Arima-HiC Mapping Pipeline

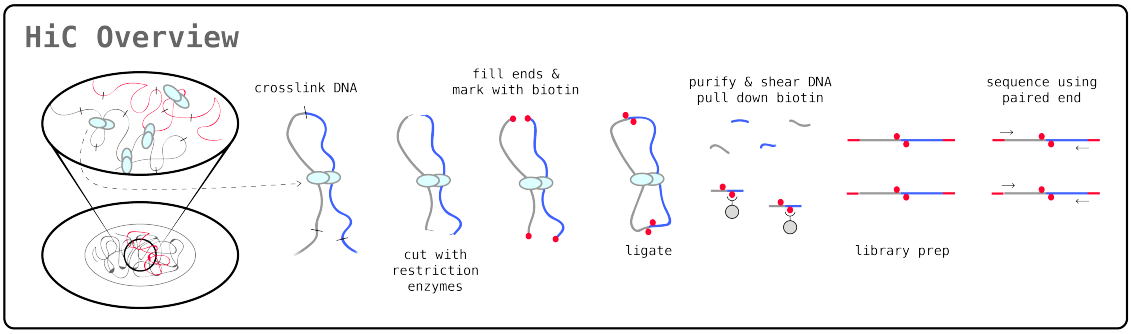
Arima Genomics, Inc.

December 6, 2017

For questions or inquiries about other applications please contact:
Anthony Schmitt - anthony@arimagenomics.com

Workflow Co-Developed With:
Bing Ren Lab - *Ludwig Institute for Cancer Research*

HiC Overview



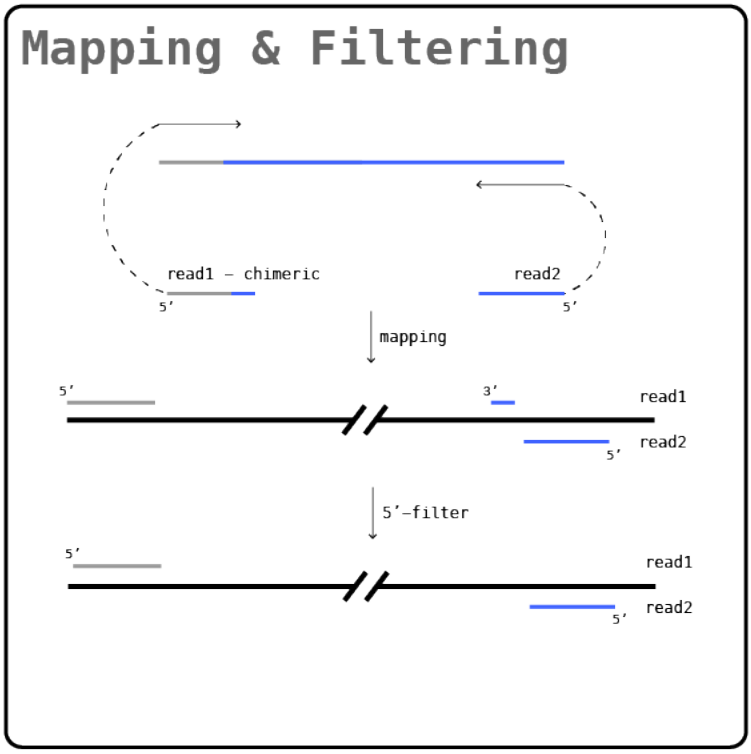
The workflow is meant to assist in *mapping* HiC paired-end reads (e.g. obtained via Arima-HiC kits or services) to reference sequences.

The Arima-HiC kits and services utilize an experimental protocol that captures inherent three-dimensional (3D) conformation of the genome. The HiC libraries are subjected to Illumina short-read sequencing in "paired-end" mode and the resulting data is referred to as HiC paired-end reads. When HiC paired-end reads are mapped to reference sequences, conformational information manifested in the HiC procedure can be used to generate chromosome-span contiguity.

In the Arima-HiC methodology, a sample (cells or tissues) is first crosslinked to preserve the genome conformation. The crosslinked DNA is then digested using restriction enzymes. The single-stranded 5'-overhangs are then filled in causing digested ends to be labeled with a biotinylated nucleotide. Next, spatially proximal digested ends of DNA are ligated, preserving both short- and long-range DNA contiguity. The DNA is then purified and sheared to a size appropriate for Illumina short-read sequencing. After shearing, the biotinylated fragments are enriched to assure that only fragments originating from ligation events are sequenced in paired-end mode via Illumina sequencers to inform DNA contiguity. See figure above.

Below are the necessary steps to map HiC paired-end reads (FASTQ format) to reference sequences.

Mapping Pipeline



Input – Illumina paired-end FASTQ
Output – BAM file

In this section, we describe specific steps to map HiC paired-end reads to reference sequences. This mapping procedure also includes steps to filter HiC reads to correct for erroneous mapping that can confound downstream analyses. See figure above.

The mapping pipeline will output a single binary alignment map file (BAM file) that contains paired and filtered HiC paired-end reads mapped to reference sequences. Below, we walk through an example of our mapping pipeline.

The first section of the pipeline defines the paths to the files and scripts needed to run our pipeline. For the mapping pipeline, you will need the software BWA, SAMtools, and Picard Tools installed on your system. You also need the scripts "filter_five_end.pl" and "two_read_bam_combiner.pl" that are provided. Please change the file paths, file names, and label names as appropriate. Note that the $REF and $FAIDX variables correspond to your reference sequence FASTA file and indexed reference sequence FASTA file, respectively.

```
#! /bin/bash

SRA='basename_of_fastq_files'
LABEL='overall_exp_name'
BWA='/software/bwa/bwa-0.7.12/bwa'
SAMTOOLS='/software/samtools/samtools-1.3.1/samtools'
IN_DIR='/path/to/gzipped/fastq/files'
REF='/path/to/reference_sequences/reference_sequeneces.fa'
FAIDX='$REF.fai'
RAW_DIR='/path/to/write/out/bams'
FILT_DIR='/path/to/write/out/filtered/bams'
FILTER='/path/to/filter_five_end.pl'
COMBINER='/path/to/two_read_bam_combiner.pl'
PICARD='/software/picard/picard-2.6.0/build/libs/picard.jar'
TMP_DIR='/path/to/write/out/temporary/files'
PAIR_DIR='/path/to/write/out/paired/bams'
REP_DIR='/path/to/where/you/want/deduplicated/files'
REP_LABEL=$LABEL\_rep1
MERGE_DIR='/path/to/final/merged/alignments/from/any/biological/replicates'
MAPQ_FILTER=10

echo "### Step 0: Check output directories exist & create them as needed"
[ -d $RAW_DIR ] || mkdir -p $RAW_DIR
[ -d $FILT_DIR ] || mkdir -p $FILT_DIR
[ -d $TMP_DIR ] || mkdir -p $TMP_DIR
[ -d $PAIR_DIR ] || mkdir -p $PAIR_DIR
[ -d $REP_DIR ] || mkdir -p $REP_DIR
[ -d $MERGE_DIR ] || mkdir -p $MERGE_DIR
```

Next, we use BWA-MEM to align the HiC paired-end reads to reference sequences. Because HiC captures conformation via proximity-ligated fragments, paired-end reads are first mapped independently (as single-ends) using BWA-MEM and are subsequently paired in a later step.

```
echo "### Step 1.A: FASTQ to BAM (1st)"
```

```
$BWA mem -t 12 -B 8 $REF $IN_DIR/$SRA\_1.fastq.gz | $SAMTOOLS view -Sb - >
$RAW_DIR/$SRA\_1.bam

echo "### Step 1.B: FASTQ to BAM (2nd)"
$BWA mem -t 12 -B 8 $REF $IN_DIR/$SRA\_2.fastq.gz | $SAMTOOLS view -Sb - >
$RAW_DIR/$SRA\_2.bam
```

Subsequent to mapping as single-ends, some of these single-end mapped reads can manifest a ligation junction and are therefore considered "chimeric" (i.e. they do not originate from a contiguous piece of DNA). When BWA-MEM maps these chimeric reads, there can be high quality mapping on both the 5'-side and 3'-side of the ligation junction within a given read. In such cases, only the 5'-side should be retained because the 3'-side can originate from the same contiguous DNA as the 5'-side of the reads mate-pair. Therefore, we retain only the portion of the chimeric read that maps in the 5'-orientation in relation to its read orientation. This is accomplished using the script "filter_five_end.pl."

```
echo "### Step 2.A: Filter 5' end (1st)"
$SAMTOOLS view -h $RAW_DIR/$SRA\_1.bam | perl $FILTER | $SAMTOOLS view -Sb -
> $FILT_DIR/$SRA\_1.bam

echo "### Step 2.B: Filter 5' end (2nd)"
$SAMTOOLS view -h $RAW_DIR/$SRA\_2.bam | perl $FILTER | $SAMTOOLS view -Sb -
> $FILT_DIR/$SRA\_2.bam
```

After filtering, we pair the filtered single-end HiC reads using "two_read_bam_combiner.pl," which outputs a sorted, mapping quality filtered, paired-end BAM file. We then add read groups to this BAM file using Picard Tools.

```
echo "### Step 3A: Pair reads & mapping quality filter"
perl $COMBINER $FILT_DIR/$SRA\_1.bam $FILT_DIR/$SRA\_2.bam $SAMTOOLS
$MAPQ_FILTER | $SAMTOOLS view -bS -t $FAIDX - | $SAMTOOLS sort -o
$TMP_DIR/$SRA.bam -

echo "### Step 3.B: Add read group"
java -Xmx2g -jar $PICARD AddOrReplaceReadGroups INPUT=$TMP_DIR/$SRA.bam
OUTPUT=$PAIR_DIR/$SRA.bam ID=$SRA LB=$SRA SM=$LABEL PL=ILLUMINA PU=none
```

We also use Picard Tools to discard any PCR duplicates present in the paired-end BAM file generated above. If applicable, we require that you merge paired-end BAM files that were sequenced via multiple Illumina lanes from the same library (i.e. technical replicates) before removing PCR duplicates. Below is example code for how to accomplish this merging step.

```
################################################################################
### How to Accommodate Technical Replicates
### This pipeline is currently built for processing a single sample with
### one read1 and read2 fastq file.
### Technical replicates (eg. one library split across multiple lanes) should
### be merged before running the MarkDuplicates command.
### If this step is run, the names and locations of input files to subsequent
### steps will need to be modified in order for subsequent steps to run
### correctly.
### The code below is an example of how to merge technical replicates.
################################################################################
REP_NUM=X #number of the technical replicate set e.g. 1
REP_LABEL=$LABEL\_rep$REP_NUM
INPUTS_TECH_REPS=('bash' 'array' 'of' 'bams' 'from' 'replicates') #BAM files
you want combined as technical replicates

#example bash array -
#INPUTS_TECH_REPS=('INPUT=A.L1.bam' 'INPUT=A.L2.bam' 'INPUT=A.L3.bam')

java -Xms4G -Xmx4G -jar $PICARD MergeSamFiles $INPUTS_TECH_REPS
OUTPUT=$TMP_DIR/$REP_LABEL.bam USE_THREADING=TRUE ASSUME_SORTED=TRUE
VALIDATION_STRINGENCY=LENIENT
```

Note, that if you preform merging of technical replicates above, then the file names and locations will change from the written flow of this pipeline. You will need to adjust the file names and locations that are used as input in the following step - PCR duplicate removal.

```
echo "### Step 4: Mark duplicates"
java -Xms24G -XX:-UseGCOverheadLimit -Xmx24G -jar $PICARD MarkDuplicates
INPUT=$PAIR_DIR/$SRA.bam OUTPUT=$REP_DIR/$REP_LABEL.bam
METRICS_FILE=$REP_DIR/metrics.$REP_LABEL.txt TMP_DIR=$TMP_DIR
ASSUME_SORTED=TRUE VALIDATION_STRINGENCY=LENIENT REMOVE_DUPLICATES=TRUE

$SAMTOOLS index $REP_DIR/$REP_LABEL.bam

echo "Finished Mapping Pipeline"
```

At this point in the pipeline, if you have two or more libraries prepared from the same sample (i.e. biological replicates), the biological replicate paired-end BAM files should be merged prior to subsequent analyses. Below is example code for how to accomplish this merging step.

```
################################################################################
### How to Accommodate Biological Replicates
### This pipeline is currently built for processing a single sample
```

```
### with one read1 and read2 fastq file.
### Biological replicates (eg. multiple libraries made from the same
### sample) should be merged before proceeding with subsequent steps.
### The code below is an example of how to merge biological replicates.
##############################################################################

INPUTS_BIOLOGICAL_REPS=('bash' 'array' 'of' 'bams' 'from' 'replicates') #BAM
files you want combined as biological replicates
#example bash array -
#INPUTS_BIOLOGICAL_REPS=('INPUT=A_rep1.bam' 'INPUT=A_rep2.bam'
'INPUT=A_rep3.bam')

java -Xms4G -Xmx4G -jar $PICARD MergeSamFiles $INPUTS_BIOLOGICAL_REPS
OUTPUT=$MERGE_DIR/$LABEL.bam USE_THREADING=TRUE ASSUME_SORTED=TRUE
VALIDATION_STRINGENCY=LENIENT

$SAMTOOLS index $MERGE_DIR/$LABEL.bam
```

The final output of this pipeline is a single BAM file that contains the paired, 5'-filtered, and duplicate-removed HiC reads mapped to the reference sequences of choice.