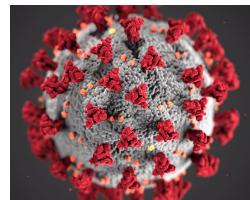


Introduction to viral genome reconstruction using massive sequencing: Sars-cov-2 use case

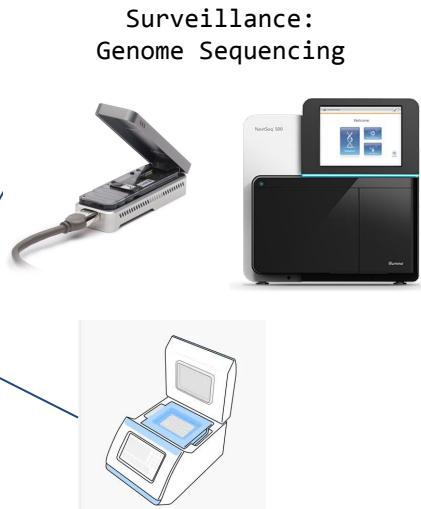
BU-ISCIII
Bioinformatics Unit, Institute of Health Carlos III
Madrid, Spain

Background



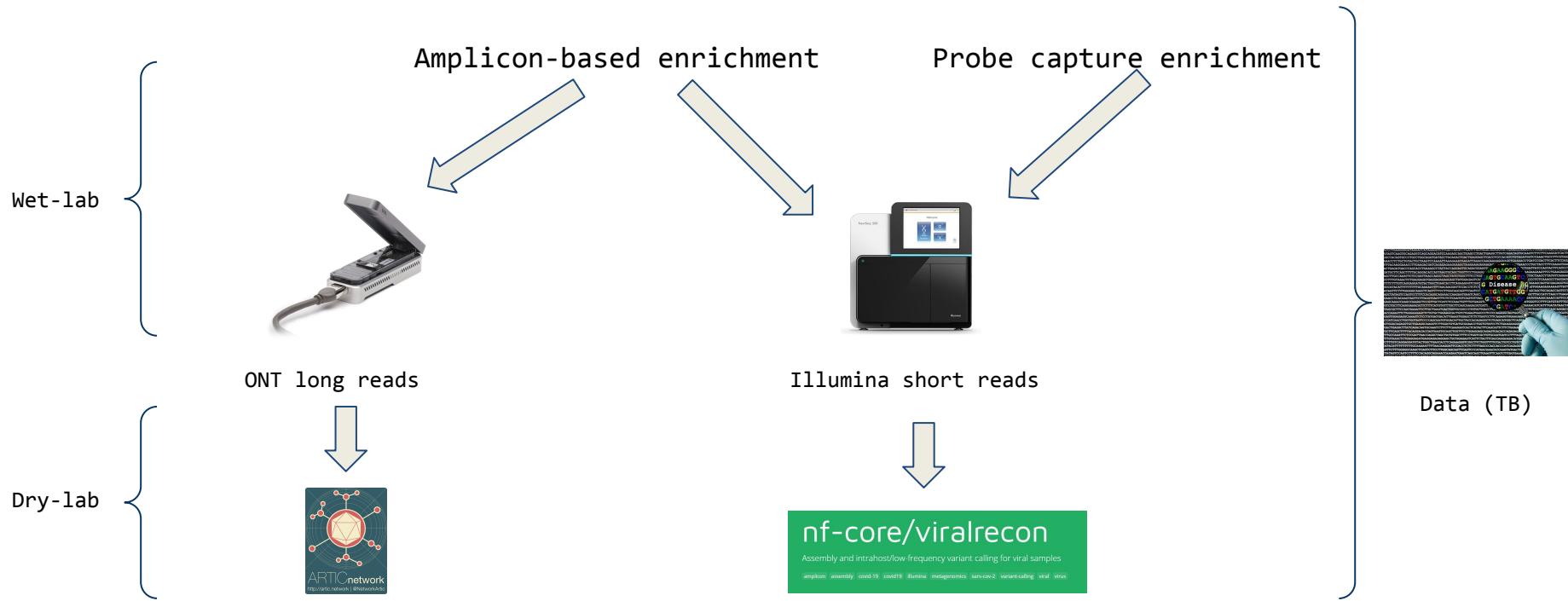
December 2019

SARS-CoV-2



- RT-PCR Primer & Probe efficiency
- Viral variability behaviour
- Correlate viral lineage & severity
- Help vaccine development
- Future established disease control (flu)
- Coordinated & rapid response
- Understand immune response to infection

Sequencing Approaches



State of the art



https://github.com/BU-ISCIII/SARS_Cov2_consensus-nf



<https://github.com/jaleezyy/covid-19-signal>



<https://github.com/nodrogluaP/nanostripper>

- ↓ January 22 2020 Artic Network protocols
- ↓ February 21 2020 Galaxy SARS-CoV2 project
- ↓ March 18 2020 Our 1st pipeline started
- ↓ March 30 2020 nf-core collaboration
- ↓ April 5 2020 1st COVID19 Virtual BioHackathon
- ↓ June 1 2020 viralrecon released v1.0.0
- ↓ June 29 2020 covid-19-signal v1.0.0
- ↓ September 3 2020 nanostripper v1.0.0

Viralrecon

nf-core/ 
viralrecon

<https://github.com/nf-core/viralrecon>



nextflow

Multiple compute infrastructures

Portable

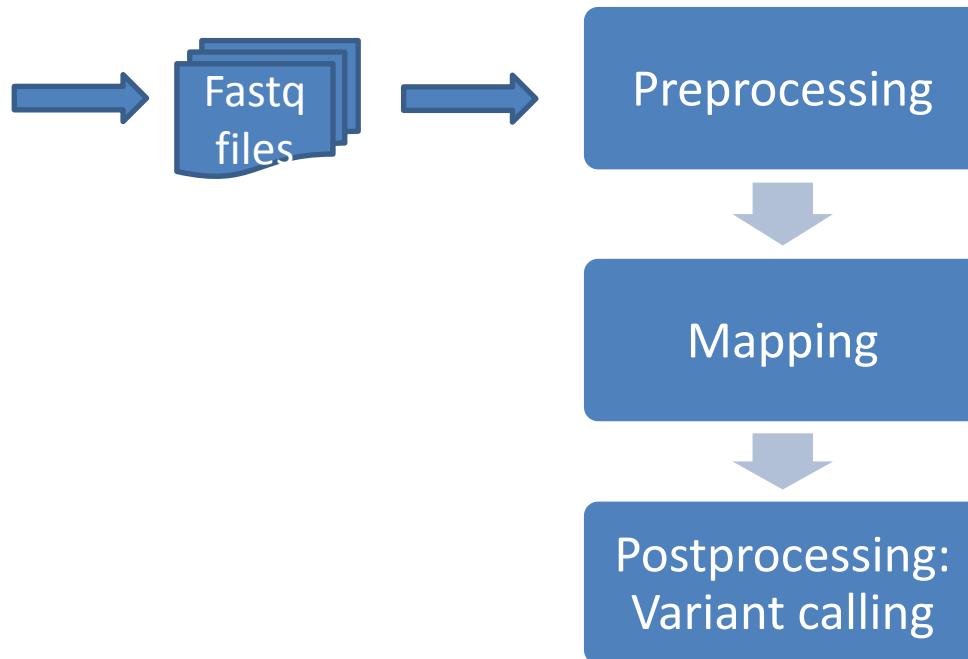
Easy to install

Reproducible

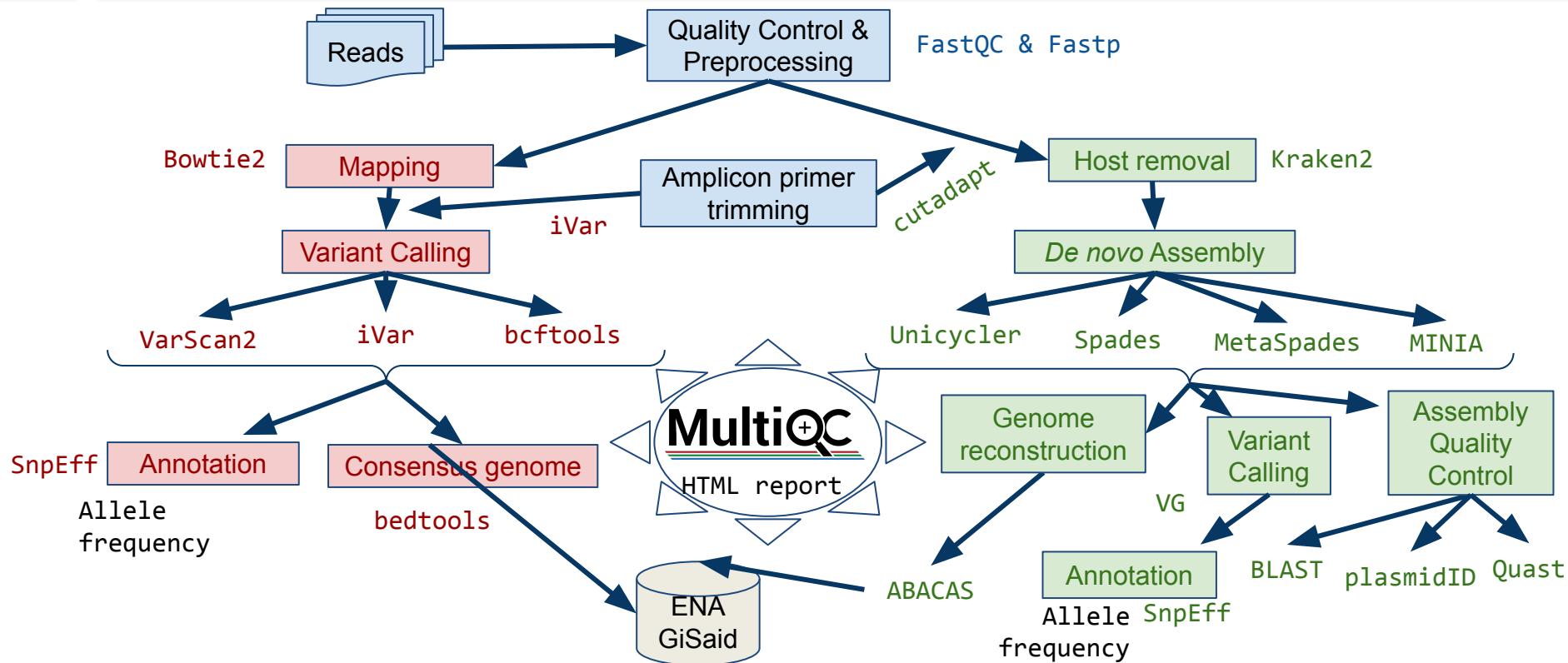
Stable code



Step in the process



Viralrecon



Input data

Which input format do we have from massive sequencing?

FASTQ format

- Is a FASTA file with quality information
- Within HTS, FASTA contain genomes y FASTQ reads

>SEQ_ID

```
AGCTTTCTTGACTGCAACGGGAAATATGTCTCTGTGTGGATTAAAAAAAGAGTGTCTGATAGCAGC  
TTCTGAAGTGGTACCTGCCGTGAGTAAATTAAATTATTGACTTAGGTCACTAAATACTTAAACCAA  
TATAGGCATAGCGCACAGACAGATAAAAATTACAGAGTACACAACATCCATGAAACGCTTAGCACCACC  
ATTACCAACCACCATCACCATTACCAACAGGTAAACGGTGCAGCTGACCGTACAGGAAACACAGAAAAAG
```

@SEQ_ID

```
GATTTGGGTTCAAAGCAGTATCGATCAAATGTAATCCATTGTTCAACTCACAGTTT
```

+

```
! ''*( (( (**+) %%++ ) %% . 1***-+*'') )**55CCF>>>>CCCCCCC65
```

Sequence

Quality: must be 1 bit

Sequencing quality assessment

- To assess quality, software uses **Phred per-base quality score** is used
- Is the **first quality control step** after sequencing. There should be one after every step of the analysis
- After quality assessment user can know how **reliable** are their datasets
- QC will determine the next **filtering** step
- Filtering decisions will **impact** directly in **further analysis**
- Many other steps also use this quality as variable in their **algorithms**

Sequencing quality assessment: Artifacts

HTS methods are bounded by their technical and theoretical limitations and sequencing errors cannot be completely eliminated (Hadigol M, Khiabanian H. 2018)

- **Artifacts in library preparation**

- Remaining adapters
- High rate of duplicates
- GC regions bias
- Polymerase error rate
- DNA damage during breakdown

- **Artifacts during sequencing**

- Low quality in sequence ends(Phasing: cluster loose sync)
- Complication in certain regions:
 - Repetitions
 - Homopolymers
 - High CG content

FastQC: Basic Statistics

- Self defined overall stats
 - Encoding: Phred33 or Phred64

Basic Statistics

Measure	Value
Filename	bad_sequence.txt
File type	Conventional base calls
Encoding	Illumina 1.5
Total Sequences	395288
Sequences flagged as poor quality	0
Sequence length	40
%GC	47

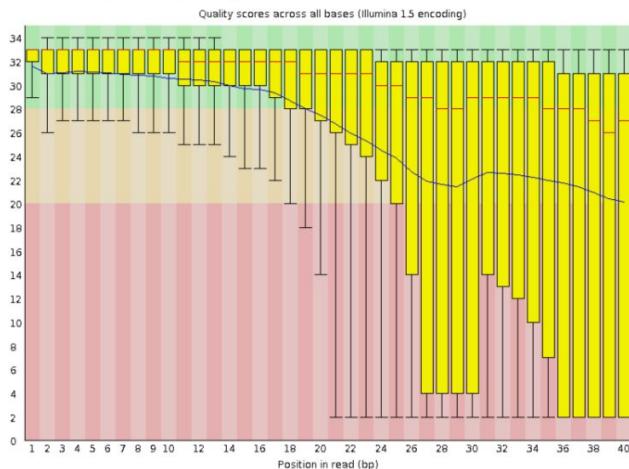
Basic Statistics

Measure	Value
Filename	good_sequence_short.txt
File type	Conventional base calls
Encoding	Illumina 1.5
Total Sequences	250000
Sequences flagged as poor quality	0
Sequence length	40
%GC	45

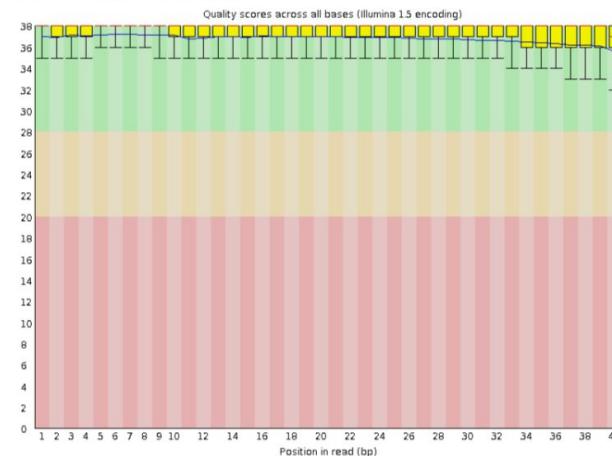
FastQC: Per base sequence quality

- Overview of the range of quality values across all bases at each position in the FastQ file
- Median, inter-quartile range (25-75%), 10-90% points, mean quality

✖ Per base sequence quality



✔ Per base sequence quality



Sequence filtering

- Remove residual adapters
 - Depending on used library
- Filtering parameters
 - Quality filtering
 - Overall mean quality
 - Local mean quality
 - Sequence end
 - Sliding window
 - Size filtering
 - Overall sequence size
 - Remaining sequence size after filtering



Mapping



Mapping software looks for the best match for each read in the genome.

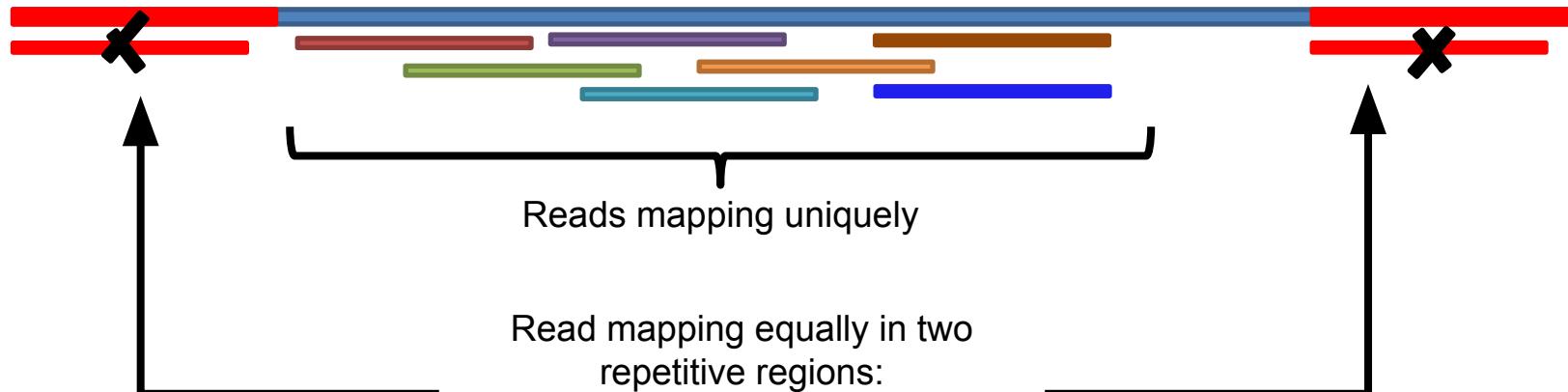
Paired-end reads helps the mapper to find the perfect spot!

Mapping

BOWTIE2

Mapping software looks for the best match for each read in the genome.
Paired-end reads help the mapper to find the perfect spot!

Reference genome



Depends on parameters!!

Input data

Which output format do we have from mapping step?

SAM format

Col	Field	Type	Regexp/Range	Brief description
1	QNAME	String	[!-?A-~]{1,255}	Query template NAME
2	FLAG	Int	[0,2 ¹⁶ -1]	bitwise FLAG
3	RNAME	String	* ![!-()+-<>~-]*	Reference sequence NAME
4	POS	Int	[0,2 ³¹ -1]	1-based leftmost mapping POSition
5	MAPQ	Int	[0,2 ⁸ -1]	MAPping Quality
6	CIGAR	String	* ([0-9]+[MIDNSHPX=])+	CIGAR string
7	RNEXT	String	* = ![!-()+-<>~-]*	Ref. name of the mate/next read
8	PNEXT	Int	[0,2 ³¹ -1]	Position of the mate/next read
9	TLEN	Int	[-2 ³¹ +1,2 ³¹ -1]	observed Template LENgth
10	SEQ	String	* [A-Za-z.=.]+	segment SEQuence
11	QUAL	String	[!-~]+	ASCII of Phred-scaled base QUALity+33

```

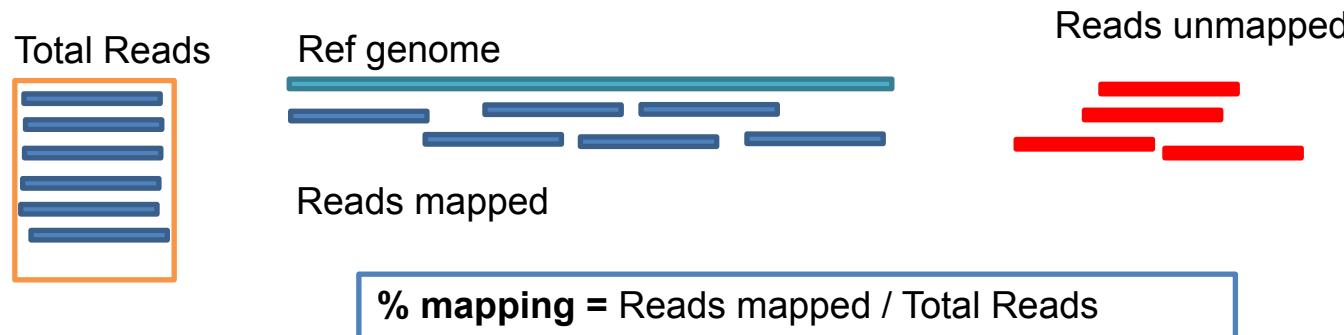
@HD VN:1.5 SO:coordinate
@SQ SN:ref LN:45
r001 99 ref 7 30 8M2I4M1D3M = 37 39 TTAGATAAAGGATACTG *
r002 0 ref 9 30 3S6M1P1I4M * 0 0 AAAAGATAAGGATA *
r003 0 ref 9 30 5S6M * 0 0 GCCTAACGCTAA * SA:Z:ref,29,-,6H5M,17,0;
r004 0 ref 16 30 6M14N5M * 0 0 ATAGCTTCAGC *
r003 2064 ref 29 17 6H5M * 0 0 TAGGC * SA:Z:ref,9,+,5S6M,30,1;
r001 147 ref 37 30 9M = 7 -39 CAGCGGCAT * NM:i:1

```

Mapping quality control

- **% mapping:** number of reads mapping againts reference genome.

Picard
Samtools



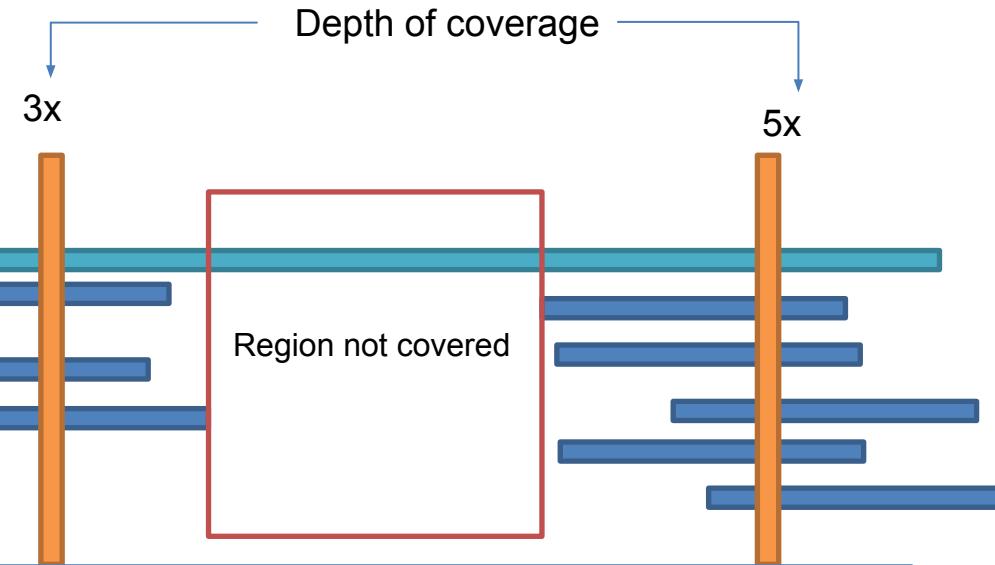
Mandatory parameter for microbial genomics!! It indicates us how many reads we have from our organism of interest. In human genomics this is almost always 99.99% unless something terrible happens. Not here!!!

Mapping quality control

- **% genome > 10x:** percentage of genome covered with more than 10 reads.
- **Mean Depth of coverage:** mean of reads covering a genome position.

Picard
Samtools

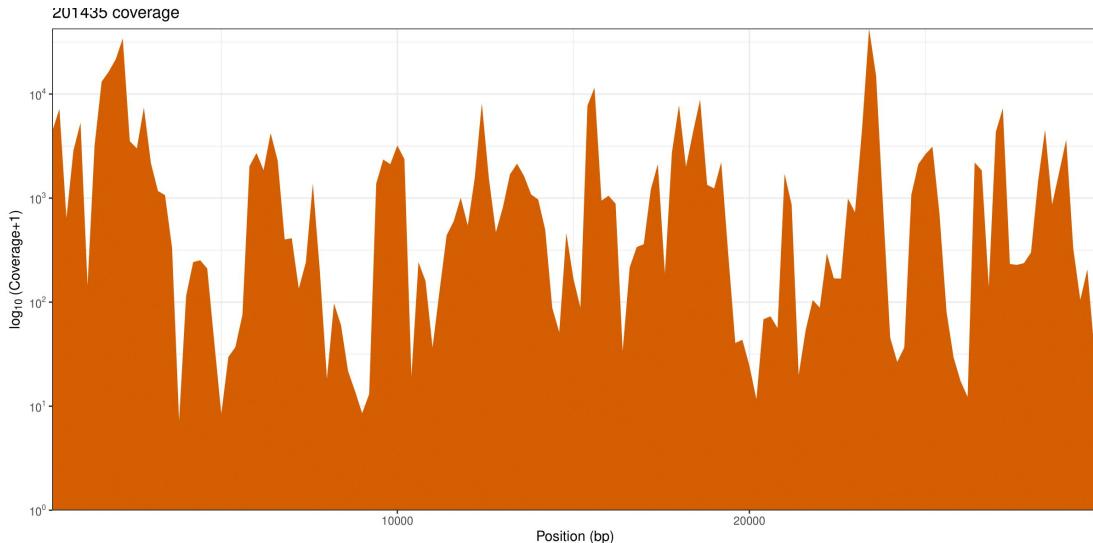
% genome > 10x =
Genome size > 10x / Genome size * 100



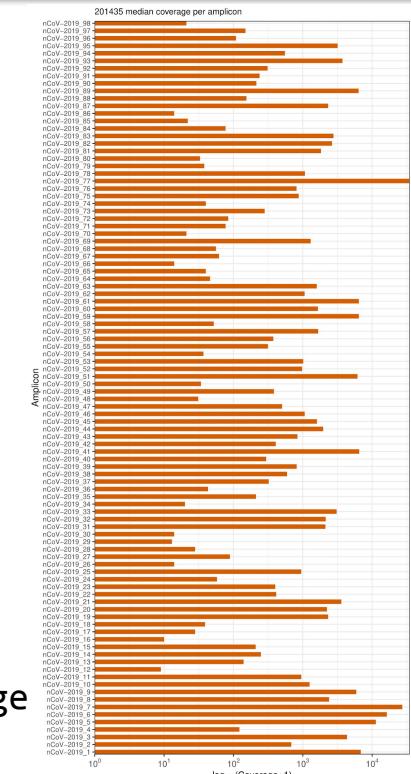
Mean Depth of coverage =
Depth of coverage in position 1, 2, 3 / Genome size

Sars-cov-2 quality control results

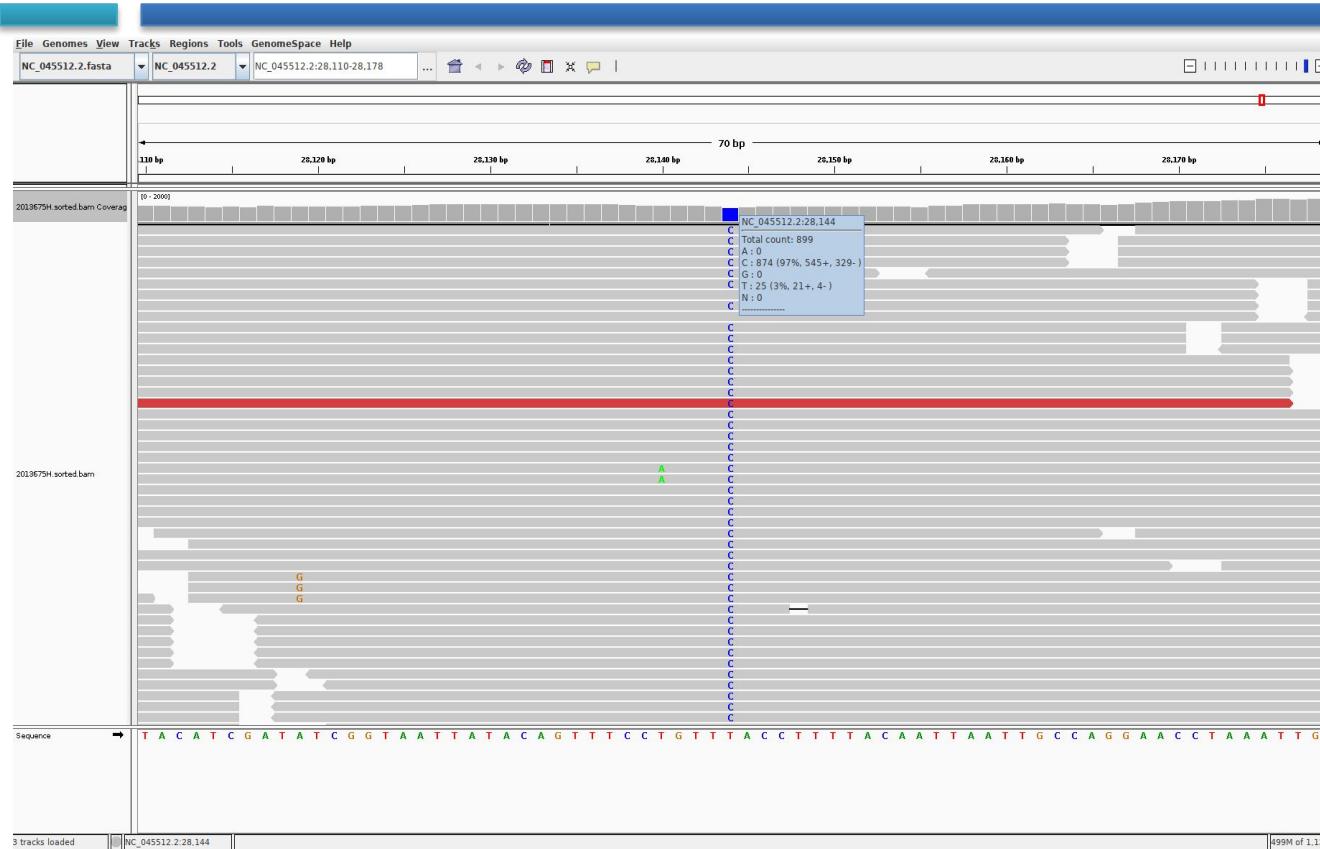
Genome coverage



Amplicon coverage



Variant calling



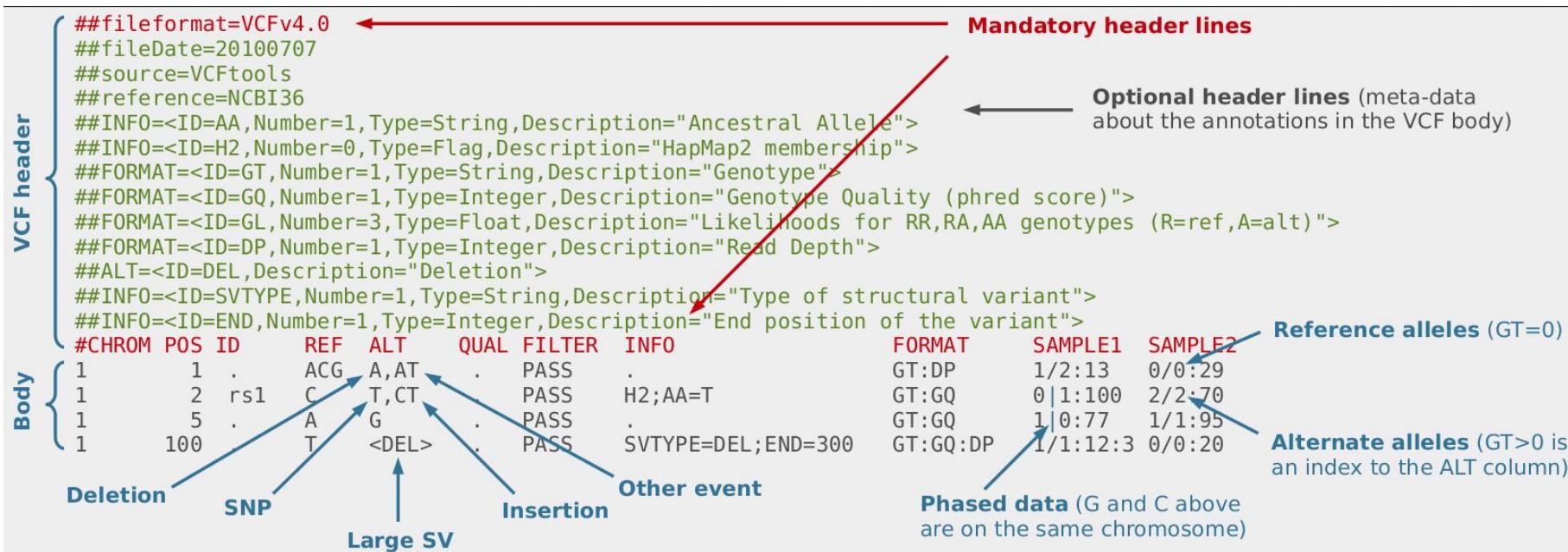
Variant callers walk through each alignment column independently and evaluate if there is a variant.

A variant is called in haploid genomes when it's supported by at least 90% of the reads.

Input data

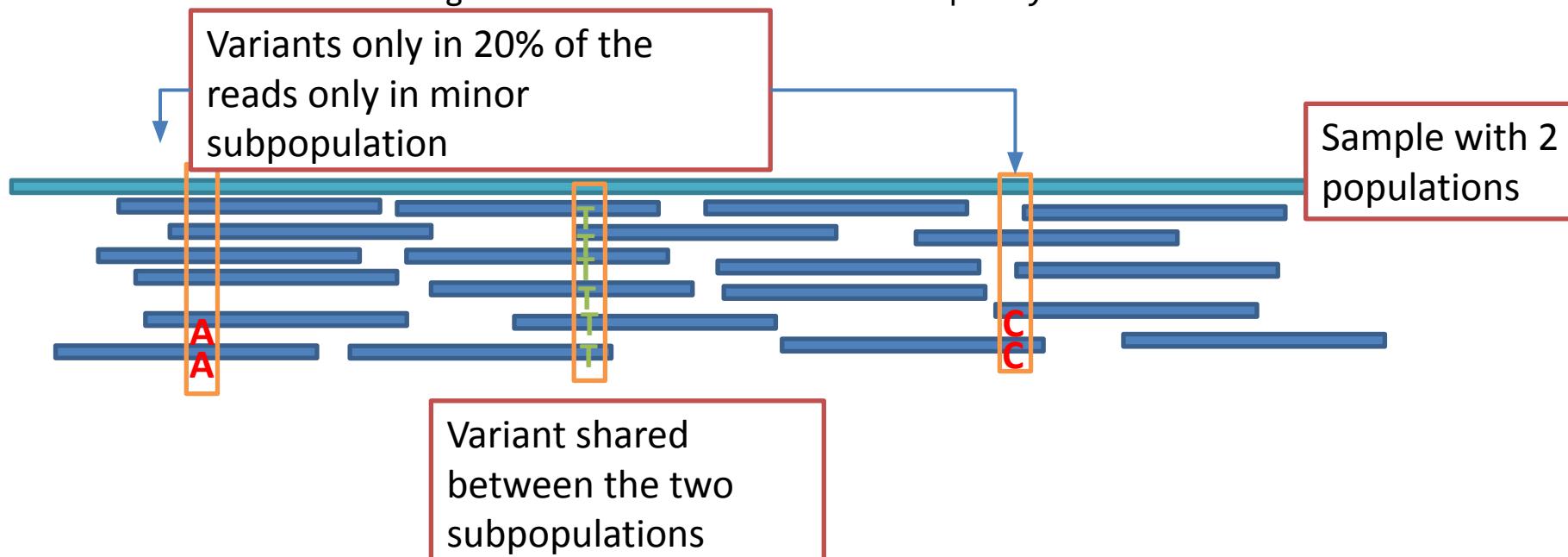
Which output format do we have from variant calling step?

VCF format



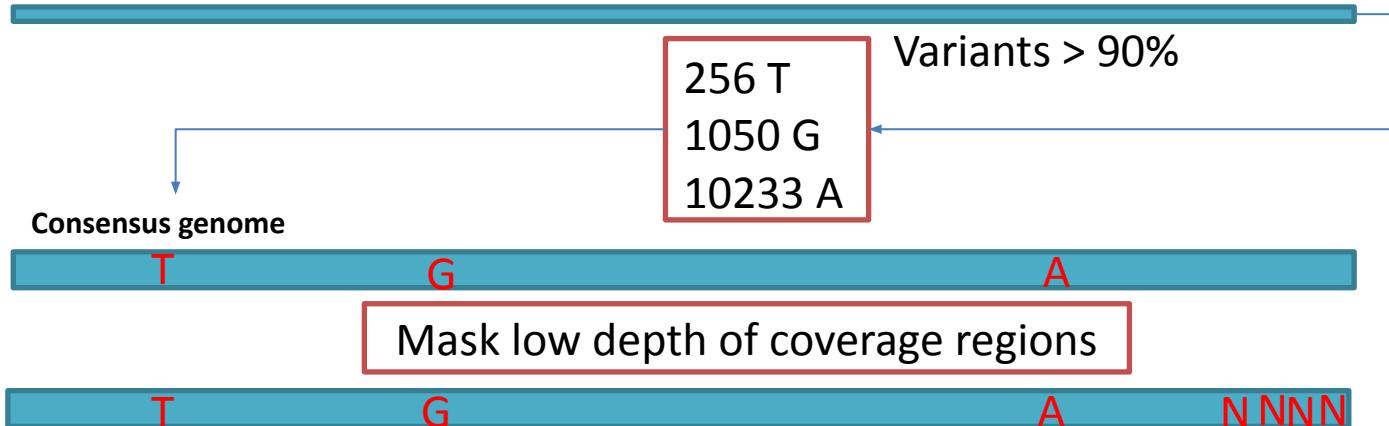
Viral subpopulation - Quasispecies

- Just as in clonal subpopulations in tumor samples, we can have viral subpopulations called quasispecies in viral samples.
- We detect them using the alternative allele frequency.

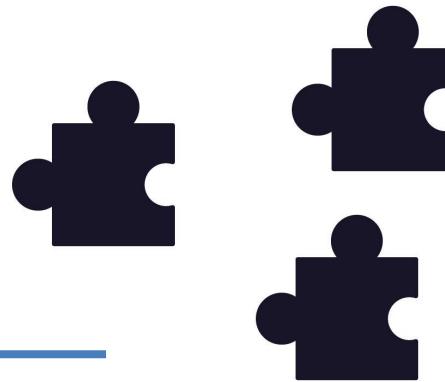
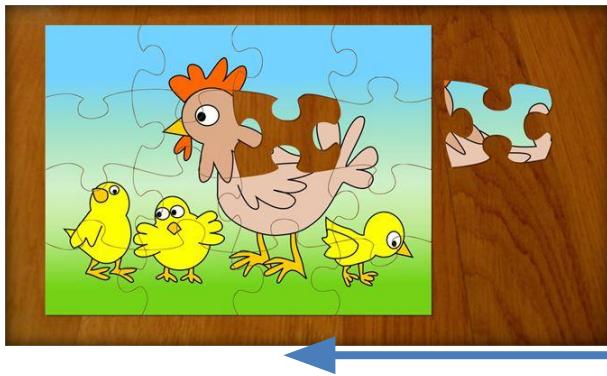


Consensus genome

- Select variants: > 75% allele frequency
- Include variants in reference genome.
- Mask low frequency positions: <10x.

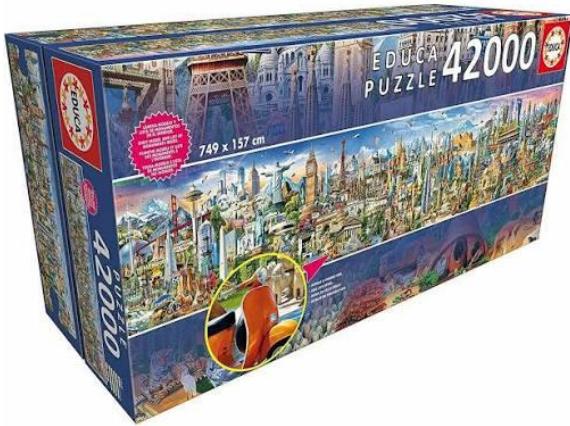


Assembly



The bigger the pieces the easier to reconstruct the puzzle!

Assembly



Obviously this a little bit
more complicated....

We have LOADS of really
little pieces and a big puzzle
in proportion

Assembly

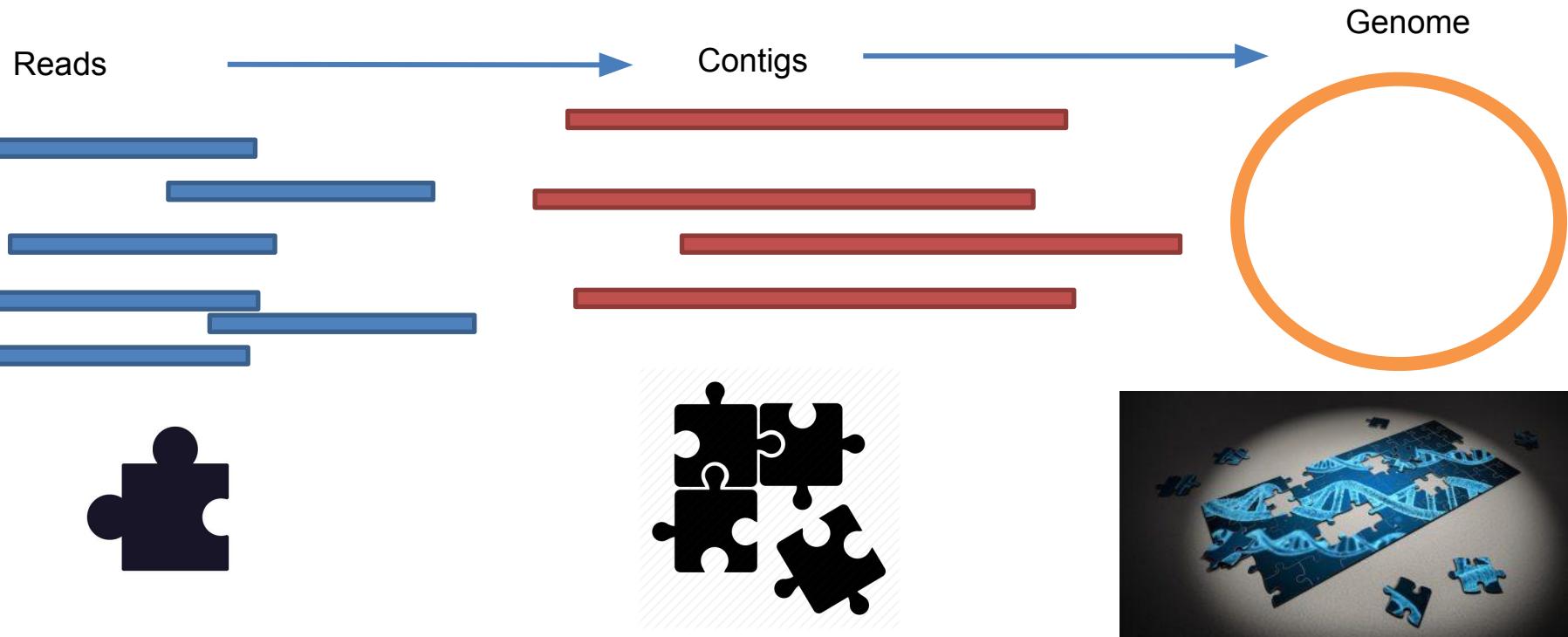


Obviously this a little bit more complicated....



Actually we don't even have the box image most of the time

Assembly



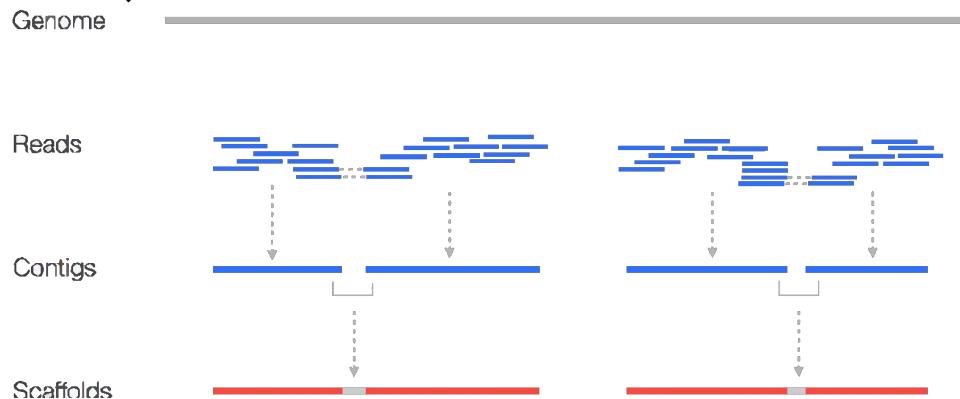
Assembly

Reconstruct a representation of the original DNA from shorter DNA sequences or small fragments known as reads

- ***De novo***: with no previous knowledge of the genome to be assembled. It overlap the end of the end of each read in order to create a longer sequence.
- ***Assembly with reference***: A similar but not identical genome guides the assembly process. Map reads over supplied genome.

Assembly: contig y scaffold

- **Contig:** continuous sequence made up of overlapping shorter sequences
- **Scaffold:** two or more contigs located and rearranged according to spatial information(pair-end, mate pair, reference)



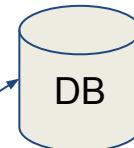
<https://www.biostars.org/p/253222/>

And...after all this?

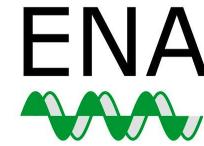
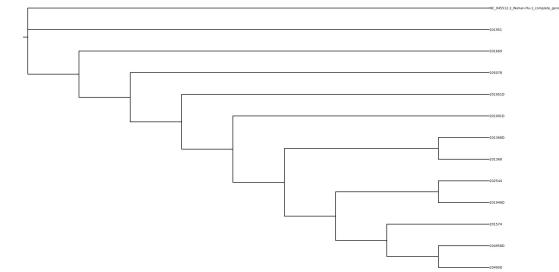
```
>NC_045512.2 Severe acute respiratory syndrome coronavirus 2 isolate Wuhan complete genome
ATTAAGGTTTACCTCC CAGGTAAACAAAC CAACCAACTTC GATCTTGTAGATCT GTTCTTAA
CGAACTTAAATCTGTGGCTGTACTGGCTGATGTTAGTGACTCACGAGTATAATTAAAC
TAATTACTGTCTTGACAGGACAGAGTAACTCGTCTATCTTGCGAGGCTGCTTACGGTTCTGCGT
TTGCAGCCGATCATCAGCACATCTAGGTTCTGCCGGTGTACCGAAAGGTAAGATGGAGAGCTTGTG
CCTGGTTAACAGGAAAACACAGTCCAACCTAGTTGGCTGTGTTACAGGTTGCCGACGCTGTGAC
GTGGCTTGGAGACTCGTGTGGAGGGAGTCTTACAGAGGCACGTCACATCTTAAAGATGGACTCTGTG
CTTAGTGAAGGTTGAAAAGGGCTTGTCTTACCTTAAAGGCTTATGTTCATCAAACGTTGGAT
GCTGCAACTGACTCATCTGGTCATGTTATGTTGAGCTGGTACGAACTCGAAGGGCATTCAGTGGTC
GTAGTGTGAGACACTTGTGCTTGTCTTGTGCTTACAGTGGCTTACCGCAAGGTTCT
TCTTGTAAAGACCGTAAATAAGGAGCTTGTGGCCAAATACAGTGGCTTACCGCAAGGTTCT
GGCGACGACTGCTTGGCACTGTCTTATGAAAGATTTCAGAAAACCTGGACACTAAACATAGCAGTTGG
TTACCGTGAECTATGGTGAECTTAAAGGGGGCATACACTCGCTATGTGATAAACACCTTGTG
CCCTGATGGTACCCCTCTGAGTCATTAAGGACTCTTAGCAGCGTGTGTTAAAGCTTCTGACTTGTG
TCCGAAACACTGGACTTTATGAGACTAAGGGGGTGTATACTGCTGGCGTGAACAT6AGCATGAAATTG
CTTGGTACCGGAAAGCTCTGAAAGAGCTTGAATTTCAGACACCTTGGAAATTAAATTGGAAAAGGA
ATTGACACCTTCAATGGGAATGTCAAATTGTTGATTTCCCTTAAATTCCATAATCAAGACTTTCAA
CCAAGGGTTGAAAAGAAAAGCTTGTGCTTATGGGTAGAATTGATCTGTCTATCCAGTGGCTCAC
```

.fasta consensus genome

Other analyses



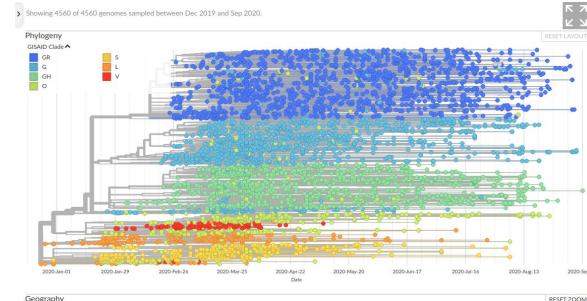
Outbreak detection



European Nucleotide Archive



Clade classification



Conclusions

You can find
viralrecon in:

<https://github.com/nf-core/viralrecon>



Screenshot of the GitHub repository page for nf-core/viralrecon.

Code tab selected. Branch: master. 3 branches. 2 tags.

This branch is 1109 commits ahead of drpateh:master.

Clone with SSH and **Use HTTPS** options. Copy SSH URL button.

Download ZIP button.

Commits table:

File	Message	Date
.github	Merge branch 'dev' into dev	3 months ago
assets	Minor fixes	3 months ago
bin	Update code	3 months ago
conf	Update time	4 months ago
docs	Update docs	3 months ago
.gitattributes	initial template build from nf-core/tools, version 1.9	6 months ago
.gitignore	initial template build from nf-core/tools, version 1.9	6 months ago
CHANGELOG.md	Merge branch 'dev' into dev	3 months ago
CITATIONS.md	Add mosdepth	3 months ago
CODE_OF_CONDUCT.md	initial template build from nf-core/tools, version 1.9	6 months ago
Dockerfile	Bump versions	3 months ago
LICENSE	initial template build from nf-core/tools, version 1.9	6 months ago
README.md	Fix markdownlint	3 months ago
environment.yml	Add biostings	3 months ago
main.nf	Fix naming	3 months ago
nextflow.config	Add --min_mapped_reads param and do some cool stuff with it	3 months ago

About section: Assembly and intrahost/low-frequency variant calling for viral samples.

Tags: nf-core/viralrecon

Topics: viral, metagenomics, amplicon, assembly, variant-calling, illumina, pipeline, workflow, nextflow, nf-core, covid-19, covid19, virus, sars-cov-2

Readme

MIT License

Releases: 2. Latest release: nf-core/viralrecon v1.1.0 - S... on Jun 23. + 1 release

Packages: No packages published. Publish your first package

Languages: Nextflow 70.7%, Python 20.1%, R 7.3%, HTML 1.5%, Dockerfile 0.4%

Thanks for your attention!



(Find us in <https://github.com/BU-ISCIII>)

Special mention to the nf-core
community (<https://nf-co.re/>)