Best practices for processing the data of SureSelect XT HS, SureSelect XT HS2 DNA and RNA, and SureSelect Max and Max MGI prior to variant discovery

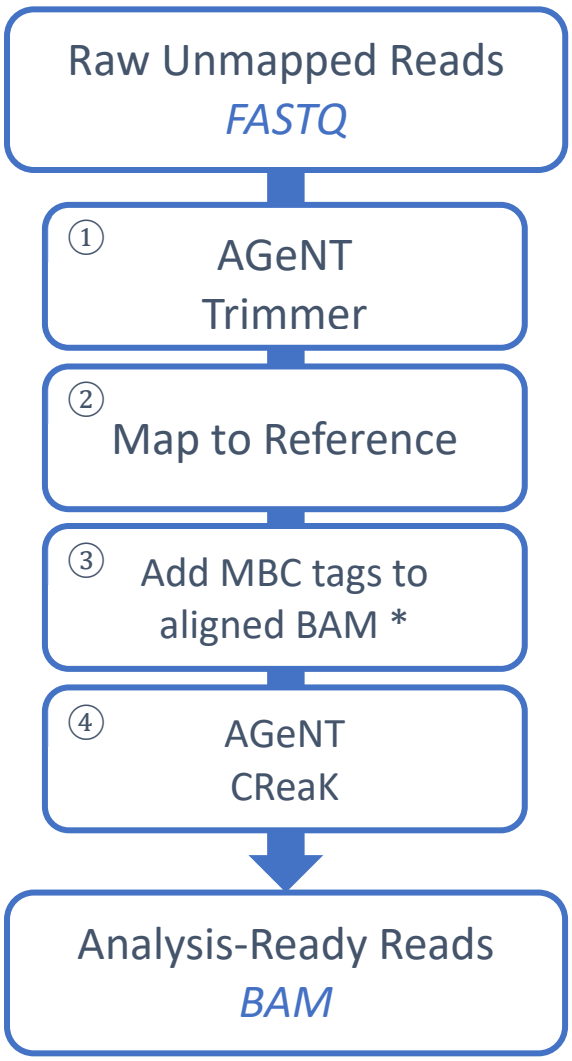
Purpose:

Prior to variant discovery, raw sequencing data must be pre-processed to correct for technical biases and align to a reference genome. For SureSelect XT HS, XT HS2, Max and Max MGI data, this process involves:

1. pre-processing the raw reads to remove sequencing adaptors and extract the molecular barcode information.
2. aligning the processed reads to the reference
3. annotating the aligned file with the molecular barcode information
4. PCR de-duplication leveraging the molecular barcode information.

We recommend using the combination of AGeNT Trimmer + AGeNT CReaK tools for running steps 1 and 4. Step 3 could be automatically handled by some aligners (like BWA-MEM and STAR) with the right setting/options.

Basic workflow using the **A**gilent **Ge**nomics **N**extGen **T**oolkit (AGeNT)



* NOTE: Step 3 is only necessary when using an aligner that can't add the MBC tags to the Aligned BAM file itself. Some aligners (like BWA-MEM and STAR) can handle this step as part of the alignment process.

Input:
This workflow operates on raw Illumina or MGI sequencing data in FASTQ format. The data is expected to be demultiplexed, **but not adaptor trimmed**.

Steps:

1. Trim adaptor sequences and extract or process MBCs.
   AGeNT Trimmer processes the reads in pairs, using the overlap between the read pairs as well as the library prep specific sequencing adaptor sequences to trim any adaptors from the 3' end of the reads. The sequencing adaptors could be specified by predefined names (input parameter "-adaptor") or specific sequences (input option "-customAdaptor").

   Example invocation (on Linux/mac) for Illumina SureSelect XT HS2 or Max:

```
agent.sh trim –adaptor IlluminaXT –mbc xths2 -fq1
  /path/to/fastq_input_dir/sample_R1.fastq.gz -fq2
  /path/to/fastq_input_dir/sample_R2.fastq.gz -out
  myOutputDirPath/myOutputFilePrefix
```

   Example invocation (on Linux/mac) for SureSelect Max MGI:

```
agent.sh trim –adaptor MGI –mbc xths2 -fq1
  /path/to/fastq_input_dir/sample_R1.fastq.gz -fq2
  /path/to/fastq_input_dir/sample_R2.fastq.gz -out
  myOutputDirPath/myOutputFilePrefix
```

   For dual-MBC kits such as SureSelect XT HS2 and Max, Trimmer extracts the molecular barcode information from reads into SAM spec tags and places these tags in the read name header for easier propagation of the MBC to the final aligned file. The MBC could be specified by predefined names (such as "-mbc xths2") or specific patterns ("-mbc YourPatternString", see more in README on how to specify the pattern).

   In the case of single-MBC kit such as SureSelect XT HS, the MBC sequence is read from the provided MBC FASTQ file (with input parameter "-fq3") if the MBC-based deduplication is required in the downstream analysis.

   Example invocation (on Linux/mac) for SureSelect XT HS:

```
agent.sh trim –adaptor IlluminaXT -fq1
  /path/to/fastq_input_dir/sample_R1.fastq.gz -fq2
  /path/to/fastq_input_dir/sample_R2.fastq.gz -fq3
  /path/to/fastq_input_dir/sample_MBC.fastq.gz -out
  myOutputDirPath/myOutputFilePrefix
```

Trimmer can output trimmed reads in either FASTQ format (default) or unaligned BAM format (with option "-bam"). When it outputs FASTQ files, it also creates a FASTQ-like txt file containing just the MBC sequences for data of dual-MBC kits such as SureSelect XTHS2 and Max.

The output FASTQ files from Trimmer have SAM style tags added to the read name headers for the barcode and MBC information.
For example:

```
@D00266:1113:HTWK5BCX2:1:1102:9976:2206 BC:Z:CTACCGAA+AAGTGTCT
  ZA:Z:TTAGT ZB:Z:TCCT RX:Z:TTA-TCC QX:Z:DDD DDA
```

List of tags

Tag	Type	Description
BC	Z	Sample barcode
RX	Z	Two MBC sequences (concatenated with "-")
QX	Z	Base quality representation of the MBCs (concatenated with space)
ZA	Z	3 MBC bases for read 1 followed by 1 or 2 dark bases
ZB	Z	3 MBC bases for read 2 followed by 1 or 2 dark bases

NOTE: If the first 4/5 bases are not recognized as a valid molecular barcode, they are masked with "**N**" and the corresponding base qualities are marked as "**$**". This allows downstream filtering of these reads.

The output unaliged BAM file from Trimmer has the barcode and MBC information as SAM tags. These SAM tags could be properly propagated into aligned BAM files with the right options for RNA aligner like STAR.
For example:

```
D00266:1113:HTWK5BCX2:1:1102:2339:2181    141   *     0     0     *
      *     0     0
      ATATGTCTTCCACCCTTGACTCTCAGGATAGTGCAGCAGGATGCAGCGGAACCCCCCCGATTTGC
  CGACAAGCCCAGACCAAACTAGAAGTGGCC
      IIIIHIIIIIIIIIIIIHIHHIIHIEHHHHIHIHHHIHHIIIIIGIIIIHHIIHHIIIIIIIIII
  HIIIIHFHIIHHHGIIIIIHHHHIIGHHII    ZA:Z:GAACT  ZB:Z:GAACT
      BC:Z:CTACCGAA+AAGTGTCT  QX:Z:DDD DDD      RX:Z:GAA-GAA
```

2. Align the trimmed reads.

BWA-MEM is strongly recommended because it contains an option that will easily propagate the SAM tags from the FASTQ read names to the final aligned BAM file (see example in Step 3). For RNAseq data, any aligner designed for RNA data should work.

But STAR is recommended because 1) It has options that can populate the SAM tags in the unaligned bam file from Trimmer to aligned bam file; 2) CReaK was tested with output from STAR.

3. Add MBC tags to the aligned file.

   If using BWA-MEM, the "-C" option will append the FASTQ comment from the read header to the SAM output.

   BWA-MEM example:

```
bwa mem -C -t 2 /hg38.fa trimmed_dir/sample_R1.cut.fastq.gz \
trimmed_dir/sample_R2.cut.fastq.gz | \
samtools view -b - > aligned_dir/sample.bam
```

   If using STAR, the option will append the FASTQ comment from the read header to the SAM output.

   STAR example (with bare minimum parameters to successfully pipe with AGeNT Trimmer and CReaK):

```
STAR --runMode alignReads --genomeDir /Path/To/GenomeDir \
--readFilesType SAM PE --readFilesCommand samtools view \
--readFilesIn /Path/To/Trimmed_unaligned.bam \
--outFileNamePrefix Your_Outputfile_prefix \
--outSAMtype BAM Unsorted \
--outSAMattributes All
```

   Note in STAR, "--outSAMattributes All" should be used to enable the generation of a few standard SAM tags like "NM:i:<n>" required by CReaK.

   If using an aligner that does not propagate SAM tags of the MBC information from either FASTQ file or unaligned BAM file, it is necessary to annotate the aligned BAM file with the RX and QX MBC tags listed in step 1.

   a. Create an unaligned BAM

      - Method 1: use Trimmer with output option "-bam"

        The "-bam" option for AGeNT Trimmer will output unaligned BAM (uBAM) rather than FASTQ.

      - Method 2: Use a 3rd party tool to convert FASTQ files into unaligned BAM

        For example, "samtools import" with the R1 and R2 FASTQ files from Trimmer and the "-T" input parameter will create an unaligned BAM using the BAM tags in the read name headers of the FASTQ files.

        Example:

```
samtools import -1 trimmed_dir/sample_R1.cut.fastq.gz \
-2 trimmed_dir/sample_R2.cut.fastq.gz -T '*' \
-o sample.unaligned.bam
```

b. Sort the uBAM and aligned BAM by read name and merge the two together

A tool such as “picard MergeBamAlignment” can be used to carry the BAM tags in the uBAM over to the aligned BAM file. This tool additionally runs several validations and cleanup on the aligned BAM to create a Picard/GATK friendly BAM file. To disable these modifications, run the tool with these options:

- --CLIP_ADAPTORS false
- --CLIP_OVERLAPPING_READS false
- --ALIGNER_PROPER_PAIR_FLAGS true
- --ADD_PG_TAG_TO_READS false
- --ADD_MATE_CIGAR false
- --MAX_INSERTIONS_OR_DELETIONS -1
- --ATTRIBUTES_TO_REMOVE RG

4. Generate consensus reads

The AGeNT CReaK tool is used to generate consensus reads using the MBCs. The tool has been tested with datasets containing up to 70 M read pairs. By default, this tool generates a file containing all the input reads, with duplicate reads flagged as *read is PCR or optical duplicate* (SAM flag 0x400) and filtered reads flagged as *read fails platform/vendor quality checks* (SAM flag 0x200). If desired, the tool can instead generate files with duplicate (SAM flag 0x400), filtered (SAM flag 0x200), secondary (SAM flag 0x100) and supplementary (SAM flag 0x800) reads all removed (using the “-r” parameter). Alternately, a third-party tool such as “samtools view” can be used to remove the marked reads.

NOTE: The options below for duplex and hybrid consensus modes are relevant for DNA workflows only. Unlike DNA, where both strands are present and the MBCs in the strands can be matched to form a duplex consensus read, **single-stranded SureSelect RNA XT HS2 library prep stops at single consensus generation.**

Example invocation (on linux/mac) for DNA:

```
agent.sh -Xmx12G creak -c HYBRID -b Covered.bed -F -f -mm 25 -o
  deduped_dir/sample.bam aligned_dir/sample.bam
```

Example invocation (on linux/mac) for RNA:

```
agent.sh -Xmx12G creak -c SINGLE -b Covered.bed -F -f -mm 25 -o
  deduped_dir/sample.bam aligned_dir/sample.bam
```

For SureSelect XT HS2 data, CReaK can run in 3 consensus generation modes:

Mode	Command-line option	Description
Single-strand consensus	`--consensus-mode (or -c) SINGLE`	Ignores strand information and treats the duplex MBC as a single MBC.
Duplex consensus	`--consensus-mode (or -c)DUPLEX`	Requires at least 1 read for each strand (reads where there is only 1 strand support for the MBC family are flagged as *read fails platform/vendor quality checks* (SAM flag 0x200)).
Hybrid consensus	`--consensus-mode (or -c)HYBRID`	This approach creates DUPLEX consensus reads when both strands are present and SINGLE consensus reads when only 1 strand is present in the MBC family. Each duplex consensus read is written *twice* to the output file to ensure that these reads are weighted properly when compared with the retained single consensus reads. The read names for the two duplex consensus reads match the read names for the two single consensus reads that were used to generate it.

For a full list of input parameters and definitions, please refer to CReaK README.

The consensus reads in the output BAM files will have additional tags added with annotations associated with filter settings as well as original read information for traceability of the consensus generation process. Please refer to the CReaK README for more detailed information on these tags.

Output:

Once CReaK consensus generation is complete, the resulting BAM files are ready for analysis with any downstream analysis tools, such as variant callers.