'PePApipe': a complete bioinformatics analysis pipeline for African Swine Fever Virus genome

Vicente Lopez-Chavarrias [1], Irene Aldea [1], Jovita Fernández-Pinero [1,*]

1. Centro de Investigación en Sanidad Animal (CISA), Instituto Nacional de Investigación y Tecnología Agraria y Alimentaria (INIA), Consejo Superior de Investigaciones Científicas (CSIC), Valdeolmos, Madrid, Spain

*Corresponding author: fpinero@inia.csic.es (JF-P)

Short title: **'PePApipe', a pipeline for ASFV genome analysis**

Abstract

African Swine Fever Virus (ASFV) is of high concern in porcine livestock across the world due to both the high mortality rates and the trade restrictions imposed on affected regions. Viral genome is large and complex, but genomic analysis is essential for tracing its origin and evolution. Although several bioinformatics tools exist for genome assembly and analysis, no single platform integrates all necessary steps in an accessible and systematic way. In this study the authors developed ´PePApipe´, a custom-built, user-friendly pipeline that enables rapid, complete, and efficient ASFV genome analysis. It is specifically designed for laboratory professionals with limited bioinformatics experience, requiring only basic command-line knowledge.

Starting from raw sequencing data, 'PePApipe' integrates thirteen software tools into one automated workflow, covering quality control and pre-processing of raw reads, *denovo* genome assembly and variant calling. Programmed in Phyton, it can be executed locally through bash scripts, or using a SLURM protocol for batch processing of multiple samples. The main outputs are the ASFV consensus genome sequence and a file listing its putative variants compared to the selected reference genome.

PePApipe classifies generated files into structures folders and produces intermediate files that can be used as inputs for further or parallel analyses; also users can enable or disable specific steps in each particular case. This pipeline is adaptable and complementary to downstream steps such as viral genome annotation or genome visualization.

By consolidating all stages of viral genome analysis into a single automated workflow, PePApipe reduces the likelihood of user error and enhances reproducibility and efficiency. This easy-to-use pipeline will facilitate the transition from sequencing to assembly and analysis of viral genomes, ensuring a fast and reliable response to molecular analysis demands. Finally, the

35 pipeline can be easily adapted to the study of other viral species, expanding its application in

36 infectious diseases surveillance.

Author summary

African swine fever is a devastating viral disease that threatens pig production worldwide, causing severe economic losses and limiting international trade. Tracing and understanding how the virus spreads and evolves are key for preparedness and control of the disease, currently being a major sanitary challenge. However, viral genome analysis is often technically demanding and difficult to standardize, especially for laboratories without specialized bioinformatics expertise. In this study, we present PePApipe, an automated and user-friendly computational pipeline designed to simplify and standardize the complete genomic analysis of African swine fever virus. Starting directly from raw sequencing data, PePApipe guides users through all essential steps, from quality control to genome assembly and variant detection, producing reliable and reproducible results in a short time. By integrating multiple established tools into a single workflow and providing clear intermediate outputs, our approach reduces the risk of user error and increases transparency. Importantly, we designed PePApipe with accessibility in mind, enabling laboratory scientists to perform advanced genomic analyses with minimal computational background. While developed for African swine fever virus, the pipeline can be adapted to other viruses, making it a flexible resource for viral genomics, outbreak investigation, and future data-driven research.

Introduction

Despite the availability of generic bioinformatics tools for genome analysis across a variety of organisms, the intrinsic biological and structural differences between genomes often necessitate the development of specialized tools tailored to the organism of interest—this is particularly true in the case of viruses. One such organism is the African swine fever virus (ASFV), a large, complex, double-stranded DNA virus belonging to the family *Asfarviridae* (1). ASFV is of critical concern to the global swine industry due to its ability to cause a haemorrhagic disease with near 100% mortality in domestic pigs, as well as the substantial trade restrictions imposed on affected regions following confirmed outbreaks.

The absence of an effective vaccine, coupled with the virus's complex biology and high environmental stability, makes ASFV a formidable challenge for veterinary and agricultural systems worldwide (2, 3). Indeed, ASFV has garnered increasing scientific interest in recent years due to its severe threat to global pig production and food security derived from the worldwide epidemiological scenario. The virus has expanded beyond its original endemic regions in sub-Saharan Africa, initially entering Eastern Europe in 2007 and further spreading throughout Europe, Asia, and reaching the Americas in 2021, which has further intensified the need to understand its pathogenesis, transmission mechanisms, and evolutionary dynamics (4, 5). Moreover, knowing the origin of a new ASFV incursion and tracing the virus spread are critically demanded right after an outbreak occurs in a new region. Besides, the socioeconomic consequences of ASF outbreaks, particularly in countries with major swine industries such as China or Germany, have highlighted the urgent demand for targeted control strategies and improved diagnostic tools (6, 7).

The analysis of the ASFV genetic content is key to infer the possible origin of the viral incursion and to inform possible measures that can be put in place to avoid further transmission of the

84 virus, thus enabling a full epidemiological picture of the situation which in turn can help bring
85 the situation back under control.

86 The problem with viruses in general, and with the ASFV in particular, is that their genome is
87 prone to change (8), and this variability in nature is typically a consequence of its replication *in*
88 *vivo* when it infects host cells (domestic pigs and wild boars). With a genome of variable size
89 between 170 to 194 kbp, ASFV is able to exchange large parts of its DNA with itself or with
90 variants of the same virus in the cell environment of the host, with the resulting consequence
91 of producing new genomes. Genome variations can be simply point mutations or small indels
92 (insertions and deletions) occurring not very frequently, but nevertheless altering the original
93 genomes, or these changes can be large indels leading to gene rearrangement, and homologous
94 recombinations could occur as well (9). An additional peculiarity of ASFV genomes is that both
95 left and right terminus regions of its DNA are very variable compared to the central section;
96 moreover, these end regions contain inverted terminal repeats which further complicate both
97 the assembly and annotation of the genomes at the analysis stage (10).

98 Although several tools exist that can be used to perform assembly and analysis of DNA genomes
99 of viruses (https://zenodo.org/records/7764938), such as ASFV (11, 12), the authors of this work
100 have not found to date a tool able to compile all steps involved in this analysis in a simple and
101 accessible way.

102 The present tool, called 'PePApipe', is fit for this purpose, and implements the work in a rapid,
103 complete, efficient and clear manner. Moreover, this pipeline has been developed bearing in
104 mind that end-users are mostly laboratory professionals with limited bioinformatics skills.

105

106

107

108 # Results and discussion

109 The pipeline described on this paper has been built having into account the specific
110 characteristics of ASF viruses. Every step was tried on the sequences from the six samples using
111 all tools known to the authors to be adequate for that purpose. After different runs, the results
112 using different tools were compared with the rest and the best among them were used to decide
113 upon which tool to include in the pipeline.

114 To examine the quality of raw reads we included FastQC (13) only once for an initial quality
115 assessment and used fastp (14) instead for the post-trimming quality assessment because fastp
116 includes this assessment as a subsequent step to the trimming process. To decide on a trimming
117 tool we considered trimmomatic (15), cuatadapt (16) and fastp (14), and we opted for the latter
118 because of the post-trimming quality check that this tool provides, being the average length of
119 the resulting fragments similar with the three tools (data not shown).

120 We tested the trimming with the three tools mentioned applying a common minimum length
121 for trimmed reads of 60 and two minimum quality thresholds: 20 and 30. The reads obtained
122 with quality thresholds of 20 were consistently longer than the reads obtained when we were
123 stricter with the quality threshold (for example 30), hence we opted to include quality thresholds
124 of 20 in the trimming tool. In our experience, when working with viruses, unless the analyst has
125 an exceptionally good sequencing sample, it is better not to be too strict on quality of reads,
126 particularly when the amount of viral DNA at the outset of the experiment is not too high, even
127 after using viral enrichment methods to prepare specific libraries prior to sequencing, as it was
128 our case.

129 For the *de novo* assembly stage we focused on the tool Unicycler (17) which has been shown to
130 perform well on assembly of viruses. Moreover, although it is a tool designed to combine short
131 and long reads to perform a hybrid assembly, it delivers similar results with viruses working with

132 short reads only, which is an advantage in this case. Of course, the tool is equally capable of
133 analysing a mixed of viral short and long reads. Unicycler already includes other assembly and
134 polishing tools such as Spades (18) and Pilon (19).

135 Two different software programmes were tested for *de novo* assembly of the sequences: SPAdes
136 and Unicycler (Table 1). The Unicycler software was used with the parameter --linear_seqs 1,
137 forcing it to return a single contig. We observed a significant difference in the number of contigs
138 corresponding to ASFV obtained with each programme. The number of contigs obtained with
139 SPAdes ranged between 16 and 504 whereas with Unicycler was between 3 and 6,
140 demonstrating the extra scaffolding gained with other programmes included in Unicycler such
141 as Pilon (19).

142 It is worth noting that we obtained more contigs with SPAdes than those shown in Table 1, but
143 we have only displayed the contigs corresponding to the virus of interest. In Sample E2, for
144 example, nearly 9000 contigs were obtained that did not correspond to ASFV, but to other
145 organisms. This large number of contigs obtained with SPAdes complicates the analysis of the
146 virus that is of interest to us.

147 Differences in the length of the longest contig were also observed. With Unicycler, we obtained
148 a contig between 40k and 90k nucleotides longer in 4 out of the 6 samples. In the samples where
149 we obtained shorter contigs with Unicycler, the difference ranged from 2 to 154 nucleotides,
150 therefore we chose Unicycler as the best *de novo* assembling tool for the virus of interest.

151

152

153

154

155

156 *Table 1. Comparison of parameters obtained using SPAdes and Unicycler as de novo assembly tools for*

157 *ASFV sequences.*

SAMPLE	ASFv contigs SPAdes	ASFv contigs Unicycler	Larger contig SPAdes	Larger contig Unicycler
E1	16	3	185354	185352
E2	120	3	185506	185352
E3	120	4	115053	185352
E4	256	6	113089	185352
E5	504	3	30747	125185
E6	117	4	145211	185353

158

159 One of the main challenges when performing bioinformatics analyses of viral sequences is the
160 need to obtain the truest and most accurate consensus sequence of the virus being analysed.
161 Any rough approximation is useless and we must aim to obtain the true genetic picture of the
162 virus we are confronting to. To attain this goal, in our pipeline we used a combination of two
163 published and reliable analysis tools: RagTag and MeDuSa.

164 Two of the four available functionalities from RagTag (20) are used in the pipeline: RagTag
165 Correct and RagTag Scaffold. Most of the contigs are joined together using a virus reference
166 genome as a guide but without inclusion of any of the nucleotides from the viral reference
167 sequence. Once this compilation job is done, MeDuSa (21) implements a final join of the
168 remaining contigs, this time using as references as many as we can/want to provide. In our case
169 we used 10 references of ASFV genotype II, as these were phylogenetically closest to our
170 problem viruses.

171 MeDuSa attempts at further connect contigs by matching our problem virus with the sequences
172 of the references provided, and similarly to RagTag, it does not use any of the nucleotides off
173 the references themselves.

174 In our experience, approximately 80% of the assemblies we obtained were reduced to a single
175 contig, and in the remaining 20% of cases the final contig sets included up to three, and in very
176 few occasions, up to five contigs. The two main reasons for not being able to assemble to a final
177 single contig were either lower quality of reads or possible contamination (in our case with
178 bacterial plasmids of around 5,000 nucleotides).

179 For the alignment with reference (mapping) of our sequences we tested mainly two viral
180 references: ASFV Georgia 2007/1 (GenBank assembly GCA_003047755.2) and ASFV Lithuania
181 LT14/1490 (GenBank assembly GCA_008932025.1). Although all results were very similar in both
182 cases in terms of quality and metrics, we opted for ASFV Georgia 2007/1 for all the analyses, as
183 it represents the initial reference strain of the entrance of ASFV genotype II into the Eurasian
184 territory

185 For the mapping, three different software programmes were tested: minimap2 (22), Bowtie2
186 (23) and bwa-mem2 (24) (Figure 2). Bowtie2 was used with two different parameter sets: --very-
187 sensitive-local and the default parameters. Both minimap2 and bwa-mem2 perform local
188 alignment by default, meaning that the entire read does not need to match the reference
189 genome. Bowtie2 does not use this type of alignment by default, so the corresponding
190 parameter must be added manually.

191 The percentage of reads mapping to the reference was higher in all cases with the bwa-mem2
192 software. As expected, the lowest percentages were obtained with Bowtie2 using the default
193 parameters, although in five out of the six samples analyzed, Bowtie2 with the local mapping
194 parameter was the second software that mapped most reads (Figure 2). The bwa-mem2

software was ultimately selected because it achieved higher percentages of mapped reads in all samples, optimizing the variant calling process.

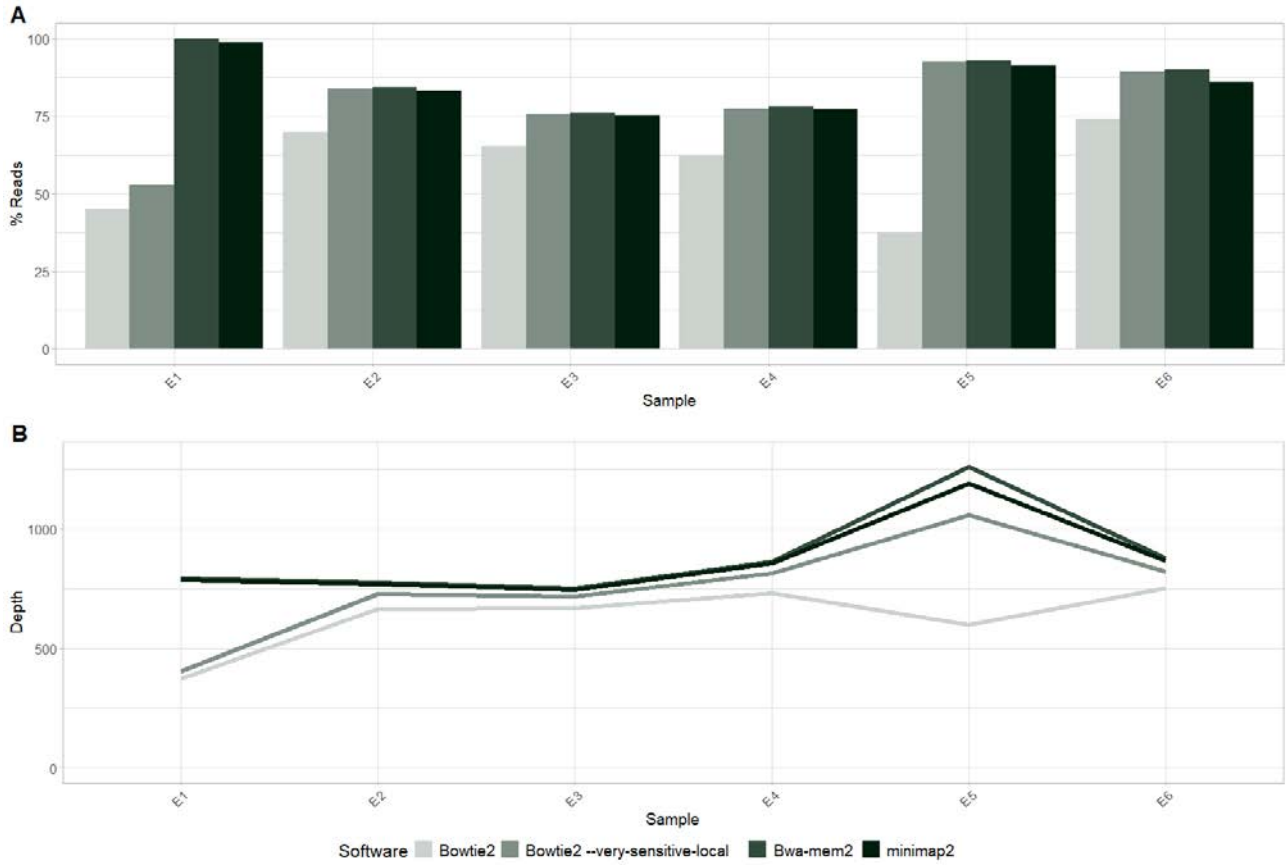

Figure 2. Comparison of Bowtie2, bwa-mem2 and minimap2 as mapping tools for ASFV sequences.

This pipeline yields optimal results when used in conjunction with an enrichment strategy designed to isolate and characterize viral sequences, such as through a custom-built viral genome library or by using amplicon-based approach. In the absence of such a capture step, the proportion of non-viral reads is likely to be very high, which can significantly reduce the reliability of the results compared to those obtained with prior enrichment.

207 Table 2 shows, as an example, a summary of results and statistics of complete genome analyses
208 from six different ASFV viral samples (in columns). The different metrics shown (in rows) give an
209 idea of the variability observed among these viral samples.

210

211

212

213

214

215

216

217

218

219

220

221

222

223

224

225

226

227

228 *Table 2. Example of summary of results and statistics of complete genome analyses of several viral*

229 *strains.*

Statistics	Sample E1	Sample E2	Sample E3	Sample E4	Sample E5	Sample E6
Per base sequence quality (average R1+R2)	36	36	36	36	36	36
% G+C (average R1+R2)	39%	38.5%	39%	39%	41%	38%
Total number of reads	1,049,376	1,211,198	1,298,236	1,486,248	2,055,450	1,282,808
Duplication rate	33.9%	17.4%	18.7%	22.3%	18.5%	15.7%
Reads passing quality filters	1,038,682	1,198,676	1,283,168	1,461,022	2,039,472	1,268,262
% reads passing quality filters	99.98%	98.97%	98.84%	98.30%	99.22%	98.87%
Reads mapped to Reference genome	1,036,938	1,009,847	976,483	1,142,610	1,897,109	1,141,496
% reads mapped to Reference genome	99.83%	84.25%	76.10%	78.21%	93.02%	90%

Mean coverage per contig (depth or vertical coverage)	795	777	751	863	1,260	877
No. of initial fragments before MeDuSa step at *de novo* assembly	2	3	2	3	5	2
No. of final fragments after MeDuSa step at *de novo* assembly	1	1	1	1	1	1
Total *de novo* assembled length (horizontal coverage)	187,794	187,892	187,891	187,898	188,341	187,893

235 The analysis of variants in genomic sequences that are subject to high variation, such as those
236 obtained with viruses, is not an easy task. ASFV contains a DNA genome and is expected to show
237 a low level of mutation. However, its large size (up to 194 kbp) and its ability to delete, insert,
238 duplicate, invert, and recombine short and large genome regions, make variant calling a
239 challenging step. We explored several tools to try to standardize this search. First we used the
240 Nextflow platform (25) nf-core/viralrecon (26), which is a tool that was developed during the
241 Covid-19 pandemic to analyze new coronavirus variants. We found the installation and
242 processing cumbersome, and the outputs did not fulfil the expectations we had beforehand on
243 this tool.

244 After searching for other appropriate tools we decided to compare two bayesian inference
245 searching tools (BCFtools (27, 28) and Freebayes (29)) with a non-bayesian tool (VarScan 2) (30),
246 using an additional fourth tool (basic variant detection included in the CLC Genomics Workbench
247 software) (31) to analyze the overall results from four perspectives, performing an orthogonal
248 analysis. The main finding after using the four tools was the lack of variants found by the two
249 Bayesian methods, compared to VarScan 2 which found several variants. Overall, CLC found the
250 same variants as VarScan 2, and the minimal discrepancies between the two were mostly related
251 to considering or not considering non-canonical nucleotides as variants.

252 In our opinion, it is hard to believe that the viral sequences we analysed did not have any variants
253 with regards to the references used, considering the high variability observed in viruses.
254 Therefore, BCFtools or Freebayes could be more adequate for other types of genomes, and we
255 decided to opt for VarScan 2 as the main tool for variant calling.

256 The validation of variants provided by VarScan 2 was carried out using a visualization tool [IGV
257 (Integrative Genomics Viewer)] (32-35), which is the software tool most universally used for
258 visualizing genome mappings. This mapping is an optional step and is not an integral part of the
259 pipeline.

260 The *de novo* consensus sequence built by the pipeline can be annotated using a reference
261 genome of choice. Although this is neither an essential step in the pipeline but rather a
262 complementary analysis that can be carried out alongside, it is an essential task to identify and
263 tag the different genes that have been assembled together. A popular tool released so far for
264 this purpose is a universal tool for general annotation that works with viruses, and specifically
265 with ASFV, called GATU (General Annotation Transfer Utility) (36).

266 To our knowledge, two other pipelines have been recently described to specifically analyse ASFV
267 genome reads, revealing the necessity and relevance of ASFV tailored bioinformatics solutions.
268 The first, published by Spinard *et al.* in August 2024 (11), is designed to process both short and
269 long reads. The second, ANASFV (37), developed by Li *et al.* and released in September 2024, is
270 specifically tailored for long-reads.

271 The pipeline developed by Spinard *et al.* was tested on two genomes only, while PePApipe was
272 evaluated with six genomes. Although expanding the number of samples for testing would be
273 ideal, this is not always feasible. Nevertheless, additional samples can be obtained from public
274 repositories to increase the dataset size; however, the sequencing quality or the total viral read
275 count from such sources may vary and not always meet desired standards.

276 For short-read analysis, Spinard *et al.* employed a reference-based approach, which, as men-
277 tioned earlier, can introduce identification bias by limiting the discovery of novel genomes. On
278 the other hand, their use of *de novo* assembly with both short and long reads helps resolve or
279 better elucidate inverted terminal repeat (ITR) regions compared to short reads alone. To ad-
280 dress this limitation, PePApipe also supports a combined approach using short and long reads,
281 with long reads incorporated in the initial *de novo* assembly step using Unicycler. One important
282 difference between the pipeline by Spinard *et al.* and PePApipe is that the former allows sub-
283 mission of up to four datasets for analysis, whereas PePApipe can process as many samples as
284 needed in batch mode, albeit at the expense of longer processing times.

285 In our analyses more than 50% of Illumina reads mapped to both the ASFV virus and *Sus scrofa*
286 (host) genomes. The straight removal of all reads mapping to *Sus scrofa* would mean losing as
287 well an important amount of reads mapping to the ASFV virus, which are essential for the viral
288 genome assembly. Moreover, the pipeline by Spinard *et al.* does not use reads mapping to both
289 the ASFV virus and host to find the most similar reference to their genome (among 33 reference
290 genomes), and this may result in an inadequate identification of such reference genome.
291 Although 10 reference genomes were used in PePApipe via MeDuSa, a larger panel of reference
292 sequences may be used. The final consensus sequence may be more reliable at the expense of
293 prolonging processing times, although the assembly improvement is likely to be marginal.

294 The first pipeline by Spinard *et al.* uses the *de novo* assembled contigs to predict a closely related
295 reference genome whereas PePApipe do not make inferences about a reference genome but
296 constructs the *de novo* assembly in one go by combining the use of RagTag and MeDuSa.

297 Moreover, Spinard *et al.* use reference genomes to align ITR segments that remain unaligned
298 and flanking those ITR segments that could be aligned in the first place. This practice can curtail
299 the possibilities of correct alignment of ORFs not present in such reference genomes but present
300 in a potential 'new' virus.

301 The search for variants strategy of Spinard *et al.* is based on the use of BCFtools but we hardly
302 found variants using this tool, as explained above. The reason for this difference could be that
303 we used samples that were very similar between each other and between them and the
304 reference genome sequence, as they were all 'in vivo' passes of the same virus. Using BCFtools
305 with a more heterogeneous dataset may result in more variants called.

306 The publication describing the ANASFV pipeline does not provide as detailed an explanation of
307 the assembly stage as PePApipe or the pipeline by Spinard *et al.* ANASFV employs reference
308 genomes to correct low-quality reads typical of Oxford Nanopore Technologies (ONT)
309 sequencing, which guides the assembly process but may overlook open reading frames (ORFs)

310 not present in the reference genomes. Additionally, the authors highlight that substantial host
311 DNA contamination is a significant challenge in ONT sequencing, often leading to inadequate
312 coverage. Rather than completely removing all reads that map to the host (*Sus scrofa*), this
313 problem might be better addressed by removing only those reads that map exclusively to the
314 host and not to the ASFV genome.

315 Regardless of the pipeline used, a suitable computational infrastructure and personnel with
316 basic bioinformatics expertise are essential. However, we believe PePApipe may need lower
317 computer resources than other pipelines as it was designed with the aim of providing simplicity
318 in the first place, thus every step is clearly detailed, as can be found in the pipeline access
319 websites.

320

321 Methods

322 DESIGN OF THE ALGORITHM

323 This pipeline, developed in Python, can be run using basic Linux commands to launch a Bash
324 script with a Slurm protocol and can be executed on multi-sample batches. The work is
325 sequentially performed over three main areas of genomic analysis: quality control and pre-
326 processing of raw reads; *denovo* genome assembly and variant calling (Figure 1). Starting from
327 raw data (paired .FASTQ files) obtained from short-read sequencing platforms (Illumina),
328 'PePApipe' implements in a sequential manner all crucial steps by using 13 software tools
329 adequately built into a single pipeline.

330 This pipeline can be run in a locally or remotely by accessing a computing server. A basic
331 knowledge of Linux programming language is necessary both to install the programmes and to
332 run the pipeline.

The pipeline with all associated information (including the pipeline itself and all accessory scripts) is accessible through the repositories Protocols.io (https://www.protocols.io/private/DE331EDEA81311F0AE550A58A9FEAC02) and GitHub (https://github.com/vilocha/PePApipe-pipeline) under the name 'PePApipe: a complete bioinformatics analysis pipeline for African Swine Fever Virus genomes'.

Detailed installation steps necessary to build the pipeline

The following steps must be implemented in the suggested order to successfully install all necessary programmes and be able to run this pipeline. The actual code lines used within the pipeline are clearly shown in the pipeline script itself accessible through GitHub and Protocols.io. These code lines are already built within the pipeline and will be executed at once when the pipeline is run.

The headers in the bash scripts specify the computation resources needed in the computer or server to allow the execution of the pipeline and its accessories.

According to needs, the different modules in the pipeline can be activated/ deactivated by just uncommenting/ commenting them in the main section of the pipeline script, respectively.

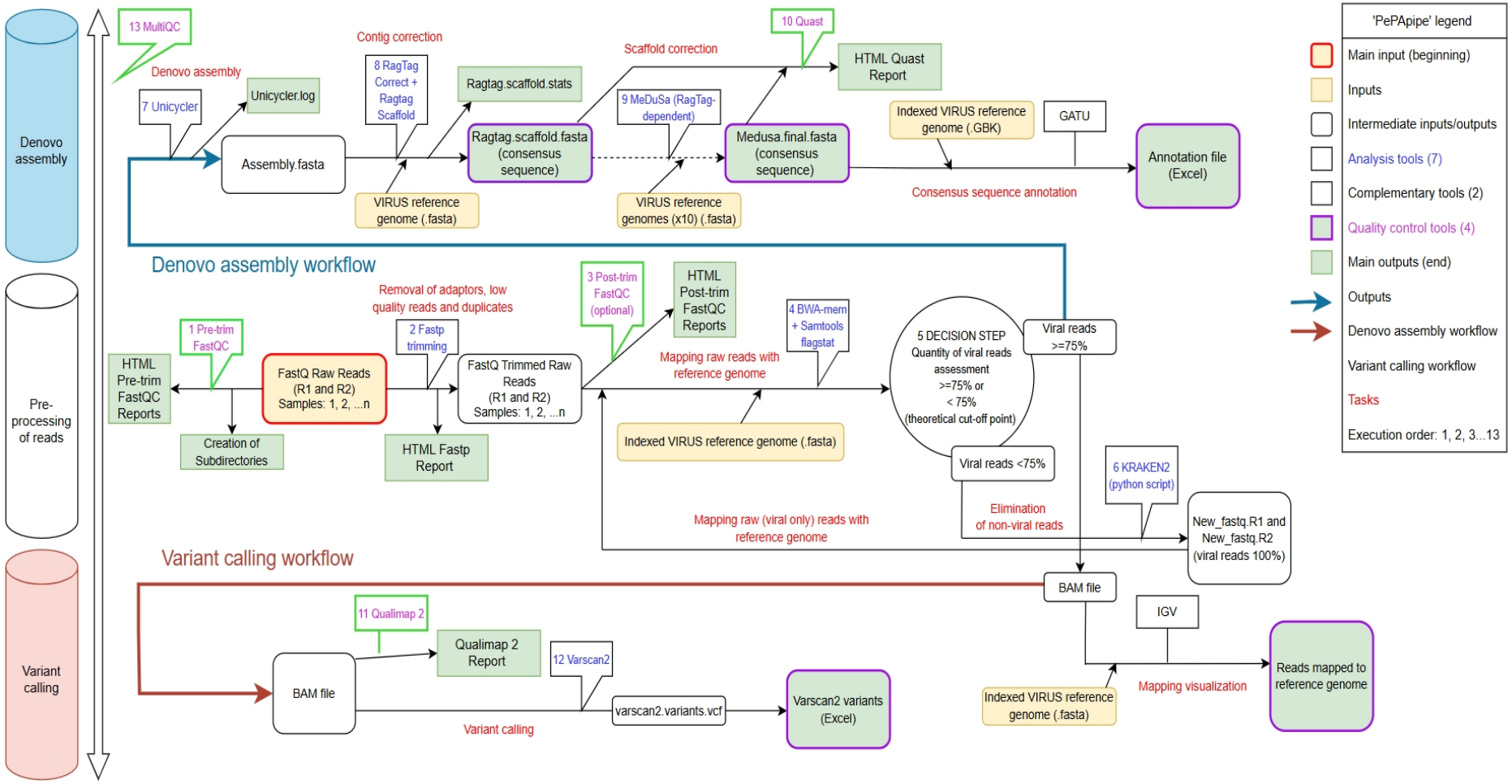

Figure 1. Overall bioinformatic analysis flow followed with PePApipe (in-house built Python pipeline specifically designed for ASF viruses).

SYSTEM AND METHODS IMPLEMENTATION

The input or starting point of the pipeline is always the set of raw reads which are the outputs of the sequencing platform (short reads from Illumina, although long reads from Oxford Nanopore Technologies can also be used) organized in two files: .FASTQ_R1, or forward reads, and .FASTQ_R2, or reverse reads. This is the standard output of pair-end sequencing protocols, which are common when working with Illumina protocols. The output files of the different steps in this pipeline are specified on the installation webpages accessible through GitHub and Protocols.io. We tested all steps with Illumina sequencing products (paired sampling) from six ASFV samples named E1 to E6 (38).

STEPS:

Quality control of reads

This is the very first step which is necessary to ensure a good quality of the .FASTQ files provided by the sequencing platform. The quality check of raw reads is carried out with the tool FastQC (13). As a general rule, the per-base sequence quality (Phred score) should always be ideally over 28, and it is recommended not to drop it below 20.

Once the quality control step is complete and one is satisfied that the quality of sequences is adequate to continue the processing, the reads are 'cleaned' with the tool fastp (14). This step is necessary to ensure that adapters and low quality and duplicate reads are removed off the .FASTQ files. In addition, fastp produces a post-trimming quality analysis report, making it unnecessary to perform a further stand-alone quality control with FastQC.

Only reads with a minimum Phred quality score of 20 (the default fastp Phred quality score is 15) and a minimum length of 15 base pairs (default parameter) are kept after applying fastp,

373 but these parameters can be changed in the code line according to user needs. Depending on
374 the number of reads available, these parameters can be changed up or down with the aim of
375 maximizing output and minimizing low quality reads.

376

377 Viral reads content check

378 In this step the reads that passed the previous filters ('clean reads') are mapped against the
379 virus reference genome in .FASTA format chosen for convenience in each particular case using
380 the software tools BWA-mem2 (24) and Samtools (28).

381 This step is probably the most important of all steps included in this part of the flow, and of
382 the pipeline for that matter. The viral genome reference used for convenience by the authors
383 of the pipeline was ASFV Georgia 2007/1 (Genbank assembly: GCA_003047755.2), since it was
384 the initial isolate of genotype II emerging in Eastern Europe in 2007 and is considered by the
385 scientific community as the reference strain.

386 The objective of this mapping is two-fold: on the one hand it allows to quantify the amount of
387 reads within the original .FASTQ files actually mapping with an ASFV (and thus to also quantify
388 the amount of reads not mapping with an ASFV), and on the other hand, to create a set of
389 reference-mapped contigs (.BAM file) as a first step in the search of genetic variants (SNPs and
390 indels) in the genome of our virus problem compared to the genome of the virus reference used.

391

392 Quality check of reads mapped to reference genome of choice

393 One of the tools most commonly used for this purpose is Qualimap 2 (39). The rationale
394 behind is to assess the quality of the mapping of trimmed raw reads to virus reference genome

395 which was performed to calculate the amount of viral reads present. The outputs can be
396 verified in .PDF and .HTML formats.

397

398 Quantitative assessment of viral reads (theoretical cut-off point: 75%)

399 It is important to carry out all bioinformatics analyses only on viral reads free of contamination
400 with reads from other origins (host, environment or handling personnel) or, if this is not possible,
401 at least on a number of viral reads as high as possible. This is because the results we obtain if
402 the analyses are performed on a high number of non-viral reads may be spurious and therefore
403 not representative of the virus we are trying to characterize.

404 This is a point in the pipeline where a decision has to be made as to the quality level we want to
405 apply to our analysis in relation to the available reads (Figure 1). A theoretical threshold (cut-off
406 point) of 75% has been established by the authors in order to decide whether to continue the
407 analyses with the original .FASTQ files already filtered ('clean reads') or whether to carry out yet
408 another filtering step to get rid of non-viral reads.

409 Apart from the methods used to isolate the viral DNA, the method used to prepare the libraries
410 (with or without enrichment) is the single most influential step on the amount of foreign (non-
411 viral) DNA reads found in our sample. If a non-targeted or non-enriched library preparation
412 method is used, there will be a huge number of non-viral reads present in the .FASTQ files, thus
413 these may be larger in size although not necessarily better in quality.

414 Hence, this decision threshold is arbitrary and may vary depending on the initial .FASTQ files and
415 whether the library has been prepared using enrichment or not. Libraries prepared for shotgun
416 sequencing are likely to contain <1% of viral fragments.

417

418 ## Removal of reads not belonging to Viruses

419 If we decide to carry out another filtering step to get rid of non-viral reads, a new set of .FASTQ
420 files must be created by mapping the 'clean reads' against all published reliable reads
421 belonging to the Superkingdom Viruses (NCBI: txid10239). To carry out the removal of non-
422 viral reads a loop with an additional Python script using the tool Kraken2 (40) is necessary. The
423 procedure consists of two steps: building of the virus database and extraction of the viral
424 reads. This extra filtering step will create a new set of .FASTQ files containing only viral reads.

425 As already mentioned, this procedure can be performed whenever the chosen threshold of
426 viral reads is not reached (in our case 75%) or it may also be systematically run every time,
427 independently of the amount of non-viral reads present in the original 'cleaned' set of .FASTQ
428 files.

429 If there is availability of a substantial amount of samples, a benchmark exercise can be carried
430 out to refine the decision threshold or cut-off percentage point, making it lower or higher than
431 75% according to each specific situation.

432

433 ## *De novo* viral assembly

434 This step is mainly performed with the software tool Unicycler, and ultimately refined with the
435 programmes RagTag (using Unicycler outputs as inputs) and MeDuSa (using RagTag outputs as
436 inputs).

437 These tools use several dependency programmes in order to obtain a full viral *de novo* genome
438 assembly starting from the .FASTQ files containing our already 'cleaned' reads, be these the
439 original .FASTQ files or the new set created using Kraken2.

440 Although Unicycler (17) was originally built for bacteria, ideally combining both short (Illumina)
441 and long reads (Oxford Nanopore Technologies), it performs equally well using only viral
442 genomes short reads.

443 A most positive aspect about this type of assembly is that a viral reference genome is not used
444 at all, thus no bias from this source can be introduced, allowing for a genuine assembly of the
445 putative whole genome of our virus problem. This step is the most important within the *de novo*
446 assembly part of the flow, but unfortunately it is also the most consuming in terms of time and
447 resources.

448 There are particularities in relation to the three tools used in the *de novo* assembly. In order to
449 obtain a proper assembly of our virus whole genome it is important to specify that what we are
450 looking for is a linear sequence, in other words, that the expected number of linear (non-circular)
451 sequences in the underlying sequence being assembled is just one, and so these requirements
452 must be included in the code in the Unicycler section.

453 After the first assembly step with Unicycler is concluded, a correction of the obtained contigs
454 converting these into scaffolds is performed with the RagTag tool (20). This step is necessary to
455 guide the *de novo* assembly performed by Unicycler into plausible contigs belonging to the ASFV
456 virus, using for that purpose the chosen viral reference genome in .FASTA format (in our case
457 ASFV Georgia 2007/1; Genbank assembly: GCA_003047755.2).

458 It is important to note that, even though a viral reference is used, not a single fragment of it is
459 incorporated into the assembly obtained with Unicycler, so the only activity performed in this
460 step is that of contig correction and scaffolding. RagTag comprises four software tools ('correct',
461 'scaffold', 'patch' and 'merge') but only the first two are used in the pipeline. 'Patch' is not used
462 to avoid adding viral reference fragments and 'merge' is not necessary since only 'scaffold' was
463 used as scaffolding method.

464 After the correction and scaffolding performed by RagTag, a third tool called MeDuSa (21) is
465 incorporated as part of the *de novo* assembly process. MeDuSa performs a scaffold correction
466 in a similar way to RagTag, but instead of using only one viral reference it uses as many well
467 curated references as we are able to provide, always making sure the references are akin to the
468 ASFV genotype we are studying. Up to 10 references belonging to ASFV virus genotype II have
469 been included in this pipeline, which in alphabetical order are: Arm_07_CBM_c2_LR812933.1,
470 ASFV_Georgia_2007_1_FR682468.2.fasta, ASFV_HU_2018_MN715134.1,
471 ASFV_LT14_1490_MK628478.fasta, Belgium_2018_1_LR536725.1, Estonia_2014_LS478113.1,
472 Korea_YC1_2019_ON075797.1, POL_2015_Podlaskie_MH681419.1,
473 Tanzania_Rukwa_2017_1_LR813622.1 and Ulyanovsk_19_WB_5699_MW306192.1.

474 The higher the number of references used, the better the result of this step will be. Briefly, the
475 contigs obtained from the previous step are further assembled into larger scaffolds with
476 MeDuSa. If the quality of the original reads is high this step generally results in one final contig,
477 which is the putative consensus sequence of the virus under study. On the contrary, poor quality
478 sequences may result in a large number of contigs, and if the number of contigs exceeds 3-5 it
479 would be advisable to reject those sequences.

480

481 Quality assessment of the *de novo* assembly

482 This is a quality analysis that can be performed over all steps in the *de novo* assembly loop with
483 the intention of investigating the overall quality of the process. The tool included in the
484 pipeline for this task is Quast (QUality ASsessment Tool) (41) and the outputs are available on
485 different formats: .TXT, .PDF, .TSV, .HTML, etc.

486

487 Search for viral genome variants (variant calling)

488 Several tools are used to reveal genetic variants (SNPs and indels) present in the virus genome
489 under study when compared to a viral reference genome.

490 In general, there are two main types of tools: those that use a Bayesian inference and those that
491 use a deterministic/numerical approach. The tool included in this pipeline is called VarScan 2
492 (30) and is one of the second type (numerical approach).

493 In order to obtain the most plausible record of variants with this tool the following five
494 parameters have been defined with the specified values:

495 - Minimum number of reads with change to call a variant: 5

496 - Minimum depth (coverage=total number of reads supporting that variant): 20

497 - P-value: 0.001

498 - Minimum average quality: 30

499 - Minimum variant frequency in that position: 0.05 (to account for true minority
500 variants)

501

502 We used a tool for variant calling provided by Qiagen CLC Genomics Workbench (31) to
503 compare the variant calling results obtained with VarScan 2 to ensure that the latter performs
504 adequately.

505

506 Overall quality assessment of the analysis

507 In order to perform an overall assessment of the analysis, the tool MultiQC (42) may be used.
508 This tool evaluates all inputs and outputs in the working directory and performs a general
509 quality summary assessment, which is based on the quality of the inputs (.FASTQ files) and on

510 the type of programmes used. This tool must be executed in the working directory where all
511 folders used during the running of the pipeline are located. This final step is valuable if we
512 want to perform benchmark exercises to compare different inputs or different tools or both.

513

514 Expected outputs and their interpretation

515 The pipeline produces all necessary files to interpret the results, classifying them into folders,
516 in an approximate time that depends on the size of the .FASTQ files, from 10 minutes with files
517 of around 0.5 Mb through to 30 minutes with files of 1 Mb in size, or to longer times if high
518 capacity sequencers are used. Several intermediate files are also created in the process, which
519 can be used as inputs for further or parallel analyses, as well as to check for possible errors
520 during the analysis. Among these complementary post-analysis steps are viral annotation with
521 GATU (https://4virology.net/virology-ca-tools/gatu/) or genome visualization with IGV
522 (https://igv.org/).

523 All steps are amenable to user control by means of switching on/off the necessary sections in
524 each particular case. There are several steps where the quality of the processes can be
525 checked to make sure the progress of the results is in line with the user expectations.

526 The pipeline can be easily adapted to viruses other than ASFV by changing the parameters
527 relevant to the new virus and running the specific pipeline sections accordingly.

528 The most important outputs are the consensus sequence of the genome of the virus we are
529 studying and the table of variants for that particular virus against the chosen reference
530 genome. Secondary outputs are reference mapping parameters, *de novo* assembly parameters
531 and quality assessment summaries.

532 PePApipe is user-friendly and can be installed through a straightforward series of steps. While
533 it is primarily designed for analyses involving short reads, it is also compatible with long-read

data. The outputs are consistently organized into folders, providing a structured basis for further analysis. Additionally, the code is open-source and can be modified and adapted to meet user specific needs, provided that appropriate credit is given.

DATA AVAILABILITY

The primary data used in this paper is available at the ENA repository (https://www.ebi.ac.uk/ena/browser/home) under accession number PRJEB105327, from the 19th December 2025.

CODE AVAILABILITY

The code needed to run the analyses with all associated information (including the pipeline itself and all accessory scripts) is open source and accessible through the repositories Protocols.io (https://www.protocols.io/private/DE331EDEA81311F0AE550A58A9FEAC02) and GitHub (https://github.com/vilocha/PePApipe-pipeline) under the name 'PePApipe: a complete bioinformatics analysis pipeline for African Swine Fever Virus genomes'.

Support contact email for users is available at the websites above.

'PePApipe' is released under the MIT License.

Private Protocols.io link for reviewers: https://www.protocols.io/private/DE331EDEA81311F0AE550A58A9FEAC02

FUTURE DIRECTIONS

The future plan is to include a viral genome annotation tool in the pipeline to harmonize this crucial and complex step for a large virus which can encompass more than 180 ORFs along the genome.

557 Finally, although the original aim of this study was to design a pipeline to specifically build and
558 analyse the ASFV genome from NGS experiments, this can be easily adapted for the study and
559 assembly of any other viral organisms.

560

561 ACKNOWLEDGMENTS

562 Dr. Carmina Gallardo and Prof. Marisa Arias, from CISA-INIA-CSIC, are recognized for their
563 contribution to provide the ASFV samples used for this study.

564

565 REFERENCES

566 1. Alonso C, Borca M, Dixon L, Revilla Y, Rodriguez F, Escribano JM, et al. ICTV Virus
567 Taxonomy Profile: Asfarviridae. J Gen Virol. 2018;99(5):613-4.
568 2. Rock DL. Challenges for African swine fever vaccine development-"... perhaps the end
569 of the beginning.". Vet Microbiol. 2017;206:52-8.
570 3. Sanchez-Cordon PJ, Montoya M, Reis AL, Dixon LK. African swine fever: A re-emerging
571 viral disease threatening the global pig industry. Vet J. 2018;233:41-8.
572 4. Costard S, Mur L, Lubroth J, Sanchez-Vizcaino JM, Pfeiffer DU. Epidemiology of African
573 swine fever virus. Virus Res. 2013;173(1):191-7.
574 5. Zhou X, Li N, Luo Y, Liu Y, Miao F, Chen T, et al. Emergence of African Swine Fever in
575 China, 2018. Transbound Emerg Dis. 2018;65(6):1482-4.
576 6. Gallardo C, Fernandez-Pinero J, Arias M. African swine fever (ASF) diagnosis, an
577 essential tool in the epidemiological investigation. Virus Res. 2019;271:197676.
578 7. Wang T, Sun Y, Qiu HJ. African swine fever: an unprecedented disaster and challenge
579 to China. Infect Dis Poverty. 2018;7(1):111.
580 8. Forth JH, Calvelage S, Fischer M, Hellert J, Sehl-Ewert J, Roszyk H, et al. African swine
581 fever virus - variants on the rise. Emerg Microbes Infect. 2023;12(1):2146537.
582 9. Zhu Z, Xiao CT, Fan Y, Cai Z, Lu C, Zhang G, et al. Homologous recombination shapes the
583 genetic diversity of African swine fever viruses. Vet Microbiol. 2019;236:108380.
584 10. Dixon LK, Chapman DA, Netherton CL, Upton C. African swine fever virus replication
585 and genomics. Virus Res. 2013;173(1):3-14.
586 11. Spinard E, Dinhobl M, Erdelyan CNG, O'Dwyer J, Fenster J, Birtley H, et al. A
587 Standardized Pipeline for Assembly and Annotation of African Swine Fever Virus Genome.
588 Viruses. 2024;16(8).
589 12. Li K, Han X, Shao Y, Zhang Y, Zhao X, Johnson EW, et al. ANASFV: a workflow for ASFV
590 whole genome sequencing, assembly, and evaluation. bioRxiv. 2024:2024.07.08.602471.
591 13. Andrews S. FastQC: A Quality Control Tool for High Throughput Sequence Data. 2010.

14. Chen S, Zhou Y, Chen Y, Gu J. fastp: an ultra-fast all-in-one FASTQ preprocessor. Bioinformatics. 2018;34(17):i884-i90.
15. Bolger AM, Lohse M, Usadel B. Trimmomatic: a flexible trimmer for Illumina sequence data. Bioinformatics. 2014;30(15):2114-20.
16. Martin M. Cutadapt removes adapter sequences from high-throughput sequencing reads. EMBnetjournal. 2011;17(1):3.
17. Wick RR, Judd LM, Gorrie CL, Holt KE. Unicycler: Resolving bacterial genome assemblies from short and long sequencing reads. PLoS Comput Biol. 2017;13(6):e1005595.
18. Prjibelski A, Antipov D, Meleshko D, Lapidus A, Korobeynikov A. Using SPAdes De Novo Assembler. Curr Protoc Bioinformatics. 2020;70(1):e102.
19. Walker BJ, Abeel T, Shea T, Priest M, Abouelliel A, Sakthikumar S, et al. Pilon: an integrated tool for comprehensive microbial variant detection and genome assembly improvement. PLoS One. 2014;9(11):e112963.
20. Alonge M, Lebeigle L, Kirsche M, Jenike K, Ou S, Aganezov S, et al. Automated assembly scaffolding using RagTag elevates a new tomato system for high-throughput genome editing. Genome Biol. 2022;23(1):258.
21. Bosi E, Donati B, Galardini M, Brunetti S, Sagot MF, Lio P, et al. MeDuSa: a multi-draft based scaffolder. Bioinformatics. 2015;31(15):2443-51.
22. Li H. Minimap2: pairwise alignment for nucleotide sequences. Bioinformatics. 2018;34(18):3094-100.
23. Langmead B, Salzberg SL. Fast gapped-read alignment with Bowtie 2. Nat Methods. 2012;9(4):357-9.
24. Vasimuddin M, Misra S, Li H, Aluru S, editors. Efficient Architecture-Aware Acceleration of BWA-MEM for Multicore Systems. 2019 IEEE International Parallel and Distributed Processing Symposium (IPDPS); 2019 20-24 May 2019.
25. Ewels PA, Peltzer A, Fillinger S, Patel H, Alneberg J, Wilm A, et al. The nf-core framework for community-curated bioinformatics pipelines. Nat Biotechnol. 2020;38(3):276-8.
26. Simonetti M, Zhang N, Harbers L, Milia MG, Brossa S, Huong Nguyen TT, et al. COVseq is a cost-effective workflow for mass-scale SARS-CoV-2 genomic surveillance. Nat Commun. 2021;12(1):3903.
27. Li H. A statistical framework for SNP calling, mutation discovery, association mapping and population genetical parameter estimation from sequencing data. Bioinformatics. 2011;27(21):2987-93.
28. Danecek P, Bonfield JK, Liddle J, Marshall J, Ohan V, Pollard MO, et al. Twelve years of SAMtools and BCFtools. Gigascience. 2021;10(2).
29. Garrison E, Marth G. Haplotype-based variant detection from short-read sequencing. arXiv preprint arXiv:12073907. 2012.
30. Koboldt DC, Zhang Q, Larson DE, Shen D, McLellan MD, Lin L, et al. VarScan 2: somatic mutation and copy number alteration discovery in cancer by exome sequencing. Genome Res. 2012;22(3):568-76.
31. Workbench QCG. QIAGEN, Aarhus, Denmark [20.0.4.:[Available from: https://digitalinsights.qiagen.com/products-overview/discovery-insights-portfolio/analysis-and-visualization/qiagen-clc-genomics-workbench/.
32. Robinson JT, Thorvaldsdottir H, Winckler W, Guttman M, Lander ES, Getz G, et al. Integrative genomics viewer. Nat Biotechnol. 2011;29(1):24-6.
33. Thorvaldsdottir H, Robinson JT, Mesirov JP. Integrative Genomics Viewer (IGV): high-performance genomics data visualization and exploration. Brief Bioinform. 2013;14(2):178-92.
34. Robinson JT, Thorvaldsdottir H, Wenger AM, Zehir A, Mesirov JP. Variant Review with the Integrative Genomics Viewer. Cancer Res. 2017;77(21):e31-e4.
35. Robinson JT, Thorvaldsdottir H, Turner D, Mesirov JP. igv.js: an embeddable JavaScript implementation of the Integrative Genomics Viewer (IGV). Bioinformatics. 2023;39(1):btac830.

643 36. Tcherepanov V, Ehlers A, Upton C. Genome Annotation Transfer Utility (GATU): rapid
644 annotation of viral genomes using a closely related reference genome. BMC Genomics.
645 2006;7:150.
646 37. Li K, Han X, Shao Y, Wu X, Zhao X, Johnson EW, et al. ANASFV: a workflow for African
647 swine fever virus whole-genome analysis. Microb Genom. 2025;11(9).
648 38. Gallardo C, Meszaros I, Soler A, Fernandez-Pinero J, van den Born E, Simon A, et al.
649 Double Deletion of EP402R and EP153R in the Attenuated Lv17/WB/Rie1 African Swine Fever
650 Virus (ASFV) Enhances Safety, Provides DIVA Compatibility, and Confers Complete Protection
651 Against a Genotype II Virulent Strain. Vaccines (Basel). 2024;12(12).
652 39. Okonechnikov K, Conesa A, Garcia-Alcalde F. Qualimap 2: advanced multi-sample
653 quality control for high-throughput sequencing data. Bioinformatics. 2016;32(2):292-4.
654 40. Wood DE, Lu J, Langmead B. Improved metagenomic analysis with Kraken 2. Genome
655 Biol. 2019;20(1):257.
656 41. Gurevich A, Saveliev V, Vyahhi N, Tesler G. QUAST: quality assessment tool for genome
657 assemblies. Bioinformatics. 2013;29(8):1072-5.
658 42. Ewels P, Magnusson M, Lundin S, Kaller M. MultiQC: summarize analysis results for
659 multiple tools and samples in a single report. Bioinformatics. 2016;32(19):3047-8.