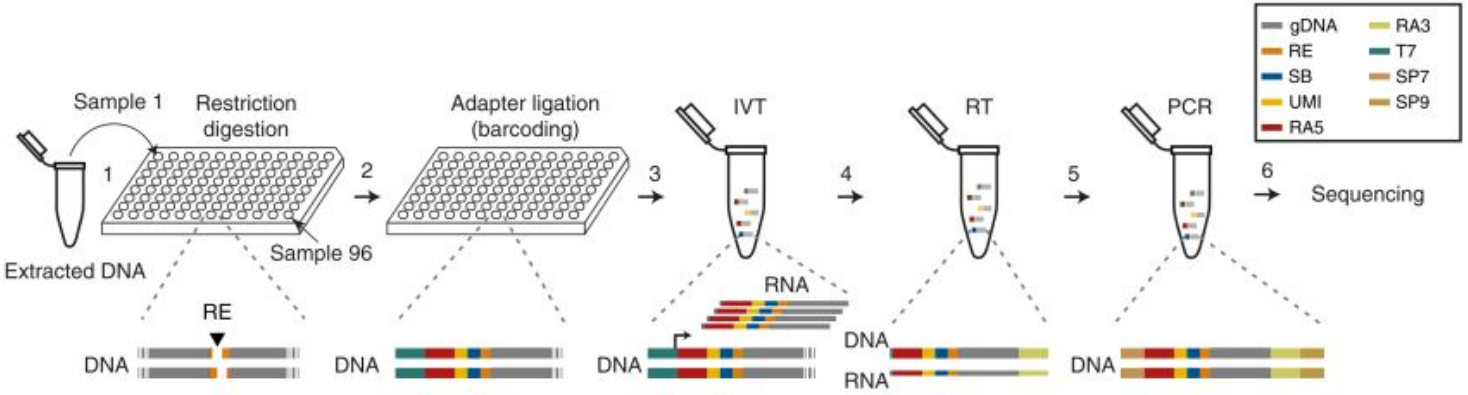
Genomic surveillance of SARS-CoV-2 using COVseq: making it FAIR

Luuk Harbers
2021-12-09

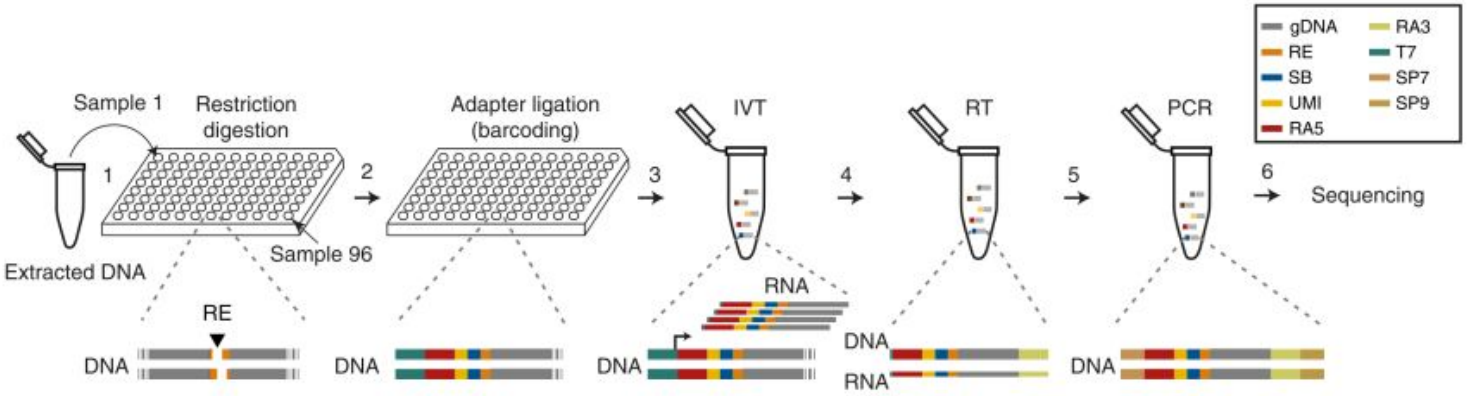
Background

- CUTseq



Background

- CUTseq



- Pandemic
  - Adaptation of CUTseq → COVseq

nature COMMUNICATIONS

ARTICLE

https://doi.org/10.1038/s41467-021-24078-9 OPEN

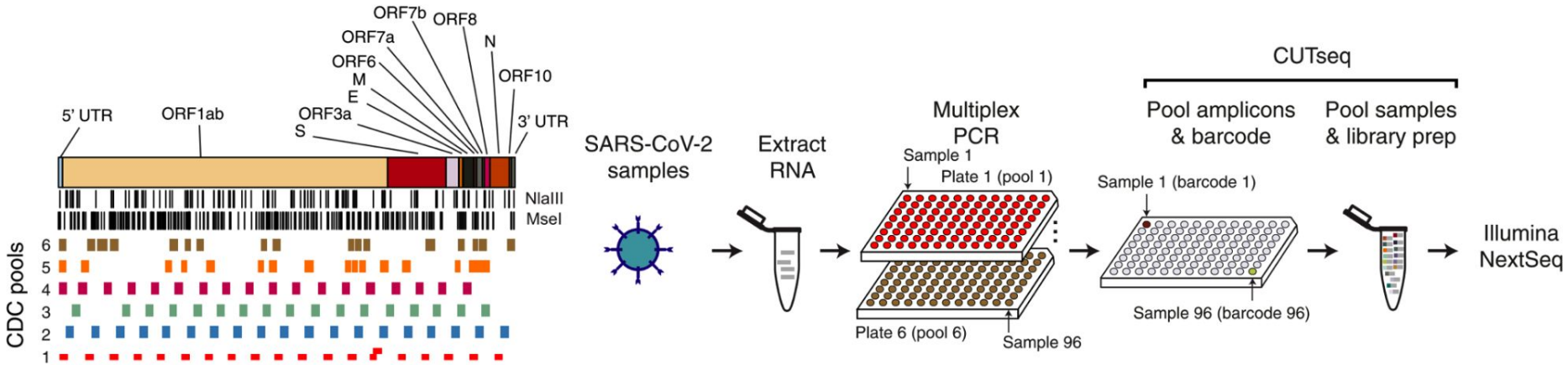
Check for updates

COVseq is a cost-effective workflow for mass-scale SARS-CoV-2 genomic surveillance

Michele Simonetti[1,2,7], Ning Zhang[1,2,3,7], Luuk Harbers[1,2,7], Maria Grazia Milia[4], Silvia Brossa[5], Thi Thu Huong Nguyen[1,2], Francesco Cerutti[4], Enrico Berrino[5,6], Anna Sapino[5,6], Magda Bienko[1,2], Antonino Sottile[5], Valeria Ghisetti[4✉] & Nicola Crosetto[1,2✉]

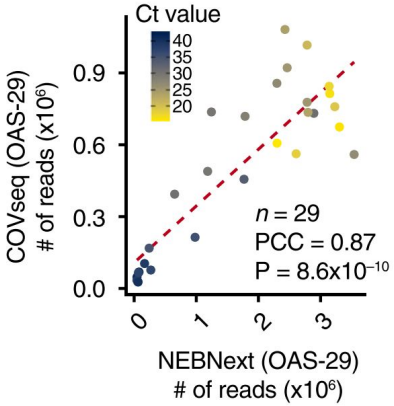
Michele Simonetti, Ning Zhang and Luuk Harbers are equally contributing authors

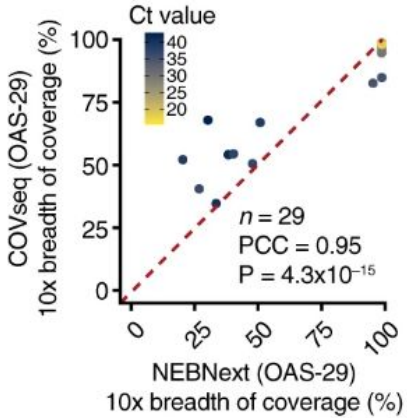
COVseq workflow

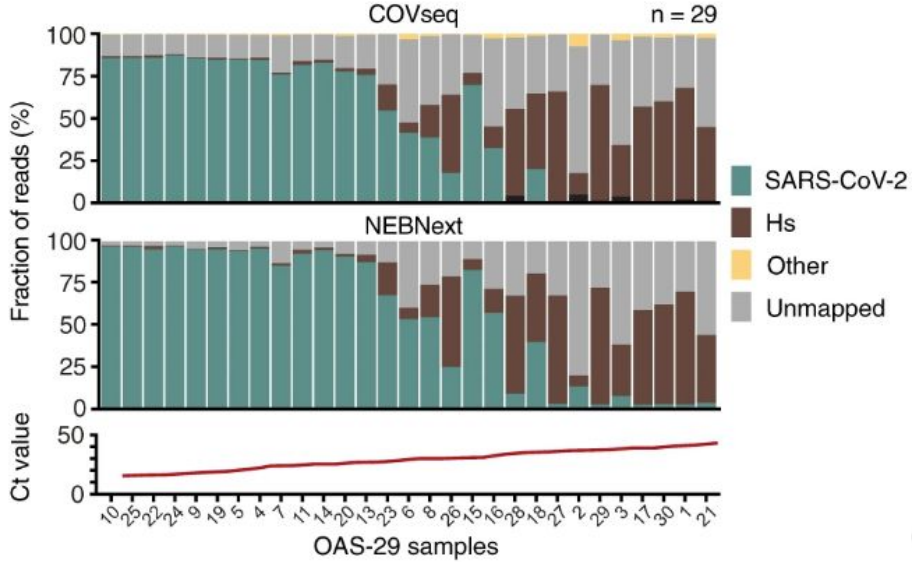


I-DOT One

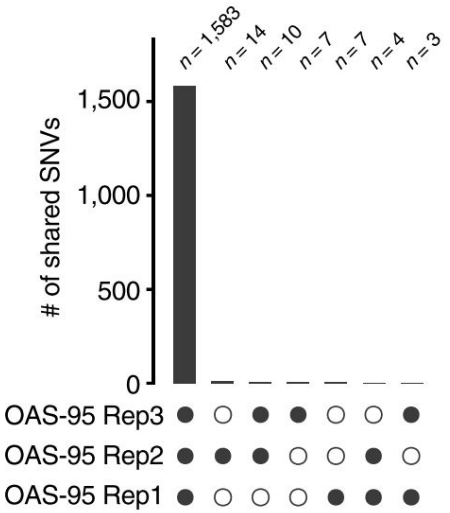
Results

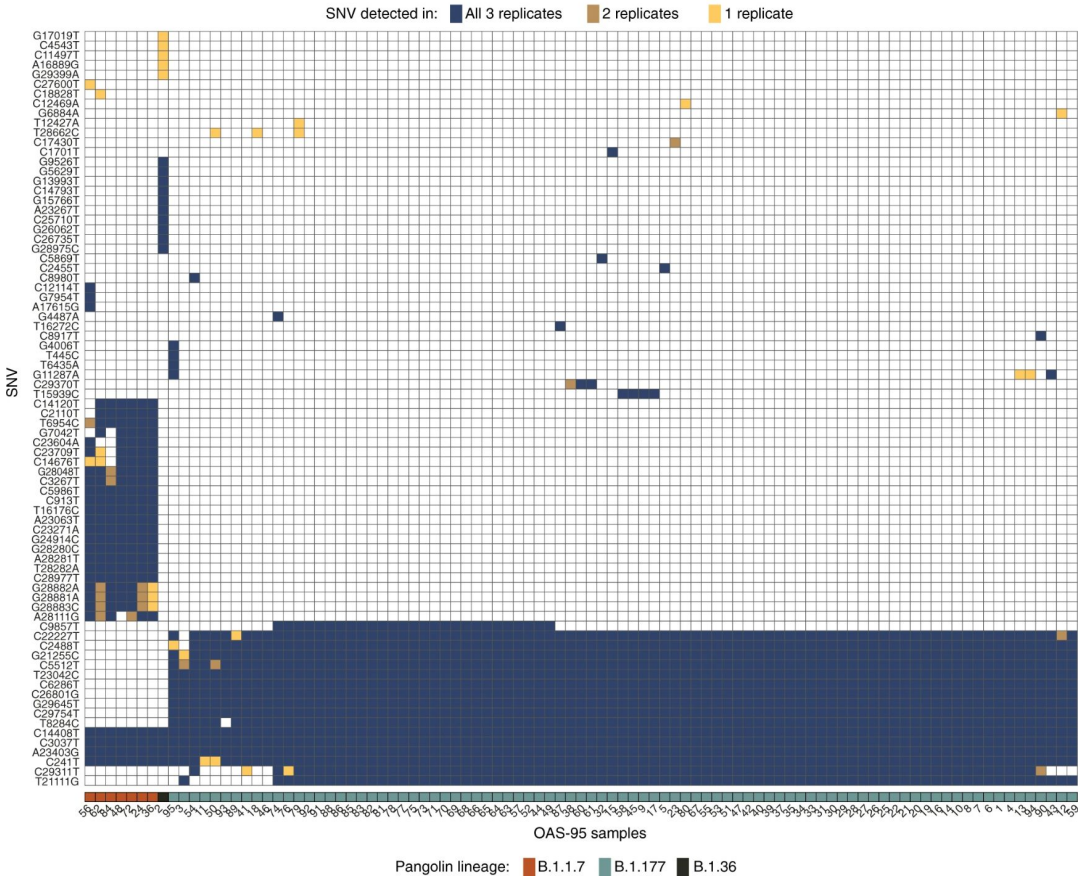

Ct value
40
35
30
25
20
COVseq (OAS-29)
# of reads (x10⁶)
0.9
0.6
0.3
0.0
n = 29
PCC = 0.87
P = 8.6x10⁻¹⁰
0
1
2
3
NEBNext (OAS-29)
# of reads (x10⁶)


Ct value
40
35
30
25
20
COVseq (OAS-29)
10x breadth of coverage (%)
100
75
50
25
0
n = 29
PCC = 0.95
P = 4.3x10⁻¹⁵
0
25
50
75
100
NEBNext (OAS-29)
10x breadth of coverage (%)

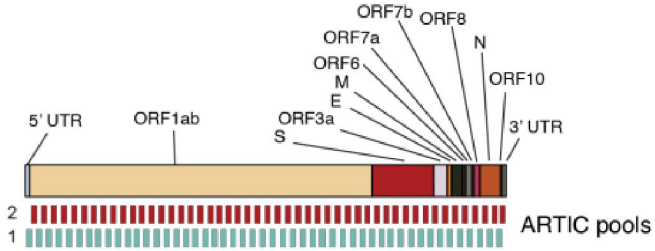

COVseq
n = 29
NEBNext
Fraction of reads (%)
100
75
50
25
0
SARS-CoV-2
Hs
Other
Unmapped
Ct value
50
0
10 25 22 24 9 19 5 4 7 11 14 20 13 23 6 8 26 15 16 28 18 27 2 29 3 17 30 1 21
OAS-29 samples

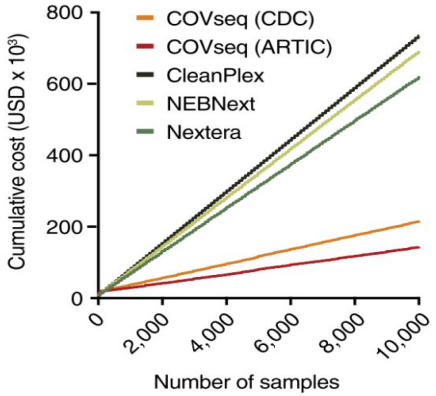
Results

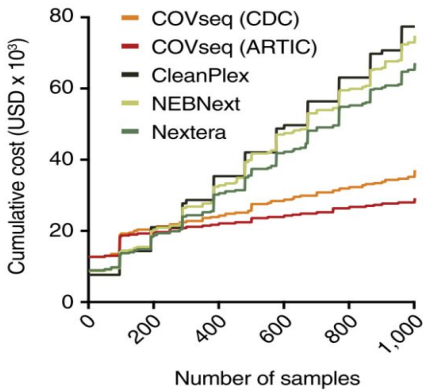

n = 1,583
n = 14
n = 10
n = 7
n = 7
n = 4
n = 3
# of shared SNVs
1,500
1,000
500
0
OAS-95 Rep3
OAS-95 Rep2
OAS-95 Rep1

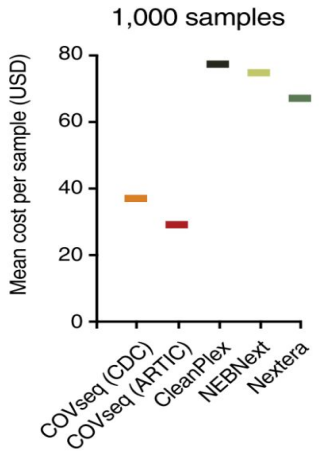

SNV detected in: All 3 replicates 2 replicates 1 replicate
SNV
OAS-95 samples
Pangolin lineage: B.1.1.7 B.1.177 B.1.36

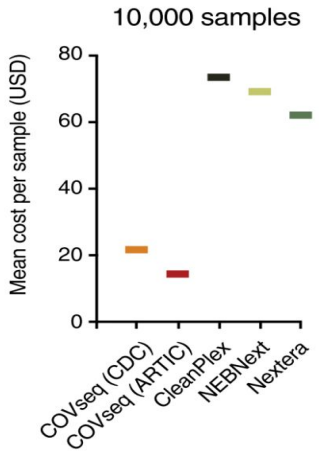
Results


ORF7b
ORF8
ORF7a
N
ORF6
ORF10
M
E
5' UTR
ORF1ab
ORF3a
3' UTR
S
2
1
ARTIC pools


COVseq (CDC)
COVseq (ARTIC)
CleanPlex
NEBNext
Nextera
Cumulative cost (USD x 10³)
Number of samples


COVseq (CDC)
COVseq (ARTIC)
CleanPlex
NEBNext
Nextera
Cumulative cost (USD x 10³)
Number of samples


1,000 samples
Mean cost per sample (USD)
COVseq (CDC)
COVseq (ARTIC)
CleanPlex
NEBNext
Nextera


10,000 samples
Mean cost per sample (USD)
COVseq (CDC)
COVseq (ARTIC)
CleanPlex
NEBNext
Nextera

FAIR principles in COVseq

- **F**indable
- **A**ccessible
- **I**nteroperable
- **R**eusable

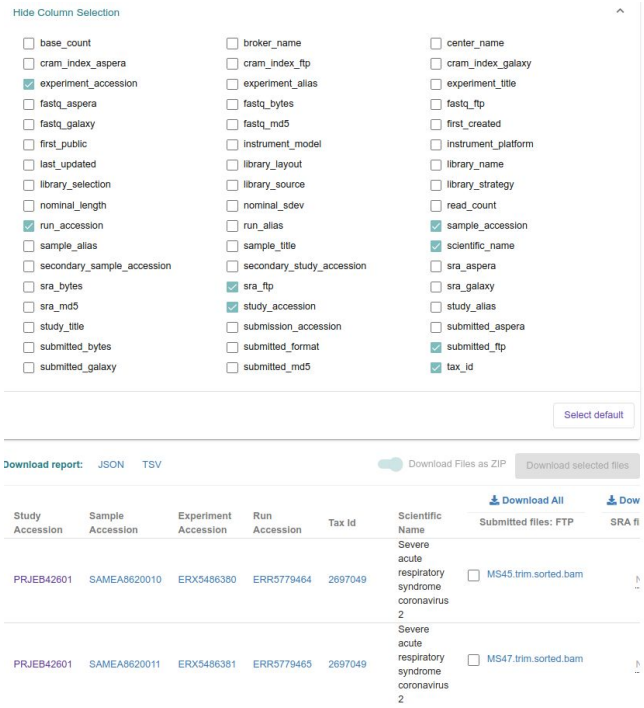
FAIR principles in COVseq

Data availability

The BAM files used to generate all the plots in the main Figures and Supplementary Figures have been deposited to the European Nucleotide Archive (ENA) and are available at the following link: https://www.ebi.ac.uk/ena/browser/view/PRJEB42601. All reference sequences used in this study are listed in Supplementary Table 2. All the GISAID data used in this study are described in Supplementary Data 7 and are available at https://www.gisaid.org.

FAIR principles in COVseq

- 691 bam files from 274 samples
  - Sequencing related metadata
  - Sample related metadata



AA	AB	AC	AD	AE	AF	AG	AH	AI	AJ
geographic location (latitude)	geographic location (longitude)	geographic location (region and locality)	host disease outcome	host age	virus identifier	definition for seropositive sample	serotype (required for seropositive sample)	host habitat	isolation source host-associated
R	R	R	R	R	R	R	R	R	R
52.2053° N	*0.1218° E*	*Hinxton-Cambridgeshire*	*recovered*	*10*		*181 60 IU/L*		*wild*	*kidney cell line vero e6*
DD	DD			years					

FAIR principles in COVseq

Code availability

All the custom code used for processing COVseq sequencing data and the custom MATLAB code used in the Cost Analysis (see Supplementary Notes) is available at https://github.com/ljwharbers/COVseq and the repository is linked to Zenodo at the following link: https://doi.org/10.5281/zenodo.4776499.

FAIR principles in COVseq

- All code available on github with a release on zenodo
- Readme file including extra information with how to run the preprocessing and analyses

README.md

For COVseq libraries make sure all the paths to the required scripts in the config are correct. Build the python/cython library using `$ python setup.py build_ext --inplace`, dependencies for demultiplexing are: pandas, argparse and pysam.

Preparation for preprocessing COVseq libraries

For the demultiplexing of fastq files a custom python script is used. Main input required for this script is the fastq file, a list of barcodes used (no column names, just one barcode per row), the length of the barcode (default 8) and the number of mismatches allowed. To filter out reads that map too far from cutsites you also need a bed file with all the cut site locations in the genome and the read length. For the ivar pipeline the only extra file that is required is the file with the primer locations used for the amplicon sequencing.

Demultiplexing COVseq libraries

Since COVseq libraries are multiplexed libraries (multiple samples in one library). These libraries need to be demultiplexed. To demultiplex COVseq fastq files you can run the script in the `Demultiplex` folder.

Build the python/cython library using `$ python setup.py build_ext --inplace` and make sure the following dependencies are met: `pandas`, `argparse` and `pysam`.

Following this you can run the demultiplexing. An example command would be: `$ demultiplex_withcython.py -f {fastq1} -f2 {fastq2} --paired -o {output} -l {logfile-output} -b {list-of-barcodes} -m {mismatches}` With the {list-of-barcodes} being a text file with 1 COVseq barcode per line (no headers). For more information regarding the different commands you can run `demultiplex_withcython.py --help`.

Running FastQ-Screen

To get information regarding to the mapping percentages of sample specific fastq files (output of demultiplexing step) we used FastQ-Screen (version 0.14.1) https://www.bioinformatics.babraham.ac.uk/projects/fastq_screen/.

Simply add another entry in the database with the SARS-CoV-2 reference genome and run with default settings.

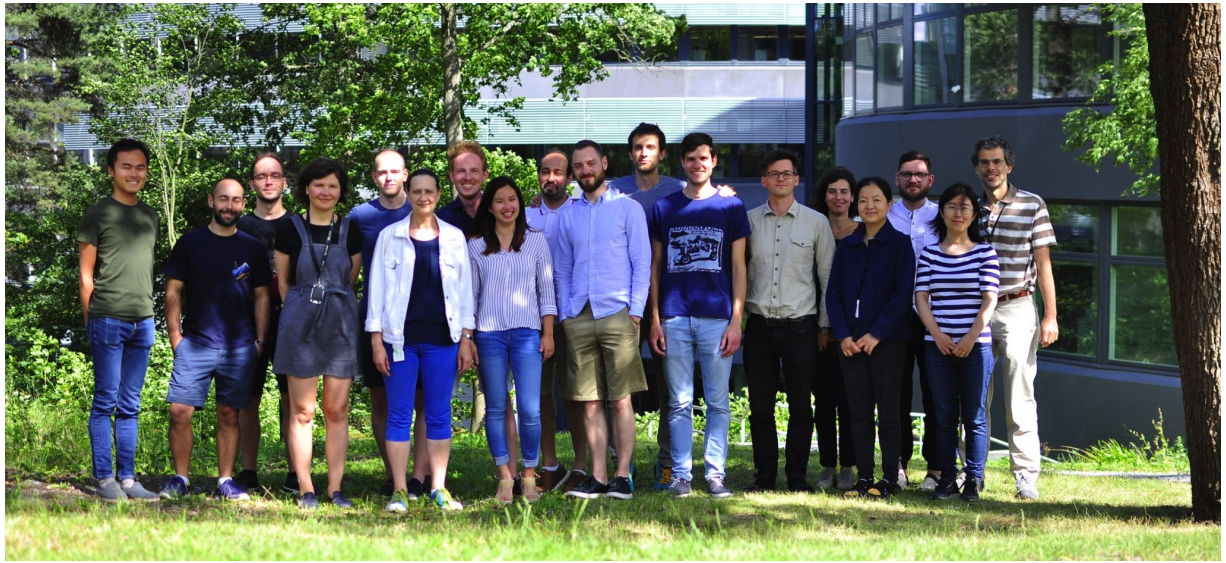
Due to sensitive patient information we do not share any patient specific fastq files.

Running nf-co.re/viralrecon

For further processing of fastq files we used the nextflow based pipeline from nf-core called viralrecon (version 1.1.0) https://nf-co.re/viralrecon/1.1.0. For any extra information or troubleshooting please check out their website and/or join the slack channel for the specific pipeline.

Commands we used to run this pipeline are as follows: `$ nextflow run nf-core/viralrecon --input {samplesheet.csv} --genome 'NC_045512.2' --fasta {sarscov2-fastafile} --save_reference --protocol amplicon --amplicon_bed {ampliconbedfile} --skip_assembly --skip_markduplicates --skip_mosdepth --callers ivar --outdir {outdir} -profile docker --max-cpus 40 -r 1.1.0`

Acknowledgements



BiCro Lab:
Nicola Crosetto
Michele Simonetti
Ning Zhang

Collaborators:
Maria Grazia Milia
Silvia Brossa
Francesco Cerutti
Enrico Berrino

CROSETTO LAB

SciLifeLab

Karolinska Institutet