Genomics Assembly and Analysis Training Module

Genomics Assembly and Analysis Training Module

Objectives

- Introduce the process of genome assembly and analysis, using Mate et. al. “Molecular Evidence of Sexual Transmission of Ebola Virus” as an example analysis.
- Starting with raw sequencing data, understand the steps for assembling a complete genome sequence.
- Compare multiple genome sequences.
- Use the output of the above analyses to draw conclusions about the biology of the samples.

Genomics Assembly and Analysis Training Module

Outline

- Brief review of Mate et. al. and Next Generation Sequencing.
- Step by step instructions for analyzing sequencing data using a Jupyter notebook.
- Glossary, FAQ, and complete breakdowns of all computational steps are provided at the end of this presentation.

ORIGINAL ARTICLE BRIEF REPORT

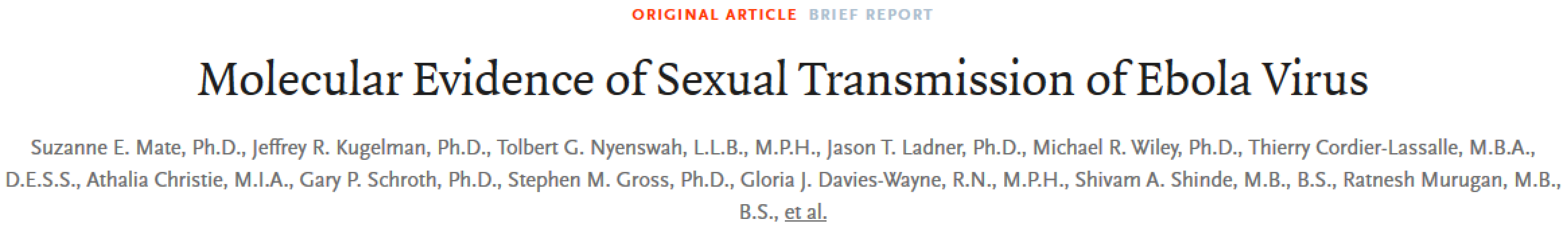
Molecular Evidence of Sexual Transmission of Ebola Virus

Suzanne E. Mate, Ph.D., Jeffrey R. Kugelman, Ph.D., Tolbert G. Nyenswah, L.L.B., M.P.H., Jason T. Ladner, Ph.D., Michael R. Wiley, Ph.D., Thierry Cordier-Lassalle, M.B.A., D.E.S.S., Athalia Christie, M.I.A., Gary P. Schroth, Ph.D., Stephen M. Gross, Ph.D., Gloria J. Davies-Wayne, R.N., M.P.H., Shivam A. Shinde, M.B., B.S., Ratnesh Murugan, M.B., B.S., et al.

- In Liberia, the partner of an Ebola survivor became sick.

- Did the partner contract Ebola through sexual transmission? Or through some other means?
- How can we tell?

- **These questions can be answered by sequencing.**



ORIGINAL ARTICLE BRIEF REPORT

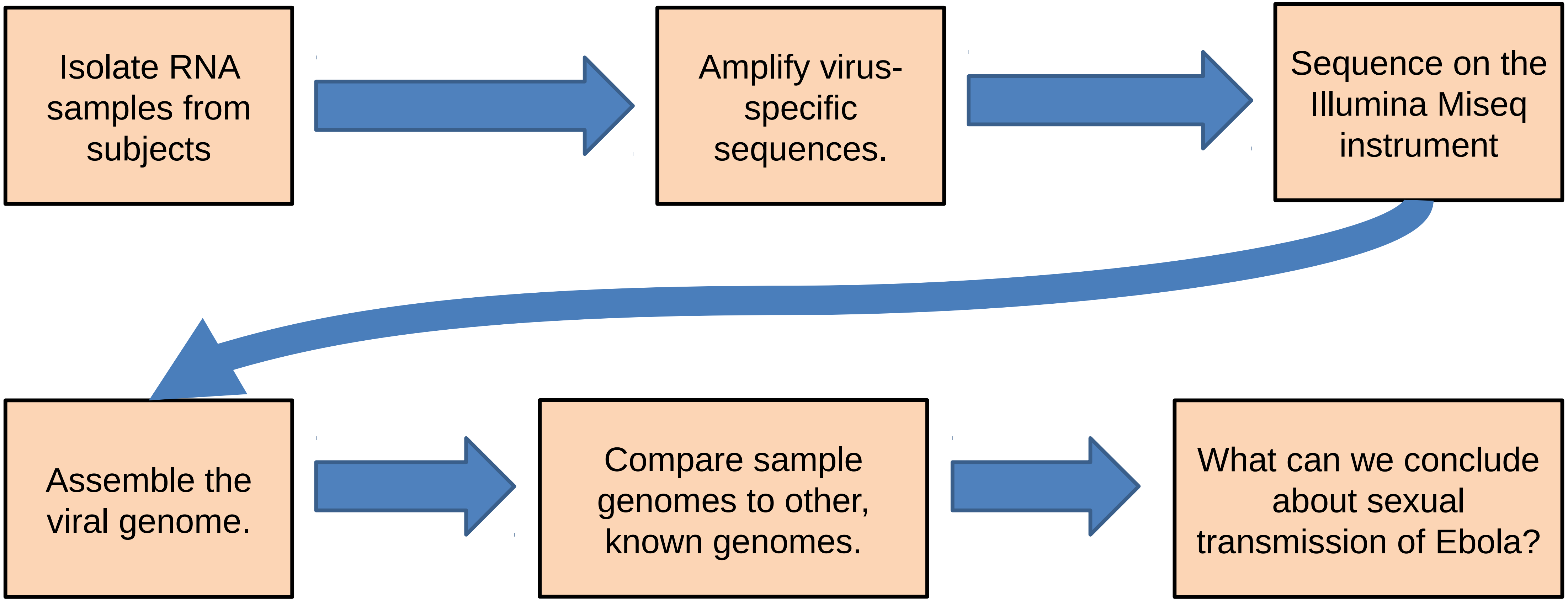
Molecular Evidence of Sexual Transmission of Ebola Virus

Suzanne E. Mate, Ph.D., Jeffrey R. Kugelman, Ph.D., Tolbert G. Nyenswah, L.L.B., M.P.H., Jason T. Ladner, Ph.D., Michael R. Wiley, Ph.D., Thierry Cordier-Lassalle, M.B.A., D.E.S.S., Athalia Christie, M.I.A., Gary P. Schroth, Ph.D., Stephen M. Gross, Ph.D., Gloria J. Davies-Wayne, R.N., M.P.H., Shivam A. Shinde, M.B., B.S., Ratnesh Murugan, M.B., B.S., et al.

To answer these questions, we can:

- Isolate virus from the survivor and their partner.
- Sequence the virus to discover the complete, accurate genome of each sample.
- Compare these sequences to each other and to other virus samples from this outbreak.
- Is the partner sample more similar to the survivor sequence? Or to the other samples from this outbreak?

We can answer these questions by following this outline:



Isolate RNA samples from subjects

Amplify virus-specific sequences.

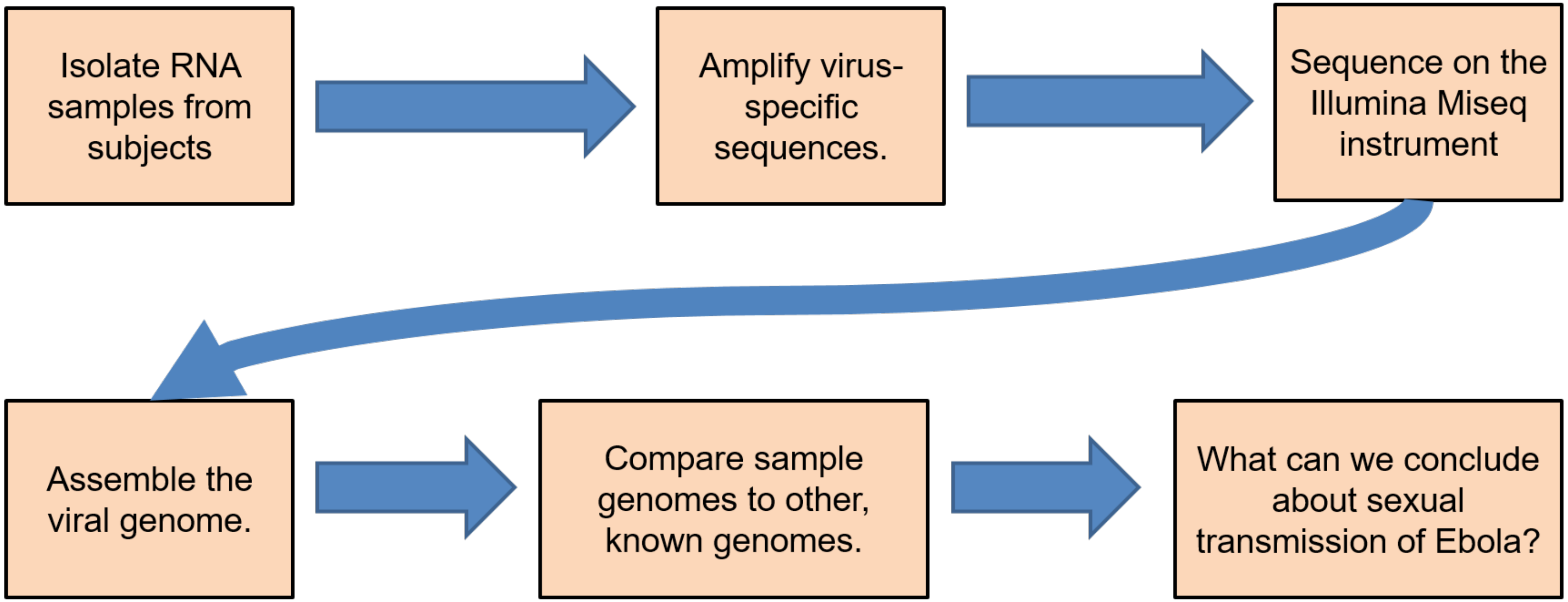
Sequence on the Illumina Miseq instrument

Assemble the viral genome.

Compare sample genomes to other, known genomes.

What can we conclude about sexual transmission of Ebola?

- This module focuses on the final three steps: analyzing the raw sequencing data.
- But it is important to first understand how the raw data is generated.



Isolate RNA samples from subjects

Amplify virus-specific sequences.

Sequence on the Illumina Miseq instrument

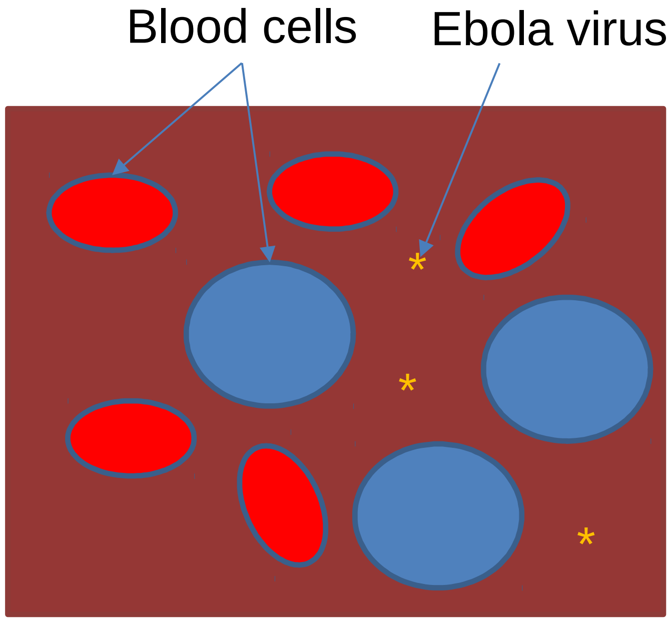
Assemble the viral genome.

Compare sample genomes to other, known genomes.

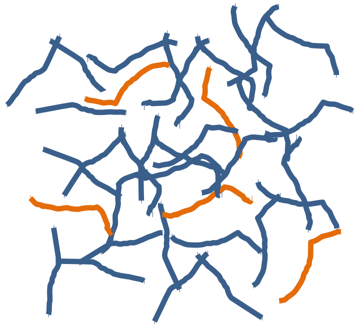
What can we conclude about sexual transmission of Ebola?

Isolate RNA samples from subjects

Collect samples



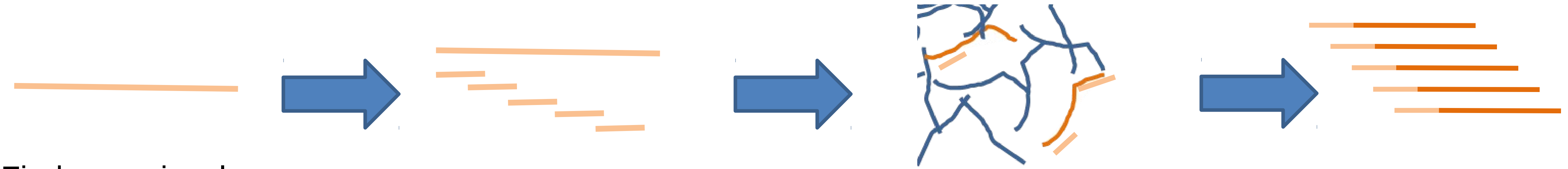
The samples contain mostly host cells.



Extract RNA, convert to DNA. It will be fragmented, and mostly from the host.

Amplify virus-specific sequences.

- Since most of the DNA will be from the host, we need to increase the proportion of viral DNA.



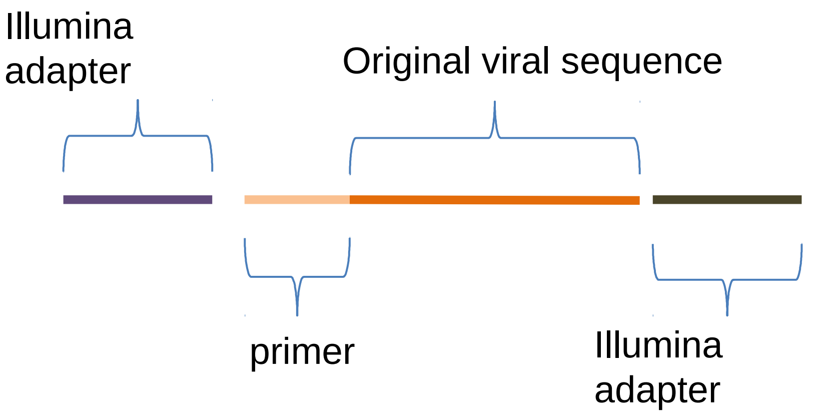
Find a previously sequenced Ebola genome (a “reference” sequence).

Make primers-short fragments of DNA-that match the reference genome.

The primers bind to the similar Ebola virus sequence found in our sample.

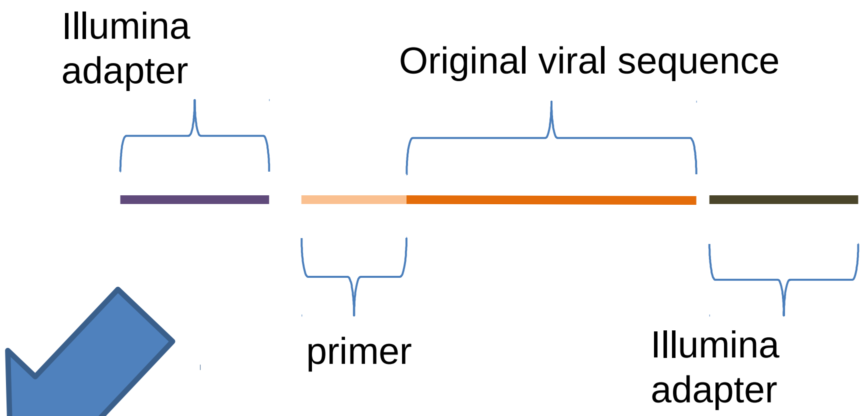
Primer-bound DNA is amplified, producing lots of viral DNA fragments.

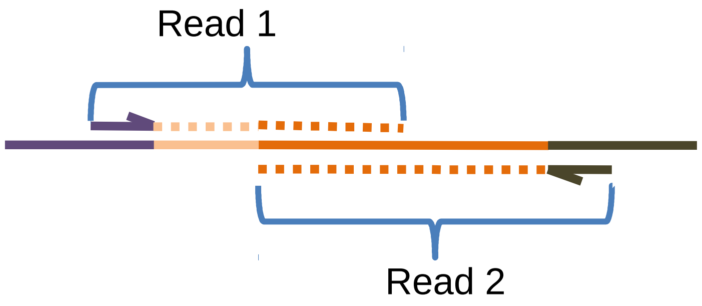
Sequence on the Illumina Miseq instrument



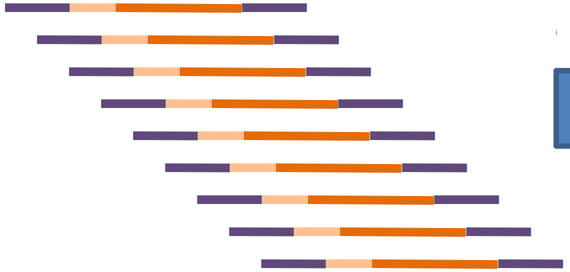
- **Add adapters to both ends of the DNA fragment to be sequenced.**
- **These are DNA sequences necessary for sequencing on the Illumina Miseq.**
- **The pool of DNA to be sequenced is known as a “library.”**

Sequence on the Illumina Miseq instrument





Each piece of DNA is sequenced twice, starting from either end (Paired end sequencing). The products of a sequencing run are called “reads.” The sequence and quality information are stored in .fastq files.

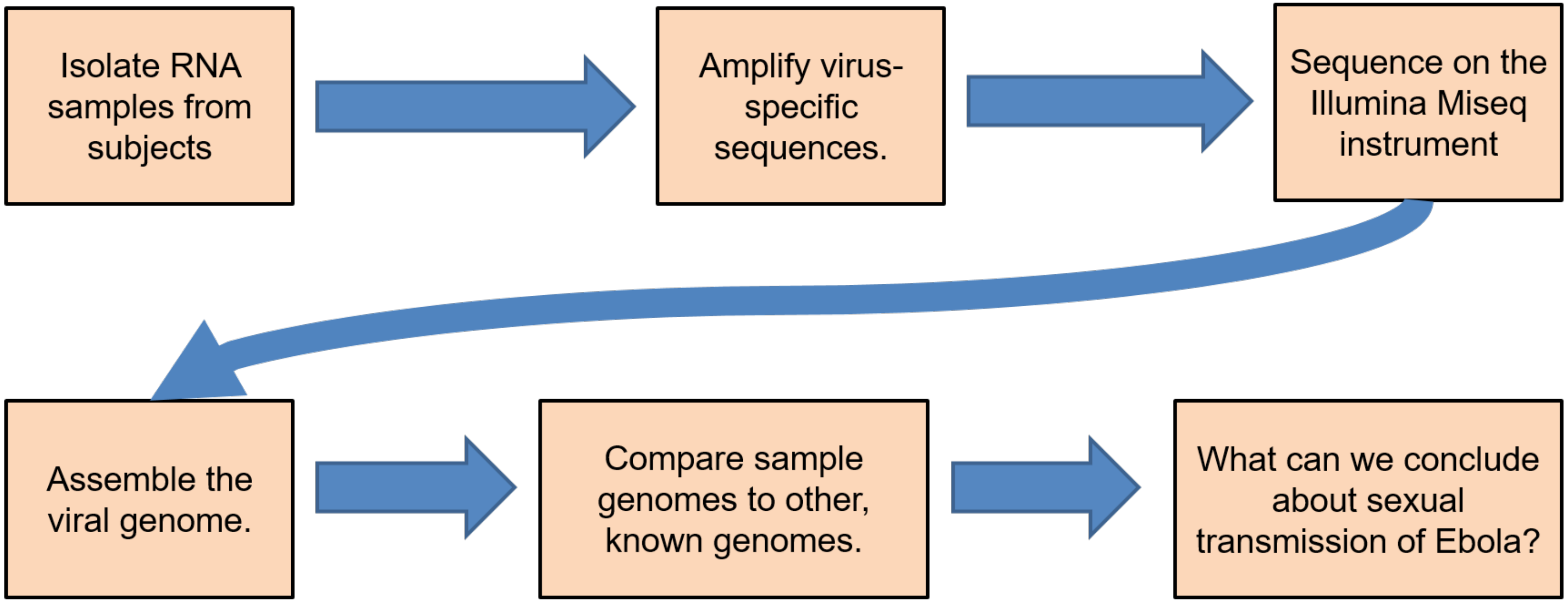


Library-DNA fragments ready to be sequenced

Illumina MiSeq

Raw data, in .fastq format.

Now we can assemble the raw data produced by the Miseq.



Isolate RNA samples from subjects

Amplify virus-specific sequences.

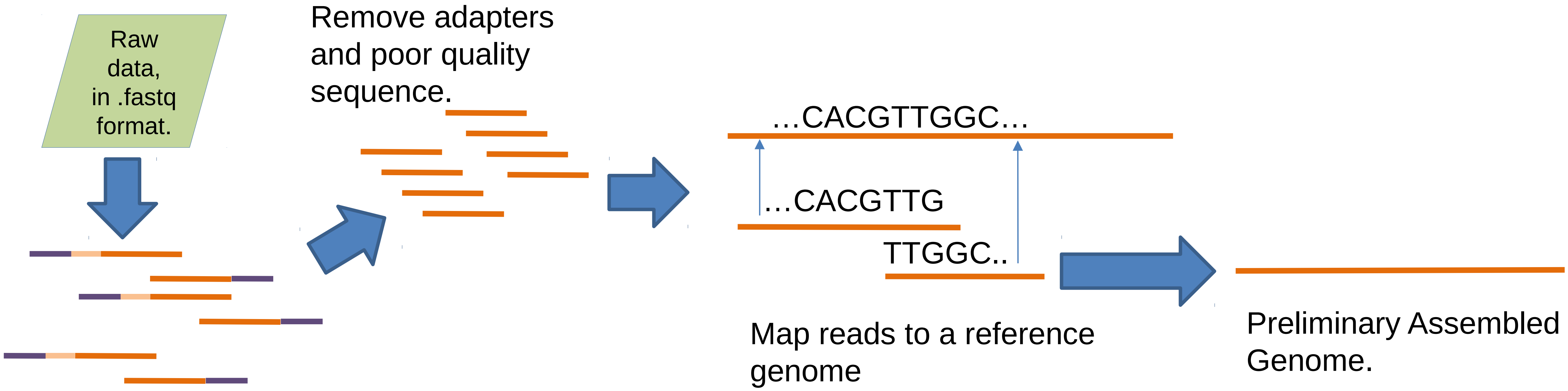
Sequence on the Illumina Miseq instrument

Assemble the viral genome.

Compare sample genomes to other, known genomes.

What can we conclude about sexual transmission of Ebola?

Assembly: Broad Overview of the Computational Steps



Raw data, in .fastq format.

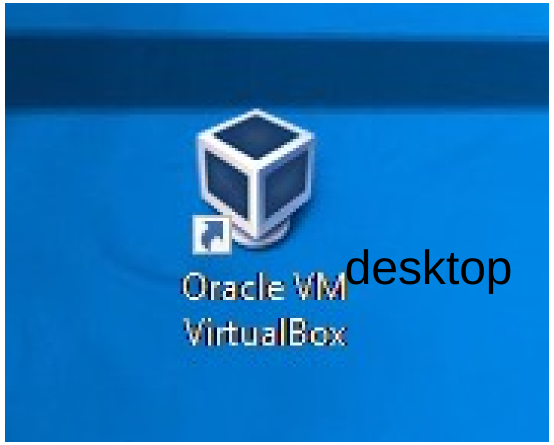
Remove adapters and poor quality sequence.

Map reads to a reference genome

Preliminary Assembled Genome.

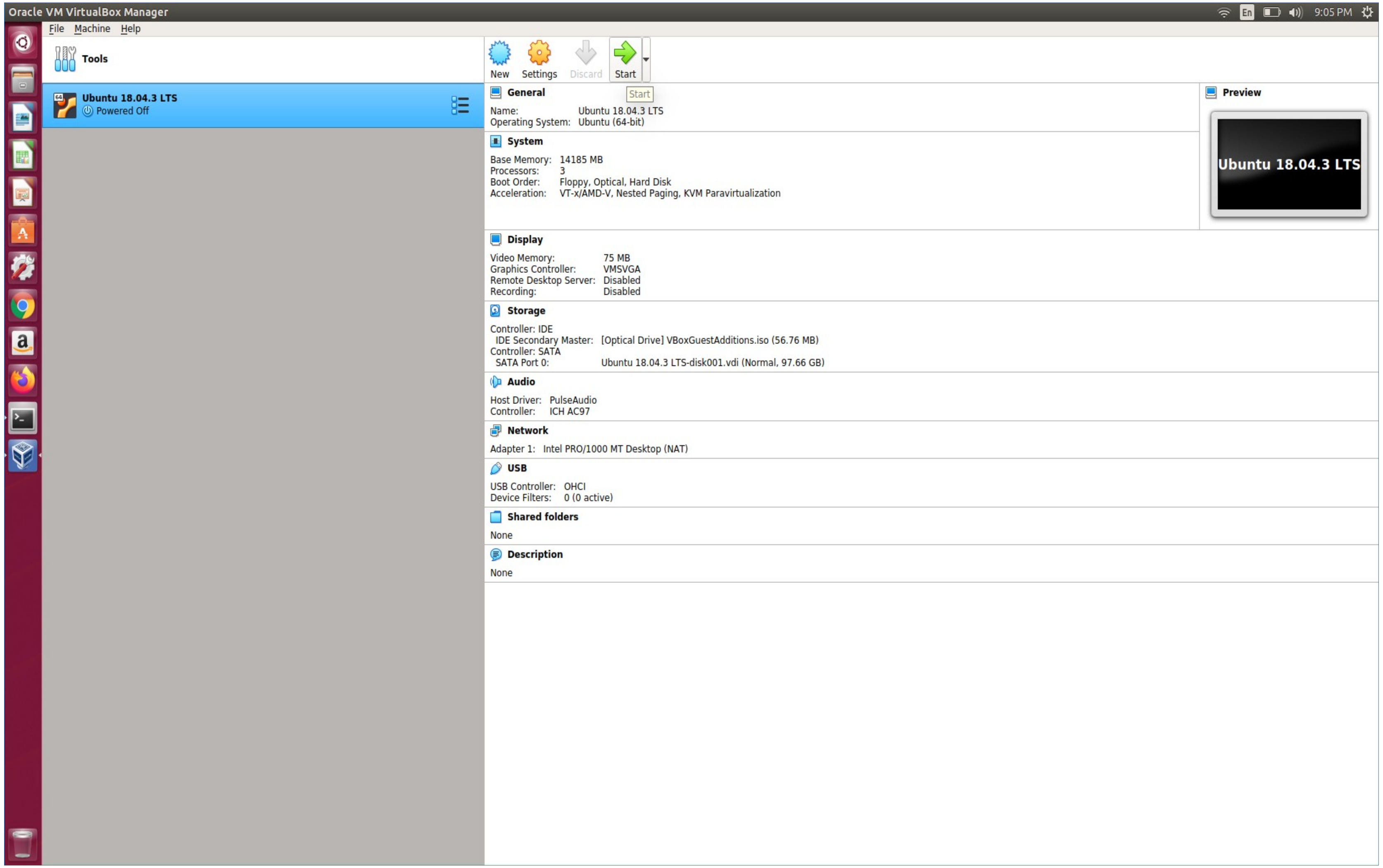
Raw data consists of sequences containing fragments of the Ebola genome. Ultimately, we need to take these fragments and assemble them into the complete genome.

Prepare for assembly: Open a virtual machine, which contains all of the data and programs you need to complete the module.



desktop

Click on the Oracle VM VirtualBox icon



Ubuntu 18.04.3 LTS
Powered Off

General

Name: Ubuntu 18.04.3 LTS
Operating System: Ubuntu (64-bit)

System

Base Memory: 14185 MB
Processors: 3
Boot Order: Floppy, Optical, Hard Disk
Acceleration: VT-x/AMD-V, Nested Paging, KVM Paravirtualization

Display

Video Memory: 75 MB
Graphics Controller: VMSVGA
Remote Desktop Server: Disabled
Recording: Disabled

Storage

Controller: IDE
IDE Secondary Master: [Optical Drive] VBoxGuestAdditions.iso (56.76 MB)
Controller: SATA
SATA Port 0: Ubuntu 18.04.3 LTS-disk001.vdi (Normal, 97.66 GB)

Audio

Host Driver: PulseAudio
Controller: ICH AC97

Network

Adapter 1: Intel PRO/1000 MT Desktop (NAT)

USB

USB Controller: OHCI
Device Filters: 0 (0 active)

Shared folders

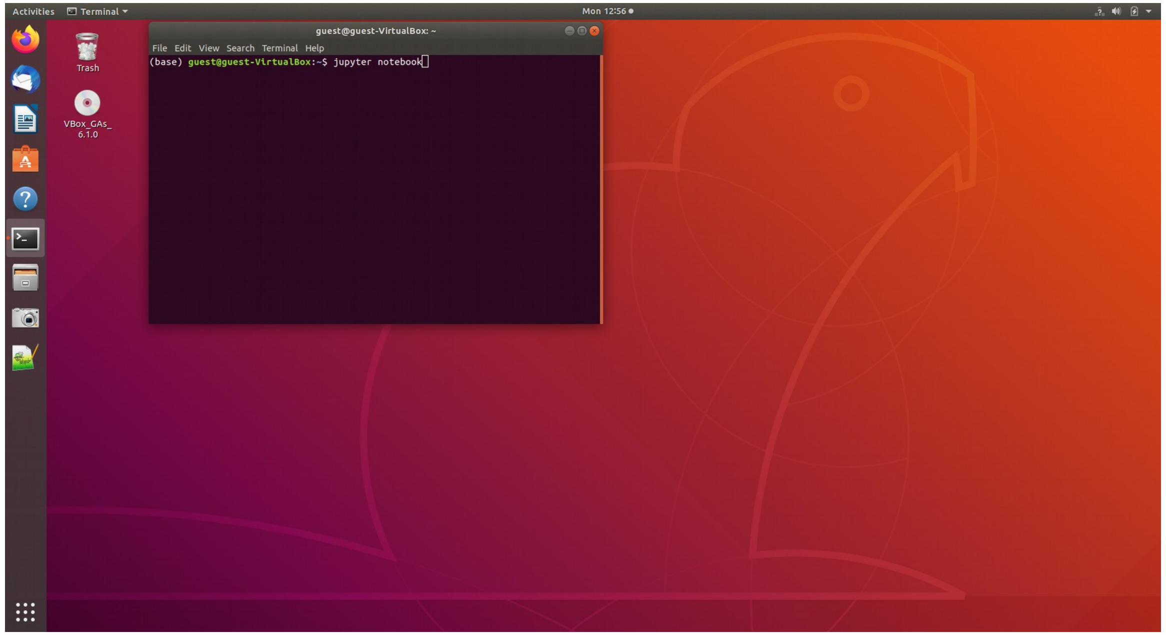
None

Description

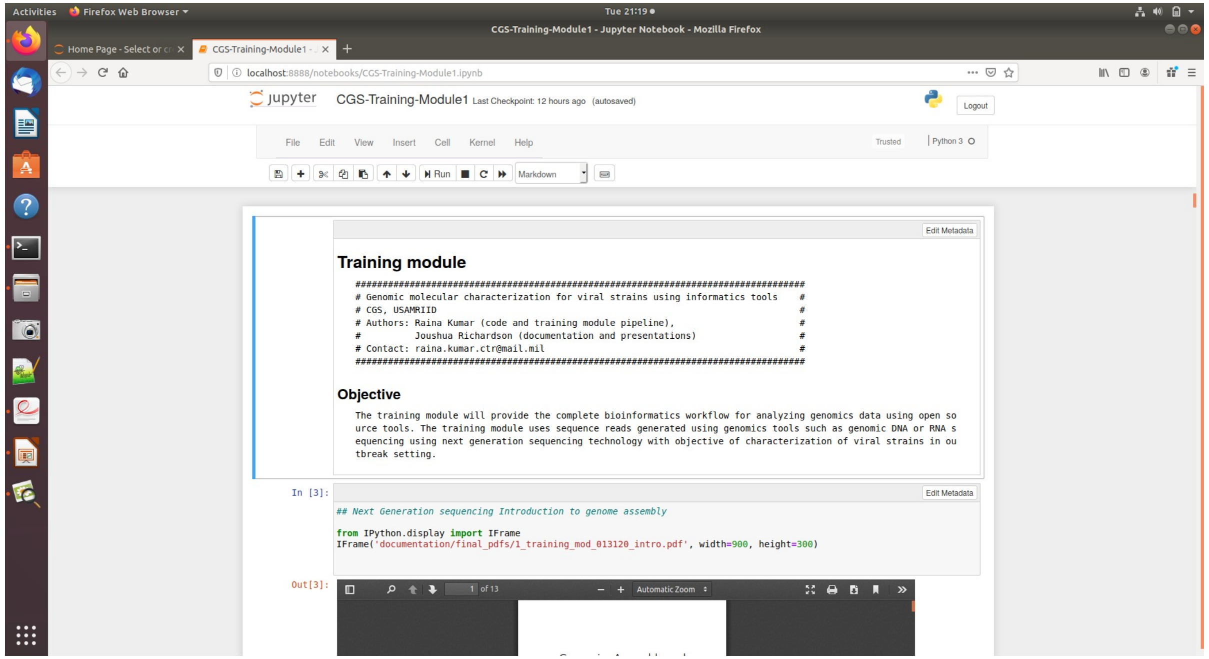
None

Open the Jupyter notebook.

- Run through each step of the Jupyter notebook, examining any slides embedded in the step.
- Steps that advance the pipeline will be accompanied by an explanation slide, which is detailed next.


Activities
Terminal
Mon 12:56
Trash
VBox_GAs_
6.1.0
guest@guest-VirtualBox: ~
File Edit View Search Terminal Help

```
(base) guest@guest-VirtualBox:~$ jupyter notebook
```



Training module

```
##################################################################################
# Genomic molecular characterization for viral strains using informatics tools    #
# CGS, USAMRIID                                                                   #
# Authors: Raina Kumar (code and training module pipeline),                       #
#          Joushua Richardson (documentation and presentations)                   #
# Contact: raina.kumar.ctr@mail.mil                                               #
##################################################################################
```

Objective

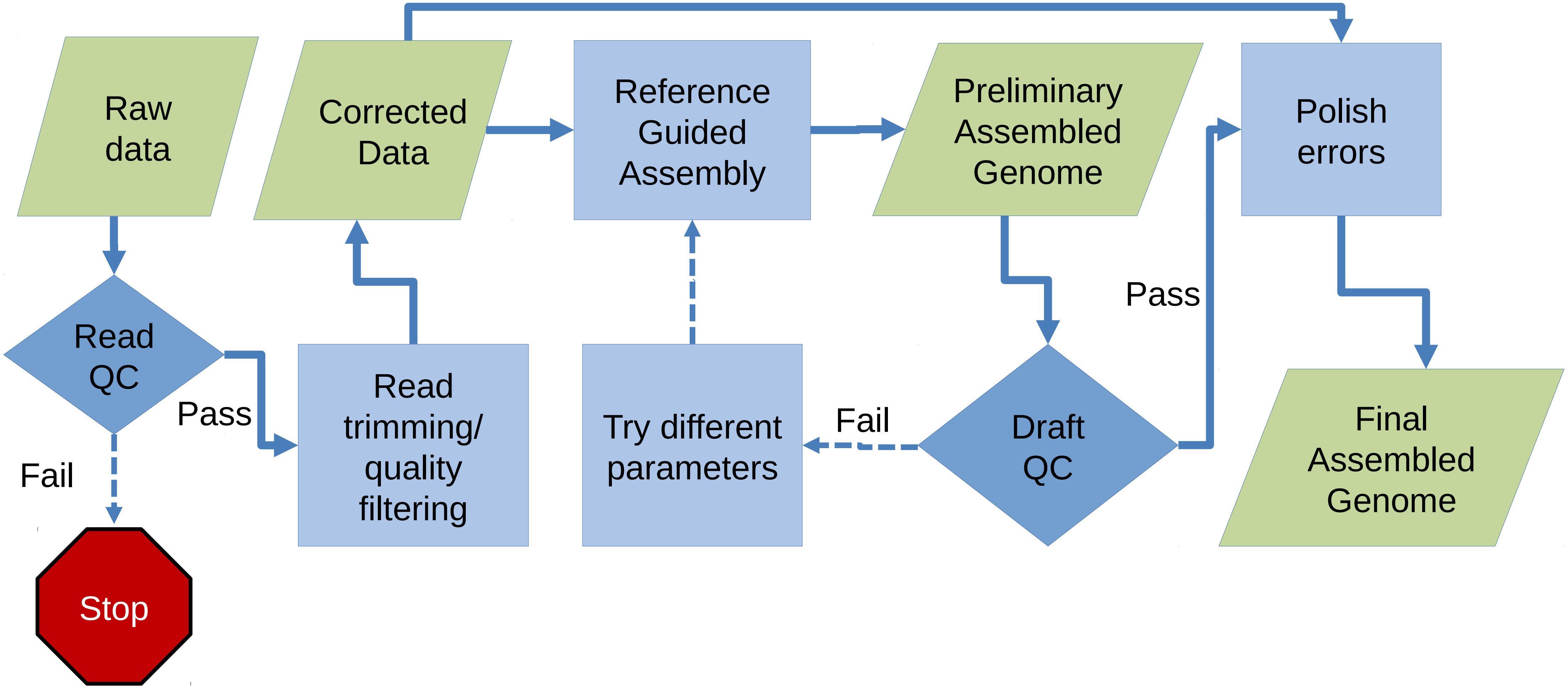
The training module will provide the complete bioinformatics workflow for analyzing genomics data using open source tools. The training module uses sequence reads generated using genomics tools such as genomic DNA or RNA sequencing using next generation sequencing technology with objective of characterization of viral strains in outbreak setting.

In [3]:

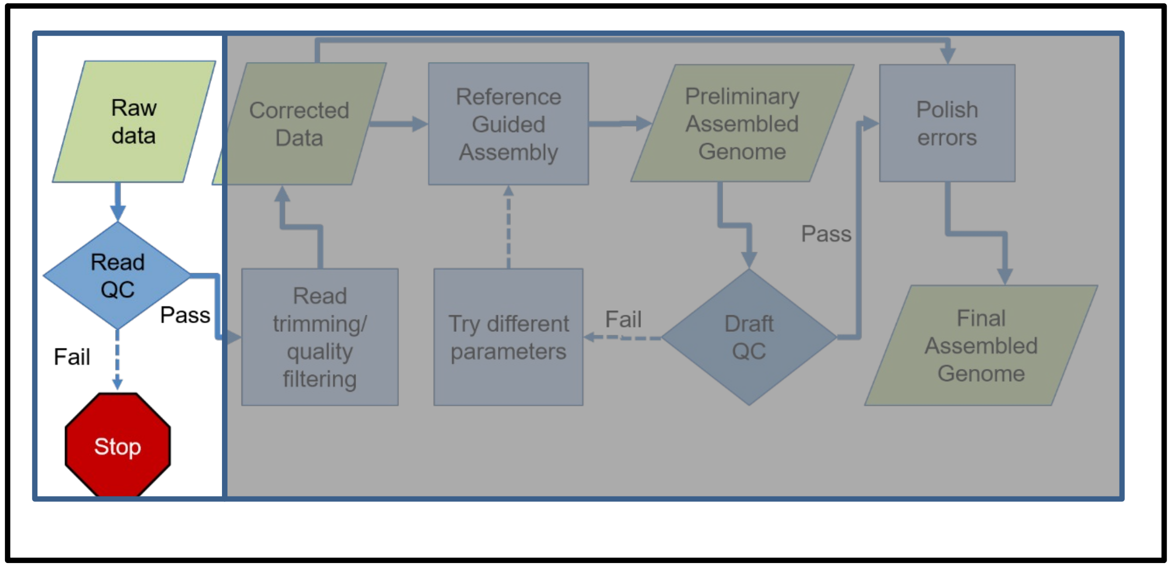
```
## Next Generation sequencing Introduction to genome assembly

from IPython.display import IFrame
IFrame('documentation/final_pdfs/1_training_mod_013120_intro.pdf', width=900, height=300)
```

Out[3]:

Assembly pipeline: How we get from raw data to the final assembled genome.

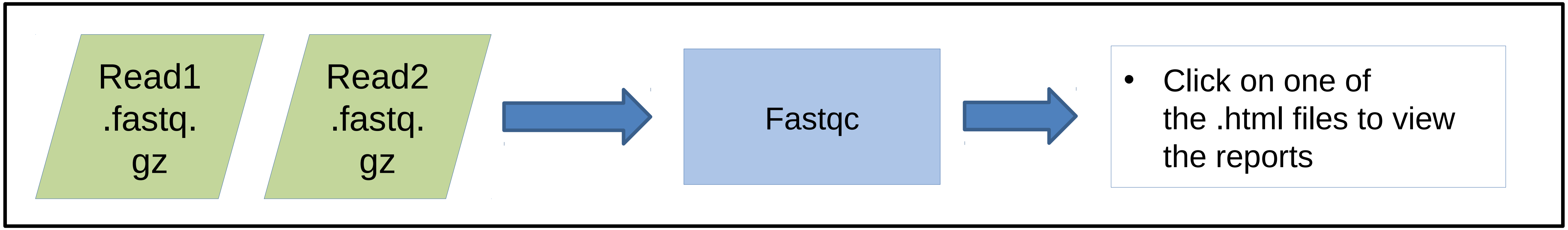


Location in the overall pipeline



- Start with raw sequencing data, in fastq format and zipped.
- Remember, there are two reads for each DNA fragment. The first read of each fragment is stored in one file, and the second read of each fragment is stored in another.
- Run Fastqc, a program that summarizes the quality of reads. Also outputs a number of useful metrics.

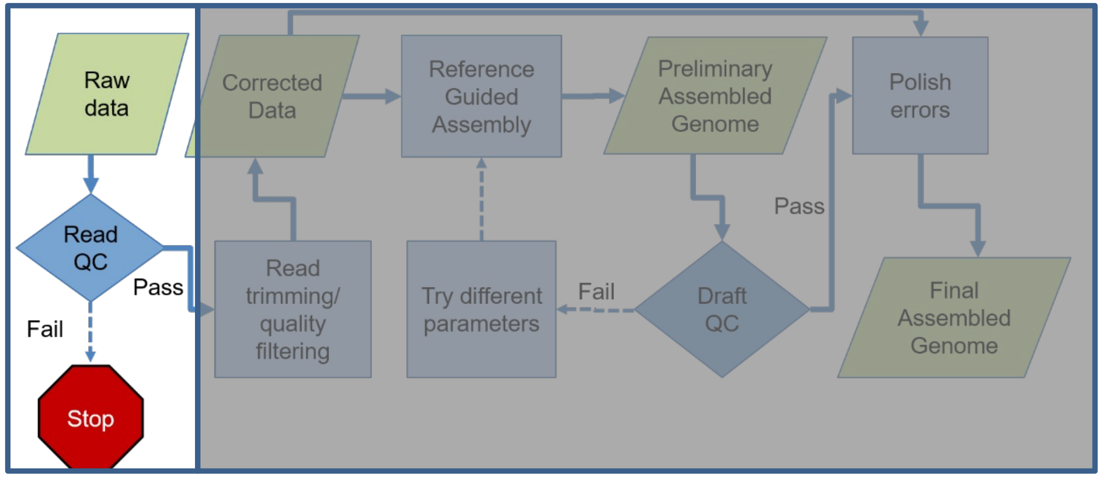
Plain English description of the steps in the pipeline.



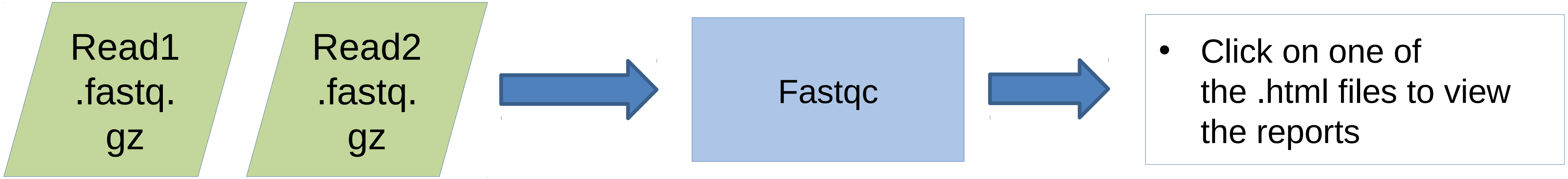
- Click on one of the .html files to view the reports

Inputs and outputs for the current step of the pipeline.

Fastqc



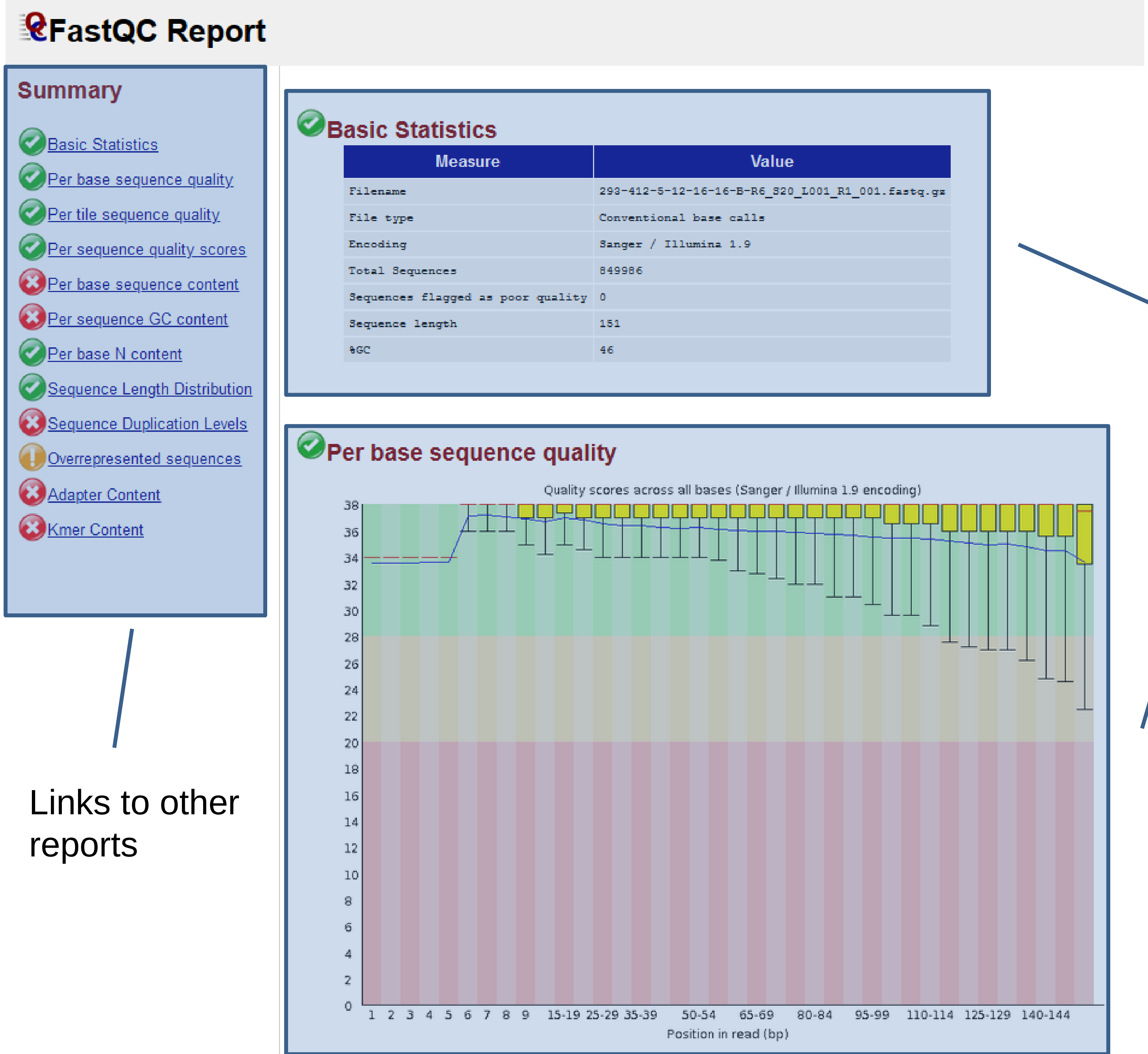
- **Start with raw sequencing data, in fastq format and zipped.**
- **Remember, there are two reads for each DNA fragment. The first read of each fragment is stored in one file, and the second read of each fragment is stored in another.**
- **Run Fastqc, a program that summarizes the quality of reads. Also outputs a number of useful metrics.**


Read1 .fastq. gz

Read2 .fastq. gz

Fastqc

- Click on one of the .html files to view the reports



FastQC Report

Summary

- ✓ Basic Statistics
- ✓ Per base sequence quality
- ✓ Per tile sequence quality
- ✓ Per sequence quality scores
- ✗ Per base sequence content
- ✗ Per sequence GC content
- ✓ Per base N content
- ✓ Sequence Length Distribution
- ✗ Sequence Duplication Levels
- ! Overrepresented sequences
- ✗ Adapter Content
- ✗ Kmer Content

✓ Basic Statistics

Measure	Value
Filename	293-412-5-12-16-16-B-R6_S20_L001_R1_001.fastq.gz
File type	Conventional base calls
Encoding	Sanger / Illumina 1.9
Total Sequences	849986
Sequences flagged as poor quality	0
Sequence length	151
%GC	46

✓ Per base sequence quality

Links to other reports

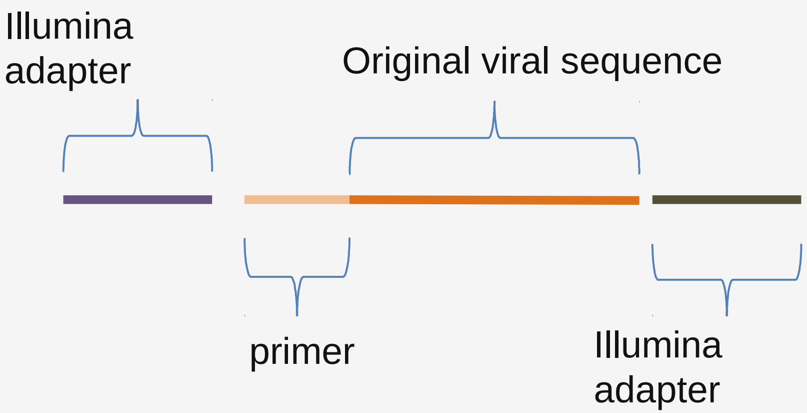
Fastqc Report

Preliminary information, number of sequences in file, average sequence length, etc.

Across all sequences, at each base position, what is the average quality score?

Quality>30 is good for most purposes.

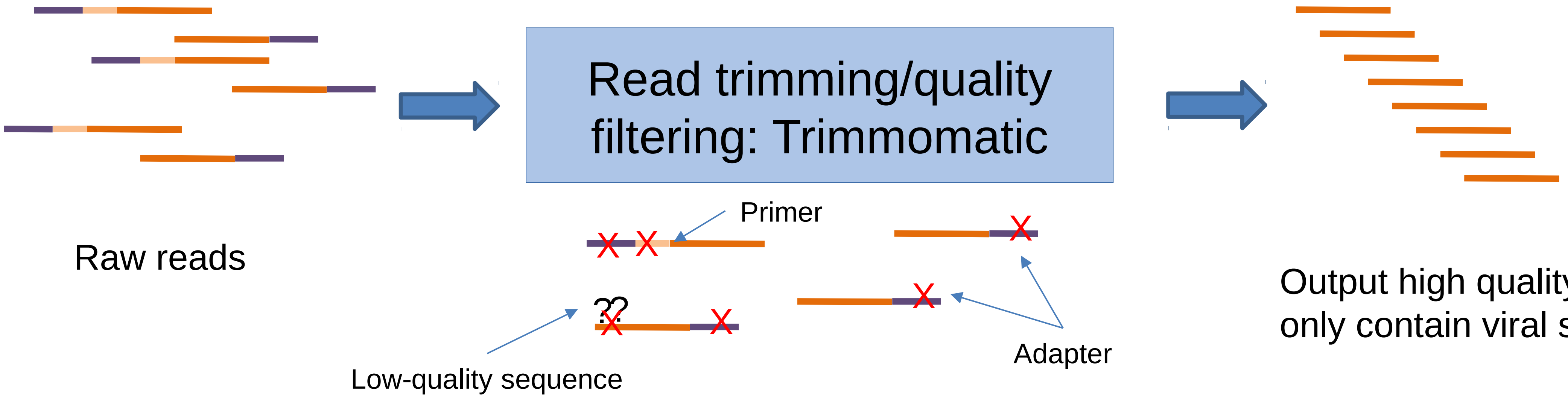
The quality scores are high at each position of the read. We can proceed with the analysis.



- **Add adapters to both ends of the DNA fragment to be sequenced.**
- **These are DNA sequences necessary for sequencing on the Illumina Miseq.**
- **The pool of DNA to be sequenced is known as a “library.”**

- **Remember that sequences were added during sample and library preparations that are not part of the original viral sequence.**
- **We need to remove those sequences now.**

Read trimming/quality filtering



Raw reads

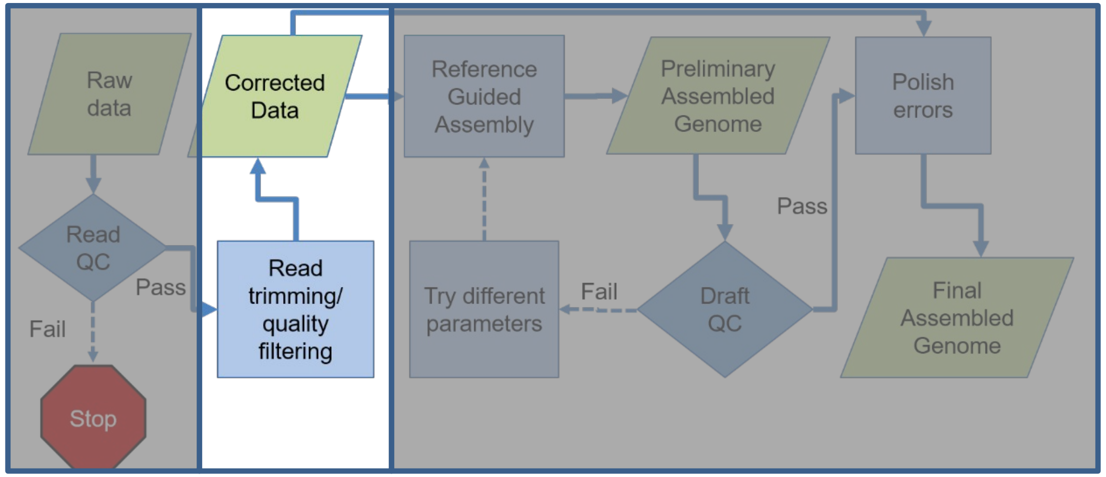
Read trimming/quality filtering: Trimmomatic

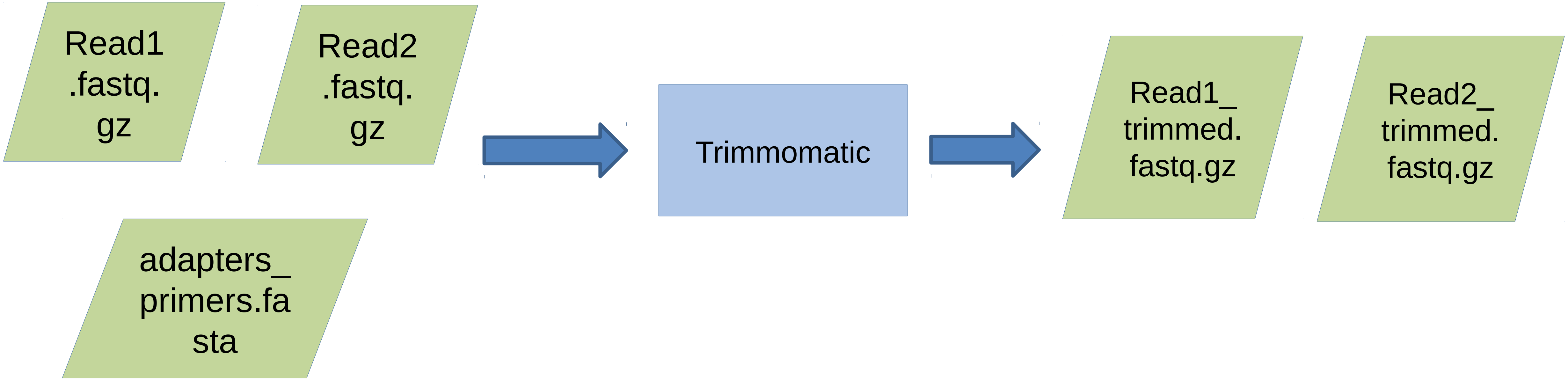
Output high quality reads that only contain viral sequence.

Find and remove adapters, primers and any low quality sequence

Read Trimming and Quality Control

- **Take raw reads and a list of sequences added during library prep.**
- **Remove those sequences, and any sequence of low quality**





Read1 .fastq. gz

Read2 .fastq. gz

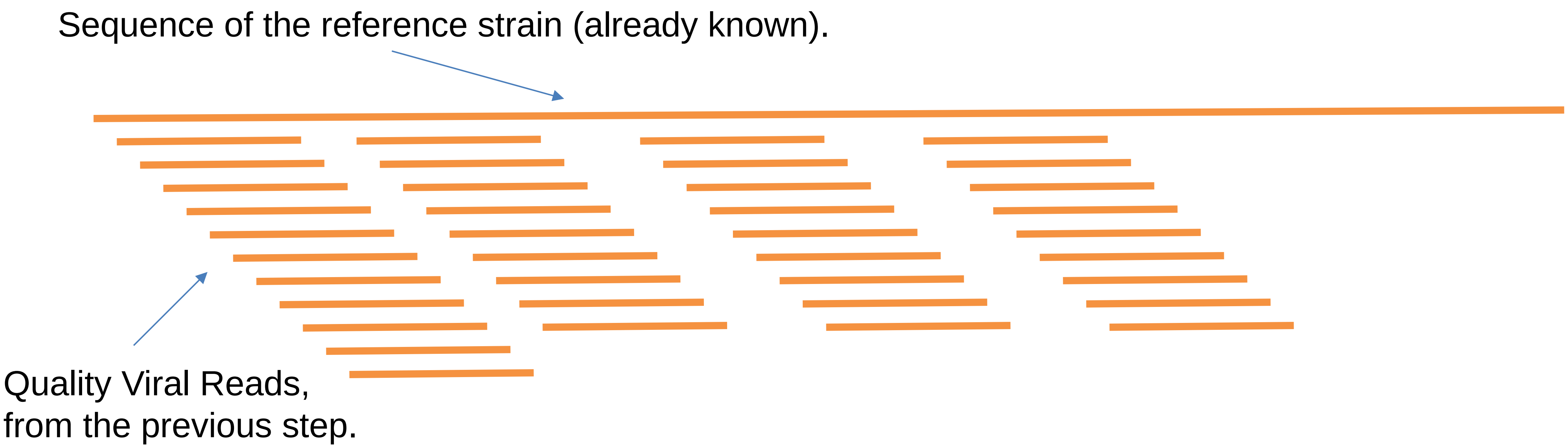
adapters_ primers.fa sta

Trimmomatic

Read1_ trimmed. fastq.gz

Read2_ trimmed. fastq.gz

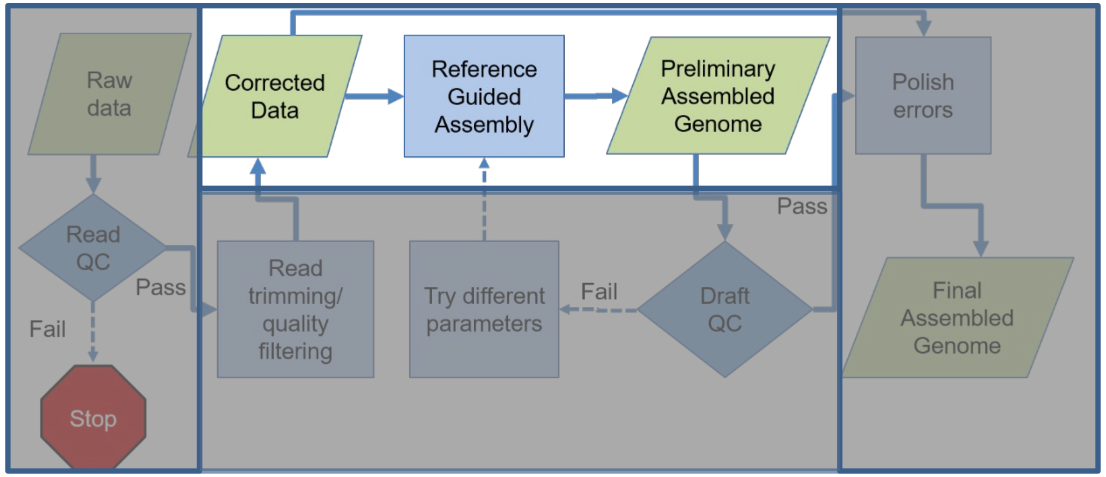
We can now align the viral reads to a known reference sequence.


Sequence of the reference strain (already known).

Quality Viral Reads,
from the previous step.

- **Use the “bwa” program to map the viral reads to the known reference assembly.**

Reference Guided Assembly, pt. 1



Raw data

Read QC

Pass

Fail

Stop

Corrected Data

Read trimming/ quality filtering

Reference Guided Assembly

Try different parameters

Preliminary Assembled Genome

Fail

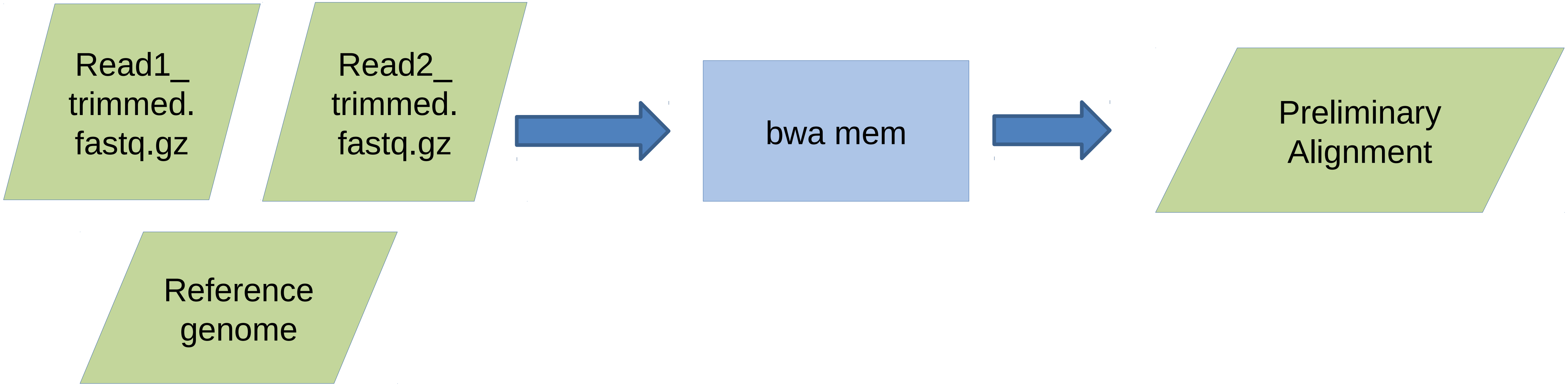
Draft QC

Pass

Polish errors

Final Assembled Genome

- Start with quality reads and a reference genome.
- The reference genome is the known sequence from the same species.
- Reads are mapped to the reference genome, creating a preliminary alignment



Read1_ trimmed. fastq.gz

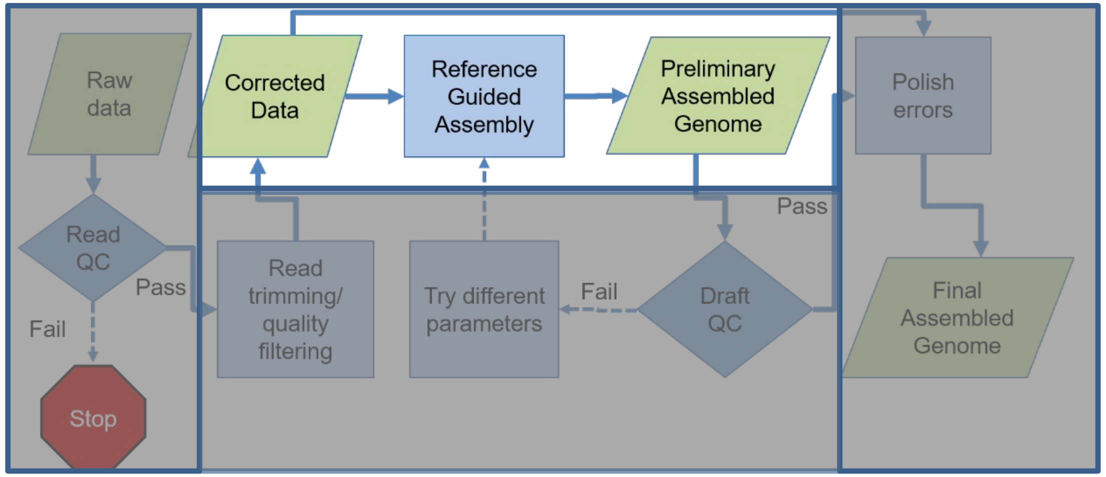
Read2_ trimmed. fastq.gz

Reference genome

bwa mem

Preliminary Alignment

Reference Guided Assembly, pt. 2



Raw data

Read QC

Pass

Fail

Stop

Corrected Data

Reference Guided Assembly

Preliminary Assembled Genome

Read trimming/ quality filtering

Try different parameters

Fail

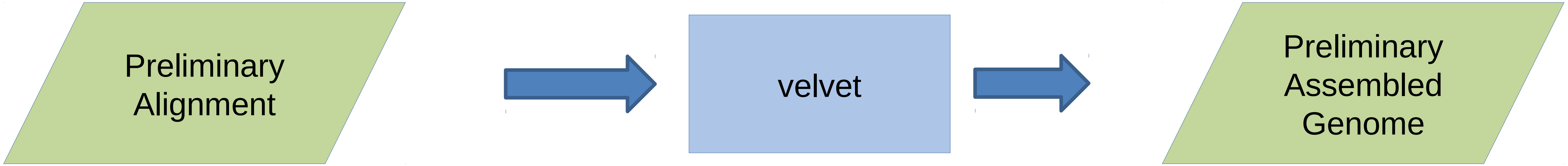
Draft QC

Pass

Polish errors

Final Assembled Genome

- Starting with the preliminary alignment, run the velveth and velvetg programs.
- These programs compare the aligned reads to the reference genome and output a preliminary assembled genome.
- This preliminary assembly is now our best guess of the genome sequence of the virus isolated from our sample.

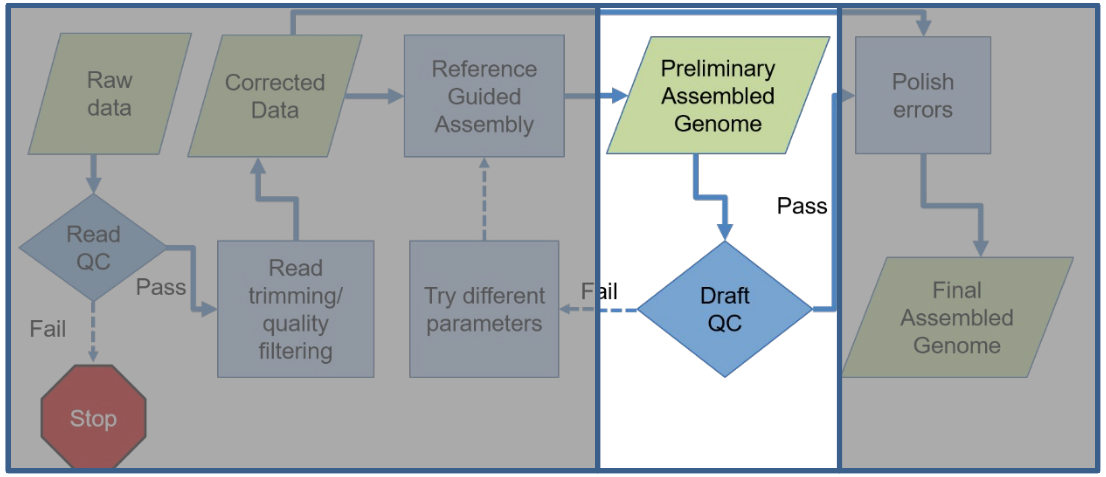


Preliminary Alignment

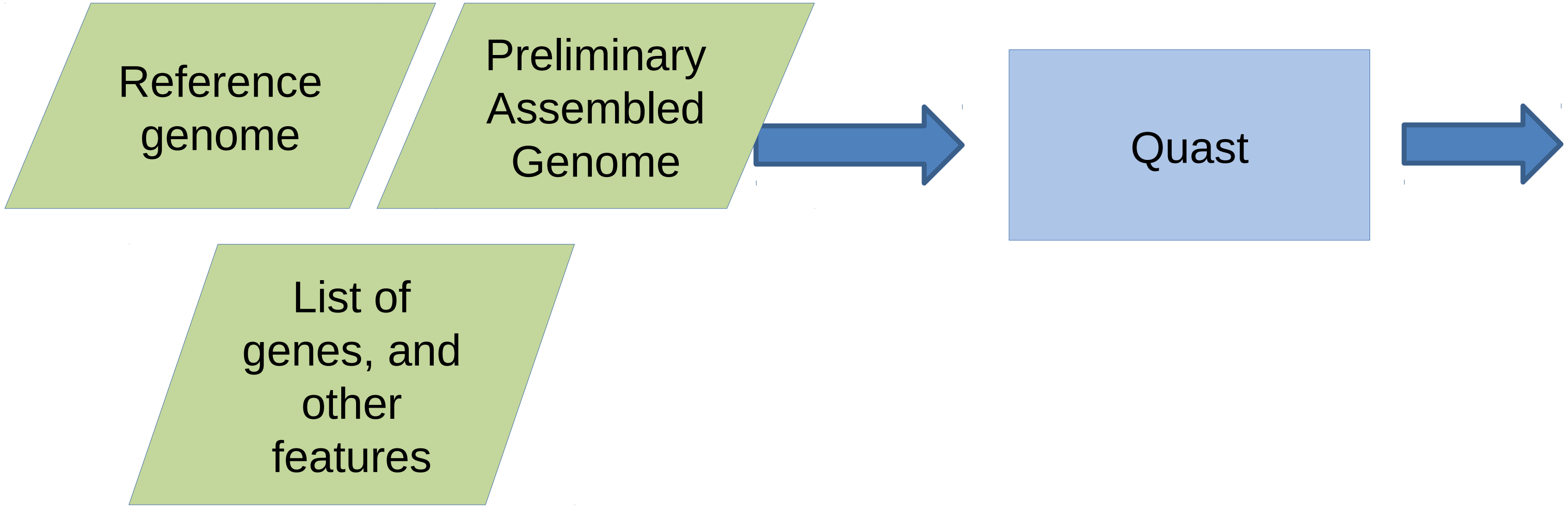
velvet

Preliminary Assembled Genome

Quality Control of the Preliminary Assembled Genome

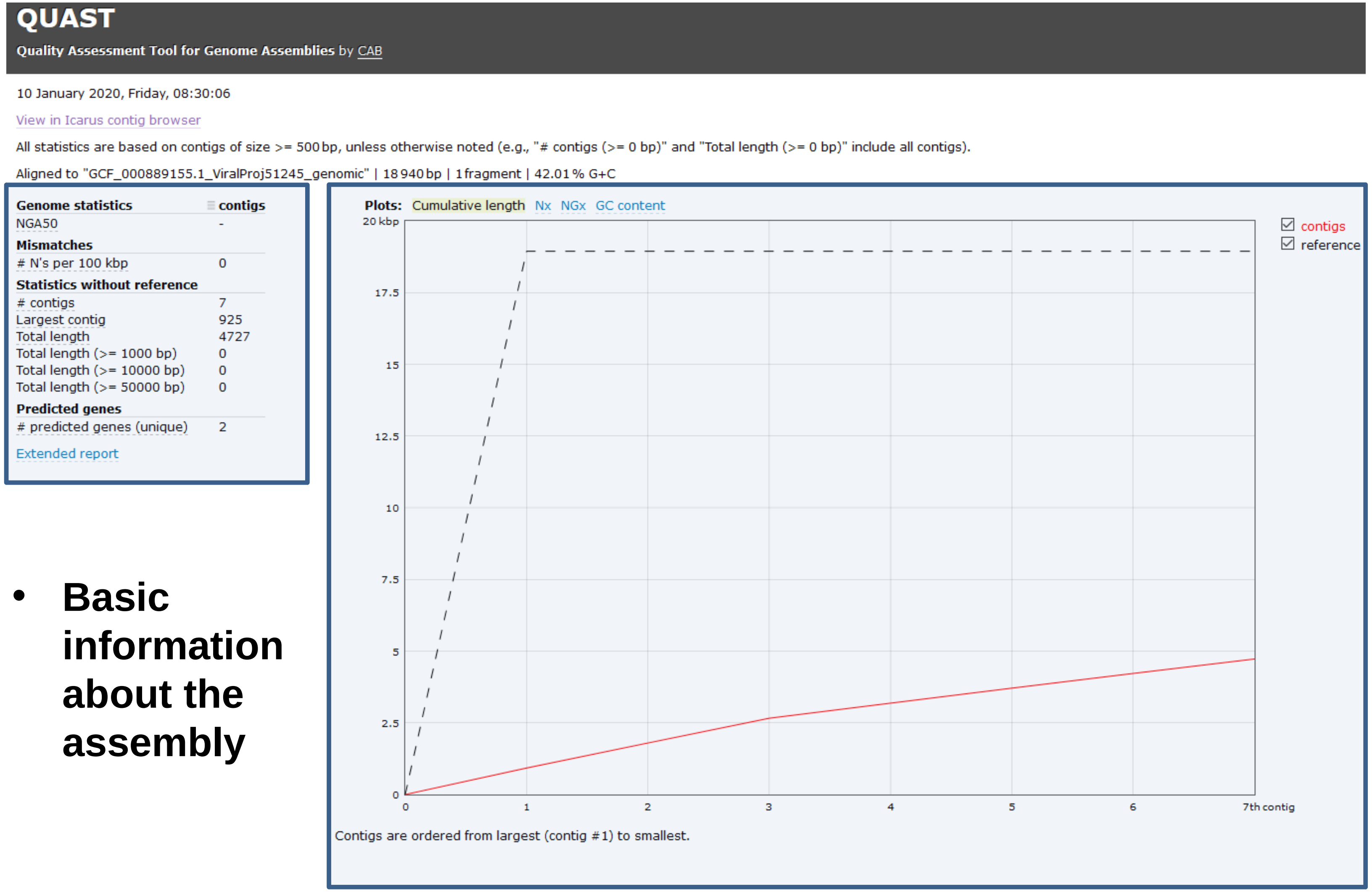


- **Quast compares the preliminary assembly to a known genome from the same species.**
- **Also, identifies functional sequences, like genes and RNAs.**


Reference genome
Preliminary Assembled Genome
List of genes, and other features
Quast

- Click on the .html file to open up the quast output.

Assessing genome quality with QUAST. FIX



QUAST

Quality Assessment Tool for Genome Assemblies by CAB

10 January 2020, Friday, 08:30:06

View in Icarus contig browser

All statistics are based on contigs of size >= 500 bp, unless otherwise noted (e.g., "# contigs (>= 0 bp)" and "Total length (>= 0 bp)" include all contigs).

Aligned to "GCF_000889155.1_ViralProj51245_genomic" | 18 940 bp | 1 fragment | 42.01 % G+C

Genome statistics	**contigs**
NGA50	-
Mismatches	
# N's per 100 kbp	0
Statistics without reference	
# contigs	7
Largest contig	925
Total length	4727
Total length (>= 1000 bp)	0
Total length (>= 10000 bp)	0
Total length (>= 50000 bp)	0
Predicted genes	
# predicted genes (unique)	2

Extended report

Plots: Cumulative length Nx NGx GC content

Contigs are ordered from largest (contig #1) to smallest.

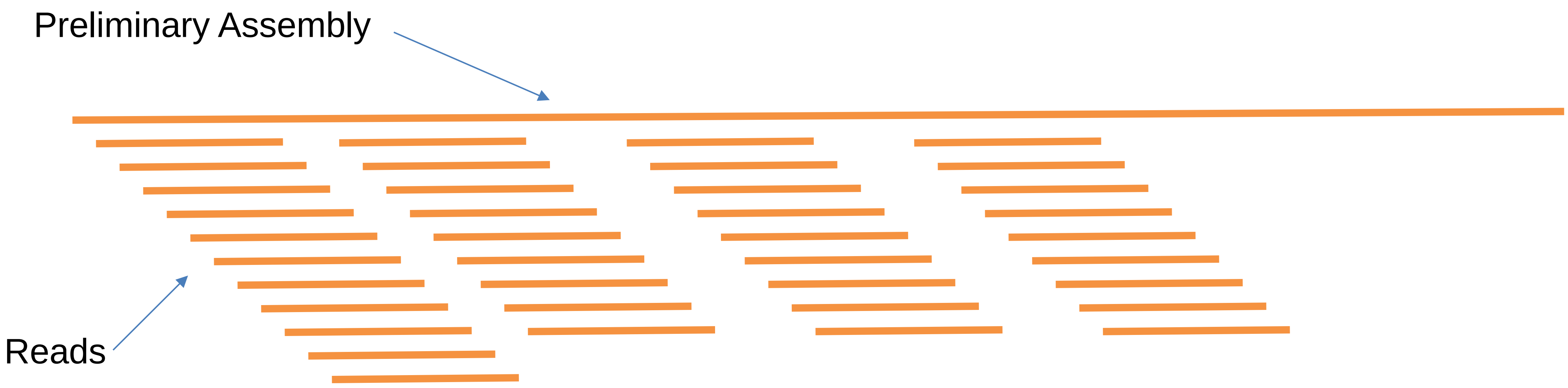
- **Basic information about the assembly**

- **Compare the length of the reference to the cumulative length of the contigs.**
- **Could the preliminary assembly contain a complete genome?**

We now have a preliminary assembly

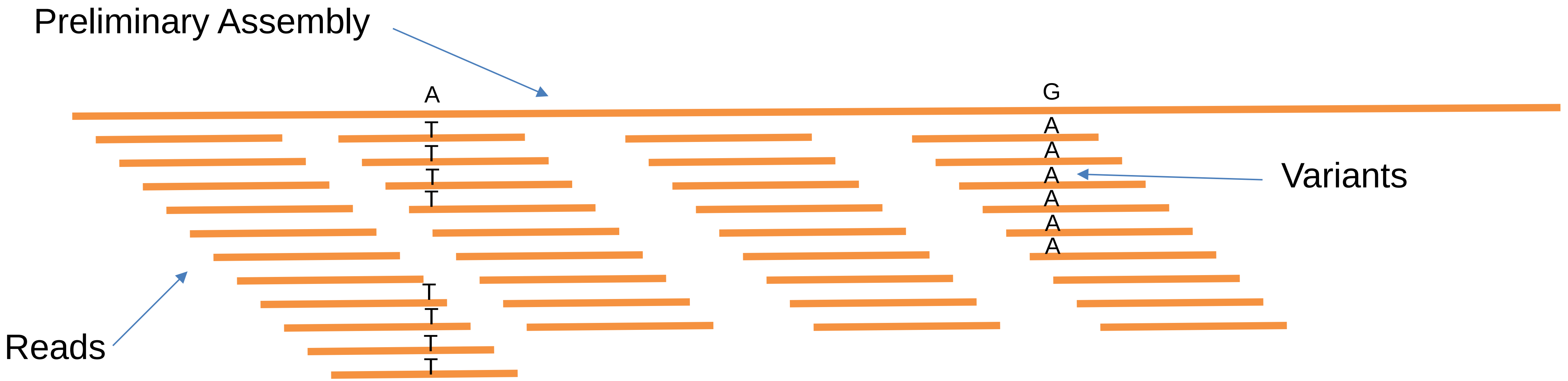
Preliminary Assembly

Improve the quality of the assembly: Polishing



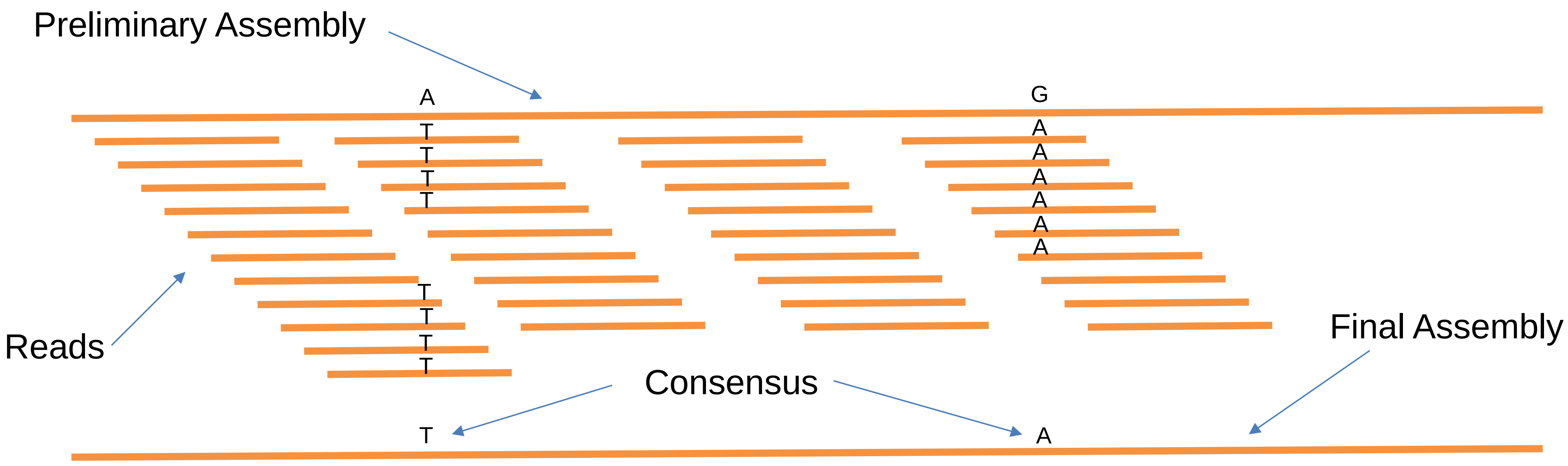
- **We can map the quality filtered reads to this preliminary assembly.**
- **This is similar to the initial read mapping to the reference genome done previously.**

Improve the quality of the assembly: Polishing



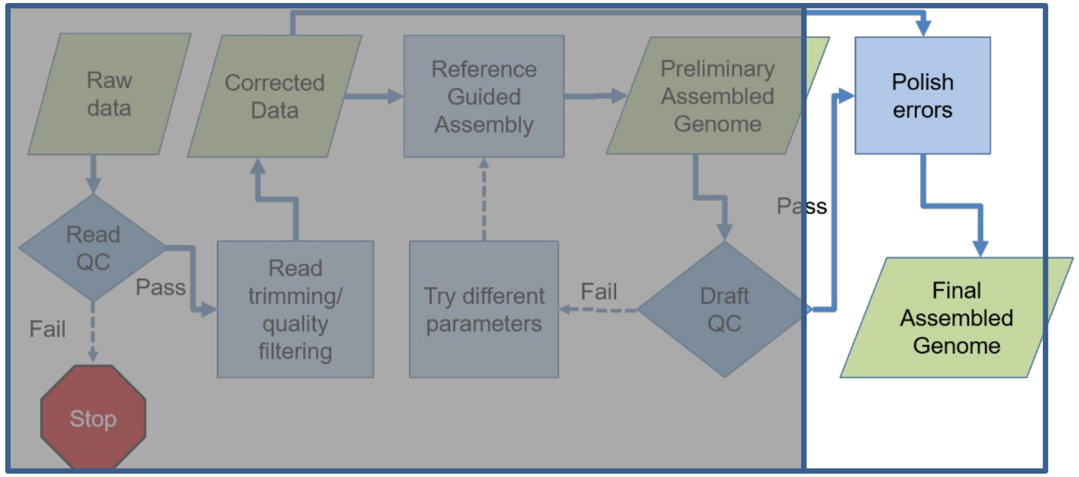
- **The read sequences may disagree with the assembly sequence at certain positions. The divergent sequences are known as “variants.”**
- **Identifying these differences will enable us to correct small-scale errors, yielding a more accurate final assembly.**

Improve the quality of the assembly: Polishing

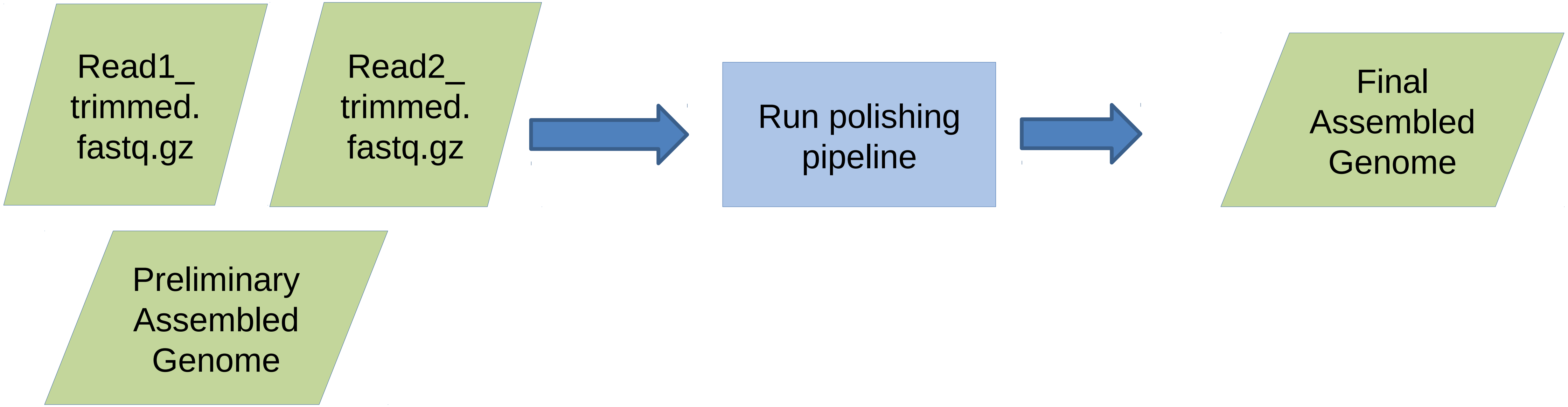


- **A final assembly is generated with the consensus sequences, which are generally the most common sequence at the position.**

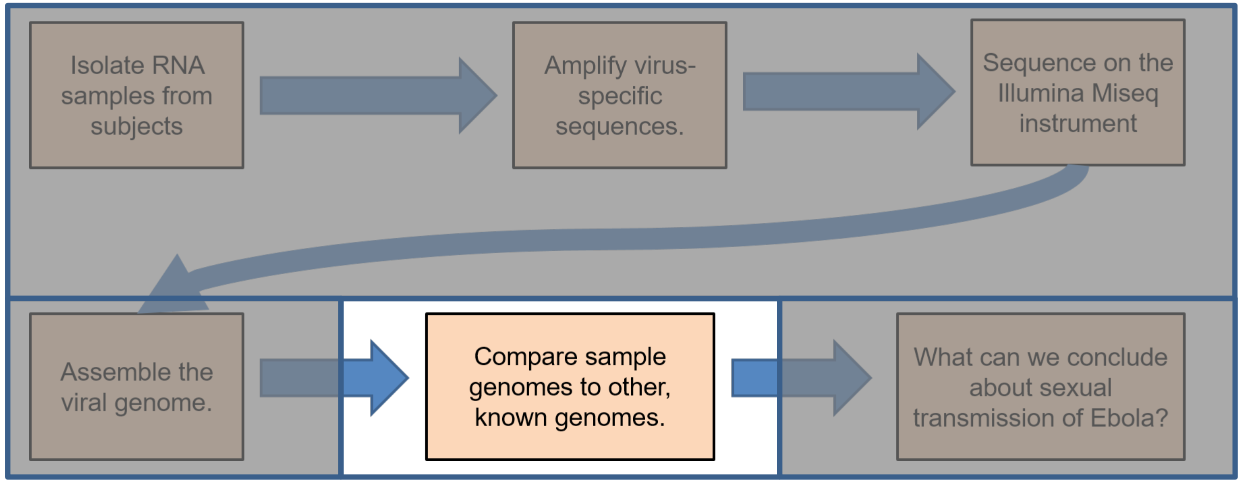
Polishing the Genome



- **“Polish” out errors in the assembly by mapping the reads back to the assembly.**
- **Identify positions where the read sequences differ from the draft genome.**
- **Correct the draft sequence at those positions, producing a higher quality final assembled genome.**



Multiple Genome Alignment



Isolate RNA samples from subjects

Amplify virus-specific sequences.

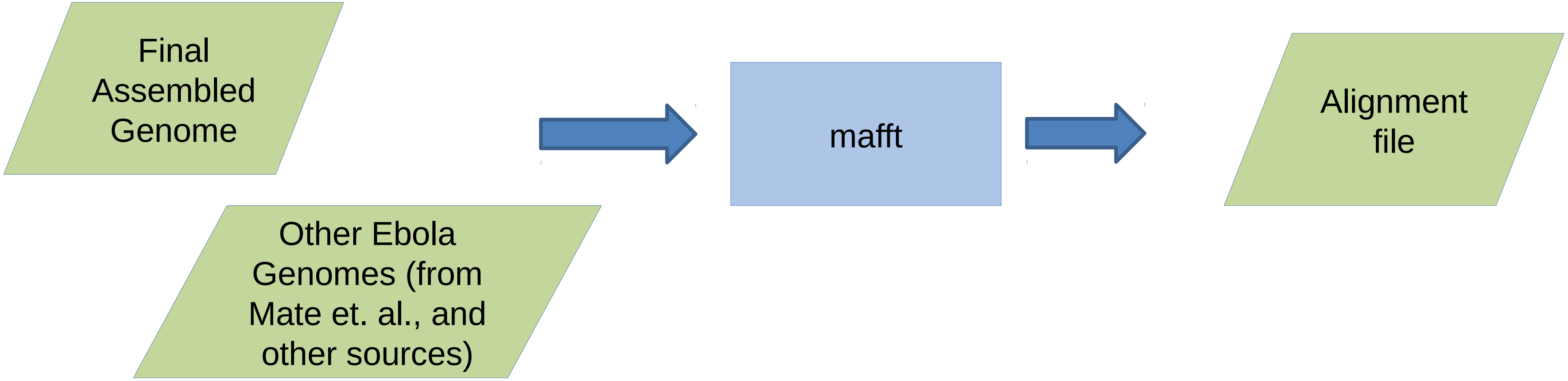
Sequence on the Illumina Miseq instrument

Assemble the viral genome.

Compare sample genomes to other, known genomes.

What can we conclude about sexual transmission of Ebola?

- **Take the final assembled genome, along with a diversity of other Ebola genomes.**
- **Align the genomes to each other, allowing us to quantify how different the genomes from each patient are from each other, and from other Ebola sequences.**



Final Assembled Genome

Other Ebola Genomes (from Mate et. al., and other sources)

mafft

Alignment file

Summary of the alignment.

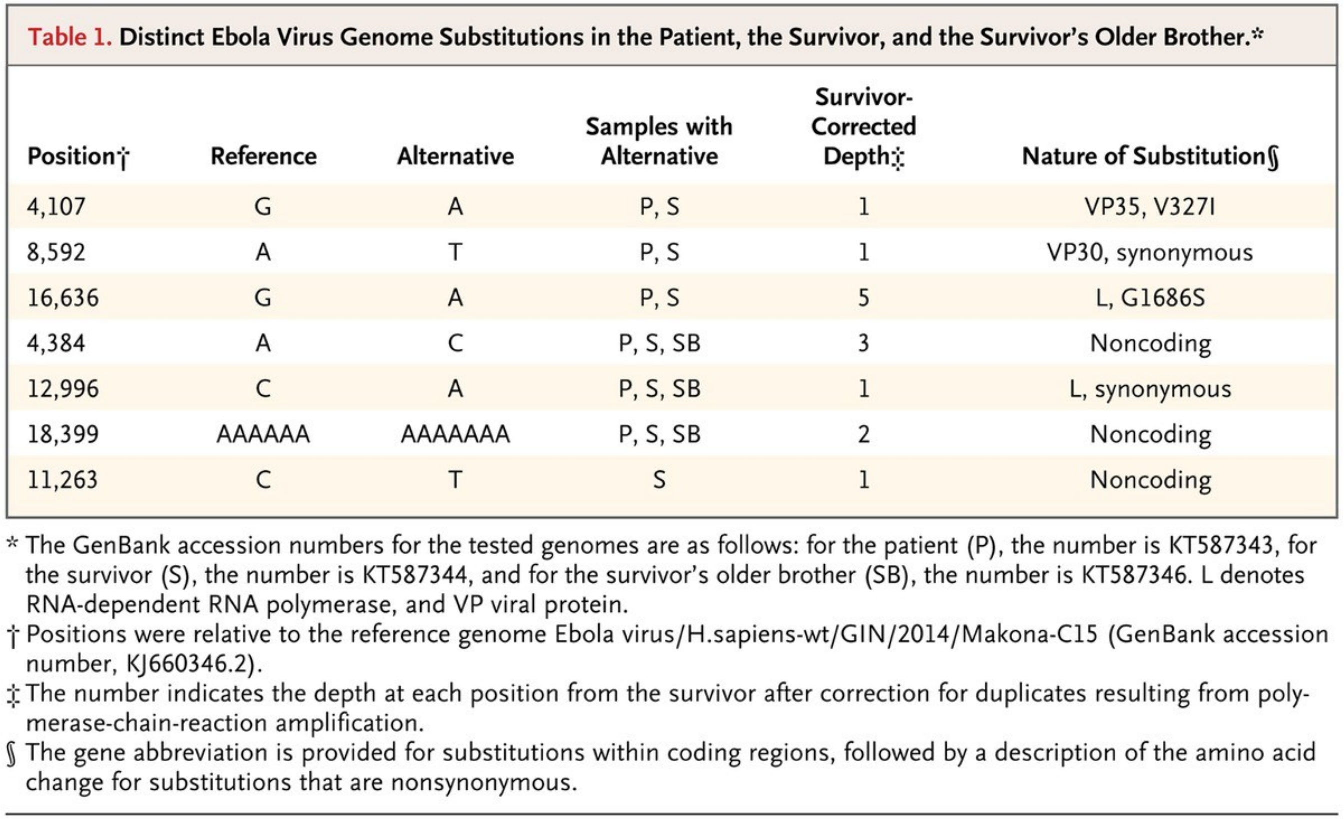


Table 1. Distinct Ebola Virus Genome Substitutions in the Patient, the Survivor, and the Survivor's Older Brother.*

Position†	Reference	Alternative	Samples with Alternative	Survivor-Corrected Depth‡	Nature of Substitution§
4,107	G	A	P, S	1	VP35, V327I
8,592	A	T	P, S	1	VP30, synonymous
16,636	G	A	P, S	5	L, G1686S
4,384	A	C	P, S, SB	3	Noncoding
12,996	C	A	P, S, SB	1	L, synonymous
18,399	AAAAAA	AAAAAAA	P, S, SB	2	Noncoding
11,263	C	T	S	1	Noncoding

* The GenBank accession numbers for the tested genomes are as follows: for the patient (P), the number is KT587343, for the survivor (S), the number is KT587344, and for the survivor's older brother (SB), the number is KT587346. L denotes RNA-dependent RNA polymerase, and VP viral protein.

† Positions were relative to the reference genome Ebola virus/H.sapiens-wt/GIN/2014/Makona-C15 (GenBank accession number, KJ660346.2).

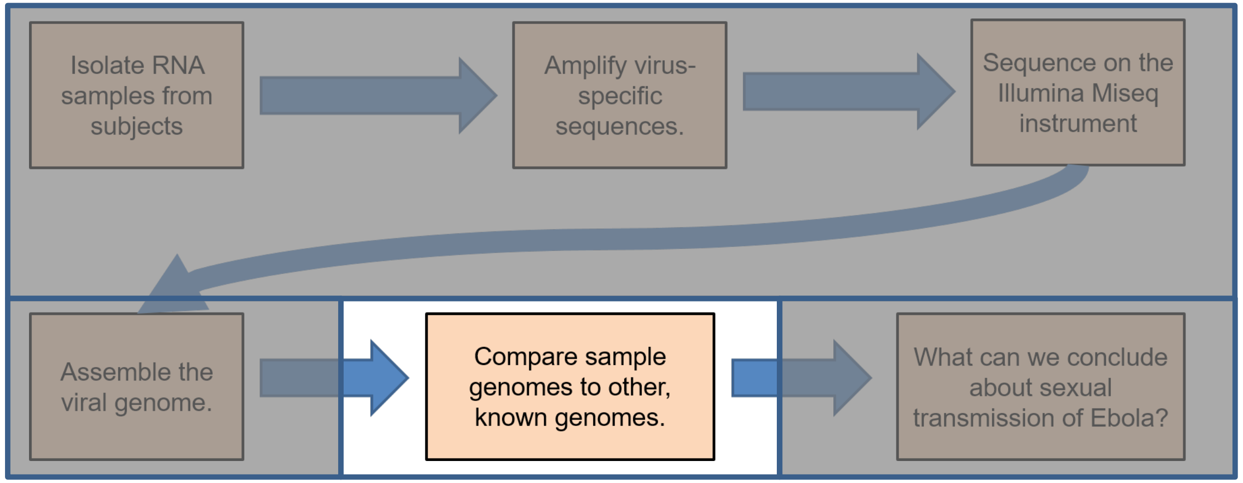
‡ The number indicates the depth at each position from the survivor after correction for duplicates resulting from polymerase-chain-reaction amplification.

§ The gene abbreviation is provided for substitutions within coding regions, followed by a description of the amino acid change for substitutions that are nonsynonymous.

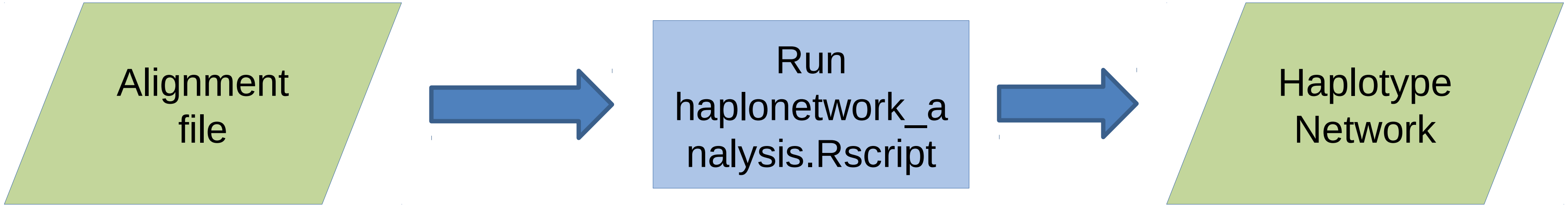
- **List the differences between the Mate et al. samples and a reference genome in a chart.**
- **There are three positions where the Survivor and Survivor's Partner differ from the reference, but not from each other (lines 1-3).**
- **There are three positions where the Survivor, Survivor's partner and Survivor's brother differ from the reference (lines 4-6).**
- **There is one position where only the Survivor differs from the reference (line 7).**

Show Alignment File

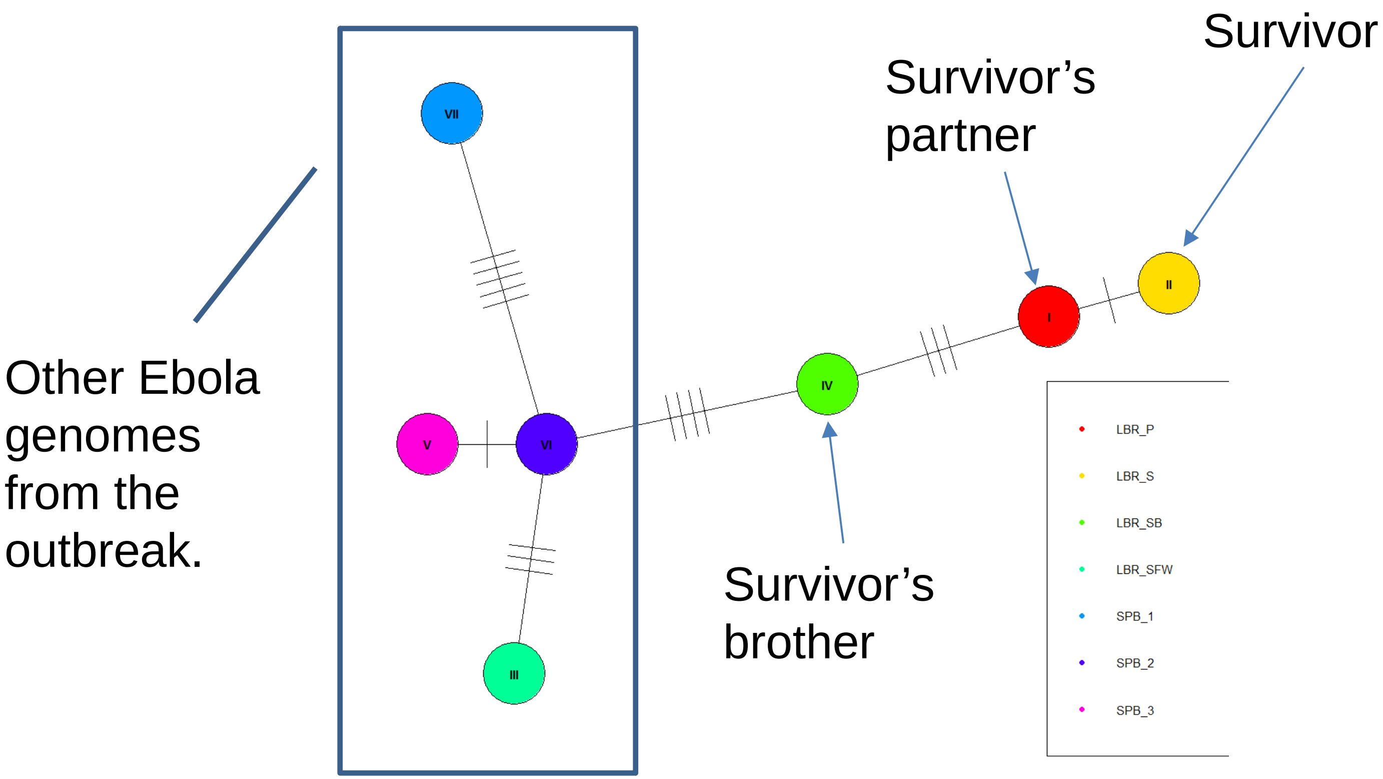
Haplotype Network Analysis


Isolate RNA samples from subjects
Amplify virus-specific sequences.
Sequence on the Illumina Miseq instrument
Assemble the viral genome.
Compare sample genomes to other, known genomes.
What can we conclude about sexual transmission of Ebola?

- **Start with the alignment file made in the previous step.**
- **Arrange the genomes in a haplotype network: where each genome is connected by a line to the genomes it is most similar to.**
- **This allows us to visually depict the differences between several genomes.**



Compare the genomes in the study and other Ebola genomes using a haplotype network



Other Ebola genomes from the outbreak.

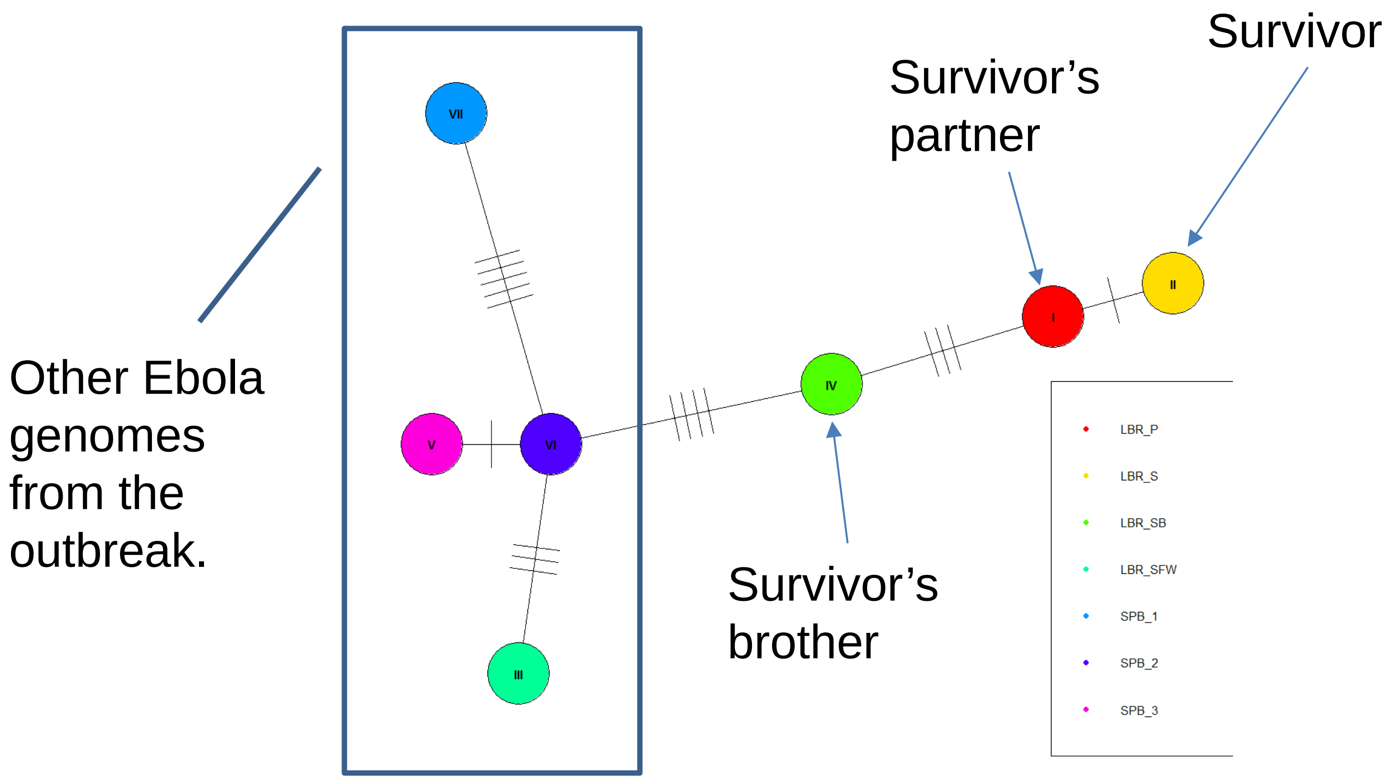
Survivor's partner

Survivor

Survivor's brother

- A haplotype is a region of DNA inherited from the parent.
- NOTE: for viruses, the haplotype is the full genome.
- Each circle (node) represents a genome sequence.
- Hash marks on the lines show the number of differences between the genomes connected by the lines.
- <u>Remember our original question</u>: Is the partner sample more similar to the survivor sequence? Or to the other samples from this outbreak?

Compare the genomes in the study and other Ebola genomes using a haplotype network



Other Ebola genomes from the outbreak.

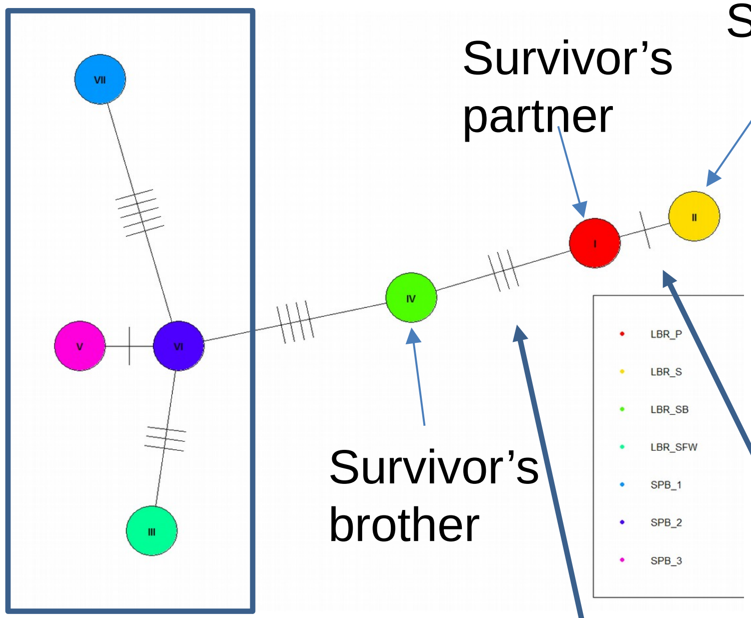
Survivor's brother

Survivor's partner

Survivor

- The Survivor and Survivor's Partner have one difference between them.
- There are at least three differences between the Survivor's Partner and the next most similar genome.

What can we conclude about sexual transmission of Ebola?



Other Ebola genomes from the outbreak.

Survivor's partner

Survivor

Survivor's brother

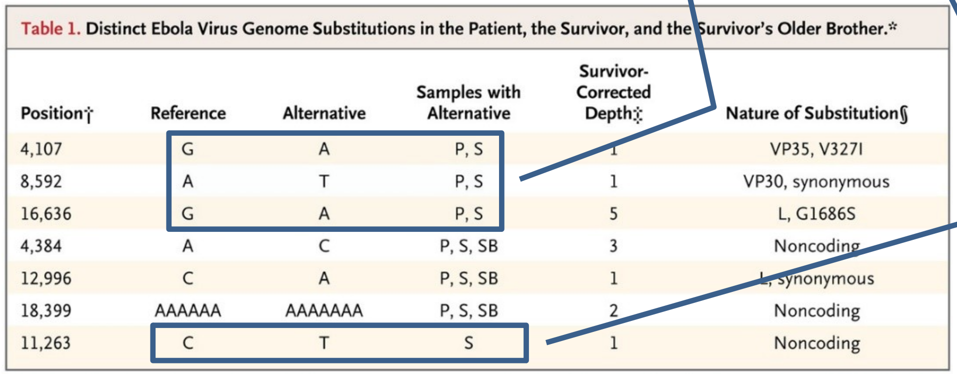


Table 1. Distinct Ebola Virus Genome Substitutions in the Patient, the Survivor, and the Survivor's Older Brother.*

Position†	Reference	Alternative	Samples with Alternative	Survivor-Corrected Depth‡	Nature of Substitution§
4,107	G	A	P, S	1	VP35, V327I
8,592	A	T	P, S	1	VP30, synonymous
16,636	G	A	P, S	5	L, G1686S
4,384	A	C	P, S, SB	3	Noncoding
12,996	C	A	P, S, SB	1	L, synonymous
18,399	AAAAAA	AAAAAAA	P, S, SB	2	Noncoding
11,263	C	T	S	1	Noncoding

- We see a similar pattern from the chart we previously produced.
- There is one position unique to the Survivor.
- There are three positions shared by the Survivor and Survivor's partner, but different in the Survivor's brother.

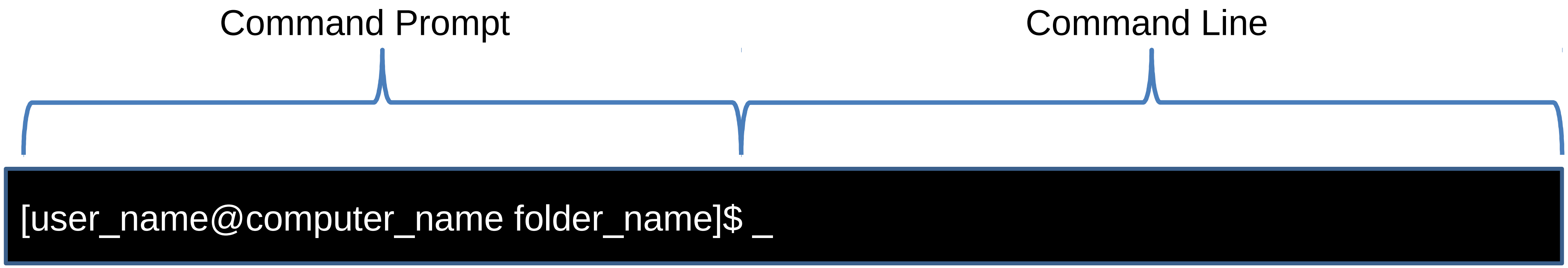
- **This data is consistent with sexual transmission from the Survivor to the Survivor's Partner.**

Assembly through the command line.

- The following details how to run the analysis through the command line.
- The underlying analysis is the same, but we go into greater detail, breaking down each command that ran behind the scenes in the previous tutorial.

The Shell

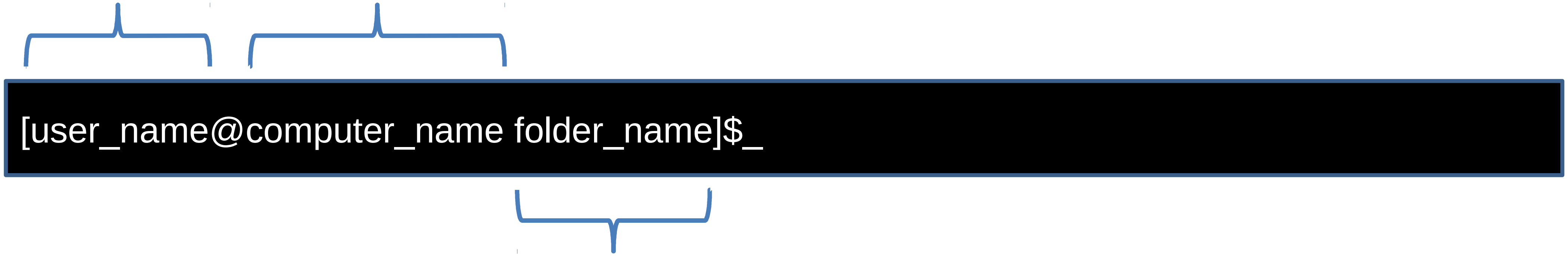
- A Shell is a program that provides a text only user interface for interacting with the computer.
- The shell consists of a command prompt, showing the user name and location, and the command line, where commands are entered.



```
[user_name@computer_name folder_name]$ _
```

The Command Prompt

- The command prompt shows basic information.



Name of
logged in user.

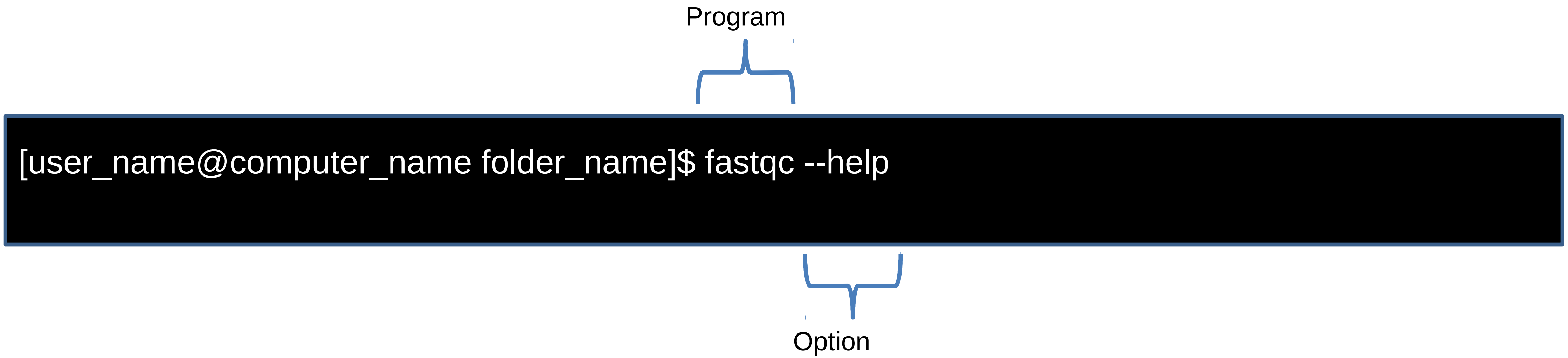
Name of computer or server
where the user is logged in.

```
[user_name@computer_name folder_name]$_
```

Current location in the
computer’s file structure.

The Command Line

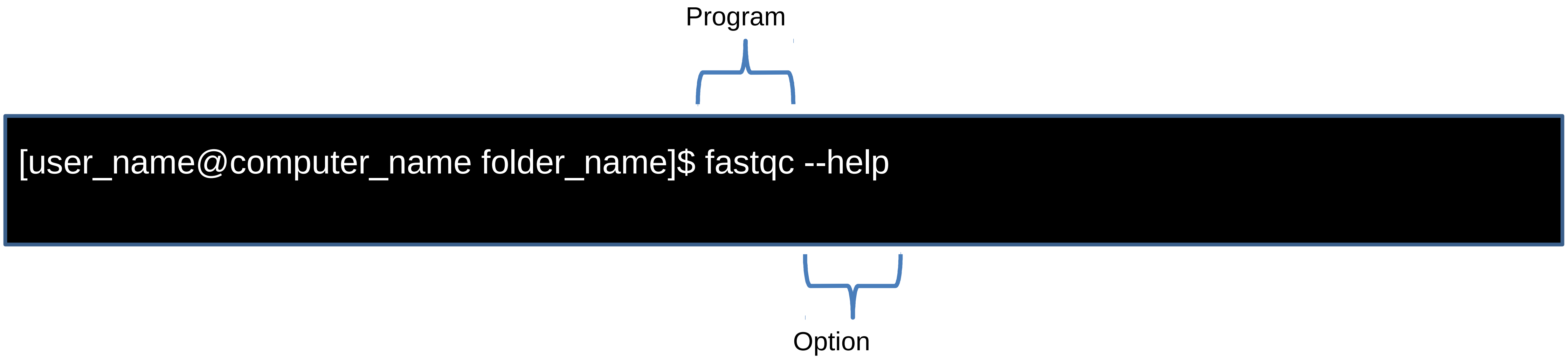
- Run programs and navigate files by typing commands into the command line, next to the command prompt.



```
[user_name@computer_name folder_name]$ fastqc --help
```

--help

- Note that for most programs that run on the command line, the manual can be accessed by typing the name of the program, followed by a space and “-h” or “--help” as in the example below.



Program

```
[user_name@computer_name folder_name]$ fastqc --help
```

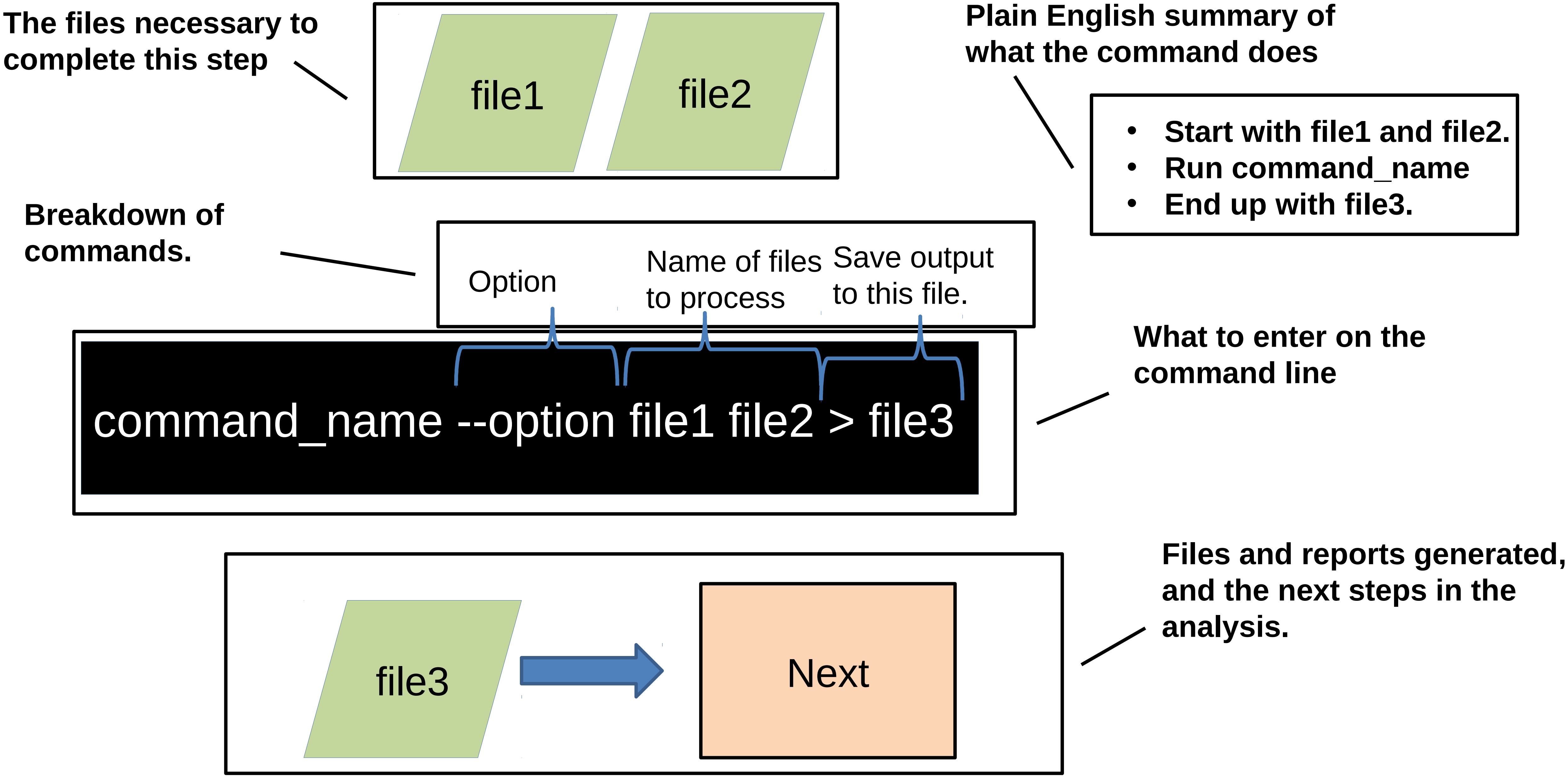
Option

Command line caveats

- Commands are case sensitive. Enter commands exactly as written!
- Spacing and ordering of arguments are important, and can change the output of the command, so enter the commands exactly as written!
- Pressing enter runs the command as it appears in the command line. There is no warning or confirmation!
- Shells are text only interfaces. You cannot click on a space in the

Benefits of using the command line.

- Finer control of program parameters.
- Can string together multiple programs into analysis pipelines.
- Record of exactly what commands and parameters have been run.
- Increased portability and reproducibility.



The files necessary to complete this step

Plain English summary of what the command does

- Start with file1 and file2.
- Run command_name
- End up with file3.

Breakdown of commands.

Option

Name of files to process

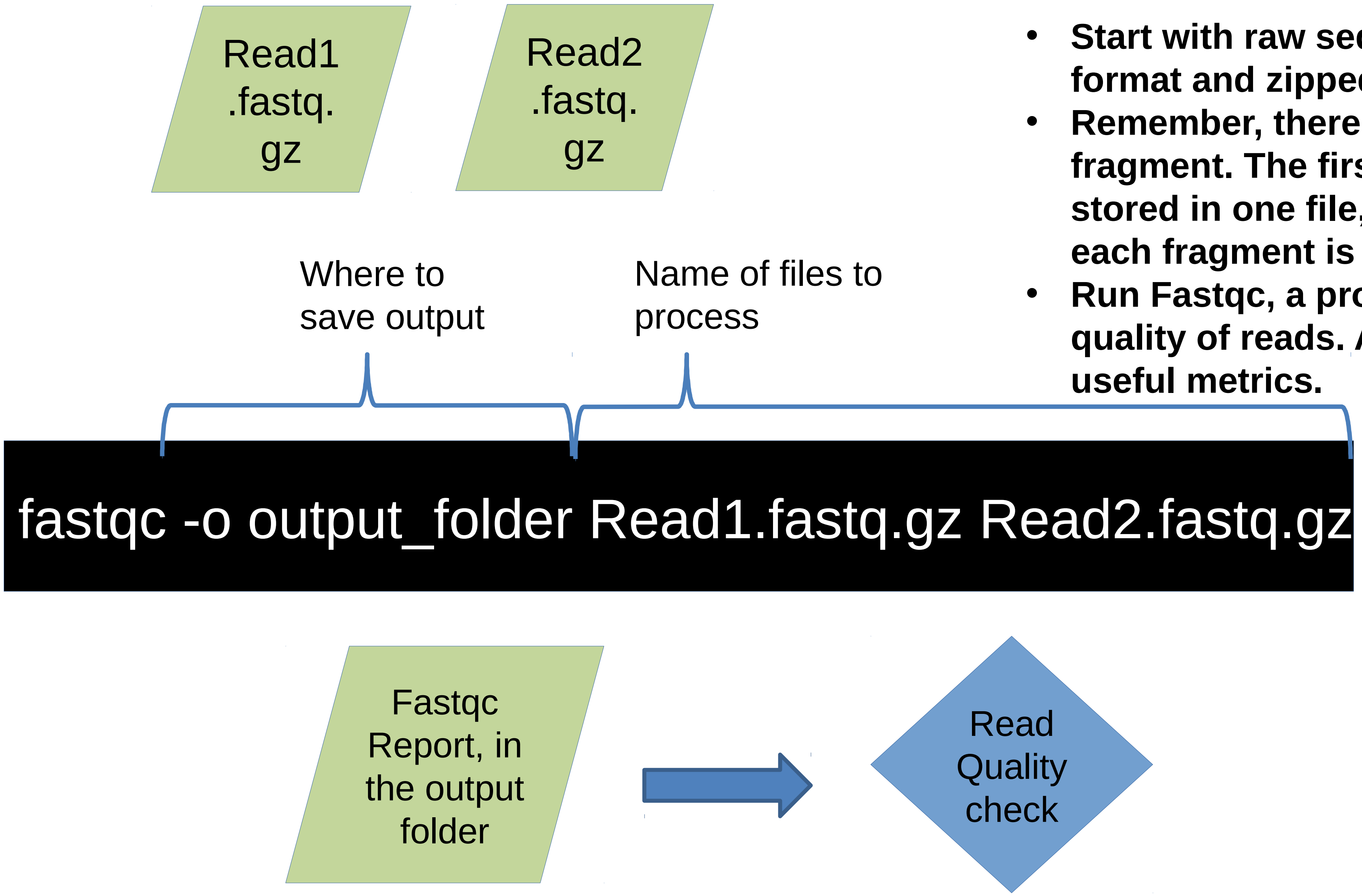
Save output to this file.

```
command_name --option file1 file2 > file3
```

What to enter on the command line

Files and reports generated, and the next steps in the analysis.

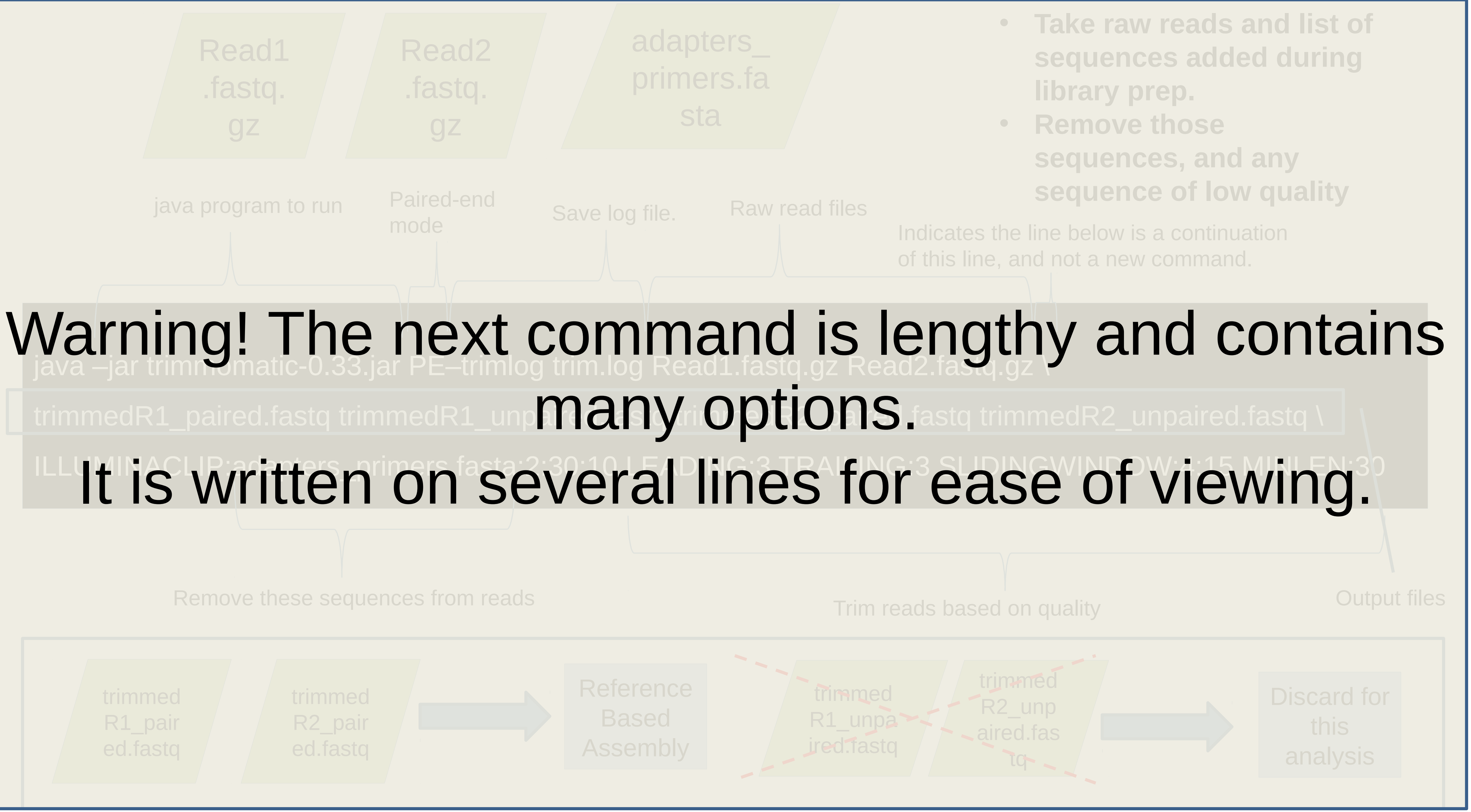
- **Start with raw sequencing data, in fastq format and zipped.**
- **Remember, there are two reads for each DNA fragment. The first read of each fragment is stored in one file, and the second read of each fragment is stored in another.**
- **Run Fastqc, a program that summarizes the quality of reads. Also outputs a number of useful metrics.**



```
fastqc -o output_folder Read1.fastq.gz Read2.fastq.gz
```

Warning! The next command is lengthy and contains many options.

It is written on several lines for ease of viewing.



Read1.fastq.gz

Read2.fastq.gz

adapters_primers.fasta

- **Take raw reads and list of sequences added during library prep.**
- **Remove those sequences, and any sequence of low quality**

java program to run

Paired-end mode

Save log file.

Raw read files

Indicates the line below is a continuation of this line, and not a new command.

```
java –jar trimmomatic-0.33.jar PE–trimlog trim.log Read1.fastq.gz Read2.fastq.gz \
trimmedR1_paired.fastq trimmedR1_unpaired.fastq trimmedR2_paired.fastq trimmedR2_unpaired.fastq \
ILLUMINACLIP:adapters_primers.fasta:2:30:10 LEADING:3 TRAILING:3 SLIDINGWINDOW:4:15 MINLEN:30
```

Remove these sequences from reads

Trim reads based on quality

Output files

trimmedR1_paired.fastq

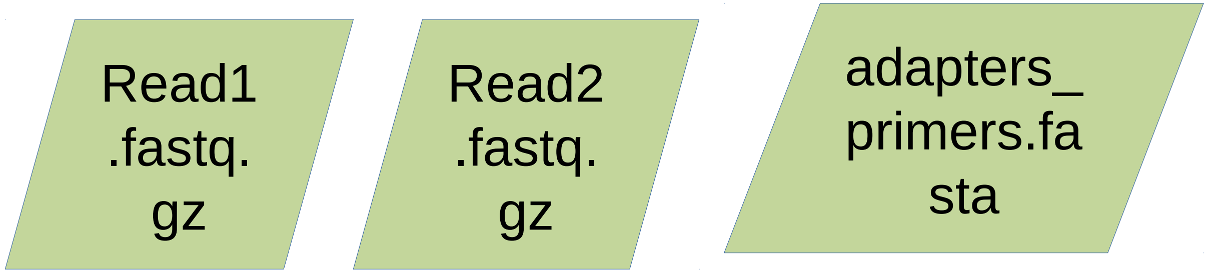
trimmedR2_paired.fastq

Reference Based Assembly

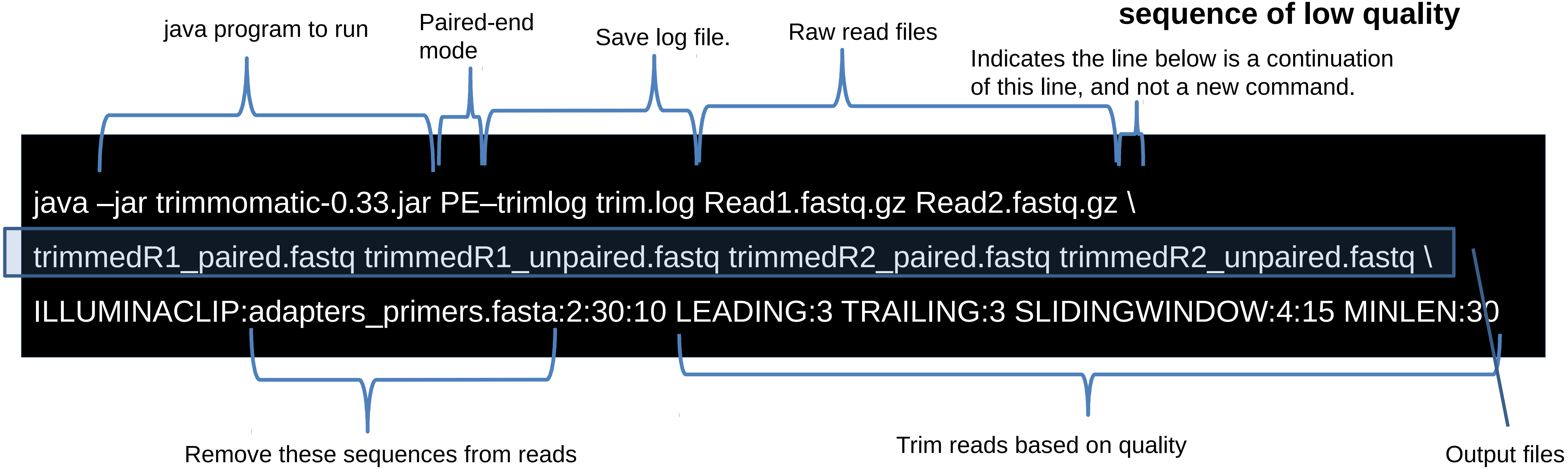
trimmedR1_unpaired.fastq

trimmedR2_unpaired.fastq

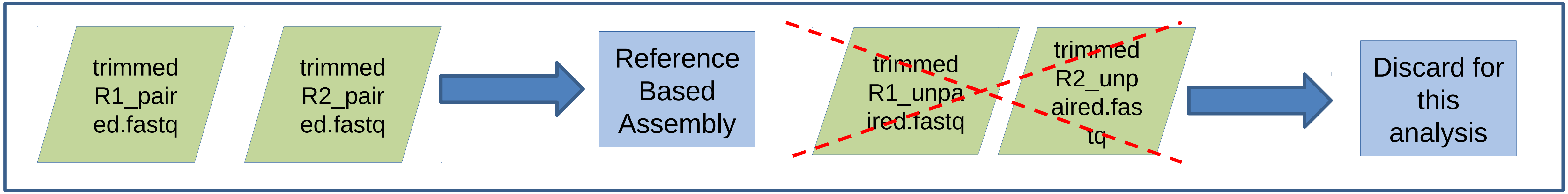
Discard for this analysis



- **Take raw reads and list of sequences added during library prep.**
- **Remove those sequences, and any sequence of low quality**


java program to run
Paired-end mode
Save log file.
Raw read files
Indicates the line below is a continuation of this line, and not a new command.
Remove these sequences from reads
Trim reads based on quality
Output files

```
java –jar trimmomatic-0.33.jar PE–trimlog trim.log Read1.fastq.gz Read2.fastq.gz \
trimmedR1_paired.fastq trimmedR1_unpaired.fastq trimmedR2_paired.fastq trimmedR2_unpaired.fastq \
ILLUMINACLIP:adapters_primers.fasta:2:30:10 LEADING:3 TRAILING:3 SLIDINGWINDOW:4:15 MINLEN:30
```



- Take raw reads and list of sequences added during library prep.
- Remove those sequences, and any sequence of low quality

java program to run

Paired-end mode

Save log file.

Raw read files

Indicates the line below is a continuation of this line, and not a new command

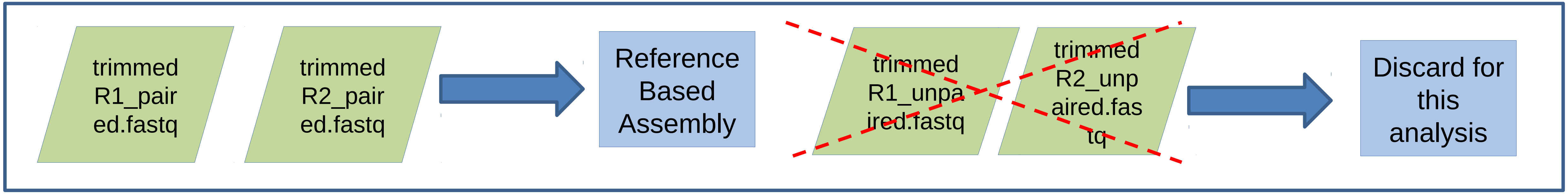
Note that this command requires outputting 4 files, but we will only use two in the subsequent steps.

```
java -jar trimmomatic-0.33.jar PE -trimlog trim.log Read1.fastq.gz Read2.fastq.gz \
trimmedR1_paired.fastq trimmedR1_unpaired.fastq trimmedR2_paired.fastq trimmedR2_unpaired.fastq \
ILLUMINACLIP:adapters_primers.fasta:2:30:10 LEADING:3 TRAILING:3 SLIDINGWINDOW:4:15 MINLEN:30
```

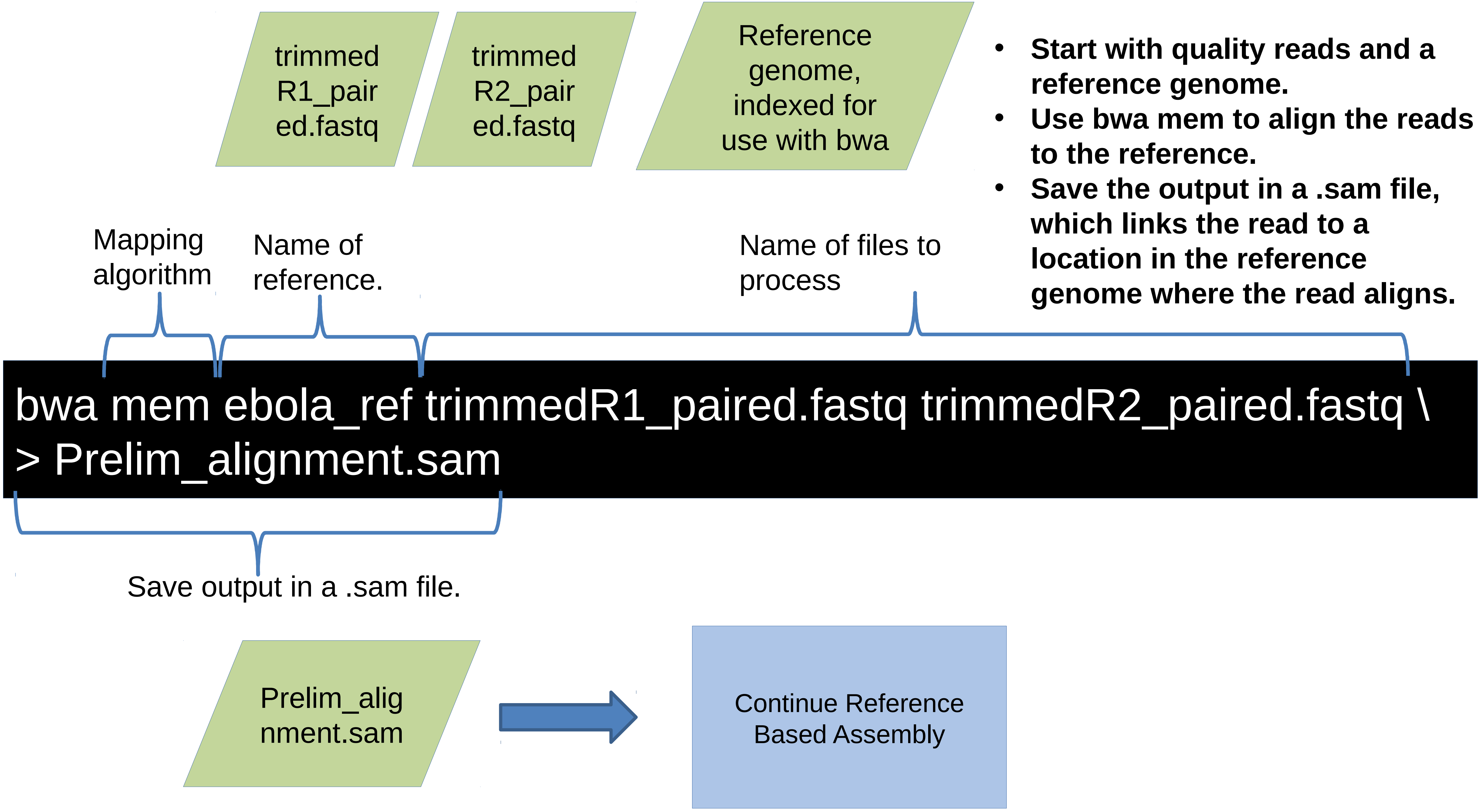
Remove these sequences from reads

Trim reads based on quality

Output files



- **Start with quality reads and a reference genome.**
- **Use bwa mem to align the reads to the reference.**
- **Save the output in a .sam file, which links the read to a location in the reference genome where the read aligns.**



trimmed R1_paired.fastq

trimmed R2_paired.fastq

Reference genome, indexed for use with bwa

Mapping algorithm

Name of reference.

Name of files to process

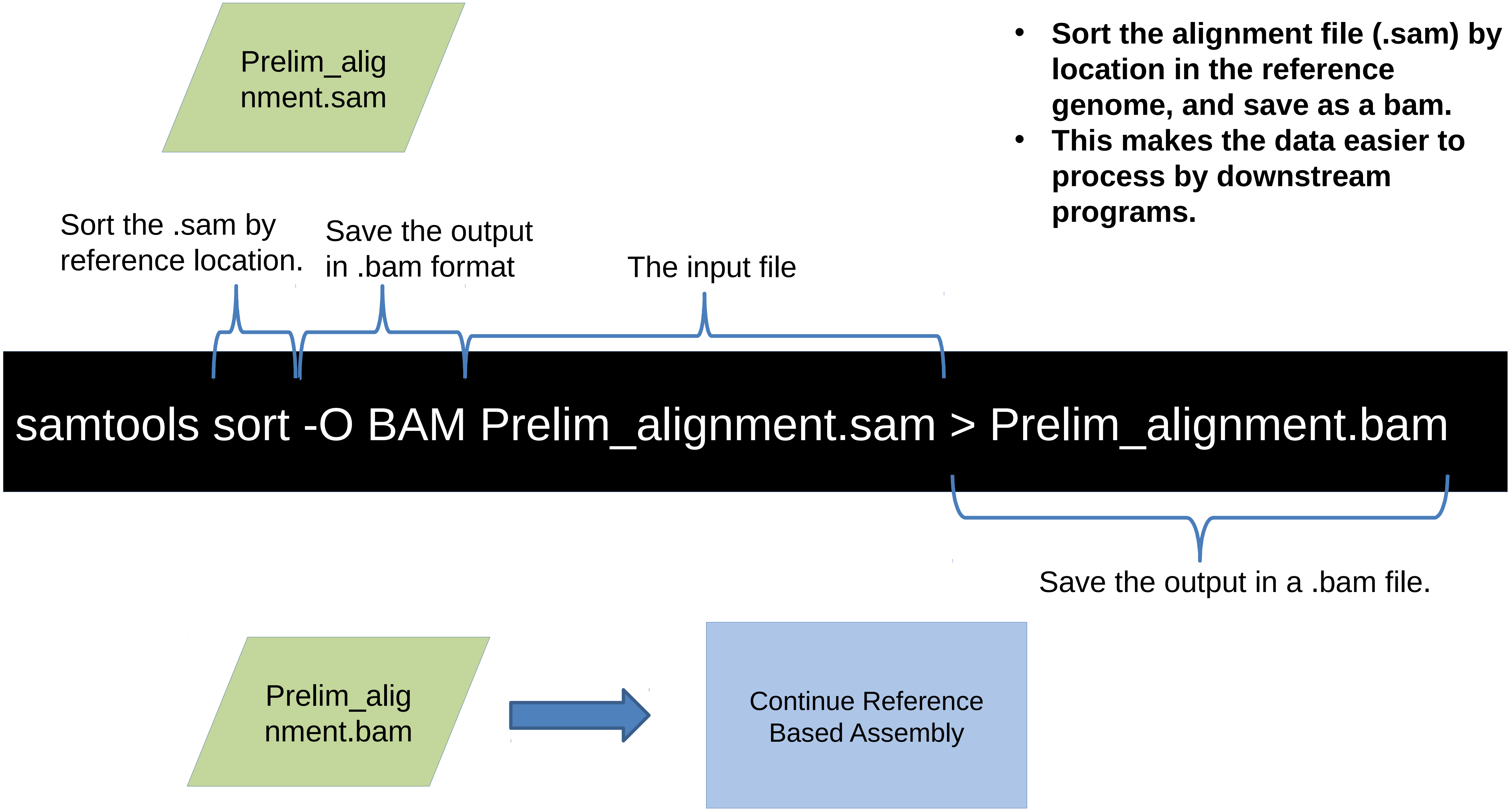
```
bwa mem ebola_ref trimmedR1_paired.fastq trimmedR2_paired.fastq \
> Prelim_alignment.sam
```

Save output in a .sam file.

Prelim_alignment.sam

Continue Reference Based Assembly

- **Sort the alignment file (.sam) by location in the reference genome, and save as a bam.**
- **This makes the data easier to process by downstream programs.**



Prelim_alignment.sam

Sort the .sam by reference location.

Save the output in .bam format

The input file

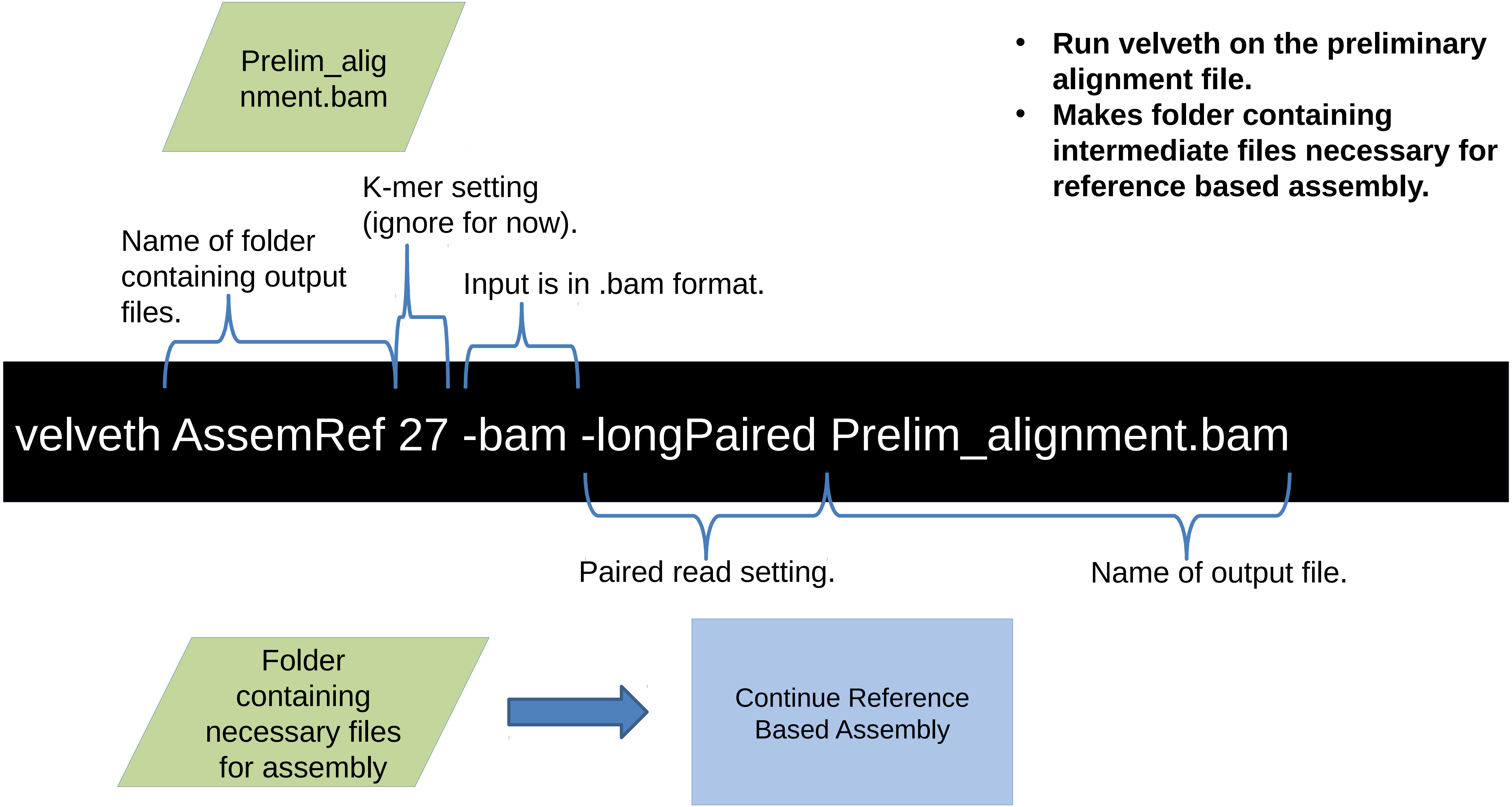
```
samtools sort -O BAM Prelim_alignment.sam > Prelim_alignment.bam
```

Save the output in a .bam file.

Prelim_alignment.bam

Continue Reference Based Assembly

- **Run velveth on the preliminary alignment file.**
- **Makes folder containing intermediate files necessary for reference based assembly.**



```
velveth AssemRef 27 -bam -longPaired Prelim_alignment.bam
```

Prelim_alignment.bam

Name of folder containing output files.

K-mer setting (ignore for now).

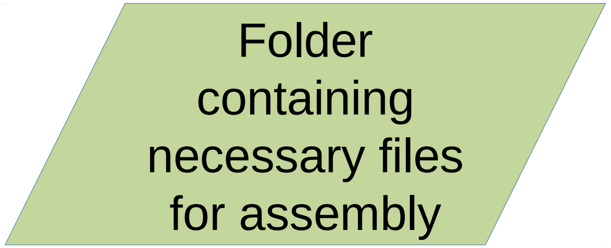
Input is in .bam format.

Paired read setting.

Name of output file.

Folder containing necessary files for assembly

Continue Reference Based Assembly



- **Run the assembly program, velvetg.**
- **Reads the alignment file, in the folder containing intermediate files, and generates the preliminary assembly.**

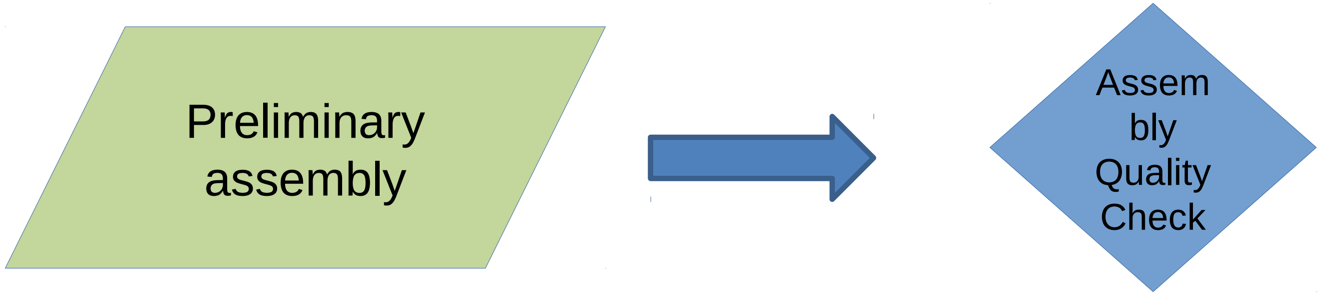
Name of folder containing output files.

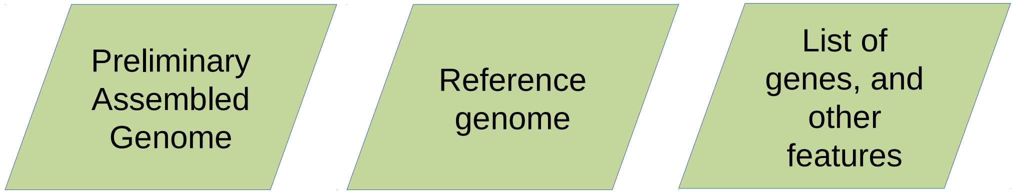
Make a file for QC purposes

Save a log file.

```
velvetg AssemRef -amos_file yes > logfile_assemref_27.txt 2>&1 &
```

Save any error messages, and run this command in the background.




Preliminary Assembled Genome
Reference genome
List of genes, and other features

- **Quast compares the preliminary assembly to a known genome from the same species.**
- **Also, identifies functional sequences, like genes and RNAs.**

Preliminary assembly

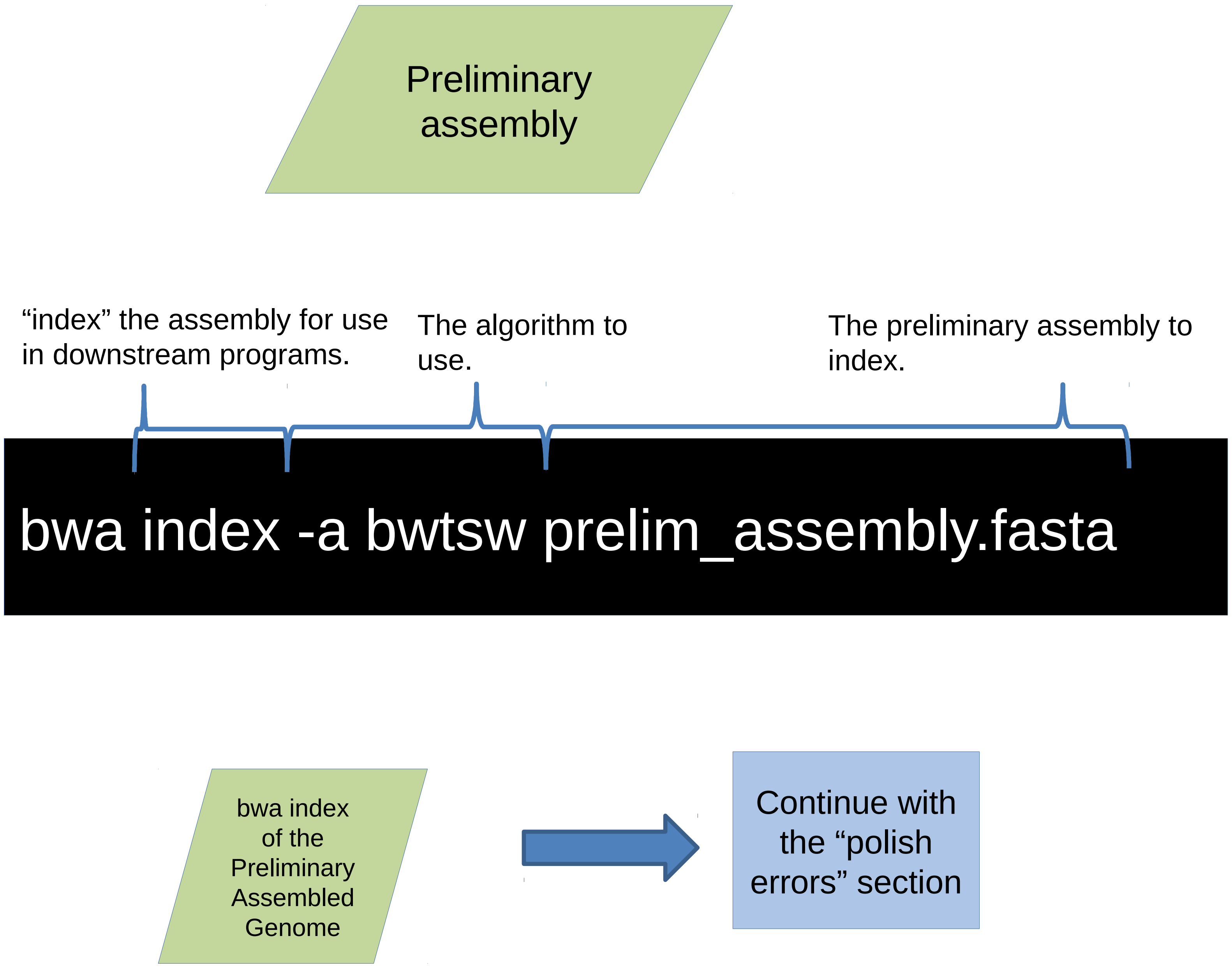
The reference .fasta file.

```
quast.py prelim_assembly.fasta –R ebola_ref.fasta \
–G ebola_ref_genes.gff –o output_folder –glimmer
```

List of genes in the reference.

Output folder
containing results

Provide list of gene locations in
the preliminary assembly.



Preliminary assembly

“index” the assembly for use in downstream programs.

The algorithm to use.

The preliminary assembly to index.

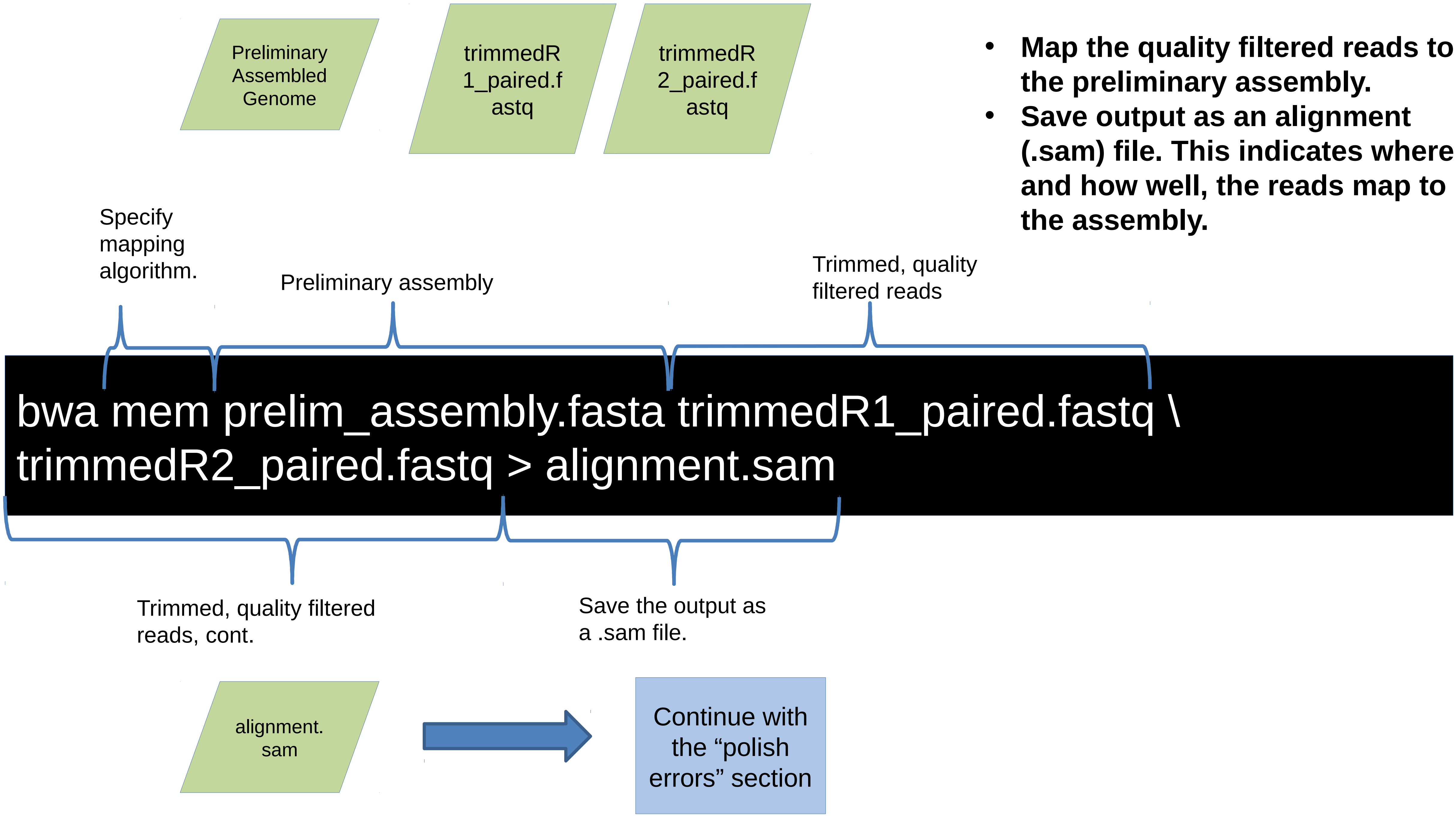
```
bwa index -a bwtsw prelim_assembly.fasta
```

bwa index of the Preliminary Assembled Genome

Continue with the “polish errors” section

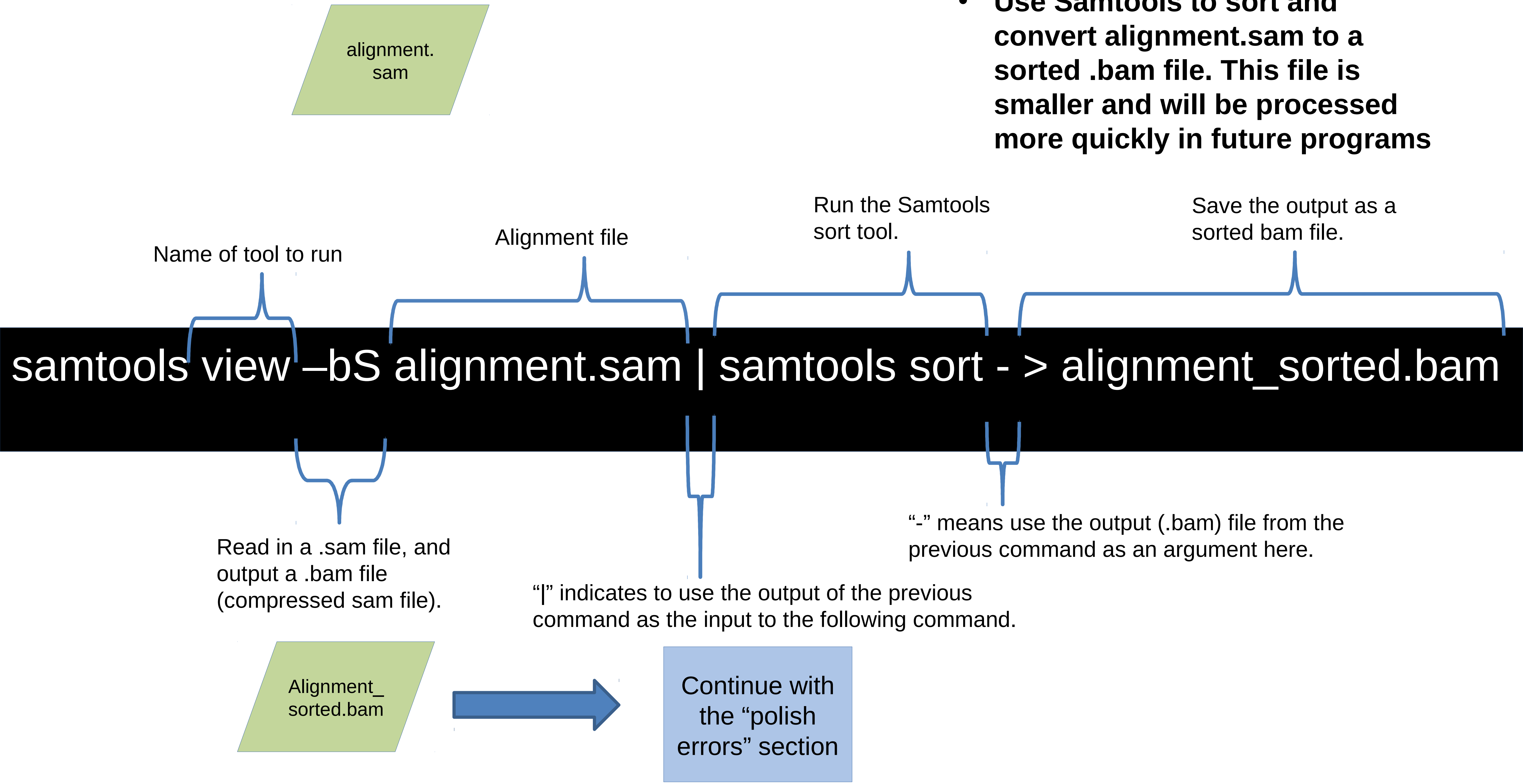
- **“Polish” out errors in the assembly by mapping the reads back to the assembly.**
- **This will take several steps.**
- **Map the quality filtered reads to the preliminary assembly.**
- **The index can then be used by the mapping program, bwa mem, in the next step.**

- **Map the quality filtered reads to the preliminary assembly.**
- **Save output as an alignment (.sam) file. This indicates where, and how well, the reads map to the assembly.**


Preliminary Assembled Genome
trimmedR1_paired.fastq
trimmedR2_paired.fastq
Specify mapping algorithm.
Preliminary assembly
Trimmed, quality filtered reads
Trimmed, quality filtered reads, cont.
Save the output as a .sam file.
alignment.sam
Continue with the “polish errors” section

```
bwa mem prelim_assembly.fasta trimmedR1_paired.fastq \
trimmedR2_paired.fastq > alignment.sam
```

- **Use Samtools to sort and convert alignment.sam to a sorted .bam file. This file is smaller and will be processed more quickly in future programs**



alignment.sam

Name of tool to run

Alignment file

Run the Samtools sort tool.

Save the output as a sorted bam file.

```
samtools view –bS alignment.sam | samtools sort - > alignment_sorted.bam
```

Read in a .sam file, and output a .bam file (compressed sam file).

“|” indicates to use the output of the previous command as the input to the following command.

“-” means use the output (.bam) file from the previous command as an argument here.

Alignment_sorted.bam

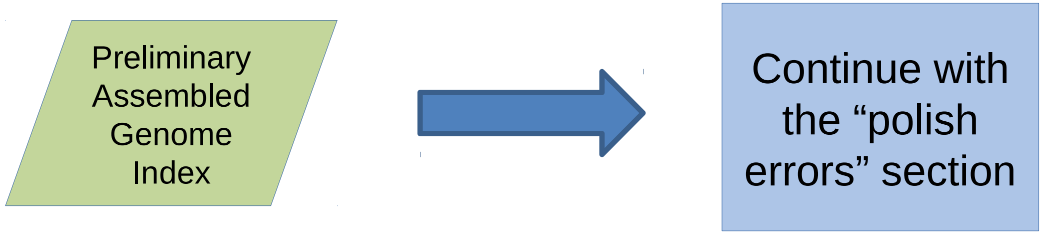
Continue with the “polish errors” section

- **Make an index of the preliminary assembled genome.**
- **This is necessary to use the assembly in the next step.**

Make a samtools compatible index.

Preliminary assembled genome

```
samtools faidx prelim_assembly.fasta
```

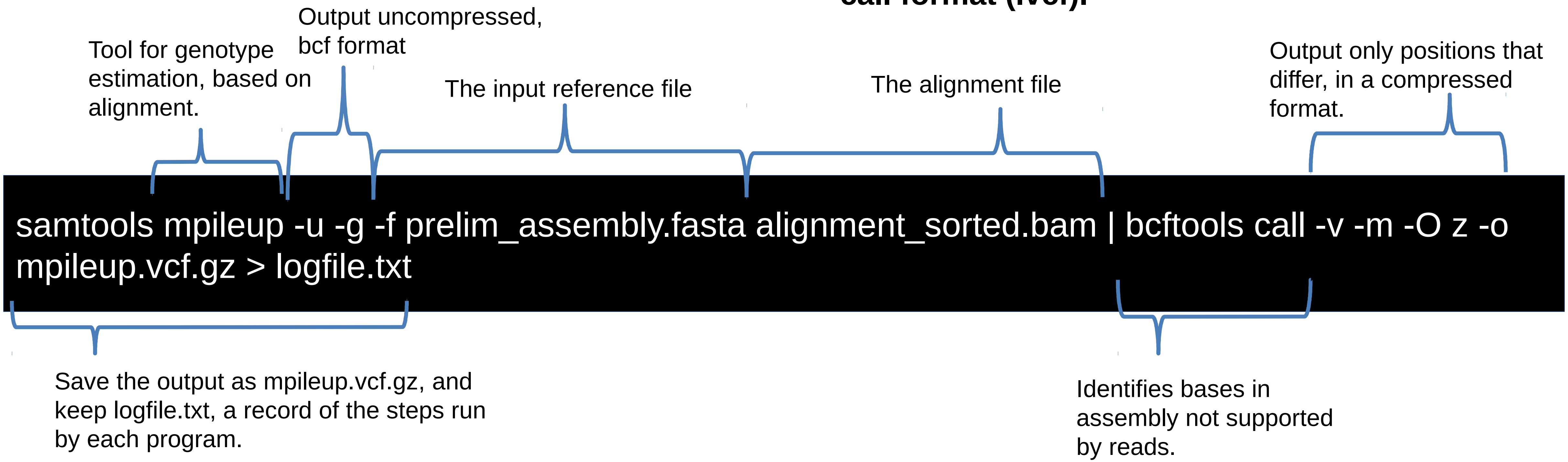


Preliminary Assembly, indexed

trimmedR1_paired.fastq

trimmedR2_paired.fastq

- **Samtools mpileup estimates the probability that each base in the preliminary assembly is correct, given the reads that map to it.**
- **bcftools call determines if the reads provide support for a different base at any given position in the assembly, based on these probabilities.**
- **The output is a list of positions in the preliminary assembly that should be changed, in the “variant call format (.vcf).”**



Tool for genotype estimation, based on alignment.

Output uncompressed, bcf format

The input reference file

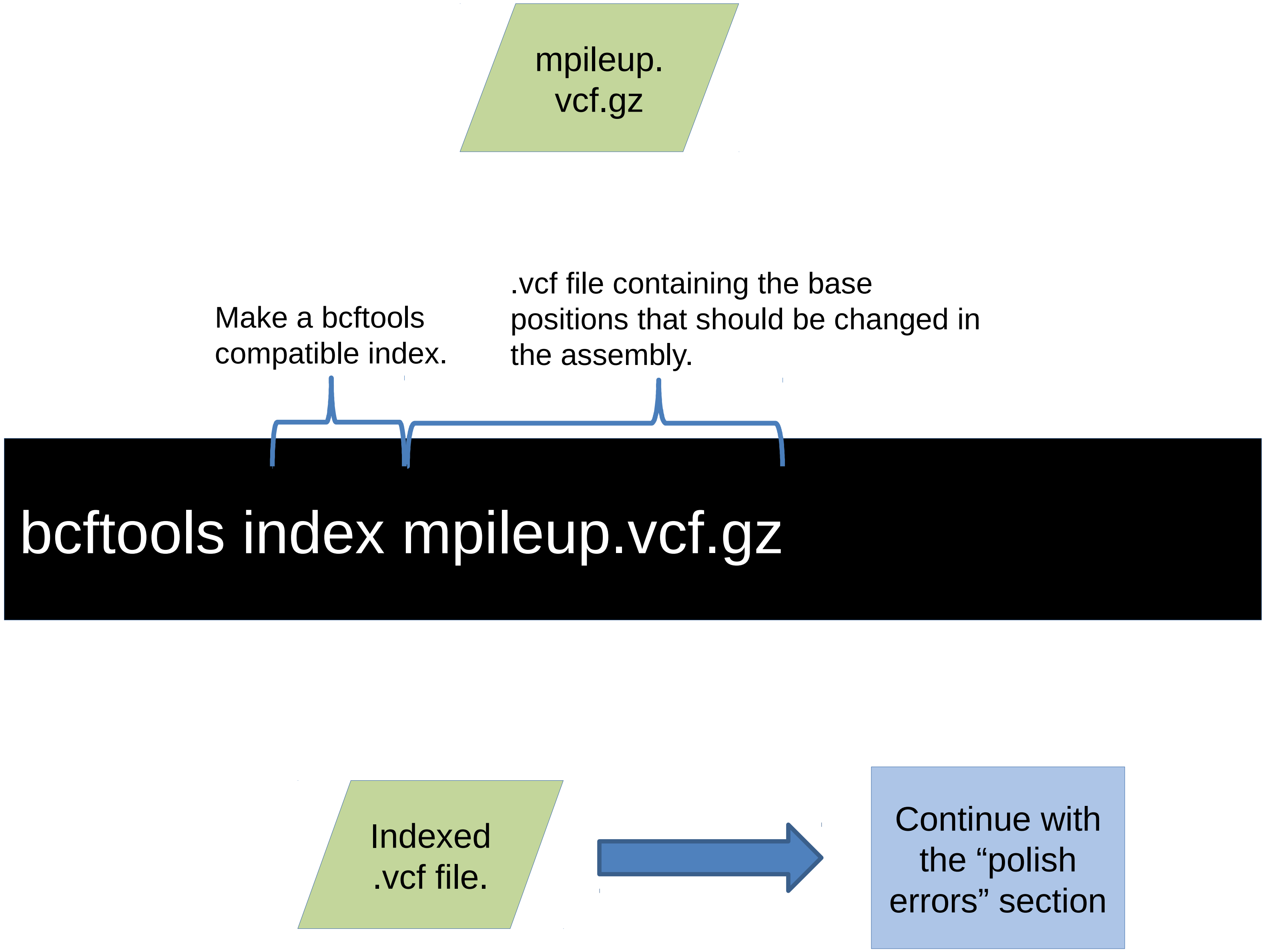
The alignment file

Output only positions that differ, in a compressed format.

```
samtools mpileup -u -g -f prelim_assembly.fasta alignment_sorted.bam | bcftools call -v -m -O z -o
mpileup.vcf.gz > logfile.txt
```

Save the output as mpileup.vcf.gz, and keep logfile.txt, a record of the steps run by each program.

Identifies bases in assembly not supported by reads.



mpileup.
vcf.gz

Make a bcftools
compatible index.

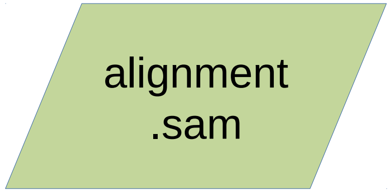
.vcf file containing the base
positions that should be changed in
the assembly.

```
bcftools index mpileup.vcf.gz
```

Indexed
.vcf file.

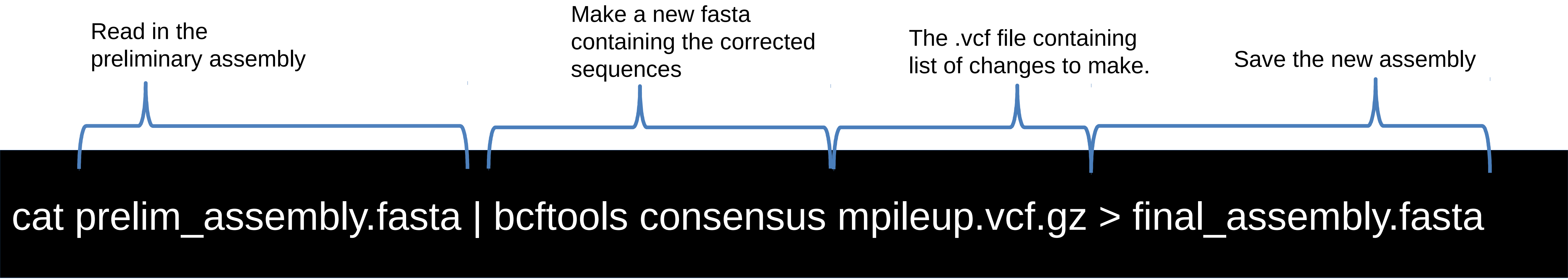
Continue with
the “polish
errors” section

- **Make an index of the .vcf file, which will make processing the file easier in the next step.**



alignment .sam

- **Read in the preliminary assembly.**
- **Use “bcftools consensus” to make a new fasta with the corrected sequences.**



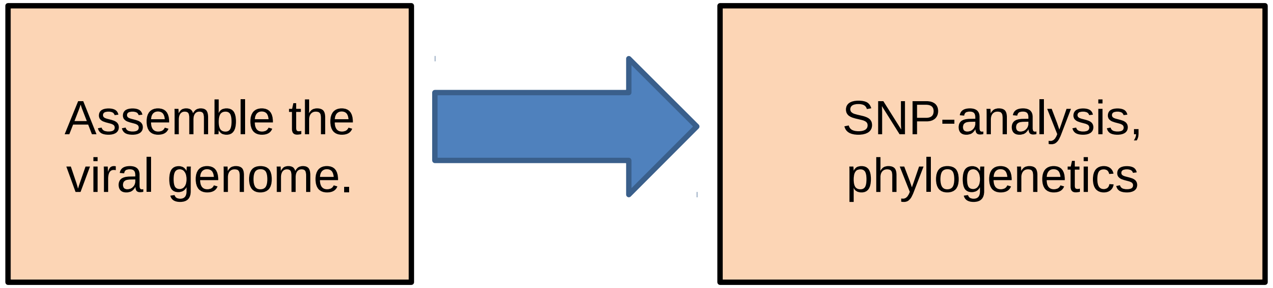
Read in the preliminary assembly

Make a new fasta containing the corrected sequences

The .vcf file containing list of changes to make.

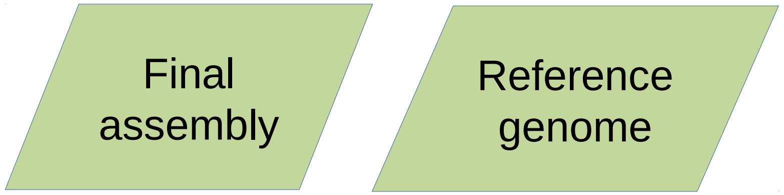
Save the new assembly

```
cat prelim_assembly.fasta | bcftools consensus mpileup.vcf.gz > final_assembly.fasta
```



Assemble the viral genome.

SNP-analysis, phylogenetics



- **Align the final assembly to the reference genome.**
- **This alignment will be used as input for the phylogenetic analysis.**

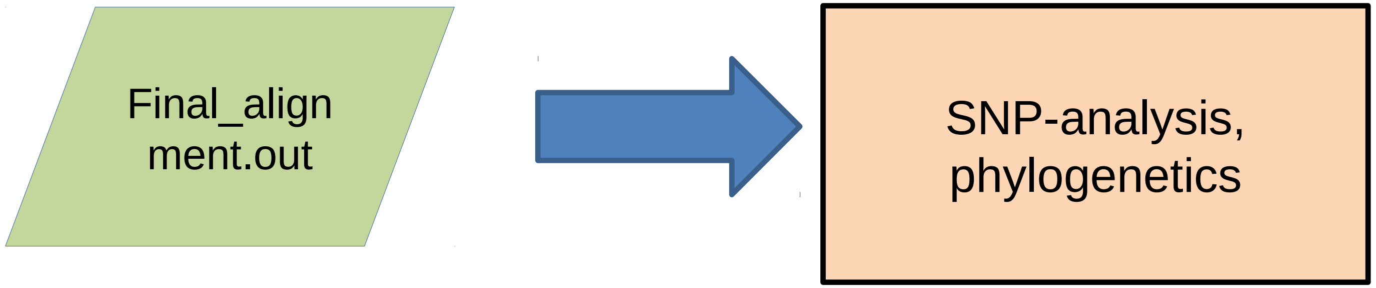
Read in the preliminary assembly

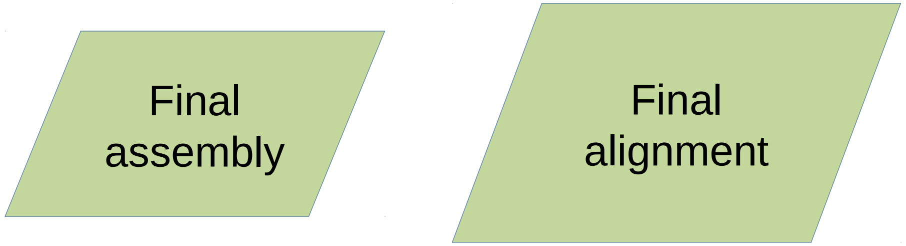
Run another aligner: mafft

Reference Genome

Save the alignment file.

```
cat final_assembly.fasta | mafft ebola_ref.fasta > Final_alignment.out
```





Final assembly

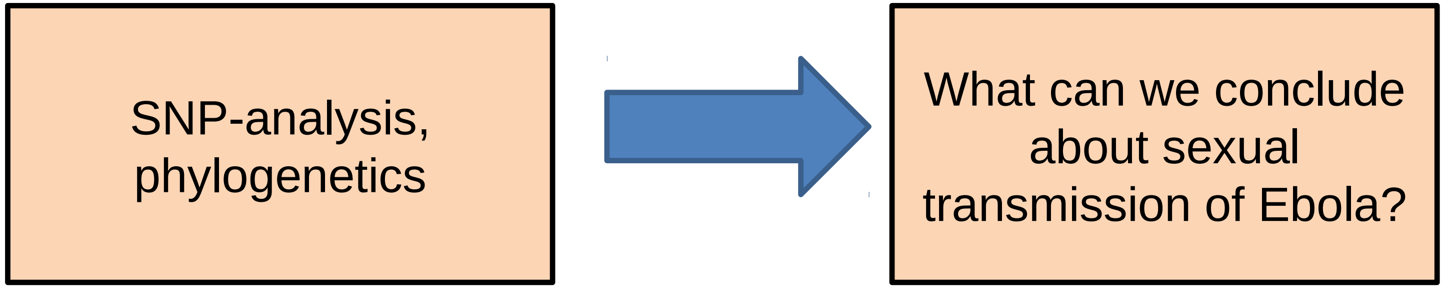
Final alignment

- **Run the haplonetwork_analysis.Rscript file, which contains instructions for running the haplotype network analysis in the R environment.**

A script (code) that runs the phylogenetic analysis.

The final alignment file.

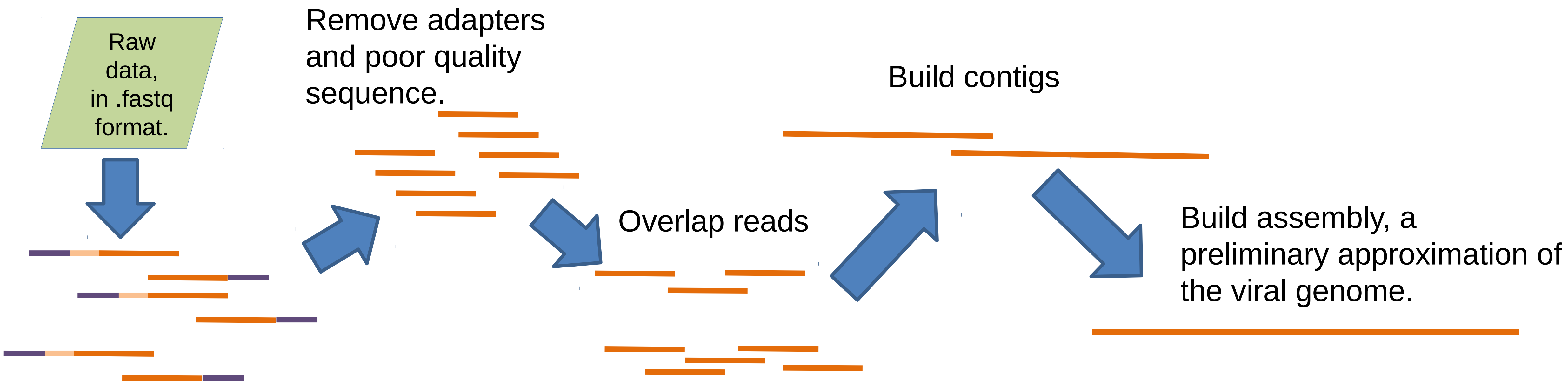
```
Rscript haplonetwork_analysis.Rscript Final_alignment.out
```



SNP-analysis, phylogenetics

What can we conclude about sexual transmission of Ebola?

De Novo Assembly



Raw data, in .fastq format.

Remove adapters and poor quality sequence.

Overlap reads

Build contigs

Build assembly, a preliminary approximation of the viral genome.

Raw data consists of sequences containing fragments of the Ebola genome. Ultimately, we need to take these fragments and assemble them into the complete genome.

File types

.gz	Appended to files that are compressed
.fasta	Simple format for storing sequence information.
.fastq	Stores sequence and quality information
.gff	General Feature Format: a list of genes and other genomic features, and their location in a particular genome.
.sam	Sequence Alignment/Map format. Links sequences (as from reads) to a position in a reference genome.
.bam	The compressed version of a .sam file.
.vcf	Variant Call Format; stores information about variation between sequences, as between reads and a reference genome.