

# **Genomes On The Cloud**

## **GotCloud**

University of Michigan  
Center for Statistical Genetics  
Mary Kate Wing  
Goo Jun

# Why GotCloud?

- Connects sequence analysis tools together
  - Alignment, quality control, variant calling
  - So you don't have to manually run/configure each one
- Divides large jobs into many small pieces
  - Simplifies running on clusters
  - Re-start after failure
- Can analyze many samples together
- Can run on Amazon Cloud or on your own clusters

# GotCloud Initial Setup

- Download & build gotcloud
  - <http://genome.sph.umich.edu/wiki/GotCloud>
- Download Reference Files
  - Will be available on the wiki (next week)
- Get your data files
  - From where you get your data
- Tutorial data is also available on the wiki
  - [http://genome.sph.umich.edu/wiki/Tutorial:\\_GotCloud](http://genome.sph.umich.edu/wiki/Tutorial:_GotCloud)
- Join the mailing list:
  - <http://groups.google.com/group/GotCloud>

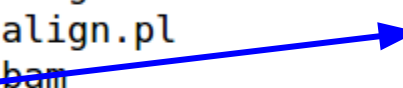
# What was preinstalled for GotCloud Tutorial?

- GotCloud Programs: /usr/local/bin/gotcloud/
  - gotcloud script
  - subscripts
  - programs & tools called by the script
- Test data files: /opt/gotcloudExample/
  - Reference Files
  - FASTQ files
  - BAM files
  - GotCloud index files
  - GotCloud configuration files

# /usr/local/gotcloud/

```
workshop:~> ls -l /usr/local/gotcloud/
total 16
drwxr-xr-x 2 root root 4096 Mar  1 18:30 bin
-rw-r--r-- 1 root root   5 Feb 28 13:18 release_version.txt
drwxr-xr-x 2 root root 4096 Mar  1 18:30 scripts
drwxr-xr-x 5 root root 4096 Mar  1 18:30 test
workshop:~>
```

```
workshop:~> ls /usr/local/gotcloud/bin
alignDefaults.conf  glfMultiples      samtools-hybrid
align.pl            gotcloud          tabix
bam                 infoCollector     thunderVCF
beagle.20101226.jar make_indexfile.pl umakeDefaults.conf
bgzip               MergeSamFiles.jar umake.pl
bwa                 Multi.pm          vcfCooker
glfExtract          qplot             vcfPileup
glfMerge            samtools          verifyBamID
workshop:~>
```



Only tool  
you need  
to call  
yourself

```
workshop:~> ls /usr/local/gotcloud/scripts/
bams2vcfMerge.pl      glfAlias2Ped.pl    vcf2Beagle.pl
beagle2Vcf.pl         ligateVcf.pl       vcfPaste.pl
check_requirements.sh phasedVcf2Beagle.pl vcfSplit.pl
diff_results_align.sh runcluster.pl       vcf-summary
diff_results_umake.sh run_svm.pl
filt_svm.pl           subsetVcf.pl
workshop:~>
```

# /opt/gotcloudExample/

```
workshop:~> ls -l /opt/gotcloudExample/
total 36
drwxrwxr-x 2 root root 4096 Feb 25 09:30 bams
drwxrwxr-x 2 root root 4096 Feb 25 09:17 chr20Ref
drwxrwxr-x 2 root root 4096 Feb 21 14:28 fastq
-rw-r--r-- 1 root root 272 Mar 6 13:51 GBR2align.conf
-rw-r--r-- 1 root root 1388 Mar 6 13:51 GBR2fastq.index
-rw-rw-r-- 1 root root 1740 Feb 21 09:37 GBR60bam.index
-rw-rw-r-- 1 root root 658 Mar 5 16:36 GBR60vc.conf
-rw-r--r-- 1 root root 10 Feb 22 08:16 test.GBR60.dat
-rw-r--r-- 1 root root 2336 Feb 22 08:16 test.GBR60.ped
workshop:~>
```

BAM files for Variant Calling

Chr20 Reference Files

FASTQ files for Alignment

User Generated Input Files

# Reference Files

- Available for Download on wiki
- 4 Types of Reference Files

Indel VCF file

DBSNP VCF  
file & index

HAPMAP  
VCF file &  
index

```
workshop:/opt/gotcloudExample/chr20Ref> ls
1kg.pilot_release.merged.indels.sites.hg19.chr20.vcf
dbsnp135_chr20.vcf.gz
dbsnp135_chr20.vcf.gz.tbi
hapmap_3.3.b37.sites.chr20.vcf.gz
hapmap_3.3.b37.sites.chr20.vcf.gz.tbi
human_g1k_v37_chr20-bs.umfa
human_g1k_v37_chr20.dict
human_g1k_v37_chr20.fa
human_g1k_v37_chr20.fa.amb
human_g1k_v37_chr20.fa.ann
human_g1k_v37_chr20.fa.bwt
human_g1k_v37_chr20.fa.fai
human_g1k_v37_chr20.fa.GCcontent
human_g1k_v37_chr20.fa.pac
human_g1k_v37_chr20.fa.rbwt
human_g1k_v37_chr20.fa.rpac
human_g1k_v37_chr20.fa.rsa
human_g1k_v37_chr20.fa.sa
workshop:/opt/gotcloudExample/chr20Ref>
```

Genome Reference  
Files

various formats required  
for various steps

Variant Calling only needs:  
.fa, -bs.umfa, .fa.fai

Rest are Aligner only





# Alignment Pipeline Inputs

- FASTQs -> BAMs
- Required Inputs for any Alignment Pipeline
  - Reference Files
  - FASTQ files
    - provided to you by someone
- User Generated Inputs required for GotCloud
  - Index file of FASTQs
    - Points to your FASTQ files
  - Configuration File
    - Points to your FASTQ index file
    - Points to your reference files
    - User specific configuration

# Tutorial FASTQ Data Files

```
workshop:~> ls /opt/gotcloudExample/fastq/  
HG00096_SRR062634_1.fastq  HG00096_SRR062641.fastq  
HG00096_SRR062634_2.fastq  HG00100_ERR013140_1.fastq  
HG00096_SRR062634.fastq    HG00100_ERR013140_2.fastq  
HG00096_SRR062635_1.fastq  HG00100_ERR013140.fastq  
HG00096_SRR062635_2.fastq  HG00100_ERR016352_1.fastq  
HG00096_SRR062635.fastq    HG00100_ERR016352_2.fastq  
HG00096_SRR062641_1.fastq  HG00100_ERR016352.fastq  
HG00096_SRR062641_2.fastq  
workshop:~> █
```



# FastQ Index File

- Points gotcloud to run's FASTQs files
- Associates additional information with the FASTQs
- Required Header Line (tab separated)
  - MERGE\_NAME (required)
    - base filename name for resulting sample-level BAM (typically just the sample name)
    - groups multiple lines/fastqs for a sample
  - FASTQ1 (required)
    - name of fastq or first in the pair
  - FASTQ2 (optional)
    - name of the 2nd fastq in paired-end
    - '.' if single-end

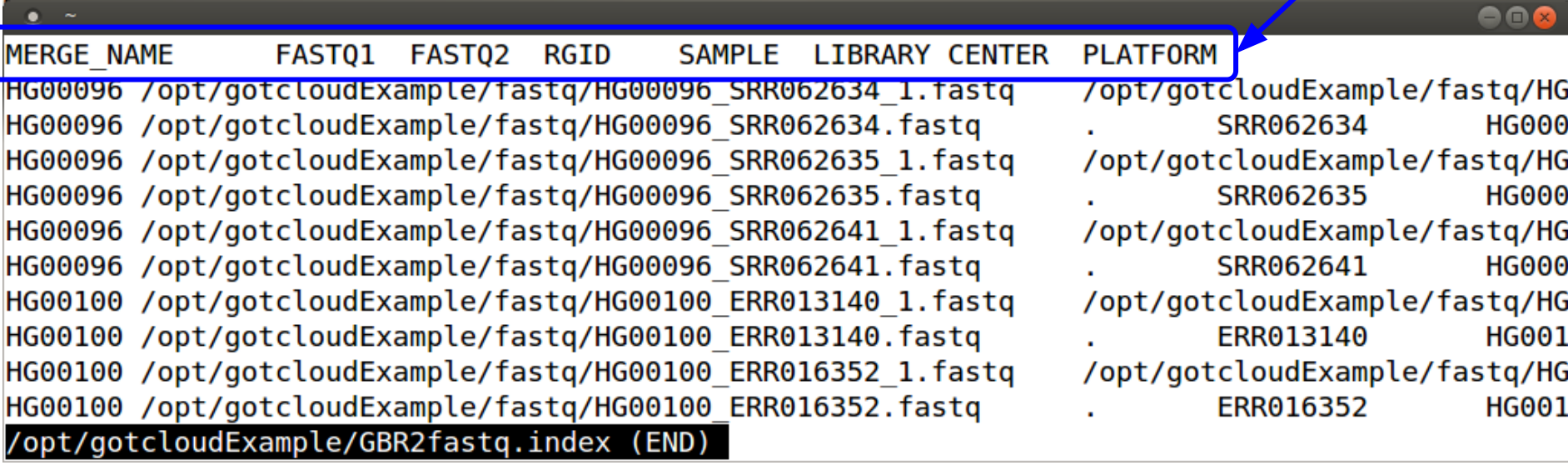
# FastQ Index File

- Additional Optional Column Names are for populating the Read Group Information in the BAM file (Leave out column or use "." if N/A)
  - RGID - Read Group ID
  - SAMPLE - Sample Name
  - LIBRARY - Library
  - CENTER - Center Name
  - PLATFORM - Platform
- RGID & Library are used in future processing as some logic is specific to an RG or Library
- If you don't know what to put, just leave out or assign everything the same RGID/LIBRARY

# Tutorial FastQ Index File

- `less -S /opt/gotcloudExample/GBR2fastq.index`

Header Line



```
MERGE_NAME    FASTQ1    FASTQ2    RGID    SAMPLE    LIBRARY    CENTER    PLATFORM
HG00096 /opt/gotcloudExample/fastq/HG00096_SRR062634_1.fastq /opt/gotcloudExample/fastq/HG
HG00096 /opt/gotcloudExample/fastq/HG00096_SRR062634.fastq . SRR062634 HG000
HG00096 /opt/gotcloudExample/fastq/HG00096_SRR062635_1.fastq /opt/gotcloudExample/fastq/HG
HG00096 /opt/gotcloudExample/fastq/HG00096_SRR062635.fastq . SRR062635 HG000
HG00096 /opt/gotcloudExample/fastq/HG00096_SRR062641_1.fastq /opt/gotcloudExample/fastq/HG
HG00096 /opt/gotcloudExample/fastq/HG00096_SRR062641.fastq . SRR062641 HG000
HG00100 /opt/gotcloudExample/fastq/HG00100_ERR013140_1.fastq /opt/gotcloudExample/fastq/HG
HG00100 /opt/gotcloudExample/fastq/HG00100_ERR013140.fastq . ERR013140 HG001
HG00100 /opt/gotcloudExample/fastq/HG00100_ERR016352_1.fastq /opt/gotcloudExample/fastq/HG
HG00100 /opt/gotcloudExample/fastq/HG00100_ERR016352.fastq . ERR016352 HG001
/opt/gotcloudExample/GBR2fastq.index (END)
```

# Tutorial FastQ Index File

- `less -S /opt/gotcloudExample/GBR2fastq.index`

```
MERGE_NAME      FASTQ1 FASTQ2  RGID   SAMPLE  LIBRARY CENTER  PLATFORM
HG00096 /opt/gotcloudExample/fastq/HG00096_SRR062634_1.fastq /opt/gotcloudExample/fastq/HG
HG00096 /opt/gotcloudExample/fastq/HG00096_SRR062634.fastq . SRR062634 HG000
HG00096 /opt/gotcloudExample/fastq/HG00096_SRR062635_1.fastq /opt/gotcloudExample/fastq/HG
HG00096 /opt/gotcloudExample/fastq/HG00096_SRR062635.fastq . SRR062635 HG000
HG00096 /opt/gotcloudExample/fastq/HG00096_SRR062641_1.fastq /opt/gotcloudExample/fastq/HG
HG00096 /opt/gotcloudExample/fastq/HG00096_SRR062641.fastq . SRR062641 HG000
HG00100 /opt/gotcloudExample/fastq/HG00100_ERR013140_1.fastq /opt/gotcloudExample/fastq/HG
HG00100 /opt/gotcloudExample/fastq/HG00100_ERR013140.fastq . ERR013140 HG001
HG00100 /opt/gotcloudExample/fastq/HG00100_ERR016352_1.fastq /opt/gotcloudExample/fastq/HG
HG00100 /opt/gotcloudExample/fastq/HG00100_ERR016352.fastq . ERR016352 HG001
/opt/gotcloudExample/GBR2fastq.index (END)
```

Sample Name used for MERGE\_NAME. We will end up with 2 BAMs: HG00096.bam & HG00100.bam

# Tutorial FastQ Index File

- `less -S /opt/gotcloudExample/GBR2fastq.index`

```
MERGE_NAME      FASTQ1 FASTQ2  RGID      SAMPLE  LIBRARY CENTER  PLATFORM
HG00096 /opt/gotcloudExample/fastq/HG00096_SRR062634_1.fastq /opt/gotcloudExample/fastq/HG
HG00096 /opt/gotcloudExample/fastq/HG00096_SRR062634.fastq . SRR062634 HG000
HG00096 /opt/gotcloudExample/fastq/HG00096_SRR062635_1.fastq /opt/gotcloudExample/fastq/HG
HG00096 /opt/gotcloudExample/fastq/HG00096_SRR062635.fastq . SRR062635 HG000
HG00096 /opt/gotcloudExample/fastq/HG00096_SRR062641_1.fastq /opt/gotcloudExample/fastq/HG
HG00096 /opt/gotcloudExample/fastq/HG00096_SRR062641.fastq . SRR062641 HG000
HG00100 /opt/gotcloudExample/fastq/HG00100_ERR013140_1.fastq /opt/gotcloudExample/fastq/HG
HG00100 /opt/gotcloudExample/fastq/HG00100_ERR013140.fastq . ERR013140 HG001
HG00100 /opt/gotcloudExample/fastq/HG00100_ERR016352_1.fastq /opt/gotcloudExample/fastq/HG
HG00100 /opt/gotcloudExample/fastq/HG00100_ERR016352.fastq . ERR016352 HG001
/opt/gotcloudExample/GBR2fastq.index (END)
```

FASTQs - first in paired-end and the only in single-end  
Full path tells GotCloud where to find them



# Tutorial FastQ Index File

- `less -S /opt/gotcloudExample/GBR2fastq.index`

```
MERGE_NAME      FASTQ1 FASTQ2  RGID      SAMPLE  LIBRARY CENTER  PLATFORM
HG00096 /opt/gotcloudExample/fastq/HG00096_SRR062634_1.fastq /opt/gotcloudExample/fastq/HG
HG00096 /opt/gotcloudExample/fastq/HG00096_SRR062634.fastq . SRR062634 HG000
HG00096 /opt/gotcloudExample/fastq/HG00096_SRR062635_1.fastq /opt/gotcloudExample/fastq/HG
HG00096 /opt/gotcloudExample/fastq/HG00096_SRR062635.fastq . SRR062635 HG000
HG00096 /opt/gotcloudExample/fastq/HG00096_SRR062641_1.fastq /opt/gotcloudExample/fastq/HG
HG00096 /opt/gotcloudExample/fastq/HG00096_SRR062641.fastq . SRR062641 HG000
HG00100 /opt/gotcloudExample/fastq/HG00100_ERR013140_1.fastq /opt/gotcloudExample/fastq/HG
HG00100 /opt/gotcloudExample/fastq/HG00100_ERR013140.fastq . ERR013140 HG001
HG00100 /opt/gotcloudExample/fastq/HG00100_ERR016352_1.fastq /opt/gotcloudExample/fastq/HG
HG00100 /opt/gotcloudExample/fastq/HG00100_ERR016352.fastq . ERR016352 HG001
/opt/gotcloudExample/GBR2fastq.index (END)
```

FASTQ2 - "." for single-end and the path to the 2nd in a pair

# FastQ Index File

Read Group ID  
(RGID)

LIBRARY

PLATFORM

```
G00096_SRR062634_2.fastq  SRR062634  HG00096  2845856850  WUGSC  ILLUMINA
096 2845856850      WUGSC      ILLUMINA
G00096_SRR062635_2.fastq  SRR062635  HG00096  2845856850  WUGSC  ILLUMINA
096 2845856850      WUGSC      ILLUMINA
G00096_SRR062641_2.fastq  SRR062641  HG00096  2845856850  WUGSC  ILLUMINA
096 2845856850      WUGSC      ILLUMINA
G00100_ERR013140_2.fastq  ERR013140  HG00100  g1k-sc-HG00100-A  SC      ILLUMINA
100 g1k-sc-HG00100-A  SC      ILLUMINA
G00100_ERR016352_2.fastq  ERR016352  HG00100  g1k-sc-HG00100-A  SC      ILLUMINA
100 g1k-sc-HG00100-A  SC      ILLUMINA
(END)
```

The table displays the structure of a FastQ index file. Each line represents a read group. The first column contains the filename. The second column is the Read Group ID (RGID), highlighted with a blue box. The third and fourth columns are the LIBRARY (HG00096) and the read ID (2845856850), both highlighted with pink boxes. The fifth and sixth columns are the CENTER (WUGSC) and the PLATFORM (ILLUMINA), highlighted with green and orange boxes respectively. The last two columns show the sequencing center (SC) and the platform (ILLUMINA) for reads from a different center (G00100). Arrows point from the labels above to the corresponding fields in the table.

SAMPLE

CENTER

Primarily only useful if  
data comes from multiple  
centers

# Configuration File

- Tells GotCloud about your run
- Specifies:
  - FASTQ Index File to use
  - Reference Files to use
  - any other setting overrides
- Uses "KEY = VALUE" pairs for specifying information
- \$(KEY) can be used as VALUE in other lines:
  - KEY1 = value1
  - KEY2 = \$(KEY1)/value2
    - gets translated to: KEY2 = value1/value2
- # indicates a comment

# Alignment Pipeline Configuration File

Path to FASTQ Index file

```
workshop:~> head /opt/gotcloudExample/GBR2align.conf
INDEX_FILE = /opt/gotcloudExample/GBR2fastq.index
#####
# References
REF_DIR = /opt/gotcloudExample/chr20Ref
AS = NCBI37
FA_REF = $(REF_DIR)/human_g1k_v37_chr20.fa
DBSNP_VCF = $(REF_DIR)/dbsnp135_chr20.vcf.gz
HM3_VCF = $(REF_DIR)/hapmap_3.3.b37.sites.chr20.vcf.gz
workshop:~> █
```

Reference Files

KEY Substitution for reference directory

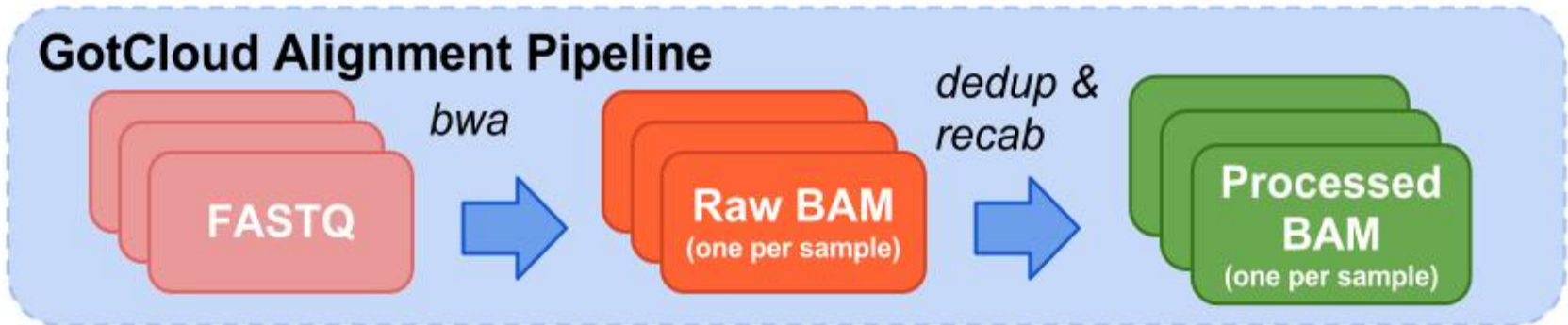
FA\_REF file ends in .fa extension, but this is used to also look for all of the other .fa... files

# Ready to Run Alignment Pipeline

- Type the following in your terminal to run:

```
gotcloud align --conf /opt/gotcloudExample/GBR60align.conf --outdir ~/gotcloudTutorial
```

- This runs the whole alignment pipeline



# Variant Calling Inputs

- BAMs -> VCFs
- Required Inputs for any Variant Calling
  - Reference Files
  - BAM files, either:
    - output of Alignment Pipeline
    - provided to you from someone
- User Generated Inputs required for GotCloud
  - Index file of BAMs
    - Points to your BAM files
  - Configuration File
    - Points to your BAM index file
    - Points to your reference files
    - User specific configuration

# Variant Calling BAM Files

```
workshop:~> ls /opt/gotcloudExample/bams
HG00096.bam      HG00119.bam      HG00138.bam      HG00152.bam      HG00242.bam
HG00096.bam.bai  HG00119.bam.bai  HG00138.bam.bai  HG00152.bam.bai  HG00242.bam.bai
HG00100.bam      HG00120.bam      HG00139.bam      HG00154.bam      HG00243.bam
HG00100.bam.bai  HG00120.bam.bai  HG00139.bam.bai  HG00154.bam.bai  HG00243.bam.bai
HG00103.bam      HG00122.bam      HG00140.bam      HG00155.bam      HG00244.bam
HG00103.bam.bai  HG00122.bam.bai  HG00140.bam.bai  HG00155.bam.bai  HG00244.bam.bai
HG00106.bam      HG00123.bam      HG00141.bam      HG00156.bam      HG00245.bam
HG00106.bam.bai  HG00123.bam.bai  HG00141.bam.bai  HG00156.bam.bai  HG00245.bam.bai
HG00108.bam      HG00124.bam      HG00142.bam      HG00157.bam      HG00246.bam
HG00108.bam.bai  HG00124.bam.bai  HG00142.bam.bai  HG00157.bam.bai  HG00246.bam.bai
HG00111.bam      HG00125.bam      HG00143.bam      HG00158.bam      HG00247.bam
HG00111.bam.bai  HG00125.bam.bai  HG00143.bam.bai  HG00158.bam.bai  HG00247.bam.bai
HG00112.bam      HG00126.bam      HG00144.bam      HG00159.bam      HG00249.bam
HG00112.bam.bai  HG00126.bam.bai  HG00144.bam.bai  HG00159.bam.bai  HG00249.bam.bai
HG00114.bam      HG00127.bam      HG00145.bam      HG00160.bam      HG00250.bam
HG00114.bam.bai  HG00127.bam.bai  HG00145.bam.bai  HG00160.bam.bai  HG00250.bam.bai
HG00115.bam      HG00131.bam      HG00146.bam      HG00231.bam      HG00251.bam
HG00115.bam.bai  HG00131.bam.bai  HG00146.bam.bai  HG00231.bam.bai  HG00251.bam.bai
HG00116.bam      HG00133.bam      HG00147.bam      HG00232.bam      HG00252.bam
HG00116.bam.bai  HG00133.bam.bai  HG00147.bam.bai  HG00232.bam.bai  HG00252.bam.bai
HG00117.bam      HG00136.bam      HG00148.bam      HG00233.bam      HG00253.bam
HG00117.bam.bai  HG00136.bam.bai  HG00148.bam.bai  HG00233.bam.bai  HG00253.bam.bai
HG00118.bam      HG00137.bam      HG00149.bam      HG00239.bam      HG00254.bam
HG00118.bam.bai  HG00137.bam.bai  HG00149.bam.bai  HG00239.bam.bai  HG00254.bam.bai
workshop:~>
```

Each Sample has:

- 1 .bam file
- 1 .bam.bai (index) file


# Telling GotCloud About Your BAMs

- BAM Index File
  - 1 line per sample
- Columns (tab separated):
  1. Sample ID
  2. Comma separated list of population labels
    - Typically N/A, so specify ALL
  3. path to BAM File 1
  - ...
  - N. path to BAM File N (if applicable)
    - If you have more than one BAM file per Sample



# BAM Index :

## Tells GotCloud About Your BAMs



```
workshop:~> head /opt/gotcloudExample/GBR60bam.index
HG00096 GBR      bams/HG00096.bam
HG00100 GBR      bams/HG00100.bam
HG00103 GBR      bams/HG00103.bam
HG00106 GBR      bams/HG00106.bam
HG00108 GBR      bams/HG00108.bam
HG00111 GBR      bams/HG00111.bam
HG00112 GBR      bams/HG00112.bam
HG00114 GBR      bams/HG00114.bam
HG00115 GBR      bams/HG00115.bam
HG00116 GBR      bams/HG00116.bam
workshop:~> █
```

- 1st Column -> Sample Names

# BAM Index :

## Tells GotCloud About Your BAMs

```
workshop:~> head /opt/gotcloudExample/GBR60bam.index
HG00096 GBR bam/HG00096_bam
HG00100 GBR ba
HG00103 GBR ba
HG00106 GBR ba
HG00108 GBR ba
HG00111 GBR ba
HG00112 GBR ba
HG00114 GBR ba
HG00115 GBR ba
HG00116 GBR bam
workshop:~>
```

### Population

- Used in Idrefine step to split into population specific VCFs
- If population is N/A:
  - set this to "ALL"
  - used in directory/file names

- 2nd Column -> Population

# BAM Index :

## Tells GotCloud About Your BAMs

```
workshop:~> head /opt/gotcloudExample
HG00096 GBR bams/HG00096.bam
HG00100 GBR bams/HG00100.bam
HG00103 GBR bams/HG00103.bam
HG00106 GBR bams/HG00106.bam
HG00108 GBR bams/HG00108.bam
HG00111 GBR bams/HG00111.bam
HG00112 GBR bams/HG00112.bam
HG00114 GBR bams/HG00114.bam
HG00115 GBR bams/HG00115.bam
HG00116 GBR bams/HG00116.bam
workshop:~>
```

### Paths to BAM Files

- Tutorial only has 1 BAM per sample
- Relative to index file location
- Can use full path

- 3rd - Nth Columns -> path to BAMs

# Creating BAM Index File For Your Own Data

- 1 row for each Sample you are processing
  - Enter Sample Name as 1st column
- Identify populations per sample
  - ALL, unless you are running multiple populations that need to be analyzed separately (like for 1000Genomes project)
- Enter all BAM files for the sample
  - Each in their own column, typically just 1 per sample
- Reminders:
  - Columns are separated by tabs
  - Populations are separated by ","s

# Configuration File

- Tells GotCloud about your run
- Specifies:
  - BAM Index File to use
  - Reference Files to use
  - Specifies Chromosomes to Analyze
  - any other setting overrides
- Uses "KEY = VALUE" pairs for specifying information
- \$(KEY) can be used as VALUE in other lines:
  - KEY1 = value1
  - KEY2 = \$(KEY1)/value2
    - gets translated to: KEY2 = value1/value2
- # indicates a comment

# Tutorial Configuration File

```
workshop:~> cat /opt/gotcloudExample/GBR60vc.conf
CHRS = 20
BAM_INDEX = /opt/gotcloudExample/GBR60bam.index
#####
# References
REF_ROOT = /opt/gotcloudExample/chr20Ref
#
REF = $(REF_ROOT)/human_g1k_v37_chr20.fa
INDEL_PREFIX = $(REF_ROOT)/1kg.pilot_release.merged.indels.sites.hg19
DBSNP_VCF = $(REF_ROOT)/dbsnp135_chr20.vcf.gz
HM3_VCF = $(REF_ROOT)/hapmap_3.3.b37.sites.chr20.vcf.gz

# Update thunder options so it will run faster for the tutorial:
# * run with only 10 rounds instead of 30 (-r option)
# * run without the --compact option
# This will use more memory, but is not an issue for this small sample data.
THUNDER = $(UMAKE_ROOT)/bin/thunderVCF -r 10 --phase --dosage --inputPhased

workshop:~> █
```

# Tutorial Configuration File

```
workshop:~> cat /opt/gotcloudExample/GBR60vc.conf
```

```
CHRS = 20
```

```
BAM_INDEX = /opt/gotclo
```

```
#####
```

```
# References
```

```
REF_ROOT = /opt/gotclou
```

```
#
```

```
REF = $(REF_ROOT)/human
```

```
INDEL_PREFIX = $(REF_RO
```

```
DBSNP_VCF = $(REF_ROOT
```

```
HM3_VCF = $(REF_ROOT)/
```

```
# Update thunder options so it will run faster for the tutorial:
```

```
# * run with only 10 rounds instead of 30 (-r option)
```

```
# * run without the --compact option
```

```
# This will use more memory, but is not an issue for this small sample data.
```

```
THUNDER = $(UMAKE_ROOT)/bin/thunderVCF -r 10 --phase --dosage --inputPhased
```

```
workshop:~> █
```

Space separated list of chromosomes

In this case: only chromosome 20

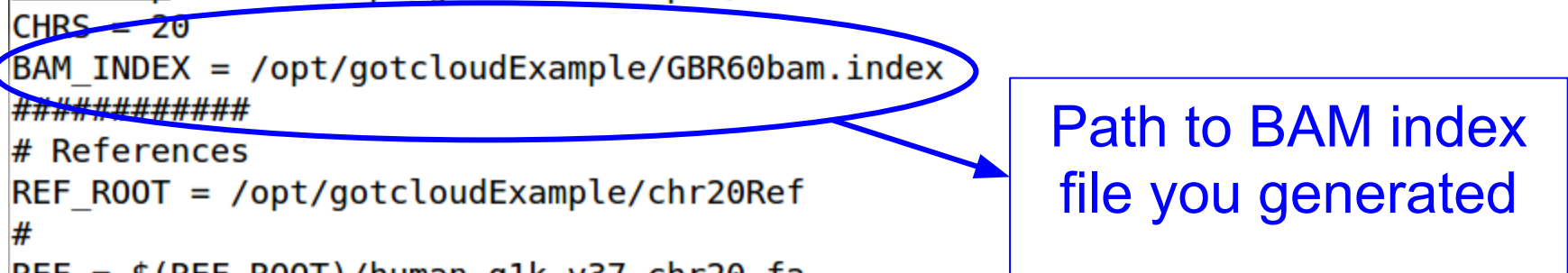
Remove this line to run the default of  
chr 1-22, X, Y

# Tutorial Configuration File

```
workshop:~> cat /opt/gotcloudExample/GBR60vc.conf
CHRS = 20
BAM_INDEX = /opt/gotcloudExample/GBR60bam.index
#####
# References
REF_ROOT = /opt/gotcloudExample/chr20Ref
#
REF = $(REF_ROOT)/human_g1k_v37_chr20.fa
INDEL_PREFIX = $(REF_ROOT)/1kg.pilot_release.merged.indels.sites.hg19
DBSNP_VCF = $(REF_ROOT)/dbsnp135_chr20.vcf.gz
HM3_VCF = $(REF_ROOT)/hapmap_3.3.b37.sites.chr20.vcf.gz

# Update thunder options so it will run faster for the tutorial:
# * run with only 10 rounds instead of 30 (-r option)
# * run without the --compact option
# This will use more memory, but is not an issue for this small sample data.
THUNDER = $(UMAKE_ROOT)/bin/thunderVCF -r 10 --phase --dosage --inputPhased

workshop:~> █
```



Path to BAM index file you generated



# Tutorial Configuration File

```
workshop:~> cat /opt/gotcloudExample/GBR60vc.conf
CHRS = 20
BAM_INDEX = /opt/gotcloudExample/GBR60bam.index
#####
# References
REF_ROOT = /opt/gotcloudExample/chr20Ref
#
REF = $(REF_ROOT)/human_g1k_v37_chr20.fa
INDEL_PREFIX = $(REF_ROOT)/1kg.pilot_release.merged.indels.sites.hg19
DBSNP_VCF = $(REF_ROOT)/dbSNP135_chr20.vcf.gz
HM3_VCF = $(REF_ROOT)/hapmap_3.3.b37.sites.chr20.vcf.gz

# Update thunder options so it will run faster for the tutorial:
# * run with only 10 rounds instead of 30 (-r option)
# * run without the --compact option
# This will use more memory, but is not an issue for this small sample data.
THUNDER = $(UMAKE_ROOT)/bin/thunderVCF -r 10 --phase --dosage --inputPhased

workshop:~> █
```


Path to Reference Files so reference directory only has to be set once

# Tutorial Configuration File

```
workshop:~> cat /opt/gotcloudExample/GBR60vc.conf
CHRS = 20
BAM_INDEX = /opt/gotcloudExample/GBR60bam.index
#####
# References
REF_ROOT = /opt/gotcloudExample/chr20Ref
#
REF = $(REF_ROOT)/human_g1k_v37_chr20.fa
INDEL_PREFIX = $(REF_ROOT)/1kg.pilot_release.merged.indels.sites.hg19
DBSNP_VCF = $(REF_ROOT)/dbsnp135_chr20.vcf.gz
HM3_VCF = $(REF_ROOT)/hapmap_3.3.b37.sites.chr20.vcf.gz

# Update thunder options so it will run faster for the tutorial:
# * run with only 10 rounds instead of 30 (-r option)
# * run without the --compact option
# This will use more memory, but is not an issue for this small sample data.
THUNDER = $(UMAKE_ROOT)/bin/thunderVCF -r 10 --phase --dosage --inputPhased

workshop:~> █
```



# Tutorial Configuration File

```
workshop:~> cat /opt/gotcloudExample/GBR60vc.conf
CHRS = 20
BAM_INDEX = /opt/gotcloudExample/GBR60bam.index
#####
# References
REF_ROOT = /opt/gotcloudExample/chr20Ref
#
REF = $(REF_ROOT)/human_g1k_v37_chr20.fa
INDEL_PREFIX = $(REF_ROOT)/1kg.pilot_release.merged.indels.sites.hg19
DBSNP_VCF = $(REF_ROOT)/dbsnp135_chr20.vcf.gz
HM3_VCF = $(REF_ROOT)/hapmap_3.3.b37.sites.chr20.vcf.gz
```

Override THUNDER  
command so it will  
run faster for the  
tutorial

```
# Update thunder options so it will run faster for the tutorial:
# * run with only 10 rounds instead of 30 (-r option)
# * run without the --compact option
# This will use more memory, but is not an issue for this small sample data.
THUNDER = $(UMAKE_ROOT)/bin/thunderVCF -r 10 --phase --dosage --inputPhased
```

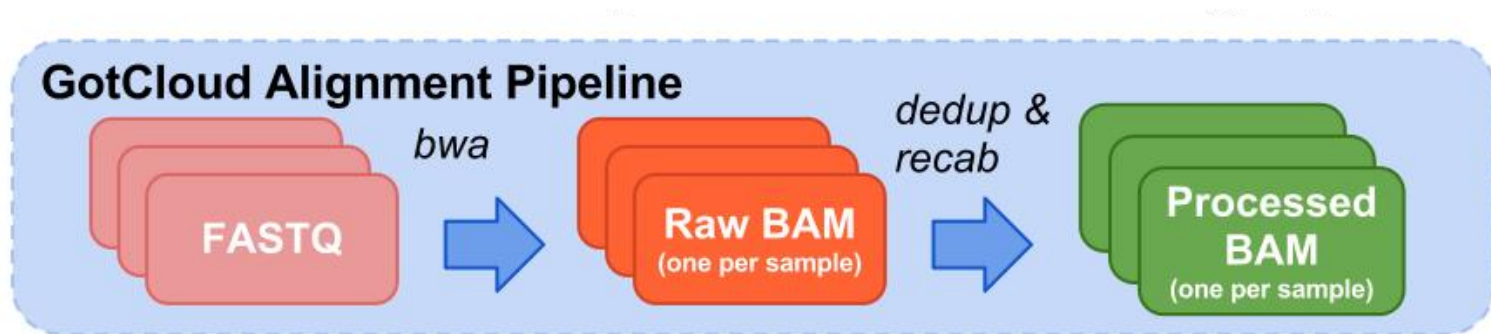
```
workshop:~> █
```

# Ready to Run Variant Calling Pipeline

- Type the following in your terminal to run:

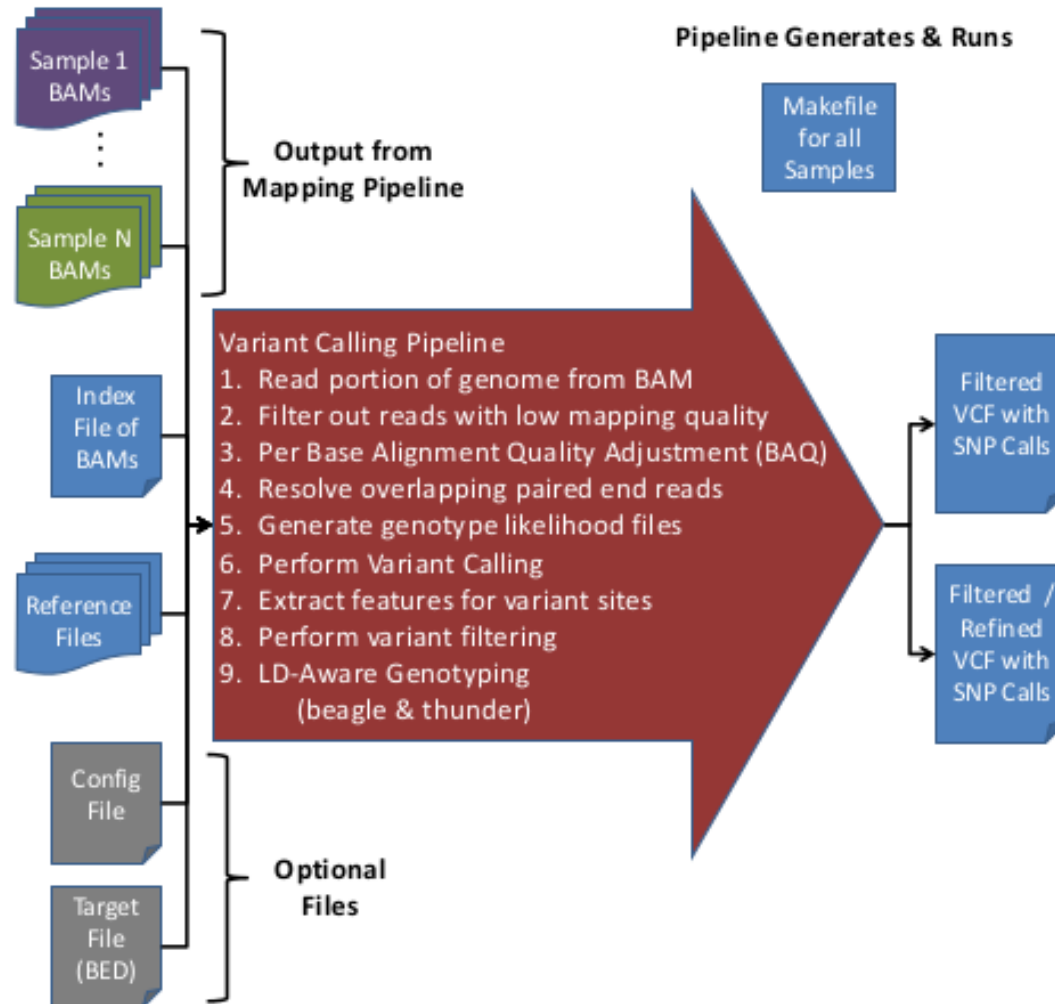
```
gotcloud align --conf /opt/gotcloudExample/GBR60align.conf --outdir ~/gotcloudTutorial
```

- This runs the whole alignment pipeline



# Variant Calling Pipeline

## Automatically Runs Several Steps



# Alignment Pipeline Running in Parallel

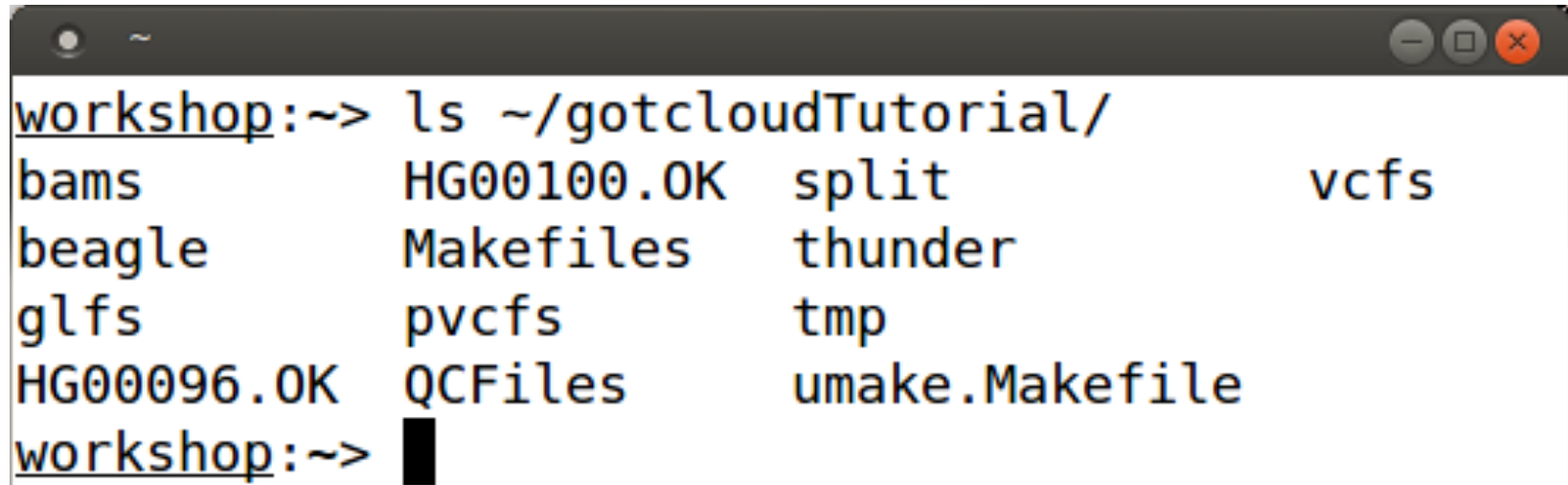
- Specify # of samples to run at once
  - `--numcs` or `--numconcurrentsamples`
- Number of jobs per sample
  - `--numjobs` or `--numjobspersample`
- Run multiple threads of BWA
  - Set `BWA_THREADS = -t N`
  - where N is the number of threads to run
- Run on a cluster
  - `--batchtype clusterType`
    - local (default), sge, slurm, mosix
  - `--batchopts clusterOptions`
    - Specify any options you need for running on your cluster

# Variant Calling Pipeline Running in Parallel

- Number of jobs to run at once
  - `--numjobs`
- Run on a cluster
  - `--batchtype clusterType`
    - local (default), sge, slurm, mosix
  - `--batchopts clusterOptions`
    - Specify any options you need for running on your cluster
- Variant calling pipeline breaks up processing by sample & region, then just by region
- Ultimately, 1 VCF per chromosome

# Tutorial Output Directory

- Output put into ~/gotcloudTutorial/

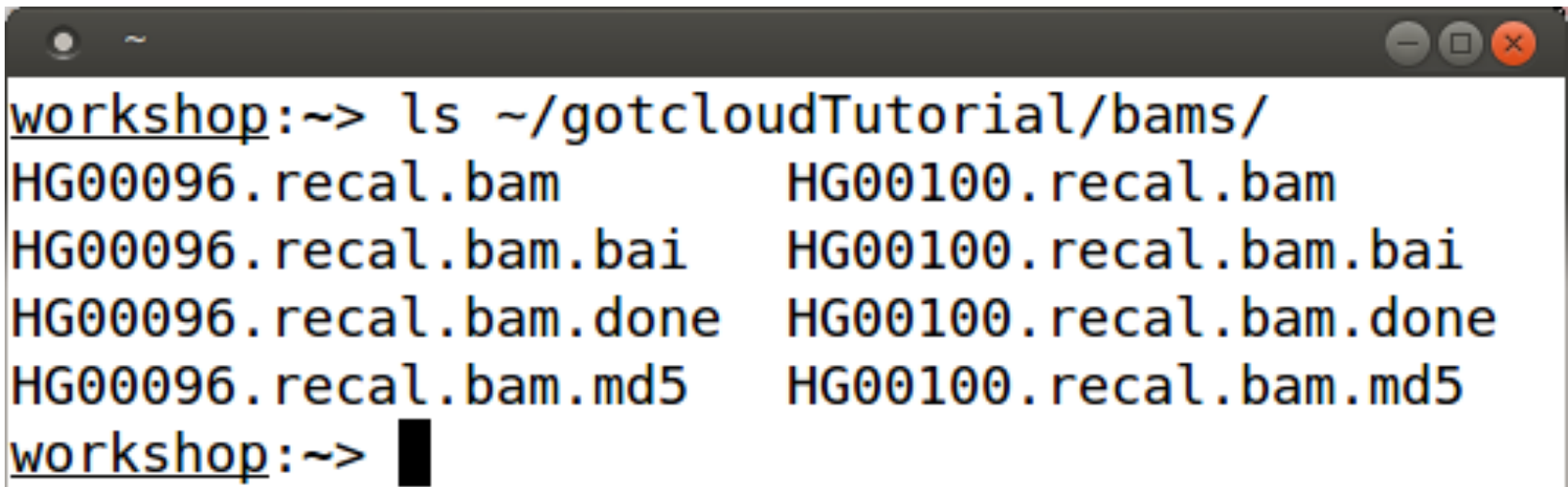


```
workshop:~> ls ~/gotcloudTutorial/  
bams          HG00100.OK    split          vcfs  
beagle        Makefiles     thunder  
glfs          pvcfs         tmp  
HG00096.OK   QCFiles       umake.Makefile  
workshop:~> █
```



# Alignment Pipeline Output

- ~/gotcloudTutorial/bams/ contains final output of Alignment Pipeline



```
workshop:~> ls ~/gotcloudTutorial/bams/  
HG00096.recal.bam          HG00100.recal.bam  
HG00096.recal.bam.bai     HG00100.recal.bam.bai  
HG00096.recal.bam.done    HG00100.recal.bam.done  
HG00096.recal.bam.md5     HG00100.recal.bam.md5  
workshop:~> █
```

# Variant Calling Pipeline Output

```
workshop:~> ls ~/gotcloudTutorial/vcfs/chr20/chr20.filtered.*  
/home/marykt/gotcloudTutorial/vcfs/chr20/chr20.filtered.sites.vcf  
/home/marykt/gotcloudTutorial/vcfs/chr20/chr20.filtered.sites.vcf.log  
/home/marykt/gotcloudTutorial/vcfs/chr20/chr20.filtered.sites.vcf.summary  
/home/marykt/gotcloudTutorial/vcfs/chr20/chr20.filtered.vcf.gz  
/home/marykt/gotcloudTutorial/vcfs/chr20/chr20.filtered.vcf.gz.OK  
/home/marykt/gotcloudTutorial/vcfs/chr20/chr20.filtered.vcf.gz.tbi  
workshop:~> █
```

```
workshop:~> cd ~/gotcloudTutorial/thunder/chr20/GBR/thunder/  
workshop:~/gotcloudTutorial/thunder/chr20/GBR/thunder> ls *thunder.vcf*  
chr20.filtered.PASS.beagled.GBR.thunder.vcf.gz  
chr20.filtered.PASS.beagled.GBR.thunder.vcf.gz.err  
chr20.filtered.PASS.beagled.GBR.thunder.vcf.gz.tbi  
workshop:~/gotcloudTutorial/thunder/chr20/GBR/thunder> █
```

# More on Configuration Files

- **How to Override Settings**
  - There is a default configuration file for each pipeline
  - Anything set in there can be overridden in the user configuration file
  - Just set the KEY to a different value in your configuration file
  - The Reference file settings you have in GBR2.conf override the default settings (whole genome)
- **Most settings in default configuration files should not be modified**
  - They are there so new options/modifications can easily be tested

# more /usr/local/gotcloud/bin/alignDefaults.conf

```
#####  
# References  
#####  
REF_DIR = $(PIPELINE_DIR)/ref  
AS = NCBI37  
FA_REF = $(REF_DIR)/hs37d5.fa  
DBSNP_VCF = $(REF_DIR)/dbsnp_135.b37.vcf.gz  
HM3_VCF = $(REF_DIR)/hapmap_3.3.b37.sites.vcf.gz  
#####  
# Configure QC steps on  
#####  
VERIFY_BAM_ID_OPTIONS =  
  
#####  
# Configure QC steps on  
#####  
RUN_QPLOT = 1  
RUN_VERIFY_BAM_ID = 1  
#####  
# Output Directory  
#####  
FINAL_BAM_DIR = $(OUT_DIR)/bams  
#####  
# BINARIES  
#####  
--More-- (46%)
```

GBR2.conf overrides these reference settings

Turn off QC steps by setting:

- RUN\_QPLOT = 0
- RUN\_VERIFY\_BAM\_ID = 0

in your GBR2.conf  
Do not edit alignDefaults.conf

# rest of /usr/local/gotcloud/bin/alignDefaults.conf

```
#####  
BIN_DIR = $(PIPELINE_DIR)/bin  
MD5SUM_EXE = md5sum  
SAMTOOLS_EXE = $(BIN_DIR)/samtools  
BWA_EXE = $(BIN_DIR)/bwa  
VERIFY_BAM_ID_EXE = $(BIN_DIR)/verifyBamID  
QPLOT_EXE = $(BIN_DIR)/qplot  
BAM_EXE = $(BIN_DIR)/bam  
#####  
# Alignment Info  
#####  
BWA_THREADS = -t 1  
BWA_QUAL = -q 15  
BWA_MAX_MEM = 2000000000  
#####  
# Temporary Directories  
#####  
TMP_DIR = $(OUT_DIR)/tmp  
SAI_TMP = $(TMP_DIR)/bwa.sai.t  
ALN_TMP = $(TMP_DIR)/alignment.bwa  
POL_TMP = $(TMP_DIR)/alignment.pol  
MERGE_TMP = $(TMP_DIR)/alignment.pol  
DEDUP_TMP = $(TMP_DIR)/alignment.dedup  
RECAL_TMP = $(TMP_DIR)/alignment.recal  
QC_DIR = $(OUT_DIR)/QCFiles  
workshop:~> █
```

BWA settings  
you can modify

# more /usr/local/gotcloud/bin/umakeDefaults.conf

```
#####  
# UMAKE DEFAULT CONFIGURATION FILE  
# This configuration file contains default run-time configuration of  
# UMAKE SNP calling pipeline.  
# The user configuration file is read prior to reading this file.  
# Only keys that have not yet been set are read from this file, preserving  
# the user configuration values.  
# UMAKE_ROOT is defined in the script prior to reading any configuration and is  
# set to one directory above the umake.pl script.  
#####  
## REQUIRED ELEMENTS FOR THE USER TO SET VIA CONF OR PARAMETERS  
#####  
#OUT_DIR=  
#INPUT_ROOT=  
#BAM_INDEX=  
CHRS = 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 X Y # List of chromos  
omes to call SNPs. For multiple chromosomes, separate by whitespace  
#####  
## Optional Settings  
#####  
OUT_PREFIX = umake # prefix of output Makefile $(OUT_PREFIX).Makefile will be generat  
ed  
#PED_INDEX = $(INPUT_ROOT)/umake-example.ped # SAMPLE PED FILE (required only for c  
hrX calling)  
#  
#####  
- More - (14%)
```

## Default Set of Chromosomes

CHRS = 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 X Y # List of chromosomes to call SNPs. For multiple chromosomes, separate by whitespace

```
## ARGUMENT FOR VCF FILTERING
#####
# The following set of values are used for applying filters to the VCF.
# To remove a filter, set it to blank or off in your user configuration file

# The following values set the min/max depth filter.
# The minDP filter is calculated using FILTER_MIN_SAMPLE_DP * numSamples or
# The maxDP filter is calculated using FILTER_MAX_SAMPLE_DP * numSamples or
FILTER_MAX_SAMPLE_DP = 20 # Max Depth per Sample
FILTER_MIN_SAMPLE_DP = 1 # Min Depth per Sample

# To remove a filter, set it to blank or off in your user configuration file
# The values of these filters must be numbers (or comma/space separated list of numbers)
# These rules apply to the following filters:
#   Specifying 1 value in the filter will turn that filter on and use that value.
#   Specifying 2 values in the filter (separated by ',' and/or ' ') turns on the filter.
#       Use the 1st value if the number of samples is below FILTER_FORMULA_MIN_SAMPLES
#       Use the 2nd value if the number of samples is above FILTER_FORMULA_MAX_SAMPLES
#       If the number of samples is between the MIN & MAX, a logscale is used:
#        $(\text{minVal} - \text{maxVal}) * (\log(\text{maxSamples}) - \log(\text{numSamples})) / (\log(\text{maxSamples}) - \log(\text{minSamples})) + \text{maxVal}$ 
FILTER_FORMULA_MIN_SAMPLES = 100
FILTER_FORMULA_MAX_SAMPLES = 1000
FILTER_WIN_INDEL = 5
--More-- (28%)
```

```
FILTER_MAX_AOI = 5
FILTER_MAX_ABL = 70, 65
FILTER_MAX_STR = 20, 10
FILTER_MIN_STR = -20, -10
FILTER_MAX_STZ = 5, 10
FILTER_MIN_STZ = -5, -10
FILTER_MIN_FIC = -20, -10

#####
## ARGUMENT FOR SAMTOOLS FILTERof reads
#####
SAMTOOLS_VIEW_FILTER = -q 20 -F 0x0704 # samtools view filter (-q by MQ, -F by flag)

#####
## STEPS TO RUN : COMMENT OUT TO EXCLUDE CERTAIN STEPS
## --snpcall, --extract, --beagle, --thunder commands automatically set them
#####
RUN_INDEX = TRUE          # create BAM index file
RUN_PILEUP = TRUE         # create GLF file from BAM
RUN_GLFMULTIPLES = TRUE  # create unfiltered SNP calls
RUN_VCFPILEUP = TRUE     # create PVPF files using vcfPileup and run infoCollector
RUN_FILTER = TRUE        # filter SNPs using vcfCooker
RUN_SPLIT = TRUE         # split SNPs into chunks for genotype refinement
RUN_BEAGLE = TRUE        # BEAGLE - MUST SET AFTER FINISHING PREVIOUS STEPS
RUN_SUBSET = TRUE        # SUBSET FOR THUNDER - MAY BE SET WITH BEAGLE STEP TOGETHER
RUN_THUNDER = TRUE       # THUNDER - MUST SET AFTER FINISHING PREVIOUS STEPS
--More-- (42%)
```



```
#
#####
## OPTIONS FOR GLFEXTRACT (GLFMULTIPLES, VCFPILEUP, FILTER MUST BE TURNED OFF)
#####
#RUN_EXTRACT = TRUE # Instead of VCF_EXTRACT
#VCF_EXTRACT = # whole-genome information to genotype (such a
#####
## OPTIONS FOR EXOME/TARGETED SEQUENCING
#####
#WRITE_TARGET_LOCI = TRUE # FOR TARGETED SEQUENCING ONLY -- Write loci file when performing pileup
#UNIFORM_TARGET_BED = $(INPUT_ROOT)/umake-example.bed # Targeted sequencing : When all individuals has the same target. Otherwise, comment it out
#OFFSET_OFF_TARGET = 50 # Extend target by given # of bases
#MULTIPLE_TARGET_MAP = # Target per individual : Each line contains [SM_ID] [TARGET_BED]
#TARGET_DIR = target # Directory to store target information
#SAMTOOLS_VIEW_TARGET_ONLY = TRUE # When performing samtools view, exclude off-target regions (may make command line too long)
#####
## RESOURCE FILES : Download the full resources for full genome calling
#####
--More-- (57%)
```

EXOME/Targeted sequencing settings that you would set in your configuration file if applicable

```
REF = $(UMAKE_ROOT)/ref/karma.ref/human.g1k.v37.fa
INDEL_PREFIX = $(UMAKE_ROOT)/ref/indels/1kg.pilot_release.merged.indels.sites.hg19 # 1
000 Genomes Pilot 1 indel VCF prefix
DBSNP_VCF = $(UMAKE_ROOT)/ref/dbSNP/dbsnp_135.b37.vcf.gz # dbSNP file
HM3_VCF= $(UMAKE_ROOT)/ref/HapMap3/hapmap_3.3.b37.sites.vcf.gz # HapMap3 polymorphic
site
#
#####
## BINARIES
#####
SAMTOOLS_FOR_PILEUP = $(UMAKE_ROOT)/bin/samtools-hybrid # for samtools pileup
SAMTOOLS_FOR_OTHERS = $(UMAKE_ROOT)/bin/samtools-hybrid # for samtools view and calmd
GLFMERGE = $(UMAKE_ROOT)/bin/glfMerge # glfMerge when multiple BAMs exist per individua
l
GLFMULTIPLES = $(UMAKE_ROOT)/bin/glfMultiples --minMapQuality 0 --minDepth 1 --maxDepth
h 10000000 --uniformTsTv --smartFilter # glfMultiples and options
GLFEXTRACT = $(UMAKE_ROOT)/bin/glfExtract # glfExtract for obtaining VCF for known si
tes
VCFPILEUP = $(UMAKE_ROOT)/bin/vcfPileup # vcfPileup to generate rich per-site infor
mation
INFOCOLLECTOR = $(UMAKE_ROOT)/bin/infoCollector # create filtering statistics
VCFMERGE = perl $(UMAKE_ROOT)/scripts/bams2vcfMerge.pl # merge multiple BAMs separated
by chunk of genomes
VCFCOOKER = $(UMAKE_ROOT)/bin/vcfCooker # vcfCooker for filtering
VCFSUMMARY = perl $(UMAKE_ROOT)/scripts/vcf-summary # Get summary statistics of discov
ered site
--More-- (72%)
```

```
VCFSPLIT = perl $(UMAKE_ROOT)/scripts/vcfSplit.pl # split VCF into overlapping chunks
for genotype refinement
VCFPASTE = perl $(UMAKE_ROOT)/scripts/vcfPaste.pl # vcfPaste to generate filtered geno
type VCF
BEAGLE = java -Xmx4g -jar $(UMAKE_ROOT)/bin/beagle.20101226.jar seed=993478 gprobs=tru
e niterations=50 lowmem=true # BEAGLE BINARY : NEED TO COPY BEAGLE TO $(UMAKE_ROOT)/ex
t DIRECTORY BEFORE RUNNING PIPELINE
VCF2BEAGLE = perl $(UMAKE_ROOT)/scripts/vcf2Beagle.pl --PL # convert VCF (with PL tag)
into beagle input
BEAGLE2VCF = perl $(UMAKE_ROOT)/scripts/beagle2Vcf.pl # convert beagle output to VCF
THUNDER = $(UMAKE_ROOT)/bin/thunderVCF -r 30 --phase --dosage --compact --inputPhased
# MaCH/Thunder genotype refinement step
LIGATEVCF = perl $(UMAKE_ROOT)/scripts/ligateVcf.pl # ligate multiple phased VCFs whil
e resolving the phase between VCFs
BGZIP = $(UMAKE_ROOT)/bin/bgzip
TABIX = $(UMAKE_ROOT)/bin/tabix
BAMUTIL = $(UMAKE_ROOT)/bin/bam
#
#####
## RELATIVE DIRECTORY UNDER OUT_DIR
#####
BAM_GLF_DIR = glfs/bams # BAM level GLF
SM_GLF_DIR = glfs/samples # sample level GLF (after glfMerge if necessary)
VCF_DIR = vcfs # unfiltered and filtered VCF
PVCF_DIR = pvcfs # vcfPileup results
SPLIT_DIR = split # chunks split to multiple overlappingpieces
--More-- (87%)
```

```
VCF_DIR = vcfs           # unfiltered and filtered VCF
PVCF_DIR = pvcfs        # vcfPileup results
SPLIT_DIR = split       # chunks split to multiple overlapping pieces
BEAGLE_DIR = beagle     # beagle output
THUNDER_DIR = thunder   # MaCH/thunder output
GLF_INDEX = glfIndex.ped # glfMultiples/glfExtract index file info
#
#####
## OTHER OPTIONS
#####
UNIT_CHUNK = 5000000    # Chunk size of SNP calling : 5Mb is default
LD_NSNS = 10000        # Chunk size of genotype refinement : 10,000 SNPs
LD_OVERLAP = 1000      # Overlapping # of SNPs between chinks : 1,000 SNPs
RUN_INDEX_FORCE = FALSE # Regenerate BAM index file even if it exists
MERGE_BEFORE_FILTER = FALSE # Merge across the chromosome before filtering
NOBAQ_SUBSTRINGS = SOLID # Avoid BAQ if the BAM file contains the substring
ASSERT_BAM_EXIST = FALSE # Check if BAM file exists
#
#####
## CLUSTER SETTING : CURRENTLY COMPATIBLE WITH MOSIX PLATFORM
#####
SLEEP_MULT =
BATCH_TYPE =
BATCH_OPTS =
REMOTE_PREFIX = # REMOTE_PREFIX : Set if cluster node see the directory differently (
e.g. /net/mymachine/[original-dir])
workshop:~> █
```

**Thanks!**

# Recalibration

- **Covariates**
  - Read Group
  - Quality
  - Cycle
  - 1st/2nd read in pair
  - Previous Base
  - This Base
- **Skip**
  - duplicates, unmapped, mapping quality 0/255, insertions, dbsnp position, base quality < min (default 5)
- **Look at Matches vs Mismatches to calculate new quality**

# gotcloud Commands

- gotcloud Usage

```
workshop:~> gotcloud
ERROR: Missing command. Please see the usage below.
Usage:
  gotcloud [command] [options]

Command:
  help          Print out brief help message
  man           Print the full documentation in man page style
  align         Run the alignment pipeline
  snpcall       Run the snp calling pipeline
  ldrefine      Run the LD-aware genotype refinement pipeline

Visit http://genome.sph.umich.edu/wiki/GotCloud for more detailed documentation

workshop:~> █
```