Author/s:
Wick, RR;Judd, LM;Stinear, TP;Monk, IR

Title:
Are reads required? High-precision variant calling from bacterial genome assemblies

Date:
2025-05-01

Citation:
Wick, R. R., Judd, L. M., Stinear, T. P. & Monk, I. R. (2025). Are reads required? High-precision variant calling from bacterial genome assemblies. Access Microbiology, 7 (5), pp.001025.v3-. https://doi.org/10.1099/acmi.0.001025.v3.

Persistent Link:
https://hdl.handle.net/11343/360795

ACCESS MICROBIOLOGY
an open research platform

RESEARCH ARTICLE
Wick *et al.*, *Access Microbiology* 2025;7:001025.v3
DOI 10.1099/acmi.0.001025.v3

MICROBIOLOGY SOCIETY
OPEN DATA OPEN ACCESS

# Are reads required? High-precision variant calling from bacterial genome assemblies

Ryan R. Wick[1,2,*], Louise M. Judd[1,2], Timothy P. Stinear[1,2] and Ian R. Monk[1]

**Abstract**

Accurate nucleotide variant calling is essential in microbial genomics, particularly for outbreak tracking and phylogenetics. This study evaluates variant calls derived from genome assemblies compared to traditional read-based variant-calling methods, using seven closely related *Staphylococcus aureus* isolates sequenced on Illumina and Oxford Nanopore Technologies platforms. By benchmarking multiple assembly and variant-calling pipelines against a ground truth dataset, we found that read-based methods consistently achieved high accuracy. Assembly-based approaches performed well in some cases but were highly dependent on assembly quality, as errors in the assembly led to false-positive variant calls. These findings underscore the need for improved assembly techniques before the potential benefits of assembly-based variant calling (such as reduced computational requirements and simpler data management) can be realized.

**Impact Statement**

Variant calling is foundational to microbial genomics, yet traditional workflows rely heavily on sequencing reads, which, for a typical bacterial genome, can be hundreds of megabytes. In contrast, genome assemblies are far smaller (usually just a few megabytes) making them significantly easier to manage. If accurate variant calls could be made directly from assemblies, this would reduce computational demands and, in some cases, may even eliminate the need to retain raw sequencing reads. This study addresses the key question of whether variant calling from assemblies is accurate enough to replace read-based methods. Our findings show that whilst assembly-based variant calling can achieve high accuracy, this is only possible with error-free assemblies. Since most assemblies contain errors, assembly-based variant-calling approaches should currently be used with caution. Nevertheless, as sequencing and assembly technologies continue to advance, improved assembly accuracy may make assembly-based variant calling a viable alternative, reducing data complexity and storage demands whilst streamlining microbial genomic analyses.

## DATA SUMMARY

Supplementary methods, data, figures and tables are available at github.com/rrwick/Are-reads-required, which is also archived on Zenodo (10.5281/zenodo.14868870).

BioProject: PRJNA1193226.

*Access Microbiology* is an open research platform. Pre-prints, peer review reports, and editorial decisions can be found with the online version of this article.
Received 24 March 2025; Accepted 09 May 2025; Published 28 May 2025
**Author affiliations:** [1]Department of Microbiology and Immunology, The University of Melbourne at the Peter Doherty Institute for Infection and Immunity, Melbourne, Victoria, Australia; [2]Centre for Pathogen Genomics, The University of Melbourne, Parkville, Victoria, Australia.
***Correspondence:** Ryan R. Wick, ryan.wick@unimelb.edu.au;rrwick@gmail.com
**Keywords:** genome assembly; microbial genomics; *Staphylococcus aureus*; variant calling.
**Abbreviations:** FNs, false negatives; FPs, false positives; indel, insertion or deletion; ONT, Oxford Nanopore Technologies; SNP, single nucleotide polymorphism; TPs, true positives; VCF, variant call format; WT, wild type.
Three supplementary figures and seven supplementary tables are available with the online version of this article.

BioSample: SAMN45134309, SAMN45134310, SAMN45134311, SAMN45134312, SAMN45134313, SAMN45134314, SAMN45134315.

SRA reads: SRR31579284, SRR31579285, SRR31579286, SRR31579287, SRR31579288, SRR31579289, SRR31579290, SRR31579291, SRR31579292, SRR31579293, SRR31579294, SRR31579295, SRR31579296, SRR31579297.

## INTRODUCTION

Accurate identification of genetic variants is crucial in microbial genomics, particularly in applications such as building phylogenies, tracking outbreaks and studying evolutionary processes [1, 2]. SNPs are often the primary focus of variant-calling efforts, but other variants, such as insertions and deletions (indels), can also provide valuable information [3]. High-precision variant calls (i.e. with few to no false positives) are especially important for closely related samples, where even small differences, such as ten SNPs, can determine whether or not bacterial isolates are considered part of the same outbreak [4].

In many ways, genome assemblies offer significant advantages over whole-genome sequencing reads: they have smaller file sizes, are easier to manage and are conceptually simpler. Long-read sequencing, such as from Oxford Nanopore Technologies (ONT) platforms, can usually enable complete genome assemblies, where each genomic element is assembled into a single contiguous sequence, revealing the large-scale structure of the genome [5]. Given these benefits, it would be convenient to extract variants directly from assemblies, removing the need to store and process reads. However, the question remains: can assembly-based variant calling achieve the level of accuracy required for microbial analyses?

To explore this question, we investigated the accuracy of variant calls from both read-based and assembly-based methods from a series of *Staphylococcus aureus* suppressor mutants. These isolates were sequenced using both Illumina short reads and ONT long reads. By comparing multiple variant-calling approaches, we aimed to determine whether read-based methods are necessary for precise microbial variant detection, or whether assembly-based approaches can provide similarly accurate results.

## METHODS

### Isolates and mutant generation

In this study, we used seven closely related bacterial isolates. The parental strain NRS384 is a USA300 community-associated methicillin-resistant clone of *S. aureus*. A targeted mutant, WalK$^{T389A}$, was constructed by altering the essential histidine kinase WalK to inactivate its phosphatase activity [6]. Suppressor mutants arose at high frequency during aerobic growth of WalK$^{T389A}$ and were distinguished by phenotypic analysis on sheep blood agar and *Micrococcus luteus* agar that showed changes in haemolytic activity, Atl activity and/or pigmentation (Fig. S1, available in the online Supplementary Material). Suppressors (sample names beginning with IMAL) were single-colony purified and stored at −80 °C.

### Sequencing

Genomic DNA was extracted from 1 ml of overnight culture (DNeasy Blood and Tissue Kit, Qiagen) pre-treated with 100 µg of lysostaphin (Sigma-Aldrich, L7386) [6]. ONT sequencing was performed on an R10.4.1 MinION flow cell using the native barcoding kit (SQK-NBD114-96) and then basecalled using Dorado v0.7.3 using the sup@v5.0.0 model. ONT reads were quality-controlled by discarding any read with an average qscore (as reported by Dorado) of less than 12.5 or a length of less than 1 kbp. Illumina sequencing was performed on a NextSeq 2000 using the NextSeq 1000/2000 P2 Reagents v3 kit (300 Cycles; Illumina, 20046813) to generate 150 bp paired-end reads. Illumina reads were quality-controlled using fastp [7] v0.23.4 with default parameters. See Table S1 for read details and accessions.

### Reference assemblies

For each of the seven genomes, we generated carefully curated ground-truth assemblies using Trycycler [5] v0.5.5, Polypolish v0.6.0 and Pypolca [8] v0.3.1, and each assembly was reorientated to a consistent starting position using Dnaapler [9] v0.8.1. To check for potential errors, we ran Clair3 [10] v1.0.10 (using ONT reads), Sniffles2 [11] v2.4 (using ONT reads) and freebayes [12] v1.3.8 (using Illumina reads) to screen for variants (i.e. potential errors) in the assemblies, each of which found no discrepancies. Each assembly contained a 2.88 Mbp chromosome (which differed slightly across the seven genomes) and two small plasmids (4,439 bp and 3,125 bp, which were identical across all seven genomes).

### Read-based variant calling

For each isolate, we used Rasusa [13] v2.1.0 to randomly subsample both the ONT and Illumina read sets to 20× (low), 50× (medium) and 100× (high) depths, calculated as total read bases divided by the genome size. Each subsampled read set was then used to call variants against the reference assembly (WT NRS384), using Clair3 v1.0.10 for ONT reads and freebayes v1.3.8 for

Illumina reads. For Clair3, variants were kept that had a filter status of 'PASS' [14]. For freebayes, variants were kept that had a quality score ≥100 (the threshold used by Snippy [15]). See supplementary methods for the exact commands used.

### Assembly-based variant calling

Each subsampled read set was then assembled using a variety of methods. Shovill v1.1.0, Unicycler [16] v0.5.1 and SKESA [17] v2.5.1 were used for short-read-only assemblies (both Shovill and Unicycler serving as wrappers for SPAdes [18] v4.0.0). Canu [19] v2.2, Flye [20] v2.9.5 and Raven [21] v1.8.3 were used for long-read-only assemblies. Unicycler v0.5.1 and Hybracter [22] v0.9.0 were used for hybrid (both short- and long-read) assemblies. Long-read-only assemblies were additionally polished using Medaka [23] v2.0.0 using its bacterial model.

Variants were called from each assembly (all assemblies of subsampled reads as well as our ground-truth assemblies) using three different methods (Fig. 1). The first method, which we abbreviate as 'MUMmer', uses the dnadiff tool from MUMmer [24] v4.0.0rc1 to directly align the assembly to the reference genome (WT NRS384) and identify differences, which were then converted to VCF using all2vcf [25] v0.7.8. The second method, which we abbreviate as 'Shred', uses wgsim to generate synthetic reads (150 bp, error-free, 100× depth), aligns them to the reference with BWA MEM [26] v0.7.18 and then calls variants using freebayes v1.3.8 (with the same ≥100 quality threshold used in our read-based variant calls). The third method uses the ska map command from SKA2 [27] v0.3.11 to find variants against the reference. Each of these methods is bundled into an easy-to-run script available in supplementary methods.

### Variant call assessment

In total, we produced 756 VCF files: 42 from read-based variant calling (2 read types × 3 depths × 7 isolates), 21 from ground-truth assemblies (3 variant-calling methods × 7 isolates), 189 from short-read assemblies (3 assembly methods × 3 depths × 3 variant-calling methods × 7 isolates), 378 from long-read assemblies [3 assembly methods × 3 depths × 3 variant-calling methods × 2 polishing methods (no polishing and Medaka) × 7 isolates] and 126 from hybrid assemblies (2 assembly methods × 3 depths × 3 variant-calling methods × 7 isolates).

To establish a truth set of variants, we built a whole-genome multiple sequence alignment using each of the variant-calling methods by applying the variants to the reference genome (WT NRS384) using BCFtools [28] v1.21 and Trycycler v0.5.5. The following methods all produced the same results: Illumina-read freebayes (all read depths), ONT-read Clair3 (100× depth), ground-truth assembly Shred and MUMmer and Hybracter assembly Shred and MUMmer (all read depths). Since all of these methods produced the same variants, we considered these to be the ground-truth variants against which all methods were then assessed (one ground-truth VCF for each of the seven isolates).

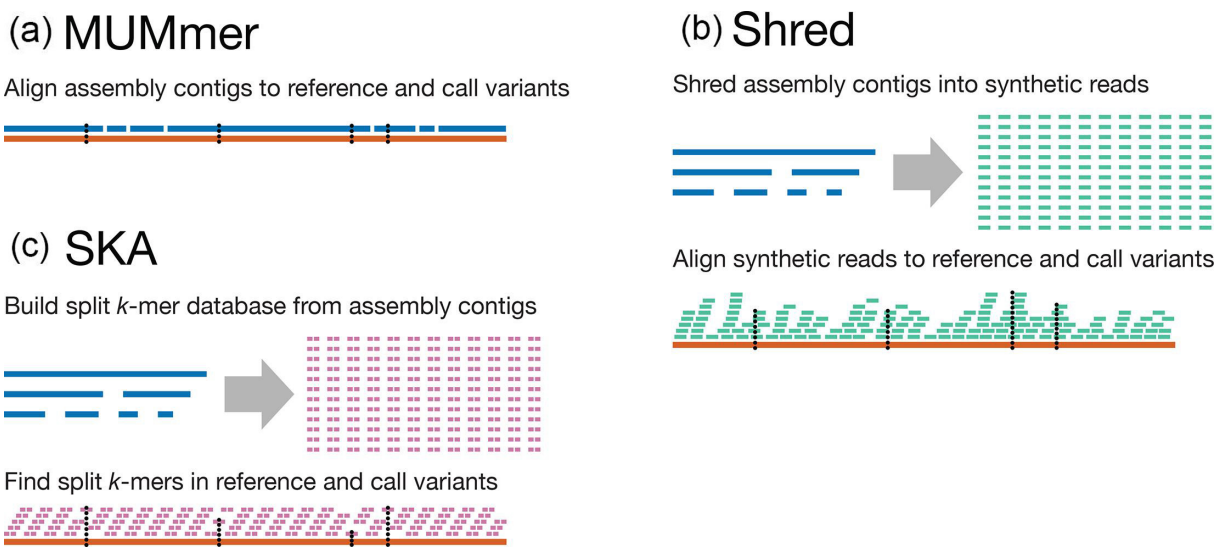


**Fig. 1.** Methods for calling variants from assemblies. (a) The MUMmer method relies on directly aligning the assembly's contigs to the reference genome, with variants called from these alignments. (b) The Shred method produces synthetic error-free reads from the assembly's contigs and aligns them to the reference genome, allowing for the use of read-based variant callers such as freebayes. (c) The SKA method uses split *k*-mers to efficiently find SNPs between the assembly's contigs and the reference genome.

To quantify the accuracy of the variant calls, we used vcfdist [29] v2.5.3 to compare each VCF file to its corresponding ground-truth VCF file. The following classification metrics were counted for both SNPs and indels and stored in Table S2: true positives (TPs), false negatives (FNs) and false positives (FPs). This allowed for the calculation of sensitivity ($\frac{TP}{TP+FN}$, the probability that a true variant will be called) and precision ($\frac{TP}{TP+FP}$, the probability that a called variant is true) (Table S3). To investigate potential causes of false-positive variant calls, we annotated the genome with repeat regions (via minimap2 self-alignment) and low-complexity regions (via Dustmasker [30] and tantan [31]) and assessed the position of each variant call relative to these features (Tables S4–S6).

## RESULTS

### Ground-truth variants

Totalled across all seven isolates, there were 23 ground-truth SNPs and 2 ground-truth indels (Fig. 2a). As expected, the two SNPs introduced to create the WalK$^{T389A}$ mutation (position 26287 A>G and 26289 A>T) were included in the ground-truth variants.

### Assembly-based variant-calling methods

We first assessed which of the three methods for assembly-based variant calling performed best, with Table 1 showing SNP and indel metrics based on all 714 assembly-based VCF files. Both MUMmer and Shred had perfect sensitivity, i.e. they never failed to call an existing SNP or indel. SKA failed to call SNPs in close proximity to other variants and could not call indels, both limitations of its split-*k*-mer-based method.

Table S3 shows summary statistics separated by assembly type: short-read-based (Shovill, SKESA and Unicycler) vs long-read-based (Canu, Flye, Raven and Hybracter). Sensitivity was consistent across variant-calling methods and assembly types. For all assembly-based variant calling methods, SNP precision was higher for long-read-based assemblies than short-read-based assemblies. Indel precision showed more variation: for Shred, short-read-based assemblies had much better indel precision, whilst for MUMmer, long-read-based assemblies were slightly better.

A large proportion of FPs were in repeat regions, particularly for certain assembly methods (Table S6). For example, over 75% of FPs in Unicycler-hybrid assemblies fell within repeat regions, which make up only 8.1% of the genome. Low-complexity regions were also enriched for FPs in some cases, though to a lesser extent than repeats.

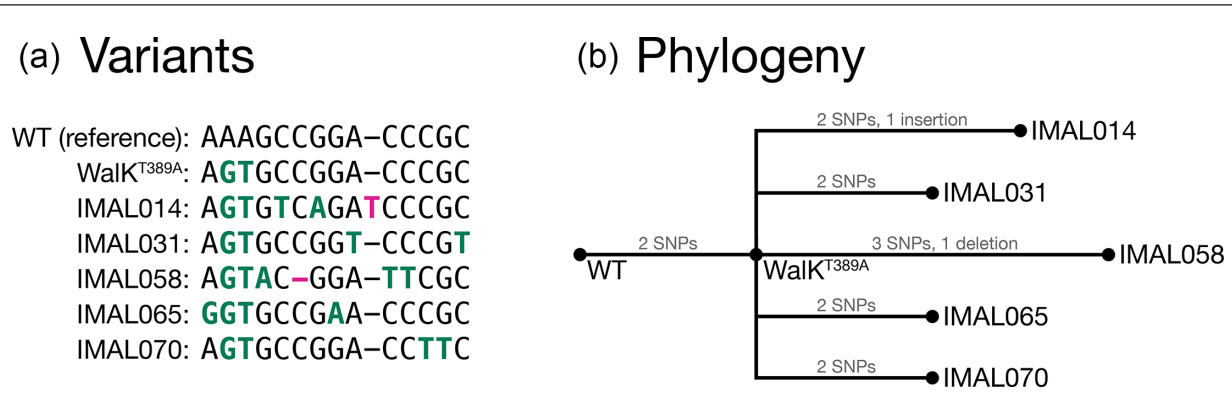


**Fig. 2.** The relationship between the seven *S. aureus* NRS384 isolates used in this study. (a) An invariant-free multiple sequence alignment, showing all variants between the isolates and the WT reference: 23 SNPs (green) and 2 indels (pink). (b) The phylogenetic relationship between the isolates, as constructed by [1] IQ-TREE v2.3.6 (using the --polytomy option and rooted on the WT reference).

**Table 1.** TP, FN and FP variant calls for each of the assembly-based variant-calling methods, for both SNPs and indels, along with sensitivity (Sens) and precision (Prec). Summary statistics separated by assembly type are provided in Table S3

| | SNPs | | | | | Indels | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | **TP** | **FN** | **FP** | **Sens** | **Prec** | **TP** | **FN** | **FP** | **Sens** | **Prec** |
| **MUMmer** | 782 | 0 | 4,669 | 1.0 | 0.14 | 68 | 0 | 719 | 1.0 | 0.086 |
| **Shred** | 782 | 0 | 4,001 | 1.0 | 0.16 | 68 | 0 | 699 | 1.0 | 0.089 |
| **SKA** | 374 | 408 | 7,431 | 0.48 | 0.048 | 0 | 68 | 0 | 0.0 | NA |

Overall, MUMmer and Shred performed similarly, but Shred produced slightly fewer FPs for both SNPs and indels. We therefore chose to only use Shred-based variant calls, discarding MUMmer- and SKA-based variant calls for the subsequent analyses.

### Read- and assembly-based variant calling

Using only the best-performing assembly-based variant-calling method (Shred), we then compared read-based and assembly-based variant calls (Fig. 3). Overall, the best results came from Illumina-read variant calls (freebayes) and Hybracter-assembly variant calls, both of which were error-free. ONT-read variant calls (Clair3) contained no FPs but did miss some SNPs at the 20× and 50× read depths (Fig. S2). All other methods were prone to false-positive calls for both SNPs and indels.

Short-read assemblies were particularly prone to SNP errors, whilst long-read assemblies were more prone to indel errors. SKESA was the best-performing short-read assembler, likely due to its conservative heuristics (it prefers breaking contigs over assembling uncertain sequences). Canu was the best-performing long-read assembler, though Flye did nearly as well. For hybrid assemblers, there was a marked difference between Unicycler (which uses a short-read-first assembly method) and Hybracter (which uses a long-read-first assembly method), the latter performing much better for both SNPs and indels (Fig. 3).

### Medaka polishing

For each of the long-read assemblies, we performed polishing with Medaka in an attempt to improve the sequence accuracy. Surprisingly, this caused a large increase in the total number of variant-calling errors (Fig. S3). Further investigation revealed that these Medaka-introduced errors were primarily caused by two factors. The first was caused by an 850 bp region of homologous sequence shared between the *S. aureus* chromosome and the 4.4 kbp plasmid. The inability to assemble small plasmids is a common error in long-read assemblies [32], leading to the absence of the 4.4 kbp plasmid in some assemblies. When this occurred, Medaka's alignment of reads to the assembly resulted in plasmid reads erroneously aligning to the chromosome, causing errors in the consensus algorithm.

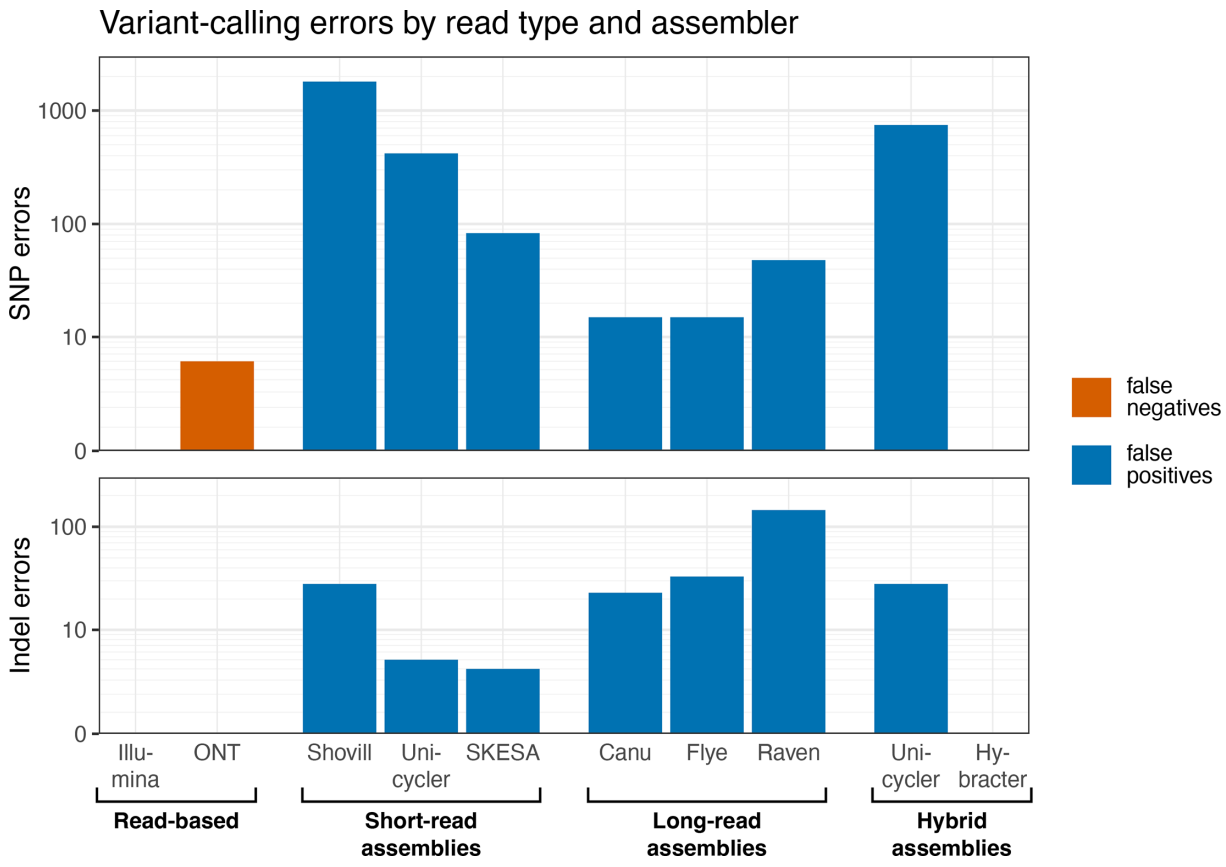


**Fig. 3.** Variant-calling errors for both read-based and assembly-based variant-calling methods. For each method, the errors were summed across all genomes and read depths. See Fig. S2 for the results for each depth separately and for sensitivity and precision values. The y-axes have a pseudo-log transformation.

The second factor related to increasing Medaka errors was when the assembly contained a duplicated sequence from the genome. This occurred when the start and end of a circular contig overlapped and when a region of the genome was assembled into both a primary and an alternative sequence. When these issues occurred, Medaka was prone to introducing errors in the duplicated regions, leading to false-positive assembly-based variant calls. Canu was prone to both start–end overlaps (up to 72 kbp) and alternative-sequence contigs (what it calls 'bubbles', up to 95 kbp), leading to the largest increase in variant-calling errors after Medaka polishing.

In assemblies where neither of these factors was present, we found that Medaka often improved the accuracy of assembly-based variant calls (Table S7). This suggests that Medaka-polishing can be beneficial for variant calling when the assembly is correctly finished and contains the entire genome with no duplicated sequence [33].

## DISCUSSION

In this study, we examined variant calling in a group of closely related *S. aureus* isolates, which only had 25 combined variants across six mutant strains relative to the WT strain. We found that it was possible to obtain precise variant calls from genome assemblies (i.e. without directly using the reads), but this required very high-quality assemblies: only our ground-truth assemblies (Trycycler with manual curation) and Hybracter assemblies (a long-read-first hybrid assembly pipeline) yielded perfect variant calls. All other assembly methods resulted in false-positive calls, sometimes exceeding 100 errors per genome, leading to poor precision. This suggests that most genome assemblies are not suitable for variant calling if high precision is required.

In general, we found that short-read assemblies were more prone to false-positive SNP calls, whilst ONT assemblies were more prone to false-positive indel calls. Our ONT assemblies were based on the latest ONT chemistry and basecalling model, and it is likely that older ONT assemblies will have reduced precision for both SNPs and indels. Whilst we expected Medaka polishing to improve assembly-based variant calls for ONT assemblies, missing or duplicated sequences in the assemblies often led to Medaka introducing errors. This emphasizes the need for an assembly to be structurally correct before Medaka polishing.

Whilst we compared three different methods for assembly-based variant calling (MUMmer, Shred and SKA), we did not attempt to optimize any of these methods. This is a key limitation of this study, and so, higher precision assembly-based variant-calling methods are likely possible. Such optimizations might include masking parts of the reference genome (e.g. repeats and low-complexity regions; see Table S6) or excluding parts of the assembly (e.g. short contigs and low-quality bases [34]). If variant calling from assemblies is necessary (e.g. because reads are not available), then further work is required to determine the optimal method. Since SKA's split-*k*-mer approach cannot identify indels or closely spaced SNPs, the optimal assembly-based variant-calling method will likely involve alignment.

The *S. aureus* genomes used in this study were relatively easy to assemble – our Illumina assemblies often contained fewer than 100 contigs (Table S2), and error-free ONT assemblies were possible (Table S7). More challenging genomes (e.g. containing more repetitive regions) would likely make assembly-based variant calling even more challenging. Of the 30 samples assembled in the Hybracter manuscript [22], 12 had zero errors in their assembly, whilst 18 contained errors. This suggests that even when performing a robust Hybracter hybrid assembly, false-positive assembly-based variant calls may occur for approximately half of bacterial genomes. Also, the *S. aureus* genomes in this study contained no structural differences relative to the reference genome. When structural differences occur, there is the possibility of more variant-calling errors for both read- and assembly-based approaches.

In conclusion, we found that assembly-based variant calling is prone to false-positive errors, so traditional read-based variant calling is preferred when possible. However, error-free assemblies can produce error-free variant calls, so as sequencing and assembly methods become more reliable in the future, assembly-based variant calls may become more feasible.

Funding information
R.R.W. is supported by an ARC Discovery Early Career Researcher Award (DE250100677). T.P.S. is supported by an NHMRC Research Fellowship (APP1105525) and an ARC Discovery Project (DP240102465).

Acknowledgements
This research was performed in part at the Centre for Pathogen Genomics Innovation Hub, Department of Microbiology and Immunology, University of Melbourne, at the Peter Doherty Institute for Infection and Immunity.

Author contributions
Conceptualization: R.R.W., L.M.J. and I.R.M. Data curation: R.R.W. and L.M.J. Formal analysis: R.R.W. Funding acquisition: R.R.W., I.R.M. and T.P.S. Investigation: R.R.W., L.M.J. and I.R.M. Methodology: R.R.W., L.M.J. and I.R.M. Project administration: I.R.M. and T.P.S. Software: R.R.W. Resources: L.M.J. and I.R.M. Supervision: I.R.M. and T.P.S. Visualization: R.R.W. Writing – original draft: R.R.W. Writing – review and editing: R.R.W., L.M.J., T.P.S. and I.R.M.

Conflicts of interest
The authors declare that there are no conflicts of interest.

Ethical statement
This study did not involve human participants, animal subjects or clinical data. Therefore, no ethical approval was required.

References


1. Minh BQ, Schmidt HA, Chernomor O, Schrempf D, Woodhams MD, *et al*. IQ-TREE 2: new models and efficient methods for phylogenetic inference in the genomic era. *Mol Biol Evol* 2020;37:1530–1534.
2. Gorrie CL, Da Silva AG, Ingle DJ, Higgs C, Seemann T, *et al*. Key parameters for genomics-based real-time detection and tracking of multidrug-resistant bacteria: a systematic analysis. *Lancet Microbe* 2021;2:e575–e583.
3. Redelings BD, Suchard MA. Incorporating indel information into phylogeny estimation for rapidly emerging pathogens. *BMC Evol Biol* 2007;7:40.
4. Gorrie CL, Mirčeta M, Wick RR, Edwards DJ, Thomson NR, *et al*. Gastrointestinal carriage is a major reservoir of *Klebsiella pneumoniae* infection in intensive care patients. *Clin Infect Dis* 2017;65:208–215.
5. Wick RR, Judd LM, Cerdeira LT, Hawkey J, Méric G, *et al*. Trycycler: consensus long-read assemblies for bacterial genomes. *Genome Biol* 2021;22:266.
6. Sharkey LKR, Guerillot R, Walsh CJ, Turner AM, Lee JYH, *et al*. The two-component system WalKR provides an essential link between cell wall homeostasis and DNA replication in *Staphylococcus aureus*. *mBio* 2023;14:e0226223.
7. Chen S, Zhou Y, Chen Y, Gu J. fastp: an ultra-fast all-in-one FASTQ preprocessor. *Bioinformatics* 2018;34:i884–i890.
8. Bouras G, Judd LM, Edwards RA, Vreugde S, Stinear TP, *et al*. How low can you go? short-read polishing of oxford nanopore bacterial genome assemblies. *Microb Genom* 2024;10:001254.
9. Bouras G, Grigson SR, Papudeshi B, Mallawaarachchi V, Roach MJ. Dnaapler: a tool to reorient circular microbial genomes. *JOSS* 2024;9:5968.
10. Zheng Z, Li S, Su J, Leung AWS, Lam TW, *et al*. Symphonizing pileup and full-alignment for deep learning-based long-read variant calling. *Nat Comput Sci* 2022;2:797–803.
11. Smolka M, Paulin LF, Grochowski CM, Horner DW, Mahmoud M, *et al*. Detection of mosaic and population-level structural variants with Sniffles2. *Nat Biotechnol* 2024;42:1571–1580.
12. Garrison E, Marth G. Haplotype-based variant detection from short-read sequencing. *arXiv* 2012.
13. Hall M. Rasusa: randomly subsample sequencing reads to a specified coverage. *JOSS* 2022;7:3941.
14. Hall MB, Wick RR, Judd LM, Nguyen AN, Steinig EJ, *et al*. Benchmarking reveals superiority of deep learning variant callers on bacterial nanopore sequence data. *Elife* 2024;13:RP98300.
15. Seemann T. Snippy [Internet]; 2020. https://github.com/tseemann/snippy
16. Wick RR, Judd LM, Gorrie CL, Holt KE. Unicycler: resolving bacterial genome assemblies from short and long sequencing reads. *PLoS Comput Biol* 2017;13:e1005595.
17. Souvorov A, Agarwala R, Lipman DJ. SKESA: strategic *k*-mer extension for scrupulous assemblies. *Genome Biol* 2018;19:153.
18. Prjibelski A, Antipov D, Meleshko D, Lapidus A, Korobeynikov A. Using SPAdes de novo assembler. *Curr Protoc Bioinform* 2020;70:e102.
19. Koren S, Walenz BP, Berlin K, Miller JR, Bergman NH, *et al*. Canu: scalable and accurate long-read assembly via adaptive *k*-mer weighting and repeat separation. *Genome Res* 2017;27:722–736.
20. Kolmogorov M, Yuan J, Lin Y, Pevzner PA. Assembly of long, error-prone reads using repeat graphs. *Nat Biotechnol* 2019;37:540–546.
21. Vaser R, Šikić M. Time- and memory-efficient genome assembly with Raven. *Nat Comput Sci* 2021;1:332–336.
22. Bouras G, Houtak G, Wick RR, Mallawaarachchi V, Roach MJ, *et al*. Hybracter: enabling scalable, automated, complete and accurate bacterial genome assemblies. *Microb Genom* 2024;10:001244.
23. Wright C, Wykes M. Medaka [Internet]; 2022. https://github.com/nanoporetech/medaka
24. Marçais G, Delcher AL, Phillippy AM, Coston R, Salzberg SL, *et al*. MUMmer4: a fast and versatile genome alignment system. *PLoS Comput Biol* 2018;14:e1005944.
25. Schiavinato M. all2vcf [Internet]; 2024. https://github.com/rrwick/all2vcf
26. Li H. Aligning sequence reads, clone sequences and assembly contigs with BWA-MEM. *arXiv* 2013.
27. Derelle R, von Wachsmann J, Mäklin T, Hellewell J, Russell T, *et al*. Seamless, rapid, and accurate analyses of outbreak genomic data using split *k*-mer analysis. *Genome Res* 2024;34:1661–1673.
28. Danecek P, Bonfield JK, Liddle J, Marshall J, Ohan V, *et al*. Twelve years of SAMtools and BCFtools. *Gigascience* 2021;10:giab008.
29. Dunn T, Narayanasamy S. vcfdist: accurately benchmarking phased small variant calls in human genomes. *Nat Commun* 2023;14:8149.
30. Morgulis A, Gertz EM, Schäffer AA, Agarwala R. A fast and symmetric DUST implementation to mask low-complexity DNA sequences. *J Comput Biol* 2006;13:1028–1040.
31. Frith MC. A new repeat-masking method enables specific detection of homologous sequences. *Nucleic Acids Res* 2011;39:e23.
32. Wick RR, Judd LM, Wyres KL, Holt KE. Recovery of small plasmid sequences via oxford nanopore sequencing. *Microb Genom* 2021;7:000631.
33. Wick RR. Medaka v2: progress and potential pitfalls [Internet]; 2024. https://rrwick.github.io/2024/10/17/medaka-v2.html
34. Wick RR. FASTQ assemblies with dorado polish [Internet]; 2025. https://rrwick.github.io/2025/02/19/fastq-assemblies.html