Alternative Base Callers Aid Real-Time Analysis of SARS-CoV-2 Sequencing Runs

Vladimír Boža[1], Matej Fedor[1], Kristína Boršová[2,3], Viktória Čabanová[3], Jana Černíková[1], Viktória Hodorová[2], Peter Perešíni[1], Klára Sládečková[1], Boris Klempa[3], Jozef Nosek[2], Broňa Brejová[1], Tomáš Vinař[1]

[1] Faculty of Mathematics, Physics and Informatics, Comenius University, Bratislava, Slovakia
[2] Faculty of Natural Sciences, Comenius University, Bratislava, Slovakia
[3] Biomedical Research Center of the Slovak Academy of Sciences, Bratislava, Slovakia

Abstract: One of the advantages of nanopore sequencing is its ability to provide data in real time, which allows monitoring, early stopping, and fast identification of mutations in sequenced material. Nanopore sequencer measures electrical current induced by the DNA passing through a pore and this signal needs to be translated to a string over the alphabet {A,C,G,T} through a process called base calling. To achieve base calling in real time, the mainstream tools (such as Guppy provided by Oxford Nanopore Technologies) require the support of high-performance GPUs. This is prohibitive in many settings. Here, we evaluate the accuracy of several alternative base callers, which only require use of a desktop CPU or a support of low-cost USB-connected accelerator. While their accuracy is, in general, lower than that of Guppy in a high-accuracy mode using GPUs, we show that these alternative base callers can act as a replacement for monitoring and mutation detection in SARS-CoV-2 sequencing runs, without sacrificing the accuracy of the final result.

Availability: `http://compbio.fmph.uniba.sk/sars-cov-2-sequencing/`

1 Introduction

The ARTIC protocol has originally been developed for sequencing viral genomes with nanopore sequencing devices (Quick et al., 2016), and it has become a commonly used protocol for SARS-CoV-2 sequencing (Tyson et al., 2020). Briefly, overlapping segments of the viral genome are first amplified using PCR, and the resulting amplicons are sequenced using nanopore sequencing (see a simplified illustration in Figure 1). Typically, multiple samples are sequenced in parallel using barcoding. In bioinformatics post processing, the individual reads are first assigned to individual samples, using demultiplexing according to the barcodes. Stricter parameters (requiring the presence of barcodes on both ends of the read) are typically used in order to avoid barcode bleeding and to discard partially sequenced reads. The reads are then aligned to the reference genome and mutations are discovered with the aid of the raw sequencing signal using nanopolish (Loman et al., 2015).

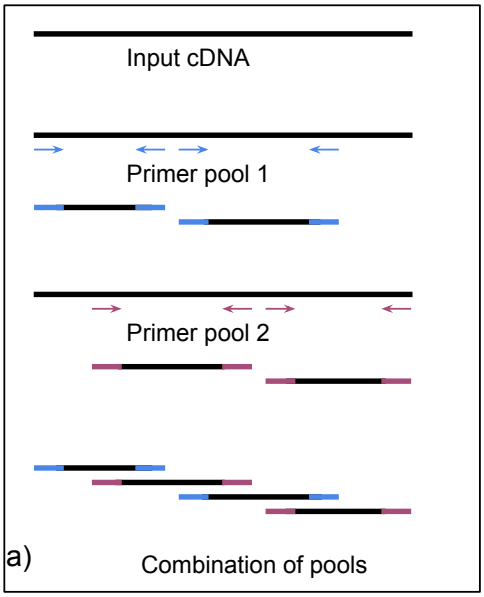
One of the problems with this protocol is that the PCR amplification step introduces wide variation in coverage, both between samples and between different amplicons within a sample. Due to the high error rate of nanopore sequencing, it is not advisable to determine mutations in regions with low coverage (the standard pipeline set the coverage threshold at 20). In such scenarios, it is difficult to estimate when to stop data acquisition. Fortunately, results of nanopore sequencing can be processed in real-time and on-the-fly monitoring during sequencing helps to inform decisions on when to stop the run.

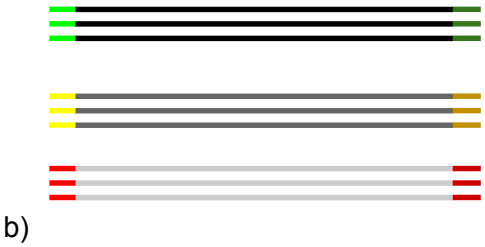
A nanopore sequencer reads an electrical signal induced by the DNA passing through a pore and before subsequent analysis, this signal needs to be translated to DNA bases via base calling. A base caller provided by manufacturer (called Guppy), requires a machine with a high performance GPU, which is not available in many laptop computers and is also problem in desktops due to current NVIDIA GPU shortages.

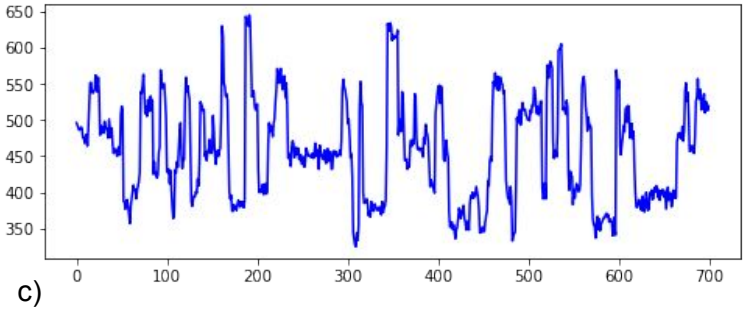
In this work, we propose to use alternative base callers with lower demands on computational resources, albeit producing reads with a slightly lower accuracy (Boža et al., 2020; Perešíni et al., 2020; Boža et al., 2021). We demonstrate that using our alternative base caller not only allows monitoring, but can also produce the final sequence of similar quality as using the standard base caller. Moreover, we are able to call tentative variants during sequencing from incomplete sequence using a custom made classifier. This allows us to report important information about virus lineage determination already during the sequencing run, well before the full sequence is determined.

2 Evaluation of Alternative Base Callers

We have evaluated three alternative base callers that can achieve real-time base calling without the use of a GPU: Deepnano-blitz (Boža et al., 2020), Deepnano-Coral (Perešíni et al., 2020), and Osprey (Boža et al., 2021). There are also other alternative base callers such as Bonito (Seymour, 2020) and SACall (Huang et al., 2020), but none of them offers real-time base calling on a CPU or a low power USB-connected TPU.







AAAGTAGATGCTAAAGCTTACAAAGAAGT

GGGCCTTTTTATATATCCTACTATTGTTT

TATCTCTGCTATAGTAACCTGAAAGTCTC

AAAATTCTTTTAAGGCGGGTCATGGTAGT

TATTTATGTTCTTTTAACGTGCAACCCTC

d)

AAA GTAGATGCTAAAGCTTACAAAGA AGT
AAA ATTCTTTTAAGGCGGGTCATGGT AGT

GGG CCTTTTTATATATCCTACTATTG TTT

TAT CTCTGCTATAGTAACCTGAAAGT CTC
TAT TTATGTTCTTTTAACGTGCAACC CTC

e)

AGGTGCCACTACTTGTGGTTACTTACCCCAAAATGCTGTTGTTAAAATTTATTGTCCAGC
AGGTGCCACTACATGTGGTTACTTACCCCAAAA
GGTGCCACTACATGTGGTTACTTACCCCAAAAT
GTGCCACTACTTGTGGTTACTTACCCCAA
GGTGCCACTACATGTGGTTACTTACCCCAAAA
GTGCCACTACATGTGGTTACTTACCCCAAAA
TTACCCCAAAATGCTGTTGTT-AAATTTATTGTCCAGC
TACCCCAAAATGCTGTTGTT-AAATTTATTGTCCAG
CTTACCCCAAAATGCTGTTGTT-AAATTTATTGTCC
TTACCCCAAAATGCTGTTGTT-AAATTTATTGTCCAG
ACCCCAAAATGCTGTTGTT-AAATTTATTGTCCAGC

f)

AGGTGCCACTACATGTGGTTACTTACCCCAAAATGCTGTTGTT-AAATTTATTGTCCAGC

g)

Figure 1: A simplified illustration of the ARTIC protocol workflow. (a) First each virus sample is amplified, using target-specific primers. Two pools of primers are used to obtain overlapping amplicons. (b) Amplicons from multiple samples are tagged using barcodes and sequenced together. (c) All reads are sequenced on a MinION device which for each sequenced molecule produces an electrical signal. (d) The electrical signal is converted to the string using base calling software. (e) Barcodes at ends of reads are recognized by the demultiplexing software and individual reads are assigned to their samples. (f) Reads are mapped to the reference. (g) Mutations are called based on the consensus of multiple reads.

Deepnano-blitz (Boža et al., 2020) is a real-time CPU base caller based on recurrent neural networks. Deepnano-blitz allows adjustment of the time vs. accuracy tradeoff by changing the size of the neural network model. Smaller version (48) can run in real time on a single CPU core, larger version (96) requires multiple cores to achieve real-time performance. The accuracy of the smaller version is slightly lower than the accuracy of Guppy 4.4 in the fast mode, the larger version is comparable to Guppy 4.4 in the fast mode.

Deepnano-Coral (Perešíni et al., 2020) is a convolutional neural network base caller. It requires Coral Edge TPU, which is a sub-$100 accelerator from Google that can connect to a USB port, with very low power requirements. Deepnano-Coral is best suited for laptop computers that do not have GPU support, as well as in scenarios where power consumption may become a limiting factor (such as sequencing in the field). The accuracy of real-time base calling with Deepnano-Coral falls between Guppy 4.4 fast and high-accuracy (HAC) modes.

Osprey (Boža et al., 2021) is a CPU-based base caller that uses architecture similar to Deepnano-Coral, but is further improved by using a technique called dynamic pooling and decoding via transducers. The accuracy of real-time base calling is equivalent to Guppy 3.4 HAC and better than Guppy 4.4 fast. Computational requirements are similar to Deepnano-blitz 96.

Using faster base callers usually results in sacrificing accuracy at the individual read level. However, in case of the ARTIC pipeline, multiple reads are aligned to each region, and only differences that consistently occur in many reads are considered proper variants. Moreover, the ARTIC pipeline uses Nanopolish (Loman et al., 2015), which works directly with the raw sequencing signal, as an underlying variant caller. Therefore the base calling accuracy is not as important, since base calls are only used for demultiplexing and for the initial alignment of the read to the reference in Nanopolish.

The ARTIC pipeline sometimes calls a particular base as *unknown* (denoted as N in the sequence). This can happen for two reasons: low coverage of an amplicon or conflicting information from different sequencing reads. Assigning an unknown base represents a conservative decision and is used wherever it is impossible to decide whether a particular base is the same as the reference or represents a mutation with high enough confidence.

We have evaluated the performance of each of the above mentioned base callers in the context of the ARTIC pipeline. For the evaluation purposes, we have used a sequencing run from January 13, 2021 with 23 barcoded SARS-CoV-2 samples (the 24th sample was excluded due to very low coverage) using a MinION run with R9.4.1 flow cell, LSK109 chemistry, and 2-kbp amplicon scheme by Resende et al. (2020). In the standard software pipeline, we use Guppy 4.4 (highest version available in the time of analysis) in the high accuracy mode to base call the reads, followed by the ARTIC pipeline for variant calling. We used the results of this standard pipeline as a ground truth.

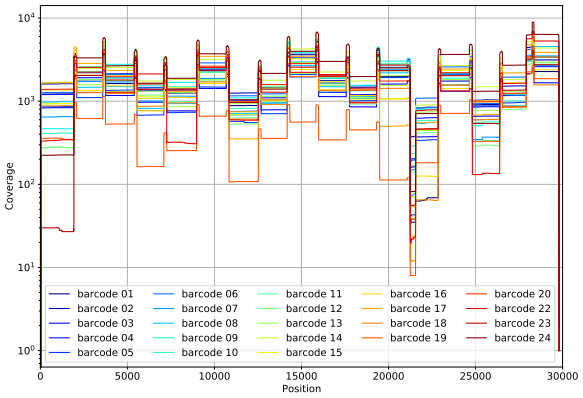


Figure 2: Variance in the coverage within and between samples.

For each of the above mentioned base callers, as well as for Guppy 3.4 in the high-accuracy mode, we reran the ARTIC pipeline with their base calls and compare the results (see Table 1). Guppy 3.4 was used as a representative base caller from a year ago. We also run all of our base callers with a lower demutiplexing threshold, which slightly increases the coverage, due to more reads being demutiplexed to individual samples.

Only very few positions (up to 2 in 23 samples) are called differently (B→B column). Even though these clearly represent erroneous base calls (see Table 2), there are so few of them that they do not impact the overall accuracy significantly. The largest problem presents an increased number of "unknown" calls (B→N column). These are mainly concentrated within a single 310bp region (21242-21551) which in several samples had an extremely low coverage (see Figure 2). With lower efficiency of demultiplexing due to base calling errors, the coverage of this region was in some samples pushed below the minimum coverage threshold of the ARTIC pipeline and consequently was masked with Ns in the result. There were several additional "unknown" calls of individual bases which were clustered around certain positions in the genome. We suspect that this is due to some biases stemming from nanopore sequencing, where variants of some bases in certain contexts are difficult to distinguish.

On the other hand, some additional bases are called compared to baseline (N→B column). In all cases, these were called as the original reference. Almost all cases were at positions 16255 and 16256 and one case was in the region 21220-21296, where coral-q50 increased the coverage over the minimum threshold.

While in some cases the use of our alternative base callers may result in an incomplete sequence (compared to the baseline), in general our results show that each of these tools is a viable alternative to the standard base calling with Guppy 4.4 in high accuracy mode with similar quality of the final sequence. While Guppy 4.4 HAC re-

Table 1: Comparison of the results of the ARTIC pipeline using different base callers. The values represent the total number and median of differences in 23 consensus sequences compared to the baseline. N→B: position marked as unknown in the baseline was resolved as a base. B→N: position resolved as a base in the baseline was marked as unknown. B→B: a different base was called. Q50: lower the required demultiplexing score from 60 to 50.

	Total 23 samples			Median 23 samples			
Base caller	N→B	B→N	B→B	N→B	B→N	B→B	Hardware to achieve real time
Guppy 3.4 HAC	2	116	0	0	0	0	High-performance GPU
Blitz48	4	1135	1	0	4	0	Single desktop CPU core
Blitz48-Q50	4	404	2	0	3	0	
Blitz96	4	273	0	0	1	0	Multi-core desktop CPU
Blitz96-Q50	4	131	0	0	2	0	
Coral	4	261	0	0	1	0	Sub-$100 Coral accelerator
Coral-Q50	80	113	0	0	1	0	
Osprey	4	253	1	0	0	0	Multi-core desktop CPU
Osprey-Q50	4	108	1	0	0	0	

Table 2: Mutations identified by various base callers in addition to the gold standard calls.

Base caller	Barcode	Position	Reference	Variant	Coverage	Notes
Blitz48	18	22009	C	CA	31	Frameshift, thus likely invalid mutation
Blitz48-Q50	18	22009	C	CA	46	
Blitz48-Q50	7	1706	TC	T	473	Frameshift, thus likely invalid mutation
Osprey	23	237	G	GT	25	Not present in GISAID before,
Osprey-Q50	23	237	G	GT	30	probably invalid

quires high performance GPU to have a reasonable running time, the alternatives only require a CPU or a sub-$100 accelerator connected through a standard USB port.

3 Determining Virus Lineages During Sequencing from Incomplete Data

One of the key tasks in analysis of sequenced SARS-CoV-2 samples is determination of the virus lineage according to the standardized lineage classification (Rambaut et al., 2020). The standard tool to accomplish this task is pangolin (O'Toole et al., 2021), which uses machine learning approach to determine the lineage from the finished sequence. Pangolin currently fits a (single) decision tree classifier to sequence data to determine the lineage. While this approach seems to have high accuracy for complete sequence data, it handles incomplete sequences by simply filling them using bases from the reference sequence. This naturally leads to unpredictable changes in classification as sequence is being completed, since each new mutation might lead to a complete change in the decision tree path.

To quickly make provisional lineage classification, even for incomplete sequences during the sequencing, we propose a simple classification scheme based on a manually curated list of characteristic mutations. We identify a list of these characteristic mutations for expected lineages of interest for a particular country at a particular time, and each lineage also has a threshold for number of mutations required to be present to make a call as shown in an example in Figure 3.

During the sequencing run, we use a fast base caller (Deepnano-blitz 48 in our experiments) to provide live base calling and by aligning individual sequencing reads to the reference sequence and simply counting the support for a mutation at a particular position, we make provisional variant calls. Note that this would be highly imprecise for insertions and deletions due to the frequent indels in nanopore sequencing reads. For this reason, we only focus on single nucleotide variants. Once the number of characteristic mutations passes the threshold for a particular sample, the lineage is provisionally called.

We have integrated our tool within the RAMPART sequencing run monitoring framework (Hadfield, 2021) and tested its performance on three runs: one run with 24 barcoded samples, and two with 96 barcoded samples each. There were no disagreements when both our tool and pangolin called the lineage, however, in certain cases one or the other tool did not make a call (see a summary of results in Table 3).

Figure 4 shows that our tool can provide early information about lineages detected in the sequencing run. Even though barcodes in our samples were highly unbalanced, some samples can be identified within minutes of start-

```
B.1.1.7 14 C3267T C5388A T6954C A23063T C23271A C23604A C23709T T24506G G24914C C27972T G28048T
          A28111G C28977T G28280C A28281T T28282A
B.1.160 5 G9526T G15766T A16889G G17019T G22992A T26876C
B.1.177 6 T445C C6286T G21255C C22227T C26801G C28932T G29645T
B.1.258 4 G12988T G15598A G18028T T24910C T26972C
B.1.221 4 C21855T A25505G G25906C C28651T C28869T
A.23.1 3 C10747T G11521T C23604G T24097C
B.1.351 6 G174T G5230T G23012A A21801C A23063T C28253T C23664T
P.1 8 T733C C2749T C3828T A5648C C12778T C13860T G17259T C21614T C21621A
P.2 7 T10667G C11824T A12964G G23012A C28253T G28628T G28975T C29754T
B.1.427/9 4 G17014T G22018T C26681T A28272T C28887T
B.1.525 8 C1498T A1807G G2659A C6285T T8593C C14407T A21717G C21762T T24224C
B.1.526 8 T9867C C25517T C27925T A20262G C21575T C21846T A22320G C23664T C28869T
B.1.617 8 G210T T22917G C23604G C25469T T27638C G28881T G29402T G29742T
R.1 8 C14340T G17551A C18877T A19167G C19274A G22017T G23012A G23868T T26604C
```

Figure 3: Mutation specification for determining virus lineages. Note that only certain lineages of interest are included. For example, to identify sample as B.1.1.7 variant, we require 14 out of 16 mutations listed.

Table 3: Comparison of our lineage identification with pangolin results. There were no disagreements (if both tools identified a lineage, the output was always the same). In a small number of cases, a lineage was identified only by one of the tools.

		Lineage identified by		
Dataset date	barcodes	both	pangolin	our tool
2021-02-03	24	22	2	0
2021-03-11	96	80	1	0
2021-03-25	96	70	1	2

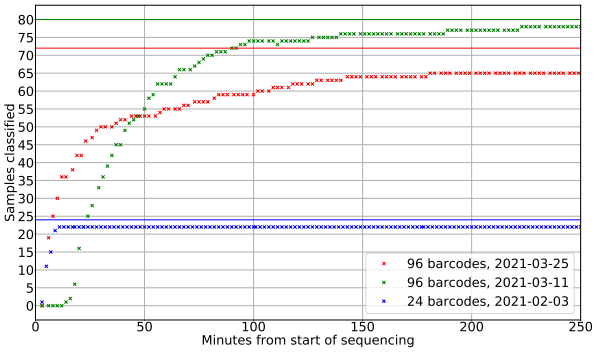


Figure 4: The number of samples with lineage classification over time. Horizontal lines show the number of classified samples at the end of the sequencing run.

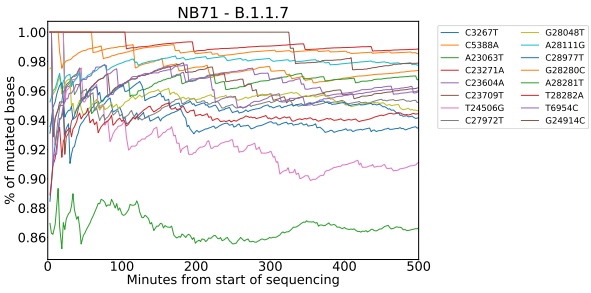


Figure 5: The percentage of overlapping reads supporting individual identified mutations over time.

ing the run, and our tool has provided accurate detection of lineages for 50% of barcodes as early as 40 minutes from the start of a 96-barcode runs. Due to the low quality of some samples, we typically run the sequencing for approximately 24 hours, so such on-the-fly analysis provides us an opportunity to report the basic information on sequenced samples to health authorities as early as one day before the final analysis is finished.

While our determination of single nucleotide sequence variants is somewhat simplistic, Figure 5 shows that on real data even such a simple method can achieve results with high confidence. In all cases, mutations were supported by over 85% of reads and there were no calls that would suffer from ambiguity.

4 Conclusions and Discussion

One of the great advantages of nanopore sequencing is the ability to analyze data as they are sequenced. Fast base callers that can replace default base callers provided by Oxford Nanopore Technologies are a key in utilizing this advantage. Here, we have evaluated fast base callers in the context of the ARTIC pipeline and determined that they can provide results with similar quality at a fraction of computational cost.

In the case of the ARTIC pipeline, the quality of base calls mainly affects the demultiplexing stage, and does not play as important role in the variant calling since this

is done with the assistance of the raw sequencing signal. Moreover, we have also demonstrated that fast base callers can be used in the context of RAMPART monitoring tool to identify virus lineages on-the-fly during the sequencing. Such application allows us to relay important information to health authorities much faster.

One of the advantages of RAMPART monitoring tool is that it can monitor in real time the coverage of all regions in all barcoded samples, allowing us to make an informed determination when to stop the sequencing run. As a future work, we would like to use a similar framework in connection with the selective sequencing (Payne et al., 2021) to achieve a more uniform coverage between samples, as well as to mitigate uneven coverage within samples stemming from varying efficiency of individual PCR primers, by rejecting reads belonging to the regions that are already well covered.

Acknowledgements. This research was supported by a grant ITMS:313011ATL7 "Pangenomics for personalized clinical management of infected persons based on identified viral genome and human exome" from the Operational Program Integrated Infrastructure (90%) co-financed by the European Regional Development Fund. The research was also supported by VEGA 1/0458/18 to TV (10%).

References

Boža, V., Perešíni, P., Brejová, B., and Vinař, T. (2020). Deepnano-blitz: a fast base caller for minion nanopore sequencers. *Bioinformatics*, 36(14):4191–4192.

Boža, V., Perešíni, P., Brejová, B., and Vinař, T. (2021). Dynamic Pooling Improves Nanopore Base Calling Accuracy. London Calling 2021 poster.

Hadfield, J. (2021). Rampart: Read assignment, mapping, and phylogenetic analysis in real time. https://github.com/artic-network/rampart.

Huang, N., Nie, F., Ni, P., Luo, F., and Wang, J. (2020). Sacall: a neural network basecaller for oxford nanopore sequencing data based on self-attention mechanism. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*.

Loman, N. J., Quick, J., and Simpson, J. T. (2015). A complete bacterial genome assembled de novo using only nanopore sequencing data. *Nat Methods*, 12(8):733–735.

O'Toole, A., Scher, E., Underwood, A., Jackson, B., Hill, V., McCrone, J., Ruis, C., Abu-Dahab, K., Taylor, B., Yeats, C., du Plessis, L., Aanensen, D., Holmes, E., Pybus, O., and Rambaut, A. (2021). pangolin: lineage assignment in an emerging pandemic as an epidemiological tool. github.com/cov-lineages/pangolin.

Payne, A., Holmes, N., Clarke, T., Munro, R., Debebe, B. J., and Loose, M. (2021). Readfish enables targeted nanopore sequencing of gigabase-sized genomes. *Nature biotechnology*, 39(4):442–450.

Perešíni, P., Boža, V., Brejová, B., and Vinař, T. (2020). Nanopore Base Calling on the Edge. Technical Report arXiV:2011.04312, arXiv.

Quick, J., Loman, N. J., Duraffour, S., Simpson, J. T., Severi, E., Cowley, L., Bore, J. A., Koundouno, R., Dudas, G., Mikhail, A., Ouedraogo, N., Afrough, B., Bah, A., Baum, J. H., Becker-Ziaja, B., Boettcher, J. P., Cabeza-Cabrerizo, M., Camino-Sanchez, A., Carter, L. L., Doerrbecker, J., Enkirch, T., Dorival, I. G. G., Hetzelt, N., Hinzmann, J., Holm, T., Kafetzopoulou, L. E., Koropogui, M., Kosgey, A., Kuisma, E., Logue, C. H., Mazzarelli, A., Meisel, S., Mertens, M., Michel, J., Ngabo, D., Nitzsche, K., Pallash, E., Patrono, L. V., Portmann, J., Repits, J. G., Rickett, N. Y., Sachse, A., Singethan, K., Vitoriano, I., Yemanaberhan, R. L., Zekeng, E. G., Trina, R., Bello, A., Sall, A. A., Faye, O., Faye, O., Magassouba, N., Williams, C. V., Amburgey, V., Winona, L., Davis, E., Gerlach, J., Washington, F., Monteil, V., Jourdain, M., Bererd, M., Camara, A., Somlare, H., Camara, A., Gerard, M., Bado, G., Baillet, B., Delaune, D., Nebie, K. Y., Diarra, A., Savane, Y., Pallawo, R. B., Gutierrez, G. J., Milhano, N., Roger, I., Williams, C. J., Yattara, F., Lewandowski, K., Taylor, J., Rachwal, P., Turner, D., Pollakis, G., Hiscox, J. A., Matthews, D. A., O'Shea, M. K., Johnston, A. M., Wilson, D., Hutley, E., Smit, E., Di Caro, A., Woelfel, R., Stoecker, K., Fleischmann, E., Gabriel, M., Weller, S. A., Koivogui, L., Diallo, B., Keita, S., Rambaut, A., Formenty, P., Gunther, S., and Carroll, M. W. (2016). Real-time, portable genome sequencing for Ebola surveillance. *Nature*, 530(7589):228–232.

Rambaut, A., Holmes, E. C., O'Toole, A., Hill, V., McCrone, J. T., Ruis, C., du Plessis, L., and Pybus, O. G. (2020). A dynamic nomenclature proposal for SARS-CoV-2 lineages to assist genomic epidemiology. *Nat Microbiol*, 5(11):1403–1407.

Resende, P. C. et al. (2020). SARS-CoV-2 genomes recovered by long amplicon tiling multiplex approach using nanopore sequencing and applicable to other sequencing platforms. Technical Report doi:10.1101/2020.04.30.069039, bioRxiv.

Seymour, C. (2020). Bonito: A pytorch basecaller for oxford nanopore reads. https://github.com/nanoporetech/bonito.

Tyson, J. R. et al. (2020). Improvements to the ARTIC multiplex PCR method for SARS-CoV-2 genome sequencing using nanopore. Technical Report doi:10.1101/2020.09.04.283077, bioRxiv.