Toward comprehensive characterization of chromatin state

Timothy John Durham

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

Doctor of Philosophy

University of Washington

2019

Reading Committee:

William Stafford Noble, Chair

Robert H. Waterston, Chair

Celeste A. Berg

Cole Trapnell

Program Authorized to Offer Degree:
Genome Sciences

University of Washington

Abstract

Toward comprehensive characterization of chromatin state

Timothy John Durham

Co-Chairs of the Supervisory Committee:
Professor William Stafford Noble
Genome Sciences

Professor Robert H. Waterston
Genome Sciences

One of the principal questions in biology is how the genome encodes the information required for producing a multicellular organism. Somehow, the structure of the genome maps to every function in the living organism, from how to assemble the body plan during development to how to react to environmental stimuli or stressors. We know that much of this encoded information is decoded in the cell by gene regulatory networks; sets of transcription factor genes and their repertoire of binding sites throughout the genome that allow them to turn sets of genes on and off. If we could comprehensively map these gene regulatory networks and the settings in which they are active, we would have a deep, mechanistic understanding of how and why cells behave the way they do and how and why mutations in the genome affect phenotype. However, we are still in the very early days of this effort. In order to infer the gene regulatory networks, we first need to understand the "parts list" consisting of every gene and regulatory site, as well as precisely where and when those genes and regulatory sites are active. I present two projects that move the field closer to attaining this comprehensive census of gene expression and regulatory site activity. One is a project to apply single-cell Assay for Transposase-Accessible Chromatin followed by sequencing (scATAC-seq) to generate the first cell type-specific map of chromatin accessibility in *Caenorhabditis elegans*, a promising model organism for comprehensive regulatory network

inference. The other is a machine learning framework for jointly modeling at once thousands of genome-wide experiments from large epigenomics data collections; the model can be used to summarize information from the collection and to impute (i.e. infer computationally) the results of missing experiments. To conclude, I describe the next challenge of mapping the connections among genes and regulatory sites, and one way that emerging single-cell and genome editing technologies might be used to begin attacking this problem at scale.

TABLE OF CONTENTS

Page

LIST OF FIGURES

LIST OF TABLES

Table Number Page

ACKNOWLEDGMENTS

First, I would like to acknowledge my advisers Bill Noble and Bob Waterston for their support and guidance, and for giving me the opportunity to work on two very different projects. I came into graduate school with little experience in either machine learning or bench work, and I am emerging on the other side with substantial experience and expertise in both. I branched out intellectually in many ways over the course of my graduate school career, and I am grateful for their patience and feedback as I made my way. Thank you also to my committee: Celeste Berg and Cole Trapnell, who served on my reading committee, along with Michael Ailion and Elhanan Borenstein for excellent feedback throughout my PhD. And thank you to Jay Shendure, who supported my sciATAC-seq work by sharing reagents, time, and lab expertise.

I was repeatedly helped up some steep learning curves by many generous and friendly people. Thank you especially to Riza Daza, who was a sounding board for debugging ATAC-seq experiments from early on, and who taught me and shepherded me through the sciATAC-seq protocol that is central to Chapter 2 of this dissertation. Also, Chau Huynh, who was my first mentor in the wet lab and taught me all I know about performing experiments on *C. elegans*, and Anh Leith, who helped me with the FACS and went above and beyond on multiple occasions to rescue my experiments. On the computational side, I would like to thank Rob Fatland for his help getting started with cloud computing and for supporting my research on PREDICTD both directly and by advocating for my project with contacts at Amazon and Microsoft. Thanks to the Amazon Web Services Cloud Credits for Research program and the Microsoft Azure4Research program for providing free compute cycles to support my research. And to Max Libbrecht and Jeff Howbert, my Noble Lab-mates and co-authors on PREDICTD, for discussing my work and for patiently teaching me a lot of math and machine learning concepts. Lou Gevirtzman from the Waterston Lab, who

taught me about Bayesian modeling and who dropped everything at the 11^{th} hour to implement updates to the LDA code that greatly improved Chapter 3. Thank you to GS-IT, Brian Giebel, Paul Mantey, and the rest of the GS administrative team for helping over the years in so many ways. And thank you to my GS classmates, and especially my cohort, for making Genome Sciences such a great place to work.

I have been fortunate to have had some extraordinary mentors who also played a role in getting me to this point. I would particularly like to thank Claire Ting, my undergraduate honors thesis advisor, who first introduced me to doing biological research and showed me I could succeed at it; Noam Shoresh and Charles Epstein, my managers and mentors during my time at the Broad Institute before coming to graduate school; and Alon Goren and Melissa Gymrek for being both mentors and friends.

Thanks to my friends Nate Clark and Jane Hu for Ravenna Brewing and backpacking adventures – there's nothing like some peace and quiet in the mountains to clear your mind. Jason Klein for being the most optimistic and can-do person I know (and for many miles of cycling). To Sam Entwisle, my roommate for five years; they say friends are the family you choose... maybe that's why people mistake us for brothers? Cecilia Noecker, for being one of my earliest and best friends in Seattle. Allyson Goldberg for encouragement and patience as I wrote this dissertation, and for reminding me of the most important things in life. And to the Husky Triathlon Club for fun and stress relief these past five years. Go Dawgs!

Last, but not least, I wish to thank my family. My parents Michael and Sharon, who raised me to believe I could do anything I set my mind to. My twin brother Steve, who blazed the PhD trail and showed me the way it's done. And my sister Kelley, and my brother Chris, who have supported me throughout the PhD and lifted me up and inspired me probably more than they realize.

I dedicate this dissertation in memory of my grandparents, who taught me from an early age to approach life with a strong work ethic, open heart, good humor, and an independent spirit. I try to live up to their example every day, and I hope that I make them proud.

Chapter 1: INTRODUCTION

How does our DNA encode the vast amount of information required to make a human being? Since DNA was first discovered as the molecule responsible for genetic inheritance,[1] this question has been one of the key drivers in biology. When the Human Genome Project began in 1990 with the goal of sequencing all 3 billion bases in the human genome, some estimated that the project would discover over 100,000 human genes. By the time the draft sequence was published in 2001,[2] the estimate was around 30,000-40,000 genes, and when the finished sequence was published in 2003[3] the new estimate was even lower: just 20,000-25,000 genes, not much more than the 19,000-20,000 genes found in the comparatively simple model organism, *Caenorhabditis elegans*.[4,5] The gene complement of the genome contained less information than many expected, and it was clear that much of the additional information encoded in the human genome would be found in the other 98% of the sequence.

The 3 Gbp human genome contains almost thirty times as much DNA as the 100 Mbp worm genome. Much of that sequence is likely non-functional DNA that has not been selected out by evolution, but a substantial amount of it is dedicated to regulatory sites that control gene expression. The amount of regulatory sequence exceeds that of protein coding sequence by a substantial margin; estimates vary widely, but while protein coding genes make up about 2% of the human genome, regulatory regions make up about 10%.[6,7] Regulatory regions contain binding sites for regulatory proteins, like transcription factors (TFs) and chromatin regulators, that act to either promote or repress expression of various target genes. Importantly, the genes that a given TF targets include other regulatory genes, and frequently even the TF gene itself. Furthermore, we also know that multiple regulatory sites can impact the regulation of a single gene, and that a single regulatory site can be regulated by multiple genes.[8] The influence of many TFs together leads to a gene regulatory network of interactions that collectively implement a system of robust and precise temporal

and spatial transcriptional control that underlies important biological processes like development and differentiation. Thus, mapping these information-rich regulatory sites in the genome, and gaining a deeper understanding of gene regulatory networks, will yield insight into some of the most important processes in biology.

1.1 CHARACTERIZING GENE REGULATORY NETWORKS

Since at least the 1960s, when Jacob and Monod dissected the regulatory mechanisms of the *lac* operon in *E. coli*,[9] biologists have been striving to understand the way that genes are regulated, both in response to environmental stimuli and in the process of development. Five key pieces of information are most informative for elucidating and validating gene regulatory networks: 1) the genes involved, 2) the regulatory sites involved, 3) the effect of perturbing a transcription factor gene on target gene expression, 4) the effect of perturbing nearby regulatory sites on expression of the target gene, and 5) the effect of perturbing nearby regulatory sites on the binding of the transcription factor.[10,11] So far, relatively few gene regulatory networks have been reconstructed in this level of detail. One of the best examples is the regulatory network underlying endomesoderm development in the purple sea urchin.[12] Despite using bulk samples of cells that may mask some cell type-specific regulatory patterns and using qPCR for selected target genes instead of the unbiased, whole genome approaches like RNA-seq that we have available to us today, this work convincingly showed that development proceeds via a flow of information through a network of interactions among signaling molecules, transcription factors, and regulatory sites in the genome that collectively implement logical circuits. As with the circuits found in today's computers and other electronics, these networks are highly modular; and this modularity implies that the evolution of regulatory DNA, though it proceeds through seemingly minor changes in DNA sequence, can have major effects at the level of developmental processes by changing the spatial and temporal activation of these network modules.[13]

1.2 ADVANCES IN FUNCTIONAL GENOMICS

The implications of accurately mapping these gene regulatory networks are profound and would lead to a deeper understanding of the structure and function of the genome that could be applied to better understand and treat human disease. We are still in the early stages of achieving this goal. In the case of the purple sea urchin, it took about a decade of careful work for researchers to decode the gene regulatory networks underlying the development of a subset of early embryonic tissues;[14] it remains a grand challenge to collect all five pieces of information and fully elucidate the regulatory networks underlying the development of more complex multicellular organisms. Nevertheless, since the publication of the human genome sequence[2] there has been immense progress toward meeting the first two information requirements by collecting the "parts list" of gene networks. Broad, collaborative efforts working on human, mouse, worm, and fly are identifying genes,[15] recording their expression patterns across tissues and developmental time points,[6,7,16–18] and mapping the locations of regulatory DNA and the chromatin states that correlate with gene expression patterns.[6,7,16–19]

Massive amounts of data have been collected, and the analysis of these data have led to new insights into many facets of chromatin biology, including tissue-specific gene regulation,[20] modules of chromatin modifying enzymes,[21] and maps of distinct types of chromatin states in different cell types.[22–24] However, despite these advances, the data from the thousands of experiments that have been collected so far have some important limitations. First, the vast majority of the experiments have been done on bulk populations of cells, either from cell lines[6] or from sections of primary tissue.[7] Bulk samples of primary tissue are problematic because they are generally composed of heterogeneous cell types, and the signal for a bulk experiment is a population average that does not necessarily reflect the signal found in any particular cell. Thus, from the perspective of regulatory network inference, these experiments can provide the "parts lists" of regulatory sites and genes that are expressed in a given tissue at a particular time, but do not tell us which are active in the same specific cells or cell types. The relevant cell type for a given gene or regulatory element is even less certain for the data collected from whole flies and worms, organisms for which dissecting out

all tissues is infeasible at the scale required for projects like the model organism Encyclopedia of DNA Elements (modENCODE)[16,17] and model organism Encyclopedia of Regulatory Networks (modERN).[19] Second, each bulk assay is time intensive and expensive to perform, and it is impractical to execute such experiments in all cell types and all developmental stages. The limitations of bulk assays are especially relevant for measuring the genome-wide binding locations of proteins such as transcription factors, chromatin regulators, and post-translationally modified histones. The gold standard assay for protein-DNA binding *in vivo* is chromatin immunoprecipitation followed by sequencing (ChIP-seq), which involves fixing cells with formaldehyde to preserve protein-DNA interactions, lysing the cells and fragmenting the chromatin, using antibodies with affinity for specific proteins (or affinity for a tag engineered onto those proteins) to pull down the protein and any crosslinked DNA fragments, and finally sequencing those fragments. ChIP-seq is challenging for multiple reasons. The first is its reliance on the antibody. A good antibody for ChIP must have high specificity and sensitivity for the target protein; this can be very difficult for certain proteins, and the quality of antibodies can vary dramatically batch to batch just from the variation introduced by raising the antibodies in different animals with different immune systems. Even when a single antibody can be used to detect a variety of tagged proteins, a lot of work must be invested to engineer a different strain of lab animal for each tagged protein (and this is impossible to do in some organisms, like humans). Furthermore, in order to use ChIP-seq to gain a comprehensive view of chromatin state, a separate bulk experiment must be executed for every protein of interest in every cell type of interest. As a result of these challenges, in human and mouse, which have more than 1000 TFs in addition to many chromatin regulators and histone modifications, relatively few proteins have been assayed by ChIP-seq in more than a few tissues or cell types. Efforts to map TF binding in fly and worm[16,19] have mapped a greater percentage of the repertoire of hundreds of TFs in these organisms, but in most cases have done so for only one or a handful of developmental stages, and with no cell type resolution.

In order to be able to map gene regulatory networks and to better understand how they give rise to various developmental or phenotypic outcomes, we need ways to finish compiling the full "parts lists" by more specifically and comprehensively measuring gene expression and chromatin state at

all developmental stages and in all cell types. Such a full accounting of genes and regulatory sites will undoubtedly come from developing both better biological assays for measuring gene expression and chromatin state, and more advanced computational approaches capable of integrating and analyzing the complex data that will be generated.

One of the most important advances in biology in the past decade is the advent and maturation of technologies for single-cell genomics.[25] The first application of a functional genomics technology to single cells was a study that used mRNA-seq to measure transcript and isoform abundance in single mouse oocytes and blastomeres.[26] Over the ensuing years, a variety of new technologies were introduced that increased sensitivity and throughput, making it possible to measure the transcriptomes of thousands, and even millions, of cells in a single experiment. The key step in these approaches is in tagging the data from individual cells in a high-throughput fashion; there are two main strategies to accomplish this goal. One strategy is to encapsulate the cell along with uniquely-barcoded reverse transcription primers inside water droplets in oil,[27–29] and executing the reverse transcription in nanoliter-scale reaction volumes. The other strategy is to take a multi-step approach to indexing the cells, by sorting them into wells, doing the reverse transcription reaction with well-specific barcodes, and then pooling the cells and re-sorting into new wells for amplification of the cDNA libraries, and integration of a second barcode, by PCR.[30,31] In this way, cells are never individually isolated, but instead are probabilistically assigned unique pairs of barcodes due to the rounds of random assortment into wells; the data coming from single cells are inferred after the experiment by computationally segregating sequencing reads based on their combination of barcodes. single-cell RNA-seq has proven to be a powerful way to investigate complex mixtures of cells, including immune cell types,[29] tumor cell heterogeneity,[32] whole *Caenorhabditis elegans* nematodes,[30,33] developing mouse embryos,[31] and *in vitro* tissue differentiation.[34]

Other promising single-cell technologies measure different facets of the epigenome, and are beginning to produce high resolution maps of the regulatory DNA elements that are active in different cell types. Chief among these technologies is the single-cell variant of the Assay for Transposase-Accessible Chromatin followed by sequencing (scATAC-seq),[35,36] which measures chromatin accessibility. Active regulatory regions in the genome must be accessible to transcription factors,

other regulatory proteins, and the transcriptional machinery in order to fulfill their function, and as a result, they are less tightly packed around nucleosomes that would otherwise block other proteins from binding to the DNA. Thus, one way to map regulatory regions in an unbiased manner is to measure the accessibility of the DNA to cutting enzymes. Traditionally, chromatin accessibility assays have used DNase (e.g. DNase-seq[37]), but due to complexities in the protocol, DNase-based assays are difficult to adapt to the extremely low amounts of input DNA involved in single-cell assays. In contrast, ATAC-seq uses a hyperactive Tn5 transposome to cut the DNA at accessible sites and ligate DNA sequencing adapters onto either side of the cut, all in a single reaction. The only step required before sequencing is PCR amplification, and the lack of liquid handling steps allows ATAC-seq to work on very low input amounts. As with single-cell RNA-seq, scATAC-seq is compatible both with methods that individually isolate cells,[38] and that use a combinatorial indexing strategy.[39–41] The data from scATAC-seq are very sparse; a diploid genome provides only two chances for an accessible site to be measured by a Tn5 insertion, and this limits the sensitivity of the assay for any particular single cell. Also, many open sites are not cut at all. Nevertheless, by pooling the signal from similar cells studies have been able to gain insights into cell type-specific gene regulatory sites, including finding 85 different chromatin accessibility patterns from 13 different mouse tissues,[41] mapping regulatory sites that are specific to the development of different germ layers in *Drosophila* embryogenesis,[40] investigating different cell populations in the mouse hippocampus,[42] and inferring connections between regulatory elements and target genes during *in vitro* myoblast differentiation.[8] As scATAC-seq becomes more routine, we will increasingly have paired chromatin accessibility and RNA-seq data for the same cell populations, and even the same cells.[43] These paired data types at single-cell resolution will begin to provide the kind of comprehensive gene regulatory network "parts list" that is needed to understand and model gene regulation on a fine scale.

1.3 Computational Advances

Modern high throughput sequencing-based assays, particularly scRNA-seq and scATAC-seq, are rapidly expanding our view of the cell type-specific transcriptional and regulatory landscape of the genome. Accompanying the advances in data collection afforded by single-cell genomics are advances in computational techniques. In particular, methods for integrating different data types are key. ChIP-seq, ATAC-seq, and RNA-seq data all represent different "views" of the same cell states; in order to achieve a holistic view of genome biology, including inferring gene regulatory networks, we must be able to leverage information from all of these views at once. Many successful efforts have taken steps in this direction. A few of these include using probabilistic models like hidden Markov models[22] or dynamic Bayesian networks[23] to segment the genome into sections that show similar patterns across data types.[24] Different types of segments can be correlated with different known genomic elements, like transcribed gene bodies, promoters, distal enhancers, and heterochromatic regions; and annotating the genome in this way can provide cell type-specific lists of candidate regulatory elements and generate testable hypotheses about the effects of mutations associated with disease.[24] Another method tries to learn patterns from a variety of types of existing data in order to impute the results of those assays in under-characterized cell types.[44] Still other methods strive to apply advances in machine learning, especially deep neural networks, to challenges like interpreting the effects of genetic variants in non-coding DNA.[45,46]

The computational methods discussed so far were all designed for bulk epigenomics data sets, however single-cell experiments present additional challenges and opportunities for computational analysis. single-cell functional genomics data are much more sparse, with many missing values in each data set, and because each cell is an individual sample, analyses of single-cell data also have to contend with higher dimensionality. Many of the tools developed for single-cell analysis are concerned with grouping or ordering cells based on the data. Some tools, like Uniform Manifold Approximation and Projection (UMAP)[47] and t-Stochastic Neighbor Embedding (t-SNE),[48] are general dimensionality reduction techniques that are helpful for visualization of the relationships among cells. Others, like Louvain-style community detection,[49] are effective at clustering the cells

based on density, which is valuable for detecting irregularly shaped groups of cells that might not be effectively captured by more traditional clustering approaches, like k-means clustering, that make assumptions about the distribution of the data within clusters. Methods like Monocle[34,50,51] and Seurat[52,53] build on these basic algorithms and implement sophisticated modeling approaches with appropriate constraints that help to find more biologically plausible solutions. Monocle is especially well-suited to analyzing single-cell data collected from a cell population that is differentiating or otherwise undergoing a change over time, as it can order these cells along a "pseudotime" trajectory, and even find tree structures in the data where cell fates diverge with time.

As mentioned above, a major challenge posed by single-cell data that must be handled computationally is sparsity. Due to the limited efficiency of molecular biological assays and the extremely limited amount of DNA or RNA in single cells, it is impossible to expect that every transcript (for scRNA-seq) or every accessible region (for scATAC-seq) will be measured. Therefore, if a gene or accessible site is missing from a particular cell, it is difficult to decide whether that measurement is missing because it was not present in that cell or because it was missed in the assay. Two of the most effective models for handling this sparsity are latent semantic indexing (LSI)[40,41] and latent Dirichlet allocation (LDA).[54] Both approaches were developed in the machine learning field of natural language processing.[55,56] They model documents as vectors of word counts, and for a given document corpus represented as a documents-by-words matrix they attempt to find a low-rank representation of the matrix that can distinguish among informative words and among documents. Intuitively, these models can be thought of as identifying the underlying topics that characterize these documents and the usage of words within them. LSI leverages singular value decomposition to find a certain number of latent dimensions that capture variation in the words and documents, while LDA takes a Bayesian approach and models each document as a probability distribution over some number of latent topics and each topic as a probability distribution over the words. After modeling, even though the input data are sparse (i.e. a given document will not necessarily contain all words that can be relevant to its main topic) the resulting low-rank factors have no missing data and can be used for further analysis. In the case of single-cell genomics data sets, cells are treated as documents, and genes (for scRNA-seq) or accessible regions (for scATAC-seq)

are treated as words. Although LSI and LDA are perhaps the most natural and effective solutions for accounting for sparsity in single-cell data, still other methods have been developed for imputing missing values,[57] and for merging results across experiments.[53]

1.4 Organization of this dissertation

With ever increasing volumes of functional genomics data measuring the state of a genome across more and more cell states and tissues, and continually improving computational methods for integrating these results, the field is getting closer to the goal of comprehensively characterizing genome state in all cell types and at all developmental time points. The genes and regulatory sites found to be relevant at various cell states represent nodes in the gene regulatory network, and further experiments will begin to perturb the genes and regulatory sites to identify interactions and fill in the network edges. Here I present two studies that bring the field closer to identifying the network nodes.

In Chapter 2 I describe work collecting single-cell ATAC-seq data for all tissues in the second larval stage (L2) of *Caenorhabditis elegans* nematodes. *C. elegans* is a premier experimental system for studying development: its lineage of cell divisions that give rise to the adult worm has been completely characterized;[58,59] it has a high fidelity genome sequence with extensively curated gene annotations;[5,60] it has a short generation time and is amenable to genetic manipulation; and it has a limited number of cells that make the worm a tractable system for truly comprehensive mapping of transcriptomic and chromatin states. Recently, single-cell RNA-seq atlases were published for L2 larvae[30] and embryonic development.[33] With this in mind, I decided to collect scATAC-seq data in L2 larvae to match the gene expression data. In addition, the L2 stage is a good starting point because it has fully differentiated cells that should show diverse patterns of chromatin accessibility, but it is not so far along in development that germline cells have begun to proliferate and outnumber the somatic cells. Indeed, I show that unsupervised clustering of the scATAC-seq data yields 26 groups of cells that are consistent with tissue types identified through scRNA-seq. Together, the scATAC-seq and scRNA-seq data sets provide a high resolution view of the state of the genome in

different *C. elegans* tissues, and lay the groundwork for both characterization of additional developmental stages and identification of interactions between genes and regulatory sites that will be critical for mapping the gene regulatory networks controlling cell state in the larval worm.

Chapter 3 details a project, called PaRallel Epigenomics Data Imputation with Cloud-based Tensor Decomposition (PREDICTD), to impute missing data in the human NIH Roadmap Epigenomics Project[7] and Encyclopedia of DNA Elements (ENCODE).[6] This project is similar in aim to that of another program, ChromImpute,[44] but takes a different approach. Specifically, while ChromImpute trains a model for each missing data set individually, PREDICTD trains a single joint model on all data at once. It accomplishes this integration by modeling the data sets as a three dimensional tensor, with one dimension corresponding to cell types, another to assays (e.g. DNase-seq, and various histone modifications targeted by ChIP-seq), and the third to the genome (binned at 25 bp resolution). PREDICTD imputes the missing data in this tensor by using a machine learning approach called PARAFAC[61] to factor the tensor into three low-rank matrices that summarize the data. I show results suggesting that the model parameters effectively capture biological information that may be useful as summary features in downstream machine learning applications, that by recombining the factors PREDICTD imputes missing data in the tensor that is comparable to the state of the art ChromImpute imputed data, and that the imputed data can be used to investigate tissue-specific regulatory sites in under-characterized cell types.

In Chapter 4, I discuss these results and provide some final thoughts on future directions and how the field is progressing toward a full characterization of the gene regulatory networks implemented by the genome.

Chapter 2: COMPREHENSIVE CHARACTERIZATION OF TISSUE-SPECIFIC CHROMATIN ACCESSIBILITY IN L2 *CAENORHABDITIS ELEGANS* NEMATODES

This work will be submitted for publication with the following author list: Timothy J. Durham, Riza M. Daza, Louis Gevirtzman, Darren A. Cusanovich, William S. Noble, Jay Shendure, and Robert H. Waterston.

2.1 AUTHOR CONTRIBUTIONS

TJD and RHW conceived of the experiment. TJD grew the worms and isolated the nuclei. TJD and RMD performed the sciATAC-seq experiments. DAC helped perform the pilot sciATAC-seq experiment, and shared his code for the initial processing and QC of the sequencing data. JS provided important reagents and lab equipment for the sciATAC-seq protocol. TJD analyzed the data and wrote the paper with input from RHW and WSN. LG wrote the parallelized LDA Java program. RHW, LG, and WSN contributed to discussions of the data analysis.

2.2 ABSTRACT

Recently developed single-cell technologies are allowing researchers to characterize cell states at ever greater resolution and scale. *C. elegans* is a particularly tractable system for studying development, and a recent single-cell RNA-seq study characterized the gene expression patterns for nearly every cell in the second larval stage (L2). Gene expression patterns are useful for learning about gene function and give insight into the biochemical state of different cell types; however, in order to understand these cell types, we must also determine how these gene expression levels are regulated. We present the first single-cell ATAC-seq study in *C. elegans*. We collected data in L2 larvae to match the available single-cell RNA-seq data set, and show that we identify tissue-specific chromatin accessibility patterns that align well with existing data, including the L2 single-cell

RNA-seq results. Our chromatin accessibility data provide novel insight into which genomic loci may be participating in cell type-specific gene regulation, with promise for better understanding cellular differentiation in the worm.

2.3 INTRODUCTION

Nearly all critical cellular processes are dependent on fine-tuned control of gene expression levels. From properly responding to environmental stimuli, to progressing through the stages of development and differentiation, specific changes in gene expression play an important role in facilitating precise changes in cellular state. Recent advances in single-cell transcriptomics have enabled the massively-parallel measurement of gene expression, giving unprecedented genome-wide insight into which genes are regulated together in the same cell, and into their dynamics over time. Cataloging which genes are important in which cells under which conditions is critical to a deeper understanding of their function; however, this enumeration is only part of the story. In order to truly understand how gene expression reflects and influences cell state, we must also understand how it is controlled. The nematode *Caenorhabditis elegans* is a particularly powerful system in which to apply single-cell genomics technologies because it has limited cell numbers that nonetheless form diverse tissue types; it is very amenable to genetic manipulation; and the developmental lineage of every cell is known and invariant. In 2017, the most comprehensive cell type-specific gene expression data set of a metazoan was published using single-cell combinatorial indexing RNA-seq (sci-RNA-seq).[30] This study of the *C. elegans* second larval stage provided a view into the full complement of genes expressed in each major cell type, and even some cells present only once in the worm (e.g. the ASEL and ASER gustatory neurons). Now, in order to understand how these tissue-specific expression patterns arise, we also need to have a similarly comprehensive catalog of tissue type-resolved regulatory elements.

Several efforts have been undertaken to map regulatory DNA in the worm.[19,62–65] Collectively, these studies have identified tens of thousands of chromatin accessibility regions and transcription factor binding sites, using DNase-seq,[63] ATAC-seq,[64,65] and ChIP-seq[19,62] to assay developmental

stages throughout the worm life cycle. The results convincingly show that the activity at many regulatory sites changes dramatically over the worm's lifespan. However, the data from all of these studies is from whole worms, and thus does not resolve differences in regulatory activity across cell types. This lack of cell-type resolution is problematic for three main reasons. First, gene regulation is often highly cell type-specific,[66] and even when different cell types express the same gene, they may use different enhancers or promoters to regulate that gene.[20,67] In such a case of two sites regulating the same gene in different cell types, a whole worm chromatin accessibility data set would only show that both sites are accessible at the same time, and it would be unknown whether they act in concert in the same cell type or if they affect the same gene but act in different cell types. Distinguishing between these cases is critical for understanding and modeling gene regulation. The second reason is that whole worm data lack the sensitivity to detect regulatory events that occur in cell types that make up small fractions of the whole worm. We know from single-cell RNA-seq[30] that there are important differences in gene expression that distinguish even individual cells from the rest. Such differences are presumably driven in part by regulatory regions that are only accessible in those cells; in a whole worm assay the signal from these highly cell type-specific regions would be drowned out by the noise generated from more populous cell types. And third, the lack of cell type resolution on these regulatory DNA maps confounds our ability to draw conclusions about differential activity across development. During development, the number of cells, and with them the diversity and proportion of cell types, is constantly changing. Thus, if an accessible site is less prominent in a later larval stage compared to an embryonic stage, this change could mean the site is more important in embryogenesis than in later development, or it could reflect that the site is more specialized in later stages and is accessible in a smaller fraction of the cells. Given these important limitations in the available data on *C. elegans* gene regulation, we sought to generate cell type-resolved chromatin accessibility maps.

Over the past few years, the technology to collect chromatin accessibility profiles of single cells has improved greatly. This technology relies on the assay for transposase-accessible chromatin followed by high throughput sequencing (ATAC-seq),[35] which treats permeabilized nuclei with a hyperactive Tn5 transposome from prokaryotes[68] to simultaneously cut accessible sites in the

genome and ligate sequencing adapters onto the fragment ends on either side of the cut site (a reaction referred to as "tagmentation"). The resulting library is then amplified and sequenced. The simplicity of the assay significantly reduces the requirements for input material compared to DNase-seq,[69] and protocols have adapted ATAC-seq to work on single cells.[38,39,70] These and other studies have shown in multiple systems that single-cell ATAC-seq (scATAC-seq) can measure thousands of sites per cell type and can identify distinct cell populations with high sensitivity. single-cell chromatin accessibility measurements have been leveraged to identify differences in gene regulation across different germ layers in *Drosophila* embryogenesis;[40] to generate an atlas of 85 different clusters of cells from 13 different mouse tissues;[41] to identify fine-grained immune cell types from samples of mouse splenocytes;[70] and to identify cell types in hippocampal tissue from mice.[42] We were eager to apply this powerful chromatin profiling technology to *C. elegans*.

Here we present the most comprehensive cell type-resolved map of the regulatory DNA in a whole metazoan organism. We collected sci-ATAC-seq data from 31,611 nuclei (hereafter referred to as cells for simplicity) isolated from a synchronized population of second larval stage (L2) *Caenorhabditis elegans* nematodes. We found 26 clusters of cells that represent distinct tissue types based on mapping 74,067 peak summits to their nearest downstream genes and assessing the tissue-specific expression of those genes in L2 sciRNA-seq data.[30] To contend with low library complexity, we use a Latent Dirichlet Allocation (LDA) model[56,71] that is similar to cisTopic.[54] We report maps of chromatin accessibility summits that validate many previously reported regulatory sites, and we also annotate almost 30,000 additional novel candidate regulatory sites, most of which are accessible only in a subset of cell types. We anticipate that these data will provide a valuable resource for studying regulatory biology in the worm, and future single-cell ATAC-seq experiments on additional life stages, in conjunction with cell type-specific gene expression data, will begin to reveal the gene regulatory networks driving development in *C. elegans*.

2.4 RESULTS

2.4.1 Single-cell chromatin accessibility in C. elegans with sci-ATAC

In order to match the sci-RNA-seq data, we grew wild-type VC2010 worms to the middle of the second larval (L2) stage. At this stage, the majority of the 959 cells in the adult hermaphrodite have been produced and are terminally differentiated, but the development of the gonad has not progressed far enough to begin producing the thousands of germline nuclei that would eventually outnumber the somatic nuclei in later stages and bias our collection of tissue types. After harvesting the worms, we fixed and isolated the nuclei, froze them in aliquots, and used these wild-type nuclei as input to the single-cell combinatorial indexing assay for transposase-accessible chromatin (sciATAC-seq).[35, 39–41] This assay probabilistically identifies DNA fragments isolated from single cells by first sorting 2500 nuclei per well into a 96 well plate and treating the nuclei in each well with a Tn5 enzyme loaded with uniquely-barcoded adapters, and then pooling and re-sorting 25 nuclei per well into new 96-well plates in which a second set of barcodes are incorporated by using well-specific primers during library amplification. After sequencing, the reads can be assigned to a particular cell based on their combination of Tn5 and PCR barcodes (Fig. A.1). We collected sciATAC-seq data for 31,611 cells with at least 150 unique reads (Fig. A.2), which represents about 40x sampling of each cell in the L2 worm.

The post-sequencing pipeline consists of identifying cells and regulatory regions, and recording which regulatory regions are detected in which cells. We first trim adapter sequences from the reads, fix sequencing errors in the barcodes using a custom script,[41] align the trimmed, paired-end reads to the WS230/ce10 draft of the *C. elegans* genome, and then remove duplicate reads. In order to identify the Tn5 cutting sites from the data, we convert the aligned paired-end reads to coordinates identifying regions +/- 30 base pairs around the outer ends of the sequenced DNA fragment, and adjust the forward strand reads by +4 bp and the negative strand reads by -5 bp to account for the shape of the Tn5 cut site.[35] We then use these cut site regions as the input to peak calling as if they were single-end 60 bp reads.

There exists no unbiased annotation of cell type-resolved regulatory regions in *C. elegans*,

so we called them from the sciATAC-seq data itself. [A note on terminology: throughout this work, when referring to genomic loci enriched for chromatin accessibility, we make a distinction between "peaks" and "peak summits" (or just "summits" for short). We use the term "peaks" to refer to regions with enriched signal, usually as called by the MACS2 signal processing software,[72] while the term "summits" refers to a smaller window of about 100 bp centered on one or more local maxima within each peak region. Peaks can be thought of as regions with statistically significant chromatin accessibility, while the summits represent the locations of maximum accessibility within the peak regions.] The first step in single-cell ATAC-seq analysis is commonly to call peaks on all of the reads, as for a bulk data set; however, we found that this method resulted in calling only the strongest peaks and that the smaller, more cell type-specific peaks were drowned out in the complex mixture of cell types in the whole worm. Thus, in order to gain more sensitivity, we proceeded to call peaks in two main steps (for details, see Methods): first, we estimated peak locations by setting a total coverage threshold across the genome and used these regions to cluster the cells (Fig. A.3). Second, for each cluster, we pooled the cut site "reads" associated with that cluster, and called peaks and peak summits using MACS2[72] with a q-value threshold of 0.05 and an input control made by treating naked *C. elegans* genomic DNA with Tn5. In order to include accessibility signal in the immediate vicinity of peak summits, we expanded the regions +/- 50 bp around the summit coordinates. To make master peak and summit lists, we merged the peaks from all clusters such that any overlapping peaks were combined into a single region, and did the same for the summit regions; we identified a total of 32,486 peaks and 74,067 summits. Last, for further analysis we made a binary matrix indicating which of the merged summit regions were accessible in which cells (Fig. A.1).

2.4.2 Peak summits from sciATAC-seq validate many regulatory regions from published bulk chromatin assays

After identifying the 74,067 peak summit regions, we compared them to other maps of regulatory DNA in *C. elegans*. We intersected the summit regions with peaks from two other data sets: bulk,

whole worm ATAC-seq data from across the *C. elegans* lifespan,[65] and transcription factor (TF) binding site peaks identified from 427 TF whole-worm ChIP-seq data sets from the modERN consortium[19] (Fig. 2.1). We expected to find many overlapping sites, but there are also important differences between our data and the others. First, whole worm data sets map sites from all tissues, but with less sensitivity than sciATAC-seq. This difference is because regulatory sites that are highly cell type-specific, and perhaps only found in a small population of the cells in the worm, will be sampled at a rate similar to the background noise from all the other cells included in the assay. Such sites are likely to be missed by the peak caller. Second, in the case of the ChIP-seq data, the recovered sites are biased for the particular set of TFs that are assayed; in contrast, sciATAC-seq is relatively unbiased, but is subject to the Tn5 sequence bias.[73] Third, ChIP-seq data is known to have artifacts associated with the antibody used for immunoprecipitation.[19,62] Last, both data sets include results for multiple developmental stages in the worm, so we expect that our data will not overlap regulatory sites that are specific to other stages.

We find good overlap with both published data sets. First, we overlapped the peaks from the other data sets with our sciATAC-seq summits (Fig. 2.1a). About three quarters of bulk ATAC-seq sites overlap a sciATAC-seq summit (32,099 of 42,102, $\sim 76\%$), and these overlaps are fairly evenly split between sites classified as promoters (11,911 of 13,833, $\sim 86\%$) and enhancers (14,923 of 19,195, $\sim 78\%$), with the remaining overlaps (5,265 of 9,074, $\sim 58\%$) involving other smaller categories of regulatory elements (e.g. non-coding RNAs). In the case of the modERN TF sites, we find that the majority overlap a sciATAC-seq summit (28,545 of 41,542, $\sim 69\%$). Furthermore, nearly all of those ChIP-seq sites that do not overlap a sciATAC-seq summit are "singletons" that were only observed in one of the 427 ChIP-seq data sets (10,810 of 12,997, $\sim 83\%$). Previous work suggests that singleton sites are enriched in false positives,[62] which would be less likely to appear in an orthogonal assay like sciATAC-seq. We also find that the set of ChIP-seq sites that overlap sciATAC-seq summits is enriched for those found in larval samples, especially L2, and depleted for sites observed in embryo and young adult (Fig. 2.1c).

Reversing the comparison and overlapping our sciATAC-seq summits with peaks from the other two data sets, we find less extensive overlap (Fig. 2.1b). Intersecting the sciATAC-seq summits

with the bulk ATAC-seq data set shows 32,572 summits of 74,067 ($\sim$ 44%) overlapping a bulk ATAC-seq site overall. About 47% (15,292) of the summit overlaps were with sites classified as enhancers,[65] about 37% (12,131) were with sites classified as promoters, and 16% (5,149) were with other kinds of sites. We also find that 47,005 of the 74,067 peak summits ($\sim$ 63%) overlap TF ChIP-seq peaks from modERN (Fig. 2.1a). In the ensuing sections, we demonstrate the high quality of the sciATAC-seq data, and we suggest that many of the sciATAC-seq summits that do not overlap another data set are previously-unobserved regulatory elements.

2.4.3 Latent Dirichlet Allocation modeling reveals 26 clusters of cells.

In order to analyze the results of our sciATAC-seq experiment, we use Latent Dirichlet Allocation (LDA),[56] a modeling strategy that is well-suited for analyzing single-cell accessibility data[54] (Fig. 2.2a). LDA is a generative Bayesian modeling approach that was developed in the context of document classification. In the document classification task, the model is trained to identify information-rich words in a document corpus that are associated with latent topics that can distinguish the documents. The output consists of two matrices: one that captures the probability distribution of each topic over all words, and another that captures the probability distribution of each document over all topics. Thus, each topic is defined as some combination of words, and each document is associated with some combination of topics based on its word content.

When applied to scATAC-seq data, cells are treated as documents, and peaks are treated as words. The model learns the peaks associated with latent "regulatory topics" that distinguish groups of cells. The output consists of two matrices: one representing the distribution of peaks over topics, and another representing the distribution of topics over cells. A key advantage of LDA in this setting is that it handles sparsity in the data quite well. The data for any given cell in a sciATAC-seq experiment are extremely sparse – for any accessible site in a given cell, there are only two chances for that site to be cut by the Tn5 and sampled by the assay (one for each copy of the genome in the worm nucleus). In the case of *C. elegans*, this problem is compounded by low library complexity, perhaps due to the tiny, dense nuclei of the worm making it difficult for

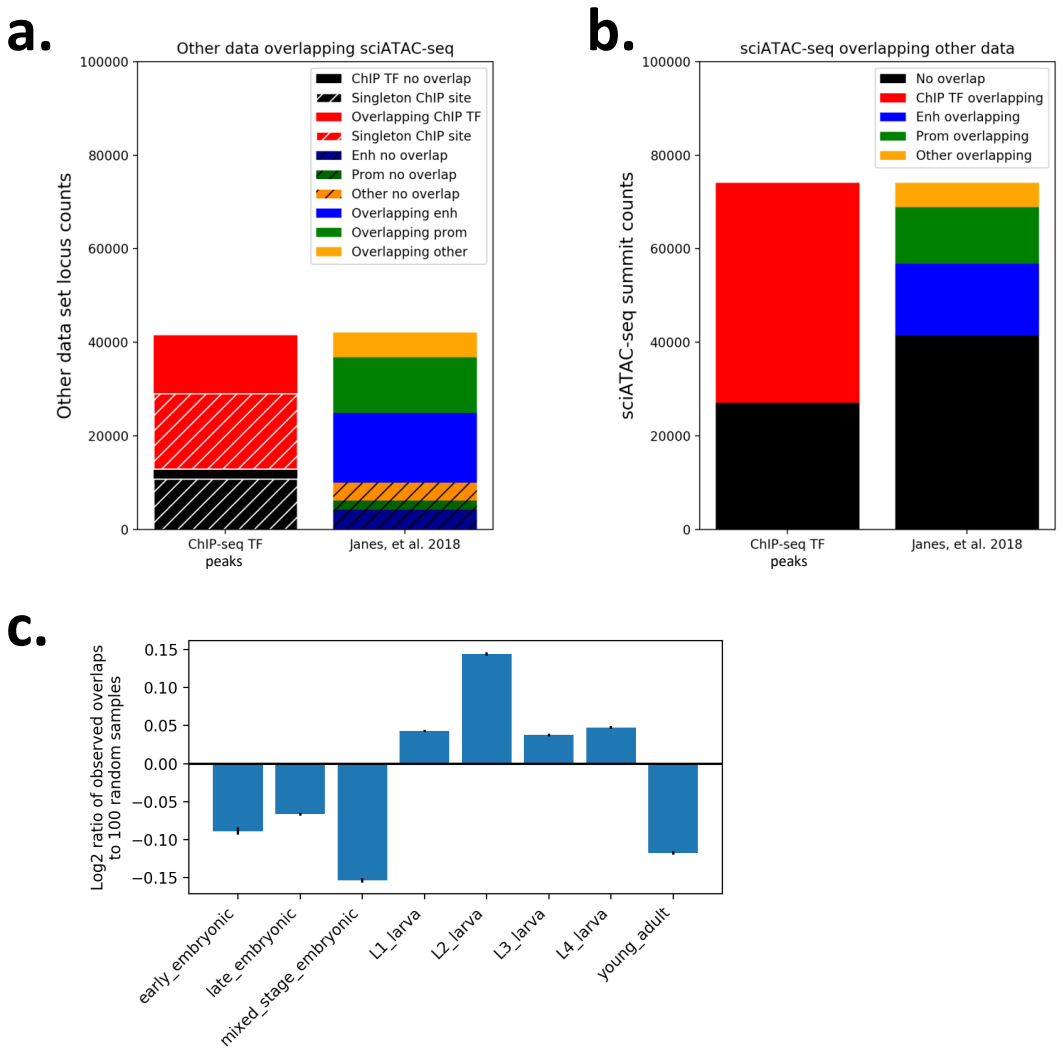

Figure 2.1: **The peaks called from sciATAC-seq data exhibit substantial overlap with existing chromatin data collected from whole worms. a.** TF ChIP-seq peaks from modERN, and bulk ATAC-seq peaks from Janes et al. 2018 show substantial overlap with sciATAC-seq summits. Most of the ChIP-seq TF peaks that do not overlap a sciATAC-seq summit are singleton peaks that are only found in a single experiment. **b.** If sciATAC-seq summits are overlapped with peaks from the same two data sets, the number of overlaps is closer to 50%, suggesting that the single-cell assay finds novel regulatory elements compared to bulk assays. **c.** Breaking out the ChIP-seq peak overlaps by the developmental stage of the worms assayed and comparing the distribution across stages of the peaks with overlaps compared to the stage distribution for randomly selected ChIP-seq peaks shows an enrichment for peaks found in larval stages, and especially L2. Error bars indicate the 95% confidence interval.

Tn5 to access the DNA, particularly after formaldehyde fixation. Our median coverage was only about 700 unique reads per cell, and even with an excellent rate of 75% of those reads mapping to peak regions (Fig. A.2) the amount of information per cell is small. As a point of comparison, sciATAC-seq on embryos from *Drosophila* yielded a median of over 10,000 unique reads per cell.[41] Nevertheless, despite the low coverage per cell, LDA leverages information from all cells at once to assign peaks to topics and all peaks at once to assign topics to cells.

We trained an LDA model with 60 topics. We chose 60 topics based on evaluation of LDA models trained with different numbers of topics using cisTopic[54] (see Discussion, Methods, Fig. A.4 for more information on choosing an appropriate number of topics). The LDA analysis yielded a cells-by-topics matrix with 31,611 rows and 60 columns, and a summits-by-topics matrix with 76,067 rows and 60 columns (Fig. 2.2a). [Note that in the text we will transpose the topics-by-summits matrix and refer to it as the summits-by-topics matrix for consistency with the cells-by-topics matrix.] The first question we sought to answer was how well the topic modeling could separate groups of cells. To visualize these groups, we reduced the dimension of the cells-by-topics matrix from 60 to 2 using the algorithm Uniform Manifold Approximation and Projection (UMAP).[47,74] UMAP applies mathematical topological theory to model the relationships between the cells in the 60-dimensional topic space, and then uses this model to map the cells to coordinates in two-dimensional space while preserving the cell relationships as much as possible. After applying UMAP and plotting the cells in two dimensions, we find very clear separation among groups of cells (Fig. 2.2b). When we investigated the topic contribution to these groups of cells, we found that most of the clusters are composed of cells for which LDA assigned a single high-probability topic (Fig. 2.2c).

To understand what information these high-probability topics were capturing. We first sought to identify which single topics are most important in these clusters of cells, and to group cells by topic for further analysis. We reasoned that by selecting a subset of topics that appeared to have strong and specific signal in the cells, we would prioritize topics that were most likely to be interpretable. We focused on finding topics associated with cells in dense clusters that were highly specific to one topic (we will refer to these groups of cells as "topic clusters"). For each topic, we

ranked the cells by their probability of belonging to that topic and calculated the average similarity of the top 50 most-specific cells to the centroid of these cells in the 60 topic space. We ranked the topics by the mean centroid similarity of their top 50 most-specific cells and took the top 36 topics, which passed a threshold of 0.2. After visually inspecting these topics in the UMAP space, we filtered out any that were either composed of only a few cells that were really part of a larger cluster or did not appear to be a single coherent cluster, resulting in a final list of 26 topics. To generate clusters, we assigned any cells with greater than 50% probability for one of these topics to a cluster associated with that topic. For eight of the topics only a small number of cells met this criterion, so in these cases we relaxed the specificity criterion by adding nearest neighbors to the topic cluster until the visually-distinguishable cluster in the UMAP plot had good coverage (this procedure resulted in adding 115-250 cells to the eight topic clusters). In the end, we assigned 18,134 cells to a topic cluster ($\sim$ 57%, Fig. 2.2b). We note that this is a fairly stringent criterion; even cells with less than 50% probability for their most-probable topic assignment still only have substantial probability for two or three topics, and those topics tend to be fairly similar to each other (Fig. 2.2c).

2.4.4 Topics correspond to specific tissue identities.

After clustering our cells based on 26 topics, we asked what these topics represent. As with other dimensionality reduction techniques (e.g. principal component analysis), LDA is an unsupervised algorithm with no restrictions on what qualities of the data it uses to determine the topics, and interpretation of the topics can be challenging. However, given that the sciATAC-seq summits showed good overlap with existing bulk data, we hypothesized that our selected topics were picking up on biological signal and that they represented different tissue types. Other topics that were less dense and/or less specific could represent different kinds of regulatory activity (e.g. promoters or enhancers[54]), cells with more complex patterns of regulatory activity, cells with noisy signal, or cells with insufficient signal to be confidently clustered.

One way to assess whether the topics show some tissue-specificity is by cross-referencing the

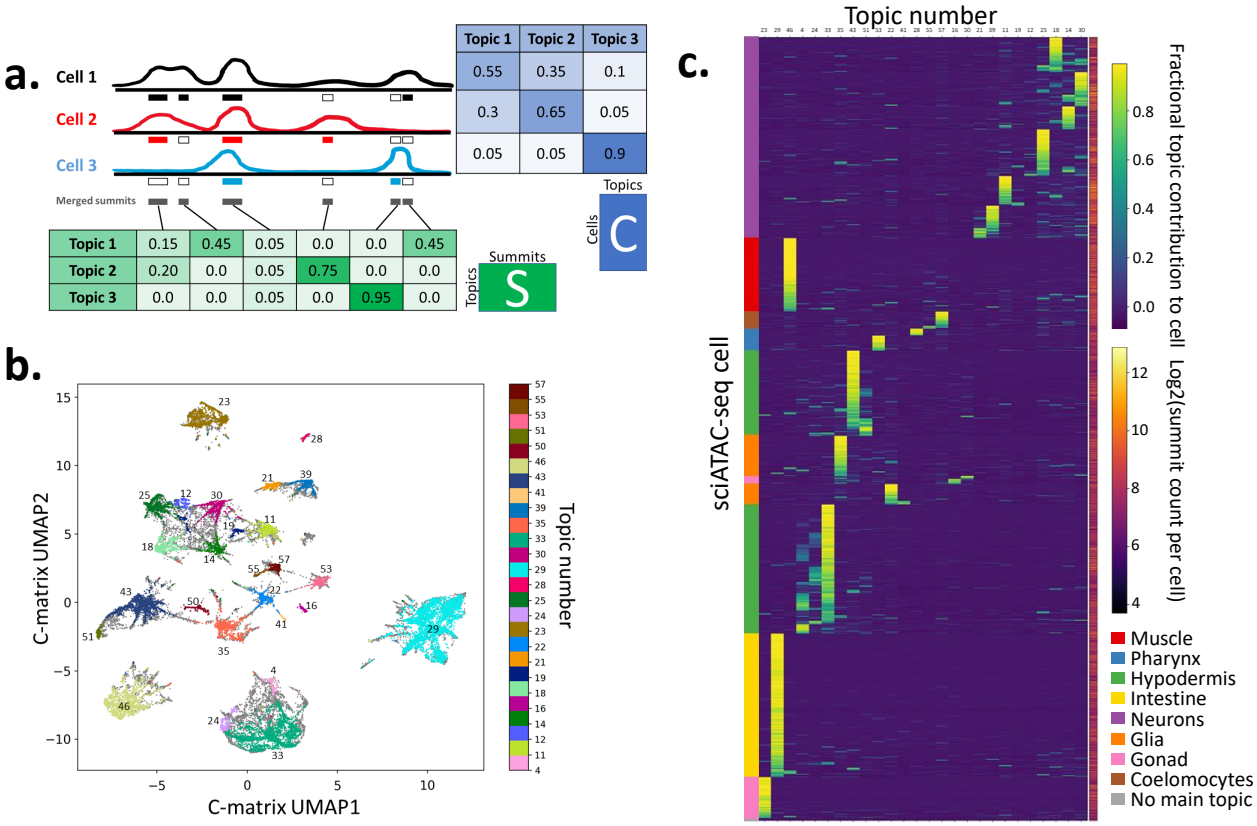

Figure 2.2: **Latent Dirichlet Allocation modeling yields 26 major cell clusters that are characterized mostly by a single topic each. a.** Schematic showing a hypothetical example of data and LDA results for three cells that typify three topics. LDA modeling learns latent topics that explain the data and return two matrices, here designated *S* and *C*. Matrix *S*, referred to in the text as the summits-by-topics matrix, captures the probability distribution of each topic over all peak summits, while matrix *C*, referred to in the text as the cells-by-topics matrix, captures the probability distribution of each cell over all topics. Cells that exhibit summits with high probability in a given topic will in turn show high probability for that topic. **b.** UMAP embedding of the *C* matrix after L2 normalization of the rows shows 26 well-separated clusters of cells that can be associated with individual topics. **c.** Heatmap showing the normalized *C* matrix values for the 26 topics associated with clusters in **b.**; this plot highlights that most cells have probability concentrated in one or a few topics. Cell types determined for the topics based on analysis of the *S* matrix are annotated on the left, and cell summit coverage is shown to the right.

sciATAC-seq summits with what is known about those loci in the literature, similarly to how marker genes are identified for clusters in scRNA-seq data.[30,33] Even though this data set is the first chromatin accessibility data set in *C. elegans* with cell type resolution, we can get an idea of whether the topics are distinguishing cell types by overlapping summits associated with each topic with ChIP-seq peaks from cell type-specific transcription factors. For each of the 26 topics that we used to cluster the cells, we found all summits in the summits-by-topics matrix with probability greater than zero for that topic and overlapped them with all available ChIP-seq peaks from sites found in 40 or fewer other ChIP-seq data sets (i.e. non-HOT sites) for three transcription factors with known cell type-specific expression patterns: *hlh-1*, a master regulator for body wall muscle;[75] *elt-1*, a master regulator for hypodermis in embryos and seam cells in L2 larvae;[76] and *elt-2*, a transcription factor important in intestine development.[77] We compared the number of overlaps in each topic to the number we would expect if the overlaps were random (i.e. if topics were not cell type-specific), and expressed this comparison as a $\log_2$ ratio between observed and random overlap counts (Fig. 2.3). We find topics with specific enrichment for overlaps with peaks from each transcription factor (95% confidence intervals are provided in Fig. 2.3). Topics 29 and 46 are most enriched for overlaps with *hlh-1* sites, while topics 43 and 51 are particularly enriched for overlaps with *elt-1* sites, and topic 29 is most enriched for overlaps with *elt-2* sites. This analysis suggests that at least some of the topics are representing different tissues, and in particular that the cells associated with topic 46 are muscle, those associated with topics 43 and 51 are hypodermis, and those associated with topic 29 are either muscle or intestine. The clusters for topics 46 and 29 are well-separated in the UMAP plot of cells, while those for topics 43 and 51 are not (Fig. 2.2b), so we expect that 46 is muscle, while 29 is intestine.

Encouraged by the analysis of overlaps with ChIP-seq data from cell type-specific TFs, we sought to leverage the L2 sciRNA-seq data[30] to obtain a more comprehensive analysis of all 26 topics. In order to do this, we mapped the sciATAC-seq summits to the nearest downstream gene that was still within 1200 bp. Assigning regulatory regions to their nearest gene is a commonly used[62] but rather naïve heuristic that nevertheless works well in *C. elegans*. In organisms like human and mouse, there are widespread and complex interactions between promoters and distal

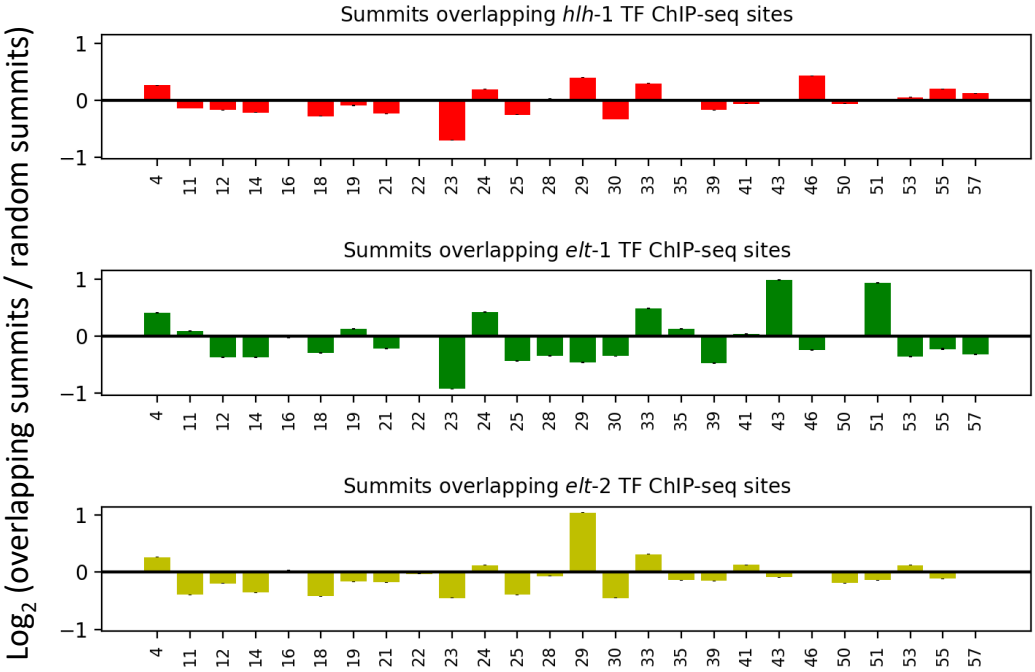

Figure 2.3: **Overlapping summits important for each topic with ChIP-seq peaks collected from cell type-specific TFs suggests at least some topics represent tissue types.** Summits associated with each topic were overlapped with ChIP-seq peaks for three cell type-specific transcription factors: *hlh-1*, which is specific for muscle (top plot); *elt-1*, which is specific for seam cells (middle); and *elt-2*, which is specific for intestine (bottom). Topics distributions for summits with peak overlaps were compared to the topic distribution for randomly sampled summits and the results are plotted here as the $\log_2$ ratio of the overlap topic distribution to the random topic distribution. Error bars represent the 95% confidence interval after comparing the overlap topic distribution to the topic distribution of 100 random samples.

enhancers that contribute to gene regulation, and mapping these interactions is a hard problem;[8] however, there is little evidence that such regulation by distal sites occurs in *C. elegans*.[78] Furthermore, the *C. elegans* genome is compact and gene-dense, meaning that most regulatory sites are found close to genes, either in intergenic sequences or in introns.[78] In total, we were able to assign 49,873 summits to 17,484 genes. The number of genes we associate with accessible regions is higher than the number of genes known to be expressed in L2 worms,[30,79] and this is most likely due to two factors: first, some of the annotated genes in the RefSeq database that we used are not protein coding genes and are not found in the sciRNA-seq data (and thus excluded from the following analysis), and second, some of the summits located in introns are in positions near the 3' end of the overlapping gene and appear unlikely to regulate it. Summits in the latter category warrant further investigation and may be examples of distal regulatory elements in gene-rich regions of the genome.

With these caveats, we used these peak-gene assignments to associate genes with topics and thereby infer whether or not the topics are clustering cells by tissue type. For each topic, we computed the mean expression distribution across tissues for the top 500 genes in that topic and then the $\log_2$-ratio of that to the mean expression distribution of 500 randomly-selected genes (Fig. 2.4). It is clear that the summits that are specific to particular topics are also near genes that show tissue-specific expression patterns in the sciRNA-seq data. In fact, many of the topics show evidence of specificity for tissue sub-types, including many different kinds of neurons and combinations of hypodermis tissues that suggest, for example, anterior versus posterior seam cells based on the presence/absence of rectum-associated genes (topic 43 versus topic 51), and the combination of muscle and sex myoblast expression in topic 55 supporting a sex myoblast identification. At this resolution, there appears to be no distinction between body wall muscle and intestinal/rectal muscle (topic 46 encompasses both), and it is not possible in all cases to assign a sub-type to topics enriched in neuronal genes. This limited cell type resolution could be due to instances of differing patterns of accessibility marking alternative promoters for the same set of genes that are expressed in many neuron sub-types (i.e. a common neuronal program); pan-neuronal, non-productive accessibility near genes that ultimately are expressed in only a specific subset of neurons; or noise in

the LDA resulting in poor separation of cells belonging to neuronal sub-types.

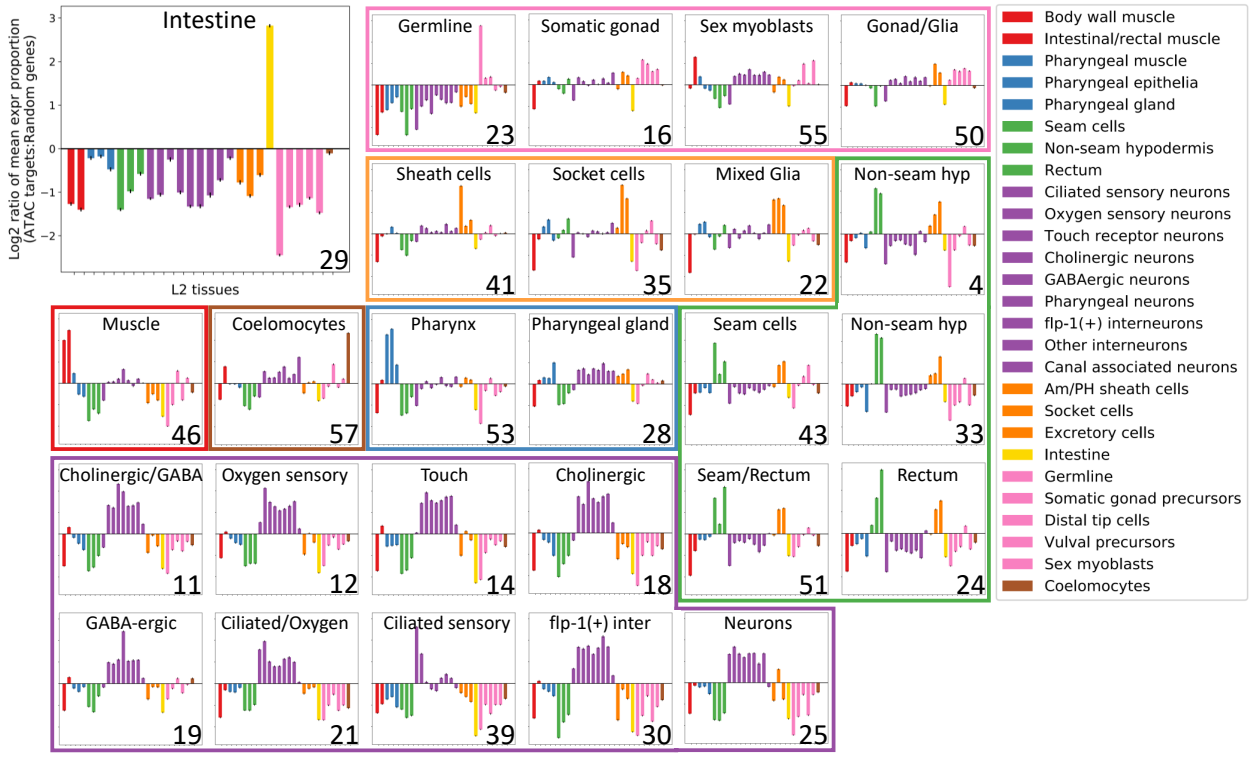

Figure 2.4: **Nearest genes to topic-specific summits tend to be tissue-specific according to single-cell RNA-seq data.** Summits associated with each topic were mapped to the nearest downstream gene, and the tissue expression distribution of the top 500 genes per topic based on summit topic specificity was compared with the tissue expression distribution of 500 randomly-selected genes. The results for each topic are plotted as the $\log_2$ ratio of the topic-associated tissue expression distribution to that of randomly selected genes. Error bars represent the 95% confidence interval after comparing to the tissue expression distribution of 100 random samples. Topics with similar tissue specificity patterns are grouped together, and the tissue type names and colors are as in Cao et al. 2017.

After identifying cell types associated with our topics, we revisited the list of summits that had no overlap with TF ChIP-seq sites to look for differences between those with overlaps and those without (Fig. 2.5). Splitting the peaks-by-topics matrix based on peaks with and without an overlap from a TF ChIP-seq peak shows that the summits overlapping a ChIP-seq peak (Fig. 2.5a) are more likely to contribute to multiple topics, and are also generally found in more cells than those

without overlaps (Fig. 2.5b). There are novel accessible sites associated with all topics/tissues, but the topics with the most novel sites are topic 23/germline and the neuron topics collectively. The high number of novel germline sites could be due to fewer germline-specific transcription factors being targeted by ChIP-seq, while the abundance of novel summits associated with neuronal topics could reflect either that the whole worm ChIP-seq assay is not sensitive enough to find sites specific to neuronal sub-types, or that ChIP-seq for the relevant TFs was not even attempted due to their expression being restricted to a small number of cells.

2.4.5 LDA modeling of cells from individual tissue types detects fine-grained cell types and sub-types.

Thus far, we have shown that the topics we identified can distinguish cells at the level of tissue type, but we wondered if a more focused analysis of cells from a particular tissue would yield more specific cell identities. We tested this hypothesis on two groups of cells: the 2,079 cells corresponding to the body wall muscle cluster (topic 46), and the 3,872 cells corresponding to the 9 neuron clusters (topics 11, 12, 14, 18, 19, 21, 25, 30, and 39). The muscle cells had 8,221 peaks, and the neurons had 17,683 peaks (after merging the peaks called on each neuron topic cluster individually). We fed the muscle and neuron data subsets each into a new LDA analysis and looked for groups of cells that correspond to specific cell types within each tissue.

Body wall muscle cells are quite similar to each other, despite differentiating from four different embryonic lineages. In previous single-cell RNA-seq studies, the body wall muscle cells all clustered together, without much separation.[30,33] Nevertheless, within the body wall muscle cluster the cells were previously found to group by anatomical position, setting up an anterior-posterior axis through the cluster that was identified by looking for the expression of specific marker genes. We reasoned that a similar pattern might exist in the sciATAC-seq data from muscle cells. We checked for evidence of chromatin accessibility near seven marker genes that are specific to different kinds of muscle (Fig. 2.6a). The markers *hlh-1* and *hlh-8* are expressed in body wall muscle and other kinds of mesodermal cells, respectively. We found no evidence of chromatin accessibil-

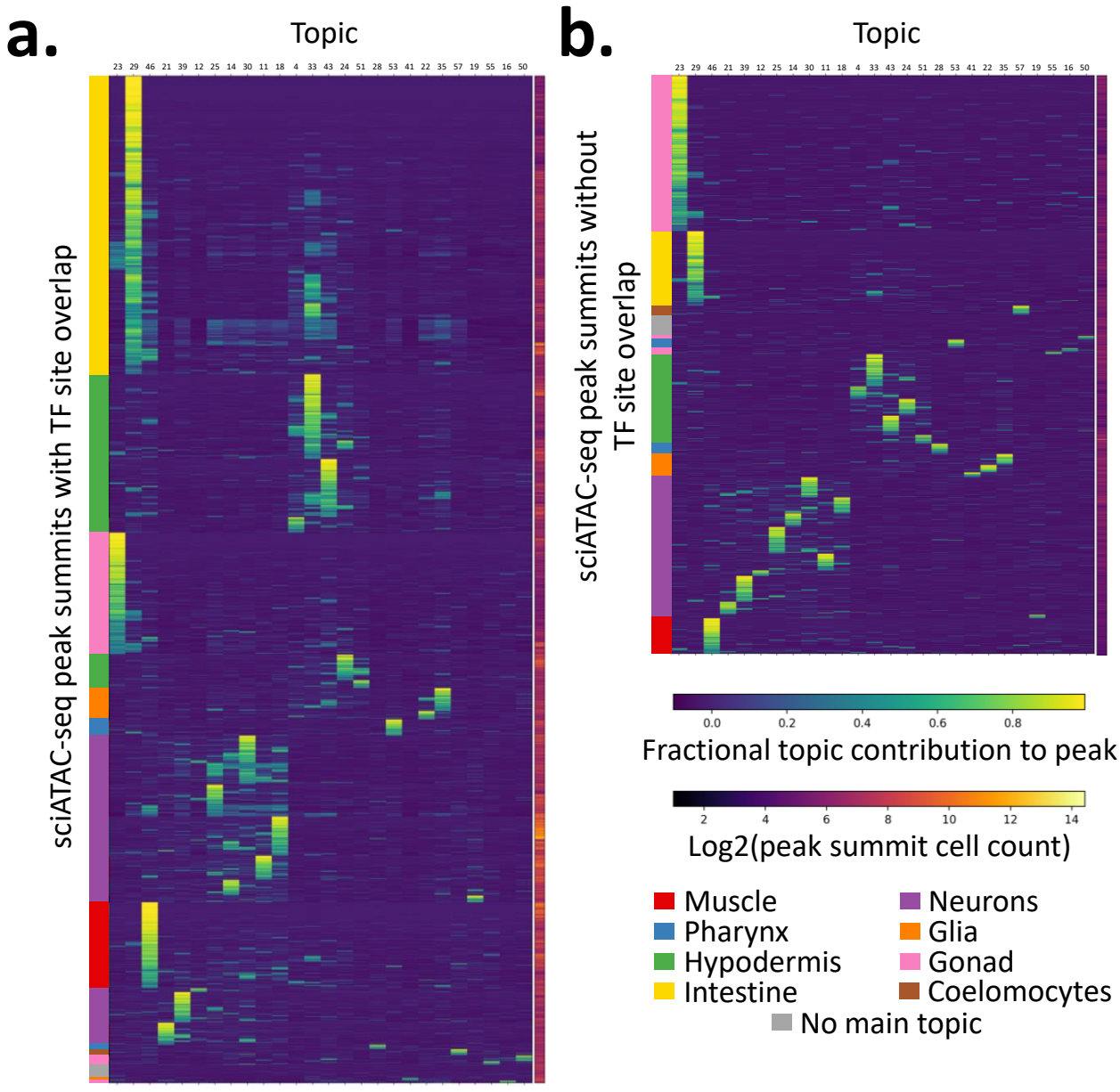

Figure 2.5: **Sites of accessible chromatin with no overlapping modERN ChIP-seq peaks suggest novel regulatory sites, especially for germline and neurons.** We compare the normalized summit-by-topic matrix values between the summits that overlap a ChIP-seq peak (**a.**) and those that do not (**b.**). The non-overlapping summits are enriched for topics associated with gonad (especially germline/topic 23) and topics associated with neurons. The non-overlapping summits also tend to be observed in fewer cells.

ity upstream of *hlh-8* in topic 46 (Fig. 2.6a), but a very prominent peak near *hlh-1*, suggesting that this cluster consists largely of body wall muscle. We also checked for accessibility near five genes that serve as markers for body wall muscle cells at different points along the anterior-posterior axis of the worm. The genes *eya-1* and *ceh-34* are expressed in anterior body wall muscles found in the head and neck, while *ceh-13* is expressed in the body wall muscles in the middle section of the worm, and *cwn-1* in the posterior body wall muscles. The gene *egl-20* is expressed at the posterior extreme and in very few cells (only two body wall muscle cells during embryogenesis). We find chromatin accessibility in all of the positional marker genes except for *egl-20*. See Figure A.9 for more details about the marker genes and the nearby patterns of chromatin accessibility in different cell types.

In order to use these marker genes to detect muscle subtypes, we first trained a new LDA model with 5 topics (evaluated using 5-fold cross validation, see Fig. A.8) on just the cells in the topic 46 cluster, projected the cells into two dimensions using UMAP, and plotted the cells as a scatter plot. As in the scRNA-seq papers, we find that the muscle cells mostly stay in a single large cluster (Fig. 2.6b). Next, we highlighted cells with a read overlapping the closest peak of chromatin accessibility to the 5' end of a marker gene (but also no more than 1200 bp away). We find that cells with accessibility near *hlh-1* are distributed throughout the cluster, confirming the high content of body wall muscles in this cluster. When we plot the positional marker genes we find a clear anterior-posterior axis running through the cluster along the UMAP2 dimension.

Next, we conducted a similar analysis on all cells from neuron-enriched clusters. The tissue type categories based on the L2 scRNA-seq work[30] already break gene expression distributions down into nine different neuronal subtypes. However, we do not always see obvious concordance between our topic clusters and these neuronal subtypes; it can be hard to say to which subtype a given topic cluster corresponds (Fig. 2.4). In order to take a closer look at the neuron cells, we gathered all cells from topic clusters showing neuronal enrichment (topics 11, 12, 14, 18, 19, 21, 25, 30, and 39), and performed the same analysis that we did for body wall muscle above. We trained a 20-topic LDA model (Fig. A.10), projected the cells to two dimensions using UMAP, and colored cells in the resulting scatter plot by their evidence for chromatin accessibility near marker

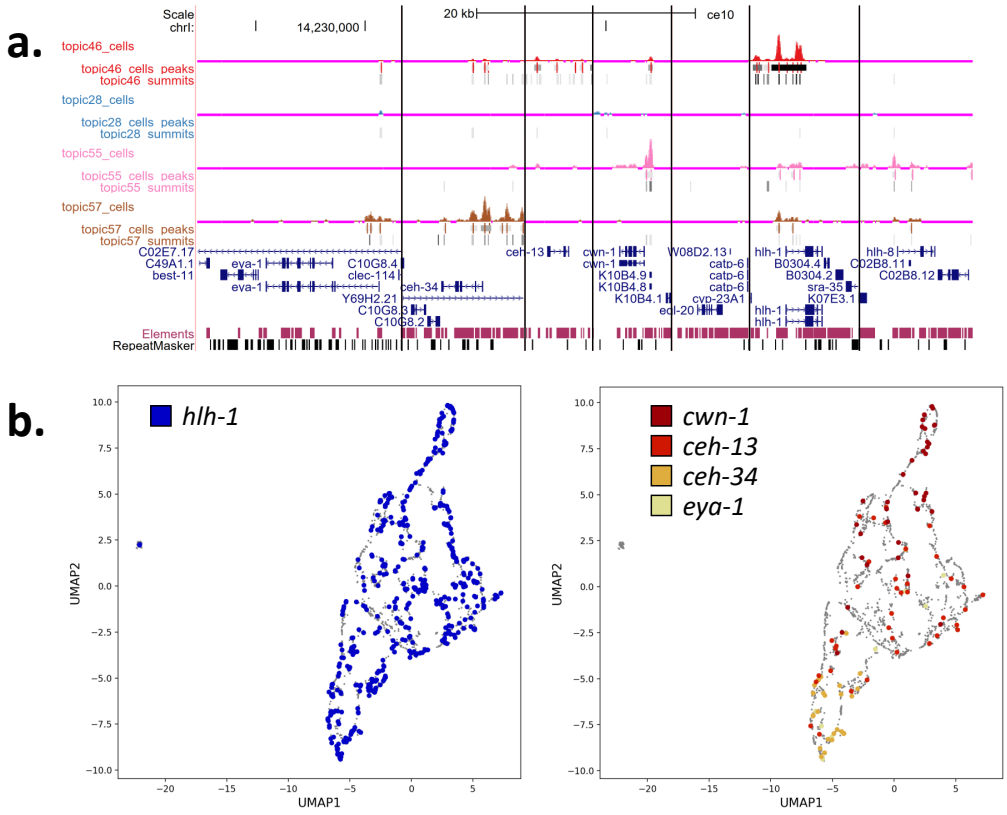

Figure 2.6: **Subclustering of muscle cells separates cells by position along the anterior-posterior body axis. a.** Multiview figure from the UCSC Genome Browser displaying data for regions around 7 marker genes for different types of muscle. Signal tracks show the aggregated cut site coverage of cells from topic clusters identified as muscle-related (topic 46: body wall/intestinal/rectal muscle, topic 28: pharynx, topic 55: sex myoblasts, topic 57: coelomocytes). Each signal track is paired with two BED tracks. The top one in each pair indicates peaks called on the topic-specific data shown in the signal tracks, while the bottom one in each pair shows the sciATAC-seq summits used to define the topic clusters that led to the signal data. The color intensity of the peaks indicates their level of statistical significance, while the color intensity of the summits indicates their level of topic-specificity. All genes except for *egl-20* and *hlh-8* show called peaks in topic 46. **b.** Cells associated with topic 46 (body wall/intestinal/rectal muscle) were modeled on their own with LDA and embedded in a 2D space with UMAP for visualization. Then, the nearest peak to each marker gene was identified and cells showing a read at that peak are emphasized with larger dots and color. Cells throughout the plot show evidence of accessibility at the master body wall muscle transcription factor, *hlh-1* (left plot), but based on markers for head and neck body wall muscle (*eya-1*, *ceh-34*), anterior/middle body wall muscle (*ceh-13*), and posterior body wall muscle (*cwn-1*), there is an anterior/posterior axis to this cluster that roughly corresponds to UMAP2.

genes (Fig. A.11). We evaluated five marker genes in this analysis that we chose for their tissue-specific expression patterns based on the scRNA-seq data: *bbs-8* is expressed in ciliated sensory and oxygen sensory neurons, *gcy-32* is expressed exclusively in oxygen sensory neurons, *unc-30* is expressed in GABA-ergic neurons, *mec-7* is expressed in touch sensitive neurons, and *ceh-24* is expressed in cholinergic neurons. We find that the cells with chromatin accessibility near these genes are associated with well-separated clusters in UMAP space (Fig. 2.7a), suggesting cell type identities for these clusters.

In order to verify the neuron subtypes identified by single marker genes and also attempt to identify subtypes for the other clusters of cells, we followed up with a more global analysis. For each neuron subtype from the scRNA-seq paper, we ranked the genes by the proportion of their total expression that comes from that subtype. Next, we picked the top 50 of those subtype-specific genes that had an accessible site within 1200 bp of the 5' end of the gene. Also, after finding that topic 16 had a high probability in cells throughout the UMAP, we additionally required the accessible sites associated with the 50 genes to have low probability in topic 16. Once we identified the 50 accessible sites, we picked the top topic for each site and took a weighted average of the probabilities of those topics in all of the cells such that the contribution of each topic to the average was proportional to the number of peaks for which that topic was the most probable (Fig. 2.7b). Using 50 genes to associate each neuronal subtype with UMAP cell clusters results in assignments that are largely consistent with the neuronal subtypes assigned to clusters by the individual marker genes, but the 50 gene analysis assigns some of the clusters to multiple subtypes and some subtypes to multiple clusters. For example, the *ceh-24* marker gene identified a cluster at the top of the UMAP plot as cholinergic neurons, but by considering the top 50 most-specific cholinergic neuron genes we also find evidence for a cholinergic identity for a cluster at the bottom of the plot. Also, touch receptor and pharyngeal subtypes appear to be very similar by this analysis; and in all subtypes except for GABAergic neurons, one or more high probability clusters are also high probability in another subtype. This result suggests that the neuron subtype LDA clusters do not correspond in many cases with the scRNA-seq subtype classifications. We will have to do more work to understand why the two analyses do not align more closely, but one intriguing possibility is

that the RNA-seq and ATAC-seq patterns are different because of different biological phenomena captured in gene expression versus chromatin accessibility data.

2.5 Discussion

We used the sciATAC-seq assay to assemble the first cell type-resolved map of regulatory elements in *C. elegans*. We found 74,067 peak summits, which we used to assign 18,134 of our 31,611 cells to one of 26 different clusters (Fig. 2.2) that represent distinct differentiated tissues in the L2 nematode (Fig. 2.4). Our map contains almost 30,000 novel putative regulatory sites (Fig. 2.1) that are often highly cluster specific and associated with cell types with small populations in the worm, such as specific types of muscles and neurons (Figs. 2.4, 2.6, 2.7).

A limitation of these data is that the cell type resolution is not as high as we achieved with sciRNA-seq.[30,33] Partly, this lower resolution is due to the inherent challenges of single-cell chromatin accessibility data. single-cell ATAC-seq suffers from a low dynamic range because there are only one or two chances in a diploid cell to sample a given accessible locus, depending on whether or not both alleles are accessible. This is in contrast to single-cell RNA-seq, which has hundreds or even thousands of chances for measuring highly expressed genes. In addition, there exists less information in the literature about which regulatory regions are useful for inferring cell types. We also encountered challenges applying sciATAC-seq in worms. For unclear reasons, the worm yields sciATAC-seq libraries with many fewer unique fragments per cell than other organisms. In the first reported sciATAC-seq results, human and mouse cell lines yielded a median of 2,503 fragments per cell,[39] and in more recent work on fly and mouse cells the median yield was over 10,000 fragments per cell.[40,41] In contrast, despite using the most up-to-date protocol,[40,41] the median *C. elegans* cell yielded only about 700 fragments (Fig. A.2). We hypothesize that this can be improved by optimizing the nuclear isolation and permeabilization conditions – L2 nuclei are extremely small, compact, and dense (about 2μm in diameter), and possibly after formaldehyde fixation they are very difficult to permeabilize adequately for the Tn5 to access the chromatin.

Nevertheless, by modeling the data with LDA, which handles sparsity well, and analyzing the

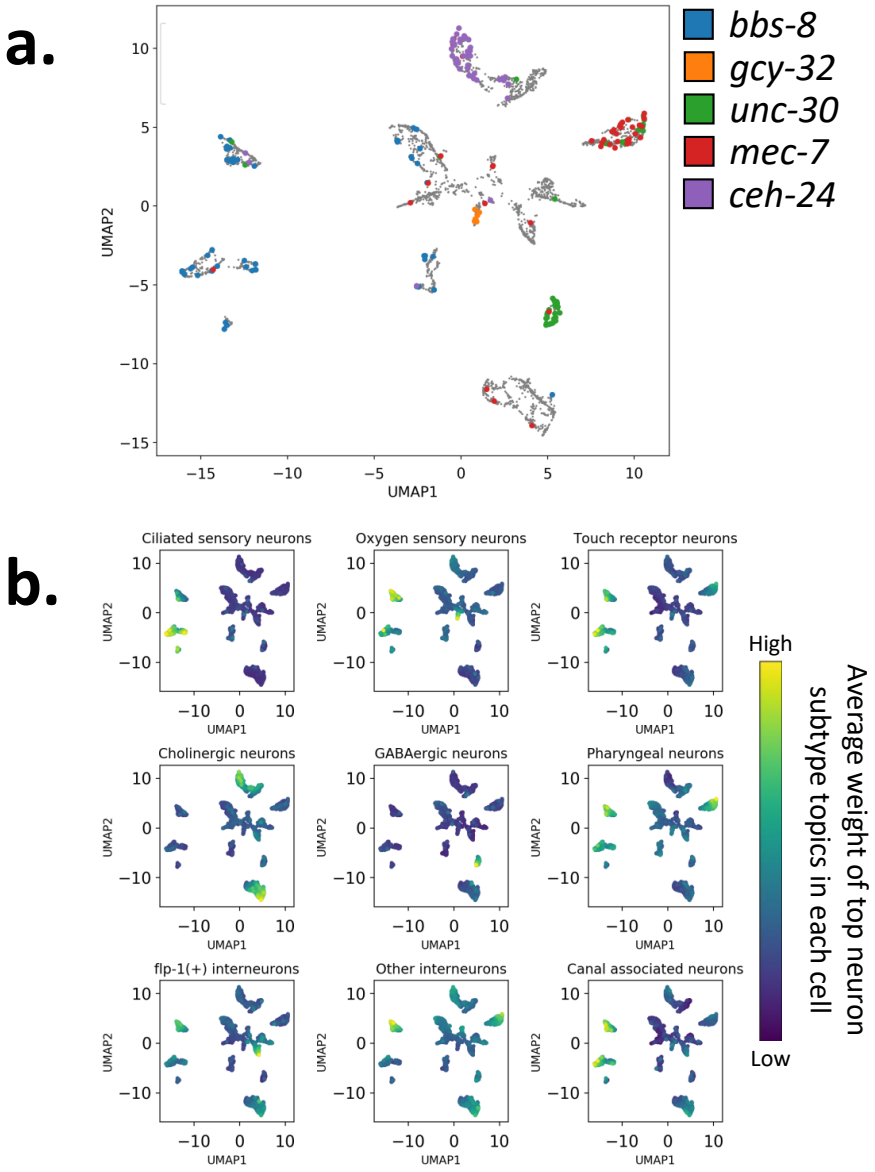

Figure 2.7: **Subclustering of neurons reveals finer structure that distinguishes different types of neurons. a.** Cells with reads in peaks near genes with expression patterns specific to neuron subtypes cluster together (*bbs-8*: ciliated sensory and oxygen sensory neurons, *gcy-32*: oxygen sensory neurons, *unc-30*: GABA-ergic neurons, *mec-7*: touch receptor neurons, *ceh-24*: cholinergic neurons). **b.** We identified the peaks nearest to the top 50 genes with the most specific expression for each neuron subtype (based on the L2 scRNA-seq data set) and the top topic for each of those peaks. Then, we colored each cell in UMAP space by the average probability in that cell of the top topics. In each of the plots, cells with a high average probability are most consistent with the peaks that were associated with that neuron subtype.

cells and peaks in topic space, we identified many cell types and regulatory sites that are specific to each of them. In order to improve these results, it will be important to either increase the yield of fragments per cell, or better tune our modeling approach. We recently implemented a more principled evaluation procedure for choosing the best number of topics to use in LDA (see Methods). After evaluating our choice of 60 topics by using this new procedure (Fig. A.4), it appears that 60 topics is too many. For our initial clustering of cells, we should use as few as 10-15 topics (Fig. A.4a), and for the subsequent LDA model trained on the MACS2 peak summits we should use closer to 30-40 topics (Fig. A.4b). It will also be important to do further evaluation to ensure that the model is converged before we stop training, and we may be able to use our topic number evaluation procedure to also choose better values for the Dirichlet prior hyperparameters, α and β. Another way to improve our results could be to better integrate additional information into the analysis. One potentially fruitful analytic approach could be to more tightly integrate the analysis of the sciATAC-seq and sciRNA-seq data sets. There now exist multiple approaches for projecting single-cell data from different modalities into the same embedding space.[41,53,80] By jointly analyzing the sciATAC-seq data and sciRNA-seq data with one of these methods it may be possible to improve the cell type resolution of our chromatin accessibility maps.

Such accessibility maps with high cell-type resolution will be important for understanding gene regulation on the scale of the whole genome across the whole organism. In addition, regulatory sites are hypothesized to play a major role in common disease and evolutionary adaptation, and so maps of regulatory sites will aid in interpreting the effects of genetic variation. For example, many mutations that appear linked to some phenotype by approaches like GWAS do not fall in genes. The implication is that, if one of the mutations is indeed causal, it must fall in a regulatory sequence of DNA. Thus, maps of cell type-specific regulatory regions can help interpret and prioritize candidate causal variants, and will be a useful complement to genetic resources in *C. elegans*, including the *C. elegans* Natural Diversity Resource[81] and the Million Mutations Project strains.[82] Better annotations of regulatory DNA can also help with understanding comparative genomics and the evolution or conservation of stretches of non-coding DNA.

Given the importance of mapping regulatory sites for understanding genome structure and func-

tion, and the power of *C. elegans* as a model organism, improving and expanding our maps of regulatory regions should be a high priority. In particular, collecting accessibility data for additional developmental stages in worm will provide valuable insight into the dynamics of gene regulation over the course of development as cells differentiate. These data can be paired with new scRNA-seq data collected from throughout *C. elegans* embryogenesis,[33] moving the field closer to having a truly comprehensive map of gene expression and regulation for every cell throughout development in *C. elegans*.

2.6 METHODS

2.6.1 Nuclear isolation from whole L2 worms

We grew wild-type *Caenorhabditis elegans* worms (VC2010 strain) at 21°C on nine 150 mm plates and synchronized the population by bleaching (2% bleach, 0.5 M KOH) young adults with 8-12 embryos to isolate embryos, hatching them at room temperature in egg buffer (118 mM NaCl, 48 mM KCl, 2 mM $CaCl_2$, 2mM $MgCl_2$, HEPES 25 mM at pH 7.3) for 12-16 hours, and re-plating the L1 hatchlings onto nine more 150 mm plates at a density of approximately 60,000 worms per plate. After two rounds of this bleach synchronization and plating, the L1 worms were allowed to grow at 21°C for 19 hours after plating to reach the middle of the L2 stage. The worms were washed off eight of the plates with M9 buffer (22 mM KH_2PO_4, 22 mM Na_2HPO_4, 85 mM NaCl, 1 mM $MgSO_4$ at pH 6.5) into a 50 ml conical tube. Bacteria were removed from the suspension by spinning the tube at $\sim 3,000 \times g$, aspirating the supernatant, resuspending in fresh M9. The M9 wash was repeated, and nearly all of the supernatant was aspirated, leaving a worm pellet in $\sim$ 1ml of M9. The worm pellet was flash frozen by using a P1000 to transfer the worms drop by drop into a mortar containing liquid nitrogen. The frozen worms were crushed into powder with a pestle such that each worm broke into 3-4 chunks, and the powder was transferred to a 50 ml falcon containing 8.75 ml of 1.1% formaldehyde in egg buffer supplemented with 1x protease inhibitor. Worms were rocked at room temperature for 10 min before the fixation reaction was quenched by adding 1.25 ml 1M glycine (final concentration $\sim$ 125mM) and incubated another 5 min at room temperature.

The fixed worms were pelleted at $3220 \times g$ for 5 min at 4°C, the supernatant was removed, and the pellet was resuspended in 10ml ice cold egg buffer. Fixed worms were pelleted again by spinning at $3220 \times g$ for 5 min at 4°C. The egg buffer supernatant, and the pellet was resuspended in ice cold 2x nuclear preparation buffer (20 mM HEPES pH 7.6, 20 mM KCl, 3 mM $MgCl_2$, 2 mM EGTA, 0.5 M sucrose, 0.05% Triton X-100 in egg buffer) supplemented with protease inhibitor (NPB+PI). The following steps were all performed at 4°C or on ice: The solution was transferred to a 7 ml Dounce homogenizer and the fixed worm chunks were homogenized with 20 loose pestle strokes followed by ten tight pestle strokes. The Dounce was spun for 90 seconds at $\sim 200 \times g$ in a swing-arm centrifuge to loosely pellet debris, and the top 1000 μl of supernatant (containing the nuclei) was removed to a 15 ml falcon tube on ice. 1 ml of fresh NPB+PI was added to the Dounce, the debris pellet was gently resuspended, and the Douncing and spinning were repeated three more times, resulting in the collection of 4 ml of nuclei. The suspension of nuclei was cleaned by gently passing through a 10µm syringe filter pre-wetted and chased with 1 ml ice cold NPB+PI into a new 15 ml falcon on ice. The nuclei were split evenly into 1.5 ml eppendorf tubes and pelleted at $2000 \times g$ for 10 min at 4°C. All supernatant was removed, and the pellets were each gently resuspended in 1 ml freezing solution (50 mM Tris at pH 8.0, 25% glycerol, 5 mM $Mg(OAc)_2$, 0.1 mM EDTA, 5 mM DTT, 1 protease inhibitor cocktail (Roche), 1:2,500 superasin (Ambion)).[40] The resuspended nuclei were transferred to 2 ml cryotubes, flash frozen in liquid nitrogen, and stored at -80°C.

2.6.2 single-cell ATAC-seq via combinatorial indexing

The sci-ATAC-seq protocol was as described by Cusanovich, et al. 2018.[40] Briefly, flash-frozen VC2010 nuclei were thawed in a 37°C water bath and put immediately on ice. The nuclei were transferred to a 1.5 ml eppendorf tube and spun at 2000xg for 10 min. The supernatant was aspirated, and the pellet was resuspended in 200 µl of ATAC-OMNI[36] RSB (10 mM Tris-HCl pH 7.4, 10 mM NaCl, and 3 mM $MgCl_2$ in water) supplemented with 0.01% Digitonin, 0.1% IGEPAL-630, and 0.1% Tween-20, allowed to stand for 3 min on ice, and then quenched by adding 1 ml

of RSB supplemented with 0.1% Tween-20. The resuspended and lysed nuclei were stained with 1x Hoechst and a BD FACS Aria II was used to distribute 2500 nuclei into each well of a 96-well v-bottom plate (Eppendorf twin.tec LoBind skirted 96 well PCR plate) prepared with 19µl of tagmentation reaction solution (10µl 2x Nextera TD buffer, 3.3µl 1X DPBS, 0.2µl 1% Digitonin, 0.2µl 10% Tween-20, 5.3µl H_2O).[36] After sorting, 1.0 µl of 2.5 µM uniquely-barcoded Tn5 from Illumina[39] was pipetted into each well of the 96 well plate, and the transposition reaction was allowed to proceed at 55°C for 30 minutes. Next, 20 µl of STOP reaction buffer (40 mM EDTA and 1 mM Spermidine) was added to quench the reaction, and the plate was put at 37°C for 15 min. After stopping transposition, all nuclei were pooled into a 15 ml conical tube, re-stained with 1x Hoechst, and distributed by FACS into twenty eight 96-well v-bottom plates at 25 nuclei per well. The 96-well plates contained 12µl per well of reverse cross-linking buffer (0.83 mg/ml Proteinase K and 0.042% SDS in Qiagen EB buffer), and were put on ice, spun down, and frozen at -20°C in batches during sorting. Later, these plates were thawed in groups of four for reversing crosslinks by incubating at 65°C for 16 hours, after which the transposed and un-crosslinked fragments were amplified using uniquely-barcoded PCR primers. We ran four wells as test reactions in qPCR and monitored the libraries for saturation of SYBR-Green signal to identify the number of cycles required for appropriate amplification,[40] and then amplified the rest of the wells for either 22 cycles with Illumina NPM 2x PCR master mix or 23 cycles with NEBNext 2x PCR master mix (PCR Reaction: 12.0 µl of nuclei in reverse crosslinking buffer; 2.5 µl of 5 µM Nextera v2 barcoded P7 PCR primer; 2.5 µl of 5 µM Nextera v2 barcoded P5 PCR primer; 1.0 µl of 100X BSA; 25.0 µl of 2x NEBNext PCR Master Mix (NEB cat M0541); and 7.0 µl nuclease free H_2O) (NPM PCR protocol: 72°C for 3:00, 98°C for 0:30, repeat$\times$22(98°C for 0:10, 63°C for 0:30, 72°C for 1:00), 4°C HOLD; NEBNext PCR protocol: 72°C for 5:00, 98°C for 0:30, repeat$\times$23(98°C for 0:10, 63°C for 0:30, 72°C for 1:00), 4°C HOLD). After amplification, the fragments were cleaned up by pooling the contents of all wells and splitting across four Zymo Clean and Concentrate columns (cat. D4014), eluted each in 25 µl Qiagen EB, combined the eluates, and then further cleaned and concentrated with 1x Ampure XP magnetic beads, and finally eluted in 25 µl. Using an Agilent TapeStation D5000 kit (Screentape cat. 5067-5588, and reagents cat. 5067-5589), library quality and molarity

were quantified, establishing a 200 - 1000 base pair window for fragments that will cluster well during sequencing. Libraries were then diluted to 2 nM, pooled equimolar and denatured using Illumina's conditions and specification. To achieve proper cluster density, a loading concentration of 15 pM for MiSeq and 1.8 pM for NextSeq were used during sequencing with custom primers and recipe from Illumina.

2.6.3 Generation of genomic DNA input control

In order to control for the sequence cutting bias of Tn5,[73] we treated naked *C. elegans* genomic DNA with the bulk ATAC-seq protocol.[35] We isolated genomic DNA with phenol:chloroform extraction and ethanol precipitation. In order to keep the Tn5:DNA ratio similar to a bulk ATAC-seq experiment with 50,000 cells, we estimated that a typical *C. elegans* nucleus will contain $1e^6$ bp $\times$ 2 genomes $\times$ 660 MW/bp $\times$ $1.67e^{-12}$ pg/MW $\approx$ 0.22 pg/nucleus, or $\sim$ 11 ng in 50,000 nuclei. We diluted the DNA to a concentration of $\sim$ 0.87 ng/µl, as measured with the Qubit High Sensitivity assay (Invitrogen), and used 11.5 µl as input to a 25 µl reaction with 12.5 µl of Nextera TD buffer and 1.0 µl of Nextera Tn5 enzyme. The reaction was incubated at 37°C for 30 min, cleaned up with a Qiagen MinElute column, and amplified using NEBNext 2x PCR Mix with primers from Buenrostro, et al. 2013.[35] The libraries were cleaned up with 1:1 AMPure XP magnetic beads, and sequenced on the Illumina NextSeq platform.

2.6.4 ATAC-seq processing pipeline

Initial processing of the sequencing results was done as reported in Cusanovich, et al. 2018,[40] with some changes. Sequencing results were converted to FASTQ format with the Illumina `bcl2fastq` program (v. 2.19). First, the integrity of the barcode sequences was checked for each of the four components of the barcode (tagmentation barcodes from both sides of the cut and the P5 and P7 primer indices, added during PCR amplification, total 36 bp) by matching the sequencing results to the known barcode sequences. Any read that had three or fewer edits compared to the best-matching known barcode sequence and that had no other known barcode sequences match-

ing with five or fewer edits were corrected and assigned to the best-matching barcode sequence. Any read-through of short templates was corrected by trimming adapter sequences from reads using Trimmomatic[83] (v0.36) with the options `ILLUMINACLIP:NexteraPE-PE:2:30:10:1:true`, `TRAILING:3`, `SLIDINGWINDOW:4:10`, and `MINLEN:20`. Next, read sequences were aligned to the WS230/ce10 build of the *C. elegans* genome with bowtie2[84] with options `-X 2000` and `-3 1`, properly paired reads with mapping scores greater than 10 were kept, any reads mapping to the mitochondrial DNA were filtered out, and read pairs with identical barcode sequence and identical starting and ending mapping coordinates were identified as PCR duplicates and collapsed to 1 using a custom script.[40] Next, read coverage for each cell was calculated and cell barcodes with fewer than 100 reads were removed from the cells from the MiSeq run, and cell barcodes with fewer than 150 reads were removed from the two NextSeq runs (Fig. A.2). The reads that made it through filtering for each batch of sequencing were merged into a single BAM file using the Picard MergeSamFiles program (http://broadinstitute.github.io/picard). Last, the reads in the merged BAM file were converted into cut sites by taking 60 bp intervals centered on the fragment ends, shifting those for reads mapping to the forward strand by +4 bp and the negative strand by -5 bp (to account for the shape of the Tn5 cut site[35]), and writing the resulting coordinates to a BED file for peak calling.

2.6.5 Peak calling to identify accessible regions

First, we did a rough estimate of peak locations by setting a cut site coverage threshold of 40 on 25 bp bins across the genome. This threshold provided a balance between finding as many enriched regions as possible while avoiding false positives, as estimated using our negative control bulk ATAC-seq results from naked *C. elegans* DNA (Fig. A.3). We then merged adjacent enriched 25 bp bins, and we used the resulting set of 49,319 regions as our initial peak set. We made a binary matrix indicating which cells had evidence for which peaks (i.e. one or more cut sites overlapping that peak). This binary matrix was used to train a Latent Dirichlet Allocation (LDA) model[54] with 60 topics. Next, we used the resulting cells-by-topics matrix to cluster the cells. We generated a 25

nearest neighbors graph in which nodes are cells using a KDTree (`sklearn.neighbors.KDTree` with euclidean distance metric) to identify the nearest neighbors, and then used the Leiden community detection algorithm[49] to find clusters of tightly connected cells in the graph. For each cluster we wrote a file containing the cut sites from each cell in that cluster, and then called peaks and summits with MACS2[72] (v. 2.1.1) with options `--format=BED`, `-g 9e7`, `--nomodel`, `--qvalue=0.05`, `--SPMR`, `--tsize=60`, `--bdg`, `--keep-dup all`, and `--call-summits`. Additionally, we provided as an input control a bulk ATAC-seq data set collected on naked *C. elegans* genomic DNA. MACS2 found 269,108 total peak regions in all clusters (min: 3,172 peaks in cluster 24; max: 23,027 in cluster 6; avg: 10,764 peaks per cluster), and this was reduced to a set of 32,486 distinct peaks after merging overlapping peaks identified in different clusters with `bedtools merge`.[85] Many peaks had multiple summits, and the summit patterns frequently were different in location or amplitude across clusters, even in shared peaks; so, in order to capture the information contained in these summit regions, we used the summit regions +/- 50 bp as the set of accessible regions for downstream data analysis. We merged the summit regions from all clusters, again using `bedtools merge`, to make a master list of summits containing 74,067 regions. We used this list to assemble a binary matrix identifying which summits are found in which cell, with a row for each cell and a column for each merged summit.

2.6.6 Overlapping peaks with other data sets

Supplementary data for figure 2 from Jänes, et al. 2018[65] was downloaded from the eLife website (filename: `janes2018_fig2_data1_v2.txt`). This file was parsed using Unix tools and `bedtools` (v. 2.25.0)[85] into three files: a file containing all of the peaks in BED format with overlapping sites merged (i.e. using `bedtools merge` with default parameters), a file containing promoter-annotated peaks (those annotated as "coding_promoter", "unassigned_promoter", or "pseudogene_promoter") with overlapping peaks merged, and a file containing enhancer-associated peaks (those annotated as "putative_enhancer") with overlapping peaks merged. These files were overlapped with the sciATAC-seq summits, and the sciATAC-seq summits were overlapped with

these files, using `bedtools intersect`.

Peak loci from the modERN project were downloaded from the EPIC website (http://epic.gs.washington.edu/modERN/) for reference WS245/ce11 using the "Download Aggregated Peaks" and "Download Clustered Peaks" buttons on the "Worm By LifeStage" tab of the user interface. These coordinates were converted to the WS230/ce10 reference using the UCSC liftOver command line tool (v. 357 from bioconda). Peaks were parsed into different files based on developmental stage, and any overlapping peak regions were merged in the final files. As above, these files were overlapped with the sciATAC-seq summits, and the sciATAC-seq summits were overlapped with these files using `bedtools intersect`.

2.6.7 Latent Dirichlet Allocation modeling

Inspired by the effectiveness of LDA as implemented in cisTopic,[54] we applied the same approach to analyzing our sciATAC-seq data. Briefly, LDA is a Bayesian modeling strategy that was originally developed in the setting of document classification. It assumes that each document is characterized by one or more latent "topics", and that these topics are characterized by subsets of the words in the document. The LDA model works by placing Dirichlet priors over the topics in each document and over the words in each topic (i.e. the probability of each document being associated with each topic sums to 1.0, and the probability of each topic being associated with each word sums to 1.0). Our implementation uses a collapsed Gibbs sampler to speed up training and convergence by sampling the latent parameters of the model from the full conditional posterior.[71] It does this by iterating over the entire vocabulary defined by the documents it is modeling and proposing a topic for every instance of every word in every document. The probability of picking a topic for a given word and document is computed based on the current probability distribution of topics for that document and the probability of words for each topic. At the end of training, the probability distributions for the topics over the documents and the words over the topics can be calculated by summing the topic proposals for all peaks, and for all documents, respectively. When applied to single-cell ATAC-seq data, as in cisTopic, cells are treated as the documents and peaks or peak

summits are treated as the words. The LDA model then learns topics that distinguish among the cells based on which peaks tend to be accessible in similar patterns across all cells, and outputs two matrices that capture the relationship between peaks and topics, and cells and topics: the first matrix contains the counts of the number of cells for which a given peak was assigned to each topic (the peaks-by-topics matrix), and the second matrix contains how many peaks from each cell were assigned to each topic (the cells-by-topics matrix).

We began by using cisTopic itself, but found the R implementation to take almost two days to process the full data set. In order to speed up the modeling, we implemented a parallelized version in Java that can split the training of a single model across multiple cores, reducing the run time to just a couple of hours. In the end, we used 60 topics and set the alpha parameter for the Dirichlet priors to 0.1 to concentrate the probability distributions into just a few peaks/topics. The Java code will be made available with the publication of this work in a scientific journal.

2.6.8 Latent Dirichlet Allocation model tuning

Choosing hyperparameter values is one of the most challenging aspects of training models like LDA. In particular, using an appropriate number of topics is critical to getting good results, and picking this number requires an empirical approach. In cisTopic,[54] the authors recommend training several model instances, each with a different number of topics, and choosing the number of topics that gives the best log likelihood of your input data. However, increasing the number of topics adds parameters to the model, which makes the model better able to fit the training data, even if it has already fit the true signal and begins to train on noise (i.e. it is overfitting). Since the cisTopic procedure uses the same data for training and evaluation, it does not test the generalizability of the model parameters, and cannot tell when the model starts to overfit. Ultimately, it will recommend using a higher number of topics than can be supported by the data. In order to identify a suitable number of topics that avoids overfitting, we implemented the following cross-validation procedure.

First, the cells are evenly and randomly split into five sets for five-fold cross validation. Then, for each number of topics that we would like to test, five LDA models are trained, with each model

training on four of the folds and holding one out for evaluation. Once each model is done training, it estimates the likelihood of the data in the held-out test fold with a Chib-style estimator.[86] In this estimation procedure, the peak-topic probabilities learned from the training data are fixed, and then a cell-topic vector is trained for each held out cell based on the fixed peak-topic probabilities. The log likelihood of each held out cell is estimated based on sampling from the posterior of the model trained on that held out cell. We convert these log likelihoods to perplexity, which is defined as

$$\text{perplexity}(w) = \exp\left(-\frac{\mathscr{L}(w|\theta)}{N}\right)$$

where w is a held out test cell, $\mathscr{L}(w|\theta)$ is the log likelihood of that test cell given the LDA model, and N is the number of peaks found in that cell. Because perplexity is inversely related to the log likelihood, smaller values are better. The best number of topics to use is the one that produces the lowest mean perplexity from the held out data. This evaluation procedure proved very effective for analyzing muscle and neuron subtypes (see Figs 2.6, A.8, and Figs 2.7, A.10, respectively).

2.6.9 Cell clustering by topics

To visualize the relationship between cells and topics after LDA training, we used UMAP[47] to reduce the dimensions of the cells-by-topics matrix to 2 from 60 (Fig. 2.2b). We first row-normalized the counts in the cells-by-topics matrix with the L2-norm and then used the Python implementation of UMAP (`umap-learn`, v. 0.3.8) with default parameters, and plotted the cells as a scatter plot based on their coordinates in 2D UMAP space.

Next, we devised a procedure for clustering cells based on the LDA topics. Our goal was to find dense clusters of cells that were specific to particular topics. To find topic-specific cells, we normalized the cells-by-topics counts to the row sums, which converts the topic counts for each cell to topic probabilities. The probabilities are useful as a measure of topic-specificity because a high probability (i.e. greater than 0.5) indicates that no other topic can have more weight; and furthermore, the higher the maximum topic probability is for a cell, the more specific that cell is for that topic.

Additionally, in order to find which topics are associated with dense, coherent groups of cells, for each topic, we ranked the cells by their probability for that topic, and calculated a centroid for the top 50 cells by taking the mean value of the topic probability distributions for those cells. Then we calculated the dot product similarity of each cell to the centroid and averaged these similarities to get a density score for the topic. Then we ranked the topics by this density measure and selected the top 36 topics (those with average centroid similarity greater than 0.2).

The last step was to match topic-specific cells with the topics forming coherent clusters. To do this matching, we simply assigned any cell with a maximum cell-topic probability greater than 0.5 for one of the 36 coherent-cluster topics to that topic's cluster. Six of the coherent topics had no cells with a probability greater than 0.5, leaving us with 30 topic clusters. We filtered out an additional 4 topic clusters after examining them in the two dimensional UMAP and determining that they were either composed of only a few cells that were really part of a larger cluster or did not appear to be a single coherent cluster.

Thus, we settled on 26 topic clusters. Eight of the 26 clusters had fewer than 100 cells that met the topic-specificity criterion, even though there were other cells nearby in the UMAP plot that were visually also part of the cluster. In order to include these "overlooked" cells, we relaxed the topic specificity criterion and expanded the eight clusters by taking their centroid in a 10-dimensional UMAP space and adding 115-250 of the nearest neighbors to the cluster (as long as they weren't already assigned to a different cluster). The exact number of cells added to each expanded cluster depended on checking the coverage of the visual cluster in UMAP space. In the end, we assigned 18,134 cells to topic clusters. Supplementary figure A.7 shows the signal tracks and peaks that we generated by aggregating the signal in each cluster of cells to make synthetic bulk ATAC-seq data tracks. Each track exhibits highly topic-specific signal at accessible sites near genes with known tissue-specific expression patterns.

2.6.10 Cell-by-topic and summit-by-topic heatmaps

To generate the heatmaps as in Fig. 2.2c and Fig. 2.5, we first mean-centered the rows of the cell-by-topic or the summits-by-topic count matrix and then row-normalized by the L2-norm. Then the rows and columns were hierarchically clustered in Python (v. 3.6.7) with the `scipy` module (v. 1.2.1). We used the `scipy.spatial.distance.pdist` function with the `cosine` metric to compute the pairwise distance matrix, then `scipy.cluster.hierarchy.linkage` with method `average` to generate clusters, and finally `scipy.cluster.hierarchy.dendrogram` to order the rows and columns after clustering. The clusters were classified by tissue type based on the dominant tissue type for each topic in Fig. 2.4. The cell/summit coverage information was calculated as the $\log_2$-transformed sum of peaks found per cell for the cell-by-topic matrix or the $\log_2$-transformed sum of cells exhibiting a summit for the summit-by-topic matrix.

2.6.11 Identifying tissue-specific topics

We identified topic tissue-specificity in two ways: by overlapping summits from each topic with peaks from ChIP-seq of cell type-specific factors (Fig. 2.3), and by assigning summits to the nearest gene and assessing which tissues those genes tend to be expressed in (Fig. 2.4).

To compare the overlap of topic-associated summits with TF ChIP-seq peaks, for each topic we wrote a BED file that contained the coordinates of any sciATAC-seq summit with non-zero probability for that topic based on the LDA model. Then, from the list of ChIP-seq peaks that we downloaded from the modERN project, we created separate BED files for any peaks from ChIP-seq experiments for *hlh-1*, *elt-1*, or *elt-2*, filtered out any peaks overlapping peaks from more than 40 other ChIP experiments (i.e. high occupancy target, or HOT, sites), and merged any remaining overlapping peaks with `bedtools merge`. We intersected the summits from each topic with the peaks from each transcription factor and recorded the number of overlapping summits for each topic. In order to understand whether the observed overlaps per topic were surprising, we generated a null distribution by sampling a number of summits equal to the number of observed overlaps, with each summit being drawn from a particular topic with a probability based on the total number of

summits associated with that topic. We then took the $\log_2$ ratio of the topic distribution of observed overlaps to the topic distribution of each sample of randomly drawn summits. We plot the mean $\log_2$ ratio, and use the samples to compute a 95% confidence interval around each bar.

Our approach for comparing the overlap of the topic-associated summits with the sciRNA-seq gene expression data was similar to that for comparing to the cell type-specific TF ChIP-seq data. We began by writing a BED file for each topic containing the coordinates of all summits with non-zero probability for that topic. We also included in the file the topic-specificity of each summit, which we define as the fractional value obtained for a particular summit and topic after normalizing the summits-by-topics matrix by the row sums (i.e. the fraction of a summit's total probability across all topics that is due to a particular topic). Next, we associated the summits with their nearest downstream gene (within 1200 bp) (using `bedtools closest` with options `-D b`, `-io`, and `-id`), ranked the summits by topic-specificity, and took the top 500 unique genes associated with those ranked summits. Similarly to the cell type-specific TF ChIP-seq analysis above, we then wanted to ask whether the expression distribution of these top genes across tissues is enriched on average for particular tissues. So, we drew 100 samples of 500 genes from the null distribution of all genes with sciRNA-seq data, and compared the expression distribution of these gene sets with our 500 topic-specific genes by computing the $\log_2$ ratio of the mean topic-specific tissue expression distribution to the mean tissue expression distribution of each random sample. We again reported the mean $\log_2$ ratio and computed the 95% confidence interval around the mean.

2.6.12 Generating UCSC Genome Browser tracks for the topic clusters

In order to visualize the chromatin accessibility signal from the clusters of cells defined by their topic enrichment (Fig. 2.2b), we first combined the 60 bp cut site coordinates from all cells in each cluster into a single BED file per cluster. This BED file was input to MACS2[72] (v. 2.1.1) with the same options used in the initial data processing pipeline, including using the naked genomic DNA bulk ATAC-seq data as an input control. The input control signal was subtracted from the treat-

ment signal using `macs2 bdgcmp -m subtract`, and the resulting BEDGraph file was converged to BigWig format using the `bedGraphToBigWig` utility from UCSC.[87] These BigWig tracks are displayed along with the peaks called by MACS2 on the aggregated reads from each cluster of cells and the original summit regions used for generating the clusters, colored by their specificity for each topic.

2.6.13 Identifying tissue sub-types using marker genes

In order to identify tissue sub-types for muscle (topic 46) and neuron (topics 11, 12, 14, 18, 19, 21, 25, 30, and 39), we trained a new LDA model for each tissue. First, we used 5-fold cross validation and the Chib-style estimator described above to find a suitable number of topics. For the muscle cells we tested 2, 5, 10, 15, 20, 25, 30, 35, 40, and 60 topics, and for the neuron cells we tested 5, 10, 20, 30, 40, 50, 75, and 100 topics. The α and β hyperparameters were both set to 0.1, we trained each model with 1500 iterations total with the first 750 as burn-in, and evaluated with 1000 iterations and 200 as burn-in. The results showed that 5 topics was appropriate for the muscle cells (Fig. A.8), and 20 for the neuron cells (Fig. A.10). Next, we trained another LDA model on all of the cells at once (i.e. no held out data) with the appropriate number of topics, α and β set to 0.1, and 5000 iterations total with 2500 burn-in iterations.

The cell-by-topic matrix from the final trained model was read in, the data for each cell were L2-normalized, and the cells were projected into two dimensions by UMAP with default parameters and visualized as a scatter plot. Next, each marker gene was checked for nearby chromatin accessibility (i.e. a called peak) that was no more than 1200 bp upstream from the 5' end of the gene. Last, cells in the heatmap are colored based on whether they have a read overlapping a peak for one of the marker genes.

The neuron cells were further analyzed based on the neuron subtypes defined in the L2 scRNA-seq results.[30] In this analysis, for each neuron subtype, genes were ranked by the proportion of their expression coming from that subtype. Then, starting with the gene with the highest proportion (i.e. the most specific gene), each was checked for a nearby chromatin accessibility peak within

1200 bp of the 5' end of the gene. If no peak was present, then the algorithm continued with the gene that had the next-highest neuron subtype proportion. When a peak was found, its most probable topic based on the peaks-by-topics matrix was checked. If the most probable topic was 16 (a topic that had very little ability to distinguish among clusters of cells), then the gene was skipped and the algorithm continued down the list. Otherwise, the cell probabilities for the top topic were retrieved from the cells-by-topics matrix and added as a new column to a matrix. Once the number of columns in that matrix grew to 50 (i.e. the algorithm found the 50 most specific genes with chromatin accessibility nearby and weight in a topic other than 16), then the rows of the matrix were averaged to obtain the mean weight for topics associated with subtype-specific genes, and the cells in the scatter plot were colored based on these values.

2.7 ACKNOWLEDGEMENTS

We would like to thank Anh Leith for sharing her expertise in FACS and helping us to collect samples and run the sciATAC protocol, and Chau Huynh for guidance when developing the worm nuclei isolation protocol and when optimizing our early experiments on bulk ATAC-seq. Olubusayo Bolonduro, as a summer intern in the Waterston Lab, also helped optimize conditions for ATAC-seq in worm and performed the input control experiment to collect ATAC-seq data from naked *C. elegans* genomic DNA. We would like to thank Illumina for providing the indexed Tn5, and also PCR reagents for the preparation of some of our sciATAC-seq libraries. This work was funded by National Institutes of Health awards U41 HG007355 and R01 GM072675. Jay Shendure is a Howard Hughes Medical Institute Investigator.

Chapter 3: PREDICTD: PARALLEL EPIGENOMICS DATA IMPUTATION WITH CLOUD-BASED TENSOR DECOMPOSITION

This chapter is adapted with minimal modification from:

Durham, Timothy J., Maxwell W. Libbrecht, J. Jeffry Howbert, Jeff Bilmes, and William Stafford Noble. PREDICTD PaRallel Epigenomics Data Imputation with Cloud-Based Tensor Decomposition. Nature Communications 9, no. 1 (April 11, 2018): 1402. https://doi.org/10.1038/s41467-018-03635-9.

3.1 AUTHOR CONTRIBUTIONS

WSN, JJH, and JB conceived the tensor decomposition approach for data imputation. TJD implemented PREDICTD, executed all analyses, including designing and executing the model evaluation and non-coding human accelerated region analyses, and wrote the manuscript. WSN, JB, MWL, and JJH provided essential input on the mathematical and machine learning components of the project, along with additional critical feedback throughout the publication and review process. All authors read and approved the final manuscript.

3.2 ABSTRACT

The Encyclopedia of DNA Elements (ENCODE) and the Roadmap Epigenomics Project seek to characterize the epigenome in diverse cell types using assays that identify, for example, genomic regions with modified histones or accessible chromatin. These efforts have produced thousands of data sets but cannot possibly measure each epigenomic factor in all cell types. To address this challenge, we present a method, PaRallel Epigenomics Data Imputation with Cloud-based Tensor Decomposition (PREDICTD), to computationally impute missing experiments. PREDICTD leverages an elegant model called "tensor decomposition" to impute many experiments simultaneously. Compared with the current state-of-the-art method, ChromImpute, PREDICTD produces

lower overall mean squared error, and combining the two methods yields further improvement. We show that PREDICTD data capture enhancer activity at non-coding human accelerated regions. PREDICTD provides reference imputed data and open-source software for investigating new cell types, and demonstrates the utility of tensor decomposition and cloud computing, both promising technologies for bioinformatics.

3.3 INTRODUCTION

Understanding how the genome is interpreted by varied cell types in different developmental and environmental contexts is a key question in biology. With the advent of high throughput next generation sequencing technologies, over the past decade we have witnessed an explosion in the number of assays to characterize the epigenome and interrogate chromatin state genome-wide. Assays to measure chromatin accessibility (DNase-seq, ATAC-seq, FAIRE-seq), DNA methylation (RRBS, WGBS), and histone modification and transcription factor binding (ChIP-seq) have been leveraged in large projects such as the Encyclopedia of DNA Elements (ENCODE)[6] and the Roadmap Epigenomics Project[7] to characterize patterns of biochemical activity across the genome in many different cell types and developmental stages. These projects have produced thousands of genome-wide data sets, and studies leveraging these data sets have provided insight into multiple aspects of genome regulation, including mapping different classes of genomic elements,[22,24] inferring gene regulatory networks,[20] and providing insights into possible disease-causing mutations identified in genome-wide association studies.[7]

Despite the progress made by these efforts to map the epigenome, much work remains to be done. Due to time and funding constraints, data have been collected for only a fraction of the possible pairs of cell types and assays defined in these projects (Fig. 3.1a). Furthermore, taking into account all possible developmental stages and environmental conditions, the number of possible human cell types is nearly infinite, and it is clear that we will never be able to collect data for all cell type/assay pairs. However, understanding the epigenome is not an intractable problem because in reality many of the assays detect overlapping signals such that most of the unique information

can be recovered from just a subset of experiments. One solution is thus to prioritize experiments for new cell types based on analysis of existing data.[88] Alternatively, one may exploit existing data to accurately impute the results of missing experiments.

Ernst and Kellis pioneered this imputation approach, and they achieved remarkable accuracy with their method, ChromImpute.[44] Briefly, this method imputes data for a particular target assay in a particular target cell type by 1) finding the top ten cell types most correlated with the target cell type based on data from non-target assays, 2) extracting features from the data for the target assay from the top ten non-target cell types, and also extracting features from the data for non-target assays in the target cell type, and 3) training a regression tree for each of the top ten most correlated cell types. Data points along the genome are imputed as the mean predicted value from the collection of trained regression trees. Although ChromImpute produces highly accurate imputed data, this training scheme is complicated and not very intuitive, and results in a fragmented model of the epigenome that is very difficult to interpret. We hypothesized that an alternative approach, in which a single joint model learns to impute all experiments at once, would simplify model training and improve interpretability while maintaining accurate imputation of missing data.

Accordingly, we present PaRallel Epigenomics Data Imputation using Cloud-based Tensor Decomposition (PREDICTD), which treats the imputation problem as a tensor completion task and employs a parallelized algorithm based on the PARAFAC/CANDECOMP method.[61,89] Our implementation, developed on consumer cloud infrastructure, achieves high-accuracy imputation of ENCODE and Roadmap Epigenomics data, and predicts all data sets jointly in a single model. We used PREDICTD to impute the results for 3048 experiments across 127 cell types and 24 assays from the Roadmap Epigenomics project, and these imputed data are available for download through ENCODE (https://www.encodeproject.org/). In the following sections we explain the model, discuss its performance on held-out experiments from the Roadmap Epigenomics Consolidated data,[7] show that the model parameters summarize biologically relevant features in the data, and demonstrate that imputed data can recapitulate important cell type-specific gene regulatory signals in non-coding human accelerated regions of the genome.[90]

3.4 RESULTS

3.4.1 Epigenomic maps can be imputed using tensor factorization

Data from the Roadmap and ENCODE projects can be organized into a three-dimensional tensor, with axes corresponding to cell types, assays, and genomic positions (Fig. 3.1b). This tensor is long and skinny, with many fewer cell types and assays than genomic positions, and the data for experiments that have not been done yet are missing in the tensor fibers along the genome dimension. Our strategy for imputing these fibers is to jointly learn three factor matrices that can be combined mathematically to produce a complete tensor that both approximates the observed data and predicts the missing data. These three factor matrices are of shape $C \times L$, $A \times L$, and $G \times L$, where C, A, and G indicate the numbers of cell types, assays, and genomic positions, respectively, and L indicates the number of "latent factors" that the model trains (Fig. 3.1b), and thus the number of model parameters.

We developed and trained our implementation of this tensor factorization model, PREDICTD, using 1014 data sets from the Roadmap Epigenomics Consolidated Epigenomes[7] (Fig. 3.1a). To assess model performance, we split the data sets into five training/test splits, and we report on the results of imputing each test set at 25 base pair resolution. The model training proceeds by distributing the data and genome parameters across the nodes of the cluster, and then sharing the cell type and assay parameters across all nodes using a parallelized training procedure (See Methods, Fig. 3.1c). We find that training on a randomly selected 0.01% of the genome provides enough signal for learning the cell type and assay parameters; these parameters are then applied across all genomic positions of interest by training the genome parameters for each position while holding the cell type and assay parameters constant. We report results from imputing just over 1% of the genome, including the ENCODE Pilot Regions[6] and 2,640 non-coding human accelerated regions.[90] All subsequent references to the genome dimension in this manuscript refer to this subset of loci.

Our model formulation and implementation offer several important advantages. First, training a single model to impute all data sets at once is a straightforward and intuitive way of solving this

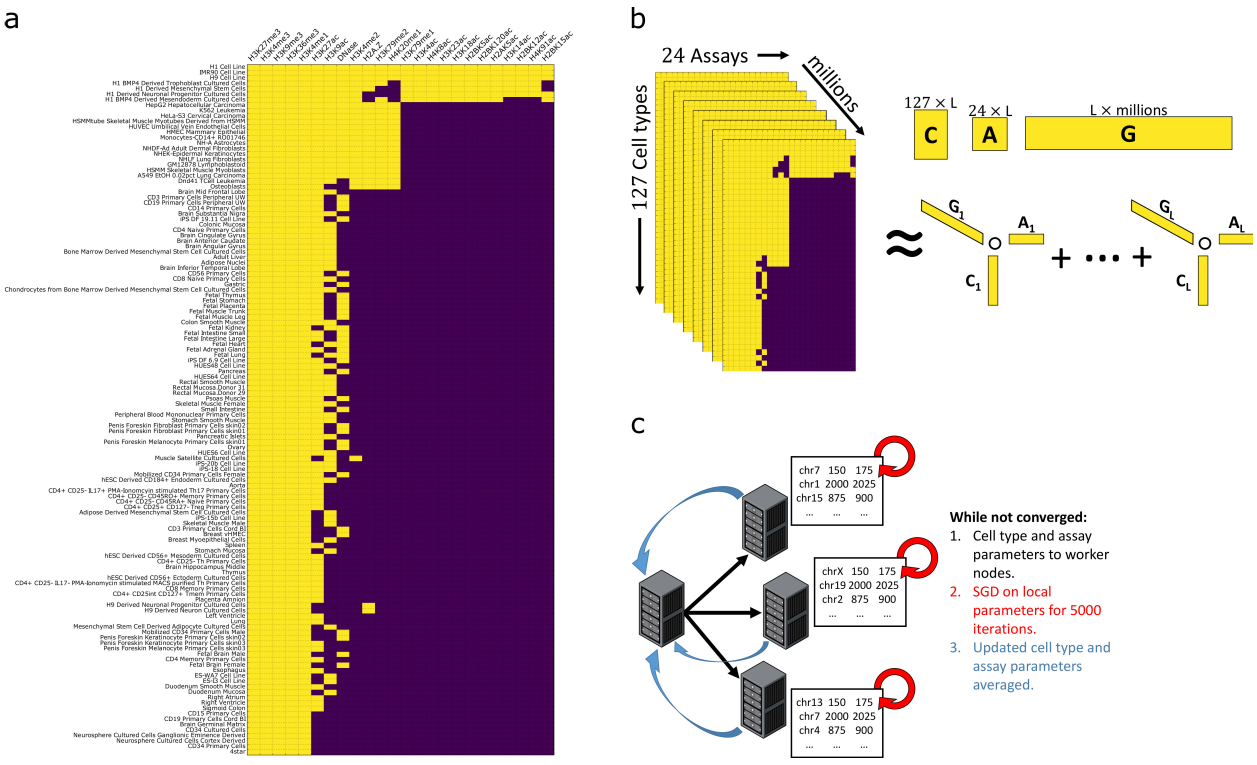

Figure 3.1: **Overview. a** Matrix representing the subset of the Roadmap Epigenomics consolidated data set used in this study. Experiments in yellow have observed data, while missing experiments are purple. **b.** We model the experiments in **a.** as a three-dimensional tensor, and find three low-rank factor matrices (denoted C, A, and G) that can be combined by summing the outer products of each of the L latent factor vector triplets to reconstruct a complete tensor with no missing values that both approximates the existing data and imputes the missing data. **c.** The genome dimension is very large, so in order to fit all of the data in memory and to speed up training, we distribute the tensor across multiple cluster nodes running Apache Spark. Then we use parallel stochastic gradient descent[91] to share the A and C matrices across all nodes.

problem. Second, as we demonstrate below, the model can leverage the joint training to perform well even on cell types with a single informative experiment. Third, the parameters of the trained model have the same semantics across all input data sets and, although a full investigation of model interpretability is outside the scope of this work, we show that the trained parameters show different patterns for different cell types, assays, and genomic elements. We take these results as evidence that the PREDICTD model itself holds the potential to be interrogated to learn about relationships among assays, cell types, and genomic loci. Last, PREDICTD software is open source (https://bitbucket.org/noblelab/predictd), and is also implemented and distributed on the consumer cloud, which makes our model immediately accessible to and easily runnable by nearly anyone.

3.4.2 PREDICTD imputes epigenomics experiments with high accuracy

PREDICTD imputes missing data with high accuracy based on both visual inspection and quality measures (Fig. 3.2). Visually, the imputed signal pattern closely matches that of observed data, and recapitulates the known associations of epigenomic marks with genomic features (Fig. 3.2a, Supplementary Fig. B.1). For example, as expected, H3K4me3 imputed signal is strongly enriched in narrow peaks at promoter regions near the transcription start site of active genes, and H3K36me3, known to mark transcribing regions, is enriched over gene bodies.

We also show strong performance of PREDICTD on ten different quality measures (see Methods, Supplementary Figs. B.2, B.14, B.15, Supplementary Data 13-16), especially the global mean squared error quality measure (MSEglobal). As a key part of the PREDICTD model's objective function, MSEglobal is explicitly optimized during model training (see Methods). The MSEglobal measure has a mean of 0.1229, and it ranges from 0.0359 for H3K4me3 in the "NHLF Lung Fibroblasts" cell type to 0.4511 for H4K20me1 in "Monocytes CD14+ RO01746". Other key quality measures include the genome-wide Pearson correlation (GWcorr, mean: 0.6886, min: 0.0790 for H3K36me3 in "Right Atrium", max: 0.9391 for H3K4me3 in "HUES64 Cell Line"), and the area under the receiver operating characteristic curve for recovering observed peak regions from im-

puted data (CatchPeakObs, mean: 0.9565, min: 0.5503 for H3K36me3 in "Right Atrium", max: 0.9984 for H3K4me3 in "NHLF Lung Fibroblasts"). Note that seven of our ten quality measures, including GWcorr and CatchPeakObs, were also used in the ChromImpute publication.[44]

As a baseline, we compared the performance of PREDICTD to a simple "Main Effects" model, which computes the global mean of the observed data, and then the column and row residuals of each two-dimensional slice of the tensor along the genome dimension, and imputes a given cell in the tensor by summing the global mean and the corresponding row and column residual means. PREDICTD outperforms this baseline model for MSEglobal on all but two assays (Fig. 3.2b). Furthermore, PREDICTD similarly outperforms the Main Effects on all additional performance measures (Supplementary Fig. B.2).

3.4.3 PREDICTD performs well on cell types with few assays

A key application of PREDICTD will be to impute results for cell types that may have only one or two data sets available. To investigate the performance of PREDICTD in this context, we trained a model on all available data for all cell types, except that we only included one or two experiments for the "CD3 Cord Blood Primary Cells" cell type. In particular, one model had just H3K4me1 in the training set for this cell type, one had just H3K4me3, and one had both H3K4me3 and H3K9me3. Comparing the performance measures between these experiments and the imputed results from our original models trained on the five test sets, we find that the results of training with just H3K4me3 or both H3K4me3 and H3K9me3 are nearly as good as (and sometimes better than) the results from the original models with training data that included five or six experiments for this cell type (Fig. 3.2c, Supplementary Fig. B.3). Imputing only based on H3K4me1 signal did not perform as well as imputing based on only H3K4me3. This observation is consistent with previous results on assay prioritization[88] indicating that H3K4me3 is the most information-rich assay. Furthermore, this result is not specific to the "CD3 Cord Blood Primary Cells" cell type. We find that the results for imputing four other cell types ("GM12878 Lymphoblastoid", "Fetal Muscle Trunk", "Brain Anterior Caudate", and "Lung") just based on H3K4me3 signal showed

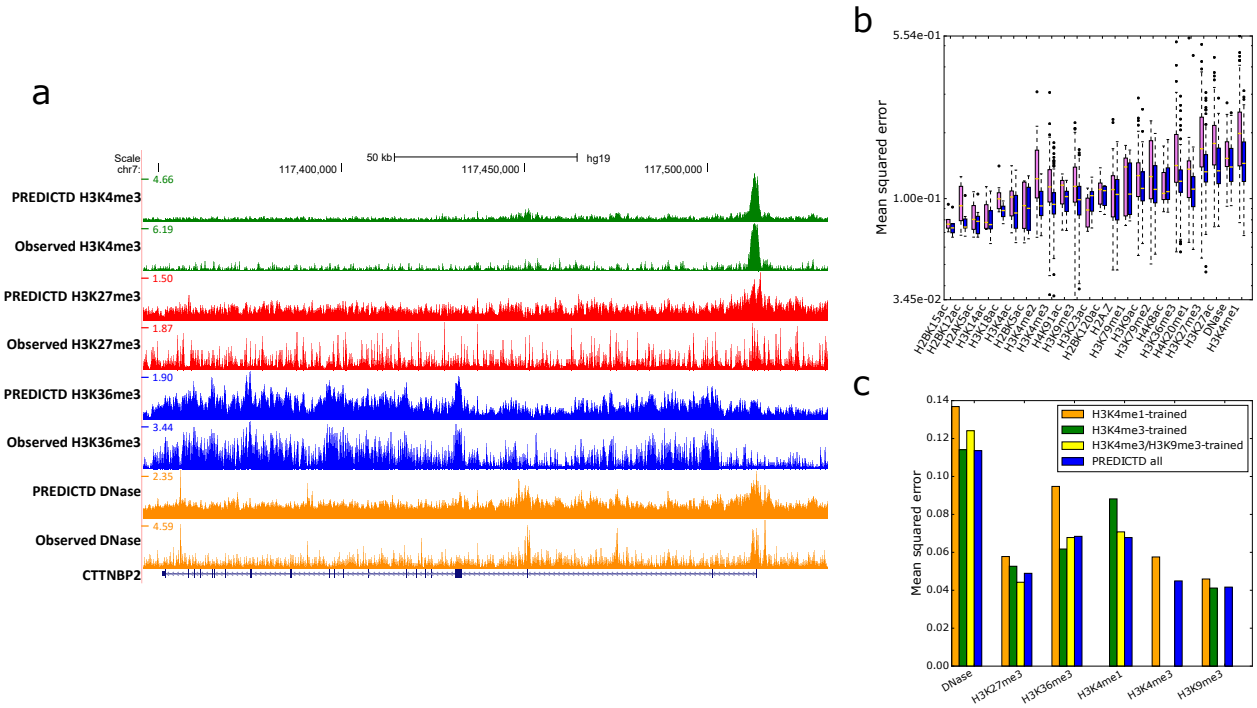

Figure 3.2: **PREDICTD imputes missing epigenomics data with high accuracy. a.** Comparison of PREDICTD and observed data in H1 embryonic stem cells, which is one of three cell types with observed data for all assays. For each assay, the top track is PREDICTD and the bottom is observed. See Supplementary Fig. B.1 for more assays. **b.** Global mean squared error (MSEglobal) for each data set, shown as box plots for the baseline Main Effects model (pink), and PREDICTD (blue). For each distribution, the box shows the inter-quartile range (IQR), whiskers show 1.5 times the IQR, and flier points show individual data sets that are outliers. The median is indicated by a horizontal gold line on each box plot. See Supplementary Fig. B.2 for more quality measures. **c.** Simulated analysis of a cell type with few available data sets. The bar graph shows the MSEglobal measure for the six available assays for the "CD3 Primary Cells from Cord Blood" cell type on four models. In blue is the full PREDICTD model shown in **a.** and **b.**, while the other bars represent models trained with all observed data for other cell types, but only H3K4me1 (orange), only H3K4me3 (green), or only H3K4me3 and H3K9me3 (yellow) for "CD3 Primary Cells from Cord Blood". See Supplementary Fig. B.3 for more quality measures.

similar results (Supplementary Figs. B.5, B.6, B.7, B.8). We conclude that PREDICTD performs well on under-characterized cell types and will be useful for studying new cell types for which few data sets are currently available.

3.4.4 Model parameters capture patterns in each tensor dimension

The fact that PREDICTD performs well on the imputation task implies that the parameters learned by the model captures patterns that can distinguish among different cell types, assays, and genomic positions, and we next present results showing that this is the case. We think it important to note that it would be incorrect at this point to interpret any particular latent factor as having a specific biological meaning. We place no *a priori* constraints on what patterns in the data PREDICTD uses to arrive at a solution, and any signal with relevance to a particular biological feature is likely distributed across multiple latent factors. As such, here we simply show that the parameters, in aggregate, exhibit different patterns between different cell types, assays, and genomic loci; a full investigation of the ways to gain biological insight from these parameters is outside the scope of our present study.

Although we cannot definitively assign semantics to individual latent factors, we find that their values in aggregate show patterns that recover known relationships among the cell types, assays and genomic loci (Fig. 3.3). Hierarchical clustering on the rows of the cell type factor matrix shows that similar cell types are grouped together (Fig. 3.3a), producing large clades for embryonic stem cells (magenta), immune cell types (green), and brain tissues (cyan), among others (Fig. 3.3a, Supplementary Fig. B.9). In the same way, assays with similar targets cluster together (Fig. 3.3b), with the colored clades from top to bottom representing acetylation marks generally associated with active promoters (magenta), marks strongly associated with active regulatory regions (cyan/blue), and broad marks for transcription (red) and repression (green). The assays cluster perfectly except that, biologically, H3K23ac should be grouped with either the active regulatory marks (cyan/blue) or the active acetylation marks (magenta). This is one of the two assays for which PREDICTD failed to outperform the Main Effects, and it was one of the worst performing assays for ChromImpute as

well, so it appears to be a difficult mark to impute. Nevertheless, most of the cell types and assays cluster correctly, and these results are highly non-random. We quantified this by comparing our clustering results to randomly shuffled cluster identities using the Calinski-Harabaz Index, which assesses how well the clustering separates the data by comparing the average distance among points between clusters to the average distance among points within clusters (Supplementary Fig. B.10).

For the genome factor matrix, we projected the coordinates of each gene from the GENCODE v19 human genome annotation[15] (https://www.gencodegenes.org/releases/19.html) onto an idealized gene model that includes nine parts from 5' to 3' in the gene: the promoter, 5' untranslated region (UTR), first exon, first intron, middle exons, middle introns, last intron, last exon, and 3' UTR. This procedure produced a summary of the genome latent factors (Fig. 3.3c) that, when reading each column of the heat map as a feature vector for a particular location in a gene, shows distinct patterns at different gene components. For example, latent factors that on average have high or low values at regions (i.e. heat map columns) near the transcription start site are different from those with high or low values at other gene components, like exons and introns.

In addition to investigating patterns in the genome parameters at genes, we checked to see whether distal regulatory regions showed a pattern distinct from gene components. P300 is a chromatin regulator that associates with active enhancers.[92] We therefore searched for patterns in the genome latent factors at windows +/- 1 kb around annotated P300 sites from published ENCODE tracks (see Methods). Note that no P300 data was used to train PREDICTD. Nevertheless, we find a striking pattern, with many latent factors showing average values of larger magnitude within the 400 bp region surrounding the center of the peak, and some others showing larger average magnitude in a flanking pattern in the bins 200-400 bp away from the peak center (Fig. 3.3c,d). Again, note that these results do not imply a biological meaning for any particular latent factor; instead, we hypothesize that the genome latent factors as a whole might be useful as features for classification or deeper characterization of genomic elements. Last, if we randomize the latent factors at each genomic location and do these same analyses we find no discernible pattern (Supplementary Fig. B.11). We thus conclude that the trained model parameters encode patterns that correspond to biology.

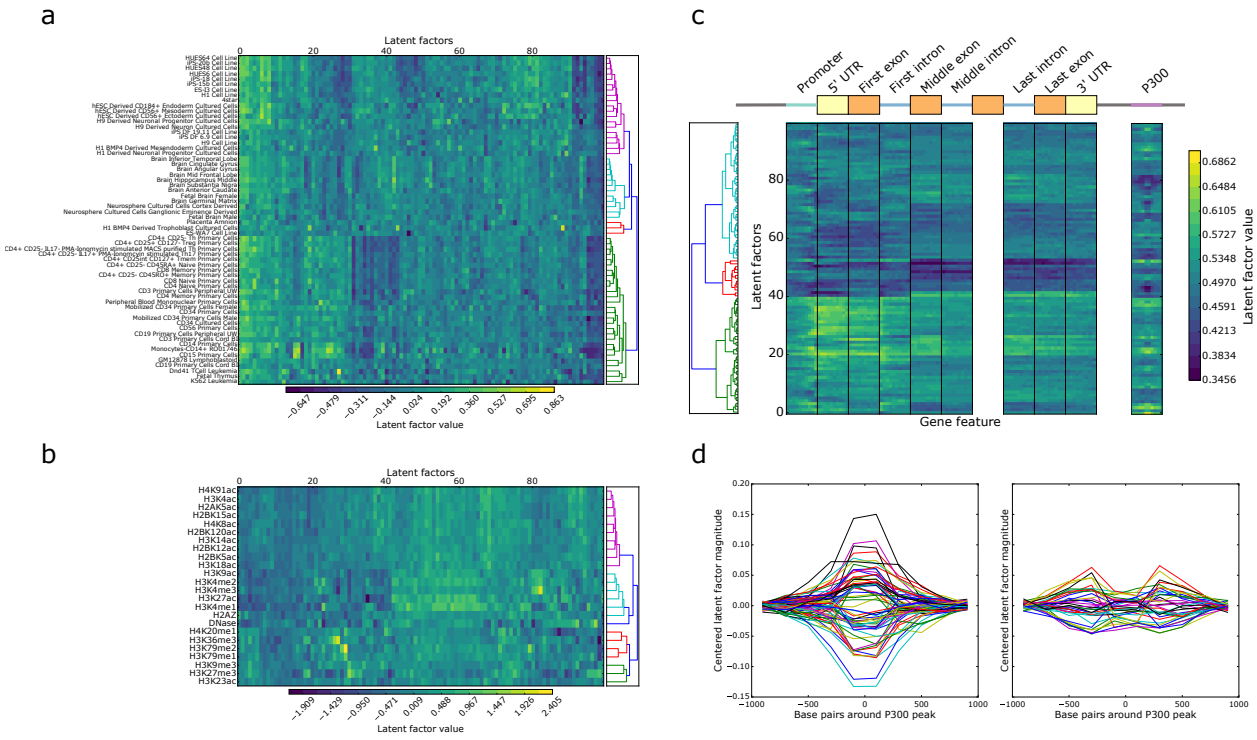

Figure 3.3: **The model parameters can distinguish among elements in each tensor dimension.** Plots show the values (or average values) for the 100 latent factors from one of the models trained for this manuscript, and that these values show patterns that distinguish among cell types, assays, and genomic elements. **a.** Hierarchical clustering of the cell types based on cell type factor matrix values shows that similar cell types tend to cluster together. This is a subset of cell types; for a clustering of all cell types see Supplementary Fig. B.9. **b.** Hierarchical clustering of the assays based on assay factor matrix values shows that similar assays tend to cluster together. **c.** Average values from the genome factor matrix show different patterns at different parts of the gene and P300 peaks called from ENCODE data. **d.** Average values from the genome factor matrix for each latent factor plotted as a line spanning the region +/- 1kb around the center of P300 peaks. Parameter values are centered at zero and plotted based on whether they show the highest magnitude at the peak (left, 64 latent factors) or flanking the peak (right, 36 latent factors).

3.4.5 PREDICTD and ChromImpute data are similar and complementary

As described in the Introduction, the ChromImpute method[44] provides high quality imputed data but employs a complicated model and training procedure tuned to each individual experiment. In contrast, our tensor decomposition approach imputes all missing experiments by using a single model, which we argue is conceptually simpler and addresses the problem in a more natural way. Furthermore, we find that our model outperforms ChromImpute on our primary performance measure (MSEglobal), and yields similar performance on nine additional measures (Fig. 3.4a, Table 3.1, Supplementary Figs. B.14, B.15, and Supplementary Data 1,13-16). Also see Supplementary Figs. B.4, B.2, B.3, B.5, B.6, B.7, B.8 for figures similar to those in Fig. 3.2 but with ChromImpute values included. The correlation of quality measures between PREDICTD and ChromImpute is higher than the correlation between the Main Effects method and ChromImpute, indicating that PREDICTD agrees with ChromImpute more often than Main Effects does. Furthermore, the mean log ratio of quality measures on corresponding experiments imputed by PREDICTD and ChromImpute show smaller differences than the log ratios for Main Effects and ChromImpute (Table 3.1, Supplementary Data 1, Fig. 3.4, Supplementary Figs. B.14, B.15). Thus, PREDICTD produces high quality imputed data that is almost as good, or better than, ChromImpute predictions, depending upon which quality measure is employed.

Table 3.1: **Statistics comparing models across five quality measures show PREDICTD outperforms Main Effects and has similar performance to ChromImpute.** See Supplementary Data 1 for the statistics on all quality measures.

Measure	**PREDICTD vs ChromImpute**			**PREDICTD vs Main Effects**			**Main Effects vs ChromImpute**		
	corr	**log ratio**		**corr**	**log ratio**		**corr**	**log ratio**	
		mean	**std**		**mean**	**std**		**mean**	**std**
MSEglobal	0.689	-0.151	0.266	0.835	-0.188	0.212	0.510	0.037	0.373
GWcorr	0.977	-0.039	0.072	0.883	0.100	0.163	0.866	-0.139	0.164
Catch1obs	0.979	-0.028	0.055	0.916	0.097	0.161	0.886	-0.125	0.177
Catch1imp	0.973	-0.023	0.073	0.876	0.155	0.293	0.848	-0.178	0.306
CatchPeakObs	0.923	-0.008	0.017	0.776	0.025	0.037	0.812	-0.032	0.035

We also calculated the distribution of the differences between imputed values and observed values for experiments imputed by both PREDICTD and ChromImpute, and we found that ChromImpute tends to impute higher values than PREDICTD (Supplementary Fig. B.13). We hypothesized that the two models each perform better on different parts of the genome, and so we tried averaging the PREDICTD and ChromImpute results. By the MSEglobal measure, we do see a marked improvement relative to both models, and other quality measures on which ChromImpute out-performed PREDICTD alone show parity between ChromImpute and the averaged model. (Fig. 3.4b).

3.4.6 Imputed data recovers cell type-specific enhancer signatures

Human accelerated regions (HARs) are genomic loci that are highly conserved across mammals but harbor more mutations in human than would be expected for their level of conservation (reviewed in[93]). Although some HARs overlap coding regions, the overwhelming majority (>90%) are found in non-coding portions of the genome (non-coding human accelerated regions, or ncHARs),[90,93] and ncHARs are thought to be enriched for mutations that affect the regulation of genes underlying human-specific traits. Non-coding variation is thought to account for much of our phenotypic divergence from other primates,[94] and additional evidence in support of this hypothesis comes from observations that ncHARs cluster around developmental and transcription factor genes,[90,93] transgenic assays for functional validation of enhancer activity,[90,95–97] and computational epigenomics and population genetics studies.[90,98,99]

In particular, ncHARs are enriched in developmental enhancer activity.[90,98] In Capra et al. 2013,[90] EnhancerFinder,[98] a program to predict genomic regions with tissue-specific developmental enhancer activity, was trained on ENCODE epigenomics maps[6] and results from the VISTA enhancer database,[100] and applied to ncHARs. EnhancerFinder predicted enhancer activity for 773 of 2649 ncHARs, but the authors note that the characterization of these regions remains incomplete due to limitations in the available data. To our knowledge, no one has yet analyzed enhancer signatures of ncHARs in the context of the Epigenomics Roadmap data. Thus, we ad-

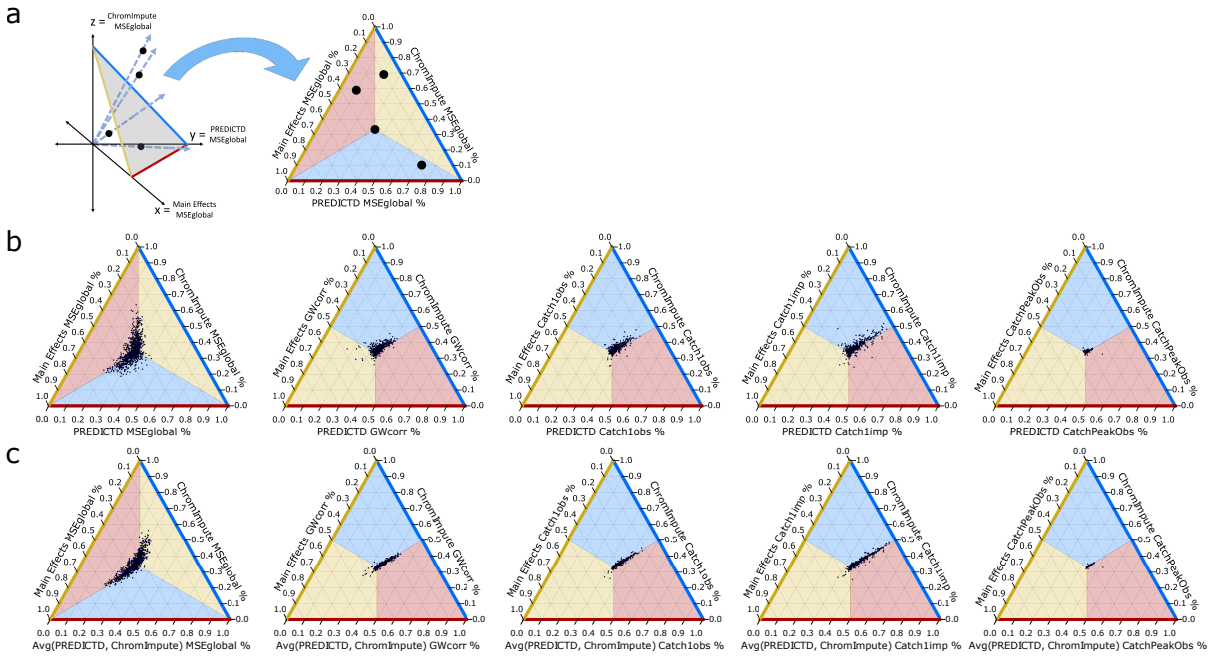

Figure 3.4: **PREDICTD performs comparably to ChromImpute, and combining the models improves the result. a.** Schematic describing how a ternary plot relates to Cartesian coordinates. Each experiment (represented by a black dot) is plotted in Cartesian space based on the values of a particular quality score for imputed data (in this example, MSEglobal) from PREDICTD, ChromImpute, and Main Effects. Each point in this space is then projected onto a plane by a vector drawn through the point and the origin. The resulting ternary plot summarizes the relative magnitude of the quality score for the three models. If all models achieve the same quality measure score for a particular experiment, then that point will be projected onto the center of the ternary plot. Deviation towards a point of the triangle indicates that one model has a higher value for that quality measure than the other two, and deviation from the center towards one of the edges of the triangle indicates that one model has a lower value. Color shading of the plot area marks the regions of the ternary plot that indicate superior performance of each model on a particular quality measure. The pattern of the colors changes based on whether it is better to have a low value on that quality measure (as with mean squared error) or a high values (for example, the genome wide correlation). **b.** Comparing PREDICTD, ChromImpute, and Main Effects models across five quality measures: the global mean squared error (MSEglobal), the genome-wide Pearson correlation (GWcorr), the percent of the top 1% of observed data windows by signal value found in the top 5% of imputed windows (Catch1obs), the percent of the top 1% of imputed windows by signal value found in the top 5% of observed windows (Catch1imp), and the area under the receiver operating characteristic curve for recovery of observed peak calls from all imputed windows ranked by signal value (CatchPeakObs). **c.** The same as in **b.**, except that the quality measures for the averaged results of ChromImpute and PREDICTD are plotted along the bottom (red) axis instead of the measures for PREDICTD alone.

dressed this question as a way to validate PREDICTD in a biological application and to extend the EnhancerFinder results by assessing cell type-specific enhancer activity in the ncHARs based on the Roadmap data set.

Briefly, we imputed data for three enhancer-associated assays (DNase, H3K27ac, and H3K4me1) in all cell types, and averaged the imputed signal over each ncHAR to produce a small tensor with axes corresponding to three assays, 2640 ncHARs, and 127 cell types. We flattened the assay dimension of this tensor by taking the first principal component, then used a biclustering algorithm to group the ncHARs and cell types (see Methods). The resulting cell type groups are consistent with tissue of origin (Fig. 3.5a, Supplementary Data 2), and the ncHARs cluster based on enhancer-associated signal in different cell type clusters as follows: No signal (77% of the ncHARs), Brain/ES (13%), Epithelial/Mesenchymal (7%), Non-immune (2%), and Immune (1%) (Fig. 3.5a, Supplementary Data 3). Using the same strategy to cluster the available observed data gives very similar results, as quantified by the adjusted Rand index (Fig. 3.5b), especially when compared to two background models: Shuffled, in which the ncHAR coordinates have been randomly shuffled along the genome; and Other, in which the enhancer-associated marks were exchanged for three non-enhancer-associated marks (H3K4me3, H3K27me3, H3K36me3). A heatmap showing the clustering of observed data is provided in Supplementary Fig. B.17.

These biclustering results also agree with and expand upon previously published tissue specificity predictions from EnhancerFinder.[90,98] The brain enhancer predictions from that study are visibly enriched in our Brain/ES cluster, and limb and heart predictions are enriched in our clusters showing activity in differentiated, epithelial, and mesenchymal cell types (Fig. 3.5a). If we treat the EnhancerFinder tissue assignments[90] as another clustering of the ncHARs, we find that they are more similar to our clustering (both for observed and imputed data) than to either background clustering (Fig. 3.5b). In addition, our results expand on EnhancerFinder by assigning to cell type-associated clusters 289 ncHARs (11% of ncHARs) characterized by EnhancerFinder as either having activity in "other" tissues (98 ncHARs) or no developmental enhancer activity ("N/A", 191 ncHARs). We also find that our clustering successfully predicts enhancer activity for many functionally validated ncHARs, and furthermore assigns most of them to the correct cell types

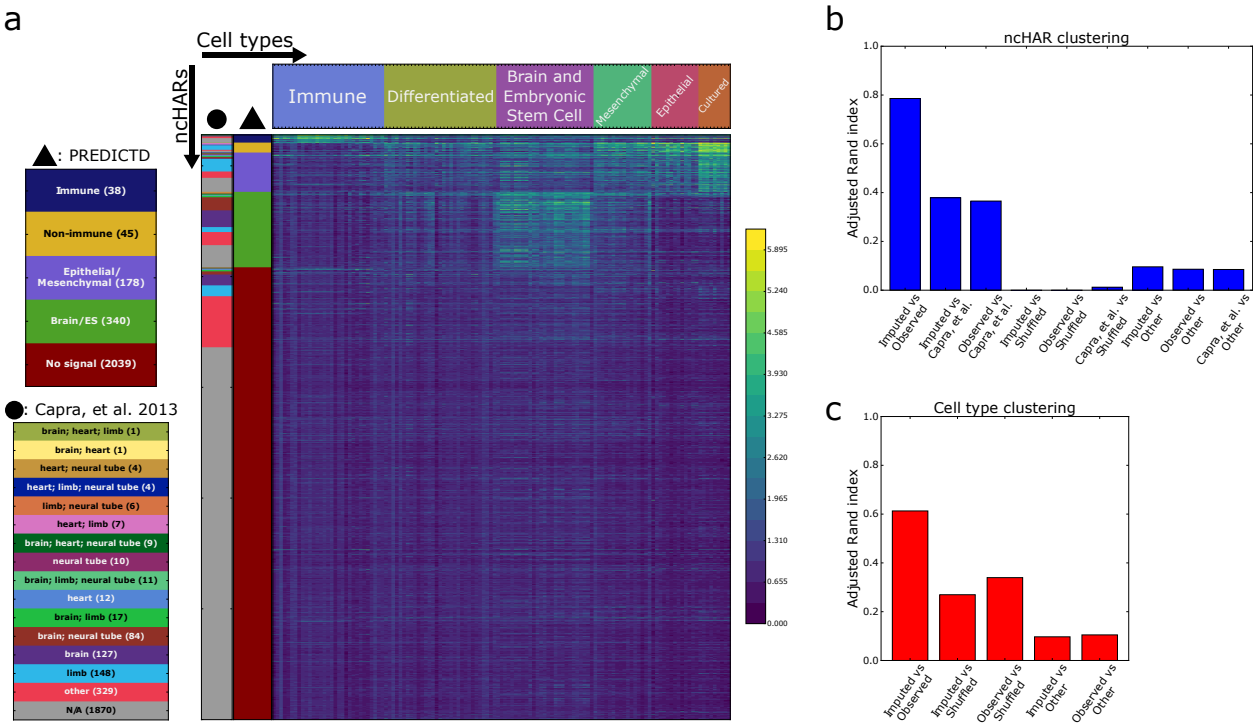

Figure 3.5: **Imputation of enhancer marks reveals tissue-specific patterns of enhancer-associated marks at non-coding human accelerated regions (ncHARs). a.** Average PREDICTD signal at each ncHAR was compiled for H3K4me1, H3K27ac, and DNase assays from all cell types. The first principal component with respect to the three assays was used in a biclustering to find 6 and 5 clusters along the cell type and ncHAR dimensions, respectively. The inverse hyperbolic sine-transformed signal from each of these assays was summed per cell type and ncHAR, and the resulting values were plotted as a heat map. The column marked with a black triangle at the top designates the color key for the ncHAR clusters. The leftmost column, designated with a black circle, identifies ncHARs with predicted tissue-specific developmental enhancer activity based on EnhancerFinder analysis from Capra et al, 2013. **b. and c.** Evaluation of the clustering results with the adjusted Rand index. The clustering results for observed data and PREDICTD for the ncHAR (**b.**) and cell type (**c.**) clusterings, and also those from Capra, et al. 2013 for the ncHARs, all show higher adjusted Rand index scores than the clustering results for observed data with shuffled ncHAR coordinates (Shuffled) or for observed data from non-enhancer-associated marks (Other).

(Supplementary Data 4). Briefly, we correctly identify enhancer activity in 10 of 23 ncHARs with evidence in the VISTA database,[90,100] and 6 of 7 ncHARs with validation results suggesting enhancer activity specific to the human allele and not the chimp allele;[90] we find evidence of enhancer identity for one of three ncHARs associated with *AUTS2*, a gene associated with autism spectrum disorder, and this enhancer was one of two from that study that showed transgenic enhancer activity;[97] *NPAS3* is a gene associated with schizophrenia that lies in a large cluster of 14 ncHARs, and we find enhancer signal for 7 of them, 6 of which have validated enhancer activity;[96] last, HAR2 is a ncHAR with validated human-specific limb enhancer activity that clusters with our Brain/ES category.[95] Thus, assessing potential enhancer activity based on the Roadmap Epigenomics data, which encompasses different cell types and developmental stages than ENCODE, agrees with previous results and expands on them to characterize more ncHARs as having potential tissue-specific enhancer activity.

Finally, we asked what types of biological processes these putative enhancers might regulate. We extracted the genomic coordinates of the ncHARs in each cluster and used the Genomic Regions Enrichment of Annotations Tool (GREAT)[101] to test for enriched ontology terms. Using the total list of ncHARs as the background, we found that the Brain/ES cluster of ncHARs is enriched for GO Biological Process terms associated with cell migration in different brain regions; the Epithelial/Mesenchymal cluster shows enrichment for terms associated with tissue development, particularly mesenchymal cell differentiation; and, although there are no significantly enriched GO Biological Process terms for the Non-immune cluster, there are enriched terms from a Mouse Phenotype ontology indicating these ncHARs could be associated with embryonic development and morphology (Table 3.2, Supplementary Data 5-12). We found no significantly enriched terms for the Immune cluster.

The question of whether ncHARs are active enhancers in modern humans or whether they are regions that formerly had enhancer activity that has been lost over the course of our evolution is a central question to the study of ncHAR biology. With this analysis we shed more light on which ncHARs have enhancer activity, and even provide some insight into the relevant developmental stage for such activity, as our cell types are derived from embryonic, fetal, and adult

Table 3.2: **Ontology search results are consistent with ncHAR cluster cell type identities.** We used GREAT to find enriched ontology terms associated with genes that are possibly regulated by ncHARs from each cluster. The list of all ncHARs was used as the background, and the terms are significant at FDR < 0.05 for the hypergeometric test and have at least a two-fold enrichment over expected.

ncHAR Cluster	Ontology	Enriched Term	FDR
Non-immune	Mouse Phenotype	abnormal craniofacial development	2.506e-03
		abnormal embryogenesis/ development	8.269e-03
		hemorrhage	2.022e-02
		abnormal embryonic tissue morphology	2.204e-02
		abnormal basioccipital bone morphology	2.907e-02
		partial neonatal lethality	2.944e-02
		abnormal skeleton development	3.439e-02
		abnormal placental labyrinth vasculature morphology	3.465e-02
		perinatal lethality	3.601e-02
		abnormal embryo size	3.605e-02
		abnormal craniofacial morphology	3.703e-02
		decreased embryo size	3.805e-02
		abnormal blood circulation	3.873e-02
		decreased skeletal muscle fiber number	3.931e-02
		abnormal embryonic growth/weight/body size	4.263e-02
		neonatal lethality	4.344e-02
Epithelial/Mesenchymal	GO Biological Process	embryonic organ development	2.487e-02
		embryo development ending in birth or egg hatching	2.502e-02
		somite development	2.514e-02
		tissue morphogenesis	2.746e-02
		mesenchyme development	2.948e-02
		stem cell differentiation	3.109e-02
		anterior/posterior pattern specification	3.237e-02
		mesenchymal cell development	3.310e-02
		chordate embryonic development	3.431e-02
		somitogenesis	3.569e-02
Brain/ES	GO Biological Process	telencephalon cell migration	2.150e-02
		cerebral cortex cell migration	2.897e-02
		forebrain cell migration	3.563e-02
		cerebral cortex radially oriented cell migration	4.523e-02

tissues. Taken together, these results show that PREDICTD imputed data can capture cell type-specific regulatory signals and that PREDICTD can be used as a tool to study the biology of new and under-characterized cell types in the future.

3.5 DISCUSSION

PREDICTD imputes thousands of epigenomics maps in parallel using a three-dimensional tensor factorization model. Our work makes several important contributions. First, the model leverages a machine learning method, tensor decomposition, that holds particular promise in genomics for analyzing increasingly high-dimensional data sets. Tensor factorization with the PARAFAC/CANDECOMP procedure was first proposed by two groups independently in 1970 in the context of analyzing psychometric electroencephalogram (EEG) data.[61,89] Tensor decomposition by this and related methods has since been applied in many other fields,[102,103] and increasingly in biomedical fields as well.[104–106] Tensor decomposition has advantages over two-dimensional methods because taking into account more than two dimensions reduces the rotational flexibility of the model and helps drive the factors to a solution that can explain patterns in all dimensions at once. Our particular application, completing a tensor with missing data, is an area of active research[107] and is analogous to methods for matrix factorization that have proven effective in other machine learning applications like recommender systems.[108] To our knowledge, PREDICTD is just the third application of the tensor decomposition approach to epigenomics data,[105,106] and the first to use a tensor completion approach to impute missing data in this setting. As such, our method demonstrates another way forward for integrating and jointly analyzing increasingly large and complex data sets in the field.

Second, PREDICTD provides some key advantages over the current state of the art for epigenomics data imputation. The best alternative method for predicting raw epigenomics signal is ChromImpute.[44] Our tensor factorization approach is simpler and arguably more elegant than ChromImpute because it naturally models the three key dimensions of the imputation problem while training on and imputing all data at once. In addition, PREDICTD is less computation-

ally intensive than ChromImpute and scales better to imputing large numbers of experiments (see Methods - Computing resource requirements). Furthermore, as a single model that describes all experiments, the parameters PREDICTD learns during training have the same semantics across different cell types, assays, and genomic positions. We show that these parameters contain information that can be used to distinguish different types of cells, assays, or genomic elements, and future work will investigate how the PREDICTD model itself might be used to gain biological insight. Last, we show that PREDICTD outperforms ChromImpute on the global mean squared error quality measure, despite generally slightly under-performing ChromImpute on other measures (Fig. 3.4, Supplementary Fig. B.14). There could be multiple reasons for this observation. First, as a tree-based model, ChromImpute can learn non-linear relationships in the data that PREDICTD cannot, and it is possible that this accounts for some of the difference in performance between the two approaches. Second, the mean squared error is central to the PREDICTD objective function, and so it is the quality measure on which the model should perform best; if another quality measure were used in the objective function, then PREDICTD might out-perform ChromImpute on that one instead. Nevertheless, the fact that averaging the PREDICTD and ChromImpute results outperforms both methods alone suggests that the two approaches are complementary, and we are interested in exploring additional methods, particularly non-linear models like deep neural networks, that might be able to combine the best of both approaches to further improve the imputed data quality.

Last, imputed data represents an important tool for guiding epigenomics studies. Such data are far cheaper to produce than observed data, closely match the data observed from experimental assays, and are useful in a number of contexts to generate hypotheses that can be explored in the wet lab. We showed that imputed data can provide insights into ncHARs; and Ernst and Kellis[44] previously showed that imputed data tend to have a higher signal-to-noise ratio than observed data, that imputed data can be used to generate high-quality automated genome annotations, and that regions of the genome with high imputed signal tend to be enriched in single nucleotide polymorphisms identified in genome-wide association studies (GWAS). In addition, raw imputed data includes information about signal amplitude and shape, which can provide insight into the types of

regulators and binding events that are producing that signal.[109–111] In contrast, other methods that use epigenomics data for various prediction tasks[45,112,113] all impute binarized epigenomics signal (i.e. peak calls) and do not preserve peak shape or amplitude. Raw imputed data sets, such as those produced by PREDICTD, make no assumptions about what research questions they will be used to address, and are widely applicable to any study that analyzes ChIP-seq or DNase-seq data. Thus, in conclusion, imputed data can provide insight into cell type-specific patterns of chromatin state and act as a powerful hypothesis generator. With just one or two epigenomics maps from a new cell type, PREDICTD can leverage the entire corpus of Roadmap Epigenomics data to generate high quality predictions of all assays.

3.6 METHODS

3.6.1 Data

We downloaded the consolidated genome-wide signal ($-\log_{10} p$) coverage tracks in bigWig format from the Roadmap Epigenomics data portal (http://egg2.wustl.edu/roadmap/web_portal/processed_data.html#ChipSeq_DNaseSeq).[7] These tracks are uniformly processed and currently represent the best-curated collection of epigenomic maps available. In addition, these are the same tracks that Ernst and Kellis[44] used to train ChromImpute, making it easier to compare our modeling approaches.

All observed signal tracks show a higher variance at regions of high signal than at regions of low signal. In order to stabilize this variance across the genome and to make the data more tractable for PREDICTD's Gaussian error model, we applied an inverse hyperbolic sine transform. This transformation, which has been used in previous studies of epigenomic maps,[23] is similar to a log transform but is defined for zero values.

After variance stabilization, we defined five training and test splits such that each observed experiment was in one test set. First, we removed any cell types or assays with fewer than five completed experiments to ensure that there would be enough support for training in each dimension in our model. This left 127 cell types and 24 assays, and a total of 1014 completed experiments

(66.6% missing). Next, we split these experiments into five test sets by randomly generating five disjoint subsets of experiments that each contained a stratified sample from across the available cell types and assays. Thus, in each split 20% of experiments comprise the test set and 80% the training set. In addition to the held out test set, PREDICTD requires a held out validation set to detect model convergence. To ensure that all data in the training data set contributed equally to the final imputation, the training data for each test set were further split into eight validation sets by cell type/assay pair so that for any pair of test and validation sets the data split is 20% test (203), 10% validation (100), and 70% training (711). The imputed values reported in this paper are the average test set predictions from eight models trained on the eight validation sets corresponding to that test set. 153 experiments from the first test set were held out of our model tuning procedure as a final test set to show that the model generalizes (Supplementary Fig. B.15).

Last, the data for each experiment was averaged into 25 bp bins across the genome using the `bedtools map` command,[85] and the bins overlapping the ENCODE Pilot Regions and 1kb windows centered at non-coding human accelerated regions were extracted for training the PREDICTD model. The resulting data set contains just over 1.3 million bins, or about 1% of the genome. All experiments reported here were conducted using models trained on this subset of the genome. We find that this is more than enough data to train the model, and imputing the entire genome is a relatively simple matter of applying the learned cell type and assay factors across all positions in the genome.

3.6.2 Model

In the following sections we present the PREDICTD model. As mentioned above, the data set can be represented as a 3D tensor with the axes being the cell types, the assays, and the locations across the genome. We refer to these axes as the cell type, assay, and genome dimensions, respectively. We use capital letters, J, K, and I to refer to the cardinality of each of these dimensions, and lowercase j, k, i, to refer to specific indices in each corresponding dimension. We use the same convention to refer to the number of latent factors in the model, L, and individual latent factor

indices, l. Each dimension has two learned data structures associated with it: a factor matrix, and a bias vector. We use bold capital letters to refer to the factor matrices, and bold lowercase letters to refer to the bias vectors. The cell type factor matrix and bias vector and their dimensions are $\mathbf{C}_{J\times L}$, and $\mathbf{c}_{J\times 1}$, respectively. Similarly, for the assay factor matrix and bias vector: $\mathbf{A}_{K\times L}$, and $\mathbf{a}_{K\times 1}$, and for the genome dimension: $\mathbf{G}_{I\times L}$, and $\mathbf{g}_{I\times 1}$.

Three main "axes" contribute to the observed biological signal in epigenomic maps: the cell type, the assay, and the genomic location that was measured. Having three axes on which to distribute the available data sets naturally lends the full data set the structure of the three dimensional tensor (i.e., a stack of two dimensional matrices) in which the size of one dimension corresponds to the number of cell types in the data set, another to the number of assays, and the third to the number of genomic locations. One might want to use other qualities or attributes to analyze the data (subject to one's research question and having enough training data), such as the lab that generated the data, the treatment that was applied, etc., but we are interested in parsing the data along the main cell type, assay, and genomic locus axes so that our model can most generally describe the biological phenomena in normal tissues.

Motivated by this 3D structure, we use tensor decomposition because this type of method factors the full data tensor into smaller components that summarize the contributions of each axis to the total data. The key to using tensor decomposition for imputing missing data is that the smaller components that are learned by the model do not have missing values by definition. This means that when we recombine the components to reconstruct the original tensor, the resulting reconstructed tensor will not only have values that approximate the existing data in the original tensor, but also predicted values for any missing entries in the original tensor. Our particular strategy, known in the literature as PARAFAC, finds a two-dimensional matrix (i.e. one "smaller component") for each dimension. All such matrices are of the same size along one dimension; this dimension is what we refer to as the number of "latent factors." The number of latent factors determines the complexity of the model, how well it can capture the information in each axis of the tensor, and thus how well the reconstructed tensor matches the original tensor. Each element of the reconstructed tensor is calculated by multiplying the three corresponding values (one from each matrix) for each latent

factor and then summing those products to arrive at a single number.

In order to perform imputation, we train the PREDICTD tensor decomposition model using the PARAFAC/CANDECOMP procedure,[61,89] which can very naturally model the three dimensional problem explained above. It also has several additional advantages: it is relatively simple to implement, it has the ability to scale to a large tensor size, and it holds the possibility of producing latent factors that can provide biological insight. Briefly, in this procedure the three-dimensional tensor is factored into three low-rank matrices, each with the same (user-specified) number of column vectors. These column vectors are called "latent factors," and the tensor is reconstructed by summing the outer products of the corresponding latent factor vector triplets. These factor matrices have no missing values, so when they are combined to reconstruct the original data tensor, the reconstructed tensor contains imputed data values that not only approximate the existing tensor data, but also fill in the missing values. More precisely, we start with a three-dimensional data tensor $\mathscr{D}$ with dimensions $J \times K \times I$, where $J = 127$ is the number of cell types, $K = 24$ is the number of assays, and $I = 1,309,125$ is the number of genomic locations (in our case the ENCODE Pilot Regions and 2640 ncHARs at 25 bp resolution), represented by the tensor. This tensor has missing data in fibers along the genome dimension, corresponding to experiments on cell type/assay pairs that have yet to be completed. The completed experiments, corresponding to tensor fibers that contain data, are split into training, validation, and test subsets, or $\mathbb{S}^{\text{train}}$, $\mathbb{S}^{\text{valid}}$, and $\mathbb{S}^{\text{test}}$, respectively.

We factor the tensor $\mathscr{D}$ into three factor matrices, and three bias vectors **a**, **c**, and **g**. These bias vectors are meant to capture global biases for each cell type, assay, or genomic location, respectively. Essentially, these terms subtract out the mean for each cell type, assay, and genomic location, which helps to mathematically center all of the data in the tensor around the same point so that the patterns that we want the model to learn are not obscured by trivial differences in scale along the axes. It is a common strategy for models like PARAFAC that perform best on data that is all on the same scale.

We train the model to find the values of these terms that minimize the following objective function:

$$\text{argmin}_{\mathbf{C},\mathbf{A},\mathbf{G},\mathbf{c},\mathbf{a},\mathbf{g}} \sum_{j,k,i \in \mathbb{S}^{\text{train}}} \left(\mathscr{D}^{\text{train}}_{j,k,i} - \left[\sum_{l=1}^{L} \mathbf{C}_{j,l} * \mathbf{A}_{k,l} * \mathbf{G}_{i,l} + \mathbf{c}_j + \mathbf{a}_k + \mathbf{g}_i \right] \right)^2 + \lambda_{\mathbf{C}} \|\mathbf{C}\|_2^2 + \lambda_{\mathbf{A}} \|\mathbf{A}\|_2^2 + \lambda_{\mathbf{G}} \|\mathbf{G}\|_2^2 \tag{3.1}$$

The objective function (Eq. (3.1)) has two main parts. The first part calculates the squared error between the training data, $\mathscr{D}^{\text{train}}_{j,k,i}$, and the model's prediction, $\sum_{l=1}^{L} \mathbf{C}_{j,l} * \mathbf{A}_{k,l} * \mathbf{G}_{i,l} + \mathbf{c}_j + \mathbf{a}_k + \mathbf{g}_i$. This term penalizes the distance between the imputed and observed data. The last three terms, $\lambda_{\mathbf{C}} \|\mathbf{C}\|_2^2 + \lambda_{\mathbf{A}} \|\mathbf{A}\|_2^2 + \lambda_{\mathbf{G}} \|\mathbf{G}\|_2^2$, implement L2 regularization on the factor matrices. This type of regularization penalizes large parameter values, and thus causes the model to strongly prefer a solution with small values on the parameters. Such regularization helps to reduce the flexibility of the model and helps to avoid overfitting the training data. Furthermore, we note that our choice of PARAFAC, which is a linear model with a limited number of latent dimensions, is itself a form of regularization in the sense that such a model is less flexible than more complex models like deep neural nets. PARAFAC is therefore inherently less prone to overfitting the training data compared to a non-linear model given the same model dimensionality.

Equation (3.1) cannot be solved analytically, so we solve it numerically using stochastic gradient descent (SGD). In SGD, we first initialize the three factor matrices with random values from a uniform distribution on the domain (-0.33 to 0.33) and the three bias vectors with the mean value from each corresponding plane in the tensor. Then we randomly iterate over the training set data points in the tensor, at each iteration calculating the gradient of the objective function (Eq. (3.1)) with respect to each factor matrix and bias vector, and then adding a fraction of this gradient to the corresponding parameter values. Over time, as more and more gradients are calculated and used to update the parameter values in the factor matrices and bias vectors, the model as a whole "moves" along the high-dimensional surface defined by the objective function and "down" toward a minimum that (ideally) represents a good solution. We track the model's progress toward this solution by periodically saving the value of the mean squared error on the held-out validation data points. Eventually, the validation mean squared error stops decreasing, which indicates that the model parameters have converged on a solution. Importantly, there is no guarantee that this solution is the best possible one, as in the case of PREDICTD (and PARAFAC more generally) the

objective function is not convex.

We should also note that PREDICTD incorporates several other modifications to this SGD procedure to improve the speed, reliability, and accuracy of training. First, in order to take full advantage of our compute cluster, we use parallel stochastic gradient descent,[91] which is discussed in detail in the Implementation section below. And second, to improve model convergence under SGD training, PREDICTD implements the Adam optimizer[114] with Nesterov Accelerated Gradient[115] (Fig. 3.1). Finally, we note that because there is no non-negativity constraint on the model training, a small fraction of imputed values are negative (Supplementary Fig. B.12). Negative values are invalid for $-log_{10}$ p-value tracks, so we set any such imputed values to zero in the final output.

There are many tensor decomposition methods (reviewed in[103]), however we chose the PARAFAC model because of its relative simplicity. It is not only straightforward to implement and parallelize, but it also requires fewer parameters than other tensor factorization methods.[102,103] Note that we implemented the model as described in the original publication,[61] and we included no additional constraints on the model during training except what was imposed by the L2 regularization terms in the objective and the constraints naturally imposed by using a relatively simple linear model on complex data with potentially non-linear underlying factors. The PARAFAC model also has the nice property that as long as mild conditions hold it will find a solution that is unique with respect to rotation transformations;[102,103] this is not a property of other tensor factorization approaches, including Tucker decomposition, which was used in.[105]

3.6.3 Implementation

PREDICTD is implemented in Python 2.7 and built using the Apache Spark 1.6.2 distributed computation framework (http://spark.apache.org). The code is open-source and available on BitBucket (https://bitbucket.org/noblelab/predictd), and the environment we used to train the model is available on Amazon Web Services as an Amazon Machine Image (see the BitBucket repository for info). Models were trained using Amazon Web Services (AWS) Elastic Compute Cloud (EC2)

(http://aws.amazon.com) and Microsoft Azure Spark on HDInsight (http://azure.microsoft.com). We bootstrapped an EC2 cluster running Apache Spark 1.6.2 by running the spark-ec2 script (https://github.com/amplab/spark-ec2) on a small EC2 instance (e.g. m3.medium) that we subsequently terminated after the cluster was up and running. Standard cluster configuration was a single m4.xlarge head node instance and one r3.8xlarge worker instance, giving a total cluster size of 2 nodes, 36 cores, and 260 GB of memory. Whenever possible, we used SPOT instances to make the computation more affordable. Microsoft Azure HDInsight clusters had similar resources. All data input to the model and all model output was written to cloud storage; either Simple Storage Service (S3) on AWS, or Blob Storage on Azure.

The data tensor is assembled into a Spark Resilient Distributed Data Structure (RDD) and partitioned among the cluster nodes such that each partition is stored on a single node and contains the data for 1000 genomic loci. This results in about 1300 partitions. The data in each of the 1000 elements in each partition is represented as a `scipy.sparse.csr_matrix`[116] object storing all observed data values for a particular genomic position. Each element of the data RDD also contains the corresponding entries from the **G** factor matrix, **g** bias vector, and data structures for the Adam optimizer[114] that are specific to each genomic locus (Fig. 3.1).

The first step of training selects a random 1% of available genomic positions ($\sim$13,000 positions, or $\sim$0.01% of all 25 bp bins in the genome) for training the cell type and assay parameters. Although this seems like a small sample of the genome, our results indicate that this is enough data to faithfully represent the distribution of signal across the tensor. We do see a slight improvement in performance if we include more of the genome in training, but at a cost of correspondingly increased memory usage and compute time. The main training phase then proceeds through a series of parallel stochastic gradient descent[91] iterations (Fig. 3.1c) on this subset of positions. Briefly, at the start of each parallel iteration, copies of the cell type and assay parameters, **C**, **c**, **A**, and **a**, are sent out to each partition. Each partition undergoes local stochastic gradient descent for 5000 iterations and applies the updates to the local copies of the assay and cell type parameters. The updated cell type and assay parameter values are then passed back to the master node where they are averaged element-wise with the results from all other partitions. The resulting averaged pa-

rameters are then copied and distributed to the partitions for the next round of parallel SGD. Note that over all rounds of SGD, we use a learning rate decay schedule of $\eta_t = \eta \times (\phi_\eta)^{t-1}$, where the learning rate decay parameter $\phi_\eta = 1 - 1e^{-6}$, and similarly for the Adam first moment parameter: $\beta 1_t = \beta 1 \times (\phi_{\beta_1})^{t-1}$, where $\phi_{\beta_1} = 1 - 1e^{-6}$.

Averaging the parameters after the parallel SGD updates allows the model to share information across the genome dimension; however, the averaging can initially make it harder for the model to converge. The **C** and **A** matrices are initialized randomly from a uniform distribution on the domain $(-0.33, 0.33)$, and thus during the first round of parallel SGD the independent nature of the local updates can lead to inconsistent updates to the latent factors in different partitions. When the results of these inconsistent updates are averaged, they produce poor parameter values, and it then takes many parallel iterations before the parameter values begin to converge. To combat this effect, the main training phase begins with a burn-in stage before attempting parallel SGD. In the burn-in stage, local SGD is performed for one epoch on 8000 genomic loci in a single partition, and after this, the updated **C**, **c**, **A**, and **a** parameters are used in a round of local SGD across the entire training subset to bring the genome dimension up to the same number of updates. This burn-in procedure allows the latent factors to have a consistent initial "identity" across the cluster when starting the parallel SGD updates.

Every three parallel SGD iterations, the mean squared error (MSE) is computed for each subset of data (training, validation, and test) and recorded. If the validation MSE is the lowest yet encountered by the model, the parameters from that iteration are copied and saved. Once a minimum number of parallel iterations have completed, the model tests for convergence by collecting the MSE on the validation set for iterations $t - 35$ to $t - 20$ (window 1), and $t - 15$ to t (window 2), and using a Wilcoxon rank sum test to determine if window $2 + 1e^{-5} >$ window 1, with one-tailed $p < 0.05$. If this convergence criterion is met, then one of two things happens. First, the model will check whether or not the user has requested a line search on the learning rate. If so, then it will reset the cell type, assay, and genome parameters to those found at the iteration with the minimum validation MSE and resume parallel SGD after halving the learning rate and reducing the Adam first moment weight $\beta 1_{\text{new}} = \beta 1_{\text{old}} - (1.0 - \beta 1_{\text{old}})$. When training the model, we used a

line search of length three, so the model was restarted from the current minimum and learning rate halved and $\beta 1$ adjusted three times. See Supplementary Fig. B.18 for an example of what the error curves from the parallel SGD look like after training a PREDICTD model. Once the line search is complete, or if no line search was requested, then the model stops parallel SGD, fixes the assay and cell type parameters, and finishes training on the genome parameters only.

Once the main phase of training is complete, the last phase of model training applies the cell type and assay parameters across all genomic positions. This is accomplished by fixing the cell type and assay parameters and calculating the second order solution on the genome parameters only. This requires just a single parameter update per genomic position, which is possible using least squares because fixing the cell type and assay parameters makes our objective function convex over the genome parameters. Once the final genome parameters are calculated, the assay, cell type, and genome parameters are saved to cloud storage, and the imputed tensor is computed and saved to the cloud for further analysis. On average, the entire training takes about 750 parallel iterations, and about 2 hours (wall clock time).

The above procedure is executed for every validation set associated with a given test set, and then the final imputed values for the held-out test data sets are calculated as a simple average of the corresponding imputed values from each validation set. Thus, for the results we report here, each imputed value represents the consensus of eight trained models.

3.6.4 Hyperparameter selection

One of the challenges of working with this type of model is that there are many hyperparameters to tune. These include the number of latent factors L, the learning rate η, the learning rate decay ϕ_η, the Adam first moment coefficient β_1 and its decay rate ϕ_{β_1}, a regularization coefficient for each latent parameter matrix ($\lambda_{\mathbf{A}}$, $\lambda_{\mathbf{C}}$, $\lambda_{\mathbf{G}}$), and one more regularization parameter for the second order genome updates ($\lambda_{\mathbf{G}_2}$).

Of these hyperparameters, perhaps the most important one for PREDICTD performance is the number of latent factors. This setting controls the dimensionality of the model, and thus the num-

ber of parameters that must be optimized during model training. Ideally, assuming a perfect match between the modeling approach and the data, the number of latent factors will equal the true underlying dimensionality of the data set. However, in practice this assumption does not really hold. First, real world data is often noisy enough that the "true" dimensionality of the input data is the full rank, and so instead we are forced to use fewer latent factors that approximate the dimensionality of theoretical, noiseless, data. Second, PREDICTD implements the original PARAFAC specification,[61] which relies on simple linear combinations of the corresponding latent factors in each dimension. However, in real data there could be factors that have nonlinear relationships, and there is evidence that PARAFAC in some cases will attempt to fit these relationships by adding additional factors to explicitly take them into account as if they are additional linear terms. This phenomenon was explored in an example from the original PARAFAC paper in which the best PARAFAC solution for a rank-2 synthetic data set with an interaction between the two dimensions used three latent factors: one for each dimension, plus another for the product of the two.[61] In the end, the best number of latent factors to use is simply the number that minimizes the error of the model while preserving its generalization performance, and this must be evaluated empirically.

Empirically searching for the best number of latent factors is non-trivial. The number of latent factors changes the dimensionality of the model, and thus the balance between bias and variance, which means that the regularization coefficients must be tuned in parallel with the latent factors. A simple strategy that has been shown to be surprisingly effective searching high dimensional hyperparameter space is simple random search, in which different random hyperparameter values are tested until a combination is found that provides good performance of the model.[117] Although the simplicity of the random choice strategy makes it very appealing, it can still require many iterations before one is confident that good hyperparameters have been found, which is a severe drawback when trying to optimize settings for a model like PREDICTD that takes multiple hours to train. Thus, hoping to find good hyperparameter settings in as few iterations as possible, we decided to use an auto-tuning software package called Spearmint.[118] Spearmint treats the PREDICTD model as a black box function and iteratively tries different hyperparameter settings; it uses Bayesian optimization to fit a Gaussian process that can predict the hyperparameter settings that will maximize

the improvement in model performance in the next iteration. There is still some debate in the field as to whether or not this kind of auto-tuning strategy reliably finds better hyperparameter values than simple random search;[119] however evidence shows that such Bayesian approaches tend to converge to a good selection of hyperparameters in fewer iterations than random search,[118] and thus minimize the time spent searching hyperparameters.

We ran Spearmint multiple times as we developed the PREDICTD model, each time holding out the first test set so that we would have some data to test the generalizability of PREDICTD. Early Spearmint runs and some manual grid search of the hyperparameters suggested that 100 latent factors was a good setting for the model dimensionality. Once we settled on 100 latent factors, we ran Spearmint again to fine tune the learning rate and regularization coefficients. We let it train 188 PREDICTD models with different hyperparameter settings and selected the settings from the model that gave the lowest observed validation MSE. During this process, we discovered that PREDICTD is relatively insensitive to the particular values of the three regularization coefficients $\lambda_{\mathbf{C}}$, $\lambda_{\mathbf{A}}$, and $\lambda_{\mathbf{G}}$, but that it seemed to prefer extremely low values (essentially, no regularization) on at least one of the matrix factors. In contrast, the hyperparameter search revealed that PREDICTD performance depends more heavily on particular values for the learning rate, η, and the second order genome update regularization, $\lambda_{\mathbf{G}_2}$. We also found that our imputation scheme of averaging eight models trained with different validation sets imposed extra regularization on the ultimate averaged solution, and that to achieve the best generalizability of our averaged solution we had to compensate for the regularization introduced by the averaging by choosing a lower $\lambda_{\mathbf{G}_2}$ than the one suggested by Spearmint as the best setting for a single model. After trying different $\lambda_{\mathbf{G}_2}$ values (Supplementary Fig. B.21), we decided to reduce $\lambda_{\mathbf{G}_2}$ by a factor of 10 since this showed that the validation MSE stayed roughly constant or a little bit lower than the minimum validation MSE from the parallel SGD iterations, and thus we were not lowering the regularization so much that the model overfit and increased the validation MSE. Our final chosen hyperparameter values are given in Table 3.3.

Table 3.3: **Hyperparameter values.** The third column indicates whether the hyperparameter value was selected using Spearmint, and an asterisk indicates the final value was tuned by hand after Spearmint optimization.

Hyperparameter	Value	Spearmint?
η	0.0045	Y
ϕ_η	$1-1e^{-6}$	N
$\beta 1$	0.9	N
ϕ_{β_1}	$1-1e^{-6}$	N
$\beta 2$	0.999	N
L	100	Y*
$\lambda_{\mathbf{C}}$	4.792	Y
$\lambda_{\mathbf{A}}$	$8.757e^{-27}$	Y
$\lambda_{\mathbf{G}}$	$8.757e^{-27}$	Y
$\lambda_{\mathbf{G}_2}$	0.4122	Y*

In addition to using Spearmint for model selection, we also used it to systematically explore the effects of changing the model dimensionality by changing the number of latent factors (Supplementary Figs. B.19, B.20). In this hyperparameter search, we fixed the dimensionality at one of 17 levels between 2 and 512 latent factors, and then used Spearmint to optimize the other hyperparameters (η, $\lambda_{\mathbf{C}}$, $\lambda_{\mathbf{A}}$, $\lambda_{\mathbf{G}}$, and $\lambda_{\mathbf{G}_2}$). We allowed the Spearmint runs with larger numbers of latent factors to train longer to give them more chances to explore the more complex solution space of these higher dimensionality models. We used a systematic stopping criterion as follows: each Spearmint search had to train for at least 50 iterations or 40% of the number of latent factors, whichever was more, and had to stop after it had trained at least 20 iterations or 15% of the number of latent factors, whichever was more, past its best result (Supplementary Fig. B.19 blue/red bars). After this search, we noticed that there was a plateau in the validation MSE from 16 latent factors to 64 latent factors, so to gain more resolution on this range of latent factors, we trained the 32 and 64 Spearmint searches out to 120 iterations. We found that the solutions for both models improved, and 64 latent factors improved more than 32 latent factors, but that neither model found a better solution than 100 latent factors (Supplementary Fig. B.19 brown/orange bars). In order to avoid biasing Spearmint's choice of hyperparameter settings for a particular validation set or subset of

genomic locations, we had allowed the validation set and the training subset of genomic windows to vary randomly over the course of the hyperparameter search. However, this meant that any given best Spearmint result could still be due to the model getting "lucky" and finding a validation set or set of genomic windows that was particularly favorable for training. To convince ourselves that the trend our Spearmint search revealed is real, we took the best hyperparameter settings for each latent factor level (for 32 and 64 latent factors these were the results of the expanded search) and trained ten models each with fixed validation sets and a fixed set of genomic windows, only varying the random initialization of the factor matrices from model to model (Supplementary Fig. B.20). The results show the same trend as a function of model dimensionality as in our original hyperparameter search (Supplementary Fig. B.19), and we also verified that the distribution of validation MSE for 64 latent factors is significantly different than that for 100 latent factors (Wilcoxon rank-sum test $p < 0.05$).

To save time on model training during the Spearmint iterations, we relaxed the convergence criteria to use a larger shift between the two samples in the Wilcoxon rank-sum test ($5e^{-05}$ instead of $1e^{-05}$) and we only did a single line search after the model first converged instead of three. It is important to note that, despite our efforts, there may be even better hyperparameter settings that our search did not encounter. As new discoveries concerning hyperparameter tuning unfold in the machine learning literature the settings for PREDICTD can be revisited to perhaps further increase its performance.

3.6.5 Imputing the whole genome

Although for the purpose of analyzing the PREDICTD model we only imputed about 1% of the genome, we generated whole genome imputed tracks in bigWig format for the UCSC Genome Browser. These tracks are available for download from the ENCODE project website (http://encodeproject.org).

3.6.6 Imputing data for a novel cell type

We provide a tutorial on the BitBucket site (https://bitbucket.org/noblelab/predictd/wiki/Home) that details how a user can train a PREDICTD model to generate imputed data for a new cell type. Briefly, a user can upload $-log_{10}$ p-value tracks in bigWig format to an Amazon S3 bucket, and then PREDICTD will add that data to the Roadmap Epigenomics tensor, train the model, and write imputed data for the new cell type back to S3 in bigWig format. The tutorial demonstrates how this is done with seven data sets from the Fetal Spinal Cord cell type that we downloaded from the ENCODE portal (http://www.encodeproject.org).

3.6.7 Computing resource requirements

The resource requirements of PREDICTD are not very great considering the size of the model. We find that training a single PREDICTD model on the tensor described in the paper (127 x 24 x 1.3e6) takes on average just under two hours on a two node cluster consisting of a head node with 4 cores (Intel Xeon E5-2676 v3 Haswell or Xeon E5-2686 v4 Broadwell processors) and 16 GB of memory (e.g. an m4.xlarge AWS EC2 instance) and a worker node with 32 cores (Intel Xeon E5-2670 v2 Ivy Bridge) and 244 GB of memory (e.g. an r3.8xlarge AWS EC2 instance). For this manuscript, each experiment was imputed as an average of eight models trained with random starts and different validation sets, so one could train these models to use for imputation in about 16 hours. After training the models, imputing values for the limited subset of genomic positions used for training is quite fast. However, if one needs to impute the whole genome it takes longer because the learned cell type and assay factors must be applied across all genomic locations. To do this without having to store the entire tensor in memory at once (all genomic positions and no missing values), we read in data for batches of genomic positions, train the corresponding genome parameters based on the existing cell type and assay parameters, and then write out the imputed values for each batch. For imputing whole genome data for one new cell type (that is, 24 whole genome imputed experiments) the cluster configuration described above requires an additional 24 hours, for a total of $\sim$40 hours for model training and whole genome imputation.

In this manuscript we present a more extreme case in which we impute all 3048 possible experiments in the Roadmap Epigenomics tensor at 25 base pair resolution, and to do this we used a larger worker node to increase throughput. If we use a x1.16xlarge instance as the worker node, which has 64 cores (Intel Xeon E7-8880 v3 Haswell) and 976 GB of memory, we can use the trained models to impute the whole genome for all 3048 experiments in approximately 88 hours. The resulting imputed tracks represent the consensus of eight models for each experiment, and these experiments were split into five test sets, giving a total of 40 models that took about 76.5 hours to train. Thus, training and imputation for the 3048 Roadmap Epigenomics tracks takes a total time of $\sim$164.5 hours.

To compare with ChromImpute's runtime, we can convert this wall-clock time to an approximate number of CPU hours required to run PREDICTD on the full tensor. Using the smaller cluster to train the 48 models, we calculate PREDICTD requires about 36 cores $\times$ 76.5 hours $=$ $2,754$ CPU hours. Switching to the larger cluster for imputation, we find that PREDICTD consumes about an additional 68 cores $\times$ 88 hours $= 5,984$ CPU hours. Thus, in total PREDICTD can train the models and impute 3048 experiments in $\sim 8,738$ CPU hours. This run time is more than an order of magnitude less than that quoted in the ChromImpute supplement,[44] which reports that ChromImpute requires a total run time of 103,560 CPU hours for model training and output generation. Even taking into account the fact that we imputed about 25% fewer experiments for this paper than were imputed in the ChromImpute manuscript, ChromImpute still requires on the order of ten times more CPU hours to train the models and impute the Roadmap Epigenomics tensor than PREDICTD does.

3.6.8 Advantages of the consumer cloud

Cloud computing is becoming a powerful tool for bioinformatics. Large consortia such as the Encyclopedia of DNA Elements[6] and The Cancer Genome Atlas (http://cancergenome.nih.gov) are making their data available on cloud platforms. As computational analyses grow more complex and require more computing resources to handle larger data sets, the cloud offers two distinct

advantages. First, cloud services provide a centralized way to host large data sets used by the community that makes data storage, versioning, and access more simple and efficient. Transferring gigabytes, or even terabytes, of data is slow and expensive in terms of network bandwidth, but moving code and computation to the data is fast and cheap. Second, in addition to hosting data sets, cloud services can host saved computing environments. Such virtual machine images can help with reproducibility of results for complex analyses because the code can be written in such a way that other users can not only use the same code and data as the original authors, but they can run the analysis in the same computing environment. One downside of cloud computing for labs that have access to a local cluster is that cloud resources are charged by usage; nevertheless, generating high quality imputed data using PREDICTD is extremely cost effective compared to collecting the observed data. Training the models and generating the final imputed data for this paper costs on the order of US $0.10 per data set, which is orders of magnitude lower than the cost of completing these experiments in the wet lab, and this cost can be expected to drop as computational resources become cheaper and more efficient optimization methods are devised.

3.6.9 Imputation quality measures

We generated tracks for the imputed data by extracting the data for each 25 bp bin from the imputed results, writing the results to file in bedGraph format, then converting to bigWig using the bedGraphToBigWig utility from UCSC. Imputed tracks were visually inspected alongside Roadmap Consolidated data tracks and peak calls in the UCSC Genome Browser. We did not reverse the variance stabilizing inverse hyperbolic sine transform when evaluating model performance. This is appropriate because it maintains the Gaussian error model that underlies the PREDICTD optimization.

We also implemented ten different quality assessment measures (listed below), the last seven of which were first reported for ChromImpute.[44] We report these measures for held out test set experiments and compute them over the ENCODE Pilot Regions (Supplementary Fig. B.14).

- **MSEglobal:** Mean squared error between the imputed and observed values at all available

genomic positions.

- **MSE1obs:** Mean squared error between the imputed and observed values in the top 1% of genomic positions ranked by the observed signal values.

- **MSE1imp:** Mean squared error between the imputed and observed values in the top 1% of genomic positions ranked by the imputed signal values.

  - **MSE1imppred:** Mean squared error between the imputed and observed values in the top 1% of genomic positions ranked by the signal values imputed by PREDICTD.
  - **MSE1impchrimp:** Mean squared error between the imputed and observed values in the top 1% of genomic positions ranked by the signal values imputed by ChromImpute.
  - **MSE1impme:** Mean squared error between the imputed and observed values in the top 1% of genomic positions ranked by the signal values imputed by Main Effects.

- **GWcorr:** Pearson correlation between imputed and observed values at all available genomic positions.

- **Match1:** Percentage of the top 1% of genomic positions ranked by observed signal that are also found in the top 1% of genomic positions ranked by imputed signal.

- **Catch1obs:** Percentage of the top 1% of genomic positions ranked by observed signal that are also found in the top 5% of genomic positions ranked by imputed signal.

- **Catch1imp:** Percentage of top 1% of genomic positions ranked by imputed signal that are also found in the top 5% of genomic positions ranked by observed signal.

- **AucObs1:** Recovery of the top 1% of genomic positions ranked by observed signal from all genomic positions ranked by imputed signal calculated as the area under the curve of the receiver operating characteristic.

- **AucImp1:** Recovery of the top 1% of genomic positions ranked by imputed signal from all genomic positions ranked by observed signal calculated as the area under the curve of the receiver operating characteristic.

- **CatchPeakObs:** Recovery of genomic positions at called peaks in observed signal from all genomic positions ranked by imputed signal calculated as the area under the curve of the receiver operating characteristic.

3.6.10 Analyzing model parameters

The parameter values corresponding to individual latent factors are not individually interpretable, but intuitively we can understand that each latent factor describes some pattern in the data that the model finds useful for imputation. For example, the first latent factor (i.e., column 0 in each of the three factor matrices) might contain values that capture a pattern of high signal in promoter marks, in blood cell types, at active genes. In such a case the value at this latent factor for a particular assay might suggest how often that mark is found at promoters; for a particular cell type its relatedness to blood; and for a genomic locus how many promoter-associated features occur there in blood cell types. If these three conditions hold, then the model is likely to have more extreme values for these parameters that end up imputing a high value for that cell type/assay pair at that genomic position.

3.6.11 Clustering cell types and assays

The rows of the cell type and assay factor matrices, with each row containing the model parameters for a particular cell type or assay, respectively, were clustered using hierarchical clustering. This analysis was implemented in Python 2.7 using `scipy.spatial.distance.pdist` with `metric='cosine'` to generate the distance matrix, and `scipy.cluster.hierarchy.linkage` with `method='average'` to generate clusters. The columns of each factor matrix (i.e. the latent factor vectors) were also clustered in the same way to help with visualizing the clusters. The parameter values were plotted as a heat map with rows and columns ordered according to the results of the hierarchical clustering.

3.6.12 Summarizing latent factor patterns at genomic elements

The genome factor matrix is too large to usefully visualize as a heatmap, so we sought to aggregate the parameter values across different types of genomic features. We mapped all annotated protein-coding genes from the GENCODE v19 human genome annotation[15] (https://www.gencodegenes.org/releases/19.html) with a designated primary transcript isoform (called by the APPRIS pipeline) to a canonical gene model consisting of nine components: promoter, 5' UTR, first exon, first intron, middle exon, middle intron, last exon, last intron, and 3' UTR. The promoter for each gene was defined as the 2 kb region flanking the 5' end of the gene annotation, while the other components were either taken directly from the GENCODE annotation (5' UTR, exons, 3' UTR) or were inferred (introns). For each gene, each component was split into ten evenly spaced bins and the values for each latent factor were averaged so that there was a single value for each latent factor for each bin. Coding regions for genes with a single exon or two exons were mapped only to first exon, or first exon and last exon components, respectively. Genes with only one or two introns were handled analogously. For genes with multiple middle exons and introns, each exon/intron was binned independently and the data for each middle exon/intron bin was averaged across all middle exons/introns. In order to plot the results, outlier values in the bins (defined as any values outside 1.5 * IQR) were removed and the remaining values averaged across corresponding bins for all binned gene models. This resulted in a matrix containing latent factors on the rows and gene model bins on the columns. The latent factors (rows) were clustered using hierarchical clustering, with `scipy.spatial.distance.pdist(metric='euclidean')` to generate the distance matrix and `scipy.cluster.hierarchy.linkage(method='ward')` to generate clusters, and this matrix was plotted as a heat map.

To compile a reference list of genome coordinates containing distal regulatory elements that is orthogonal to our imputed data, we downloaded P300 peak data from six ENCODE cell lines (A549, GM12878, H1, HeLa, HepG2, and K562), filtered for peaks with FDR < 0.01, merged the peak files with `bedtools merge` to create a single reference list, and averaged genome latent factor values as in the gene model explained above for ten 200 bp bins covering 2kb windows

centered on these peaks.

To validate that the patterns in the genome parameters were not due to chance, we generated the same heatmap, but before averaging the bins for each gene model and P300 site we randomly permuted the order of the genome latent factors (Supplementary Fig. B.11).

3.6.13 Comparing to ChromImpute

To compare the performance of PREDICTD with ChromImpute, we downloaded the ChromImpute results from the Roadmap Epigenomics website and put them through the same pipeline as for the observed data: Convert to bedgraph, use `bedtools map` to calculate the mean signal over 25 bp bins, extract the bins overlapping the ENCODE Pilot Regions, apply the inverse hyperbolic sine transform, and store the tracks in a Spark Resilient Distributed Dataset (RDD) containing a list of `scipy.sparse.csr_matrix` objects.

We calculated all of the quality measures on these ChromImpute data sets and plotted these results against those for PREDICTD for each experiment as a ternary scatter plot (Fig. 3.4b, Supplementary Fig. B.14). We also averaged each element of this ChromImpute RDD with its corresponding element in the PREDICTD results, calculated the quality measures, and compared them in the same way (Fig. 3.4c). In order to compare both ChromImpute and PREDICTD to the baseline Main Effects model, we used ternary plots[120] to project the three dimensional comparison of each experiment to two dimensions. Each point on these ternary plots represent the relative magnitude of each dimension for that point. So, each coordinate (x,y,z) in Cartesian space is projected to a point (x',y',z') such that $x' = \frac{x}{x+y+z}$, $y' = \frac{y}{x+y+z}$, and $z' = \frac{z}{x+y+z}$. Thus, for the case where $x = y = z$ the corresponding point $(x',y',z') = (0.33, 0.33, 0.33)$ and will fall at the center of the ternary plot, while points that lie along the Cartesian axes will fall at the extreme points of the ternary plot (e.g. $(x,y,z) = (1,0,0) = (x',y',z')$).

It is important to emphasize that the quality measure with the best PREDICTD performance, MSEglobal, is also explicitly optimized by the PREDICTD objective function during training. This shows that PREDICTD is doing well on its assigned learning task, and highlights the importance

of designing an objective function that reflects the task that the model will address. As such, it should be possible to tune the objective function to perform better on other quality measures if need be. For example, in an attempt to boost PREDICTD's performance on regions with higher signal we experimented with weighting genomic positions by ranking them by the sum of their signal level ranks in each training data set. This provided some improvement on the MSE at the top 1% of observed signal windows measure (MSE1obs), but we ultimately decided to pursue the simpler and more balanced objective function presented here.

3.6.14 Assessing cell type-specific enhancer signatures at ncHARs

We downloaded the non-coding human accelerated region (ncHAR) coordinates used in Capra, 2013,[90] removed any that overlapped a protein-coding exon according to the GENCODE v19 annotations,[15] and extracted all available observed and imputed data for the enhancer-associated assays H3K4me1, H3K27ac, and DNase at these regions. Some cell types were lacking observed data for H3K27ac (29) and/or DNase (74), but observed data for H3K4me1 was available in all cell types. We took the mean signal for observed experiment at each ncHAR coordinate and used that as input to the subsequent analysis.

First, we extracted the first principal component of the three assays for all ncHARs and cell types using `sklearn.decomposition.TruncatedSVD`[121] to reduce the assay dimension length from three to one and construct a matrix of ncHARs by cell types. This also had the effect of filling in missing values for the observed data. Next we wanted to cluster the ncHARs and cell types, and so we first used the matrix based on imputed data to assess how many clusters would be appropriate for the data. Briefly, for both the ncHAR and cell type dimensions, we conducted an elbow analysis by calculating the Bayesian information criterion (BIC) for k-means clustering results for all values $2 <= k <= 40$, as well as a silhouette analysis on the same range of values for k (Supplementary Fig. B.16). Based on the results, we decided that $k = 5$ for the ncHARs and $k = 6$ for cell types would give us a good balance of distance between clusters and number of clusters.

Next, we clustered the imputed and observed matrices with the scikit-learn

`sklearn.cluster.bicluster.SpectralBiclustering` class[121] to generate a biclustering using six column clusters and five row clusters. And finally, we plotted the clustering results for the imputed data as a heatmap in which each cell is the inverse hyperbolic sine-transformed sum of the mean H3K4me1, H3K27ac, and DNase signals at a particular ncHAR in a particular cell type. We also plotted the tissue assignments for ncHARs with predicted developmental enhancer activity based on EnhancerFinder[98] calls in the Capra, 2013[90] paper alongside our ncHAR clusters (Fig. 3.5a). The same plot for the observed data is shown in Supplementary Fig. B.17.

In order to gain further insight into the genes associated with our ncHAR clusters, we extracted the genomic coordinates of the ncHARS in each cluster and input these regions to the Genomic Regions Enrichment of Annotations Tool (GREAT)[101] to find enriched ontology terms associated with nearby genes. We used GREAT version 3.0.0 on the human hg19 assembly with the default association rule parameters (Basal+extension: 5000 bp upstream, 1000 bp downstream, 1000000 bp max extension, curated regulatory domains included). We first analyzed each cluster for term enrichment against a whole genome background (Supplementary Data 5,6,8,10,12), and then ran the test with the same parameters against the list of all ncHARs as the background (3.2, Supplementary Data 7,9,11). No terms were significantly enriched for cluster 0 (No Signal) or cluster 4 (Immune) when using the all ncHAR background, and so we omit these results from the supplement. When reporting the results in the main text we used the default GREAT filters for significant terms: FDR < 0.05 for the hypergeometric test with at least a two-fold enrichment over expected.

Last, in order to compare the clustering results on the imputed data to the observed data, we used the adjusted Rand index, which assesses how often pairs of data points are put in the same or different clusters, on the ncHAR and cell type clusters independently. As negative controls, we also conducted the same clustering analysis on the observed data after shuffling the ncHARs to other non-coding coordinates (Shuffled), and after switching out the enhancer-associated marks for H3K4me3, H3K36me3, and H3K27me3, which are not associated with enhancers (Other). We compared the resulting clusters with the enhancer-associated imputed data and observed data clusters, again using the adjusted Rand index. Last, we used the adjusted Rand index once more

to assess the similarity of our biclustering results to the grouping of ncHARs based on predicted tissue-specific developmental enhancer activity from Capra, 2013[90] (Fig. 3.5b).

3.6.15 Code availability

The PREDICTD code base is open source and made available through the MIT License. All code and documentation required to run PREDICTD, including tutorials and command line usage, are available through the PREDICTD repository hosted on BitBucket: https://bitbucket.org/noblelab/predictd.

3.6.16 Data availability

- The Roadmap Epigenomics Consolidated Data are available through the project data portal, http://egg2.wustl.edu/roadmap/web˙portal/processed˙data.html#ChipSeq˙DNaseSeq.

- ChromImpute data sets are also available through the Roadmap Epigenomics project data portal, http://egg2.wustl.edu/roadmap/web˙portal/imputed.html#imp˙sig.

- All imputed data generated for this paper are available through the ENCODE project portal, https://www.encodeproject.org/, and the list of accession IDs is provided in the SupplementaryData17.xlsx file associated with this manuscript.

- The Amazon Machine Image for running the PREDICTD software, along with the associated reference data files, are hosted on Amazon Web Services. The download locations are provided in the documentation with the PREDICTD code (see the code availability statement above).

- Data for the quality measures reported for PREDICTD in Figs. 3.2b, 3.4, and Supplementary Figs. B.2, B.14, and B.15 are provided in Supplementary Data, as are the results from the GREAT analysis of ncHAR clusters (see Supplementary Information).

3.7 ACKNOWLEDGEMENTS

We gratefully acknowledge support from the Amazon Web Services Cloud Credits for Research program and Microsoft Azure for Research program for providing computing cycles to help with the development of PREDICTD. We would also like to thank Dr. Rob Fatland and the UW High Performance Computing Club for assistance with the cloud computing aspects of our project and for granting us additional Amazon Web Services credits. This work was funded by National Institutes of Health awards R01 ES024917 and U41 HG007000.

3.8 STATEMENT OF CONFLICT OF INTEREST

The authors declare no conflict of interest.

Chapter 4: FINAL THOUGHTS

In this dissertation I reported on two projects that move the field closer to a comprehensive characterization of chromatin state in *C. elegans* and human cell types. I discussed a range of implications of the results and various future directions in the discussion sections of the individual chapters; here I will focus on how my two projects advance our understanding of gene regulatory networks and where the field is heading.

In Chapter 2, I showed that sciATAC-seq[39,40] is effective in *C. elegans* cells, providing the first cell type-resolved maps of chromatin accessibility in worm. By assigning accessible sites to their nearest downstream gene and cross-referencing with sciRNA-seq data,[30] I detected every major tissue type in worm, despite extreme sparsity in the data. The candidate regulatory sites have extensive overlap with existing bulk ChIP-seq[19] and ATAC-seq[65] data sets, but also contain a substantial number of new candidate regulatory sites. Most sites show some level of tissue specificity based on latent Dirichlet allocation topic modeling, highlighting the importance of mapping chromatin state in specific cell types. A working protocol for single-cell ATAC-seq in worms adds to this model organism's considerable advantages for studying development and genomics, including a limited number of cells and cell types that develop through an invariant cell lineage,[58,59] a mature and well-annotated genome assembly,[5,60] the ability to reproduce asexually, a short generation time, and powerful genetic tools. *C. elegans* holds promise to be the first metazoan to have gene expression measurements and chromatin accessibility maps for every cell throughout development. We already have single-cell RNA-seq data for nearly every cell from early- to late-embryonic development,[33] and a near term future direction is to collect scATAC-seq data to match. With continued optimization of sciATAC-seq in worms to yield more complex libraries and thus higher cell type resolution, the goal of a comprehensive map of regulatory sites and gene expression in *C. elegans* should be achievable.

Although *C. elegans* is among the most tractable systems for comprehensively measuring chro-

matin state, other efforts are underway to do the same for more complex organisms like human[6,7] and mouse.[18,31,41,43] Characterizing chromatin state in humans and mice poses tremendous practical challenges compared to doing so in the worm. The developmental time for humans and mice is much greater than *C. elegans*, so there are more time points to sample and those samples take longer to collect; the body plans are much more complex and cell types more diverse, so more samples are required for sufficient coverage of all cell types; and despite the requirement for greater numbers of samples, it is also more challenging and expensive to collect those samples because of the slow generation time, the requirement of dissection in order to obtain primary samples, and the critical ethical considerations when working in mouse and human. One way to address these challenges is through the continued development and application of single-cell technologies that can detect different cell types with more sensitivity and lower amounts of input tissue. Another way is to apply computational techniques that can help to prioritize experiments or even remove the requirement for performing some of them.

In Chapter 3 I presented one such machine learning method, called PaRallel Epigenomics Data Imputation with Cloud-based Tensor Decomposition (PREDICTD), to learn from available ChIP-seq and chromatin accessibility data and predict the results of experiments that have not yet been performed. PREDICTD models an epigenomics data compendium as a three-dimensional tensor with one dimension corresponding to cell types, another to epigenomics assays such as those for chromatin accessibility and protein binding sites, and the third to windows across the genome. For a given number of cell types C, assays A, and genomic loci G, any entry $d_{c,a,g}$ in the tensor corresponds to the measurement from assay a in cell type c at genomic position g. The model uses tensor decomposition to learn three low-rank matrix factors that share a common dimension K; each vector $k \in (1,2,...,K)$, or latent factor, captures an underlying informative pattern in the data. I showed that these latent factors summarize information that can distinguish among different cell types, assays, and types of genomic loci, that the latent factors can be used to fill in the missing entries in the tensor to predict the results of missing experiments, and that the quality of the imputed data is competitive with that of another state-of-the-art method, ChromImpute.[44]

Imputation methods like PREDICTD and ChromImpute struggle to predict the signals that are

only observed in one or a few cell types, and as such, imputed data will never be able to completely replace real measurements. However, the imputed data are useful for prioritizing experiments and proposing candidate regulatory elements that can be validated with further experimental work. In addition, models like PREDICTD that can integrate different types of data and summarize them in a shared representation (e.g. the genomic latent factors) can distill the information contained in large numbers of experiments into a more manageable compressed representation for aiding the characterization of regulatory elements,[122] even in cases where there are no missing data. Last, as generating various kinds of single-cell data become more common and we gain the ability to analyze increasingly specific and numerous cell types, imputation methods will be a key part of the computational toolbox for summarizing and learning from ever larger data corpora.

As we get closer to achieving the goal of comprehensively mapping cell type-specific gene expression patterns and chromatin states, we will increasingly need to look ahead to the next steps in deciphering the gene regulatory networks that control cellular states. Recall that describing a gene regulatory network requires five key pieces of information: 1) the genes involved, 2) the regulatory sites involved, 3) the effect of perturbing a transcription factor gene on target gene expression, 4) the effect of perturbing nearby regulatory sites on expression of the target gene, and 5) the effect of perturbing nearby regulatory sites on the binding of the transcription factor.[10,11] This dissertation has been concerned with collecting the first two pieces of information to identify the nodes of the regulatory network, and although there is still work to be done on this front, another grand challenge is to connect the nodes and fill in the edges of the network.

The best way to establish a regulatory connection between two genes is to do a knockout or perturbation experiment in which the activity of the regulatory gene is disrupted, and the candidate target gene is monitored for corresponding changes in expression level. Historically, for example in mapping the purple sea urchin developmental regulatory network,[10–12,14] this has been done by targeting specific mRNAs for disruption or degradation with morpholinos or RNAi, and monitoring the effects in bulk cell preparations with assays like qPCR, macroarrays, and microarrays. In the bulk sample setting, differentiating between direct and indirect effects of the perturbation is rather complicated. One first has to filter the affected genes for those that are expressed at the same

time and place as the gene targeted by the perturbation, for example by cross-referencing with whole mount *in situ* hybridization data. Then, further filtering and validation of the candidate interactions is required to filter out intracellular indirect effects. In some cases intracellular indirect regulation can be inferred heuristically based on the observation that direct targets of the perturbed gene tend to have the greatest changes in gene expression, but many of these connections require further investigation, for example by perturbation of nearby cis-regulatory sites, to confidently identify direct regulation.[10] Another way to confirm indirect connections is to attempt to rescue the expression of a candidate indirect target by introducing the mRNA of a likely intermediate regulator; if the regulation is indirect via that intermediate then the expression of the target gene will return to normal despite the perturbation of the indirect regulator.[12] Carefully testing for regulatory interactions in this way, one regulator at a time, requires thousands of experiments and years of work to solve a relatively small part of development. In order to achieve a comprehensive understanding of regulatory networks in simple organisms, and to begin to make a dent in more complex organisms, we need more powerful, high throughput techniques.

One way forward is to apply a new class of technologies that combine the high throughput and statistical power of single-cell RNA-seq with the modularity of CRISPR for making targeted genomic perturbations in high throughput.[123–126] Technologies such as Perturb-seq,[123,124] and others like it,[125,126] work by introducing a library of thousands of different CRISPR guide RNAs (gRNA) to cells that stably express either the Cas9 nuclease[123,125] or a nuclease-dead Cas9 that is fused to a KRAB repressor domain.[124,126] The gRNAs target Cas9 nuclease or dCas9-KRAB to thousands of sites in the genome in a single experiment where the Cas9 or dCas9-KRAB enzymes introduce mutations or repressive chromatin, respectively. The number of guides per cell, and thus the number of perturbations, can be tuned to measure the effect of single perturbations or many at once, and the genome-wide effects of knocking out or repressing the target genes are read out by single-cell RNA-seq. The plasmids containing the gRNA sequence are cleverly designed so that each gRNA is identifiable in each cell's scRNA-seq read out, tying the gene expression measurements to the specific perturbation(s) in that cell. This gRNA:scRNA-seq link can be established by either coexpressing a unique barcode sequence from the same plasmid as the gRNA,[123,124] or by

using the gRNA itself as the barcode.[125,126] After collecting the scRNA-seq data, the effects of the different perturbations can be assessed using the statistical power of having hundreds or thousands of single-cell observations for each perturbation.

The studies accompanying the publication of these methods for combining CRISPR-targeted perturbations with scRNA-seq show that this approach is a powerful method for high throughput interrogation of regulatory interactions; however, one limitation is that they were all either performed in cell lines or in cultured primary cells. For investigating developmental gene regulatory networks that function in the context of a whole developing organism, it will be necessary to adapt Perturb-seq to work in a free-living organism. This requirement is particularly important for *C. elegans*, which lacks a reliable system for long-term cell culture. The key steps to solve are to 1) get Cas9 and one or more gRNAs into cells, 2) express them at some developmental time point of interest, 3) allow some time for the effects of the knock down to take hold, and then 4) perform scRNA-seq. CRISPR works rather well in *C. elegans*, and although most of the work on CRISPR in the worm concerns integration of constructs into the genome to generate new worm strains,[127] other studies have detailed a heat shock-inducible CRISPR system that works in a single generation.[128,129]

Here is how the inducible CRISPR system might be used in *C. elegans* to perform high throughput perturbations. First, the plasmids containing the Cas9 and gRNAs are delivered by micro-injection directly into the *C. elegans* gonad. In the distal part of the gonad, meiotic nuclei exist in a syncytium before they are packaged into oocytes. Injection of the CRISPR-Cas9 plasmids allows them to be packaged into the newly formed oocytes, and during subsequent cell divisions the plasmids will be randomly partitioned among the cells of the developing embryo in a mosaic fashion. Similar to the transfection of cultured cells with a complex library of different gRNAs, the micro-injection mix can contain a variety of guides targeting various loci. Each injected worm can produce up to ~ 300 progeny, and a skilled technician can inject 100 or more worms in a session, giving a reasonably large sample size depending on the complexity of the gRNA pool. Once the constructs are delivered, there must be a way to activate the CRISPR system components to make the perturbations. Constitutively active expression would be sufficient for assessing the

earliest stages of development, but in order to perturb later developmental stages we need to ensure that the earlier stages can progress as normal. Inducible CRISPR systems work by expressing the CRISPR components off of a chemical-[130] or heat shock-inducible promoter,[128,129] and with such a system the embryos containing the injected plasmids could develop until the desired developmental stage, and then the perturbations could be induced. After allowing enough time for the perturbations to have an effect on gene regulation, the worms can be dissociated and assayed with scRNA-seq[30,33] to read out the effects. A potential limitation is that the CRISPR plasmids will be diluted out as cells divide and development proceeds, leading to some unperturbed cells at later developmental stages. In order to enrich the scRNA-seq input for cells that have a perturbation, it may be desirable to include a marker, such as GFP, that would allow selection for cells with CRISPR plasmids by FACS before doing the single-cell assay. There are surely additional technical challenges that would have to be overcome, but this type of experiment would be a way to scale up our ability to test regulatory networks in high throughput in developing multicellular organisms.

Further validation of the perturbation results will come from additional assays and computational analyses. In particular, although we can identify candidate regulatory sites in high throughput with sciATAC-seq, the data do not tell us which proteins are bound to those sites. Assays like single-cell ChIP-seq,[131] which relies on single-cell isolation and barcoding of fragments in droplets before breaking the emulsion and doing immunoprecipitation, and CUT-AND-RUN,[132,133] a low-input assay that uses a proteinA-antibody complexes to direct micrococcal nuclease to fragment the DNA near a protein of interest and release short fragments for sequencing, are two examples of technology that could one day allow us to assay the binding of specific proteins along the genome at single-cell resolution. Another approach to identifying the regulators binding to accessible regions identified by ATAC-seq is motif scanning. Tools like the MEME suite[134] can identify short DNA sequences that are enriched in accessible sites compared to control sites, and these enriched sequences can be compared to known transcription factor binding preferences[135] to infer which regulatory protein(s) are binding at those sites. This can work well in some cases, but the motif scanning approach has some important limitations. First, motifs are known or inferred for only a

fraction of proteins with DNA binding domains, so there are many transcription factors for which binding sites cannot be inferred. Second, motifs can be nearly identical for proteins from the same family, so even when a motif match is found there may be many transcription factors associated with that motif. And third, many proteins that contribute to gene regulation (e.g. chromatin regulators) have no sequence specificity of their own, but instead complex with various binding partners that end up obscuring the identity of the regulator. Making things even more complicated, there is evidence that tissue-specific regulation of target genes in some cases is dependent on "sequence suboptimization".[136] Suboptimal regulatory sequences achieve proper spatial and temporal expression of a target gene when the regulatory site incorporates motifs that differ from the optimal sequence preference of the regulating transcription factor. If the regulatory site is engineered to have the optimal motif instead, the target gene shows ectopic expression in tissues that do not normally express it. Furthermore, related evidence suggests that the ordering and spacing of suboptimal binding sites also play roles in the proper regulation of the target gene.[137] It is possible that to some extent the patterns of suboptimal sites can be learned by new computational models that incorporate DNA sequence. Such models may help us to better understand and predict transcription factor binding,[46] but in the end it will still be important to have a single-cell or low-input assay that can validate the inferred binding events from scATAC-seq.

As the field continues to advance toward a more comprehensive understanding of the gene regulatory networks underpinning development, differentiation, and cell state, new approaches will need to incorporate additional data types that capture still more aspects of gene regulation. For example, the high throughput procedure for perturbing and assaying regulatory networks *in vivo* that I sketched above focuses on the intracellular effects of perturbations in single cells, but we know that this is not the whole story. Throughout development, cells are nudged down the path to differentiation and induced to activate certain regulatory programs in response to signals from their neighboring cells and from other environmental inputs. Most single-cell technologies treat every cell as an independent sample and lose positional information that might tell us about which cells are signaling to each other, or which cells need to respond to an environmental cue. To some extent, because the developmental lineage is known in *C. elegans*, cell position can be inferred

based on the expression of marker genes,[33] but similar inferences may be impossible or misleading in the context of a perturbation experiment that changes the expression of the marker genes. New approaches and experiments will have to be designed going forward to add spatial information to this analysis, perhaps by integration of microscopy data[66] or an as-yet uninvented high throughput sequencing assay.

Lack of spatial information is not the only limitation imposed by current single-cell technologies. An important limitation for scRNA-seq in particular is that so far all scRNA-seq technologies rely on priming reverse transcription from the polyA tail of mature mRNA transcripts. Measuring only transcripts with polyA tails selects against many non-coding RNAs that lack a polyA tail, but may nevertheless contribute to developmental gene regulation. In addition, data from the current scRNA-seq methods are heavily biased for recovery of the 3' end of transcripts, and the lack of coverage for the middle and the 5' end of transcripts precludes learning about differential isoform usage during development and across cell types. Different isoforms can vary widely in length and can incorporate different sets of functional domains that might be expected to dramatically affect the function of the resulting protein. Last, a truly systems-level understanding of gene regulation will also have to incorporate information at the level of proteins. Proteins are the effectors of the connections in the gene regulatory networks we are trying to learn about – they implement the edges of the network. We also know that transcript abundance is frequently a poor predictor of protein abundance,[138] and furthermore, that protein activity is not only a function of abundance, but also of cell state and whether the protein itself is activated. ChIP-seq is a step in the right direction, but further characterization of regulatory proteins will be required for a complete understanding of how regulatory networks are implemented. For example, what are the protein complexes involved in gene regulation? What post-translational modifications are necessary to activate the functions of the proteins involved, and what signaling pathways provide input to modules in the gene regulatory network? A full understanding of the role that different transcription factors and chromatin regulators play in a given regulatory context, and why they play that role, will ultimately require measuring properties of proteins over and above whether or not they bind to DNA.

Twenty one years after *C. elegans* became the first multi-cellular organism with a sequenced

genome, we are still trying to understand how the genome encodes the information necessary to build an organism. The field will continue to build a mechanistic understanding of how and why development and differentiation proceed as they do. We will progress toward a better understanding of how the structure and function of the genome are related, which in turn will provide testable hypotheses and predictive models that will provide deep insights into some of the other big questions in biology, including how disease states arise and how to treat them, and how the evolution of the genome proceeds.

BIBLIOGRAPHY

[1] Avery, O. T., MacLeod, C. M. & McCarty, M. Studies on the Chemical Nature of the Substance Inducing Transformation of Pneumococcal Types. *The Journal of Experimental Medicine* **79**, 137–158 (1944). URL https://www.ncbi.nlm.nih.gov/pmc/articles/PMC2135445/.

[2] Lander, E. S. *et al.* Initial sequencing and analysis of the human genome. *Nature* **409**, 860–921 (2001). URL http://www.nature.com/nature/journal/v409/n6822/full/409860a0.html.

[3] Finishing the euchromatic sequence of the human genome. *Nature* **431**, 931 (2004). URL https://www.nature.com/articles/nature03001.

[4] Consortium*, T. C. e. S. Genome Sequence of the Nematode C. elegans: A Platform for Investigating Biology. *Science* **282**, 2012–2018 (1998). URL https://science-sciencemag-org.offcampus.lib.washington.edu/content/282/5396/2012.

[5] Hillier, L. W. *et al.* Genomics in C. elegans: So many genes, such a little worm. *Genome Research* **15**, 1651–1660 (2005). URL http://genome.cshlp.org/content/15/12/1651.

[6] An Integrated Encyclopedia of DNA Elements in the Human Genome. *Nature* **489**, 57–74 (2012). URL http://www.ncbi.nlm.nih.gov/pmc/articles/PMC3439153/.

[7] Roadmap Epigenomics Consortium *et al.* Integrative analysis of 111 reference human epigenomes. *Nature* **518**, 317–330 (2015). URL http://www.nature.com/nature/journal/v518/n7539/full/nature14248.html.

[8] Pliner, H. A. *et al.* Cicero Predicts cis-Regulatory DNA Interactions from Single-Cell Chromatin Accessibility Data. *Molecular Cell* **71**, 858–871.e8 (2018). URL http://www.sciencedirect.com/science/article/pii/S1097276518305471.

[9] Jacob, F. & Monod, J. Genetic regulatory mechanisms in the synthesis of proteins. *Journal of Molecular Biology* **3**, 318–356 (1961). URL http://www.sciencedirect.com/science/article/pii/S0022283661800727.

[10] Li, E. & Davidson, E. H. Building Developmental Gene Regulatory Networks. *Birth defects research. Part C, Embryo today : reviews* **87**, 123–130 (2009). URL http://www.ncbi.nlm.nih.gov/pmc/articles/PMC2747644/.

[11] Peter, I. S. & Davidson, E. H. Modularity and design principles in the sea urchin embryo gene regulatory network. *FEBS Letters* **583**, 3948–3958 (2009). URL http://www.sciencedirect.com/science/article/pii/S0014579309009739.

[12] Davidson, E. H. A Genomic Regulatory Network for Development. *Science* **295**, 1669–1678 (2002). URL http://www.sciencemag.org/cgi/doi/10.1126/science.1069883.

[13] Davidson, E. H. Emerging properties of animal gene regulatory networks. *Nature* **468**, 911–920 (2010). URL http://www.ncbi.nlm.nih.gov/pmc/articles/PMC3967874/.

[14] Davidson, E. H. *et al.* A Provisional Regulatory Gene Network for Specification of Endomesoderm in the Sea Urchin Embryo. *Developmental Biology* **246**, 162–190 (2002). URL http://www.sciencedirect.com/science/article/pii/S0012160602906354.

[15] Harrow, J. *et al.* GENCODE: The reference human genome annotation for The ENCODE Project. *Genome Research* **22**, 1760–1774 (2012). URL http://genome.cshlp.org/content/22/9/1760.

[16] Gerstein, M. B. *et al.* Integrative Analysis of the Caenorhabditis elegans Genome by the modENCODE Project. *Science* **330**, 1775–1787 (2010). URL http://www.sciencemag.org/content/330/6012/1775.

[17] Gerstein, M. B. *et al.* Comparative analysis of the transcriptome across distant species. *Nature* **512**, 445–448 (2014). URL http://www.nature.com/nature/journal/v512/n7515/full/nature13424.html.

[18] Yue, F. *et al.* A comparative encyclopedia of DNA elements in the mouse genome. *Nature* **515**, 355–364 (2014). URL http://www.nature.com/nature/journal/v515/n7527/full/nature13992.html.

[19] Kudron, M. M. *et al.* The ModERN Resource: Genome-Wide Binding Profiles for Hundreds of Drosophila and Caenorhabditis elegans Transcription Factors. *Genetics* **208**, 937–949 (2018). URL http://www.genetics.org/content/208/3/937.

[20] Neph, S. *et al.* Circuitry and Dynamics of Human Transcription Factor Regulatory Networks. *Cell* **150**, 1274–1286 (2012). URL http://www.sciencedirect.com/science/article/pii/S0092867412006393.

[21] Ram, O. *et al.* Combinatorial patterning of chromatin regulators uncovered by genome-wide location analysis in human cells. *Cell* **147**, 1628–1639 (2011). URL http://www.ncbi.nlm.nih.gov/pmc/articles/PMC3312319/.

[22] Ernst, J. *et al.* Systematic analysis of chromatin state dynamics in nine human cell types. *Nature* **473**, 43–49 (2011). URL http://www.ncbi.nlm.nih.gov/pmc/articles/PMC3088773/.

[23] Hoffman, M. M. *et al.* Unsupervised pattern discovery in human chromatin structure through genomic segmentation. *Nature methods* **9**, 473–476 (2012). URL http://www.ncbi.nlm.nih.gov/pmc/articles/PMC3340533/.

[24] Hoffman, M. M. *et al.* Integrative annotation of chromatin elements from ENCODE data. *Nucleic Acids Research* **41**, 827–841 (2013). URL http://nar.oxfordjournals.org/content/41/2/827.

[25] Fiers, M. W. E. J. *et al.* Mapping gene regulatory networks from single-cell omics data. *Briefings in Functional Genomics* **17**, 246–254 (2018). URL https://academic.oup.com/bfg/article/17/4/246/4803107.

[26] Tang, F. *et al.* mRNA-Seq whole-transcriptome analysis of a single cell. *Nature Methods* **6**, 377–382 (2009). URL https://www.nature.com/articles/nmeth.1315.

[27] Klein, A. *et al.* Droplet Barcoding for Single-Cell Transcriptomics Applied to Embryonic Stem Cells. *Cell* **161**, 1187–1201 (2015). URL http://www.sciencedirect.com/science/article/pii/S0092867415005000.

[28] Macosko, E. *et al.* Highly Parallel Genome-wide Expression Profiling of Individual Cells Using Nanoliter Droplets. *Cell* **161**, 1202–1214 (2015). URL http://www.sciencedirect.com/science/article/pii/S0092867415005498.

[29] Zheng, G. X. Y. *et al.* Massively parallel digital transcriptional profiling of single cells. *Nature Communications* **8**, 14049 (2017). URL https://www.nature.com/articles/ncomms14049.

[30] Cao, J. *et al.* Comprehensive single-cell transcriptional profiling of a multicellular organism. *Science* **357**, 661–667 (2017). URL http://science.sciencemag.org/content/357/6352/661.

[31] Cao, J. *et al.* The single-cell transcriptional landscape of mammalian organogenesis. *Nature* **566**, 496 (2019). URL https://www-nature-com.offcampus.lib.washington.edu/articles/s41586-019-0969-x.

[32] Patel, A. P. *et al.* Single-cell RNA-seq highlights intratumoral heterogeneity in primary glioblastoma. *Science* **344**, 1396–1401 (2014). URL https://science-sciencemag-org.offcampus.lib.washington.edu/content/344/6190/1396.

[33] Packer, J. S. *et al.* A lineage-resolved molecular atlas of C. elegans embryogenesis at single cell resolution. *bioRxiv* 565549 (2019). URL https://www.biorxiv.org/content/10.1101/565549v2.

[34] Trapnell, C. *et al.* The dynamics and regulators of cell fate decisions are revealed by pseudotemporal ordering of single cells. *Nature Biotechnology* **32**, 381–386 (2014). URL http://www.nature.com/nbt/journal/v32/n4/full/nbt.2859.html.

[35] Buenrostro, J. D., Giresi, P. G., Zaba, L. C., Chang, H. Y. & Greenleaf, W. J. Transposition of native chromatin for fast and sensitive epigenomic profiling of open chromatin, DNA-binding proteins and nucleosome position. *Nature Methods* **10**, 1213–1218 (2013). URL http://www.nature.com.offcampus.lib.washington.edu/nmeth/journal/v10/n12/full/nmeth.2688.html.

[36] Corces, M. R. *et al.* An improved ATAC-seq protocol reduces background and enables interrogation of frozen tissues. *Nature Methods* **advance online publication** (2017). URL http://www.nature.com.offcampus.lib.washington.edu/nmeth/journal/vaop/ncurrent/full/nmeth.4396.html.

[37] Boyle, A. P. *et al.* High-Resolution Mapping and Characterization of Open Chromatin across the Genome. *Cell* **132**, 311–322 (2008). URL https://www.cell.com/cell/abstract/S0092-8674(07)01613-3.

[38] Buenrostro, J. D. *et al.* Single-cell chromatin accessibility reveals principles of regulatory variation. *Nature* **523**, 486–490 (2015). URL http://www.nature.com/articles/nature14590.

[39] Cusanovich, D. A. *et al.* Multiplex single-cell profiling of chromatin accessibility by combinatorial cellular indexing. *Science* **348**, 910–914 (2015). URL http://www.sciencemag.org.offcampus.lib.washington.edu/content/348/6237/910.

[40] Cusanovich, D. A. *et al.* The *cis*-regulatory dynamics of embryonic development at single-cell resolution. *Nature* (2018). URL https://www.nature.com/articles/nature25981.

[41] Cusanovich, D. A. *et al.* A Single-Cell Atlas of In Vivo Mammalian Chromatin Accessibility. *Cell* **174**, 1309–1324.e18 (2018). URL https://www.cell.com/cell/abstract/S0092-8674(18)30855-9.

[42] Sinnamon, J. R. *et al.* The accessible chromatin landscape of the murine hippocampus at single-cell resolution. *Genome Research* gr.243725.118 (2019). URL http://genome.cshlp.org.offcampus.lib.washington.edu/content/early/2019/04/01/gr.243725.118.

[43] Cao, J. *et al.* Joint profiling of chromatin accessibility and gene expression in thousands of single cells. *Science* **361**, 1380–1385 (2018). URL https://science-sciencemag-org.offcampus.lib.washington.edu/content/361/6409/1380.

[44] Ernst, J. & Kellis, M. Large-scale imputation of epigenomic datasets for systematic annotation of diverse human tissues. *Nature Biotechnology* **advance online publication** (2015). URL http://www.nature.com/nbt/journal/vaop/ncurrent/full/nbt.3157.html.

[45] Zhou, J. & Troyanskaya, O. G. Predicting effects of noncoding variants with deep learning-based sequence model. *Nature Methods* **12**, 931–934 (2015). URL http://www.nature.com/nmeth/journal/v12/n10/full/nmeth.3547.html.

[46] Wang, M., Tai, C., E, W. & Wei, L. DeFine: deep convolutional neural networks accurately quantify intensities of transcription factor-DNA binding and facilitate evaluation of functional non-coding variants. *Nucleic Acids Research* **46**, e69–e69 (2018). URL https://academic.oup.com/nar/article/46/11/e69/4958204.

[47] McInnes, L., Healy, J. & Melville, J. UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction. *arXiv:1802.03426 [cs, stat]* (2018). URL http://arxiv.org/abs/1802.03426. ArXiv: 1802.03426.

[48] Maaten, L. v. d. & Hinton, G. Visualizing data using t-SNE. *Journal of machine learning research* **9**, 2579–2605 (2008).

[49] Traag, V., Waltman, L. & van Eck, N. J. From Louvain to Leiden: guaranteeing well-connected communities. *arXiv:1810.08473 [physics]* (2018). URL http://arxiv.org/abs/1810.08473. ArXiv: 1810.08473.

[50] Qiu, X. *et al.* Single-cell mRNA quantification and differential analysis with Census. *Nature Methods* **14**, 309–315 (2017). URL https://www.nature.com/articles/nmeth.4150.

[51] Qiu, X. *et al.* Reversed graph embedding resolves complex single-cell trajectories. *Nature Methods* **14**, 979–982 (2017). URL https://www.nature.com/articles/nmeth.4402.

[52] Butler, A., Hoffman, P., Smibert, P., Papalexi, E. & Satija, R. Integrating single-cell transcriptomic data across different conditions, technologies, and species. *Nature Biotechnology* **36**, 411–420 (2018). URL http://www.nature.com/articles/nbt.4096.

[53] Stuart, T. *et al.* Comprehensive integration of single cell data. *bioRxiv* 460147 (2018). URL https://www.biorxiv.org/content/10.1101/460147v1.

[54] Gonzlez-Blas, C. B. *et al.* cisTopic: cis-regulatory topic modeling on single-cell ATAC-seq data. *Nature Methods* 1 (2019). URL https://www.nature.com/articles/s41592-019-0367-1.

[55] Deerwester, S., Dumais, S. T., Furnas, G. W., Landauer, T. K. & Harshman, R. Indexing by latent semantic analysis. *Journal of the American Society for Information Science* **41**, 391–407 (1990). URL https://onlinelibrary.wiley.com/doi/abs/10.1002/%28SICI%291097-4571%28199009%2941%3A6%3C391%3A%3AAID-ASI1%3E3.0.CO%3B2-9.

[56] Blei, D. M., Ng, A. Y. & Jordan, M. I. Latent Dirichlet Allocation. *J. Mach. Learn. Res.* **3**, 993–1022 (2003). URL http://dl.acm.org/citation.cfm?id=944919.944937.

[57] van Dijk, D. *et al.* Recovering Gene Interactions from Single-Cell Data Using Data Diffusion. *Cell* **174**, 716–729.e27 (2018). URL https://linkinghub.elsevier.com/retrieve/pii/S0092867418307244.

[58] Sulston, J., Schierenberg, E., White, J. & Thomson, J. The embryonic cell lineage of the nematode Caenorhabditis elegans. *Developmental Biology* **100**, 64–119 (1983). URL http://www.sciencedirect.com/science/article/pii/0012160683902014.

[59] Sulston, J. & Horvitz, H. Post-embryonic cell lineages of the nematode, Caenorhabditis elegans. *Developmental Biology* **56**, 110–156 (1977). URL http://www.sciencedirect.com/science/article/pii/0012160677901580.

[60] Waterston, R. & Sulston, J. The genome of Caenorhabditis elegans. *Proceedings of the National Academy of Sciences of the United States of America* **92**, 10836–10840 (1995). URL http://www.ncbi.nlm.nih.gov/pmc/articles/PMC40526/.

[61] Harshman, R. A. Foundations of the PARAFAC procedure: Models and conditions for an" explanatory" multi-modal factor analysis. *UCLA Working Papers in Phonetics* **16** (1970). URL http://www.psychology.uwo.ca/faculty/harshman/wpppfac0.pdf.

[62] Araya, C. L. *et al.* Regulatory analysis of the C. elegans genome with spatiotemporal resolution. *Nature* **512**, 400–405 (2014). URL http://www.nature.com/nature/journal/v512/n7515/full/nature13497.html.

[63] Ho, M. C. W., Quintero-Cadena, P. & Sternberg, P. W. Genome-wide discovery of active regulatory elements and transcription factor footprints in Caenorhabditis elegans using DNase-seq. *Genome Research* **27**, 2108–2119 (2017). URL http://genome.cshlp.org/content/27/12/2108.

[64] Daugherty, A. C. *et al.* Chromatin accessibility dynamics reveal novel functional enhancers in C. elegans. *Genome Research* (2017). URL http://genome.cshlp.org/content/early/2017/11/15/gr.226233.117.

[65] Jänes, J. *et al.* Chromatin accessibility dynamics across C. elegans development and ageing. *eLife* **7**, e37344 (2018). URL https://doi.org/10.7554/eLife.37344.

[66] Murray, J. I. *et al.* Multidimensional regulation of gene expression in the C. elegans embryo. *Genome Research* **22**, 1282–1294 (2012). URL http://genome.cshlp.org/content/22/7/1282.

[67] DiLeone, R. J., Russell, L. B. & Kingsley, D. M. An Extensive 3 Regulatory Region Controls Expression of Bmp5 in Specific Anatomical Structures of the Mouse Embryo. *Genetics* **148**, 401–408 (1998). URL https://www.genetics.org/content/148/1/401.

[68] Adey, A. *et al.* Rapid, low-input, low-bias construction of shotgun fragment libraries by high-density in vitro transposition. *Genome Biology* **11**, R119 (2010). URL http://genomebiology.com/content/11/12/R119/abstract.

[69] Song, L. & Crawford, G. E. DNase-seq: A High-Resolution Technique for Mapping Active Gene Regulatory Elements across the Genome from Mammalian Cells. *Cold Spring Harbor Protocols* **2010**, pdb.prot5384 (2010). URL http://cshprotocols.cshlp.org/content/2010/2/pdb.prot5384.

[70] Chen, X., Miragaia, R. J., Natarajan, K. N. & Teichmann, S. A. A rapid and robust method for single cell chromatin accessibility profiling. *Nature Communications* **9**, 5345 (2018). URL https://www.nature.com/articles/s41467-018-07771-0.

[71] Griffiths, T. L. & Steyvers, M. Finding scientific topics. *Proceedings of the National Academy of Sciences* **101**, 5228–5235 (2004). URL https://www.pnas.org/content/101/suppl_1/5228.

[72] Zhang, Y. *et al.* Model-based Analysis of ChIP-Seq (MACS). *Genome Biology* **9**, R137 (2008). URL https://doi.org/10.1186/gb-2008-9-9-r137.

[73] Green, B., Bouchier, C., Fairhead, C., Craig, N. L. & Cormack, B. P. Insertion site preference of Mu, Tn5, and Tn7 transposons. *Mobile DNA* **3**, 3 (2012). URL https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3292447/.

[74] Becht, E. *et al.* Dimensionality reduction for visualizing single-cell data using UMAP. *Nature Biotechnology* **37**, 38–44 (2019). URL http://www.nature.com/articles/nbt.4314.

[75] Krause, M., Fire, A., Harrison, S. W., Priess, J. & Weintraub, H. CeMyoD accumulation defines the body wall muscle cell fate during C. elegans embryogenesis. *Cell* **63**, 907–919 (1990). URL http://www.sciencedirect.com/science/article/pii/009286749090494Y.

[76] Page, B. D., Zhang, W., Steward, K., Blumenthal, T. & Priess, J. R. ELT-1, a GATA-like transcription factor, is required for epidermal cell fates in Caenorhabditis elegans embryos. *Genes & Development* **11**, 1651–1661 (1997). URL http://genesdev.cshlp.org/content/11/13/1651.

[77] Fukushige, T., Hawkins, M. G. & McGhee, J. D. The GATA-factor elt-2 is essential for formation of the Caenorhabditis elegans intestine. *Developmental Biology* **198**, 286–302 (1998). URL http://www.sciencedirect.com/science/article/pii/S0012160698800067.

[78] Reinke, V. Transcriptional regulation of gene expression in C. elegans. *WormBook* 1–31 (2013). URL http://www.wormbook.org/chapters/www_transcriptionalregulation.2/transregulate.html.

[79] Boeck, M. E. *et al.* The time-resolved transcriptome of C. elegans. *Genome Research* **26**, 1441–1450 (2016). URL http://genome.cshlp.org/content/26/10/1441.

[80] Welch, J. D., Hartemink, A. J. & Prins, J. F. MATCHER: manifold alignment reveals correspondence between single cell transcriptome and epigenome dynamics. *Genome Biology* **18**, 138 (2017). URL https://doi.org/10.1186/s13059-017-1269-0.

[81] Cook, D. E., Zdraljevic, S., Roberts, J. P. & Andersen, E. C. CeNDR, the Caenorhabditis elegans natural diversity resource. *Nucleic Acids Research* **45**, D650–D657 (2017). URL https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5210618/.

[82] Thompson, O. *et al.* The million mutation project: A new approach to genetics in Caenorhabditis elegans. *Genome Research* **23**, 1749–1762 (2013). URL http://genome.cshlp.org/content/23/10/1749.

[83] Bolger, A. M., Lohse, M. & Usadel, B. Trimmomatic: a flexible trimmer for Illumina sequence data. *Bioinformatics* **30**, 2114–2120 (2014). URL https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4103590/.

[84] Langmead, B. & Salzberg, S. L. Fast gapped-read alignment with Bowtie 2. *Nature Methods* **9**, 357–359 (2012). URL https://www.nature.com/articles/nmeth.1923.

[85] Quinlan, A. R. & Hall, I. M. BEDTools: a flexible suite of utilities for comparing genomic features. *Bioinformatics* **26**, 841–842 (2010). URL https://academic.oup.com/bioinformatics/article/26/6/841/244688/BEDTools-a-flexible-suite-of-utilities-for.

[86] Wallach, H. M., Murray, I., Salakhutdinov, R. & Mimno, D. Evaluation methods for topic models. In *Proceedings of the 26th Annual International Conference on Machine Learning - ICML '09*, 1–8 (ACM Press, Montreal, Quebec, Canada, 2009). URL http://portal.acm.org/citation.cfm?doid=1553374.1553515.

[87] Kent, W. J., Zweig, A. S., Barber, G., Hinrichs, A. S. & Karolchik, D. BigWig and BigBed: enabling browsing of large distributed datasets. *Bioinformatics* **26**, 2204–2207 (2010). URL https://academic.oup.com/bioinformatics/article/26/17/2204/199001.

[88] Wei, K., Libbrecht, M. W., Bilmes, J. A. & Noble, W. S. Choosing panels of genomics assays using submodular optimization. *Genome Biology* **17**, 229 (2016). URL http://dx.doi.org/10.1186/s13059-016-1089-7.

[89] Carroll, J. D. & Chang, J.-J. Analysis of individual differences in multidimensional scaling via an n-way generalization of Eckart-Young decomposition. *Psychometrika* **35**, 283–319 (1970). URL http://link.springer.com/article/10.1007/BF02310791.

[90] Capra, J. A., Erwin, G. D., McKinsey, G., Rubenstein, J. L. R. & Pollard, K. S. Many human accelerated regions are developmental enhancers. *Philosophical Transactions of the Royal Society B: Biological Sciences* **368** (2013). URL http://www.ncbi.nlm.nih.gov/pmc/articles/PMC3826498/.

[91] Zinkevich, M., Weimer, M., Li, L. & Smola, A. J. Parallelized stochastic gradient descent. In *Advances in neural information processing systems*, 2595–2603 (2010). URL http://papers.nips.cc/paper/4006-parallelized-stochastic-gradient-descent.

[92] Visel, A. *et al.* ChIP-seq accurately predicts tissue-specific activity of enhancers. *Nature* **457**, 854–858 (2009). URL http://www.nature.com.offcampus.lib.washington.edu/nature/journal/v457/n7231/full/nature07730.html.

[93] Hubisz, M. J. & Pollard, K. S. Exploring the genesis and functions of Human Accelerated Regions sheds light on their role in human evolution. *Current Opinion in Genetics & Development* **29**, 15–21 (2014). URL http://www.sciencedirect.com/science/article/pii/S0959437X14000781.

[94] King, M.-C. & Wilson, A. C. Evolution at two levels in humans and chimpanzees. *Science* **188**, 107–116 (1975). URL http://www.reed.edu/biology/professors/srenn/

pages/teaching/bio431s05_2008/431S05_readings/431s05_examples/king_wilson_1975(classic).pdf.

[95] Prabhakar, S. *et al.* Human-Specific Gain of Function in a Developmental Enhancer. *Science* **321**, 1346–1350 (2008). URL http://www.jstor.org/stable/20144754.

[96] Kamm, G. B., Pisciottano, F., Kliger, R. & Franchini, L. F. The Developmental Brain Gene NPAS3 Contains the Largest Number of Accelerated Regulatory Sequences in the Human Genome. *Molecular Biology and Evolution* **30**, 1088–1102 (2013). URL https://academic.oup.com/mbe/article-lookup/doi/10.1093/molbev/mst023.

[97] Oksenberg, N., Stevison, L., Wall, J. D. & Ahituv, N. Function and Regulation of AUTS2, a Gene Implicated in Autism and Human Evolution. *PLOS Genetics* **9**, e1003221 (2013). URL http://journals.plos.org/plosgenetics/article?id=10.1371/journal.pgen.1003221.

[98] Erwin, G. D. *et al.* Integrating Diverse Datasets Improves Developmental Enhancer Prediction. *PLoS Computational Biology* **10**, e1003677 (2014). URL http://dx.plos.org/10.1371/journal.pcbi.1003677.

[99] Pickard, B. S. *et al.* Interacting haplotypes at the NPAS3 locus alter risk of schizophrenia and bipolar disorder. *Molecular Psychiatry* **14**, 874–884 (2008). URL http://www.nature.com/mp/journal/v14/n9/full/mp200824a.html.

[100] Visel, A., Minovitsky, S., Dubchak, I. & Pennacchio, L. A. VISTA Enhancer Browsera database of tissue-specific human enhancers. *Nucleic Acids Research* **35**, D88–D92 (2007). URL http://www.ncbi.nlm.nih.gov/pmc/articles/PMC1716724/.

[101] McLean, C. Y. *et al.* GREAT improves functional interpretation of cis-regulatory regions. *Nature Biotechnology* **28**, 495–501 (2010). URL http://www.nature.com/doifinder/10.1038/nbt.1630.

[102] Bro, R. PARAFAC. Tutorial and applications. *Chemometrics and intelligent laboratory systems* **38**, 149–171 (1997). URL http://www.sciencedirect.com/science/article/pii/S0169743997000324.

[103] Kolda, T. G. & Bader, B. W. Tensor decompositions and applications. *SIAM review* **51**, 455–500 (2009). URL http://epubs.siam.org/doi/abs/10.1137/07070111X.

[104] Luo, Y., Wang, F. & Szolovits, P. Tensor factorization toward precision medicine. *Briefings in Bioinformatics* bbw026 (2016). URL https://academic.oup.com/bib/article-lookup/doi/10.1093/bib/bbw026.

[105] Zhu, Y. *et al.* Constructing 3d interaction maps from 1d epigenomes. *Nature Communications* **7**, 10812 (2016). URL http://www.nature.com/ncomms/2016/160310/ncomms10812/full/ncomms10812.html.

[106] Hore, V. *et al.* Tensor decomposition for multiple-tissue gene expression experiments. *Nature Genetics* **48**, 1094–1100 (2016). URL https://www.nature.com/articles/ng.3624.

[107] Acar, E., Dunlavy, D. M., Kolda, T. G. & Mrup, M. Scalable tensor factorizations for incomplete data. *Chemometrics and Intelligent Laboratory Systems* **106**, 41–56 (2011). URL http://www.sciencedirect.com/science/article/pii/S0169743910001437.

[108] Koren, Y., Bell, R. & Volinsky, C. Matrix Factorization Techniques for Recommender Systems. *IEEE Computer* **42**, 30–37 (2009). URL http://ieeexplore.ieee.org.offcampus.lib.washington.edu/xpl/articleDetails.jsp?tp=&arnumber=5197422&queryText%3Dmatrix+factorization+techniques+for+recommender+systems.

[109] Datta, V., Siddharthan, R. & Krishna, S. Detection Of Cooperatively Bound Transcription Factor Pairs Using ChIP-seq Peak Intensities And Expectation Maximization. *bioRxiv* 120113 (2017). URL http://biorxiv.org/content/early/2017/05/18/120113.

[110] Cremona, M. A. *et al.* Peak shape clustering reveals biological insights. *BMC Bioinformatics* **16** (2015). URL http://www.ncbi.nlm.nih.gov/pmc/articles/PMC4625869/.

[111] Schweikert, G., Cseke, B., Clouaire, T., Bird, A. & Sanguinetti, G. MMDiff: quantitative testing for shape changes in ChIP-Seq data sets. *BMC Genomics* **14**, 826 (2013). URL https://doi.org/10.1186/1471-2164-14-826.

[112] Benveniste, D., Sonntag, H.-J., Sanguinetti, G. & Sproul, D. Transcription factor binding predicts histone modifications in human cell lines. *Proceedings of the National Academy of Sciences* **111**, 13367–13372 (2014). URL http://www.pnas.org/content/111/37/13367.

[113] Whitaker, J. W., Chen, Z. & Wang, W. Predicting the human epigenome from DNA motifs. *Nature Methods* **12**, 265–272 (2015). URL http://www.nature.com/nmeth/journal/v12/n3/abs/nmeth.3065.html.

[114] Kingma, D. & Ba, J. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014). URL http://arxiv.org/abs/1412.6980.

[115] Dozat, T. Incorporating Nesterov Momentum into Adam. *Stanford University, Tech. Rep.* (2015). URL http://cs229.stanford.edu/proj2015/054_report.pdf.

[116] Jones, E., Oliphant, T., Peterson, P. & others. *SciPy: Open source scientific tools for Python* (2001). URL http://www.scipy.org/.

[117] Bergstra, J. & Bengio, Y. Random search for hyper-parameter optimization. *Journal of Machine Learning Research* **13**, 281–305 (2012). URL http://www.jmlr.org/papers/v13/bergstra12a.html.

[118] Snoek, J., Larochelle, H. & Adams, R. P. Practical bayesian optimization of machine learning algorithms. In *Advances in neural information processing systems*, 2951–2959 (2012). URL http://papers.nips.cc/paper/4522-practical.

[119] Recht, B. The News on Auto-tuning (2016). URL http://benjamin-recht.github.io/2016/06/20/hypertuning/.

[120] Harper, M. *et al.* python-ternary: Ternary Plots in Python. *Zenodo* URL https://zenodo.org/record/34938.

[121] Pedregosa, F. *et al.* Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research* **12**, 28252830 (2011). URL http://jmlr.csail.mit.edu/papers/v12/pedregosa11a.html.

[122] Schreiber, J., Durham, T. J., Bilmes, J. & Noble, W. S. Multi-scale deep tensor factorization learns a latent representation of the human epigenome. *bioRxiv* 364976 (2018). URL https://www.biorxiv.org/content/early/2018/07/08/364976.

[123] Dixit, A. *et al.* Perturb-Seq: Dissecting Molecular Circuits with Scalable Single-Cell RNA Profiling of Pooled Genetic Screens. *Cell* **167**, 1853–1866.e17 (2016). URL http://www.sciencedirect.com/science/article/pii/S0092867416316105.

[124] Adamson, B. *et al.* A Multiplexed Single-Cell CRISPR Screening Platform Enables Systematic Dissection of the Unfolded Protein Response. *Cell* **167**, 1867–1882.e21 (2016). URL https://www.cell.com/cell/abstract/S0092-8674(16)31660-9.

[125] Datlinger, P. *et al.* Pooled CRISPR screening with single-cell transcriptome readout. *Nature Methods* **14**, 297–301 (2017). URL http://www.nature.com/articles/nmeth.4177.

[126] Gasperini, M. *et al.* A Genome-wide Framework for Mapping Gene Regulation via Cellular Genetic Screens. *Cell* **176**, 377–390.e19 (2019). URL https://www.cell.com/cell/abstract/S0092-8674(18)31554-X.

[127] Au, V. *et al.* CRISPR/Cas9 Methodology for the Generation of Knockout Deletions in Caenorhabditis elegans. *G3: Genes, Genomes, Genetics* **9**, 135–144 (2019). URL https://www.g3journal.org/content/9/1/135.

[128] Shen, Z. *et al.* Conditional Knockouts Generated by Engineered CRISPR-Cas9 Endonuclease Reveal the Roles of Coronin in C. elegans Neural Development. *Developmental Cell* **30**, 625–636 (2014). URL https://www.cell.com/developmental-cell/abstract/S1534-5807(14)00483-3.

[129] Li, W., Yi, P. & Ou, G. Somatic CRISPRCas9-induced mutations reveal roles of embryonically essential dynein chains in Caenorhabditis elegans cilia. *J Cell Biol* **208**, 683–692 (2015). URL http://jcb.rupress.org/content/208/6/683.

[130] Cao, J. *et al.* An easy and efficient inducible CRISPR/Cas9 platform with improved specificity for multiple gene targeting. *Nucleic Acids Research* **44**, e149–e149 (2016). URL https://academic.oup.com/nar/article/44/19/e149/2468398.

[131] Rotem, A. *et al.* Single-cell ChIP-seq reveals cell subpopulations defined by chromatin state. *Nature Biotechnology* **33**, 1165–1172 (2015). URL http://www.nature.com/articles/nbt.3383.

[132] Skene, P. J., Henikoff, J. G. & Henikoff, S. Targeted *in situ* genome-wide profiling with high efficiency for low cell numbers. *Nature Protocols* **13**, 1006–1019 (2018). URL http://www.nature.com/articles/nprot.2018.015.

[133] Hainer, S. J., Bokovi, A., Rando, O. J. & Fazzio, T. G. Profiling of pluripotency factors in individual stem cells and early embryos. *bioRxiv* 286351 (2018). URL https://www.biorxiv.org/content/10.1101/286351v2.

[134] Bailey, T. L., Johnson, J., Grant, C. E. & Noble, W. S. The MEME Suite. *Nucleic Acids Research* **43**, W39–W49 (2015). URL https://academic.oup.com/nar/article/43/W1/W39/2467905.

[135] Weirauch, M. *et al.* Determination and Inference of Eukaryotic Transcription Factor Sequence Specificity. *Cell* **158**, 1431–1443 (2014). URL http://www.sciencedirect.com/science/article/pii/S0092867414010368.

[136] Farley, E. K. *et al.* Suboptimization of developmental enhancers. *Science* **350**, 325–328 (2015). URL http://science.sciencemag.org/content/350/6258/325.

[137] Farley, E. K., Olson, K. M., Zhang, W., Rokhsar, D. S. & Levine, M. S. Syntax compensates for poor binding sites to encode tissue specificity of developmental enhancers. *Proceedings*

of the National Academy of Sciences **113**, 6508–6513 (2016). URL http://www.pnas.org/lookup/doi/10.1073/pnas.1605085113.

[138] Liu, Y., Beyer, A. & Aebersold, R. On the Dependency of Cellular Protein Levels on mRNA Abundance. *Cell* **165**, 535–550 (2016). URL https://linkinghub.elsevier.com/retrieve/pii/S0092867416302707.

Appendix A: CHAPTER 2 SUPPLEMENT

A.1 Supplementary Information

Durham, et al. Comprehensive characterization of tissue-specific chromatin accessibility in L2 *Caenorhabditis elegans* nematodes.

A.2 Supplementary Figures

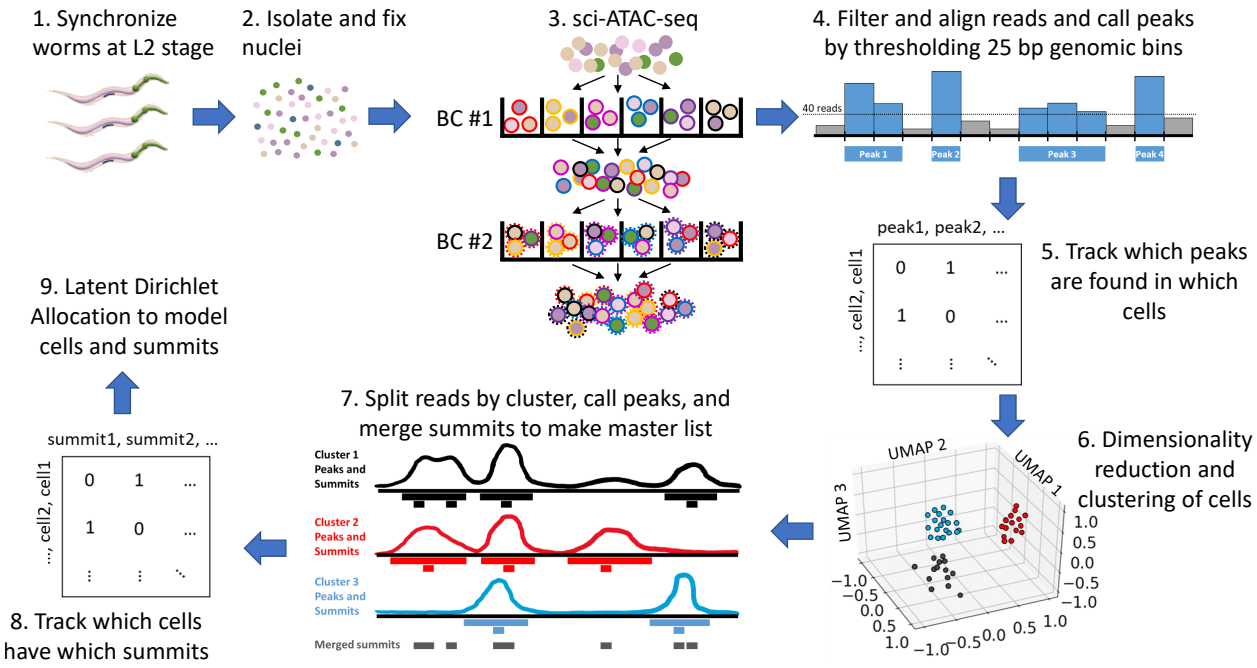

Figure A.1: **Overview of experimental design and analysis workflow.** Formaldehyde-fixed nuclei were isolated from synchronized L2 worms and assayed with sciATAC-seq. The sequencing data were filtered, aligned, and read pileups were used to identify genomic loci with elevated accessibility signal. A binary matrix summarizing which cells show accessibility in which regions was used to cluster the cells into 25 groups of similar cells. Reads associated with cells in each cluster were pooled, and statistically significant peaks and summits were called for each cluster, merged, and used to make a new binary matrix summarizing which cells show accessibility in which summits. This matrix was modeled with Latent Dirichlet Allocation and the modeling results were used for subsequent analysis.

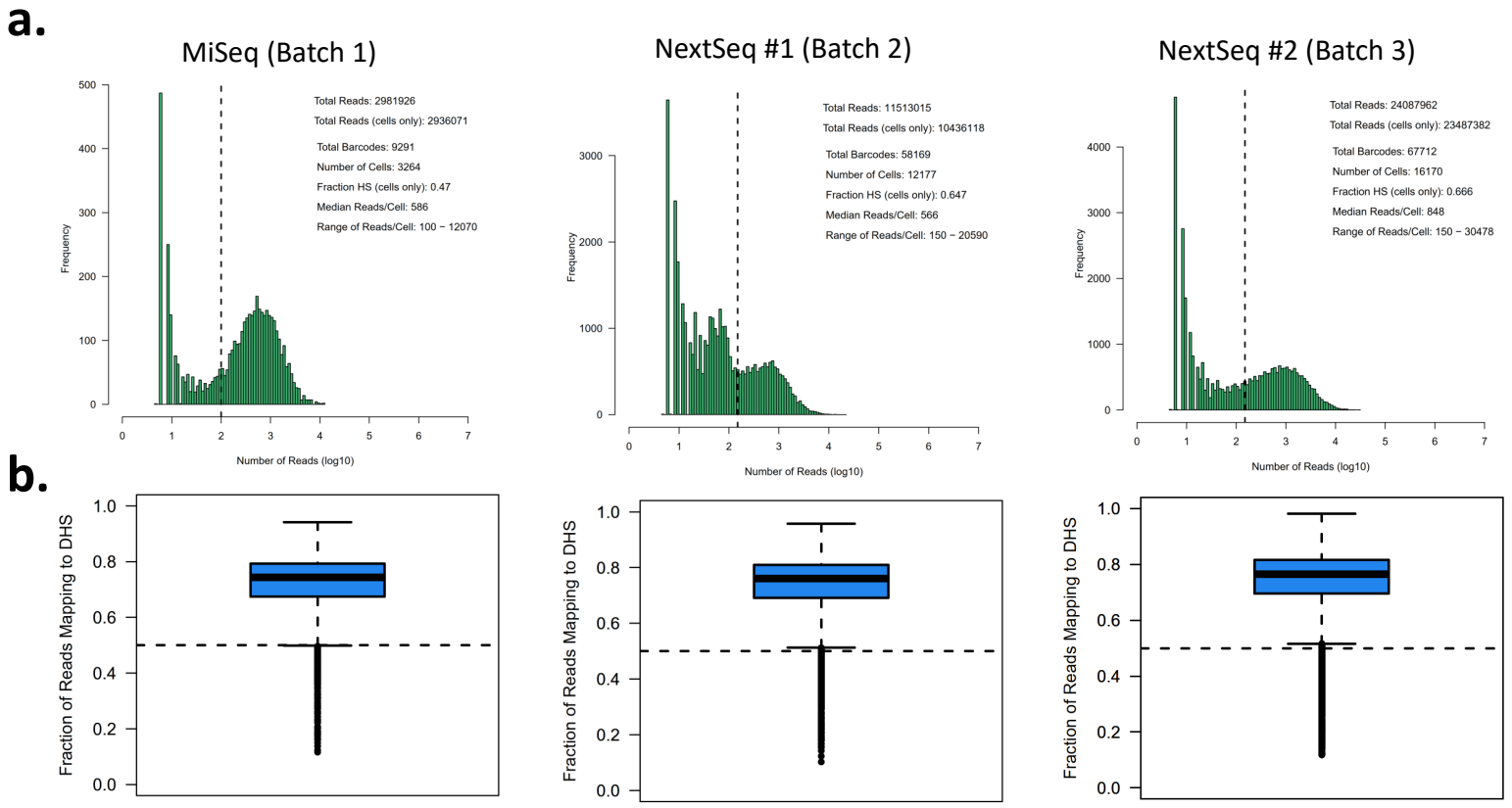

Figure A.2: **After thresholding cell coverage distribution, we recover a total of 31,611 cells from three sequencing batches. a.** The histograms of unique reads per cell barcode for each sequencing batch are shown. Batch 1 was a smaller pilot batch for which we set the read coverage threshold to 100 reads/cell. For the larger batches 2 and 3 we used a more stringent threshold of 150 reads/cell. **b.** Quality of the data were high as measured by the fraction of reads mapping in peaks (FRiP), which was about 0.75 for all three batches.

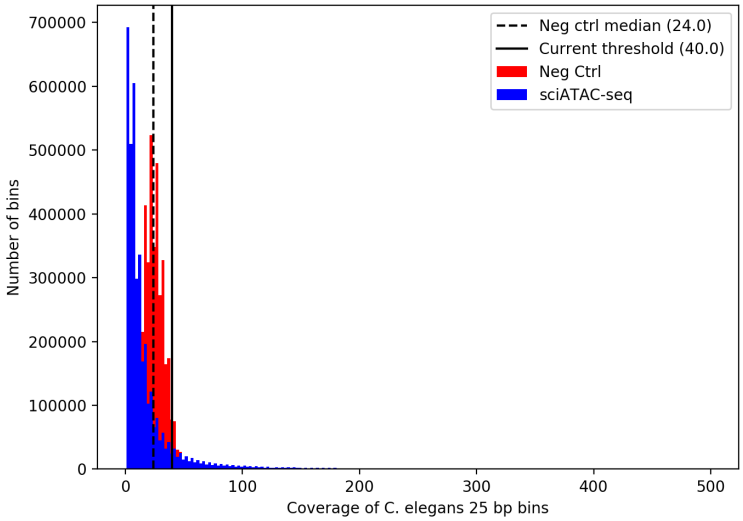

Figure A.3: **Thresholding of read coverage of genomic bins for initial peak calling.** Initial peaks were called by thresholding the number of cut sites overlapping 25 bp bins across the *C. elegans* genome. Comparing the bin coverage distribution of sciATAC-seq (blue histogram) with the bin coverage distribution of bulk ATAC-seq on naked genomic DNA (red histogram) led us to select a coverage threshold of 40 for calling bins enriched for signal.

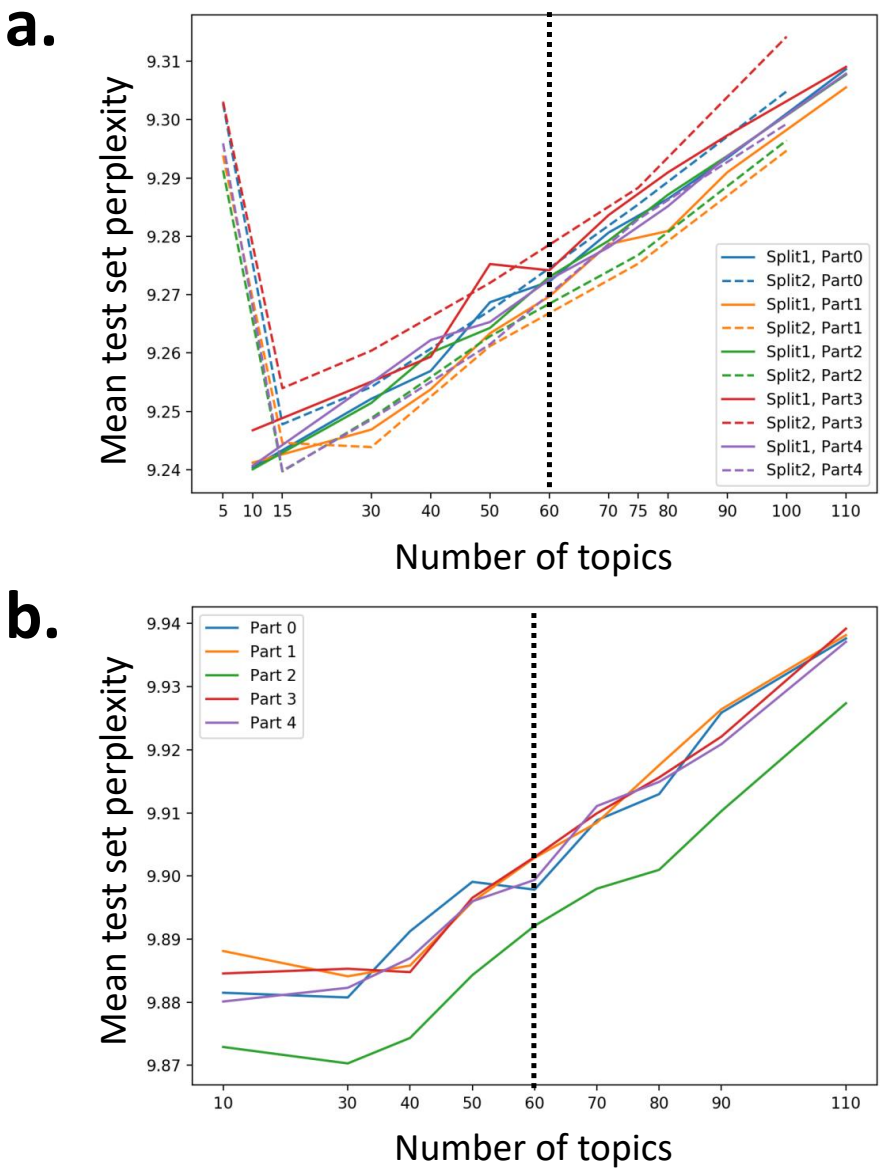

Figure A.4: **Tuning the number of topics using 5-fold cross validation.** We checked our setting of 60 topics by doing 5-fold cross validation in which we trained five LDA models for each number of topics, one for each fold of cells. We report the mean per-cell perplexity for the cells in each fold. Lower perplexity values are better, so although it seems like 60 topics is a local minimum for some folds, the results suggest that using fewer topics would result in a more generalizable model. (**a.**) Peaks called by thresholding 25 bp genomic bins. The first cross validation did not show any topics with a clear minimum perplexity, so we ran a second random split and cross validation with as few as 5 topics to check that the model got worse with too few topics. (**b.**) Peak summits called by MACS2.

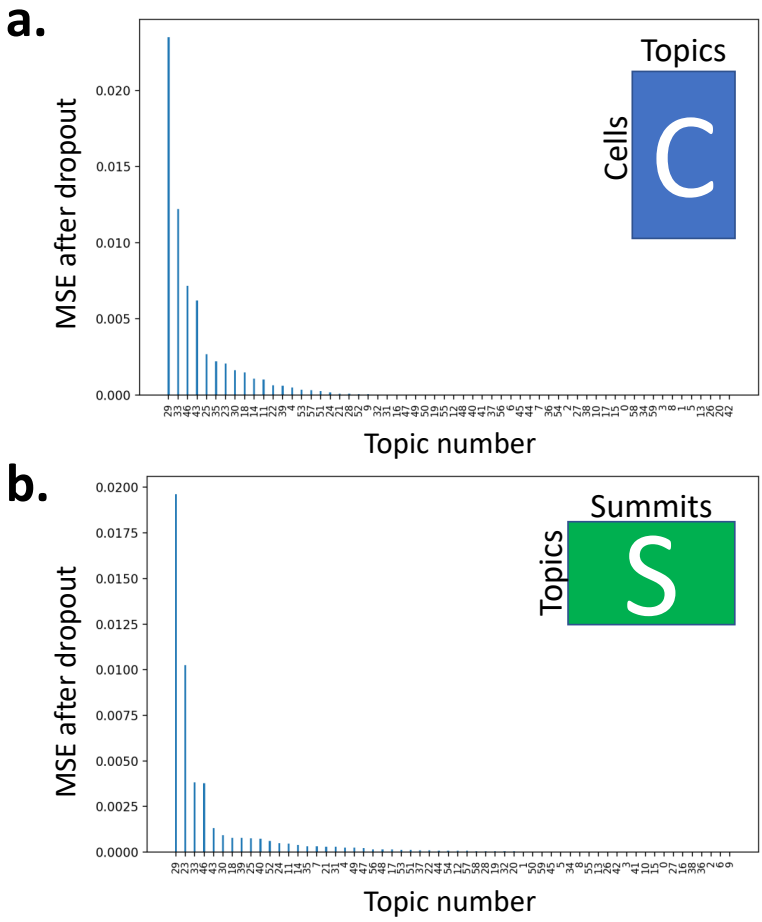

Figure A.5: **Topic contributions to the nuclei-by-topic and peak-by-topic matrices vary widely and are not the same.** To gain insight into which topics were most important after LDA training, we measured the pairwise distance for all pairs of cells (**a.**) and all pairs of summits (**b.**), and then repeatedly recomputed these pairwise distances, dropping out a single topic each time. The importance of each topic is indicated by calculating the mean squared error (MSE) between the original set of pairwise distances and the set with that topic dropped out.

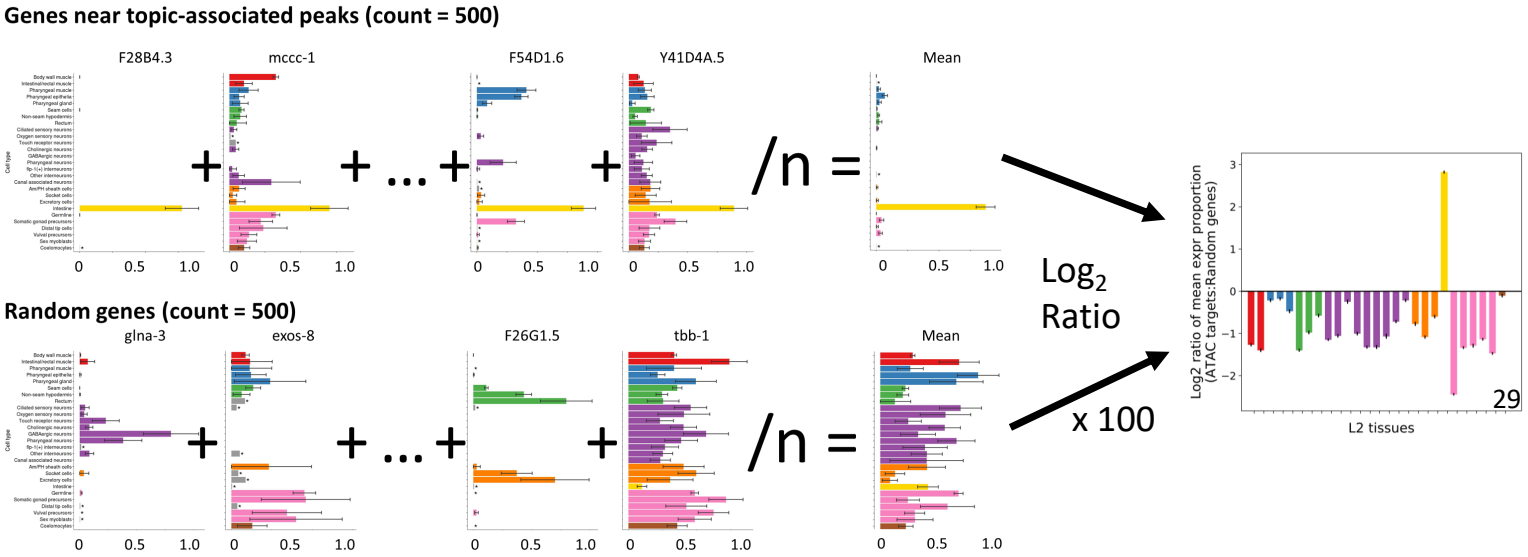

Figure A.6: **Schematic describing how to compute tissue enrichment.** For Fig. 2.4, we computed the tissue enrichment values by normalizing the tissue expression values for each gene to sum to one, then calculating the $\log_2$ ratio of the mean expression distribution for the top 500 genes by summit topic-specificity to the mean tissue expression distribution of 500 randomly-selected genes. This analysis was repeated for 100 random samples of 500 genes, and the mean $\log_2$ ratio was plotted with error bars indicating the 95% confidence interval for the enrichment of each tissue.

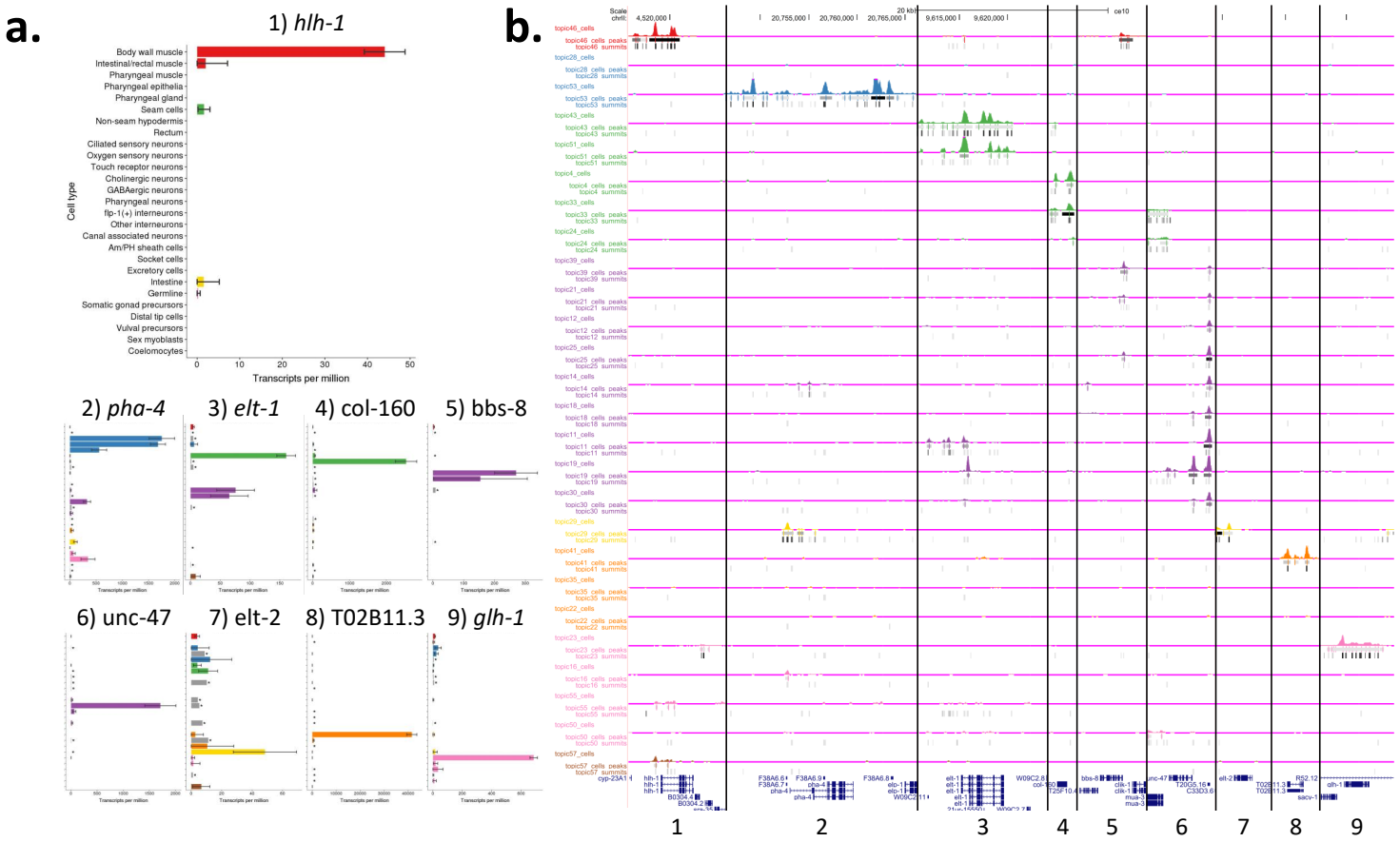

Figure A.7: **Multi-region browser shot demonstrating accessibility associated with tissue-specific genes. a.** Tissue expression distribution from sciRNA-seq[30] for 9 tissue-specific genes. **b.** UCSC Genome Browser Multiview showing the genomic loci for these 9 genes end-to-end. Three tracks are shown for each of 26 topics selected for analysis in this paper. The top track in each triplet is a signal track that shows the aggregated cut site coverage of cells from each topic cluster. Each signal track is paired with two BED tracks. The top BED in each pair indicates peaks called on the topic-specific data shown in the signal tracks, while the bottom one in each pair shows the sciATAC-seq summits used to define the topic clusters that led to the signal data. The color intensity of the peaks (top BED) indicates their level of statistical significance, while the color intensity of the summits (bottom BED) indicates their level of topic-specificity.

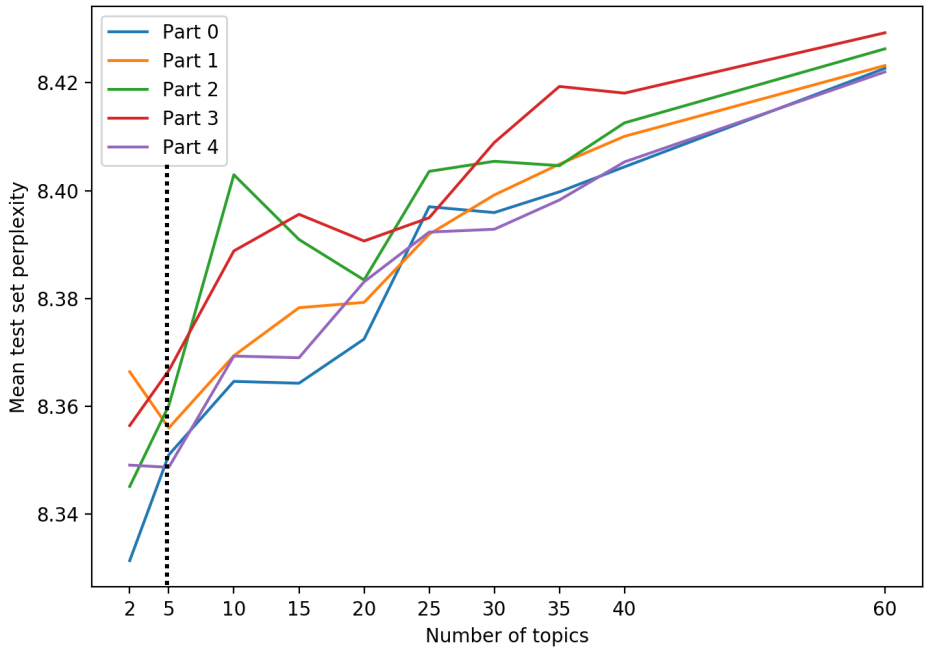

Figure A.8: **Tuning the number of topics for body wall muscle subclustering using 5-fold cross validation.** We ran a 5 fold cross validation and tested a range of topics for modeling the cells from topic 46 (bod wall muscle). We chose to use 5 topics for the analysis.

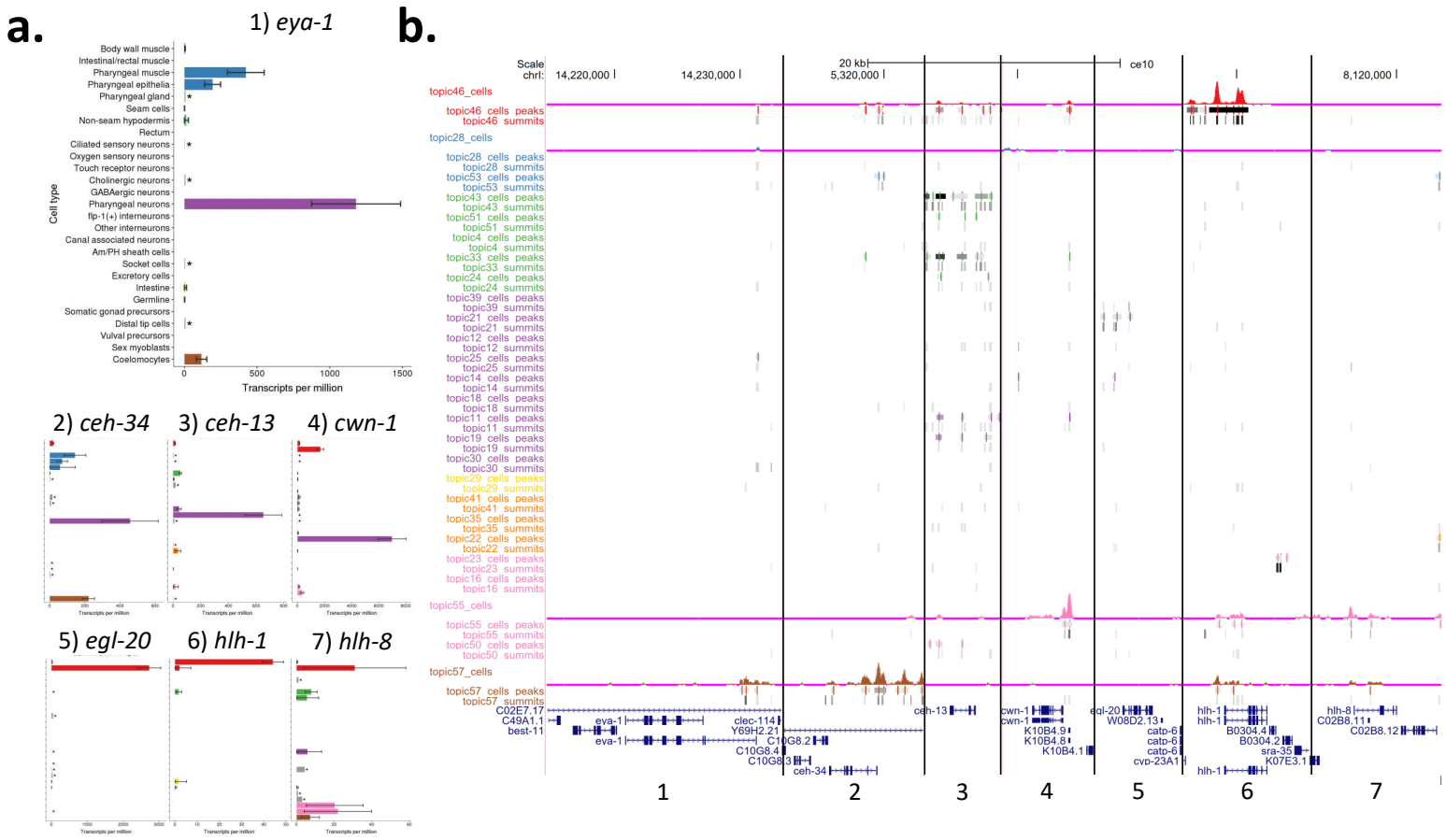

Figure A.9: **Multi-region browser shot demonstrating accessibility associated with muscle-specific genes.** **a.** Tissue expression distribution from sciRNA-seq[30] for 7 marker genes for types of muscle. **b.** UCSC Genome Browser Multiview showing the genomic loci for these 9 genes end-to-end. Track triplets for muscle-associated topics are as described in the legend for Fig. 2.6. The data shown in the browser here is expanded by showing the peak calls and summits from all 26 topics, not just the muscle genes.

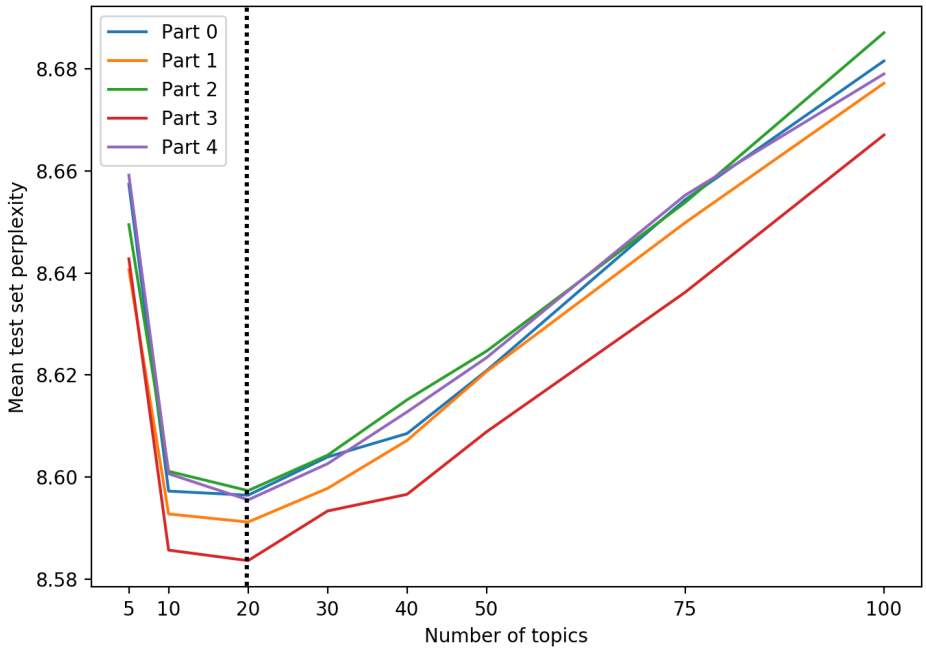

Figure A.10: **Tuning the number of topics for neuron subclustering using 5-fold cross validation.** We ran a 5 fold cross validation and tested a range of topics for modeling the cells from the neuron-associated topics (11, 12, 14, 18, 19, 21, 25, 30, 39). We chose to use 20 topics for the analysis.

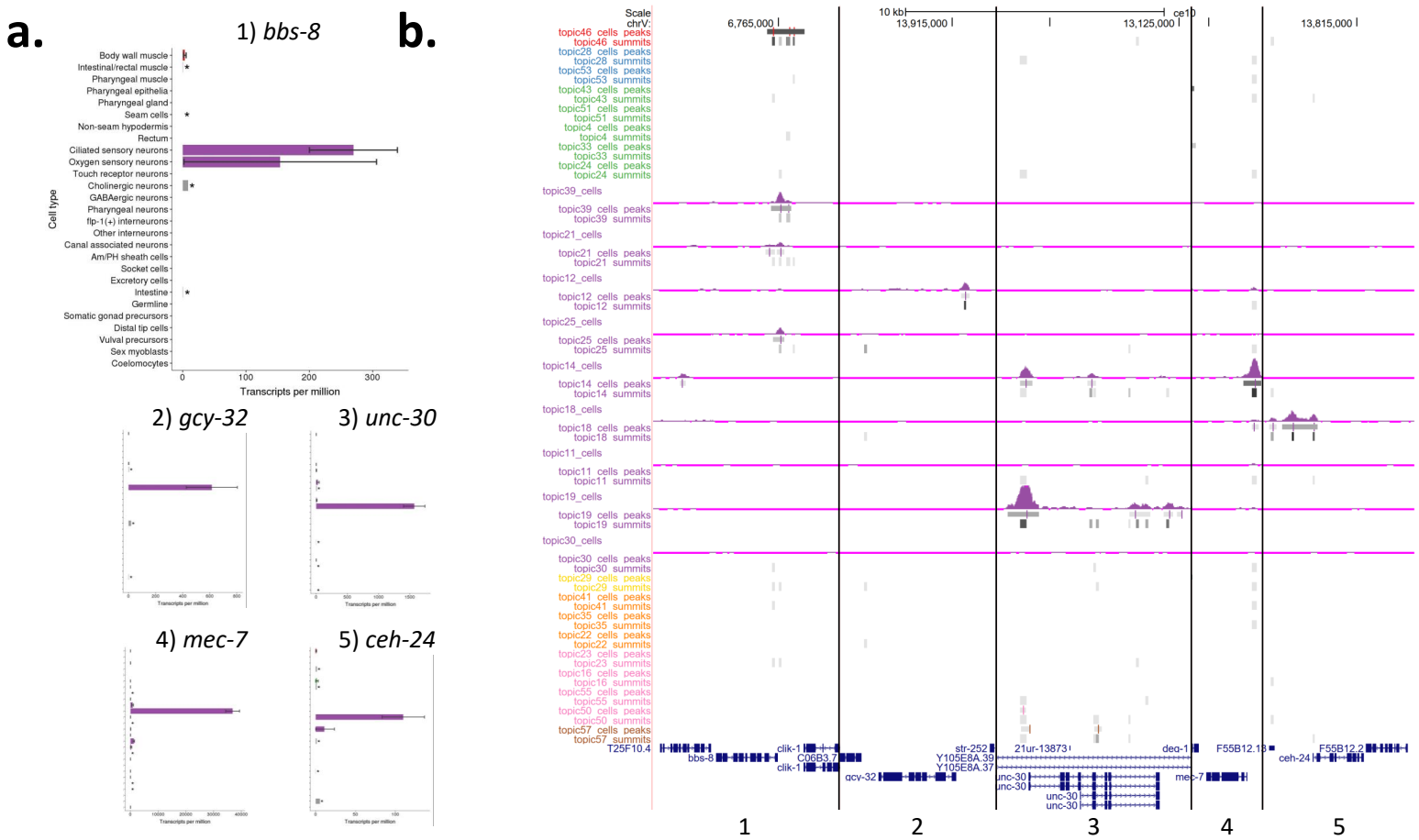

Figure A.11: **Multi-region browser shot demonstrating accessibility associated with neuron-specific genes. a.** Tissue expression distribution from sciRNA-seq[30] for 5 genes expression that is specific to different neuron subtypes. **b.** UCSC Genome Browser Multiview showing the genomic loci for these 5 genes. Track triplets for neuron-associated topics are as described in the legend for Fig. 2.6.

Appendix B: CHAPTER 3 SUPPLEMENT

B.1 SUPPLEMENTARY INFORMATION

Durham, et al. PREDICTD PaRallel Epigenomics Data Imputation with Cloud-based Tensor Decomposition

These supplementary files can be found with the published version of this paper:

Durham, Timothy J., Maxwell W. Libbrecht, J. Jeffry Howbert, Jeff Bilmes, and William Stafford Noble. PREDICTD PaRallel Epigenomics Data Imputation with Cloud-Based Tensor Decomposition. Nature Communications 9, no. 1 (April 11, 2018): 1402. https://doi.org/10.1038/s41467-018-03635-9.

- Supplementary Data 1 - Correlation and log ratio for all quality measures
- Supplementary Data 2 - Cell type clusters from spectral biclustering of imputed and observed data
- Supplementary Data 3 - ncHAR clusters from spectral biclustering of imputed and observed data
- Supplementary Data 4 - ncHARs with validated enhancer activity from the literature
- Supplementary Data 5 - GREAT analysis results for ncHARs from the “No Signal” cluster, whole genome background
- Supplementary Data 6 - GREAT analysis results for ncHARs from the “Brain/ES” cluster, whole genome background
- Supplementary Data 7 - GREAT analysis results for ncHARs from the “Brain/ES” cluster, all ncHARs background
- Supplementary Data 8 - GREAT analysis results for ncHARs from the “Epithelial/Mesenchymal” cluster, whole genome background
- Supplementary Data 9 - GREAT analysis results for ncHARs from the “Epithelial/Mesenchymal” cluster, all ncHARs background

- Supplementary Data 10 - GREAT analysis results for ncHARs from the “Non-immune” cluster, whole genome background
- Supplementary Data 11 - GREAT analysis results for ncHARs from the “Non-immune” cluster, all ncHARs background
- Supplementary Data 12 - GREAT analysis results for ncHARs from the “Immune” cluster, whole genome background
- Supplementary Data 13 - PREDICTD quality measures by experiment
- Supplementary Data 14 - Main Effects quality measures by experiment
- Supplementary Data 15 - ChromImpute quality measures by experiment
- Supplementary Data 16 - PREDICTD and ChromImpute Average Model quality measures by experiment
- Supplementary Data 17 - ENCODE accession identifiers for PREDICTD imputed data sets

B.2 Supplementary Figures

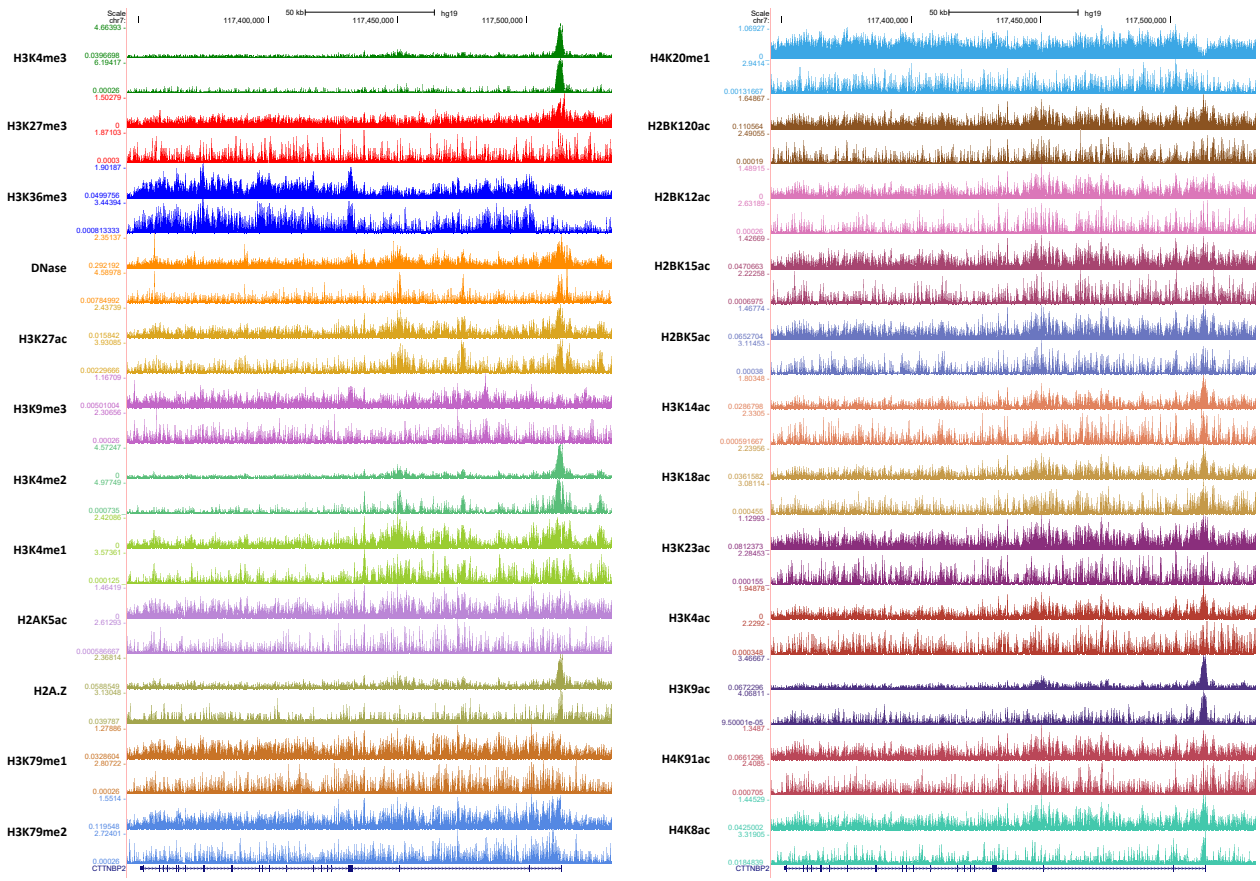

Figure B.1: **Example tracks show accuracy and diversity of imputed signals.** Tracks showing paired PREDICTD (top) and observed (bottom) data for the H1 Cell Line cell type, which is one of only three cell types for which observed data are available for all assays. The signal is variance stabilized with the inverse hyperbolic sine transform, and the tracks are auto-scaled by the genome browser to highlight the shape of the signal.

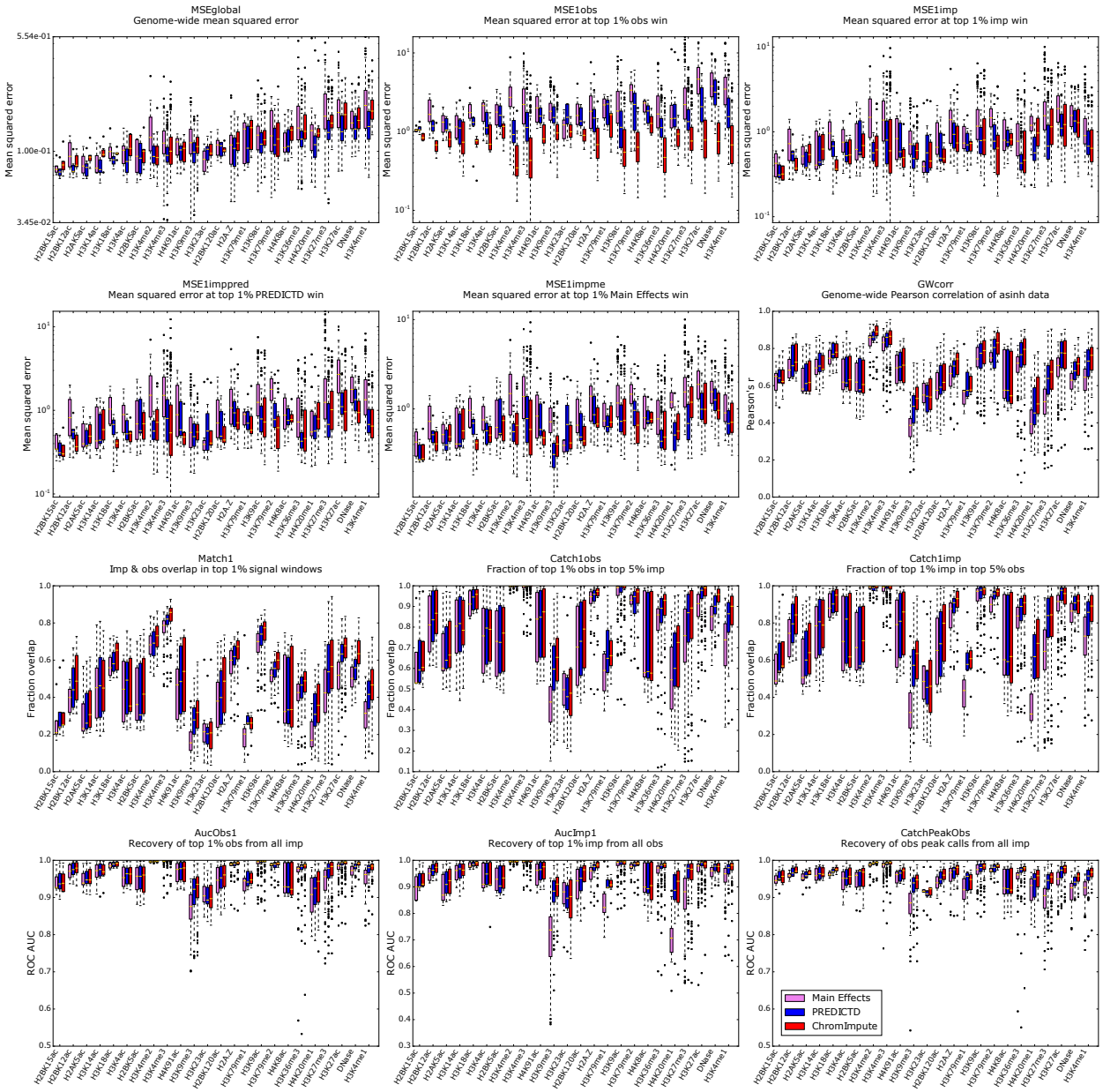

Figure B.2: **Plots for all quality measures evaluating the performance of three imputation methods.** Box plots describe the distribution of quality measure values for Main Effects (pink), PREDICTD (blue), and ChromImpute (red) for each assay. Each plot shows a different quality measure, and for each distribution of scores the box shows the inter-quartile range (IQR), whiskers show 1.5 times the IQR, and flier points show scores for individual experiments that are outliers. The median is indicated by a horizontal gold line on each box plot.

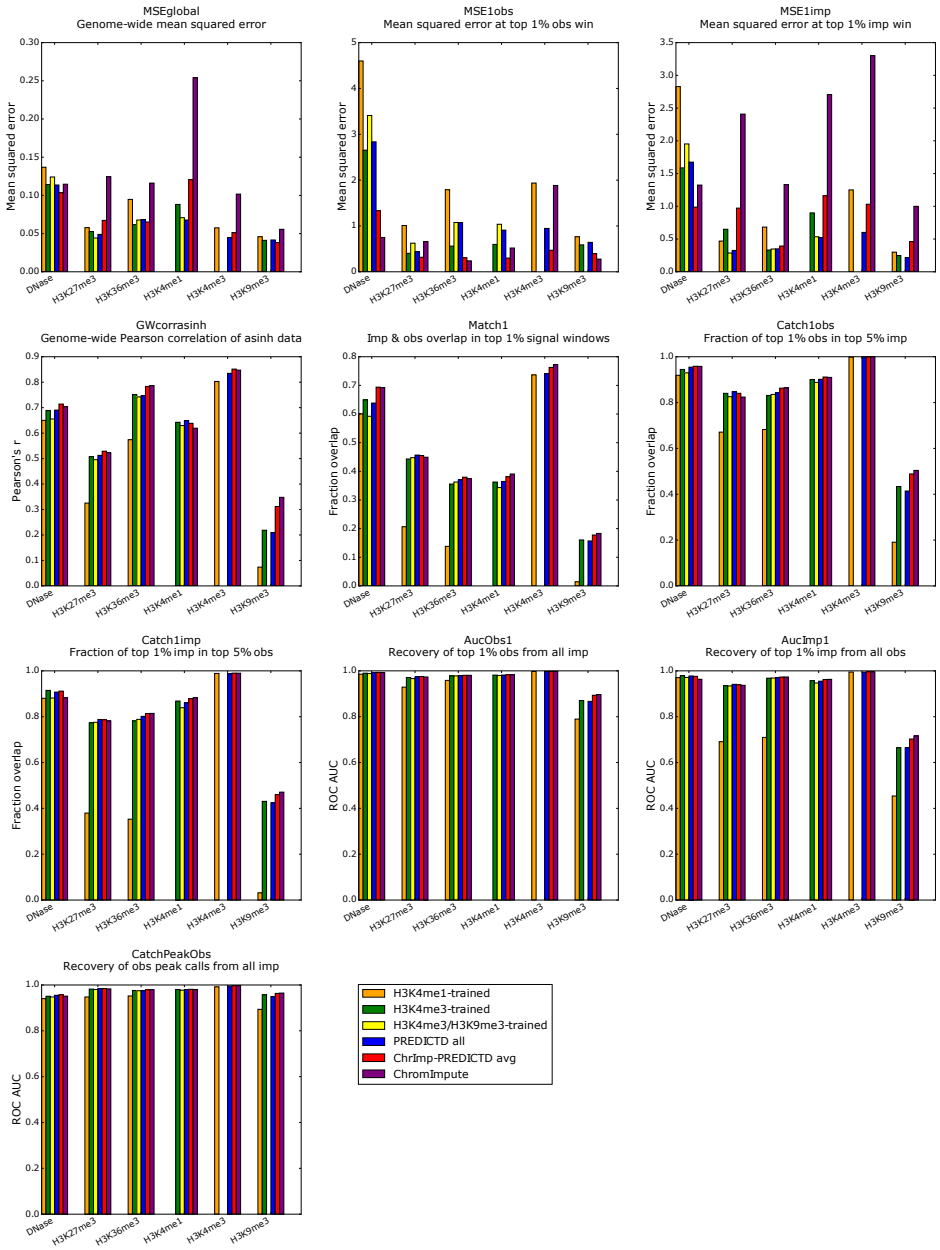

Figure B.3: **PREDICTD can still impute accurate data for "CD3 Primary Cells from Cord Blood" even with only one or two assays included in the training set.** Each plot shows the data for a different quality measure, and the bars compare the results of running PREDICTD with just H3K4me1 (orange), just H3K4me3 (green), or H3K4me3 and H3K9me3 (yellow) data included in the training set for "CD3 Primary Cells from Cord Blood". The quality of these imputation results are compared with the full PREDICTD imputation results (blue), the average of PREDICTD and ChromImpute (red), and ChromImpute alone (purple).

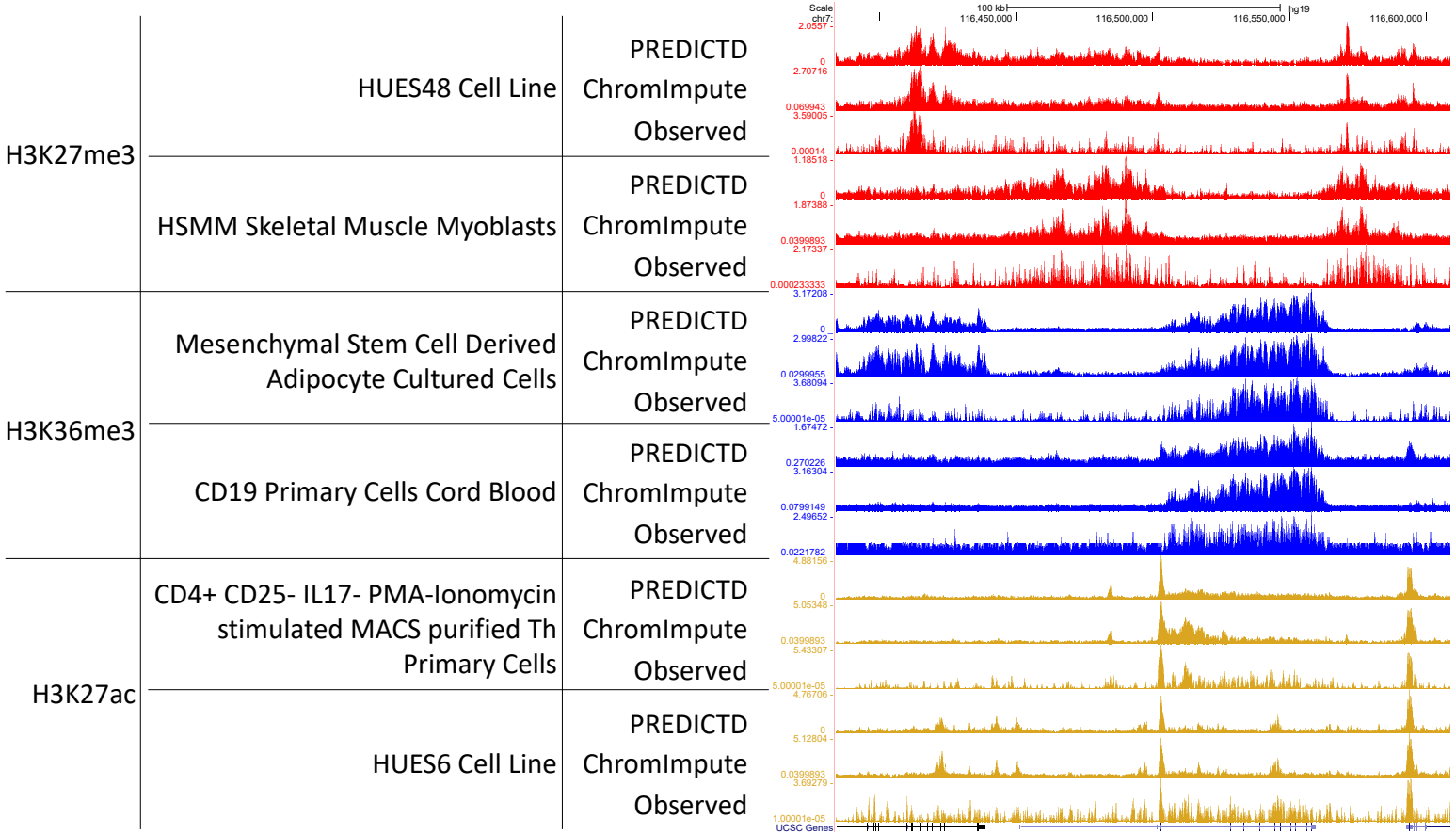

Figure B.4: **Selected tracks comparing PREDICTD imputed signal, ChromImpute imputed signal, and observed signal for three assays and six cell types.**

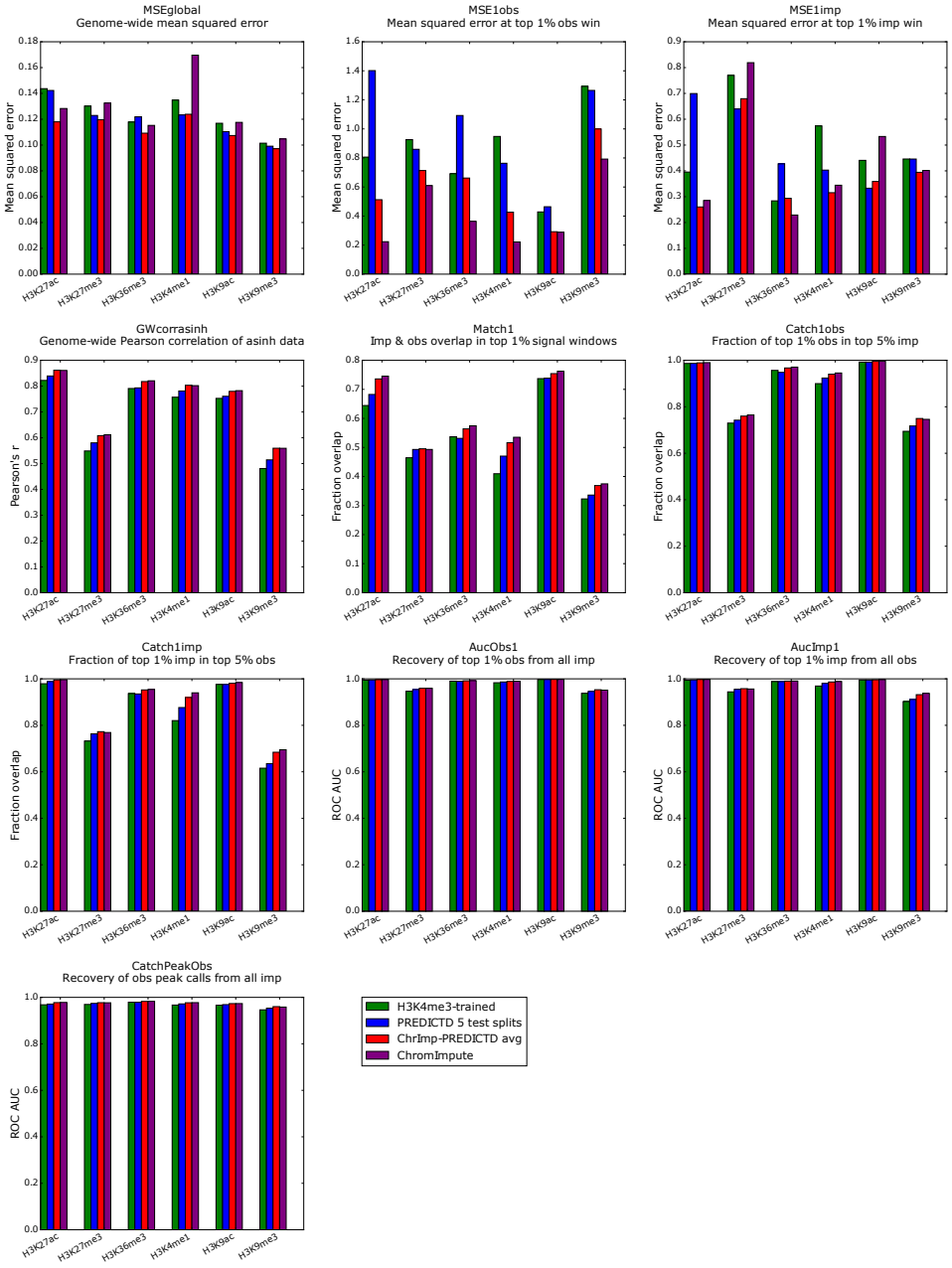

Figure B.5: **PREDICTD can still impute accurate data for "Brain Anterior Caudate" even with only the H3K4me3 assay included in the training set.** Each plot shows the data for a different quality measure, and the bars compare the results of running PREDICTD with just H3K4me3 (green) data included in the training set for "Brain Anterior Caudate". The quality of these imputation results are compared with the full PREDICTD imputation results (blue), the average of PREDICTD and ChromImpute (red), and ChromImpute alone (purple).

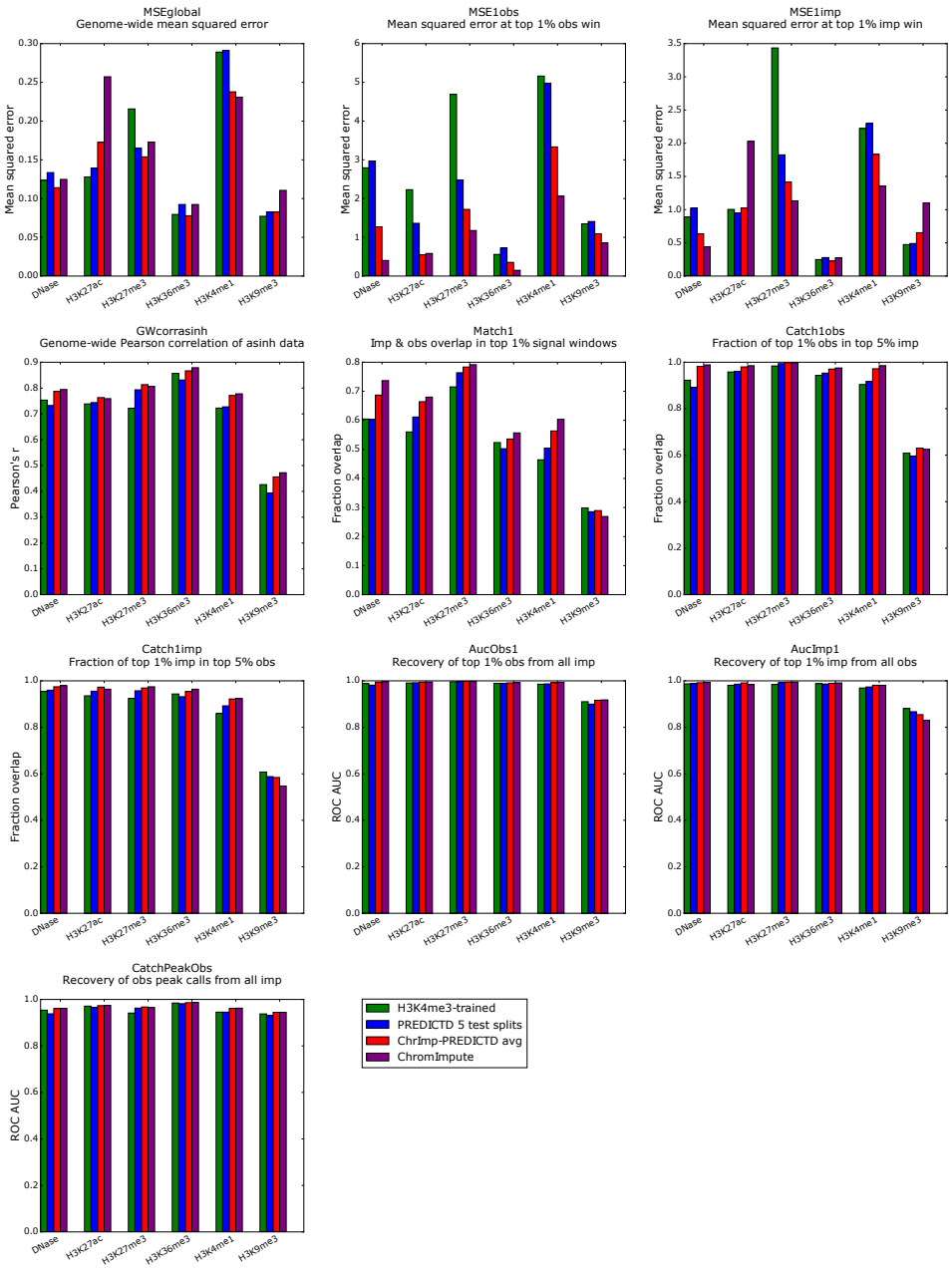

Figure B.6: **PREDICTD can still impute accurate data for "Fetal Muscle Trunk" even with only the H3K4me3 assay included in the training set.** Each plot shows the data for a different quality measure, and the bars compare the results of running PREDICTD with just H3K4me3 (green) data included in the training set for "Fetal Muscle Trunk". The quality of these imputation results are compared with the full PREDICTD imputation results (blue), the average of PREDICTD and ChromImpute (red), and ChromImpute alone (purple).

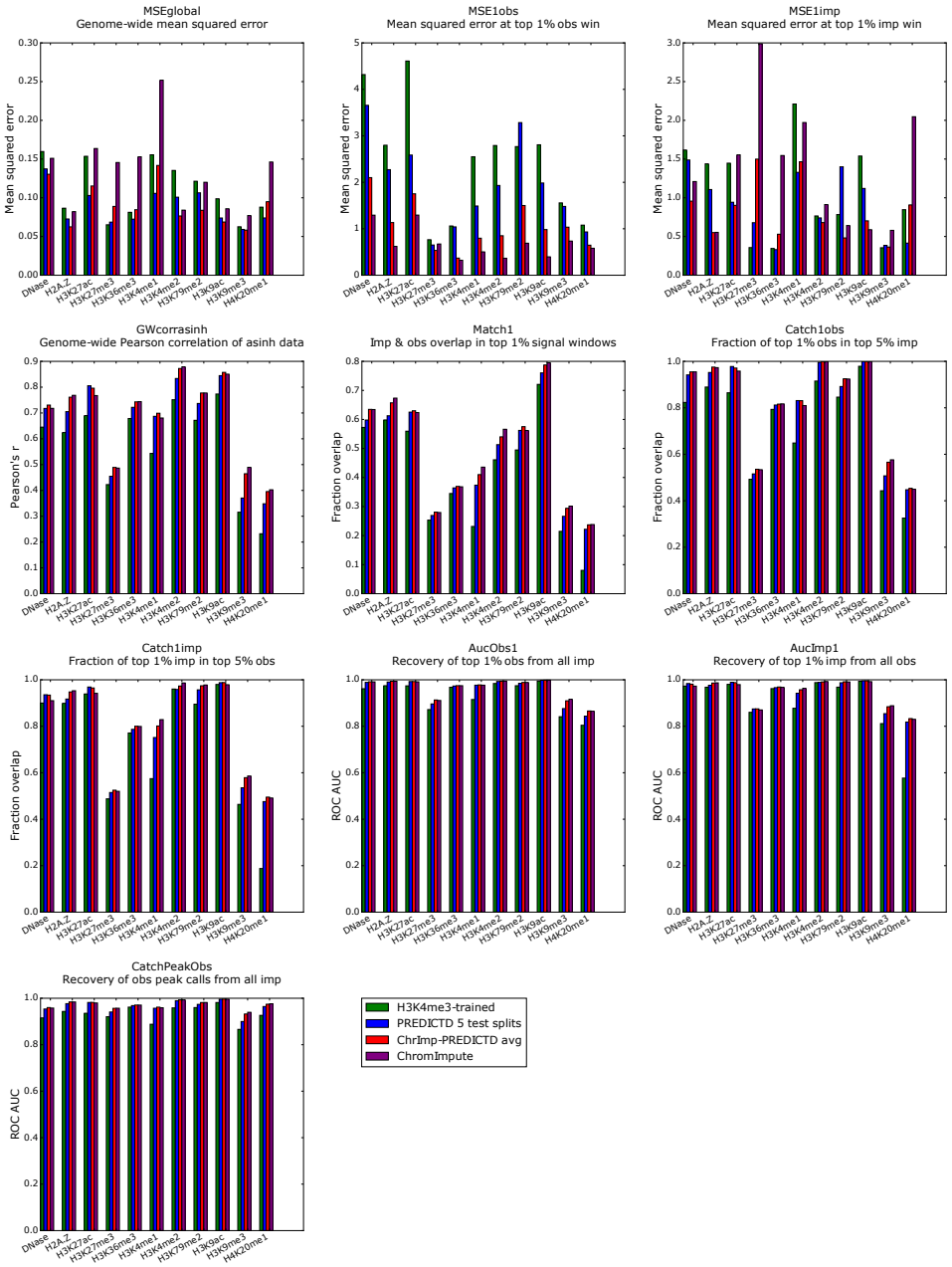

Figure B.7: **PREDICTD can still impute accurate data for "GM12878 Lymphoblastoid" even with only the H3K4me3 assay included in the training set.** Each plot shows the data for a different quality measure, and the bars compare the results of running PREDICTD with just H3K4me3 (green) data included in the training set for "GM12878 Lymphoblastoid". The quality of these imputation results are compared with the full PREDICTD imputation results (blue), the average of PREDICTD and ChromImpute (red), and ChromImpute alone (purple).

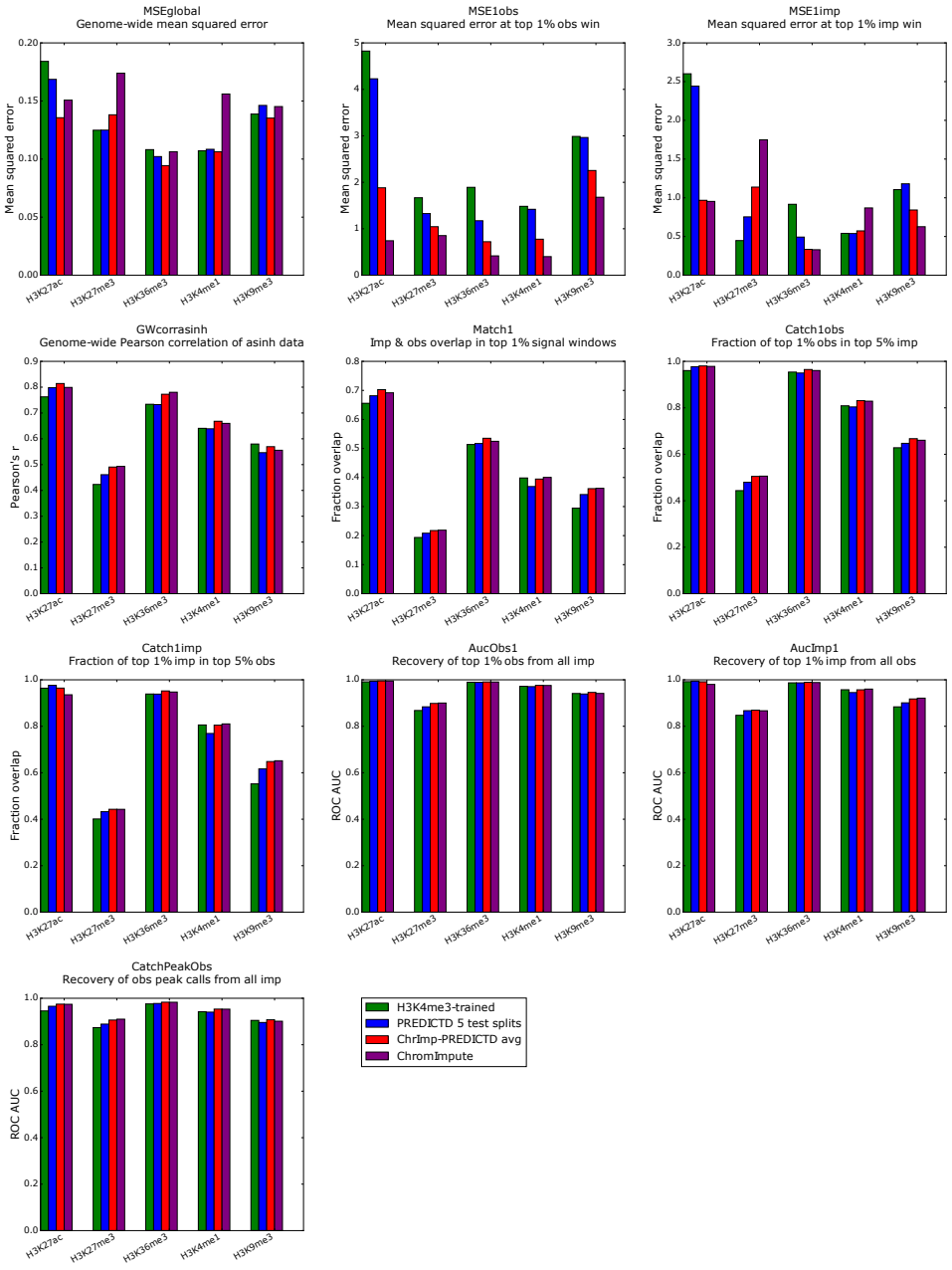

Figure B.8: **PREDICTD can still impute accurate data for "Lung" even with only the H3K4me3 assay included in the training set.** Each plot shows the data for a different quality measure, and the bars compare the results of running PREDICTD with just H3K4me3 (green) data included in the training set for "Lung". The quality of these imputation results are compared with the full PREDICTD imputation results (blue), the average of PREDICTD and ChromImpute (red), and ChromImpute alone (purple).

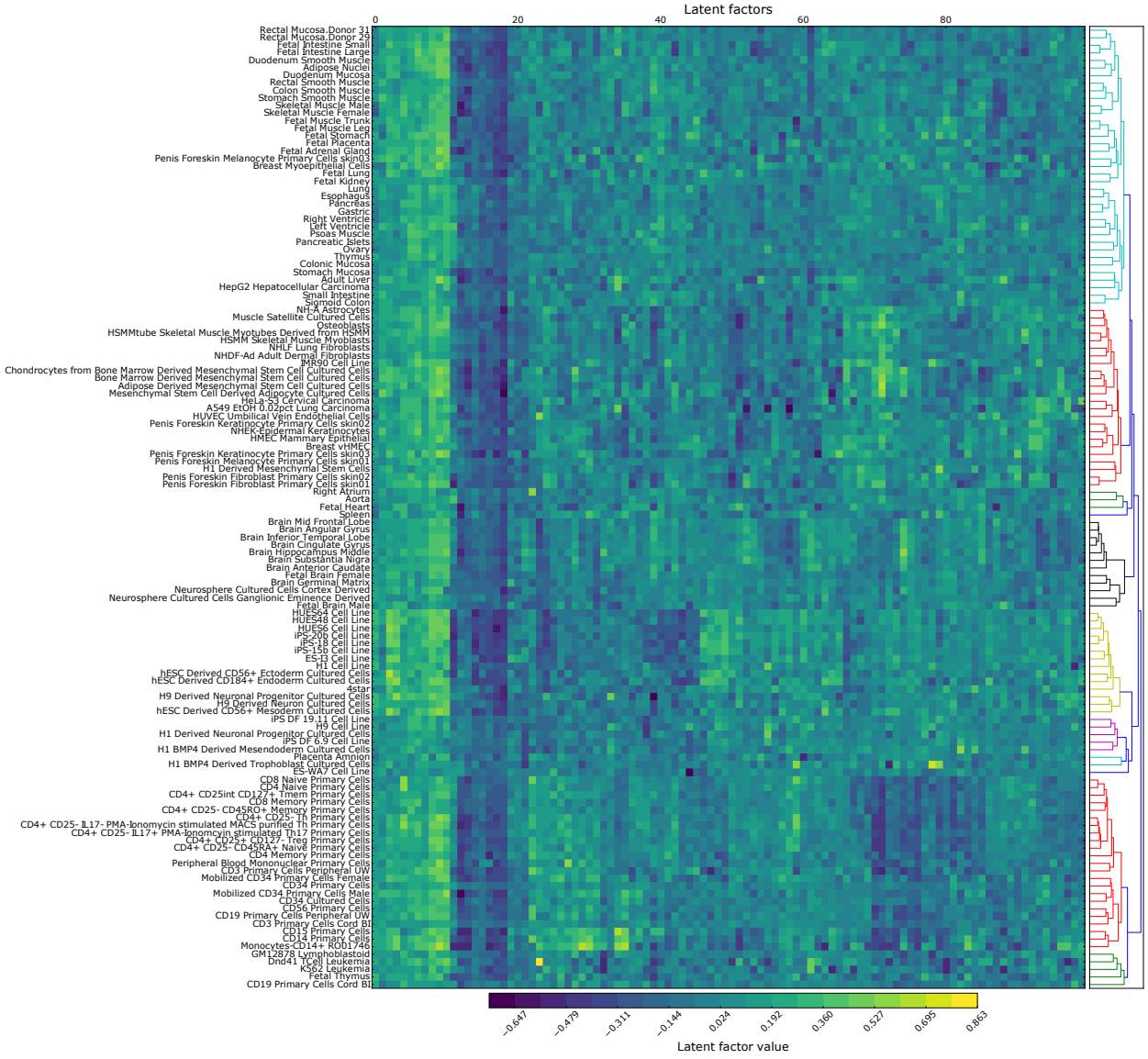

Figure B.9: **PREDICTD parameter values can distinguish among cell types.** Hierarchical clustering of all cell types based on parameter values from the cell type factor matrix. For legibility, Fig. 3.3a shows only the lower 62 rows of this figure.

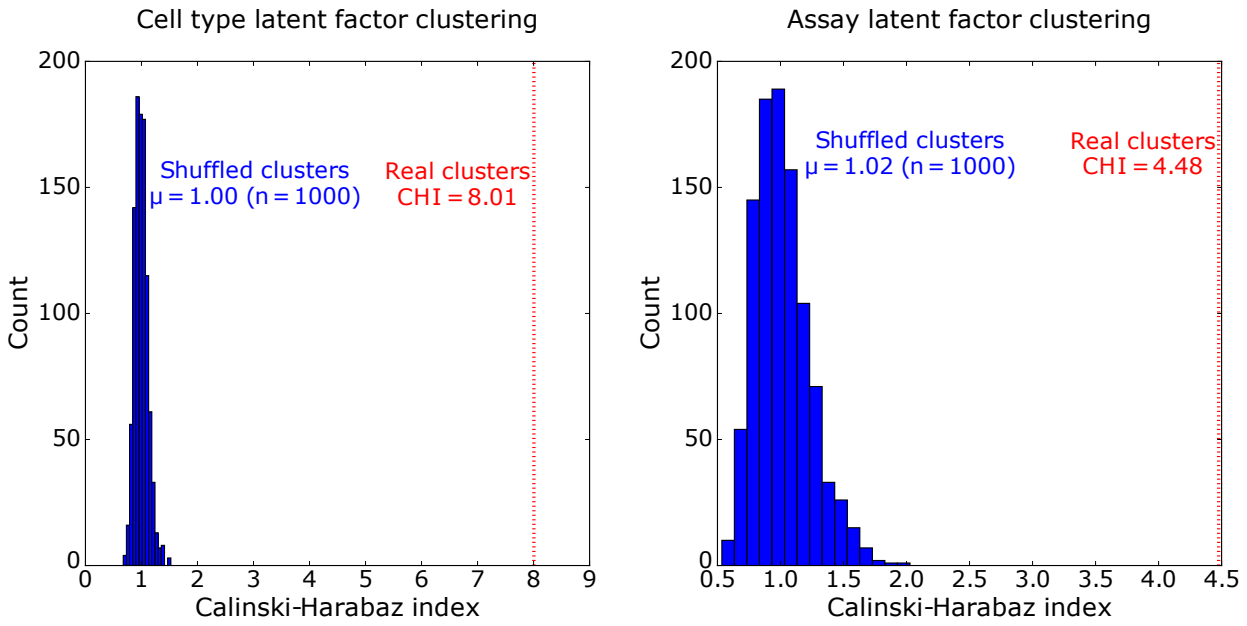

Figure B.10: **Hierarchical clustering of cell types and assays by latent factor parameter values is highly non-random.** Separation of clusters was tested using the Calinski-Harabaz Index after randomly assigning cluster identities to cell types and assays. This was repeated 1000 times and compared with the separation achieved by the true latent factor clustering. A higher value on the Calinski-Harabaz Index indicates that the clusters are denser and better-separated. Linkage trees were cut at eight clusters for cell types and four clusters for assays.

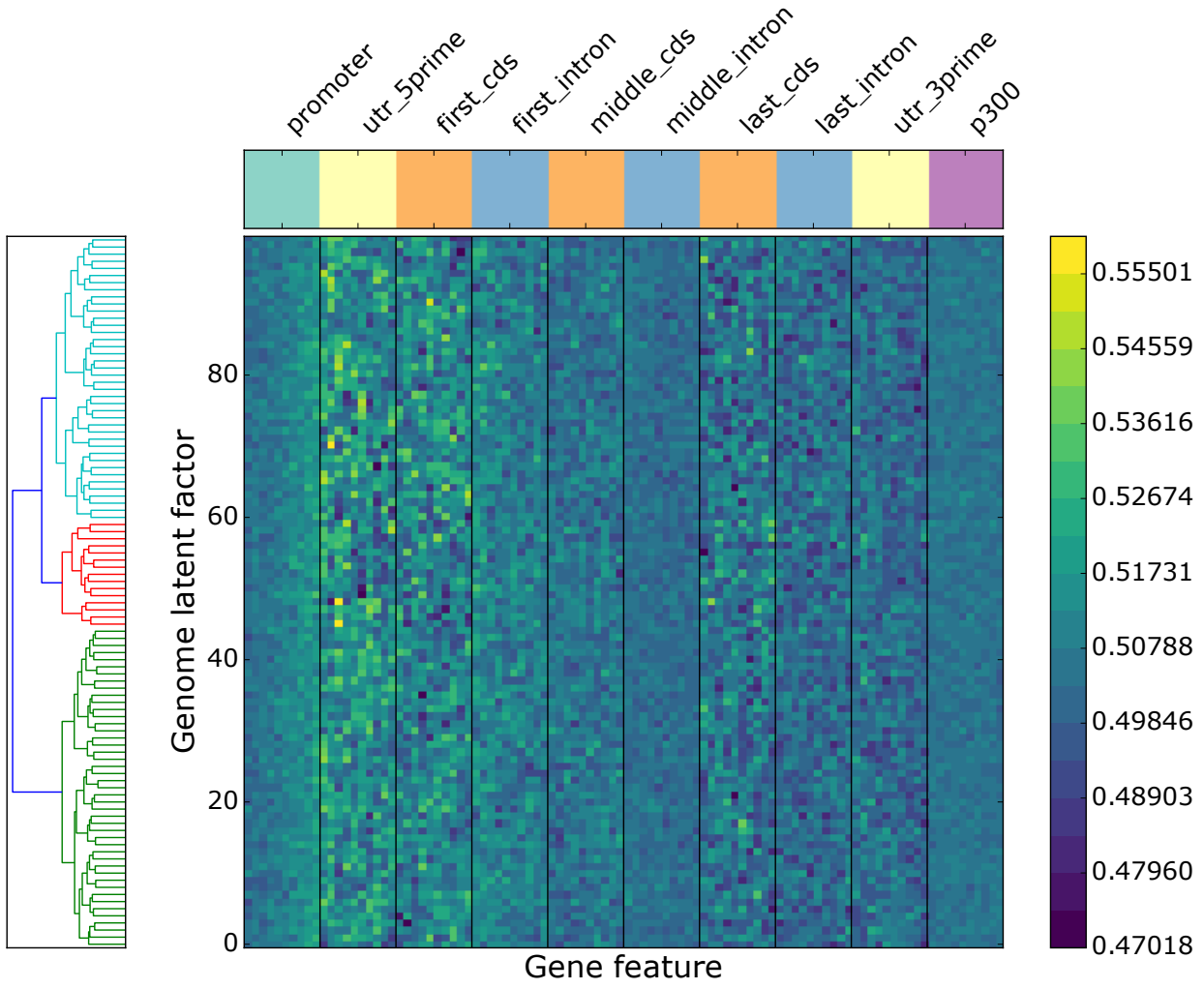

Figure B.11: **Patterns in the average latent factor values at different classes of genomic elements are non-random.** The same analysis was completed as in Fig. 3.3c, but after randomly permuting the latent factors at each genomic position. Randomly permuted latent factors do not show distinct patterns at different genomic elements.

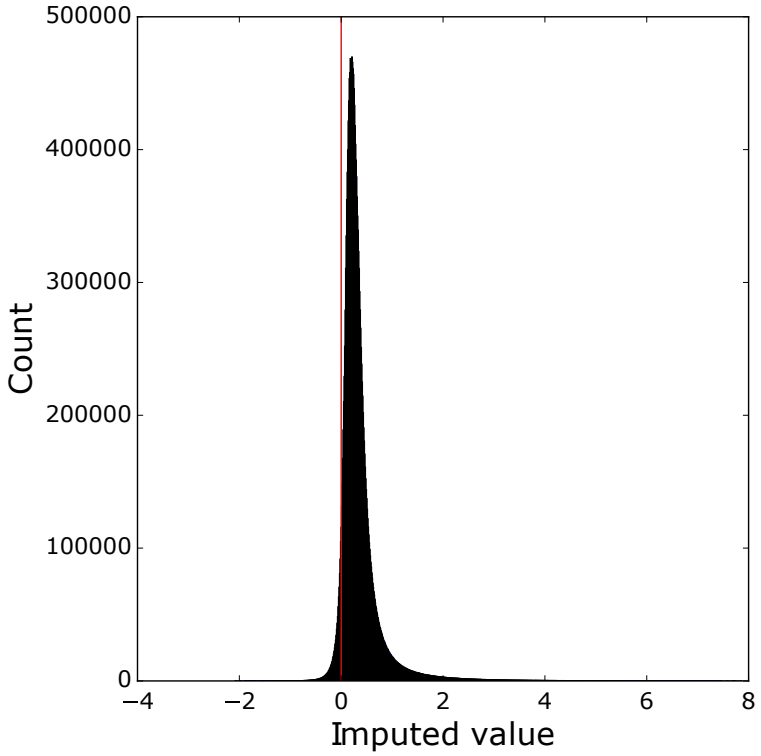

Figure B.12: **Most imputed values are positive despite allowing model parameters to have negative values.** The cell type, assay, and genome parameters from one of the 48 models trained on the ENCODE Pilot Regions plus non-coding human accelerated regions were used to impute values for 100000 randomly selected genomic positions, and these imputed values are plotted here as a histogram.

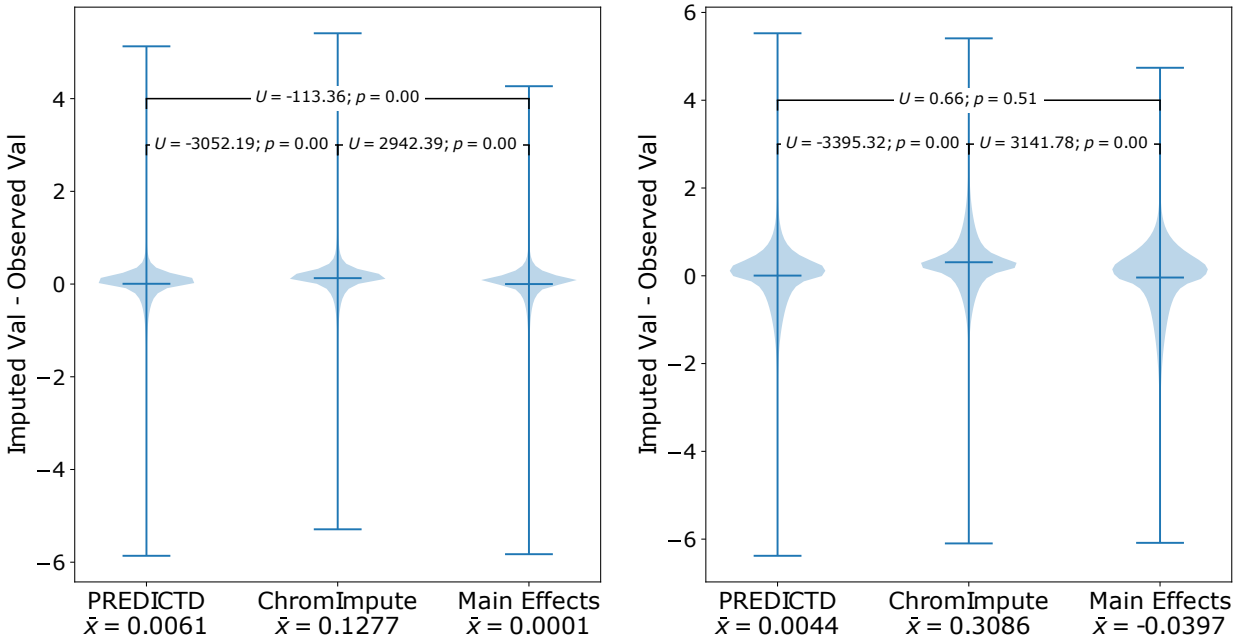

Figure B.13: **The error distribution of ChromImpute values is more positive than that of PREDICTD.** This means that on average ChromImpute tends to over-estimate signal amplitude compared to PREDICTD. Error distributions for each model were compared with the Mann-Whitney U test, and the sample mean is reported for each model on the x-axis. Violin plots show the distribution of error values, with a horizontal line at the mean and whiskers indicating the extrema. **a.** Random sample of 100,000 genomic positions. **b.** Top 100,000 genomic positions by summing all observed data values at each position.

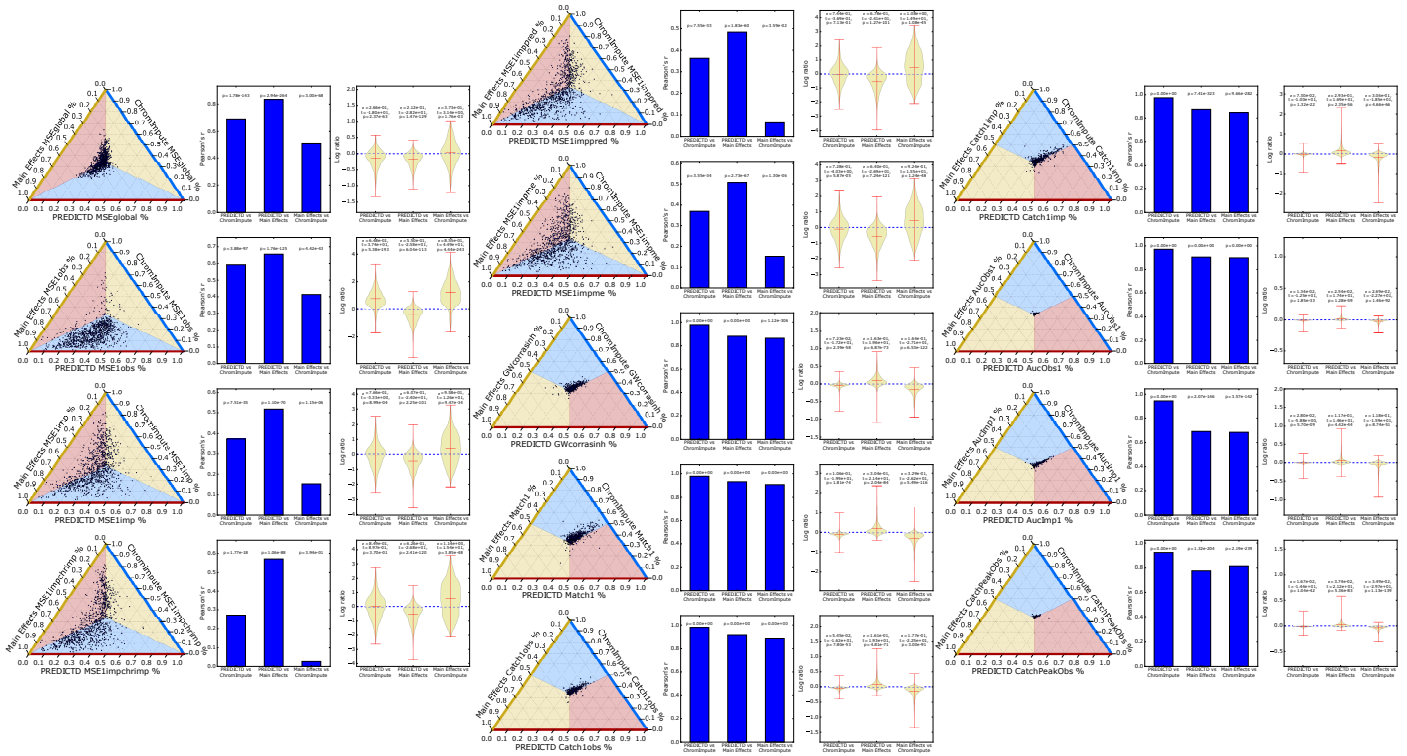

Figure B.14: **Comparison of all quality measures for PREDICTD, ChromImpute, and Main Effects models.** The first plot in each triplet is the ternary plot, as described in Fig. 3.4, the second is the Pearson correlation between the quality measure values for each pair of models, and the third is the distribution of the natural log fold-change in quality measure value between corresponding experiments in pairs of models. Each violin plot shows the distribution of the natural log fold-change values, with a horizontal line at the mean and whiskers indicating the extrema. Note that the correlation of PREDICTD with ChromImpute is always higher than the correlation between Main Effects and ChromImpute, indicating that PREDICTD tends to agree more with ChromImpute than Main Effects does. In addition, the mean log fold-change between PREDICTD and ChromImpute is always either closer to zero than the log fold-change between Main Effects and ChromImpute, indicating more comparable quality measure values between PREDICTD and ChromImpute, or the mean log fold-change indicates stronger performance by PREDICTD (MSEglobal, MSE1imp, and MSE1impme) than ChromImpute. In all, the quality measures show that PREDICTD performs very similarly to ChromImpute, and more so than Main Effects does.

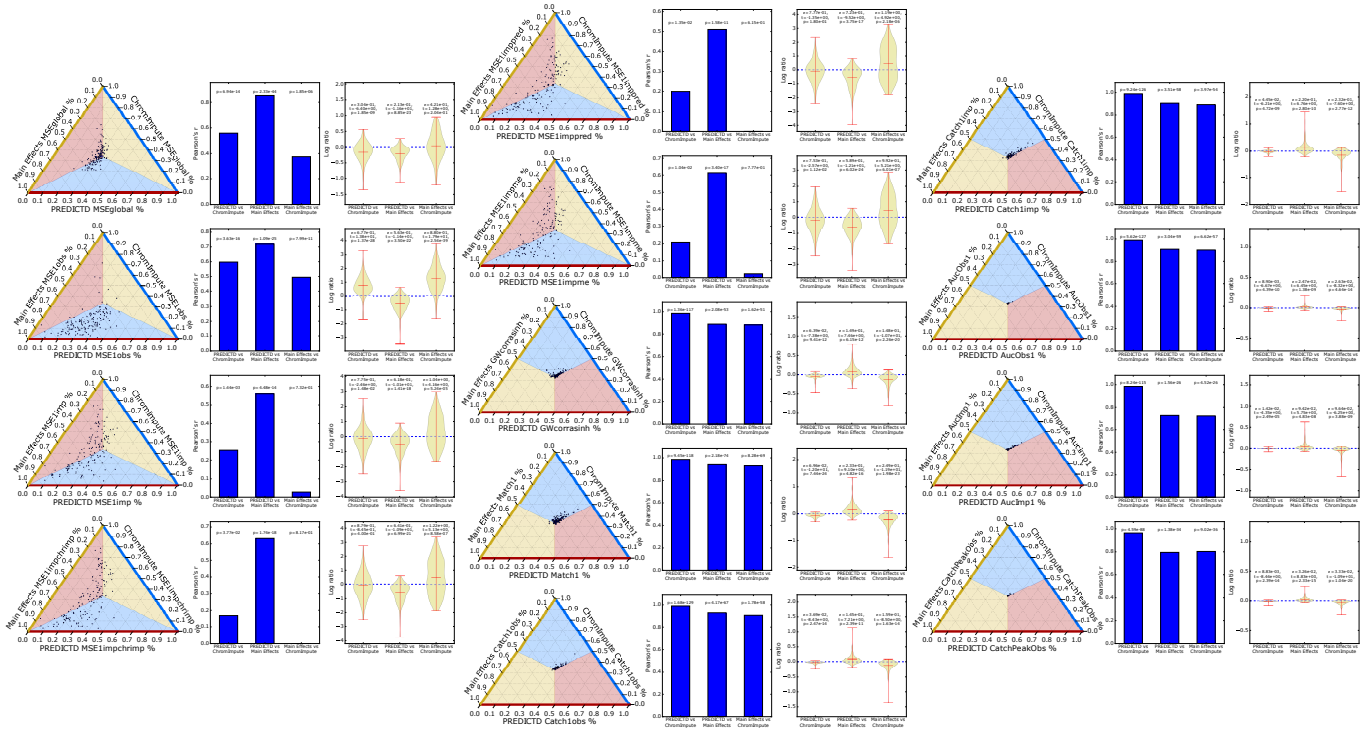

Figure B.15: **Comparison of all quality measures for PREDICTD, ChromImpute, and Main Effects models on just the 153 held out final test experiments that were not used in hyperparameter tuning.** The first plot in each triplet is the ternary plot, as described in Fig. 3.4, the second is the Pearson correlation between the quality measure values for each pair of models, and the third is the distribution of the natural log fold-change in quality measure value between corresponding experiments in pairs of models. Each violin plot shows the distribution of the natural log fold-change values, with a horizontal line at the mean and whiskers indicating the extrema. Note that the correlation of PREDICTD with ChromImpute is always higher than the correlation between Main Effects and ChromImpute , indicating that PREDICTD tends to agree more with ChromImpute than Main Effects does. In addition, the mean log fold-change between PREDICTD and ChromImpute is always either closer to zero than the log fold-change between Main Effects and ChromImpute, indicating more comparable quality measure values between PREDICTD and ChromImpute, or the mean log fold-change indicates stronger performance by PREDICTD (MSEglobal, MSE1imp, and MSE1impme) than ChromImpute. In all, the quality measures show that PREDICTD performs very similarly to ChromImpute, and more so than Main Effects does.

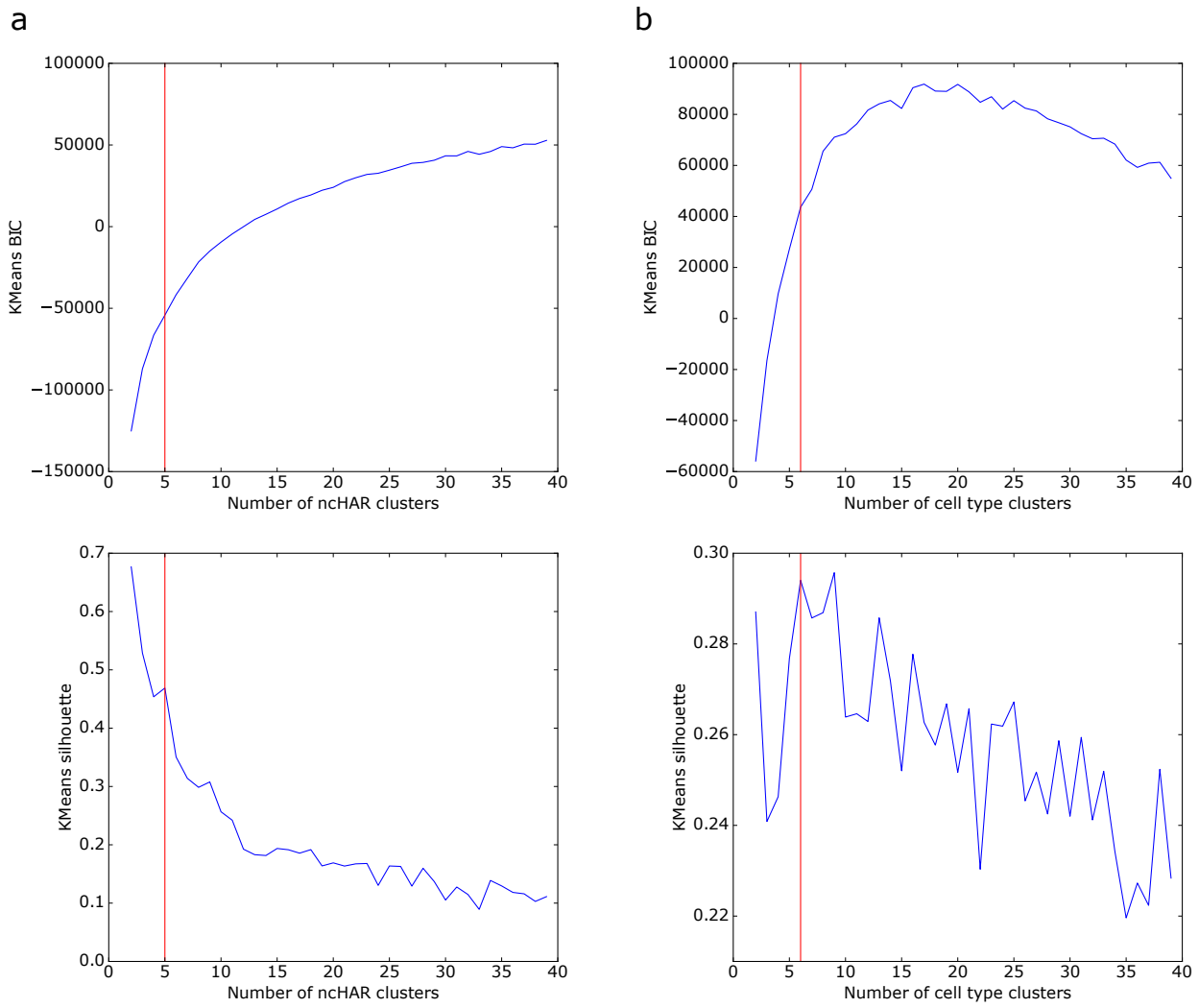

Figure B.16: **Elbow and silhouette analysis of imputed enhancer mark data over ncHARs supports the a choice of 5 ncHAR clusters and 6 cell type clusters.** To pick the number of ncHAR and cell type clusters, we used k-means clustering on the rows (ncHARs) and columns (cell types) of the biclustering input matrix (see Methods) for imputed data and conducted a Bayesian Information Criterion (BIC) "elbow" analysis, as well as a silhouette score analysis. Assessment of the quality of the clustering is based on finding a balance that achieves an appropriate number of clusters that are still well-separated. Elbow analysis based on BIC, as well as silhouette analysis suggests that, **a.** 5 is a reasonable number of clusters for ncHARs, and **b.** 6 is a reasonable number for cell types. The chosen cluster number is indicated by a vertical red line. In both cases the number of clusters is near the maximum of the second derivative in the elbow plots and also at a local maximum in the silhouette plots.

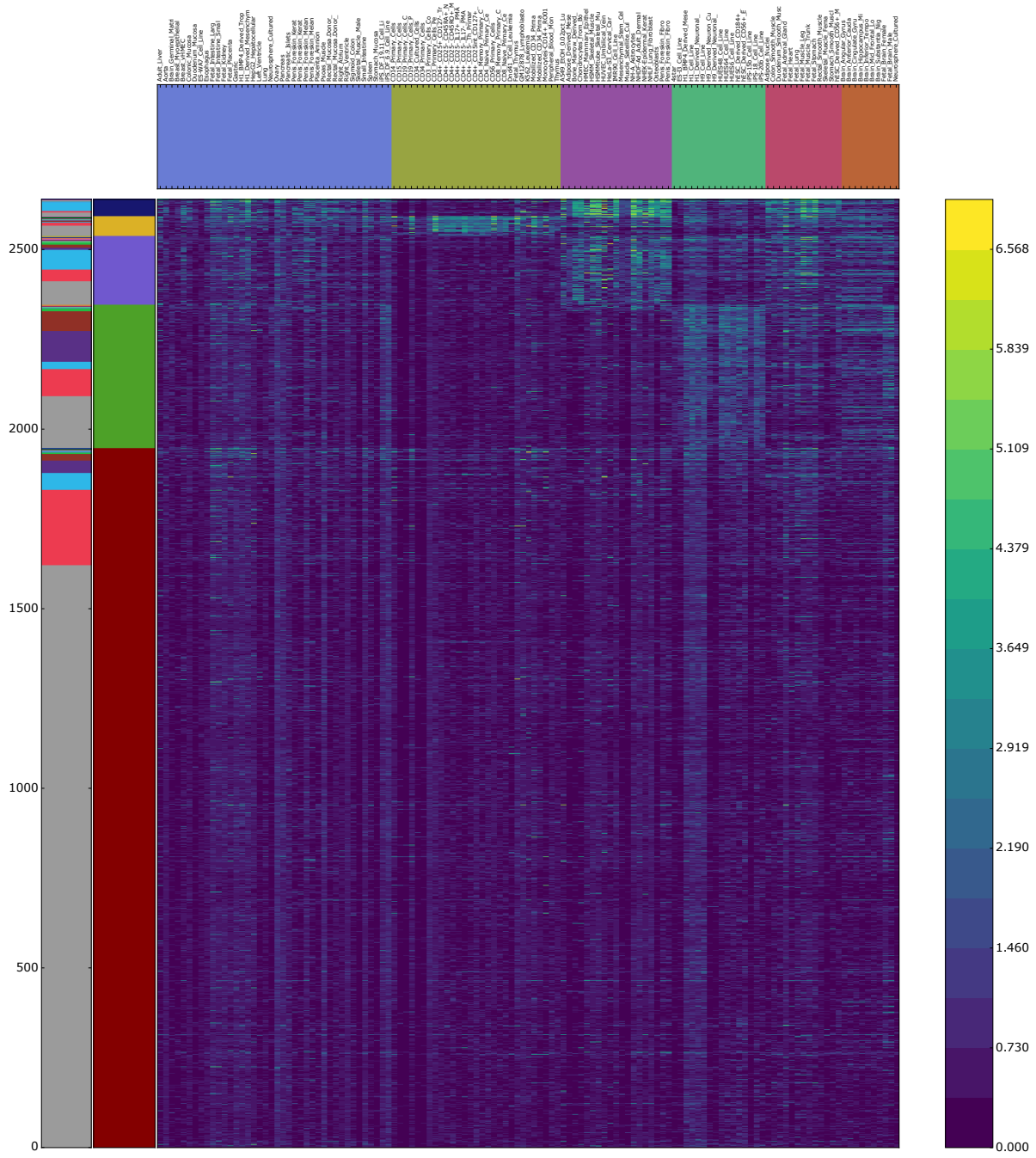

Figure B.17: **Heatmap showing biclustering results and signal from observed data at ncHARs for the H3K27ac, H3K4me1, and DNase assays.** The clustering results are very similar to those for imputed data reported in Fig. 3.5a.

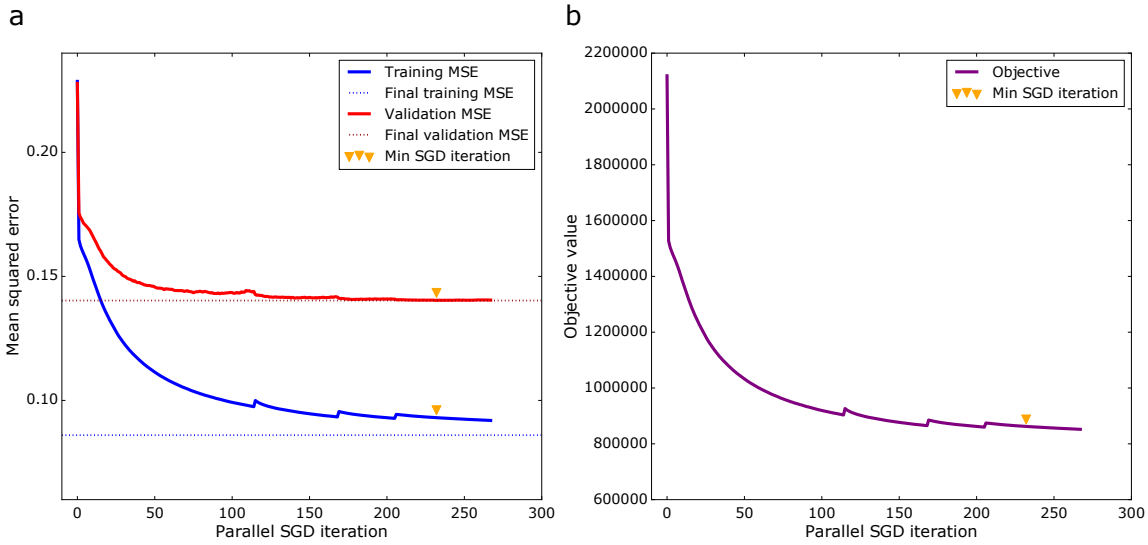

Figure B.18: **Training is halted before validation error increases. a.** Training and validation MSE as a function of the number of parallel SGD iterations during PREDICTD training. The initial vertical drop in the error corresponds to the burn-in phase of training before the parallel SGD iterations begin. The jags in the error curves indicate where the stopping criterion was met, so the training procedure reset the parameters to their values from the iteration with the previous minimum validation MSE, halved the learning rate, and continued training. Orange triangles indicate where the minimum validation MSE was achieved during parallel SGD, and the dotted lines indicate the model training and validation MSE after the final second order update of the genome parameters. The second order update decreased the validation MSE from 0.14034 to 0.14026. **b.** A similar plot of the objective value as a function of parallel SGD iterations shows that the objective value decreases and follows the same trend as the training error.

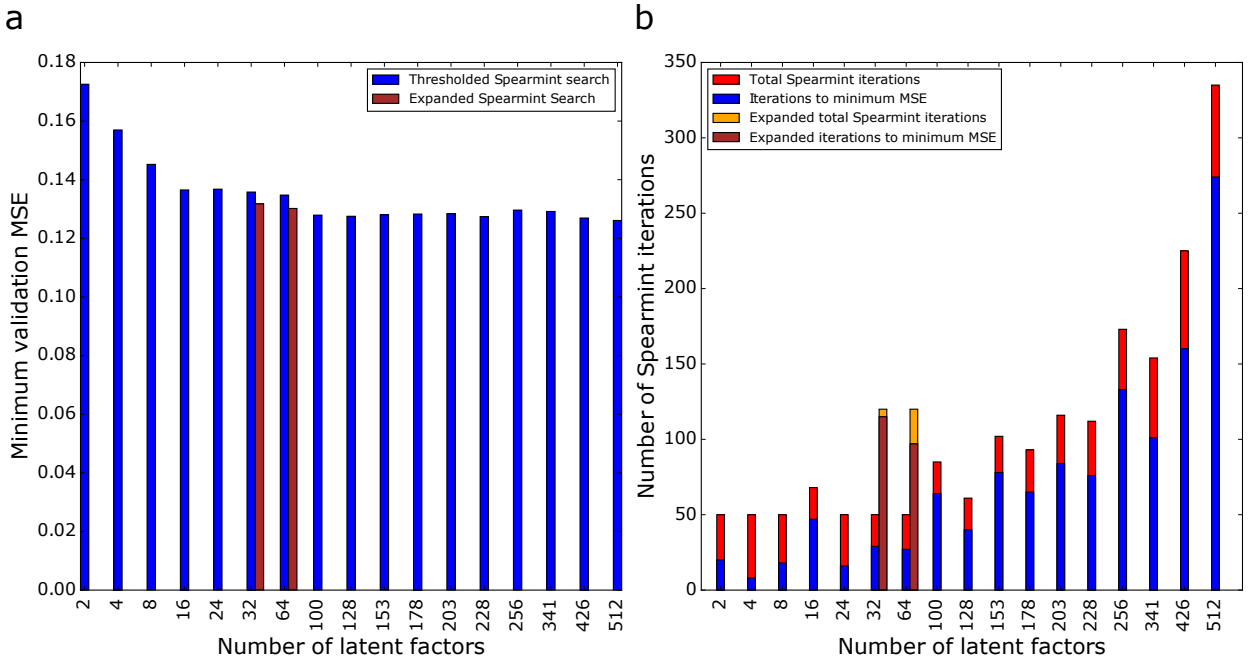

Figure B.19: **An extensive hyperparameter search supports selecting 100 latent factors as the model dimensionality for maximizing imputation performance. a.** Minimum validation error MSE for the best models found in each of 17 different Spearmint hyperparameter searches with different numbers of latent factors. The minimum validation MSE decreases as a function of increasing latent factor number until about 100 latent factors, suggesting that this dimensionality maximizes model performance while minimizing the redundancy of latent factors. Furthermore, if we allow the Spearmint hyperparameter search to continue until 120 iterations for 32 and 64 latent factors, we see that the 100 latent factor setting still finds a lower validation MSE. **b.** We required hyperparameter searches to get longer as the model dimensionality increased to allow sufficient time for Spearmint to search the solution spaces that become correspondingly more complex. We trained each level of latent factors for at least 50 Spearmint iterations or 40% of the number of latent factors, whichever was more, and only stopped Spearmint after it had additionally trained at least 20 iterations or 15% of the number of latent factors, whichever was more, past its best result (blue/red bars). We expanded the Spearmint search to 120 iterations for the 32 and 64 latent factor settings (orange/brown bars) to better resolve the solution space for the models that were only slightly less complex than our chosen setting of 100 latent factors.

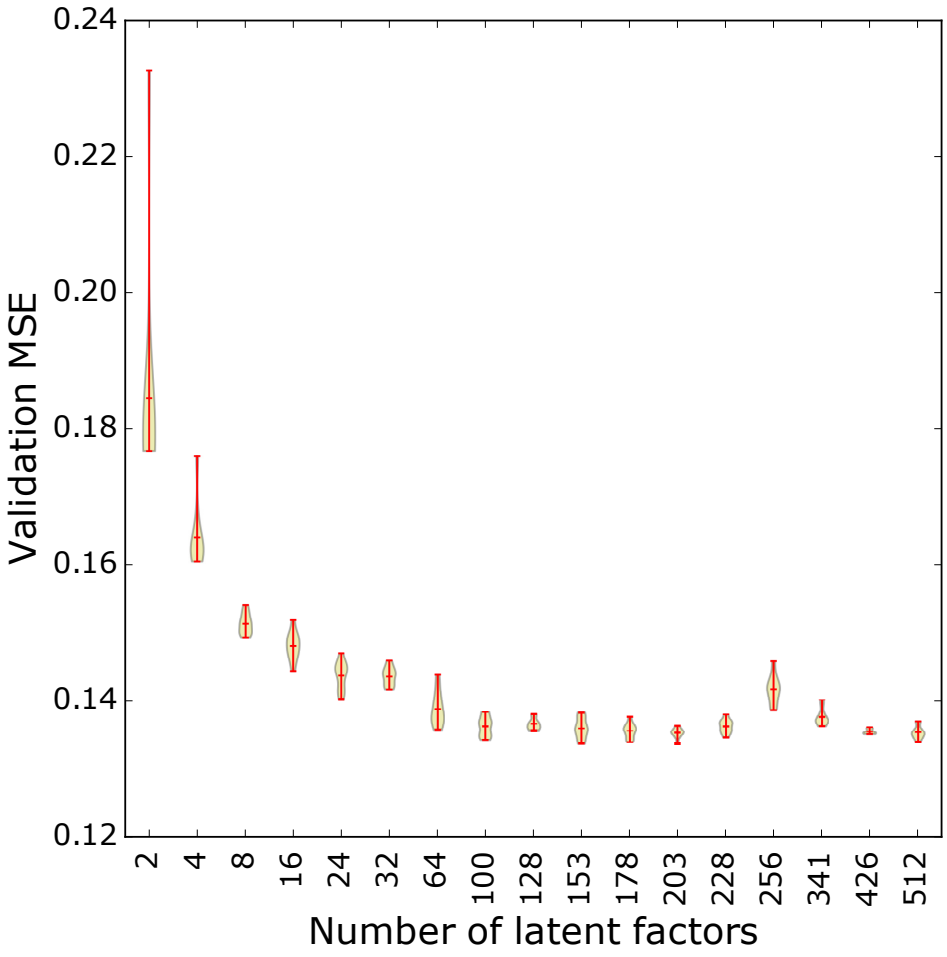

Figure B.20: **Training multiple models with different random initializations for each latent factor setting confirms the choice of 100 latent factors.** After doing an extensive hyperparameter search at 17 different latent factor settings (Supplementary Fig. B.19), we used the same training/validation sets and the same genomic positions for training all models, and trained ten models with different random seeds for the best hyperparameter settings for each latent factor setting. For the 32 and 64 latent factor models, we used the hyperparameter settings from the expanded Spearmint search (Supplementary Fig. B.19). The 64 latent factor distribution is greater than the 100 latent factor distribution by the Wilcoxon rank sum test with $p < 0.05$. The violin plots show the distribution of validation MSE for the ten trained models at each latent factor setting, with a horizontal line at the mean and whiskers indicating the extrema.

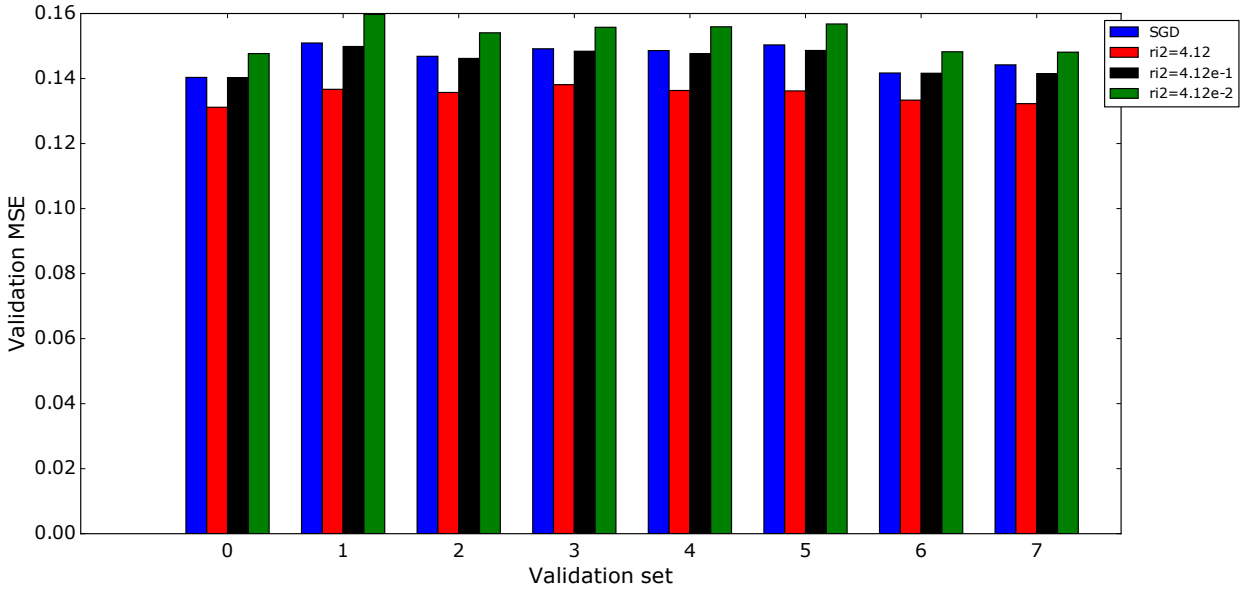

Figure B.21: **Averaging models provides additional regularization that can be balanced by reducing the regularization of the second order genome update.** The validation set MSE for the eight different validation folds corresponding to test set 0 after the second order genome update with different values of $\lambda_{\mathbf{G}_2}$. We chose $\lambda_{\mathbf{G}_2} = 0.412$ (black bars) as a value that imposed approximately the same amount of regularization as the stochastic gradient descent phase of training (blue bars) to avoid having too much regularization when we averaged the eight models together to calculate our final imputed values for the test set.

VITA

Timothy John Durham was raised in Guilford, CT. He attended Williams College for his undergraduate studies, where he earned a Bachelor of Arts in Biology and Computer Science. His undergraduate thesis in the lab of Dr. Claire Ting, entitled "Genomic, Proteomic, and Physiological Approaches Toward Understanding the Ecological Distribution of *Prochlorococcus* in the Open Oceans," earned highest honors. After graduating in 2009, he joined the J. Craig Venter Institute in Rockville, MD as a bioinformatics engineer working on eukaryotic genome annotation. In 2010 he moved to the Broad Institute of MIT and Harvard in Cambridge, MA, where he worked under Dr. Noam Shoresh and Dr. Charles Epstein in the Epigenomics Program, affiliated with the labs of Dr. Bradley Bernstein and Dr. Alexander Meissner. In 2013 he enrolled in graduate school in the Department of Genome Sciences at the University of Washington in Seattle, where he has been co-advised for the past five and a half years by Dr. William Noble and Dr. Robert Waterston, and where he performed the work here presented.