Summer 2020

Unix/Linux for Informatic Analysis
https://github.com/sabrsyed/InformaticsTools_2020

Week 1

- Unix Primer for Biologists: Chapters U1 – U16
    - Learn how to use UNIX/Linux
- Logging in to the Cluster
    - Learn to navigate the Cluster
- Powerpoint Presentation: learn the technology behind genome sequencing, what does ChIP-Seq data look like
- Pipeline for ChIP alignment
- IGV and UCSC for viewing .bedgraph and .bam files

Tasks:
Install IGV and view Mtf1 ChIP-Seq .bedgraph files
Identify modules available on Cottontail

Week 2

- Unix Primer for Biologists: Chapters U17 – U34
    - Learn how to use UNIX/Linux
- GEO Omnibus for publicly available datasets
- https://www.ebi.ac.uk/ena for FASTQ files
- Filezilla
    - Uploading files to the Cluster
- Powerpoint Presentation: RNA-Sequencing, what does RNA-Seq data look like
- Pipeline for RNA-Seq alignment

Optional Exercise:
https://github.com/sabrsyed/InformaticsTools_2020/blob/master/01_Unix_QuickReview_ProblemSet.md

Week 3

- Unix Problem Set Q&A
- Review ChIP-Seq and RNA-Seq pipelines

- Loading files to the cluster using FileZilla and rsync

- On the ghpcc06 cluster, read criteria for running bowtie2, picard, samtools, macs2, etc.
- See sample bsub commands for running each module
- Look at bowtie2 and macs2 output files
- Use IGV to view .bam and .bed files

- On the ghpcc06 cluster, read criteria for running rsem-calculate-expression on the cluster
  https://bioinfo.umassmed.edu/index.php?p=33
- RNA-Seq – Look at RSEM output files
- See sample bsub commands for rsem-calculate-expression

Tasks:
Create lab schedule for using the ghpcc06 cluster
Explore the cluster and examine modules and tools for data analysis that are available
Upload Fastq files to ghpcc06 cluster

Week 4

- Introducing NGSplot: Quick mining and visualization of NGS data by integrating genomic databases
  https://github.com/shenlab-sinai/ngsplot
- Introducing HOMER: Software for motif discovery and next-gen sequencing analysis
  http://homer.ucsd.edu/homer/ngs/

- Running NGSplot on the ghpcc06 cluster for making ChIP-Seq and RNA-Seq heatmaps and extracting tag density files
- Running HOMER on the ghpcc06 cluster for annotating bed files and performing motif analysis
- Running bedtools on the ghpcc06 cluster for comparing .bed files of different experiments

Problem Set

1. Log into your machine.
2. What is the full path to your home directory?
3. Go up one directory?
    - How many files does it contain?
    - How many directories?
4. Make a directory called `problemsets`.
5. Navigate into this new directory called problemsets. Verify that you are in the correct directory by using `pwd`.
6. Use `wget` to copy https://raw.githubusercontent.com/sabrsyed/pfb2017/master/files/seq.nt.fa from the web into your problemsets directory. If `wget` is not available on your system, use `curl -O` as an alternative.
7. Without using a text editor calculate or report these qualities for the file `sequences.nt.fa`. This file can be found here https://raw.githubusercontent.com/sabrsyed/pfb2017/master/files/sequences.nt.fa
    - How many lines does this file contain?
    - How many characters? (Hint: check out the options of wc)
    - What is the first line of this file? (Hint: read the man page of head)
    - What are the last 3 lines? (Hint: read the man page of tail)
    - How many sequences are in the file? (Hint: use grep) (Note: The start of a sequence is indicated by a `>` character.)

NAME

wc — word, line, character, and byte count

SYNOPSIS

wc [**--libxo**] [**-Lclmw**] [*file* **. . .**]

DESCRIPTION

The **wc** utility displays the number of lines, words, and bytes contained in each input *file*, or standard input (if no file is specified) to the standard output. A line is defined as a string of characters delimited by a ⟨newline⟩ character. Characters beyond the final ⟨newline⟩ character will not be included in the line count.

A word is defined as a string of characters delimited by white space characters. White space characters are the set of characters for which the `iswspace`(3) function returns true. If more than one input file is specified, a line of cumulative counts for all the files is displayed on a separate line after the output for the last file.

The following options are available:

--libxo
: Generate output via `libxo`(3) in a selection of different human and machine readable formats. See `xo_parse_args`(3) for details on command line arguments.

-L
: Write the length of the line containing the most bytes (default) or characters (when **-m** is provided) to standard output. When more than one *file* argument is specified, the longest input line of *all* files is reported as the value of the final "total".

-c
: The number of bytes in each input file is written to the standard output. This will cancel out any prior usage of the **-m** option.

-l
: The number of lines in each input file is written to the standard output.

-m
: The number of characters in each input file is written to the standard output. If the current locale does not support multibyte characters, this is equivalent to the **-c** option. This will cancel out any prior usage of the **-c** option.

-w
: The number of words in each input file is written to the standard output.

When an option is specified, **wc** only reports the information requested by that option. The order of output always takes the form of line, word, byte, and file name. The default action is equivalent to specifying the **-c**, **-l** and **-w** options.

If no files are specified, the standard input is used and no file name is displayed. The prompt will accept input until receiving EOF, or [^D] in most environments.

Week 3

- Unix Problem Set Q&A
- Review ChIP-Seq and RNA-Seq pipelines

- Loading files to the cluster using FileZilla and rsync

- On the ghpcc06 cluster, read criteria for running bowtie2, picard, samtools, macs2, etc.
- See sample bsub commands for running each module
- Look at bowtie2 and macs2 output files
- Use IGV to view .bam and .bed files

- On the ghpcc06 cluster, read criteria for running rsem-calculate-expression on the cluster
  https://bioinfo.umassmed.edu/index.php?p=33
- RNA-Seq – Look at RSEM output files
- See sample bsub commands for rsem-calculate-expression

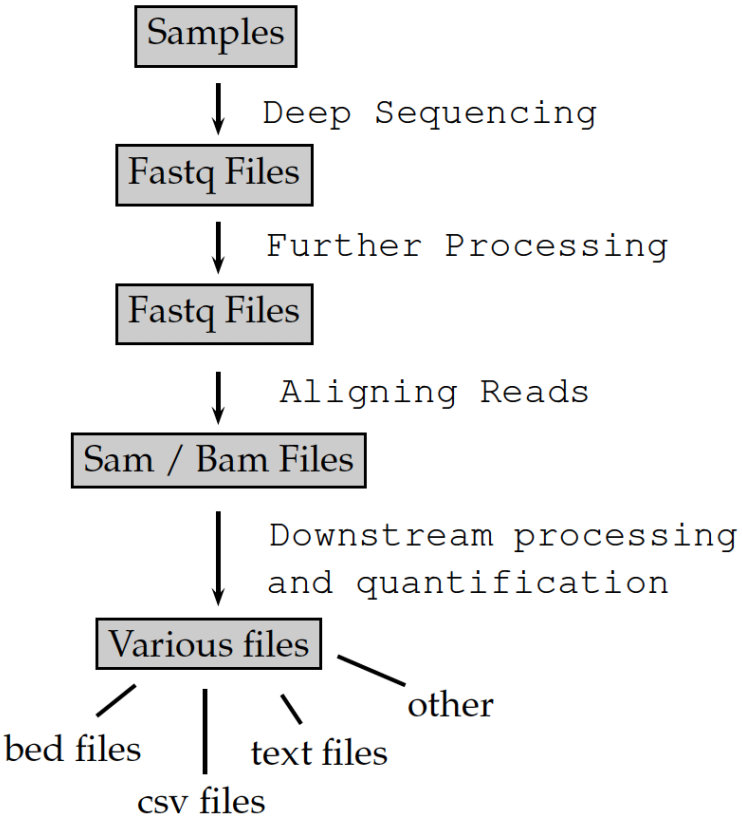
Tasks:
Create lab schedule for using the ghpcc06 cluster
Explore the cluster and examine modules and tools for data analysis that are available
Upload Fastq files to ghpcc06 cluster

Week 4

- Introducing NGSplot: Quick mining and visualization of NGS data by integrating genomic databases
  https://github.com/shenlab-sinai/ngsplot
- Introducing HOMER: Software for motif discovery and next-gen sequencing analysis
  http://homer.ucsd.edu/homer/ngs/

- Running NGSplot on the ghpcc06 cluster for making ChIP-Seq and RNA-Seq heatmaps and extracting tag density files
- Running HOMER on the ghpcc06 cluster for annotating bed files and performing motif analysis
- Running bedtools on the ghpcc06 cluster for comparing .bed files of different experiments

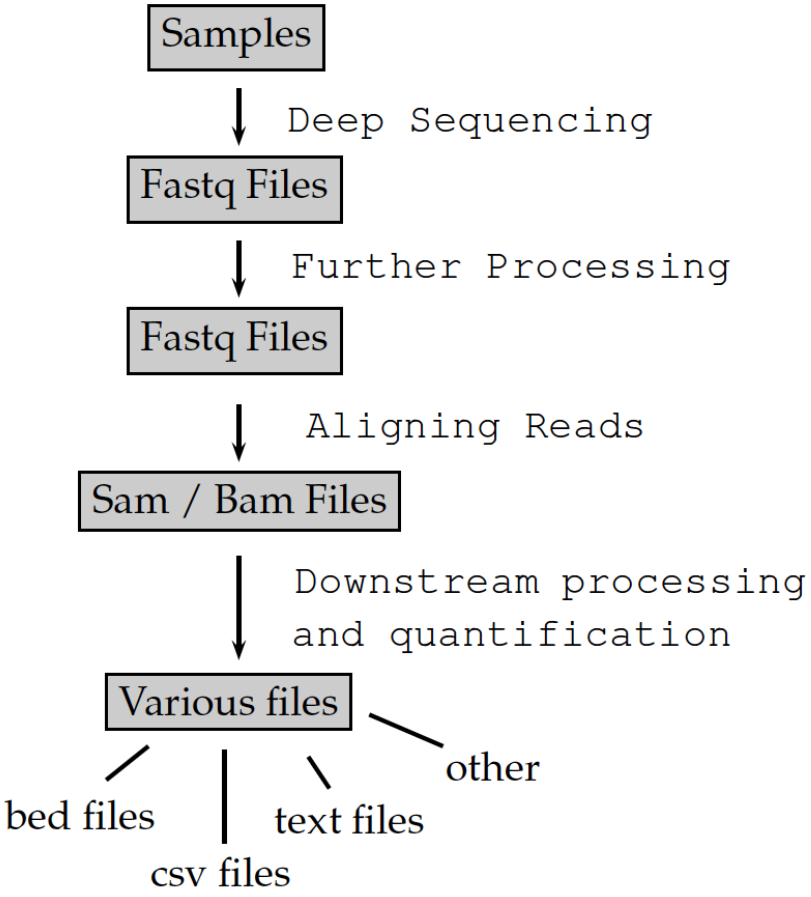
A Typical Deep-Sequencing Workflow



Deep Sequencing Data pipelines involve a lot of text processing.

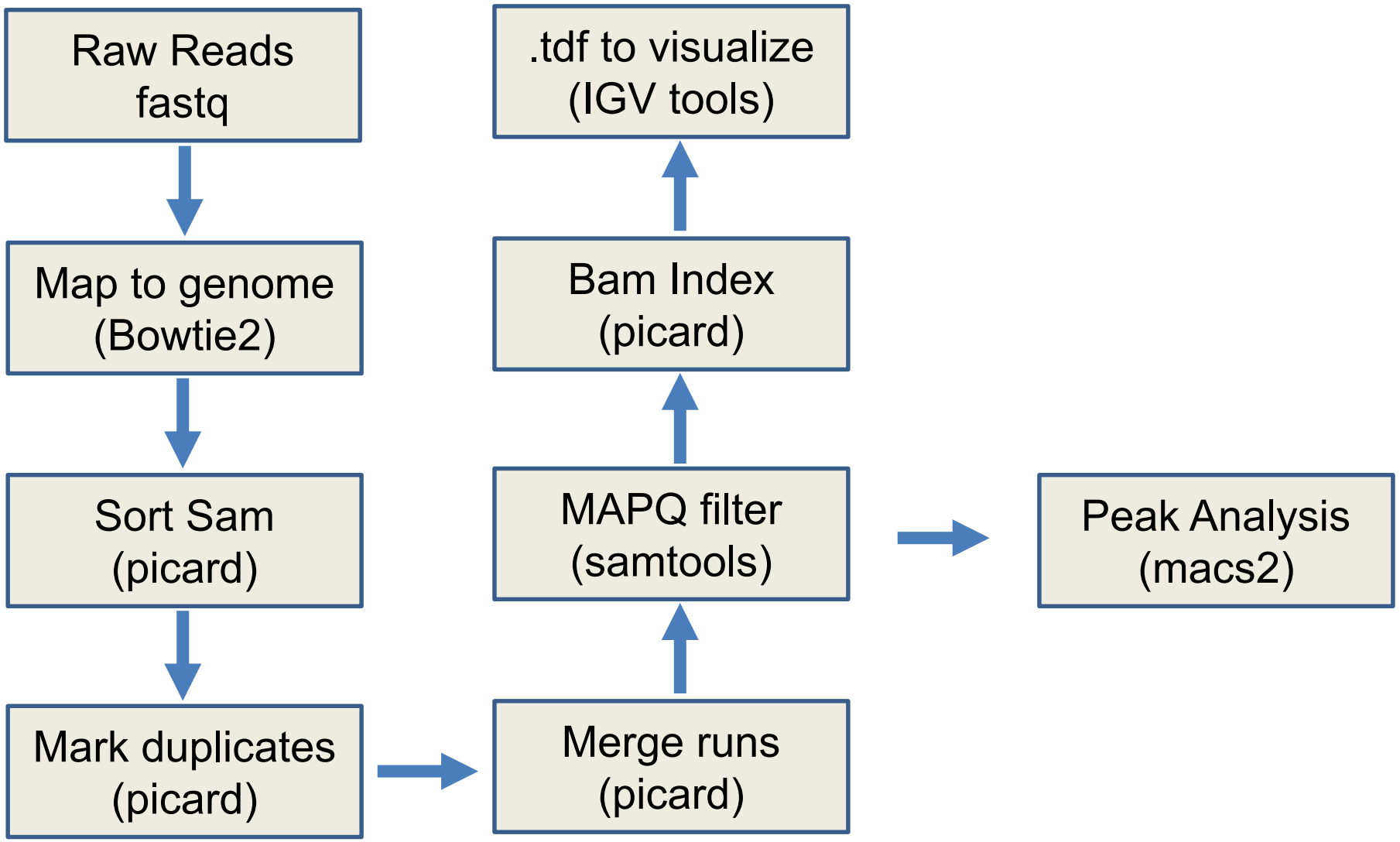
This is an oversimplified model and your workflow can look different from this!

SAM / BAM Files



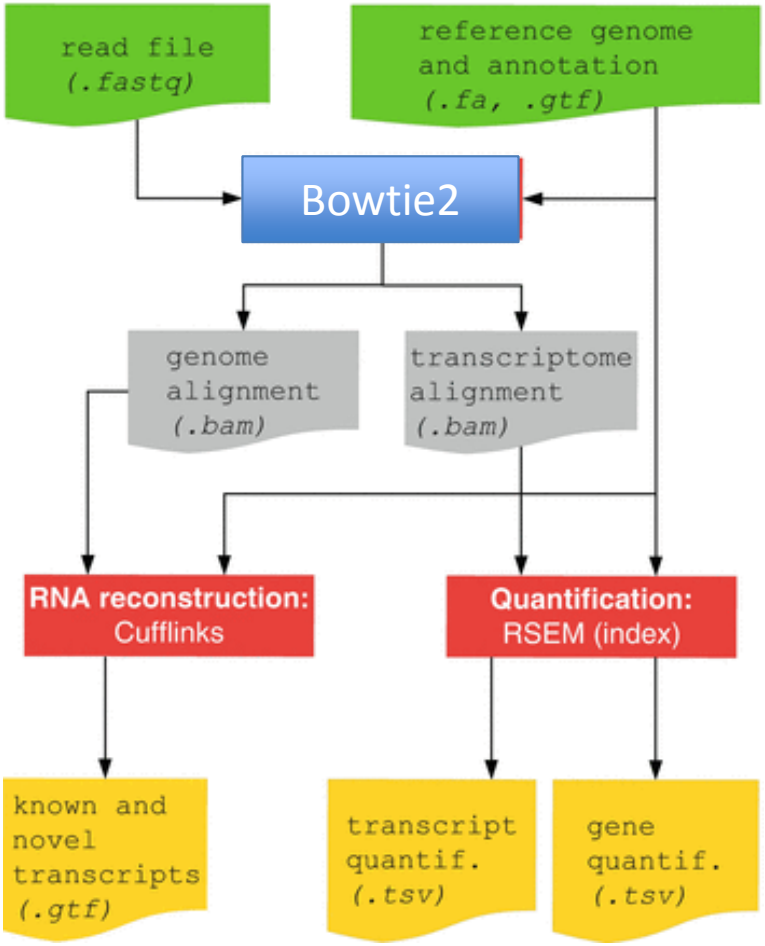
When a fastq file is aligned against a reference genome, a sam or a bam file is created as the ultimate output of the alignment. These files tell us where and how reads in the fastq file are mapped.

My ChIP-Sequencing Workflow



RNA-Seq Pipline

Methods and Tools

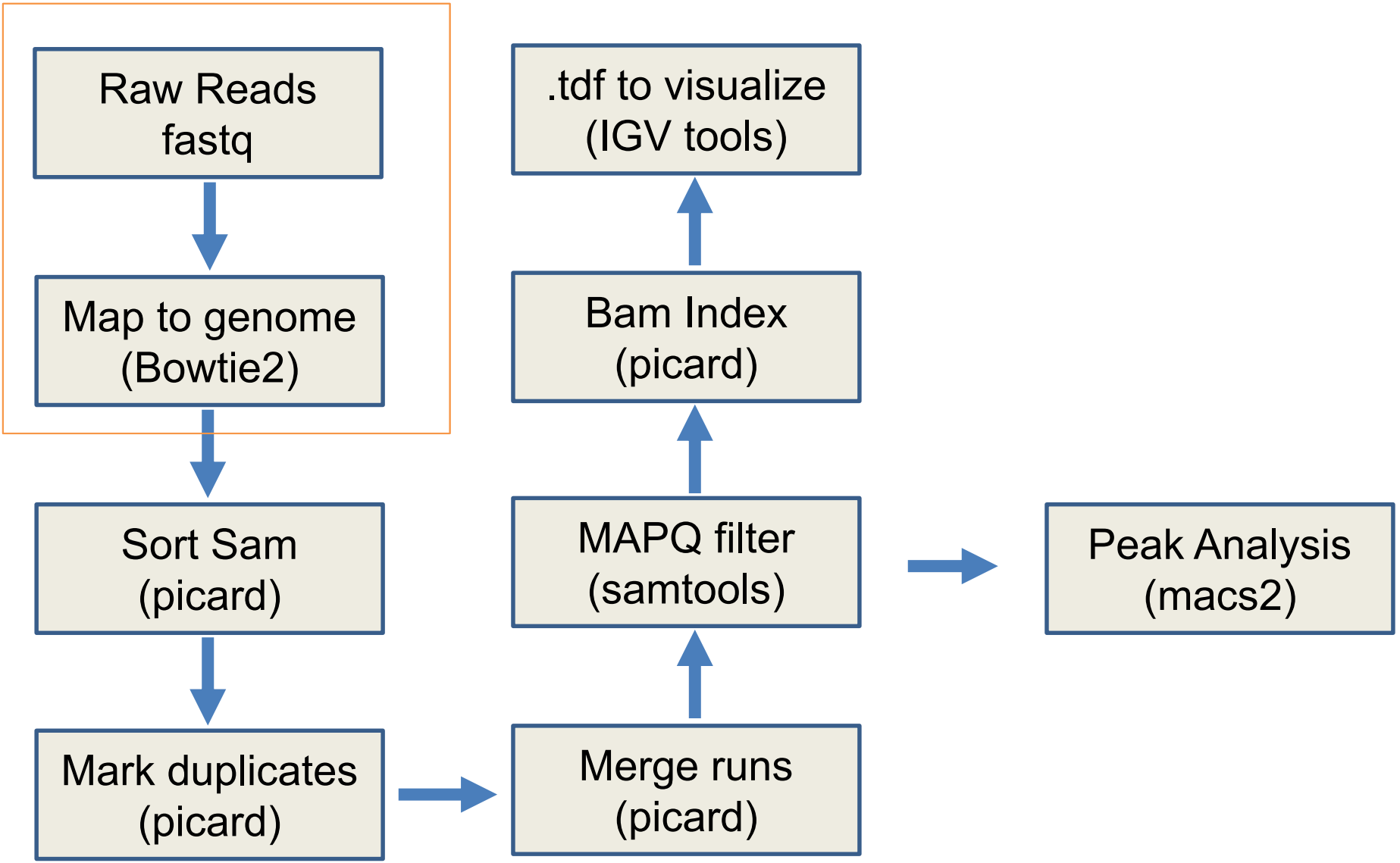


Bowtie and Bowtie2 use Burrows-Wheeler indexing for aligning reads. With bowtie2 there is no upper limit on the read length

Bowtie2 to align reads in a splice-aware manner and aids the discovery of new splice junctions

RSEM is a software package for estimating gene and isoform expression levels from RNA-Seq data. The RSEM package provides an user-friendly interface, supports threads for parallel computation of the EM algorithm, single-end and paired-end read data, quality scores, variable-length reads and RSPD estimation.

My ChIP-Sequencing Workflow



```
[ss45w@ghpcc06 TereChIP]$ bowtie2 -h
Bowtie 2 version 2.3.4.3 by Ben Langmead (langmea@cs.jhu.edu, www.cs.jhu.edu/~langmea)
Usage:
  bowtie2 [options]* -x <bt2-idx> {-1 <m1> -2 <m2> | -U <r> | --interleaved <i>} [-S <sam>]

  <bt2-idx>  Index filename prefix (minus trailing .X.bt2).
             NOTE: Bowtie 1 and Bowtie 2 indexes are not compatible.
  <m1>       Files with #1 mates, paired with files in <m2>.
             Could be gzip'ed (extension: .gz) or bzip2'ed (extension: .bz2).
  <m2>       Files with #2 mates, paired with files in <m1>.
             Could be gzip'ed (extension: .gz) or bzip2'ed (extension: .bz2).
  <r>        Files with unpaired reads.
             Could be gzip'ed (extension: .gz) or bzip2'ed (extension: .bz2).
  <i>        Files with interleaved paired-end FASTQ reads
             Could be gzip'ed (extension: .gz) or bzip2'ed (extension: .bz2).
  <sam>      File for SAM output (default: stdout)

  <m1>, <m2>, <r> can be comma-separated lists (no whitespace) and can be
  specified many times.  E.g. '-U file1.fq,file2.fq -U file3.fq'.

Options (defaults in parentheses):

 Input:
  -q                 query input files are FASTQ .fq/.fastq (default)
  --tab5             query input files are TAB5 .tab5
  --tab6             query input files are TAB6 .tab6
  --qseq             query input files are in Illumina's qseq format
  -f                 query input files are (multi-)FASTA .fa/.mfa
  -r                 query input files are raw one-sequence-per-line
  -F k:<int>,i:<int> query input files are continuous FASTA where reads
                     are substrings (k-mers) extracted from a FASTA file <s>
                     and aligned at offsets 1, 1+i, 1+2i ... end of reference
  -c                 <m1>, <m2>, <r> are sequences themselves, not files
  -s/--skip <int>    skip the first <int> reads/pairs in the input (none)
```

#Alignment
.fastq --> .sam using bowtie2 default settings

#example

module load bowtie2/2.3.4.3

bsub -q long -o log -n 1 -R rusage[mem=20000] -R span[hosts=1] -W 480 "bowtie2 -x /share/data/umw_biocore/genome_data/mouse/mm10/mm10 –q /nl/umw_anthony_imbalzano/Sabriya/TereChIP/ProInoCu_IN_rep1.fq.gz -S ProInoCu_IN_rep1.sam"

```
[ss45w@ghpcc06 ~]$ ls /share/data/umw_biocore/genome_data/mouse/mm10
chrLength.txt                   mm10.5.ht2                  rsem_ref.grp
chrNameLength.txt               mm10.6.ht2                  rsem_ref.idx.fa
chrName.txt                     mm10.7.ht2                  rsem_ref.n2g.idx.fa
chrStart.txt                    mm10.8.ht2                  rsem_ref.rev.1.bt2
commondb                        mm10.bed                    rsem_ref.rev.1.ebwt
exonGeTrInfo.tab                mm10.chrom.sizes            rsem_ref.rev.2.bt2
exonInfo.tab                    mm10.dict                   rsem_ref.rev.2.ebwt
geneInfo.tab                    mm10.fa                     rsem_ref.seq
genes.hisat2_exons.txt          mm10.fa.fai                 rsem_ref_star
genes.hisat2_splice_sites.txt   mm10.gtf                    rsem_ref.ti
Genome                          mm10.phyloP.placental.mod   rsem_ref.transcripts.1.bt2
genomeParameters.txt            mm10.rev.1.bt2              rsem_ref.transcripts.2.bt2
knownIsoforms.txt               mm10.rev.1.ebwt             rsem_ref.transcripts.3.bt2
Log.out                         mm10.rev.2.bt2              rsem_ref.transcripts.4.bt2
makeBed.pl                      mm10.rev.2.ebwt             rsem_ref.transcripts.fa
mm10.1.bt2                      refACT.tab                  rsem_ref.transcripts.fa.fai
mm10.1.ebwt                     ref_flat                    rsem_ref.transcripts.rev.1.bt2
mm10.1.ht2                      refFlat                     rsem_ref.transcripts.rev.2.bt2
mm10.2bit                       ref_flat.mm10               SA
mm10.2.bt2                      rsem_ref.1.bt2              SAindex
mm10.2.ebwt                     rsem_ref.1.ebwt             sizes
mm10.2.ht2                      rsem_ref.2.bt2              sjdbInfo.txt
mm10.3.bt2                      rsem_ref.2.ebwt             sjdbList.fromGTF.out.tab
mm10.3.ebwt                     rsem_ref.3.bt2              sjdbList.out.tab
mm10.3.ht2                      rsem_ref.3.ebwt             transcriptInfo.tab
mm10.4.bt2                      rsem_ref.4.bt2              ucsc.gtf
mm10.4.ebwt                     rsem_ref.4.ebwt             ucsc_into_genesymbol
mm10.4.ht2                      rsem_ref.chrlist            ucsc_into_genesymbol.rsem
```

Bowtie2 sample output

```
[ss45w@ghpcc06 ChIP-Seq_1]$ bsub -q interactive -n 4 -R rusage[mem=4096] -R
span[hosts=1] -W 360 -Is "bowtie2 -x
/share/data/umw_biocore/genome_data/mouse/mm10/mm10 -q -1
LIB034321_CHS00123006_S1_L001_R1.fastq.bz2 -2
LIB034321_CHS00123006_S1_L001_R2.fastq.bz2 -S
LIB034321_CHS00123006_S1_L001.sam"
Job <4575939> is submitted to queue <interactive>.
```

Log file output:

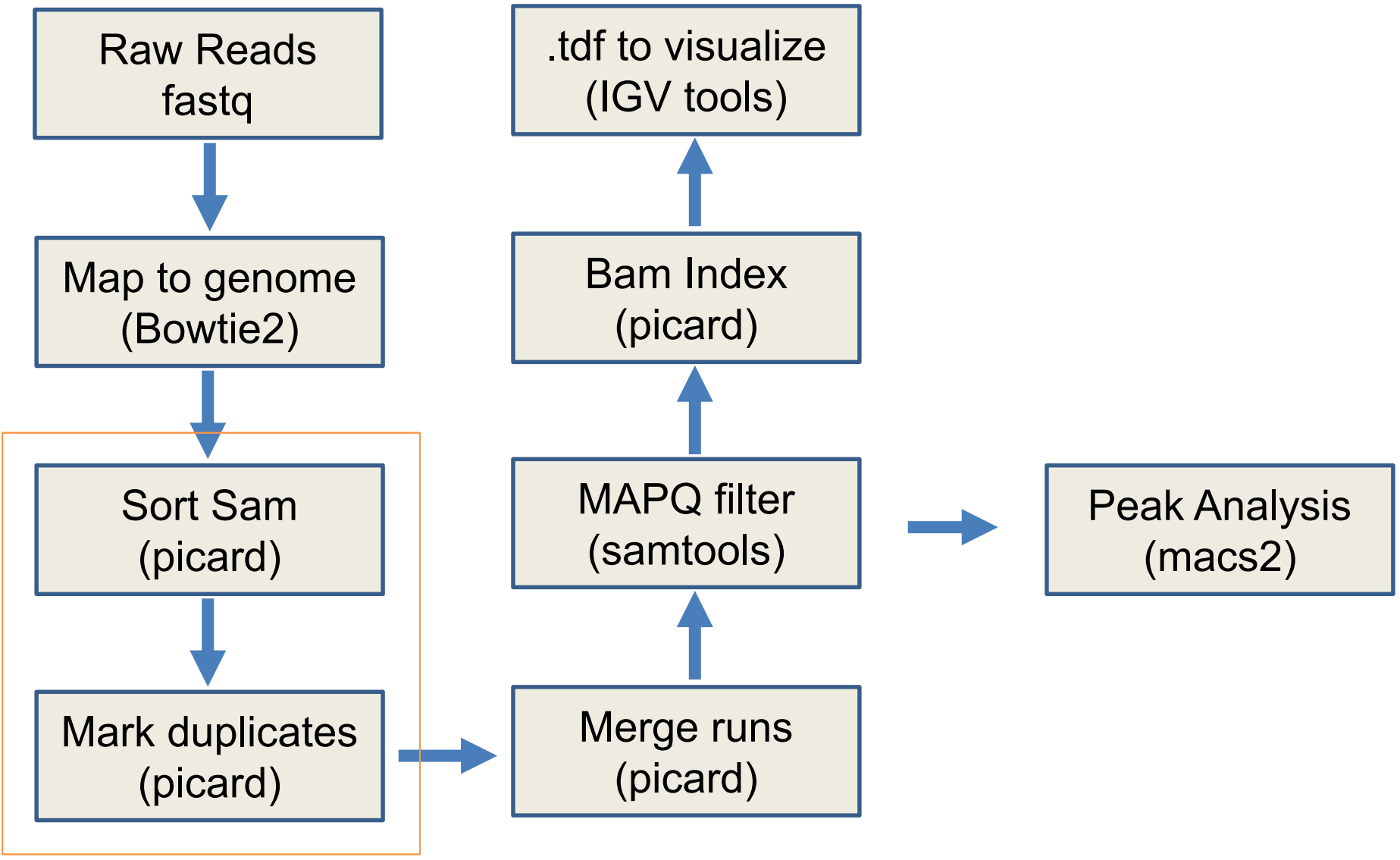
```
Warning: gzbuffer added in zlib v1.2.3.5. Unable to change buffer size from default of 8192.
Warning: gzbuffer added in zlib v1.2.3.5. Unable to change buffer size from default of 8192.
2803003 reads; of these:
  2803003 (100.00%) were paired; of these:
    482561 (17.22%) aligned concordantly 0 times
    1761290 (62.84%) aligned concordantly exactly 1 time
    559152 (19.95%) aligned concordantly >1 times
    ----
    482561 pairs aligned concordantly 0 times; of these:
      156820 (32.50%) aligned discordantly 1 time
    ----
    325741 pairs aligned 0 times concordantly or discordantly; of these:
      651482 mates make up the pairs; of these:
        490472 (75.29%) aligned 0 times
        71150 (10.92%) aligned exactly 1 time
        89860 (13.79%) aligned >1 times
91.25% overall alignment rate
```

Example of BSUB lines

```
#!/bin/bash
###################### BSUB LINES TO CONFIGURE SCHEDULING OF JOB
#BSUB -q amber128
#####BSUB -q mwgpu
#BSUB -n 1
#BSUB -R "rusage[gpu=1:mem=12288],span[hosts=1]"
#BSUB -J p531_pep_EMIN
#BSUB -o out
#BSUB -e err
```

```
File Edit Options Buffers Tools Sh-Script Help
#!/bin/bash
##################### BSUB LINES TO CONFIGURE SCHEDULING OF JOB
#BSUB -q amber128
#####BSUB -q mwgpu
#BSUB -n 1
#BSUB -R "rusage[gpu=1:mem=12288],span[hosts=1]"
#BSUB -J p531_pep_EMIN
#BSUB -o out
#BSUB -e err

## leave sufficient time between job submissions (30-60 secs)
## the number of GPUs allocated matches -n value automatically
## always reserve GPU (gpu=1), setting this to 0 is a cpu job only
## reserve 6144 MB (5 GB + 20%) memory per GPU
## run all processes (1<=n<=4)) on same node (hosts=1).


# cuda 8 & mpich *NEW* for AMBER16
export PATH=/usr/local/cuda/bin:$PATH
export LD_LIBRARY_PATH=/usr/local/cuda/lib64:$LD_LIBRARY_PATH
export PATH=/usr/local/mpich-3.1.4/bin:/home/apps/amber/12cpu-only/bin:$PATH
export LD_LIBRARY_PATH=/usr/local/mpich-3.1.4/lib:$LD_LIBRARY_PATH

# unique job scratch dirs *NEW* for AMBER16
MYSANSCRATCH=/sanscratch/$LSB_JOBID
MYLOCALSCRATCH=/localscratch/$LSB_JOBID
export MYSANSCRATCH MYLOCALSCRATCH
#cd $MYLOCALSCRATCH

## AMBER we need to recreate env, /home/apps/amber/12cpu-only is already set
export PATH=/share/apps/CENTOS6/python/2.7.9/bin:$PATH
export LD_LIBRARY_PATH=/share/apps/CENTOS6/python/2.7.9/lib:$LD_LIBRARY_PATH
source /usr/local/amber16/amber.sh

# commands to run calculations go here.
# Note: emin does not run in parallel on GPUs. Do NOT use pmemd.cuda.MPI for emins, although that is fin

# energy minimization
```

My ChIP-Sequencing Workflow



SAM files are human-readable text **files**, and **BAM files** are simply their binary equivalent, whilst CRAM **files** are a restructured column-oriented binary container format. **BAM files** are typically compressed and more efficient for software to work with than **SAM**.

en.wikipedia.org › wiki › SAMtools ▾

SAMtools - Wikipedia

```
Sabriyas-MBP:~ sabriya$ less /Volumes/syeds/Imbalzano_Lab/ChIP-Seq_PearsonCorrelation/D0_Prmt5_n1n2merged.sam

NB501715:240:HFJGMBGX7:1:22206:24600:12679      163     chr10   3100009 42      36M1D39M        =       3100358 424     ACCTGA
GGAATACCGAATGGCAGAGAAACACCTGAAAAAATGTTCAACATCCTTAATCATCAGGGAAATGCAAAT     AAAAAEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEE
EEEEEEEEEEEEEEEEEEEEE     MD:Z:7A28^T39   PG:Z:MarkDuplicates.1.2 XG:i:1  NM:i:2  XM:i:1  XN:i:0  XO:i:1  AS:i:-13        XS
:i:0  YS:i:0  YT:Z:CP
NB501715:240:HFJGMBGX7:3:11605:14826:5394       99      chr10   3100077 40      75M     =       3100412 410     ATGCAAATCAAAAC
AACACTGAGATTCCACTTCACTCCAGTCAGAATGGCTAAGATCAAAAACTCAGGTGACAGC     AAAAAEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEE
EEEEEEEEEEEEEE     MD:Z:41T31A1    PG:Z:MarkDuplicates.6.5 XG:i:0  NM:i:2  XM:i:2  XN:i:0  XO:i:0  AS:i:-10        XS:i:0  YS
:i:-15        YT:Z:CP
NB502075:261:HM32LBGX9:3:11604:1377:5930        163     chr10   3100206 32      75M     =       3100218 87      TGCAAGCTTGTACA
ACCACTCTGGAAATCAGTCTGGCGGTTCCTCAGAAAATTGGACATAGTACTACCGGAGTAC     AAAAAEEEEEEEEEEEEEEEEEEEEE6EEEEEEEEE/EEEEEEEEEEEEEEEEEEEEE
EEEAEAEAEAEEE/E     MD:Z:75 PG:Z:MarkDuplicates.6.7 XG:i:0  NM:i:0  XM:i:0  XN:i:0  XO:i:0  AS:i:0  XS:i:-5 YS:i:0  YT:Z:CP
NB502075:261:HM32LBGX9:3:11604:1377:5930        83      chr10   3100218 32      75M     =       3100206 -87     CAACCACTCTGGAA
ATCAGTCTGGCGGTTCCTCAGAAAATTGGACATAGTACTACCGGAGTACCCAGATGCCCCT     EEEAEEEEEEEEEEEEEEE<EEEE<E/EEEEEEE/EEA6EEEEEEAEEEEEEEAEEEEE
EEEEEEEEEEAAAAA     MD:Z:75 PG:Z:MarkDuplicates.6.7 XG:i:0  NM:i:0  XM:i:0  XN:i:0  XO:i:0  AS:i:0  XS:i:-31        YS:i:0  YT
:Z:CP
NB501715:240:HFJGMBGX7:3:11501:24920:19766      163     chr10   3100257 42      75M     =       3100275 93      TTGGACATAGTACT
ACCGGAGTACCCAGATGCCCCTCAACAGAGGAATGGATACAGAAAATGTGGTACATTTACA     AAAAAEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEE
EEEEEEEEEEEEEE     MD:Z:75 PG:Z:MarkDuplicates.6.5 XG:i:0  NM:i:0  XM:i:0  XN:i:0  XO:i:0  AS:i:0  YS:i:0  YT:Z:CP
NB502075:150:H2Y5MBGX7:3:13403:22305:6913        163     chr10   3100257 42      75M     =       3100275 93      TTGGACATAGTACT
ACCGGAGTACCCAGATGCCCCTCAACAGAGGAATGGATACAGAAAATGTGGTACATTTACA     AAAAAEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEE
EEEEEEEEEEEEEE     MD:Z:75 PG:Z:MarkDuplicates.6   XG:i:0  NM:i:0  XM:i:0  XN:i:0  XO:i:0  AS:i:0  YS:i:0  YT:Z:CP
NB501715:240:HFJGMBGX7:3:11501:24920:19766      83      chr10   3100275 42      75M     =       3100257 -93     GAGTACCCAGATGC
CCCTCAACAGAGGAATGGATACAGAAAATGTGGTACATTTACACAATGGAATACTACTCAG     EEEEEEEEEEEEEEEEEEEEEEEEAEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEE
EEEEEEEEEEAAAAA     MD:Z:75 PG:Z:MarkDuplicates.6.5 XG:i:0  NM:i:0  XM:i:0  XN:i:0  XO:i:0  AS:i:0  XS:i:-5 YS:i:0  YT:Z:CP
NB502075:150:H2Y5MBGX7:3:13403:22305:6913        83      chr10   3100275 42      75M     =       3100257 -93     GAGTACCCAGATGC
CCCTCAACAGAGGAATGGATACAGAAAATGTGGTACATTTACACAATGGAATACTACTCAG     EEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEE
EEEEEEEEEEAAAAA     MD:Z:75 PG:Z:MarkDuplicates.6   XG:i:0  NM:i:0  XM:i:0  XN:i:0  XO:i:0  AS:i:0  XS:i:-5 YS:i:0  YT:Z:CP
NB501715:240:HFJGMBGX7:1:22206:24600:12679      83      chr10   3100358 42      75M     =       3100009 -424    AAATGAATTTATGA
AATTCCTAGGCAAATGGATGGACCTGGAGGGTATCATCCTGAGTGAAGTAACCCAATCACA     EEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEAEEEE
EEEEEEEEEEAAAAA     MD:Z:75 PG:Z:MarkDuplicates.1.2 XG:i:0  NM:i:0  XM:i:0  XN:i:0  XO:i:0  AS:i:0  XS:i:0  YS:i:-13        YT
:Z:CP
NB501715:240:HFJGMBGX7:3:11605:14826:5394       147     chr10   3100412 40      75M     =       3100077 -410    GTGTGAAGTAACCC
AATCACAAAGGAACTCGCACAATATGTACTCACTGATAAGTGAATAATAGCCTAGAAACTT     EAEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEE
EEEEEEEEEEAAAAA     MD:Z:1A54G9C8   PG:Z:MarkDuplicates.6.5 XG:i:0  NM:i:3  XM:i:3  XN:i:0  XO:i:0  AS:i:-15        XS:i:-25
        YS:i:-10        YT:Z:CP
NB501715:240:HFJGMBGX7:1:11306:9111:6917        73      chr10   3100904 35      75M     =       3100904 0       TGAACTAACCAGTA
```

https://medium.com/@shilparaopradeep/samtools-guide-learning-how-to-filter-and-manipulate-with-sam-bam-files-2c28b25d29e8

SAM file format

SAM stands for Sequence Alignment/Map format. It is a TAB-delimited text format consisting of a header section, which is optional, and an alignment section. If present, the header must be prior to the alignments. Header lines start with '@', while alignment lines do not.

A full description of the SAM format can be found here. SAM aims to be a format that:

1. Is flexible enough to store all the alignment information generated by various alignment programs;
2. Is simple enough to be easily generated by alignment programs or converted from existing alignment formats;
3. Is compact in file size;
4. Allows most of operations on the alignment to work on a stream without loading the whole alignment into memory;
5. Allows the file to be indexed by genomic position to efficiently retrieve all reads aligning to a locus.

The BAM file must be sorted by the reference ID and then the leftmost coordinate before indexing.

#Picard
sam --> sorted bam

module load picard/2.17.8

#example

bsub -o log -n 1 -q short -R rusage[mem=20000] -W240 "java –jar /share/pkg/picard/2.17.8/picard.jar SortSam I=/nl/umw_anthony_imbalzano/Sabriya/TereChIP/ProInoCu_IN_rep1.sam O=ProInoCu_IN_rep1.sorted.bam SO=coordinate"

SortSam

Sorts a SAM or BAM file. This tool sorts the input SAM or BAM file by coordinate, queryname (QNAME), or some other property of the SAM record. The SortOrder of a SAM/BAM file is found in the SAM file header tag @HD in the field labeled SO.

For a coordinate sorted SAM/BAM file, read alignments are sorted first by the reference sequence name (RNAME) field using the reference sequence dictionary (@SQ tag). Alignments within these subgroups are secondarily sorted using the left-most mapping position of the read (POS). Subsequent to this sorting scheme, alignments are listed arbitrarily.

For queryname-sorted alignments, all alignments are grouped using the queryname field but the alignments are not necessarily sorted within these groups. Reads having the same queryname are derived from the same template.

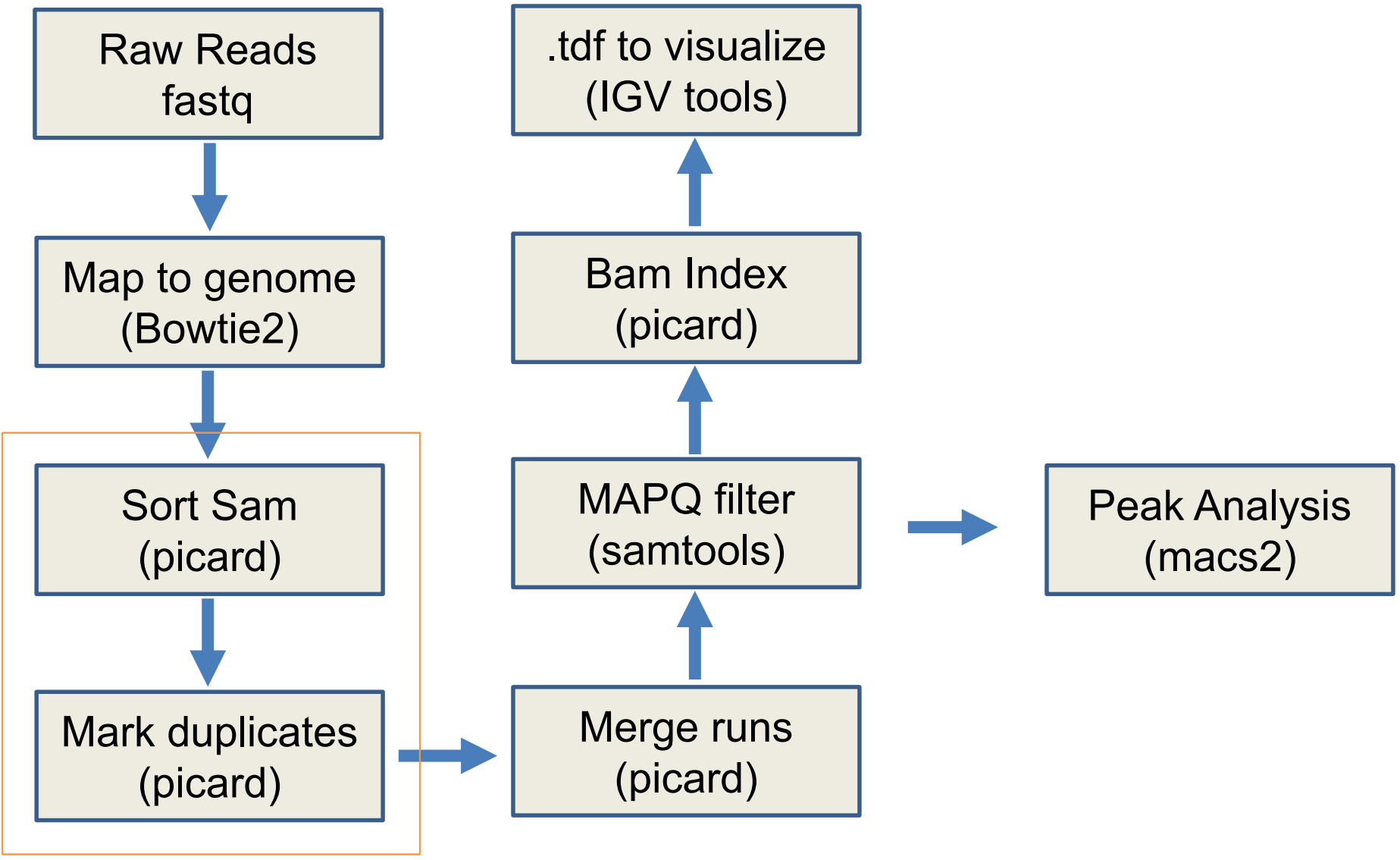
Usage example:

```
java -jar picard.jar SortSam \
      I=input.bam \
      O=sorted.bam \
      SORT_ORDER=coordinate
```

Option	Description
INPUT (File)	The BAM or SAM file to sort. Required.
OUTPUT (File)	The sorted BAM or SAM output file. Required.
SORT_ORDER (SortOrder)	Sort order of output file Required. Possible values: {unsorted, queryname, coordinate, duplicate, unknown}

```
The output (if any) is above this job summary.

12:03:13.413 INFO  NativeLibraryLoader - Loading libgkl_compression.so from jar:file:/share/pkg/picard/2.17.8/picard.jar
!/com/intel/gkl/native/libgkl_compression.so
[Mon Jun 22 12:03:13 EDT 2020] SortSam INPUT=/nl/umw_anthony_imbalzano/Sabriya/TereChIP/ProlnoCu_IN_rep1.sam OUTPUT=Prol
noCu_IN_rep1.sorted.bam SORT_ORDER=coordinate    VERBOSITY=INFO QUIET=false VALIDATION_STRINGENCY=STRICT COMPRESSION_LEV
EL=5 MAX_RECORDS_IN_RAM=500000 CREATE_INDEX=false CREATE_MD5_FILE=false GA4GH_CLIENT_SECRETS=client_secrets.json USE_JDK
_DEFLATER=false USE_JDK_INFLATER=false
[Mon Jun 22 12:03:13 EDT 2020] Executing as ss45w@c38b11 on Linux 2.6.32-754.14.2.el6.x86_64 amd64; Java HotSpot(TM) 64-
Bit Server VM 1.8.0_77-b03; Deflater: Intel; Inflater: Intel; Picard version: 2.17.8-SNAPSHOT
INFO    2020-06-22 12:04:12     SortSam Read    10,000,000 records.  Elapsed time: 00:00:58s.  Time for last 10,000,000:
   58s.  Last read position: */*
INFO    2020-06-22 12:05:07     SortSam Read    20,000,000 records.  Elapsed time: 00:01:53s.  Time for last 10,000,000:
   55s.  Last read position: */*
INFO    2020-06-22 12:05:33     SortSam Finished reading inputs, merging and writing to output now.
INFO    2020-06-22 12:06:04     SortSam Wrote    10,000,000 records from a sorting collection.  Elapsed time: 00:02:49s.
  Time for last 10,000,000:   29s.  Last read position: chr5:90,931,440
INFO    2020-06-22 12:06:26     SortSam Wrote    20,000,000 records from a sorting collection.  Elapsed time: 00:03:12s.
  Time for last 10,000,000:   22s.  Last read position: */*
[Mon Jun 22 12:06:35 EDT 2020] picard.sam.SortSam done. Elapsed time: 3.36 minutes.
Runtime.totalMemory()=11317805056

------------------------------------------------------------
Sender: LSF System <lsfadmin@c38b11>
Subject: Job 7698962: <java -jar /share/pkg/picard/2.17.8/picard.jar SortSam I=/nl/umw_anthony_imbalzano/Sabriya/TereChI
P/ProlnoCu_IN_rep1.sam O=ProlnoCu_IN_rep1.sorted.bam SO=coordinate> in cluster <umghpcc> Done

Job <java -jar /share/pkg/picard/2.17.8/picard.jar SortSam I=/nl/umw_anthony_imbalzano/Sabriya/TereChIP/ProlnoCu_IN_rep1
.sam O=ProlnoCu_IN_rep1.sorted.bam SO=coordinate> was submitted from host <ghpcc06> by user <ss45w> in cluster <umghpcc>
 at Mon Jun 22 12:03:07 2020
Job was executed on host(s) <c38b11>, in queue <short>, as user <ss45w> in cluster <umghpcc> at Mon Jun 22 12:03:07 2020
</home/ss45w> was used as the home directory.
</nl/umw_anthony_imbalzano/Sabriya/TereChIP> was used as the working directory.
Started at Mon Jun 22 12:03:07 2020
Terminated at Mon Jun 22 12:06:35 2020
Results reported at Mon Jun 22 12:06:35 2020

Your job looked like:

------------------------------------------------------------
# LSBATCH: User input
java -jar /share/pkg/picard/2.17.8/picard.jar SortSam I=/nl/umw_anthony_imbalzano/Sabriya/TereChIP/ProlnoCu_IN_rep1.sam
O=ProlnoCu_IN_rep1.sorted.bam SO=coordinate
------------------------------------------------------------

Successfully completed.

Resource usage summary:

    CPU time :                                   181.87 sec.
    Max Memory :                                 10586 MB
    Average Memory :                             8529.50 MB
    Total Requested Memory :                     20000.00 MB
    Delta Memory :                               9414.00 MB
    Max Swap :                                   -
    Max Processes :                              3
    Max Threads :                                54
    Run time :                                   208 sec.
    Turnaround time :                            208 sec.

The output (if any) is above this job summary.
```

My ChIP-Sequencing Workflow



#Picard
MARK DUPLICATES

module load picard/2.17.8

#example

bsub -o logX -n 1 -q short -R rusage[mem=20000] -W240 "java -jar /share/pkg/picard/2.17.8/picard.jar MarkDuplicates I=/nl/umw_anthony_imbalzano/Sabriya/TereChIP/ProlnoCu_IN_rep1.sorted.bam O=ProlnoCu_IN_rep1_sorted.dedup.bam M=markdup_metrics.txt REMOVE_DUPLICATES=true"

MarkDuplicates

Identifies duplicate reads.

This tool locates and tags duplicate reads in a BAM or SAM file, where duplicate reads are defined as originating from a single fragment of DNA. Duplicates can arise during sample preparation e.g. library construction using PCR. See also **EstimateLibraryComplexity** for additional notes on PCR duplication artifacts. Duplicate reads can also result from a single amplification cluster, incorrectly detected as multiple clusters by the optical sensor of the sequencing instrument. These duplication artifacts are referred to as optical duplicates.

The MarkDuplicates tool works by comparing sequences in the 5 prime positions of both reads and read-pairs in a SAM/BAM file. An BARCODE_TAG option is available to facilitate duplicate marking using molecular barcodes. After duplicate reads are collected, the tool differentiates the primary and duplicate reads using an algorithm that ranks reads by the sums of their base-quality scores (default method).

The tool's main output is a new SAM or BAM file, in which duplicates have been identified in the SAM flags field for each read. Duplicates are marked with the hexadecimal value of 0x0400, which corresponds to a decimal value of 1024. If you are not familiar with this type of annotation, please see the following **blog post** for additional information.

Although the bitwise flag annotation indicates whether a read was marked as a duplicate, it does not identify the type of duplicate. To do this, a new tag called the duplicate type (DT) tag was recently added as an optional output in the 'optional field' section of a SAM/BAM file. Invoking the TAGGING_POLICY option, you can instruct the program to mark all the duplicates (All), only the optical duplicates (OpticalOnly), or no duplicates (DontTag). The records within the output of a SAM/BAM file will have values for the 'DT' tag (depending on the invoked TAGGING_POLICY), as either library/PCR-generated duplicates (LB), or sequencing-platform artifact duplicates (SQ). This tool uses the READ_NAME_REGEX and the OPTICAL_DUPLICATE_PIXEL_DISTANCE options as the primary methods to identify and differentiate duplicate types. Set READ_NAME_REGEX to null to skip optical duplicate detection, e.g. for RNA-seq or other data where duplicate sets are extremely large and estimating library complexity is not an aim. Note that without optical duplicate counts, library size estimation will be inaccurate.

MarkDuplicates also produces a metrics file indicating the numbers of duplicates for both single- and paired-end reads.

The program can take either coordinate-sorted or query-sorted inputs, however the behavior is slightly different. When the input is coordinate-sorted, unmapped mates of mapped records and supplementary/secondary alignments are not marked as duplicates. However, when the input is query-sorted (actually query-grouped), then unmapped mates and secondary/supplementary reads are not excluded from the duplication test and can be marked as duplicate reads.

Usage example:

```
java -jar picard.jar MarkDuplicates \
      I=input.bam \
      O=marked_duplicates.bam \
      M=marked_dup_metrics.txt
```

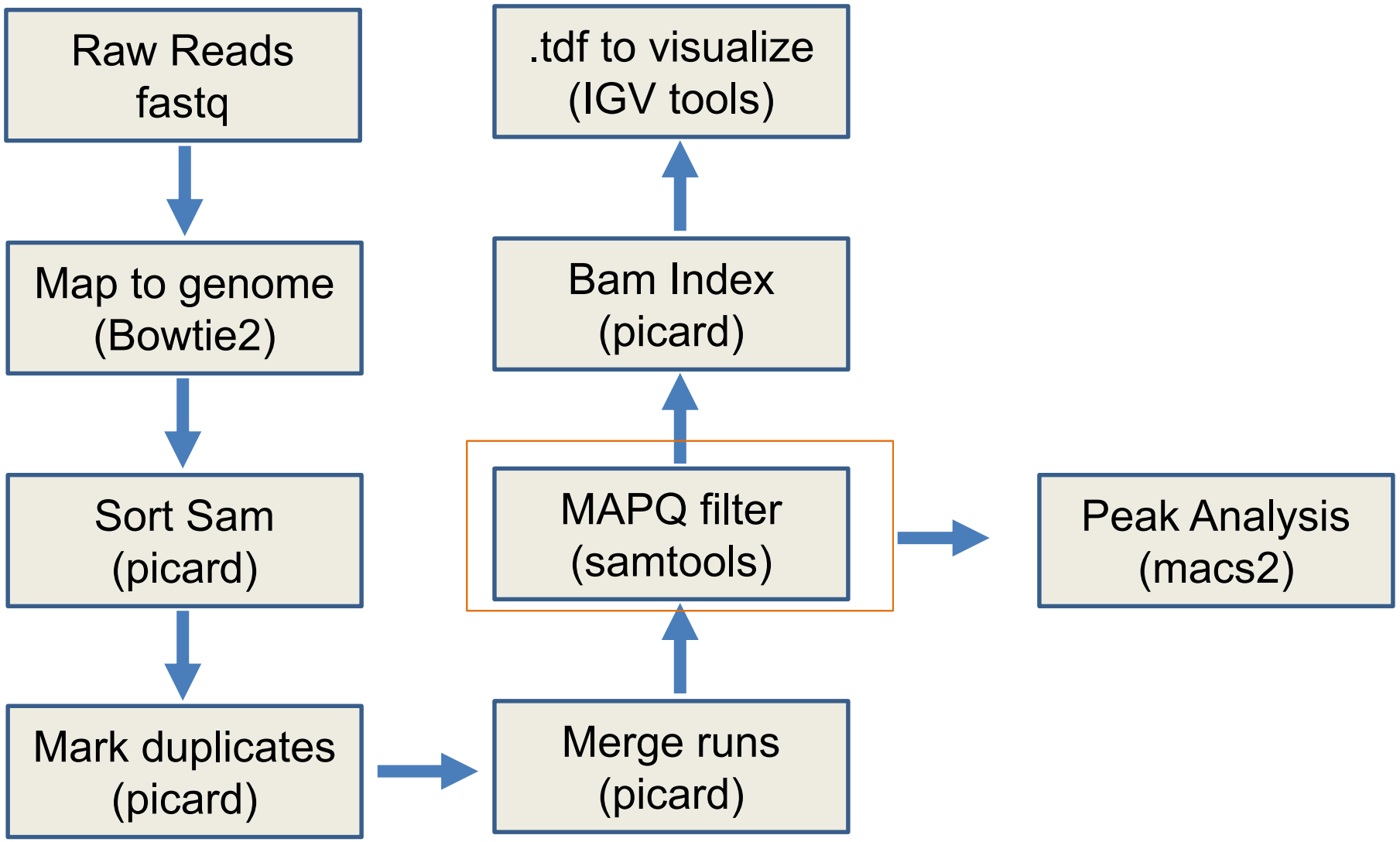
Please see **MarkDuplicates** for detailed explanations of the output metrics.

Option	Description
MAX_SEQUENCES_FOR_DISK_READ_ENDS_MAP (Integer)	This option is obsolete. ReadEnds will always be spilled to disk. Default value: 50000. This option can be set to 'null' to clear the default value.
MAX_FILE_HANDLES_FOR_READ_ENDS_MAP (Integer)	Maximum number of file handles to keep open when spilling read ends to disk. Set this number a little lower than the per-process maximum number of file that may be open. This number can be found by executing the 'ulimit -n' command on a Unix system. Default value:
REMOVE_DUPLICATES (Boolean)	If true do not write duplicates to the output file instead of writing them with appropriate flags set. Default value: false. This option can be set to 'null' to clear the default value. Possible values: {true, false}

```
12:10:48.915 INFO  NativeLibraryLoader - Loading libgkl_compression.so from jar:file:/share/pkg/picard/2.17.8/picard.jar!/com/intel/gkl/native/
libgkl_compression.so
[Mon Jun 22 12:10:48 EDT 2020] MarkDuplicates INPUT=[/nl/umw_anthony_imbalzano/Sabriya/TereChIP/ProlnoCu_IN_rep1.sorted.bam] OUTPUT=ProlnoCu_IN
_rep1_sorted.dedup.bam METRICS_FILE=markdup_metrics.txt REMOVE_DUPLICATES=true    MAX_SEQUENCES_FOR_DISK_READ_ENDS_MAP=50000 MAX_FILE_HANDLES_F
OR_READ_ENDS_MAP=8000 SORTING_COLLECTION_SIZE_RATIO=0.25 TAG_DUPLICATE_SET_MEMBERS=false REMOVE_SEQUENCING_DUPLICATES=false TAGGING_POLICY=Dont
Tag CLEAR_DT=true ADD_PG_TAG_TO_READS=true ASSUME_SORTED=false DUPLICATE_SCORING_STRATEGY=SUM_OF_BASE_QUALITIES PROGRAM_RECORD_ID=MarkDuplicate
s PROGRAM_GROUP_NAME=MarkDuplicates READ_NAME_REGEX=<optimized capture of last three ':' separated fields as numeric values> OPTICAL_DUPLICATE_
PIXEL_DISTANCE=100 MAX_OPTICAL_DUPLICATE_SET_SIZE=300000 VERBOSITY=INFO QUIET=false VALIDATION_STRINGENCY=STRICT COMPRESSION_LEVEL=5 MAX_RECORD
S_IN_RAM=500000 CREATE_INDEX=false CREATE_MD5_FILE=false GA4GH_CLIENT_SECRETS=client_secrets.json USE_JDK_DEFLATER=false USE_JDK_INFLATER=false
[Mon Jun 22 12:10:48 EDT 2020] Executing as ss45w@c38b11 on Linux 2.6.32-754.14.2.el6.x86_64 amd64; Java HotSpot(TM) 64-Bit Server VM 1.8.0_77-
b03; Deflater: Intel; Inflater: Intel; Picard version: 2.17.8-SNAPSHOT
INFO    2020-06-22 12:10:49     MarkDuplicates  Start of doWork freeMemory: 2027693088; totalMemory: 2058354688; maxMemory: 28631367680
INFO    2020-06-22 12:10:49     MarkDuplicates  Reading input file and constructing read end information.
INFO    2020-06-22 12:10:49     MarkDuplicates  Will retain up to 103736839 data points before spilling to disk.
INFO    2020-06-22 12:11:08     MarkDuplicates  Read     1,000,000 records.  Elapsed time: 00:00:18s.  Time for last 1,000,000:   18s.  Last re
ad position: chr11:61,329,449
INFO    2020-06-22 12:11:08     MarkDuplicates  Tracking 0 as yet unmatched pairs. 0 records in RAM.
INFO    2020-06-22 12:11:17     MarkDuplicates  Read     2,000,000 records.  Elapsed time: 00:00:27s.  Time for last 1,000,000:    9s.  Last re
ad position: chr12:105,487,769
INFO    2020-06-22 12:11:17     MarkDuplicates  Tracking 0 as yet unmatched pairs. 0 records in RAM.
INFO    2020-06-22 12:11:24     MarkDuplicates  Read     3,000,000 records.  Elapsed time: 00:00:34s.  Time for last 1,000,000:    7s.  Last re
ad position: chr14:46,927,256
INFO    2020-06-22 12:11:24     MarkDuplicates  Tracking 0 as yet unmatched pairs. 0 records in RAM.
INFO    2020-06-22 12:11:26     MarkDuplicates  Read     4,000,000 records.  Elapsed time: 00:00:36s.  Time for last 1,000,000:    1s.  Last re
ad position: chr16:21,388,614
INFO    2020-06-22 12:11:26     MarkDuplicates  Tracking 0 as yet unmatched pairs. 0 records in RAM.
INFO    2020-06-22 12:11:33     MarkDuplicates  Read     5,000,000 records.  Elapsed time: 00:00:44s.  Time for last 1,000,000:    7s.  Last re
ad position: chr18:12,939,021
INFO    2020-06-22 12:11:34     MarkDuplicates  Tracking 0 as yet unmatched pairs. 0 records in RAM.
INFO    2020-06-22 12:11:35     MarkDuplicates  Read     6,000,000 records.  Elapsed time: 00:00:46s.  Time for last 1,000,000:    2s.  Last re
ad position: chr1:62,655,252
INFO    2020-06-22 12:11:35     MarkDuplicates  Tracking 0 as yet unmatched pairs. 0 records in RAM.
INFO    2020-06-22 12:11:41     MarkDuplicates  Read     7,000,000 records.  Elapsed time: 00:00:51s.  Time for last 1,000,000:    5s.  Last re
ad position: chr2:56,834,929
INFO    2020-06-22 12:11:41     MarkDuplicates  Tracking 0 as yet unmatched pairs. 0 records in RAM.
INFO    2020-06-22 12:11:43     MarkDuplicates  Read     8,000,000 records.  Elapsed time: 00:00:53s.  Time for last 1,000,000:    1s.  Last re
ad position: chr3:7,867,873
INFO    2020-06-22 12:11:43     MarkDuplicates  Tracking 0 as yet unmatched pairs. 0 records in RAM.
INFO    2020-06-22 12:12:11     MarkDuplicates  Read     9,000,000 records.  Elapsed time: 00:01:21s.  Time for last 1,000,000:   28s.  Last re
ad position: chr4:62,989,068
INFO    2020-06-22 12:12:11     MarkDuplicates  Tracking 0 as yet unmatched pairs. 0 records in RAM.
INFO    2020-06-22 12:12:13     MarkDuplicates  Read    10,000,000 records.  Elapsed time: 00:01:23s.  Time for last 1,000,000:    1s.  Last re
ad position: chr5:90,931,440
INFO    2020-06-22 12:12:13     MarkDuplicates  Tracking 0 as yet unmatched pairs. 0 records in RAM.
INFO    2020-06-22 12:12:14     MarkDuplicates  Read    11,000,000 records.  Elapsed time: 00:01:25s.  Time for last 1,000,000:    1s.  Last re
ad position: chr6:115,918,015
INFO    2020-06-22 12:12:14     MarkDuplicates  Tracking 0 as yet unmatched pairs. 0 records in RAM.
INFO    2020-06-22 12:12:16     MarkDuplicates  Read    12,000,000 records.  Elapsed time: 00:01:27s.  Time for last 1,000,000:    1s.  Last re
ad position: chr7:145,076,533
INFO    2020-06-22 12:12:16     MarkDuplicates  Tracking 0 as yet unmatched pairs. 0 records in RAM.
INFO    2020-06-22 12:12:27     MarkDuplicates  Read    13,000,000 records.  Elapsed time: 00:01:38s.  Time for last 1,000,000:   10s.  Last re
ad position: chr9:22,649,876
INFO    2020-06-22 12:12:27     MarkDuplicates  Tracking 0 as yet unmatched pairs. 0 records in RAM.
INFO    2020-06-22 12:12:29     MarkDuplicates  Read    14,000,000 records.  Elapsed time: 00:01:39s.  Time for last 1,000,000:    1s.  Last re
ad position: chrX:55,299,919
INFO    2020-06-22 12:12:29     MarkDuplicates  Tracking 0 as yet unmatched pairs. 0 records in RAM.
INFO    2020-06-22 12:12:30     MarkDuplicates  Read 14386280 records. 0 pairs never matched.
INFO    2020-06-22 12:13:12     MarkDuplicates  After buildSortedReadEndLists freeMemory: 10268409144; totalMemory: 12326010880; maxMemory: 286
31367680
INFO    2020-06-22 12:13:12     MarkDuplicates  Will retain up to 894730240 duplicate indices before spilling to disk.
INFO    2020-06-22 12:13:13     MarkDuplicates  Traversing read pair information and detecting duplicates.
INFO    2020-06-22 12:13:13     MarkDuplicates  Traversing fragment information and detecting duplicates.
INFO    2020-06-22 12:13:15     MarkDuplicates  Sorting list of duplicate records.
INFO    2020-06-22 12:14:51     MarkDuplicates  After generateDuplicateIndexes freeMemory: 12152355568; totalMemory: 19500367872; maxMemory: 28
631367680
INFO    2020-06-22 12:14:51     MarkDuplicates  Marking 2784830 records as duplicates.
INFO    2020-06-22 12:14:51     MarkDuplicates  Found 0 optical duplicate clusters.
INFO    2020-06-22 12:14:51     MarkDuplicates  Reads are assumed to be ordered by: coordinate
INFO    2020-06-22 12:15:44     MarkDuplicates  Written    10,000,000 records.  Elapsed time: 00:00:53s.  Time for last 10,000,000:   53s.  Las
t read position: chr8:35,944,580
INFO    2020-06-22 12:16:19     MarkDuplicates  Written    20,000,000 records.  Elapsed time: 00:01:28s.  Time for last 10,000,000:   34s.  Las
t read position: */*
INFO    2020-06-22 12:16:25     MarkDuplicates  Before output close freeMemory: 21038461720; totalMemory: 21238906880; maxMemory: 28631367680
INFO    2020-06-22 12:16:26     MarkDuplicates  After output close freeMemory: 20682470168; totalMemory: 20882915328; maxMemory: 28631367680
[Mon Jun 22 12:16:26 EDT 2020] picard.sam.markduplicates.MarkDuplicates done. Elapsed time: 5.62 minutes.
Runtime.totalMemory()=20882915328

------------------------------------------------------------
Sender: LSF System <lsfadmin@c38b11>
Subject: Job 7698970: <java -jar /share/pkg/picard/2.17.8/picard.jar MarkDuplicates I=/nl/umw_anthony_imbalzano/Sabriya/TereChIP/ProlnoCu_IN_re
p1.sorted.bam O=ProlnoCu_IN_rep1_sorted.dedup.bam M=markdup_metrics.txt REMOVE_DUPLICATES=true> in cluster <umghpcc> Done

Job <java -jar /share/pkg/picard/2.17.8/picard.jar MarkDuplicates I=/nl/umw_anthony_imbalzano/Sabriya/TereChIP/ProlnoCu_IN_rep1.sorted.bam O=Pr
olnoCu_IN_rep1_sorted.dedup.bam M=markdup_metrics.txt REMOVE_DUPLICATES=true> was submitted from host <ghpcc06> by user <ss45w> in cluster <umg
hpcc> at Mon Jun 22 12:10:45 2020
Job was executed on host(s) <c38b11>, in queue <short>, as user <ss45w> in cluster <umghpcc> at Mon Jun 22 12:10:47 2020
</home/ss45w> was used as the home directory.
</nl/umw_anthony_imbalzano/Sabriya/TereChIP> was used as the working directory.
Started at Mon Jun 22 12:10:47 2020
Terminated at Mon Jun 22 12:16:26 2020
Results reported at Mon Jun 22 12:16:26 2020

Your job looked like:
```

```
------------------------------------------------------------
# LSBATCH: User input
java -jar /share/pkg/picard/2.17.8/picard.jar MarkDuplicates I=/nl/umw_anthony_imbalzano/Sabriya/TereChIP/ProlnoCu_IN_rep1.sorted.bam O=ProlnoC
u_IN_rep1_sorted.dedup.bam M=markdup_metrics.txt REMOVE_DUPLICATES=true
------------------------------------------------------------

Successfully completed.

Resource usage summary:

    CPU time :                                   333.23 sec.
    Max Memory :                                 18799 MB
    Average Memory :                             11595.09 MB
    Total Requested Memory :                     20000.00 MB
    Delta Memory :                               1201.00 MB
    Max Swap :                                   -
    Max Processes :                              3
    Max Threads :                                54
    Run time :                                   339 sec.
    Turnaround time :                            341 sec.

The output (if any) is above this job summary.

(END)
```

My ChIP-Sequencing Workflow



#Samtools
MAPQ

MAPQ (mapping quality — describes the uniqueness of the alignment, 0=non-unique, >10 probably unique)

module load samtools/1.9

#example

bsub -n 1 -o log -q short -R rusage[mem=20000] -W240 "samtools view -q 20 -b WCE_3T3L1_all.sorted.dedup.bam > WCE_3T3L1_all.sorted.dedup.q20.bam"

Remove low quality mapped reads

In SAM file the quality of mapped reads is defined in by so-called MAPQ values — MAPping Quality. It equals −10 log10 Probability {mapping position is wrong}, rounded to the nearest integer. A value 255 indicates that the mapping quality is not available. Now, it is very important to remember that MAPQ values generated by different aligners (e.g. Bowtie2, TopHat, BBMap, BWA) are not exactly comparable. For example, the maximum value of MAPQ score in Bowtie2 is 42, whereas in BWA — 37. You can read a great piece about why this happens in the ACGT blog post.

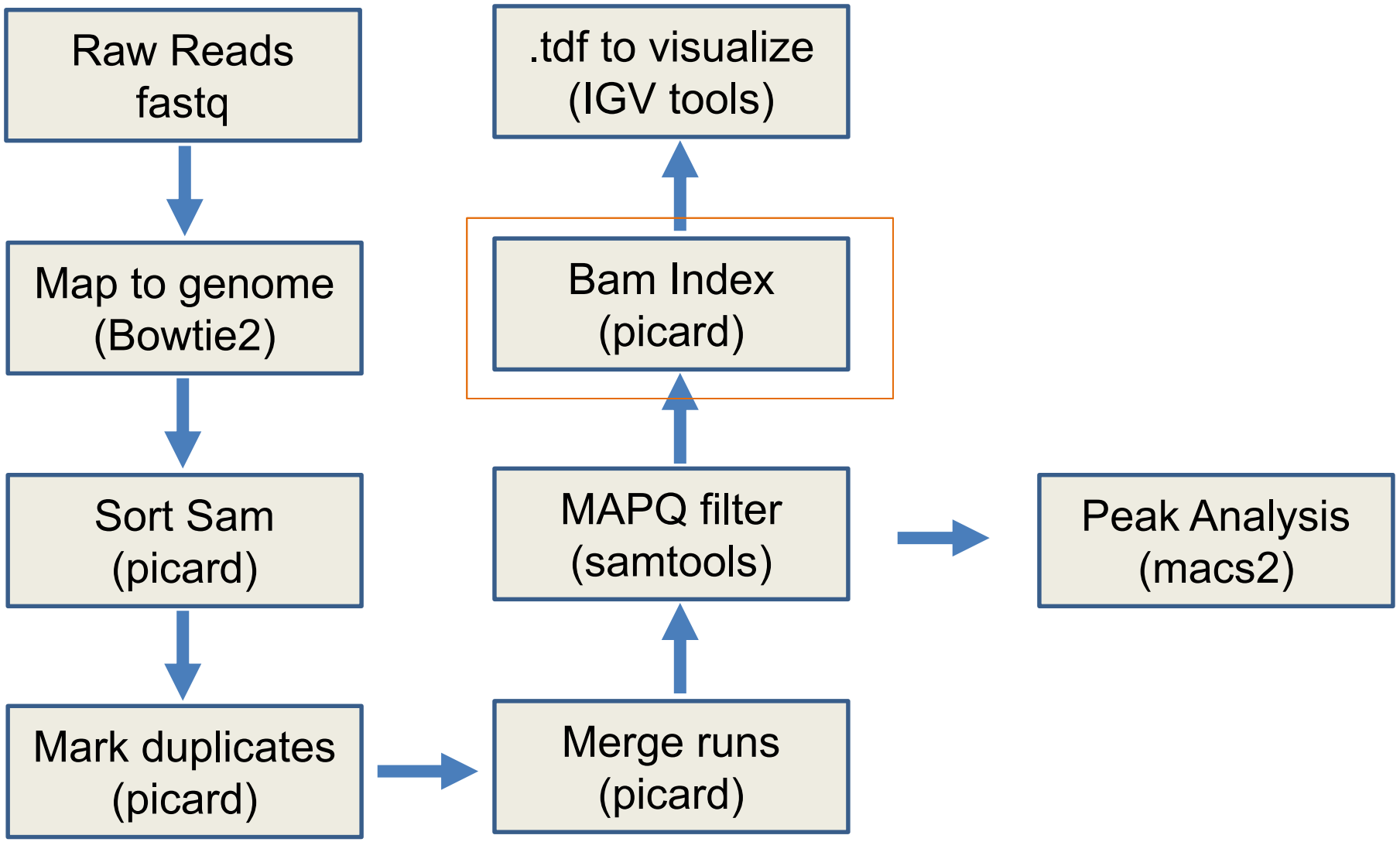
Now, as we understand a little bit better MAPQ scores, we can filter BAM/SAM files on the mapping quality. eg. getting all reads with a mapping quality larger than 30 (you could also use 49 if you wanted to:). This will remove all reads mapped the undesirable mapping qualities and will only keep uniquely mapped reads.

https://medium.com/@shilparaopradeep/samtools-guide-learning-how-to-filter-and-manipulate-with-sam-bam-files-2c28b25d29e8

```
[ss45w@ghpcc06 TereChIP]$ samtools --help

Program: samtools (Tools for alignments in the SAM format)
Version: 1.9 (using htslib 1.9)

Usage:   samtools <command> [options]

Commands:
  -- Indexing
     dict           create a sequence dictionary file
     faidx          index/extract FASTA
     fqidx          index/extract FASTQ
     index          index alignment

  -- Editing
     calmd          recalculate MD/NM tags and '=' bases
     fixmate        fix mate information
     reheader       replace BAM header
     targetcut      cut fosmid regions (for fosmid pool only)
     addreplacerg   adds or replaces RG tags
     markdup        mark duplicates

  -- File operations
     collate        shuffle and group alignments by name
     cat            concatenate BAMs
     merge          merge sorted alignments
     mpileup        multi-way pileup
     sort           sort alignment file
     split          splits a file by read group
     quickcheck     quickly check if SAM/BAM/CRAM file appears intact
     fastq          converts a BAM to a FASTQ
     fasta          converts a BAM to a FASTA

  -- Statistics
     bedcov         read depth per BED region
     depth          compute the depth
     flagstat       simple stats
     idxstats       BAM index stats
     phase          phase heterozygotes
     stats          generate stats (former bamcheck)

  -- Viewing
     flags          explain BAM flags
     tview          text alignment viewer
     view           SAM<->BAM<->CRAM conversion
     depad          convert padded BAM to unpadded BAM
```

```
------------------------------------------------------------
Sender: LSF System <lsfadmin@c38b03>
Subject: Job 7699216: <samtools view -q 20 -b ProlnoCu_IN_rep1_sorted.dedup.bam > ProlnoCu_IN_rep1_sorted.dedup.q20.bam> in cluster <
umghpcc> Done

Job <samtools view -q 20 -b ProlnoCu_IN_rep1_sorted.dedup.bam > ProlnoCu_IN_rep1_sorted.dedup.q20.bam> was submitted from host <ghpcc
06> by user <ss45w> in cluster <umghpcc> at Mon Jun 22 14:36:21 2020
Job was executed on host(s) <c38b03>, in queue <short>, as user <ss45w> in cluster <umghpcc> at Mon Jun 22 14:36:22 2020
</home/ss45w> was used as the home directory.
</nl/umw_anthony_imbalzano/Sabriya/TereChIP> was used as the working directory.
Started at Mon Jun 22 14:36:22 2020
Terminated at Mon Jun 22 14:37:31 2020
Results reported at Mon Jun 22 14:37:31 2020

Your job looked like:

------------------------------------------------------------
# LSBATCH: User input
samtools view -q 20 -b ProlnoCu_IN_rep1_sorted.dedup.bam > ProlnoCu_IN_rep1_sorted.dedup.q20.bam
------------------------------------------------------------

Successfully completed.

Resource usage summary:

    CPU time :                                   67.21 sec.
    Max Memory :                                 4 MB
    Average Memory :                             4.00 MB
    Total Requested Memory :                     20000.00 MB
    Delta Memory :                               19996.00 MB
    Max Swap :                                   -
    Max Processes :                              4
    Max Threads :                                5
    Run time :                                   79 sec.
    Turnaround time :                            70 sec.

The output (if any) is above this job summary.

(END)
```

My ChIP-Sequencing Workflow



https://medium.com/@shilparaopradeep/samtools-guide-learning-how-to-filter-and-manipulate-with-sam-bam-files-2c28b25d29e8

Sorting and indexing the BAM file

Some downstream analysis programs that use BAM files actually require indexed BAM file. For example a common tool for genome alignment visualization the Integrative Genomics Viewer needs such format (IGV).

Indexing aims to achieve a fast retrieval of alignments overlapping a specified region without going through the whole alignments. The BAM file must be sorted by the reference ID and then the leftmost coordinate before indexing.

#Picard
Build merged Bam Index

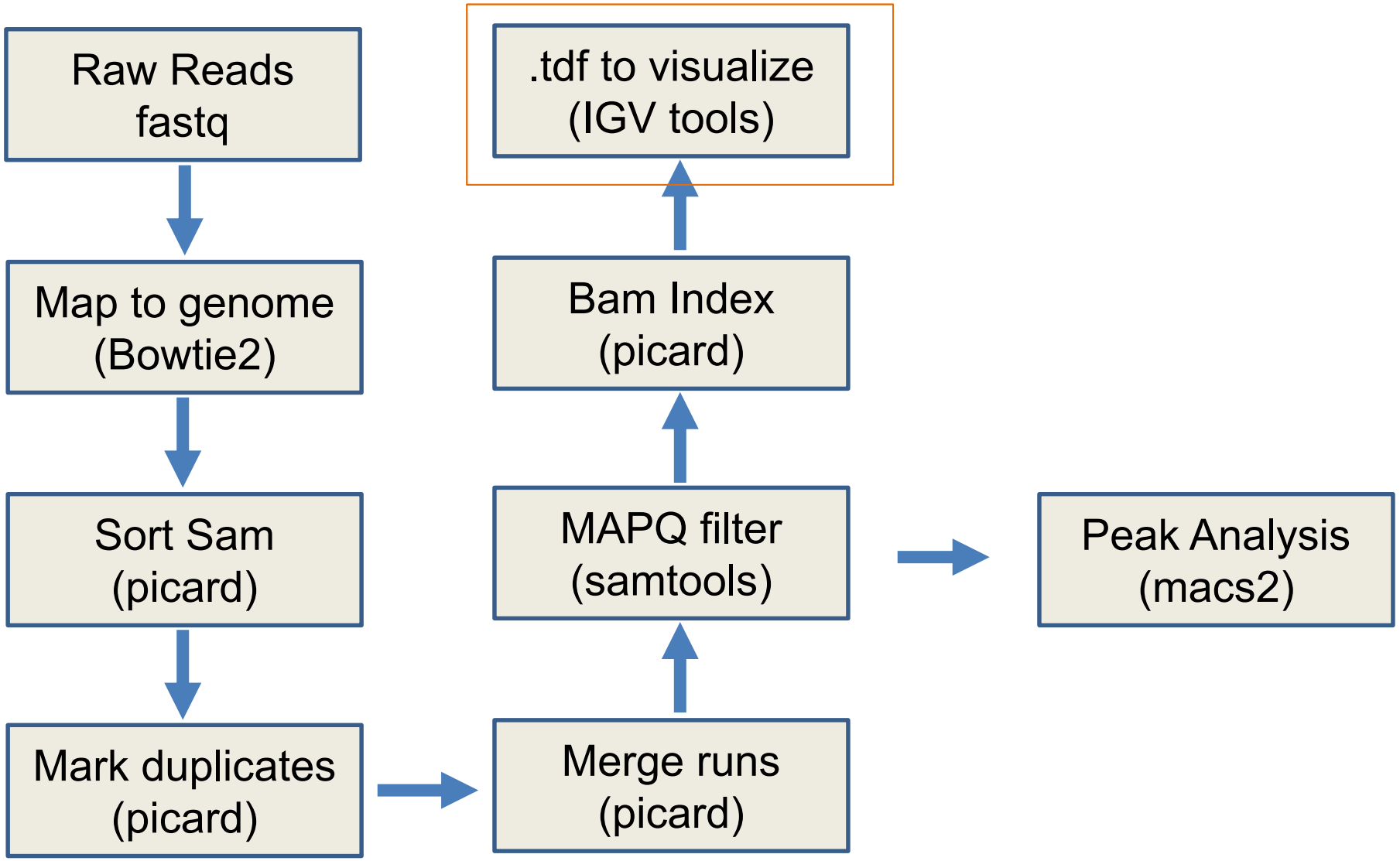
module load picard/2.17.8

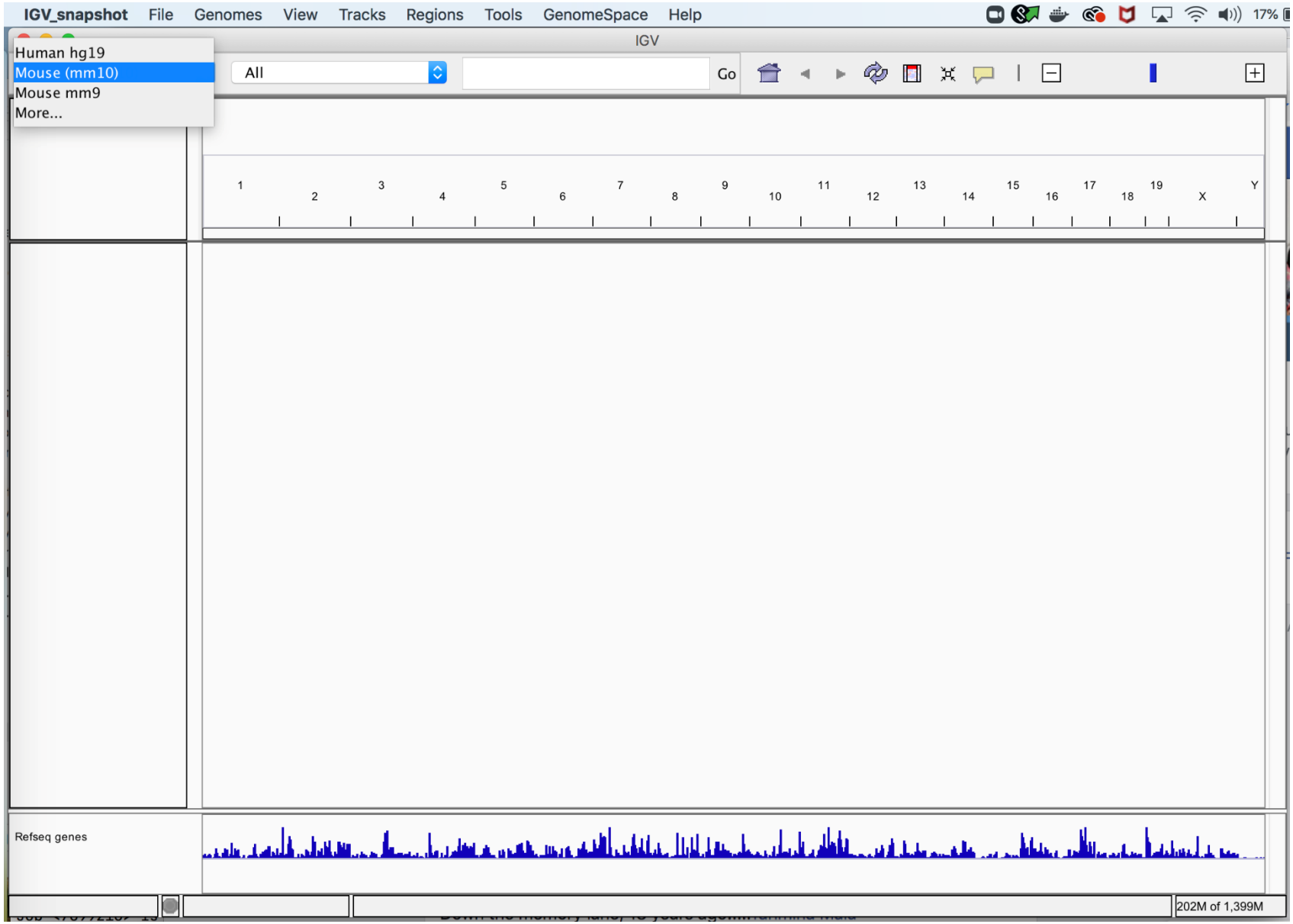
#example

bsub -o log3 -n 1 -q short -R rusage[mem=20000] -W240 "java -jar /share/pkg/picard/2.17.8/picard.jar BuildBamIndex I=ProlnoCu_IN_rep1_sorted.dedup.q20.bam"

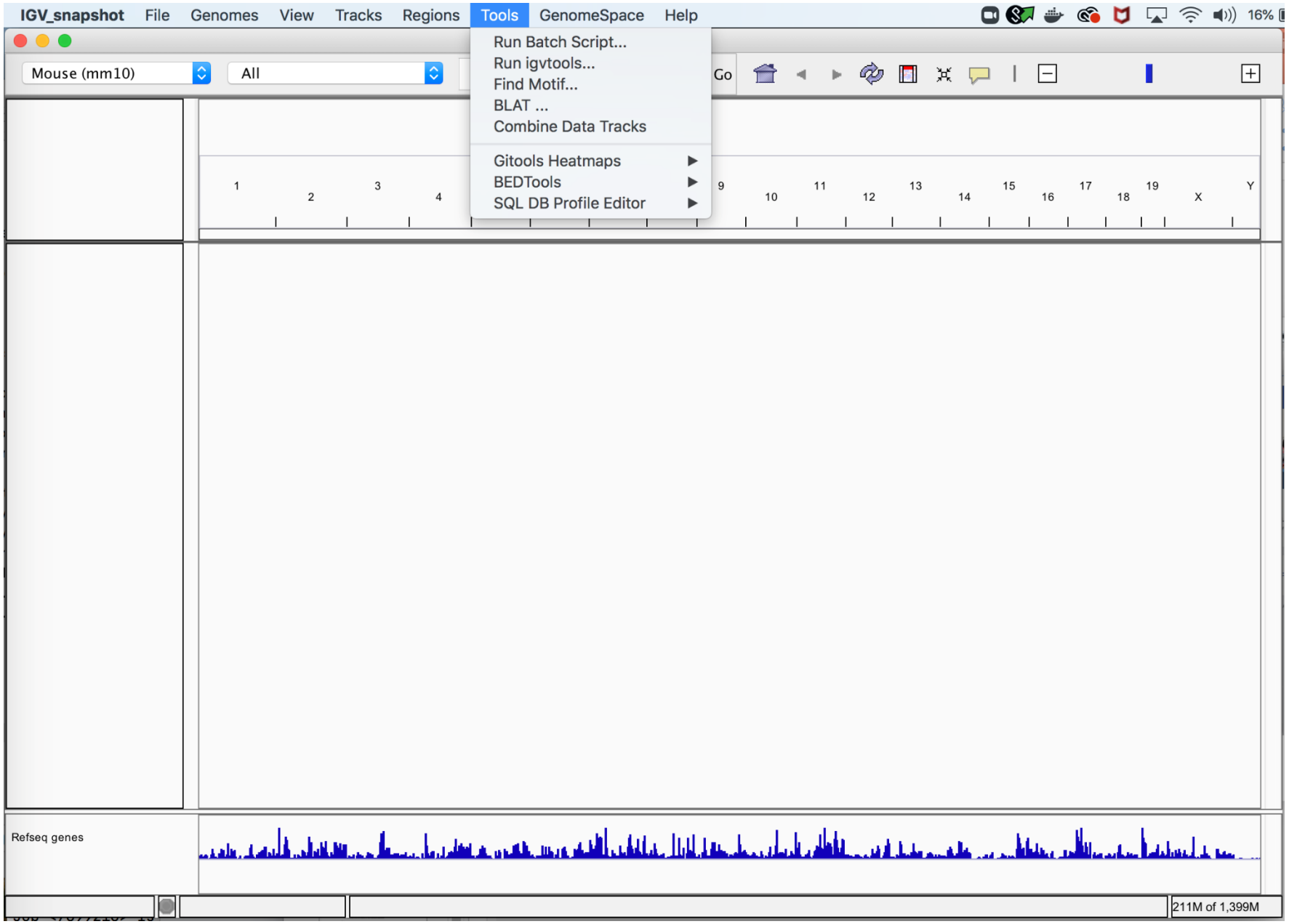
```
15:03:50.173 INFO  NativeLibraryLoader - Loading libgkl_compression.so from jar:file:/share/pkg/picard/2.17.8/picard.jar!/com/intel/gkl/native/libgkl_compression.so
[Mon Jun 22 15:03:50 EDT 2020] BuildBamIndex INPUT=ProlnoCu_IN_rep1_sorted.dedup.q20.bam    VERBOSITY=INFO QUIET=false VALIDATION_STRINGENCY=STRICT COMPRESSION_LEVEL=5 MAX_RECORDS_IN_RAM=500000 CREATE_INDEX=false CREATE_MD5_FILE=false GA4GH_CLIENT_SECRETS=client_secrets.json USE_JDK_DEFLATER=false USE_JDK_INFLATER=false
[Mon Jun 22 15:03:50 EDT 2020] Executing as ss45w@c38b02 on Linux 2.6.32-754.14.2.el6.x86_64 amd64; Java HotSpot(TM) 64-Bit Server VM 1.8.0_77-b03; Deflater: Intel; Inflater: Intel; Picard version: 2.17.8-SNAPSHOT
INFO    2020-06-22 15:04:16     BuildBamIndex   Successfully wrote bam index file /nl/umw_anthony_imbalzano/Sabriya/TereChIP/ProlnoCu_IN_rep1_sorted.dedup.q20.bai
[Mon Jun 22 15:04:16 EDT 2020] picard.sam.BuildBamIndex done. Elapsed time: 0.44 minutes.
Runtime.totalMemory()=10040639488

------------------------------------------------------------
Sender: LSF System <lsfadmin@c38b02>
Subject: Job 7699331: <java -jar /share/pkg/picard/2.17.8/picard.jar BuildBamIndex I=ProlnoCu_IN_rep1_sorted.dedup.q20.bam> in cluster <umghpcc> Done

Job <java -jar /share/pkg/picard/2.17.8/picard.jar BuildBamIndex I=ProlnoCu_IN_rep1_sorted.dedup.q20.bam> was submitted from host <ghpcc06> by user <ss45w> in cluster <umghpcc> at Mon Jun 22 15:03:45 2020
Job was executed on host(s) <c38b02>, in queue <short>, as user <ss45w> in cluster <umghpcc> at Mon Jun 22 15:03:45 2020
</home/ss45w> was used as the home directory.
</nl/umw_anthony_imbalzano/Sabriya/TereChIP> was used as the working directory.
Started at Mon Jun 22 15:03:45 2020
Terminated at Mon Jun 22 15:04:16 2020
Results reported at Mon Jun 22 15:04:16 2020

Your job looked like:

------------------------------------------------------------
# LSBATCH: User input
java -jar /share/pkg/picard/2.17.8/picard.jar BuildBamIndex I=ProlnoCu_IN_rep1_sorted.dedup.q20.bam
------------------------------------------------------------

Successfully completed.

Resource usage summary:

    CPU time :                                   26.58 sec.
    Max Memory :                                 4284 MB
    Average Memory :                             3222.75 MB
    Total Requested Memory :                     20000.00 MB
    Delta Memory :                               15716.00 MB
    Max Swap :                                   -
    Max Processes :                              3
    Max Threads :                                54
    Run time :                                   45 sec.
    Turnaround time :                            31 sec.

The output (if any) is above this job summary.

~
~
~
(END)
```

My ChIP-Sequencing Workflow

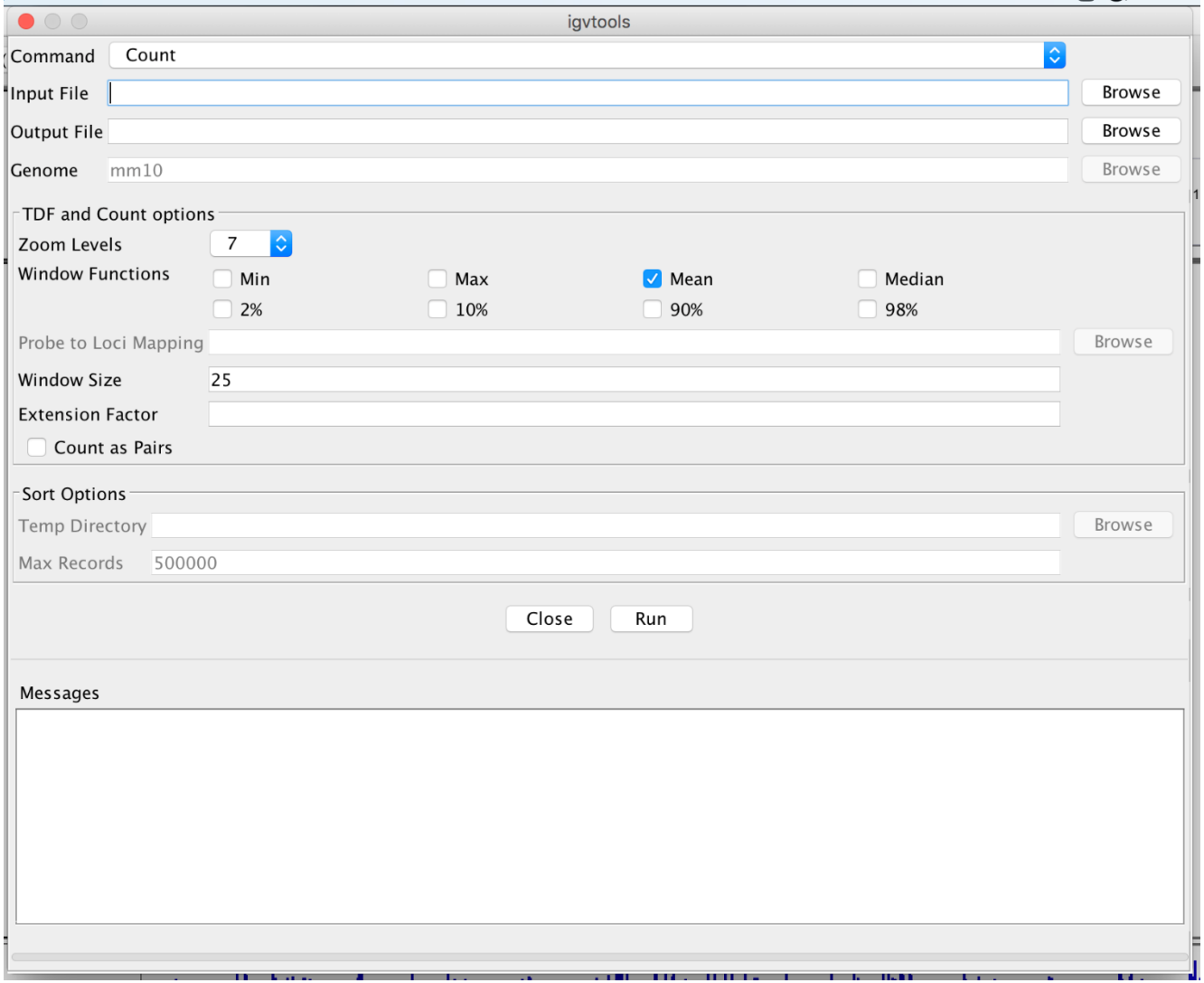




Select mm10 genome

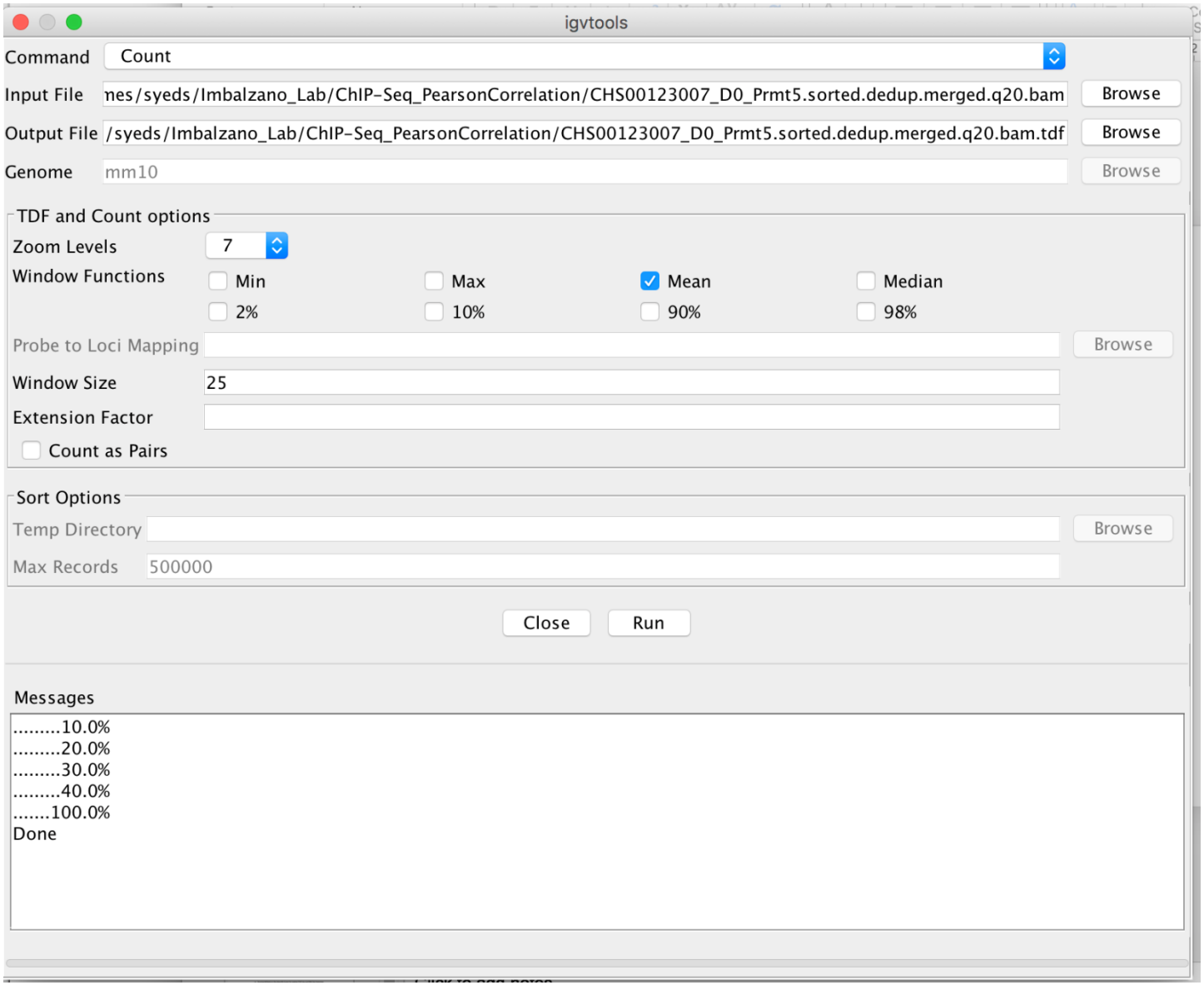


Select Tools → Run Igvtools



Input file: your bam
Output file: will keep same naming but append a .tdf

Then select Run. It will take 5-10 minutes to generate the .tdf file



.tdf file can be loaded into IGV easily for viewing
Several .tdf files can be loaded together and saved as a session as well.

Week 3

- Unix Problem Set Q&A
- Review ChIP-Seq and RNA-Seq pipelines

- Loading files to the cluster using FileZilla and rsync

- On the ghpcc06 cluster, read criteria for running bowtie2, picard, samtools, macs2, etc.
- See sample bsub commands for running each module
- Look at bowtie2 and macs2 output files
- Use IGV to view .bam and .bed files

- On the ghpcc06 cluster, read criteria for running rsem-calculate-expression on the cluster
  https://bioinfo.umassmed.edu/index.php?p=33
- RNA-Seq – Look at RSEM output files
- See sample bsub commands for rsem-calculate-expression

Tasks:
Create lab schedule for using the ghpcc06 cluster
Explore the cluster and examine modules and tools for data analysis that are available
Upload Fastq files to ghpcc06 cluster

Week 4

- Introducing NGSplot: Quick mining and visualization of NGS data by integrating genomic databases
  https://github.com/shenlab-sinai/ngsplot
- Introducing HOMER: Software for motif discovery and next-gen sequencing analysis
  http://homer.ucsd.edu/homer/ngs/

- Running NGSplot on the ghpcc06 cluster for making ChIP-Seq and RNA-Seq heatmaps and extracting tag density files
- Running HOMER on the ghpcc06 cluster for annotating bed files and performing motif analysis
- Running bedtools on the ghpcc06 cluster for comparing .bed files of different experiments