## Supplementary Methods:

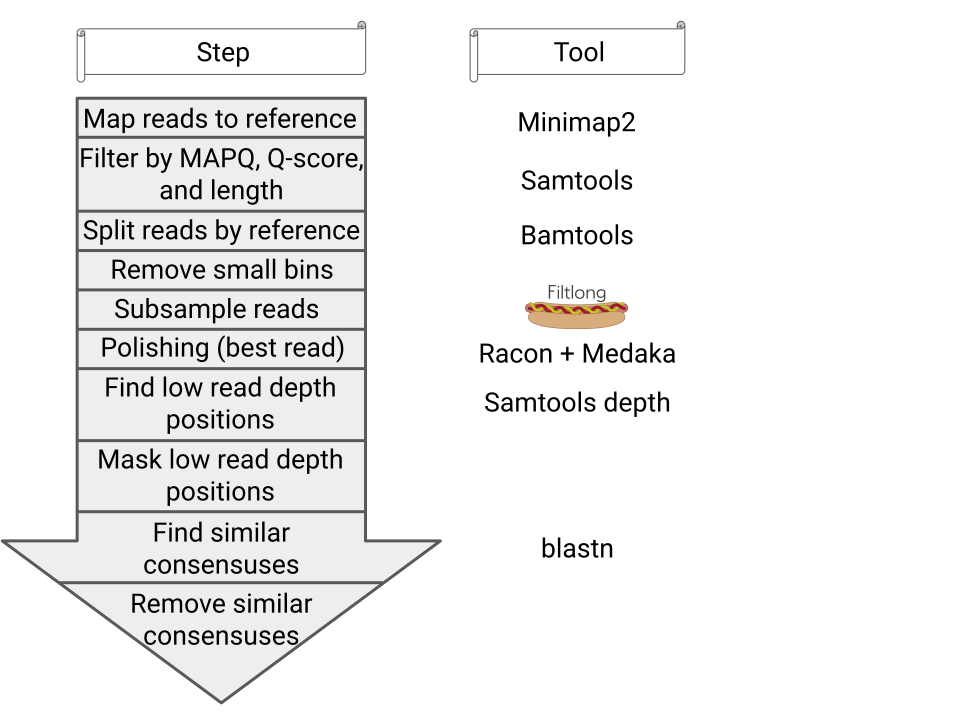
### Consensus sequence assembly form nanopore reads and mixed subtypes:

We mapped our reads to the Cp sequences in our sub-sampled database with minimap2 v2.22 ([H. Li 2018](#111kx3o)) using parameters -ax ont.. Low-quality reads were removed with samtools v1.14 ([H. Li et al. 2009](#3l18frh); [Danecek et al. 2021](#3o7alnk)), using parameters view -q 30, –min-qlen 2000, -e “avg(qual)>=30”, and -F 0x04 . The bam file from samtools was then split into separate bins using bamtools v2.5.1 with parameter -reference <https://github.com/pezmaster31/bamtools>. Each bin contained all the reads that were mapped to a single Cp sequence. We removed bins with fewer than 50 reads or fewer than 0.8% of reads that mapped to any PCV2 Cp sequence. We built consensus genomes from the remaining bins by polishing the best read from each bin with the reads in its bin (See Figure [02](#4d34og8) for a flowchart). The 300 reads with the highest final score from the table output by filtlong v0.2 using parameter –verbose <https://github.com/rrwick/Filtlong> were extracted from their bin with grep using parameters –no-group-separator -A 4. We then polished the read with the highest score using the remaining 299 reads with one round of Racon v1.4.21 using parameters -m 8 -x 6 -g 8 -w 500 <https://github.com/isovic/racon> and one round of Medaka v1.4.3 with the r941\_min\_high model <https://github.com/nanoporetech/medaka>.

We built a database used for consensus assembly, co-infection detection and phylogenetic tree building contained 5862 Cp sequences downloaded from Genbank (Supplemental file: sequences.acc). Cp sequences in our database were aligned with Mafft V7.407 ([Katoh and Standley 2013](#nmf14n)) and manually inspected to remove sequences with early stop codons or incomplete reading frames with Genious v2020.2.1 ([Kearse et al. 2012](#37m2jsg)). We removed recombinant sequences from the inspected Cp sequences with RDP4 v4.101 ([Martin et al. 2015](#1egqt2p)) using settings similar to ([Franzo and Segalés 2018](#32hioqz)). We then removed all gaps and stop codons in our remaining, aligned Cp sequences using Geneious v2020.2.1. Our final database contained 429 Cp sequences (Supplemental file: filtered-sequences.acc).

We detected co-infections between different PCV2 variants by mapping reads to the Cp sequences in our database (Supplementary Methods Figure 01). To make sure most reads from the same variant mapped to a single Cp sequence instead of a group of very similar Cp sequences in our database, we sub-sampled our database with cd-hits using parameter -c 0.98 ([W. Li and Godzik 2006](#206ipza); [Fu et al. 2012](#41mghml)). The sub-sampled database had 66 Cp sequences, with a 98% max similarity between any two Cp sequences.

We mapped our reads to a subset of PCV2 Cp sequences, representing two subtype a, one subtype b, one subtype d, one subtype g, and two subtype f ORF1 (Cp) sequences, with minimap2 v2.22 ([H. Li 2018](#111kx3o)) using parameters -ax ont. Low-quality reads were removed with samtools v1.14 ([H. Li et al. 2009](#3l18frh); [Danecek et al. 2021](#3o7alnk)), using parameters view -q 30, –min-qlen 2000, -e “avg(qual)>=30”, and -F 0x04 . The bam file from samtools was then split into separate bins using bamtools v2.5.1 with parameter -reference <https://github.com/pezmaster31/bamtools>. Each bin contained all the reads that were mapped to a single Cp sequence. We removed bins with fewer than 50 reads or fewer than 0.8% of reads that mapped to any PCV2 Cp sequence.



*Supplementary Methods Figure 01: Steps used to detect co-infections in our pipeline. Samtools fastq was used to convert the bam files from bamtools to fastq files.*

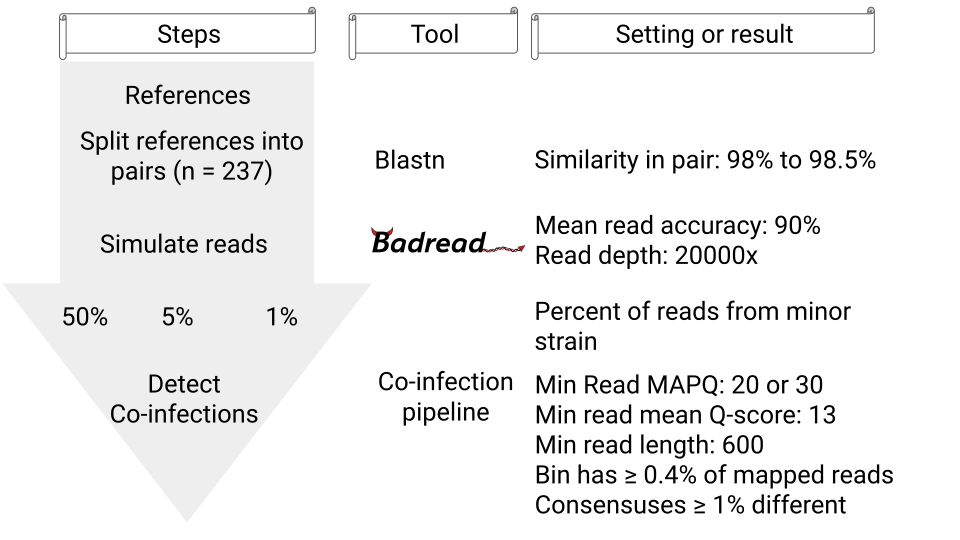
We built consensus genomes from the remaining bins by polishing the best read from each bin with the reads in its bin. The 300 reads with the highest final score from the table output by filtlong v0.2 using parameter –verbose <https://github.com/rrwick/Filtlong> were extracted from their bin with grep using parameters –no-group-separator -A 4. We then polished the read with the highest score using the remaining 299 reads with one round of Racon v1.4.21 using parameters -m 8 -x 6 -g 8 -w 500 <https://github.com/isovic/racon> and one round of Medaka v1.4.3 with the r941\_min\_high model <https://github.com/nanoporetech/medaka>.

We detected and removed consensuses built from miss-binned reads by removing consensuses that were very similar. The number of mismatches and aligned length were found by blasting a consensus genome with blastn against all consensus genomes in a sample. When two or more consensus genomes had fewer than 1.5% mismatches (100 \* number mismatches / aligned length), we kept the consensus genome with the most reads.

We automated the co-infection detection pipeline using bash scripts, available at: [www.github.com/jeremyButtler/binSubtypes](http://www.github.com/jeremyButtler/binSubtypes).

### Read simulation for co-infection simulation:.

We detected Co-infections using the steps mentioned in methods, except we used mapping qualities of 20 and 30, required at least 1% difference between consensuses, did not check for differences in mismatches, and required each bin to have 0.4% or more of all reads mapped to PCV2. For each consensus, we found the number of mismatches and indels by blasting the consensus genome against the reference pair used to simulate its reads.



*Supplementary Methods Figure 02: Steps used to test our co-infection pipeline.*

### Data Analysis:

We made graphs showing how well our pipeline detected co-infections and how accurately our consensus genomes were using ggplot2. We also tested if removing consensuses that had fewer than 100 reads and fewer than 0.3% mismatches improved how well our pipeline detected co-infections. Finally, we made graphs to show the number of indels and mismatches in our consensuses.

## Supplementary figures and tables:

*Supplementary Table 01: Samples that medaka\_variant detected co-infections in. Variant calling with medaka\_variant was done on reads that mapped to PCV2 and a reference genome from one subtype. Each column shows the subtype of the reference used with medaka\_variant to detect co-infections. While the rows show the subtypes that medaka variant detected. Co-infections are shown by subtype-1/subtype-2. Subtypes d1 and d2 were added to distinguish between the two subtype d groups Franzo and Segalés (*[*2018*](#32hioqz)*) detected. Subtypes were found by building a ML tree using RAxML with 1000 bootstraps and the references for PCV2 subtypes a, b, c, d, f, g, and h suggested by Franzo and Segalés (*[*2018*](#32hioqz)*). Samples not shown had single infections from PCV2 subtypes b, d1, or f detected for all references.*

| Sample | subtype a | subtype b | subtype c | subtype d1 | subtype d2 | subtype f | subtype g | subtype h |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Kharkiv 4 | d1 | b/d1 | d1 | d1 | b/d1 | d1 | d1 | d1 |
| Kharkiv 5 | b/d1 | b | b/d1 | d1 | b/d1 | d1 | d1 | d1 |
| Kharkiv 6 | d1 | b | d1 | b/d1 | b/d1 | d1 | b/d1 | d1 |
| Chernivtsi 1 | a/b | b | a/b | a/b | b | b | a/b | a/b |