Churchill

v1.8

Installation & User Guide

"Continuous effort – not strength or intelligence – is the key to unlocking our potential."

Sir Winston Churchill. 3rd May, 1952

What is Churchill?

Churchill brings together the most commonly utilized tools for discovery of genetic variation in to a single pipeline using currently accepted best practices, fully automating alignment, deduplication, local realignment, base quality score recalibration, variant calling and genotyping. Churchill achieves high levels of balanced parallelism throughout the analysis workflow, producing deterministic results no matter the analysis scale and regardless of the platform it is executed on. Reproducible data analysis can be rapidly completed without sacrificing data quality or integrity.

Setup and execution of Churchill is performed with a single command and only requires a small number of pre-installed components. A single configuration file defines the paths to raw data, installed software, required database files, and delivery directories. To ensure that Churchill would be of utility to the widest number of researchers, the pipeline was developed such that it is compatible with a wide range of Linux systems including high-performance workstations, small single servers, moderate in-house clusters with shared or non-shared memory servers, large HPC systems housed at supercomputing centers and the cloud.

The software is suitable for researchers with limited bioinformatics experience and by design limits the number of configuration options for the multiple components of the analysis process. In addition to maintaining a simplified workflow, limiting configuration options was essential to maintain deterministic and optimized performance.

Please cite the original Churchill paper in all work resulting from the use of this software:

> Benjamin J Kelly, James R Fitch, Yangqiu Hu, Donald J Corsmeier, Huachun Zhong, Amy N Wetzel, Russell D Nordquist, David L Newsom and Peter White. Churchill: an ultra-fast, deterministic, highly scalable and balanced parallelization strategy for the discovery of human genetic variation in clinical and population-scale genomics. Genome Biology. 2015.

Minimum System Requirements

CHURCHILL requires the following:

- Quad-core Intel or AMD processor
- 16GB RAM
- 2TB free hard disk space

Recommended System Requirements

The following are recommended:

- Multiple multi-core Intel or AMD processors
- 4GB RAM per core
- Multiple hard disks in RAID 0 configuration or high-performance storage arrays such as Isilon or Lustre

Additional Packages Required

CHURCHILL requires the following third-party software packages installed and correctly configured:

- Linux
- Java 1.6-1.7
- Python 2.7
- BWA 0.5.10 – 0.7.10
- SAMtools 0.1.19
- Picard Tools 1.104
- GATK 1.6 – GATK 3.2
- PySam 0.7.5
- FreeBayes 0.9.14 (optional)

The Lite version of GATK may be downloaded from:

```
ftp://ftp.broadinstitute.org/distribution/gsa/GenomeAnalysisTK/
```

The latest version of GATK may be downloaded from:

```
https://www.broadinstitute.org/gatk/download
```

1. Installation Procedure

1.1 **Decompressing the Churchill files.** In the directory in which you have downloaded the Churchill pipeline bundle, extract the archived files using the following command:

```
tar zxvf Churchill.tar.gz
```

1.2 **Creating the reference library.** You can create or update your own reference library and annotation databases by downloading the data sets from their respective repositories. However, for maximum compatibility and ease of installation we recommend downloading the Genome Analysis Toolkit (GATK) resource bundle from the Broad Institute at http://www.broadinstitute.org/gatk/download.

The resource bundle contains recent versions of the reference and annotation files utilized by the Churchill pipeline, including the following:

- The human reference sequence
- dbSNP VCF files
- HapMap genotypes and sites VCF files
- OMNI 2.5 gentotypes and sites VCF files
- 1000 Genomes Phase 1 indel calls
- Mills and 1000 Genome gold standard indels

These files must be decompressed prior to use with Churchill. The reference genome must be indexed using your installed version of BWA. Also, a fasta file index (created using SAMtools) and a sequence dictionary (created using Picard Tools) must be generated. For full details see:

http://gatkforums.broadinstitute.org/discussion/2798/howto-prepare-a-reference-for-use-with-bwa-and-gatk

2. The Churchill Analysis Configuration File

2.1 **Configuring Churchill for your environment.** The configuration parameters for Churchill, as well as paths to reference files and applications, are contained within a single, plain text configuration file that is specified as an input parameter to the `churchill.py` script. You will need to change these parameters to reflect your installation environment and the current analysis project.

2.2 **Experiment-specific parameters.** The `EXPERIMENT` parameter is the project name of your choice. `EXPERIMENT_OUT_DIRECTORY` is the directory in which you want Churchill to place the analysis output. `EXPERIMENT_TYPE` is either `EXOME` (the default) or `GENOME`; this parameter is used to appropriately set configuration options.

`NUM_REGIONS` tells Churchill how many regions in which to split the genome for the analysis. Note: if Churchill is being run in a Make or Shared Memory environment, this should be set to the twice the number of cores available. This will help with load balancing, producing results more efficiently. A minimum of 2 regions is required.

Set `MEM_PER_CORE` to a value that reflects your hardware and environment.

The `GATK_DCOV` parameter allows you to specify the down-sampling value that GATK uses. For deterministic behavior, this needs to be 0 (default).

If you want Churchill to process reads with a mapping quality of zero (i.e. reads mapping equally to multiple locations), set `FILTER_MAPQ0` to "N". The default behavior ("Y") is to filter these reads out and not include them in analysis.

The `MULTISAMPLE_VC` option specifies whether you want Churchill to process the samples individually or as a collective analysis.

The `VCF_VQSR` option specifies whether or not for Churchill to run GATK VQSR after variant calling. This option is NOT recommended for a small number of samples.

The `GATK_VER` option should be set to "V1" if you are using a 1.X release of GATK, "V2" for a 2.X release or "V3" for a 3.X release. "V1" is the default.

The `VAR_CALLER` option specifies which variant caller to use and can be set to "GATK_UG", "GATK_HC", or "FREEBAYES". GATK_HC only works with release 3.x or greater of GATK and the FREEBAYES option requires BIN_FREEBAYES to be set. **NOTE: options GATK_HC and FREEBAYES may introduce non-determinism.**

The `BIN_FREEBAYES` option specifies the location of the FreeBayes executable.

The `ALIGNER` option specifies the aligner to use and can be set to "BWA_MEM" or "BWA_BT". BWA_MEM is the default and requires version 0.7.X or greater of BWA. BWA_BT is recommended to only be used with versions 0.6.X or lower of BWA.

```
# Experiment name (required) - must not contain spaces
EXPERIMENT = Test_Experiment

# Experiment output directory (required) - analysis output will go here
EXPERIMENT_OUT_DIR = /projects/example_output_folder

# Experiment type (required) - options: EXOME (default), GENOME
EXPERIMENT_TYPE = GENOME

# Number of regions to be used (required)
NUM_REGIONS = 48

#Churchill options
MEM_PER_CORE=4g
ALIGNER=BWA_MEM
BWA_TRIMMING_Q=0
FILTER_MAPQ0=Y
GATK_VER=V1
VAR_CALLER=GATK_UG
GATK_DCOV=0
MULTISAMPLE_VC=Y
VCF_VQSR=N

# Job packaging options (required for SGE/PBS)
JOB_SCHEDULER=SGE
JOB_HEADER=/path-to/header.txt
# JOBS_PER_PACKAGE required for all versions (set to 1 for Make/Shared)
JOBS_PER_PACKAGE=16

# Reference files and application paths
FA_HG19=/hg19/ucsc.hg19.fasta
VCF_DBSNP=/hg19/dbsnp_138.vcf
VCF_1000G_INDEL=/hg19/1000G_phase1.indels.vcf
VCF_MILLS_DEVINE_INDEL=/hg19/Mills_and_1000G_gold_standard.indels.vcf
VCF_HAPMAP_33=/hg19/hapmap_3.3.vcf
VCF_1000G_OMNI25=/hg19/1000G_omni2.5.vcf
JAR_GATK=/applications/GenomeAnalysisTK-1.6-13/GenomeAnalysisTK.jar
GATK_RESOURCES=/applications/gatk/GenomeAnalysisTK-1.6/public/R
BIN_BWA=/applications/bwa/bwa-0.7.10/bwa
BIN_SAMTOOLS=/applications/samtools/samtools-0.1.19/samtools
BIN_FREEBAYES=/applications/freebayes/bin/freebayes
DIR_PICARD=/applications/picard/picard-tools-1.104
PICARD_TMP_DIR=/temp

# List of samples
SAMPLES += Sample1
Sample1.FLOWCELL_ID = C01AWW30X
Sample1.RUN_DATE = 140618
Sample1.FASTQ_BASEDIRS += /path-to/Sample1-fastq/

SAMPLES += Sample2
Sample2.FLOWCELL_ID = C01AWW30X
Sample2.RUN_DATE = 140618
Sample2.FASTQ_BASEDIRS += /path-to/Sample2-fastq
```

Figure 2.1 The Churchill Analysis Configuration File

2.3 **Job scheduler and packaging parameters.** When Churchill is run in an environment with a job scheduler, i.e., Sun Grid Engine (SGE) or Portable Batch System (PBS), you will also need to specify the following options, according to your environment:

The `JOB_SCHEDULER` parameter specifies which job scheduler you use in your environment. The value can be `SGE` or `PBS`.

The `JOB_HEADER` parameter should point to a job header file that specifies command options shown below. This should be customized to your environment but should always include the queue to submit the jobs to and how many cores to use per job package (i.e. in an SGE environment, "-pe smp 16" would create job packages with 16 parallel running processes to run on a node where 16 cores are available).

```
# Example job header text file for SGE
#
#$ -q all.q
#$ -j y
#$ -S /bin/bash
#$ -pe smp 16
```

`JOBS_PER_PACKAGE` should be set to match the number of cores specified for the symmetric multiprocessor parallel environment in the job header file (e.g. 16 if "-pe smp 16" is used within the header file).

2.4 **Sample-specific parameters.** Sample information is specified as shown in Figure 2.1. First, a given sample is added to the collection of `SAMPLES` with the following syntax:

```
SAMPLES += Sample1
```

Sample names must be 2 or more characters in length. Next, sample-specific parameters are given. `FLOWCELL_ID` and `RUN_DATE` can be set to whatever you wish (no spaces), but typically reflect the headers in the FASTQ files. With the `FASTQ_BASEDIRS` parameter(s), you specify the location of the input FASTQ files. Multiple locations may be specified, as shown in the following example:

```
# List of samples
SAMPLES += Sample1
Sample1.FLOWCELL_ID = C01AWW30X
Sample1.RUN_DATE = 130618
Sample1.FASTQ_BASEDIRS += /path-to/fastq-folder1/
Sample1.FASTQ_BASEDIRS += /path-to/fastq-folder2/
```

The prefix for the sample-specific parameters (shown highlighted) must match the sample name.

Note: the way Churchill determines whether the dataset for a given experiment is single- or paired-end is by the naming convention of the FASTQ files. More specifically, a paired-end dataset will consist of FASTQ file pairs ending with "_R1" and "_R2" and a single-end dataset will consist of FASTQ files ending with "_R1". You may need to adjust the FASTQ file names depending on your specific platform: **it is essential that FASTQ files following this naming convention**.

Note: implementation of alignment within the Churchill pipeline utilizes an approach whereby the total raw input sequencing data (typically 400-800 million paired reads) is split into multiple smaller FASTQ files and aligned using multiple single-threaded parallel instances of the alignment algorithm. The number of paired-end FASTQ files generated during the sequencing run is controlled by the --fastq-cluster-count parameter of Illumina's BCL-conversion process (CASAVA 1.8.2), which specifies the maximum number of reads per output FASTQ file. The default value of 4,000,000 works well with Churchill. However, decreasing the number of reads per FASTQ to 1,000,000 results in increased alignment speed due to better load balancing. If you are starting with a single merged FASTQ file, NGSUtils has a program to split large FASTQ files into smaller chunks (http://ngsutils.org/modules/fastqutils/split/).

2.5 **Reference file and application paths.** Finally, the directory locations of your reference files and the paths to the prerequisite application folders are specified. (See the sections "**Additional Packages Required**" and "**1.2 Creating the references library.**" for more information.)

3. Running the Churchill Pipeline

3.1 **Running the pipeline.** Once the analysis configuration file is customized for the given analysis run, the Churchill pipeline can be started. From the directory to which you unzipped the Churchill package, run the `churchill.py` script by issuing the following

```
python ./churchill.py <run_mode> config_file
```

where `run_mode` is any of the following, according to your environment:

```
runPackaged:  Use the Sun Grid Engine (SGE) or Portable Batch
                    System (PBS/TORQUE) job schedulers for
                    running Churchill
runMake:      Use GNU Make for batch-queueing
runShared:    Run Churchill in a shared memory environment (e.g.
                    on a single computer)
```

The `config_file` parameter is simply the full path name of the config file you've created for the given analysis run. For example, the following command will run Churchill in a shared memory environment using the config file `my_experiment.config`:

```
./churchill.py runShared my_experiment.config
```

Churchill reads the configuration information from the configuration file and creates the project folder directory structure accordingly. The basic structure is shown below:

```
/<path-to>/projects/example_output_folder/multisample
                                          /Sample1/
                                          /Sample2/
```

Note: the multisample subdirectory is present only if you specify that Churchill runs a multisample analysis (see section "**2.2 Experiment-specific parameters**" for more information).

3.2 **Immediate output.** If output is not redirected, Churchill will display the configuration parameters you have specified to the screen, as well as information on the experiment, including experiment and sample info. **Figure 3.1** displays a typical Churchill output screen for the case of an SGE run.

```
Churchill:
  Run Mode = SGE
  SGE Queue = all.q
  Experiment = Test_Experiment
  Experiment Type = GENOME
  Experiment Out Dir = /projects/example_output_folder
  Num Regions = 48
Options:
  BWA_TRIMMING_Q = 0
  MEM_PER_CORE = 4g
  GATK_DCOV = 0
  FILTER_MAPQ0 = Y
  MULTISAMPLE_VC = Y
Sample Sample1 Info:
  RUN_DATE = 130618
  FASTQ_BASEDIRS = ['/path-to/Sample1-fastq/']
  FLOWCELL_ID = C01AWW30X
Sample Sample2 Info:
  RUN_DATE = 130618
  FASTQ_BASEDIRS = ['/path-to/Sample2-fastq/']
  FLOWCELL_ID = C01AWW30X
Input Files:
  Sample Sample1 (Sample1) has 57 pairs of PE FASTQ files
Input Files:
  Sample Sample2 (Sample2) has 51 pairs of PE FASTQ files
Cleaning up old jid files..
Submitting bwa job packages..
Submitting merge_bams job packages..
Submitting merge_dedup job packages..
Submitting realign/dedup/cc job packages..
Submitting merge csv job packages..
Submitting recal job packages..
Submitting ug job packages..
Submitting vqsr job packages..
```

Figure 3.1 Running the Churchill Pipeline. Churchill displays configuration parameters as it sets up the run. If you are running the pipeline in an environment with a job scheduler (SGE or PBS) Churchill will also display the job submission steps as shown.

4. Output from the Churchill Pipeline

4.1 **Output directory.** The output directory specified in the Churchill analysis configuration file parameter `EXPERIMENT_OUT_DIR` is created as a part of Churchill's analysis setup process. (See section "**2.2 Experiment-specific parameters.**" for more information).

```
EXPERIMENT_OUT_DIR = /projects/example_output_folder
```

4.2 **Final output for each sample.** Within this directory, each sample has its own subdirectory in which the output is written. For instance:

```
/projects/example_output_folder/Sample1/
```

In this case, `Sample1` is the output folder for the sample of the same name, as specified in the Churchill configuration file.

4.3 **Final output Binary Alignment/Map format (BAM) files.** Churchill stores the final BAM file output data in the following sub-directories of the sample directory:

```
.../Sample1/Interchromosomal_BAMs/
.../Sample1/Mapq0_BAMs/
.../Sample1/Processed_BAMs/
.../Sample1/Unmapped_BAMs/
```

Churchill stores the data that BWA was unable to align in the `Unmapped_BAMs` folder and does not use it in the analysis. Reads for which a map quality of zero was assigned are not used for variant calling, but are stored separately in the `Mapq0_BAMS` folder (only if option to filter reads with mapping quality 0 out of analysis is turned on).

Churchill creates a separate merged BAM file containing just the interchromosomal reads in the `Interchromosomal_BAMs` folder. These reads are appropriately processed by Churchill and are used for variant calling, and therefore are also stored in the final processed BAMs (see the Churchill publication for more information).

The number of final processed BAMs depends on the configuration parameter NUM_REGIONS (see section "**2.1 Analysis-specific parameters**"). There will be NUM_REGIONS individual regional BAM files, all stored in the `Processed_BAMs` folder. These can be easily viewed in the Integrated Genomics Viewer (IGV) as a single dataset by opening the *BAM list file* that is created in the same location:

```
.../Sample1/Processed_BAMs/Sample1.processed.bam.list
```

A region definition file can also be found in the Processed_BAMs folder.

4.4 Final output variant call format (VCF) files. Churchill stores the final VCF file output in the following sub-directory of the sample directory:

```
.../Sample1/VCFs
```

The raw SNP and INDEL calls produced by the Unified Genotyper step of the Genome Analysis Toolkit (GATK) are stored respectively in the files:

```
.../Sample1/VCFs/Sample1.raw_snp.churchill.vcf
.../Sample1/VCFs/Sample1.raw_indel.churchill.vcf
```

The final combined VCF, in which merged the SNPs and INDELs and filtered out variants with excess coverage when appropriate is stored in the file:

```
.../Sample1/VCFs/Sample1.final_combined.churchill.vcf
```

4.5 Multisample analysis output. If multisample variant calling is specified via the `MULTISAMPLE_VC` configuration file parameter, the location of the following final output files will be under the `multisample` subdirectory accordingly:

Processed BAM files:

```
.../multisample/Processed_BAMs/
```

Raw VCF files:

```
.../multisample/VCFs/multisample.raw_snp.churchill.vcf
.../multisample/VCFs/multisample.raw_indel.churchill.vcf
```

Final VCF file:

```
.../multisample/VCFs/multisample.final_combined.churchill.vcf
```

4.6 Run summary report. Basic run statistics and environment information are recorded in …/Logs/S3report.txt and uploaded to Amazon's S3 cloud storage service. GATK collects similar usage metrics as detailed here:

http://gatkforums.broadinstitute.org/discussion/1250/what-is-phone-home-and-how-does-it-affect-me