



**TEXAS WOMAN'S
UNIVERSITY**

**Project ACCESS Bioinformatics Workshop
Summer 2024 – July 29, 30; 2024
Micah Andrew Thornton ©**

Who am I?

- B.Sc. - Statistical Science & Computer Engineering (SMU, 2017)
- M.S – Computer Engineering (SMU, 2017)
- Ph. D. - Biostatistics (SMU/UTSW, 2021)
- Post-Doctoral Research Certification (UTSW 2024) –
 - Daehwan Kim Lab (UTSW, 2021-2023)
 - Sequence Alignment and Statistical Models
 - Lee Kraus Lab (UTSW, 2023)
 - Statistical Models for RPol-II Propagation



What is this Workshop?

- Hosted by Texas Woman's University's Project ACCESS
- Held over two days (July 29, 30; 2024)
- Using the tools of Bioinformatics:
 - Gives introduction and theory background.
 - Provides hands-on tool use experience.



Day 1 (July 29, 2024)

- Focus on the basics:
 - Introduction:
 - What is/is not Bioinformatics?
 - Assembly/Human Reference Genome
 - Tools of Bioinformatics:
 - Standard File Formats
 - Genome Browsers
 - Multiple Sequence Alignment - BLAST



Day 2 (July 30, 2024)

- Command-line Tools:
 - Review of Day 1
 - Linux – History and Basic Utilities
 - Sequence Aligners – HISAT2
 - SAMTOOLS and BCFTOOLS
- R Tools:
 - DESeq2



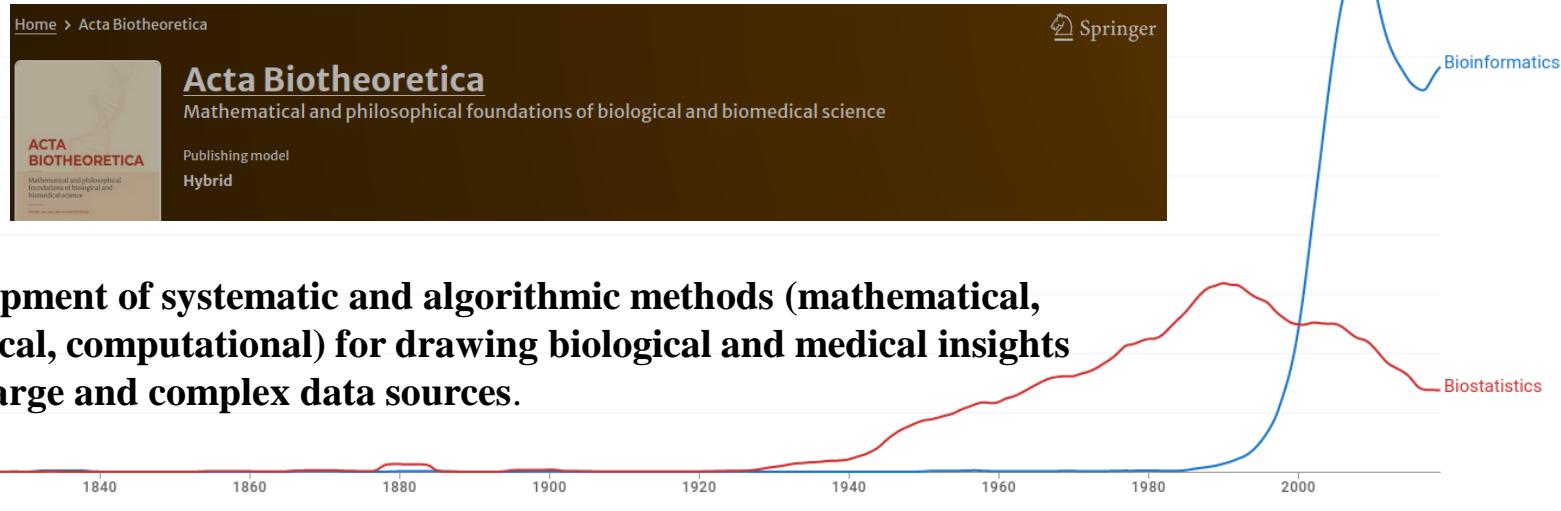
Day 1



TEXAS WOMAN'S
UNIVERSITY

Introduction to Bioinformatics

- First recorded use of the term Bioinformatics was in 1970.



- Development of systematic and algorithmic methods (mathematical, statistical, computational) for drawing biological and medical insights from large and complex data sources.
- Lin, Y., Michel, J.-B., Aiden, E. L., Orwant, J., Brockman, W., and Petrov, S. (2012). Syntactic annotations for the Google Books Ngram Corpus. Proceedings of the ACL 2012 system demonstrations, 169-174.



History of Bioinformatics

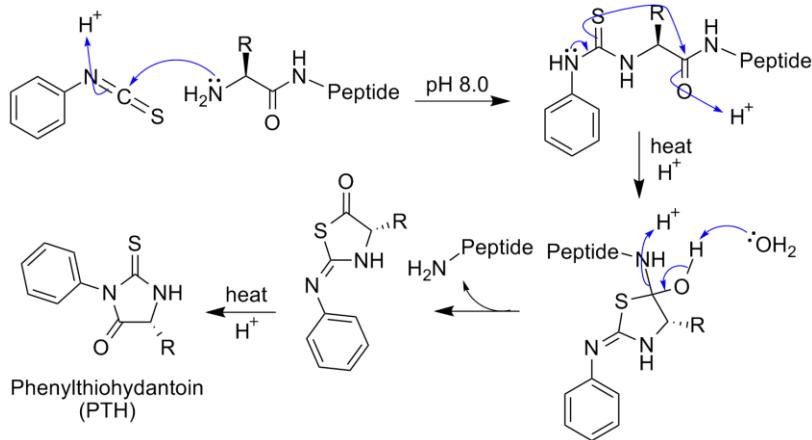
- 1950
 - 1960
 - 1970
 - 1980
 - 1990
 - 2000
- 
- (late 1950s) – Edman (Pehr Edman) Protein Sequencing – ~60 AA segments



History of Bioinformatics



- Phenyl Isothiocyanate + • N-Terminus Peptide

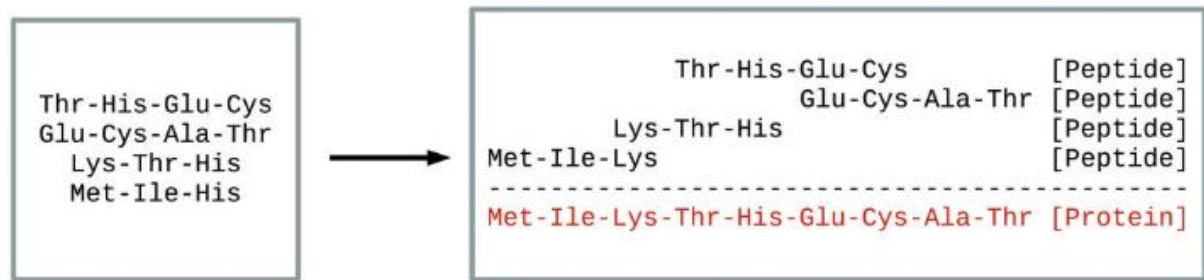


History of Bioinformatics

- 1950
- 1960
 - (late 1950s) – Edman (Pehr Edman) Protein Sequencing – ~60 AA segments
- 1970
 - (1962) - COMPROTEIN Developed (Margaret Oakley Dayhoff) – Allowed Assembling of peptide sequences into larger proteins
- 1980
- 1990
- 2000



History of Bioinformatics



- Jeff Gauthier, Antony T Vincent, Steve J Charette, Nicolas Derome, A brief history of bioinformatics, *Briefings in Bioinformatics*, Volume 20, Issue 6, November 2019, Pages 1981–1996, <https://doi.org/10.1093/bib/bby063>



History of Bioinformatics

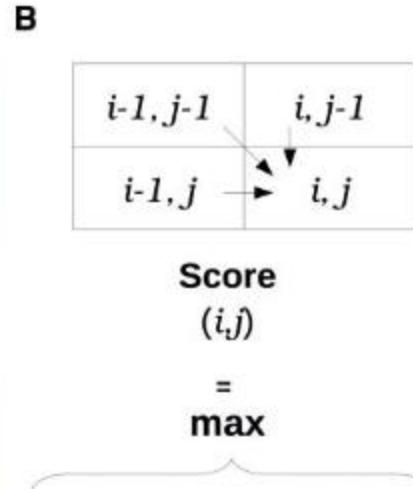
- 1950
 - (late 1950s) - Edman (Pehr Edman) Protein Sequencing – ~60 AA segments
- 1960
 - (1962) - COMPROTEIN Developed (Margaret Oakley Dayhoff) – Allowed Assembling of peptide sequences into larger proteins
- 1970
 - **(1970) – Needleman and Wunsch – Develop Dynamic Programming Approach for computing Sequence Dissimilarity + End to End Alignment.**
- 1980
- 1990
- 2000



History of Bioinformatics

A match +5 mismatch -4 gap -1

	A	T	C	G	
A	0	5	-1	-1	-1
T	0	4	10	9	8
G	0	3	9	8	14



- Score $(i-1, j-1)$ + Match / Mismatch
- Score $(i, j-1)$ + gap
- Score $(i-1, j)$ + gap

C
Best Alignment : ATCG
(Score = 38)
|||
AT G



- Jeff Gauthier, Antony T Vincent, Steve J Charette, Nicolas Derome, A brief history of bioinformatics, *Briefings in Bioinformatics*, Volume 20, Issue 6, November 2019, Pages 1981–1996,
<https://doi.org/10.1093/bib/bby063>



History of Bioinformatics

- 1950
 - (late 1950s) - Edman Protein Sequencing – ~60 AA segments
 - 1960
 - (1962) - COMPROTEIN Developed – Allowed Assembling of peptide sequences into larger proteins
 - 1970
 - (1970) – Needleman and Wunsch – Develop Dynamic Programming Approach for computing Sequence Dissimilarity.
 - * (1981) – Smith and Waterman – Build upon Needleman and Wunsch for Local Alignment
 - 1990
 - 2000
- * 1975 – Sanger Sequencing for DNA invented



History of Bioinformatics



- Updated Needleman-Wunsch (NW) in 2 ways:
 - Add a fixed minimum (say 1)
 - Terminate Local alignments when fixed minimum met/Start anywhere in matrix



History of Bioinformatics

	Y	R	M	N	P	Y	C	N	M
0	0	0	0	0	0	0	0	0	0
Y	0	5	4	3	2	1	5	4	3
C	0	4	3	2	1	1	4	10	9
N	0	3	2	1	7	6	3	9	15
P	0	2	1	1	6	12	2	8	14

Local alignment for 'YCNP' in the longer but similar string as before
'YRMNPYCNM'

Alignment 1:
 $(15 + 10 + 5 = 30)$
'YRMNPYCNM'
,

Alignment 2:
 $(19+6+5 = 30)$
'YRMNPYCNM'
'Y-CNP'



History of Bioinformatics

- 1950
 - (late 1950s) - Edman Protein Sequencing – ~60 AA segments
- 1960
 - (1962) - COMPROTEIN Developed – Allowed Assembling of peptide sequences into larger proteins
- 1970
 - (1970) – Needleman and Wunsch – Develop Dynamic Programming Approach for computing Sequence Dissimilarity.
- 1980
 - (1981) – Smith and Waterman – Build on Needleman and Wunsch (Local Alignment)
 - (1985) – **FASTA Algorithm and files created by Lipman and Pearson.**
- 1990
 - (1989) – **BLAST Algorithm created (Myers, Altschul, Gish, Lipman, and Miller).**
- 2000
 -



History of Bioinformatics

- 1950
 - (late 1950s) - Edman Protein Sequencing – ~60 AA segments
- 1960
 - (1962) - COMPROTEIN Developed – Allowed Assembling of peptide sequences into larger proteins
- 1970
 - (1970) – Needleman and Wunsch – Develop Dynamic Programming Approach for computing Sequence Dissimilarity.
- 1980
 - (1981) – Smith and Waterman – Build on Needleman and Wunsch (Local Alignment)
 - (1985) – FASTA Algorithm and files created by Lipman and Pearson.
- 1990
 - (1989) – BLAST Algorithm created (Myers, Altschul, Gish, Lipman, and Miller).
- 2000
 - (1994) – ClustalW released for practical Multiple Sequence Alignment



History of Bioinformatics

- 2000
 - (2003) – Human Reference Genome Version 1 Completed.
 - (2005) – Next Generation Sequencing Introduced.
- 2010
- 2020



History of Bioinformatics

- 2000
 - (2003) – Human Reference Genome Version 1 Completed.
 - (2005) – Next Generation Sequencing Introduced.
- 2010
 - (2009) – Bowtie (Ben Langmead, Steven Salzberg) Introduced.**
- 2020



History of Bioinformatics

- 2000
 - (2003) – Human Reference Genome Version 1 Completed.
 - (2005) – Next Generation Sequencing Introduced.
- 2010
 - (2009) – Bowtie (Ben Langmead, Steven Salzberg) Introduced.
 - (2015) – HISAT (Daehwan Kim, Ben Langmead, Steven Salzberg) Introduced.**
- 2020



History of Bioinformatics

- 2000
 - (2003) – Human Reference Genome Version 1 Completed.
 - (2005) – Next Generation Sequencing Introduced.
- 2010
 - (2009) – Bowtie (Ben Langmead, Steven Salzberg) Introduced.
 - (2015) – HISAT (Daehwan Kim, Ben Langmead, Steven Salzberg) Introduced.
- 2020
 - (2019) – HISAT2 (Daehwan Kim, Ben Langmead, Steven Salzberg) Introduced.**



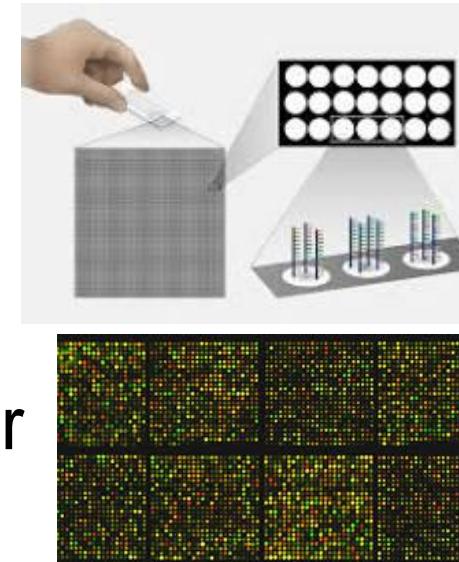
History of Bioinformatics

- 2000
 - (2003) – Human Reference Genome Version 1 Completed.
 - (2005) – Next Generation Sequencing Introduced.
- 2010
 - (2009) – Bowtie (Ben Langmead, Steven Salzberg) Introduced.
 - (2015) – HISAT (Daehwan Kim, Ben Langmead, Steven Salzberg) Introduced.
- 2020
 - (2019) – HISAT2 (Daehwan Kim, Ben Langmead, Steven Salzberg) Introduced.
 - (2021) – HISAT3N (Y. Zhang, C. Park, C. Bennett, M. Thornton, and D. Kim).**



Sequencing Technologies – MicroArrays

- Older Technology, used 1995 - ~2010.
- Detects genetic expression through oligonucleotide probes
- Fluorescence intensity used as proxy for sequence density.



- <https://www.youtube.com/watch?v=NgRfc6atXQ8>
- https://en.wikipedia.org/wiki/DNA_microarray

Credit NIH NHGRI &
National Cancer Institute



Sequencing Technologies – NGS Short Reads

- Newer technology, used 2000 - ~2024
- Detects sequence directly through fluorescently labeled nucleotides.
- Produces many short “reads” – subsequences of original strand
 - <https://www.youtube.com/watch?v=CZeN-lgjYCo>



Credit Illumina



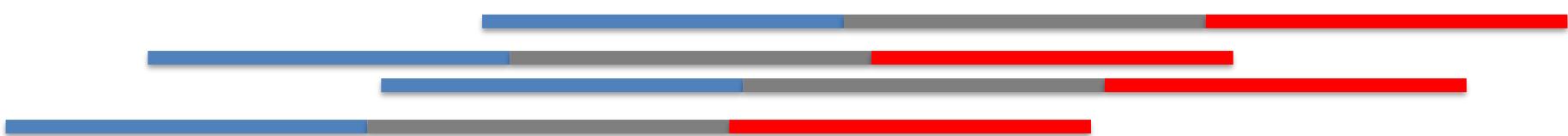
TEXAS WOMAN'S
UNIVERSITY

Paired end vs Single Reads

- Reads can be sequenced as a single read



- Or paired end reads with “inserts”

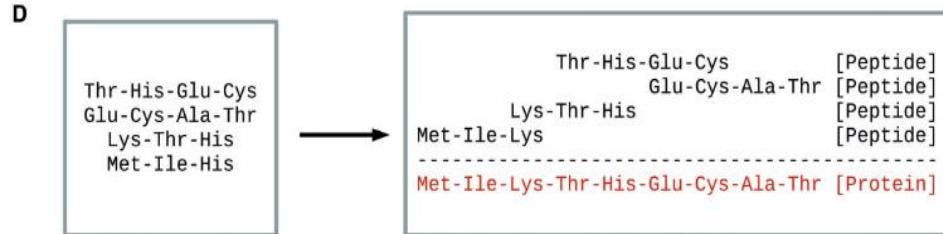


- Generated by repeating last few steps of NGS



Assembly

- Two types:
 - Reference-guided
 - De Novo (seen for first time)
- Before reference was created we had to use De novo assembly methods!



Recall Dayhoff's Comprotein!



De Novo Assembly – Naïve Approach

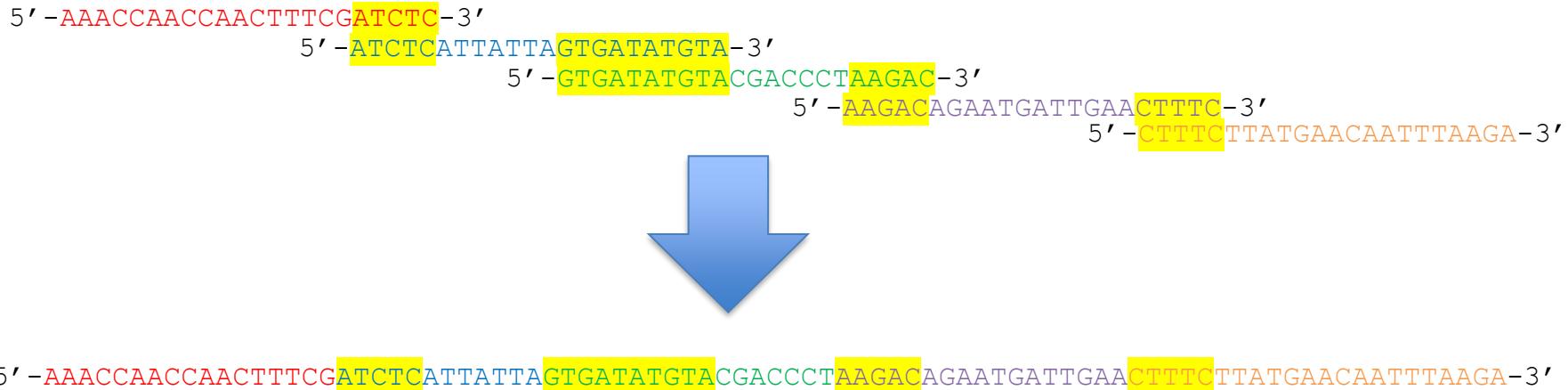
5' -AACCAACCAACTTTGATCTC-3'
5' -ATCTCATTATTAGTGATATGTA-3'
5' -GTGATATGTACGACCCTAAGAC-3'
5' -AAGACAGAATGATTGAACCTTTC-3'
5' -CTTTCTTATGAACAATTAAAGA-3'

Adapted from Robert Edwards' San Diego
State University Course



TEXAS WOMAN'S
UNIVERSITY

De Novo Assembly – Naïve Approach



Adapted from Robert Edwards' San Diego
State University Course



TEXAS WOMAN'S
UNIVERSITY

De Novo Assembly – Assumptions

- There are four nucleotides $\{A, C, T/U, G\}$
 - This means for a “ k -mer” of length k the total number of possibilities are 4^k .
 - Therefore, in the last problem,
 - by random chance alone,
 - If selecting a random 5-mer,
 - we would expect a “5-mer”, such as the 1,3, and 4th to occur with probability $p_i = \frac{1}{4^5} = \frac{1}{1024} = 0.098\%$
- The previous data came from the SARS-CoV2 genome, which is approximately 30,000 nucleotides long, and thus contains approximately 29,995 contiguous and overlapping 5-mers.
- Therefore we would expect to see the 5-mer “CTTTC”, $\frac{29995}{1024} \approx 30$ times,
 - We do observe this sequence exactly 30 times in SARS-CoV2’s Genome, however, more common mers such as “ATTAA” we observe closer to 50 times, or 166% of the expectation.



De Novo Assembly – Naïve Approach

- Pros:
 - Simple to understand,
 - Easy to implement
- Cons:
 - Short segments often repeated in sequences
 - Organisms are diploid (or polyploid)
 - Mismatches cause misassembles



De Novo Assembly – Greedy Approach

- Start with one sequence,
 - Match to all other sequences,
 - If there is a match extend the overall sequence until you cannot extend any further, this forms a “contig”
 - Exclude all sequences within that contig,
 - Repeat until all contigs found



De Novo Assembly – OLC Approach

- Overlap Layout Consensus –
 - Perform Multiple Sequence Alignment on Reads and then take the consensus (most commonly occurring base).



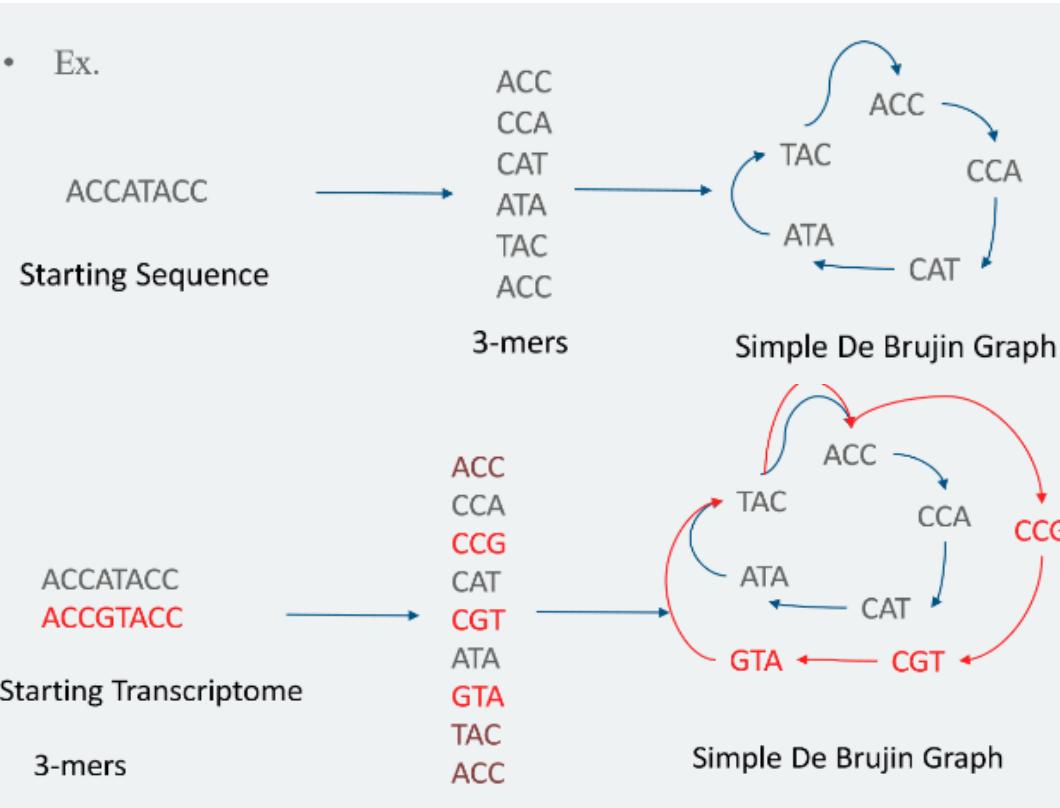
De Novo Assembly – De Brujin Approach

- De Brujin Graphs can be used to represent a sequence or a subsequence of a sequence in a mathematical graph structure.



De Brujin Graphs

- Ex.



- De Brujin graphs are a simple mathematical structure that can be used to represent a sequence by sub-sequences
- The “colored” version of the debrujin graph represents sets of sequences by different “colors”
- We will see these again tomorrow for pseudo-alignment!



De Bruijn Graph Assembly

5' - AACCGGTTA - 3'

5' -GGTTATAAC - 3'

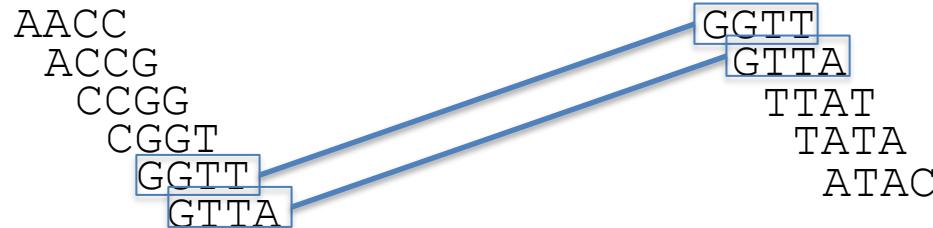


TEXAS WOMAN'S
UNIVERSITY

De Bruijn Graph Assembly

5' - AACCGGTTA - 3'

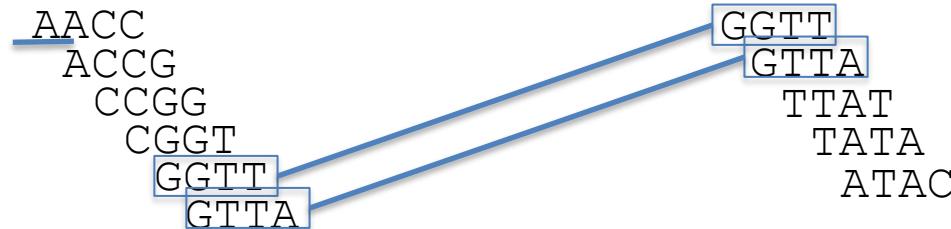
5' - GGTTATAC - 3'



De Bruijn Graph Assembly

5' - AACCGGTTA - 3'

5' - GGTTATAC - 3'



5' - A - 3'

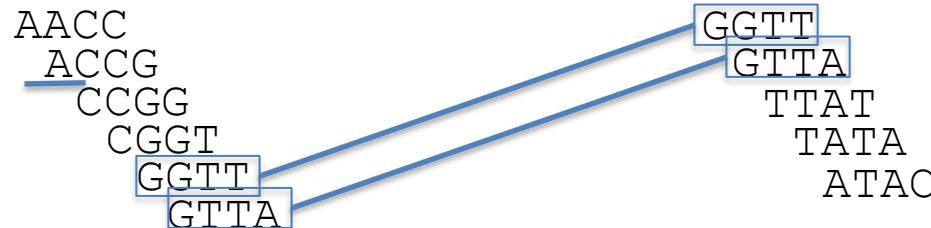


TEXAS WOMAN'S
UNIVERSITY

De Bruijn Graph Assembly

5' - AACCGGTTA - 3'

5' - GGTTATAC - 3'



5' - AA - 3'

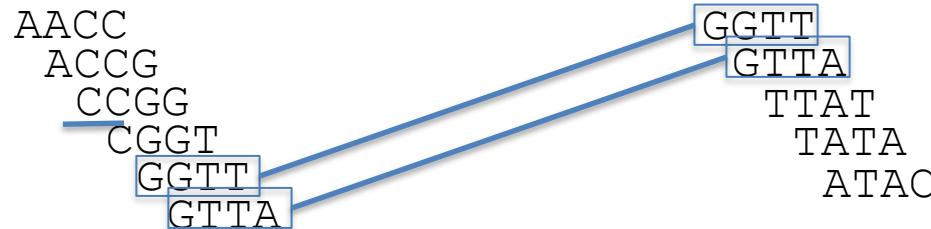


TEXAS WOMAN'S
UNIVERSITY

De Bruijn Graph Assembly

5' - AACCGGTTA - 3'

5' - GGTTATAC - 3'



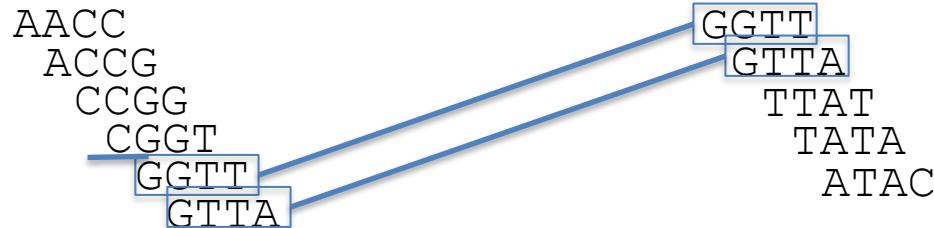
5' - AAC - 3'



De Bruijn Graph Assembly

5' - AACCGGTTA - 3'

5' - GGTTATAC - 3'



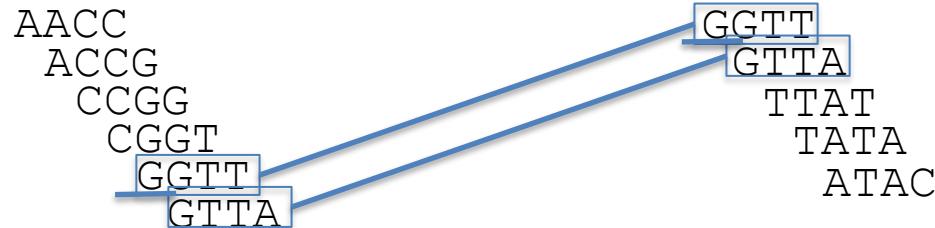
5' - AACCC - 3'



De Bruijn Graph Assembly

5' - AACCGGTTA - 3'

5' - GGTTATAC - 3'



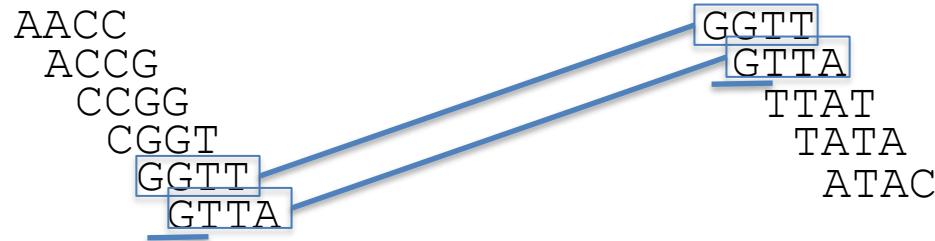
5' - AACCG - 3'



De Bruijn Graph Assembly

5' - AACCGGTTA - 3'

5' - GGTTATAC - 3'



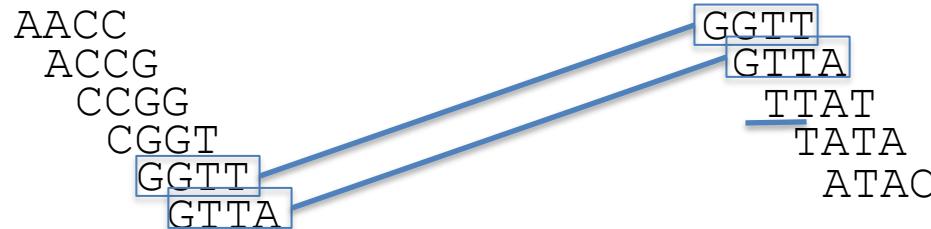
5' - AACCGG - 3'



De Bruijn Graph Assembly

5' - AACCGGTTA - 3'

5' - GGTTATAC - 3'



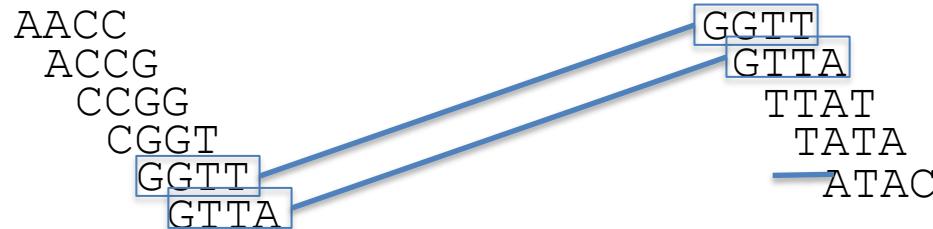
5' - AACCGGT - 3'



De Bruijn Graph Assembly

5' - AACCGGTTA - 3'

5' - GGTTATAC - 3'



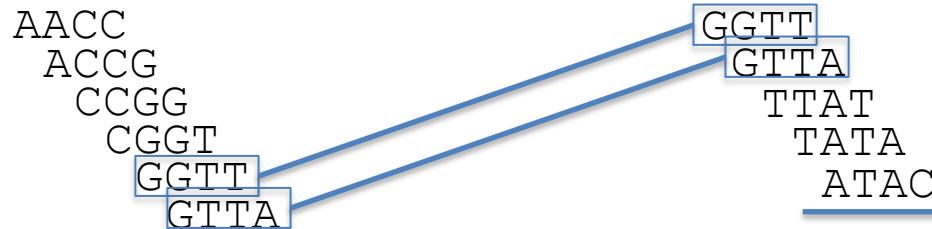
5' - AACCGGTT - 3'



De Bruijn Graph Assembly

5' - AACCGGTTA - 3'

5' - GGTTATAC - 3'



5' - AACCGGTTATAC - 3'



Building Reference Genome

- Reference Genomes
 - summarize information from multiple organisms,
 - are usually haploid (exception discussed later),
 - present an idealized sequence (by consensus),
 - and built using De Novo Sequence Assembly approaches
 - SPAdes
 - Abyss



Abyss tutorial

- Install abyss software:
 - sudo apt-get install abyss

```
micah@MicahsPC:~/TWU/assembly$ sudo apt-get install abyss
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following NEW packages will be installed:
  abyss
0 upgraded, 1 newly installed, 0 to remove and 0 not upgraded.
Need to get 3247 kB of archives.
After this operation, 9831 kB of additional disk space will be used.
Get:1 http://archive.ubuntu.com/ubuntu jammy/multiverse amd64 abyss amd64 2.3.1-1 [3247 kB]
Fetched 3247 kB in 3s (1092 kB/s)
Selecting previously unselected package abyss.
(Reading database ... 66210 files and directories currently installed.)
Preparing to unpack .../abyss_2.3.1-1_amd64.deb ...
Unpacking abyss (2.3.1-1) ...
Setting up abyss (2.3.1-1) ...
Processing triggers for man-db (2.10.2-1) ...
```



Abyss tutorial

- Download Sars-Cov-2 genome:
 - https://www.ncbi.nlm.nih.gov/nuccore/NC_045512.2?report=fasta

National Library of Medicine
National Center for Biotechnology Information

Nucleotide Nucleotide Advanced

FASTA Send to: ▾

Severe acute respiratory syndrome coronavirus 2 isolate Wuhan-Hu-1, complete genome

NCBI Reference Sequence: NC_045512.2

[GenBank](#) [Graphics](#)

```
>NC_045512.2 Severe acute respiratory syndrome coronavirus 2 isolate Wuhan-Hu-1, complete genome
ATTAAGGTTATACTTCCAGGTAAACAAACCAACCTTGATCTTGTAGATCTGTTCTAA
CGAACCTTAAATCTGTGGCTGTCACTCGGCTGATGCTTAGTCAGCTACGGCAGTATAATTAAAC
TAATTACTGCTTGTGACAGGACACAGAGTAACTCGTCTATCTTCTGCAGGCTGCTTACGGTTCTGCGT
TTGCAAGCGATCATCAGCACATCTAGGTTCTGCCGGGTGTGACCGAAAGGTAAGATGGAGAGCCTTGT
```



Abyss Tutorial

National Library of Medicine
National Center for Biotechnology Information

Nucleotide Nucleotide Advanced

FASTA ▾

Severe acute respiratory syndrome coronavirus 2 isolate complete genome

NCBI Reference Sequence: NC_045512.2

[GenBank](#) [Graphics](#)

```
>NC_045512.2 Severe acute respiratory syndrome coronavirus 2 isolate Wuhan-Hu-1, genome
CTTAAAGGTTTACCTTCCCAGGTAACAAACCAACCAACTTCTGATCTTGTAGATCTGTTCTAAA
CGAACCTTAAATCTGTGCTGGCTGTCACTCGGCTGCATGCTTAGTCAGCTACAGCAGTATAATTATAAC
TAATTACTGCTGGACAGGACACGAGTAACCTGCTATCTCTGCAAGGCTGCAGGTTACGGTTCTGCCGTG
TTGCAAGCCGATCATCAGCACATCTAGGTTCTGCCGGGTGACGGAAAGGTAAGATGGAGAGCCCTTGTC
CCTGCTTCAACGAGAAAAACACAGCTCAACTCAGTTGCTGTGTTTACAGGTTGCGGACGTGCTCGTAC
GTGGCTTGGGAGACTCCGTGGAGGGGCTTATCAGAGGCACGTCAACATCTAAAGATGGCACTTGTGG
CTTAGTGAAGGTGAAAAAAGGCGTTTGCCTCAACTGAAACGCCCTATGTTCATCAAAGCTTCCGAT
GCTCGAACTGCACCTCATGGTCATGTTAGCTGGTAGCAGAACTCGAAGGCATTGATACGGTC
GTATGTTGAGACACTTGTGCTCTGTCCCTATGTTGGCAGAACATCCAGTGGCTTACCGCAAGGTTCT
TCTTCGTAAGAACGTTAATAAAGGAGCTGGTGGCCATAGTTACGGGCCGATCTAAAGTCATTGACTTA
GGCGACGAGCTTGGACTGTACCTTATGAAAGATTTCAAGAAAATGGAAACACTAAACATAGCAGTGGT
TTACCCGTGAACTCATGGTGAGCTAACGGGGGGCATACACTCGTATGTCGATAACAACTTGTGG
```

Send to:

Complete Record
 Coding Sequences
 Gene Features

Choose Destination

File Clipboard
 Collections Analysis Tool

Download 1 item.

Format: FASTA ▾

Show GI

Create File (highlighted)

Related inform
Assembly



TEXAS WOMAN'S
UNIVERSITY

Abyss tutorial

- Create working directory for assembly under home

```
micah@MicahsPC:~$ mkdir workshop_asm  
micah@MicahsPC:~$ cd workshop_asm  
micah@MicahsPC:~/workshop_asm$ |
```



Abyss tutorial

- Move the downloaded genome file to this directory:
 - cp /mnt/c/Users/windows_username/Downloads/sequence.fasta .
 - NOTE: Change windows_username to your windows username

```
micah@MicahsPC:~/workshop_asm$ cp /mnt/c/Users/Micah/Downloads/sequence.fasta .
micah@MicahsPC:~/workshop_asm$ |
```

- Verify file moved with 'ls'

```
micah@MicahsPC:~/workshop_asm$ ls
sequence.fasta
```



Abyss tutorial

- Install python for simulating reads from genome file (this may take a minute [73 Mb])
 - sudo apt-get install python3
- Install pip package manager for python
 - sudo apt-get install pip



Abyss tutorial

- Install Biopython for dependencies
 - sudo pip install biopython
- Retrieve python script
 - wget https://raw.githubusercontent.com/mathornton01/twupa_bioinformaticsws_2024/main/tools/sample.fasta.py



Abyss tutorial

- Sample the Coronavirus genome
 - python3 sample_fasta.py sequence.fasta
1000000 100 300 simsam
- Check simulated samples with
 - head simsam_R1.fasta
 - head simsam_R2.fasta



Abyss Tutorial

```
micah@MicahsPC:~/workshop_asm$ head simsam_R1.fasta
>read_1/1
GTAGTAGTACTTCTTTGAACTTACATGCACCAGCAACTGTTGTGGACCTAAAAGTCTACTAATTGGTAAAAACAAATGTGTCATT
TCAACT
>read_2/1
AATGGAGGTAAAGGCTTTGCAAACATACACAATTGGAATTGTGTTAATTGTGATACTTCTGTGCTGGTAGTACATTATTAGTGATGAAGTTG
CGAGAG
```

```
micah@MicahsPC:~/workshop_asm$ head simsam_R2.fasta
>read_1/2
AGCAACCTGGTTAGAAGTATTGTTCTGGTGTATAACACTGACACCACCAAAAGAACATGGTGTAAATGTCAGAACATCTCAAGTGTCTGTGGA
TCACGG
>read_2/2
AATAGGCAATGAACCTTAGTGTATTAGCTCTCAGGTTGTCAGTTAACAAATGAGAGAGAGAATGTCTTCATAAGTCTTGGACAGCT
TTATCA
```



Abyss Tutorial

- Run abyss-pe on simulated reads.
 - abyss-pe k=25 name=TWU
in="simsam_R1.fasta simsam_R2.fasta"

```
micah@MicahsPC:~/workshop_asm$ abyss-pe k=25 name=TWU in="simsam_R1.fasta simsam_R2.fasta"
ABYSS -k25 -q3 --coverage-hist=coverage.hist -s TWU-bubbles.fa -o TWU-1.fa simsam_R1.fasta
simsam_R2.fasta
ABYSS 2.3.1
ABYSS -k25 -q3 --coverage-hist=coverage.hist -s TWU-bubbles.fa -o TWU-1.fa simsam_R1.fasta sim
samt_R2.fasta
Reading 'simsam_R1.fasta'...
Reading 'simsam_R2.fasta'...
```



Abyss Tutorial

- Check output in file TWU-1.fa



Abyss Tutorial

>0 29694 151574475

```
TTCGTCCGTGTTGCAGCCGATCATCAGCACATCTAGGTTCGTCGGGTGTGACCGAAAGGTAAGATGGAGAGCCTGTCCTGGTTCAACGA  
GAAAACACACGTCCAACTCAGTTGCCTGTTTACAGGTTCGCAGCTGCTCGTACGTGGCTTGGAGACTCCGTGGAGGGCTTACAGAG  
GCACGTCAACATCTTAAAGATGGCACTTGTGGCTTAGTAGAAGTTGAAAAAGGCGTTGCCTCAACTGAACAGCCCTATGTGTTCATCAAAC  
GTTCGGATGCTCGAACCTCATGGTCATGTTATGGTTGAGCTGGTAGCAGAACTCGAAGGCATTCACTGAGCTGGTAGCAGTACGGTCGTAGTGGTGAGACACT  
TGGTGCCTTGTCCCTCATGTGGCGAAATACCAGTGGCTACCGCAAGGTTCTTCTCGTAAGAACGGAATAAAGGAGCTGGTGGCCATAGT  
TACGGCGCCGATCTAAAGTCATTGACTTAGGCAGCAGCTGGCACTGATCCTATGAAGATTTCAGAAAATGGAACACTAAACATAGCA
```

```
micah@MicahsPC:~/workshop_asm$ head sequence.fasta  
>NC_045512.2 Severe acute respiratory syndrome coronavirus 2 isolate Wuhan-Hu-1, complete genome  
ATTAAAGGTTTACCTCCCAGGTAAACAAACCAACCTTCGATCTCTGTAGATCTGTTCTCTAA  
CGAACCTTAAATCTGTGTGGCTGTCACTCGGCTGCATGCTTAGTGCACTCACGCAGTATAATTAAAC  
TAATTACTGTCGTTGACAGGACACGGAGTAACTCGTCTATCTTCTGCAGGCTGCTTACGGTTTCGTCCGTG  
TTGCAGCCGATCATCAGCACATCTAGGTTCGTCGGGTGTGACCGAAAGGTAAGATGGAGAGCCTGTC  
CCTGGTTCAACGAGAAAACACACGTCCAACTCAGTTGCCTGTTACAGGTTCGCGACGTGCTCGTAC  
GTGGCTTGGAGACTCCGTGGAGGGAGGTCTTATCAGAGGCACGTCAACATCTAAAGATGGCACTTGTGG  
CTTAGTAGAAGTTGAAAAAGGCGTTGCCTCAACTGAAACAGCCCTATGTGTTCATCAAACGTTGGAT  
GCTCGAACTGCACCTCATGGTCATGTTATGGTGAGCTGGTAGCAGAACTCGAAGGCATTCACTGAGCTGGTC  
GTAGTGGTGAGACACTTGGTGTCCCTCATGTGGCGAAATACCAGTGGCTACCGCAAGGTTCT
```



Human Reference Genome

- The human reference genome (GRCh38) is a complete assembly of the human genome
- It is a haploid representation of a diploid sequence.
- It is comprised of the DNA from eight people
 - One individual accounts for 66 % of the DNA.
- Contains the O allele for ABO blood group (but others are annotated).
- Can be accessed and downloaded here:
https://www.ncbi.nlm.nih.gov/datasets/genome/GCF_000001405.26/
- Approximately 3.2 GB in size.



By Russ London at English Wikipedia, CC BY-SA 3.0,
<https://commons.wikimedia.org/w/index.php?curid=9923576>



Standard FASTA Format

- FASTA stands for Fast-All
 - All refers to any possible sequence of Latin or Alphanumeric/Special characters
- Utilizes standard “alphabets” for protein sequences or genetic sequences



FASTA Format

- Starts label lines with “>” followed by the label of the subsequent sequence.

```
micah@MicahsPC:~/TWU$ cat example_fasta_1.fa
>seq_1
ACTTAATATCGTATACTAatCgaGTAACg
```

- Basis for the fasta-quality files we will discuss next



FASTA Format - Genes

- Nucleic Acid Codes contain more than just ACGTU
 - Can be upper or lower case – meaning differs
 - Lower case – can indicate low-confidence calls or masked bases

Nucleic Acid Code *	Meaning	Mnemonic *
A	A	Adenine
C	C	Cytosine
G	G	Guanine
T	T	Thymine
U	U	Uracil
(i)	i	inosine (non-standard)
R	A or G (l)	puRine
Y	C, T or U	pYrimidines
K	G, T or U	bases which are Ketones
M	A or C	bases with aMino groups
S	C or G	Strong interaction
W	A, T or U	Weak interaction
B	not A (i.e. C, G, T or U)	B comes after A
D	not C (i.e. A, G, T or U)	D comes after C
H	not G (i.e., A, C, T or U)	H comes after G
V	neither T nor U (i.e. A, C or G)	V comes after U
N	A C G T U	Nucleic acid
-	gap of indeterminate length	



FASTA Format – Proteins

Amino Acid Code	Meaning
A	Alanine
B	Aspartic acid (D) or Asparagine (N)
C	Cysteine
D	Aspartic acid
E	Glutamic acid
F	Phenylalanine
G	Glycine
H	Histidine
I	Isoleucine
J	Leucine (L) or Isoleucine (I)
K	Lysine
L	Leucine
M	Methionine/Start codon

N	Asparagine
O	Pyrrolysine (rare)
P	Proline
Q	Glutamine
R	Arginine
S	Serine
T	Threonine
U	Selenocysteine (rare)
V	Valine
W	Tryptophan
Y	Tyrosine
Z	Glutamic acid (E) or Glutamine (Q)
X	any
*	translation stop
-	gap of indeterminate length



FASTQ Format - Quality

- Instead of 2 lines per sequence has 4 lines
 - Line 1 starts with @ and has Sequence ID
 - Line 2 Contains the sequence
 - Line 3 is + just as a separator
 - Line 4 contains quality information stored as ASCII character where 0x21 (!) is lowest quality and 0x7e (~) is highest



FASTQ quality (Phred Scores)

- Quality values are integer mappings (hexadecimal) of
 - $p_k = \Pr(\text{base } k \text{ is incorrect})$
- Phred Score (probability formulation)
 - $Q_k = -10 \log_{10}(p_k)$
- Earlier Illumina Score (odds formulation)
 - $Q_k = -10 \log_{10} \left(\frac{p_k}{1-p_k} \right)$



Break-out Session 1 – Quality Scores

- Using the probability formulation, what is the probability of an incorrect base at a PHRED score of ‘;’?
- Using the odds formulation?

55	067	37	00110111	7
56	070	38	00111000	8
57	071	39	00111001	9
58	072	3A	00111010	:
59	073	3B	00111011	:
60	074	3C	00111100	<
61	075	3D	00111101	=
62	076	3E	00111110	>
63	077	3F	00111111	?
64	100	40	01000000	@
65	101	41	01000001	A
66	102	42	01000010	B
67	103	43	01000011	C



Quality Scores

$$Q_k = -10 \log_{10}(p_k)$$

$$\frac{Q_k}{-10} = \log_{10}(p_k)$$

$$10^{\frac{Q_k}{-10}} = p_k$$

$$\text{PHRED} = “;” \Rightarrow Q_k = 59 - 33 = 26 \Rightarrow p_k = 10^{-2.6}$$
$$p_k \approx 0.002512$$



Quality Scores

$$Q_k = -10 \log_{10} \left(\frac{p_k}{1 - p_k} \right)$$

$$\frac{Q_k}{-10} = \log_{10} \left(\frac{p_k}{1 - p_k} \right)$$

$$10^{\frac{Q_k}{-10}} = \frac{p_k}{1 - p_k}$$



Quality Scores

$$10^{\frac{Q_k}{-10}}(1 - p_k) = p_k$$

$$10^{\frac{Q_k}{-10}} = \left(1 + 10^{\frac{Q_k}{-10}}\right)p_k$$

$$p_k = \frac{10^{\frac{Q_k}{-10}}}{1 + 10^{\frac{Q_k}{-10}}}$$



Quality Scores

$$\text{PHRED} = \text{";"} \Rightarrow p_k = \frac{10^{-2.6}}{1+10^{-2.6}} \approx 0.0025056$$



Quality Scores

- Probability formulation at “!”
- Odds formulation?

DEC	OCT	HEX	BIN	Symbol
32	040	20	00100000	SP
33	041	21	00100001	!
34	042	22	00100010	"
35	043	23	00100011	#
36	044	24	00100100	\$
37	045	25	00100101	%
38	046	26	00100110	&
39	047	27	00100111	,
40	050	28	00101000	(
41	051	29	00101001)
42	052	2A	00101010	*



Quality Scores

$$Q_k = -10 \log_{10}(p_k)$$
$$\Rightarrow p_k = 10^{\frac{Q_k}{-10}}$$

$$\text{PHRED} = “!” \Rightarrow Q_k = 33 - 33 = 0 \Rightarrow p_k = 10^0 = 1$$



Quality Scores

$$Q_k = -10 \log_{10} \left(\frac{p_k}{1 - p_k} \right)$$
$$\Rightarrow p_k = \frac{10^{\frac{Q_k}{-10}}}{1 + 10^{\frac{Q_k}{-10}}}$$
$$\Rightarrow p_k = \frac{10^0}{1 + 10^0} = \frac{1}{2} = 0.5$$

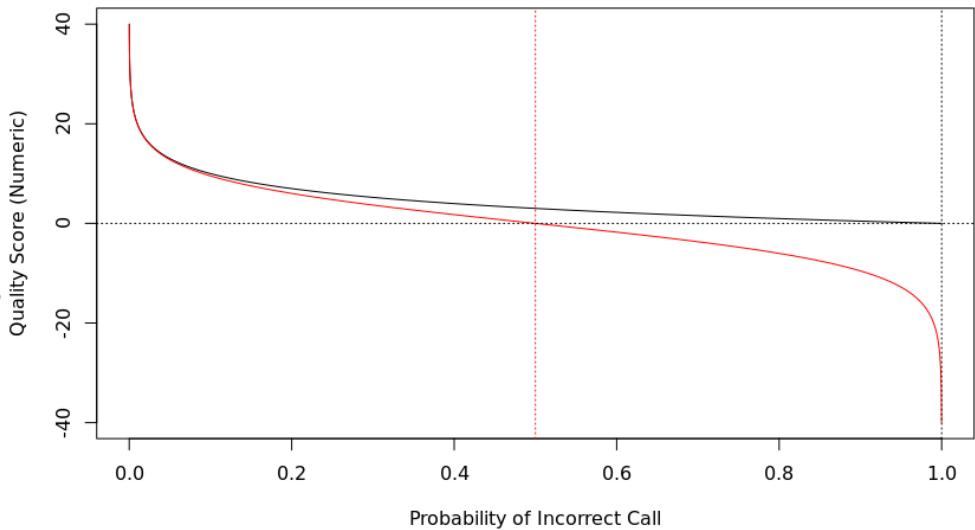


TEXAS WOMAN'S
UNIVERSITY

Quality Scores

```
1 Q_probability <- Vectorize(function(p){  
2   return(-10*log10(p))  
3 })  
4  
5 Q_odds <- Vectorize(function(p){  
6   return(-10*log10(p/(1-p)))  
7 })  
8  
9 p <- seq(0,1,0.0001)  
10  
11 plot(p,  
12   Q_probability(p),  
13   type='l',  
14   ylim=c(-40,40),  
15   main = "Comparison of Probability and Odds Formulation \n for PHRED Quality Scores",  
16   ylab = "Quality Score (Numeric)",  
17   xlab="Probability of Incorrect Call")  
18 lines(p,Q_odds(p),type='l',col='red')  
19 abline(h=0,lty=3)  
20 abline(v=0.5,lty=3,col='red')  
21 abline(v=1,lty=3,col='black')  
22 legend(0.6,40,legend=c("Probability","Odds"),lty=1,col=c('black','red'))
```

Comparison of Probability and Odds Formulation for PHRED Quality Scores



Break (10 min)



TEXAS WOMAN'S
UNIVERSITY

Sequence Alignment Maps

- SAM Files are Sequence Alignment Maps
- They are Tab-separated Files
- The fields of SAM files are well defined.
- SAM files are the output of alignment steps
- SAM files tell you where each read in a read file maps to a genome from a genome file



Sequence Alignment Maps

- Each field contains vital information about the alignment of a given read from a read file

Col	Field	Type	Brief description
1	QNAME	String	Query template NAME
2	FLAG	Int	bitwise FLAG
3	RNAME	String	References sequence NAME
4	POS	Int	1- based leftmost mapping POSition
5	MAPQ	Int	MAPping Quality
6	CIGAR	String	CIGAR string
7	RNEXT	String	Ref. name of the mate/next read
8	PNEXT	Int	Position of the mate/next read
9	TLEN	Int	observed Template LENgth
10	SEQ	String	segment SEQuence
11	QUAL	String	ASCII of Phred-scaled base QUALity+33



Sequence Alignment Maps - Flags

- FLAGS are field 2, displayed as integer

Integer	Binary	Description (Paired Read Interpretation)
1	000000000001	template having multiple templates in sequencing (read is paired)
2	000000000010	each segment properly aligned according to the aligner (read mapped in proper pair)
4	000000000100	segment unmapped (read1 unmapped)
8	00000001000	next segment in the template unmapped (read2 unmapped)
16	00000010000	SEQ being reverse complemented (read1 reverse complemented)
32	00000100000	SEQ of the next segment in the template being reverse complemented (read2 reverse complemented)

64	000001000000	the first segment in the template (is read1)
128	000010000000	the last segment in the template (is read2)
256	000100000000	not primary alignment
512	001000000000	alignment fails quality checks
1024	010000000000	PCR or optical duplicate
2048	100000000000	supplementary alignment (e.g. aligner specific, could be a portion of a split read or a tied region)



Sequence Alignment Maps - FLAGS

- Suppose 2145 is provided as the flag on a line.
- Convert 2145 from Base 10 to Base 2 (Binary)



Sequence Alignment Maps - FLAGS

- Recall: Conversion of base 10 to base 2 requires repeated division by 2.



Sequence Alignment Maps - FLAGS

$$(2145)_{10} \Rightarrow (?)_2$$

$$2145 \div 2 = 1072 R 1$$

$$1072 \div 2 = 536 R 0$$

$$536 \div 2 = 268 R 0$$

$$268 \div 2 = 134 R 0$$

$$134 \div 2 = 67 R 0$$

$$67 \div 2 = 33 R 1$$

$$33 \div 2 = 16 R 1$$

$$16 \div 2 = 8 R 0$$

$$8 \div 2 = 4 R 0$$

$$4 \div 2 = 2 R 0$$

$$2 \div 2 = 1 R 0$$

$$1 \div 2 = 0 R 1$$



Sequence Alignment Maps - FLAGS

$$(2145)_{10} \Rightarrow (100001100001)_2$$

Integer	Binary	Description (Paired Read Interpretation)
1	000000000001	template having multiple templates in sequencing (read is paired)
2	000000000010	each segment properly aligned according to the aligner (read mapped in proper pair)
4	00000000100	segment unmapped (read1 unmapped)
8	00000001000	next segment in the template unmapped (read2 unmapped)
16	00000010000	SEQ being reverse complemented (read1 reverse complemented)
32	00000100000	SEQ of the next segment in the template being reverse complemented (read2 reverse complemented)

64	000001000000	the first segment in the template (is read1)
128	000010000000	the last segment in the template (is read2)
256	000100000000	not primary alignment
512	001000000000	alignment fails quality checks
1024	010000000000	PCR or optical duplicate
2048	100000000000	supplementary alignment (e.g. aligner specific, could be a portion of a split read or a tied region)



Sequence Alignment Map - FLAGS

$$(2145)_{10} \Rightarrow (100001100001)_2$$

Flag Value	Meaning
1	read is paired
32	read2 was reverse complemented
64	read1
2048	Supplementary alignment



FLAGS

- Please describe a read that has FLAG
 $(1683)_{10}$

Integer	Binary	Description (Paired Read Interpretation)
1	000000000001	template having multiple templates in sequencing (read is paired)
2	000000000010	each segment properly aligned according to the aligner (read mapped in proper pair)
4	00000000100	segment unmapped (read1 unmapped)
8	00000001000	next segment in the template unmapped (read2 unmapped)
16	00000010000	SEQ being reverse complemented (read1 reverse complemented)
32	00000100000	SEQ of the next segment in the template being reverse complemented (read2 reverse complemented)

64	000001000000	the first segment in the template (is read1)
128	000010000000	the last segment in the template (is read2)
256	000100000000	not primary alignment
512	001000000000	alignment fails quality checks
1024	010000000000	PCR or optical duplicate
2048	100000000000	supplementary alignment (e.g. aligner specific, could be a portion of a split read or a tied region)



FLAGS

$$(1683)_{10} \Rightarrow (011010010011)_2$$

Integer	Binary	Description (Paired Read Interpretation)
1	00000000001	template having multiple templates in sequencing (read is paired)
2	00000000010	each segment properly aligned according to the aligner (read mapped in proper pair)
4	00000000100	segment unmapped (read1 unmapped)
8	00000001000	next segment in the template unmapped (read2 unmapped)
16	00000010000	SEQ being reverse complemented (read1 reverse complemented)
32	00000100000	SEQ of the next segment in the template being reverse complemented (read2 reverse complemented)

64	000001000000	the first segment in the template (is read1)
128	000010000000	the last segment in the template (is read2)
256	000100000000	not primary alignment
512	001000000000	alignment fails quality checks
1024	010000000000	PCR or optical duplicate
2048	100000000000	supplementary alignment (e.g. aligner specific, could be a portion of a split read or a tied region)



Sequence Alignment Maps - MAPQ

- Map Quality is very similar to base call quality,
- $M(p) = -10 \log_{10} p$
- $p = \Pr(\text{Mapped position is wrong})$

Col	Field	Type	Brief description
1	QNAME	String	Query template NAME
2	FLAG	Int	bitwise FLAG
3	RNAME	String	References sequence NAME
4	POS	Int	1- based leftmost mapping POSition
5	MAPQ	Int	MAPping Quality
6	CIGAR	String	CIGAR string
7	RNEXT	String	Ref. name of the mate/next read
8	PNEXT	Int	Position of the mate/next read
9	TLEN	Int	observed Template LENgth
10	SEQ	String	segment SEQuence
11	QUAL	String	ASCII of Phred-scaled base QUALity+33



Sequence Alignment Maps - CIGAR

- CIGAR
Describes how
the sequence
maps to the
given location

Col	Field	Type	Brief description
1	QNAME	String	Query template NAME
2	FLAG	Int	bitwise FLAG
3	RNAME	String	References sequence NAME
4	POS	Int	1- based leftmost mapping POSition
5	MAPQ	Int	MAPping Quality
6	CIGAR	String	CIGAR string
7	RNEXT	String	Ref. name of the mate/next read
8	PNEXT	Int	Position of the mate/next read
9	TLEN	Int	observed Template LENgth
10	SEQ	String	segment SEQuence
11	QUAL	String	ASCII of Phred-scaled base QUALity+33

Concise Idiosyncratic Gapped Alignment Report



TEXAS WOMAN'S
UNIVERSITY

Sequence Alignment Maps - CIGAR

CIGAR Format [\[edit\]](#)

Ref. : GTCGTAGAATA

Read: CACGTAG—TA

CIGAR: 2S5M2D2M where:

2S = 2 soft clipping (could be mismatches, or a read longer than the matched sequence)

5M = 5 matches or mismatches

2D = 2 deletions

2M = 2 matches or mismatches

- Soft clipping retains unaligned portion of read from SAM
- Hard clipping removes unaligned portion of read from SAM

CIGAR Code	BAM Integer	Description
M	0	alignment match (can be a sequence match or mismatch)
I	1	insertion to the reference
D	2	deletion from the reference
N	3	skipped region from the reference
S	4	soft clipping (clipped sequences present in SEQ)
H	5	hard clipping (clipped sequences NOT present in SEQ)
P	6	padding (silent deletion from padded reference)
=	7	sequence match
X	8	sequence mismatch



CIGAR

- Suppose we have the following rather poor alignment, please write down the CIGAR string you would find in the SAM file.

REF : AGCATTACGATACTA
READ: gggATTA---TAggg



CIGAR

REF: AGCATTACGATACTA
READ: gggATTA---TAggg

- 3S4M3D2M3S
 - 3S – three bases softclipped at beginning of read
 - 4M – four matches
 - 3D – 3 Deletions
 - 2M – two matches
 - 3S – three bases softclipped at end of read



Sequence Alignment Maps - BAM

- The sequence alignment map can be binarized to save room (that is converted into a completely binary file).



BAM Specification

Field	Description	Type	Value
magic	BAM magic string	char[4]	BAM\1
l_text	Length of the header text, including any NUL padding	uint32_t	< 2 ³¹
text	Plain header text in SAM; not necessarily NUL-terminated	char[l_text]	
n_ref	# reference sequences	uint32_t	< 2 ³¹
<i>List of reference information (n=n_ref)</i>			
l_name	Length of the reference name plus 1 (including NUL)	uint32_t	limited
name	Reference sequence name; NUL-terminated	char[l_name]	
l_ref	Length of the reference sequence	uint32_t	< 2 ³¹
<i>List of alignments (until the end of the file)</i>			
block_size	Total length of the alignment record, excluding this field	uint32_t	limited
refID	Reference sequence ID, -1 ≤ refID < n.ref; -1 for a read without a mapping position	int32_t	[-1]
pos	0-based leftmost coordinate (= POS - 1)	int32_t	[-1]
l_read_name	Length of read_name below (= length(QNAME) + 1)	uint8_t	
mapq	Mapping quality (=MAPQ)	uint8_t	
bin	BAI index bin, see Section 4.2.1	uint16_t	
n_cigar_op	Number of operations in CIGAR, see Section 4.2.2	uint16_t	
flag	Bitwise flags (= FLAG) ²⁹	uint16_t	
l_seq	Length of SEQ	uint32_t	limited
next_refID	Ref-ID of the next segment (-1 ≤ next_refID < n.ref)	int32_t	[-1]
next_pos	0-based leftmost pos of the next segment (= PNEXT - 1)	int32_t	[-1]
tlen	Template length (= TLEN)	int32_t	[0]
read_name	Read name, NUL-terminated (QNAME with trailing '\0') ³⁰	char[l_read_name]	
cigar	CIGAR: op_len<<4 op. 'MIDNSHP=X' → '012345678'	uint32_t[n_cigar_op]	
seq	4-bit encoded read: '='ACMGRSVTWYHKDBN' → [0,15]. See Section 4.2.3	uint8_t[(l_seq+1)/2]	
qual	Phred-scaled base qualities. See Section 4.2.3	char[l_seq]	
<i>List of auxiliary data (until the end of the alignment block)</i>			
tag	Two-character tag	char[2]	
val_type	Value type: AcCsSiIfZHB, see Section 4.2.4	char	
value	Tag value	(by val_type)	



TEXAS WOMAN'S
UNIVERSITY

BAMFile Index

0 (0–144 kbp)								
1 (0–48 kbp)			2 (48–96 kbp)			3 (96–144 kbp)		
4 (0–16k)	5 (16–32k)	6 (32–48k)	7 (48–64k)	8 (64–80k)	9 (80–96k)	10	11	12

An alignment starting at 65 kbp and ending at 67 kbp would have a bin number 8, which is the smallest bin containing the alignment. Similarly, an alignment starting at 51 kbp and ending at 70 kbp would go to bin 2, while an alignment between [40k, 49k] to bin 0. Suppose we want to find all the alignments overlapping

Questions:

- 1) What is the bin number for an alignment spanning [90k,99k]
- 2) [30k,39k]?
- 3) [0,9k]?



Sequence Alignment Maps - BAI

5.2 The BAI index format for BAM files

Field	Description	Type	Value
magic	Magic string	char[4]	BAI\1
n_ref	# reference sequences	uint32_t	< 2 ³¹
<i>List of indices (n=n_ref)</i>			
n_bin	# distinct bins (for the binning index)	uint32_t	≤ 37451
<i>List of distinct bins (n=n_bin)</i>			
bin	Distinct bin	uint32_t	≤ 37450
n_chunk	# chunks	uint32_t	limited ³⁵
<i>List of chunks (n=n_chunk)</i>			
chunk_beg	(Virtual) file offset of the start of the chunk	uint64_t	
chunk_end	(Virtual) file offset of the end of the chunk	uint64_t	
n_intv	# 16kbp intervals (for the linear index)	uint32_t	≤ 2 ¹⁷
<i>List of intervals (n=n_intv)</i>			
ioffset	(Virtual) file offset of the first alignment in the interval	uint64_t	
n_no_coor (optional)	Number of unplaced unmapped reads (RNAME *)	uint64_t	

From SAM Format Specification:

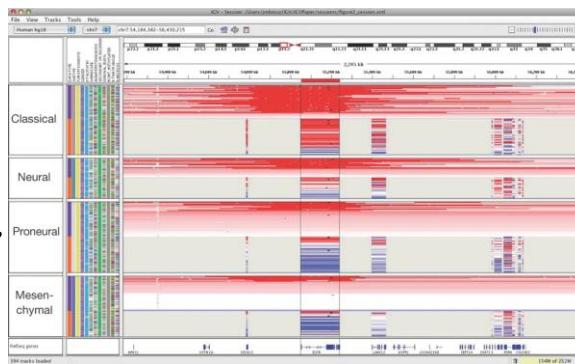
<https://samtools.github.io/hts-specs/SAMv1.pdf>



TEXAS WOMAN'S
UNIVERSITY

The Broad Institute Integrative Genomics Viewer

- Developed and Maintained by the **Broad Institute**:
 - A **Joint Institute of Harvard and MIT**.
 - Founded in 2004 (from organizational remnants of the Human Genome Project) for the purposes of [from <https://www.broadinstitute.org/about-us>]:
 - “Illuminating Human Disease”
 - “Reading and Editing Genomes”
 - “Sharing Data and Tools”
 - “Building Communities”
 - “Developing Diagnostics and Treatments”
 - “Collaborating, Innovating, and Empowering”
- **Integrative Genomics Viewer (IGV)** is a software environment for analyzing NGS data
 - Provided and maintained by The Broad Institute, with IGV team based at UC San Diego, and MIT/Harvard Broad Institute.
 - Supported by funding from:
 - The National Cancer Institute (NCI)
 - The National Institutes of Health (NIH)
 - Informatics Technology for Cancer Research (ITCR)
 - Starr Cancer Consortium
 - Written in Java, hence allowing use of platform independent jvm.



From Robinson, J., Thorvaldsdóttir, H., Winckler, W. et al. Integrative genomics viewer. *Nat Biotechnol* **29**, 24–26 (2011). <https://doi.org/10.1038/nbt.1754>



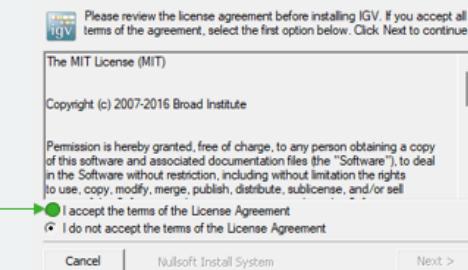
Locally Installing the IGV

- IGV is freely available for download and use, and since it was written using java, versions exist for all major operating systems (on which the JVM can run):*



The screenshot shows the "Downloads" section of the IGV website. It features a sidebar with links like Home, Downloads, Documents, IGV User Guide, File Formats, Tutorial Videos, Hosted Genomes, FAQ, Release Notes, Credits, and Contact. Below the sidebar is a search bar. The main content area displays download links for different platforms:

- IGV Mac App Java included** (green button)
- IGV MacOS App Separate Java 11 required** (yellow button)
- IGV for Windows Java included** (purple button)
- IGV for Windows Separate Java 11 required** (yellow button)
- IGV for Linux Java included** (blue button)
- Command line IGV and igyttools for all platforms Separate Java 11 required** (yellow button)

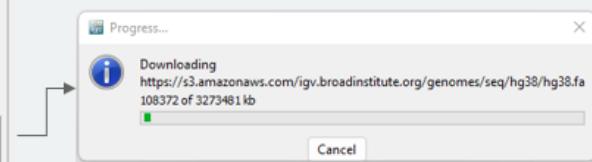
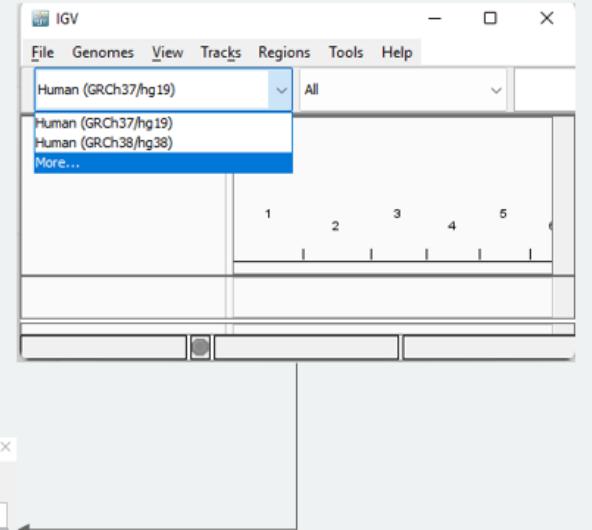
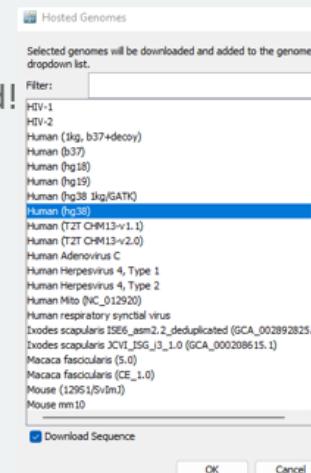
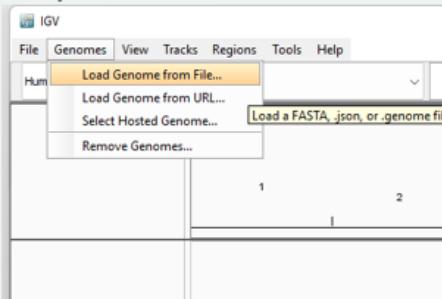


- Note:** Must agree to The MIT License to install and use IGV.

- Very non-restrictive license.
- May do many things, as long as copy of original license included.
- More info: <https://choosealicense.com/licenses/mit/>

Installing new reference genomes in IGV

- Recall that reference genomes are very large files which contain FASTA format versions of the consensus of many samples of the same organism.
 - The most commonly used version of the human reference genome is called GRCh38, which was the co-ordinated result (by the Genome Reference Consortium) of many different studies including
 - 1000 Genomes Project
- GRCh38 was released in December of 2013.
 - Most recent *version* February 2022
 - Gapless except chromosome Y.*
- Many different references available for download!
- Can also install directly from downloaded or Manually assembled FASTA reference file.



Loading a Sequence Alignment Map File (aligned sequencing reads file)

Two Result Files necessary for display:

1. Binary Alignment Map (.bam)
2. Binary Alignment Index (.bai)

Example Queries and Reference (input reads to aligner)

```

Coor 12345678901234 5678901234567890123456789012345
ref AGCATGTTAGATAA**GATAGCTGCTAGTAGGCAGTCAGGGCCAT

+r001/1      TTAGATAAAGCATA*CTG
+r002          aaaAGATAA*GGATA
+r003          gcctaAGCTAA
+r004          ATAGCT.....TCAGC
-r003          tttagctTAGGC
-r001/2          CAGCGGGCAT

```

5.2 The BAI index format for BAM files

Field	Description	Type	Value
magic	Magic string	char[4]	BAM\1
n_ref	# reference sequences	uint32_t	< 2 ³¹
list	List of indices (n_i, n_j)	uint32_t	
n_bin	# distinct bins for the binning index	uint32_t	< 2 ³¹
bin	Distinct bin	uint32_t	< 2 ³¹
n_chunk	# chunks	uint32_t	< 2 ³¹
chunk	(Virtual) file offset of the start of the chunk	uint64_t	
chunk_end	(Virtual) file offset of the end of the chunk	uint64_t	
n_mapped	# likely intervals (for the linear index)	uint32_t	< 2 ³¹
offset	(Virtual) file offset of the first alignment in the interval	uint64_t	
n_no_coor	(optional) Number of unaligned mapped reads (RNAME = *)	uint64_t	

The index file may optionally contain additional metadata providing a summary of the number of mapped and unmapped read segments per reference sequence, and of any unaligned unmapped read segments.⁸ This is stored in an optional extra metadata parameter for each reference sequence, and in the optional trailing n_no_coor field at the end of the file.

The pseudo-bins appear in the references lists of distinct bins as bin number 21705 (which is beyond the normal range) and are laid out so as to be compatible with real bins and their chunks:

bin	Magic bin number	uint32_t	21705
n_chunk	# chunks	uint32_t	2
ref	Reference ID	uint32_t	1
ref_end	(Virtual) file offset of the start of reads placed on this reference	uint64_t	
ref_start	(Virtual) file offset of the end of reads placed on this reference	uint64_t	
n_mapped	Number of mapped read segments for this reference	uint64_t	
n_unmapped	Number of unmapped read segments for this reference	uint64_t	

BAI – Binary Alignment Index

Alignment tool:

- HISAT2
- Bowtie
- BWT
- STAR

Example SAM output (input to igv)

```

@HD VN:1.6 SO:coordinate
@SQ SN:ref LN:45
r001 99 ref 7 30 8M214M1D3M = 37 39 TTAGATAAAGGATACTG *
r002 0 ref 9 30 3S6M1P1I4M * 0 0 AAAAGATAAGGATA *
r003 0 ref 9 30 5S6M * 0 0 GCTTAAGCTAA * SA:Z:ref,29,-,6H5M,17,0;
r004 0 ref 16 30 6M14NSM * 0 0 ATAGCTTTCAGC *
r003 2064 ref 29 17 6H5M * 0 0 TAGGC * SA:Z:ref,9,+,5S6M,30,1;
r001 147 ref 37 30 9M = 7 -39 CAGCGGGCAT * NM:i:1

```

samtools view -bs file.sam > file.bam

Fields of SAM

Col	Field	Type	Regexp/Range	Brief description
1	QNAME	String	[1-7A-]{1,254}	Query template NAME
2	FLAG	Int	[0,2 ¹⁶ -1]	bitwise FLAG
3	RNAME	String	V* [rname:^=]* [rname:]*	Reference sequence NAME ¹¹
4	POS	Int	[0,2 ³¹ -1]	1-based leftmost mapping POSition
5	MAPQ	Int	[0,2 ⁸ -1]	MAPping Quality
6	CIGAR	String	V* (D-9)+(MIDNOSHDX-1)*	CIGAR string
7	RNEXT	String	V* 1 [rname:^=]* [rname:]*	Reference name of the mate/next read
8	PNEXT	Int	[0,2 ³¹ -1]	Position of the mate/next read
9	TLEN	Int	[-2 ³¹ + 1, 2 ³¹ - 1]	observed Template LENgth
10	SEQ	String	V* [A-Za-z-]*	segment SEQuence
11	QUAL	String	[!-]*	ASCII of Phred-based base QUALity+33

BAM is compressed in the BGZF format. All multi-byte numbers in BAM are little-endian, regardless of the machine endianness. The format is formally described in the following table where values in brackets are used when the corresponding information is not available; an underlined word in uppercase denotes a field in the SAM format.

Field	Description	Type	Value
magic	BAM magic string	char[4]	F8B8\1
list	Length of the header text, including any BME padding	uint32_t	< 2 ³¹
test	Plain header text in SAM; not necessarily BME-terminated	char [list]	
n_ref	# reference sequences	uint32_t	< 2 ³¹
name	Length of the reference name plus 1 (including BME)	uint32_t	limited
ref	Reference sequence name; BME-terminated	char [name]	
len	Length of the reference sequence	uint32_t	< 2 ³¹
list	Length of the reference sequence list (with the end of the list)	uint32_t	limited
block_size	Total length of the alignment record, excluding this field	uint32_t	limited
refID	Reference sequence ID, -1 ≤ refID < n_ref - 1 for a read	int32_t	[-1, n_ref - 1]
pos	0 based position indicator (-100-1)	int32_t	[-1]
tread	Length of read name below (= length(RNAME) + 1)	uint32_t	
mapq	Mapping quality (= MAPQ)	uint8_t	
is_mate	Indicates if this is a mate (Section 4.2.1)	uint8_t	
n_cigar	Number of operations in CIGAR, see Section 4.2.2	uint32_t	
flag	Bitwise flags (= FLAG) ¹⁰	uint16_t	
len	Length of SEQ	uint32_t	limited
next_refID	Reference sequence ID of the next segment (-1 ≤ next_refID < n_ref)	int32_t	[-1, n_ref - 1]
next_pos	0-based leftmost pos of the next segment (= PNEXT - 1)	int32_t	[-1]
then	Template length (= TLEN)	int32_t	0
read_name	Read name (including BME, trailing '0')	char [tread]	
cigar	CIGAR operation type (0, WILDCARD, MATCH/MISMATCH, DELETION, Insertion, N, equal, not equal)	uint8_t	
seq	4-bit encoded read (= ACGRDTYVHBR + 0, 15). See Section 4.2.3	uint8_t	
qual	Phred-scaled base qualities. See Section 4.2.3	char [tread]	
tag	Two-character tag	char [2]	
val_type	Value type: AcCdEfFfGfHf, see Section 4.2.4	char	
value	Tag value	[by val_type]	

Format Specification Images all defined in SAM format document, images from document available below, with more detail:

[\(SAM Specification\)](https://samtools.github.io/hts-specs/SAMv1.pdf)



TEXAS WOMAN'S
UNIVERSITY

BAM – Binary Encoding of SAM

Finding and downloading Read Files for your analyses using SRA

One way: Get sra-toolkit for linux! (or windows subsystem for linux)

Note: Add to .bashrc for persistent use

1. Go to the sequencing reads archive (NCBI SRA)

1. <https://www.ncbi.nlm.nih.gov/sra/>

2. Create your search for the terms you are interested in investigating (ie. "CMML")

The screenshot shows the NCBI SRA search interface. A green circle with the number 1 is overlaid on the top left. The search bar contains 'SRA' and 'CMML'. Below the search bar, a message says 'SRA - Now available on the cloud'. On the left, there's a sidebar with filters activated for 'DNA, exome' and a 'Clear all' button. The main search results area is mostly obscured by a large red rectangular box.

3. Select a result from those displayed (Bonus, use filters to subset the results displayed)

```
wget http://ftp-trace.ncbi.nlm.nih.gov/sra/sdk/2.4.1/srato toolkit.2.4.1-ubuntu64.tar.gz  
tar xzvf srato toolkit.2.4.1-ubuntu64.tar.gz  
export PATH=$PATH:/directory/srato toolkit.2.4.1-ubuntu64/bin
```

Use 'prefetch' command to speed up fastq-dump
To download only a specific number of reads uses the -X flag with fastq-dump (note prefetch fetches the entire file by default, the -X flag for prefetch does something different.)

https://www.reneshbedre.com/blog/ncbi_sra_toolkit.html
(SRA Toolkit Download Full Tutorial)

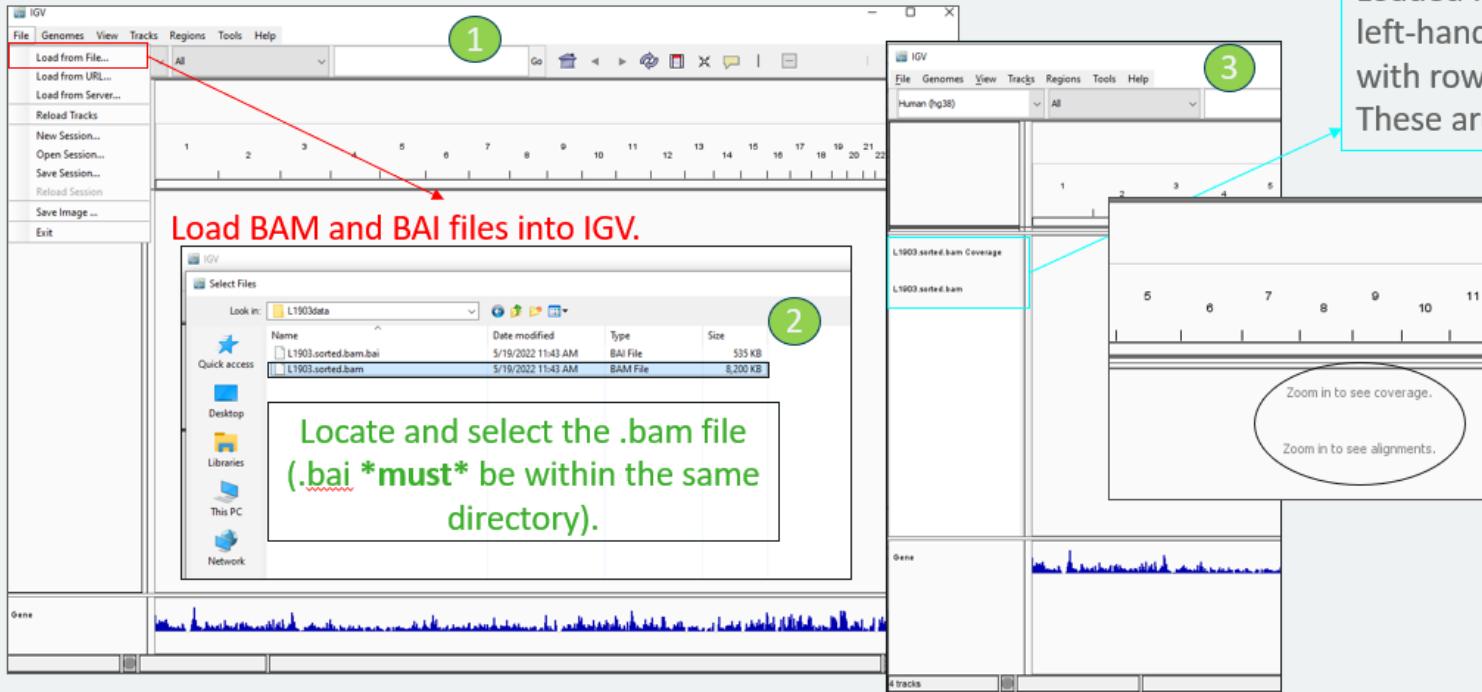
The screenshot shows the NCBI SRA search results for 'CMML'. A green circle with the number 2 is overlaid on the top left. The results list several experiments. One experiment is highlighted with a green circle containing the number 3. The details for this experiment are shown in a larger box:
Experiment ID: ERR3299798
Title: Whole exome sequencing of CD14+ CMML patient cells or of iPSC cells derived from CMML patient cells
Created by: UMR1170, Inserm & Gustave Roussy UMR8590, CNRS & University Paris 1 (UMR1170, Inserm & Gustave Roussy UMR8590, CNRS)
Library:
- Library: Patient A (CD14)
Instrument: Illumina Novaseq 6000
Strategy: WGS
Source: Human
Selection: None
Layout: PAIRED
Construction: Peripheral blood was collected on ethylenediaminetetraacetic acid (EDTA) patients with a CMML diagnosis according to the World Health Organization criteria. Peripheral blood mononuclear cells were separated on Ficoll-Hypaque. Peripheral blood CD14+ monocytes were sorted with magnetic beads and the AutoFacs system (Miltenyi Biotech, Bergisch Gladbach, Germany). Control samples were peripheral blood CD3 positive T lymphocytes sorted with the AutoFacs system. iPSC-derived cells were generated using standard protocol from CD34+ cells using the iCell reagent kit (CellStem Cell, Cambridge, MA, USA). iPSCs were differentiated into hematopoietic progenitors using the Hematopoietic cMyc. Two days later, cells were plated on murine embryonic fibroblast (MEF) inactivated by gamma irradiation (BioSystems, Lille, France), then transferred on day 7 in iPSC medium containing DMEM/F12 GlutaMAX supplement (ThermoFisher Scientific). Medium was replaced every other day for 3-4 weeks before manually picking single colonies (IPSC morphology and expanding). DNA was extracted from each sample using the QIAamp DNA mini kit (QIAGEN, Valencia, CA, USA). Genomic DNA was sheared using the Covaris S2 system (LGC Genomics). DNA fragments were end-repaired, extended with an 'A' base on the 3'-end, ligated with paired-end adaptors and amplified (10 cycles) using a Dnae automated platform (Agilent technologies). Exome-containing adaptor-ligated libraries were hybridized for 24 h using biotinylated probes and enriched with streptavidin-conjugated magnetic beads using SureSelect (Agilent technologies). The final libraries were indexed by PCR.

This will download a fastq file,

Caution, fastq files can be quite large!

Loading a Sequence Alignment Map File (aligned sequencing reads file)

1. In this section we will look at some human sequencing reads that have been aligned to the human reference genome (GRCh38) using the HISAT2 software aligner.
2. The data has been converted into the binary format (BAM) using samtools, and an index has been properly created using the same (BAI).



Loaded files appear in the left-hand column (along with rows for coverage). These are called TRACKS.

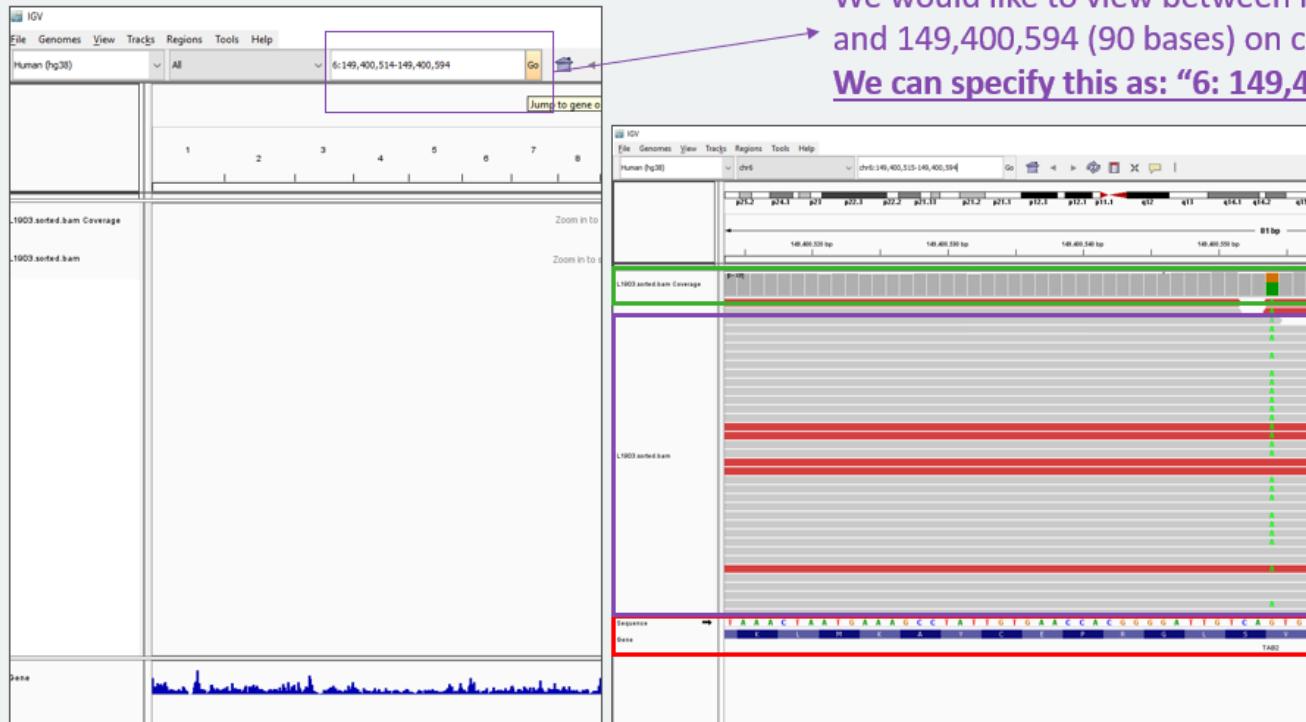
To save memory,

At this full genome level resolution (all chromosomes shown) reads not shown, must zoom in first.



Selecting a new location in a reference genome with IGV

1. IGV Will not display reads at the full-reference level, (or in many cases even the chromosome level by default).
 1. We will change this in view>preferences soon.
2. For now, we can change both the window zoom level (range) and location by specifying them in the “Enter a gene or locus” search box in the center of the top tool bar.



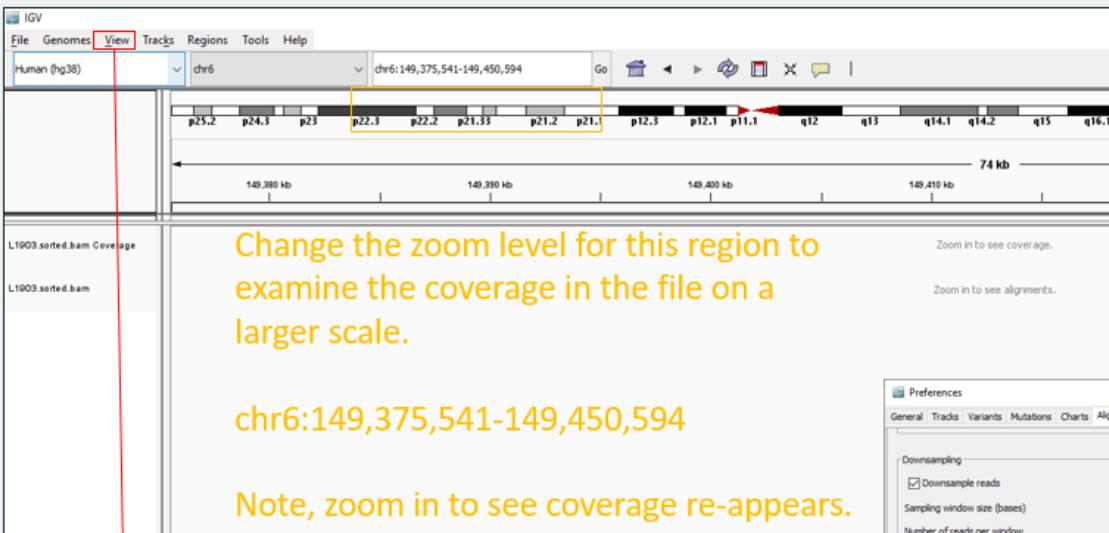
We would like to view between reference locus 149,400,514 and 149,400,594 (90 bases) on chromosome 6.

We can specify this as: "6: 149,400,514-149,400,594"

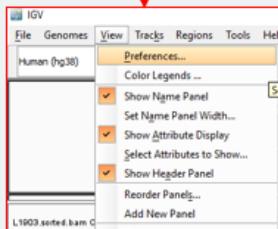
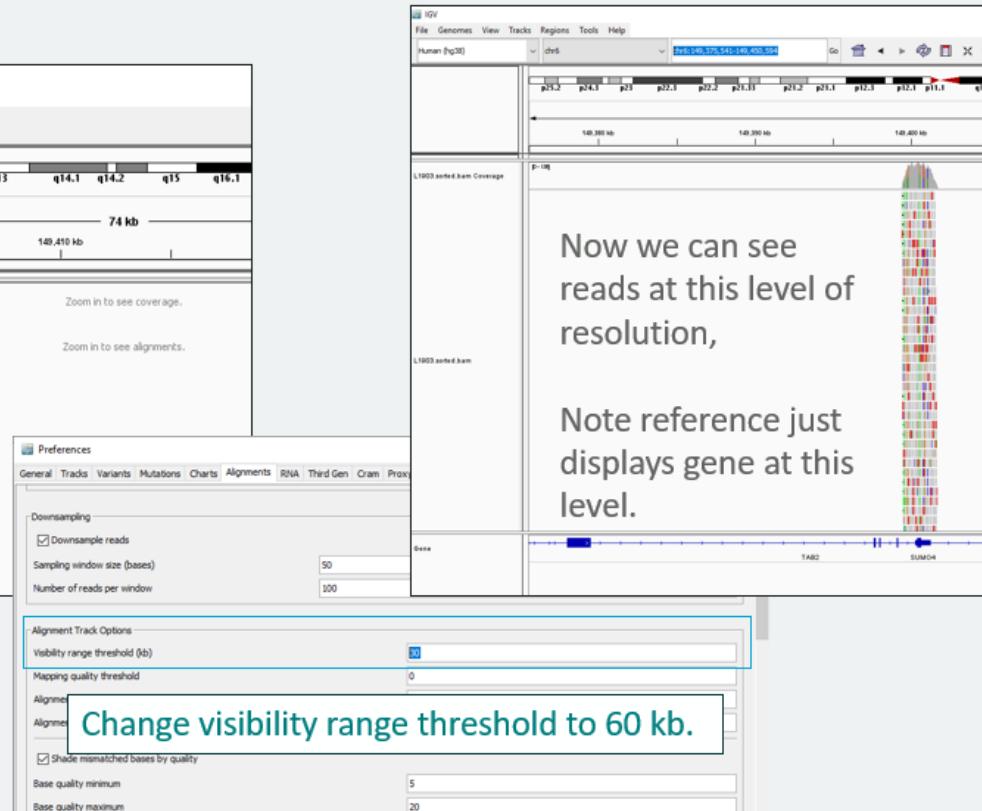
At this level of resolution (90 bp) we get **coverage information** from the **bam/bai** in the form of a **histogram**, **read alignment** information where reference matches are grey, and alternative bases are colored and labeled from the **bam/bai**, and **reference sequence information** from the **reference selected**.

Selecting a new location in a reference genome with IGV (changing visibility range)

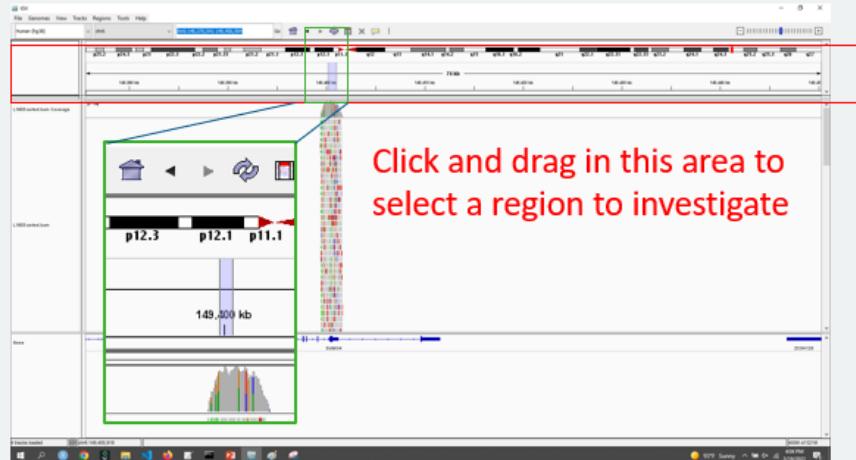
A
C
G
T



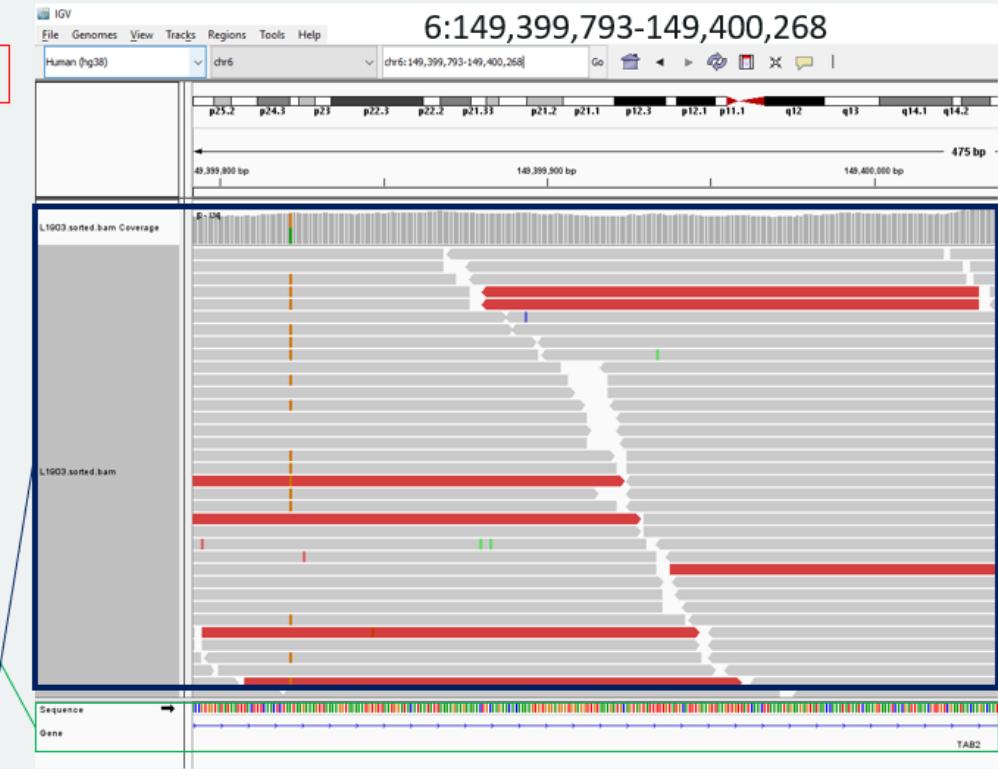
- To change the default for displaying coverage to allow viewing at this level go to
 - view>preferences>Alignments
 - In the Alignment Track Options box, change the Visibility range threshold (kb) from 30 to 60 and press “save”.



Navigating in the IGV Browser Window.



- At this level of resolution we can see bases are labeled by color in the reference
- Those reads with base calls differing from the reference are marked by color, those which are the same are grey.
- Click and drag in this region to pan manually in the sample.
- Scroll in this region to view more reads and files.



Navigating in the IGV Browser Window.

Zoom in and out manually with '+' and '-' buttons here.



Notice at this level of resolution that reads are sometimes displayed with different colors.

Red indicates there is possible evidence of deletion

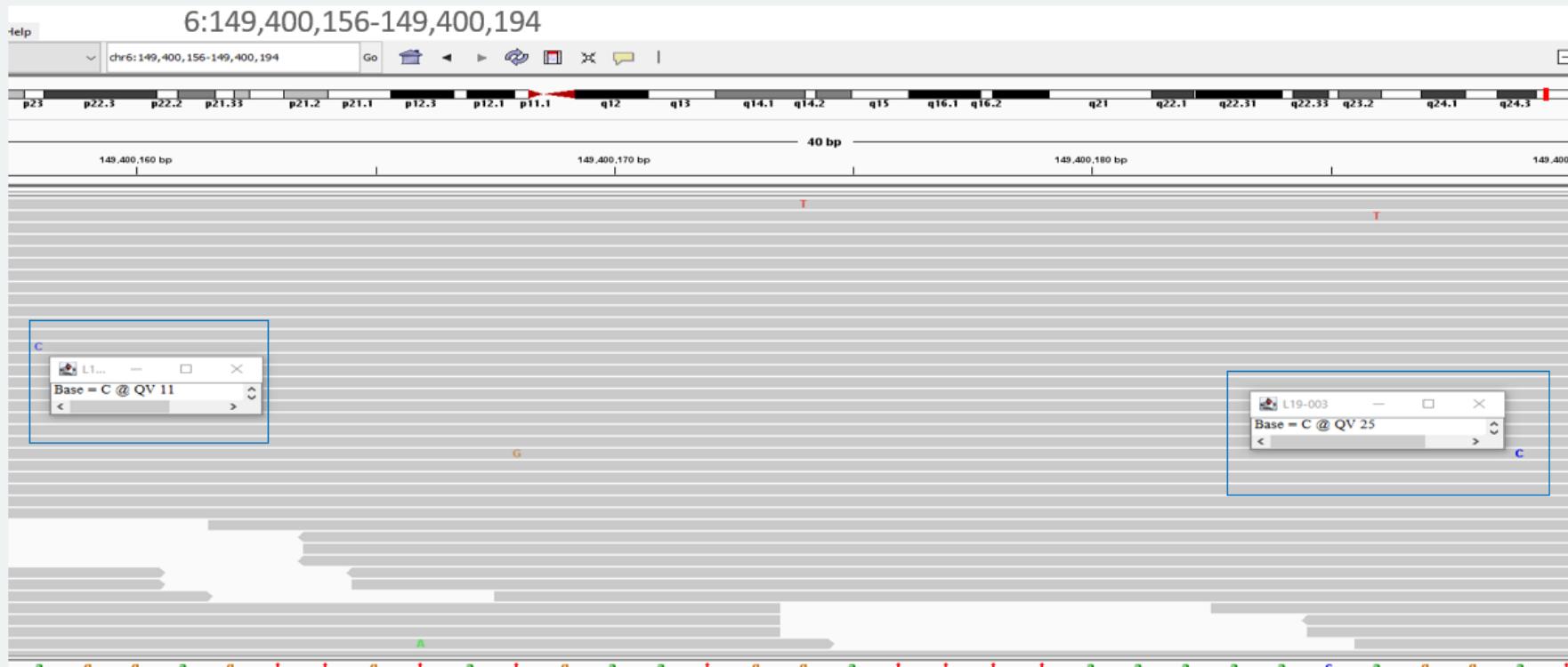
Blue indicates that there is possible evidence of an insertion

Green indicates orientation of the read orientation RL



A
C
G
T

Navigating in the IGV Browser Window.

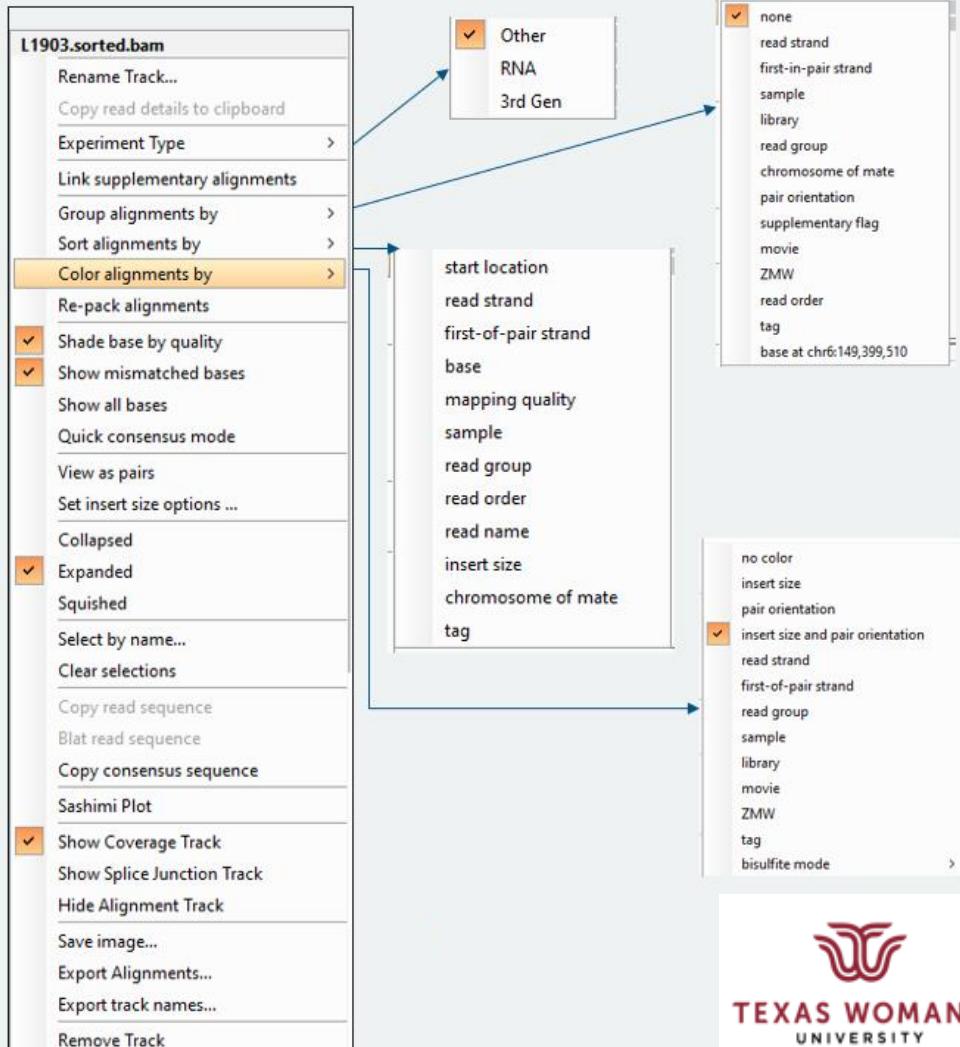


By Default, at this higher level of resolution the actual reference and base character is shown, we can see that they are also shaded lighter and darker by default depending on the quality value.

Changing IGV Window Options

In the right click menu, we can toggle the options in the display window, for instance, we can:

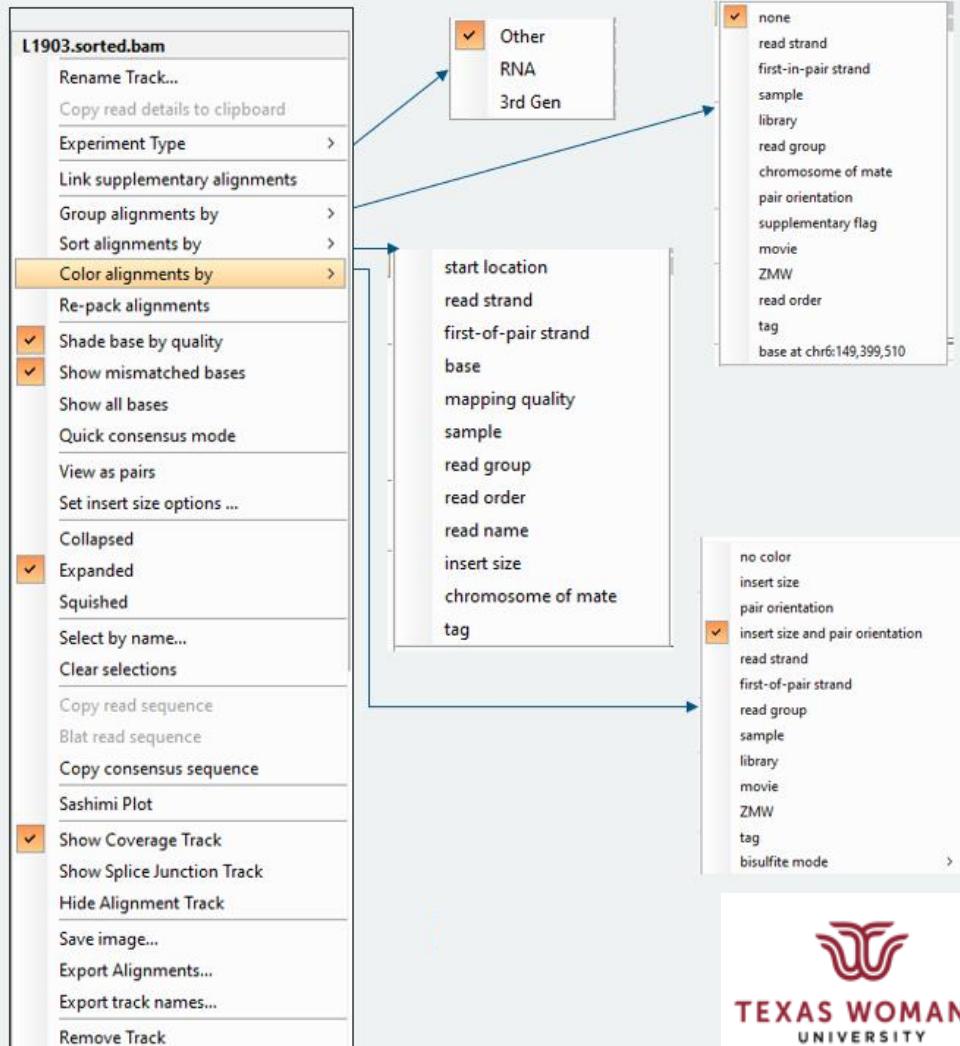
1. Change the name on the left, with rename track...
2. Change the Experiment type (other, RNA, 3rd gen)
3. Add additional alignments
4. Group alignments,
5. Sort alignments,
6. Color alignments,
7. Repack – redisplay
8. Toggle shading characters (darker for higher quality)
9. Toggle Whether the mismatched bases are highlighted.
10. Toggle Whether all of the bases are shown
11. Toggle quick consensus mode (only highlight alternative bases If consensus of base calls at loci [mode] is different.)
12. Toggle display to show pairs (and identify inserts)
13. Change insert options.



Changing IGV Window Options (2)

In the right click menu, we can toggle the options in the display window, for instance, we can:

14. Change the vertical resolution (scrunch in the rows to show them all at once (squished), maximum detail (Expanded), and medium detail (Collapsed)).
15. Select a specific read by it's name [if you happened to know it].
16. Clear your selection.
17. Copy the consensus (only for the selected region).
18. Display a sashimi plot [more useful for RNA sequences].
19. Show and hide default tracks (junctions off by default, really only useful for RNA-Seq).
20. Can save images of the current display
21. You can select specific alignments, and export them.
22. If you have a lot of tracks you can export track names to read them back in later.
23. Remove the track (you will have to add it back later to interpret it again).



Displaying info about a specific base call

A
C
G
T



```
L19-003
Read name = A00479:94:HWL7VDSXX:3:1340:21721:12900
Read length = 151bp
-----
Mapping = Primary @ MAPQ 60
Reference span = chr6:149,399,457-149,399,607 (+) = 151bp
Cigar = 151M
Clipping = None
-----
Mate is mapped = yes
Mate start = chr6:149399761 (-)
Insert size = 456
Second in pair
Pair orientation = F2R1
-----
XG = 0
NH = 1
NM = 5
XM = 5
XN = 0
XO = 0
AS = -15
YS = 0
ZS = -21
YT = CP
Hidden tags: MD
Location = chr6:149,399,513
Base = A @ QV 37
```

Information can be toggled to be displayed:

1. On mouse-button click.
2. On mouse-cursor hover.

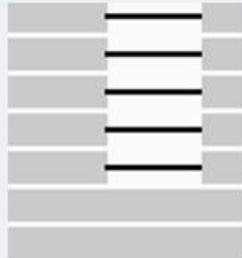
Can be persisted to compare.

Not persistent!

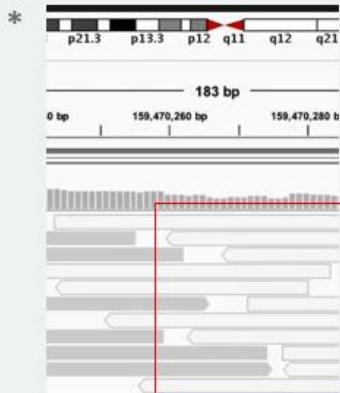
Recall that IGV is primarily a tool for displaying the results of sequence alignment, therefore the information we find here is what we would see in the SAM file.



A few additional notes on basic analysis with IGV:

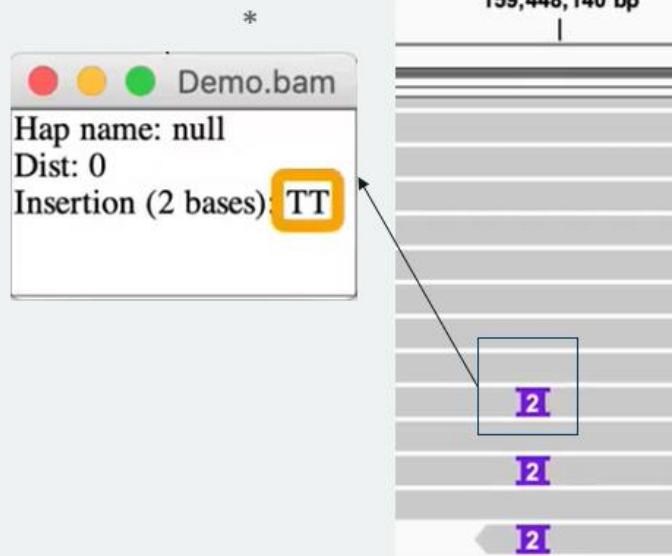


These symbols in the view window indicate gaps in the alignment, which is evidence of deletion.



Hollow reads represent low quality alignments

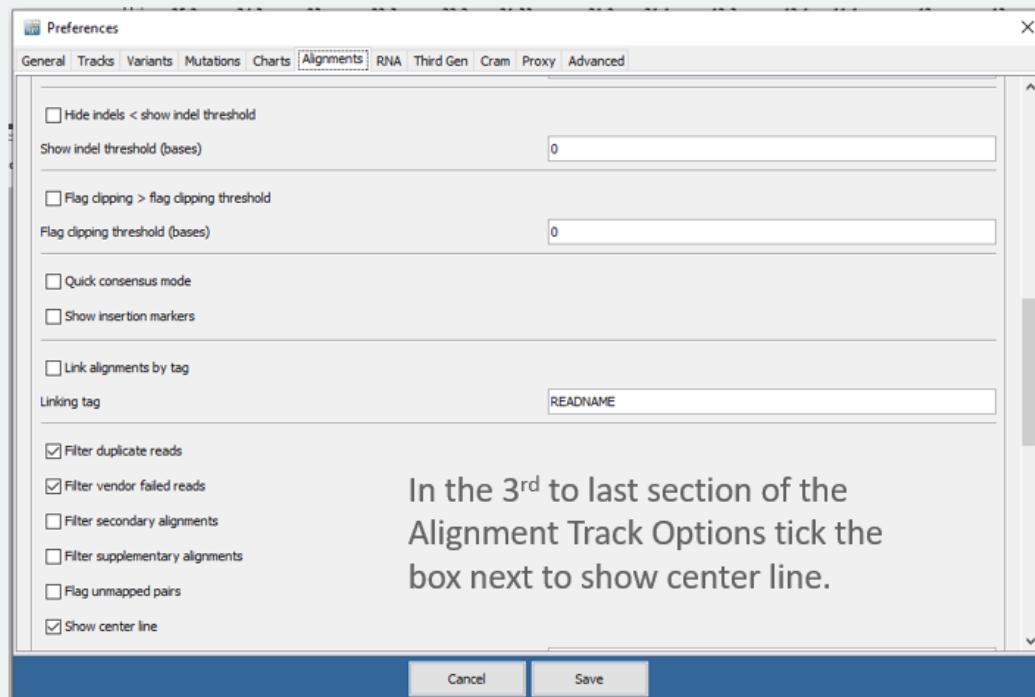
Insertions are shown by integers in these purple icons, when clicked the inserted bases will be displayed.



* From Official video tutorial: https://www.youtube.com/watch?v=E_G8z_2gTYM

A
C
G
T

Display helpful centerline in view window



In the 3rd to last section of the Alignment Track Options tick the box next to show center line.

The screenshot shows the IGV Preferences dialog box with the 'Alignments' tab selected. The 'Show center line' checkbox is highlighted with a green arrow. Other options include 'Hide indels < show indel threshold', 'Flag clipping > flag clipping threshold', 'Quick consensus mode', 'Show insertion markers', 'Link alignments by tag', 'Linking tag (READNAME)', and several filtering options like 'Filter duplicate reads' and 'Filter vendor failed reads'. At the bottom are 'Cancel' and 'Save' buttons.

To access the settings for displaying the centerline on the plot (dotted vertical lines) go through:

view>preferences>alignments

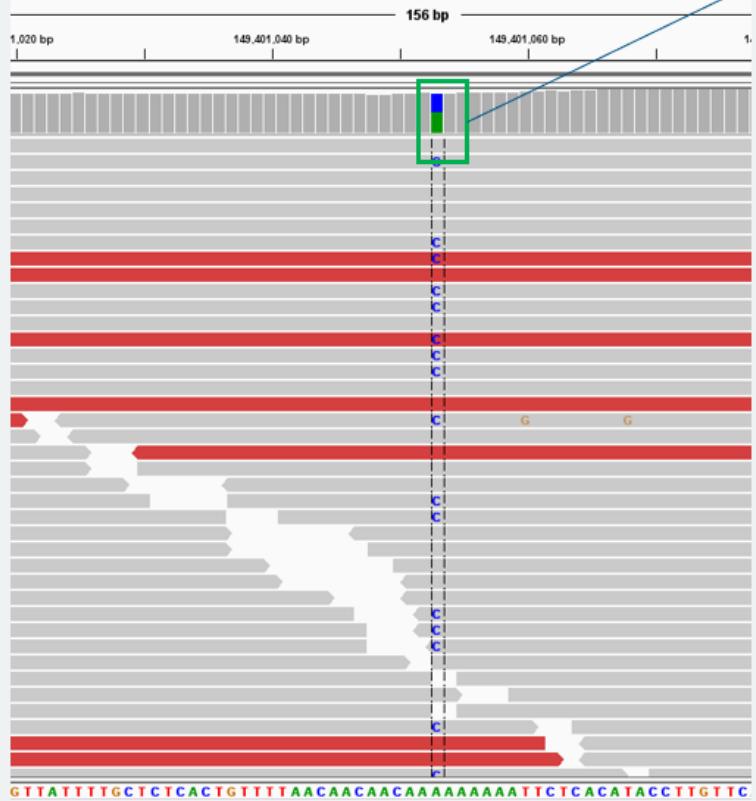
Note there are many useful options to explore here, depending on your experiment you may want to spend some time customizing your view window to display information in the most meaningful way.



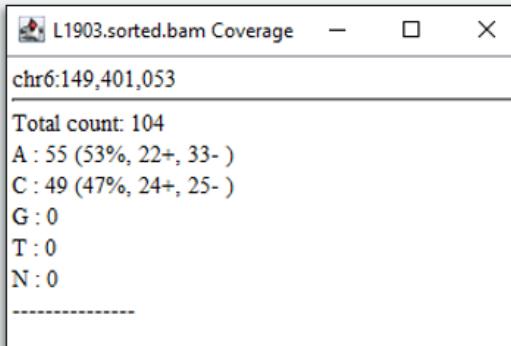
TEXAS WOMAN'S
UNIVERSITY

Viewing SNV information.

chr6:149,400,976-149,401,130



Examine Histogram colors in the coverage TRACK for a quick peak at how the base call distribution is broken down for a specific position.

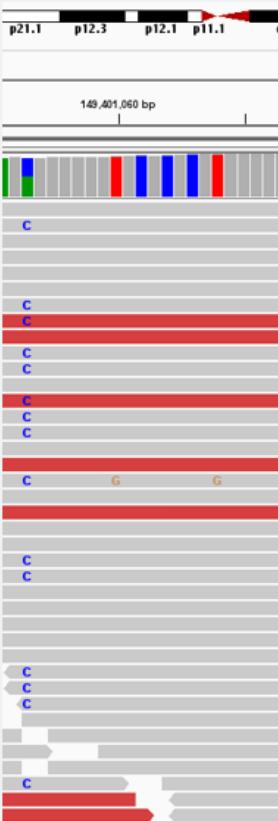


Click the bar in the coverage track for a more detailed breakdown of the distribution of base calls at a particular locus.

Note: Colored bars indicate imbalance in read base calls with frequency greater than a value defined by the user under

view>preferences
>alignments

In the Coverage Track Options box, in the Coverage allele-fraction threshold change 0.2f (20 percent floating point) for example to 0.001f, ridiculous, but can now see even one mismatch.



Viewing SNV information.

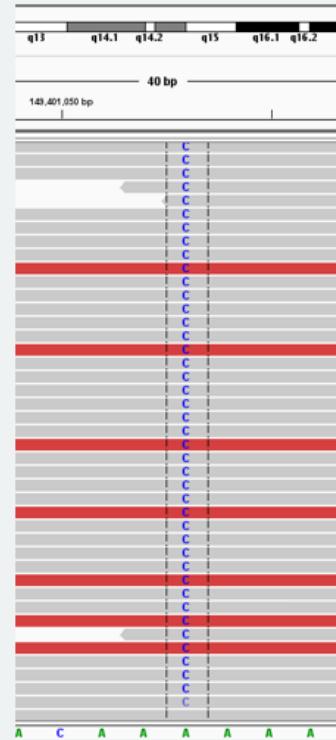
- Sorting reads by base can allow easy identification of snps.
- Note, these reads are also sorted in descending order of base call quality by default

L19-003

- Rename Track...
- Copy read details to clipboard
- Experiment Type >
- Link supplementary alignments
- Group alignments by >
- Sort alignments by >**
- Color alignments by >
- Re-pack alignments
- Shade base by quality
- Show mismatched bases
- Show all bases
- Quick consensus mode
- View as pairs
- Go to mate
- View mate region in split screen
- Set insert size options ...
- Collapsed

base

- start location
- read strand
- first-of-pair strand
- base
- mapping quality
- sample
- read group
- read order
- read name
- insert size
- chromosome of mate
- tag

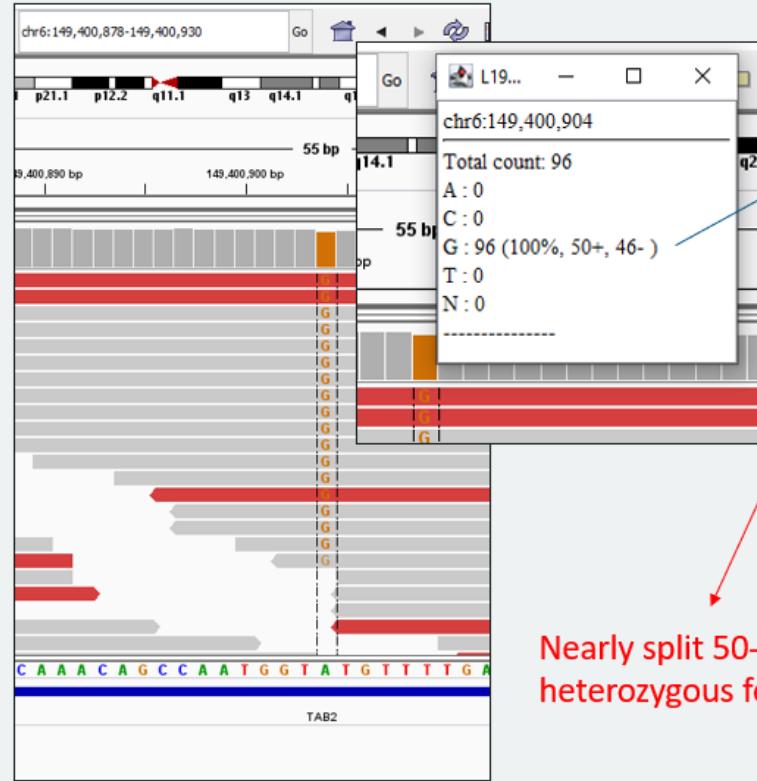


Be Cautious of setting values too low,
This may make the visualization
Misleading.

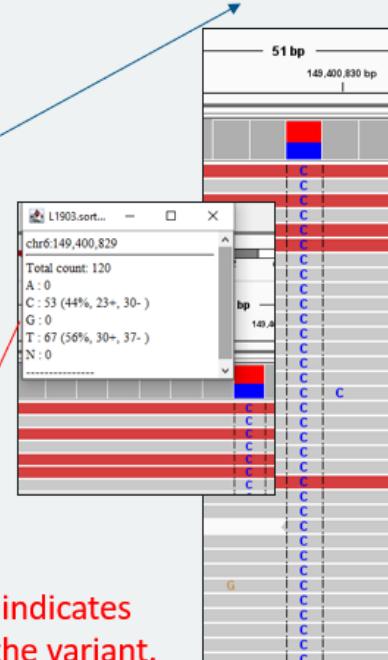
Recall, decreasing quality is
evidenced by fading of the
intensity of the color displayed.

Viewing SNV information.

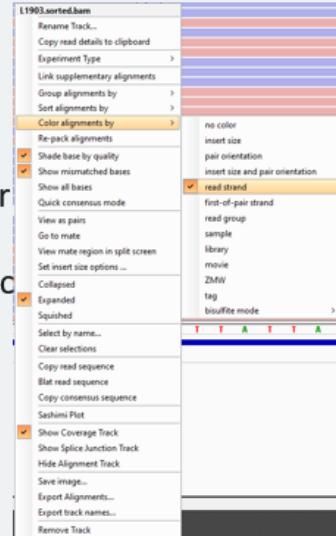
- We can also use IGV to make determinations as to whether particular variants indicate homozygous or heterozygous.
- Consider Chr6:149,400,904



All reads mapping to location differ from reference, and therefore this variant (A>G) appears to be homozygous in the sample.



Can investigate strand bias
(variant only appearing in near 50-50 ratio on strands of specific orientation) by coloring alignments by read strand.



- Consider Chr6:149,400,829

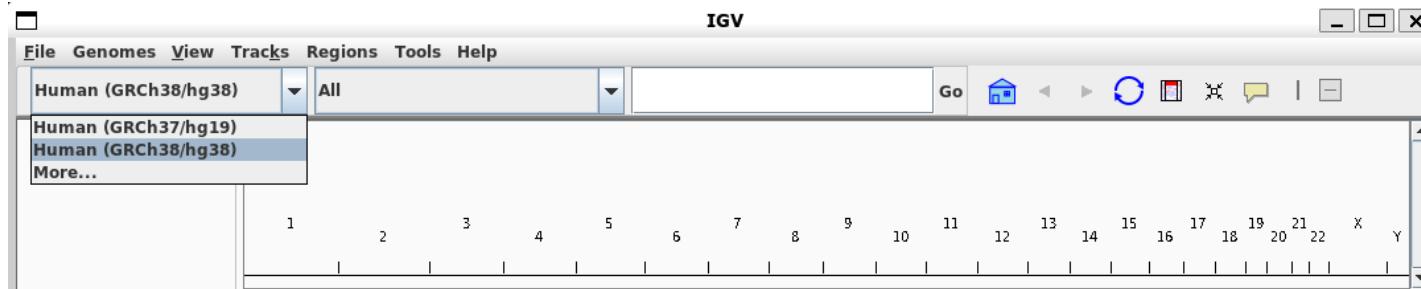
IGV Tutorial

- Now we are going to work with IGV to investigate a sample.
- SRA Toolkit helps to download samples easily



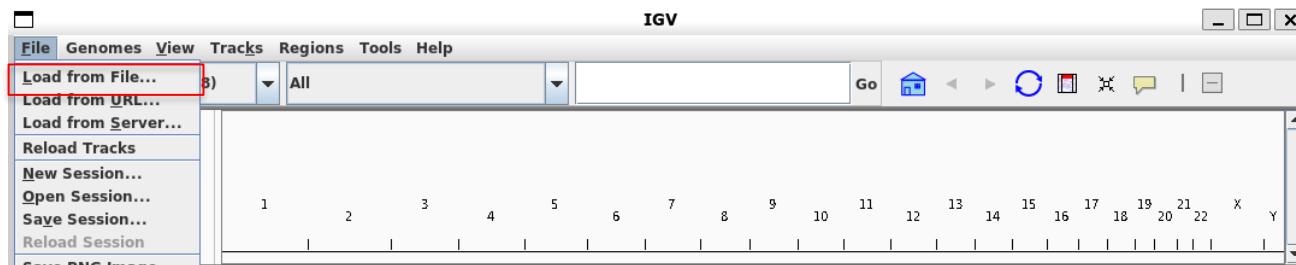
IGV Tutorial

- Download the .bam and .bai alignment files by opening a terminal window and typing the following command in
 - `wget https://github.com/mathornton01/twupa_bioinformaticsws_2024/raw/main/data/bamfiles/eyes_1.zip`
- Extract the zipped bamfiles folder with
 - `unzip eyes_1.zip`
- Open IGV and load the Hg38 human genome reference (you should have this downloaded already)

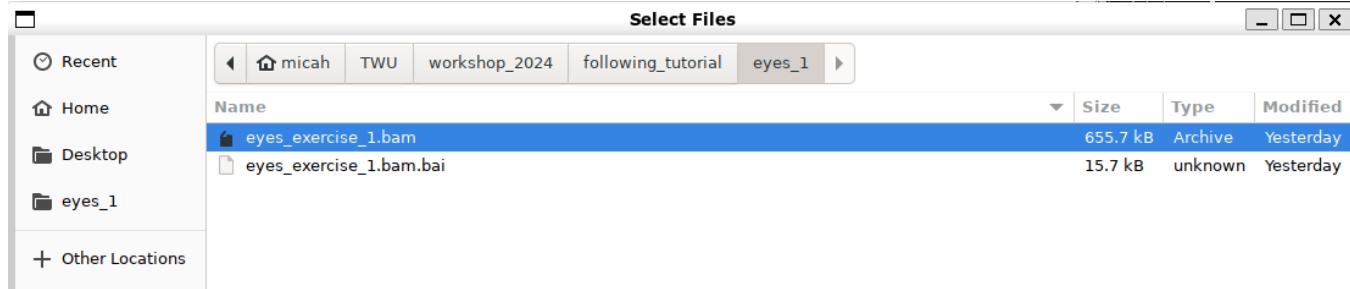


IGV Tutorial

- Load the eyes_exercise_1.bam file (note that the .bai index file must be in the same directory as the .bam file)



IGV Tutorial

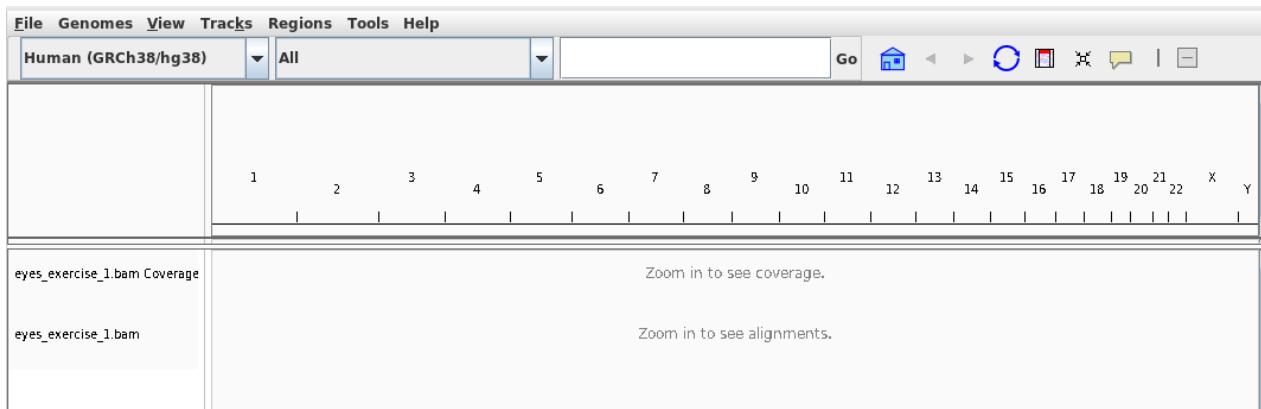


- Select the `eyes_exercise_1.bam` file (you may have to navigate to the appropriate directory [linux])



IGV Tutorial

- You should now see the coverage and alignment tracks for the eyes_exercise_1.bam file



IGV Tutorial

Table 2
Eye color predictor.

Gene	SNP ID	Genotype	Eye color (predicted)
HERC2	rs12913832	G/G	Not brown
HERC2	rs12913832	G/A	Not blue
HERC2	rs12913832	A/A	Not blue
HERC2	rs12913832	A/A	Not blue
IRF4	rs12203592	T/T	
HERC2	rs12913832	G/A	Green
IRF4	rs12203592	T/T	
HERC2	rs12913832	G/G	Green
SLC45A2	rs16891982	C/C	
HERC2	rs12913832	A/A or G/A	Brown
SLC45A2	rs16891982	C/C	
HERC2	rs12913832	A/A or G/A	Brown
OCA2	rs1545397	T/T	
HERC2	rs12913832	A/A or G/A	Brown
MC1R	rs885479	A/A	
HERC2	rs12913832	A/A or G/A	Brown
ASIP	rs6119471	G/G	

- This table will help us identify the eye color of the individual from whom the eye_exercise_1.bam file was taken.



IGV Tutorial

Table 2
Eye color predictor.

Gene	SNP ID	Genotype	Eye color (predicted)
HERC2	rs12913832	G/G	Not brown
HERC2	rs12913832	G/A	Not blue
HERC2	rs12913832	A/A	Not blue
HERC2	rs12913832	A/A	Not blue
IRF4	rs12203592	T/T	
HERC2	rs12913832	G/A	Green
IRF4	rs12203592	T/T	
HERC2	rs12913832	G/G	Green
SLC45A2	rs16891982	C/C	
HERC2	rs12913832	A/A or G/A	Brown
SLC45A2	rs16891982	C/C	
HERC2	rs12913832	A/A or G/A	Brown
OCA2	rs1545397	T/T	
HERC2	rs12913832	A/A or G/A	Brown
MC1R	rs885479	A/A	
HERC2	rs12913832	A/A or G/A	Brown
ASIP	rs6119471	G/G	

- We need to identify where the SNPs for these given SNP IDs are located in GRCh38



IGV Tutorial

Table 2
Eye color predictor.

Gene	SNP ID	Genotype	Eye color (predicted)
HERC2	rs12913832	G/G	Not brown
HERC2	rs12913832	G/A	Not blue
HERC2	rs12913832	A/A	Not blue
HERC2	rs12913832	A/A	Not blue
IRF4	rs12203592	T/T	
HERC2	rs12913832	G/A	Green
IRF4	rs12203592	T/T	
HERC2	rs12913832	G/G	Green
SLC45A2	rs16891982	C/C	
HERC2	rs12913832	A/A or G/A	Brown
SLC45A2	rs16891982	C/C	
HERC2	rs12913832	A/A or G/A	Brown
OCA2	rs1545397	T/T	
HERC2	rs12913832	A/A or G/A	Brown
MC1R	rs885479	A/A	
HERC2	rs12913832	A/A or G/A	Brown
ASIP	rs6119471	G/G	

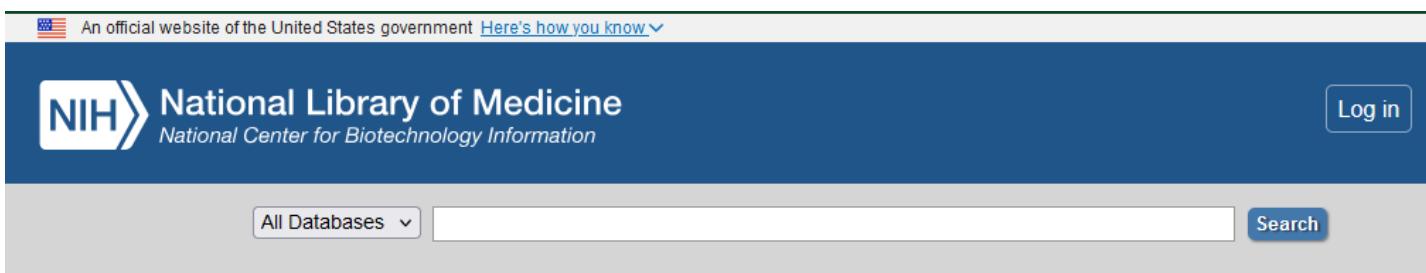
- Let us identify where the SNP corresponding to ID: “rs12913832” is located.



IGV Tutorial

- Visit the website:

<https://www.ncbi.nlm.nih.gov>



An official website of the United States government [Here's how you know](#)

NIH National Library of Medicine
National Center for Biotechnology Information

All Databases

NCBI Home
Resource List (A-Z)
All Resources
Chemicals & Bioassays
Data & Software

Welcome to NCBI

The National Center for Biotechnology Information advances science and health by providing access to biomedical and genomic information.

[About the NCBI](#) | [Mission](#) | [Organization](#) | [NCBI News & Blog](#)

Popular Resources

[PubMed](#)
[Bookshelf](#)
[PubMed Central](#)
[BLAST](#)



TEXAS WOMAN'S
UNIVERSITY

IGV Tutorial

- Select SNP from the dropdown menu

The screenshot shows the NCBI homepage with the 'All Databases' dropdown menu open. The 'SNP' option is highlighted in the list.

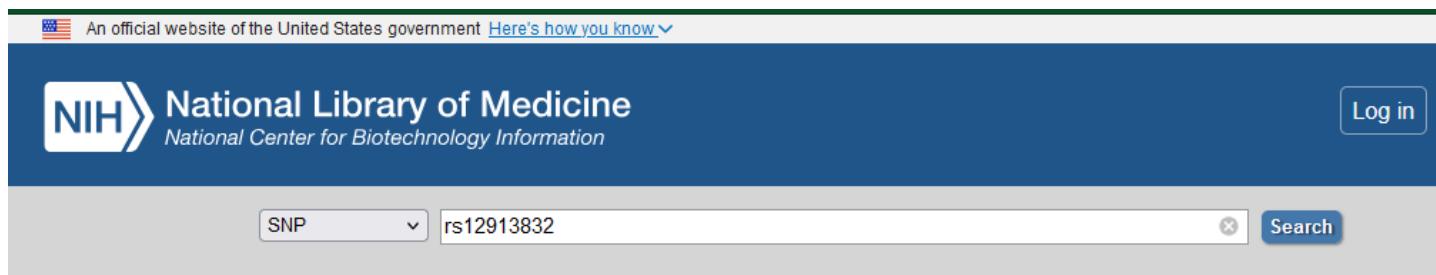
- All Databases
- OMIM
- PMC
- PopSet
- Protein
- Protein Clusters
- Protein Family Models
- PubChem BioAssay
- PubChem Compound
- PubChem Substance
- PubMed
- SNP**
- SRA
- Structure
- Taxonomy
- ToolKit
- ToolKitAll
- ToolKitBook



TEXAS WOMAN'S
UNIVERSITY

IGV Tutorial

- Type the SNP ID we are trying to identify (rs12913832) into the search bar.



TEXAS WOMAN'S
UNIVERSITY

IGV Tutorial

- Collect the location and allelic variant information from this page.
- Position- 15:28120472
- Variants- A>C/G

SNP rs12913832 Search Create alert Advanced

Display Settings: Summary, Sorted by SNP_ID Send to: Filters: Manage Filters

Search results Items: 2

rs12913832 [Homo sapiens]

1.

Variant type:	SNV
Alleles:	A>C, G [Show Flanks]
Chromosome:	15:28120472 (GRCh38) 15:28365618 (GRCh37)

Canonical SPDI: NC_000015.10:28120471:A;NC_000015.10:28120471:A;G
Gene: HERC2 (Vanview)
Functional Consequence: generic_downstream_transcript_variant,intron_variant association
Clinical significance:
Validated:
by frequency, by alfa, by cluster
MAF:
A=0.366882/89771 (ALFA)
G=0.000035/1 (TOMMO)
G=0.001027/3 (KOREAN)
...more
HGVS:
NC_000015.10:g.28120472A>C, NC_000015.10:g.28120472A>G,
NC_000015.9:g.28365618A>C, NC_000015.9:g.28365618A>G,
NG_016355.1:g.206678T>G, NG_016355.1:g.206678T>C,
NM_044200.7:c.112_113G>A, NM_044200.7:c.112_113G>C

Find related data Database: Select Find items

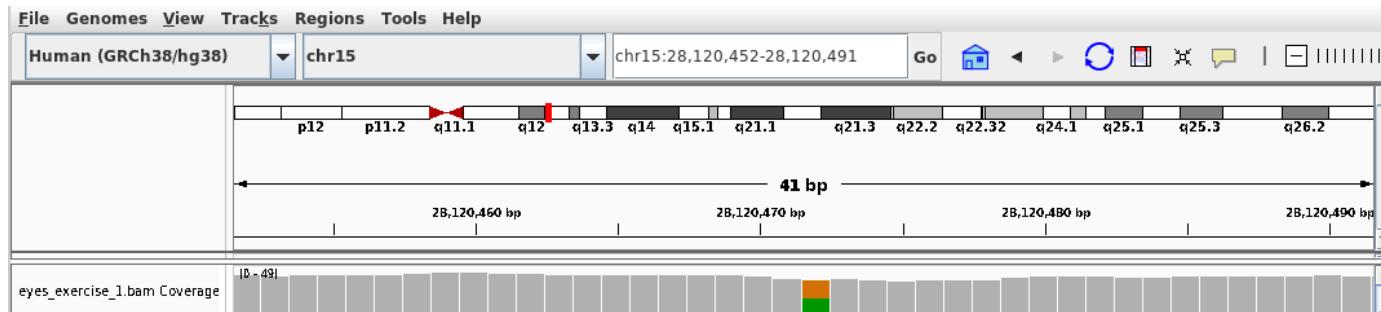
Search details rs12913832[All Fields] Search

Recent activity rs12913832 (2) Tun High coverage genome of tl



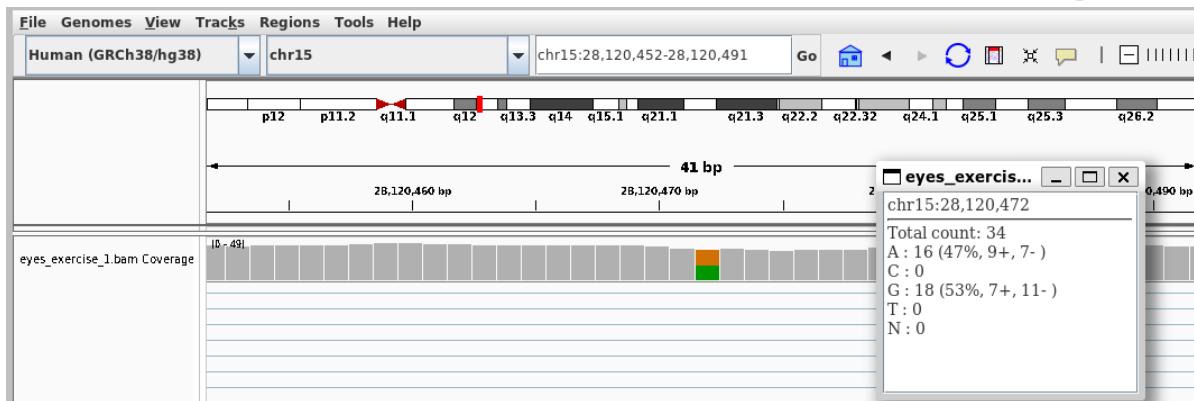
IGV Tutorial

- Return to IGV, and type this location into the search bar at the top of the application window then press “Go”.



IGV Tutorial

- Select the dual-colored coverage histogram indicator in the coverage track for information about the sample.



IGV Tutorial

- What would you guess was the genotype of this individual? (A/G)
- What does this indicate about the individual according to our table?

Table 2
Eye color predictor.

Gene	SNP ID	Genotype	Eye color (predicted)
HERC2	rs12913832	G/G	Not brown
HERC2	rs12913832	G/A	Not blue
HERC2	rs12913832	A/A	Not blue
HERC2	rs12913832	A/A	Not blue
IRF4	rs12203592	T/T	
HERC2	rs12913832	G/A	Green
IRF4	rs12203592	T/T	
HERC2	rs12913832	G/G	Green
SLC45A2	rs16891982	C/C	
HERC2	rs12913832	A/A or G/A	Brown
SLC45A2	rs16891982	C/C	
HERC2	rs12913832	A/A or G/A	Brown
OCA2	rs1545397	T/T	
HERC2	rs12913832	A/A or G/A	Brown
MC1R	rs885479	A/A	
HERC2	rs12913832	A/A or G/A	Brown
ASIP	rs6119471	G/G	



IGV Tutorial

- Which SNP should we check next?
 - How about rs12203592?
 - What is the most likely eye color for this person?

Table 2
Eye color predictor.

Gene	SNP ID	Genotype	Eye color (predicted)
HERC2	rs12913832	G/G	Not brown
HERC2	rs12913832	G/A	Not blue
HERC2	rs12913832	A/A	Not blue
HERC2	rs12913832	A/A	Not blue
IRF4	rs12203592	T/T	
HERC2	rs12913832	G/A	Green
IRF4	rs12203592	T/T	
HERC2	rs12913832	G/G	Green
SLC45A2	rs16891982	C/C	
HERC2	rs12913832	A/A or G/A	Brown
SLC45A2	rs16891982	C/C	
HERC2	rs12913832	A/A or G/A	Brown
OCA2	rs1545397	T/T	
HERC2	rs12913832	A/A or G/A	Brown
MC1R	rs885479	A/A	
HERC2	rs12913832	A/A or G/A	Brown
ASIP	rs6119471	G/G	



Minor Allele Frequency

- Minor Allele Frequency (MAF) is an important value in genomic studies.
- Each read indicates one allele (for haploid organisms there is only one allele in the sample)
- For polyploid (diploid and higher ploidy) there can be up to four possible alleles.



Minor Allele Frequency

- In a given population the frequency of the least commonly occurring allele is known as the minor allele frequency.



Minor Allele Frequency

- Suppose we observe the following haplotypes in the population.
 - [AT, TT, TT, TT, TA, AA, TA]
- To calculate the MAF, we must first identify the number of A and T in the sample
- There are 5 A, and 9 T, for a total of 14 Bases.
 - The minor Allele for this particular SNP then would be A, and the MAF is 5/14 which is a little less than 1/3.



Minor Allele Frequency

- The Multinomial distribution is a distribution on categorical values.
 - Just as the binomial distribution is a distribution on two values (usually represented 0, and 1)



Minor Allele Frequency

- Suppose we have a random variable A_i , (usually random variables are represented by capital Latin characters)
- Let the support of A_i be denoted symbolically by the set \mathcal{S}_A ,
 - If A_i is defined as the allele observed at position p of the i th read in a sample of N reads, we may write \mathcal{S}_A as {A,C,G,T}.
 - Now let us define π_j as the probability that $A_i = j$ for $j \in \mathcal{S}_A$



Minor Allele Frequency

- Now let us assume $A_i \stackrel{IID}{\sim} (\pi)$, that is A_i are distributed identically and independently according to the parameter $\pi = [\pi_A, \pi_C, \pi_G, \pi_T]^T$



Minor Allele Frequency

- Suppose the vector random variable
 - $A = [A_1, A_2, \dots, A_N]^T$
- Takes on valuation
 - $a = [a_1, a_2, \dots, a_N]^T$ for $a_i \in \mathcal{S}_A$
- For a given read i we then have
 - $\Pr(A_i = a_i) = \pi_A^{\mathbb{1}(a_i \equiv A)} \pi_C^{\mathbb{1}(a_i \equiv C)} \pi_G^{\mathbb{1}(a_i \equiv G)} \pi_T^{\mathbb{1}(a_i \equiv T)}$



Minor Allele Frequency

- Here $\mathbb{1}(\cdot) = 1$ if \cdot is True, otherwise 0 if False.
- Now suppose we are interested in the number of reads that indicate the base at position p is “A”.

$$- C_B = \sum_{\{i=1\}}^N \mathbb{1}(A_i = B)$$



Minor Allele Frequency

- Suppose that the coverage is $2x$, therefore 2 reads cover position p ,
 - Symbolically $N = 2$,
 - C_A, C_C, C_G, C_T may take on only the values 0, 1 or 2.
 - $\mathbf{C} = [C_A, C_C, C_G, C_T]$ then can only take on values such that its sum is 0, 1, or 2.



Minor Allele Frequency

- $C \in$
 - $[0,0,0,2], [0,0,2,0], [0,2,0,0], [2,0,0,0]$
 - $[0,0,1,1], [0,1,0,1], [1,0,0,1]$
 - $[0,1,1,0], [1,0,1,0],$
 - $[1,1,0,0]$



Minor Allele Frequency

- Say we have two reads r_1, r_2 Then C takes on values according to the following
 - $[0,0,0, r_1 \& r_2], [0,0, r_1 \& r_2, 0], [0, r_1 \& r_2, 0,0], [r_1 \& r_2, 0,0,0]$
 - $\{[0,0,r_1, r_2], [0,0,r_2, r_1]\}, \{[0, r_2,0, r_1], [0, r_1,0, r_2]\}, \{[r_2,0,0,r_1], [r_1,0,0,r_2]\}$
 - $\{[0,r_1,r_2,0], [0,r_2,r_1,0]\}, \{[r_2,r_1,0,0], [r_1,r_2,0,0]\},$
 - $\{[r_1,r_2,0,0], [r_2,r_1,0,0]\}$



Minor Allele Frequency

- Let's work out the probability for each:
 - $\Pr(C = [0,0,0,2]) = (1) \cdot \pi_T^2$
 - $\Pr(C = [0,0,2,0]) = (1) \cdot \pi_G^2$
 - $\Pr(C = [0,2,0,0]) = (1) \cdot \pi_C^2$
 - $\Pr(C = [2,0,0,0]) = (1) \cdot \pi_A^2$



Minor Allele Frequency

- Let's work out the probability for each:

- $\Pr(C = [0,0,1,1]) = \pi_T^1 \pi_G^1 + \pi_G^1 \pi_T^1 = (2) \cdot \pi_T^1 \pi_G^1$

- $\Pr(C = [0,1,0,1]) = \pi_T^1 \pi_C^1 + \pi_C^1 \pi_T^1 = (2) \cdot \pi_T^1 \pi_C^1$

- $\Pr(C = [1,0,0,1]) = \pi_T^1 \pi_A^1 + \pi_G^1 \pi_A^1 = (2) \cdot \pi_T^1 \pi_A^1$



Minor Allele Frequency

- Let's work out the probability for each:

- $\Pr(C = [0,1,1,0]) = \pi_C^1 \pi_G^1 + \pi_G^1 \pi_C^1 = (2) \cdot \pi_C^1 \pi_G^1$

- $\Pr(C = [1,0,1,0]) = \pi_A^1 \pi_G^1 + \pi_G^1 \pi_A^1 = (2) \cdot \pi_G^1 \pi_A^1$

- $\Pr(C = [1,1,0,0]) = \pi_C^1 \pi_A^1 + \pi_C^1 \pi_A^1 = (2) \cdot \pi_C^1 \pi_A^1$



Minor Allele Frequency

- In general for N reads we can write:
 - $\Pr(\mathbf{C} = [c_A, c_C, c_G, c_T] = [c_A, c_C, c_G, c_T]) =$
$$\frac{N!}{c_A!c_C!c_G!c_T!} \pi_A^{\{c_A\}} \pi_C^{\{c_C\}} \pi_G^{\{c_G\}} \pi_T^{\{c_T\}}$$
- And $c_A + c_C + c_G + c_T = N$
- And Say $\mathbf{C} \sim \text{MN}(\boldsymbol{\pi})$ (\mathbf{C} is distributed multinomially with parameter $\boldsymbol{\pi}$.)



Minor Allele Frequency

- We are interested in the distribution of the minimum category
 - Turns out, this is possibly impossible to specify analytically
 - We can simulate the distribution though!

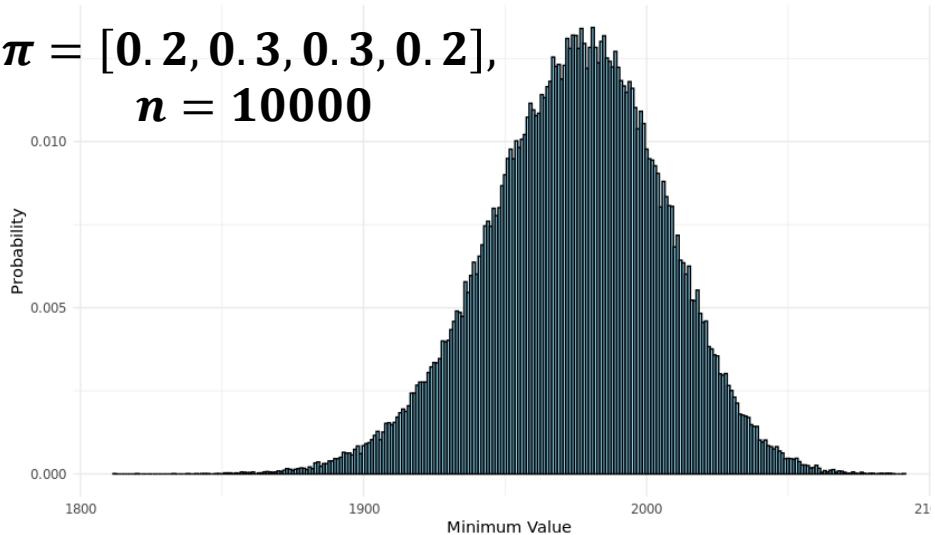


Minor Allele Frequency

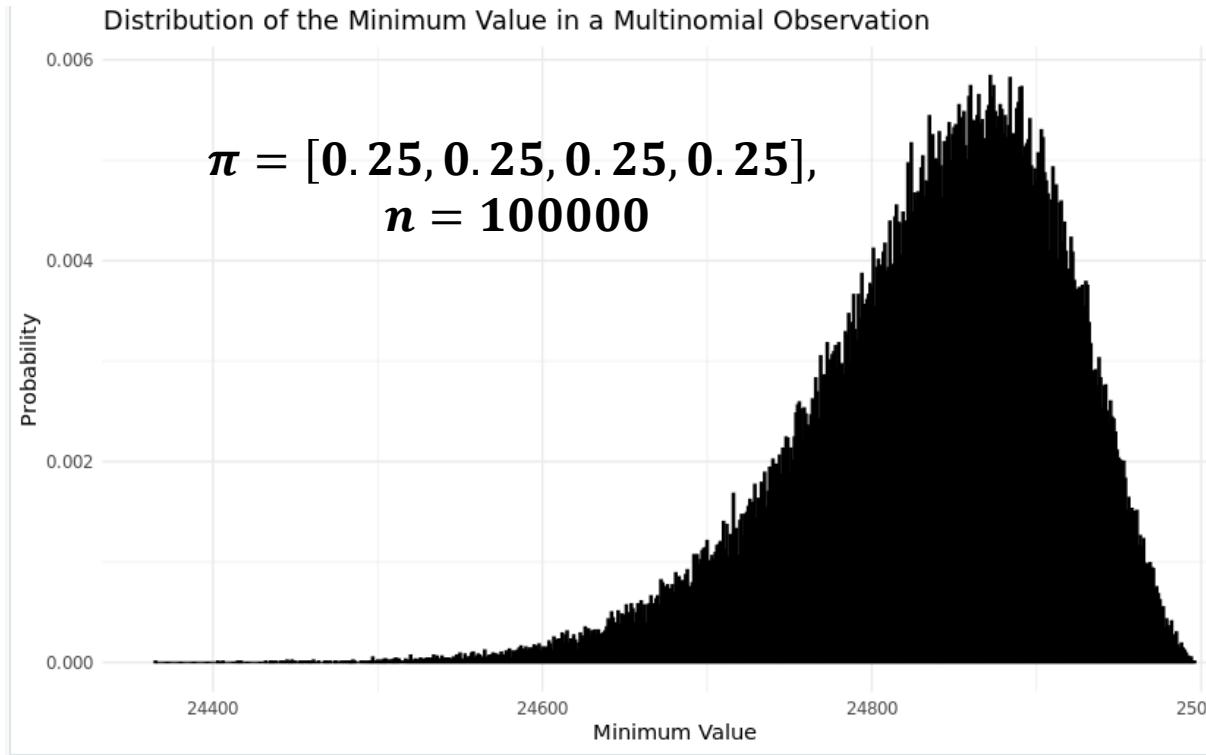
```
1 # Load necessary libraries
2 library(ggplot2)
3
4 # Parameters
5 n <- 10000 # number of trials
6 p <- c(0.2, 0.3, 0.3, 0.2) # probabilities for each outcome
7 num_samples <- 100000 # number of samples for the simulation
8
9 # Function to generate samples and compute the minimum values
10 simulate_minimum_multinomial <- function(n, p, num_samples) {
11   min_values <- numeric(num_samples)
12   for (i in 1:num_samples) {
13     sample <- rmultinom(1, n, p)
14     min_values[i] <- min(sample)
15   }
16   return(min_values)
17 }
18
19 # Generate the samples
20 min_values <- simulate_minimum_multinomial(n, p, num_samples)
21
22 # Create a data frame for plotting
23 df <- data.frame(min_values)
24
25 # Plot the distribution of the minimum values
26 ggplot(df, aes(x = min_values)) +
27   geom_histogram(binwidth = 1, color = "black", fill = "skyblue", aes(y = ..density..)) +
28   labs(x = "Minimum Value", y = "Probability", title = "Distribution of the Minimum Value in a Multinomial Observation") +
29   theme_minimal()
```

Distribution of the Minimum Value in a Multinomial Observation

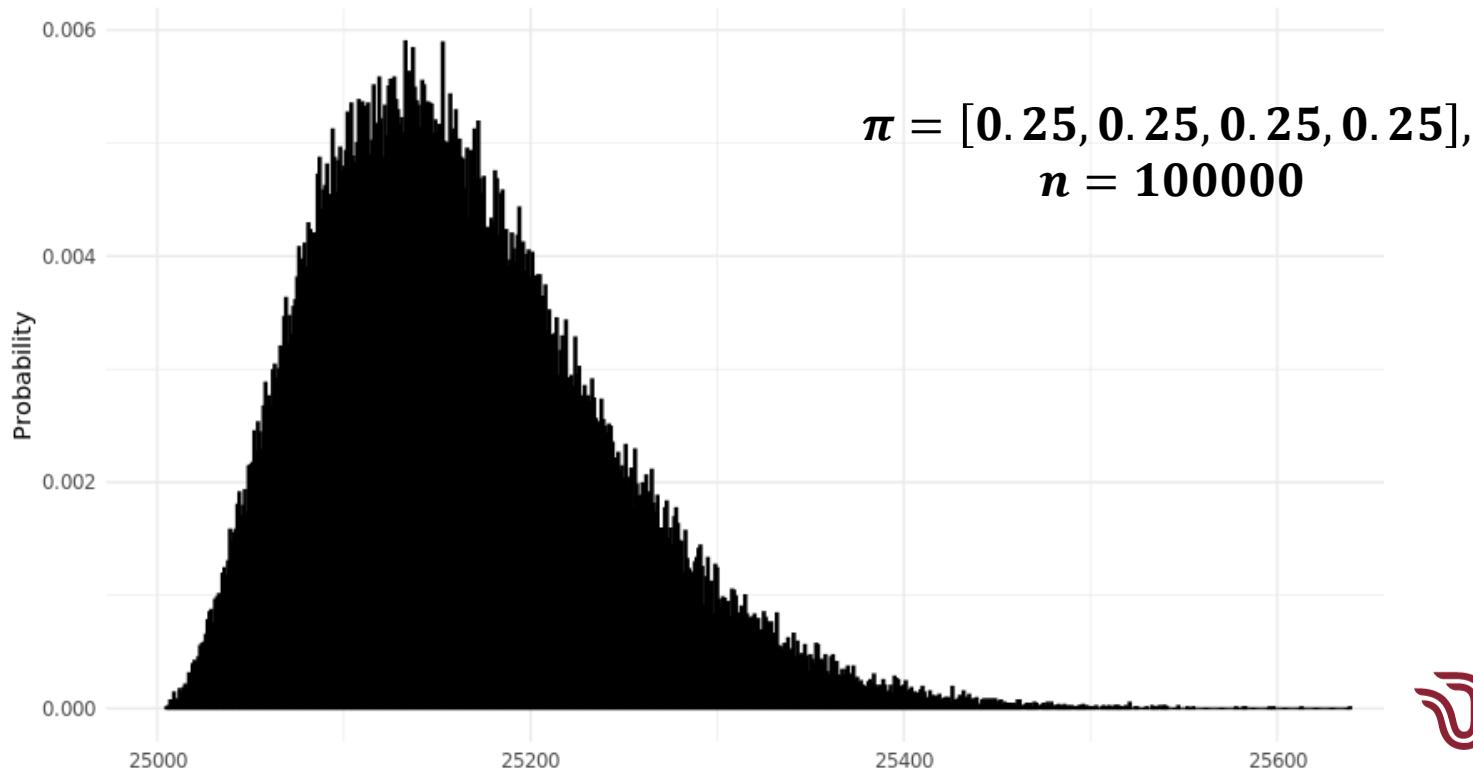
$$\pi = [0.2, 0.3, 0.3, 0.2], \\ n = 10000$$



Minor Allele Frequency



Maximum Allele Frequency



TEXAS WOMAN'S
UNIVERSITY

Allelic Frequencies

- Open problem to characterize distribution of minimum for uniform probabilities, or to find stricter boundaries for it.

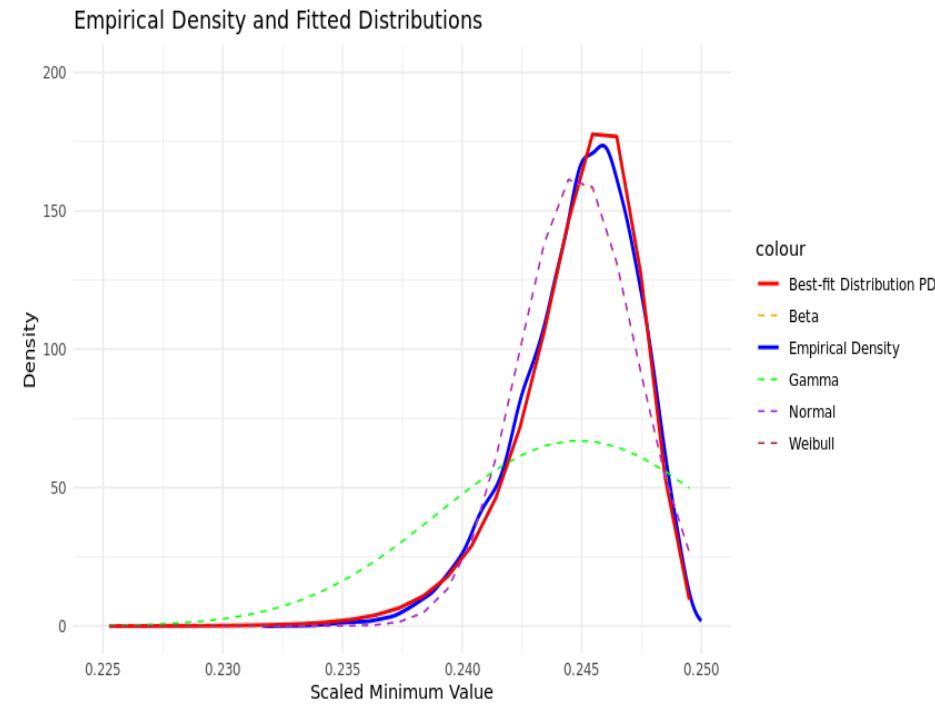


Fitting distributions

$$f(x) = \frac{122.3058}{0.2459783} \left(\frac{x}{0.2459783} \right)^{122.3058-1} \exp \left(-\left(\frac{x}{0.2459783} \right)^{122.3058} \right)$$

Question:

How do the corresponding parameters (number of bins, and number of observations) relate to the parameters of the Weibull Distribution?



Back to IGV

- Download the following file

```
wget https://github.com/mathornton01/twupa\_bioinformaticsws\_2024/raw/main/data/bamfiles/eth\_1.zip
```
- **Use what you have learned so far to look up and write down the alleles for each of the following SNPs.**
- **Then guess what the most likely ethnicity of the person is from the graphic.**



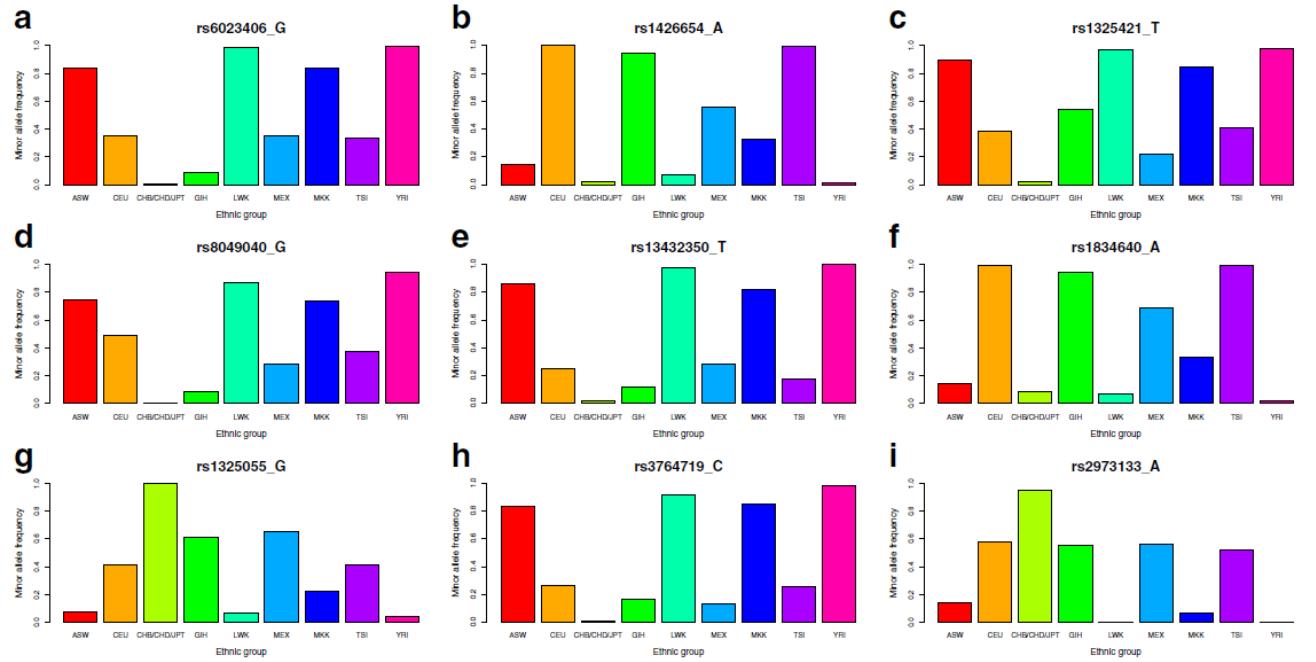


Fig. 3 The minor allele frequency of the top nine SNPs in each ethnic group. The minor allele frequencies of the top three SNPs, rs6023406 (a), rs1426654 (b), rs1325421 (c), rs8049040 (d), rs13432350 (e), rs1834640 (f), rs1325055 (g), rs3764719 (h), rs2973133 (i) in the nine ethnic groups were plotted. Each ethnic group has their own specific alleles. For example, the allele frequencies of rs6023406_G, rs1426654_A, rs1325421_T, rs8049040_G, rs13432350_T, rs1834640_A and rs3764719_C were very low, but those of rs1325055_G and rs2973133_A were very high in the Asian population (CHB/CHD/JPT)

Huang, T., Shu, Y. & Cai, YD. Genetic differences among ethnic groups. *BMC Genomics* **16**, 1093 (2015). <https://doi.org/10.1186/s12864-015-2328-0>

Index	Abbreviation	Full Name	Training Sample Size	Independent Test Sample Size
1	ASW	African ancestry in Southwest USA	74	13
2	CEU	Utah residents with Northern and Western European ancestry from the CEPH collection	140	25
3	CHB/CHD/JPT	Han Chinese in Beijing, China/ Chinese in Metropolitan Denver, Colorado/Japanese in Tokyo, Japan	305	54
4	GIH	Gujarati Indians in Houston, Texas	86	15
5	LWK	Luhya in Webuye, Kenya	94	16
6	MEX	Mexican ancestry in Los Angeles, California	73	13
7	MKK	Maasai in Kinyawa, Kenya	156	28
8	TSI	Tuscan in Italy	87	15
9	YRI	Yoruban in Ibadan, Nigeria (West Africa)	173	30
Total			1188	209

Huang et al. Ethnicity SNP Exercise.

<https://bmcbioinformatics.biomedcentral.com/articles/10.1186/s12864-015-2328-0>

rs6023406

<https://www.ncbi.nlm.nih.gov/snp>

NIH National Library of Medicine
National Center for Biotechnology Information

dbSNP **BID**: **sr0023406** **Create alert** **Advanced**

Display Settings **Summary**, **Sorted by BID**, **0**

Send to: **E-mail** **Manage filters**

Find related data
Database: **Select** **Find source**

Search results
Items: **3**

Chromosome	Start	End	Genomic coordinate	Gene	Description	Protein	Protein ID	Protein Accession	Protein ID	Protein Accession
20	54823422	54823422	GRCA3B	GRCA3	GRCA3B	GRCA3B	GRCA3B	GRCA3B	GRCA3B	GRCA3B
20	53719901	53719901	GRCA3C	GRCA3	GRCA3C	GRCA3C	GRCA3C	GRCA3C	GRCA3C	GRCA3C
20	53719902	53719902	GRCA3D	GRCA3	GRCA3D	GRCA3D	GRCA3D	GRCA3D	GRCA3D	GRCA3D

Search details
sr0023406[All Fields]

Search **See more**

Search for SNP name, get info about location and variants! Including location. (chr20:54603422)



TEXAS WOMAN'S UNIVERSITY

Index	Abbreviation	Full Name	Training Sample Size	Independent Test Sample Size
1	ASW	African ancestry in Southwest USA	74	13
2	CEU	Utah residents with Northern and Western European ancestry from the CEPH collection	140	25
3	CHB/CHD/JPT	Han Chinese in Beijing, China/ Chinese in Metropolitan Denver, Colorado/ Japanese in Tokyo, Japan	305	54
4	GIH	Gujarati Indians in Houston, Texas	86	15
5	LWK	Luhya in Webuye, Kenya	94	16
6	MEX	Mexican ancestry in Los Angeles, California	73	13
7	MKK	Maasai in Kinyawa, Kenya	156	28
8	TSI	Tuscan in Italy	87	15
9	YRI	Yoruban in Ibadan, Nigeria (West Africa)	173	30
Total			1188	209



**TEXAS WOMAN'S
UNIVERSITY**

Variant	Chr	Position	Gene	A	B	MAF	Color	OR	95% Lower CI	95% Upper CI	P val	Other
rs16891982	5	33987450	SLC45A2	G	C	0.02	Black	5.11	1.79	14.55	0.002	Yes
rs28777	5	33994716	SLC45A2	A	C	0.02	Black	7.05	2.23	22.25	0.001	Yes
rs26722	5	33999627	SLC45A2	G	A	0.02	Black	5.53	1.64	18.68	0.006	Yes
rs12203592	6	341321	IRF4	C	T	0.08	Black	2.35	1.22	4.54	0.011	Yes
rs9378805	6	362727	IRF4	A	C	0.45					0.103	
rs4959270	6	402748	EXOC2	C	A	0.46	Black	0.56	0.35	0.91	0.020	Yes
rs1408799	9	12662097	TYRP1	C	T	0.29					0.097	
rs2733832	9	12694725	TYRP1	T	C	0.40					0.177	
rs683	9	12699305	TYRP1	A	C	0.34					0.099	
rs35264875	11	68602975	TPCN2	A	T	0.23					0.158	
rs3829241	11	68611939	TPCN2	G	A	0.37					0.183	
rs2305498	11	68623490	TPCN2	G	A	0.27					0.230	
rs1011176	11	68690473	TPCN2	T	C	0.36					0.096	
rs1042602	11	88551344	TTR	C	A	0.28					0.255	
rs1393350	11	88650694	TTR	G	A	0.25	Brown	1.70	1.02	2.82	0.041	Yes
rs12821256	12	87852466	KITLG	T	C	0.09					0.052	
rs12896399	14	91843416	SLC24A4	G	T	0.44					0.064	
rs4904868	14	91850754	SLC24A4	C	T	0.46	Blond	0.64	0.43	0.97	0.037	
rs2402130	14	91870956	SLC24A4	A	G	0.16	D-blond	0.62	0.39	0.96	0.033	
rs1800407	15	25903913	OCA2	C	T	0.07	Red	3.23	1.07	9.76	0.038	
rs1800401	15	25933648	OCA2	C	T	0.06					0.105	
rs16950821	15	25957102	OCA2	C	T	0.12					0.170	
rs7174027	15	26002360	OCA2	C	T	0.12					0.146	
rs4778138	15	26009415	OCA2	T	C	0.16	Brown	1.80	1.02	3.19	0.043	Yes
rs4778241	15	26012308	OCA2	G	T	0.18					0.078	
rs7495174	15	26017833	OCA2	T	C	0.05					0.069	
rs12913832	15	26039213	HERC2	C	T	0.22	Black	3.33	1.99	5.57	4.3E-06	Yes
rs7183877	15	26039328	HERC2	C	A	0.07					0.124	
rs11635884	15	26042564	HERC2	T	C	0.01					0.135	
rs916977	15	26186959	HERC2	C	T	0.15	Red	0.34	0.18	0.65	0.001	Yes
rs8039195	15	26189679	HERC2	T	C	0.11	Red	0.30	0.14	0.64	0.002	Yes
MC1R_R	16		MC1R	w	R	0.31	Red	12.64	7.03	22.74	2.5E-17	Yes
MC1R_r	16		MC1R	w	r	0.20	Red	2.50	1.35	4.31	0.003	Yes
rs1805005	16	89985844	MC1R	G	T	0.08	Blond	2.99	1.52	5.86	0.001	Yes
Y152OCH	16	89986122	MC1R	A	C	0.00					0.982	
N29msA	16	89985753	MC1R	-	insA	0.01	Red	53.60	1.29	2221.72	0.036	
rs1805006	16	89985918	MC1R	C	A	0.00					0.476	
rs2228479	16	89985940	MC1R	G	A	0.09	Red	0.43	0.19	0.97	0.043	
rs11547464	16	89986091	MC1R	G	A	0.02	Red	3.35	1.04	10.76	0.042	
rs1805007	16	89986117	MC1R	C	T	0.11	Red	6.69	3.50	12.79	9.3E-09	Yes
rs1110400	16	89986130	MC1R	T	C	0.02					0.314	
rs1805008	16	89986144	MC1R	C	T	0.16	Red	5.69	3.31	9.78	3.2E-10	Yes
rs885479	16	89986154	MC1R	G	A	0.03	Blond	2.90	1.21	6.96	0.017	
rs1805009	16	89986546	MC1R	G	C	0.01	Red	31.85	2.61	388.28	0.007	Yes
rs1015362	20	32202273	ASIP	C	T	0.30	B-red	1.67	1.02	2.75	0.043	
rs6058017	20	32320659	ASIP	A	G	0.13					0.211	
rs2378249	20	32681751	ASIP	A	G	0.18	Red	2.34	1.14	4.82	0.021	Yes

MAF minor allele frequency, Color the most significantly associated color, OR the allelic odds ratio for the minor B allele, shown only if $P < 0.05$, P val the P value adjusted for age and gender, Other if the SNP is also associated with other colors with $P < 0.05$

Branicki et al. Hair-Color SNP Exercise.

<https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3057002/>

Probably a good idea to check those statistically significant variants first.

Branicki, Wojciech et al. "Model-based prediction of human hair color using DNA variants." *Human genetics* vol. 129,4 (2011): 443-54.
doi:10.1007/s00439-010-0939-8

Download the read data for human genome 138 here:

wget https://github.com/mathornton01/twupa_bioinformaticsws_2024/raw/main/data/bamfiles/hg138.zip

Determine the haircolor, ethnicity, and eye color if you have time.



TEXAS WOMAN'S
UNIVERSITY

Spichenok et al. Eye-Color SNP Exercise.

Table 2
Eye color predictor.

Gene	SNP ID	Genotype	Eye color (predicted)
HERC2	rs12913832	G/G	Not brown
HERC2	rs12913832	G/A	Not blue
HERC2	rs12913832	A/A	Not blue
HERC2	rs12913832	A/A	Not blue
IRF4	rs12203592	T/T	
HERC2	rs12913832	G/A	Green
IRF4	rs12203592	T/T	
HERC2	rs12913832	G/G	Green
SLC45A2	rs16891982	C/C	
HERC2	rs12913832	A/A or G/A	Brown
SLC45A2	rs16891982	C/C	
HERC2	rs12913832	A/A or G/A	Brown
OCA2	rs1545397	T/T	
HERC2	rs12913832	A/A or G/A	Brown
MC1R	rs885479	A/A	
HERC2	rs12913832	A/A or G/A	Brown
ASIP	rs6119471	G/G	

<https://www.sciencedirect.com/science/article/pii/S1872497310001717?via%3Dihub>

Spichenok O, Budimlija ZM, Mitchell AA, Jenny A, Kovacevic L, Marjanovic D, Caragine T, Prinz M, Wurmbach E. Prediction of eye and skin color in diverse populations using seven SNPs. Forensic Sci Int Genet. 2011 Nov;5(5):472-8. doi: 10.1016/j.fsigen.2010.10.005. Epub 2010 Nov 2. PMID: 21050833.



Resources on this section:

- IGV video Tutorials.
 - https://www.youtube.com/watch?v=E_G8z_2gTYM
- IGV Manual, and Installation Tutorial
 - <https://software.broadinstitute.org/software/igv/UserGuide>
- Broad Institute Website
 - <https://www.broadinstitute.org/>
- Papers for Exercises
 - Ethnicity - <https://bmcgenomics.biomedcentral.com/articles/10.1186/s12864-015-2328-0>
 - Hair-Color - <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3057002/>
 - Eye-Color - <https://www.sciencedirect.com/science/article/pii/S1872497310001717?via%3Dhub>
- Databases
 - NCBI - <https://www.ncbi.nlm.nih.gov/>
 - SNP - <https://www.ncbi.nlm.nih.gov/snp/>
 - SRA - <https://www.ncbi.nlm.nih.gov/sra/>



BLAST Tutorial

- Go to this website for the MTHFR Genetic Sequence assembly (from GRCh38).
 - https://www.ncbi.nlm.nih.gov/nuccore/NC_00001.11?report=fasta&from=11785723&to=11805964&strand=true
- Highlight the sequence and copy it.



BLAST

- Visit this website:
 - <https://blast.ncbi.nlm.nih.gov/Blast.cgi>

Basic Local Alignment Search Tool

BLAST finds regions of similarity between biological sequences. The program compares nucleotide or protein sequences to sequence databases and calculates the statistical significance.

[Learn more](#)

NEWS

BLAST+ 2.15.0 is here!

We have included two exciting new features in the latest BLAST+ release

Tue, 28 Nov 2023

[More BLAST news...](#)

Web BLAST

Nucleotide BLAST
nucleotide ▶ nucleotide

blastx
translated nucleotide ▶ protein

tblastn
protein ▶ translated nucleotide

Protein BLAST
protein ▶ protein



TEXAS WOMAN'S
UNIVERSITY

BLAST

- Select Nucleotide BLAST, we will be comparing the gene MTHFR from the reference genome to other species.

BLAST® » blastn suite

Home Recent Results Saved Strategies Help

Standard Nucleotide BLAST

blastn blastp blastx tblastn tblastx

BLASTN programs search nucleotide databases using a nucleotide query. more... Reset page Bookmark

Enter Query Sequence

Enter accession number(s), gi(s), or FASTA sequence(s) Query subrange

From
To

Or, upload file No file selected.

Job Title

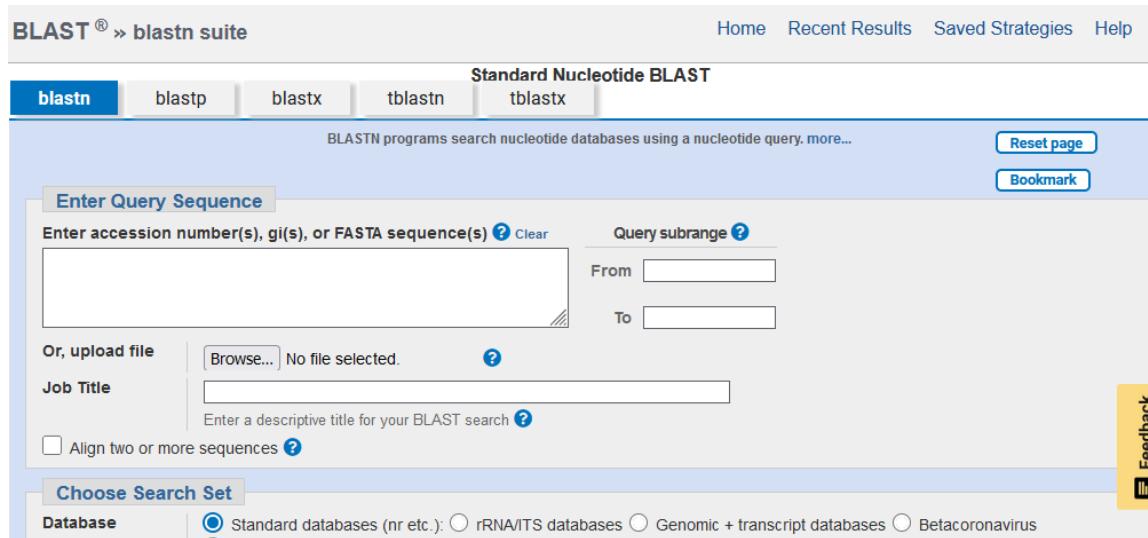
Enter a descriptive title for your BLAST search

Align two or more sequences

Feedback

Choose Search Set

Database Standard databases (nr etc.) rRNA/ITS databases Genomic + transcript databases Betacoronavirus



BLAST

- Paste the sequence you copied into the box at the top of the page, then
- Press BLAST at the bottom of the page.

Descriptions Graphic Summary Alignments Taxonomy

Sequences producing significant alignments Download Select columns Show 100

select all 100 sequences selected GenBank Graphics Distance tree of results MSA View

	Description	Scientific Name	Max Score	Total Score	Query Cover	E value	Per. Ident	Acc. Len	Accession
<input checked="" type="checkbox"/>	Homo sapiens methylenetetrahydrofolate reductase (MTHFR) <small>RefSeq</small> Homo sapiens 37381 39190 100% 0.0 100.00% 27374 NG_013351.1								
<input checked="" type="checkbox"/>	Human DNA sequence from clone RP11-56N19 on chromosome 1 <small>cDNA clone</small> Homo sapiens 37381 46148 100% 0.0 100.00% 102509 AL953897.6								
<input checked="" type="checkbox"/>	Homo sapiens 5,10-methylenetetrahydrofolate reductase (NADPH) <small>(NADPH)</small> Homo sapiens 29132 30353 78% 0.0 99.87% 15835 AY338232.1								
<input checked="" type="checkbox"/>	Rhesus Macaque BAC CH250-220G4 () complete sequence Macaca mulatta 10098 40278 99% 0.0 92.67% 195789 AC191444.6								



BLAST

- Many options are available through the web application interface.
- BLAST is also available as a commandline tool, but this is just really an interface to the online application.



BLAST

- To install the commandline application open a powershell type ‘wsl’ then type “sudo apt-get install ncbi-blast+”

```
(base) micah@MicahsPC:~/TWU/workshop_2024/twupa_bioinformaticsws_2024/simulations$ sudo apt-get install ncbi-blast+
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following packages were automatically installed and are no longer required:
  apport-symptoms distro-info distro-info-data gir1.2-atk-1.0 gir1.2-atspi-2.0 gir1.2-freedesktop
  gir1.2-gdkpixbuf-2.0 gir1.2-glib-2.0 gir1.2-gtk-3.0 gir1.2-harfbuzz-0.0 gir1.2-packagekitglib-1.0
  gir1.2-pango-1.0 iso-codes libappstream4 libblkid-dev libbrotli-dev libcairo-script-interpreter2
  libdattrie-dev libdbus-1-dev libdeflate-dev libdw1 libegl-dev libegl1-mesa-dev libepoxy-dev libffi-dev
  libfontconfig-dev libfontconfig1-dev libfreetype-dev libfreetype6-dev libfribidi-dev
  libgd-pixbuf-xlib-2.0-0 libgirepository-1.0-1 libgl-dev libgles-dev libgles1 libgles2 libglib2.0-bin
  libglvnd-core-dev libglvnd-dev libglx-dev libgraphite2-dev libgstreamer1.0-0 libharfbuzz-gobject0
  libharfbuzz-icu0 libice-dev libjbig-dev libmount-dev libnetplan0 libopengl-dev libpackagekit-glib2-18
  libpangooxft-1.0-0 libpcre16-3 libpcre3-dev libpcre32-3 libpcrecpp0v5 libpixman-1-dev libselinux1-dev
  libsep0-dev libsm-dev libstemmer0d libthai-dev libtiff-dev libtiffxx5 libwayland-bin libwayland-dev
  libxcb-render0-dev libxcb-shm0-dev libcomposite-dev libxcursor-dev libxdamage-dev libxfixes-dev
  libxf86-dev libxi-dev libxinerama-dev libxkbcommon-dev libxmlb2 libxrandr-dev libxrender-dev libxtst-dev
  libyaml-0-2 packagekit packagekit-tools pangol.0-tools python-apt-common run-one uuid-dev
  wayland-protocols zstd
Use 'sudo apt autoremove' to remove them.
The following additional packages will be installed:
  libmbcrypto7 libmbedtls14 libmbedx509-1 ncbi-data
The following NEW packages will be installed:
  libmbcrypto7 libmbedtls14 libmbedx509-1 ncbi-blast+ ncbi-data
0 upgraded, 5 newly installed, 0 to remove and 0 not upgraded.
Need to get 16.1 MB of archives.
After this operation, 72.7 MB of additional disk space will be used.
Do you want to continue? [Y/n] Y
```



BLAST

- BLAST requires a database to search against and a query file, or you can send the search to the remote database.
- Create a new .fasta file called mthfr.fasta by typing touch mthfr.fasta.



BLAST

- Open the file in the vim text editor by typing vim mthfr.fasta.
- Press “i” to enter insert mode
- Then paste the sequence
- PRESS escape and then type “:wq” and presss enter.



BLAST

- Now type:
“blastn -db nt -query mthfr.fasta -o
ut results.out –remote”

```
blastn: argument error: too many positional arguments (-o), the offending value: -o
(base) micah@MicahsPC:~/TWU/workshop_2024/twupa_bioinformaticsws_2024/data/bamfiles$ blastn -db nt -query mthfr.fasta -o
ut results.out -remote
```



BLAST

- Results are written to the file results.out in the multiple alignment format,

>XM_003936021.3 PREDICTED: Saimiri boliviensis boliviensis methylenetetrahydrofolate
reductase (MTHFR), transcript variant X1, mRNA
Length=7201

Score = 1236 bits (669), Expect = 0.0
Identities = 1289/1572 (82%), Gaps = 107/1572 (7%)
Strand=Plus/Plus

Query	16875	GAGCTGGGAAGCTGTACTCAGAGCAGAGCAAATGAGGGAGGGGGCGCTCAGGACCCAGGC	16934
Sbjct	4423	GAG-TGTGAAGCTGTGCTCAGAGTGGAGCAAAGGGAGGGAGGGGGCGCTCGGGACCAAGGC	4481



BLAST

- To see all the sequences that it aligned to type “grep “^>” results.out”

```
(base) micah@MicahsPC:~/TWU/workshop_2024/twupa_bioinformaticsws_2024/data/bamfiles$ grep '^>' results.out  
>NG_013351.1 Homo sapiens methylenetetrahydrofolate reductase (MTHFR), RefSeqGene  
>AL953897.6 Human DNA sequence from clone RP11-56N19 on chromosome 1, complete  
>AY338232.1 Homo sapiens 5,10-methylenetetrahydrofolate reductase (NADPH)  
>AC191444.6 Rhesus Macaque BAC CH250-220G4 () complete sequence  
>NM_005957.5 Homo sapiens methylenetetrahydrofolate reductase (MTHFR), transcript  
>NM_001410750.1 Homo sapiens methylenetetrahydrofolate reductase (MTHFR), transcript  
>XM_005263463.5 PREDICTED: Homo sapiens methylenetetrahydrofolate reductase (MTHFR),  
>XM_047421178.1 PREDICTED: Homo sapiens methylenetetrahydrofolate reductase (MTHFR),  
>XM_005263462.5 PREDICTED: Homo sapiens methylenetetrahydrofolate reductase (MTHFR),  
>NM_001330358.2 Homo sapiens methylenetetrahydrofolate reductase (MTHFR), transcript  
>XM_054336707.1 PREDICTED: Homo sapiens methylenetetrahydrofolate reductase (MTHFR),  
>XM_054336702.1 PREDICTED: Homo sapiens methylenetetrahydrofolate reductase (MTHFR),  
>XM_047421181.1 PREDICTED: Homo sapiens methylenetetrahydrofolate reductase (MTHFR),  
>XM_047421180.1 PREDICTED: Homo sapiens methylenetetrahydrofolate reductase (MTHFR),  
>XM_017001328.3 PREDICTED: Homo sapiens methylenetetrahydrofolate reductase (MTHFR),  
>XM_047421179.1 PREDICTED: Homo sapiens methylenetetrahydrofolate reductase (MTHFR),  
>XM_047421174.1 PREDICTED: Homo sapiens methylenetetrahydrofolate reductase (MTHFR),  
>XM_011541496.4 PREDICTED: Homo sapiens methylenetetrahydrofolate reductase (MTHFR),  
>AF398933.1 Homo sapiens methylenetetrahydrofolate reductase (MTHFR) gene,
```



BLAST

- Identify possible sources of the following sequence using blastn with the remote nt database:

```
ATATATGGTATTGCCCTGGTGATTTGCTGCTAGAGACCTCATTTGTGACAAAAGTTAACGGCCTTC TGTTTGCCACCTTGCTCACAGATGAAATGATTGCTCAATACACTTGTGACTGTAGCGGTACAATC  
ACTTCTGGTGGACCTTGGTGCGAGGTGCTGCATTACAATACCATTGCTATGCAAATGGCTTATAGGT TTATGGTATTGGAGTTACACAGAATGTTCTCTATGAGAACCAAAAATTGATGCCAACCAATTAAAG  
TGCTATTGGCAAAATTCAAGACTCATTCTCCACAGCAAGTGCACTTGGAAAACCTCAAGATGGCTC ACCATAATGACACAGCTTAAACAGCCTGGTTAAACAGCCTTAAACAGCCTTAAACAGCCTTAAACAGCCT  
TTTAAATGATATCTTACGCTCTGACAAAGTTGAGGCTGAAGTGCACAGG CAGACTCAAAGTTGAGCAGACATATGTGACTCAACAATTAAAGAGCTGCAGAAATCAGAGCTCTGCT  
AATCTTGGCTGACTAAAATGTCAGAGGTGACTTGGAAACATCAAAAGAGTTGATTTTGTGAAAGG GCTATCATCTTATGCTCTCCCTAGTCAGCACCTCATGGTAGTGTAGCTCTTGATGTGACTTATGTCCC  
TGACACAAAAAAAGACTTCAACACTCTGCTGCCATTGTGATGAGGAAAGCACAACACTTCTCTGTA GGTTGCTTTCAATGGCACACACTGGTTGACACAAAGGAATTITATGAACCCAAAATCTTAA  
CTACAGACACACATTGTGTAACTGTGATGGTAATAGGAATTGTCAACACACAGTTATGA TCTTTGCAACCTGAAATTAGATTCTCAAGGGAGGAGTTAGAAAAATTITTAAGAACTACACATCACCA  
GATTTGATTAGGTGACATCTGGCTTAAACATTGCTCAGTTGACAAACATTGACCGCC TCAATGAGGTTGCCAAGAAATTAAATGAACTCTCATCGATCTCAAGAAACTTGGAAAGTATGAGCAGTA  
TATAAAATGGCCTGGTACATTGGCTAGGTTTATAGCTGGCTGATTGCCATAGTAATGGTACAATT ATGCTTGTGATGACCAGTTGCTGTAGTTGCTCAAGGGCTGTTCTGGATCCCTGCTGCAAAT  
TTGATGAAGACGACTCTGAGGCCAGTGTCAAGGGAGTCAAAATTACATACAAACGCACTTATGGAT TTGTTTATGAGAACTTCAACATTGGAACTGTAACTTGAAGGCAAGGGAAATCAAGGATGCTACTCCT  
CAGATTGTTGCCACTGCAACGATACCAAGCCTCACTCCCTGGATGCCATTGGTGG CGTGCACITCTGCTGTTTCAAGGCCTCCAAAATCATAACTCTCAAAAGAGATGGCAACTAGCA  
CTCTCCAAGGGTGTCACTTGTGTTGCAACTTGTGCTGTTGTAACAGTTACTCACCT
```



Thanks for your attention!



Questions/Comments/Suggestions?



Backup Slides



TEXAS WOMAN'S
UNIVERSITY

Markov Boundary

- Markov Boundary:
 - Let X be a random variable that can take on values in the support $\mathcal{S}_X = \{x \mid x \in \mathbb{R} \cup x \geq 0\}$
 - In other words, X is a non-negative random variable.
 - Furthermore let $a > 0$



Markov Boundary

- X non-negative, and $a > 0$ we have:

$$P(X \geq a) \leq \frac{\mathbb{E}(X)}{a}$$



Markov Boundary

- This boundary is measure theoretic and is sometimes known as Chebyshev's Inequality.



Markov Boundary Probability Theory

Derivation

- To derive the Markov boundary consider the following argument.

$$\mathbb{E}(X) = \int_0^{\infty} x f(x) dx$$

- For some value $a \geq 0$ we then have

$$\mathbb{E}(X) = \int_0^a x f(x) dx + \int_a^{\infty} x f(x) dx$$



Markov Boundary Probability Theory

Derivation

- Clearly

$$\int_0^a x f(x)dx + \int_a^\infty x f(x)dx \geq \int_a^\infty x f(x)dx$$

- Since a is lowest value in integral we know

$$\int_a^\infty x f(x)dx \geq \int_a^\infty a f(x)dx$$



Markov Boundary Probability Theory

Derivation

- Therefore

$$\mathbb{E}(X) \geq \int_a^{\infty} a f(x) dx$$

$$\frac{\mathbb{E}(X)}{a} \geq \int_a^{\infty} f(x) dx = \Pr(X \geq a)$$



Markov Boundary Probability

- Use the Markov boundary probability to provide a boundary that the binomial distribution with $n = 10000$, and $p = 0.25$ will produce a value that is beyond 2550.



Markov Boundary

- Recall the expectation for the binomial distribution is $n \cdot p = 2500$ therefore we can write down a boundary for the probability that $X \geq 2550$ as:
- $\frac{2500}{2550} = 0.98039215686 \geq \Pr(X \geq 2250)$



Markov Boundary

- This is clearly not a very tight boundary.
- Can we use the Chernoff Bound to do better?



Chernoff Bound

- The Chernoff Bound is based on the moment generating function, and therefore incorporates information for all of the moments of a given distribution.



Chernoff Bound

- What are the moments of a distribution?
$$\mu'_1 = \mathbb{E}(X), \mu'_2 = \mathbb{E}(X^2), \mu'_3 = \mathbb{E}(X^3), \dots$$
 - First, moment related to center of distribution
 - Second, related to dispersion around center
 - Third, dispersion of dispersion, or skewness
 - Fourth, skewness of dispersion, flatness or kurtosis



Chernoff Bound

- Would be nice if one function characterized all the moments of a distribution – The moment generating function does this.



Chernoff Bound

- We write the moment generating function as a function of a t (supposing that the function of t is defined in some real neighborhood around 0).
- The moment generating function is defined as the expectation of the exponential function of t and valuation x



Chernoff Bound

- We can write the moment generation function of the random variable X

$$M_X(t) = \mathbb{E}(e^{tx})$$

- What is this expectation for the binomial distribution?



Chernoff Bound

$$\begin{aligned}\mathbb{E}(e^{tx}) &= M_X(t) = \sum_{x=1}^N e^{tx} \binom{N}{x} p^x (1-p)^{N-x} \\ &= \sum_{x=1}^N \binom{N}{x} (pe^t)^x (1-p)^{N-x}\end{aligned}$$



Chernoff Bound

- Recall the Binomial Coefficients Theorem

$$(x + y)^n = \sum_{k=0}^n \binom{n}{k} x^{n-k} y^k = \sum_{k=0}^n \binom{n}{k} x^k y^{n-k}.$$

Pascal's Triangle – Blaise Pascal
Cambrdige University



Chernoff Bound

$$(x + y)^n = \sum_{k=0}^n \binom{n}{k} x^{n-k} y^k = \sum_{k=0}^n \binom{n}{k} x^k y^{n-k}.$$

$$\sum_{x=1}^N \binom{N}{x} (pe^t)^x (1-p)^{N-x} = (pe^t + (1-p))^N$$

$$M_X(t) = (pe^t + (1-p))^N$$



TEXAS WOMAN'S
UNIVERSITY

Aside – Moment Generating Functions

- How does this generate moments?
 - Well,

$$M'_X(0) = \mu'_1$$

$$M''_X(0) = \mu'_2$$

$$M_X^{(3)}(0) = \mu'_3$$



Moment Generating Functions

- [Proof]

$$\begin{aligned}\frac{d}{dt} M_X(t) &= \frac{d}{dt} \int_{-\infty}^{\infty} e^{tx} f_X(x) dx \\ &= \int_{-\infty}^{\infty} \frac{d}{dt} e^{tx} f_X(x) dx\end{aligned}$$

How?



Moment Generating Functions

$$\frac{d}{dt} \int_{-\infty}^{\infty} e^{tx} f_X(x) dx = \int_{-\infty}^{\infty} \frac{d}{dt} e^{tx} f_X(x) dx$$

$$\int_{-\infty}^{\infty} x e^{tx} f_X(x) dx = \mathbb{E}(X e^{tX})$$

$$M'_X(0) = \mathbb{E}(X e^{0 \cdot X}) = \mathbb{E}(X)$$

$$\begin{aligned} & \frac{d}{dx} \left(\int_{a(x)}^{b(x)} f(x, t) dt \right) \\ &= f(x, b(x)) \cdot \frac{d}{dx} b(x) - f(x, a(x)) \cdot \frac{d}{dx} a(x) + \int_{a(x)}^{b(x)} \frac{\partial}{\partial x} f(x, t) dt \end{aligned}$$

Lebesgue's Dominated
Convergence Theorem – Beyond
our scope (Measure Theory)

- Repeating the differentiation with respect to t gives $\mathbb{E}(X^2 e^{tX})$, and similarly differentiating n times gives $\mathbb{E}(X^n e^{tX})$ and therefore

$$M_X^n(0) = E(X^n) = \mu'_n$$



Chernoff Bound

$$\frac{\mathbb{E}(X)}{a} \geq \Pr(X \geq a)$$

- Note

$$P(X \geq a) = P(e^{tX} \geq e^{ta})$$

- Applying the Markov Inequality we have:

$$P(e^{tX} \geq e^{ta}) \leq \frac{\mathbb{E}(e^{tX})}{e^{ta}} = M_X(t)e^{-ta}$$



Chernoff Bound

$$P(e^{tX} \geq e^{ta}) \leq M_X(t)e^{-ta}$$

- This bound holds for all positive t so let us use the value of t which minimizes $M_X(t)e^{-ta}$. That minimum value of $M_X(t)e^{-ta}$ is called the infimum with respect to $t > 0$.

$$P(X \geq a) \leq \inf_{t>0} M_X(t)e^{-ta} \quad (\text{Chernoff Bound})$$



Chernoff Bound

- For the Binomial Distribution then we have:

$$P(X \geq a) \leq \inf_{t>0} M_X(t)e^{-ta} = \inf_{t>0} (pe^t + (1-p))^N e^{-ta}$$

To derive a useful bound however, we must consider an alternative form.



Chernoff Bound for Binomial

- One useful property of the Moment generating function is that
 - THEOREM: the Moment generating function of the sum of a set of random variables is the product of their moment generating functions.
 - Proof is trivial



Chernoff Bound for Binomial

- We can consider the binomial random variable X to be the constituent sum of random variables X_1, X_2, \dots, X_N ;

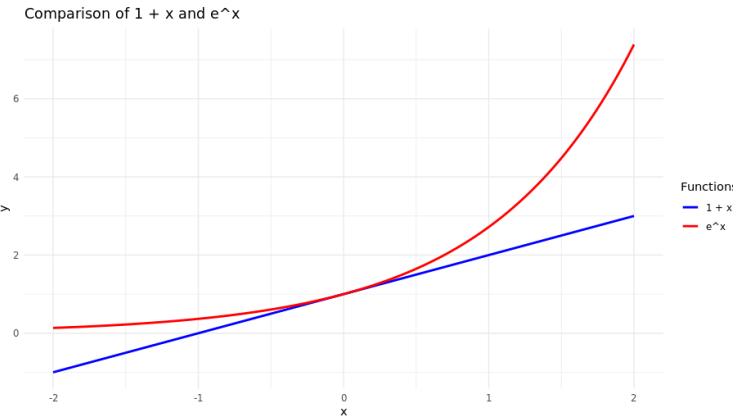
$$X = \sum_{i=1}^N X_i \Rightarrow E(e^{Xt}) = E\left(e^{t \sum_{i=1}^N X_i}\right)$$



Chernoff Bound for Binomial

$$\begin{aligned} E\left(e^{t \sum_{i=1}^N X_i}\right) &= E\left(\prod_{i=1}^N e^{tX_i}\right) = \prod_{i=1}^N E(e^{tX_i}) \\ &= (M_{X_i}(t))^N = (pe^t + 1 \cdot (1 - p))^N \end{aligned}$$

$$(1 + p(e^t - 1)) \leq e^{p(e^t - 1)}$$



Chernoff Bound for Binomial

- Therefore, for $a > 0$ we have:

$$\Pr(X \geq a) \leq \inf_{t>0} \frac{(e^{p(e^t - 1)})^n}{e^{ta}}$$

- Suppose we would like a bound on
 $\mu(1 + \delta)$



Chernoff Bound for Binomial

$$\Pr(X \geq \mu(1 + \delta)) \leq \inf_{t>0} \frac{(e^{p(e^t - 1)})^n}{e^{t\mu(1+\delta)}}$$

- Now we must find t which minimizes expression, and determine what minimum is.



Chernoff Bound for Binomial

$$\begin{aligned} \inf_{t>0} \left(\frac{\left(e^{p(e^t - 1)} \right)^n}{e^{t\mu(1+\delta)}} \right) &= \\ \inf_{t>0} \left(e^{np(e^t - 1) - t\mu(1+\delta)} \right) &= \\ e^{\inf_{t>0} (np(e^t - 1) - t\mu(1+\delta))} \end{aligned}$$



Chernoff Bound for Binomial

$$\begin{aligned}\frac{\partial}{\partial t} (np(e^t - 1) - t\mu(1 + \delta)) &= \\ npe^t - \mu - \delta\mu &\Rightarrow \\ npe^{t^*} - \mu - \delta\mu &= 0 \Rightarrow \\ npe^{t^*} &= \mu(1 + \delta) \Rightarrow \\ t^* &= \ln(1 + \delta)\end{aligned}$$



Chernoff Bound for Binomial

$$\begin{aligned} t^* &= \ln(1 + \delta) \Rightarrow \\ \inf_{t>0} \left(\frac{(e^{p(e^t - 1)})^n}{e^{t\mu(1+\delta)}} \right) &= \frac{e^{np(\delta)}}{(1 + \delta)^{\mu(1+\delta)}} \\ \Rightarrow \Pr(X \geq (1 + \delta)\mu) &\leq \left(\frac{e^{(\delta)}}{(1 + \delta)^{(1+\delta)}} \right)^\mu \end{aligned}$$



Chernoff Bound for Binomial

- Recall we want a bound for the probability of observing a value greater than 2550 when $n = 10000$ and $p = 0.25$,
- $(1 + \delta)np = 2550 \Rightarrow \delta = \frac{2550}{2500} - 1 = 0.02$
- $\Pr(X \geq 2550) \leq \left(\frac{e^{(0.2)}}{(1.02)^{(1.02)}} \right)^{2500}$

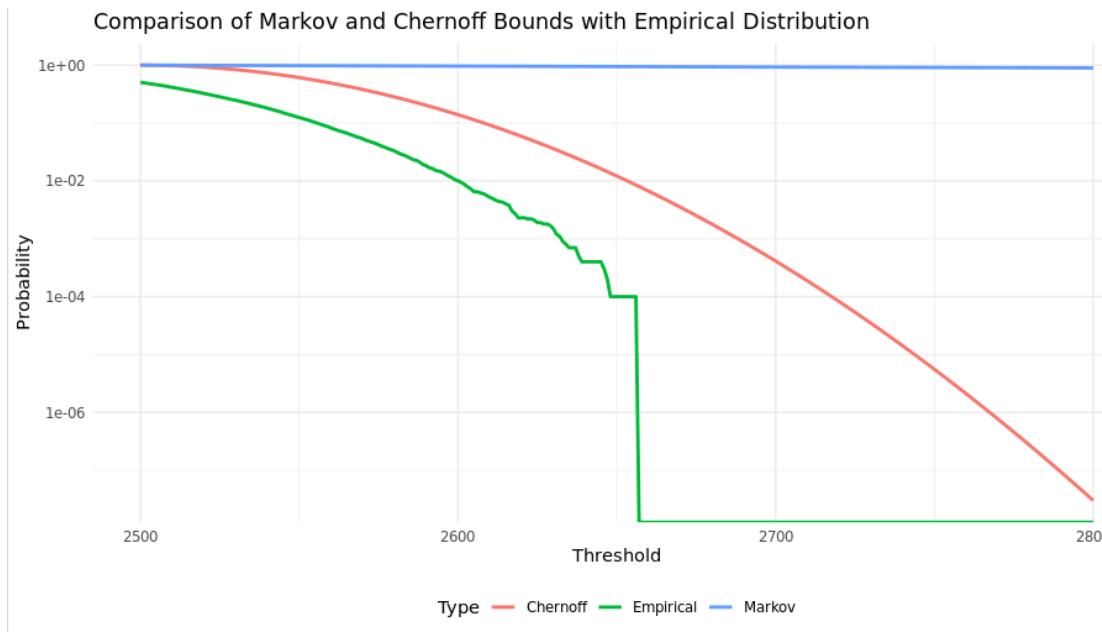


Chernoff Bound for Binomial

- Hence the Chernoff bound is very close to 1,
- This is not as useful at this distance as the Markov bound.



Chernoff and Markov Bounds





**TEXAS WOMAN'S
UNIVERSITY**

**Project ACCESS Bioinformatics Workshop
Summer 2024 – July 29, 30; 2024
Micah Andrew Thornton ©**

Who am I?

- B.Sc. - Statistical Science & Computer Engineering (SMU, 2017)
- M.S – Computer Engineering (SMU, 2017)
- Ph. D. - Biostatistics (SMU/UTSW, 2021)
- Post-Doctoral Research Certification (UTSW 2024) –
 - Daehwan Kim Lab (UTSW, 2021-2023)
 - Sequence Alignment and Statistical Models
 - Lee Kraus Lab (UTSW, 2023)
 - Statistical Models for RPol-II Propagation



What is this Workshop?

- Hosted by Texas Woman's University's Project ACCESS
- Held over two days (July 29, 30; 2024)
- Using the tools of Bioinformatics:
 - Gives introduction and theory background.
 - Provides hands-on tool use experience.



Day 1 (July 29, 2024)

- Focus on the basics:
 - Introduction:
 - What is/is not Bioinformatics?
 - Assembly/Human Reference Genome
 - Tools of Bioinformatics:
 - Standard File Formats
 - Genome Browsers
 - Multiple Sequence Alignment - BLAST



Day 2 (July 30, 2024)

- Command-line Tools:
 - Review of Day 1
 - Linux – History and Basic Utilities
 - Sequence Aligners – HISAT2
 - SAMTOOLS and BCFTOOLS
- R Tools:
 - DESeq2



Day 2



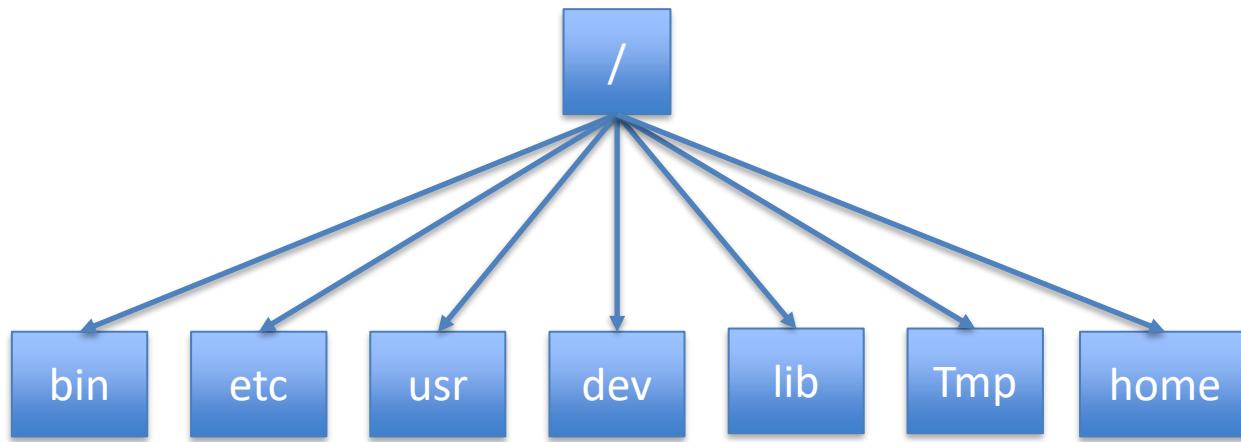
TEXAS WOMAN'S
UNIVERSITY

Unix File Systems

- Files are a Collection of Data (binary or ASCII text)
- Files are Organized in “directories”
 - similar to folders in a file cabinet
 - file cabinets are similar to file systems
 - File systems may be disks, tapes, etc.
- File Systems are Hierarchical
- A Directory may contain other directories
 - These are called “subdirectories”
- The “/” character is used to access subdirectories
- The Main “root” directory always has the name “/”



Unix File System Hierarchy



Unix File System Commands

- man – displays manual pages
- ls – lists the names of files in the current directory
- file – tells you the type of file
- cat – sends contents of file to stdout (terminal)
- grep – searches for a character string in a text file



Accessing windows files from WSL

- To open a Windows subsystem for linux shell (bash) open the powershell and type “wsl”

```
micah@MicahsPC: /mnt/c/Users | + | 
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\Micah> wsl
micah@MicahsPC:/mnt/c/Users/Micah$ |
```



Opens Mounted C Drive

- pwd – tells you the current working directory

```
micah@MicahsPC: /mnt/c/Users/Micah$ pwd
/mnt/c/Users/Micah
micah@MicahsPC:/mnt/c/Users/Micah$ |
```

- cd – Changes directory

```
micah@MicahsPC:/mnt/c/Users/Micah$ cd /home/micah
micah@MicahsPC:~$ pwd
/home/micah
micah@MicahsPC:~$ |
```



Creating a new directory

- ls – lists content of current directory

```
micah@MicahsPC: ~      X + ▾
micah@MicahsPC:~$ ls
R power_analysis_RCT_2024_dua.Rmd power_analysis_RCT_2024_dua.html
micah@MicahsPC:~$ |
```

- mkdir – creates a new directory

```
micah@MicahsPC:~$ mkdir TWU
micah@MicahsPC:~$ ls
R TWU power_analysis_RCT_2024_dua.Rmd power_analysis_RCT_2024_dua.html
micah@MicahsPC:~$ |
```



Man gives info on command

- man – stands for manual
- The UNIX manual is available online
- Each command has flags, what are the flags for ls?
- Enter “man ls” at the linux prompt.



LS(1)

User Commands

LS(1)

NAME

ls - list directory contents

SYNOPSIS

ls [OPTION]... [FILE]...

DESCRIPTION

List information about the FILEs (the current directory by default). Sort entries alphabetically if none of **-cftuvSUX** nor **--sort** is specified.

Mandatory arguments to long options are mandatory for short options too.

-a, --all

do not ignore entries starting with .

-A, --almost-all

do not list implied . and ..

--author

with **-l**, print the author of each file

-b, --escape

print C-style escapes for nongraphic characters

Press q to quit, ls -lah

- To leave the manual entry for a command type “q”.
- Type ls –lah at command prompt (-l, -a, -h are flags)
 - -l means return info in list format (with extra info about files)
 - -a means list all files (even hidden files that start with a .)
 - -h means list sizes in human readable format



```
micah@MicahsPC:~$ ls -lah
total 936K
drwxr-x--- 10 micah micah 4.0K Jul 16 12:57 .
drwxr-xr-x  4 root  root  4.0K Jul  9 12:59 ..
-rw-r--r--  1 micah micah 189 Jul  9 19:26 .Rhistory
-rw-------  1 micah micah 3.8K Jul 16 12:52 .bash_history
-rw-r--r--  1 micah micah 220 Jun 10 13:28 .bash_logout
-rw-r--r--  1 micah micah 3.7K Jun 10 13:28 .bashrc
drwxr-xr-x  7 micah micah 4.0K Jul  9 19:18 .cache
drwx-----  3 micah micah 4.0K Jun 10 15:21 .config
-rw-------  1 micah micah  20 Jul 16 12:57 .lessht
drwxr-xr-x  5 micah micah 4.0K Jul  9 12:55 .local
-rw-r--r--  1 micah micah    0 Jul 16 12:33 .motd_shown
drwx-----  3 micah micah 4.0K Jun 10 15:21 .pki
-rw-r--r--  1 micah micah 807 Jun 10 13:28 .profile
-rw-------  1 micah micah  12 Jul  9 12:54 .python_history
drwxr-xr-x  3 micah micah 4.0K Jul  9 19:26 .r
-rw-r--r--  1 micah micah    0 Jun 10 13:36 .sudo_as_admin_successful
drwxr-xr-x  5 micah micah 4.0K Jul  9 19:18 .vscode-server
drwxr-xr-x  3 micah micah 4.0K Jun 10 15:21 R
drwxr-xr-x  2 micah micah 4.0K Jul 16 12:53 TWU
-rw-r--r--  1 micah micah 5.9K Jul 10 12:14 power_analysis_RCT_2024_dua.Rmd
-rw-r--r--  1 micah micah 860K Jul 10 12:14 power_analysis_RCT_2024_dua.html
```



Creating a file with touch

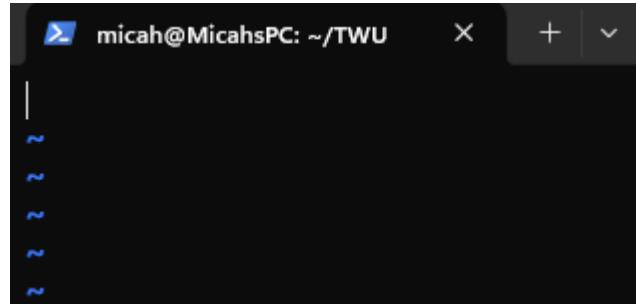
- Type “cd TWU” to navigate to the TWU folder we created.
- Type “touch workshop.txt” to create file workshop.txt

```
micah@MicahsPC:~/TWU$ touch workshop.txt
micah@MicahsPC:~/TWU$ ls
workshop.txt
micah@MicahsPC:~/TWU$ |
```



Editing Files with vim

- Type “vim workshop.txt” to edit workshop.txt



- Press ‘i’ to enter insert mode.



Enter information

- Type your name, the date, “Introduction to Bioinformatics Workshop”, and “Day 2” on separate lines

Micah Thornton
07-30-2024
Introduction to Bioinformatics Workshop
Day 2
~
~
~
-- INSERT --



TEXAS WOMAN'S
UNIVERSITY

Exit insert mode

- To return to command mode press “esc”
- To save what you’ve written and quit vim type “:wq” and hit enter

```
Micah Thornton
07-30-2024
Introduction to Bioinformatics Workshop
Day 2
~
~
~
:wq|
```



Print contents of file

- Type “Cat workshop.txt” to display the contents of the file

```
micah@MicahsPC:~/TWU$ cat workshop.txt
Micah Thornton
07-30-2024
Introduction to Bioinformatics Workshop
Day 2
micah@MicahsPC:~/TWU$ |
```



Print contents of file - head

- To print just first and second lines type
“head –n 2 workshop.txt”

```
micah@MicahsPC:~/TWU$ head -n 2 workshop.txt
Micah Thornton
07-30-2024
micah@MicahsPC:~/TWU$ |
```



Print contents of file - tail

- To print only the last two lines use the “tail” command.

```
micah@MicahsPC:~/TWU$ tail -n 2 workshop.txt
Introduction to Bioinformatics Workshop
Day 2
micah@MicahsPC:~/TWU$
```



Case sensitivity

- Note that everything we have seen so far is case-sensitive.
- That means that you won't be able to access the content of a file named FILE.txt by typing cat file.txt, you must use capital letters.



Case sensitivity

```
(base) micah@MicahsPC:~/TWU$ cat WORKSHOP.txt
cat: WORKSHOP.txt: No such file or directory
(base) micah@MicahsPC:~/TWU$ cat workshop.txt
Micah Thornton
07-30-2024
Introduction to Bioinformatics Workshop
Day 2
(base) micah@MicahsPC:~/TWU$
```



UNIX File System Commands

- `pwd` – gives the path of the current directory
- `mkdir` – creates a subdirectory
- `cd` – changes the current working directory
- `cp` – used to copy a file from one location to another
- `mv` – used to rename a file



Useful shorthand directories

- ‘.’ to reference the current directory and
- ‘..’ to reference the directory one above the current directory.
- ‘~’ references the home directory, and
- ‘-’ references the last directory that where you were navigating.
- ‘/’ references the root of the filesystem



Unix Shells

- There are many UNIX shells available – they all have the same base capabilities, just slightly different syntaxes.
 - We will use bash (the Bourne-Again Shell)
- Your user entry in /etc/passwd file specifies the shell that runs on startup.

```
micah:x:1000:1000:,:/home/micah:/bin/bash
```



Unix Variables

- Variables are defined in the shell via:
 - varname=value (no spaces)

```
(base) micah@MicahsPC:~/TwU$ sequence_tech=RNA
(base) micah@MicahsPC:~/TwU$ echo $sequence_tech
RNA
(base) micah@MicahsPC:~/TwU$ |
```

- “\$” used in front of the variable name indicates that the variable content should be used here.



Unix Default Variables

- Two important variables are HOME and PATH.
- HOME is where your home directory is

```
(base) micah@MicahsPC:~/TWU$ echo $HOME  
/home/micah  
(base) micah@MicahsPC:~/TWU$ echo $PATH  
/home/micah/.local/bin:/home/micah/anaconda3/bin:/home/micah/anaconda3/condabin:/usr/local/sbin:/usr/local/bin:/usr/sbin  
:/usr/bin:/sbin:/bin:/usr/games:/usr/local/games:/usr/lib/wsl/lib:/mnt/c/Program Files/Oculus/Support/oculus-runtime:/mn  
t/c/Program Files/Common Files/Oracle/Java/javapath:/mnt/c/WINDOWS/system32:/mnt/c/WINDOWS:/mnt/c/WINDOWS/System32/Wbem:  
/mnt/c/WINDOWS/System32/WindowsPowerShell/v1.0:/mnt/c/WINDOWS/System32/OpenSSH:/mnt/c/Program Files/Git/cmd:/mnt/c/Progr  
am Files/dotnet:/mnt/c/WINDOWS/system32:/mnt/c/WINDOWS:/mnt/c/WINDOWS/System32/Wbem:/mnt/c/WINDOWS/System32/WindowsPower  
Shell/v1.0:/mnt/c/WINDOWS/System32/OpenSSH:/mnt/c/Users/Micah/AppData/Local/Programs/Python/Python310/Scripts:/mnt/c/Use  
rs/Micah/AppData/Local/Programs/Python/Python310:/mnt/c/Users/Micah/AppData/Local/Microsoft/WindowsApps:/mnt/c/Users/Mic  
ah/AppData/Local/Programs/Microsoft VS Code/bin:/mnt/c/msys64\ucrt64/bin:/mnt/c/Users/Micah/AppData/Local/Pandoc:/mnt/c/  
Users/Micah/AppData/Local/Muse Hub/lib:/snap/bin  
(base) micah@MicahsPC:~/TWU$ |
```



Unix Default Variables

- PATH contains a list of everywhere to search for executables by default, separated by “：“. To add a new path to the Path variable use:
- Export \$PATH=\$PATH:<dir name>
- With dirname replaced with the new directory to search for executables in.
 - To see all variables type “set” to see only exported variables “export” – Export is like making a variable global.



Alias command

The alias command is a way of having one string substituted for another:

```
alias name=value
```

An example:

```
alias ls="ls -aCF"
```

This is nice if you always want to execute the ‘ls’ command with the flags ‘-aCF’ (these flags cause the ls command to display more information about the list files).

Commonly used aliases are typically placed in the user’s “*~/.profile*” file so that they are executed when a *ksh* shell starts.

To see a list of all aliases, just type “alias” with no arguments.



Alias Command

```
(base) micah@MicahsPC:~/TWU$ alias my_list="ls -lah"
(base) micah@MicahsPC:~/TWU$ my_list
total 28K
drwxr-xr-x  5 micah micah 4.0K Jul 18 15:24 .
drwxr-x--- 19 micah micah 4.0K Jul 20 17:54 ..
drwxr-xr-x  2 micah micah 4.0K Jul 16 14:54 assembly
-rw-r--r--  1 micah micah   38 Jul 16 14:12 example.fasta_1.fa
drwxr-xr-x  3 micah micah 4.0K Jul 18 15:33 tutorial_example
-rw-r--r--  1 micah micah   72 Jul 16 13:09 workshop.txt
drwxr-xr-x  4 micah micah 4.0K Jul 19 15:23 workshop_2024
```



My .profile file

- You have a .profile file that is loaded at startup.
- You can edit it with vim by typing “vim ~/.profile”, you can enter insert mode with “i” and add your alias/export commands here.



Sort command

- Type sort workshop.txt

```
(base) micah@MicahsPC:~/TWU$ sort workshop.txt
07-30-2024
Day 2
Introduction to Bioinformatics Workshop
Micah Thornton
```

- We see the sorted output is displayed in the terminal window.



Input/Output redirection

- Instead of redirecting to the terminal window by default you can redirect into a file with the “>” or “>>” symbol.
- To write a new file use the “>” symbol, whereas to add to an existing file use the “>>” symbol.



Input/Output redirection

```
(base) micah@MicahsPC:~/TWU/redirection$ pwd  
/home/micah/TWU/redirection  
(base) micah@MicahsPC:~/TWU/redirection$ ls  
(base) micah@MicahsPC:~/TWU/redirection$ echo "Hello World!" > hw.txt  
(base) micah@MicahsPC:~/TWU/redirection$ ls  
hw.txt  
(base) micah@MicahsPC:~/TWU/redirection$ cat hw.txt  
Hello World!  
(base) micah@MicahsPC:~/TWU/redirection$ |
```



Input/Output redirection

```
(base) micah@MicahsPC:~/TWU/redirection$ echo "This is Bioinformatics!" >> hw.txt
(base) micah@MicahsPC:~/TWU/redirection$ cat hw.txt
Hello World!
This is Bioinformatics!
(base) micah@MicahsPC:~/TWU/redirection$ echo "This is Bioinformatics!" > hw.txt
(base) micah@MicahsPC:~/TWU/redirection$ cat hw.txt
This is Bioinformatics!
(base) micah@MicahsPC:~/TWU/redirection$ |
```



Input/Output redirection

- You can apply multiple functions in parallel by using the pipe symbol “|” to pipe the output of one program into the output of another.



Connecting Programs via Pipes

- Suppose we want to count the number of files in a directory.
- We could list the files one per line using the ls –l program, then pipe this into word count to determine the number of lines.



Connecting Programs via Pipes

```
(base) micah@MicahsPC:~/TWU$ ls -l | wc  
    7      56     351
```

- We can see that there are a total of 7 lines in the output of ls –l
- One line (the first) does not represent a file, so we know there are $7-1=6$ files Present in the file.

```
(base) micah@MicahsPC:~/TWU$ ls -l  
total 24  
drwxr-xr-x 2 micah micah 4096 Jul 16 14:54 assembly  
-rw-r--r-- 1 micah micah    38 Jul 16 14:12 example.fasta_1.fa  
drwxr-xr-x 2 micah micah 4096 Jul 22 16:34 redirection  
drwxr-xr-x 3 micah micah 4096 Jul 18 15:33 tutorial_example  
-rw-r--r-- 1 micah micah    72 Jul 16 13:09 workshop.txt  
drwxr-xr-x 4 micah micah 4096 Jul 19 15:23 workshop_2024
```



Connecting Programs via Pipes

- Suppose we wanted to quickly determine how many subsequences are listed in a FASTA file.
- We could search for lines that begin with ">" by using "grep ">" ex.fasta" and pipe the output into word-count by ending the line with " | wc"



Connecting Programs via Pipes

```
(base) micah@MicahsPC:~/TWU$ grep ">" workshop_2024/twupa_bioinformaticsws_2024/data/Homo_sapiens.GRCh38.dna.primary_assembly.fa | wc  
194      776    12524
```

Here we can see there were 194 lines, so there are 194 sequence labels in the file, and therefore we can assume that this version of the human reference genome has 194 contiguous sequences (some of these are what is known as decoy, those will be discussed later.)



Viewing running processes

- Sometimes it can take a while for a process to complete, and we may want to check whether it is still running.
- We can do this with the “ps” command.



Ps command

```
(base) micah@MicahsPC:~/TWU$ sort workshop_2024/twupa_bioinformaticsws_2024/data/Homo_sapiens.GRCh38.dna.primary_assembly.fa &
[1] 2558
(base) micah@MicahsPC:~/TWU$ ps
 PID TTY      TIME CMD
2466 pts/2    00:00:00 bash
2558 pts/2    00:00:00 sort
2559 pts/2    00:00:00 ps
20486 pts/2   00:00:01 bash
(base) micah@MicahsPC:~/TWU$ ps
 PID TTY      TIME CMD
2466 pts/2    00:00:00 bash
2558 pts/2    00:00:02 sort
2560 pts/2    00:00:00 ps
20486 pts/2   00:00:01 bash
(base) micah@MicahsPC:~/TWU$ |
```



Kill PID

```
(base) micah@MicahsPC:~/TWU$ ps
 PID TTY      TIME CMD
2466 pts/2    00:00:00 bash
2558 pts/2    00:02:20 sort
2568 pts/2    00:00:00 ps
20486 pts/2   00:00:01 bash
(base) micah@MicahsPC:~/TWU$ kill 2558
(base) micah@MicahsPC:~/TWU$ ps
 PID TTY      TIME CMD
2466 pts/2    00:00:00 bash
2569 pts/2    00:00:00 ps
20486 pts/2   00:00:01 bash
[1]+  Terminated                  sort workshop_2024/twupa_bioinformaticsws_2024/data/Homo_sapiens.GRCh38.dna.primary_assembly.fa
(base) micah@MicahsPC:~/TWU$ |
```



Use “&” at end of line to run in the background

```
(base) micah@MicahsPC:~/TWU$ sort workshop_2024/twupa_bioinformaticsws_2024/data/Homo_sapiens.GRCh38.dna.primary_assembly.fa &
[1] 2558
(base) micah@MicahsPC:~/TWU$ ps
 PID TTY      TIME CMD
2466 pts/2    00:00:00 bash
2558 pts/2    00:00:00 sort
2559 pts/2    00:00:00 ps
20486 pts/2   00:00:01 bash
(base) micah@MicahsPC:~/TWU$ ps
 PID TTY      TIME CMD
2466 pts/2    00:00:00 bash
2558 pts/2    00:00:02 sort
2560 pts/2    00:00:00 ps
20486 pts/2   00:00:01 bash
(base) micah@MicahsPC:~/TWU$ |
```



History shows previous commands

```
631 echo "This is Bioinformatics!" >> hw.txt
632 cat hw.txt
633 echo "This is Bioinformatics!" > hw.txt
634 cat hw.txt
635 ls
636 cd ..
637 ls
638 clrst
639 clear
640 ls
641 ls -l | wc
642 ls -l
643 grep "^>" workshop_2024/twupa_bioinformaticsws_2024/data/Homo_sapiens.GRCh38.dna.primary_assembly.fa | wc
644 ps
645 clear
646 ls
647 sort workshop_2024/twupa_bioinformaticsws_2024/data/Homo_sapiens.GRCh38.dna.primary_assembly.fa &
648 ps
649 kill 2558
650 ps
651 history
(base) micah@MicahsPC:~/TwU$ |
```



Several text editors available

- There are several text editors available via bash, vim is the best if you can modally edit, but this takes much practice.
- Nano, pico, emacs, are built into the terminal
- You can use non-terminal text editors too:
 - Sublime text:
 - wget -qO - https://download.sublimetext.com/sublimehq-pub.gpg | gpg --dearmor | sudo tee /etc/apt/trusted.gpg.d/sublimehq-archive.gpg > /dev/null
 - echo "deb https://download.sublimetext.com/ apt/stable/" | sudo tee /etc/apt/sources.list.d/sublime-text.list
 - sudo apt-get update
 - sudo apt-get install sublime-text
 - subl workshop.txt



HISAT2 Alignment

- Now we know enough linux that we can actually start to do some bioinformatics
- We are going to build an index of the SARS-CoV-2 genome, so we can align some real data to it.



HISAT2 Alignment

- Download the SARS-CoV-2 Genome with wget:
 - wget https://github.com/mathornton01/twupa_bioinformaticsws_2024/raw/main/data/genomefiles/sarscov2.fasta
- **Build the hisat2 index with hisat2-build**
 - hisat2-build sarscov2.fasta sarscov2

```
len: 29903
gbwtLen: 29904
nodes: 29904
sz: 7476
gbwtSz: 7477
lineRate: 6
offRate: 4
offMask: 0xffffffff0
ftabChars: 10
eftabLen: 0
eftabSz: 0
ftabLen: 1048577
ftabSz: 4194308
offsLen: 1869
offsSz: 7476
lineSz: 64
sideSz: 64
sideGbwtSz: 48
sideGbwtLen: 192
numSides: 156
numLines: 156
gbwtTotLen: 9984
gbwtTotSz: 9984
reverse: 0
linearFM: Yes
Total time for call to driver() for forward index: 00:00:00
(base) micah@MicahsPC:~/TWU/workshop_2024/twupa_bioinformaticsws_2024/data/genomefiles$ ls
sarscov2.1.ht2 sarscov2.3.ht2 sarscov2.5.ht2 sarscov2.7.ht2 sarscov2.fasta
sarscov2.2.ht2 sarscov2.4.ht2 sarscov2.6.ht2 sarscov2.8.ht2 sequence.fasta
```



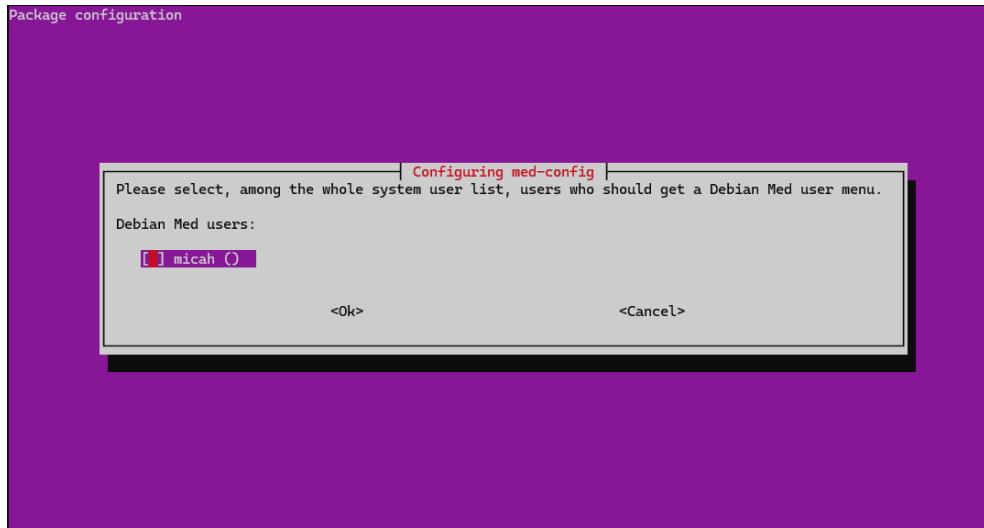
SRA Toolkit

- I will provide the files for this workshop, but you should know how to find your own files and download them
- For this you can use the SRA Toolkit



SRA Toolkit

- You can download it
 - Sudo apt-get install sra-toolkit



SRA Toolkit

- Just press enter on that screen.
- Type “which prefetch

```
(base) micah@MicahsPC:~/TWU/workshop_2024/twupa_bioinformaticsws_2024/data/genomefiles$ prefetch
Usage:
prefetch [options] <SRA accession> [...]
Download SRA files and their dependencies

prefetch [options] --cart <kart file>
Download kart file

prefetch [options] <URL> --output-file <FILE>
Download URL to FILE

prefetch [options] <URL> [...] --output-directory <DIRECTORY>
Download URL or URL-s to DIRECTORY

prefetch [options] <SRA file> [...]
Check SRA file for missed dependencies and download them

(base) micah@MicahsPC:~/TWU/workshop_2024/twupa_bioinformaticsws_2024/data/genomefiles$ which prefetch
/usr/bin/prefetch
```



SRA Toolkit

- We can download sequencing reads using the SRA-Toolkit.
- We type:
 - “prefetch <ACCESSION #>
 - “fastq-dump –X <num_reads> <.SRA file>



SRA Toolkit

- For example, suppose we are interested in the differential expression of genes by cancer cells that are introduced to a PARP-7 Inhibitor.



Finding Study Data

- We can search for “PARP7” in BioProject.
- Select the third result: “Transcriptional changes by PARP7 Inhibitor in cancer cell lines”

BioProject

Create alert Advanced Browse by Project attributes

Project Types Primary submission (5)

Data Types Other (1) Transcriptome (4)

Project Data SRA (5) GEO DataSets (5)

Scope Multi-isolate (5)

Attributes: Data Types/Material Transcriptome (4) Other (1)

Attributes: Capture Whole (5)

Attributes: Method Type Sequencing (4) Other (1)

Organism Groups

Display Settings: Summary, Sorted by Default order

Send to:

Filters:

Database:

Find items

Search results

Items: 5

[Transcriptional changes by RBN-2397 treatment and FRA1 knockdown in NCI-H1975 cells](#)
Organism: Homo sapiens
Taxonomy: [Homo sapiens \(human\)](#)
Project data type: Transcriptome or Gene expression
Scope: Multisolate
Group Hottiger, DMDD, University of Zurich
Accession: PRJNA955271 ID: 955271

[CRISPR/a functional screen with PARP7 inhibitor in NCI-H1373 cells](#)
Organism: Homo sapiens
Taxonomy: [Homo sapiens \(human\)](#)
Project data type: Other
Scope: Multisolate
Ribon Therapeutics, Inc.
Accession: PRJNA738855 ID: 738855

[Transcriptional changes by PARP7 inhibitor in cancer cell lines](#)
Organism: Homo sapiens
Taxonomy: [Homo sapiens \(human\)](#)

Search details

PARP7[All Fields]

Search

Recent activity

Turn

PARP7 (5)

Parp7 inhibitor (3)



Finding Study Data

- Luckily there are 84 sequencing data available, as well as 1 GEO dataset, we will look at this later, for now click the hyperlinked 84.

Resource Name	Number of Links
SEQUENCE DATA	
SRA Experiments	84
PUBLICATIONS	
PubMed	1
OTHER DATASETS	
BioSample	84
GEO DataSets	1



Finding Study Data

- We can tell some things about the study from the naming scheme of the SRA files.
- Select the first file:

Links from BioProject

Items: 1 to 20 of 84

<< First < Prev Page of 5 Next > Last >>

- [GSM5375104: NCI-H1373_DMSO_24hr_rep3; Homo sapiens; RNA-Seq](#)
 - 1. 1 ILLUMINA (NextSeq 500) run: 1.2M spots, 48.8M bases, 29Mb downloads
Accession: SRX11127241



Finding Study Data

- We can download the data using the SRA-toolkit, copy the run accession number at the bottom of the page. (SRR14794084)

Library:

Instrument: NextSeq 500

Strategy: RNA-Seq

Source: TRANSCRIPTOMIC

Selection: cDNA

Layout: SINGLE

Construction protocol: RNA was isolated using the MagMAX™-96 for Microarrays Total RNA Isolation Kit (Thermo Fisher Scientific, #AM1839) per manufacturer's instructions. RNA concentration was determined using a NanoDrop 8000 (ThermoFisher Scientific, Waltham, MA). Paired-end sample libraries consisting of 60 bp with 6 nucleotide indices were prepared for measuring high-throughput 3' digital gene expression based on a previously published protocol (Massachusetts Institute of Technology, Cambridge, MA) (Soumilon, 2014).

Experiment attributes:

GEO Accession: GSM5375104

Links:

NCBI link: [NCBI Entrez \(gds\)](#)

Runs: 1 run, 1.2M spots, 48.8M bases, [29Mb](#)

Run	# of Spots	# of Bases	Size	Published
SRR14794084	1,218,977	48.8M	29Mb	2021-06-14



Finding Study Data

- Download the file, we are going to be working with this file.
- At the terminal type:
 - “prefetch SRR14794084”
 - “fastq-dump SRR14794084/SRR14794084.sra”



Finding study data.

```
(base) micah@MicahsPC:~/TWU/workshop_2024/twupa_bioinformaticsws_2024/data/genomefiles/de$ prefetch SRR14794084
2024-07-23T16:41:09 prefetch.2.11.3: Current preference is set to retrieve SRA Normalized Format files with full base quality scores.
2024-07-23T16:41:09 prefetch.2.11.3: 1) Downloading 'SRR14794084'...
2024-07-23T16:41:09 prefetch.2.11.3: SRA Normalized Format file is being retrieved, if this is different from your preference, it may be due to current file availability.
2024-07-23T16:41:09 prefetch.2.11.3:  Downloading via HTTPS...
2024-07-23T16:41:12 prefetch.2.11.3:  HTTPS download succeed
2024-07-23T16:41:12 prefetch.2.11.3: 'SRR14794084' is valid
2024-07-23T16:41:12 prefetch.2.11.3: 1) 'SRR14794084' was downloaded successfully
(base) micah@MicahsPC:~/TWU/workshop_2024/twupa_bioinformaticsws_2024/data/genomefiles/de$ fastq-dump SRR14794084/SRR14794084.sra
Read 1218977 spots for SRR14794084/SRR14794084.sra
Written 1218977 spots for SRR14794084/SRR14794084.sra
(base) micah@MicahsPC:~/TWU/workshop_2024/twupa_bioinformaticsws_2024/data/genomefiles/de$ |
```



Finding study data

- We need to align to the human reference genome, we will use hisat2
- Hisat2 requires an index of the human reference genome to be generated.
- Hisat2-build utility can be used to generate an index, we will do this for the sars-cov-2 genome shortly, but for now, we will download the index from online.



Downloading HISAT2 GRCh38 index

- Type “wget https://genome-dl.s3.amazonaws.com/hisat/grch38_genome.tar.gz” in terminal to download the index for the human reference genome.
- This will be about 10 GB uncompressed, make sure you have space if following along.



Downloading HISAT2 GRCh38 index

- This will take a few minutes, in the meantime let us discuss what the hisat2 index is.

```
(base) micah@MicahsPC:~/TWU/workshop_2024/twupa_bioinformaticsws_2024/data/genomefiles/de$ wget https://genome-idx.s3.amazonaws.com/hisat/grch38_genome.tar.gz
--2024-07-23 11:44:43--  https://genome-idx.s3.amazonaws.com/hisat/grch38_genome.tar.gz
Resolving genome-idx.s3.amazonaws.com (genome-idx.s3.amazonaws.com)... 52.216.60.153, 52.217.100.228, 52.216.217.105, ...
Connecting to genome-idx.s3.amazonaws.com (genome-idx.s3.amazonaws.com)|52.216.60.153|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 4210306865 (3.9G) [binary/octet-stream]
Saving to: 'grch38_genome.tar.gz'

grch38_genome.tar.gz          20%[=====>]  809.09M  18.1MB/s   eta 3m 35s |
```



HISAT2 Index

- The HISAT2 index is a convenient representation of the human reference genome.
- It allows for fast alignment, including when reads are spliced.
- HISAT2 is very powerful software.



Burrows-Wheeler Transform

Suppose we would like to search for 'ACG' in the string 'GACGTG'.

- To Build the Index (FM),
 - Take the Burrows Wheeler Transform, and store the first column

1. Add '\$' and Take all rotations:
2. Sort the Rotations:

GACGTG\$
ACGTG\$G
CGTG\$GA
GTG\$GAC
TG\$GACG
G\$GACGT
\$GACGTG



\$GACGTG
ACGTG\$G
CGTG\$GA
G\$GACGT
GACGTG\$
GTG\$GAC
TG\$GACG

3. Take the Last and first Columns:

First: ([\$,1],[A,1],[C,1],[G,3],[T,1])

Last: GGAT\$CG



Burrows-Wheeler Transform

First-Last Mapping Property makes Querying
'ACG' in string easier

\$ GACGT G_0
 A_0 CGTG\$ G_1
 C_0 GTG\$G A_0
 G_0 \$GACG T_0
 G_1 ACGTG \$
 G_2 TG\$GA C_0
 T_0 G\$GAC G_2



Burrows-Wheeler Transform

1. Start from back of query, 'G' in 'ACG', Find all starting with 'G'

\$ GACGT G_0
A $_0$ CGTG\$ G_1
C $_0$ GTG\$G A $_0$
G $_0$ \$GACG T $_0$
G $_1$ ACGTG \$
G $_2$ TG\$GA C $_0$
T $_0$ G\$GAC G $_2$



Burrows-Wheeler Transform

2. Check Last Column of Corresponding Rows for 'C' in 'ACG', note C_0

\$	GACGT	G_0
A_0	CGTG\$	G_1
C_0	GTG\$G	A_0
G_0	\$GACG	T_0
G_1	ACGTG	\$
G_2	TG\$GA	C_0
T_0	G\$GAC	G_2



Burrows-Wheeler Transform

3. Check C_0 in First column, note A_0 in Last Column, corresponding to 'A' in 'ACG'.

\$	GACGT	G_0
A_0	CGTG\$	G_1
C_0	GTG\$G	A_0
G_0	\$GACG	T_0
G_1	ACGTG	\$
G_2	TG\$GA	C_0
T_0	G\$GAC	G_2

4. Therefore, alignment of 'ACG' corresponds with position of first 'A' or position 2 in 'GACGTG'



Burrows-Wheeler Transform

5. Note we also technically need to store the position of each of the characters in the last column of the input, so that we can map them back to the originating position when a match is found.

These things and more go into the HISAT2 index to help with spliced alignment of reads.



HISAT2 Index

- Now the hisat2 index should be nearly finished downloading.
- Unzip it with:
 - “gunzip grch38_genome.tar.gz”
- Untar it with:
 - “tar –xvf grch38_genome.tar”



HISAT2 index

```
(base) micah@MicahsPC:~/TWU/workshop_2024/twupa_bioinformaticsws_2024/data/genomefiles/de$ gunzip grch38_genome.tar.gz
(base) micah@MicahsPC:~/TWU/workshop_2024/twupa_bioinformaticsws_2024/data/genomefiles/de$ ls
SRR14794084 SRR14794084.fastq grch38_genome.tar
(base) micah@MicahsPC:~/TWU/workshop_2024/twupa_bioinformaticsws_2024/data/genomefiles/de$ tar -xvf grch38_genome.tar
grch38/
grch38/genome.5.ht2
grch38/genome.2.ht2
grch38/make_grch38.sh
grch38/genome.3.ht2
grch38/genome.4.ht2
grch38/genome.7.ht2
grch38/genome.1.ht2
grch38/genome.6.ht2
grch38/genome.8.ht2
(base) micah@MicahsPC:~/TWU/workshop_2024/twupa_bioinformaticsws_2024/data/genomefiles/de$ |
```



TEXAS WOMAN'S
UNIVERSITY

Read Quality Control

- The reads we downloaded from SRA likely need to be trimmed, and potentially other things need to be done with them prior to alignment.
- Let us investigate them with fastqc.



FASTQC

- Fastqc is an application that allows you to ascertain information about the quality of the reads.
- If you are on linux download it with the
 - “sudo apt-get install fastqc”
- Command.



FASTQC

- Run fastqc on the downloaded sequencing reads file with
 - “fastqc SRR14794084.fastq”

```
(base) micah@MicahsPC:~/TwU/workshop_2024/twupa_bioinformaticsws_2024/data/genomefiles/de$ fastqc SRR14794084.fastq
Started analysis of SRR14794084.fastq
Approx 5% complete for SRR14794084.fastq
Approx 10% complete for SRR14794084.fastq
Approx 15% complete for SRR14794084.fastq
Approx 20% complete for SRR14794084.fastq
Approx 25% complete for SRR14794084.fastq
Approx 30% complete for SRR14794084.fastq
Approx 35% complete for SRR14794084.fastq
Approx 40% complete for SRR14794084.fastq
Approx 45% complete for SRR14794084.fastq
Approx 50% complete for SRR14794084.fastq
Approx 55% complete for SRR14794084.fastq
Approx 60% complete for SRR14794084.fastq
Approx 65% complete for SRR14794084.fastq
Approx 70% complete for SRR14794084.fastq
Approx 75% complete for SRR14794084.fastq
Approx 80% complete for SRR14794084.fastq
Approx 85% complete for SRR14794084.fastq
Approx 90% complete for SRR14794084.fastq
Approx 95% complete for SRR14794084.fastq
Analysis complete for SRR14794084.fastq
```



FASTQC

- Fastqc produced an html file with the quality report, let's check it by navigating to it and opening it in a web browser.

Summary

- ✓ [Basic Statistics](#)
- ✓ [Per base sequence quality](#)
- ✓ [Per tile sequence quality](#)
- ✓ [Per sequence quality scores](#)
- ✗ [Per base sequence content](#)
- ! [Per sequence GC content](#)
- ✓ [Per base N content](#)
- ✓ [Sequence Length Distribution](#)
- ! [Sequence Duplication Levels](#)
- ! [Overrepresented sequences](#)
- ! [Adapter Content](#)



FASTQC

- The metrics are decent, we could probably stand to trim the data, but let us try to align it to the reference genome as it is.



HISAT2 Alignment

- Run the following command to align our fastq data to the human reference genome:

```
“hisat2 –x grch38/genome  
–U SRR14794084.fastq  
–S SRR14794084.sam”
```



HISAT2 Alignment

```
(base) micah@MicahsPC:~/TWU/workshop_2024/twupa_bioinformaticsws_2024/data/genomefiles/de$ hisat2 -x grch38/genome -U SRR14794084.fastq -S SRR14794084.sam  
1218977 reads; of these:  
1218977 (100.00%) were unpaired; of these:  
 416348 (34.16%) aligned 0 times  
 559278 (45.88%) aligned exactly 1 time  
 243351 (19.96%) aligned >1 times  
65.84% overall alignment rate
```

65.84% overall alignment rate is not great, but will serve for our purposes.

We now want to count all instances of gene hits and estimate the relative transcript abundances, we can do this using one of several tools, but the classic tool is from the “subread” package and is called “featureCounts”

To install the subread package type: “sudo apt-get install subread”

Check if featureCounts is installed then with the “which featureCounts” command



featureCounts

- In order to determine if a read hits a specific transcript we must know where they are, this is specified in the gtf file, download it here:
 - wget https://ftp.ebi.ac.uk/pub/databases/gencode/Gencode_human/release_46/gencode.v46.annotation.gtf.gz
 - gunzip gencode.v46.annotation.gtf.gz



featureCounts

- featureCounts produces a counts.txt table containing the raw expression of genes.
 - featureCounts -a gencode.v46.annotation.gtf -o counts.txt SRR14794084.sam

```
(base) micalah@MicalahPC:~/TMI/workshop_2024/tmupa_bionformaticsws_2024/data/genomefiles/de$ featureCounts -a gencode.v46.annotation.gtf -o counts.txt SRR14794084.sam
=====
featureCounts setting =====
Input files : 1 SAM file
               SRR14794084.sam
Output file : counts.txt
               Summary : counts.txt.summary
Paired-end : no
Count read pairs : no
Annotation : gencode.v46.annotation.gtf (GTF)
Dir for temp files : ./
Threads : 1
Level : meta-feature level
Multimapping reads : not counted
Multi-overlapping reads : not counted
Min overlapping bases : 1
=====
Running =====
Load annotation file gencode.v46.annotation.gtf ...
```



TEXAS WOMAN'S
UNIVERSITY

featureCounts

- Produced a tab-separated file, where column 1 is gene name, and 7 is expression in our file of interest.

```
(base) micah@MicahsPC:~/TWU/workshop_2024/twupa_bioinformaticsws_2024/data/genomefiles/de$ head -n 10 counts.txt | cut -f1,7
# Program:featureCounts v2.0.3; Command:"featureCounts" "-a" "gencode.v46.annotation.gtf" "-o" "counts.txt" "SRR14794084.sam"
Geneid  SRR14794084.sam
ENSG00000290825.1      0
ENSG00000223972.6      0
ENSG00000227232.6      3
ENSG00000278267.1      0
ENSG00000243485.5      0
ENSG00000284332.1      0
ENSG00000237613.2      0
ENSG00000268920.3      0
```



featureCounts

- To perform differential expression analysis we would need to replicate this process for each of the 84 files – or write a script to generalize this.



featureCounts

- Luckily the raw gene-counts matrix file is provided for this study in GEO.
- Download it:
 - wget <https://ftp.ncbi.nlm.nih.gov/geo/series/GSE177nnn/GSE177494/suppl/GSE177494%5FRaw%5Fgene%5Fcounts%5Fmatrix%2Etxt%2Egz>
 - gunzip GSE177494_Raw_gene_counts_matrix.txt.gz



DESeq2

- We are going to use R and the R-package DESeq2 to analyze this expression data.
- Open R in the terminal where the downloaded expression data is with “R”



DESeq2

- To create a DESeq2 dataset we need two dataframes, one with the expression data, and one with column attributes.
- Let us extract column attributes from column names.



DESeq2

- Optionally open R studio here
 - Run “rstudio”
 - Change directories with the `setwd("")` command
- Read in the transcript abundance data with this R command.
- `df <- read.table("GSE177494_Raw_gene_counts_matrix.txt")`



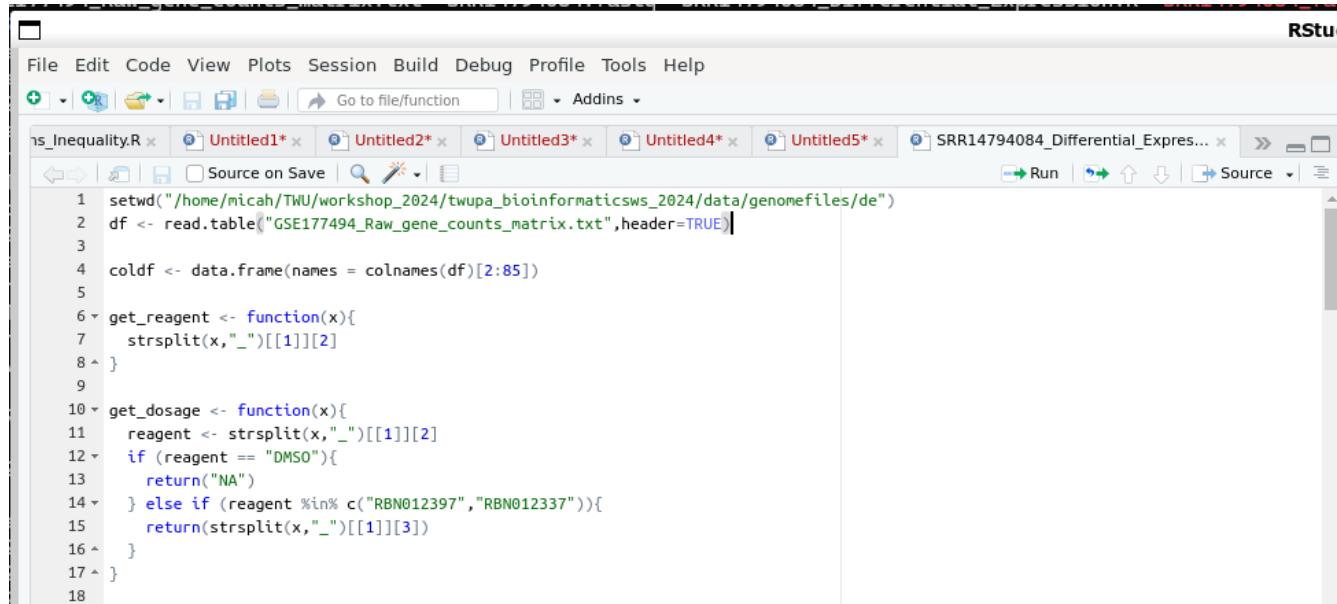
DESeq2

- Download the Analysis R script I wrote here:
 - wget https://github.com/mathornton01/twupa_bioinformaticsws_2024/raw/main/tools/SRR14794084_Differential_Expression.R
- Open The R script in rstudio by typing:
 - rstudio SRR14794084_Differential_Expression.R



DESeq2

- You should now see this



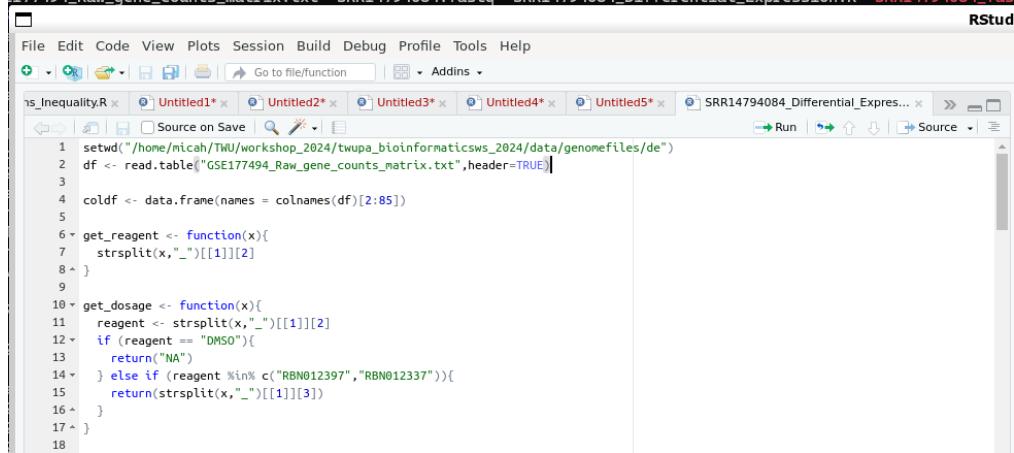
The screenshot shows the RStudio interface with the following details:

- File Menu:** File, Edit, Code, View, Plots, Session, Build, Debug, Profile, Tools, Help.
- Toolbar:** Includes icons for New, Open, Save, Run, and Source.
- Text Editor:** Displays an R script named "ns_Inequality.R".
- Code Content:**

```
1 setwd("/home/micah/TWU/workshop_2024/twupa_bioinformaticsws_2024/data/genomefiles/de")
2 df <- read.table("GSE177494_Raw_gene_counts_matrix.txt",header=TRUE)
3
4 coldf <- data.frame(names = colnames(df)[2:85])
5
6 get_reagent <- function(x){
7   strsplit(x,"_")[[1]][2]
8 }
9
10 get dosage <- function(x){
11   reagent <- strsplit(x,"_")[[1]][2]
12   if (reagent == "DMSO"){
13     return("NA")
14   } else if (reagent %in% c("RBN012397","RBN012337")){
15     return(strsplit(x,"_")[[1]][3])
16   }
17 }
18
```
- Run Buttons:** Run, Source.



DESeq2



The screenshot shows the RStudio interface with the title bar "RStud". The menu bar includes File, Edit, Code, View, Plots, Session, Build, Debug, Profile, Tools, Help. The toolbar has icons for file operations like Open, Save, and Run. The code editor window contains the following R script:

```
1 setwd("/home/nicah/TWU/workshop_2024/twupa_bioinformaticsws_2024/data/genomefiles/de")
2 df <- read.table("GSE177494_Raw_gene_counts_matrix.txt", header=TRUE)
3
4 coldf <- data.frame(names = colnames(df)[2:85])
5
6 get_reagent <- function(x){
7   strsplit(x, "_")[[1]][2]
8 }
9
10 get_dosage <- function(x){
11   reagent <- strsplit(x, "_")[[1]][2]
12   if (reagent == "DMSO"){
13     return("NA")
14   } else if (reagent %in% c("RBN012397", "RBN012337")){
15     return(strsplit(x, "_")[[1]][3])
16   }
17 }
18
```

Change the setwd command to point to the directory where you have stored the GSE177494_Raw_gene_counts_matrix.txt



DESeq2

- First we will read the counts matrix and set up the column data matrix.
 - Pull reagent,
 - Dosage,
 - Time
 - Replicate

```
coldf <- data.frame(names = colnames(df)[2:85])

get_reagent <- function(x){
  strsplit(x, "_")[[1]][2]
}

get_dosage <- function(x){
  reagent <- strsplit(x, "_")[[1]][2]
  if (reagent == "DMSO"){
    return("NA")
  } else if (reagent %in% c("RBN012397", "RBN012337")){
    return(strsplit(x, "_")[[1]][3])
  }
}

get_time <- function(x){
  reagent <- strsplit(x, "_")[[1]][2]
  if (reagent == "DMSO"){
    return(strsplit(x, "_")[[1]][3])
  } else if (reagent %in% c("RBN012397", "RBN012337")){
    return(strsplit(x, "_")[[1]][4])
  }
}
```



DESeq2

- Once the column data is set up, we can load the DESeq2 library.
- BiocManager::install("DESeq2")

```
28 ▼ get_replicate <- function(x){  
29   reagent <- strsplit(x,"_")[[1]][2]  
30 ▼ if (reagent == "DMSO"){  
31   return(strsplit(x,"_")[[1]][4])  
32 ▼ } else if (reagent %in% c("RBN012397","RBN012337")){  
33   return(strsplit(x,"_")[[1]][5])  
34 ▲ }  
35 ▲ }  
36  
37 coldf$reagent <- factor(unlist(lapply(coldf$names, get_reagent)))  
38 coldf$dosage <- factor(unlist(lapply(coldf$names, get_dosage)))  
39 coldf$time <- factor(unlist(lapply(coldf$names, get_time)))  
40 coldf$replicate <- factor(unlist(lapply(coldf$names, get_replicate)))  
41 coldf  
42  
43 library(DESeq2)  
44
```



DESeq2

```
43 library(DESeq2)
44
45 rownames(df) <- df[,1]
46 df <- df[,2:85]
47
48 dds <- DESeqDataSetFromMatrix(countData=round(df[,coldf$dosage %in% c("10nM","NA") & coldf$time == "6hr" & coldf$reagent %in% c("RBN012397","DMSO")]),
49                               colData=coldf[coldf$dosage %in% c("10nM","NA") & coldf$time == "6hr" & coldf$reagent %in% c("RBN012397","DMSO"),],
50                               design = ~ reagent)
51
52 dds <- DESeq(dds)
53 res <- results(dds)
54 res
55
56 resOrdered <- res[order(res$padj),]
57
58 RBN012397_dose_10nM_time_6hr_up <- rownames(resOrdered[!is.na(resOrdered$padj) & resOrdered$padj < 0.1 & resOrdered$log2FoldChange > 0,])
59 RBN012397_dose_10nM_time_6hr_down <- rownames(resOrdered[!is.na(resOrdered$padj) & resOrdered$padj < 0.1 & resOrdered$log2FoldChange < 0,])
60
```

- We create the DESeqDataSet using this code, note we have to round because values were normalized, DESeq expects raw counts.



DESeq2

- We apply this same procedure several times, subsetting the dataset appropriately to get the up-regulated and down-regulated genes under each condition for each reagent.



DESeq2

```
library("ggvenn")
A <- list(RBN012397_300nM=RBN012397_dose_300nM_time_24hr_up,RBN012397_10nM=RBN012397_dose_10nM_time_24hr_up,RBN012397_30nM=RBN012397_dose_30nM_time_24hr_up)
ggvenn(A)

library("VennDiagram")
draw.pairwise.venn(area1 = 217+189, area2 = 209, cross.area= 189)

library("eulerr")
fit1 <- euler(c("300nM"=201,"10nM"=19,"30nM"=5,"300nM&10nM"=82, "300nM&30nM"=16,"10nM&30nM"=1,"300nM&10nM&30nM"=107))
plot(fit1)
```

- There are several ways to form a Venn's diagram concerning what genes were in common and what were not



BCFTools and Variant Calling

- Now we are switching gears to look at calling variants in a different type of data.
- Let us download the SARS-CoV-2 Genome:
 - wget https://github.com/mathornton01/twupa_bioinformaticsws_2024/raw/main/data/genomefiles/sarscov2.fasta



BCFTools and Variant Calling

- Build a hisat2 index for the sars-cov-2 genome:
 - hisat2-build sarscov2.fasta sarscov2
- Builds index for sarscov2.fasta, and uses prefix sarscov2 for filenames



BCFTools and Variant Calling

- Fetch some SARS-CoV-2 Sequencing Reads
 - prefetch SRR29438173
- Dump 200 K reads from the file
 - fastq-dump -X 200000 SRR29438173/SRR29438173.sra



BCFTools and Variant Calling

- Run fastqc on the data and check the results
 - fastqc SRR29438173.fastq
- Trim the reads using TrimGalore



TrimGalore

- # Check that cutadapt is installed
 - cutadapt –version
- # Check that FastQC is installed
 - fastqc -v#
- Install Trim Galore
 - curl -fsSL <https://github.com/FelixKrueger/TrimGalore/archive/0.6.10.tar.gz> -o trim_galore.tar.gz
 - tar xvzf trim_galore.tar.gz
- # Run Trim Galore
 - ./TrimGalore-0.6.10/trim_galore SRR29438173.fastq -o SRR29438173_trim



BCFTools and Variant Calling

- Now we can align to the reference.
 - hisat2 -x sarscov2 -U SRR29438173_trim/SRR29438173_trimmed.fq -S SRR29438173_trimmed.sam
- Now we can generate the bam with samtools
 - samtools view -bS SRR29438173_trimmed.sam | samtools sort -o SRR29438173_trimmed.bam
- Finally we can use bcftools to generate the variant call file.
 - bcftools mpileup -Ob -o SRR29438173_trimmed.bcf -f sarscov2.fasta SRR29438173_trimmed.bam
 - bcftools call -vmO z -o SRR29438173_trimmed.vcf.gz SRR29438173_trimmed.bcf --ploidy 1
 - gunzip SRR29438173_trimmed.vcf.gz
- Now we can inspect the variant call file,

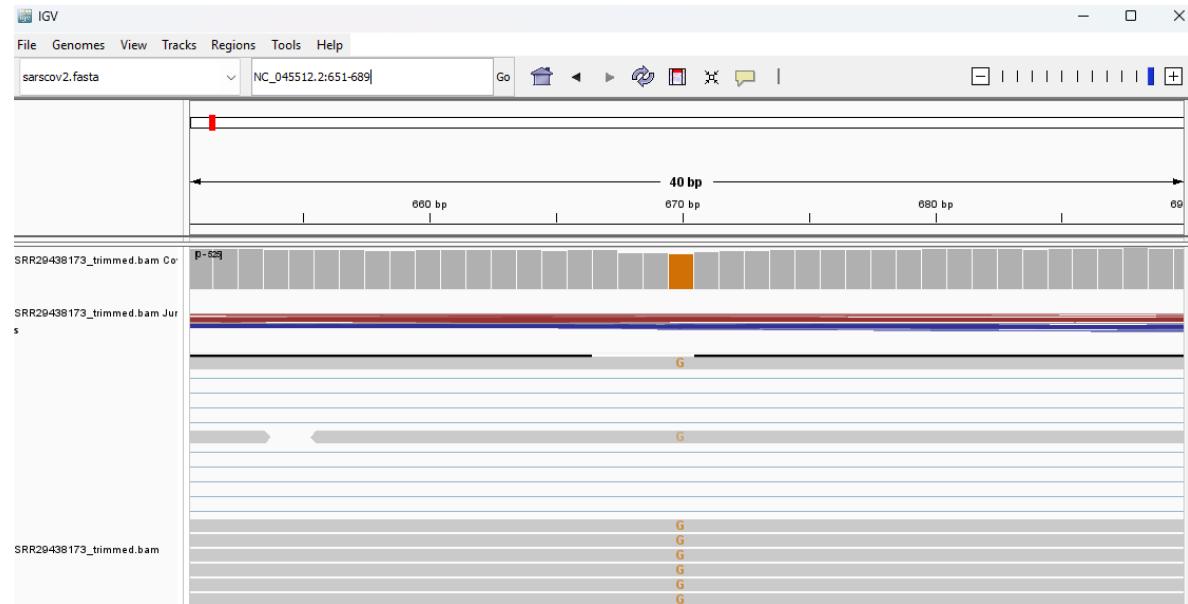


BCFTools and Variant Calling

```
(base) micah@MicahsPC:~/TwU/workshop_2024/twupa_bionformatics_sws_2024/data/genomefiles/variant_calling$ cat SRR29438173_trimmed.vcf
##fileformat=VCFv4.2
##FILTER=<ID=PASS,Description="All filters passed">
##bcftoolsVersion=1.13+htslib-1.13+ds
##bcfCommand=mpileup -Ob -o SRR29438173_trimmed.bcf -f sarscov2.fasta SRR29438173_trimmed.bam
##reference=file://sarscov2.fasta
##contig=<ID=NC_045512.2,length=29903>
##ALT=<ID=*,Description="Represents allele(s) other than observed.">
##INFO=<ID=INDEL,Number=0,Type=Flag,Description="Indicates that the variant is an INDEL.">
##INFO=<ID=IDV,Number=1,Type=Integer,Description="Maximum number of raw reads supporting an indel">
##INFO=<ID=IMF,Number=1,Type=Float,Description="Maximum fraction of raw reads supporting an indel">
##INFO=<ID=DP,Number=1,Type=Integer,Description="Raw read depth">
##INFO=<ID=VDB,Number=1,Type=Float,Description="Variant Distance Bias for filtering splice-site artefacts in RNA-seq data (bigger is better)",Version="3">
##INFO=<ID=RPBZ,Number=1,Type=Float,Description="Mann-Whitney U-z test of Read Position Bias (closer to 0 is better)">
##INFO=<ID=MQBZ,Number=1,Type=Float,Description="Mann-Whitney U-z test of Mapping Quality Bias (closer to 0 is better)">
##INFO=<ID=BOBZ,Number=1,Type=Float,Description="Mann-Whitney U-z test of Base Quality Bias (closer to 0 is better)">
##INFO=<ID=MQSBZ,Number=1,Type=Float,Description="Mann-Whitney U-z test of Mapping Quality vs Strand Bias (closer to 0 is better)">
##INFO=<ID=SCBZ,Number=1,Type=Float,Description="Mann-Whitney U-z test of Soft-Clip Length Bias (closer to 0 is better)">
##INFO=<ID=FS,Number=1,Type=Float,Description="Phred-scaled p-value using Fisher's exact test to detect strand bias">
##INFO=<ID=SGB,Number=1,Type=Float,Description="Segregation based metric.">
##INFO=<ID=MQ0F,Number=1,Type=Float,Description="Fraction of MQ0 reads (smaller is better)">
##FORMAT=<ID=PL,Number=G,Type=Integer,Description="List of Phred-scaled genotype likelihoods">
##FORMAT=<ID=GT,Number=1,Type=String,Description="Genotype">
##INFO=<ID=AC,Number=A,Type=Integer,Description="Allele count in genotypes for each ALT allele, in the same order as listed">
##INFO=<ID=AN,Number=1,Type=Integer,Description="Total number of alleles in called genotypes">
##INFO=<ID=DP4,Number=4,Type=Integer,Description="Number of high-quality ref-forward , ref-reverse, alt-forward and alt-reverse bases">
##INFO=<ID=MQ,Number=1,Type=Integer,Description="Average mapping quality">
##bcftools_callCommand=call -vm0 z -o SRR29438173_trimmed.vcf.gz --ploidy 1 SRR29438173_trimmed.bcf; Date=Tue Jul 23 15:54:49 2024
##bcftools_callCommand=call -vm0 z -o SRR29438173_trimmed.vcf.gz --ploidy 1 SRR29438173_trimmed.bcf; Date=Tue Jul 23 15:54:49 2024
#CHROM POS ID REF ALT QUAL FILTER INFO FORMAT SRR29438173_trimmed.bam
NC_045512.2 21 . C T 7.30814 . DP=1;SGB=-0.379885;FS=0;MQ0F=0;AC=1;AN=1;DP4=0,0,0,1;MQ=60 GT:PL 1:36,0
NC_045512.2 241 . C T 4.38466 . DP=1;SGB=-0.379885;FS=0;MQ0F=0;AC=1;AN=1;DP4=0,0,1,0;MQ=60 GT:PL 1:32,0
NC_045512.2 670 . T G 225.417 . DP=176;VDB=0.0248337;SGB=-0.693147;MQSBZ=0.746674;FS=0;MQ0F=0.00568182;AC=1;AN=1;DP4=0,0,113,63;MQ=59 GT:PL 1:255,0
NC_045512.2 897 . C A 225.417 . DP=133;VDB=0.0451553;SGB=-0.693147;FS=0;MQ0F=0;AC=1;AN=1;DP4=0,0,108,25;MQ=60 GT:PL 1:255,0
NC_045512.2 2790 . C T 225.417 . DP=77;VDB=0.588039;SGB=-0.693147;FS=0;MQ0F=0;AC=1;AN=1;DP4=0,0,55,22;MQ=60 GT:PL 1:255,0
NC_045512.2 3037 . C T 225.417 . DP=89;VDB=0.102794;SGB=-0.693147;FS=0;MQ0F=0;AC=1;AN=1;DP4=0,0,74,15;MQ=60 GT:PL 1:255,0
NC_045512.2 3431 . G T 228.398 . DP=80;VDB=0.0852523;SGB=-0.693147;RPBZ=1.711028;MQSBZ=-0.160128;BQBZ=-0.398077;SCBZ=-3.74559;FS=0;MQ0F=0.025;AC=1;AN=1;DP4=0,0,58 GT:PL 1:255,0
NC_045512.2 3565 . T C 225.417 . DP=25;VDB=0.142777;SGB=-0.692914;FS=0;MQ0F=0;AC=1;AN=1;DP4=0,0,13,12;MQ=60 GT:PL 1:255,0
NC_045512.2 4184 . G A 225.417 . DP=92;VDB=0.0375051;SGB=-0.693147;FS=0;MQ0F=0;AC=1;AN=1;DP4=0,0,69,23;MQ=60 GT:PL 1:255,0
NC_045512.2 6183 . A G 225.417 . DP=86;VDB=0.171423;SGB=-0.693147;FS=0;MQ0F=0;AC=1;AN=1;DP4=0,0,63,23;MQ=60 GT:PL 1:255,0
NC_045512.2 7842 . A G 225.417 . DP=73;VDB=0.0689107;SGB=-0.693147;FS=0;MQ0F=0;AC=1;AN=1;DP4=0,0,54,19;MQ=60 GT:PL 1:255,0
NC_045512.2 8293 . C T 225.417 . DP=72;VDB=0.184533;SGB=-0.693147;FS=0;MQ0F=0;AC=1;AN=1;DP4=0,0,55,17;MQ=60 GT:PL 1:255,0
NC_045512.2 8393 . G A 225.417 . DP=64;VDB=0.12862;SGB=-0.693147;FS=0;MQ0F=0;AC=1;AN=1;DP4=0,0,52,12;MQ=60 GT:PL 1:255,0
NC_045512.2 9244 . C T 225.417 . DP=25;VDB=0.301763;SGB=-0.693147;FS=0;MQ0F=0;AC=1;AN=1;DP4=0,0,60,16;MQ=60 GT:PL 1:255,0
```

BCFTools and Variant Calling

- We can verify that these variants are actually there by looking at the BAM file in IGV



Pseudo and Quasi Alignment & Quantification Resolution



TEXAS WOMAN'S
UNIVERSITY

RNA-Seq Experiments (Key Difference from DNA)

- In DNA-Seq, we usually align ‘NGS sequencing reads’ to a reference genome.
 - These sequences include all exons and introns, in fixed order.



- In RNA-Seq on the other hand we might try to align to a ‘transcriptome’.
 - These include a set of ‘transcripts’ which contain different possible sequences encoded by the same gene



RNA-Seq Experiments (Key Questions)

- In RNA-Seq experiments researchers hope to answer questions such as:
 1. How do abundances of a particular gene transcripts in the same population vary?
 - Ex. Do subjects with a high abundance of transcript A, tend also to exhibit relatively high abundances of transcript B?
 2. Do the abundances of gene transcripts vary with some observable phenotypic characteristic?
 - Ex. Do patients with latent state tuberculosis exhibit more abundance of transcript A than those with active state?
 3. What constitutes the most likely genetic transcript profile for a particular subject?
 - Ex. Is Transcript A more abundant in this subject?
 4. Etc ...
- So if we do not have exact *positional* alignments for reads, that is okay, as long as we are able to determine the most likely transcript that they came from. (Expectation Maximization for Multinomial Data)



RNA-Seq Experiments Overview

- In some RNA-Seq Experiments the *actual* alignment of reads may not be measurable in some cases.
- For instance, consider the following small example,



- It is possible that many RNA-Seq reads might be compatible with the same transcripts.
- Therefore for each of the RNA-Seq reads, the *actual* alignment is not directly recoverable.



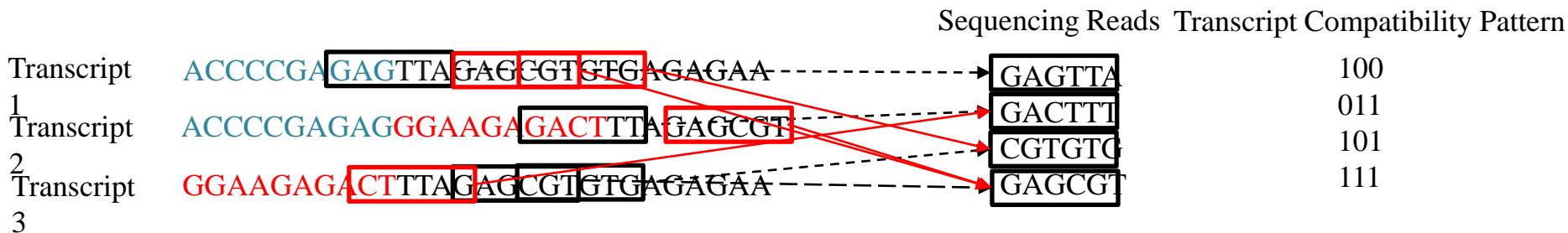
RNA-Seq Experiments Overview

- Instead of working directly with the sequences to produce *positional* alignments (the exact position where the read came from), coarsened compatibility patterns may be observed for the transcripts the read is compatible with.
- From the previous example,



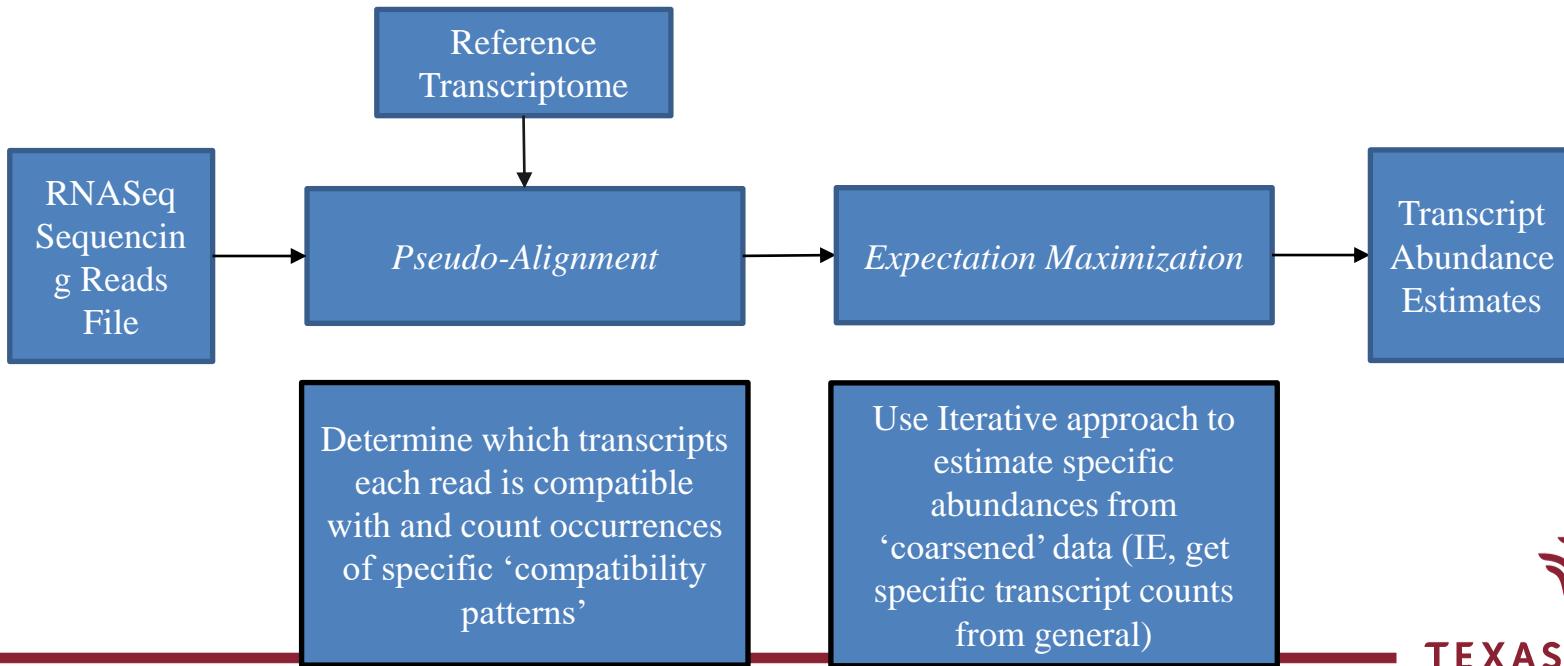
RNA-Seq Experiments Overview

- Instead of working directly with the sequences to produce *positional* alignments (the exact position where the read came from), coarsened compatibility patterns may be observed for the transcripts the read is compatible with.
- From the previous example,

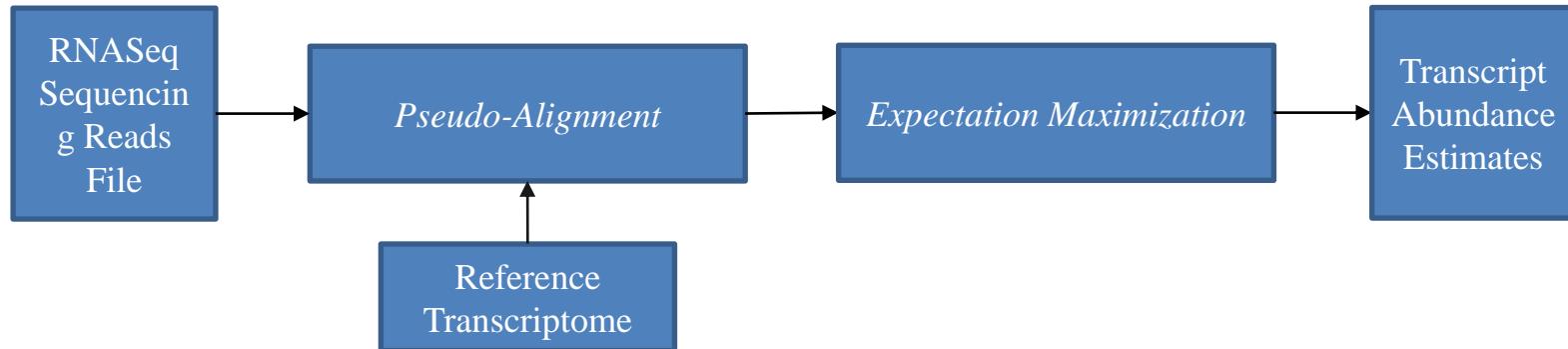


RNA-Seq Experiments Overview

- Now suppose that there are 10 Million such reads aligning to the transcriptome.
- Since *positional* alignment information may not allow the discernment of transcripts,
 - Not necessary to determine and report such information.
- Instead, many RNA-Seq transcript quantification tools use the following general procedure:



RNA-Seq Experiments Overview



- First data is collected from the RNA of the subject(s) under study using NGS technologies (short read sequencing)
- Compatibility for each read with each transcript in a *Reference Transcriptome* is determined by some Pseudo-Alignment procedure.
- Expectation Maximization for abundances of the true transcript counts X_i are determined from the coarsened pattern counts Y_i .
- Estimates of the parameters of the distribution of Y_i (for $i = 1, \dots, N_t$), γ_i , are presented as abundance estimates, and multiplied by the number of reads N_r for expectations.



RNA-Seq Abundance Estimation: Kallisto

- First data is collected from the RNA of the subject(s) under study using NGS technologies (short read sequencing)
- Compatibility for each read with each transcript in a *Reference Transcriptome* is determined by some Pseudo-Alignment procedure.
- Expectation Maximization for abundances of the true transcript counts X_i are determined from the coarsened pattern counts Y_i .
- Estimates of the parameters of the distribution of Y_i (for $i = 1, \dots, N_t$), γ_i , are presented as abundance estimates, and multiplied by the number of reads N_r for expectations.
- Sometimes TPM (Transcripts per million reads) are also presented.
- Two popular procedures which implement this approach are Salmon and Kallisto.
 - These will be presented and a demonstration given.
 - Tomorrow we will walk through installation of these software and their usage, and a third software will be demonstrated*.



Salmon



Kallisto

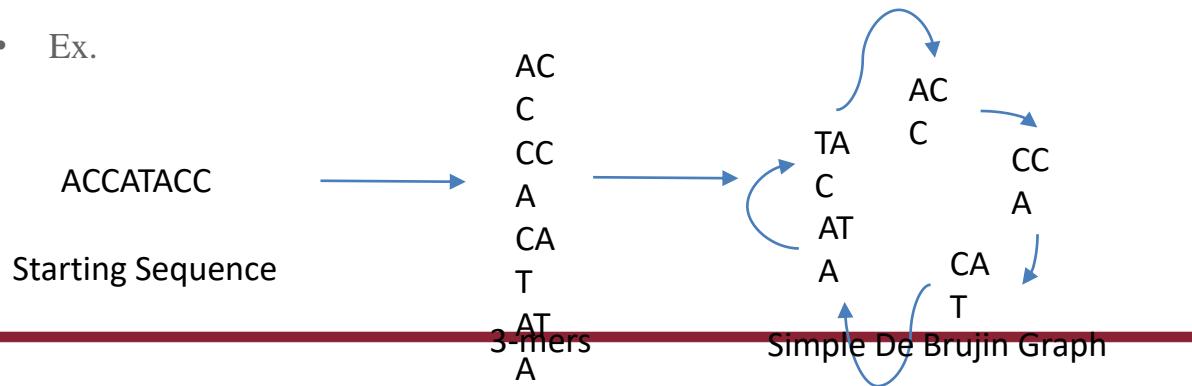
* we are creating a new tool called H2Q which takes into account SNP variability between alleles to provide allele specific transcript quantification results



TEXAS WOMAN'S
UNIVERSITY

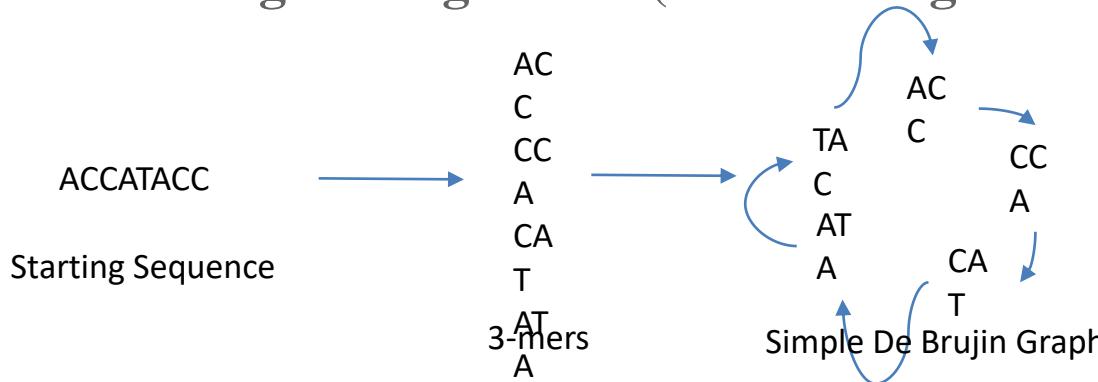
Kallisto: Understanding the Algorithm

- As previously stated, the RNA-Seq quantification algorithms used in state of the art applications generally follow two steps:
 - “Pseudo-Alignments” indicating the subset of transcripts which are ‘compatible’ with each read are determined.
 - The Expectation Maximization algorithm is applied to determine the Maximum Likelihood Estimators for the Multinomial Proportions associated with each transcript.
- Kallisto uses a transcriptome de Bruijn graph to determine which transcripts are compatible with each read (pair).
 - The De Bruijn Graph represents sequences by connecting nodes of subsequences (in this case we call them k -mers, where k denotes the size).
 - Ex.

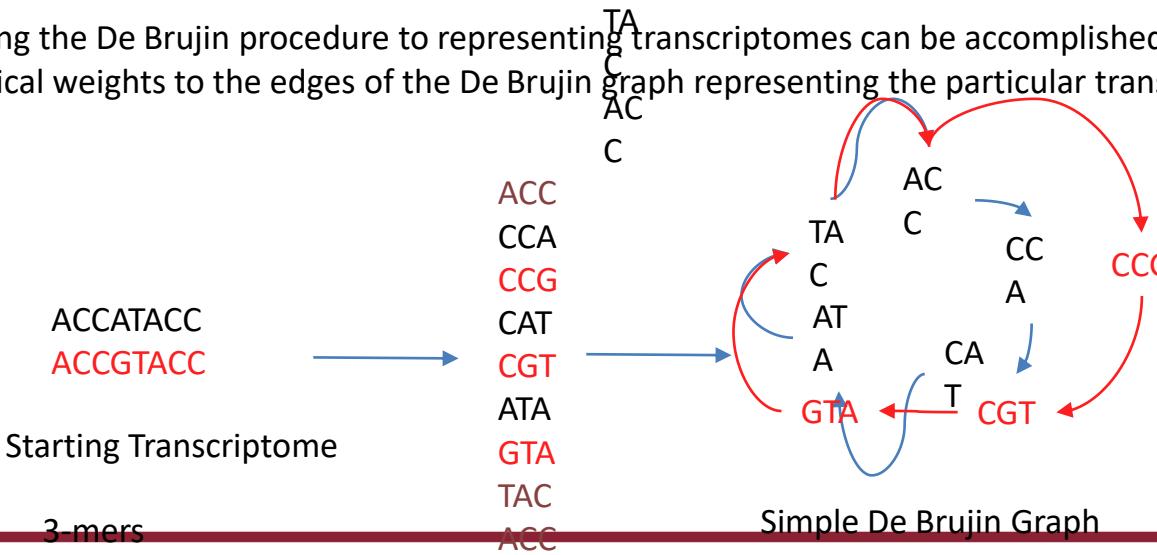


TEXAS WOMAN'S
UNIVERSITY

Kallisto: Understanding the Algorithm (Pseudo-Alignment)

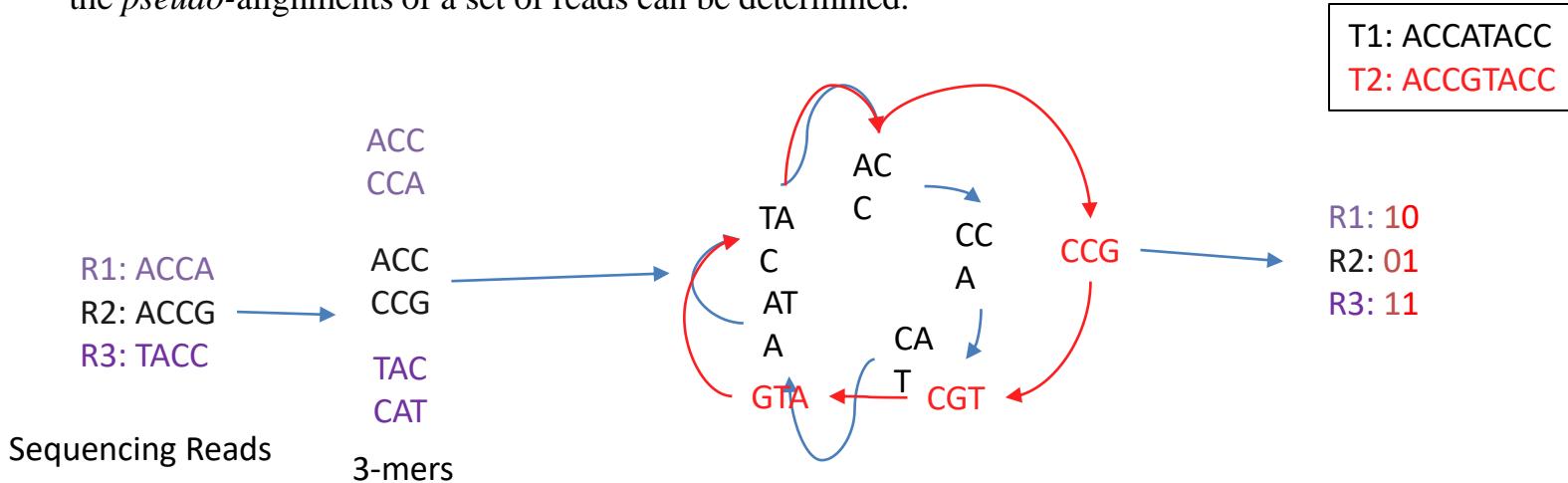


- Extending the De Bruijn procedure to representing transcriptomes can be accomplished by adding “colors” or categorical weights to the edges of the De Bruijn graph representing the particular transcript of alignment.



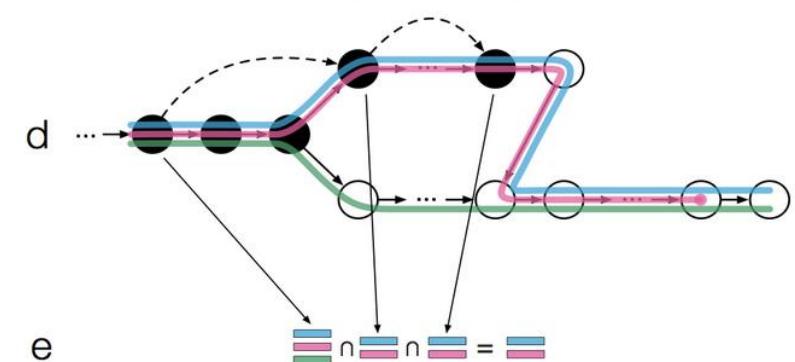
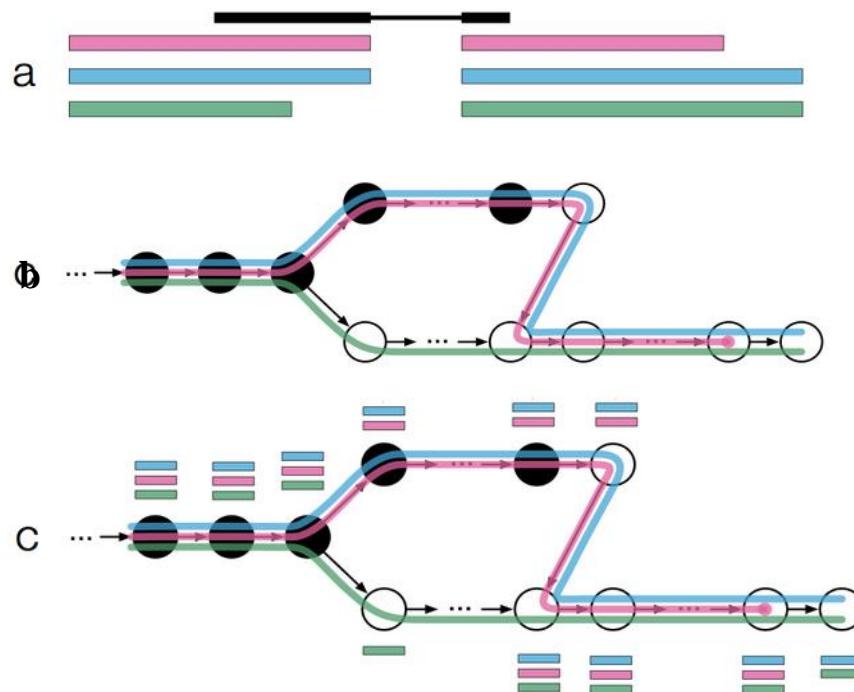
Kallisto: Understanding the Algorithm (Pseudo-Alignment)

- Given a new set of reads, the k -mers can be determined, and then using the De Bruijn Graph Transcriptome, the *pseudo-alignments* of a set of reads can be determined.



Comparison of 3-mers from reads to color DBG transcriptome allows the determination of compatibility counts.

Kallisto Pseudo-Alignment Example Backup



This is taken from Fong Chun Chan's
Blog post on "How
Pseudoalignments Work in Kallisto."

<https://tinyheero.github.io/2015/09/02/pseudoalignments-kallisto.html>



Salmon: Understanding the Algorithm (Quasi-Mapping)

$$\Pr\{f_j|t_i\} = \Pr\{\ell|t_i\} \cdot \Pr\{p|t_i, \ell\} \cdot \Pr\{o|t_i\} \cdot \underline{\Pr\{alf_j, t_i, p, o, \ell\}}$$

Equation 6

- If alignments are being used, then the Fragment-Transcript agreement model used by Salmon incorporates the true alignment information in this term (The Probability of generating alignment a of fragment f_j , being drawn from t_i , with position, orientation, and length, p, o, ℓ .
- In ‘Quasi-Mapping’ Mode however this term is fixed as 1.
- In Salmon, there are additional phases where parameters are calculated and utilized for determining a better transcript abundance quantification.

Patro R, Duggal G, Love MI, Irizarry RA, Kingsford C. Salmon provides fast and bias-aware quantification of transcript expression. *Nat Methods*. 2017;14(4):417-419. doi:10.1038/nmeth.4197

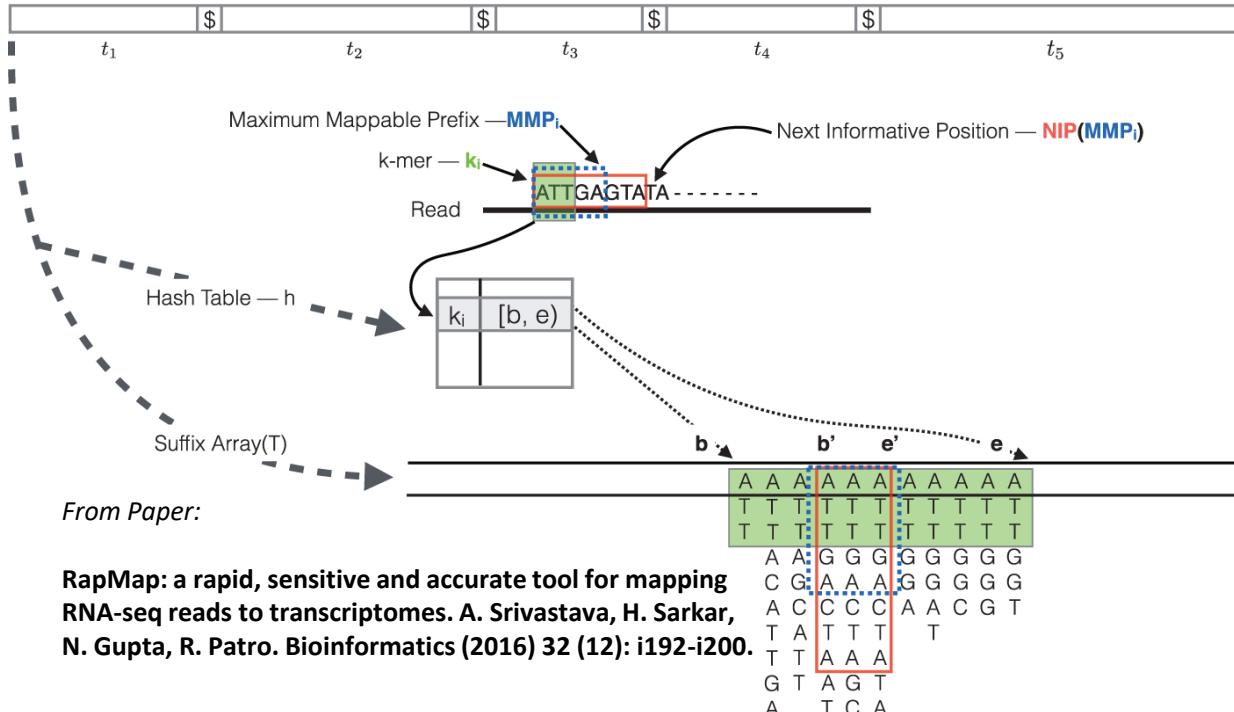
<https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5600148/#FD10>



TEXAS WOMAN'S
UNIVERSITY

Salmon: Understanding the Algorithm (Quasi-Mapping)

Transcriptome (T) with separator



1. ^{t₆} First a Suffix Array is built for the transcriptome
2. A K-mer Hash table for indexing into the the suffix array is constructed
3. Reads are scanned, and when k-mers in the hash table are found, all suffixes matching are extracted from the suffix array
4. The longest prefix in common among all the suffixes (called the MMP Maximal Matching Prefix) is found.
5. This is repeated for all kmers, and then the intersection of transcripts in the MMP
6. The Transcripts which are intersected by the associated MMP are provided as the compatible quasi-mappings



* From Article: Quantification of transcript abundance using Salmon Introduction to bulk RNA-seq

https://hbctraining.github.io/Intro-to-rnaseq-hpc-salmon-flipped/lessons/08_quasi_alignment_salmon.html

Questions about Pseudo-Alignment/Quasi-Mapping?



TEXAS WOMAN'S
UNIVERSITY

Pseudo/Quasi Alignment in RNA Experiments

- Sometimes the *exact* position of a sequencing read is not of critical import.
 - There are a few approaches for resolving the *approximate* location of a read.
 - Procedures work by determining the subset of *transcript isoforms* compatible with a read.
- Two such approaches are known as:
 - Pseudo-Alignment
 - The Approach used by **Kallisto**.
 - Uses the De Bruijn ('Deh-Broine') graph procedure.
 - Quasi-Alignment
 - The Approach used by **Salmon**.
 - Uses a *K*-mer Hash table and Suffix Array.

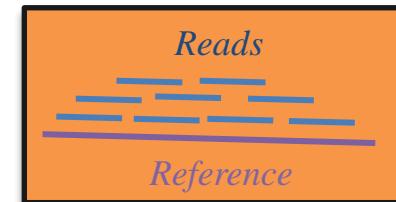
Resources – Kallisto (Pseudo-alignment)

1. <https://tinyheero.github.io/2015/09/02/pseudoalignments-kallisto.html> (Higher Level Overview pseudo alignment)
2. <https://www.youtube.com/watch?v=f-ecmECK7lw> (Video Describing how To Build The De Bruijn graph)
3. <https://www.nature.com/articles/nbt.2023> (Nature Primer on Using De Bruijn Graphs for Genomic Alignments).

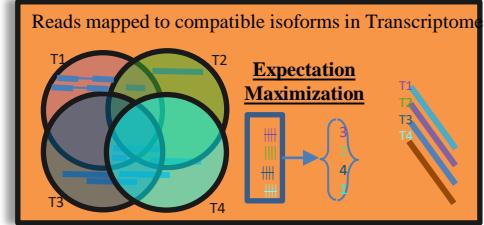
Resources – Salmon (Quasi-alignment)

1. https://hbctraining.github.io/Intro-to-rnaseq-hpc-salmon-flipped/lessons/08_quasi_alignment_salmon.html (Higher Level Overview Quasi-Alignment)
2. <https://academic.oup.com/bioinformatics/article/32/12/i192/2288985?login=true> (RapMap Paper and Description).

Typical 'DNA-Seq Like' Experiment



Typical 'RNA-Seq Like' Experiment

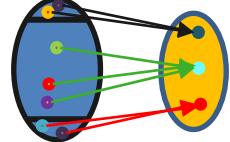


Recall that in most typical sequencing experiments we are dealing with a large collection of shorter subsequences called **reads**, which we attempt to map to a larger sequence known as the **reference**.



Expectation Maximization (in general) – Incomplete Data & A Restricted Case

Many-to-one relationship
 \mathcal{X} \mathcal{Y}



- Two general uses include:
 - determination of maximum likelihood estimates for parameters when missing data is present and
 - estimation of missing or otherwise incomplete data.
- In general, suppose that we would like to observe the values, $x_1, x_2, \dots x_n$, to determine something about the parameters of the random variable X which has sample space \mathcal{X} as shown (top right).
 - However, we are only able to observe, $y_1, y_2, \dots y_n$, valuations of the random variable Y which has sample space \mathcal{Y} onto which there exists a many-to-one mapping from \mathcal{X} .
 - In other words, there are multiple values possible to observe in \mathcal{X} corresponding to the same value in \mathcal{Y} .
- Suppose, at first, that the distribution of \mathbf{X} (note boldface indicates that \mathbf{X} could be a vector quantity) is one of the exponential family of distributions generally denoted,

$$f_{\mathbf{X}}(\mathbf{x}|\boldsymbol{\theta}) = b(\mathbf{x})e^{(\boldsymbol{\theta}\mathbf{t}(\mathbf{x})^T)}a(\boldsymbol{\theta})^{-1}$$

$\boldsymbol{\theta}$ is a parameter [column]-vector (of size r).

$\mathbf{t}(\mathbf{x})^T$ is the sufficient statistic [row]-vector (of size r).

$a(\cdot), b(\cdot)$, are any arbitrary function.

e is the natural number.

See section II of the Dempster, Laird, Rubin paper mentioned below for more details about natural parameters.



Expectation Maximization (in general) – The Algorithm

- The “simple characterization” of the EM algorithm according to Dempster, Laird, and Rubin (DLR77) is:
 - (1) With $\theta^{(p)}$ indicating the estimate of θ at the p^{th} step of the algorithm, estimate the complete-data sufficient statistics $\mathbf{t}(\mathbf{x})$ by finding

$$\mathbf{t}^{(p)} = E(\mathbf{t}(\mathbf{x}) | \mathbf{y}, \boldsymbol{\theta}^{(p)}).$$

- (2) Perform maximum likelihood estimation to determine $\theta^{(p+1)}$ from $\mathbf{t}^{(p)}$,

$$E(\mathbf{t}(\mathbf{x}) | \boldsymbol{\theta}) = \mathbf{t}^{(p)}.$$

- Proof of convergence to the maximum likelihood value is given the DLR77, as are details regarding further generalizations of the expectation maximization algorithm.
- The algorithm is broadly applicable in many cases, and not all of the applications have been discovered yet.



TEXAS WOMAN'S
UNIVERSITY

Expectation Maximization (in general) – A Multinomial Example

- Suppose that there are marbles of five colors in a bag.
 - Red marbles are denoted by ‘R’
 - Orange marbles are denoted by ‘O’
 - Yellow marbles by ‘Y’
 - Green marbles by ‘G’
 - Blue marbles by ‘B’
- Now, you personally cannot tell a difference between the orange and the yellow marbles by eye, and therefore are able to produce counts of four categories of marbles only (that is: “Red”, “Orange or Yellow”, “Green”, and “Blue”).

[EXAMPLE]

- Suppose it is known ahead of time that the proportions of the *actual* colors of each of the marbles are related via an unknown parameter π , such that for the unobservable true color of an arbitrarily selected marble i , denoted c_i (true color) given below induces a distribution on the observable o_i (observed color):

$$P \left(c_i = \begin{pmatrix} \text{Red} \\ \text{Orange} \\ \text{Yellow} \\ \text{Green} \\ \text{Blue} \end{pmatrix} \right) = \begin{pmatrix} (1 - \pi)/4 \\ \pi/4 \\ 1/2 \\ (1 - \pi)/4 \\ \pi/4 \end{pmatrix} \Rightarrow P \left(o_i = \begin{pmatrix} \text{Red} \\ \text{Orange or Yellow} \\ \text{Green} \\ \text{Blue} \end{pmatrix} \right) = \begin{pmatrix} (1 - \pi)/4 \\ 1/2 + \pi/4 \\ (1 - \pi)/4 \\ \pi/4 \end{pmatrix}$$



Expectation Maximization (in general) – A Multinomial Example (Continued)

Suppose that we observe 197 marbles, and arrive at the following counts:

R—Red: 18
 OY—Orange or Yellow: 125
 G—Green: 20
 B — Blue: 34

$$x_j = \sum_{i=1}^{197} \mathbb{1}(c_i \equiv j) \quad j \in \begin{pmatrix} \text{Red} \\ \text{Orange} \\ \text{Yellow} \\ \text{Green} \\ \text{Blue} \end{pmatrix}$$

$$y_t = \sum_{i=1}^{197} \mathbb{1}(o_i \equiv t) \quad t \in \begin{pmatrix} \text{Red} \\ \text{Orange or Yellow} \\ \text{Green} \\ \text{Blue} \end{pmatrix}$$

- The Likelihood on π for the full data can be expressed as:

$$f(x|\pi) = \frac{(\sum_{i=1}^5 x_i)!}{\prod_{i=1}^5 (x_i!)} \cdot \left(\frac{1-\pi}{4}\right)^{x_1} \cdot \left(\frac{\pi}{4}\right)^{x_2} \cdot \left(\frac{1}{2}\right)^{x_3} \cdot \left(\frac{1-\pi}{4}\right)^{x_4} \cdot \left(\frac{\pi}{4}\right)^{x_5}$$

- The coarsened/incomplete Likelihood on π for the full data can be expressed as:

$$g(y|\pi) = \frac{(\sum_{i=1}^4 y_i)!}{\prod_{i=1}^4 (y_i!)} \cdot \left(1 - \frac{\pi}{4}\right)^{y_1} \cdot \left(\frac{1}{2} + \frac{\pi}{4}\right)^{y_2} \cdot \left(1 - \frac{\pi}{4}\right)^{y_3} \cdot \left(\frac{\pi}{4}\right)^{y_4}$$

$$P(c_i = \begin{pmatrix} \text{Red} \\ \text{Orange} \\ \text{Yellow} \\ \text{Green} \\ \text{Blue} \end{pmatrix}) = \begin{pmatrix} (1-\pi)/4 \\ \pi/4 \\ 1/2 \\ (1-\pi)/4 \\ \pi/4 \end{pmatrix} \Rightarrow P(c_i = \begin{pmatrix} \text{Red} \\ \text{Orange or Yellow} \\ \text{Green} \\ \text{Blue} \end{pmatrix}) = \begin{pmatrix} (1-\pi)/4 \\ 1/2 + \pi/4 \\ (1-\pi)/4 \\ \pi/4 \end{pmatrix}$$

- Let the *actual* color counts be denoted by the values $(x_1, x_2, x_3, x_4, x_5)$ such that x_1 corresponds to the count of marbles which were actually red, x_2 to those which were Orange, and so on...
- Let the observed color counts be denoted by the values (y_1, y_2, y_3, y_4) which are given in this example as $(18, 125, 20, 34)$.
- Furthermore, it is known that $y_2 = x_2 + x_3$.



Expectation Maximization (in general) – A Multinomial Example (E-Step)

- Clearly, due to the fact that a marble cannot *actually* be two colors simultaneously, there is no probability that any marble is *truly* both orange and yellow at the same time, therefore we may express the probability that a marble is orange or yellow as follows:

$$\begin{aligned} P(o_i = (\text{Orange or Yellow})) &= P\left(c_i \in \begin{pmatrix} \text{Orange} \\ \text{Yellow} \end{pmatrix}\right) = P(c_i = \text{Orange}) + P(c_i = \text{Yellow}) - P(c_i = \text{Yellow} \& c_i = \text{Orange}) \\ &= P(c_i = \text{Orange}) + P(c_i = \text{Yellow}) - 0 = \frac{\pi}{4} + \frac{1}{2} \end{aligned}$$

- From here we can derive the expression for the maximum likelihood estimates of the unobserved counts for orange and yellow marbles (x_2, x_3) in terms of the observed count of “orange or yellow” marbles (y_2).

$$P(c_i = \text{Orange} | o_i = (\text{Orange or Yellow})) = \frac{P(o_i = (\text{Orange or Yellow}) \& c_i = \text{Orange})}{P(o_i = (\text{Orange or Yellow}))} = \frac{P(c_i = \text{Orange})}{P(o_i = (\text{Orange or Yellow}))} = \frac{\frac{\pi}{4}}{\frac{\pi}{4} + \frac{1}{2}}$$

$$P(c_i = \text{Yellow} | o_i = (\text{Orange or Yellow})) = \frac{\frac{1}{2}}{\frac{\pi}{4} + \frac{1}{2}}$$

Therefore the conditional expectation of x_2 and x_3 are:

$$E(x_2|y_2) = y_2 \frac{\frac{\pi}{4}}{\frac{\pi}{4} + \frac{1}{2}} \quad \text{and} \quad E(x_3|y_2) = y_2 \frac{\frac{1}{2}}{\frac{\pi}{4} + \frac{1}{2}}$$



Expectation Maximization (in general) – A Multinomial Example (M-Step)

- Recall that the full likelihood for the multinomial distribution was given by:

$$f(\mathbf{x}|\pi) = \frac{(\sum_{i=1}^5 x_i)!}{\prod_{i=1}^5 (x_i!)} \cdot \left(\frac{1-\pi}{4}\right)^{x_1} \cdot \left(\frac{\pi}{4}\right)^{x_2} \cdot \left(\frac{1}{2}\right)^{x_3} \cdot \left(\frac{1-\pi}{4}\right)^{x_4} \cdot \left(\frac{\pi}{4}\right)^{x_5}$$

$$\Rightarrow \log L(\pi|\mathbf{x}) = \log \frac{(\sum_{i=1}^5 x_i)!}{\prod_{i=1}^5 (x_i!)} + x_1 \log\left(\frac{(1-\pi)}{4}\right) + x_2 \log\left(\frac{\pi}{4}\right) + x_3 \log\left(\frac{1}{2}\right) + x_4 \log\left(\frac{(1-\pi)}{4}\right) + x_5 \log\left(\frac{\pi}{4}\right)$$

- In this example, only x_2 and x_3 are unobservable, the rest are known:

$$\begin{aligned} \frac{\partial \log L(\pi|\mathbf{x})}{\partial \pi} &= x_1 \left(\frac{4}{1-\pi}\right) \left(-\frac{1}{4}\right) + x_2 \left(\frac{4}{\pi}\right) \left(\frac{1}{4}\right) + x_4 \left(\frac{4}{1-\pi}\right) \left(-\frac{1}{4}\right) + x_5 \left(\frac{4}{\pi}\right) \left(\frac{1}{4}\right) = \frac{x_1}{\pi-1} + \frac{x_2}{\pi} + \frac{x_4}{\pi-1} + \frac{x_5}{\pi} \\ \Rightarrow \frac{x_1}{\hat{\pi}-1} + \frac{x_2}{\hat{\pi}} + \frac{x_4}{\hat{\pi}-1} + \frac{x_5}{\hat{\pi}} &= 0 \Rightarrow (x_2 + x_5)(1 - \hat{\pi}) = (x_1 + x_4)\hat{\pi} \Rightarrow x_2 + x_5 - x_2\hat{\pi} - x_5\hat{\pi} = x_1\hat{\pi} + x_4\hat{\pi} \\ \Rightarrow x_2 + x_5 &= (x_1 + x_2 + x_4 + x_5)\hat{\pi} \Rightarrow \hat{\pi} = \frac{(x_2 + x_5)}{x_1 + x_2 + x_4 + x_5} \Rightarrow -x_1\hat{\pi} - x_4\hat{\pi} + x_1 + x_4 = x_2\hat{\pi} + x_5\hat{\pi} \\ \Rightarrow x_1 + x_4 &= x_2\hat{\pi} + x_5\hat{\pi} + x_1\hat{\pi} + x_4\hat{\pi} \end{aligned}$$

Suppose that we observe 197 marbles, and
arrive at the following counts:

$$\begin{pmatrix} x_1 \\ x_4 \\ x_5 \end{pmatrix} = \begin{pmatrix} 18 \\ 20 \\ 34 \end{pmatrix} \Rightarrow \hat{\pi} = \frac{x_2 + 34}{18 + x_2 + 20 + 34}$$

$$\begin{pmatrix} \text{R-Red: 18} \\ \text{OY-Orange or Yellow: 125} \\ \text{G-Green: 20} \\ \text{B-Blue: 34} \end{pmatrix}$$

$$\therefore \frac{1}{\hat{\pi}} = \frac{18 + x_2 + 20 + 34}{x_2 + 34} = 1 + \frac{38}{x_2 + 34} \Rightarrow \hat{\pi} = \frac{1}{1 + \frac{38}{x_2 + 34}}$$



Expectation Maximization (in general) – A Multinomial Example (Iteration)

- Taking the conditional expectations for the computation of x_2 and x_3 will depend on a particular estimation of π , an initial estimate ($\pi^{(0)}$) must be supplied to the algorithm to start the procedure, then conditional expectations for the missing (coarsened) data at the p^{th} step (where $p \in \{1, 2, \dots\}$) is given by:

[E – Step] $E_{(p)}(x_2|y_2) = y_2 \frac{\frac{\pi^{(p-1)}}{4}}{\frac{\pi^{(p-1)}}{4} + \frac{1}{2}}$ - and - $E_{(p)}(x_3|y_2) = y_2 \frac{\frac{1}{2}}{\frac{\pi^{(p-1)}}{4} + \frac{1}{2}}$

[M – Step] $\widehat{\pi^{(p)}} = \frac{1}{1 + \frac{38}{E_{(p)}(x_2|y_2) + 34}}$

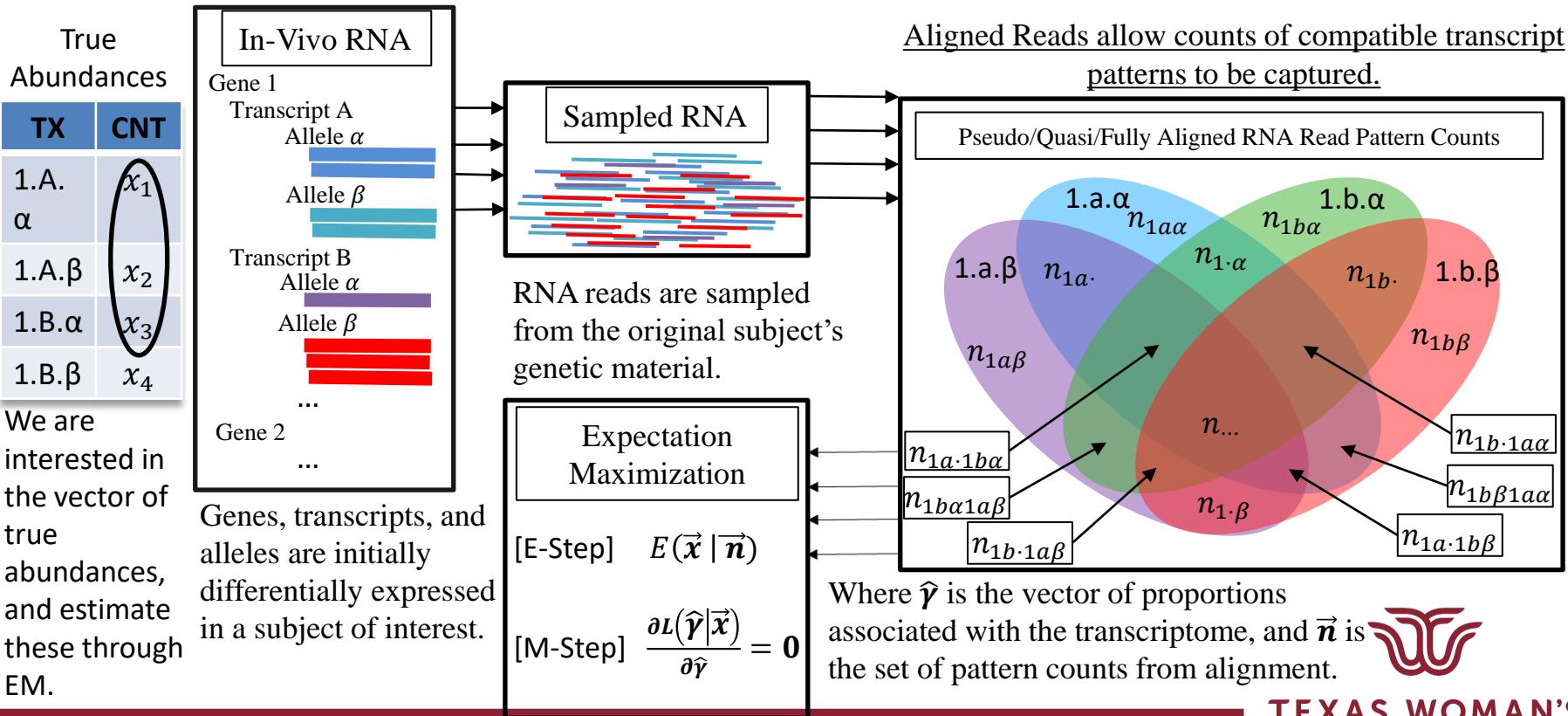
- Convergence Criteria:
 - Generally we use relative convergence criteria (when the change in the parameters from step p to step $p + 1$ falls below a relative tolerance ε_R) to determine when to stop iterating, for instance, the iteration will continue until:

[Convergence]
$$\left(\frac{1}{1 + \frac{38}{E_{(p)}(x_2|y_2) + 34}} - \frac{1}{1 + \frac{38}{E_{(p-1)}(x_2|y_2) + 34}} \right)^2 \leq \varepsilon_R$$



Expectation Maximization (Genetic Abundance Estimation)

- We observe N_r RNA-seq reads from an experiment involving a transcriptome of size T .
 - Each of the N_r reads came specifically from *only* one of the T categories.



Expectation Maximization (Genetic Abundance Estimation) [M-Step]

- Clearly, the distribution of the reads among their true transcript sources can be modeled as multinomial.
 - The probability distribution of the count vector of the true abundances, \vec{X} , is

$$\Pr(\vec{X} = \vec{x}) = \frac{(\sum_{i=1}^N x_i)!}{\prod_{i=1}^N (x_i)!} \prod_{i=1}^N \gamma_i^{x_i}$$

- This probability distribution function doubles as the Likelihood for the parameter vector $\vec{\gamma}$ under the observed data \vec{x} .

$$L(\vec{\gamma}|\vec{x}) = \frac{(\sum_{i=1}^N x_i)!}{\prod_{i=1}^N (x_i)!} \prod_{i=1}^N \gamma_i^{x_i} \Rightarrow \ell(\vec{\gamma}|\vec{x}) = \log \frac{(\sum_{i=1}^N x_i)!}{\prod_{i=1}^N (x_i)!} + \sum_{i=1}^N x_i \log \gamma_i \Rightarrow \frac{\partial \ell(\vec{\gamma}|\vec{x})}{\partial \gamma_i} = \frac{x_i}{\gamma_i} = \frac{N}{\sum_{i=1}^N \gamma_i} = 1$$

- Do not forget that there is an inherent constraint on the parameter space (the sum of all proportions must be one).
 - We must optimize $\ell(\vec{\gamma}|\vec{x})$ subject to the constraint: $\sum_{i=1}^N \gamma_i = 1$.
 - This is accomplished by using the method of Lagrange multipliers.

$$\ell'(\vec{\gamma}, \lambda) = \ell(\vec{\gamma}) + \lambda \left(1 - \sum_{i=1}^N \gamma_i \right) \Rightarrow \frac{\partial \ell'(\vec{\gamma}, \lambda)}{\partial \gamma_i} = \frac{x_i}{\gamma_i} - \lambda \Rightarrow \frac{x_i}{\hat{\gamma}_i} - \lambda = 0 \Rightarrow \hat{\gamma}_i = \frac{x_i}{\lambda}$$

$$\hat{\gamma}_i = \frac{x_i}{\lambda} \Rightarrow \sum_{i=1}^N \frac{x_i}{\lambda} = 1 \Rightarrow \frac{1}{\lambda} \sum_{i=1}^N x_i = 1 \Rightarrow \lambda = \sum_{i=1}^N x_i = N_r \Rightarrow \hat{\gamma}_i = \frac{x_i}{N_r}$$



Expectation Maximization (Genetic Abundance Estimation) [E-Step]

- The algorithm calculates the conditional expectation for missing x_i values during the E-Step.
 - If x_i is missing, we must first determine a valid estimate of x_i using the parameter estimated from the previous step (or the initial value used).
 - Instead of observing the vector \vec{x} directly, we observe the pattern count vector \vec{n} .
 - Let the elements of \vec{n} , $(n_1, n_2, \dots, n_{N_k})$ be indexed by j , which runs from 1 to the number of unique compatibility patterns (N_k).
 - The conditional expectations of the missing components of \vec{x} are computed using the elements of \vec{n} , which compose the counts of patterns including those same missing components.
 - For example, if x_1 is missing, but we determine there are reads which align to x_1 as well as others, say n_1, n_3 , and, n_5 are compatible with transcript 1, then each of these quantities would be used to compute the conditional expectation of the missing value.
 - Note, we either begin with $\gamma^{(0)}$, or have iterated to the p^{th} step, and have $\gamma^{(p-1)}$.
 - The conditional expectation of the missing value, x_i , is determined by considering the observations of those elements of \vec{n} which contain alignments to transcript i .
 - **Let the indicator ψ_{ij} be 1 if read i is present in compatibility pattern j and 0 otherwise.**

$$E_{(p)}(x_i | \vec{n}) = \frac{\gamma_i^{(p-1)}}{\sum_{j=1}^{N_p} \psi_{ij} n_j \gamma_i^{(p-1)}} N_r$$



Expectation Maximization (Genetic Abundance Estimation)

- The EM algorithm amounts to applying these two operations in alternative order until there is convergence in the parameter vector.

$$\text{[E-Step]} \quad E_{(p)}(x_i | \vec{n}) = \frac{\widehat{\gamma_i^{(p-1)}}}{\sum_{j=1}^{N_k} \psi_{ij} n_j \widehat{\gamma_i^{(p-1)}}} n_j \quad \text{[M-Step]} \quad \widehat{\gamma_i}^{(p)} = \frac{E_{(p)}(x_i | \vec{n})}{N_r}$$

- The EM Algorithm will achieve convergence when the change from step $p - 1$ to p is below some user selected relative tolerance ε_r .

$$\text{[Convergence Criteria]} \quad \widehat{\gamma_i}^{(p)} - \widehat{\gamma_i}^{(p-1)} \leq \varepsilon_r$$

- Note the Expectation Maximization algorithm for Multinomial count data (as above) can be applied in a general case. This algorithm is implemented in multiple software packages available for use, but we have created a general version (For a copy, please request via email at this stage).



Expectation Maximization (Multinomial algorithm example)

Suppose we have true abundances
 Transcript 1 : 500 (0.5)
 Transcript 2 : 200 (0.2)
 Transcript 3 : 300 (0.3)

$$\boldsymbol{\gamma}^{(0)} = \begin{bmatrix} \frac{1}{3} & \frac{1}{3} & \frac{1}{3} \\ 1 & 0 & 1 \\ 1 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 550 \\ 400 \\ 50 \end{bmatrix} = \mathbf{n}$$

But that we can only observe whether reads are in the following:
 (T1,T3): $300+250 = 550$
 (T1,T2): $200+200 = 400$
 (T3): $50 = 50$

It is typical to start with uniform probabilities for transcripts

$$\gamma_i^{(0)} = \frac{1}{N_r} \forall i$$

$$\boldsymbol{\gamma}^{(1)} = \begin{bmatrix} \frac{475}{1000} & \frac{20}{100} & \frac{325}{1000} \\ 1 & 0 & 1 \\ 1 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 550 \\ 400 \\ 50 \end{bmatrix} = \mathbf{n}$$

E Step (1)

$$E(x|\mathbf{n}, \boldsymbol{\gamma}^{(0)}) = \begin{bmatrix} \frac{\left(\frac{1}{3}\right)}{\frac{1}{3} + \frac{1}{3}} (550) + \frac{\left(\frac{1}{3}\right)}{\frac{1}{3} + \frac{1}{3}} (400) \\ \frac{\left(\frac{1}{3}\right)}{\frac{1}{3} + \frac{1}{3}} (400) \\ \frac{\left(\frac{1}{3}\right)}{\frac{1}{3} + \frac{1}{3}} (550) + (1)(50) \end{bmatrix} = \begin{bmatrix} 475 \\ 200 \\ 325 \end{bmatrix}$$

$$\boldsymbol{\gamma}^{(1)} = \begin{bmatrix} 0.475 \\ 0.2 \\ 0.325 \end{bmatrix}$$

M Step (1)



Expectation Maximization (Multinomial algorithm example)

Suppose we have true abundances

Transcript 1 : 500 (0.5)

Transcript 2 : 200 (0.2)

Transcript 3 : 300 (0.3)

But that we can only observe whether reads are in the following:

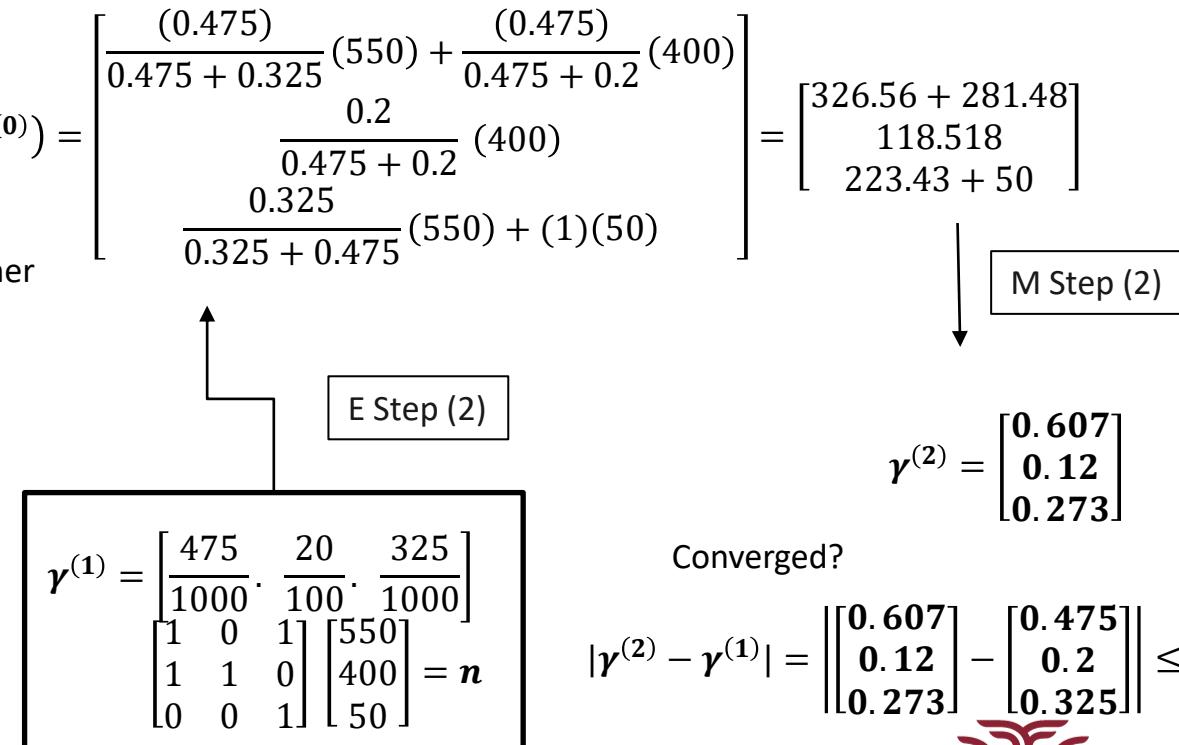
$$(T1,T3): 300+250 = 550$$

$$(T1,T2): 200+200 = 400$$

$$(T3): 50 = 50$$

It is typical to start with uniform probabilities for transcripts

$$\gamma_i^{(0)} = \frac{1}{N_r} \forall i$$



Genetic Transcript Abundance Software Kallisto & Salmon

- Today, just follow along on the screen with me, tomorrow we will work through getting Kallisto and Salmon on your personal device, and working through some example problems with them together.
- In order to run Kallisto and Salmon, you first need to have a transcriptome in the FASTA format (from which the sequencing reads file of interest is taken)
 - **Note if you do not have a transcriptome file, you might need to produce one by first parsing GTF/SNP/ or another Variant Call Format like file, and the reference sequence to which it corresponds.**
 - **For example, we created an RNA-Seq read simulator which will allow for us to produce a transcriptome file for GTF/SNP/FASTA (reference) files.**
- In this short demonstration We will use an example transcriptome from a subset of genes on human chromosome 22.



Genetic Transcript Abundance Software Kallisto & Salmon

- To access the help for either Salmon or Kallisto, you can use:

```
kallisto 0.46.0    kallisto
```

Usage: kallisto <CMD> [arguments] ..

Where <CMD> can be one of:

index	Builds a kallisto index
quant	Runs the quantification algorithm
bus	Generate BUS files for single-cell
data	
pseudo	Runs the pseudoalignment step
merge	Merges several batch runs
h5dump	Converts HDF5-formatted results to
plaintext	Inspects and gives information about
inspect	an index
version	Prints version information
cite	Prints citation information

Running kallisto <CMD> without arguments prints usage information for <CMD>

```
Salmon -h
```

```
salmon v1.8.0
```

Usage: salmon -h|--help or
salmon -v|--version or
salmon -c|--cite or
salmon [--no-version-check] <COMMAND> [-h
| options]

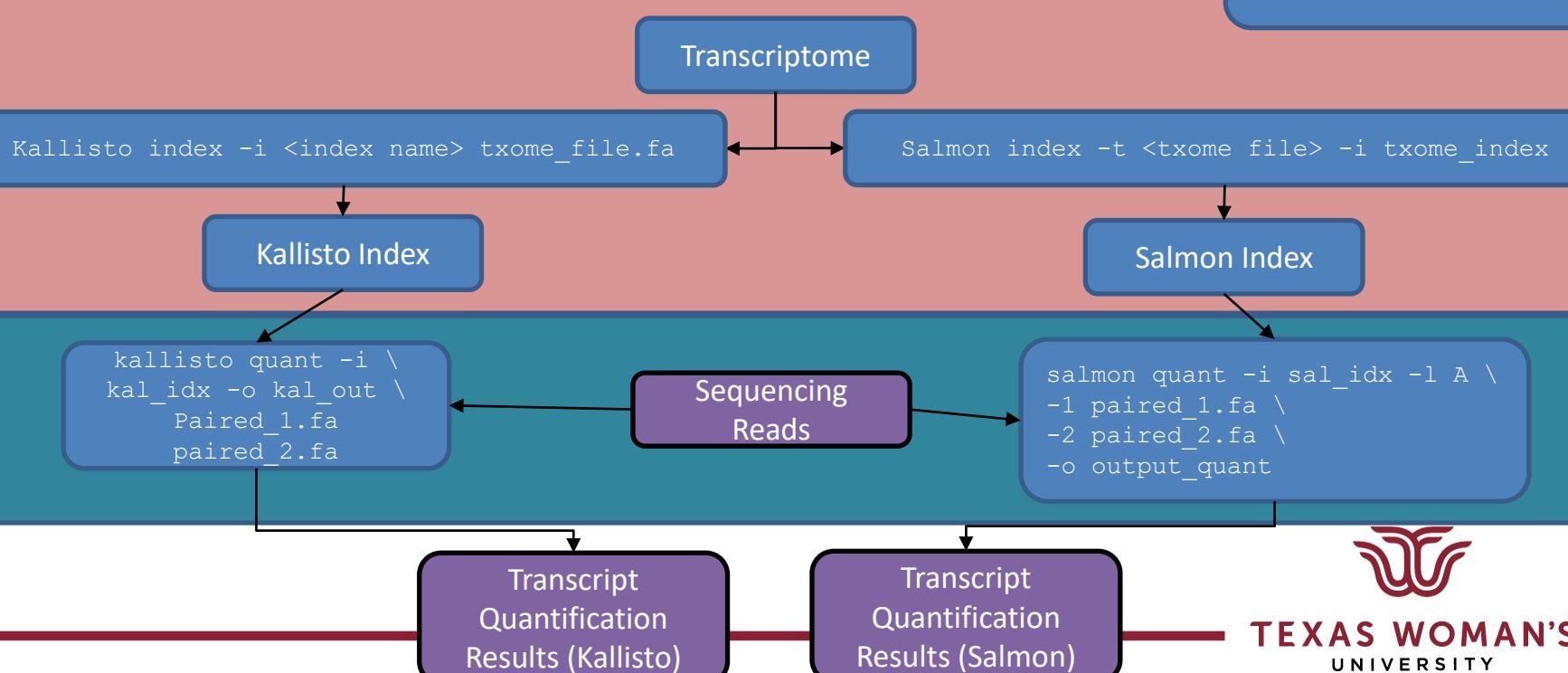
Commands:

index	: create a salmon index
quant	: quantify a sample
alevin	: single cell analysis
swim	: perform super-secret operation
quantmerge	: merge multiple quantifications into a single file



Genetic Transcript Abundance Software Kallisto & Salmon

- As we can see, Salmon and Kallisto have many options which are subdivided into sub-commands, to access the help for a subcommand you can use Kallisto <CMD>, or Salmon <CMD> -h
- The basic flow of using the tools is as follows:



TEXAS WOMAN'S
UNIVERSITY

Genetic Transcript Abundance Software Kallisto & Salmon

- Example Transcriptome file (Note Allele-Specific Names)

```
(base) micah@sw525709:~/projects/h2q/h2q/h2q/simulator$ head c22.8.419.txome.fa -n 10
>ENST00000521832
TCACCTTGTCTGATTCTCACCCACGATCATTTGCTGCAGGTTCCCTCTCTCAGCTTGCCTTATTCACTGCACATCTGAGAAATGCTATAAGACTCCCTGTACCCCAA
GGCTGCATTGTTGGAGGCTGTACATATTCTGTACAGCAGAAGTCACGGAGCCTCCCGATCTCTGAGATACTACTCAGACTCAAGTAAGCACCAGGACTCTGGGTCCCAGTCACTTCTGTG
AAAGATCCCTCGGGCAATGTGAGGATTCTGCACATTCTGAGCAGGCTGAGATCAAGTCGACTATTACTATTTACATATCACAGAACAGTGGCACTTCACTGTGCTCCAGACTACGGGAAGTG
TAGAACCTCCCTGCATCTCTGCTTGTGCGAGCAACAAATACACTGTCTGGG
>ENST00000521832 alt
TCACCTTGTCTGATTCTCACCCACGATCATTTGCTGCAGGTTCCCTCTCTCAGCTTGCCTTATTCACTGCACATCTGAGAAATGCTATAAGACTCCCTGTACCCCAA
GGCTGCATTGTTGGAGGCTGTACATATTCTGTACAGCAGAAGTCACGGAGCCTCCCGATCTCTGAGATACTACTCAGACTCAAGTAAGCACCAGGACTCTGGGTCCCAGTCACTTCTGTG
AAAGATCCCTCGGGCAATGTGAGGATTCTGCACATTCTGAGCAGGCTGAGATCAAGTCGACTATTACTATTTACATATCACAGAACAGTGGCACTTCACTGTGCTCCAGACTACGGGAAGTG
TAGAACCTCCCTGCATCTCTGCTTGTGCGAGCAACAAATACACTGTCTGGG
>ENST00000627251
GCCAGGGCGTGGTGGGGGGCACCTGAAATCCAGCTACTGGGAGGCTGAGGCAGGAGAACATCTGAACCCATGAGGCGGAGGTTGCAGTGAGCCGAGATTACGCCATTGCACTCCAGCCGGGTGACAGT
GACTCCATAAAAAAAAAAAATAAAAAAATAAATAATTTAAACAAATTAAAAATGGGATTTTTTGAGACAGAGTCTCACTGTGCCCCAGGCTGGAGTGCACTGGCATGATGATG
CTCACTGCAACCTCCGCCCTGGGTCAAGTGATTCTCTGCCCTCA
>ENST00000627251 alt
GCCAGGGCGTGGTGGGGGGCACCTGAAATCCAGCTACTGGGAGGCTGAGGCAGGAGAACATCTGAACCCATGAGGCGGAGGTTGCAGTGAGCCGAGATTACGCCATTGCACTCCAGCCGGGTGACAGT
GACTCCATAAAAAAAAAAAATAAAAAAATAAATAATTTAAACAAATTAAAAATGGGATTTTTTGAGACAGAGTCTCACTGTGCCCCAGGCTGGAGTGCACTGGCATGATG
CTCACTGCAACCTCCGCCCTGGGTCAAGTGATTCTCTGCCCTCA
>ENST00000407835
ACTACCCAGACTGCCCTGCTTCAGGCATCGAGGACTCGGGCGGAACTAAGCTACAATCCTCTCAAACCACACAAAAACCTATTGCAAACGTAATGCTGCCCTATTGAAAGAAAGGAAGGCTGGG
AAGAAGCTATTGGTTGGTGTACCTGGACCAATCGGAGGAGCGTGTATTGGCGGGAGCTTGCACCGCCCGGGCTTGTACCTCAGCCGAGCAGCAGGCTCCGCCCGTGGCATGTTCTGTG
TTCCGGCAAAAGAGTTCTACGAGGTGGTCCAGGCCAGGGGCTCTCTCTCTGGTGGAGCTGCTCTGTCAGGCTGAGATCCTTCAGGCTTGTCCAGTGACCTGCAATATACG
TCCAGTTCTGGTGGCAAGAACACTTGCATTCTGAGCATAAAGAACAGTTCTATTCTCATAAACTGTGAGGCTAATGGACCTATTGGATATTCTCAACCTGTGAAAGACACTG
CTTGTGTTGACACCCATAGGCCAGTCAATGCTCAATGTATACAAGCATCCAGATCAAATTACTCATAAACAAAGATGATGACTCTGAAGTTCCGCCCTATGAAGACATCTCAGGGATGAAGAGG
TGAAGAGCATTCAAGGAAATGACAGTGATGGTCAGAGCTTCTGAGAACGCGCACAGGTTAGAACAGGAGATGTTGGAGCAACCATGCGGAGGAGCAGGGCAGAGTGGGAGGCCGGAGAACAGAC
CTTGTGACTCAGGACATGATGAAATCATGGACATCTGTCAGGCATGTTGGATGCTGCTGGAGCTGCTGCCAGGACCTGAATGACATGCTGTTGGGAGCTGTTGACTAACAGACAGTGG
AGACAGATCACTAAAGAACATGGACTGATGTTGGTGTCTCTGGTCTCTGAGGCCAGCTTCCCGGCCAACCCAGGAAACACCTGGGAGGAGAACACACTCTGGGAGCTGCAACAGGATCTCCTTGAGT
CCTCCGGCTTGGTCTCTACCGACACTGGTCCCTCCATGACAGCCTGTCACACAGCTATACCGCAGGCCAGGTTCAAGCTGTTGGTCTGTCATGGACAGAAGGGCTCCAGGAGTTCTTGAGACATGG
TCCCTGAAAGCAGGTGAGCAGAACAGTCCAGGCCATGGACATCTCTTGAAGGAGAATTGCGGGAAATGATTGAAGAGTCTGCAAAATAAATTGGGATGAAGGACATGCGCTGCGACACTTCAGCATG
TGGGTTCAAGCACAAGTTCTGGCCAGCGACGTGTTCTTGCACCATGTCATTGATGGAGAGCCCGAGAACGGATGCTCAGGGACAGATCACTTCATCCAGGCTCTCCAGGAGTAAC
CAAGGTGTAACCATGGGCTGGAACTCGCCAAGAAGCAAGTGGCAGGCCAACCCAGCAGAACCATGGCAGCTGCCATTGCAACACCTCGTCATCTCCAGGGGCCCTTCTGACTGCTCTCATGGAGGGCG
```

The transcriptome file can quickly become very large when dealing with many different transcripts of genes (and possibly different allele-specific versions of the same transcript).



TEXAS WOMAN'S
UNIVERSITY

Produce by internal Python Simulator (just inserts mutations):

Python hisat2_simulate_reads -f ref.fa -g ref.gtf -s ref.snp -o ref.txome.fa

Genetic Transcript Abundance Software Kallisto & Salmon

- From the transcriptome FASTA, we can use Kallisto & Salmon to produce their respective indexes (Colored De Bruijn graph for Kallisto & K-mer table + Suffix Array for Salmon)

Produce Index from

```
build] loading fasta file ..../c22.425.txome.fa  
build] k-mer length: 31  
build] counting k-mers ... done.  
build] building target de Bruijn graph ... done  
build] creating equivalence classes ... done  
build] target de Bruijn graph has 22 contigs and contains 2166 k-mers
```

```
Max Junction ID: 61
seen.size():497 kmerInfo.size():62
approximateContigTotalLength: 1748
counters for complex kmers:
[prec>1 & succ>1]=0 | (succ<1 & isStart)=0 | (prec<1 & isEnd)=0 | (isStart & isEnd)=0
contig count: 23 element count: 2026 complex nodes: 0
# of ones in rank vector: 22
[2022-06-27 11:49:10.477] [puff::index::jointLog] [info] Starting the Pufferfish indexing by reading the GFA binary file.
[2022-06-27 11:49:10.478] [puff::index::jointLog] [info] Setting the index/BinaryGfa directory sal_idx
size = 2826

| Loading contigs | Time = 8.6302 ms
-----
size = 2826

| Loading contig boundaries | Time = 5.9665 ms
-----
Number of ones: 22
Number of ones per inventory item: 512
Inventory entries filled: 1
o2
[2022-06-27 11:49:10.506] [puff::index::jointLog] [info] Done wrapping the rank vector with a rank9sel structure.
[2022-06-27 11:49:10.507] [puff::index::jointLog] [info] contig count for validation: 22
[2022-06-27 11:49:10.510] [puff::index::jointLog] [info] Total # of Contigs : 22
[2022-06-27 11:49:10.511] [puff::index::jointLog] [info] Total # of numerical Contigs : 22
[2022-06-27 11:49:10.512] [puff::index::jointLog] [info] Total # of contig vec entries: 46
[2022-06-27 11:49:10.513] [puff::index::jointLog] [info] bits used for posvec entry 6
[2022-06-27 11:49:10.518] [puff::index::jointLog] [info] bits used for constraining the contig vector. 23
[2022-06-27 11:49:10.523] [puff::index::jointLog] [info] # segments = 22
[2022-06-27 11:49:10.524] [puff::index::jointLog] [info] total length = 2,826
[2022-06-27 11:49:10.525] [puff::index::jointLog] [info] Reading the reference files ...
[2022-06-27 11:49:10.534] [puff::index::jointLog] [info] positional integer width = 12
[2022-06-27 11:49:10.534] [puff::index::jointLog] [info] seqSize = 2,826
[2022-06-27 11:49:10.535] [puff::index::jointLog] [info] rankSize = 2,826
[2022-06-27 11:49:10.536] [puff::index::jointLog] [info] num keys = 2,166
[2022-06-27 11:49:10.537] [puff::index::jointLog] [info] edgeVecSize = 0
[2022-06-27 11:49:10.538] [puff::index::jointLog] [info] num keys = 2,166
for info, total work write each : 2.331 total work iworm from level 3 : 4.322 total work raw : 25.000
Building BooPHF1 100 % elapsed: 0 min 0 sec remaining: 0 min 0 sec
Bitarray 17472 bits (100.00%) (array + ranks)
final hash 0 bits (0.00%) (nb in final hash 0)
[2022-06-27 11:49:10.676] [puff::index::jointLog] [info] mphf size = 0.00208282 MB
[2022-06-27 11:49:10.677] [puff::index::jointLog] [info] chunk size = 2,826
[2022-06-27 11:49:10.678] [puff::index::jointLog] [info] chunk 0 = [0, 2,795]
[2022-06-27 11:49:10.679] [puff::index::jointLog] [info] finished populating pos vector
[2022-06-27 11:49:10.680] [puff::index::jointLog] [info] writing index components
[2022-06-27 11:49:10.690] [puff::index::jointLog] [info] finished writing dense pufferfish index
[2022-06-27 11:49:10.702] [log] [info] done building index
```

TEXAS WOMAN'S
UNIVERSITY

Genetic Transcript Abundance Software Kallisto & Salmon

- Run the pseudo-alignment and quantification procedures using Kallisto and Salmon to produce the results (quantification of abundances).

Paired end Read

```
(base) micah@sw525709:~/projects/h2q/h2q/h2q/nanocourse/kallisto$ head ..../c22.425_1.fa -n 10
>1
GTCGGAGATCTTCCAAATAGYLTATCAGATTCGGGTACAGATGGGTACACACTGAGGCCACCTTATAAGTATATCAAATCCCTCAATG
>2
ATGCAGTTGCACATAAACAGGTCACCAAAGGTAAGACAAATGAGATGTCAGTTGGGTGCTCCATCTCCCTCCCGATCTGATCAGTGACAGCCA
>3
ATCGACGAAGCTTAGCAGACGAGCCTAAGCAGCCCTAACAGTGGTGTGCTGTCAGTGGCTTGTCTTGGTCAGGCCATCTGCTTCTATCTCCGGAT
>4
GCCACCTCAGATAATACCCCTGCCAGTTACTAACAGTGGTACTTCCCACAGATGGAAAACCACAAATCAATTGAGCATCACCTGCTTGGCCA
>5
TAATGCCCTCCCTGTCTCTCTTAAAGGCAATGTTGGGGGGTTGGCTTCAAGGCAATGCCAAAGCCTTCAGCTCATTTCAATTATCTCTTATG
(base) micah@sw525709:~/projects/h2q/h2q/h2q/nanocourse/kallisto$ head ..../c22.425_2.fa -n 10
>1
TGTGAAGGACATTCTGGCTGAATAACAGATTCTAAATGCCGATGTGACTCTACGTAGTGTGCTCACAGCTGATGACCTCATTGATGTTGGGAAGAAC
>2
TCTCTCGAAAGTGTTAAAGGACATACGTTGGAGATGAGGTGTCATTCAAATTTGAAGAAGTGAACCTTCCCTTCCATCTGGCGGACAACT
>3
TACCGCGGAACTCTCTCGCGTAATTCTGGAGTACCGCGCTGGTGCAGTAGCCCTGGTGCGGTGCGAGTTTGGCCCGGGTGTGAGGGAGAAC
>4
CGGAGATAGAAGCAGAGATGGCTGGACTCAAAGAACAGGCCACAGCACACCACTTGGGCTGCTTGGCTAACGCTGTCAGAAGT
>5
AGATACAAAGGTGCCAACATCACGCTCTGGATCTCCAGGTATCATTGAAGGTGCCAGGATGGAAAGGTAGAGGTGTCAGTCATTGAGTCAGTGGCC
```

Kallisto Transcript Quantification

```
[quant] fragment length distribution will be estimated from the data
[index] k-mer length: 31
[index] number of targets: 6
[index] number of k-mers: 2,166
[index] number of equivalence classes: 14
[quant] running in paired-end mode
[quant] will process pair 1: ..../c22.425_1.fa
                  ..../c22.425_2.fa
[quant] finding pseudoalignments for the reads ... done
[quant] processed 100,000 reads, 100,000 reads pseudoaligned
[quant] estimated average fragment length: 250
[em] quantifying the abundances ... done
[em] the Expectation-Maximization algorithm ran for 52 rounds
```

```
(base) micah@sw525709:~/projects/h2q/h2q/h2q/nanocourse/kallisto$ salmon quant -l 1 -i sal_idx --> sal_count -c ..../c22.425_1.fa -r ..../c22.425_2.fa
[base] 2022-06-27 12:09:45.227 [jointlog] INFO  Salmon quantifier version 0.8.1
[base] 2022-06-27 12:09:45.227 [jointlog] INFO  This is the latest version of salmon with important bug fixes and improvements is available. ****
[base] 2022-06-27 12:09:45.227 [jointlog] INFO  The newest version, available at https://github.com/COMBINE-lab/salmon/releases
[base] 2022-06-27 12:09:45.227 [jointlog] INFO  contains new features, improvements, and bug fixes, please upgrade at your
[base] 2022-06-27 12:09:45.227 [jointlog] INFO  earliest convenience.
[base] 2022-06-27 12:09:45.227 [jointlog] INFO  Sign up for the salmon mailing list to hear about new versions, features and updates at:
[base] 2022-06-27 12:09:45.227 [jointlog] INFO  https://oceanigenomics.com/subscribe
[base] 2022-06-27 12:09:45.227 [jointlog] INFO  *** salmon (selective-alignment-based) v1.6.0
[base] 2022-06-27 12:09:45.227 [jointlog] INFO  * command line arguments
[base] 2022-06-27 12:09:45.227 [jointlog] INFO  | command | => quant
[base] 2022-06-27 12:09:45.227 [jointlog] INFO  | libtype | => (A )
[base] 2022-06-27 12:09:45.227 [jointlog] INFO  |   command | => (A )
[base] 2022-06-27 12:09:45.227 [jointlog] INFO  |   output | => (sal_count)
[base] 2022-06-27 12:09:45.227 [jointlog] INFO  |   matesel | => (..../c22.425_1.fa)
[base] 2022-06-27 12:09:45.227 [jointlog] INFO  |   reads | => (..../c22.425_2.fa)
[base] 2022-06-27 12:09:45.227 [jointlog] INFO  logs will be written to sal_count/logs
[base] 2022-06-27 12:09:45.227 [jointlog] INFO  2022-06-27 12:09:45.227 [jointlog] INFO  setting maxNbasesPerSiteThreads to 1
[base] 2022-06-27 12:09:45.227 [jointlog] INFO  2022-06-27 12:09:45.227 [jointlog] INFO  setting minScoreFraction below threshold. Incompatible fragments will be ignored.
[base] 2022-06-27 12:09:45.227 [jointlog] INFO  2022-06-27 12:09:45.227 [jointlog] INFO  Usage of --validateMappings implies use of minScoreFraction. Since not explicitly specified, it is being set to 0.65
[base] 2022-06-27 12:09:45.227 [jointlog] INFO  2022-06-27 12:09:45.227 [jointlog] INFO  setting consensusSlack to selective-alignment default of 0.35.
[base] 2022-06-27 12:09:45.227 [jointlog] INFO  2022-06-27 12:09:45.227 [jointlog] INFO  setting logFormat to logfmt by default
[base] 2022-06-27 12:09:45.227 [jointlog] INFO  2022-06-27 12:09:45.227 [jointlog] INFO  library is in library.
[base] 2022-06-27 12:09:45.227 [jointlog] INFO  2022-06-27 12:09:45.227 [jointlog] INFO  Loading pufferfish index.
[base] 2022-06-27 12:09:45.227 [jointlog] INFO  2022-06-27 12:09:45.227 [jointlog] INFO  Loading dense pufferfish index.

| Loading config table | Time = 3.7397 ms
size = 23
Loading config offsets | Time = 6.4059 ms
Loading reference lengths | Time = 77.4 us
Loading mphf table | Time = 2.9421 ms
```

Salmon Transcript Quantification



Genetic Transcript Abundance Results

```
(base) micah@sw525709:~/projects/h2q/h2q/h2q/nanocourse$ cat c22.425.stat  
ENSG00000185721 ENST00000331457_ref 14996  
ENSG00000185721 ENST00000331457_alt 14996 snps  
ENSG00000185721 ENST00000416465_ref 6527  
ENSG00000185721 ENST00000416465_alt 6527 snps  
ENSG00000185721 ENST00000433341_ref 7757  
ENSG00000185721 ENST00000433341_alt 7757 snps True Simulation  
ENSG00000185721 ENST00000486584_ref 5709  
ENSG00000185721 ENST00000486584_alt 5709 snps Positions  
ENSG00000185721 ENST00000469673_ref 5118  
ENSG00000185721 ENST00000469673_alt 5117 snps  
ENSG00000185721 ENST00000548143_ref 9894  
ENSG00000185721 ENST00000548143_alt 9893 snps
```

Reads were simulated from allele-specific (randomly mutated) versions of 6 transcripts of gene ENSG00000185721

```
(base) micah@SW525709:~/projects/h2q/h2q/h2q/nanocourse/salmon$ cd sal_count/  
(base) micah@SW525709:~/projects/h2q/h2q/h2q/nanocourse/salmon/sal_count$ ls  
aux_info cmd_info.json libParams lib_format_counts.json logs quant.sf  
(base) micah@SW525709:~/projects/h2q/h2q/h2q/nanocourse/salmon/sal_count$ cat quant.sf  
Name Length EffectiveLength TPM NumReads  
ENST00000331457 1746 1495.297 39821.114702 30031.986  
ENST00000416465 612 361.297 71683.353315 13062.473  
ENST00000433341 808 557.297 54808.844438 15405.660  
ENST00000486584 318 67.297 336398.521190 11418.000 Salmon Transcript  
ENST00000469673 677 426.297 47881.328053 10294.880  
ENST00000548143 338 87.297 449406.838302 19787.000 Results
```

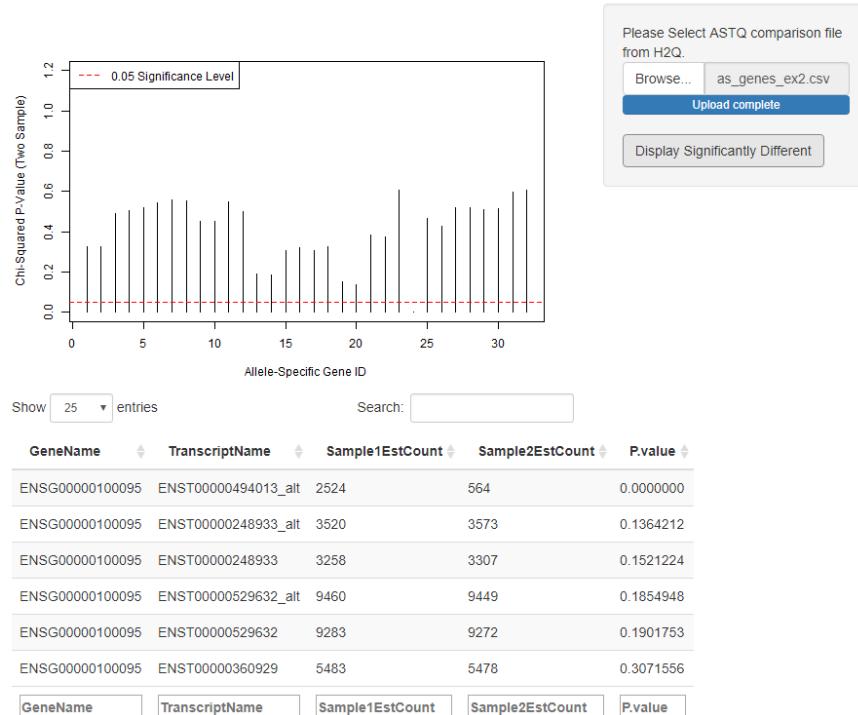
Since a non-allele specific transcriptome was used, Salmon and Kallisto cannot provide more specific quantification of these results than at the transcript level

```
(base) micah@SW525709:~/projects/h2q/h2q/h2q/nanocourse/kallisto$ cd kal_out/  
(base) micah@SW525709:~/projects/h2q/h2q/h2q/nanocourse/kallisto/kal_out$ ls  
abundance.h5 abundance.tsv run_info.json  
(base) micah@SW525709:~/projects/h2q/h2q/h2q/nanocourse/kallisto/kal_out$ cat abundance.  
cat: abundance.: No such file or directory  
(base) micah@SW525709:~/projects/h2q/h2q/h2q/nanocourse/kallisto/kal_out$ cat abundance.tsv  
target_id length eff_length est_counts tpm  
ENST00000331457 1746 1497 30079.9 40686.8  
ENST00000416465 612 363 11467.7 63968.9  
ENST00000433341 808 559 16977.1 61496.8  
ENST00000486584 318 69 11418 335074 Kallisto Transcript  
ENST00000469673 677 428 10270.3 48589.3  
ENST00000548143 338 89 19787 450184 Results
```



Genetic Transcript (Allele-Specific Abundance Results with H2Q ~ Teaser)

Allele-Specific Gene Transcript Quantification Viewer



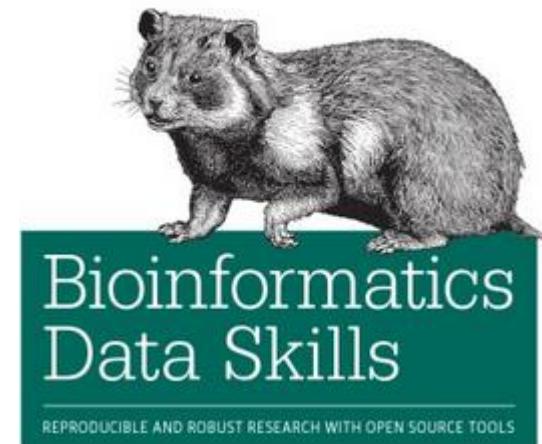
- By using the graph-alignment procedures of HISAT2, we are able to produce exact alignments about as quickly as Kallisto and Salmon produce Pseudoalignments.
- This also allows us to identify allelic markers more easily (without rewriting the allelic variants into a transcriptome ahead of time).



Recommended Resources

- Bioinformatics Data Skills
 - O'Reilly Library
 - By Vince Buffalo
 - Published 2015

O'REILLY®



TEXAS WOMAN'S
UNIVERSITY