Introduction:

The variant calling pipeline is an automated workflow containing the following steps:

- processing read files
- quality- and adapter trimming of the reads
- mapping of the reads to a reference genome
- sorting and merging of all mapped read sets of one sample
- filtering duplicate reads from the alignment files
- variant calling

Used Programs:

For some programs in the pipeline, is is possible to add some custom parameters in the config.json file *(eg. ploidy level in the freebayes parameters)*.
Detailed documentation of the used programs can be found here:

wget:	https://www.gnu.org/software/wget/
python 3:	https://www.python.org/download/releases/3.0.1/
snakemake:	https://bitbucket.org/johanneskoester/snakemake/wiki/Home
Trimmomatic**:**	http://www.usadellab.org/cms/uploads/supplementary/Trimmomatic/TrimmomaticManual_V0.32.pdf
BWA:	http://bio-bwa.sourceforge.net/bwa.shtml
Samtools:	http://www.htslib.org/doc/samtools-1.2.html
Picardtools:	https://broadinstitute.github.io/picard/command-line-overview.html
Freebayes:	https://github.com/ekg/freebayes/blob/master/README.md

code from the VLPB git repository: https://github.com/vlpb3/NGS_snakemake_pipelines

Prerequisites:

In the supplied JSON file, add the correct paths and other metadata of the reads that will be used in the variantcalling pipeline.

Note that the forward - and reverse read files must be gzipped fastq files, and require the postfix *“*_1.fastq.gz”* and *“*_2.fastq.gz”* respectively for the workflow to run properly.

Also the prefix of both forward - and reverse read file must be equal.
So :

> **sample1_readsetA_1.fastq.gz**
> **sample1_readsetA_2.fastq.gz**

are valid names, unlike:

> **sample1_readsetA_1.fastq.gz**
> **Sample1_readsetA_2.fastq.gz**

these are not valid, because the prefix of both files is not the same (case sensitive)

note:
In the working directory where the snakemake pipeline is started, the following temporarily subdirectories are created, and removed afterwards:

> ./trimmed
> ./mapped
> ./mapped.sorted

Make sure that there are no matching directories in your working directory, they will be removed!

CONFIG FILE:

All the metadata needed for the workflow to run, is combined in one config file.
This config file is in JSON format (detailed information about JSON and an online JSON validator can be found here: http://jsonformatter.curiousconcept.com/)

The following snippets need to be edited before use:

config.json snippets:
readfiles block:

```
{
  "samples":{
    "SAMPLE_NAME_1":{
      "LIBRARY_NAME_A":{
        "readsets":{
          "READSET_NAME_A1":[
            "/PATH_TO_FORWARD_READS_sample1-libA-readset-A1/READSFILE_1.fastq.gz",
            "/PATH_TO_REVERSE_READS_sample1-libA-readset-A1/READSFILE_2.fastq.gz"
          ],
          "READSET_NAME_A2":[
            "/PATH_TO_FORWARD_READS_sample1-libA-readset-A2/READSFILE_1.fastq.gz",
            "/PATH_TO_FORWARD_READS_sample1-libA-readset-A2/READSFILE_2.fastq.gz"
          ]
        },
        "type":"pe",
        "platform":"illumina"
      }
    },
    "SAMPLE_NAME_2":{
      "LIBRARY_NAME_B":{
        "readsets":{
          "READSET_NAME_B1":[
            "/PATH_TO_FORWARD_READS_sample2-libB-readset-B1/READSFILE_1.fastq.gz",
            "/PATH_TO_FORWARD_READS_sample2-libB-readset-B1/READSFILE_2.fastq.gz"
          ]
        },
        "type":"pe",
        "platform":"illumina"
      }
    }
  },
```

All colored fields are custom fields and have to be edited. Other fields should not be changed.

SAMPLE_NAME: ***name of the sample (string, eg “Athal_WU_0”)***
LIBRARY_NAME: ***name of the library (string eg. “SRA116748”)***
READSET_NAME: ***name ot the readset (string, eg. “1” or “A”)***
PATH_TO_FORWARD_READS: ***absolute path to forward readfile (string, eg:***

“/data/reads/Athal_WU_0/SRA116748_1.fastq.gz”)

PATH_TO_REVERSE_READS:: ***absolute path to forward readfile (string, eg: “/data/reads/Athal_WU_0/SRA116748_2.fastq.gz”)***

For the variant calling pipeline the insertSize and insertSizeStDev is not required

config.json snippet:
Trimmomatic block**:**

Documentation about Trimmomatic parameters can be found here**:**
http://www.usadellab.org/cms/uploads/supplementary/Trimmomatic/TrimmomaticManual_V0.32.pdf

These parameters can be adjusted in the config.json file:

```
"illuminaAdapters":    "/PathToAdaptersFasta/TruSeq2-PE.fa",
"trimmomatic":{
    "seedMisMatches":"2",
    "palindromeClipTreshold":"30",
    "simpleClipThreshhold":"10",
    "LeadMinTrimQual":"3",
    "TrailMinTrimQual":"3",
    "windowSize":"4",
    "avgMinQual":"15",
    "minReadLen":"36",
    "phred":"-phred33"
```

config.json snippet:
freebayes block**:**

Freebayes: https://github.com/ekg/freebayes/blob/master/README.md

These parameters can be adjusted in the config.json file:

```
"freebayes":{
    "optionalOpts":{
        "--ploidy":4,
        "--read-indel-limit":3,
        "--min-mapping-quality":30,
        "--min-base-quality":13,
        "--theta":0.01,
        "--min-alternate-count":4,
        "--min-alternate-fraction":0.15
```

config.json snippet:
executable locations block**:**

If some executables are allready installed on your system, it is possible to add the paths in the executables block:

```
 "executables":{
     "trimmomatic": "/PATH_TO_TRIMMOMATIC/trimmomatic-0.32.jar",
     "bwaMem" : "/PATH_TO_BWA/bwa-0.7.7/",
     "samtools" : "/PATH_TO_SAMTOOLS/bin/",
     "picardTools" : "/PATH_TO_PICARDTOOLS/",
     "freebayes" : "/PATH_TO_FREEBAYES/bin/"
```

Execution of the pipeline:

In the working directory where your config.json file is stored, type:

```
snakemake --snakefile {/path/to/your/workflow}
```

Output:

after the snakemake pipeline finishes, there are three subdirectories within your working directory where the snakemake pipeline was started::

- reads contain symlinks to the readfiles supplied in the config.json file
- processedbam contain processed (rm duplicates, add readgroups, sorted, merged) alignment (.bam) files with index (.bai) for all samples These files are used in the variantcalling step..
- variantCalling contains the variantCall (.vcf) files for each sample

Information about the bam/sam files and the vcf files can be found here:

https://samtools.github.io/hts-specs/SAMv1.pdf
https://samtools.github.io/hts-specs/VCFv4.2.pdf

Testdata:

when using the supplied testdatasets of Solanum tuberosum (also used in the VLPB hands-on workshop of december 2014):

READS:

```
 "samples":{
   "ST_sample1":{
     "lib1":{
       "readsets":{
         "1":[
           "/PathTo/Solanum_tuberosum/raw_reads/ST_sample001_1.fastq.gz",
           "/PathTo/Solanum_tuberosum/raw_reads/ST_sample001_2.fastq.gz"
         ]
       },
       "type":"pe",
       "platform":"illumina"
     }
   },
   "ST_sample2":{
     "lib1":{
       "readsets":{
         "1":[
           "/PathTo/Solanum_tuberosum/raw_reads/ST_sample002_1.fastq.gz",
           "/PathTo/Solanum_tuberosum/raw_reads/ST_sample002_2.fastq.gz"
         ]
       },
       "type":"pe",
       "platform":"illumina"
     }
   }
  },
```

REFERENCE:

```
""refGenome":"/PathTo/refgenome/ST4.03ch05.fasta",
```

OPTIONS:

```
"freebayes":{
     "optionalOpts":{
      "--ploidy":4,
      "--read-indel-limit":3,
      "--min-mapping-quality":30,
      "--min-base-quality":13,
      "--theta":0.01,
      "--min-alternate-count":4,
      "--min-alternate-fraction":0.15
    }
  },
  "illuminaAdapters":"/PathTo/Trimmomatic-0.32/adapters/TruSeq2-PE.fa",
  "trimmomatic":{
    "seedMisMatches":"2",
    "palindromeClipTreshold":"30",
    "simpleClipThreshhold":"10",
    "LeadMinTrimQual":"3",
    "TrailMinTrimQual":"3",
    "windowSize":"4",
    "avgMinQual":"15",
    "minReadLen":"36",
    "phred":"-phred33"
```

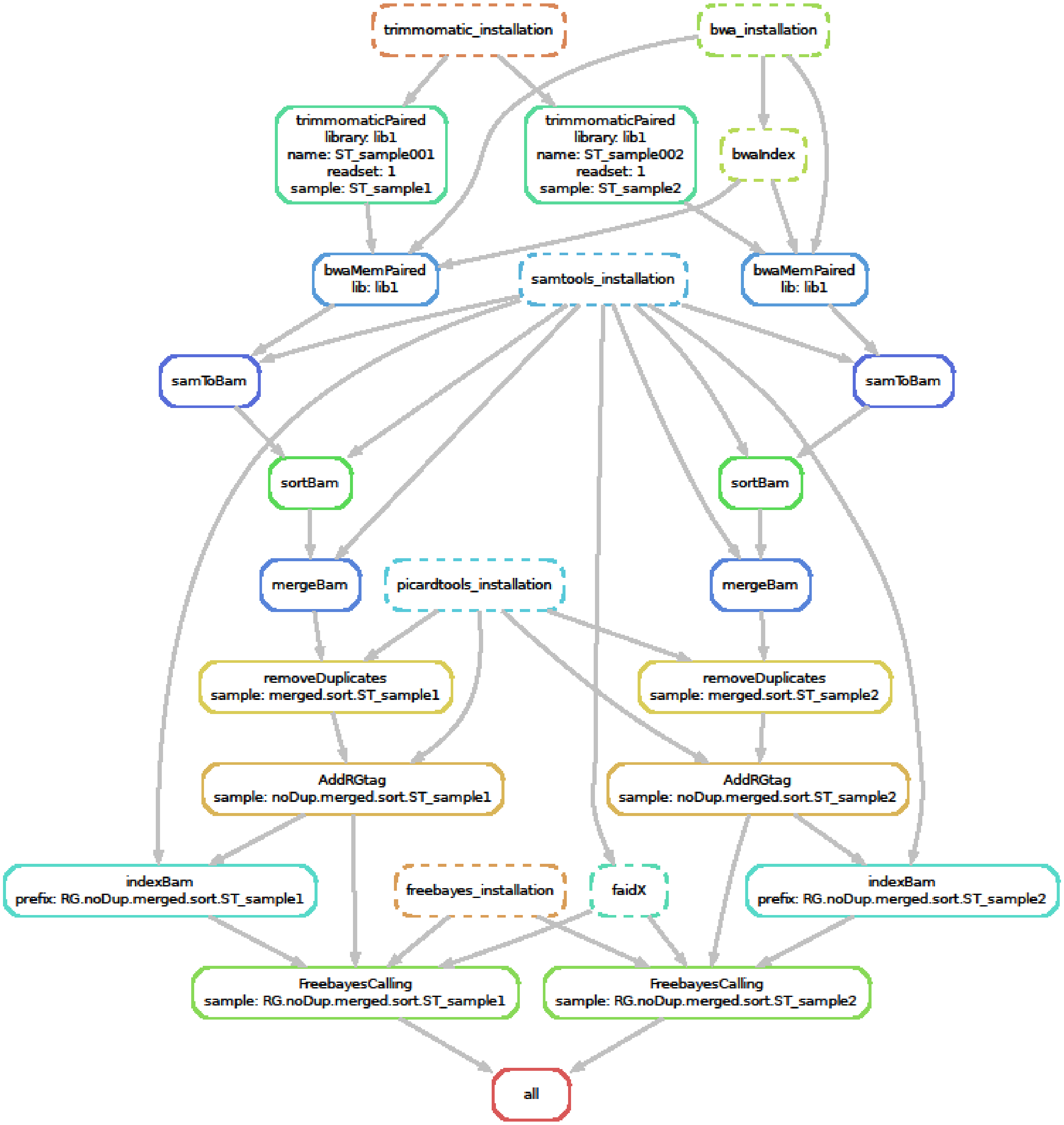
After running the freebayes variantcalling pipeline, the following files are generated:

```
./reads/ST_sample1/lib1/1/ST_sample001_1.fastq.gz  -> /abs/path/to/readfile/ST_sample001_1.fastq.gz
./reads/ST_sample1/lib1/1/ST_sample001_2.fastq.gz  -> /abs/path/to/readfile/ST_sample001_2.fastq.gz
./reads/ST_sample1/lib1/1/ST_sample002_1.fastq.gz  -> /abs/path/to/readfile/ST_sample002_1.fastq.gz
./reads/ST_sample1/lib1/1/ST_sample002_2.fastq.gz  -> /abs/path/to/readfile/ST_sample002_2.fastq.gz

./processedbam/RG.noDup.merged.sort.ST_sample1.bam
./processedbam/RG.noDup.merged.sort.ST_sample1.bam.bai
./processedbam/RG.noDup.merged.sort.ST_sample2.bam
./processedbam/RG.noDup.merged.sort.ST_sample2.bam.bai

./variantCalling/fb.RG.noDup.merged.sort.ST_sample1.vcf
./variantCalling/fb.RG.noDup.merged.sort.ST_sample2.vcf
```

the variant files (.vcf) contain 816 and 827 variants respectively:

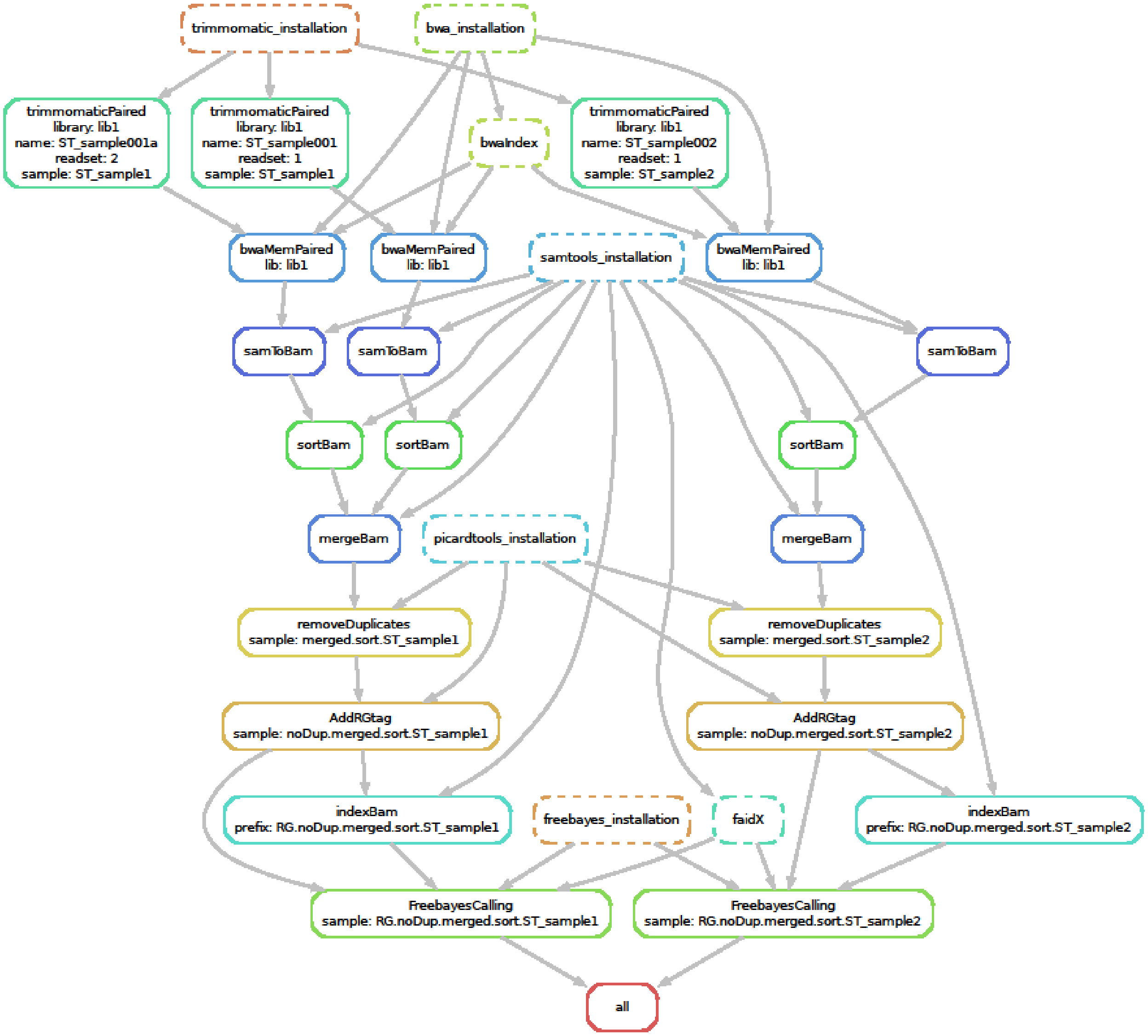
```
cat fb.RG.noDup.merged.sort.ST_sample1.vcf |grep -c '^ST'
816
cat fb.RG.noDup.merged.sort.ST_sample2.vcf |grep -c '^ST'
827
```

Directed Acyclic Graph (DAG) diagram of the variantcalling pipeline on the Solanum tuberosum testset:



:

Here an example workflow with one sample (sample1) having multiple readsets:


trimmomatic_installation
bwa_installation
trimmomaticPaired
library: lib1
name: ST_sample001a
readset: 2
sample: ST_sample1
trimmomaticPaired
library: lib1
name: ST_sample001
readset: 1
sample: ST_sample1
bwaIndex
trimmomaticPaired
library: lib1
name: ST_sample002
readset: 1
sample: ST_sample2
bwaMemPaired
lib: lib1
bwaMemPaired
lib: lib1
samtools_installation
bwaMemPaired
lib: lib1
samToBam
samToBam
samToBam
sortBam
sortBam
sortBam
mergeBam
picardtools_installation
mergeBam
removeDuplicates
sample: merged.sort.ST_sample1
removeDuplicates
sample: merged.sort.ST_sample2
AddRGtag
sample: noDup.merged.sort.ST_sample1
AddRGtag
sample: noDup.merged.sort.ST_sample2
indexBam
prefix: RG.noDup.merged.sort.ST_sample1
freebayes_installation
faidX
indexBam
prefix: RG.noDup.merged.sort.ST_sample2
FreebayesCalling
sample: RG.noDup.merged.sort.ST_sample1
FreebayesCalling
sample: RG.noDup.merged.sort.ST_sample2
all