CGS-Training-Module1

February 4, 2020

1 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 da

[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)
```

[3]:
```
<IPython.lib.display.IFrame at 0x7f7b8b648290>
```

[3]:
```
## Introduction to genomics assembly workflow
from IPython.display import IFrame
IFrame('documentation/final_pdfs/2_training_mod_013120_AssemblyPipe.pdf',␣
 ↪width=900, height=300)
```

[3]:
```
<IPython.lib.display.IFrame at 0x7f248164a890>
```

[2]:
```
# Step 1
# Define paths for input base directory, work directory and result directory in␣
 ↪config.yaml for any new datasets
#

base_dir ="/home/guest/projects/"
work_dir  = "/home/guest/projects/makono"
result_dir = "/home/guest/projects/results/"
reference_dir ="/home/guest/projects/makona/references/"
```

```
srefindex="/home/guest/projects/makona/seqindex/"
sreference="/home/guest/projects/makona/references/GCF_000848505.
↪1_ViralProj14703_genomic.fna"
pri_adaptors="/home/guest/projects/makona/references/pri_adaptors.fa"
```

[4]:
```
## Step 2

##Run following:
##
## shell command
## For paired end data
##   test fastqc read.R1_001.fastq.gz read.R2_001.fastq.gz -f fastq -o results/
↪fastqc > log.txt

from IPython.display import IFrame
IFrame('documentation/final_pdfs/3_training_mod_013120__Fastqc.pdf', width=900,␣
↪height=300)
```

[4]:
```
<IPython.lib.display.IFrame at 0x7f2495729510>
```

[13]:
```
## Run Step 2

!snakemake -s "popgen_fastqc.smk"
```

```
/home/guest/projects//makona/results directory exists
|--- Results directory is: /home/guest/projects//makona/results
|--- The current working directory is /home/guest/projects//makona
['Brett424_1_S4_L001']
|--- Number of samples to analyze: 1
|--- Sample Brett424_1_S4_L001 (reads: Brett424_1_S4_L001_R1_001.fastq.gz &
Brett424_1_S4_L001_R2_001.fastq.gz) will be processed
Building DAG of jobs…
Using shell: /bin/bash
Provided cores: 3
Rules claiming more threads will be scaled down.
Job counts:

        count   jobs

        1       all

        1       raw_fastqc

        2

[Wed Jan 29 09:54:16 2020]
```

```
rule raw_fastqc:
    input: samples/raw/Brett424_1_S4_L001_R1_001.fastq.gz,
samples/raw/Brett424_1_S4_L001_R2_001.fastq.gz
    output: results/fastqc/Brett424_1_S4_L001_R1_001_fastqc.html,
results/fastqc/Brett424_1_S4_L001_R1_001_fastqc.zip,
results/fastqc/Brett424_1_S4_L001_R2_001_fastqc.html,
results/fastqc/Brett424_1_S4_L001_R2_001_fastqc.zip,
results/fastqc/Brett424_1_S4_L001_fastqc.logfc.txt
    jobid: 1
    wildcards: smp=Brett424_1_S4_L001

Started analysis of Brett424_1_S4_L001_R1_001.fastq.gz
Approx 5% complete for Brett424_1_S4_L001_R1_001.fastq.gz
Approx 10% complete for Brett424_1_S4_L001_R1_001.fastq.gz
Approx 15% complete for Brett424_1_S4_L001_R1_001.fastq.gz
Approx 20% complete for Brett424_1_S4_L001_R1_001.fastq.gz
Approx 25% complete for Brett424_1_S4_L001_R1_001.fastq.gz
Approx 30% complete for Brett424_1_S4_L001_R1_001.fastq.gz
Approx 35% complete for Brett424_1_S4_L001_R1_001.fastq.gz
Approx 40% complete for Brett424_1_S4_L001_R1_001.fastq.gz
Approx 45% complete for Brett424_1_S4_L001_R1_001.fastq.gz
Approx 50% complete for Brett424_1_S4_L001_R1_001.fastq.gz
Approx 55% complete for Brett424_1_S4_L001_R1_001.fastq.gz
Approx 60% complete for Brett424_1_S4_L001_R1_001.fastq.gz
Approx 65% complete for Brett424_1_S4_L001_R1_001.fastq.gz
Approx 70% complete for Brett424_1_S4_L001_R1_001.fastq.gz
Approx 75% complete for Brett424_1_S4_L001_R1_001.fastq.gz
Approx 80% complete for Brett424_1_S4_L001_R1_001.fastq.gz
Approx 85% complete for Brett424_1_S4_L001_R1_001.fastq.gz
Approx 90% complete for Brett424_1_S4_L001_R1_001.fastq.gz
Approx 95% complete for Brett424_1_S4_L001_R1_001.fastq.gz
Started analysis of Brett424_1_S4_L001_R2_001.fastq.gz
Approx 5% complete for Brett424_1_S4_L001_R2_001.fastq.gz
Approx 10% complete for Brett424_1_S4_L001_R2_001.fastq.gz
Approx 15% complete for Brett424_1_S4_L001_R2_001.fastq.gz
Approx 20% complete for Brett424_1_S4_L001_R2_001.fastq.gz
Approx 25% complete for Brett424_1_S4_L001_R2_001.fastq.gz
Approx 30% complete for Brett424_1_S4_L001_R2_001.fastq.gz
Approx 35% complete for Brett424_1_S4_L001_R2_001.fastq.gz
Approx 40% complete for Brett424_1_S4_L001_R2_001.fastq.gz
Approx 45% complete for Brett424_1_S4_L001_R2_001.fastq.gz
Approx 50% complete for Brett424_1_S4_L001_R2_001.fastq.gz
Approx 55% complete for Brett424_1_S4_L001_R2_001.fastq.gz
```

```
Approx 60% complete for Brett424_1_S4_L001_R2_001.fastq.gz
Approx 65% complete for Brett424_1_S4_L001_R2_001.fastq.gz
Approx 70% complete for Brett424_1_S4_L001_R2_001.fastq.gz
Approx 75% complete for Brett424_1_S4_L001_R2_001.fastq.gz
Approx 80% complete for Brett424_1_S4_L001_R2_001.fastq.gz
Approx 85% complete for Brett424_1_S4_L001_R2_001.fastq.gz
Approx 90% complete for Brett424_1_S4_L001_R2_001.fastq.gz
Approx 95% complete for Brett424_1_S4_L001_R2_001.fastq.gz
[Wed Jan 29 09:54:30 2020]
Finished job 1.
1 of 2 steps (50%) done

[Wed Jan 29 09:54:30 2020]
localrule all:
    input: samples/raw/Brett424_1_S4_L001_R1_001.fastq.gz,
samples/raw/Brett424_1_S4_L001_R2_001.fastq.gz,
results/fastqc/Brett424_1_S4_L001_R1_001_fastqc.html,
results/fastqc/Brett424_1_S4_L001_R1_001_fastqc.zip,
results/fastqc/Brett424_1_S4_L001_R2_001_fastqc.html,
results/fastqc/Brett424_1_S4_L001_R2_001_fastqc.zip,
results/fastqc/Brett424_1_S4_L001_fastqc.logfc.txt
    jobid: 0

[Wed Jan 29 09:54:30 2020]
Finished job 0.
2 of 2 steps (100%) done
Complete log:
/home/guest/projects/.snakemake/log/2020-01-29T095416.164303.snakemake.log
Workflow finished, no error
```

[14]:
```
# Step 2 Fastqc results
from IPython.display import FileLink, FileLinks
FileLinks('makona/results/fastqc/.')
```

[14]:
```
makona/results/fastqc/./
  Brett424_1_S4_L001_fastqc.logfc.txt
  Brett424_1_S4_L001_R2_001_fastqc.html
  Brett424_1_S4_L001_R1_001_fastqc.zip
  Brett424_1_S4_L001_R2_001_fastqc.zip
  Brett424_1_S4_L001_R1_001_fastqc.html
```

[15]:
```
# Step 3
## Trimming the bait illumina adaptors and primers from Illumina sequencing␣
 ↪protocol using tool trimmomatic

##
## shell command
## For Paired end data
# "time java -jar trimmomatic-0.33.jar PE -threads 3 -trimlog logprefix input.
 ↪read.R1_001.fastq.gz input.read.R2_001.fastq.gz out.read.paired.R1.fastq out.
 ↪read.unpaired.R1.fastq out.fastq.paired.R2.fastq out.fastq.unpaired.R2.fastq␣
 ↪ILLUMINACLIP:input.primer.adaptor.fa:2:30:10 LEADING:3 TRAILING:3␣
 ↪SLIDINGWINDOW:4:15 MINLEN:30"
##


from IPython.display import IFrame
IFrame('documentation/final_pdfs/4_training_mod_013120__Trimv2.pdf', width=900,␣
 ↪height=300)
```

[15]:
```
<IPython.lib.display.IFrame at 0x7f07ae99e910>
```

[16]:
```
# Step 3 Run Trimmomatic on sequence reads using snakemake rule trimmomatics

!snakemake -s "popgen_trimmomatics.smk"
```

```
/home/guest/projects//makona/results directory exists
|--- Results directory is: /home/guest/projects//makona/results
|--- The current working directory is /home/guest/projects//makona
['Brett424_1_S4_L001']
|--- Number of samples to analyze: 1
|--- Sample Brett424_1_S4_L001 (reads: Brett424_1_S4_L001_R1_001.fastq.gz &
Brett424_1_S4_L001_R2_001.fastq.gz) will be processed
Building DAG of jobs…
Nothing to be done.
```

[16]:
```
# Sequence read summary after trimming adaptors and primers

# Reports

from IPython.display import FileLink, FileLinks
FileLinks('makona/results/primer_adapt_removed/.')
```

[16]:
```
makona/results/primer_adapt_removed/./
  Brett424_1_S4_L001_R1_unpaired.fastq
  Brett424_1_S4_L001_trimmolog.txt
  Brett424_1_S4_L001_R2_unpaired.fastq
  Brett424_1_S4_L001_R2_paired.fastq
```

```
Brett424_1_S4_L001_R1_paired.fastq
```

[6]:
```
# Step 4

## Reference mapping for Read correction
## Align reads to makona viral genome assembly fasta file

## Shell command
## time bwa mem -t 30 makona/references/GCF_000848505.1_ViralProj14703_genomic.
 ↪fna input.read.1.fastq input.read.2.fastq > sample1.assembly_align_mem_ref.
 ↪sam



from IPython.display import IFrame
IFrame('documentation/final_pdfs/5_training_mod_013120__Alignmentv2.pdf',␣
 ↪width=900, height=300)
```

[6]:
```
<IPython.lib.display.IFrame at 0x7f24815d2fd0>
```

[17]:
```
# Run step 4 for reference mapping for read correction using snakemake rule␣
 ↪refmapsam

!snakemake -s "popgen_refmapsam.smk" -n
```

```
/home/guest/projects//makona/results directory exists
|--- Results directory is: /home/guest/projects//makona/results
|--- The current working directory is /home/guest/projects//makona
['Brett424_1_S4_L001']
|--- Number of samples to analyze: 1
|--- Sample Brett424_1_S4_L001 (reads: Brett424_1_S4_L001_R1_001.fastq.gz &
Brett424_1_S4_L001_R2_001.fastq.gz) will be processed
Building DAG of jobs…
Nothing to be done.
```

[1]:
```
# Output from reference mapping
from IPython.display import FileLink, FileLinks
FileLinks('makona/results/ref_aligned/')
```

[1]:
```
makona/results/ref_aligned/
  Brett424_1_S4_L001_assembly_align_mem_ref_sorted.bam
  Brett424_1_S4_L001_assembly_align_mem_ref.sam
```

[7]:
```
## Step 5

## Sort sam file and convert to bam format file using samtools software
## Shell command:
```

```
## "time samtools sort -O BAM makona.aligned.mem.sam > sample1.
 ↪assembly_align_mem_ref_sorted.bam"
```

[18]:
```
!snakemake -s "popgen_samsort2bam.smk" -n
```

```
/home/guest/projects//makona/results directory exists
|--- Results directory is: /home/guest/projects//makona/results
|--- The current working directory is /home/guest/projects//makona
['Brett424_1_S4_L001']
|--- Number of samples to analyze: 1
|--- Sample Brett424_1_S4_L001 (reads: Brett424_1_S4_L001_R1_001.fastq.gz &
Brett424_1_S4_L001_R2_001.fastq.gz) will be processed
Building DAG of jobs…
Nothing to be done.
```

[17]:
```
# Output from reference mapping


from IPython.display import FileLink, FileLinks
FileLinks('makona/results/ref_aligned/.')
```

[17]:
```
makona/results/ref_aligned/./
  Brett424_1_S4_L001_assembly_align_mem_ref_sorted.bam
  Brett424_1_S4_L001_assembly_align_mem_ref.sam
```

[17]:
```
# Step 6

## Reference Guided Assembly graph using velvet assembler

## Shell Command:
## "time velveth out.assembly.dir input.kmernumber -bam -longPaired {output.
 ↪assembly.dir"


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

[17]:
```
<IPython.lib.display.IFrame at 0x7f07ae977410>
```

[19]:
```
!snakemake -s "popgen_assembly.smk" -n
```

```
/home/guest/projects//makona/results directory exists
|--- Results directory is: /home/guest/projects//makona/results
|--- The current working directory is /home/guest/projects//makona
['Brett424_1_S4_L001']
|--- Number of samples to analyze: 1
```

```
|--- Sample Brett424_1_S4_L001 (reads: Brett424_1_S4_L001_R1_001.fastq.gz &
Brett424_1_S4_L001_R2_001.fastq.gz) will be processed
Building DAG of jobs…
Nothing to be done.
```

[5]:
```
# Output from reference mapping


from IPython.display import FileLink, FileLinks
FileLinks('makona/results/velvet_assembly/')
```

[5]:
```
makona/results/velvet_assembly/
  Brett424_1_S4_L001_assembly_log.txt
  Brett424_1_S4_L001_logfile_assemref_27.txt
  Brett424_1_S4_L001_logfile_cindex.txt
  Brett424_1_S4_L001_reindex.log.txt
makona/results/velvet_assembly/Brett424_1_S4_L001_AssemRef/
  contigs.fa.fai
  contigs.fa
  Log
  Roadmaps
  contigs.fa.bwt
  contigs.fa.ann
  PreGraph
  contigs.fa.pac
  contigs.fa.sa
  velvet_asm.afg
  LastGraph
  Sequences
  stats.txt
  Graph
  contigs.fa.amb
```

[4]:
```
# Step 7

## Reference Guided Assembly map using velvet assembler
## Shell Command:

## "time velvetg input.out.assembly.dir -amos_file yes > output.logfile"


from IPython.display import IFrame
IFrame('documentation/final_pdfs/5_training_mod_013120__Alignmentv2.pdf',␣
 ↪width=900, height=300)
```

[4]:
```
<IPython.lib.display.IFrame at 0x7f9f8a5cf7d0>
```

[20]:
```
!snakemake -s "popgen_assembly_sgraph.smk" -n
```

```
/home/guest/projects//makona/results directory exists
|--- Results directory is: /home/guest/projects//makona/results
|--- The current working directory is /home/guest/projects//makona
['Brett424_1_S4_L001']
|--- Number of samples to analyze: 1
|--- Sample Brett424_1_S4_L001 (reads: Brett424_1_S4_L001_R1_001.fastq.gz &
Brett424_1_S4_L001_R2_001.fastq.gz) will be processed
Building DAG of jobs…
Nothing to be done.
```

[6]:
```
## Step 7 output
## # Output from velvet assembly

from IPython.display import FileLink, FileLinks
FileLinks('makona/results/velvet_assembly/.')
```

[6]:
```
makona/results/velvet_assembly/
  Brett424_1_S4_L001_assembly_log.txt
  Brett424_1_S4_L001_logfile_assemref_27.txt
  Brett424_1_S4_L001_logfile_cindex.txt
  Brett424_1_S4_L001_reindex.log.txt
makona/results/velvet_assembly/Brett424_1_S4_L001_AssemRef/
  contigs.fa.fai
  contigs.fa
  Log
  Roadmaps
  contigs.fa.bwt
  contigs.fa.ann
  PreGraph
  contigs.fa.pac
  contigs.fa.sa
  velvet_asm.afg
  LastGraph
  Sequences
  stats.txt
  Graph
  contigs.fa.amb
```

[8]:
```
# Step 8

## Assembly quality assesment stastics and gene prediction
## Shell Command:

## "time quast.py step7.input.contig.fa -R chk.genome.fa -G chk.genome.gff -o␣
 ↪out.assembly.stat.reports --glimmer > output.logfile"
```

```
from IPython.display import IFrame
IFrame('documentation/final_pdfs/6_training_mod_013120__DraftQC.pdf',␣
 ↪width=900, height=300)
```

[8]: <IPython.lib.display.IFrame at 0x7f24815a2c10>

[21]:
```
!snakemake -s "popgen_assembly_predictgene.smk" -n
```

```
/home/guest/projects//makona/results directory exists
|--- Results directory is: /home/guest/projects//makona/results
|--- The current working directory is /home/guest/projects//makona
['Brett424_1_S4_L001']
|--- Number of samples to analyze: 1
|--- Sample Brett424_1_S4_L001 (reads: Brett424_1_S4_L001_R1_001.fastq.gz &
Brett424_1_S4_L001_R2_001.fastq.gz) will be processed
Building DAG of jobs…
Nothing to be done.
```

[10]:
```
## Step 8 Assembly reports

from IPython.display import HTML
HTML(filename="./makona/results/assembl_stats/
 ↪Brett424_1_S4_L001_reference_stats/report.html")
```

[10]: <IPython.core.display.HTML object>

[9]:
```
# Step 9
## Create index of contigs and map reads back to contig
## Shell command:

## "time bwa index -a bwtsw step7.input.contig.fa > output.logfile"

from IPython.display import IFrame
IFrame('documentation/final_pdfs/7_training_mod_013120__Polishv2.pdf',␣
 ↪width=900, height=300)
```

[9]: <IPython.lib.display.IFrame at 0x7f24816c9290>

[22]:
```
!snakemake -s "popgen_bwaindex_contig.smk" -n
```

```
/home/guest/projects//makona/results directory exists
|--- Results directory is: /home/guest/projects//makona/results
|--- The current working directory is /home/guest/projects//makona
['Brett424_1_S4_L001']
|--- Number of samples to analyze: 1
```

```
|--- Sample Brett424_1_S4_L001 (reads: Brett424_1_S4_L001_R1_001.fastq.gz &
Brett424_1_S4_L001_R2_001.fastq.gz) will be processed
Building DAG of jobs…
Nothing to be done.
```

[10]:
```
# Step 10
## "time bwa mem -t 30 step8.input.contig.fa {input.read1p} {input.read2p} >␣
↪{output.contigalign}"
```

[28]:
```
!snakemake -s "popgen_alignreads2contig.smk" -n
```

```
/home/guest/projects//makona/results directory exists
|--- Results directory is: /home/guest/projects//makona/results
|--- The current working directory is /home/guest/projects//makona
['Brett424_1_S4_L001']
|--- Number of samples to analyze: 1
|--- Sample Brett424_1_S4_L001 (reads: Brett424_1_S4_L001_R1_001.fastq.gz &
Brett424_1_S4_L001_R2_001.fastq.gz) will be processed
Building DAG of jobs…
Nothing to be done.
```

[28]:
```
# Step 11
## Coordinate sort sam files and convert to bam file using samtools
```

[28]:
```
<IPython.lib.display.IFrame at 0x7f18d467e210>
```

[29]:
```
!snakemake -s "popgen_sortSAM.smk" -n
```

```
/home/guest/projects//makona/results directory exists
|--- Results directory is: /home/guest/projects//makona/results
|--- The current working directory is /home/guest/projects//makona
['Brett424_1_S4_L001']
|--- Number of samples to analyze: 1
|--- Sample Brett424_1_S4_L001 (reads: Brett424_1_S4_L001_R1_001.fastq.gz &
Brett424_1_S4_L001_R2_001.fastq.gz) will be processed
Building DAG of jobs…
Nothing to be done.
```

[12]:
```
# Step 12

## "time samtools faidx configs.fa > output.logfile"
!snakemake -s "popgen_reindexContig.smk" -n
```

```
/home/guest/projects//makona/results directory exists
|--- Results directory is: /home/guest/projects//makona/results
|--- The current working directory is /home/guest/projects//makona
['Brett424_1_S4_L001']
|--- Number of samples to analyze: 1
```

```
|--- Sample Brett424_1_S4_L001 (reads: Brett424_1_S4_L001_R1_001.fastq.gz &
Brett424_1_S4_L001_R2_001.fastq.gz) will be processed
Building DAG of jobs…
Nothing to be done.
```

[30]:
```
# Step 13

## Variant Calling using samtools mpileup

## Shell Command:

## "time samtools mpileup -u -g -f step8.input.contig.fa step11.contig.read.
 ↪sorted.aligned.bam | bcftools call -v -m -O z -o output.mpileup.vcf.gz >␣
 ↪output.logfile"


from IPython.display import IFrame
IFrame('documentation/command_pdfs/training_mod_Draft_S139.pdf', width=900,␣
 ↪height=300)
```

[30]:
```
<IPython.lib.display.IFrame at 0x7f18c3fc41d0>
```

[31]:
```
!snakemake -s "popgen_variantsCall.smk" -n
```

```
/home/guest/projects//makona/results directory exists
|--- Results directory is: /home/guest/projects//makona/results
|--- The current working directory is /home/guest/projects//makona
['Brett424_1_S4_L001']
|--- Number of samples to analyze: 1
|--- Sample Brett424_1_S4_L001 (reads: Brett424_1_S4_L001_R1_001.fastq.gz &
Brett424_1_S4_L001_R2_001.fastq.gz) will be processed
Building DAG of jobs…
Nothing to be done.
```

[38]:
```
## Reports

from IPython.display import FileLink, FileLinks
FileLinks('makona/results/variants_calling/.')
```

[38]:
```
makona/results/variants_calling/./
  Brett424_1_S4_L001_mpileup.vcf.gz
  Brett424_1_S4_L001_mpileup.vcf.gz.csi
  Brett424_1_S4_L001_vcfindex.txt
  Brett424_1_S4_L001_snpcall.txt
```

[32]:
```
!snakemake -s "popgen_vcfindex.smk" -n
```

```
/home/guest/projects//makona/results directory exists
```

```
|--- Results directory is: /home/guest/projects//makona/results
|--- The current working directory is /home/guest/projects//makona
['Brett424_1_S4_L001']
|--- Number of samples to analyze: 1
|--- Sample Brett424_1_S4_L001 (reads: Brett424_1_S4_L001_R1_001.fastq.gz &
Brett424_1_S4_L001_R2_001.fastq.gz) will be processed
Building DAG of jobs…
Nothing to be done.
```

[12]:
```
# Step 15

## Build sequences consensus

## Shell Command:

## "time cat step8.input.contig.fa | bcftools consensus output.mpileup.vcf.gz >␣
↪output.consensus.fa
```

[33]:
```
!snakemake -s "popgen_buildConsensus.smk" -n
```

```
/home/guest/projects//makona/results directory exists
|--- Results directory is: /home/guest/projects//makona/results
|--- The current working directory is /home/guest/projects//makona
['Brett424_1_S4_L001']
|--- Number of samples to analyze: 1
|--- Sample Brett424_1_S4_L001 (reads: Brett424_1_S4_L001_R1_001.fastq.gz &
Brett424_1_S4_L001_R2_001.fastq.gz) will be processed
Building DAG of jobs…
Nothing to be done.
```

[39]:
```
## Reports

from IPython.display import FileLink, FileLinks
FileLinks('makona/results/consensus_seq/.')
```

[39]:
```
makona/results/consensus_seq/./
  Brett424_1_S4_L001_consensus.fa
```

[2]:
```
# Step 16

## Consensus muliple alignment

## Shell Command:

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

```
from IPython.display import IFrame
IFrame('documentation/final_pdfs/8_training_mod_013120__GenAlignv3.pdf',␣
 ↪width=900, height=300)
```

[2]: <IPython.lib.display.IFrame at 0x7f7b8b62a190>

[5]:
```
!snakemake -s "popgen_maff_alignment_view.smk"
```

```
/home/guest/projects//makona/results directory exists
|--- Results directory is: /home/guest/projects//makona/results
|--- The current working directory is /home/guest/projects//makona
['Brett424_1_S4_L001']
|--- Number of samples to analyze: 1
|--- Sample Brett424_1_S4_L001 (reads: Brett424_1_S4_L001_R1_001.fastq.gz &
Brett424_1_S4_L001_R2_001.fastq.gz) will be processed
Building DAG of jobs…
Using shell: /bin/bash
Provided cores: 3
Rules claiming more threads will be scaled down.
Job counts:
        count   jobs
        1       all
        1       maff_align
        2

[Mon Feb  3 14:15:06 2020]
Job 2:  --- mafft alignment

nthread = 0
nthreadpair = 0
nthreadtb = 0
ppenalty_ex = 0
stacksize: 8192 kb
generating a scoring matrix for nucleotide (dist=200) … done
Gap Penalty = -1.53, +0.00, +0.00



Making a distance matrix ..

There are 9495 ambiguous characters.
    1 / 7
done.

Constructing a UPGMA tree (efffree=0) …
```

```
    0 / 7
done.

Progressive alignment 1/2…
STEP     6 / 6
done.

Making a distance matrix from msa..
    0 / 7
done.

Constructing a UPGMA tree (efffree=1) …
    0 / 7
done.

Progressive alignment 2/2…
STEP     6 / 6
done.

disttbfast (nuc) Version 7.450
alg=A, model=DNA200 (2), 1.53 (4.59), -0.00 (-0.00), noshift, amax=0.0
0 thread(s)


Strategy:
 FFT-NS-2 (Fast but rough)
 Progressive method (guide trees were built 2 times.)

If unsure which option to use, try 'mafft --auto input > output'.
For more information, see 'mafft --help', 'mafft --man' and the mafft page.

The default gap scoring scheme has been changed in version 7.110 (2013 Oct).
It tends to insert more gaps into gap-rich regions than previous versions.
To disable this change, add the --leavegappyregion option.


real    0m1.312s
user    0m1.196s
sys     0m0.049s
[Mon Feb  3 14:15:07 2020]
Finished job 2.
1 of 2 steps (50%) done

[Mon Feb  3 14:15:07 2020]
```

```
localrule all:
    input: samples/raw/Brett424_1_S4_L001_R1_001.fastq.gz,
samples/raw/Brett424_1_S4_L001_R2_001.fastq.gz,
results/fastqc/Brett424_1_S4_L001_R1_001_fastqc.html,
results/fastqc/Brett424_1_S4_L001_R1_001_fastqc.zip,
results/fastqc/Brett424_1_S4_L001_R2_001_fastqc.html,
results/fastqc/Brett424_1_S4_L001_R2_001_fastqc.zip,
results/fastqc/Brett424_1_S4_L001_fastqc.logfc.txt,
results/primer_adapt_removed/Brett424_1_S4_L001_R1_paired.fastq,
results/primer_adapt_removed/Brett424_1_S4_L001_R1_unpaired.fastq,
results/primer_adapt_removed/Brett424_1_S4_L001_R2_paired.fastq,
results/primer_adapt_removed/Brett424_1_S4_L001_R2_unpaired.fastq,
results/primer_adapt_removed/Brett424_1_S4_L001_trimmolog.txt,
results/ref_aligned/Brett424_1_S4_L001_assembly_align_mem_ref.sam,
results/ref_aligned/Brett424_1_S4_L001_assembly_align_mem_ref_sorted.bam,
results/velvet_assembly/Brett424_1_S4_L001_assembly_log.txt,
results/velvet_assembly/Brett424_1_S4_L001_logfile_assemref_27.txt,
results/assembl_stats/Brett424_1_S4_L001_logfile_assembly_predictgene.txt,
results/velvet_assembly/Brett424_1_S4_L001_logfile_cindex.txt,
results/assembly_aligned/Brett424_1_S4_L001_contigalign.sam,
results/assembly_aligned/Brett424_1_S4_L001_contigalign.bam,
results/velvet_assembly/Brett424_1_S4_L001_reindex.log.txt,
results/variants_calling/Brett424_1_S4_L001_snpcall.txt,
results/variants_calling/Brett424_1_S4_L001_mpileup.vcf.gz,
results/variants_calling/Brett424_1_S4_L001_vcfindex.txt,
results/consensus_seq/Brett424_1_S4_L001_consensus.fa,
results/variants_stats/Brett424_1_S4_L001_vcf.stats,
results/maff_alignment/Brett424_1_S4_L001_mafft_catconsensus.out,
results/maff_alignment/Brett424_1_S4_L001_mafft_align.out
    jobid: 0

[Mon Feb  3 14:15:07 2020]
Finished job 0.
```

```
2 of 2 steps (100%) done
Complete log:

/home/guest/projects/.snakemake/log/2020-02-03T141506.500464.snakemake.log
Workflow finished, no error
```

[ ]:
```
## View MSA alignment

library(shiny)
runApp()
```

```
Listening on http://127.0.0.1:7764
```

[2]:
```
## Reports

from IPython.display import FileLink, FileLinks
FileLinks('makona/results/maff_haplo/.')
```

[2]:
```
makona/results/maff_haplo/./
  makona_muliple_alignment.out
  final_contactenated_mafft.fa
```

[3]:
```
## Shell Command:
# "time bcftools stats -F step8.input.contig.fa -s step11.output.mpileup.vcf.gz␣
 ↪> output.variants.stat"

!snakemake -s "popgen_variants_stat.smk" -n
```

```
/home/guest/projects//makona/results directory exists
|--- Results directory is: /home/guest/projects//makona/results
|--- The current working directory is /home/guest/projects//makona
['Brett424_1_S4_L001']
|--- Number of samples to analyze: 1
|--- Sample Brett424_1_S4_L001 (reads: Brett424_1_S4_L001_R1_001.fastq.gz &
Brett424_1_S4_L001_R2_001.fastq.gz) will be processed
Building DAG of jobs…
Nothing to be done.
```

[4]:
```
## Reports

from IPython.display import FileLink, FileLinks
FileLinks('makona/results/variants_stats/.')
```

[4]:
```
makona/results/variants_stats/./
  Brett424_1_S4_L001_vcf.stats
```

[5]:
```
!head -100 makona/results/variants_stats/Brett424_1_S4_L001_vcf.stats
```

```
# This file was produced by bcftools stats (1.9+htslib-1.9) and can be plotted
using plot-vcfstats.
# The command line was: bcftools stats  -F
results/velvet_assembly/Brett424_1_S4_L001_AssemRef/contigs.fa -s -
results/variants_calling/Brett424_1_S4_L001_mpileup.vcf.gz
#
# Definition of sets:
# ID    [2]id   [3]tab-separated file names
ID      0       results/variants_calling/Brett424_1_S4_L001_mpileup.vcf.gz
# SN, Summary numbers:
#   number of records   .. number of data rows in the VCF
#   number of no-ALTs   .. reference-only sites, ALT is either "." or identical
to REF
#   number of SNPs      .. number of rows with a SNP
#   number of MNPs      .. number of rows with a MNP, such as CC>TT
#   number of indels    .. number of rows with an indel
#   number of others    .. number of rows with other type, for example a
symbolic allele or
#                          a complex substitution, such as ACT>TCGA
#   number of multiallelic sites     .. number of rows with multiple alternate
alleles
#   number of multiallelic SNP sites .. number of rows with multiple alternate
alleles, all SNPs
#
#   Note that rows containing multiple types will be counted multiple times, in
each
#   counter. For example, a row with a SNP and an indel increments both the SNP
and
#   the indel counter.
#
# SN    [2]id   [3]key  [4]value
SN      0       number of samples:      1
SN      0       number of records:      15583
SN      0       number of no-ALTs:      0
SN      0       number of SNPs: 14943
SN      0       number of MNPs: 0
SN      0       number of indels:       640
SN      0       number of others:       0
SN      0       number of multiallelic sites:   14
SN      0       number of multiallelic SNP sites:       14
# TSTV, transitions/transversions:
# TSTV  [2]id   [3]ts   [4]tv   [5]ts/tv        [6]ts (1st ALT) [7]tv (1st ALT)
[8]ts/tv (1st ALT)
TSTV    0       7096    7861    0.90    7092    7851    0.90
# ICS, Indel context summary:
```

```
# ICS   [2]id   [3]repeat-consistent    [4]repeat-inconsistent  [5]not
applicable       [6]c/(c+i) ratio
ICS     0       136     74      430     0.6476
# ICL, Indel context by length:
# ICL   [2]id   [3]length of repeat element     [4]repeat-consistent deletions)
[5]repeat-inconsistent deletions        [6]consistent insertions
[7]inconsistent insertions      [8]c/(c+i) ratio
ICL     0       2       33      10      83      27      0.7582
ICL     0       3       7       5       7       12      0.4516
ICL     0       4       3       2       2       4       0.4545
ICL     0       5       0       2       1       4       0.1429
ICL     0       6       0       2       0       4       0.0000
ICL     0       7       0       0       0       0       0.0000
ICL     0       8       0       1       0       0       0.0000
ICL     0       9       0       0       0       1       0.0000
ICL     0       10      0       0       0       0       0.0000
# SiS, Singleton stats:
# SiS   [2]id   [3]allele count [4]number of SNPs       [5]number of transitions
[6]number of transversions      [7]number of indels     [8]repeat-consistent
[9]repeat-inconsistent  [10]not applicable
SiS     0       1       1532    908     624     164     44      18      102
# AF, Stats by non-reference allele frequency:
# AF    [2]id   [3]allele frequency     [4]number of SNPs       [5]number of
transitions        [6]number of transversions      [7]number of indels
[8]repeat-consistent    [9]repeat-inconsistent  [10]not applicable
AF      0       0.000000        1532    908     624     164     44      18
102
AF      0       0.990000        13425   6188    7237    476     92      56
328
# QUAL, Stats by quality:
# QUAL  [2]id   [3]Quality      [4]number of SNPs       [5]number of transitions
(1st ALT)      [6]number of transversions (1st ALT)    [7]number of indels
QUAL    0       3       983     438     545     48
QUAL    0       4       382     188     194     23
QUAL    0       5       597     281     316     49
QUAL    0       6       324     141     183     22
QUAL    0       7       319     157     162     23
QUAL    0       8       2291    1286    1005    26
QUAL    0       9       181     84      97      12
QUAL    0       10      225     108     117     42
QUAL    0       11      197     94      103     8
QUAL    0       12      179     90      89      13
QUAL    0       13      179     95      84      8
QUAL    0       14      192     89      103     16
QUAL    0       15      174     83      91      8
QUAL    0       16      175     83      92      11
QUAL    0       17      155     71      84      10
QUAL    0       18      234     122     112     11
```

```
QUAL    0       19      137     58      79      8
QUAL    0       20      157     68      89      9
QUAL    0       21      231     137     94      10
QUAL    0       22      132     64      68      9
QUAL    0       23      127     64      63      6
QUAL    0       24      166     72      94      11
QUAL    0       25      163     69      94      7
QUAL    0       26      126     48      78      4
QUAL    0       27      154     68      86      3
QUAL    0       28      124     65      59      9
QUAL    0       29      134     64      70      6
QUAL    0       30      551     279     272     9
QUAL    0       31      116     57      59      6
QUAL    0       32      123     66      57      4
QUAL    0       33      135     68      67      7
QUAL    0       34      99      51      48      5
QUAL    0       35      91      38      53      9
QUAL    0       36      97      47      50      6
QUAL    0       37      88      41      47      5
QUAL    0       38      130     54      76      1
QUAL    0       39      107     52      55      4
QUAL    0       40      96      40      56      5
QUAL    0       41      90      45      45      5
QUAL    0       42      91      42      49      4
QUAL    0       43      92      39      53      2
QUAL    0       44      80      37      43      4
QUAL    0       45      142     77      65      5
```

[19]:
```
## Haplotype network and SNP analysis

## Shell

!snakemake -s "popgen_haplonetwork.smk"
```

```
/home/guest/projects//makona/results directory exists
|--- Results directory is: /home/guest/projects//makona/results
|--- The current working directory is /home/guest/projects//makona
['Brett424_1_S4_L001']
|--- Number of samples to analyze: 1
|--- Sample Brett424_1_S4_L001 (reads: Brett424_1_S4_L001_R1_001.fastq.gz &
Brett424_1_S4_L001_R2_001.fastq.gz) will be processed
Building DAG of jobs…
Nothing to be done.
Complete log:

/home/guest/projects/.snakemake/log/2020-02-03T220909.326866.snakemake.log
Workflow finished, no error
```

[20]:
```
from IPython.display import IFrame
IFrame('documentation/final_pdfs/9_training_mod_013120__HapNetv2.pdf',␣
↪width=900, height=300)
```

[20]:
```
<IPython.lib.display.IFrame at 0x7f9f899ff410>
```

[21]:
```
## Reports

from IPython.display import FileLink, FileLinks
FileLinks('makona/results/haplotype_network/')
```

[21]:
```
makona/results/haplotype_network/
  Brett424_1_S4_L001_logfileR.txt
  study_haplonetwork.png
```

[11]:
```
from IPython.display import HTML
HTML(filename="./rscript_haplo.nb.html")
```

[11]:
```
<IPython.core.display.HTML object>
```

[10]:
```
## References

## Shell

from IPython.display import IFrame
IFrame('documentation/final_pdfs/10_training_mod_013120__CommandLine.pdf',␣
↪width=900, height=300)
```

[10]:
```
<IPython.lib.display.IFrame at 0x7f07ae977210>
```

[ ]:
```

```