Arima-HiC Mapping Pipeline

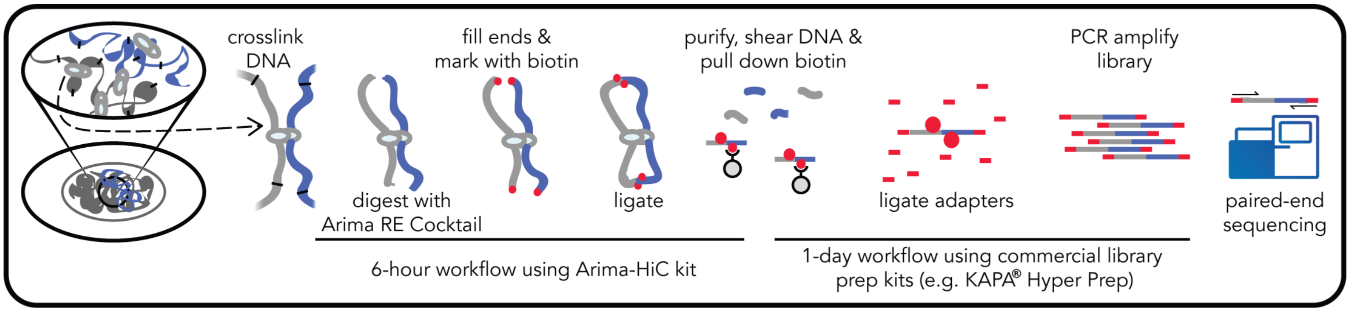
Document Part Number: A160156 v03
Release Date: Jan 2024

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

Revision History

Document	Date	Description of Change
Document Part Number: A160156 v03	Jan 2024	□ Added a section for trimming 5 bases from the 5’ end of both read 1 and read 2 when using Arima Hi-C library prep kit.
Document Part Number: A160156 v02	May 2019	□ Removed last paragraph containing duplicated sentences.
Document Part Number: A160156 v01	February 2019	□ Added support for multi-threading □ Optimized memory allocation for Java □ Added option to index the FASTA file using BWA
Document Part Number: A160156 v00	November 2018	Initial release.

Hi-C Overview


crosslink DNA
digest with Arima RE Cocktail
fill ends & mark with biotin
ligate
purify, shear DNA & pull down biotin
ligate adapters
PCR amplify library
paired-end sequencing
6-hour workflow using Arima-HiC kit
1-day workflow using commercial library prep kits (e.g. KAPA® Hyper Prep)

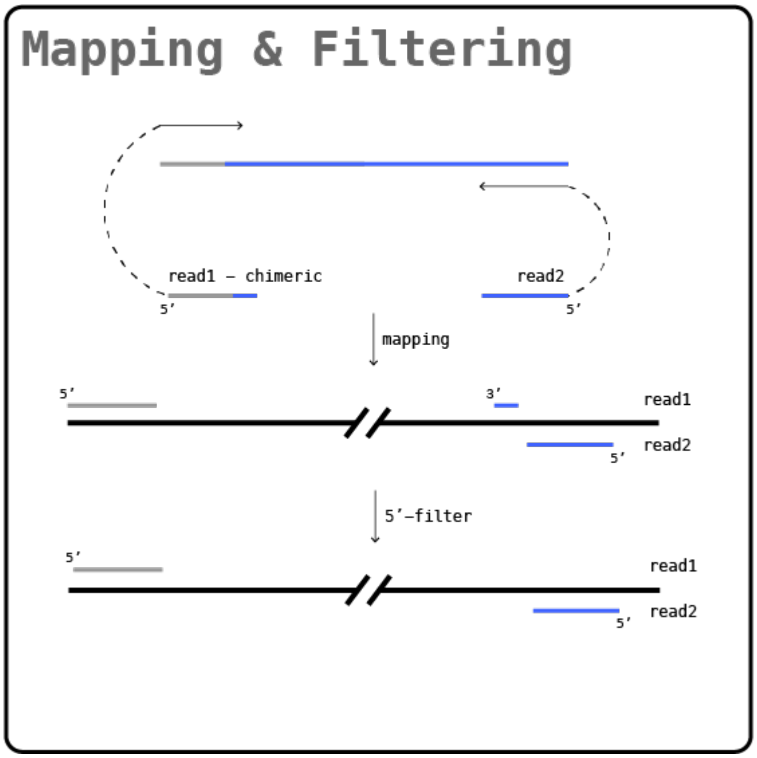
The workflow is meant to assist in *mapping* Hi-C paired-end reads (e.g. obtained via Arima-HiC kits or services) to reference sequences.

The Arima-HiC kits and services utilize an experimental protocol that captures inherent three-dimensional (3D) conformation of the genome. The Hi-C libraries are subjected to Illumina short-read sequencing in "paired-end" mode and the resulting data is referred to as Hi-C paired-end reads. When Hi-C paired-end reads are mapped to reference sequences, conformational information manifested in the Hi-C 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, this document describes the necessary steps to map Hi-C 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 Hi-C paired-end reads to reference sequences. This mapping procedure also includes steps to filter Hi-C 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 Hi-C 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, scripts, and output directories utilized by our pipeline, and then creates any of those output directories that do not already exist. 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='$REF.fai'
PREFIX='bwa_index_name'
RAW_DIR='/path/to/write/out/bams'
FILT_DIR='/path/to/write/out/filtered/bams'
FILTER='/path/to/filter_five_end.pl'
COMBINER='/path/to/two_read_bam_combiner.pl'
STATS='/path/to/get_stats.pl'
PICARD='/software/picard/picard-2.6.0/build/libs/picard.jar'
TMP_DIR='/path/to/write/out/temporary/files'
PAIR_DIR='/path/to/write/out/paired/bams'
REP_DIR='/path/to/where/you/want/deduplicated/files'
REP_LABEL=${LABEL}_rep1
MERGE_DIR='/path/to/final/merged/alignments/from/any/biological/replicates'
MAPQ_FILTER=10
CPU=12

echo "### Step 0: Check output directories’ existence & 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

# Run only once! Skip this step if you have already generated BWA index
files
echo "### Step 0: Index reference"
$BWA index -a bwtsw -p $PREFIX $REF
```

Note: When utilizing libraries generated with the Arima Hi-C library prep kit, we suggest trimming 5 bases from the 5’ end of both read 1 and read 2 before using the Arima mapping pipeline.

The purpose of this modification is to remove over-represented 3bp molecular barcode UMI sequences and 2 dark bases from the 5’ end of reads, resulting in improved results.

For chromatin conformation applications, such as genome-wide Hi-C and Capture Hi-C, trimming of bases is typically unnecessary when using the recommended Arima Capture Hi-C pipeline, Arima SV pipeline, and the Juicer pipeline as outlined in the Arima Genomics Hi-C Bioinformatics User Guide. However, it is generally good practice to trim 5' end adapter sequences, which can enhance mapping rates and potentially benefit other Hi-C analysis pipelines.

This can be achieved using tools like Cutadapt, Trimmomatic, fastp, or a simple Shell script provided below:

```
# Run once for each FASTQ file! Skip this step if your files are NOT prepared
with the Arima Hi-C library prep kit!
echo "### Step 0: Trimming the reads"
zcat INPUT.fastq.gz | awk '{ if(NR%2==0) {print substr($1,6)} else {print}
}' | gzip > OUTPUT.fastq.gz
```

Next, we use BWA-MEM to align the Hi-C paired-end reads to reference sequences. Because Hi-C 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 $CPU $REF $IN_DIR/${SRA}_1.fastq.gz | $SAMTOOLS view -@ $CPU -Sb
- > $RAW_DIR/${SRA}_1.bam

echo "### Step 1.B: FASTQ to BAM (2nd)"
$BWA mem -t $CPU $REF $IN_DIR/${SRA}_2.fastq.gz | $SAMTOOLS view -@ $CPU -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 Hi-C 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 -@ $CPU -o
$TMP_DIR/$SRA.bam -
```

```
echo "### Step 3.B: Add read group"
java -Xmx4G -Djava.io.tmpdir=temp/ -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 a
# 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 -Xmx8G -Djava.io.tmpdir=temp/ -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 perform 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 -Xmx30G -XX:-UseGCOverheadLimit -Djava.io.tmpdir=temp/ -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

perl $STATS $REP_DIR/$REP_LABEL.bam > $REP_DIR/$REP_LABEL.bam.stats

echo "Finished Mapping Pipeline through Duplicate Removal"
```

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. If you do not have biological replicates, then the pipeline is complete. The final output of this pipeline is a single BAM file that contains the paired, 5'-filtered, and duplicate-removed Hi-C reads mapped to the reference sequences of choice. The resulting statistics file has a breakdown of the total number of intra-contig read-pairs, long-range intra-contig read-pairs, and inter-contig read-pairs in the final processed BAM file.

```
############################################################################
# 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 -Xmx8G -Djava.io.tmpdir=temp/ -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
#
# perl $STATS $MERGE_DIR/$LABEL.bam > $MERGE_DIR/$LABEL.bam.stats
#
# echo "Finished Mapping Pipeline through merging Biological Replicates"
```