HISAT2

graph-based alignment of next generation sequencing reads to a population of genomes

HISAT2 is a fast and sensitive alignment program for mapping next-generation sequencing reads (both DNA and RNA) to a population of human genomes as well as to a single reference genome. Based on an extension of BWT for graphs (Sirén et al. 2014), we designed and implemented a graph FM index (GFM), an original approach and its first implementation. In addition to using one global GFM index that represents a population of human genomes, **HISAT2** uses a large set of small GFM indexes that collectively cover the whole genome. These small indexes (called local indexes), combined with several alignment strategies, enable rapid and accurate alignment of sequencing reads. This new indexing scheme is called a Hierarchical Graph FM index (HGFM).

The HISAT-3N paper published at *Genome Research*. 7/1/2021

HISAT-3N beta release 12/14/2020

HISAT-3N is a software system for analyzing nucleotide conversion sequencing reads. See the HISAT-3N for more details.

Search

- Main
- About
- Manual
- HISAT-3N
- Download
- HowTo
- Links

Unit 2: Mapping with HISAT2

February 23, 2023

First, grab these files:

```
cp -r /gpfs1/cl/mmg232/course_materials/HISAT2_example .
```

Thursday, February 23rd

Time	Topic
~20 mins	More BASH commands
~30 mins	HISAT2: intro & script

Homework Assignment 7 Due: Selection of NGS dataset (120 points)

Inside HISAT2_example folder

(2) FASTQ files

(1) Script called hisat2_align.sh

Run script:

sbatch hisat2_align.sh

Outline for today

~10 mins	Part 1 Learning more UNIX command
~15 mins	Part 2 Mapping with HISAT2 Lecture
~20 mins	Part 3 Writing & understanding the code that we just submitted using sbatch

More bash!

Please grab these files too –

```
cp –r /gpfs1/cl/mmg232/course_materials/six_commands/ .
```

Thursday, February 23rd

Time	Topic
~20 mins	More BASH commands
~30 mins	HISAT2: intro & script

Homework Assignment 7 Due: Selection of NGS dataset (120 points)

PART 2

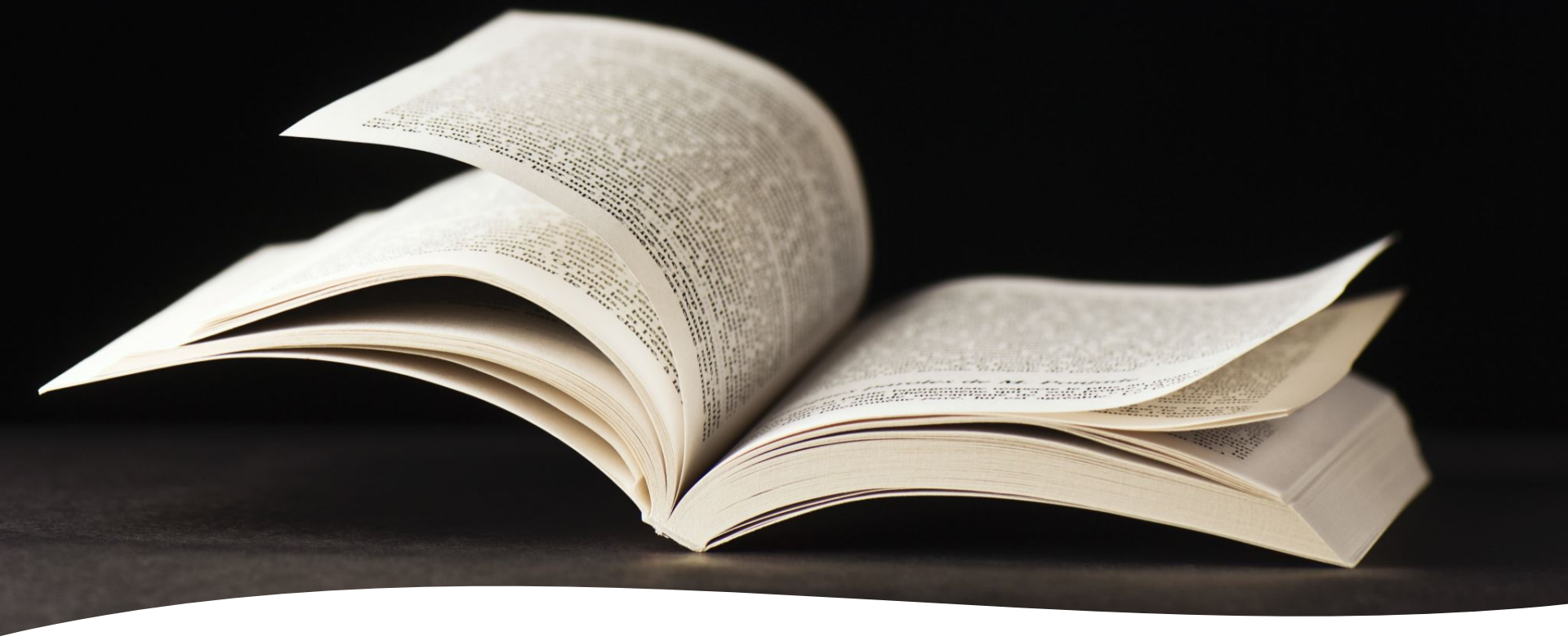
Mapping with HISAT2

Learning objectives

- Explore the splice-aware mapper called HISAT2
- Create a bash script used to align and generate the desired outputs using HISAT2 and SAMtools programs
- *Understand the output files from HISAT2 & SAMtools*

ALIGNER	Designed for (Type of Data)	Programming Language	Algorithm	Mode	Clip
BOWTIE2	DNA, RNA	C++	BWT	Local	Soft
HISAT2	DNA, RNA	Python	BWT	Local	Soft
STAR	RNA	C++	BWT	Local	Soft
TOPHAT2	RNA	C++	BWT	Global	Soft

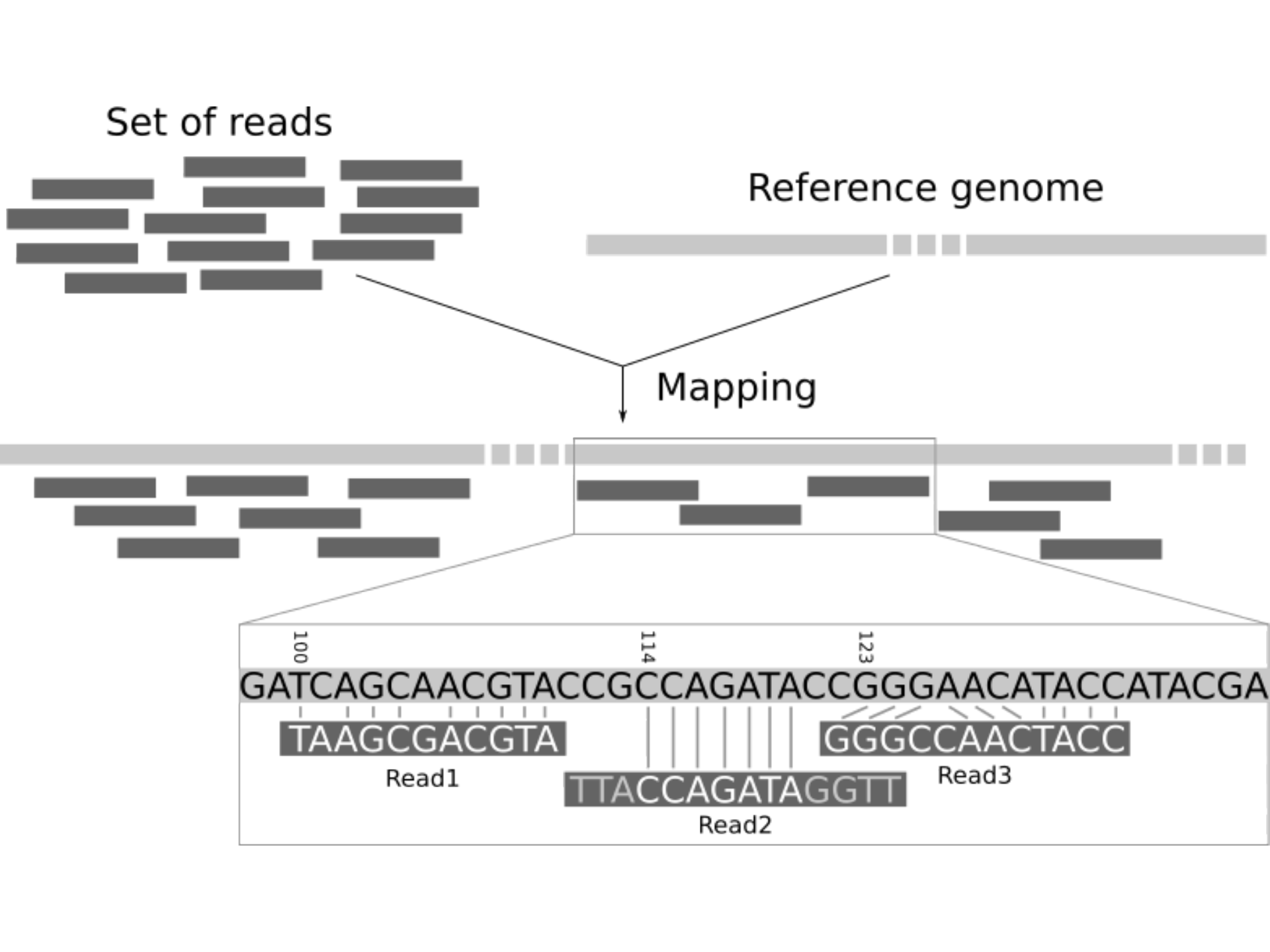
BWT = Burrows-Wheeler transform

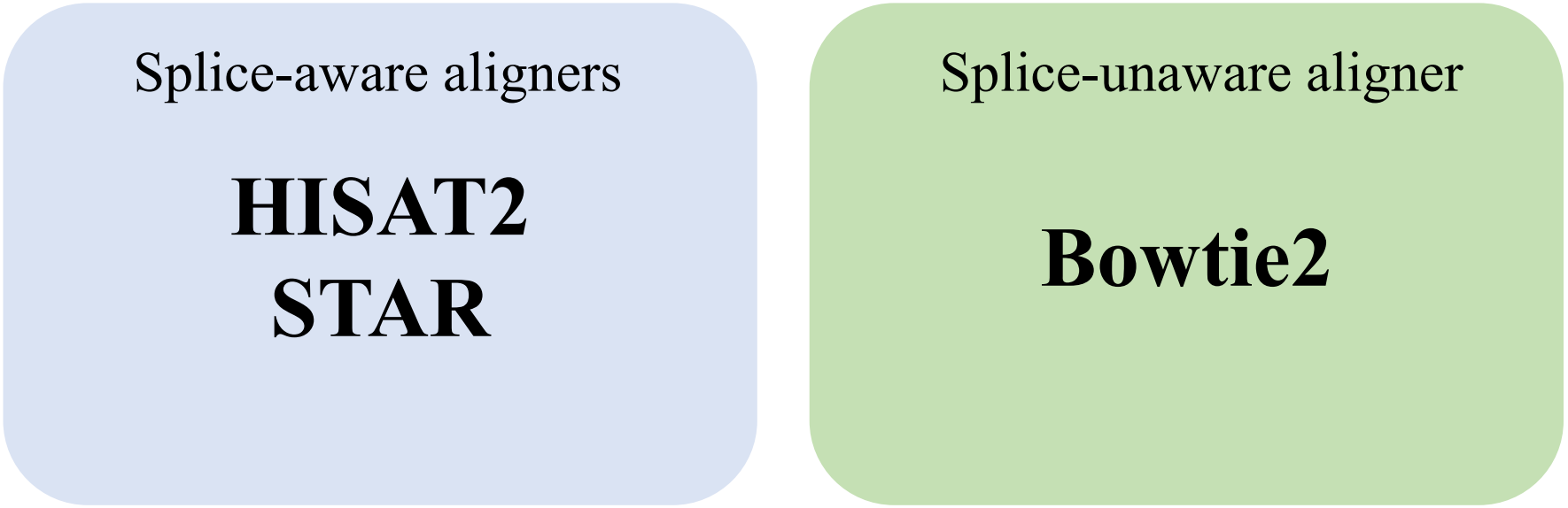


Indexing benefits

Indexing a genome can be explained **similar to indexing a book**.

If you want to know on which page a chapter begins, it is much more efficient to look up the page number in a pre-built index (table of contents) than going through every page of the book until you found the chapter you are looking for.


Set of reads
Reference genome
Mapping
100
114
123
GATCAGCAACGTACCGCCAGATACCGGGGAACATACCATACGA
TAAGCGACGTA
Read1
TTACCAGATAGGTT
Read2
GGGCCAACTACC
Read3



RNA-Seq

ChIP-Seq

What does it mean to be splice-aware?

- The major problem is that introns not only vary in length but that they can also be very long.

- A splice-unaware aligner will have to introduce a **long gap** in the mapping of a read to span an intron.

- Would know not to try to align reads to introns, and identifies exons and tries to align to those instead, ignoring introns altogether

Two groups of splice-aware aligners

- **Group 1:** Splice-aware aligners that use the genome sequence and known gene annotations to calculate gene or transcript abundance – these ***can not*** be used to identify new splice junctions

- **Group 2:** *de novo* splice-aware aligners which can align RNA-Seq reads to a reference genomic sequence without prior information on gene annotations
  - STAR (Spliced Transcripts Alignment to a Reference)

HISAT2

- Stands for **h**ierarchical **i**ndexing for **s**pliced **a**lignment of **t**ranscripts 2
- HISAT2 is an aligner that is used for mapping next-generation sequencing reads
  - Used for whole genome, whole-exome, and transcriptome datasets

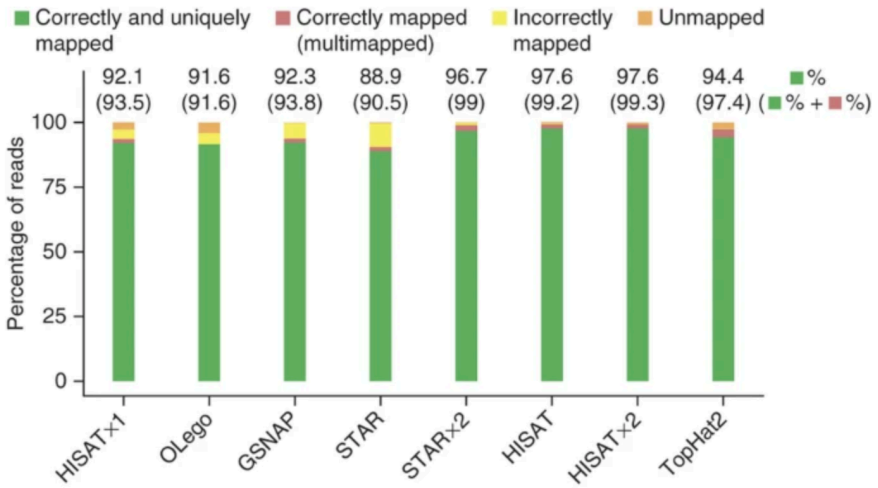
HISAT2 is fast

- Is the fastest spliced mapper currently available
- This is possible due to its novel indexing strategy
- Uses BWT algorithm to compress genomes therefore, requiring very little memory to store.
- But additionally, it builds these genomes using FM-indexing.
  - Using **both** makes it possible to search through genomes rapidly

HISAT2 has a small memory footprint

- The STAR program runs faster than TopHat2 but both have a memory requirement of ~28GB
- The memory requirement for HISAT2 is ~5GB
  - This makes it possible to do alignments on your laptop

Figure 3: Alignment accuracy of spliced alignment software for 20 million simulated 100-bp reads.



The dataset

nature COMMUNICATIONS

ARTICLE

https://doi.org/10.1038/s41467-021-26159-1 OPEN

Check for updates

Tcf1 and Lef1 provide constant supervision to mature CD8$^+$ T cell identity and function by organizing genomic architecture

Qiang Shan[1,5], Xiang Li[2,5], Xia Chen[3], Zhouhao Zeng[2], Shaoqi Zhu[2], Kexin Gai[1], Weiqun Peng[2✉] & Hai-Hui Xue[1,4✉]

T cell identity is established during thymic development, but how it is maintained in the periphery remains unknown. Here we show that ablating Tcf1 and Lef1 transcription factors in mature CD8$^+$ T cells aberrantly induces genes from non-T cell lineages. Using high-throughput chromosome-conformation-capture sequencing, we demonstrate that Tcf1/Lef1 are important for maintaining three-dimensional genome organization at multiple scales in CD8$^+$ T cells. Comprehensive network analyses coupled with genome-wide profiling of chromatin accessibility and Tcf1 occupancy show the direct impact of Tcf1/Lef1 on the T cell genome is to promote formation of extensively interconnected hubs through enforcing chromatin interaction and accessibility. The integrative mechanisms utilized by Tcf1/Lef1 underlie activation of T cell identity genes and repression of non-T lineage genes, conferring fine control of various T cell functionalities. These findings suggest that Tcf1/Lef1 control global genome organization and help form intricate chromatin-interacting hubs to facilitate promoter-enhancer/silencer contact, hence providing constant supervision of CD8$^+$ T cell identity and function.

SRR_number	datatype	treatment	cell	replicate
SRR13423162	RNAseq	WT	CD8 T cell	1
SRR13423163	RNAseq	WT	CD8 T cell	2
SRR13423164	RNAseq	WT	CD8 T cell	3
SRR13423165	RNAseq	TCF1 - KO	CD8 T cell	1
SRR13423166	RNAseq	TCF1 - KO	CD8 T cell	2
SRR13423167	RNAseq	TCF1 - KO	CD8 T cell	3

SRR_number	datatype	treatment	cell	replicate
SRR13423162	RNAseq	WT	CD8 T cell	1
SRR13423163	RNAseq	WT	CD8 T cell	2
SRR13423164	RNAseq	WT	CD8 T cell	3
SRR13423165	RNAseq	TCF1 - KO	CD8 T cell	1
SRR13423166	RNAseq	TCF1 - KO	CD8 T cell	2
SRR13423167	RNAseq	TCF1 - KO	CD8 T cell	3

This script will contain:

- Variables
- Sed command
- For loop
- Module load
- hisat2
- samtools

Jupyter Notebook version: 04240a1

This app will launch a Jupyter Notebook server on one or more nodes.

Partition

bluemoon

- To request a GPU specify a partition such as dggpu or bdgpu

Number of hours (min-1, max-48)

1

Number of nodes (min-1, max-4)

1

Number of cores per node (min-1, max-32)

1

Number of GPUs per node (min-0, max-2)

0

- If requesting GPU nodes, you must enter a GPU-enabled Partition above or the job will fail.

Or use your favorite text editor!

Extra Modules

- Specify additional environment modules here (space delimited)

How to create a variable

nameofvarible=valueofvarible

To recall the contents of the variable

```
echo $nameofvariable
```

Utility of variables

Variables can be used to store information that can be used later in the script (once or many times over)

Example of Variable usage

```
DBDIR=/users/p/d/pdrodrig/genome_in
dex
```

In this case, this is showing a location to a specific directory

Curly braces {}

- https://devhints.io/bash

- Substitutions, manipulations, etc.. So much more!

backslash \

is used by *bash* to indicate a line continuation and is commonly used in *bash* scripts

Indenting and blank lines

- Indenting is done to clarify your code. Indentations are usually used for loops, if statements, function definitions to make it easy to see what statements are part of that loop or part of the if statement.

- Another trick to make your code more readable is adding blank lines to separate out blocks of related code.

for loop is classified as an iteration statement

For loops are constructed with 4 basic words:

Words	What it does
for	set the loop variable name
in	specify whatever it is we are looping over
do	specify what we want to do with each item
done	tell the computer we are done

for loop syntax

putting these together, for loop syntax is as follows:

```
for VARIABLE in file1 file2 file3
do
  command1 on $VARIABLE
  command2
done
```

A basic loop looks something like this when it's written within a job script:

```
for i in A B C
do
  echo $i
done
```

HISAT2 usage

- http://daehwankimlab.github.io/hisat2/

- hisat2 [options]* -x <hisat2-idx> {-1 <m1> -2 <m2> | -U <r> | --sra-acc <SRA accession number>} [-S <hit>]

HISAT2 usage

- -x <hisat2-idx>
  The basename of the index for the reference genome. The basename is the name of any of the index files up to but not including the final .1.ht2 / etc. hisat2 looks for the specified index first in the current directory, then in the directory specified in the HISAT2_INDEXES environment variable.

/gpfs1/cl/mmg232/course_materials/hisat2_index

```
[pdrodrig@vacc-user1 hisat2_index]$ ls
align_human_7376182.out  GRCh38.3.ht2  GRCh38.8.ht2  GRCm39.5.ht2
align_mouse_7375972.out  GRCh38.4.ht2  GRCm39.1.ht2  GRCm39.6.ht2
align_mouse_7375976.out  GRCh38.5.ht2  GRCm39.2.ht2  GRCm39.7.ht2
GRCh38.1.ht2             GRCh38.6.ht2  GRCm39.3.ht2  GRCm39.8.ht2
GRCh38.2.ht2             GRCh38.7.ht2  GRCm39.4.ht2  hisat2_index_script
```

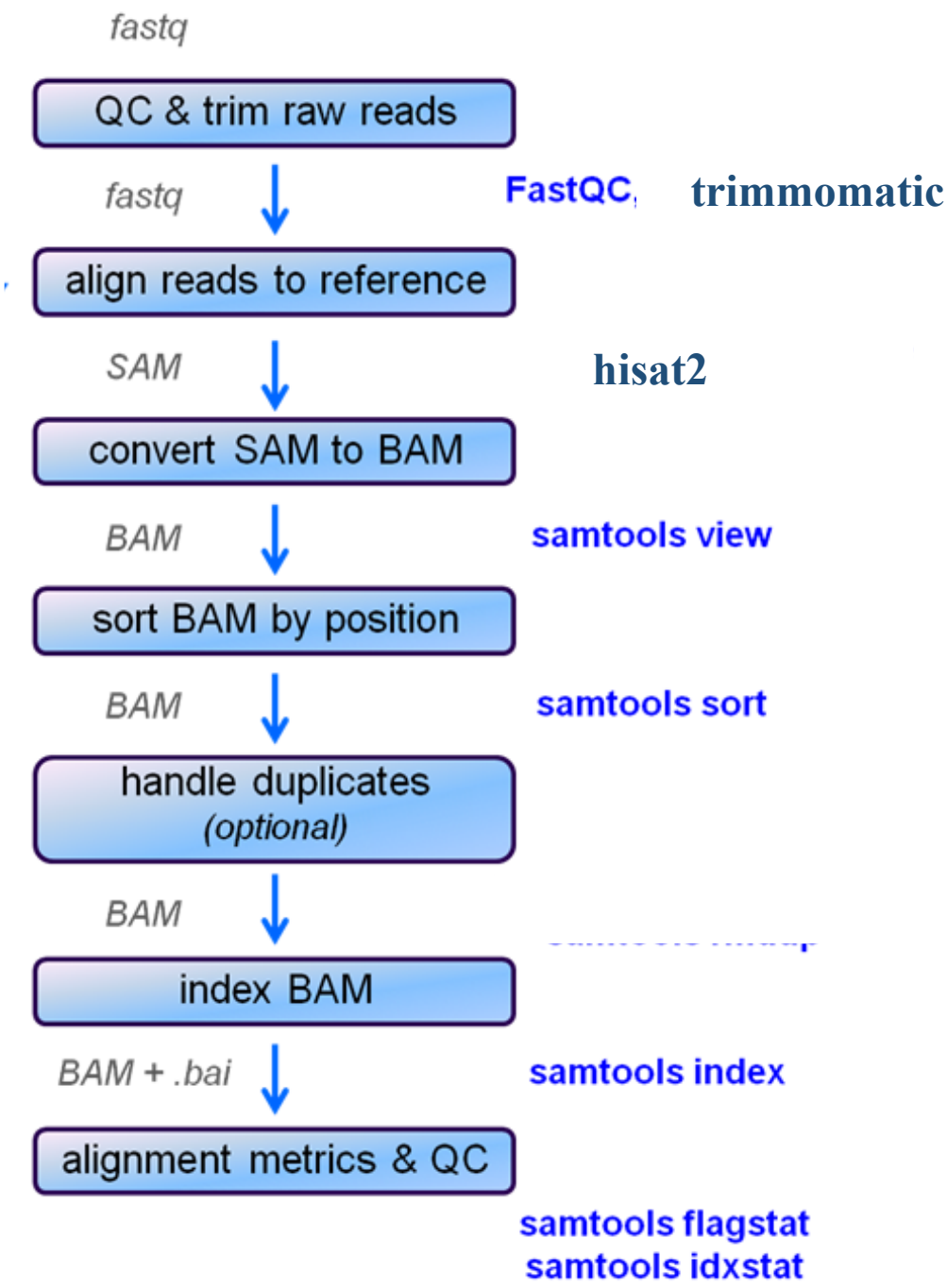
HISAT2 usage

- -1 <m1>
  Comma-separated list of files containing mate 1s (filename usually includes _1), e.g. -1 flyA_1.fq,flyB_1.fq. Sequences specified with this option must correspond file-for-file and read-for-read with those specified in <m2>. Reads may be a mix of different lengths. If - is specified, hisat2 will read the mate 1s from the “standard in” or “stdin” filehandle.
- -2 <m2>
  Comma-separated list of files containing mate 2s (filename usually includes _2), e.g. -2 flyA_2.fq,flyB_2.fq. Sequences specified with this option must correspond file-for-file and read-for-read with those specified in <m1>. Reads may be a mix of different lengths. If - is specified, hisat2 will read the mate 2s from the “standard in” or “stdin” filehandle.

HISAT2 usage

- -U <r>
  Comma-separated list of files containing unpaired reads to be aligned, e.g. lane1.fq,lane2.fq,lane3.fq,lane4.fq. Reads may be a mix of different lengths. If - is specified, hisat2 gets the reads from the “standard in” or “stdin” filehandle.
- -S <hit>
  File to write SAM alignments to. By default, alignments are written to the “standard out” or “stdout” filehandle (i.e. the console).

Alignment Workflow



SAMtools usage

- http://www.htslib.org/doc/samtools.html

samtools view [*options*] *in.sam*|*in.bam*|*in.cram* [*region*...]

samtools flagstat *in.sam*|*in.bam*|*in.cram*

samtools sort [**-l** *level*] [**-u**] [**-m** *maxMem*] [**-o** *out.bam*] [**-O** *format*] [**-M**] [**-K** *kmerLen*] [**-n**] [**-t** *tag*] [**-T** *tmpprefix*] [**-@** *threads*] [*in.sam*|*in.bam*|*in.cram*]

samtools index [**-bc**] [**-m** *INT*] *aln.sam*|*aln.bam*|*aln.cram* [*out.index*]