Establishing Data Analysis Pipeline for Bulk ATAC-Seq Datasets

January 2026

Owen Leiser
Emma Carlson
Javier Flores
Andy Lin
Tony Chiang
Amy Sims

DISCLAIMER

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor Battelle Memorial Institute, nor any of their employees, makes **any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights**. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof, or Battelle Memorial Institute. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

Establishing Data Analysis Pipeline for Bulk ATAC-Seq Datasets

January 2026

Owen Leiser
Emma Carlson
Javier Flores
Andy Lin
Tony Chiang
Amy Sims

Acknowledgments

Contents

Figures

Tables

1.0   Introduction

We developed an analysis pipeline for transposase-accessible chromatin sequencing (ATAC-Seq) data derived from bulk samples, which brings together publicly available R packages in addition to command-line tools designed for analysis of bulk ATAC-Seq data and can be run on any computer running a Linux-like operating system such as Ubuntu or Apple OSX (Figure 1 and detailed below). For smaller input read files (~500 MB – 1 GB) the pipeline can be run on a reasonably powerful Apple laptop (M1; 8 core, 32 GB RAM), making it usable by a wide range of users. However, more powerful computing resources such as are available through PNNL High Performance Computing (HPC e.g., the Deception cluster; 64 cores, 256 GB RAM per node) are necessary for larger input sequence read files (> 1 GB), as use of the pipeline will require both increases in memory/processing speed and parallel processing abilities.

Process and Align Reads
- Remove low quality sequencing reads
- Align to human reference genome with ATAC-specific parameters

Process Alignment Files
- Remove duplicate alignments and alignments to mitochondria
- Check mapping statistics and return properly paired read alignments

Call Open Chromatin
- Call peaks, filter blacklist regions
- Annotate peaks to gene regions/TSS
- Count peaks

Differential Analysis
- Pairwise statistical analysis between sample groups
- Output analysis, visualize group data

Figure 1: Overview of ATAC-Seq analysis pipeline

1.1   Outline of the Pipeline

A detailed step-by-step outline of the pipeline including system requirements and software/dependencies is available in Appendix 1. The steps comprising this pipeline are listed briefly below:

Process and Align Reads
1. Input fastq read file cleanup
   a. Rename files to conform to Illumina naming convention
   b. Sequentially removes sequencing adapters, low quality reads, and PhiX control reads

2. Index human genome reference and align reads to reference
   a. This step is performed with ATAC-Seq specific parameters

Process Alignment Files
3. Remove duplicate alignments derived from PCR amplification during library preparation
4. Remove mitochondrial reads, which are uninformative to ATAC-Seq analysis
5. Check mapping and other quality control metrics
6. Filter alignment files to only include uniquely mapping reads and proper read pairs

Call Open Chromatin
7. Call ATAC peaks
8. Filter peaks to exclude calls in regions prone to inaccurate mapping
9. Remove redundant peak calls
10. Count reads aligned within each peak

Differential Analysis
11. Perform differential analysis on read counts on a per-peak basis between sample types

We tested this pipeline using selected read files from two publicly available datasets [1, 2] and generated various output streams suitable for downstream biological analysis, including:
- Quality metrics with mapping statistics across chromosomes and distribution of sequence library insert sizes across nucleosome states
- Annotation of gene regions comprising open chromatin regions based on known transcription start sites
- Profiles of open chromatin regions by sample across the entire chromosome
- Gene ontology enrichment of identified open chromatin regions for cellular components, biological processes, and molecular function
- Differential analysis of annotated gene regions between pairs of sample groups

2.0 Analysis of Test Data and Experimental Design

The total number of reads mapping to each identified open chromatin region ("peaks") for all test samples were used as input for a power analysis to inform our experimental design. Power analyses were based on a two-sample t-test applied to log2-transformed data. A type 1 error of 5% was assumed, and the variance levels used to generate each power/sample size curve in Figure 2 were based on median (standard deviation (sd) = 0.751), third-quartile (sd = 0.977), and maximum (sd = 2.846) values observed across chromatin regions in the test data. Use of these higher quartiles for estimates of data variability provided a conservative estimate for the number of samples required to achieve a desired power level. Power analysis indicated that these data allow the identification of low-level (two-fold) changes in peak abundance between cell types/treatments using a reasonable number of experimental replicates – in the case of our test data, as few as 3-4 replicates (Figure 2).
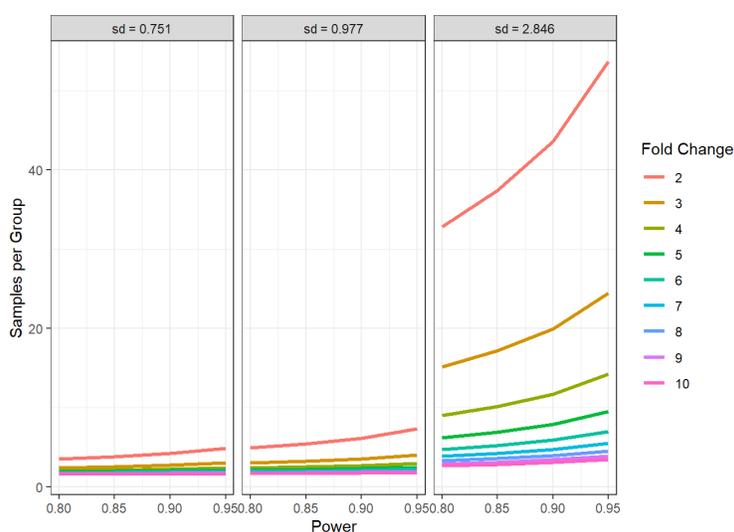
Figure 2: Results from example power analysis performed on test data, modeling three different levels of data variance

To determine how viral infection alters host cell chromatin accessibility, immortalized human cells were treated with media alone, ultra-violet light (UV) inactivated viral particles or replication competent human coronavirus 229E (HCoV-229E) for 24 hours prior to ATAC-Seq analysis. For each condition, ten replicates were harvested to ensure robustness against sample loss during sample processing and data analysis. Our experimental setup comprised three sample categories:

- Huh7 cells "mock" infected (base infection media without virus)
- Huh7 cells infected with HCoV-229E
- Huh7 cells infected with UV-treated HCoV-229E (test whether viral replication is required to elicit host phenotypes during infection)

Potential batch effects were mitigated in our experimental design by randomizing the distribution of replicate samples for each treatment group prior to sample barcoding (four sense and four antisense primers were used) and assignment to one of four Illumina HiSeq lanes, such that every combination of primers and sequencing lane contained similar numbers of each sample type.

3.0 Overcoming Technical Hurdles During Sample Sequencing

Infection of host cells, sequence library preparation, and sample submission to a third-party sequencing core (Azenta) were completed within 2 months. During initial quality control by Azenta technicians, it was discovered that our sequencing libraries had an average fragment size greater than the expected size distribution. Ideally, the fragment size distribution for an ATAC-Seq sequencing library peaks at around 150-300 bases, with a rapidly decreasing tail to around 800 bases. However, our libraries had size distributions peaking between 650-800 bases (Figure 3). After consultation with Azenta and colleagues who have experience performing ATAC-Seq, we determined the cause of the abnormal distribution was most likely to be undertagmentation during sample preparation. Tagmentation is the process by which hyperactive Tn5 transposase fragments bind sequencing adapters to regions of open chromatin, and undertagmentation can result from several factors including insufficient transposase-chromatin incubation time or excess cell death during sample preparation. As it was not feasible to repeat the experiment, Azenta performed a size selection step to enrich for appropriately sized libraries prior to sequencing. The end goal was to obtain 50 million paired-end reads per sample.

Upon completion of sequencing Azenta quality control identified several samples with sequencing files that contained either abnormally high or low numbers of reads. One each from mock infected and UV-treated virus infected samples had roughly double the target number of reads, while four HCoV-229E infected samples contained far fewer reads than expected – between 40,000 and 17 million reads. It is likely that the root cause of this unevenness stemmed from a reduction in total library quantity during the required size selection, with low-abundance libraries experiencing low clustering efficiency during sequencing. These six samples were excluded from the data analysis pipeline, resulting in a total of nine replicates each for mock infected and UV-treated virus, and six replicates for infected samples. Despite this setback, our initial power analyses supported continuing with the analysis as the remaining number of replicates per group were sufficient to ensure adequately powered (>80%) analyses.

To process the substantially higher number of reads in our experimental data, the pipeline was ported to PNNL HPC resources, as the computers used to design and test the pipeline using publicly available data were insufficient (in terms of memory, parallel processing, and storage capabilities). This change allowed us to process the bulk sample ATAC-Seq data in far less time than it would have taken on even the most powerful of standalone computers available for our use.

4.0 Initial Analysis

Several informative metrics of data quality are generated during the initial steps of our ATAC-Seq data processing pipeline. Prior to mapping reads to reference sequence, the distribution of fragment sizes is plotted. Ideally, fragment distribution follows the pattern show in the inset of Figure 3 (derived from test data [1]), with pronounced peaks corresponding to nucleosome-free regions, followed by mononucleosome, dinucleosome, etc. The size distribution of our experimental library fragments was not ideal, with most fragments corresponding to mononucleosome-bound chromatin. Nevertheless, we observed enough fragments corresponding to open chromatin (<150 bases) to proceed (Figure 3).
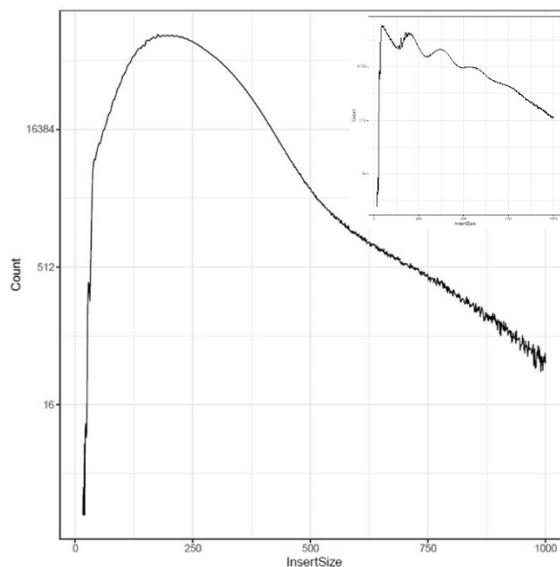
Figure 3: Example visualization of fragment size distribution. Inset, ideal size distribution from test data showing peaks indicative of open chromatin, followed by mononucleosome-bound chromatin, dinucleosome, etc.

After mapping to the human reference chromosome GRCh38 using Bowtie2 [3], four additional quality metrics were generated: library complexity, mapping distribution, fraction of reads in peaks, and distribution of peak annotation. Library complexity is determined as a function of distinct fragments relative to the total number of sequenced fragments. Our experimental data show an ideal complexity curve, with an early steep slope tapering off as additional fragments are added (Figure 4).
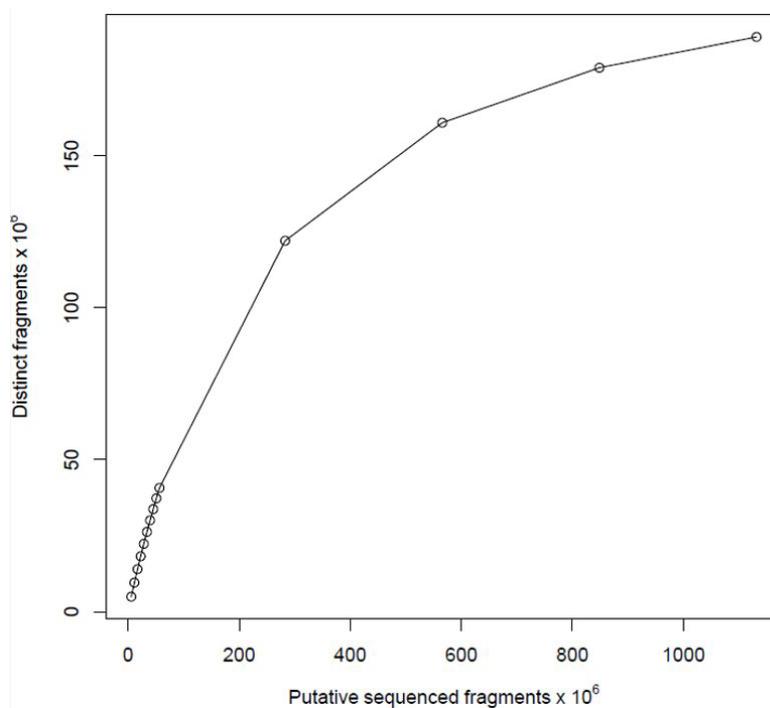
Figure 4: Plot of example library complexity

Distribution of read mapping across 24 chromosomes was similar to test data (Figure 5). Importantly, reads mapping to the mitochondrial chromosome were excluded from downstream analysis, as the nature of this chromosome makes it hyper-available to the Tn5 transposase during tagmentation and therefore not informative to ATAC-Seq. The final quality metric produced by our analysis pipeline is the Fraction of Reads in Peaks (FRIP), which quantifies the number of reads that map to regions of open chromatin after ATAC peaks are identified. An ideal FRIP is above 0.3, with usable but suboptimal FRIP ranges from 0.1-0.29. Our data averaged a FRIP of 0.16, which was lower than ideal and likely related to the fragment size distribution discussed above. We hope that this metric will improve with subsequent experiments as sample collection procedures are optimized.
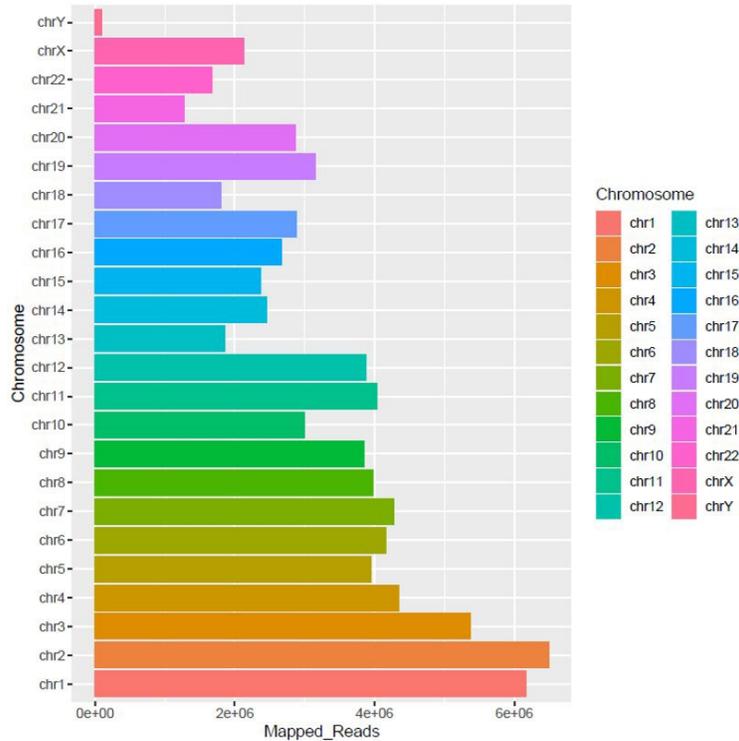
Figure 5: Read mapping distribution across all human chromosomes. Reads mapping to the mitochondrial chromosome were excluded as they are not informative to ATAC-Seq analysis

ATAC peaks identified by our pipeline [4] were annotated based on their proximity to known transcription start sites (TSS) within the human genome, the distribution of which is shown in Figure 6. In typical ATAC-Seq data, around 25% of peaks map to promoter regions less than 1000 nucleotides from TSS and around 50% map to intronic and distal intergenic regions. Our data are similar, though not identical, to typical ATAC-Seq data: peaks mapping to promoter regions accounted for 31.5%, 30.5%, and 35.5% of total peaks in mock infected, UV-treated, and infected samples, respectively, while peaks mapping to intronic and distal intergenic regions accounted for 55.6%, 56.5%, and 52.2% in the same samples. In our hands the annotation distributions are approximately equal across sample treatments, increasing confidence in the comparability of data between these sample types.
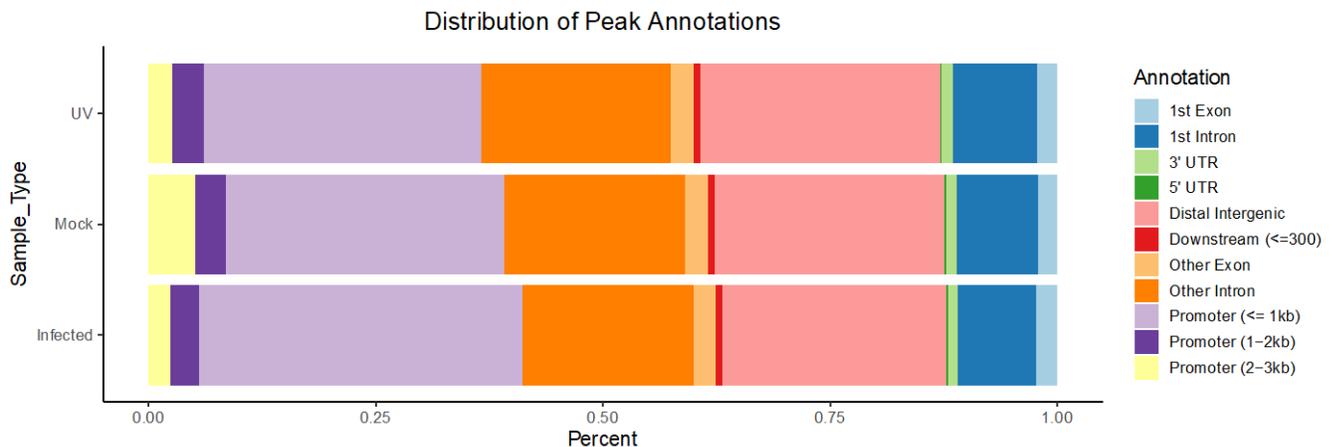
Figure 6: Distribution of ATAC peak annotations relative to transcription start sites

5.0 Differential Analysis

Three comparisons of peak abundance were made between sample treatments using DESeq2, which normalizes peak data to total read count:
- Mock infection vs. infection
- UV-treatment vs. infection
- Mock infection vs. UV treatment

Comparisons were initially made at the peak level, with potentially multiple peaks corresponding to a given coding region/gene, because ATAC-Seq ultimately gives a measure of differentially accessible chromatin, not necessarily differential gene expression. Principal component analysis was conducted using data from these comparisons. This analysis demonstrates a clear global separation between infected cells and both UV-treated and mock infected cells (Figure 7). Additionally, it shows a lack of separation between mock infected and UV-treated cells indicating a requirement for viral replication for eliciting the phenotypes observed in infected cells.
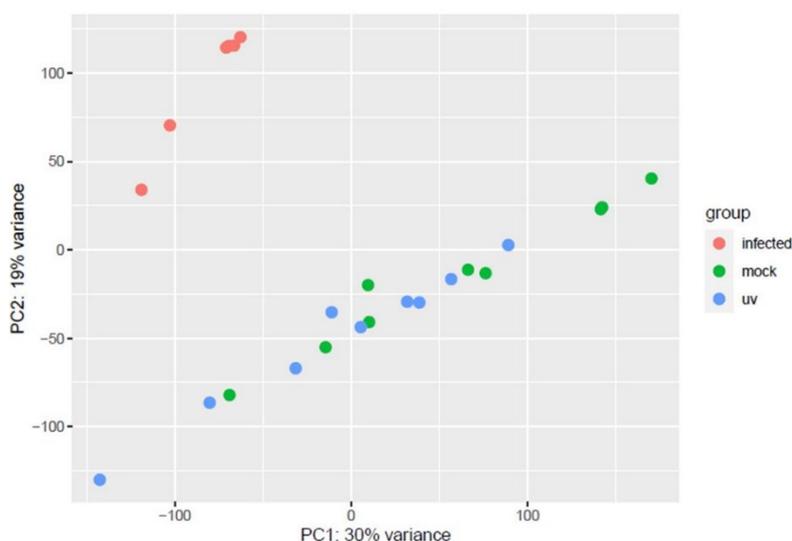
Figure 7: Principal coordinate analysis plot of differentially abundant ATAC peaks

Differential detection analysis was reduced in scope to the three comparisons separately to better quantify the differences in chromatin accessibility. Results from these comparisons are summarized in Table 1 and visualized as volcano plots in Figure 8. Figure 8 does not contain any visualizations for the comparison of mock and UV-treatment as there were no peaks with significantly different abundance between the two conditions, therefore no further analysis of this comparison was undertaken.

Table 1: Summary of ATAC peak differential analysis

COMPARISON	TOTAL NUMBER OF QUANTIFIED PEAKS	TOTAL NUMBER OF SIGNIFICANT PEAKS	SIGNIFICANT PEAKS ±2-FOLD CHANGE	PEAKS WITH INCREASED ABUNDANCE	PEAKS WITH DECREASED ABUNDANCE

MOCK VS. UV	114,207	0	0	0	0
MOCK VS. INFECTED	114,207	15,186	6,089	3,439	2,650
UV VS. INFECTED	114,207	12,053	4,997	1,982	3,015
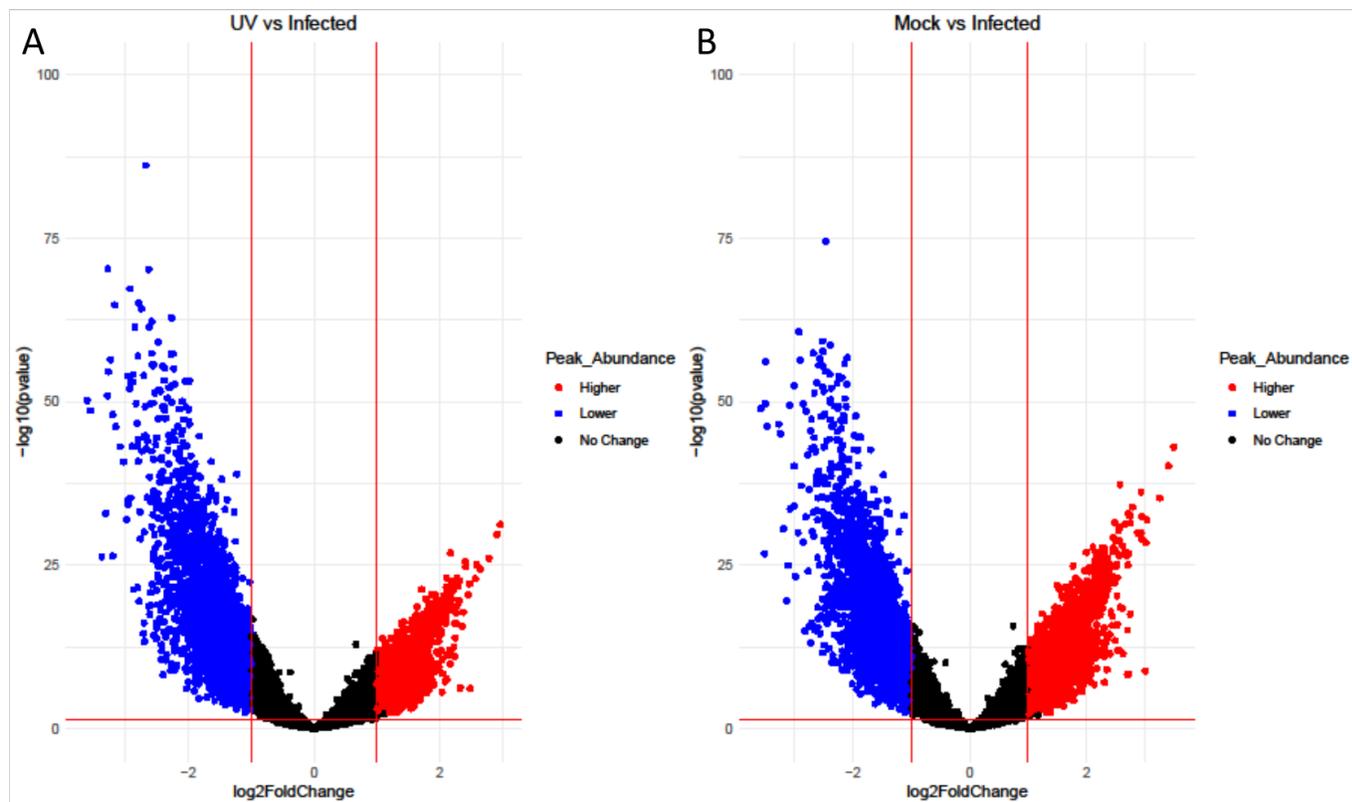
Figure 8: Volcano plots of differentially abundant ATAC peaks for UV vs. Infected (A) and Mock vs. Infected (B). Vertical lines represent 2-fold higher or lower abundance. Horizontal lines indicate p-value cutoff of 0.05.

When analyzing differential detection, we chose to impose an additional cutoff of ±2-fold change in peak abundance to simplify exploration of the data and to adhere to the power analysis performed on test data. In our hands, between-sample variance was lower than in the test data, suggesting we can detect smaller effect sizes at a reasonable power, however in our analysis the more conservative cutoff was used. We compared the differential peak calls from the two conditions (mock infection vs. infection, UV-treatment vs. infection) to determine overlap between the two (Figure 9) and observed a significant overlap in peak abundance between the two treatments, with 67% of differential peak calls being shared. Encouragingly, differentially abundant peaks identified in infected cells common to both UV-treated and mock infection comparisons exhibited the same change, e.g., they were both increased or both decreased. There were no instances of peak abundance changes in the opposite direction between the two conditions.
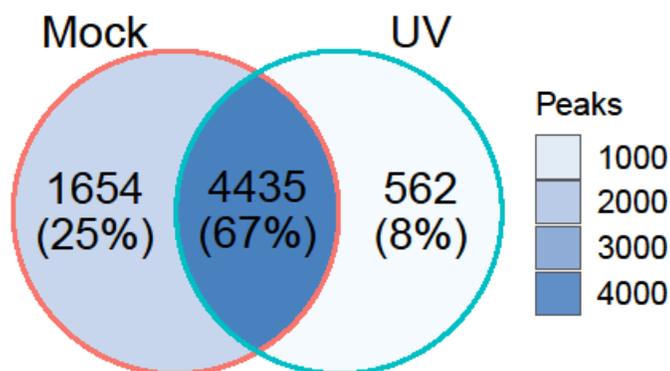
Figure 9: Overlap of differentially abundant ATAC peak calls in infected samples relative to mock infection and UV-treatment.

To assess the biological relevance of differential chromatin accessibility, we subjected significantly different peaks to pathway analysis using the R package PathfindR [5], which uses a clustering algorithm to detect pathway enrichment. Prior to pathway analysis, peaks were annotated to the nearest gene using known TSS locations. PathfindR filters redundant gene calls from analysis, leaving only data from the entry with the lowest p-value for a given gene. This analysis identified a total of 206 enriched pathway terms, with 180 terms shared between UV-treatment and mock infected comparisons (Figure 10). The top 20 enriched pathways for each condition are visualized in Figure 11. A full listing of enriched pathways for both conditions, including genes assigned to those pathways, is available as an embedded object in Appendix 2. Work to establish the biological significance of these enriched pathways as well as the genes making up the pathways is ongoing.
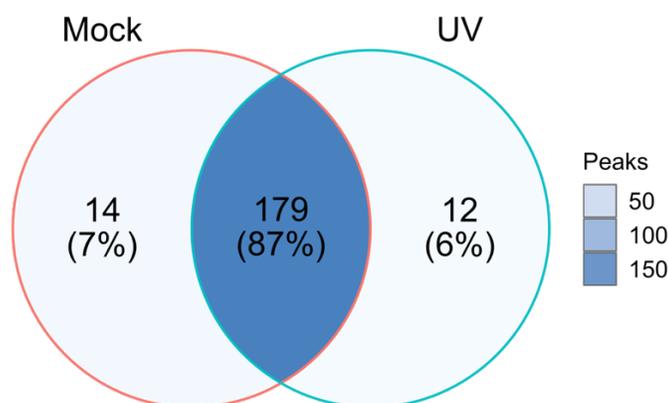
Figure 10: Overlap of enriched pathway terms between mock infection and UV-treatment conditions
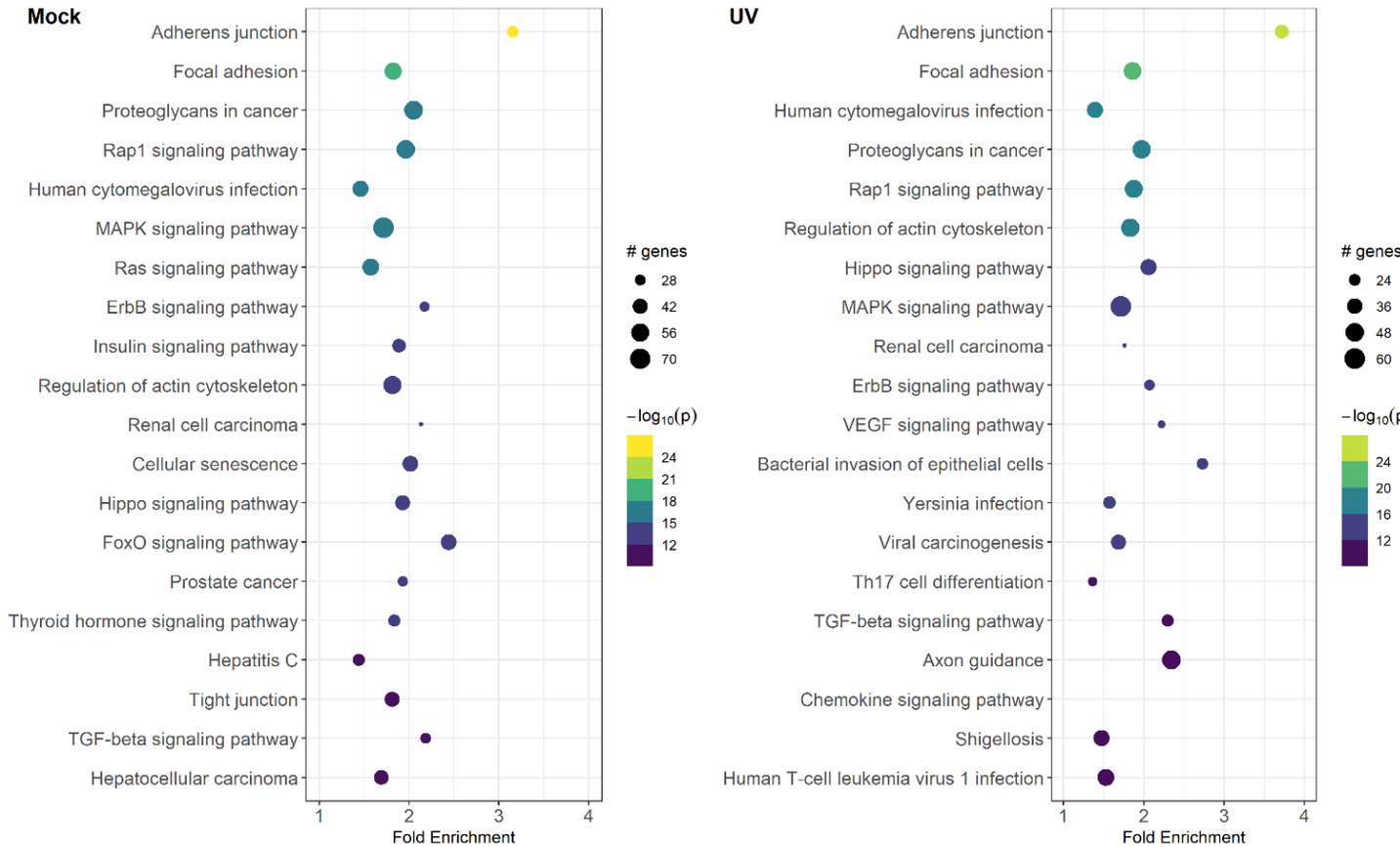
Figure 11: Top 20 enriched pathway terms for HCoV-229E ATAC-Seq data compared to mock infection (A) and UV-treatment (B)

Genes comprising enriched pathways were visualized within PathfindR by mapping them to the Kyoto Encyclopedia of Genes and Genomes (KEGG; [6-8]) database. An example of this mapping, for the pathway of adherens junction formation, is found in Figure 12. Adherens junctions are protein complexes present at the interfaces between cells, and are linked to the actin cytoskeleton [9]. In HCoV-229E infected cells the chromatin regulating adherens junction gene transcription is not accessible (is closed) suggesting that infection reduces cellular adhesion.

Figure 12: Example annotated KEGG pathway diagram, showing genes identified as differentially accessible in mock infection vs. infection analysis

6.0 Preparing for Future Experiments

Evaluating single-cell ATAC-Seq analysis software: In addition to analysis of bulk ATAC-seq data, we began exploring analysis software for single-cell experimental data. Analysis of single-cell data is nontrivial and requires significant computational resources, such as the HPC infrastructure at PNNL. We have identified software purpose-built for processing and alignment of single-cell ATAC sequence data (Cell Ranger 10x Genomics) and have successfully processed ~10 publicly available test datasets of varying sizes using PNNL HPC resources. Additionally, we have identified and explored two software packages designed for analysis of these data. Loupe (10x Genomics) was designed to work seamlessly with Cell Ranger output as a graphical user interface, while ArchR, is an R package that allows users a high degree of customization and granularity for data analysis but requires some knowledge of the R language. We will use Cell Ranger, Loupe, and/or ArchR for future studies with single-cell ATAC-Seq sequencing datasets.

These software packages are also applicable to other experimental techniques which measure relative abundance of nucleic acids, such as single cell RNASeq or 3' expression libraries. Although they are not directly applicable to single cell proteomics analysis such as nanoPOTS (nanodroplet processing in one pot for trace samples, [10]), analysis of these data can be accomplished using standard proteomics pipelines already developed at PNNL i.e., MaxQuant [11, 12]. MaxQuant has already been optimized for use on local and HPC resources. We intend to leverage these PNNL resources for analysis and integration of multi-omics data streams.

7.0 References

1. Corces, M.R., et al., *An improved ATAC-seq protocol reduces background and enables interrogation of frozen tissues.* Nature Methods, 2017. **14**(10): p. 959-+.
2. Xu, W., et al., *A plate-based single-cell ATAC-seq workflow for fast and robust profiling of chromatin accessibility.* Nature Protocols, 2021. **16**(8): p. 4084-4107.
3. Langmead, B. and S.L. Salzberg, *Fast gapped-read alignment with Bowtie 2.* Nature Methods, 2012. **9**(4): p. 357-U54.
4. Zhang, Y., et al., *Model-based Analysis of ChIP-Seq (MACS).* Genome Biology, 2008. **9**(9).
5. Ulgen, E., O. Ozisik, and O.U. Sezerman, *pathfindR: An R Package for Comprehensive Identification of Enriched Pathways in Omics Data Through Active Subnetworks.* Frontiers in Genetics, 2019. **10**.
6. Kanehisa, M., *Toward understanding the origin and evolution of cellular organisms.* Protein Science, 2019. **28**(11): p. 1947-1951.
7. Kanehisa, M., et al., *KEGG: integrating viruses and cellular organisms.* Nucleic Acids Research, 2021. **49**(D1): p. D545-D551.
8. Kanehisa, M. and S. Goto, *KEGG: Kyoto Encyclopedia of Genes and Genomes.* Nucleic Acids Research, 2000. **28**(1): p. 27-30.
9. Meng, W.X. and M. Takeichi, *Adherens Junction: Molecular Architecture and Regulation.* Cold Spring Harbor Perspectives in Biology, 2009. **1**(6).
10. Zhu, Y., et al., *Nanodroplet processing platform for deep and quantitative proteome profiling of 10-100 mammalian cells.* Nature Communications, 2018. **9**.
11. Cox, J. and M. Mann, *MaxQuant enables high peptide identification rates, individualized p.p.b.-range mass accuracies and proteome-wide protein quantification.* Nature Biotechnology, 2008. **26**(12): p. 1367-1372.
12. Tyanova, S., T. Temu, and J. Cox, *The MaxQuant computational platform for mass spectrometry-based shotgun proteomics.* Nature Protocols, 2016. **11**(12): p. 2301-2319.

Appendix A - Detailed Bulk ATAC-Seq Computational Pipeline

A.1   System Requirements

This pipeline is designed to run on a Linux-like operating system such as Ubuntu or OSX. Several programs integral to the pipeline are not available for Windows operating systems. While a desktop/laptop computer with a reasonable amount of memory (>= 16 GB) and hard disk space can be used with small (~100-200 MB) input read files, most read files from current generation high-throughput sequencers require the use of a parallel processing computer, for example a high-performance computing cluster. Output file size scales with input read size. Users should edit steps using multiple threads and/or that specify amounts of memory to be used to reflect the capabilities of their specific computing resources.

A.2   Software and Dependencies Required

The pipeline is contained within two R scripts, atac_bulk_slurm.R and atac_differential.R[1], and should be opened using RStudio[2] running base R version >=4.2[3]. R packages are required for this pipeline are listed in an early code chunk and in a section of this document, below. It is recommended that these be installed one by one: depending on the exact setup of a user computer it may be necessary to install additional libraries or packages. Bash shell should be used for programs called via the system() command.

Several programs and their dependencies are unavailable as R packages and are called using the system bash shell within the pipeline script. These can be downloaded manually by a user or (for most) installed using a package manager such as Homebrew[4] or pip[5].

---

[1] https://stash.pnnl.gov/projects/ATACPILOT/repos/bulk_final_scripts/browse

[2] https://www.rstudio.com/products/rstudio/download/

[3] https://cran.r-project.org/

[4] https://brew.sh/

[5] https://pip.pypa.io/en/stable/

8.0 Software available from Homebrew if installing on a local machine with administrator privileges

- Burrows-Wheeler Aligner (bwa)[1]
- Picard Tools[2]
- Samtools[3] (note that this is duplicated by an R package, rsamtools. Both are required and are called by different programs)

8.1 Software available from pip/conda if installing on a local machine with administrator privileges

- MACS3[4] (note MACS2 is available as an R package and is likely to function properly, but has not been tested for this purpose)
- Bowtie2[5] (note that this is duplicated by an R package, Rbowtie2. Both are required and are called by different programs)
- Bioawk[6]

8.2 Software requiring manual download if installing on a shared machine without administrator privileges

- Bowtie2[7] (note that this is duplicated by an R package, Rbowtie2. Both are required and are called by different programs)
- MACS3[8] (note MACS2 is available as an R package and is likely to function properly, but has not been tested for this purpose)
- Picard Tools[9]
- Samtools[10] (note that this is duplicated by an R package, rsamtools. Both are required and are called by different programs)
- Bioawk[11]

8.3 Software requiring manual download on all systems

These are available as precompiled binary files and do not require compiling. They can be placed in a user's PATH or called using hard links.

- BBTools[12]

---

[1] http://bio-bwa.sourceforge.net/

[2] https://broadinstitute.github.io/picard/

[3] http://www.htslib.org/

[4] https://github.com/macs3-project/MACS

[5] https://sourceforge.net/projects/bowtie-bio/files/bowtie2/

[6] https://github.com/lh3/bioawk

[7] https://sourceforge.net/projects/bowtie-bio/files/bowtie2/

[8] https://github.com/macs3-project/MACS

[9] https://broadinstitute.github.io/picard/

[10] http://www.htslib.org/

[11] https://github.com/lh3/bioawk

[12] https://jgi.doe.gov/data-and-tools/software-tools/bbtools/

- Genome Analysis Toolkit (GATK)[1]
- Integrative Genome Viewer (IGV)[2]

8.4   R packages required (available through base R)

- stringi[3]
- BiocManager[4]
- Tidyverse[5]
- ggplot2 (part of Tidyverse)[6]
- magrittr (part of Tidyverse)[7]
- dplyr (part of Tidyverse)[8]
- DT[9]
- tidyr (part of Tidyverse)[10]
- ggupset[11]
- readxl (part of Tidyverse)[12]
- devtools[13]
- ggrepel[14]
- plyr[15]
- pathfindR[16]
- ggpubr[17]
- ggVennDiagram[18]

8.5   R packages required (available through Bioconductor)

- Rsubread[19]
- Rsamtools[20]

---

[1] https://gatk.broadinstitute.org/hc/en-us

[2] https://software.broadinstitute.org/software/igv/

[3] https://cran.r-project.org/web/packages/stringi/index.html

[4] https://cran.r-project.org/web/packages/BiocManager/vignettes/BiocManager.html

[5] https://www.tidyverse.org/

[6] https://ggplot2.tidyverse.org/

[7] https://magrittr.tidyverse.org/

[8] https://dplyr.tidyverse.org/

[9] https://rstudio.github.io/DT/

[10] https://tidyr.tidyverse.org/

[11] https://cran.r-project.org/web/packages/ggupset/

[12] https://readxl.tidyverse.org/

[13] https://www.r-project.org/nosvn/pandoc/devtools.html

[14] https://cran.r-project.org/web/packages/ggrepel/

[15] https://cran.r-project.org/web/packages/plyr/index.html

[16] https://cran.r-project.org/web/packages/pathfindR/index.html

[17] https://cran.r-project.org/web/packages/ggpubr/index.html

[18] https://cran.r-project.org/web/packages/ggVennDiagram/index.html

[19] https://bioconductor.org/packages/release/bioc/html/Rsubread.html

[20] https://bioconductor.org/packages/release/bioc/html/Rsamtools.html

- GenomicAlignments[1]
- TxDb.Hsapiens.UCSC.hg38.knownGene[2]
- soGGi[3]
- DelayedMatrixStats[4]
- ATACseqQC[5]
- rtracklayer[6]
- ChIPQC[7]
- ChiPseeker[8]
- rGREAT[9]
- limma[10]
- DESeq2[11]
- BSgenome.Hsapiens.UCSC.hg38[12]
- tracktables[13]
- ACME[14]
- Organism.dplyr[15]
- Rbowtie2[16]

8.6   Files required for various processing steps

- Human reference chromosome, available from University of California Santa Cruz (UCSC) Genomics Institute[17]. As of late 2022 hg38/GRCh38 is the most current assembly version. If an updated reference genome is used for mapping, all other steps requiring genome information (e.g., annotation) must also use the same updated reference.
- Blacklist file, containing known problematic regions for ATAC peak calling. Available from ENCODE project[18]

---

[1] https://bioconductor.org/packages/release/bioc/html/GenomicAlignments.html

[2] https://bioconductor.org/packages/release/data/annotation/html/TxDb.Hsapiens.UCSC.hg38.knownGene.html

[3] https://bioconductor.org/packages/release/bioc/html/soGGi.html

[4] https://bioconductor.org/packages/release/bioc/html/DelayedMatrixStats.html

[5] https://bioconductor.org/packages/release/bioc/html/ATACseqQC.html

[6] https://bioconductor.org/packages/release/bioc/html/rtracklayer.html

[7] https://bioconductor.org/packages/release/bioc/html/ChIPQC.html

[8] https://bioconductor.org/packages/release/bioc/html/ChIPseeker.html

[9] https://bioconductor.org/packages/release/bioc/html/rGREAT.html

[10] https://bioconductor.org/packages/release/bioc/html/limma.html

[11] https://bioconductor.org/packages/release/bioc/html/DESeq2.html

[12] https://bioconductor.org/packages/release/data/annotation/html/BSgenome.Hsapiens.UCSC.hg38.html

[13] https://bioconductor.org/packages/release/bioc/html/tracktables.html

[14] https://www.bioconductor.org/packages/release/bioc/html/ACME.html

[15] https://www.bioconductor.org/packages/release/bioc/html/Organism.dplyr.html

[16] https://www.bioconductor.org/packages/release/bioc/html/Rbowtie2.html

[17] https://genome.ucsc.edu/cgi-bin/hgGateway

[18] https://www.encodeproject.org/files/ENCFF356LFX/

9.0 Input data requirements

Sequence reads must be paired-end and should be compressed using gzip. Read names must follow Illumina naming convention, i.e., *_R<1/2>_001.fastq.gz. Names can be changed as a batch using a code chunk if necessary; users can determine if the code needs to be altered to accommodate their specific read names.

Reference sequence chromosomes should follow the UCSC naming convention, e.g., Chr1, Chr2, ChrM, etc. and should be concatenated into a single reference with .fa suffix.

All input files should be placed into the same working directory, and that directory specified in the setup chunk. The pipeline script itself does not need to reside in this directory.

10.0 atac_bulk_slurm.R Pipeline Steps

10.1 Set up environment

Clears R environment, specifies working directory, and loads necessary packages. Best practice for libraries is to create a specific project directory to install and load packages. This helps with version control and mitigates risks associated with package updates from unrelated work.

10.2 Rename input fastq (optional)

Renames reads in _R1/_R2 format to comply with Illumina naming convention. Users should only run this chunk if their reads are not already compliant; the script will run on files ending in 001.fastq.gz but with adverse results. Users may adjust this script if their reads are in another format than _R1/_R2. This chunk may be commented out if read files are already properly named.

10.3 Process input fastq

Sequentially removes sequencing adapters, low quality reads, and PhiX control reads. Users may also calculate average read lengths by uncommenting those lines. This chunk runs through bash, and users must specify the hard path to bbduk.sh (part of BBTools) if it is not in their PATH. Users should also specify the amount of memory to use through the -Xmx flag. Intermediate files are deleted to save disk space, with original and fully processed reads kept.

10.4 Build reference index

Indexes the reference prior to alignment. This step generates several intermediate files that are collapsed into .bt2 reference files upon completion.

10.5 Align reads to reference

Aligns input reads to indexed reference. Options in flags are nonstandard for whole-genome alignment and are specific to bulk ATAC-seq:

--end-to-end requires that entire read aligns from end to end without trimming
--very-sensitive is slower but more accurate
--no-unal suppresses sam records that do not align
--no-mixed only aligns concordant read pairs
--phred33 defines input quality format
-X specifies a maximum fragment for valid PE alignments
--threads defines number of processing threads
-x defines location of reference fasta
-1/2 define reads
-S output sam alignment file

10.6 Remove pseudoreplicated read alignments

Removes duplicate read alignments from output .sam file. These generally arise from PCR duplication during sequencing library preparation and can bias downstream abundance

measurements. This step is performed in bash using GATK, because the R package Rsamtools lacks a critical command (fixmate) required to subsequently flag and remove duplicates.

Alignments are sorted by read names, converted to .bam files, and duplicates are removed.

10.7 Sort and index alignment files

Original output .bam files are sorted again by name and indexed for calculation of library complexity.

10.8 Calculate library complexity

Calculates input sequence library complexity as a function of unique reads relative to total number of reads sequenced. An ideal complexity curve starts out very steep and plateaus once all reads have been examined. This step requires alignments that have not been deduplicated.

10.9 Sort and index deduped

Deduplicated .bam files are sorted again by name and indexed for downstream steps.

10.10 Remove mitochondrial reads

Mitochondrial DNA is packaged differently than nuclear DNA, in such a way as to make it more available to Tn5 during tagmentation and uninformative to ATAC-seq analysis. This step removes any alignments mapping to the mitochondrial chromosome (ChrM). It requires a significant amount of memory (8-11 GB RAM for 2-3 GB input .bam file).

10.11 Check mapping distribution

Removes .bam files containing mitochondrial alignments and generates figures with mapping statistics across remaining chromosomes.

10.12 Return proper pairs

This step ensures that only proper open chromatin regions are identified, without ambiguity from reads mapping to multiple locations on the chromosome. It also visualizes read insert size distribution as a quality check for nucleosome-free, mono- and di-nucleosome regions. If desired, the commented "which" command can be used to subset for specific chromosomes, otherwise the default is all chromosomes.

IRanges refers to the length of the chromosome in question.
mapq score is -10log10 {mapping position is wrong}; 0.99 probability of correct mapping ~mapq20

This step also plots ATAC signal against TSS regions and generates "bam" files, which are in turn used to generate BigWig files for visualization in Integrated Genome Viewer. IGV is useful for visualizing gross changes in ATAC profiles across multiple samples at once. Bam files generated in this step are not suitable for peak calling in the next step - the input *DeDuped_sort_noM files should be used instead as they are properly formatted.

10.13 MACS3 peak calling

Calls open chromatin peaks using MACS3 in bash from deduplicated, sorted/indexed .bam files without mitochondrial alignments. Output narrowPeaks files are used for downstream steps.

10.14 Filter and annotate peaks

Filters peaks called in regions prone to inaccurate and/or duplicate mapping using a predetermined blacklist file[1]. This file can be placed in the working directory or linked from another directory. Users can subset to specific chromosomes if desired.

Peaks are annotated to genes (more accurately, gene regions), annotations are written to file, and metrics are generated. Peaks are then analyzed for gene ontology placement and results are written to file for molecular function, biological process, and cellular component.

---

[1] https://www.encodeproject.org/files/ENCFF356LFX/; current as of June 2022

11.0 atac_differential.R Pipeline Steps

11.1 Identify set of non-redundant peaks

First step in differential analysis. Removes redundant peaks from MACS3 narrowPeak output. Consensus peaks are identified and any peaks corresponding to blacklist regions are removed.

Note, users must manually edit this step such that replicate samples have numbers appended to their names (_1, _2, etc.), and that these names are in the same order as peaks values. Names can differ from read/alignment file names, for example can be human-readable instead of sequentially numbered with laboratory sample IDs. "Group" is the same as names, but without numbers appended.

11.2 Visualize overlap

Optional step to generate Venn diagrams of peak call overlap between input samples.

11.3 Filter peaks and count reads

Removes singleton peaks (i.e., peaks containing a single read alignment), then returns a matrix of alignment counts per peak across all samples. Colnames is the same list as names in the nonredundant peak step.

11.4 Differential analysis

Returns statistical analyses of pairwise comparisons between sample groups. The accessory file coldata.xlsx is required, which contains information necessary for DESeq2 to correctly parse replicate samples and sample groups. Users must edit this file as necessary for their own samples. Sample order in this file must match that of previous lists.

Users can set a minimum number of reads required to consider a peak using the commented lines. Depending on the within-sample group variability, it may be desired to run specific pairs by themselves instead of including high-variability groups as part of the analysis.

Pairwise comparisons are made using the contrast argument of results function. Users must manually specify which comparisons should be made. Users may add or remove copies of this section to perform more or fewer comparisons. Outputs are written to .csv files.

11.5 Generate volcano plots

Visualizes expression differences between conditions as volcano plots. Users can set thresholds for significance, fold change, etc. as desired.

11.6 Map midpoint of peaks to nearest gene

Maps positions of peak midpoints to nearest gene for annotation and pathway enrichment steps. Because most genes comprise multiple ATAC peaks, mapping is usually non-unique on a per-gene basis.

11.7 Compare datasets at pathway level

Obtains all pairwise comparisons of differentially abundant peaks and outputs peak lists based on user input. For example users may choose to identify all differentially abundant peaks common to conditions A and B relative to condition C, differentially abundant peaks in condition A and not condition B relative to condition C, etc.

11.8 Pathway enrichment analysis

Reads differential peak abundance data mapped to nearest genes and outputs pathway enrichments. PathfindR collapses multiple peak calls for a given gene into a single value during input processing. Users may choose to run the wrapper function as a single command, although this severely limits the customization and output options. PathfindR also maps annotated enrichment data to KEGG pathways and outputs color-coded diagrams in an automated manner.

Appendix B - Pathfind R Results for infected Cells compared to mock infection and UV-treatment

The embedded object below can be opened as a spreadsheet in Microsoft Excel by right-clicking and selecting "open."

Pacific Northwest
National Laboratory

902 Battelle Boulevard
P.O. Box 999
Richland, WA 99354

1-888-375-PNNL (7665)

www.pnnl.gov