Secondary Analysis V4 (Changed on 9/10/2019)

New changes:

1. The new @RG in the header of BAMs are changed as:

@RG ID: AHTLY7BBXX.8 SM: SC310221 LB: SC310221_ATGCCTAA PL:ILLUMINA PU: AHTLY7BBXX20180606.8. ATGCCTAA CN:CGR

Remove PU, LB in the body part of BAMs

2. This new BAM format is more precisely for BQSR and dedup and save space

Dedup uses LB (all read IDs within one LB do picard dedup)

BQSR uses PU (all read IDs within one PU do BQSR in one batch)

3. In gatk_build_bam_for_single_name_v4.sh, skip phix removal, in general those reads have been removed in the primary analysis; (picard ReorderSam still can remove them if BAMs containing phix reads);

This script will also check if lane-level BAMs are new format or not, if not, use sed and samtools reformat them;

4. Step7a can also check if subject-level BAMs are old (in original), if yes, run misc_fix_bam_header_dedup.sh reformat and dedup a subject-level BAM.
5. recalibrate_bam.sh also skip check @RG header and the final consolidate to single @RG line to keep information complete.
6. The new BAMs are placed to /DCEG/Projects/Exome/SequencingData/BAM_reformatted

Usage:

1. If you first check out the scripts from gitlab, please do followings:

```
ln -s /DCEG/Projects/Exome/SequencingData/secondary_buf/global_config_bash_production.rc
global_config_bash.rc
```

2. **step0_run_scan_quality_trimming.sh ${DATE}** (e.g., step0_run_scan_quality_trimming.sh 2017-12-01)

The date means scan the logs after that date. So, this date can be changed as the most recent secondary analysis date. If don’t supply the date, the script will scan all logs and take longer time.

step0_run_scan_quality_trimming.sh will generate the **qt_all_${Date}.txt**, **qt_errors_${Date}.txt** and **qt_all.txt** under /DCEG/Projects/Exome/SequencingData/secondary_buf/input subfolder, which includes all new quality trimmed FASTQ files (qt_all.txt concatenated qt_all_${Date}.txt and previous qt_all_base.txt. After finish this step, qt_all.txt will copy a backup as qt_all_base.txt for next time. The qt_all_base.txt is used to prevent the tape archiving of log files in primary analysis).

***Note that any primary analysis quality trimming or demultiplexing in future, will need to change the scan_quality_trimming.sh accordingly.

3. **step0a_check_downsample_history.sh ${MANIFEST}**

Check if have any lane-level BAMs have been actually downsampled in previous build, if yes, double check with Kristie to see if they are fine to be used in the new build

4. **step0b_parse_qt.sh ${MANIFEST}**

(e.g., step0b_parse_qt.sh /home/wangm6/tmp2/AUG-FAMILIAL-POP-CONTROL-MANIFEST.csv)

The script will generate three files: **fastq_restoration.txt, samples_retrimming.txt, qt_err.txt and samples_new_trimmed.txt** in the /DCEG/Projects/Exome/SequencingData/secondary_buf/output/${DATE} folder.

In this step, the java already hardcoded two situations: BC3CLRACXX flowcell and seq date before 2014/04/30 used different CASAVA directory.

First check if "Comparison is done!" is in the end of log file /DCEG/Projects/Exome/SequencingData/secondary_buf/logs/parse_qt_${DATE}.stdout

Ask Nathan to restore FASTQ files listed in fastq_restoration.txt if not existed, then reprocess the primary analysis for those files using samples_retrimming.txt.

samples_retrimming.txt: prepared for redo quality trimming;

samples_new_trimmed.txt: the samples are already new quality trimmed;

Optional: also check **qt_err.txt**: The FATQ file in "Error: …" lines cannot be compared for quality trimming. (Error reasons: the sample Name have more than two "_" or no "_"; Flowcell name contains five or more "_" or no "_"; and some empty fields from 1-6)

5. If restore FASTQ in the last step, after primary analysis is finished, rerun previous two steps.

6. **step1_run_parse.sh**

    1) Two parameters are needed:

    sh ./step1_run_parse.sh /home/wangm6/tmp2/AUG-FAMILIAL-POP-CONTROL-MANIFEST.csv [redo|update] [depth]

    Parameters:

a. redo|update: redo means that redo merge no matter that the BAMs are already merged or not before; if using redo, the script can reprocess all lane-level BAMs; if using update, step7a will fix the old original BAMs; For most of small builds, it is recommended to use “redo” because step7a cannot handle duplicate RG IDs for control samples; no dedup metrics file not correct because the original BAMs already deduped). In step2, BAMs will be reformatted for the old lane-level BAMs; if not redo, need to run misc_fix_bam_header_dedup.sh using **step7a** to reprocess the old subject-level BAMs prior to variant calling;

b. depth is optional, if not set, the default is 60 means the required coverage used for downsampling ratio calculation;

2) Ouput files in the folder /DCEG/Projects/Exome/SequencingData/secondary_buf/output/${DATE}/

a. **manifest_errs.txt** (all the errors found in this step, including Manifest file format issue, all merging records, quality trimming records);

b. **new_update_analysisIDs.txt** (all samples merge lane changed or new sample or quality trimming changes) This step will also check if all lane-level BAMs for a sample in the manifest are new quality trimmed. If not, need redo merge. So, please check the file /DCEG/Projects/Exome/SequencingData/secondary_buf/output/${DATE}/new_update_ananlysisIDs.txt.

E.g. To check BC3C6EACXX//AGCCATGC/SB864474/2 is not in new QT list!
BC3C6EACXX//AGCCATGC/SB864474/6 is not in new QT list!
CMM_5925_3001_A BAM is not in new QT list!
In /20180103/new_update_analysisIDs.txt to see above errors. If have, it indicates not all the files are new QT yet.

(The new feature is: further check if have any underlying lane(s) are new quality trimmed based on time stamps)

filenames_restore.txt (all lane-level BAMs need restore); This step will generate the lane-bams which need to be restored; /DCEG/Projects/Exome/SequencingData/secondary_buf/output/${DATE}/filenames_restore.txt; Ask Nathan restore files using above **filenames_restore.txt** if not empty;

7. Check if merge needed for each sample. Generate **step2_merge_${DATE}.sh**

After restoration done, run above **step2_merge_${DATE}.sh**. (Note that, step2_merge_${DATE}.sh will be generated in /DCEG/Projects/Exome/SequencingData/secondary_buf/output/${DATE}; Before running, change SCRIPT_HOME variable according to production home location or copy step2_merge_${DATE}.sh to your local production script folder)

If using redo in 5, ask Nathan retrieve files which can be listed by following command line:

grep lane /DCEG/Projects/Exome/SequencingData/variant_scripts/logs/GATK/patch_build_bam_${date}/*.stdout | cut -d" " -f6

If the lane-level BAMs are old ones, the script gatk_build_bam_for_single_name_v4.sh will generate a new BAM suffixed as _reformated.bam and create a final soft link to this file.

After all jobs done on SGE, the log files will be generated in the /DCEG/Projects/Exome/SequencingData/variant_scripts/logs/GATK/patch_build_bam_${DATE}. Please use grep "Error" *.stdout to check if have any errors; Check the total number of BAM in /DCEG/Projects/Exome/SequencingData/BAM_new_incoming and the total number should be equals the line number in new_update_analysisIDs.txt in the step 2;

8. Now, run **step3_run_scan.sh** to rescan all log files in /DCEG/Projects/Exome/SequencingData/variant_scripts/logs/GATK to update the files in output folder. After this step done, two files: **all_merging_err.txt** and **all_merging_records.txt** will be generated in /DCEG/Projects/Exome/SequencingData/secondary_buf/input. Using grep "Error" all_merging_err.txt to check if have any problems in the merging steps. All the up-to-date merging records are listed in all_merging_records.txt.

(The log file of step3 itself is here: /DCEG/Projects/Exome/SequencingData/secondary_buf/logs/scan_merging_records_${DATE}.stdout

9. Run **step4_remove_P.sh**
10. Run QC (will not impact the processing. That means you can skip QC for the following data processing steps);
11. run **step7a_reformat_bams.sh $MANIFEST $RESTORE_FILE**
This step call misc_fix_bam_header_dedup.sh reformat old original BAMs.
If there are files have been achieved in BAM_original folder, (BAM_original backup in 6-month frequency) ask Nathan to retrive them using $RESTORE_FILE
Rerun step7a until $RESTORE_FILE is null and no qsub anymore (which means all BAMs are new formatted)
If some jobs are failed, please check logs in /DCEG/Projects/Exome/SequencingData/BAM_reformatted/BAM_original/LOGs. Please also delete WORKING flag files prior to resubmit jobs.
12. run QC (??)
13. run **step7b_take_incoming_bam.sh**
Please make sure that all jobs submitted by step7a or step2 have been finished successfully before launch this step7b, otherwise, the BAMs maybe incomplete.

14. Run **step8_sync_and_recalibrate_bam.sh $Manifest_file [SOMATIC|GERMLINE]**

If no manifest or type supplied, will search whole BAM_origninal BAMs and using GERMLINE as default.

In this step, if “SOMATIC” is selected, an additional BQSR and without properly paired BAM will be generated for each sample. It also provides the possibilities to any build if BQSR not needed.

For all projects, will generate BAM files in /DCEG/Projects/Exome/SequencingData/BAM_reformatted/BAM_recalibrated/${Disease_group}

E.g., /DCEG/Projects/Exome/SequencingData/BAM_reformatted/BAM_recalibrated/EC/EC_EC_TYE00905a_germline.bam

For the SOMATIC project, will also generate BQSR without properly paired BAM as

*_bqsr_final_out.bam in the same folder.

Note that because any old pipeline does not do somatic separately, for all new future somatic build if using old samples from previous build, “redo” is required in previous step1_run_parse.sh. that means will generate all new merged BAMs in incoming and BAM_original folders. Then, can make step8 redo all recalibration for those BAMs according to timestamp.

The recalibrate_bam.sh can be adjusted in case want to remove the BQSR step only skip properly-paired filtering step.

15. step9_construct_BAM_recaliberated_per_manifest.sh $Manifest_File $Build_name SOMATIC|GERMLINE to generate soft links for a build

Others:

The feature that checking the consistency between the manifest and actual merging records:

This feature is based on merged BAM’s records: read IDs, PU and manifest information. It includes two steps:

1) sh ./run_misc_step1_check_BAMs.sh $BUILD_FOLDER

e.g., sh ./run_misc_step1_check_BAMs.sh /DCEG/Projects/Exome/builds/build_SR0471-001-NRTIs-Sequencing_2018_22331

This script will scan all merged BAMs (in bam_location) and output the scanned records to

/DCEG/Projects/Exome/SequencingData/secondary_buf/output/bam_results_${BUILD_NAME}.

And copy the manifest to that folder as well.

The logs are placed in
/DCEG/Projects/Exome/SequencingData/secondary_buf/logs/parse_${BUILD_NAME}

Check have error for each BAM

```
grep Error /DCEG/Projects/Exome/SequencingData/secondary_buf/logs/parse_build_SR0493-001_Data_Delivery_2019_22641/*.stdout
```

2) sh ./ run_misc_step2_merge_BAM_output.sh $BUILD_FOLDER

The match report will be showed in the **report_cmp.txt** in
/DCEG/Projects/Exome/SequencingData/secondary_buf/output/bam_results_${BUILD_NAME}.

And related **err_cmp.txt** in the same folder.

3) For downsample ratios specified in the Manifest file, please run **run_misc_downsample_BAMs.sh**