Bridging the gaps in bioinformatics/Genome assembly

# Assembling influenza genomes with IRMA

**Marta Maria Ciucani, MSc, PhD**
**Bioinformatician**
**Influenza group**
**Statens Serum Institut**

18/05/2023

# Intended Learning Outcomes (ILOs)

01

Understanding the genomic structure of influenza virus

02

Summarize key challenges related to sequencing influenza genomes

03

Use a published pipeline (IRMA) for generating a viral genome: intro and usage

04

Understand the result of the assembly (I.e. the output-files generated by the pipeline)

05

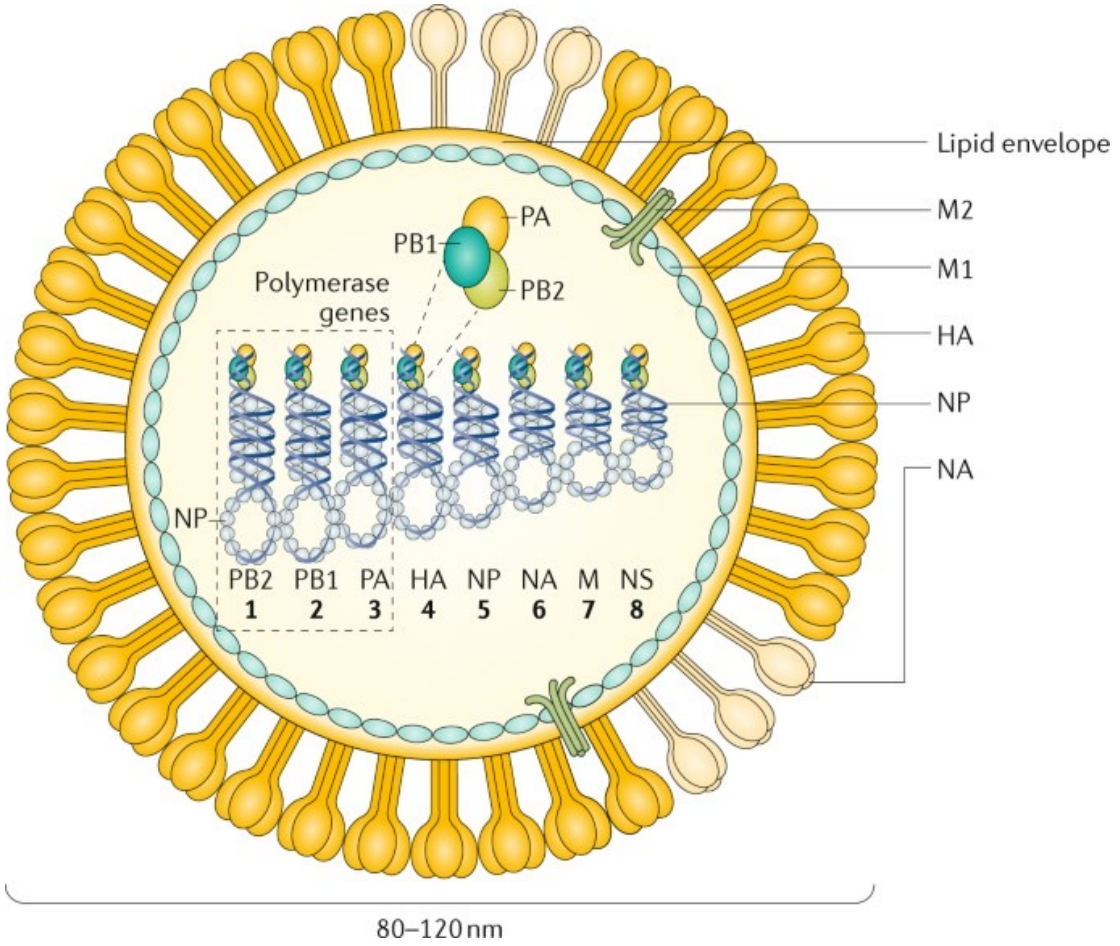
Know how to modify parameters to adapt the analysis to user-specific needs

# Influenza genomes

The influenza virus is an RNA virus with a segmented genome consisting of eight negative-sense single-stranded RNA segments.

Each RNA segment codes for at least one protein, with some segments encoding multiple proteins through alternative splicing.

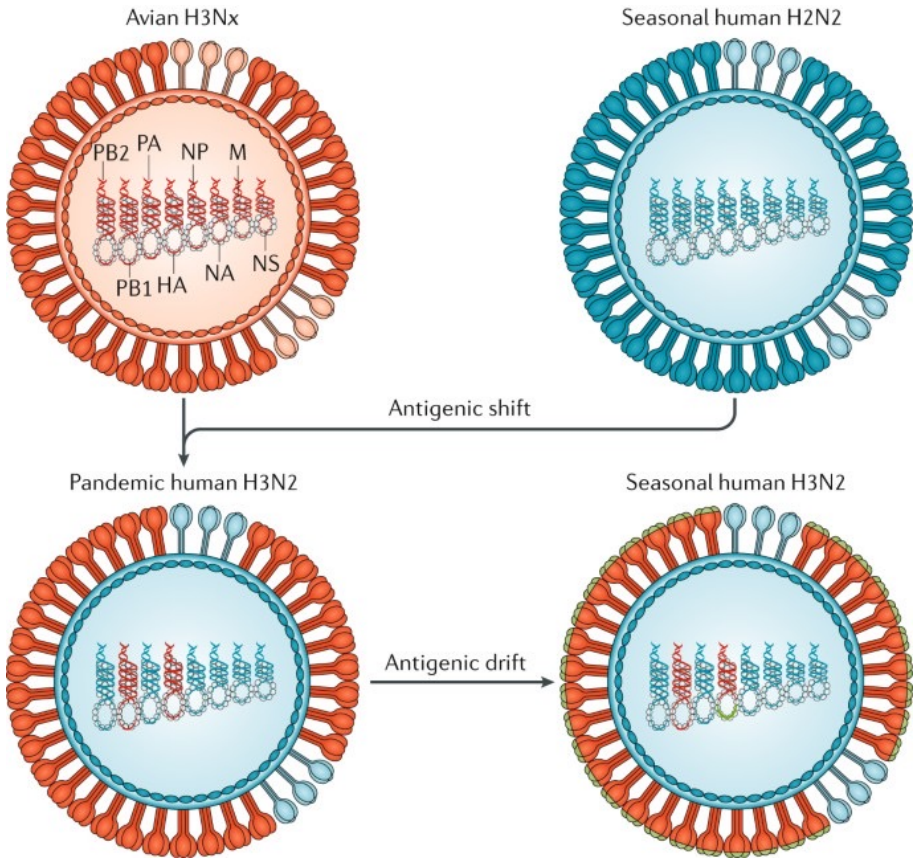
The genome segments are enclosed within a protein shell, or nucleocapsid, and surrounded by a lipid envelope containing two major surface glycoproteins, hemagglutinin (HA) and neuraminidase (NA), which are responsible for the virus's antigenicity and host specificity.




source: https://www.nature.com/articles/s41572-018-0002-y

# Influenza genomes

Influenza viruses undergo frequent genetic changes, both through antigenic drift and antigenic shift.

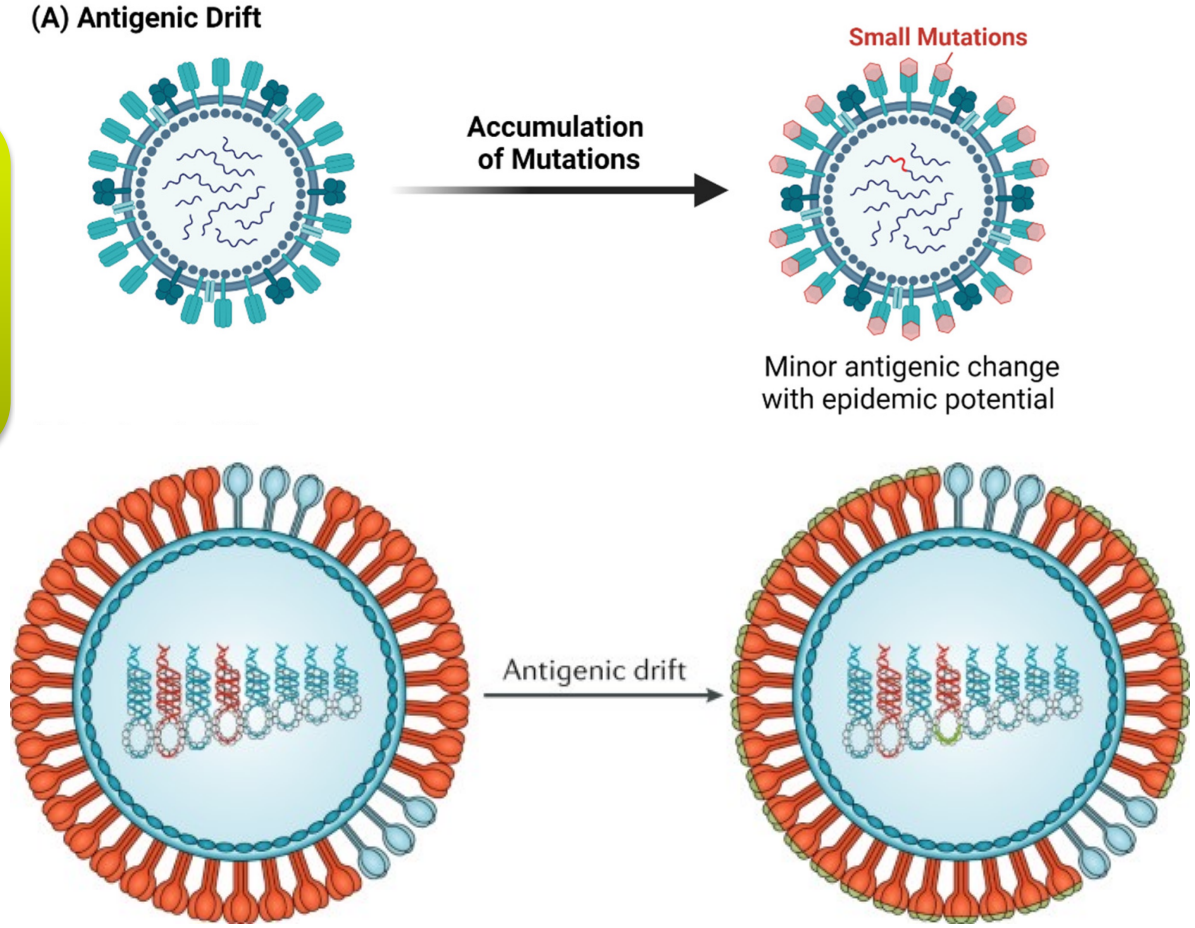



source: https://www.nature.com/articles/s41572-018-0002-y

# Antigenic drift

Gradual accumulation of point mutations in the genes encoding the HA and NA proteins, which results in minor changes to the virus's surface antigens (HA and NA).

- HA and NA are recognized by the immune system and can trigger an immune response (including producing antibodies to fight infection).
- The small genetic changes usually produce viruses that are closely related to one another, which can be illustrated by their location close together on a phylogenetic tree.




sources: https://www.mdpi.com/1999-4915/13/11/2276#
https://www.nature.com/articles/s41572-018-0002-y

# Antigenic drift

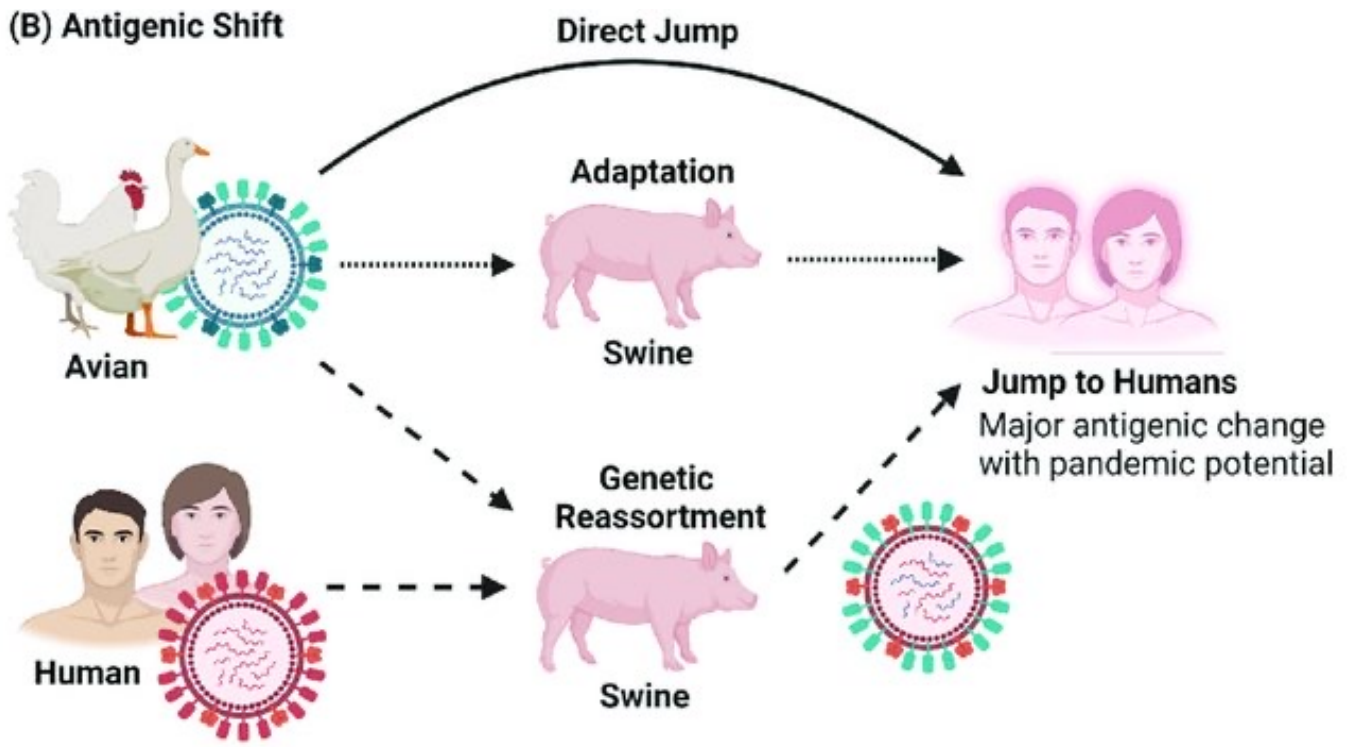
Over time mutations lead to viruses that are antigenically different, meaning a person's antibodies bind differently or not at all to the virus, resulting in a loss or reduction in protection against that particular flu virus.

Can cause multiple flu infection over our life time

It's the reason why we annually change the flu vaccine composition.

# Antigenic shift

Occurs when two different influenza viruses infect the same host cell and exchange genome segments, resulting in a novel virus with new surface antigens.

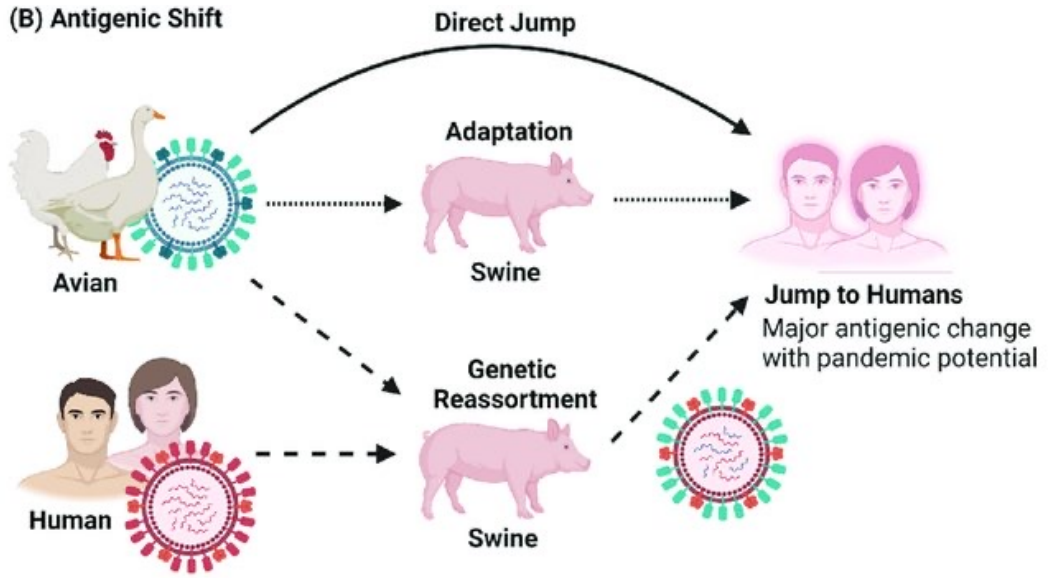



source: https://www.mdpi.com/1999-4915/13/11/2276

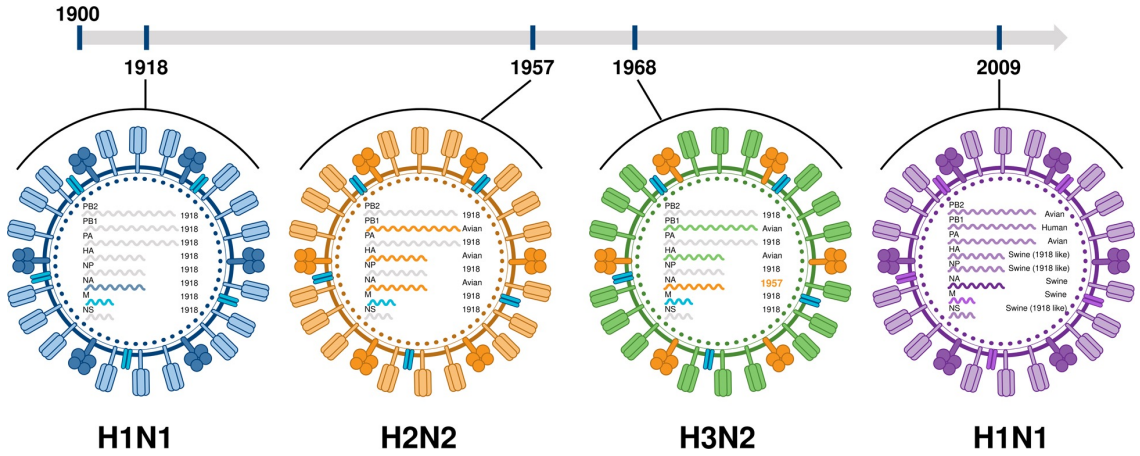
# Antigenic shift

Results in major antigenic change via direct jump, adaptation and genetic reassortment.

Antigenic shift is responsible for the emergence of pandemic influenza strains, such as the H1N1 strain that caused the 1918 Spanish flu pandemic and the H5N1 strain that has caused outbreaks in birds and humans in recent years.




source: https://www.mdpi.com/1999-4915/13/11/2276




source: https://doi.org/10.1038/s12276-021-00603-0

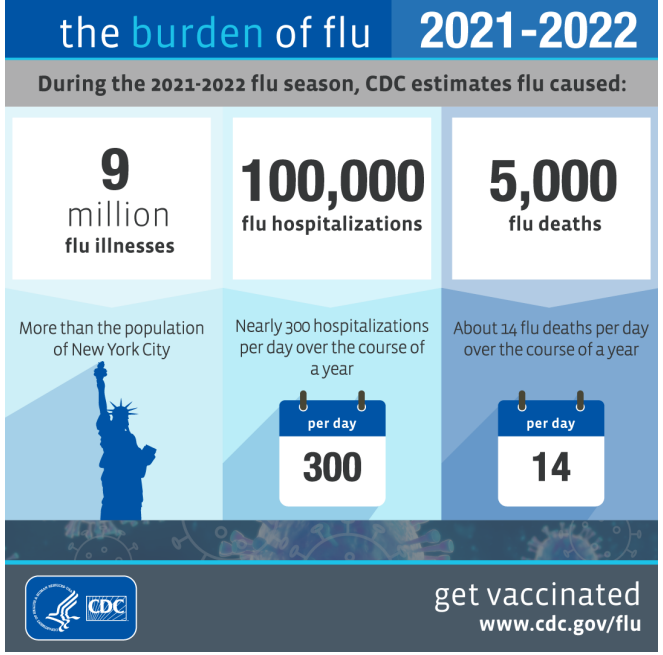
# Sars-Cov2 VS Influenza

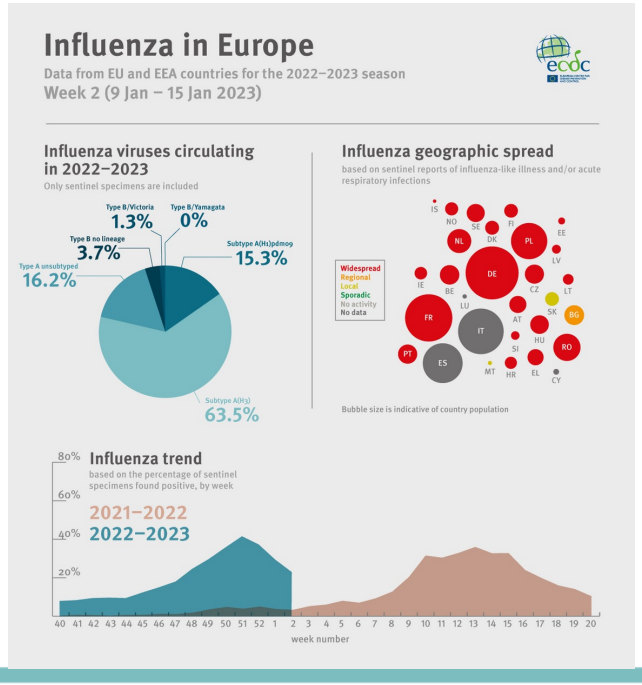
Genome size and structure: The genome of SARS-CoV-2 is a single-stranded RNA molecule, about 30,000 nucleotides long, which encodes for 29 proteins. In contrast, the influenza virus genome is composed of eight segments of single-stranded RNA, encoding for a total of 11 proteins.

Mutation rate: RNA viruses like SARS-CoV-2 and influenza viruses are known for their high mutation rates, which allow them to adapt quickly to changing environments. However, the mutation rate of SARS-CoV-2 is lower than that of influenza viruses.

Antigenic variation: Influenza viruses are notorious for their ability to undergo rapid antigenic drift, which allows them to evade the host immune system and cause seasonal epidemics. SARS-CoV-2 also exhibits some degree of antigenic variation, but it appears to be less pronounced than that of influenza viruses.

# Background




Influenza viruses cause a significant disease burden as a result of seasonal activity and outbreaks

Disease severity and fast mutation rate -> large global surveillance network is required

The high level of mutation inherent in influenza virus reproduction leads to antigenic drift within gene segments while reassortment of segments causes antigenic shift which can cause outbreaks of infection

sources: ECDC - https://www.ecdc.europa.eu/en/publications-data/weekly-influenza-update-week-2-january-2023
and CDC - https://www.cdc.gov/flu/resource-center/freeresources/graphics/flu-burden-2022-2023.htm

# Background


Rapid expansion in surveillance efforts for zoonotic viruses and use of NGS technologies.

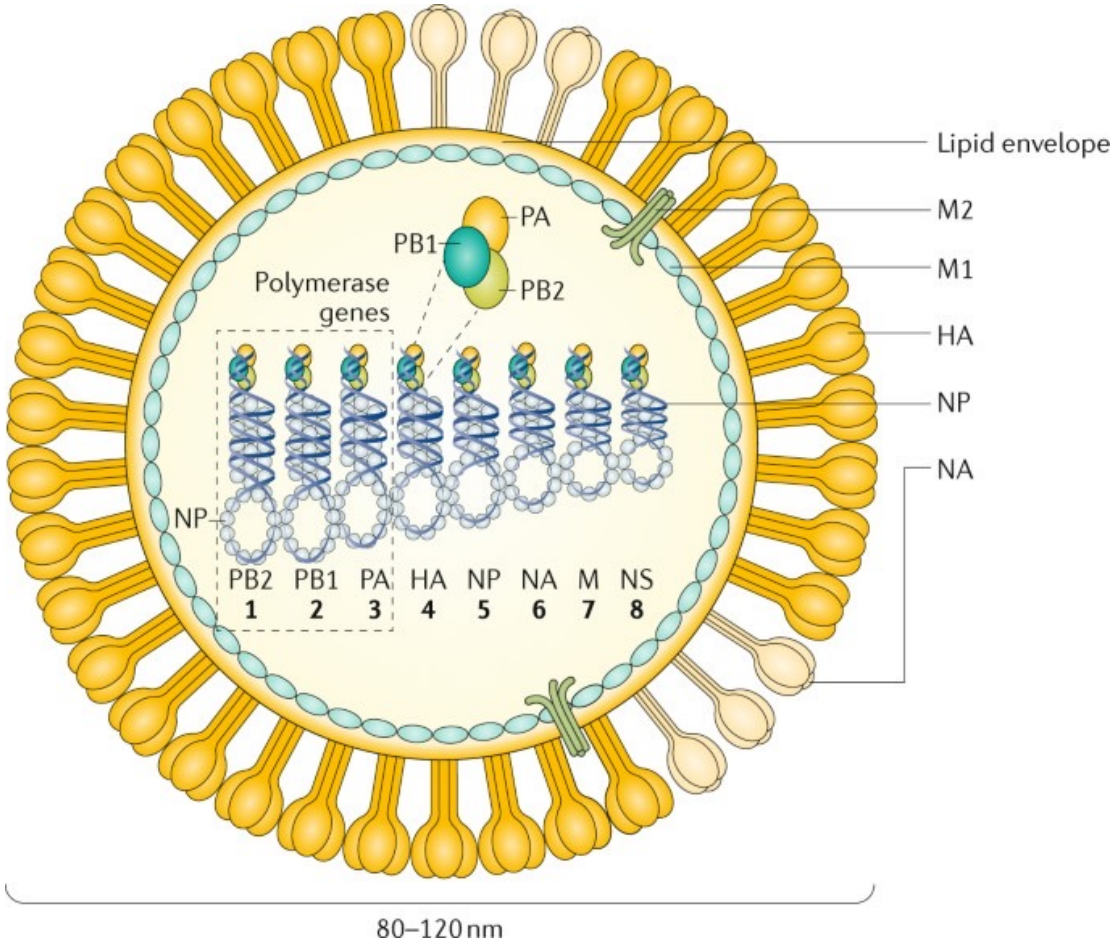
NGS offers advantages for surveillance and outbreak investigation in terms of speed and resolution of sequence differences

# Background

Reference based NGS assembly programs do not perform well with the influenza segmented genome.

These programs discard read sequences from assembly that have too many mismatches or insertions/deletions (indels)

These approaches minimise coverage and prevent complete assembly




source: https://www.nature.com/articles/s41572-018-0002-y

Shepard *et al. BMC Genomics* (2016) 17:708
DOI 10.1186/s12864-016-3030-6

BMC Genomics

METHODOLOGY ARTICLE Open Access

# Viral deep sequencing needs an adaptive approach: IRMA, the iterative refinement meta-assembler

Samuel S. Shepard[1*], Sarah Meno[1], Justin Bahl[2], Malania M. Wilson[1,3], John Barnes[1] and Elizabeth Neuhaus[1*]

1- Influenza Division, Centers for Disease Control and Prevention, 1600 Clifton Road, Atlanta, GA 30329, USA.
2-Center for Infectious Diseases, The University of Texas School of Public Health, Houston, TX, USA.
3-Battelle Memorial Research Institute, 1600 Clifton Road, Atlanta, GA 30329, USA.

# IRMA: the iterative refinement meta- assembler

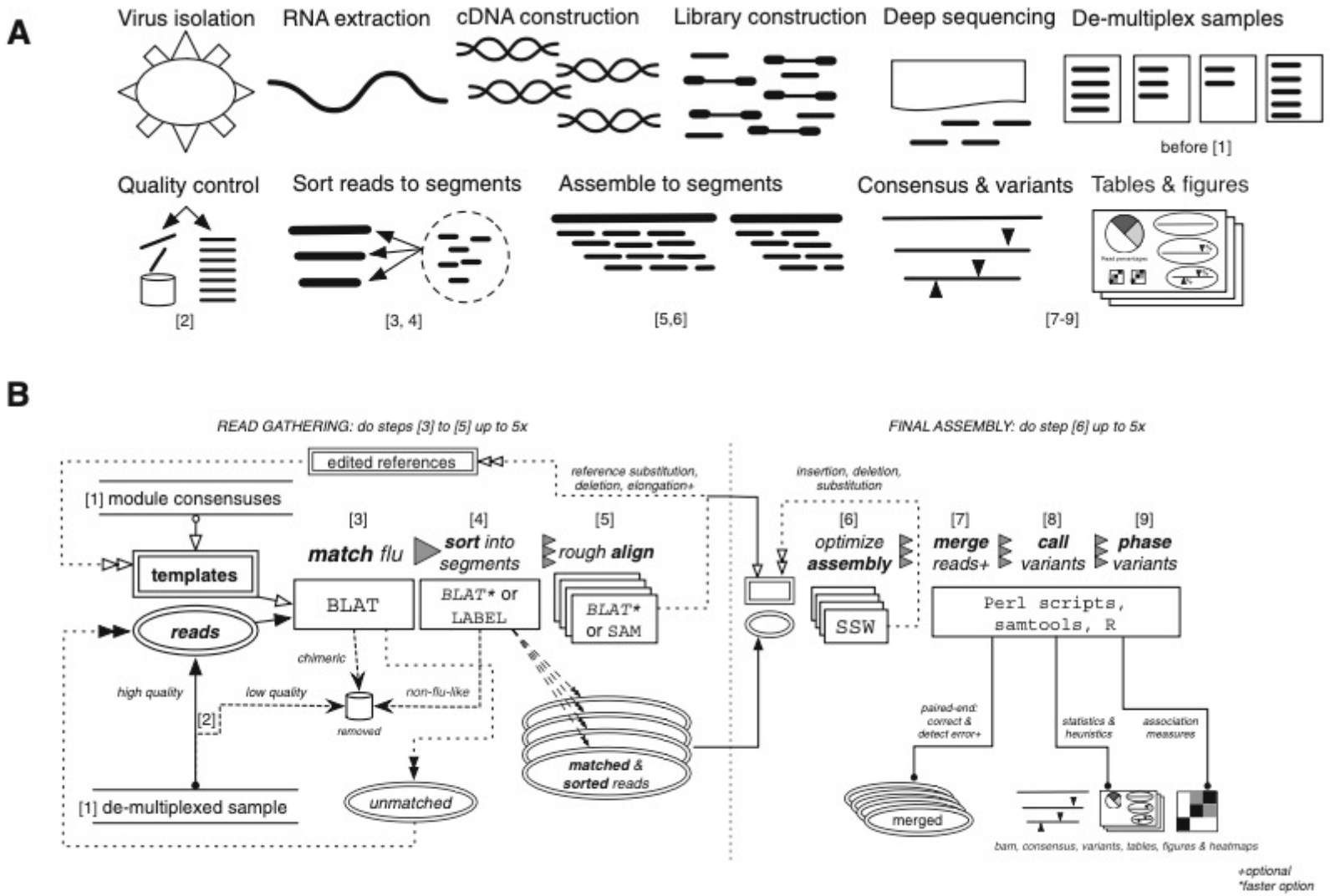
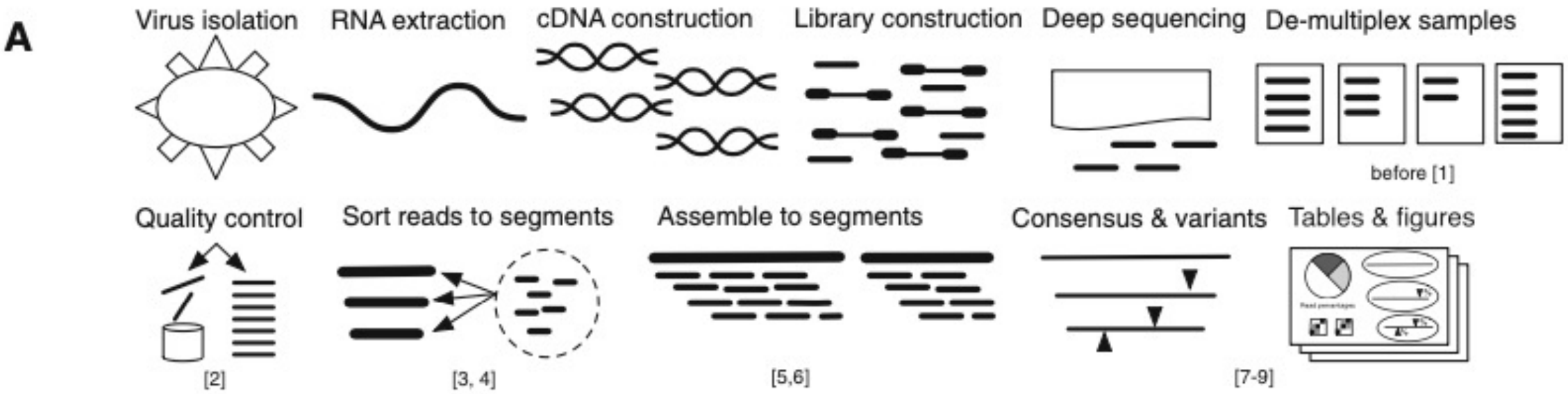



**Fig. 1** Iterative refinement meta-assembler (IRMA) workflow: the influenza module. (**a**) The general process of sequencing a segmented RNA virus and assembling with IRMA. (**b**) Diagram of IRMA steps 1 through 9, showing the iterative processes involved. Steps in (**b**) are also labeled under the steps of (**a**) where they correspond

source: https://bmcgenomics.biomedcentral.com/articles/10.1186/s12864-016-3030-6




source: https://bmcgenomics.biomedcentral.com/articles/10.1186/s12864-016-3030-6

# Diagram of IRMA workflow

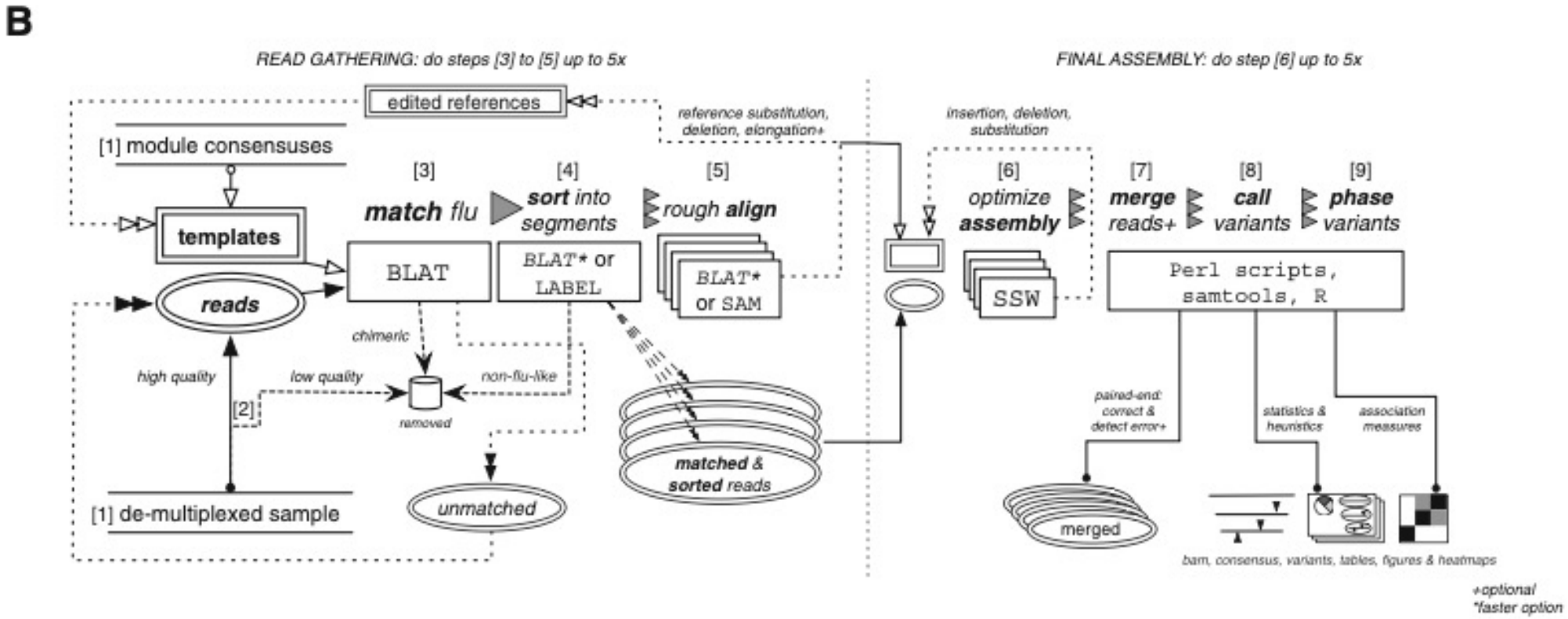



**Fig. 1** Iterative refinement meta-assembler (IRMA) workflow: the influenza module. (**a**) The general process of sequencing a segmented RNA virus and assembling with IRMA. (**b**) Diagram of IRMA steps 1 through 9, showing the iterative processes involved. Steps in (**b**) are also labeled under the steps of (**a**) where they correspond

source: https://bmcgenomics.biomedcentral.com/articles/10.1186/s12864-016-3030-6

# IRMA: the iterative refinement meta-assembler

Developed as a flexible approach that more thoroughly addresses viral diversity

Provides a comprehensive solution to address each aspect of NGS assembly, as it applies to RNA virus evolution, in a flexible and robust manner

Used to process genome sequence data derived from the large volume of surveillance specimens characterized at CDC

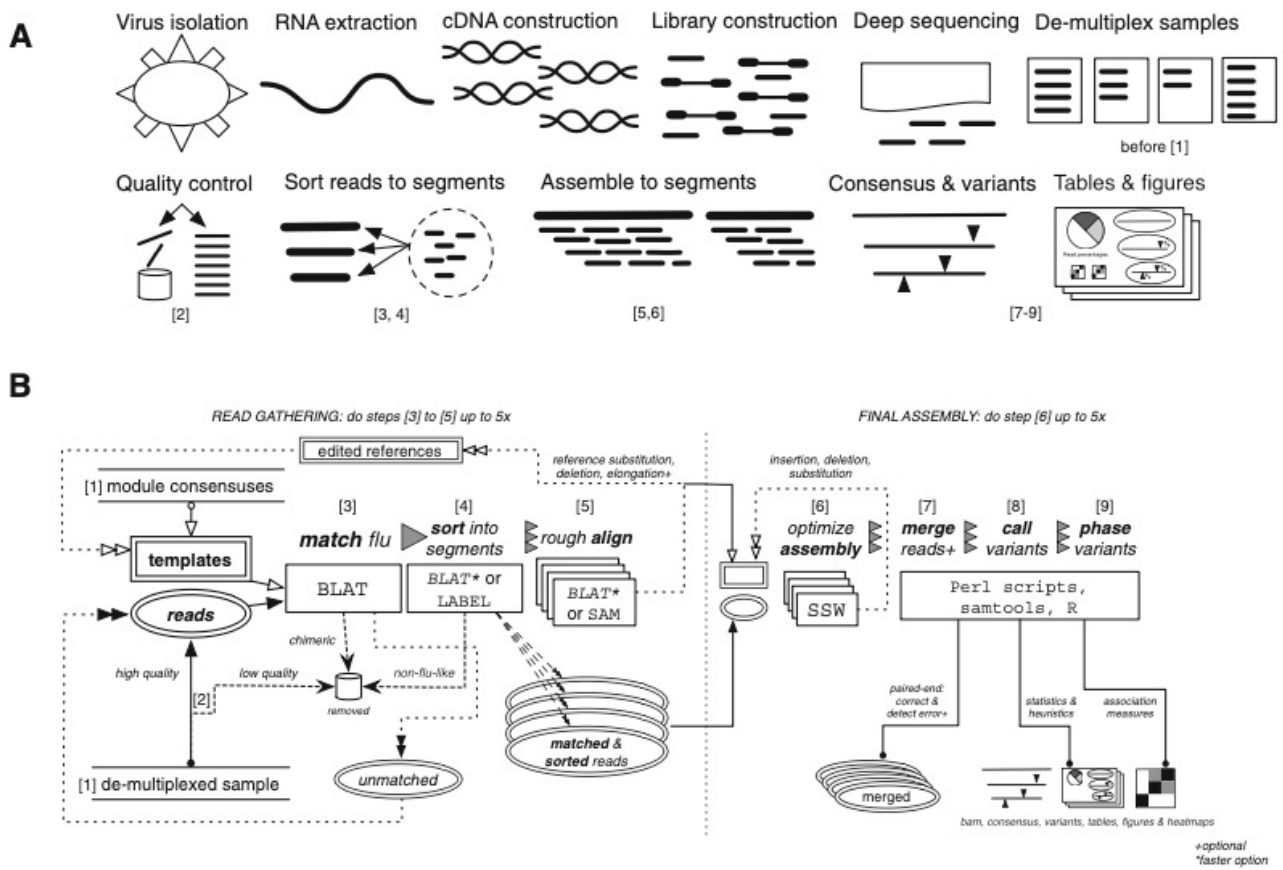



**Fig. 1** Iterative refinement meta-assembler (IRMA) workflow: the influenza module. (**a**) The general process of sequencing a segmented RNA virus and assembling with IRMA. (**b**) Diagram of IRMA steps 1 through 9, showing the iterative processes involved. Steps in (**b**) are also labeled under the steps of (**a**) where they correspond

source: https://bmcgenomics.biomedcentral.com/articles/10.1186/s12864-016-3030-6

# IRMA: the iterative refinement meta- assembler

Used successfully to identify low frequency variants

IRMA applies an iterative refinement process that improves assembly accuracy

Uses a hierarchical approach that can efficiently assemble reads across a range of coverage depths and phased minor variants

Provides a comprehensive set of analysis outputs.

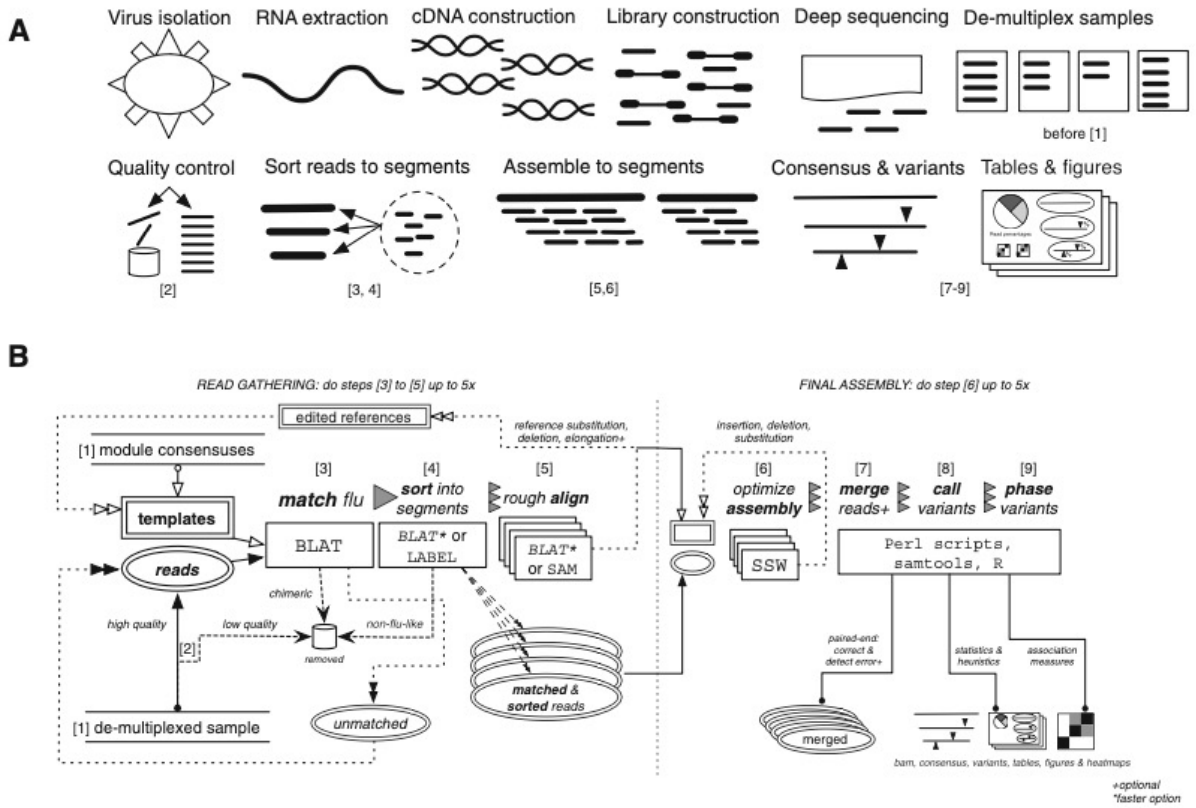



**Fig. 1** Iterative refinement meta-assembler (IRMA) workflow: the influenza module. (**a**) The general process of sequencing a segmented RNA virus and assembling with IRMA. (**b**) Diagram of IRMA steps 1 through 9, showing the iterative processes involved. Steps in (**b**) are also labeled under the steps of (**a**) where they correspond

source: https://bmcgenomics.biomedcentral.com/articles/10.1186/s12864-016-3030-6

# Install IRMA

Warning!! IRMA was designed for use with Linux and Mac OS, not Windows.

IRMA requires at least Perl version 5 and BASH version 3, which is standard on most Linux & Mac OS X systems.

R must be available on any computer which runs IRMA processes.

Download:  get the latest version of IRMA & LABEL and unzip the archive in the desired <install_path>.

More instruction at this link:
https://wonder.cdc.gov/amd/flu/irma/install.html

# Dependencies

**BLAT** for the matching step of the flu reads

**LABEL** (for sorting reads into segments), which also packages certain resources used by IRMA:

- Sequence Alignment and Modeling System (SAM) for both the rough align and sort steps
- Shogun Toolbox, which is an essential part of LABEL, is used in the sort step

**SSW** (modification of Smith-Waterman algorithm) for the final assembly step

**samtools** for BAM-SAM conversion as well as BAM sorting and indexing

# How to install IRMA

```
name: irmaenv2
channels:
  - conda-forge
  - bioconda
  - defaults
dependencies:
  - bash >=4.4.18
  - perl >=5.26.2
  - irma =1.0.2
  - r-base >=3.5.1
  - parallel >=20170422
  - zip >=3.0
  - pigz >=2.3.4
  - fasttree >=2.1.10
  - mafft >=7.407
  - samtools >=1.9
  - blat >=36
  - fastp=0.20.1
  - snakemake=6.6.1
  - python=3.10.2
```

Or, you can simply use a conda environment!

So make sure you install conda or miniconda

```
# Install Conda
wget https://repo.continuum.io/miniconda/Miniconda3-latest-Linux-x86_64.sh -O miniconda.sh

bash miniconda.sh -b -p $HOME/miniconda
```

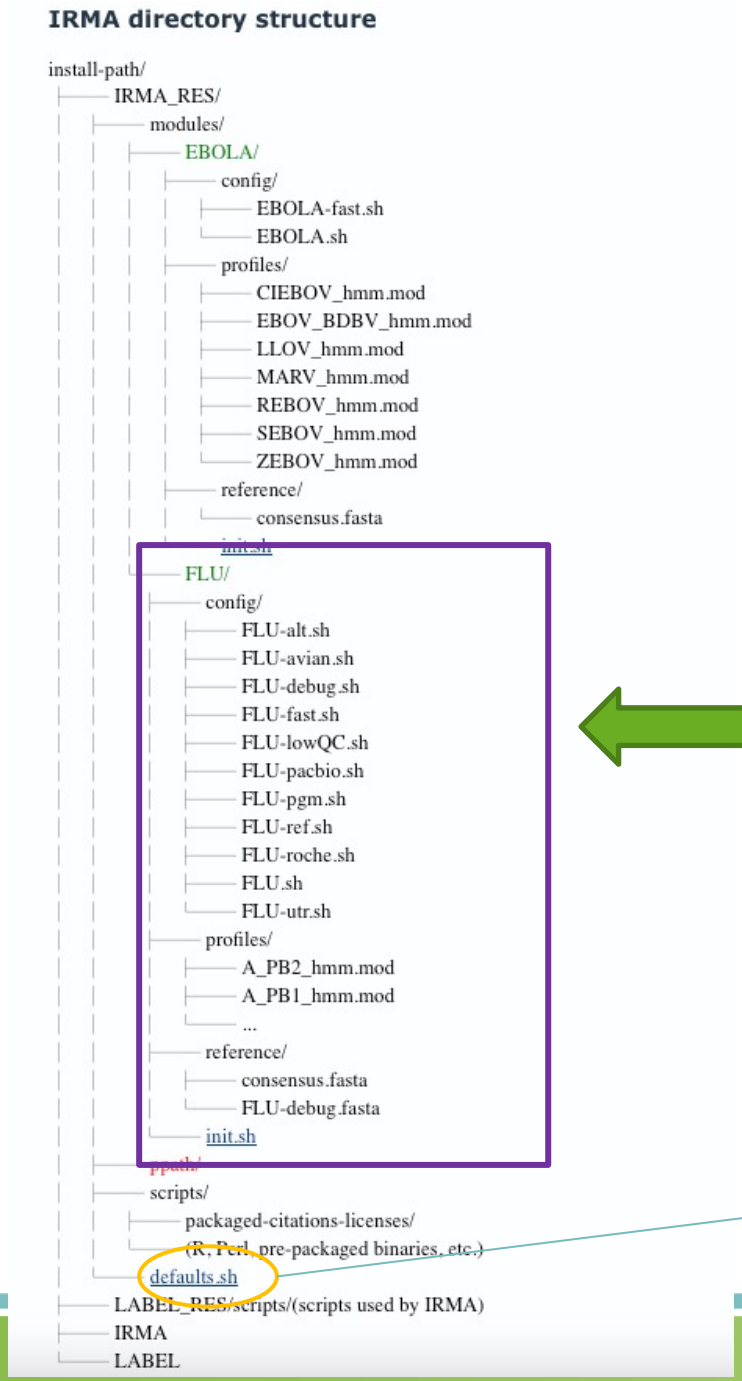
And then:

```
# Download repo
git clone https://github.com/peterk87/irma.git
cd irma
# create IRMA conda env
conda env create --file=conda_env.yaml

# activate IRMA conda env

conda activate irmaenv
```

```
name: irmaenv
channels:
  - conda-forge
  - bioconda
  - defaults
dependencies:
  - bash >=4.4.18
  - perl >=5.26.2
  - r-base >=3.5.1
  - parallel >=20170422
  - zip >=3.0
  - pigz >=2.3.4
  - fasttree >=2.1.10
  - mafft >=7.407
  - samtools >=1.9
  - blat >=36
```

More instruction at this link: https://wonder.cdc.gov/amd/flu/irma/install.html

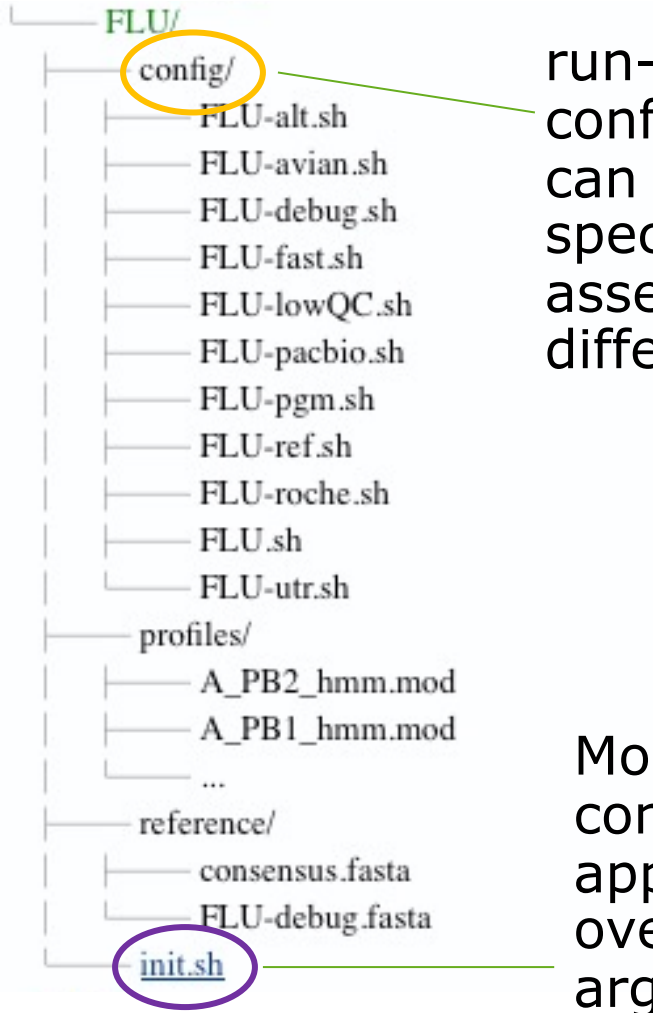
# Irma directory




Global conf file.
Variables should not be deleted from this file

More info here: https://wonder.cdc.gov/amd/flu/irma/modules.html

# Irma directory




run-specific named configuration files can be applied to specialize the assembly for different situations.

Module-specific configurations are applied that may override any global arguments. These configurations help adjust the assembly to the organism of interest.

More info here: https://wonder.cdc.gov/amd/flu/irma/modules.html

# init.sh file

```
### PERFORMANCE ###
GRID_ON=0               # grid computation on [1,0] for on or off
GRID_PATH=""            # grid path, defaults to the IRMA_RES path if left empty string, do not include quotes for tilde prefix
SINGLE_LOCAL_PROC=16    # local maximum processes
DOUBLE_LOCAL_PROC=8     # local maximum processes (double this number)
ALLOW_TMP=1             # if GRID_ON=0, try to use /tmp for working directory
TMP=/tmp                # the scratch/tmpfs for working on the assemblies

### REFERENCE ###
MIN_FA=1                # no alternative reference [0..1]
MIN_CA=20               # minimum count for alternative finished assembly
SKIP_E=1                # skip reference elongation
REF_SET=$DEF_SET        # Same as the "consensus.fasta" in the reference folder for the module.
MIN_CONS_SUPPORT=100    # minimum allele coverage depth to call plurality consensus, otherwise calls "N".

### READ GATHERING ###
MAX_ROUNDS=5                    # round of read gathering
USE_MEDIAN=1                    # use the median quality or the average [1,0]
QUAL_THRESHOLD=30               # minimum read statistic
MIN_LEN=125                     # minimum read length
ENFORCE_CLIPPED_LENGTH=0        # Off. Reads are filtered for minimum length post adapter trimming.

## MATCH STEP
MATCH_PROC=20           # grid maximum processes for the MATCH
MATCH_PROG="BLAT"       # match (all or any match) program [BLAT]
MIN_RP=15               # minimum read pattern count to continue
MIN_RC=15               # minimum read count to continue

## SORT STEP
SORT_PROG="BLAT"        # [LABEL,BLAT]
SORT_PROC=80            # currently not used
NONSEGMENTED=0
# Pattern list to group gene segment lineages into gene for primary/secondary sorting.
SORT_GROUPS="PB2,PB1,PA,HA,NP,NA,MP,NS"
SEG_NUMBERS="B_PB1:1,B_PB2:2,A_PB2:1,A_PB1:2,PA:3,HA:4,NP:5,NA:6,M:7,NS:8"

# LABEL
SECONDARY_SORT=1                                        # LABEL sorting fast-mode
LABEL_MODULE="irma-FLU"                                 # if LABEL SECONDARY SORT is 0, use LABEL_MODULE
SECONDARY_LABEL_MODULES="irma-FLU-HA,irma-FLU-NA:irma-FLU-OG"   # otherwise, search for primary classification from BLAT and use the modules accordingly
GENE_GROUP="HA,NA:OG"                                   # specify primary sorting gene groups for BLAT

## ALIGN STEP ##
ALIGN_PROG="SAM"        # rough assembly / alignment to working reference [SAM,BLAT]
ALIGN_PROC=20           # grid maximum processes for the rough align

### FINISHING ASSEMBLY ###
ASSEM_PROG="SSW"        # assembly program [SSW]
ASSEM_PROC=20           # grid maximum processes for assembly
INS_T=0.25              # minimum frquenncy threshold for insertion refinement
DEL_T=0.60              # minimum frequency threshold for deletion refinement
MIN_AMBIG=0.20          # minimum called SNV frequency for mixed base in amended consensus folder
```

keep an eye on the parameters that you would like to change and adapt to your needs!

More info here: https://wonder.cdc.gov/amd/flu/irma/configuration.html

# How to run IRMA

Easy peasy!

Calling IRMA requires three components:

(1) a module argument specifying the organism and an optional run-specific configuration,

(2) the input fastq data, and

(3) the output name for the sample.

Note: If more than one fastq are needed per sample, then one needs to concatenate the appropriate read files.

# How to run IRMA

**Paired-end files:**

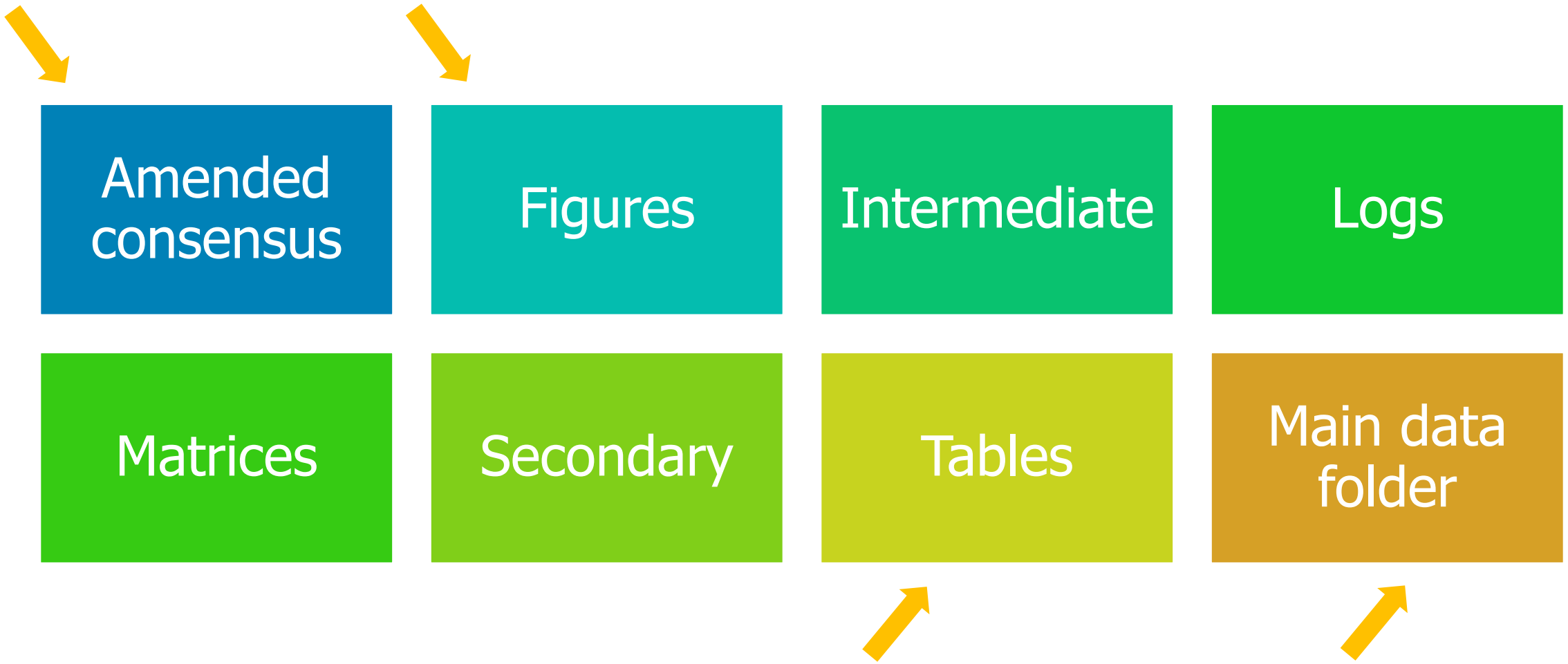
```
USAGE:          IRMA <MODULE-config> <R1.fastq.gz/R1.fastq> <R2.fastq.gz/R2.fastq> <sample_name>
Example 1:      IRMA FLU Sample1_R1.fastq.gz Sample1_R2.fastq.gz Sample1
Example 2:      IRMA EBOLA Patient1_R1.fastq Patient1_R2.fastq MyPatient
Example 3:      IRMA FLU-utr Sample1_R1.fastq.gz Sample1_R2.fastq.gz Sample1WithUTRs
```

**Single read files:**

```
USAGE:          IRMA <MODULE-config> <fastq/fastq.gz> <sample_name>
Example 1:      IRMA FLU SingleEndIllumina.fastq.gz MyIlluminaSample
Example 2:      IRMA FLU-pacbio ccs_reads.fastq MyPacBioSample
Example 3:      IRMA FLU-pgm pgm_reads.fastq MyIonTorrentSample
```

source: https://wonder.cdc.gov/amd/flu/irma/run.html

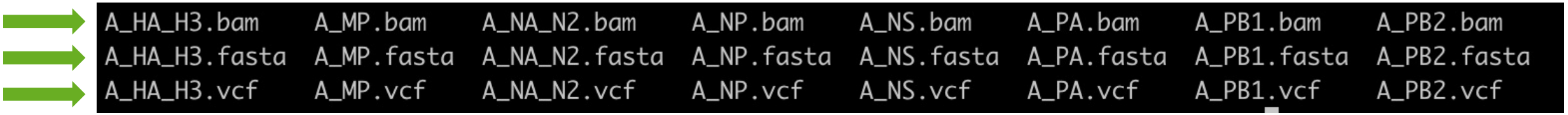
# IRMA outputs



# IRMA outputs

| | | | |
|---|---|---|---|
| Amended consensus | Figures | Intermediate | Logs |
| Matrices | Secondary | Tables | Main data folder |

# Irma outputs – Main Data Folder




bam

fasta

vcf

**Plurality consensus sequences**

- They are named after the virus genome or gene segment class label that was matched to.
- Useful when used with the BAM file to look at minor variants
- A plurality rule was chosen over majority consensus because it is more inclusive for pattern matching purposes and does not assign strict thresholds for the dominant virus phase in the sample.
- Other parameters are available to restrict the quality of the consensus alleles as part of the amended consensus.

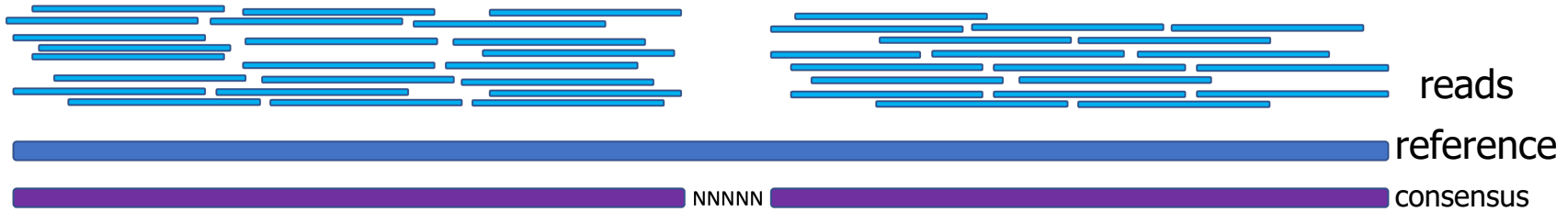
# IRMA outputs

| | | | |
|---|---|---|---|
| Amended consensus | Figures | Intermediate | Logs |
| Matrices | Secondary | Tables | Main data folder |

# Amended consensus

These sequences are modifications to the plurality consensus.

- The first type of amendment is base ambiguation for mixed alleles.
- The second type of base amendment is for consensus allele quality control.

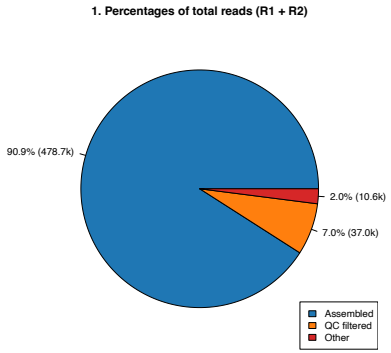



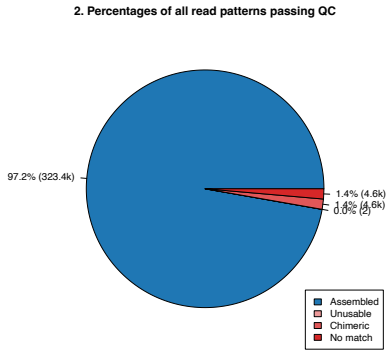
# IRMA outputs

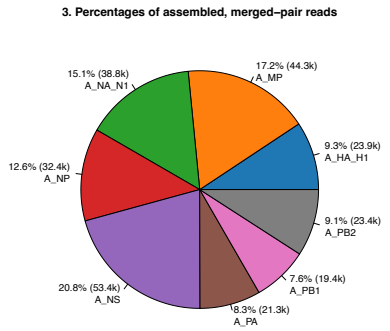
| | | | |
|---|---|---|---|
| Amended consensus | Figures | Intermediate | Logs |
| Matrices | Secondary | Tables | Main data folder |

# Figures

- A_HA_H1-coverageDiagram.pdf
- A_HA_H1-EXPENRD.pdf
- A_HA_H1-heuristics.pdf
- A_HA_H1-JACCARD.pdf
- A_HA_H1-MUTUALD.pdf
- A_HA_H1-NJOINTP.pdf
- A_MP-coverageDiagram.pdf
- A_MP-heuristics.pdf
- A_NA_N1-coverageDiagram.pdf
- A_NA_N1-heuristics.pdf
- A_NP-coverageDiagram.pdf
- A_NP-heuristics.pdf
- A_NS-coverageDiagram.pdf
- A_NS-heuristics.pdf
- A_PA-coverageDiagram.pdf
- A_PA-heuristics.pdf
- A_PB1-coverageDiagram.pdf
- A_PB1-EXPENRD.pdf
- A_PB1-heuristics.pdf
- A_PB1-JACCARD.pdf
- A_PB1-MUTUALD.pdf
- A_PB1-NJOINTP.pdf
- A_PB2-coverageDiagram.pdf
- A_PB2-heuristics.pdf
- READ_PERCENTAGES.pdf



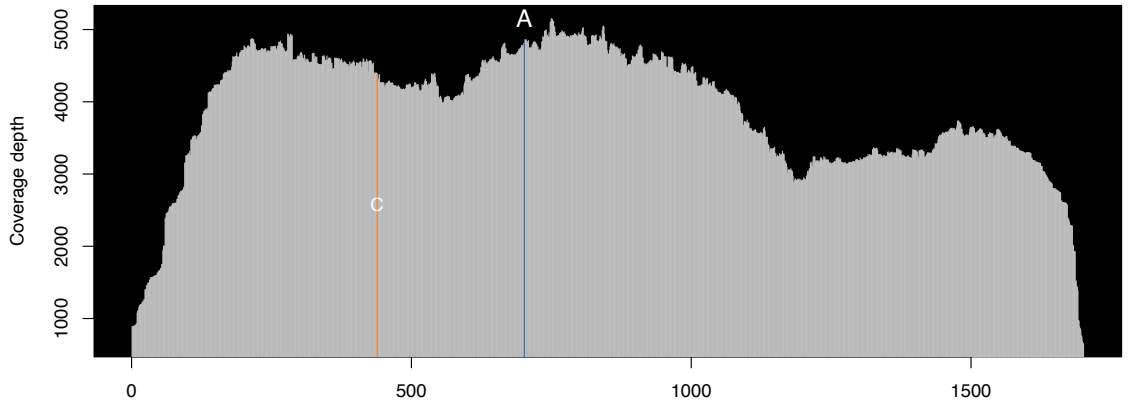



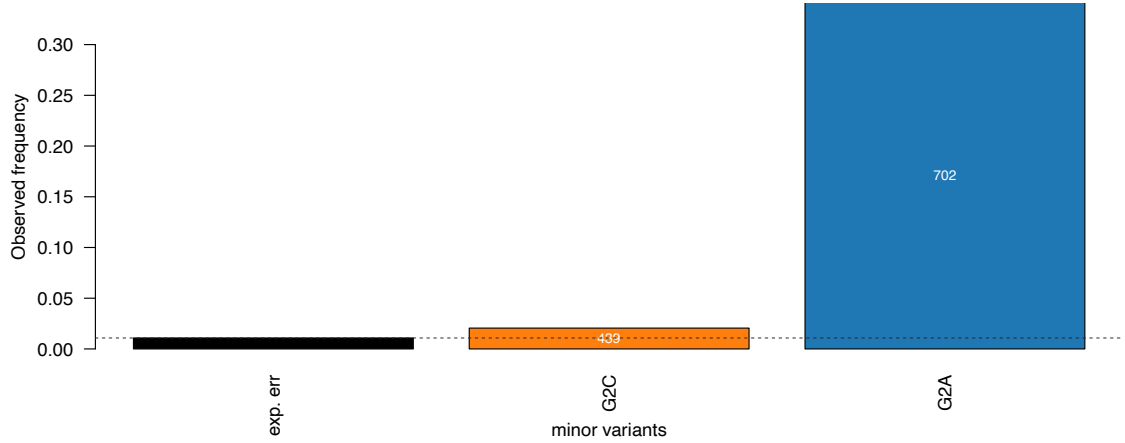

READ PROPORTIONS.

1. Percentages of total read counts (R1 & R2)
   – ASSEMBLED: influenza reads in final assemblies.
   – QC FILTERED: didn't pass length/median quality thresholds.
   – OTHER: non–flu and contaminant/poor flu signal.

2. Percentages of all read patterns passing QC process
   – Patterns are clustered or non–redundant reads.
   – ASSEMBLED: excellent influenza read patterns.
   – UNUSABLE: poor or contaminant flu patterns.
   – CHIMERIC: flu patterns matching both strands.
   – NO MATCH: non–flu read patterns.

3. Percentages of assembled, merged–pair read counts
   – Shows the proportion of gene segments to the genome.
   – Paired–end reads have been merged into a single count unless not applicable: single–end reads have been used.

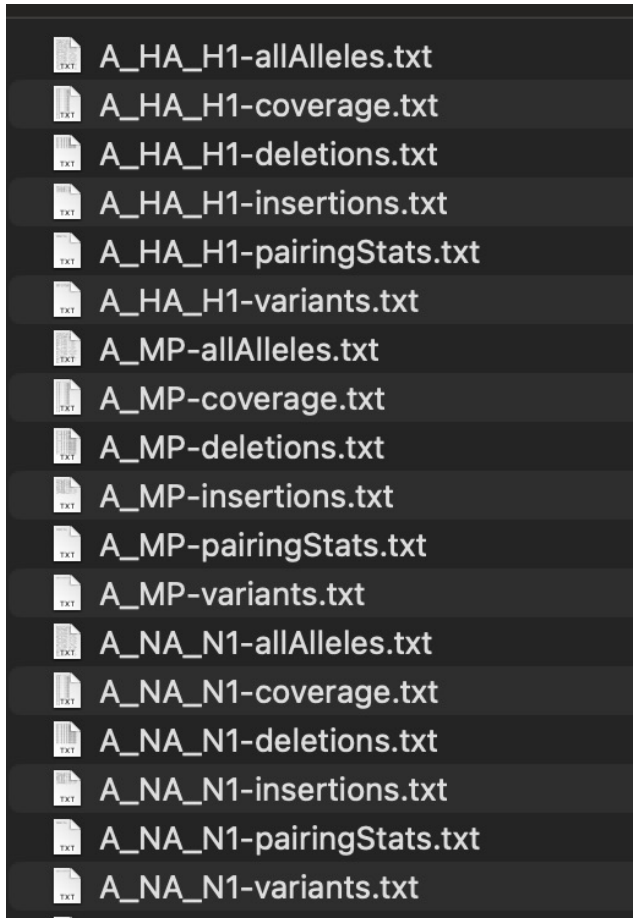


# IRMA outputs

| | | | |
|---|---|---|---|
| Amended consensus | Figures | Intermediate | Logs |
| Matrices | Secondary | Tables | Main data folder |

# Tables




## A_HA_H1-allAlleles.txt

| Reference_Name | Position | Allele | Count | Total | Frequency | Average_Quality | ConfidenceNotMacErr | PairedUB | QualityUB | Allele_Type |
|---|---|---|---|---|---|---|---|---|---|---|
| A_HA_H1 | 1 | A | 888 | 888 | 1 | 37.5078828828829 | 0.999822494541887 | 0.0255430148517389 | 0.00718698336856045 | Consensus |
| A_HA_H1 | 2 | T | 890 | 890 | 1 | 37.9820224719101 | 0.999840853257818 | 0.0255216051451136 | 0.00712365465317622 | Consensus |
| A_HA_H1 | 3 | G | 891 | 891 | 1 | 38.0347923681257 | 0.999842775304177 | 0.025510931638226 | 0.007111093682798 | Consensus |
| A_HA_H1 | 4 | G | 892 | 892 | 1 | 37.6457399103139 | 0.999828040565212 | 0.0255002789419813 | 0.0071424265893119 | Consensus |
| A_HA_H1 | 5 | A | 899 | 899 | 1 | 37.265850945495 | 0.999812321335043 | 0.0254262873508367 | 0.00713175805240175 | Consensus |
| A_HA_H1 | 6 | T | 1 | 902 | 0.00110864745011086 | 38 | 0.857042634040008 | 0.0253948824265928 | 0.00703314728625189 | Minority |
| A_HA_H1 | 6 | G | 901 | 902 | 0.998891352549889 | 37.4716981132075 | 0.9998208107568 | 0.0253948824265928 | 0.00708704021930316 | Consensus |
| A_HA_H1 | 7 | G | 904 | 905 | 0.998895027624309 | 36.837389380531 | 0.999792632252292 | 0.0253636585111471 | 0.007138623436904 | Consensus |
| A_HA_H1 | 7 | A | 1 | 905 | 0.00110497237569061 | 37 | 0.819428760495317 | 0.0253636585111471 | 0.00711883106767695 | Minority |
| A_HA_H1 | 8 | G | 1 | 922 | 0.00108459869848156 | 36 | 0.768404071014817 | 0.02519006284585 | 0.00713100016922393 | Minority |
| A_HA_H1 | 8 | C | 921 | 922 | 0.998915401301518 | 37.1216069489685 | 0.999805772555554 | 0.02519006284585 | 0.00698337277291578 | Consensus |
| A_HA_H1 | 9 | A | 927 | 927 | 1 | 37.2729234088457 | 0.999812626720059 | 0.0251400587893557 | 0.00693127453467275 | Consensus |
| A_HA_H1 | 10 | G | 1077 | 1077 | 1 | 36.5162488393686 | 0.999776963923812 | 0.0238293934757529 | 0.00612904825148871 | Consensus |
| A_HA_H1 | 11 | C | 2 | 1102 | 0.00181488203266788 | 27 | 0 | 0.0236411773022205 | 0.00978309679349504 | Minority |
| A_HA_H1 | 11 | T | 1100 | 1102 | 0.998185117967332 | 37.0236363636364 | 0.999801195929184 | 0.0236411773022205 | 0.00594041414185906 | Consensus |
| A_HA_H1 | 12 | A | 1113 | 1113 | 1 | 37.1967654986523 | 0.999809311962068 | 0.0235606741494169 | 0.00586698692904603 | Consensus |
| A_HA_H1 | 13 | C | 1151 | 1152 | 0.999131944444444 | 37.251954821894 | 0.999811556278106 | 0.0232859531920869 | 0.00567949510601033 | Consensus |
| A_HA_H1 | 13 | A | 1 | 1152 | 0.000868055555555556 | 17 | 0 | 0.0232859531920869 | 0.0356205802432355 | Minority |
| A_HA_H1 | 14 | C | 2 | 1177 | 0.00169923534409516 | 25 | 0 | 0.023118112019733 | 0.0114882450938743 | Minority |
| A_HA_H1 | 14 | T | 1175 | 1177 | 0.998300764655905 | 37.2936170212766 | 0.999813199990445 | 0.023118112019733 | 0.00556494379179234 | Consensus |
| A_HA_H1 | 15 | G | 2 | 1186 | 0.00168634064080944 | 38 | 0.906015833687056 | 0.0230591795557705 | 0.00545407877354896 | Minority |
| A_HA_H1 | 15 | A | 1184 | 1186 | 0.998313659359191 | 37.6875 | 0.999829398444496 | 0.0230591795557705 | 0.00548476703369799 | Consensus |
| A_HA_H1 | 16 | G | 1191 | 1192 | 0.999161073825503 | 37.6868178001679 | 0.999829516358246 | 0.0230203160326505 | 0.00545954087312827 | Consensus |
| A_HA_H1 | 16 | C | 1 | 1192 | 0.000838926174496644 | 39 | 0.849936090914135 | 0.0230203160326505 | 0.00534345231486696 | Minority |
| A_HA_H1 | 17 | T | 1197 | 1199 | 0.998331943286072 | 37.3024227234754 | 0.999813584182242 | 0.0229753975041523 | 0.00547101113686062 | Consensus |
| A_HA_H1 | 17 | C | 2 | 1199 | 0.00166805671392827 | 15.5 | 0 | 0.0229753975041523 | 0.0456896151571877 | Minority |
| A_HA_H1 | 18 | G | 1 | 1201 | 0.000832639467110741 | 35 | 0.620210453013778 | 0.0229626462002908 | 0.0057895039276099 | Minority |
| A_HA_H1 | 18 | A | 1200 | 1201 | 0.999167360532889 | 37.81 | 0.999834285022823 | 0.0229626462002908 | 0.00540973253400022 | Consensus |

# Conclusions

IRMA addresses viral diversity, which is critical for surveillance of rapidly evolving RNA viruses

IRMA can efficiently assemble reads across a range of coverage depths and phased minor variants

IRMA is customizable for different applications and organisms and provides a comprehensive set of analysis outputs

IRMA provides a comprehensive solution that addresses each aspect of NGS assembly, as it applies to RNA virus evolution, in a flexible and robust manner

IRMA has been used successfully to identify low frequency variants.

# Acknowledgements

**The creation of this training material was commissioned by ECDC to Statens Serum Institut (SSI) with the direct involvement of Marta Maria Ciucani**