High-accuracy SNV calling for bacterial isolates using deep learning with AccuSNV

Herui Liao, Arolyn Conwill, Ian Light-Maka, et al.

Genome Res. published online July 6, 2026
Access the most recent version at doi:10.1101/gr.281341.125

High-accuracy SNV calling for bacterial isolates using deep learning with AccuSNV

Herui Liao[1,2], Arolyn Conwill[1,2], Ian Light-Maka[5,6], Martin Fenk[5,7], Alyssa H. Mitchell[1,2], Evan B. Qu[1,2], Paul Torrillo[1,2], Jacob S. Baker[1,2], Lilly R. Bartsch[5], Felix M. Key[5], Tami D. Lieberman[1,2,3,4,*]

- [1] Institute for Medical Engineering and Sciences, Massachusetts Institute of Technology; Cambridge, MA 02139, USA
- [2] Department of Civil and Environmental Engineering, Massachusetts Institute of Technology; Cambridge, MA 02139, USA
- [3] Broad Institute of MIT and Harvard; Cambridge, MA 02139, USA
- [4] Ragon Institute of MGH, MIT, and Harvard; Cambridge, MA 02139, USA
- [5] Max Planck Institute for Infection Biology; Berlin 10117, Germany
- [6] Charité-Universitätsmedizin Berlin, Berlin 10117, Germany
- [7] Humboldt-Universität zu Berlin, Faculty of Life Sciences, Berlin, Germany
- [*] Corresponding author. Email: tami@mit.edu

Abstract

Accurate detection of mutations within bacterial species is critical for fundamental studies of microbial evolution, reconstructing transmission events, and identifying antimicrobial resistance mutations. While many tools have been developed to identify single nucleotide variants (SNVs) from whole-genome sequencing, they often suffer from high false positive rates due to the complexity of bacterial genomes and the need for different filtering cutoffs across sample types and sequencing depths. As datasets increase in size, the manual filtering required for high accuracy presents a significant obstacle. Here, we present AccuSNV, a novel deep learning-based tool for high-precision and automated bacterial SNV calling. Unlike traditional methods that process one sample at a time, AccuSNV leverages a convolutional neural network (CNN) that integrates alignment information across multiple samples, enhancing precision through learned across-sample patterns. We evaluate AccuSNV against seven popular SNV calling tools using simulated data from six bacterial species with varied sequencing depths, numbers of isolates, mutations, and divergence levels. To further validate its real-world utility, we test AccuSNV on multiple curated bacterial datasets containing reported SNVs. In both simulated and real-world scenarios, AccuSNV consistently achieves the best performance. Moreover, AccuSNV provides comprehensive user-friendly downstream analysis modules and outputs, including mutation annotation information, phylogenetic inference, d_N/d_S calculations, and optional manual

filtering. Together with the automated deep learning–based calling, these features make AccuSNV broadly accessible to users with different levels of computational expertise.

Introduction

Single nucleotide variants (SNVs), single-base substitutions in a genome, represent the most common form of genetic variation. In bacteria, accurate identification of SNVs is a critical step in connecting genetic variation to phenotype (Lopatkin et al. 2021; Schrader et al. 2021) and for reconstructing bacterial evolution (Barrick et al. 2009). In particular, SNV mutations form the basis of most phylogenetic inference approaches, making their accurate detection essential for epidemiology (Halachev et al. 2014). Even small errors in SNV detection can have outsized effects on downstream analyses. This is especially true for many bacterial species, which accumulate SNV mutations at a rate of only 1 - 10 SNVs per genome per year (Key et al. 2023). This low rate of mutation accumulation makes evolutionary inference methods sensitive to false positives, thus requiring high-precision SNV calling.

SNV calling across bacterial genomes of the same species remains a challenging task. One major source of SNV calling errors arises from alignment inaccuracies, which are exacerbated by the high diversity within bacterial species. Unlike the relatively stable and homogeneous human genome, bacteria within the same species can vary up to 5% in nucleotide identity and share only 40% of their gene content. These characteristics increase the possibility of misalignments and false variant calls (Zojer et al. 2017), particularly in low-coverage datasets or when the sample strain diverges from the reference genome (Bush et al. 2020). In addition, cross-contamination during high-throughput sequencing preparation can lead to false positive and false negative SNV calls, posing challenges for standard mapping-based SNV filtering methods (Goig et al. 2020). These errors can significantly reduce the precision of bacterial SNV calling.

Several SNV calling tools have been developed and applied to identify SNVs across bacterial isolates (Li et al. 2009; Koboldt et al. 2009; McKenna et al. 2010; Garrison and Marth 2012; Barrick et al. 2014; Seemann 2015; Yoshimura et al. 2019). Most of these tools combine probabilistic models (likelihoods or statistics) for each genotype in each sample based on alignment metrics such as base quality, read depth, and strand bias, and then apply filters or thresholds to make allele calls in per-sample basis. Some of these tools, such as GATK and FreeBayes, were originally designed for eukaryotic organisms like humans (McKenna et al. 2010) and are not specifically optimized for bacterial genomes, while others like breseq and Snippy are tailored for microbial analysis. These methods are widely used due to their simplicity and informative visualizations that support additional filtering as

needed. A second category integrate both mapping and assembly information. For example, BactSNP, performs *de novo* assembly of input reads and aligns the assembled contigs to the reference genome to identify SNVs. Through this approach, BactSNP aims to reduce false positives caused by read misalignment in complex genomic regions. However, the accuracy of this approach depends heavily on the quality of the assembly and the similarity between sample and reference genomes, and it remains computationally intensive for large datasets (Supplementary Table S1).

Despite their respective strengths, these existing methods share common limitations. First, most tools typically call variants by considering each isolate individually, which obscures across-sample patterns that can reveal systematic alignment errors at specific genomic positions. False SNVs emerging from alignment errors exhibit distinct alignment patterns across samples (Fig. 1A-1D), and failing to consider these patterns while focusing solely on single-sample evidence can lead to false-positive calls. Another common issue is reference bias: when all samples share a nucleotide that differs from the reference, methods that do not consider raw read data may erroneously call the site as polymorphic (e.g. where the reference nucleotide stems from low-coverage assembly errors), resulting in false positives. Lastly, these tools rely on manually defined thresholds or assumptions that may not generalize well, and often require extensive manual filtering to reduce false positives.

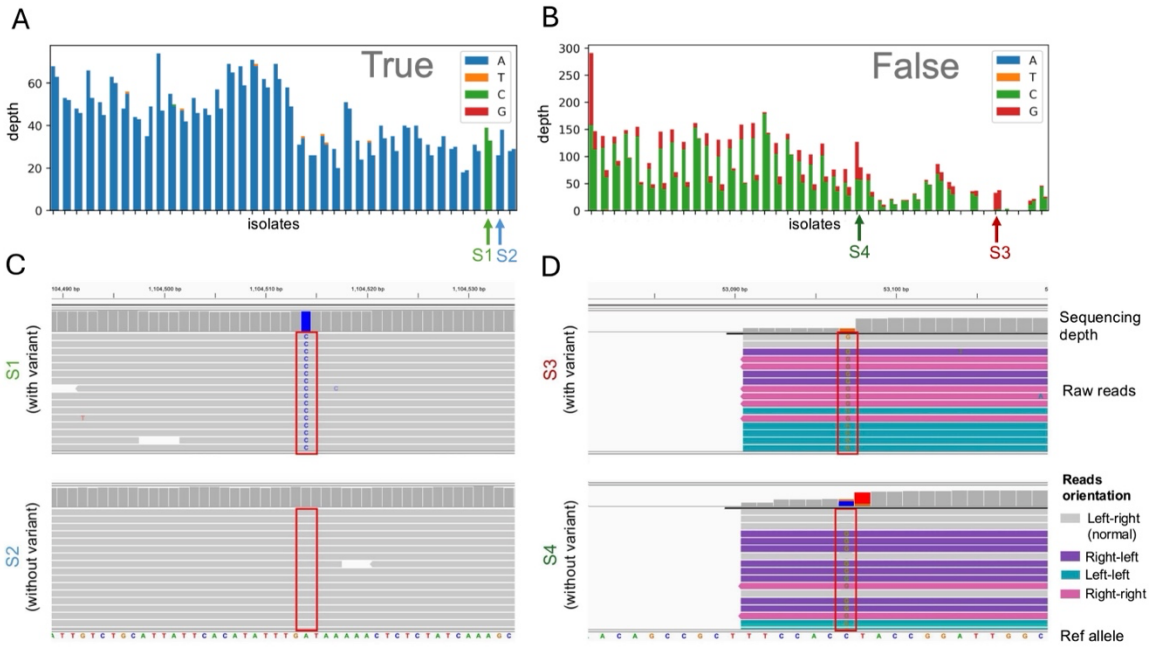

Figure 1. ***False SNVs arising from alignment errors are detectable from across-sample patterns*** *(A-B): Bar charts show sequencing depth across isolates for an example true SNV (A) and a false SNV (B) from real-world clinical data, with each pair of bars representing reads aligned to the forward and reverse strands for a*

single isolate, colored by nucleotide supported. Arrows indicate two randomly selected isolates (S1, S2 for the true SNV; S3, S4 for the false SNV) that are shown in (C) and (D). (C-D): Integrative Genomics Viewer (IGV) views of the selected representative isolates S1, S2 (C) and S3, S4 (D) carrying either the major or alternative allele for the true and false SNV, respectively. The candidate SNV position is highlighted by red boxes, with the reference allele shown at the very bottom of the figure. In panel D, the false positive arises from systematic mis-mapping of reads to this genomic position in both samples. The presence of a nearby structural variant results in the preferential recruitment of erroneous reads, leading to a variant call in the absence of sufficient reference allele support. Reads are colored by pair orientation. Only left–right orientation (with the upstream read on the forward strand and downstream read on the reverse strand) is considered normal; abnormal orientations often indicate alignment errors or structural variants.

These constraints limit precision and scalability, particularly in studies involving low-depth sequencing, high strain diversity, or large-scale comparative analyses. Recent advances in deep learning (Min et al. 2016; Hernández Medina et al. 2022; Liao et al. 2023) have opened new possibilities for addressing these limitations. Unlike traditional rule-based or statistical approaches, deep learning models can automatically learn complex patterns from raw input data (Ahmed et al. 2023). In the context of bacterial SNV calling, this allows models to move beyond fixed thresholds and handcrafted features, instead identifying informative signals directly from sequencing alignments. Importantly, deep learning architectures, such as convolutional neural networks (CNNs), are well-suited to capture both local alignment features and broader patterns across samples, offering a powerful framework for distinguishing true variants from false positives in diverse and noisy datasets.

In this work, we have developed AccuSNV, a deep learning-based SNV calling tool optimized for bacterial whole-genome sequencing (WGS) data. AccuSNV leverages a unique data structure that summarizes across-sample alignment information at each candidate SNV and a CNN model that derives informative patterns from this multisample read alignment data. By capturing both within-sample signals and across-sample patterns, it offers improved robustness to low-depth coverage, reference and genome divergence, and diverse real-world datasets. AccuSNV takes a reference genome and raw short-read WGS data from multiple ($>= 3$) isolates as input, and outputs high-confidence SNVs alongside predicted probabilities, variant annotations, per-sample allele calls, and informative visualizations. Through comprehensive benchmarking across both simulated and real-world datasets, we show that AccuSNV achieves consistently superior precision and generalization across a wide range of bacterial species and sequencing conditions.

Results

AccuSNV employs a CNN model trained on real-world bacterial isolate data

In this study, we curated real-world bacterial isolate WGS data (short reads) with manually labeled SNVs from prior studies conducted by our lab (Zhao et al. 2019; Conwill et al. 2022; Baker et al. 2025) to serve as training and validation datasets of AccuSNV. These datasets span 3 different species, with significantly different genome sizes and GC contents (Supplementary Table S2). We chose this data because: (1). real-world data contains authentic sequencing noise, technical artifacts, and biological complexity that are difficult to simulate, providing a more robust foundation for model training compared to simulated data; (2). validated SNV datasets are rare, and most publicly available ones report only true positives without false SNVs, making it impossible for models to learn the features of false calls. Our lab's datasets overcome this limitation through manual labeling of both true and false SNVs, with standardized formats and stringent filtering (Supplementary Table S3); (3). for these datasets, pre-extracted alignment-derived features were already available, which greatly reduced the time and computational cost of model training since raw sequencing data did not need to be reprocessed. In addition, these curated features supported the quick generation of bar charts (Fig. 1) that visually distinguish high-quality from low-quality SNVs, providing an additional manual validation beyond automated filters (see Methods) and thereby enhancing label reliability; (4). the diversity of bacterial species and experimental contexts provides a foundation for generalization.Given these datasets, each candidate SNV site was encoded as a feature vector summarizing read-level signals and cross-sample alignment patterns (Fig. 2A), and a CNN was trained to classify them as true or false variants (Fig. 2B).

Critically, alignment error patterns, such as depth imbalances and mixed allelic signals in false positives, remain consistent across the three species used for training because false-positives are primarily driven by common failure modes of short-read sequencing and alignment algorithms (Supplementary Fig. S2). This observation led us to hypothesize that a model could be built which effectively generalizes by learning these shared patterns.

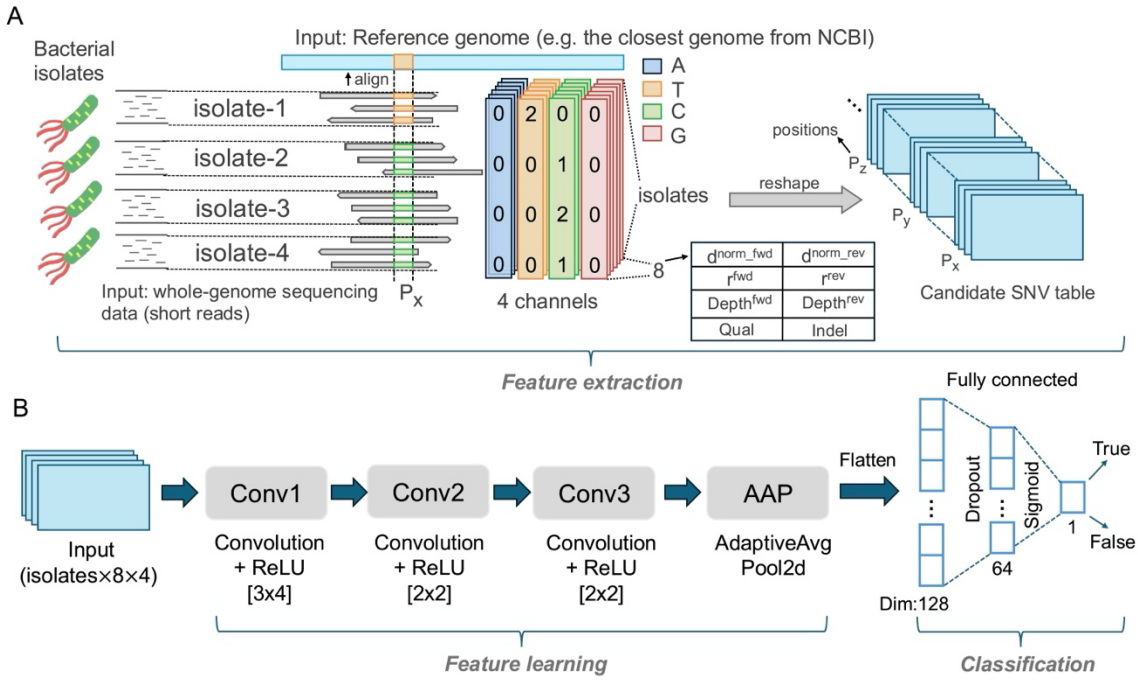

Figure 2. A deep learning framework that leverages across-sample comparisons for bacterial SNV calling. (A). Feature extraction pipeline: short reads from bacterial isolates are aligned to a reference genome, features at candidate SNV positions are extracted across four channels (A, T, C, G). For each channel, there are 8 features (see Supplementary Fig. S1 for full details): $Depth^{fwd}$, $Depth^{rev}$*: raw forward- and reverse-strand depths;* d^{norm_fwd}, d^{norm_rev}*: forward- and reverse-strand normalized depths at each locus and each channel;* r^{fwd}, r^{rev}*: forward- and reverse-strand relative depths showing the fraction of total coverage contributed by each isolate at each position and each channel;* $Qual$*: Consensus quality (FQ) scores produced by SAMtools;* $Indel$*: the number of reads supporting insertions and deletions at each position in that channel. These features are reshaped into a 4D (positions × isolates × features × channels) tensor and stored in a candidate SNV table for neural network input. (B). Model architecture: the input tensor is processed through three convolutional layers with ReLU activation and varying kernel sizes (shown in brackets), followed by adaptive average pooling. The output is then flattened and passed through fully connected layers with dropout for binary classification. The model finally outputs prediction probabilities, with SNVs classified as True if probabilities* > 0.5*, otherwise False.*

The CNN model outperforms classical machine learning methods

To build AccuSNV, we first explored several classical machine learning models using the labeled SNV dataset derived from read alignments of real bacterial sequencing data from the Lieberman Lab. Input features (Supplementary Fig. S1) were constructed from read alignment patterns at candidate sites along with across-sample alignment statistics, and these were used to train XGBoost, random

forest, support vector machines (SVM), and logistic regression models. We compared their performance against the CNN model with two and three convolutional layers to determine the optimal model complexity for capturing local sequence context and across-sample alignment patterns. To minimize bias, we randomly split the dataset into training and validation sets five times, retrained all models on each split, and reported average performance metrics on the validation data (Supplementary Fig. S3).

The CNN model consistently achieved the highest scores across all evaluation metrics on the validation dataset (Supplementary Fig. S3), outperforming all classical models ($p<0.05$, Wilcoxon signed-rank test). Notably, the three-layer CNN also outperformed a shallower two-layer CNN architecture, demonstrating the importance of sufficient model depth for learning complex alignment patterns.

To understand which features contribute most to the CNN's performance, we conducted an ablation study by systematically removing different feature groups, retraining the model, and evaluating performance on the validation set (Supplementary Fig. S4). Consensus quality scores (Qual), derived from SAMtools FQ score, showed the largest impact when removed (0.66% F1 drop), consistent with their established importance in variant calling. However, removing any other single feature group resulted in modest performance decreases (<0.4% F1 drop), indicating that the CNN's robustness stems from integrating multiple complementary features rather than relying on any single dominant signal.

The superior performance of CNN likely stems from fundamental architectural differences between CNN and classical machine learning approaches in handling genomic alignment data. Specifically, CNN preserves the relationship between alleles and their features across samples, while classical models require flattening to 2D vectors that lose crucial across-sample patterns. Additionally, CNNs can automatically learn complex patterns from the multidimensional alignment data and detect recurring motifs (e.g. the same minor allele across many samples, Fig. 1B) regardless of their position in the tensor, whereas classical models rely on hand-crafted features and cannot capture these spatial relationships. These results motivated our choice of CNN with 3 convolutional layers as the core of the AccuSNV framework.

Overview of benchmark experiments

To evaluate the performance of AccuSNV, we benchmarked it against seven widely used SNV calling tools (GATK v.4.5.0.0, FreeBayes v.1.3.6, SAMtools v.1.20, BactSNP v.1.1.0, VarScan v.2.4.6, breseq v.0.39.0, and Snippy v.4.6.0) using both simulated and real-world datasets. The simulated data consisted of Illumina-like reads generated from six representative bacterial species across a range of sequencing depths (10× to 50×), divergence levels, and contamination scenarios, totaling thousands of samples. All tools were run using default or recommended

parameters (see Supplementary Section 1). Across these diverse conditions, AccuSNV achieved consistently high accuracy, demonstrating robustness to challenges including low coverage, reference and genome divergence, and complex sequencing and alignment artifacts.

AccuSNV achieves high precision across sequencing depths

Sequencing depth serves as a critical factor influencing the accuracy of SNV calling, where low coverage can lead to false positives or missed true SNVs (Fuchs et al. 2024). In this experiment, we simulated datasets of 10 isolates with sequencing depths of 10×, 20×, 30×, 40×, and 50× to compare the performance of different tools across varying sequencing depths. First, we selected six representative bacterial species (*C. acnes*, *C. diffile*, *E. coli*, *K. pneumoniae*, *S. aureus*, *S. pneumoniae*) and retrieved representative reference genomes (see Methods) from NCBI (Sayers et al. 2024). To mimic real-world scenarios, we simulated datasets of closely related organisms and a more distant reference genome. As illustrated in Supplementary Fig. S5A, we introduced 1% SNVs and 500 indels into each reference genome with SimuG (Yue and Liti 2019) to create a root genome, which serves as the common ancestral sequence from which all simulated isolates will subsequently diverge. From this, we generated 10 mutant genomes (with about 70 mutant positions introduced per genome on average) per species with a phylogenetic structure intended to mimic realistic evolutionary relationships and simulated Illumina reads at 10× to 50× coverage using ART (Huang et al. 2011), resulting in 300 samples (10 isolates × 6 species × 5 depths). All reads were aligned back to the corresponding reference genome, SNV calling was performed using each tool, and precision, recall, and average F1 scores of each tool were calculated (Fig. 3).

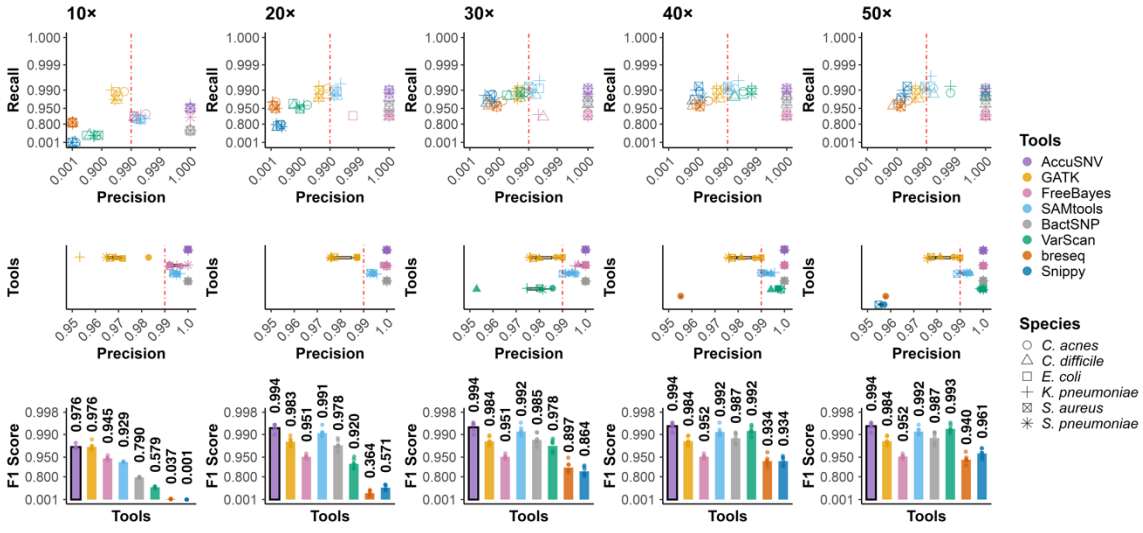

Figure 3. ***AccuSNV achieves high precision across simulated datasets with different sequencing depths.*** *Precision, recall, and average F1 scores of AccuSNV*

and seven other variant-calling tools were evaluated across simulated bacterial datasets at five sequencing depths (10×, 20×, 30×, 40×, and 50×). Axes use a transformed scale ($-log_{10}(1.0001 - x)$) to enhance visualization in the higher range (0.9-1). Shapes represent species and colors represent tools. The vertical line at 0.99 precision highlights how more tools achieve high performance at greater sequencing depths. Top panels show precision–recall scatter plots for all tools and species at each depth. Middle panels provide a zoomed-in view of precision distributions across tools and species. Bottom panels summarize the average F1 score for each tool. The bar corresponding to AccuSNV is highlighted by a black outline.

The performance of all tools improves with increasing sequencing depth, highlighting the importance of high-depth sequencing data for existing tools to accurately identify bacterial SNVs. In addition, AccuSNV achieves the highest average F1 score across all tested datasets (Fig. 3). In particular, AccuSNV achieves F1 scores of 97.6% and 99.4% at coverages as low as 10× and 20×, respectively, while maintaining 100% precision across all experiments. Among other tools, BactSNP also achieves perfect precision but with lower recall compared to AccuSNV. At a sequencing depth of 10×, GATK has higher recall than AccuSNV, but its average precision is 96.8%, compared to 100% for AccuSNV. SAMtools and VarScan perform competitively at higher depths but underperform at 10×, with F1 scores dropping to 92.9% and 57.9%, respectively. The remaining three tools, FreeBayes, breseq, and Snippy show poorer performance compared to other tested tools across all datasets. The performance of both breseq and Snippy drop markedly at 10× depth, because the conservative filters, such as minimum depth, allele-fraction and strand-balance requirements, and evidence-count thresholds, discard many true variants and true reference calls once the effective coverage falls below these cutoffs.

AccuSNV maintains high accuracy on large and noisy simulated datasets

In real-world scenarios, sequencing samples from different isolates often have varying depths and complex phylogenetic relationships. Moreover, previous studies have shown that the divergence between the target genome and the reference genome significantly impacts bacterial SNV calling performance (Bush et al. 2020). Lastly, many real-world datasets contain large isolate collections, introducing additional complexity to SNV detection. To better reflect these real-world challenges, we generated an additional large-scale, noisy simulated dataset for evaluation. Notably, FreeBayes and breseq were excluded from this experiment and subsequent real-world data evaluations due to their lower accuracy in earlier evaluations and substantially longer runtimes (Supplementary Table S1).

To investigate the influence of the reference genome, we first generated four root genomes per species of varying reference divergence from the reference genome.

From each root genome, we simulated a phylogenetic tree with 100 strains using Msprime (Baumdicker et al. 2021), which provided the number of mutations along each branch for the given mutation rate (used 5×10^{-10} here). We then assigned 80% of the mutations as SNVs and 20% as indels, and used SimuG to generate the corresponding mutated genomes. We focused solely on SNVs and indels without larger recombination or rearrangement events, as this analysis targets very closely related isolates where such structural variations are rare. Paired-end reads were simulated at random depths between 15× and 70×. In total, this yielded 2,400 samples (100 isolates × 6 species × 4 reference divergence levels) for evaluating SNV calling tools (Supplementary Fig. S5B).

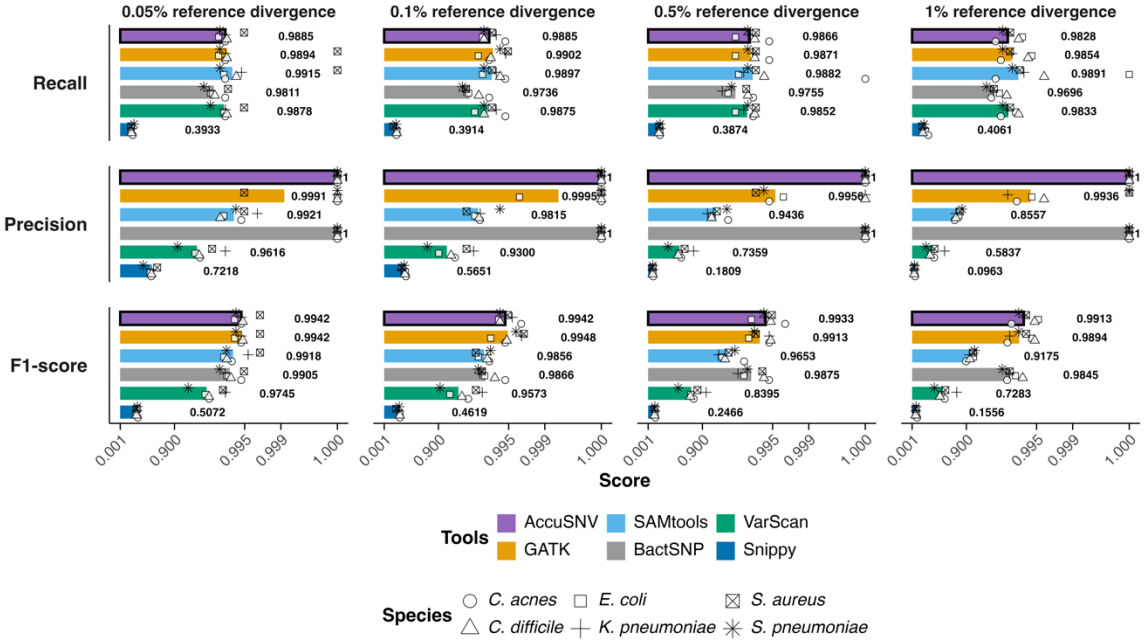

Figure 4. ***AccuSNV maintains high accuracy on large and noisy simulated datasets.*** *Performance comparison of six SNV calling tools across four reference divergence levels. Axes use a transformed scale ($-log_{10}(1.0001 - x)$) to enhance visualization in the higher range (0.9-1). Shapes represent species and colors represent tools. Panels show average recall (top), precision (middle), and F1 score (bottom) across all species. AccuSNV bars are highlighted with a black outline.*

AccuSNV exhibits the most competitive performance among all tested tools, with average F1 scores exceeding 99% across all datasets (Fig. 4). It maintains perfect precision across all reference divergence levels and achieves an F1 score above 99.1% even at the highest reference divergence (1%). In contrast, other tools show greater variability: VarScan, SAMtools, and Snippy all experience significant drops in precision or recall at higher reference divergence levels. Although GATK performs comparably to, or slightly better than, AccuSNV at the 0.05% and 0.1% reference divergence levels in terms of F1 score, this is driven by increased recall at the cost of reduced precision, which is suboptimal when comparing closely-related

genomes where precision is critical (Bush 2021). BactSNP also maintains perfect precision across all datasets but with consistently lower recall, leading to reduced F1 scores, especially under higher reference divergence.

Overall, all tools exhibit decreased performance as the divergence between the root genome and the reference genome increases, highlighting the growing difficulty of variant calling when read-reference mismatches become more prevalent. Despite this, AccuSNV shows minimal performance fluctuation across reference divergence levels. In addition, tools such as SAMtools and VarScan, which performed well in the previous experiments with only 10 strains at 50×, uniform sequencing depth, and simplified phylogenetic structure, showed markedly reduced performance on this more realistic and noisy dataset. The performance degradation is particularly pronounced for Snippy, which is sensitive to false-positive calls under increased reference divergence and noise due to misalignments near indels and other structural differences (Supplementary Fig. S6) (Yoshimura et al. 2019). These results indicate that the robustness of existing tools may decline under more complex and realistic sequencing conditions. In contrast, AccuSNV maintains both high accuracy and consistency, supporting its use in large-scale bacterial variant detection across diverse and complex conditions.

While the experiment with varying sequencing depths reflects the realistic heterogeneity commonly observed in bacterial genomics studies, using a uniform sequencing depth allows us to examine the effect of reference divergence without the confounding effect of sequencing depth. We therefore conducted an additional controlled experiment with a uniform sequencing depth of 50× across all isolates to disentangle the impact of reference divergence from depth variation (Supplementary Fig. S7). Under this controlled setting, AccuSNV maintains top-tier performance across all reference divergence levels (0.05%–1%), still achieving F1 scores above 99.1% with perfect precision even at 1% divergence.

Experiments on highly variable isolates

Some real-world bacterial populations, such as those from environmental samples, can exhibit substantial genetic variability between isolates. To evaluate how well different tools perform under such conditions, we varied the mutation rate in 100 *E. coli* genomes using Msprime (see Supplementary Fig. S5B), based on a root genome containing 1% sequence variation. Specifically, we tested mutation rates of 5×10^{-9}, 5×10^{-8}, and 1×10^{-7}, along with the original rate of 5×10^{-10} per site per generation. We reran all tools on these datasets and recorded the results (Supplementary Fig. S8).

As mutation rates increased, all tools exhibited more false positives and false negatives. Nevertheless, AccuSNV achieved a high average F1 score (98.85%) by balancing recall and precision. GATK (with 98.81% average F1 score) showed the most competitive F1 score at higher mutation rates but had lower precision than AccuSNV. In contrast, BactSNP (with 98.11% average F1 score) maintained higher

precision across all datasets, but its recall was substantially lower. SAMtools (with 98.33% average F1 score) achieved the best recall across all datasets, yet it produced more false positives at higher mutation rates. Overall, AccuSNV demonstrated the highly robust and reliable performance across mutation rates, underscoring its effectiveness for analyzing highly variable bacterial isolates.

Computational efficiency comparison

To compare the running time and memory usage of different tools, we applied all tools to 10-strain and 100-strain *E. coli* simulated datasets and recorded the computational performance metrics (Supplementary Table S1). AccuSNV demonstrated competitive efficiency, completing analysis in 30.3 minutes for 10 isolates (50×) and 64.5 minutes for 100 isolates (15×-70×), with memory usage of 1,140 MB and 1,340 MB respectively. While some tools like Snippy showed faster execution times (5 minutes on a 10-strain dataset and 50 minutes on a 100-strain dataset), AccuSNV maintained better performance without the extreme computational demands observed in tools like FreeBayes and breseq, which failed to complete within reasonable time limits on the larger dataset (running more than 2 days on the 100-strain dataset). The results demonstrate that AccuSNV provides a balance between accuracy and computational efficiency for bacterial SNV calling applications.

AccuSNV shows good robustness in simulated contaminated samples

To assess the robustness of SNV calling tools under contamination, we simulated datasets with increasing levels of contamination, either from other strains of the same species or from closely related species, under both high and low sequencing depths. AccuSNV consistently achieved higher F1 scores than other tools in low-depth contaminated datasets and maintained perfect precision in all tested datasets. In high-depth settings (50× depth) with contamination from both other strains and closely related species, VarScan achieved the highest F1 scores across all datasets, with AccuSNV ranking second due to slightly lower recall. However, in this high-depth setting, when contamination originated from closely related species, VarScan exhibited reduced precision compared to AccuSNV. Full results are provided in Supplementary Section 2.1, Fig. S9-S12.

AccuSNV demonstrates high accuracy on real-world test data

Real-world data from independent studies, featuring diverse sequencing protocols, depths, species, and collection contexts, offer a more rigorous assessment for evaluating the robustness and accuracy in practical applications, as simulations rarely include all types of error and variability in the real world. In this experiment, we evaluated AccuSNV and other tools using four publicly available bacterial sequencing datasets (Snitkin et al. 2012; Giulieri et al. 2022; Kim et al. 2014; Buddle

et al. 2024), each derived from a distinct study, author group, and accompanied by manually curated and reported SNVs that underwent careful quality control rather than being taken directly from automated tool outputs. The four datasets differ in read types (single-end vs. paired-end), bacterial species, and isolate numbers (Supplementary Fig. S13). The dataset from Snitkin et al. (2012) comprises single-end reads from a hospital outbreak of *Klebsiella pneumoniae*, while the others involve paired-end sequencing of *Staphylococcus aureus* or *Clostridioides difficile* collected under varying evolutionary or clinical contexts. These differences in sequencing protocols, strain diversity, and sampling settings create a heterogeneous and realistic benchmark for evaluating SNV-calling accuracy and robustness.

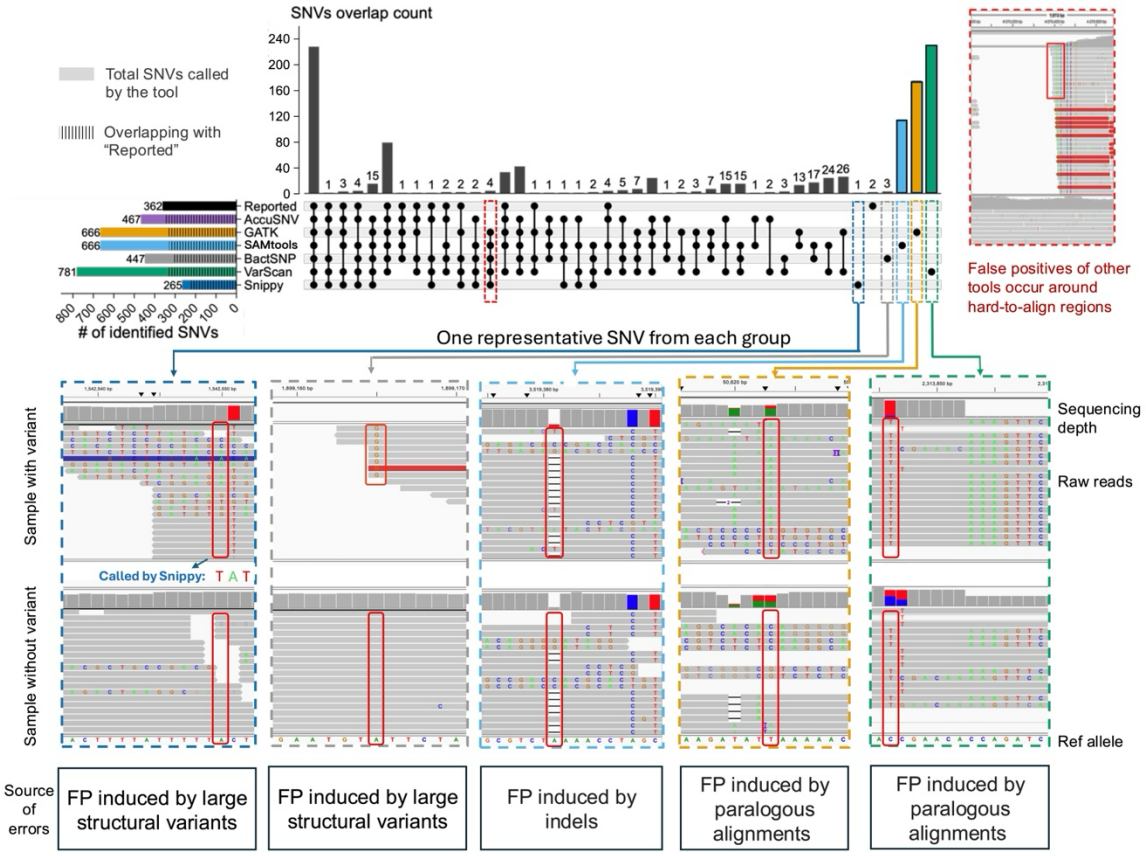

Figure 5. ***AccuSNV demonstrates high accuracy and precision on additional real-world bacterial sequencing datasets.*** *We evaluated SNV concordance between AccuSNV and five other tools across four published bacterial datasets with curated SNVs (Supplementary Fig. S13). UpSet plots (center) display the SNVs overlap count of tool-identified SNVs and reported SNVs. The red-outlined dash box highlights a group of four SNVs identified by all tools except AccuSNV and the original study (corresponding IGV (Robinson et al. 2011) screenshot is shown in the top right panel). The IGV screenshot includes two sample tracks, one carrying the reference allele and one carrying the alternative allele, with the variant position*

highlighted by a red box. Red reads indicate an inferred insert size larger than expected, which may suggest the presence of a deletion or a possible alignment artifact. The bottom panels display IGV screenshots of five unique SNVs identified by other methods but not supported by AccuSNV or the original study. The colored borders around each IGV screenshot correspond to the dashed boxes in the UpSet plot, indicating the group from which each SNV originated. Below each IGV screenshot is a description of the source of errors, highlighting typical error signatures such as misalignments near large structural variants, indels, and paralogous alignments. (as reported in (Koboldt 2020)). We note that these error modes are not generally tool specific, but that these tool-specific SNVs happen to have these error modes. Also see Supplementary Fig. S14-S17.

All tools called distinct sets of SNVs across these data sets, with none matching the reported SNVs exactly (Fig. 5). Across all four datasets, AccuSNV demonstrates strong concordance with reported SNVs while minimizing tool-specific SNVs unsupported by other methods or the original studies (only 110 unreported SNVs, compared to 127-422 for other tools, except Snippy, which had 26 but identified much fewer reported SNVs).

Inspection of discordances among SNV calling methods illustrates that AccuSNV avoids false positives called by other tools, generally caused by alignment errors. For example, a noteable group of SNVs is highlighted in the red dashed box (Fig. 5); these four SNVs from the same study were called by all other tools but not called by AccuSNV or the original study. To investigate the basis of this discrepancy, we visualized the corresponding BAM files of these four SNVs, one of which is shown in the panel outlined on the right. These four SNVs appear to be false positives located in hard-to-align regions, due to divergent regions between the sample and reference.

In addition, all tools except AccuSNV reported unique SNVs that were not supported by any other method or the original study. To further examine these unique SNVs, we manually reviewed five random unique SNVs from tested tools and visualized their read alignments (bottom panel of Fig. 5). In all cases, the alignments revealed problematic mapping features, such as soft clipping, inconsistent base support, or alignment gaps. These findings suggest that these unique SNVs called by other tools are false positives resulting from alignment errors. This observation agrees with previous studies, such as the one by Koboldt (2020) (Koboldt 2020), which identified similar artifacts in short read alignments leading to false-positive variant calls. While the IGV examples in Fig. 5 highlight representative alignment-related error modes, we further assessed whether these patterns generalize across datasets by quantifying site-level signatures associated with reported and unique SNVs for each variant caller. Across four real-world datasets, we observed that different callers exhibit distinct but dataset-dependent distributions of allele balance, sequencing depth, and mapping quality (Supplementary Fig. S14-S17). These results indicate that although certain error

modes recur, their quantitative manifestations vary substantially across datasets, complicating the use of universal hard filtering thresholds. In contrast, AccuSNV is able to filter these false positives by capturing informative across-sample features from alignment data, rather than relying on manually defined thresholds or manual filtering of individual positions.

Output and downstream analysis modules of AccuSNV

To support users with varying levels of computational expertise, AccuSNV offers comprehensive output and downstream analysis modules based on the identified SNVs. The core output includes a compressed SNV table in NumPy's `.npz` format (a compact binary file for storing arrays), a human-readable tab-separated summary of all the SNVs (TSV), including the allele call for each sample, a graphical HTML report, and a set of quality control (QC) figures (Fig. 6).

The HTML report provides an interactive platform for variant review. Each predicted SNV is visualized with a bar chart showing allele frequency distributions across isolates (as in Fig. 1), alongside detailed metadata such as strand-specific depth, mapping quality statistics, and various filter flags. A demo output HTML report can be found at https://heruiliao.github.io/. This design helps users assess variant confidence and identify potentially ambiguous cases that may require manual curation. In addition, AccuSNV generates multiple QC figures to assist in data quality assessment, including heatmaps of genome coverage and allele count distributions across samples. The human-readable tab-separated TSV file presents essential information for each SNV for rapid inspection and integration with external pipelines. Lastly, the compressed SNV Numpy table stores all variant-level features and prediction outputs in an efficient binary format, which serves as input for downstream modules.

To facilitate related applications, AccuSNV includes built-in modules for common downstream analyses. These modules enable manual filtering, phylogenetic tree construction, calculation of d_N/d_S ratios to assess selective pressures, and identification of homoplasic SNVs (Edwards et al. 2021). Additionally, users can export filtered variants for phylogenetic analysis with external tools.

Together, these outputs and utilities make AccuSNV a versatile and accessible tool for high-confidence bacterial SNV detection and interpretation.

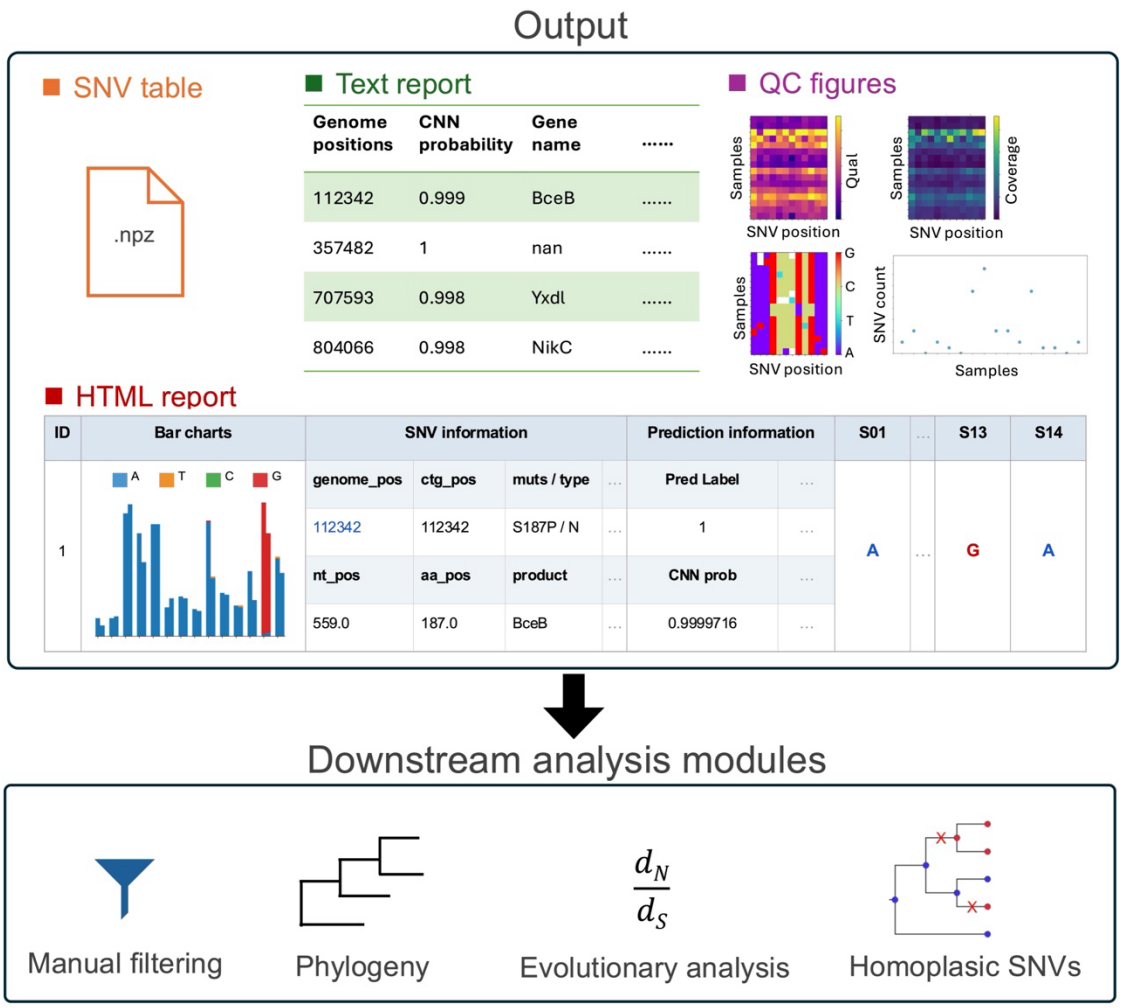

Figure 6. Output and downstream analysis modules of AccuSNV. The core outputs include: (1). a compressed SNV table in NumPy's .npz format storing detailed feature vectors and prediction scores, (2). a human-readable text summary file (TSV) listing all identified SNVs with key attributes, (3). a set of quality control (QC) figures summarizing read coverage, base calls, mapping quality scores, etc, (4). a HTML report that integrates summary tables and bar charts of identified variants. Homoplasic SNVs in the figure refers to variants where the same derived nucleotide arises independently in two or more lineages since their divergence from a common ancestor with a different ancestral base. These SNVs can be used to infer parallel, convergent, or revertant evolutionary events (Edwards et al. 2021).

Discussion

In this study, we present AccuSNV, a deep learning-based variant calling framework designed for high-precision single nucleotide variant (SNV) detection from bacterial whole-genome sequencing (WGS) data. By encoding multisample read alignment

data into structured feature vectors and leveraging a convolutional neural network (CNN), AccuSNV effectively captures across-sample patterns that are often ignored by conventional methods. Our approach addresses key challenges in bacterial variant calling, including low sequencing depth, high intraspecies diversity, and genome complexity. As a result, AccuSNV enables accurate bacterial SNV calling for users with varying levels of expertise, without the need for hard-coded thresholds or extensive manual filtering. While a few tools occasionally achieved comparable or better performance under specific conditions, AccuSNV was consistently the best or among the top-performing methods across all sequencing depths, reference and sequence divergence levels, and contamination scenarios (Figs. 3-5), demonstrating its broad applicability and scalability for bacterial genomics research. Interestingly, GATK performed as or nearly as well as AccuSNV under many conditions with default parameters on simulated data, likely because of its integration of across-sample comparisons (Fig. 3 and Fig. 4), but still had more identified false-positives when challenged with real world data (Fig. 5), likely due to structural variations between the reference and real samples. We note that all the tested pipelines can yield reliable results in many applied studies, particularly when users manually tune parameters, apply additional post hoc filtering, or visually inspect alignments, especially for high-depth datasets with closely related references. However, this reliance on dataset-specific decisions and expert judgment can limit reproducibility and scalability across large or heterogeneous cohorts, where systematic alignment-related error modes become more prominent (Yoshimura et al. 2019). AccuSNV minimizes manual parameter tuning by learning to identify false-positive SNVs directly from across-sample alignment patterns, enabling consistent performance across sequencing depths and reference divergence levels. Feature ablation analysis revealed that this robustness arises from integrating multiple complementary features rather than depending on any single signal, allowing the CNN to adapt to diverse sequencing conditions (Supplementary Fig. S4).

Despite these advances, AccuSNV has several limitations. First, the current implementation is designed and trained specifically for short-read sequencing data. Extending AccuSNV to long-read sequencing platforms such as Oxford Nanopore or PacBio would require retraining on platform-specific datasets and potentially incorporating features that capture their distinct error profiles. Second, misclassfications may still arise in extremely challenging or ambiguous cases encountered in real-world data, especially those involving false-positive SNVs arising from an error mode not represented in the training set. For example, although the training and test species span a broad range of genome sizes and GC content commonly observed in bacterial pathogens and commensals (Supplementary Table S2), AccuSNV has not yet been evaluated on bacteria with extreme genomic features (e.g., GC content <20% or >70%) or other haploid organisms such as yeast To enable users to evaluate AccuSNV's performance on new use cases, AccuSNV provides an interactive HTML report and multiple quality

control visualizations that allow users to inspect candidate SNVs and flag potential false positives for downstream curation. Third, AccuSNV is specifically designed for SNV calling and does not currently support the identification of structural variants (SVs), such as insertions, deletions, or inversions. Fourth, the method is not directly applicable to metagenomic datasets, where a single sample may contain multiple closely related strains, resulting in more complex and ambiguous alignment patterns than those encountered in isolate-based WGS. Finally, while the across-sample alignment patterns that AccuSNV leverages to identify false positives may be consistent across organism types, the current implementation is limited to haploid genomes. Diploid genomes present a fundamentally different classification problem: distinguishing true heterozygous variants from sequencing or alignment errors, rather than simply identifying sites that deviate from a single reference allele.

We note that in all of our evaluation experiments, we assess each genomic position as either a true or false variant site, rather than evaluating individual per-sample variant calls. Thus, many individual false calls are reduced to only a single count in our evaluations. In practical applications, especially under low coverage or high reference divergence, many samples may have false-positives or false-negatives at a single position. As such, the position-level evaluation presented here provides a conservative benchmark of tool performance in these noisy and complex scenarios.

Looking forward, several directions can extend the utility of AccuSNV. One promising avenue is to expand the framework to support small indel detection by incorporating indel-aware features and labels during model training. Another direction is to explore whether across-sample alignment patterns in metagenomic data can be modeled to distinguish strain-specific SNVs, potentially enabling AccuSNV to operate in complex microbial communities. Extending the approach to diploid genomes is another promising direction that would require incorporating heterozygosity-aware features and training on datasets with appropriately labeled homozygous and heterozygous variants. However, these efforts may be constrained by the limited availability of real-world datasets with manually validated labels.

In conclusion, by reducing the need for manual thresholding and incorporating interpretable quality controls, AccuSNV makes high-confidence bacterial SNV calling accessible to users with varying levels of computational expertise. This accessibility makes it particularly well suited for studies of within-microbiome evolution, large-scale evolution experiments, and bacterial epidemiology, even when working with low-coverage or noisy sequencing data. By lowering technical barriers, AccuSNV enables broader adoption of high-precision SNV calling in diverse research and clinical settings.

Methods

Overview of AccuSNV

AccuSNV is designed for high-precision single nucleotide variant (SNV) calling from whole-genome sequencing data of bacterial isolates. It takes a reference genome and whole-genome sequencing short-read data from multiple bacterial samples (isolates) as input, and outputs identified SNVs along with associated information such as annotations, predicted probabilities, and visualization results, etc (Fig. 2 and 6). AccuSNV leverages deep learning to distinguish true variants from sequencing or mapping artifacts, eliminating the need for manual filtering. Specifically, to address challenges such as low sequencing depth, genomic intraspecies diversity, variability in sample size, and sequencing quality across datasets, AccuSNV employs a CNN-based framework to learn informative patterns from multi-sample read alignment data. Here, we chose CNN for its ability to extract local features and maintain translation invariance (Lecun et al. 1998), which enables the robust detection of across-sample patterns from diverse and noisy bacterial isolate alignments. To implement this CNN-based approach, AccuSNV first identifies candidate SNV sites using SAMtools and extracts read-level features from bacterial isolate alignments using a Snakemake-based pipeline (Fig. 2A), including reads orientation, reads counting, mapping quality score, etc (Supplemetary Fig. S1). These features are converted into a four-dimensional numerical feature vector, allowing the CNN to better capture both local and across-sample patterns. The CNN model (Fig. 2B) was trained using these four-dimensional feature vectors as input, and the final pretrained model is provided to perform binary SNV classification on new datasets.

Reference genomes used in this study

Cutibacterium acnes (*C. acnes C1*, GCF_000302515.1), *Clostridium difficile* (*C. diffcile R20291*, GCF_000027105.1), *Escherichia coli* (*E. coli str. K-12 substr. MG1655*, GCF_000005845.2), *Klebsiella pneumoniae* (*K. pneumoniae subsp. pneumoniae NTUH-K2044*, GCF_000009885.1), *Staphylococcus aureus* (*S. aureus subsp. aureus NCTC 8325*, GCF_000013425.1), *Streptococcus pneumoniae* (*S. pneumoniae R6*, GCF_000007045.1) genomes were selected as reference genomes in this study.

Training and validation datasets

To train and evaluate the model, we collected a total of 7,638 “quality-filtered” SNVs (labeled as “True”) from 5,432 isolates from three datasets published in previous studies (Zhao et al. 2019; Conwill et al. 2022; Baker et al. 2025) conducted by our lab, and 6,306 “low-quality” SNVs filtered out during processing for these studies (labeled as “False”). These datasets comprise 222 lineages (finely resolved

643 clades separated by fewer than 100 mutations across the core genome (Conwill et
644 al. 2022; Baker et al. 2025)) and include carefully curated SNVs based on extensive
645 manual validation of filters in a study-specific manner. The criteria used in each
646 study to define "quality-filtered" SNVs are summarized in Supplementary Table S3.
647 To verify the collected positions in the training data, we randomly picked up to 10
648 true and 10 false positions per lineage (for a maximum of 20 per lineage), but some
649 lineages contained fewer than 20 available sites, resulting in 2,859 positions that
650 were manually inspected using generated bar charts (see example in Fig. 1). These
651 bar charts are available at https://zenodo.org/records/17058222. Manual
652 inspection confirmed that the assigned labels were accurate and consistent with
653 the expected patterns. These data were split into training and validation sets at an
654 4:1 ratio, stratified by lineages across the three datasets. This ensured that the
655 validation set contained isolates from different lineages and datasets, resulting in
656 10,887 SNVs for training and 3,057 SNVs for validation, and thereby improving the
657 generalizability of the trained model. Accession numbers and additional details
658 regarding the training data are provided in Supplementary Table S4. Additionally, to
659 assess tool performance on real-world data from independent studies, we
660 collected 362 reported SNVs from four different published datasets, where SNVs
661 had been carefully validated in the original studies through stringent filtering and
662 manual inspection rather than taken directly from automated tool outputs. Detailed
663 information on all seven datasets is provided in Table 1.

Datasets ID	Type	Species	# of lineages	# of isolates	# of True SNVs	# of False SNVs
Zhao_2019 (Zhao et al. 2019)	Training/Validation	*B. fragilis*	11	556	227	492
Conwill_2022 (Conwill et al. 2022)		*C. acnes*	50	854	1,972	3,840
Baker_2025 (Baker et al. 2025)		*C. acnes*	85	2,007	3,444	554
		S. epidermidis	76	2,015	1,995	1,420
Kim_2014 (Kim et al. 2014)	Test	*S. aureus*	1	121	265	-
Buddle_2024 (Buddle et al. 2024)		*C. difficile*	1	96	55	-

Datasets ID	Type	Species	# of lineages	# of isolates	# of True SNVs	# of False SNVs
Snitkin_2012 (Snitkin et al. 2012)		*K. pneumoniae*	1	20	31	-
Giulieri_2022 (Giulieri et al. 2022)		*S. aureus*	1	16	11	-

Table 1. Summary of the seven real bacterial whole-genome sequencing datasets used in this study. The first three datasets were used for model training and validation, while the remaining four served as independent test sets for additional evaluation. SNV calling was performed within lineages, defined as closely related clades identified based on core-genome similarity (Conwill et al. 2022; Baker et al. 2025). "-" indicates no reported false SNVs in the original study.

Identification of candidate SNVs

To identify candidate SNV sites from raw sequencing data, we implement a modular pipeline built with Snakemake (Mölder et al. 2025). The process begins with quality trimming of reads using cutadapt (v.1.18) (Martin 2011) and Sickle (Joshi and Fass 2011) (v.1.33; -g -q 20 -l 50 -x -n), followed by reference-based alignment via BWA-MEM (Li and Durbin 2009) (v.0.7.18). Aligned reads are converted to sorted and indexed BAM files. SAMtools markdup (v.1.21.1; -r -s -d 100 -m s) is used to remove duplicate reads from each sample. Pileup and VCF files are then generated using SAMtools (Li et al. 2009) mpileup (v.1.21.1; -q30 -x -s -O -d3000), and BCFtools (Danecek et al. 2021) view (v.1.21.1; -Oz -v snps -q .75) . Critically, AccuSNV leverages Snakemake's workflow management to automatically handle task dependencies and enable parallel execution of independent, sample-level preprocessing steps across multiple cores or cluster nodes when resources are available. This parallel execution strategy substantially reduces computational time compared to sequential preprocessing approaches, particularly for large datasets. As a result, while runtime scales approximately linearly with dataset size for tools that require predominantly sequential preprocessing (e.g., GATK), AccuSNV maintains efficient performance when scaling from tens to hundreds of isolates (Supplementary Table S1).

For each input sample, preliminary SNVs are extracted from VCF files and saved in compressed intermediate files. Then, the pipeline aggregates variant positions across all samples, and combines these positions with alignment-derived metrics, such as read counting and mapping quality score, into a candidate mutation table using custom Python scripts. This table contains informative alignment statistics for each candidate SNV and serves as the input for downstream analysis.

Feature extraction

To ensure the CNN can effectively capture informative patterns across bacterial isolates, AccuSNV transforms raw alignment data into structured four-dimensional feature vectors. Specifically, for each candidate SNV position, we aggregate information from all reads supporting each allele in each sample, including base counts on forward and reverse strands, mapping quality scores, and indel signals (the number of reads supporting insertions and deletions, broadcast across the four base channels per sample; Supplementary Fig. S1). Notably, these features were selected because they are commonly used in the manual filtering step across previous studies (Zhao et al. 2019; Conwill et al. 2022; Key et al. 2023) for distinguishing "quality-filtered" SNVs from "low-quality" positions. In addition, we incorporate two normalized coverage-based features, $d_{i,j}^{\text{norm}}$ and $r_{i,j}$, to better represent both across-position and across-sample variation. These two features are computed separately for forward and reverse strands to retain strand-specific information:

$$d_{i,j}^{\text{norm}} = \frac{d_{i,j}}{median(D_i)}$$

where $d_{i,j}$ is the raw read depth of sample i at site j, and $median(D_i)$ is the median depth across all positions in sample i.

$$r_{i,j} = \frac{d_{i,j}}{\sum_{x=1}^{n} d_{x,j}}$$

Here, $r_{i,j}$ refers to the proportion of read depth from sample i relative to the total depth across all n samples at position j, serving as a relative coverage signal across isolates.

During feature extraction, we further observed that certain low-quality positions display extremely unbalanced coverage distributions, where the depth of the putative minor allele is orders of magnitude lower than that of the major allele (see Supplementary Fig. S18). Such unusual "gap cases" are underrepresented in the training data and tend to be misclassified by the CNN. To ensure these extreme cases are consistently represented, we introduced a normalization-based adjustment in the feature encoding step, implemented through a *Z*-score framework:

$$Z = \frac{\bar{x}_1 - \bar{x}_2}{\sigma_1}$$

where $\bar{x}_1$ and $\bar{x}_2$ denote the normalized read depths ($d_{i,j}^{\text{norm}}$) of the major-allele–supporting and alternate-allele–supporting samples, respectively, and σ_1 is the standard deviation of normalized read depths among the major-allele–supporting samples. The one-sided p-value is then calculated as:

$$p = 1 - \Phi(|Z|)$$

where Φ denotes the cumulative distribution function of the standard normal distribution. Positions with $p < 0.01$ are flagged as gap cases, and the corresponding feature values are normalized to ensure consistent representation across isolates. This adjustment is fully integrated into the feature construction process, making the AccuSNV more robust to highly skewed depth profiles. We also conducted an ablation study on this normalization-based adjustment to evaluate it's impact on performance (Supplementary Section 2.2). The results show that this design improves the robustness of AccuSNV on datasets containing positions with extreme coverage imbalance by reducing false positives while maintaining high recall (Supplementary Fig. S8).

These features are then encoded as a four-dimensional feature vector (Fig. 2), where the last dimension represents the nucleotide channels. This four-dimensional feature encoding has two key advantages over traditional feature representation like raw VCF statistics or flattened alignment matrices. First, it encodes comprehensive alignment features in a structured, image-like format that is well suited for the CNN model, enabling it to leverage translation invariance and local feature extraction. Second, it preserves across-sample information, enabling the model to capture shared or contrasting patterns among isolates at each site. These patterns are not accessible to conventional single-sample variant callers but are essential for distinguishing true SNVs from sequencing or alignment artifacts.

Deep learning framework

Given the extracted feature vectors, we employ a convolutional neural network to classify the candidate SNVs. There are two major challenges in applying CNN to this task. First, the number of bacterial isolates varies across datasets, leading to input tensors with different dimensions along the sample axis. Such variability poses a problem for standard CNN architectures, which require fixed input dimensions for training and inference. Second, compared to natural image tensors that often contain rich spatial information and high-dimensional features, our input tensors have a much lower feature dimensionality. Commonly used deep or parameter-heavy models are therefore prone to overfitting, especially given the limited number of labeled training examples.

To address these issues, we designed a CNN architecture to accommodate the varying alignment data. The model consists of convolutional, pooling, and fully connected layers (Fig. 2B). Specifically, we used three convolutional layers to hierarchically capture both localized signal structures and broader across-sample patterns. These three layers use 32, 64, and 128 filters with kernel sizes of 3×4, 2×1, and 1×1, respectively. Each convolutional layer is followed by a ReLU activation function. To accommodate inputs with variable sample sizes, we apply an adaptive average pooling operation (Hsin and Su 2021) along the sample axis,

compressing variable-length inputs into a fixed-size representation. Unlike standard pooling layers with fixed kernel sizes, adaptive pooling dynamically divides the input into regions such that the output always has the same predefined shape, regardless of the input size. Let the $\boldsymbol{X} \in \mathbb{R}^{x \times y \times z \times 4}$ be the input feature vector, where x is the number of candidate SNV postions, y is the number of samples (bacterial isolates), z is the number of features, and 4 is the input channels. Then, the output tensor $\boldsymbol{Z}$ of the adaptive average pooling layer is:

$$\boldsymbol{Z} = \text{AdaptiveAvgPool}(f_{\text{conv}}(\boldsymbol{X})), \boldsymbol{Z} \in \mathbb{R}^{x \times c \times 1 \times 1}$$

where c is the number of output channels from the last convolutional layer. This operation enables the model to generalize across datasets with different sample sizes while learning informative patterns across isolates. The output of the pooling layer is fed into a fully connected layer with 64 units, followed by dropout ($p = 0.5$) and a final sigmoid output node that produces the probability of the SNV being true. By default, the classification cutoff is 0.5.

Model training

The model was trained using real-world bacterial whole-genome sequencing data with curated labels obtained from previously published studies (Baker et al. 2025; Zhao et al. 2019; Conwill et al. 2022), as summarized in Table1. During training, the binary cross-entropy loss function was used as the objective function.

$$\mathcal{L} = -\frac{1}{N}\sum_{i=1}^{N}[y_i \log(\hat{y}_i) + (1 - y_i)\log(1 - \hat{y}_i)]$$

where $y_i \in \{0,1\}$ is the true label of the i-th candidate SNV and $\hat{y}_i \in (0,1)$ is the predicted probability. We implemented the model using PyTorch, and trained it using the Adam optimizer with a learning rate of 0.0001, a batch size of 32, and early stopping based on validation loss. The model was trained for 150 epochs, and the one with the lowest validation loss was saved for all classification tasks in our evaluation experiments. The learning curve of the model training is shown in Supplementary Fig. S19. To assess feature importance, we performed an ablation study by masking individual feature groups (setting values to zero) and retraining the model using the same hyperparameters and data splits (Supplementary Fig. S4).

Evaluation metrics

To evaluate the performance of different tools, we used five standard metrics: accuracy, precision, recall, F1 score, and area under the curve (AUC). Let TP, FP, TN and FN denote the number of true positives, false positives, true negatives, and false negatives, respectively. The evaluation metrics are defined as:

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$
$$\text{Precision} = \frac{TP}{TP + FP}$$
$$\text{Recall} = \frac{TP}{TP + FN}$$
$$\text{F1 score} = \frac{2 \cdot \text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}$$

AUC was calculated based on different models' predicted probabilities and reflects their ability to distinguish true SNVs from false ones across different decision thresholds.

Data sets

Real-world datasets used in this paper are all publicly available. Bar charts for the 2,859 positions selected for manual label inspection are available at https://zenodo.org/records/17058222. A demo output HTML report of AccuSNV can be found at https://heruiliao.github.io/.

Code availability

The source code of AccuSNV is freely available at https://github.com/liaoherui/AccuSNV and as Supplemental Code. AccuSNV is also available via Bioconda at https://anaconda.org/bioconda/accusnv. The command lines for running other SNV calling tools are provided in the Supplemental Methods. The custom scripts necessary to reproduce the benchmarking experiments can be accessed at GitHub (https://github.com/liaoherui/accusnv-eval) and as Supplemental Code.

Competing interests

The authors declare no competing interests.

Acknowledgements

This work has been supported by MIT-Novo Nordisk Artificial Intelligence Postdoctoral Fellowship to HL, NIH grant 1DP2GM140922 and R35GM156282 to TDL. ILM, MF, and FMK were supported by the Max Planck Society. During the preparation of this work, the authors used ChatGPT (https://chatgpt.com/) in order to rephrase sentences. We thank members of the Lieberman Lab and Key Lab for helpful comments.

Author contributions

Methodology: HL, PT, AC, EQ, AHM, JSB, MF, ILM, FMK, TDL; Investigation: HL, AHM, TDL, LB; Funding acquisition: HL, TDL; Writing: HL, TDL.

References

Ahmed, Shams Forruque, Md. Sakib Bin Alam, Mahtabin Rodela Rozbu, et al. 2023. "Deep Learning Modelling Techniques: Current Progress, Applications, Advantages, and Challenges." *Artificial Intelligence Review* 56 (11): 13521–617. https://doi.org/10.1007/s10462-023-10466-8.

Baker, Jacob S., Evan Qu, Christopher P. Mancuso, A. Delphine Tripp, Arolyn Conwill, and Tami D. Lieberman. 2025. "Intraspecies Dynamics Underlie the Apparent Stability of Two Important Skin Microbiome Species." *Cell Host Microbe* 33: 643–56. https://doi.org/10.1016/j.chom.2025.04.010.

Barrick, Jeffrey E, Geoffrey Colburn, Daniel E Deatherage, et al. 2014. "Identifying Structural Variation in Haploid Microbial Genomes from Short-Read Resequencing Data Using Breseq." *BMC Genomics* 15: 1039. https://doi.org/10.1186/1471-2164-15-1039.

Barrick, Jeffrey E., Dong Su Yu, Sung Ho Yoon, et al. 2009. "Genome Evolution and Adaptation in a Long-Term Experiment with Escherichia Coli." *Nature* 461: 1243–47. https://doi.org/10.1038/nature08480.

Baumdicker, Franz, Gertjan Bisschop, Daniel Goldstein, et al. 2021. "Efficient Ancestry and Mutation Simulation with Msprime 1.0." *Genetics* 220: iyab229. https://doi.org/10.1093/genetics/iyab229.

Buddle, Jessica E., Lucy M. Thompson, Anne S. Williams, et al. 2024. "Identification of Pathways to High-Level Vancomycin Resistance in Clostridioides Difficile That Incur High Fitness Costs in Key Pathogenicity Traits." *PLOS Biology* 22 (8): e3002741. https://doi.org/10.1371/journal.pbio.3002741.

Bush, Stephen J. 2021. "Generalizable Characteristics of False-Positive Bacterial Variant Calls." *Microbial Genomics* 7 (8): 000615. https://doi.org/10.1099/mgen.0.000615.

Bush, Stephen J, Dona Foster, David W Eyre, et al. 2020. "Genomic Diversity Affects the Accuracy of Bacterial Single-Nucleotide Polymorphism–Calling Pipelines." *GigaScience* 9 (2): giaa007. https://doi.org/10.1093/gigascience/giaa007.

Conwill, Arolyn, Anne C. Kuan, Ravalika Damerla, et al. 2022. "Anatomy Promotes Neutral Coexistence of Strains in the Human Skin Microbiome." *Cell Host Microbe* 30 (2): 171–182.e7. https://doi.org/10.1016/j.chom.2021.12.007.

865 Danecek, Petr, James K Bonfield, Jennifer Liddle, et al. 2021. “Twelve Years of
866 SAMtools and BCFtools.” *GigaScience* 10 (2): giab008.
867 https://doi.org/10.1093/gigascience/giab008.

868 Edwards, David J., Sebastián Duchene, Bernard Pope, and Kathryn E. Holt. 2021.
869 “SNPPar: Identifying Convergent Evolution and Other Homoplasies from Microbial
870 Whole-Genome Alignments.” *Microbial Genomics* 7 (12): 000694.
871 https://doi.org/10.1099/mgen.0.000694.

872 Fuchs, Sebastian A., Lisanna Hülse, Teresa Tamayo, Susanne Kolbe-Busch, Klaus
873 Pfeffer, and Alexander T. Dilthey. 2024. “NanoCore: Core-Genome-Based Bacterial
874 Genomic Surveillance and Outbreak Detection in Healthcare Facilities from
875 Nanopore and Illumina Data.” *mSystems* 9 (11): e01080–24.
876 https://doi.org/10.1128/msystems.01080-24.

877 Garrison, Erik, and Gabor Marth. 2012. *Haplotype-Based Variant Detection from*
878 *Short-Read Sequencing*. arXiv. https://doi.org/10.48550/arXiv.1207.3907.

879 Giulieri, Stefano G, Romain Guérillot, Sebastian Duchene, et al. 2022. “Niche-
880 Specific Genome Degradation and Convergent Evolution Shaping Staphylococcus
881 Aureus Adaptation During Severe Infections.” *eLife* 11: e77195.
882 https://doi.org/10.7554/elife.77195.

883 Goig, Galo A., Silvia Blanco, Alberto L. Garcia-Basteiro, and Iñaki Comas. 2020.
884 “Contaminant DNA in Bacterial Sequencing Experiments Is a Major Source of False
885 Genetic Variability.” *BMC Biology* 18 (1): 24. https://doi.org/10.1186/s12915-020-
886 0748-z.

887 Halachev, Mihail R, Jacqueline Z-M Chan, Chrystala I Constantinidou, et al. 2014.
888 “Genomic Epidemiology of a Protracted Hospital Outbreak Caused by Multidrug-
889 Resistant Acinetobacter Baumanniiin Birmingham, England.” *Genome Medicine* 6
890 (11): 70. https://doi.org/s13073-014-0070-x.

891 Hernández Medina, Ricardo, Svetlana Kutuzova, Knud Nor Nielsen, et al. 2022.
892 “Machine Learning and Deep Learning Applications in Microbiome Research.” *ISME*
893 *Communications* 2 (1): 98. https://doi.org/10.1038/s43705-022-00182-9.

894 Hsin, Hsi-Chin, and Chien-Kun Su. 2021. “Adaptive Pooling for Convolutional
895 Neural Networks with Arbitrary Input Sizes.” *2021 IEEE 3rd Eurasia Conference on*
896 *IOT, Communication and Engineering (ECICE)*, 196–98.
897 https://doi.org/10.1109/ecice52819.2021.9645730.

898 Huang, Weichun, Leping Li, Jason R. Myers, and Gabor T. Marth. 2011. “ART: A Next-
899 Generation Sequencing Read Simulator.” *Bioinformatics* 28 (4): 593–94.
900 https://doi.org/10.1093/bioinformatics/btr708.

Joshi, NA, and JN Fass. 2011. *Sickle: A Sliding-Window, Adaptive, Quality-Based Trimming Tool for FastQ Files*. GitHub repository.

Key, Felix M., Veda D. Khadka, Carolina Romo-González, et al. 2023. "On-Person Adaptive Evolution of Staphylococcus Aureus During Treatment for Atopic Dermatitis." *Cell Host Microbe* 31 (4): 593–603.e7. https://doi.org/10.1016/j.chom.2023.03.009.

Kim, Seungsoo, Tami D. Lieberman, and Roy Kishony. 2014. "Alternating Antibiotic Treatments Constrain Evolutionary Paths to Multidrug Resistance." *Proceedings of the National Academy of Sciences* 111 (40): 14494–99. https://doi.org/10.1073/pnas.1409800111.

Koboldt, Daniel C. 2020. "Best Practices for Variant Calling in Clinical Sequencing." *Genome Medicine* 12 (1): 91. https://doi.org/10.1186/s13073-020-00791-w.

Koboldt, Daniel C., Ken Chen, Todd Wylie, et al. 2009. "VarScan: Variant Detection in Massively Parallel Sequencing of Individual and Pooled Samples." *Bioinformatics* 25 (17): 2283–85. https://doi.org/10.1093/bioinformatics/btp373.

Lecun, Y., L. Bottou, Y. Bengio, and P. Haffner. 1998. "Gradient-Based Learning Applied to Document Recognition." *Proceedings of the IEEE* 86 (11): 2278–324. https://doi.org/10.1109/5.726791.

Li, Heng, and Richard Durbin. 2009. "Fast and Accurate Short Read Alignment with Burrows–Wheeler Transform." *Bioinformatics* 25 (14): 1754–60. https://doi.org/10.1093/bioinformatics/btp324.

Li, Heng, Bob Handsaker, Alec Wysoker, et al. 2009. "The Sequence Alignment/Map Format and SAMtools." *Bioinformatics* 25 (16): 2078–79. https://doi.org/10.1093/bioinformatics/btp352.

Liao, Herui, Jiayu Shang, and Yanni Sun. 2023. "GDmicro: Classifying Host Disease Status with GCN and Deep Adaptation Network Based on the Human Gut Microbiome Data." *Bioinformatics* 39 (12): btad747. https://doi.org/10.1093/bioinformatics/btad747.

Lopatkin, Allison J., Sarah C. Bening, Abigail L. Manson, et al. 2021. "Clinically Relevant Mutations in Core Metabolic Genes Confer Antibiotic Resistance." *Science* 371 (6531): eaba0862. https://doi.org/10.1126/science.aba0862.

Martin, Marcel. 2011. "Cutadapt Removes Adapter Sequences from High-Throughput Sequencing Reads." *EMBnet.journal* 17 (1): 10. https://doi.org/10.14806/ej.17.1.200.

McKenna, Aaron, Matthew Hanna, Eric Banks, et al. 2010. “The Genome Analysis Toolkit: A MapReduce Framework for Analyzing Next-Generation DNA Sequencing Data.” *Genome Research* 20 (9): 1297–303. https://doi.org/10.1101/gr.107524.110.

Min, Seonwoo, Byunghan Lee, and Sungroh Yoon. 2016. “Deep Learning in Bioinformatics.” *Briefings in Bioinformatics* 18: bbw068. https://doi.org/10.1093/bib/bbw068.

Mölder, Felix, Kim Philipp Jablonski, Brice Letcher, et al. 2025. “Sustainable Data Analysis with Snakemake.” *F1000Research* 10: 33. https://doi.org/10.12688/f1000research.29032.3.

Robinson, James T, Helga Thorvaldsdóttir, Wendy Winckler, et al. 2011. “Integrative Genomics Viewer.” *Nature Biotechnology* 29 (1): 24–26. https://doi.org/10.1038/nbt.1754.

Sayers, Eric W, Jeffrey Beck, Evan E Bolton, et al. 2024. “Database Resources of the National Center for Biotechnology Information in 2025.” *Nucleic Acids Research* 53 (D1): D20–29. https://doi.org/10.1093/nar/gkae979.

Schrader, Sarah M., Hélène Botella, Robert Jansen, et al. 2021. “Multiform Antimicrobial Resistance from a Metabolic Mutation.” *Science Advances* 7 (35): eabh2037. https://doi.org/10.1126/sciadv.abh2037.

Seemann, Torsten. 2015. *Snippy: Fast Bacterial Variant Calling from NGS Reads*. GitHub repository.

Snitkin, Evan S., Adrian M. Zelazny, Pamela J. Thomas, et al. 2012. “Tracking a Hospital Outbreak of Carbapenem-Resistant Klebsiella Pneumoniae with Whole-Genome Sequencing.” *Science Translational Medicine* 4 (148): 148ra116. https://doi.org/10.1126/scitranslmed.3004129.

Yoshimura, Dai, Rei Kajitani, Yasuhiro Gotoh, et al. 2019. “Evaluation of SNP Calling Methods for Closely Related Bacterial Isolates and a Novel High-Accuracy Pipeline: BactSNP.” *Microbial Genomics* 5 (5): 000261. https://doi.org/10.1099/mgen.0.000261.

Yue, Jia-Xing, and Gianni Liti. 2019. “simuG: A General-Purpose Genome Simulator.” *Bioinformatics* 35 (21): 4442–44. https://doi.org/10.1093/bioinformatics/btz424.

Zhao, Shijie, Tami D. Lieberman, Mathilde Poyet, et al. 2019. “Adaptive Evolution Within Gut Microbiomes of Healthy People.” *Cell Host Microbe* 25 (5): 656–667.e8. https://doi.org/10.1016/j.chom.2019.03.007.

969 Zojer, Markus, Lisa N. Schuster, Frederik Schulz, Alexander Pfundner, Matthias
970 Horn, and Thomas Rattei. 2017. “Variant Profiling of Evolving Prokaryotic
971 Populations.” *PeerJ* 5: e2997. https://doi.org/10.7717/peerj.2997.