at −80 °C using standard NEB® Turbo Competent E. coli (NEB, USA). To recover library diversity, LB agar selection plates were prepared by supplementing molten agar (cooled to 55 °C) with 2% (w/v) dextrose and chloramphenicol (20 μg/mL, from a 200 mg/mL stock diluted 1:1000). Plates were poured at approximately 4 mm thickness under laminar flow and incubated at 37 °C until required.

For plating, the glycerol stock was thawed on ice and diluted 1:200 in 2×YT medium (2 mL total). The optical density at 600 nm (OD_{600}) was measured using a spectrophotometer blanked against plain 2×YT. Colony-forming units per millilitre (CFU/mL) were estimated using the approximation 1 $OD_{600} \approx 5 \times 10^8$ CFU/mL. A target of approximately 150,000 CFU per plate was calculated, and the appropriate volume was spread onto 8 selection plates under sterile conditions. Plates were incubated inverted at 37 °C for 16 hours (overnight).

After overnight incubation, plates were transferred to 4 °C for at least 2 hours. Colony counts were recorded from each plate to estimate total library diversity and to confirm that the target minimum of 150,000 CFU was achieved across the eight plates combined. Colonies were subsequently scraped from all plates using sterile cell scrapers into 1.5 mL microcentrifuge tubes and resuspended by pipetting. The pooled cell suspension was processed immediately for plasmid extraction as described in Section 2.1.5.

2.1.5 Plasmid Extraction

Approximately 14 mL of cell suspension was recovered by scraping the eight selection plates. An aliquot of 3 mL (approximately 510 OD units) was processed using the E.Z.N.A.® Plasmid DNA Maxi Kit (Omega Bio-tek) according to the manufacturer's protocol. Cells were pelleted by centrifugation at 4,000 ×g for 10 minutes and resuspended in Solution I (RNase A resuspension buffer). Alkaline lysis was performed by the addition of Solution II (NaOH/SDS) with gentle inversion, followed by neutralisation with Solution III (potassium acetate buffer). The lysate was clarified by centrifugation at 15,000 ×g for 10 minutes at 4 °C, and the cleared supernatant was loaded onto a HiBind® DNA Maxi Column. The column was washed with HBC buffer and DNA Wash Buffer, and plasmid DNA was eluted in 3 mL of Elution Buffer.

2.1.6 Primer Design

Four oligonucleotide primers (KH049, KH050, KH053, KH054) were designed for a two-round nested PCR strategy to amplify the IgA Fc fragment from the phagemid library. The first-round primer pair (KH049/KH050) flanked the full Fc fragment region and incorporated dual UMI motifs (NNNWNNDV and NNNSNNNB), giving a combined theoretical diversity of 452,984,832 (18,432 x 24,576). The second-round primers (KH053/KH054) annealed to the ends of the first-round amplicon, introducing partial Nextera R1 and R2 adapter sequences for downstream MGI adaptor ligation. The primer sequences and properties are provided in Appendix A.

2.1.7 Fragment Extraction

An initial approach to isolate the Fc fragment derived from the phagemid backbone was attempted via restriction enzyme digestion. The IgA Fc variant plasmids extracted (270.2 ng/µL) were digested with SfiI (20 U) in 1× rCutSmart™ Buffer at 50 °C for 1 hour in a 50 µL reaction volume. The digest products were resolved on a 1.5% agarose gel stained with GelPilot DNA Loading Dye and run at 100 V for 30 minutes. The expected ~700 bp Fc band was excised under a blue-light transilluminator using a scalpel, with excess agarose trimmed to within 2 mm of the band. The excised gel slice was dissolved in 3× volume (w/v) QIAGEN QG solubilisation buffer at 55 °C for 30 minutes, then mixed with 1× gel volume isopropanol and purified using a Zymo-Spin I column. After washing with TE buffer and eluting in 6 µL of Zymo DNA Elution Buffer (10 mM Tris-HCl, pH 8.5, 0.01 mM EDTA), the recovered fragment concentration was measured by NanoDrop.

2.1.8 PCR Amplification

Two-round nested PCR was performed using Q5 High-Fidelity DNA Polymerase (NEB #M0491). Each 1st round PCR reaction (10 µL) contained 1× Q5 Reaction Buffer, 0.2 mM dNTPs, 0.5 µM each of the forward and reverse primers, 20 ng/µL template DNA, and 0.02 U/µL Q5 polymerase, with nuclease-free water to volume. Thermocycling conditions comprised initial denaturation at 98 °C for 20 seconds; 5 cycles of 98 °C for 10 seconds, annealing at the primer-specific temperature for 30 seconds, and extension at 72 °C for 30 seconds; followed by a final extension at 72 °C for 2 minutes. Products were held at 12 °C. The annealing temperature (68 °C) was optimised by gradient PCR across 62.1–70.1 °C to

accommodate the different melting temperatures of the KH049 (T_m = 70 °C) and KH050 (T_m = 67 °C) primer pair.

For the 2nd round PCR, the purified 1st round PCR product served as the template at 20 ng/µL per reaction (10 µL), with 5 PCR cycles. Reactions were otherwise identical to those in the first-round PCR. Annealing temperature (60 °C) was optimised by gradient PCR across 55.9–62 °C to accommodate the different melting temperatures of the KH053 (T_m = 62 °C) and KH054 (T_m = 57 °C) primer pair. Full thermocycling parameters for both rounds are listed in Appendix B.

Post-PCR cleanup was performed in two steps. Residual primers and dNTPs were first degraded by EXO-CIP treatment: 1 µL each of Exo-CIP A and Exo-CIP B was added to the pooled reactions, which were then incubated at 37 °C for 5 minutes. Amplicons were then purified using Mag-Bind® TotalPure NGS at a 0.65× bead-to-sample ratio, which preferentially retains fragments of 700–800 bp. After a 10-minute binding incubation at room temperature, beads were captured on a magnetic rack, washed twice with freshly prepared 85% ethanol, and air-dried for 5 minutes. DNA was eluted in 20 µL of nuclease-free water, incubated at room temperature for 10 minutes, incubated on the magnet for 5 minutes, and 16 µL of eluate was recovered. Amplicon size and purity were verified by agarose gel electrophoresis (1.2% gel, 90 V, 30 minutes).

2.1.9 Sequencing Prep

Amplicon libraries for Oxford Nanopore sequencing were prepared using the Native Barcoding Kit 24 V14 (SQK-NBD114-24). Purified 1st round PCR amplicons were end-prepared with the NEBNext Ultra II End Repair/dA-Tailing Module (NEB E7546) by incubation at 20 °C for 5 minutes, followed by 65 °C for 5 minutes. Native barcodes were ligated using NEB Blunt/TA Ligase Master Mix at room temperature. Barcoded samples were then pooled, purified with magnetic beads, ligated to sequencing adapters, and subjected to a final bead clean-up before loading onto a used R10.4.1 flow cell for sequencing on a MinION Mk1B.

Purified 2nd round PCR amplicons were submitted to an external sequencing provider (Livestock Improvement Corporation, Hamilton, New Zealand) for short-read sequencing on an MGI DNA nanoball-based platform. The provider performed downstream library preparation and sequencing in accordance with their internal protocol.

2.2 Computational Methods

2.2.1 Dataset Design and Inputs

The computational workflow used three previously produced nanopore fine-tuning inputs: `A_full.pod5` (restriction-digest-based dataset with WT IgA Fc fragment) and `B_full.pod5` (PCR-based dataset with a mixture of WT and library-derived IgA Fc fragments), and `AB_full.pod5` (combined `A_full` + `B_full` fine-tuning dataset) to derive the `Alpha`, `Beta`, and `Gamma` model families. One held-out nanopore test dataset (`2025_07_31.pod5`) and one orthogonal short-read ground-truth dataset (`UDP0057.fq`) were used for validation. Dual unique molecular identifiers (UMIs) introduced during the first-round PCR design enabled read-level linkage between nanopore and MGI datasets when both UMIs were recovered from the amplicon sequence.

All alignment, polishing, and variant calling steps used the same single-contig 687 bp WT Fc amplicon reference (`fc_reference.fa`). DMS mutational windows were defined by the intervals 40-75, 211-243, 439-459, and 577-627, and these coordinates were supplied to the downstream correction and variant-calling tools as a fixed zone file (`DMSZones.txt`). Model hyperparameters and training summaries can be found in Appendix D. Source code and computed analytical metrics are available on GitHub (Appendix E). Two tables containing brief descriptions of each custom-written program/workflow and each third-party tool/library are also provided in Appendix E.

2.2.2 Basecalling, Filtering, and Alignment

Raw nanopore signals were basecalled with `dna_r10.4.1_e8.2_400bps_sup@v5.2.0` using the kit specification `SQK-NBD114-24` and the `--trim-all` option to remove the barcode and adapter sequences. `FASTQ` conversion was performed with the `samtools FASTQ` command.

For fine-tuning subset construction, reads from `A_full.pod5`, `B_full.pod5`, and `AB_full.pod5` were basecalled, converted to `FASTQ`, and subjected to three sequential filters: (i) length filter retaining reads between 600 and 800 bp; (ii) quality filter retaining reads with a per-read mean Phred quality score of $\bar{Q} \geq 15$, where $\bar{Q}$ is computed by power-transforming per-base quality scores Q_i to error probabilities P_i:

$$P_i = 10^{\frac{-Q_i}{10}} \tag{2.1}$$

then calculating the arithmetic mean of error probabilities across all *n* bases:

$$\bar{P} = \frac{1}{\mathrm{n}} \sum_{i=1}^{n} P_i \tag{2.2}$$

and log-transforming as:

$$\bar{Q} = -10 \log_{10} \bar{P} \tag{2.3}$$

Moreover, (iii) alignment filter retaining reads with a primary alignment to the Fc reference. Only reads that passed all three filters had their read IDs extracted. These IDs were used to recover the corresponding nanopore signal subsets `A_filtered.pod5`, `B_filtered.pod5`, and `AB_filtered.pod5`.

For model benchmarking on `2025_07_31.pod5`, models were used together with `Dorado` for basecalling, and the resulting `BAM` was converted to `FASTQ` with `samtools`.

Length filtering was then performed using `fastplong` with a minimum read length of 600 bp and a maximum read length of 800 bp. Quality filtering was not used during benchmarking, so each fine-tuned model was assessed without data loss from miscalibration of the Q score. By contrast, the $\bar{Q} \geqslant 15$ filter was applied only when generating filtered subsets for training.

Alignments were generated against the WT Fc reference using `minimap2` with the `map-ont` preset for nanopore reads and the `sr` preset for short reads, followed by name-sorting with `samtools`. Primary alignments were extracted by excluding unmapped, secondary, and supplementary records (`samtools view -F 0x904`) before DMS filtering and single-read correction. Primary alignment summaries were derived from `samtools stats` and `CIGAR`-based insertion and deletion counts, yielding counts of inserted and deleted bases, pairwise-derived mismatches (not from X/= symbols), mapped bases, error rate (mismatch rate), and overall error rate.

2.2.3 UMI Processing and Annotation

Dual UMI extraction was performed by searching each read for the 1st round PCR primer pair and recovering the degenerate nucleotides within the primer sequences. A custom C++ extractor, `UmiExtractor`, was written and used for this step. For reads in the forward orientation, the algorithm searched from the 5' end for the forward primer and from the 3' end for the reverse primer (reverse-complemented), allowing a user-specified number of mismatches. If this configuration was not matched, the reverse orientation was tested by searching for the reverse primer at the 5' end and the forward primer (reverse complemented) at the 3' end. Reads matching only in the reverse orientation were reverse complemented before downstream processing, thereby placing all retained reads in a canonical forward orientation. Once a valid primer span was identified, the degenerate positions within each primer were extracted as the forward and reverse UMI sequences, respectively, and appended to the `FASTQ` header as `UMI_<forward>_<reverse>.` The read was then trimmed to the insert region bounded by the UMI pair. Reads for which no valid primer span was identified or reads containing characters outside the permitted degenerate nucleotide at UMI positions, were discarded.

Two supporting C++ utilities were used to transfer or reinterpret UMI annotations between `FASTQ` files. The first tool, `UmiCloner`, transferred UMI-contained `FASTQ` headers from one file to another by matching reads on their underlying read IDs, independent of any pre-existing UMI suffix. This allowed UMI tags identified in one basecalled dataset to be applied to an alternative basecalled dataset derived from the same underlying `POD5` signal set. The second tool, `UmiSpliter`, parsed dual-UMI headers and wrote two derived `FASTQ` output files: one containing only the first UMI and the other containing only the second UMI. During this step, the program also enumerated distinct, duplicated, and singleton dual-UMI and single-UMI keys, providing diagnostic information on UMI diversity and collision behaviour within each dataset.

2.2.4 Model Fine-Tuning and Training-Set Construction

Nine derivative models were produced by fine-tuning the `dna_r10.4.1_e8.2_400bps_sup@v5.2.0` model without changing the underlying neural network architecture. The exported `Bonito` configuration defined a transformer-based model with five convolutional blocks, an 18-layer transformer encoder (*d_model = 512, 8 attention heads, feed-forward width = 2048*), a linear upsampling layer, and a CRF head for output logits.

Training inputs were prepared as `NumPy` n-dimensional array triplets (`chunks.npy`, `references.npy`, and `reference_lengths.npy`). `Alpha`, `Beta`, and `Gamma` were trained on `A_full`, `B_full`, and `AB_full`, respectively. `AlphaQCL`, `BetaQCL`, and `GammaQCL` were trained on the corresponding filtered signal subsets. In contrast, `AlphaQCH`, `BetaQCH`, and `GammaQCH` used the same filtered dataset but applied a stricter `Bonito` alignment-retention setting (`--min-accuracy-save-ctc 0.99`) during array generation so that only the highest-confidence signal-to-reference matches were retained.

All fine-tuning was conducted in a single iteration over the corresponding training set (i.e., 1 epoch). It is possible to fine-tune a model over many epochs with a reduced learning rate to reach the same training and validation losses. However, it takes much longer, and early testing showed no difference in performance compared to models trained with only 1 epoch when optimal learning rates were used. Family-specific learning rates, batch sizes, and train/validation chunk counts were recorded in the archived `config.toml`, `training.csv`, and `losses_1.csv` files. By default, output weights from fine-tuning are stored in the compressed `weight_1.tar` file, which must be decompressed to raw weights using `Bonito export` before the `Dorado basecaller` can basecall with the fine-tuned model weights. Hyperparameter tuning for each model was guided by the goal of achieving the lowest validation loss, whilst allowing a slightly higher training loss.

Several auxiliary Python utilities were written to support these steps. `POD5Splitter.py` partitioned large `POD5` files into even read-count subsets for parallel processing; `POD5Merger.py` reassembled split `POD5` outputs (i.e., A + B), which is required for streaming reads from different `POD5` files in a round-robin fashion, and shuffled reads after merging to ensure homogeneity during training; and `MergeNumPy.py` merged `Bonito` training arrays using memory-mapped streaming to avoid large system memory spikes during dataset construction.

Fine-tuned nanopore models were compared with the pretrained `dna_r10.4.1_e8.2_400bps_sup@v5.2.0` baseline by matching tensor names between exported weight files. `Weight_stats.py` script was written for model weight analysis. For each matched tensor T_k, the weight change was calculated as:

$$\Delta W(T_k) = W_{fine-tuned}(T_k) - W_{sup}(\mathrm{T_k}) \tag{2.4}$$

Where $W_{sup}(\mathrm{T_k})$ denotes the corresponding tensor in the pretrained `sup` baseline. Matched tensors were then grouped into architectural components, including convolutional kernels, transformer self-attention, transformer feed-forward network, transformer norm/deepnorm layers, linear upsample layer, and CRF head.

For each component *C,* the weight changes were estimated by combining all weight changes from tensors assigned to that component using the L2 norm**:**

$$\|\Delta W(C)\|_2 = \sqrt{\sum_k (\Delta W(T_k)^2} \tag{2.5}$$

To compare weight change across components with different magnitudes, the normalised L2 norm was calculated as:

$$\frac{\|\Delta W(C)\|_2}{\|W_{sup}(C)\|_2} = \frac{\sqrt{\sum_k (\Delta W(T_k)^2}}{\sqrt{\sum_k (W_{sup}(T_k)^2}} \tag{2.6}$$

The component-level squared L2 norm was also calculated:

$$\|\Delta W(C)\|_2^2 = \sum_k (\Delta W(T_k)^2 \tag{2.7}$$

Although this quantity is directly related to the L2 norm, it is useful because squared L2 norms are additive across components. Therefore, the model-level squared L2 norm of the model was calculated as:

$$\|\Delta W(M)\|_2^2 = \sum_C \|\Delta W(C)\|_2^2 \tag{2.8}$$

where *M* denotes the full model, and *C* denotes each architectural component. The share of weight change for each component was then calculated as:

$$P_{\|\Delta W(M)\|_2^2}(\mathrm{C}) = \frac{\|\Delta W(C)\|_2^2}{\|\Delta W(M)\|_2^2} \tag{2.9}$$

The metric estimates the proportion of model-level squared L2 norm contributed by each component as a result of fine-tuning.

To assess whether there are any extreme localised weight changes induced by fine-tuning, the $P_{99.9}$ standardised absolute change was calculated. For each tensor, absolute weights were first standardised by the standard deviation of the corresponding pretrained `sup` tensor:

$$Z_{\Delta}(\mathrm{T_k}) = \frac{|\Delta W(\mathrm{T_k})|}{SD(W_{sup}(T_k))} \tag{2.10}$$

This value is not a conventional mean-subtracted statistical z-score; rather, it is a standardised absolute weight change measure. For each component, all standardised absolute weight change values were flattened and concatenated, and the 99.9$^{\text{th}}$ percentile was calculated:

$$P_{99.9}(C) = Percentile_{99.9}\left(\left\{\frac{|\Delta W(T_k)|}{SD\left(W_{sup}(T_k)\right)}\right\}_{T_k \in C}\right) \tag{2.11}$$

A model-level $P_{99.9}$ standardised absolute weight change was calculated in the same manner after pooling standardised absolute weight changes across all matched tensors in the model:

$$P_{99.9}(M) = Percentile_{99.9}\left(\left\{\frac{|\Delta W(T_k)|}{SD\left(W_{sup}(T_k)\right)}\right\}_{T_k \in M}\right) \tag{2.12}$$

For the transformer block analysis in Figure 29, the normalised L2 norm was applied at a finer granularity. Matched tensors from each transformer encoder block b were grouped into self-attention (*SA*), feed-forward (*FF*), or norm/deepnorm (*NDN*) layers, with normalised L2 norm calculated separately within each block. $\Delta W^b(C)$ denote the difference between the fine-tuned and pretrained `sup` weights for component C in transformer block b, and $W_{sup}^b(C)$ denotes the corresponding pretrained `sup` weights.

Finally, Spearman's rank correlation coefficient was used to examine associations between selected component-level weight change and downstream metrics. For each comparison, Spearman's ρ was calculated as the Pearson correlation between ranked values of the selected component-level weight change X, and the downstream metric Y:

$$p_s = corr(rank(X), rank(Y)) \tag{2.13}$$

Only the 9 fine-tuned models were included in the weight change analysis. The pretrained `sup` baseline was excluded because its weight change relative to itself is zero. The Spearman associations were purely exploratory due to the small number of fine-tuned models and the dependence introduced by partial sharing of training data across the same model families.

2.2.5 DMS-aware Single-read Correction and Variant Extraction

Single-read correction and variant extraction were performed with the custom C++ program `DualSiteDMSFilter`. For each primary alignment, the string was walked against the Fc reference to reconstruct per-position read bases. A corrected read of reference length was initialised as the WT sequence with Q40 at every position; bases observed within the defined DMS regions were then copied from the aligned read into that corrected sequence, whereas bases outside the DMS regions were restored to the WT reference.

Reads were rejected under any of the following criteria: (i) more than three indel events or more than three total indel bases across the full-length read (a threshold selected to exclude reads with excessive structural error whilst permitting a small number of isolated nanopore indels inherent to the sequencing chemistry); (ii) incomplete coverage of first and last DMS regions (same as all DMS regions for continuous amplicon); (iii) more than two mutated codons across all DMS regions; or (iv) a mutated codon not consistent with the NNK library design rule, whereby the third codon position must be G or T. Under the NNK constraint, each accepted mutated codon was required to be fully observed in-frame with adequate base quality at all three positions. A stricter all-DMS-regions coverage requirement was implemented in the tool, although during early testing, this approach did not affect the comparison.

Passing reads were written both as corrected `FASTQ` records and as `variant_key/count` `TSV` tables keyed by sorted substitution strings. Because the corrected `FASTQ` output was forced back to wild type outside the DMS regions, framework noise was quantified separately from the raw

aligned read bases rather than from the corrected sequences. The polished `FASTQ` output was then realigned to the Fc reference with `minimap2` for post-correction error estimation.

In addition to sequence correction, `DualSiteDMSFilter` quantified residual sequencing error separately for DMS and framework regions. DMS-region mismatches were counted from the corrected sequence within the four mutational windows, reflecting genuine variant calls and any residual miscalls within those windows. Framework mismatches were counted independently from raw aligned read bases outside the DMS regions because the corrected sequence is intentionally forced to WT at framework positions and therefore cannot be used to estimate errors in those regions. The program reported per-read mismatch burdens and normalised mismatch rates at both the nucleotide and codon levels for DMS regions, framework regions, and their difference.

2.2.6 Benchmark and Evaluation Metrics

Model performance was assessed at two complementary levels: reference-based alignment evaluation and variant-abundance concordance scoring against the short-read ground truth. A third benchmarking methodology, UMI-paired nucleotide accuracy, was developed and implemented in the custom C++ program `Benchmarker`.

UMI-paired Nucleotide Accuracy

For UMI-paired nucleotide benchmarking, the custom C++ program `Benchmarker` was designed to match nanopore reads to short reads from `UDP0057` using dual UMI tags appended to `FASTQ` headers. Each matched pair was programmed to be compared using a banded global Needleman–Wunsch alignment using unit edit costs *(match = 0, substitution = 1, insertion = 1, deletion = 1)* and a band width of *max(50, |n – m| + 50),* where n and m are the respective sequence lengths. Global alignment was chosen instead of local alignment, such as Smith-Waterman, because the amplicon is expected to cover the entire reference sequence. Ground-truth positions bearing a quality character of '#' (low-confidence positions in the MGI paired-end overlap region) were excluded from comparison. At the nucleotide level, $Precision_{nucleotide}$ was defined as:

$$Precision_{nucleotide} = \frac{matched}{matched + mismatches + insertions} \qquad (2.14)$$

furthermore, $Recall_{nucleotide}$ was defined as:

$$Recall_{nucleotide} = \frac{matches}{matches + mismatches + deletions} \quad (2.15)$$

with $F1_{nucleotide}$ derived from these metrics:

$$F1_{nucleotide} = \frac{2 \times Precision_{nucleotide} \times Recall_{nucleotide}}{Precision_{nucleotide} + Recall_{nucleotide}} \quad (2.16)$$

If the banded global Needleman-Wunsch fails to locate a valid alignment path, the algorithm falls back to an unbanded full dynamic programming matrix to guarantee alignment completion, which significantly increases computational time.

Reference-based Alignment Evaluation

For reference-based alignment evaluation, each nanopore dataset and the `UDP0057` ground truth were aligned to the 687-bp WT Fc reference. `CIGAR`-based counts of inserted bases (N_{ins}), deleted bases (N_{del}), mismatched bases (N_{sub}), and total mapped bases (N_{mapped}) were extracted using `samtools` stats and custom parsing. The primary overall error rate (E) was defined as the composite proportion of erroneous aligned bases:

$$E = \frac{N_{sub} + N_{ins} + N_{del}}{N_{mapped}} \quad (2.17)$$

This composite metric was preferred over a substitution-only rate because nanopore sequencing mistakes include a significant proportion of indel and homopolymer errors. The DMS filter pass rate was defined as:

$$DMS\ pass\ rate = \frac{N_{pass}}{N_{aligned}} \quad (2.18)$$

where N_{pass} is the number of reads satisfying all four DMS filter criteria and $N_{aligned}$ is the total number of primary-aligned reads. Primary and polished summaries were derived separately, the latter from realignment of corrected `FASTQ` reads to the Fc reference.

Variant-abundance Concordance Scoring

For variant-abundance concordance, nanopore-derived and short-read-derived variant tables (each containing `variant_key/count` pairs produced by `DualSiteDMSFilter`) were compared at minimum count thresholds t = 1 to 10 using the custom C++ `VariantConcordance` program. At each threshold, the source variant set was restricted to nanopore variants with count ≥ t and the ground-truth set to short-read variants with count ≥ t. The intersection (variants present in both sets above t) and union (variants present in either set above t) were recorded, together with the retained variant counts for each dataset. Variant-level precision, recall, and F1 score were calculated to quantify population-level agreement. For a given threshold, a variant was classified as a true positive (*TP*) if it was observed in both datasets, a false positive (*FP*) if present only in the nanopore set, and a false negative (*FN*) if present only in the ground truth. At the variant-level, $Precision_{variant}$ was defined as:

$$Precision_{variant} = \frac{V_{TP}}{V_{TP} + V_{FP}} \tag{2.19}$$

$Recall_{variant}$ was defined as:

$$Recall_{variant} = \frac{V_{TP}}{V_{TP} + V_{FN}} \tag{2.20}$$

The $F1_{variant}$ was defined as the harmonic mean of $Precision_{variant}$ and $Recall_{variant}$:

$$F1_{variant} = \frac{2 \times Precision \times Recall}{Precision + Recall} \tag{2.21}$$

where V_{TP} denotes the set of true-positive variants, V_{FP} denotes the set of false-positive variants, and V_{FN} denotes the set of false-negative variants.

In addition to variant-level concordance, three complementary variant-abundance concordance statistics were also computed from the retained variant sets. Exact variant overlap mass was defined as the sum of the smaller normalised abundance across all variants in the union, where abundance was normalised within each dataset independently to sum to 1 across all retained variants:

$$Exact\ overlap\ mass = \sum_{v} \min\left(f_{ONT}(v), f_{MGI}(v)\right) \tag{2.22}$$

where $f_{ONT}(v)$ and $f_{MGI}(v)$ are the normalised frequencies of variant v in the nanopore and ground-truth datasets, respectively, and the sum is taken over all variants in the union. A value of 1.0 indicates that both datasets assign the same normalised abundance to each variant; a value of 0 indicates no shared abundance.

Weighted Jaccard similarity (*WJS*) was computed as a count-weighted measure of variant-set overlap:

$$WJS = \frac{\sum_{v} \min\left(c_{ONT}(v), c_{MGI}(v)\right)}{\sum_{v} \max\left(c_{ONT}(v), c_{MGI}(v)\right)} \tag{2.23}$$

where $c_{ONT}(v)$ and $c_{MGI}(v)$ are the raw counts for variant v in the nanopore and ground-truth datasets, respectively, and the sum is taken over all variants in the union (with a count of zero for variants absent from one dataset). This metric penalises both set-level differences (variants found in only one dataset increase the denominator) and abundance-magnitude differences (a large count discrepancy at a shared variant reduces the ratio).

Jensen–Shannon similarity (*JSS*) was used as a distribution-level concordance measure. The Jensen–Shannon divergence (*JSD*) between the normalised nanopore abundance distribution P and the normalised ground-truth distribution Q was computed as:

$$JSD(P,Q) = \frac{1}{2}KL(P||M) + \frac{1}{2}KL(Q||M), M = \frac{P+Q}{2} \tag{2.24}$$

where *KL* denotes the Kullback–Leibler divergence and M is the pointwise mean distribution. *JSS* was defined as $1 - \sqrt{JSD(P,Q)}$, which ranges from 0 (completely dissimilar distributions) to 1 (identical distributions). Higher similarity values indicate that the variant abundance spectrum of the nanopore-derived table closely matches that of the ground truth, regardless of whether the same individual variants are retained.

Composite Concordance Score (CCS)

To summarise concordance across multiple count thresholds, a threshold-robust score (CCS) was calculated for each model from the threshold-response profiles of four metrics: weighted Jaccard similarity, Jensen–Shannon similarity, exact overlap mass, and variant F1 score. Each metric was evaluated at minimum variant counts ranging from 1 to 10. For each model and metric, the threshold-response profile was summarised as the area under the curve using trapezoidal integration across the discrete threshold range. Given that the variant-count threshold ranges from 1 to 10 and each metric is bounded on [0,1], the theoretical maximum AUC is 9 (nine unit-width trapezoids at the ceiling value). Each of the four AUCs computed was therefore normalised by dividing by 9, mapping all scores to the range [0,1] on an absolute scale. Unlike min-max normalisation, this approach ensures that the normalised AUC of each model is independent of the other models in the comparison set. The final score was then computed as the weighted geometric mean of the four normalised AUC scores:

$$CCS = J^{0.2} \times JS^{0.2} \times M^{0.3} \times F1^{0.3} \tag{2.25}$$

where J is the normalised weighted Jaccard AUC, JS is the normalised Jensen–Shannon AUC, M is the normalised exact overlap mass AUC, and $F1$ is the normalised variant F1 AUC. Greater weight was assigned to exact overlap mass and variant F1 because these metrics directly penalise incorrect abundance assignment and false-positive or false-negative variant calls, respectively. The geometric mean was used rather than the arithmetic mean so that poor performance in any single component substantially reduces the overall score; if any normalised component equalled zero, the composite was set to zero. The weighting emphasised recovery of correct variant mass and balanced variant detection while retaining abundance-distribution concordance as supporting metrics. CCS, therefore, rewarded models that consistently maintained concordance across different aspects under varying degrees of filtering, rather than models that performed well at a single arbitrary count threshold.

For Section 3.6 final synthesis, each fine-tuned model configuration was retrained across 10 random seeds, with the training input, model configuration, and downstream benchmarking workflow held fixed for that configuration. Each seeded model export was processed through the same basecalling, DMS filtering, variant extraction, threshold concordance, AUC normalisation, and CCS calculation pipeline. Seeded-level CCS and normalised AUC scores were summarised using the mean, sample standard deviation, and a two-sided 95% t-based confidence interval.

The 95% confidence interval was calculated as:

$$CI_{95\%} = \bar{x} \pm t_{0.975,n-1}\frac{s}{\sqrt{n}} \quad (2.26)$$

where n is the number of model runs, and s is the sample standard deviation across runs. Given 10 random seeds, n = 10 and $t_{0.975,9}$ = 2.262. These intervals describe only fine-tuning seed variation; they do not estimate uncertainty from independent sequencing runs, library preparation, biological replication, or read sampling.

Empirical Cumulative Distribution Function (ECDF) of Intersecting Variant Abundance

ECDF plots were used to characterise the distribution of variant abundance among the variants shared between each nanopore model and the ground-truth dataset. For each model, the nanopore-derived and ground-truth variant tables were filtered at a minimum count threshold of 1, WT entries were removed, and counts were normalised within each dataset so that the retained non-WT abundances summed to 1. The two tables were then joined using an exact variant key inner join, retaining only the variants present in both datasets (intersecting variants) for the ECDF analysis. Two complementary ECDF views were produced from the same intersecting-variant table. In the model scale abundance ECDF (Figure 33 and Figure F5), the x-axis represents the abundance assigned by the nanopore model to each intersecting variant. In the ground-truth scale ECDF (Figures 34 and F6), the x-axis represents the abundance assigned by `UDP0057` to the intersecting variants. In both cases, the y-axis represents the cumulative proportion of intersecting variants with abundance at or below the corresponding x-axis value. The two views together allow assessment of both how the model ranks intersecting variants internally and whether the variants recovered by the model tend to be high- or low-abundance in the ground truth.

All ECDF plots were rendered on a $\log_{10}$ abundance axis because intersecting variant abundances spanned several orders of magnitude. The ground truth ECDF (all retained non-WT ground-truth variants) was included in both panels as a reference curve, providing a view of the complete detectable variant abundance spectrum against which intersecting variant recovery can be assessed. This analysis complements the threshold-based concordance metrics by revealing not only whether the same variants are recovered but also how their abundances are distributed within the intersection set.

Chapter 3. Results & Discussion

3.1 Dataset Overview and Benchmark Construction

To assess whether domain-specific model training enhances variant-calling accuracy for the IgA Fc DMS library, the benchmark design incorporated both wet-laboratory workflows and biological considerations. Figure 6 presents the study as a paired experimental-computational pipeline. Initially, selected IgA Fc phage variants and WT clones were validated for their ability to bind to FcαRI. Then, the dual-site IgA Fc phage library was sequenced with nanopore (`2025_07_31.pod5`) and MGI short-read (`UDP0057.fq`) platforms. Model outputs were compared based on read retention, read quality, error modes (indels, substitutions, noise, etc.), and variant concordance with the short-read reference. No single metric provided a deterministic guarantee of performance. For example, a model might retain many reads after the length and alignment filters, but fail to cover the DMS region. Alternatively, it might reduce error rates by collapsing rare yet genuine variants. The benchmark was structured to reward models that preserved more usable reads, adhered to the dual-site IgA Fc DMS library constraints, and accurately reconstructed the population structure observed in the short-read reference data using nanopore test data. Training and evaluation datasets were kept separate. Training datasets were collected in separate sequencing runs from the test DMS library. Concordance was assessed against an orthogonal ground-truth dataset rather than relying solely on reference-based metrics.

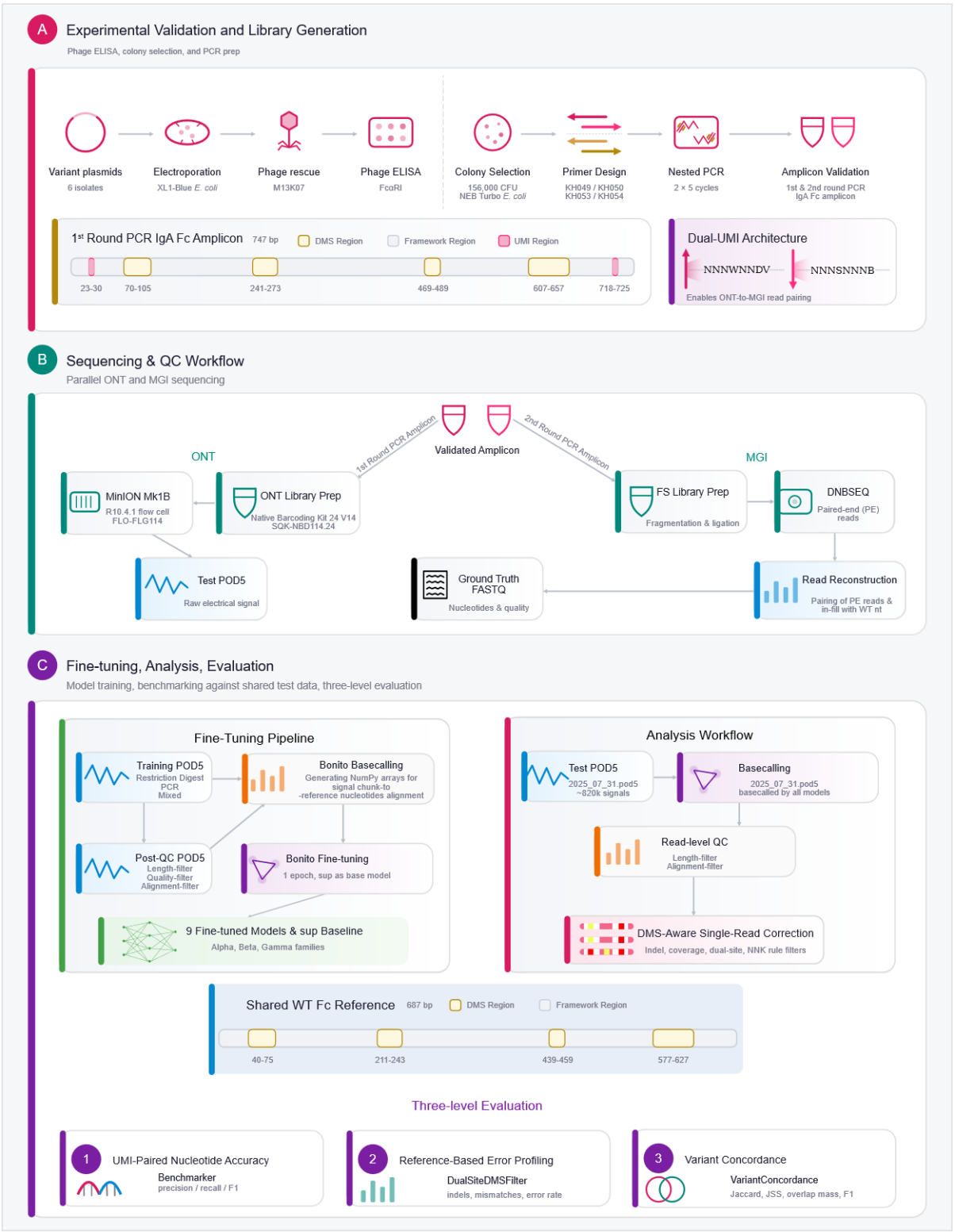

Figure 6. Overview of the experimental and computational study design. *Schematic of the three-stage workflow used to evaluate fine-tuned basecalling models for single-read variant detection in an IgA Fc phage-display library. (A) Sandwich ELISA characterised selected IgA Fc phage variants from a prior enrichment screen against FcαRI to confirm receptor-binding activity. The enriched second-round library was independently recovered by colony selection from glycerol stocks using NEB Turbo E. coli on eight chloramphenicol selection plates. (B) The IgA Fc fragment was amplified by two rounds of nested PCR incorporating dual UMI sequences (Sections 2.1.6, 2.1.8, and 2.1.9). First-round PCR products were sequenced on an ONT MinION Mk1B (R10.4.1 flow cell, SQK-NBD114-24); second-round products were sequenced on an MGI DNA-nanoball platform by LIC. (C) Nine derivative basecalling models were fine-tuned using WT IgA Fc nanopore runs generated by Dr Kyrin Hanning in 2023. All models were evaluated on the held-out test data (`2025_07_31.pod5`; 1st round PCR IgA Fc*

product of dual-site DMS library), with the short-read MGI reference dataset (`UDP0057.fq`; 2nd round PCR IgA Fc product of dual-site DMS library) serving as the orthogonal ground truth.

Before library preparation, six variants selected from the second round of the dual-site IgA Fc phage library screening, along with an IgA Fc WT phage, were evaluated in the laboratory to confirm display of active and functional Fc-pIII fusion protein against FcαRI. The ELISA curves in Figure 7 show that two out of six phage variants maintained strong, concentration-dependent binding to FcαRI, closely matching the WT control at the highest tested concentration of 1.7×10^{10} CFU/mL (normalised to the phage titre for each sample). A11 yielded a well-resolved five-parameter logistic fit with $R^2 = 0.962$, while B03 exhibited an even more precise fit with $R^2 = 0.999$. Neither variant surpassed the WT signal ceiling, consistent with preserved receptor engagement rather than with the identification of affinity-enhancing mutations. All four other variants displayed no or minimal binding to FcαRI. The fitted curves visually seem exponential, unlike the most common sigmoidal form for the ELISA response, primarily due to the low concentration range tested. Nevertheless, these results indicate that the library subjected to sequencing remained biologically active, expressing IgA Fc variants capable of binding the target receptor. Consequently, downstream sequence analysis could be interpreted as reflecting variation within a functionally enriched context rather than noise from inactive background variants.

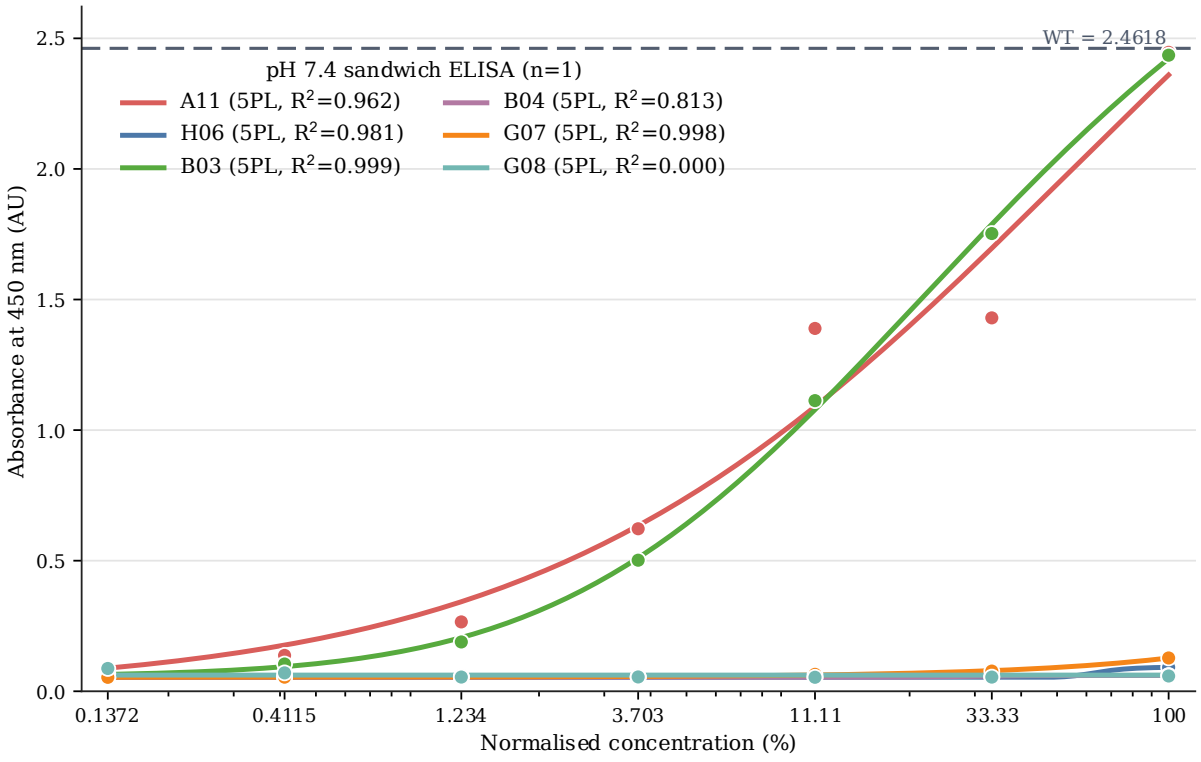

Figure 7. ELISA binding curves for selected IgA Fc phage variants and WT against FcαRI. *Binding curves showing the binding of six selected IgA Fc phage variants to immobilised recombinant FcαRI, assessed by sandwich ELISA across a*

normalised dilution series. Each colour represents one variant, and the dashed horizontal line marks the WT control signal. Each point represents the mean measured absorbance at 450 nm from technical duplicates. Curves were fitted using a five-parameter logistic (5PL) regression model. For relevant protocols used, see Sections 2.1.1–2.1.3.

Four oligonucleotide primers were custom-designed and synthesised by Integrated DNA Technologies (IDT). As the original IgA Fc phage library lacked dual-UMI tags around Fc fragments, a two-round nested PCR strategy was implemented. 1st PCR primers incorporated ambiguous nucleotide sequences in the UMI region (Figure 8), enabling the generation of amplicons containing dual-UMI tags with sufficient diversity to achieve a one-to-one linkage between dual-UMI tags and Fc variants. In practice, each Fc variant could be represented by multiple amplicons, and the association between dual-UMI tags and variant sequences diminished across consecutive PCR cycles, even with a high-fidelity polymerase such as Q5. Therefore, only five PCR cycles were conducted in both the 1st and 2nd rounds to minimise the accumulation of PCR errors. To offset the limited number of cycles, a high amount of template DNA was used (200 ng plasmid per reaction in the 1st round PCR; 200 ng 1st round PCR amplicon per reaction in the 2nd round PCR).

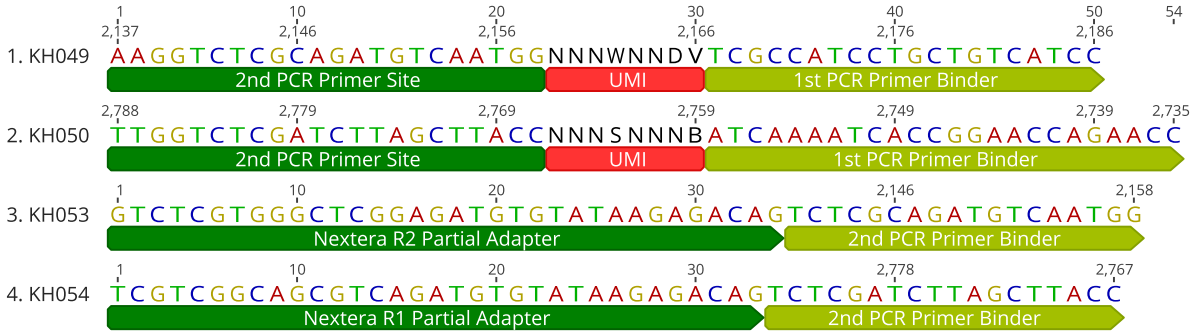

Figure 8. Primers used in two-round nested IgA Fc PCR for dual-UMI amplicon generation & sequencing. *Diagram illustrating the binding positions and functional components of the four oligonucleotide primers used to prepare IgA Fc amplicons for nanopore and MGI sequencing. Rows 1 and 2 show 1st round PCR primers KH049 and KH050, including the degenerate dual-UMI sequences NNNWNNNDV and NNNSNNNB to enable direct molecule-to-molecule benchmarking between platforms; rows 3 and 4 show 2nd round PCR primers KH053 and KH054, which anneal to the terminal regions of the first-round PCR amplicon, adding partial Nextera R1 and R2 adapter sequences required for downstream MGI adapter ligation. A primer table is provided in Appendix A.*

The initial experimental approach utilised *SfiI* for restriction digestion followed by gel excision to isolate the Fc fragment from the variant plasmid prior to the 1st round PCR. Digest controls performed as anticipated, with the undigested backbone appearing as the dominant high-molecular-weight band and the digested sample resolving into the expected fragment pattern, as shown in panel A of Figure 9. Two digested bands were observed at approximately 700 nucleotides (the Fc digest fragment is 729 bp between the two SfiI sites; lanes 2 and 3). These bands were excised from the gel, purified, and quantified using NanoDrop. The left band yielded a concentration of 21.365 ng/µL, while the right band had a concentration of 18.277

ng/µL. In total, 10 µL of purified eluate was obtained. The A260/A230 and A260/A280 ratios were both well below their optimal values of ~2.0-2.2 and ~1.8, respectively, indicating the presence of protein contamination and organic contaminants including EDTA. Even if the entire 10 µL consisted of a high-purity Fc fragment, an additional concentration step would have been necessary to achieve sufficient template DNA input, and the total yield would have supported only a single PCR reaction.

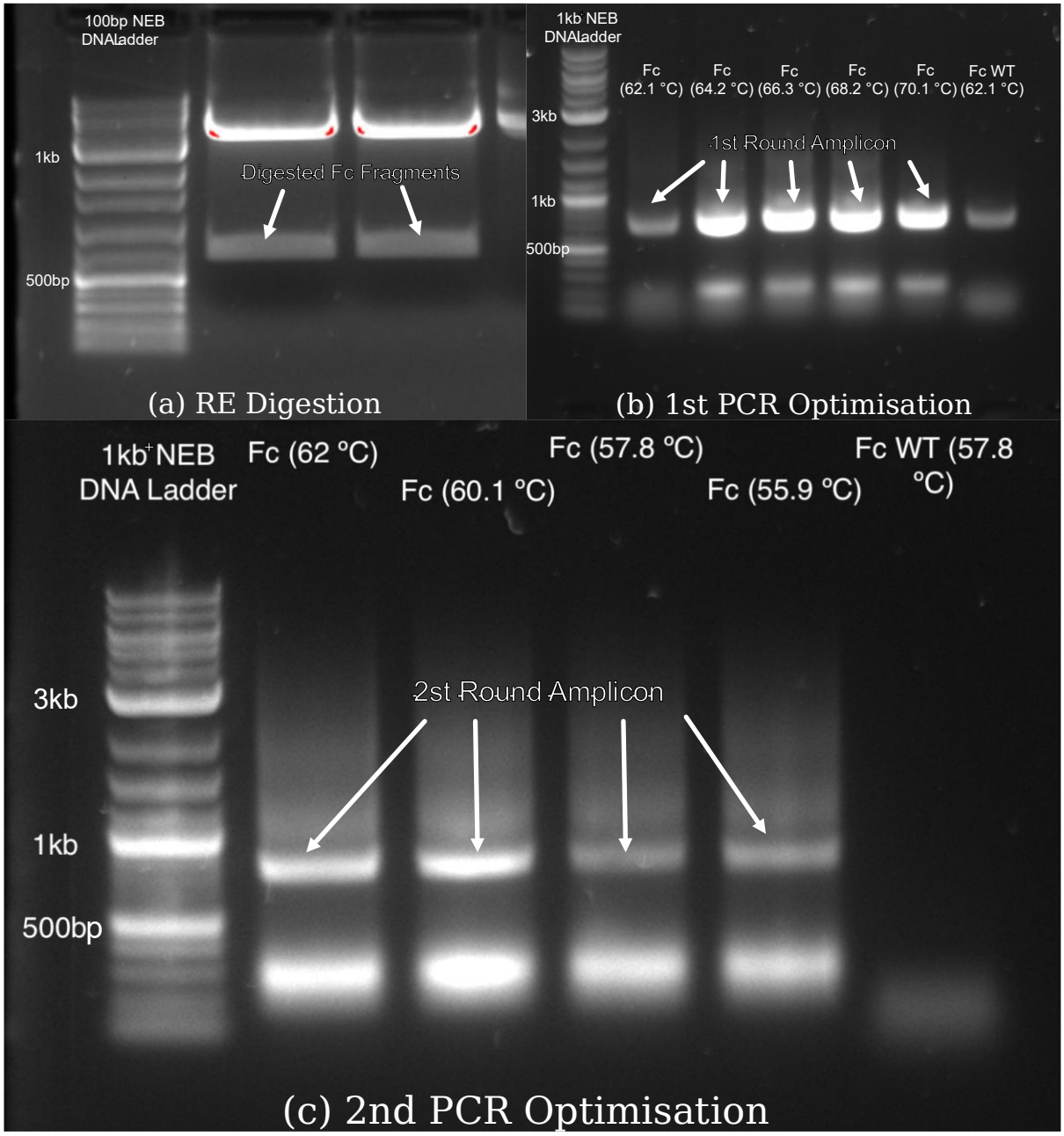

Figure 9. Agarose gel electrophoresis of restriction digestion and PCR product during IgA Fc library preparation. *(A) Restriction-digested Fc fragment from the phagemid using SfiI enzyme; the labelled Fc bands were subsequently excised, purified, and the concentrations were confirmed using NanoDrop. (B) 1st round gradient PCR was used to optimise the annealing temperature for the KH049 and KH050 primers. The 1st round Fc amplicon (747 bp) is clearly observed from 64.2 °C to 70.1 °C. (C) 2nd round gradient PCR was used to optimise the annealing temperature for the KH053 and KH054*

primers. 2nd round Fc amplicon (806 bp) is clearly observed at low 60.1 °C-62 °C. For PCR protocol & conditions used, see Section 2.1.8 and Appendix B.

To simplify the experimental procedure, direct phagemid PCR amplification was implemented. Both 1st and 2nd round PCR reactions underwent annealing temperature optimisation using gradient PCR across a temperature range that included the predicted annealing temperatures for the forward and reverse primers, as estimated by the IDT OligoAnalyzer tool. Optimal annealing temperatures of 68 °C and 60 °C were established for the 1st and 2nd PCR rounds, respectively. At these temperatures, the 1st round PCR yielded 16 µL of product (pooled from four reactions) at a concentration of 59.1 ng/µL after EXO-CIP treatment and bead cleanup. DNA quality was confirmed by near-optimal A260/A230 (2.042) and A260/A280 (1.904) ratios. The DNA yield was sufficient to proceed to the 2nd round PCR without further amplification or concentration. The 2nd round PCR utilised the purified amplicon from the 1st round PCR as input DNA, without dilution. Similarly, 16 µL of purified product at 59.2 ng/µL was obtained after EXO-CIP treatment and bead cleanup. DNA quality remained high, with A260/A230 and A260/A280 ratios of 1.915 and 1.878, respectively. These results collectively confirmed that the IgA Fc phage library contained the intended functional Fc fragment.

Subsequent steps in the experimental workflow included in-house nanopore sequencing of the 1st round PCR amplicon using an ONT MinION Mk1B device at room temperature. The ONT Native Barcoding Kit 24 V14 (SQK-NBD114-24) facilitated ligation of native barcodes and sequencing adapters. A total of 200 fmol of the 1st round PCR amplicon (747 bp), equivalent to 97 ng of input DNA, was barcoded with barcode 1. The sequencing run lasted over 24 hours and generated 827,519 raw nanopore signals.

Library preparation and sequencing of the 2nd round PCR amplicon were performed on a DNA-nanoball platform (MGI Tech) in collaboration with Livestock Improvement Corporation (see Section 2.1.9). Sequencing yielded a total of 1,694,634 `FASTQ` reads after pairing and polishing of R1 and R2.

Five datasets were used in this thesis (Figure 10). The restriction digest dataset consisted of nanopore signals from restriction-digested WT Fc fragments, while the PCR dataset included nanopore signals from PCR-amplified WT and variant Fc amplicons. Both datasets were previously sequenced by Dr Kyrin Hanning in 2023. The mixed dataset combined the restriction digest and PCR datasets, with the signal orders shuffled to enhance homogeneity. These three datasets were used to fine-tune the `sup` v5.2.0 model. The test dataset consisted of

the 1st round PCR product sequenced by nanopore with dual-UMI tags. The ground-truth dataset consisted of the orthogonal 2nd round PCR product sequenced on an MGI DNA-nanoball platform. An initial quality analysis of each dataset was conducted during thesis preparation; an overview is provided in the remainder of this section.

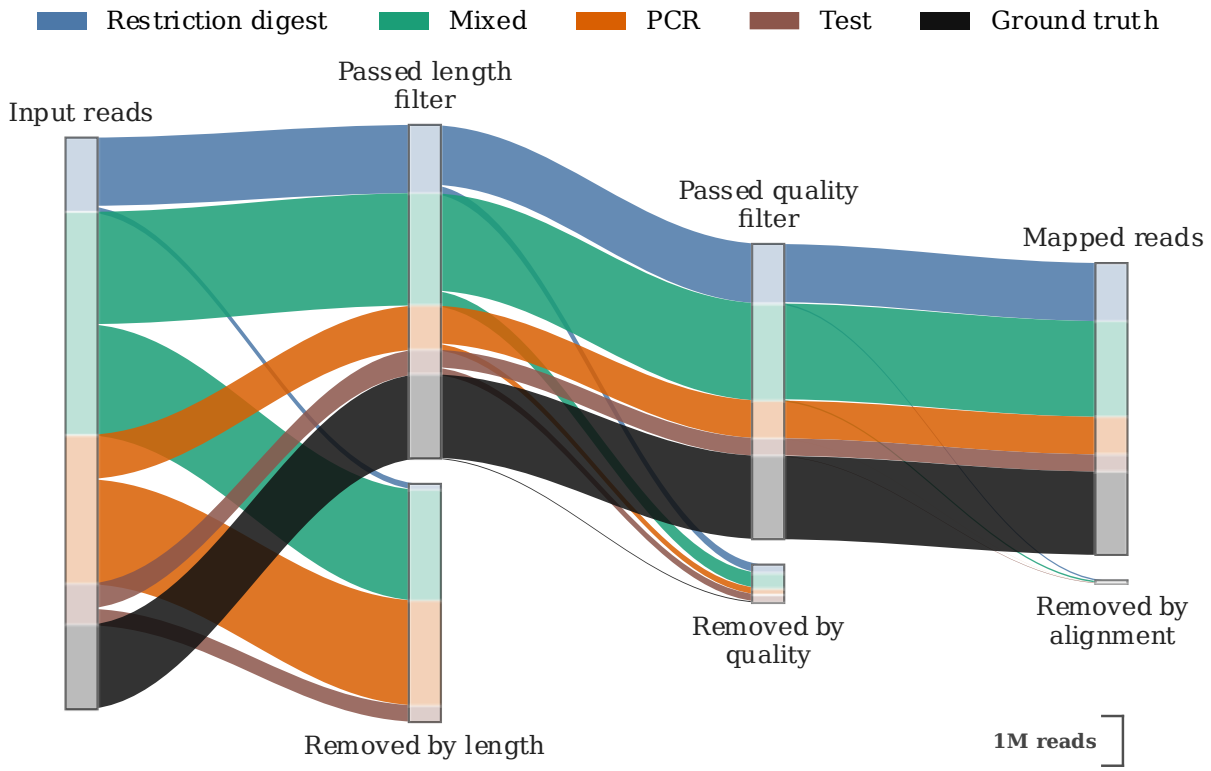

Figure 10. Read retention evaluation across QC stages for all datasets. *Sankey diagram summarising read retention for all five datasets through the initial QC pipeline. Flows show all input reads and the fractions passing the length, quality, and alignment filters for the Restriction digest (`A_full.pod5` – WT Fc fragment isolated by SfiI gel excision), PCR (`B_full.pod5` – WT and variant Fc amplicon produced from PCR), Mixed (`AB_full.pod5` – combined A and B), Test (`2025_07_31.pod5` – held-out IgA Fc variant library sequencing run), and Ground truth (`UDP0057.fq` – MGI short-read sequencing of the same held-out library) datasets. All datasets were basecalled with the baseline `dna_r10.4.1_e8.2_400bps_sup@v5.2.0` model before filtering; the length filter retained reads 600-800 nt, the quality filter retained reads with per-read Q-score >= 15, and the alignment filter retained reads with a primary alignment to the Fc reference. A full quality-control summary is available in `data.csv` on GitHub (Appendix E).*

The analysis applied read-level filtering, including a length filter (retaining reads between 600 and 800 bp), a quality filter (retaining reads with per-read Q-score ≥ 15), and an alignment filter (retaining reads with a primary alignment to the Fc reference). Figure 10 shows that the restriction digest dataset retained 1,168,677 of 1,493,513 reads (78.2%) after filtering through the complete pipeline, while the larger PCR-derived dataset retained only 756,314 of 2,995,685 reads (25.2%). The combined mixed dataset exhibited an intermediate retention rate, and the held-out nanopore test set retained 41.7% of reads. On the other hand, the MGI ground truth remained nearly intact after processing, with 98.9% of reads retained (i.e. an initial round of

filtering was already done during R1 and R2 short-read pairing). The substantially higher read recovery rate observed in the restriction digest dataset compared to the PCR dataset is expected, given the absence of PCR-introduced error. This difference had downstream implications for model training performance, as discussed in subsequent sections. Furthermore, the comparison between datasets generated by different experimental procedures indicates that most attrition resulted from nanopore read quality and sequence integrity, rather than from overly stringent downstream filtering. This finding underscores the necessity of comparing model performance across a range of metrics that reflect biological plausibility and concordance, rather than relying on the absolute number of raw reads produced.

Per-read Q-score distributions were plotted for each dataset to provide a more comprehensive assessment of dataset quality (Figure 11). The restriction digest, PCR, and mixed datasets entered the QC pipeline with comparatively strong quality distributions. In contrast, the held-out test dataset exhibited a broader, lower-quality tail prior to filtering. Specifically, only 58.4% of reads in the test dataset achieved Q20 or higher, compared with 72.7% in the restriction digest dataset and 74.2% in the PCR dataset. This discrepancy likely results from the use of a fresh nanopore flow cell for the restriction digest and PCR datasets, whereas the test dataset utilised an older flow cell.

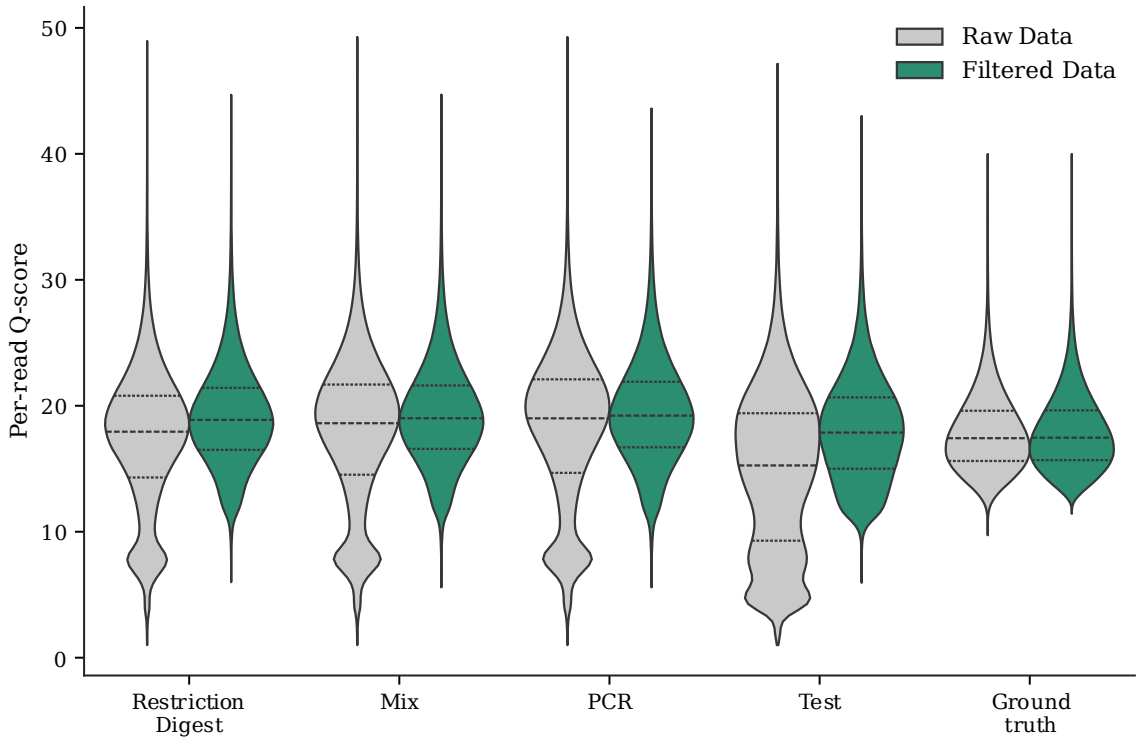

Figure 11. Violin plots comparing the distribution of per-read Q-scores before and after quality control by dataset*. For each dataset, the grey violin shows raw basecalled reads and the green violin shows reads retained after length, quality, and*

alignment filtering; violin widths are proportional to the local density of observations at each Q-score value. Horizontal dashes indicate the 25th, 50th, and 75th percentiles. Per-read Q score was calculated by converting per-base quality scores to error probabilities, taking the arithmetic mean, then log-transforming back to the Q-score scale. A full quality-control summary is available in `data.csv` *on GitHub (Appendix E).*

During the initial stage of model fine-tuning, only `Alpha`, `Beta`, and `Gamma` models were trained on the restriction digest, PCR, and mixed datasets, respectively. A total of 27,063, 57,394, and 84,517 chunks of signals (each chunk consists of 12,000 signal samples with corresponding reference nucleotide labels) were generated during the `Bonito` basecalling process for training. Preliminary testing indicated that `Alpha` achieved the highest performance despite having the smallest training dataset (see Section 3.2 for further discussion). This result prompted an investigation into whether training data quality exerts a greater influence than quantity on nanopore model fine-tuning. To test this hypothesis, an additional six models were tested: `AlphaQCL`, `AlphaQCH`, `BetaQCL`, `BetaQCH`, `GammaQCL`, and `GammaQCH`. QCL models were trained on post-quality-control nanopore datasets containing only passed reads; QCH models were trained on the same passed reads with stricter alignment-quality requirements (see Section 2.2.4). Analyses in subsequent sections demonstrate that the experimental procedure used to collect the training data had a greater impact on downstream model performance than the computational quality-control pipeline applied to the data.

3.2 Basecalling and Correction Performance Across Models

The primary objective of the section was not to evaluate the model solely by error rate, but to determine which model produced data that passed filtering, adhered to the dual-site DMS design, and accurately preserved the ground-truth variant landscape, thereby enabling robust analysis of the variant-calling process. This distinction is critical because minor basecalling errors can have disproportionately large downstream effects in amplicon sequencing: for example, a noisy signal segment, a single insertion or deletion, or a few substitutions may generate false-positive low-abundance variants, distort the relative abundance of true variants, or cause an otherwise usable read to fail the DMS filter. Consequently, models were evaluated collectively across metrics of retention, error rate, post-alignment quality, and variant concordance with ground truth.

Read retention following quality control clearly distinguished model performance across families (Figure 12). The `Alpha` family consistently preserved the largest proportion of usable reads: `AlphaQCH` achieved the highest DMS-pass rate at 43.6% (237,170 reads), followed by `Alpha` at 38.7% (253,187 reads) and `AlphaQCL` at 37.0% (234,161 reads). `Gamma` models formed an intermediate tier, retaining 28.5–29.9% of reads, while the `Beta` family retained 22.2–27.4%. The pretrained `sup` model was the least efficient, with only 14.3% of reads (65,509) passing the DMS filter. Relative to the `sup` model, the fine-tuned models converted between 1.62 times (`BetaQCL`) and 3.86 times (`Alpha`) more raw signal into interpretable reads, demonstrating that domain adaptation substantially increases usable yield beyond what the pretrained model obtained. Short-read ground truth achieved a DMS-pass rate of 41.5% (703,635 reads), exceeded by `AlphaQCH`, with `Alpha` and `AlphaQCL` following immediately after. However, this does not mean `AlphaQCH` is better than short-read ground truth at recovering the variant landscape; in fact, the discrepancy stems from `AlphaQCH` overfitting, as discussed later in this chapter. These retention gains were not simply a product of training on quality-controlled data. Within each model family, QC-trained models did not consistently outperform their base counterparts. Although `AlphaQCH` achieved the highest DMS-pass rate within the `Alpha` family, `GammaQCL` and `BetaQCH` showed no comparable advantage over their respective base models. This pattern suggests that the quality of the learned signal representation for IgA Fc sequences, rather than a conservative decoding bias introduced by cleaner training data, was the primary driver of differences in retention.

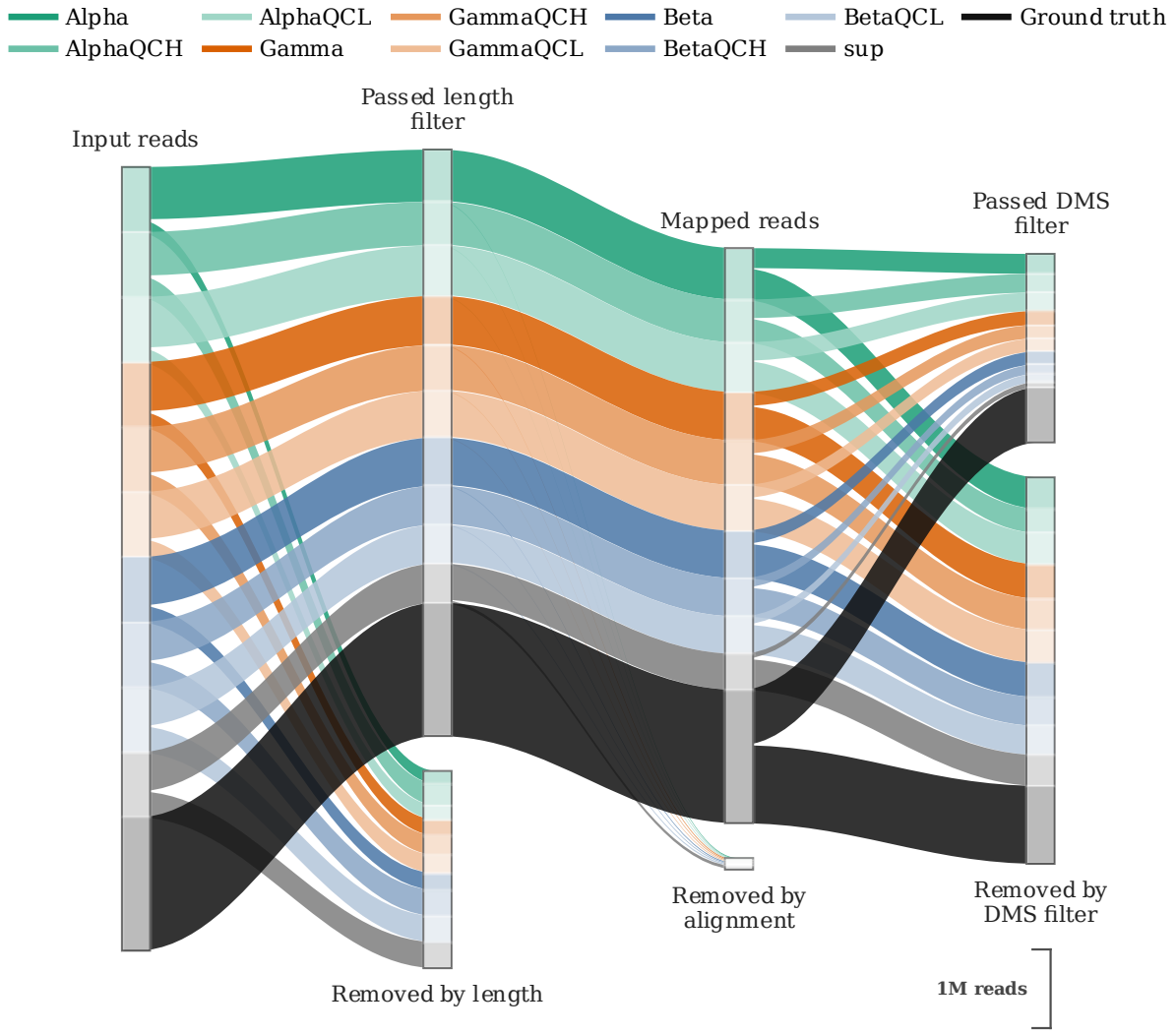

Figure 12. Read retention evaluation across filtering stages for different basecalling models and ground truth. *Sankey diagram summarising read retention for the held-out test dataset after basecalling with each evaluated model. Flows show reads that pass the 600-800 nt length filter, the primary-alignment filter, and the DMS filter for all basecalling models and the ground truth. The DMS filter retained reads satisfying the dual-site DMS filter constraints: no more than three indel events or three indel bases, full coverage across the analysed DMS regions, no more than two mutated codons, and compatibility with the NNK mutational pattern. For a complete description of the filter methodology, see Section 2.2.5. Models are labelled by family (`Alpha`, `Beta`, `Gamma`), corresponding to the source training dataset (A_full, B_full, AB_full, respectively) and quality-control level (no suffix = unfiltered; QCL = length + quality + primary-alignment filtered; QCH ≥ 99% alignment-identity filtered). For fine-tuning details, see Section 2.2.2 and Appendix D. The full test summary is available in `test.csv` on GitHub (Appendix E).*

Pipeline pass rate accounts for length and alignment filters, further reinforcing this interpretation. Base models consistently achieved the highest pipeline pass rates within each family: `Alpha` (30.63%) > `AlphaQCH` (28.72%) > `AlphaQCL` (28.32%), `Beta` (19.91%) > `BetaQCH` (14.38%) > `BetaQCL` (12.84%), and `Gamma` (21.43%) > `GammaQCH` (20.55%) > `GammaQCL` (20.01%), all well above the `sup` baseline of 7.92%, and approaching closer to ground-truth pass rate (41.5%). Because pipeline pass rate captures the full set of applied filters, it provides a more comprehensive measure of model performance than DMS pass rate alone. However, it cannot

isolate the contribution of the dual-site DMS library constraint specifically. Taken together, these results indicate that the most effective models learned signal and sequence features that genuinely minimise violations of library design constraints, rather than merely shifting the balance of passing and failing reads.

Length retention provided a further dimension of differentiation. `Alpha` family models produced read length distributions mostly concentrated within the expected 600–800 bp range, retaining 80.33% of reads through the length filter, substantially higher than both the other fine-tuned families and the `sup` model (60.04%). Within all families, base models consistently outperformed their QC-trained counterparts on length retention, and the near-identical performance of `BetaQCL` (59.83%) and `BetaQCH` (59.96%) relative to the `sup` baseline indicates that QC-filtered training data conferred little benefit for length generalisation in that family. This consistent advantage of base models suggests that training on the full, unfiltered signal distribution supports broader generalisation to natural variation in read length, rather than overfitting to the length characteristics of the filtered training set.

Figure 13 presents the correlation between evaluated metrics derived from the primary-alignment subset of the alignment `BAM` file and the primary-aligned reads (`FASTQ`). In Panel A, the `Alpha` model achieved the lowest primary overall error rate at 2.98%, outperforming `AlphaQCH` (3.37%) and `AlphaQCL` (4.40%). The `Gamma` and `Beta` families exhibited higher noise levels, with error rates clustering between 5.58% and 8.29%, while the pretrained `sup` model performed worst at 8.95%. These findings challenge the assumption that more aggressive fine-tuning (in terms of data size and cleanliness) within a model family consistently improves read quality. The most effective model was not the one with the lowest validation loss (`AlphaQCH`) or the lowest training loss (`AlphaQCL`), but rather `Alpha`, which most effectively controlled the error modes that disrupt variant calling: false-positive mutations, indels, and coverage distortion, in the dual-site IgA Fc library. An ordinary least squares (OLS) regression across basecalling models yielded an R^2 of 0.905, confirming the expected strong inverse relationship between primary overall error rate and DMS pass rate: lower basecalling noise increases the likelihood that a read satisfies the dual-site IgA Fc DMS library design constraints.

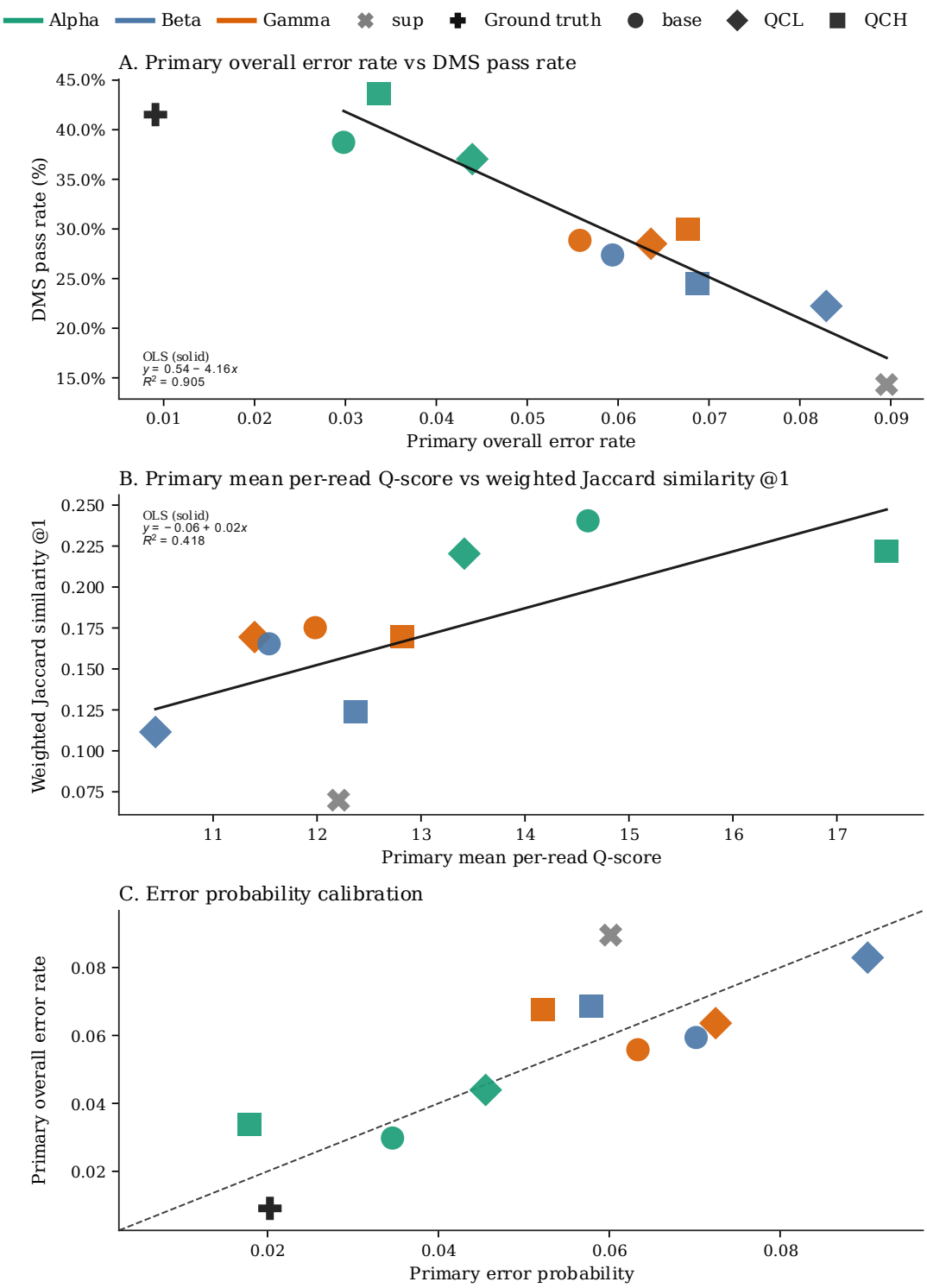

Figure 13. Comparison of post-alignment metrics of model basecalls on the held-out test dataset. *All three scatter plots compare model performance across different dimensions. Panel A correlates a lower error rate with how well the reads meet the dual-site DMS library design constraints. Panel B correlates model confidence with concordance against the MGI ground truth. Panel C correlates model confidence against true error rate. All metrics compared are computed from the primary-alignment subset of the alignment* `BAM` *file or the primary-aligned reads (*`FASTQ`*). Each point corresponds to one nanopore model output or the MGI ground truth; colour encodes model family, and marker shape encodes training subset type (base, QCL, or QCH). The primary overall error rate calculated from primary alignments, which was preferred over the mismatch rate because indel- and homopolymer-based errors are common in nanopore sequencing. DMS pass rate is the proportion of primary-aligned reads passing the DMS. The primary mean per-read Q-score was calculated using the primary aligned reads by first computing per-read Q-scores, then power-transforming them to error probabilities, taking the arithmetic mean across all reads, and finally log-transforming back to the Q-score scale. Weighted Jaccard similarity @1 was computed from non-*

WT variant tables after applying a minimum count threshold of one. Weighted Jaccard similarity is a count-weighted measure of variant-set overlap; a value of 1.0 indicates perfect agreement in both variant identity and relative abundance. For weighted Jaccard similarity, each model output is compared against the ground truth. Primary error probability was calculated from primary aligned reads by power-transforming the primary mean per-read Q-score to obtain the probability. Ordinary least squares (OLS) regression was used to fit a solid line across nanopore models for Panels A and B (ground truth excluded). The dashed line is drawn along the y=x axis for panel C to highlight the ideal output from a perfectly calibrated model. For a full description of the metrics, see Section 2.2. Full test summary is available in `test.csv` *on GitHub (Appendix E).*

Panel B reveals a dissociation between per-read quality scores and downstream reliability. `AlphaQCH` achieved the highest primary mean per-read Q-score (17.48), followed by the ground truth and `Alpha`. However, several models, including `sup`, `Gamma`, and `Beta` families, have reasonable raw Q-scores yet significantly underperformed in concordance with the ground truth. This suggests that Q-scores alone are insufficient indicators of downstream performance. During variant calling, abundance distortion and DMS-window violations were as influential as nominal per-base confidence; a model may assign high-quality scores to a systematically biased reconstruction of the library, and the downstream benchmark penalises such outcomes. The lower OLS R^2 (0.418) reflects the fact that weighted Jaccard similarity depends on multiple factors beyond model confidence, including the variant-calling count threshold, abundance distribution, error context in DMS regions, and family-specific biases (discussed later). Excluding the `sup` model, which deviates substantially below the fitted trend, increases R^2 to 0.567.

Panel C evaluates model calibration by comparing the primary overall error rate against the primary estimated error probability. A perfectly calibrated model lies on the y = x diagonal. Although `AlphaQCH` achieved the lowest estimated error rate, it exhibited a moderately higher overall error rate, indicating inferior calibration relative to `Alpha` and `AlphaQCL`. All other model families performed worse in both dimensions. Nevertheless, all fine-tuned models were better calibrated than the `sup` baseline and achieved calibration comparable to the short-read ground truth. This outcome is expected: MGI quality scores derive from fluorescent signal intensities via a proprietary signal-to-error model, whereas ONT Q-scores are based on basecalling probabilities and are empirically rescaled; fine-tuning closes this calibration gap.

The calibration diagonal also partitions the plot into two regimes. Models above the line underestimate their true error rate (estimated error probability lower than actual), while models below overestimate it. Overestimation is generally less problematic for variant calling, as it reduces true-positive variants, while underestimation retains false-positive variants. In fact, it is possible to view slight overestimation as an advantage, as it makes the model more reliable, especially in scenarios where a ground truth is not readily available. By this criterion, `Alpha` is

the best-performing model, closely followed by `AlphaQCL`, though the MGI ground truth outperforms all models by a moderate margin.

Combining all three panels makes it clear that `AlphaQCH` showed signs of overfitting to the Fc WT training data. Despite achieving the highest mean per-read Q-score, it was less well calibrated than the other `Alpha`-family models and exhibits lower weighted Jaccard similarity than `Alpha`, yet a higher DMS pass rate. These symptoms are consistent: `AlphaQCH` basecalls false-positive nucleotides at an elevated rate, artificially inflating both the Q-score and the proportion of reads that pass the DMS library design filter.

In summary, Figures 12 and 13 illustrate that basecalling improvement is multidimensional. The desirable operating region differs by axis: low error rate with high DMS pass rate (Panel A, upper-left), high Q-score with high weighted Jaccard similarity (Panel B, upper-right), and low error probability with low error rate (Panel C, lower-left). Across all panels, `Alpha`-family models generally perform best, followed by `Gamma` and then `Beta`. This ranking reflects the training data: `Alpha`-family models were trained on restriction-digested IgA Fc WT fragments free of PCR error, `Beta`-family models on the PCR dataset, and `Gamma`-family models on a mixture that diluted high-quality restriction-derived features with noisier PCR-derived signals. No single axis captures the full complexity; the decision boundary between strong and weak models depends on whether the priority is raw sequence accuracy, retention of biologically plausible reads, or preservation of the ground-truth abundance distribution.

Figure 14 displays a z-score-normalised heatmap comparing all basecalling models against the ground-truth dataset. The `sup` model underperformed every fine-tuned model across all metrics, with the most pronounced deficit in DMS pass rate: `sup` achieved 14.32%, 2.71 times lower than `Alpha`'s 38.82%. This gap propagated into variant recovery: `sup` identified 46,573 unique variants from 65,509 DMS-pass reads, whereas `AlphaQCH` identified 117,476 unique variants from 253,187 DMS-pass reads. The unique-variant-to-DMS-pass-reads ratio further distinguishes the models: `sup` yielded 0.711 compared with `Alpha`'s 0.464, indicating that `Alpha` not only recovered more distinct variants but also achieved substantially higher read depth per variant. This enhanced capacity to denoise signals and retain reads that would otherwise be excluded by length and DMS filters was consistent across all fine-tuned models, suggesting that even relatively low-quality training data can improve basecalling performance in domain-specific contexts. All fine-tuned models also demonstrated significantly lower framework nucleotide mismatch rates than both the `sup` baseline and the ground truth, a notable result

given that the `sup` model was noisier than the ground truth, yet even the weakest fine-tuned models reversed this trend. These improvements collectively highlight the adaptability of the `sup` architecture to domain-specific signal features when fine-tuned, even with minimal or low-quality data. Crucially, the results also argue against simple memorisation of training data. If the models had memorised the WT Fc reference sequence, neither the number of unique variants nor the concordance scores would be expected to increase; instead, both metrics would likely decline.

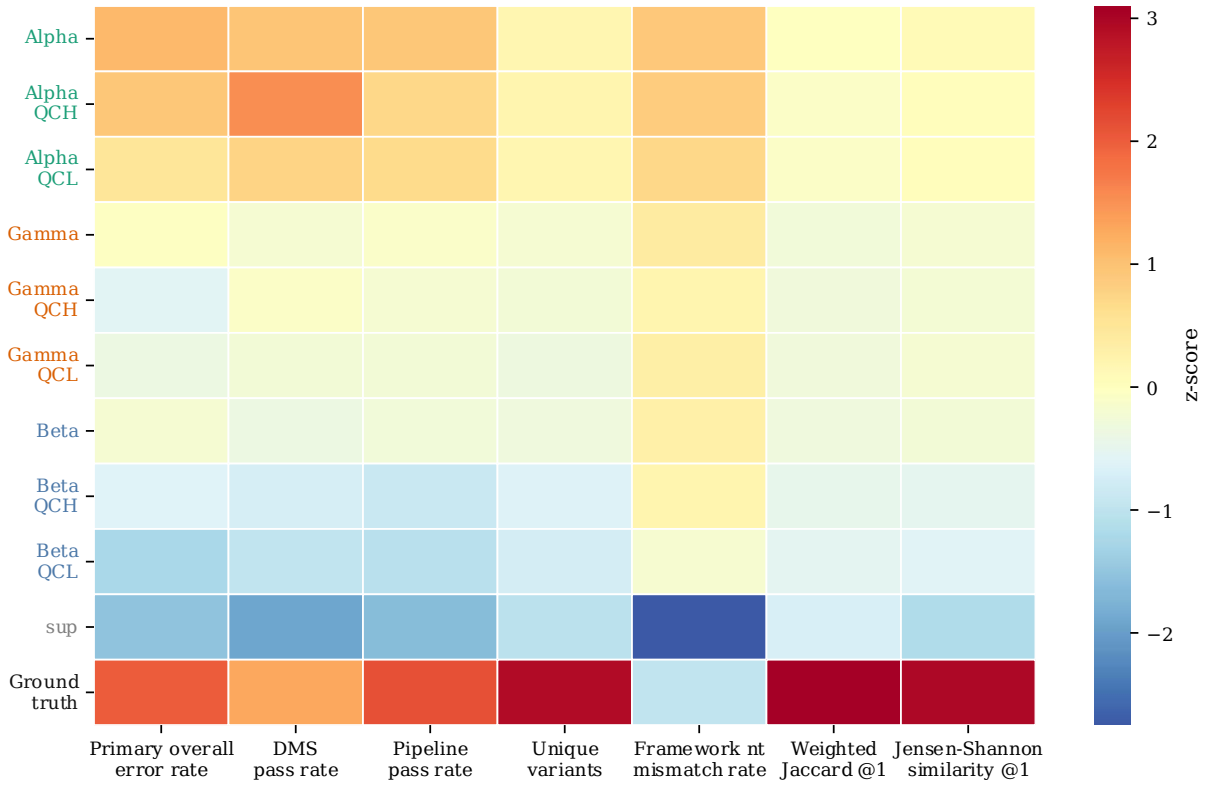

Figure 14. Multi-metric basecalling/variant-calling performance comparison. *Z-score normalised heatmap comparing multiple performance metrics across the evaluated model outputs and the short-read ground truth. Columns are primary overall error rate, DMS pass rate, pipeline pass rate, unique variants, framework nucleotide mismatch burden, weighted Jaccard similarity @1, and Jensen-Shannon similarity @1. Each cell is a z-score computed across the given read set for the corresponding metric, allowing metrics with different scales to be compared on a common standardised scale. For concordance measures (Weighted Jaccard similarity@1 and Jensen-Shannon similarity @1), the model basecalls are compared against the ground truth, and the ground truth is compared against itself in the last row. Full test summary is available in* `test.csv` *on GitHub (Appendix E).*

The final analysis in this section examined how validation loss affects concordance with the ground truth, constraints on the dual-site IgA Fc library, and the overall error rate (Figure 15). Validation loss was the main diagnostic tool during model fine-tuning, guiding the choice of the learning rate. All models were fine-tuned for one cycle (epoch) using the full dataset. The learning rate was adjusted to reduce validation loss while keeping a slightly higher training loss to avoid overfitting. During fine-tuning, `Bonito` reserved approximately 10% of the available training chunks as an internal validation subset. This proportion follows a common machine-

learning train/validation split and was sufficient here because the datasets contained tens of thousands of chunks, leaving several thousand validation examples for loss estimation. The validation subset was randomly sampled from the same source data as the training subset, so it was expected to contain similar signal distribution, sequence context, and noise characteristics.

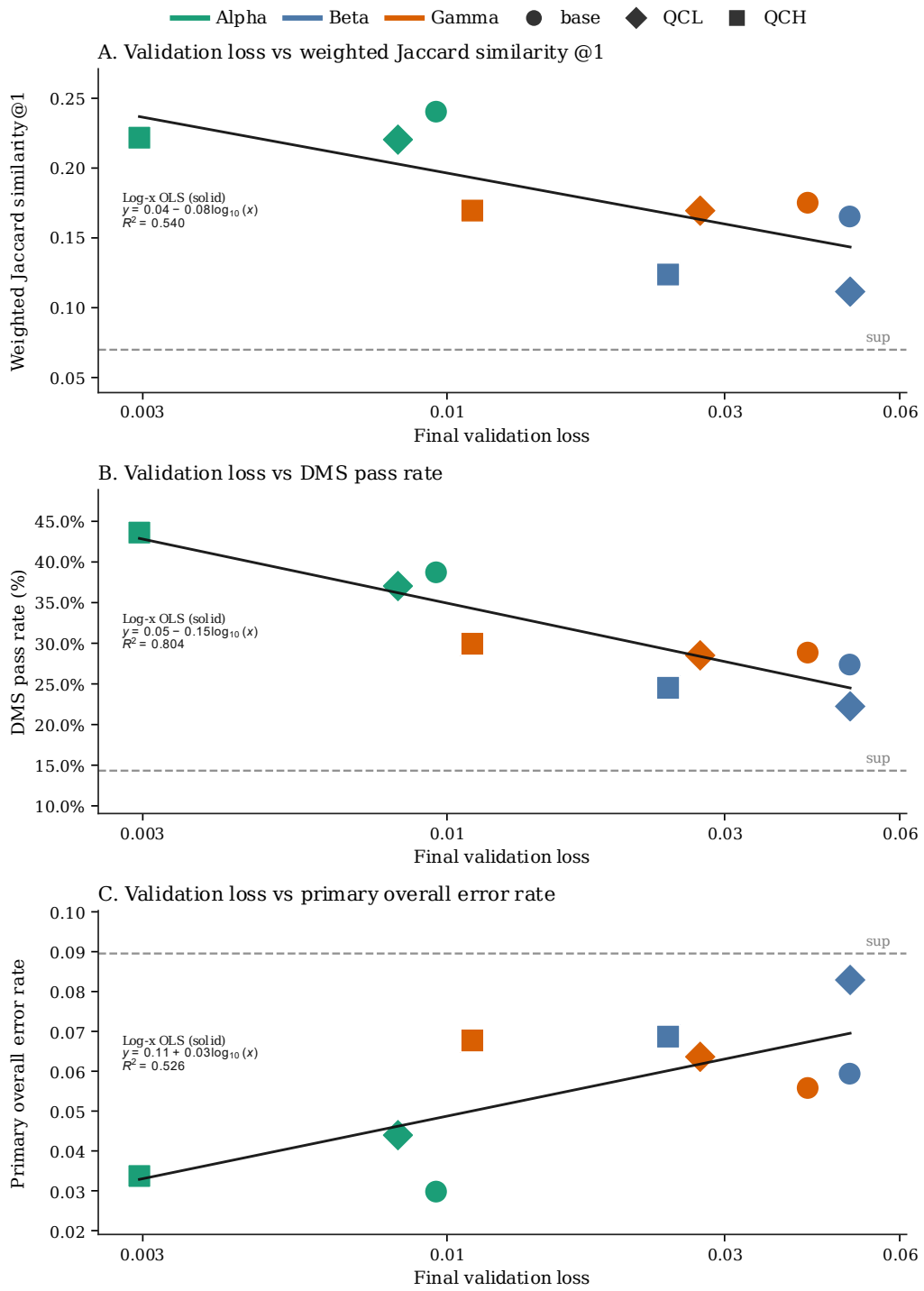

Figure 15. Correlation between model validation loss against downstream metrics on the held-out test dataset by model. *(A). Scatter plot of validation loss against weighted Jaccard similarity @1. (B). Scatter plot of validation loss against DMS pass rate. (C) Scatter plot of validation loss against primary overall error rate. Each point represents one model, coloured by family and shaped by training subtype, and the validation-loss axis is shown on a log scale. Validation loss was recorded from*

the archived `training.csv` file after each fine-tuning epoch. Lower validation loss indicates better signal-to-nucleotide prediction loss on held-out validation data (~10% of training data). Weighted Jaccard similarity @1 was computed from non-WT variant tables after applying a minimum count threshold of one. Weighted Jaccard similarity is a count-weighted measure of variant-set overlap; a value of 1.0 indicates perfect agreement in both variant identity and relative abundance. For weighted Jaccard similarity, each model's basecalls are compared against the ground truth. DMS pass rate is the fraction of primary aligned reads that passed the DMS filter/dual-site IgA Fc library. The primary overall error rate is composed of substitution, insertion, and deletion relative to the number of mapped bases against the Fc reference. It was calculated from primary alignments and was preferred over the mismatch rate because indel-based and homopolymer-based errors are common in nanopore sequencing. For a full description of the metrics, see Section 2.2. For relevant numerical summaries, see test.csv, correlation.csv, and model.csv on GitHub (Appendix E).

The heatmap in Figure 14 and the validation-loss comparison in Figure 15 show a key contrast. While training diagnostics, such as validation loss, indicated direct improvement, downstream benchmarking told a different story. `AlphaQCH` achieved the lowest validation loss within the `Alpha`-family, while `BetaQCH` and `GammaQCH` did so in their own groups. However, these reductions did not yield the best composite biological performance. More notably, the `Alpha` model, rather than `AlphaQCH`, had the strongest nanopore output across major concordance metrics (Weighted Jaccard and Jensen-Shannon similarities), the primary overall error rate, and the pipeline pass rate. Similarly, despite `BetaQCH` and `GammaQCH` having lower validation losses than their family baselines, neither outperformed its baseline in concordance with the MGI ground truth. This shows a clear contrast: optimising the training objective benefited the supervised signal-to-nucleotide task, but did not improve the nanopore error patterns most relevant for variant calling in a constrained mutational library. QCH models learned a cleaner but narrower signal distribution, yielding tidier reads, but lacked robustness for broader sequence-signal contexts and low-abundance variants in the test library.

In summary, Section 3.2 shows that fine-tuning improved model performance relative to the `sup` baseline in several ways, though gains varied across models. Domain-specific training consistently outperformed the pretrained `sup` model, even in the presence of intrinsic differences between IgA Fc WT data and dual-site IgA Fc variant data. This approach led to notable gains in read retention and in the generation of biologically plausible, concordant reads. The most robust results came from balanced `Alpha` family models rather than the more aggressively trained `Beta` and `Gamma` families. However, because validation loss alone did not reliably predict downstream variant-calling performance, Section 3.5 further examines whether the internal weight changes induced by fine-tuning provide a mechanistic explanation for why `Alpha` generalised better than the lower-loss `AlphaQCH` model.

3.3 Error Profiles and the Biological Consequences of DMS Filter

Understanding why reads pass or fail the DMS filter is essential for interpreting model behaviour and comparing models beyond basic metrics, such as model confidence, error rate, or the number of unique variants. The DMS filter reflected the constraints of the dual-site IgA Fc library design and was not chosen at random. It required that reads match the reference and that mutations be restricted to designated regions (Section 2.2.5). With this context, the section now focuses on what happens beneath the surface. The analysis examines the types of errors that occur before and after corrections, where they occur, and how they affect the basecalling model's ability to identify and count variants. In addition to error patterns, passed reads are analysed at various levels: dataset, read, codon, base, position, and mutation, giving a structured view of how fine-tuning affects basecalling at different levels.

Figure 16, Panel A, shows that polishing reduced error rates across all models and the ground truth, compressing the range from 0.91–8.95% to 0.51–0.59%. The magnitude of improvement scaled inversely with initial quality: the pretrained `sup` model improved by 93.4%, followed by `Beta` (90.7%), `Gamma` (90.0%), and `Alpha` (81.9%), while the ground truth dropped by only 0.44%. The ground truth's minimal improvement is expected, as the reads were paired-end merged with WT in-fill between forward and reverse reads. Hence, the intervening nucleotides align perfectly and do not contribute to the error rate before polishing. Despite this post-polishing convergence in overall error rate, Panel B reveals that the models still differed in error composition. Substitutions dominated the error profile of all nanopore models. However, insertions and deletions, though smaller in magnitude, carried disproportionate biological consequences: either error type can shift the reading frame, violate the DMS window, and cause a read to fail quality-control filters. Consequently, models with similar aggregate error rates after polishing still varied in their capacity to preserve biologically correct sequences.

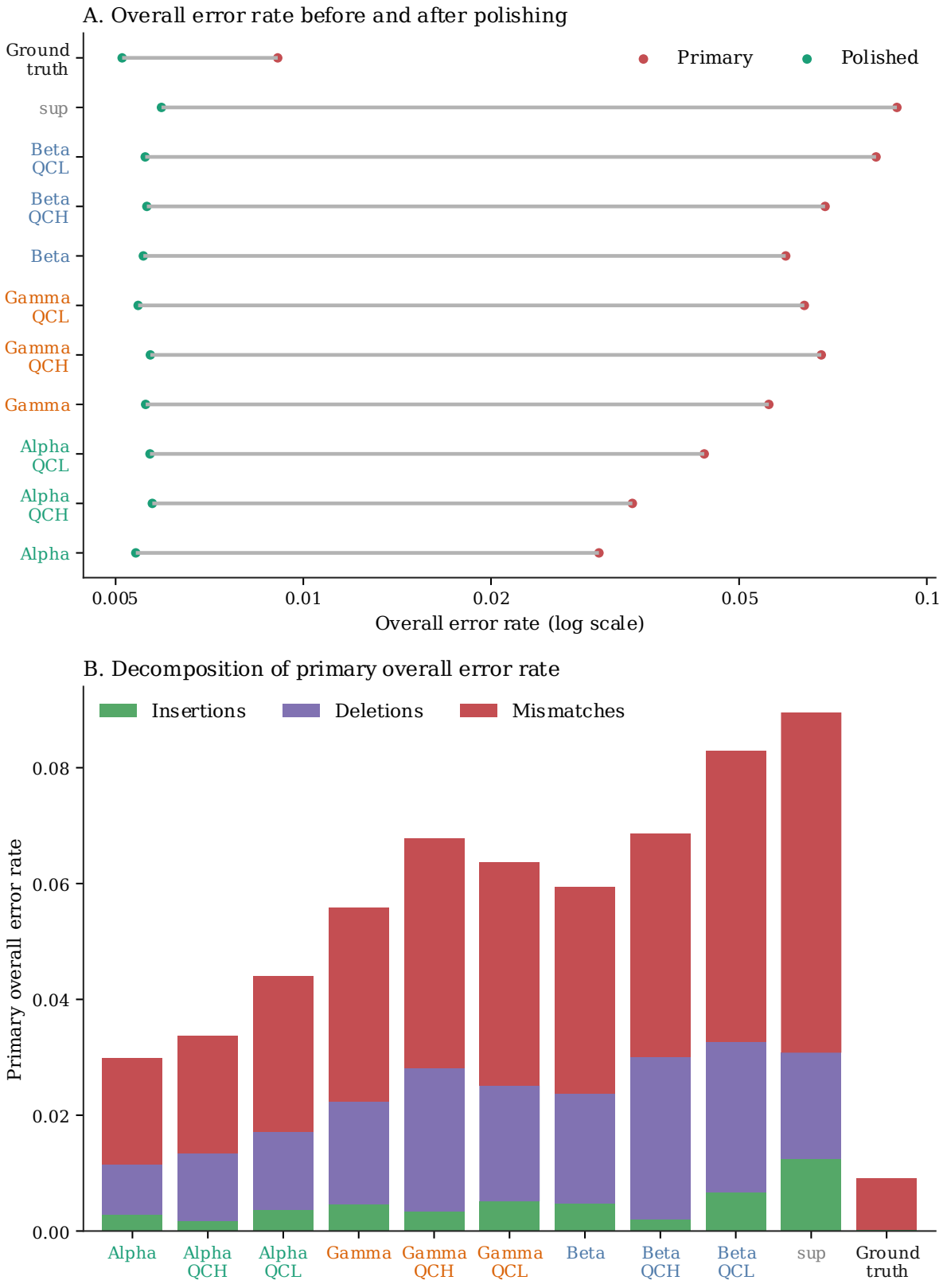

Figure 16. Dataset-level error analysis. (A). Lollipop chart comparing the overall error rates for primary and polished outputs on the held-out test dataset. *Each horizontal segment connects the primary (red dot) and polished (green dot) overall error rates for a single model output or the MGI ground truth on a log-scaled x-axis. The primary overall error rate is calculated as the proportion of aligned bases that are substitutions, insertions, or deletions relative to the WT Fc reference using primary alignments. The polished overall error rate is derived from the realignment of corrected* `FASTQ` *reads that passed the DMS filter; because DMS-aware correction in* `DualSiteDMSFilter` *restores all non-DMS positions to WT, the*

polished overall error rate reflects mismatches only within DMS regions. (B). Stacked bars decomposing the primary overall error rate into insertion, deletion, and mismatch components. The plot includes all nanopore outputs and the MGI ground truth, with each bar normalised to the number of mapped bases in the primary alignment. Counts were derived from primary alignment for the held-out test dataset, and the three stacked components sum to the plotted primary overall error rate. The high indel error rate in nanopore data motivates using overall error-rate metrics rather than the mismatch rate. The ground-truth dataset was provided as a trimmed R1 + R2 sequence and pre-polished using the WT Fc reference nucleotide for infill; therefore, no insertion or deletion errors were observed. Full test summary is available in `test.csv` *on GitHub (Appendix E).*

The ranking of model families by polished overall error rate did not match the clear hierarchy observed earlier (`Alpha` > `Gamma` > `Beta`). After polishing, some `Gamma`-family models (`GammaQCL` and `Gamma`) achieved lower overall error rates than both `AlphaQCL` and `AlphaQCH`. `Beta`-family models performed similarly to `Gamma`-family models in this context. Despite these shifts, `Alpha` remained the top performer, achieving a polished overall error rate of 0.538%, only 0.027% above the ground truth (0.512%), emphasising its advantage. In contrast, the pretrained `sup` model performed worst, with a polished overall error rate of 0.592%, marking a clear gap away from the fine-tuned models. `sup` has a slightly higher polished error rate (mismatch), yet it has significantly fewer unique variants and the lowest DMS pass rate. This indicates more false-positive mutations in the DMS regions, leading to less reliable variant calling. This clearly demonstrates that the extra unique variant and DMS-pass reads identified in the fine-tuned model, which are confined to the dual-site IgA Fc library constraints, are consistent with higher sensitivity to true mutations in the Fc backbone, rather than false-positive mutations.

A detailed analysis of Figure 16, Panel B, reveals several notable trends. First, all fine-tuned models exhibit a lower mismatch rate than the `sup` model. Second, only `Alpha`-family models reduced the deletion rate relative to the `sup` model; `Gamma`- and `Beta`-family models showed no such improvement and sometimes had even more deletions, distinguishing the families' behaviour. In contrast, all fine-tuned models share the common strength of reduced insertion rate. These distinctions suggest that deletions and insertions are not learned in the same way: they depend on loosely coupled or independent latent features. Third, across all model families, QCH models showed the lowest insertion error rates compared to their base and QCL counterparts, emphasising the benefit of the 99% alignment-quality threshold in QCH training. This threshold penalises reads with insertions more than reads with deletions, thereby biasing the training distribution toward insertion-free sequences and shifting the model's learned prior so that it defaults to fewer base insertions when the signal is ambiguous. Conversely, base and QCL models generally had lower deletion rates and higher insertion rates than QCH models. This is likely because their training sets did not exclude reads with indels, giving the models more opportunity to learn relevant signal patterns. These observations are consistent with the

neural network basecalling mechanism: models infer indels by learning a latent alignment between the continuous signal and the sequence, rather than detecting explicit signal breaks. Insertions and deletions arise when the modelled alignment adds or omits base transitions relative to the true sequence, a mechanism documented as the move table in the `Dorado` documentation. In summary, Figure 16 demonstrates that performance improvements in fine-tuned models are both interpretable and attributable to specific factors, rather than resulting from simple memorisation of the IgA Fc WT reference sequence. Although even the best fine-tuned model did not reach the ground-truth mismatch rate, Alpha exhibited only approximately twofold higher mismatch than the ground truth, whereas sup exceeded sixfold. Through combining domain-specific fine-tuning with a library-aware polishing pipeline, such as `DualSiteDMSFilter`, reduced the mismatch rate by restoring framework-region non-reference bases to the WT reference sequence and retaining DMS-region changes only when they satisfied the dual-site library constraints.

The mismatch-burden summaries in Figure 17 distinguish model differences at the read level. Across all nanopore models, the per-read mismatch rate was consistently higher in the DMS region than in the conserved framework. The main difference among models was the degree of separation between nucleotide/codon mismatch rate in DMS and framework regions: the `Alpha` model, for instance, had a DMS-to-framework nucleotide mismatch ratio of 29.97 and a codon ratio of 15.62, much higher than `Gamma`, `Beta`, or especially the `sup` model, whose ratios were 2.62 and 1.39. Among models, `Alpha`-family models achieved the greatest noise reduction, followed by `Gamma` and `Beta`. Crucially, improvements in DMS-region and framework noise were not proportional: high noise ratios in fine-tuned models stemmed mainly from substantial framework nucleotide/codon mismatch rate reduction, while mismatch rates in DMS regions saw only modest reduction. Notably, the `Alpha` model reduced the DMS mismatch burden to nearly the same level as the ground truth (3.701 vs 3.518), suggesting that fine-tuned models learned distinctive noise patterns from the training data, thereby reducing noise relative to the IgA Fc backbone signal without impairing true mutation detection.

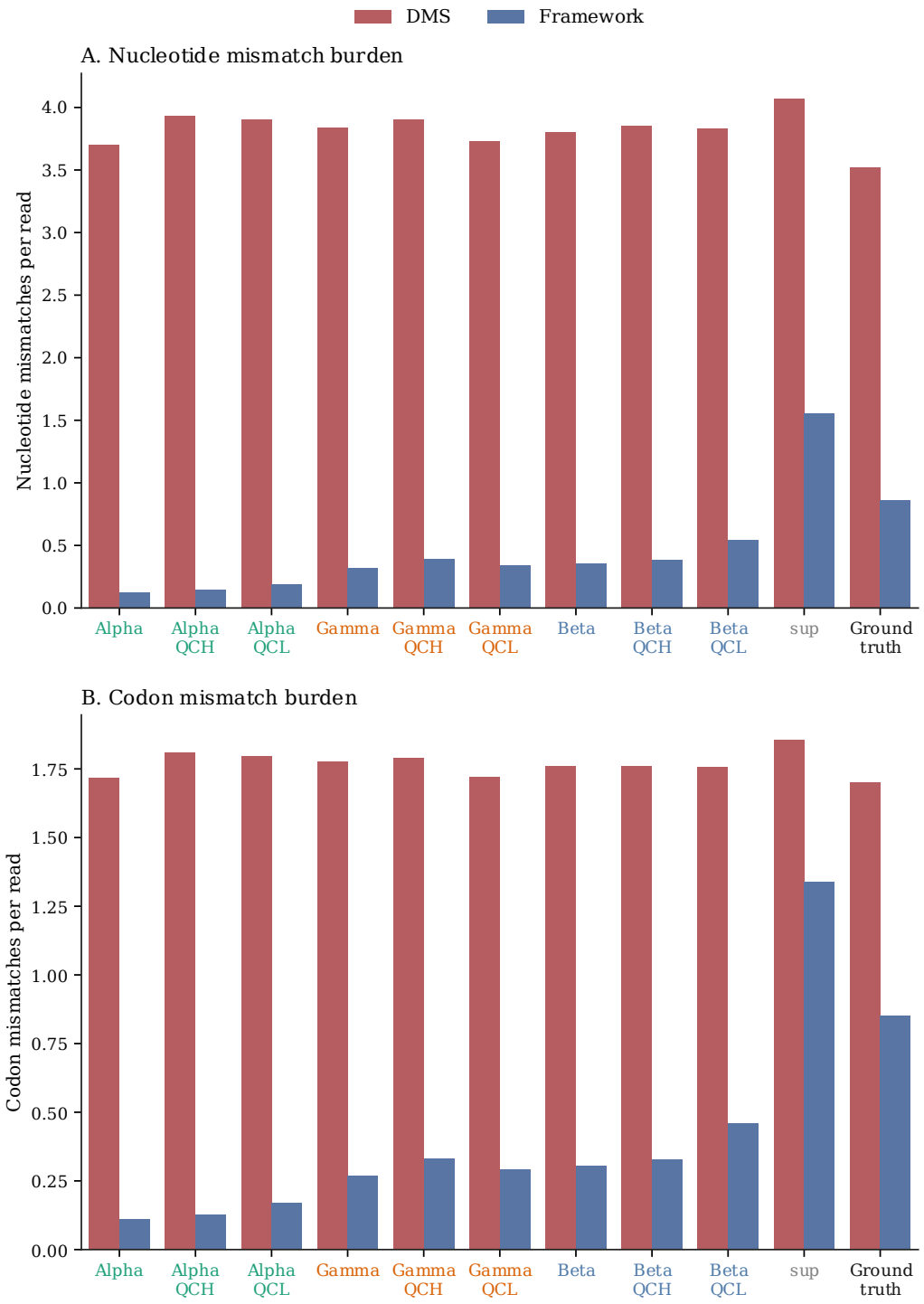

Figure 17. Read-level error analysis. *(A) Per-read nucleotide mismatch burden stratified by DMS and framework regions. For each model basecall and the ground truth, the DMS bar reports mismatches inside the analysed 4 DMS regions, and the framework bar reports mismatches outside those windows for DMS-pass reads. (B) Per-read codon mismatch burden stratified by DMS and framework regions. For each model output and the MGI ground truth, the DMS bar reports codon mismatches inside the analysed DMS regions, and the framework bar reports codon mismatches outside those windows for DMS-pass reads. A codon was counted as mismatched when at least one base in that codon differed from the IgA Fc WT reference sequence. Elevated DMS-region mismatches relative to the framework reflect genuine library diversity rather than sequencing artefact. A higher DMS-to-framework mismatch ratio indicates more faithful localisation of sequence variations to the designed mutational windows. Full test summary is available in* `test.csv` *on GitHub (Appendix E).*

Figure 18 shows the composition of outcomes for primary aligned reads during the DMS filtering step. Across all nanopore models, indels, defined as exceeding three insertions or deletions per read, whether contiguous or not, were the predominant mode of failure. Indel failures accounted for approximately 44.8% of all primary aligned reads in `Alpha` and exceeded 60% for most `Gamma` and `Beta` models, reaching 77.9% in the `sup` baseline. This pattern demonstrates that indel control, rather than general substitution suppression, remains the primary challenge for nanopore basecalling in DMS library reconstruction. The `Alpha` family was the most effective at reducing indel-related failures relative to the other families and the `sup` baseline. The fraction of reads failing the dual-site constraint remained relatively stable across all models.

As indel failures decreased in fine-tuned models, coverage and off-library failure fractions increased, most significantly in the `Alpha` family. This increase is not purely a compositional artefact of suppressing the dominant failure mode. These results can be interpreted by categorising failure modes into two groups: spatial failures, which reflect position-level accuracy relative to the IgA Fc WT reference (dual-site violations, indels within DMS windows, and off-library codons), and temporal failures, which reflect length-level accuracy (coverage shortfalls). Fine-tuned models demonstrated increased spatial sensitivity, as evidenced by a higher fraction of reads that passed and a substantially lower fraction of reads that failed due to indels. The coverage increase, however, reflects a genuine mechanistic trade-off: fine-tuned models learn a more conservative signal-to-nucleotide mapping that collapses more bases during CTC decoding, thereby producing fewer spurious insertions and, consequently, shorter sequences. This is directly observable in the mean read lengths: the pretrained `sup` model averaged 729 bp (+42 bp over the 687 bp reference), while `Alpha` averaged 679 bp (−8 bp) and `AlphaQCH` 673 bp (−14 bp). These shorter reads were more likely to fail coverage thresholds, establishing a genuine trade-off between indel accuracy and coverage completeness.

The higher fraction of off-library codon failures in fine-tuned models reflects a combination of two effects. First, a downstream effect: reads that would have been excluded by the indel filter in `sup` now survive to reach the off-library evaluation stage. Second, a potential increase in coherent false-positive codon-level mutations at DMS positions, despite the overall DMS region nucleotide mismatch rate being lower in fine-tuned models than in `sup` (e.g., 0.0263 for `Alpha` versus 0.0289 for `sup`), prevents a clean separation of these two contributions from filter-level data alone. Notably, fine-tuned models achieved framework nucleotide mismatch rates an

order of magnitude lower than `sup` (e.g., 0.00024 for `Alpha` versus 0.0029 for `sup`; see Figure 19), confirming that the increased off-library failures do not reflect a general rise in spurious mutations across the reference.

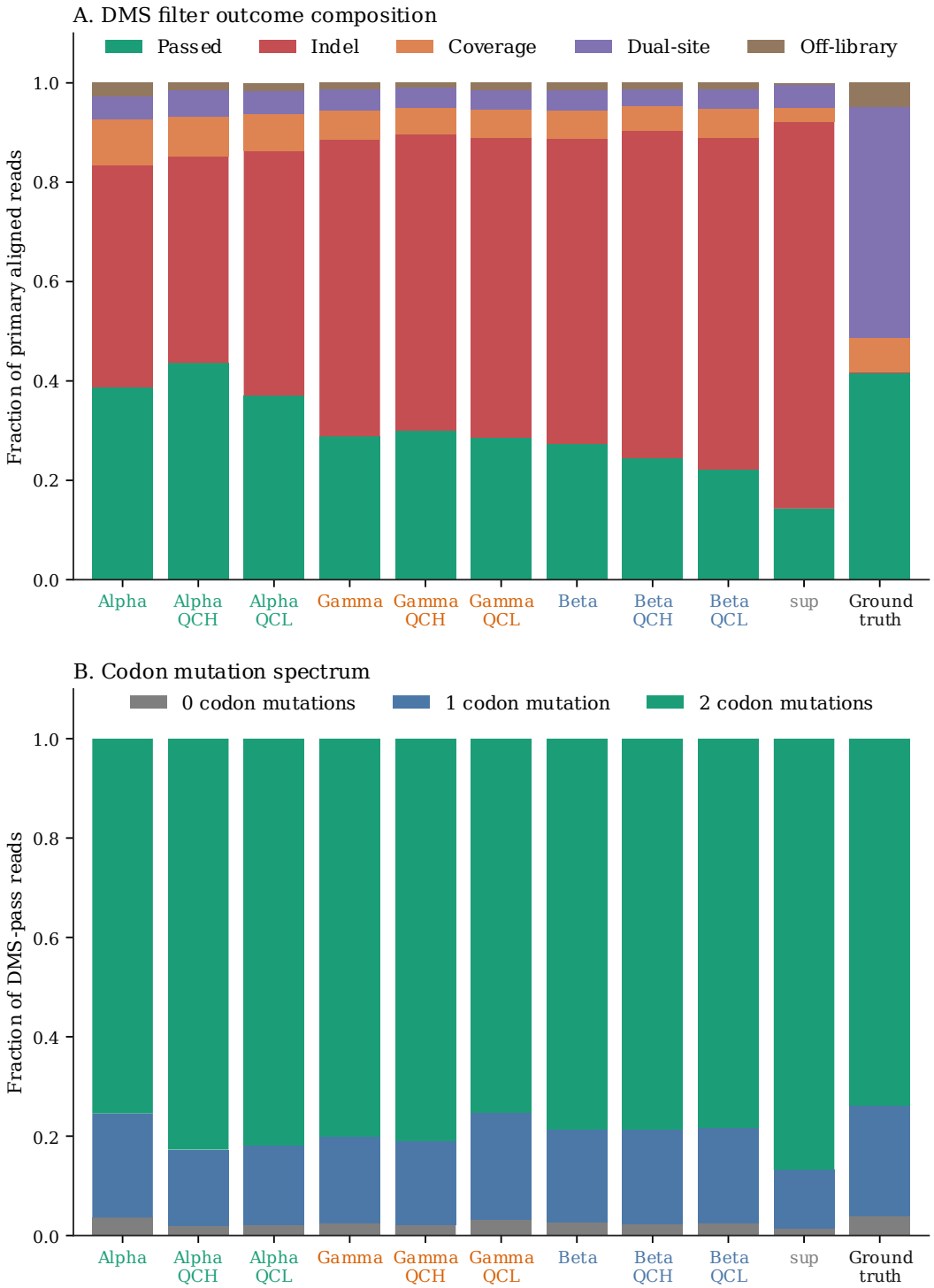

Figure 18. Composition analysis at dual-site DMS library level. *(A). Stacked bars showing why primary-aligned reads failed the DMS filter for each model output and the ground truth. The five coloured segments correspond to passed reads, excessive indels, coverage failure, dual-site constraint failure, and off-library codons, and the total number of primary-aligned reads for that output normalises each bar. Failure counts were generated from the DMS filter summary on the held-out test dataset. (B). Stacked bars showing the distribution of 0-, 1-, and 2-codon mutations among DMS-pass reads. For*

each model output and the ground truth, the bar is normalised by the number of reads passing the DMS filter and therefore sums to 1. The counts were computed after all DMS filter criteria had been applied to the held-out test dataset; the 0-codon class corresponds to retained WT reads. The distribution reflects the dual-site library design, which was constructed to introduce mutations at up to two codon positions simultaneously. Full test summary is available in `test.csv` *on GitHub (Appendix E).*

In summary, fine-tuned models predicted more true-positive mutations at designed positions without introducing spurious mutations at random framework positions, while accepting a proportionate trade-off between indel suppression and coverage completeness. The combined fraction of reads failing due to indels and coverage remained consistently lower in fine-tuned models than in the `sup` baseline. These findings reinforce the importance of evaluating model performance at a fine-grained level rather than relying on any single error metric.

Figure 19 shows that framework mismatch error rate at the nucleotide and codon levels directly correlated with downstream concordance metrics. As framework mismatch rates increase, both exact overlap mass and Jensen-Shannon similarity to the ground truth decrease. Even minor sequencing noise can therefore disrupt variant calling. Framework positions serve as internal controls, since mutations there are more likely to be sequencing artefacts (e.g., PCR, basecalling) than true variants. The `Alpha` family shows low framework mismatch and strong concordance with ground-truth abundance. In contrast, the `Beta` family exhibits more error in less-informative positions, inflating apparent diversity without improving true variant recovery. Thus, framework mismatch is useful both as a retrospective explanation and a practical screening metric when no short-read benchmark exists. At the nucleotide level, the `Alpha` family performed better than the `Gamma` family, which outperformed the `Beta` family. The `sup` model performed worse than all fine-tuned models. This shows the read-level noise pattern seen in Figure 17 was not coincidental and matched nucleotide-level measurements. Both Panels A and B were analysed using ordinary least squares trend lines, yielding R^2 values of 0.906 and 0.868. Excluding the `sup` model from the regression did not substantially alter the fit, with R^2 values of 0.857 for Panel A and 0.873 for Panel B. The y-intercepts of these fits, 0.48 for exact overlap mass and 0.61 for Jensen–Shannon similarity, indicate that even at zero framework mismatch, concordance with the ground truth would remain well below unity, suggesting that other error sources beyond framework noise impose an independent ceiling on variant-calling fidelity (e.g., mismatch between training and test set).

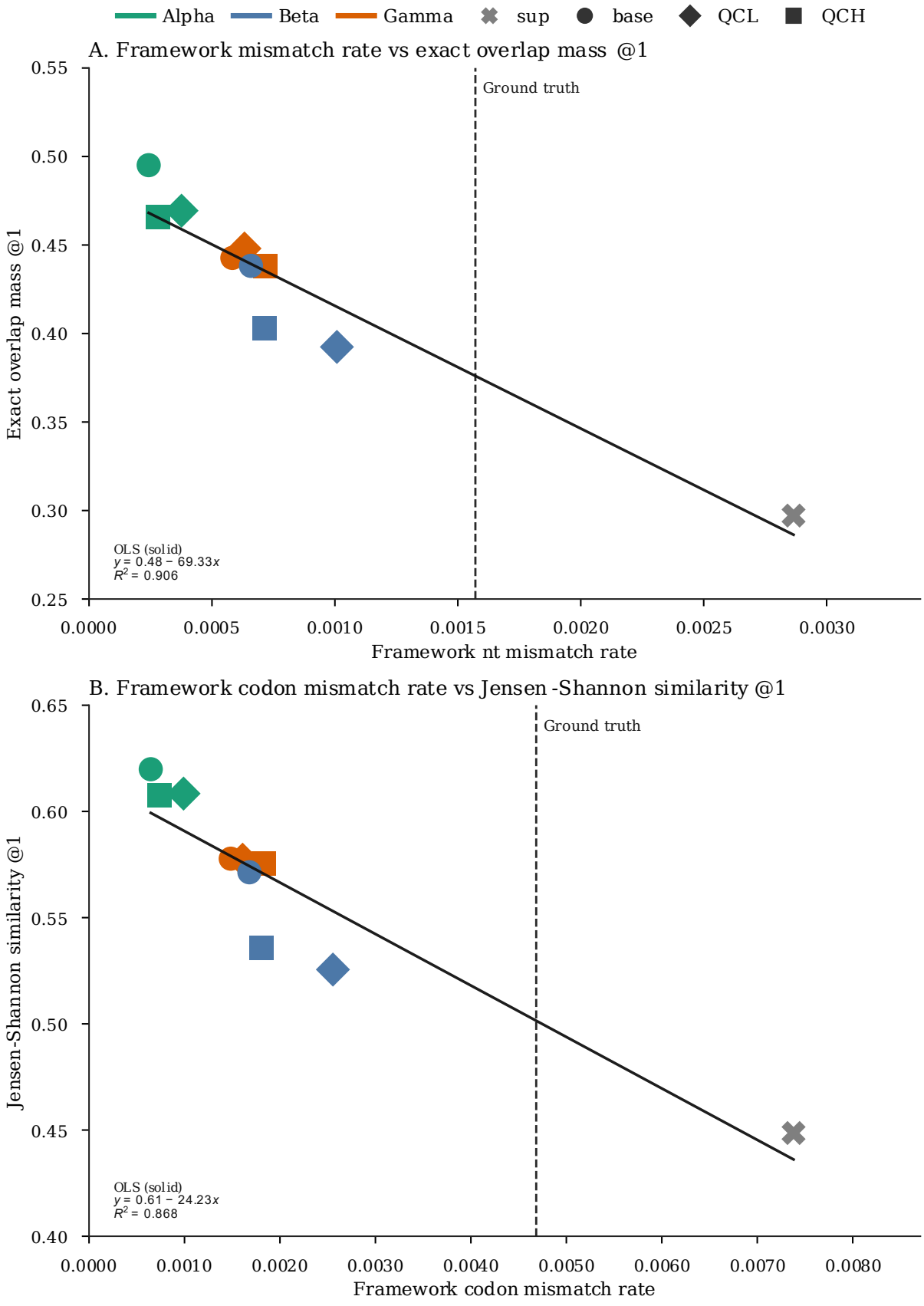

Figure 19. Nucleotide-level & Codon-level correlation analysis. *(A). Scatter plot comparing framework nucleotides mismatch rate per base with exact overlap mass @1. The x-axis uses nucleotide mismatch rate measured outside the DMS regions using raw aligned bases, and the y-axis uses exact overlap mass computed from non-WT variant tables after*

applying a minimum count threshold of one. Exact overlap mass measures the fraction of the total normalised abundance shared between the nanopore-derived and ground-truth variant tables; values closer to 1.0 indicate better-conserved abundance profiles. (B). Scatter plot comparing framework codon mismatch rate with Jensen-Shannon similarity @1. The x-axis uses the codon mismatch rate measured outside the DMS regions using raw aligned bases, and the y-axis uses the Jensen-Shannon similarity computed from non-WT variant distributions after applying a minimum count threshold of one. Jensen-Shannon similarity is a distribution-level concordance measure that compares normalised variant frequency spectrum; values closer to 1.0 indicate greater distributional agreement between the datasets. Each point represents a model output, with colour and marker shape encoding the model family and training subset. For concordance measures, each basecalled dataset is compared against the ground truth. Full test and concordance summaries are available in `test.csv` *and* `correlation.csv` *on GitHub (Appendix E), respectively.*

Figure 20 presents per-position mismatch rates for each model output mapped to the IgA Fc WT reference. All fine-tuned models suppressed framework mismatches more effectively than the `sup` baseline, with the `Alpha` family achieving framework noise distributions closest to the ground truth. Multiple models showed recurrent mismatch hotspots within framework regions, particularly near positions 660–669. These hotspots were most pronounced in the `Alpha`-family models, with a similar pattern in the `Gamma` family due to overlapping training datasets. The elevated noise near the 3' end likely reflects a sequence-context-dependent effect that persists across all models. In addition, the fact that fine-tuning suppressed framework noise broadly but did not fully resolve this localised hotspot suggests it arises from an inherent signal-level limitation in these k-mer contexts rather than a training data deficiency, which would explain why it appears across multiple families and why its magnitude scales with overall model quality rather than data source.

Within DMS regions, the polished mismatch rates at certain positions appear elevated relative to primary rates. This does not indicate error inflation from polishing; rather, DMS-aware filtering removes reads with off-target noise or indel-driven smearing before polishing, concentrating the remaining signal at positions with genuine non-reference variants. After polishing, the pretrained sup baseline, all fine-tuned models, and the short-read ground truth showed similar mismatch patterns within DMS zones. This similarity was partly expected because polishing and DMS filtering removed many false-positive reads, especially from the noisier sup basecalls, leaving retained mismatch positions concentrated around genuine DMS variants rather than random sequencing noise. Before polishing, fine-tuned models had already shown improved mismatch distributions, and the Alpha-family models, in particular, exhibited near-identical primary and polished mismatch profiles with DMS regions, indicating that their basecalls required minimal correction.

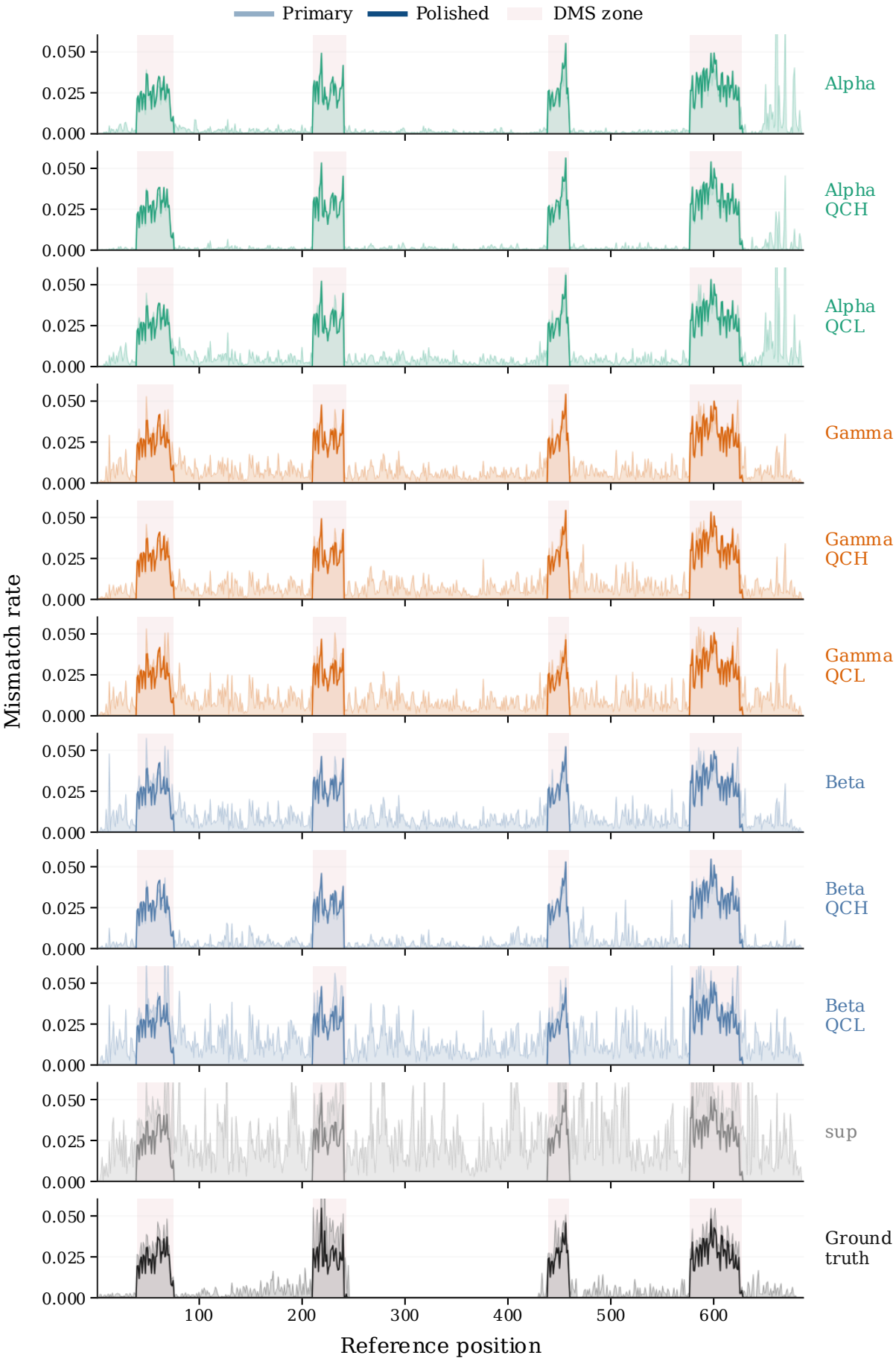
Primary
Polished
DMS zone
Mismatch rate
0.050
0.025
0.000
Alpha
Alpha QCH
Alpha QCL
Gamma
Gamma QCH
Gamma QCL
Beta
Beta QCH
Beta QCL
sup
Ground truth
100
200
300
400
500
600
Reference position

Figure 20. Per-position level noise profiling. *Per-position mismatch-rate profiles across the Fc reference for all outputs. Each row corresponds to one nanopore output; the lighter trace shows the primary mismatch rate, and the darker trace shows the polished mismatch rate, while pink shading marks the four DMS regions at positions 40-75, 211-243, 439-459, and 577-627. Mismatch rates were computed per reference base position from* `CIGAR`*-aligned reads as the proportion of reads carrying a non-reference nucleotide at each position (does not include indels). Following correction, the mismatch rate outside DMS regions collapses towards zero by design, isolating variant signal within the defined DMS regions. All primary alignments from each dataset were used to compute the primary mismatch rates. All DMS-pass reads were realigned to compute the polished mismatch rates.*

Figure 21 assessed whether discordance between nanopore and ground truth mutation frequencies arose from a few position-specific hotspots or from broader miscalibration across the DMS space. Across all nanopore model outputs, most shared mutation events showed minimal frequency differences, indicating reasonable positional fidelity in the overall mutational landscape. However, distributions were positively skewed, with nanopore models more frequently overestimating than underestimating the abundance of shared mutations relative to the MGI ground truth. Fine-tuned models, particularly `Alpha`-family QC variants, exhibited narrower deviations and fewer large discrepancies than the baseline `sup` model. No single DMS region dominated the error structure across models, suggesting that the remaining disagreement stems from small differences in variant abundance distributed across many mutation positions rather than from repeated failures at specific DMS sites.

Because this main analysis did not normalise sequencing depth between nanopore and MGI datasets, it should be interpreted as a qualitative positional diagnostic. However, the sequencing-depth-normalised control in Figure F1 showed the same overall trends. The pattern of improvement observed in the per-position frequency analysis suggests that fine-tuning primarily improved the decoding of the nanopore signal for the IgA Fc amplicon rather than teaching the model an expected variant distribution of the library. The interpretation is supported by the strong performance of `Alpha`-family models, which were trained only on a clean Restriction-digest-based WT Fc signal and were therefore not exposed to the biological mutation distribution present in the DMS library or PCR error. Their improvement came from learning Fc-backbone-specific sequence context and systematic error patterns, rather than from exposure to the biological mutation distribution itself. This learning reduced positional smearing, the tendency of basecalling errors to distribute non-reference signals across neighbouring positions, and framework miscalls during downstream variant calling. However, the persistent positive skew across all models indicates that some residual bias toward over-assigning non-reference signals remained even after fine-tuning.

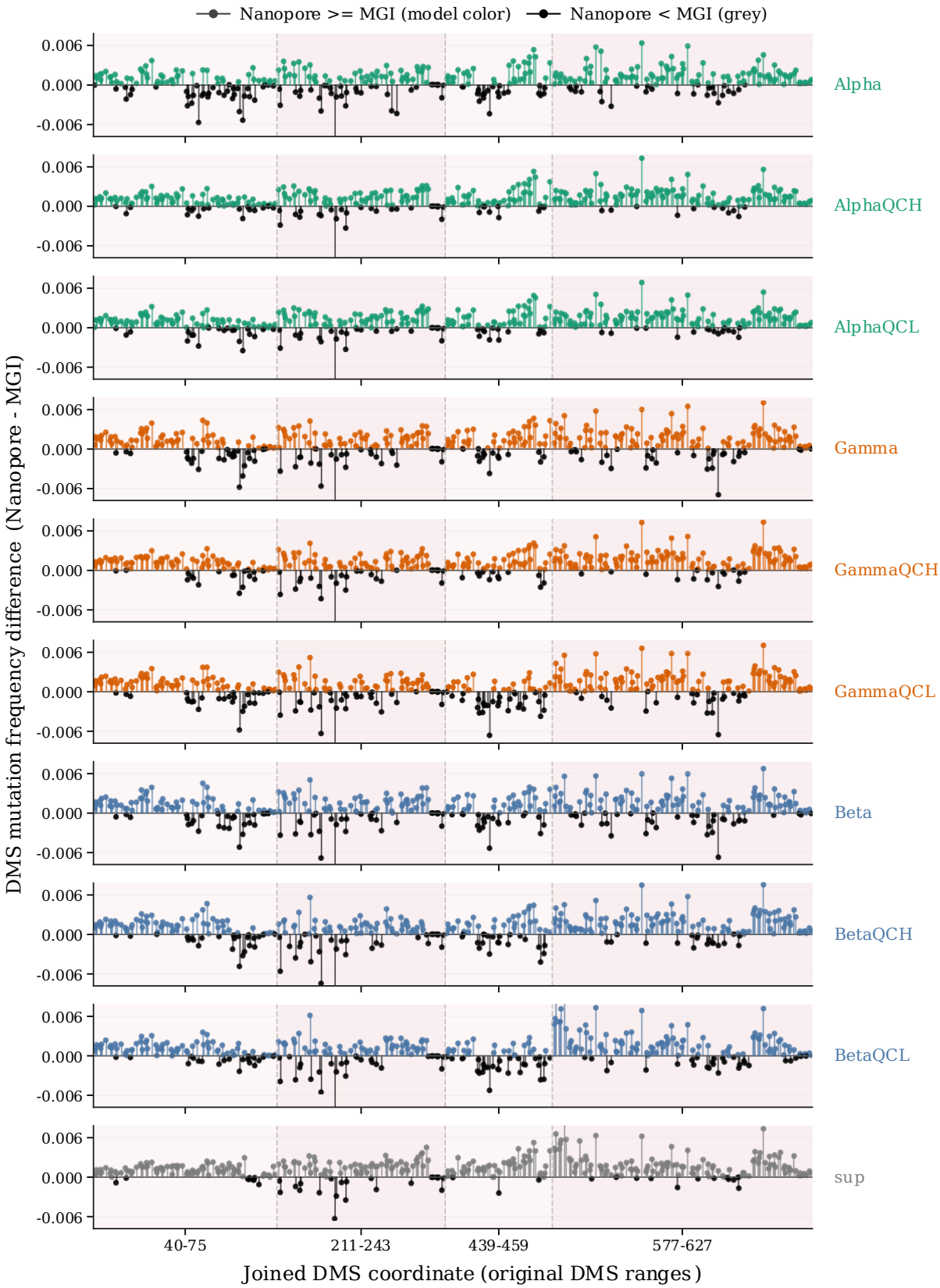

Figure 21. Mutational-event level concordance. *Per-position DMS mutation-frequency differences (nanopore minus MGI) for mutation events shared between the two variant tables at a minimum count threshold of 1, with WT variants excluded. Each row corresponds to one nanopore output; coloured points indicate nanopore overestimation and black points indicate underestimation. Frequencies were calculated without read-depth matching; the depth-matched version is shown in Figure F1. Full mutational event summary is available in* `mutation.csv` *on GitHub (Appendix E).*

3.4 Concordance to MGI-Derived Variant Profiles

Section 3.4 evaluates variant-profile concordance between nanopore basecalling outputs and the MGI ground truth library across multiple complementary metrics. Whereas the preceding sections examined error modes at the microscale, such as individual basecalling errors and read-level DMS constraint violations, this section shifts to a macroscale assessment of whether nanopore-derived variant profiles faithfully represent the composition of the ground-truth mutational library. Rather than asking whether individual reads are biologically plausible, concordance analysis asks whether the aggregate variant population recovered by each basecalling workflow reflects the true library in both identity and relative abundance. To this end, four dimensions of concordance were measured: exact variant overlap, abundance-weighted overlap, frequency correlation, and threshold robustness. Together, these metrics determine whether a model can recover both common and low-abundance variants without introducing excessive false positives or filtering out true positives, a requirement for effective variant calling in deep mutational scanning.

Figure 22 ranks each nanopore output by weighted Jaccard similarity to the MGI reference at a minimum count threshold of 1, where the metric emphasises agreement in variant abundance rather than presence alone. The `Alpha`-family models occupied the top three positions, with `Alpha` achieving the highest concordance (0.240), followed by `AlphaQCH` (0.222) and `AlphaQCL` (0.220). `Gamma` and `Beta` models formed a weaker middle tier, while the pretrained `sup` model ranked lowest (0.070), confirming a substantial gap in concordance between the baseline and all fine-tuned outputs. Although `AlphaQCH` and `AlphaQCL` retained marginally more intersecting variants than the `Alpha` model, the `Alpha` model more faithfully reproduced the ground-truth abundance distribution across shared variants. This distinction indicates that variant retention and abundance fidelity are partially independent axes of performance; that is, the most effective model weighs both rather than maximising a single count-based overlap measure.

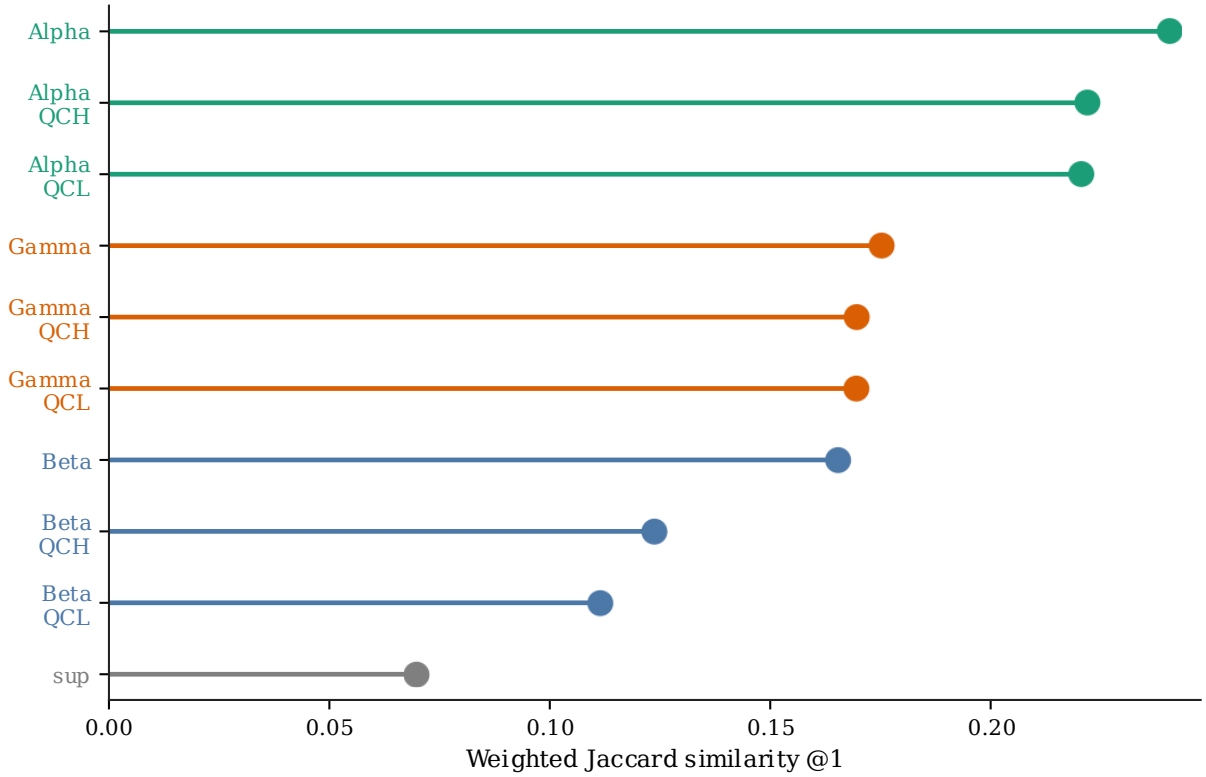

Figure 22. Weighted Jaccard concordance ranking at minimum count threshold 1. *Horizontal ranking plot of weighted Jaccard similarity at minimum count threshold 1. Each segment ranges from 0 to the score for a single output, and outputs are ordered from highest to lowest concordance with the ground truth. Scores were computed from retained non-WT variant tables after DMS filtering. A full concordance summary is available in* `correlation.csv` *on GitHub (Appendix E).*

Figure 23 separates variant intersection from abundance calibration by plotting nanopore abundance against ground-truth abundance for shared mutation events on log–log axes. Two complementary regression fits are shown in each panel. For `Alpha`, the OLS slope was 0.52 (R^2 = 0.428), while the binned-median slope rose to 0.88 (R^2 = 0.979), indicating that the underlying abundance relationship was substantially tighter than the raw scatter suggested. `Beta` and `Gamma` showed parallel but weaker patterns (binned-median slopes of 0.78; $R^2 \approx 0.97$), and the pretrained `sup` model was much weaker (binned-median slope 0.44; $R^2 = 0.914$). Slopes below 1 across all models indicate that high-abundance variants in the ground truth were systematically underestimated by the nanopore models relative to their lower-abundance neighbours, consistent with residual basecalling errors dispersing true variant signal across related false-positive calls. Overall, Figure 23 reinforces the conclusion from Figure 22 that the fine-tuning advantage goes beyond variant detection to calibration of the recovered mutational profile.

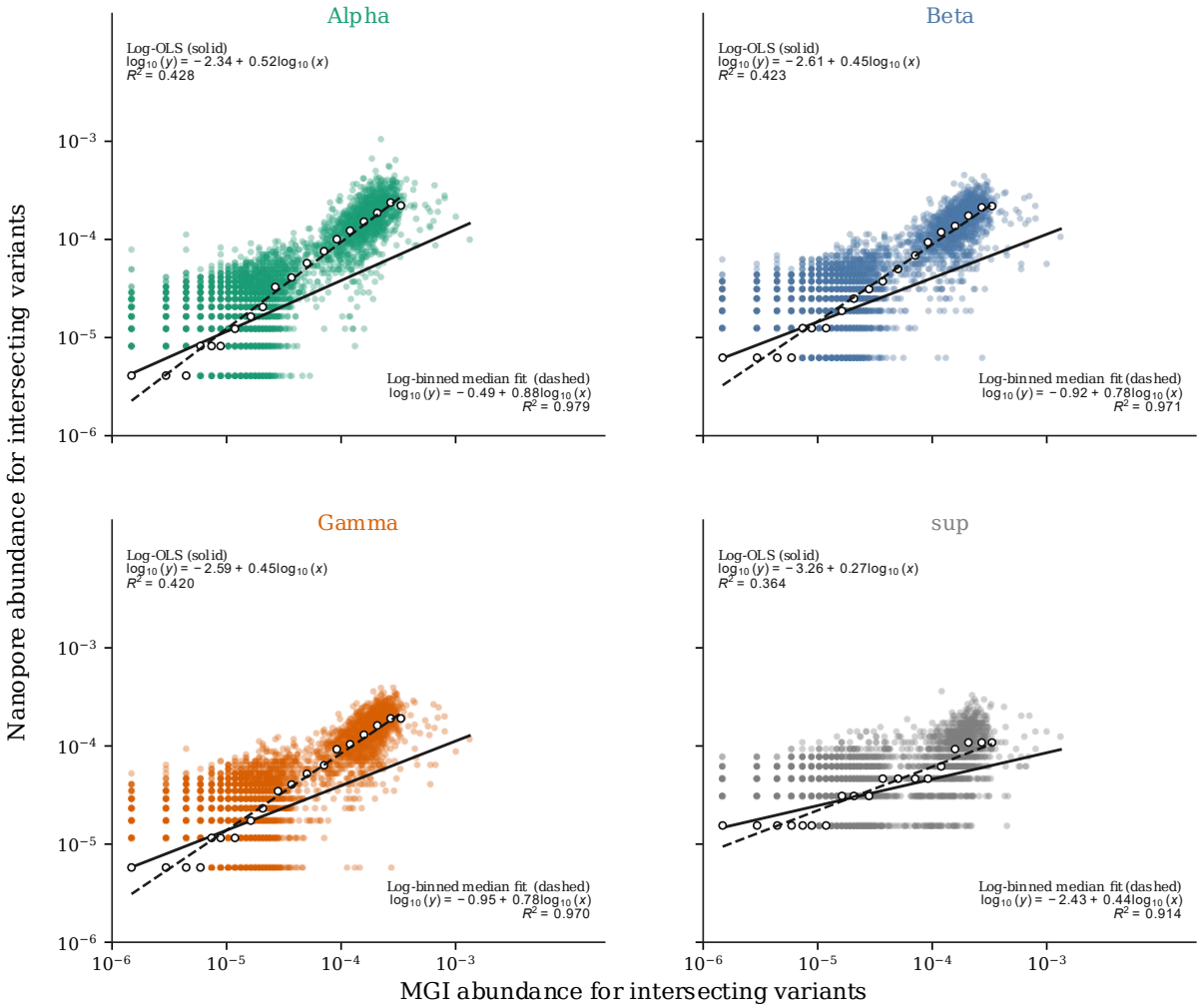

Figure 23. Abundance concordance for intersecting variants. *Log–log scatter plots of nanopore versus ground-truth abundance for shared non-WT variants (minimum count threshold of 1) across the four base-model outputs:* `Alpha` *(n = 74,476),* `Beta` *(n = 58,620),* `Gamma` *(n = 62,274), and* `sup` *(n = 33,534). Each panel displays two fits: a log-space OLS regression on all individual data points (solid line), which captures the overall abundance relationship including low-abundance scatter, and a regression through log-binned medians (dashed line), which isolates the central abundance trend.* For all models, see Figure F2. A full concordance summary is available in `correlation.csv` on GitHub (Appendix E).

Figure 24 plots the number of unique nanopore variants against the number of intersecting variants at threshold 1. Across all ten outputs, the two quantities scaled in near-perfect linearity (R^2 = 0.999), with a slope of approximately 1.69. This means that for every additional intersecting variant that a model recovered, it also introduced roughly 0.69 false-positive variants absent from the ground truth. `Alpha` family models occupied the upper-right cluster, showing both the largest intersection sets and the largest unique-variant sets. At the same time, the `sup` baseline sat in the lower-left of the same line. The tight fit indicates that no model tested escaped this proportional trade-off; models that detected more true variants also generated proportionally more false positives. Consequently, raw variant counts alone cannot distinguish augmented sensitivity from error-driven inflation, motivating the threshold robustness analysis in Figure 25.

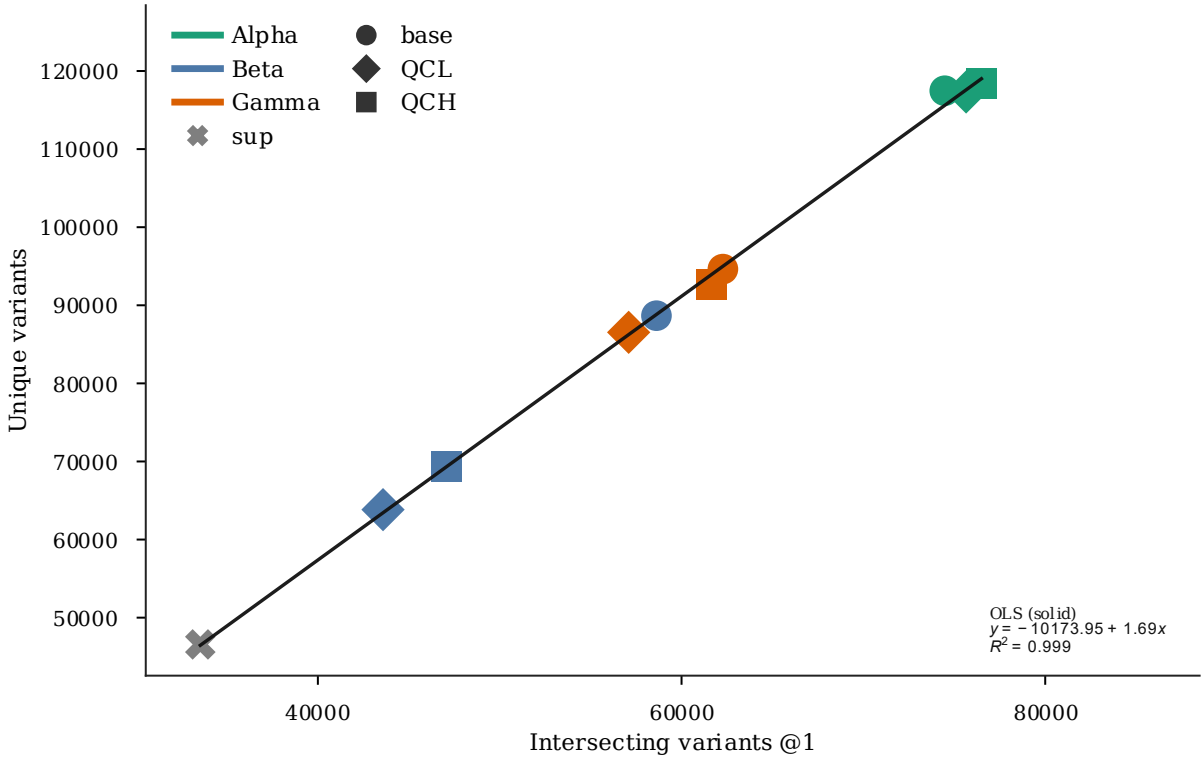

Figure 24. Intersecting variants versus retained variant count. *Scatter plot relating the number of intersecting variants at threshold 1 to the total number of unique variants retained after DMS filtering for each nanopore model. Each point is one nanopore output; colour denotes model family and marker shape denotes training subset type. Variant counts were computed from the non-WT variant tables used for concordance analysis. A full concordance summary is available in* `correlation.csv` *on GitHub (Appendix E).*

Figure 25 evaluates whether the concordance advantages of the fine-tuned models observed at threshold 1 persisted as higher minimum read count thresholds were applied. Here, a threshold of *t* means that nanopore variants observed fewer than *t* times were excluded before comparison with the short-read ground truth. Across all seven metrics, `Alpha` family models consistently occupied the top tier from thresholds 1 through 10, indicating that their advantage over the others did not depend on retaining low-count variant calls, which are disproportionately vulnerable to threshold filtering. Among the `Alpha` models, `Alpha` maintained the highest weighted Jaccard similarity, Jensen-Shannon similarity, exact overlap mass, and recall across all thresholds. At the same time, `AlphaQCH` achieved marginally higher precision at stringent cutoffs (e.g. 0.941 versus 0.930 at threshold 10). However, recall remained low for all nanopore models at high thresholds, showing the significant attrition of intersecting variants visible in the bottom panel: the number of intersecting variants declined by roughly an order of magnitude between thresholds 1 and 10 for all outputs, and by nearly two orders of magnitude for the `sup` baseline. The pretrained `sup` model illustrated the limiting case of this attrition: at threshold 10, it achieved nominal precision near 1.0, but recall fell to approximately 0.04,

indicating that high precision was achieved mostly by excluding nearly all variant calls rather than by accurate detection.

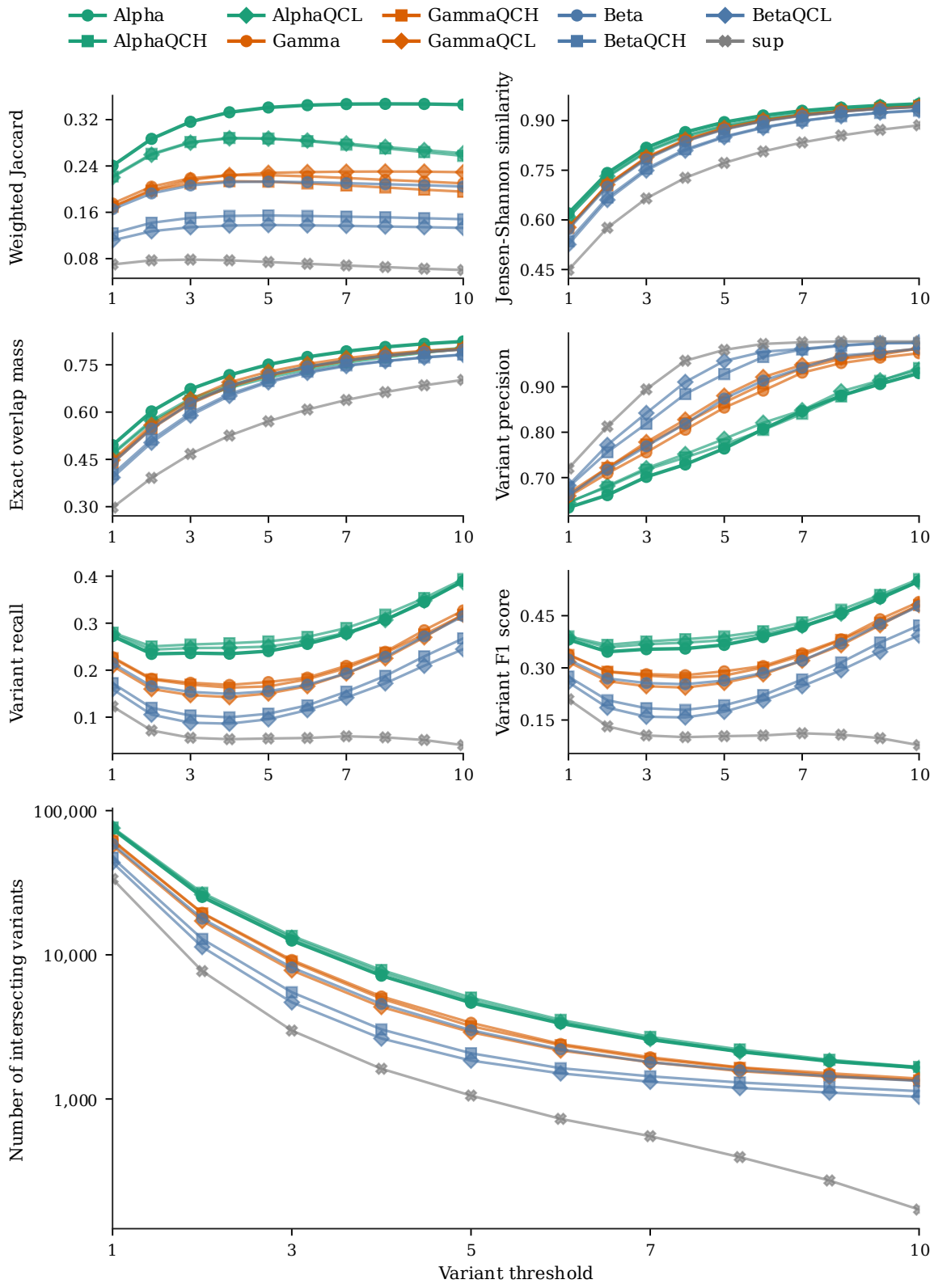

Figure 25. Threshold-robustness curves across nanopore models. *Multi-panel summary of how concordance changes as the minimum variant count threshold increases from 1 to 10. The top six panels show weighted Jaccard similarity, Jensen-*

Shannon similarity, exact overlap mass, variant precision, variant recall, and variant F1 score, respectively; the bottom stretched panel shows the number of intersecting variants on a log-scaled y-axis across the same threshold range. Each coloured line is a single nanopore output, with colour denoting family and marker shape denoting the training subset type. Metrics were computed from the concordance table derived from the held-out test data. All concordance metrics range from 0 to 1, with higher values indicating stronger concordance with the ground truth. For full metrics definitions, see Section 2.2. A full concordance summary is available in `correlation.csv` *on GitHub (Appendix E).*

Taken together, Figures 22–25 demonstrate that concordance between nanopore basecalls and ground-truth reads depends on more than just the presence of intersecting variants. The near-perfect linear relationship between unique and intersecting variants (Figure 24) indicates that increased detection inevitably leads to a proportional increase in false positives across all models, likely reflecting a shift in sequence distribution between WT training data and the variant-containing test library: both datasets share the same Fc framework, but the test library contains designed non-reference variants that were absent from the WT training data. No model escaped this trade-off; consequently, the distinguishing factor among outputs was not the rate of false-positive generation, but the fidelity with which shared variants preserved the ground-truth abundance structure. On this criterion, `Alpha` consistently outperformed other models across weighted Jaccard similarity (Figure 22), abundance correlation (Figure 23), and threshold-robustness analysis (Figure 25), with its concordance advantage persisting after removal of low-count calls rather than arising from a training artefact. These results suggest that the appropriate objective for nanopore amplicon variant calling in a DMS context is population-level reconstruction, which means faithfully recovering both the identity and relative abundance of library variants rather than minimising false-positive counts per se. By this standard, `Alpha` emerges as the most concordant model in the final synthesis in Section 3.6.

3.5 Model Weight Deep-Dive

The preceding sections showed that domain-specific fine-tuning changed downstream variant-calling behaviour, but those analyses did not explain where the changes occurred inside the model. This section examines weight change after fine-tuning by comparing the exported fine-tuned models against the pretrained `sup` baseline and asks whether different patterns of internal weight change are consistent with the observed biological performance. The analysis is exploratory rather than conclusive: only 9 fine-tuned models were trained, and models within each family are not statistically independent because of shared partial training data. The purpose here is therefore to identify potential mechanisms that explain prior benchmark results, not to prove deterministically that any single layer caused a particular downstream outcome.

Each fine-tuned model export was compared with the pretrained `sup` baseline by matching tensor names between the two weight sets. For each matched tensor, the weight difference was calculated and aggregated at different levels of granularity. Tensors were then grouped into six model components: convolutional kernels, transformer self-attention, transformer feed-forward layers, transformer norm/deepnorm layers, linear upsample layer, and the CRF head. Three complementary summaries were calculated: the first metric is the normalised L2 norm, defined as the normalised Euclidean distance of the fine-tuned model from pretrained `sup` model, the next metric is the proportion of model-level squared L2 norm, which is defined as the sum of squared weight changes in a model component divided by the sum of squared weight changes throughout the entire model (i.e., normalised proportion). The third metric is the model-level p99.9 standardised absolute weight change, defined as the 99.9^{th} percentile of absolute weight change after scaling by the standard deviation of the corresponding pretrained `sup` tensor. These summaries distinguish normalised weight change, proportional weight change, and rare absolute weight change.

Figure 26 shows that fine-tuning did not result in substantial changes in weights compared to the pretrained `sup` baseline. Instead, it produced small but structured weight change concentrated in the parts of the model most directly involved in signal interpretation and base emission (i.e., predicted logits are directly involved in beam search decoding). Across all fine-tuned models, the largest normalised weight change occurred in the convolution kernels, approximately 0.055-0.070, followed by the CRF head, approximately 0.028-0.042. In contrast, the transformer encoder blocks remained comparatively stable: weight changes in the self-

attention and FFN layers were only approximately 0.011-0.013, while changes in the norm/deepnorm layers were even smaller, at approximately 0.004. This pattern is consistent with the downstream benchmarking results. The pretrained `sup` model already contained a broadly useful nanopore sequence representation, while domain-specific fine-tuning mainly recalibrated how that representation was entered into the model and decoded for the IgA Fc sequence.

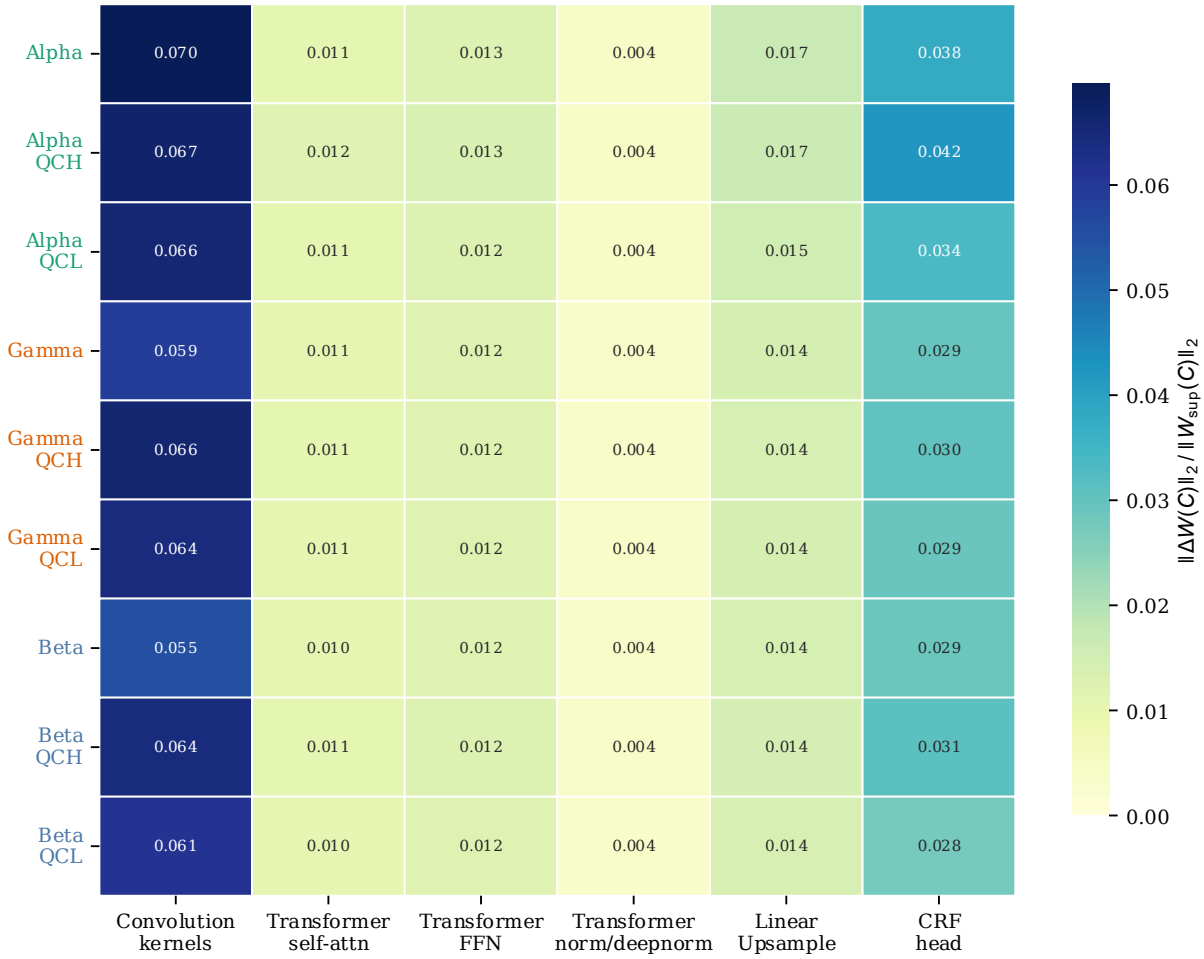

Figure 26. Normalised L2 norm in different model components across models. *The normalised L2 norm was computed as the square root of the sum of squared weight changes across relevant tensors when comparing the fine-tuned model with the pretrained `sup` model. A precise metric definition is available in Section 2.2.4.*

Figure 27 complements Figure 26 by showing the distribution of total squared weight change across model components. This metric is intentionally not normalised by the number of parameters because its purpose is to show where the aggregate weight change occurs across the full model. The transformer FFN accounted for approximately 50.7%-56.5% of the proportional weight change across all models, despite its modest normalised weight change. This is expected because the FFN contains many more parameters than the convolutional kernels or the CRF head. Convolution kernels contributed the second-largest proportional

weight change, ranging from approximately 21.9% to 28.7%, followed by transformer self-attention layers at approximately 15.7% to 17.9%. The CRF head contributed only approximately 2.6-4.9%, while the linear upsample and norm/deepnorm layers together contributed less than 1%.

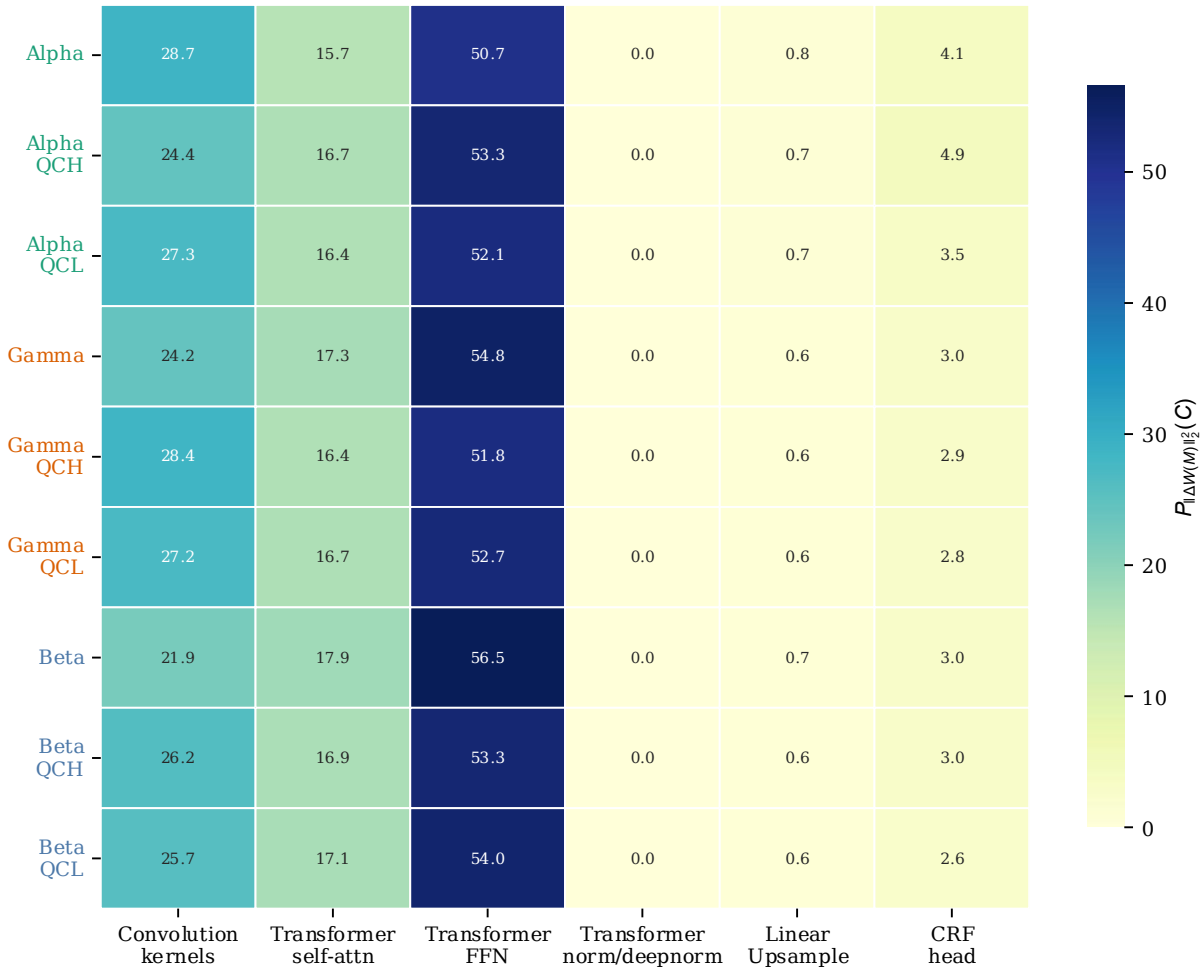

Figure 27. Proportion of model-level squared L2 norm shared by different model components across models. *The y-axis shows the proportion of model-level squared L2 norm each component is responsible for. The metric is additive, and all components add up to 100%. Unlike the normalised L2 norm, this metric is not normalised by the number of parameters or by the pretrained component norm. It therefore reflects where the total weight change is distributed in the model, rather than the relative change per parameter.*

The distinction between Figures 26 and 27 is therefore important. Figure 26 identifies which components changed most relative to the pretrained sup weights, whereas Figure 27 shows how the aggregate weight change was distributed across the model. The convolutional kernels and the CRF head showed the largest relative changes, but the FFN accounted for the largest share of total weight change because it is much larger. Therefore, fine-tuning was not confined to the convolution kernels and the CRF head, nor was downstream improvement explained solely by the transformer blocks undergoing the most weight changes: `Beta` had the highest proportional

weight change in FFN layers and one of the lowest proportional weight changes in convolution kernels, as a result it performed worse than `Alpha` and `Gamma` in read retention, framework noise, primary overall error rate, and variant-abundance concordance. The useful adaptation was therefore not simply the proportion of weight change, but whether the change occurred in functionally relevant parts of the network. `Alpha` provides the clearest example of a structured adaptation pattern. It combined the highest proportional weight change of 28.7% in convolution kernels, the relatively high proportional weight change of 4.1% in the CRF head, the lowest proportional weight change of 67.4% in the transformer blocks, and the best downstream balance of pipeline retention, sequence noise reduction, primary overall error rate, and variant-abundance concordance. This supports the interpretation that successful fine-tuning required broad adjustments to internal transformer blocks and improved calibration of the convolution kernels and CRF head, which are most directly involved in signal processing and base emission.

Figure 28 asks whether the observed weight changes included extreme updates. Across all components and models, the $P_{99.9}(C)$ remained below 1.0, meaning that even the most variable 0.1% of parameters within each component generally shifted by less than one standard deviation of the baseline tensor. This supports the conclusion that single-epoch fine-tuning is sufficient to calibrate to IgA Fc signal context without major weight drift in the underlying model.

The strongest $P_{99.9}(C)$ spike occurred in the convolutional kernels, with values ranging from approximately 0.46 to 0.60. `Alpha` had the largest convolutional tail value, 0.603, which is consistent with its high normalised L2 norm and proportional weight change across the convolution kernels and the CRF head. This is biologically plausible because the convolutional layers are closest to the pA normalised raw nanopore current signal and are therefore more likely to adapt to domain-specific features in the IgA Fc signal context, local k-mer behaviour, and upstream sequencing prep effects. `Alpha`'s downstream performance suggests that this interface recalibration was useful rather than disruptive.

The norm/deepnorm layers also showed an informative pattern. Although these layers contributed very little proportional weight change and had a very small normalised L2 norm, their p99.9 values were consistently around 0.31, with `Alpha` reaching 0.384. This indicates that a small number of normalisation-related parameters shifted moderately relative to the corresponding baseline spread. Such changes may reflect adjustments to activation scaling or stabilisation behaviour rather than a broad change in the internal transformer encoder

representation. By contrast, the $P_{99.9}(C)$ of the transformer self-attention and FFN layers were small and uniform, mostly around 0.05 and 0.04, respectively. Thus, FFN adaptation was broad and distributed, while convolutional and selected normalisation weights showed more concentrated tail movement.

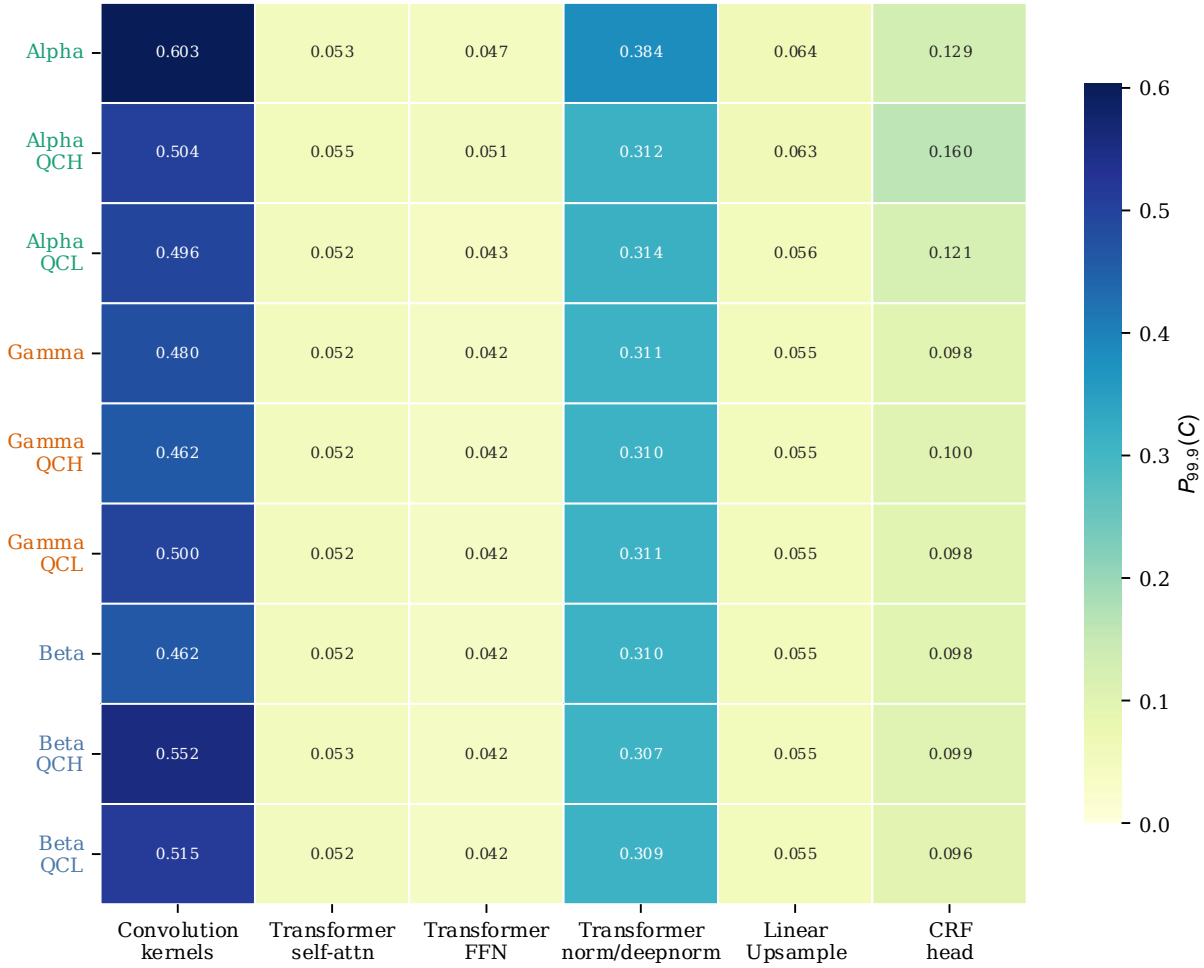

Figure 28. p99.9 standardised absolute weight change in different model components across models. *For each component, the y-axis shows the approximate 99.9th percentile of the absolute weight change, scaled by the standard deviation of the corresponding tensor in the* `sup` *baseline. This metric asks whether fine-tuning produced extreme parameter changes relative to the natural spread/variation of the pretrained weights.*

Figure 29 decomposes the normalised L2 norm by the transformer encoder block (i.e., Figure 26 showed little change, but the deeper question is whether the changes were uniform). Self-attention remained modest across all models, mostly around 0.010-0.013, with somewhat greater weight change near early and late blocks. This suggests that the baseline attention structure was mostly preserved, consistent with the view that the pretrained `sup` model already encoded useful general nanopore signal context. The FFN layers showed slightly larger and more systematic drift, approximately 0.011-0.015, with the drift tending to increase toward the final transformer blocks. This explains why the FFN dominated proportional weight change in

Figure 27; small, distributed changes across many parameters and blocks collectively account for a large proportional weight change. However, the relative magnitude remained modest, supporting the conclusion from Figure 28 that adaptation was controlled rather than disruptive.

The norm/deepnorm panel is particularly useful for interpreting the final decoding stage. Most norm/deepnorm layers changed slightly, usually by around 0.002-0.004, but block 17 showed a distinct increase across models, approximately 0.010-0.012. This suggests that the final transformer block required measurable recalibration of activation scaling immediately before the linear upsample and CRF head. In simple terms, the model did not need to memorise nucleotide-level signal context of IgA Fc WT, but it did need to adjust how the final context signal representation was transformed into output logits for the correct decoding of IgA Fc variant sequence.

Family-level patterns again caution against a simple “more change is better” interpretation. `Alpha` and `AlphaQCH` both showed somewhat higher normalised L2 norms in transformer encoder layers than `Gamma` and `Beta`, but `AlphaQCH` did not achieve the best downstream balance. Its stronger drift is consistent with the earlier evidence that it had begun to memorise the WT IgA Fc mapping: it achieved a high DMS pass rate and high model confidence, but weaker calibration and lower variant-abundance concordance than `Alpha`. Overall, Figure 29 reinforces the interpretation from Figures 26-28 that reliable fine-tuning depended on adaptation of the input interface and the output decoder head rather than on maximum weight change in the internal encoder backbone. In contrast, the minimal change in the transformer encoder highlights the importance of the signal transformation process, which the downstream CRF head and beam search decoding rely on.

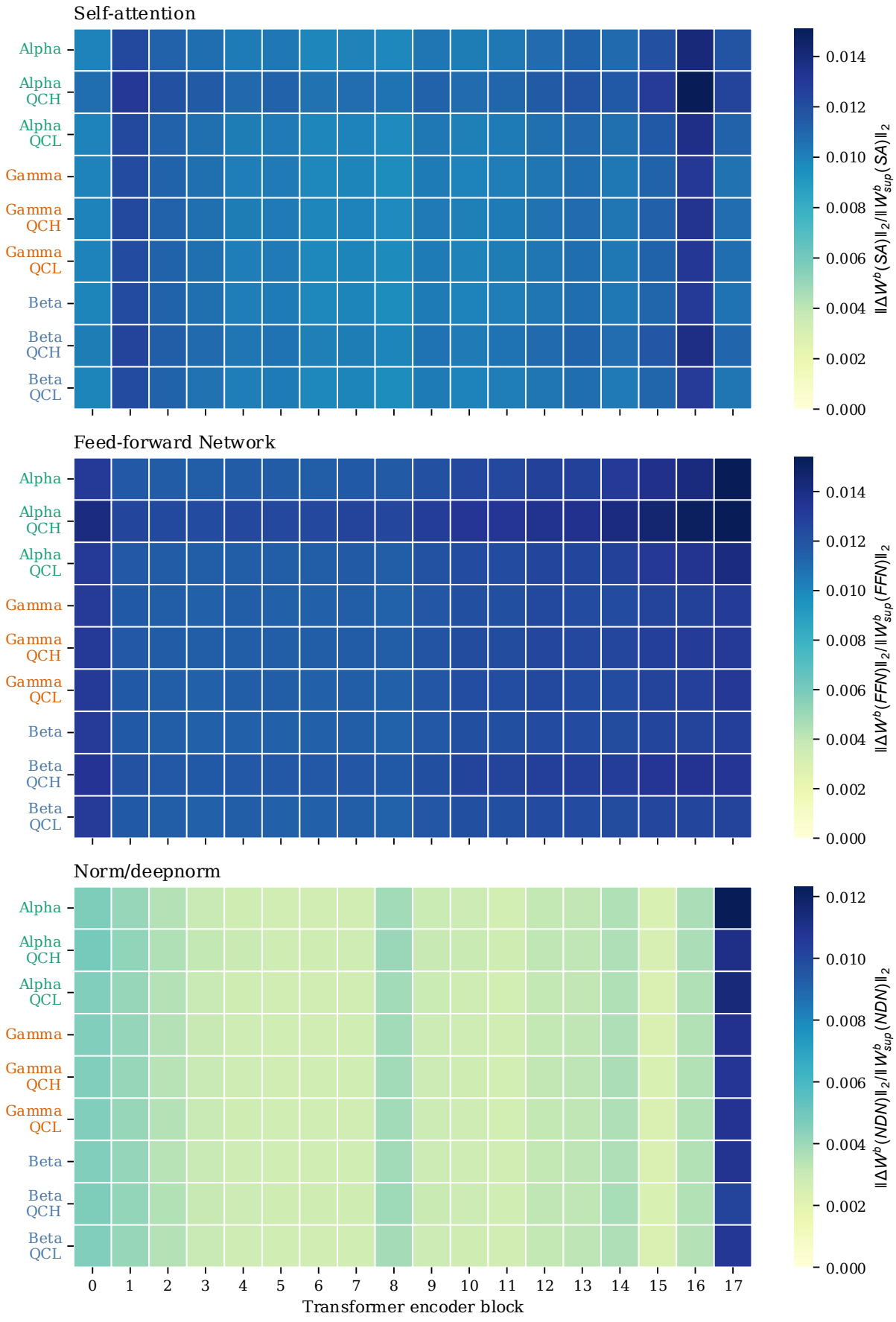
Self-attention
Feed-forward Network
Norm/deepnorm
Alpha
Alpha QCH
Alpha QCL
Gamma
Gamma QCH
Gamma QCL
Beta
Beta QCH
Beta QCL
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17
Transformer encoder block
0.014
0.012
0.010
0.008
0.006
0.004
0.002
0.000
$\|\Delta W^b(SA)\|_2/\|W^b_{sup}(SA)\|_2$
$\|\Delta W^b(FFN)\|_2/\|W^b_{sup}(FFN)\|_2$
$\|\Delta W^b(NDN)\|_2/\|W^b_{sup}(NDN)\|_2$

Figure 29. Block-level normalised L2 norm in all transformer encoder blocks across all models. *Transformer encoder blocks consist of self-attention, feed-forward, and norm/deepnorm layers. Cell colour shows block-level normalised L2 norm for each transformer encoder b, calculated separately for the self-attention (SA), feed-forward (FF), and norm/deepnorm layers using the same metric defined in Section 2.2.4.*

Figure 30 links the weight change in fine-tuned models to their downstream variant-calling performance. Panel A compares proportional weight change in transformer encoder blocks with pipeline pass rate and shows a moderate negative association (ps = -0.58). Models with more of their proportional weight changes concentrated in the transformer encoder tended to retain fewer reads through the full pipeline. This was most visible in the `Beta` family, where the base model had the largest proportional weight change in transformer encoder blocks (74.5%) but only modest pipeline retention, at approximately 19.9%. In contrast, `Alpha` had the lowest proportional weight changes in transformer encoder blocks (66.4%) and the highest pipeline pass rate, approximately 30.6%. This supports the interpretation that a broad transformer-concentrated weight change disrupted the fine-tuned model's variant-calling performance.

Panel B shows the complementary relationship between proportional weight changes in the convolution kernels, the CRF head and the weighted Jaccard similarity at a count threshold of 1. The association was moderately positive ($p_s = 0.58$), indicating that fine-tuning resulted in relatively more weight changes in the convolution kernels, and the CRF head tended to show stronger variant-abundance concordance with the short-read ground truth. `Alpha` occupied the most favourable region, with the highest combined proportional weight change in the convolution kernels and CRF head, at approximately 32.8%, and the strongest weighted Jaccard similarity of 0.240. This agrees with the component-level interpretation that the most useful fine-tuning recalibrated the convolution kernels and CRF head while preserving much of the pretrained transformer encoder backbone.

Panel C relates the model-level tail weight change $P_{99.9}(M)$ to the primary overall error rate. The negative association (p_s = -0.67) indicates that models with slightly larger $P_{99.9}(M)$ tended to have lower primary error rates. This should not be interpreted as evidence that extreme or unstable weight changes are beneficial. Figure 28 showed that the component-level $P_{99.9}(C)$ remained small relative to the natural weight spread within the tensors of the pretrained `sup` model. Within this stable regime, however, a larger change $P_{99.9}(M)$ likely indicates a useful adjustment to a small subset of sensitive parameters. The `Alpha` family models sit in this favourable region, whereas `Beta` and `Gamma` exhibit weaker $P_{99.9}(M)$ behaviour and a higher overall primary error rate.

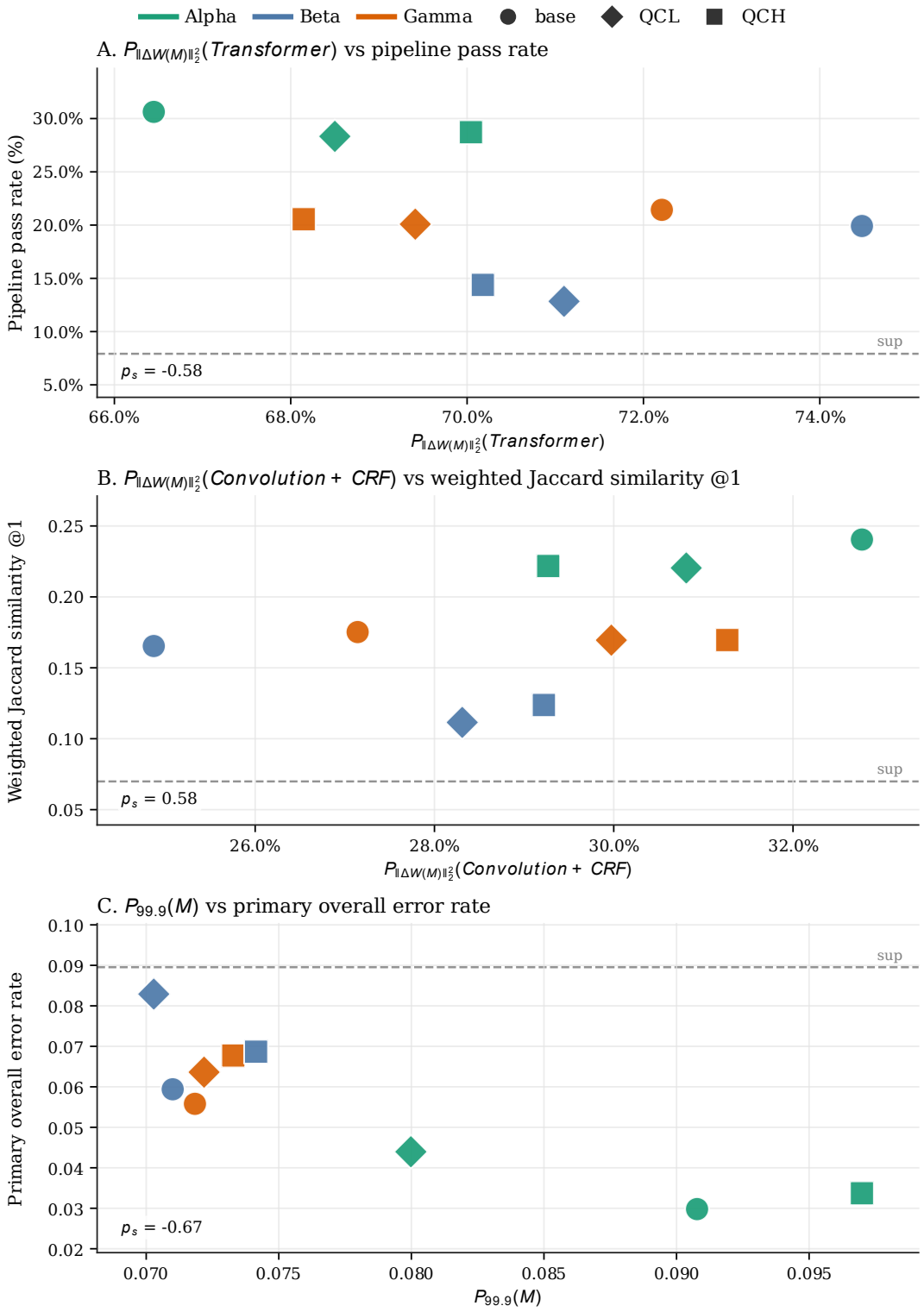

Figure 30. Association between model weight-change patterns and downstream performance. *Panels compare and link weight changes to downstream variant calling performance. Spearman's rank correlation coefficients in the figure are exploratory because only 9 fine-tuned models were available, and models from the same family are not independent due to partially shared training data.*

Together, Figures 26-30 mechanistically bridge the internal weight change of models during fine-tuning to their downstream variant-calling performance. These analyses showed that

domain-specific adaptation was small, stable, and structured. The strongest downstream model was not the one with the largest total weight change, the lowest validation loss, or the most transformer-concentrated drift. Instead, `Alpha` preserved much of the pretrained transformer encoder while making targeted changes to the convolution kernels, selected normalisation parameters, and the CRF head. This pattern helps explain why `Alpha` achieved the best balance of read retention, sequence noise reduction, primary overall error rate, and variant-abundance concordance.

These findings also suggest a practical direction for future domain-specific fine-tuning. If the goal is to preserve general-purpose basecalling performance while improving in a specific domain, architecture-aware fine-tuning should be tested rather than uniformly updating all weights. One strategy would be to freeze most transformer blocks and allow stronger controlled adaptation around the convolutional kernels, layer normalisation parameters, and CRF head. This would test whether `Alpha`-like adaptation can be deliberately reproduced, while avoiding the overfitting pattern observed in `AlphaQCH`. Such strategies should still be benchmarked using biologically meaningful downstream metrics rather than validation loss alone or model weight analysis, because the present results show that internal training diagnostics do not correlate well to downstream variant-calling performance.

3.6 Final Synthesis

Figure 31 summarises the final model comparison using the composite concordance score (CCS), which integrates multiple abundance-aware concordance metrics across variant count thresholds 1-10. Across the representative models, `Alpha` achieved the highest mean CCS, outperforming the strongest `Gamma`- and `Beta`-family models as well as the pretrained `sup` baseline. This result is consistent with the preceding analyses: `Alpha` did not dominate every intermediate metric, but it avoided the main failure modes that compromise downstream interpretation, including excessive primary error, framework leakage, abundance distortion, and over-pruning of the variant set. For DMS library sequencing, the balanced profile is more informative than dominance in a single read-level metric, because the goal is to recover an accurate representation of the underlying variant population rather than simply the cleanest individual reads.

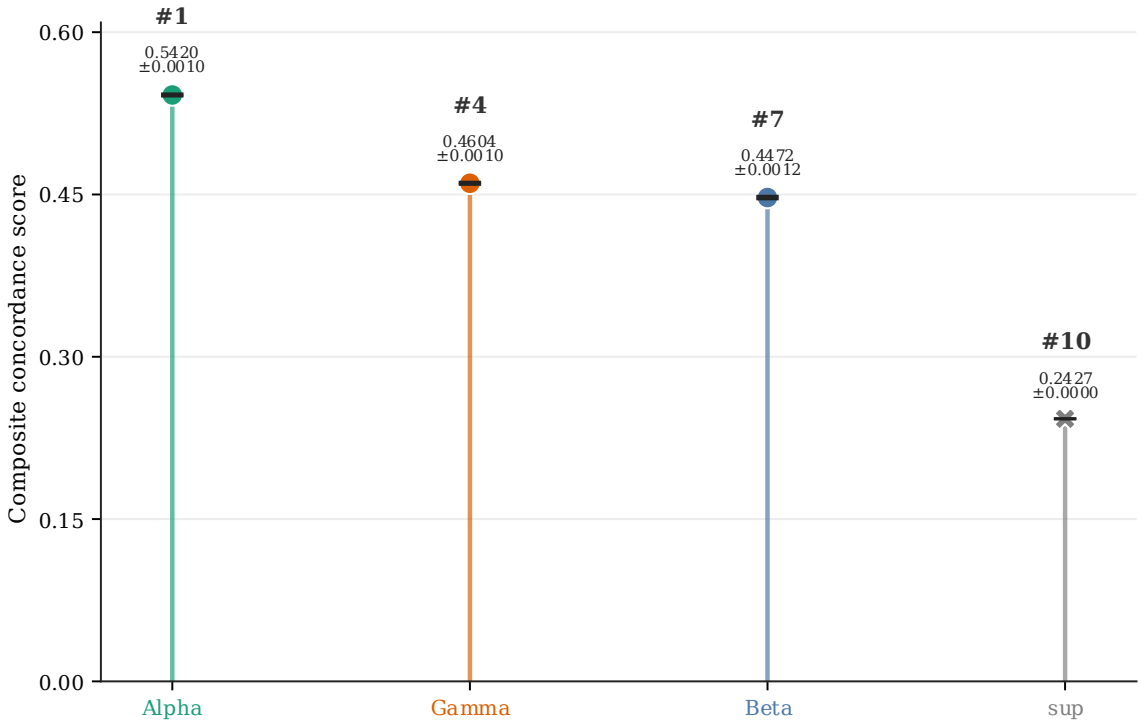

Figure 31. Mean composite concordance score (CCS) ranking. *Lollipop plot for the selected representative outputs. The figure shows the highest-ranked `Alpha`-, `Gamma`-, and `Beta`-family models together with the `sup` baseline. Ten different random seeds were used to train each model 10 times to account for non-determinism, and the ranks are computed as the mean composite concordance score (CCS) across seeded fine-tuning runs for each model. Each model displays the mean CCS under its rank, along with a 95% confidence interval. CCS is the weighted geometric mean of the normalised AUC scores for weighted Jaccard similarity, Jensen-Shannon similarity, exact overlap mass, and variant F1 across thresholds 1 to 10. Normalised AUC scores were computed from the threshold-response curves summarised in Figure 25. For CCS definition, see Equation (2.25) in Section 2.2.6. See Figure F3 for composite concordance ranking across all models.*

Because CCS is deterministic given a fixed, trained model and a benchmark dataset, the seeded analysis estimates the variability introduced by fine-tuning non-determinism, rather than the uncertainty in the sequencing experiment itself. `Alpha` retained the highest mean CCS across seeded runs, indicating that its ranking was robust to training-seed variation. However, the confidence intervals shown in Figure 31 should not be interpreted as confidence intervals over biological replication, sequencing runs, library preparation, or the underlying variant population. Generalising this ranking beyond the current test dataset would require additional uncertainty analyses, such as multinomial bootstrap resampling of reads or independent library preparation and sequencing replicates.

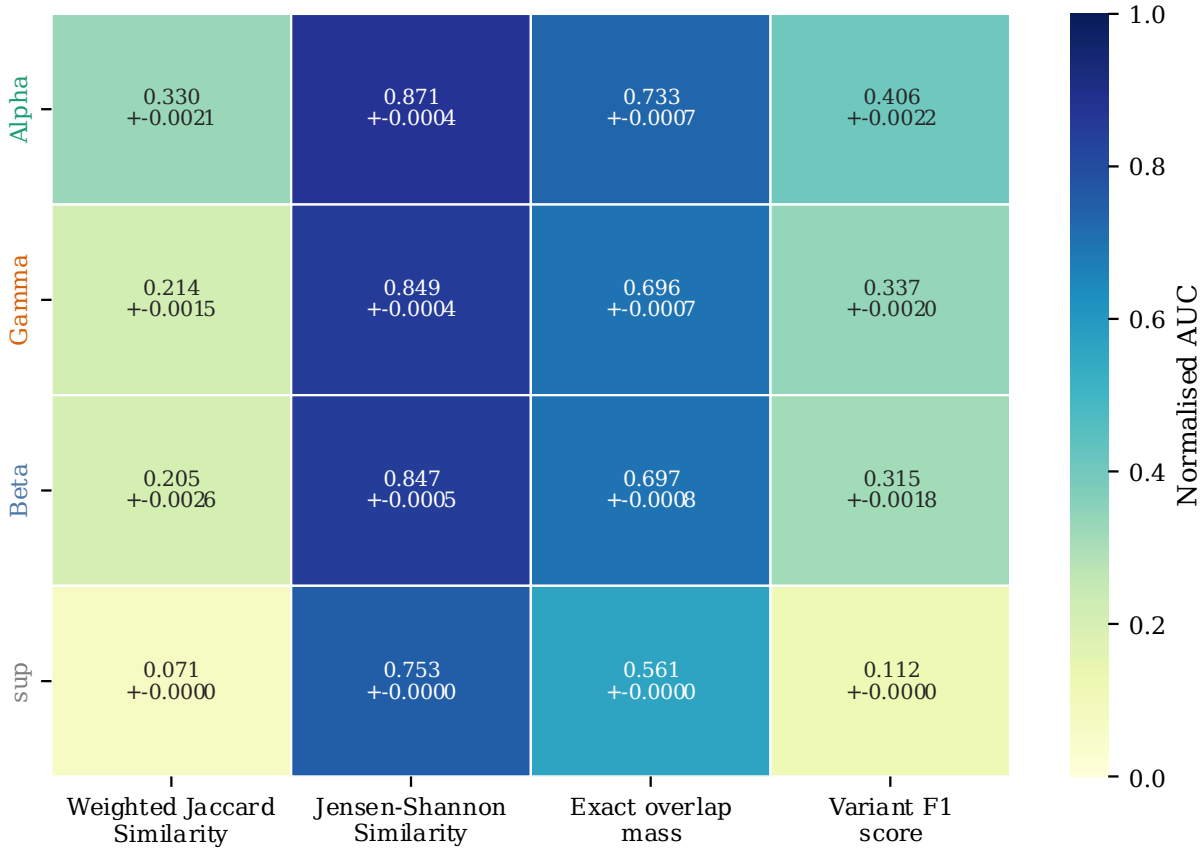

Figure 32. Mean Normalised AUC scores. *Heatmap of the mean normalised AUC scores underlying Figure 31. Each cell shows the mean normalised AUC [0,1] for one of four concordance metrics used to calculate mean CCS across 10 seeded model training runs, with the standard deviation shown below. AUC values were computed by trapezoidal integration of each metric across minimum count thresholds 1 to 10 and then normalised using the theoretical maximum AUC of 9. For all definitions of concordance metrics, see Section 2.2.6. See Figure F4 for the heatmap across all models.*

The component heatmap (Figure 32) indicates that `Alpha` achieved the highest normalised AUC across all four metrics, rather than deriving its composite score from dominance in any single term. In comparison, `AlphaQCH` and `AlphaQCL` demonstrated strengths in variant F1 score, but this was offset by weaker abundance-aware concordance (see Figure F4). `Gamma` and `Beta` produced near-identical profiles, differing only marginally in variant F1 (0.34 versus 0.32),

suggesting that these fine-tuning configurations converge to a similar performance floor. Both fell behind `Alpha`, most notably in weighted Jaccard Similarity and variant F1, indicating lower overlap quality and less robust read-level variant detection. The `sup` baseline trailed all fine-tuned models across every metric, confirming that domain-specific fine-tuning yields measurable gains. Together, these results support a practical guideline: when choosing among fine-tuned models, prioritise balanced performance across biologically meaningful metrics over the highest training performance or read retention.

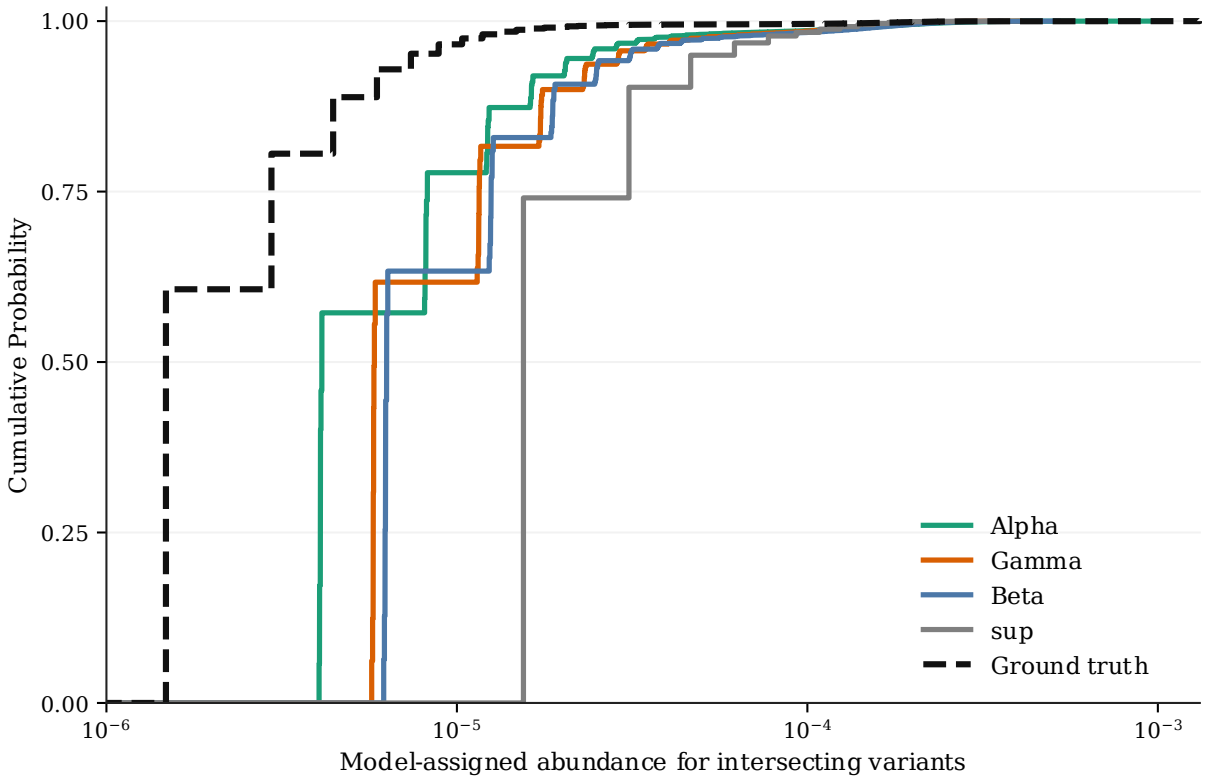

Figure 33. ECDF of model-assigned normalised abundance for non-WT variants retained at minimum count threshold 1. *For each abundance value on the x-axis, the y-axis shows the cumulative proportion of intersecting variants with model-assigned abundance less than or equal to that value. Solid-coloured curves show the model-assigned abundances for intersecting variants between each selected nanopore model and the ground truth. The dashed black curve shows the ground-truth abundance distribution as a reference. Abundances were normalised to sum to 1 within each dataset after WT removal, and model curves were computed after inner-joining on exact variant identity. Note that the ECDF presented here is derived by aggregating model-assigned abundance across all 10 seeded fine-tuning runs for each model. Slight differences in model performance across different seeded runs caused the curve to be uneven. On this ECDF, curves that rise further left indicate that a larger fraction of variants have low assigned abundance. In contrast, rightward-shifted curves indicate that the intersecting variants tend to have higher assigned abundances. For ECDF methodology, see Section 2.2.6. See Figure F5 for the corresponding ECDF figure across all models.*

Figure 33 examines how each model distributes assigned abundance across the basecalled library. The ground-truth curve rises earliest, reflecting the presence of low-abundance variants. All models shift rightward, indicating systematic overestimation of variant abundance; however, `Alpha` remains closest to the ground-truth distribution, while `sup` diverges the most. `Gamma` and `Beta` fall between the two with largely overlapping profiles.

This view is important in practice, given that in a DMS library with a long targeted protein sequence, a matched short-read ground truth will not be reproducible, so variant calling and filtering decisions depend directly on the model's own abundance estimates. A model that inflates abundance will compress dynamic range in enrichment scoring, reducing the ability to distinguish genuinely enriched variants from background noise. The tighter agreement between `Alpha` and the ground-truth distribution suggests it introduces less distortion in DMS library analysis, making it the safer option when only the model-assigned abundance is available.

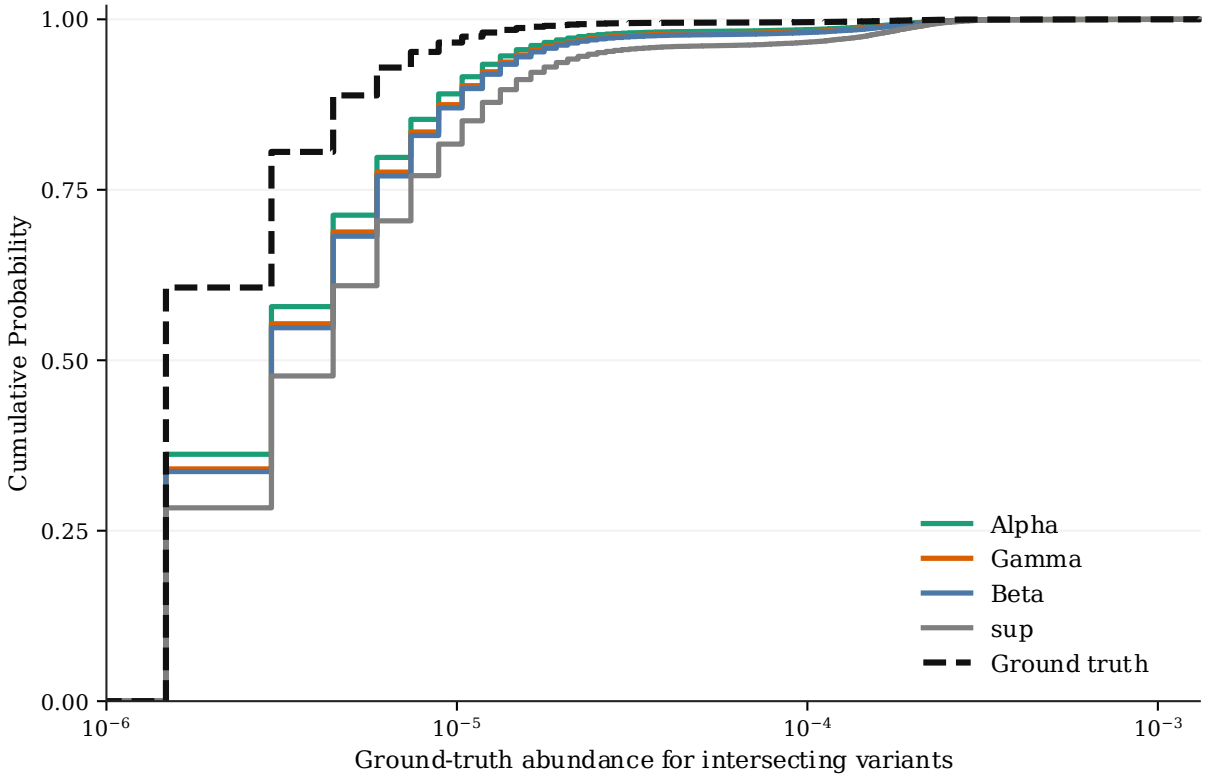

Figure 34. ECDF of ground truth normalised abundance for intersecting non-WT variants retained at minimum count threshold 1*. For each abundance value on the x-axis, the y-axis shows the cumulative proportion of intersecting variants with ground-truth abundance less than or equal to that value. Solid-coloured curves show the same selected representative model outputs from Figure 33, but the x-axis now gives the short-read ground-truth abundance of the variants recovered by each model. The dashed black curve shows the full short-read non-WT abundance distribution for reference. Ground-truth abundance was normalised to sum to 1 within each dataset after WT removal, and model curves were computed after inner joining the model and ground truth on exact variant identity. Note that the ECDF shown here is created by combining ground-truth abundance rather than model-assigned abundance across all 10 seeded fine-tuning runs for each model. This produces a smooth curve, unlike the pooled model-assigned ECDF in Figure 33. Curves shifted to the right relative to the dashed reference indicate that the variants recovered by that model are enriched for higher-abundance ground-truth variants, with fewer low-abundance variants from the true long tail. For ECDF methodology, see Section 2.2.6. See Figure F6 for the corresponding ECDF figure across all models.*

The empirical cumulative distribution function of ground-truth abundance for intersecting variants (Figure 34) reframes the comparison from the opposite direction, asking which portion of the true library each model successfully recovers. Here, the x-axis represents the known

ground-truth abundance of each variant, and a curve that tracks the dashed reference more closely indicates recovery across a wider abundance range. `Alpha` again follows the ground truth most faithfully, while `Gamma`, `Beta`, and especially `sup` shift toward higher-abundance variants, indicating significant loss of the low-abundance tail. This is a validation-oriented view: it confirms that the ranking observed in Figure 31 is not an artefact of any internal abundance assignment for any model but rather reflects genuine differences in variant-detection sensitivity. The practical consequence is that models with poorer tail recovery will underestimate the frequency of rare variants in the library, thereby narrowing the effective sequence space available for downstream structure–function analysis and variant follow-up.

Taken together, the benchmarking results show that domain-specific fine-tuning improved the biological applicability of the nanopore variant calling for the IgA Fc DMS library. The `Alpha`-family models provided the strongest balance of read retention, framework fidelity, DMS-filter performance, and abundance-aware concordance with the MGI reference. However, the remaining gap to short-read technologies was not attributable to a single source. It likely reflected the combined effects of training data quality, sequencing chemistry, sequence-context hotspots, and the tight constraints of the benchmark workflow that penalised biologically implausible deviations from the framework. These results show that read-level improvement alone is insufficient; the relevant endpoint is whether the recovered variant population remains interpretable for protein-engineering decisions.

A further limitation was that direct UMI-linked molecule-to-molecule benchmarking could not be performed in this study. The UMI-based design shown in Panel C of Figure 6 was intended to support cross-platform read comparison, and initial analysis confirmed the expected presence of dual-UMI tags at the ends of both nanopore and MGI-derived reads. However, no overlap between nanopore tag sets and the MGI UMI tag set was detected. The absence of overlap was most likely due to the high UMI diversity-to-CFU ratio (452,984,832/156,000 CFU = 2903), PCR-induced UMI errors, and library subsampling during the creation of comparison datasets. Specifically, the nanopore sequencing used the first-round PCR amplicon, and the short-read dataset used the second-round PCR amplicon. These factors prevented cross-platform UMI overlap, thereby preventing direct molecule-to-molecule comparison. Consequently, benchmarking was conducted using reference-based error profiling, DMS-filter retention, and variant-concordance analyses rather than direct molecule-level UMI matching. As anticipated, fine-tuned models showed reduced recovery of valid dual-

UMI tags compared with their broader basecalling behaviour, consistent with the trade-off introduced by domain-specific fine-tuning toward the IgA Fc target sequence rather than general-purpose sequence recovery. Relevant UMI metrics are reported separately in `umi.csv` on GitHub (Appendix E). Chapter 4, therefore, considers the broader implications of assay-aware basecalling and outlines future work needed to improve low-abundance variant recovery, UMI linked benchmarking, and experimental validation.

Chapter 4. Conclusion and Future Research

4.1 Conclusion

This thesis evaluated whether domain-specific fine-tuning of an Oxford Nanopore `sup` basecalling model could improve single-read variant calling for a dual-site IgA Fc DMS library. Fine-tuned models consistently outperformed the pretrained `sup` baseline, but the strongest evidence of improvement came from biologically relevant metrics rather than validation loss alone: pipeline read retention, DMS-filter performance, framework fidelity, and abundance-aware concordance with the MGI reference. The central contribution of this thesis is therefore twofold: it demonstrates that domain-specific fine-tuning can improve nanopore-based variant recovery, and it establishes a benchmark framework for judging whether basecaller outputs remain biologically plausible in a constrained DMS library.

The strongest result was obtained from the `Alpha` family, trained on restriction-digest WT Fc signal data. `Alpha` outperformed the PCR-trained `Beta` family, the mixed-data `Gamma` family, and the pretrained `sup` baseline across the main downstream metrics, indicating that clean, domain-matched training data were more valuable than larger or more aggressively filtered datasets. The QCL and QCH results across families further show that lower validation loss or higher nominal read quality did not necessarily translate into better variant calling. Although these filtered models often performed well on training diagnostics, they ranked below their corresponding base-family models in read retention, error profiles, framework noise reduction, and variant concordance. This suggests that overly strict filtering may remove signal variation needed for robust decoding of variant-rich test data. In this context, the most useful training data were not simply the cleanest subset, but the dataset that best balanced experimental cleanliness, signal diversity, and realistic noise exposure. This ranking of models was robust across the 10 seeded fine-tuning runs used for the final synthesis, although the reported 95% confidence interval reflects training-seed variation rather than independent sequencing, library-preparation, or biological replication.

The model-weight analysis supports this interpretation from a mechanistic perspective. The advantage of `Alpha` was not associated with the largest overall model change, but with a stable, targeted pattern of adaptation involving the convolution kernels and CRF head, while largely preserving the pretrained transformer encoder. This suggests that successful fine-

tuning did not overwrite the entire pretrained basecaller. Instead, it recalibrated the parts of the model most relevant to interpreting the IgA Fc signal context and producing biologically plausible variant calls.

These findings support the broader use of assay-aware basecalling for DMS libraries, particularly as target sequences become too long for short-read sequencing. Nanopore reads can preserve linkage between distant mutations, making them attractive for longer sequence constructs and other multi-site libraries. However, this advantage depends on reducing long-read errors enough to support variant-level data interpretation. The `Alpha` results show that domain-specific fine-tuning moves nanopore output closer to that requirement, while residual framework hotspots, indel-driven DMS-filter failures, and incomplete recovery of the low-abundance variant tail show that further improvement is still needed.

Overall, this project establishes a reproducible workflow linking library validation, low-cost WT nanopore training data, model fine-tuning, assay-aware filtering, single-read correction, and short-read benchmarking in an antibody engineering context. For protein engineering applications, the relevant question is not only whether a basecaller reduces errors but also whether it preserves the library's mutational constraints, abundance structure, and interpretability for downstream data analysis and decision-making.

4.2 Future Research

Future research should first improve model training while preserving one of the central practical findings of this thesis: experimentally simple WT IgA Fc nanopore signal data can provide a useful domain-specific signal for basecaller fine-tuning. Rather than constructing a dedicated variant library solely for training, a more appropriate next step would be to test whether computationally augmenting WT signals can improve generalisation to variant-rich samples. One approach would be to introduce synthetic single-nucleotide substitutions, codon-level changes, or local sequence perturbations into mutation regions, while preserving realistic k-mer-dependent nanopore signal structure. This would test whether controlled exposure to mutations improves residual basecalling failure modes, including indel errors, false-positive mutations, mutation-region coverage, off-library codons, and incomplete recovery of the low-abundance variant tail. In addition, the model weight analysis motivates architecture-aware fine-tuning strategies. Since the strongest downstream performance was associated with preserving transformer weights and targeted adaptation around the convolutional kernels and

CRF head, future work should test whether freezing most transformer blocks while allowing weight changes to selected convolutional layers and the CRF head improves domain-specific variant calling. Together, these approaches would clarify whether the remaining gap relative to the MGI reference reflects limited exposure to variant-rich signal contexts, model-architecture limitations, or constraints imposed by the sequencing chemistry and library-preparation workflow.

A second priority is to revisit UMI-linked benchmarking using a library with more realistic UMI diversity. In the present study, exact cross-platform UMI overlap was absent because the dual-UMI space greatly exceeded the effective number of sequenced molecules and was further reduced by PCR error and subsampling. Future work should redesign the UMI scheme to better match the expected number of unique starting molecules while maintaining headroom to avoid excessive UMI collisions (Andersson et al., 2024; Clement et al., 2018). In addition, the same amplicon pool should be divided earlier into parallel ONT and short-read library preparation, so that both platforms sample the same Fc fragment population before later amplification steps introduce divergence. With this design, the custom `UmiExtractor`, `UmiOverlap`, and `Benchmarker` tools developed in this thesis could be used to perform direct molecule-to-molecule comparisons between platforms.

Such an experiment would add a benchmarking layer that was not possible in the present study. Population-level concordance can indicate whether variant distributions agree, but it cannot determine whether matching abundance profiles arise from true-positive molecules or from false positives and false negatives cancelling each other out. Matched UMI pairs would allow calculation of nucleotide-level precision, recall, F1 scores; quantification of indels, substitutions, and truncations of bases; and provide a direct test of whether per-read quality scores reflect true error rates (Deng et al., 2024; Karst et al., 2021; Zurek et al., 2020).

The next priority is to deploy the best-performing model in a real dual-site DMS screening workflow with experimental validation. `Alpha` was identified as the strongest model for reconstructing the dual-site IgA Fc library, but the evaluation in this thesis was primarily based on sequencing concordance rather than functional validation. Sequencing real pre- and post-screen libraries would enable estimation of enrichment scores across recovered variants, after which representative clones could be selected for wet-lab validation. These should cover highly enriched hits, borderline variants, and structurally diverse mutational patterns. Testing these clones experimentally, for example, by phage-display ELISA, would determine whether

nanopore-derived variant rankings are biologically meaningful rather than merely computationally concordant.

If this experimental validation succeeds, the resulting enrichment profiles and experimentally confirmed binders could form a labelled dataset for predictive modelling of the IgA Fc domain. An initial discriminative model could be trained to predict variant retention or enrichment after screening. With a sufficient number of experimentally classified variants, a later generative model could then propose new Fc candidates within the constrained multi-site design space (Irvine et al., 2025).

Another potential study is to extend the benchmark framework to longer antibody constructs that further exceed the practical limits of short-read technologies. For example, extending the construct to a full-length IgA heavy chain (V_H–C_{H1}–hinge–C_{H2}–C_{H3}), which encodes approximately 450 amino acids (~1,350 bp of coding sequence), places it well beyond paired-end short-read coverage in a single contiguous read. Although emerging 2 x 500 bp short-read chemistries may extend this range, their accuracy and practical suitability for the DMS application remain uncertain. At this length, nanopore sequencing becomes essential for preserving linkage between distant mutations, particularly for capturing cross-domain epistatic interactions between Fc mutations and residues in the V_H, C_{H1}, or hinge regions that may influence immune receptor engagement, effector function, or structural stability. Studying such interactions is biologically important because FcαRI binding occurs at the C_{H2}–C_{H3} junction yet can propagate long-range conformational effects (Posgai et al., 2018), and hinge-region variation is known to affect susceptibility to bacterial IgA1 proteases and the molecular geometry of Fc receptor engagement.

However, extending from Fc to a full heavy-chain construct would require reconsideration of the display platform. The current IgA Fc phage display library uses M13-based phagemid display with Fc–pIII fusion, which is well-suited for the Fc fragment because it folds independently as a homodimer without requiring a light chain partner. M13 phage display has been shown to tolerate fusion proteins up to approximately 100 kDa, including bivalently linked Fab fragments (Lee et al., 2004). A full-length heavy chain introduces the V_H and C_{H1} domains, which normally require pairing with a cognate light chain for proper folding and structural stability. In the absence of a light chain, the unpaired V_H–C_{H1} segment is prone to misfolding and aggregation in the oxidising E. coli periplasm.

Yeast display is a natural alternative for this extension. Yeast display supports eukaryotic protein folding and disulfide bond formation, enables co-expression of paired heavy and light chains, and permits FACS-based screening of displayed antibody variants. Full-length IgG display and library screening on yeast has been demonstrated, for example, through the REAL-Select system, in which secreted native full-length antibodies are captured on the Saccharomyces cerevisiae cell surface via an immobilised ZZ domain, a tandem repeat of engineered Protein A-derived Z domains that bind the IgG Fc region, enabling FACS-based selection of functional full-length antibodies without requiring fusion of the heavy chain to an anchor protein. The REAL-Select ZZ-domain display strategy is designed around IgG Fc binding and should not be assumed to translate to IgA. In addition, IgA Fc recognition involves distinct receptor interactions (e.g., FcαRI), particularly at the C_{H2}/C_{H3} interface (Bakema & van Egmond, 2011; Pleass et al., 1999). Therefore, a full-length IgA yeast display system would require an alternative display strategy to preserve the IgA Fc receptor-binding surface. Although YSD library sizes (~10^5–10^7) are smaller than the phage display size limit, they are sufficient for designed DMS libraries where diversity is constrained by the number of mutational positions and codon schemes, and the FACS-based quantitative readout can directly replace phage panning-based enrichment. Transitioning to a yeast-displayed full-length IgA heavy-chain library would generate fragments compatible only with long-read sequencing, directly validating the domain-specific fine-tuning framework established in this thesis and enabling analysis of cross-domain interactions that are not detectable with the current Fc construct.

Together, these projects define a coherent research programme that extends from computational augmentation of WT IgA Fc nanopore signal data and architecture-aware domain-specific fine-tuning, through UMI-linked molecule-level benchmarking, to experimental screening validation, predictive modelling, and longer antibody constructs. In doing so, the work shifts from isolated assessment of nanopore basecalling models toward an integrated experimental-computational feedback loop in which sequencing, model training, screening, and protein design iteratively inform one another.

References

Adams, R. M., Kinney, J. B., Walczak, A. M., & Mora, T. (2019). Epistasis in a Fitness Landscape Defined by Antibody-Antigen Binding Free Energy. *Cell Systems*, *8*(1), 86-93.e3. https://doi.org/10.1016/j.cels.2018.12.004

Adams, R. M., Mora, T., Walczak, A. M., & Kinney, J. B. (2016). Measuring the sequence-affinity landscape of antibodies with massively parallel titration curves. *eLife*, *5*, e23156. https://doi.org/10.7554/eLife.23156

Aird, D., Ross, M. G., Chen, W.-S., Danielsson, M., Fennell, T., Russ, C., Jaffe, D. B., Nusbaum, C., & Gnirke, A. (2011). Analyzing and minimizing PCR amplification bias in Illumina sequencing libraries. *Genome Biology*, *12*(2), R18. https://doi.org/10.1186/gb-2011-12-2-r18

Andersson, D., Kebede, F. T., Escobar, M., Österlund, T., & Ståhlberg, A. (2024). Principles of digital sequencing using unique molecular identifiers. *Molecular Aspects of Medicine*, *96*, 101253. https://doi.org/10.1016/j.mam.2024.101253

Bakema, J. E., & van Egmond, M. (2011). The human immunoglobulin A Fc receptor FcαRI: A multifaceted regulator of mucosal immunity. *Mucosal Immunology*, *4*(6), 612–624. https://doi.org/10.1038/mi.2011.36

Bashir, S., & Paeshuyse, J. (2020). Construction of Antibody Phage Libraries and Their Application in Veterinary Immunovirology. *Antibodies*, *9*(2), 21. https://doi.org/10.3390/antib9020021

Beg, M., Taka, J., Kluyver, T., Konovalov, A., Ragan-Kelley, M., Thiéry, N. M., & Fangohr, H. (2021). Using Jupyter for reproducible scientific workflows. *Computing in Science & Engineering*, *23*(2), 36–46. https://doi.org/10.1109/MCSE.2021.3052101

Ben Mkaddem, S., Rossato, E., Heming, N., & Monteiro, R. C. (2013). Anti-inflammatory role of the IgA Fc receptor (CD89): From autoimmunity to therapeutic perspectives. *Autoimmunity Reviews*, *12*(6), 666–669. https://doi.org/10.1016/j.autrev.2012.10.011

Bentley, D. R., Balasubramanian, S., Swerdlow, H. P., Smith, G. P., Milton, J., Brown, C. G., Hall, K. P., Evers, D. J., Barnes, C. L., Bignell, H. R., Boutell, J. M., Bryant, J., Carter, R. J., Keira Cheetham, R., Cox, A. J., Ellis, D. J., Flatbush, M. R., Gormley, N. A., Humphray, S. J., … Smith, A. J. (2008). Accurate whole human genome sequencing using reversible terminator chemistry. *Nature*, *456*(7218), 53–59. https://doi.org/10.1038/nature07517

Bird, J. E., Marles-Wright, J., & Giachino, A. (2022). A User's Guide to Golden Gate Cloning Methods and Standards. *ACS Synthetic Biology*, *11*(11), 3551–3563. https://doi.org/10.1021/acssynbio.2c00355

Bloom, J. D. (2014). An Experimentally Determined Evolutionary Model Dramatically Improves Phylogenetic Fit. *Molecular Biology and Evolution*, *31*(8), 1956–1978. https://doi.org/10.1093/molbev/msu173

Boder, E. T., & Wittrup, K. D. (1997). Yeast surface display for screening combinatorial polypeptide libraries. *Nature Biotechnology*, *15*(6), 553–557. https://doi.org/10.1038/nbt0697-553

Boross, P., Lohse, S., Nederend, M., Jansen, J. H. M., van Tetering, G., Dechant, M., Peipp, M., Royle, L., Liew, L. P., Boon, L., van Rooijen, N., Bleeker, W. K., Parren, P. W. H.

I., van de Winkel, J. G. J., Valerius, T., & Leusen, J. H. W. (2013). IgA EGFR antibodies mediate tumour killing in vivo. *EMBO Molecular Medicine*, *5*(8), 1213–1226. https://doi.org/10.1002/emmm.201201929

Boža, V., Brejová, B., & Vinař, T. (2017). DeepNano: Deep Recurrent Neural Networks for Base Calling in MinION Nanopore Reads. *PLOS ONE*, *12*(6), e0178751. https://doi.org/10.1371/journal.pone.0178751

Chen, F., Dong, M., Ge, M., Zhu, L., Ren, L., Liu, G., & Mu, R. (2013). The History and Advances of Reversible Terminators Used in New Generations of Sequencing Technology. *Genomics, Proteomics & Bioinformatics*, *11*(1), 34–40. https://doi.org/10.1016/j.gpb.2013.01.003

Clement, K., Farouni, R., Bauer, D. E., & Pinello, L. (2018). AmpUMI: Design and analysis of unique molecular identifiers for deep amplicon sequencing. *Bioinformatics*, *34*(13), i202–i210. https://doi.org/10.1093/bioinformatics/bty264

Çubuk, H., Jin, X., Phipson, B., Marsh, J. A., & Rubin, A. F. (2025). Variant scoring tools for deep mutational scanning. *Molecular Systems Biology*, *21*(10), 1293–1305. https://doi.org/10.1038/s44320-025-00137-x

Danecek, P., Bonfield, J. K., Liddle, J., Marshall, J., Ohan, V., Pollard, M. O., Whitwham, A., Keane, T., McCarthy, S. A., Davies, R. M., & Li, H. (2021). Twelve years of SAMtools and BCFtools. *GigaScience*, *10*(2), giab008. https://doi.org/10.1093/gigascience/giab008

David, M., Dursi, L. J., Yao, D., Boutros, P. C., & Simpson, J. T. (2017). Nanocall: An open source basecaller for Oxford Nanopore sequencing data. *Bioinformatics*, *33*(1), 49–55. https://doi.org/10.1093/bioinformatics/btw569

de Sousa-Pereira, P., & Woof, J. M. (2019). IgA: Structure, Function, and Developability. *Antibodies*, *8*(4), 57. https://doi.org/10.3390/antib8040057

Delahaye, C., & Nicolas, J. (2021). Sequencing DNA with nanopores: Troubles and biases. *PLoS ONE*, *16*(10), e0257521. https://doi.org/10.1371/journal.pone.0257521

Deng, D. Z. Q., Verhage, J., Neudorf, C., Corbett-Detig, R., Mekonen, H., Castaldi, P. J., & Vollmers, C. (2024). R2C2 + UMI: Combining concatemeric and unique molecular identifier–based consensus sequencing enables ultra-accurate sequencing of amplicons on Oxford Nanopore Technologies sequencers. *PNAS Nexus*, *3*(9), pgae336. https://doi.org/10.1093/pnasnexus/pgae336

Doud, M. B., & Bloom, J. D. (2016). Accurate Measurement of the Effects of All Amino-Acid Mutations on Influenza Hemagglutinin. *Viruses*, *8*(6), 155. https://doi.org/10.3390/v8060155

Engler, C., Gruetzner, R., Kandzia, R., & Marillonnet, S. (2009). Golden Gate Shuffling: A One-Pot DNA Shuffling Method Based on Type IIs Restriction Enzymes. *PLOS ONE*, *4*(5), e5553. https://doi.org/10.1371/journal.pone.0005553

Faure, A. J., Schmiedel, J. M., Baeza-Centurion, P., & Lehner, B. (2020). DiMSum: An error model and pipeline for analyzing deep mutational scanning data and diagnosing common experimental pathologies. *Genome Biology*, *21*, 207. https://doi.org/10.1186/s13059-020-02091-3

Ferguson, S., McLay, T., Andrew, R. L., Bruhl, J. J., Schwessinger, B., Borevitz, J., & Jones, A. (2022). Species-specific basecallers improve actual accuracy of nanopore sequencing in plants. *Plant Methods*, *18*, 137. https://doi.org/10.1186/s13007-022-00971-2

Fowler, D. M., & Fields, S. (2014). Deep mutational scanning: A new style of protein science. *Nature Methods*, *11*(8), 801–807. https://doi.org/10.1038/nmeth.3027

Gao, P., Morita, N., & Shinkura, R. (2024). Role of mucosal IgA antibodies as novel therapies to enhance mucosal barriers. *Seminars in Immunopathology*, *47*(1), 1. https://doi.org/10.1007/s00281-024-01027-4

Hanning, K. R., Minot, M., Warrender, A. K., Kelton, W., & Reddy, S. T. (2022). Deep mutational scanning for therapeutic antibody engineering. *Trends in Pharmacological Sciences*, *43*(2), 123–135. https://doi.org/10.1016/j.tips.2021.11.010

Hanning, K. R., Walker, E. J., Beijerling, K., Irvine, E. B., Steel, J. J., & Kelton, W. (2025). *Simple high-throughput encoding of deep mutational scanning libraries by oligo-based Golden Gate assembly* (p. 2025.07.16.665225). bioRxiv. https://doi.org/10.1101/2025.07.16.665225

Harris, C. R., Millman, K. J., Walt, S. J. van der, Gommers, R., Virtanen, P., Cournapeau, D., Wieser, E., Taylor, J., Berg, S., Smith, N. J., Kern, R., Picus, M., Hoyer, S., Kerkwijk, M. H. van, Brett, M., Haldane, A., Río, J. F. del, Wiebe, M., Peterson, P., … Oliphant, T. E. (2020). Array Programming with NumPy. *Nature*, *585*(7825), 357–362. https://doi.org/10.1038/s41586-020-2649-2

Hon, T., Mars, K., Young, G., Tsai, Y.-C., Karalius, J. W., Landolin, J. M., Maurer, N., Kudrna, D., Hardigan, M. A., Steiner, C. C., Knapp, S. J., Ware, D., Shapiro, B., Peluso, P., & Rank, D. R. (2020). Highly accurate long-read HiFi sequencing data for five complex genomes. *Scientific Data*, *7*, 399. https://doi.org/10.1038/s41597-020-00743-4

Hunter, J. D. (2007). Matplotlib: A 2D Graphics Environment. *Computing in Science & Engineering*, *9*(3), 90–95. https://doi.org/10.1109/MCSE.2007.55

Irvine, E. B., Bikias, T., Stamkopoulos, E., Frei, L., Schürmann, N., Warrender, A. K., Schmid, H., Coukos, D., Yang, H., Minot, M., Kelton, W., & Reddy, S. T. (2025). *Generative design of antibody Fc-variants with synthetic and programmable functional profiles* (p. 2025.10.10.681689). bioRxiv. https://doi.org/10.1101/2025.10.10.681689

Jann, J., Gagnon-Arsenault, I., Pageau, A., Dubé, A. K., Fijarczyk, A., Durand, R., & Landry, C. R. (2026). A cost-effective and scalable barcoded library construction method for deep mutational scanning studies. *PLoS Biology*, *24*(2), e3003645. https://doi.org/10.1371/journal.pbio.3003645

Jaroszewicz, W., Morcinek-Orłowska, J., Pierzynowska, K., Gaffke, L., & Węgrzyn, G. (2022). Phage display and other peptide display technologies. *FEMS Microbiology Reviews*, *46*(2), fuab052. https://doi.org/10.1093/femsre/fuab052

Johnson, M. S., Venkataram, S., & Kryazhimskiy, S. (2023). Best Practices in Designing, Sequencing, and Identifying Random DNA Barcodes. *Journal of Molecular Evolution*, *91*(3), 263–280. https://doi.org/10.1007/s00239-022-10083-z

Kaplon, H., Muralidharan, M., Schneider, Z., & Reichert, J. M. (2020). Antibodies to watch in 2020. *mAbs*, *12*(1), 1703531. https://doi.org/10.1080/19420862.2019.1703531

Karst, S. M., Ziels, R. M., Kirkegaard, R. H., Sørensen, E. A., McDonald, D., Zhu, Q., Knight, R., & Albertsen, M. (2021). High-accuracy long-read amplicon sequences using unique molecular identifiers with Nanopore or PacBio sequencing. *Nature Methods*, *18*(2), 165–169. https://doi.org/10.1038/s41592-020-01041-y

Kirby, M. B., & Whitehead, T. A. (2022). Facile Assembly of Combinatorial Mutagenesis Libraries using Nicking Mutagenesis. *Methods in Molecular Biology (Clifton, N.J.)*, *2461*, 85–109. https://doi.org/10.1007/978-1-0716-2152-3_6

Kitware, Inc. (2026). *Kitware/CMake* [C]. https://github.com/Kitware/CMake (Original work published 2010)

Lee, C. V., Sidhu, S. S., & Fuh, G. (2004). Bivalent antibody phage display mimics natural immunoglobulin. *Journal of Immunological Methods*, *284*(1), 119–132. https://doi.org/10.1016/j.jim.2003.11.001

Li, H. (2018). Minimap2: Pairwise alignment for nucleotide sequences. *Bioinformatics*, *34*(18), 3094–3100. https://doi.org/10.1093/bioinformatics/bty191

Liu-Wei, W., van der Toorn, W., Bohn, P., Hölzer, M., Smyth, R. P., & von Kleist, M. (2024). Sequencing accuracy and systematic errors of nanopore direct RNA sequencing. *BMC Genomics*, *25*(1), 528. https://doi.org/10.1186/s12864-024-10440-w

Macpherson, A. J., McCoy, K. D., Johansen, F.-E., & Brandtzaeg, P. (2008). The immune geography of IgA induction and function. *Mucosal Immunology*, *1*(1), 11–22. https://doi.org/10.1038/mi.2007.6

Mamedov, T. G., Pienaar, E., Whitney, S. E., TerMaat, J. R., Carvill, G., Goliath, R., Subramanian, A., & Viljoen, H. J. (2008). A fundamental study of the PCR amplification of GC-rich DNA templates. *Computational Biology and Chemistry*, *32*(6), 452–457. https://doi.org/10.1016/j.compbiolchem.2008.07.021

Matreyek, K. A., Starita, L. M., Stephany, J. J., Martin, B., Chiasson, M. A., Gray, V. E., Kircher, M., Khechaduri, A., Dines, J. N., Hause, R. J., Bhatia, S., Evans, W. E.,

Relling, M. V., Yang, W., Shendure, J., & Fowler, D. M. (2018). Multiplex assessment of protein variant abundance by massively parallel sequencing. *Nature Genetics*, *50*(6), 874–882. https://doi.org/10.1038/s41588-018-0122-z

McKinney, W. (2011). *pandas: A Foundational Python Library for Data Analysis and Statistics*. https://www.semanticscholar.org/paper/pandas:-a-Foundational-Python-Library-for-Data-and-McKinney/1a62eb61b2663f8135347171e30cb9dc0a8931b5

Meyer, S., Nederend, M., Jansen, J. H. M., Reiding, K. R., Jacobino, S. R., Meeldijk, J., Bovenschen, N., Wuhrer, M., Valerius, T., Ubink, R., Boross, P., Rouwendal, G., & Leusen, J. H. W. (2016). Improved in vivo anti-tumor effects of IgA-Her2 antibodies through half-life extension and serum exposure enhancement by FcRn targeting. *mAbs*, *8*(1), 87–98. https://doi.org/10.1080/19420862.2015.1106658

OpenGene - Open Source Genomics Toolbox. (2026). *OpenGene/fastplong* [C++]. https://github.com/OpenGene/fastplong (Original work published 2024)

Oxford Nanopore Technologies. (2026). *Nanoporetech/bonito* [Python]. https://github.com/nanoporetech/bonito (Original work published 2019)

Oxford Nanopore Technologies. (2026a). *Nanoporetech/pod5-file-format* [C++]. https://github.com/nanoporetech/pod5-file-format (Original work published 2022)

Oxford Nanopore Technologies. (2026b). *Nanoporetech/dorado* [C++]. https://github.com/nanoporetech/dorado (Original work published 2022)

Pagès-Gallego, M., & de Ridder, J. (2023). Comprehensive benchmark and architectural analysis of deep learning models for nanopore sequencing basecalling. *Genome Biology*, *24*, 71. https://doi.org/10.1186/s13059-023-02903-2

Pleass, R. J., Dunlop, J. I., Anderson, C. M., & Woof, J. M. (1999). Identification of residues in the CH2/CH3 domain interface of IgA essential for interaction with the human fcalpha receptor (FcalphaR) CD89. *The Journal of Biological Chemistry*, *274*(33), 23508–23514. https://doi.org/10.1074/jbc.274.33.23508

Posgai, M. T., Tonddast-Navaei, S., Jayasinghe, M., Ibrahim, G. M., Stan, G., & Herr, A. B. (2018). FcαRI binding at the IgA1 C_H2-C_H3 interface induces long-range conformational changes that are transmitted to the hinge region. *Proceedings of the National Academy of Sciences of the United States of America*, *115*(38), E8882–E8891. https://doi.org/10.1073/pnas.1807478115

Püllmann, P., Ulpinnis, C., Marillonnet, S., Gruetzner, R., Neumann, S., & Weissenborn, M. J. (2019). Golden Mutagenesis: An efficient multi-site-saturation mutagenesis approach by Golden Gate cloning with automated primer design. *Scientific Reports*, *9*(1), 10932. https://doi.org/10.1038/s41598-019-47376-1

Rasila, T. S., Pajunen, M. I., & Savilahti, H. (2009). Critical evaluation of random mutagenesis by error-prone polymerase chain reaction protocols, *Escherichia coli* mutator strain, and hydroxylamine treatment. *Analytical Biochemistry*, *388*(1), 71–80. https://doi.org/10.1016/j.ab.2009.02.008

Rhoads, A., & Au, K. F. (2015). PacBio Sequencing and Its Applications. *Genomics, Proteomics & Bioinformatics, SI: Metagenomics of Marine Environments*, *13*(5), 278–289. https://doi.org/10.1016/j.gpb.2015.08.002

Samarakoon, H., Wan, Y. K., Parameswaran, S., Göke, J., Gamaarachchi, H., & Deveson, I. W. (2025). Leveraging basecaller's move table to generate a lightweight k-mer model

for nanopore sequencing analysis. *Bioinformatics*, *41*(4), btaf111. https://doi.org/10.1093/bioinformatics/btaf111

Schaffitzel, C., Hanes, J., Jermutus, L., & Plückthun, A. (1999). Ribosome display: An in vitro method for selection and evolution of antibodies from libraries. *Journal of Immunological Methods*, *231*(1), 119–135. https://doi.org/10.1016/S0022-1759(99)00149-0

Sellés Vidal, L., Isalan, M., Heap, J. T., & Ledesma-Amaro, R. (2023). A primer to directed evolution: Current methodologies and future directions. *RSC Chemical Biology*, *4*(4), 271–291. https://doi.org/10.1039/d2cb00231k

Sereika, M., Kirkegaard, R. H., Karst, S. M., Michaelsen, T. Y., Sørensen, E. A., Wollenberg, R. D., & Albertsen, M. (2022). Oxford Nanopore R10.4 long-read sequencing enables the generation of near-finished bacterial genomes from pure cultures and metagenomes without short-read or reference polishing. *Nature Methods*, *19*(7), 823–826. https://doi.org/10.1038/s41592-022-01539-7

Slatko, B. E., Gardner, A. F., & Ausubel, F. M. (2018). Overview of Next Generation Sequencing Technologies. *Current Protocols in Molecular Biology*, *122*(1), e59. https://doi.org/10.1002/cpmb.59

Slavny, P., Hegde, M., Doerner, A., Parthiban, K., McCafferty, J., Zielonka, S., & Hoet, R. (2024). Advancements in mammalian display technology for therapeutic antibody development and beyond: Current landscape, challenges, and future prospects. *Frontiers in Immunology*, *15*, 1469329. https://doi.org/10.3389/fimmu.2024.1469329

Smith, G. P. (1985). Filamentous Fusion Phage: Novel Expression Vectors That Display Cloned Antigens on the Virion Surface. *Science*, *228*(4705), 1315–1317. https://doi.org/10.1126/science.4001944

Steiner, K., & Schwab, H. (2012). Recent advances in rational approaches for enzyme engineering. *Computational and Structural Biotechnology Journal*, *2*, e201209010. https://doi.org/10.5936/csbj.201209010

Sullivan, B., Walton, A. Z., & Stewart, J. D. (2013). Library construction and evaluation for site saturation mutagenesis. *Enzyme and Microbial Technology*, *53*(1), 70–77. https://doi.org/10.1016/j.enzmictec.2013.02.012

Tan, G., Opitz, L., Schlapbach, R., & Rehrauer, H. (2019). Long fragments achieve lower base quality in Illumina paired-end sequencing. *Scientific Reports*, *9*, 2856. https://doi.org/10.1038/s41598-019-39076-7

Teng, H., Cao, M. D., Hall, M. B., Duarte, T., Wang, S., & Coin, L. J. M. (2018). Chiron: Translating nanopore raw signal directly into nucleotide sequence using deep learning. *GigaScience*, *7*(5), giy037. https://doi.org/10.1093/gigascience/giy037

tqdm developers. (2026). *Tqdm/tqdm* [Python]. https://github.com/tqdm/tqdm (Original work published 2015)

Vereecke, N., Bokma, J., Haesebrouck, F., Nauwynck, H., Boyen, F., Pardon, B., & Theuns, S. (2020). High quality genome assemblies of Mycoplasma bovis using a taxon-specific Bonito basecaller for MinION and Flongle long-read nanopore sequencing. *BMC Bioinformatics*, *21*, 517. https://doi.org/10.1186/s12859-020-03856-0

Wan, Y. K., Hendra, C., Pratanwanich, P. N., & Göke, J. (2022). Beyond sequencing: Machine learning algorithms extract biology hidden in Nanopore signal data. *Trends in Genetics*, *38*(3), 246–257. https://doi.org/10.1016/j.tig.2021.09.001

Wang, B., Jia, P., Gao, S., Zhao, H., Zheng, G., Xu, L., & Ye, K. (2025). Long and Accurate: How HiFi Sequencing is Transforming Genomics. *Genomics, Proteomics & Bioinformatics*, *23*(1), qzaf003. https://doi.org/10.1093/gpbjnl/qzaf003

Wang, H., & Liu, R. (2014). Advantages of mRNA display selections over other selection techniques for investigation of protein–protein interactions. *Expert Review of Proteomics*, *8*(3), 335–346. https://doi.org/10.1586/epr.11.15

Wang, Y., Zhao, Y., Bollas, A., Wang, Y., & Au, K. F. (2021). Nanopore sequencing technology, bioinformatics and applications. *Nature Biotechnology*, *39*(11), 1348–1365. https://doi.org/10.1038/s41587-021-01108-x

Wang, Z., Fang, Y., Liu, Z., Hao, N., Zhang, H. H., Sun, X., Que, J., & Ding, H. (2024). Adapting nanopore sequencing basecalling models for modification detection via incremental learning and anomaly detection. *Nature Communications*, *15*(1), 7148. https://doi.org/10.1038/s41467-024-51639-5

Wang, Z., Tu, M.-J., Song, C., Liu, Z., Wang, K. K., Chen, S., Yu, A.-M., & Ding, H. (2024). The Precise Basecalling of Short-Read Nanopore Sequencing. *bioRxiv*, 2024.09.12.612746. https://doi.org/10.1101/2024.09.12.612746

Waskom, M. (2021). seaborn: Statistical data visualization. *Journal of Open Source Software*, *6*(60), 3021. https://doi.org/10.21105/joss.03021

Wei, H., & Li, X. (2023). Deep mutational scanning: A versatile tool in systematically mapping genotypes to phenotypes. *Frontiers in Genetics*, *14*, 1087267. https://doi.org/10.3389/fgene.2023.1087267

Weile, J., Sun, S., Cote, A. G., Knapp, J., Verby, M., Mellor, J. C., Wu, Y., Pons, C., Wong, C., van Lieshout, N., Yang, F., Tasan, M., Tan, G., Yang, S., Fowler, D. M., Nussbaum, R., Bloom, J. D., Vidal, M., Hill, D. E., … Roth, F. P. (2017). A framework for exhaustively mapping functional missense variants. *Molecular Systems Biology*, *13*(12), MSB177908. https://doi.org/10.15252/msb.20177908

Wenger, A. M., Peluso, P., Rowell, W. J., Chang, P.-C., Hall, R. J., Concepcion, G. T., Ebler, J., Fungtammasan, A., Kolesnikov, A., Olson, N. D., Töpfer, A., Alonge, M., Mahmoud, M., Qian, Y., Chin, C.-S., Phillippy, A. M., Schatz, M. C., Myers, G., DePristo, M. A., … Hunkapiller, M. W. (2019). Accurate circular consensus long-read sequencing improves variant detection and assembly of a human genome. *Nature Biotechnology*, *37*(10), 1155–1162. https://doi.org/10.1038/s41587-019-0217-9

Wick, R. R., Judd, L. M., & Holt, K. E. (2019). Performance of neural network basecalling tools for Oxford Nanopore sequencing. *Genome Biology*, *20*(1), 129. https://doi.org/10.1186/s13059-019-1727-y

Xie, S., Cooley, A., Armendariz, D., Zhou, P., & Hon, G. C. (2018). Frequent sgRNA-barcode recombination in single-cell perturbation assays. *PLoS ONE*, *13*(6), e0198635. https://doi.org/10.1371/journal.pone.0198635

Xu, X., Bhalla, N., Ståhl, P., & Jaldén, J. (2023). Lokatt: A hybrid DNA nanopore basecaller with an explicit duration hidden Markov model and a residual LSTM network. *BMC Bioinformatics*, *24*(1), 461. https://doi.org/10.1186/s12859-023-05580-x

Zade, H. M., Keshavarz, R., Shekarabi, H. S. Z., & Bakhshinejad, B. (2017). Biased selection of propagation-related TUPs from phage display peptide libraries. *Amino Acids*, *49*(8), 1293–1308. https://doi.org/10.1007/s00726-017-2452-z

Zhang, T., Li, H., Jiang, M., Hou, H., Gao, Y., Li, Y., Wang, F., Wang, J., Peng, K., & Liu, Y.-X. (2024). Nanopore sequencing: Flourishing in its teenage years. *Journal of Genetics and Genomics*, *51*(12), 1361–1374. https://doi.org/10.1016/j.jgg.2024.09.007

Zhou, C., Jacobsen, F. W., Cai, L., Chen, Q., & Shen, W. D. (2010). Development of a novel mammalian cell surface antibody display platform. *mAbs*, *2*(5), 508–518. https://doi.org/10.4161/mabs.2.5.12970

Zurek, P. J., Knyphausen, P., Neufeld, K., Pushpanath, A., & Hollfelder, F. (2020). UMI-linked consensus sequencing enables phylogenetic analysis of directed evolution. *Nature Communications*, *11*(1), 6023. https://doi.org/10.1038/s41467-020-19687-9

Appendices

Appendix A. Primer Sequence and Properties

Table A1. Sequences of 1st round PCR primers (KH049, KH050), and 2nd round PCR primers (KH053, and KH054).

ID	Type	Length	Complete Sequence	Binding Sequence
KH049	Forward	50	AAGGTCTCGCAGATGTCAATGGNNNWNNDVTCGCCATCCTGCTGTCATCC	TCGCCATCCTGCTGTCATCC
KH050	Reverse	54	TTGGTCTCGATCTTAGCTTACCNNNSNNNBATCAAAATCACCGGAACCAGAACC	ATCAAAATCACCGGAACCAGAACC
KH053	Forward	52	GTCTCGTGGGCTCGGAGATGTGTATAAGAGACAGTCTCGCAGATGTCAATGG	TCTCGCAGATGTCAATGG
KH054	Reverse	51	TCGTCGGCAGCGTCAGATGTGTATAAGAGACAGTCTCGATCTTAGCTTACC	TCTCGATCTTAGCTTACC

Table A2. Properties of 1st round PCR primers (KH049, KH050), and 2nd round PCR primers (KH053, and KH054).

ID	Amplicon Size	GC%	Tm (ºC)	PCR Optimised Annealing Tm (ºC)	MAX Hairpin Tm (ºC)	MAX Hairpin Free Energy Change (kcal/mol)	MAX SELF-DIMER Free Energy Change (kcal/mol)
KH049	747	60	70	68	36.1	-2.2	-38.13
KH050	747	46	67	68	29.5	-0.34	-26.49
KH053	806	50	62	60	44.6	-3.85	-6.09
KH054	806	44	57	60	38.7	-2.76	-6.76

Appendix B. PCR Conditions

Each PCR round was conducted as four 10 µL reactions, made up to volume with Type I water, giving a combined reaction volume of 40 µL. After cycling, the four reactions were pooled (32 µL total), and a 0.65× bead cleanup was performed. Approximately 16 µL of eluate was recovered. Five extension cycles were used in both PCR rounds to minimise PCR error whilst generating sufficient amplicon for downstream processing.

Table B1. Reagent concentrations for two-round nested PCR.

Material	Concentration
Q5 Reaction Buffer	1X
dNTPs	0.2 mM
Forward Primer	0.5 µM
Reverse Primer	0.5 µM
Template DNA	20 ng/µL
Q5 High-Fidelity DNA Polymerase	0.02 U/µL

Table B2. The thermocycler program for the first-round PCR.

Step	Temp (ºC)	Time (s)
Initial Denaturation	98	20
Denaturation	98	10
Annealing	68	30
Extension	72	30
Final Extension	72	120
Hold	12	∞

Table B3. The thermocycler program for the second-round PCR.

Step	Temp (ºC)	Time (s)
Initial Denaturation	98	180
Denaturation	98	30
Annealing	60	30
Extension	72	30
Final Extension	72	120
Hold	12	∞

Appendix C. Cell Lines & Phage

Table C1. The cell lines and phage used in the web-lab part of this thesis.

Cell Line	Use
Electrocompetent *E. coli* XL1-Blue	Transformation host for selected IgA Fc variant phagemid and bacterial host for helper phage rescue during single-isolate phage production.
NEB Turbo competent *E. coli* glycerol stocks	Recovery and expansion of the enriched phagemid library during colony selection prior to plasmid extraction.
M13K07 helper phage	Rescue of phagemid particles for display, production, precipitation, titration, and ELISA normalisation.

Appendix D. Model Training Details

Table D1. Alpha family training settings and validation summary.

Name	`Alpha`	`AlphaQCL`	`AlphaQCH`
training_data	A_full	A_filtered	A_filtered
min_accuracy_save_ctc	0	0	0.99
training_chunks	27063	21136	10339
validation_chunks	2700	2100	1000
lr	0.00001	0.00001	0.0002
training_loss	0.0150	0.0087	0.0501
validation_loss	0.0095	0.0082	0.0029
validation_mean	99.73	99.78	99.96
validation_median	100	100	100

Table D2. Beta family training settings and validation summary.

Name	`Beta`	`BetaQCL`	`BetaQCH`
training_data	B_full	B_filtered	B_filtered
min_accuracy_save_ctc	0	0	0.99
training_chunks	57394	14911	2939
validation_chunks	5700	1500	300
lr	1.7E-06	0.00001	0.0002
training_loss	0.0503	0.1152	0.2827
validation_loss	0.0492	0.0493	0.0239
validation_mean	98.5	98.54	99.45
validation_median	99.12	98.97	99.46

Table D3. Gamma family training settings and validation summary.

Name	`Gamma`	`GammaQCL`	`GammaQCH`
training_data	AB_full	AB_filtered	AB_filtered
min_accuracy_save_ctc	0	0	0.99
training_chunks	84517	36042	13281
validation_chunks	8450	3600	1300
lr	1.4E-06	0.000004	3.25E-05
training_loss	0.0418	0.0344	0.0524
validation_loss	0.0416	0.0272	0.011
validation_mean	98.75	99.19	99.78
validation_median	99.31	99.52	100

`Bonito` v1.0.1 (https://github.com/nanoporetech/Bonito/releases) was used to fine-tune all models listed in the table above. Representative commands are provided below; parameters enclosed in angle brackets (<>) varied between models.

Basecalling, chunkify, and signal-to-reference alignment:

```
Bonito basecaller --reference fc_reference.mmi --device cuda --seed <> --save-ctc --no-quantize --min-accuracy-save-ctc <> --overlap 600 --chunksize 12000 --batchsize 128 --alignment-threads 32 --mm2-preset map-ont dna_r10.4.1_e8.2_400bps_sup@v5.2.0/ A_full/ > data/basecalls.BAM
```

Fine-tuning using `dna_r10.4.1_e8.2_400bps_sup@v5.2.0` as base model:

```
Bonito train --pretrained dna_r10.4.1_e8.2_400bps_sup@v5.2.0/ --directory data/ --device cuda --lr <> --seed <> --epochs 1 --batch <> --chunks <> --valid-chunks <> --num-workers 32 --restore-optim v1
```

Export model weight at epoch 1 in `Dorado`-compatible format:

```
Bonito export --output Alpha --config v1/config.toml v1
```

Ten seeds were used for '`Bonito basecaller`' and '`Bonito train`' for the final synthesis figures. For all other figures, 1608668 was used as a seed.

```
1608668, 800022071, 1364179249, 1418059007, 2068136963, 2072361201, 2575967173, 2891725340, 2918267749, 3763609086
```

Appendix E. Source Code and Software Tools

The complete thesis codebase is hosted on GitHub: https://github.com/Amekn/COMPX593-Thesis.git. The repository is organised into `src/c++`, `src/bash`, `src/python`, and `src/ipynb` directories. Compilation instructions for Linux, Windows, and macOS are provided in the repository README.md. Source datasets used for training and benchmarking are not included in the repository. The fine-tuned DNA model is provided in the `models/` directory, while the `weight_1.tar` file required for further fine-tuning is available on request. Summary tables are provided in the `tables/` directory. Table E1 below lists the custom programs and workflows developed during this thesis. Table E2 below lists the third-party tools/libraries used during this thesis.

Table E1. Custom programs and workflows developed for this thesis

Program/Workflow	Type	Description
`DualSiteDMSFilter`	C++ executable	Performs DMS-aware filtering and correction from sorted primary alignment `BAM` input, writes polished `FASTQ` records, and exports DMS variant-key count `TSV` tables.
`Benchmarker`	C++ executable	Compares dual-UMI nanopore `FASTQ` reads against UMI-matched short-read ground truth and reports nucleotide-level precision, recall, F1-score, and error rate estimate.
`VariantConcordance`	C++ executable	Computes population-level concordance metrics between nanopore and ground-truth variant-key/count tables across minimum-count thresholds.
`VariantKeyOverlap`	C++ executable	Compares sets of DMS variant keys across multiple variant-count tables and reports, intersecting, unique, and pairwise-overlap statistics.
`UmiExtractor`	C++ executable	Extracts dual-UMIs from ambiguous primer positions, appends canonical UMI tags to `FASTQ` headers, and trims reads to the Fc reference.
`UmiCloner`	C++ executable	Transfers dual-UMI-annotated `FASTQ` headers onto matching reads from a second basecall of the same `POD5` input.
`UmiSpliter`	C++ executable	Splits dual-UMI `FASTQ` records into separate single-UMI `FASTQ` outputs and reports UMI-complexity summaries.
`UmiFilter`	C++ executable	Performs single-pass dual-UMI deduplication by retaining the first `FASTQ` record for each distinct UMI key.
`UmiOverlap`	C++ executable	Calculates overlap between unique dual-UMI sets from two UMI-annotated `FASTQ` files.
`SingleUmiOverlap`	C++ executable	Calculates read-level and unique-key overlap for single-UMI `FASTQ` files after dual-UMI splitting.
`PrimerTrimmer`	C++ executable	Trims reads to the span encapsulated by forward and reverse primers and normalises reverse-orientation reads to the forward frame.

`MQAAR`	C++ executable	Calculates dataset Q-score from sequencing-summary tables and optionally converts it to estimated basecalling accuracy.
`ParseStats`	C++ executable	Parses the summary number section of `samtools` stats output into a compact metrics table for alignment reports.
`DMSPolishing.sh`	Bash workflow	Runs the end-to-end per-model basecalling, length filtering, alignment, DMS filtering and single-read correction, realignment, and metric-logging workflow.
`CalcStats.sh`	Bash workflow	Wraps `samtools` stats and `CIGAR` parsing to calculate mapped-read, mismatch, insertion, deletion, error rate, and overall error rate from `BAM` files.
`GenerateDataCSV.sh`	Bash workflow	Generates the per-dataset quality-control table by applying basecalling, length filtering, quality filtering, alignment filtering, and summary metric extraction.
`GenerateTestCSV.sh`	Bash workflow	Generates the per-model benchmarking table by running basecalling or `FASTQ` passthrough (i.e., for `UDP0057.fq`), length filtering, alignment filtering, DMS-aware filtering and summaries, and row-level metrics aggregation.
`GenerateCorrelationCSV.sh`	Bash workflow	Generates the variant concordance table by comparing per-model variant-key tables against the short-read ground truth at specified thresholds.
`POD5Splitter.py`	Python utility	Splits large `POD5` files into near-equal read-count shards for parallel preprocessing.
`POD5Merger.py`	Python utility	Merges multiple `POD5` files into a single output while interleaving reads in a round-robin fashion to ensure homogeneous data distribution (i.e., for training).
`MergeNumPy.py`	Python utility	Merges `Bonito` training array triplets using memory-mapped streaming to avoid high RAM usage during dataset construction. (i.e., >128GB used without streaming, reduced to ~1-2 GB with streaming).
`PlotSignal.py`	Python utility	Renders an individual raw nanopore signal trace from a `POD5` read as a high-quality SVG figure (Figure 1).
`weight_stats.py`	Python Utility	Analyses `.tensor` files of fine-tuned models against the pretrained `sup` baseline; the output is six `CSV` tables containing aggregated summaries used by `plot.ipynb` for plotting.
`plot.ipynb`	Figure Plotting	A `Jupyter Notebook` was used to plot all the analysis figures used in this thesis. The main sources of data include disk files and `CSV` tables generated by utilities and workflows.

Table E2. Third-party software and libraries used throughout this thesis.

Program/Workflow	Version	Type	Description
`Dorado`	1.3.0	Basecaller	Oxford Nanopore production basecaller used to basecall `POD5` files with `sup` baseline and fine-tuned model exports (Oxford Nanopore Technologies, 2022/2026b).

`Bonito`	1.0.1	Basecaller	Oxford Nanopore research basecaller used for `NumPy` ndarray generation, model fine-tuning, and model export (Oxford Nanopore Technologies, 2019/2026).
`minimap2`	2.28	Aligner	Aligned nanopore reads to the Fc amplicon with the `map-ont` preset and aligned short-read ground truth with the `sr` preset (Li, 2018).
`samtools`	1.19.2	Alignment Toolkit	Converted `BAM` to `FASTQ`, filtered primary alignments, sorted `BAM` files, and generated alignment summary statistics (Danecek et al., 2021).
`fastplong`	0.2.2	`FASTQ` QC/Filtering Tool	Applied length and quality filters to long-read `FASTQ` files and generated per-stage read-quality reports (OpenGene - Open Source Genomics Toolbox, 2024/2026).
`Jupyter Notebook`	7.4	Interactive Analysis	Hosted the plotting notebook used to generate thesis figures, panels, and summary tables (Beg et al., 2021).
`NumPy`	2.3.2	Python Numerical Library	Supported array operations, `Bonito` training-array inspection, memory-mapped merging, and numerical analysis (Harris et al., 2020).
`pandas`	3.0.1	Python Data Analysis	Loaded, transformed, joined, and summarised `CSV`/`TSV` tables for benchmarking and plotting (McKinney, 2011).
`matplotlib`	3.10.8	Python Plotting Library	Rendered thesis figures and SVG panels, including new raw signal traces and a summary (Hunter, 2007).
`seaborn`	0.13.2	Python Statistical Plotting Library	Provided statistical plotting functions and styles for distribution and comparison figures (Waskom, 2021).
`POD5`	0.3.28	Python `POD5` API	Reads, splits, merges, and inspects Oxford Nanopore `POD5` signal files in custom preprocessing utilities (Oxford Nanopore Technologies, 2022/2026a).
`tqdm`	4.67.1	Python Progress Reporting Library	Displayed progress bars during long-running Python preprocessing (`tqdm` developers, 2015/2026).
`CMake`	4.0.3	Build system	Configured and built the custom C++ command-line tools used in the computational pipeline (Kitware, Inc., 2010/2026).

Appendix F. Supplementary Figures

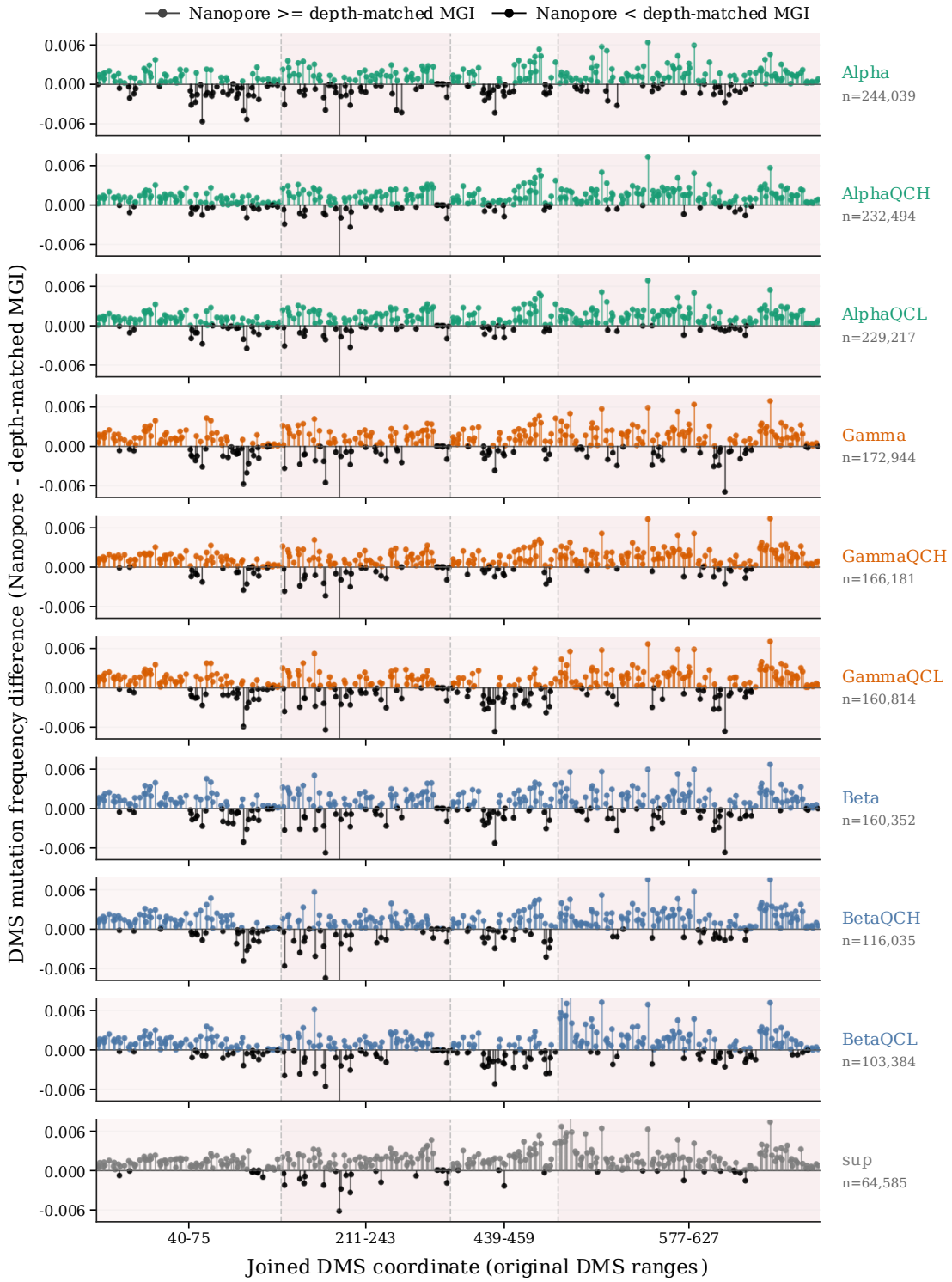

Figure F1. Depth-matched DMS mutation-frequency differences across all models. *Each row corresponds to one nanopore output; the short-read ground-truth was downsampled to that model's non-WT depth shown at right (n), and the plotted short-read frequency is the median over 256 resamples. The x-axis is the joined DMS coordinate, the y-axis is nanopore minus depth-matched MGI frequency, coloured points indicate events with higher nanopore frequency than the depth-matched short-read estimate, and black points indicate events with lower frequency. Frequencies were calculated for shared non-WT mutation events with a minimum count threshold of 1. Full mutational event summary is available in* `mutation.csv` *on GitHub (Appendix E).*

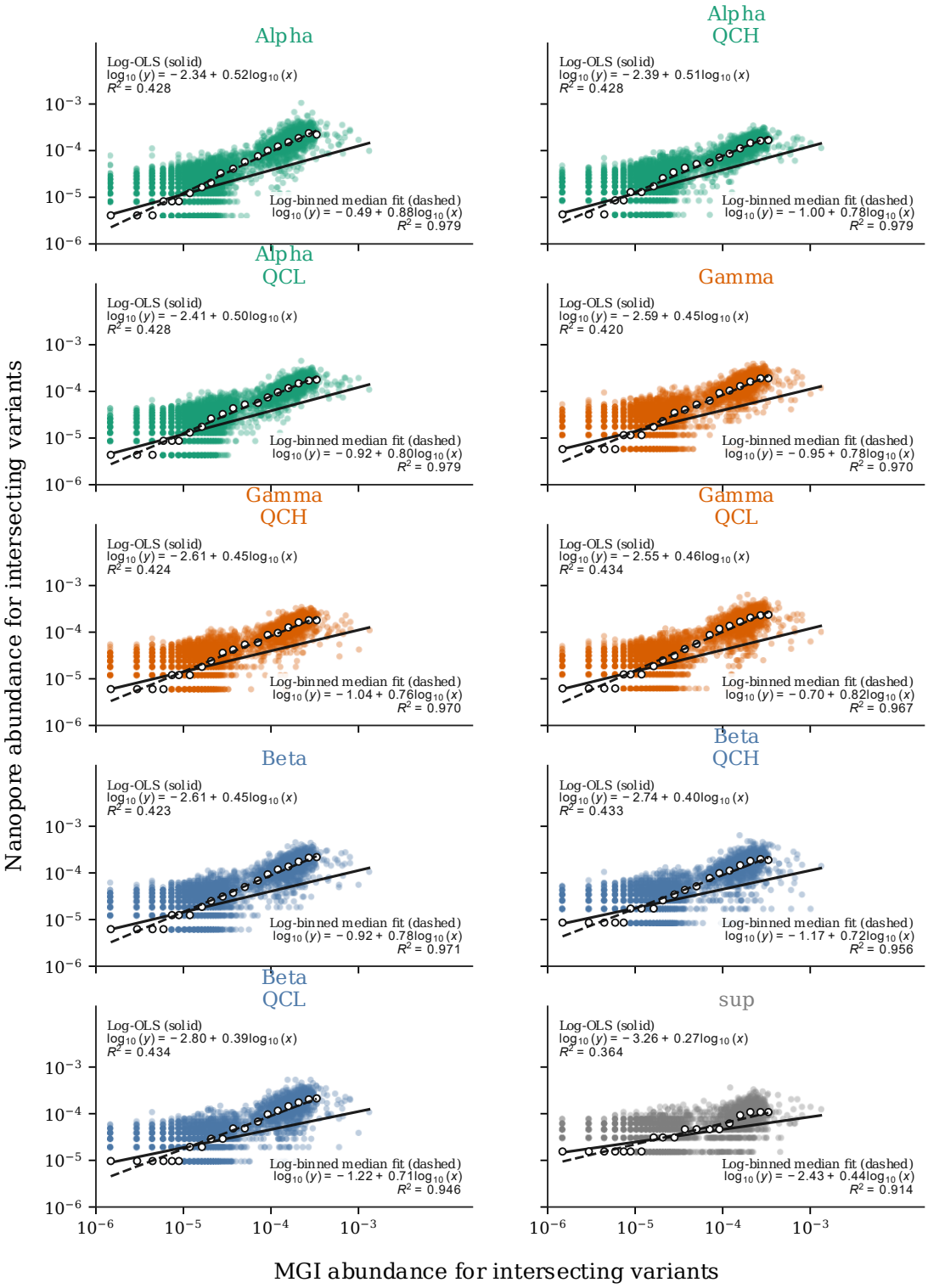

Figure F2. Abundance concordance for intersecting variants across all models. *Log-log comparison of nanopore and short-read abundance for intersecting variants across all nanopore outputs. The ten panels show* `Alpha`, `AlphaQCH`, `AlphaQCL`, `Gamma`, `GammaQCH`, `GammaQCL`, `Beta`, `BetaQCH`, `BetaQCL`, *and* `sup`*; each point is one intersecting non-WT variant at a minimum count threshold of 1. Each panel displays two fits: a log-space OLS regression on all individual data points (solid line), which captures the overall abundance relationship including low-abundance scatter, and a regression through log-binned medians (dashed line), which isolates the central abundance trend. Full concordance summary is available in* `correlation.csv` *on GitHub (Appendix E).*

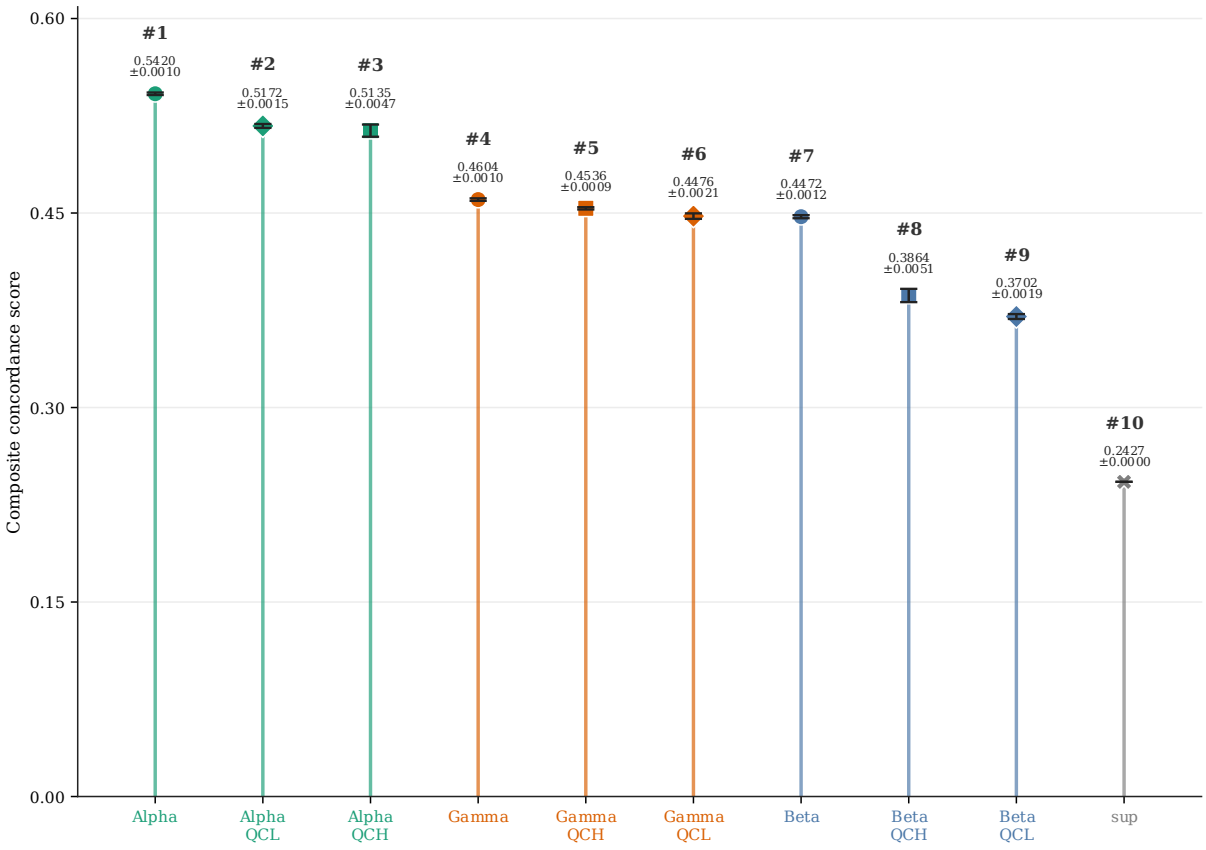

Figure F3. Mean composite concordance score (CCS) ranking across all models. *Lollipop plot for all model outputs. Ten different random seeds were used to train each model 10 times to mitigate training non-determinism, and the ranks are computed as the mean composite concordance score (CCS) across seeded fine-tuning runs for each model. Each model displays the mean CCS under its rank, along with a 95% confidence interval. CCS is the weighted geometric mean of the normalised AUC values for weighted Jaccard similarity, Jensen-Shannon similarity, exact overlap mass, and variant F1 across thresholds 1 to 10. Normalised AUC Scores were computed from the threshold-robustness curves in Figure 25. For CCS definition, see Equation (2.25) in Section 2.2.6.*

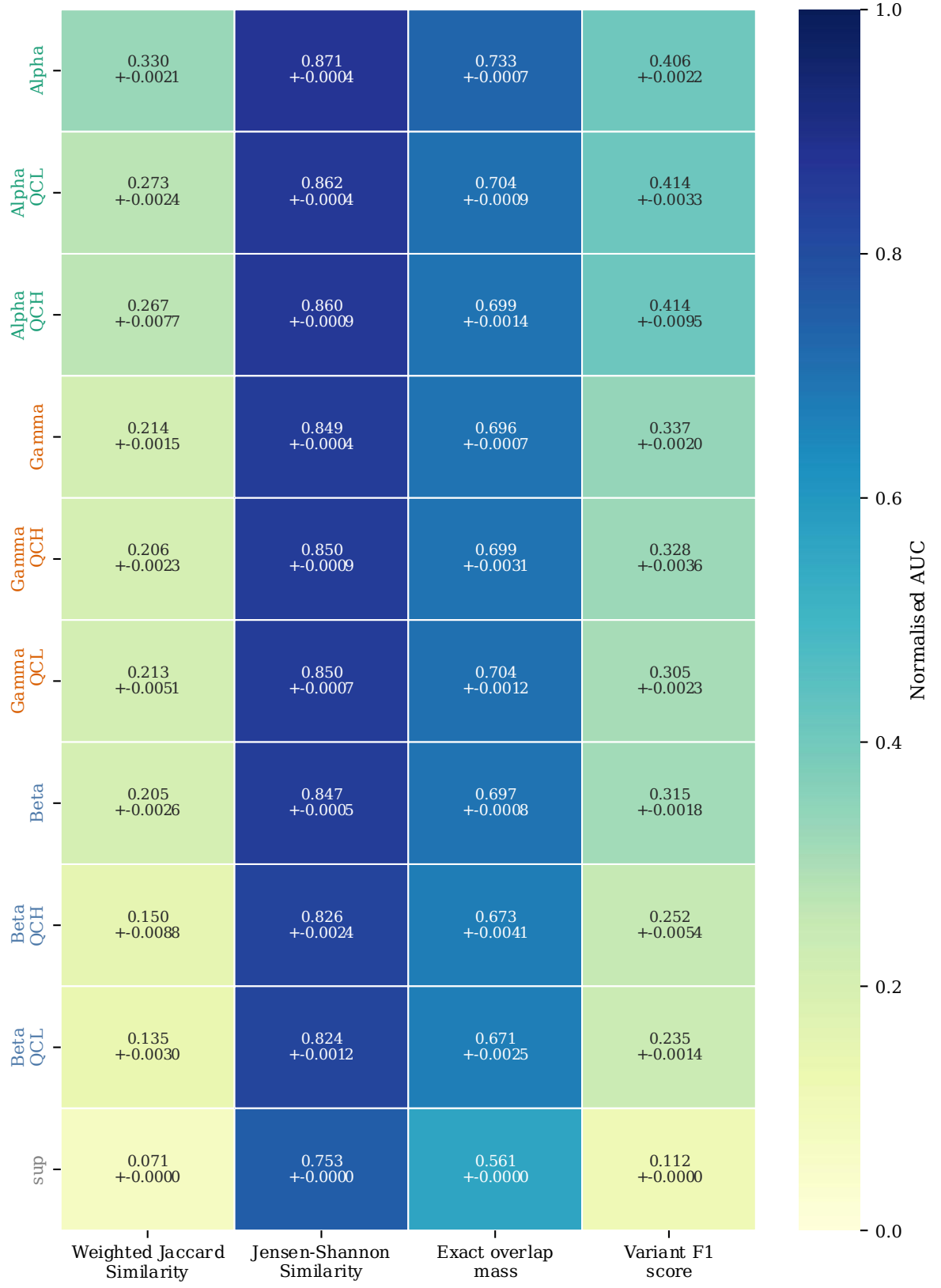

Figure F4. Mean Normalised AUC scores heatmap across all models. *Heatmap of the mean normalised AUC scores underlying Figure F3. Each cell shows the mean normalised AUC [0,1] for one of four concordance metrics used to calculate mean CCS across 10 seeded model training runs, with the standard deviation shown below. AUC values were computed by trapezoidal integration of each metric across minimum count thresholds 1 to 10 and then normalised using the theoretical maximum AUC of 9. For all definitions of concordance metrics, see Section 2.2.6.*

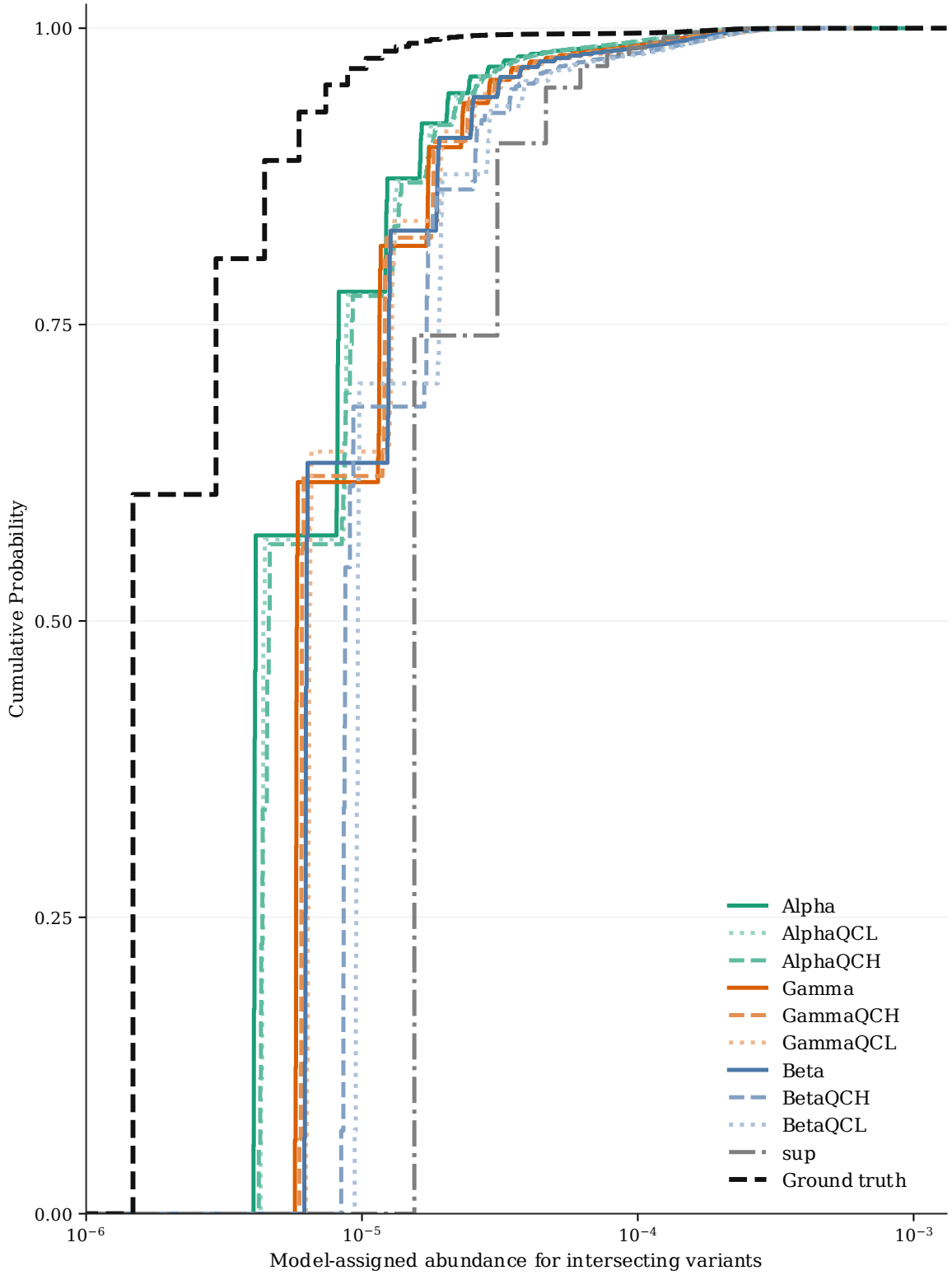

Figure F5. ECDF of model-assigned abundance for intersecting variants across all models. *For each abundance value on the x-axis, the y-axis shows the cumulative proportion of intersecting variants with model-assigned abundance less than or equal to that value. Curves are coloured by family; QCH and QCL variants are distinguished by line style; the grey curve shows the* `sup` *baseline; and the dashed black curve shows the full short-read non-WT distribution. Note that the ECDF presented here is derived by aggregating model-assigned abundance across all 10 seeded fine-tuning runs for each model. Slight differences in model performance across different seeded runs caused the curve to be uneven. On this ECDF, curves that rise further left indicate that a larger fraction of variants have low assigned abundance. In contrast, rightward-shifted curves indicate that the intersecting variants tend to have higher assigned abundances. For ECDF methodology, see Section 2.2.6.*

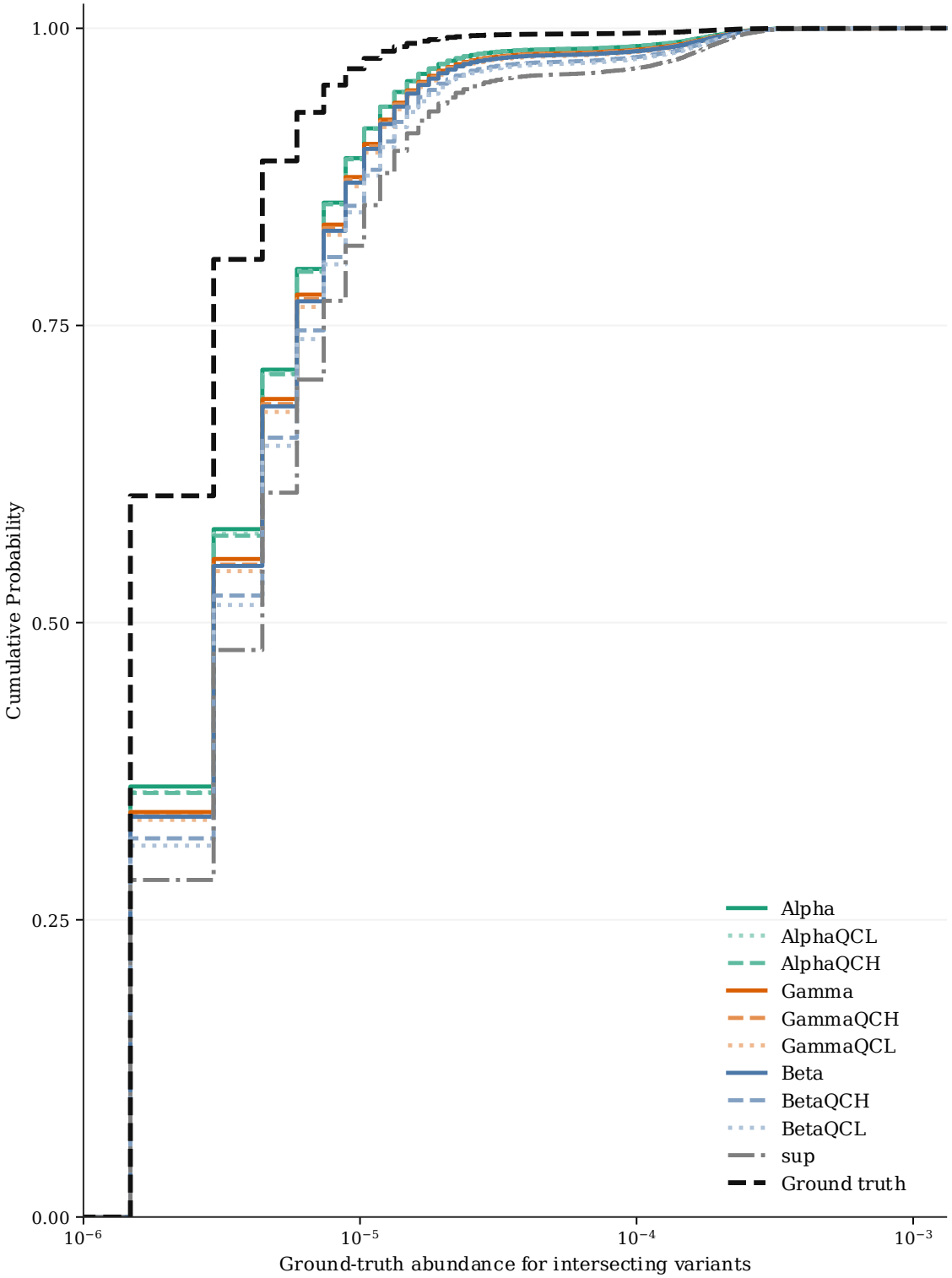

Figure F6. ECDF of ground-truth abundance for intersecting variants across all models. *For each abundance value on the x-axis, the y-axis shows the cumulative proportion of intersecting variants with ground-truth abundance less than or equal to that value. Curves are coloured by family; QCH and QCL variants are distinguished by line style; the grey curve shows the* `sup` *baseline; and the dashed black curve shows the full short-read non-WT distribution. Note that the ECDF shown here is created by combining ground-truth abundance rather than model-assigned abundance across all 10 seeded fine-tuning runs for each model. This produces a smooth curve, unlike the pooled model-assigned ECDF in Figure F5. Curves shifted to the right relative to the dashed reference indicate that the variants recovered by that model are enriched for higher-abundance ground-truth variants, with fewer low-abundance variants from the true long tail. For ECDF methodology, see Section 2.2.6.*