# **Base Workflow**

Trello: <https://trello.com/b/pcWKqbXv/stream-b-ngs-variant-calling-illumina>

The 'core' pipeline is based on [GATK Best Practices](https://software.broadinstitute.org/gatk/best-practices/bp_3step.php?case=GermShortWGS) version **v3.5**:

### **Input: uBAM, or BAM, or Fastq**

### **Steps**

#### FastQC (step 2 does not depend on this output)

See below for testing

#### Trimmomatic - Phele

Notes:

* Using branch: <https://github.com/common-workflow-language/workflows/tree/new_trimmo>

#Manual run:

java org.usadellab.trimmomatic.Trimmomatic [SE|PE] \

-threads 5 [-phred33|-phred64] [input\_fastq(s)] \

[input\_fastq].trimmed.fastq \

ILLUMINACLIP:[NexteraPE-PE].fa:2:30:10:1:true \

MAXINFO:40:0.70 \

MINLEN:60

#### 

#Test yml:

input\_read1\_fastq\_file:

class: File

path: /data\_small/small\_fastq/project.NIST\_NIST7035\_H7AP8ADXX\_TAAGGCGA\_1\_NA12878.bwa.markDuplicates.chr22\_1.fq

input\_read2\_fastq\_file:

class: File

path: /data\_small/small\_fastq/project.NIST\_NIST7035\_H7AP8ADXX\_TAAGGCGA\_1\_NA12878.bwa.markDuplicates.chr22\_1.fq

phred: "33"

illuminaclip: 2:30:10:1:true

input\_adapters\_file:

class: File

path: NexteraPE-PE.fa

maxinfo: "40:0.70"

minlen: 60

end\_mode: PE

nthreads: 5

#Run:

cwl-runner trimmomatic.cwl ../test/trimmomatic-test.yml

#### Alignement: BWA-mem - Phele

#Notes

* Using branch: <https://github.com/common-workflow-language/workflows/tree/new_bwa-mem>

#Manual run:  
bwa mem -M -p -t [num\_threads] \

-R "@RG\tID:1\tPL:ILLUMINA\tPU:pu\tLB:group1\tSM:SAMPLEID" \

#### [reference\_fasta] \

#### [input\_fastq] > [output]

#Test yml:

output\_filename: "align.bam"

reference:

class: File

path: /data\_small/small\_bwa/indexes/hg19/genome.fa.gz

reads:

- class: File

path: /data\_small/small\_fastq/project.NIST\_NIST7035\_H7AP8ADXX\_TAAGGCGA\_1\_NA12878.bwa.markDuplicates.chr22\_1.fq

reads:

- class: File

path: /data\_small/small\_fastq/project.NIST\_NIST7035\_H7AP8ADXX\_TAAGGCGA\_1\_NA12878.bwa.markDuplicates.chr22\_2.fq

threads: 5

#Run:

cwl-runner bwa-mem.cwl ../test/bwa-mem.yml

To avoid the error: [E::bwa\_set\_rg] the read group line is not started with @RG Set

[ReadGroup data]“@RG\tID:SRR622461\tPL:Illumina\tSM:NA12878\tPI:250\tCN:ILLUMINA\tPL:ILLUMINA\tDS:SRP000547"

If the RG info is missing replace with; "@RG\tID:1\tPL:ILLUMINA\tPU:pu\tLB:group1\tSM:SAMPLEID"

This value could be obtained using samtools view -H | grep “@RG”

#### Samtools Sort Bam Hocine & Yassine

samtools view -Shu in.bam | samtools sort -o -l 0 -@ 4 - \_sort > out.bam

Notes:

* This is using Yassing’s feature branch <https://github.com/common-workflow-language/workflows/blob/feature/feature-GATK-Yassine>

Test:

# get sample data

# test yml params for Azure host

{

"input": {

"class": "File",

"path": "/mnt/test-YS/NA12878.mapped.illumina.mosaik.CEU.exome.20110411.chr21.bam"

},

"compression\_level": 9,

"memory": "1G",

"threads": 4,

"output\_name": "NA12878.mapped.illumina.mosaik.CEU.exome.20110411.chr21.sorted.bam"

}

# run CWL

cwltool samtools-sort.cwl ../test/samtools-sort-job.json

#### Samtools Index Bam

Notes:

* This is using Yassine’s feature branch: <https://github.com/common-workflow-language/workflows/blob/feature/feature-GATK-Yassine>

Test:

# get sample data

/data\_small/small\_bam/project.NIST\_NIST7035\_H7AP8ADXX\_TAAGGCGA\_1\_NA12878.bwa.markDuplicates.chr22.markdup.bam

# test yml params for Azure host:

{

"input": {

"class": "File",

"path": "/data\_small/small\_bam/project.NIST\_NIST7035\_H7AP8ADXX\_TAAGGCGA\_1\_NA12878.bwa.markDuplicates.chr22.markdup.bam"

},

"bai": true

}

# run CWL

cwltool samtools-index.cwl ../test/samtools-index-job.json

#### Samtools View SAM

samtools view -Shu in.sam

Notes:

* This is using Yassing’s feature branch <https://github.com/common-workflow-language/workflows/blob/feature/feature-GATK-Yassine>

Test:

# get sample data

# test yml params for Azure host

{

"input": {

"class": "File",

"path": "/mnt/test-YS/NA12878.mapped.illumina.mosaik.CEU.exome.20110411.chr21.bam"

},

"compression\_level": 9,

"memory": "1G",

"threads": 4,

"output\_name": "NA12878.mapped.illumina.mosaik.CEU.exome.20110411.chr21.sorted.bam"

}

# run CWL

cwltool samtools-view.cwl ../test/samtools-view-job.json

#### MarkDuplicates: GATK Yassine & Brian

**Notes:**

* This is tested via CWL, see <https://github.com/common-workflow-language/workflows/pull/105>

**Testing:**

# original sample command

java -Djava.io.tmpdir=$tmpDir -Xmx8G \

-jar [picard\_dir]/MarkDuplicates.jar \

TMP\_DIR=$tmpDir \

OUTPUT=$tmpDir/out.bam \

METRICS\_FILE=$tmpDir/out.metrics \

ASSUME\_SORTED=True \

CREATE\_INDEX=True \

COMPRESSION\_LEVEL=0 \

MAX\_RECORDS\_IN\_RAM=1000000 \

VALIDATION\_STRINGENCY=SILENT \

VERBOSITY=INFO

# get sample data

/data\_small/small\_bam/project.NIST\_NIST7035\_H7AP8ADXX\_TAAGGCGA\_1\_NA12878.bwa.markDuplicates.chr22.bam

# test yml params for Azure host:

{

"inputFileName\_markDups": [{"class": "File", "path": "/data\_small/small\_bam/project.NIST\_NIST7035\_H7AP8ADXX\_TAAGGCGA\_1\_NA12878.bwa.markDuplicates.chr22.bam"}],

"outputFileName\_markDups": "/data\_small/small\_bam/project.NIST\_NIST7035\_H7AP8ADXX\_TAAGGCGA\_1\_NA12878.bwa.markDuplicates.chr22.markdup.bam",

"metricsFile": "metricsFile-markDuplicates",

"tmpdir": "test/test-files",

}

# docker image

scidap/picard:v1.141

# run manually

docker run -it scidap/picard:v1.141 /bin/bash

java -Xmx4g -jar /usr/local/bin/picard.jar MarkDuplicates INPUT=/data\_small/small\_bam/project.NIST\_NIST7035\_H7AP8ADXX\_TAAGGCGA\_1\_NA12878.bwa.markDuplicates.chr22.bam OUTPUT=/data\_small/small\_bam/project.NIST\_NIST7035\_H7AP8ADXX\_TAAGGCGA\_1\_NA12878.bwa.markDuplicates.chr22.markdup.bam METRICS\_FILE=metricsFile-markDuplicates CREATE\_INDEX=true TMP\_DIR=test/test-files

# run CWL

cwltool picard-MarkDuplicates.cwl ../test/picard-MarkDuplicates-job.json

#### IndelRealigner:

* Hocine, see <https://github.com/common-workflow-language/workflows/blob/feature/feature-GATK-Yassine>

# test yml params for Azure host:

{

"inputBam\_realign": {

"class": "File",

"path": "/data\_small/small\_bam/project.NIST\_NIST7035\_H7AP8ADXX\_TAAGGCGA\_1\_NA12878.bwa.markDuplicates.bam"

},

"reference": {

"class": "File",

"path": "/mnt/bwa-mem-index/hg19/genome.fa",

},

"intervals": {

"class": "File",

"path": "/mnt/gatk\_bundle/2.8/hg19/realignTargetCreator\_output.intervals",

},

"known": [

{"class": "File", "path": "/data\_small/ftp.broadinstitute.org/bundle/2.8/hg19/dbsnp\_138.hg19.vcf"},

{"class": "File", "path": "/data\_small/ftp.broadinstitute.org/bundle/2.8/hg19/Mills\_and\_1000G\_gold\_standard.indels.hg19.sites.vcf"},

{"class": "File", "path": "/data\_small/ftp.broadinstitute.org/bundle/2.8/hg19/1000G\_phase1.indels.hg19.sites.vcf"}

],

"outputfile\_indelRealigner": "indelRealigner.bam"

}

# run CWL

cwltool GATK-IndelRealigner.cwl ../test/GATK-IndelRealigner-job.json

#### RealignerTargetCreator

* Hocine, see <https://github.com/common-workflow-language/workflows/blob/feature/feature-GATK-Yassine>

**Testing:**

**# Command to run:**

java -Djava.io.tmpdir=$tmpDir -Xmx12G \

-jar [gatk.jar] \

-T RealignerTargetCreator \

-R [reference\_fasta] \

-o $tmpDir/[chrom].intervals \

--known [1kindel\_vcf] \

--known [mills\_vcf] \

--num\_threads 4 \

-L [chrom] [gatk\_realigntarget]

# get sample data

* /data\_small/small\_bam/
* /mnt/bwa-mem-index/hg19

# test yml params for Azure host:

{

"inputBam\_realign": {

"class": "File",

"path": "/data\_small/small\_bam/project.NIST\_NIST7035\_H7AP8ADXX\_TAAGGCGA\_1\_NA12878.bwa.markDuplicates.bam"

},

"reference": {

"class": "File",

"path": "/mnt/bwa-mem-index/hg19\_new/genome.fa",

},

"known": [

{"class": "File", "path": "/data\_small/ftp.broadinstitute.org/bundle/2.8/hg19/dbsnp\_138.hg19.vcf"},

{"class": "File", "path": "/data\_small/ftp.broadinstitute.org/bundle/2.8/hg19/Mills\_and\_1000G\_gold\_standard.indels.hg19.sites.vcf"},

{"class": "File", "path": "/data\_small/ftp.broadinstitute.org/bundle/2.8/hg19/1000G\_phase1.indels.hg19.sites.vcf"}

],

"outputfile\_realignTarget": "realignTargetCreator\_output.intervals"

}

# CWL command:

cwltool GATK-RealignTargetCreator.cwl ../test/GATK-RealignTargetCreator-job.json

#### BaseRecalibration and PrintReads (BQSR): GATK Yassine

java -Djava.io.tmpdir=$tmpDir -Xmx8G \

-jar [gatk.jar] \

-T BaseRecalibrator \

-R [reference\_fasta] \

-I [input.bam] \

-nct 4 \

-T BaseRecalibrator \

-R reference.fa \

-o recal\_data.table \

-knownSites [dbsnp\_vcf] \

-knownSites [1komni\_vcf] \

-knownSites [1kindel\_vcf] \

-knownSites [mills\_vcf]

**Notes:**

* This is testing <https://github.com/common-workflow-language/workflows/blob/feature/feature-GATK-Yassine>

**Testing:**

#Job file

"inputBam\_BaseRecalibrator": {

"class": "File",

"path": "/mnt/test-YS/NA12878.mapped.illumina.mosaik.CEU.exome.20110411.chr21.bam"

},

"reference": {

"class": "File",

"path": "/mnt/bwa-mem-index/hg19\_new/genome.fa",

},

"known": [ {"class": "File", "path": "/data\_small/ftp.broadinstitute.org/bundle/2.8/hg19/dbsnp\_138.hg19.vcf"},

{"class": "File", "path": "/data\_small/ftp.broadinstitute.org/bundle/2.8/hg19/Mills\_and\_1000G\_gold\_standard.indels.hg19.sites.vcf"},

{"class": "File", "path": "/data\_small/ftp.broadinstitute.org/bundle/2.8/hg19/1000G\_phase1.indels.hg19.sites.vcf"},

{"class": "File", "path": "/data\_small/ftp.broadinstitute.org/bundle/2.8/hg19/1000G\_omni2.5.hg19.sites.vcf"}

],

"outputfile\_BaseRecalibrator": "baseRecalibrator.table"

#cwl command

cwltool ../tools/GATK-BaseRecalibrator.cwl GATK-BaseRecalibrator-job.json

#### Haplotype Caller: GATK Yassine

java -jar GenomeAnalysisTK.jar \

-R reference.fasta \

-T HaplotypeCaller \

-I sample1.bam [-I sample2.bam ...] \

[--dbsnp dbSNP.vcf] \

-stand\_call\_conf 30 \

-stand\_emit\_conf 10 \

-o output.raw.snps.indels.vcf \

-nct 4

--variant\_index\_type LINEAR]

* This is on <https://github.com/common-workflow-language/workflows/blob/feature/feature-GATK-Yassine>

**Testing**

#Job file

{

"inputBam\_HaplotypeCaller": {

"class": "File",

"path": "/mnt/test-YS/samtools-sort-2016-08-04-wRG.bam"

},

"reference": {

"class": "File",

"path": "/mnt/bwa-mem-index/hg19/genome.fa",

},

"dbsnp": {

"class": "File",

"path": "/mnt/gatk\_bundle/2.8/hg19/dbsnp\_138.hg19.nochr.vcf",

},

"outputfile\_HaplotypeCaller": "test/test-files/haplotypeCaller.vcf"

}

#cwl command

cwltool --debug ../tools/GATK-HaplotypeCaller.cwl GATK-HaplotypeCaller-job.json

#### VariantRecalibrator & ApplyRecalibration (VQSR):

VQCR-SNP

#Notes:

* Using branch: <https://github.com/common-workflow-language/workflows/tree/feature/feature-GATK-Yassine>

#Manual run:

java -Djava.io.tmpdir=$tmpDir -Xmx8G \

-jar gatk.jar \

-T VariantRecalibrator \

-R reference.fasta \

-input raw\_variants.vcf \

-resource:hapmap,known=false,training=true,truth=true,prior=15.0 hapmap\_3.3.b37.sites.vcf \

-resource:omni,known=false,training=true,truth=false,prior=12.0 1000G\_omni2.5.b37.sites.vcf \

-resource:1000G,known=false,training=true,truth=false,prior=10.0 1000G\_phase1.snps.high\_confidence.vcf

-resource:dbsnp,known=true,training=false,truth=false,prior=2.0 dbsnp\_135.b37.vcf \

-an QD -an MQ -an MQRankSum -an ReadPosRankSum -an FS -an SOR -an InbreedingCoeff \

-mode SNP \

-recalFile output.recal \

-tranchesFile output.tranches \

-rscriptFile output.plots.R

#Test yml:

haplotypecaller\_snps\_vcf:

class: File

path: /data\_small/ftp.broadinstitute.org/bundle/2.8/hg19/NA12878.HiSeq.WGS.bwa.cleaned.raw.subset.hg19.vcf

multithreading\_nt:

reference:

class: File

path: /mnt/bwa-mem-index/hg19\_new/genome.fa

resource\_omni:

class: File

path: /data\_small/ftp.broadinstitute.org/bundle/2.8/hg19/1000G\_omni2.5.hg19.sites.vcf

resource\_dbsnp:

class: File

path: /data\_small/ftp.broadinstitute.org/bundle/2.8/hg19/dbsnp\_138.hg19.vcf

resource\_1kg:

class: File

path: /data\_small/ftp.broadinstitute.org/bundle/2.8/hg19/1000G\_phase1.snps.high\_confidence.hg19.sites.vcf

resource\_hapmap:

class: File

path: /data\_small/ftp.broadinstitute.org/bundle/2.8/hg19/hapmap\_3.3.hg19.sites.vcf

#Run command:

cwl-runner \

GATK-VariantRecalibrator-SNPs.cwl \

GATK-VariantRecalibrator-SNPs-test.yml

VQCR-IndelS

#Notes:

* Using branch: <https://github.com/common-workflow-language/workflows/tree/feature/feature-GATK-Yassine>

#Manual run: (Same as “SNP” run with difference in resource vcf files) -resource:mills,known=false,training=true,truth=true,prior=12.0 {s[mills\_vcf]} -resource:dbsnp,known=true,training=false,truth=false,prior=2.0 {s[dbsnp\_vcf]}

#Test yml:

haplotypecaller\_snps\_vcf:

class: File

path: /data\_small/ftp.broadinstitute.org/bundle/2.8/hg19/NA12878.HiSeq.WGS.bwa.cleaned.raw.subset.hg19.vcf

multithreading\_nt:

reference:

class: File

path: /mnt/bwa-mem-index/hg19\_new/genome.fa

resource\_mills:

class: File

path: /data\_small/ftp.broadinstitute.org/bundle/2.8/hg19/Mills\_and\_1000G\_gold\_standard.indels.hg19.sites.vcf

resource\_dbsnp:

class: File

path: /data\_small/ftp.broadinstitute.org/bundle/2.8/hg19/dbsnp\_138.hg19.excluding\_sites\_after\_129.vcf

#Run:

cwl-runner \

GATK-VariantRecalibrator-Indels.cwl \

GATK-VariantRecalibrator-Indels-test.yml

#### Apply VQSR: GATK Yassine

java -Djava.io.tmpdir=$tmpDir -Xmx8G \

4 -jar [gatk.jar] \

5 -T ApplyRecalibration \

i 6 -R [reference\_fasta] \

i 7 -recalFile $tmpDir/out.recal \

i 8 -tranchesFile $tmpDir/out.tranches \

o 9 -o $tmpDir/out.vcf \

10 --ts\_filter\_level 99.9 \

11 -mode [glm] \

12 -nt 4

### **Output: VCF (merged or not)**

# **Annotation**

### **ANNOVAR**

We won’t use this currently because of the license.

### **SNPEff**

**Notes:**

* Hocine, see his **feature branch** on: <https://github.com/common-workflow-language/workflows/blob/snpeff_hocine/tools/snpEff.cwl>
* I issued a pull request to Michael: <https://github.com/common-workflow-language/workflows/pull/101>

**Testing:**

# get sample data

wget ftp://[ftp-trace.ncbi.nlm.nih.gov/giab/ftp/data/NA12878/NA12878\_PacBio\_MtSinai/NA12878.sorted.vcf.gz](http://ftp-trace.ncbi.nlm.nih.gov/giab/ftp/data/NA12878/NA12878_PacBio_MtSinai/NA12878.sorted.vcf.gz)

# unzip this

# test yml params for Azure host:

genome: hg19

variant\_calling\_file:

class: File

path: "/data\_small/small\_vcf/NA12878.sorted.vcf"

nodownload: true

output\_format: vcf

data\_dir:

class: Directory

location: "/data\_small/snpEff\_hg19/data"

# build it

docker build -t quay.io/snpeff:4.3 -f snpEff\_Dockerfile .

# run manually

snpEff ann -dataDir /data\_small/snpEff\_hg19/data hg19 /data\_small/small\_vcf/NA12878.sorted.vcf

# run CWL

../env/bin/cwltool --debug snpEff.cwl ../test/snpEff-job.yml

# **QC**

### **FastQC:**

* Hocine, see <https://github.com/common-workflow-language/workflows/blob/master/tools/fastqc.cwl>
* The file can be found in the **master** branch

### **Testing:**

# fastq file is in /data\_small/fastq\_small directory

# run manually

docker run -it -v /data\_small/small\_fastq:/data genomicpariscentre/fastqc --outdir . /data/project.NIST\_NIST7035\_H7AP8ADXX\_TAAGGCGA\_1\_NA12878.bwa.markDuplicates.chr22\_1.fq

# test yml for Azure host

fastqFile:

class: File

path: /data\_small/small\_fastq/project.NIST\_NIST7035\_H7AP8ADXX\_TAAGGCGA\_1\_NA12878.bwa.markDuplicates.chr22\_1.fq

# test run CWL

cwltool --debug fastqc.cwl ../test/fastqc-job.cwl |& less

**Note:** fastqc.cwl import fastqc\_dockerfile.yml which pull a fastqc image.

### **DepthOfCoverage (GATK):**

**Notes:**

* Brian, see <https://github.com/common-workflow-language/workflows/blob/feature/feature-GATK-DoC-Brian/tools/GATK-DepthOfCoverage.cwl>
* I created a feature branch for Michael to review: <https://github.com/common-workflow-language/workflows/pull/107>

**Testing:**

# get data, this is not in the /data\_small directory

wget ftp://ftp-trace.ncbi.nlm.nih.gov/giab/ftp/data/NA12878/Garvan\_NA12878\_HG001\_HiSeq\_Exome/project.NIST\_NIST7035\_H7AP8ADXX\_TAAGGCGA\_1\_NA12878.bwa.markDuplicates.bai

wget ftp://ftp-trace.ncbi.nlm.nih.gov/giab/ftp/data/NA12878/Garvan\_NA12878\_HG001\_HiSeq\_Exome/project.NIST\_NIST7035\_H7AP8ADXX\_TAAGGCGA\_1\_NA12878.bwa.markDuplicates.bam

# run manually

docker run -it -v `pwd`:/root/data scidap/gatk:v3.5 /bin/bash

/usr/local/bin/GenomeAnalysisTK -nt 16 --omitIntervalStatistics --omitDepthOutputAtEachBase -T DepthOfCoverage -R ftp.broadinstitute.org/bundle/2.8/hg19/ucsc.hg19.fasta -o report -I small\_bam/project.NIST\_NIST7035\_H7AP8ADXX\_TAAGGCGA\_1\_NA12878.bwa.markDuplicates.bam

# test json for Azure host

{

"threads" : 16,

"omitIntervalStatistics" : true,

"omitDepthOutputAtEachBase" : true,

"inputBam\_DepthOfCoverage": {

"class": "File",

"path": "/data\_small/small\_bam/project.NIST\_NIST7035\_H7AP8ADXX\_TAAGGCGA\_1\_NA12878.bwa.markDuplicates.bam"

},

"reference": {

"class": "File",

"path": "/data\_small/ftp.broadinstitute.org/bundle/2.8/hg19/ucsc.hg19.fasta",

},

"outputfile\_DepthOfCoverage": "sample\_report"

}

# test run CWL

git pull; cwltool --debug GATK-DepthOfCoverage.cwl ../test/GATK-DepthOfCoverage-job.json

### **BAMStats:**

**Notes:**

* Brian (stream C is doing this)
* Hocine is testing this <https://github.com/briandoconnor/dockstore-tool-bamstats> for both the Dockerfile and CWL
* The git repo is: <https://github.com/briandoconnor/dockstore-tool-bamstats>
* The quay repo is: <https://quay.io/repository/briandoconnor/dockstore-tool-bamstats?tab=tags>

**Testing:**

# Testing on Azure host.

# get data, this is not in the /data\_small directory

# test json for Azure host

{

"bam\_input": {

"class": "File",

"format": "http://edamontology.org/format\_2572",

"path": "/data\_small/small\_bam/project.NIST\_NIST7035\_H7AP8ADXX\_TAAGGCGA\_1\_NA12878.bwa.markDuplicates.chr22.bam"

},

"bamstats\_report": {

"class": "File",

"format": "http://edamontology.org/format\_3615",

"path": "/tmp/bamstats\_report.zip"

}

}

# test run CWL

cwltool Dockstore.cwl sample\_configs.local.2.json

# **Code Location**

Let’s use the CWL workflows repo, each making a “feature branch” for our respective changes which can be used for a “pull request” back to develop. This will let us individually work on our individual tools.

<https://github.com/common-workflow-language/workflows>

# **Reference Data**

location: ftp.broadinstitute.org/bundle

Build: hg19

username: gsapubftp-anonymous

Password:

You can use a URL in Chrome like: ftp://gsapubftp-anonymous:@[ftp.broadinstitute.org/bundle/2.8/hg19/](http://ftp.broadinstitute.org/bundle/2.8/hg19/)

SNPEff:

wget '<http://downloads.sourceforge.net/project/snpeff/databases/v4_3/snpEff_v4_3_hg19.zip>'

# **Sample Workflows/Tools**

See <https://github.com/common-workflow-language/workflows>

Pipeline tools (commands): <https://github.com/LPM-HMS/GenomeKey/blob/master/genomekey/tools/pipes.py>

Pipeline def: <https://github.com/LPM-HMS/GenomeKey/blob/master/genomekey/workflows/pipeline.py>

I’m putting together a listing of CWL workflows/tools and Dockerfiles/Docker images here: <https://github.com/ga4gh/dockstore/wiki/Potential-Sources-of-CWL-Workflows-and-Tools>

# **Sample Data**

* NIST GIAB comparison script: <https://gist.github.com/cc2qe/6163ce5c30ba17a4df5d>
* GIAB NA12878 fastq Exome: ftp://[ftp-trace.ncbi.nlm.nih.gov/giab/ftp/data/NA12878/Garvan\_NA12878\_HG001\_HiSeq\_Exome/](http://ftp-trace.ncbi.nlm.nih.gov/giab/ftp/data/NA12878/Garvan_NA12878_HG001_HiSeq_Exome/)
* NA12878 sequence:<http://www.ncbi.nlm.nih.gov/sra/?term=SRX1049768>
* Also see the paper:<http://jimb.stanford.edu/giab-resources>
* Sorted VCF file: ftp://[ftp-trace.ncbi.nlm.nih.gov/giab/ftp/data/NA12878/NA12878\_PacBio\_MtSinai/](http://ftp-trace.ncbi.nlm.nih.gov/giab/ftp/data/NA12878/NA12878_PacBio_MtSinai/)