RESEARCH ARTICLE

PEPATAC: An optimized ATAC-seq pipeline with serial alignments

Jason P. Smith[1,4], M. Ryan Corces[5], Jin Xu[5], Vincent P. Reuter[6], Howard Y. Chang[5], and Nathan C. Sheffield[1,2,3,4,✉]

[1]Center for Public Health Genomics, University of Virginia
[2]Department of Public Health Sciences, University of Virginia
[3]Department of Biomedical Engineering, University of Virginia
[4]Department of Biochemistry and Molecular Genetics, University of Virginia
[5]Center for Personal Dynamic Regulomes, Stanford University, Stanford, CA
[6]Genomics and Computational Biology Graduate Group, University of Pennsylvania
✉ Correspondence: nsheffield@virginia.edu

Motivation: As chromatin accessibility data from ATAC-seq experiments continues to expand, there is continuing need for standardized analysis pipelines. Here, we present PEPATAC, an ATAC-seq pipeline that is easily applied to ATAC-seq projects of any size, from one-off experiments to large-scale sequencing projects.
Results: PEPATAC leverages unique features of ATAC-seq data to optimize for speed and accuracy, and it provides several unique analytical approaches. Output includes convenient quality control plots, summary statistics, and a variety of generally useful data formats to set the groundwork for subsequent project-specific data analysis. Downstream analysis is simplified by a standard definition format, modularity of components, and metadata APIs in R and Python. It is restartable, fault-tolerant, and can be run on local hardware, using any cluster resource manager, or in provided Linux containers. We also demonstrate the advantage of aligning to the mitochondrial genome serially, which improves the accuracy of alignment statistics and quality control metrics. PEPATAC is a robust and portable first step for any ATAC-seq project.
Availability: BSD2-licensed code and documentation at `https://pepatac.databio.org`.

Introduction

Because cells package chromatin differently depending on their function and phenotype, profiling chromatin accessibility is a primary experimental approach for understanding cell state[1–3]. The number of chromatin accessibility experiments has grown dramatically in recent years with the introduction of the assay for transposase-accessible chromatin (ATAC-seq)[4]. With the ATAC-seq method now widespread, there is demand for analytical approaches[5,6], including systematic processing pipelines to facilitate the goal of reproducible research and ease cross-study comparisons[7,8].

To address this need we developed PEPATAC, a fast and effective ATAC-seq pipeline that easily generalizes across compute contexts and research environments. This pipeline has been built over years of experience analyzing chromatin accessibility experiments and implements several concepts that make it effective. These include ATAC-specific quality control outputs, both nucleotide-resolution and smoothed signal tracks, and a serial alignment strategy to deal with high mitochondrial contamination. Our serial alignment strategy, or 'prealignments', allows the user to easily configure a series of genomes to align to before the primary genome. PEPATAC provides a framework that allows a user to align serially in customized order to as many genomes as desired, which will be useful for many situations, including species contamination, dual-species experiments, repeat model alignments, decoy contamination, or spike-in controls.

In PEPATAC, modularity and flexibility are paramount design considerations. PEPATAC is compatible with the Portable Encapsulated Projects (PEP) format[9], which defines a common project metadata description, allowing projects that use PEPATAC to be easily analyzed using any PEP-compatible tool. It also provides the possibility for a single project description to be shared across pipelines, computing environments, and analytical teams. PEPATAC is also easily customizable, including changing individual command settings or even swapping specific software components by modifying a few lines of human readable configuration files.

PEPATAC does not rely on any specific local or cloud computing infrastructure, and it has already been deployed successfully in various compute environments at multiple research institutes to yield several peer-reviewed studies[10–14]. To simplify installation, we also enable a computing environment with the command-line tools required to run PEPATAC using either docker or singularity with the bulker multi-container environment manager[15].

PEPATAC includes a well-documented code base along with detailed installation instructions, tutorials, and example projects, so it is useful for both the bench biolo-

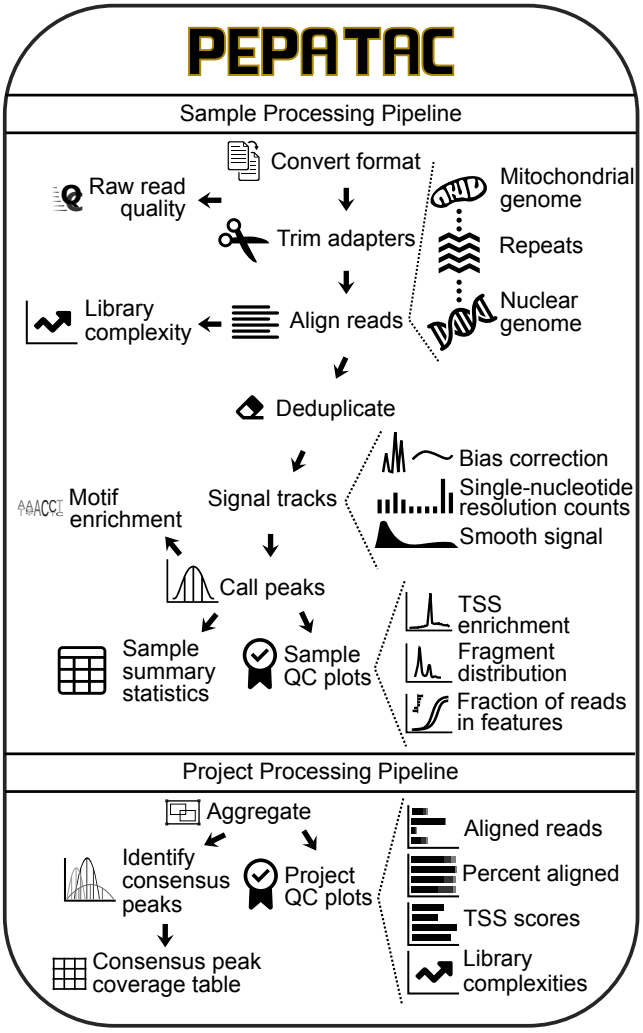

Fig. 1: PEPATAC pipeline steps. *Reads are first preprocessed, then serially aligned to the mitochondrial genome, curated repeats, and then the nuclear genome. PEPATAC generates both smooth and exact signal plots, called peaks, and QC output plots and tables.*

gist and bioinformatician alike. We anticipate that this pipeline will provide a useful complete analysis for basic ATAC-seq projects and serve as a unified starting point for more advanced ATAC-seq projects.

Design and Implementation

PEPATAC configuration

The PEPATAC pipeline is divided into two major parts (Fig. 1): First, it processes each sample individually in the *sample-level* part. Once sample processing is complete, the *project-level* part aggregates, analyzes, and summarizes the results across samples. PEPATAC is composed of two primary python scripts that may be run from the command-line. Sample information and parameters are passed to the pipeline as command-line arguments (see `pepatac.py --help`), making it simple to use as a standalone pipeline for individual samples without requiring a complete project configuration. Project level output is produced using the project level pipeline (see `pepatac_collator.py --help`). PEPATAC is built using the python module `pypiper`[16], which provides restartability, file integrity protection, copious logging, resource monitoring, and other features. Individual pipeline settings can also be configured using a pipeline configuration file (`pepatac.yaml`), which enables a user to specify absolute or relative paths to installed software, change adapter input files for trimming, and parameterize alignment and peak calling software tools. This configuration file comes with sensible defaults and will work out-of-the-box for research environments that include required software in the shell PATH, but it also may be configured to fit any computing environment and adapt to project-specific parameterization needs.

Refgenie reference assembly resources

Like any genome analysis, PEPATAC relies on reference genome annotations. To ensure that results are comparable across runs, it's important to use the same reference assembly. To manage these assets in a reproducible and robust manner, PEPATAC uses *refgenie*. `Refgenie` is a reference genome assembly asset manager that simplifies access to pre-indexed genomes and annotations for common assemblies, and also allows generating new standard reference genomes or annotations as needed while maintaining asset provenance[17]. For a complete analysis, PEPATAC requires several refgenie-managed assets: fasta, chrom_sizes, bowtie2_index, blacklist, refgene_tss, and feat_annotation. These can be either downloaded automatically or built manually, which would require a genome fasta file, a gene set annotation file from RefGene, and an Ensembl regulatory build annotation file. Using PEPATAC with `seqOutBias` requires the additional refgenie tallymer_index asset built for the same read length as the data. Many of these assets may also be directly specified at the command line should a user not have refgenie managed versions available. The TSS annotation file, region blacklist, and feature annotation file may all be specified to use a local, user-specified file. For example, while ENCODE provides a common set of regions that are aberrantly overrepresented in sequencing experiments (e.g. a blacklisted set of regions)[18], a user may create their own version of regions that should be excluded from consideration and point to this file manually.

File inputs and adapter trimming

PEPATAC sequentially trims, aligns, and analyzes sequences (Fig. 1). PEPATAC accepts sequence data input in 3 formats: unaligned BAM, separated FASTQ, or interleaved FASTQ format. The pipeline first converts the input format into FASTQ (if necessary) for adapter trimming. For adapter trimming, users may select between trimmomatic[19] and skewer[20] using command-line arguments or the PEP configuration file. The pipeline

stores quality control results including the number of raw, trimmed, or duplicated reads, and runs `FastQC`[21] if installed.

Prealignments and mitochondrial DNA

Because ATAC-seq data can have a high proportion of reads mapping to the mitochondrial genome (from 15%-50% in a typical experiment up to 95% in some experiments[22]), we considered how to optimize the pipeline to deal with abundant mitochondrial DNA (mtDNA). The typical strategy is to align to the mitochondrial and nuclear genomes simultaneously, and then remove nuclear-mitochondrial DNA (NuMts) post-hoc using a blacklist, but this suffers from three disadvantages: First, it is inefficient to align lots of mtDNA to the larger nuclear genome; second, reads that match both nuclear and mitochondrial DNA will be (incorrectly) split between the two, and third, this approach relies on an accurate pre-constructed annotation of NuMt locations, which may not be available for every reference genome. We found that by separately aligning first to the mitochondrial genome, we improved both the efficiency and the accuracy of the pipeline while alleviating all three of these challenges with simultaneously alignments. Moreover, to capture NuMts that span the artificial breakpoint induced by converting the circular mitochondrial DNA into a linear representation for alignment, we use a doubled mitochondrial reference sequence, which enables non-circular aligners to align reads that span the breakpoint. By default, the pipeline is configured to align reads first to the doubled mitochondrial reference genome, but may be easily configured to perform any number of additional serial alignments.

Alignments, deduplication, and library complexity

For prealignments and primary alignment, PEPATAC employs bowtie2 by default[23]. Bowtie2 settings are configurable in the pipeline configuration file but come with sensible defaults of `-k 1 -D 20 -R 3 -N 1 -L 20 -i S,1,0.50` for prealignments and `--very-sensitive -X 2000` for nuclear genome alignment. Users may optionally use bwa[24] with settings similarly configurable in the pipeline configuration file (default: `-M`). Following alignment, residual mitochondrial reads are removed and read deduplication is carried out using samblaster[25], but picard's MarkDuplicates[26] may also be utilized based on user preference. PEPATAC utilizes *preseq*[27] to calculate and plot sample library complexity at the current depth, reporting the number of duplicates (Fig 2a). The pipeline also projects the unique fraction of the library at 10M total reads. These metrics provide an estimate of library complexity and allow the user to determine the value of subsequent sequencing.

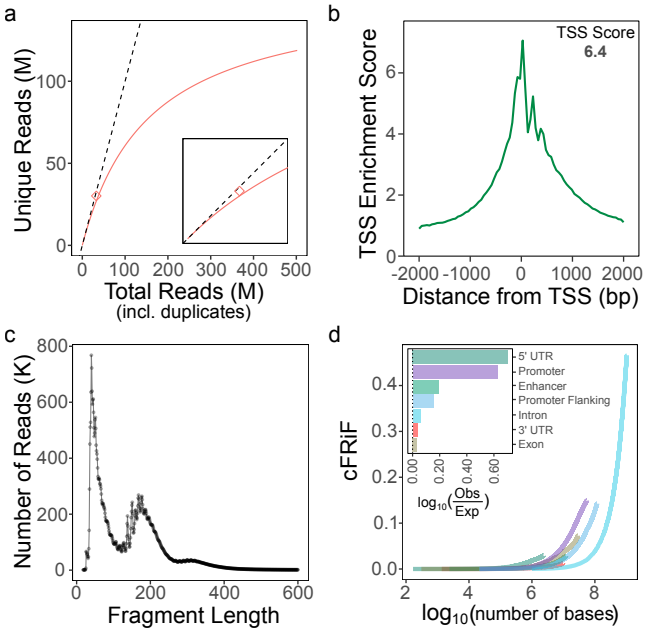

Fig. 2: Example PEPATAC QC plots for all reads. *(a) Library complexity plots the read count versus externally calculated deduplicated read counts (b) TSS enrichment quality control plot. (c) Fragment length distribution showing characteristic peaks at mono-, di-, and tri-nucleosomes. (d) Cumulative fraction of reads in annotated genomic features (cFRiF). Inset: Fraction of reads in those features (FRiF). Data from SRR5427743.*

Read QC metrics

For quality control, PEPATAC provides a TSS enrichment plot, produced by aggregating reads present in regions 2000 bases upstream and downstream of a reference set of TSSs (Fig 2b). Enrichment is calculated as the average number of reads in a 100 bp window around the TSS divided by the average number of reads in the first 200 bases of the entire region. This yields low signals in the tails with a peak in the center, which we take to be the TSS enrichment score. PEPATAC also produces a fragment length distribution plot (Figure 2c). A standard quality ATAC-seq library is expected to yield clearly defined peaks at open chromatin (<100bp), mononucleosomes (200 bp), and sequentially smaller peaks representing multi-nucleosomes at regular intervals. To evaluate the enrichment of all reads across genomic partitions, PEPATAC plots both the fraction and cumulative fraction of reads (FRiF, cFRiF respectively) in genomic features (Fig 2d). A novel feature of PEPATAC includes the plotting of the fraction of reads in any feature type, not solely in peaks. This is plotted as the cumulative sum of reads in each feature divided by the total number of aligned reads against the cumulative sum of bases in each feature. The relative proportion of each feature can be then be directly compared. For a quality sample, the proportion of reads in peaks should be the most enriched, reflecting the specificity of the peak calls for that sample.

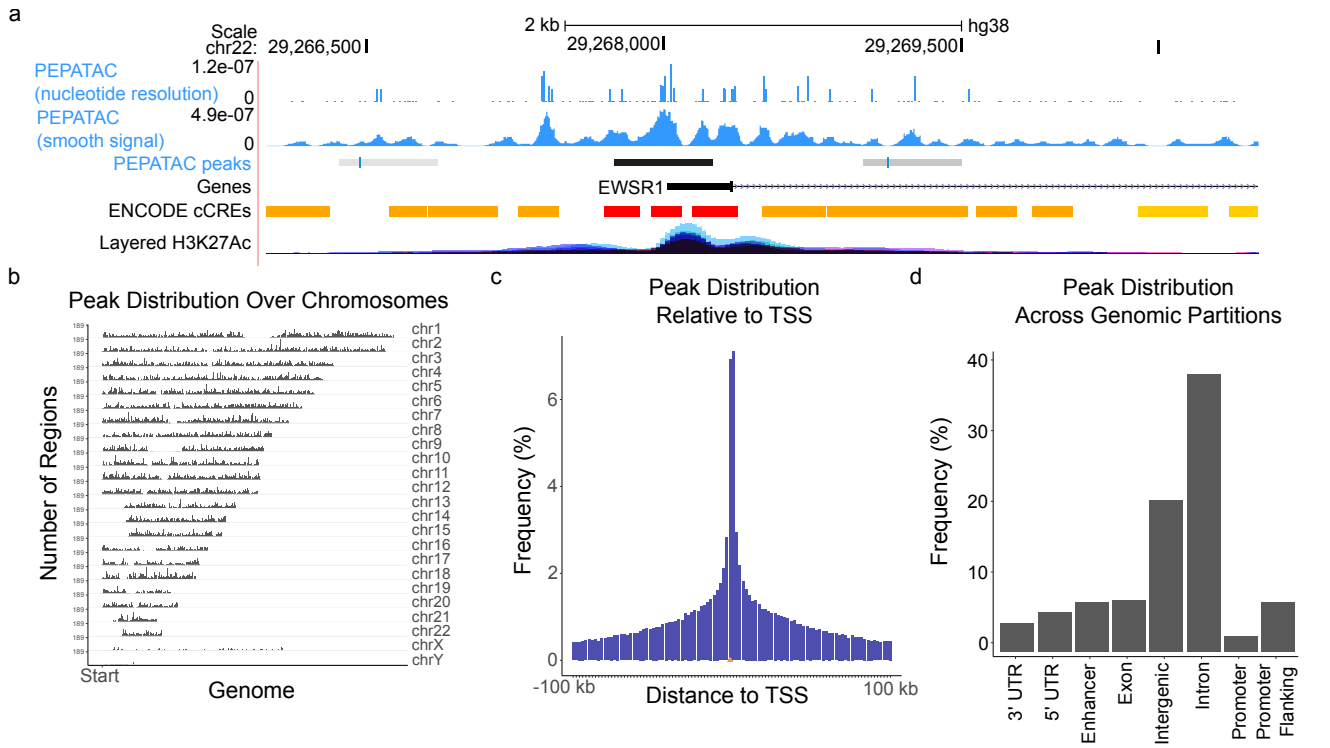

Fig. 3: Example PEPATAC QC plots for reads in peaks. *(a) Signal tracks including: nucleotide-resolution and smoothed signal tracks. PEPATAC default peaks are called using the default pipeline settings for MACS2[28]. (b) Distribution of peaks over the genome. (c) Distribution of peaks relative to TSS. (d) Distribution of peaks in annotated genomic partitions. Data from SRR5427743.*

Signal tracks and peak calling

Alignments are used to generate two signal tracks: one that records the exact location of transposition events, and one that is smoothed (Fig 3a). These two signal tracks may be used for different downstream analyses; the exact track is useful for analysis that requires nucleotide-resolution, while the smoothed version is often preferred for visualization and peak analysis. `seqOutBias` is an optional tool that can be used to correct for enzymatic (e.g. Tn5 transposase) bias and generate tracks for visualization[29]. The bias itself is corrected using a k-mer mask for the plus and minus strand Tn5 recognition sites and by taking the ratio of genome-wide observed read counts to the expected sequence based counts for each k-mer[29]. The k-mer counts take into account mappability at a given read length using GenomeTools' Tallymer program[30].

An earlier study found multiple peak callers worked well with chromatin accessibility data[31], and PEPATAC provides the option to use either F-Seq[32] or MACS2[28] for peak calling, with parameters customizable in the pipeline configuration file. MACS2 is used by default (`--shift -75 --extsize 150 --nomodel --call-summits --nolambda --keep-dup all -p 0.01`). Called peaks are standardized by extending up and down 250 bases (a tunable parameter, `--extend`) from the summit of each peak to establish peaks 500 bases in width. Any peaks which then extend beyond chromosome boundaries are trimmed. Peak scores are normalized to score per million by dividing by the sum of scores over 1M.

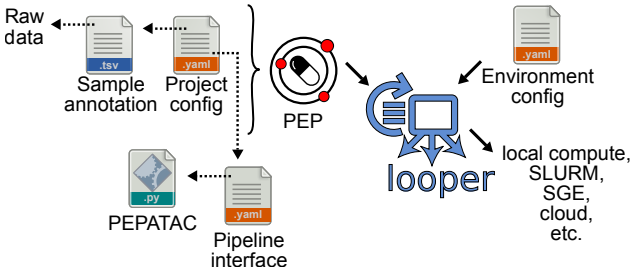

Fig. 4: Deploying PEPATAC across multiple samples using looper. *The PEPATAC pipeline can be easily run across multiple samples in any computing environment using looper.*

PEPATAC also produces several plots detailing enrichment of reads in peaks including: the distribution of peaks across the genome by chromosomal location (Fig 3b), the distribution of reads in peaks relative to TSSs (Fig 3c), and the distribution of peaks within genomic partitions (Fig 3d). The TSS distance distribution shows the distance of called peaks with respect to TSSs grouped in log-scale bins. Finally, users may optionally employ `Homer` to calculate motif enrichments in called peaks[33].

Deploying PEPATAC across multiple samples

To deploy the pipeline across multiple samples in a larger project, the pipeline uses the job submission engine `looper`[34], which employs the Portable Encapsulated Project standardized definition of project

metadata[9](Fig. 4). This standard project format enables a pipeline to be run on any project that follows the format, which is simple, standardized, and well-documented. `Looper` enables the PEPATAC pipeline to be run in any compute environment, including locally (the default) on a single laptop or desktop, or with any cluster resource manager. It also can be used with linux containers. Additionally, use of `looper's` project format gives pipeline users access to APIs written in Python and R for downstream analysis of pipeline results.

For the user whose environment is set up to run containers, we enable container use with either Docker or Singularity through the multi-container environment manager, `bulker`[15]. Using `bulker`, PEPATAC may be run in containers across samples and compute environments, simplifying deployment by requiring only `bulker` and the PEPATAC pipeline itself, eliminating the need to install each required package independently.

Aggregating results from multiple samples

To summarize and incorporate data across samples, the second step in a PEPATAC analysis is to run a project-level pipeline (`pepatac_collator.py`) that identifies consensus peaks across a project and calculates sample coverage of those consensus peaks in a convenient table for easy downstream analysis. To establish consensus peaks, PEPATAC identifies overlapping peaks between every sample in a project and defines the consensus peak's coordinates based on the overlapping peak with the highest score. Finally, only peaks present in at least 2 samples with a minimum score per million greater than or equal to 5 are retained. A peak count table is then provided where every sample peak set is overlapped against the consensus peak set. Individual peak scores for a hit (e.g. an overlapping peak) are weighted by dividing by the percent overlap of the sample peak with the consensus peak.

For navigating results, PEPATAC provides both sample and project level reports in a convenient, easy-to-navigate HTML report with project-level summary table and plots, job status page, and individual sample pages with sample statistics and QC plots all at your fingertips. In addition, `looper` will produce summary plots from individual sample statistics including the number of aligned reads, percent aligned reads, TSS scores, and library complexities. A user can produce the HTML report at any point during or after pipeline completion, with the job status page providing information on whether a sample has failed, is still running, or has already completed.

Results

To demonstrate PEPATAC's default workflow and output, we analyzed samples from the original standard ATAC[4], fast ATAC[36], and omni ATAC[37] protocol papers. This dataset includes human ATAC-seq reads from 33 standard ATAC, 152 fast ATAC, and 139 omni ATAC samples (Supplemental file 1). PEPATAC provides output and quality control results both for individual samples and for the project as a whole. For each sample, PEPATAC produces narrowPeak and bigWig files to visualize nucleotide-resolution alignments, smoothed alignments, and peak calls. PEPATAC also produces summary statistics files that report the number of reads, duplicates, genome alignment rates, transcription start site (TSS) enrichment score, number of called peaks, fraction of reads in peaks (FRiP), and job runtime among others for every sample in a project.

Prealignments

To evaluate the advantage of serially aligning to the mitochondrial genome separately, we measured the total alignment runtime of synthetic mixtures of mitochondrial-aligning (mtDNA) and whole human-aligning (hg38) sequences with and without prealignments. We constructed libraries of mixed mtDNA:hg38 mapping ATAC-seq reads from 0% to 100% mtDNA in increments of 10%, at 10 million, 20 million, and up to 200 million total reads in increments of 20 million reads, resulting in 121 different library combinations. We recorded the alignment time for each input file with and without prealignments (Fig. 5b). To determine for which scenarios using prealignments is beneficial, we calculated the log ratio of run times with prealignments versus without prealignments and found that using prealignments reduces the total time of alignment even when mtDNA alignment rates are under 10% (Fig. 5c). In addition to speed and efficiency gains, PEPATAC with prealignment to mtDNA yields higher alignment rates to mitochondrial sequence than aligning to a combined human and mitochondrial genome as is commonly performed (Fig. 5d). This is true for every sample tested no matter the library preparation protocol nor percent mitochondrial contamination (Fig. 5e). This result indicates that the common approach of simultaneously aligning to the nuclear and mitochondrial genomes systematically underestimates the fraction of mitochondrial reads in an experiment. We therefore propose that mitochondrial alignment rates are generally underestimated by about 1-5% in published reports. In conclusion, the serial alignment approach not only improves the efficiency of alignment in most cases, but it also improves the accuracy.

Fraction of reads in peaks

We next sought to understand how the serial alignment strategy affects calculation of Fraction of Reads in Peaks (FRiP). FRiP is a common qualitative measure of enrichment and sample quality. However, FRiP calculations are poorly defined, making it dangerous to compare FRiP scores among different protocols and approaches. ENCODE defines the denominator of the FRiP score

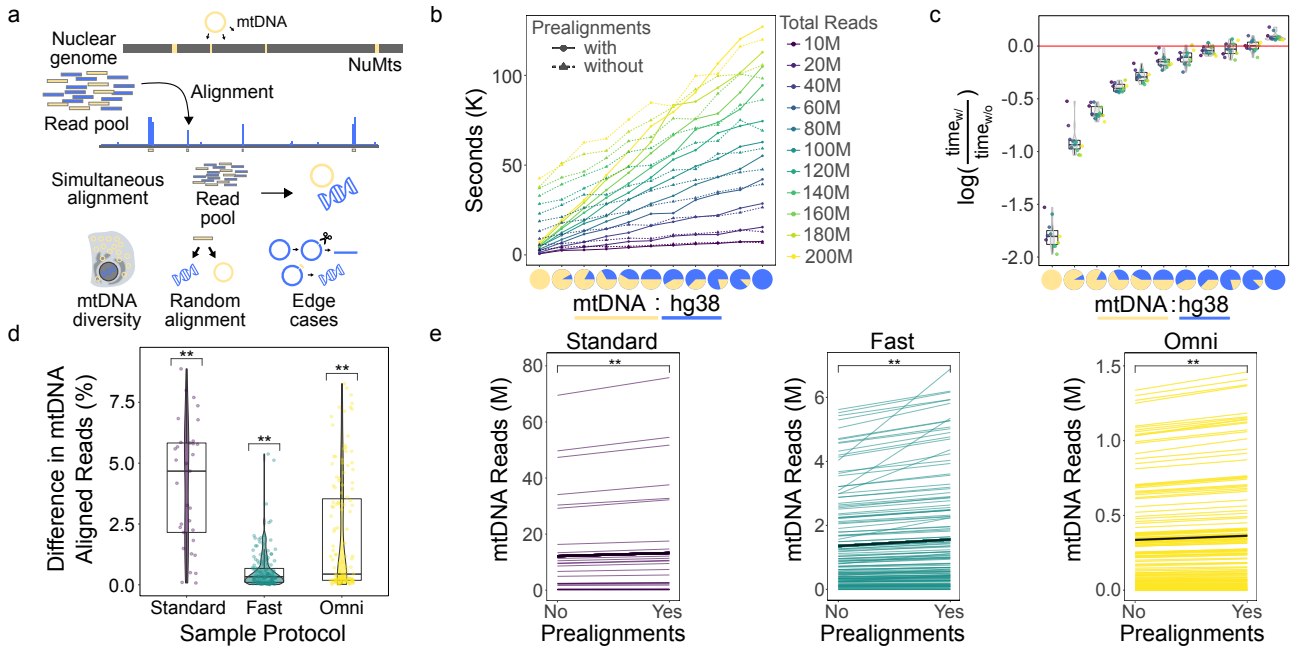

Fig. 5: PEPATAC prealignments increase mapped mtDNA reads and improve computational efficiency. *(a) NuMTs represent a significant complication of simultaneous alignment. (b) At mtDNA percentages from 10-100% at total read numbers ranging from 10-200M, using prealignments dramatically reduces run time. (c) Log ratio of prealignments runtimes versus no prealignment runtimes indicates significant savings, particularly when mtDNA contamination is >10%, no matter the total read count. (d) There is a significant increase in the percent of reads mapped to mitochondrial sequence when using prealignments versus not across standard, fast, and omni-ATAC protocols. (e) Within each ATAC-seq library preparation protocol, every sample shows increased mtDNA alignment when utilizing prealignments (The gray lines represent the mean increase within each protocol type. (d-e) ** = $p < 0.001$; t-test (mu = 0) with Benjamini-Hochberg correction.)*

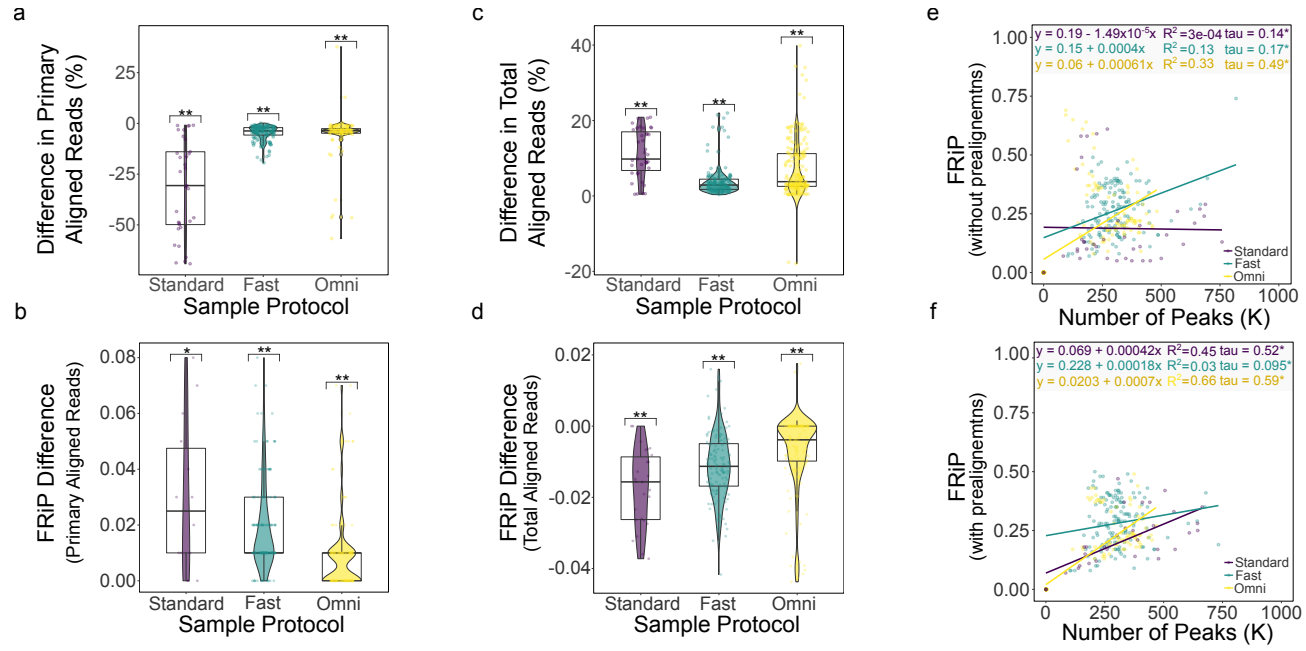

Fig. 6: The fraction of reads in peaks (FRiP) is dependent on mapping strategy. *(a) The number of primary, nuclear genome mapped reads is reduced when using prealignments. (b) However, the total number of mapped reads is increased with prealignments due to more specific read mapping. (c) The FRiP is increased with prealignments when using primary, nuclear genome mapped reads as the denominator. (d) In contrast, when using the total mapped reads the FRiP is reduced when using prealignments due to a larger mapped read pool in the denominator ((a-d): * = $p < 0.01$; ** = $p < 0.001$; t-test (mu = 0) with Benjamini-Hochberg correction). (e) As described in ChIP-seq[35], FRiP is positively correlated with the number of called peaks. (f) With prealignments, the positive correlation between FRiP and the number of called peaks tends to increase ((e-f):* = $p < 0.0001$; Kendall rank correlation coefficient).*

to be total mapped reads ENCODE Terms. If only one genome is used for alignment, then the calculation is clear, but for a serial alignment pipeline, the FRiP score depends on whether the denominator includes reads mapped to the nuclear genome only, or to all genomes (Fig. 6a,b). By default, PEPATAC uses the number of mapped reads (i.e. PEPATAC's "Aligned_reads" stat) which is relative to the primary alignment and excludes reads mapped during prealignments. This has the consequence of changing the FRiP calculation based on whether prealignments were used (Fig. 6c,d). When using prealignments, the default FRiP calculation will significantly increase, because the number of reads mapped to the primary genome is reduced due to reads mapping more accurately to the mitochondrial genome (Fig. 6c,d). When FRiP is calculated using the total mapped reads (prealignments **and** primary alignment), these relationships are inversed (Fig. 6b,d). In any scenario, prealignments lead to more total mapped reads, due to more efficient mitochondrial alignment. As more recent ATAC-seq sample preparation protocols intentionally reduce mitochondrial contamination, these differences are most pronounced when using the original, standard ATAC-seq protocol. It has also been reported that, at least in ChIP-seq experiments, FRiP correlates positively with the number of identified peaks[35]. Interestingly, this correlation is less obvious without prealignments, likely due to obfuscation through high numbers of mitochondrial aligning reads included in the FRiP denominator (Fig. 6e,f). Therefore, reliance on a specific cutoff (e.g. 0.3 or greater) as indicative of a quality sample must be relative to protocol and method.

Availability and Future Directions

PEPATAC is an efficient, user-friendly ATAC-seq pipeline that produces helpful quality control plots and signal tracks that provide a comprehensive starting point for further downstream analysis. Two key benefits of the PEPATAC pipeline over existing pipelines are its flexibility and modularity. PEPATAC is uniquely flexible, for example, by allowing pipeline users to serially align to multiple genomes, to select from multiple aligners, peak callers, and adapter trimmers, all while providing a convenient configurable interface so a user can adjust parameters for individual pipeline tasks. Furthermore, PEPATAC reads projects in PEP format, a standardized, well-described project definition format, providing a reproducible interface with Python and R APIs to simplify downstream analysis.

Because PEPATAC is built on `looper`, it is easily deployable on any compute infrastructure, including a laptop, a compute cluster, or the cloud. It is thereby inherently expandable from single to multi-sample analyses with both project level and individual sample level quality control reporting. This means that a user may submit any number of samples using *a single looper command and corresponding PEP metadata file.* Its design allows for simple restarts at any step in the process should the pipeline be interrupted. Due to its modular construction multiple software options for primary pipeline steps are available, creating a swappable pipeline flow path with individual steps adaptable to future changes in the field. PEPATAC is a rapid, flexible, and portable ATAC-seq project analysis pipeline providing a standardized foundation for more advanced inquiries.

Documentation and links

- PEPATAC: pepatac.databio.org.
- PEP metadata standards: pep.databio.org.
- `Looper` job submission engine: looper.databio.org.
- `Refgenie` reference genomes: refgenie.databio.org.
- Source code to reproduce output for this paper: github.com/databio/pepatac_paper_data.

Acknowledgments

We would like to acknowledge helpful discussions and contributions from Yuning Wei, Ying Shen, and Jason Buenrostro on early versions of the pipeline. This work was supported by NIH grants RM1-HG007735 (to H.Y.C.) and 1R35GM128636-01 (to N.C.S.). H.Y.C. is an Investigator of the Howard Hughes Medical Institute. M.R.C. is supported by a K99 (K99-AG059918) award and is an American Society of Hematology Scholar.

Declaration

H.Y.C. is a co-founder of Accent Therapeutics, Boundless Bio, and a consultant for Arsenal Biosciences and Spring Discovery. Stanford University holds a patent on ATAC-seq on which H.Y.C. is a named inventor.

Works Cited

1. Thurman, R. E. *et al.* The accessible chromatin landscape of the human genome. *Nature* **489,** 75–82 (2012).

2. Sheffield, N. C. *et al.* Patterns of regulatory activity across diverse human cell types predict tissue identity, transcription factor binding, and long-range interactions. *Genome research* **23,** 777–788 (2013).

3. Sheffield, N. & Furey, T. Identifying and characterizing regulatory sequences in the human genome with chromatin accessibility assays. *Genes* **3,** 651–670 (2012).

4. Buenrostro, J. D., Giresi, P. G., Zaba, L. C., Chang, H. Y. & Greenleaf, W. J. Transposition of native chromatin for multimodal regulatory analysis and personal epigenomics. *Nature methods* **10,** 1213 (2013).

5. Yan, F., Powell, D. R., Curtis, D. J. & Wong, N. C. From reads to insight: A hitchhiker's guide to ATAC-seq data analysis. *Genome Biology* **21,** 22 (2020).

6. Smith, J. P. & Sheffield, N. C. Analytical approaches for atac-seq data analysis. *Current Protocols in Human Genetics* **106,** e101 (2020).

7. Collins, F. S. & Tabak, L. A. Policy: NIH plans to enhance reproducibility. *Nature* **505,** 612–613 (2014).

8. Lauer, M., Tabak, L. & Collins, F. Opinion: The next generation researchers initiative at NIH. *Proceedings of the National Academy of Sciences of the United States of America* **114,** 11801–11803 (2017).

9. Sheffield, N. C., Stolarczyk, M., Reuter, V. P. & Rendeiro, A. Linking big biomedical datasets to modular analysis with portable encapsulated projects. (2020). doi:10.1101/2020.10.08.331322

10. Corces, M. R. *et al.* The chromatin accessibility landscape of primary human cancers. *Science (New York, N.Y.)* **362,** (2018).

11. Ram-Mohan, N. *et al.* Integrative profiling of early host chromatin accessibility responses in human neutrophils with sensitive pathogen detection. *bioRxiv* (2020). doi:10.1101/2020.04.28.066829

12. Granja, J. M. *et al.* ArchR: An integrative and scalable software package for single-cell chromatin accessibility analysis. *bioRxiv* (2020). doi:10.1101/2020.04.28.066498

13. Zhou, J. *et al.* CATA: A comprehensive chromatin accessibility database for cancer. *bioRxiv* (2020). doi:10.1101/2020.05.16.099325

14. Fan, H. *et al.* Epigenetic reprogramming towards mesenchymal-epithelial transition in ovarian cancer-associated mesenchymal stem cells drives metastasis. *bioRxiv* (2020). doi:10.1101/2020.02.25.964197

15. Sheffield, N. C. Bulker: A multi-container environment manager. *OSF Preprints* (2019). doi:10.31219/osf.io/natsj

16. Rendeiro, A. F. *et al.* Pypiper: A python toolkit for building restartable pipelines. (2020).

17. Stolarczyk, M., Reuter, V. P., Magee, N. E. & Sheffield, N. C. Refgenie: A reference genome resource manager. *Gigascience* (2020). doi:10.1101/698704

18. Amemiya, H. M., Kundaje, A. & Boyle, A. P. The encode blacklist: Identification of problematic regions of the genome. *Scientific reports* **9,** 9354 (2019).

19. Bolger, A. M., Lohse, M. & Usadel, B. Trimmomatic: A flexible trimmer for illumina sequence data. *Bioinformatics (Oxford, England)* **30,** 2114–2120 (2014).

20. Jiang, H., Lei, R., Ding, S.-W. & Zhu, S. Skewer: A fast and accurate adapter trimmer for next-generation sequencing paired-end reads. *BMC bioinformatics* **15,** 182 (2014).

21. Andrews, S. FastQC: A quality control tool for high throughput sequence data. (2010).

22. Wu, J. *et al.* The landscape of accessible chromatin in mammalian preimplantation embryos. *Nature* **534,** 652–657 (2016).

23. Langmead, B. & Salzberg, S. L. Fast gapped-read alignment with bowtie 2. **9,** 357–359 (2012).

24. Li, H. & Durbin, R. Fast and accurate short read alignment with burrows-wheeler transform. *Bioinformatics (Oxford, England)* **25,** 1754–1760 (2009).

25. Faust, G. G. & Hall, I. M. SAMBLASTER: Fast duplicate marking and structural variant read extraction. *Bioinformatics (Oxford, England)* **30,** 2503–2505 (2014).

26. Institute, B. Picard toolkit. *Broad Institute, GitHub repository* (2019).

27. Daley, T. & Smith, A. D. Modeling genome coverage in single-cell sequencing. *Bioinformatics* **30,** 3159–3165 (2014).

28. Zhang, Y. *et al.* Model-based analysis of chip-seq (macs). *Genome biology* **9,** R137 (2008).

29. Martins, A. fqdedup: Remove PCR duplicates from FASTQ files. (2018).

30. Kurtz, S., Narechania, A., Stein, J. C. & Ware, D. A new method to compute k-mer frequencies and its application to annotate large repetitive plant genomes. *BMC genomics* **9,** 517 (2008).

31. Koohy, H., Down, T. A., Spivakov, M. & Hubbard, T. A comparison of peak callers used for dnase-seq data. *PloS one* **9,** e96303 (2014).

32. Boyle, A. P., Guinney, J., Crawford, G. E. & Furey, T. S. F-seq: A feature density estimator for high-throughput sequence tags. *Bioinformatics (Oxford, England)* **24,** 2537–2538 (2008).

33. Heinz, S. *et al.* Simple combinations of lineage-determining transcription factors prime cis-regulatory elements required for macrophage and b cell identities. *Molecular cell* **38,** 576–89 (2010).

34. Stolarczyk, M. *et al.* Looper: A python-based pipeline submission engine and project manager. (2020).

35. Landt, S. G. *et al.* ChIP-seq guidelines and practices of the encode and modENCODE consortia. *Genome research* **22,** 1813–1831 (2012).

36. Corces, M. R. *et al.* Lineage-specific and single-cell chromatin accessibility charts human hematopoiesis and leukemia evolution. *Nature genetics* **48,** 1193–1203 (2016).

37. Corces, M. R. *et al.* An improved ATAC-seq protocol reduces background and enables interrogation of frozen tissues. *Scientific reports* **14,** 959–962 (2017).