Identification of new SARS-CoV-2 strains using Galaxy

Register and run this workshop on usegalaxy.eu
Workshop adapted from galaxy training workshop material, review for full tutorial.

Questions:

- How can we extract annotated allelic variants in SARS-Cov-2 sequences in Galaxy?
- Which tools and workflows can we use to identify SARS-CoV-2 lineages in Galaxy?

Objectives:

- Repeat SARS-CoV-2 data preparation
- Select and run workflow to extract annotated allelic variants from FASTQ files
- Run workflow to summarize and generate report for previously called allelic variants
- Interpret summaries for annotated allelic variants
- Run workflow to extract consensus sequences
- Select and run tools to assign clades/lineages

Introduction

Effectively monitoring global infectious disease crises, such as the COVID-19 pandemic, requires capacity to generate and analyze large volumes of sequencing data in near real time. These data have proven essential for monitoring the emergence and spread of new variants, and for understanding the evolutionary dynamics of the virus.

Two sequencing platforms (Illumina and Oxford Nanopore) in combination with several established library preparation (Ampliconic and metatranscriptomic) strategies are predominantly used to generate SARS-CoV-2 sequence data. However, data alone do not equal knowledge: they need to be analyzed. The Galaxy community has developed high-quality analysis workflows to support

- sensitive identification of SARS-CoV-2 allelic variants (AVs) starting with allele frequencies as low as 5% from deep sequencing reads
- generation of user-friendly reports for batches of results
- reliable and configurable consensus genome generation from called variants

This tutorial will teach you how to obtain, run and combine these workflows appropriately for different types of input data, be it:

- Single-end data derived from Illumina-based RNAseq experiments
- Paired-end data derived from Illumina-based RNAseq experiments
- Paired-end data generated with Illumina-based Ampliconic (ARTIC) protocols, or
- ONT FASTQ files generated with Oxford nanopore (ONT)-based Ampliconic (ARTIC) protocols

Get sequencing data

Before we can begin any Galaxy analysis, we need to upload the input data: FASTQ files with the sequenced viral RNA from different patients infected with SARS-CoV-2. Several types of data are possible:

- Single-end data derived from Illumina-based RNAseq experiments
- Paired-end data derived from Illumina-based RNAseq experiments
- Paired-end data generated with Illumina-based Ampliconic (ARTIC) protocols
- ONT FASTQ files generated with Oxford nanopore (ONT)-based Ampliconic (ARTIC) protocols

Wwe provide some example datasets (paired-end data generated with Illumina-based Ampliconic (ARTIC) protocols) from COG-UK, the COVID-19 Genomics UK Consortium.

There are several possibilities to upload the data depending on how many datasets you have and what their origin is:

- Import datasets
    - from your local file system,
    - from a given URL or
    - from a shared data library on the Galaxy server you are working on

  and organize the imported data as a dataset collection.
- Import from NCBI's Sequence Read Archive (SRA) at NCBI with the help of a dedicated tool, which will organize the data into collections for you.

As an alternative to uploading the data from a URL or your computer, the files may also have been made available from a *shared data library*:

- Go into **Shared data** (top panel) then **Data libraries**
- Navigate to

  GTN - Material / Variant analysis / Mutation calling, viral genome reconstruction and lineage/clade assignment from SARS-CoV-2 sequencing data / DOI: 10.5281/zenodo.5036686

- Select the desired files
- Click on the **To History** button near the top and select **as Datasets** from the dropdown menu
- In the pop-up window, select the history you want to import the files to (or create a new one)
- Click on **Import**

Choose to import your 2 datasets either using the Shared Library or external server URL.

1. Import **two datasets** from this list for the analysis. You could choose to import more, but this will increase the processing time, which may not complete by the end of this workshop.
2. Create a collection to organize the data

   Click on **Operations on multiple datasets** (check box icon) at the top of the history panel 

- Check all the datasets in your history you would like to include

- Click **For all selected..** and choose **Build List of Dataset Pairs**
    - Change the text of *unpaired forward* to a common selector for the forward reads
    - Change the text of *unpaired reverse* to a common selector for the reverse reads
- Click **Pair these datasets** for each valid *forward* and *reverse* pair.
    - Enter a name for your collection
- Click **Create List** to build your collection
- Click on the checkmark icon at the top of your history again

For the example datasets:

- Since the datasets carry `_1` and `_2` in their names, Galaxy may already have detected a possible pairing scheme for the data, in which case the datasets will appear in green in the lower half (the paired section) of the dialog.

  You could accept this default pairing, but as shown in the middle column of the paired section, this would include the `.fastqsanger` suffix in the pair names (even with `Remove file extensions?` checked Galaxy would only remove the last suffix, `.gz`, from the dataset names.

  It is better to undo the default pairing and specify exactly what we want:

    - at the top of the *paired section*: click `Unpair all`
      This will move all input datasets into the *unpaired section* in the upper half of the dialog.
    - set the text of *unpaired forward* to: `_1.fastqsanger.gz`
    - set the text of *unpaired reverse* to: `_2.fastqsanger.gz`
    - click: `Auto-pair`

  All datasets should be moved to the *paired section* again, but the middle column should now show that only the sample accession numbers will be used as the pair names.

- Make sure *Hide original elements* is checked to obtain a cleaned-up history after building the collection.
- Click *Create Collection*

Import auxiliary datasets

Besides the sequenced reads data, we need at least two additional datasets for calling variants and annotating them:

- the SARS-CoV-2 reference sequence NC_045512.2 to align and compare our sequencing data against
- a tabular dataset defining aliases for viral gene product names, which will let us translate NCBI RefSeq Protein identifiers (used by the SnpEff annotation tool) to the commonly used names of coronavirus proteins and cleavage products.

Another two datasets are needed only for the analysis of ampliconic, e.g., ARTIC-amplified, input data:

- a BED file specifying the primers used during amplification and their binding sites on the viral genome
- a custom tabular file describing the amplicon grouping of the primers

Import the auxiliary datasets:

- the SARS-CoV-2 reference (`NC_045512.2_reference.fasta`)

- gene product name aliases (`NC_045512.2_feature_mapping.tsv`)
- ARTIC v3 primer scheme (`ARTIC_nCoV-2019_v3.bed`)
- ARTIC v3 primer amplicon grouping info (`ARTIC_amplicon_info_v3.tsv`)

From the shared data library

As an alternative to uploading the data from a URL or your computer, the files may also have been made available from a *shared data library*:

- Go into **Shared data** (top panel) then **Data libraries**
- Navigate to
  GTN - Material / Variant analysis / Mutation calling, viral genome reconstruction and lineage/clade assignment from SARS-CoV-2 sequencing data / DOI: 10.5281/zenodo.5036686 or the correct folder as indicated by your instructor
- Select the desired files
- Click on the **To History** button near the top and select **as Datasets** from the dropdown menu
- In the pop-up window, select the history you want to import the files to (or create a new one)
- Click on **Import**

Changing Data Types

Click on the galaxy-pencil **pencil icon** for the dataset to edit its attributes

- In the central panel, click on the galaxy-chart-select-data **Datatypes** tab on the top
- Select your desired datatype
- Click the **Save** button

From FASTQ to annotated allelic variants

To identify the SARS-CoV-2 allelic variants (AVs), a first workflow converts the FASTQ files to annotated AVs through a series of steps that include quality control, trimming, mapping, deduplication, AV calling, and filtering.

Four versions of this workflow are available with their tools and parameters optimized for different types of input data as outlined in the following table:

Workflow version	Input data	Read aligner	Variant caller
Illumina RNAseq SE	Single-end data derived from RNAseq experiments	**bowtie2** Langmead and Salzberg 2012	**lofreq** Wilm *et al.* 2012
Illumina RNAseq PE	Paired-end data derived from RNAseq experiments	**bwa-mem** Li and Durbin 2010	**lofreq** Wilm *et al.* 2012
Illumina ARTIC	Paired-end data generated with Illumina-based Ampliconic (ARTIC) protocols	**bwa-mem** Li and Durbin 2010	**lofreq** Wilm *et al.* 2012
ONT ARTIC	ONT FASTQ files generated with Oxford nanopore (ONT)-based Ampliconic (ARTIC) protocols	**minimap2** Li 2018	**medaka**

About the workflows

The two Illumina RNASeq workflows (Illumina RNAseq SE and Illumina RNAseq PE) perform read mapping with **bwa-mem** and **bowtie2**, respectively, followed by sensitive allelic-variant (AV) calling across a wide range of AFs with **lofreq**.

- The workflow for Illumina-based ARTIC data (Illumina ARTIC) builds on the RNASeq workflow for paired-end data using the same steps for mapping (**bwa-mem**) and AV calling (**lofreq**), but adds extra logic operators for trimming ARTIC primer sequences off reads with the **ivar** package. In addition, this workflow uses **ivar** also to identify amplicons affected by ARTIC primer-binding site mutations and excludes reads derived from such “tainted” amplicons when calculating alternative allele frequences (AFs) of other AVs.
- The workflow for ONT-sequenced ARTIC data (ONT ARTIC) is modeled after the alignment/AV-calling steps of the ARTIC pipeline. It performs, essentially, the same steps as that pipeline’s minion command, i.e. read mapping with **minimap2** and AV calling with **medaka**. Like the Illumina ARTIC workflow it uses **ivar** for primer trimming. Since ONT-sequenced reads have a much higher error rate than Illumina-sequenced reads and are therefore plagued more by false-positive AV calls, this workflow makes no attempt to handle amplicons affected by potential primer-binding site mutations.

All four workflows use **SnpEff**, specifically its 4.5covid19 version, for AV annotation.

Workflows default to requiring an AF ≥ 0.05 and AV-supporting reads of ≥ 10 (these and all other parameters can be easily changed by the user). For an AV to be listed in the reports, it must surpass these thresholds in at

least one sample of the respective dataset. We estimate that for AV calls with an AF ≥ 0.05, our analyses have a false-positive rate of < 15% for both Illumina RNAseq and Illumina ARTIC data, while the true-positive rate of calling such low-frequency AVs is ~80% and approaches 100% for AVs with an AF ≥ 0.15. This estimate is based on an initial application of the Illumina RNAseq and Illumina ARTIC workflows to two samples for which data of both types had been obtained at the virology department of the University of Freiburg and the assumption that AVs supported by both sets of sequencing data are true AVs. The second threshold of 10 AV-supporting reads is applied to ensure that calculated AFs are sufficiently precise for all AVs.

More details about the workflows, including benchmarking of the tools, can be found on covid19.galaxyproject.org

For this workshop, we will be using the Illumina ARCTIC PE workflow.

Get the workflow for your data into Galaxy

- Option 1: Find workflows on the WorkflowHub and run them directly on usegalaxy.eu

  Please note that this option currently works *only* with usegalaxy.eu!

  - Open the workflow page on the WorkflowHub
    - **Illumina ARTIC PE - The one to use for example datasets**
    - Illumina RNAseq SE
    - Illumina RNAseq PE
    - ONT ARTIC
  - Click on `Run on usegalaxy.eu` at the top right of the page

    The browser will open a new tab with Galaxy's workflow invocation interface.

Run **COVID-19: variation analysis on …** workflow using the following parameters:

Send results to a new history": `No`

- For **Illumina ARTIC PE** workflow (named **COVID-19: variation analysis on ARTIC PE data**), *to use for example datasets*
- param-file *"1: ARTIC primers to amplicon assignments"*: `ARTIC_amplicon_info_v3.tsv` or `ARTIC amplicon info v3`
- param-file *"2: ARTIC primer BED"*: `ARTIC_nCoV-2019_v3.bed` or `ARTIC nCoV-2019 v3`
- param-file *"3: FASTA sequence of SARS-CoV-2"*: `NC_045512.2_reference.fasta` or `NC_045512.2 reference sequence`
- param-collection *"4: Paired Collection (fastqsanger) - A paired collection of fastq datasets to call variant from"*: paired collection created for the input datasets

The execution of the workflow takes some time. It is possible to launch the next step even if it is not done, as long as all steps are successfully scheduled.

From annotated AVs per sample to AV summary

Once the jobs of previous workflows are done, we identified AVs for each sample. We can run a “Reporting workflow” on them to generate a final AV summary.

This workflow takes the collection of called (with lofreq) and annotated (with SnpEff) variants (one VCF dataset per input sample) that got generated as one of the outputs of any of the four variation analysis workflows above, and generates two tabular reports and an overview plot summarizing all the variant information for your batch of samples.

Use the correct collection of variants!!

The variation analysis workflow should have generated *two* collections of annotated variants - one called `Final (SnpEff-) annotated variants`, the other one called `Final (SnpEff-) annotated variants with strand-bias soft filter applied`.

We analyzed ampliconic data with any of the **variation analysis of ARTIC** data workflows, then please consider the strand-bias soft-filtered collection experimental and proceed with the `Final (SnpEff-) annotated variants` collection as input to the next workflow.

Get the workflow into Galaxy

- Option 1: Find workflows on the WorkflowHub and run them directly on usegalaxy.eu

  Please note that this option currently works *only* with usegalaxy.eu!

  - Open the workflow page on WokflowHub
  - Click on `Run on usegalaxy.eu` on the top right of the page

    The browser will open a new tab with Galaxy’s workflow invocation interface.

Run **COVID-19: variation analysis reporting** workflow using the following parameters:

“Send results to a new history”: `No`

- *“1: AF Filter - Allele Frequency Filter”*: `0.05`

  This number is the minimum allele frequency required for variants to be included in the report.

- *“2: DP Filer”*: `1`

  The minimum depth of all alignments required at a variant site; the suggested value will, effectively, deactivate filtering on overall DP and will result in the DP_ALT Filter to be used as the only coverage-based filter.

- *“3: DP_ALT Filter”*: `10`

  The minimum depth of alignments at a site that need to support the respective variant allele

- *“4: Variation data to report”*: `Final (SnpEff-) annotated variants`

The collection with variation data in VCF format: the output of the previous workflow

- *"4: gene products translations"*: `NC_045512.2_feature_mapping.tsv` or `NC_045512.2 feature mapping`

  The custom tabular file mapping NCBI RefSeq Protein identifiers (as used by snpEff version 4.5covid19) to their commonly used names, part of the auxillary data; the names in the second column of this dataset are the ones that will appear in the reports generated by this workflow.

- *"5: Number of Clusters"*: `1 (if we were using the entire dataset, 3)`

  The variant frequency plot generated by the workflow will separate the samples into this number of clusters.

The three key results datasets produced by the Reporting workflow are:

1. Combined Variant Report by Sample: This table combines the key statistics for each AV call in each sample. Each line in the dataset represents one AV detected in one specific sample

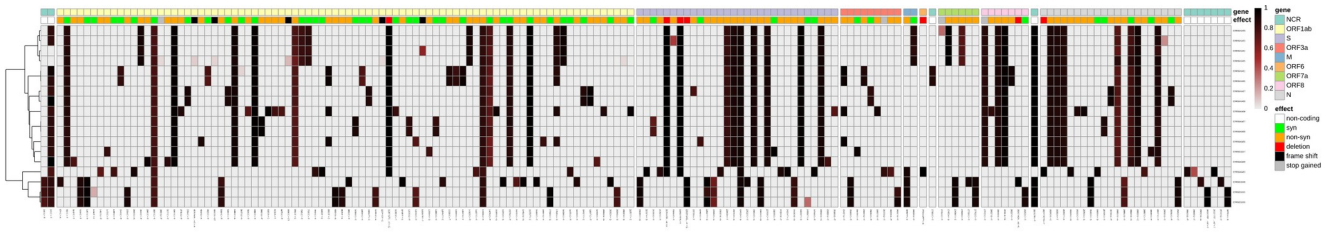
Column	Field	Meaning
1	`Sample`	SRA run ID
2	`POS`	Position in NC_045512.2
3	`FILTER`	`Filter` field from VCF
4	`REF`	Reference base
5	`ALT`	Alternative base
6	`DP`	Sequencing depth
7	`AF`	Alternative allele frequency
8	`SB`	Strand bias P-value from Fisher's exact test calculated by `lofreq`
9	`DP4`	Depth for Forward Ref Counts, Reverse Ref Counts, Forward Alt Counts, Reverse Alt Counts
10	`IMPACT`	Functional impact (from SNPEff)
11	`FUNCLASS`	Funclass for change (from SNPEff)
12	`EFFECT`	Effect of change (from SNPEff)
13	`GENE`	Gene name
14	`CODON`	Codon
15	`AA`	Amino acid
16	`TRID`	Short name for the gene
17	`min(AF)`	Minimum Alternative Allele Freq across all samples containing this change
18	`max(AF)`	Maximum Alternative Allele Freq across all samples containing this change
19	`countunique(change)`	Number of distinct types of changes at this site across all samples
20	`countunique(FUNCLASS)`	Number of distinct FUNCLASS values at this site across all samples

Column	Field	Meaning
21	change	Change at this site in this sample

2. Combined Variant Report by Variant: This table combines the information about each AV *across* samples.

Column	Field	Meaning
1	POS	Position in NC_045512.2
2	REF	Reference base
3	ALT	Alternative base
4	IMPACT	Functional impact (from SnpEff)
5	FUNCLASS	Funclass for change (from SnpEff)
6	EFFECT	Effect of change (from SnpEff)
7	GENE	Gene
8	CODON	Codon
9	AA	Amino acid
10	TRID	Short name for the gene (from the feature mapping dataset)
11	countunique(Sample)	Number of distinct samples containing this change
12	min(AF)	Minimum Alternative Allele Freq across all samples containing this change
13	max(AF)	Maximum Alternative Allele Freq across all samples containing this change
14	SAMPLES(above-thresholds)	List of distinct samples where this change has frequency abobe threshold (5%)
15	SAMPLES(all)	List of distinct samples containing this change at any frequency (including below threshold)
16	AFs(all)	List of all allele frequencies across all samples
17	change	Change

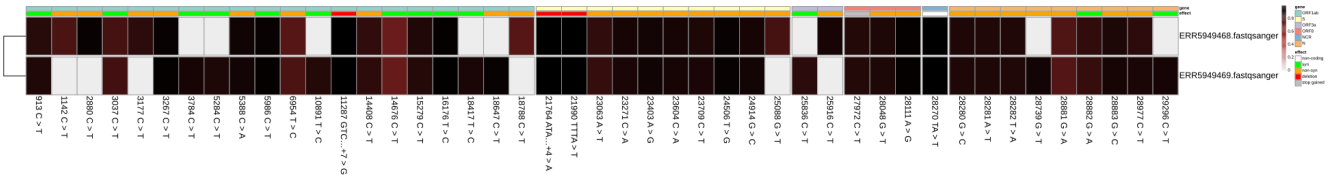
3. Variant frequency plot



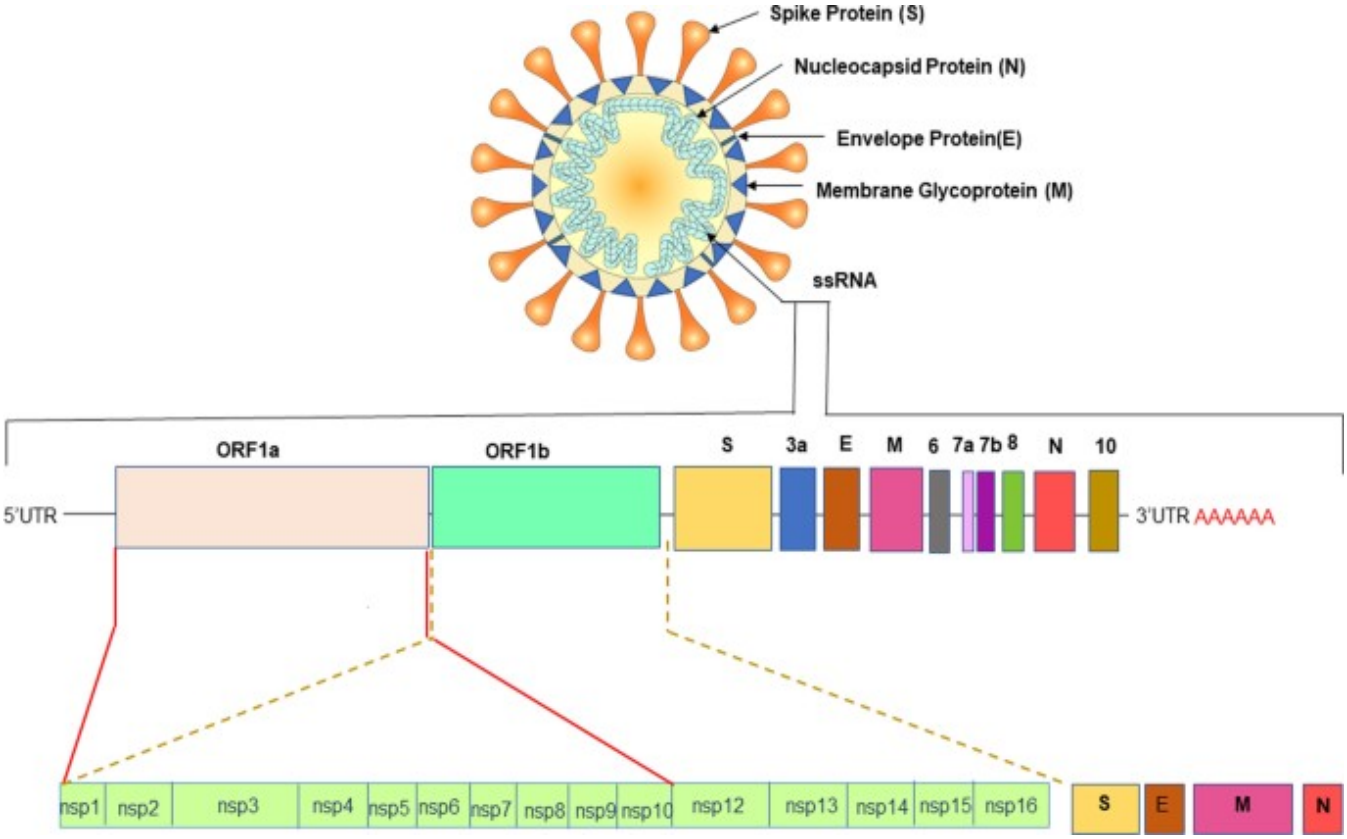
This plot represents AFs (cell color) for the different AVs (columns) and the different samples (rows). The AVs are grouped by genes (different colors on the 1st row). Information about their effect is also represented on the 2nd row. The samples are clustered following the tree displayed on the left.

In the example datasets, the samples are clustered in 3 clusters (as we defined when running the workflow), that may represent different SARS-CoV-2 lineages as the AVs profiles are different.

For this workshop, you only chose 1 or 2 samples to run, your frequency plot will look different from the image above using all samples. An example of how your image may look is below



For Reference: here is the structure of the genome.



(Taken from: https://respiratory-research.biomedcentral.com/articles/10.1186/s12931-020-01581-z)

Question: How many AVs are found for each sample?

To get the number of AVs for each sample, we can run **Group data** Tool: Grouping1 with the following parameters:

- param-file "*Select data*": `Combined Variant Report by Sample`
- "*Group by column*": `Column: 1`
- In "*Operation*":
    - In "*1: Operation*":
        - "*Type*": `Count`
        - "*On column*": `Column: 2`

With our example datasets, it seems that samples have between 42 and 56 AVs.

From AVs to consensus sequences

For the variant calls, we can now run a workflow which generates reliable consensus sequences according to transparent criteria that capture at least some of the complexity of variant calling:

- Each consensus sequence is guaranteed to capture all called, filter-passing variants as defined in the VCF of its sample that reach a user-defined consensus allele frequency threshold.
- Filter-failing variants and variants below a second user-defined minimal allele frequency threshold are ignored.
- Genomic positions of filter-passing variants with an allele frequency in between the two thresholds are hard-masked (with N) in the consensus sequence of their sample.
- Genomic positions with a coverage (calculated from the read alignments input) below another user-defined threshold are hard-masked, too, unless they are consensus variant sites.

The workflow takes a collection of VCFs and a collection of the corresponding aligned reads (for the purpose of calculating genome-wide coverage) such as produced by the first workflow we ran.

Get the workflow into Galaxy

- Option 1: Find workflows on the WorkflowHub and run them directly on usegalaxy.eu

  Please note that this option currently works *only* with usegalaxy.eu!

  - Open the workflow page on WokflowHub
  - Click on `Run on usegalaxy.eu` on the top right of the page

    The browser will open a new tab with Galaxy's workflow invocation interface.

Run **COVID-19: consensus construction** workflow using the following parameters:
Send results to a new history": `No`

- *"1: Variant calls"*: `Final (SnpEff-) annotated variants`

The collection with variation data in VCF format: the output of the first workflow

- *"2: min-AF for consensus variants*: `0.7`

Only variant calls with an AF greater than this value will be considered consensus variants.

- *"3: min-AF for failed variants"*: `0.25`

Variant calls with an AF higher than this value, but lower than the AF threshold for consensus variants will be considered questionable and the respective sites be masked (with Ns) in the consensus sequence. Variants with an AF below this threshold will be ignored.

- *"4: aligned reads data for depth calculation"*: `Fully processed reads for variant calling`

Collection with fully processed BAMs generated by the first workflow.

For ARTIC data, the BAMs should NOT have undergone processing with **ivar removereads**

- *"5: Depth-threshold for masking"*: `5`

Sites in the viral genome covered by less than this number of reads detection of variants is considered to become unreliable. Such sites will be masked (with Ns) in the consensus sequence unless there is a consensus variant call at the site.

- *"6: Reference genome"*: `NC_045512.2_reference.fasta` or `NC_045512.2 reference sequence`

SARS-CoV-2 reference genome, part of the auxillary data.

The main outputs of the workflow are:

- A collection of viral consensus sequences.
- A multisample FASTA of all these sequences.

The last one can be used as input for tools like **Pangolin** or **Nextclade**.

From consensus sequences to clade/lineage assignments

To assign lineages to the different samples from their consensus sequences, two tools are available: **Pangolin** and **Nextclade**.

With Pangolin

Pangolin (Phylogenetic Assignment of Named Global Outbreak LINeages) can be used to assign a SARS-CoV-2 genome sequence the most likely lineage based on the PANGO nomenclature system.

1. **Pangolin** Tool: with the following parameters:
   - param-file *"Input FASTA File(s)"*: `Multisample consensus FASTA`
2. Inspect the generated output

Pangolin generates a table file with taxon name and lineage assigned. Each line corresponds to each sample in the input consensus FASTA file provided. The columns are:

Column	Field	Meaning
1	`taxon`	The name of an input query sequence, here the sample name.
2	`lineage`	The most likely lineage assigned to a given sequence based on the inference engine used and the SARS-CoV-2 diversity designated. This assignment may be is sensitive to missing data at key sites. Lineage Description List
3	`conflict`	In the pangoLEARN decision tree model, a given sequence gets assigned to the most likely category based on known diversity. If a sequence can fit into more than one category, the conflict score will be greater than 0 and reflect the number of categories the sequence could fit into. If the conflict score is 0, this means that within the current decision tree there is only one category that the sequence could be assigned to.
4	`ambiguity_score`	This score is a function of the quantity of missing data in a sequence. It represents the proportion of relevant sites in a sequence which were imputed to the reference values. A score of 1 indicates that no sites were imputed, while a score of 0 indicates that more sites were imputed than were not imputed. This score only includes sites which are used by the decision tree to classify a sequence.
5	`scorpio_call`	If a query is assigned a constellation by scorpio this call is output in this column. The full set of constellations searched by default can be found at the constellations repository.
6	`scorpio_support`	The support score is the proportion of defining variants which have the alternative allele in the sequence.
7	`scorpio_conflict`	The conflict score is the proportion of defining variants which have the reference allele in the sequence. Ambiguous/other non-ref/alt bases at each of the variant positions contribute only to the denominators of these scores.

Column	Field	Meaning
8	`version`	A version number that represents both the pango-designation number and the inference engine used to assign the lineage.
9	`pangolin_version`	The version of pangolin software running.
10	`pangoLEARN_version`	The dated version of the pangoLEARN model installed.
11	`pango_version`	The version of pango-designation lineages that this assignment is based on.
12	`status`	Indicates whether the sequence passed the QC thresholds for minimum length and maximum N content.
13	`note`	If any conflicts from the decision tree, this field will output the alternative assignments. If the sequence failed QC this field will describe why. If the sequence met the SNP thresholds for scorpio to call a constellation, it'll describe the exact SNP counts of Alt, Ref and Amb (Alternative, Reference and Ambiguous) alleles for that call.

With Nextclade

Nextclade assigns clades, calls mutations and performs sequence quality checks on SARS-CoV-2 genomes.

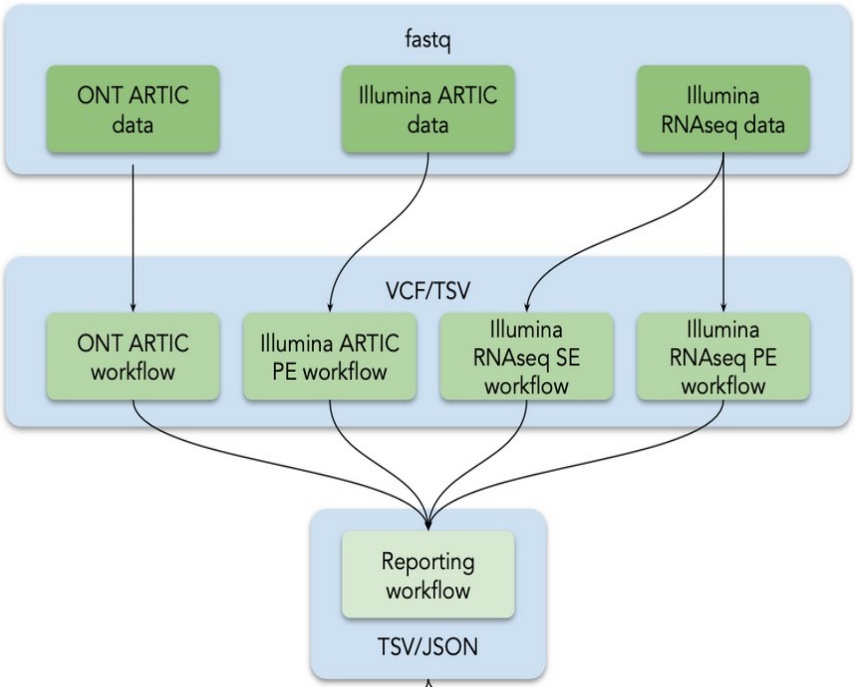
1. **Nextclade** Tool: toolshed.g2.bx.psu.edu/repos/iuc/nextclade/nextclade/0.14.4+galaxy0 with the following parameters:
    - param-file *"SARS-CoV-2 consensus sequences (FASTA)"*: `Multisample consensus FASTA`
    - param-check *"Output options"*: `Tabular format report`
2. Inspect the generated output

Column	Field	Meaning
1	`seqName`	Name of the sequence in the source data, here the sample name
2	`clade`	The result of the clade assignment of a sequence, as defined by Nextstrain. Currently known clades are depicted in the schema below
3	`qc.overallScore`	Overall QC score
4	`qc.overallStatus`	Overall QC status
5	`totalGaps`	Number of - characters (gaps)
6	`totalInsertions`	Total length of insertions
7	`totalMissing`	Number of N characters (missing data)
8	`totalMutations`	Number of mutations. Mutations are called relative to the reference sequence Wuhan-Hu-1

Column	Field	Meaning
9	totalNonACGTNs	Number of non-ACGTN characters
10	totalPcrPrimerChanges	Total number of mutations affecting user-specified PCR primer binding sites
11	substitutions	List of mutations
12	deletions	List of deletions (positions are 1-based)
13	insertions	Insertions relative to the reference Wuhan-Hu-1 (positions are 1-based)
14	missing	Intervals consisting of N characters
15	nonACGTNs	List of positions of non-ACGTN characters (for example ambiguous nucleotide codes)
16	pcrPrimerChanges	Number of user-specified PCR primer binding sites affected by mutations
17	aaSubstitutions	List of aminoacid changes
18	totalAminoacidSubstitutions	Number of aminoacid changes
19	aaDeletions	List of aminoacid deletions
20	totalAminoacidDeletions	Number of aminoacid deletions
21	alignmentEnd	Position of end of alignment
22	alignmentScore	Alignment score
23	alignmentStart	Position of beginning of alignment
24	qc.missingData.missingDataThreshold	Threshold for flagging sequences based on number of sites with Ns
25	qc.missingData.score	Score for missing data
26	qc.missingData.status	Status on missing data
27	qc.missingData.totalMissing	Number of sites with Ns
28	qc.mixedSites.mixedSitesThreshold	Threshold for flagging sequences based on number of mutations relative to the reference sequence
29	qc.mixedSites.score	Score for high divergence
30	qc.mixedSites.status	Status for high divergence
31	qc.mixedSites.totalMixedSites	Number of sites with mutations
32	qc.privateMutations.cutoff	Threshold for the number of non-ACGTN characters for flagging sequences
33	qc.privateMutations.excess	Number of ambiguous nucleotides above the threshold
34	qc.privateMutations.score	Score for ambiguous nucleotides
35	qc.privateMutations.status	Status for ambiguous nucleotides
36	qc.privateMutations.total	Number of ambiguous nucleotides
37	qc.snpClusters.clusteredSNPs	Clusters with 6 or more differences in 100 bases

Column	Field	Meaning
38	`qc.snpClusters.score`	Score for clustered differences
39	`qc.snpClusters.status`	Status for clustered differences
40	`qc.snpClusters.totalSNPs`	Number of differences in clusters
41	`errors`	Other errors (e.g. sequences in which some of the major genes fail to translate because of frame shifting insertions or deletions)

In this tutorial, we used a collection of Galaxy workflows for the detection and interpretation of sequence variants in SARS-CoV-2:



References

1. Li, H., and R. Durbin, 2010 **Fast and accurate long-read alignment with Burrows–Wheeler transform**. Bioinformatics 26: 589–595. 10.1093/bioinformatics/btp698
2. Langmead, B., and S. L. Salzberg, 2012 **Fast gapped-read alignment with Bowtie 2**. Nature Methods 9: 357–359. 10.1038/nmeth.1923
3. Wilm, A., P. P. K. Aw, D. Bertrand, G. H. T. Yeo, S. H. Ong *et al.*, 2012 **LoFreq: a sequence-quality aware, ultra-sensitive variant caller for uncovering cell-population heterogeneity from high-throughput sequencing datasets**. Nucleic Acids Research 40: 11189–11201. 10.1093/nar/gks918
4. Li, H., 2018 **Minimap2: pairwise alignment for nucleotide sequences**. Bioinformatics 34: 3094–3100.

Citation of main Galaxy Tutorial

1. Wolfgang Maier, Bérénice Batut, 2021 **Mutation calling, viral genome reconstruction and lineage/clade assignment from SARS-CoV-2 sequencing data (Galaxy Training Materials)**. https://training.galaxyproject.org/training-material/topics/variant-analysis/tutorials/sars-cov-2-variant-discovery/tutorial.html Online; accessed Wed Feb 02 2022
2. Batut et al., 2018 **Community-Driven Data Analysis Training for Biology** Cell Systems 10.1016/j.cels.2018.05.012