Chromatin Accessibility Beyond the Peaks

Morgan Hamm

A dissertation
submitted in partial fulfillment of the
requirements for the degree of

Doctor of Philosophy

University of Washington
2026

Reading Committee:
Christine Queitsch, Co-Chair
Cole Trapnell, Co-Chair
Josh Cuperus

Program Authorized to Offer Degree:
Genome Sciences

© Copyright 2026

Morgan Hamm

University of Washington

Abstract

Chromatin Accessibility Beyond the Peaks

Morgan Hamm

Co-chairs of the Supervisory Committee:

Professor Christine Queitsch
Genome Sciences

Associate Professor Cole Trapnell
Genome Sciences

Chromatin state resides at the intersection of *trans*-acting factors that operate globally over the genome but respond to changing conditions, and DNA sequence which is invariant across conditions but varies locally around genes. Assays that measure features of chromatin state can help us understand this interplay between sequence and *trans* factors that ultimately governs transcriptional regulation. My dissertation work has been focused on the analysis of a long read chromatin accessibility assay called Fiber-seq. In this body of work I show that this single assay that, on its face, measures chromatin accessibility, is incredibly information rich and may help decode the regulatory logic governing gene expression. I present fiber-views, a software package for analysis of Fiber-seq data at aligned genomic positions. In the application of Fiber-seq to *Zea mays* I show Fiber-seq detects twice as many ACRs as ATAC-seq in paired samples. Fiber-seq is particularly good at identifying ACRs with short accessible elements and ACRs in repetitive regions, including transposable elements. Finally I present a novel analysis approach converting the single molecule data from Fiber-seq to a set of feature tracks. I show that these feature tracks are able to recapitulate chromatin states typically derived from multiple ChIP-seq assays. Fiber-seq derived features can predict gene expression, capturing nearly 60% of expression variation in maize. Patterns of Fiber-seq features can also be used to categorize ACRs reflecting their function and underlying sequence.

DEDICATION

To my mom, Tylee, who kindled in me a love of science and life-long learning.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 The importance of chromatin

DNA encodes all the proteins a cell or organism needs to function and survive. To function efficiently, cells need to regulate the rate individual genes are transcribed and translated into proteins. One level this regulation can take place is at transcription; by regulating the rate that mRNAs are produced, mRNA abundance can be controlled, ultimately contributing to controlling translation. Cells need to adapt their transcriptional control to varying conditions and to embody distinct cell types and yet their DNA sequence is generally invariant. Varying the abundances of *trans*-acting factors in the nucleus is one way cells can respond to conditions. Cells also need to carefully regulate the abundance of individual proteins relative to each other (dosage). However, within a nucleus, the abundances of *trans* factors are the same for all genes. DNA sequence – particularly the regulatory DNA surrounding genes, or *cis*-acting factors – allows genes to be regulated independently from each other.

Gene regulation depends on the interplay between *trans*-acting factors and *cis*-acting DNA elements. *Trans* factors operate globally across the genome and respond dynamically to cellular conditions, while DNA sequence provides stable, gene-specific regulatory information. Chromatin occupies the interface between these regulatory inputs, translating global *trans*-factor abundance into local, sequence-dependent chromatin states that determine transcriptional activity. These chromatin states can also persist as regulatory memory,

with different chromatin features having different levels of persistence.

1.1.1 Key features of chromatin

Below I provide a brief overview of the major classes of chromatin features.

Chromatin accessibility

Chromatin accessibility refers to the ability of some molecule to access and modify DNA. Accessible regions typically mark sites where transcription factors bind and regulatory activity occurs, making accessibility a key indicator of functional regulatory elements. Various assays have been developed to measure accessibility, all of which involve treating intact nuclei with a molecule that will alter the DNA either by digestion or modification. Regions of DNA that are not tightly bound to protein complexes such as nucleosomes will be affected by the assay molecule while bound regions are not.

Chromatin accessibility has been of great interest in part due to its ability to highlight regions of regulatory DNA where variants contribute to phenotypic variation or disease. Rodgers-Melnick et al. (2016) identified that 38% of explainable heritable variation in maize NAM lines is attributable to SNPs in MNase accessible sites which make up 1% of the genome (an 18-fold enrichment) [1]. In humans, Maurano et al (2012) found that 76.6% of noncoding GWAS SNPs were either in **DNase I** accessible sites or in perfect linkage disequilibrium to SNPs in accessible sites [2].

Although there is a great deal of correlation between locations identified as accessible by different assays, some differences are apparent. For example, Zhao et al (2020) cataloged differences between accessible regions found using MNase-seq and ATAC-seq, hypothesizing that the large size of Tn5 dimer complex may limit its ability to access certain locations identified as MNase peaks [3]. In published work presented in chapter 3 [4], we show a similar phenomenon in which Fiber-seq identified short (size) accessible chromatin regions (ACRs) that were not detected by ATAC-seq.

DNA methylation

5-Methylcytosine (5mC) is the most common native DNA methylation observed in plants as well as vertebrates. In mammals, methylation occurs predominantly at CG dinucleotides (commonly known as CpGs)

and is generally associated with repression [5]. In plants, 5mC methylation occurs in three sequence contexts: CG, CHG, and CHH (where H = A, T, or C), and while also associated with repression, CpG methylation is often observed in transcribed gene bodies. Both plants and vertebrates/mammals have systems for maintaining CpG methylation that use the symmetrical nature of the methylated dinucleotide as a template (i.e., the reverse complement of 5mCpG is often also 5mCpG), resulting in some level of heritability through both mitosis and meiosis. However, CHG and CHH methylation lack this symmetrical template aspect and therefore require specialized protein complexes to methylate *de novo* after each DNA replication.

Histone variants and histone marks

Nucleosomes consist of approximately 147 bp of DNA wrapped around an octamer of histone proteins (two copies each of H2A, H2B, H3, and H4). Eukaryotic genomes encode multiple variants of core histones (Table 1). Histone variants can alter nucleosome stability, dynamics, and interactions with regulatory factors [6]. For example, H2A.Z incorporation is associated with nucleosome instability and is enriched at promoters and regulatory regions [7].

Histone	Arabidopsis thaliana [8]	Homo sapiens [9]
H1	3	13
H2A	13	43
H2B	11	32
H3	13	26
H4	8	16

Table 1.1: Number of histone variant genes in the Arabidopsis and human genomes

Beyond variant composition, histones are subject to numerous post-translational modifications including methylation, acetylation, and phosphorylation. These modifications can directly affect chromatin compaction or serve as binding sites for effector proteins that recognize specific modifications [10], [11]. Histone acetylation generally correlates with open chromatin and active transcription, while histone methylation effects depend on the specific methylation. For example, H3K4me3 marks active promoters, H3K36me3 marks actively transcribed gene bodies, while H3K9me2/me3 and H3K27me3 are associated with heterochromatin and Polycomb-mediated repression, respectively [12].

The ENCODE project [13] lists 32 histone modifications for which data was collected via ChIP-seq or

similar experiments.

Inferred chromatin states

Cells use these chromatin features – signals associated with DNA – as part of their operating logic, deciding which genes to turn on and off based on developmental and environmental cues. It has been a research goal for decades to decode these signals. As many of the chromatin features described above were being collected during the ENCODE project [13] it became apparent that many of them are correlated (or anti-correlated), contributing redundant information. Some may be simply irrelevant to gene regulation. Researchers discovered that the information from these many chromatin *features* can be collapsed into a modest number (8-15) of discrete chromatin *states* [10]–[12]. Hidden Markov models and dynamic Bayesian networks like ChromHMM and Segway have been used to classify and annotate these chromatin states across the genome [14]–[16].

1.2 Fiber-seq: a rich source of chromatin features

In 2020, Stergachis et al introduced Fiber-seq, which captures accessibility on individual chromatin molecules using long-read sequencing. Fiber-seq is information-rich, revealing a number of chromatin features in addition to simple peaks of accessibility at genomic loci, such as the size of accessible regions and nucleosome-protected regions, as well as the phasing of accessible sites and nucleosome positions across molecules at a single locus. Collectively these features suggest that Fiber-seq may provide the most powerful set of features in a single experiment. In Chapter 4 I begin to explore the extent to which Fiber-seq can supersede other chromatin assays. Furthermore, as shown in my 2025 publication (Chapter 3), Fiber-seq reveals accessibility in regions of the genome missed by other assays, including within LTR retrotransposons, loci that become accessible in other tissues, and loci that are prone to transposon insertion.

1.3 Plants are not only useful, they have interesting genomes

Humans have been adapting plant traits for over 10,000 years and plants have been shown to be incredibly receptive to this manipulation. Part of the reason for this is the plasticity of their genome. While many of the

major genetic changes that occurred during domestication are Mendelian in nature, such as the Tb1 mutation that converts a bushy plant to one with a single primary stalk [17] or the Tga1 mutation that converts a plant with encased kernels to one with naked kernels [18], many others are the result of complicated reshufflings of the genome. Examples of this latter category include the diversification of Brassica species [19] and the massive growth in maize ear size over the last century [20]. Plants show remarkable adaptation to large scale genomic change.

More recently, scientists have developed ways to genetically engineer plants to confer herbicide or pest resistance (roundup ready, BT). Most current genetic engineering examples are addition or modification of a small number of genes to confer a specific trait. This is partially because such changes are easily patentable and threfore worth the regulatory effort. Plants are still relatively challenging to transform and genetic modification procedures often require developing species-specific transformation protocols, but technologies are rapidly improving [21], [22].

These more modern efforts continue to be worthwhile even for the more challenging genetics of complex traits. Plants are incredibly good at producing small molecules and have complex metabolic processes to do so. Cannabis is an example of a plant that has been bred to significantly increase the yield of a specific metabolite (THC) showing a rapid increase in potency in the last 50 years. Plants are superior to bacteria and yeast at producing many small molecules, especially those for which the pathway already exists in plants and is too complex to reconstitute in unicellular organisms, or when eukaryotic-style protein folding is required. Plants are inherently scalable; we have been growing large amounts of plant biomass for millennia.

Fiber-seq can provide a nuanced understanding of chromatin and gene regulation in plants that will help us create the next generation of modified plants for food, small molecule production, and more.

In my time as a graduate student I had the opportunity to work with Fiber-seq when it was a relatively nascent technology. In this dissertation I hope to demonstrate both the richness of Fiber-seq data, as well as show my contributions in the realm of data analysis of this assay. In Chapter 2, I present a software package for analysis and visualization called Fiber-views. In Chapter 3, I shows the application of Fiber-seq to *Zea mays*(maize or corn), a major global crop. In Chapter 4, I present a novel approach to the analysis of Fiber-seq data that attempts to exploit the richness of chromatin features captured by the assay. Not included

in this dissertation are other publications to which I contributed and am an author [23]–[26]

Chapter 2

Fiber-views

fiber-views puts Fiber-seq data into annotated data matrices (anndata) for easy manipulation, clustering, analysis, and visualization.

2.1 Introduction and Problem Statement

2.1.1 Fiber-seq Data and Storage Format

Fiber-seq captures a remarkably rich set of chromatin features from individual DNA molecules. A single Fiber-seq read reports on DNA sequence, m6A methylation indicating chromatin accessibility, and endogenous CpG methylation. After some processing additional information encoding the positions and sizes of nucleosomes and methylation-sensitive patches (MSPs) is added. Additionally, several quality metrics are associated with each read (read quality, number of CCS passes...). This multi-layered information makes Fiber-seq data well suited for understanding chromatin state, but also presents challenges for data organization and analysis.

The standard format for storing Fiber-seq data is the SAM file format, (typically compressed as BAM or CRAM files) which uses a nested data structure where each aligned read is represented as a list of fields, and many of those fields are themselves lists or arrays. Base modifications like m6A and CpG methylation are encoded using standardized MM and ML tags defined in the SAM format specification. Additional Fiber-

seq-specific features such as nucleosome and MSP boundaries are stored in custom tags, with conventions established by the Fibertools software package [24]. This nested structure is flexible and extensible via custom BAM tags, and thus meets the needs of storing Fiber-seq and other methylation based long read chromatin assays.

2.1.2 Data Structure requirements for matrix based analysis

Analysis of Fiber-seq data typically involves examining fibers that align to specific genomic positions of interest. Genome browsers provide one approach to this, presenting a visual alignment of reads to a reference genome where individual fibers are visually stacked vertically at a locus. This visualization of the data is useful for exploring, but doesn't provide easy access to manipulate or quantitatively analyze the underlying data.

There are several tools that do provide low level access to the underlying data. Pysam is a general Python package for working with SAM files, and supports some methods relating to the MM, ML base modification tags. Fibertools also provides several commands for accessing the low level methylation and region data, (ft extract, ft center, ft pileup). PacBio provides some tools like pb-CpG-tools, methBAT for looking at (CpG) methylation. Generally these tools that provide low level access to the data maintain a nested structure to the data similar to how it is stored. This is logical, but it leaves it to the user to develop their own methods for putting data of one type from different fibers into a single data structure.

Many analytical techniques that could be useful if applied to Fiber-seq data like clustering algorithms, dimensionality reduction methods, and machine learning approaches require data in matrix format. These matrix-based methods can reveal patterns in chromatin state, identify groups of fibers with similar accessibility profiles, or quantify aggregate chromatin features across many molecules.

A critical requirement throughout these analyses is preserving the metadata associated with each fiber. Information like read quality scores, genomic context (which gene or regulatory element the fiber spans), and experimental conditions need to follow the Fiber-seq read through clustering, filtering, and visualization steps. Losing track of metadata like sample identity or genomic location during these re-arrangements would undermine downstream analysis.

The gap between how Fiber-seq data is stored and how it needs to be analyzed creates practical challenges. BAM files organize data by read, with a nested structure that encodes all features for each molecule. This organization is ideal for storage and for tools that process reads sequentially. However, converting from BAM format to a matrix for analysis often means losing annotations. For example, extracting m6A methylation into a matrix indicating accessible positions works well for analyzing accessibility patterns. However, analyzing CpG methylation patterns or nucleosome positions requires returning to the BAM file and regenerating a new data structure. When clustering fibers to group similar chromatin states together for visualization, reordering operations must be applied across multiple independent data structures without introducing errors. The fundamental issue is the lack of a data format for analysis and visualization that preserves all Fiber-seq features and metadata through transformations like subsetting and reordering.

2.1.3 Similarity Between Fiber-seq and Single-Cell Genomics Data

This challenge of maintaining per-sample metadata through transformations is not unique to Fiber-seq. Single-cell genomics faces similar issues: each cell has associated metadata (cell type annotations, experimental batch, quality metrics...) that needs to follow that cell through clustering, filtering, dimensionality reduction, and visualization. Losing track of which cluster corresponds to which cell type, or which batch a cell came from, would make the analysis uninterpretable.

The single-cell genomics community has developed a number of mechanically similar data structures to address this problem including the cell_data_set, and SingleCellExperiment classes in R [27], and AnnData in Python commonly used with the scanpy package for single cell analysis. All of these data structures provide a standardized way to organize high-dimensional data where observations (cells) are annotated with metadata, features (genes) have their own annotations, and the data can be represented in multiple layers (raw counts, normalized values, RNA, ATAC...). These objects maintain associations between metadata annotations and data rows through subsetting, concatenation, and transformation operations. When clustering cells and reordering them, all the metadata follows automatically. In particular, AnnData has been co-developed with Scanpy, and intentionally designed to abstract away any single-cell specific language and methods from the core annotated data matrix format, so that it can be adapted for use in other fields.

Fiber-seq data has striking parallels to single-cell data: individual fibers are like individual cells, genomic

positions are like genes, and we have multiple data layers (sequence, modifications, regions) that all need to stay coordinated. Given the level of adoption and support anndata has as well as the abstraction away from single-cell specific features and its support for numpy compatible arrays I decided to adapt anndata's architecture for Fiber-seq data rather than building a new data structure from scratch.

2.2 fiber-views: Design and Rationale

2.2.1 Overview and Design Philosophy

I developed the Python package fiber-views to address the data structure challenges outlined above. I consider fiber-views to consist of 2 components. The first is the implementation of how Fiber-seq data is stored in AnnData objects. The second is the set of functions provided in the Python package for creating, manipulating, and visualizing the Fiber-seq data in that format. The core principle behind fiber-views is to provide access to all features of Fiber-seq data in matrix form while preserving all fiber metadata.

2.2.2 Alternative approaches

As mentioned above, AnnData was chosen as the basic data object for fiber-views for a number of reasons. AnnData is well documented and supported. It supports solving the exact challenges faced when working with Fiber-seq data in matrix format; maintaining annotations through complex analytical workflows. It supports storing data in sparse numpy-compatible matrices, this means data covering tens of thousands of bases, across thousands of fibers can efficiently be stored and worked with. It also means the data is fully compatible Python's ecosystem of numerical analysis and visualization tools like Numpy, Scipy, scikit-learn, pandas, and matplotlib.

However, there are many alternative approaches that can be taken. Below I list a few alternative approaches, and mention some of their pros and cons.

Store fiber-metadata in separate table linked by row-name The issue of metadata following data rows (fibers) could be solved simply by using a separate data frame indexed by read name or alternative identifier. However, this approach requires manual coordination when subsetting or reordering, making it error-prone

and requiring custom code for each transformation.

Apply row transformations to all sets each time This approach would maintain a list of matrices (sequence, modifications, regions) and metadata dataframes, then apply subsetting or reordering operations to each structure independently. AnnData effectively implements this internally, eliminating the need to recreate this coordination logic.

Re-create matrix data each time rows are re-organized Rather than maintaining matrices, this approach would store a collection of BAM entries indexed by read ID and regenerate matrices as needed after reordering. The computational cost of re-generating matrices multiple times would not be an issue when working with a single or small number of regions but would add up when performing systematic analyses.

xarrays Python package (dataset object) The xarray Dataset object provides aligned data arrays with labeled dimensions and supports subsetting opperations across the set of aligned arrays. However, xarray has limited support for mixed-type dataframes needed for fiber metadata (combining text fields like read names with numeric quality scores). Additionally, AnnData already provides the necessary structure without requiring custom implementation.

Sub-class the Anndata class to create a unique Fiber-view class This would allow Fiber-seq specific functions (tools.py) to be integrated into the object and callable. Note, Scanpy, the pre-eminent application of Anndata does not do this. I did not sub-class Anndata for ease of development.

2.2.3 Addressing Fiber-seq-Specific Challenges

While the AnnData class solves most problems associated with representing Fiber-seq data, there are some Fiber-seq specific challenges that required custom solutions.

Encoding Regions The most significant Fiber-seq specific challenge is efficiently storing region features like nucleosomes and MSPs. These features are fundamentally different from point modifications, they have variable lengths, and can cover from a small fraction to the majority of read bases. The most memory efficient way to store this information is in list(s) of the start positions, lengths, and optionally scores of these

regions. However, these lists would need to be updated when the AnnData object is subset or re-ordered, and no such functionality exists to do this automatically. The most obvious matrix based approach to handling this would be a matrix where every position covering a region is marked, however this is not a memory efficient solution; regions can be hundreds of bases long, and in a sparse matrix each of those bases would be occupied with redundant information. An ideal solution would be to use a row-wise run-length encoding sparse matrix. Unfortunately, there is no support for such a matrix type compatible with AnnData that could elegantly solve this problem.

The solution I developed is to represent each region type using three coordinated sparse matrices. The first matrix stores position information—specifically, the offset from the start of the region to the reporting position. The second stores the total length of each region. The third stores an associated score (like FIRE scores for MSPs or occupancy for nucleosomes). Regions are reported at their start and end positions and at regular intervals controlled by a parameter called region-report-interval. The regular intervals the region is reported ensure that fiber-view object can be subset column wise down to at minimum the region-report-interval length, while preserving data from any regions that span the subset columns. Each reporting position contains all information to reconstruct the reported region, and because the region start position is encoded relative to the current position, no external reference is needed to correctly position the start of the region. This approach efficiently stores region data, and supports subsetting, though some basic helper functions are needed to convert from this format to a dense matrix or list/dataframe representation of the regions for analysis and visualization.

supporting different modification and region types fiber-views needs to flexibly handle multiple types of base modifications and genomic regions. Different sequencing platforms (PacBio vs. Oxford Nanopore) and different base-calling parameters produce different modification calls. Similarly, different versions of analysis software encode regions using different SAM tag conventions. For example, earlier versions of Fibertools defined MSP regions using tags 'as' and 'al' for region start and length, while current versions include an additional 'aq' tag for FIRE scores. fiber-views includes a standardized way to specify modification definitions, including which MM tag codes correspond to which modification type, what probability thresholds to use for calling modifications, and how to handle strand-specific offsets (like the +1 offset

needed for CpG methylation on the reverse strand). Region definitions similarly specify which BAM tags encode region boundaries and scores. Preset definitions are provided for common experimental configurations (PacBio Fiber-seq, ONT Fiber-seq, nucleosome calls, FIRE calls), but the new or unanticipated region and modification types can be accommodated.

2.3 fiber-views Data Format

2.3.1 Overview of Data Organization

The fiber-views format organizes Fiber-seq data into several components within an AnnData object. The main data matrix (X) is technically required by AnnData but is kept as an empty sparse matrix in fiber-views. Actual data lives in named 'layers' where it can be organized more appropriately.

Layers

The layers component holds the core Fiber-seq data across multiple matrices, all with the same dimensions (number of fibers by number of base positions). The sequence layer 'seq' stores DNA sequences as a character byte array. Base modification layers such as m6A, CpG methylation, and optionally others are stored as sparse boolean matrices where true values indicate the presence of a modification at that position. As mentioned above, data for each region type is stored across 3 sparse matrices. These are associated by use of a common base name, for example, MSP regions use the base name 'msp' with matrices named 'msp_pos', 'msp_len', and 'msp_score'.

obs

Observation metadata (obs) captures per-fiber annotations in a pandas DataFrame. Fields may include read name, genomic context information like chromosome, position, and strand, read quality metrics from BAM tags, associated gene or other annotation feature, and experimental metadata (cell type, treatment condition, biological replicate).

var

The variable metadata (var) contains metadata about the columns of the layers. For fiber-views that can include genomic position or position relative to the window start and stop. these can be useful for labeling axes of plots and visualizations.

uns

The unstructured metadata (uns) stores metadata not associated with specific rows or columns of the data. There are several fields that certain functions in the fiber-views package expects to see to function properly. These fields are generated automatically when a new fiber-views AnnData object is created, but may need to be updated in some cases.

region_report_interval This specifies the spacing at which regions are reported in the sparse region representation. It determines the minimum window size for column-wise subsetting while preserving complete region information.

mods This field contains a list of layer names corresponding to base modifications. Aggregation functions use this list to ensure all base modification types are included in the aggregated result.

region_base_names This field contains a list of base names used for region types. This is also used by aggregation functions and should be updated when new region layers are created. For example, the `tools.filter_regions()` function creates a new region type and updates this field automatically.

2.4 fiber-views Package Functionality

This section outlines some of the useful functionality of the fiber-views package. Complete documentation on each function can be found in Appendix A.

Creating fiber-views objects

Currently fiber-views provides two main functions for building fiber-view objects from BAM files.

`build_single_fview()` extracts fibers aligned to a single genomic position

`build_multi_fview()` processes multiple sites.

These two functions share most of their arguments with the exception of site specification: `build_single_fview()` takes a single site as a dictionary or pandas Series `site_info`, while `build_multi_fview()` takes a DataFrame of sites `sites_df`. Other key parameters include `window`, which specifies the number of bases upstream and downstream around each site to extract, `mod_defs` and `region_defs`, which define which modifications and regions to extract (these can be preset strings for common assay types or custom definition lists). The `fully_span` parameter controls whether to require reads to span the entire window. Complete parameter descriptions are available in the function documentation.

Below is minimal example code showing how to create a fiber-view.

```
import fiber_views as fv

bam_path = fv.example_bam_path
site_info = {'seqid' : 'chr3', 'pos' : 10000, 'strand' : '-'}

fview = fv.build_single_fview(bam_file=bam_path, site_info=site_info,
                mod_defs=fv.PB_FS_mod_defs, region_defs=fv.NUC_region_defs,
                window=(-2000, 2000), fully_span=False)
```

Plotting

The plot submodule provides a number of plotting functions that can be used to can be used to build layered visualizations of Fiber-seq data.

`plot.annotate_boundaries()` Adds 's_pos' and 'e_pos' columns to the observation metadata, marking the start and end positions for each fiber based on first and last non '-' sequence characters.

`plot.make_plot_ax()` Creates or configures a matplotlib axis with appropriate x and y limits for plotting fibers.

`plot.draw_fiber_lines()` Draws thin horizontal lines representing the span of each fiber. This visualization works well for fewer than 150 fibers.

`plot.draw_fiber_bars()` Draws rectangular bars representing fibers. use this with 'width=1' for plotting hundreds of fibers.

`plot.draw_regions()` Draws regions (nucleosomes, MSPs, or others) as colored rectangles.

`plot.draw_mods()` Draws base modifications as narrow rectangles at single base positions.

`plot.draw_split_lines()` Draws horizontal dividing lines between groups of fibers based on a grouping variable in the 'obs' data frame. This is useful when looking at multiple sites, or after clustering fibers.

The plotting functions that begin with 'draw' can be called sequentially to build up layers of a visualization. For example, the following code creates a fiber-view centered on a gene transcription start site, and creates a visualization of MSPs and CpG methylation patterns surrounding the TSS.

```
import fiber_views as fv
import matplotlib.pyplot as plt

bam_path = fv.example_bam_path
example_genes_bed = fv.example_bed_path
bed_data = fv.read_bed(example_genes_bed)
anno_df = fv.bed_to_anno_df(bed_data)

fview = fv.build_single_fview(bam_file=bam_path, site_info=anno_df.iloc[16, :],
                mod_defs=fv.PB_FS_mod_defs, region_defs=fv.NUC_region_defs,
                window=(-2000, 2000), fully_span=False)

fv.tools.mark_cpg_sites(fview)

wd = 0.7
fig, ax = plt.subplots(figsize=(12, 6))
ax = fv.plot.make_plot_ax(fview, ax)
fv.plot.draw_fiber_bars(fview, ax, width=wd)
fv.plot.draw_regions(fview, ax, color="purple", width=wd)
fv.plot.draw_mods(fview, ax, mod='cpg_sites', width=wd, color='blue')
fv.plot.draw_mods(fview, ax, mod='cpg', width=wd, color='red')
```

```
ax.set_ylim(59.5, -1)

ax.set_title(f'MSPs around TSS of {fview.obs.gene_id.iloc[0]}, ' \
             'first 60 fibers, {fview.obs.site_name.iloc[0]}')
ax.set_xlabel('position from TSS')
ax.set_ylabel('fiber')
plt.tight_layout()
```

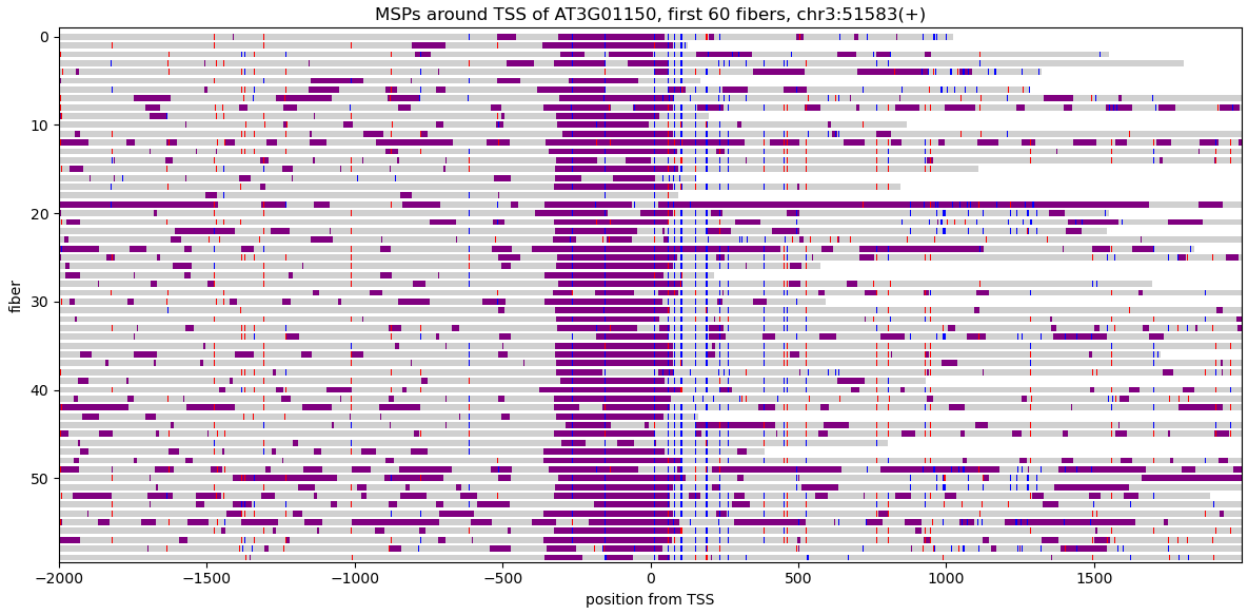

Fig. 2.1: Example of plotting using fiber-views.

Region helper functions

`tools.make_region_df()` constructs a pandas DataFrame where each row represents one region with columns indicating start position, length, and score.

`tools.make_dense_regions()` creates a dense matrix where every position covered by a region is marked. This is useful for calculating overlap statistics or creating heatmap visualizations.

`tools.filter_regions()` subsets regions based on length or score thresholds, either replacing the original regions or creating a new region type. This can be useful when distinguishing nucleosomal linkers from larger accessible patches for example.

`tools.split_fire()` splits the ‘msp‘ region type into ‘lnk‘ and ‘fire‘ region types based on the region scores (usualy derived from BAM tag ’as’). This is a convenience implementation of `tools.filter_regions()`.

Aggregation

`tools.agg_by_obs_and_bin()` aggregates a fiber-view by rows and columns. The ’obs_group_var’ parameter sets the column used for aggregating rows. Rows with the same value in this column will be aggregated together. The ’bin_width’ parameter sets the number of adjacent positions to be aggregated in each bin across the columns. Setting ’obs_group_var’ to ’None’ or ’bin_width’ to 1 will negate aggregation across rows or columns respectively.

Aggregation combines multiple fibers and bins adjacent genomic positions together, producing a more compact representation of the data. The function returns a new AnnData object with a transformed set of layers, stored as dense matrices.

The sequence layer from the original fiber-view is converted into five layers in the aggregate object. Four of these layers count occurrences of each base (labeled ’A_count’, ’C_count’, ’G_count’, and ’T_count’). The fifth layer, ’read_coverage’, sums these four base counts to indicate total read coverage at each bin.

For each basemod layer listed in ’uns[mods]’, a corresponding count layer is created that tallies modification occurrences within each bin. For example, the ’m6a’ layer becomes ’m6a_count’. It is useful to run `mark_cpg_sites()` prior to aggregation, which creates a ’cpg_sites’ layer marking all CpG dinucleotides. This allows aggregation to report both total CpG sites and methylated CpGs in each bin.

For each region type listed in ’uns[region_base_names]’, a coverage layer is created representing the total number of bases covered by that region type within each bin. For example, the ’msp’ region type produces an ’msp_coverage’ layer.

k-mer counting

`tools.count_kmers()` creates a k-mer count matrix labeled ’kmers’ in the ’obsm’ fields of the AnnData object. Rows of this matrix are the k-mer counts for individual fibers, the columns represent specific

k-mers. Two fields are also added to 'uns': 'kmer_len' is the value of k used for the matrix, and 'kmer_idx' is the column index for the created k-mer count matrix. This function only works on fiber-views where all fibers fully span the window (using `fully_span=True` when creating a fiber-view).

`tools.calc_kmer_dist()` calculates a pairwise distance matrix of k-mer count vectors between all fibers. The distance matrix is saved as 'kmer_dist' in 'obsp'. This function requires the 'kmers' matrix from `tools.count_kmers()`. Existing Python clustering tools like 'scipy.cluster.hierarchy' can be used to cluster fibers by their sequence. this can be useful for separating haplotypes, or identifying when a heterogeneous set of reads are mapping to a single genomic loci. The rDNA loci are good examples of this, which are often collapsed in reference genomes, despite distinct sequence variants existing among different rDNA repeats.

2.5 Discussion

fiber-views addresses a key challenge in Fiber-seq analysis by organizing multi-layered chromatin data into annotated matrices that preserve metadata through analytical workflows. By adapting the AnnData framework to accommodate Fiber-seq specific features, fiber-views enables matrix-based analyses that were previously difficult to implement consistently.

fiber-views enables several analytical approaches that benefit from matrix representations of chromatin data. Clustering methods can identify groups of fibers with similar accessibility patterns or sequence composition, revealing heterogeneity in chromatin states at individual loci. The k-mer counting functionality supports sequence-based clustering to identify groups of fibers with similar underlying sequences aligning to a single genomic loci. The matrix format can also facilitate pre-processing Fiber-seq data for machine learning applications. The utility of fiber-views has been demonstrated in published work. Bohaczuk et al. (2024) [26] used fiber-views for clustering fibers with similar chromatin accessibility patterns. Bubb, Hamm et al. (2025) [4] applied fiber-views for creating aggregate accessibility plots, splitting accessible patches by size categories, and calculating custom accessibility scores. These applications illustrate how fiber-views facilitates a variety of different analytical approaches.

This chapter has delved into the motivation behind creating fiber-views, as well as describing the implementation of the package. Complete documentation is available at the project repository (https://github.com/MorganHamm/fiber-views) and Read the Docs site (https://fiber-views.readthedocs.io/en/latest/). Both resources are publicly accessible. The package is released under an open-source license and is available for use and modification by other researchers.

Chapter 3

The regulatory potential of transposable elements in maize

This chapter includes materials originally published in:

Bubb, K., Hamm, M. et al. *The regulatory potential of transposable elements in maize*

Nat. Plants 11, 1181–1192 (2025) [4].

Referenced supplemental or extended data figures can be found in Appendix B

3.1 Introduction

Transposable elements (TEs), first described as 'controlling elements' by Barbara McClintock [28]–[34], have the potential to shape the regulation of the host genome [35]–[38]. For example, the insertion of a TE in a regulatory region of the maize domestication gene *teosinte branched1* (*tb1*) enhances its expression, contributing to the increased apical dominance of maize compared with its ancestor teosinte [17]. Although over 80% of the maize genome is annotated as intact TEs or TE fragments [39], a comprehensive analysis of their regulatory potential is lacking. Commonly used methods to map regulatory elements (that is, accessible chromatin regions, ACRs) have relied on short sequence reads which rarely map uniquely within TEs. Here we use the long-read method Fiber-seq to overcome this limitation and map ACRs across the

maize B73 genome. Fiber-seq uses a non-specific DNA N^6-adenine methyltransferase to methylate accessible adenines [40]—a modification that is extremely sparse in plants [41], including maize (Extended Data Fig. 3.1a)—followed by single-molecule PacBio sequencing of ĩ8kb maize chromatin fibres, enabling the synchronous detection of accessible adenines (m6A) and endogenous cytosine methylation (5mCpG).

3.2 Assessing the single-molecule regulatory landscape of maize

We compared Fiber-seq and assay for transposase-accessible chromatin using sequencing (ATAC-seq) using paired samples of leaf protoplasts isolated from 14-day-old dark-grown maize seedlings (Fig. 3.1a and Extended Data Fig. 3.1b–d). The use of leaf protoplasts minimized cell-type heterogeneity as leaf tissue is enriched in mesophyll cells. We observed that Fiber-seq-derived m6A and 5mCpG calls showed the expected signals at ATAC-seq-derived ACRs and cap analysis of gene expression sequencing (CAGE)-defined transcription start sites (TSSs), in addition to the expected correlation of signal intensity with gene expression at TSSs (Extended Data Fig. 3.1e–h). However, unlike ATAC-seq, Fiber-seq also revealed periodic m6A signals downstream of the TSS that were most pronounced for highly expressed genes, reflecting promoter-proximal well-positioned nucleosomes typically measured by micrococcal nuclease sequencing (MNase-seq) (Extended Data Fig. 3.1f,g) [42].

To rigorously distinguish regions with elevated exogenous m6A signal (methyltransferase-sensitive patches, MSP) due to nucleosome linkers from regions representing ACRs (Extended Data Fig. 3.1g), we called FIRE elements (Fiber-seq Inferred Regulatory Elements) with the semi-supervised machine learning classifier 'fiberseq-FIRE' [43]. After recalibrating fiberseq-FIRE for maize, 4.6 million methyltransferase-sensitive patches were classified as actuated FIRE elements (precision >0.9), with the remaining 150 million classified as nucleosome linkers. By aggregating single-molecule FIRE elements across the genome, we called 106,867 FIRE ACRs (false discovery rate (FDR)<0.01, Fig. 3.1b, and Supplementary Tables 2 and 3). In contrast, we called only 51,817 ACRs with ATAC-seq (q-value<0.01, Supplementary Table 3), consistent with Fiber-seq revealing a more comprehensive regulatory landscape of maize. Fiber-seq identified the vast majority of ACRs called with ATAC-seq in paired samples (Fig. 3.1b), added ACRs in repeat regions with low mappability, and corrected for false-positive ATAC ACRs, such as those in nuclear genomic regions

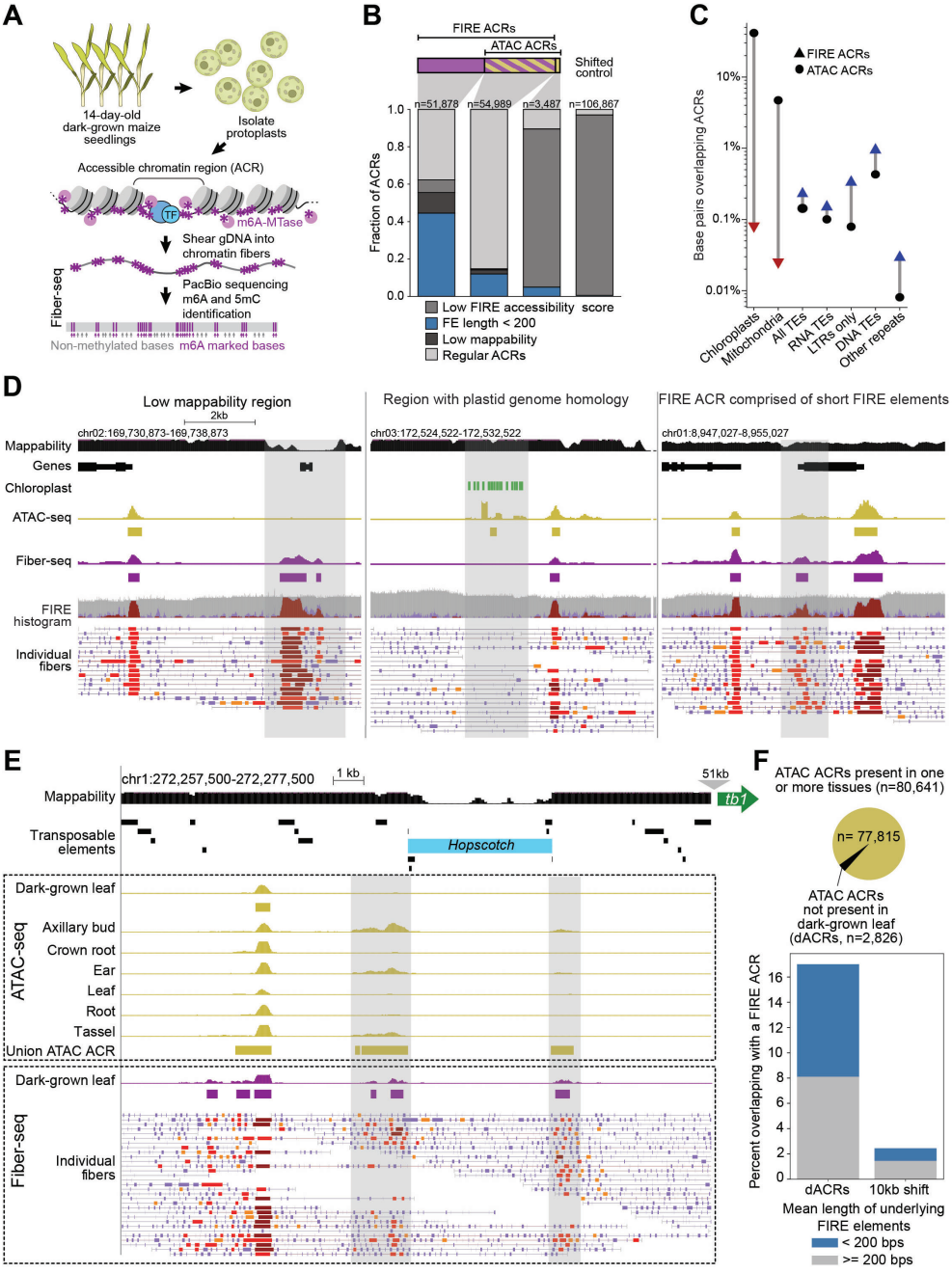

Fig. 3.1: (Caption next page.)

Fig. 3.1: (Previous page.) **a**, Experimental scheme. **b**, ACRs called in paired Fiber-seq and ATAC-seq experiments shown in bar graphs representing FIRE ACRs (purple) that did not overlap with ATAC ACRs (n=51,878), FIRE ACRs that overlapped with ATAC ACRs (purple/gold, n=54,989), ATAC ACRs that did not overlap with FIRE ACRs (gold, n=3,487), and bar graph representing shifted control regions (10kb downstream of FIRE ACRs, n=106,867). ACRs in each category were hierarchically classified as having (1) a FIRE accessibility score of <0.25, (2) a mean FE length <200bp, (3) low short-read mappability or (4) none of the above (regular ACR). Stacked bar charts indicate the distribution of these classifiers for each ACR category. **c**, Percentage of base pairs (y axis) overlapping with ACRs called by either Fiber-seq (FIRE ACRs, triangles) or ATAC-seq (ATAC ACRs, circles) for distinct genomic regions (x axis). **d**, Screenshots of three genomic regions illustrating marked differences between FIRE and ATAC ACR calls. Top to bottom tracks: genomic location, mappability calculated as in Extended Data Fig. 2a, annotated genes, chloroplast sequence, ATAC-seq signal with ATAC ACRs indicated below as rectangles in gold, Fiber-seq signal with FIRE ACRs indicated below as rectangles in purple, FIRE histogram and individual chromatin fibres with FIRE elements in shades of red, with darker shades indicating greater significance. Left: the region highlighted in grey contains two FIRE ACRs but no ATAC ACRs because of low mappability. Middle: the chloroplast sequence track indicates high sequence homology at this nuclear locus with the plastid genome. The ATAC ACR in the highlighted region is a false positive due to incorrect mapping of short sequence reads. Right: the highlighted region shows a FIRE ACR with underlying short FIRE elements. No ATAC ACR was called. Individual fibres are annotated with MSPs (light purple) and FIRE elements (reds, FDR$\leq$5%; oranges, 5%<FDR$\leq$10%). **e**, Loci lacking ATAC ACRs in dark-grown maize leaves that show ATAC ACRs in other tissues often overlap with FIRE ACRs composed of short FIRE elements. A screenshot is shown for the region upstream of the tb1 gene (green arrow, Zm00001eb054440) in which a hopscotch TE insertion (light blue) generated an enhancer. Top to bottom tracks: genomic locus; mappability as in Extended Data Fig. 2a; ATAC-seq data (in first dashed box): ATAC-seq signal (gold) for dark-grown leaves (this study), subsequent tracks pseudobulked single-cell ATAC-seq signal for indicated tissues19, and union ATAC ACRs (last track) (present in at least one tissue) as golden rectangles; Fiber-seq data (in second dashed box): Fiber-seq signal (purple) in dark-grown leaves and indicated below FIRE ACRs as purple rectangles, individual fibres with short FIRE elements in shades of red. In dark-grown leaves, ACRs were detected by Fiber-seq but not ATAC-seq in the loci flanking the hopscotch TE. However, ATAC ACRs were detected in these loci in axillary bud, ear and tassel tissue. f, Of the 80,641 union ATAC ACRs across these seven tissues, 2,826 were not detected in dark-grown leaves (differentially accessible ACRs, dACRs, see Methods for details). About 17% of the loci overlapping with these differentially accessible ACRs overlap with FIRE ACRs, and about half of these are FIRE ACRs composed of short FIRE elements. Statistical analyses and P values for Fig. 3.1b,f are in Supplementary Table 1.

with homology to plastid or mitochondrial genomes [44] (Fig. 3.1c,d, and Supplementary Tables 4 and 5). Signal intensity at ACRs in the paired bulk ATAC-seq strongly correlated with the Fiber-seq signal (Extended Data Fig. 3.1i). However, for a set of $\tilde{4}$0,000 shared ACRs, most were detected with Fiber-seq on half or more of the sequenced chromatin fibres, whereas fewer than 5% of cells showed Tn5 insertions in these ACRs in single-cell ATAC-seq [45] (Extended Data Fig. 3.1j and Supplementary Table 6). This comparison illustrates the limitations of single-cell ATAC-seq as a quantitative measure of per-molecule chromatin

accessibility. Taken together, our results show that Fiber-seq accurately captures chromatin accessibility and 5mCpG in maize, with single-molecule and single-nucleotide precision, at a sensitivity twice that of ATAC-seq.

3.3 Short FIRE elements indicate ATAC ACRs in other tissues

The nearly twice as many ACRs identified by Fiber-seq as compared with the paired ATAC-seq ACRs were only in part explained by low mappability of ATAC-seq reads (Extended Data Fig. 3.2a,b). Rather, over half of the FIRE ACRs missed by ATAC-seq were composed of short FIRE elements (<200bp) (Fig. 3.1b and Extended Data Fig. 3.2c,d). These FIRE ACRs shared the features typical of ACRs comprising long FIRE elements identified by both methods, such as enrichment of the 6mA signal, depletion of the 5mCpG signal (Extended Data Fig. 3.2e) and genomic distribution (Extended Data Fig. 3.2r).

We detected ACRs composed of short FIRE elements flanking the TE insertion that introduced a *tb1* enhancer [17] but failed to detect these by ATAC-seq (Fig. 3.1e). However, these flanking regions were detected as ATAC ACRs in embryonic and reproductive tissues (axillary bud, tassel and ear) [45], suggesting that ACRs comprising short FIRE elements may mark genomic loci with tissue-specific chromatin accessibility in maize (Fig. 3.1e). To systematically evaluate this possibility, we identified ATAC ACRs present in one or more tissues (that is, union ACRs) [45], and then filtered for the subset of these union ACRs for which the corresponding genomic loci showed only background ATAC signal in dark-grown leaves (that is, differential ACRs not present in dark-grown leaves, dACRs), yielding 2,826 dACRs from the total of 80,641 union ACRs (Supplementary Tables 7, 8). Of the 2,826 dACRs, 480 overlapped with a FIRE ACR (17%) and over half of the 480 overlapped with FIRE ACRs comprising short FIRE elements (251/480, Fig. 3.1f). This result is thus consistent with these ACRs composed of short FIRE elements corresponding to functional regulatory elements in maize that display tissue-selective activity.

3.4 Distinctive ACRs mark functional LTR retrotransposons

We next sought to interrogate ACRs in TEs, focusing on long terminal repeat (LTR) retrotransposons because of their prevalence in the maize genome (74.4%) [39]. Intact LTR retrotransposons are class I TEs with

bilateral LTRs that flank an internal region (Fig. 3.2a,b). Each of the bilateral LTRs are thought to contain the regulatory elements, promoters and adjacent enhancers that drive expression of the TE genes encoded in the internal region [46]. LTR retrotransposons mobilize through reverse transcription of their mRNA and integration of the complementary DNA into another genomic location. They are divided into autonomous (which encode the proteins needed for transposition) and non-autonomous (which require proteins encoded by other elements for transposition) LTR retrotransposons. It has been challenging to determine the functional activity of individual LTR retrotransposons because their high sequence identity limits the ability of short-read data to be uniquely mapped to individual LTR retrotransposons [36], [38], [46].

Using Fiber-seq, we mapped ACRs residing within each of the 51,882 intact LTR retrotransposons in the maize genome (Supplementary Table 9), as well as ACRs in solo LTRs (Extended Data Fig. 3 and Supplementary Table 10). Only 2% (941/51,882) of intact LTR retrotransposons contained at least one FIRE ACR entirely within one of their bilateral LTRs (Supplementary Table 9), consistent with widespread epigenetic silencing of maize LTRs by RNA-mediated DNA methylation, a plant-specific pathway that targets TEs [47]. Of the 941 ACR-containing LTR retrotransposons, 21% (201/941) contained two adjacent ACRs (paired ACRs, Fig. 3.2a and Supplementary Table 9) in one or both of their LTRs. Furthermore, 94 of these contained the two adjacent ACRs in both LTRs (paired bilateral ACRs, Supplementary Table 9).

The paired bilateral ACRs almost always exhibited single-molecule co-accessibility and hypo-5mCpG methylation (Fig. 3.2a). Given their position relative to the 5' end of the transposon and the putative transcription start site, these ACRs probably correspond to the putative LTR promoter and enhancer elements [46]. Consistent with this assumption, the putative promoters showed far higher predicted promoter scores than the putative enhancers when using a validated convolutional neural net model for promoter strength [48] (Extended Data Fig. 4a). Furthermore, the putative LTR promoter and enhancer elements are enriched in distinct transcription factor (TF) motifs, consistent with them having distinct regulatory roles (Supplementary Tables 11–14). To further validate whether the putative LTR enhancers and promoters have distinct regulatory roles, we applied FiberHMM [42] to call protein occupancy footprints within them. Previous studies using Fiber-seq show that eukaryotic RNA polymerases display MTase footprints of 40–60bp size, which is distinct from those of TFs, which are typically <40bp in size [38], [49]. Using this approach, we observed that LTR putative promoters were uniquely marked by MTase footprints of 40–60bp

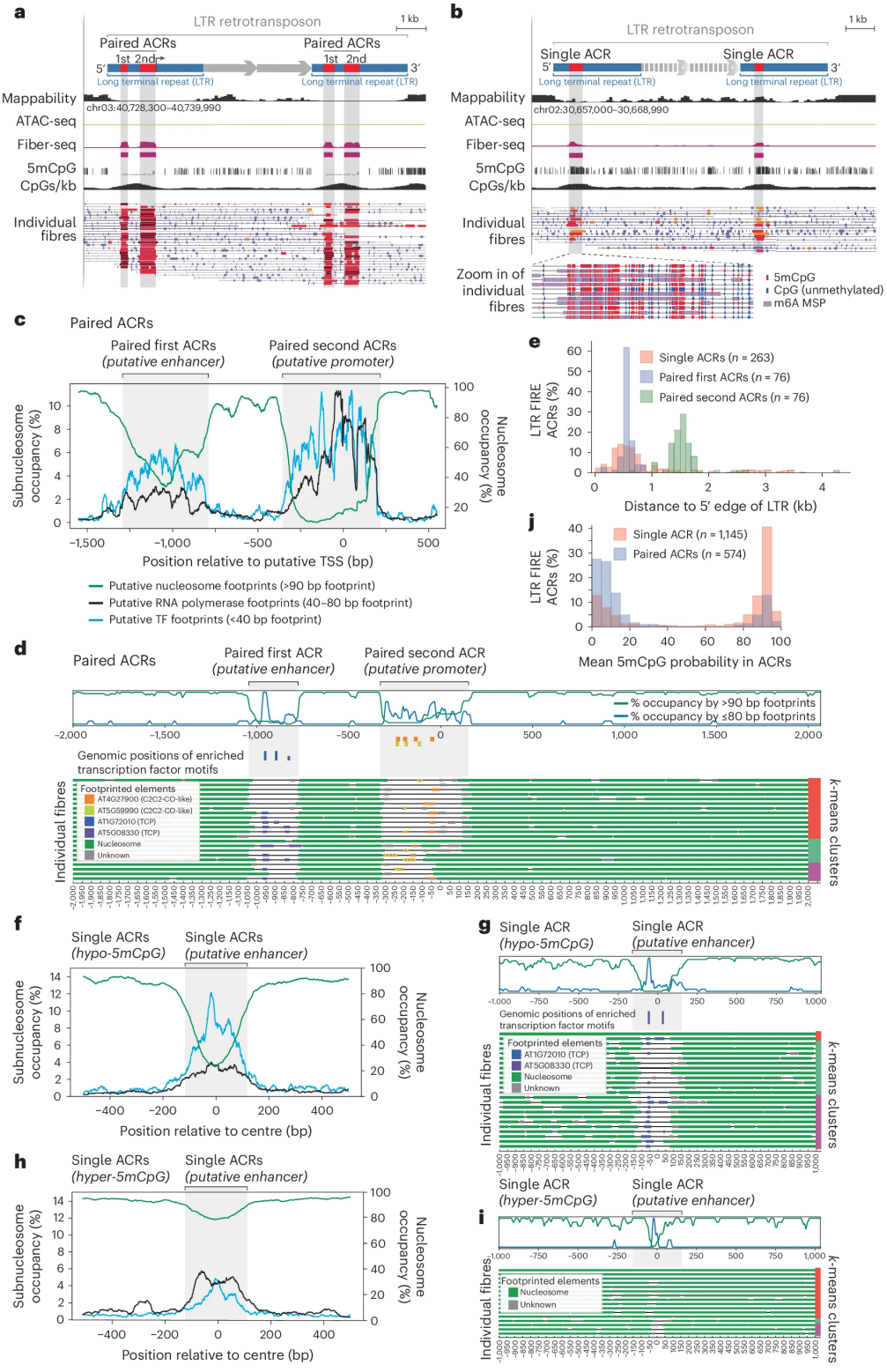

Fig. 3.2: (Caption next page.)

Fig. 3.2: (Previous page.) **a**, Representative example of an intact LTR retrotransposon with paired FIRE ACRs both in the left and the right long terminal repeats (paired bilateral ACRs in LTRs, ID: LTRRT_14411). Putative transcription start site is indicated with black arrow, and genes in internal region are indicated. Paired bilateral ACRs in LTR retrotransposons tended to be hypomethylated as expected for accessible regions. **b**, Representative example of an intact LTR retrotransposon with one FIRE ACR both in the left and the right LTR (single bilateral ACRs in LTRs, ID: LTRRT_8308). Tracks as in a. The single FIRE ACRs in this LTR retrotransposon showed high levels of 5mCpG methylation coinciding with the m6A signal; magnified detail below shows methylated 5mCpGs (red) and unmethylated CpGs (blue) and m6A methyltransferse-sensitive patches (purple) on individual fibres. **c**, Plot depicting the aggregate percent occupancy of nucleosome-sized footprints (>90bp, green), putative polymerase-sized footprints (40–80bp, dark grey) and putative TF-sized footprints (10–40bp, azure blue) within the putative promoters of all paired ACRs in LTRs. On the x axis, zero indicates a selected base proximal to the 3' end of the putative promoter (typically 75bp upstream) that is highly conserved across all putative promoters. **d**, Representative example of footprinted Fiber-seq reads at a paired ACR (zero: chr05:195,444,369, reverse strand). Top: tracks showing the count of subnucleosomal sized footprints (<80bp, dark blue) and nucleosome-sized footprints (>90bp, green) at each position normalized to the maximum count within the region. Middle: blocks showing the genomic position of enriched TF motifs, coloured by identity. Bottom: individual Fiber-seq reads with footprints represented by blocks coloured by predicted identity (accessible, thin black line; nucleosome, green; unknown, grey; TF, coloured corresponding to overlapped motif with multiple overlapped motifs indicated via stripes). Reads are clustered and sorted via k-means clustering with 3 clusters, indicated via a column of colours at right. **e**, LTR retrotransposons with single bilateral FIRE ACRs tended to maintain the ACRs marking putative enhancers. Histogram of FIRE ACR location relative to the 5' edge of a given retrotransposon, stratified by type of ACR. **f**, Plot depicting the aggregate percent occupancy of nucleosome-sized footprints (>90bp, green), putative polymerase-sized footprints (40-80bp, dark grey) and putative TF-sized footprints (10–40bp, azure blue) at positions surrounding the centre of all hypo-5mCpG-methylated single ACRs. **g**, Representative example of footprinted Fiber-seq reads at a single hypomethylated ACR (chr04: 170,173,642–170,173,927, forward strand), with position indicated relative to the centre of the ACR. Tracks as in d. **h**, Plot depicting the aggregate percent occupancy of nucleosome-sized footprints (>90bp, green), putative polymerase-sized footprints (40–80bp, dark grey) and putative TF-sized footprints (10–40bp, azure blue) at positions surrounding the centre of all hyper-5mCpG-methylated single ACRs. **i**, Representative example of footprinted Fiber-seq reads at a single hypermethylated ACR (chr04: 206,713,520–206,713,627, forward strand), with position indicated relative to the centre of the ACR. Tracks as in d, with the motif track omitted due to a lack of motifs. **j**, Single FIRE ACRs in LTR retrotransposons were more likely to be 5mCpG methylated than paired bilateral FIRE ACRs, regardless of their position.

size (Fig. 3.2c), consistent with these elements being RNA polymerase-bound promoters. In contrast, the LTR putative enhancers were largely marked by MTase footprints of <40bp size that were well aligned with enriched TF motifs (Fig. 3.2c,d), consistent with them being occupied by TFs. The putative enhancer and promoter ACRs were generally separated by 3 or 4 well-positioned mononucleosome footprints, with the regions upstream and downstream of the putative promoter showing larger, less clearly phased di- and trinucleosome footprints, typically associated with heterochromatin (Extended Data Fig. 5a). Although the

putative enhancer and promoter ACRs were nearly constitutively accessible on all fibres, protein occupancy by 40–60bp footprints and <40bp footprints varied substantially between them (Fig. 3.2c).

Together, these findings demonstrate that Fiber-seq enables the identification of individual LTRs within the maize genome that contain active chromatin at both the putative enhancers and promoters, consistent with these LTRs being poised to be functionally active. In total, in maize leaves there are only 94 LTR retrotransposons that contain the functional regulatory elements required for transposon mobilization, with only 76 of these being autonomous LTR retrotransposons.

3.5 Single LTR ACRs are putative enhancers

Most LTR retrotransposons that harbour an ACR contained only a single ACR in one or both of their LTRs (Fig. 3.2b and Supplementary Table 9). Intact LTR retrotransposons with single ACRs were enriched for containing a single ACR in both of their bilateral LTRs (that is, single bilateral ACRs, 499/941, 53%). The single ACRs exhibited far greater single-molecule heterogeneity than paired ACRs (Extended Data Fig. 3.4b). Specifically, while paired ACRs showed a bimodal actuation distribution with over half being supported by FIRE elements called in 75% of underlying fibres, only 7% of single ACRs crossed this actuation threshold (Extended Data Fig. 3.4b).

We next examined whether single ACRs preferentially localized to the putative LTR enhancer or to the putative LTR promoter. To accomplish this, we first examined the distance of the ACR to the 5' LTR edge, which could be measured within the subset of 268 autonomous LTR retrotransposons containing an ACR, as strandedness could be inferred at these sites. We observed that nearly all LTR retrotransposons with a single ACR selectively retained the ACR that positionally corresponds to the putative LTR enhancer element (Fig. 3.2e), suggesting that chromatin accessibility is lost at the position of the putative LTR promoters.

Single ACRs contained predicted TF motifs that were more similar to those enriched in the putative LTR enhancers than to those enriched in the putative LTR promoters of paired ACRs (Extended Data Fig. 4c and Supplementary Tables 11–14). Consistently, the single ACRs showed small TF footprints (<40bp) that were well aligned with the enriched TF motifs (Fig. 3.2f,g and Supplementary Tables 11–14). Per-fibre chromatin accessibility at these single ACRs was more heterogeneous than that observed at putative enhancers of paired

ACRs (Fig. 3.2f–i and Extended Data Fig. 4b). However, the occupancy of small footprints along accessible fibres within these single ACRs was consistent with that of the putative LTR enhancers of paired ACRs (Fig. 3.2f–i).

In stark contrast to paired ACRs in LTRs, or ACRs elsewhere in the maize genome, we observed that a subset of single ACRs in LTRs exhibited hyper-5mCpG methylation coinciding with chromatin accessibility (Figs. 3.2b,j and 3.3a), two epigenetic marks thought to be mutually exclusive. Leveraging the single-molecule nature of our chromatin accessibility and 5mCpG methylation calls, we demonstrated that chromatin accessibility and hyper-5mCpG methylation co-occurred and overlapped along the same chromatin fibre at these single ACRs of LTR retrotransposons (Fig. 3.2b and Supplementary Table 15). Hyper-5mCpG-methylated ACRs and hypo-5mCpG-methylated ACRs showed similar TF footprint occupancy in accessible fibres (Extended Data Fig. 5b).

As expected, hyper-5mCpG methylation was rarely seen overlapping FIRE ACRs elsewhere in the maize genome. The rare ACRs with simultaneous hyper-5mCpG methylation and chromatin accessibility were almost exclusively present within repeat elements (Fig. 3.3b), with 15% of these corresponding to single ACRs in intact LTR retrotransposons. In humans, this unexpected co-occurrence of chromatin accessibility and hyper-5mCpG methylation was also rare but not substantially enriched in repeats (Fig. 3.3c). These results point to the plant-specific RNA-mediated DNA methylation pathway as contributing to this unusual co-occurrence of these two epigenetic marks in maize. However, further analysis of the methylation signatures typical of RNA-mediated DNA methylation or other chromatin states at these low-mappability loci was not feasible because the publicly available data sets resulted from short-read sequencing [51].

Given the distinct chromatin features of LTRs containing paired and single ACRs, we reasoned that LTR retrotransposons containing the former might be evolutionarily younger TEs, while TEs containing the latter might be older but still younger than the many fully silent transposons. To address evolutionary age, we examined the sequence similarity between the left and the right LTR of each intact LTR retrotransposon as a metric reflecting time since transposition. LTR retrotransposons with exactly one FIRE ACR per LTR (single ACR) showed greater mean sequence similarity than those without FIRE ACRs (99.0% versus 98.7%, P=0.004, Mann–Whitney U-test) (Fig. 3.3d). LTR retrotransposons with exactly two FIRE ACRs per LTR

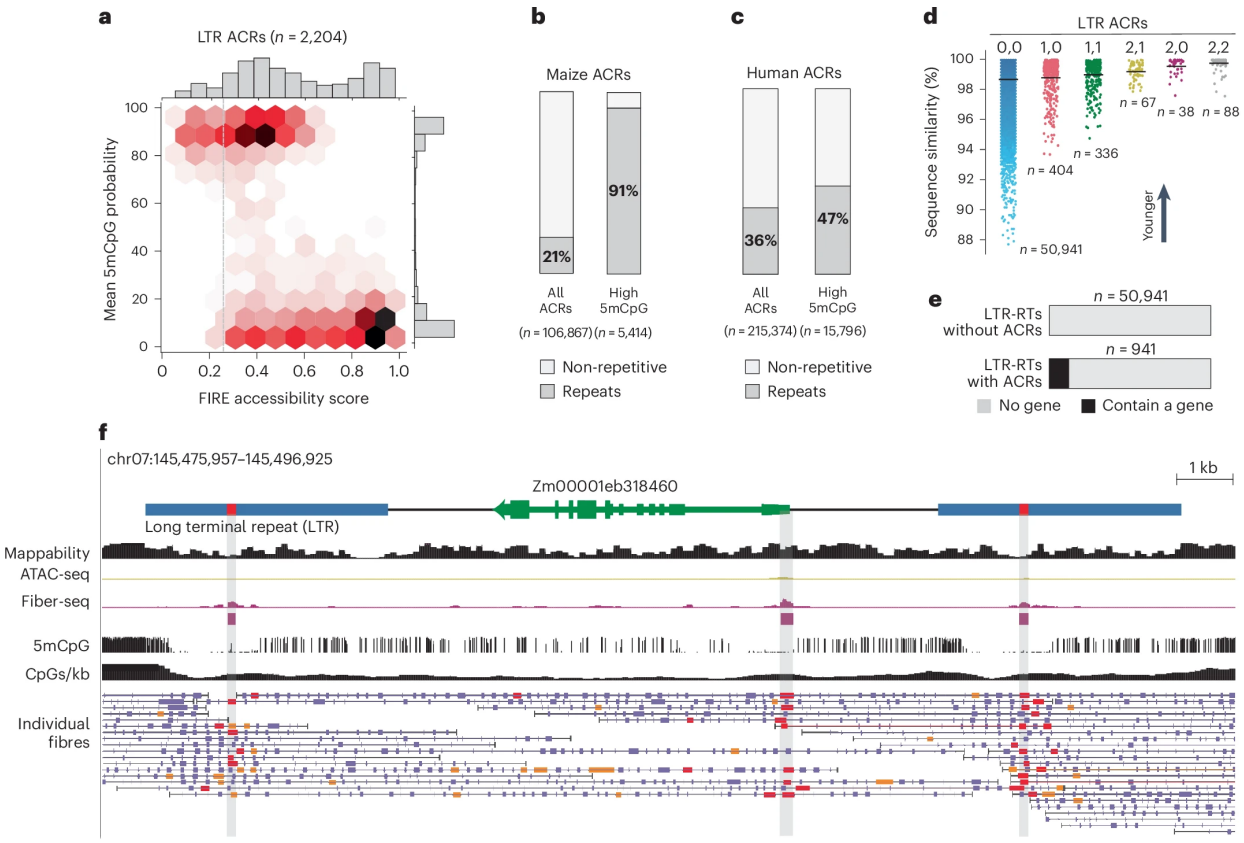

Fig. 3.3: **a**, Hexbin plot shows FIRE accessibility scores (x axis) and mean 5mCpG probabilities (y axis) for 2,204 ACRs in LTRs. Shades of red denote frequency, also shown in plotted histograms (top and right). Of FIRE ACRs in LTRs, 57% (1,261/2,204) were highly 5mCpG methylated and 37% (733/1,978) of high-confidence FIRE ACRs within LTRs (FIRE accessibility score >0.25, grey dashed line) were highly 5mCpG methylated. **b**, Percentage of all FIRE ACRs and FIRE ACRs with high 5mCpG methylation (mean 5mCpG methylation >50%) that overlap with an annotated repeat by >50bp. **c**, Fraction of all human FIRE ACRs and human FIRE ACRs with high 5mCpG methylation (mean CpG methylation of >50%) that overlap an annotated repeat by >50bp. FIRE ACR calls from human cell line GM12878 (ref. [50]). **d**, The presence of FIRE ACRs correlates with the sequence similarity of left and right LTRs, a measure of evolutionary age. LTRs with paired bilateral FIRE ACRs showed the greatest sequence similarity, while those without FIRE ACRs showed the least. 0,0, no ACRs; 1,0, single unilateral ACR; 1,1, single bilateral ACRs; 2,1, paired ACR in one LTR, single in the other; 2,0, paired ACR in one LTR, none in the other; 2,2, paired bilateral ACRs. Rare instances of other configurations are omitted. **e**, Fraction of intact LTR retrotransposons with or without at least one FIRE ACR that contain an annotated gene. **f**, The highly duplicated, well-annotated gene (Zm00001eb318460, green) within an LTR retrotransposon is a candidate for TE-enabled gene amplification. Tracks as in Fig. 3.2a. There are single bilateral ACRs present in the LTRs, in addition to a FIRE ACR marking the transcription start site of this gene (highlighted in grey). Statistical analyses and P values for Fig. 3.3b–e are in Supplementary Table 1.

(paired ACRs) showed a mean sequence similarity of 99.8%, significantly greater than those with one FIRE ACR per LTR (P=6.8×10^{-26}, Mann–Whitney U-test) (Fig. 3.3d). Thus, recently transposed LTR retrotransposons have a characteristic chromatin accessibility and 5mCpG methylation pattern that degenerates with evolutionary age.

3.6 LTRs can contain active promoters and propagate host genes

TEs have been long thought to shape host gene regulation by adding or disrupting promoters, enhancers, insulators and coding regions [51], [52]. In humans, TE-derived promoters have been inferred via mapping of transcription start sites [53] and TF binding sites overlapping TE sequences [54], [55]. However, in maize, attempts to infer the regulatory effects of LTR retrotransposons have largely been limited to studying gene expression patterns associated with the presence or absence of neighbouring polymorphic TEs that overlap ATAC ACRs [54]—an analysis that is severely limited by short-read mappability issues inherent to TEs. We sought to leverage our comprehensive maps of FIRE ACRs across intact maize LTR retrotransposons to identify LTRs that may be shaping host gene regulation (Extended Data Fig. 6a,b). Using this approach, we discovered that the putative target gene impacted by LTR FIRE ACRs often resided within the LTR retrotransposon itself. In fact, of the 941 LTR retrotransposons with ACRs in B73, 114 (12%) contained an annotated gene within the intact retrotransposon (Extended Data Fig. 6c,d), 24-fold greater than LTR retrotransposons without ACRs (Fig. 3.3e). Of these 114, the promoters of 49 annotated genes were marked by a FIRE ACR, with 48 co-opting one of the LTR ACRs as their promoter (Extended Data Fig. 6d). Overall, these findings indicate that a major way that LTRs shape maize host gene regulation is by providing a novel LTR-encoded promoter element.

In one case, the internal gene (Zm00001eb318460) maintained its promoter, marked by a FIRE ACR, in addition to single FIRE ACRs in the flanking bilateral LTRs (Fig. 3.3f). This histone deacetylase complex gene is highly expressed (79% percentile in dark-grown maize leaves) [56] and has orthologues in the close maize relative *Sorghum bicolor*, its ancestor teosinte (*Zea mays* ssp. *mexicana*) [57] and other grasses. In human, Alu TEs have been proposed to enable segmental duplication [58], so we sought to evaluate whether the gene's residence within the LTR retrotransposon might be associated with its duplication within the B73

genome. Consistent with this hypothesis, we found numerous Zm00001eb318460 paralogues in the B73 genome (21 amino acid blast hits with *e*-value<1×10^{-50}). This is a highly unusual level of gene duplication for maize genes with a similar expression level and length, as 93% of similar genes showed <5 amino acid blast hits (Supplementary Table 16). Taken together, these findings provide evidence that TEs enable gene amplification in maize.

3.7 Diffuse chromatin accessibility marks insertions of hAT TEs

In general, transposons have been shown to preferentially insert into accessible chromatin both in vitro and in vivo [59], [60]. However, the epigenetic features that predispose genomic loci to insertion of class II (DNA) TEs, in particular hAT TEs, remain unresolved. The first gene reported by McClintock to be susceptible to insertion of hAT TEs is the *C* locus [59]–[63], now called *C1* or *coloured aleurone 1* gene. The *C1* gene body showed unusual diffuse Fiber-seq chromatin accessibility (Fig. 3.4a), ranking among the top 5% of all genes (Fig. 3.4b). We also observed hypo-5mCpG methylation across the *C1* gene body, ranking among the bottom 5% of all genes (Fig. 3.4c). Other genes identified by McClintock as having hAT TE insertions also showed diffuse Fiber-seq chromatin accessibility and hypo-5mCpG methylation within their respective gene bodies (Extended Data Fig. 7) [61]–[65]. While diffuse gene body chromatin accessibility was weakly correlated with gene expression (Fig. 3.4b), this feature was more strongly anticorrelated with hypo-5mCpG methylation (Fig. 3.4c) [50], [66]–[69]. These findings suggest that both diffuse gene body chromatin accessibility and hypo-5mCpG methylation may uniquely mark preferred hAT TE insertion sites. To test this hypothesis, we identified over 32,000 loci in the B73 maize reference genome that contain hAT TE insertions in exactly 1 of the 25 non-B73 NAM founder lines [70]. Indeed, the hAT TE insertion sites were substantially more accessible in B73 than control regions (Fig. 3.4d and Supplementary Table 17) and were preferentially marked by hypo-5mCpG methylation (Supplementary Table 17). We thus show that the pervasive epigenetic marks of diffuse Fiber-seq chromatin accessibility and hypo-5mCpG methylation guide the insertional landscape of hAT TEs in maize, including those initially described by McClintock.

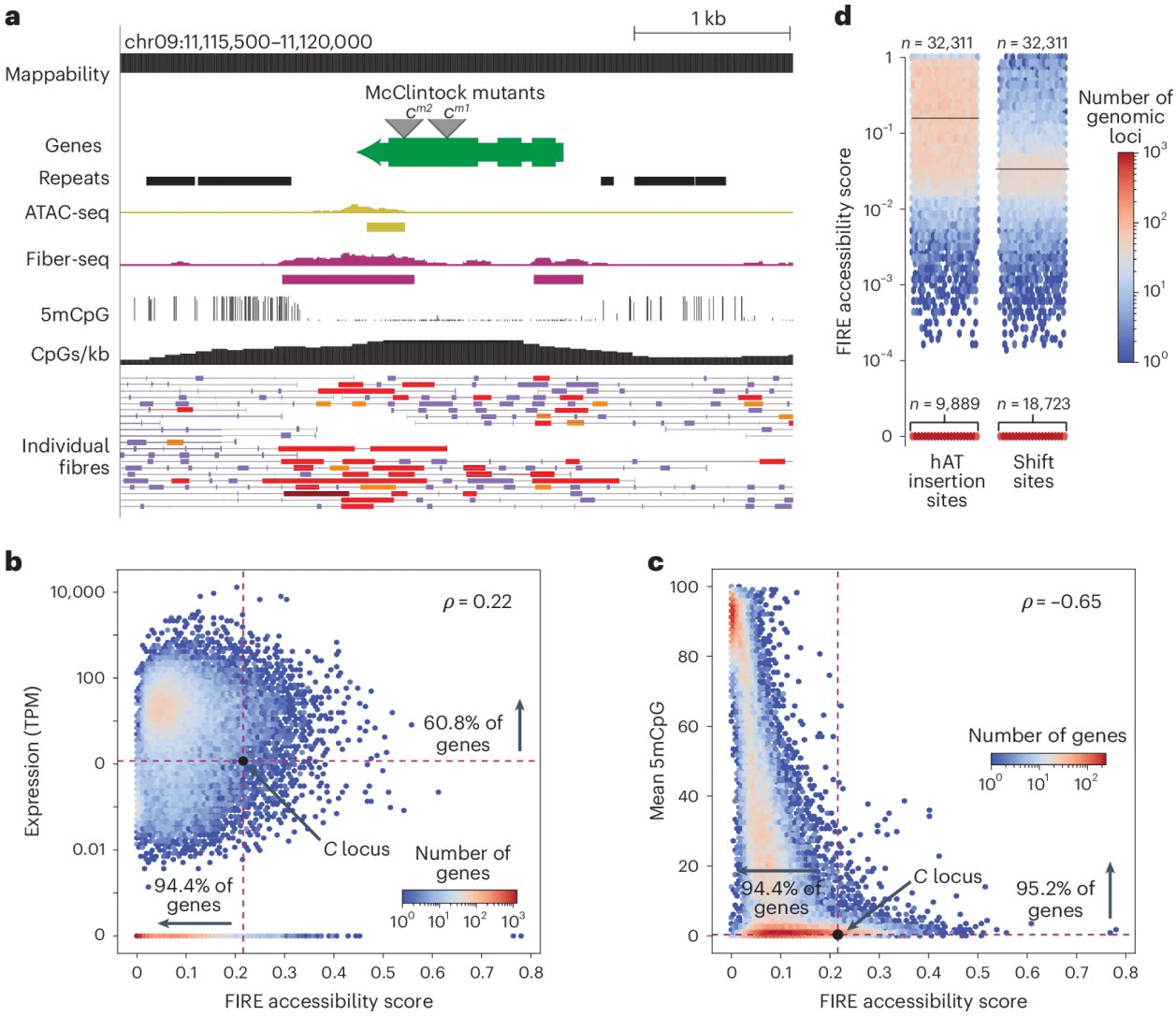

Fig. 3.4: **a**, Screenshot of the locus containing the C1 gene (green, Zm00001eb373660). Tracks as in Fig. 3.2. hAT TE insertions identified by McClintock as mutant alleles cm1 and cm2. The C1 gene in B73, which does not contain a hAT TE, showed diffuse gene body accessibility detected in Fiber-seq (purple) coinciding with unusual gene body hypo-5mCpG methylation. **b**, The C1 gene body showed higher FIRE accessibility scores than 94.4% of other genes while showing only slightly above average gene expression. Dashed red lines signify location of C locus. **c**, The C1 gene body showed lower 5mCpG gene body methylation than 95.2% of genes. d, Loci identified as hAT TE insertion sites in exactly 1 of the 25 non-B73 NAM strains45 were more likely to show diffuse gene body accessibility (shown as FIRE accessibility scores) than control loci (same-size regions 10kb shifted to the right in the B73 genome). Mean and median FIRE accessibility scores for hAT TE insertion sites were 0.159 and 0.046. Mean and median FIRE accessibility scores for shifted control regions were 0.035 and 0. Horizontal lines indicate mean FIRE accessibility scores. Thousands of hAT TE insertion sites showed FIRE accessibility scores of 0 (n=9,889). Statistical analysis and P value for Fig. 3.4d are in Supplementary Table 1.

3.8 Discussion

The contemporary maize genome is largely composed of repetitive DNA, with the crop's $\tilde{4}$0,000 genes clustered in small islands of non-repetitive sequence [71]. The promise of long-read sequencing for probing regulatory activity in repetitive plant genomes has been illustrated by previous studies in maize [71] and *Arabidopsis* [72]. However, the reduced accuracy associated with these previous studies' reliance on single-pass long-read sequencing [24] has limited their ability to perform an in-depth de novo characterization of individual regulatory elements and their associated single-molecule protein occupancy patterns within transposons. Using Fiber-seq paired with PacBio HiFi sequencing in dark-grown maize leaves, we perform de novo identification of ACRs in maize, identifying twice as many ACRs as compared with using ATAC-seq in paired samples. This dramatic increase in ACR calls is primarily due to the experimental issues with ATAC-seq for detecting short regulatory elements. The ACRs specific to Fiber-seq share the genomic and epigenetic features of ACRs identified by both methods. Furthermore, they often mark loci that show a strong ATAC-seq signal in other tissues. We attribute this phenomenon of 'ACR foreshadowing' to the less frequent and more stochastic TF occupancy at these sites in leaf tissue, which can be captured by Fiber-seq due to its single-molecule resolution. By contrast, the more frequent and more consistent TF occupancy of these same sites in a different tissue results in ACRs that are detectable by both methods.

Unencumbered by mappability concerns, we discover that only 2% (941/51,882) of intact LTR retrotransposons in the maize genome show evidence of regulatory activity, highlighting the efficiency of the plant-specific RNA-mediated DNA methylation pathway to silence the vast majority of transposons in maize. Furthermore, only 94 intact LTR retrotransposons in the maize genome show chromatin accessibility in LTRs in their putative enhancer and promoter elements, chromatin features presumably essential for enabling transposon mobilization.

The presence and specific configurations of ACRs in LTR retrotransposons is correlated with the evolutionary age of that LTR retrotransposon, with accessibility increasingly lost in older transposons. LTR retrotransposons that lose chromatin accessibility preferentially maintain an accessible putative enhancer element. However, the epigenetic pattern at these putative LTR enhancer elements markedly diverges from that of canonical regulatory elements in the rest of the genome. Specifically, on the same chromatin fibre, these

enhancer elements can show the unexpected co-occurrence of chromatin accessibility and hyper-5mCpG methylation, two epigenetic marks widely thought to be mutually exclusive. This plant-specific epigenetic pattern raises questions about the nature of the DNA-binding proteins generating accessibility at these sites, which we show can directly occupy these hyper-5mCpG-methylated accessible elements. Taken together, these results suggest that plant-specific DNA methylation pathways first suppress the proliferation of LTR retrotransposons by silencing their putative promoters, and subsequently prevent the putative LTR enhancers from being exapted as gene enhancers by hyper-5mCpG methylation. In contrast, we present evidence that a primary mechanism by which LTR retrotransposons may shape maize gene regulation is by creating novel gene promoters from ACRs within the LTRs. We also show an example of an ACR-containing LTR retrotransposon facilitating host gene amplification.

Finally, we find that the loci in which Barbara McClintock discovered TE insertions show unusually low gene body methylation and unusually high gene body accessibility, consistent with the common assumption that gene body methylation protects against TE insertion. While the mechanistic underpinnings of these correlated epigenetic features are unclear, this finding adds to our understanding of the complexity of epigenetics and genome evolution in plants.

3.9 Methods

3.9.1 Maize mesophyll protoplast generation

We used the polyethylene glycol transformation method of maize mesophyll protoplasts as described in ref. [73]. Maize (*Zea mays* L. cultivar B73) seeds were soaked in water overnight at 25°C. The seeds were germinated in soil for 3days under long-day conditions (16h light, 8h dark) at 25°C, then moved to complete darkness at 25°C for 10–11days. From each seedling, 10-cm sections from the second and third leaf were cut into thin 0.5-mm strips perpendicular to veins and immediately submerged in 10ml of protoplasting enzyme solution (0.6M mannitol, 10mM MES pH 5.7, 15mgml^{-1} cellulase R10, 3mgml^{-1} macerozyme, 1mM $CaCl_2$, 0.1% (w/v) BSA and 5mM beta-mercaptoethanol). The mixture was covered in foil to keep out light, vacuum infiltrated for 3min at r.t. and incubated on a shaker at 40rpm for 2.5h at r.t. Protoplasts were released by incubating for an extra 10min at 80rpm. To quench the reaction, 10ml ice-cold MMG (0.6M mannitol,

4mM MES pH 5.7, 15mM $MgCl_2$) was added to the enzyme solution and the whole solution was filtered through a 40-μM cell strainer. To pellet protoplasts, the filtrate was split into equal volumes of no more than 10ml in chilled round-bottom glass centrifuge vials and centrifuged at 100×g for 4min at r.t. Pellets were resuspended in 1ml cold MMG each and combined into a single round-bottom vial. To wash, MMG was added to make a total volume of 5ml, and the solution was centrifuged at 100×g for 3min at r.t. This wash step was repeated two more times. The final pellet was resuspended in 1–2ml of MMG. A sample of the resuspended protoplasts was diluted 1:20 in MMG and used to count the number of viable cells using fluorescein diacetate as a dye.

3.9.2 ATAC-seq data collection

An aliquot of 50,000 isolated protoplasts was added to new tubes and spun down (4°C, 2,000g) for 10min. Supernatant was discarded and the pellet of protoplasts was washed with 750μl of lysis buffer (0.4M sucrose, 10mM $MgCl_2$, 25mM Tris-HCl pH 8.0, 0.1× protease inhibitor, 0.5% Triton X). Samples were then spun down (4°C, 1,500g) for 5min and the supernatant discarded. Samples were then washed once more with buffer (0.4M sucrose, 10mM $MgCl_2$, 25mM Tris-HCl pH 8.0, 0.1× protease inhibitor) at 4°C and 1,500g for 3min to remove the lysis buffer. The nuclear pellet was then resuspended in 22.5 μl double distilled H_20 followed by adding 25μl of 2× TD buffer (20mM Tris-HCl pH 7.6, 10mM $MgCl_2$, 20% (v/v) dimethylformamide) and 2.5μl of Tn5. Samples were then incubated at 37°C for 5min. Reaction was stopped by adding 250μl of Zymo Research DNA Binding buffer and DNA was purified using Zymo research Clean and Concentrator kit. Samples were size selected using 1.8× ampure beads and barcoded with Illumina Nextera Index primers. Final library concentrations were determined using Qubit DNA HS assay, and average fragment length was determined using TapeStation D1000 ScreenTape Assay.

3.9.3 Fiber-seq data collection

Isolated protoplasts (1–5 million) were spun down at 2,000g and resuspended in a 100μl working buffer (400mM sucrose, 15mM Tris-Cl, 15mM NaCl, 60mM KCl, 1mM EDTA, 0.5mM EGTA, 0.5mM spermidine), with 1.5μl of 32mM S-adenosylmethionine added to a final concentration of 0.8mM along with 0.5μl of Hia5 MTase (100U), then carefully mixed by pipetting 10 times with wide bore tips. Reactions were incu-

bated for 10min at 25°C, then stopped with 3μl of 20% SDS (1% final concentration) and transferred to new 1.7ml microfuge tubes. High molecular weight DNA was then extracted using the Promega Wizard HMW DNA extraction kit A2920. PacBio SMRTbell libraries were then constructed using the manufacturer's SMRTbell prep kit 3.0 procedure. Two replicate samples were processed.

3.9.4 Quantification of 6mA/dA by UHPLC–MS/MS

Samples for quantification were treated as previously described [41]. In brief, 50ng of DNA from each sample was mixed with 0.02U phosphodiesterase I (Worthington, LS003926), 1U Benzonase (Millipore Sigma, E1014) and 2.5U Quick CIP (NEB, M0525S) in digestion buffer (10mM Tris, 1mM MgCl pH 8 at r.t.) for 3h at 37°C. Single nucleotides were separated from the enzymes by collecting the flow-through of a Nanosep centrifugal filter (MWCO 3kDa, Pall, OD003C33). The UHPLC–MS/MS analysis of adenosine and m6A was performed on an ACQUITY Premier UPLC System coupled with a XEVO-TQ-XS triple quadrupole mass spectrometer. UPLC was performed on a ZORBAX Eclipse Plus C18 column (2.1×50mm i.d., 1.8μm particle size) (Agilent, 959757–902). 6mA and dA were eluted and separated using 2–50% linear gradient of solvent B (0.1% acetic acid in 100% methanol) in solvent A (0.1% acetic acid in water) within 10min, at a flow rate of 0.3mlmin^{-1}. MS/MS analysis was operated in positive ionization mode with 3,000V capillary voltage, as well as 150°C and 1,000lh^{-1} nitrogen drying gas. A multiple reaction monitoring (MRM) mode was adopted with the following *m*/*z* transition: 252.10–>136.09 for dA (collision energy, 14eV) and 266.2–>150.2 for 6mA (collision energy, 15eV). 6mA and dA were mixed to create standards from 0 to 100nM 6mA, and a new standard was measured and used for each run. MassLynX was used to quantify the data.

3.9.5 Fiber-seq data processing

Fibertools [74] was used to call m6A methylation and label regions as MSPs and nucleosomes on individual reads. Fiberseq-FIRE was used to assign FDR values to MSPs and call Fiber-seq ACRs using a FIRE model customized for the maize genome [75]. Since we trained our model and called FIRE ACRs and elements, the model format used by fiberseq-FIRE has changed. To accommodate these changes, a maize model should be re-trained in the future, which can be readily performed and documented in the fiberseq-FIRE GitHub

repository [43], [76]. For ACR calling, we used the set of peaks identified by the FIRE pipeline with an FDR threshold of 1%. Data for two replicates were combined.

3.9.6 ATAC-seq data processing

ATAC-seq read pairs were aligned to the MaizeV5 reference genome [39] using bwa (v.0.7.17-r118) [77]. The resulting bams were filtered using samtools view [78] to discard reads that were unmapped (-F 4) or had map quality of zero (-q 1). ACRs were called using MACS2 (v.2.2.7.1) [79], and the narrowPeaks output was merged to generate a non-overlapping set of ACRs. The ATAC-seq signal track is a sliding-window histogram displaying the number of ATAC read ends, with the height of each 20-bp bar representing the number of Tn5 insertions within a 100-bp window centred on that 20bp.

3.9.7 RNA-seq data used to define expression quantiles and TSSs

Publicly available RNA-seq reads (64,500,061) were obtained from NCBI SRA: ERR3322830. These reads were derived from the second leaves of 9-day-old etiolated seedlings [80]. Reads were aligned to the maizeV5 annotation using hisat2 and counts were tallied using htseq-count. Transcripts per million (TPM) were calculated for each gene. A total of 13,542 genes had a TPM of zero. The remainder were split into deciles by expression level, with each decile containing 2,991 or 2,992 genes. TSS positions were obtained using CAGE data [81].

3.9.8 Methylation rate (m6A and m5CpG)

For each genomic locus being aggregated, at each 20bp bin, the number of possible methylation sites was calculated from the individual fibre sequences. The observed methylation events were tallied and divided by the number of possible sites to get a fraction of sites methylated.

3.9.9 MSP score and FIRE accessibility score

MSP score is the fraction of fibre-bases within a given region that are annotated as MSPs. FIRE accessibility score is the fraction of fibre-bases within a given region that are annotated as FIRE element (Extended Data Fig. 3.1b).

3.9.10 Percent actuation

For a given genomic region, percent actuation is the number of unique reads (fibres) with at least one FIRE element overlapping the region, divided by the total number of unique reads overlapping the region.

3.9.11 Comparing single-cell ATAC-seq to Fiber-seq

The sparse matrix containing binary (cut or no cut) information for all cells from all tissues and all peaks reported in ref. [45] were downloaded from https://ftp.ncbi.nlm.nih.gov/geo/series/GSE155nnn/GSE155178/suppl/GSE155178%5FACR%5Fx%5Fcell.binary.sparse.txt.gz. We then generated a bed file consisting of only peaks with at least one Tn5 insertion in a leaf-designated cell, and reported, for each peak, the fraction of total leaf cells having one or more Tn5 insertions at that site. We used liftOver [82] to convert the genomic positions of the peaks in this file from V4 to V5 coordinates, then filtered peaks to retain those that (1) overlap by MACS2 peaks by 100bp or more and (2) overlap our FIRE-peaks by 100bp or more; 39,132 peaks remain. The percentage of cells with one or more Tn5 insertions was plotted against the percentage of fibres containing a FIRE element (% actuation) in Fig. 3.1g. For this analysis, 58,712 MACS2 narrowPeaks were called on an alignment file (bam) containing 94,945,002 mapped 50-bp paired-end reads. These peaks were merged (bedops -m), resulting in 50,349 MACS2 peaks used above.

3.9.12 Short-read mappability analysis

A total of 2.1 billion fragments were generated evenly distributed across the B73 reference genome chromosomes 1–10. Fragment lengths were sampled from a log-normal distribution fit to one of our ATAC-seq data sets. For each simulated fragment, a paired-end read was generated with 50 base reads on either end of the fragment. The true start and end of the fragment was encoded in the read name. We did not simulate per-base errors in these reads; each read matches exactly the reference sequence from which it was generated. These reads were then mapped back to the genome using BWA. The 'fraction mapped' for a given region or window was calculated as the number of correctly mapped reads with mapq score >0, divided by the total number of simulated reads with the outer end (Tn5 insertion) falling in the region (Extended Data Fig. 2b).

3.9.13 Annotation of repetitive regions, including all TEs

Annotation file Zm-B73-REFERENCE-NAM-5.0.TE.gff3.gz was downloaded from maizegdb.org [83].

3.9.14 Annotation of regions of the nuclear genome with homology to organellar genomes

Regions of homology within the nuclear genome to organellar genomes were identified as follows for each of the chloroplast and mitochondrial genomes, separately. Paired-end reads were simulated to achieve 100× coverage (142,724 and 579,124 read pairs, respectively), then mapped to the MaizeV5 reference genome [39] using bwa (v.0.7.17-r118) [77]. The resulting bams were filtered using samtools view [78] to discard reads that were unmapped (-F 4), had map quality of zero (-q 1), or mapped to the centromere [84]. Alignment files (bams) were then converted to bed files and overlapping regions were merged.

3.9.15 Classification of ACRs

ACRs were sorted hierarchically as follows: (1) all ACRs with low FIRE accessibility score were included in the 'low FIRE score' set (medium grey), (2) ACRs with FIRE element (FE) length <200bp and high FIRE accessibility score were included in the 'FE length <200' set (blue), (3) ACRs with mappability <80% and both high FIRE accessibility score and FE length $\geq$200bp were included in the 'Low mappability' set (dark grey) and (4) ACRs with high FIRE accessibility score, high FE length and high mappability are in the 'Regular ACRs' set (light grey).

3.9.16 FiberHMM

Footprints were called on Fiber-seq reads using FiberHMM v.1.3.1. FiberHMM is based on a hidden Markov model (HMM) with two hidden states: accessible and inaccessible. The emission probabilities used in the model are probabilities of methylation of a given base within a 6-nt sequence context in a known accessible or inaccessible state, based on experiment-derived methylation rates from a dechromatinized or a Hia5 untreated control dataset, respectively [42]. Transition and starting probabilities for the HMM were trained on a group of 1,000 reads sampled from the Maize dataset shuffled in order 20 times, with initial probabilities picked from the Dirichlet distribution with all parameters set to 1. The best model was chosen and then used for all subsequent footprint calling.

3.9.17 Identifying nearby enhancers

The precise bounds of accessible putative enhancer regions were found using a Gaussian Mixture Model Hidden Markov model (GMM-HMM) to segment regions on the basis of percent nucleosome footprint occupancy. As the positioned mononucleosome footprint found downstream of putative enhancers provided a consistent and clearly defined reference position, the upstream edge of that nucleosome was used to centre the footprint occupancy patterns for metaprofiles of the enhancers.

3.9.18 Labelling footprints in Fiber-seq

Nucleosome footprints were defined as footprints greater than 90bp. Putative transcription factor footprints were identified on the basis of a combination of their sub-40bp size and their overlap with known motifs. Transcription factor footprints overlapping multiple motifs were assigned multiple possible identities, indicated in the corresponding visualization. Putative polymerase footprints were defined on the basis of position relative to the predicted TSS of the promoter, and a size of between 40–80bp, as previously observed in ref. [85].

3.9.19 Enrichment of GWAS SNPs within different classes of ACRs

Single-nucleotide polymorphisms (SNPs) associated with 41 distinct phenotypes [86] were used to assess whether newly called FIRE ACRs have a similar enrichment of genome-wide association study (GWAS) SNPs to ATAC-called ACRs. GWAS SNPs with reads mapped in peaks <0.05 were removed as described in the paper. FIRE ACRs were split into two categories on the basis of whether they overlap ATAC-seq ACRs as in Fig. 3.2c. For both categories, an enrichment was calculated by comparing the fraction of ACR bases covered by GWAS SNPs to the fraction covered in the shifted control category. FIRE ACRs overlapping and not overlapping ATAC-seq ACRs were found to have enrichment values of 3.37 and 3.16, respectively.

3.9.20 Calling differential ACRs

ATAC-seq reads from the following six tissues were downloaded from the NCBI Gene Expression Omnibus [45]: Tassel (GSM4696882), Ear (GSM4696883), Root1 (GSM4696884), Axillary_bud1 (GSM4696886), Crown_root1 (GSM4696888), Leaf2 (GSM4696890). For each sample, 100 million read

pairs were downloaded, trimmed to 50bp. Each of the six downloaded samples as well as reads from our in-house dark leaf protoplast sample were aligned to the MaizeV5 reference genome [39] using bwa (v.0.7.17-r118) [77]. The resulting bams were filtered using samtools view [78] to discard reads that were unmapped (-F 4), had map quality of zero (-q 1), or mapped to the centromere [84]. Because the number of MACS2 peaks is correlated with the number of mapped reads, for each of the seven samples, the number of aligned reads was subsampled to 16M. Peaks were called using MACS (v.2.2.7.1) [79], and the narrowPeaks output was merged to generate a non-overlapping set of peaks for each of the seven samples. A union set of 80,641 peaks was generated by merging the seven sets of peaks (bedops -m) [85]. TN5 insertions were tallied in each union peak for each of the seven samples, and per-bp accessibility was calculated by dividing by the peak length. Because our aim was to find differential ACRs that were inaccessible in dark leaf protoplast, we defined differential ACRs as those that (1) had fewer per-bp TN5 insertions than twice the minimum dark leaf protoplast cutcounts in a union peak overlapping a called dark leaf protoplast peak and (2) the difference between the accessibility of most-accessible sample and the dark leaf protoplast sample was in the 75th percentile or greater. These 2,826 dACRs are in Supplementary Table 8.

3.9.21 Identification of solo LTRs

LTR sequences from intact LTR retrotransposons containing at least one FIRE ACR within either LTR were aligned to the maize genome using blastn [87]. Matches with bitscore greater than 1,400 and length greater than 1,000bps were retained and merged (bedops -m). Next, we identified matches that (1) did not overlap another intact LTR retrotransposon and (2) contained a FIRE ACR. These are listed in Supplementary Table 10.

3.9.22 Identification of hAT insertion sites

hAT insertion sites are defined as 200-bp windows centred on the location of a hAT transposon polymorphism in which B73 lacks the hAT transposon and exactly 1 of the 25 NAM lines contains a hAT transposon [70].

3.10 Data availability

Raw and processed sequencing data are available from the NCBI Short-Read Archive (SRA) under Bioproject PRJNA1119563.

Chapter 4

Beyond Peaks: Chromatin State and ACR Architecture from Fiber-seq Features

4.1 Introduction

In Chapters 2 and 3, I described tools for analyzing Fiber-seq data and demonstrated its application to maize. However, these analyses largely followed traditional peak-based approaches. In this chapter, I explore whether the rich, multi-layered information captured by Fiber-seq can supersede traditional single-track chromatin assays and enable new analytical approaches that exploit information beyond simple accessibility peaks.

Traditional chromatin accessibility assays such as DNase-seq, ATAC-seq, and MNase-seq generate a single genomic track representing the density of cuts or inserts at each position. This has driven how we look at accessibility. Researchers typically use peak callers like MACS2 [79] to identify accessible chromatin regions (ACRs), and then quantify the density of cuts in those ACRs to compare across conditions for example. Fiber-seq is much more information rich than traditional accessibility assays, and thus requires new approaches to analysis beyond peak calling. With Fiber-seq each sequenced molecule reports not only accessible regions, but also their size, the positioning and size of nucleosome and other footprints, and endogenous 5mCpG methylation. This information richness may enable analyses that would traditionally

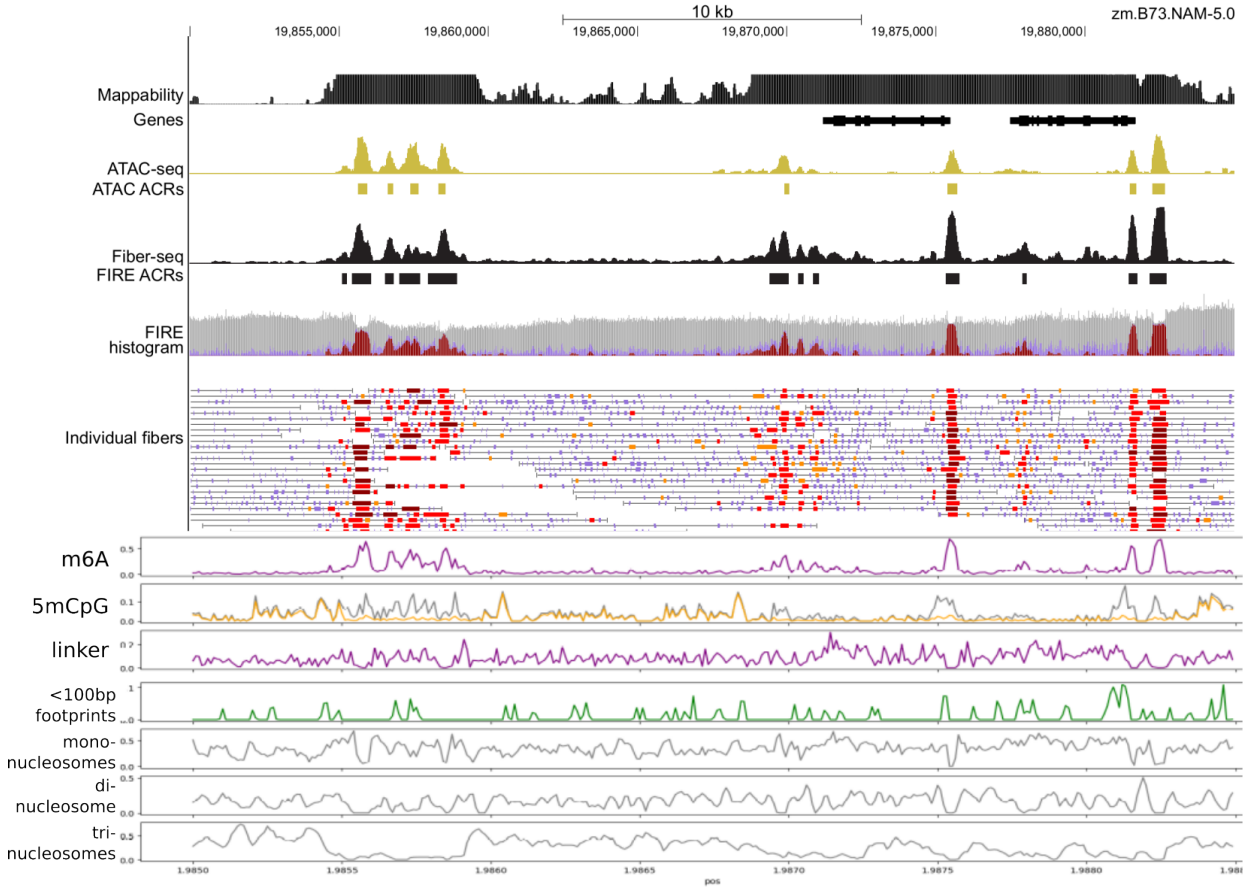

Fig. 4.1: A 35kb example region from maize. Tracks shown from top to bottom: mappability of 100bp bins, ATAC-seq and Fiber-seq signal and ACRs, Fiber-seq element histogram, individual fibers with accessible patches colored, below are several tracks derived from the single-molecule data: fraction of adenines methylated, abundance of CpG sites (grey) and 5mC methylated CpGs (orange), abundance of linkers, abundance of mono-, di-, and tri-nucleosomes.

require integrating data from multiple assays such as short-read accessibility assays along with ChIP-seq assays, and opens the door for new analysis approaches.

In this chapter, I aggregate single-molecule Fiber-seq data into a collection of feature tracks that preserve information from the single fiber data. I apply these tracks to two complementary analysis strategies. First, I use chromatin state segmentation to determine whether Fiber-seq alone captures sufficient complexity to identify biologically relevant chromatin states. Second, I use the positional distribution of features surrounding ACRs to identify patterns and group or categorize ACRs with similar features together.

Figure 4.1 shows an example region from the maize genome. Above the individual fibers and FIRE histogram we see the single accessibility tracks and ACRs from ATAC-seq and Fiber-seq. Below we see a number of different tracks derived from Fiber-seq, each contributing some unique information. This example shows multiple tracks of information can be derived from the single molecule Fiber-seq data, each containing unique information about chromatin features.

4.2 Results

4.2.1 Deriving aggregate feature tracks from Fiber-seq data

Fiber-seq fibers are marked by alternating patches of methylated and unmethylated DNA. The size distribution of patches in a given region can provide information about the underlying chromatin state. For example in a region where chromatin is more compact, we might see the methylated linker patches between nucleosomes smaller than average, and an increased occurrence of unmethylated patches spanning multiple nucleosomes.

The processing pipeline developed by the Stergachis lab [24] calls methylation sensitive patches (MSPs) down to single-base size and nucleosomes down to a minimum size of 80bp. Fiber-HMM [42] calls inaccessible footprints below 80bp, capturing sub-nucleosomal footprints such as polymerase complexes, the CCCTC-binding factor (CTCF), and possibly other transcription factor occupancy. I defined a set of 16 feature tracks measuring the fractional coverage of these patches within specific size ranges. I defined 3 additional tracks: the fractional methylation of adenines, and 5mC methylation of CpG di-nucleotides, as

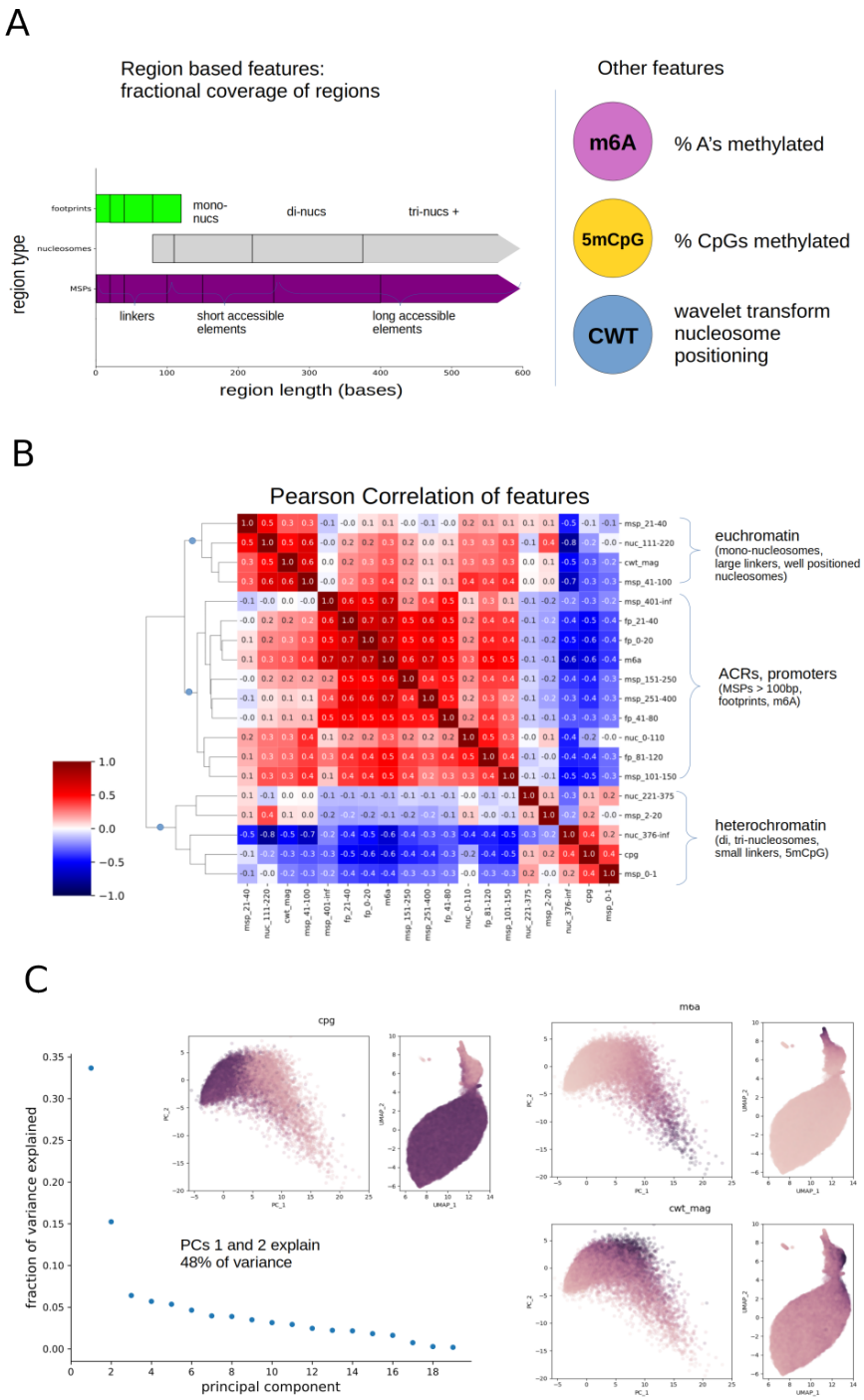

Fig. 4.2: **(A)** Schematic of 19 derived features including 16 region based features (right) and 3 additional features (left). **(B)** Pearson correlations of derived features across 10,000 600bp windows randomly selected from the maize genome. **(C)** Principal component analysis of 100,000 randomly chosen 600bp windows, PCA and UMAP plots showing distribution of three selected features.

well as a track using continuous wavelet transform to quantify nucleosome positioning consistency. These tracks are intended to preserve some single fiber information (size of individual accessible or inaccessible patches) while creating aggregate features that allow genome wide analysis using standard methods. Figure 4.2 A shows how the size ranges of the 3 patch types are subdivided to form the features.

To determine how related the 19 features are to each other, I sampled 10,000 600bp windows in the maize genome and computed the average value over each interval for each feature. In Figure 4.2 B the Pearson correlation between all features shows three main clusters of related features. The first cluster contains features associated with less tightly packed chromatinized DNA, including mono-nucleosomes, large linkers, and well positioned nucleosomes. Features in the second cluster are associated with ACRs and footprints; this includes MSPs over 100bp, all footprint features, and nucleosomes less than 110bp. The third cluster is associated with heterochromatin, including features like di- and tri-nucleosomes, 5mCpG methylation, and single-base linkers.

Principal component analysis reveals 50% of the variance can be explained by the first two principal components (Figure 4.2 C). However, the additional components contain meaningful information; this can be seen comparing a scatter plot of the first 2 principal components to a UMAP plot which utilizes all the information (not just PC1 and PC2). In the PCA scatterplot the hypo-methylated and accessible windows (visible in plots colored by 5mCpG and m6A methylation) are diffuse and un-structured. In contrast, in the UMAP plot, much more structure is visible; particularly associated with accessible windows.

4.2.2 Chromatin state segmentation captures biologically meaningful states

The ENCODE project amassed numerous ChIP-seq, DNase-seq, and other datasets to characterize chromatin state across different cell lines and biosamples [13], [88]. Each of these tracks captures a specific aspect of chromatin, however, some of this information is redundant across tracks. For example, different histone modifications often co-occur in predictable patterns. Tools including ChromHMM and Segway were developed to integrate these datasets into more condensed and human interpretable forms [14]–[16]. These tools use unsupervised learning approaches to identify recurring patterns across input tracks and assign genomic regions to a relatively small number of discrete states. Despite the reduction in complexity, these segmentations successfully identify biologically meaningful elements such as promoters, enhancers,

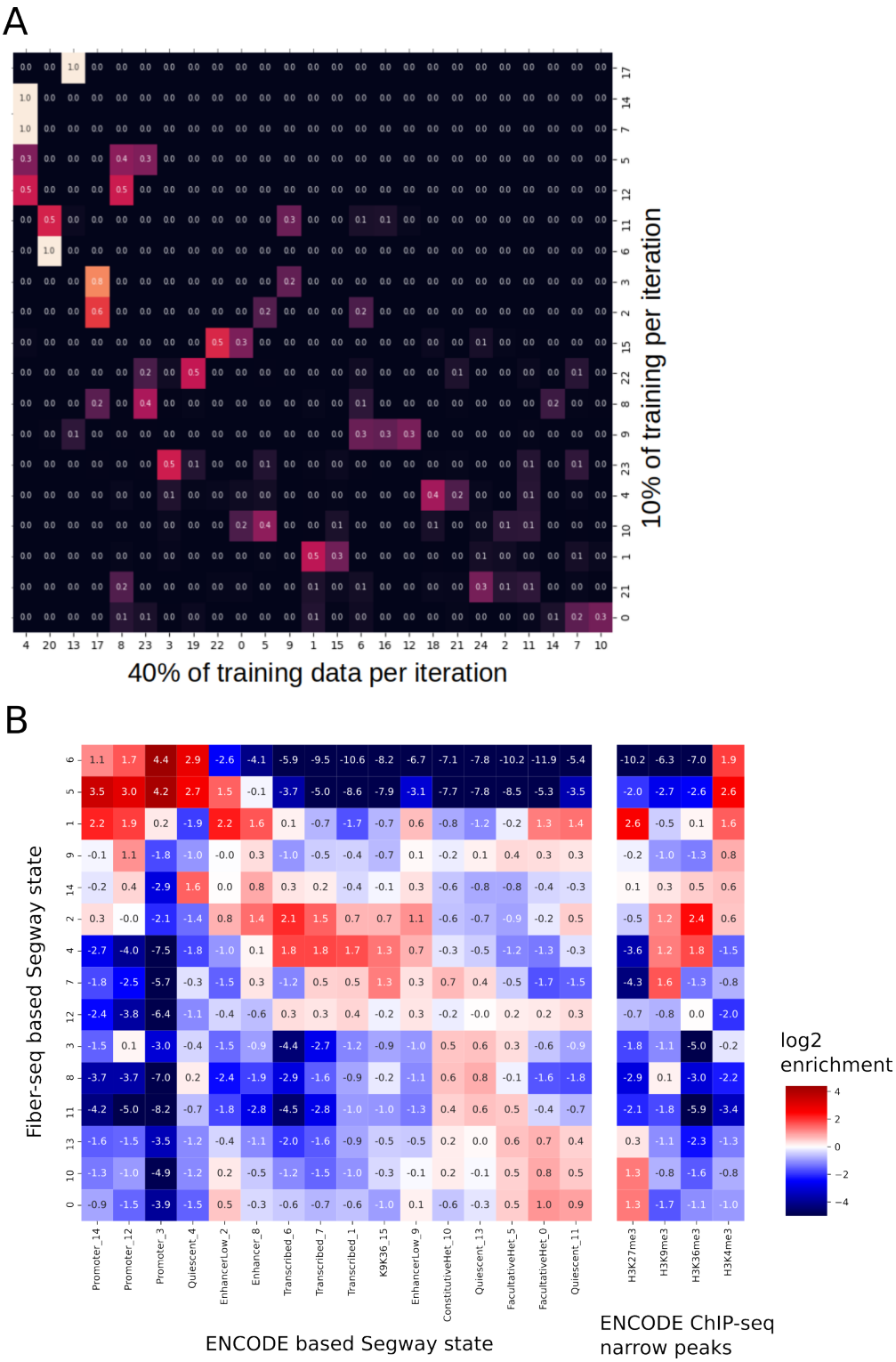

Fig. 4.3: **(A)** Row normalized heatmap of overlap of Segway states from 2 training instances. **(B)** Enrichment of Fiber-seq based states (rows) in ENCODE based Segway states and ENCODE ChIP-seq narrow peaks (columns).

repressed and transcribed regions. Typically chromatin accessibility assays like DNase-seq or ATAC-seq are used as one of the many input tracks for these segmentation algorithms.

I applied Segway to the 19 Fiber-seq feature tracks I developed. This allowed me to assess whether this single assay could produce biologically meaningful segmentations comparable to those derived from multiple ENCODE assays. I performed segmentation on both maize and the human cell line GM12878 Fiber-seq data. GM12878 has been extensively profiled by ENCODE and used in prior Segway studies [14], [15], [89], allowing me to perform direct comparisons of segmentation results.

Reproducibility of Fiber-seq segmentation

To assess the reproducibility of Fiber-seq based segmentation, I trained two independent Segway models using different random subsets of the genome (0.001 and 0.004 of the genome, respectively). Figure 4.3 A shows the overlap between states from the two replicates. Rows are normalized to sum to 100%. The majority of states from one replicate overlap with one or two states from the other replicate by more than 50%, indicating that Segway consistently identifies similar chromatin states despite using different training data. The fact that a state from one trained model maps to multiple states in the other model indicates some merging and splitting of states between trainings.

Correspondence with ENCODE chromatin states

Fiber-seq derived states showed correspondence with ENCODE chromatin states from Farahbod et al. (2023) [89] and expected histone modification patterns (Figure 4.3 B). In GM12878 states 5 and 6 were enriched for ENCODE promoter states and H3K4me3, consistent with active promoters. States 4 and 2 showed enrichment for ENCODE transcribed states and the elongation mark H3K36me3, indicating transcribed gene bodies. These states also show enrichment for the 5mCpG methylation input feature.

States 3, 8, and 11 were enriched for ENCODE based states labeled heterochromatin, including a constitutive heterochromatin state, these states also have under-representation of H3K27me3 ENCODE peaks, which is consistent with constitutive heterochromatin (H3K27me3 is associated with active polycomb mediated repression). Fiber-seq Segway states 0 and 10 show enrichment for a different (but overlapping) set of heterochromatin annotated ENCODE states. 2 of the three highest enriched ENCODE states are labeled

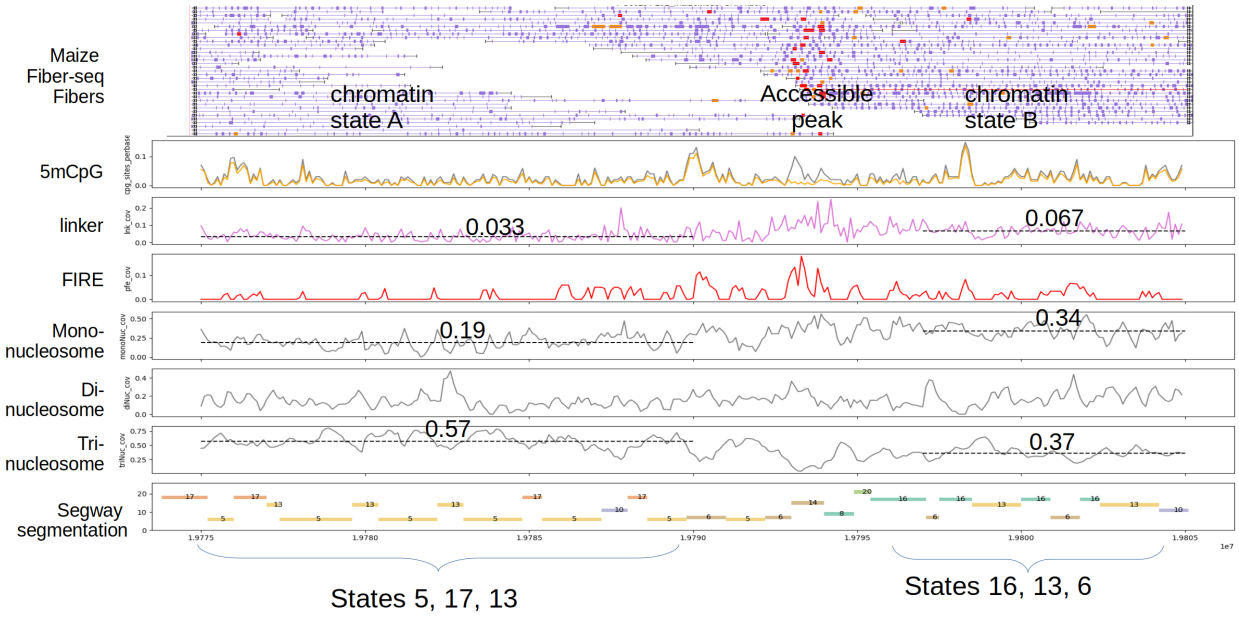

Fig. 4.4: ACR at boundary between chromatin states in maize.

facultative heterochromatin. Additionally, Fiber-seq Segway states 0 and 10 are enriched in H3K27me3 (contrasting states 3, 8 and 11). This suggests heterogeneity in repressed chromatin (facultative vs constitutive) that both models are capturing, though not with perfect correspondence. and that Fiber-seq features may be able to distinguish facultative vs constitutive heterochromatin.

Fiber-seq state 13 is also associated with ENCODE heterochromatin states but appears to be an intermediate state, not showing strong enrichment or depletion of H3K27me3 ChIP-seq peaks.

Another interesting state is Fiber-seq based state 1, with enrichment of both ENCODE promoter states and H3K27me3. This state is enriched for input features including sub-nucleosomal footprints , well positioned nucleosomes (CWT feature), and accessible elements intermediate in size between linkers and regulatory elements (MSPs 41-100 bases in length). This could be a bivalent promoter state.

Boundaries between chromatin states in maize

In maize, intergenic ACRs may mark boundaries between regions of differing chromatin compaction states. Figure 4.4 shows an example where Fiber-seq derived features and segmentation capture such a boundary. An intergenic ACR separates two regions with distinct chromatin organization: on the left side, labeled

chromatin state A, nucleosomes are more tightly packed with fewer mono-nucleosomes (0.19 vs 0.34), more tri nucleosomes (0.57 vs 0.37), smaller linkers, and lower overall m6A methylation (0.033 vs 0.067). Fiber-seq derived Segway annotation primarily marks this region with states 5, 17, and 13. On the right side (chromatin state B), with larger linkers, more mono-nucleosomes, Segway marks this region with a different set of states: 16, 13, and 6. This example demonstrates that Fiber-seq segmentation can identify transitions in chromatin compaction states. A more systematic analysis is needed to determine whether intergenic ACRs consistently mark boundaries between regions with different chromatin organization, and if ACRs that sit at the boundary of chromatin states have unique characteristics.

4.2.3 ACR annotation

Traditional chromatin accessibility assays like DNase-seq and ATAC-seq produce a single track of information representing the density of cuts or insertions across the genome. Peak calling algorithms such as MACS2 [79] identify accessible chromatin regions (ACRs) from these tracks. These ACRs are typically quantified with a single number representing the level of accessibility signal within them. However, ACRs are functionally diverse, including promoters, enhancers, insulators, and other regulatory elements. ACRs may be accessible through different mechanisms and occupied by distinct sets of DNA binding proteins. The single track of information provided by short read based accessibility assays is insufficient to capture this complexity, so additional information from different assays or genome annotations are typically required to further annotate ACRs. For example, segmentation algorithms like Segway attempt to distinguish promoters vs enhancers based on their histone marks, gene annotations are also commonly used to label promoters based on proximity to transcription start sites.

To determine if the additional information provided by Fiber-seq enables improved annotation and categorization of ACRs, I used the Fiber-seq derived features to create feature arrays for each ACR in the maize genome. I hypothesize that the positional distribution of these features surrounding an ACR may capture important information about its function. The clearest example of this is promoters, which have well positioned nucleosomes as well as polymerase complex footprints on the downstream side. Feature arrays for each ACR were constructed using the 19 feature tracks at 44 positional, 50bp wide bins covering the ACRs and their surroundings, for a total of 836 positional feature values for each ACR (see Figure 4.5 A).

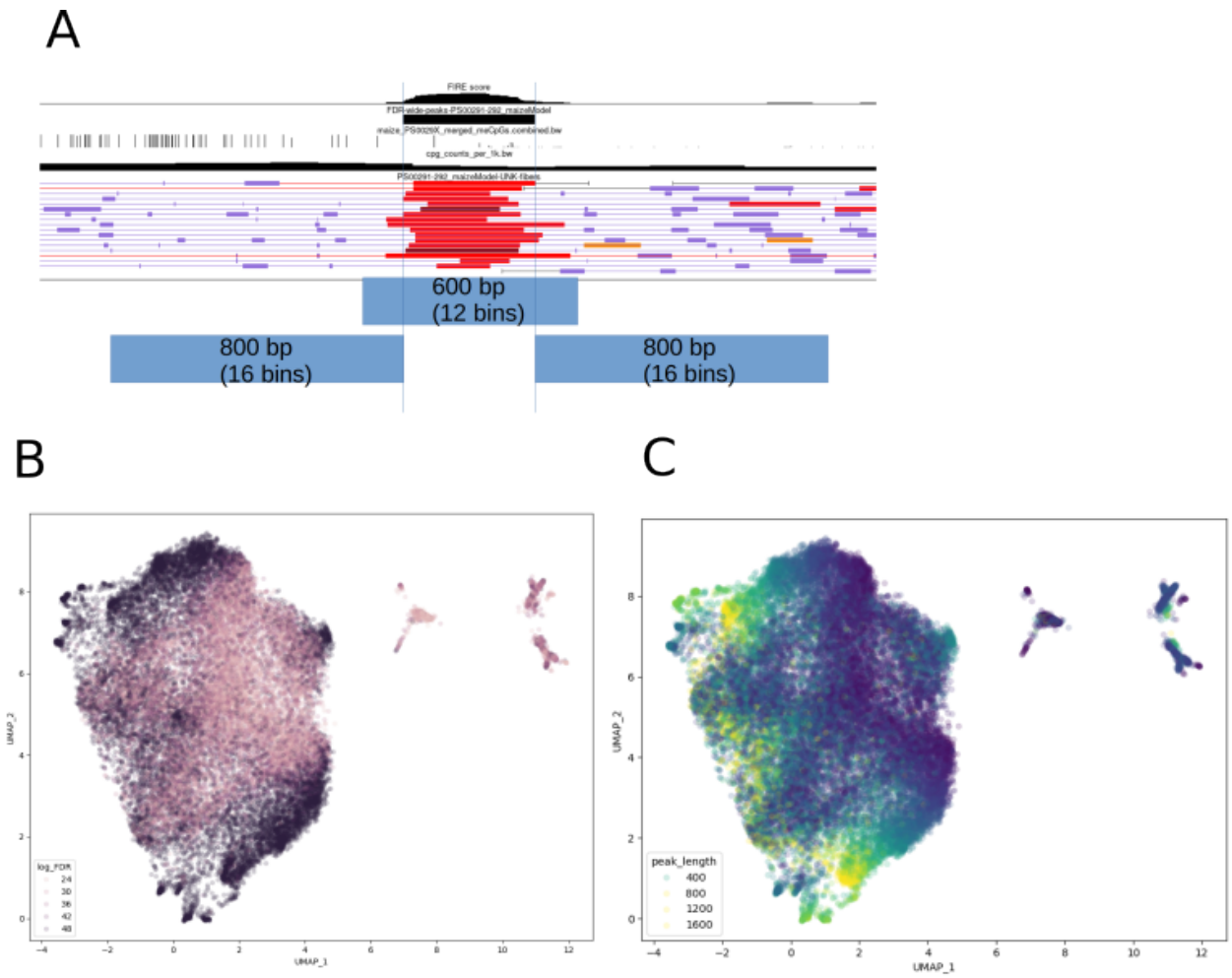

Fig. 4.5: **(A)** Schematic of regions surrounding each ACR to define positional bins for ACR categorization. **(B)** UMAP of maize ACRs colored by accessibility score. **(C)** UMAP of maize ACRs colored by ACR length.

UMAP visualization of ACR features

To visualize the relationships between ACRs based on their features, I applied UMAP dimensionality reduction. Figure 4.5 B-C shows the UMAP colored by ACR accessibility score and length. The structure of the UMAP is complex with multiple areas showing gradients from high to low accessibility and short to long ACR length. The overall visual symmetry of the UMAP is due to the fact that ACRs with asymmetric features (such as promoters with distinct upstream and downstream patterns) are randomly oriented in the genome, and thus for any given pattern of positional features around an ACR, there are likely to be other ACRs with roughly mirror images of those features (Watson vs Crick stranded genes). The small clusters to the right side of the UMAP are ACRs with CpG methylation.

Promoters occupy distinct regions of the UMAP

Annotated promoters are highly enriched in specific regions of the UMAP (Figure 4.6 A). Promoters form a continuum across the UMAP rather than discrete clusters. To understand what distinguishes promoters in different regions, I examined the feature patterns of forward promoters at several positions along this continuum (Figure 4.6 B). Forward promoters along the lower right outside edge of the UMAP show a canonical regulatory architecture with a single, well-defined ACR at the transcription start site and well-positioned nucleosomes on the downstream side. Promoter ACRs are wide at the bottom of this region and get progressively narrower traveling up from there until this continuum curves toward the center of the UMAP. Promoters in this region have less distinct boundaries with additional accessible patches appearing first upstream, then on both sides as the path crosses the middle. The few bi-directional promoters are found most commonly along the axis of symmetry. As the continuum crosses to the other side of the UMAP, promoters again show well-defined ACRs, but now with large accessible regions immediately upstream.

Gene expression levels vary across these promoter configurations (Figure 4.6 C). Promoters with canonical architecture are enriched for highly expressed genes. Following the curve toward the center, where promoter boundaries become less distinct, genes are found generally in lower expression deciles. Across the center line, where large upstream accessible regions appear, expression levels increase again.

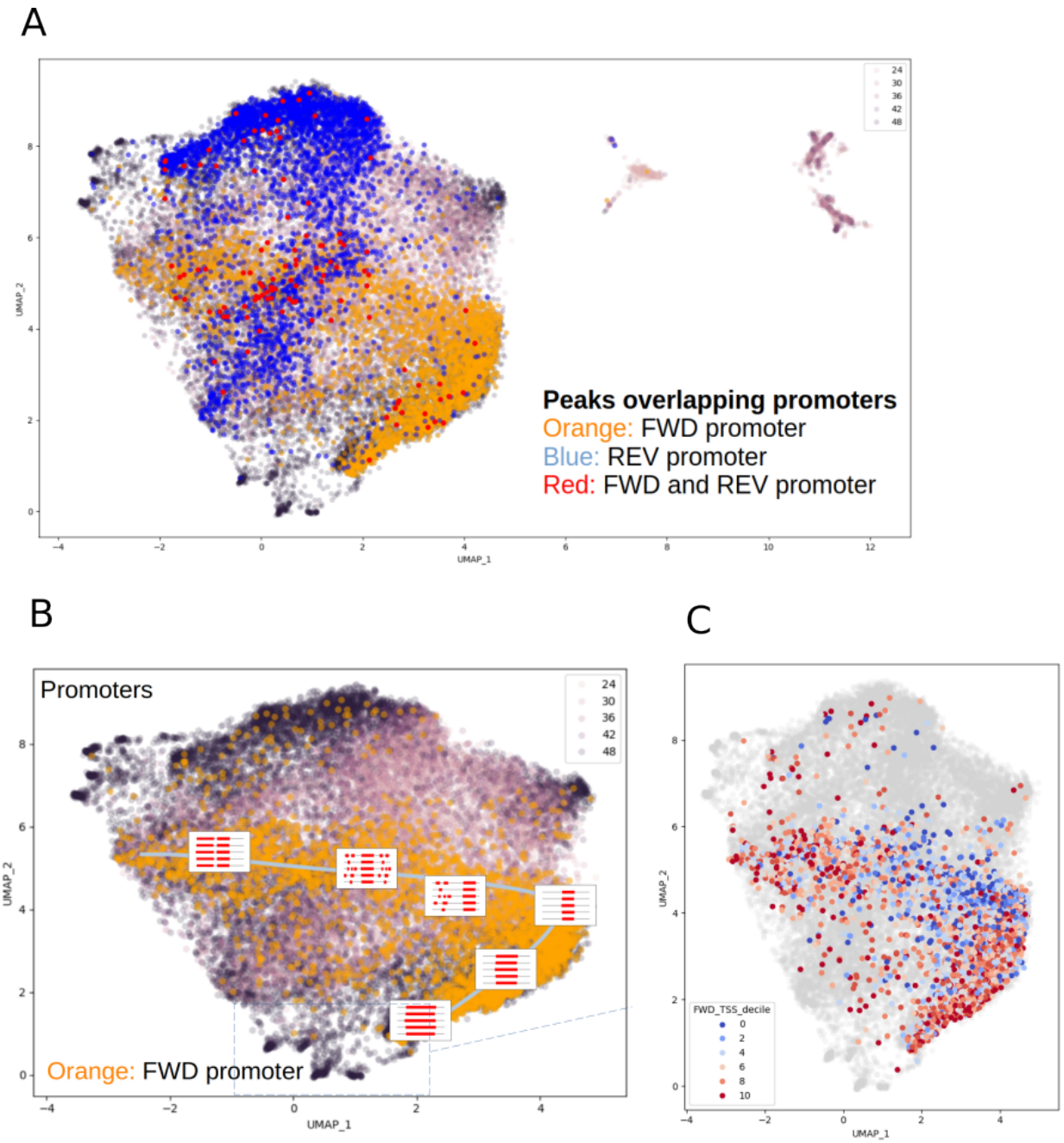

Fig. 4.6: **(A)** UMAP of maize ACRs with ACRs in promoters colored: forward promoters colored orange, reverse promoters colored blue, ACRs in forward and reverse promoters (bi-directional) colored red. **(B)** ACR UMAP with forward promoters colored in orange and cartoon representation of typical accessible element patterns. **(C)** ACR UMAP with forward promoters colored by expression decile of associated gene.

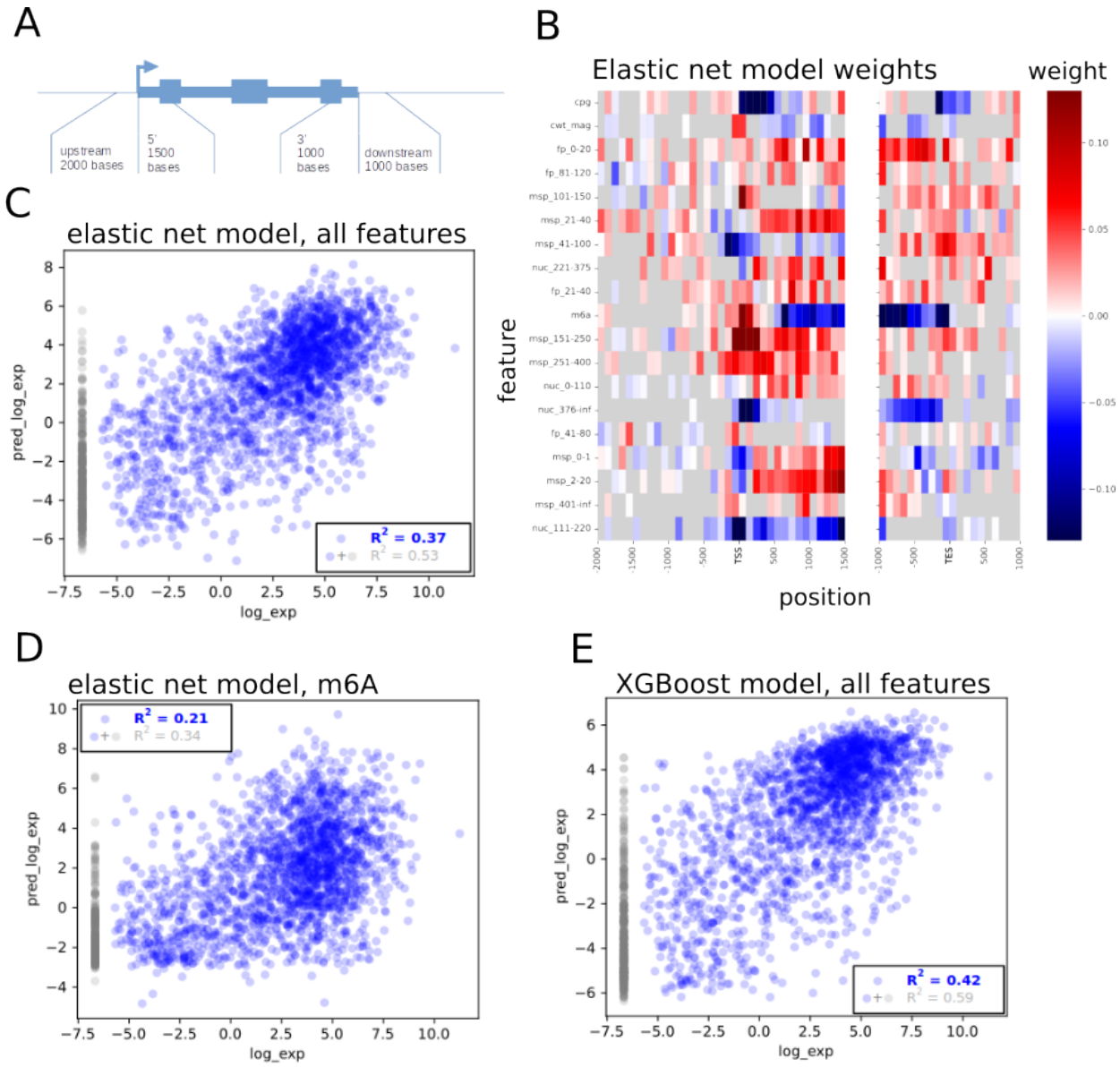

Fig. 4.7: **(A)** Diagram of regions surrounding genes used for expression prediction. **(B)** Weights of elastic net linear model. **(C)** Predicted vs. measured log expression on genes from chromosome 8, (held out from training) blue dots indicate non-zero measured expression, grey dots indicate zero measured expression (about 15% of genes), R-squared values shown with and without zero expression genes. **(D)** Same as (C) using elastic net model only including m6A feature. **(E)** Same as (C) using XGBoost model.

Fiber-seq features predict gene expression

The enrichment of promoters with different expression levels in distinct regions of the UMAP suggested that Fiber-seq derived features could predict gene expression. To test this, I constructed feature arrays for annotated genes, with positional bins extending upstream and downstream of both the transcription start site (TSS) and the 3' end (Figure 4.7 A). The 3' end was included because maize genes often show patterns of accessibility in this region (see the two genes in Figure 4.1 for example).

An elastic net model achieved a Pearson's R^2 of 0.37 on chromosome 8 genes held out from training when excluding genes with zero measured expression, and 0.53 when including them (Figure 4.7 C). The model weights were distributed across many of the features (Figure 4.7 B). CpG methylation showed strong negative weights at the TSS, while m6A methylation had positive weights near the TSS and negative weights in the gene body.

To assess whether these additional features provide information beyond standard accessibility measurements, I trained a model using only the m6A feature as a proxy for short-read accessibility assays. This single-feature model returned an R^2 of 0.21 excluding zero expression genes and 0.34 including them (Figure 4.7 D), substantially lower than the full feature model. This demonstrates that the diverse chromatin features captured by Fiber-seq contribute meaningful information beyond overall accessibility levels.

A non-linear XGBoost model using all features achieved R^2 values of 0.42 excluding zero expression genes and 0.59 including them (Figure 4.7 E). These predictions are likely limited by biological factors that expression measurements capture but chromatin states do not, such as post-transcriptional regulation. Additionally, noise at low expression levels, visible as increased scatter in the lower range of the predicted versus measured plots, contributes to unexplained variance.

Methylated LTR retrotransposon ACRs

To better understand the methylated ACRs on the right side of the UMAP in Figure 4.5, I examined patterns across the clusters (Figure 4.8 A, labeled 1-5). Across these clusters except for cluster 3, we often see a pattern with a dearth of CpG dinucleotide sites, and tightly compacted chromatin on one side of the ACR with reduced m6A methylation, short linkers, and tri-nucleosome footprints. These appear on the right side

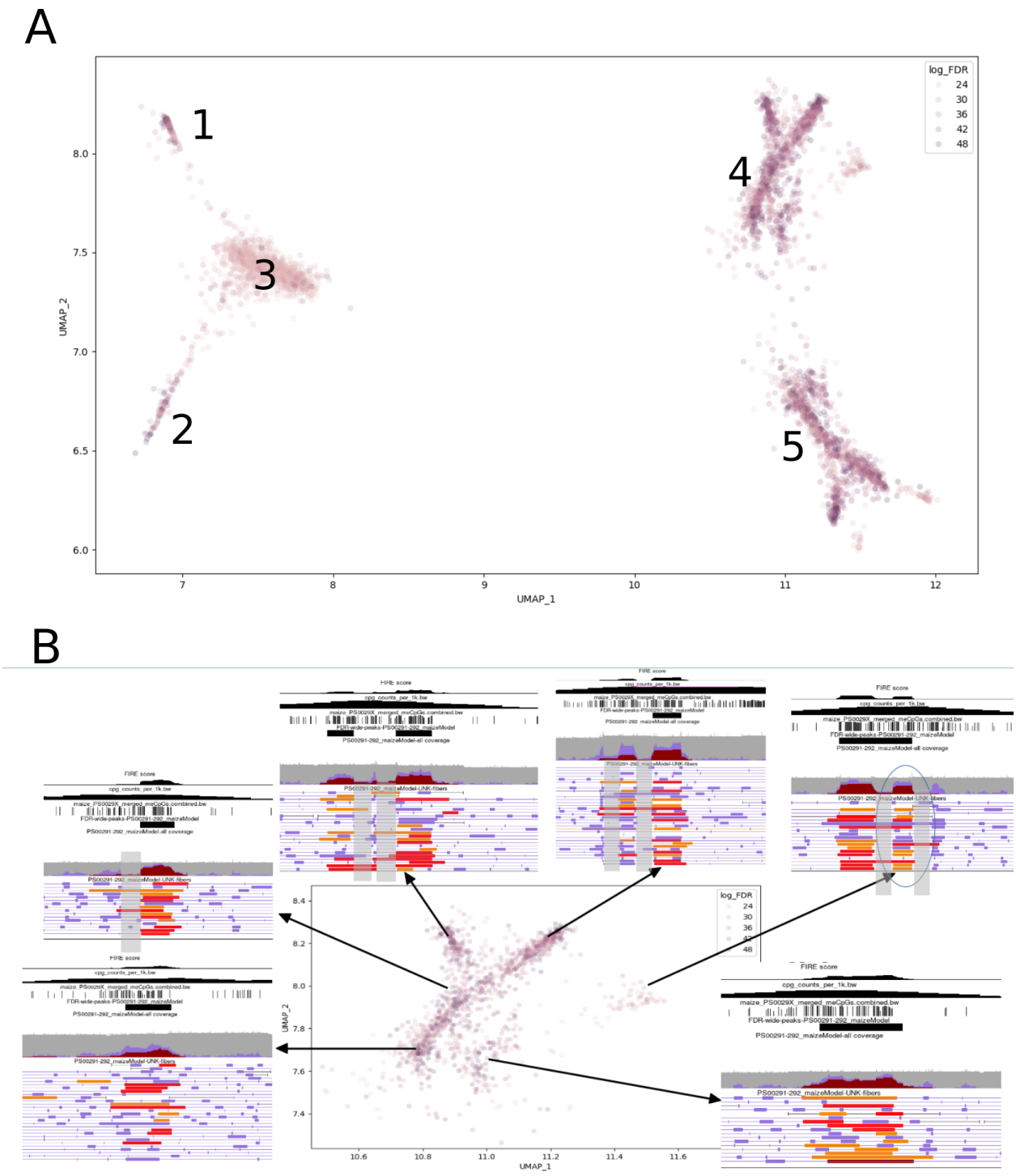

Fig. 4.8: **(A)** UMAP clusters of ACRs with 5mCpG methylation, labeled 1-5. **(B)** Cluster 4 from (A) showing representative browser views with distinct patterns of nucleosome and accessible element placement.

in clusters 1 and 4, and the left side in clusters 2 and 5. Clusters 1 and 2 form mirror images, with ACRs often narrow with well-positioned nucleosomes extending in one direction. Cluster 3 lies along the axis of symmetry and contains ACRs with low accessibility and no clear organizational pattern. This cluster is enriched for low-coverage ACRs, though not exclusively.

Clusters 4 and 5 are also mirror images of each other. Each forms a Y-shaped structure with a couple diffuse sub-clusters adjacent to it (Figure 4.8 B). At the base of the Y in cluster 4, ACRs resemble those in cluster 1, though the ACRs are somewhat wider and lack clear structure on the left. Moving toward the branch point, the ACR becomes more defined and a single well-positioned nucleosome begins to appear on the left side. Along the left branch of the Y, this pattern elaborates into a well-positioned nucleosome, a short consistent linker, a second well-positioned nucleosome, and in some fibers, a short accessible element emerging to the left of this second nucleosome. The right branch shows a related but distinct arrangement: a well-positioned nucleosome, followed by a short accessible element, another well-positioned nucleosome, and then a short linker. This pattern of a shorter accessible element flanked by positioned nucleosomes adjacent to a longer ACR is reminiscent of the paired ACRs in 2-2 LTRRTs described in Chapter 3, where the shorter element may function as an enhancer and the longer as a promoter.

The UMAP visualization of these methylated ACRs in LTRRTs reveal a distinct and structurally diverse class of regulatory elements, not simply methylated versions of the unmethylated paired ACRs. In our previous work (Chapter 3), counting ACRs in LTRRTs may have conflated some methylated yet accessible elements with the unmethylated 2-2 pattern, suggesting that paired ACRs can become methylated over time while remaining accessible. The current analysis suggests that these methylated elements likely belong to separate LTRRT lineages with different chromatin architectures. The Y-shaped organization of cluster 4 suggests a continuum of related configurations, possibly corresponding to different LTRRT clades. This hypothesis could be tested through phylogenetic analysis of sequences from different positions within the cluster. Grouping LTRs by patterns in these Fiber-seq derived feature tracks, rather than presence/absence or counting of ACRs should provide a more accurate grouping relative to the underlying biological phenomena, aiding future analysis.

4.3 Discussion

Fiber-seq derived features capture rich information from single molecule data

Here I demonstrate that multiple aggregate feature tracks can be constructed that preserve the informational richness of this single molecule chromatin assay. Gene expression prediction demonstrates this, where models using the full set of features substantially outperformed models using only the m6A track as a proxy for accessibility. This multi-track representation of Fiber-seq data also enables analyses that were simply not possible with traditional chromatin accessibility assays alone. Segway segmentation using these features identified biologically meaningful chromatin states. Notably, the segmentation distinguished multiple heterochromatin states that may correspond to facultative and constitutive heterochromatin, as supported by their enrichment and depletion for H3K27me3 ChIP-seq peaks respectively. The systematic analysis of ACRs based on positional patterns of features would also not be practical if applied to a single accessibility track. This approach enabled a birds-eye view of the possible chromatin configurations surrounding promoters and ACRs in LTRRTs.

Alternative methods for defining feature tracks

The feature tracks used here were largely based on size bins of accessible and inaccessible regions. The boundaries between bins were somewhat arbitrary, with biological backing like the size of nucleosomes and other known protein footprints. This is by no means the only way, nor likely the best way to construct feature tracks. Alternative approaches could produce features with better predictive power or capture aspects of chromatin organization not represented in these tracks. For example, features that quantify some aspect of heterogeneity across fibers. Sequence-based features derived from genome foundation models or k-mer embeddings could also be incorporated to capture the contribution of underlying DNA sequence. An alternative approach to generating feature tracks could use machine learning to generate embeddings to use as features. A variational autoencoder could be trained to reconstruct methylation patterns in Fiber-seq data. The latent space of this autoencoder would consist of a dense set of features capable of producing an approximation of the input methylation data.

However, one benefit of the features used here is that they are interpretable. This was useful for example

paired with the elastic net gene expression model when assessing how features contributed to the prediction. I was also able to look at how these features contributed to Segway states, helping understand what features differentiate the putative constitutive vs facultative heterochromatin.

Chromatin boundaries in plants

The example in Figure 4.4 suggests that intergenic ACRs may mark boundaries between chromatin compaction states. Determining whether this pattern holds genome-wide is particularly interesting because maize and other plants lack CTCF, the primary insulator protein organizing chromatin domains in mammals. If intergenic ACRs frequently mark chromatin state boundaries in plants, they may represent a plant-specific boundary element. Systematically identifying these sites with Fiber-seq and characterizing their sequence features and bound proteins could reveal how plants organize chromatin domains in the absence of CTCF.

4.4 Methods

4.4.1 Feature Tracks

Feature tracks used have been defined at the resolution of 100bp bins. feature tracks are saved as bedgraph and bigwig format. A snakemake pipeline was created to automatically generate the feature tracks based on configuration files that define what tracks to create.

m6a, cpg

These tracks represent the approximate fraction of available sites (adenines or CpG dinucleotides) that are methylated. The numerator is the number of methylated sites, while the denominator is the number of sites in the reference genome multiplied by the coverage.

msp_X-Y, nuc_X-Y

These tracks represent the fractional coverage of methylation sensitive patches (MSPs) or nucleosomes with sizes ranging between X and Y bases.

fp_X-Y

These tracks represent the fractional coverage of Fiber-HMM footprints with sizes ranging between X and Y bases.

cwt_mag

This track represents the ridge magnitude of the continuous wavelet transform (CWT) applied to a pileup of MSPs less than 100 bases in length (linkers).

cwt_freq

This track represents the period (in bases) of the ridge in the CWT scalogram where the magnitude exceeds a defined threshold.

4.4.2 Continuous Wavelet Transform (CWT)

The fractional coverage of MSPs with length less than 100bp (linkers) at single-base resolution was used as the input signal for the CWT. The complex Morlet wavelet was used for the analysis, implemented with the PyWavelets package. Different wavelet parameters were used for computing cwt_mag versus cwt_freq: for cwt_mag, a wavelet with better spatial resolution (fewer oscillations) was used, while for cwt_freq, a wavelet with better frequency resolution was employed.

4.4.3 Segway

Segway version 3.0.4 was used with 15 states and 30 rounds of training. For maize, the model was trained on 0.5% of the genome selected randomly. For GM12878, different fractions of training data (0.001 and 0.004) were used to assess reproducibility, but no substantial differences in results were observed between training instances.

4.4.4 ACR UMAP

For each ACR, three windows were created: 800 bases immediately upstream and downstream of the ACR boundaries (flanks), and one 600 base window centered on the ACR. Each window was divided into 50bp

bins for a total of 44 positional bins. The value of each of the 19 Fiber-seq derived feature tracks was collected at each bin for a total of 836 positional feature values per ACR. All 106,867 FIRE ACRs identified in Chapter 3 were included in the analysis. UMAP dimensionality reduction was performed using the Python umap-learn package with default parameters.

4.4.5 Gene expression prediction

Genes with CAGE-corrected transcription start sites (TSSs) were used for this analysis. Expression data from dark-grown leaf tissue were obtained from Stelpflug et al. (2016) [56]. Expression values were log-transformed as log(CPM + 0.01) for model training and evaluation. Feature arrays for genes were constructed using 100bp positional bins extending 2000bp upstream and 1500bp downstream of the TSS, and 1000bp upstream and 1000bp downstream of the 3' end. The value of each of the 19 Fiber-seq derived feature tracks was collected at each bin. Elastic net linear models were implemented using scikit-learn and XGBoost models using the xgboost package. Models were trained using genes from all chromosomes except 8 and 9, which were held out for testing. Model performance was evaluated on chromosome 8 genes using Pearson's R^2. R^2 values were calculated both including and excluding genes with zero observed transcripts, which represent approximately 15% of genes in the analysis.

Chapter 5

Conclusion

This dissertation demonstrates the power of Fiber-seq for revealing chromatin regulatory landscapes that extend beyond simple accessibility peaks. Across four chapters, I have presented tools, applications, and analytical approaches that exploit the multi-layered information captured by this single-molecule assay.

In Chapter 2, I developed fiber-views, a software package that organizes Fiber-seq data into annotated matrices compatible with standard analytical workflows. By adapting the anndata framework from single-cell genomics, fiber-views enables matrix-based analyses—including clustering, dimensionality reduction, and machine learning—while preserving the associations between chromatin features and fiber metadata. The package has already been applied in multiple published studies and is publicly available for other researchers to use or modify.

In Chapter 3, I applied Fiber-seq to maize, demonstrating its advantages over short-read accessibility assays in repetitive genomes. Fiber-seq identified twice as many accessible chromatin regions as paired ATAC-seq experiments, including short FIRE elements that mark loci with tissue-specific accessibility and ACRs within LTR retrotransposons that are missed by short-read assays.

In Chapter 4, I moved beyond peak-based analysis to develop approaches that leverage the full complexity of Fiber-seq data. The 19 aggregate feature tracks I created preserve information about accessible and inaccessible region sizes, nucleosome positioning, and endogenous 5mCpG methylation patterns. Chromatin

state segmentation using these features identified biologically meaningful states comparable to those derived from multiple ENCODE assays, including apparent distinctions between facultative and constitutive heterochromatin. Systematic analysis of ACRs based on positional feature distributions revealed a continuum of promoter architectures associated with different expression levels and identified structurally distinct classes of 5mCpG methylated ACRs associated with LTR retrotransposons. Gene expression prediction demonstrated that these chromatin features contribute information beyond simple accessibility measurements.

The tools and approaches developed here have applications beyond the specific biological questions addressed in this dissertation. The fiber-views package provides a general framework for organizing and analyzing single-molecule chromatin data that can be applied to any organism or tissue. The feature-based analysis approach, deriving multiple tracks that preserve single-molecule information shows promise for exploring different states of heterochromatin and functionally characterizing ACRs, but more work is needed to fully develop this approach.

Bibliography

[1] E. Rodgers-Melnick, D. L. Vera, H. W. Bass, and E. S. Buckler, "Open chromatin reveals the functional maize genome," *Proceedings of the National Academy of Sciences*, vol. 113, no. 22, E3177–E3184, May 31, 2016. DOI: 10.1073/pnas.1525244113. [Online]. Available: https://www.pnas.org/doi/10.1073/pnas.1525244113 (visited on 07/31/2025).

[2] M. T. Maurano, R. Humbert, E. Rynes, *et al.*, "Systematic Localization of Common Disease-Associated Variation in Regulatory DNA," *Science*, vol. 337, no. 6099, pp. 1190–1195, Sep. 7, 2012. DOI: 10.1126/science.1222794. [Online]. Available: https://www.science.org/doi/full/10.1126/science.1222794 (visited on 08/01/2025).

[3] H. Zhao, W. Zhang, T. Zhang, *et al.*, "Genome-wide MNase hypersensitivity assay unveils distinct classes of open chromatin associated with H3K27me3 and DNA methylation in Arabidopsis thaliana," *Genome Biology*, vol. 21, no. 1, p. 24, Feb. 3, 2020, ISSN: 1474-760X. DOI: 10.1186/s13059-020-1927-5. [Online]. Available: https://doi.org/10.1186/s13059-020-1927-5 (visited on 07/31/2025).

[4] K. L. Bubb, M. O. Hamm, T. W. Tullius, *et al.*, "The regulatory potential of transposable elements in maize," *Nature Plants*, vol. 11, no. 6, pp. 1181–1192, Jun. 2025, ISSN: 2055-0278. DOI: 10.1038/s41477-025-02002-z. [Online]. Available: https://www.nature.com/articles/s41477-025-02002-z (visited on 07/31/2025).

[5] S. Kumar and T. Mohapatra, "Dynamics of DNA Methylation and Its Functions in Plant Growth and Development," *Frontiers in Plant Science*, vol. 12, May 21, 2021, ISSN: 1664-462X. DOI: 10.3389/

fpls.2021.596236. [Online]. Available: https://www.frontiersin.org/journals/plant-science/articles/10.3389/fpls.2021.596236/full (visited on 12/22/2025).

[6] M. Foroozani, D. H. Holder, and R. B. Deal, "Histone Variants in the Specialization of Plant Chromatin," *Annual Review of Plant Biology*, vol. 73, pp. 149–172, Volume 73, 2022 May 20, 2022, ISSN: 1543-5008, 1545-2123. DOI: 10.1146/annurev-arplant-070221-050044. [Online]. Available: https://www.annualreviews.org/content/journals/10.1146/annurev-arplant-070221-050044 (visited on 12/19/2025).

[7] L. Cole, S. Kurscheid, M. Nekrasov, *et al.*, "Multiple roles of H2A.Z in regulating promoter chromatin architecture in human cells," *Nature Communications*, vol. 12, no. 1, p. 2524, May 5, 2021, ISSN: 2041-1723. DOI: 10.1038/s41467-021-22688-x. [Online]. Available: https://www.nature.com/articles/s41467-021-22688-x (visited on 12/19/2025).

[8] "Histone Database 2.0." (), [Online]. Available: https://www.ncbi.nlm.nih.gov/research/histonedb/human/ (visited on 07/31/2025).

[9] P. B. Talbert, K. Ahmad, G. Almouzni, *et al.*, "A unified phylogeny-based nomenclature for histone variants," *Epigenetics & Chromatin*, vol. 5, p. 7, Jun. 21, 2012, ISSN: 1756-8935. DOI: 10.1186/1756-8935-5-7. PMID: 22650316.

[10] S. Nitsch, L. Zorro Shahidian, and R. Schneider, "Histone acylations and chromatin dynamics: Concepts, challenges, and links to metabolism," *EMBO Reports*, vol. 22, no. 7, e52774, Jul. 5, 2021, ISSN: 1469-221X. DOI: 10.15252/embr.202152774. PMID: 34159701. [Online]. Available: https://pmc.ncbi.nlm.nih.gov/articles/PMC8406397/ (visited on 12/22/2025).

[11] Y.-Z. Zhang, J. Yuan, L. Zhang, *et al.*, "Coupling of H3K27me3 recognition with transcriptional repression through the BAH-PHD-CPL2 complex in Arabidopsis," *Nature Communications*, vol. 11, no. 1, p. 6212, Dec. 4, 2020, ISSN: 2041-1723. DOI: 10.1038/s41467-020-20089-0. [Online]. Available: https://www.nature.com/articles/s41467-020-20089-0 (visited on 12/22/2025).

[12] J. Wang, S. T. Jia, and S. Jia, "New Insights into the Regulation of Heterochromatin," *Trends in Genetics*, vol. 32, no. 5, pp. 284–294, May 1, 2016, ISSN: 0168-9525. DOI: 10.1016/j.tig.2016.02.005. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0168952516000299 (visited on 12/22/2025).

[13] ENCODE Project Consortium, "An integrated encyclopedia of DNA elements in the human genome," *Nature*, vol. 489, no. 7414, pp. 57–74, Sep. 6, 2012, ISSN: 1476-4687. DOI: 10.1038/nature11247. PMID: 22955616.

[14] M. M. Hoffman, O. J. Buske, J. Wang, Z. Weng, J. A. Bilmes, and W. S. Noble, "Unsupervised pattern discovery in human chromatin structure through genomic segmentation," *Nature Methods*, vol. 9, no. 5, pp. 473–476, May 2012, ISSN: 1548-7105. DOI: 10.1038/nmeth.1937. [Online]. Available: https://www.nature.com/articles/nmeth.1937 (visited on 07/31/2025).

[15] R. C. W. Chan, M. W. Libbrecht, E. G. Roberts, J. A. Bilmes, W. S. Noble, and M. M. Hoffman, "Segway 2.0: Gaussian mixture models and minibatch training," *Bioinformatics*, vol. 34, no. 4, pp. 669–671, Feb. 15, 2018, ISSN: 1367-4803. DOI: 10.1093/bioinformatics/btx603. [Online]. Available: https://doi.org/10.1093/bioinformatics/btx603 (visited on 07/31/2025).

[16] J. Ernst and M. Kellis, "Chromatin-state discovery and genome annotation with ChromHMM," *Nature Protocols*, vol. 12, no. 12, pp. 2478–2492, Dec. 2017, ISSN: 1750-2799. DOI: 10.1038/nprot.2017.124. [Online]. Available: https://www.nature.com/articles/nprot.2017.124 (visited on 12/22/2025).

[17] A. Studer, Q. Zhao, J. Ross-Ibarra, and J. Doebley, "Identification of a functional transposon insertion in the maize domestication gene tb1," *Nat. Genet.*, vol. 43, 2011. DOI: 10.1038/ng.942. [Online]. Available: https://doi.org/10.1038/ng.942.

[18] H. Wang, A. J. Studer, Q. Zhao, R. Meeley, and J. F. Doebley, "Evidence That the Origin of Naked Kernels During Maize Domestication Was Caused by a Single Amino Acid Substitution in tga1," *Genetics*, vol. 200, no. 3, pp. 965–974, Jul. 2015, ISSN: 1943-2631. DOI: 10.1534/genetics.115.175752. PMID: 25943393.

[19] F. Cheng, J. Wu, and X. Wang, "Genome triplication drove the diversification of Brassica plants," *Horticulture Research*, vol. 1, p. 14 024, 2014, ISSN: 2662-6810. DOI: 10.1038/hortres.2014.24. PMID: 26504539.

[20] J. F. Crow, "90 Years Ago: The Beginning of Hybrid Maize," *Genetics*, vol. 148, no. 3, pp. 923–928, Mar. 1, 1998, ISSN: 1943-2631. DOI: 10.1093/genetics/148.3.923. [Online]. Available: https://doi.org/10.1093/genetics/148.3.923 (visited on 12/21/2025).

[21] W. Jiang, H. Zhou, H. Bi, M. Fromm, B. Yang, and D. P. Weeks, "Demonstration of CRISPR/Cas9/sgRNA-mediated targeted gene modification in Arabidopsis, tobacco, sorghum and rice," *Nucleic Acids Research*, vol. 41, no. 20, e188, Nov. 1, 2013, ISSN: 0305-1048. DOI: 10.1093/nar/gkt780. [Online]. Available: https://doi.org/10.1093/nar/gkt780 (visited on 12/22/2025).

[22] L. Wang, S. O'Conner, R. Tanvir, *et al.*, "CRISPR/Cas9-based editing of NF-YC4 promoters yields high-protein rice and soybean," *New Phytologist*, vol. 245, no. 5, pp. 2103–2116, 2025, ISSN: 1469-8137. DOI: 10.1111/nph.20141. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1111/nph.20141 (visited on 12/22/2025).

[23] M. W. Dorrity, C. M. Alexandre, M. O. Hamm, *et al.*, "The regulatory landscape of Arabidopsis thaliana roots at single-cell resolution," *Nature Communications*, vol. 12, no. 1, p. 3334, Jun. 7, 2021, ISSN: 2041-1723. DOI: 10.1038/s41467-021-23675-y. PMID: 34099698.

[24] A. Jha, S. C. Bohaczuk, Y. Mao, *et al.*, "DNA-m6A calling and integrated long-read epigenetic and genetic analysis with fibertools," *Genome Research*, vol. 34, no. 11, pp. 1976–1986, Jan. 11, 2024, ISSN: 1088-9051, 1549-5469. DOI: 10.1101/gr.279095.124. PMID: 38849157. [Online]. Available: http://genome.cshlp.org/content/34/11/1976 (visited on 12/17/2025).

[25] T. Jores, M. Hamm, J. T. Cuperus, and C. Queitsch, "Frontiers and techniques in plant gene regulation," *Current Opinion in Plant Biology*, vol. 75, p. 102 403, Oct. 1, 2023, ISSN: 1369-5266. DOI: 10.1016/j.pbi.2023.102403. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1369526623000687 (visited on 07/29/2025).

[26] S. C. Bohaczuk, Z. J. Amador, C. Li, *et al.*, "Resolving the chromatin impact of mosaic variants with targeted Fiber-seq," *Genome Research*, vol. 34, no. 12, pp. 2269–2278, Jan. 12, 2024, ISSN: 1088-9051, 1549-5469. DOI: 10.1101/gr.279747.124. PMID: 39653420. [Online]. Available: http://genome.cshlp.org/content/34/12/2269 (visited on 12/21/2025).

[27] C. Trapnell, D. Cacchiarelli, J. Grimsby, *et al.*, "The dynamics and regulators of cell fate decisions are revealed by pseudotemporal ordering of single cells," *Nature Biotechnology*, vol. 32, no. 4, pp. 381–386, Apr. 2014, ISSN: 1546-1696. DOI: 10.1038/nbt.2859. [Online]. Available: https://www.nature.com/articles/nbt.2859 (visited on 10/16/2025).

[28] B. McClintock, "The origin and behavior of mutable loci in maize," *Proc. Natl Acad. Sci. USA*, vol. 36, 1950. DOI: 10.1073/pnas.36.6.344. [Online]. Available: https://doi.org/10.1073/pnas.36.6.344.

[29] B. McClintock, "Induction of instability at selected loci in maize," *Genetics*, vol. 38, 1953. DOI: 10.1093/genetics/38.6.579. [Online]. Available: https://doi.org/10.1093/genetics/38.6.579.

[30] B. McClintock, "The significance of responses of the genome to challenge," *Science*, vol. 226, 1984. DOI: 10.1126/science.15739260. [Online]. Available: https://doi.org/10.1126/science.15739260.

[31] N. V. Fedoroff, "Transposable genetic elements in maize," *Sci. Am.*, vol. 250, 1984. DOI: 10.1038/scientificamerican0684-84. [Online]. Available: https://doi.org/10.1038/scientificamerican0684-84.

[32] R. N. Jones, "McClintock's controlling elements: The full story," *Cytogenet. Genome Res.*, vol. 109, 2005. DOI: 10.1159/000082387. [Online]. Available: https://doi.org/10.1159/000082387.

[33] N. V. Fedoroff, "McClintock's challenge in the 21st century," *Proc. Natl Acad. Sci. USA*, vol. 109, 2012. DOI: 10.1073/pnas.1215482109. [Online]. Available: https://doi.org/10.1073/pnas.1215482109.

[34] B. Mcclintock, "Controlling elements and the gene," *Cold Spring Harb. Symp. Quant. Biol.*, vol. 21, 1956. DOI: 10.1101/SQB.1956.021.01.017. [Online]. Available: https://doi.org/10.1101/SQB.1956.021.01.017.

[35] X. Cui and X. Cao, "Epigenetic regulation and functional exaptation of transposable elements in higher plants," *Curr. Opin. Plant Biol.*, vol. 21, 2014. DOI: 10.1016/j.pbi.2014.07.001. [Online]. Available: https://doi.org/10.1016/j.pbi.2014.07.001.

[36] I. Makarevitch, "Transposable elements contribute to activation of maize genes in response to abiotic stress," *PLoS Genet.*, vol. 11, 2015. DOI: 10.1371/journal.pgen.1004915. [Online]. Available: https://doi.org/10.1371/journal.pgen.1004915.

[37] H. Zhao, "Proliferation of regulatory DNA elements derived from transposable elements in the maize genome," *Plant Physiol.*, vol. 176, 2018. DOI: 10.1104/pp.17.01467. [Online]. Available: https://doi.org/10.1104/pp.17.01467.

[38] J. M. Noshay, "Assessing the regulatory potential of transposable elements using chromatin accessibility profiles of maize transposons," *Genetics*, vol. 217, 2021. DOI: 10.1093/genetics/iyaa003. [Online]. Available: https://doi.org/10.1093/genetics/iyaa003.

[39] M. B. Hufford, "De novo assembly, annotation, and comparative analysis of 26 diverse maize genomes," *Science*, vol. 373, 2021. DOI: 10.1126/science.abg5289. [Online]. Available: https://doi.org/10.1126/science.abg5289.

[40] A. B. Stergachis, B. M. Debo, E. Haugen, L. S. Churchman, and J. A. Stamatoyannopoulos, "Single-molecule regulatory architectures captured by chromatin fiber sequencing," *Science*, vol. 368, 2020. DOI: 10.1126/science.aaz1646. [Online]. Available: https://doi.org/10.1126/science.aaz1646.

[41] Y. Kong, "Critical assessment of DNA adenine methylation in eukaryotes using quantitative deconvolution," *Science*, vol. 375, 2022. DOI: 10.1126/science.abe7489. [Online]. Available: https://doi.org/10.1126/science.abe7489.

[42] T. W. Tullius, "RNA polymerases reshape chromatin architecture and couple transcription on individual fibers," *Mol. Cell*, vol. 84, 2024. DOI: 10.1016/j.molcel.2024.08.013. [Online]. Available: https://doi.org/10.1016/j.molcel.2024.08.013.

[43] "Vollger, M. R. et al. A haplotype-resolved view of human gene regulation. Preprint at bioRxivhttps://doi.org/10.1101/2024.06.14.599122 (2024).,"

[44] K. L. Bubb and R. B. Deal, "Considerations in the analysis of plant chromatin accessibility data," *Curr. Opin. Plant Biol.*, vol. 54, 2020. DOI: 10.1016/j.pbi.2020.01.003. [Online]. Available: https://doi.org/10.1016/j.pbi.2020.01.003.

[45] A. P. Marand, Z. Chen, A. Gallavotti, and R. J. Schmitz, "A cis-regulatory atlas in maize at single-cell resolution," *Cell*, vol. 184, 2021. DOI: 10.1016/j.cell.2021.04.014. [Online]. Available: https://doi.org/10.1016/j.cell.2021.04.014.

[46] A. G. Uren, J. Kool, A. Berns, and M. Lohuizen, "Retroviral insertional mutagenesis: Past, present and future," *Oncogene*, vol. 24, 2005. DOI: 10.1038/sj.onc.1209043. [Online]. Available: https://doi.org/10.1038/sj.onc.1209043.

[47] R. M. Erdmann and C. L. Picard, "RNA-directed DNA methylation," *PLoS Genet.*, vol. 16, 2020. DOI: 10.1371/journal.pgen.1009034. [Online]. Available: https://doi.org/10.1371/journal.pgen.1009034.

[48] T. Jores, "Synthetic promoter designs enabled by a comprehensive analysis of plant core promoters," *Nat. Plants*, vol. 7, 2021. DOI: 10.1038/s41477-021-00932-y. [Online]. Available: https://doi.org/10.1038/s41477-021-00932-y.

[49] B. Leduque, A. Edera, C. Vitte, and L. Quadrana, "Simultaneous profiling of chromatin accessibility and DNA methylation in complete plant genomes using long-read sequencing," *Nucleic Acids Res.*, vol. 52, 2024. DOI: 10.1093/nar/gkae306. [Online]. Available: https://doi.org/10.1093/nar/gkae306.

[50] S. Takuno and B. S. Gaut, "Body-methylated genes in Arabidopsis thaliana are functionally important and evolve slowly," *Mol. Biol. Evol.*, vol. 29, 2012. DOI: 10.1093/molbev/msr188. [Online]. Available: https://doi.org/10.1093/molbev/msr188.

[51] N. Colonna Romano and L. Fanti, "Transposable elements: Major players in shaping genomic and evolutionary patterns," *Cells*, vol. 11, 2022. DOI: 10.3390/cells11061048. [Online]. Available: https://doi.org/10.3390/cells11061048.

[52] C. J. Cohen, W. M. Lock, and D. L. Mager, "Endogenous retroviral LTRs as promoters for human genes: A critical assessment," *Gene*, vol. 448, 2009. DOI: 10.1016/j.gene.2009.06.020. [Online]. Available: https://doi.org/10.1016/j.gene.2009.06.020.

[53] V. Sundaram, "Widespread contribution of transposable elements to the innovation of gene regulatory networks," *Genome Res.*, vol. 24, 2014. DOI: 10.1101/gr.168872.113. [Online]. Available: https://doi.org/10.1101/gr.168872.113.

[54] P. Medstrand, J. R. Landry, and D. L. Mager, "Long terminal repeats are used as alternative promoters for the endothelin B receptor and apolipoprotein C-I genes in humans," *J. Biol. Chem.*, vol. 276, 2001. DOI: 10.1074/jbc.M006557200. [Online]. Available: https://doi.org/10.1074/jbc.M006557200.

[55] C. A. Dunn, P. Medstrand, and D. L. Mager, "An endogenous retroviral long terminal repeat is the dominant promoter for human beta1,3-galactosyltransferase 5 in the colon," *Proc. Natl Acad. Sci. USA*, vol. 100, 2003. DOI: 10.1073/pnas.2134464100. [Online]. Available: https://doi.org/10.1073/pnas.2134464100.

[56] S. C. Stelpflug, R. S. Sekhon, B. Vaillancourt, *et al.*, "An Expanded Maize Gene Expression Atlas based on RNA Sequencing and its Use to Explore Root Development," *The Plant Genome*, vol. 9, no. 1, plantgenome2015.04.0025, 2016, ISSN: 1940-3372. DOI: 10.3835/plantgenome2015.04.0025.

[Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.3835/plantgenome2015.04.0025 (visited on 12/17/2025).

[57] N. Yang, "Two teosintes made modern maize," *Science*, vol. 382, 2023. DOI: 10.1126/science.adg8940. [Online]. Available: https://doi.org/10.1126/science.adg8940.

[58] J. A. Bailey, G. Liu, and E. E. Eichler, "An Alu transposition model for the origin and expansion of human segmental duplications," *Am. J. Hum. Genet.*, vol. 73, 2003. DOI: 10.1086/378594. [Online]. Available: https://doi.org/10.1086/378594.

[59] J. Cao, "Epigenetic and chromosomal features drive transposon insertion in Drosophila melanogaster," *Nucleic Acids Res.*, vol. 51, 2023. DOI: 10.1093/nar/gkad054. [Online]. Available: https://doi.org/10.1093/nar/gkad054.

[60] J. D. Buenrostro, P. G. Giresi, L. C. Zaba, H. Y. Chang, and W. J. Greenleaf, "Transposition of native chromatin for fast and sensitive epigenomic profiling of open chromatin, DNA-binding proteins and nucleosome position," *Nat. Methods*, vol. 10, 2013. DOI: 10.1038/nmeth.2688. [Online]. Available: https://doi.org/10.1038/nmeth.2688.

[61] N. Fedoroff, S. Wessler, and M. Shure, "Isolation of the transposable maize controlling elements Ac and Ds," *Cell*, vol. 35, 1983. DOI: 10.1016/0092-8674(83)90226-X. [Online]. Available: https://doi.org/10.1016/0092-8674(83)90226-X.

[62] U. Courage-Tebbe, H. P. Döring, N. Fedoroff, and P. Starlinger, "The controlling element Ds at the Shrunken locus in Zea mays: Structure of the unstable sh-m5933 allele and several revertants," *Cell*, vol. 34, 1983. DOI: 10.1016/0092-8674(83)90372-0. [Online]. Available: https://doi.org/10.1016/0092-8674(83)90372-0.

[63] M. Shure, S. Wessler, and N. Fedoroff, "Molecular identification and isolation of the *Waxy* locus in maize," *Cell*, vol. 35, no. 1, pp. 225–233, Nov. 1, 1983, ISSN: 0092-8674. DOI: 10.1016/0092-8674(83)90225-8. [Online]. Available: https://www.sciencedirect.com/science/article/pii/0092867483902258 (visited on 12/17/2025).

[64] N. V. Fedoroff, D. B. Furtek, and O. E. Nelson, "Cloning of the bronze locus in maize by a simple and generalizable procedure using the transposable controlling element Activator (Ac)," *Proc. Natl*

Acad. Sci. USA, vol. 81, 1984. DOI: 10.1073/pnas.81.12.3825. [Online]. Available: https://doi.org/10.1073/pnas.81.12.3825.

[65] J. Paz-Ares, U. Wienand, P. A. Peterson, and H. Saedler, "Molecular cloning of the c locus of Zea mays: A locus regulating the anthocyanin pathway," *EMBO J.*, vol. 5, 1986. DOI: 10.1002/j.1460-2075.1986.tb04291.x. [Online]. Available: https://doi.org/10.1002/j.1460-2075.1986.tb04291.x.

[66] S. C. Elgin, "DNAase I-hypersensitive sites of chromatin," *Cell*, vol. 27, 1981. DOI: 10.1016/0092-8674(81)90381-0. [Online]. Available: https://doi.org/10.1016/0092-8674(81)90381-0.

[67] A. M. Sullivan, "Mapping and dynamics of regulatory DNA and transcription factor networks in A. thaliana," *Cell Rep.*, vol. 8, 2014. DOI: 10.1016/j.celrep.2014.08.019. [Online]. Available: https://doi.org/10.1016/j.celrep.2014.08.019.

[68] S. L. French, Y. N. Osheim, F. Cioci, M. Nomura, and A. L. Beyer, "In exponentially growing Saccharomyces cerevisiae cells, rRNA synthesis is determined by the summed RNA polymerase I loading rate rather than by the number of active genes," *Mol. Cell. Biol.*, vol. 23, 2003. DOI: 10.1128/MCB.23.5.1558-1568.2003. [Online]. Available: https://doi.org/10.1128/MCB.23.5.1558-1568.2003.

[69] A. M. Muyle, D. K. Seymour, Y. Lv, B. Huettel, and B. S. Gaut, "Gene body methylation in plants: Mechanisms, functions, and important implications for understanding evolutionary processes," *Genome Biol. Evol.*, vol. 14, 2022. DOI: 10.1093/gbe/evac038. [Online]. Available: https://doi.org/10.1093/gbe/evac038.

[70] M. Munasinghe, "Combined analysis of transposable elements and structural variation in maize genomes reveals genome contraction outpaces expansion," *PLoS Genet.*, vol. 19, 2023. DOI: 10.1371/journal.pgen.1011086. [Online]. Available: https://doi.org/10.1371/journal.pgen.1011086.

[71] N. V. Fedoroff, "Transposable elements, epigenetics, and genome evolution," *Science*, vol. 338, 2012. DOI: 10.1126/science.338.6108.758. [Online]. Available: https://doi.org/10.1126/science.338.6108.758.

[72] W. Mo, "Single-molecule targeted accessibility and methylation sequencing of centromeres, telomeres and rDNAs in Arabidopsis," *Nat. Plants*, vol. 9, 2023. DOI: 10.1038/s41477-023-01498-7. [Online]. Available: https://doi.org/10.1038/s41477-023-01498-7.

[73] J. Tonnies, N. A. Mueth, S. Gorjifard, J. Chu, and C. Queitsch, "Scalable Transfection of Maize Mesophyll Protoplasts," *Journal of Visualized Experiments (JoVE)*, no. 196, e64991, Jun. 23, 2023, ISSN: 1940-087X. DOI: 10.3791/64991. [Online]. Available: https://app.jove.com/t/64991/scalable-transfection-of-maize-mesophyll-protoplasts (visited on 12/17/2025).

[74] "Fibertools-Rs: Tools for fiberseq data written in rust. GitHubhttps://github.com/fiberseq/fibertools-rs (2024).," [Online]. Available: https://github.com/fiberseq/fibertools-rs.

[75] "Hamm, M. Fiber-seq FIRE maize model. Zenodohttps://doi.org/10.5281/ZENODO.14641792 (2025).,"

[76] "FIRE: A snakemake workflow for calling fiber-seq inferred regulatory elements (FIREs) on single molecules. GitHubhttps://github.com/fiberseq/FIRE (2024).," [Online]. Available: https://github.com/fiberseq/FIRE.

[77] H. Li and R. Durbin, "Fast and accurate short read alignment with Burrows–Wheeler transform," *Bioinformatics*, vol. 25, 2009. DOI: 10.1093/bioinformatics/btp324. [Online]. Available: https://doi.org/10.1093/bioinformatics/btp324.

[78] "SAMtools. GitHubhttps://samtools.sourceforge.net/ (2024).," [Online]. Available: https://samtools.sourceforge.net/.

[79] J. M. Gaspar. "Improved peak-calling with MACS2." (Dec. 17, 2018), [Online]. Available: https://www.biorxiv.org/content/10.1101/496521v1 (visited on 12/17/2025), pre-published.

[80] R. .-. W. Hendron and S. Kelly, "Subdivision of light signaling networks contributes to partitioning of C4 photosynthesis," *Plant Physiol.*, vol. 182, 2020. DOI: 10.1104/pp.19.01053. [Online]. Available: https://doi.org/10.1104/pp.19.01053.

[81] M. K. Mejía-Guerra, "Core promoter plasticity between maize tissues and genotypes contrasts with predominance of sharp transcription initiation sites," *Plant Cell*, vol. 27, 2015. DOI: 10.1105/tpc.15.00630. [Online]. Available: https://doi.org/10.1105/tpc.15.00630.

[82] A. S. Hinrichs, "The UCSC Genome Browser Database: Update 2006," *Nucleic Acids Res.*, vol. 34, 2006. DOI: 10.1093/nar/gkj144. [Online]. Available: https://doi.org/10.1093/nar/gkj144.

[83] "Welcome to MaizeGDB (MaizeGDB, 2024); https://www.maizegdb.org/," [Online]. Available: https://www.maizegdb.org/.

[84] T. K. Wolfgruber, "Maize centromere structure and evolution: Sequence analysis of centromeres 2 and 5 reveals dynamic loci shaped primarily by retrotransposons," *PLoS Genet.*, vol. 5, 2009. DOI: 10.1371/journal.pgen.1000743. [Online]. Available: https://doi.org/10.1371/journal.pgen.1000743.

[85] S. Neph, "BEDOPS: High-performance genomic feature operations," *Bioinformatics*, vol. 28, 2012. DOI: 10.1093/bioinformatics/bts277. [Online]. Available: https://doi.org/10.1093/bioinformatics/bts277.

[86] J. G. Wallace, "Association mapping across numerous traits reveals patterns of functional variation in maize," *PLoS Genet.*, vol. 10, 2014. DOI: 10.1371/journal.pgen.1004845. [Online]. Available: https://doi.org/10.1371/journal.pgen.1004845.

[87] C. Camacho, "BLAST+: Architecture and applications," *BMC Bioinformatics*, vol. 10, 2009. DOI: 10.1186/1471-2105-10-421. [Online]. Available: https://doi.org/10.1186/1471-2105-10-421.

[88] J. E. Moore, M. J. Purcaro, H. E. Pratt, *et al.*, "Expanded encyclopaedias of DNA elements in the human and mouse genomes," *Nature*, vol. 583, no. 7818, pp. 699–710, Jul. 2020, ISSN: 1476-4687. DOI: 10.1038/s41586-020-2493-4. [Online]. Available: https://www.nature.com/articles/s41586-020-2493-4 (visited on 12/21/2025).

[89] M. Farahbod, A. R. Diab, P. Sud, *et al.* "Integrative chromatin state annotation of 234 human ENCODE4 cell types using Segway reveals disease drivers." (Oct. 31, 2023), [Online]. Available: https://www.biorxiv.org/content/10.1101/2023.10.26.564254v1 (visited on 12/22/2025), pre-published.

Appendix A

fiber-views reference

This documentation can be found online at https://fiber-views.readthedocs.io/en/latest/fiber_views.html

A.1 fiber_views.fiber_views module

Main module.

fiber_views.fiber_views.**bed_to_anno_df**(*bed_df*, *entry_name_type='gene_id'*, *aligned_pos='start'*)

Convert a data frame in BED format to another data frame with a different layout.

Parameters

- **bed_df** (*pandas.DataFrame*) – Data frame in BED format, with columns 'chrom', 'start', 'end', 'strand', 'name', and 'score'.
- **entry_name_type** (*str, optional*) – Column name for the unique identifier for each feature. The default is "gene_id".
- **aligned_pos** (*str, optional*) – The position in the bed entries to use as the 'pos' or aligned base. Can be 'start', 'end', or 'center' (default is 'start').

Returns

anno_df – Data frame with columns 'seqid', 'pos', 'strand', [entry_name_type], and 'score'.

Return type

pandas.DataFrame

fiber_views.fiber_views.**build_multi_fview**(*bam_file*, *sites_df*, *mod_defs*, *region_defs*, *window=(-1000, 1000)*, *fully_span=True*, *region_interval=30*, *filter_args={'cutoff': 2, 'dist': 3000}*, *tags=['np', 'ec', 'rq']*, *max_reads=300*)

Build an AnnData object centered at a multiple genomic sites.

Parameters

- **bam_file** (*str*) – Path to the BAM file containing Fiber-seq reads.
- **sites_df** (*pandas.DataFrame*) – genomic positions to center on, pandas.DataFrame with columns 'seqid', 'pos', and 'strand'.
- **mod_defs** (*list of dict*) – List of modification definitions, each dict describing a modification to extract.
- **region_defs** (*list of dict*) – List of region definitions, each dict describing a region type to extract.
- **window** (*tuple of int, optional*) – window (upstream, downstream) around the site to extract (default is (-1000, 1000)).
- **fully_span** (*bool, optional*) – If True, only include reads fully spanning the window (default is True).
- **region_interval** (*int, optional*) – Interval size used for region feature binning (default is 30).
- **filter_args** (*dict, optional*) – Arguments for filtering reads by methylation endpoints, should include 'dist' and 'cutoff' (default is {'dist': 3000, 'cutoff': 2}).
- **tags** (*list of str, optional*) – List of BAM tags to extract for annotation (default is ['np', 'ec', 'rq']).
- **max_reads** (*int, optional*) – Maximum number of reads to extract (default is 300).

Returns

Annotated data matrix containing read, sequence, modification, and region layers for the site. Returns None if no reads pass filtering.

Return type

AnnData or None

fiber_views.fiber_views.**build_single_fview**(*bam_file*, *site_info*, *mod_defs='PacBio_Fiberseq'*, *region_defs='FIRE'*, *window=(-1000, 1000)*, *fully_span=True*, *region_interval=30*, *filter_args={'cutoff': 2, 'dist': 3000}*, *tags=['np', 'ec', 'rq']*, *max_reads=300*)

Build an AnnData object centered at a single genomic site.

Parameters

- **bam_file** (*str*) – Path to the BAM file containing Fiber-seq reads.
- **site_info** (*dict or pandas.Series*) – genomic position to center on, dict or series with keys 'seqid', 'pos', and 'strand'.
- **mod_defs** (*str OR list of dicts*) – List of modification definitions, each dict describing a modification to extract.
- **region_defs** (*str OR list of dict*) – List of region definitions, each dict describing a region type to extract.
- **window** (*tuple of int, optional*) – window (upstream, downstream) around the site to extract (default is (-1000, 1000)).
- **fully_span** (*bool, optional*) – If True, only include reads fully spanning the window (default is True).
- **region_interval** (*int, optional*) – Interval size used for region feature binning (default is 30).
- **filter_args** (*dict, optional*) – Arguments for filtering reads by methylation endpoints, should include 'dist' and 'cutoff' (default is {'dist': 3000, 'cutoff': 2}).
- **tags** (*list of str, optional*) – List of BAM tags to extract for annotation (default is ['np', 'ec', 'rq']).
- **max_reads** (*int, optional*) – Maximum number of reads to extract (default is 300).

Returns

Annotated data matrix containing read, sequence, modification, and region layers for the site. Returns None if no reads pass filtering.

Return type

AnnData or None

fiber_views.fiber_views.**read_bed**(*bed_file*)

Read a BED file and return a pandas DataFrame.

Parameters

bed_file (*str*) – The file path of the BED file to be read. The bed file should follow bed standard and not include column names.

Returns

A DataFrame containing the data from the BED file.

Return type

pandas.DataFrame

A.2 fiber_views.plot module

Created on Wed Aug 28 13:33:42 2024

@author: morgan

fiber_views.plot.**annotate_boundaries**(*fview*)

Add start and end position columns to the observation metadata of a fiber view.

This function calculates the leftmost and rightmost positions of actual sequence data (excluding gaps) for each fiber and adds them as 's_pos' and 'e_pos' columns to the obs dataframe.

Parameters

fview (*anndata.AnnData*) – The fiber view object to annotate.

Returns

The function modifies the fiber view object in place.

Return type

None

fiber_views.plot.**draw_fiber_bars**(*fview*, *ax=None*, *color='#d0d0d0'*, *width=0.8*)

Draw fibers as horizontal bars on a matplotlib axis.

This function draws each fiber in the fiber view as a rectangular bar spanning from the start to the end of the sequence data. This visualization is suitable for fewer than ~150 fibers.

Parameters

- **fview** (*anndata.AnnData*) – The fiber view object containing sequence data.
- **ax** (*matplotlib.axes.Axes, optional*) – An existing axis to draw on. If None, a new axis will be created. The default is None.
- **color** (*str, optional*) – The color of the fiber bars. The default is "#d0d0d0".
- **width** (*float, optional*) – The width (height) of each bar in axis units. The default is DEFAULT_WIDTH (0.8).

Returns

The axis with fibers drawn as horizontal bars.

Return type

matplotlib.axes.Axes

fiber_views.plot.**draw_fiber_lines**(*fview*, *ax=None*, *color='#606060'*)

Draw fibers as horizontal lines on a matplotlib axis.

This function draws each fiber in the fiber view as a horizontal line spanning from the start to the end of the sequence data. This visualization is suitable for fewer than ~150 fibers.

Parameters

- **fview** (*anndata.AnnData*) – The fiber view object containing sequence data.

- **ax** (*matplotlib.axes.Axes, optional*) – An existing axis to draw on. If None, a new axis will be created. The default is None.
- **color** (*str, optional*) – The color of the fiber lines. The default is "#606060".

Returns
The axis with fibers drawn as horizontal lines.

Return type
matplotlib.axes.Axes

fiber_views.plot.**draw_mods**(*fview*, *ax=None*, *mod='m6a'*, *width=0.8*, *color='#000000'*)

Draw base modifications as vertical marks on a matplotlib axis.

This function visualizes base modifications (such as m6A or CpG methylation) as small vertical rectangles at each modified position along the fibers.

Parameters

- **fview** (*anndata.AnnData*) – The fiber view object containing modification data.
- **ax** (*matplotlib.axes.Axes, optional*) – An existing axis to draw on. If None, a new axis will be created. The default is None.
- **mod** (*str, optional*) – The name of the modification layer to draw (e.g., 'm6a', 'cpg'). The default is 'm6a'.
- **width** (*float, optional*) – The width (height) of each modification mark in axis units. The default is DEFAULT_WIDTH (0.8).
- **color** (*str, optional*) – The color of the modification marks. The default is '#000000' (black).

Returns
The axis with modifications drawn as vertical marks.

Return type
matplotlib.axes.Axes

fiber_views.plot.**draw_mods_offset**(*fview*, *ax=None*, *mod='m6a'*, *width=0.8*, *color='#000000'*)

fiber_views.plot.**draw_regions**(*fview*, *ax=None*, *base_name='msp'*, *color='red'*, *width=0.8*)

Draw genomic regions as colored rectangles on a matplotlib axis.

This function visualizes regions (such as nucleosomes or MSPs) as colored rectangles overlaid on the fiber view. Each region is drawn at its corresponding position along the fiber.

Parameters

- **fview** (*anndata.AnnData*) – The fiber view object containing region data.

- **ax** (*matplotlib.axes.Axes, optional*) – An existing axis to draw on. If None, a new axis will be created. The default is None.
- **base_name** (*str, optional*) – The name of the region type to draw (e.g., 'msp', 'nuc', 'fire'). The default is 'msp'.
- **color** (*str, optional*) – The color of the region rectangles. The default is "red".
- **width** (*float, optional*) – The width (height) of each rectangle in axis units. The default is DEFAULT_WIDTH (0.8).

Returns
The axis with regions drawn as colored rectangles.

Return type
matplotlib.axes.Axes

fiber_views.plot.**draw_split_lines**(*fview*, *ax=None*, *split_var='site_name'*, *color='black'*)

Draw horizontal lines separating groups in a fiber view.

This function draws horizontal dividing lines between groups of fibers based on a grouping variable in the observation metadata. This is useful for visually separating fibers from different sites or conditions.

Parameters
- **fview** (*anndata.AnnData*) – The fiber view object with grouped observations.
- **ax** (*matplotlib.axes.Axes, optional*) – An existing axis to draw on. If None, a new axis will be created. The default is None.
- **split_var** (*str, optional*) – The column name in obs to use for determining group boundaries. The default is "site_name".
- **color** (*str, optional*) – The color of the dividing lines. The default is "black".

Returns
The axis with horizontal dividing lines drawn between groups.

Return type
matplotlib.axes.Axes

fiber_views.plot.**make_plot_ax**(*fview*, *ax=None*)

Create or prepare a matplotlib axis for plotting a fiber view.

Parameters
- **fview** (*anndata.AnnData*) – The fiber view object to plot.
- **ax** (*matplotlib.axes.Axes, optional*) – An existing axis to use. If None, a new figure and axis will be created. The default is None.

Returns
The prepared axis with xlim and ylim set appropriately for the fiber view.

Return type
matplotlib.axes.Axes

A.3 fiber_views.tools module

Created on Wed Sep 7 14:37:04 2022

@author: morgan

@description: A set of usefull tools for workign with fiber views

fiber_views.tools.**agg_by_obs_and_bin**(*fview*, *obs_group_var='site_name'*, *bin_width=10*, *obs_to_keep=['seqid', 'pos', 'strand', '']*, *fast=True*, *region_weights='ones'*)

Aggregate fiber view data by a group variable in the *obs* dataframe and bin by *bin_widht* basepairs.

Parameters

- **fview** (*anndata.AnnData*) – The fiber view object containing the data to be aggregated.
- **obs_group_var** (*str, optional*) – The name of the *obs* group variable to use for aggregation. The default value is 'site_name'. If *obs_group_var* is set to None, the fiber view will not be aggregated by rows and the row ordering will be preserved.
- **bin_width** (*int, optional*) – The width of each bin, in base pairs. The default value is 10. If 'bin_width' is 1, the data will not be binned.
- **obs_to_keep** (*list of str, optional*) – A list of observation metadata columns to keep in the aggregated data. The default value is ['seqid', 'pos', 'strand', ''].
- **fast** (*bool, optional*) – If True, the modification matrices will be converted to dense matrices for faster calculations. The default value is True. This may use more memory for large fiber view objects.
- **region_weights** (*str, optional*) – how to weight regions when aggregating must be one of 'ones', 'length' or 'score'

Returns
An aggregated version of the input fiber view object, with observations grouped and binned according to the specified parameters.

Return type
anndata.AnnData

fiber_views.tools.**bin_sparse_regions**(*fview*, *base_name='nuc'*, *bin_width=10*, *interval=3*)

Bin regions in a fiber view by averaging their length and score over a set of consecutive bins.

Parameters

- **fview** (*anndata.AnnData*) – The fiber view object containing the region data.
- **base_name** (*str, optional*) – The name of the type of regions to bin. This should be one of 'nuc' (nucleosomes), or 'msp' (methylation sensitive patches). The default value is 'nuc'.
- **bin_width** (*int, optional*) – The width of each bin, in base pairs. The default value is 10.
- **interval** (*int, optional*) – The interval between bins, in base pairs. The default value is 3.

Returns

A tuple containing the position, length, and score data for the binned regions, stored as COOrdinate format sparse matrices.

Return type

tuple of scipy.sparse.coo_matrix

fiber_views.tools.**calc_kmer_dist**(*fview*, *metric='cityblock'*)

Calculate pairwise k-mer distances between fibers in a fiber view.

This function calculates pairwise distances between fibers in a fiber view based on the k-mer counts stored in the 'kmers' element of the *obsm* attribute. The distance metric can be specified using the *metric* parameter (default is 'cityblock'). The resulting distance matrix is stored in the 'kmer_dist' element of the *obsp* attribute of the fiber view.

Parameters

- **fview** (*anndata.AnnData*) – Fiber view object containing k-mer count data in the 'kmers' element of the *obsm* attribute.
- **metric** (*str, optional*) – Distance metric to use for calculating pairwise distances. The default is 'cityblock'.

Return type

None

fiber_views.tools.**count_kmers**(*fview*, *k*)

Count k-mers in each fiber in a fiber view.

This function counts the occurrences of k-mers in each fiber in a fiber view, and stores the resulting k-mer counts in the 'kmers' element of the *obsm* attribute of the fiber view. The length of the k-mers (*k*) and the mapping from k-mer strings to column indices in the k-mer count matrix are stored in the 'kmer_len' and 'kmer_idx' elements of the *uns* attribute, respectively.

Parameters

- **fview** (*anndata.AnnData*) – Fiber view object containing DNA sequence data in the 'seq' element of the *layers* attribute.
- **k** (*int*) – Length of the k-mers to count.

Returns

The function updates the fiber view object in place, adding a new observation matrix 'kmers' containing the counts of each k-mer for each fiber, and adds two new entries to the 'uns' dictionary: 'kmer_len' and 'kmer_idx'. 'kmer_len' is the length of the k-mers that were counted, and 'kmer_idx' is a list of the k-mers that were counted, with each k-mer represented as a bytes object.

Return type

None

fiber_views.tools.**filter_regions**(*fview*, *base_name='nuc'*, *new_base_name=None*, *length_limits=(-inf, inf)*, *score_limits=(-inf, inf)*, *inplace=False*)

Filter base modifications in a fiber view by length and score limits.

Parameters

- **fview** (*anndata.AnnData*) – The fiber view object containing the base modification data.
- **base_name** (*str, optional*) – The name of the type of regions to bin. This should be one of 'nuc' (nucleosomes), or 'msp' (methylation sensitive patches). The default value is 'nuc'.
- **new_base_name** (*str, optional*) – If not *None*, the new region base name to save the filtered regions to (region information at base_name will not be modified). If new_base_name is None, the filtered regions will be saved to base_name.
- **length_limits** (*tuple of float, optional*) – The lower and upper limits for the length of the base modifications. Modifications with lengths outside of these limits will be filtered out. The default value is (-inf, inf), which includes all modifications.
- **score_limits** (*tuple of float, optional*) – The lower and upper limits for the score of the base modifications. Modifications with scores outside of these limits will be filtered out. The default value is (-inf, inf), which includes all modifications.
- **inplace** (*bool, optional*) – If True, the function will filter the base modifications in place and return None. If False (default), the function will return a new fiber view.

Returns

The function updates the fiber view object in place or returns a new fiber view with the selected region type filtered.

Return type
None or anndata.AnnData

fiber_views.tools.**get_seq_records**(*fview*, *id_col='read_name'*)

Convert fiber view sequences to BioPython SeqRecord objects.

Parameters

- **fview** (*anndata.AnnData*) – The fiber view object containing sequence data.
- **id_col** (*str, optional*) – The column name in obs to use as the sequence ID. The default is "read_name".

Returns
A list of SeqRecord objects where each record contains the sequence from one row of the fiber view.

Return type
list of Bio.SeqRecord.SeqRecord

fiber_views.tools.**get_sequences**(*fview*)

Returns a list of strings where each string is the sequence of one row of the fview object.

Parameters
fview (*AnnData object*) – The fiber view object containing the sequence data.

Returns
sequences – A list of strings where each string is the sequence of one row of the fview object.

Return type
list

fiber_views.tools.**make_dense_regions**(*fview*, *base_name='nuc'*, *report='ones'*)

Create a dense matrix containing a representation of region infromation in a fiber view.

Parameters

- **fview** (*anndata.AnnData*) – The fiber view object containing the base modification data.
- **base_name** (*str, optional*) – The name of the type of regions to bin. This should be one of 'nuc' (nucleosomes), or 'msp' (methylation sensitive patches). The default value is 'nuc'.
- **report** (*str, optional*) – The data to include in the dense matrix. This should be one of 'ones', 'score' or 'length'. The default value is 'score'.

Returns
A dense matrix of size (number of fibers, number of bases) containing the specified region data. Each position in the matrix where a region is not present is set to 0, positions where ar region is present may be set to either the length or score value of the region occupying that position.

Return type
numpy.ndarray

fiber_views.tools.**make_region_df**(*fview*, *base_name='nuc'*, *zero_pos='left'*)

Create a dataframe containing the positions and lengths of regions in a fiber view.

Parameters

- **fview** (*anndata.AnnData*) – The fiber view object containing the base modification data.
- **base_name** (*str, optional*) – The name of the type of regions to bin. This should be one of 'nuc' (nucleosomes), or 'msp' (methylation sensitive patches). The default value is 'nuc'.
- **zero_pos** (*str, optional*) – The position to use as the zero point for the start positions of the base modifications. This should be one of 'left', 'center', or 'right'. The default value is 'left'.

Returns
A dataframe with columns 'row' (the fiber index), 'start' (the start position of the base modification), 'length' (the length of the base modification), and 'score' (the score of the base modification).

Return type
pandas.DataFrame

fiber_views.tools.**mark_cpg_sites**(*fview*, *sparse=True*)

Identify and mark CpG sites in a fiber view.

This function creates a new layer 'cpg_sites' in the fiber view with True values at positions that are CpG dinucleotides (C followed by G). The 'cpg_sites' layer is also added to the 'mods' list in uns.

Parameters

- **fview** (*anndata.AnnData*) – The fiber view object to mark CpG sites in.
- **sparse** (*bool, optional*) – If True, store the CpG sites as a sparse matrix. If False, store as a dense array. The default is True.

Returns
The function modifies the fiber view object in place, adding a 'cpg_sites' layer.

Return type
None

Notes

Known issue: All Cs at the end of each sequence are marked as not CpGs.

fiber_views.tools.**split_fire**(*fview*, *input_region='msp'*, *threshold=1*, *output_regions=['lnk', 'fire']*)

Split methylation-sensitive patches (MSPs) into linker and FIRE regions based on score.

This function creates two new region types by filtering the input region type based on a score threshold. Regions with scores below the threshold are classified as one type (default: linker), and regions with scores above the threshold are classified as another type (default: FIRE).

Parameters

- **fview** (*anndata.AnnData*) – The fiber view object containing region data.
- **input_region** (*str, optional*) – The name of the region type to split. The default is 'msp'.
- **threshold** (*float, optional*) – The score threshold for splitting regions. The default is 1.
- **output_regions** (*list of str, optional*) – A list of two names for the output region types [low_score, high_score]. The default is ['lnk', 'fire'].

Returns

The function modifies the fiber view object in place, adding new region layers.

Return type

None

A.4 fiber_views.utils module

Created on Tue Aug 30 16:05:34 2022

@author: morgan

class fiber_views.utils.**ReadList**(*normal_list=[]*, *strand='+'*)

Bases: list

A simple list of pysam.libcalignedsegment.PileupRead objects, plus methods useful for constructing anndata elements from the read objects. also tracks strand info of the genomic query position.

Parameters

- **normal_list** (*list, optional*) – A list of pysam.libcalignedsegment.PileupRead objects. The default is [].
- **strand** (*str, optional*) – The strand of the genomic query position. The default is "+".

strand

The strand of the genomic query position.

Type

str

build_anno_df(*anno_series*, *tags=['np', 'ec', 'rq']*)

Create a data frame with annotation data for the reads in the *ReadList* object.

Parameters

- **anno_series** (*pandas Series*) – A pandas Series containing annotation data for the genomic query position.
- **tags** (*list, optional*) – A list of tags to include in the data frame. The default is ['np', 'ec', 'rq'].

Returns

df – A data frame with annotation data for the reads in the *ReadList* object.

Return type

pandas DataFrame

build_mod_array(*window*, *mod_type=[('A', 0, 'a'), ('T', 1, 'a')]*, *strand=None*, *sparse=True*, *score_cutoff=200*)

Create a base modification matrix for the reads in the *ReadList* object.

Parameters

- **window** (*tuple*) – A tuple of integers representing the window of +/- window_offset. The tuple should be of the form (window_start, window_end).
- **mod_type** (*list, optional*) – A list of tuples representing the base modification type to consider. The default is M6A_MODS.
- **strand** (*str, optional*) – The strand of the genomic query position. If not provided, the strand information of the *ReadList* object is used. The default is None.
- **sparse** (*bool, optional*) – If True, the base odification matrix is returned in sparse format. If False, the matrix is returned in dense format. The default is True.
- **score_cutoff** (*int, optional*) – The minimum score required for a base modification to be considered. The default is 200.

Returns

mod_mtx – A base modification matrix for the reads in the *ReadList* object.

Return type

numpy array or scipy sparse matrix

build_mod_array_from_def(*window*, *mod_def*, *strand=None*, *sparse=True*)

Create a base modification matrix for reads using a modification definition dictionary.

Parameters

- **window** (*tuple*) – A tuple of integers representing the window of +/- window_offset. The tuple should be of the form (window_start, window_end).

- **mod_def** (*dict*) – A modification definition dictionary containing 'mod_code', 'threshold', and 'rev_offset' keys.
- **strand** (*str, optional*) – The strand of the genomic query position. If not provided, the strand information of the *ReadList* object is used. The default is None.
- **sparse** (*bool, optional*) – If True, the base modification matrix is returned in sparse format. If False, the matrix is returned in dense format. The default is True.

Returns

A base modification matrix for the reads in the *ReadList* object, with modifications defined by mod_def.

Return type

scipy.sparse.coo_matrix or numpy.ndarray

build_seq_array(*window*, *strand=None*)

Create a byte array of the sequences for the reads in the *ReadList* object.

Parameters

- **window** (*tuple*) – A tuple of integers representing the window of +/- window_offset. The tuple should be of the form (window_start, window_end).
- **strand** (*str, optional*) – The strand of the genomic query position. If not provided, the strand information of the *ReadList* object is used. The default is None.

Returns

char_array – A byte array of the sequences for the reads in the *ReadList* object.

Return type

numpy array

Notes

Make sure to filter the reads in the *ReadList* object before using this method.

build_sparse_region_array(*window*, *tags=('ns', 'nl')*, *interval=30*, *strand=None*)

Create a sparse region matrix for the reads in the *ReadList* object.

Parameters

- **window** (*tuple*) – A tuple of integers representing the window of +/- window_offset. The tuple should be of the form (window_start, window_end).
- **tags** (*tuple, optional*) – A tuple of tags to consider. The default is ('ns', 'nl').

- **interval** (*int, optional*) – The interval at which to report region information. This determines the minimum window size that can be subset to that still preserve region info. The default is 30.
- **strand** (*str, optional*) – The strand of the genomic query position. If not provided, the strand information of the *ReadList* object is used. The default is None.

Returns

region_mtx – A sparse region matrix for the reads in the *ReadList* object.

Return type

scipy sparse matrix

filter_by_end_meth(*dist=3000*, *cutoff=2*, *inplace=False*)

Remove reads if they have fewer than [cutoff] m6A mods within [dist] base pairs of the read start and end.

Parameters

- **dist** (*int, optional*) – The distance in base pairs from the read start and end to consider. The default is 3000.
- **cutoff** (*int, optional*) – The minimum number of m6A mods within [dist] base pairs of the read start and end required to keep the read. The default is 2.
- **inplace** (*bool, optional*) – If True, the *ReadList* object is modified in place. If False, a new *ReadList* object is returned. The default is False.

Returns

If *inplace* is True, returns None. If *inplace* is False, returns a new *ReadList* object with the filtered reads and strand information.

Return type

None or *ReadList*

filter_by_window(*window*, *inplace=False*, *strand=None*)

Remove reads that do not fully span a given window of +/- window_offset.

Parameters

- **window** (*tuple*) – A tuple of integers representing the window of +/- window_offset. The tuple should be of the form (window_start, window_end).
- **inplace** (*bool, optional*) – If True, the *ReadList* object is modified in place. If False, a new *ReadList* object is returned. The default is False.
- **strand** (*str, optional*) – The strand of the genomic query position. If not provided, the strand information of the *ReadList* object is used. The default is None.

Returns

If *inplace* is True, returns None. If *inplace* is False, returns a new *ReadList* object with the filtered reads and strand information.

Return type

None or *ReadList*

get_reads(*alignment_file*, *ref_pos*, *max_reads*)

Retrieve reads from a BAM file for a given reference position.

Parameters

- **alignment_file** (*pysam.AlignmentFile*) – A BAM file opened with pysam.
- **ref_pos** (*tuple*) – A tuple containing the reference name, position, and strand of the genomic query position. The tuple should be of the form (reference_name, position, strand).
- **max_reads** (*int*) – the max number of reads to load from the bam file, usefull to speed up processing when coverage is deep.

Returns

self – The *ReadList* object with the reads and strand information.

Return type

ReadList

Example

reads = ReadList().get_reads(bamfile, ('chr3', 200000, '+'))

print_aligned_centers(*offset=5*)

Print the center positions of the reads in the *ReadList* object with a specified number of bases on either side. This is a test function to make sure reads are aligning correctly

Parameters

offset (*int, optional*) – The number of bases on either side of the center position to include in the output. The default is 5.

Return type

None

fiber_views.utils.**get_mod_pos_from_rec**(*rec*, *mods=[('A', 0, 'a'), ('T', 1, 'a')]*, *score_cutoff=200*)

Retrieve positions of modified bases in a record.

Parameters

- **rec** (*pysam.libcalignedsegment.AlignedSegment*) – A record containing modified bases.

- **mods** (*list, optional*) – A list of modified bases to consider, in the form (base, index, code). The default is M6A_MODS.
- **score_cutoff** (*int, optional*) – The minimum score required for a modified base to be included. The default is 200.

Returns
mod_positions – An array of positions of modified bases.

Return type
numpy.ndarray

Example

mod_positions = get_mod_pos_from_rec(read.alignment)

fiber_views.utils.**get_strand_correct_mods**(*read*, *mod_type=[('A', 0, 'a'), ('T', 1, 'a')]*, *centered=False*, *score_cutoff=200*)

Retrieve modified bases in a read and correct their positions to match the forward genomic strand.

Parameters

- **read** (*pysam.libcalignedsegment.AlignedSegment*) – A read containing modified bases.
- **mod_type** (*list, optional*) – A list of modified bases to consider, in the form (base, index, code). The default is M6A_MODS.
- **centered** (*bool, optional*) – Whether to center the positions around the query position of the read. The default is False.
- **score_cutoff** (*int, optional*) – The minimum score required for a modified base to be included. The default is 200.

Returns
mods – An array of positions of modified bases, corrected for strand.

Return type
numpy.ndarray

Example

mods = get_strand_correct_mods(read)

fiber_views.utils.**get_strand_correct_mods_from_def**(*read*, *mod_def*, *centered=False*)

Retrieve modified bases in a read using a modification definition and correct positions for strand.

This function extracts modification positions from a read using a custom modification definition dictionary and corrects the positions to match the forward genomic strand.

Parameters

- **read** (*pysam.libcalignedsegment.AlignedSegment*) – A read containing modified bases.
- **mod_def** (*dict*) – A modification definition dictionary containing 'mod_code' (list of tuples), 'threshold' (int), and 'rev_offset' (int) keys.
- **centered** (*bool, optional*) – Whether to center the positions around the query position of the read. The default is False.

Returns

An array of positions of modified bases, corrected for strand. Returns None if no modifications are found.

Return type

numpy.ndarray or None

Example

mods = get_strand_correct_mods_from_def(read, mod_def)

fiber_views.utils.**get_strand_correct_regions**(*read*, *tags=('ns', 'nl')*, *centered=False*)

Retrieve start positions, lengths, and scores of regions in a read and correct them to match the forward genomic strand.

Parameters

- **read** (*pysam.libcalignedsegment.AlignedSegment*) – A read containing regions.
- **tags** (*tuple, optional*) – A tuple of tags containing the start positions, lengths, and scores of the regions. The default is ('ns', 'nl').
- **centered** (*bool, optional*) – Whether to center the positions around the query position of the read. The default is False.

Returns

- **starts** (*numpy.ndarray*) – An array of start positions of regions, corrected for strand.
- **lengths** (*numpy.ndarray*) – An array of lengths of regions.
- **scores** (*numpy.ndarray*) – An array of scores of regions.

Example

starts, lengths, scores = get_strand_correct_regions(read)

fiber_views.utils.**make_sparse_regions**(*region_df*, *shape*, *bin_width=1*, *interval=30*)

Make a sparse matrix representing genomic regions.

This function takes a DataFrame containing region information, as well as the shape of the resulting matrix and other parameters, and returns three sparse matrices representing the positions, lengths, and scores of the regions within the matrix.

Parameters

- **region_df** (*pandas.DataFrame*) – A DataFrame containing the region information. The DataFrame should have columns for *row*, *start*, *length*, and *score*, representing the row index of the matrix, the starting position of the region (0-based), the length of the region, and the score associated with the region, respectively.
- **shape** (*tuple*) – The shape of the resulting matrix before binning. The first element should be the number of rows in the matrix, and the second element should be the number of columns.
- **bin_width** (*int, optional*) – The width of each bin in the resulting matrix, in base pairs. The default is 1.
- **interval** (*int, optional*) – The interval at which to report region information. This determines the minimum window size that can be subset to that still preserve region info. The default is 30. interval is in number of bins, not bp

Returns

A tuple containing three sparse matrices representing the positions, lengths, and scores of the regions within the matrix. The matrices are in the form of COO sparse matrices. position values are still in base pairs after binning. and may be negative for the first reported pos of a region

Return type

tuple of scipy.sparse.coo_matrix

fiber_views.utils.**print_mod_contexts**(*read*, *mod_positions*, *offset=5*, *use_strand=True*)

Print the contexts surrounding modified bases in a read.

Parameters

- **read** (*pysam.libcalignedsegment.AlignedSegment*) – A read containing modified bases.
- **mod_positions** (*numpy.ndarray*) – An array of positions of modified bases in the read.
- **offset** (*int, optional*) – The number of bases on either side of the modified base to include in the context. The default is 5.
- **use_strand** (*bool, optional*) – Whether to use the strand information in the read to determine the context. If True, the context will be reversed if the read is on the negative strand. The default is True.

Example

print_mod_contexts(read, mod_positions)

Appendix B

Supplemental material for chapter 3

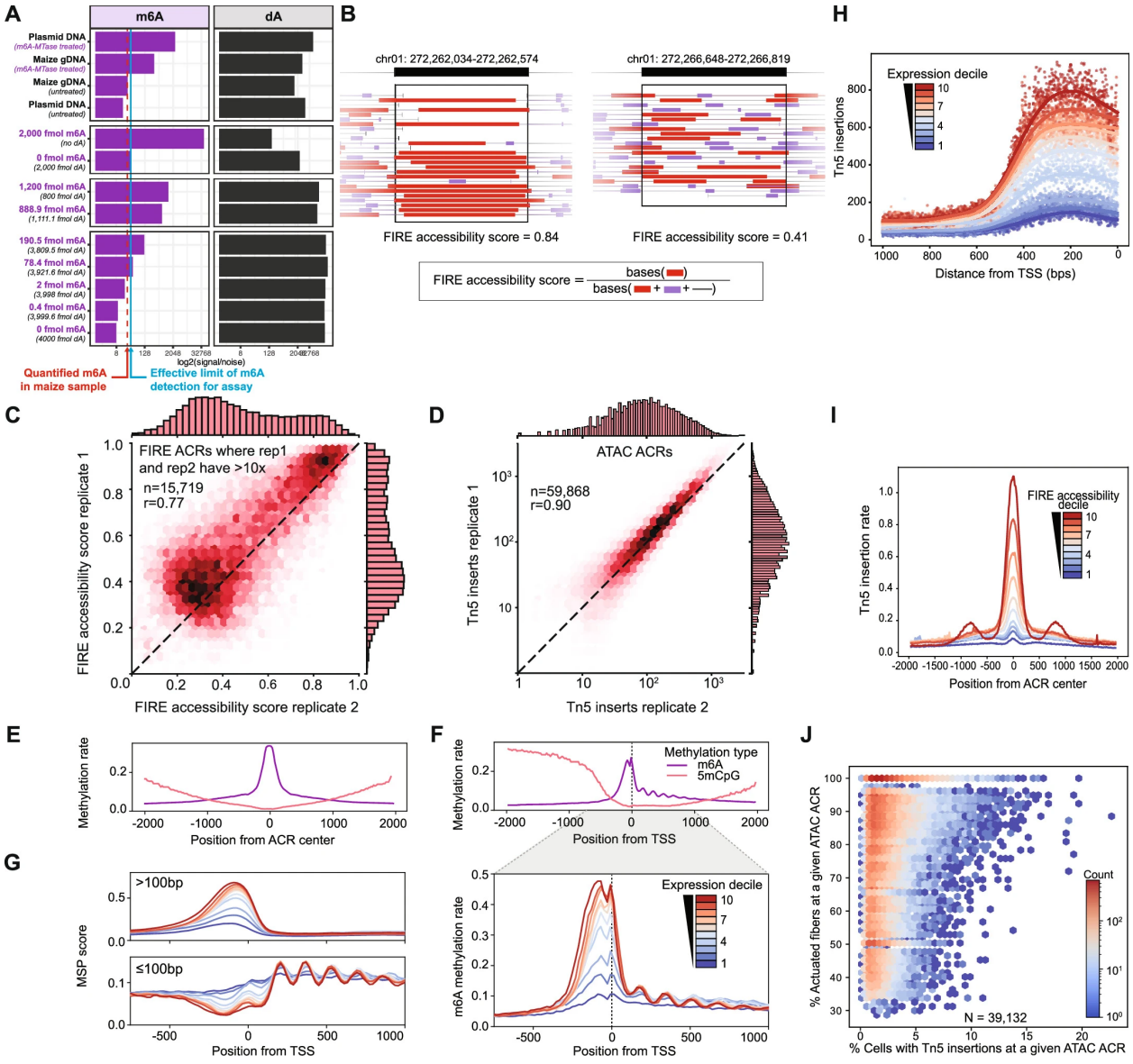

Fig. B.1: (Caption next page.)

Fig. B.1: (Previous page.) **(A)** Small molecule mass spectrometry data on nucleotides purified from various samples with known and unknown quantities of m6A and adenine (dA). Shown is the log-scale of the signal-to-noise ratio for each sample. Includes plasmid DNA isolated from bacteria that lack any m6A-MTases. As positive controls these samples were treated with a non-specific m6A-MTase before nucleotide isolation. For maize, samples were prepared as for Fiber-seq with or without m6A-MTase-treatment. In addition, standards containing defined amounts of m6A and dA were used. Red dashed line corresponds to the signal observed in untreated maize genomic DNA. Solid blue line shows the limit of detection of m6A signal-to-noise for this assay based on the sample that has no m6A injected with a similar amount of total dA. **(B)** Schematic illustrating the calculation of the FIRE accessibility score. Black boxes mark the respective FIRE ACRs (black bars on top). FIRE accessibility scores are shown for the two example ACRs. **(C)** Correlation between FIRE accessibility scores for Fiber-seq replicates 1 and 2. Each dot corresponds to ACRs where both replicates have >10x coverage. **(D)** Correlation of Tn5 insertions for union ACRs identified in ATAC-seq replicates 1 and 2. **(E)** m6A methylation peaked at the centre of ATAC-seq derived ACRs in paired samples. **(F)** m6A and 5mCpG methylation rates surrounding CAGE-defined transcription start sites (TSSs). Average strength of m6A methylation rate upstream of TSSs was monotonically related to expression level of respective downstream genes (expression deciles). **(G)** Methyltransferase-sensitive patches (MSPs) larger than 100bp constituted the majority of the m6A signal at TSSs, while MSPs shorter than 100bp showed patterns consistent with well-positioned nucleosomes. MSP scores were calculated in aggregate for each non-overlapping 20bp window in the region 750bp upstream and 1kb downstream of each TSS. **(H)** Aggregate plot of Tn5 insertions/base in the 1kb window upstream of TSSs stratified by downstream gene expression for paired ATAC-seq data, comparable to (F). **(I)** Aggregate plot of FIRE ACRs stratified into ten deciles based on their FIRE accessibility score. For each FIRE score accessibility decile, the number of Tn5 insertions at each bp within 2kb of the FIRE ACR centre is shown. Accessibility measured by ATAC-seq and Fiber-seq is monotonically correlated. Highly accessible FIRE ACRs tend to show neighbouring FIRE ACRs (symmetric signal at highest decile). This signal is in part due to FIRE ACRs in low-mappability LTR retrotransposons. **(J)** The single-molecule method Fiber-seq outperforms single-cell ATAC-seq as a quantitative measure of chromatin accessibility. 39,132 ACRs were identified as shared FIRE ACRs in dark-grown maize leaves and ATAC ACRs in a pseudobulked leaf sample (GSM4696890) from Marand et al. The percentage of cells containing at least one Tn5 insertion within a shared ACR (% cells accessible) is compared to the percentage of actuated fibers (that is, with a called FIRE element, % actuated Fibers within a given ATAC ACR) underlying the same shared ACR. Each dot represents one shared ACR. Hexbin color reflects the number of dots.

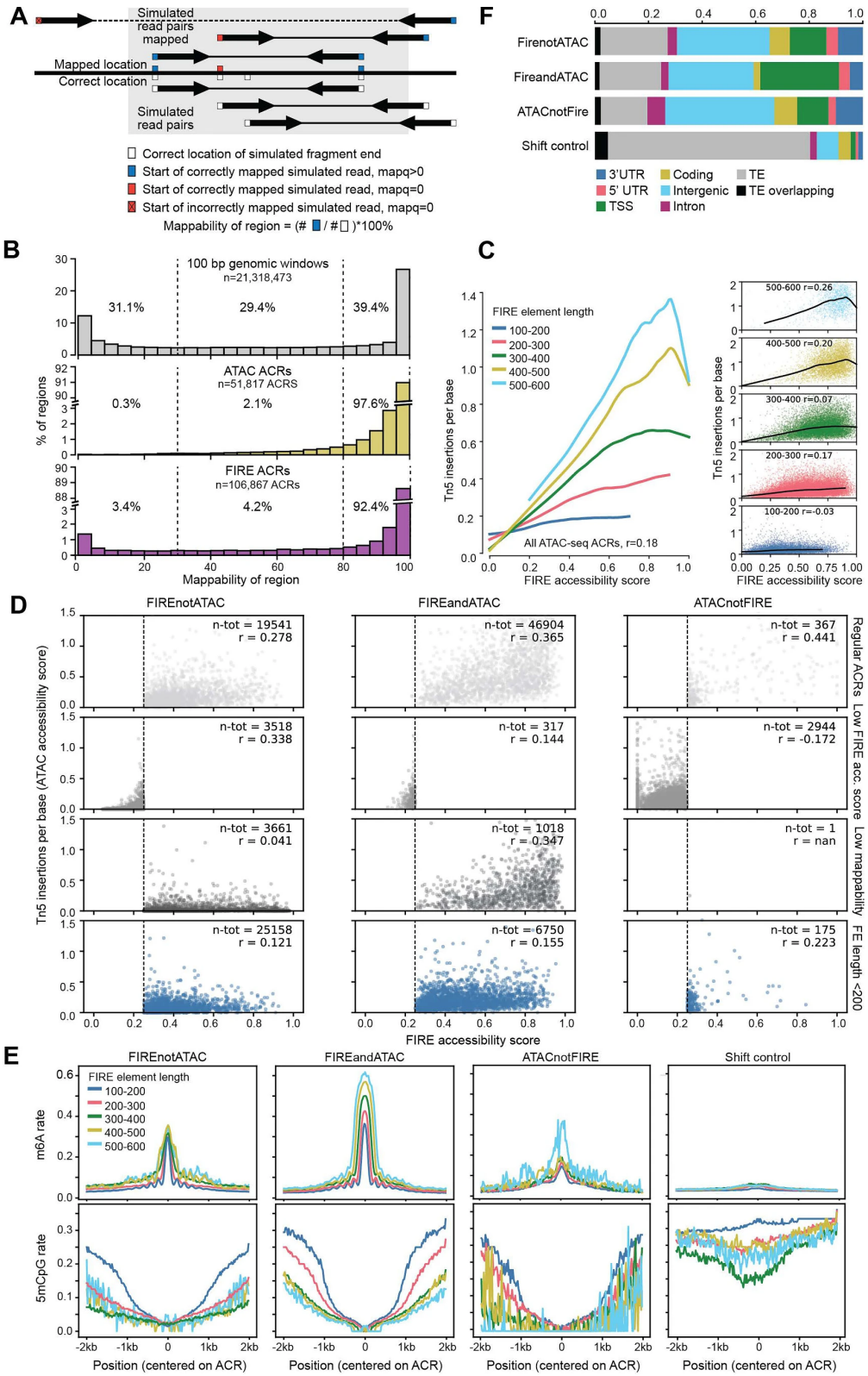

Fig. B.2: (Caption next page.)

Fig. B.2: (Previous page.) **(A)** Schematic describing short-read simulation and mappability calculation. We generated 2.1 billion fragments evenly distributed across the B73 reference genome chromosomes 1-10 (see Methods). For each simulated fragment, 50bp paired-end reads were generated (indicated with thick black arrows). Each read matched exactly the reference sequence from which it was generated. These simulated reads were then mapped back to the genome using BWA. The 'fraction mapped' for a given region or window was calculated as the number of correctly mapped reads with mapq score > 0 divided by the total number of simulated reads with the outer end (Tn5 insertion) falling in the region. Mapq scores are indicated by blue and red boxes, incorrectly mapped simulated read shows X in red box (top row). Mappability of regions was determined as percentage of correctly mapped reads with mapq>0. **(B)** Histograms of mappability as in (A) for all 21,318,473 non-overlapping 100bp windows in the maize genome (top panel, grey), 51,817 ATAC ACRs (middle panel, gold), and 106,867 FIRE ACRs (bottom panel, purple). Low mappability explains only in part why Fiber-seq detects many more ACRs than ATAC-seq. **(C)** FIRE ACRs comprised of short FIRE elements are not detected by ATAC-seq. Correlation between FIRE accessibility scores and Tn5 insertions/ base (chromatin accessibility as measured by ATAC-seq) for FIRE ACRs comprised of FIRE elements of indicated length (see inset for legend). Left, LOWESS curves fitted to FIRE ACRs in respective length categories. Right, plots showing individual values for FIRE ACRs belonging to the five length categories. **(D)** FIRE accessibility score by Tn5 insertions/base (that is, ATAC accessibility score) for ACRs stratified into 12 categories. Each dot represents an ACR with the labelled row and column properties. As the row categories are overlapping, ACRs were sorted hierarchically as follows: all ACRs with low FIRE accessibility score were included in the 'low FIRE acc. score' rows; ACRs with FE length < 200bp and high FIRE accessibility score were included in the 'FE length <200' rows; ACRs with mappability < 80% and both high FIRE accessibility score and FE length >=200bp were included in the 'Unmappable' rows. **(E)** FIRE ACRs that do not overlap with ATAC ACRs show similar patterns of the m6A signal (top) and the 5mCpG signal (bottom) as FIRE ACRs that overlap with ATAC ACRs. Shifted control regions do not display these properties. FIRE element length underlying FIRE ACRs is indicated as in (C). **(F)** FIRE ACRs that do not overlap with ATAC ACRs show a similar distribution across genomic compartments as FIRE ACRs that overlap with ATAC ACRs.

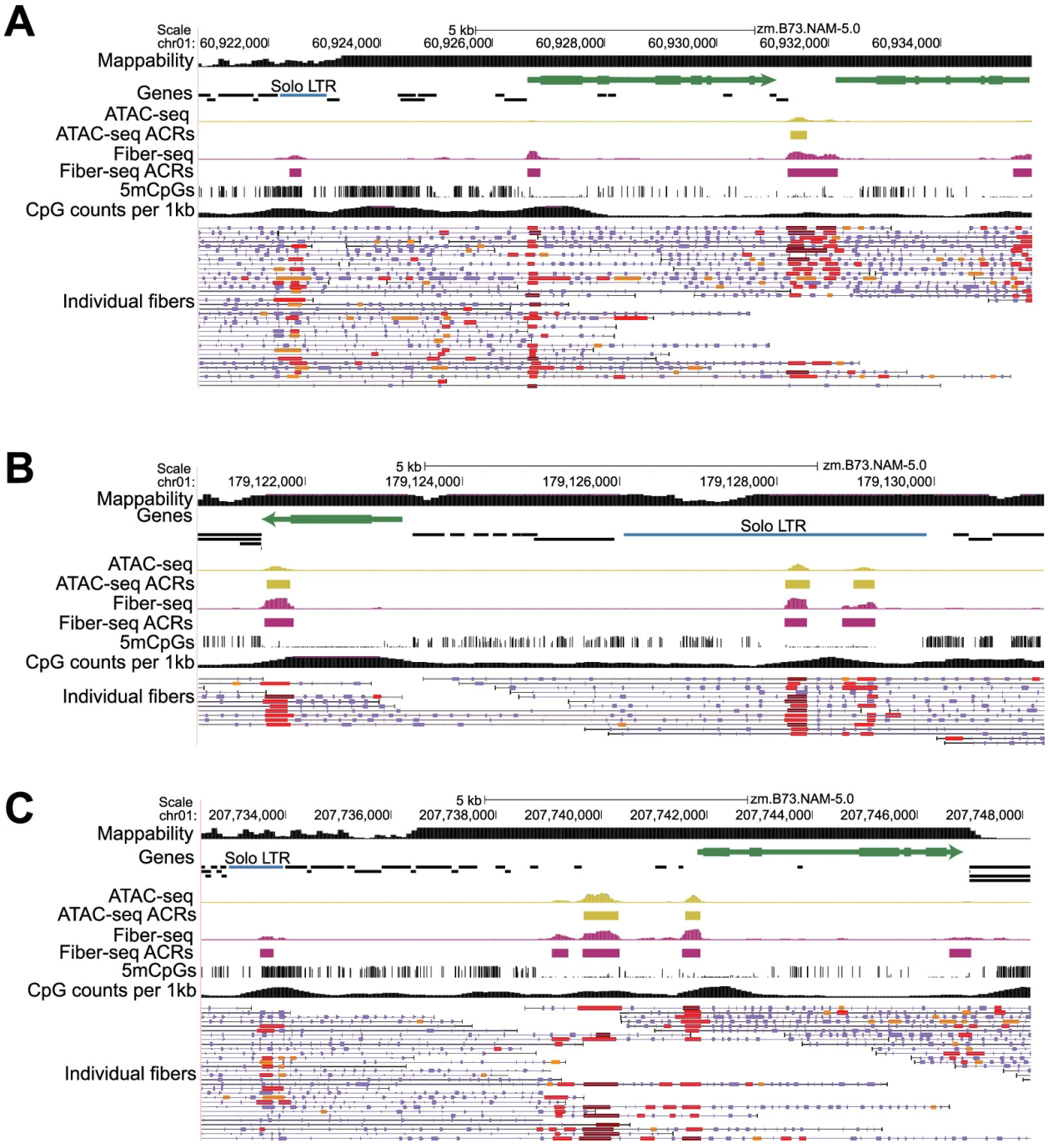

Fig. B.3: **(A-C)** Solo LTRs containing FIRE ACR are colored blue. **(A)** [chr01:60,920,594-60,935,475] **(B)** [chr01:179,120,635-179,131,399] **(C)** [chr01:207,732,409-207,748,141].

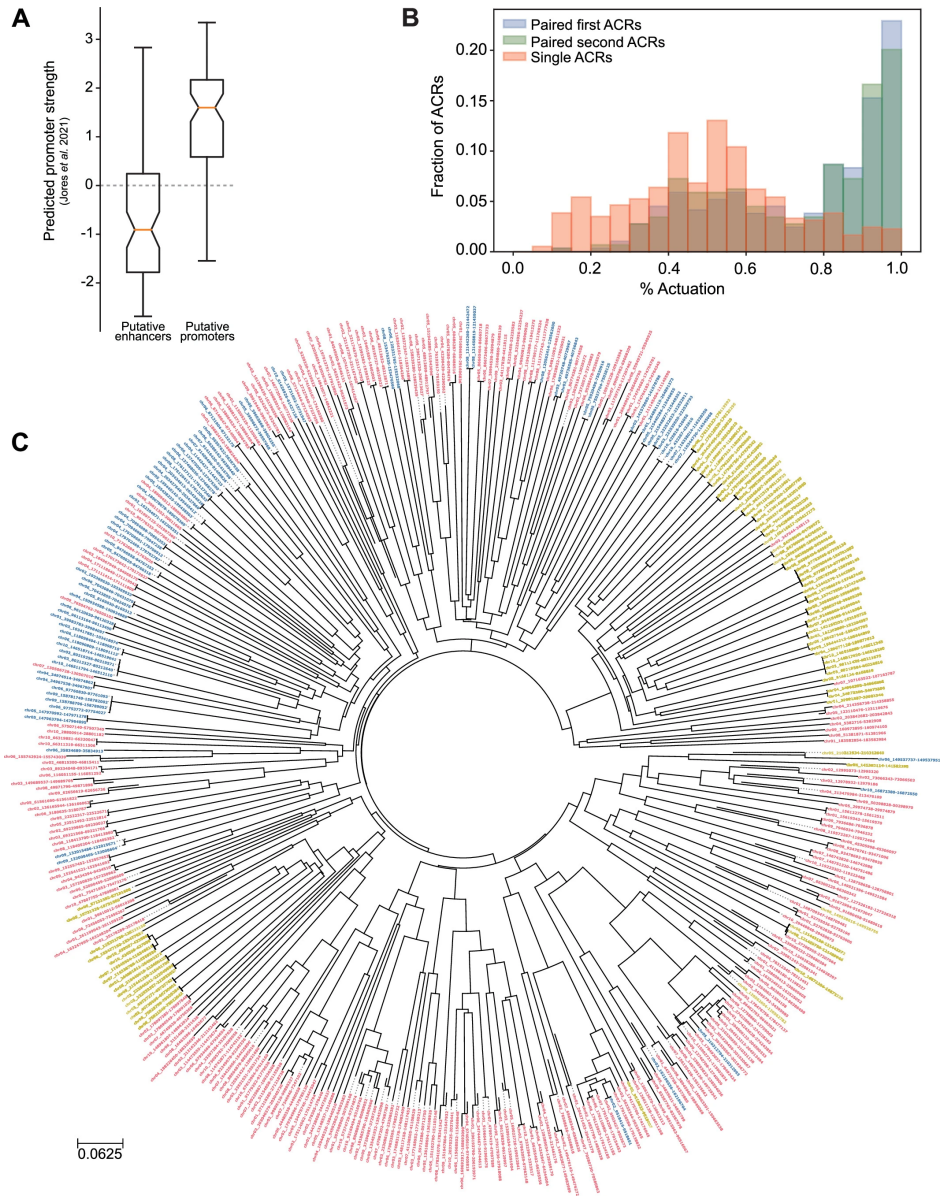

Fig. B.4: **(A)** For each putative enhancer-promoter pair, a sequence starting at the 5' end of the putative enhancer ACR and ending at the 3'end of the putative promoter ACR was extracted. The boxplot shows the predicted promoter strength of the first (coinciding with the putative enhancer) and last 170bp (coinciding with the putative promoter) of this window. Predictions were made with a CNN model trained on Plant-STARR-seq data for 75,000 TSS-proximal ACRs (170bp in length) from Arabidopsis, maize, and sorghum. **(B)** Histograms for the percentage actuation (that is, the percentage of fibers with a FIRE element that comprise a FIRE ACR) for the first of two paired ACRs (putative enhancers), the second of two paired ACRs (putative promoters), and single ACRs. **(C)** Phylogeny of LTR ACRs. Branch length units are in estimated substitutions per site. Colors indicate ACR types with blue denoting paired first ACRs (putative enhancers), yellow denoting paired second ACRs (putative promoters) and red denoting single ACRs in LTRs.

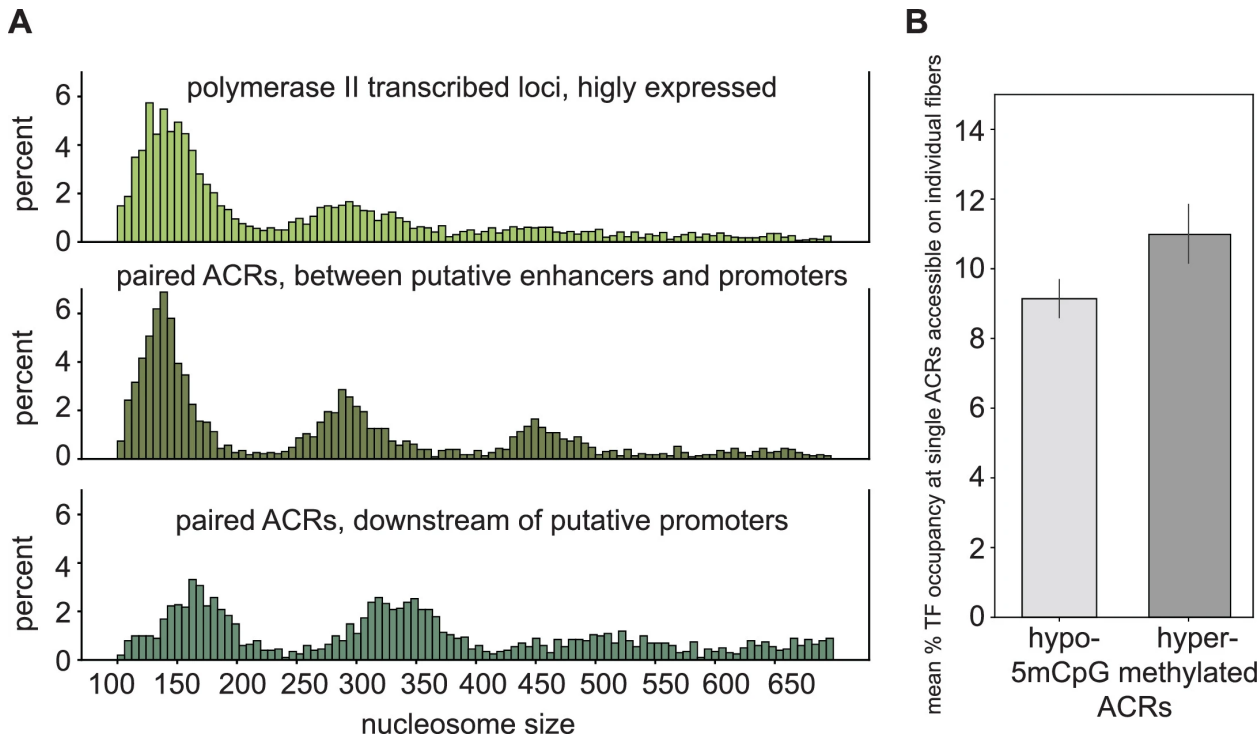

Fig. B.5: **(A)** Histogram showing the percent of footprints sized between 100 and 700bp identified (top) 100 to 1100bp downstream of the TSS of Pol II genes in the top 3 deciles of expression, (middle) between putative enhancer/promoter pairs, and (bottom) 300-1300bp downstream of the putative promoter. **(B)** Bar plot showing the mean percent putative TF (10-40bp) occupancy within +/- 100bp of the center of all hypo- or hyper-methylated single ACRs calculated from reads where the center position of the ACR was not occluded by a nucleosome. Error bars represent the 95th percentile range calculated from 10,000x bootstrapped re-sampling of each group of reads.

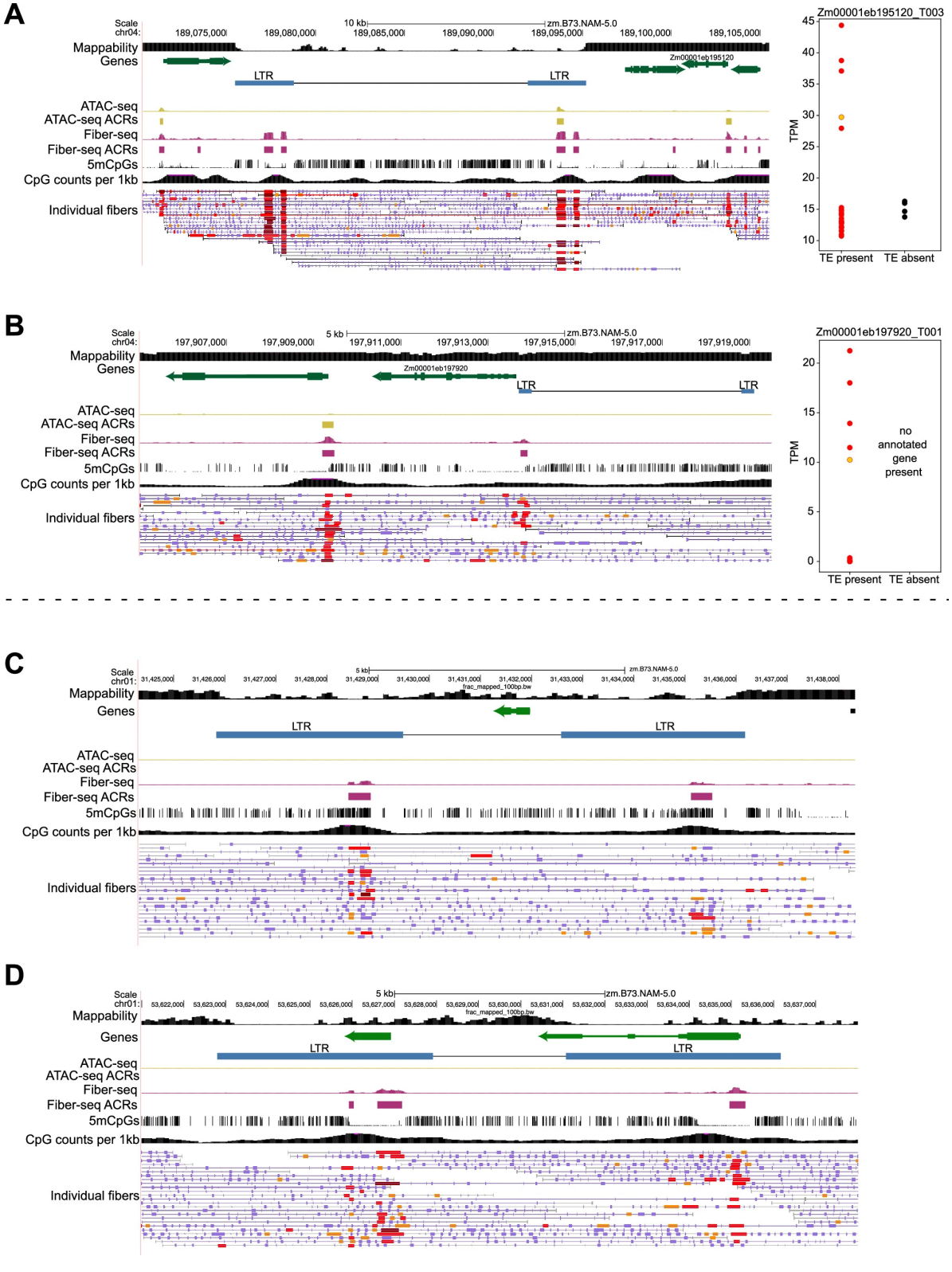

Fig. B.6: (Caption next page.)

Fig. B.6: (Previous page.) **(A)** Left, intact LTR retrotransposon with blue LTRs is absent in NAM lines: Il14H, Ki3, M37W, P39. Tracks in screenshot as in Fig. 3.2. Right, expression level of indicated gene in lines with and without the TE, B73 is labeled in yellow. **(B)** Left, intact retrotransposon with blue LTRs is absent in NAM lines: B97, CML228, CML52, Ki11, Ky21, Mo18W, P39. Tracks in screenshot as in Fig. 3.2. Right, expression level of indicated gene in lines with and without the TE, B73 is labeled in yellow. **(C)** Example of an intact LTR retrotransposon containing one annotated gene between the LTRs and lacking an ACR at the transcription start site. **(D)** Example of an intact LTR retrotransposon containing two annotated genes. For each gene, transcription begins at a FIRE ACR within the LTR.

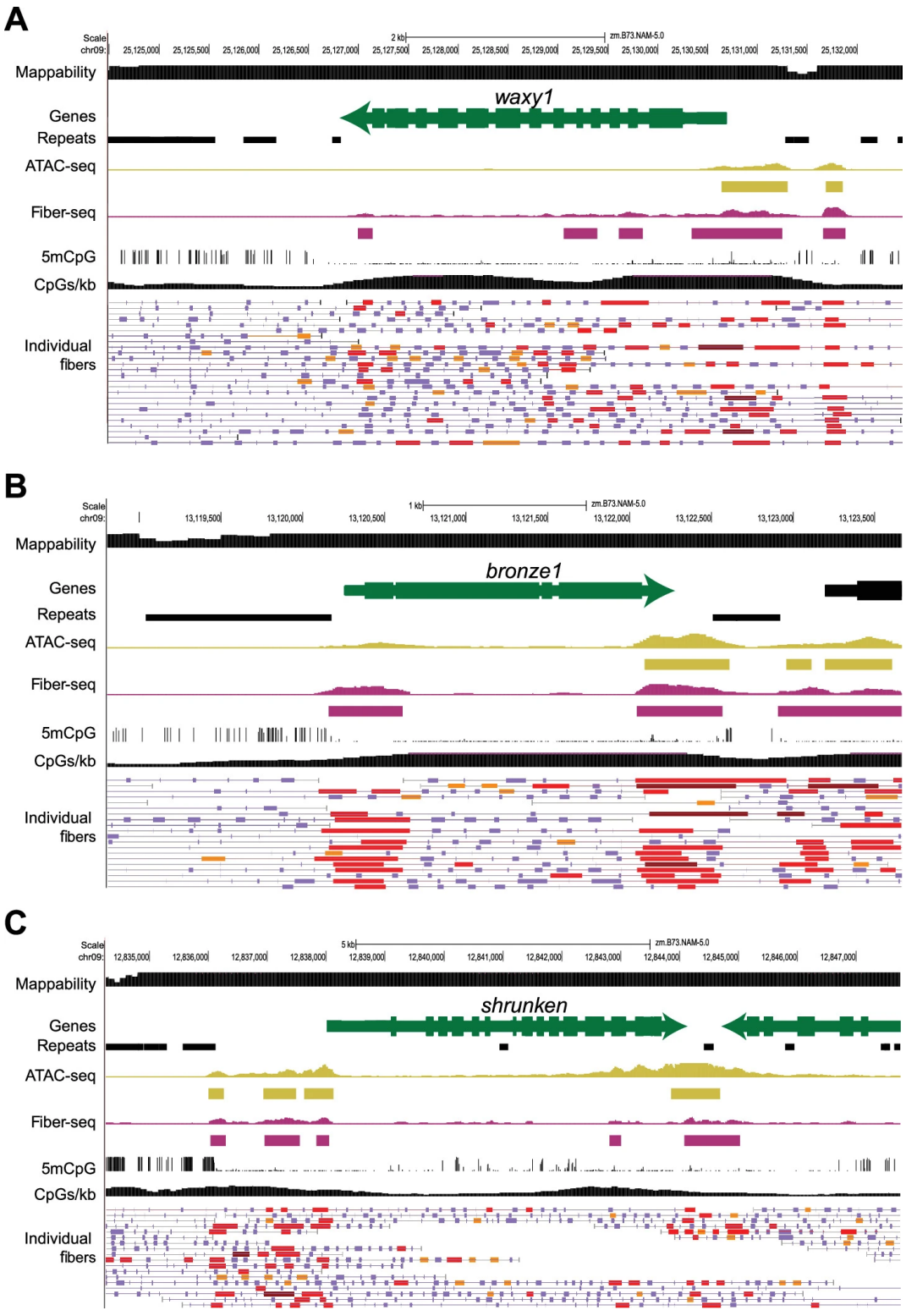

Fig. B.7: (Caption next page.)

Fig. B.7: (Previous page.) **(A)** waxy1 (Zm00001eb378140; chr09:25,127,146 - 25,129,800), one of the first genes identified by McClintock as having a hAT TE insertion, shows higher gene-body chromatin accessibility than 84.4% of other genes. McClintock identified alleles Ds wx-m9, Ds wx-m6, Ac wx-m9, with the Ds or Ac prefix indicating whether it was a nonautonomous or autonomous hAT TE, respectively. **(B)** bronze1 (Zm00001eb374230; chr09:13,118,806-13,123,664), one of the first genes identified by McClintock as having a hAT TE insertion. McClintock identified the Ac bz-m2 allele. The Ac prefix indicates insertion of an autonomous hAT TE. **(C)** shrunken (Zm00001eb374090; chr09:12,836,508-12,845,499), one of the first genes identified by McClintock as having a hAT TE insertion. McClintock identified two germinally-stable alleles, Ds-4864A and Ds-5245, that were "genetically indistinguishable and located just distal to the Shrunken (Sh) locus on the short arm of chromosome 9" and three germinally-unstable alleles, sh-m6233, sh-m5933, sh-m6258, that contain rearrangements at the Sh locus related to a hAT insertion, one of which contains a Ds-mediated 30kb insertion. The Ds prefix indicates insertion of a nonautonomous hAT TE.